Design and Development of IoT Applications

Dr. –Ing. Vo Que Son

Email: sonvq@hcmut.edu.vn

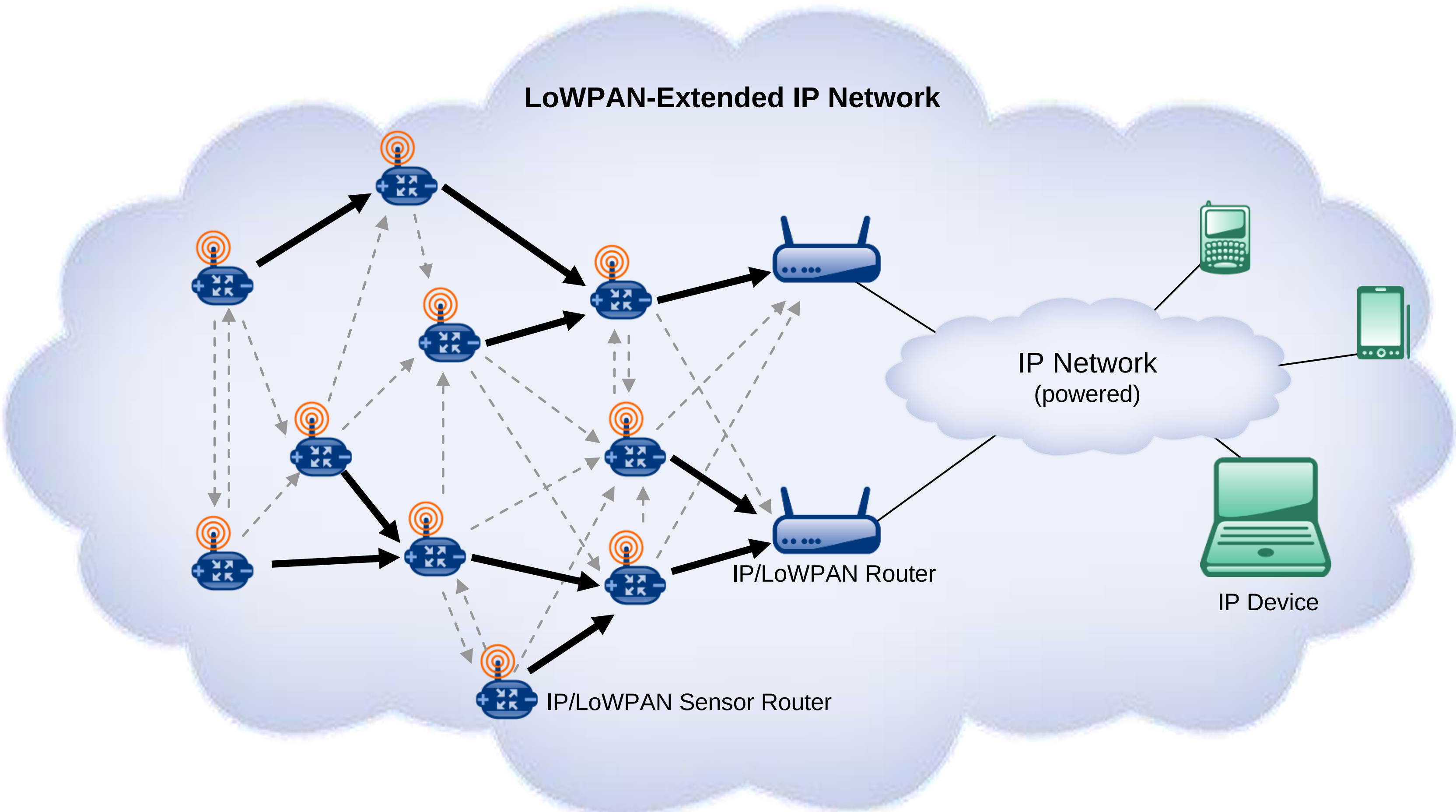
Content

- Chapter 5: Routing in WSNs
  - Multi-hop communication
  - Link characteristics
  - Collection Tree Protocol/DCP
  - Trickle algorithm
- Chapter 6: 6LoWPAN and IPv6
  - Challenges in WSNs and IP
  - IPv6 addressing
  - Fragmentation
  - 6LoWPAN Header compression
  - Bootstrapping
  - Border Router

Question

- If Wireless Sensor Networks represent a future of "billions of information devices embedded in the physical world". ***Why don't they run THE standard internetworking protocol?***
- The Answer:
  - Substantially advances the state-of-the-art in both domains.
  - Implementing IP requires tackling the general case, not just a specific operational slice
    - Interoperability with all other potential IP network links
    - Potential to name and route to any IP-enabled device within security domain
    - Robust operation despite external factors
      - Coexistence, interference, errant devices, ...
  - While meeting the critical embedded wireless requirements
    - High reliability and adaptability
    - Long lifetime on limited energy
    - Manageability of many devices
    - Within highly constrained resources

Low Power Wireless Internet

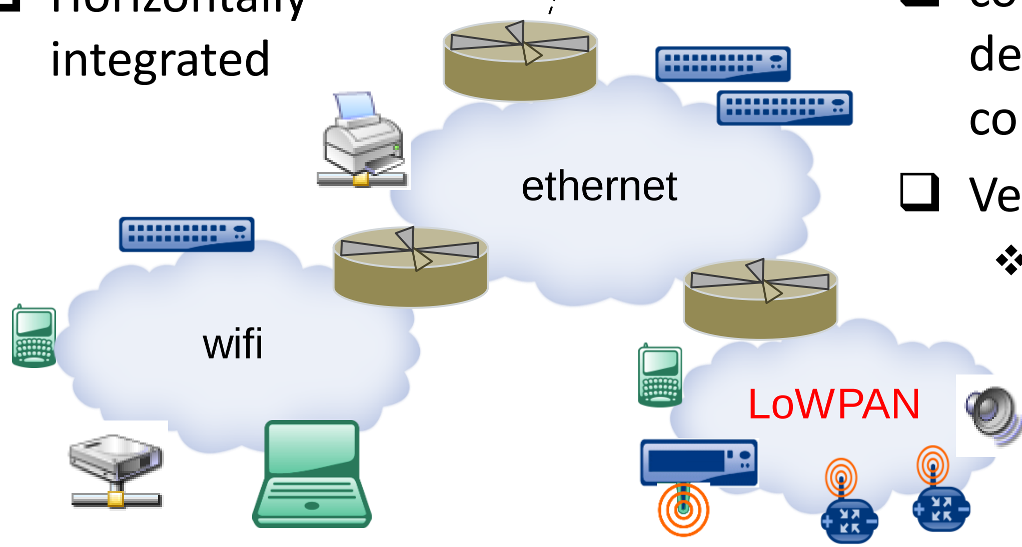


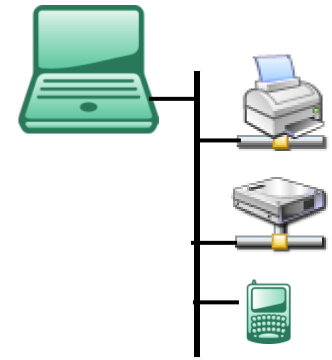
Internet – Networks of Networks

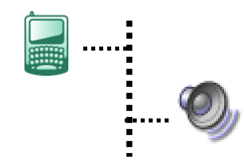
- Networks
  - Ethernet
  - WiFi
  - Serial links
- connect hosts and devices and other networks together
- Horizontally integrated

vs

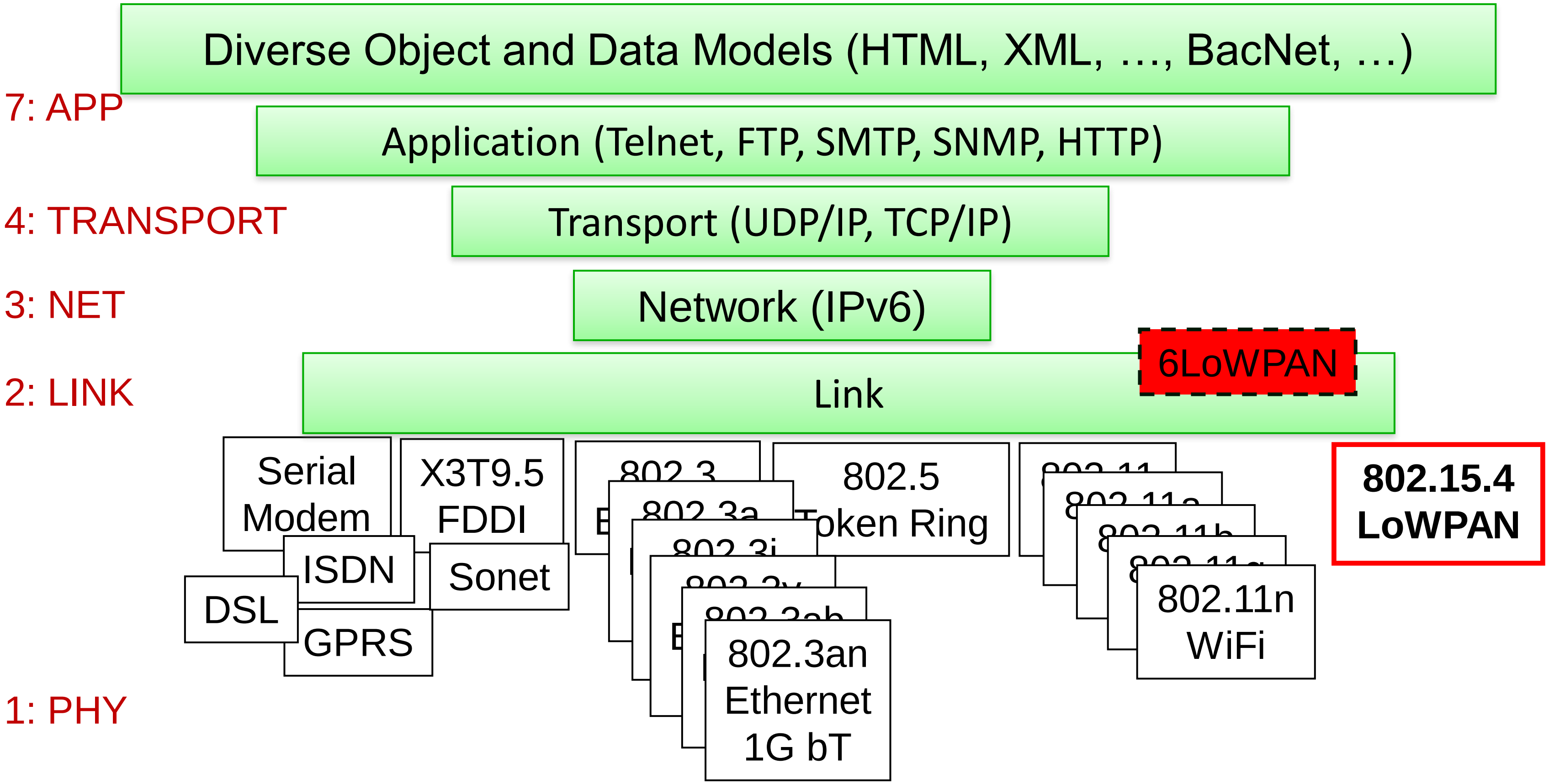
- Peripheral Interconnects
  - USB, Firewire
  - IDE / SCSI
  - RS232,RS485
  - IRDA
  - BlueTooth
- connect one or more devices to a host computer
- Vertically integrated
  - Physical link to application







Introducing IEEE 802.15.4 into the IP family

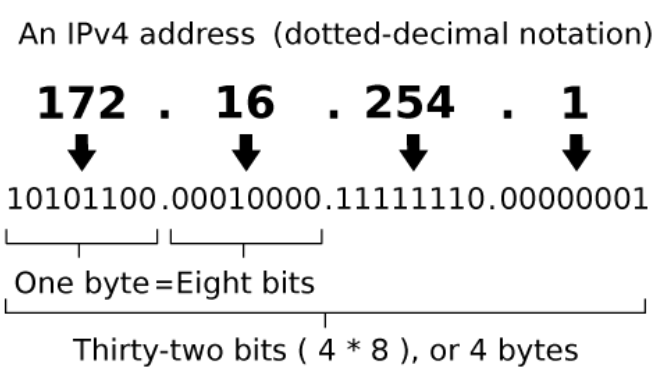


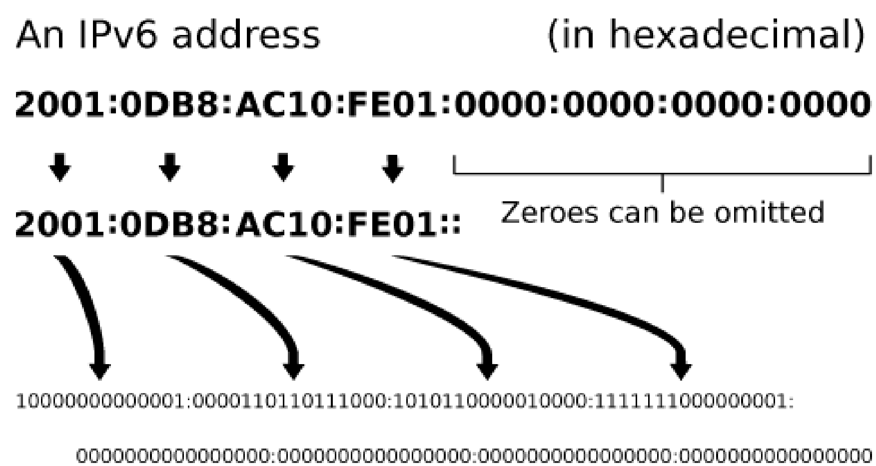
Many Advantages of IP

- Extensive interoperability
  - Other wireless embedded 802.15.4 network devices
  - Devices on any other IP network link (WiFi, Ethernet, GPRS, Serial lines, …)
- Established security
  - Authentication, access control, and firewall mechanisms
  - Network design and policy determines access, not the technology
- Established naming, addressing, translation, lookup, discovery
- Established proxy architectures for higher-level services
  - NAT, load balancing, caching, mobility
- Established application level data model and services
  - HTTP/HTML/XML/SOAP/REST, Application profiles
- Established network management tools
  - Ping, Traceroute, SNMP, … OpenView, NetManager, Ganglia, …
- Transport protocols
  - End-to-end reliability in addition to link reliability
- • Most “industrial” (wired and wireless) standards support an IP option

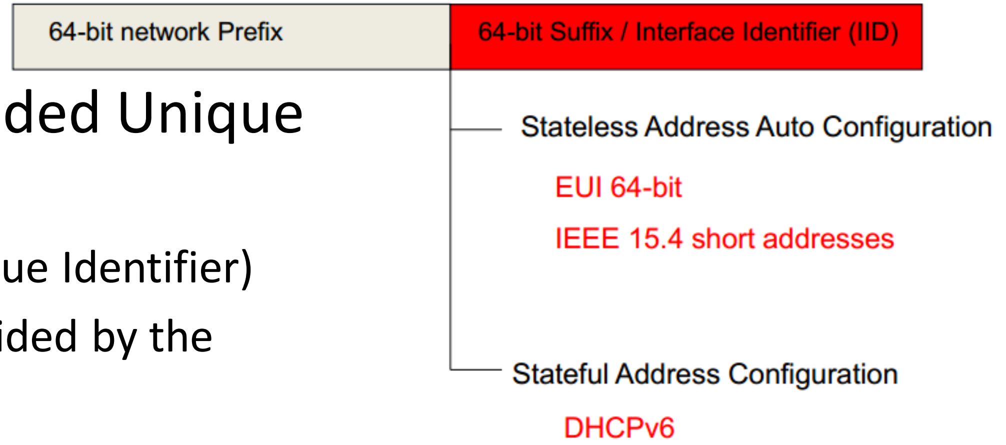
IPv4 and IPv6 addressing

- 128-bit IPv6 address : each 16 bit values (four hex digits) separated by colons, one sequence of all-zero 16-bit values can be replaced by a double colon indicating a longer sequence of zeros.
- Unicast, Anycast, Multicast address
- Unicast address:
  - Global Unicast
  - Local Unicast addresses: involves IP adaptation





IPv6 Address configuration



- 64-bit IEEE EUI-64 bit (Extended Unique Identifier)
  - 24-bit OUI (Organizationally Unique Identifier)
  - 40-bit of extension identifier decided by the manufacturer

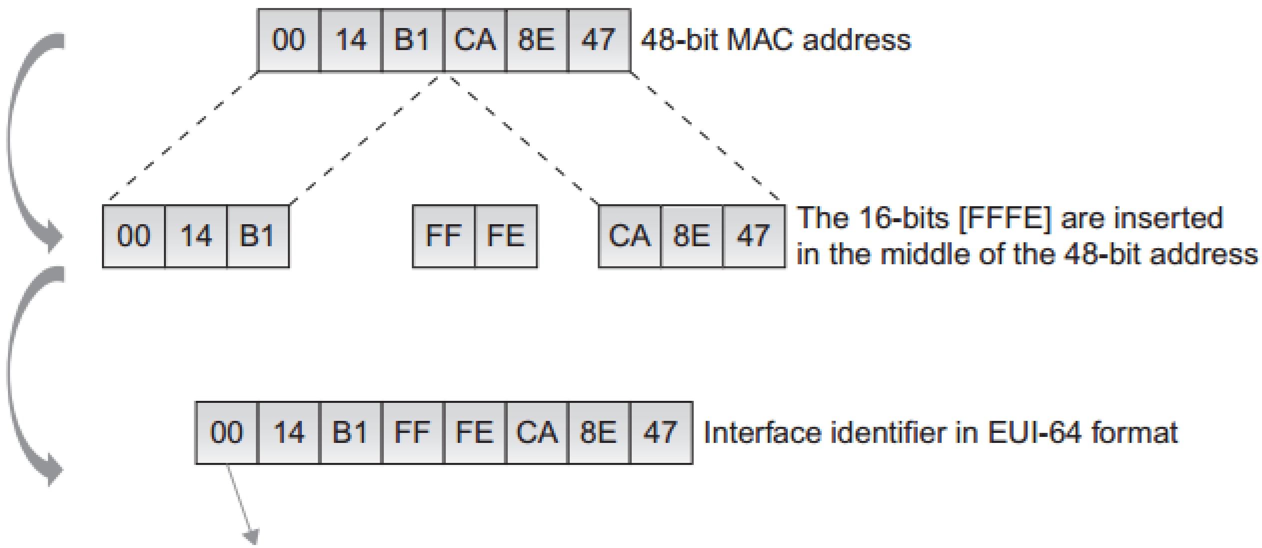
```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     OUI     |L|M|           OUI (cont.)         |               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               +
|                     extension identifier                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                                 |
+                             Prefix                              +
|                                                                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     OUI     |U|M|           OUI (cont.)         |               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               +
|                     extension identifier                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

***M**: distinguish multicast addresses from unicast ones*
***L**: distinguish locally assigned addresses from universal addresses assigned globally*
***U**: inverted **L** bit*

IPv6 Address configuration

- 64-bit IID based on MAC address:
  - MAC address: **00:14:B1:CA:8E:47** converts to 64-bit EUI address



  - **IID**: 00:1D:BA:**FF:FE**:06:37:64 - Based on MAC address
  - Network Prefix: **2001:db8:1:2::/64**
  - Full IPv6 Address : 2001:db8:1:2:**02**1D:BAFF:FE06:3764

IPv6 Address configuration

❑ 64-bit IID based on IEEE 802.15.4 short address:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      PAN Id 0 entifier         |   0   |   0   |   F   |   F   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   F   |   E   |   0   |   0   |     16-bit short address      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

❑ IPv6 Addresses in Hexadecimal Notation

Long form	Abbreviated form	Explanation
2001:DB8:0.0:8:800:200C.417A	2001:DB8::8:800:200C:417A	A unicast address
FF01:0:0:0:0:0:0:101	FF01::101	A multicast address
0:0:0:0:0:0:0:1	::1	Loopback address
FE80:0:0:0:	FE80::	A Link Local address

IPv6 Header

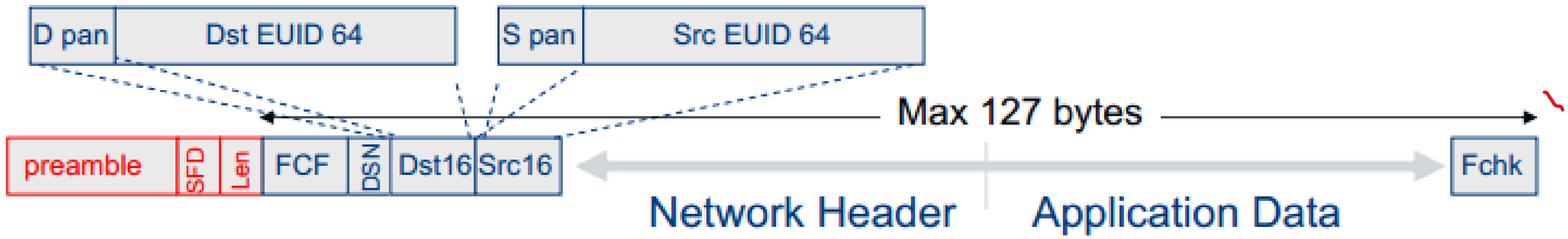
```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Version| Traffic Class |           Flow Label                  | IPv6
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |
|         Payload Length        |  Next Header  |   Hop Limit   |  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |
|                                                               |  |
+                                                               +  |
|                                                               |  |
+                         Source Address                        +  |
|                                                               |  |
+                                                               +  |
|                                                               |  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |
|                                                               |  |
+                                                               +  |
|                                                               |  |
+                      Destination Address                      +  |
|                                                               |  |
+                                                               +  |
|                                                               |  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Source Port          |       Destination Port        | UDP
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+  |
|            Length             |           Checksum            |  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

6 bits: DSCP
2 bits: ECN

Support future services: mobility, security

- The packet header in IPv6 is simpler than that used in IPv4.
  - IPv6 routers do not perform fragmentation
  - The IPv6 header is not protected by a checksum.
  - IPv6 routers do not need to recompute a checksum when header fields (such as the hop limit change)
  - The TTL field of IPv4 has been renamed to Hop Limit

IEEE 802.15.4 Frame Format



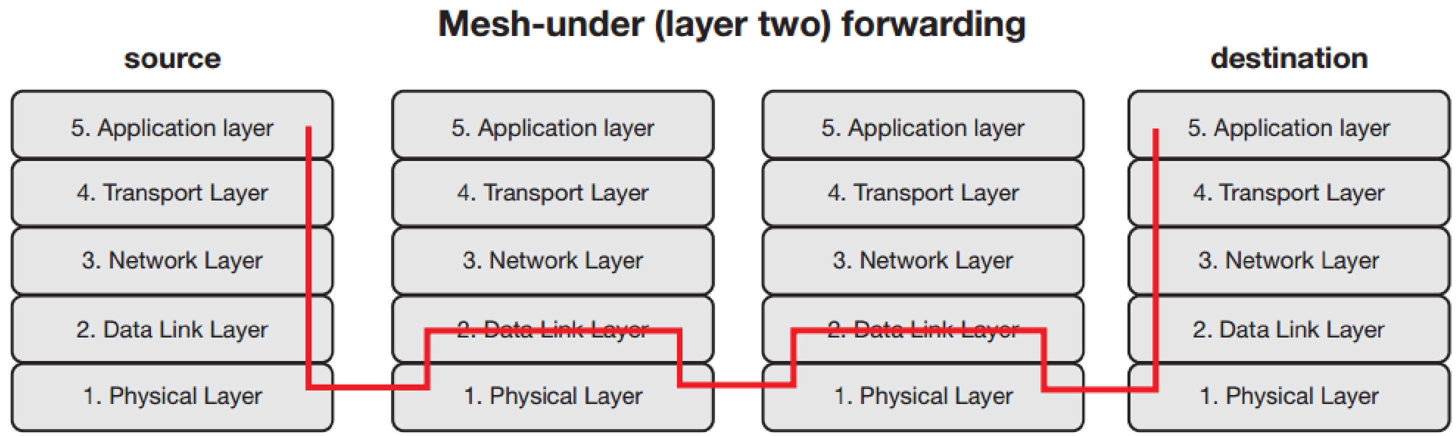
- ❑ Low Bandwidth (250 kbps), low power (1 mW) radio
- ❑ Moderately spread spectrum (QPSK) provides robustness
- ❑ Simple MAC allows for general use
  - ❖ Many TinyOS-based protocols (MintRoute, LQI, BVR), TinyAODV, Zigbee,
  - ❖ SP100.11, Wireless HART, …
  - ❖ 6LoWPAN => IP
- ❑ Choice among many semiconductor suppliers
- ❑ Small Packets to keep packet error rate low and permit media sharing

Key Factors for IP over 802.15.4

- Header
  - Standard IPv6 header is 40 bytes [RFC 2460]
  - Entire 802.15.4 MTU is 127 bytes [IEEE]
  - Often data payload is small
- Fragmentation
  - Interoperability means that applications need not know the constraints of physical links that might carry their packets
  - IP packets may be large, compared to 802.15.4 max frame size
  - IPv6 requires all links support 1280 byte packets [RFC 2460]
- Allow link-layer ***mesh routing*** under IP topology
  - 802.15.4 subnets may utilize multiple radio hops per IP hop
  - Similar to LAN switching within IP routing domain in Ethernet
- Allow IP ***routing over*** a mesh of 802.15.4 nodes
  - Options and capabilities already well-defines
  - Various protocols to establish routing tables
- Energy calculations and 6LoWPAN impact

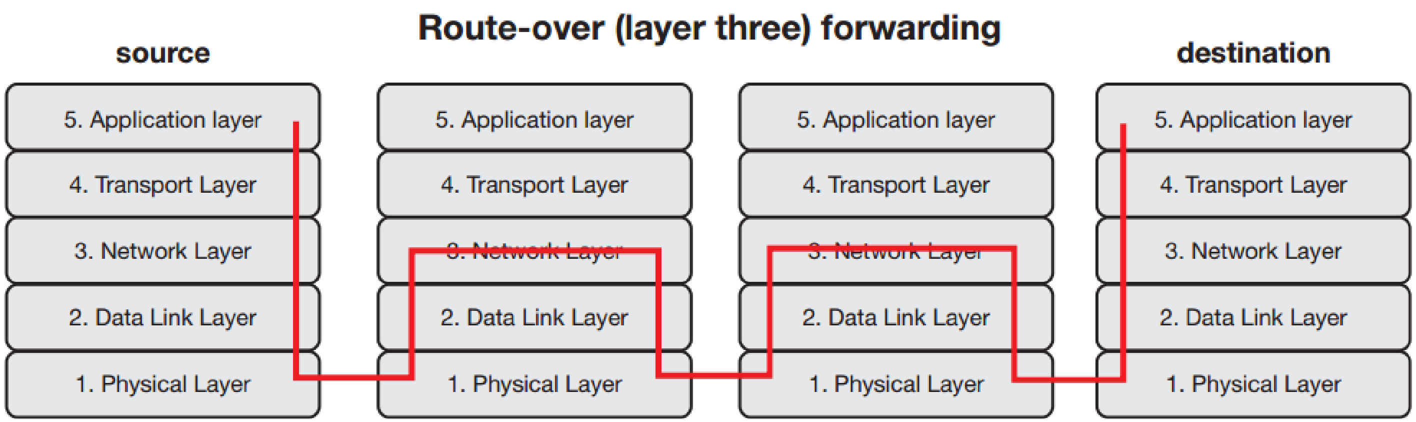
Forwarding and Routing

- Layer 2 Forwarding (**Mesh-Under**):
  - All nodes are on the same link: Single hop
  - One router: 6LBR acts as the IPv6 router
  - Performs forwarding based on L2 addresses (64 bit EUI or 16 bit short addresses)
  - Invisible to 6LoWPAN layer (e.g. ISA 100 defines a mesh routing protocol): DLL mesh forwarding below 6LoWPAN layer
  - If actual link-layer forwarding is not hidden from 6LoWPAN layer, nodes need to know *Originator* and *Final Destination Address*: stored in Mesh header. And 6LoWPAN adaptation layer performs mesh forwarding



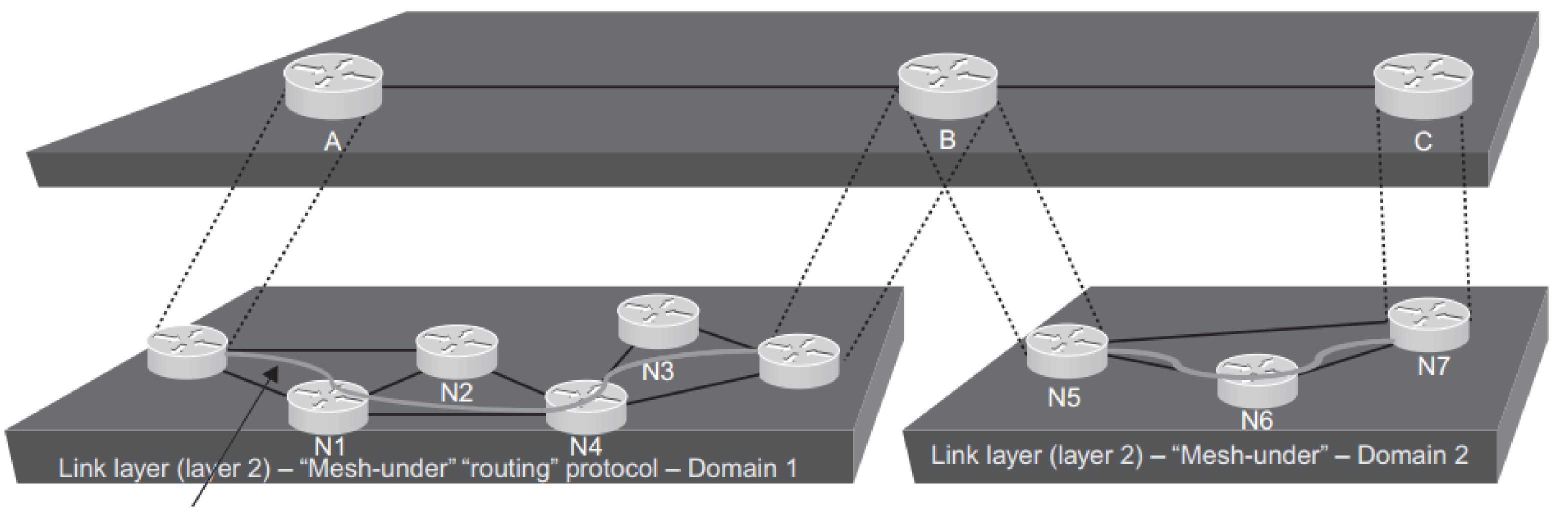
Forwarding and Routing

- Layer 3 Routing (**Route-Over**):
  - Multiple links in an 6LoWPAN: Multiple hop
  - Two types of routers: 6LBR and 6LR
  - Allows routing across multiple link type
  - Does not require any special support from 6LoWPAN adaptation layer.
  - Fragmentation and reassembly are performed at each hop.
  - Routing based on IPv6



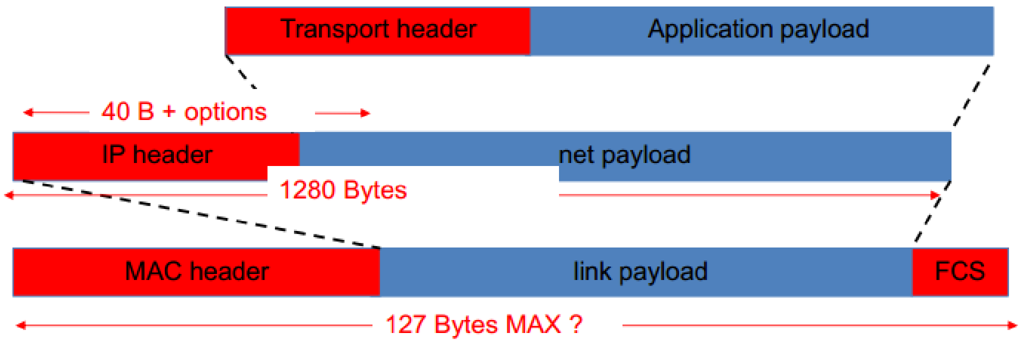
Mesh-Under vs Route-Over

- A-N1-N4-N3-B is the link layer path computed by the "mesh-under" routing protocol operating at the link layer in domain 1.
- At the IP layers: Nodes perform IP routing function and do not "see" the nodes at the link layer. A and B have no visibility on the link layer topology. E.g., nodes N1, N2, N3, and N4 in this example.



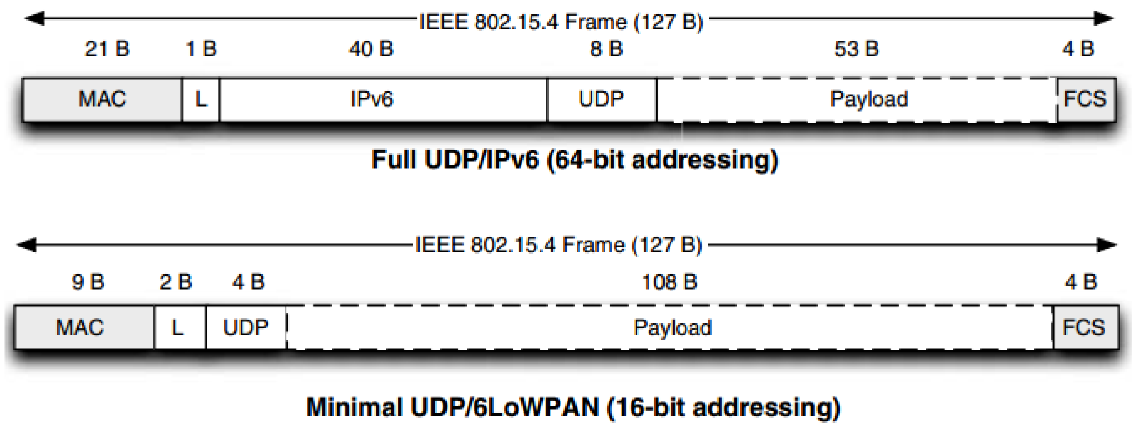
IP over 802.15.4

- **6LoWPAN** : IPv6 over Low-power Wireless Personal Area Networks
- Defined by IETF standards (draft-ietf-6lowpan-hc)
  - RFC 4944
  - RFC 6282
- Header compression
  - Stateless
  - Context based
- Neighbor discovery
  - draft-ietf-6lowpan-nd (RFC 6775)
- Minimal use of code and memory
- Direct end-to-end Internet integration

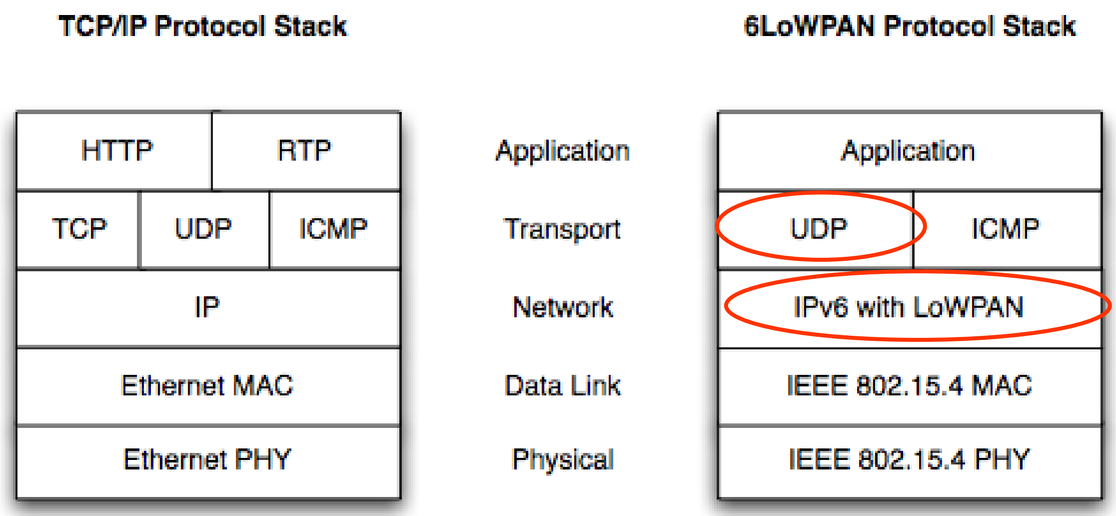


Protocol Stack

- 6LoWPAN is an adaptation header format
  - Enables the use of IPv6 over low-power wireless links
  - IPv6 header compression
  - UDP header compression

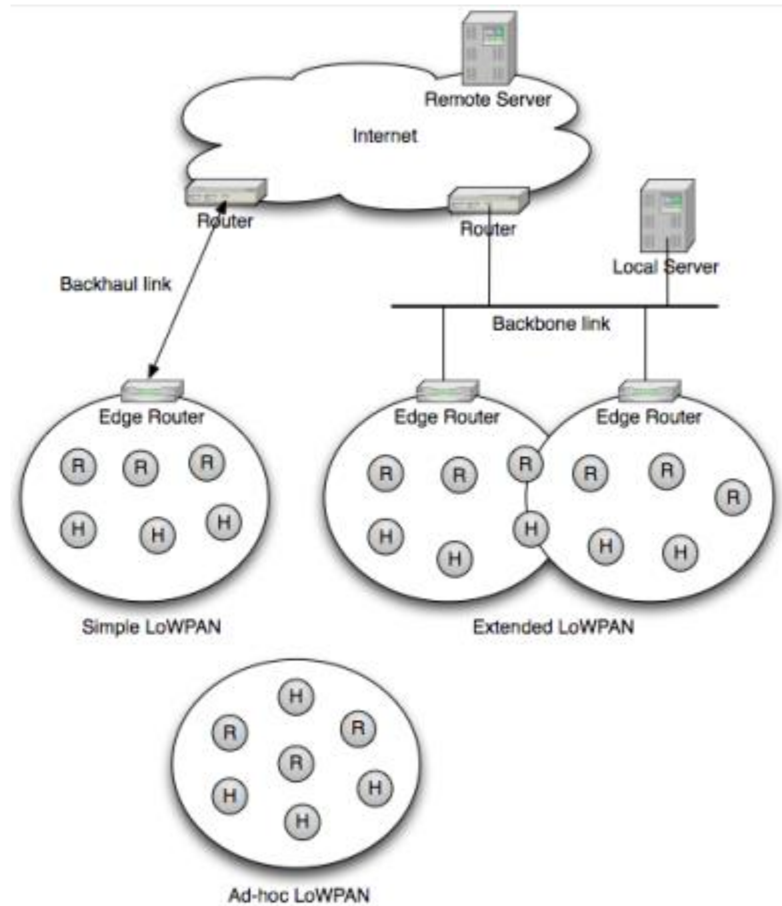


L: LoWPAN Header



6LoWPAN Architecture

- ❑ LoWPANs are stub networks
  - ❖ Nodes (Host/Router)
- ❑ Simple LoWPAN
  - ❖ A collection of nodes which shares a common IPv6 address prefix
  - ❖ Single Edge Router (handles compressions & ND, IPv4 interconnectivity)
- ❑ Extended LoWPAN
  - ❖ Multiple Edge Routers with common backbone link
- ❑ Ad-hoc LoWPAN
  - ❖ Not connected to the Internet

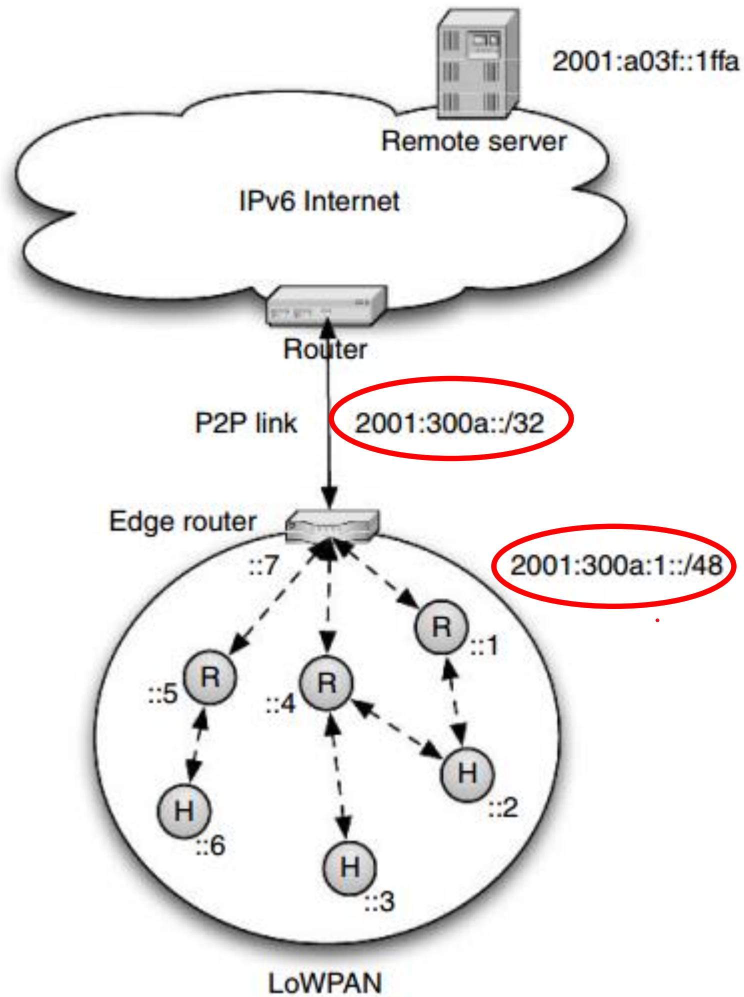


WSNs: Addressing

- IPv6 addresses are typically formed using
  - Network Prefix: Prefix of the LoWPAN
  - IID: Subfix - Link Layer address of the wireless interface (64bit EUI / 16 bit short address)
- Flat LL addressing for all the devices
  - Support unique long addresses (EUI-64)
  - Configurable short addresses (usually 8-16 bits in length)
- Also support broadcast (0xFFFF in IEEE 802.15.4), but do not support native multicast

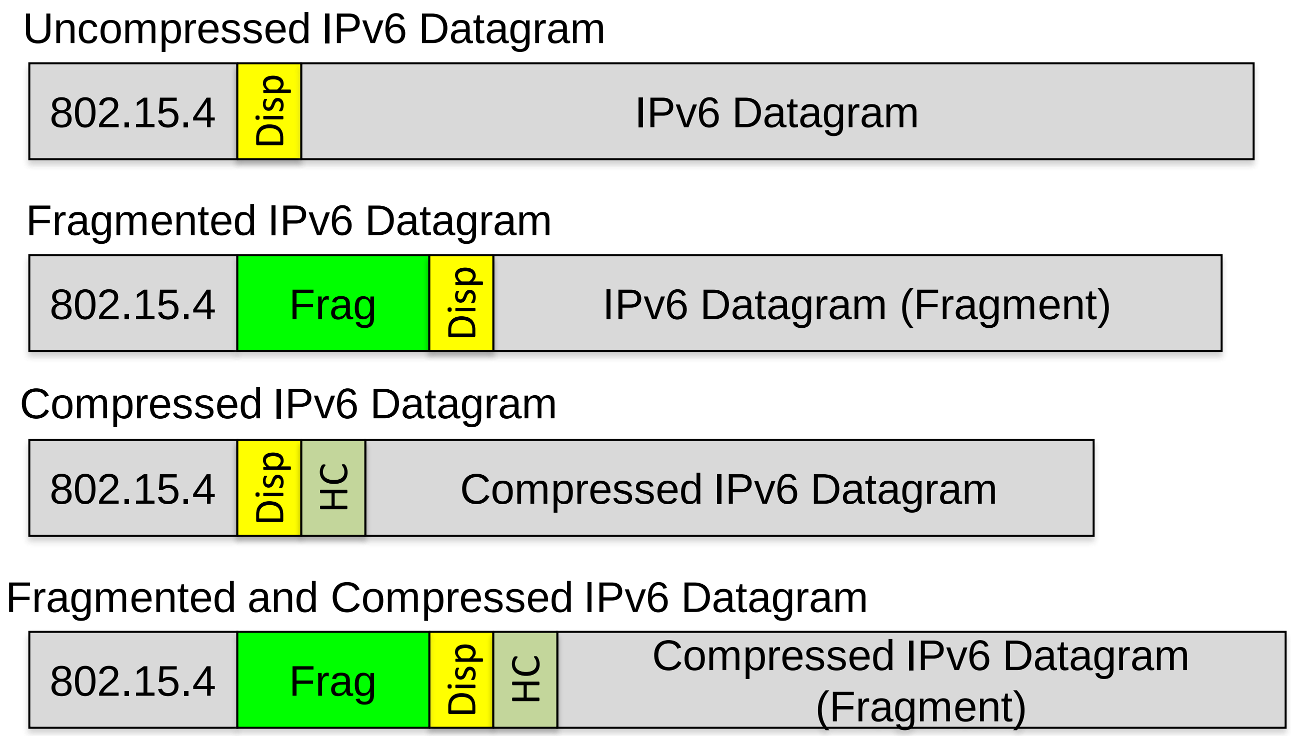
Addressing : Example 6LoWPAN

- Assumption: LoWPAN uses IEEE 802.15.4 and IP routing
- Edge Router
  - Configures IPv6 prefix to its 802.15.4 wireless interface (**2001:300a:1::/48**)
  - Advertizes IPv6 prefix to nodes in the LoWPAN
- LoWPAN node
  - Get IPv6 address with a 64-bit IID (**subfix**)
  - Receives generated IPv6 address with a 16 bit IID (HOST :: 1 -> 2001:300a:1::1)
- Routing
  - Packets sent within LoWPAN do not require to have IPv6 network prefix inline
    - E.g. from **::6** -> **::5**: Link layer header contains 802.15.4 Source and Destination addresses
    - E.g. from **::3** -> **::7**: require 16-bit IID: used for routing the packet.
  - Packets destined outside LoWPAN -> require to have full IPv6 address inline



RFC4944 Header Format

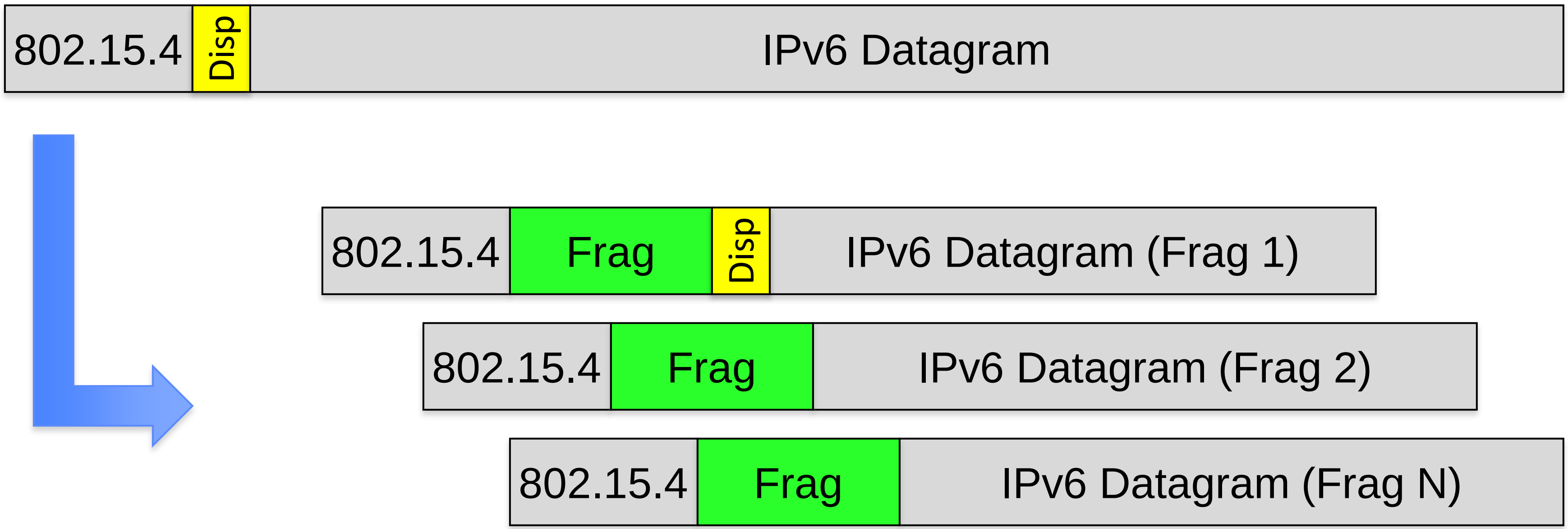
- Header chaining (borrowed from IPv6)
  - Compact, simple, flexible
  - Only include functionality as needed
- Each header includes header type (dispatch)



RFC4944 Fragmentation

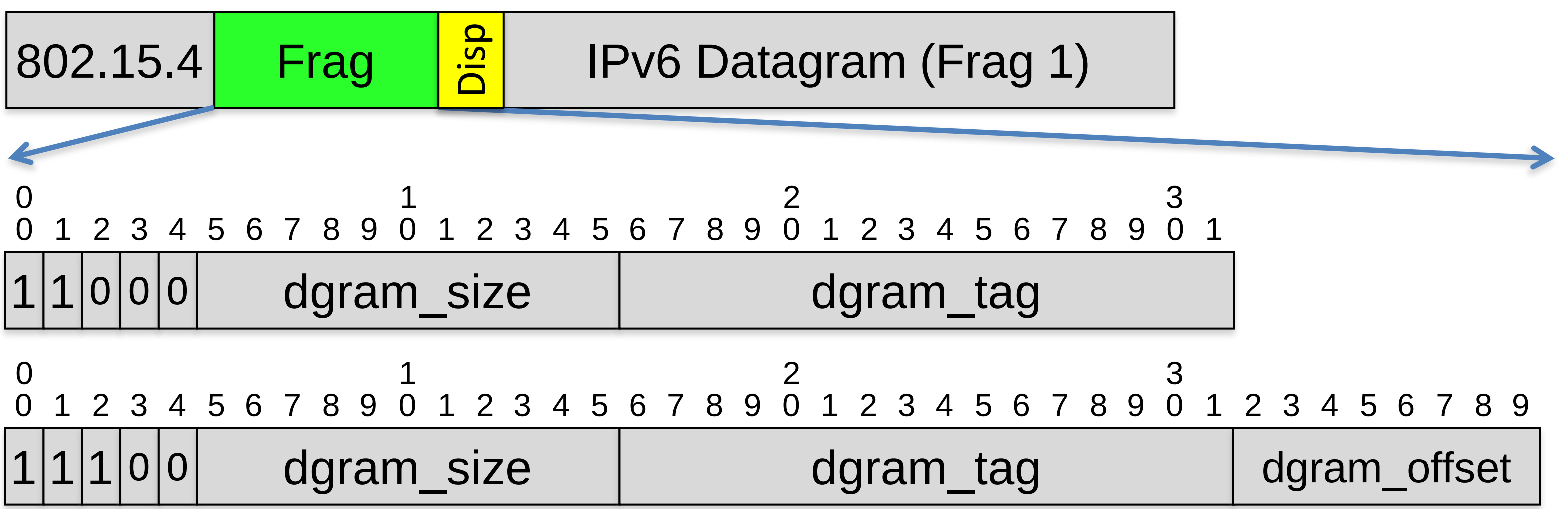
- ❑ 802.15.4-2006 has a link MTU of 127 bytes
- ❑ IPv6 requires a min link MTU of 1280 bytes

→ 6LoWPAN must provide fragmentation



RFC4944 Fragmentation

- ❑ Size: size of datagram in bytes
  - ❖ Included in all fragments to simplify buffer allocation
- ❑ Tag: identifies all fragments of a datagram
- ❑ Offset: location of fragment in 8-byte units
  - ❖ Elided in first fragment



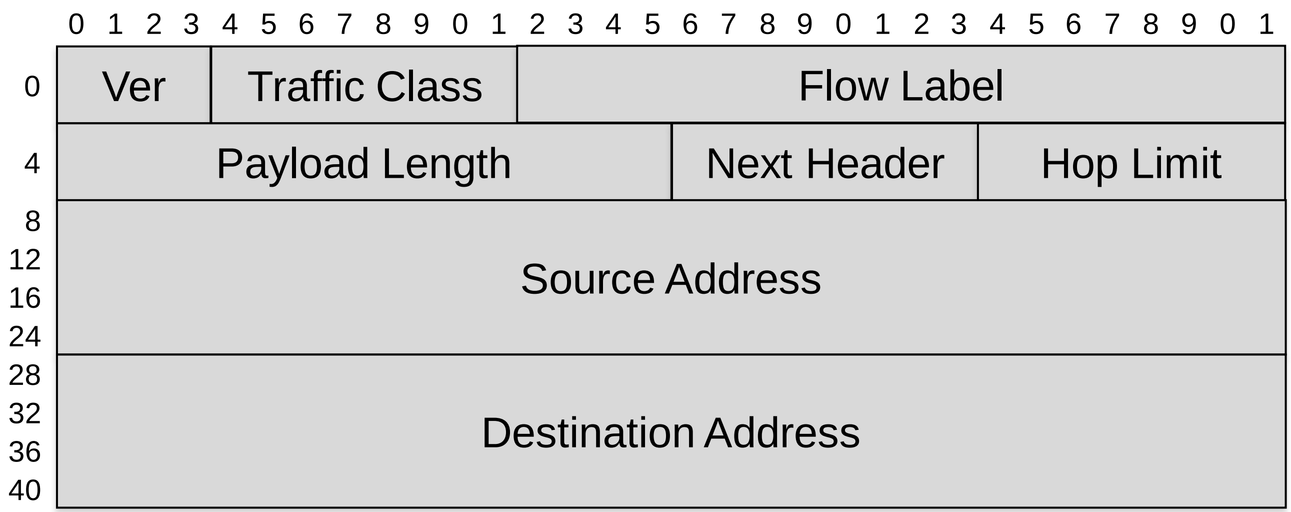
RFC4944 Header Compression

- Traditional flow-based methods are not ideal
  - Increased state management
  - Limited flexibility in forwarding via different neighbors
- Use a stateless compression mechanism
  - Applicable to all flows with any neighbor

RFC4944 Header Compression

- Assume common values for header fields and define compact forms
  - Version is always 6 (IPv6)
  - Traffic Class and Flow Label are zero
  - Payload Length always derived from L2 header
  - Next Header is UDP, TCP, or ICMPv6
  - Source and Destination Addrs are link-local and derived from L2 addrs

IPv6 Header



RFC4944 Header Compression

- Assume common values for header fields and define compact forms
  - Ports within 61616 to 61632 (4 bits)
  - Length derived from IPv6 Length
  - Checksum always carried inline
- No definition for TCP or ICMPv6

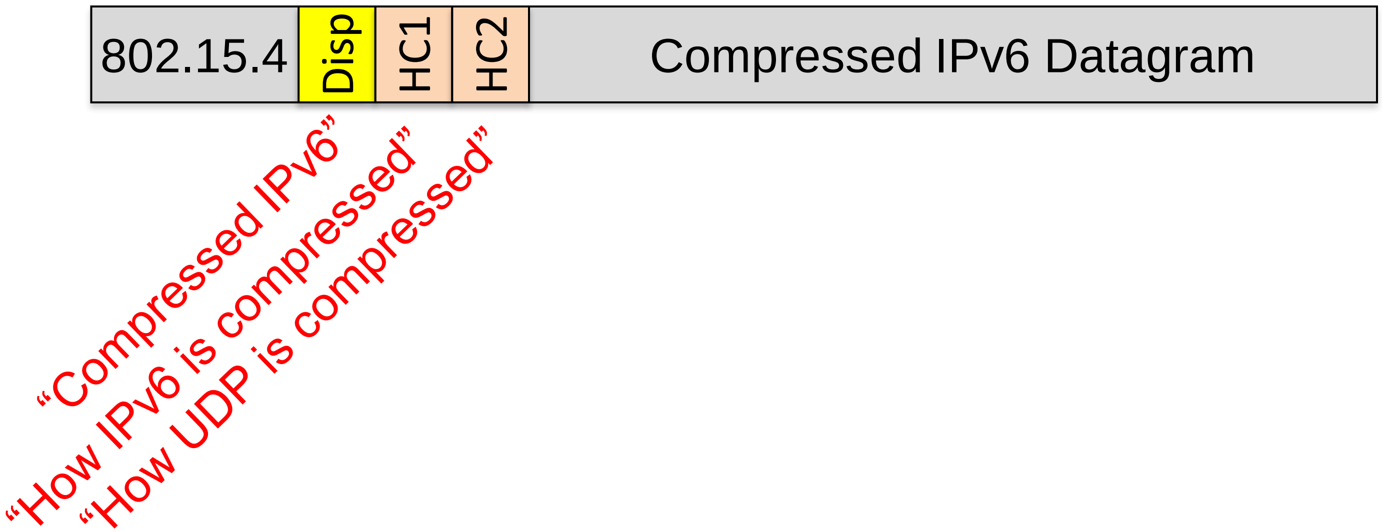
UDP Header

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	Source Port	Destination Port
4	Length	Checksum

RFC4944 Header Compression

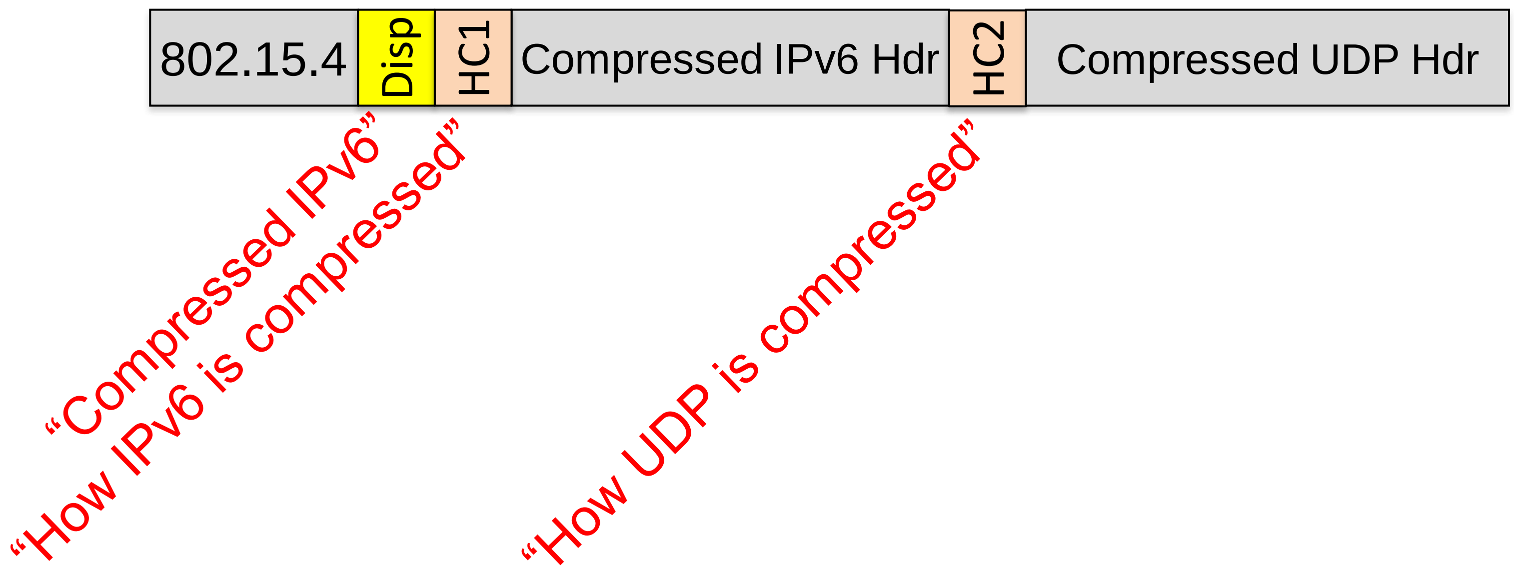
- Insert control bytes to indicate how IPv6 and UDP headers are compressed



Next Header Compression

- ❑ Each compressed header indicates if the next header is also compressed
- ❑ Following control byte(s) include next header identifier

→ Framework for defining arbitrary Next Header compression methods



Example: Link-Local Unicast

Link Hdr

Len = 50	FCF	DSN	DSTPAN
DST = 00-17-3B-00-AA-BB-CC-DD			
SRC = 00-17-3B-00-11-22-33-44			

IPv6 Hdr

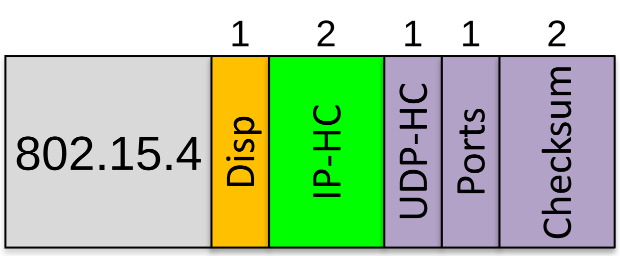
Ver = 6	Traffic Class = 0	Flow Label = 0
Payload Length	Next Header=UDP	Hop Limit = 1
Source Prefix = fe80::/64		
Source IID = 0217:3B00:AABB:CCDD		
Dest Prefix = fe80::/64		
Dest IID = 0217:3B00:1122:3344		

UDP Hdr

Source Port	Destination Port
Length	Checksum

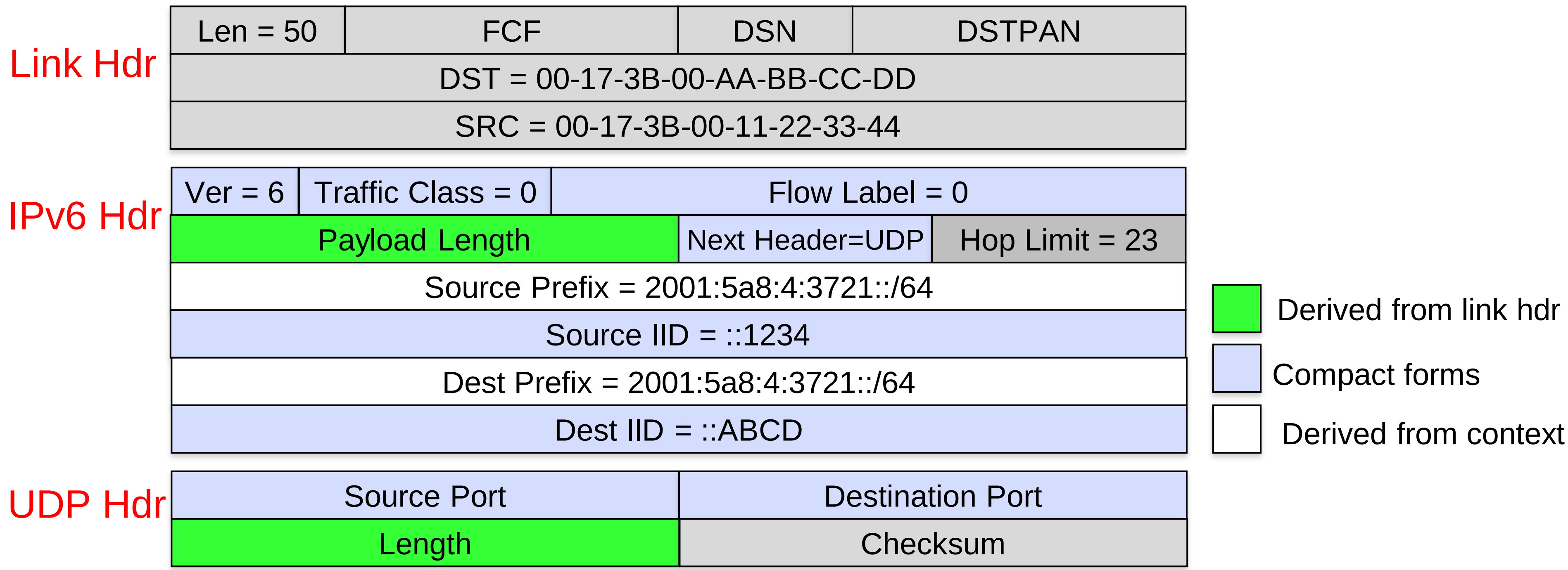
Derived from link hdr

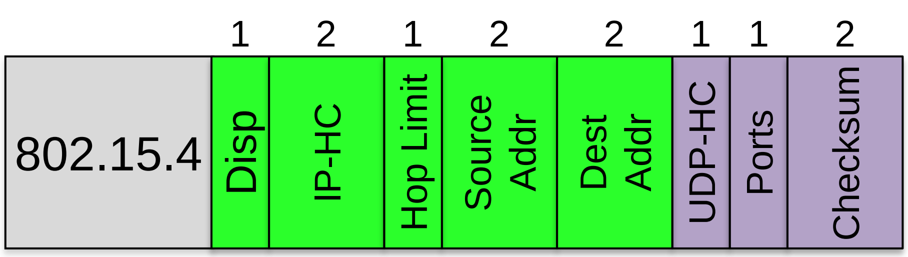
Compact forms



48-byte UDP/IPv6 Hdr → 7 bytes

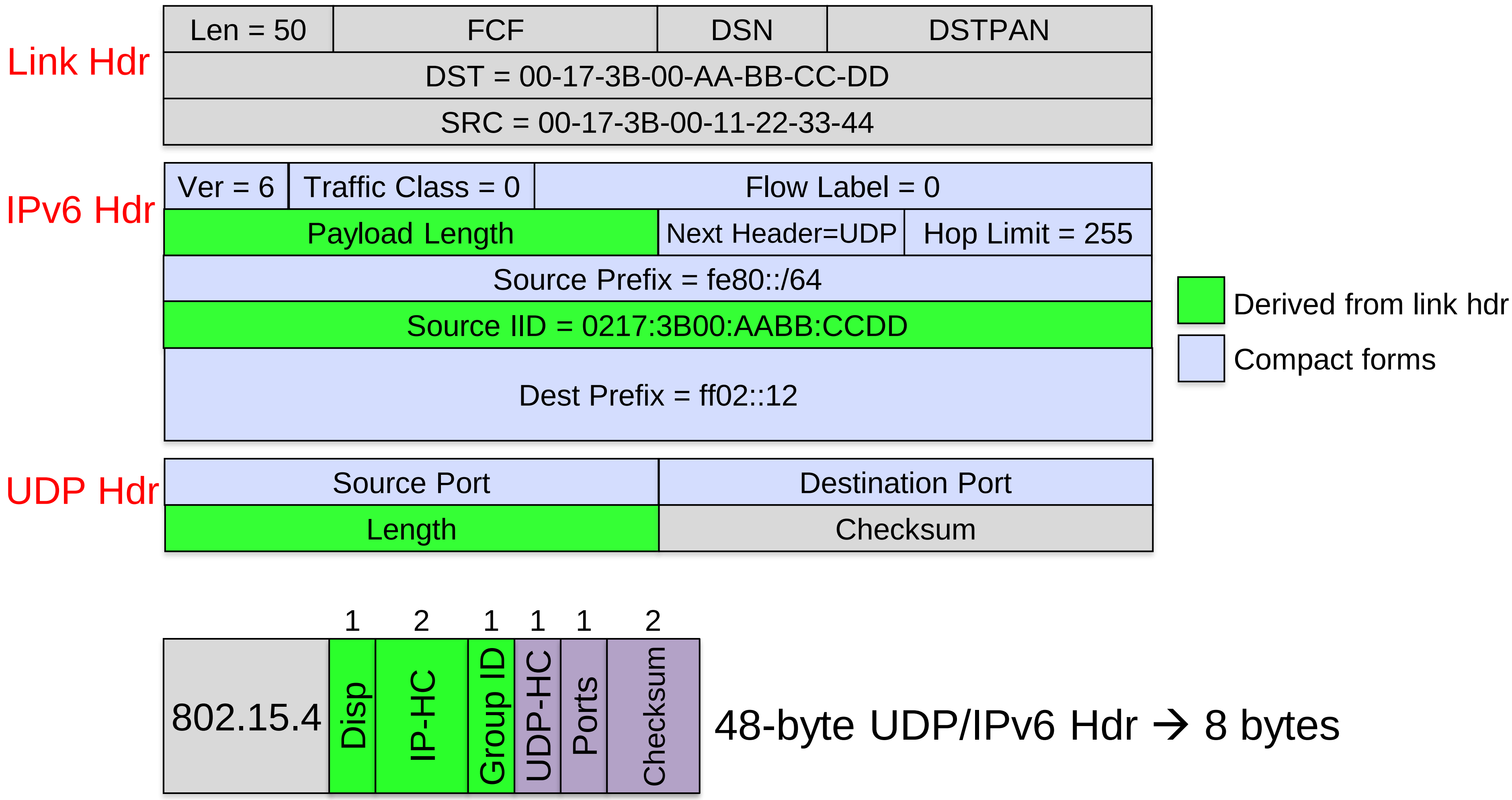
Example: Global Unicast





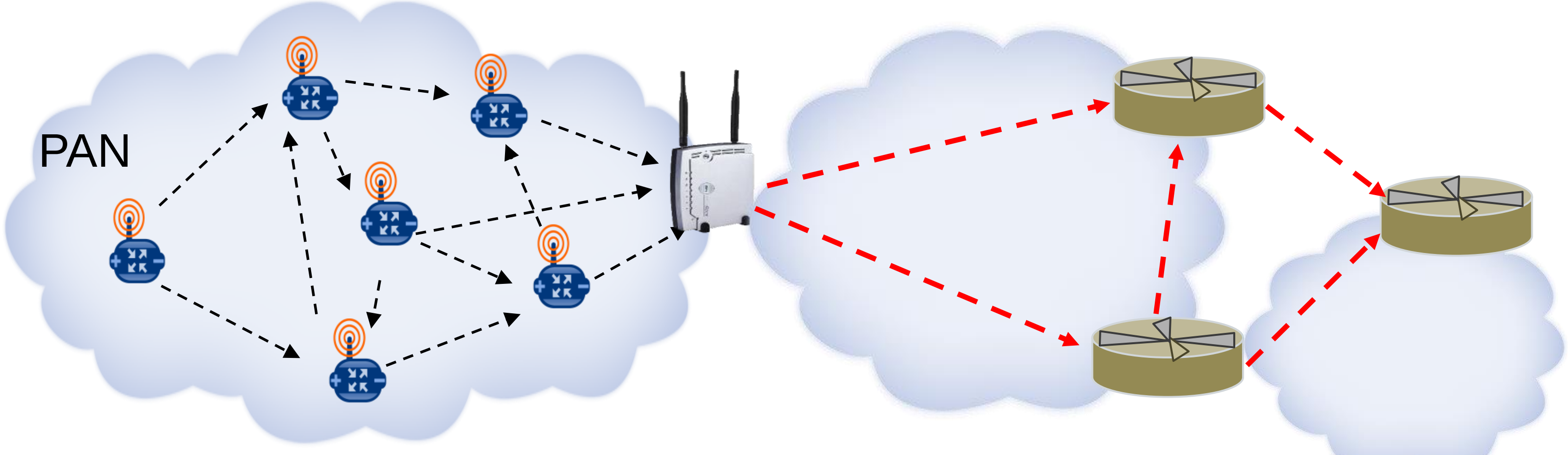
48-byte UDP/IPv6 Hdr → 12 bytes

Example: Link-Local Multicast



Meshing vs Routing

- Short-range radios & Obstructions => Multi-hop Communication is often required:
  - i.e. Routing and Forwarding
  - That is what IP does!
- "Mesh-under": multi-hop communication at the link layer
  - Still needs routing to other links or other PANs
- "Route-over": IP routing within the PAN
- 6LoWPAN supports both



"Mesh Under" Header

- Originating node and Final node specified by either short (16 bit) or EUID (64 bit) 802.15.4 address
  - In addition to IP source and destination
- Hops Left (up to 14 hops, then add byte)
- Mesh protocol determines node at each mesh hop

LoWPAN mesh header

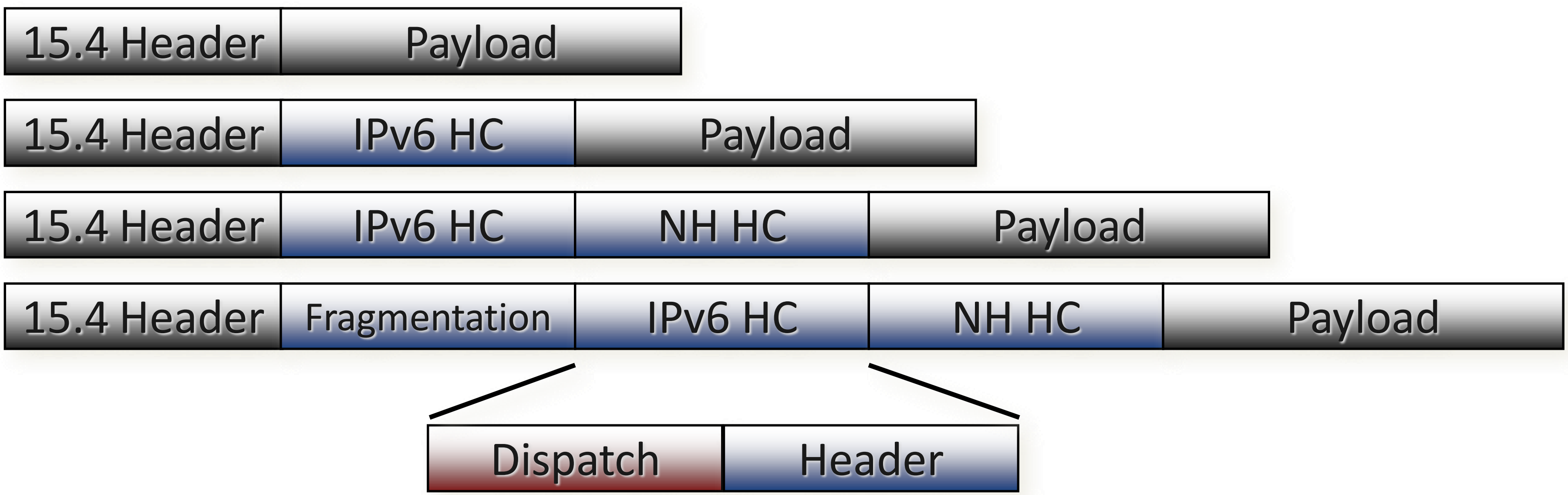
10	o	f	hops left	orig. addr (16/64)	final. addr (16/64)

final 16/64 address

originator 16/64 address

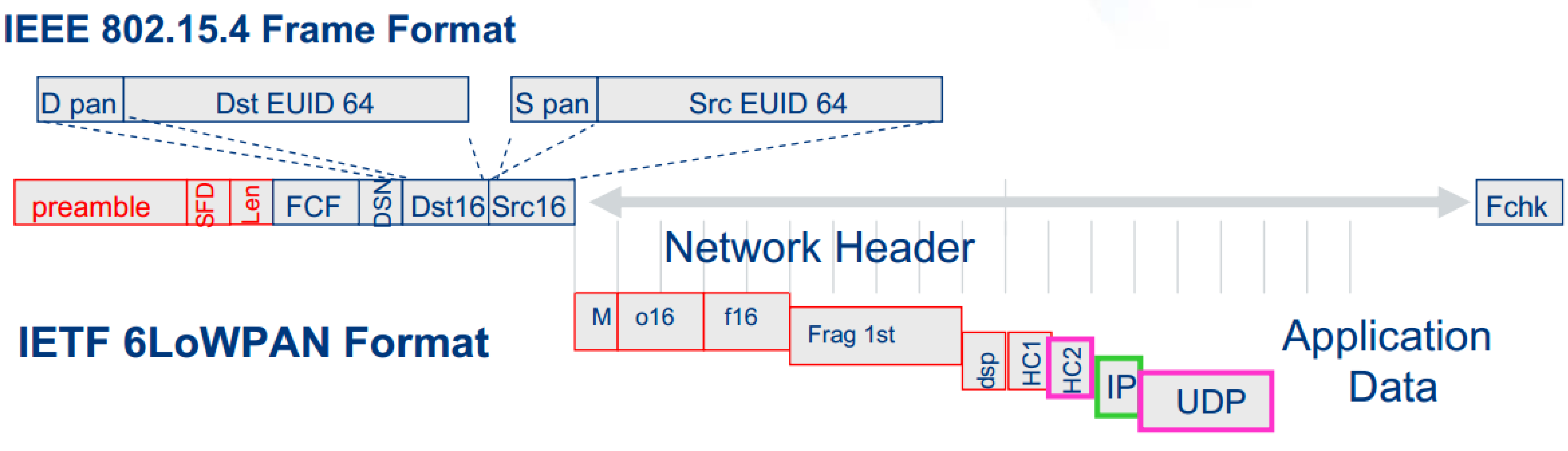
Adaptation Summary
Efficient Transmission of IPv6 Datagrams

6LowPAN Stacked Adaptation Header Format



http://tools.ietf.org/html/rfc4944

6LoWPAN Ex: Mesh/Fragmented/Compressed/UDP



Dispatch: Mesh under, orig short, final short

Mesh: orig addr, final addr

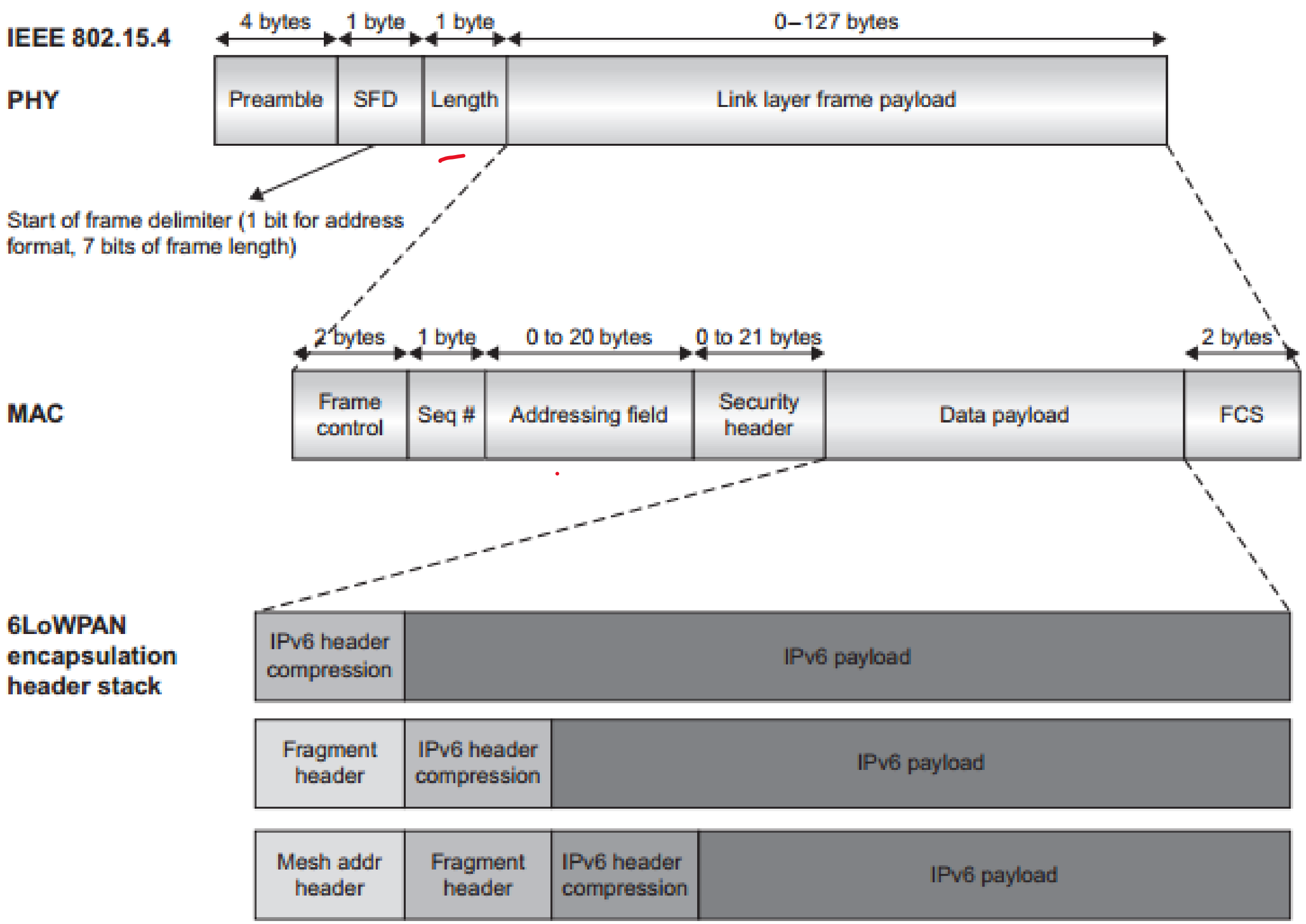
Dispatch: Fragmented, First Fragment, Tag, Size

Dispatch: Compressed IPv6

HC1: Source & Dest Local, next hdr=UDP
IP: Hop limit
UDP: HC2 + 3-byte header

6LoWPAN encapsulation header stack

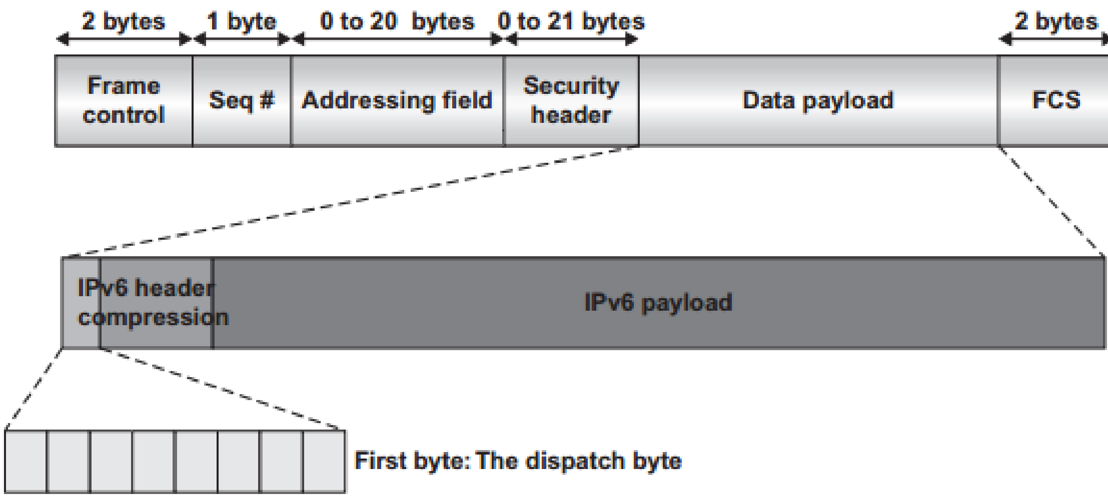
- Support 3 headers: Mesh, Fragmentation and IPv6 header



The first byte: 6LoWPAN Dispatch

- The first byte is used to defined the next header. For example, if the first 2 bits are 11, the next header is a fragmentation header

The 6LoWPAN dispatch byte (first byte)

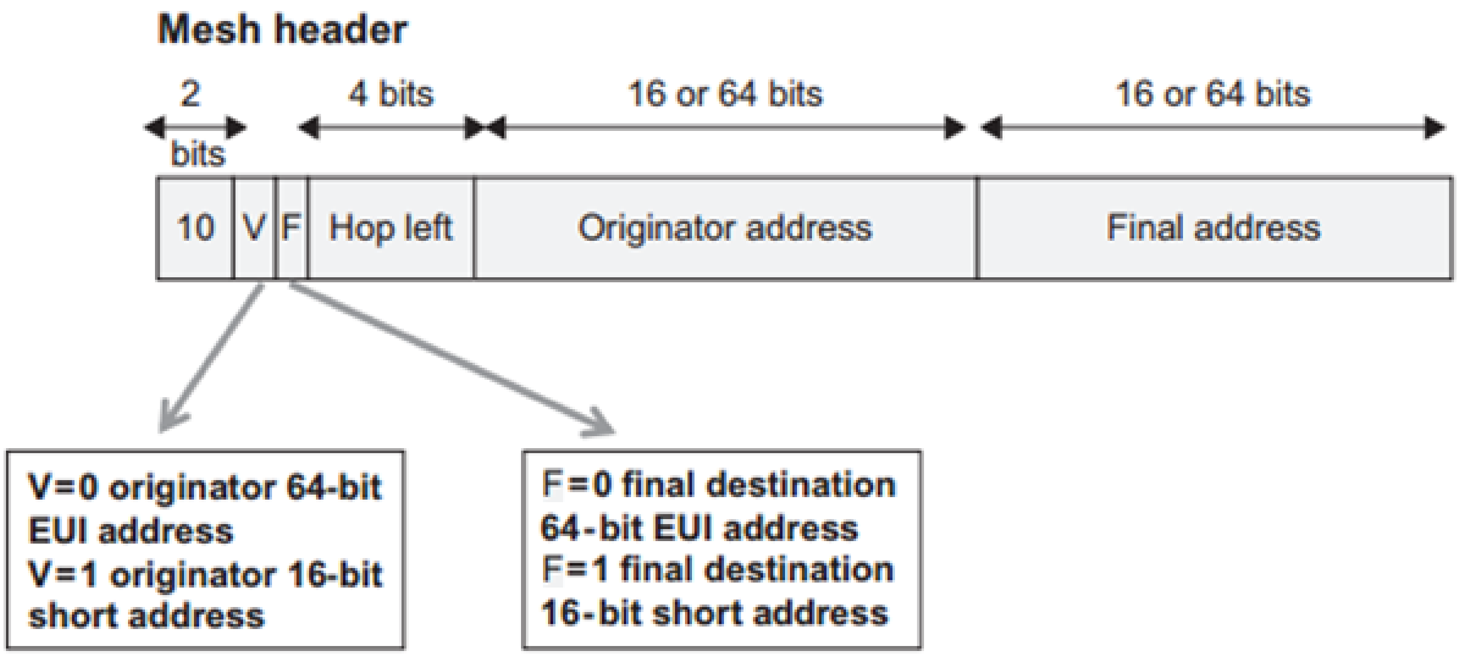


00	**Not a 6LoWPAN frame**
01	IPv6 addressing header
10	Mesh header
11	Fragmentation header (6 lower bits are 100xxx)

Pattern	Header type
00 xxxxxx	NALP - not a LoWPAN frame
01 000001	IPv6 - uncompressed IPv6 addresses
01 000010	LOWPAN_HC1-LOWPAN_HC1 compressed IPv6
01 000011	reserved - reserved for future use
...	reserved - reserved for future use
01 001111	reserved - reserved for future use
01 010000	LOWPAN_BCO - LOWPAN_BCO broadcast
01 010001	reserved - reserved for future use
...	reserved - reserved for future use
01 111110	reserved - reserved for future use
01 111111	ESC - additional dispatch byte follows
10 xxxxxx	MESH - Mesh header
11 000xxx	FRAG1 - fragmentation header (first)
11 001000	reserved - reserved for future use
...	reserved - reserved for future use
11 011111	reserved - reserved for future use
11 100xxx	FRAGN - fragmentation header (subsequent)
11 101000	reserved - reserved for future use
...	reserved - reserved for future use
11 111111	reserved - reserved for future use

Mesh 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 source and destination nodes are then referred to as the *originator* and *final destination*, respectively

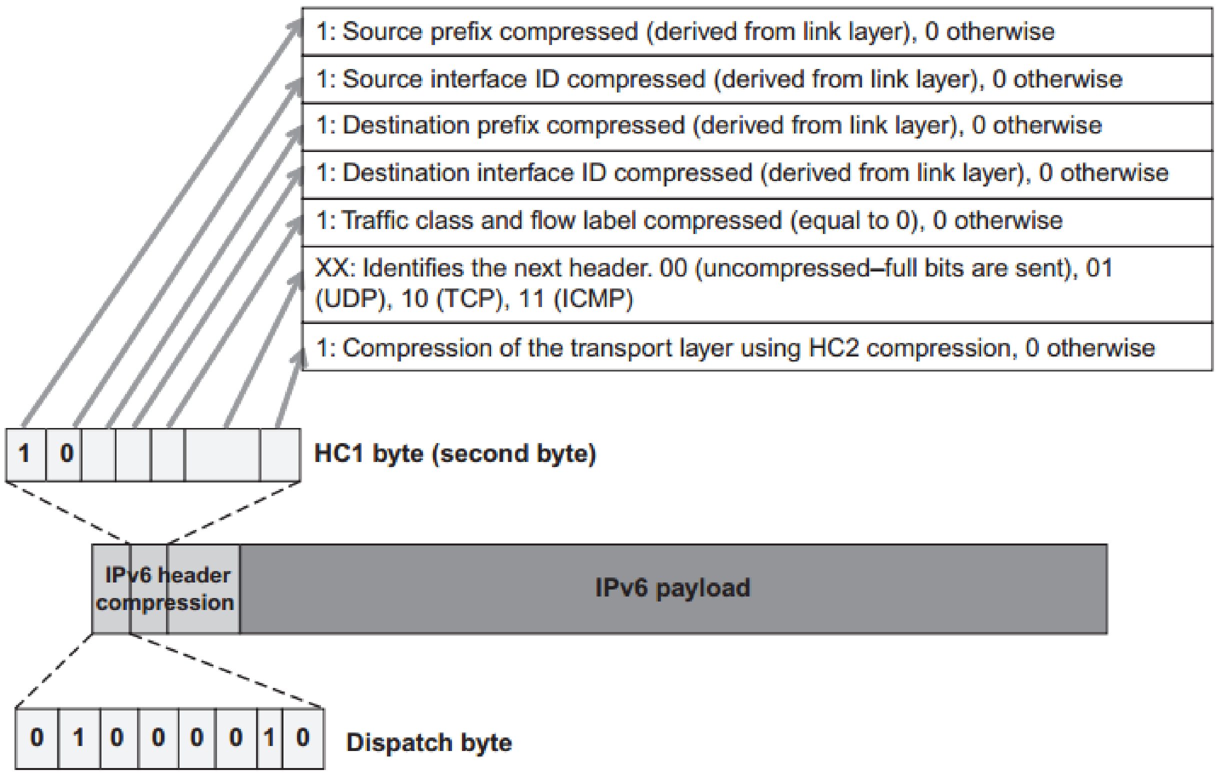


Mesh header

- When a node A sends a frame to a final destination C via the node B:
  - The originator address of the mesh header is set to the link layer address of A.
  - The final destination address of the mesh header is set to the link layer address of C.
  - The source address of the IEEE 802.15.4 frame is the address of the node sending the frame (A).
- The destination address of the IEEE 802.15.4 frame is the link layer address of the next-hop node as determined by the mesh-under routing protocol (B in this example). Upon receiving the frame, B performs the following process:
  - The hop left field is decremented.
  - If the ***hop left*** field is not equal to 0 (if equal to 0, the frame is discarded), then B determines that the next hop is C.
  - The originator and final destination address of the mesh header are unchanged.
  - The source address of the 802.15.4 frame is set to the link layer address of B.
  - The destination address of the 802.15.4 frame is set to the link layer address of C.

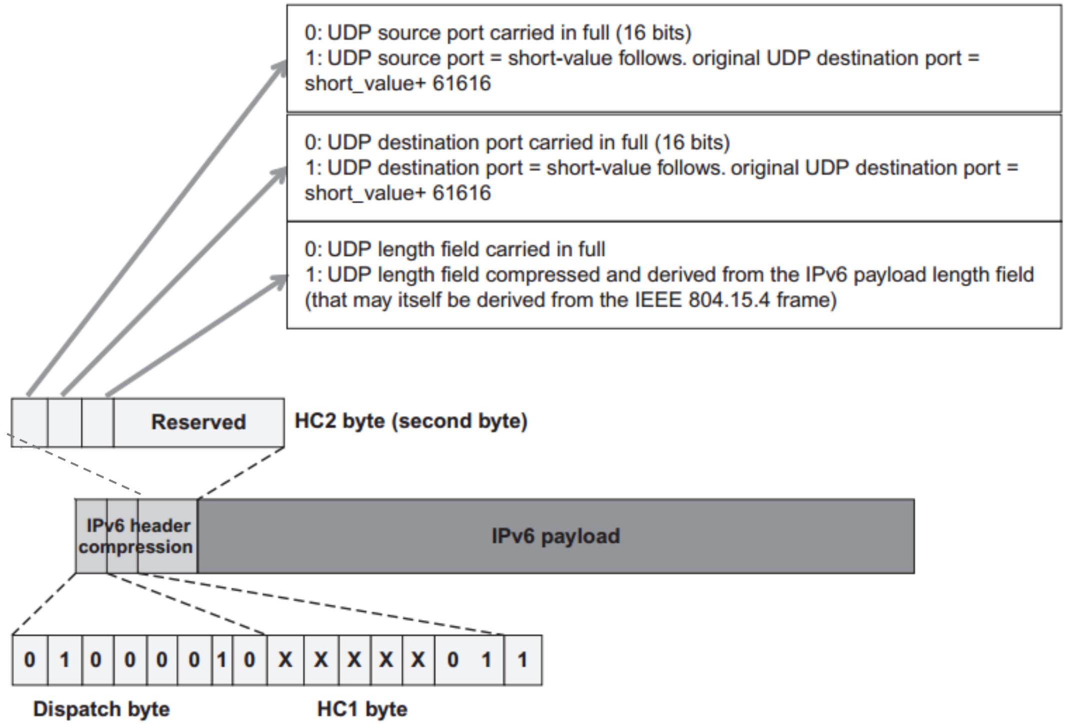
HC1 Compressed IPv6 Header

- The only IPv6 header field that cannot be compressed and must be carried in full is the 1-byte *hop limit* field. This leads to only 3 bytes instead of the 40-byte IPv6 header: 1 byte for the dispatch byte (equal to 01000010), followed by a 1-byte HC1 byte, and 1 byte for the *hop limit* field.



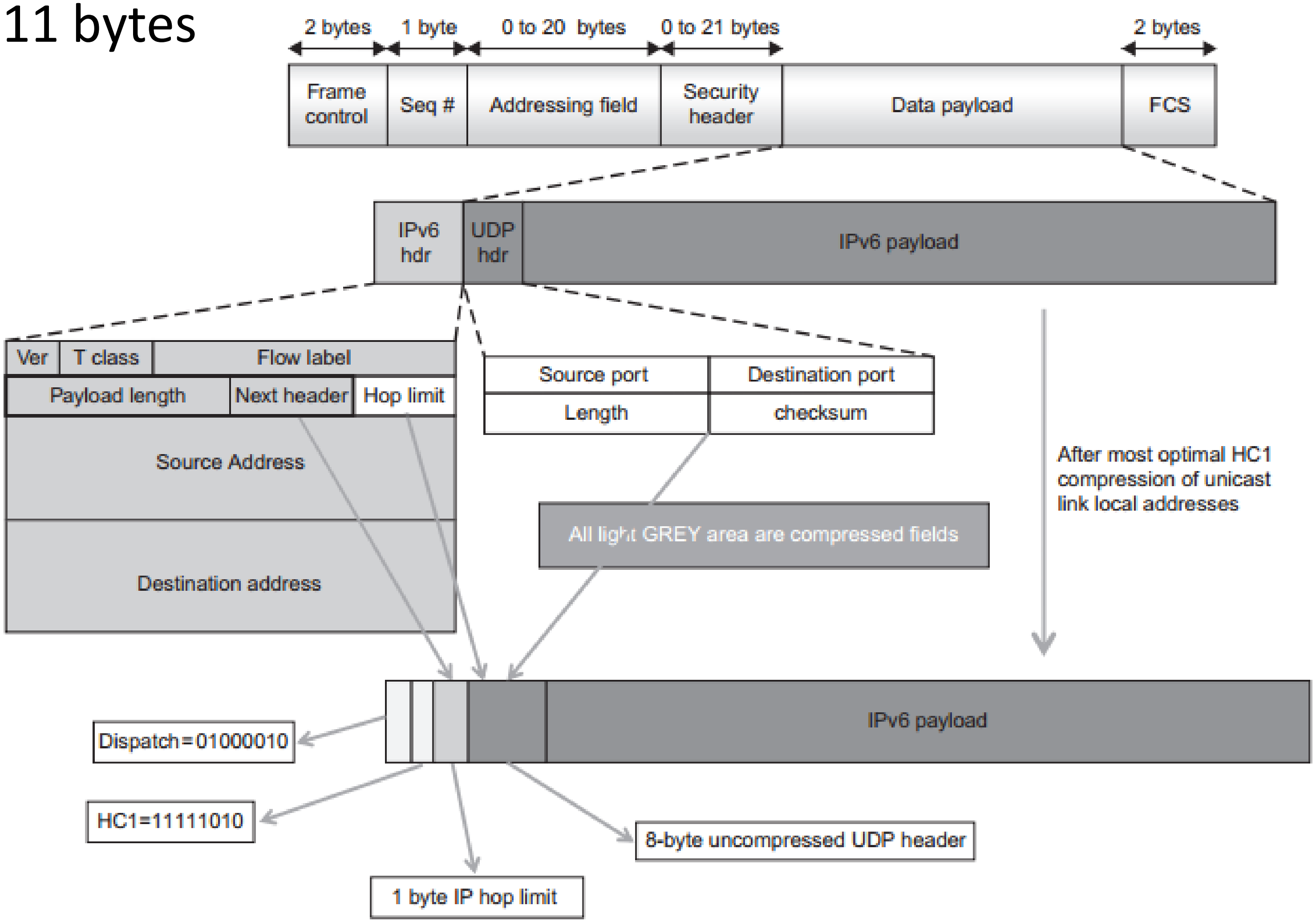
6LoWPAN – Compressed / UDP

- 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).
- HC1 and HC2 header compression allows a very efficient compression technique for reducing the header size from 40 bytes (IPv6 header) + 8 bytes (UDP header) down.



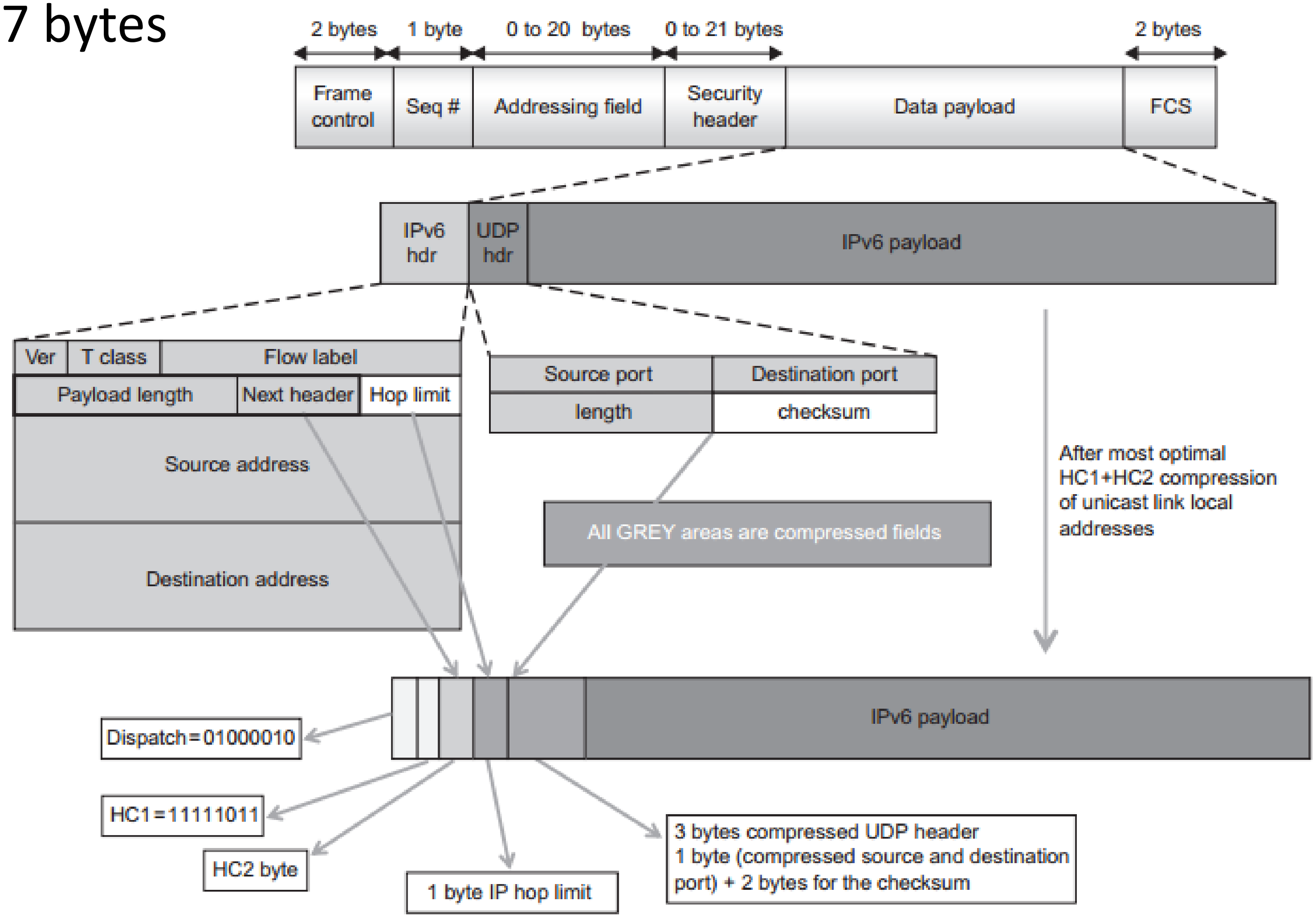
HC1 without UDP header compression

- ❑ Need 11 bytes



HC1+HC2 with UDP header compression

- Need 7 bytes



Is Stateless Header Compression Enough?

- Stateless header compression for IPv6 packets (LOWPAN_HC1 & LOWPAN_HC2) -> reduce relatively large IPv6 and UDP headers down to few bytes
  - Most effective for link local unicast communication
- Limitations of HC1 & HC2
  - Link local communications are mainly used for local protocols (neighbor discovery, DHCPv6, etc..) within a LoWPAN network. Application layer protocols use global IPv6 addresses
    - HC1 should carry 64 bit prefix when using global addresses
    - HC1 should carry 128 bits address when using multicast communication
    - IPv6 next headers cannot be encoded efficiently
- Compression: LOWPAN_IPHC and LOWPAN_NHC

6LoWPAN Operations

- In order for a 6LoWPAN network to start functioning:

  1. Link-layer connectivity between nodes (commissioning) Compatible physical and link-layer settings (modulation, channel, addressing, security, etc..)

  2. Network layer address configuration, discovery of neighbors, registrations (bootstrapping): 6LoWPAN-ND

  3. Routing algorithm sets up paths (route initialization)

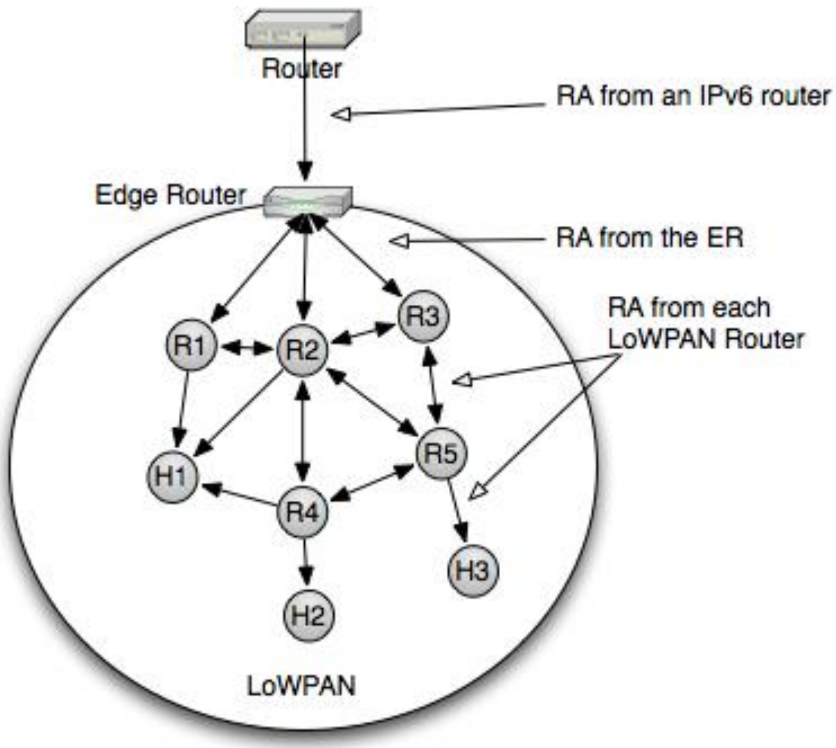
  4. Continuous maintenance of 1-3

Bootstrapping

- 6LoWPAN nodes must *auto-configure* themselves without human intervention: need Bootstrapping.
- Standard ND for IPv6 is not appropriate for 6LoWPAN:
  - Assumption of a single link for an IPv6 subnet prefix
  - Assumption that nodes are always on
  - Heavy use of multicast traffic (broadcast/flood in 6LoWPAN)
  - No efficient multi-hop support over e.g. 802.15.4
- 6LoWPAN Neighbor Discovery provides:
  - An appropriate link and subnet model for low-power wireless
  - Minimized node-initiated control traffic
  - Node Registration (NR) and Confirmation (NC)
  - Duplicate Address Detection (DAD) and recovery
  - Support for extended Edge Router infrastructures
- ND for 6LoWPAN has been specified in draft-ietf-6lowpan-nd: ***work in progress***

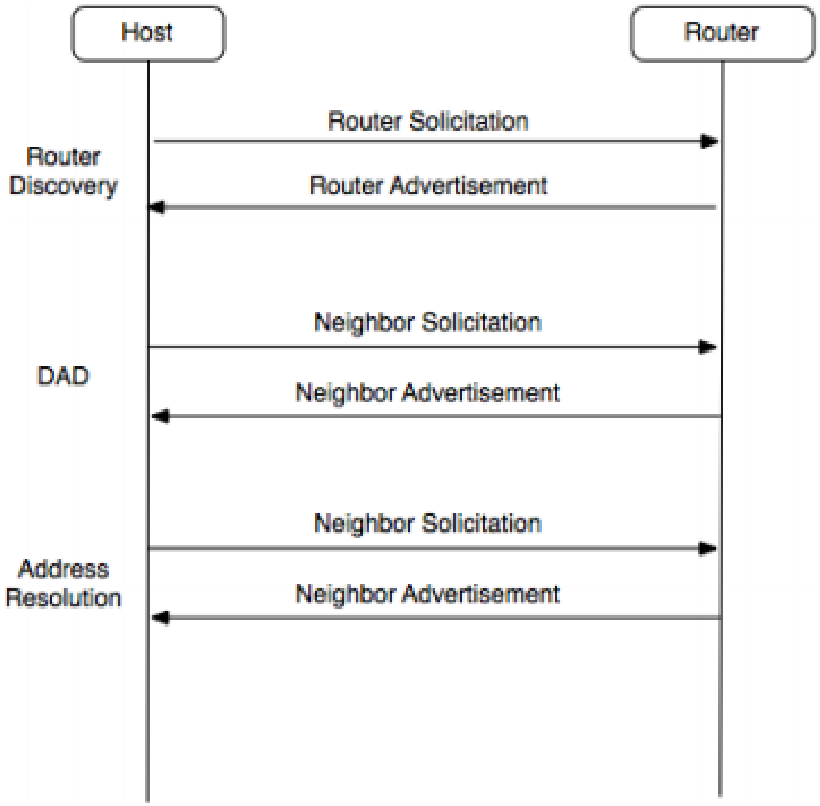
Prefix Dissemination

- In normal IPv6 networks RAs are sent to a link based on the information (prefix etc.) configured for that router interface
- In ND for 6LoWPAN RAs are also used to automatically disseminate router information across multiple hops



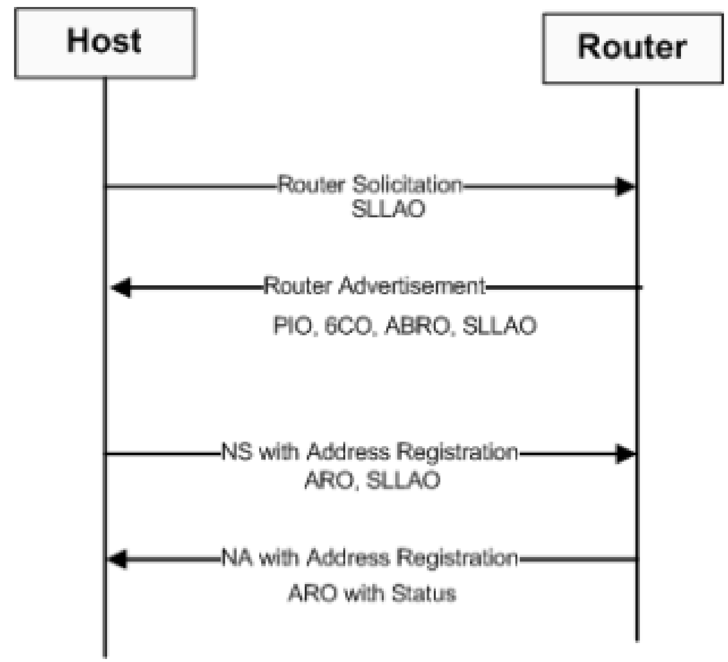
IPv6 vs 6LoWPAN Neighbor Discovery

IPv6 ND (RFC 4861) – Mulicast



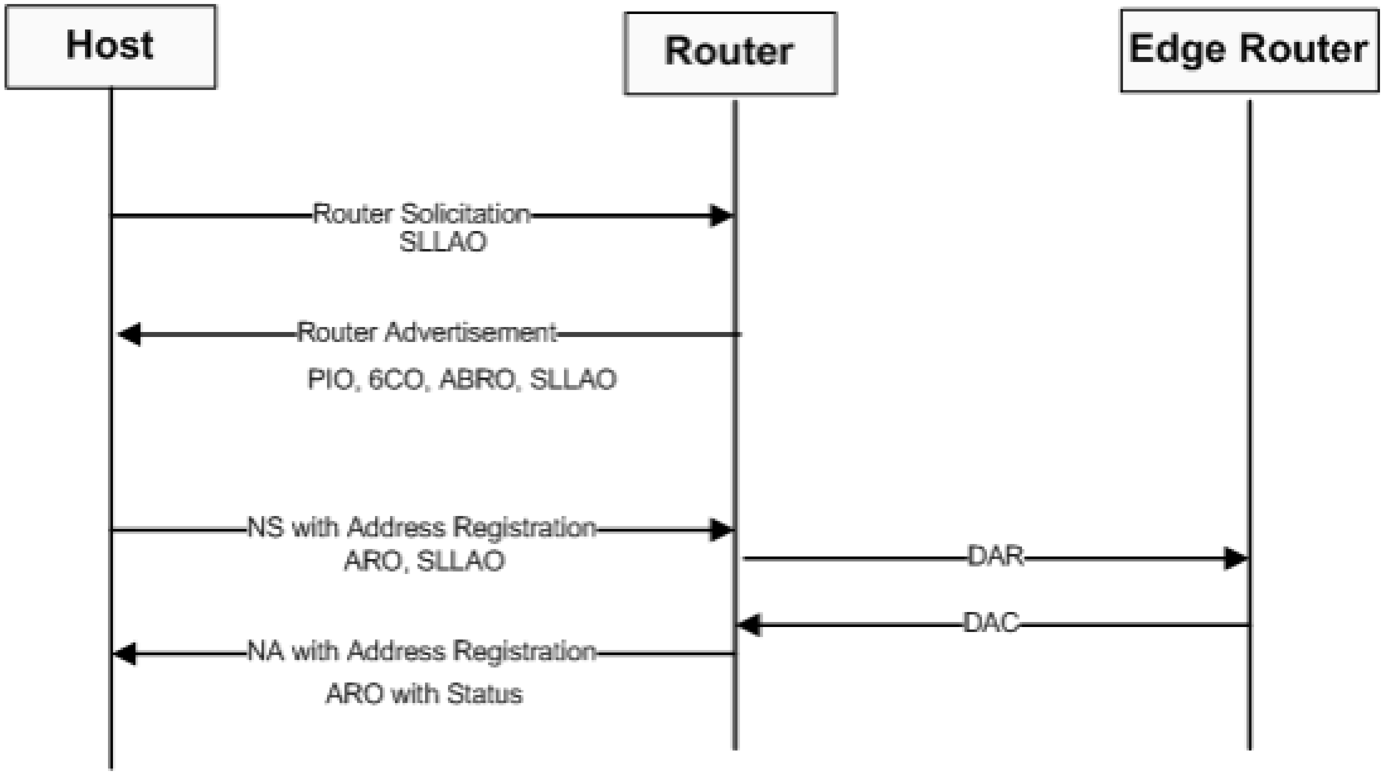
ND: Neighbor Discovery   NS: Neighbor Solicitation
NA: Neighbor Advertisement
DAD: Duplicate Address Detection

6LoWPAN ND (draft-ietf-6lowpan-nd-18) - Unicast



SLLAO: Source Link Layer Address Option
PIO: Prefix Information Option   ARO: Address Registration Option
6CO: 6LoWPAN Context Option
ABRO: Authorative Border Router Option

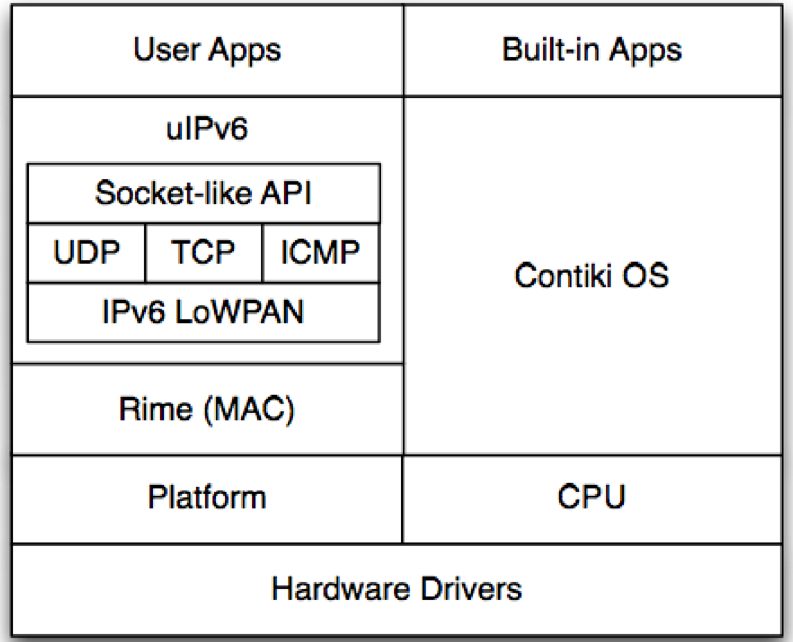
6LoWPAN – Duplicate Address Detection



- ❑ DAR: Duplicate Address Registration
- ❑ DAC: Duplicate Address Confirmation

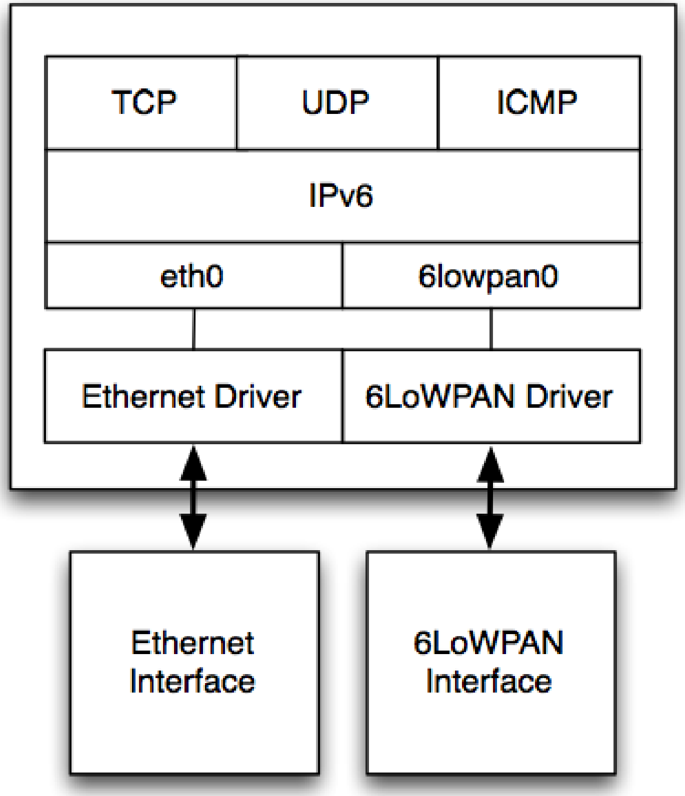
Contiki uIPv6

- Popular embedded OS for small microcontrollers
  - MSP430, AVR, PIC, 8051 etc.
- http://www.sics.se/contiki
- Standard C-based
- Portable applications
- Lightweight protothreads
- uIPv6 Stack
  - Full IPv6 support
  - RFC4944 + 6lowpan-hc
  - UDP, TCP, ICMPv6
- Great for research



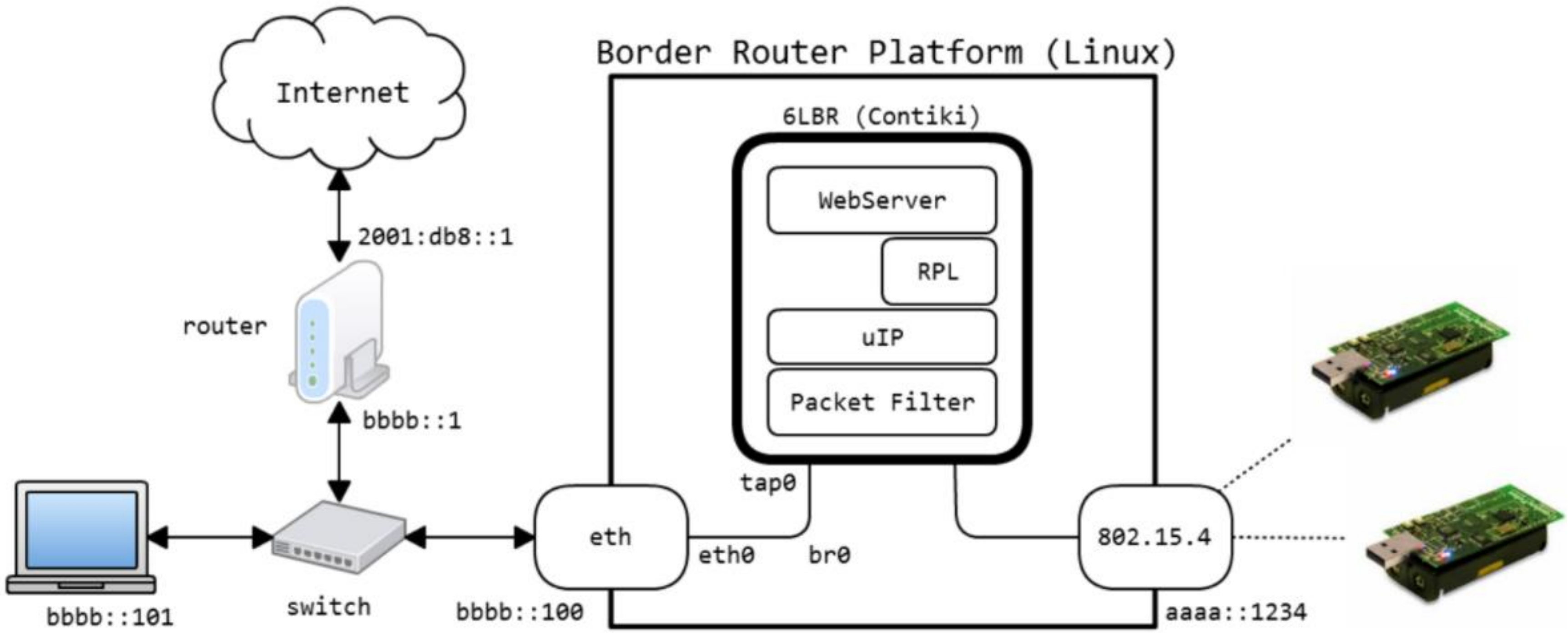
Router Integration

- ❑ Edge Routers/Border Routers interconnect the IPv6 world and 6LoWPAN
- ❑ An BR needs to implement:
  - ❖ 6LoWPAN interface(s)
  - ❖ 6LoWPAN adaptation
  - ❖ Simple 6LoWPAN-ND
  - ❖ A full IPv6 protocol stack
- ❑ Other typical features include:
  - ❖ IPv4 support and tunneling
  - ❖ Application proxy techniques
  - ❖ Extended LoWPAN support
  - ❖ A firewall
  - ❖ Management



6LoWPAN Router: 6LBR

- Rhttps://github.com/cetic/6lbr/wiki
- Can be used with Cooja simulator
- Run with multiple platforms: VM, Raspberry, BeagleBone, CC2538,



6LoWPAN Router: 6LBR

- **WSN Management**
  - RPL storing mode and non storing mode
  - 6LoWPAN
  - Variety of network architectures
    - **Router**: Real routing between IP and 6LoWPAN, treated as independent networks.
    - **Smart-Bridge**: Bridged deployment between the 802.15.4 and Ethernet interfaces.
    - **Transparent-Bridge**: 802.15.4 interface fully-bridged with the Ethernet interface
  - Network auto-configuration
  - Statefull NAT64 to provides bidirectional IPv4 connectivity
  - DNS Proxy
  - Multicast communication
- **Radio**
  - MAC layers supported : CSMA, ContikiMAC, TSCH
  - Local or remote slip-radio
  - Multi-radio support

6LoWPAN Router: 6LBR

Configuration

- Reconfiguration without recompiling also on embedded platforms
  - IP configuration
  - WSN configuration
  - RPL Parameters
  - Router Advertisement
  - MAC and Security Layers
- An enhanced **webserver** with configuration commands
- Command line tool
- Node port mapping configuration

Monitoring

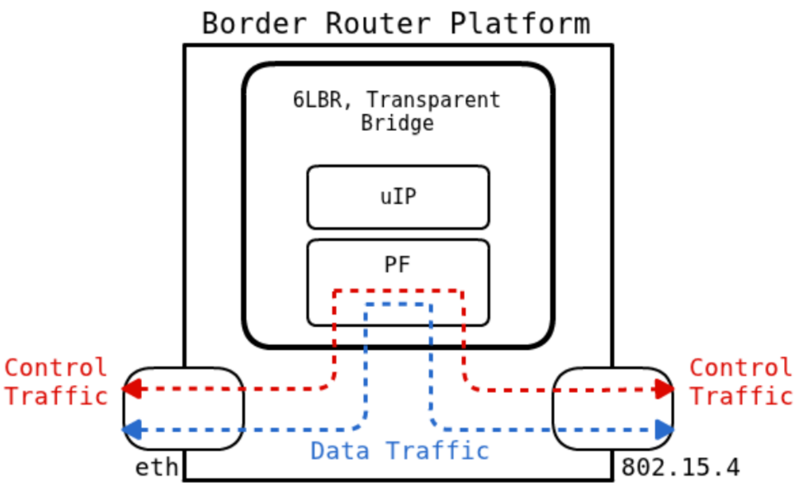
- RPL DAG visualization
- List of connected elements
- Traffic statistics per node
- Router statistics

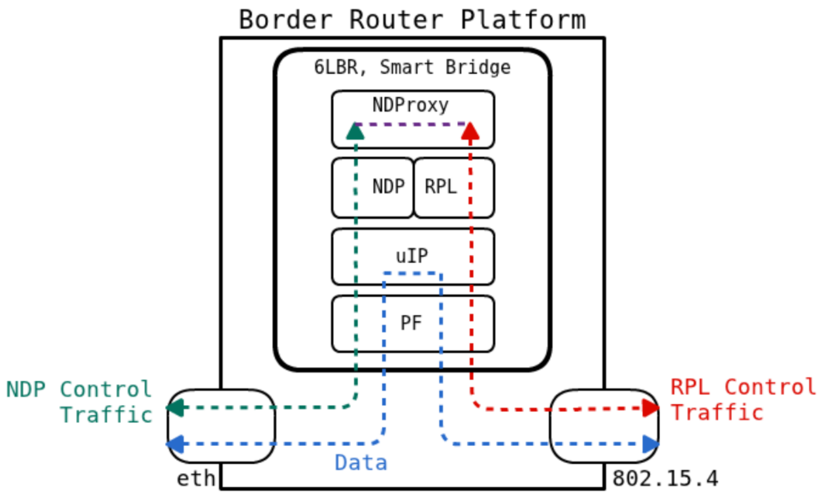
6LoWPAN Router: 6LBR

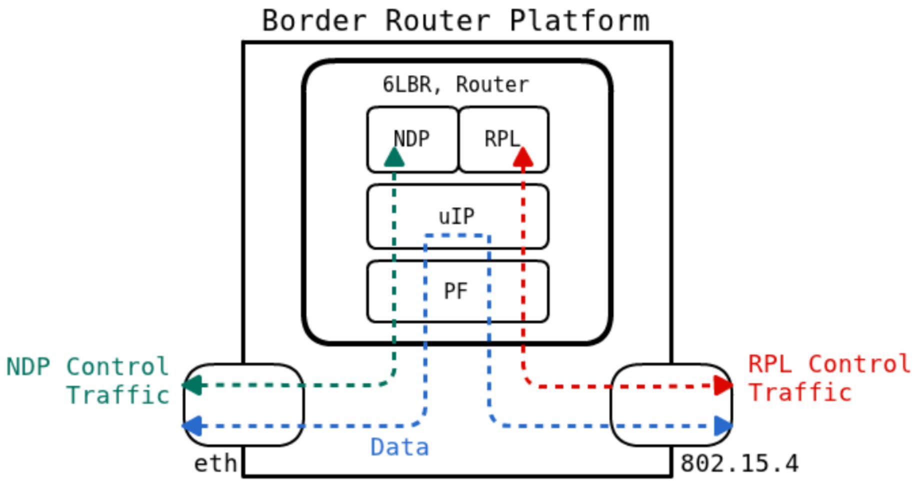
- **Security**
  - Optional 802.15.4 Security Layer
  - Node filtering
  - Webserver configuration and deactivation
  - DTLS examples
- **Expandability**
  - User scriptable **ifup/ifdown**
  - Plug-ins support on Linux based platforms
  - User modules on embedded platforms.

6LBR: Modes

NDP: Neighbor Discovery Proxies
Modified from [RFC4389]

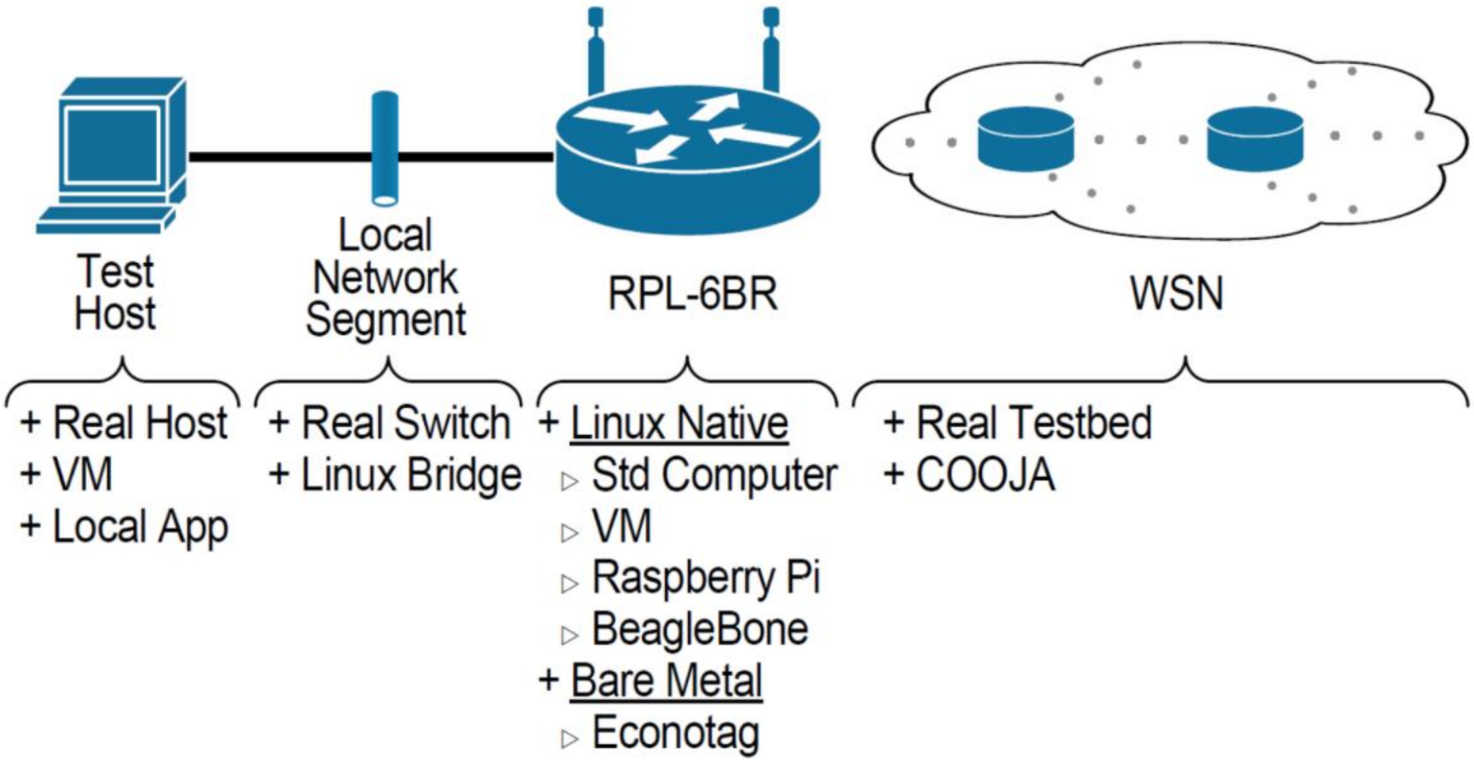






6LBR Framework

- Framework testing:



https://github.com/cetic/6lbr/wiki/COOJA-Interface

6LBR Platforms

RPi + Nooliberry

RPi + any « Contiki Mote »

BeagleBone

Redwire, LLC

Econotag

BR12