Dr.E Suresh Babu

Digitally signed by Dr.E Suresh Babu
DN: cn=Dr.E Suresh Babu, o=NIT Warangal, ou=NIT Warangal, email=esbabu@nitw.ac.in, c=IN
Date: 2020.12.15 13:02:46 +05'30'

Bluetooth Low Energy(BLE)

4.0 Bluetooth

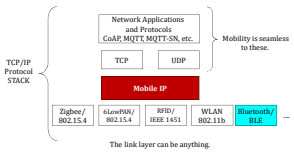
Dr. E.SURESH BABU
Assistant Professor
Computer Science and Engineering Department
National Institute of Technology, Warangal
Warangal

Session Outline

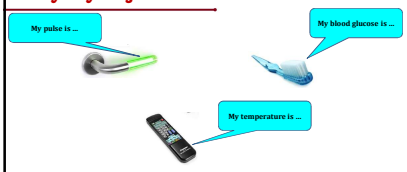
1. Introduction to Bluetooth Low Energy
2. BLE Architectural Overview
3. Controller : Physical, Link Layer and Direct Test Mode Sub Layer
4. Host Layer : L2CAP Layer.
5. Generic Attribute Profile, Generic Access Profile
6. Security Specifications and Applications

Introduction to Bluetooth Low Energy

IoT Protocol Stack.



Everyday objects can become sensors



... and monitor things remarkably

IoT Applications

Smart Homes

IoT Applications

Connected Camera | Wearable devices | Automobile Industry

...and many, many more

Short Range Wireless Application Areas

	Voice	Data	Audio	Video	State
Bluetooth ACL/HS	X	Y	Y	X	X
Bluetooth SCO/eSCO	Y	X	X	X	X
Bluetooth low energy	X	X	X	X	Y
Wi-Fi	(VoIP)	Y	Y	Y	X
Wi-Fi Direct	Y	Y	Y	X	X
ZigBee	X	X	X	X	Y
ANT	X	X	X	X	Y

State = Low Bandwidth, Low Latency Data

Low Power

Speeds(Data Rate) Almost Always Increase

Modems
V.21: 0.3kbps
V.22: 1.2kbps
V.32: 9.6kbps
V.34: 28.8kbps

Ethernet
802.3i: 10Mbps
802.3u: 100Mbps
802.3ab: 1000Mbps
802.3an: 10000Mbps

Wi-Fi
802.11: 2Mbps
802.11b: 11Mbps
802.11g: 54Mbps
802.11n: 135Mbps

Bluetooth
v1.1: 1Mbps
v2.0: 3Mbps
v3.0: 54Mbps
v4.0: 0.3Mbps

Problems with Bluetooth Classic (w.r.t Energy)

- Traditional Bluetooth is **Connection Oriented.**
  - ✓ When a **device is connected**, **a link is maintained**, even if there is **no data flowing.**
- **Sniff modes** allow **devices to sleep**
  - ✓ **Reducing power consumption to give months of battery life. But not more**
- **Peak transmit** current is typically around **25mA**
- **Better lower power** than **other radio standards**
  - ✓ it is still not **low enough power** for **coin cells** and **energy harvesting applications**

Whether Bluetooth Low Energy is Good for IoT Applications?

- **Connecting the things** we carry with us:
  - ✓ **Watches:** remote display from other devices
  - ✓ **Tags:** locate objects or keep track of them (e.g. warning if you walk away)
  - ✓ **Health & fitness sensors** (e.g. pedometer in your shoes)
  - ✓ **Body sensors** (e.g. blood pressure, pulse rate, blood glucose, etc)
- **Accessing the things** around us:
  - ✓ Use **proximity as a security/access control** means
  - ✓ **Home and office automation**

Whether Bluetooth Low Energy is Good for IoT Applications?

- **Low duty cycle M2M communication:**
  - ✓ **Sensors and controls** in homes, offices and factories
- **Communication within a system**
  - ✓ **Car to car wheels/tires**
  - ✓ **Connecting anything** that has intrinsic data to the Internet

Why Bluetooth Low Energy?

What is Bluetooth Low Energy

- **Bluetooth Low Energy** is a brand new technology designed as
  - ✓ Complementary technology to **Classic Bluetooth**
  - ✓ **Low Power Wireless Technology**
- **Bluetooth Low Energy** takes a **completely different direction**.
  - ✓ It has been optimized for **ultra-low power consumption.**
  - ✓ **Don't** get **High Data Rates,**
  - ✓ **Don't** get the **connection** for many hours or days.

Introduction

- **Bluetooth Low Energy** is designed to deploy **extremely high volumes of devices**
  - ✓ These **Devices** should be available **extremely low cost.**
- **For Example,**
  - ✓ **Radio frequency identification (RFID) tags** can be deployed in **very high volumes**, because they are **very low cost.**

Bluetooth Low Energy Radio

- The design for the **Bluetooth Low Energy Device** must include the following:
  - ✓ **Worldwide operation**
  - ✓ **Low cost**
  - ✓ **Robust**
  - ✓ **Short range**
  - ✓ **Lower power consumption**
  - ✓ **Lower complexity**
  - ✓ **Devices with coin cell batteries**
  - ✓ **Lower data rates**

What is Bluetooth Low Energy

- **Bluetooth Low Energy**
  - ✓ A **Wireless Protocol** for connecting **two devices.**
  - ✓ Uses **Open and license free standard**
  - ✓ Easily **integrated** within **existing Bluetooth technology**
  - ✓ Uses **Ultra-low Power Consumption**- Ideal for devices with **very low battery capacity**
    - **Button Cell** will be the **main power supply** for peripherals
      - **< 15ma peak current**
      - **< 1ua average current**

Reason for Low Cost.

- The **Reason of BLE System** is of low cost.
  - ✓ **ISM Band : 2.4GHz ISM band** is a terrible place to design and use a wireless technology
  - ✓ **Low Power : Low energy** is to work with **Button-cell Batteries**- the **smallest, cheapest**, and most **readily battery available**.

Device Types

Device Types

- **Bluetooth Low Energy** can be build in **two types of devices**:
  - ✓ **Dual-Mode Devices.**
  - ✓ **Single-Mode Devices.**

Dual-Mode Devices.

- A **Dual-mode Device** is a **Bluetooth device** that has support for both **Bluetooth classic** as well as **Bluetooth low energy.**
  - ✓ The **dual-mode device** can talk with the **billions of existing Bluetooth devices.**
  - ✓ **Dual-mode devices** are new devices that require **new hardware and firmware** in the **controller and software** in the host.
  - ✓ It is **not possible** in **existing Bluetooth** classic controller or host to **upgrade and support low energy**

Dual-Mode Devices.

- **Dual-Mode Devices** will acts as a **Central Device**, These devices can also acts as **Client,**
- Typically for **Master devices** -**Phones, PC, Tablet, PMP, TV**
  - ✓ More processing power
  - ✓ More memory
  - ✓ e.g. Smartphone and tablets

Central

Master
Client
Can read/write data to Slave/Server

Single-mode Device.

- A **Single-mode Device** is a **Bluetooth device** that only supports **Bluetooth low energy**.
- A new **single-mode devices** are **highly optimized** for **ultra-low power consumption**, that are powered by **button-cell batteries**
- A **single-mode device** with **BLE cannot talk** with the **existing Bluetooth devices**,
  - ✓ However, it can **talk** with other **single-mode devices** as well as **dual-mode devices.**

Single-mode Device.

- **Single-mode Bluetooth low energy** does not support **audio for headsets** and **stereo music** or **high data rates** for file transfers.
- Typically for **Sensors And Peripherals** (can act as **master to a small number of sensors**)

Single-mode Device.

- **Peripheral Device** is a **Single-mode Device, which acts as a Server**
  - ✓ Small
  - ✓ Low power
  - ✓ Resource constrained
  - ✓ Connects to a more powerful (central) device
  - ✓ E.g., Heart Rate Monitors, BLE proximity tags

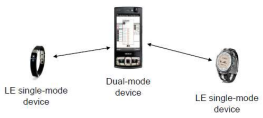
Peripheral

Slave
Server
Has read/write data

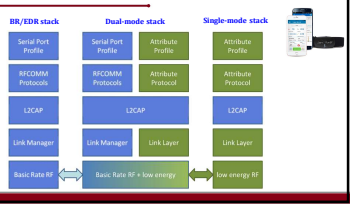
Single-Mode Vs Dual-Mode Vs Classic Compatibility

	Single-Mode	Dual-Mode	Classic
Single-Mode	LE	LE	none
Dual-Mode	LE	Classic	Classic
Classic	none	Classic	Classic

Device Modes: Dual mode + Single modes



Device Modes: Dual mode + Single modes



Architectural Overview

BLE vs. Classic BT

Spec.	BLE	BT Classic
Topology	Star	Scatternet
Throughput	0.7 – 2.1 Mbps	0.3 Mbps
Message	8 – 47 bytes	358 bytes
Power	< 15 mA	< 30 mA
Cost	Low	High
Range	50 – 150 m	< 30 m
Latency	3 ms	100 ms

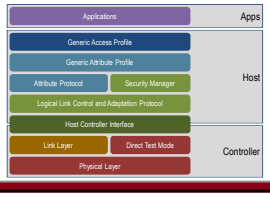
BLE Detailed Specification

Range:	**~ 150 meters open field**
Output Power:	**~ 10 mW (10dBm)**
Max Current:	**~ 15 mA**
Latency:	**3 ms**
Topology:	**Star**
Connections:	**> 2 billion**
Modulation:	**GFSK @ 2.4 GHz**
Robustness:	**Adaptive Frequency Hopping, 24 bit CRC**
Security:	**128bit AES CCM**
Sleep current:	**~ 1µA**
Modes:	**Broadcast, Connection, Event Data Models, Reads, Writes**

Terminology

- **Adaptive Frequency Hopping (AFH) :**
  - ✓ A technology whereby only a **subset of frequencies** is used.
  - ✓ This allows devices to **avoid the frequencies** that **other non-adaptive technologies** are using (e.g., a Wi-Fi access point).

Bluetooth Low Energy Architecture

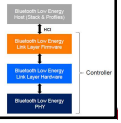


Bluetooth Core Specification

- **GATT** : Organization of Attributes into services
- **ATT** : Data exchange protocol
- **SMP** : Bonding, privacy and encryption
- **GAP** : Advertising, device discovery and connections
- **L2CAP** : Data multiplexing, fragmentation and reassembly
- **HCI** : Interface between controller and host
- **Link Layer** : Basic packets, state machine and radio control Link layer security and privacy
- **Radio** : Receives and transmits bits

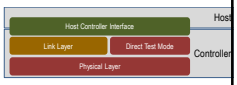
Bluetooth Low Energy Architecture

- The architecture for Bluetooth low energy is **fundamentally very simple.**
- This architecture is splits into **three basic parts:**
  1. **Controller,**
  2. **Host, And**
  3. **Applications.**



Controller:

- It is a **physical device** that can **transmit and receive radio signals**
- It understands, how these **signals can be interpreted** as **packets** with **information** within them.
- **Within the controller,**
  - ✓ **The Physical Layer and**
  - ✓ **Link Layer**
  - ✓ **Direct Test Mode**
  - ✓ **The Host Controller Interface**

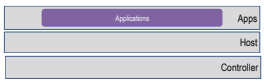


Host

- It is a **software stack** that manages how **two or more devices communicate** with **one another**
- It identifies and offers how **several different services** can be provided at the **same time over the radios.**

Bluetooth Low Energy Architecture

- **Applications:**
  - ✓ The applications use the **software stack**, and therefore the **controller, to enable a use case.**

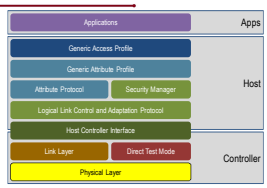


Physical Layer

Controller Layer:

- The **controller** can be identify as the **Bluetooth chip** or **radio.**
- The **controller** is composed of both **analog and digital parts** of the **radio frequency components**
  - ✓ **Hardware** to support the **transmission and reception** of packets.
- The **controller interfaces** with the outside world through an **antenna** and to the **host** through the **Host Controller Interface.**

Controller : Physical Layer

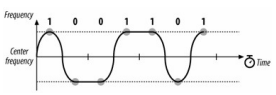


Physical Layer

- The **Physical Layer** is one of the sub layer of the **controller**
- It performs **transmitting and receiving of bits** using the **2.4GHz radio.**
  - ✓ The **2.4GHz ISM band** is a free for all for **anyone who wants to use** it.
  - ✓ This Band is also used by **Microwave Ovens, Bluetooth classic, Bluetooth low energy, IEEE 802.11, IEEE 802.11b, IEEE 802.11g, IEEE 802.11n, and IEEE 802.15.4.**

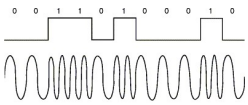
PL :Gaussian Frequency Shift Keying (GFSK).

- In BLE, the **frequency** of the **radio waves** are exposed using a **modulation scheme** called **Gaussian Frequency Shift Keying (GFSK).**



Why GFSK : Frequency Shift Keying

- The **Frequency Shift Keying** part means that **ones and zeros** are coded onto the radio by **slightly shifting** the **frequency up** and **down.**



Frequency Shift Keying

Why GFSK : Frequency Shift Keying

- If the frequency is shifted **abruptly to one side** or the **other** at the moment the **frequency changes**,
  - ✓ there is a **pulse of energy** that **spreads out** over a **wider range of frequencies**.
  - ✓ A **Filter** is used to **stop the energy spreading too far into higher or lower frequencies.**

Why Gaussian Frequency Shift Keying

- Without **GFSK modulation filter** , the **frequency shift would be dramatically quick**, **causing much noise** to be created.
- when changing from a **zero bit to a one bit**, the transition is both **fast and efficient.**

Gaussian Frequency Shift Keying

- The **GFSK modulation filter** is used to **Gaussian curves shape.** The GFSK filter used for **Bluetooth low energy.**
- **GFSK modulation filter** is one that optimizes the **transition** from **one symbol** to the **another** by **increasing the time**
  - ✓ It is used to **slide the frequency** from one value to another.
  - ✓ When transmitting data, BLE energy transmits at **one million bits per second (Mbps)**, with **one bit per symbol**.

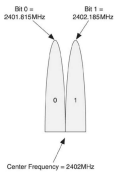
Modulation

- Using **GFSK filter modulation,**
  - ✓ The **radio signal** is **slightly widens**, which is referred to as the **Modulation Index.**

Modulation

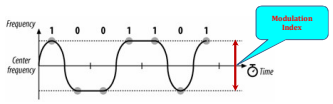
- To **send a zero**, a **negative frequency deviation** is used. To **send a one**, a **positive frequency deviation** is used.
- The **minimum frequency deviation** is about 180kHz.
  - ✓ if a **center frequency** of **2402MHz** is used,
  - ✓ A **zero** would be indicated by a transmission at **2401.820MHz**
  - ✓ A **one** would be indicated by a transmission at **2402.180MHz**

Modulation



Modulation Index.

- The **Modulation index** describes how wide the **upper and lower frequencies** used are around the **center frequency of a channel.**



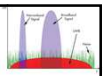
Modulation Index.

- The **mathematical representation** of frequency deviation by the **Modulation index (h),** which is the size of the deviation

$$h = \frac{\Delta f}{f_m}$$

  - ✓ Where **Δf** is the **maximum deviation from the carrier frequency**,
  - ✓ **f_m** is the **highest frequency in the source signal** being modulated onto the carrier.

Modulation Index.



- If the **modulation index** is **greater than one,**
  - ✓ The **carrier** is varying in **frequency more than the source signal**, then the signal is called a **wide-band transmission.**
- If the **modulation index** is **less than one**,
  - ✓ The **carrier** is varying in **frequency less than the source**, then the signal is called a **narrow-band transmission.**
- A **modulation index of 0.5** is considered a very special value, which is a **minimum-shift keying** or MSK and it is **very efficient**.

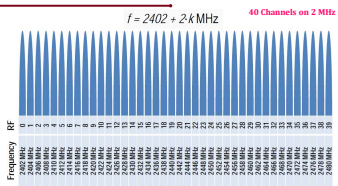
Frequency Band

- **Bluetooth Low Energy** uses the **2.4GHz ISM band** for **transmitting information**.
- **2.4GHz ISM band** can be used by **other radios** in the **same area transmitting** at the **same time,**
- The **2.4GHz band** is split up into **40 separate RF channels**,
  - ✓ Each **2MHz** apart from **one another.**

Radio Channels-Frequency Band

$f = 2402 + 2k$ MHz

40 Channels on 2 MHz spacing



RF	Frequency
0	2402 MHz
1	2404 MHz
2	2406 MHz
3	2408 MHz
4	2410 MHz
5	2412 MHz
6	2414 MHz
7	2416 MHz
8	2418 MHz
9	2420 MHz
10	2422 MHz
11	2424 MHz
12	2426 MHz
13	2428 MHz
14	2430 MHz
15	2432 MHz
16	2434 MHz
17	2436 MHz
18	2438 MHz
19	2440 MHz
20	2442 MHz
21	2444 MHz
22	2446 MHz
23	2448 MHz
24	2450 MHz
25	2452 MHz
26	2454 MHz
27	2456 MHz
28	2458 MHz
29	2460 MHz
30	2462 MHz
31	2464 MHz
32	2466 MHz
33	2468 MHz
34	2470 MHz
35	2472 MHz
36	2474 MHz
37	2476 MHz
38	2478 MHz
39	2480 MHz

Transmitter And Receiver Characteristics

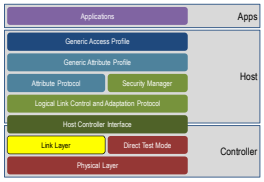
- **Transmit output power**
  - ✓ The maximum transmit power is +10dBm (10mW).
  - ✓ The Minimum transmit power is –20dBm (10µW).
- **Receive sensitivity**
  - ✓ **-70dBm** (-90dBm is expected performance) words- it has to be able to pick up **0.0000001mW of electromagnetic energy** to be able to work.

Session Outline

1. Introduction to Bluetooth Low Energy
2. **BLE Architectural Overview**
3. **Controller : Physical, Link Layer and Direct Test Mode Sub Layer**
4. **Host Layer : L2CAP Layer.**
5. **Generic Attribute Profile, Generic Access Profile**
6. **Security Specifications and Applications**

Link Layer

Link Layer



Controller : The Role of Link Layer

- The **Link Layer** is probably the **single most complex part** of the **Bluetooth low energy architecture**.
  - ✓ It is responsible for **advertising, scanning, and creating** and **maintaining connections.**
  - ✓ It is also responsible for ensuring the **packets are structured, Packet correctness and encryption sequences.**

Controller : The Role of Link Layer

- The **Link Layer** performs the **three basic operations**
  1. **Link Layer channels,**
  2. **Packets Format**
  3. **Procedures.**

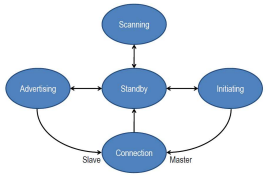
Controller : Link Layer

- The **Link Layer defines**
  - ✓ how **two devices** can use a **radio** to **transmit information** between **one another.**
  - ✓ **Details of a packets, Advertising and Data channels.**
  - ✓ Procedures for **discovering other devices**, **Broadcasting the Data,**
  - ✓ **Making and managing the Connections** and **sending the data** within connections.

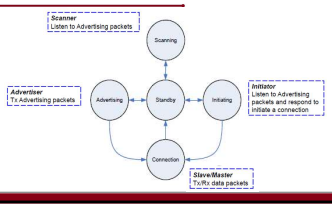
The Link Layer State Machine

- **Bluetooth low energy design** make use of **Link Layer state machine**, which is responsible for **creating and maintaining connections.**
- The **Link Layer state machine** defines just **five states:**
  1. **Standby**
  2. **Advertising**
  3. **Scanning**
  4. **Initiating**
  5. **Connections**

The Five State Machine



The Advertising State

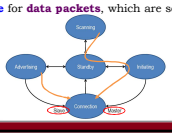


The Five State Machine

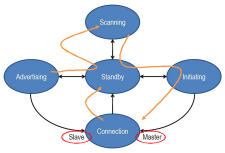
State		State Description
Standby		Does not transmit or receive packets
Advertising		Broadcasts advertisements in advertising channels
Scanning		Looks for advertisers
Initiating		Initiates connection to advertiser
Connection	Master	Communicates with device in the Slave role, defines timings of transmissions
	Slave	Communicates with device in Master Role

The Role of States

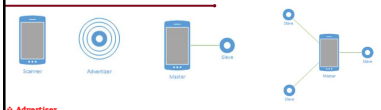
- The **Link Layer state machine** has
  - ✓ **Three states** for **advertising packets,** which are sent or received,
  - ✓ **One state** for **data packets**, which are sent and received.



The Standby State

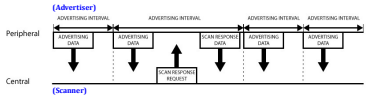


Connection between Advertiser and Scanner



- **Advertiser**
  - ✓ Only sends out advertisements i.e. Broadcast data
  - ✓ Can allow or disallow connections
- **Slave**
  - ✓ Connected to one master (BT 4.0)
  - ✓ Connected to multiple masters (BT 4.1)
- **Scanner**
  - ✓ Only listens for advertisements
  - ✓ Can also connect an advertiser
- **Master**
  - ✓ Connected to one or multiple slaves

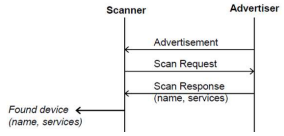
Connection between Advertiser and Scanner



The Scanning State

- In the **Scanning State**
  - ✓ A **Device** will receive **advertising channel packets**.
  - ✓ This **Scanning Device** will **simply listen** to see what **devices are advertising** in the local area.
  - ✓ It is only possible to move from the **scanning state** into the **standby state.** This is done by **stopping scanning.**

The Scanning State



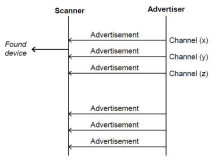
Types of Scanning

- **Scanning** is composed of **two different sub-states**:
  1. **Passive Scanning**
  2. **Active Scanning.**

Passive Scanning

- This Device has a **only receiver** performs the **passive scanning**
- The device just performs **passively scans**, never **transmitting anything.**
- **Passive scanning** only **receives advertising packets.**

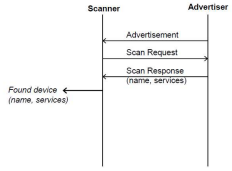
Passive Scanning



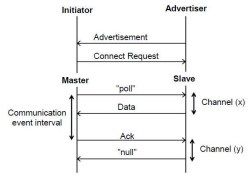
Active Scanning

- In **Active Scanning,** Whenever a **new device** is discovered by the **Link Layer,**
  - ✓ A **scan request** is sent to the **advertising device,** and a **scan response** is expected in reply.
  - ✓ The **scan requests and response packets** are transmitted on the **advertising channel.**
- **Active scanning** also **sends scan requests** to **advertising devices** to obtain additional **scan response data.**

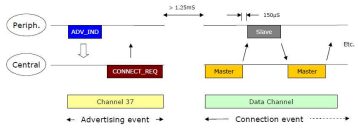
Active Scanning



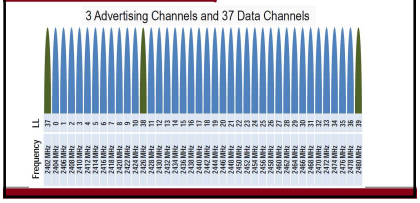
Initiating a Connection



Connection from Initiator



Physical Channels



The Connection State

- The **Final State** of the **Link Layer state machine** is the **connection state.**
- **Connection State** can be **entered** either via the **advertising state** or the **initiating state**.
- The **transitions should be initiated** by an **initiating device** sending a **connect request packet** to an **advertising device.**

Packets Structure

Access Address

- The **next 32 bits** of a **packet** are the **access address**.
- There are two types of **access address** :
  1. **Advertising Access Address**
  2. **Data Access Address**

Access Address

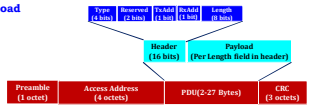
- The **advertising access address** is used
  - ✓ When **broadcasting data** or when **advertising, scanning, or initiating connections.**
- The **data access address** is used in a connection
  - ✓ After a **connection has been established** between two devices.

Payload or PDUs

- The **payload or PDUs** is the **actual real data** that is being transmitted.
- The **Payload** could be
  - ✓ **Advertising Data** about the **device** or
  - ✓ **Service data** that is being **broadcast to devices** in the local area.

Payload or PDUs

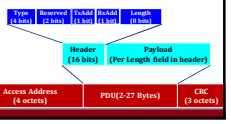
- The **payload Consists of Two Parts**
  1. **Header**
  2. **Payload**



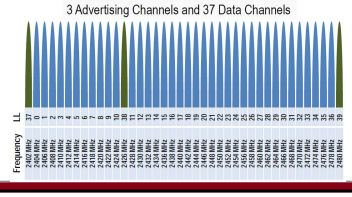
Advertising Packets (Header)

Type	Packet	Usage
0000	ADV_IND	Connectable undirected advertising event
0001	ADV_DIRECT_IND	Connectable directed advertising event
0010	ADV_NONCONN_IND	Non-connectable undirected advertising event
0011	SCAN_REQ	Scan request for further information from advertiser
0100	SCAN_RSP	Response to scan request from scanner
0101	CONNECT_REQ	Connect request by Initiator
0110	ADV_DISCOVER_IND	Discoverable undirected advertising event

- **Preamble** – frequency synchronization and AGC training (10101010)
- **Access Address** – 0x8e89bedd6
- **CRC** – computed over PDU
- **TxAdd, RxAdd** – PDU type-specific information



Physical Channels



Channels used by BLE Device

- In the Link Layer, the **channels** are **divided into two types**:
  1. **Advertising channels**
  2. **Data channels.**
- These **channel types** are aligned with the **advertising packets** and **data packets**
- There are **3 advertising channels** and **37 data channels**

Advertising channels

- When a **packet is transmitted,** if the packet is sent on an **advertising channel,** it is an **advertising packet**.
- The **advertising channels** are rendered in **darker shading**
- The **3 advertising channels** are not all placed in the **same part** of the **ISM band**
  - ✓ **Any deep fade** in a **single part of the band** would stop **all advertising.**
  - ✓ **Advertising channels** are placed a minimum of **24MHz apart** from **one another.**

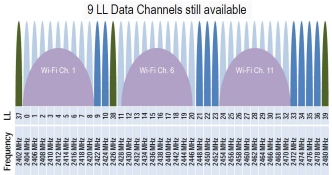
The Link Layer Channel Map

- In **Channel Map**
  - ✓ The **advertising channels** are placed strategically away from **significant interferers** such as a **Wi-Fi access point**.
  - ✓ **The Wi-Fi access point** typically use **one of three 802.11 channels**, either **channel 1, channel 6, or channel 11.**
    - **Center frequencies** of these channels are **2412MHz, 2437MHz, and 2462MHz** with width of **20MHz.**
    - **channel 1 extends from 2402MHz to 2422MHz**, **channel 6 extends from 2427MHz to 2447MHz**, and **channel 11 extends from 2452MHz to 2472MHz.**

The Link Layer Channel Map

- The **advertising channels** are placed at **2402MHz, 2426MHz, and 2480MHz.**
  - ✓ **First advertising channel** is below **Wi-Fi channel 1**,
  - ✓ **Second advertising channel** is between **Wi-Fi channel 1 and channel 6**,
  - ✓ **Third advertising channel** is **above Wi-Fi channel 11**.
- **The 3 advertising channels, 37, 38, and 39, are all interference free.**
- **3 Wi-Fi channels** have blocked the **use of data channels 0 to 8, 11 to 20, and 34 to 32.**

Advertising Channels avoid 802.11



Data Channel

- If the **packet** is sent on a **data channel**, then it is a **data packet**.
- The **advertising channels** are numbered from **37 to 39;**
- The **data channels** are numbered from **0 to 36**.
- The **separation** of the **data channel and advertising channel numbers will make** very easy to implement the **frequency-hopping algorithm**

Frequency Hopping

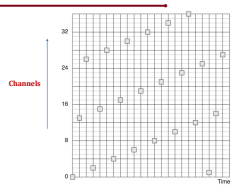
- When in a **data connection**, a **frequency-hopping algorithm** is used. Because
  - ✓ The **number of data channels is 37,** which is a prime number, the **hopping algorithm** is very simple,

$$f_{n+1} = (f_n + hop) \ mod \ 37$$

Frequency Hopping

- The **hop value** is a **value** that can **range from 5 to 16;**
  - ✓ This **value is added** onto **last frequency modulo 37** every time the **frequency-hopping algorithm** is used.
  - ✓ This means that **every frequency will be used** with **equal priority.**
  - ✓ The channels chosen, given a **hop value of 13**, are shown over time.

Frequency Hopping Of Data Channels Over Time



Direct Test Mode Sub Layer

Direct Test Mode

- **Direct Test Mode** is a **novel approach** to the **testing of the Physical Layer.**
- In most wireless standards, there is **no standard way** to get a **device to perform standard Physical Layer tests.**
  - ✓ This leads to the **problem** of many **different companies** building their **own proprietary methods** to **test only their Physical Layers,** which **increases the cost**

Direct Test Mode

- **Direct Test Mode** allows a **tester** to **command** a **controller's Physical Layer** either
  - ✓ **Transmit a sequence** of **test packets**
  - ✓ **Receive a sequence of test packets.**
- The **tester** can then **analyse the packets received**, or the **number of packets** that the **device under test received**, to determine
  - ✓ if the **Physical Layer** is **working** according to the **specification**

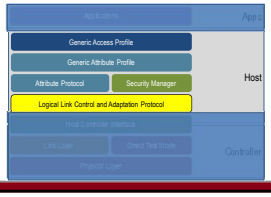
Host- Controller Interaction

Host Controller Interface

- The **HCI** provides a **uniform interface** method of accessing **Bluetooth Controller's capabilities** (command **PHY and LL, Access Hardware Status, Control Registers**)
- Possibility of realizing **separate Host and Controller**
  - ✓ **Interoperability of different subsystems**
- It allows a **host to send commands** and **data to the controller** and the **controller to send events** and **data to the host.**

Host Layer

Host Layer

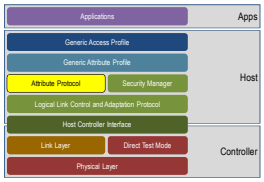


Purpose of Host Layer

- **Logical Link Control and Adaptation Protocol** is a **multiplexing layer**
- The **Security Manager** performs the **authentication and setting up secure connections**
- The **Attribute Protocol** exposes the **state of data on a device.**
- **Generic Attribute Profile** defines how the **Attribute Protocol** is used to enable **reusable services** that expose the **standard characteristics of a device.**
- The **Generic Access Profile** defines how **devices find and connect** with one another in an **interoperable manner.**

Attribute Protocol

Attribute Protocol



Attribute

- **Attribute** is "data"
  - ✓ **Value with a meaning (UUID), Permissions (Read/Write),** that is addressable by **a Handle**
  - ✓ **Attributes** expose data on a **Remote Device**
  - ✓ **ATT** is a **peer-to-peer protocol** between **a server and a client**
    - ***Server***: contains attributes, receives requests, executes, responds, can indicate values
    - ***Client***: sends requests, commands, waits for responses, can confirm indications

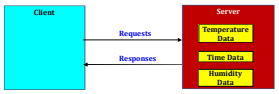
The Attributes of Attributes

Handle	Type	Value	Meaning
0x0009	«Device Name»	0x54656d706572261747572 652053656e736f72	"Temperature Sensor"
0x0022	«Battery State»	0x04	Discharging
0x0098	«Temperature»	0x0802	20.5 °C

The Attributes of Attributes

- **Attributes** have ***values***
  - ✓ Array of up to **512 octets, fixed or variable length**
- **Attributes** have ***Handles***
  - ✓ Used to **address an individual attribute** by a **client**
- **Attributes** have a ***Type***
  - ✓ **<<UUID>>**, determines what the **value means**
  - ✓ Defined by **GAP, GATT, "Characteristic Specifications"**
- **Attributes** have ***Permissions***
  - ✓ **Read, Write**
  - ✓ May require **authentication or authorization** to read or write

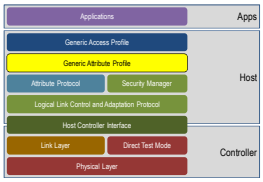
Process of ATT with Client and Server



- **Client Server Architecture**
  - ✓ Servers Have Data
  - ✓ Clients Request Data To/From Servers
- **Servers expose data using Attributes**

Generic Attribute Profile

Generic Attribute Profile



Generic Attribute Profile

- The **Generic Attribute Profile** sits above the **Attribute Protocol.**
- **Generic Attribute Profile** defines
  - ✓ Defines the **types of attributes** and how **they are used.**
  - ✓ It introduces a **number of concepts**, including **"characteristics", "services", "relationships between services".**
  - ✓ It also defines a **number of procedures** that can be used **to discover the services, characteristics, and relationships between services.**
  - ✓ The **read and write characteristic** values.

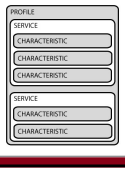
GATT Profiles

- The advantage of **standardized profiles** is **interoperability.**

Profile:
- Predefined collection of Services, compiled by either Bluetooth SIG or a peripheral designer.

Service:
- A collection of characteristics.
- Unique ID called UUID (16 bit – 128 bit)

Characteristic:
- Encapsulates a single data point.
- Can be used for sending data back to a peripheral as we can write into a characteristic.



GATT Profiles

- A **GATT profile groups** conceptually related **attributes (variables)** and defines the **format of their content** and how they can be **identified and accessed.**
- The **GATT specification** defines **usage profiles**
  - ✓ **BLE profiles** are **'Battery Service', 'Current Time Service', 'Health Thermometer Profile', 'Heart Rate Profile', 'HID over GATT', 'Proximity Profile', and 'Phone Alert Service',**

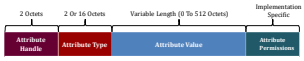
Profiles and Services

Profiles
Network Availability
Proximity
Find Me
Soft Button
Battery
Glucose Meter
Heart Rate Belt
Weight Scale
Device Power
Light Switch
HID
Watchdog

Services
Network Availability
Immediate Alert
Tx Power
Link Loss Alert
Generic Control
Battery
Glucose Meter
Manufacturer
Time
Heart Rate
Weight Scale
HID
Watchdog

GATT uses Attribute Protocol

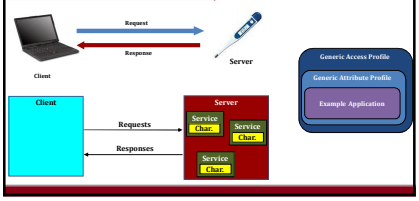
- **Attribute Protocol** defines a server with a **set of attributes**
  - ✓ Addressable with a **handle**
  - ✓ Typed using a **UUID**
  - ✓ Includes **data in an attribute value**
  - ✓ Includes **permissions**



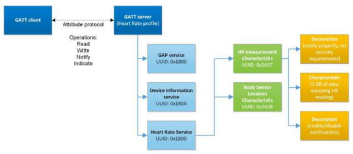
GATT Attribute Grouping

Handle	Type	Value	Permissions
0x0001	«Primary Service»	«GAP»	R
0x0002	«Characteristic»	{r, 0x0003, «Device Name»}	R
0x0003	«Device Name»	"Temperature Sensor"	R
0x0004	«Characteristic»	{r, 0x0006, «Appearance»}	R
0x0006	«Appearance»	«Thermometer»	R
0x000F	«Primary Service»	«GATT»	R
0x0010	«Characteristic»	{r, 0x0012, «Attribute Opcodes Supported»}	R
0x0012	«Attribute Opcodes Supported»	0x00003FDF	R
0x0020	«Primary Service»	«Temperature»	R
0x0021	«Characteristic»	{r, 0x0022, «Temperature Celsius»}	R
0x0022	«Temperature Celsius»	0x0802	R*

Client Server Architecture

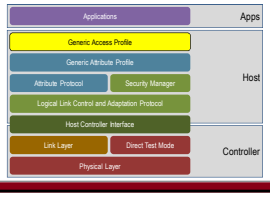


Data Exchange Between Client and Server



Generic Access Profile

Generic Access Profile



The Generic Access Profile

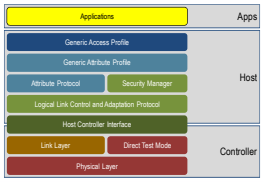
- The **Generic Access Profile** defines
  - ✓ How **devices discover, connect, and present useful information** to the **users.**
  - ✓ It also defines how **devices can create a permanent relationship,** called **Bonding.**
- The **GAP profile** defines how **devices** can be **discoverable, connectable, and bondable**.

The Generic Access Profile

- **GAP** also describes, how **devices can use procedures**
  - ✓ **To discover other devices**
  - ✓ **Broadcast data (beacons),**
  - ✓ **Connect to other devices,**
  - ✓ **Read the device name,**
  - ✓ **Bond with other device.**

Applications

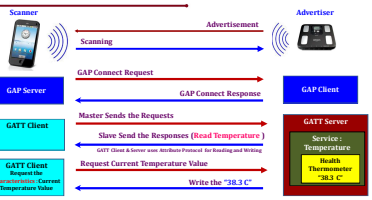
Applications



The Application Layer

- **Application Layer** is above of the **controller and the host.**
- The **Application Layer** defines **three types of specifications**:
  - ✓ **Characteristic**
  - ✓ **Service**
  - ✓ **Profile.**
- Each of these **specifications** is built on **top of the Generic Attribute Profile.**

How BLE Works



Observation

- **Bluetooth low energy** uses a **Client / Server model**.
  - ✓ A **Client (that "wants data")** connects and accesses **one or several Servers (that "has data").**
  - ✓ The **Client** typically **operates** in the **Central role** and the **Server operates** in the **Peripheral role.**
  - ✓ **A Sensor** or **An Accessory** is the **Server / Peripheral**
  - ✓ **Computer, Phone, Or Tablet** is the **Client / Central device**
  - ✓ The **Gateway** typically takes on the **Client / Central role**

Thank U

Bluetooth Low Energy Over IPv6

Dr. E.SURESH BABU
Assistant Professor
Computer Science and Engineering Department
National Institute of Technology, Warangal
Warangal

Session Outline

1. **BLE Over IPv6 Introduction**
2. **The Protocol Stack of BLE Over IPv6**
3. **Mobile Gateways**
4. **BLE Over IPv6 Mechanism**
5. **Use-Cases**

BLE Over IPv6

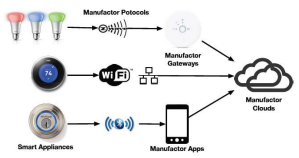
Introduction to BLE Over IPv6

- **Bluetooth Low Energy (BT-LE)** is
  - ✓ Rapidly **emerging ultra-low-power radio technology**
  - ✓ Can be **incorporated in billions** of **IoT devices**
  - ✓ Definitely **a strong contender** for **many useful IoT applications.**
  - ✓ To run **IPv6 over BT-LE opens** new **doors to the IoT** and leverages **BT-LE** towards **new application areas**.

IP-based protocol stack on BLE devices(1/2)

- For **Internet of Things (IoT)** applications, **only low power is not everything.**
  - ✓ The **ability to internetwork** is **more important**
  - ✓ **Carrying Internet protocol (IP) traffic** is necessary.
  - ✓ **Internetworking** capable **Bluetooth LE Protocol Stack** is required.
- Already, **IEEE 802.11 and 802.15.4 based ZigBee** have the lead in **networking with IP.**

IP-based protocol stack on BLE devices (2/2)



Challenges(1/3)

- **BT-LE** for **Internet connectivity** and **applications** poses challenges
  1. Beyond **IPv6 packet transport**, including **gateway operation**, **application protocol efficiency**, and **security.**
  2. To **interoperable** and **communicate** with the **devices of the IoT**
     - **BLE devices** cannot **simply use** their **ordinary BLE packets.**

Challenges(2/3)

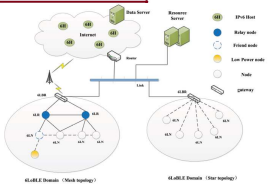
4. **Billions of devices** are **connected to the Internet today.** The **problem with the Internet** is that it is **built around a connection oriented model.**
   - A **TCP connection** is a **session-oriented channel** that is **established between two devices**, which takes **time to set up**

Challenges(3/3)

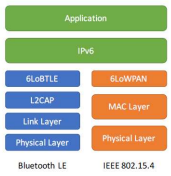
5. **Running IPv6** protocol over **low power radio technologies** incurs
   - ✓ Has **large overheads**, with a **header size of 40 bytes.**
   - ✓ **Communication overhead** to **establish and maintain connections.**
   - ✓ The **protocols** running on **top of IPv6** might not be **optimised for low-power applications**, which may **add overhead.**

The Protocol Stack of BLE Over IPv6

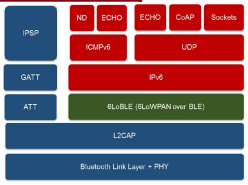
BLE Over IPv6



BLE Over IPv6 Vs IEEE 802.15.4 Over IPv6



The Protocol Stack of BLE Over IPv6



The Protocol Stack of BLE Over IPv6

- To **connect to the Internet** with **Bluetooth LE, Three additional pieces** must be **added:**
  1. **BLE Stack**
  2. **The Internet Protocol Support Profile (IPSP),**
  3. **An IPv6 stack**
  4. **The 6LoWPAN layer.**

The Protocol Stack of BLE Over IPv6

- **Bluetooth Stack**:
  - ✓ **GATT-based profile Internet Protocol Support Profile (IPSP)**
  - ✓ **L2CAP Connection Oriented Channels as bearer for 6Lo traffic**
- **Adaptation layer:**
  - ✓ **6LowPAN- 6LoBLE: Optimization of IPv6 packets for Bluetooth**
- **IPv6 Stack:**
  - ✓ **ICMPv6, UDP, CoAP, ND, etc.**

BT-LE Device Addressing

- Every **BT-LE device** is identified by a **48-bit device address**. The **devices** are **identified** using this **device address**.
- **Device addresses** may be either a
  - ✓ **Public Device Address**
  - ✓ **Random Device Address**.

BT-LE Device Addressing

- The **public device addresses** are based on the **IEEE 802-2001 standard**.
- The **Random Device Addresses** are generated as defined in the **BLUETOOTH specification**.
- The **device addresses** are always **unique within a BT-LE piconet**, but the **random addresses** are not guaranteed to be **globally unique.**

Issue with IPv6 Packets

- The **L2CAP channel characteristics** for the **transmission of IPv6 packets** on **top of BT-LE** are the following:
  - ✓ **IPv6** requires **support** for an **MTU of 1280 bytes.**
  - ✓ Whereas **BT-LE Packets Sizes and Maximum size** of the **payload** in the **BT-LE data channel PDU** is **27 bytes.**

Adaptation Layer: 6LoWPAN (1/3)

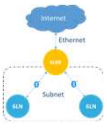
- The **6LoWPAN** is an **adaptation layer**
  - ✓ **Enables IPv6 communication** on **BLE link layers.**
  - ✓ **Methods** to **compress ordinary IPv6 trace** to the **needs of BLE nodes**.
  - ✓ Provides **useful functionality** for **reducing overhead,** which can be **applied to BT-LE**.
  - ✓ **Encapsulate the packets** before **forwarding them down** to the **L2CAP layer**.

Adaptation Layer: 6LoWPAN (2/3)

- The **6LowPAN / IPv6** operates in **parallel** with the **GATT profile**.
  - ✓ **IPv6 support** requires the **use of the 6LoWPAN protocol**
  - ✓ The **6LoWPAN protocol** is added directly on top of the **Bluetooth low energy L2CAP protocol layer.**
  - ✓ Using **6LoWPAN protocol, IPv6** can be used **end-to-end** from the **local Bluetooth low energy device** to an **Internet service.**

6LoWPAN Layer : Node Types

- **6LoWPAN Border Router (6LBR)**
  - ✓ IPSP Router and GAP Central
  - ✓ Manages multiple connections with 6LNs
  - ✓ Routes and forwards packets between 6LNs and to and from the Internet
  - ✓ Prevents address collision
- **6LoWPAN Node (6LN)**
  - ✓ IPSP Node and GAP Peripheral
  - ✓ Communicate with other 6LN only via the 6LBR and using prefix
  - ✓ Joins the all-nodes multicast address



6LoWPAN Functionality

- **6LoWPAN Functionality** comprises of
  - ✓ **Link-local Ipv6 Addresses**
  - ✓ **Stateless Ipv6 Address Auto-configuration**,
  - ✓ **Neighbour Discovery**
  - ✓ **Header Compression**.

Stateless Address Auto-Configuration

- **IPv6 addresses** are composed of a **prefix** and an **Interface Identifier (IID).**
- A **BT-LE 6LN (BLE Node)** performs **stateless address auto-configuration**.
  - ✓ A **64-bit Interface identifier (IID)** for a **BT-LE interface** may be formed by utilizing the **48-bit BLUETOOTH device address.**

Stateless Address Auto-Configuration

- The **IPv6 link-local address** is formed by appending the **IID to the link-local unicast prefix fe80::/64**

```
  10 bits        54 bits                 64 bits
+----------+-----------------+-----------------------+
|1111111010|      zeros      |  Interface Identifier |
+----------+-----------------+-----------------------+
```

IPv6 Link-local Address in BT-LE

- **Global IPv6 addresses** are formed by **prepending a valid prefix to the IID.**

Neighbour Discovery in BT-LE

- To enable **IPv6 over BT-LE** includes the use of a **simplified version of 6LoWPAN ND.**
- **Neighbour Discovery Optimization** for **6LoWPANs** describes the **neighbour discovery**.
- The **multi-hop functionality** in **6LoWPAN ND** (which comprises **multi-hop duplicate address detection**.
- In BT-LE, **a slave (IoT device-6LR)** and **a master (6LBR)** play the roles of Client and Server.

Header Compression

- **Header compression**
  - ✓ Specifies the **compression format** for **IPv6 datagrams** on top of **BLE.**
  - ✓ Introduced in **6LoWPAN over Bluetooth Low Energy** increases **over all performance** of **IPv6 connection establishment** and **maintenance.**

First Issue:

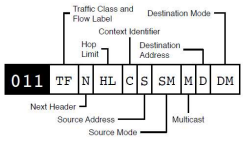
- A standard **IPv6 header** is **40 bytes**, while **Bluetooth Low Energy** have **small MTUs** of **27 bytes**
  - ✓ **IPv6 addresses** dominate the **header**
    - There are **context-based** and **context-free compression schemes** for **unicast and multicast IPv6 addresses.**
    - Other fields within **IP and UDP headers** may be compressed.
  - ✓ **BLE** often send **small data payloads** (e.g., **10 bytes**). **6LoWPAN** therefore provides **mechanisms** to **heavily compress IPv6 packets.**

Solution

- **6LowPAN** can **compress** a **40-byte IPv6 header** to **two(2) bytes**:
  - ✓ One to specifies how to **derive addresses** and **other fields**,
  - ✓ One containing the **hop limit.**
- For **small payloads** typical in **low-power devices**, **6LoWPAN** can **reduce the overhead** of **IPv6** from **400% to 20%.**

Header Compression

- **The IPv6 headers** MUST be **compressed** according to encoding formats.



BLE Over IPv6 Mechanism

BLE Network

- The **simplest subnet (BLE Network)** which **exchanges data** using **IPv6 over BLE links.**
  - ✓ It consists of **six node devices**,
  - ✓ **Each of Five Nodes** acts as a **BLE peripheral device**
  - ✓ **One Single Border Router**, acting as a **BLE central device**.
- Each of the **node devices** is connected to the **border router** via an **individual BLE data connection**.

Border Router

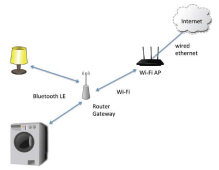
- The **border router** in **IPv6 over BLE networks** acts as an **IPv6 router** and **not as a link-layer switch.**
- **Nodes connected** to the **same border router** uses the **shared prefix** to **exchange data.**

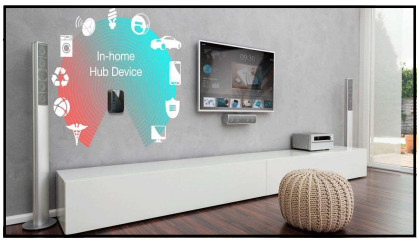
Link-Local IPv6 Address

- **BLE Devices** generate their **link-local IPv6 address** based on the **48-bit Bluetooth device address (MAC Permanent Address).**
- A **64-bit IID (interface identifier)** is formed and an **IPv6 address prefix is appended**
- **Non-link-local IPv6 addresses (Random Address)** do not embed the **Bluetooth device address** in their **64-bit IID by default.**
  - ✓ Instead, **cryptographic, hash-based**, or **other privacy enabling methods** are used **to create the IID.**

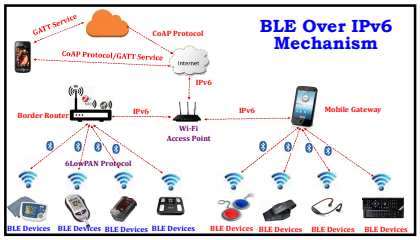
Use-Cases: BLE Over IPv6

Example : IPv6 with Bluetooth LE.





Example : IPv6 with Bluetooth LE using Mobile Gateway.



Thank U

Network Technology : IEEE 802.15.4 Standard

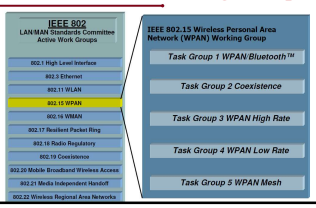
Dr. E.SURESH BABU
Assistant Professor
Computer Science and Engineering Department
National Institute of Technology, Warangal
Warangal

Session Outline

1. Overview IEEE 802.15.4
2. IEEE 802.15.4 Architecture
3. IEEE 802.15.4 Physical Layer
4. IEEE 802.15.4 MAC Layer
5. IEEE 802.15.4 MAC Access Mechanisms
6. IEEE 802.15.4 Device Types
7. 802.15.4 Addresses
8. The Network Topologies

Overview IEEE 802.15.4

IEEE 802.15 WPAN Working Group



Comparison between WPAN

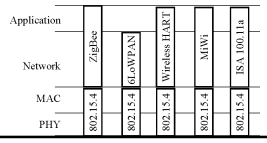
IEEE standard	Topic	Data throughput	Suitable applications	QoS needs
802.15.1	Bluetooth	1Mbp	Cell phones, Computers, Personal Digital Assistants (PDAs)/Handheld Personal Computers (HPCs), printers, microphones, speakers, headsets, bar code readers, sensors, displays, pagers, and cellular & Personal Communications Service (PCS) phones.	QoS suitable for voice applications
802.15.2	Coexistence of Bluetooth and 802.11b	N/A	N/A	N/A
802.15.3	High-rate WPAN	>20Mbps	Low power, low cost solutions for portable consumer of digital imaging and multimedia applications	Very high QoS
802.15.4	Low-rate WPAN	< 0.25 Mbps	Industrial, agricultural, vehicular, residential, medical applications, sensors and actuators with very low power consumption and low cost	Relaxed needs for data rate and QoS

Low-Rate Wireless Personal Area Network (LR-WPAN)

- **LR-WPAN** standard, to be called **802.15.4.**
- **802.15.4 Standard** was defined by **IEEE 802.15 task group.**
  - ✓ The standard was first **published in 2003**, then **revised in 2006.**
- **IEEE 802.15.4** provides
  - ✓ **Low-data-rate Communications** in an **Adhoc Self-organizing Network.**

IEEE 802.15.4

- **IEEE 802.15.4** used by several **"Internet of Things" protocols:**
  - ✓ **ZigBee, 6LowPAN, Wireless HART, MiWi, and ISA 100.11a**



IEEE 802.15.4 Features

- The **IEEE 802.15.4** standard **main features** are
  - ✓ **Network Flexibility,**
  - ✓ **Low Cost,**
  - ✓ **Low Power Consumption**
  - ✓ **Ultra-low complexity,**
  - ✓ **Low-data-rate**
  - ✓ Suitable for **Inexpensive Fixed, Portable**, And **Moving Devices.**

IEEE 802.15.4 Applications

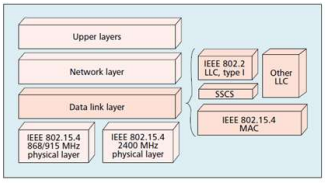
Application Scenarios
- Home Networking
- Automotive Networks
- Industrial Networks
- Interactive Toys
- Remote Metering
- . . .

General Characteristics of IEEE-802-15-4

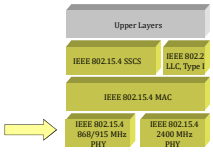
Property	Range
Raw data rate	868 MHz: 20 kb/s; 915 MHz: 40 kb/s; 2.4 GHz: 250 kb/s
Range	10–20 m
Latency	Down to 15 ms
Channels	868/915 MHz: 11 channels 2.4 GHz: 16 channels
Frequency band	Two PHYs: 868 MHz/915 MHz and 2.4 GHz
Addressing	Short 8-bit or 64-bit IEEE
Channel access	CSMA-CA and slotted CSMA-CA
Temperature	Industrial temperature range –40 to +85 C

IEEE 802.15.4 Architecture

IEEE 802.15.4 in the OSI-ISO Layer Network Model



Architecture



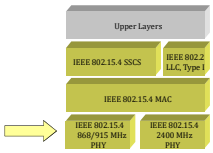
IEEE 802.15.4 Offers

IEEE 802.15.4

The IEEE standard 802.15.4 offers physical and media access control layers for low-cost, low-speed, low-power wireless personal area networks (WPANs)

IEEE 802.15.4 Physical Layer

Architecture



Physical Layer

- **IEEE 802.15.4** makes use of **PHY Options** that combine with the **MAC** to **enable** a **broad range of networking applications**.
- The **Physical Layer(PHY)** standard provides **two different services**
  1. **PHY Data Service**
  2. **Management Service**

Services offers by Physical Layer

- The **PHY data service**
  - ✓ **Controls the radio** to perform the **transmission and reception of the PPDUs.**
- The **PHY Management Service** performs
  - ✓ **Energy Detection in the channel,**
  - ✓ **Clear Channel Assesment** before sending the messages
  - ✓ Provides **LQI for the Received packets.**

PHY Functionalities

- **Activation and Deactivation** of the **radio transceiver**
- **Energy Detection** within the **current channel**
- **Link quality indication** for **received packets**
- **Clear Channel Assessment** for **CSMA-CA**
- **Channel Frequency Selection**
- **Data Transmission and Reception**

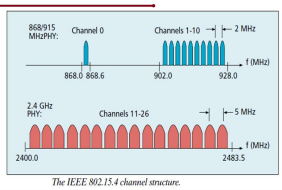
802.15.4 Data Rates

- The **2.4 GHz PHY** provides a **transmission rate** of **250 kb/s**
  - ✓ **Higher rate of the 2.4 GHz PHY** can be used to **attain higher throughput**, **lower latency**, or **lower duty cycle.**
- The **868/915 MHz PHY** offers **transmission rates of 20 kb/s** and **40 kb/s** for its **two bands**, respectively.

802.15.4 Frequency Bands

- The standard specifies **two PHYs :**
  1. **868 MHz/915 MHz Direct Sequence Spread Spectrum (DSSS) PHY (11 channels)**
     - **1 channel (20Kb/s)** in **European 868MHz band**
     - **10 channels (40Kb/s)** in **915 (902-928)MHz ISM band**
  2. **2450 MHz Direct Sequence Spread Spectrum (DSSS) PHY (16 channels)**
     - **16 channels (250Kb/s)** in **2.4GHz band**

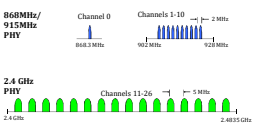
Channelization



The IEEE 802.15.4 channel structure.

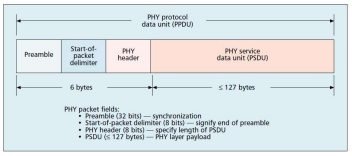
Channelization

- **Twenty-seven frequency channels** are available across the **three bands**
- **Channels:**



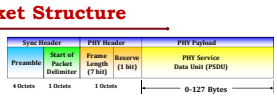
Packet Structure

Packet Structure



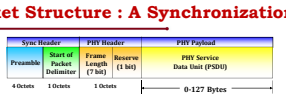
IEEE 802.15.4 physical layer packet structure.

Packet Structure



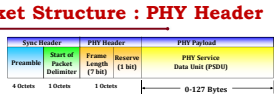
- Each **Packet, or PHY protocol data unit (PPDU),** contains
  1. **A Synchronization Header**
  2. **PHY header**
  3. **PHY Payload**

Packet Structure : A Synchronization Header



- **Synchronization Header** contains
  1. **Preamble (32 bits)** – **Used for Synchronization**
  2. **Start of Packet Delimiter (8 bits)** – shall be **formatted as "11100101"**

Packet Structure : PHY Header



- **PHY Header** contains
  1. **Frame Length (7-Bit)- Specify the Length of the Payload (In Bytes).**

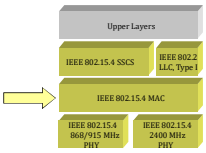
Packet Structure : PHY Payload

- **PHY Payload** contains

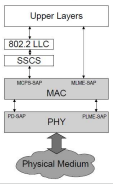
  1. **PSDU (0 to 127 bytes) – Data Field**

IEEE 802.15.4 MAC Layer

Architecture



IEEE 802.15.4 MAC Layer



IEEE 802.15.4 MAC

- The **IEEE 802.15.4 MAC** provides **services**
  - ✓ **IEEE 802.2 type-I LLC** through the **Service-Specific Convergence Sublayer (SSCS)**, or
  - ✓ **A proprietary LLC** can **access the MAC services** directly **without going** through the **SSCS**.
- The **802.15.4 MAC** allows the **more complex network topologies.**

Service-Specific Convergence Sublayer (SSCS

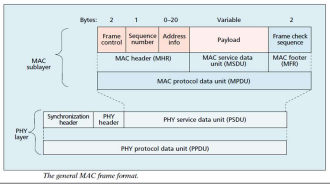
- The **SSCS** ensures
  - ✓ **Compatibility** between **different LLC sublayers**
  - ✓ **Allows the MAC** to be accessed through a **Single Set of Access Points.**

IEEE 802.15.4 MAC Features

- **Association** and **Disassociation**,
- **Acknowledged Frame Delivery,**
- **Channel Access Mechanism,**
- **Frame Validation,**
- **Guaranteed Time Slot Management**, and
- **Beacon Management**.

MAC Frame Format

General MAC Frame Format



The general MAC frame format.

MAC Frame Format

- In 802.15.4, the **MAC** maintains a **simple protocol**.
  - ✓ The **MAC frame structure** is kept **very flexible** to accommodate the needs of **different applications** and **network topologies.**

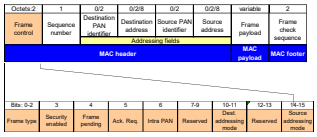
MAC Frame Format

- The **MAC frame** is called the **MAC protocol data unit (MPDU)** and is composed of
  1. **The MAC header (MHR),**
  2. **MAC Service Data Unit (MSDU),**
  3. **MAC Footer (MFR).**

The MAC Header (MHR)

- The **first field** of the **MAC header** is the **frame control field.**
- **Frame Control Field** indicates
  - ✓ The **type of MAC frame** being **transmitted,**
  - ✓ Specifies the **format of the address field**, and
  - ✓ **Controls the acknowledgment.**
  - ✓ In short, the frame control field specifies how the **rest of the frame looks** and **what it contains**.

The MAC Header (MHR): Frame Control Field



Frame Control Field

The MAC Header (MHR) : Address Field

- The **size of the address** field may vary between **0 and 20 bytes.**
  - ✓ For instance, A **data frame** may contain both **source and destination information**, while the **return acknowledgment frame** does not contain **any address information** at all.
  - ✓ On the other hand, a **beacon frame** may **only contain source address information.**

The MAC Header (MHR) : Address Field

- **Short 8-bit device addresses** or **64-bit IEEE device** addresses may be used.
  - ✓ This **flexible structure** helps **increase the efficiency** of the **protocol** by **keeping the packets short.**

The MAC Header (MHR) : SQ Field

- The **sequence number field** is used to **match the acknowledgment frames** with the **previous transmission**
- The **transaction** is considered **successful only**
  - ✓ when the **acknowledgment frame** contains the **same sequence number** as the **previously transmitted frame.**

MAC Service Data Unit (MSDU) : The Payload Field

- **The Payload Field** is **variable in length**;
  - ✓ But, the **complete MAC frame** may not **exceed 127 bytes** in length.
  - ✓ The **data** contained in the **payload** is **dependent** on the **frame type.**

MAC Layer Frame Formats

- The **IEEE 802.15.4 MAC** has **four different frame types**.
  1. **Beacon Frame,**
  2. **Data Frame,**
  3. **Acknowledgment Frame,** and
  4. **MAC Command Frame.**

Purpose of MAC Layer Frame Formats

- Only the **data** and **beacon frames** actually contain **information sent** by **higher layers**;
- The **acknowledgment** and **MAC command frames** originate in the **MAC** and are used for **MAC peer-to-peer communication**.

MAC Footer (MFR): Frame Check Sequence (FCS).

- The **FCS** in an **IEEE 802.15.4 MAC frame** is a **16-bit cyclic redundancy check (CRC).**
- The **FCS** helps **verify the integrity** of the MAC frame.

Data Frame Format

Data Frame Format

Octets:2	1	4 to 20	variable	2
Frame control	Data sequence number	Address information	Data payload	Frame check sequence
MAC header			**MAC Payload**	**MAC footer**

Data Service

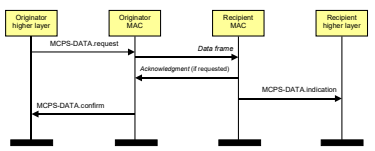
- **Data transfer** to neighboring devices
  - ✓ **Acknowledged or unacknowledged**
  - ✓ **Direct or indirect**
  - ✓ **Using GTS service**
- **Maximum data length (MSDU) 102 bytes**

Acknowledgement Frame Format

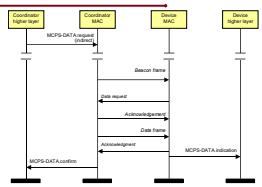
Acknowledgement Frame Format

Octets:2	1	2
Frame control	Data sequence number	Frame check sequence
MAC header		**MAC footer**

Data Transfer : Message Sequence Diagram



Indirect Data Transfer : Message Sequence Diagram



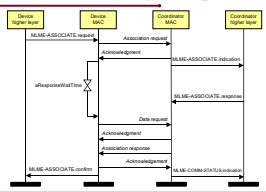
MAC Command Frame

MAC Command Frame

Octets: 2	1	4 to 20	1	variable	2
Frame control	Data sequence number	Address information	Command type	Command payload	Frame check sequence
MAC header			MAC payload		MAC footer

- **Command Frame Types**
  - ✓ **Association request**
  - ✓ **Association response**
  - ✓ **Disassociation notification**
  - ✓ **Data request**
  - ✓ **PAN ID conflict notification**
  - ✓ **Beacon request**
  - ✓ **Coordinator realignment**
  - ✓ **GTS request**

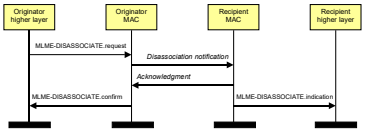
Association : Message Sequence Diagram



Association : Message Sequence Diagram

- A **node joins the network** by **sending an association request** to the **coordinator's address**.
  - The **association request** specifies the **PAN ID** that the **node wishes to join**
- In its response,
  - The **coordinator** assigns a **16-bit short address** to the **device** or specifies the **reason for failure** (access denied or lack of capacity).

Disassociation : Message Sequence Diagram



Disassociation : Message Sequence Diagram

- The **device** and the **coordinator** can issue a **disassociation request** to **end the association.**
- When a **device loses its association** with its **parent** (e.g., it has been **moved out of range**),
  - it sends **orphan notifications** (a **frame composed of a MAC header, followed by the orphan command code**).

Thank U

Session Outline

1. Overview IEEE 802.15.4
2. IEEE 802.15.4 Architecture
3. IEEE 802.15.4 Physical Layer
4. IEEE 802.15.4 MAC Layer
5. **IEEE 802.15.4 MAC Access Mechanisms**
6. **IEEE 802.15.4 Device Types**
7. **802.15.4 Addresses**
8. **The Network Topologies**

IEEE 802.15.4 MAC Access Mechanisms

MAC Access Mechanisms

- An **LRWPAN** may use **One of Two Channel Access Mechanisms.**
  1. In a **beacon-enabled network with super frames**, **a Slotted (CSMA-CA)** mechanism is used.
  2. In **networks without beacons**, **Un-slotted or Standard CSMA-CA is used.**

The Beacon-Enabled Access Method (or Slotted CSMA/CA).

The Beacon-Enabled Access Method (or Slotted CSMA/CA).

- In a **Beacon-enabled Network**,
  - Any **device** wishing to **transmit** during the **contention access period** waits for the **beginning of the next time slot.**
  - **Determines** if **another device** is currently transmitting in the **same slot.**
  - If **another device** is already **transmitting in the slot**,
    - ✓ The device **backs off for a random number of slots** or indicates a **transmission failure** after some retries.

The Beacon-Enabled Access Method (or Slotted CSMA/CA).

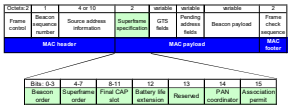
- In a **Beacon-enabled Network**,
  - **Acknowledgment frames** of these networks **do not use CSMA.**
  - An important function of the **MAC** is **confirming successful reception** of a **received frame**.
  - **Successful reception** and **validation** of a **data or MAC command frame** is confirmed with an **acknowledgment.**
  - If the **receiving device** is **unable to handle** the **incoming message** for any reason, the **receipt is not acknowledged.**

The Super-Frame Structure

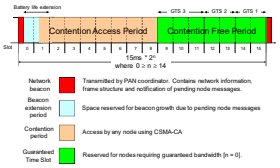
Need of Super-Frame

- **Some applications** may require **dedicated bandwidth** to achieve **low latencies**.
- To accomplish these **low latencies**,
  - ✓ **IEEE 802.15.4 LR-WPAN** can operate in an **Optional Super-Frame Mode.**

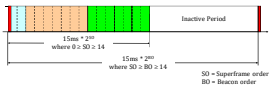
IEEE 802.15.4 Super-Frame Format



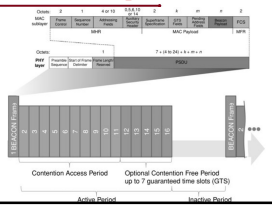
IEEE 802.15.4 Super-Frame Structure



Super frame may have Inactive Period

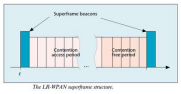


IEEE 802.15.4 Super-Frame Structure



The Super-Frame Structure

- In a **Super frame,**
  - ✓ A **PAN coordinator**, transmits **super frame beacons** in **predetermined intervals.**



The LR-WPAN superframe structure.

The Super-Frame Structure

- These **intervals** can be as short as **15 ms** or as long as **245 s.**
  - ✓ The **time** between **two beacons** is divided into **16 equal time slots** independent of the **duration of the super frame**.
  - ✓ A **device** can **transmit** at **any time** during a **slot**, but must **complete its transaction** before the **next super frame beacon**.

The Super-Frame Structure

- The **size of the contention-free period** may vary **depending on demand** by the **associated network devices;**
- when **GTS are employed**, **all devices** must **complete their contention-based transactions** before the **contention-free period begins.**
- The **beginning** of the **contention-free period** and **duration of the super frame** are **communicated** to the **attached network devices** by the **PAN coordinator** in its **beacon.**

When Beacon-Enabled Access Method is Used

- The **PAN Coordinator** periodically **broadcasts a super frame** composed of
  - A **Starting and Ending Beacon Frame**,
  - **15 Time Slots,** and
  - An **optional inactive period** during which the **coordinator may enter a low-power mode**

CAP or CFP in Beacon-Enabled Access

- The **First Time Slots** define
  - ✓ The **Contention Access Period (CAP),** during which the **other nodes should attempt to transmit using CSMA/CA**.
  - ✓ The **last N (N ≤ 7) time slots** form the **optional contention free period (CFP),** for use by **nodes requiring deterministic network access or guaranteed bandwidth.**

Beacon-Enabled Access Method using CAP or CFP

- **Devices** willing to **send data** to a **co-ordinator**
  - First, **listen** to the **super frame beacon**,
  - **Synchronize to the super frame** and
  - **Transmit data** either during the **CAP using CSMA/CA**, or **during the CFP.**

The Non-Beacon Enabled Access Method (Unslotted CSMA/CA).

The Non-Beacon Enabled Access Method (Un-Slotted CSMA/CA).

- In a **Non-beacon-enabled Network**,
  - First, the **device checks** if any **another device** is **currently transmitting** on the **same channel**.
  - If so, the **device may back off** for a **random period**, or **indicate a transmission failure** if **unsuccessful after some retries.**

The Non-Beacon Enabled Access Method (Un-Slotted CSMA/CA).

- **All nodes** access the **network using CSMA/CA.**
- This is the mode used by **ZigBee and 6LoWPAN**.
- The **coordinator** provides a **beacon only,** when **requested by a node**
  - ✓ **Sets the beacon-order (BO) parameter** to **15** to indicate use of the **non-beacon-enabled access method.**

IEEE 802.15.4 Device Types

IEEE 802.15.4 Device Types

- **802.15.4 networks** are composed of **several device types**:
- There are **two different device types** :
  1. **PAN coordinator node**
  2. **A Full Function Device (FFD)**
  3. **A Reduced Function Device (RFD)**

PAN Coordinator Node

- **802.15.4 Networks** are setup by a **PAN coordinator node** or simply **coordinator**.
  - ✓ It is identified by **PAN ID** for **each network**
  - ✓ Responsible for **scanning the network**
  - ✓ Selecting the **Optimal RF channel**
  - ✓ **Selecting the 16 bits PAN ID** for the network.
  - ✓ Other **802.15.4 nodes** must send an **association request** for this **PAN ID** to the **PAN coordinator** in order to become part of the **802.15.4 network**.

Full Function Device (FFD) Device Type

- **Full Function Devices (FFD)** can be one of the **coordinators**
  - ✓ **Devices** are capable of **relaying messages** to other **FFDs**
- The **FFD can operate** in three modes by serving as

Full Function Device (FFD)
- Any topology
- PAN coordinator capable
- Talks to any other device
- Implements complete protocol set

Reduced Function Device (RFD) Device Type

- The **RFD can operate**

Reduced Function Device (RFD)
- Reduced protocol set
- Very simple implementation
- Cannot become a PAN coordinator
- Limited to leafs in more complex topologies

Reduced Function Devices (RFD)

- It cannot **route messages**.
  - ✓ Usually their **receivers** are **switched off** except during **transmission.**
- It can be **attached to the network** only as **leaf nodes**.

Reduced Function Devices (RFD) : Methods

- **Two topology models** are used within each network for **data-transfer**
  1. **The Star Topology**: **Data Transfers** are possible only between the **PAN coordinator** and the **devices.**
  2. **Peer To Peer Topology**: **Data Transfers** can occur between **any two devices**. However, it is possible only networks comprising **permanently listening devices.**

FFD vs RFD

- **Full function device (FFD)**
  - ✓ Any topology
  - ✓ Network coordinator capable
  - ✓ Talks to any other device

- **Reduced function device (RFD)**
  - ✓ Limited to star topology
  - ✓ Cannot become a network coordinator
  - ✓ Talks only to a network coordinator
  - ✓ Very simple implementation

802.15.4 Addresses

802.15.4 Addresses: EUI-64

- Each **802.15.4 node** is required to have a **unique 64-bit address**, called the **Extended Unique Identifier (EUI-64).**
  - ✓ In order to ensure **global uniqueness**,
    - **Device manufacturers** should acquire a **24-bit prefix,** the **organizationally unique identifier (OUI),** and
    - For **each device**, concatenate a **unique 40-bit extension identifier** to form the **complete EUI-64.**

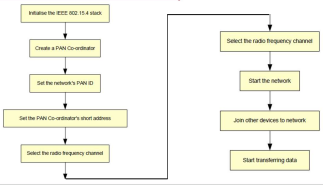
802.15.4 Addresses: EUI-64

- In the **OUI**,
  - **One bit (M) is reserved** to indicate the **nature of the EUI-64 address (unicast or multicast)**, and
  - **Another Bit (L) is reserved** to indicate whether the **address was assigned locally, or is a universal address**

16-Bit Short Addresses

- The **longer addresses increase** the **packet size**, therefore **require more transmission time** and **more energy**,
  - ✓ **Devices** can also request a **16-bit short address** from the **PAN controller**.

Setting up an IEEE 802.15.4 based Network.



The Network Topologies

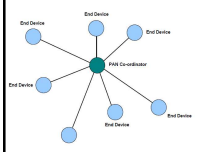
Network Topology

- The **IEEE 802.15.4** draft standard supports **multiple network topologies,** including both **Star and Peer-To-Peer Networks**.
- A **802.15.4 network** can be formed of **multiple clusters** (each having its **own PAN ID**) in a **tree configuration**
  - ✓ Each network, **identified by its PAN ID**, is called a **Cluster.**

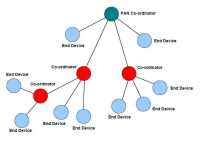
Network Topology

- The **topology is an application design choice;**
- Some applications, may require the **low-latency connection** of the **star network**,
- **Perimeter security**, **large-area coverage** requires **peer-to-peer networking.**
- **Multiple address types**, including both physical (i.e., **64-bit IEEE)** and **short (i.e., 8-bit network-assigned)** are provided.

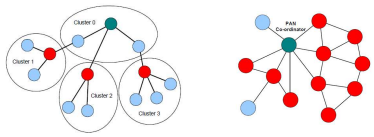
Star Topology



Tree Topology



Cluster Tree Topology Mesh Topology



802.15.4: Observation

- **802.15.4** is clearly targeted at **sensor and automation applications**.
- Both **ZigBee and 6LoWPAN** introduce **segmentation mechanisms** that **overcome the issue of small and hard to predict application payload sizes at the MAC layer**.

Thank U

ZigBee Technology

Dr. E.SURESH BABU
Assistant Professor
Computer Science and Engineering Department
National Institute of Technology, Warangal
Warangal

Session Outline

1. ZigBee Technology Standard
2. ZigBee Architecture/Protocol Stack
3. ZigBee Node Types
4. ZigBee Network Layer
5. ZigBee Address
6. ZigBee Communication Modes
7. ZigBee Routing Methods
8. The ZigBee APS Layer

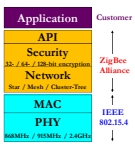
ZigBee Technology Standard

ZigBee Technology Standard

- **ZigBee** is a **technological standard** designed for **control and sensor networks**
- **ZigBee Technology Standard** uses
  - ✓ **802.15.4 Standard**
  - ✓ **ZigBee Alliance**

ZigBee Technology Standard

- **ZigBee Alliance**
  - ✓ **"The software"**
  - ✓ **Network, Security & Application layers**
  - ✓ **Brand management**
- **IEEE 802.15.4**
  - ✓ **"The hardware"**
  - ✓ **Physical & Media Access Control layers**



802.15.4 Standard

- The **802.15.4 standard** provides
  - ✓ **Physical** and **Link Layer Technology** of **low-rate WPAN,** which is optimized for **low bitrate, low duty cycle** applications.
- In practice,
  - ✓ **Sensor and Control Applications** needs a **mesh networking layer**.

ZigBee Alliance

- **ZigBee alliance** has developed
  - ✓ To build the **missing standard layers** that would be required to enable a **mesh network** on top of **802.15.4 radio links**.
  - ✓ Adds **network construction** (**Star Networks, Peer-to-peer/Mesh Networks, and Cluster-tree Networks**)

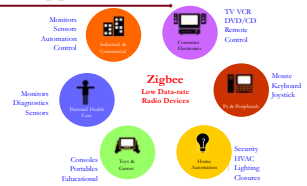
Zigbee Alliance

- Operates in **Personal Area Networks (PAN's)** and **device-to-device networks**
  - ✓ **Two-way Wireless Communications Standard.**
  - ✓ Connectivity between **small packet devices**
  - ✓ Control of **lights, switches, thermostats, appliances**, etc.

Characteristics of Zigbee

- **Low cost**
- **Low power consumption**
- **Low data rate**
- **Relatively short transmission range**
- **Scalability**
- **Reliability**
- **Flexible protocol design suitable for many applications**
- **Standards based**
- **Can be used globally**
- **Supports large number of nodes**
- **Easy to deploy**
- **Very long battery life**
- **Secure**

Zigbee Applications

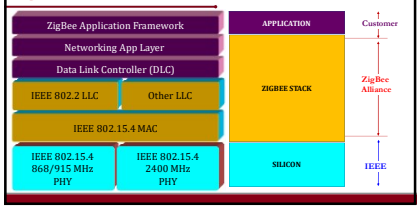


ZigBee Architecture/Protocol Stack

How is ZigBee related to IEEE 802.15.4?

- **ZigBee**
  - ✓ Takes full advantage of a **powerful physical radio** specified by **IEEE 802.15.4.**
  - ✓ Adds **logical network**, **security** and **application software.**
  - ✓ Works closely with the **IEEE** to ensure an **integrated** and **complete solution** for the market

Zigbee/IEEE 802.15.4 Architecture



Stack Reference Model

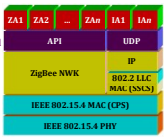
End developer applications, designed using application profiles

Application interface designed using general profile

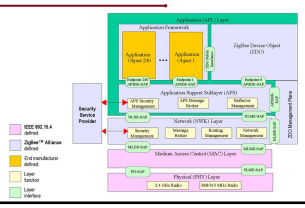
Topology management, MAC management, routing, discovery protocol, security management

Channel access, PAN maintenance, reliable data transport

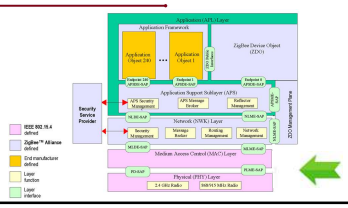
Transmission & reception on the physical radio channel



ZigBee Stack Architecture


Application/API Layer
Application Framework
Application Object 240
Application Object 1
ZigBee Device Object (ZDO)
Application Support Sublayer (APS)
Network (NWK) Layer
Medium Access Control (MAC) Layer
Physical (PHY) Layer
Security Service Provider
Security Management
Network Management
Service Discovery
Binding Management
Routing Management
Message Router

The Role of 802.15.4 in ZigBee



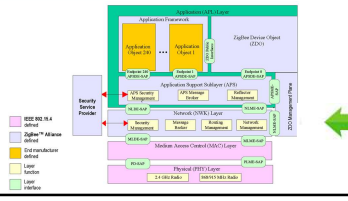
The Role of 802.15.4 in ZigBee

- **802.15.4** offers **16 channels** on the **2.4 GHz**, **numbered 11 to 26.**
- **ZigBee** uses only the **non-beacon-enabled mode** of **802.15.4**
  - ✓ All nodes use **CSMA/CA** to **access the network**,
  - ✓ **No option** to **reserve bandwidth** or to **access the network deterministically.**

The Role of 802.15.4 in ZigBee

- **ZigBee** restricts **PAN IDs** to the **0x0000 – 0x3FFF range**,
  - ✓ A subset of the **802.15.4 PAN ID range** (**0x0000-0xFFFE**).
- All **unicast ZigBee commands** request a **Hop By Hop Acknowledge**

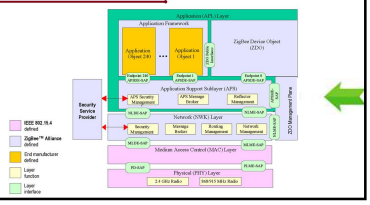
ZigBee Network Layer



ZigBee Network Layer

- The **ZigBee network layer** provides
  - ✓ The **functionality** of the **TCP/IP network layer**, adding the **missing mesh routing protocol** to **802.15.4.**
  - ✓ **Encapsulates** the **network formation primitives** of the **802.15.4 MAC layer**
    - **Network Forming** and **Joining**.

ZigBee Stack Architecture



Replacement

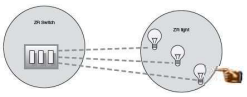
- **6LoWPAN-based networking layer** is likely to replace the **original networking layers of ZigBee**

Application Support Sublayer (APS) layer

- The **Application Support Sublayer (APS) layer** has several functions:
  - ✓ **Multiplexing/De-multiplexing**
  - ✓ **Binding**
  - ✓ **64-bit IEEE** to **16-bit ZigBee network node address mapping**.
  - ✓ **Management of end to end acknowledgements.**
  - ✓ **Fragmentation**
  - ✓ **Group addressing**
  - ✓ **Security**

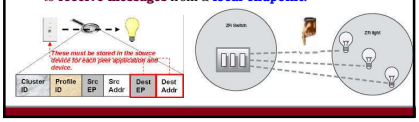
Multiplexing/De-multiplexing

- It forwards the **network layer messages** to the **appropriate application objects.**
- Each **application** is allocated an **endpoint ID**



Binding

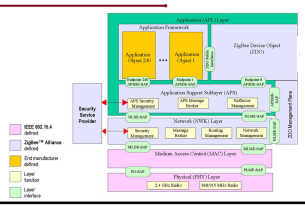
- The **APS layer** maintains the **local binding table**
  - ✓ **Records remote nodes** and **endpoints** which have **registered** to **receive messages** from a **local endpoint.**



Management of End to End Acknowledgements.

- The **application layer** supports **acknowledgements independently** of the **link layer acknowledgements** of **802.14.4.**
- The **APS** manages **retries** and **duplicate filtering** as required, simplifying **application programming.**

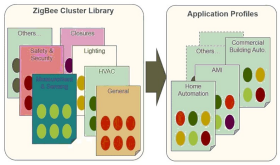
ZigBee Stack Architecture



ZigBee Device Object (ZDO) Layer

- This **ZDO layer** is a **specific application** running on **endpoint 0**
  - ✓ It is designed to **manage the state** of the **ZigBee node.**
  - ✓ The **ZDO application** implements the **interfaces** defined by the **ZigBee device profile** (**ZDP, application profile ID 0x0000**).
  - ✓ These **primitives** encapsulate the **802.15.4 network formation primitives** of the **ZigBee network layer**
    - **Node Discovery, Network Joining.**

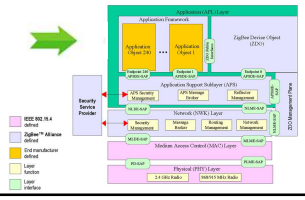
ZigBee Stack Architecture: Cluster



ZigBee Cluster Library (ZCL)

- **ZigBee Cluster Library (ZCL)**
  - ✓ **Late addition** to **ZigBee.**
  - ✓ It is now considered as **one of the key assets of ZigBee**
  - ✓ It consists in a **library of interface specifications** (**cluster commands and attributes**)
  - ✓ Used in **public and private application profiles.**

Application Framework Layer



Application Framework Layer

- It provides the **API environment** of **ZigBee application developers**,
- It is specific of **each ZigBee stack.**
- Each **application** is assigned an **Endpoint ID.**

ZigBee Node Types

ZigBee Node Types

- **Three kinds of devices** in the network layer
  1. **ZigBee End Device**
  2. **ZigBee Router**
  3. **ZigBee Coordinator**

ZigBee End-Device (ZED)

- This **node type** corresponds to the **802.15.4 reduced function device.**
  - ✓ It is a node with a **low duty cycle** (i.e. usually in a **sleep state** and **not permanently listening**), designed for **battery operation**.
  - ✓ It must **join a network** through **a router node**, which is their **parent.**

The Role of ZigBee End-Device (ZED)

- **ZigBee End-Devices** are responsible for:
  - ✓ **Finding and joining** the " correct " network
  - ✓ **Polling their parents** to see if any **messages were sent** to them **while they were asleep**
  - ✓ **Finding a new parent** if the link to the **old parent is lost (NWK rejoin)**
  - ✓ **Sleeping most of the time** to **conserve batteries** when **not in use by the application**

ZigBee Router (ZR)

- This **node type** corresponds to the **802.15.4 full function device (FFD).**
  - ✓ **ZigBee routers** are **permanently listening devices** that act as **packet routers**, once they have **joined an existing ZigBee network.**

The Role of ZigBee Router(ZR)

- **ZigBee Routers** are responsible for:
  - ✓ **Finding and joining** the " correct " network
  - ✓ Perpetuating **broadcasts across the network**
  - ✓ Participating in **routing, including discovering and maintaining routes**
  - ✓ Allowing **other devices to join the network** (if permit-join enabled)
  - ✓ **Storing packets on behalf of sleeping children**

ZigBee Coordinator (ZC)

- **ZigBee Coordinator** is responsible for **initializing, maintaining, and controlling the network**
  - ✓ It corresponds to an **802.15.4 full function device (FFD)**
  - ✓ **Capable to form a network** and can become an **802.15.4 PAN coordinator.**
  - ✓ It can form a **network**, or **join an existing network**

The Role of ZigBee Coordinator (ZC)

- It **forms a network.**
- It **establishes the 802.15.4 channel** on which the **network will operate.**
- It establishes the **extended and short PAN ID** for the network.
- It decides on the **stack profile to use** (compile or run-time option).
- It acts as the **Trust Centre for secure applications** and **networks.**
- It acts as a **router for mesh routing.**
- It acts as the **top of the tree, if tree routing is enabled.**

ZigBee Network Layer

ZigBee Network Layer

- **Network layer (NWK)**
  - ✓ Ensures the **proper functioning** of the **MAC layer**
  - ✓ Provides an **interface to the application level**
- **Network layer (NWK) Services**
  - ✓ Services for **Creating a PAN (ZigBee Coordinator)**
  - ✓ Services for **Device Association (ZigBee Router and End Devices)**
  - ✓ Forming **Network Topology**
  - ✓ Logical **Address Assignment And Routing (ZigBee Router)**

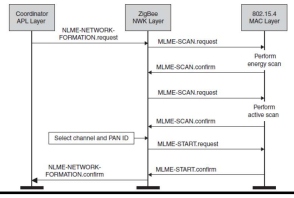
Network Creation

- The **Application layer** sets the **Coordinator**
  - ✓ **Self-forming network**
- **ZigBee Coordinators (ZCs)** form **networks.**
  - ✓ Determine a **unique identifier** for the **network,** called a **PAN ID.**
  - ✓ Choose **one of the sixteen 802.15.4 channels** (11–26)

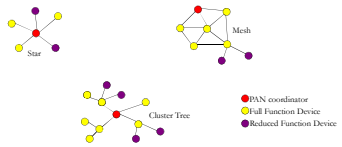
Network Creation: Forming a Network

- A **ZigBee coordinator**
  - ✓ First performs an **Active Scan** (**it sends beacon requests**) on **all channels**
  - ✓ It then **selects the channels** with the **fewer networks.**
  - ✓ Finally **broadcasts a 802.15.4 beacon** for the **selected PAN ID** on the **selected radio channel**.

Network Creation : Active Scan and Energy Scan



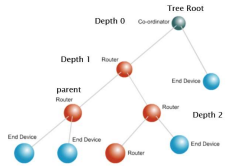
ZigBee Network Topologies



ZigBee Network Topologies

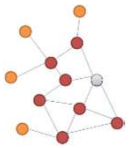
- **Star Topology**
  - **Advantage**
    - ✓ Easy to synchronize
    - ✓ Low latency
  - **Disadvantage**
    - ✓ Small scale

Star Topology

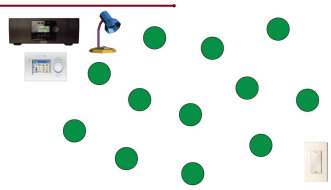


ZigBee Network Topologies

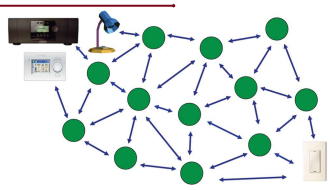
- **Mesh Topology**
  - **Advantage**
    - ✓ Robust multi-hop communication
    - ✓ Network is more flexible
    - ✓ Lower latency
  - **Disadvantage**
    - ✓ Route discovery is costly
    - ✓ Needs storage for routing table



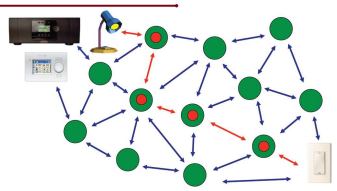
ZigBee Mesh Networking



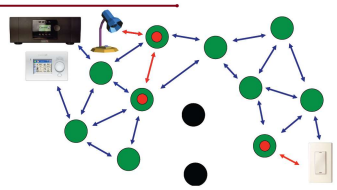
ZigBee Mesh Networking : Available Routes



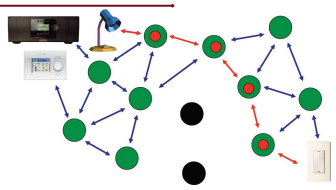
ZigBee Mesh Networking : Route Selection



ZigBee Mesh Networking : Power Loss



ZigBee Mesh Networking : Re-Route



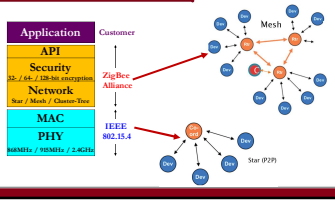
ZigBee Network Topologies

- **Cluster Tree**
  - **Advantage**
    - ✓ Low routing cost
    - ✓ Allow multi-hop communication
  - **Disadvantage**
    - ✓ Route reconstruction is costly
    - ✓ Latency may be quite long

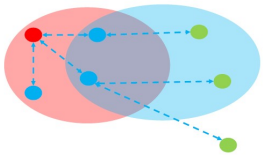
Tree Vs Mesh

- In a tree network, the **coordinator and routers** can announce **beacons.**
- In a mesh network, there is **no regular beacon**.
  - ✓ Devices in a **mesh network can only communicate** with each other in a **peer-to-peer manner**

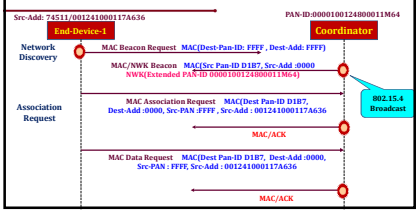
Which Layer are Involved?



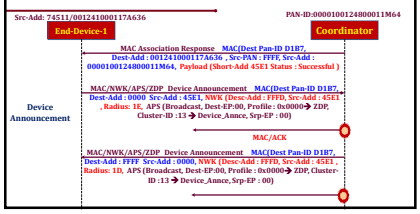
ZigBee Device Join Network Via Association



Joining a Parent Node in a Network Using 802.15.4 Association



Joining a Parent Node in a Network Using 802.15.4 Association



Data Frame Format.

- The **ZigBee network layer** uses a **specific data frame format.**

Octets: 2	2	2	1	1	Variable
Frame Control	Destination Address	Source Address	Radius[a]	Sequence Number[b]	Frame Payload
	Routing Fields				
NWK Header					NWK Payload

Session Outline

1. ZigBee Technology Standard
2. ZigBee Architecture/Protocol Stack
3. ZigBee Node Types
4. ZigBee Network Layer
5. **ZigBee Address**
6. **ZigBee Communication Modes**
7. **ZigBee Routing Methods**
8. **The ZigBee APS Layer**

ZigBee Address

Addressing ZigBee

- **PAN ID (MAC)**
- **Nwk-Addr (NWK)**
- **Endpoint (APS)**
- **Profile ID (APS)**
- **Cluster (APS)**
- **Command and/or attribute (ZCL)**

ZigBee PANs (1/2)

- **ZigBee PAN IDs** are **16-bit numbers** that range from **0x0000 to 0x3fff.**
  - ✓ Note this is **different than the 802.15.4 specification**, which allows **PAN IDs from 0x0000 to 0xfffe.**
- **One device only allowed** to **join one PAN** at a time.
  - ✓ When **PAN is configured as 0xffff**, the device will **join/create Radom PAN**

ZigBee PANs (2/2)

- **Extended PAN ID** is a **64-bit number** that help the **device to join the correct network.**
- **Extended PAN IDs** are **completely unrelated** to **PAN IDs,** and also to **MAC addresses.**
- They are a **unique 64-bit number**, used solely for **helping nodes find the right network.**

ZigBee Communication Modes

ZigBee Communication Modes: Packet Forwarding

- At the **network layer**, **ZigBee packets** can be:
  1. **Unicast**
  2. **Broadcast**
  3. **Multicast**
  4. **Many-to-one**

ZigBee Communication Modes: Unicast

- **Unicast:**
  - ✓ The **message** is sent to the **16-bit address** of the **destination node**
  - ✓ **ZigBee unicast packets** are always **acknowledged hop by hop.**

ZigBee Communication Modes: Broadcast (2/2)

- **Broadcast packets** are **not acknowledged**
  - ✓ Usually **retransmitted several times** by the **ZigBee stacks** of the **originating node** and by **ZigBee routers** on the path.
- **Broadcast messages** are sent to **group IDs** are not always **broadcast** at the **link layer level.**

ZigBee Communication Modes: Multicast

- **Multicast:**
  - ✓ The **message is sent** to a **16-bit group ID.**
  - ✓ A **group of devices** are the **destination** (e.g., **a number of lights** in the **room**)
  - ✓ **Any node** on the network can **initiate it** (**member or non-member**)
- **Many-to-one**
  - ✓ **Single sink node** receiving **all the traffic**

Broadcast, Groupcast, Multicast Models(1/2)

- At the **802.15.4 level**,
  - ✓ **Messages** sent to **multiple destinations** are always **broadcast.**

Broadcast, Groupcast, Multicast Models(1/2)

- At the **network layer,**
- **ZigBee** offers **more possibilities**, depending on the **destination address:**
  - ✓ **0xffff broadcast** to all **nodes** including **sleeping devices**;
  - ✓ **0xfffd broadcast** to all **awake devices**;
  - ✓ **0xfffc broadcast** only to **routers, not to end devices**.

ZigBee Routing Methods

Routing in ZigBee

- The **network layer** is required
  - ✓ **Multi-hop routing of data packets** in the **mesh network**
  - ✓ One of the **key missing elements** of **802.15.4.**
- **ZigBee** uses the **AODV public-domain mesh algorithm**.

Routing Methods

- **ZigBee employs** a **variety of methods** for **routing packets** from **one node to another:**
  - ✓ **Broadcasting** (from one to many nodes)
  - ✓ **Mesh Routing** (**unicast** from one node to another)
  - ✓ **Tree Routing** (**unicast** from one node to another, stack profile 0x01 only)
  - ✓ **Source Routing** (**unicast** from one node to another, stack profile 0x02 only)

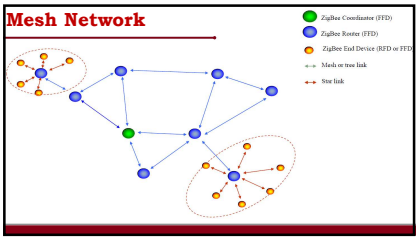
ZigBee Routing Methods

	Broadcast	Mesh	Tree	Source Route
Multi-hop	Up to 30	Up to 30	Up to 10	Up to 5 hops
Multiple destinations	Yes	No	No	No
One-to-one	No	Yes	Yes	Yes
Bandwidth efficient	No	Yes	Yes	Yes
Payload efficient	Yes	Yes	Yes	No
Acknowledged	No	Yes	Yes	Yes

Mesh Routing

- **Mesh Routing** is the **default routing model** of **ZigBee.**
- **ZigBee** uses the advanced **Ad-hoc On-demand Distance Vectoring (AODV) Algorithm,** which is a **public-domain mesh algorithm**.

Mesh Network



AODV (Ad hoc On-Demand Distance Vector) Routing Protocol

Ad-hoc On-demand Distance Vector (AODV)

- **A**d-hoc **O**n-demand **D**istance **V**ector(AODV) is a Reactive routing protocol
  - ✓ **Charles E. Perkins, 1999**
- **Routes are discovered** only on a need-to basis
- The **route is distributed**: Each node in the **route keeps track** of the **next hop** for the **route in a routing table.**

On-Demand Routes

- **In AODV,**
  - ✓ A **route is established** only when it is **required by a source node** for transmitting data packets.
  - ✓ The **source node** and **intermediate nodes** store the **next-hop information**
  - ✓ **Routes are unidirectional** (a route must be discovered each way for bidirectional communication
    - **Data packet transmission for each flow.**

Routing Tables of AODV

- **AODV** utilizes **routing tables** to **store routing information**. The routing table contains
  - ✓ **Destination IP addresses,**
  - ✓ **Next-hop IP addresses** and
  - ✓ The **destination sequence number**

Destination	Next Hop	IsValid
0x000a	0x0000	Yes
0x0009	----	Pending
----	----	No

destination addr	next-hop addr	destination sequence

The AODV Routing Procedure

- If a node wants to **send a packet** to some destination.
  - ✓ At first, it checks its **routing table** to determine whether it has a **current route** to the **destination or not.**
    - If **Yes**, it forwards the **packet to next hop node** of the route.
    - If **No**, it initiates a **ROUTE DISCOVERY PROCESS.**

AODV Uses Control Messages

- AODV requires **two types** of Control Messages
  - ✓ **Route Discovery**
    - **Route Request**
    - **Route Reply**
  - ✓ **Route Maintenance**
    - **Failed routes are communicated back** to the **originating node,** allowing it to **discover a new route**

Route Discovery Process (RREQ) (1/3)

- **Route Discovery Process** begins with the creation of a **Route Request (RREQ)packet. Broadcasting** is done via **Flooding.**
  - ✓ It broadcasts a **route request** to **network address 0xfffc** (**routers only**), which **propagates** through the **network**.

Route Discovery Process (RREQ) (3/3)

- When a **Intermediate Node** re-broadcasts a **Route Request**, it sets up a **REVERSE PATH POINTING** towards the source
  - ✓ **ZigBee mesh AODV routing** assumes **symmetrical link quality**

Route Discovery Process (RREP) (1/4)

- Once the **intended destination** receives a **Route Request Packet,** then it replies by sending a **Route Reply.**
- **Intended Destination** will wait for a **while until**
  - ✓ It **receives all broadcast route requests**,
  - ✓ **Computes the lowest path cost** and
  - ✓ **Responds by a unicast route reply** along the **best route that was just discovered**.

Route Discovery Process (RREP) (2/2)

- **Route Reply** travels along the **reverse path set-up** when **Route Request is forwarded**

Role of Intermediate Node in RD Process

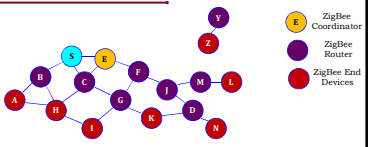
- An **intermediate node** (**not the destination**) may also send a **Route Reply (RREP) packet** provided
  - ✓ it knows a **more recent path** than the one **previously known** to sender S.

Illustration of Route Establishment in AODV

- **Node S** needs a **routing path** to **node D.**
- **Node S** creates a RREQ packet

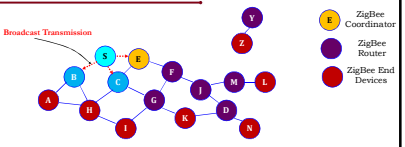
  RREQ {**D's IP Addr, D Seq#, S's IP Addr, Seq#, Link-Cost, HopCcount**}
- Node S broadcasts **RREQ to its neighbors.**

Route Requests in AODV: Example



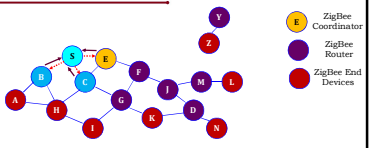
S Represents a node that has received **RREQ for D from S**

Route Requests in AODV: Example



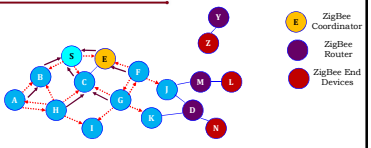
Represents transmission of RREQ

Route Requests in AODV: Example



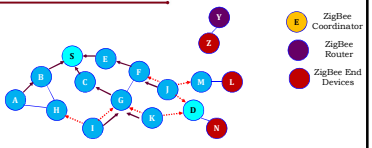
Represents links on Reverse Path

Route Requests in AODV: Example

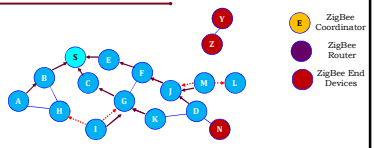


- **Node C** receives **RREQ** from **G and H,** but does not forward it again, because node C has **already forwarded RREQ** once

Route Requests in AODV: Example

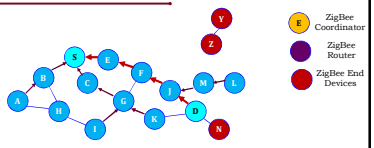


Route Requests in AODV: Example

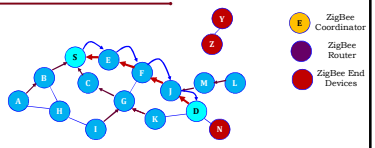


❖ Node **D does not forward RREQ**, because node D is the **intended target** of the RREQ

Route Requests in AODV: Example



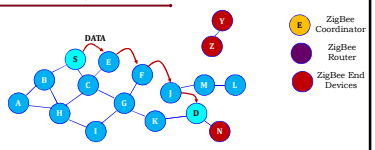
Forward Path Setup in AODV



Forward links are setup when **RREP travels** along the reverse path

Represents a **link on the forward path**

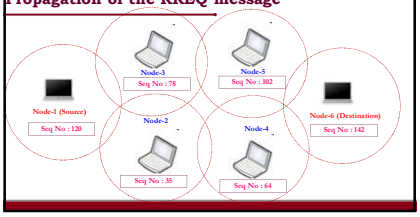
Data Delivery in AODV



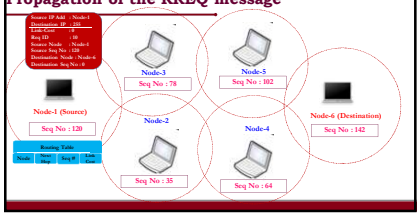
❖ **Routing table entries** are used to forward data packet.

❖ Route is *not* included in packet header.

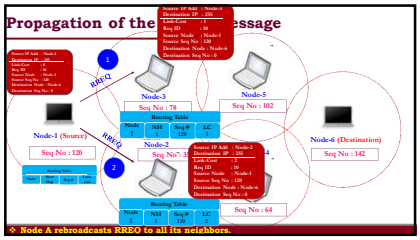
Propagation of the RREQ message



Propagation of the RREQ message

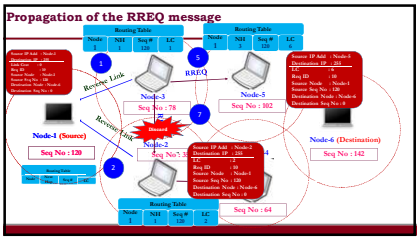


Propagation of the RREQ message

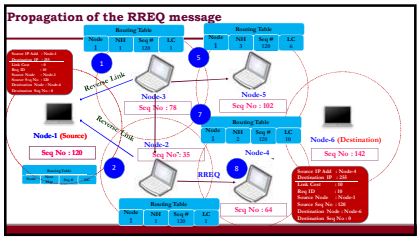


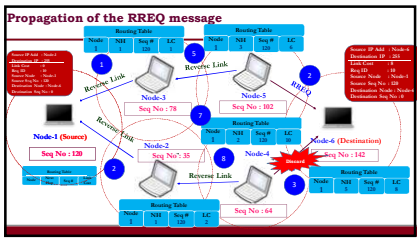
❖ **Node A rebroadcasts RREQ to all its neighbors.**

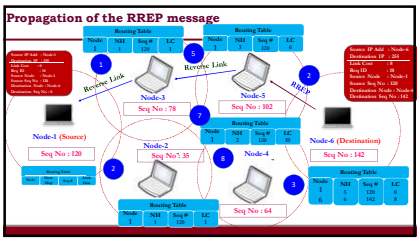
Propagation of the RREQ message

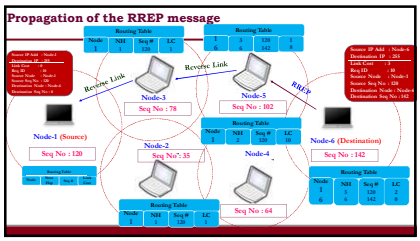


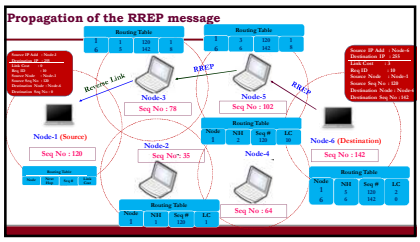
Propagation of the RREQ message

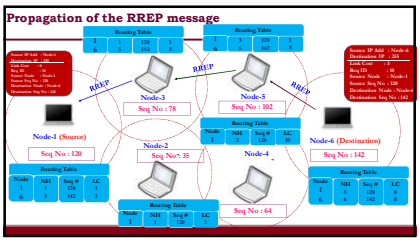


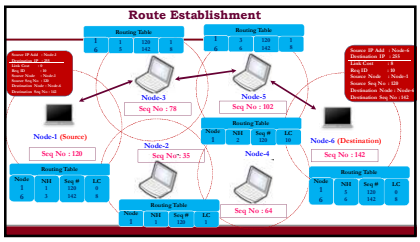










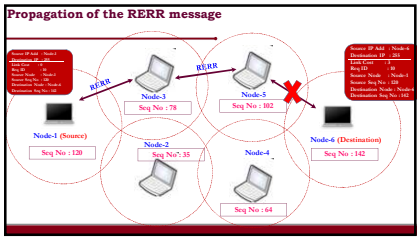


Route maintenance in AODV (Path broken due to host mobility)

- If **intermediate nodes** or the **destination move.**
  - ✓ The next hop **links break.**
  - ✓ **Routing tables** are **updated** for the **link failures.**
  - ✓ All **active neighbors** are informed by **Route Error (RRER)packet**.

Route maintenance in AODV (Path broken due to host mobility)

- When a **source node** receives an **RERR**,
  - ✓ **RERR message** is now propagated towards the **appropriate source.**
- **Source node re-initiates** the **route discovery process.**
  - ✓ It can be also dealt with by a **Local Fix Scheme**
  - ✓ **Modify the routing table! Self-healing Feature**



The ZigBee APS Layer

The ZigBee APS Layer

- **The ZigBee APS Layer** will interfaces the **application layer (AP) with the network layer**
- **APS layer** responsible for **management** and **support** of **local applications.**
- **APS layer** defines all the **concepts** to **develop and interconnect ZigBee applications**:
  - ✓ **Endpoints**
  - ✓ **Groups**
  - ✓ **Bindings.**

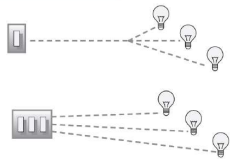
Endpoints

- **Endpoints** serve **three purposes** in ZigBee:
  - ✓ **Endpoints** allow for **different application profiles** to exist within **each node**
  - ✓ **Endpoints** allow for **separate control points** to **exist within each node**
  - ✓ **Endpoints** allow for **separate devices** to **exist within each node**

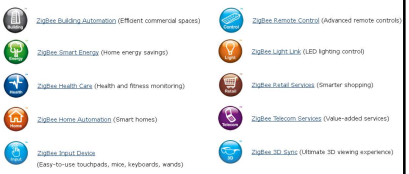
Simple Descriptor

- Each **endpoint** is further characterized by a **simple descriptor.**
- The **simple descriptor** contains
  - ✓ **The endpoint number,**
  - ✓ **Application profile ID,**
  - ✓ **Application device ID,**
  - ✓ **An application version ID,**
  - ✓ **The list of input clusters and the list of output clusters.**

Endpoints



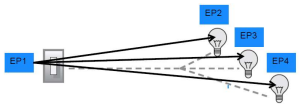
Profile ID



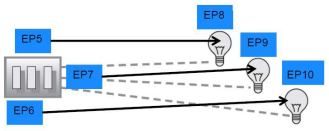
Binding

- **Bindings** are one of the **publish/subscribe models** in **ZigBee specification.**
- **Binding** provides a **mechanism** for **attaching an endpoint on one node to one or more endpoints** on **another node.**
- **Binding** can even be **destined for groups of nodes.**

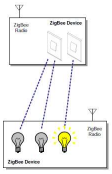
Binding Connects One Endpoint to One or More Endpoints



Binding Connects One Endpoint to One or More Endpoints



Sample Addressing at the Application Layer



Home Control Profile

- Light Control (On/Off)
- Dimmer
- Motion Detection

Legend

ZigBee and Bluetooth Comparison

ZigBee and Bluetooth Comparison

- Optimized for Different Applications

1. **ZigBee**
   - ✓ Smaller packets over large network
   - ✓ Mostly Static networks with many, infrequently used devices
   - ✓ Home automation, toys, remote controls, etc.
2. **Bluetooth**
   - ✓ Larger packets over small network
   - ✓ Ad-hoc networks
   - ✓ File transfer
   - ✓ Screen graphics, pictures, handsfree audio, Mobile phones, headsets, PDAs, etc.

ZigBee and Bluetooth Comparison

Feature(s)	Bluetooth	ZigBee
Power Profile	days	years
Complexity	complex	Simple
Nodes/Master	7	64000
Latency	10 seconds	30 ms – 1s
Range	10m	70m ~ 300m
Extendibility	no	Yes
Data Rate	1 Mbps	250 Kbps
Security	64bit, 128bit	128bit AES and Application Layer

Thank U

Network Adaptation Layer : 6LoWPAN

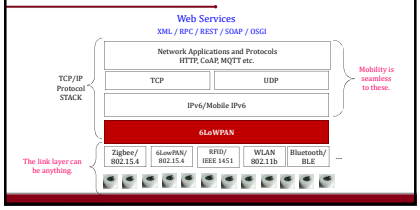
Dr. E.SURESH BABU
Assistant Professor
Computer Science and Engineering Department
National Institute of Technology, Warangal
Warangal

Session Outline

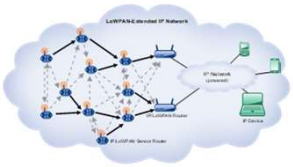
1. 6LoWPAN Motivation
2. Applications of 6LoWPAN
3. Challenges with IoT Devices
4. Need of 6LoWPAN
5. The 6LoWPAN Architecture
6. The 6LoWPAN Protocol Stack
7. 6LoWPAN Header Format
8. 6LoWPAN Header Compression

6LoWPAN Motivation

IoT Protocol Stack.



Low Power Wireless Internet



Source : IPSO

Why 6LoWPAN(1/2)

- **6LoWPAN** is the key protocol in realizing the **Wireless Embedded Internet**.
- The **Wireless Internet** is a subset of the **Internet of Things**
  - ✓ Often **battery powered**,
  - ✓ Connected by **low-power, low-bandwidth wireless networks to the Internet.**

Why 6LoWPAN(2/2)

- **6LoWPAN** was developed to enable the **Wireless Internet** by
  - ✓ **Simplifying IPv6 Functionality**,
  - ✓ Defining **very compact header formats** and
  - ✓ **Taking the nature of wireless networks** into account.

Applications of 6LoWPAN

Various Applications requires Wireless Internet

Challenges with IoT Devices

Need of Internet Protocol

- The **benefits of using Internet protocols** in these applications, and thus **integrating them with the Internet of Things**
  - ✓ **IP-based devices** can be connected easily to **other IP networks without the need** for **translation gateways or proxies**.
  - ✓ **IP networks** allow the use of **existing network infrastructure**.

Challenges of IoT Devices

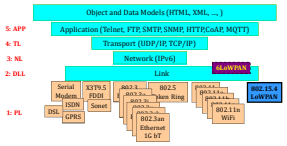
- **Traditional Internet protocols** are possess the challenges for **IoT devices** for the **following reasons**:
  - ✓ **Security**
  - ✓ **Web services**
  - ✓ **Frame size**
  - ✓ **Power and Duty-Cycle**
  - ✓ **Multicast**
  - ✓ **Mesh topologies**
  - ✓ **Bandwidth and Frame Size**
  - ✓ **Reliability**

Need of 6LoWPAN

Need of 6LoWPAN

- **6LoWPAN** was created to **tackle below problems**, and to **specifically enable IPv6** to be used with **IoT devices and networks.**
  - ✓ **Security**
  - ✓ **Web services**
  - ✓ **Frame size**
  - ✓ **Power and Duty-Cycle**
  - ✓ **Multicast**
  - ✓ **Mesh topologies**
  - ✓ **Bandwidth and Frame Size**
  - ✓ **Reliability**

6LoWPAN is an Adaptation Layer



What is 6LoWPAN?

- IPv**6** over **Lo**w power **W**ireless **P**ersonal Area **N**etworks (**6LoWPAN**)
  - ✓ An **adaptation layer for IPv6** over **IEEE 802.15.4 links**
- **6LoWPAN standards** enable
  - ✓ The **efficient use of IPv6 over low power, low-rate wireless networks** on simple **IoT devices** through an **adaptation layer** and the **optimization of related protocols.**

What is 6LoWPAN?

- **Features of the IPv6 design** such as a **simple header structure**, and **its hierarchical addressing model**,
  - ✓ it made **ideal for use in IoT networks** with **6LoWPAN.**
- **LoWPAN** is composed of **IoT devices** conforming to the **IEEE 802.15.4-2003** standard **defined by the IEEE.**

Requirement of 6LoWPAN

- The **minimum requirements** for implementing **a lightweight IPv6 stack with 6LoWPAN** could be aligned with the **most minimal devices.**
- **6LoWPAN** provides **efficient extension of IPv6** into the **IoT domain**,
  - ✓ **Enabling end-to-end IP networking** and **features for a wide range of IoT applications**.

The 6LoWPAN Architecture

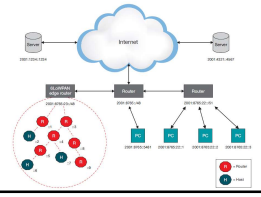
The 6LoWPAN Architecture

- The **Wireless Embedded Internet** is created by **connecting islands of wireless embedded devices**,
  - ✓ Each island being a **stub network** on the Internet.
  - ✓ A **stub network** is a network, which **IP packets are sent** from or **destined to**, but which **doesn't act as a transit to other networks.**

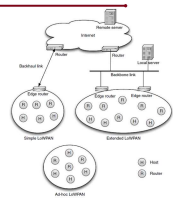
The 6LoWPAN Architecture

- The **6LoWPAN architecture** is made up of **low-power wireless area networks (LoWPANs)**, which are **IPv6 stub networks.**
- The **overall 6LoWPAN architecture** have been defined with **Three different kinds** of **LoWPANs**
  1. **Simple LoWPANs,**
  2. **Extended LoWPANs, and**
  3. **Ad hoc LoWPANs.**

Network Architecture of 6LoWPAN



6LoWPAN Architecture



Edge Router(1/2)

- The **Edge Router** plays an important role as
  - ✓ It **routes traffic** in and out of the **LoWPAN**
  - ✓ While handling **6LoWPAN compression** and **Neighbour Discovery** for the LoWPAN.
  - ✓ If the **LoWPAN** is to be connected to an **IPv4 network**, the **edge router** will also **handle IPv4 interconnectivity**.

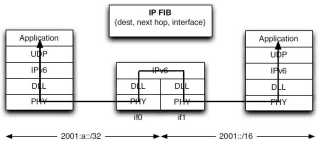
Network Operation

- In order to facilitate **efficient network operation**, **nodes register** with an **edge router**.
  - ✓ These **operations** are part of **Neighbour Discovery (ND)**, which is an **important basic mechanism of IPv6.**
  - ✓ **Neighbour Discovery** defines how **hosts and routers** interact with **each other** on the **same link.**

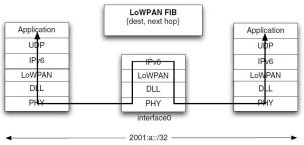
LoWPAN Uses Mesh Topology

- A **Multi-hop Mesh Topology** within the **LoWPAN** is achieved using
  1. **Through link-layer forwarding (called Mesh-Under)**
  2. **Through using IP routing (called Route-Over).**
- Both techniques are supported by **6LoWPAN.**

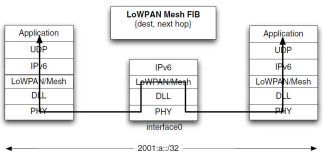
The IP Routing Model



The LoWPAN Routing Model (L3 Routing, "Route-Over").



(Mesh Under) LoWPAN Mesh Forwarding.

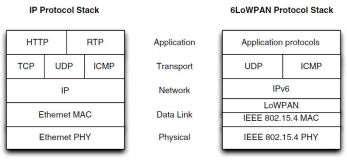


Adhoc LoWPAN

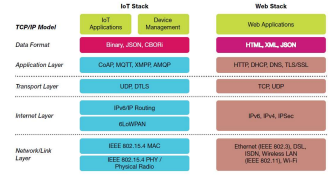
- **6LoWPAN** does not require an **infrastructure to operate**, but may also **operate as an Ad hoc LoWPAN [ID-6lowpan-nd].**
- In this topology, **one router** must be configured to act as a **simplified edge router**, implementing **two basic functionalities**:
  - ✓ **Unique local unicast address (ULA) generation**
  - ✓ **Handling 6LoWPAN Neighbour Discovery registration functionality**.

The 6LoWPAN Protocol Stack

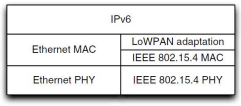
IP and 6LoWPAN Protocol Stacks.



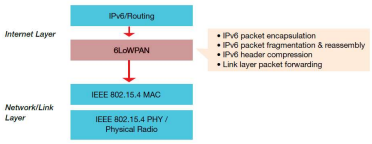
Recap: IoT Stack



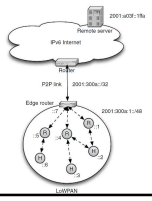
IPv6 Edge Router with 6LoWPAN Support.



IPv6 Sent into 802.15.4 MAC by 6LoWPAN



6LoWPAN Network Example

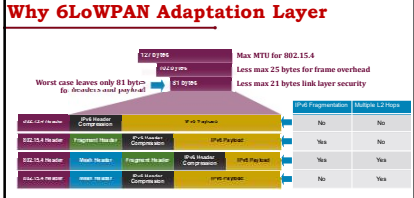


THE 6LoWPAN Adaptation Layer

THE 6LoWPAN Adaptation Layer

- **IPv6** mandates **supporting links** with an **MTU (Maximum Transmission Unit)** of **1280 bytes**.
- it was necessary for **IEEE 802.15.4 links** that have an **MTU of 127 bytes** to **specify an adaptation layer below IP** responsible for handling **packet fragmentation and reassembly**.

Why 6LoWPAN Adaptation Layer



Source : Cisco

MTU Requirement of IEEE 802.15.4 frame

- The **IEEE 802.15.4 frame MTU** is **127 bytes** minus a **set of protocol fields:**
- **Maximum MAC frame Overhead = 25 Bytes**
  - ✓ **Frame control (2 bytes)** +
  - ✓ **Sequence number (1 byte)** +
  - ✓ **Addressing field (up to 20 bytes** with the source and destination PAN ID and the source and destination 64-bit extended addresses) +
  - ✓ **FCS (2 bytes)**

MTU Requirement of IEEE 802.15.4 frame

- The **IEEE 802.15.4 frame MTU** is **127 bytes** minus a **set of protocol fields:**
- **MAC security header: 21 Bytes**
  - ✓ **21 bytes (AES-CCM-128)**,
  - ✓ **13 bytes (AES-CCM-64)**, and
  - ✓ **9 bytes (AESCCM-32).**

MTU Requirement of IEEE 802.15.4 frame

- After removing the **size of the IPv6 header (40 bytes),** there are **41 bytes left for (Transport Layer)**
- Next, we must **deduct the transport layer protocol header (8 bytes for UDP** and **20 bytes for TCP)**, thus leading to a **very short payload** for the **application layer.**

Summary : Header Size Calculation

Header Size Calculation. . .

- IPv6 header is 40 octets, UDP header is 8 octets
- 802.15.4 MAC header can be up to 25 octets (null security) or 25+21=46 octets (AES-CCM-128)
- With the 802.15.4 frame size of 127 octets, we have
  - 127-25-40-8 = 54 octets (null security)
  - 127-46-40-8 = 33 octets (AES-CCM-128)

of space left for application data!

6LoWPAN Motivation

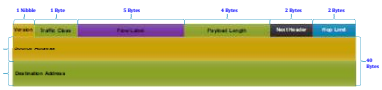
- **6LoWPAN** is an **adaptation layer,** which is needed to **comply with the IPv6 requirement**
  - ✓ To support a **minimum MTU size of 1280 bytes**
  - ✓ To support **compression techniques** to **reduce protocol overhead.**
    - The **use of efficient compression techniques** allows **most applications** to **send their data** within a **single IPv6 packet.**

Services of 6LoWPAN

- **6LoWPAN** provides
  1. **Packet Fragmentation and Reassembly**
  2. **Header Compression**
  3. **Link Layer (layer 2) forwarding when multi-hop is used by the link layer**
     - Where the **nodes** were **exclusively interconnected** by **IEEE 802.15.4 links.**

6LoWPAN Header Format

Basic IPv6 Header (Recap)

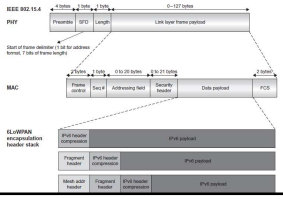


- Minimum size is 40 bytes (double that of IPv4)
- Can be extended by additional headers
- Fragmentation must be performed by end nodes

6LoWPAN Encapsulation Header Stack

- The **6LoWPAN adaptation layer** defines what is called the "**encapsulation header stack,**" which precedes **each IPv6 datagram.**

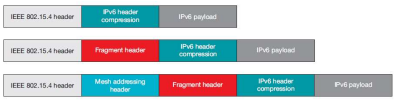
6LoWPAN Encapsulation Header Stack



Types of Headers in 6LoWPAN

- The **6LoWPAN adaptation** currently supports **three headers**:
  1. **A Mesh Addressing Header,**
  2. **The Fragment Header, and**
  3. **The IPv6 Header Compression Header** (they must **appear in that order** when present).

6LoWPAN Stacked Headers



6LoWPAN Encapsulation Header

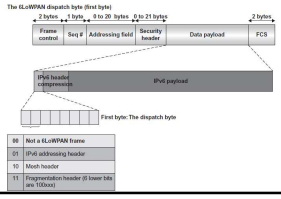
- **6LoWPAN encapsulation format** is mainly used to **provide a packet type identifier** not provided by **IEEE 802.15.4 itself.**
- **6LoWPAN** uses the **first byte of the payload** is used as a **DISPATCH BYTE** to **identify** the **6LoWPAN packet** that provides both a **type identifier and possibly further information** within the subtype

Two Most-Significant Bits in the Dispatch Byte

The two most-significant bits in the dispatch byte.

00	Not a LoWPAN packet (NALP)
01	Normal dispatch
10	Mesh header,
11	Fragmentation header,

Dispatch Byte : IPv6 Compression Header.



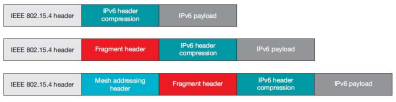
Dispatch Byte : IPv6 Compression Header.

- The **first byte** of the **encapsulation header identifies** the **next header**.
  - ✓ For example, if the **first 2 bits** are **equal to 11**, the **next header** is a **fragmentation header**.
  - ✓ If the **first 8 bits** are equal to **01000001**, what follows is an **IPv6 uncompressed packet.**
  - ✓ In contrast, **a value of 01000010** indicates that what follows is a **header** related to a **compressed header** using **HC1 compression**

Session Outline

1. 6LoWPAN Header Format
2. 6LoWPAN : The Mesh Addressing Header
3. 6LoWPAN : Fragmentation using Fragment Header
4. 6LoWPAN Header Compression
5. The HC1 Compression Technique
6. The HC2 Compression Technique(HC_UDP)
7. Improved 6LoWPAN Compression
8. Improved 6LoWPAN : The IPv6 Next Header Compression

6LoWPAN Stacked Headers(Recap)



6LoWPAN : The Mesh Addressing Header

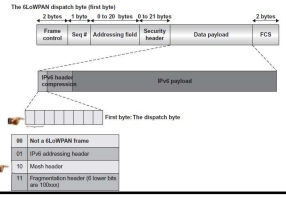
The Mesh Addressing Header

- The **mesh addressing header** is used in **conjunction** with a **mesh-under "routing" approach**
  - ✓ Where **nodes** that are **not in direct communication** make use of **multi-hop "routing"** at the **link layer** using **link layer addresses.**

The Mesh Addressing Header

- According to **IEEE 802.15.4**,
  - ✓ Only **full function devices (FFDs)** perform **mesh-under operation**.
  - ✓ **Reduced function devices (RFDs)** systematically **send all of their traffic to FFDs.**
  - ✓ The **source and destination nodes** are then referred to as the **originator** and **final destination,** respectively.

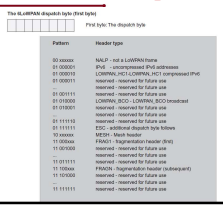
Dispatch Byte : The Mesh Addressing Header



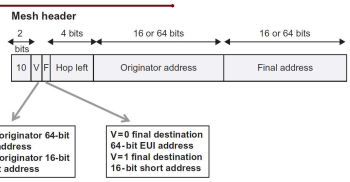
Dispatch Byte : The Mesh Addressing Header

- As shown in Figure
  - ✓ The **first 2 bits** of the **dispatch byte** identify the **presence of a mesh header** and are **equal to 10.**

Value of the 6LoWPAN Dispatch Byte.



6LoWPAN : The Mesh Header

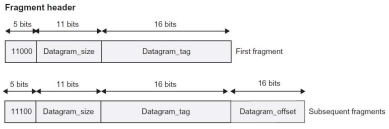


6LoWPAN : Fragmentation using Fragment Header

Fragmentation

- **Fragmentation** may be required at the **6LoWPAN adaptation layer**
  - ✓ When the **IPv6 payload** cannot be carried within a **single IEEE 802.15.4 frame** because it **exceeds the MTU size**.
  - ✓ In this case, **the link frame** is broken into **multiple link fragments** using the **fragment header** shown in Figure.

Fragment Header



Description of the Datagram-Size:

- **Datagram-Size:**

  - ✓ This **11-bit field** is used to indicate the **size in 8-byte units** of the **original IPv6 packet.**
  - ✓ **Link layer fragmentation** supports a **1280-byte packet** as mandated by the **IPv6 specification**
  - ✓ The **Datagram-Size** may only be needed in the **first link fragment** and then **elided in other link fragments.**

Limitation.

- The **drawback of this approach** is that **subsequent link fragments** (other than the first link fragment) **may arrive first,** especially in the **presence of multi-hop routing**.
- In this case the **receiver would not know** how much **memory should be allocated for the entire frame.**

Description of the Datagram-Tag

- **Datagram-Tag:**

  - ✓ This **field** is used in conjunction with the **IEEE 802.15.4 source address** (or **originator address** if a **mesh header** is present),
  - ✓ The **IEEE 802.15.4 destination address** (or the **final destination address** if a mesh header is present),

Description of the Datagram-Tag

- **Datagram-Tag:**

  - ✓ The **datagram_size** to **uniquely identify** the **fragmented frame** and must **be identical for all link fragments**.
  - ✓ It is recommended to **increment the datagram_tag** for **successive fragmented frames.**

Description of the Datagram-Offset

- The **8-bit datagram-offset field** is present in **all link fragments** except the **first fragment**.
- It indicates the **offset in 8-byte units** from the **beginning of the payload datagram.**

Description of the Datagram-Offset

- **Final Destination** uses of a **reassembly timer** that is started
  - ✓ When **receiving the first link fragment** and upon the **expiration of which**,
  - ✓ if **not all link fragments have been received, all fragments must be discarded.**
  - ✓ The **maximum value of the reassembly timer** is **60 seconds.**

6LoWPAN Header Compression

6LoWPAN Header Compression

- The **general idea** of the **6LoWPAN header compression** is
  - ✓ To **derive the IP address** from **link layer addresses**
  - ✓ To **avoid needless information duplication**
  - ✓ To **suppress IPv6 headers** that **have common values.**

Shared Contexts

- The **use of shared contexts** such as
  - ✓ The **use of a common network prefix** for the **LoWPAN** allows **address compression** of **IPv6 global addresses**.

Types of Header Compression

- **6LoWPAN Header Compression** is **flow independent** that can be **Stateless or Stateful Compression**
  1. **The HC1 Compression Technique**
  2. **The HC_UDP Compression Technique (HC2 Byte)**

Need of Header Compression

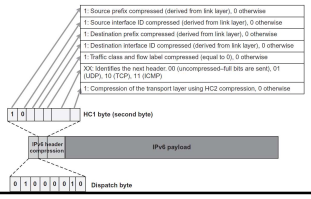
- Several of the **IPv6 headers** have **common values** and are **easily compressed**;
  - ✓ For example, **the IP version (v6), the flow label, TC etc**.
- **Information such as the IPv6 interface ID** can be **derived** from the **link layer frame** when using **extended 64-bit 802.15.4 addresses**.

The HC1 Compression Technique

The HC1 Compression Technique

- The **HC1 compression technique** relies on the following observations: These observations allow a considerable **reduction of the protocol overhead.**
  - ✓ **IP version** is always **6.**
  - ✓ **HC1 is optimized** for **link-local addresses**,
  - ✓ the **IPv6 interface ID** (bottom **64 bits of the IPv6 address**) can be inferred from the **link layer MAC address.**

6LoWPAN HC1 Byte.

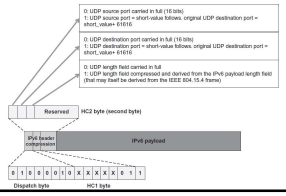


6LoWPAN HC1 Byte.

- This leads to only **3 bytes** instead of the **40-byte IPv6 header:**
  - ✓ **1 byte** for the **dispatch byte** (equal to **01000010**),
  - ✓ **1-byte HC1 byte**
  - ✓ **1 byte** for the **hop limit field**,
- When **set, bit 7** of the **HC1 byte** allows for the **compression of the next header** of the **original IPv6 header.**

The HC2 Compression Technique(HC_UDP)

The HC2 Compression Technique(HC_UDP)



The HC2 Compression Technique(HC_UDP)

- When **bit 7** of the **HC1 byte** is **set,**
  - ✓ It indicates **more header compression** according to the **HC2 encoding format.**
  - ✓ If **bits 5 and 6** of the **HC1 byte** are **equal to 0 and 1,** respectively,
    - This indicates the **compression of the UDP header** (called **HC_UDP encoding**).

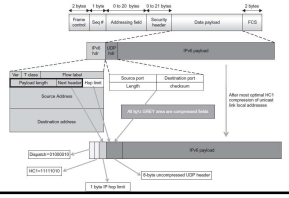
The HC2 Compression Technique(HC_UDP)

- **The HC2 byte** immediately follows **the HC1 byte** (thus before the **IP hop limit field**)
  - ✓ Provides **information on the UDP header compression scheme.**
- The **HC_UDP compression technique** allows **compression of the UDP header** to **various degrees**.

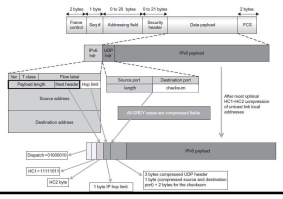
Compression of UDP Port

- The main idea for **compressing the source and destination UDP port** uses a **short-value 4-bit field** instead of the **original 16-bit field.**
  - ✓ The **original 16-bit field** is simply obtained by the **formula short-value : 61616 (0xF0B0).**

Compression of the IPv6 HC1 Link-local IPv6 Addresses without UDP Header Compression.



Maximized Compression of IPv6 and UDP Headers using HC1 and HC2 with link-local IPv6 addresses.



Different Examples of Header Compression.

- Figure shows **HC1+ HC2 header compression** allows a **very efficient compression technique** for **reducing**
  - ✓ **The header size** from **40 bytes (IPv6 header)** + **8 bytes (UDP header) down** to **7 bytes.**

Session Outline

1. 6LoWPAN Header Format
2. 6LoWPAN : The Mesh Addressing Header
3. 6LoWPAN : Fragmentation using Fragment Header
4. 6LoWPAN Header Compression
5. The HC1 Compression Technique ✓
6. The HC2 Compression Technique(HC_UDP) ✓
7. **Improved 6LoWPAN Compression** ✓
8. **Improved 6LoWPAN : The IPv6 Next Header Compression**

Improved 6LoWPAN Compression : Stateful Shared Context-based Compression

HC1 and HC2 Compression Techniques

- The **HC1 and HC2 compression techniques** are **quite efficient** for **unicast link-local addresses**
  - ✓ Useful in many circumstances such as **ND, DHCP, and other local protocols**),
  - ✓ But have a **very limited effect on global** and **multicast addresses**.

Compression Techniques: Global IPv6 Addresses

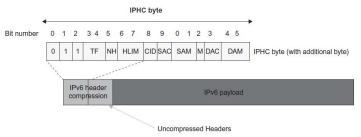
- **Efficient compression techniques** for **global IPv6 addresses** are **needed**
  - ✓ **Communication between nodes** residing in **different IP subnetworks** (Nodes in **different PANs** in the **context of IEEE 802.15.4 networks**),
  - ✓ Where **routable addresses** are required.

6LoWPAN/UDP Compressed Headers

- **Two Compression Techniques** were used by the **6LoWPAN adaptation layer**
  1. **LOWPAN_IPHC (IPHC in Short)**
  2. **LOWPAN_NHC (NHC in Short).**

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   Dispatch and LOWPAN_IPHC    |   LOWPAN_NHC  |  Src  |  Dst  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           UDP Checksum        | ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

IPHC Encoding: Traffic And Flow Label Compression.

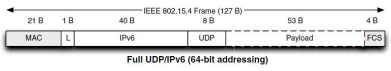


The Context Identifier (CID)

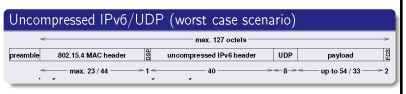
- **Byte** is used to **identify the context** and used to **identify the source and destination**
  - ✓ The **first 4 bits** called the **source context identifier** or **SCI** is used for the **source address** and
  - ✓ The **following 4 bits** called the **destination context identifier** or **DCI** is used for the **destination address**).
  - ✓ Since **4 bits** are used to **identify the context**, up to **16 contexts are allowed.**

Very Best Case: Header Compression

Full UDP/IPv6 (64-bit & 16-bit Addressing)



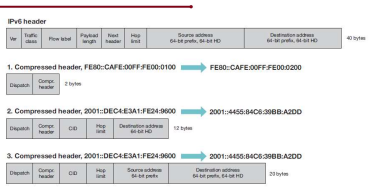
Uncompressed IPv6/UDP(Worst Case Scenario)



Very Best Case: IP Header Compression

- With **Link-Local Unicast Address**,
  - ✓ **HC1 encoding** allows **compression of the IPv6 header** to **TWO OCTETS (The dispatch byte + The HC1 byte).**

6LoWPAN IPv6 Header Compression Examples

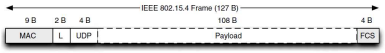


Very Best Case: IP Header Compression

- When **Routing Over Multiple Hops**,
  - ✓ The **compression still performs** quite well
  - ✓ **Compressing the IPv6 header** to **7 octets**

(The dispatch byte + the HC1 byte + 1 byte for the IP hop limit field + 2 bytes for the sources address + 2 bytes for the destination address).

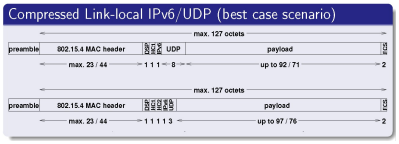
6LoWPAN Header Compression



Minimal UDP/6LoWPAN (16-bit addressing)

6LoWPAN header compression example (L = LoWPAN header).

Compressed Link Local IPv6/UDP (Best Case)



Very Best Case: IP Header Compression

- With the **Use of Global Addresses**,
  - ✓ The **IPv6 header** can be **compressed to 4 bytes**

(2 bytes for the IPHC encoding + 1 byte for context identifier extension + 1 byte for the IP hop limit field + 0 bytes for the source and destination addresses with the interface IDs).

Very Best Case: UDP Header Compression

- The **compressed UDP header** only **requires 2 bytes**

(1 byte for the NHC header + 1 byte for the compressed source and destination UDP ports) when the UDP checksum field is elided.

Thank U

IoT: Internet Layer : Routing

Dr. E.SURESH BABU
Assistant Professor
Computer Science and Engineering Department
National Institute of Technology, Warangal.
Warangal, TS, India.

Session Outline

1. IoT Routing Protocols
2. Routing Requirements
3. Challenges of IoT Routing
4. Routing Metrics in LLN
5. The Objective Function
6. RPL Routing Protocol
7. RPL Traffic Models
8. RPL Builds DODAG

IoT Routing Protocols

LLN – Low Power and Lossy Networks

- **IoT are Low Power And Lossy Networks (LLNs)** consisting **largely of constrained nodes.**
  - ✓ **Heterogeneous network, different nodes**

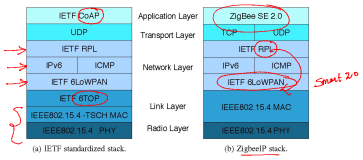
LLN – Lossy and Unstable Links

- **IoT Networks** are **Lossy and unstable links**,
  - ✓ **IP based smart devices** interconnected by **lossy links**
  - ✓ Typically supporting **low data rates, relatively low packet delivery rates**
  - ✓ A **lossy link** is not just a **link with higher BER** uniformly **distributed errors**.
  - ✓ **Packet drops** on **lossy links** are **extremely frequent**
  - ✓ The **links** may become **completely unusable** for **quite some time** due to **interference**.

LLN – Low Power and Lossy Networks

- **LLN** are **Traffic patterns** are not simply **point-to-point**,
  - ✓ ...but in many cases **point-to-multipoint** or **multipoint-to-point.**
- **LLN** potentially comprising up to **thousands of nodes.**

Protocol Stack of LLN



(a) IETF standardized stack. (b) ZigbeeIP stack.

Routing Requirements

Routing Requirements

- **Routing requirements** for networks made of **LLNs.**
- **Unicast/Anycast/Multicast**
- **Adaptive Routing**
- **Constraint-based Routing**
- **Traffic characteristics**
- **Configuration and management**
- **Scalability**
- **Node Attribute**
- **Performance**
- **Security**

Challenges of IoT Routing

Challenges of IoT Routing

- The nodes with **limited resources** also have to **connect somehow to the Internet**
  - ✓ To send their **data (to the cloud)**
  - ✓ To be **queried from distance**
- **Multi-hop Communication and Routing**
  - ✓ **Traditional Routing Solutions** are **Too Resource-hungry**
  - ✓ The goal is the **fast and reliable transmission**

Challenges of IoT Routing

- In LLN, **link failures** are usual,
  - ✓ If we would react to them, then **too much overhead**
- In **LLNs dynamic metrics**, changing in time
  - ✓ The **networks can adapt.**
- In LLNs **low data traffic,**
  - ✓ But, **No need to reconfigure** the **entire topology**

Routing Metrics in LLN

Routing Metrics in LLN

- **Routing metrics** are a **critical component** of the **routing strategy.**
- Most of the **IP routing protocols** used in **today's networks** use **static link metrics**
  - ✓ **Link** has **three metrics (e.g., Delays, Bandwidth, Cost)**
  - ✓ The **routing protocol** computes the **shortest path** taking into **account these static link metrics.**

Metric

- A **Metric** is a scalar used to **determine the best path** according to **some objective function**.
- For Example,
  - ✓ if the **link metric** is representative of the **link propagation delay**,
  - ✓ The **path cost** represents the **total propagation delay** to the **destination** and
  - ✓ The **objective function** may specify **finding the shortest path** based on the **propagation delay.**

Constraint

- A **constraint** is used to **include or eliminate links** or **nodes** that do not meet **specific criteria** (referred to as ***Constraint-based Routing***).
- **For Example:** The **objective function** may **not select any path that traverses a node that is battery-operated** or a **link** that does **not provide link layer encryption.**

RPL Uses Link & Nodes Metrics/Constrains

- **RPL** Uses **Link and Nodes Metrics/Constrains**
- Let's consider the **Link Quality Level (LQL).**
  - ✓ The **LQL** is an **integer** between **0 and 3** that characterizes the **link quality (poor, fair, good).**

Routing Objects

- **Routing Objects** that can be used as **a metric or constraint.**
  - ✓ **Aggregated Routing Metrics Versus Recorded Routing Metrics**
  - ✓ **Local Versus Global Metrics**
  - ✓ **The Routing Metrics/Constraints Common Header**
  - ✓ **Node Energy Object**
  - ✓ **Hop-count Object**
  - ✓ **Throughput Object**
  - ✓ **Latency Object**
  - ✓ **Link Reliability Object**
  - ✓ **Link Colours Attribute**

Node Energy Object

- **Energy** is a **critical metric** in **LLNs**, especially in the presence of **battery-operated nodes.**
- Several **levels of granularity** to characterize the **node energy**:
  1. The **node power mode**,
  2. Estimated **remaining lifetime**
  3. **Potentially** some detailed **set of power-related metrics** and **attributes.**

Node Power Mode

- **Three flags** are used to indicate whether the **node is**
  - ✓ **Main-powered,**
  - ✓ **Battery-powered,**
  - ✓ **Energy Scavenging** (Solar Panels, Mechanical, etc.).

Estimated Remaining Lifetime

- **Estimated remaining lifetime** provides some indication of the **power level** for both **battery-operated** and **scavenging nodes**.

Link Reliability Object

- The most **popular reliability metric** is the **expected transmission (ETX) count metric**, which characterizes
  - ✓ The **average number of packet transmissions** required to **successfully transmit a packet.**
  - ✓ The **ETX** is consequently **tightly coupled** to the **throughput along a path.**

IETF Method : ETX

- IETF method for **computing ETX values.**
- The **ETX** is a **link-specific quantity** and the **technique** used
  - ✓ To **compute the ETX value** should be **independent of the link layer** and
  - ✓ **Not specified** by the **network layer** that only carries it for **routing protocol decisions.**

IETF Method : ETX

- The **ETX for a path** is **computed** as the **sum of the ETX** for **each link along the path**
  - ✓ For Example : **RPL** reports **cumulative path ETX**.

The Objective Function

The Objective Function

- **LLNs** uses several **Objective Function (OFs)**
- **LLNs** deployments **greatly vary** with **different objectives**
  - ✓ A **single network** may **support traffic** with **very different requirements** in terms of **path quality.**

Example : The Objective Function

- Consider a **mixed network** with
  - ✓ **Battery-powered** and **main-powered nodes**
  - ✓ A **variety** of **high and low bandwidth links**, and
  - ✓ **Two main applications** (**Telemetry And Critical Alarms**).

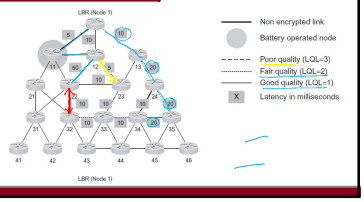
Example One : The Objective Function

- It might be **extremely useful** for **each node** supporting **both applications** to be able to **use two paths.**
  - ✓ One uses **"time sensitive" path** for **alarms** where the **objective** is to have a **short delay** and a **highly reliable path** with **no constraint** on the **type of nodes along the path to the destination**.

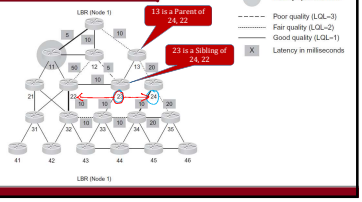
Example Two: The Objective Function

- It might be **extremely useful** for **each node** supporting **both applications** to be able to **use two paths.**
  - ✓ Another uses "**not time sensitive path** " for the **telemetry traffic** where it is **beneficial to not traverse any battery-operated node** to **preserve energy**
  - ✓ The **objective** would be to **minimize hops** to **avoid traffic congestion** in the network.

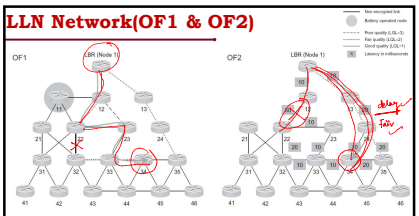
LLN Network



LLN Network



LLN Network(OF1 & OF2)



Objective Function (OF)

- An **OF** defines how **nodes select paths** towards **DODAG root**
  - ✓ Dictates rules on how **nodes satisfy a optimisation objective** (e.g., **minimise latency**)
  - ✓ Actual **routing metrics and constraints** carried **ICMPv6 control messages**
- A **rank in the DODAG** reflects its **distance from the root**

Objective Function (OF)

- There is a **single Objective Function** per **RPL Instance**
  - ✓ An **instance** can **comprise one or more DODAGs** (share same OF)

Routing Protocol for Low-Power and Lossy Networks (RPL)

Overview of RPL - Routing Protocol for LLNs

- **RPL** is an **extensible proactive IPv6 distance vector protocol**
  - ✓ Builds a **Destination Oriented Directed Acyclic Graph (DODAG)**
  - ✓ RPL supports **shortest-path constraint based routing** applied to **both links and nodes**
  - ✓ Supports **MP2P**, **P2MP** and **P2P** between **devices (leaves)** and a **root (border router)**

Overview of RPL - Routing Protocol for LLNs

- **RPL** specifically designed for **"Lossy" networks**
  - ✓ Should **not be categorised** as a **WSN routing protocol**
  - ✓ Uses **underlying link layer technologies** (**802.15.4**, **PLC**, **Low Power Wireless**, **BLE**, **RFID**)

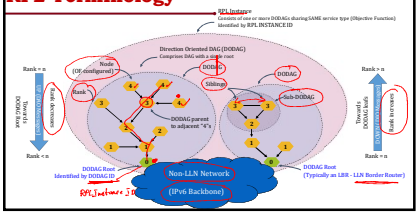
RPL Characteristic of LLN

- One of the **RPL characteristic of LLN networks** the **data traffic** is usually **limited and the control traffic** should be **reduced** whenever possible to **save bandwidth and energy**.

RPL: IP Routing Protocol for Smart Objects

- In compliance with the **layered architecture of IP**,
  - ✓ **RPL** does **not rely on any particular features** of a specific **link layer technology**.
- **RPL** is designed to be able to **operate** over a **variety of different link layers**.

RPL Terminology

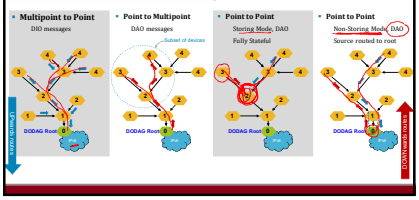


RPL Traffic Models

RPL Traffic Models

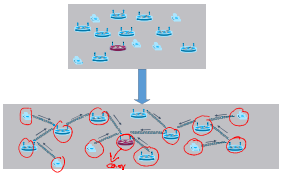
- **RPL** built on a **robust topology**
  - ✓ It supports **simple and complex traffic models** like
    - **Multipoint-to-point,**
    - **Point-to-multipoint** and
    - **Point-to-point.**

RPL Supported Traffic Flows



RPL Builds DODAG

What is a Directed Acyclic Graph?



RPL Builds DODAG

- **RPL** builds a **Destination Oriented Directed Acyclic Graph (DODAG)**
  - ✓ Has **only one route** from **each leaf node to the root** in which all the **traffic from the node will be routed to.**
- **DODAGs** also offer **redundant paths** for LLNs.
  - ✓ if the **topology permits**, **RPL** may provision **more than one path** between **a node** and the **DODAG root.**

RPL DODAGs

- Every **hop** to the **root MUST** have an **alternate path**
- A **DODAG** will **ensure nodes** always have a **path up towards the root**
- A **DODAG** is identified by {**RPL Instance ID**, **DODAG ID**}

RPL Instance

RPL Instance

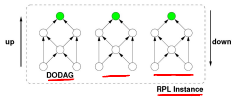
- **RPL** can form **multiple instances**
  - ✓ Each **instance honours** a particular **routing objective/constraint**
  - ✓ **Instance** consists **one or more DODAGs** derived from the ***same*** **objective function**
  - ✓ **Nodes** select a **parent (towards root)** based on **metric**, **OF** and **loop avoidance**

RPL Instance

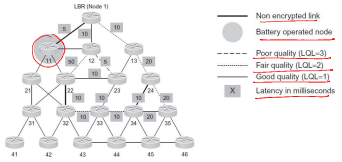
- Allows **upwards and downwards routing** (from **DODAG root**)
- **Trickle timers** used to **suppress redundant messages**
  - ✓ **Saves on energy and bandwidth**

RPL Instance

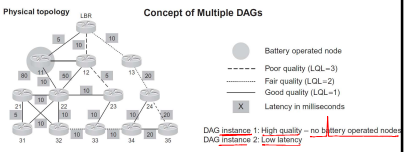
- **RPL Instance** might be useful to **form different topologies** according to **various sets of constraints** and **OFs.**
- An **RPL node** may **join multiple DODAG instances**;



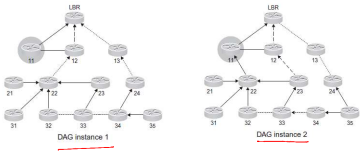
Use of Multiple DODAG and the Concept of RPL Instance



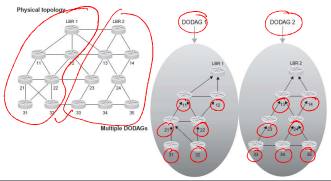
Concept of Multiple DAGs



DAG Instance-1 and DAG Instance-2



RPL Multiple DODAGs within a DODAG Instance.



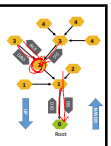
RPL Control Messages

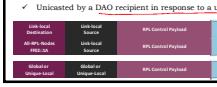
RPL Control Messages

- **RPL** specifies **three messages** (using ICMPv6)
  1. **The DODAG Information Object (DIO),**
  2. **The DODAG Destination Advertisement Object (DAO), and**
  3. **The DODAG information solicitation message (DIS).**

ICMPv6 RPL Control Messages

- **DIO - DODAG Information Object**
  - ✓ Used for DODAG discovery, formation and maintenance
- **DIS - DODAG Information Solicitation Message**
  - ✓ Used to probe for DIO messages from RPL nodes
- **DAO - DODAG Destination Advertisement Object**
  - ✓ Propagates prefix availability from leaves up the DODAG
  - ✓ Supports P2MP and P2P traffic
- **DAO-ACK - DODAG Destination Advertisement Object**
  - ✓ Unicasted by a DAO recipient in response to a unicast DAO message





Most RPL control messages have scope of a link

DAO/DAO-ACK in non-storing mode passes over multiple hops

RPL Control Messages

DAG Information Object (DIO)

- A DIO carries information that allows a node to discover an RPL Instance, learn its configuration parameters and select DODAG parents

DAG Information Solicitation (DIS)

- A DIS solicits a DODAG Information Object from an RPL node

RPL Control Messages

Destination Advertisement Object (DAO)

- A DAO propagates destination information upwards along the DODAG

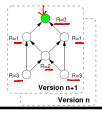
DODAG Rank

❖ The **DODAG Rank** is the **rank of the node** sending the **DIO message.**

Version Numbers and DODAG Rank

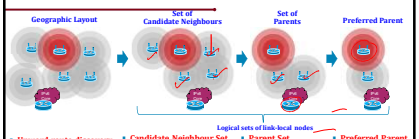
Definition

A node's Rank defines the node's individual position relative to other nodes with respect to a DODAG root. The scope of Rank is a DODAG Version.



Use of the Rank for DODAG Parent Selection

DODAG Neighbours and Parent Selection (Upward Routes)



- **Upward route discovery**
  - Comprises three logical sets of link-local nodes
  - Neighbours are learnt from DIO advertisements
- **Candidate Neighbour Set**
  - Subset of nodes reachable via link-local multicast
  - Elements in the set may belong to different DODAG versions
- **Parent Set**
  - Consists of nodes with a higher rank (lower #)
  - Elements in the set must belong to SAME DODAG version
- **Preferred Parent**
  - Preferred next-hop to the DODAG Root
  - Multiple preferred parents possible if ranks are equal

For Example

❖ A **node A** has a **rank less than** the **rank of a node B**

✓ If floor(Rank(A)/Min-Hop-Rank-Increase) is less than floor(Rank(B)/Min-Hop-Rank-Increase).

❖ A **node A** has a **rank greater than** the **rank of a node B**

✓ if floor (Rank(A)/Min-Hop-Rank-Increase) is greater than floor (Rank(B)/Min-Hop-Rank-Increase).

❖ **Two nodes A and B** are **siblings** if

✓ floor (Rank(A)/MinHopRankIncrease) == floor (Rank(B)/MinHopRankIncrease).

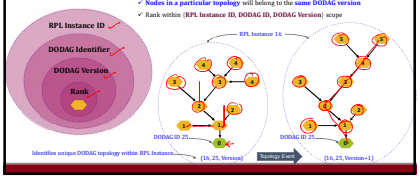
❖ **A and B are siblings** if the **integer portion** of their **rank** is equal.

RPL Identifiers

❖ **Four values** used to **identify and maintain DODAG** topology

✓ **Nodes in a particular topology** will belong to the **same DODAG version**

✓ Rank within (**RPL Instance ID, DODAG ID, DODAG Version**) scope



Route Construction and Building

Route Construction and Forwarding Rules

Route Construction

- Up routes towards nodes of decreasing rank (parents)
- Down routes towards nodes of increasing rank
  - Nodes inform parents of their presence and reachability to descendants
  - Source route for nodes that cannot maintain down routes

Forwarding Rules

Forwarding Rules

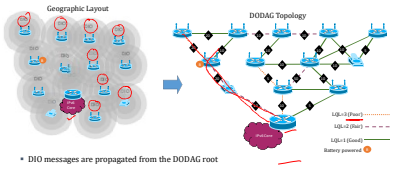
- All routes go upwards and/or downwards along a DODAG
- When going up, always forward to lower rank when possible, may forward to sibling if no lower rank exists
- When going down, forward based on down routes

Node Initialization/ Discovery Process

- When a **node starts its initialization process**
  - ✓ it may **decide to remain silent** until it **hears a DIO advertising an existing DODAG.**
  - ✓ Alternatively, the **node may issue a DIS message** to probe the **neighbourhood** and **receive DIO messages** from its **neighbours more quickly**.
  - ✓ **Unicast DIOs** are sent in **reply to unicast DIS messages** and also include a **complete set of DODAG configuration options.**

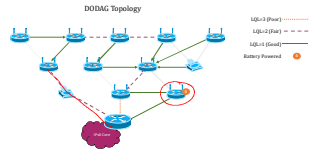
A Step-by-Step Example

DODAG Example

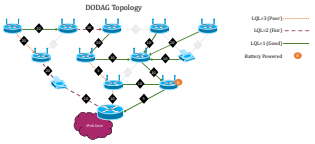


- DIO messages are propagated from the DODAG root

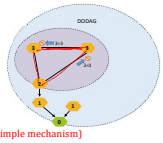
OF: Use High Quality Links, Avoid battery powered nodes



OF: Low Latency Paths only

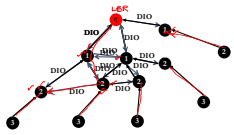


RPL Loop Detection



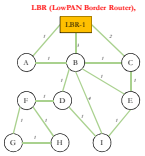
- Data path validation used to check for loops (Simple mechanism)
  - ✓ IPv6 options header carries rank of transmitter
- If node receives packet with rank <= to its own, drop packet
  - ✓ Detection happens when link is actually used.

RPL Route Construction with Rank

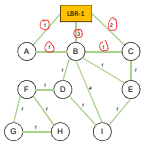


RPL DAG Construction

LBR (LowPAN Border Router),

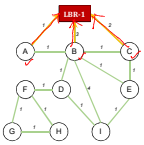


DAG Construction



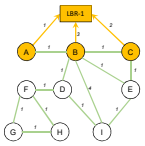
- **Objective Code Point for example**
  - Metric: ETX
  - Objective: Minimize ETX
  - Depth computation: Depth = ETX

DAG Construction



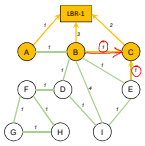
- LBR-1 multicasts RA-DIO
- Nodes A, B, C receive and process RA-DIO
- Nodes A, B, C consider link metrics to LBR-1 and the optimization objective
- The optimization objective can be satisfied by joining the DAG rooted at LBR-1
- Nodes A, B, C add LBR-1 as a DAG parent and join the DAG

DAG Construction



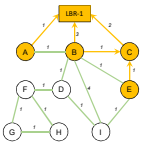
- Node A is at Depth 1 in the DAG,
- Node B is at Depth 3, Node C is at Depth 2
- Nodes A, B, C have installed default routes with LBR-1 as successor

DAG Construction



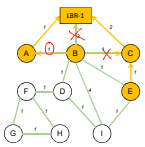
- The RA timer on Node C expires
- Node C multicasts RA-DIO
- LBR-1 ignores RA-DIO from deeper node
- Node B can add Node C as *alternate* DAG Parent, remaining at Depth 3
- Node E joins the DAG at Depth 3 by adding Node C as DAG Parent

DAG Construction: Depth



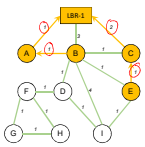
- Node A is at Depth 1, and can reach ::/0 via LBR-1 with ETX 1
- Node B is at Depth 3, with DAG Parents LBR-1, and can reach ::/0 via LBR-1 or C with ETX 3
- Node C is at Depth 2, ::/0 via LBR-1 with ETX 2
- Node E is at Depth 3, ::/0 via C with ETX 3

DAG Construction



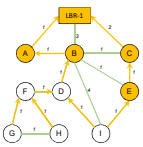
- The RA timer on Node A expires
- Node A multicasts RA-DIO
- LBR-1 ignores RA-DIO from deeper node
- Node B adds Node A
- Node B can improve to a more optimum position in the DAG
- Node B *removes* LBR-1, Node C as DAG Parents

DAG Construction: Depth



- Node A is at Depth 1, ::/0 via LBR-1 with ETX 2
- Node B is at Depth 2, ::/0 via A with ETX 2
- Node C is at Depth 2, ::/0 via LBR-1 with ETX 2
- Node E is at Depth 3, ::/0 via C with ETX 3

Final DODAG Construction



- DAG Construction continues…
- And is continuously maintained

Thank U