

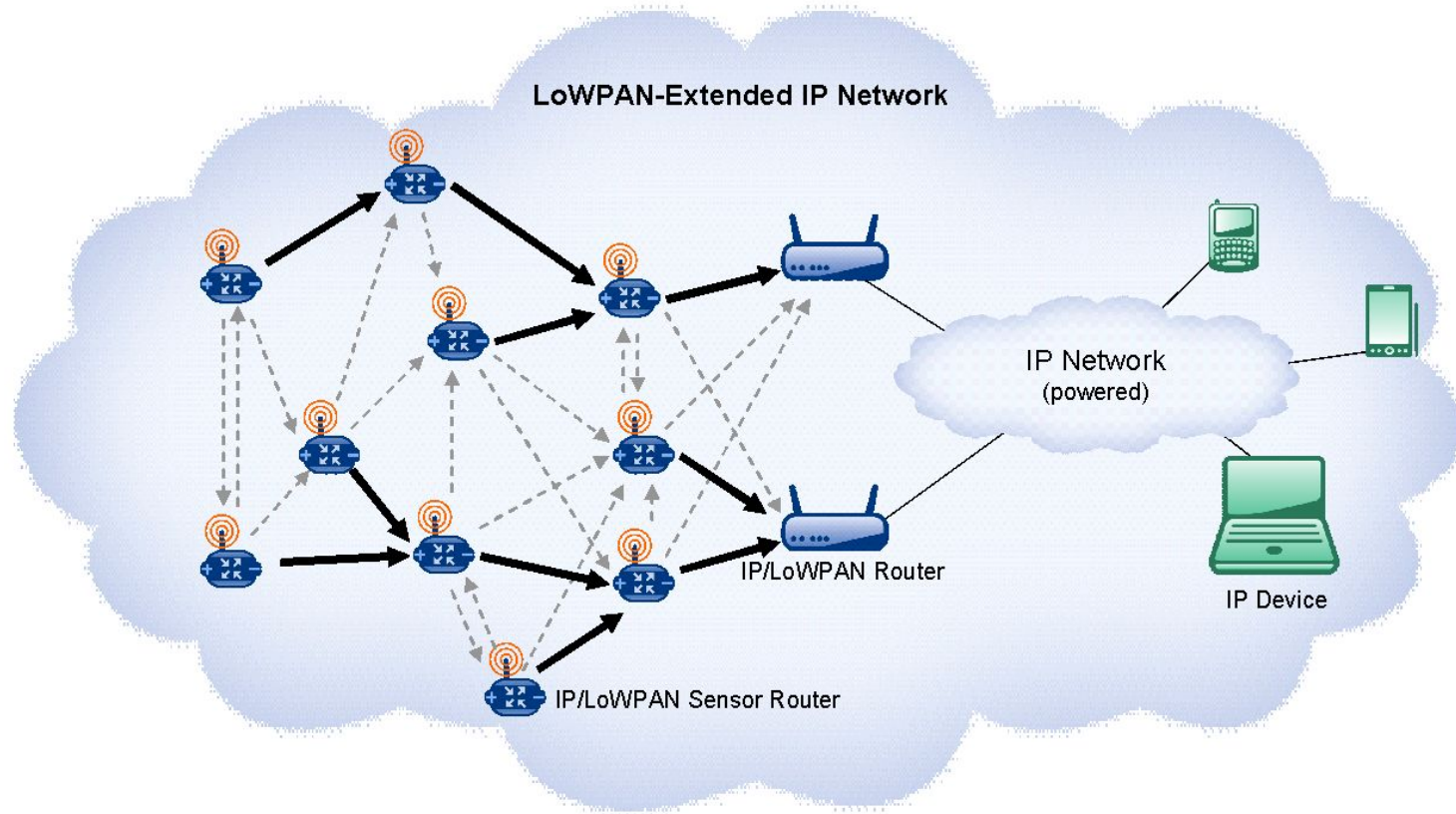


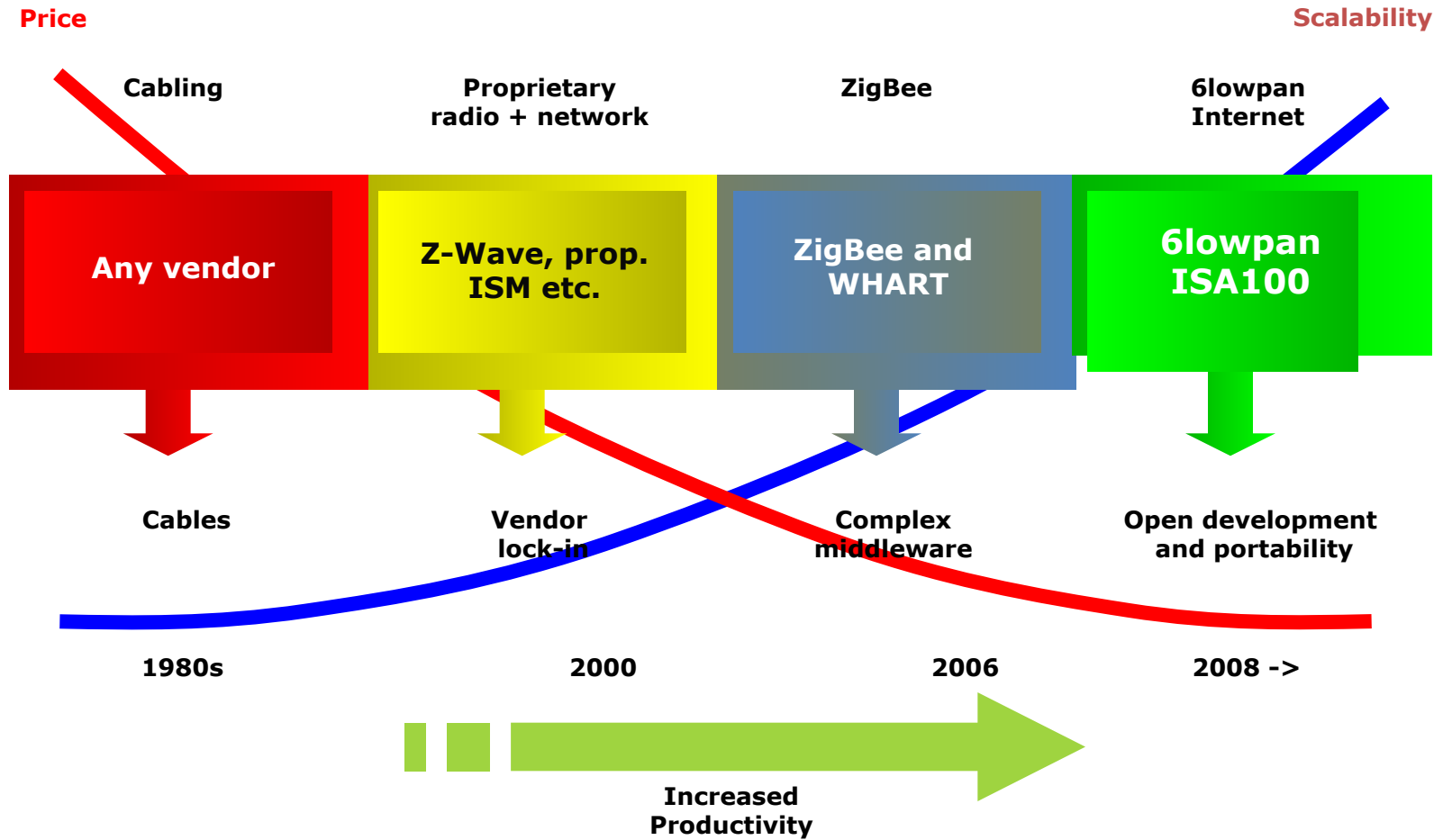
UNIVERSIDAD
COMPLUTENSE
MADRID

6LoWPAN

Networks and protocols 1

Facultad de Informática



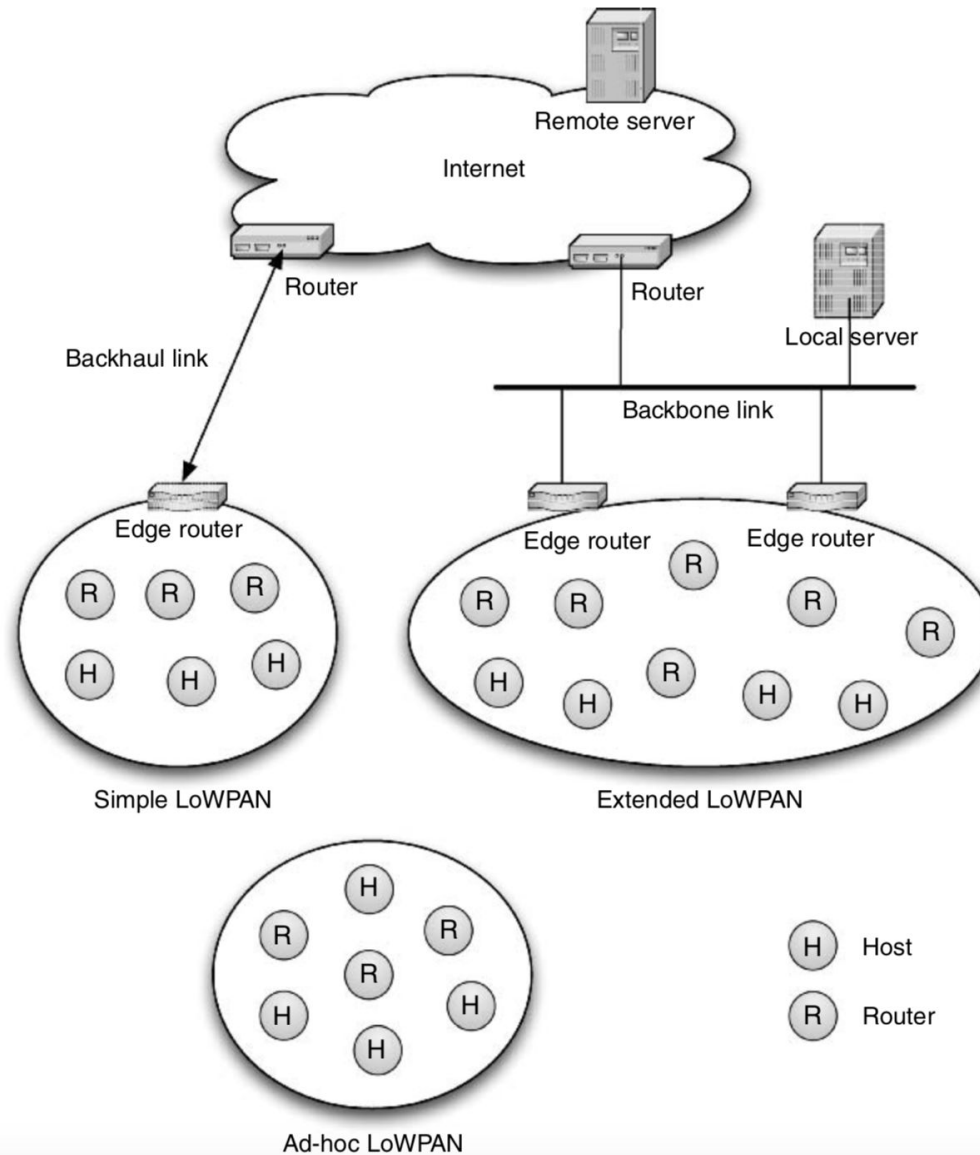


- Low power RF + IPv6 = *The Wireless Embedded Internet*
- Benefits:
 - **Open standards**, reliable and long life
 - **Easy learning curve**
 - Transparent integration in the **Internet**
 - **Global scalability**
 - **End-to-end** data flow
 - **No Gateways**

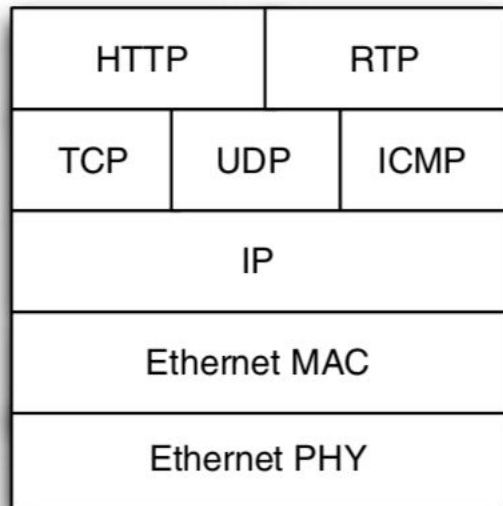


- Specified by the Internet Engineering Task Force (ETF)
 - RFC4919: requirements
 - RFC4944: 6LoWPAN encapsulation
 - RFC6282: update of the RFC4944 with new compressed format
 - RFC6550: RPL
 - RFC6568: 6LowPAN applications and use cases
 - RFC6606: Routing problem specification
 - RFC6775: Neighbour Discovery
 - RFC6066: Header extensions

6LoWPAN architecture



IP Protocol Stack



Application

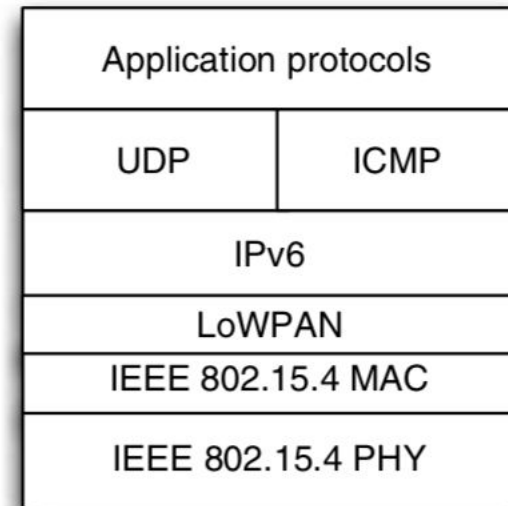
Transport

Network

Data Link

Physical

6LoWPAN Protocol Stack



- Light protocols are preferred
 - UDP instead of TCP
 - Light alternatives to other application protocols (http, rest, soap, ...)

- 128 bits (16 bytes) addresses, several notations
 - Huge space: the population of the world is reaching 2^{34} people, each of them could have 2^{94} IPv6 addresses

Binary

1000000001011011001011011001110111011100001010000000000000000000
000000000000000001111110001010111110101001100100000011111111111111

Dotted Decimal

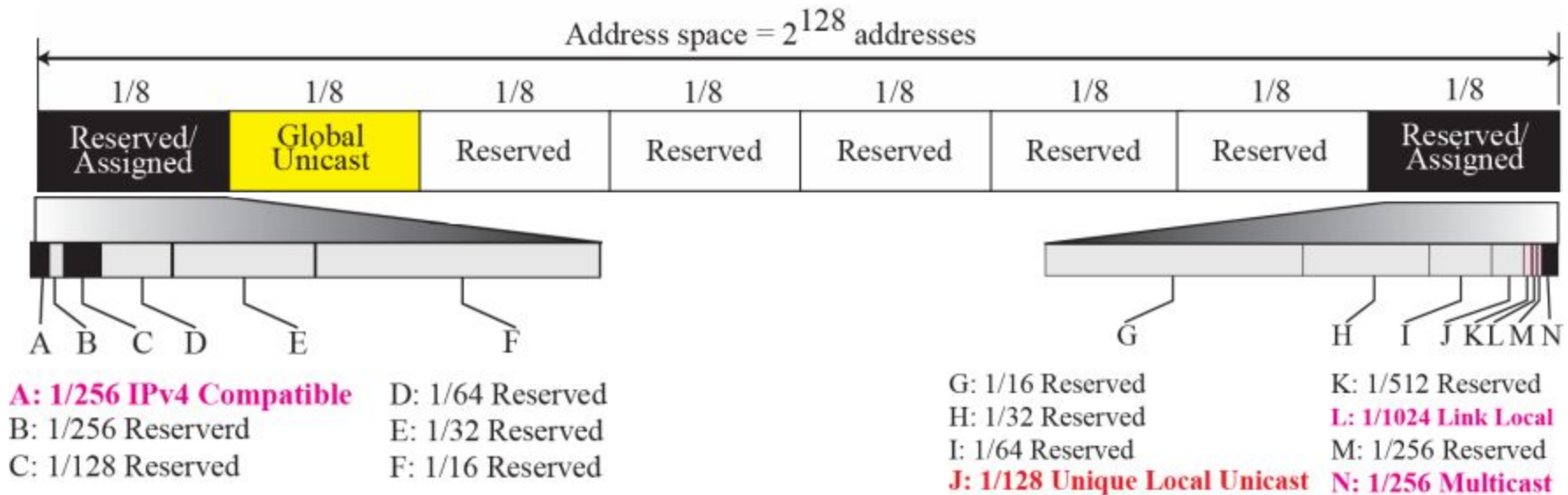
128	91	45	157	220	40	0	0	0	0	252	87	212	200	31	255
-----	----	----	-----	-----	----	---	---	---	---	-----	----	-----	-----	----	-----

	0	32	64	96	128					
Hexadecimal										
Straight Hex	805B	2D9D	DC28	0000	0000	FC57	D4C8	1FFF		
Leading-Zero Suppressed	805B	2D9D	DC28	0	0	FC57	D4C8	1FFF		
Zero-Compressed	805B	2D9D	DC28	::		FC57	D4C8	1FFF		
Mixed Notation	805B	2D9D	DC28	::		FC57	212	200	31	255

- Prefixes: common part of the network addresses, indicates the network
 - CIDR (Classless Inter Domain Routing) notation
 - Address/Prefix length in bits

FDEC :: BBFF : 0 : FFFF/60

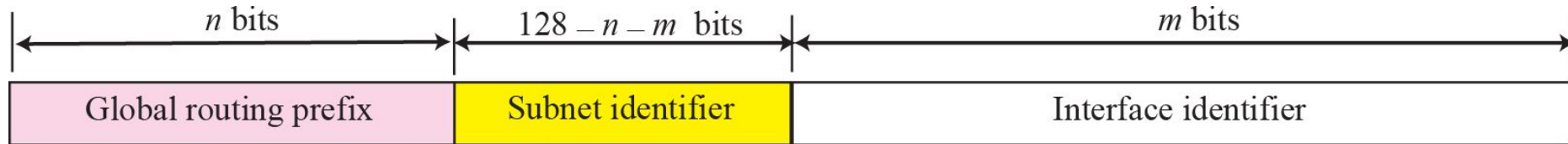
- Address space organization:



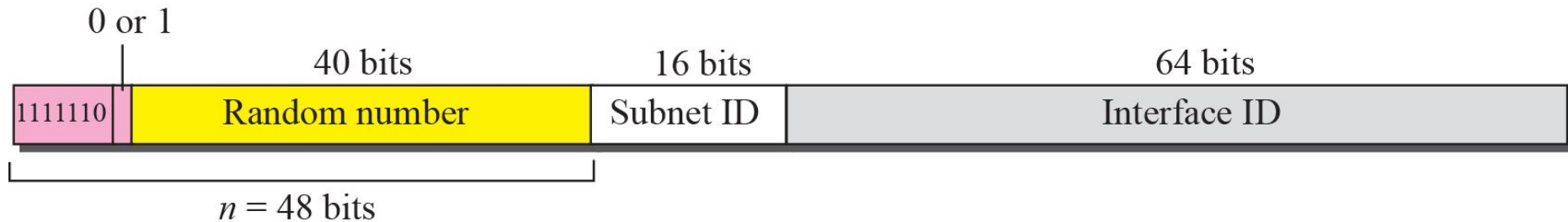
	<i>Block Prefix</i>	<i>CIDR</i>	<i>Block Assignment</i>	<i>Fraction</i>
1	0000 0000	0000:: 8</td <td>Reserved (IPv4 compatible)</td> <td>1/256</td>	Reserved (IPv4 compatible)	1/256
	0000 0001	0100:: 8</td <td>Reserved</td> <td>1/256</td>	Reserved	1/256
	0000 001	0200:: 7</td <td>Reserved</td> <td>1/128</td>	Reserved	1/128
	0000 01	0400:: 6</td <td>Reserved</td> <td>1/64</td>	Reserved	1/64
	0000 1	0800:: 5</td <td>Reserved</td> <td>1/32</td>	Reserved	1/32
	0001	1000:: 4</td <td>Reserved</td> <td>1/16</td>	Reserved	1/16
2	001	2000::<!--3</b-->	Global unicast	1/8
3	010	4000:: 3</td <td>Reserved</td> <td>1/8</td>	Reserved	1/8
4	011	6000:: 3</td <td>Reserved</td> <td>1/8</td>	Reserved	1/8
5	100	8000:: 3</td <td>Reserved</td> <td>1/8</td>	Reserved	1/8
6	101	A000:: 3</td <td>Reserved</td> <td>1/8</td>	Reserved	1/8
7	110	C000:: 3</td <td>Reserved</td> <td>1/8</td>	Reserved	1/8
8	1110	E000:: 4</td <td>Reserved</td> <td>1/16</td>	Reserved	1/16
	1111 0	F000:: 5</td <td>Reserved</td> <td>1/32</td>	Reserved	1/32
	1111 10	F800:: 6</td <td>Reserved</td> <td>1/64</td>	Reserved	1/64
	1111 110	FC00:: 7</td <td>Unique local unicast</td> <td>1/128</td>	Unique local unicast	1/128
	1111 1110 0	FE00:: 9</td <td>Reserved</td> <td>1/512</td>	Reserved	1/512
	1111 1110 10	FE80:: 10</td <td>Link local addresses</td> <td>1/1024</td>	Link local addresses	1/1024
	1111 1110 11	FEC0:: 10</td <td>Reserved</td> <td>1/1024</td>	Reserved	1/1024
	1111 1111	FF00:: 8</td <td>Multicast addresses</td> <td>1/256</td>	Multicast addresses	1/256

- Types
 - **Unicast:** addresses a single interface of a node
 - **Multicast:** addresses a group of interfaces. A datagram sent to a multicast address has to reach all the nodes that belong to the group
 - **Anycast:** addresses a group of interfaces. A datagram sent to an anycast address has to be delivered to only one of the devices in the group
- Scopes
 - **Link local:** identifies a node in its level 2 domain (link)
 - **Unique local:** identifies a node in its administrative domain
 - **Global:** identifies a node in the global Internet (unique in the whole Internet)

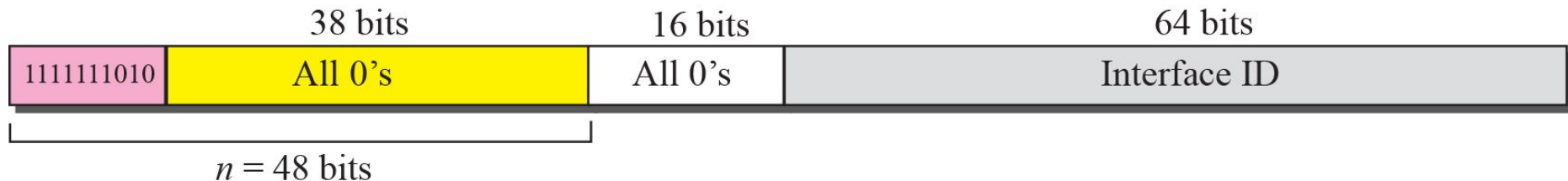
- Global unicast



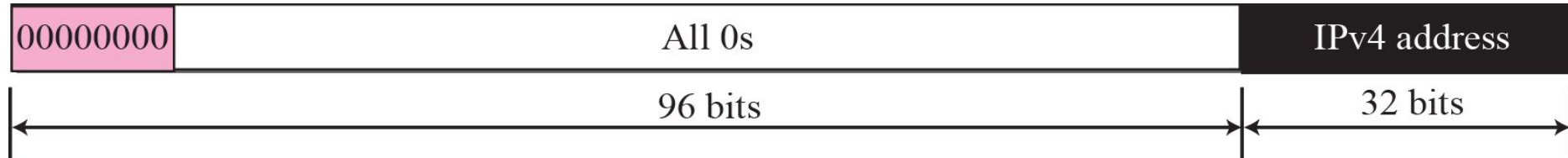
- Unique local unicast



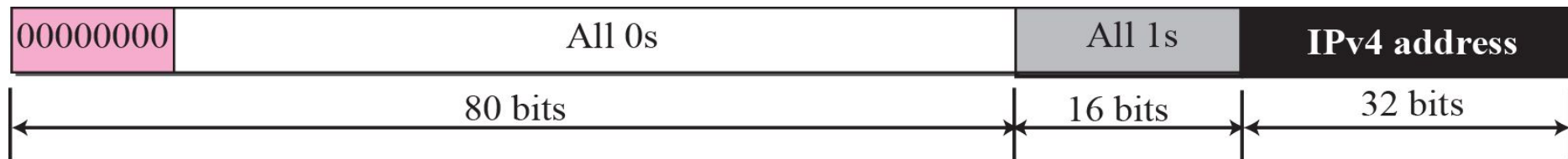
- Link local unicast



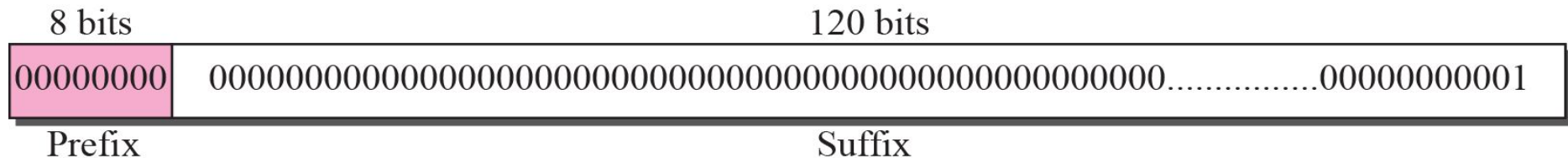
- IPv4 compatible



- IPv4 mapped



- Loopback



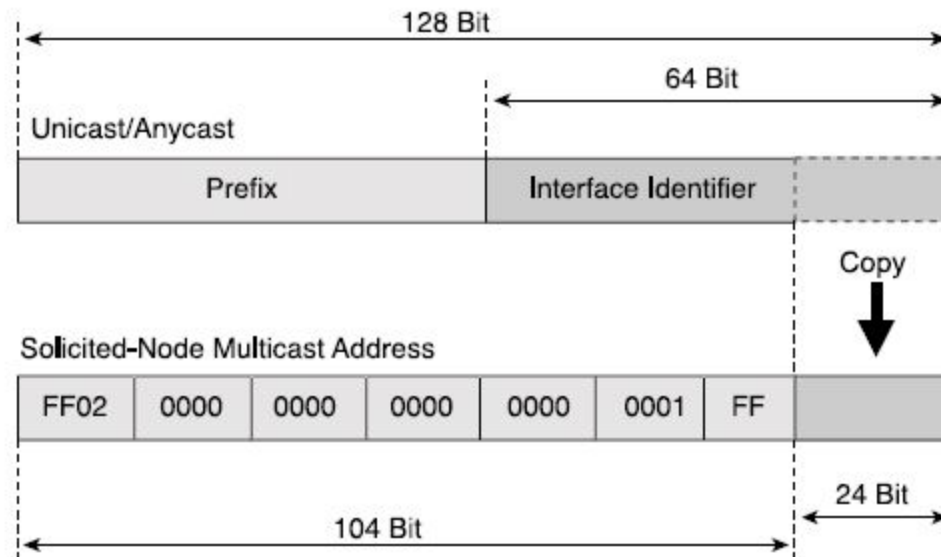
- Site-local (deprecated, in favour of the unique local)



- Multicast Address



- Solicited-Node Multicast Address (used in NDP)

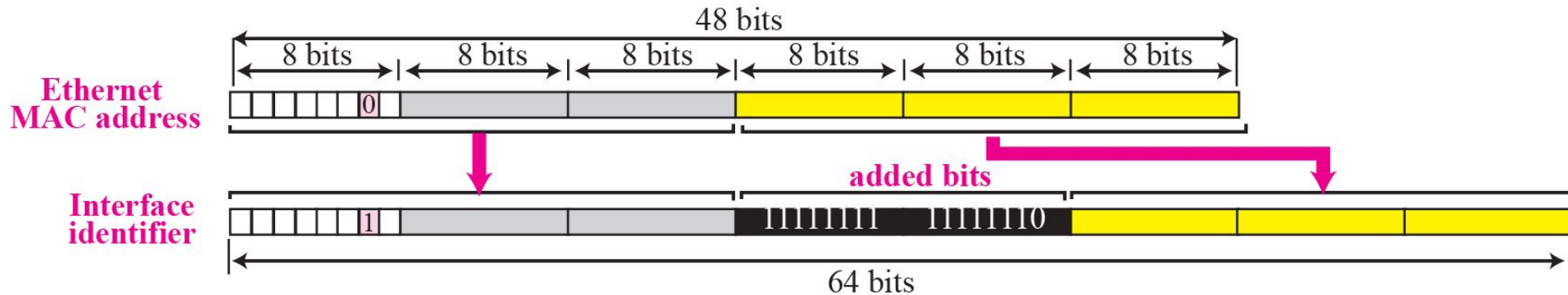


- IPv6 defines its own *Neighbor Discovery Protocol (NDP)*
 - A node uses the NDP to discover other devices in its link, obtain their MAC addresses and find routers
- NDP services
 - Router discovery
 - Prefix discovery
 - Parameter discovery (MTU, hop limit...)
 - Address autoconfiguration
 - Address resolution: obtain mac addresses from IPv6 addresses
 - DAD (Duplicate Address Detection)

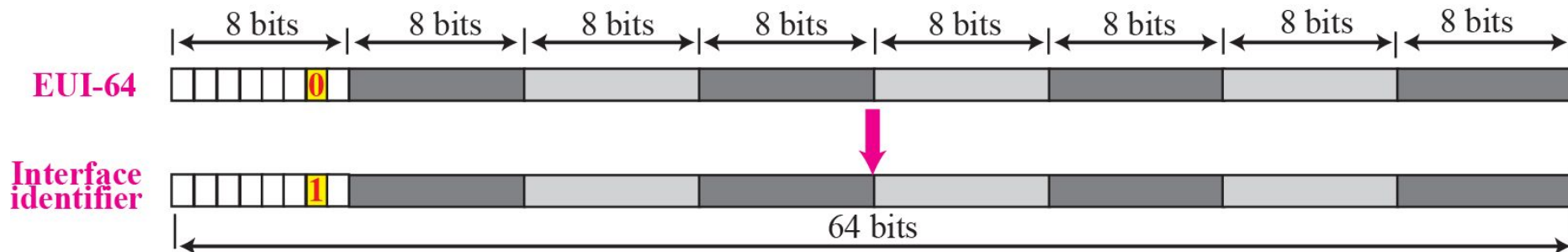
- Neighbor Solicitation Message (NS)
 - Used for address resolution, DAD, and neighbor detection
 - Sent by a node to obtain or confirm the MAC address of a neighbor known its IPv6 address
 - The neighbor responds with a NA message
- Neighbor advertisement (NA)
 - Provides the MAC address after a NS request
- Router Advertisement (RA)
 - Send periodically by the routers
 - Announce the presence of the routers and the parameters of the network (like network prefix, or if DHCP6 shall be used)
 - Send also as a response to a RS
- Router Solicitation (RS)
 - Send by a node to obtain a RA from the router
 - Destination address is usually the *all-routers multicast* (FF02::2)

- The nodes can configure their interface id part of the address:

- From a 48 bits mac address

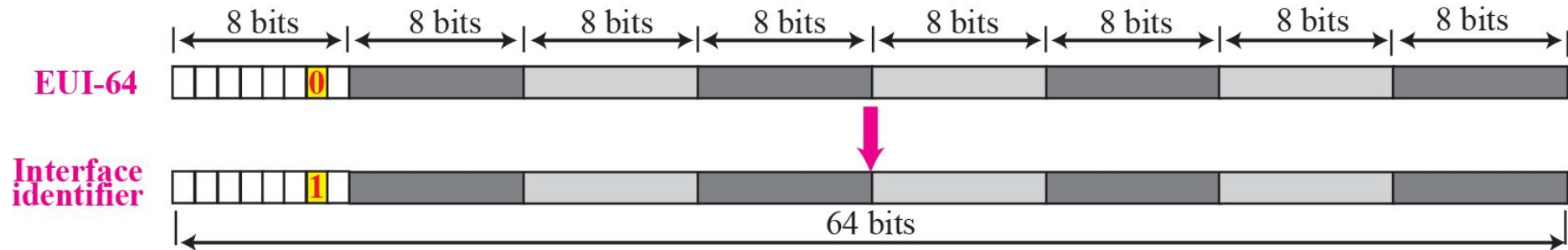


- From a EUI-64 id

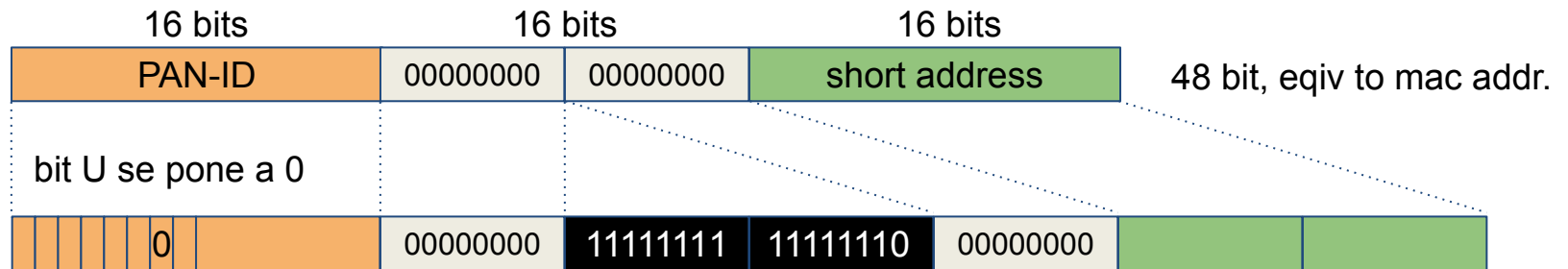


- The prefix used can be the one for a link local address or a unique local address

- From its EUI64 (standard IPv6)



- From the 16 bits id assigned by the PANC



- Before an address can be used, the device must confirm that it is unique (DAD)
 - Interchange of NS and NA messages
- To obtain the *global unicast* address the node has to request the network prefix
 - Can wait to receive a RA message or request one sending a RS to all *routers*

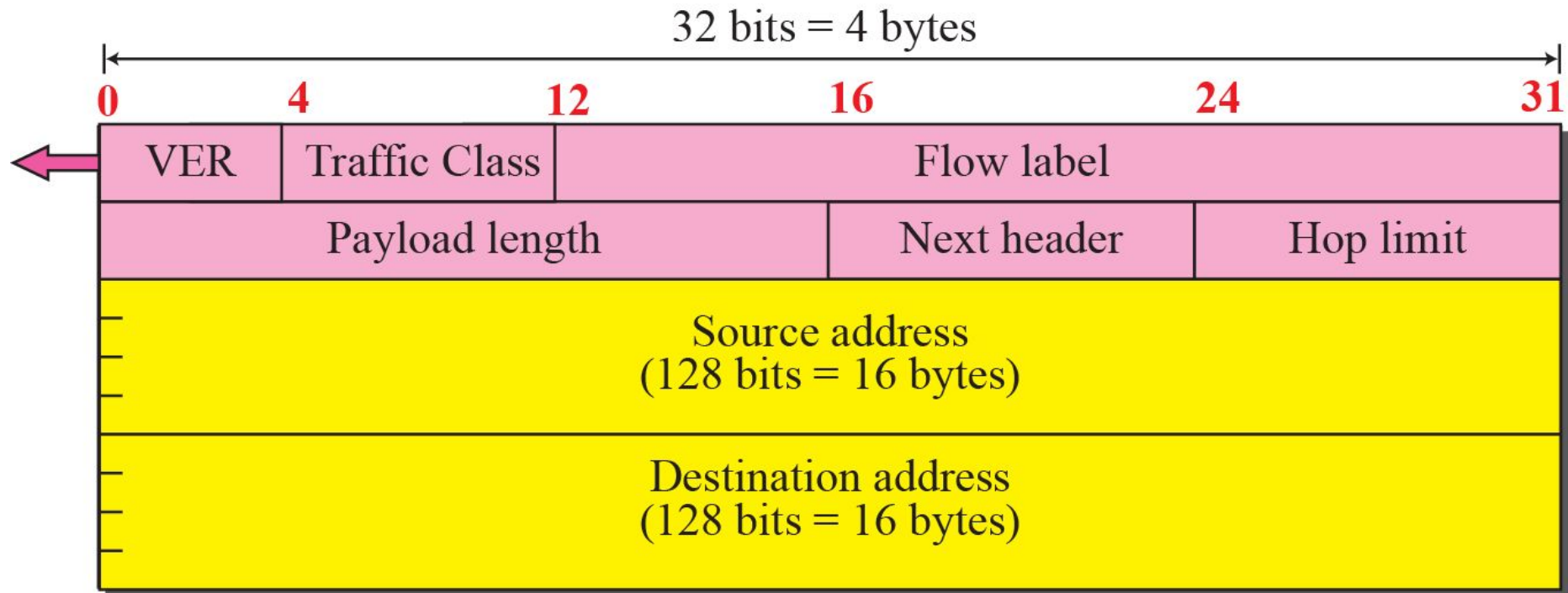
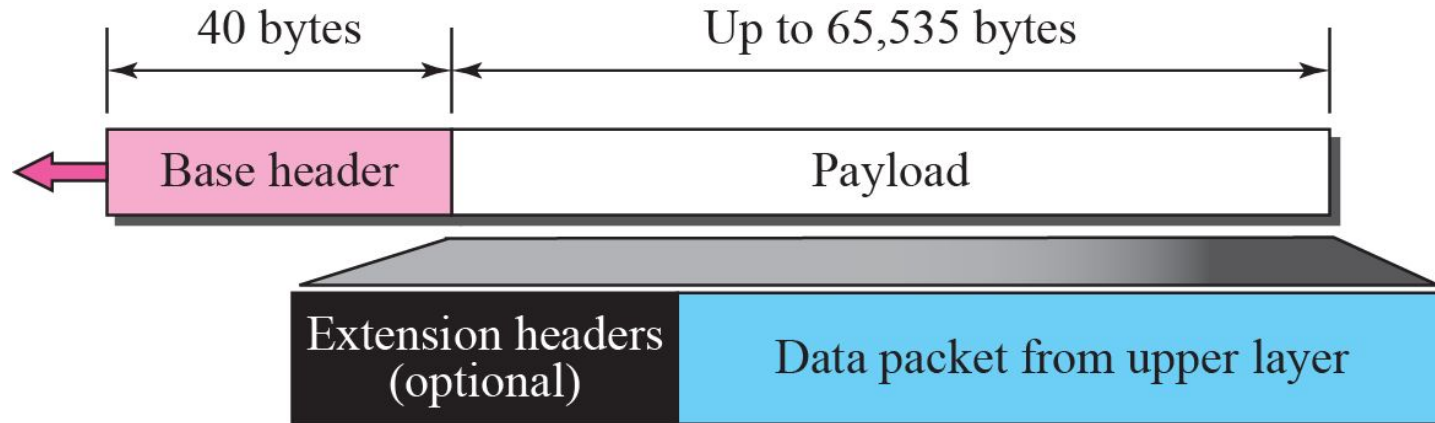


- 1) Compute the interface identifier EUI-64 format
- 2) Assign the solicited-node multicast address to the interface:
FF02:0:0:0:0:1:FFCA:8E47
- 3) Assignment of a link local address:
FF80:0:0:0:0014:B1FF:FECA:8E47
- 4) Building of the global unicast address:
2002:E0:2:3:14:B1FF:FECA:8E47
- 5) DAD process (shown for the global unicast address only)
- 6) In the absence of reply the global unicast address
(2002:E0:2:3:14:B1FF:FECA:8E47) is assigned to the
interface

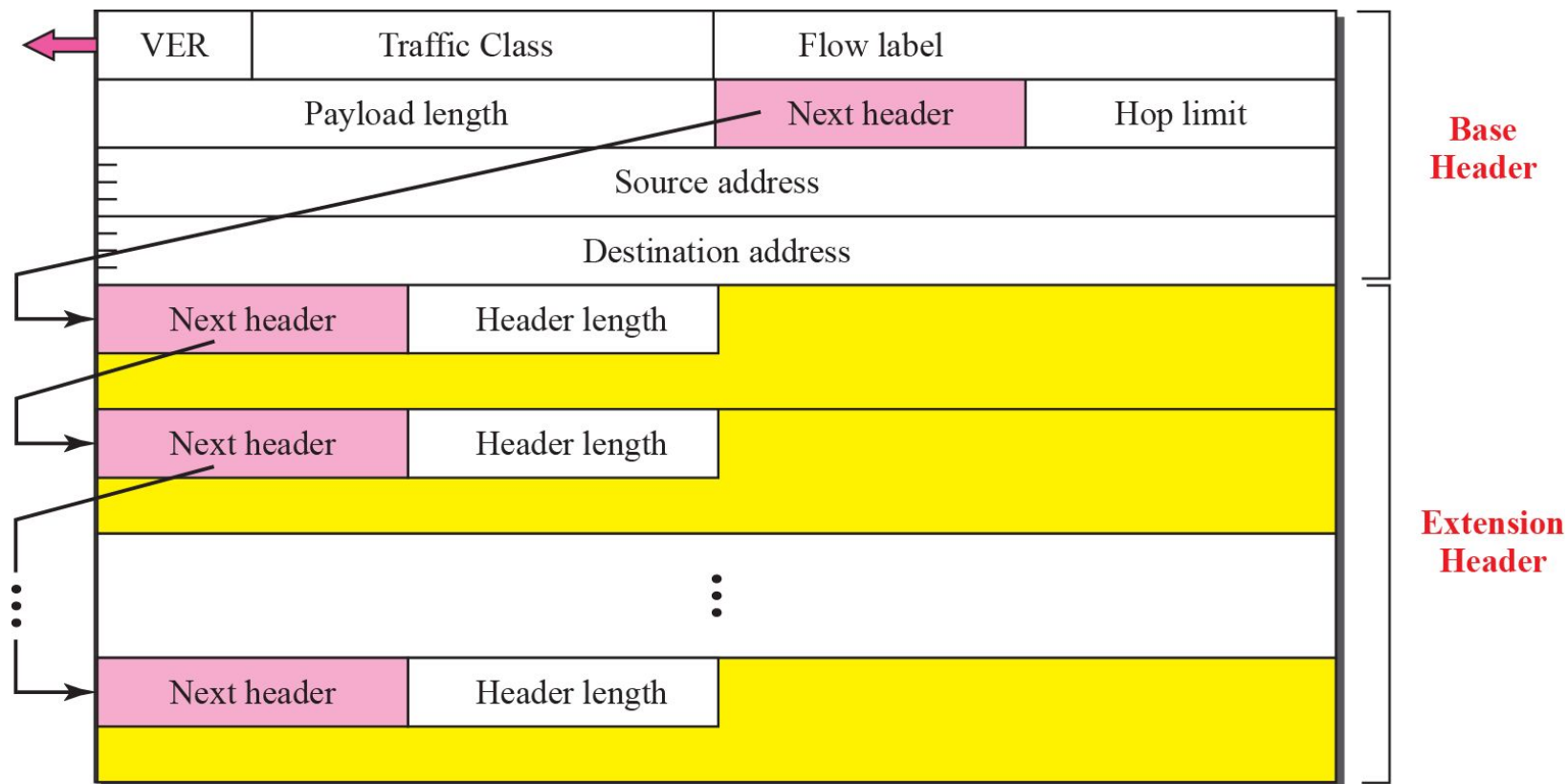
RA: IP Dest: FF02::1,
Prefix=2002:E0:2:3::/64

NS: IP Src: :: (unspecified) - IP Dest:
FF02::1:FFCA:8E47, target address:
2002:E0:2:3:14:B1FF:FECA:8E47

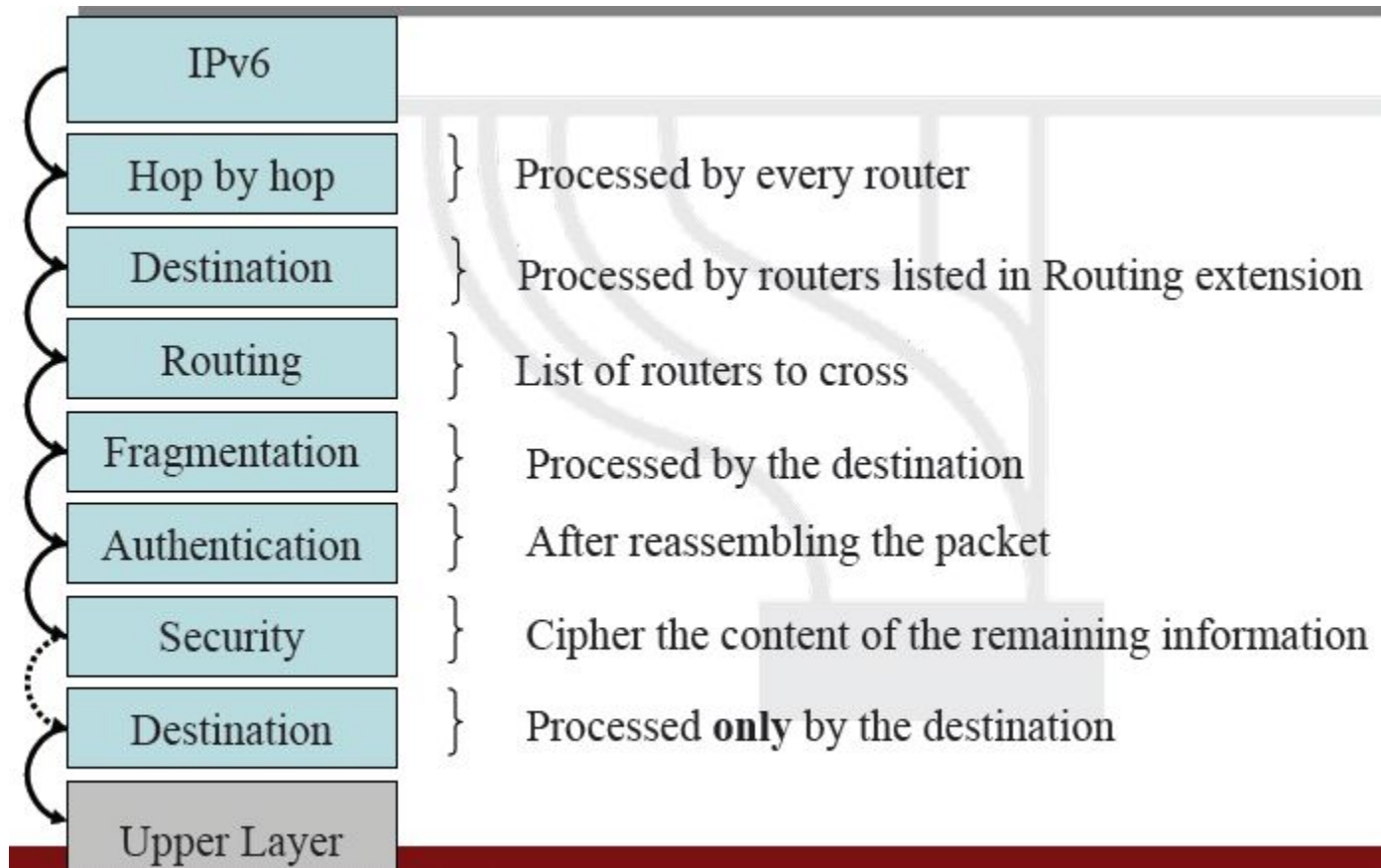
IPv6 datagram

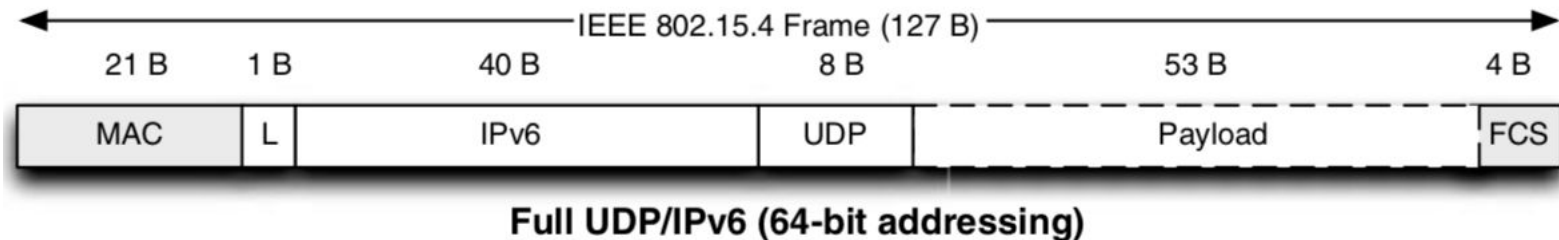


IPv6 datagram



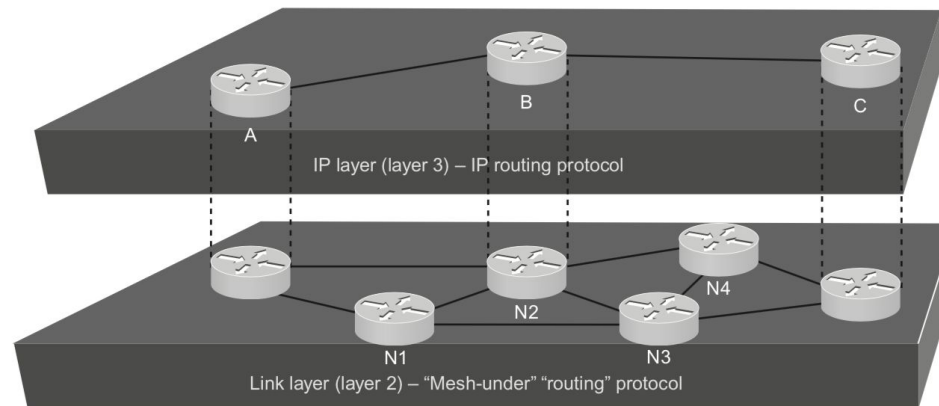
<i>Code</i>	<i>Next Header</i>	<i>Code</i>	<i>Next Header</i>
0	Hop-by-hop option	44	Fragmentation
2	ICMP	50	Encrypted security payload
6	TCP	51	Authentication
17	UDP	59	Null (No next header)
43	Source routing	60	Destination option



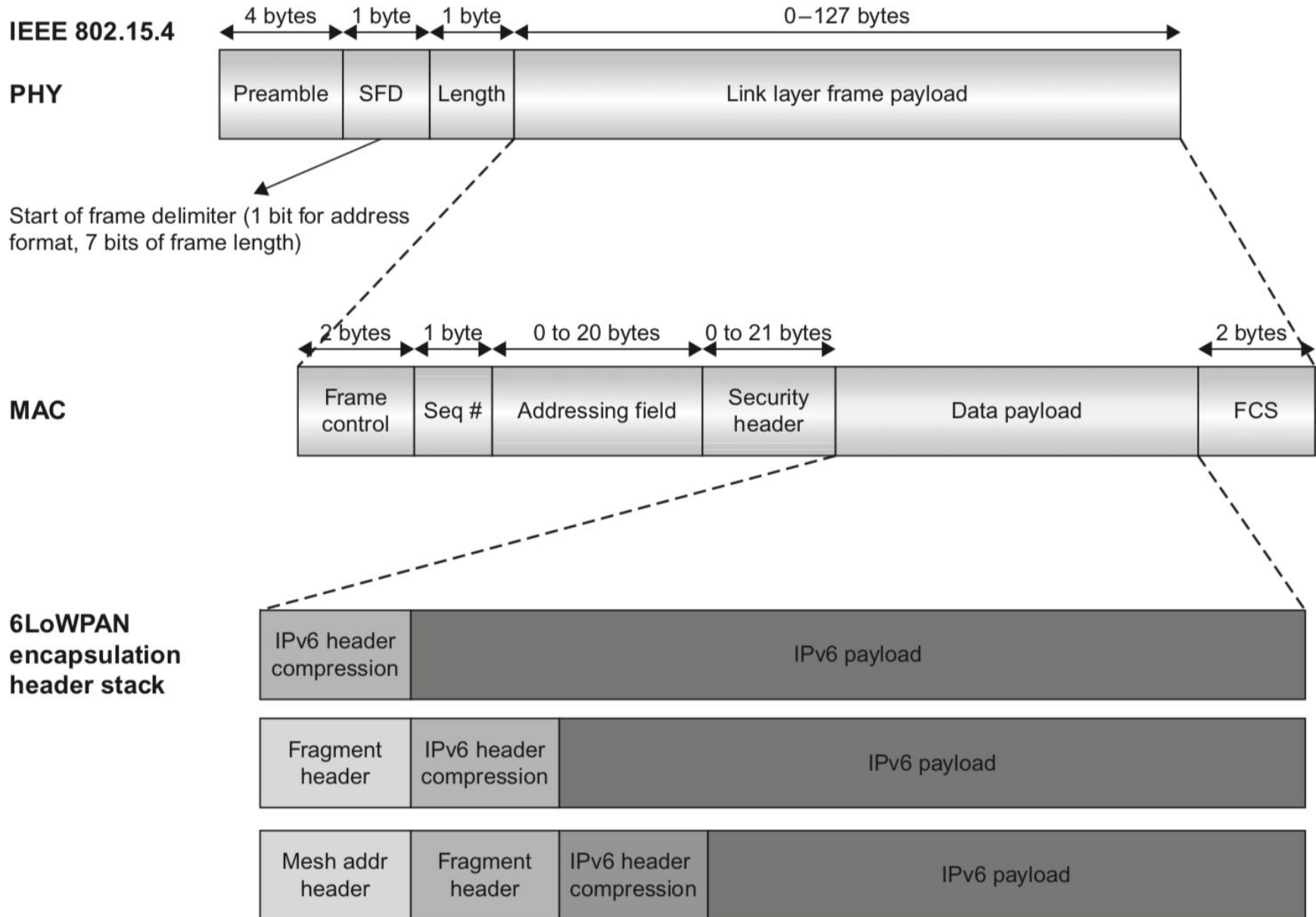


- MTU for 6LowPAN is 1280B
 - 802.15.4 packets are 127B, MSDU of 102B, removing security header (21B) remain 81B, removing 40B from the IPv6 header remain 41B, removing the 8B of the UDP header **remain only 33B** for the application
 - Header compression
 - Take advantage of L2 addresses -> 16 bit short address / 64 bit EUID
 - Fragmentación
- Stateless autoconfiguration
- Short reach => Múltiples Hops
 - Routing at several level (IP + link)

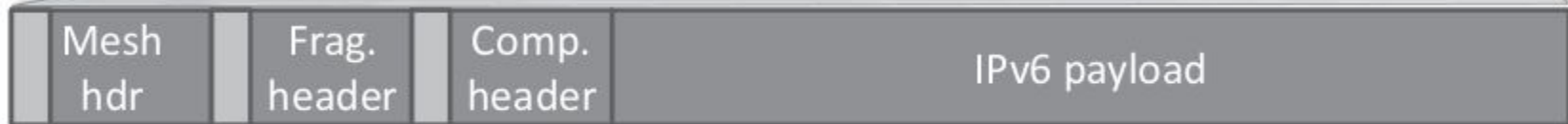
- Two types according to where is the routing done
 - Router-over -> L3
 - Mesh-under -> L2
 - Mixed



6LoWPAN encapsulation

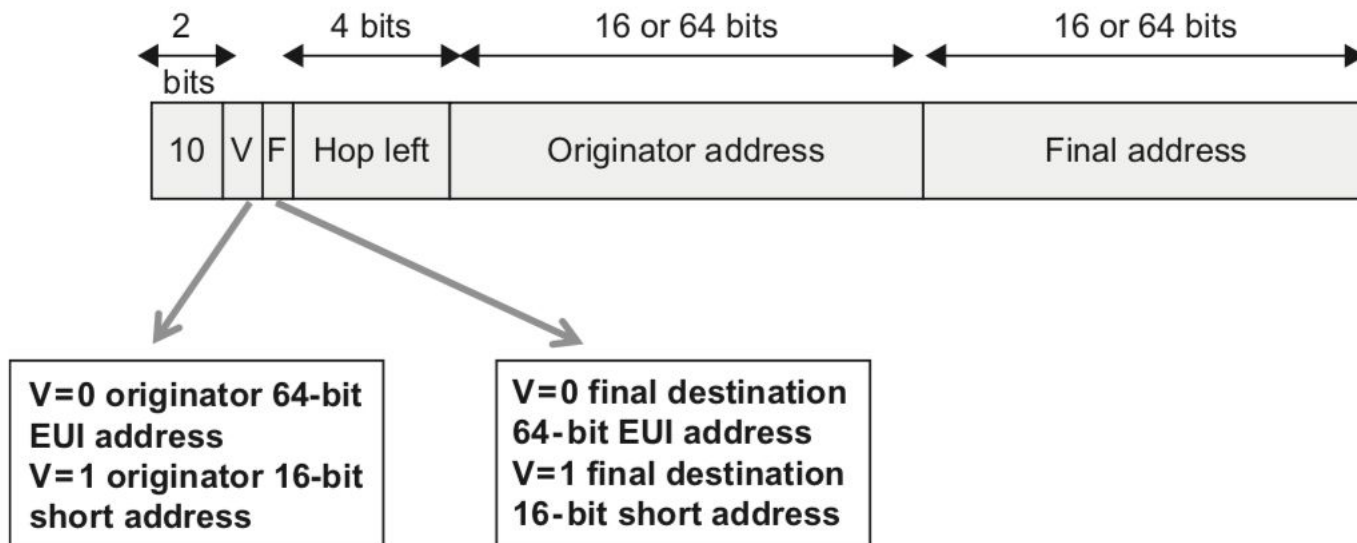


6LoWPAN encapsulation

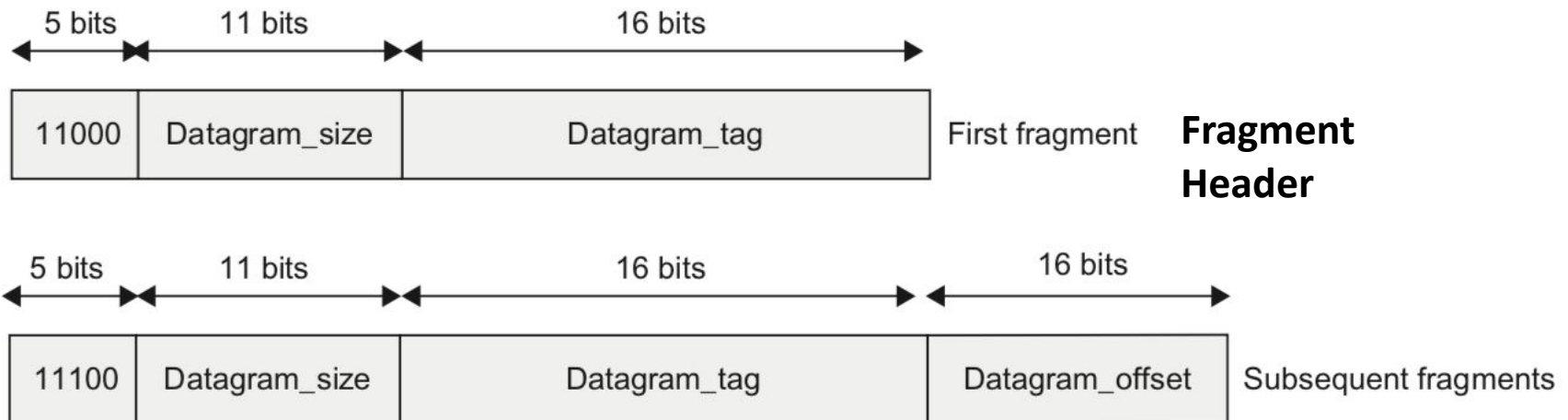


Dispatch byte		Binary	Description
00	Not 6 LowPAN	01000001	Uncompressed IPv6 frame follows
01	IPv6 Addressing header	01000010	HC1 compression follows
01		01010000	LowPANBCO broadcast
01		01111111	Escape code for additional dispatch byte
10	Mesh header		
11	Frag. header	11000xxx	First fragmentation header
		11100xxx	Subsequent fragmentation headers

- L2 Routing Protocol (*mesh-under*)
 - Only FFDs
 - Not used currently but the support for it is there
- Adds source and destination addresses to the header
 - Originator, the original source of the address
 - Final, the final destination for the packet
 - The 802.15.4 header will contain the source and destination for the current hop

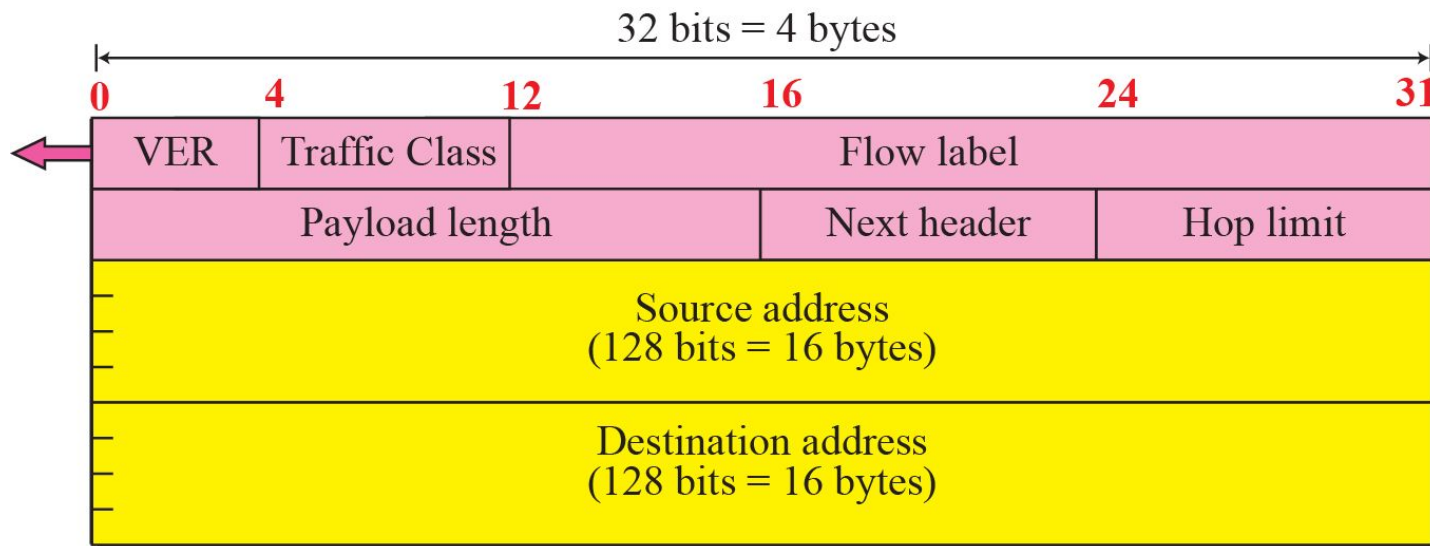


- Necessary when the payload of the IPv6 package does not fit in a single 802.15.4 frame
 - The frame is divided into several fragments
 - The size of the fragments are expressed in multiples of 8 bytes
 - Datagram_size: size of the original IPv6 datagram
 - Datagram_tag: id for the datagram. The same for all fragments
 - Used together with the source and destination addresses to identify the original datagram to which the fragment belongs
 - Datagram_offset: in blocks of 8 bytes

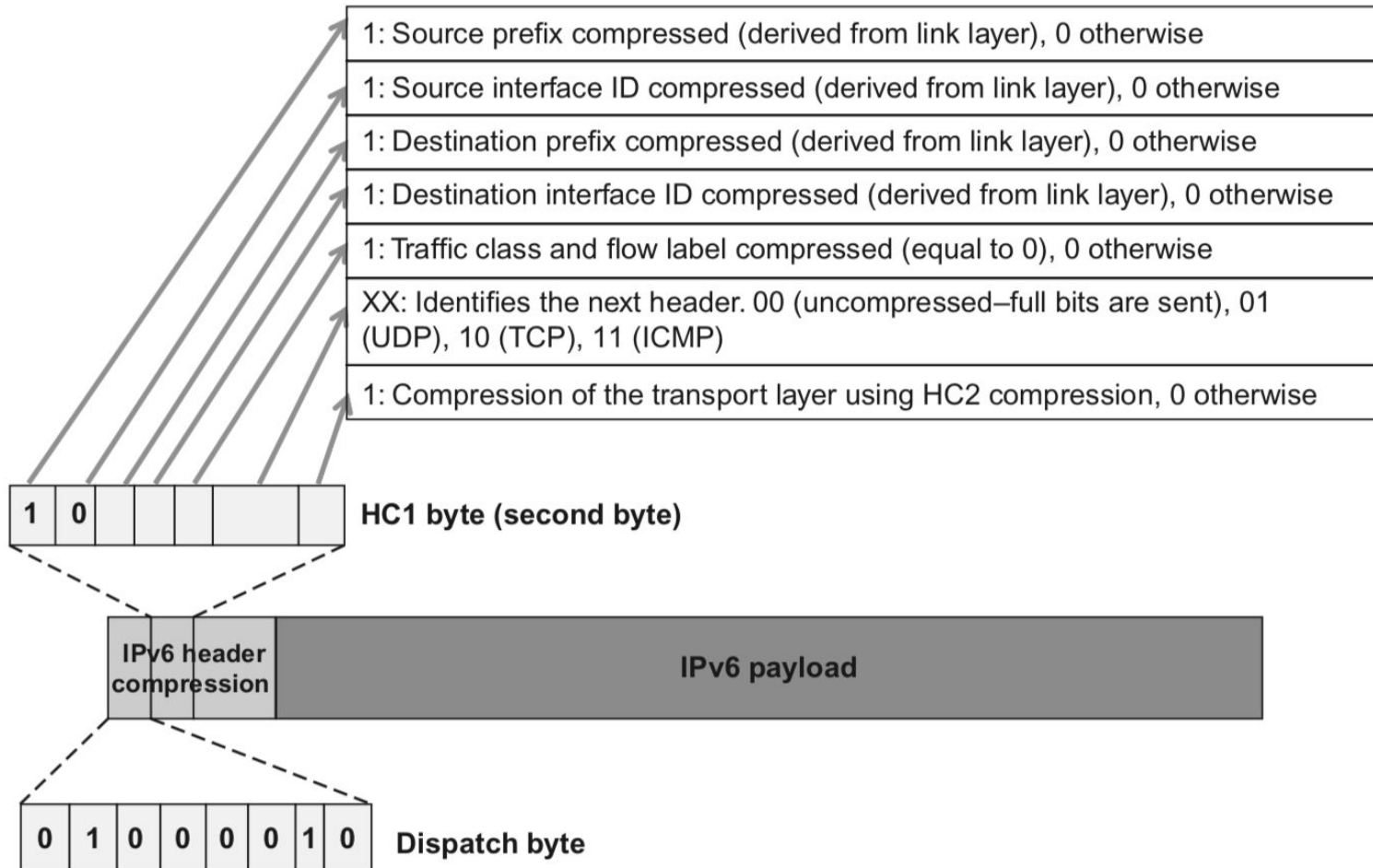


- Several compression techniques exist
- Most of them use state information to achieve higher compression rates
- A stateless compression was first designed
 - Only uses the information in each packet
 - Codes with less bits the most frequent values
 - Takes advantage of the redundancies in the lower layers

- Version: is always 6
- Source and destination addresses are frequently link-local
 - Interface ID can be obtained from the 802.15.4 header
- Length: can be obtained from the phy header in the 802.15.4 frame or the UDP header if present
- Traffic Class and Flow Label are usually 0
- Next Header usually is UDP, TCP or ICMP

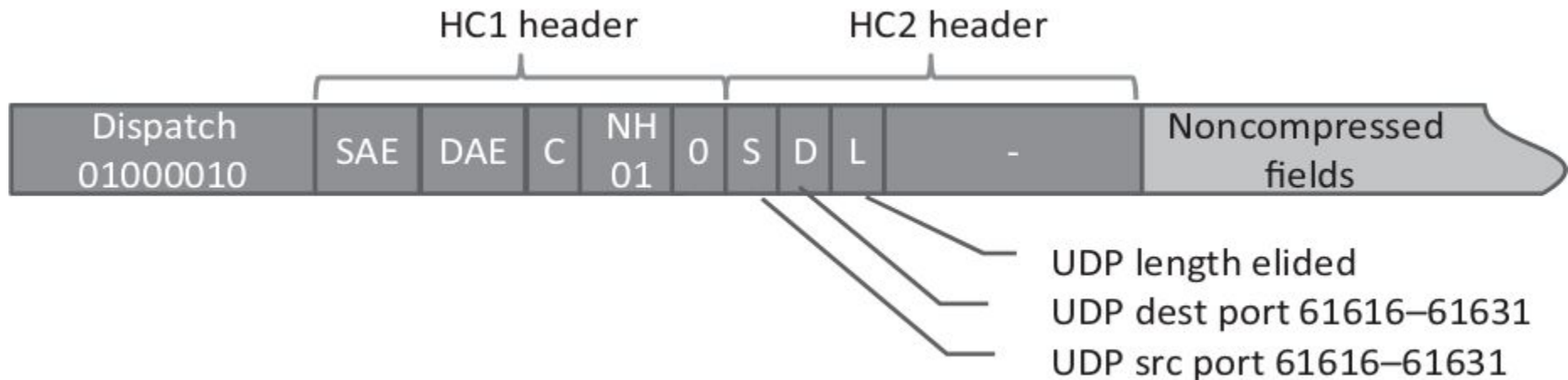


- Only the hop limit remains unmodified
 - Plus the dispatch byte and the HC1 signature a total of 3 bytes

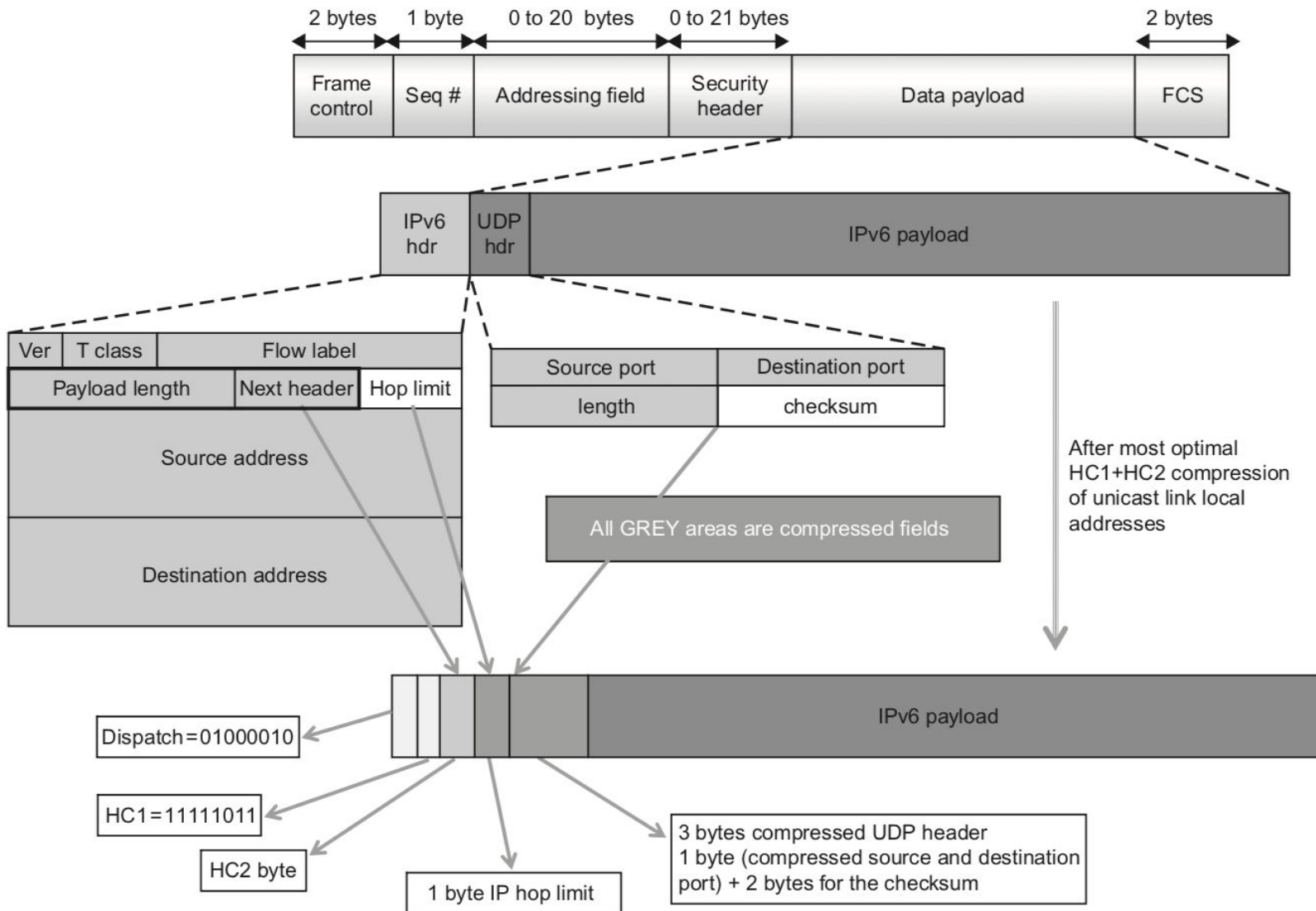


UDP compressed header (HC2)

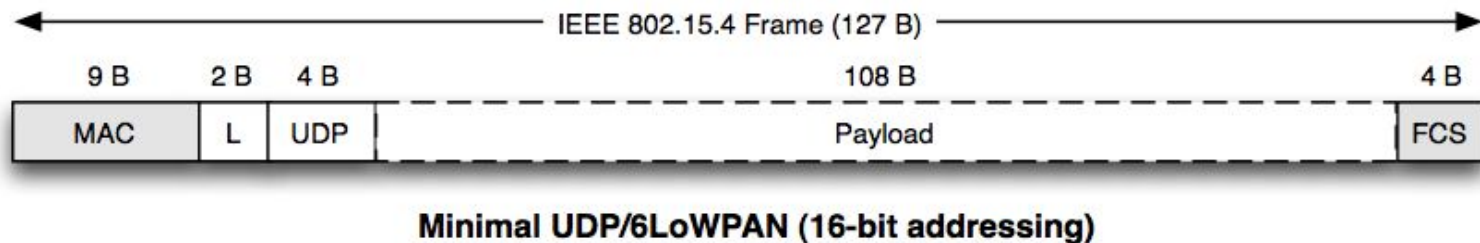
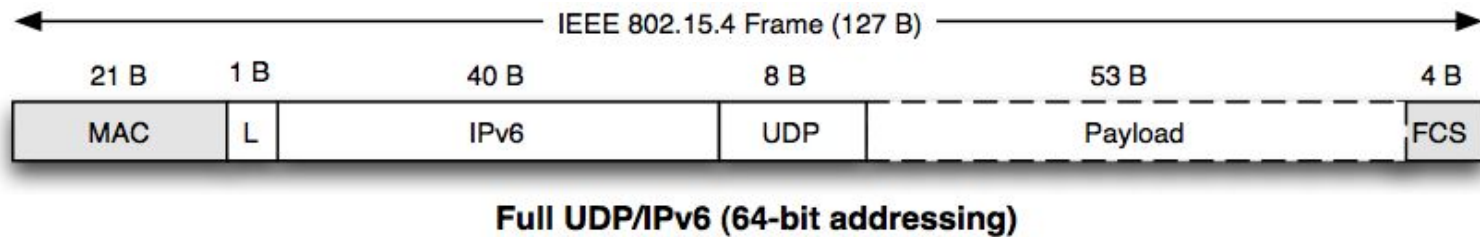
- S,D: 1 bit, indicate if the source and destination ports are in the range 62616 - 62631, and can then be encoded with only 4 bits
- L: 1 bit, indicates if the length field of the datagram has been removed



UDP datagram with HC1 and HC2



- Optimal compression for *unicast link-local* packets
 - From 48 bytes to 7 bytes (dispatch + 2 ip + 4 UDP)

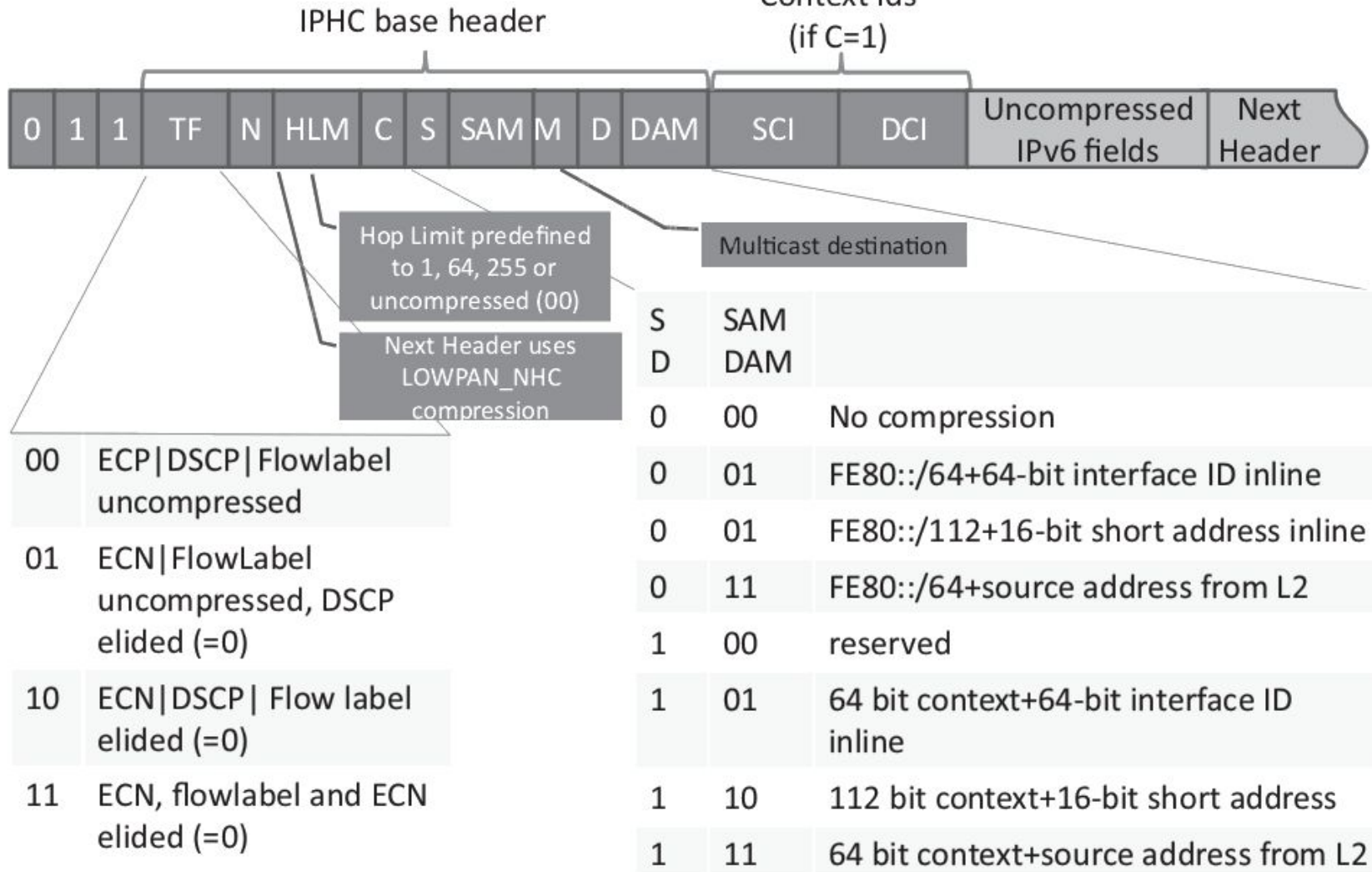


- Low effectiveness for Global Unicast addresses

ECN: Explicit Congestion Notification
DSCP: Differentiated Services Code Points

Optional source
and destination
Context Ids
(if C=1)

To index context
tables that store
network prefixes

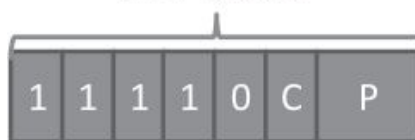


LOWPAN_NHC
base header for IPv6 extensions



EID	
000	IPv6 Hop by Hop options (RFC2460)
001	IPv6 routing (RFC2460)
010	IPv6 Fragment (RFC2460)
011	IPv6 destination options (RFC 2460)
100	IPv6 mobility header (RFC3775)
101	reserved
110	reserved
111	Nested IPv6 header

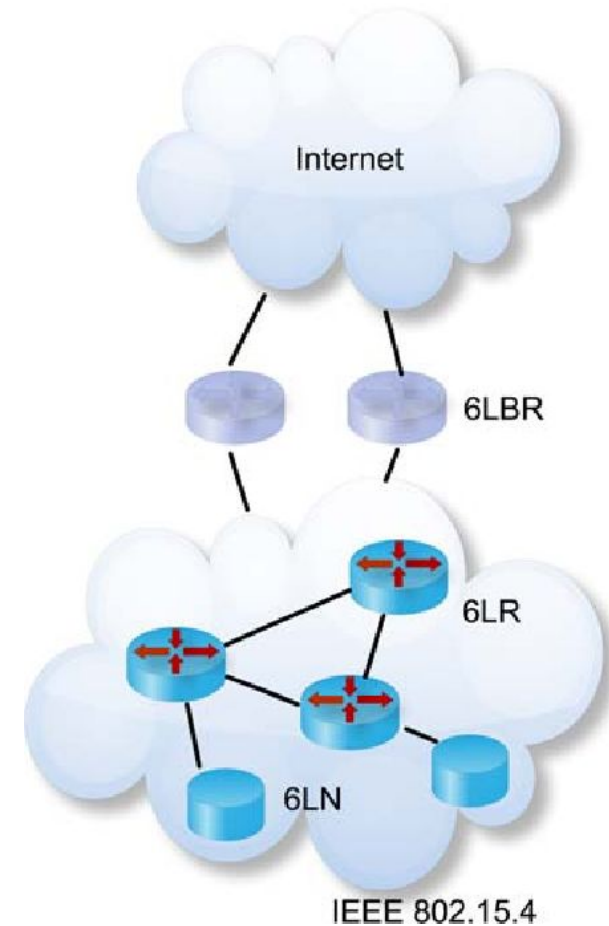
LOWPAN_NHC
UDP header



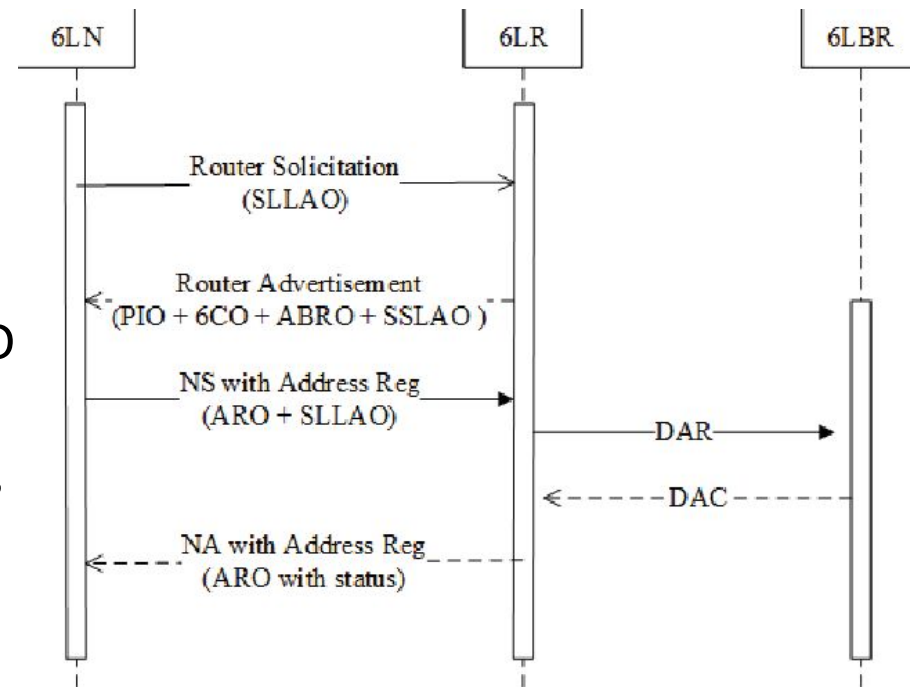
Compression of source (bit 1) and destination (bit 2) ports

UDP checksum elided

- RCFC 6775
- 6LoWPAN Border Router (6LBR)
 - Or Edge Router
 - Has the authority to establish the prefix
- 6LoWPAN Router (6LR)
 - Intermediate routers
 - Only in route-over
- 6LoWPAN Node (6LN)
 - The rest of the nodes



- No multicast for 6LN
 - The 6LN do not use the multicast solicited-node address
- Addresses are registered
 - Avoids the use of multicasts in ND
 - The 6LN can stay asleep
 - Short live time for mobile devices
- 6LR: cache the addresses
 - Route over: send requests to the 6LBR
 - Duplicate Address Request (DAR) and Confirmation (DAC)



SLLAO: Source Link Layer Address Option

ABRO: Authoritative Border Router Option

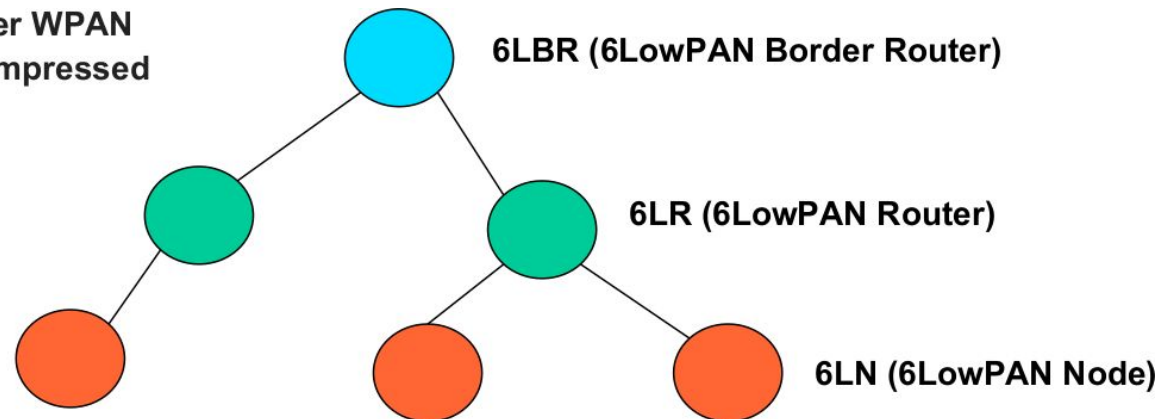
6CO: 6LoWPAN context options

PIO: Prefix information options

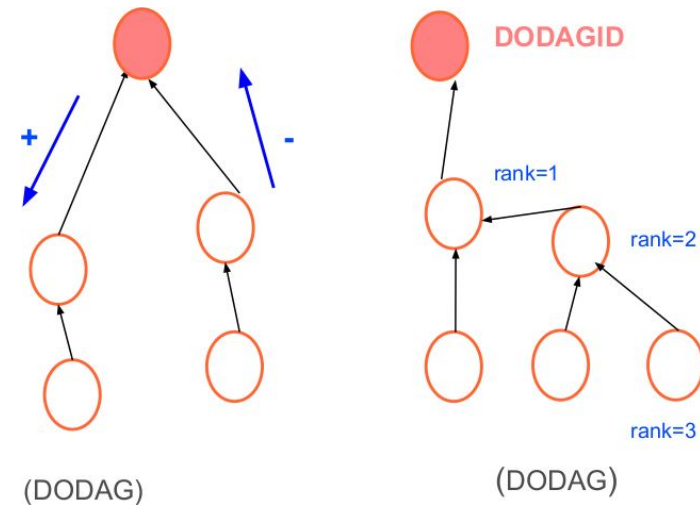
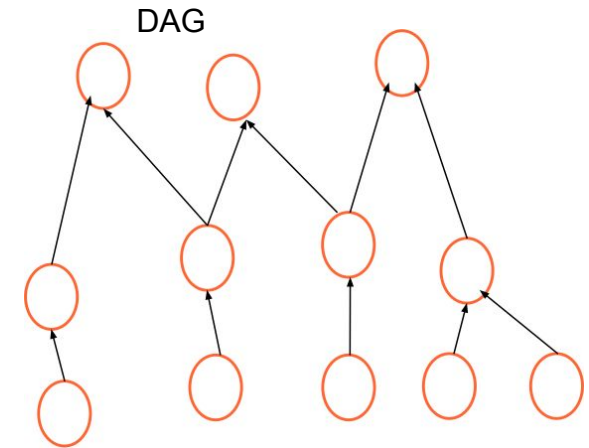
ARO: Address Registration Option

- IPv6 Routing Protocol for Low Power Lossy Networks
 - Specified by the IETF, [RFC6550](#)
- Distance Vector Routing Protocol
 - Each node maintains a table of distances to all the other nodes of the network
 - address of next hop
 - distance/cost
 - Simpler and less overhead than link state algorithms
- Intra domain routing
- Topological changes are notified to neighbors

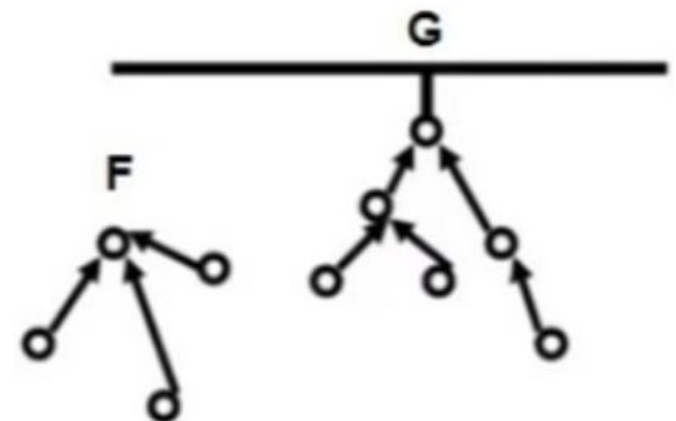
IPv6 over Low power WPAN
(6lowpan) - IPv6 compressed



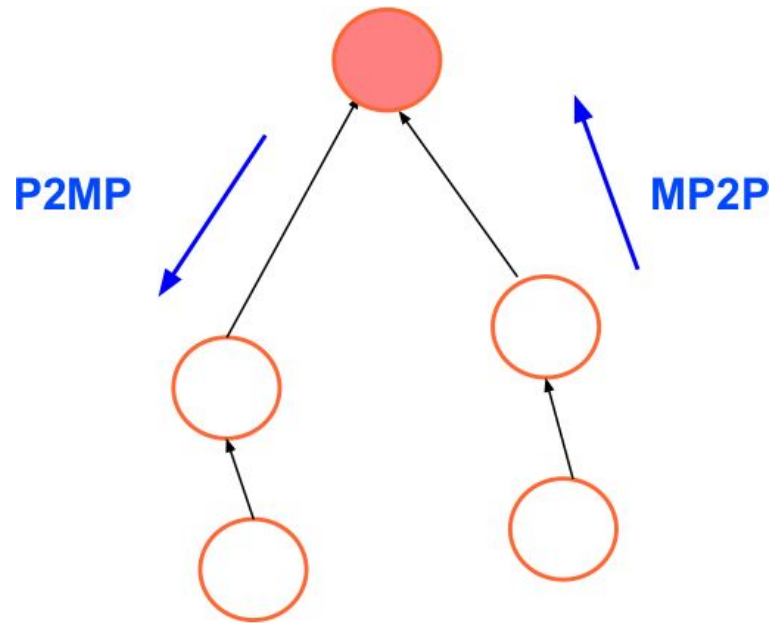
- **DAG: Acyclic Directed Graph**
 - Oriented edges
 - No cycles
 - A node can have more than one edge
- **DAG Root:**
 - A node with no output edges
- **DODAG: Destination Oriented DAG**
 - A DAG with only one root
- **DODAG Root:**
 - The root of the DODAG
 - Act as a border/edge router
 - Has a DODAGID
- **Up: upstream flow**
- **Down: downstream flow**
- **Rank: distance to the root**



- Objective function:
 - How to compute the rank
- RPLInstanceID: unique id for the RPL network
- RPL Instance:
 - A set of one or more DODAGs with the same RPLInstanceID and objective function
 - A node may belong to more than one DODAGs
- Storing Nodes:
 - store the complete routing table
- Non Storing Nodes:
 - Only store the parents
- Grounded DODAG:
 - Can reach the GOAL
- Floating DODAG:
 - Cannot reach the GOAL



- Three flows
 - Multi Point to Point (MP2P)
 - Point to Multi Point (P2MP)
 - Point to Point (P2P)
- RPL is optimised for MP2P

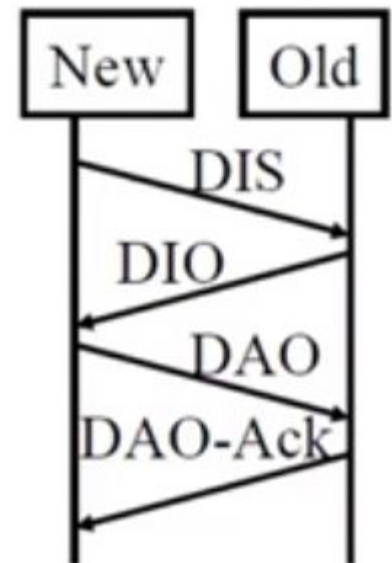


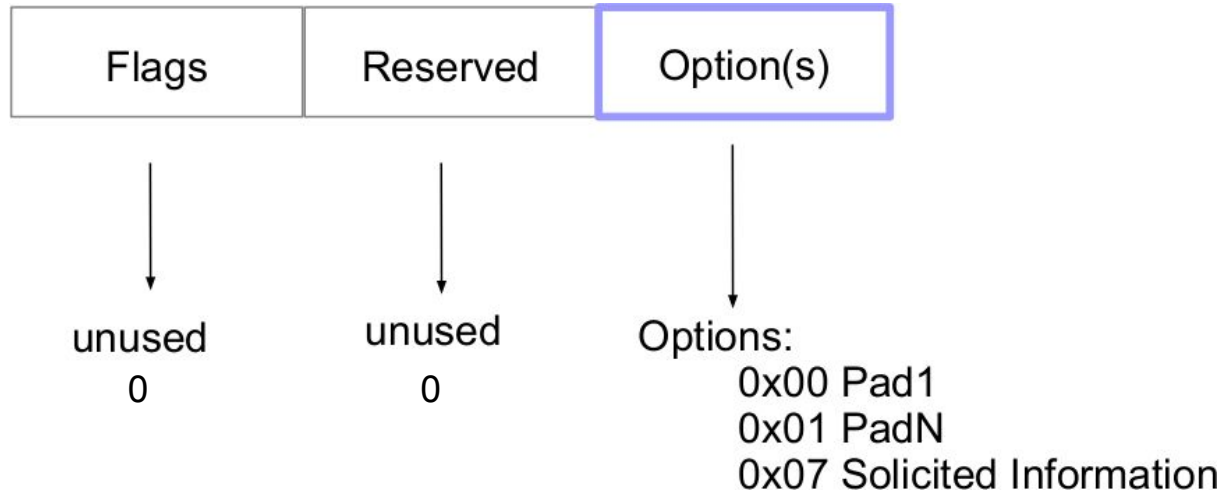
(DODAG)

- New ICMPv6 control packets

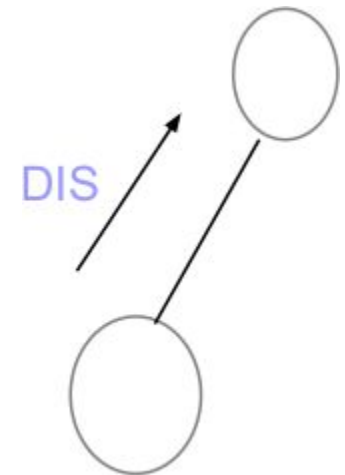


- Code identifies the type of message
 - 0x00: DODAG Information Solicitation (DIS)
 - 0x01: DODAG Information Object (DIO)
 - 0x02: Destination Advertisement Object (DAO)
 - 0x03: DAO-ACK





- Equivalent to the RS in IPv6 NDP
- Sent upstream to request a DODAG Information Object (DIO)



RPLInstanceID				Version Number		Rank	
G	0	MOP	Prf	DTSN		Flags	Reserved
DODAGID IPv6 address of the DODAG root							
Option(s)							

Field	Interpretation
G	1 - Grounded, 0 - Floating
Prf	DAGPreference 0-7, 0 lower preference
DTSN	Destination Advertisement Trigger Sequence Number, to maintain downward routes
Flags	no usado, deben ser 0
Reserved	no usado, deben ser 0

MOP	Operation Mode
0	Not maintain downward routes
1	Non Storing Mode
2	Storing Mode without multicast
3	Storing Mode with multicast
4-7	reservado

Type	Options
0x00	PAD1
0x01	PADN
0x02	DAG Metric Container
0x03	Routing Information
0x04	DODAG Configurariion
0x08	Prefix Information

Sent periodically by routers or as a response to a DIS

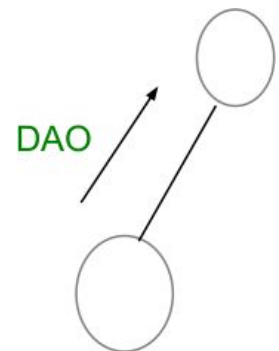
Sends downstream the DODAG Information

RPLInstanceID	K	D	Flags	Reserved	DAOSequence
<p>DODAGID</p> <p>IPv6 address of the DODAG root</p>					
<p>Option(s)</p>					

Field	Interpretation
K	Requiere DAO-ACK
D	DODAGID present
Flags	not used, 0
Reserved	not used, 0
DAOSequence	Sequence number copied on the DAO-ACK

Type	Options
0x00	PAD1
0x01	PADN
0x05	RPL Target
0x06	Transit Information
0x09	RPL Target Descriptor

- Used to send information upwards

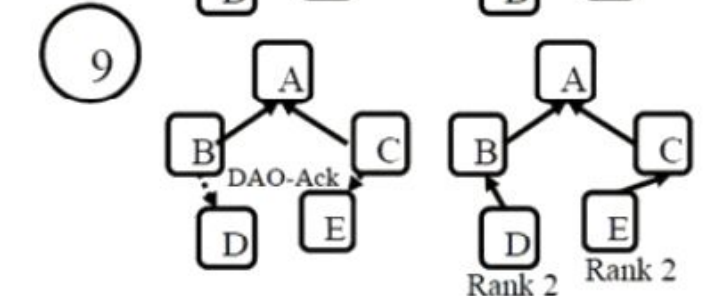
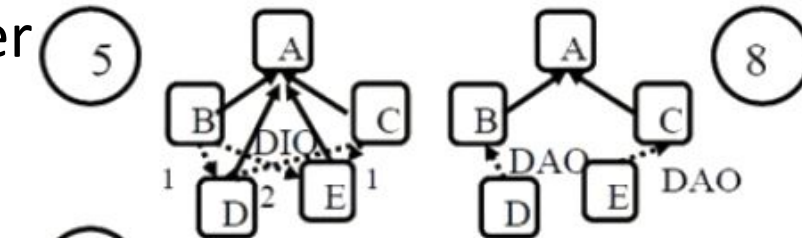
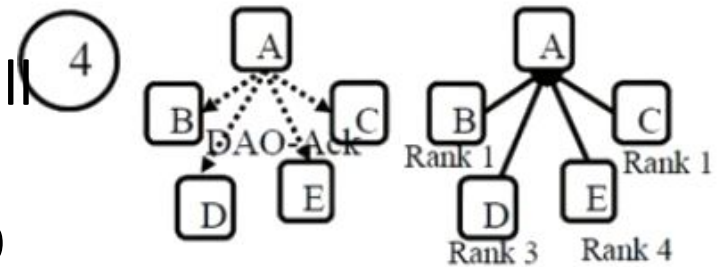
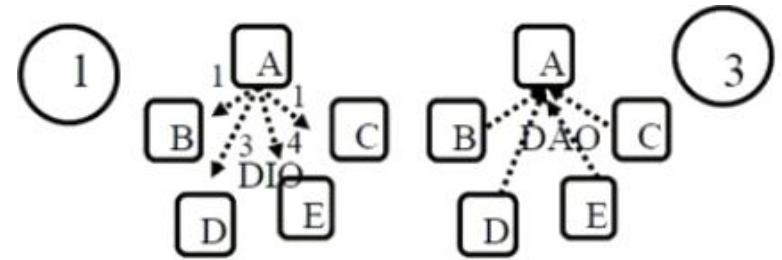


RPLInstanceID	D	Reserved	DAOSequence	Status
DODAGID IPv6 address of the DODAG root				
Option(s)				

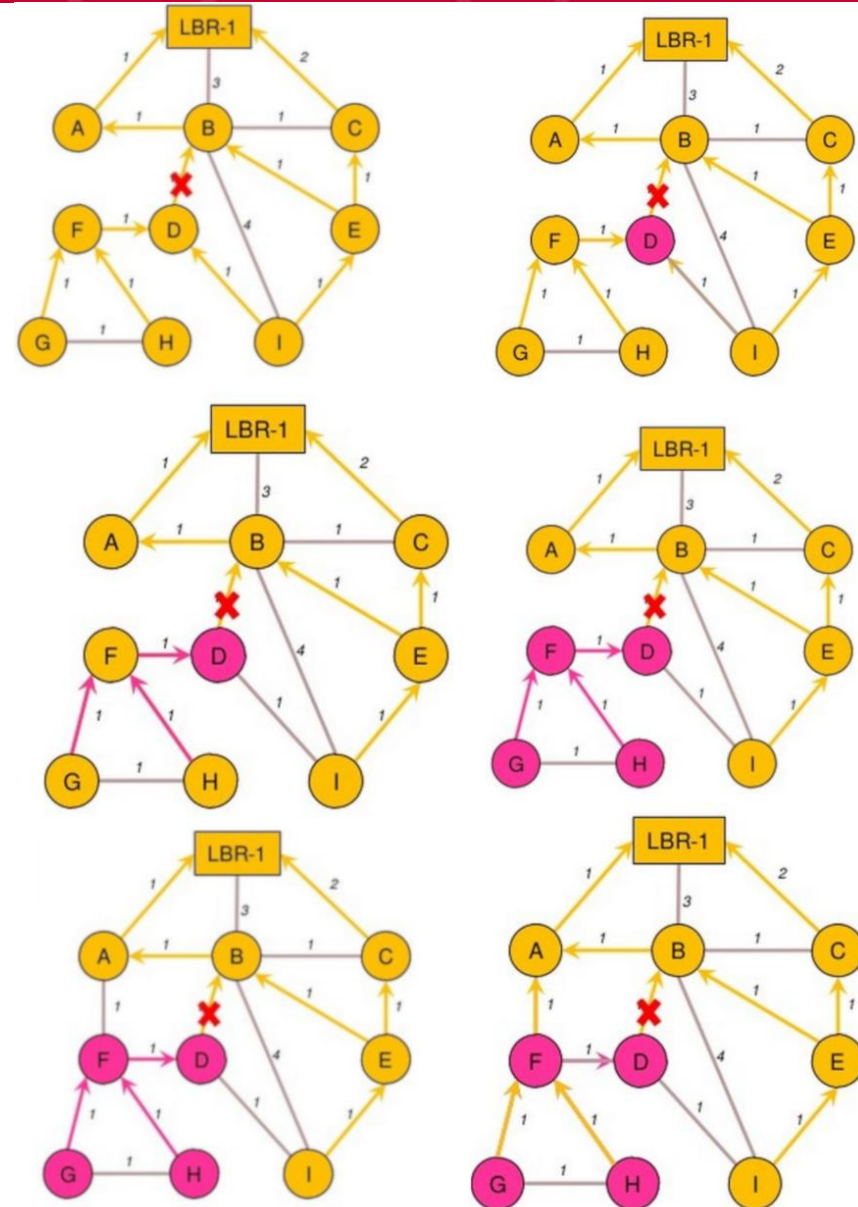
Status	Interpretation
0	Aceptado sin condiciones
1-127	Aceptado, pero se sugiere buscar un padre distinto
128-255	Rechazado, el nodo que envía el DAO-ACK rechaza actuar como padre

Sent as response to a DAO message

1. The DODAG Root (A) sends DIO
2. On reception the nodes compute the rank to A
3. The nodes send a DAO
4. El DODAG Root accepts confirms all with DAO-ACK (accepts)
5. Nodes start sending their own DIO
6. D observes that it will have a better rank through B
7. E observes the same with C
8. D and E send DAO messages to B and C
9. B and C confirm with a DAO-ACK (accept)



1. Link between D and B breaks
2. D sends a DIO to notify the changes in its sub-DAG
 - I remains in the DODAG through E
 - I removes D from the list of parents
3. F sends DIO to G and H
4. G and H join the floating DODAG
5. F receives a DIO from A
6. F sends DAO to A and joins again the DODAG LBR-1
 - Removes D from the list of parents



- You will use Cooja to simulate a 6LoWPAN network that uses RPL
 - <https://sourceforge.net/projects/contiki/files/Instant%20Contiki/>
 - # cd ~/contiki/tools/cooja/
 - # ant run
- You can analyze the network traffic with Wireshark by exporting PCAP files of the simulated traffic
 - You can eliminate duplicate packets with the edicap tool:
editcap -d original_file.cap filtered-output.cap
- To infer the DODAG topology you can
 - Filter the DIO and DAO messages
 - Activate the debug information in ~/contiki/core/net/rpl/rpl-dag.c and call call rpl_print_neighbor_list() from the rpl_process_dio() function
 - Filter UDP messages and analyse the 802.15.4 addresses to see the preferred routes for the UDP packets
 - Move/eliminate/add nodes and see how the RPL network converges

The screenshot shows the Cooja network simulator interface. The 'Tools' menu is open, with 'Radio messages...' selected. The 'Simulation control' window shows 'Run Speed limit' with buttons for 'Start', 'Pause', 'Step', and 'Reload'. The 'Notes' window is empty. The 'Mote output' window displays a table with columns 'Time', 'Mote', and 'Message'. The 'Timeline showing 11 notes' window is visible at the bottom, showing a list of notes numbered 1 through 5. The terminal window at the bottom shows the command `netstat -nr | awk '{ if ($2 == "tun0") print $1, $2, $3, $4, $5, $6, $7, $8, $9, $10, $11, $12, $13, $14, $15, $16, $17, $18, $19, $20, $21, $22, $23, $24, $25, $26, $27, $28, $29, $30, $31, $32, $33, $34, $35, $36, $37, $38, $39, $40, $41, $42, $43, $44, $45, $46, $47, $48, $49, $50, $51, $52, $53, $54, $55, $56, $57, $58, $59, $60, $61, $62, $63, $64, $65, $66, $67, $68, $69, $70, $71, $72, $73, $74, $75, $76, $77, $78, $79, $80, $81, $82, $83, $84, $85, $86, $87, $88, $89, $90, $91, $92, $93, $94, $95, $96, $97, $98, $99, $100 }'` and the output `/* Send no-path DAO only to preferred parent, if any */`. The terminal window also shows the command `user@instant-contiki:~/contiki/tools`. The system tray at the bottom shows the time as 1:24 PM and the date as Ln 834, Col 20.

Applications Places 1:24 PM

My simulation - Cooja: The Contiki Network Simulator

File Simulation Notes Tools Settings Help

Network Simulation control Notes

View Zoom

Run Speed limit

Start Pause Step Reload

Time: 00:00.000
Speed: ---

Enter notes here

Mote output

File Edit View

Time Mote Message

Filter:

Radio messages: showing 0/0 packets

File Edit Analyzer View

- No. No Analyzer
- 6LoWPAN Analyzer
- ◆ 6LoWPAN Analyzer with PCAP

Timeline showing 11 notes

File Edit View Zoom Events Notes

```

1
2
3
4
5
netstat -nr | awk '{ if ($2 == "tun0") { print $1, $2, $3, $4, $5, $6, $7, $8, $9, $10, $11, $12, $13, $14, $15, $16, $17, $18, $19, $20, $21, $22, $23, $24, $25, $26, $27, $28, $29, $30, $31, $32, $33, $34, $35, $36, $37, $38, $39, $40, $41, $42, $43, $44, $45, $46, $47, $48, $49, $50, $51, $52, $53, $54, $55, $56, $57, $58, $59, $60, $61, $62, $63, $64, $65, $66, $67, $68, $69, $70, $71, $72, $73, $74, $75, $76, $77, $78, $79, $80, $81, $82, $83, $84, $85, $86, $87, $88, $89, $90, $91, $92, $93, $94, $95, $96, $97, $98, $99, $100 } }'
user@instant-contiki:~/contiki/tools
/* Send no-path DAO only to preferred parent, if any */

```

C Tab Width: 8 Ln 834, Col 20 INS

[user@instant... My simulation... user@instant... [user@instant... rpl-daq.c (~/.c... [radiolog-153... build ContikiRPL - ...

Applications Places

radiolog-1539180843814.pcap [Wireshark 1.7.2 (SVN Rev 42506 from /trunk)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Wireshark: Open Capture File

Filter: icmpv6.type == 155

home user contiki tools cooja build

No.	Time	Source
29592	255.084000	fe80::
29593	255.084000	fe80::
29594	255.085000	fe80::
29595	255.085000	fe80::
29596	255.085000	fe80::
29597	255.085000	fe80::
29598	255.086000	fe80::
29599	255.119000	fe80::
29600	255.119000	fe80::
29601	255.120000	fe80::
29602	255.120000	fe80::
29603	255.120000	fe80::
29604	255.120000	fe80::
29605	255.121000	fe80::
29607	255.160000	fe80::
29699	448.808000	fe80::
29700	448.833000	fe80::
29701	448.833000	fe80::

ICMPv6 RPL Option (RPL)
Type: RPL Target (5)
Length: 18
Reserved
Target Length: 128
Target: aaaa::c30c:0:

Hex	ASCII
0000	61 dc 24 cd ab 01 0
0010	00 00 00 0c c1 7a 3
0020	aa aa 00 00 00 00 0
0030	05 12 00 80 aa aa 0
0040	00 00 00 08 06 04 0

Frame (76 bytes) 6LoWPAN

Frame (frame), 76 bytes

Name	Size	Modified
external_tools_unix_04.config	98 bytes	10/01/2018
external_tools_macosx.config	2.2 kB	10/01/2018
external_tools_win32.config	1.2 kB	10/01/2018
log4j_config.xml	864 bytes	10/01/2018
quickhelp.txt	469 bytes	10/01/2018
radiolog-1538649111216.pcap	327.9 kB	10/04/2018
radiolog-1538657159714.pcap	199.2 kB	10/04/2018
radiolog-1538667350956.pcap	325.1 kB	10/04/2018
radiolog-1538759361083.pcap	542.8 kB	Saturday
radiolog-1538845172332.pcap	233.3 kB	Saturday
radiolog-1538846541178.pcap	2.0 MB	Saturday
radiolog-1539176005500.pcap	85.8 kB	Yesterday at 14:54
radiolog-1539176621614.pcap	517.7 kB	Yesterday at 15:04
radiolog-1539177130204.pcap	1.5 MB	Yesterday at 15:13
radiolog-1539180672554.pcap	545.5 kB	Yesterday at 16:11
radiolog-1539180843814.pcap	5.0 MB	Yesterday at 16:41
radiolog-1539182563304.pcap	1.5 MB	Yesterday at 16:45
radiolog-1539257068192.pcap	972.1 kB	13:26
radiolog-1539257670717.pcap	817.5 kB	13:35
radiolog-1539258387027.pcap	854.9 kB	13:47
test_template.c	3.4 kB	10/01/2018

Filter: Filename: radiolog-1539258387027.pcap

Enable MAC name resolution Format: Wireshark/tcpdump/... - libpcap

Enable network name resolution Size: 854866 bytes

Enable transport name resolution Packets: 9371

First Packet: 2018-10-11 13:46:27

Elapsed time: 00:01:04

Cancel Open