

IoT App Development



6TiSCH Technology

6TiSCH Survey Papers: <https://ieeexplore.ieee.org/document/8823863>,
<https://doi.org/10.1016/j.comnet.2024.110759>

Dr. Manas Khatua

Associate Professor

Dept. of CSE, IIT Guwahati

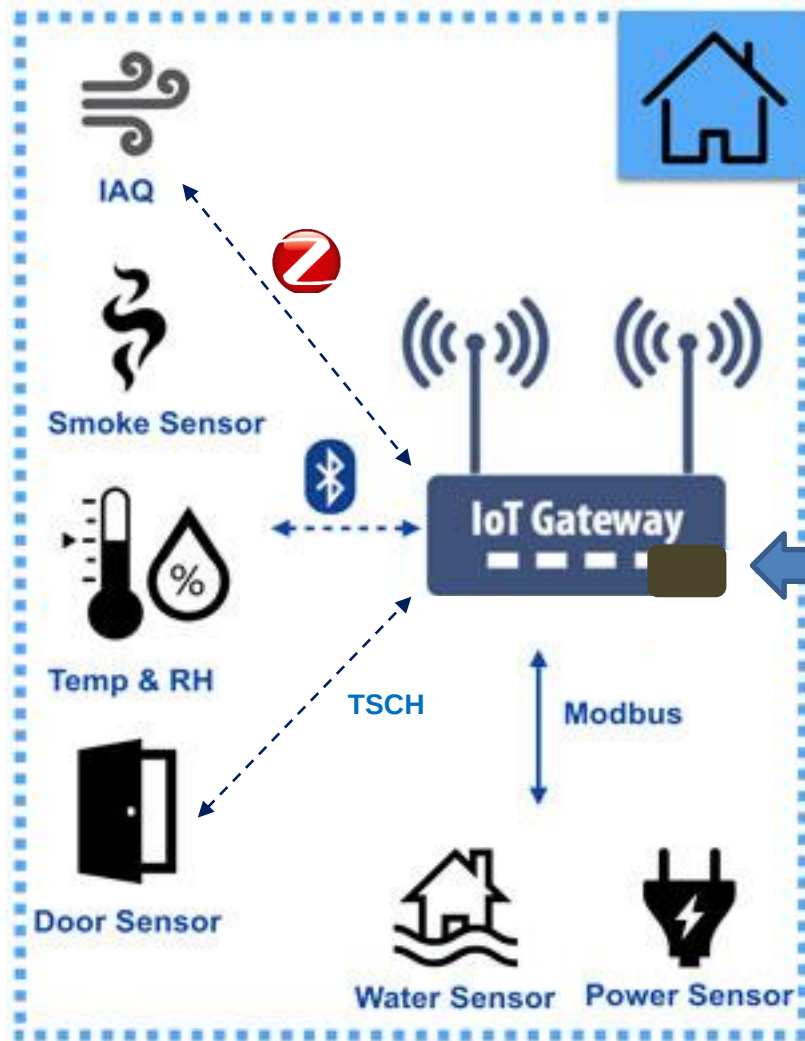
E-mail: manaskhatua@iitg.ac.in

IoT Access Technologies

- there are many IoT technologies in the market today



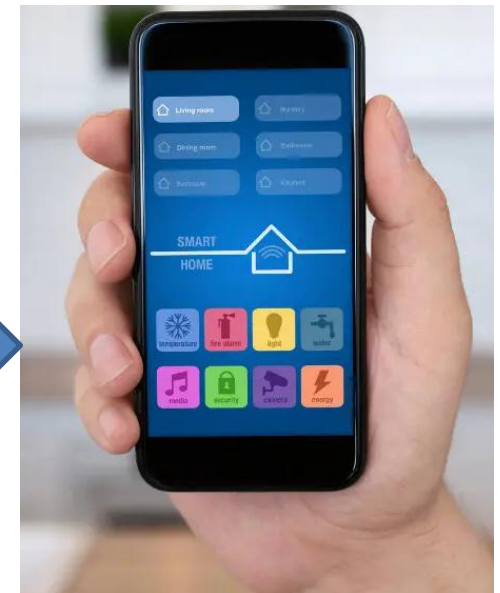
Application Viewpoint



Remote Real-Time Building Monitoring

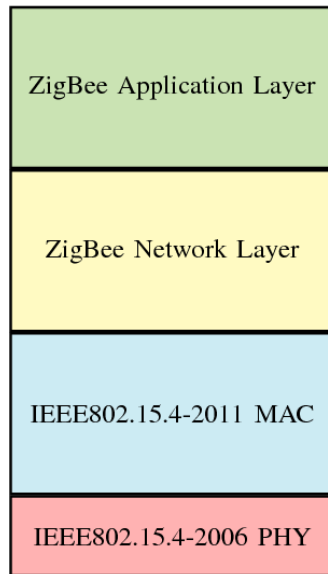
How?

using Existing
Internetworking
Infrastructure
(e.g. IP)

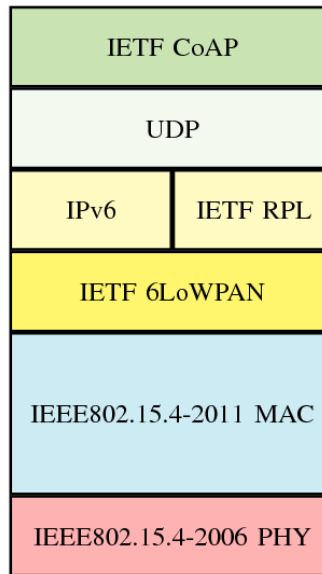


6TiSCH Working Group

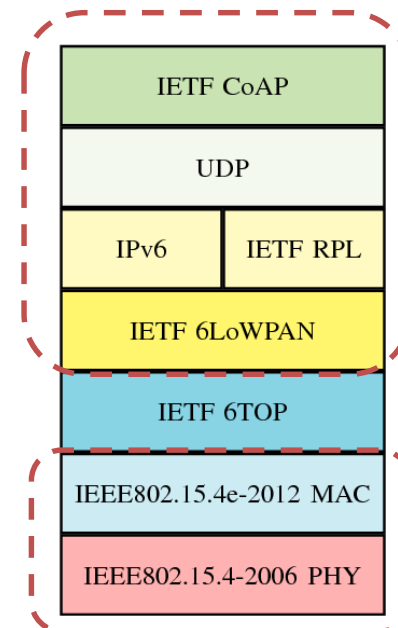
- 6TiSCH Working Group created by IETF in **October 2013**
- **Goal:** To integrate **TSCH** with the **IPv6** through the IETF upper stack
 - To **enable IPv6 over TSCH** mode of IEEE 802.15.4e
 - Defining a new **functional entity in charge** of **TSCH scheduling**



(a) ZigBee stack.



(b) ZigBeeIP stack.



(c) 6TiSCH stack.

Survey Article: “IETF 6TiSCH: A Tutorial” <https://ieeexplore.ieee.org/document/8823863>

6TiSCH Architecture (RFC 9030)

- Considers low-power lossy-network (LLN)
- Allow more than 1000 nodes
- Nodes are in same IPv6 subnet
- 6LoWPAN header compression (HC) is used to transmit packet
- Presence of high-speed backbone (e.g. WiFi mesh) to connect all nodes
- Backbone is connected to the Internet through a Gateway
- Constrained nodes are attached to backbone through backbone router (BBR) or 6LBR

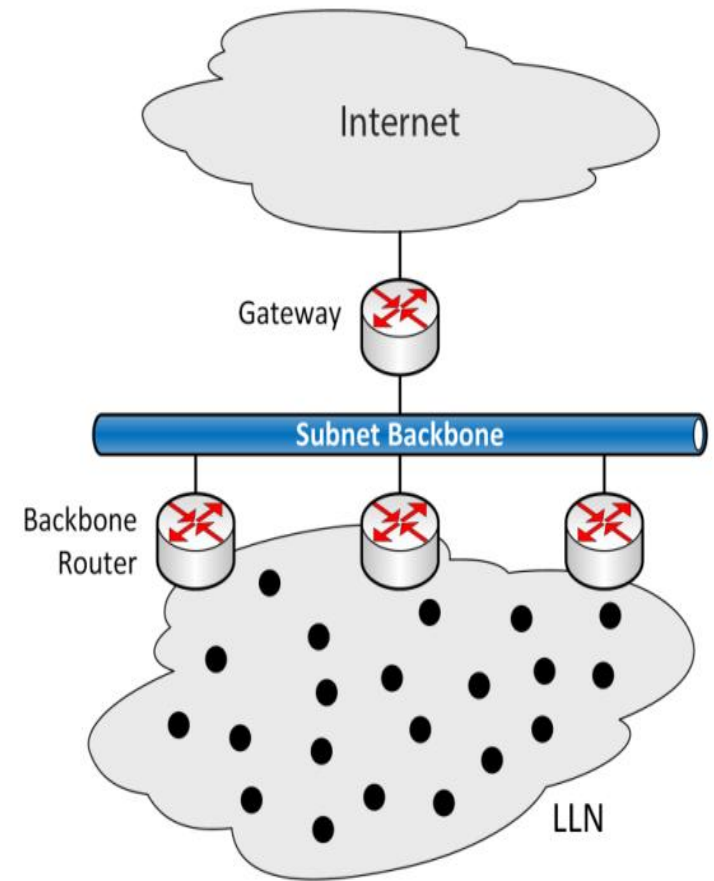


Fig. 6TiSCH Architecture

RFC 9030: An Architecture for IPv6 over the Time-Slotted Channel Hopping Mode of IEEE 802.15.4 (6TiSCH)

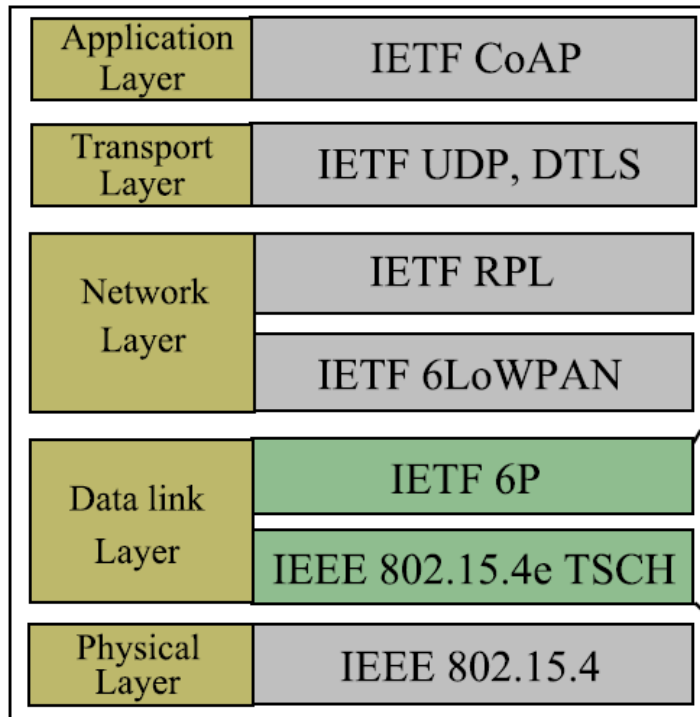
Need for 6TiSCH Operation Sub-Layer



- In 6TiSCH, the TSCH MAC mode is placed under an IPv6-enabled protocol stack:
 - Constrained Application Protocol (CoAP)
 - IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL)
 - IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN)

- TSCH does not define
 - Policies to build and maintain the data communication schedule
 - Mechanisms to adapt the resources allocated between neighbor nodes as per the data traffic flow features – change in data rates, neighbor change, etc.
 - Techniques to allow differentiated treatment of packets – data and control packets
 - Mechanisms to match the schedule to the multi-hop paths maintained by RPL

6Top Sub-Layer



6TiSCH Protocol Stack

A new sublayer, called **6Top**

- Works on top of TSCH
- **Build and manage TSCH schedule**
 - add/delete links/cells
- 6top also collects connectivity information
 - Monitors the performance of cells

RFC 9033 (Minimal scheduling)

RFC 8480 (6P protocol)

RFC 8137 (Container for 6P)

RFC 9031 (Constrained Join)

RFC 8180 (minimal 6TiSCH)

6TiSCH Operational sublayer

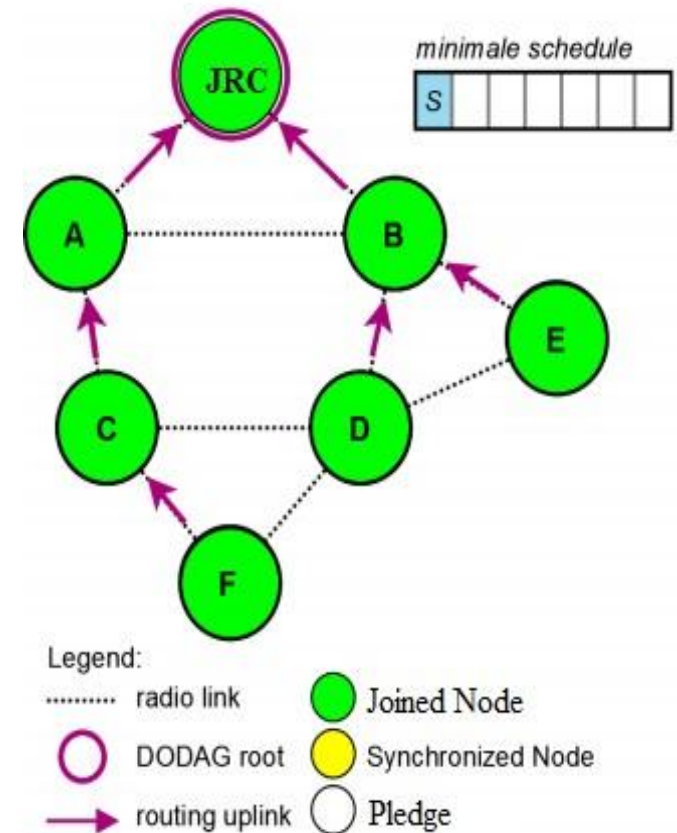
6TiSCH Protocol Stack

Application (CoAP)	RFC8613	(2019) <i>object security extension to CoAP</i>
	RFC7252	(2014) <i>base CoAP specification</i>
Routing (RPL)	RFC6554	(2012) <i>header format for routing header</i>
	RFC6553	(2012) <i>header format for RPL option</i>
	RFC6552	(2012) <i>Objective Function, RPL algorithm</i>
	RFC6550	(2012) <i>base RPL specification</i>
Adaptation (6LoWPAN)	RFC8505	(2018) <i>neighbor discovery and registration</i>
	RFC8138	(2017) <i>routing header compression</i>
	RFC8025	(2016) <i>mechanism for extending 6LoWPAN</i>
	RFC6282	(2011) <i>updated base 6LoWPAN specification</i>
	RFC4944	(2007) <i>base 6LoWPAN specification</i>
Scheduling (6TiSCH)	draft-ietf-6tisch-msf (RFC 9033)	(WIP) <i>distributed scheduling algorithm</i>
	RFC8480	(2018) <i>6P, distributed scheduling protocol</i>
	RFC8137	(2017) <i>container for 6P</i>
	draft-ietf-6tisch-minimal-security (RFC 9031)	(WIP) <i>security framework for 6TiSCH</i>
	RFC8180	(2017) <i>minimal 6TiSCH</i>
Physical layer	IEEE802.15.4	(2015) <i>2.4 GHz, 50-200 m range, 250 kbps, 127 byte frames</i>

Source: Xavier Vilajosana et al., "IETF 6TiSCH: A Tutorial" *IEEE Communications Surveys & Tutorials*, 22(1), 2020, pp. 595– 615.

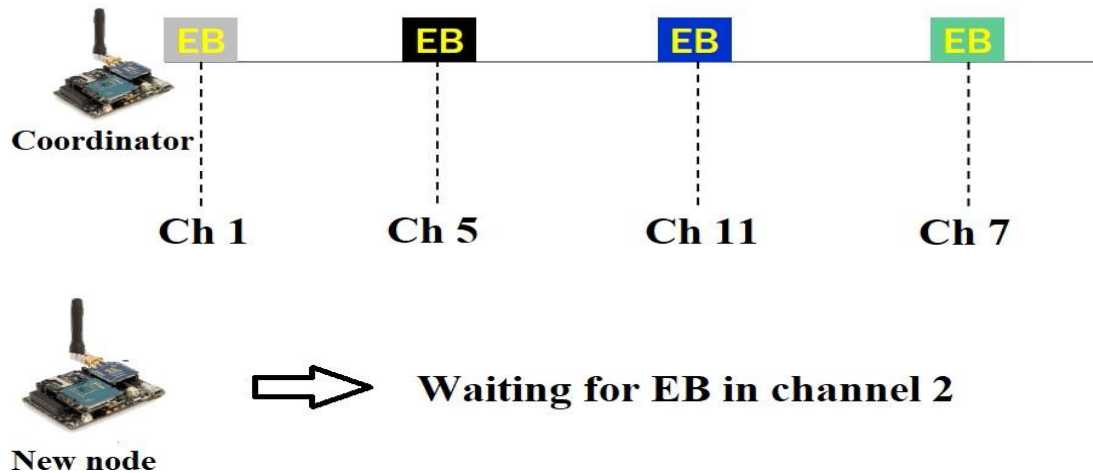
6TiSCH Network Formation Process

- **Join Registrar/Coordinator (JRC)** starts the formation process
 - Enhanced Beacon (EB)
 - Routing Information (DIO)
- Pledge (new node) scans for EB on a random channel (**RDC 100%**)
- After receiving an EB, pledge synchronized with the underlying TSCH network (**RDC ~1%**)
- Synchronized node waits for **DIO**, after exchanging **JRQ & JRS**
- After receiving DIO, pledge becomes **joined node**; can transmit own packets
- Network formation completes when all the pledge join the network



Why Network formation is an issue?

- Channel hopping feature of TSCH
 - A pledge does not know in which channel transmission of control packets is happening



- Limited resource allocated for control packets
 - Only one shared cell in a slotframe

TSCH v/s 6TiSCH network formation



- TSCH formation/synchronization time
 - A pledge gets **synchronized** with a TSCH network after **receiving** a valid **EB** frame
 - The **time** when a pledge **receives its first EB frame** is considered as **TSCH synchronization time** or **TSCH formation time**
- 6TiSCH formation time
 - When a **TSCH synchronized** node **receives** a valid **DIO packet**, it becomes a **6TiSCH joined node**
 - The DIO receiving time is considered as **6TiSCH joining time**

Thanks!



Figures and slide materials are taken from the following sources:

1. David Hanes *et al.*, “IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things”, 1st Edition, 2018, Pearson India.