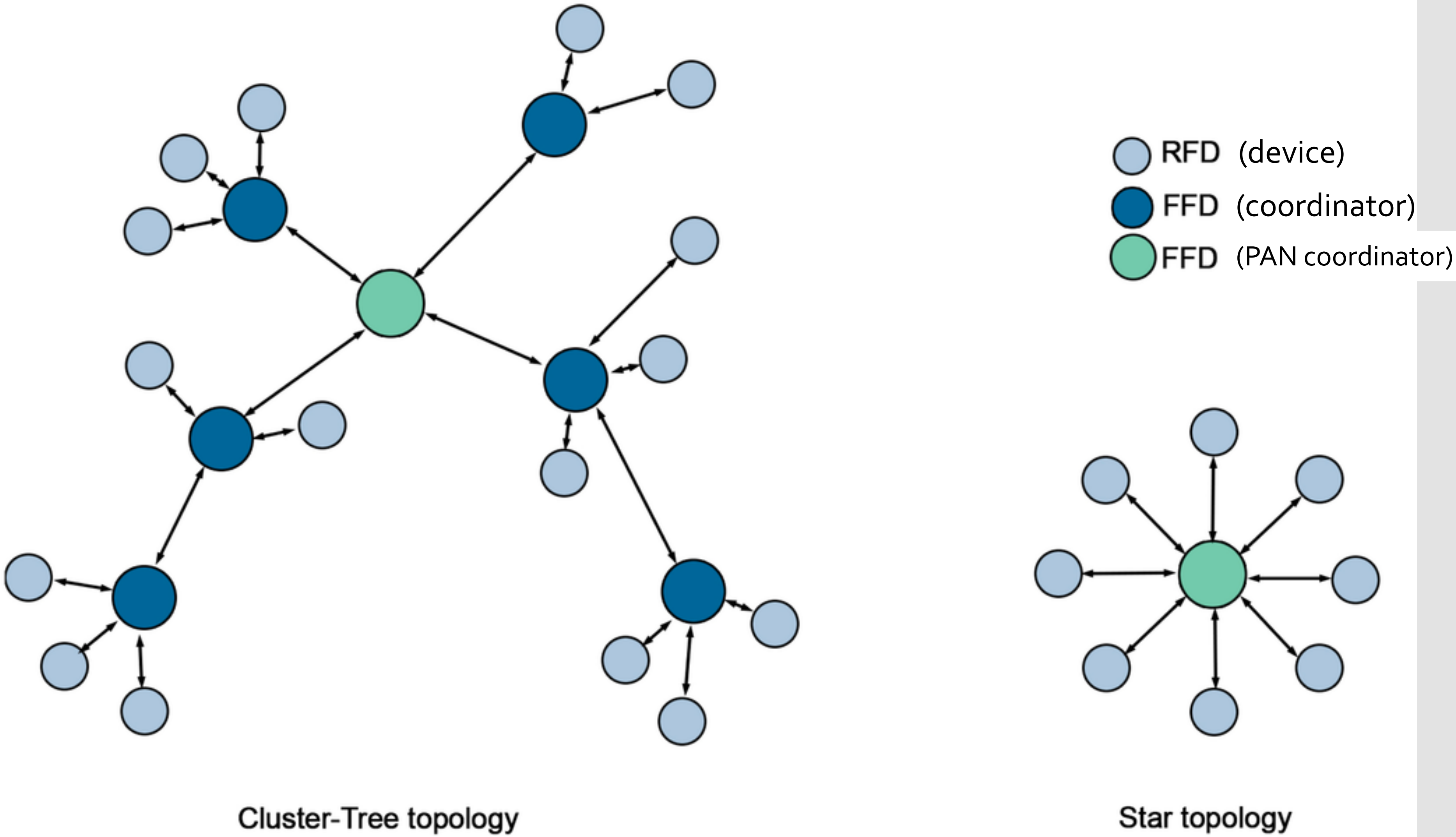
Review of previous lecture

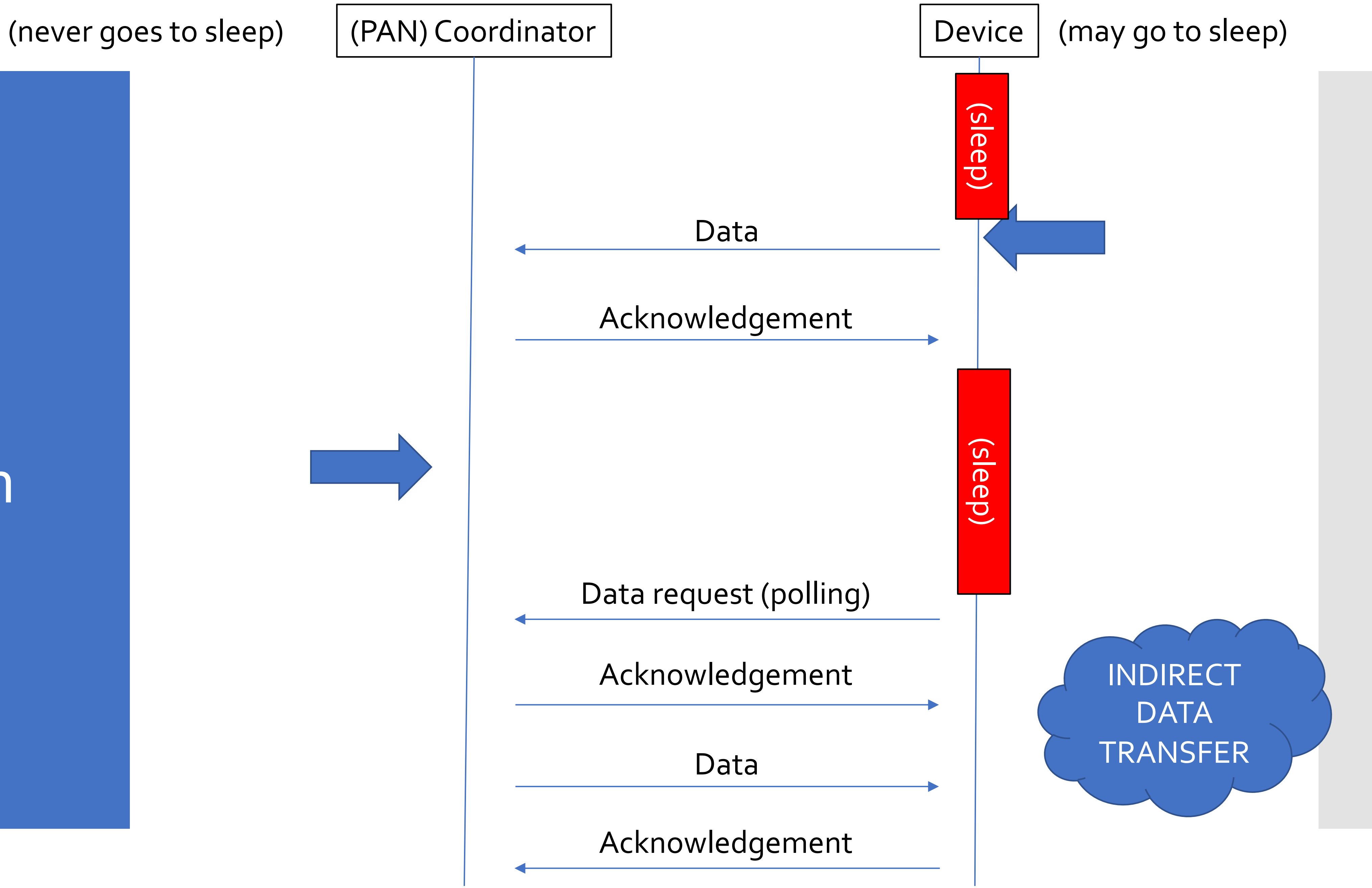
Low-Rate Wireless **Personal Area Network**

LR-WPAN

Personal Area Network (PAN) and supported topologies



Data TRx – Non-Beacon



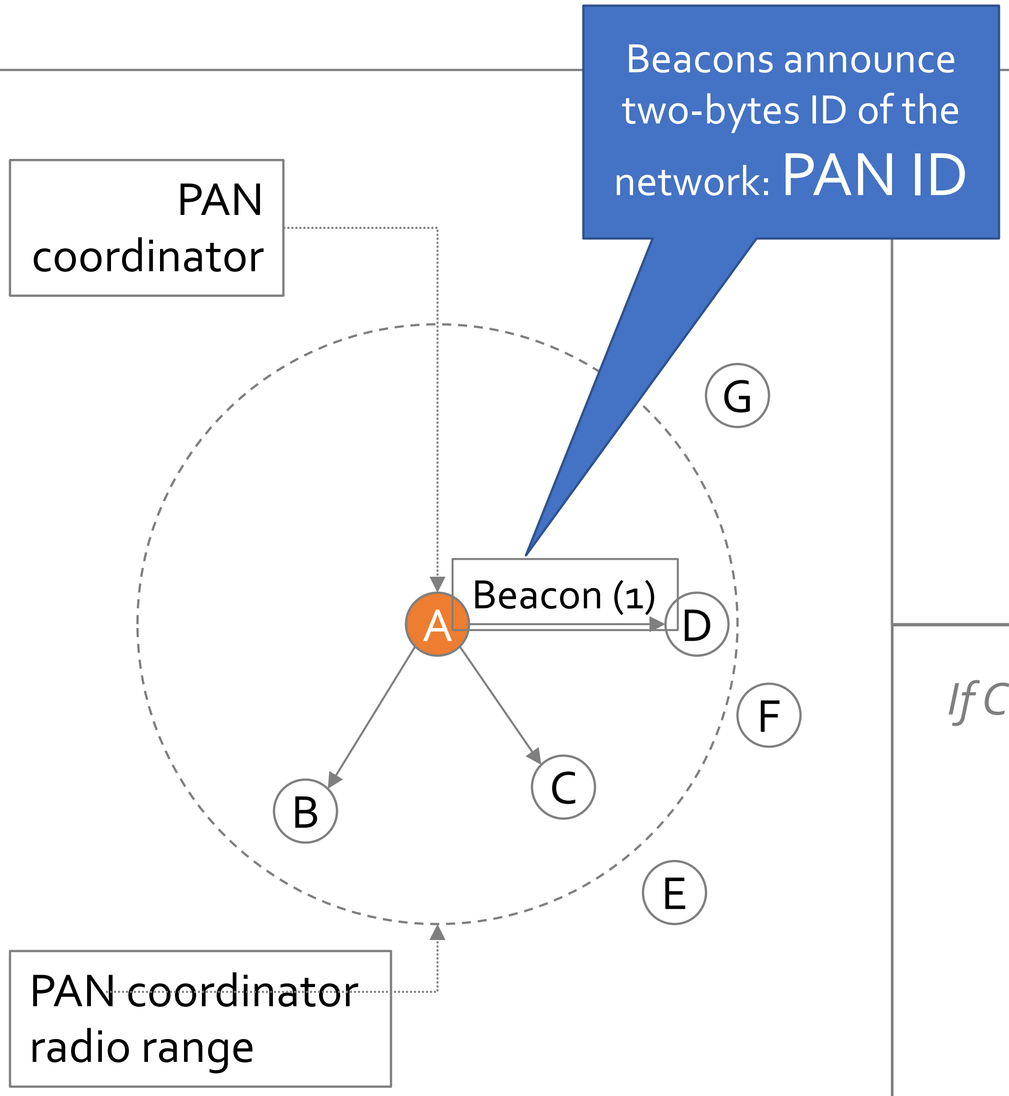
Addressing

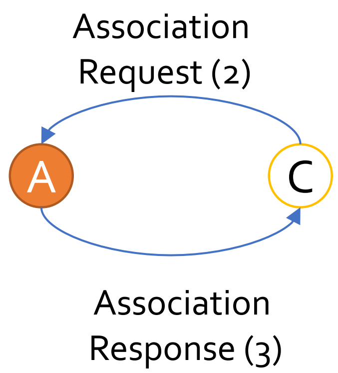
Layer			
IPv6	Network address	128-bits written in hex format in eight groups of 16-bits separated by ::	FE80::ABCD:1234:E6A8
802.15.4	Physical (MAC) address	Long: 64-bits (8 bytes) written in hex format, each byte separated by -	02-AB-4F-C9-00-AA-DE-AD
		Short: 16-bits (2 bytes) written in hex format, each byte separated by -	03-45
802.1 (Ethernet)	Physical (MAC) address	48-bits written in hex format, each byte seprated by :	ac:de:48:00:11:22

Association Control in LR-WPAN

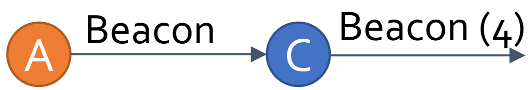
- Active/Passive Scan to discover Coordinators nearby
- Pass collected info of coordinators to higher-layer
- Higher-layer decides who we should “join” (associate)
- If beacon-enabled, synchronize first
- Possibility of requesting a 16-bit address to coordinator
- Send Association cmd and wait for Association Response (Indirect Data Transfer)

PAN formation



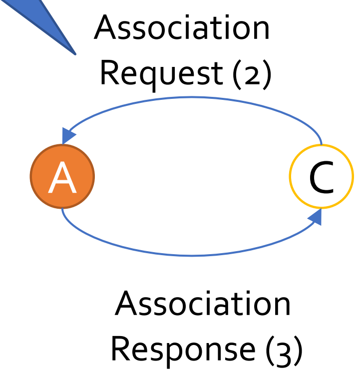


If C is a coordinator…



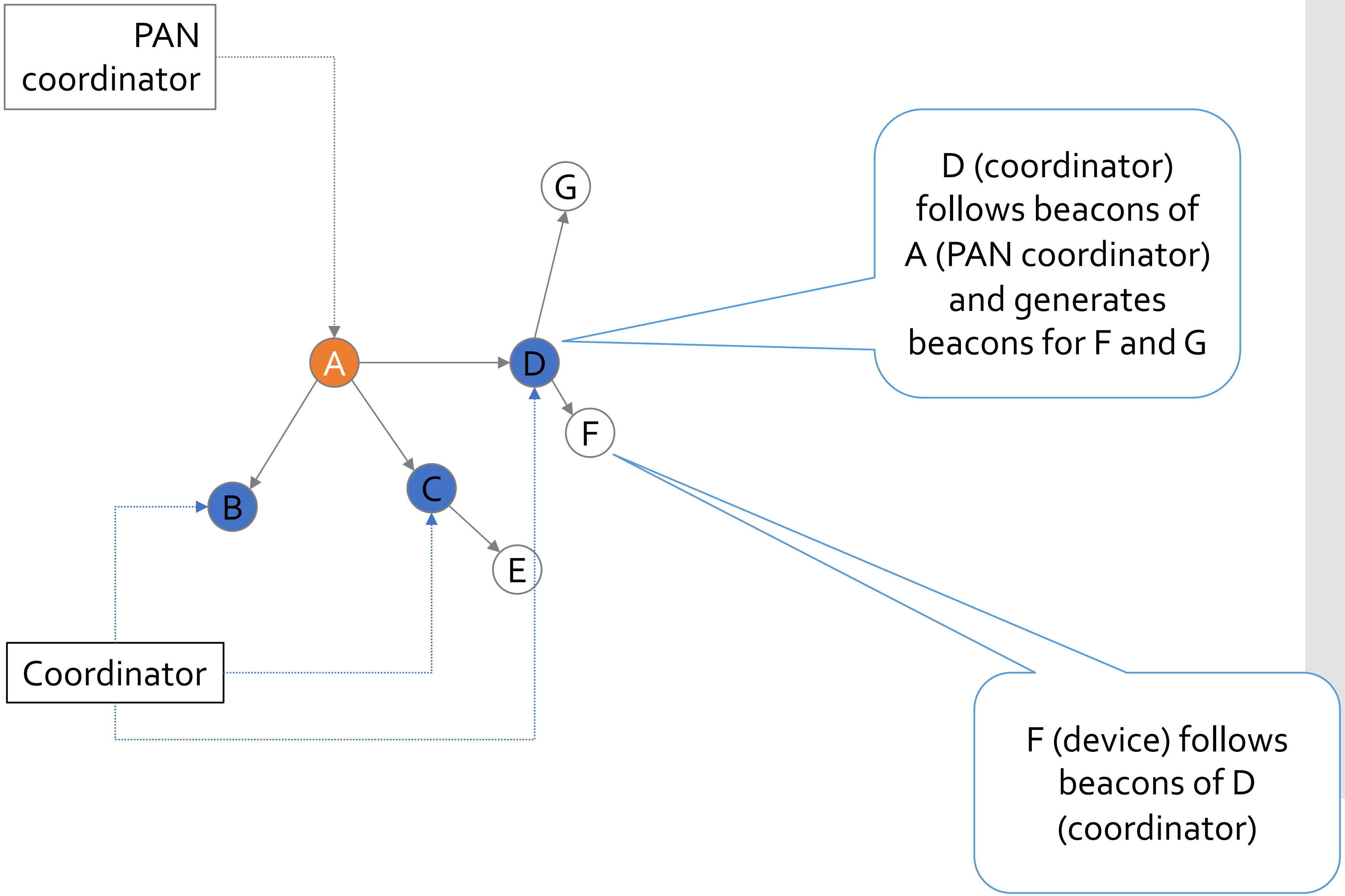
IEEE802.15.4 Short addresses

The device requesting an association can request as well a short address (2 bytes). Association Request is sent using a long address.



If short address is granted, further communications use the short address

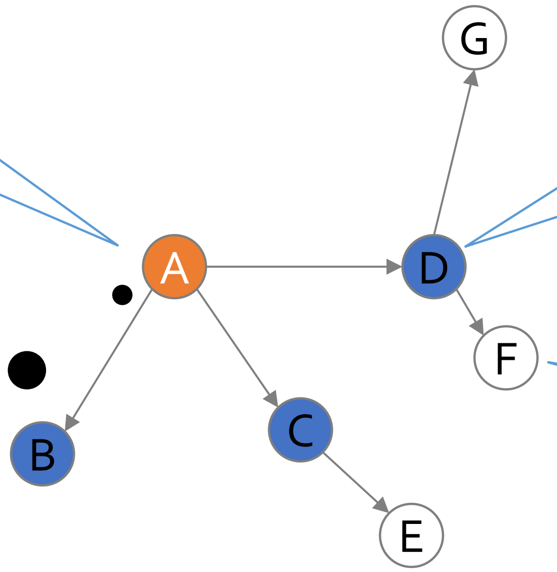
PAN formation



How to send data 2-hops away?

A reserves memory for indirect transmissions to B, C and D

A is aware only of the existence of B, C and D



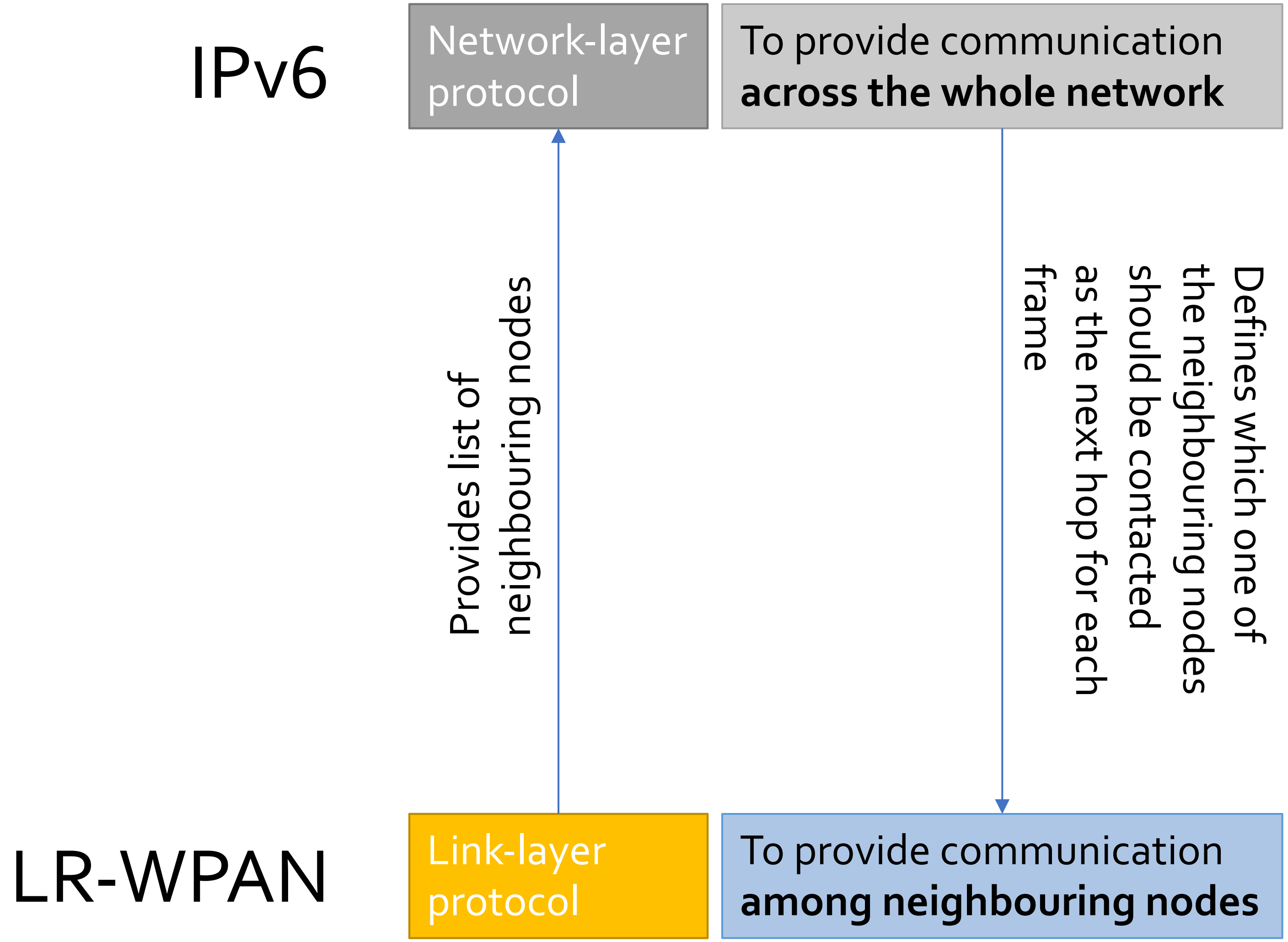
D reserves memory for indirect transmissions to F and G

D is aware only of the existence of A, F and G

F is aware only of the existence of D

If my application (A) would like to send a frame to F, what should I do?

L2 and L3 protocols (reminder)



Why network addresses?

- Link-layer addresses can be duplicated in the whole network (consider the internet)
- Link-layer addresses do not express any hierarchy
- Link-layer addresses could be too small as to be able to address all devices (consider 8 bytes as in LR-WPAN for all the internet)

IPv6 Addresses

From 32-bits to **128-bits addresses**

2000 : 0000 : 0000 : 0000 : 0217 : cbff : fe8c : 0000

8 hextets (8*16 = 128) separated by 7 colons

Hexadecimal format

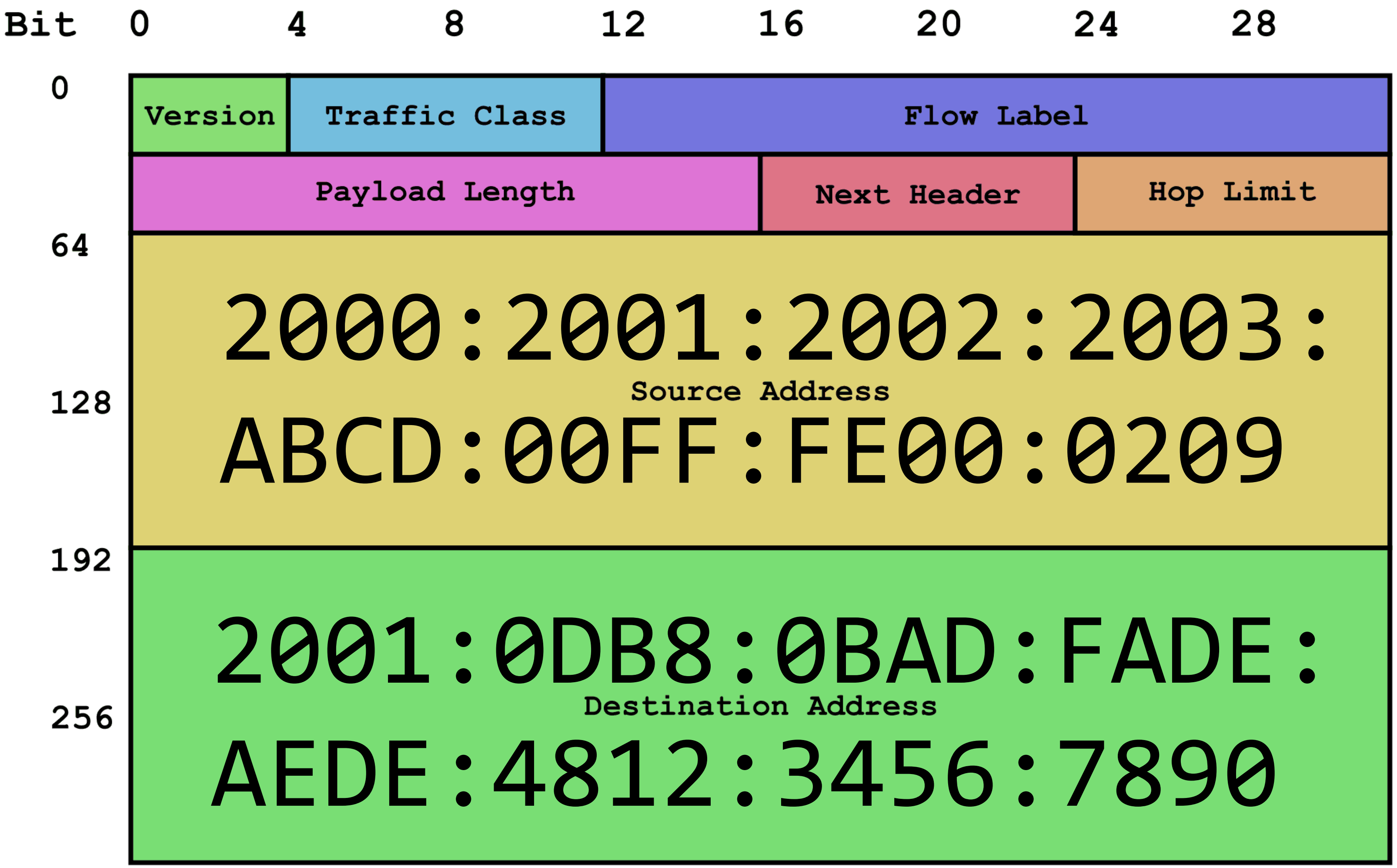
2000	0000	0000	0000	0217	cbff	fe8c	0000
0010 0000 0000 0000	0000 0000 0000 0000	0000 0000 0000 0000	0000 0000 0000 0000	0000 0010 0001 0111	1100 1011 1111 1111	1111 1110 1000 1100	0000 0000 0000 0000

One hextet is 16 bits

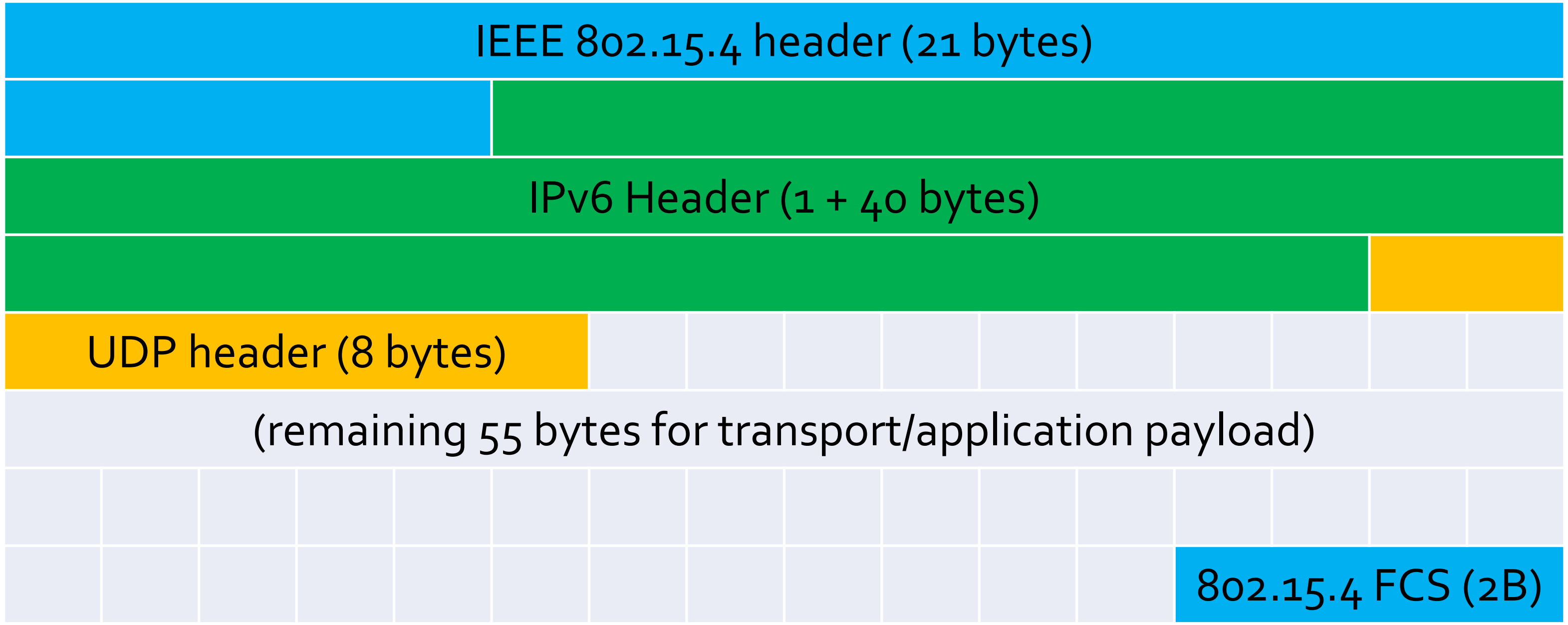
Binary representation of each hextet

3.4×10^{38}, viz., 340 undecillion, or **340** billion billion billion billion addresses (2^{128} addresses)

40 bytes for IPv6 header



Counting bytes all together



802.15.4 + IPv6 + UDP headers would take

57%

of the maximum allowed frame length!!

What's data compression?
How it is achieved?

6LoWPAN

Fernando Solano Donado

fs@tele.pw.edu.pl

Where?

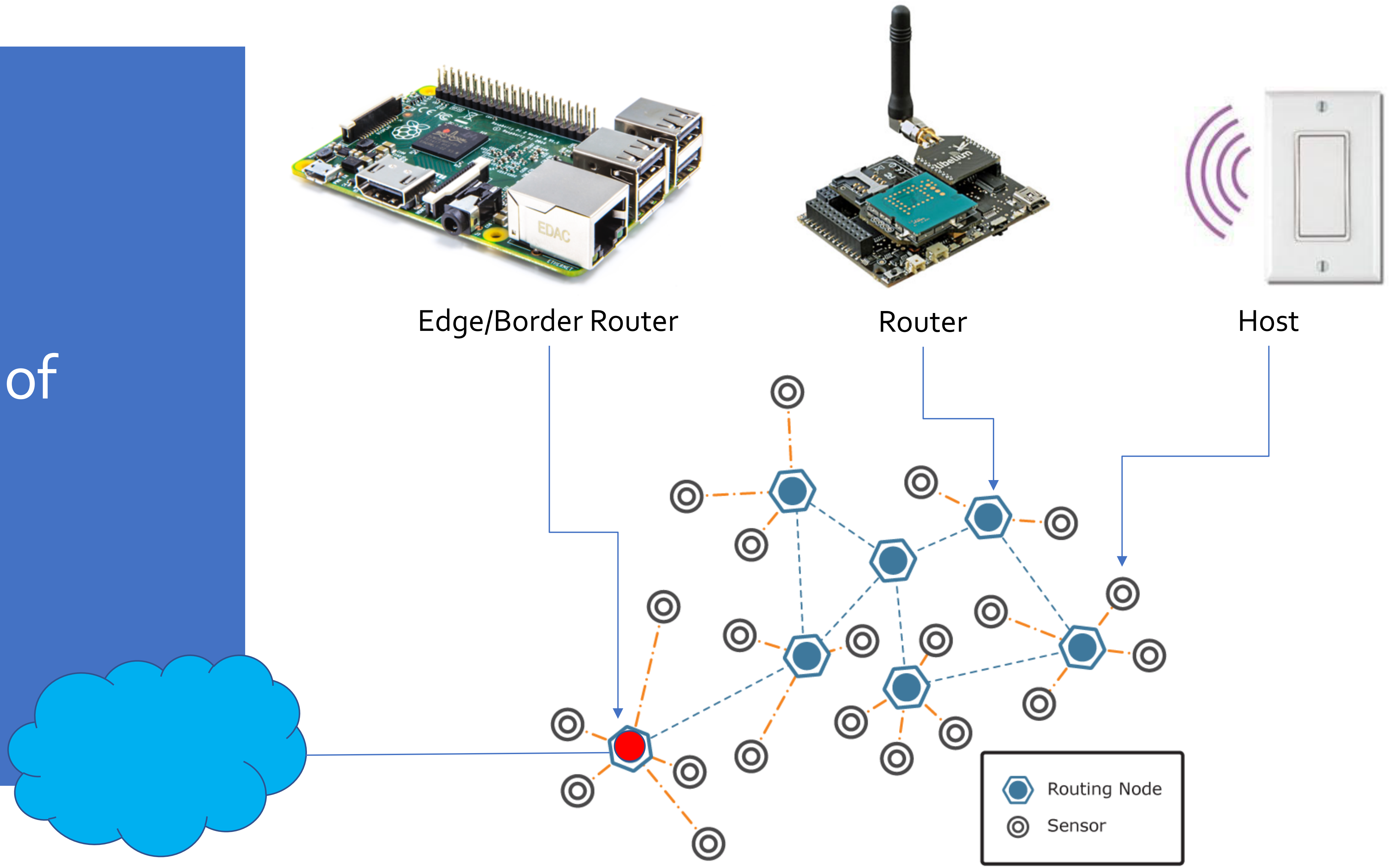
Good where:

- You need to connect your device to an external network
- You can afford some few bytes of code for making your application interoperable with the world at different net layers
- You are thinking about **IoT**

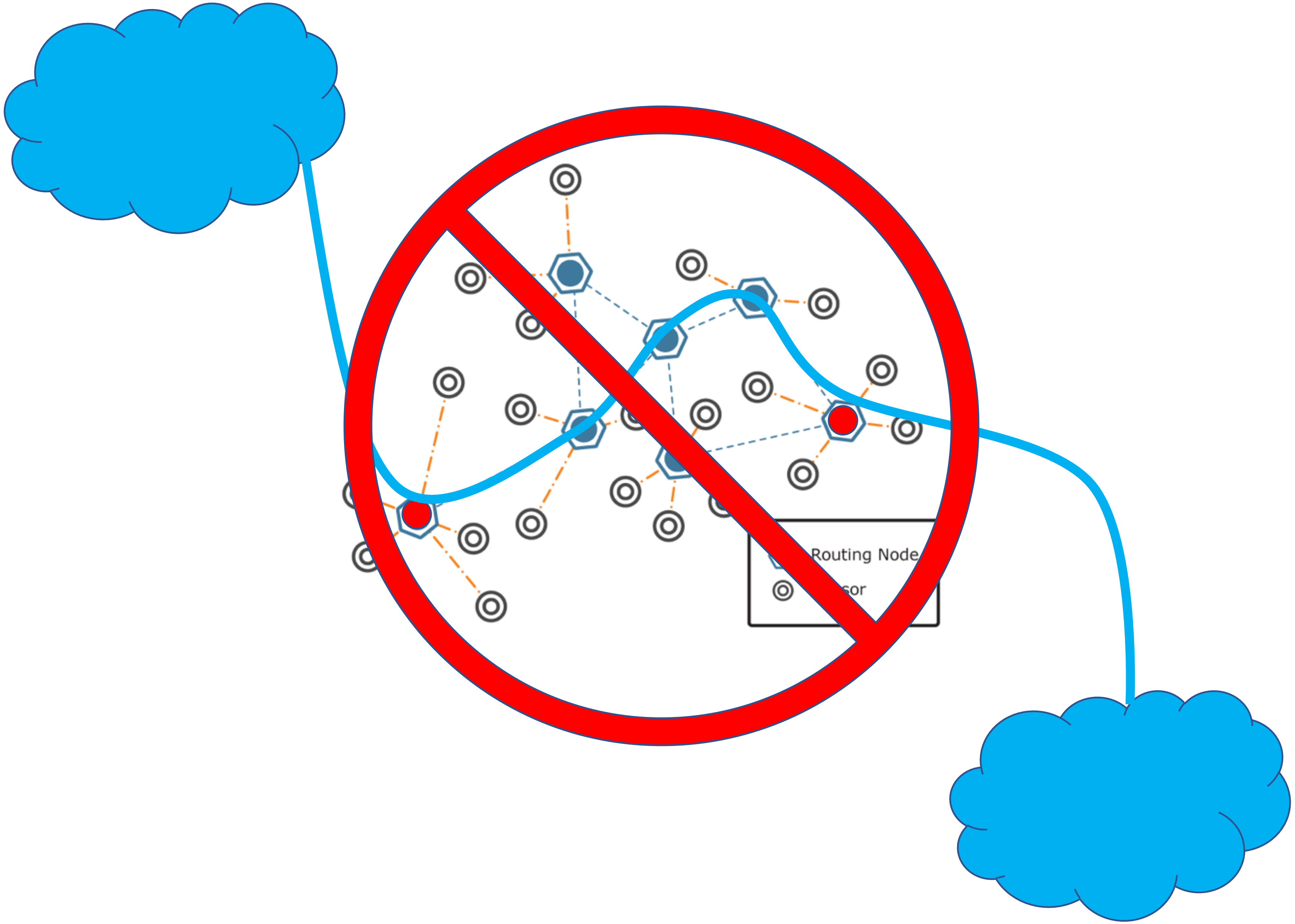
Unsuitable if:

- You are looking to an application that will be used in a close-system
- If memory for the application is extremely tiny
- You don't need a network of devices (e.g., only one link between two devices)

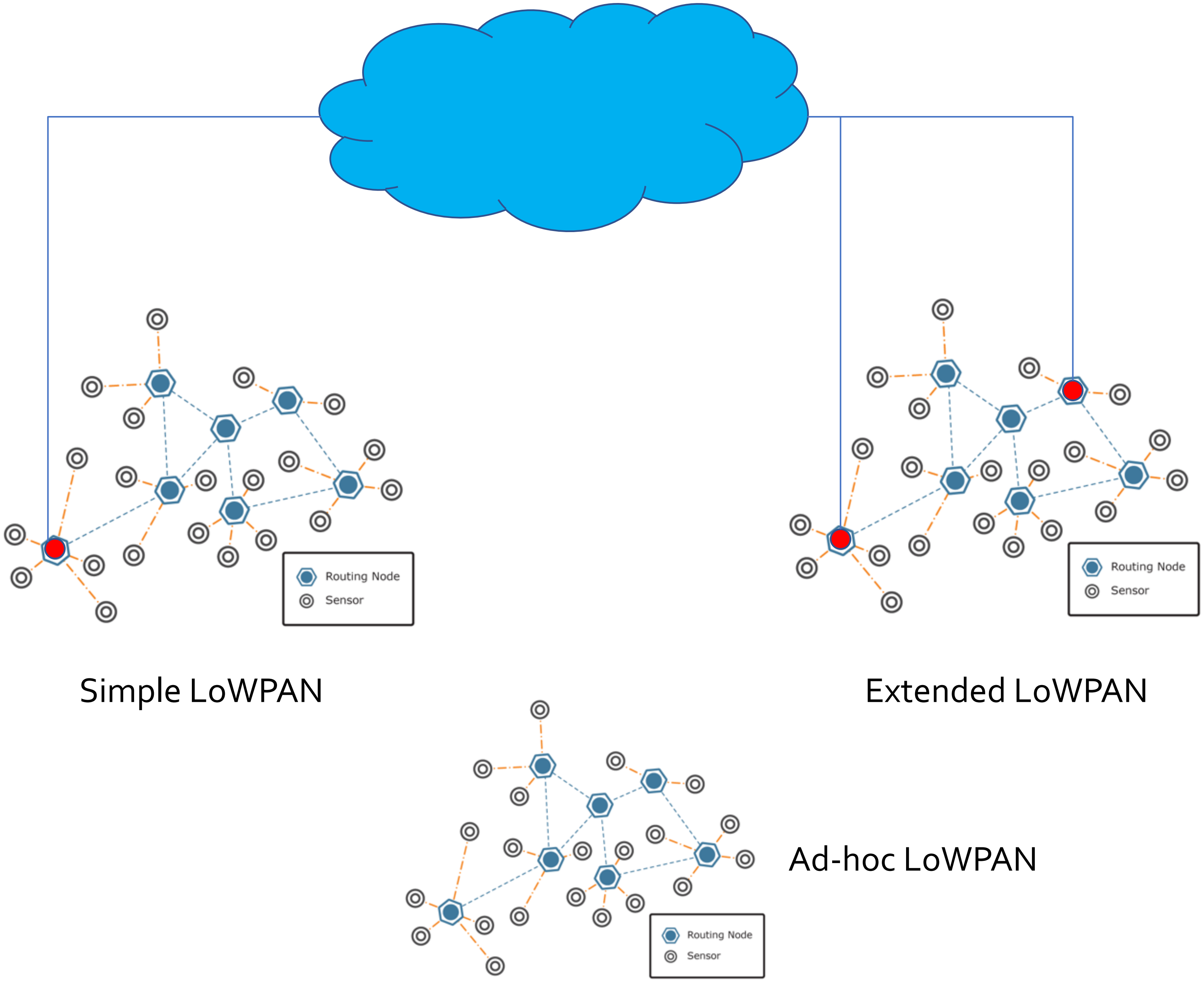
3 types of Nodes



Stub network



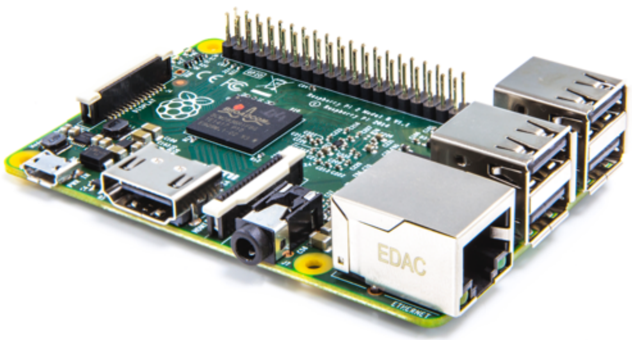
3 types of 6LoWPAN networks



Protocol stack

IP Protocol Stack		6LoWPAN
HTTP, RTP	(application)	Application protocols
TCP, UDP, ICMP	(transport)	UDP, ICMP
IPv6	(network)	IPv6
WiFi/Ethernet MAC	(data link)	6LoWPAN IEEE 802.15.4 MAC
WiFi/ Ethernet PHY	(physical)	IEEE 802.15.4 PHY

Protocol stack

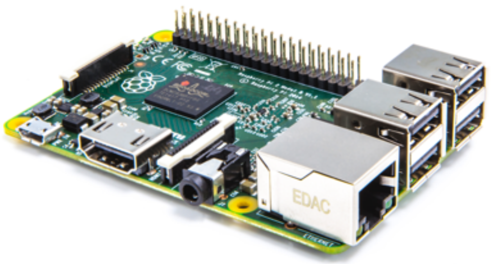


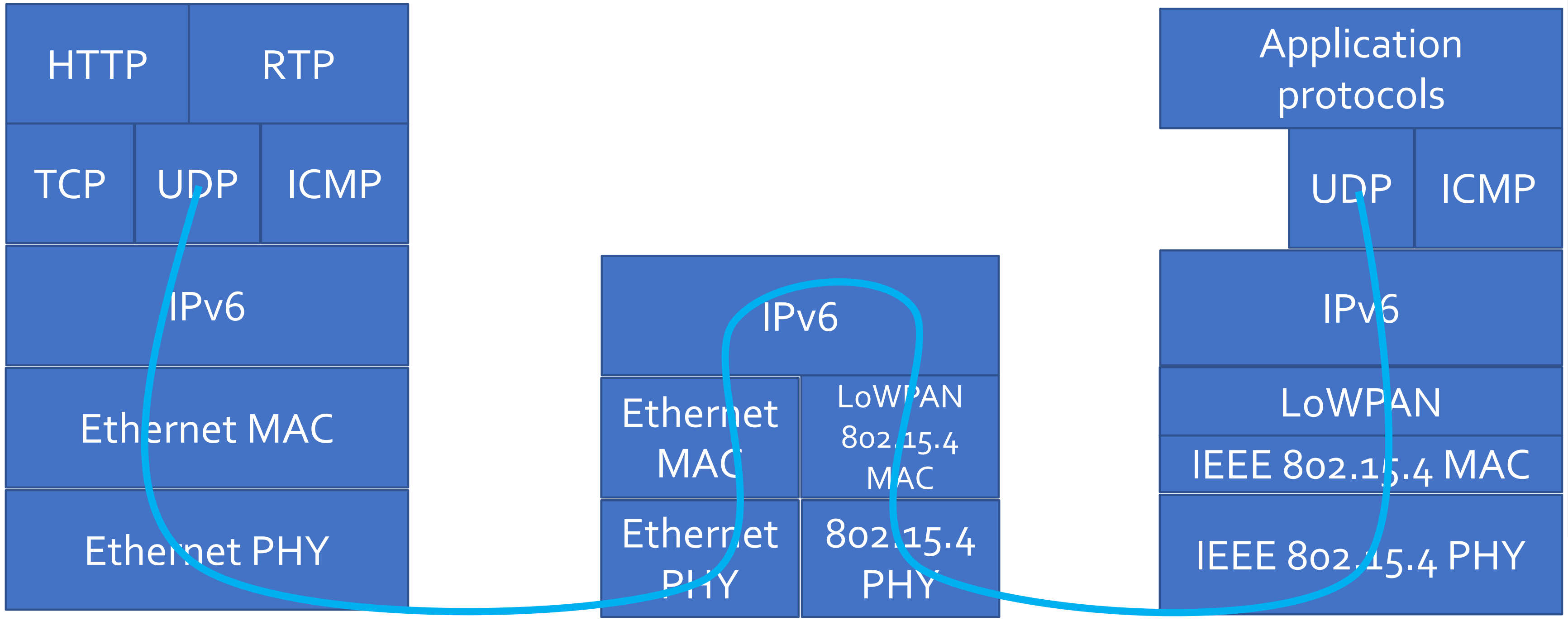
Border Router

(network)	IPv6	
(data link)	Ethernet MAC	6LoWPAN
		IEEE 802.15.4 MAC
(physical)	Ethernet PHY	IEEE 802.15.4 PHY

From smartphones to lightbulbs

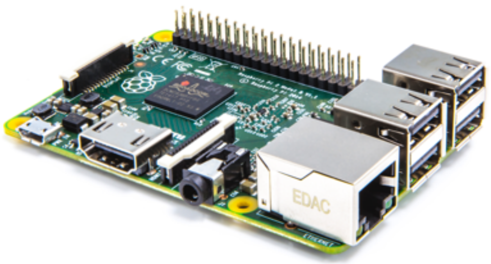
Edge Router

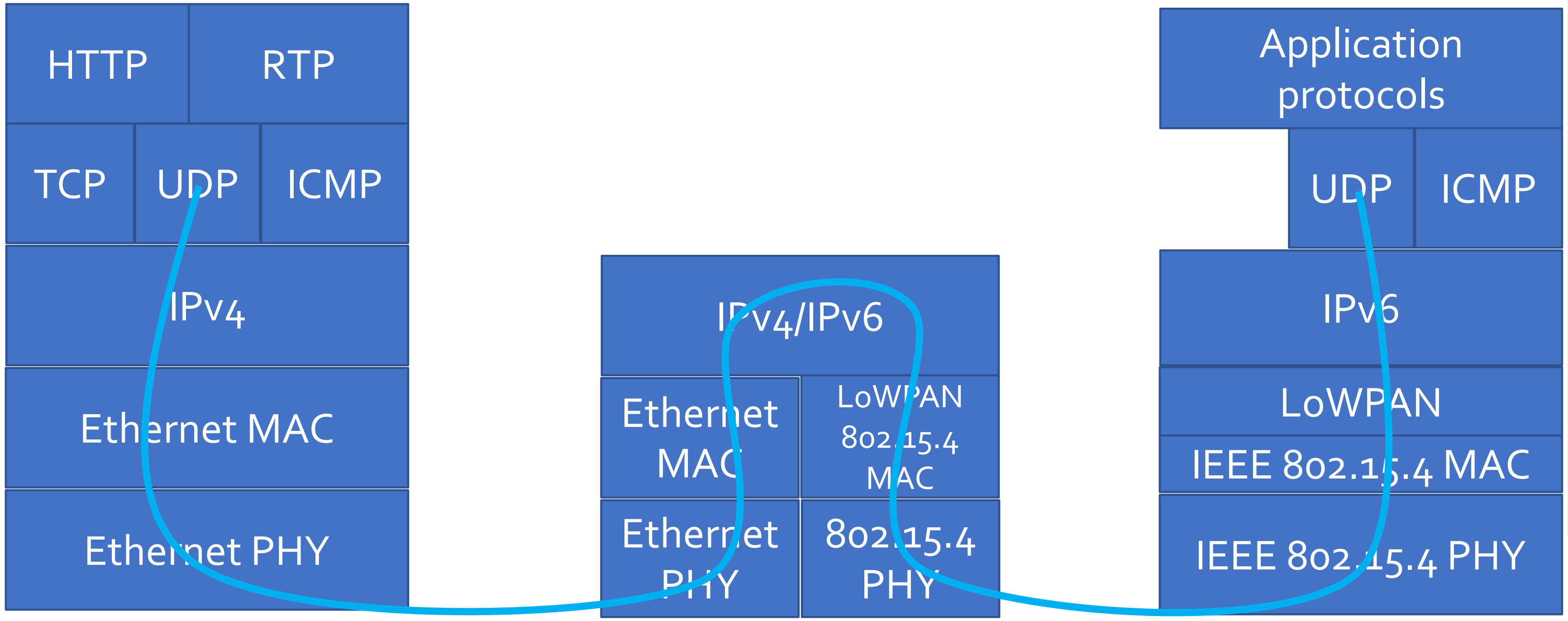




From smartphones to lightbulbs

Edge Router





StateLess Address AutoConfiguration in 6LoWPAN

Review of IPv6 and usage in 6LoWPAN

Prefix and Interface Id with Ethernet (reminder)

We are given a 64-bits network prefix - say 2000::/64

(network prefix)

2000	0000	0000	0000				

and a 48-bits Ethernet physical address - say 12-34-56-78-9A-BC

(MAC address)

01	23	45	67	89	AB

We insert FF-FE in the middle

(EUI-64)
Extended Unique Identifier

01	23	45	FF	FE	67	89	AB

Set to 1 (or flip) the 7th bit of the first byte

(Interface Id)

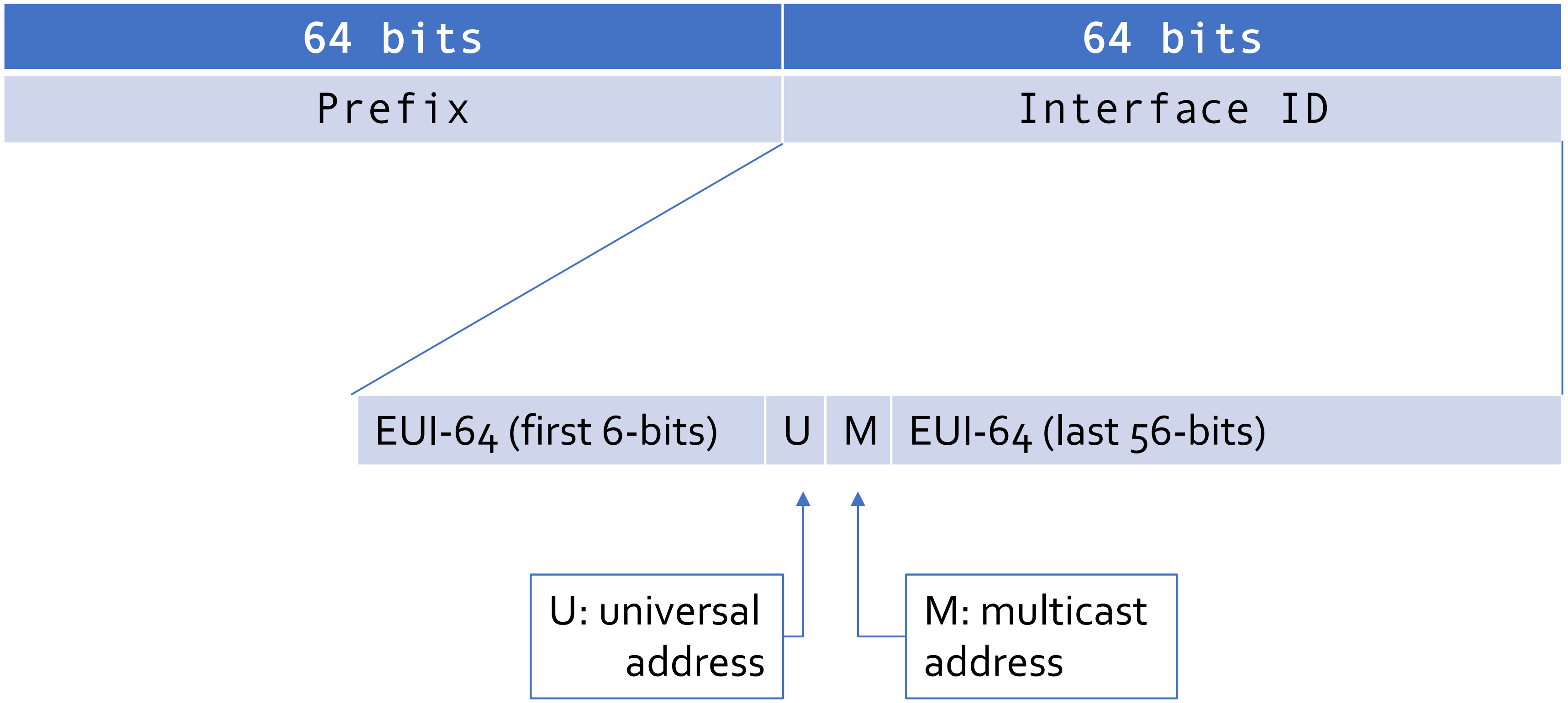
03	23	45	FF	FE	67	89	AB

Join the 64-bits prefix with the 64-bits interface id

2000	0000	0000	0000	0323	45FF	FE67	89AB

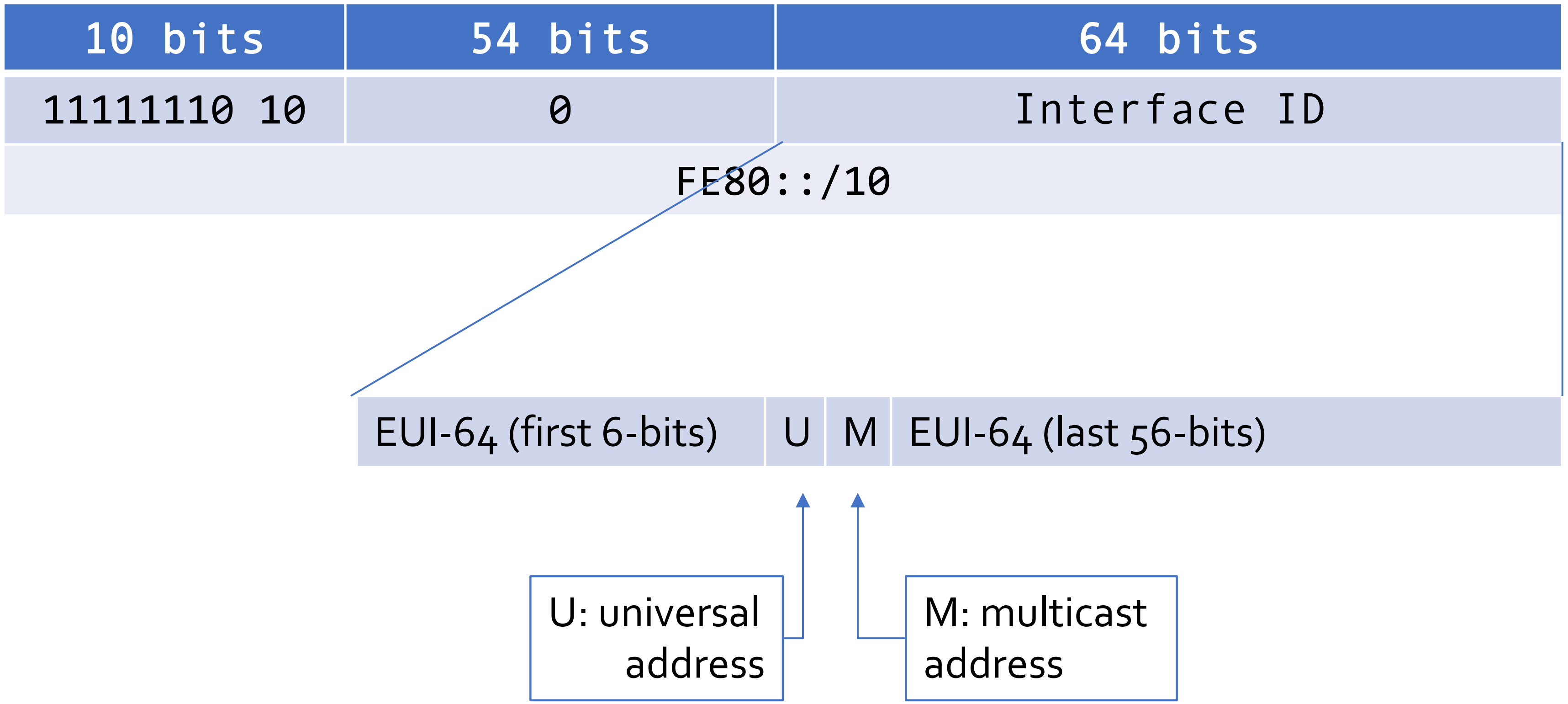
Which can be rewritten as 2000::0323:45FF:FE67:89AB

From IEEE802.15.4 long addresses to IPv6




```
2001:0DB8:0BAD:FADE::                         (prefix)
                    ACDE:4812:3456:7890   (EUI-64)
2001:0DB8:0BAD:FADE:AEDE:4812:3456:7890   (IPv6)
```

From IEEE802.15.4 long addresses to IPv6 local link




```
FE80:0000:0000:0000::                              (link local prefix)
                    ACDE:4812:3456:7890            (EUI-64)
FE80:0000:0000:0000:AEDE:4812:3456:7890            (IPv6)
```

From IEEE802.15.4 short addresses to IPv6

64 bits	64 bits
Prefix	Interface ID

EUI-64 (first 6-bits)	U	M	EUI-64 (last 56-bits)

EUI-64 =

PAN ID (16-bits)	00FF	FE00	Short address (16-bits)

```
2001:0DB8:0BAD:FADE::                         (prefix)
                    ACDE                      (PAN ID)
                                   7890       (short)
2001:0DB8:0BAD:FADE:AEDE:00FF:FE00:7890       (IPv6)
```

From IEEE802.15.4 short addresses to IPv6 link local

10 bits	54 bits	64 bits
11111110 10	0	Interface ID
FE80::/10		

EUI-64 (first 6-bits)	U	M	EUI-64 (last 56-bits)

EUI-64 =	PAN ID (16-bits)	00FF	FE00	Short address (16-bits)

```
FE80:0000:0000:0000::                      (prefix)
                     ACDE                  (PAN ID)
                                    7890   (short)
FE80:0000:0000:0000:AEDE:00FF:FE00:7890   (IPv6)
```

In RIOT

```
> ifconfig
Iface  9   HWaddr: 5d:39  Channel: hu  NID: 0x0
           Long HWaddr: 00:57:0b:00:5d:39:02:00
           TX-Power: 8319dBm  modulation: type hu 2kbps 10000khz State: IDLE
           Long HWaddr: 00:57:0b:00:5d:39:02:00
           MTU:1280  HL:64  6LO  RTR  IPHC
           Source address length: 8
           Link type: wireless
           inet6 addr: ff02::1/128  scope: local [multicast]
           inet6 addr: fe80::257:b00:5d39:200/64  scope: local
           inet6 addr: ff02::1:ff39:200/128  scope: local [multicast]
```

IPv6 Address Compression

Compression idea

Tonge twister

Betty Bo<u>tter</u> bought some bu<u>tter</u>
But she said the bu<u>tter</u>'s bitter
If I put it in my ba<u>tter</u>, it will make my batter bitter
But a bit of better butter will make my batter better
So 'twas better Betty Botter bought a bit of better butter

188 characters (no spaces)

Replacing 'tter' with '$'

Betty Bo$ bought some bu$
But she said the bu$'s bi$
If I put it in my ba$, it will make my ba$ bi$
But a bit of be$ bu$ will make my ba$ be$
So 'twas be$ Betty Bo$ bought a bit of be$ bu$

143 characters (no spaces)

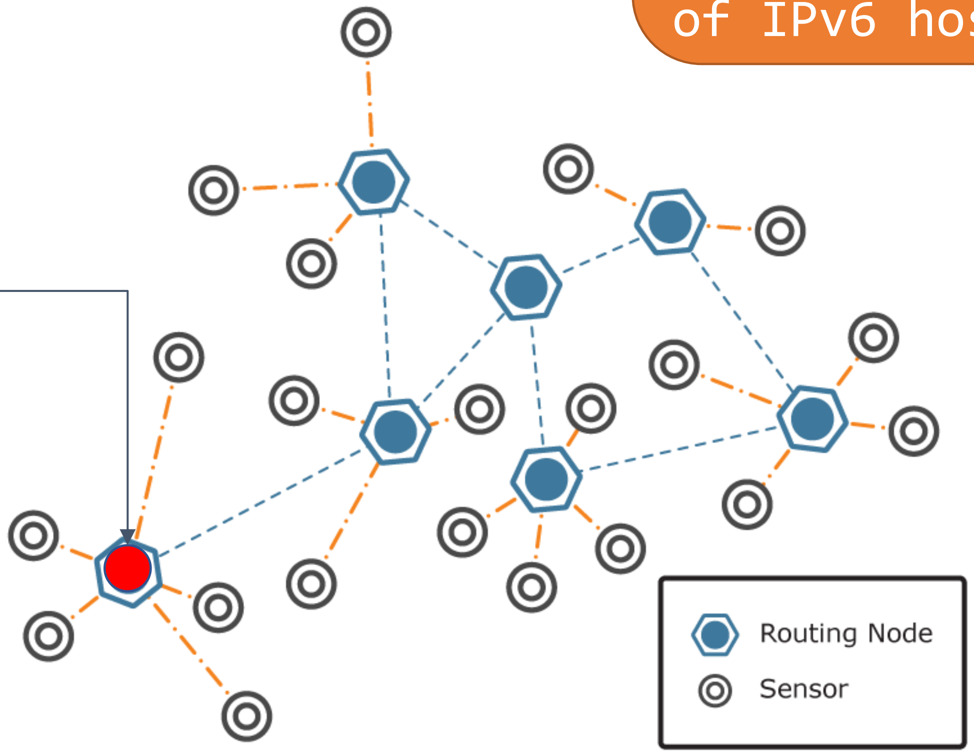
Where can it be applied?

Well-known IPv6 server address outside

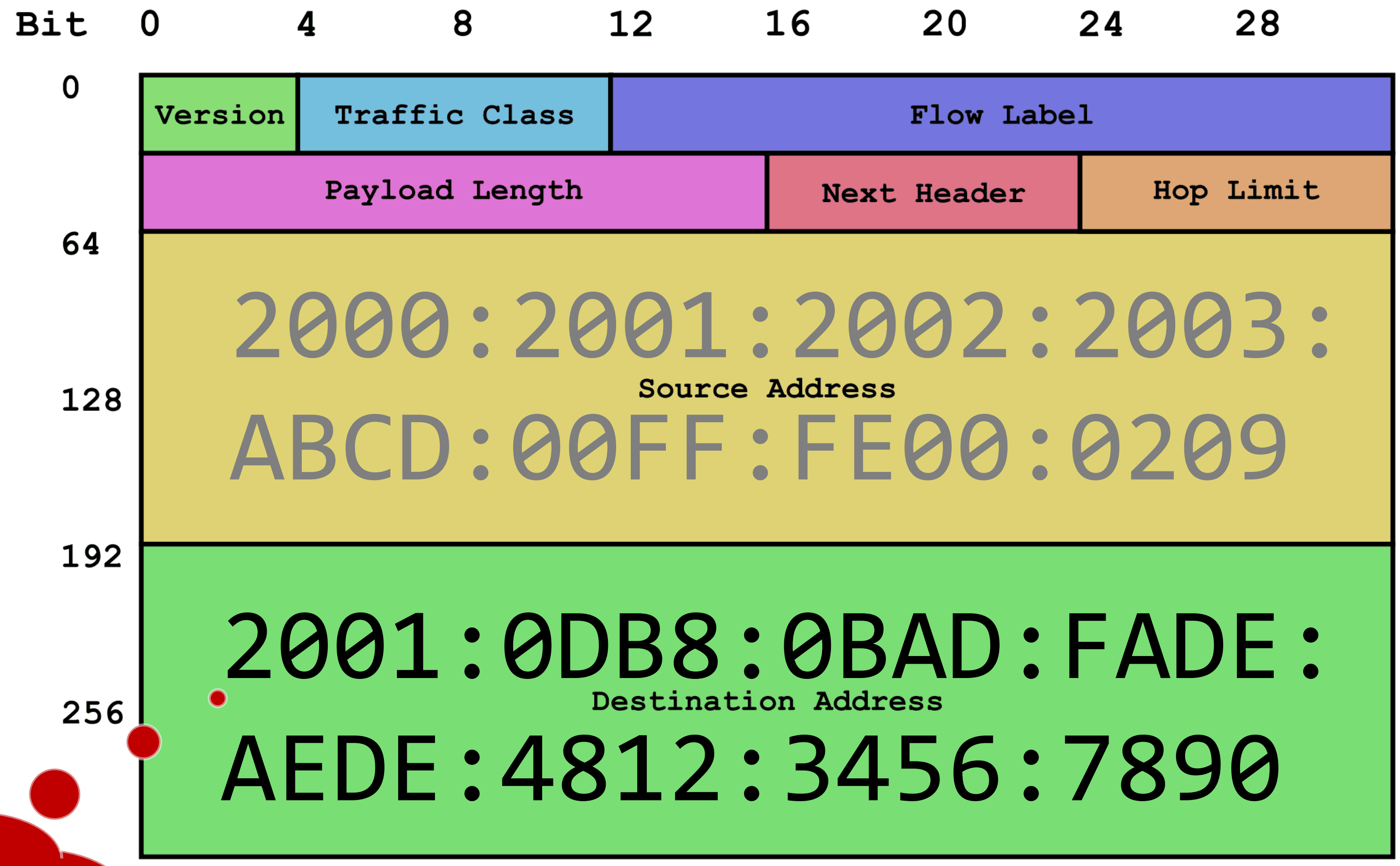
IPv6 Addresses in 6LoWPAN

2001:0DB8:
0BAD:FADE:
AEDE:4812:
3456:7890

All devices are pre-programmed and send traffic to the internet to a specific small set of IPv6 hosts



Case well-known full IPv6 address



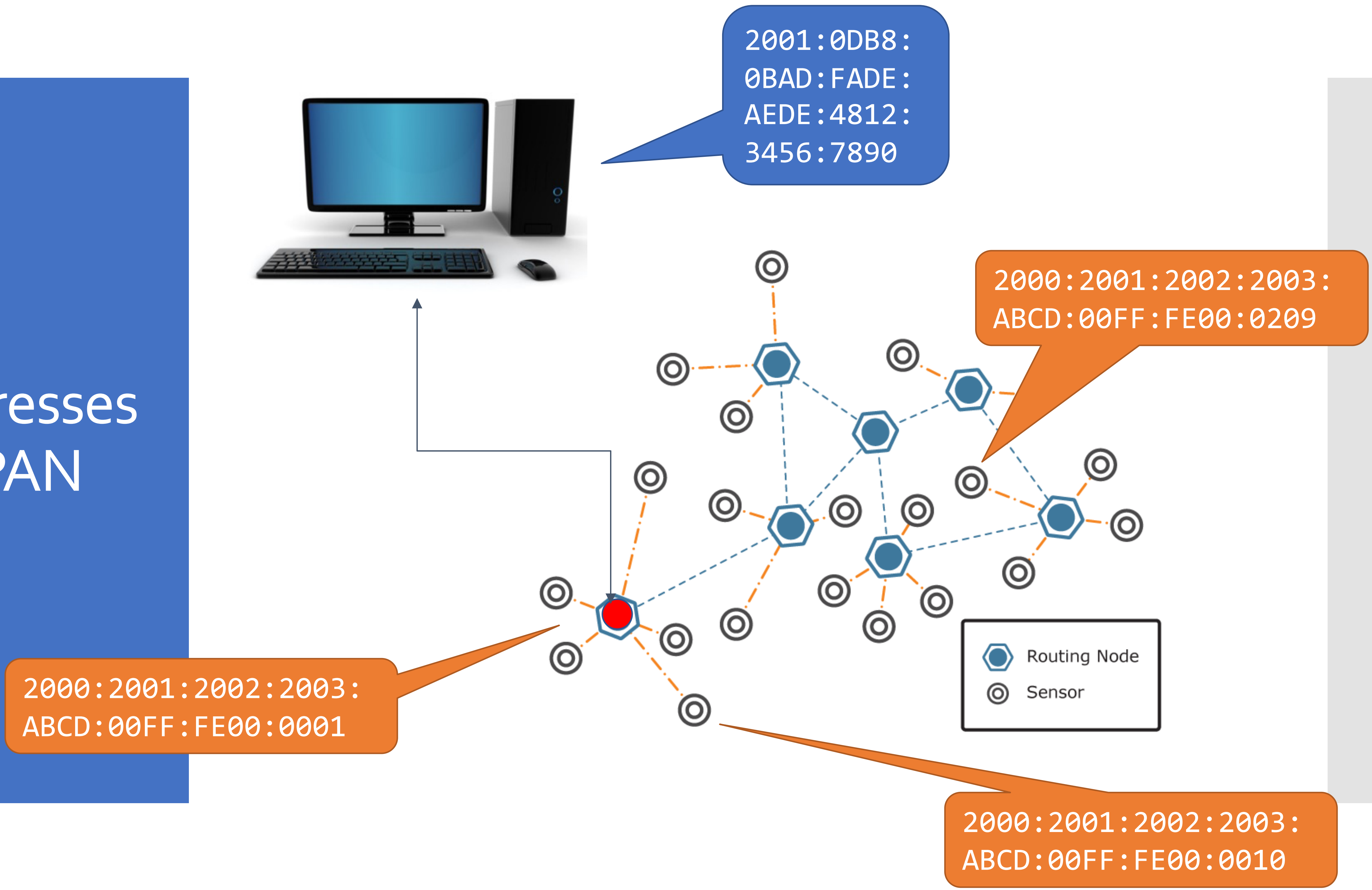
If this address is known to everyone, lets' encode it shorter!

Where can it be applied?

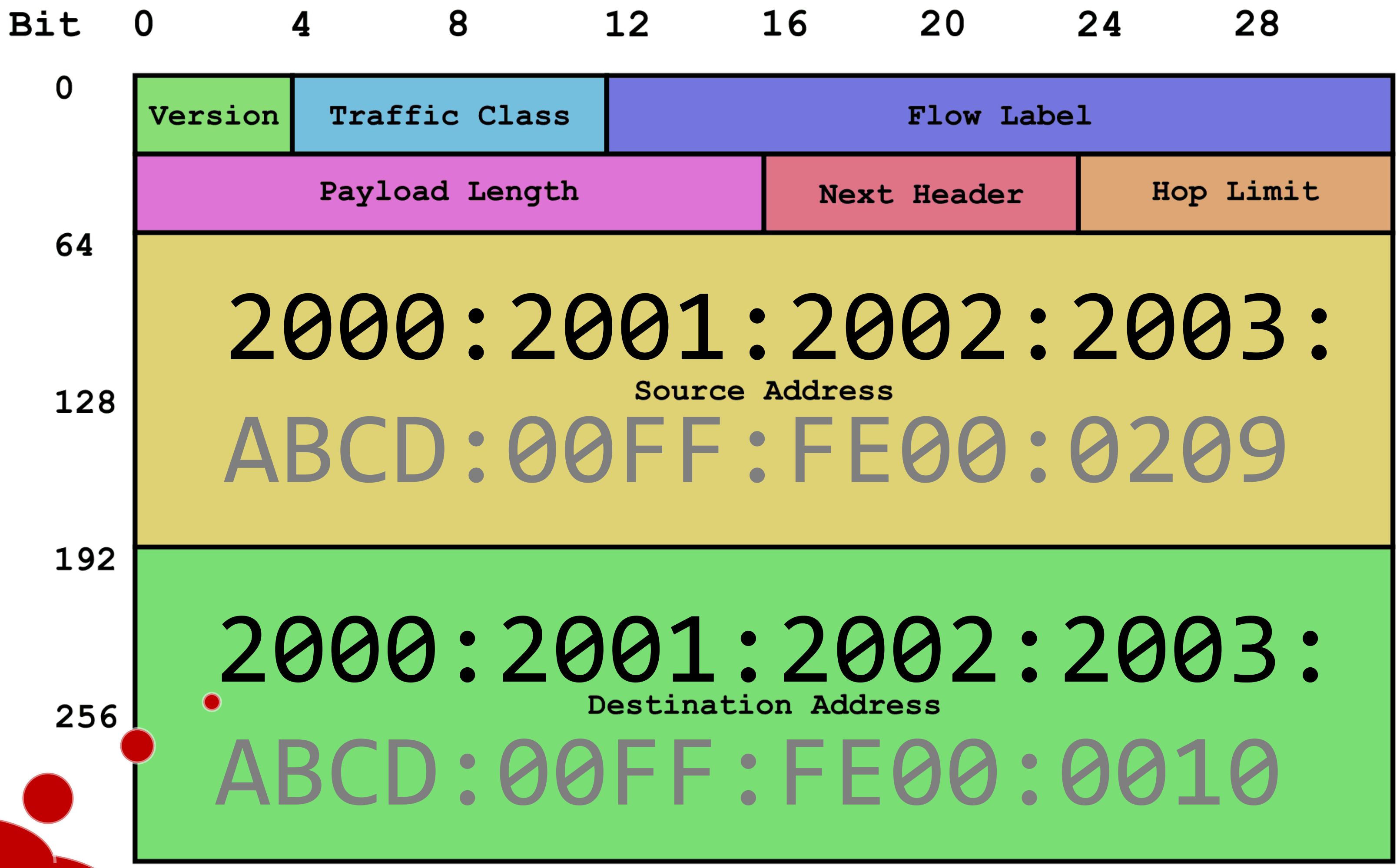
Well-known IPv6 server address outside

Well-known IPv6 **Prefix**

IPv6 Addresses in 6LoWPAN



Case well-known IPv6 network prefix

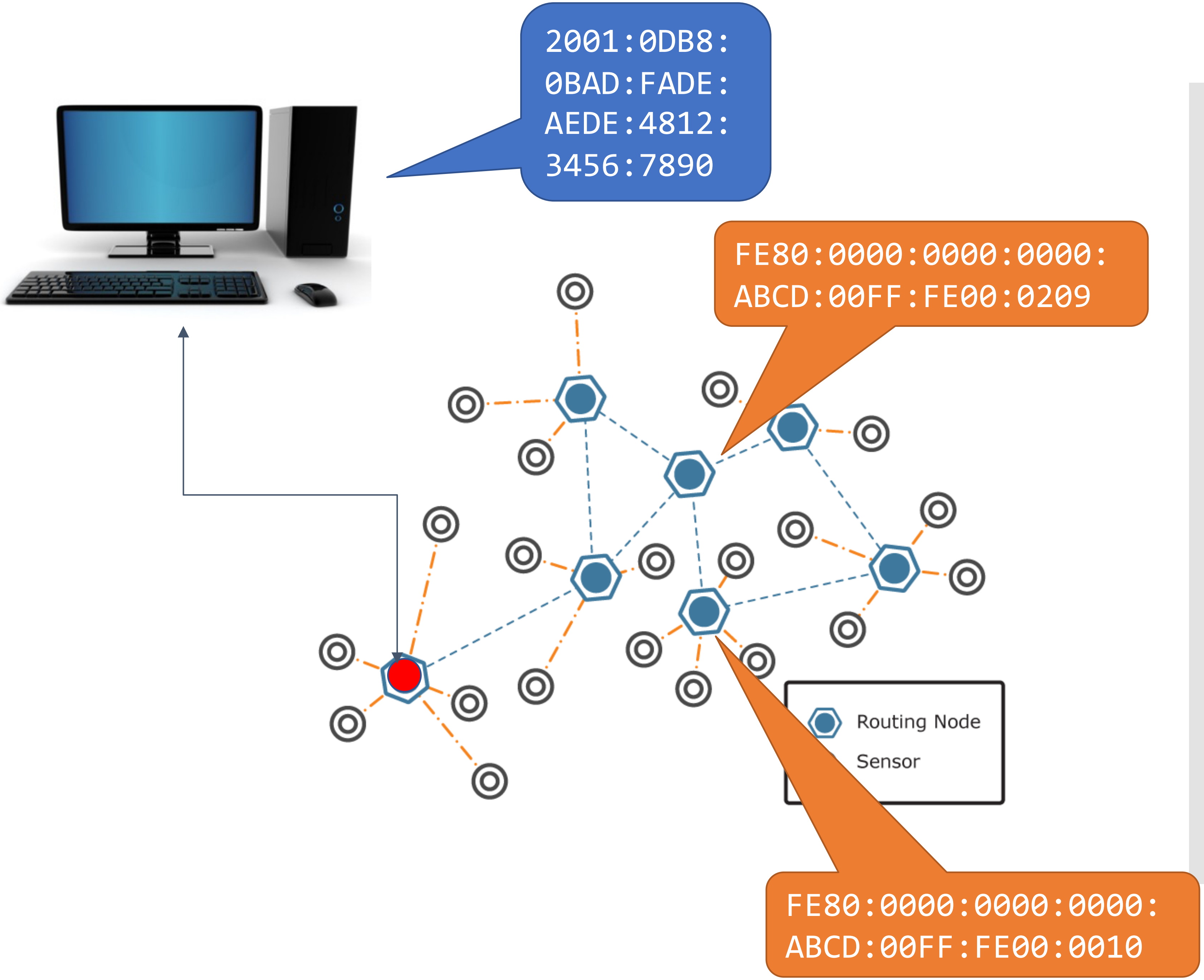


If this prefix is known to everyone, lets' encode it shorter!

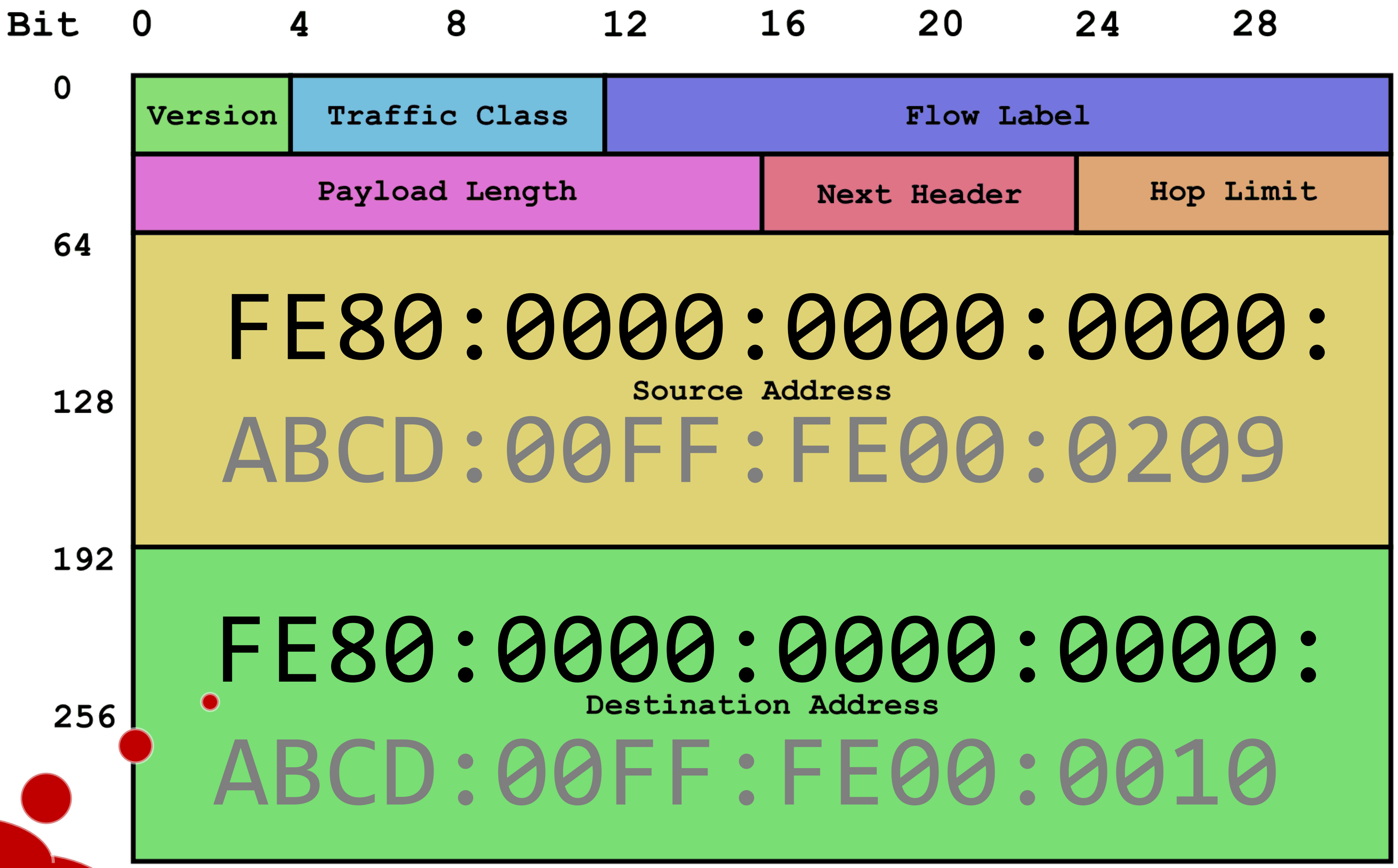
Where can it be applied?

- Well-known IPv6 server address outside
- Well-known 6LowPAN address prefix
- Well-known **Local address prefix**

IPv6 Addresses in 6LoWPAN



Case well-known IPv6 local prefix



If this prefix is known to everyone, lets' encode it shorter!

Where can it be applied?

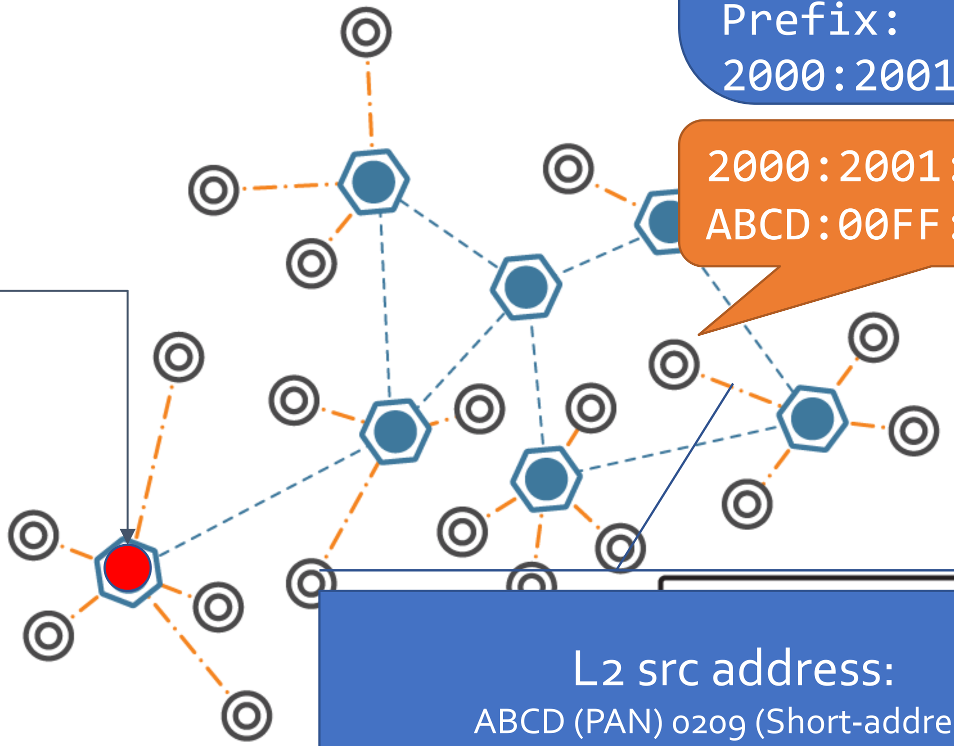
- Well-known IPv6 server address outside
- Well-known 6LowPAN address prefix
- Well-known Local address prefix
- SLAAC global address (Derived from L2 physical address

IPv6 Addresses in 6LoWPAN

2001:0DB8:
0BAD:FADE:
AEDE:4812:
3456:7890

Data from LR-WPAN
PAN ID: ABCD
Short addr: 0209
(after association)

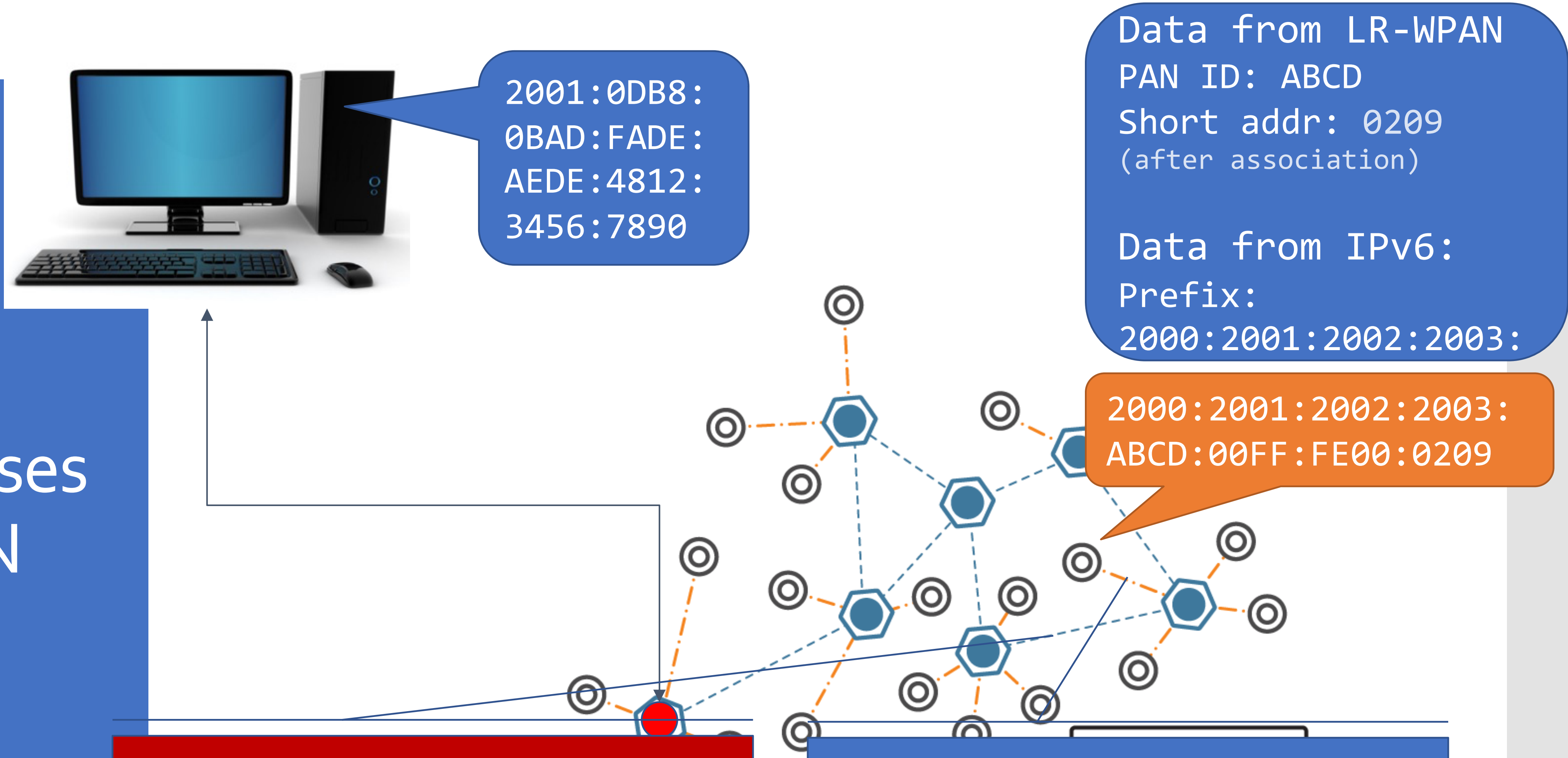
Data from IPv6:
Prefix:
2000:2001:2002:2003:



2000:2001:2002:2003:
ABCD:00FF:FE00:0209

L2 src address:
ABCD (PAN) 0209 (Short-address)
can be used to fully infer IPv6 src address:
2000:2001:2002:2003:ABCD:00FF:FE00:0209

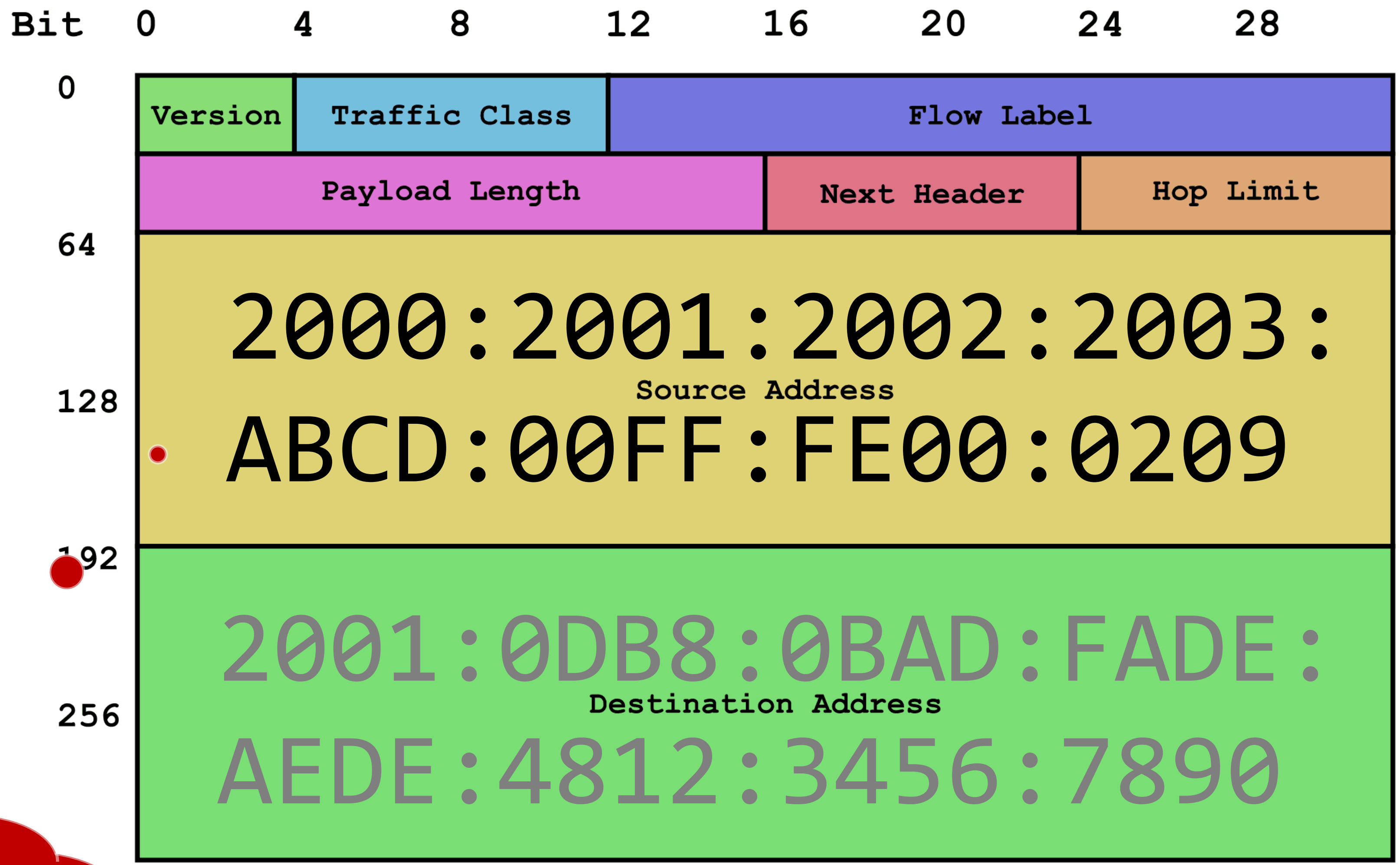
IPv6 Addresses in 6LoWPAN



Once the datagram reaches this router, the L2 address cannot be used for infering the IPv6 address!

L2 src address:
ABCD (PAN) 0209 (Short-address)
can be used to fully infer IPv6 src address:
2000:2001:2002:2003:ABCD:00FF:FE00:0209

Case well-known IPv6 address generation



If one knows how to generate this address from information on L2, lets' encode it shorter!

Where can it be applied?

- Well-known IPv6 server address outside
- Well-known 6LowPAN address prefix
- Well-known Local address prefix
- SLAAC global address (Derived from L2 physical address
- Well-known local SLAAC

Where can it be applied?

- Well-known IPv6 server address outside
- Well-known 6LowPAN address prefix
- Well-known Local address prefix
- Well-known local SLAAC
- Known global SLAAC
- Well-known Multicast groups

…

6LoWPAN Context-aware Compression (IPHC)

Two mechanisms for IPv6 header compression

Stateless Header Compression (HC1/HC2 – not recommended)	**Context-Aware Header Compression (IPHC – currently in use)**
IPv6+TCP/UDP can be compressed down to 7 bytes when: • Using SLAAC link-local addresses • Both UDP/TCP ports are in the range 61616-61631	IPv6 can be encoded in 3 bytes when: Using SLAAC link-local addresses UDP can encoded in • 2 bytes when both ports are in range 61616-61631 • 4 bytes when on one of the ports is in the range 61440-61631
If prefix is not link-local, 8-bytes prefix is carried in the header (adds 8 bytes)	If prefix is not link-local, it can be derived from a 1-byte "context" word Context can be 64, 116 or 128-bits long Suffix can be carried inline with 64, 16 or 0 bits long
Hop limit is not compressed	Hop-limit is limited to a fixed set of values

The well-known parts…

A 6LoWPAN **Context**:

a well-known **sequence of bits** of an IPv6 address that is known by **everyone** in the 6LoWPAN network

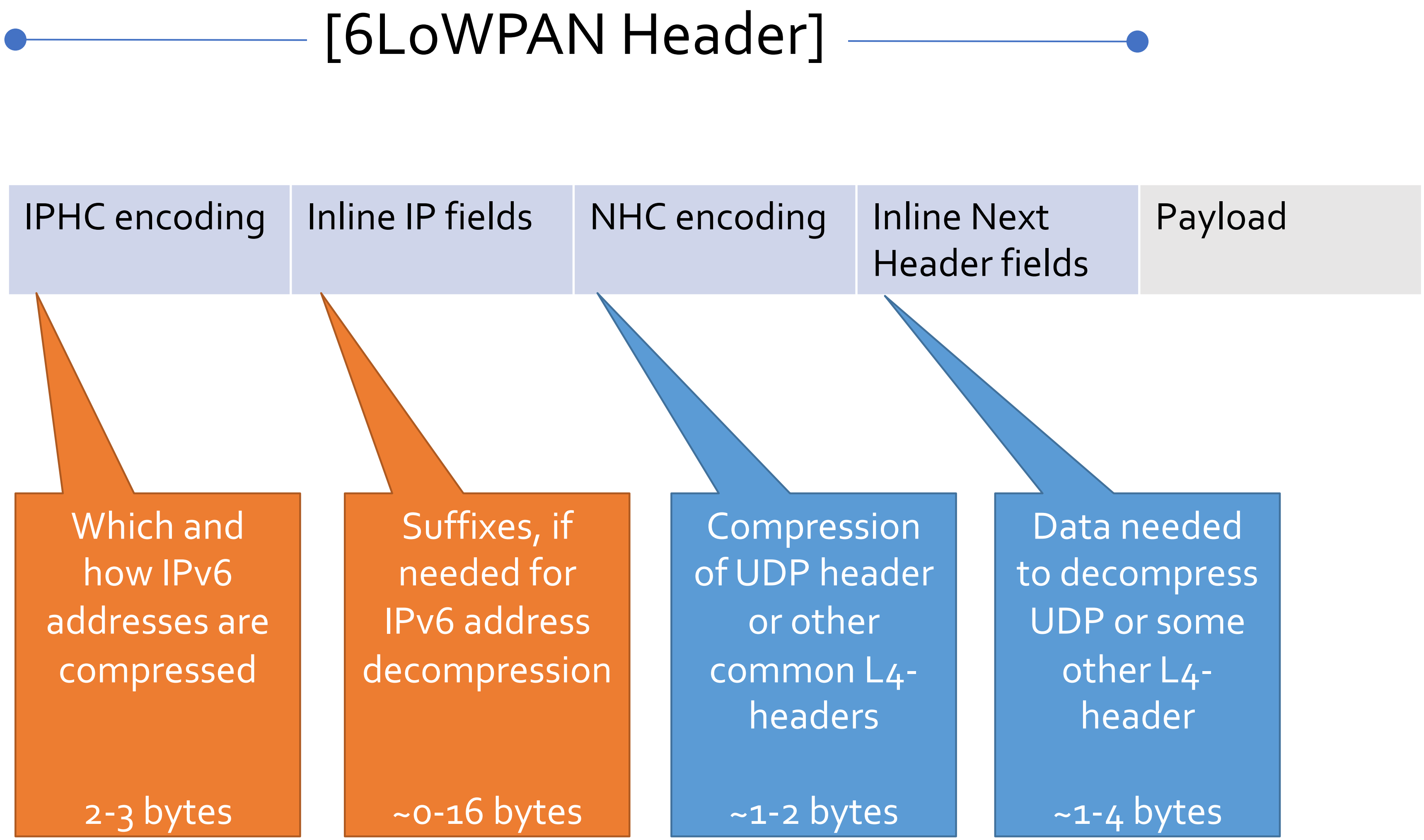
E.g., The link-local prefix:
`FE80:0000:0000:0000:`
It's a well-known **context** of 64-bits length in any network!

My edge router IPv6 address:
`2001:0DB8:0BAD:FADE:AEDE:4812:3456:7890`
Could be a **context** of <u>128-bits</u> length!

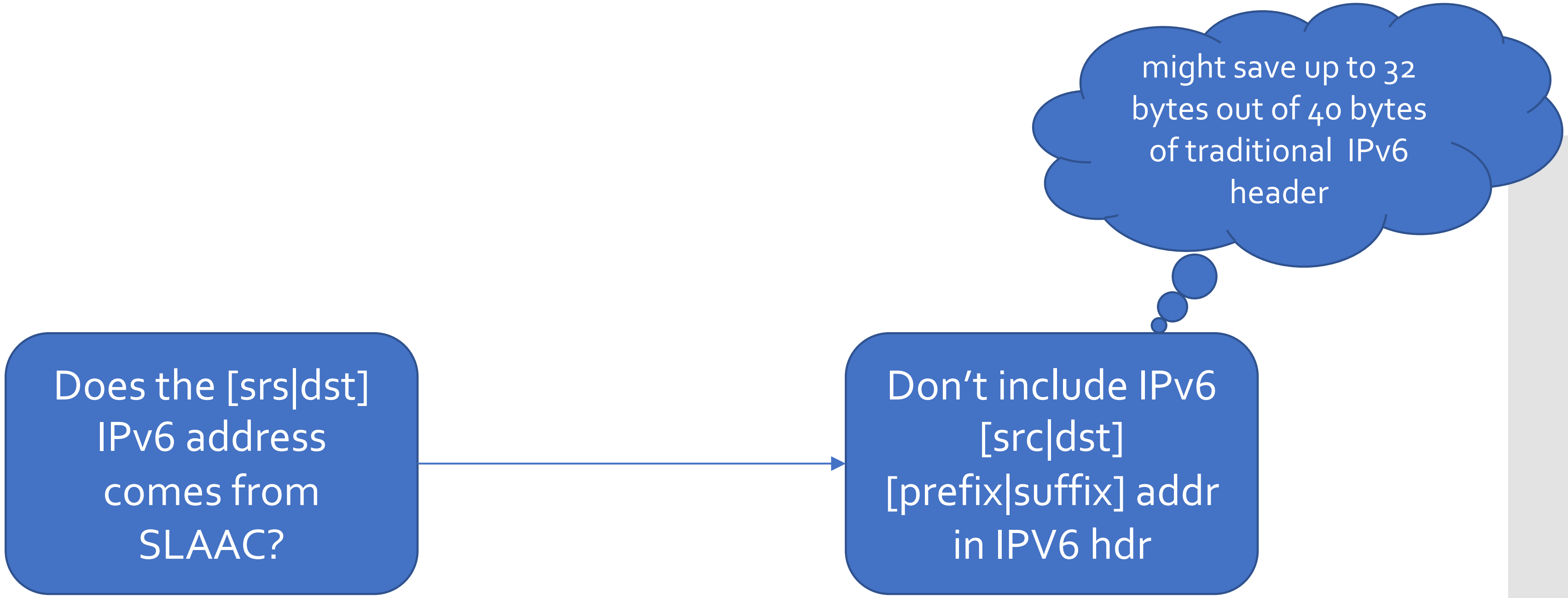
As well, a **context** could be:

`2001:0DB8:0BAD:FADE:AEDE:4812:3456:` (<u>112-bits</u> context)
`2001:0DB8:0BAD:FADE:` (<u>64-bits</u> context)

6LoWPAN General Header Format

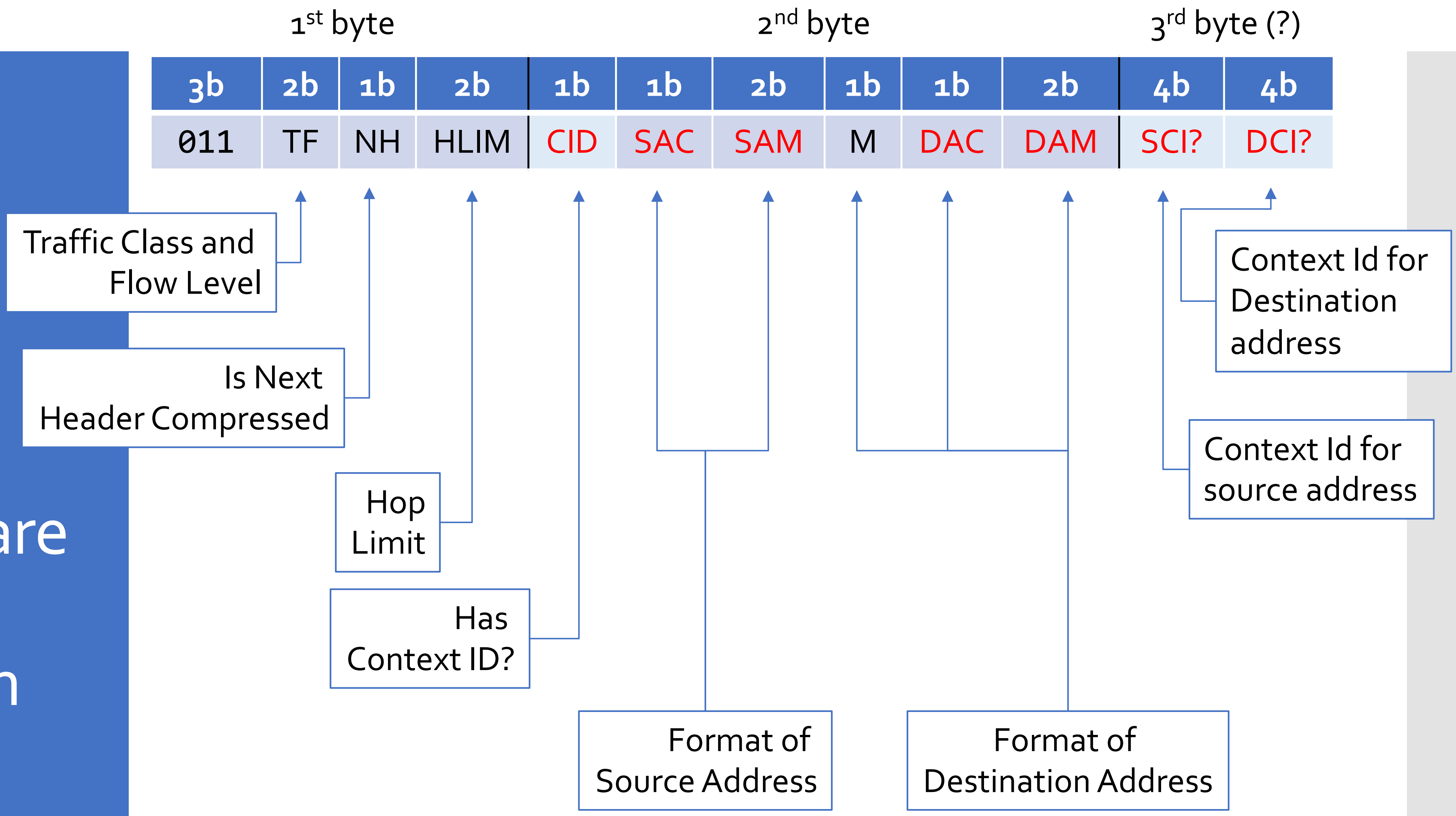


IPv6 header compression main idea



On reception of a 802.15.4 frame, the IPv6 layer must re-create the [src|dst] IPv6 address with no information, or only part of the information stored in the IPv6 header (e.g., only the prefix)

Context-aware Header Compression (IPHC)



Context-aware Header Compression

(IPHC)

3b	2b	1b	2b	1b	1b	2b	1b	1b	2b	4b	4b
011	TF	NH	HLIM	CID	SAC	SAM	M	DAC	DAM	SCI?	DCI?

SAC/ DAC	SAM/ DAM	Interpretation SAM/DAM	DAM, if M=1 (multicast)
0	00	[128-bits inline], no compression	(same as SAM)
	01	FE80:0:0:0:[64-bits inline]	FFxx::00xx:xxxx:xxxx (48-bits)
	10	FE80:0:0:0:0:0:0:[16-bits inline]	FFxx::00xx:xxxx (32-bits inline)
	11	FE80:0:0:0:link-layer (0-bits inline)	FF02::00xx (16-bits inline)

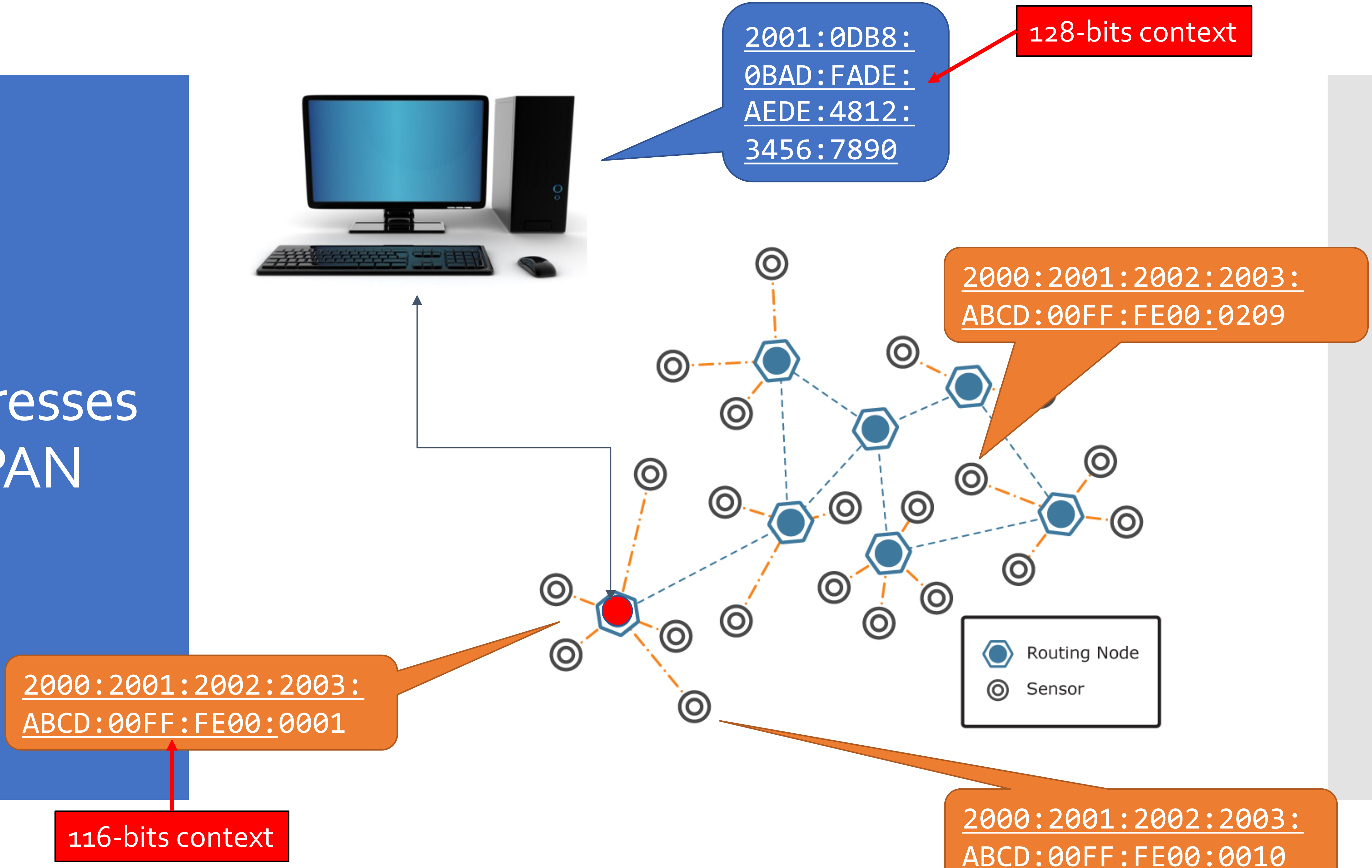
Context-aware Header Compression

(IPHC)

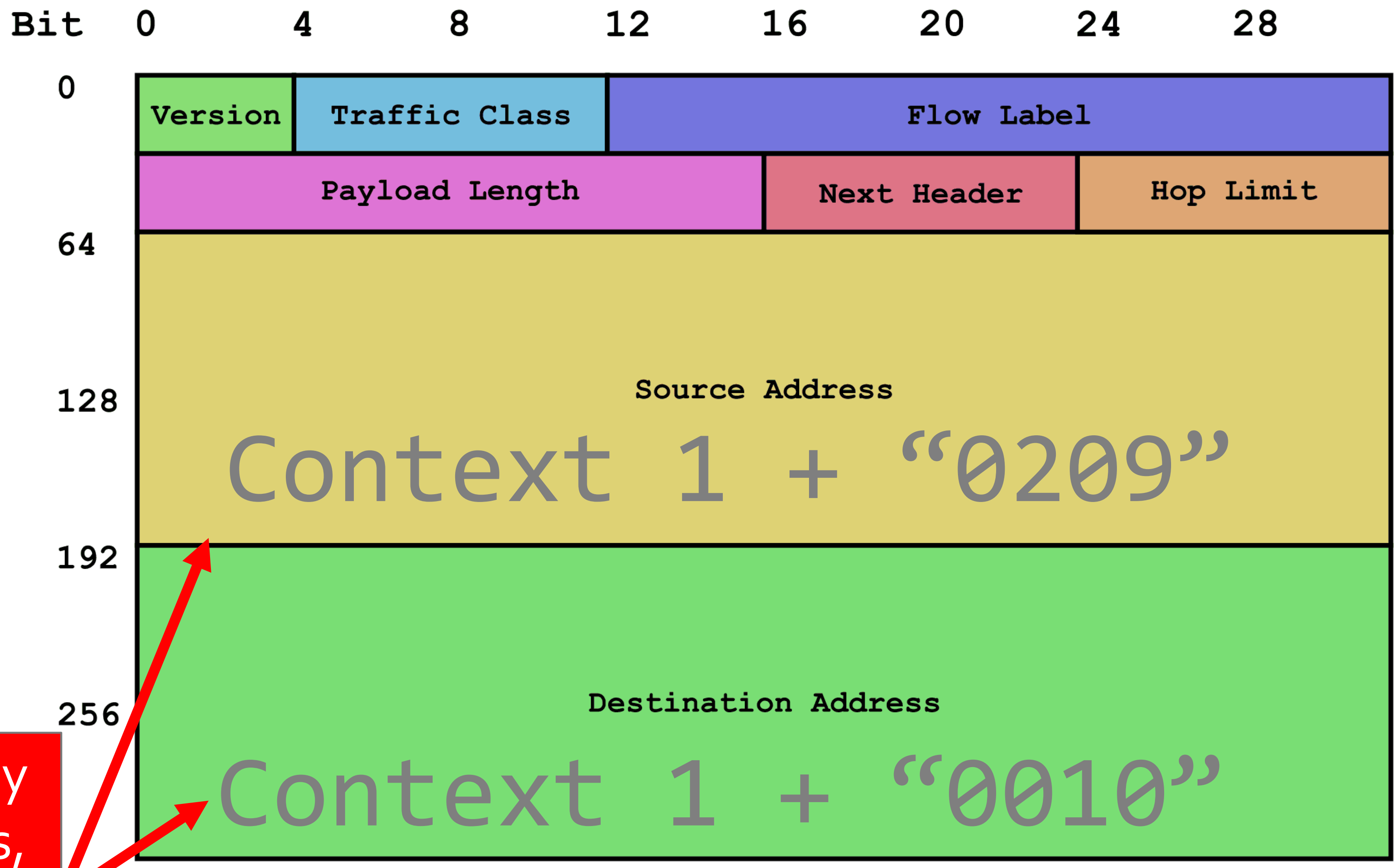
3b	2b	1b	2b	1b	1b	2b	1b	1b	2b	4b	4b
011	TF	NH	HLIM	CID	SAC	SAM	M	DAC	DAM	SCI?	DCI?

SAC/ DAC	SAM/ DAM	Interpretation SAM/DAM	DAM, if M=1 (multicast)
0	00	[128-bits inline], no compression	(same as SAM)
	01	FE80:0:0:0:[64-bits inline]	FFxx::00xx:xxxx:xxxx (48-bits)
	10	FE80:0:0:0:0:0:0:[16-bits inline]	FFxx::00xx:xxxx (32-bits inline)
	11	FE80:0:0:0:link-layer (0-bits inline)	FF02::00xx (16-bits inline)
1	00	reserved	RFC3306 and RFC3956
	01	Context:[64-bits inline]	reserved
	10	Context:[16-bits inline]	reserved
	11	Context (0-bits inline)	reserved

IPv6 Addresses in 6LoWPAN



40 bytes for IPv6 header



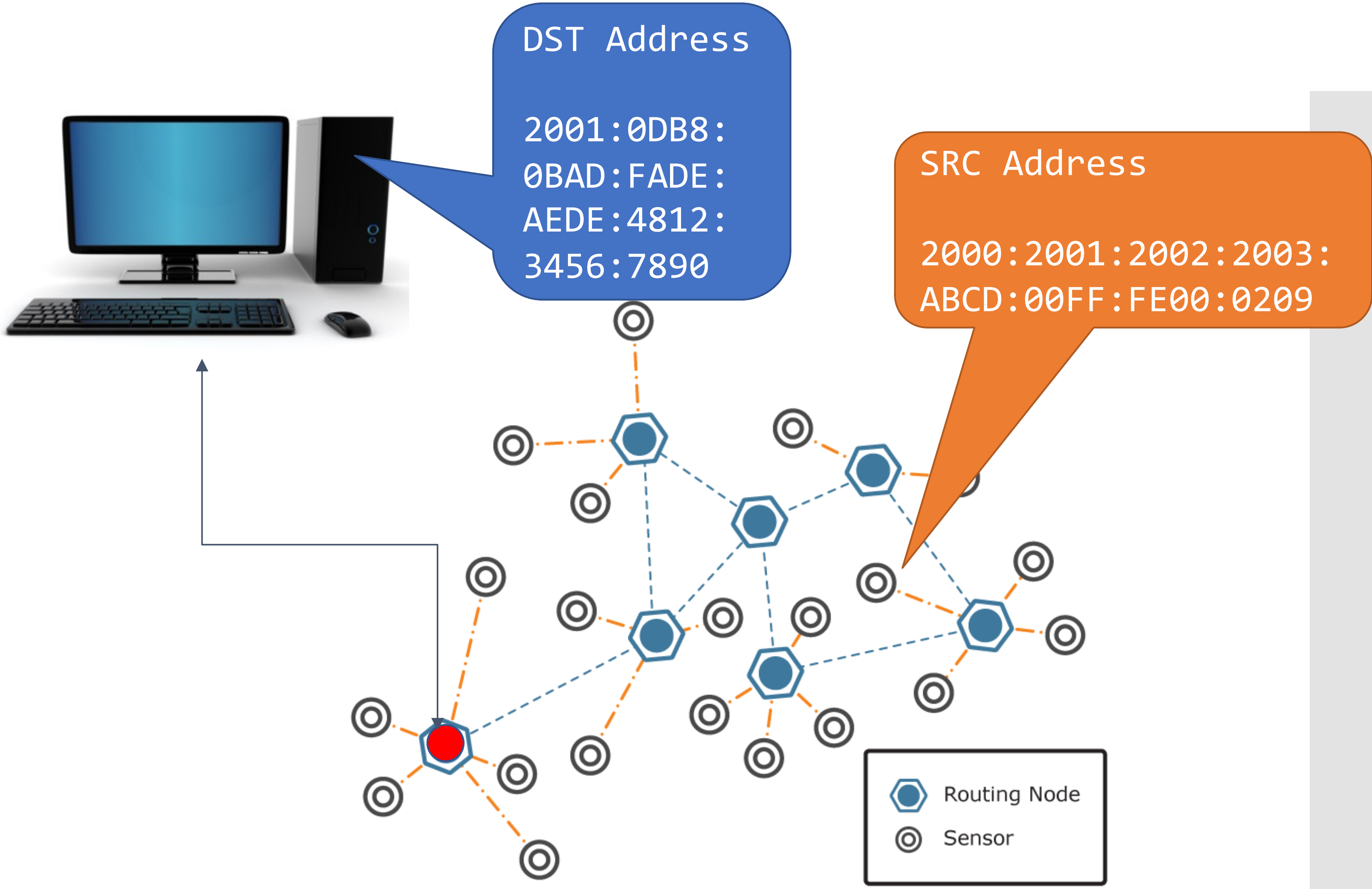
All nodes must know already what each context id means, e.g.,

2000:2001:2002:2003:ABCD:00FF:FE00

Traffic Class: 6-bits Differentiated services (classify packets).
2-bits for ECN: source provides congestion control.

Flow label: Realtime services. All packets of that flow should stay on the same path

IPHC Example 1



IPHC Example 1

From:
2000:2001:2002:2003:
ABCD:00FF:FE00:0209

To:
2001:0DB8:0BAD:FADE:
AEDE:4812:3456:7890

Context 1 (128 bits):
2001:0DB8:0BAD:FADE:
AEDE:4812:3456:7890

Context 2 (116 bits):
2000:2001:2002:2003:
ABCD:00FF:FE00:0000

From:
Context 2 + 16-bits "0209"

To:
Context 1 + 0-bits

IPHC Example 1

`0x7f = ~~0 1 1~~ 1    1 1 1 1`

011 = IPHC
11 [TF] = Traffic class and flow label elided
1 [NH] = next header compressed
11 [HLIM] = hop limit is 255

`0xe7 = 1 1 1 0    0 1 1 1`

1 [CID] = **additional 8-bit context provided**
1 [SAC] = **src addr uses context**
10 [SAM] = **src address suffix is 16-bits long**
0 [M] = destination is not multicast
1 [DAC] = **dst addr uses context**
11 [DAM] = **full address from context**

From:
2000:2001:2002:2003:
ABCD:00FF:FE00:0209

To:
2001:0DB8:0BAD:FADE:
AEDE:4812:3456:7890

Inline IP fields

25 41 98 00 00 00 ff ff 5d 39	7f e7 21 02 09
802.15.4	6Lo

Context 2 for src
Context 1 for dst

IPHC Example 2

From:
FE80::0200:FF:FE00:5D39

To:
FF02::0002

Context 1:
2001:0DB8:0BAD:FADE:
AEDE:4812:3456:7890

Context 2:
2000:2001:2002:2003:
ABCD:00FF:FE00:0000

From:
Link-local prefix + EUI-64

To:
Multicast address + 8-bits suffix

IPHC Example 2

0x7b = ~~0 1 1~~ 1 1 0 1 1

011 = IPHC

11 [TF] = Traffic class and flow label elided

0 [NH] = **full 8-bits of next header inline**

11 [HLIM] = hop limit is 255

0x3b = 0 0 1 1 1 0 1 1

0 [CID] = no additional 8-bit context

0 [SAC] = src addr uses no context

11 [SAM] = src address is link-layer derived

1 [M] = destination is multicast

0 [DAC] = dst addr uses no context

11 [DAM] = **address takes the form ff02::00xx**

From:

FE80::0200:FF:FE00:5D39

To:

FF02::0002

Inline IP fields

25 41 98 00 00 00 ff ff 5d 39 | 7b 3b 3a 02 XX

802.15.4 | 6Lo

Next header
(protocol num.)

IPHC

Next Header

IPHC encoding	Inline IP fields	**NHC encoding**	Inline Next Header fields	Payload

IPv6 Extension
Header Compression

4b	3b	1b
1110	EID	NH

0xE_

UDP
Header Compression

5b	1b	2b
11110	C	P

0xF_

C=0, checksum present in packet

0xF000 = 61440
0xF0FF = 61695
0xF0B0 = 6166
0xF0BF = 61631

P value	Src Port	Dst Port
00	16-bits inline	16-bits inline
01	16-bits inline	0xF0 + 8-bits inline
10	0xF0 + 8-bits inline	16-bits inline
11	0xF0B + 4 bits inline	0xF0B + 4-bits inline

If we were using 802.15.4 in different ways…

25 21 dc 0d 00 00 ~~02 00 00 ff fe 00~~ 12 12 ~~00 00~~
~~02 00 00 ff fe 00~~ 89 b5 7d 33 f7 00 61 6c de

Total length using long addresses: 31 bytes

19 21 98 0d ~~00 00~~ ~~12 12~~ ~~00 00~~ ~~89 b5~~ 7d 33 f7 00
61 6c de

Total length using short addresses: 19 bytes

11 61 10 0d 7d 33 f7 00 61 6c de

Total length using GTS: 11 bytes

* Coding values are shown as example

% of frame length used by headers

57%

Traditional IPv6

8,5%

GTS+IPHC+NHC

Why is all this important for you?