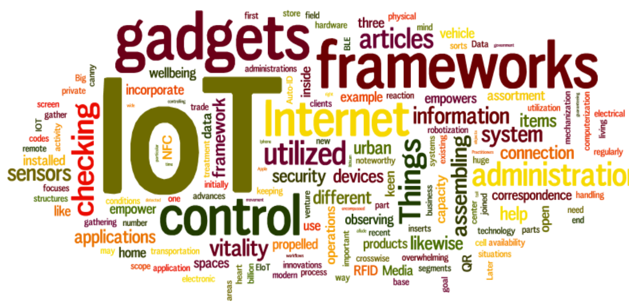
# CS578: Internet of Things

## 6LoWPAN:

## IPv6 over 802.15.4




Dr. Manas Khatua

Assistant Professor,

Dept. of CSE, IIT Guwahati

E-mail: manaskhatua@iitg.ac.in

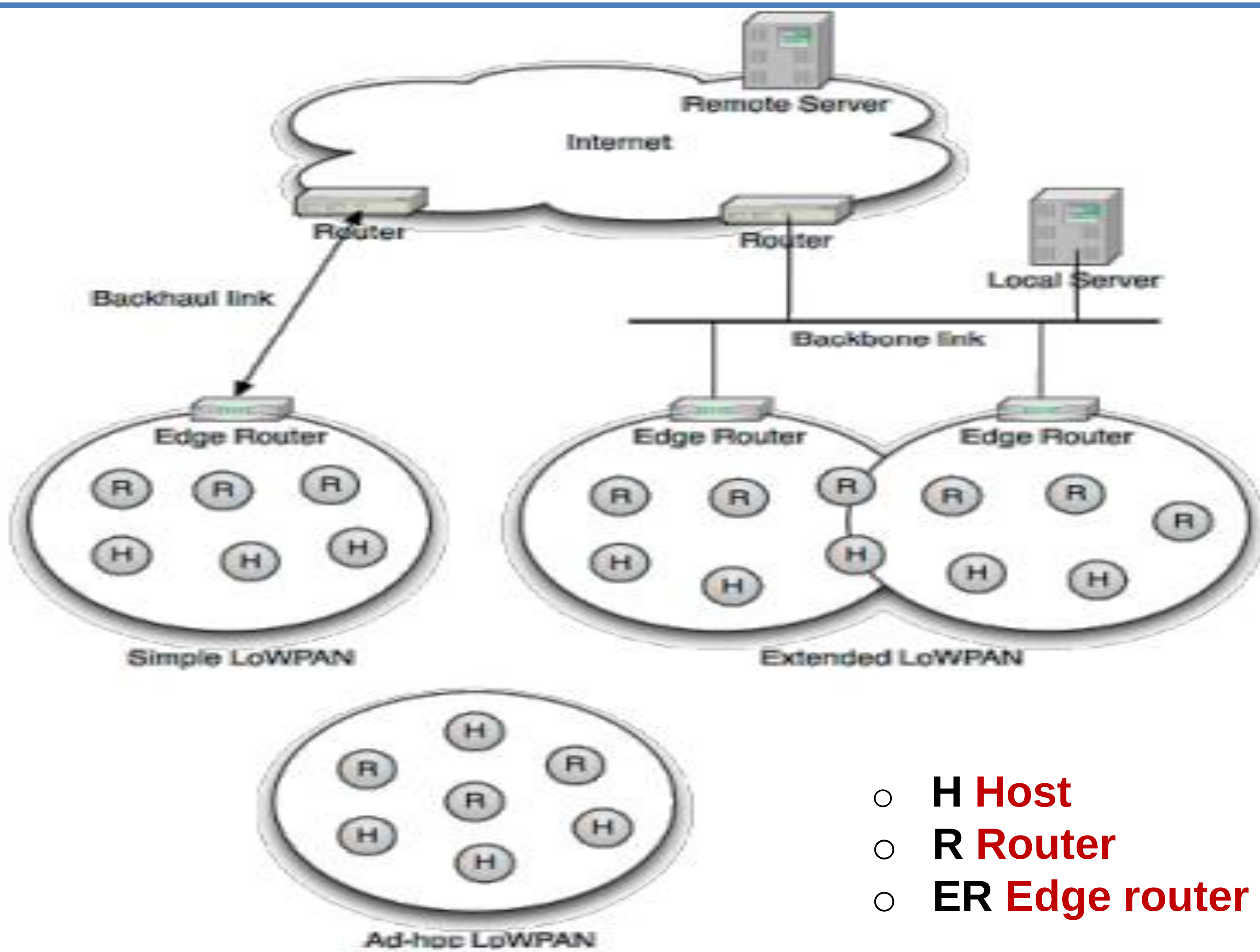
"*Peace comes from within. Do not seek it without.*" – **Gautama Buddha**

# 6LoWPAN

## 6LoWPAN: An Open IoT Networking Protocol

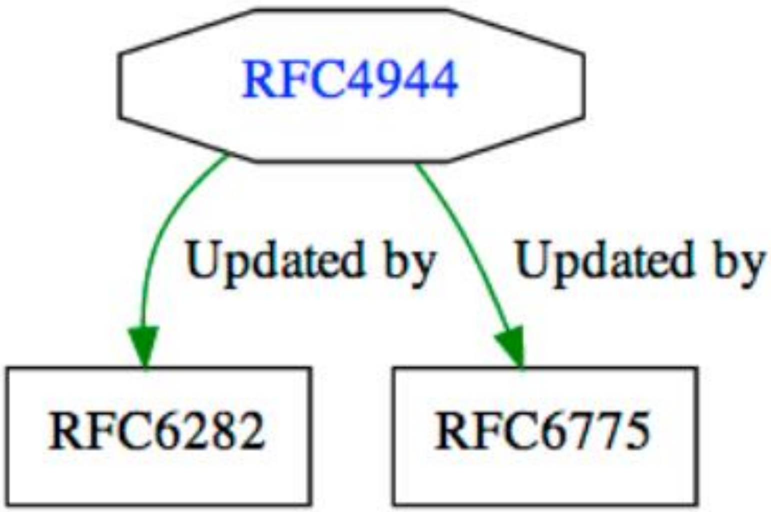
- Open: Specified by the Internet Engineering Task Force (IETF)
  - Specifications available without any membership or license fee
- IoT: Making "Things" Internet-aware
  - Usage of IPv6 to make use of internet protocols
  - Leverage the success of open protocols in contrast to proprietary solutions
- Networking: Stopping at layer 3
  - Application layer protocols are flexible and can vary

# 6LoWPAN Architecture




- **H Host**
- **R Router**
- **ER Edge router**

# RFCs




This document specifies an IPv6 header compression format for IPv6 packet delivery in 6LoWPAN

This document describes simple optimizations to IPv6 Neighbor Discovery, its addressing mechanisms, and duplicate address detection for 6LoWPAN


RFC4919: This document describes the overview, assumptions, problem statement, and goals for transmitting IP over IEEE 802.15.4 networks.

RFC6568: This document investigates potential application scenarios and use cases for LoWPANs

RFC6606: This document provides the problem statement and design space for 6LoWPAN routing. Defines how 6LoWPAN formation and multi-hop routing could be supported.

# Device Characteristics

- Dedicated to specific task/ not general purpose

**Limited hardware resources:**

- Low processing power (microcontroller/ dsp)
- Little memory
- Low power

**Limited networks capabilities:**

- Short range
- Low bitrate
- Small message-size

# Usage Scenarios

- Building automation
- Industrial automation
- Logistics
- Environmental Monitoring
- Personal/ Health Monitoring
- etc.

# Motivation

## Benefits of IP over 802.15.4 (RFC 4919):

- The pervasive nature of IP networks allows use of existing Infrastructure.
- IP-based technologies already exist, are well-known
- Open and freely available specifications vs. Closed proprietary solutions.
- Tools for diagnostic, management already exist.
- IP-based devices can be connected readily to IP-based networks, without gateways or proxies.

# Why IPv6?

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

# IPv6 Challenges

## 1. Header Size Calculation

| | | | | |
|---|---|---|---|---|
| Frame Header (25) | LLSEC (21) | IPv6 Header (40) | UDP | Payload (33) |

- IPv6 header: 40 octets
- UDP header: 8 octets
- 802.15.4 MAC header: up to 25 octets (null security)

  **or** 25+21=46 octets (AES-CCM-128)

✓ With 802.15.4 frame size of 127 octets, we have following space left for application data!

  - **127-(8+40+25) = 54 octets (null security)**
  - **127-(8+40+46) = 33 octets (AES-CCM-128)**

# Cont…

## 2. IPv6 MTU Requirements

- ✓ IPv6 requires that links support an MTU of 1280 octets
- ✓ Link-layer fragmentation / reassembly is needed

## 3. IP assumes devices are always on

- ✓ But embedded devices may not have enough power and duty cycles

## 4. Multicast support

- ✓ IEEE 802.15.4 & other radios do not support multicast (expensive)

# Goals of 6LoWPAN

- Define adaptation layer to match IPv6 MTU requirements
  - by fragmentation/reassembly
- Specify methods to do IPv6 Address formation
  - By stateless address auto configuration
- Specify/use header compression schemes.
- Methods for mesh broadcast/multicast below IP

# Adaptation Layer

- The 6LoWPAN protocol is an adaptation layer
  - allowing to transport IPv6 packets over 802.15.4 links .

- Adaptation layer sits between Datalink and (IP) Network layers

| | | |
|---|---|---|
| L5 Application Layer | Application | Application |
| L4 Transport Layer | TCP \| UDP \| ICMP | UDP \| ICMPv6 |
| L3 Network Layer | IP | IPv6 |
| | | 6LoWPAN |
| L2 Data Link Layer | Ethernet MAC | IEEE 802.15.4 MAC |
| L1 Physical Layer | Ethernet PHY | IEEE 802.15.4 PHY |

# Cont...

## Adaptation layer mainly performs the following functionalities:

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

# 6LoWPAN Headers

- All LoWPAN encapsulated datagrams are prefixed by an encapsulation header stack.

- Each header in the stack starts with a header type field followed by zero or more header fields

## Dispatch Type & header

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0 1| Dispatch    |   type-specific header
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- ✓ Defined by 1st & 2nd bits = 01.
- ✓ Dispatch **->** identifies type of header following dispatch header
- ✓ Type-specific header which is determined by dispatch header.

# Cont...

| Bit Pattern | Short Code | Description |
|---|---|---|
| 00 xxxxxx | NALP | Not A LoWPAN Packet |
| 01 000001 | IPv6 | uncompressed IPv6 addresses |
| 01 000010 | LOWPAN_HC1 | HC1 Compressed IPv6 header |
| 01 010000 | LOWPAN_BC0 | BC0 Broadcast header |
| 01 111111 | ESC | Additional Dispatch octet follows |
| 10 xxxxxx | MESH | Mesh routing header |
| 11 000xxx | FRAG1 | Fragmentation header (first) |
| 11 100xxx | FRAGN | Fragmentation header (subsequent) |

**Dispatch value bit pattern**

- ✓ NALP **->** Not a part of LOWPAN encapsulation, should be discarded.
- ✓ IPv6 **->** uncompressed IPv6 header
- ✓ LoWPAN_HC1 **->** Compressed IPv6 header.
- ✓ LoWPAN_BC0 **->** for meh broadcast/multicast support
- ✓ ESC **->** supports for dispatch values larger than 127

# Cont...

## ❑ Mesh Addressing Type & Header

```
                    -                   -                   -
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|1 0|V|F|HopsLft| originator address, final address
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

- ✓ 1st & 2nd bits = 10.

- ✓ OA **:** link layer address of originator.
- ✓ FDA **:** link layer address of final-destination

- ✓ V**:** 0,if OA is 64-bit address
  1,if 16-bit address

- ✓ F**:** 0,if FDA is 64 bit.
  1, if FDA is 16-bit address

- ✓ HopsLf**t:** 4 bit, decremented by each forwarding node

# Cont...

## ❑ Fragmentation type and Header

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|1 1 0 0 0|    datagram_size    |         datagram_tag          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 \_ dispatch _/
```

**First Fragment**

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|1 1 1 0 0|    datagram_size    |         datagram_tag          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|datagram_offset|
+-+-+-+-+-+-+-+-+
 \_ dispatch _/
```

**Subsequent Fragments**

Datagram size : 11 bit, encode the size of entire packet.
Datagram tag : 16bit, same for all link fragments of a payload.
Datagram offset : present only in subsequent fragments

# 6LoWPAN Compressions

- **Started with HC1 and HC2 compressions**
  - Assume common values for header fields and define compact forms.
  - Reduce header size by omission

**Omit headers that...**

- Can be reconstructed from L2 layer headers (redundant)
- Contain information not needed or used in the context (unnecessary)

# 6LoWPAN Compressions: HC1

## HC1: Compresses IPv6 headers

| | | |
|---|---|---|
| Version | Traffic Class | Flow Label (20 bit) |
| Payload Length (16 bit) | Next Header | Hop Limit (8 bit) |
| Source Address (128 bit) | | |
| Destination Address (128 bit) | | |

**IPv6 Header:40-Byte**

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 01 | 0 | 0 | 0 | 0 | 0 | 1 | Uncompressed IPv6 address [RFC2460] | 40 bytes |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01 | 0 | 0 | 0 | 0 | 1 | 0 | HC1 |

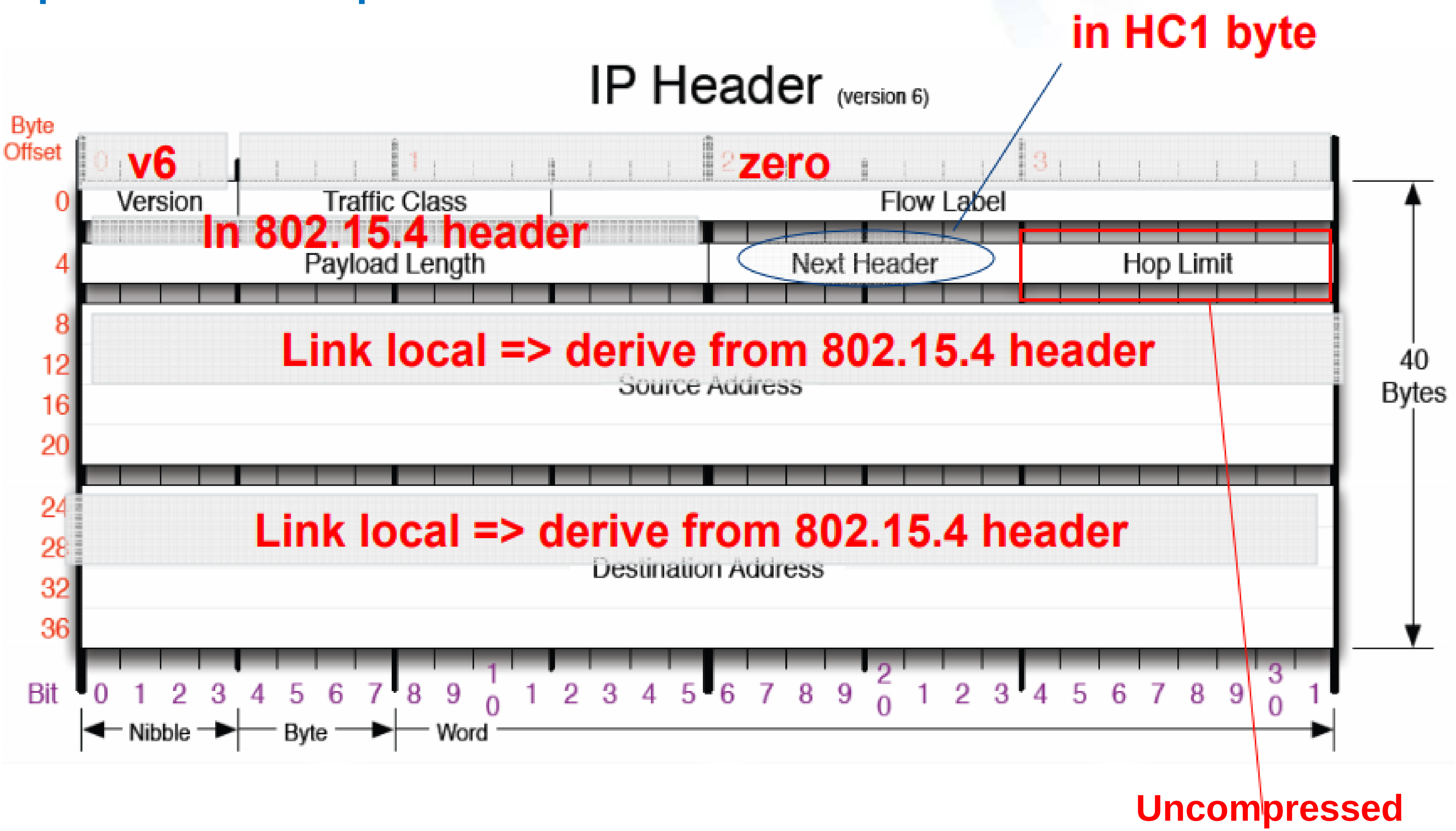
Fully compressed: 1 byte

Source address -> Derived from link address
Destination address -> Derived from link address

Traffic class & Flow level -> Zero (if ECN, DS, Flow level all zero)
Next Header -> TCP,UDP,0r ICMPv6

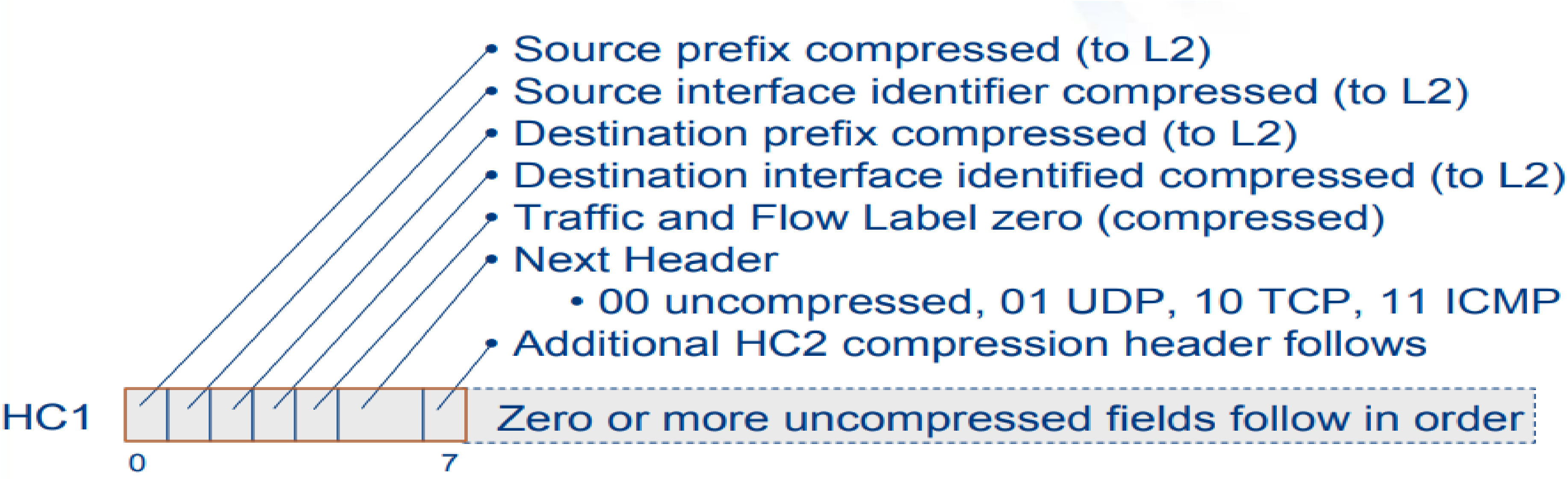
# Cont...

## Ipv6 Header compression



# Cont...

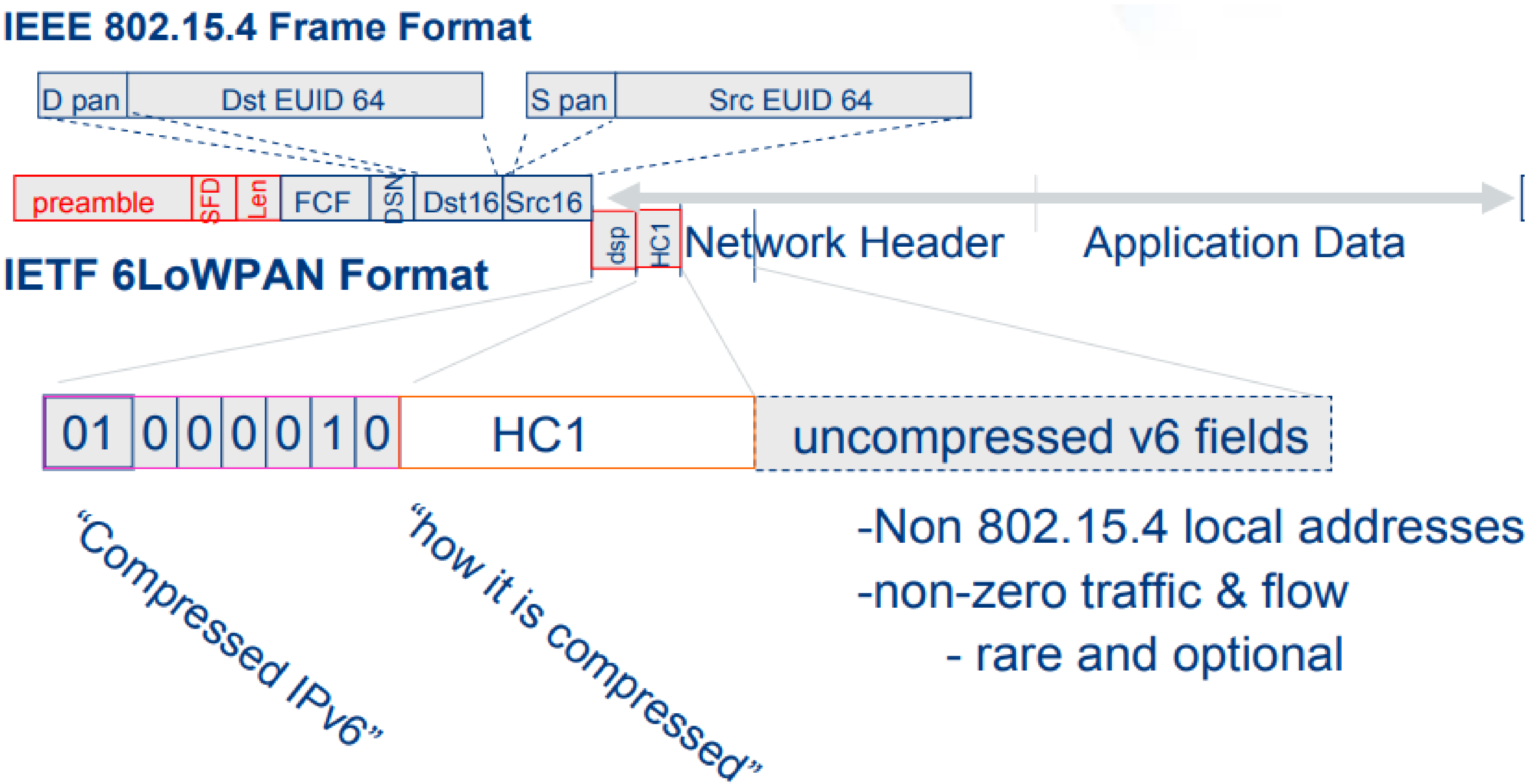
## HC1 Compressed IPv6 Header




- ✓ IPv6 address <prefix64 || Interface ID> for nodes in 802.15.4 subnet derived from the link address.
  - PAN ID maps to a unique IPv6 prefix
  - IID generated from EUID 64 or PAN ID or short address
- ✓ **Hop Limit is the only incompressible IPv6 header field.**

# Cont...

## 6LoWPAN: Compressed IPv6 Header



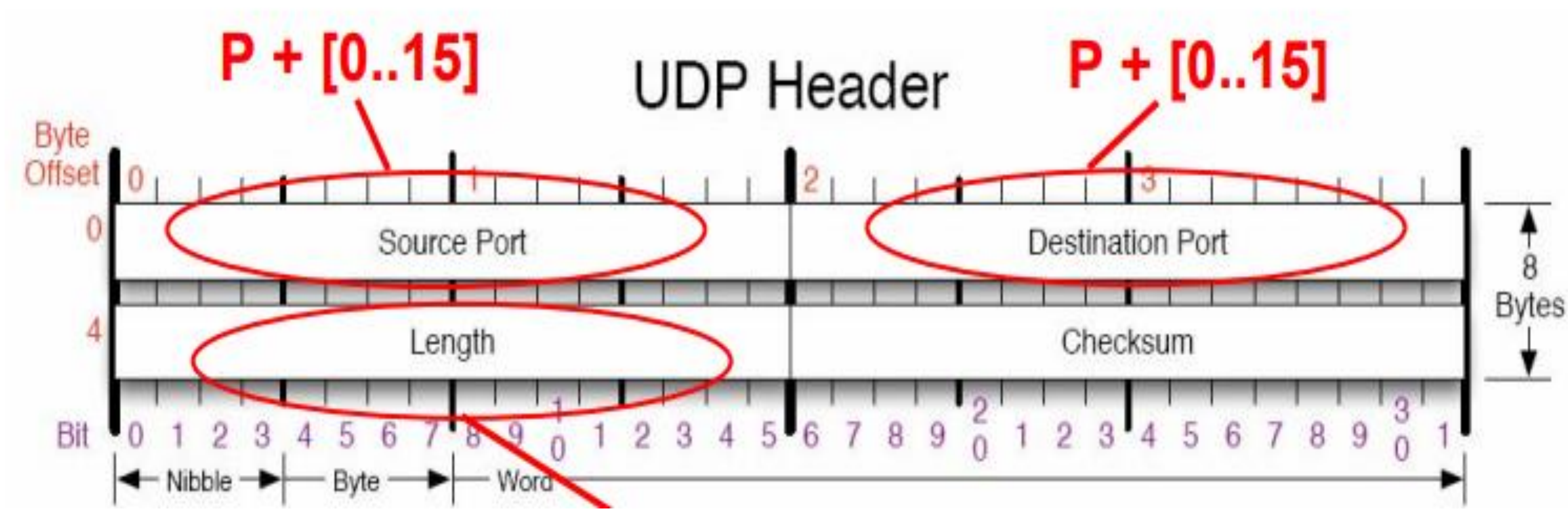
# 6LoWPAN Compressions: HC2

## HC2:Compresses UDP Headers

- ✓ Sourse port=P+4bits,P=61616
- ✓ Destination port=P+4bits,P=61616
- ✓ Length derived from IPv6 length
- ✓ Checksum is always carrie inline

| | |
|---|---|
| Source port | Destination port |
| UDP length | Checksum |

**UDP header format**




**From 15.4 header**

# Limitations of HC1&HC2

- ✓ LoWPAN_HC1 & LoWPAN_HC2 are insufficient for most practical uses.
- ✓ Effective only for link-local unicast communication,
- ✓ So, they are usually not used for application layer data traffic.

## So RFC 6282 came as an advancement

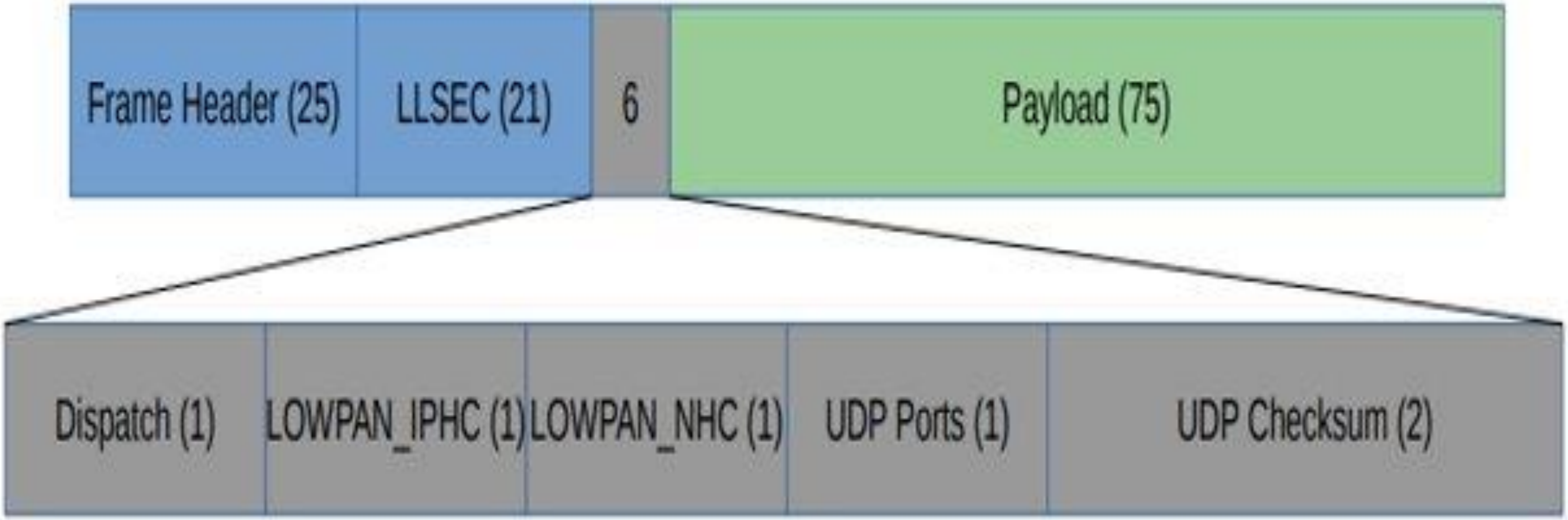
- Defines LoWPAN_IPHC
  - Better compression for global and multicast addresses not only link- local
  - Compress header fields with common values: version, traffic class, flow label, **hop-limit**

- Defines LoWPAN_NHC (for arbitrary next headers)
  - Adds ability to elide UDP checksum

✓ Possible to invent your own scheme if you have repeating usage patterns in your use case

# The Header Size Solution

**The 48-byte( IPv6 + UDP header)** could in the best cases be **reduced to 6 bytes.**



# 6LoWPAN Stacked Headers

- 6LoWPAN uses
  - stacked headers, and
  - extension headers. (analogous to IPv6)

- 6LoWPAN headers define the capability of each sub-header.
- Three sub-headers are defined:
  - *Mesh addressing*,
  - *Fragmentation,*
  - *Header compression*

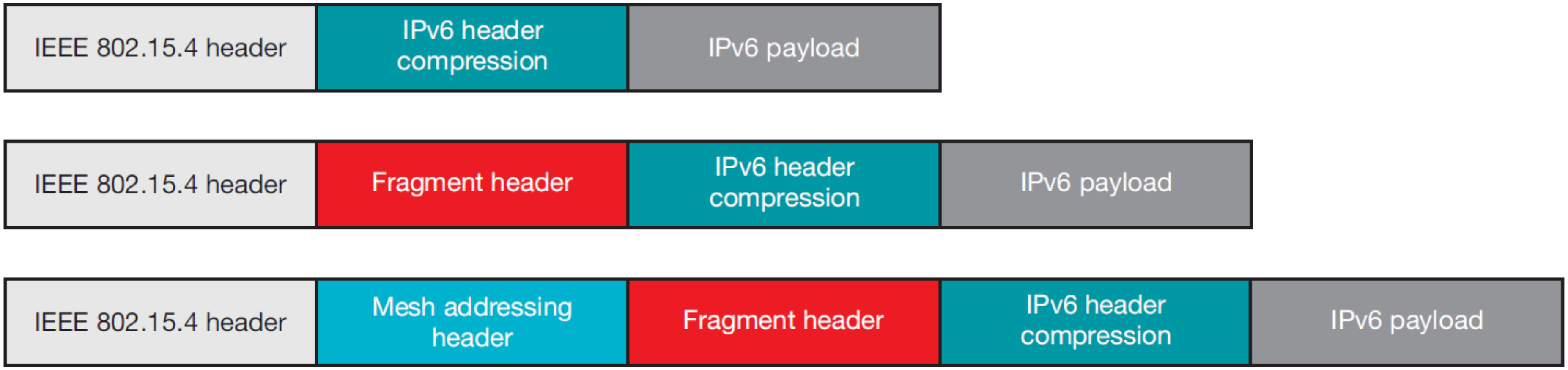



*Figure 4. 6LoWPAN stacked headers*

# 6LoWPAN Frame Format

- LoWPAN encapsulated IPv6 datagram:

```
+---------------+-------------+---------+
| IPv6 Dispatch | IPv6 Header | Payload |
+---------------+-------------+---------+
```

- LoWPAN encapsulated HC1 compressed IPv6 datagram:

```
+--------------+------------+---------+
| HC1 Dispatch | HC1 Header | Payload |
+--------------+------------+---------+
```

# Cont…

- HC1 compressed IPv6 datagram that requires Mesh addressing:

| | | | | |
|---|---|---|---|---|
| Mesh Type | Mesh Header | HC1 Dispatch | HC1 Header | Payload |

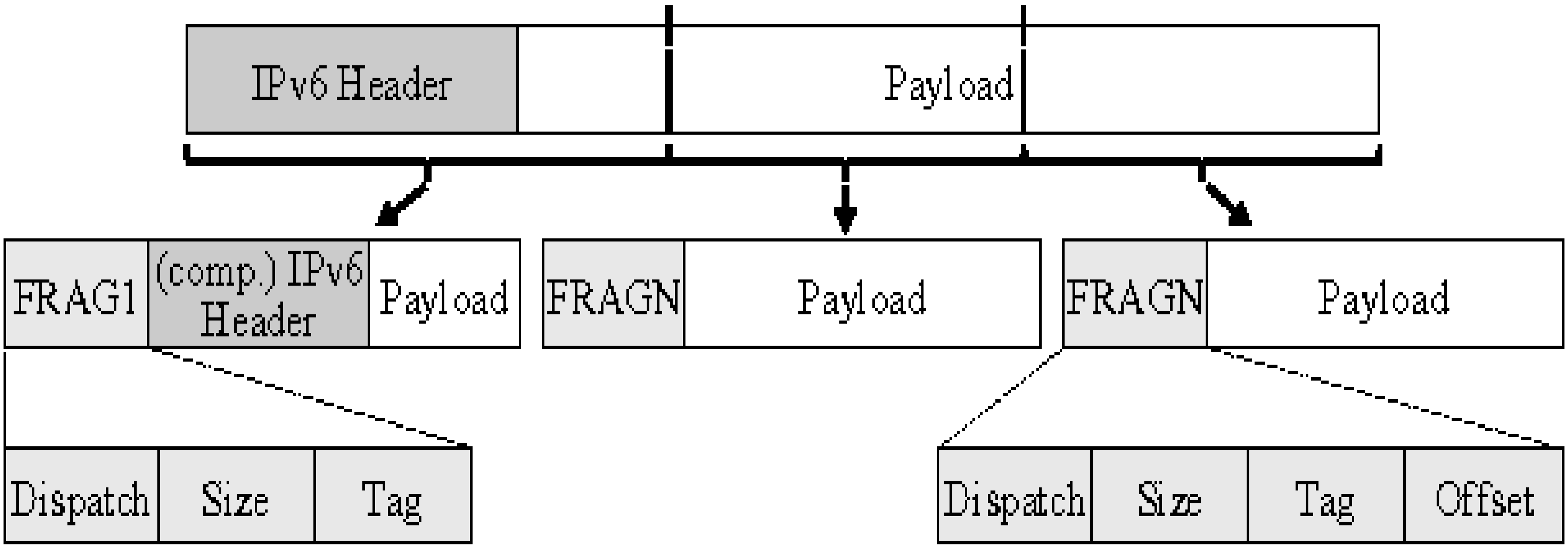
- HC1 compressed IPv6 datagram that requires Fragmentation:

| | | | | |
|---|---|---|---|---|
| Frag Type | Frag Header | HC1 Dispatch | HC1 Header | Payload |

# Fragmentation and Reassembly

## Fragmentation Principles (RFC 4944)

- When an IPv6 packet exceeds link-layer payload size then segments the packet into fragments.




**6LoWPAN packet structure of FRAG1 and FRAGN**

# Cont...

- Only the 1st fragment carries end-to-end routing information.
- First fragment carries a header that includes the datagram size and a datagram tag .
- Subsequent fragments carry the datagram size, the datagram tag, and the offset.
- Time limit for reassembly is 60 seconds.
- For a lost fragment we need to resend entire set of fragments.

# Forwarding Mechanisms

**6LoWPAN supports two routing mechanisms:**

- **Mesh-under**
- **Route-over**

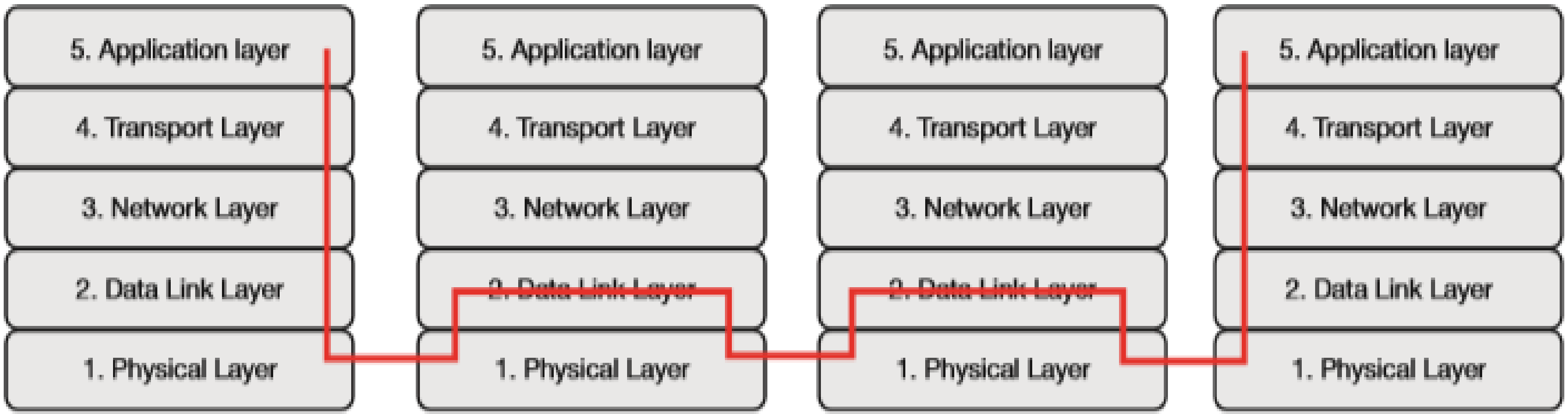
**Mesh-under**

- ✓ Uses L2 addresses to forward data
- ✓ Only IP router is the edge router
- ✓ Individual fragments may take different paths.
- ✓ Suitable for small and local networks

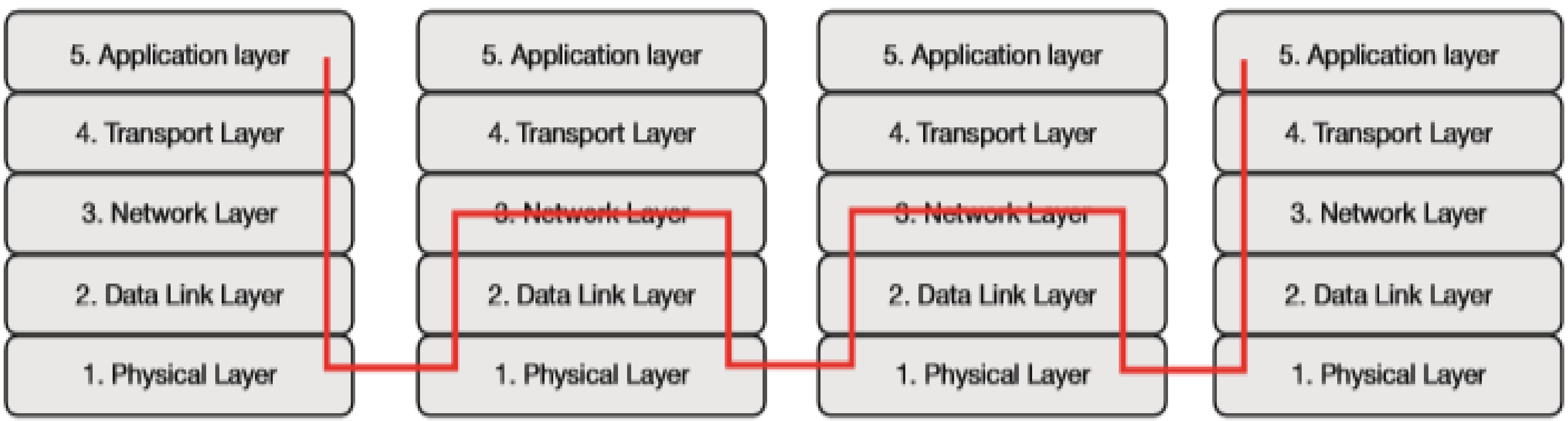
**Route-over**

- ✓ Uses L3 addresses to forward data.
- ✓ Each hop acts as an IP router.
- ✓ As routing decision taken on a per packet basis, all fragments are sent to same path

# Cont...



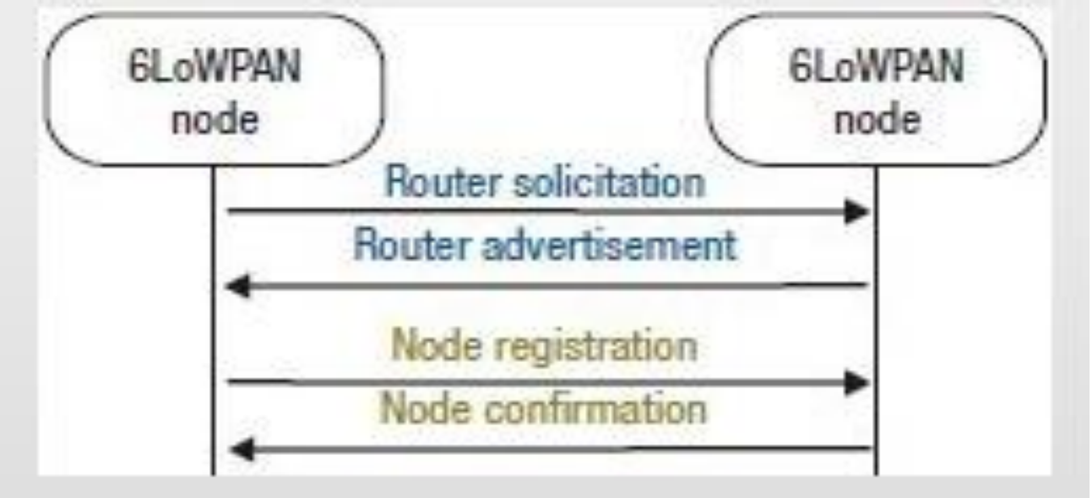

**Mesh-under**




**Route-over**

# Neighbour Discovery Protocol

- Hosts uses to discover Routers
- Duplicate Address Detection(DAD)
- Uses ICMPv6
  - RS(Router Solicitation) , RA (Router Advertisement)
  - NS(Neighbour Solicitation ) , NA (Neighbour Advertisement)




6LoWPAN Neighbour Discovery

# Cont...

## Router Solicitation and Advertisement


1—ICMP Type = 133 (RS)

Src = link-local address (FE80::1/10)

Dst = all-routers multicast address (FF02::2)

Query = please send RA

2—ICMP Type = 134 (RA)

Src = link-local address (FE80::2/10)

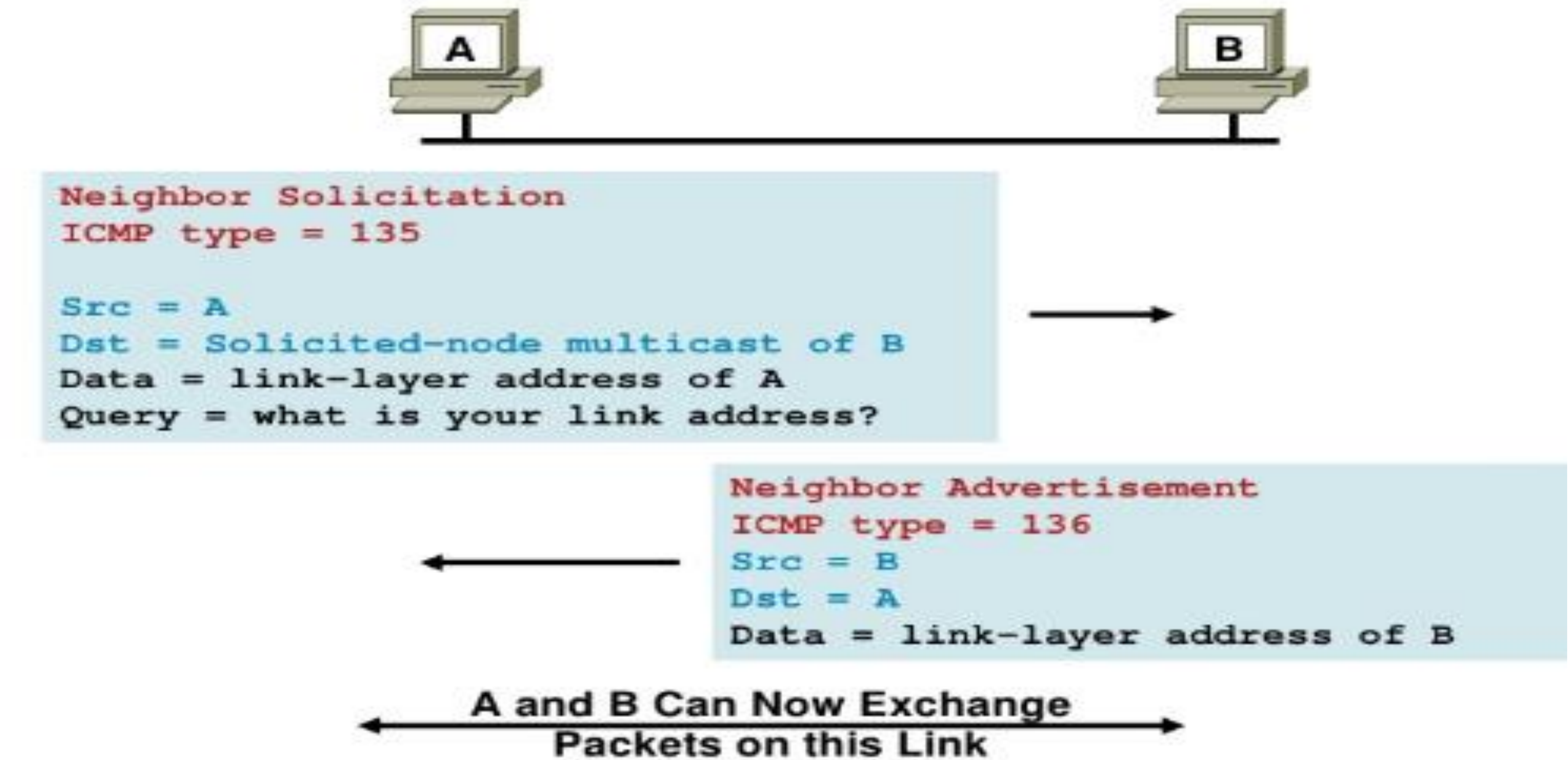
Dst = all-nodes multicast address (FF02::1)

Data = options, subnet prefix, lifetime, autoconfig flag

- Router Solicitations (RS) are sent by booting nodes to request RAs for configuring the interfaces
- Routers send periodic Router Advertisements (RA) to the all-nodes multicast address

# Cont...

## Neighbor Solicitation and Advertisement



# Stateless Autoconfiguration

## Introduction of IPv6 addressing:

✓ Addresses are 128 bit long

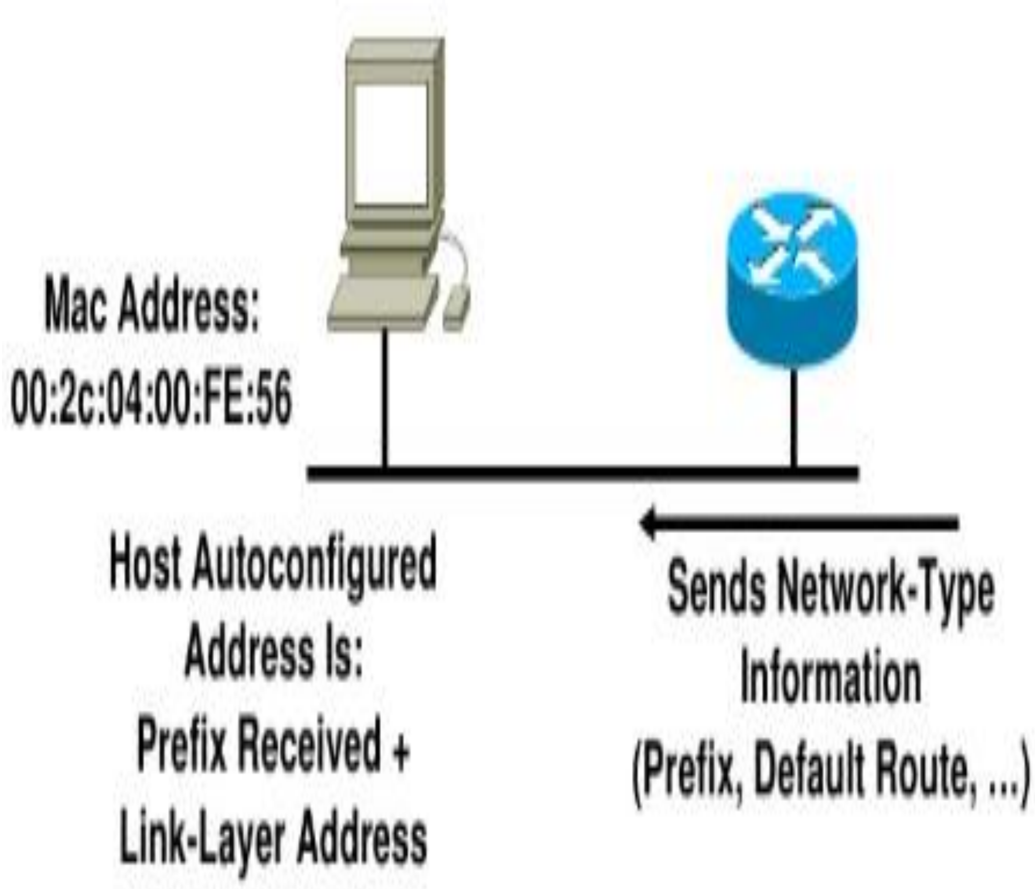
- Divided into 8 hextets,each hextet is 16 bit
- Each character is 4bit,a nibble
- A common configuration is a 48-bit network prefix,16-bit subnet mask and a 64-bit host address
- Eg 2001:1234:ABCD:0001:1023:FD45:0033:0002

✓ IPv6 can perform both stateful and stateless autoconfiguration of address for an interface.

- Allows a node to connect to the internet without DHCP server
- Uses both local and non-local information to genearte its address.

# Cont...

## Steps involved in autoconfiguration are are:

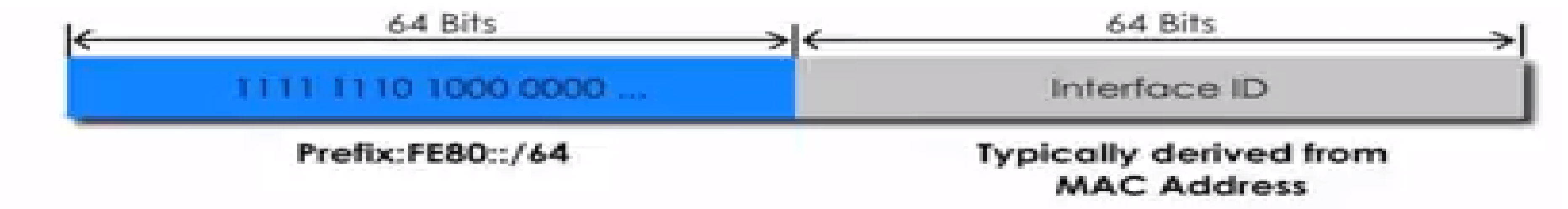



- ✓ Link-local address is generated
- ✓ Host send Router Solicitation to all routers(multicast FF02::2)
- ✓ Routers reply with Router Advertisement and announces prefix used on link.
- ✓ Host generates address by combining the prefix received and host identifier.(EUI-64)
- ✓ Hosts performs DAD
- ✓ If succesful,address becomes active

# Cont…

## How does a host generate a link-local address?

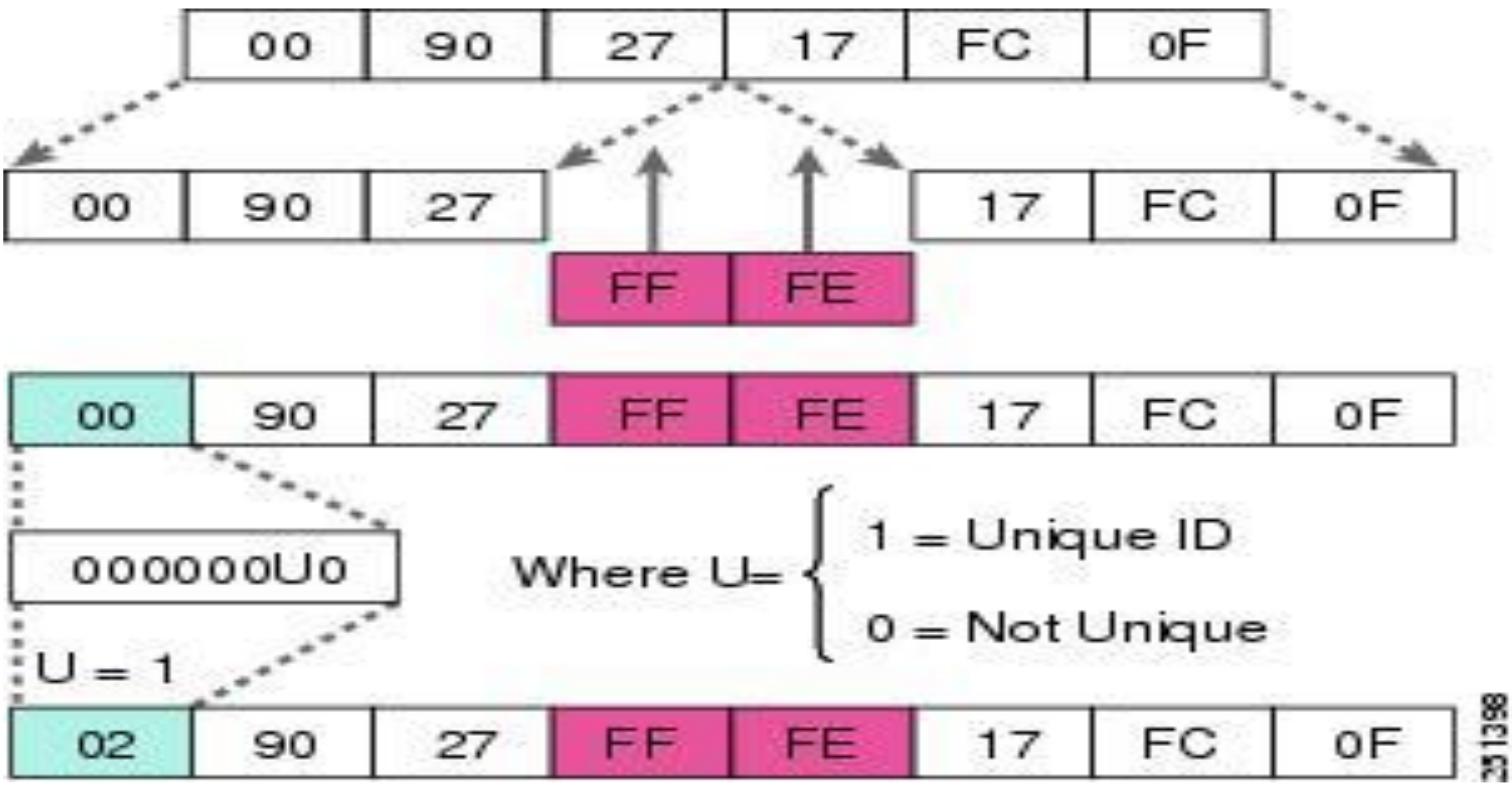
- ✓ Generated when a computer runs IPv6 boots up
- ✓ Valid only for communication on a local network
- ✓ Always have a prefix FE80::/64




Uses EUI-64 to assign itself a unique 64-bit interface ID

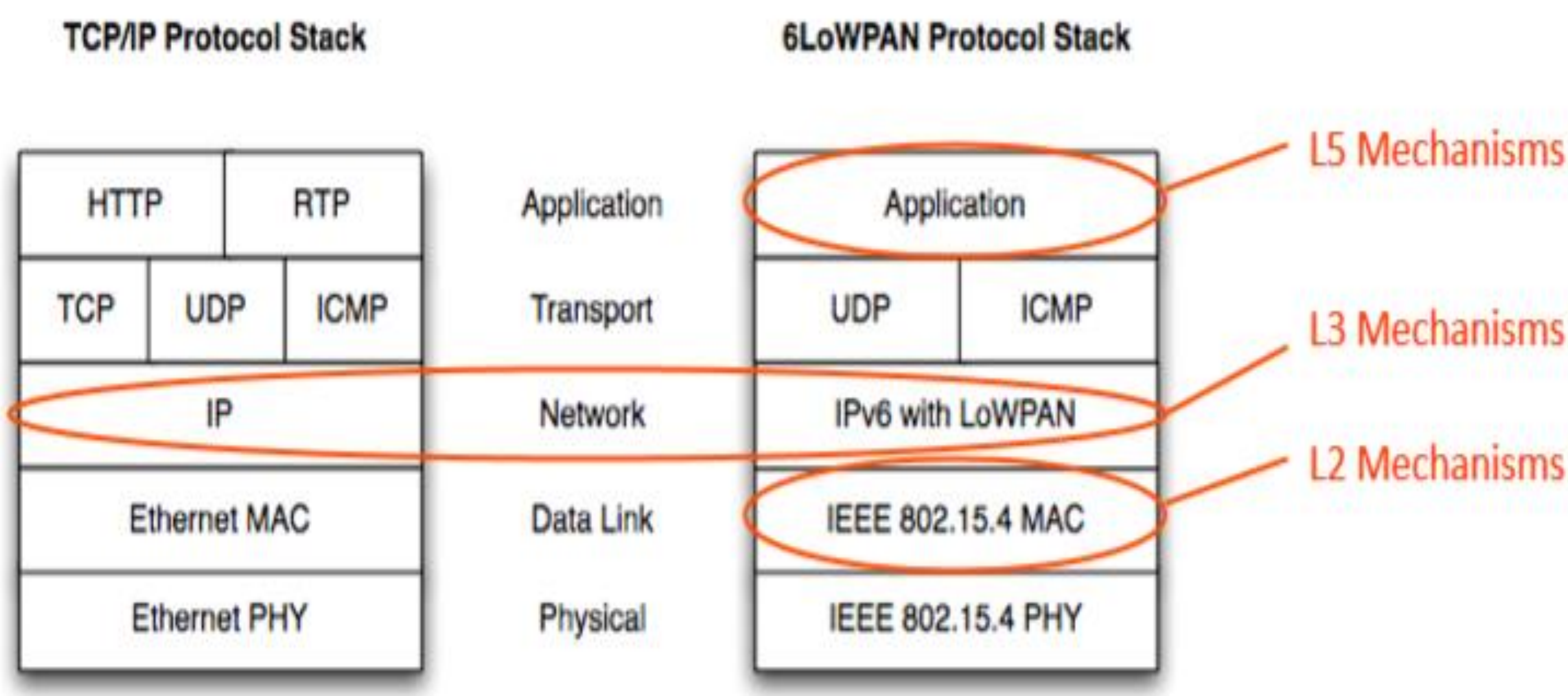
# Cont…

## Generating Interface ID(using EUI-64)



# Security

- Security is important for IOT systems
- It takes advantage of security defined for 802.15.4 link layer
- Also TLS mechanisms works for 6LoWPAN systems



# Conclusion

## 6LoWPAN…

- ✓ Is an **open** standard
- ✓ **Provides an adapter** between IEEE 802.15.4 (L1/2) and IPv6 (L3)
- ✓ **Enables interoperability** between wireless embedded devices
- ✓ **Mesh routing**
- ✓ Provides an important foundation for the Internet of Things (IoT)

# Thanks!