Internet of Things

Laurent Toutain

November 17, 2014

Table of Contents

Introduction

IoT: Where are we ?

Why Internet ?

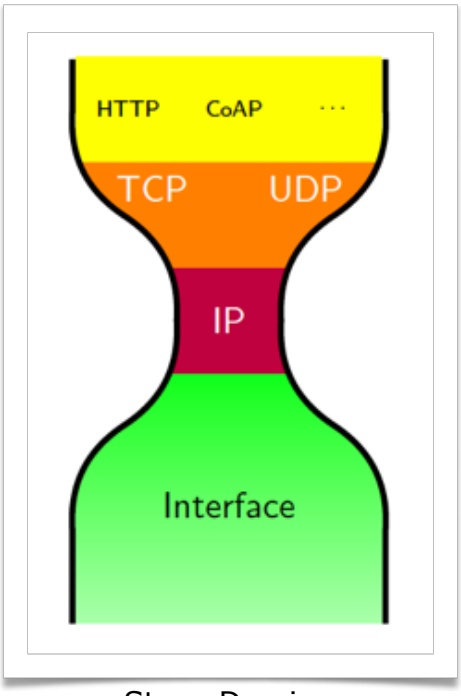
- Histoty is repeating
  - 1990 IPX, Netbios better protocol for PC
  - 1990 Frame Relay, ATM better protocol for WAN
  - IP wins to ease interconnection
    - interconnection costs decrease
  - Virtuous circle

IP is never the best protocol for a specific environment, but it is technically efficient and everywhere

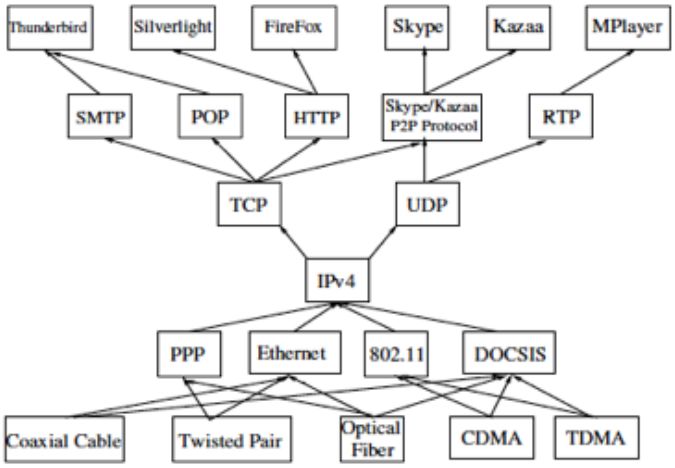
Internet Architecture Model

- **Very successful for almost 30 years**
  - Connecting almost everything
  - Flexible
    - On top of many links
      - Low speed, high speed, variable latencies
    - Large variety of applications
      - File transfer, streaming, voip,…

Hourglass Model

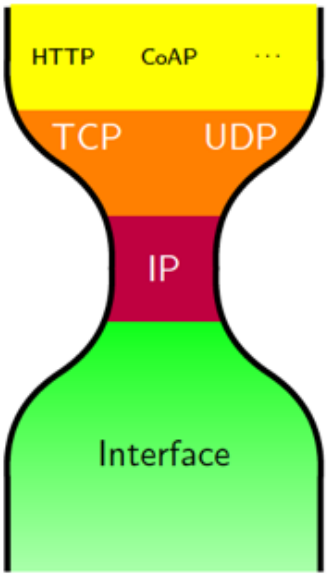


Steve Deering



The Evolution of Layered Protocol Stacks Leads to an Hourglass-Shaped Architecture
Saamer Akhshabi, Constantine Dovrolis
Sigcomm 2011

Internet Protocol



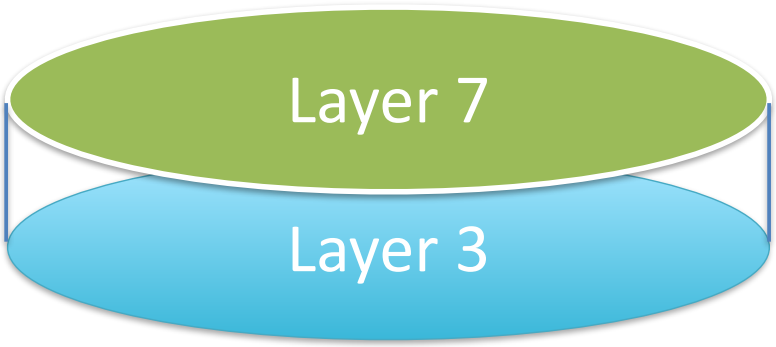
Steve Deering

- **Interoperability,**
- **But ossification.**

IPv6

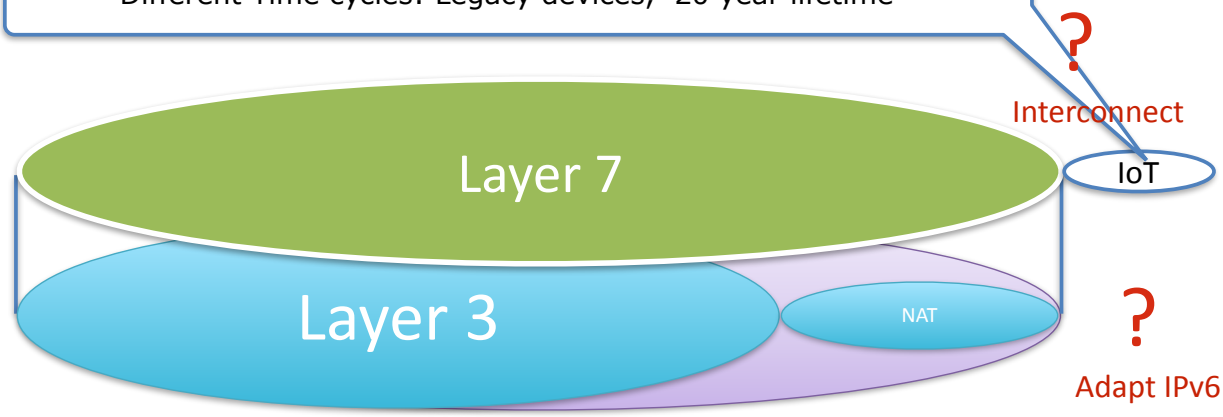
- **IPv6 slowly introduced**
  - No direct interoperability with IPv4
  - Metcalfe's law against IPv6
  - Forwarding is not the most difficult part
- **IPv6 has advantages for IoT**
  - Auto-configuration
  - Simpler
  - Layer 2 agnostic
- **But difficult to make IPv6 evolve**
  - Already an old protocol?

IP scope: End to End

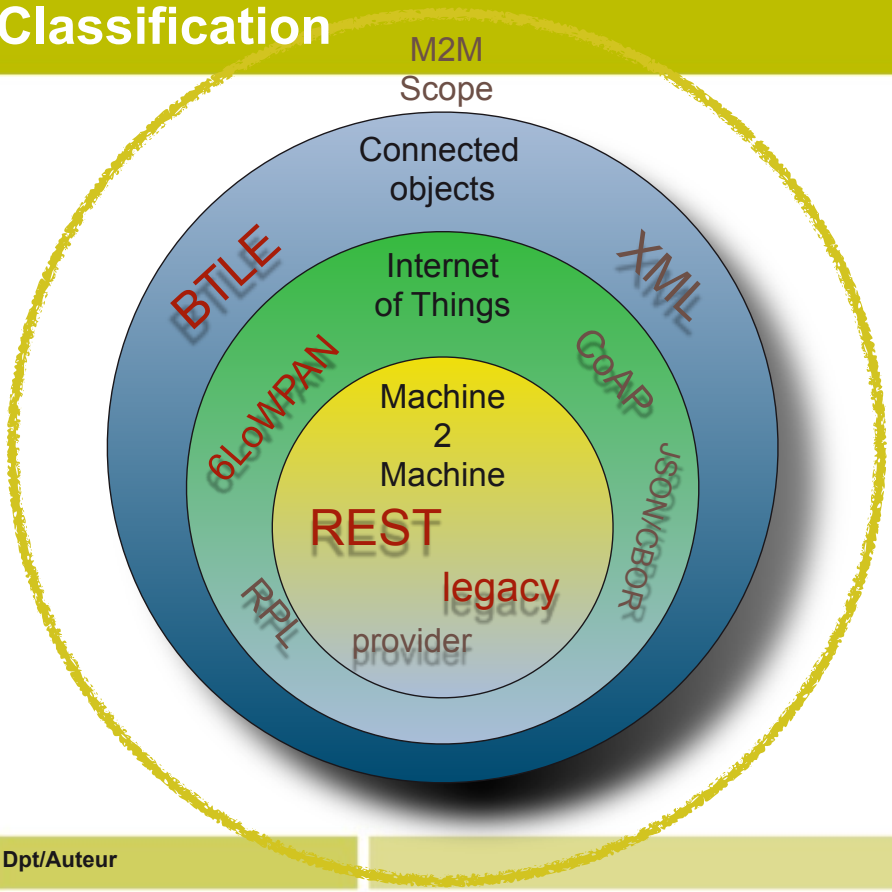


IP scope: End to End

- Cheap (Moore's law reduces costs, does not increase power)
- Low Memory
- Low Energy
- Different Time cycles: Legacy devices, 20 year lifetime



Classification



Internet of Things

- Adapt Internet to a contraint environment.
- Based on IPv6 to avoid addressing constraints
  - Auto-configuration
  - Independent from media and applications
- Routing protocol:
  - RPL
  - Versatile but lot of parameters
- REST architecture extensions to constraint environments
  - CoAP
- Bootstrapping/provisioning

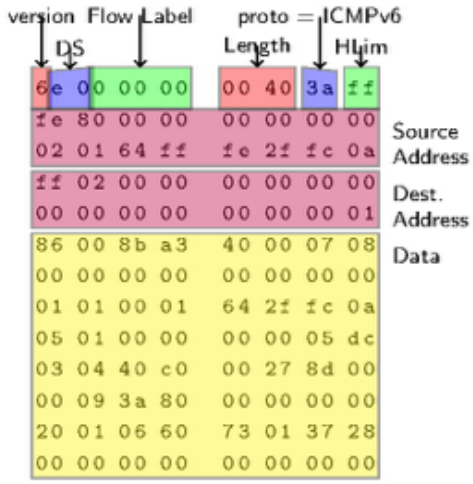
Zigbee Protocol Stack

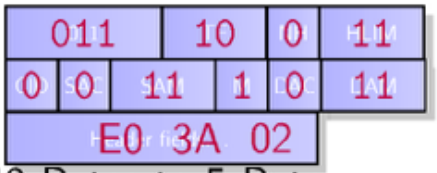
URI

IPv6

MAC

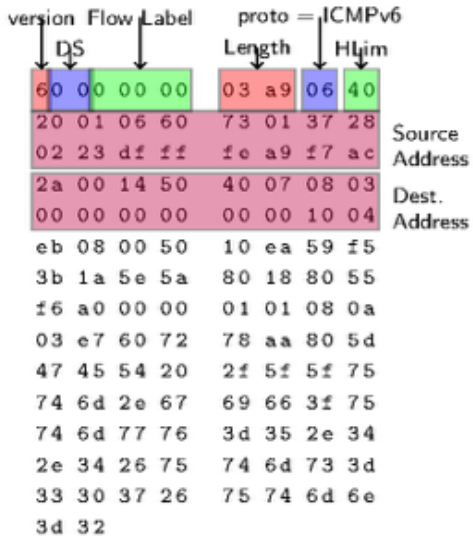
Example: Compress

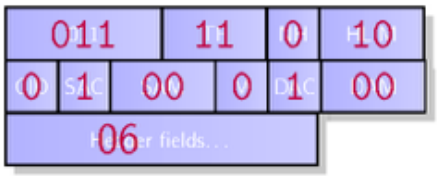




40 Bytes to 5 Bytes

Example: Compress





20 01 06 60 73 01 37 28
02 23 df ff fe a9 f7 ac

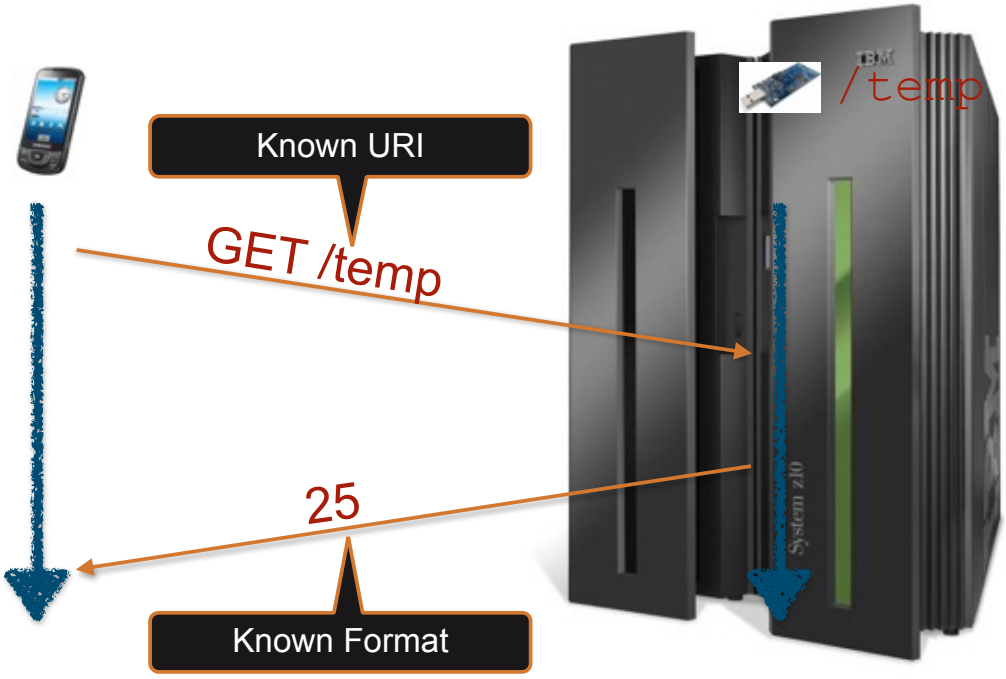
2a 00 14 50 40 07 08 03
00 00 00 00 00 00 10 04

40 Bytes to 35 Bytes

Layer 7 - Interconnection

Internet of Things

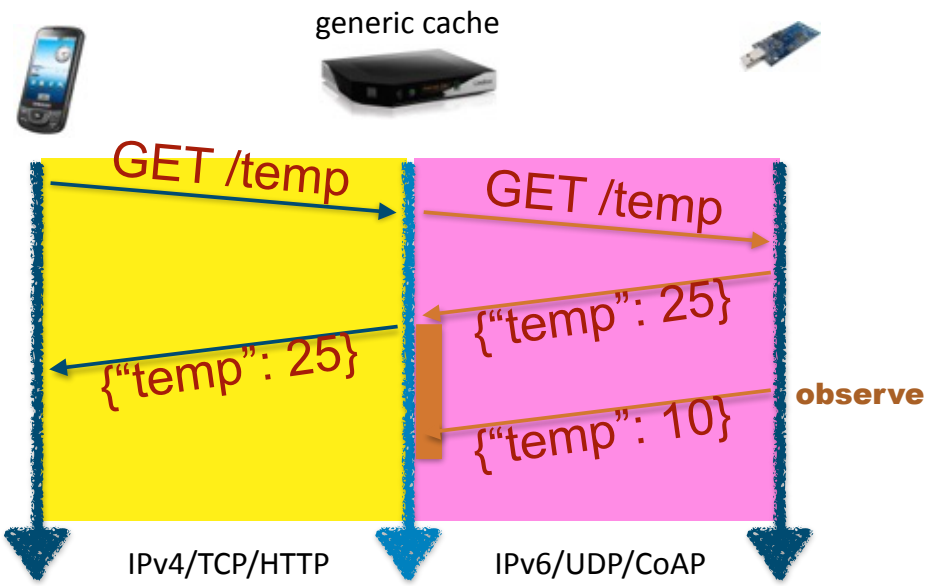
REST : Client/Server



IPv4 or IPv6 ?

- IPv6 is better for IoT…
  - but not very popular.
- Most of cloud computing, computers and smartphone uses IPv4.

Caching

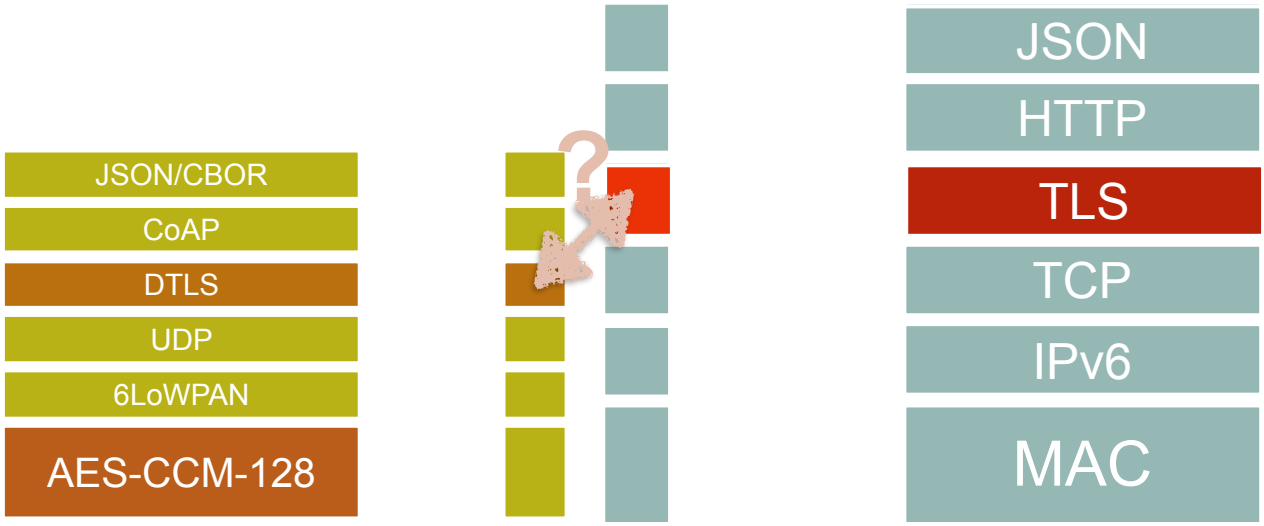


Resource definition (ontology)

Type	Path	RT	IF	Type	Unit
Instantaneous Power	/pwr/{#}/w	ipso.pwr.w	s	d	W
Cumulative Power	/pwr/{#}/kwh	ipso.pwr.kwh	s	d	kWh
Load Relay	/pwr/{#}/rel	ipso.pwr.rel	a	b	
Load Dimmer	/pwr/{#}/dim	ipso.pwr.dim	a	i	0-100 %

draft-ipso-app-framework-04

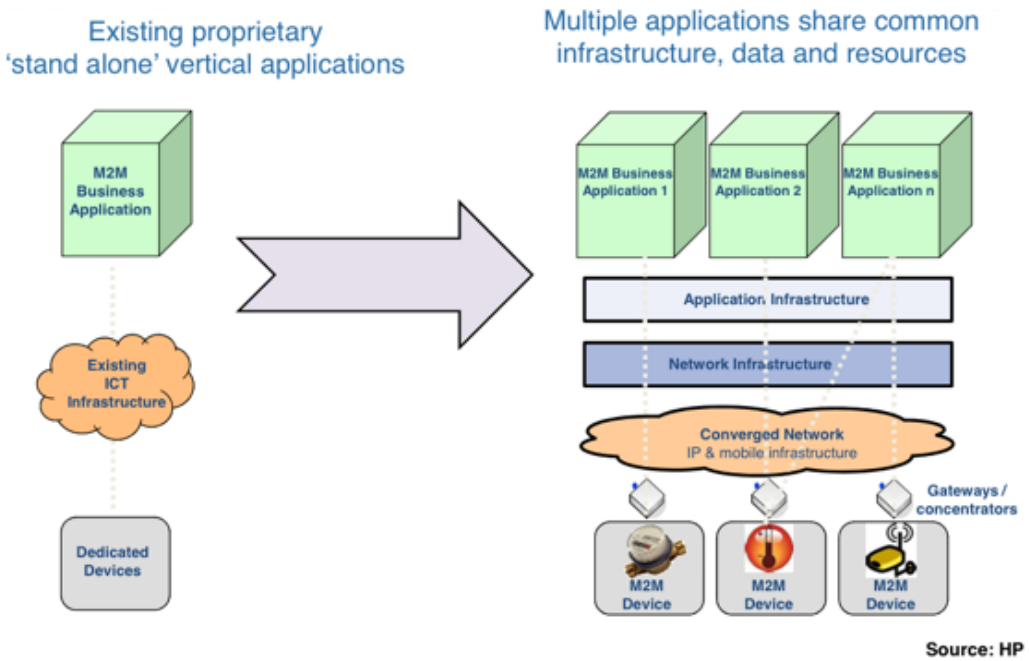
Protocol stack - maintain security associations



Layer 7 - Interconnection

Machine 2 Machine

Inverting the pipes

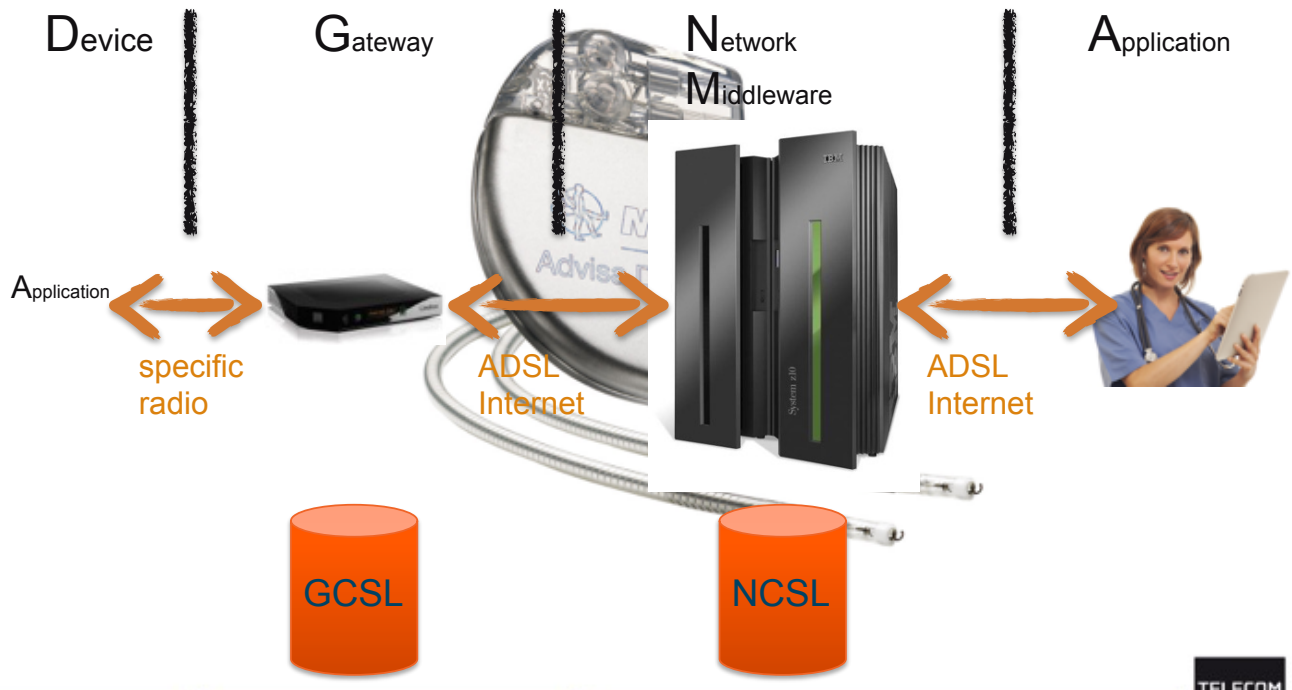


Source: HP

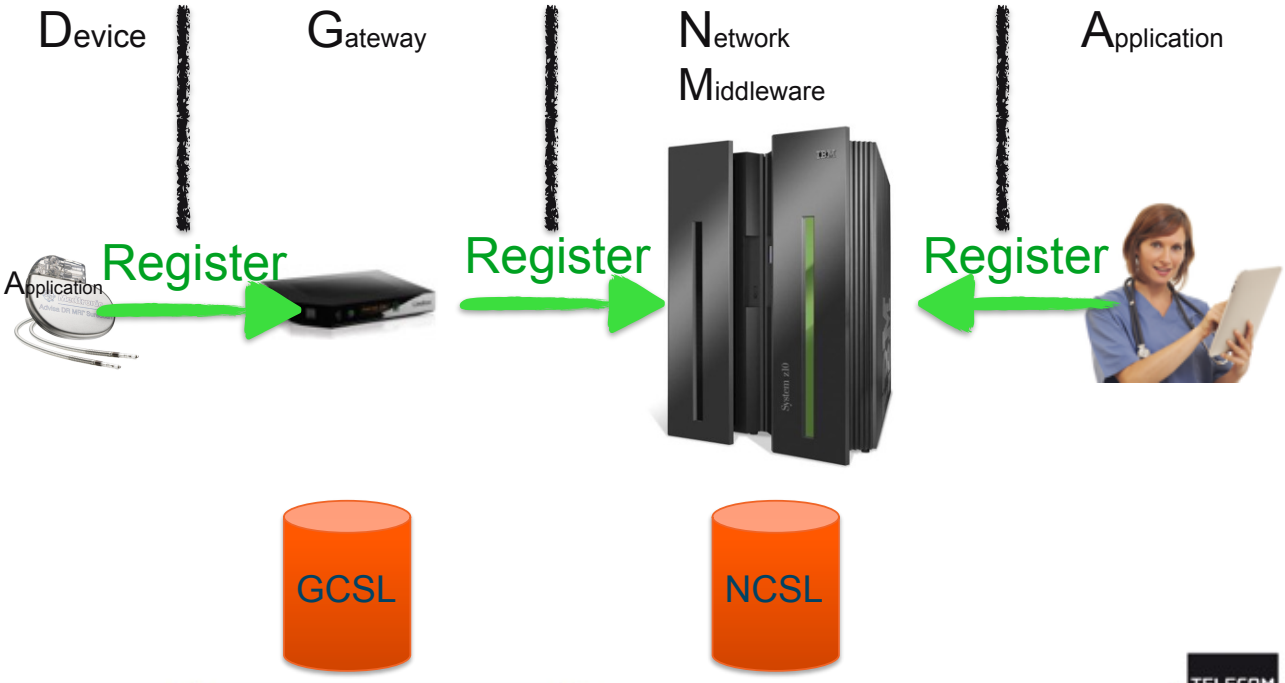
Device characteristics

- Limited in functionality
- Low-powered
- Embedded
- Here to stay

Exemple



Exemple



Hardware & Physical Layer

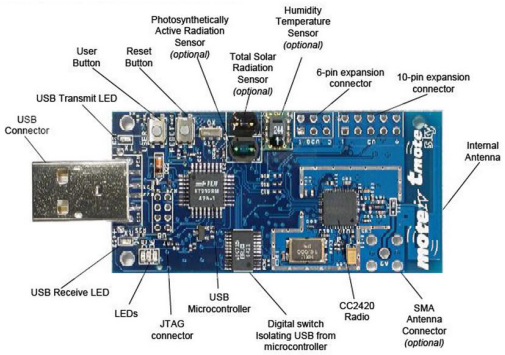
Hardware and physical layers requirements

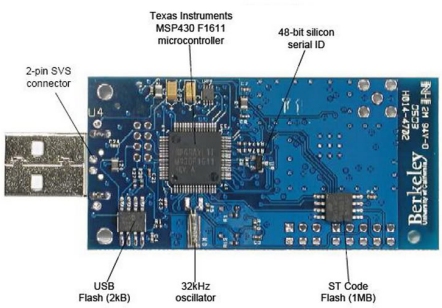
- Both hardware and Physical Layer should be designed to minimize energy consumption
  - Low-consumption processor (RISC, low frequency)
  - Favor non-volatile memory (e.g. Flash) over RAM
  - Radio coding / modulation not very efficient but interference-resilient to simplify MAC protocol

Hardware and physical layers requirements

Layer 2 protocols

- Both hardware and Physical Layer should be designed to minimize energy consumption
  - Low-consumption processor (RISC, low frequency)
  - Favor non-volatile memory (e.g. Flash) over RAM
  - Radio coding / modulation not very efficient but interference-resilient to simplify MAC protocol





Slide 13 Laurent Toutain Filière 2 TELECOM Bretagne

Some Figures

- Micro-controller (TI MSP430)
  - 8 MHz, RISC
- Memory:
  - RAM: 10 kB
  - Flash: 48 kB + 1 MB
- Wireless interface (Chipcon CC 2420):
  - Frequency: 2.4†GHz
  - Throughput: 250 kb/s
  - Max. frame size: 127 B
- Energy:
  - Consumption (Tx): 19.5 mA
  - Consumption (Rx): 21.8 mA
  - Consumption (μC): 1.8 mA
  - Consumption (sens): 54.5 μA
  - Consumption (idle): 5.1 μA
  - Flash reading: 4 mA
  - Flash writing: 20 mA
- Capacity:
  - Alkaline batteries: ˜ 2 Ah
  - Self-discharge: ˜10 μA

What is the most expensive (in term of energy) operation ?

Some Figures

Layer 2 protocols

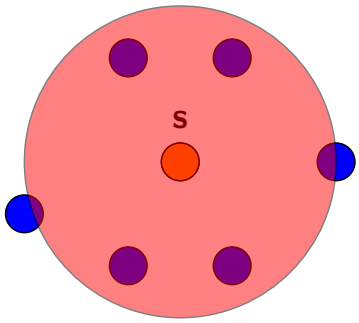
- Micro-controller (TI MSP430)
  - 8 MHz, RISC
- Memory:
  - RAM: 10 kB
  - Flash: 48 kB + 1 MB
- Wireless interface (Chipcon CC 2420):
  - Frequency: 2.4†GHz
  - Throughput: 250 kb/s
  - Max. frame size: 127 B
- Energy:
  - Consumption (Tx): 19.5 mA
  - Consumption (Rx): 21.8 mA
  - Consumption (μC): 1.8 mA
  - Consumption (sens): 54.5 μA
  - Consumption (idle): 5.1 μA
  - Flash reading: 4 mA
  - Flash writing: 20 mA
- Capacity:
  - Alkaline batteries: ˜ 2 Ah
  - Self-discharge: ˜10 μA

What is the most expensive (in term of energy) operation ?

Slide 14 Laurent Toutain Filière 2 TELECOM Bretagne

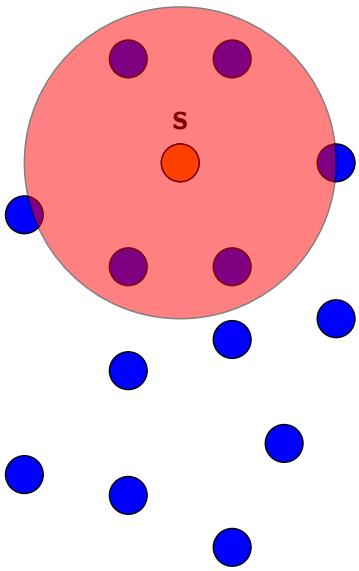
Typical sensor network sketches

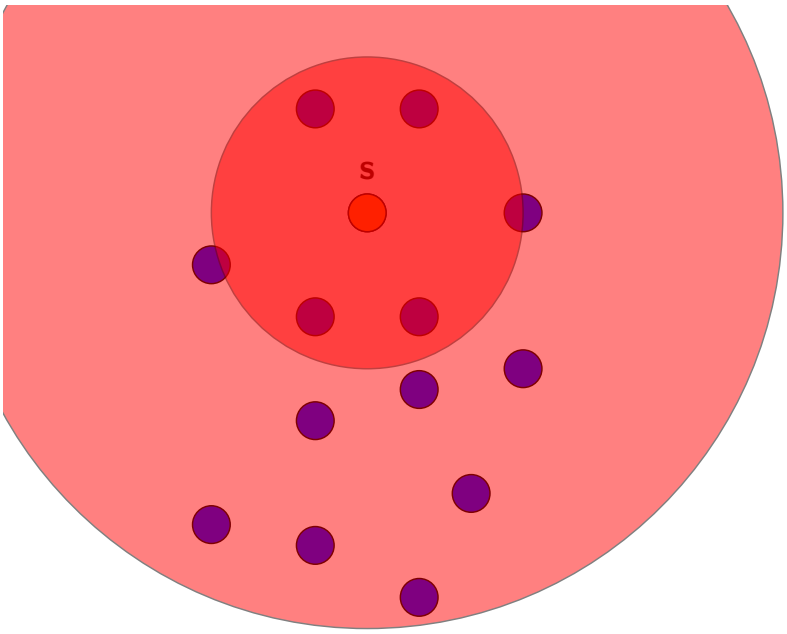
Layer 2 protocols



Typical sensor network sketches

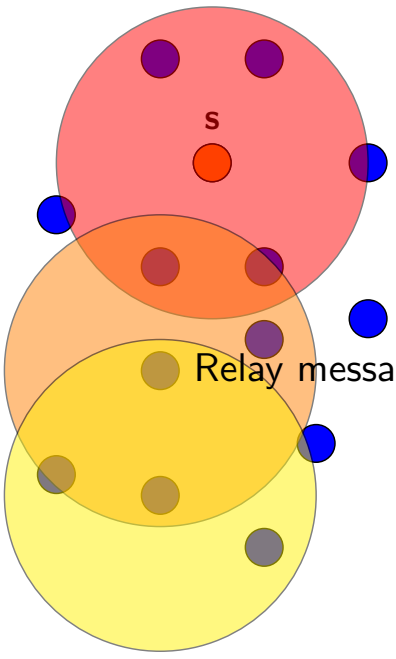
Layer 2 protocols





Increase radio range

Typical sensor network sketches
Layer 2 protocols



Relay message from node to node

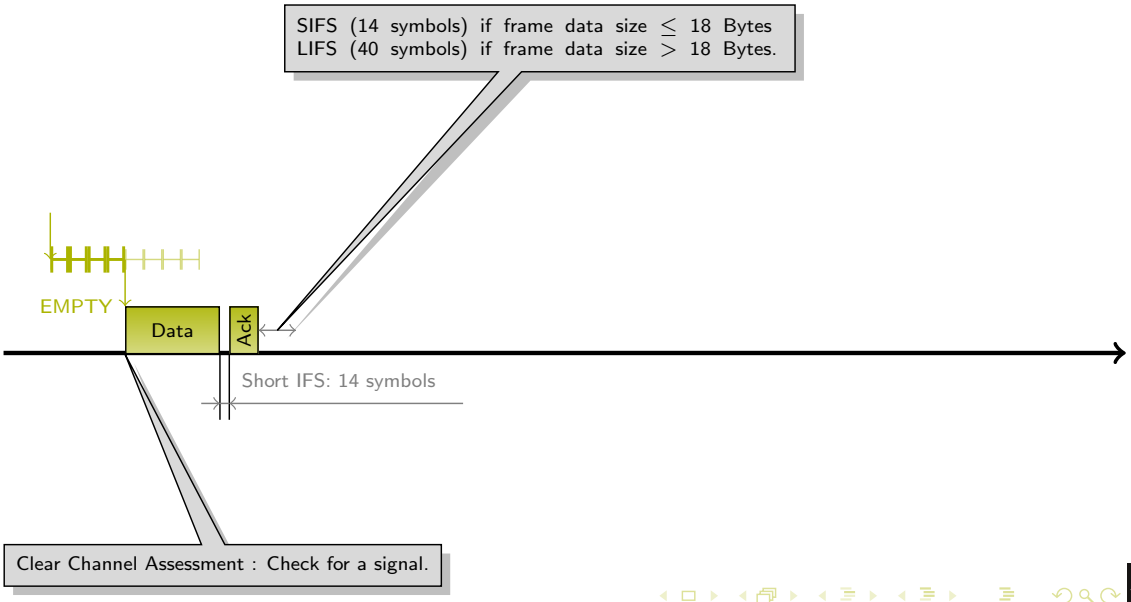
IEEE 802.15.4 CSMA/CA

Layer 2 protocols

IEEE 802.15.4 CSMA/CA

Layer 2 protocols

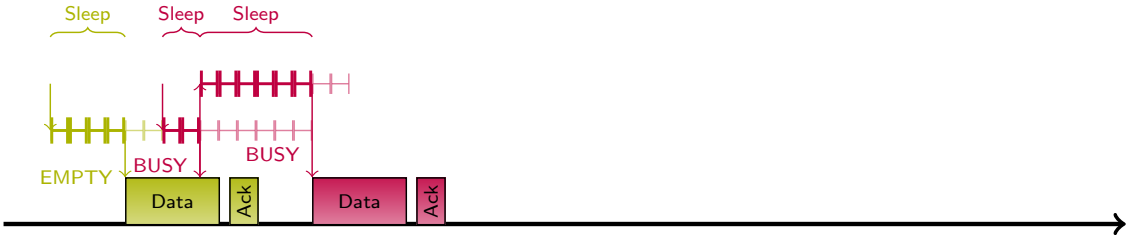
IEEE 802.15.4 CSMA/CA
Layer 2 protocols



IEEE 802.15.4 CSMA/CA
Layer 2 protocols

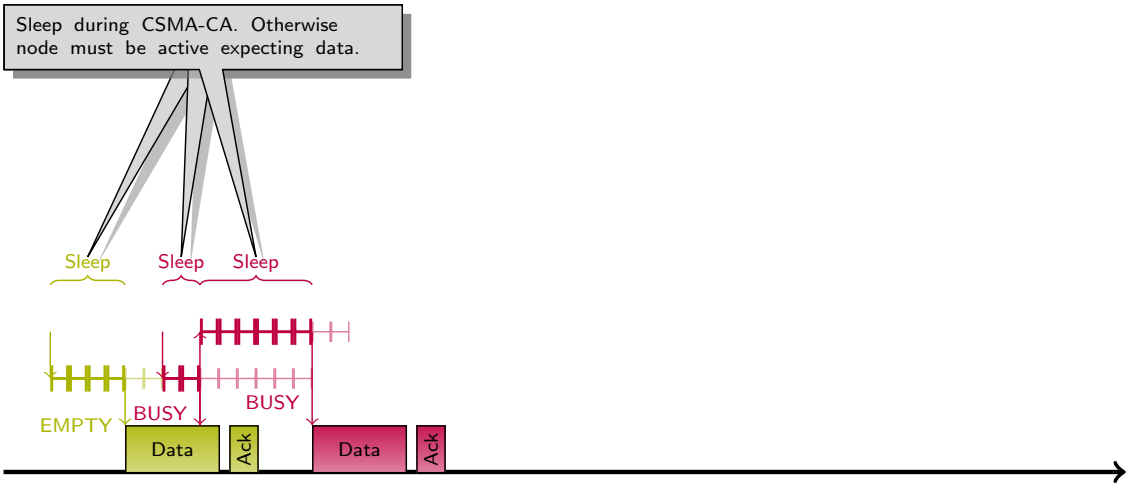
IEEE 802.15.4 CSMA/CA

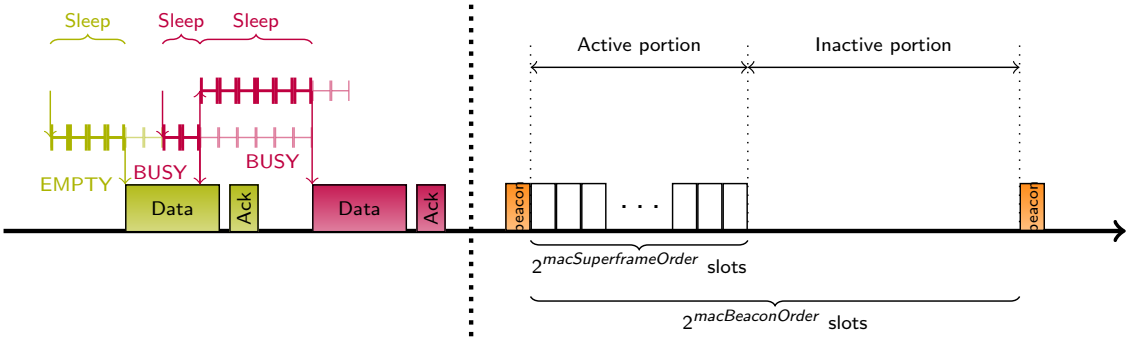
Layer 2 protocols



IEEE 802.15.4 CSMA/CA

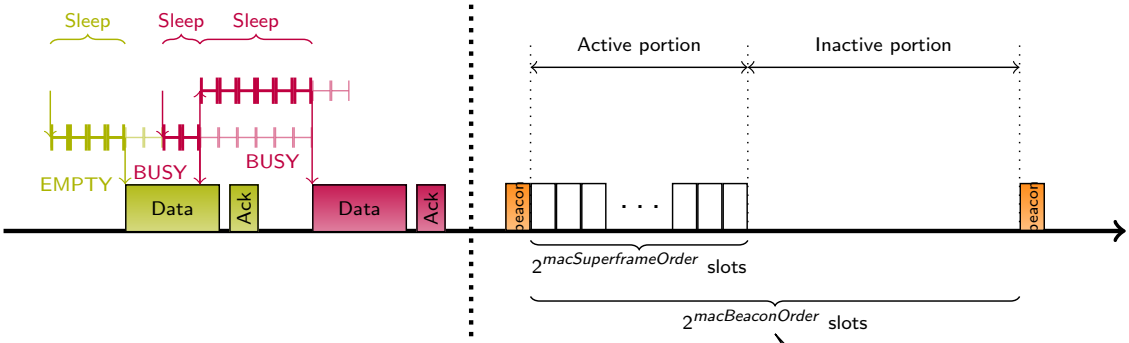
Layer 2 protocols





Beacon Less | Beacon

IEEE 802.15.4 CSMA/CA
Layer 2 protocols

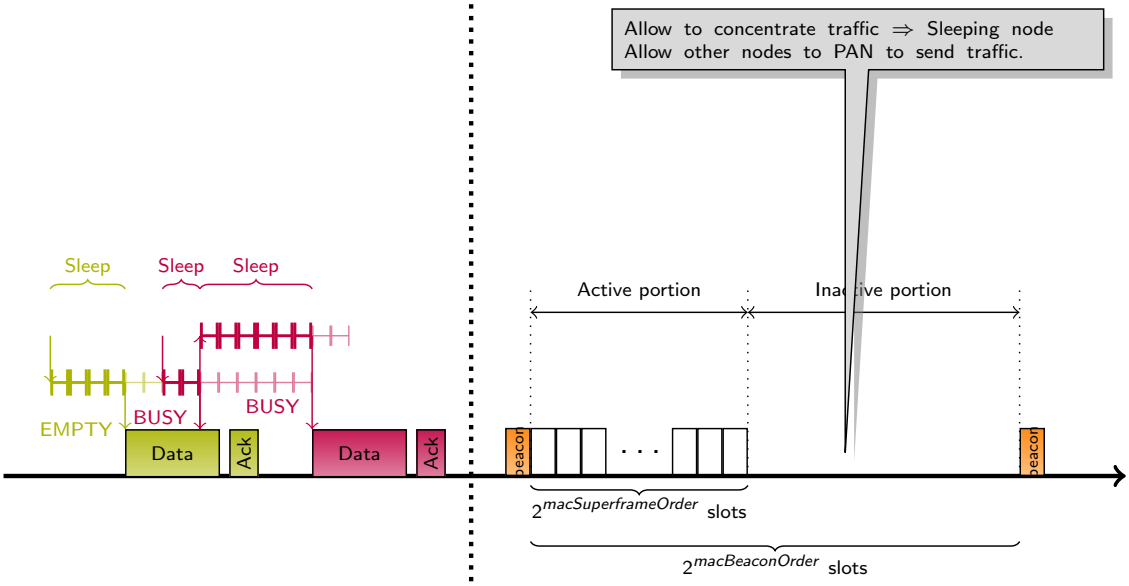


Beacon Less | Beacon

$0 \leq macSuperframerOrder \leq macBeaconOrder \leq 14$

IEEE 802.15.4 CSMA/CA

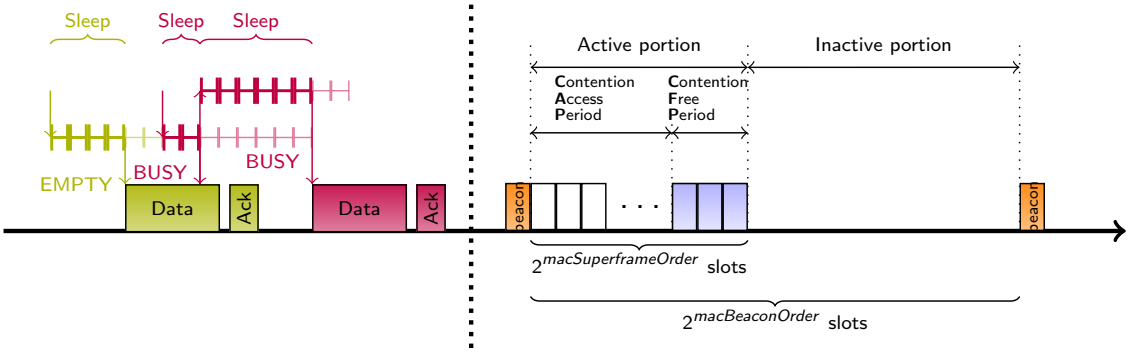
Layer 2 protocols



Beacon Less : Beacon

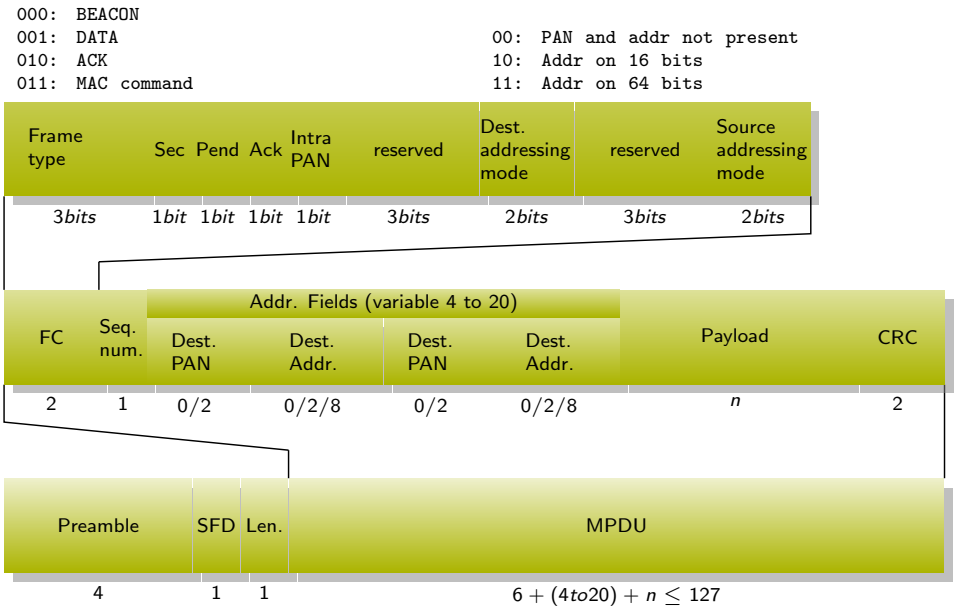
IEEE 802.15.4 CSMA/CA

Layer 2 protocols



Beacon Less : Beacon

IEEE 802.15.4 General Frame Format



IPv6 Addresses

IPv6 Benefits

Addresses

- Larger address space from 2^{32} to 2^{128}
  - Permanent address
- Stateless auto-configuration of hosts
  - Layer 3 "Plug & Play" Protocol
- Simple header ⇒ Efficient routing
  - No checksum
  - No fragmentation by routers
  - Enhanced extension system
-
-
-
-

IPv6 Benefits

Addresses

- Larger address space from 2^{32} to 2^{128}
  - Permanent address
- Stateless auto-configuration of hosts
  - Layer 3 "Plug & Play" Protocol
- Simple header ⇒ Efficient routing
  - No checksum
  - No fragmentation by routers
  - Enhanced extension system
- end to end, but...
-
-
-

IPv6 Benefits

Addresses

- Larger address space from 2^{32} to 2^{128}
  - Permanent address
- Stateless auto-configuration of hosts
  - Layer 3 "Plug & Play" Protocol
- Simple header ⇒ Efficient routing
  - No checksum
  - No fragmentation by routers
  - Enhanced extension system
- end to end, but...
- ~~Quality of service~~
-
-

IPv6 Benefits

Addresses

- Larger address space from 2^{32} to 2^{128}
  - Permanent address
- Stateless auto-configuration of hosts
  - Layer 3 "Plug & Play" Protocol
- Simple header ⇒ Efficient routing
  - No checksum
  - No fragmentation by routers
  - Enhanced extension system
- end to end, but...
- ~~Quality of service~~
- ~~Better support of mobility~~
-

IPv6 Benefits

- Larger address space from 2^{32} to 2^{128}
  - Permanent address
- Stateless auto-configuration of hosts
  - Layer 3 "Plug & Play" Protocol
- Simple header $\Rightarrow$ Efficient routing
  - No checksum
  - No fragmentation by routers
  - Enhanced extension system
- end to end, but...
- ~~Quality of service~~
- ~~Better support of mobility~~
- ~~IPsec~~

Notation

IPv6 addresses

F2C:544:9E::2:EF8D:6B7 F692:: A:1455::A:6E0 D:63:D::4:3A:55F B33:C::F2 7:5059:3D:C0::

9D::9BAC:B8CA:893F:80 1E:DE2:4C83::4E:39:F35:C875 2:: A:FDE3:76:B4F:D9D:: D6::

369F:9:F8:DBF::2 DD4:B45:1:C42F:BE6:75:: 9D7B:7184:EF::3FB:BF1A:D80 FE9::B:3

EC:DB4:B:F:F11::E9:090 83:B9:08:B5:F:3F:AF:B84 E::35B:8572:7A3:FB2 99:F:9:8B76::BC9

D64:07:F394::BDB:DF40:08EE:A79E AC:23:5D:78::233:84:8 F0D:F::F4EB:0F:5C7 E71:F577:ED:E:9DE8::

B::3 1D3F:A0AA:: 70:8EA1::8:D5:81:2:F302 26::8880:7 93:: F::9:0 E:2:0:266B::

763E:C:2E:1EB:F6:F4:14:16 E6:6:F4:B6:A888:979E:D78:09 9:754:5:90:0A78:A1A3:1:7 2:8::

97B:C4::C36 A40:7:5:7E8F:0:32EC:9A:D0 8A52::575 D::4CB4:E:2BF:5485:8CE 07:5::41 6B::A9:C

94FF:7B8::D9:51:26F 2::E:AE:ED:81 8241:: 5F97:: AD5B:259C:7DB8:24:58:552A:: 94:4:9FD:4:87E5::

5A8:2FF:1::CC EA:8904:7C:: 7C::D6B7:A7:B0:8B DC:6C::34:89 6C:1::5 7B3:6780:4:B1::E586

412:2:5E1:6DE5:5E3A:553:3:: 7F0:: B39::1:B77:DB 9D3:1F1:4B:3:B4E6:7681:09:D4A8 61:520::E0

1:28E9:0:095:DF:F2:: 1B61:4::1DE:50A 34BC:99::E9:9EFB E:EF:: BDC:672A:F4C8:A1::4:7:9CB7

C697:56AD:40:8:0::62

Notation

- Base format (a 16-octet Global IPv6 Address):
  - `2001:0db8:beef:0001:0000:0000:cafe:deca`
- Compact Format:

2001:0db8:beef:0001:0000:0000:cafe:deca

1. Remove 0 on the left of each word
2. To avoid ambiguity, substitute ONLY one sequence of zeros by `::`

- an IPv4 address may also appear : `::ffff:192.0.2.1`

Warning:

`2001:db8:3::/40` is in fact `2001:db8:0003::/40` and not `2001:db8:0300::/40`

Notation

- Base format (a 16-octet Global IPv6 Address):
  - `2001:0db8:beef:0001:0000:0000:cafe:deca`
- Compact Format:

2001:0db8:beef:0001:0000:0000:cafe:deca

1. Remove 0 on the left of each word
2. To avoid ambiguity, substitute ONLY one sequence of zeros by `::`

- an IPv4 address may also appear : `::ffff:192.0.2.1`

Warning:

`2001:db8:3::/40` is in fact `2001:db8:0003::/40` and not `2001:db8:0300::/40`

Notation

Addresses

- Base format (a 16-octet Global IPv6 Address):
  - `2001:0db8:beef:0001:0000:0000:cafe:deca`
- Compact Format:

2001:db8:beef:1:0:0:cafe:deca

1. Remove 0 on the left of each word
2. To avoid ambiguity, substitute ONLY one sequence of zeros by `::`

- an IPv4 address may also appear : `::ffff:192.0.2.1`

Warning:

`2001:db8:3::/40` is in fact `2001:db8:0003::/40` and not `2001:db8:0300::/40`

Slide 37 Laurent Toutain Filière 2 TELECOM Bretagne

Notation

- Base format (a 16-octet Global IPv6 Address):
  - `2001:0db8:beef:0001:0000:0000:cafe:deca`
- Compact Format:

2001:db8:beef:1:0:0:cafe:deca

1. Remove 0 on the left of each word
2. To avoid ambiguity, substitute ONLY one sequence of zeros by `::`

- an IPv4 address may also appear : `::ffff:192.0.2.1`

Warning:

`2001:db8:3::/40` is in fact `2001:db8:0003::/40` and not `2001:db8:0300::/40`

Notation

Addresses

- Base format (a 16-octet Global IPv6 Address):
  - `2001:0db8:beef:0001:0000:0000:cafe:deca`
- Compact Format:

2001:db8:beef:1::cafe:deca

1. Remove 0 on the left of each word
2. To avoid ambiguity, substitute ONLY one sequence of zeros by `::`

- an IPv4 address may also appear : `::ffff:192.0.2.1`

Warning:

`2001:db8:3::/40` is in fact `2001:db8:0003::/40` and not `2001:db8:0300::/40`

Slide 37 Laurent Toutain Filière 2 TELECOM Bretagne

Notation

- Base format (a 16-octet Global IPv6 Address):
  - `2001:0db8:beef:0001:0000:0000:cafe:deca`
- Compact Format:

2001:db8:beef:1::cafe:deca

1. Remove 0 on the left of each word
2. To avoid ambiguity, substitute ONLY one sequence of zeros by `::`

- an IPv4 address may also appear : `::ffff:192.0.2.1`

Warning:

`2001:db8:3::/40` is in fact `2001:db8:0003::/40` and not `2001:db8:0300::/40`

Is it enough for the future ?

- Address length
  - About $3.4x10^{38}$ addresses
  - 60 000 trillion trillion addresses per inhabitant on earth
  - Addresses for every grain of sands in the world
  - IPv4: 6 addresses per US inhabitant, 1 in Europe, 0.01 in China and 0.001 in India
- Justification of a fixed-length address

Warning:

- An address for everything on the network and not an address for everything
- No addresses for the whole life:
  - Depends on your position on the network
  - ISP Renumbering may be possible

Addressing scheme

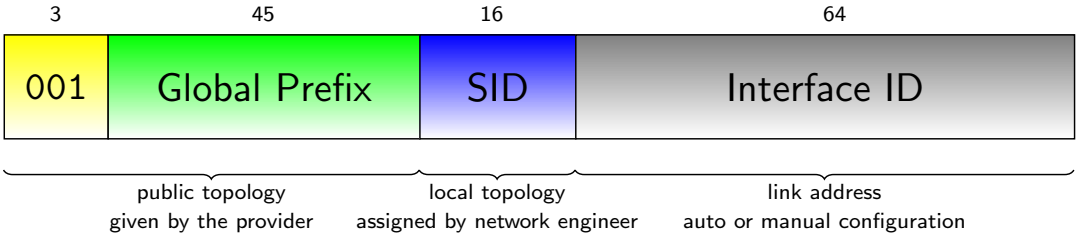
Addressing scheme

Addresses

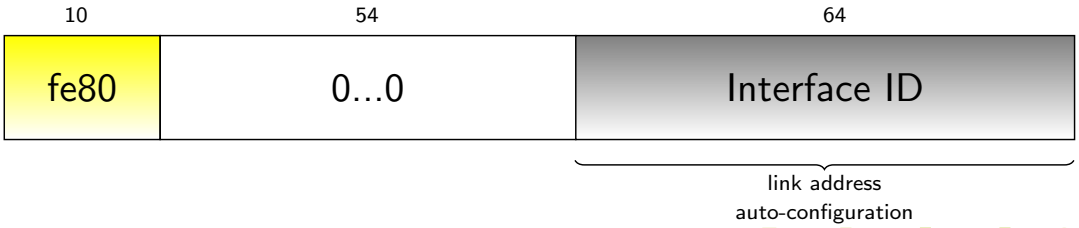
- RFC 4291 defines current IPv6 addresses
  - loopback (`::1`)
  - `link local (fe80::/10)`
  - `global unicast (2000::/3)`
  - `multicast (ff00::/8)`
- Use CIDR principles:
  - Prefix / prefix length notation
  - `2001:db8:face::/48`
  - `2001:db8:face:bed:cafe:deca:dead:beef/64`
- Interfaces have several IPv6 addresses
  - at least a link-local and a global unicast addresses

Address Format

Global Unicast Address:



Link-Local Address:



Interface Identifier

Interface ID can be selected differently

- Derived from a Layer 2 ID (I.e. MAC address) :
  - for Link Local address
  - for Global Address : plug-and-play hosts
- Assigned manually :
  - to keep same address when Ethernet card or host is changed
  - to remember easily the address
    - 1, 2, 3, ...
    - last digit of the v4 address
    - the IPv4 address (for nostalgic system administrators)
    - ...

Interface Identifier

Interface ID can be selected differently

- Derived from a Layer 2 ID (I.e. MAC address) :
  - for Link Local address
  - for Global Address : plug-and-play hosts
- Assigned manually :
  - to keep same address when Ethernet card or host is changed
  - to remember easily the address
    - 1, 2, 3, ...
    - last digit of the v4 address
    - the IPv4 address (for nostalgic system administrators)
    - ...

Interface Identifier

Interface ID can be selected differently

- Random value :
  - Changed frequently (e.g, every day, per session, at each reboot...) to guarantee anonymity
- Hash of other values (experimental) :
  - To link address to other properties
  - Public key
  - List of assigned prefixes
  - ...

Interface Identifier

Interface ID can be selected differently

- Random value :
  - Changed frequently (e.g, every day, per session, at each reboot...) to guarantee anonymity
- Hash of other values (experimental) :
  - To link address to other properties
  - Public key
  - List of assigned prefixes
  - ...

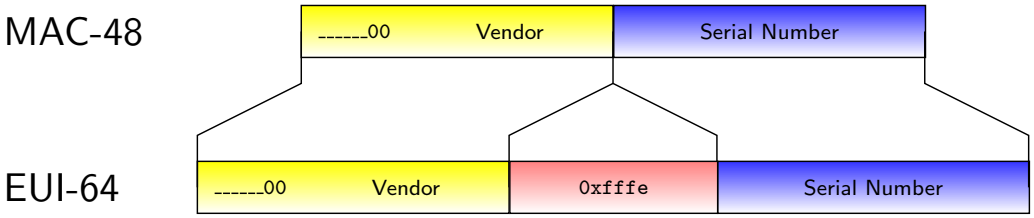
How to Construct an IID from MAC Address

- 64 bits is compatible with EUI-64 (i.e. IEEE 1394 FireWire, ...)
- IEEE propose a way to transform a MAC-48 to an EUI-64
- U/L changed for numbering purpose

- There is no conflicts if IID are manually numbered: 1, 2, 3, ...

How to Construct an IID from MAC Address

- 64 bits is compatible with EUI-64 (i.e. IEEE 1394 FireWire, ...)
- IEEE propose a way to transform a MAC-48 to an EUI-64
- U/L changed for numbering purpose

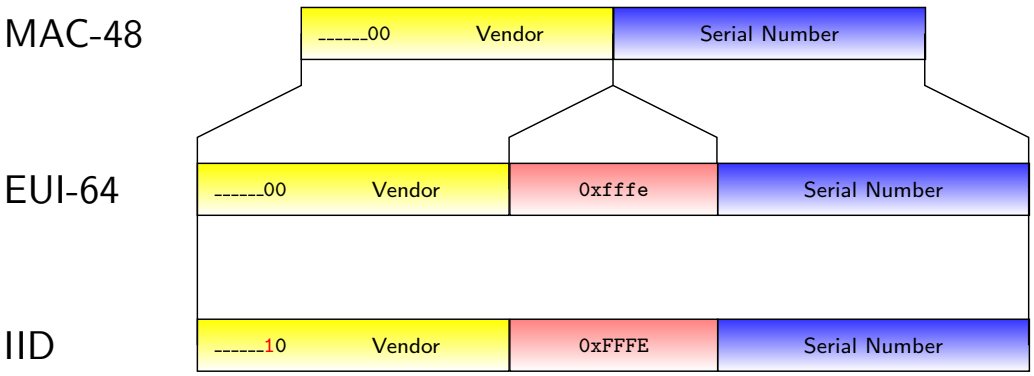


- There is no conflicts if IID are manually numbered: 1, 2, 3, ...

How to Construct an IID from MAC Address

Addresses

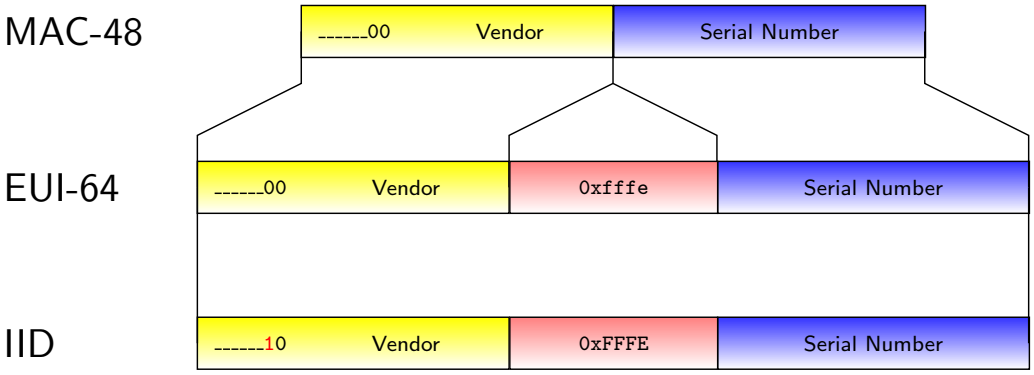
- 64 bits is compatible with EUI-64 (i.e. IEEE 1394 FireWire, ...)
- IEEE propose a way to transform a MAC-48 to an EUI-64
- U/L changed for numbering purpose



- There is no conflicts if IID are manually numbered: 1, 2, 3, ...

How to Construct an IID from MAC Address

- 64 bits is compatible with EUI-64 (i.e. IEEE 1394 FireWire, ...)
- IEEE propose a way to transform a MAC-48 to an EUI-64
- U/L changed for numbering purpose



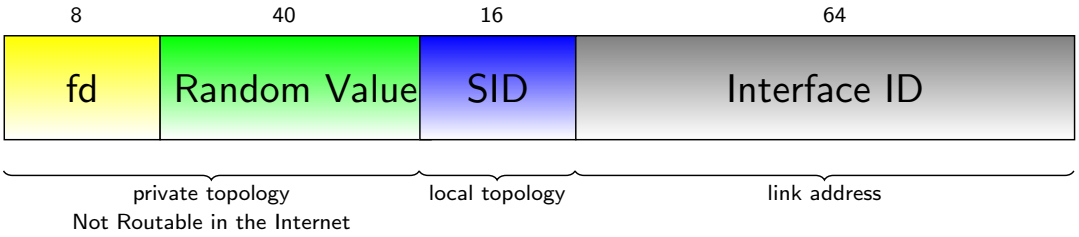
- There is no conflicts if IID are manually numbered: 1, 2, 3, ...

Kind of addresses

Other kind of addresses : ULA (RFC 4193)

- Equivalent to the private addresses in IPv4
- But try to avoid same prefixes on two different sites:
  - avoid renumbering if two company merge
  - avoid ambiguities when VPN are used
- These prefixes are not routable on the Internet

Unique Local IPv6 Unicast Addresses:



http://www.sixxs.net/tools/grh/ula/ to create your own ULA prefix.

Multicast

Generic Format:

- T (Transient) 0: well known address - 1: temporary address
- P (Prefix) 1 : assigned from a network prefix (T must be set to 1)
- R (Rendez Vous Point) 1: contains the RP address (P & T set to 1)
- Scope :
  - 1 - interface-local
  - 2 - link-local
  - 3 - reserved
  - 4 - admin-local
  - 5 - site-local
  - 8 - organisation-local
  - e - global
  - f - reserved

Some Well Known Multicast Addresses


```
ff02:0:0:0:0:0:0:1 All Nodes Address (link-local scope)
ff02:0:0:0:0:0:0:2 All Routers Address
ff02:0:0:0:0:0:0:5 OSPFIGP
ff02:0:0:0:0:0:0:6 OSPFIGP Designated Routers
ff02:0:0:0:0:0:0:9 RIP Routers
ff02:0:0:0:0:0:0:fb mDNSv6
ff02:0:0:0:0:0:1:2 All-dhcp-agents
ff02:0:0:0:0:1:ffxx:xxxx Solicited-Node Address
ff05:0:0:0:0:0:1:3 All-dhcp-servers (site-local scope)
```

http://www.iana.org/assignments/ipv6-multicast-addresses

Some Well Known Multicast Addresses

Addresses

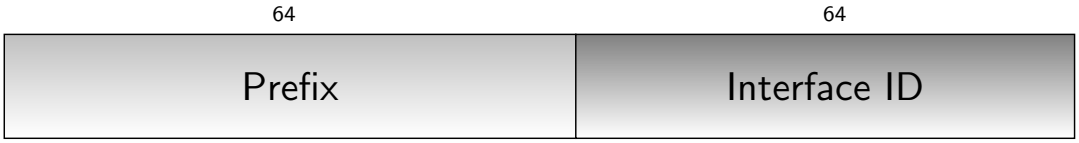

```
ff02:0:0:0:0:0:0:1 All Nodes Address (link-local scope)
ff02:0:0:0:0:0:0:2 All Routers Address
ff02:0:0:0:0:0:0:5 OSPFIGP
ff02:0:0:0:0:0:0:6 OSPFIGP Designated Routers
ff02:0:0:0:0:0:0:9 RIP Routers
ff02:0:0:0:0:0:0:fb mDNSv6
ff02:0:0:0:0:0:1:2 All-dhcp-agents
ff02:0:0:0:0:1:ffxx:xxxx Solicited-Node Address
ff05:0:0:0:0:0:1:3 All-dhcp-servers (site-local scope)
```

http://www.iana.org/assignments/ipv6-multicast-addresses

Slide 48 Laurent Toutain Filière 2

Anycast

- In the same addressing space as unicast
- No way to distiguish them
- Two anycast families:
  - Same prefix on Internet
    - same a IPv4 anycast for DNS or 6to4
  - Same address on the link
    - Must avoid DAD
    - Some IID values are reserved
    - All IID bits to 1 except last byte
    - Only `0x7E` Mobile Home Agent
  - May more addresses with Wireless Sensor Network ?
    - Temperature, presence...



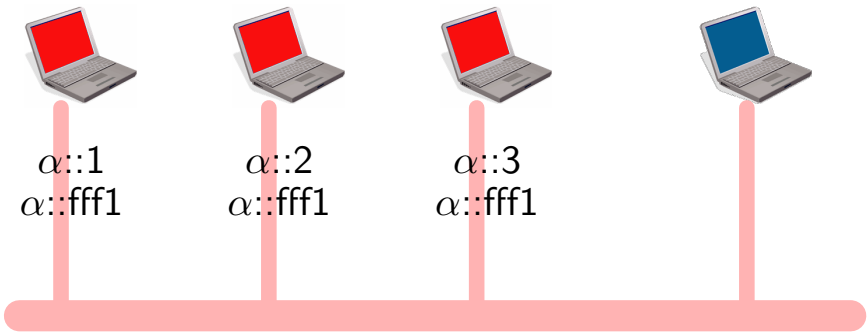
Anycast on prefix : Example from Renater


```
#traceroute6 2001:500:2f::f
traceroute6 to 2001:500:2f::f (2001:500:2f::f) from 2001:660:7301:3103:223:6cff
30 hops max, 12 byte packets
1 2001:660:7301:3103::1 4.774 ms 1.198 ms 2.764 ms
2 2001:660:7301:3036::1 3.364 ms 2.215 ms 1.417 ms
3 vl856-gi9-9-rennes-rtr-021.noc.renater.fr 2.892 ms 6.794 ms 2.195 ms
4 te4-1-caen-rtr-021.noc.renater.fr 7.706 ms 5.1 ms 4.193 ms
5 te4-1-rouen-rtr-021.noc.renater.fr 6.527 ms 6.296 ms 6.661 ms
6 te0-0-0-1-paris1-rtr-001.noc.renater.fr 8.702 ms 10.26 ms 8.696 ms
7 F-root-server.sfinx.tm.fr 8.495 ms 8.607 ms 8.664 ms
8 f.root-servers.net 8.738 ms 9.171 ms 8.702 ms
```

Anycast on prefix : Example from Hawaï


```
#traceroute6 2001:500:2f::f
traceroute6 to 2001:500:2f::f (2001:500:2f::f) from 2001:1888:0:1:2d0:b7ff:fe7d
64 hops max, 12 byte packets
1 apapane-fe0-0-1 1.169 ms 0.970 ms 0.947 ms
2 r1.mdtnj.ipv6.att.net 121.159 ms 121.737 ms 121.378 ms
3 bbr01-p1-0.nwrk01.occaid.net 130.468 ms 129.640 ms 130.845 ms
4 bbr01-g1-0.asbn01.occaid.net 131.372 ms 131.596 ms 131.421 ms
5 bbr01-g1-0.atln01.occaid.net 144.937 ms 144.550 ms 144.834 ms
6 bbr01-p1-0.dlls01.occaid.net 166.709 ms 196.177 ms 165.983 ms
7 dcr01-p1-5.lsan01.occaid.net 138.437 ms 138.690 ms 138.544 ms
8 bbr01-g0-2.irvn01.occaid.net 138.552 ms 137.956 ms 137.649 ms
9 dcr01-g1-2.psdn01.occaid.net 137.629 ms 138.030 ms 141.332 ms
10 bbr01-f1-5.snfc02.occaid.net 138.501 ms 138.511 ms 137.483 ms
11 exit.sf-guest.sfo2.isc.org 147.941 ms 144.929 ms 145.956 ms
12 f.root-servers.net 139.063 ms 139.715 ms 142.571 ms
```


OnLink Anycast: Example

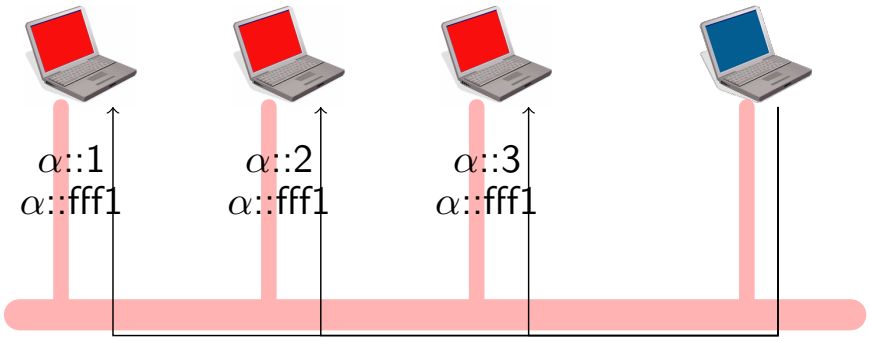


RFC 2526 Anycast values, all bit of IID set to 1 except last 8 bits:

- 0x7F: reserved
- 0x7E: Home Agent (Mobile IP)
- 0x00 to 0x7D: reserved

http://www.iana.org/assignments/ipv6-anycast-addresses

OnLink Anycast: Example



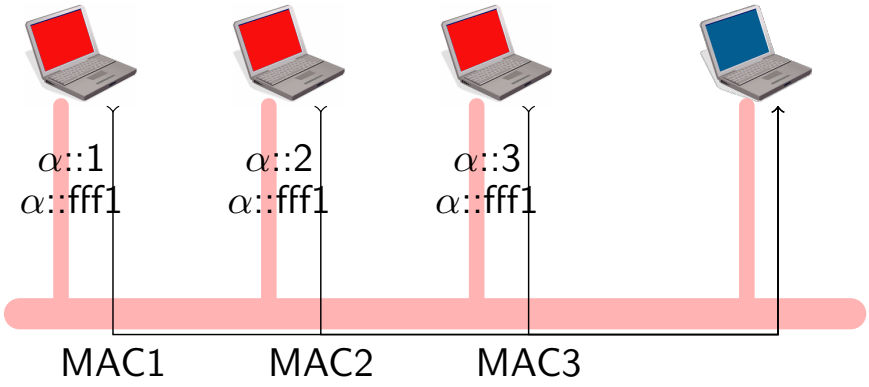
RFC 2526 Anycast values, all bit of IID set to 1 except last 8 bits:

- 0x7F: reserved
- 0x7E: Home Agent (Mobile IP)
- 0x00 to 0x7D: reserved

http://www.iana.org/assignments/ipv6-anycast-addresses

OnLink Anycast: Example

Addresses



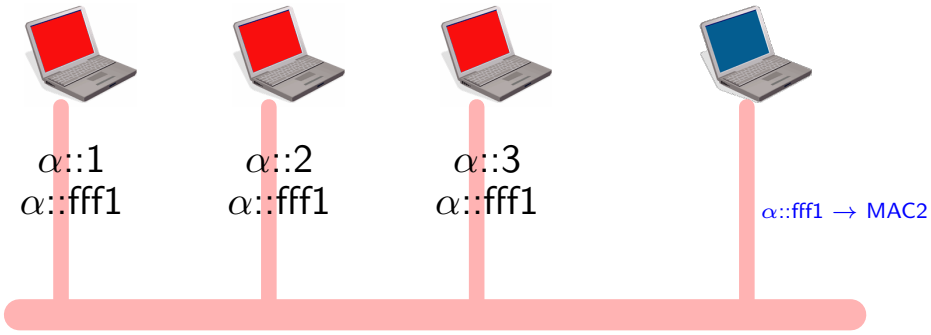
RFC 2526 Anycast values, all bit of IID set to 1 except last 8 bits:

- 0x7F: reserved
- 0x7E: Home Agent (Mobile IP)
- 0x00 to 0x7D: reserved

http://www.iana.org/assignments/ipv6-anycast-addresses

Slide 52 Laurent Toutain Filière 2 TELECOM Bretagne

OnLink Anycast: Example



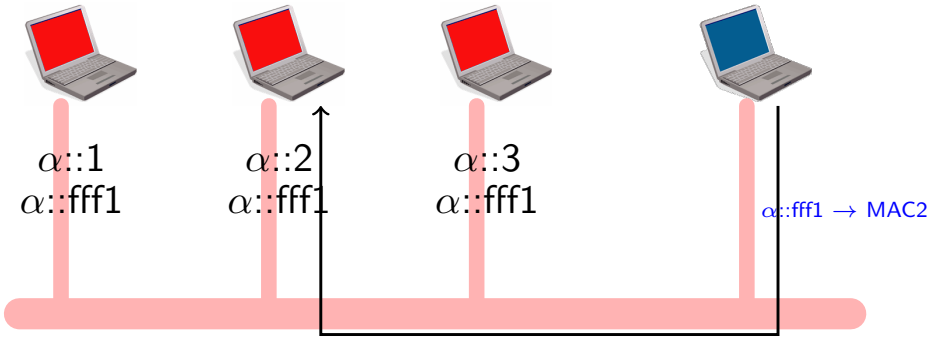
RFC 2526 Anycast values, all bit of IID set to 1 except last 8 bits:

- 0x7F: reserved
- 0x7E: Home Agent (Mobile IP)
- 0x00 to 0x7D: reserved

http://www.iana.org/assignments/ipv6-anycast-addresses

OnLink Anycast: Example

Addresses



RFC 2526 Anycast values, all bit of IID set to 1 except last 8 bits:

- 0x7F: reserved
- 0x7E: Home Agent (Mobile IP)
- 0x00 to 0x7D: reserved

http://www.iana.org/assignments/ipv6-anycast-addresses

Slide 52 Laurent Toutain Filière 2 TELECOM Bretagne

Anycast on prefix : Example from Hawaï

```
# ifconfig en3
en3: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
inet6 fe80::223:6cff:fe97:679c
inet 192.168.103.177 netmask 0xffffff00 broadcast 192.168.103.255
inet6 2001:660:7301:3103:223:65ff:fe97:679c prefixlen 64 autoconf
ether 00:23:6c:97:67:9c
media: autoselect status: active
supported media: autoselect
# ifconfig en3 inet6 2001:660:7301:3103:FF::FF anycast
# ifconfig en3
en3: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
inet6 fe80::223:6cff:fe97:679c
inet 192.168.103.177 netmask 0xffffff00 broadcast 192.168.103.255
inet6 2001:660:7301:3103:223:65ff:fe97:679c prefixlen 64 autoconf
inet6 2001:660:7301:3103:ff::ff prefixlen 64 anycast
ether 00:23:6c:97:67:9c
media: autoselect status: active
supported media: autoselect
```


IPv6 Packets

IPv6 Header

IPv6 Packet : Simpler

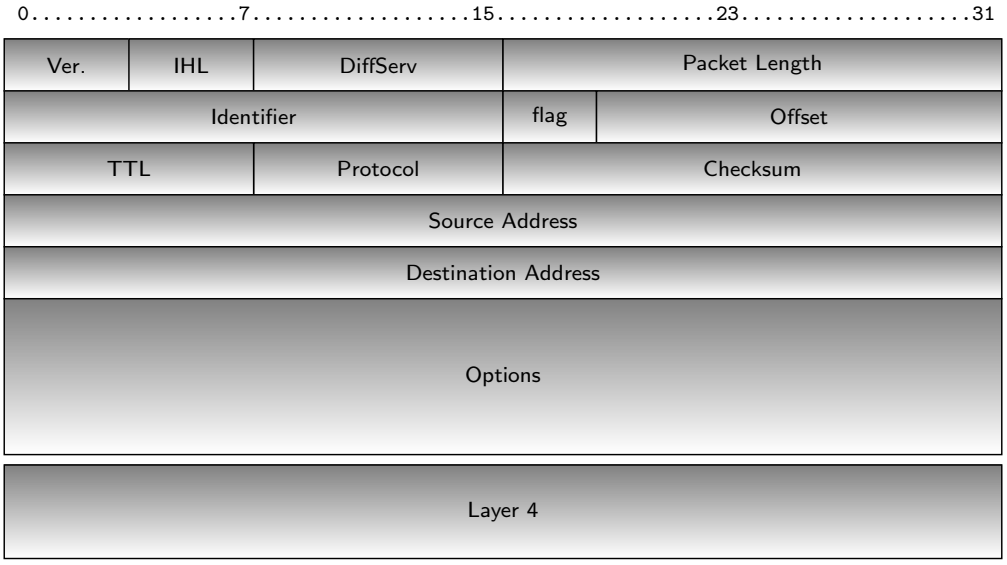
Definition

- IPv6 header follows the same IPv4 principle:
  - fixed address size ... but 4 times larger
  - alignment on 64 bit words (instead of 32)
- Features not used in IPv4 are removed
- Minimum MTU 1280 Bytes
  - If L2 cannot carry 1280 Bytes, then add an adaptation layer such as AAL5 for ATM or 6LoWPAN (RFC 4944) for IEEE 802.15.4.

Goal :

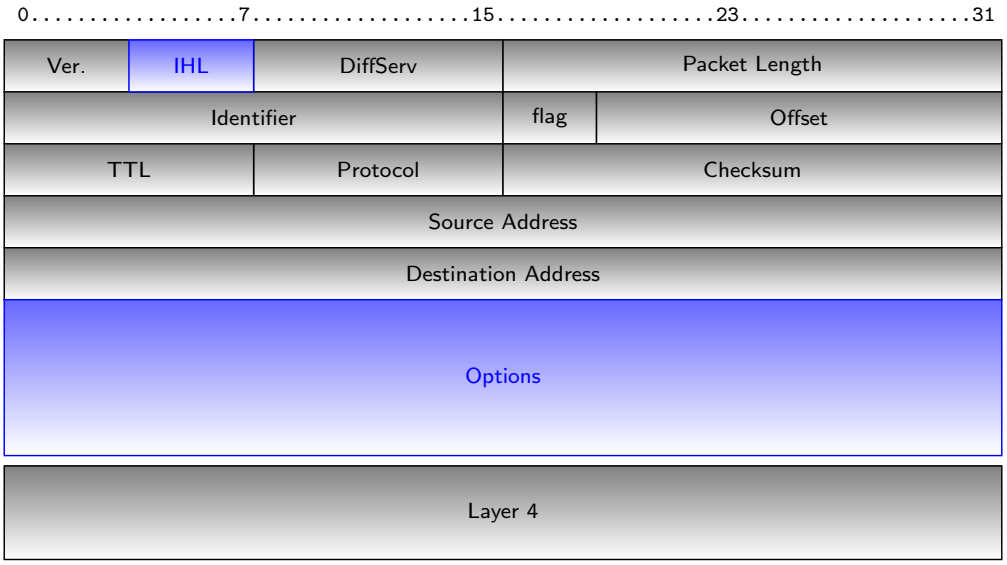
- Forward packet as fast as possible
- Less processing in routers
- More features at both ends

IPv4 Header

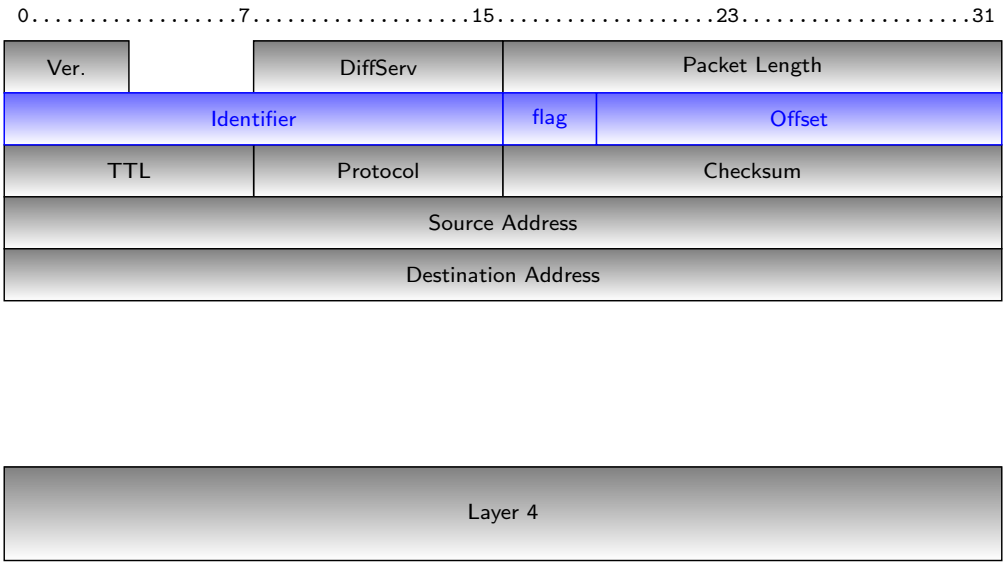


IPv4 Header

Protocol ▶ IPv6 Header

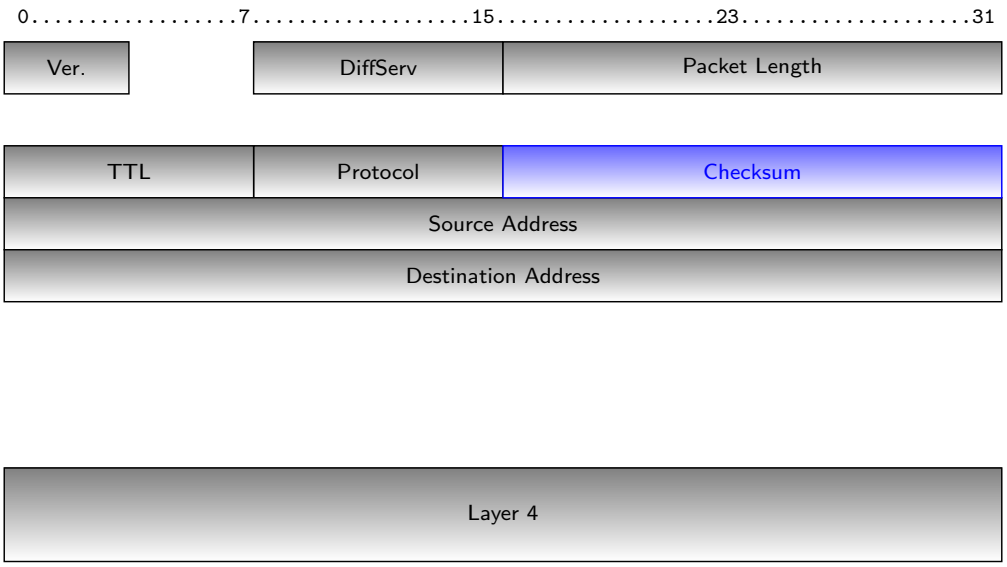


IPv4 Header

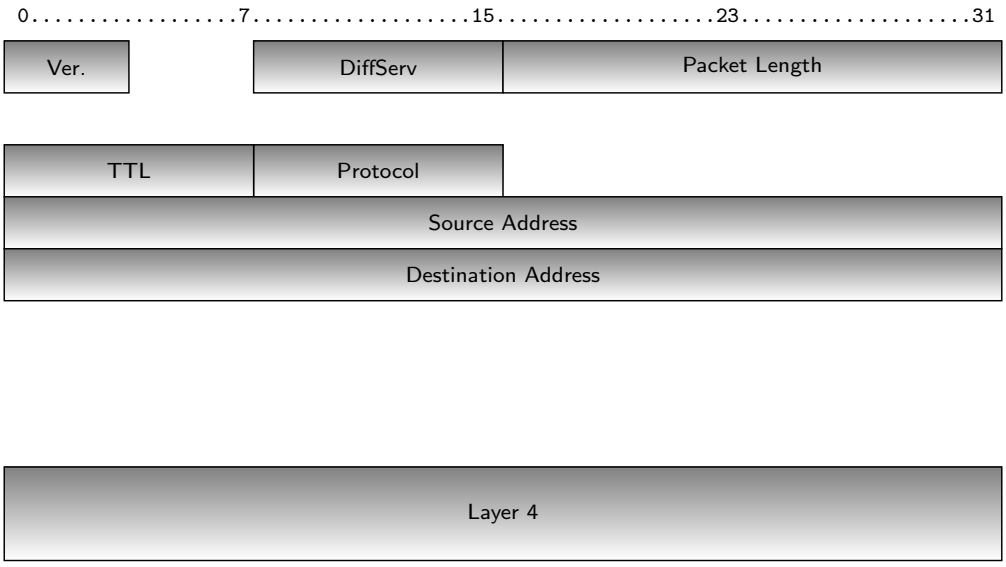


IPv4 Header

Protocol ▶ IPv6 Header

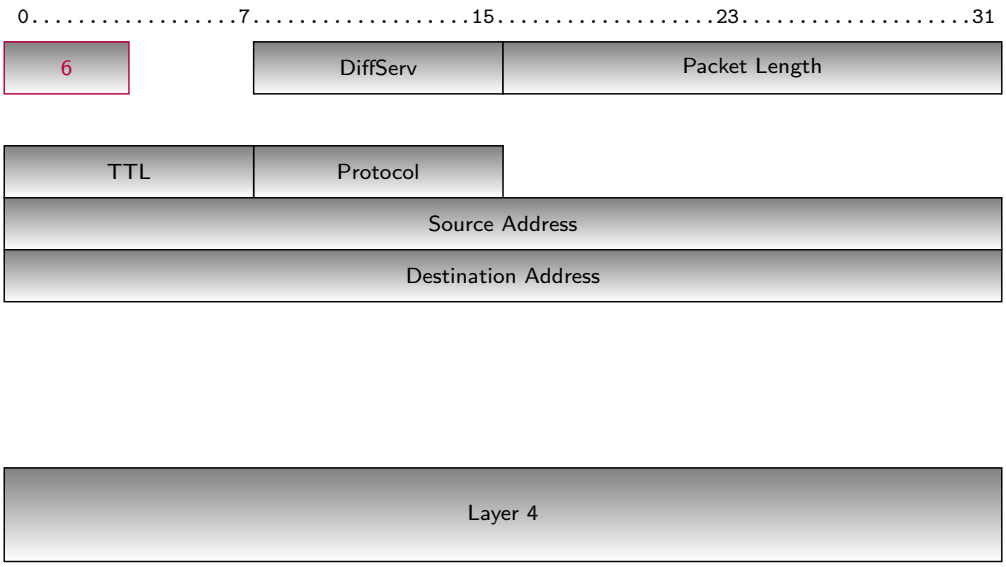


IPv4 Header

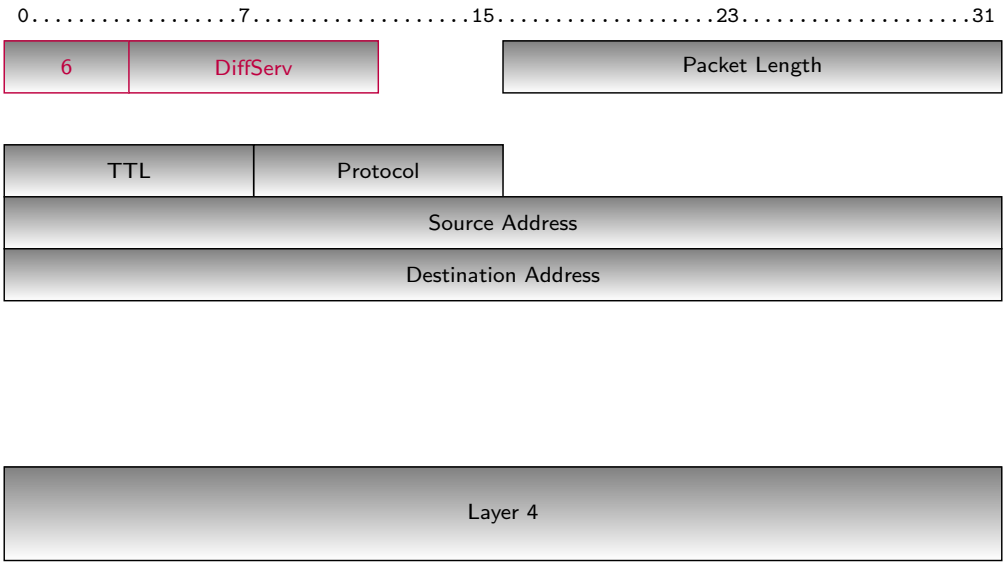


IPv4 Header

Protocol ▶ IPv6 Header

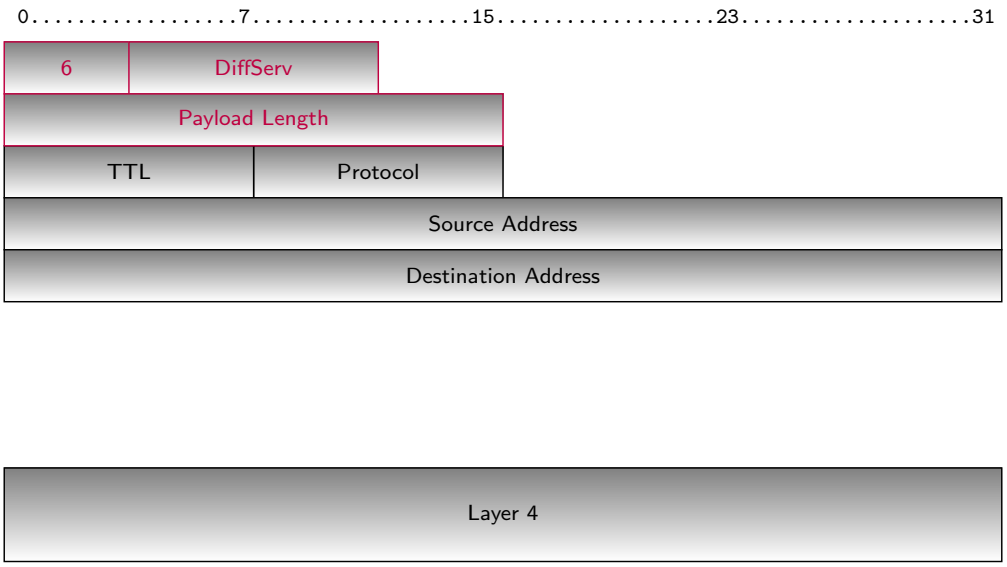


IPv4 Header

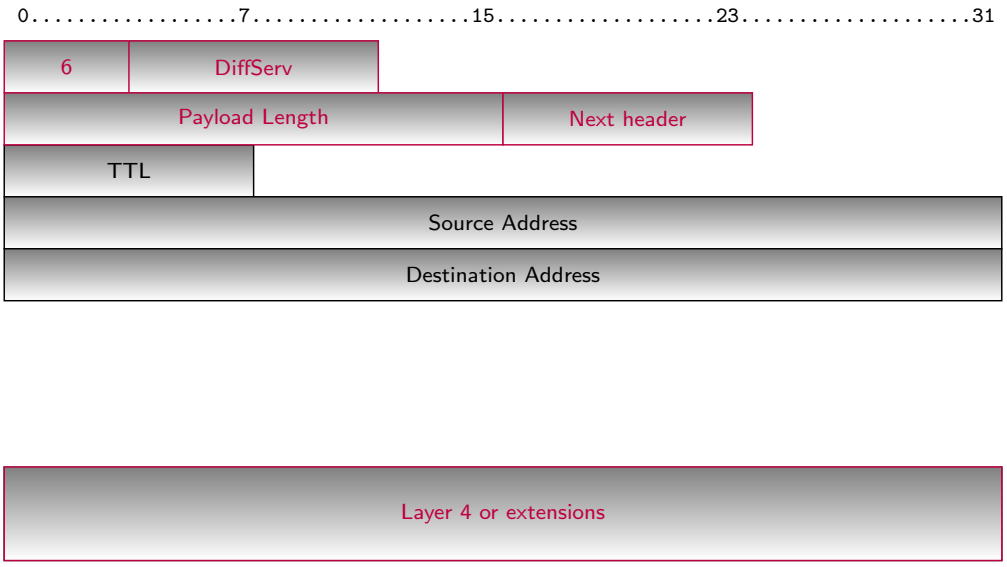


Protocol ▶ IPv6 Header

IPv4 Header

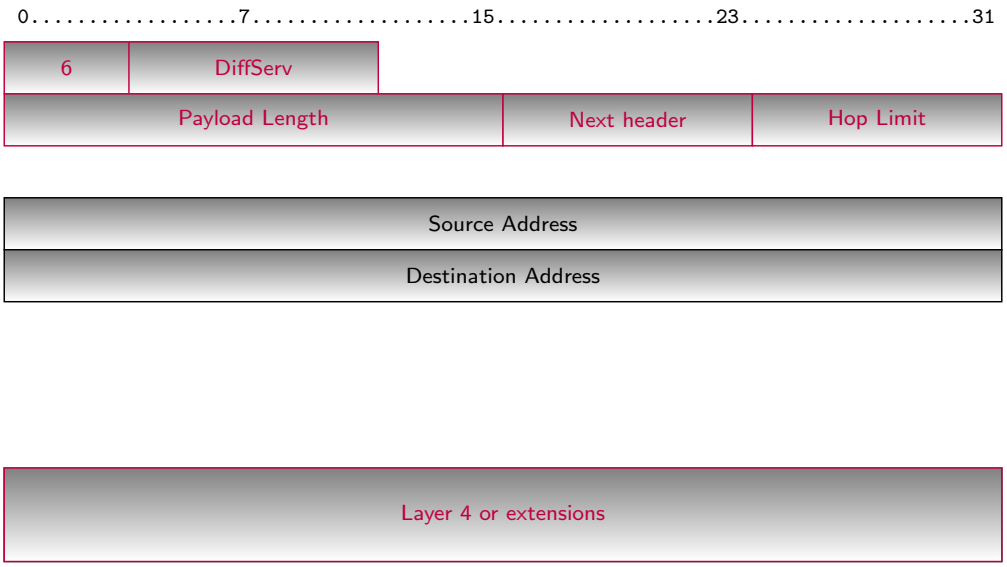


IPv4 Header

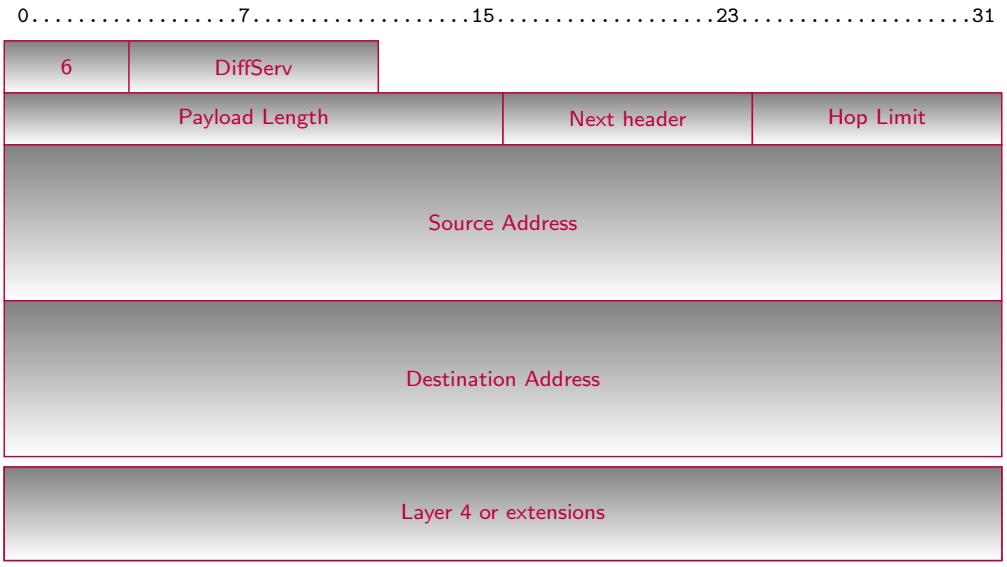


IPv4 Header

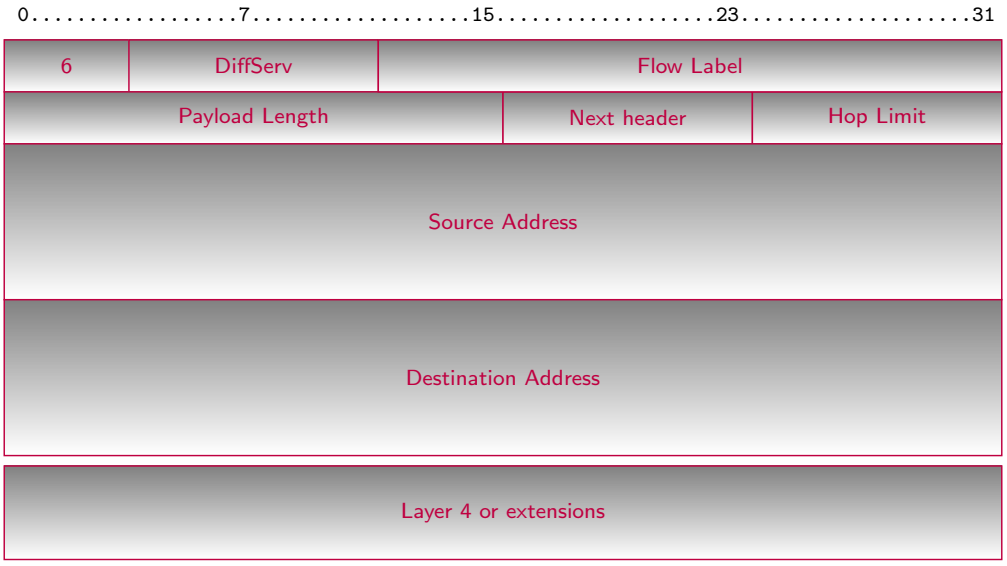
Protocol ▶ IPv6 Header



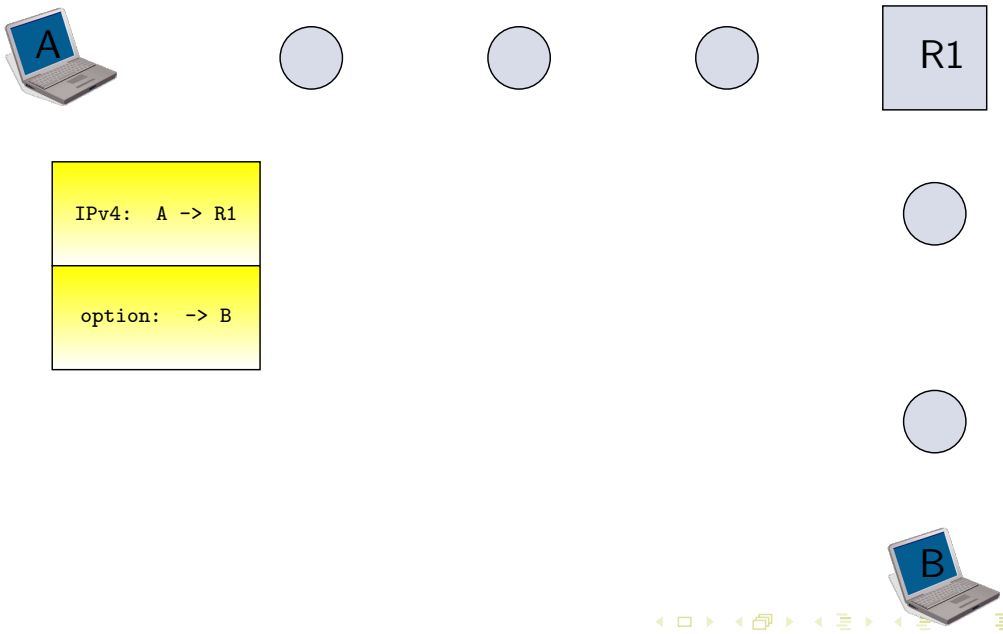
IPv4 Header



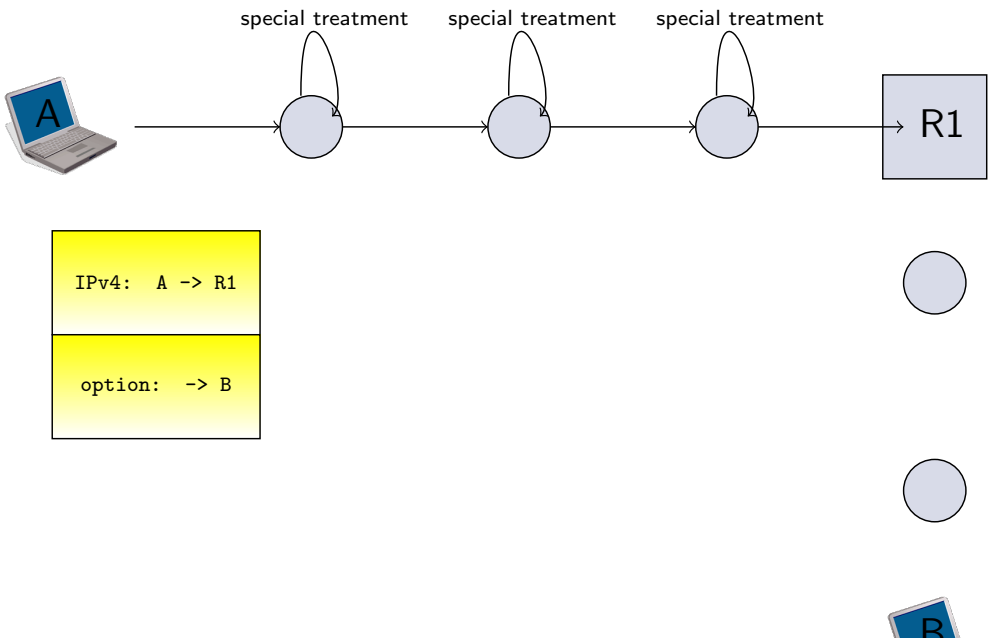
IPv6 Header



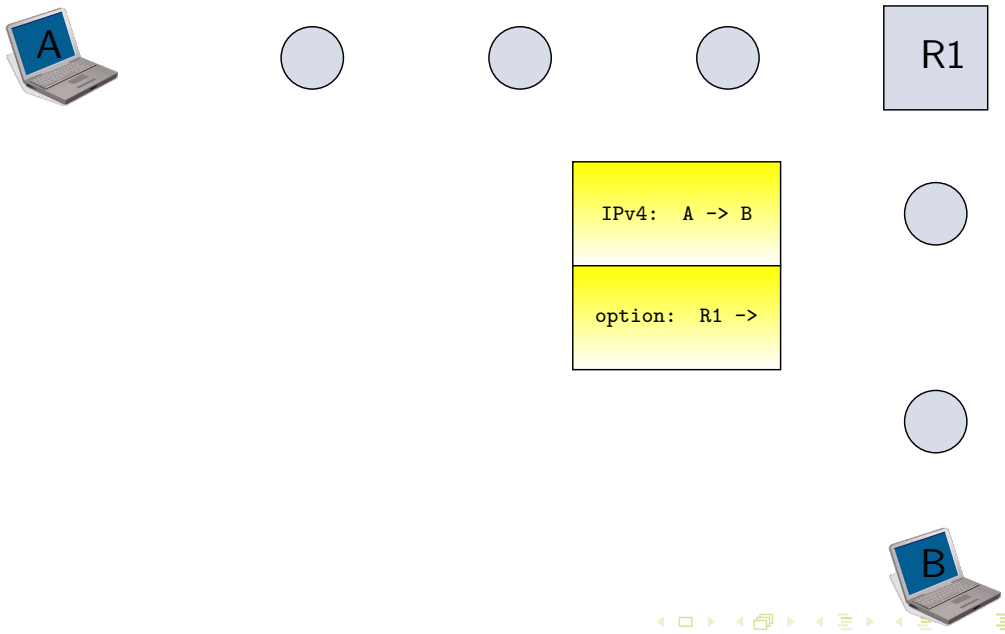
Extension Superiority



Extension Superiority

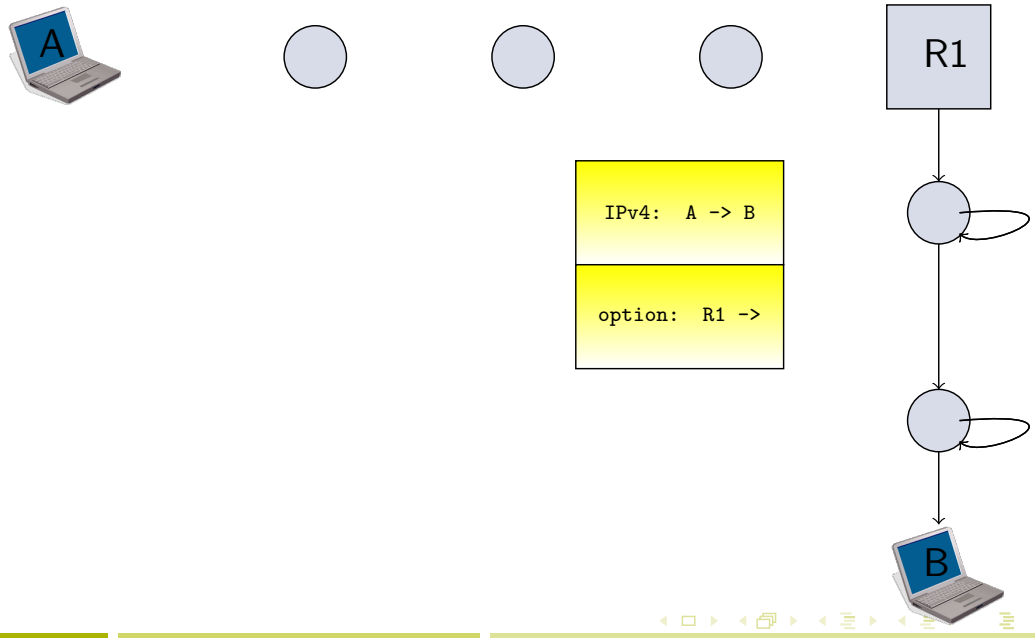


Extension Superiority

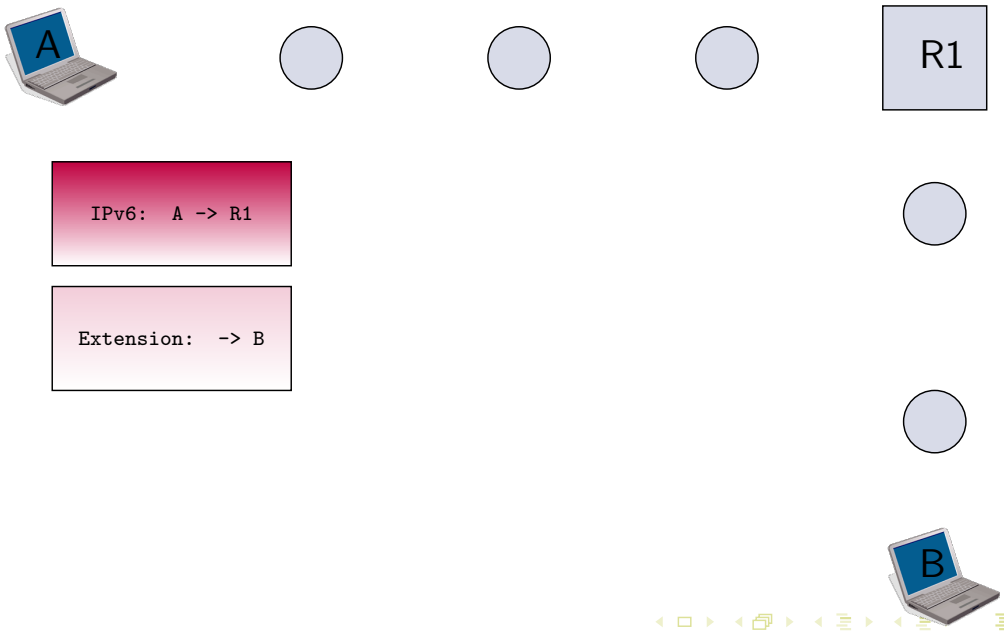


Extension Superiority

Protocol ▶ IPv6 Header

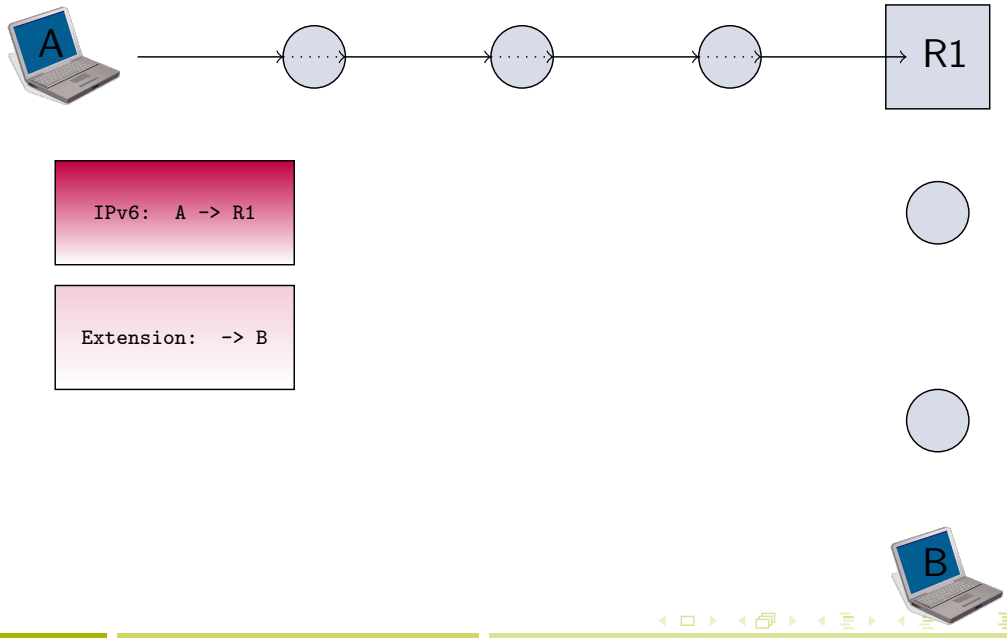


Extension Superiority

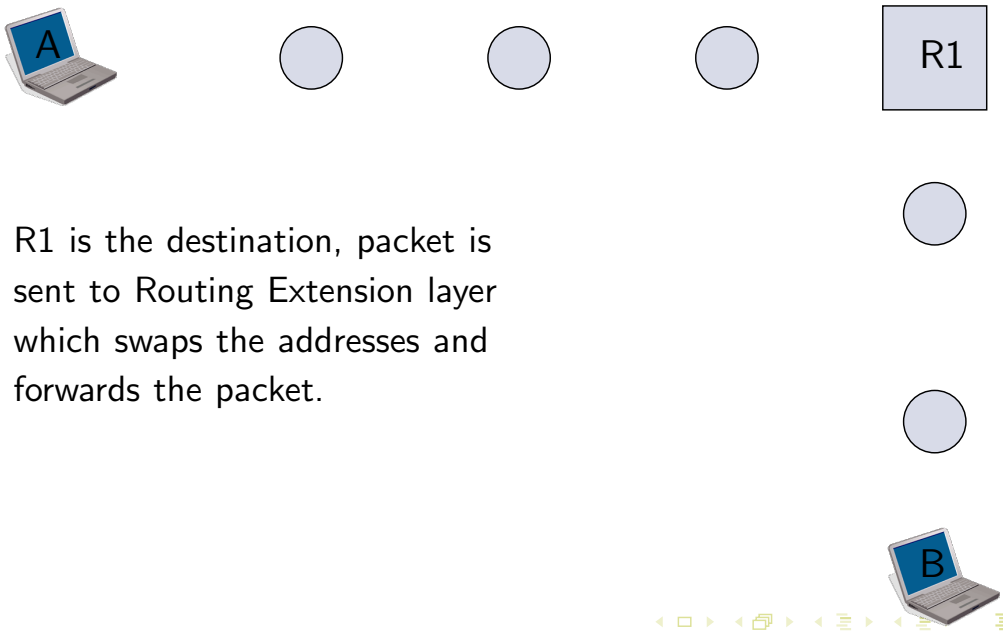


Extension Superiority

Protocol ▶ IPv6 Header



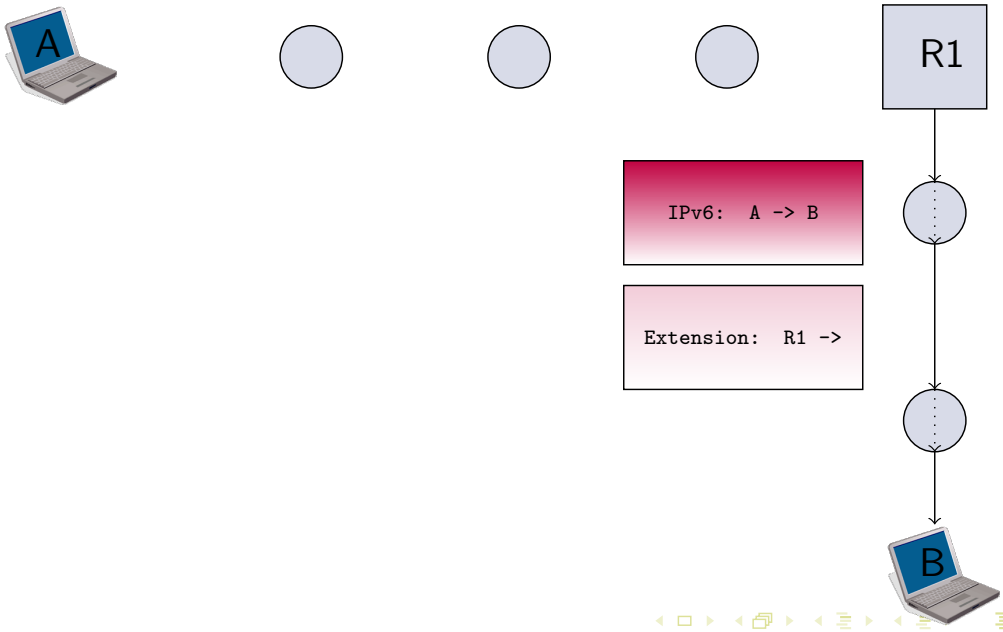
Extension Superiority



R1 is the destination, packet is sent to Routing Extension layer which swaps the addresses and forwards the packet.

Extension Superiority

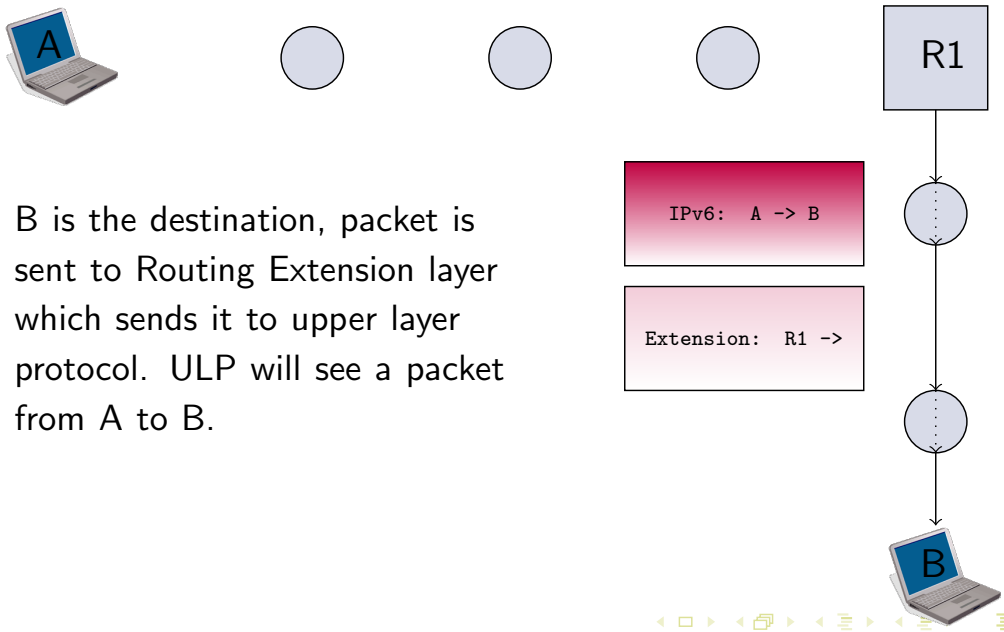
Protocol ▶ IPv6 Header



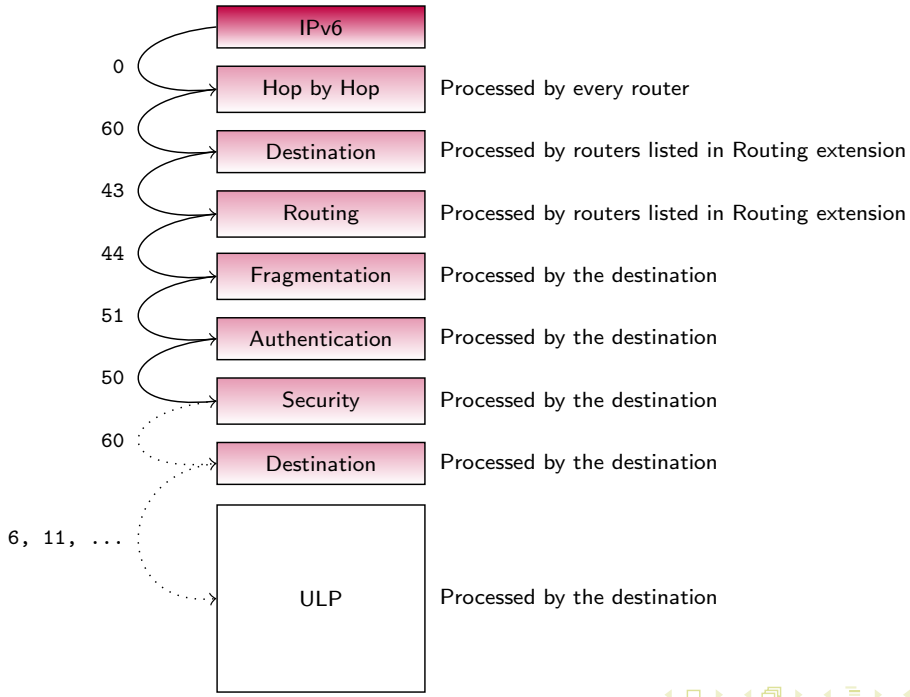
Slide 58 Laurent Toutain Filière 2

Extension Superiority

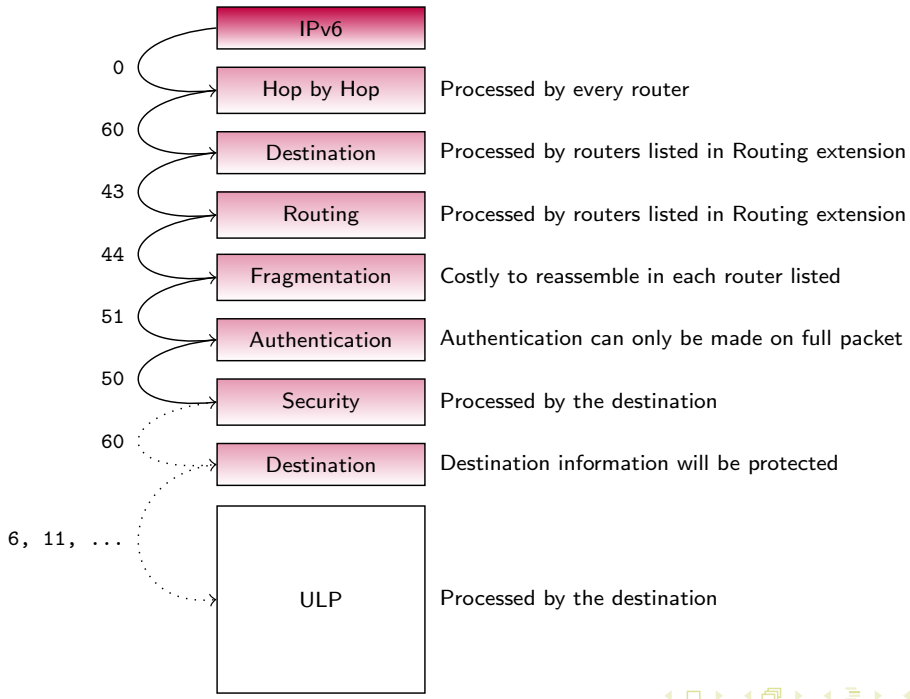
B is the destination, packet is sent to Routing Extension layer which sends it to upper layer protocol. ULP will see a packet from A to B.



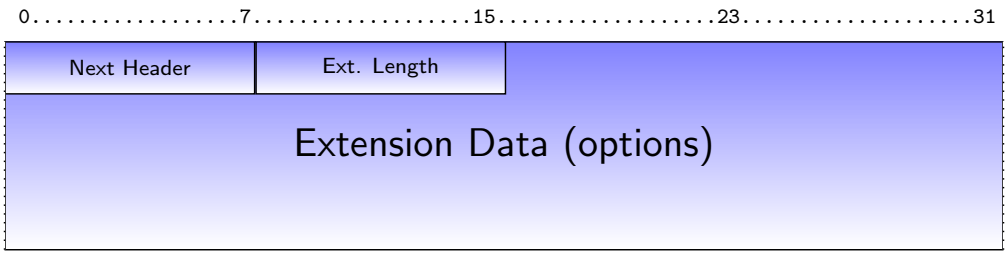
Extension Order is Important



Extension Order is Important



Extensions Generic Format



- Next Header: Save values as in IPv6 packets
- Length: numbers 64-bit long words for variable length extensions (0 for fixed length fragmentation extension)
- Data: options (Hop by hop, Destination) or specific format

Hop by Hop (NH=0)

- Always first position
- Composed of options:

Pad1	0		
Padn	1	lgth.	0 · · · 0
Router Alert	5	2	Value
CALIPSO	7	lgth.	See RFC 5570
Quick Start	38	lgth.	See RFC 4782
Jumbogram	194	4	Datagram Length

Hop by Hop (NH=0)

Protocol ▶ IPv6 Header

- Always first position
- Composed of options:

Pad1	0		
Padn	1	lgth.	0 · · · 0
Router Alert	5	2	Value
CALIPSO	7	lgth.	See RFC 5570
Quick Start	38	lgth.	See RFC 4782
Jumbogram	194	4	Datagram Length

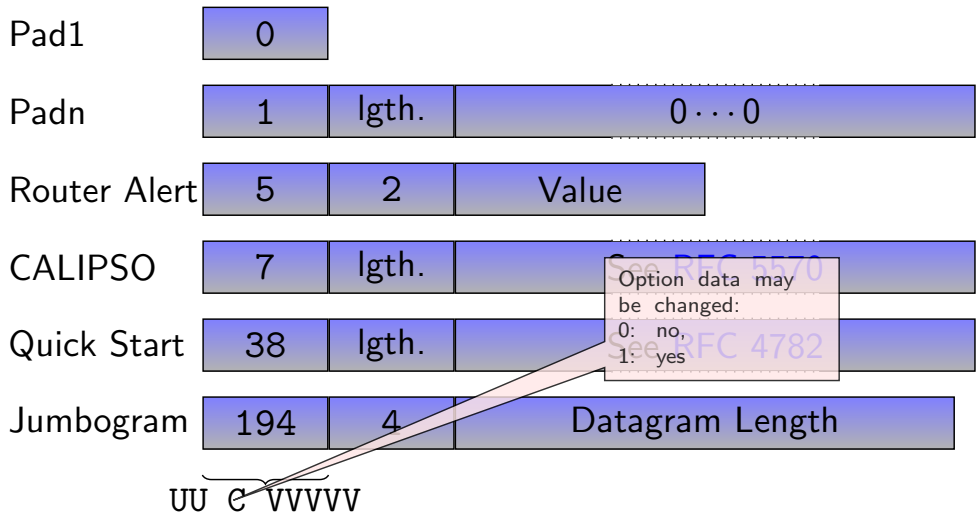
When value unknown:
00: skip,
01: discard,
10: discard + ICMP,
11: Discard + ICMP (if not multicast)

UU C VVVVV

Slide 61 Laurent Toutain Filière 2 TELECOM Bretagne

Hop by Hop (NH=0)

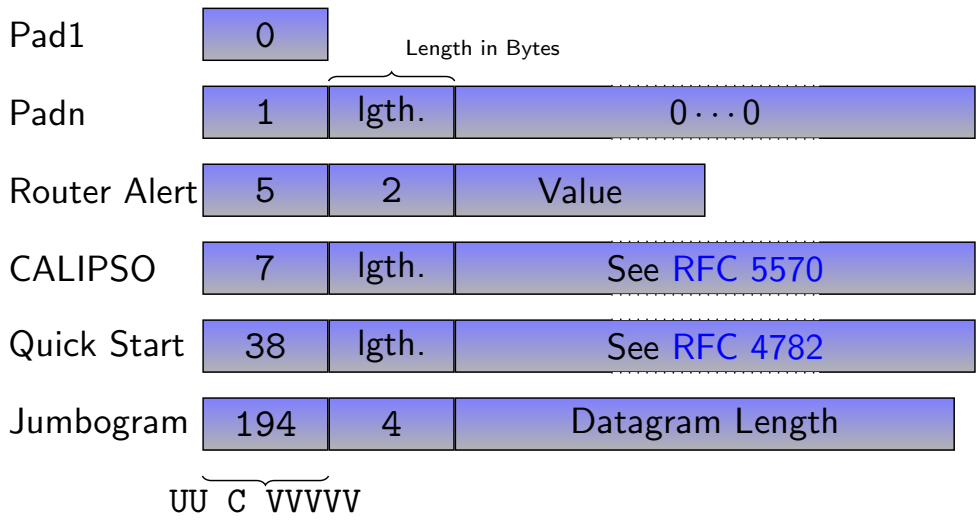
- Always first position
- Composed of options:



Hop by Hop (NH=0)

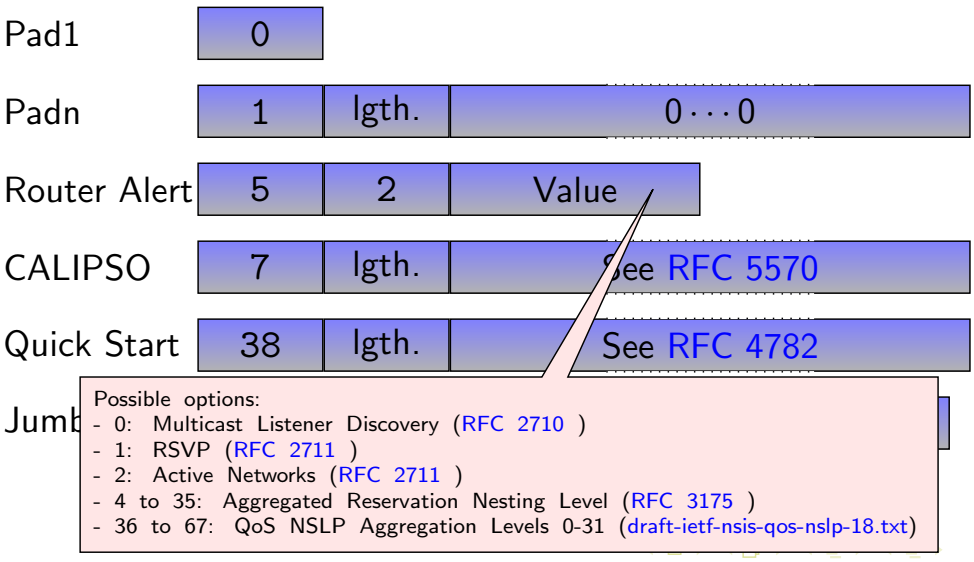
Protocol ▶ IPv6 Header

- Always first position
- Composed of options:

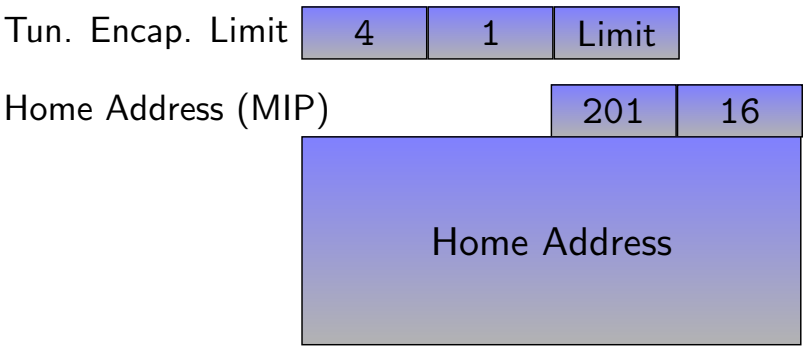


Hop by Hop (NH=0)

- Always first position
- Composed of options:

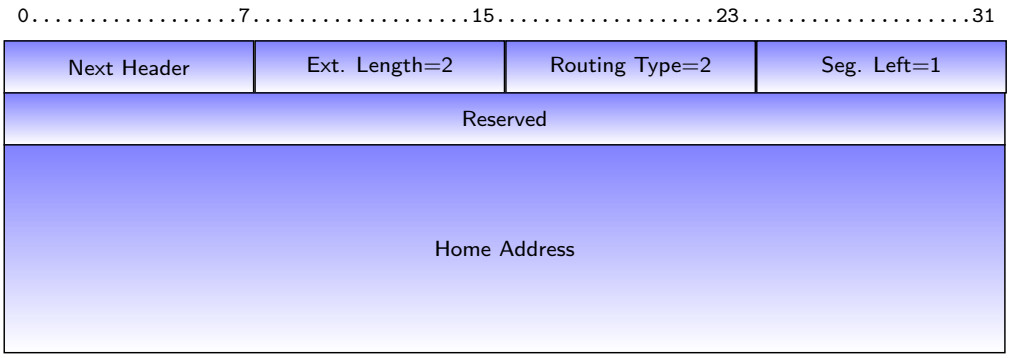


Destination (NH=60)

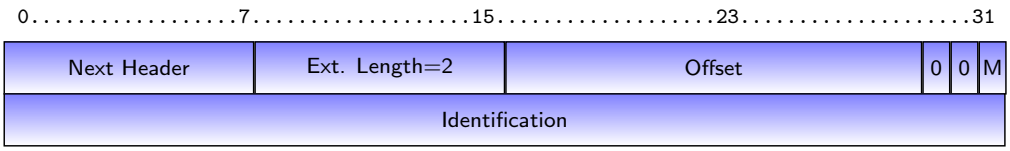


- Tunnel Encapsultation Limit (RFC 2473): the maximum number of nested encapsulations of a packet. When it reaches 0, the packet is discard and an ICMPv6 message is sent.
- Home Address (RFC 3775): Contains the Home Address of the sender (IPv6 header contains the Care-of Address).

Routing (NH=43)



Fragmentation (NH=44)



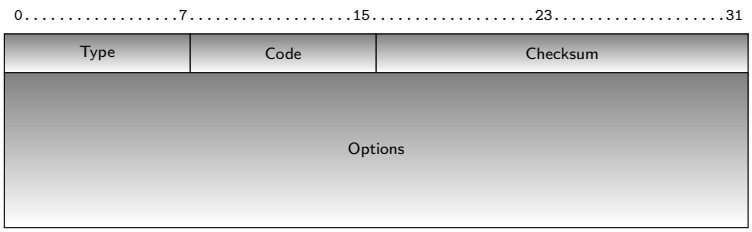
- Compared to IPv4, it is equivalent to DF=1
- A Router never fragments packets but sends an ICMPv6 message ("Packet Too Big") with the expected size
- The Sender either uses the fragmentation extension or adapts TCP segments

ICMPv6

ICMPv6

- ICMPv6 is different from ICMP for IPv4 (RFC 4443)
  - IPv6 (or extension): 58
- Features are extended and better organized
- Never filter ICMPv6 messages blindly, be careful to what you do (see RFC 4890)

Format :



Precision

type code nature of the message ICMPv6

code specifies the cause of the message ICMPv6

mandatory *checksum* used to verify the integrity of ICMP packet

ICMPv6 : Two Functions

- Error occurs during forwarding ($value < 128$)

1	Destination Unreachable
2	Packet Too Big
3	Time Exceeded
4	Parameter Problem

- Management Applications ($value > 128$)

128	Echo Request
129	Echo Reply
130	Group Membership Query
131	Group Membership Report
132	Group Membership Reduction
133	Router Solicitation
134	Router Advertissement
135	Neighbor Solicitation
136	Neighbor Advertissement
137	Redirect

IPv6 Neighbor Discovery

Neighbor Discovery

Neighbor Discovery (RFC 4861)

Associated Protocols & Mechanisms

- IPv6 nodes sharing the same physical medium (link) use Neighbor Discovery (ND) to:
  - determine link-layer addresses of their neighbors
    - IPv4 : ARP
  - Address auto-configuration
    - Layer 3 parameters: IPv6 address, default route, MTU and Hop Limit
    - Only for hosts !
    - IPv4 : impossible, mandate a centralized DHCP server
  - Duplicate Address Detection (DAD)
    - IPv4 : gratuitous ARP
  - maintain neighbors reachability information (NUD)
- Mainly uses multicast addresses but also takes into account NBMA Networks (eg., ATM)
- Protocol packets are transported/encapsulated by/in ICMPv6 messages:
  - Router Solicitation: 133 ; Router Advertisement: 134 ; Neighbor Solicitation: 135 ; Neighbor Advertisement: 136 ; Redirect: 137

Neighbor Discovery (RFC 4861)

- IPv6 nodes sharing the same physical medium (link) use Neighbor Discovery (ND) to:
  - determine link-layer addresses of their neighbors
    - IPv4 : ARP
  - Address auto-configuration
    - Layer 3 parameters: IPv6 address, default route, MTU and Hop Limit
    - Only for hosts !
    - IPv4 : impossible, mandate a centralized DHCP server
  - Duplicate Address Detection (DAD)
    - IPv4 : gratuitous ARP
  - maintain neighbors reachability information (NUD)
- Mainly uses multicast addresses but also takes into account NBMA Networks (eg., ATM)
- Protocol packets are transported/encapsulated by/in ICMPv6 messages:
  - Router Solicitation: 133 ; Router Advertisement: 134 ; Neighbor Solicitation: 135 ; Neighbor Advertisement: 136 ; Redirect: 137

Neighbor Discovery (RFC 4861)

Associated Protocols & Mechanisms

- IPv6 nodes sharing the same physical medium (link) use Neighbor Discovery (ND) to:
  - determine link-layer addresses of their neighbors
    - IPv4 : ARP
  - Address auto-configuration
    - Layer 3 parameters: IPv6 address, default route, MTU and Hop Limit
    - Only for hosts !
    - IPv4 : impossible, mandate a centralized DHCP server
  - Duplicate Address Detection (DAD)
    - IPv4 : gratuitous ARP
  - maintain neighbors reachability information (NUD)
- Mainly uses multicast addresses but also takes into account NBMA Networks (eg., ATM)
- Protocol packets are transported/encapsulated by/in ICMPv6 messages:
  - Router Solicitation: 133 ; Router Advertisement: 134 ; Neighbor Solicitation: 135 ; Neighbor Advertisement: 136 ; Redirect: 137

Slide 70 Laurent Toutain Filière 2 TELECOM Bretagne

Neighbor Discovery (RFC 4861)

- IPv6 nodes sharing the same physical medium (link) use Neighbor Discovery (ND) to:
  - determine link-layer addresses of their neighbors
    - IPv4 : ARP
  - Address auto-configuration
    - Layer 3 parameters: IPv6 address, default route, MTU and Hop Limit
    - Only for hosts !
    - IPv4 : impossible, mandate a centralized DHCP server
  - Duplicate Address Detection (DAD)
    - IPv4 : gratuitous ARP
  - maintain neighbors reachability information (NUD)
- Mainly uses multicast addresses but also takes into account NBMA Networks (eg., ATM)
- Protocol packets are transported/encapsulated by/in ICMPv6 messages:
  - Router Solicitation: 133 ; Router Advertisement: 134 ; Neighbor Solicitation: 135 ; Neighbor Advertisement: 136 ; Redirect: 137

Neighbor Discovery (RFC 4861)

Associated Protocols & Mechanisms

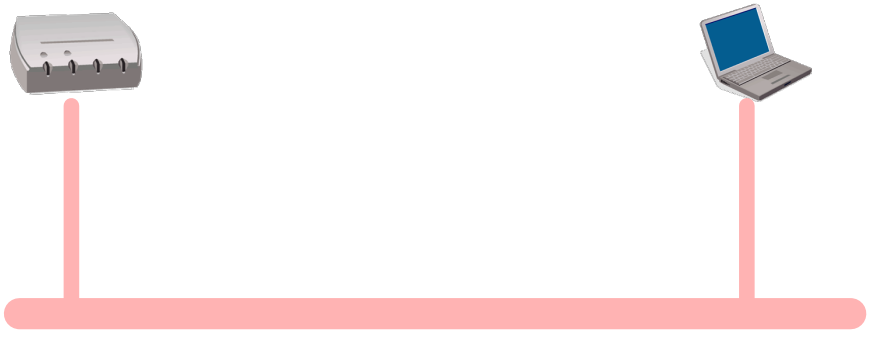
- IPv6 nodes sharing the same physical medium (link) use Neighbor Discovery (ND) to:
  - determine link-layer addresses of their neighbors
    - IPv4 : ARP
  - Address auto-configuration
    - Layer 3 parameters: IPv6 address, default route, MTU and Hop Limit
    - Only for hosts !
    - IPv4 : impossible, mandate a centralized DHCP server
  - Duplicate Address Detection (DAD)
    - IPv4 : gratuitous ARP
  - maintain neighbors reachability information (NUD)
- Mainly uses multicast addresses but also takes into account NBMA Networks (eg., ATM)
- Protocol packets are transported/encapsulated by/in ICMPv6 messages:
  - Router Solicitation: 133 ; Router Advertisement: 134 ; Neighbor Solicitation: 135 ; Neighbor Advertisement: 136 ; Redirect: 137

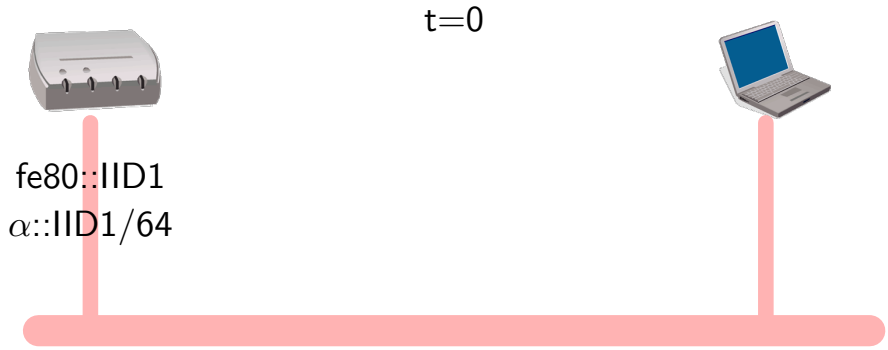
Slide 70 Laurent Toutain Filière 2 TELECOM Bretagne

Neighbor Discovery (RFC 4861)

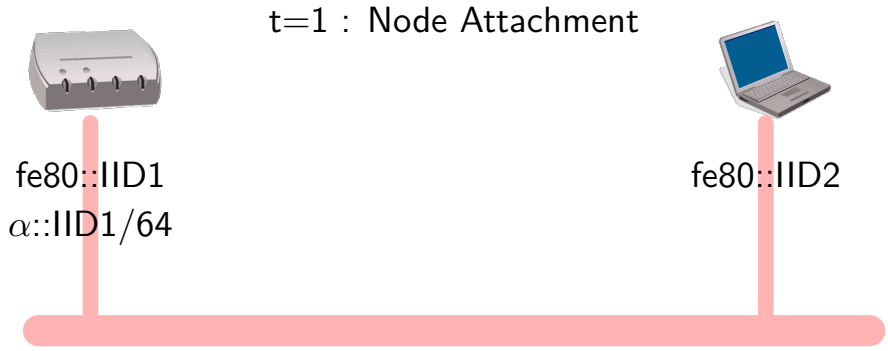
- IPv6 nodes sharing the same physical medium (link) use Neighbor Discovery (ND) to:
  - determine link-layer addresses of their neighbors
    - IPv4 : ARP
  - Address auto-configuration
    - Layer 3 parameters: IPv6 address, default route, MTU and Hop Limit
    - Only for hosts !
    - IPv4 : impossible, mandate a centralized DHCP server
  - Duplicate Address Detection (DAD)
    - IPv4 : gratuitous ARP
  - maintain neighbors reachability information (NUD)
- Mainly uses multicast addresses but also takes into account NBMA Networks (eg., ATM)
- Protocol packets are transported/encapsulated by/in ICMPv6 messages:
  - Router Solicitation: 133 ; Router Advertisement: 134 ; Neighbor Solicitation: 135 ; Neighbor Advertisement: 136 ; Redirect: 137

Stateless Auto-configuration: Basic Principles

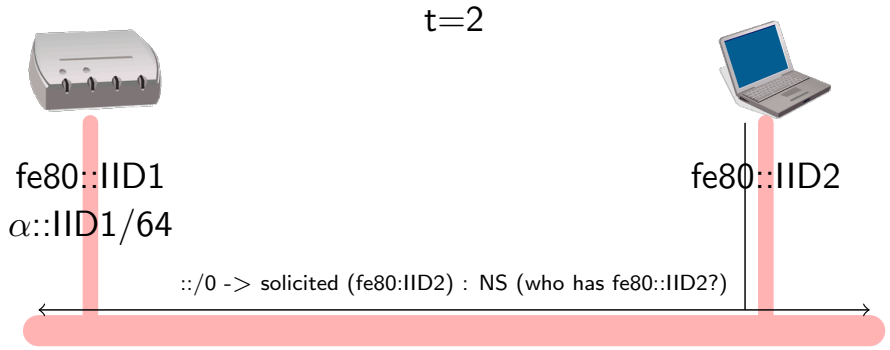




Time t=0: Router is configured with a link-local address and manually configured with a global address (α::/64 is given by the network administrator)

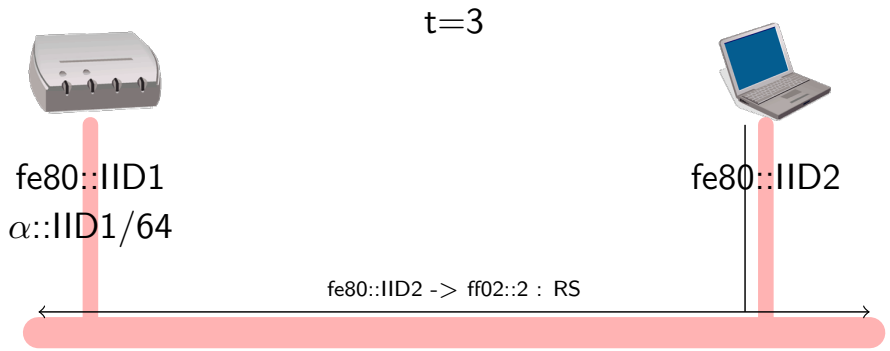


Host constructs its link-local address based on the interface MAC address

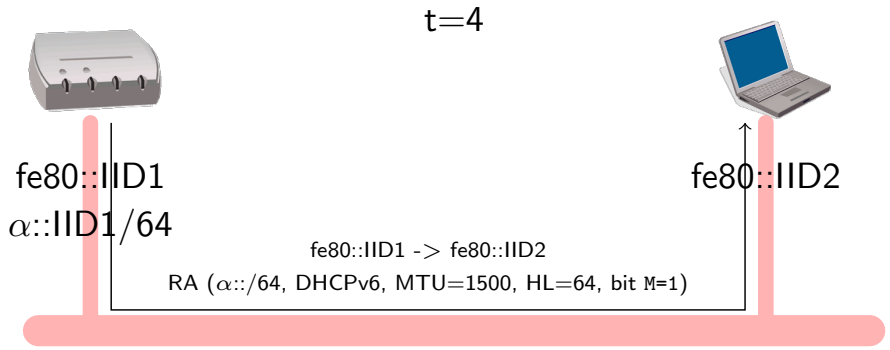


Host does a DAD (i.e. sends a Neighbor Solicitation to query resolution of its own address (tentative): no answers means no other host has this value).

Stateless Auto-configuration: Basic Principles
Associated Protocols & Mechanisms

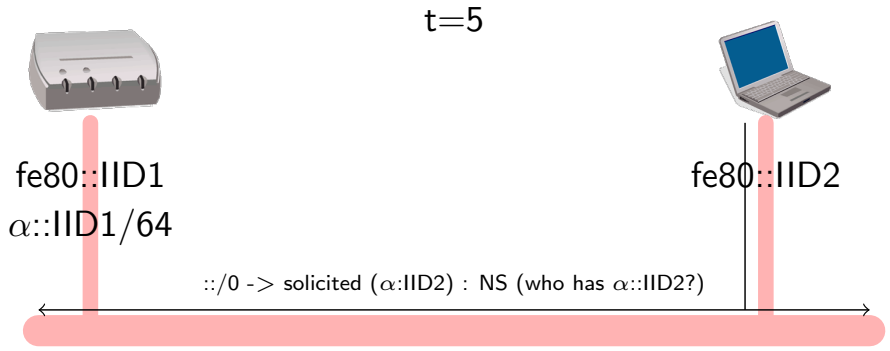


Host sends a Router Solicitation to the Link-Local All-Routers Multicast group using the newly link-local configured address

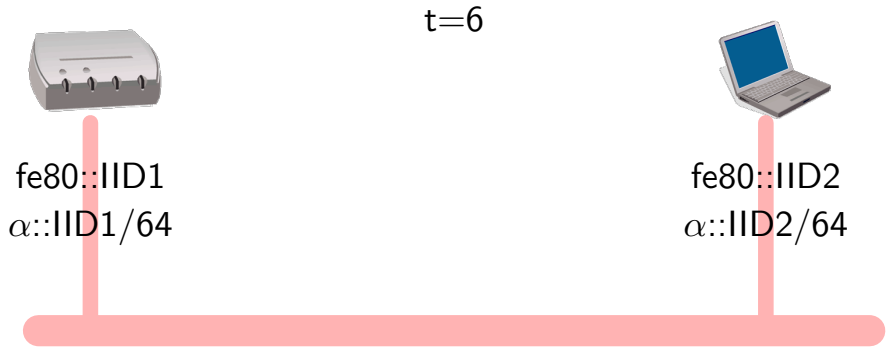


Router directly answers the host using Link-local addresses. The answer may contain a/several prefix(es). Router can also mandate hosts to use DHCPv6 to obtain prefixes (statefull auto-configuration) and/or other parameters (DNS servers...): Bit $\texttt{M} = 1$.

Stateless Auto-configuration: Basic Principles
Associated Protocols & Mechanisms



Host does a DAD (i.e. sends a Neighbor Solicitation to query resolution of its own global address: no answers means no other host as this value).



Host sets the global address and takes answering router as the default router.

- DAD is a long process:
  - Send NS
  - Timeout
  - May be repeated
- For Link-Local and Global addresses
- Mobile nodes are penalized
  - Discover Network
  - Authentication
  - DAD, RS/RA, DAD
- oDAD allows a host to use the address before DAD
- If no answer to DAD then the address becomes a valid one

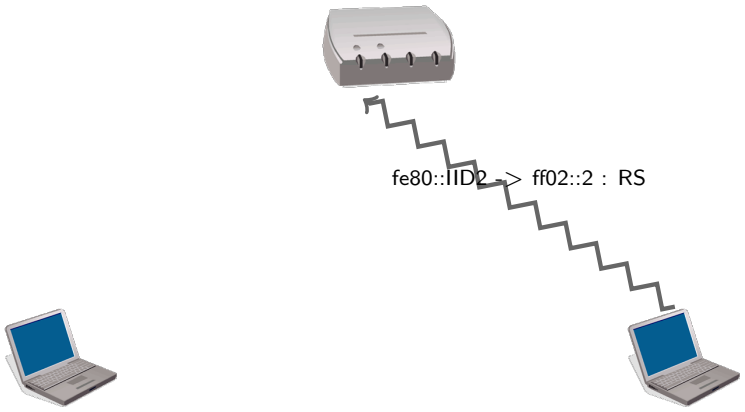
Non-Broadcast Multiple Access (NBMA) Networks

NBMA Networks

Associated Protocols & Mechanisms

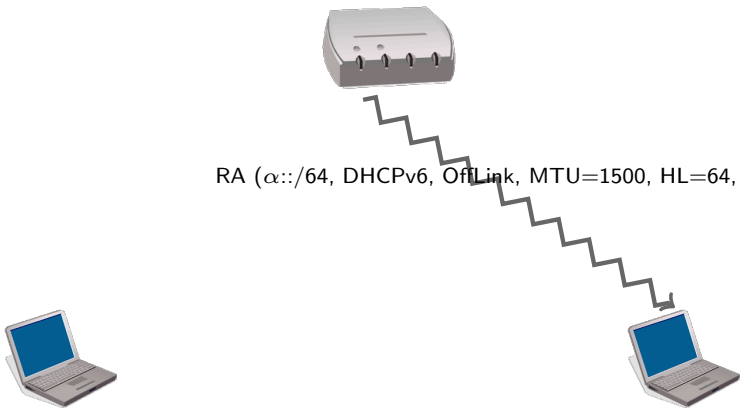
- NDP can handle efficiently NBMA networks
  - Every host can be joined separately, but no broadcast
  - Telephony network, ATM...
- Off-link bit is RA by the router to inform of a NBMA network
  - 3G, Sensor Networks (broadcast expensive)
- All packets are sent to to the router, which will forward to destination
  - No NS
  - ICMP Redirect can be used.

Off Link example



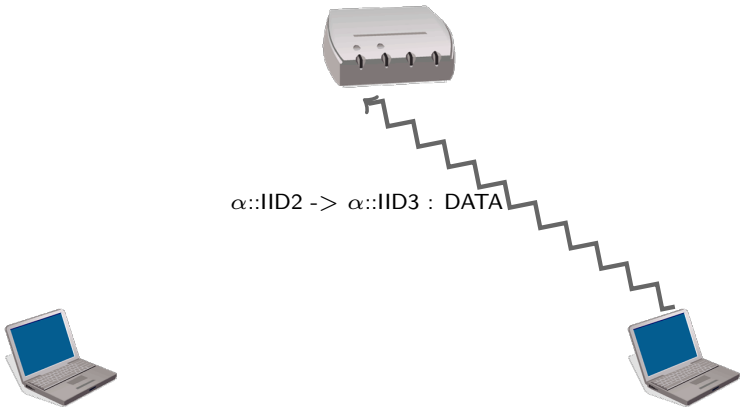
Off Link example

Associated Protocols & Mechanisms



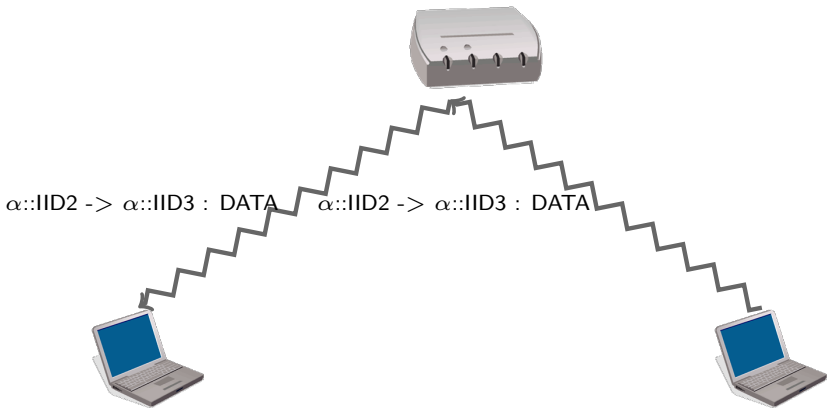
Off Link example

Associated Protocols & Mechanisms

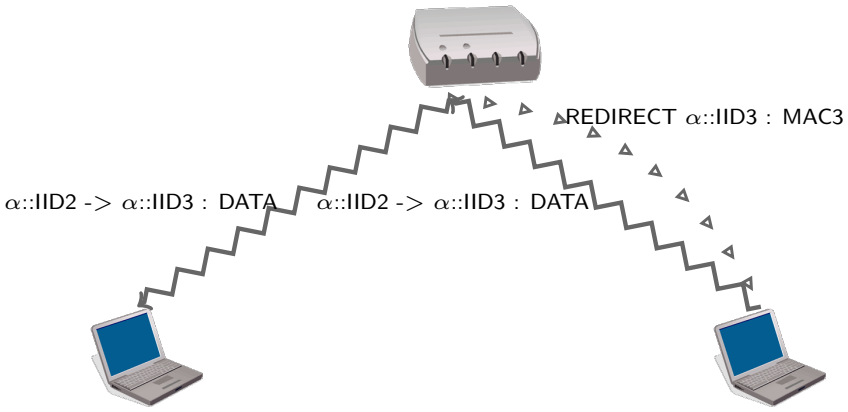


Off Link example

Associated Protocols & Mechanisms

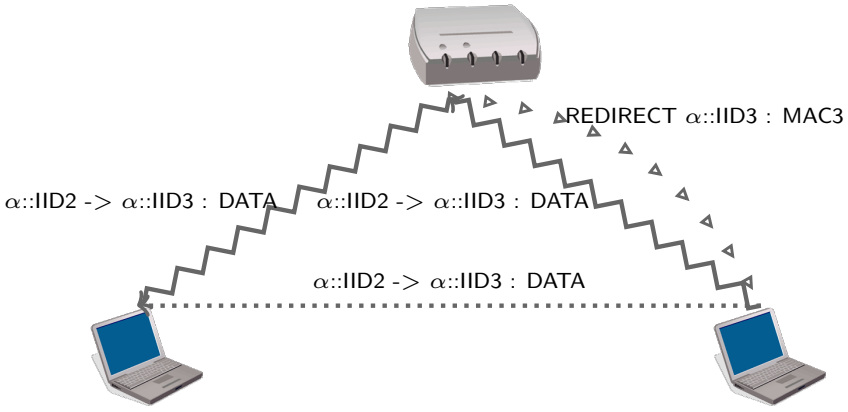


Off Link example Optional



Off Link example Optional

Associated Protocols & Mechanisms



Examples

Router Configuration Example


```
interface Vlan5
 description reseau C5
 ip address 192.108.119.190 255.255.255.128
...
 ipv6 address 2001:660:7301:1::/64 eui-64
 ipv6 enable
 ipv6 nd ra-interval 10
 ipv6 nd prefix-advertisement 2001:660:7301:1::/64 2592000\
 604800 onlink autoconfig
```

Router Configuration Example


```
interface Vlan5
 description reseau C5
 ip address 192.108.119.190 255.255.255.128
...
 ipv6 address 2001:660:7301:1::/64 eui-64
 ipv6 enable
 ipv6 nd ra-interval 10
 ipv6 nd prefix-advertisement 2001:660:7301:1::/64 2592000\
 604800 onlink autoconfig
```


Router Configuration Example

Associated Protocols & Mechanisms

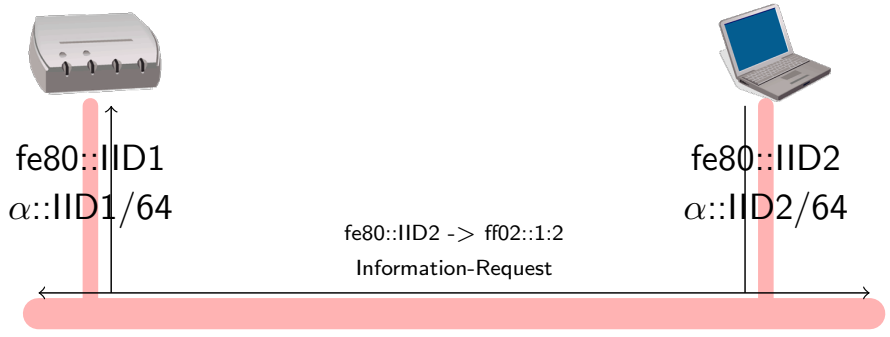
```
interface Vlan5
 description reseau C5
 ip address 192.108.119.190 255.255.255.128
...
 ipv6 address 2001:660:7301:1::/64 eui-64
 ipv6 enable
 ipv6 nd ra-interval 10
 ipv6 nd prefix-advertisement 2001:660:7301:1::/64 2592000\
 604800 onlink autoconfig
```

Slide 77 Laurent Toutain Filière 2 TELECOM Bretagne

Router Configuration Example

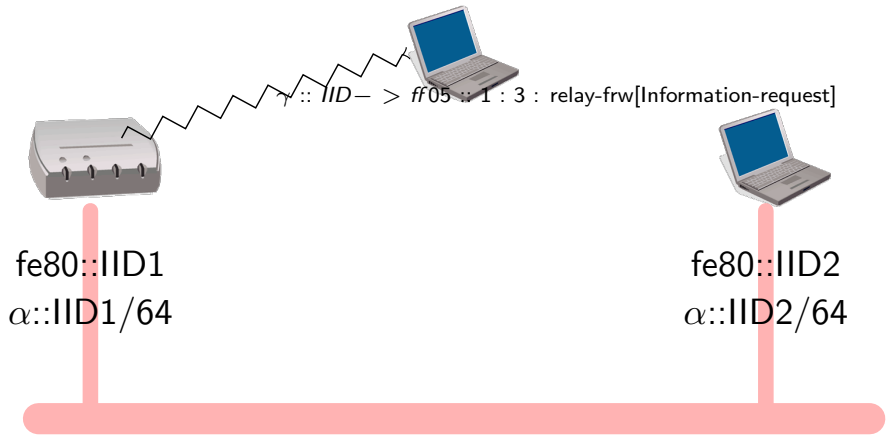

```
interface Vlan5
 description reseau C5
 ip address 192.108.119.190 255.255.255.128
...
 ipv6 address 2001:660:7301:1::/64 eui-64
 ipv6 enable
 ipv6 nd ra-interval 10
 ipv6 nd prefix-advertisement 2001:660:7301:1::/64 2592000\
 604800 onlink autoconfig
```


Stateless DHCPv6 (RFC 3736): With static parameters



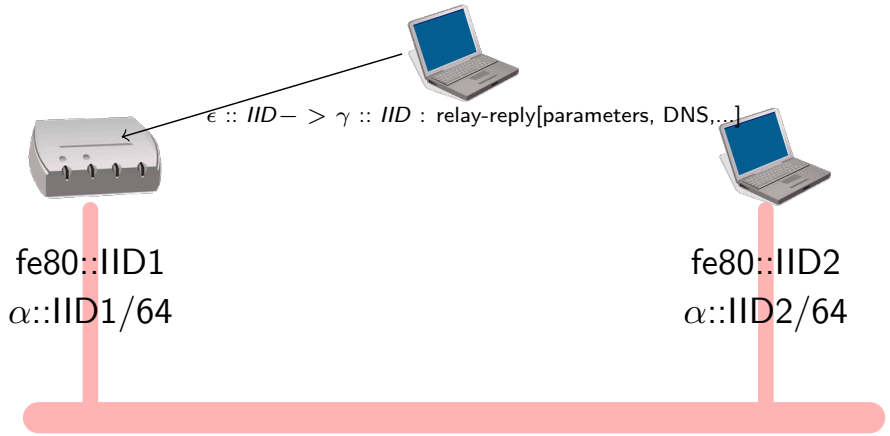
Host needs only static parameters (DNS, NTP,...). It sends an Information-Request message to All_DHCP_Agents multicast group. The scope of this address is link-local.

Stateless DHCPv6 (RFC 3736): With static parameters



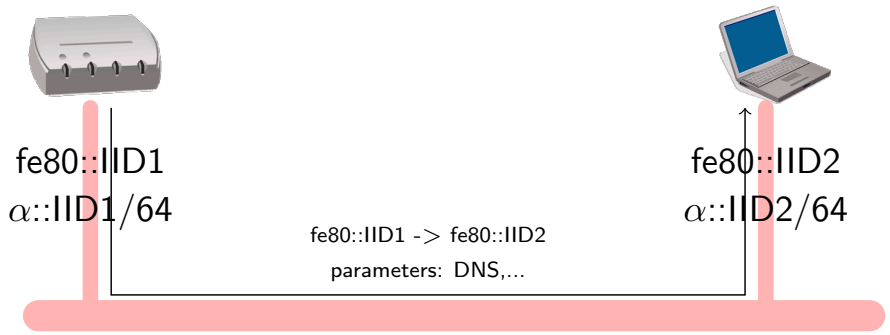
A relay (generally the router) encapsulates the request into a *Forward message and sends it either to the All_DHCP_Servers site-local multicast group or to a list of pre-defined unicast addresses.*

Stateless DHCPv6 (RFC 3736): With static parameters



The server responds to the relay

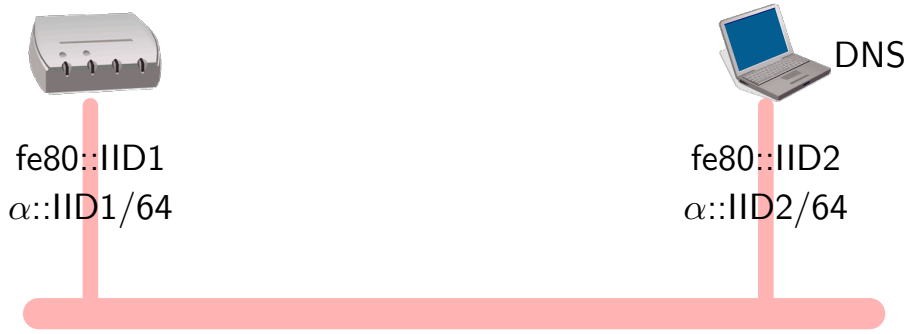
Stateless DHCPv6 (RFC 3736): With static parameters



The router extracts information from the message to create answer and sends information to the host

Stateless DHCPv6 (RFC 3736): With static parameters

Associated Protocols & Mechanisms

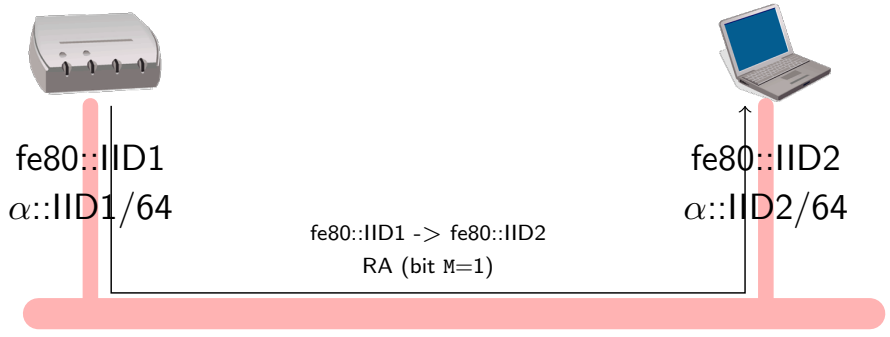


Host is now configured to resolve domain names through the DNS

DHCPv6

DHCPv6 : Stateful Auto-Configuration

Associated Protocols & Mechanisms

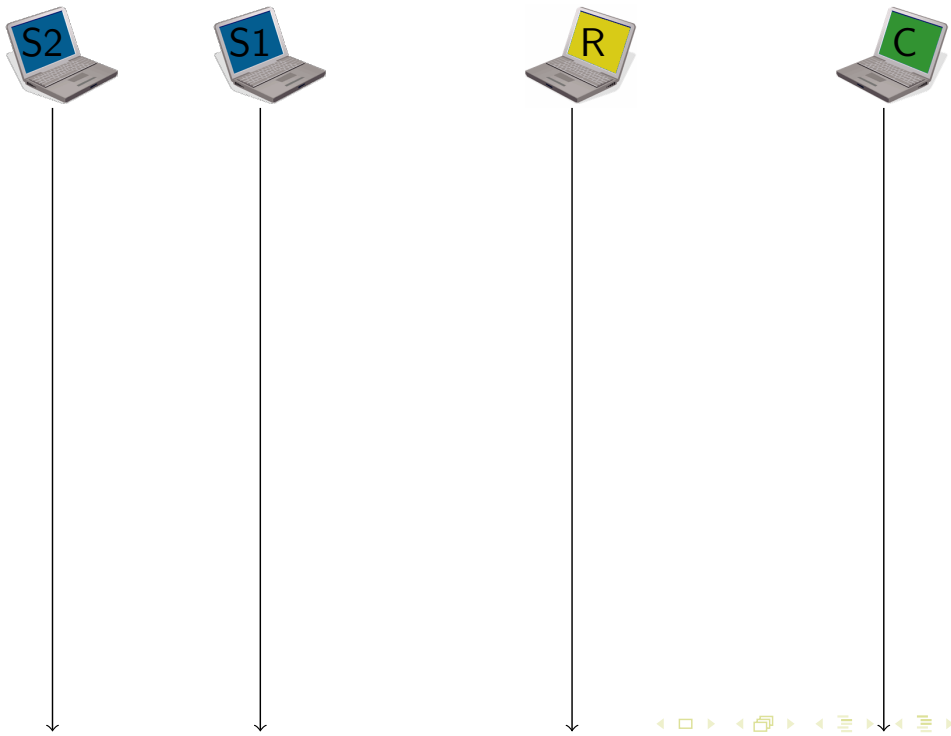


Router responds to RS with a RA message with bit `M` set to 1. Host should request its IPv6 address from a DHCPv6 server.

DHCPv6 Full Features

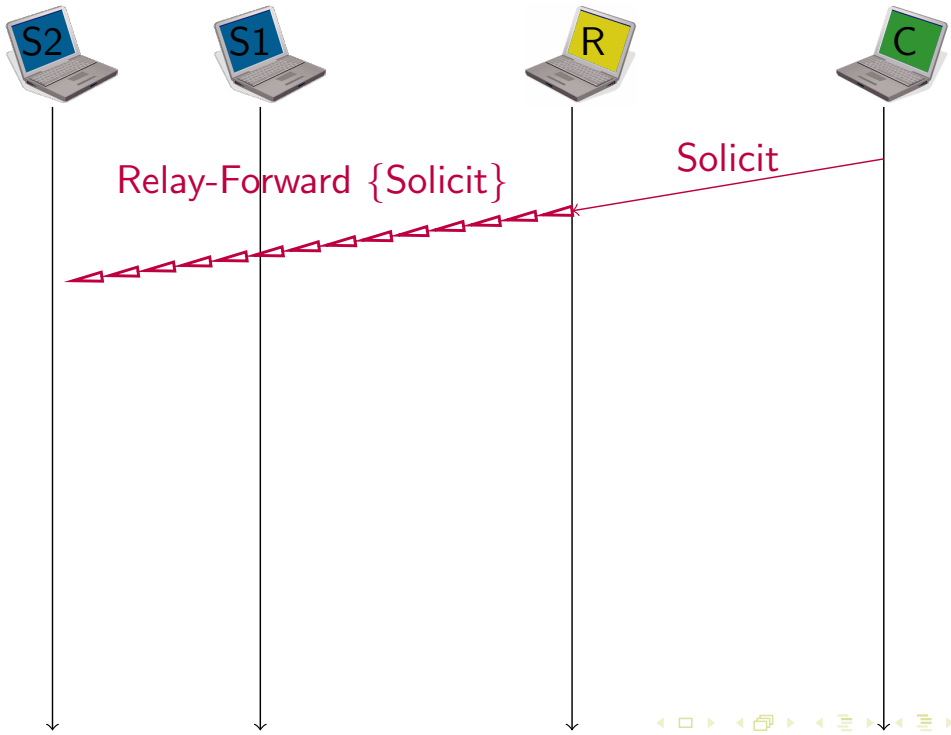
- For address or prefix allocation information form only one DHCPv6 must be taken into account. Four message exchange :
  - Solicit : send by clients to locate servers
  - Advertise : send by servers to indicate services available
  - Request : send by client to a specific server (could be through relays)
  - Reply : send by server with parameters requested
- Addresses or Prefixes are allocated for certain period of time
  - Renew : Send by the client tells the server to extend lifetime
  - Rebind : If no answer from renew, the client use rebind to extend lifetime of addresses and update other configuration parameters
  - Reconfigure : Server informs availability of new or update information. Clients can send renew or Information-request
  - Release : Send by the client tells the server the client does not need any longer addresses or prefixes.
  - Decline : to inform server that allocated addresses are already in use on the link

DHCPv6 Scenarii



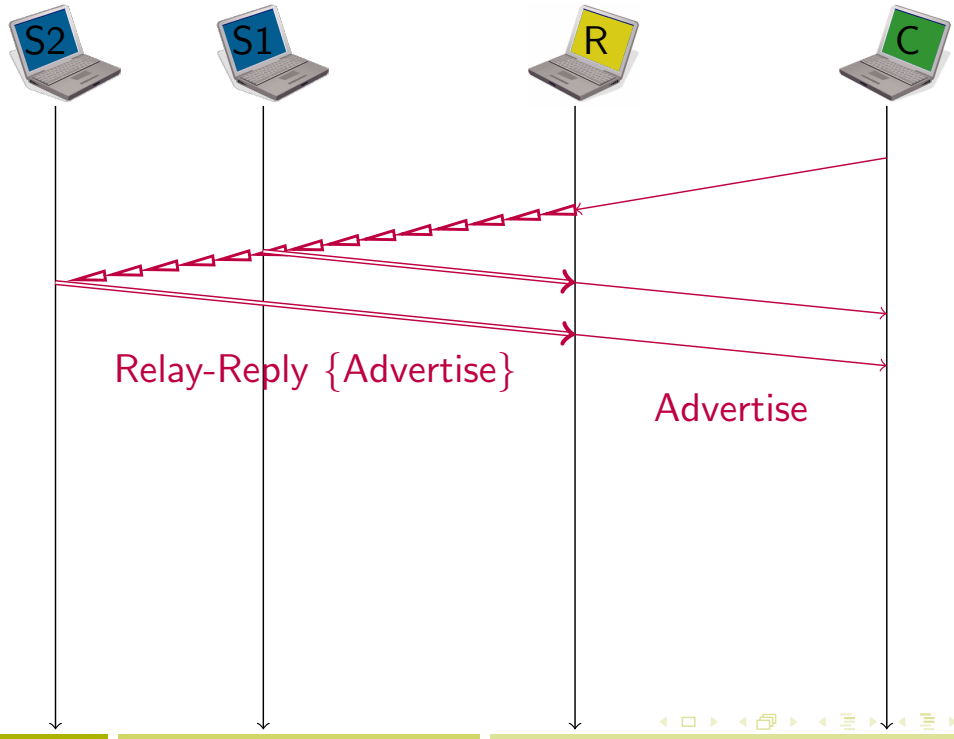
DHCPv6 Scenarii

Associated Protocols & Mechanisms



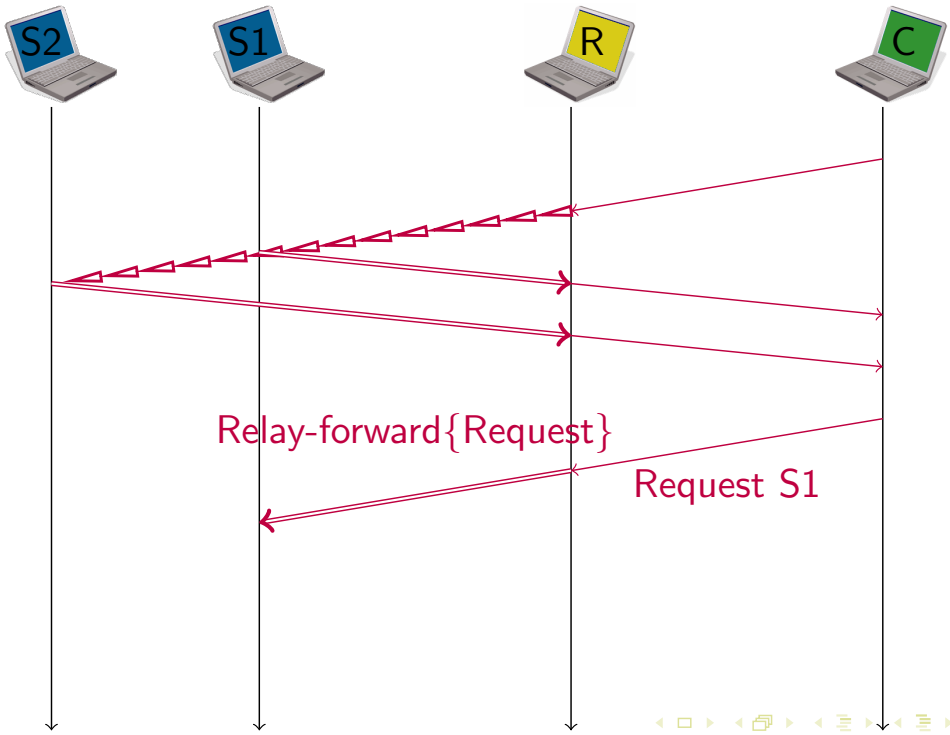
DHCPv6 Scenarii

Associated Protocols & Mechanisms



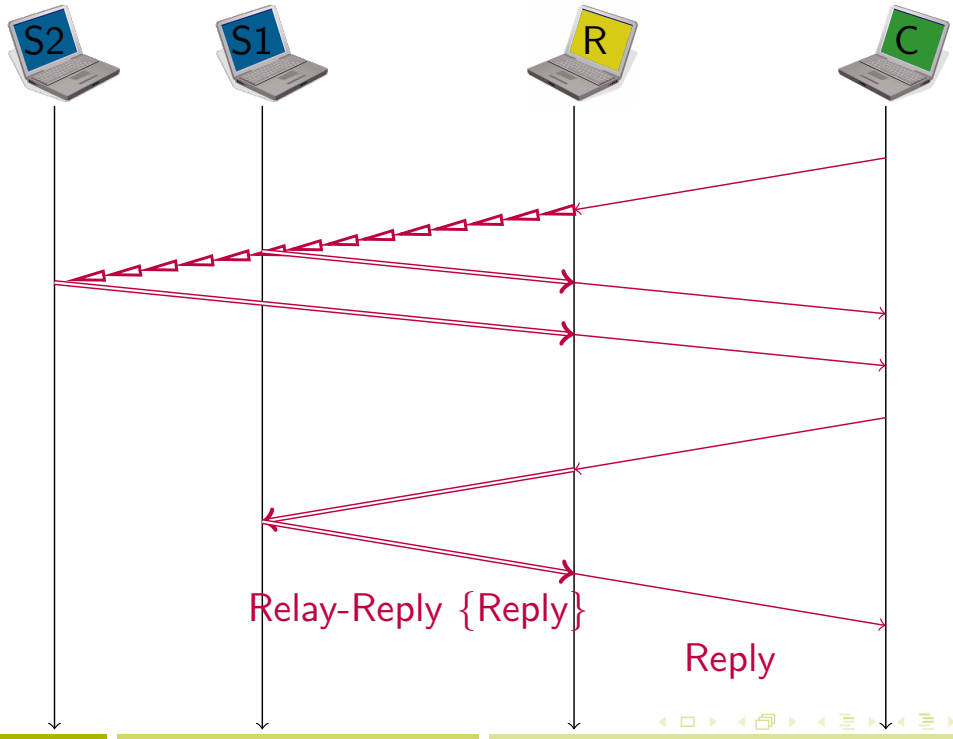
DHCPv6 Scenarii

Associated Protocols & Mechanisms



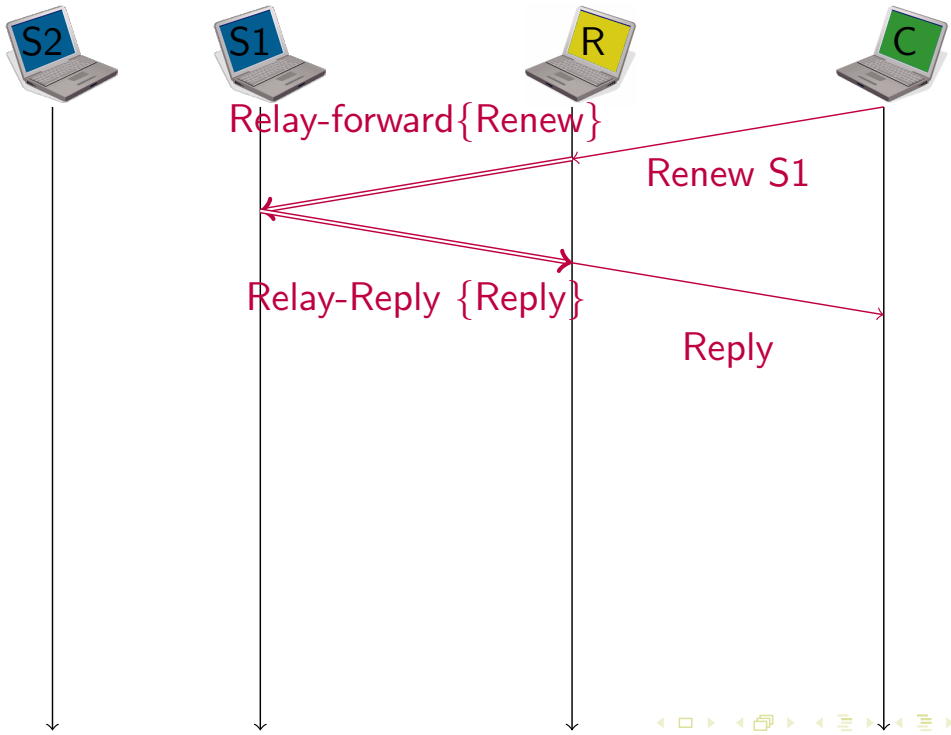
DHCPv6 Scenarii

Associated Protocols & Mechanisms



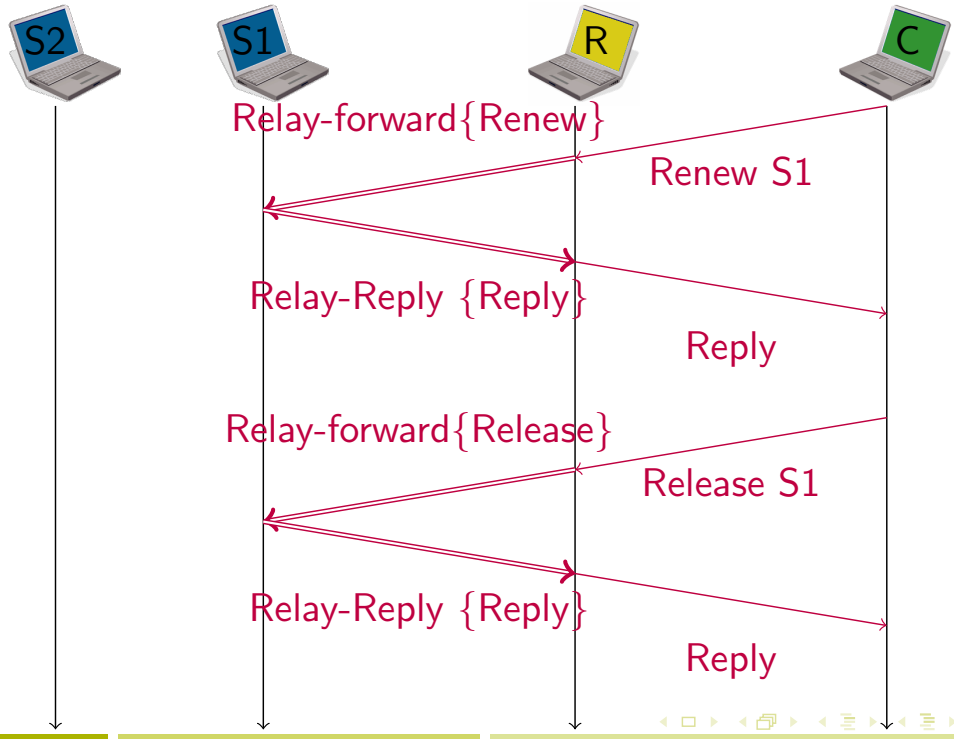
DHCPv6 Scenarii

Associated Protocols & Mechanisms



DHCPv6 Scenarii

Associated Protocols & Mechanisms



Wireless Sensor Network

IETF Working Groups

- Allow end-to-end communication.
  - Not the main feature.
- L2 Independant
- Reduce interconnection costs
- IPv6, ... but:
  - IPv6 packet are too big
    - Compress them
  - IPv6 is not energy aware
    - Reduce packet size and control plane traffic
  - IPv6 link is not well defined
    - Neighbor Discovery Protocol must be improved
  - IPv6 routing protocols (even Manet) are too expensive
    - Define a new routing protocol for LoWPAN
  - IPv6 End-to-end is insecure
    - Define standard ALG to relay messages, based on REST

- Allow end-to-end communication.
  - Not the main feature.
- L2 Independant
- Reduce interconnection costs
- IPv6, ... but:
  - IPv6 packet are too big
    - Compress them
  - IPv6 is not energy aware
    - Reduce packet size and control plane traffic
  - IPv6 link is not well defined
    - Neighbor Discovery Protocol must be improved
  - IPv6 routing protocols (even Manet) are too expensive
    - Define a new routing protocol for LoWPAN
  - IPv6 End-to-end is insecure
    - Define standard ALG to relay messages, based on REST

Wireless Sensor Network
IETF Working Groups

- Allow end-to-end communication.
  - Not the main feature.
- L2 Independant
- Reduce interconnection costs
- IPv6, ... but:
  - IPv6 packet are too big
    - Compress them
  - IPv6 is not energy aware
    - Reduce packet size and control plane traffic
  - IPv6 link is not well defined
    - Neighbor Discovery Protocol must be improved
  - IPv6 routing protocols (even Manet) are too expensive
    - Define a new routing protocol for LoWPAN
  - IPv6 End-to-end is insecure
    - Define standard ALG to relay messages, based on REST

Slide 84 Laurent Toutain Filière 2 TELECOM Bretagne

- Allow end-to-end communication.
  - Not the main feature.
- L2 Independant
- Reduce interconnection costs
- IPv6, ... but:
  - IPv6 packet are too big
    - Compress them
  - IPv6 is not energy aware
    - Reduce packet size and control plane traffic
  - IPv6 link is not well defined
    - Neighbor Discovery Protocol must be improved
  - IPv6 routing protocols (even Manet) are too expensive
    - Define a new routing protocol for LoWPAN
  - IPv6 End-to-end is insecure
    - Define standard ALG to relay messages, based on REST

Wireless Sensor Network
IETF Working Groups

- Allow end-to-end communication.
  - Not the main feature.
- L2 Independant
- Reduce interconnection costs
- IPv6, ... but:
  - IPv6 packet are too big
    - Compress them
  - IPv6 is not energy aware
    - Reduce packet size and control plane traffic
  - IPv6 link is not well defined
    - Neighbor Discovery Protocol must be improved
  - IPv6 routing protocols (even Manet) are too expensive
    - Define a new routing protocol for LoWPAN
  - IPv6 End-to-end is insecure
    - Define standard ALG to relay messages, based on REST

Slide 84 Laurent Toutain Filière 2
TELECOM Bretagne

- Allow end-to-end communication.
  - Not the main feature.
- L2 Independant
- Reduce interconnection costs
- IPv6, ... but:
  - IPv6 packet are too big
    - Compress them
  - IPv6 is not energy aware
    - Reduce packet size and control plane traffic
  - IPv6 link is not well defined
    - Neighbor Discovery Protocol must be improved
  - IPv6 routing protocols (even Manet) are too expensive
    - Define a new routing protocol for LoWPAN
  - IPv6 End-to-end is insecure
    - Define standard ALG to relay messages, based on REST

Wireless Sensor Network
IETF Working Groups

- Allow end-to-end communication.
  - Not the main feature.
- L2 Independant
- Reduce interconnection costs
- IPv6, ... but:
  - IPv6 packet are too big
    - Compress them
  - IPv6 is not energy aware
    - Reduce packet size and control plane traffic
  - IPv6 link is not well defined
    - Neighbor Discovery Protocol must be improved
  - IPv6 routing protocols (even Manet) are too expensive
    - Define a new routing protocol for LoWPAN
  - IPv6 End-to-end is insecure
    - Define standard ALG to relay messages, based on REST

Slide 84 Laurent Toutain Filière 2
TELECOM Bretagne

- Allow end-to-end communication.
  - Not the main feature.
- L2 Independant
- Reduce interconnection costs
- IPv6, ... but:
  - IPv6 packet are too big
    - Compress them
  - IPv6 is not energy aware
    - Reduce packet size and control plane traffic
  - IPv6 link is not well defined
    - Neighbor Discovery Protocol must be improved
  - IPv6 routing protocols (even Manet) are too expensive
    - Define a new routing protocol for LoWPAN
  - IPv6 End-to-end is insecure
    - Define standard ALG to relay messages, based on REST

Wireless Sensor Network
IETF Working Groups

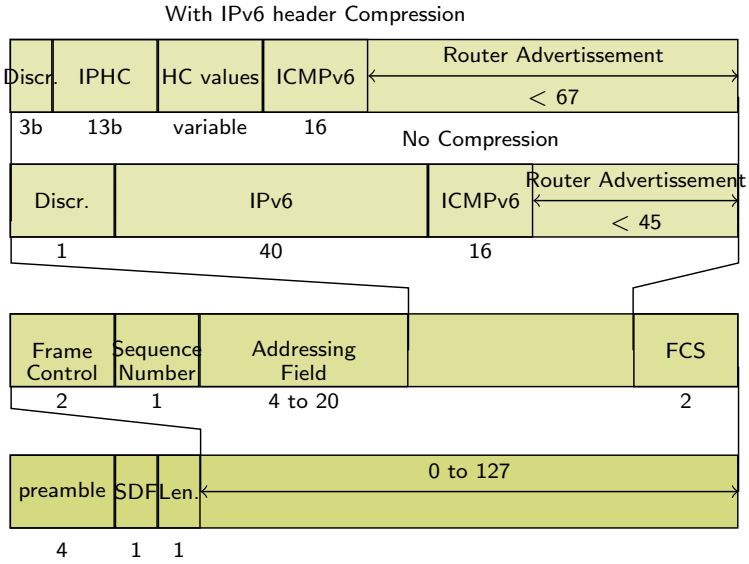
- Allow end-to-end communication.
  - Not the main feature.
- L2 Independant
- Reduce interconnection costs
- IPv6, ... but:
  - IPv6 packet are too big
    - Compress them
  - IPv6 is not energy aware
    - Reduce packet size and control plane traffic
  - IPv6 link is not well defined
    - Neighbor Discovery Protocol must be improved
  - IPv6 routing protocols (even Manet) are too expensive
    - Define a new routing protocol for LoWPAN
  - IPv6 End-to-end is insecure
    - Define standard ALG to relay messages, based on REST

Wireless Sensor Network

- Allow end-to-end communication.
  - Not the main feature.
- L2 Independant
- Reduce interconnection costs
- IPv6, ... but:
  - IPv6 packet are too big
    - Compress them
  - IPv6 is not energy aware
    - Reduce packet size and control plane traffic
  - IPv6 link is not well defined
    - Neighbor Discovery Protocol must be improved
  - IPv6 routing protocols (even Manet) are too expensive
    - Define a new routing protocol for LoWPAN
  - IPv6 End-to-end is insecure
    - Define standard ALG to relay messages, based on REST

IPv6 too big

- and L2 frames are too small.



Discriminator values

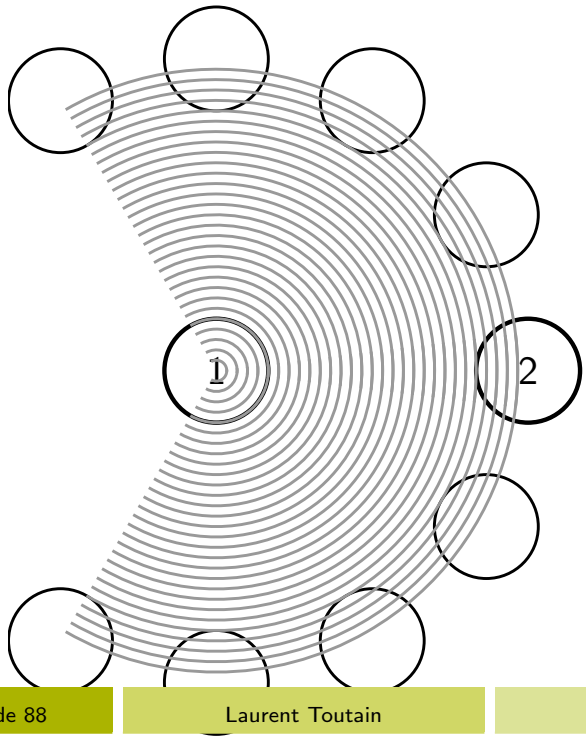
- 01 000001 Uncompressed IPv6
- 01 000010 Compressed IPv6 (obsolete)
- 01 010000 Broadcast
  - Used to suppress routing loops
- 01 1 Compressed header (new version)
- 10 xxxxxx MESH
  - Kind of tunnel to carry source and destination addresses
- 11 000xxx Fragmentation (first)?
- 11 100xxx Fragmentation (subsequent)

Discriminator cannot be used to identify Zigbee traffic.

2 models

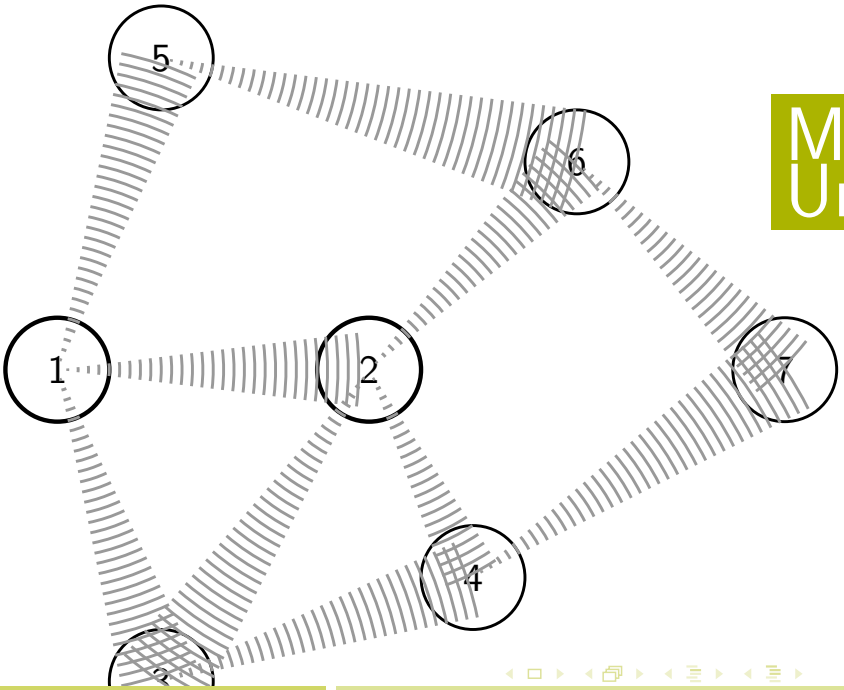
- Mesh-Under
  - L2 allows relaying between nodes
  - From IPv6, network appears as a link
  - 6LoWPAN adds two Dispatch values (Mesh and Broadcast)
- Route-Over
  - Routing (L3) is running on some nodes
  - Change for traditional IPv6 link model (no routers)
- Terminology:
  - 6LBR: Border Router (between LoWPAN and Internet)
  - 6LR: Node with routing protocol
  - 6LN: Node without routing/forwarding capabilities

6LoWPAN
IETF Working Groups ▶ 6LoWPAN



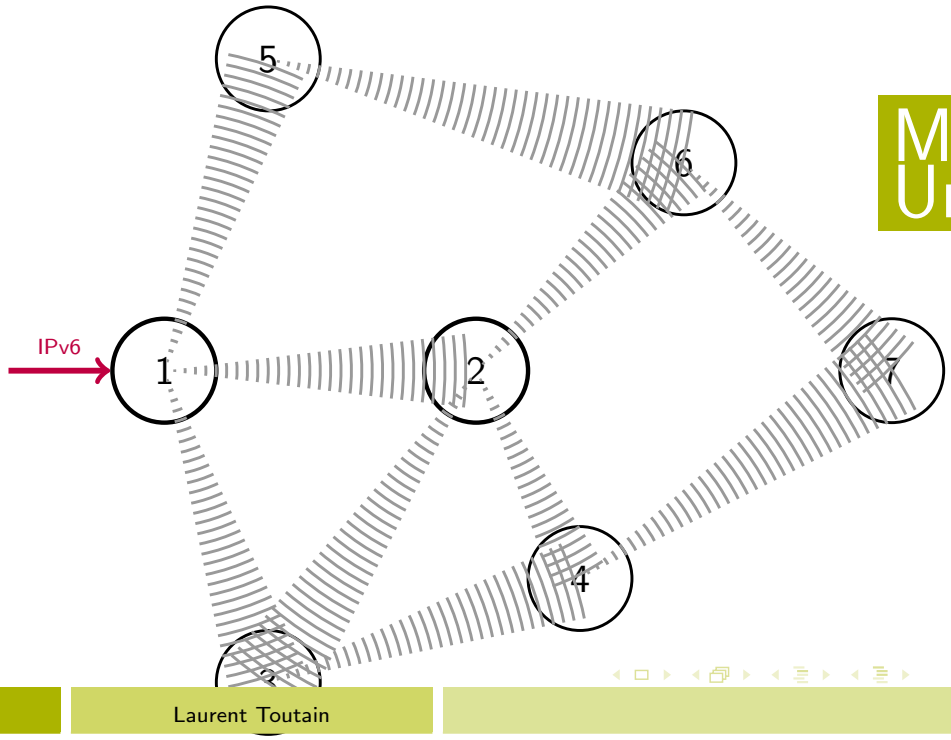
Star

Slide 88 Laurent Toutain Filière 2
TELECOM Bretagne

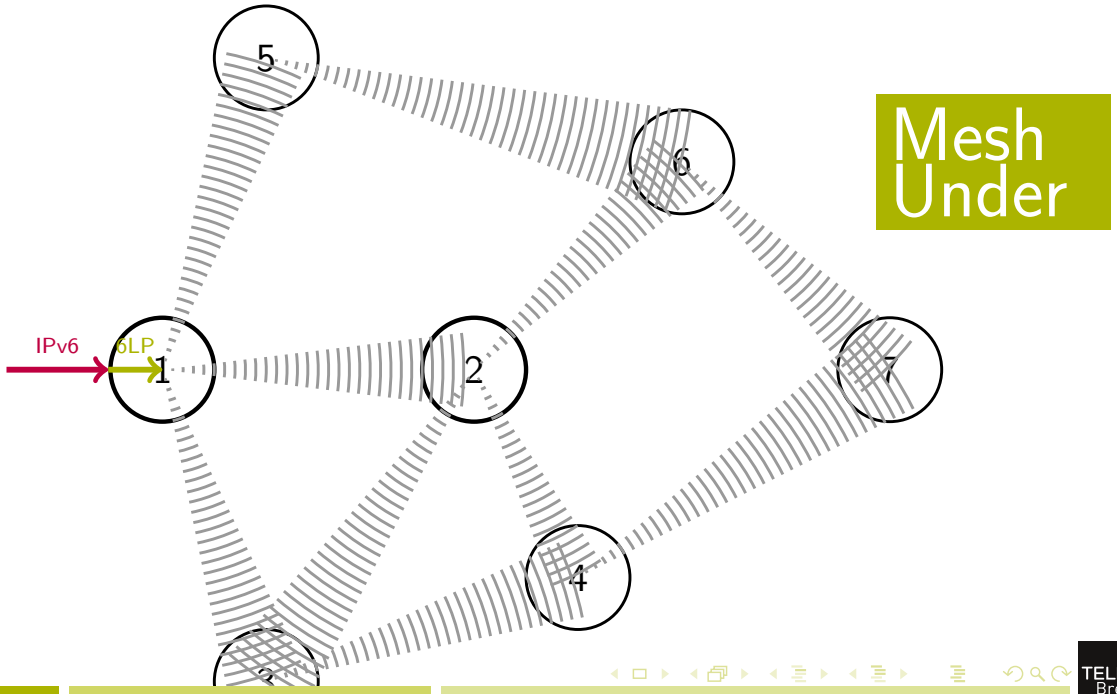


Mesh Under

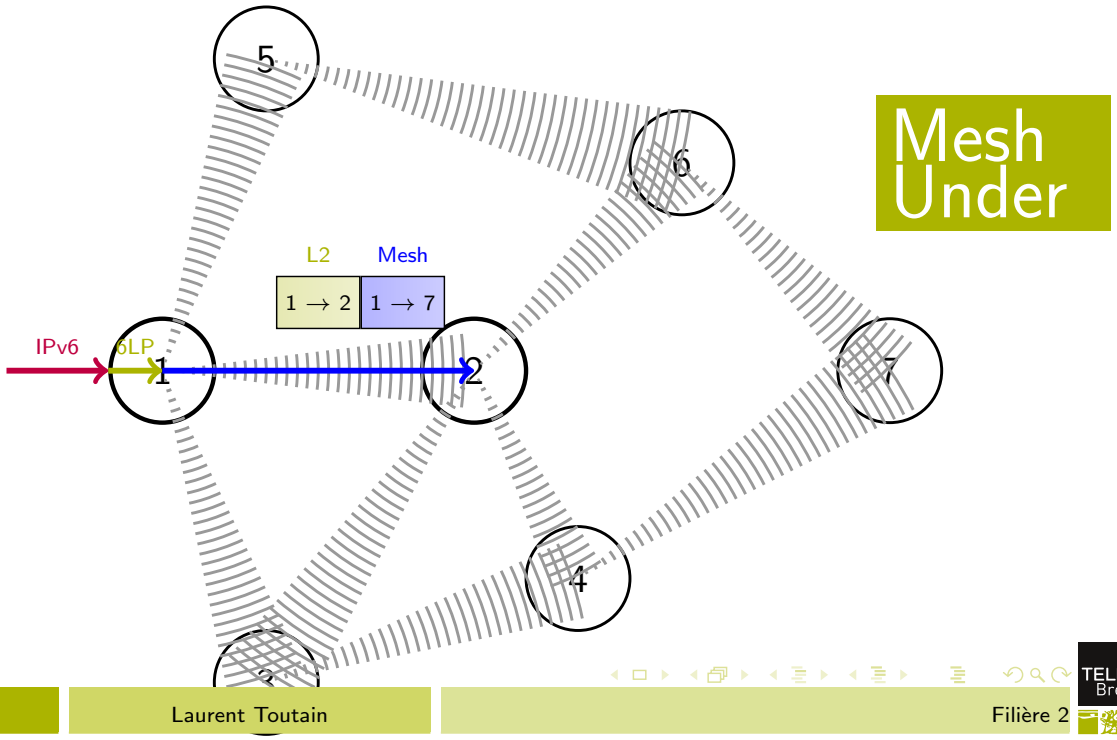
6LoWPAN
IETF Working Groups ▶ 6LoWPAN



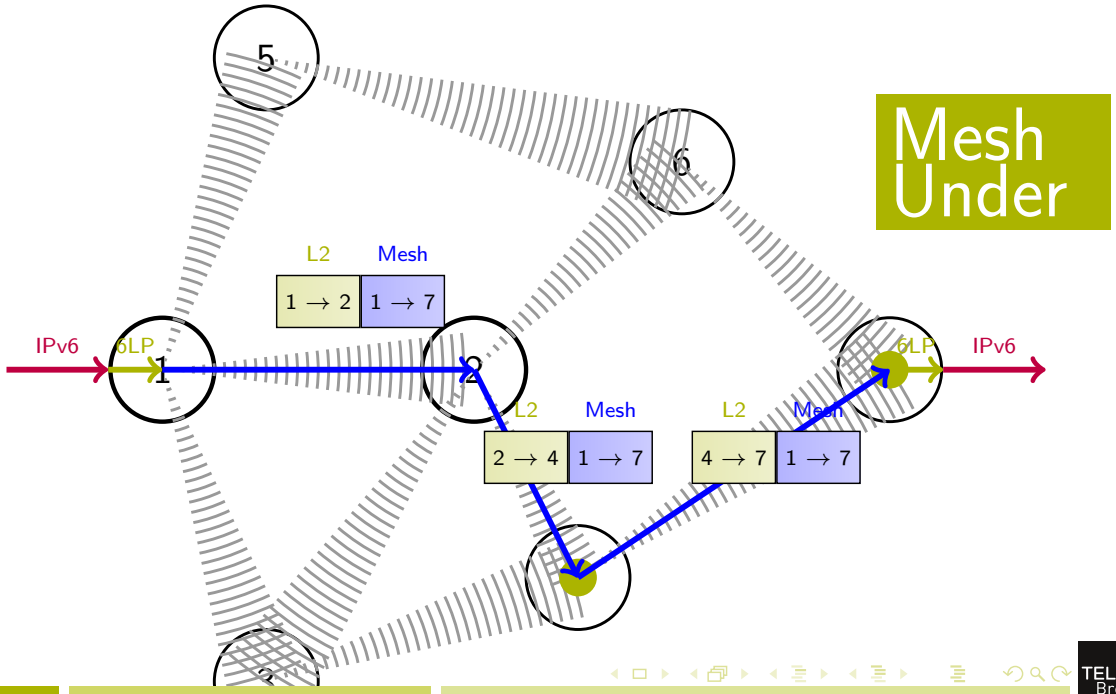
Mesh Under



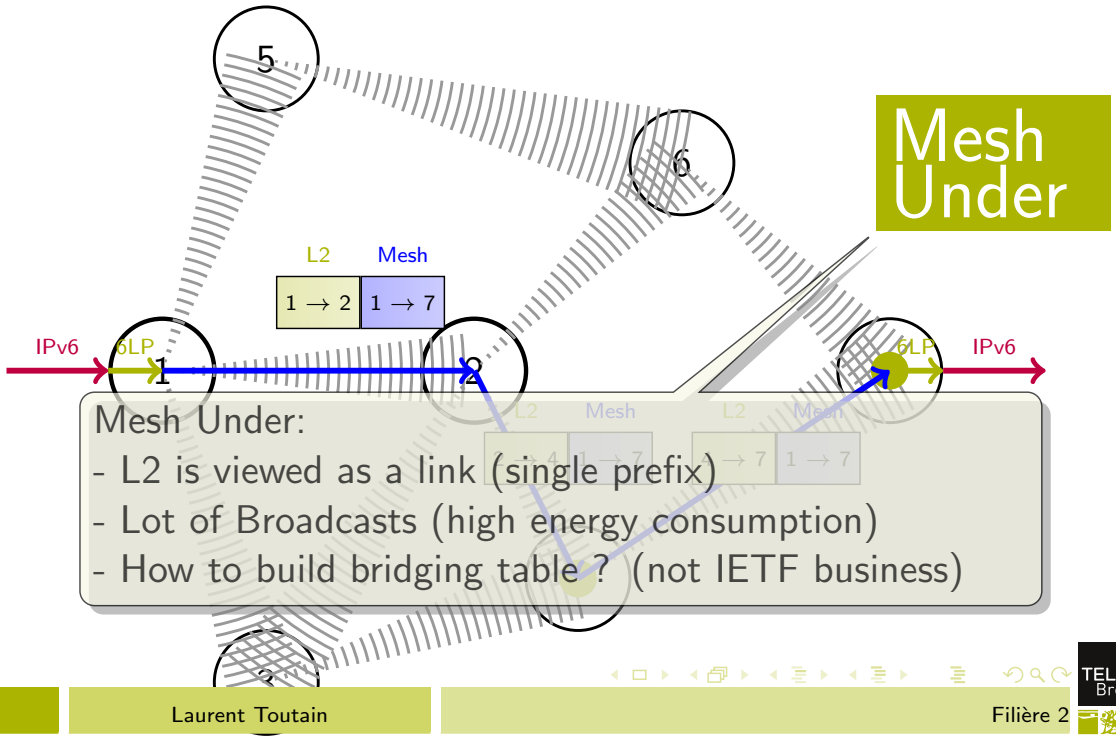
6LoWPAN
IETF Working Groups ▶ 6LoWPAN



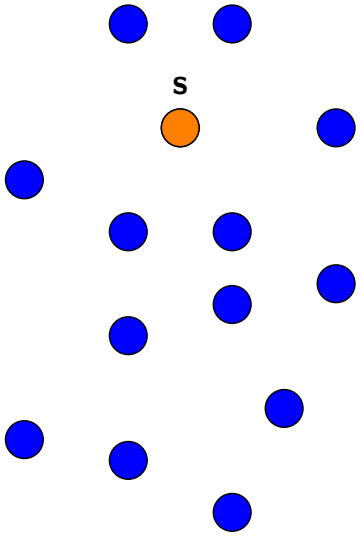
Slide 88 Laurent Toutain Filière 2
TELECOM Bretagne



6LoWPAN
IETF Working Groups ▶ 6LoWPAN

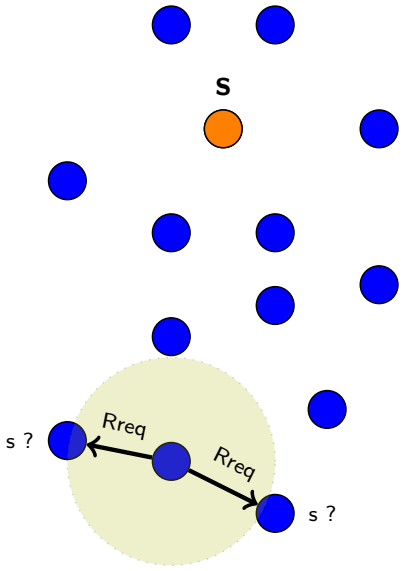


Example : Simple AODV (LOAD)

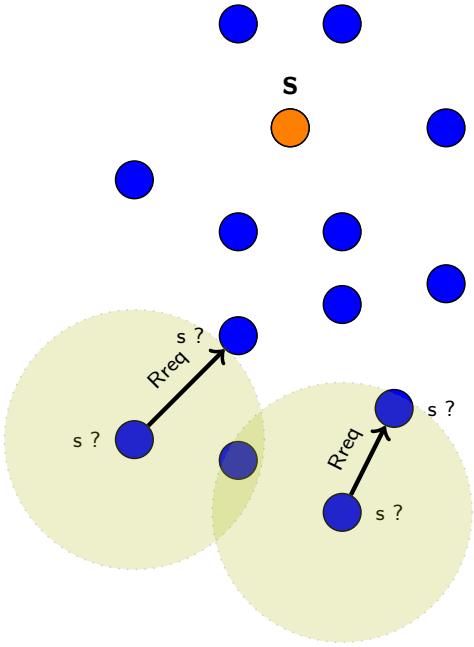


Example : Simple AODV (LOAD)

IETF Working Groups ▶ 6LoWPAN

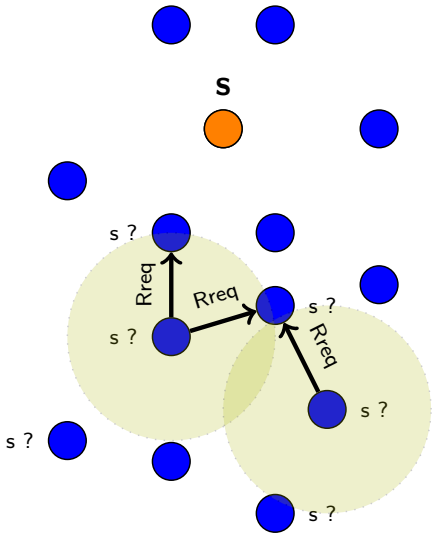


Example : Simple AODV (LOAD)

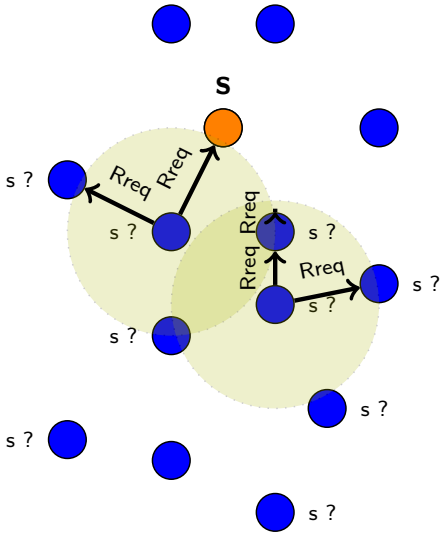


Example : Simple AODV (LOAD)

IETF Working Groups ▶ 6LoWPAN

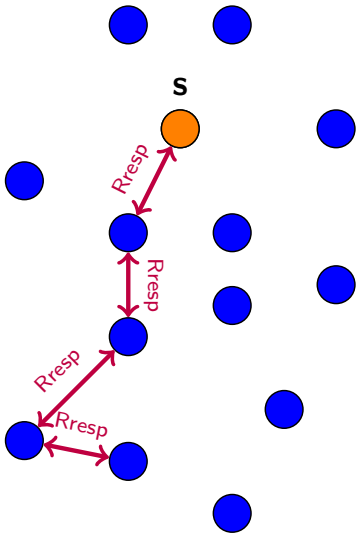


Example : Simple AODV (LOAD)



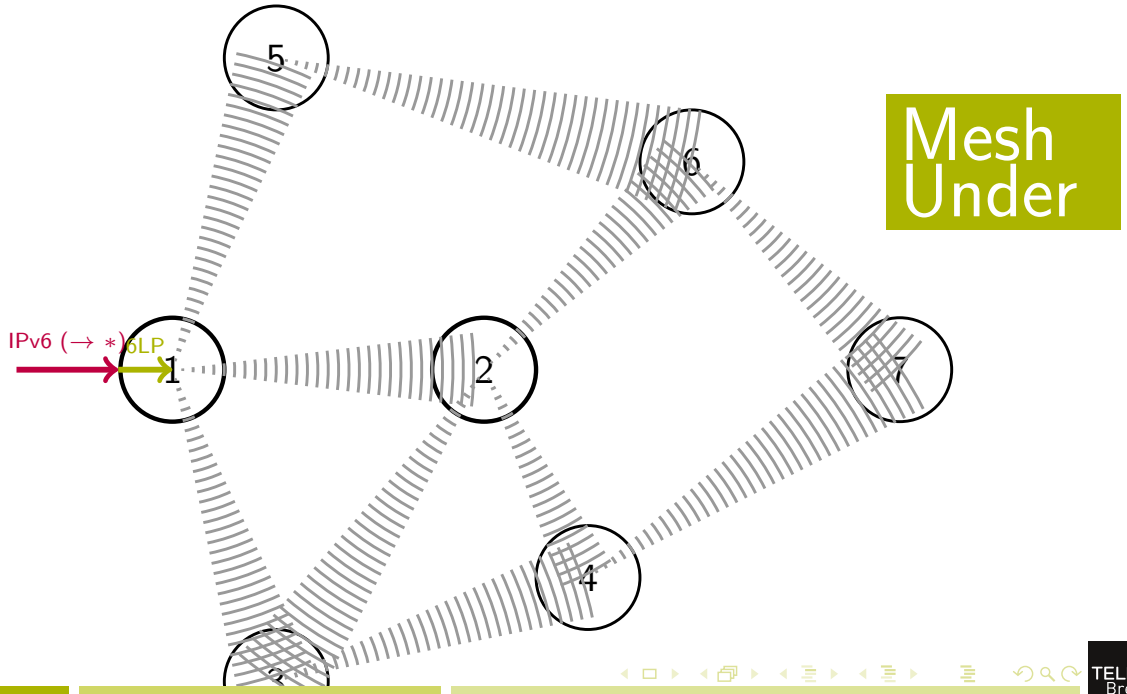
Example : Simple AODV (LOAD)

IETF Working Groups ▶ 6LoWPAN



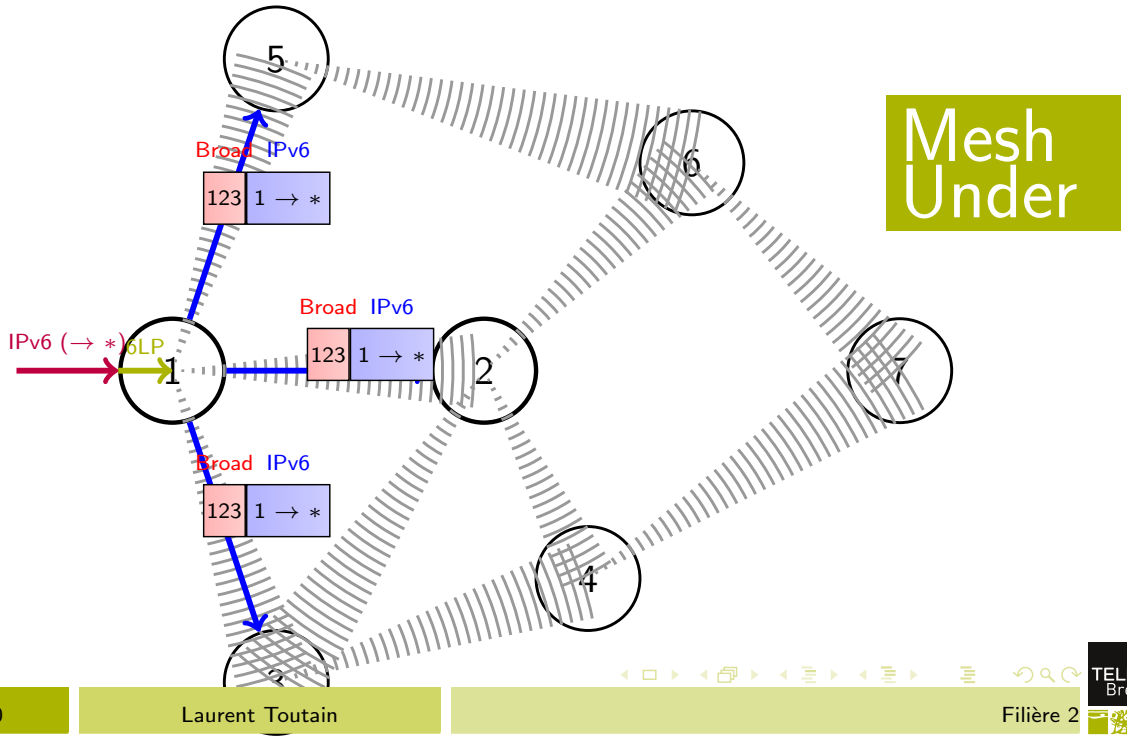
Slide 89 Laurent Toutain Filière 2

6LoWPAN: broadcast



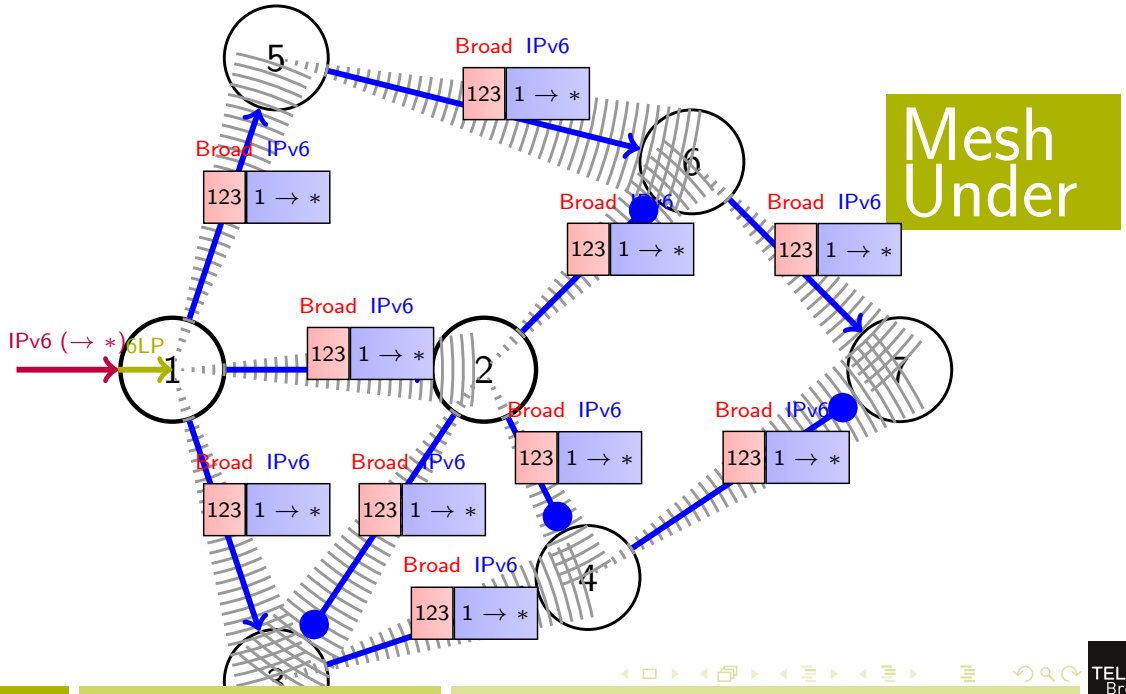
6LoWPAN: broadcast

IETF Working Groups ▶ 6LoWPAN

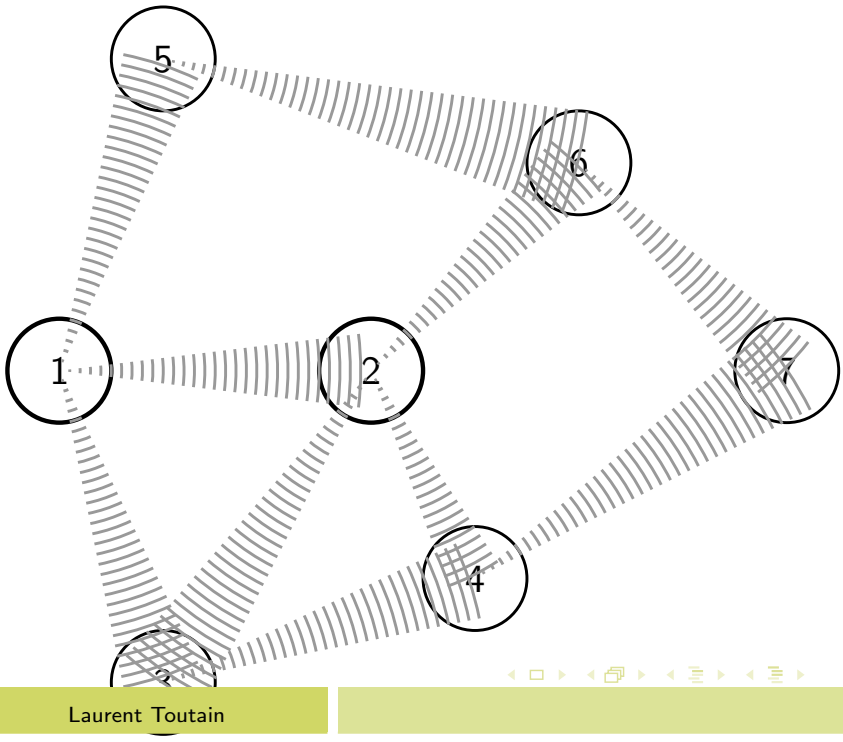


Slide 90 Laurent Toutain Filière 2
TELECOM Bretagne

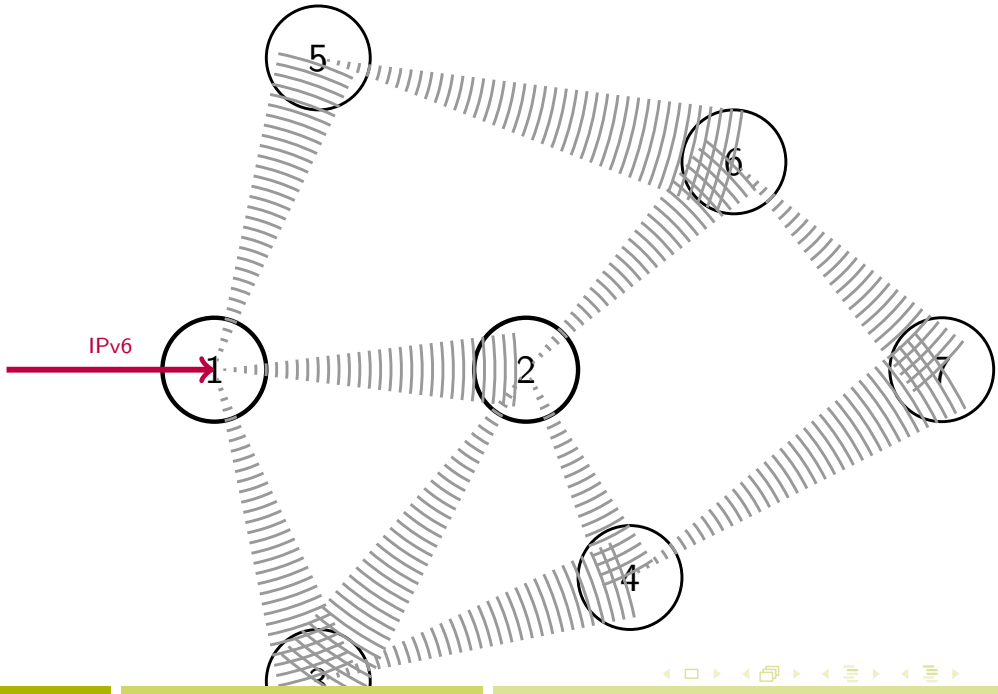
6LoWPAN: broadcast



6LoWPAN: Route Over

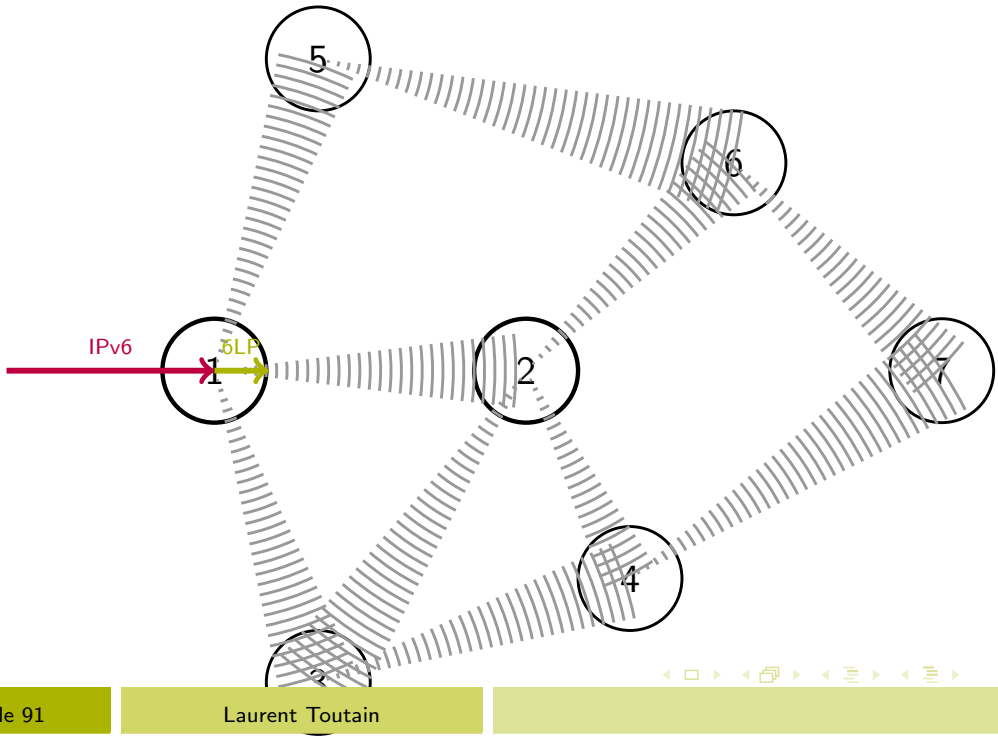


6LoWPAN: Route Over

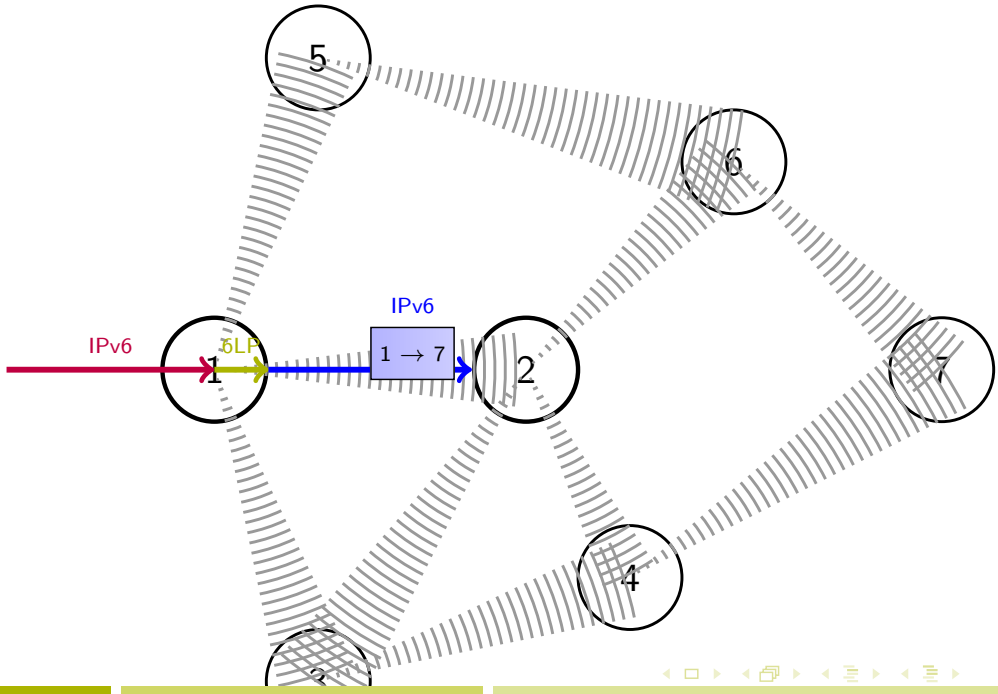


6LoWPAN: Route Over

IETF Working Groups ▶ 6LoWPAN

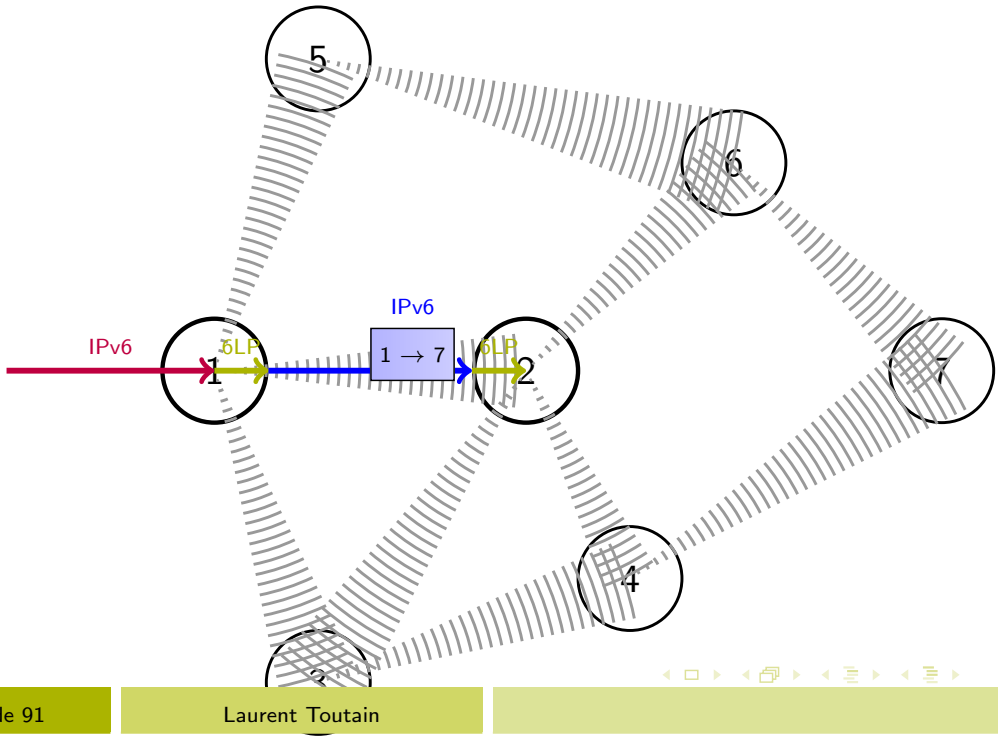


6LoWPAN: Route Over

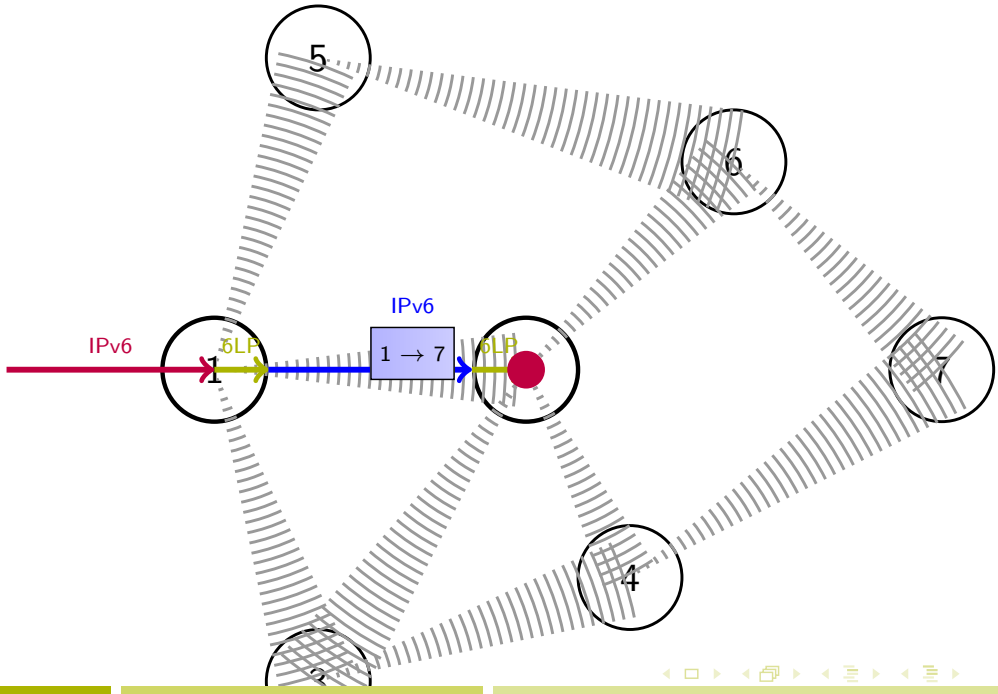


6LoWPAN: Route Over

IETF Working Groups ▶ 6LoWPAN

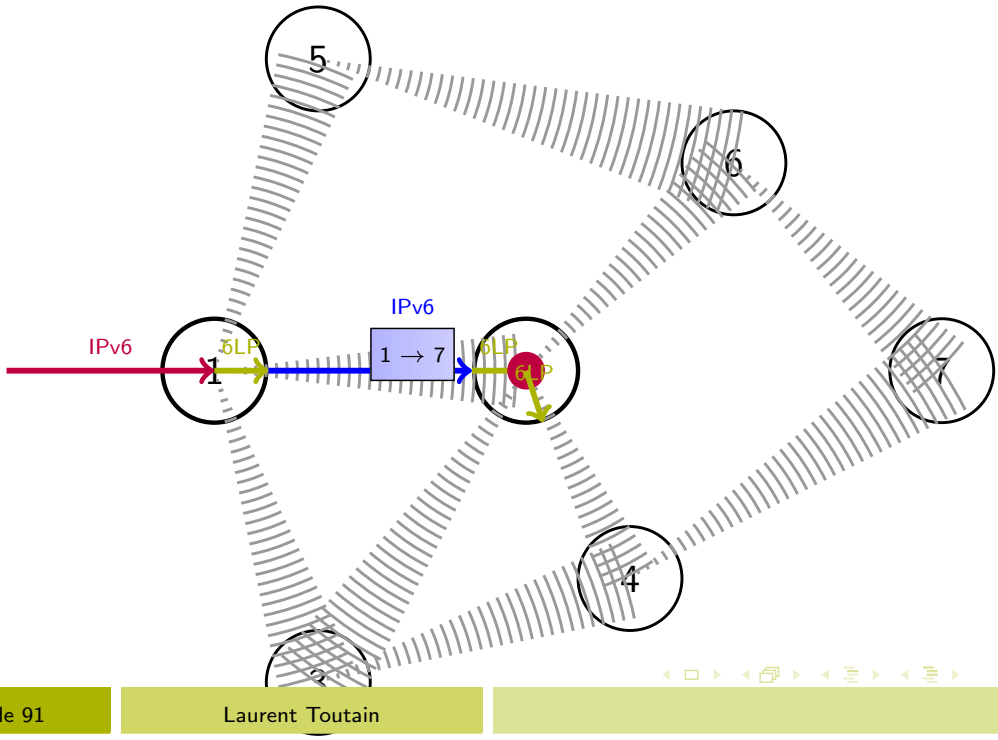


6LoWPAN: Route Over

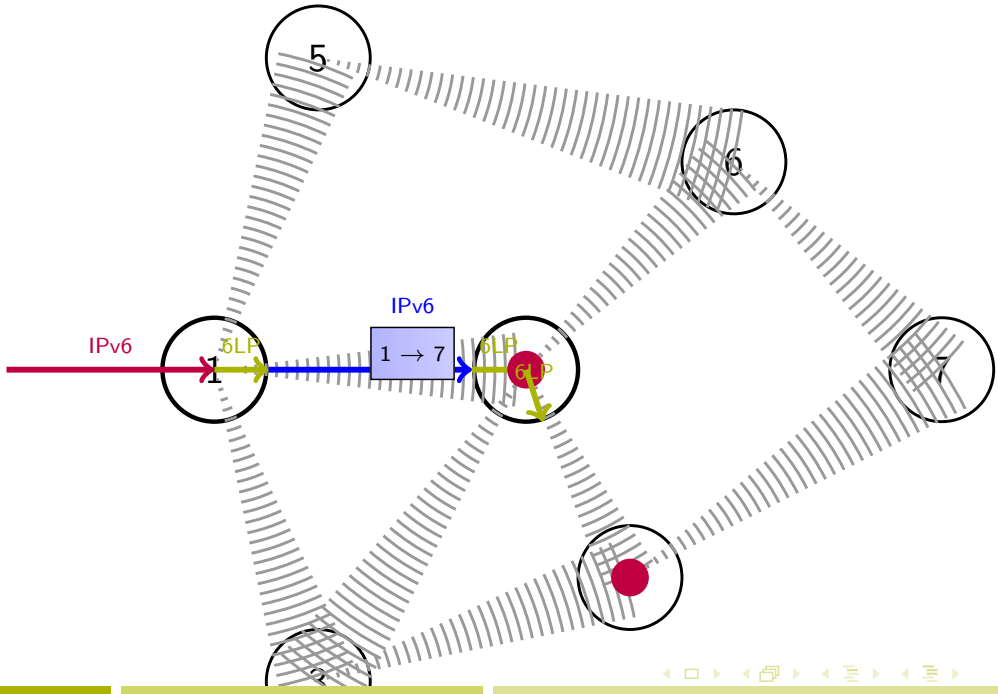


6LoWPAN: Route Over

IETF Working Groups ▶ 6LoWPAN

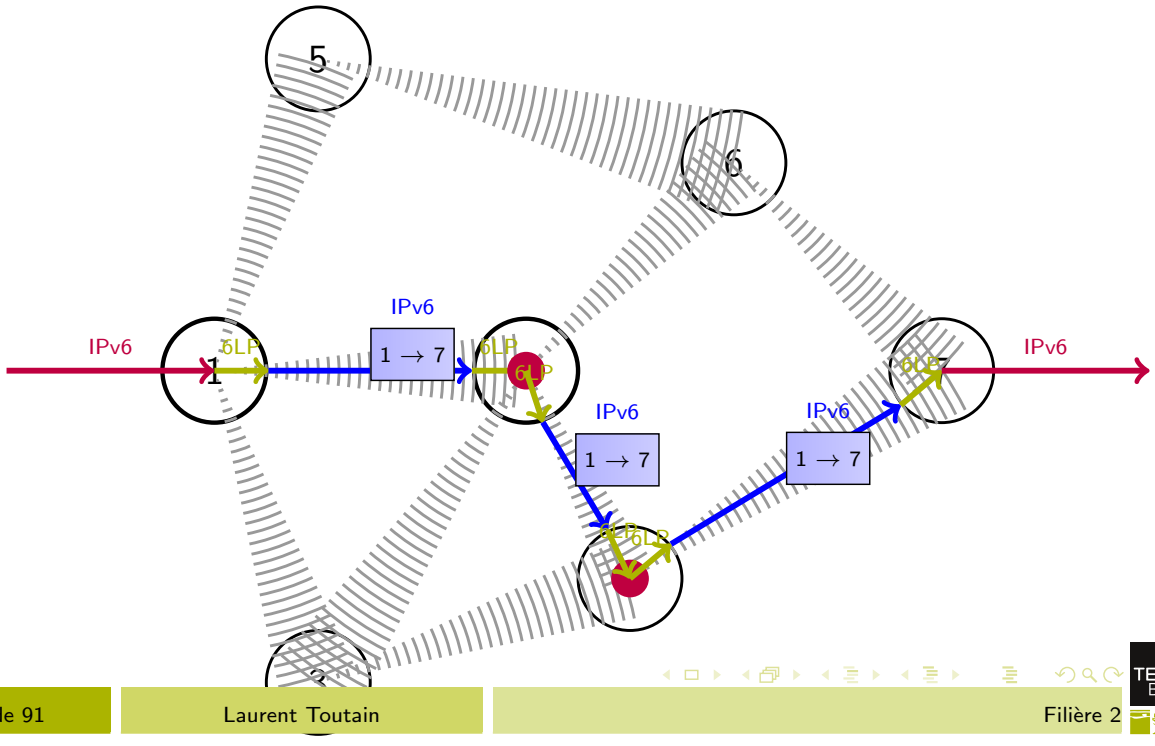


6LoWPAN: Route Over

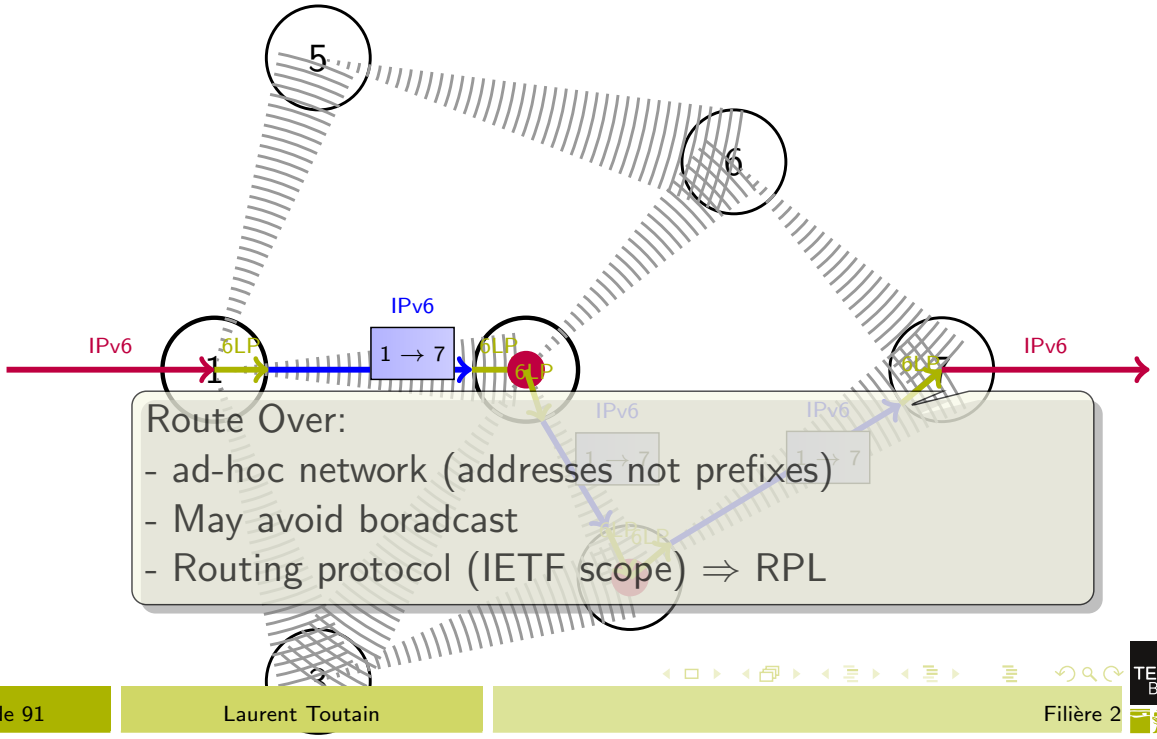


6LoWPAN: Route Over

IETF Working Groups ▶ 6LoWPAN

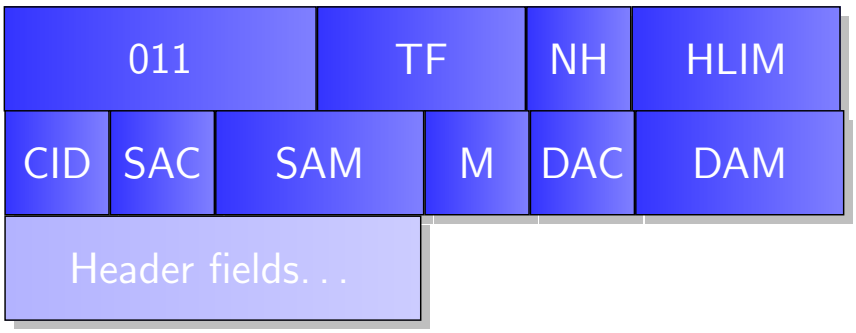


6LoWPAN: Route Over



6LoWPAN

- 6LoWPAN (RFC4944) or draft:
  - Compression of the IPv6 header



- Create contexts for well-known prefixes

Bitmap

- `TF`: DiffServ Field (DSCP: 6 bits), the Explicit Congestion Notification (ECN: 2 bits) and the flow label (20 bits)
- `NH`: 0 sent in in Header field, 1 L4 Dispatch after Header (allow L4 compression)
- `HLIM`: well know values
- `CID`: Add a context to allow 16 source and destination prefixes

SAC and SAM: Source Address Compression

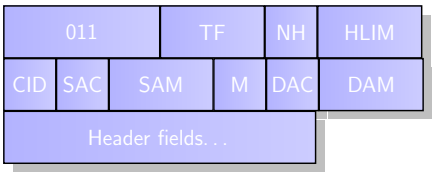
SAC \SAM	00	01	10	11
IID		64 first prefix bits are elided, IID is fully sent	112 first prefix bits are elided, last 16 IID bits are sent	128 are elided
0: LL	Address is send completely (Link Local and Global)	prefix is `FE80::/64`	prefix is `FE80::0:ff:fe00:/112`	prefix is `FE80::/64` and IID is taken from L2 source address.
1: Global	Unspecified address (`::/0` (fully elided))	prefix is given by the context	prefix is given by the context, IID starts with `0000:00ff:fe00:` and 16 bits inline	prefix is given by the contect and IID is taken from L2 source address.

M, DAC and DAM: Dest. Address Compression

M-DAC \DAM	00	01	10	11
00 Link Local	Address is send completely (Link Local and Global)	prefix is FE80::/64	prefix is FE80::0:ff:fe00:/112	prefix is FE80::/64 and IID is taken from L2 source address.
01 Global	reserved	prefix is given by the context	prefix is given by the context, IID starts with 0000:00ff:fe00: and 16 bits inline	prefix is given by the contect and IID is taken from L2 source address.
10 local Multicast	Address is send completely	48 bits are sent in-line and are spread in a multicast address the following way FFXX::00XX:XXXX:XXXX	32 bits are sent in-line and are spread in a multicast address the following way FFXX::00XX:XXXX	8 bits are sent in-line and are spread in a multicast address the following way FF02::00XX
11 Global Multicast	48 bits are sent. They are used for large scale multicast as defined in RFC 3956. Context value contains the Rendez-vous Point address	reserved	reserved	reserved

Example: Compress

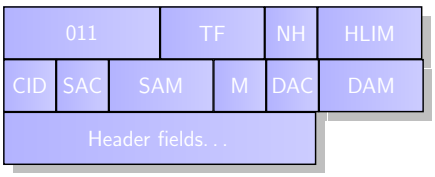

```
6e 00 00 00   00 40 3a ff
fe 80 00 00   00 00 00 00
02 01 64 ff   fe 2f fc 0a
ff 02 00 00   00 00 00 00
00 00 00 00   00 00 00 01
86 00 8b a3   40 00 07 08
00 00 00 00   00 00 00 00
01 01 00 01   64 2f fc 0a
05 01 00 00   00 00 05 dc
03 04 40 c0   00 27 8d 00
00 09 3a 80   00 00 00 00
20 01 06 60   73 01 37 28
00 00 00 00   00 00 00 00
```



Example: Compress

version

```
6e 00 00 00   00 40 3a ff
fe 80 00 00   00 00 00 00
02 01 64 ff   fe 2f fc 0a
ff 02 00 00   00 00 00 00
00 00 00 00   00 00 00 01
86 00 8b a3   40 00 07 08
00 00 00 00   00 00 00 00
01 01 00 01   64 2f fc 0a
05 01 00 00   00 00 05 dc
03 04 40 c0   00 27 8d 00
00 09 3a 80   00 00 00 00
20 01 06 60   73 01 37 28
00 00 00 00   00 00 00 00
```

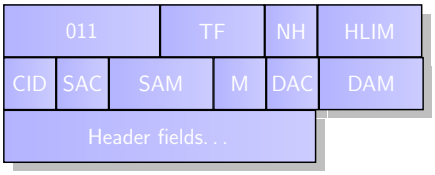


Example: Compress

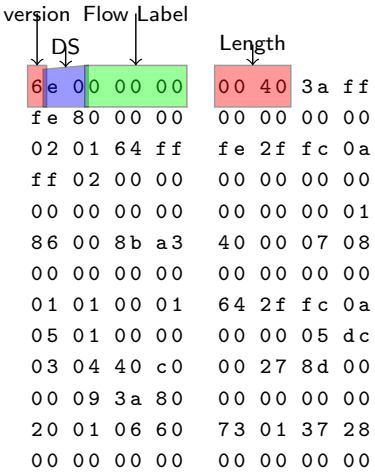
version Flow Label

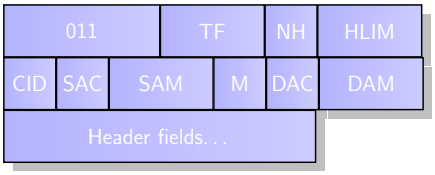
DS

```
6e 00 00 00   00 40 3a ff
fe 80 00 00   00 00 00 00
02 01 64 ff   fe 2f fc 0a
ff 02 00 00   00 00 00 00
00 00 00 00   00 00 00 01
86 00 8b a3   40 00 07 08
00 00 00 00   00 00 00 00
01 01 00 01   64 2f fc 0a
05 01 00 00   00 00 05 dc
03 04 40 c0   00 27 8d 00
00 09 3a 80   00 00 00 00
20 01 06 60   73 01 37 28
00 00 00 00   00 00 00 00
```



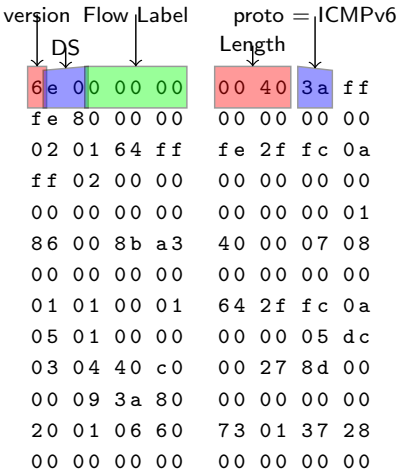
Example: Compress

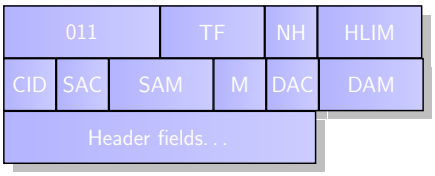




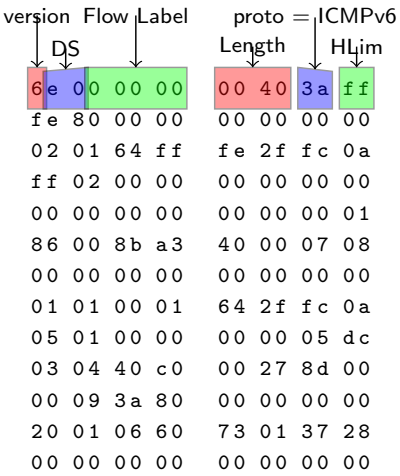
Example: Compress

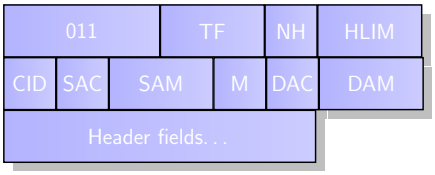
IETF Working Groups ▶ 6LoWPAN





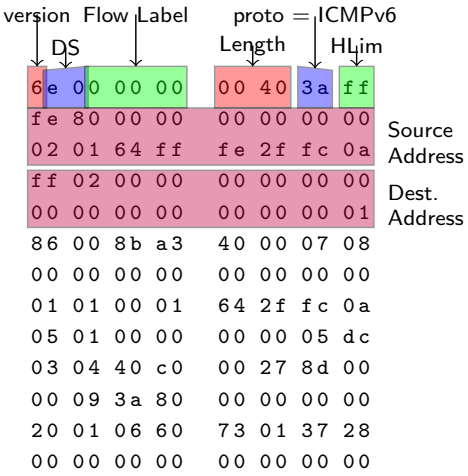
Example: Compress

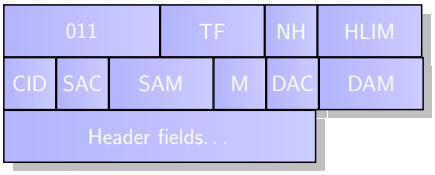




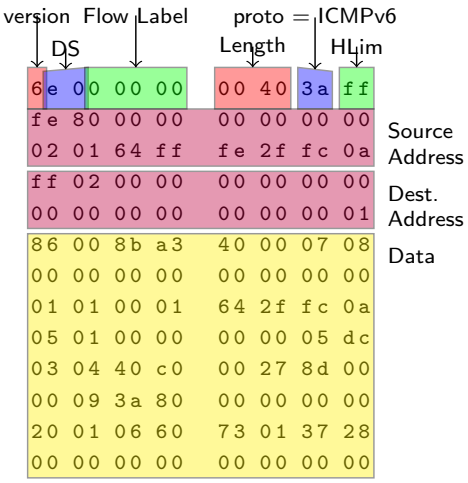
Example: Compress

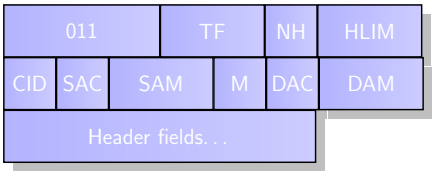
IETF Working Groups ▶ 6LoWPAN





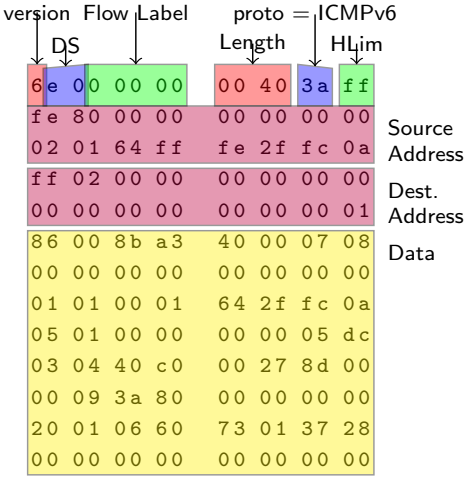
Example: Compress

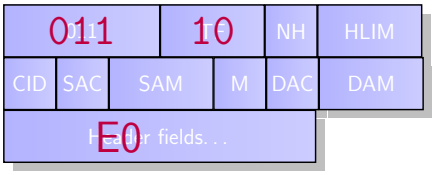




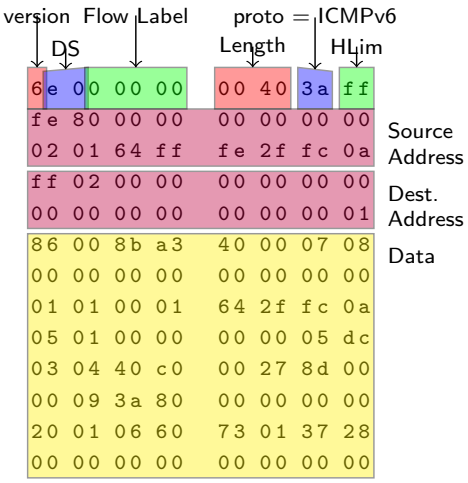
Example: Compress

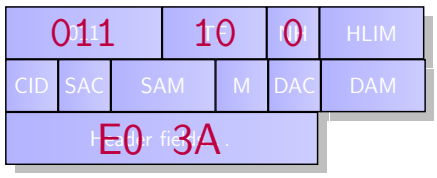
IETF Working Groups ▶ 6LoWPAN





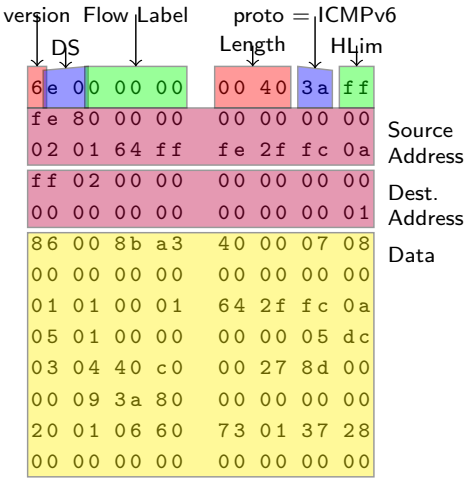
Example: Compress

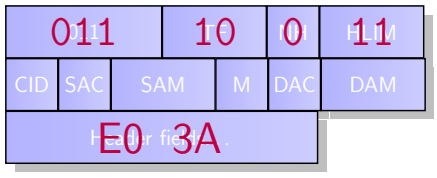




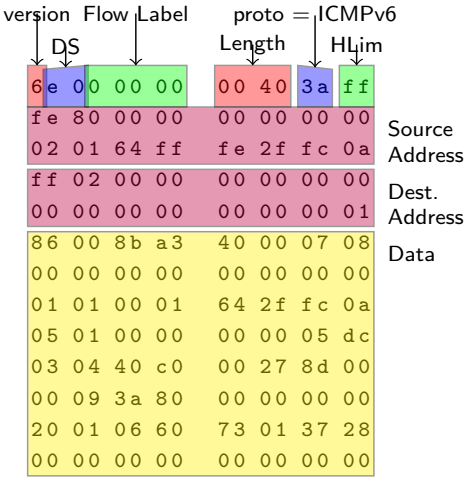
Example: Compress

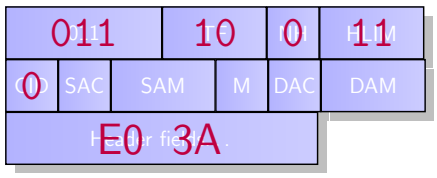
IETF Working Groups ▶ 6LoWPAN





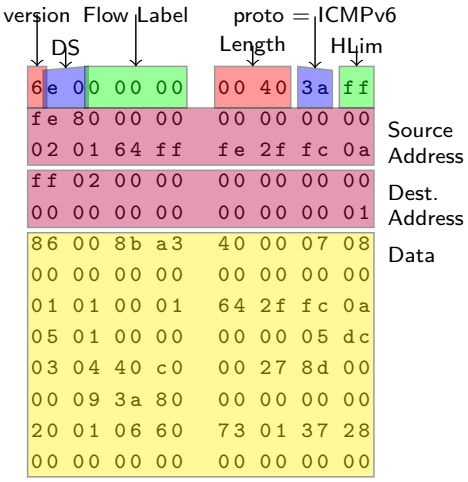
Example: Compress

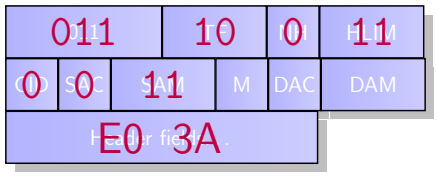




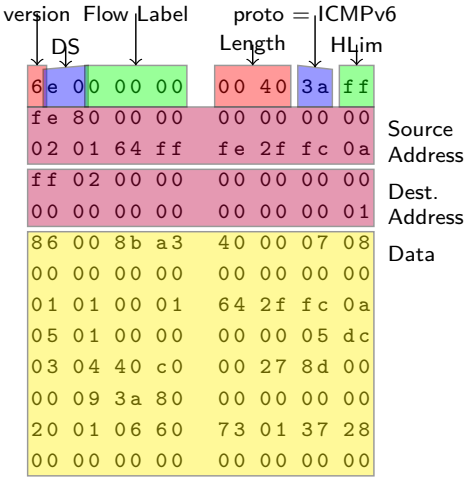
Example: Compress

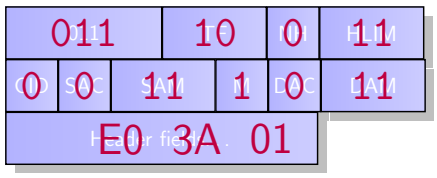
IETF Working Groups ▶ 6LoWPAN





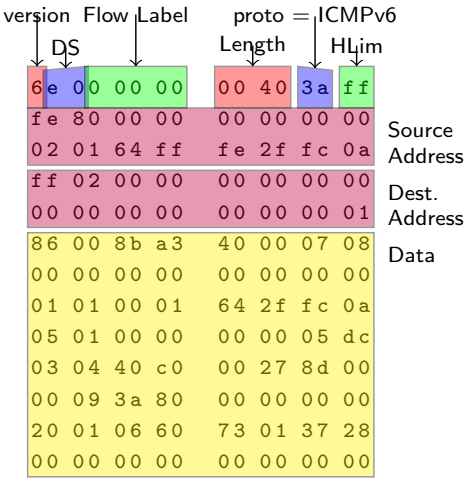
Example: Compress

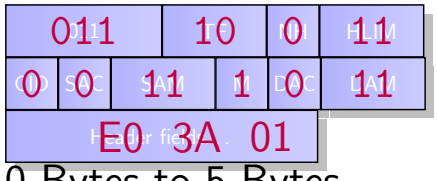




Example: Compress

IETF Working Groups ▶ 6LoWPAN

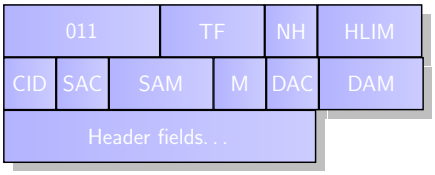




40 Bytes to 5 Bytes

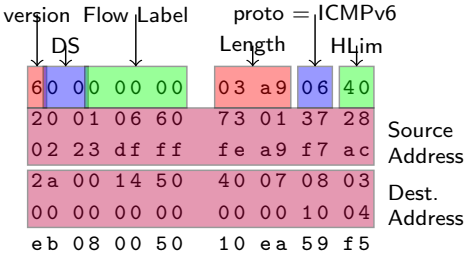
Example: Compress


```
60 00 00 00   03 a9 06 40
20 01 06 60   73 01 37 28
02 23 df ff   fe a9 f7 ac
2a 00 14 50   40 07 08 03
00 00 00 00   00 00 10 04
eb 08 00 50   10 ea 59 f5
3b 1a 5e 5a   80 18 80 55
f6 a0 00 00   01 01 08 0a
03 e7 60 72   78 aa 80 5d
47 45 54 20   2f 5f 5f 75
74 6d 2e 67   69 66 3f 75
74 6d 77 76   3d 35 2e 34
2e 34 26 75   74 6d 73 3d
33 30 37 26   75 74 6d 6e
3d 32
```

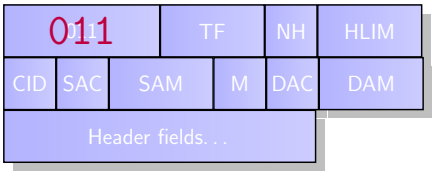


Example: Compress

IETF Working Groups ▶ 6LoWPAN

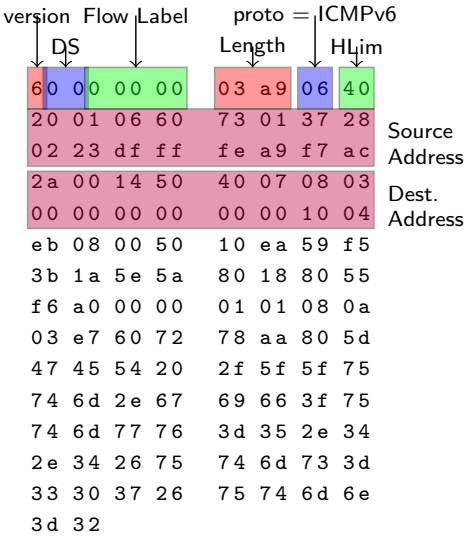


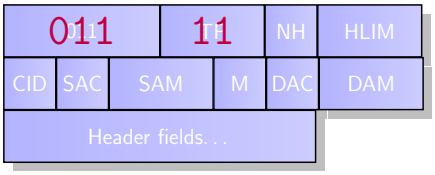

```
eb 08 00 50   10 ea 59 f5
3b 1a 5e 5a   80 18 80 55
f6 a0 00 00   01 01 08 0a
03 e7 60 72   78 aa 80 5d
47 45 54 20   2f 5f 5f 75
74 6d 2e 67   69 66 3f 75
74 6d 77 76   3d 35 2e 34
2e 34 26 75   74 6d 73 3d
33 30 37 26   75 74 6d 6e
3d 32
```



Slide 97 Laurent Toutain Filière 2

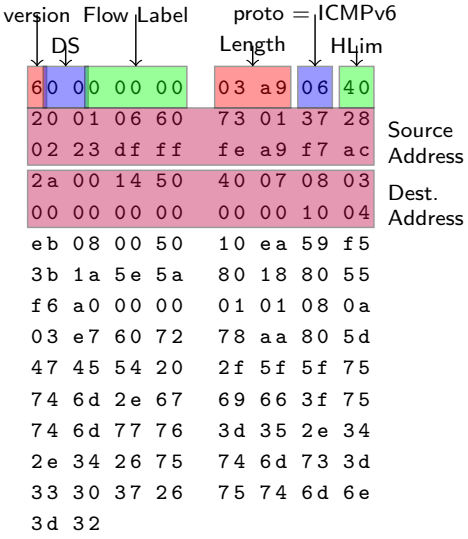
Example: Compress

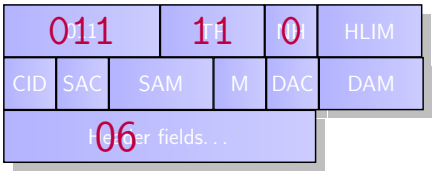




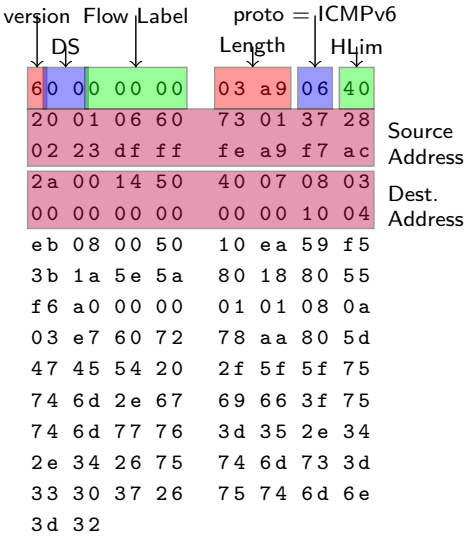
Example: Compress

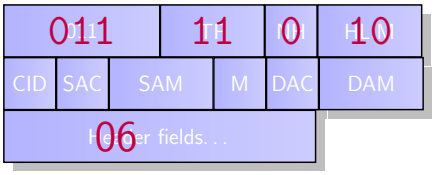
IETF Working Groups ▶ 6LoWPAN





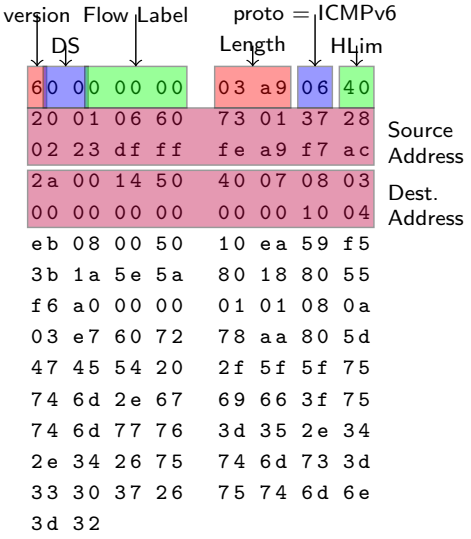
Example: Compress

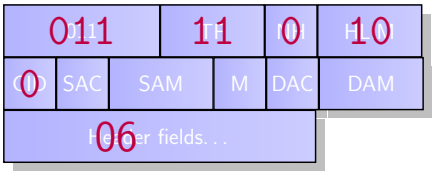




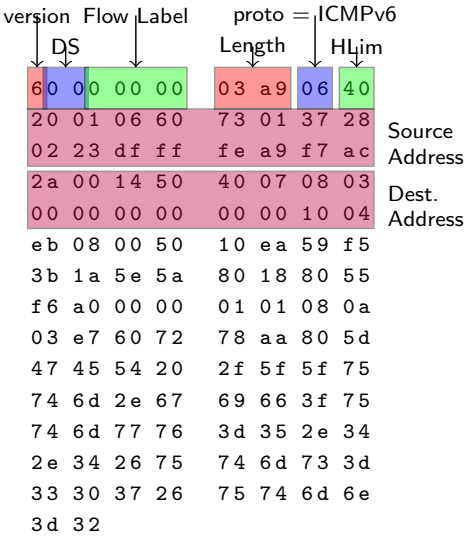
Example: Compress

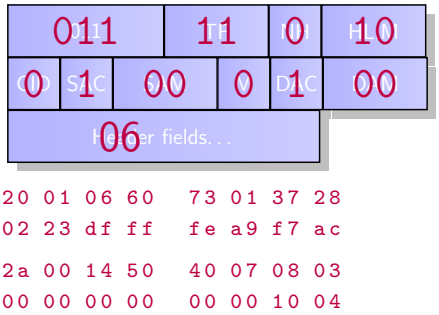
IETF Working Groups ▶ 6LoWPAN





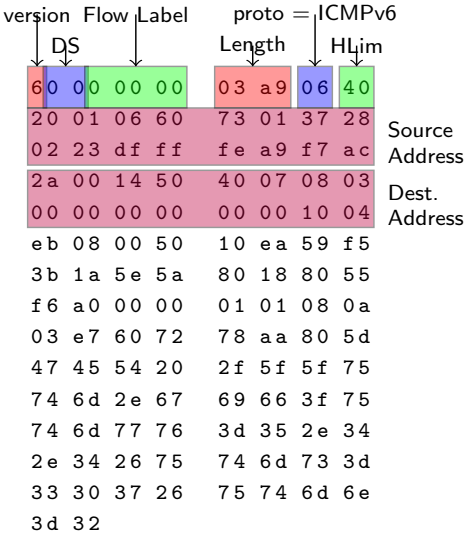
Example: Compress

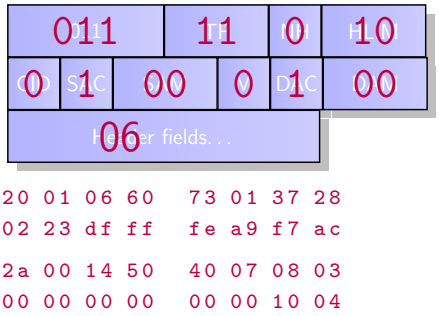




Example: Compress

IETF Working Groups ▶ 6LoWPAN





40 Bytes to 35 Bytes

Neighbor Discovery

Neighbor Discovery Protocol For LoWPAN

Limitations:

- radio range is limited, all the nodes cannot talk directly.
- range change
- Bidirectional traffic cannot be always guaranteed.
- The link definition is not clear
- Energy consumption must be limited
- Implementation must be kept as simple as possible

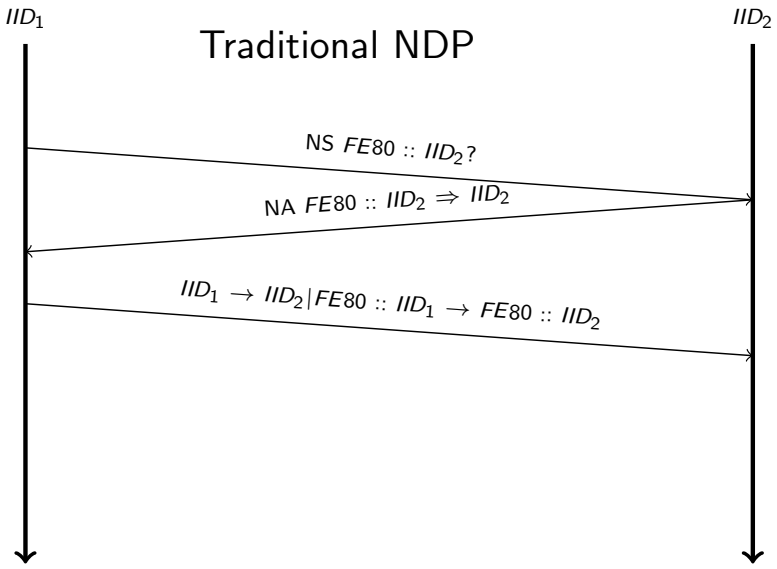
Neighbor Discovery Protocol For LoWPAN

Two models:

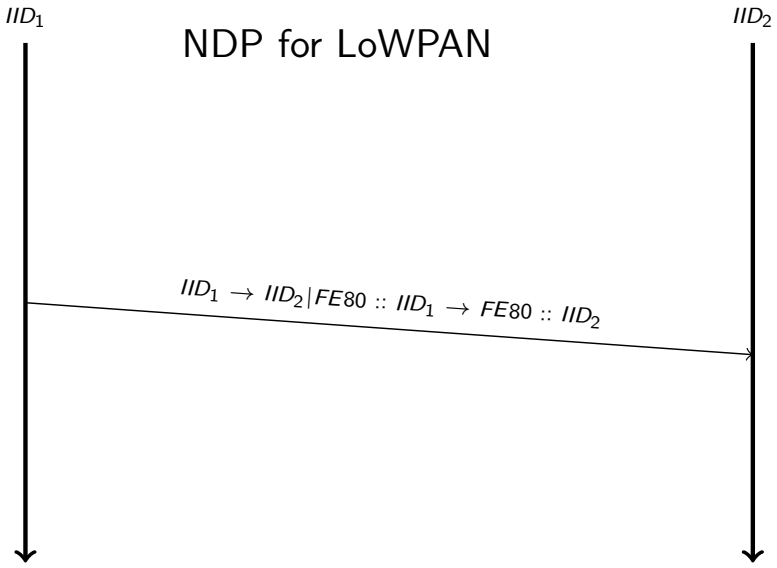
- Mesh-Under model: L3 multicast address mapped into a L2 broadcast address. No change
- Route-Over networks ≈ NBMA: control done by a server.
  - Multicast only allowed to discover neighbor routers:
    - Once the address of a router is learned, the traffic will be send in unicast.
    - No periodic RA
  - NDP do not, by construction, cross routers since original:
    - NDP for 6LoWPAN introduces the concept of Multi-Hop prefixes
- LL addresses are based on the EUI-64
  - unique: no need for DAD or NS

NDP options

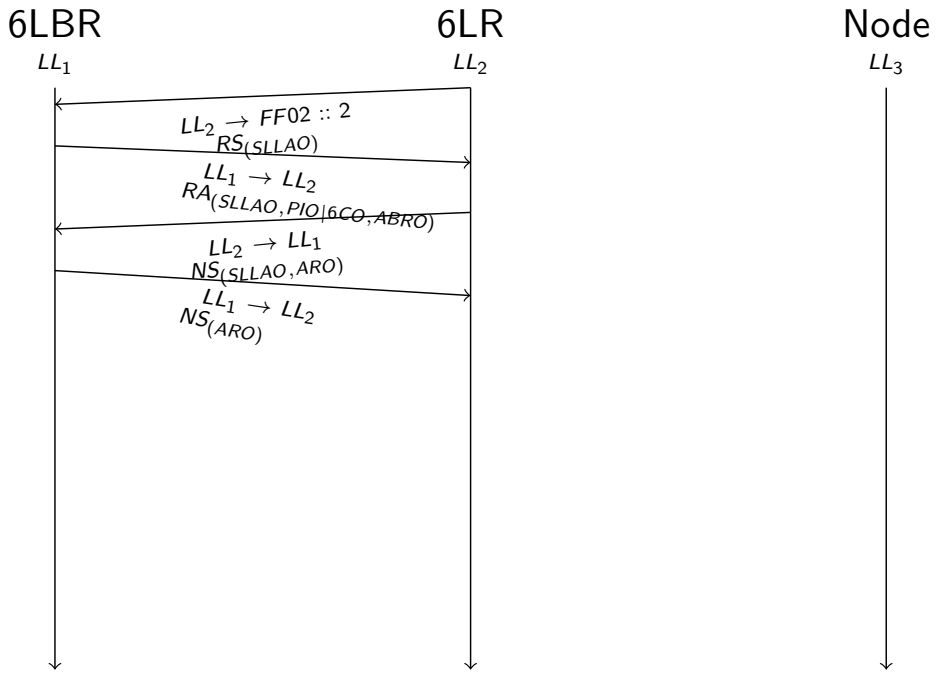
- 6LoWPAN uses RA, RS, NS and NA
- Only RS is sent in multicast: `FF02::2`
- Standard and new options are used:
  - SLLAO: Source Link-layer Address
  - PIO: Prefix Information
  - 6CO: 6LoWPAN Context Number
  - ABRO: Announcing Border Router
  - ARO: Address Registration
- If IID is based on MAC (or from DHCP): no DAD
- Currently there is no 6LoWPAN Compression for NDP messages.



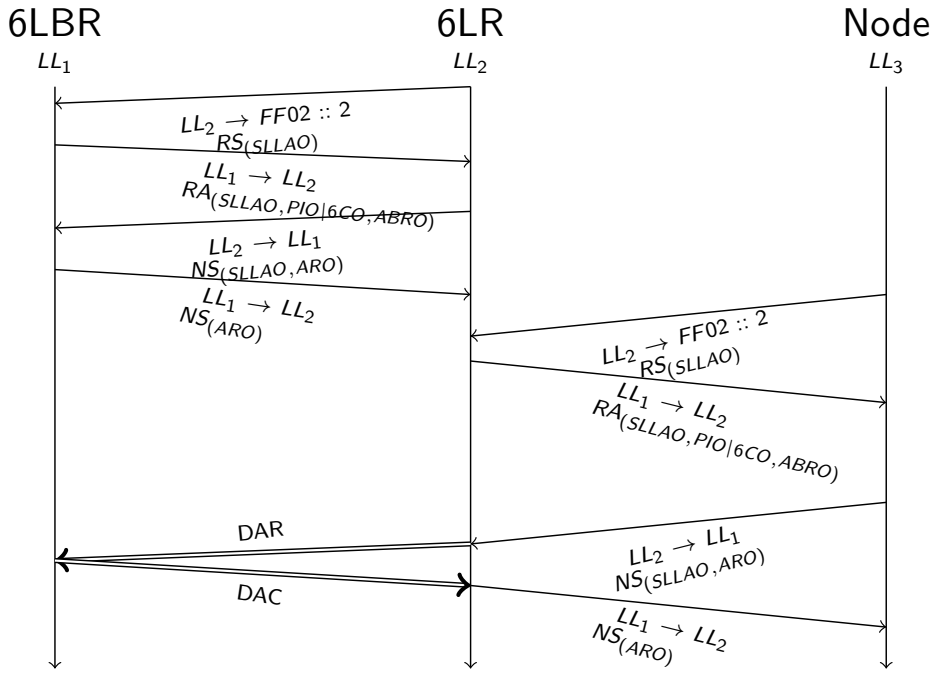
LL address
IETF Working Groups ▶ 6LoWPAN



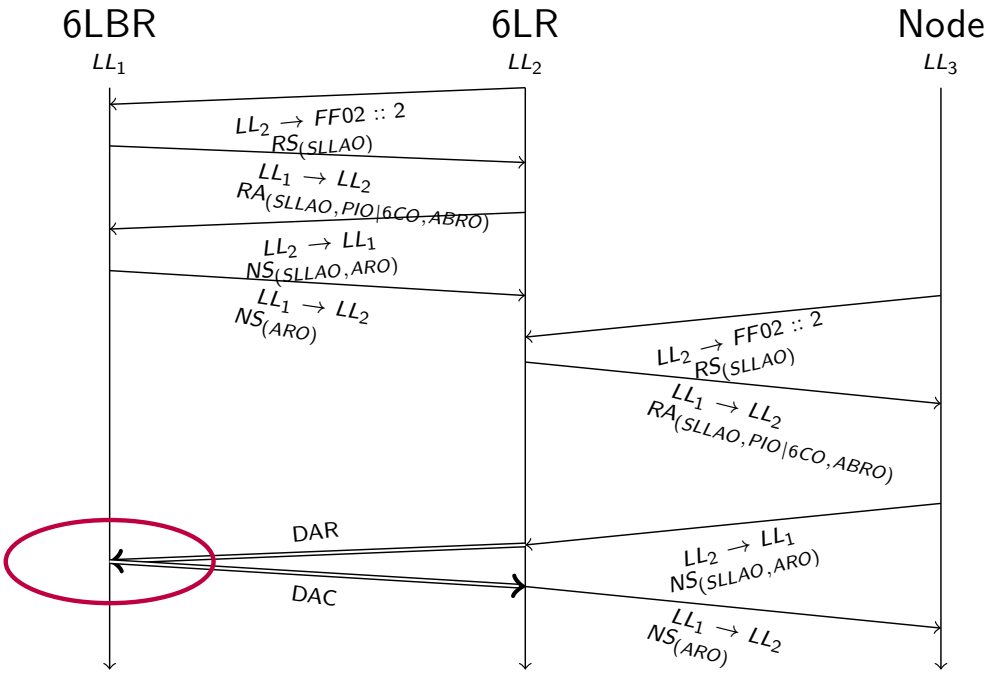
NDP for Global Addresses: Star



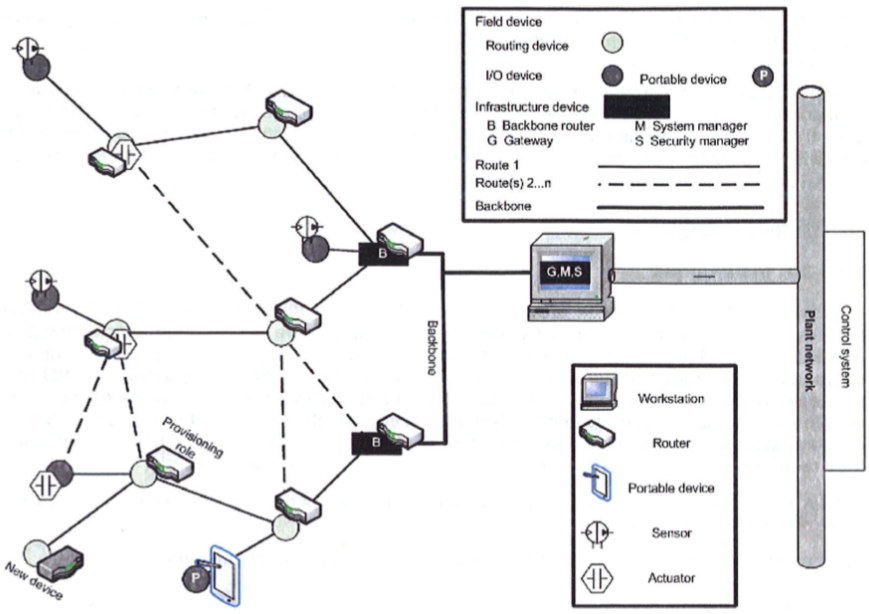
NDP for Global Addresses: Mesh

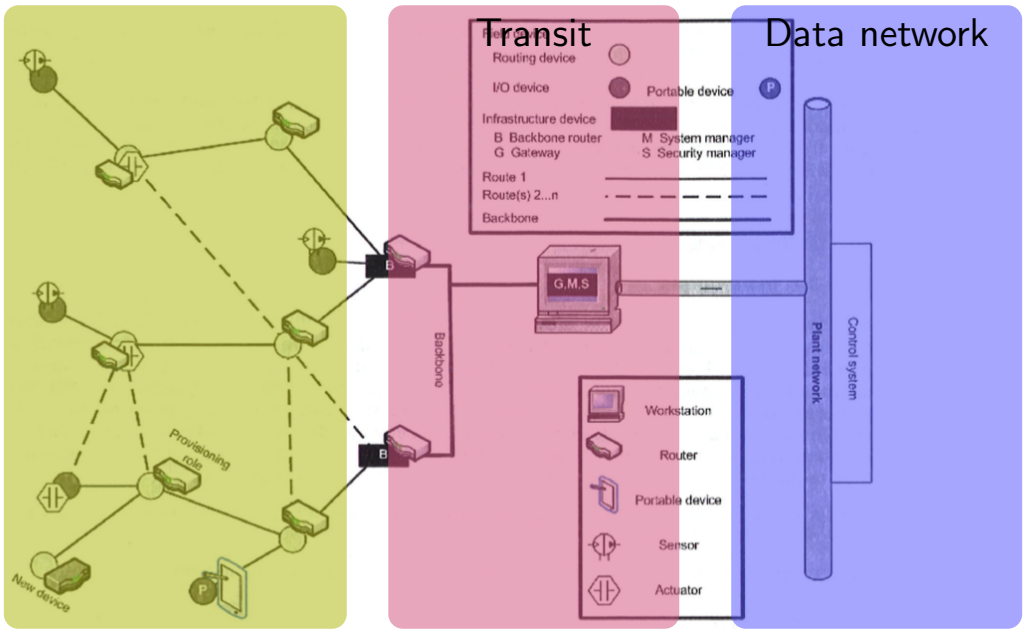


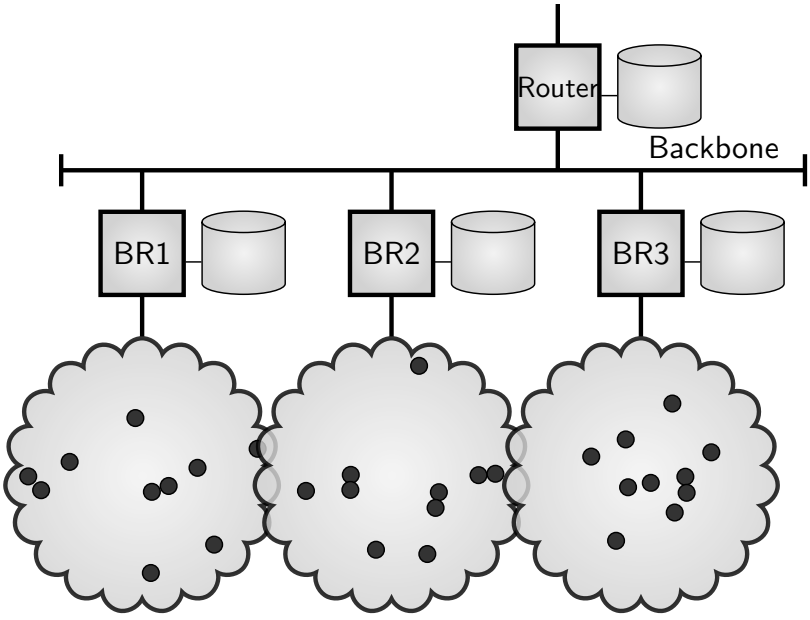
NDP for Global Addresses: Mesh



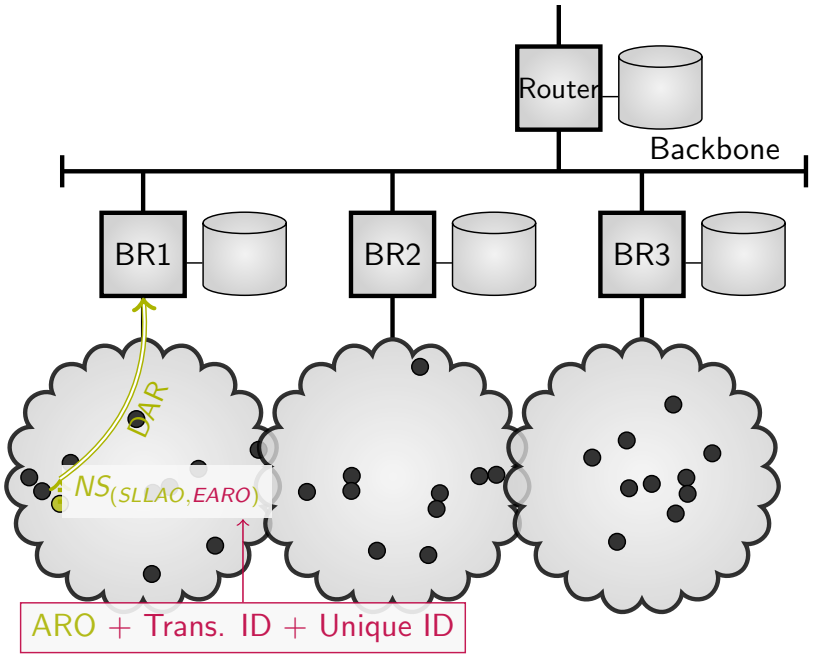
ISA 100



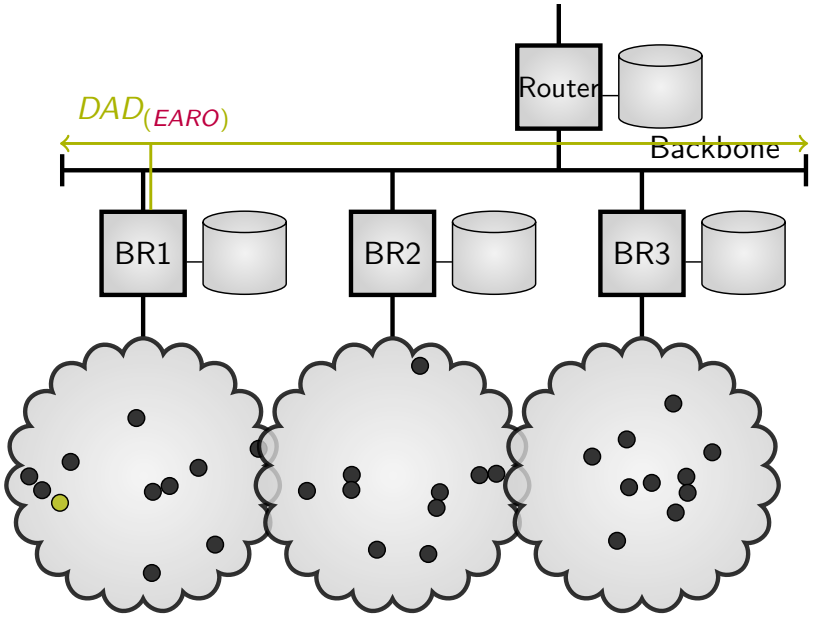

Transit
Data network
Routing device
I/O device
Portable device
Infrastructure device
B Backbone router
G Gateway
M System manager
S Security manager
Route 1
Route(s) 2...n
Backbone
G.M.S
Backbone
Plant network
Control system
Workstation
Router
Portable device
Sensor
Actuator
Provisioning role
New device


Router
Backbone
BR1
BR2
BR3

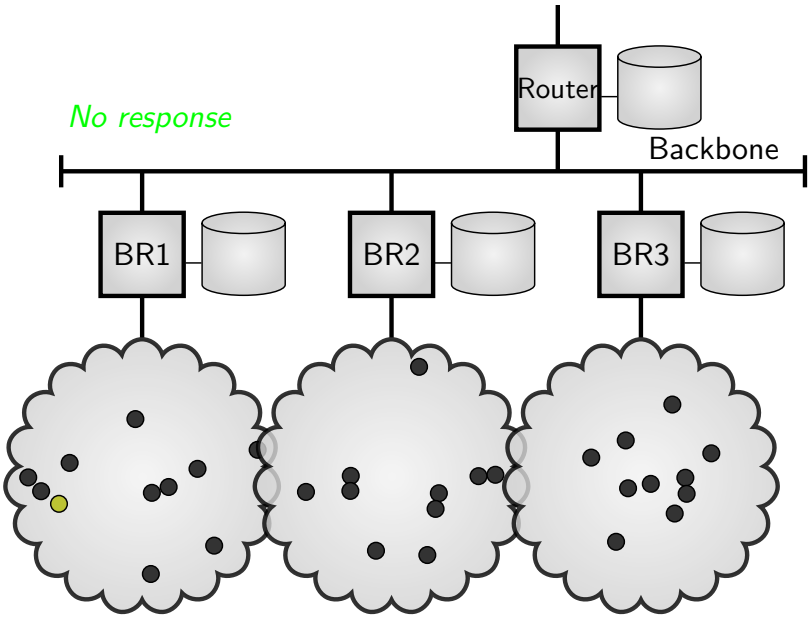
Backbone router



Backbone router

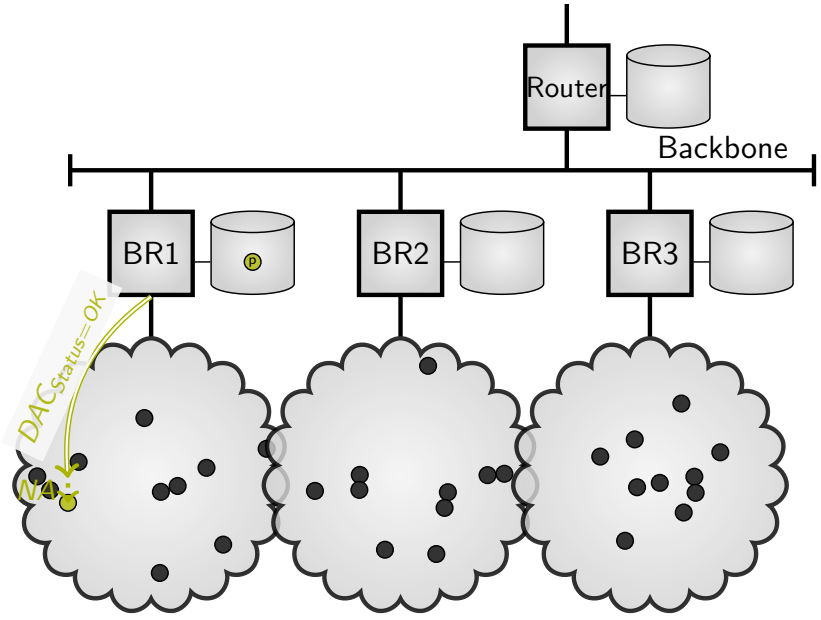


Backbone router



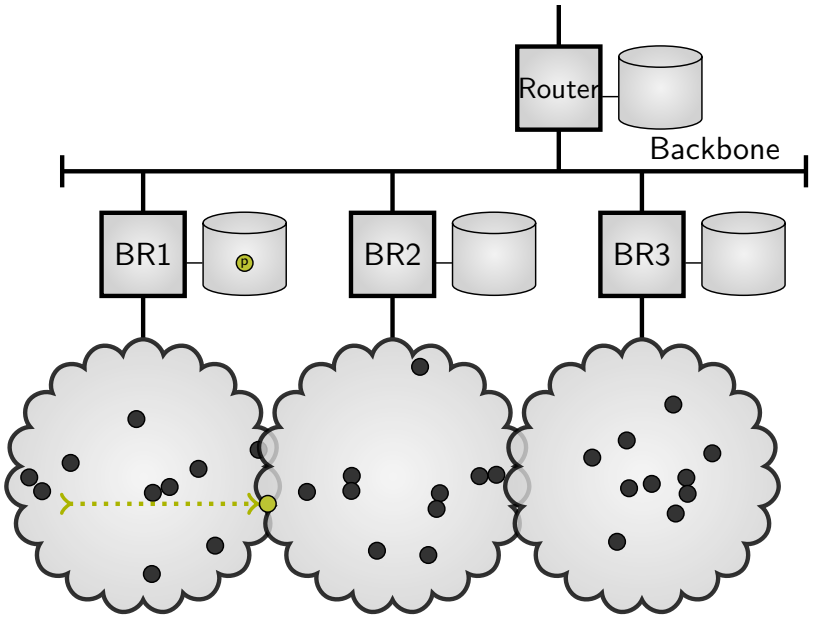
Backbone router

IETF Working Groups ▶ 6LoWPAN



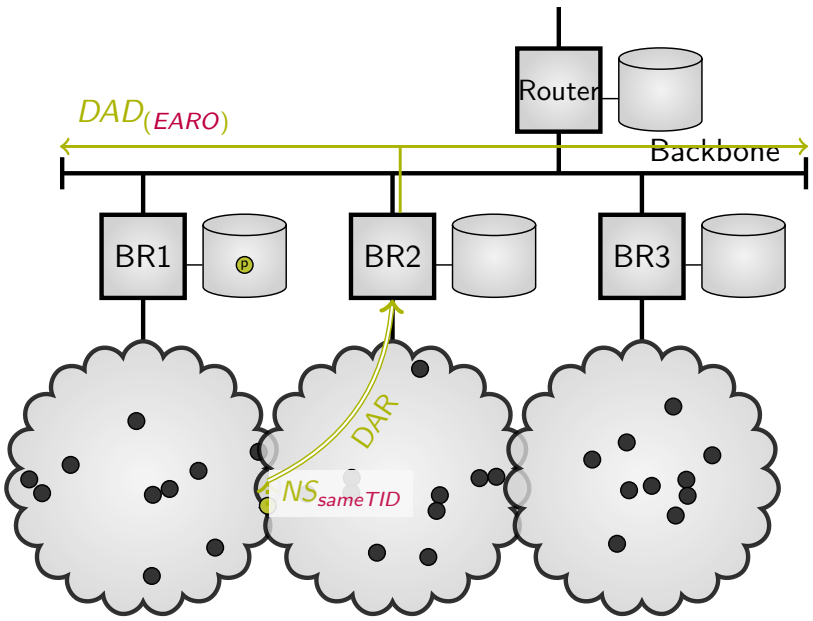
Slide 105 Laurent Toutain Filière 2
TELECOM Bretagne

Backbone router

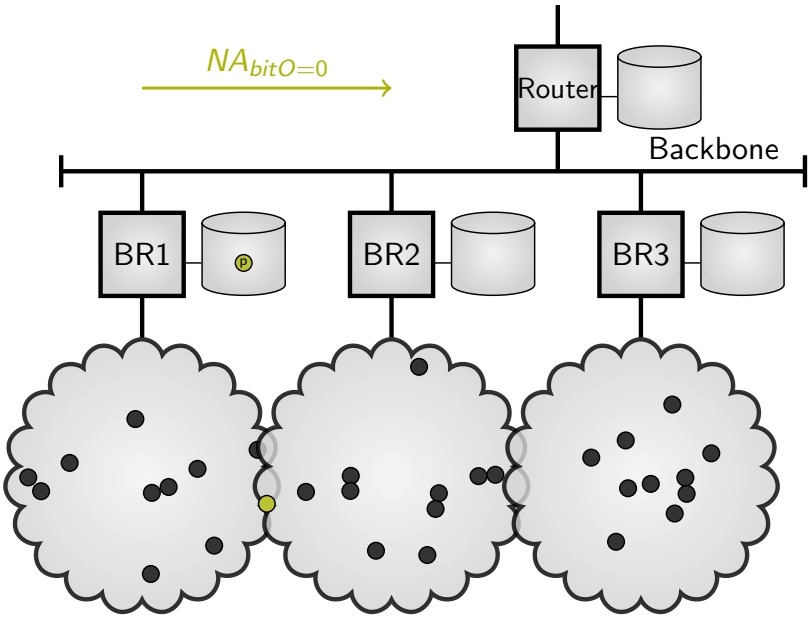


Backbone router

IETF Working Groups ▶ 6LoWPAN

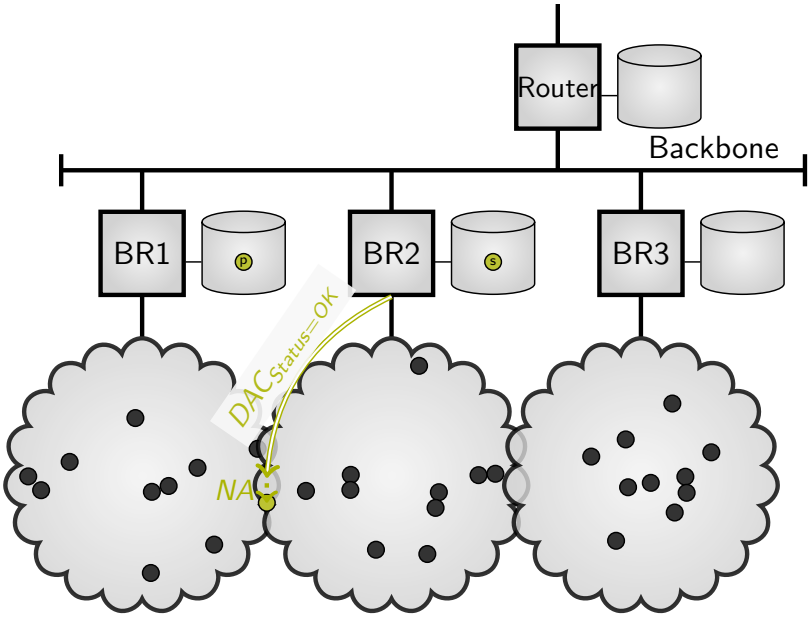


Backbone router

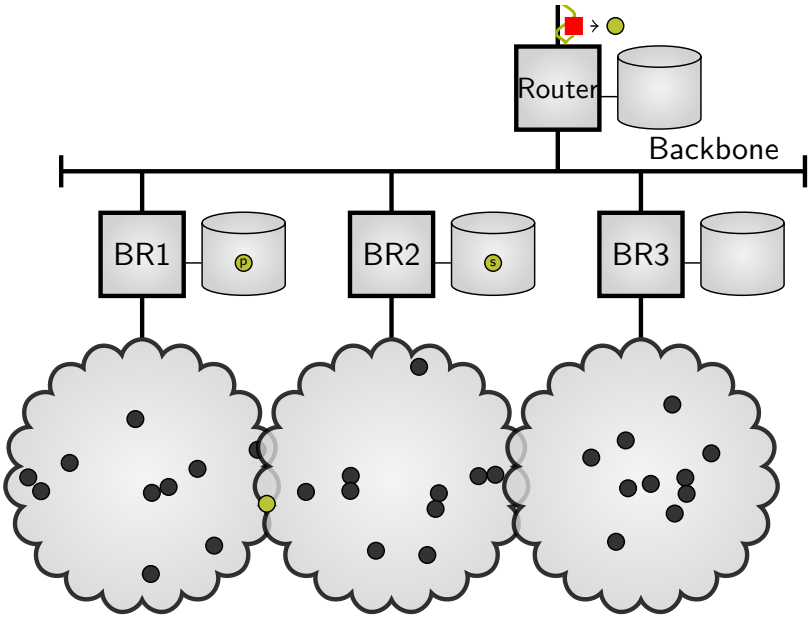


Backbone router

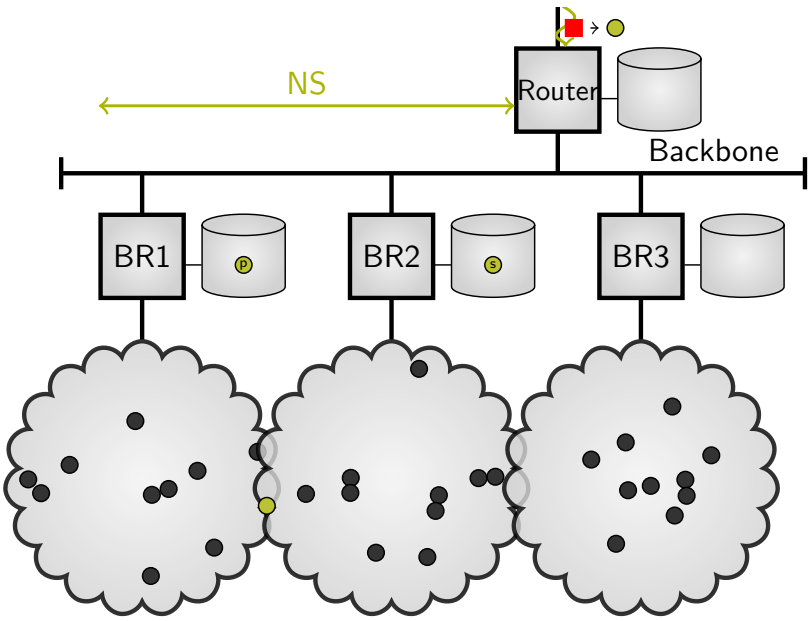
IETF Working Groups ▶ 6LoWPAN



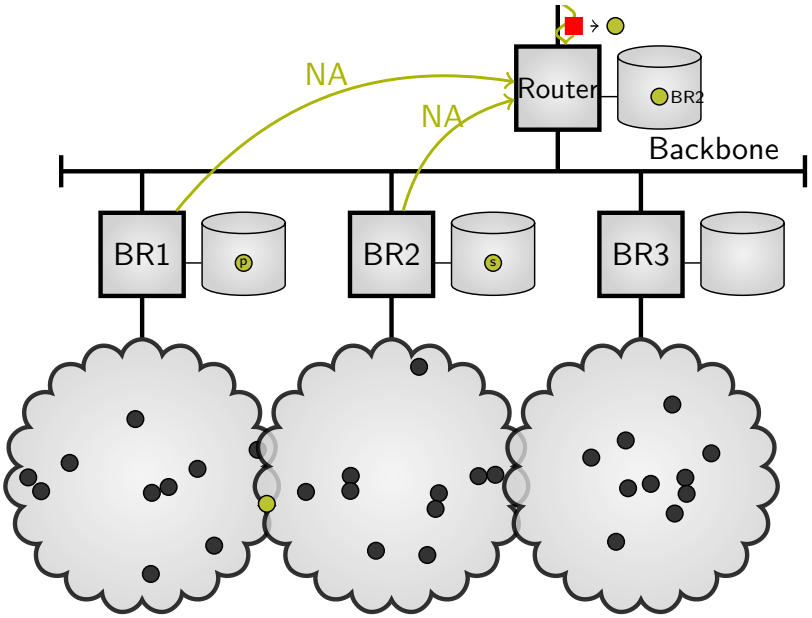
Backbone router



Backbone router

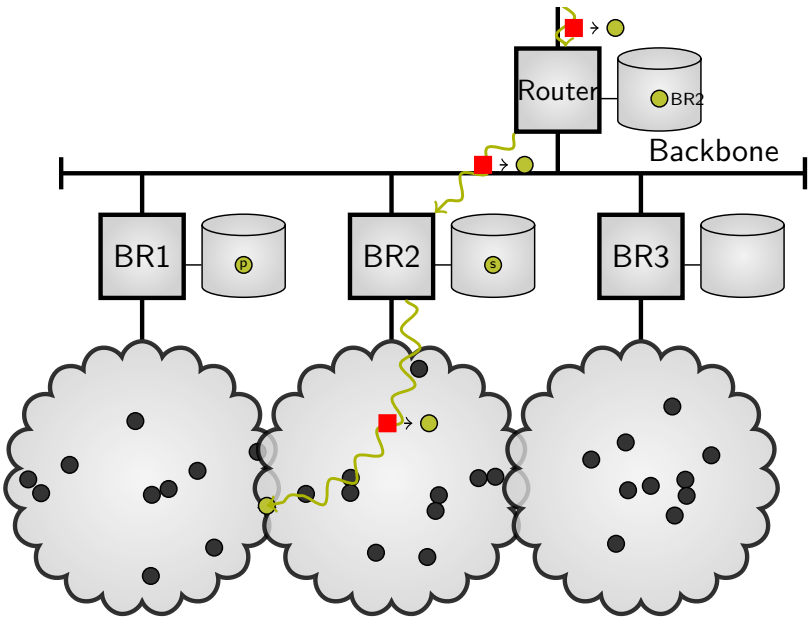


Backbone router

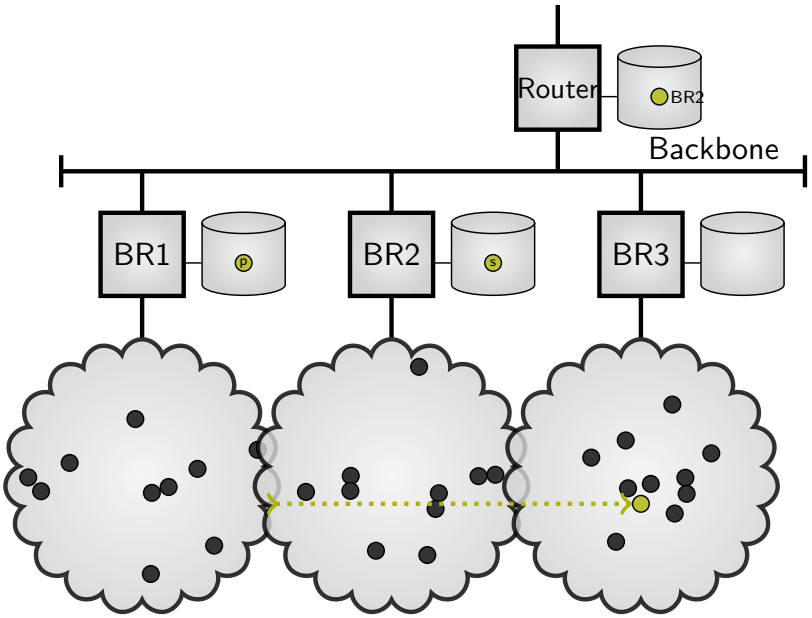


Backbone router

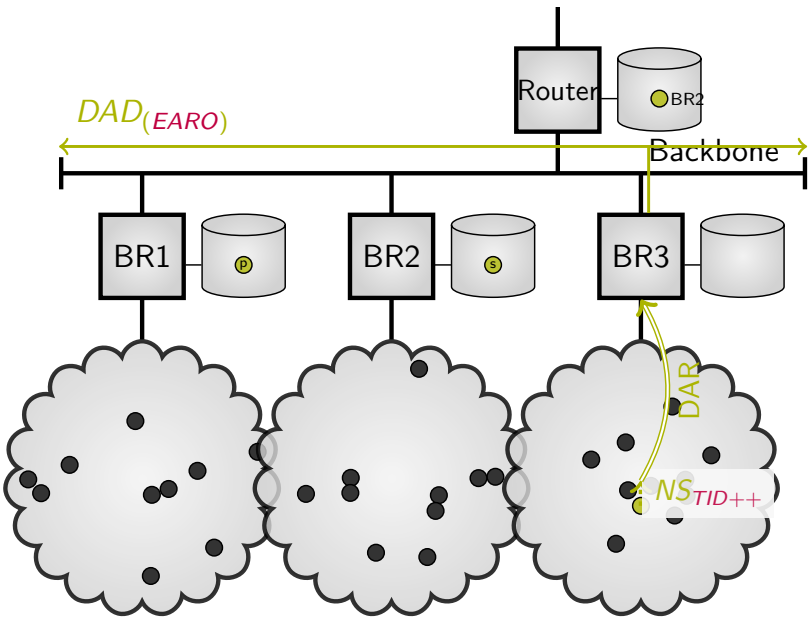
IETF Working Groups ▶ 6LoWPAN



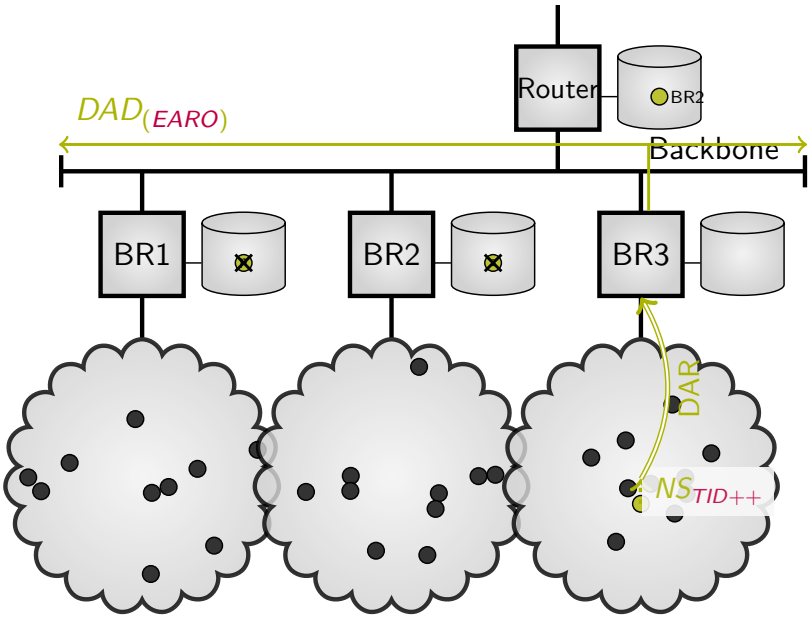
Backbone router



Backbone router

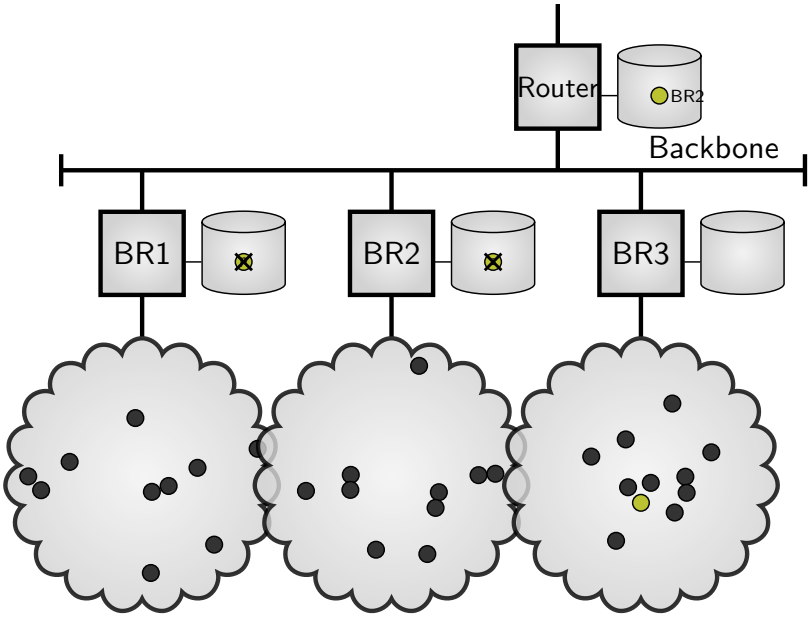


Backbone router



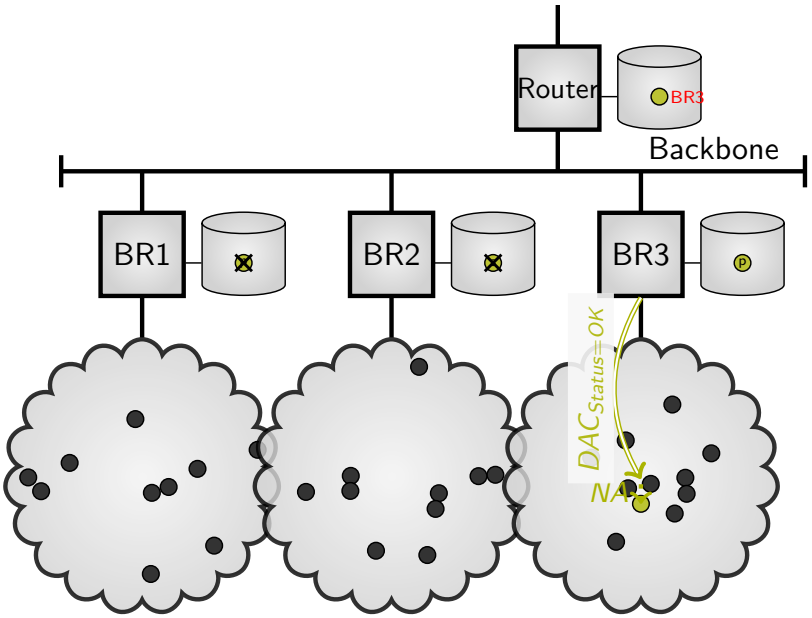
Backbone router

IETF Working Groups ▶ 6LoWPAN



Slide 105 Laurent Toutain Filière 2
TELECOM Bretagne

Backbone router



LOAD and LOADng

Pro-active protocol: LOAD

- G.9903 (Annex H) http://www.itu.int/rec/T-REC-G.9903-201210-I
  - Narrowband orthogonal frequency division multiplexing power line communication transceivers for G3-PLC networks.
- LOAD works for mesh-under (MAC-16 and MAC-64),
- LOADng includes route-over (IPv6) and IPv4 .
- Messages:
  - Route REQuest : flooded to neighbors looking to a destination.
  - Route REPly: P2P response triggered when a RREQ reach the destination.
  - RREP-ACK: optional to LOADng to be sure that the RREP has been received.
  - RERR: when impossible to forward data toward destination

Tables

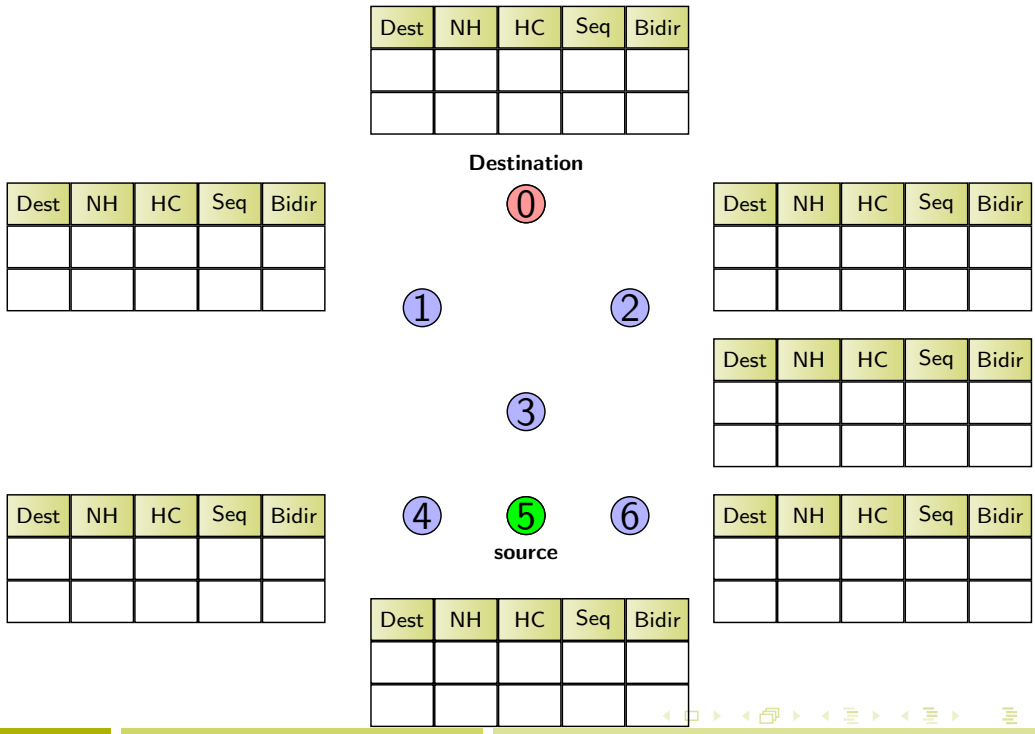
Each node maintains 4 tables:

- Routing Set:

Dest	NextHop	Metric	MetricType	HopCount	SeqNum	Bidir.	Iface	ValidTime

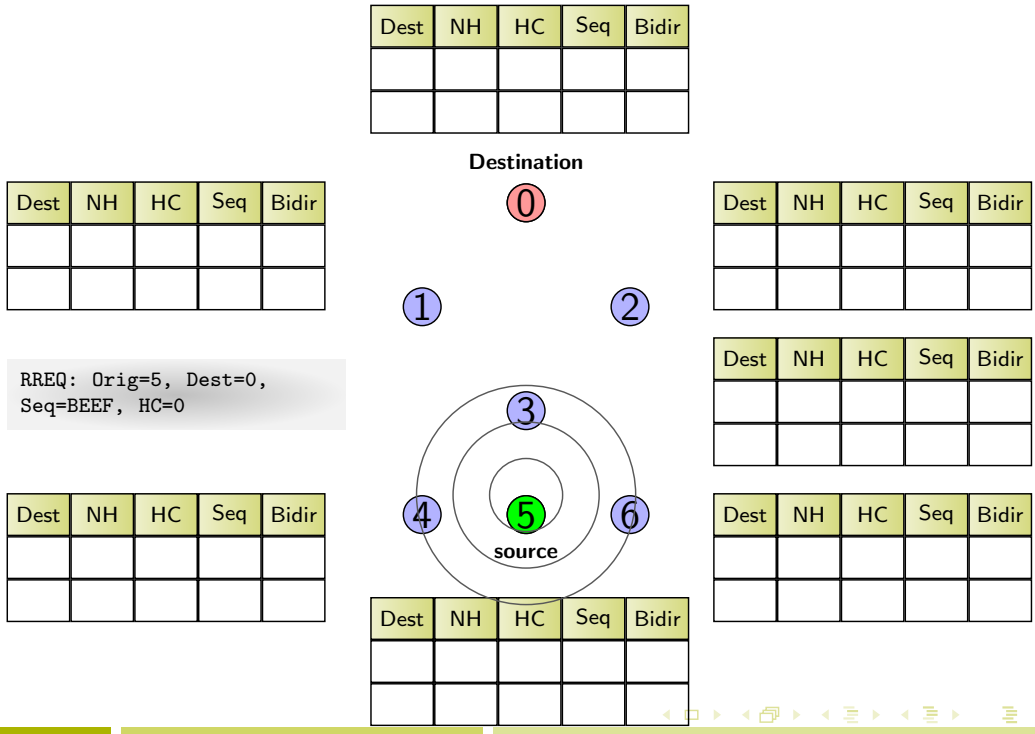
- Blacklisted Set: Neighbors with unidirectional connectivity.
- Local Interface set: addresses associated to node's interfaces.
- DestinationSet: addresses the node is responsible (ie non LOAD nodes)
- Pending ack set: if REEP-ACK is required.

Example : Simple AODV (LOAD)

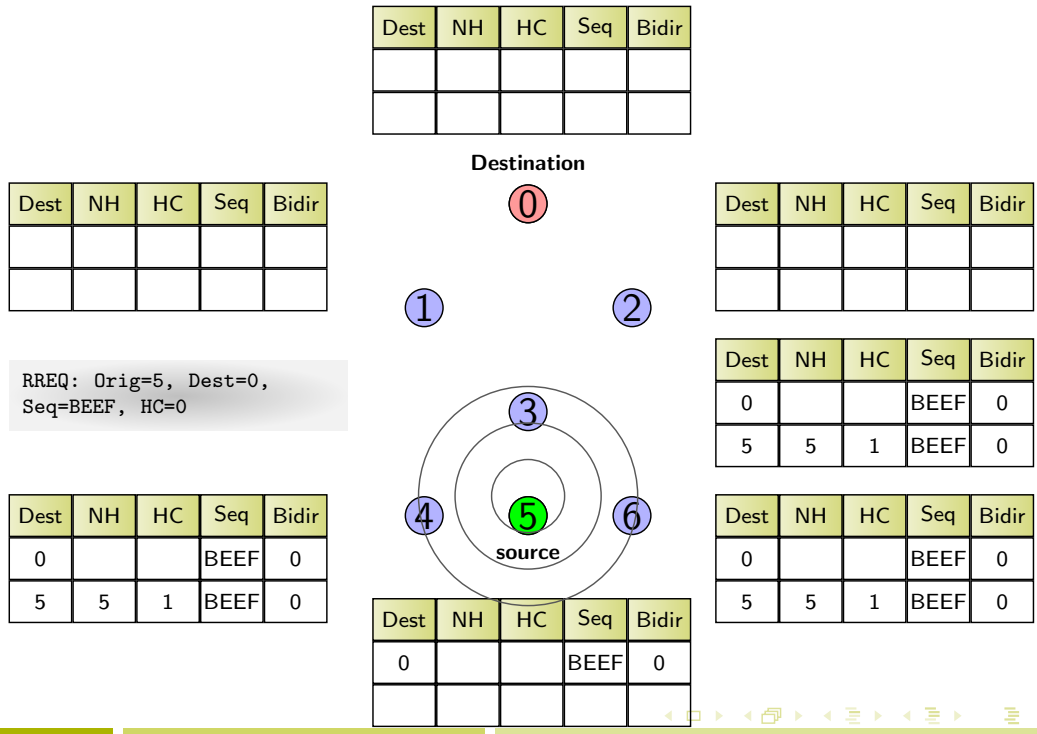


Example : Simple AODV (LOAD)

Routing Protocols

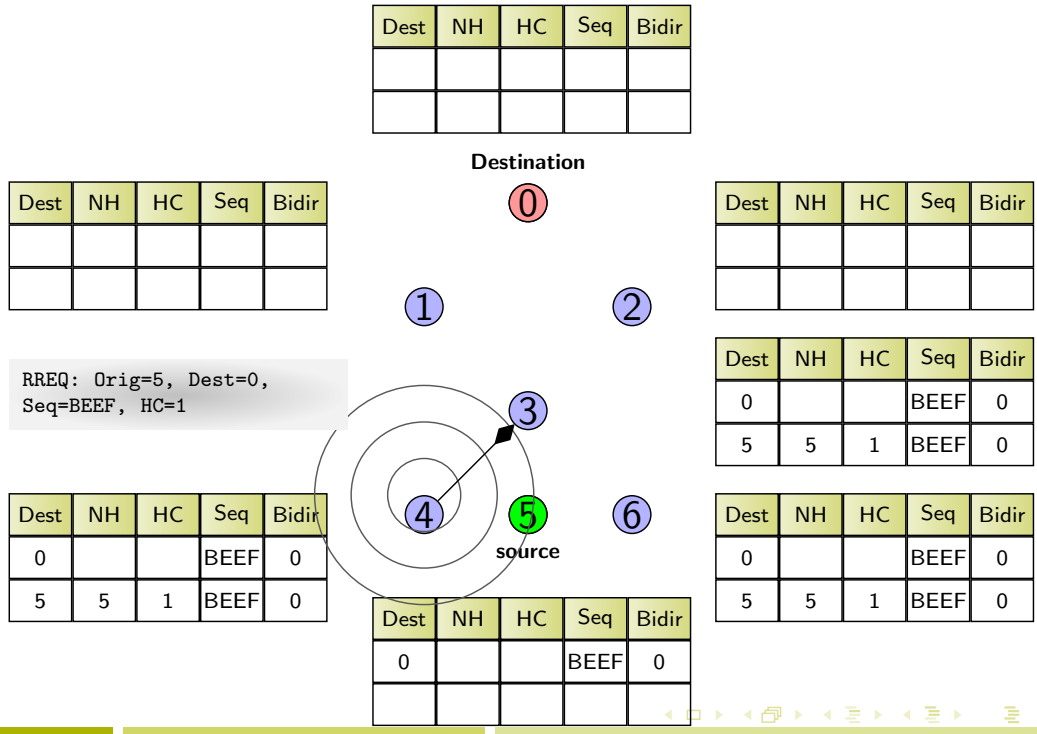


Example : Simple AODV (LOAD)

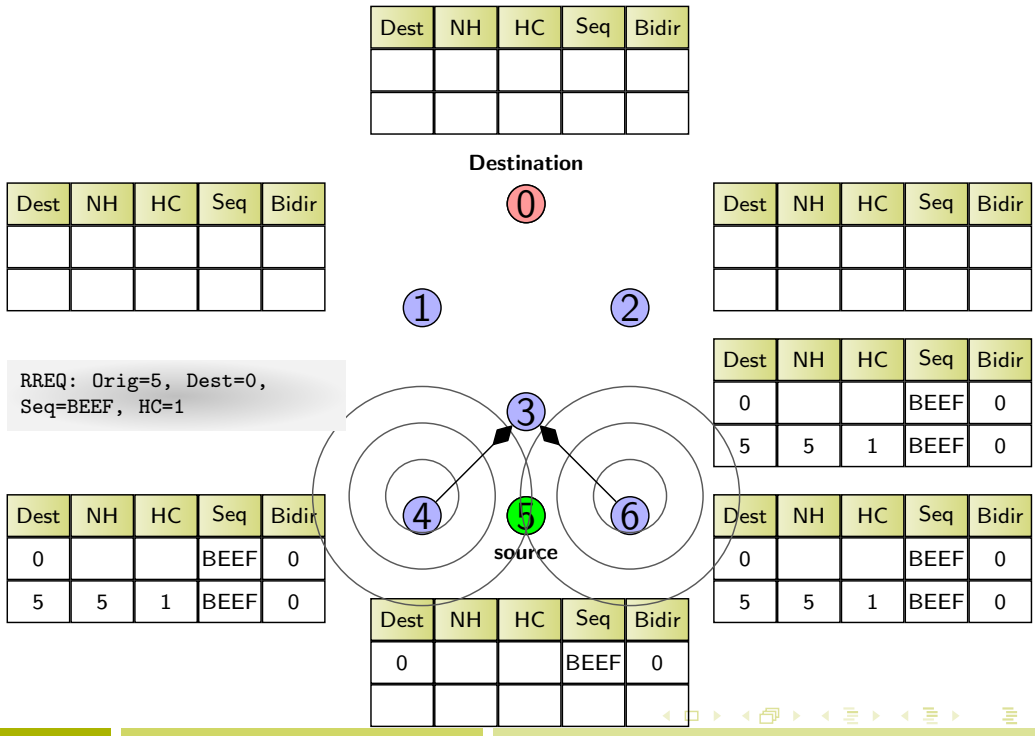


Example : Simple AODV (LOAD)

Routing Protocols

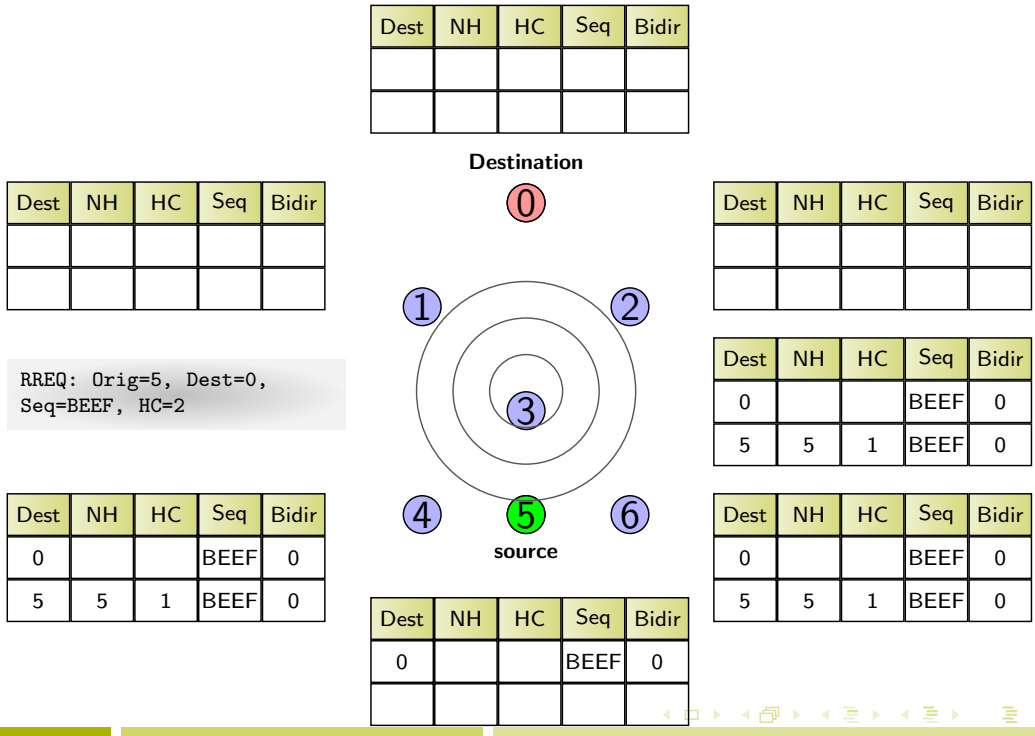


Example : Simple AODV (LOAD)

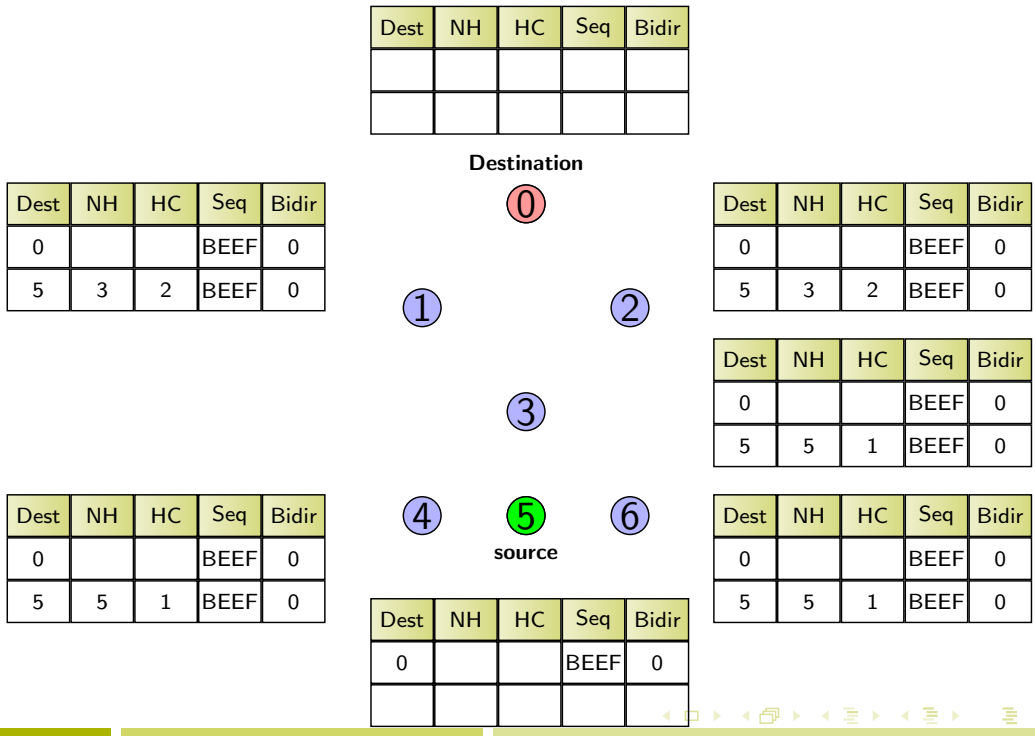


Example : Simple AODV (LOAD)

Routing Protocols

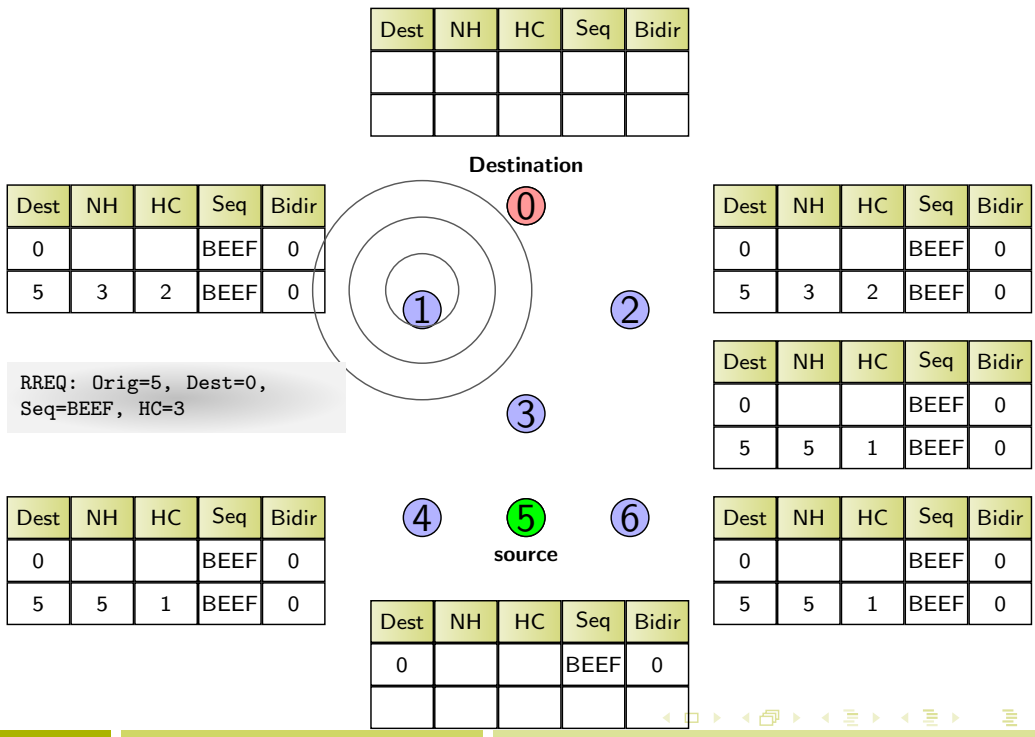


Example : Simple AODV (LOAD)

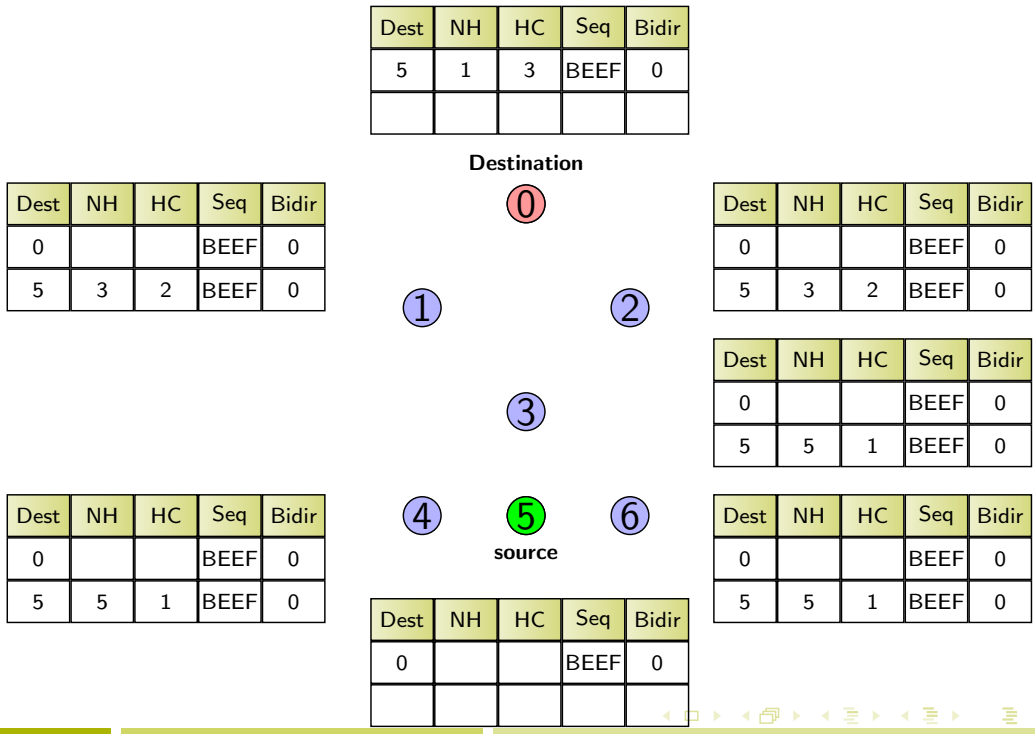


Example : Simple AODV (LOAD)

Routing Protocols

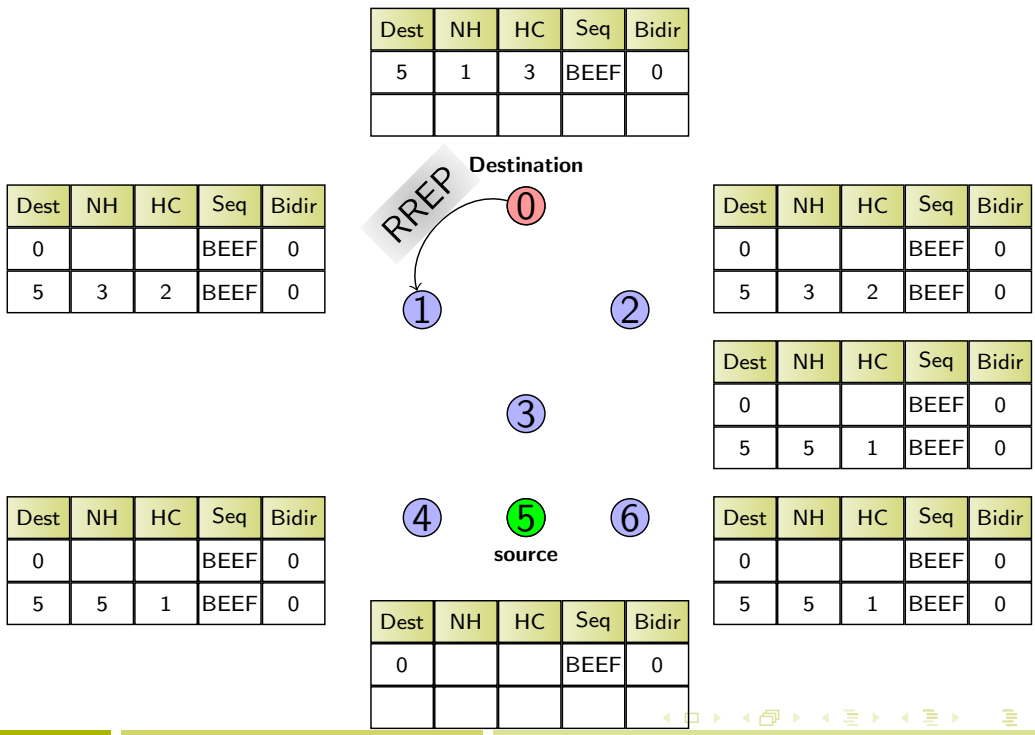


Example : Simple AODV (LOAD)



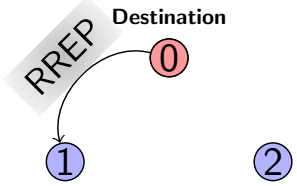
Example : Simple AODV (LOAD)

Routing Protocols



Example : Simple AODV (LOAD)

Dest	NH	HC	Seq	Bidir
5	1	3	BEEF	0



Dest	NH	HC	Seq	Bidir
0	0	1	BEEF	0
5	3	2	BEEF	0

Dest	NH	HC	Seq	Bidir
0			BEEF	0
5	3	2	BEEF	0

Dest	NH	HC	Seq	Bidir
0			BEEF	0
5	5	1	BEEF	0

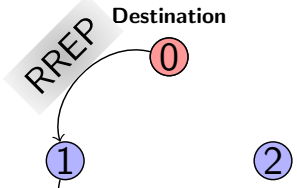
Dest	NH	HC	Seq	Bidir
0			BEEF	0
5	5	1	BEEF	0

Dest	NH	HC	Seq	Bidir
0			BEEF	0

Example : Simple AODV (LOAD)

Routing Protocols

Dest	NH	HC	Seq	Bidir
5	1	3	BEEF	0



Dest	NH	HC	Seq	Bidir
0	0	1	BEEF	0
5	3	2	BEEF	0

Dest	NH	HC	Seq	Bidir
0			BEEF	0
5	3	2	BEEF	0

Dest	NH	HC	Seq	Bidir
0	1	2	BEEF	0
5	5	1	BEEF	0

Dest	NH	HC	Seq	Bidir
0			BEEF	0
5	5	1	BEEF	0

Dest	NH	HC	Seq	Bidir
0			BEEF	0
5	5	1	BEEF	0

Dest	NH	HC	Seq	Bidir
0	3	3	BEEF	0

Example : Simple AODV (LOAD)

Dest	NH	HC	Seq	Bidir
5	1	3	BEEF	0

Dest	NH	HC	Seq	Bidir
0	0	1	BEEF	0
5	3	2	BEEF	0

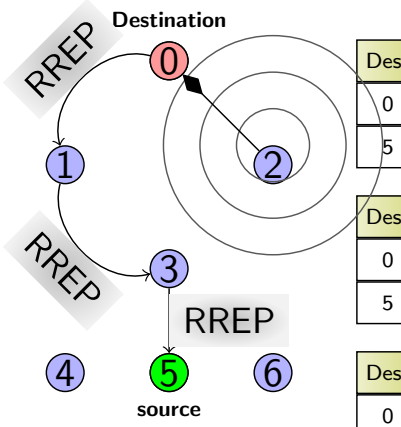
Dest	NH	HC	Seq	Bidir
0			BEEF	0
5	3	2	BEEF	0

Dest	NH	HC	Seq	Bidir
0	1	2	BEEF	0
5	5	1	BEEF	0

Dest	NH	HC	Seq	Bidir
0			BEEF	0
5	5	1	BEEF	0

Dest	NH	HC	Seq	Bidir
0			BEEF	0
5	5	1	BEEF	0

Dest	NH	HC	Seq	Bidir
0	3	3	BEEF	0



Example : Simple AODV (LOAD)

Routing Protocols

Dest	NH	HC	Seq	Bidir
5	1	3	BEEF	0

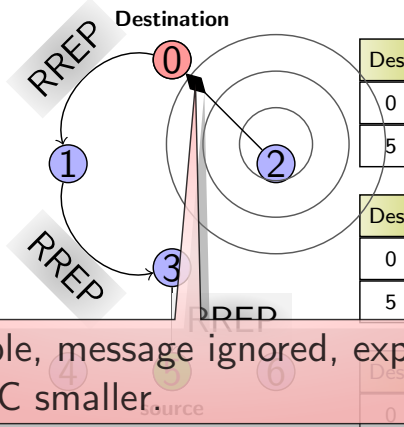
Dest	NH	HC	Seq	Bidir
0	0	1	BEEF	0
5	3	2	BEEF	0

Dest	NH	HC	Seq	Bidir
0			BEEF	0
5	3	2	BEEF	0

Dest	NH	HC	Seq	Bidir
0	1	2	BEEF	0
5	5	1	BEEF	0

Dest	NH	HC	Seq	Bidir
0			BEEF	0
5	5	1	BEEF	0

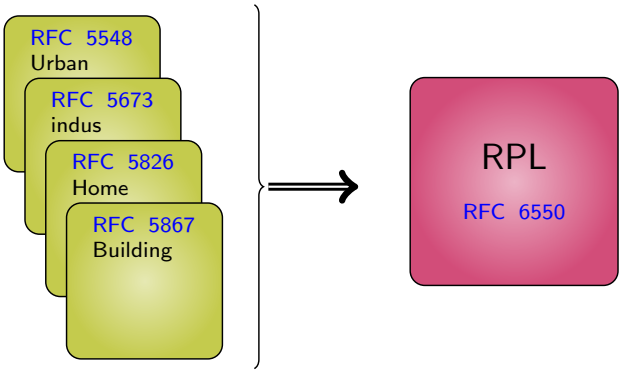
Dest	NH	HC	Seq	Bidir
0	3	3	BEEF	0



entry is in table, message ignored, expect if QoS is better or HC smaller.

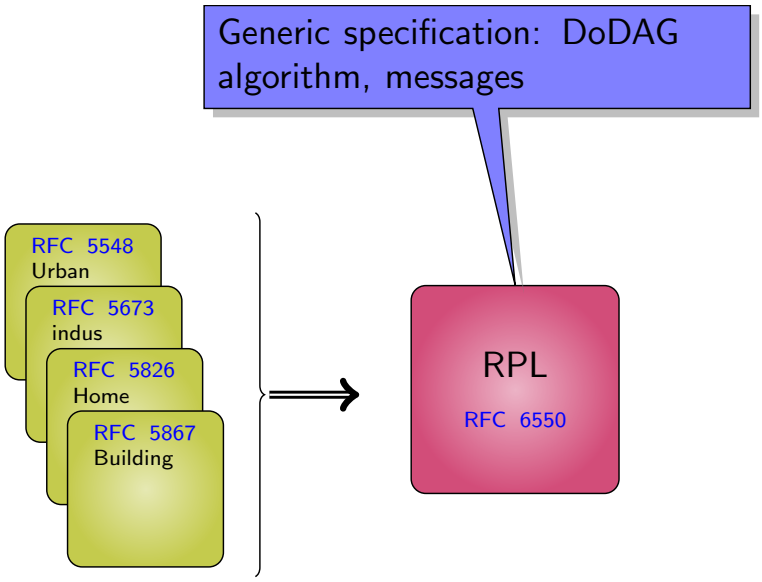
Slide 109 Laurent Toutain Filière 2

RPL Ecosystem



Thanks to Dominique Barthel (Orange)

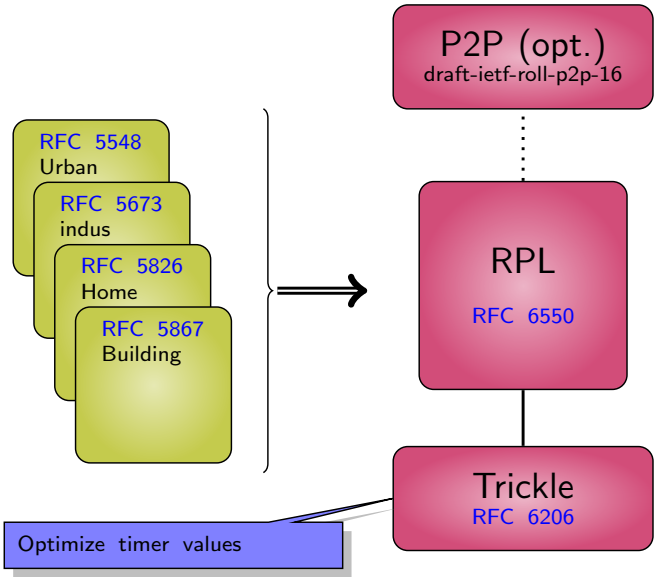
RPL Ecosystem



Thanks to Dominique Barthel (Orange)

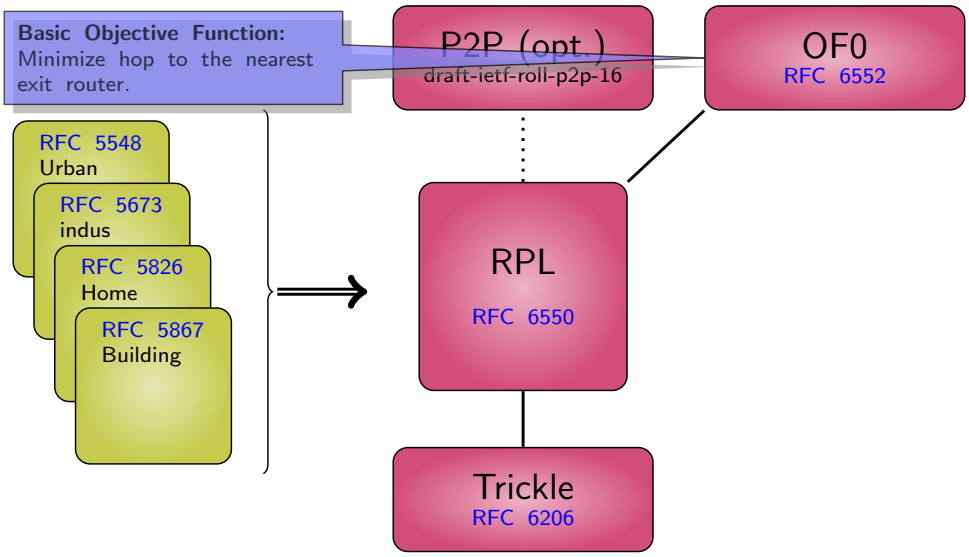
RPL Ecosystem

RPL



Thanks to Dominique Barthel (Orange)

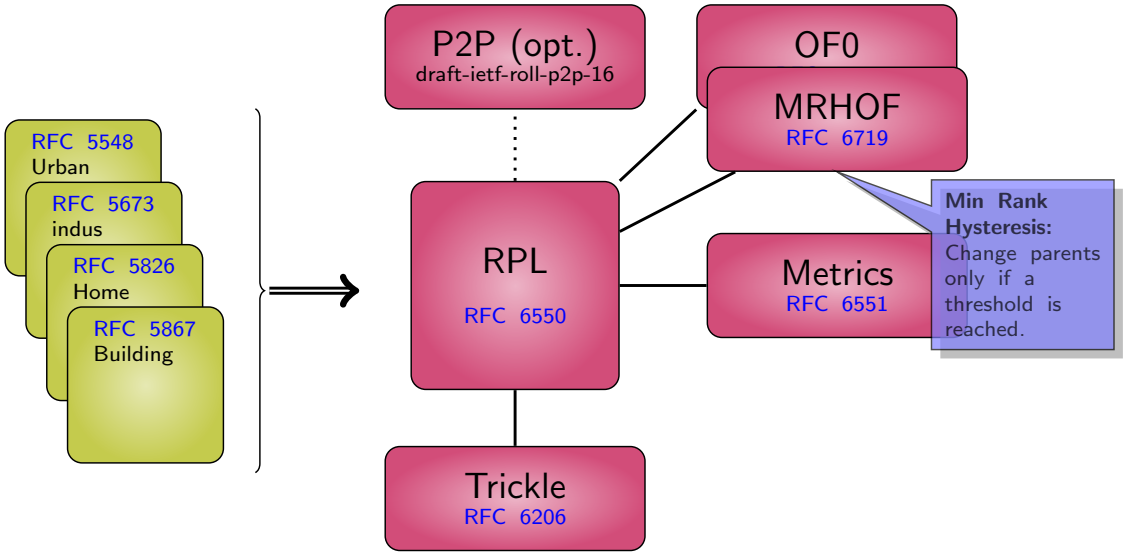
RPL Ecosystem



Thanks to Dominique Barthel (Orange)

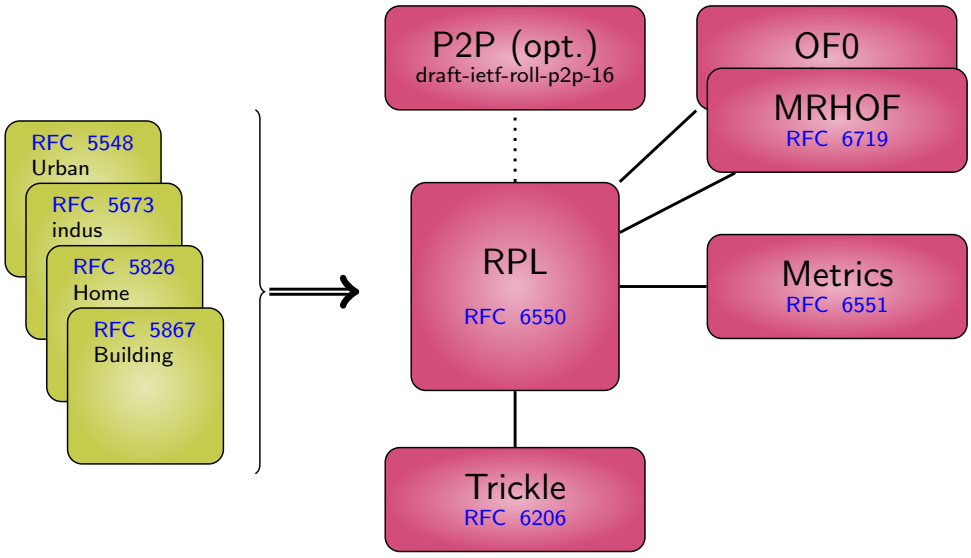
RPL

RPL Ecosystem



Thanks to Dominique Barthel (Orange)

RPL Ecosystem



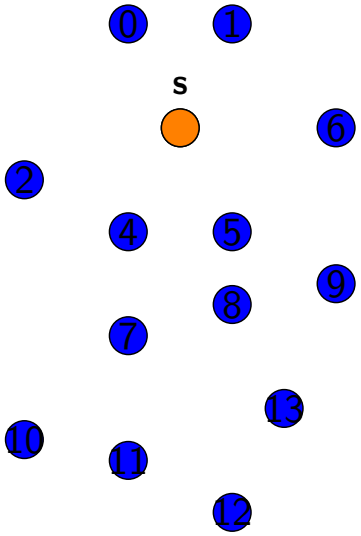
Thanks to Dominique Barthel (Orange)

RPL: Generic protocol RFC 6550

- Routing Protocol on Lossy Links
- Based on a Directed Acyclic Graph:
  - Based on Distance Vector
    - Simple to implement
    - Announcements are limited in stable network with trickle algorithm
    - Designed to be robust (several paths) and reduced convergence time
- Two kind of traffic
  - P2MP: 6LBR to 6LN
  - MP2P: 6LN to 6LBR
  - P2P is under study
- Each node can be a router and forward packets
- Some nodes can only be leave:
  - Register their address
  - Do not participate to routing announcement

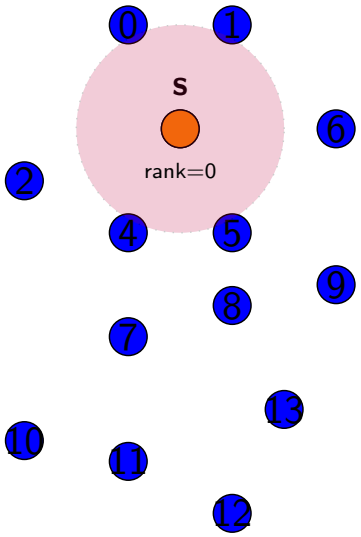
RPL - DIO

RPL



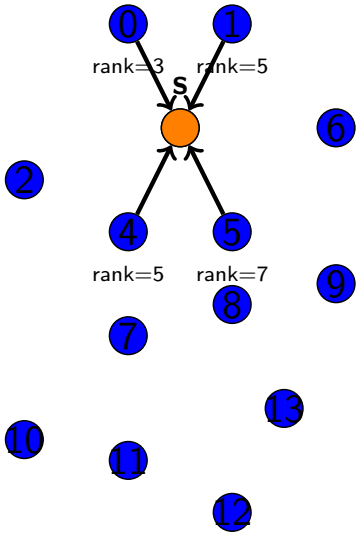
RPL - DIO

RPL



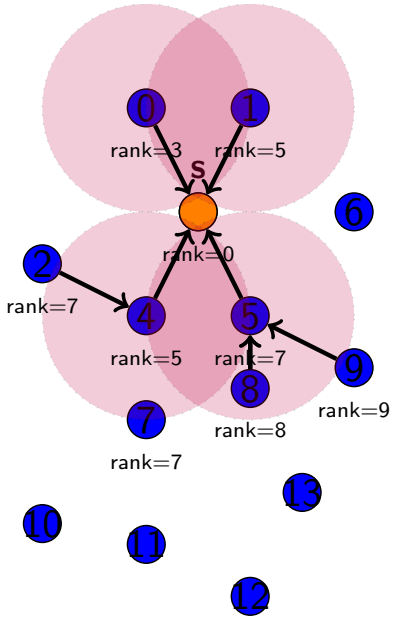
RPL - DIO

RPL



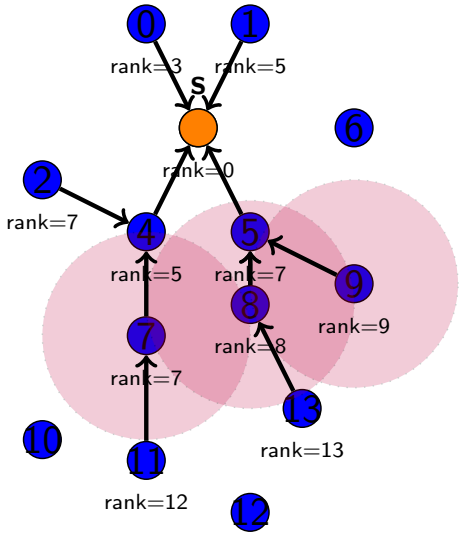
RPL - DIO

RPL



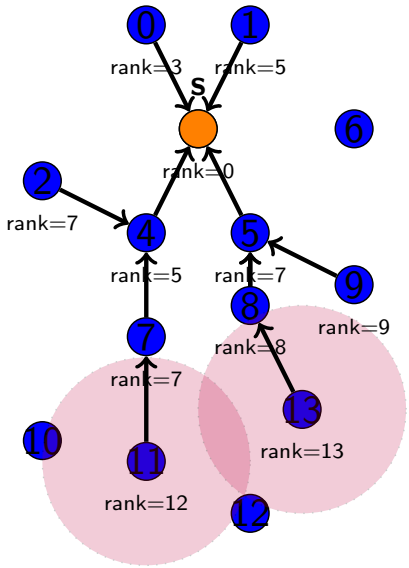
RPL - DIO

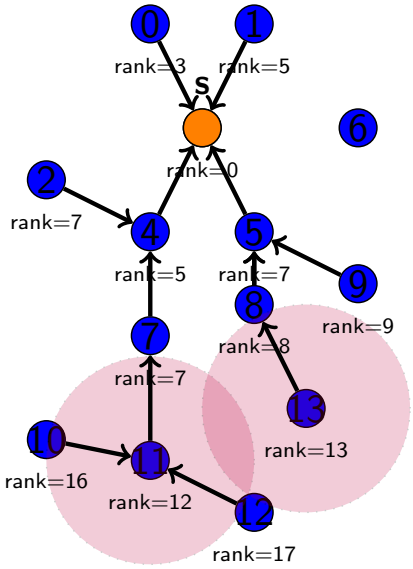
RPL



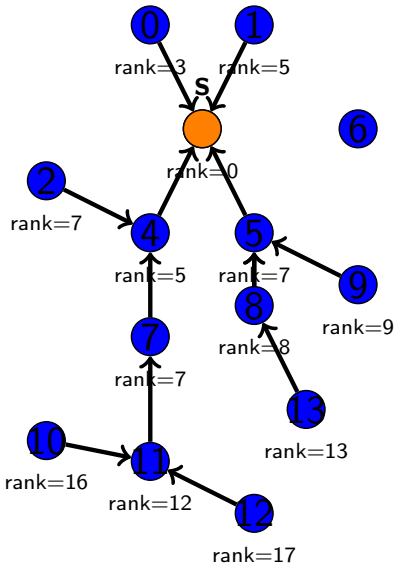
RPL - DIO

RPL

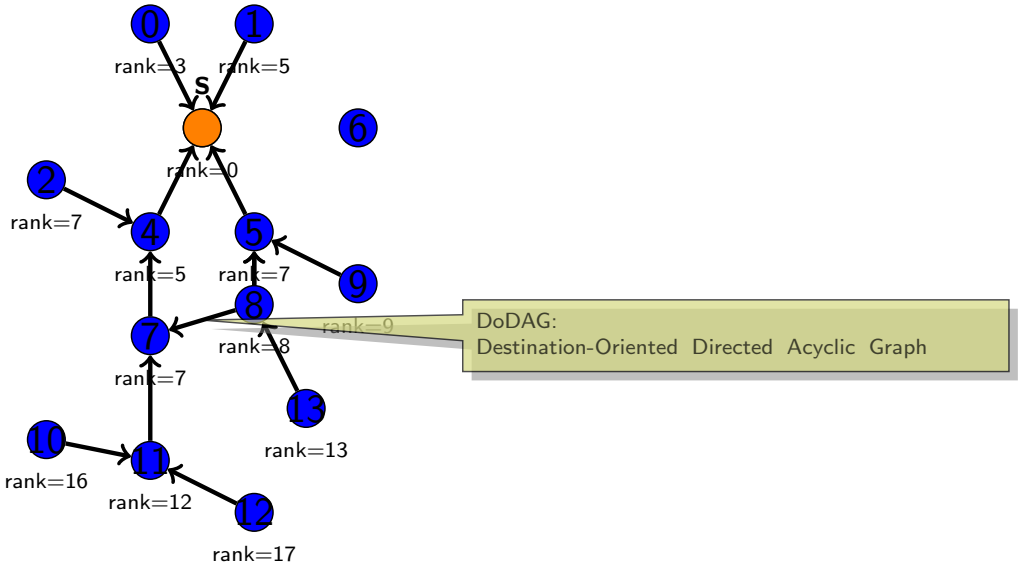



0
1
rank=3
rank=5
S
6
rank=0
2
rank=7
4
5
rank=5
rank=7
9
8
rank=9
7
rank=8
rank=7
13
10
rank=13
11
rank=16
rank=12
12
rank=17

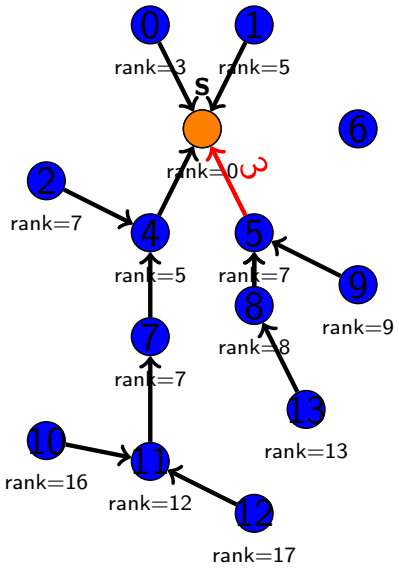
RPL - DIO
RPL


0
1
rank=3
rank=5
S
6
rank=0
2
rank=7
4
5
rank=5
rank=7
9
8
rank=9
7
rank=8
rank=7
13
10
rank=13
11
rank=16
rank=12
12
rank=17

RPL - DIO

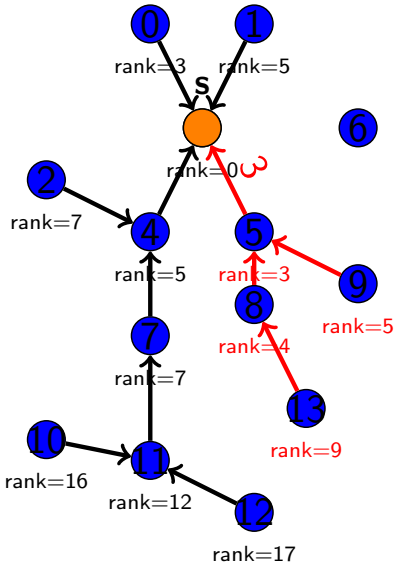


RPL - DIO



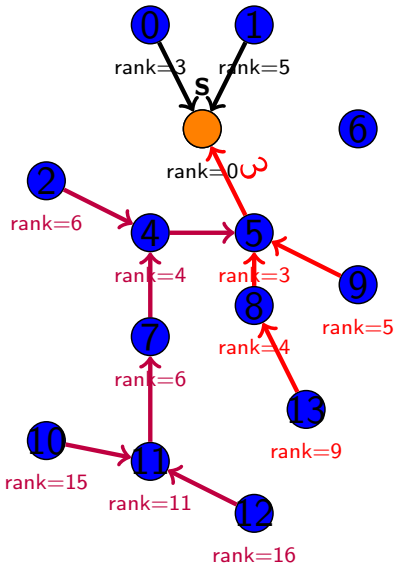
RPL - DIO

RPL

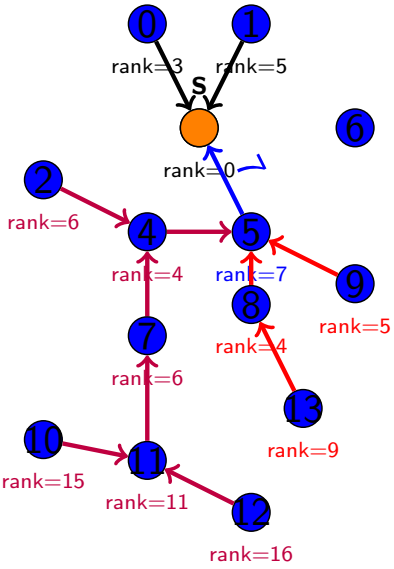


RPL - DIO

RPL

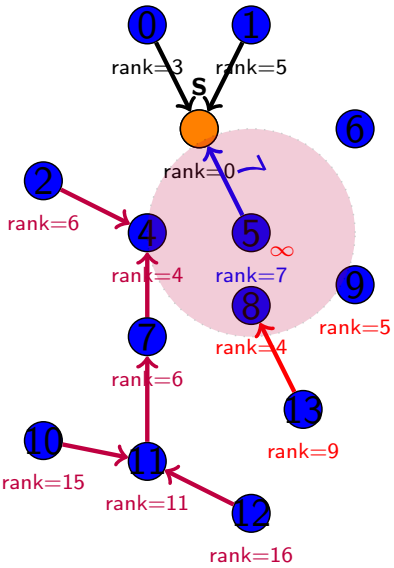


RPL - DIO



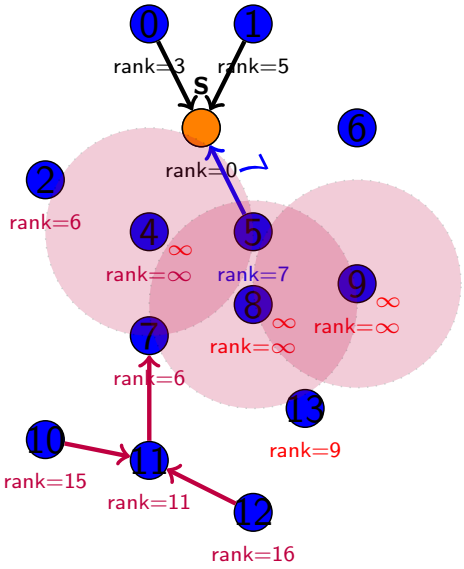
RPL - DIO

RPL



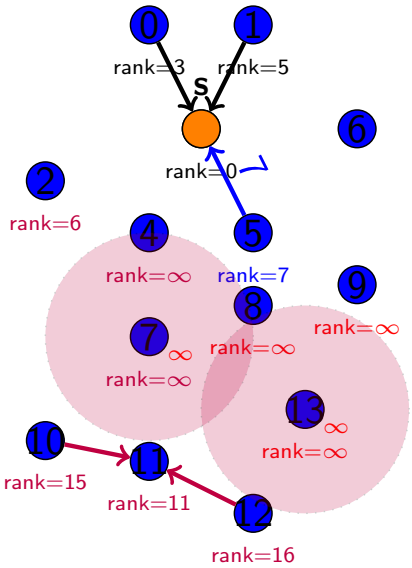
RPL - DIO

RPL



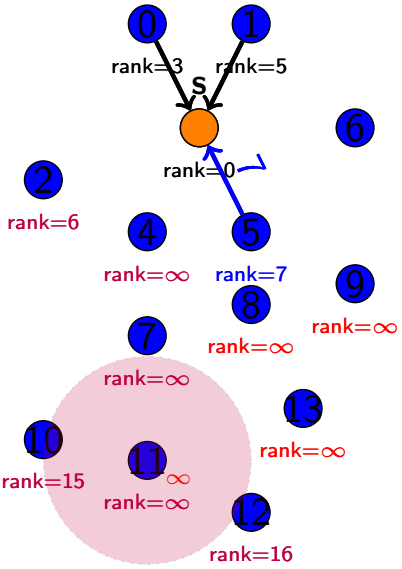
RPL - DIO

RPL



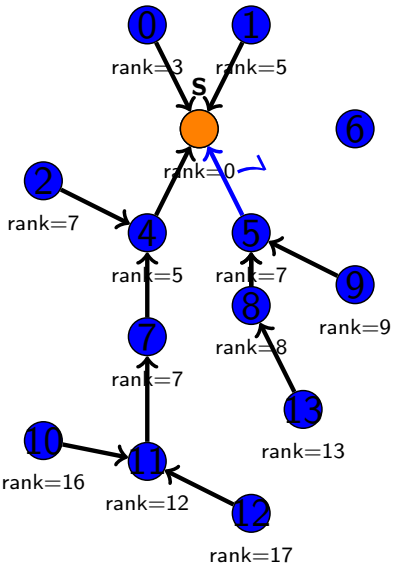
RPL - DIO

RPL



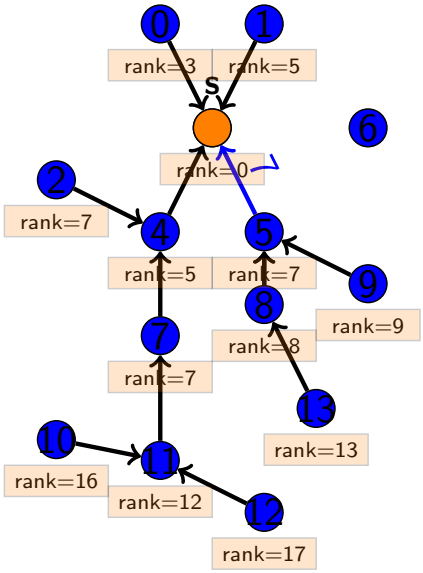
RPL - DIO

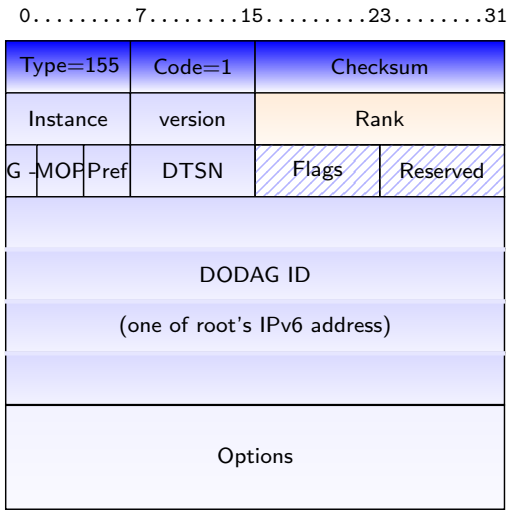
RPL



Slide 113 Laurent Toutain Filière 2

RPL - DIO

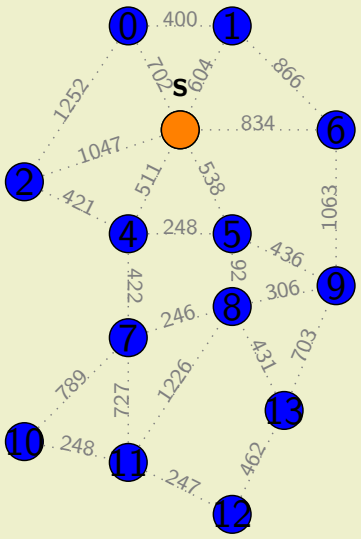




Rank

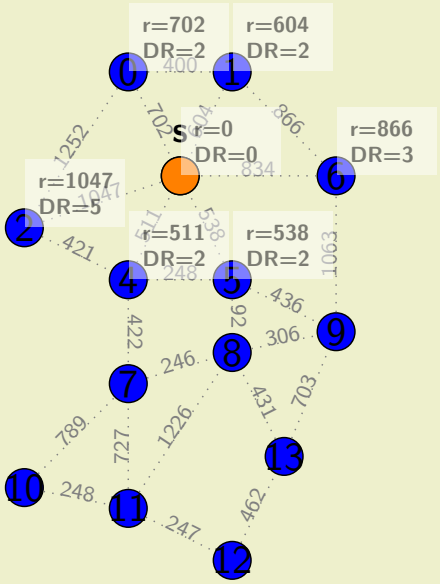
- Rank: 16 bits
- *MinHopRankIncrease*: is used to add some precision
  - by default: $MinHopRankIncrease = 256$
- $DAGRank(rank) = \left\lfloor \frac{rank}{MinHopRankIncrease} \right\rfloor$
  - by default: *rank* most significant byte.
- Relations:
  - $A_{rank} < B_{rank}$ if $DAGRank(A_{rank}) < DAGRank(B_{rank})$
  - $A_{rank} = B_{rank}$ if $DAGRank(A_{rank}) = DAGRank(B_{rank})$
  - $A_{rank} > B_{rank}$ if $DAGRank(A_{rank}) > DAGRank(B_{rank})$
- Actions:
  - $DAGRank(M) < DAGRank(N)$:
    - M can be a DoDAG parent without risk of loop
  - $DAGRank(M) = DAGRank(N)$:
    - There is a risk of loop, but sibling possibilities
  - $DAGRank(M) > DAGRank(N)$:
    - High risk of loop

Example



- *MinHopRankIncrease* = 256
- Compute the DoDAG

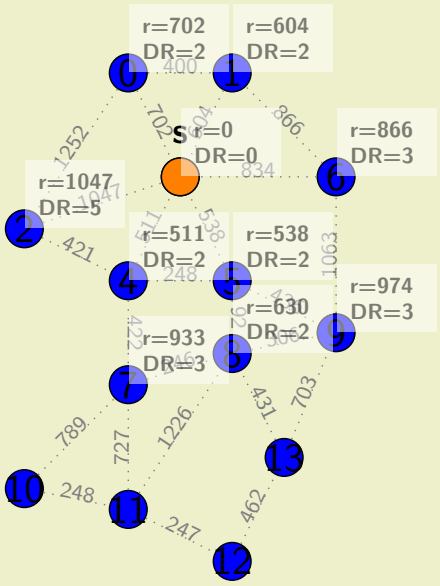
Example



- *MinHopRankIncrease* = 256
- Compute the DoDAG

Example

RPL

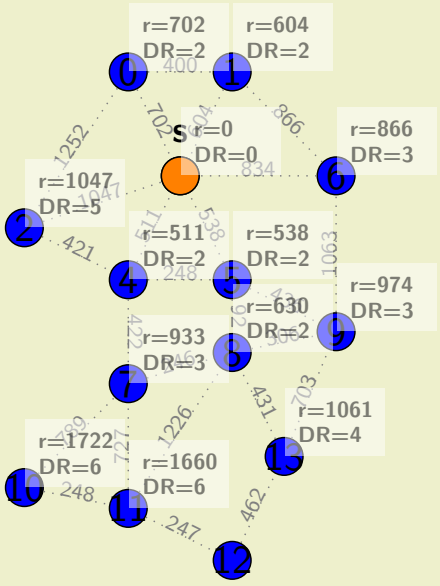


- *MinHopRankIncrease* = 256
- Compute the DoDAG

TELECOM Bretagne

Example

RPL

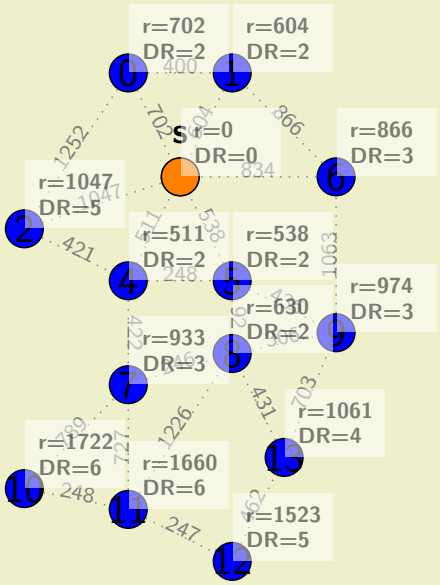


- *MinHopRankIncrease* = 256
- Compute the DoDAG

TELECOM Bretagne

Example

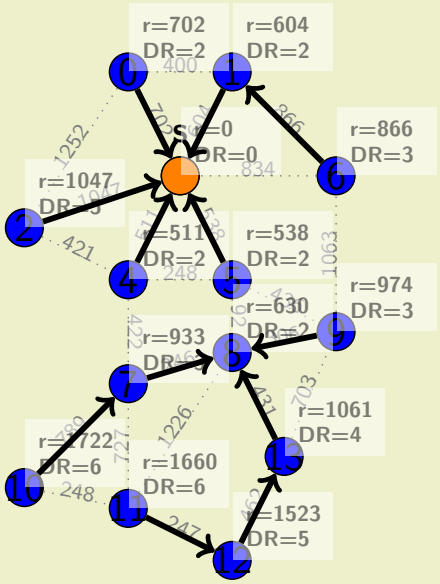
RPL



- *MinHopRankIncrease* = 256
- Compute the DoDAG

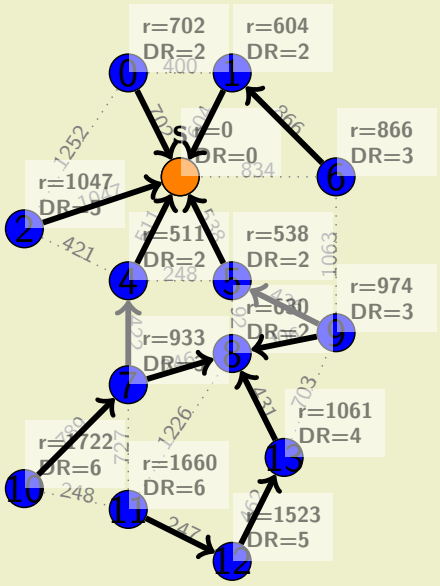
Example

RPL



- *MinHopRankIncrease* = 256
- Compute the DoDAG

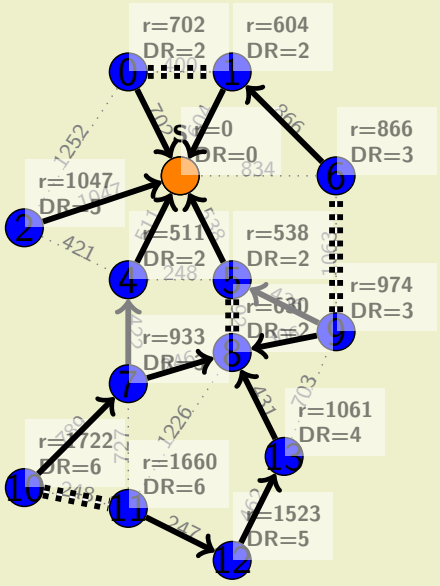
Example



- *MinHopRankIncrease* = 256
- Compute the DoDAG

Example

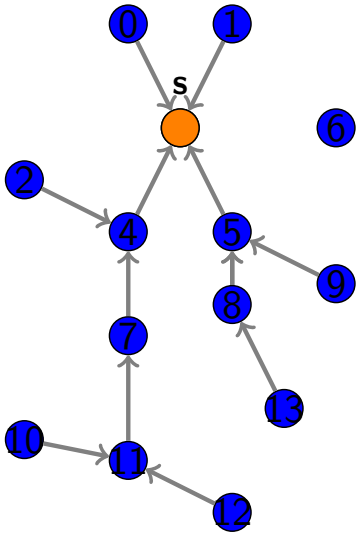
RPL



- *MinHopRankIncrease* = 256
- Compute the DoDAG

Slide 116 Laurent Toutain Filière 2

Version Number

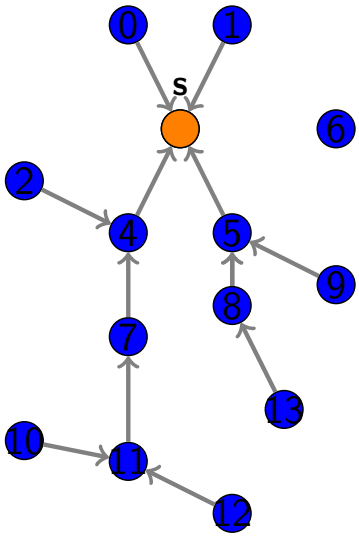


0.....7.....15.....23.....31

Type=155	Code=1	Checksum	
Instance	version=0	Rank	
G	MOP	Pref	DTSN	Flags	Reserved
DODAG ID			
(one of root's IPv6 address)			
Options			

Version Number

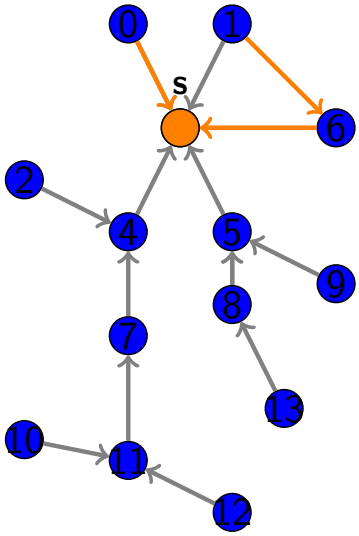
RPL



0.....7.....15.....23.....31

Type=155	Code=1	Checksum	
Instance	version=1	Rank	
G	MOP	Pref	DTSN	Flags	Reserved
DODAG ID			
(one of root's IPv6 address)			
Options			

Version Number

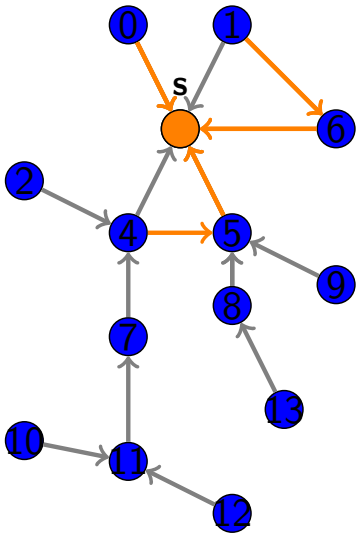


0.....7.....15.....23.....31

Type=155	Code=1	Checksum	
Instance	version=1	Rank	
G	MOP	Pref	DTSN	Flags	Reserved
DODAG ID			
(one of root's IPv6 address)			
Options			

Version Number

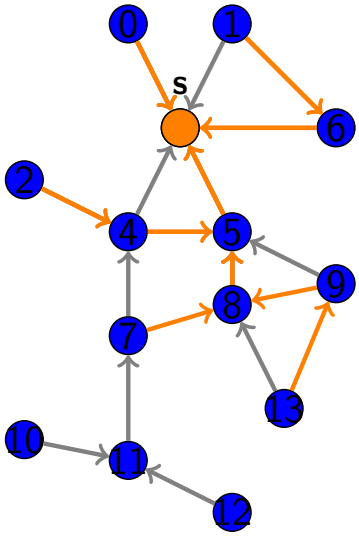
RPL



0.....7.....15.....23.....31

Type=155	Code=1	Checksum	
Instance	version=1	Rank	
G	MOP	Pref	DTSN	Flags	Reserved	
DODAG ID			
(one of root's IPv6 address)			
Options			

Version Number

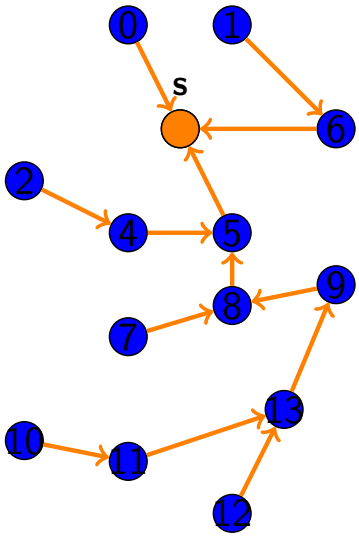


0.....7.....15.....23.....31

Type=155	Code=1	Checksum	
Instance	version=1	Rank	
G	MOP	Pref	DTSN	Flags	Reserved
DODAG ID			
(one of root's IPv6 address)			
Options			

Version Number

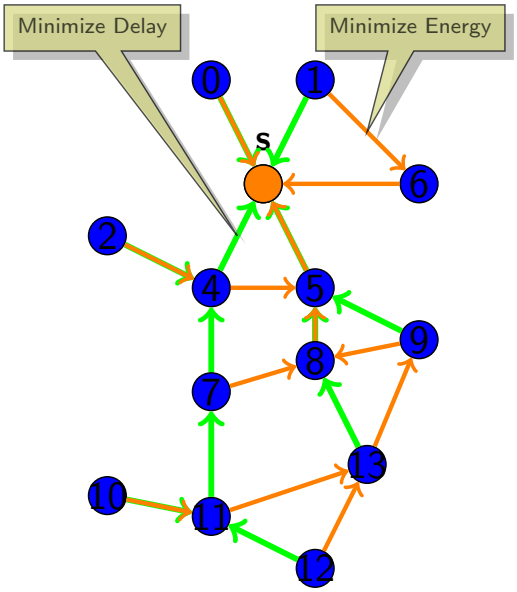
RPL



0.....7.....15.....23.....31

Type=155	Code=1	Checksum	
Instance	version=1	Rank	
G	MOP	Pref	DTSN	Flags	Reserved	
DODAG ID			
(one of root's IPv6 address)			
Options			

Instance

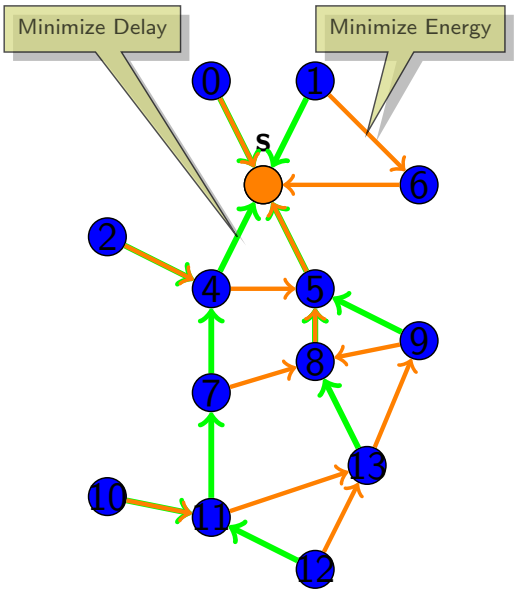


0.....7.....15.....23.....31

Type=155	Code=1	Checksum	
Instance	version	Rank	
G	MOP	Pref	DTSN	Flags	Reserved
DODAG ID			
(one of root's IPv6 address)			
Options			

Instance

RPL

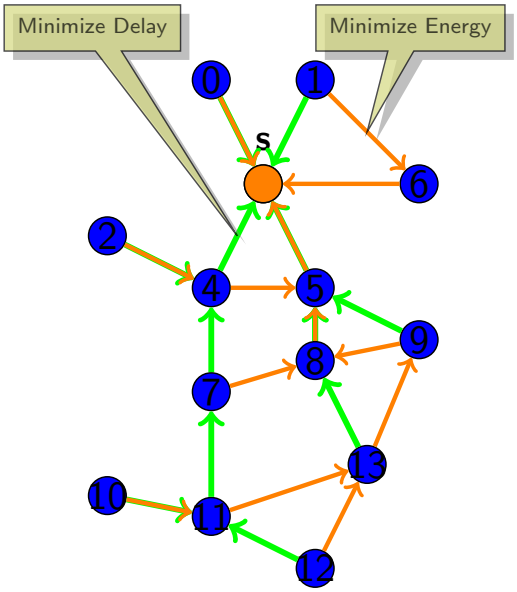


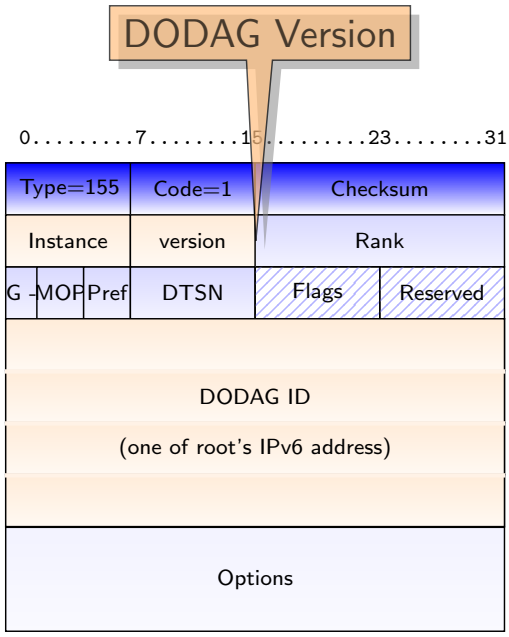
0iii iiii	**Global**: for all LLN
1Dii iiii	**Local**: for a specific root D=1: Destination

0.....7.....15.....23.....31

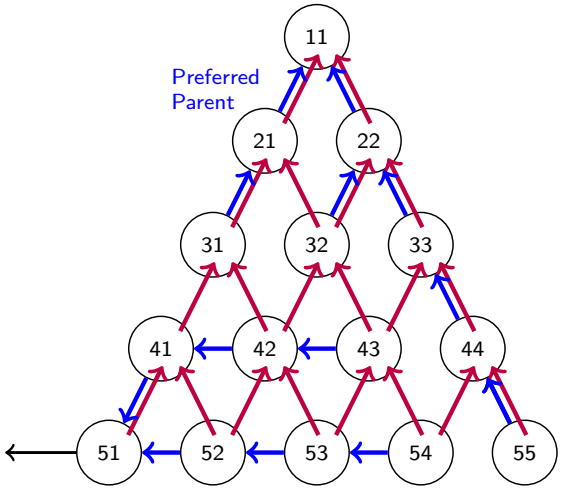
Type=155	Code=1	Checksum	
Instance	version	Rank	
G	MOP	Pref	DTSN	Flags	Reserved	
DODAG ID			
(one of root's IPv6 address)			
Options			

Instance

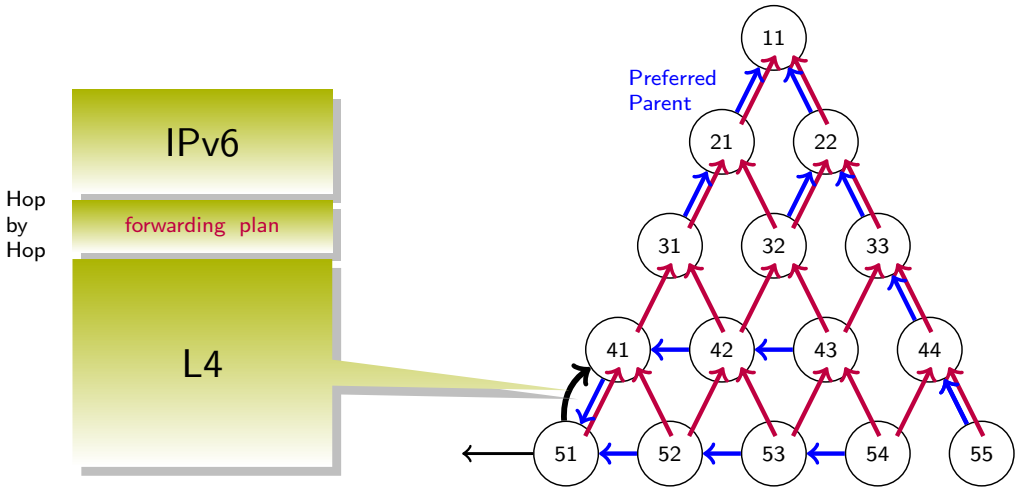




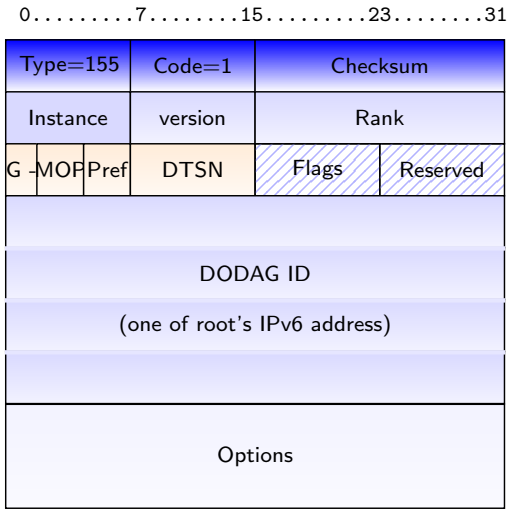
Upward traffic: DoDAG



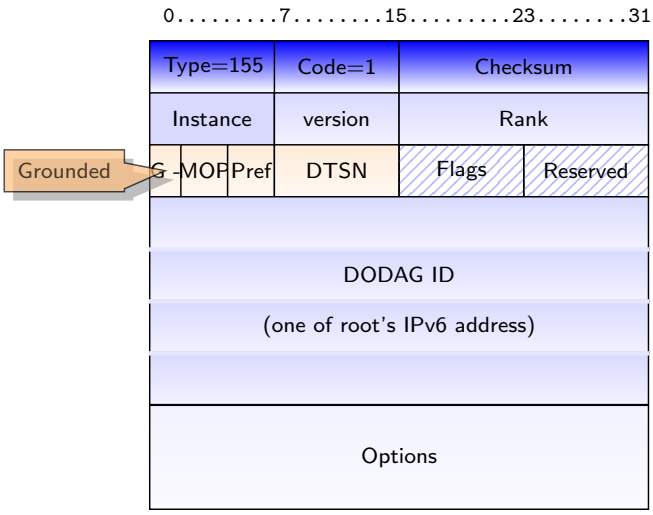
Upward traffic: DoDAG



DIO fields

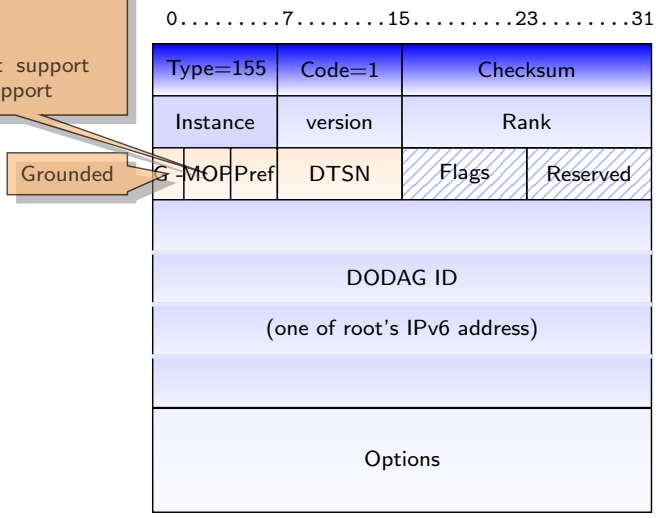


DIO fields



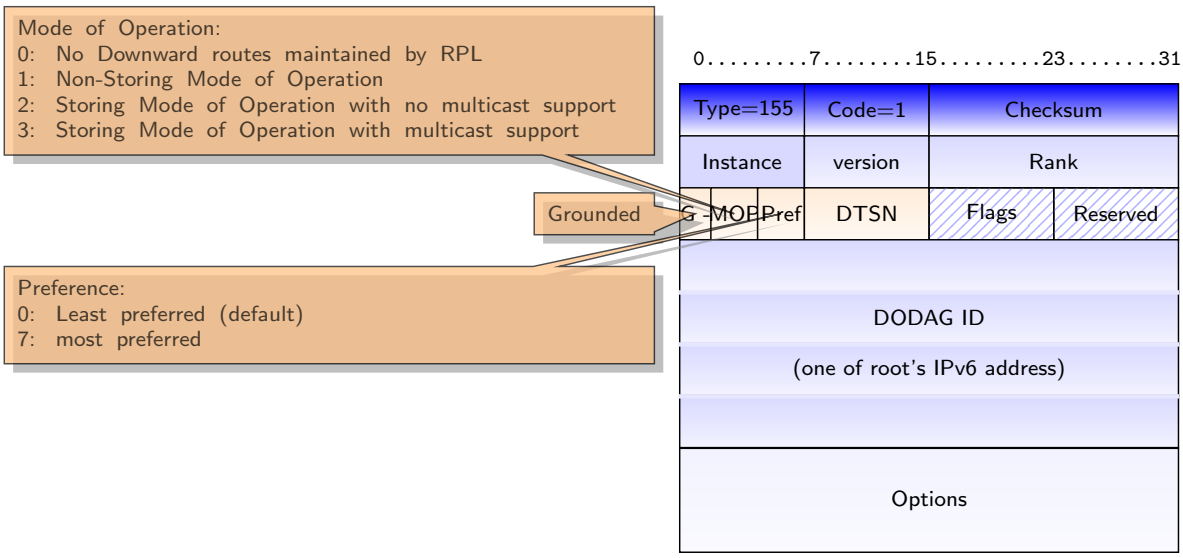
DIO fields

Mode of Operation:
0: No Downward routes maintained by RPL
1: Non-Storing Mode of Operation
2: Storing Mode of Operation with no multicast support
3: Storing Mode of Operation with multicast support



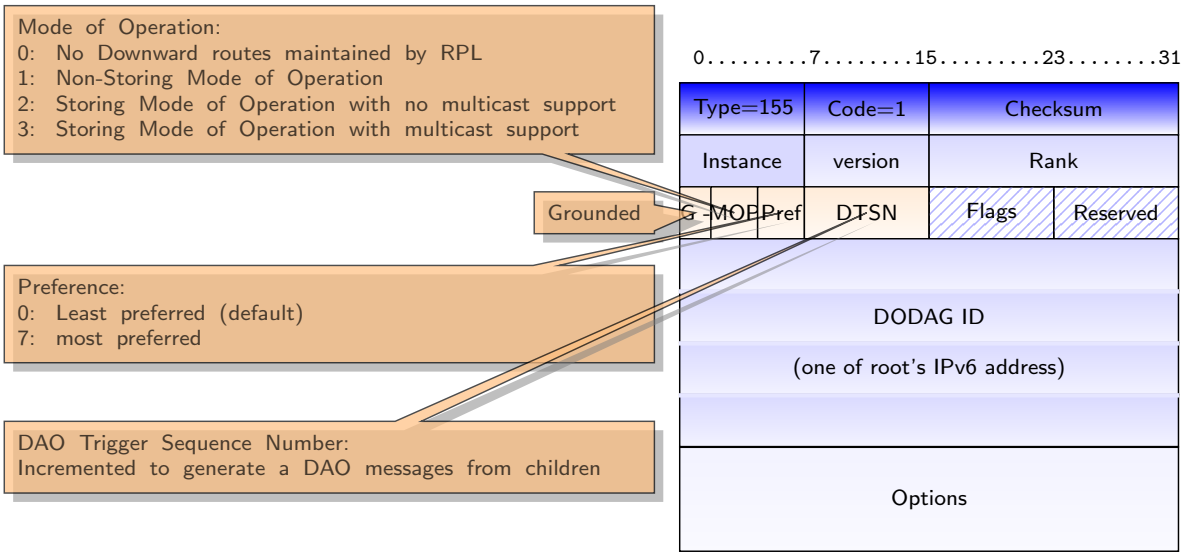
DIO fields

RPL



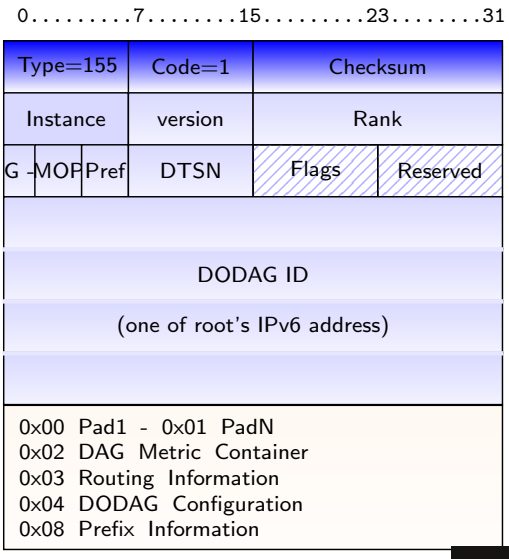
DIO fields

RPL



DIO options

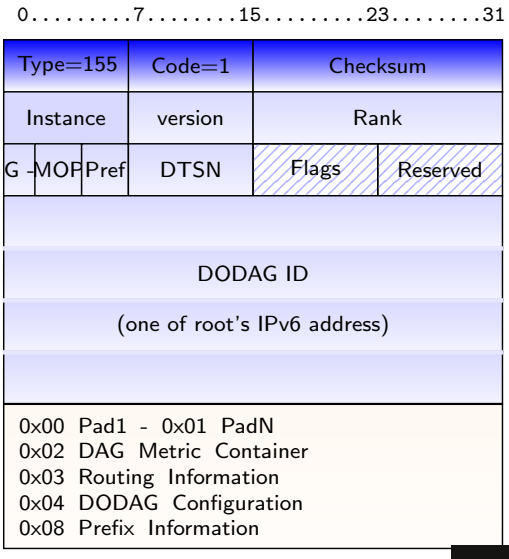
RPL



DIO options

RPL

DAG Metric Container (see RFC 6552)



DIO options

RPL

DAG Metric Container (see RFC 6552)

0x02	Length (B)	Metric Data

Route Information (from RFC 4191 Default Router Pref)

0x03	Length (B)	Pref Lenght	---PP---
Prefix Lifetime			
Prefix			

0.....7.....15.....23.....31

Type=155	Code=1	Checksum	
Instance	version	Rank	
G	MOP	Pref	DTSN	Flags	Reserved
DODAG ID			
(one of root's IPv6 address)			

0x00 Pad1 - 0x01 PadN
0x02 DAG Metric Container
0x03 Routing Information
0x04 DODAG Configuration
0x08 Prefix Information

DIO options

RPL

DAG Metric Container (see RFC 6552)

0x02	Length (B)	Metric Data

Route Information (from RFC 4191 Default Router Pref)

0x03	Length (B)	Pref Lenght	---PP---
Prefix Lifetime			
Prefix			

DODAG Configuration

0x03	14	---A	PCS	DIOIntDoubl
DIOIntMin.	DIORedun.	MaxRankIncrease		
MinHopRankIncrease		Obj. func. CP.		
Reserved	Def. Lifetime	Lifetime Unit		

0.....7.....15.....23.....31

Type=155	Code=1	Checksum	
Instance	version	Rank	
G	MOP	Pref	DTSN	Flags	Reserved	
DODAG ID			
(one of root's IPv6 address)			

0x00 Pad1 - 0x01 PadN
0x02 DAG Metric Container
0x03 Routing Information
0x04 DODAG Configuration
0x08 Prefix Information

Slide 121 Laurent Toutain Filière 2

DIO options

DAG Metric Container (see RFC 6552)

0x02	Length (B)	Metric Data

Route Information (from RFC 4191 Default Router Pref)

0x03	Length (B)	Pref Lenght	---PP---
Prefix Lifetime			
Prefix			

DODAG Configuration

0x03	14	---A	PCS	DIOIntDoubl
DIOIntMin.	DIORedun.	MaxRankIncrease		
MinHopRankIncrease		Obj. func. CP.		
Reserved	Def. Lifetime	Lifetime Unit		

Prefix Information Option (from RFC 4861 Neighbor Discovery)

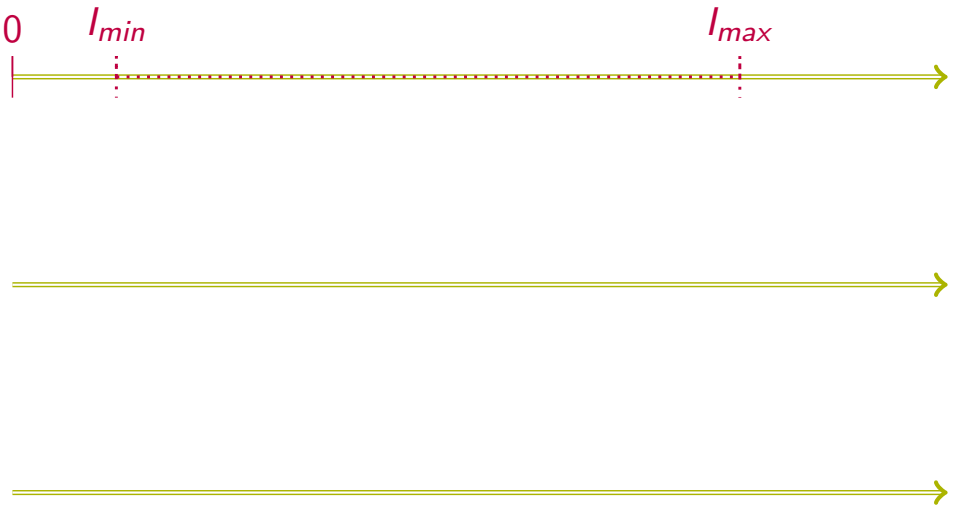
0x03	30	Pref Lenght	LAR- ----
Valid Lifetime			
Preferred Lifetime			
Reserved			
Prefix			

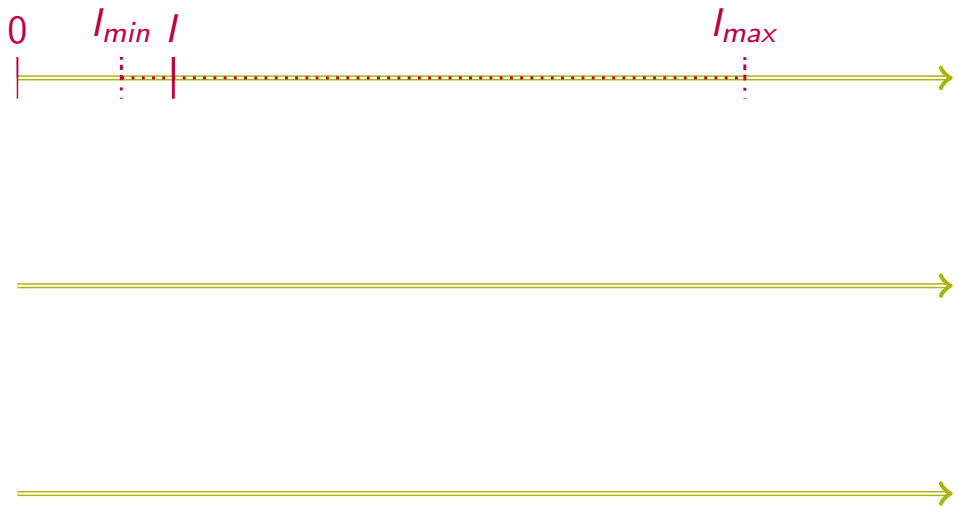
0.....7.....15.....23.....31

Type=155	Code=1	Checksum	
Instance	version	Rank	
G - MOP	Pref	DTSN	Flags / Reserved
DODAG ID			
(one of root's IPv6 address)			

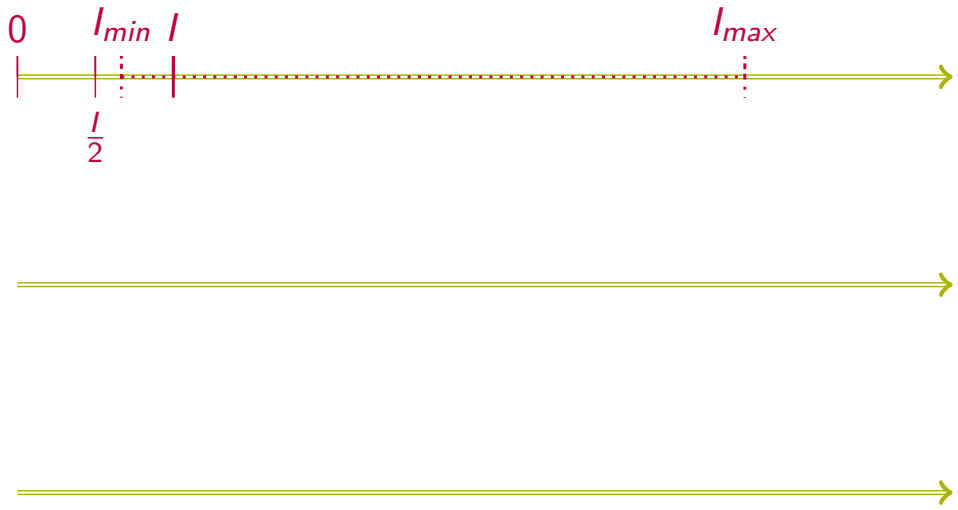
0x00 Pad1 - 0x01 PadN
0x02 DAG Metric Container
0x03 Routing Information
0x04 DODAG Configuration
0x08 Prefix Information

Trickle RFC 6202

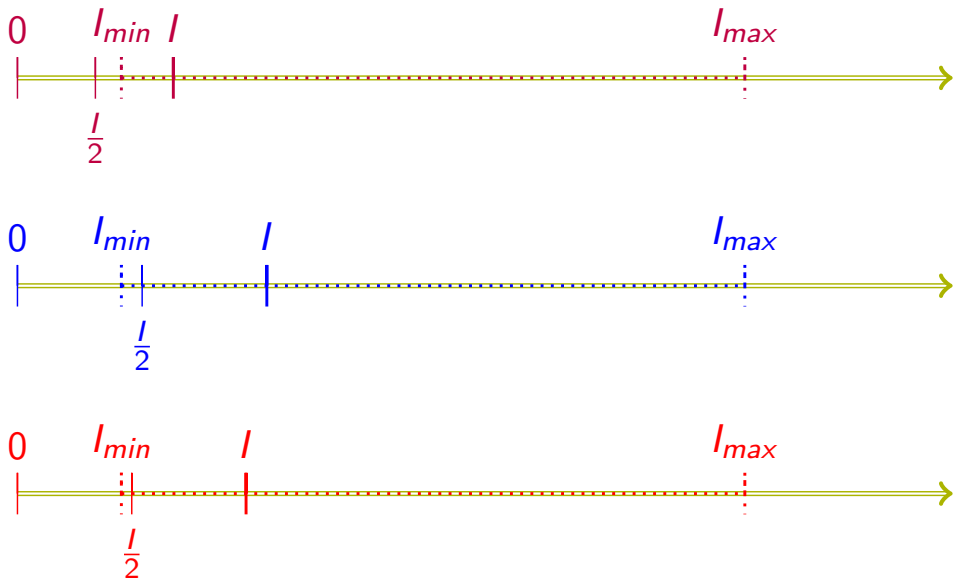




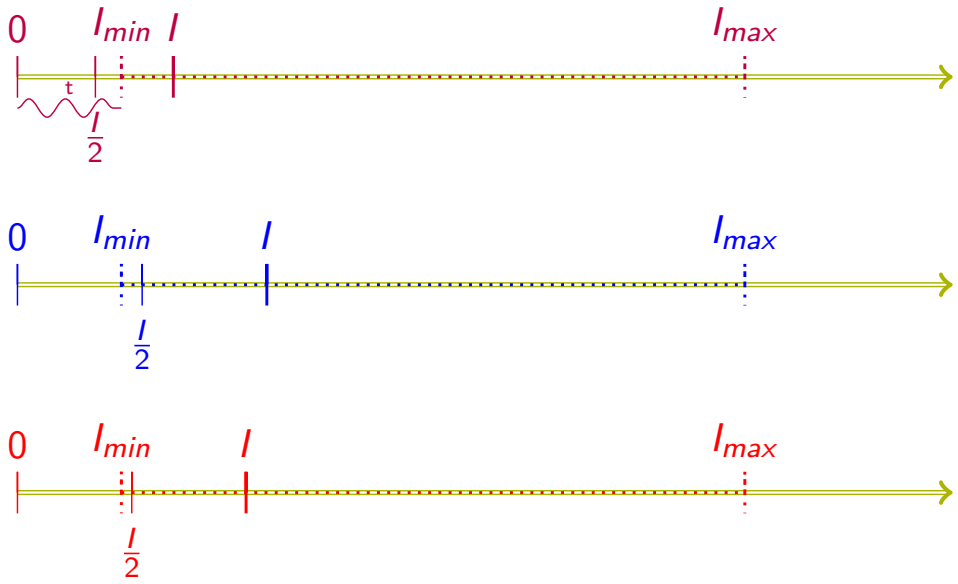
Trickle RFC 6202
RPL

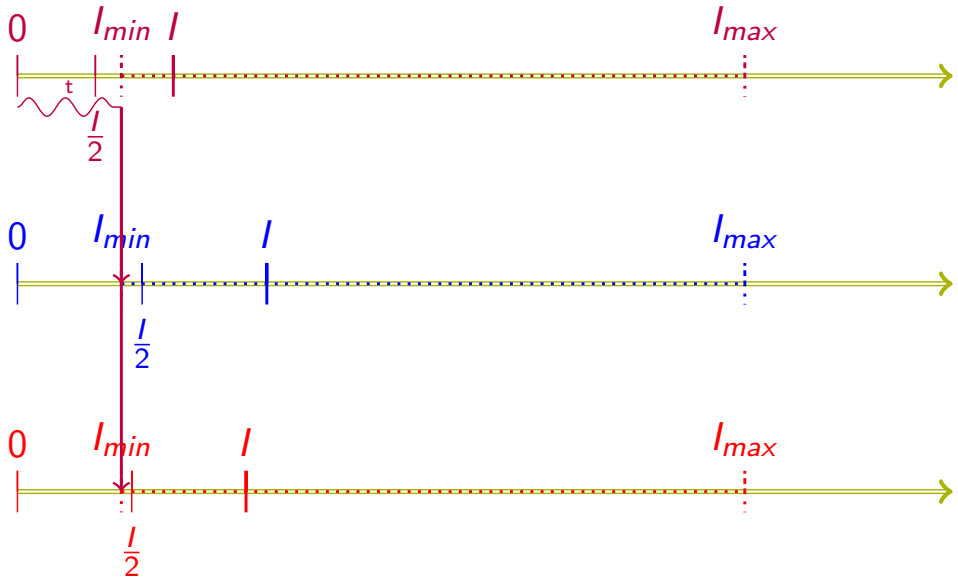


Slide 122 Laurent Toutain Filière 2
TELECOM Bretagne

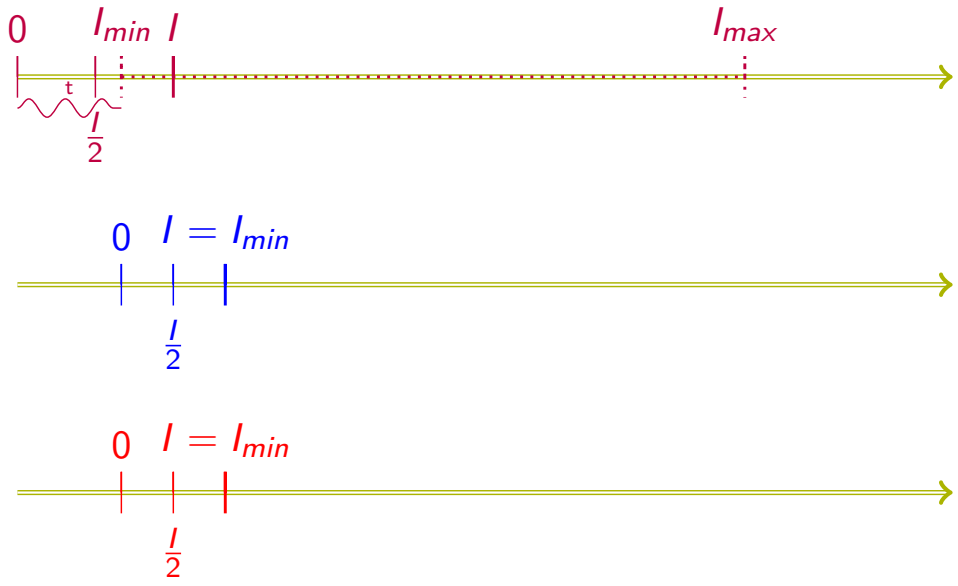


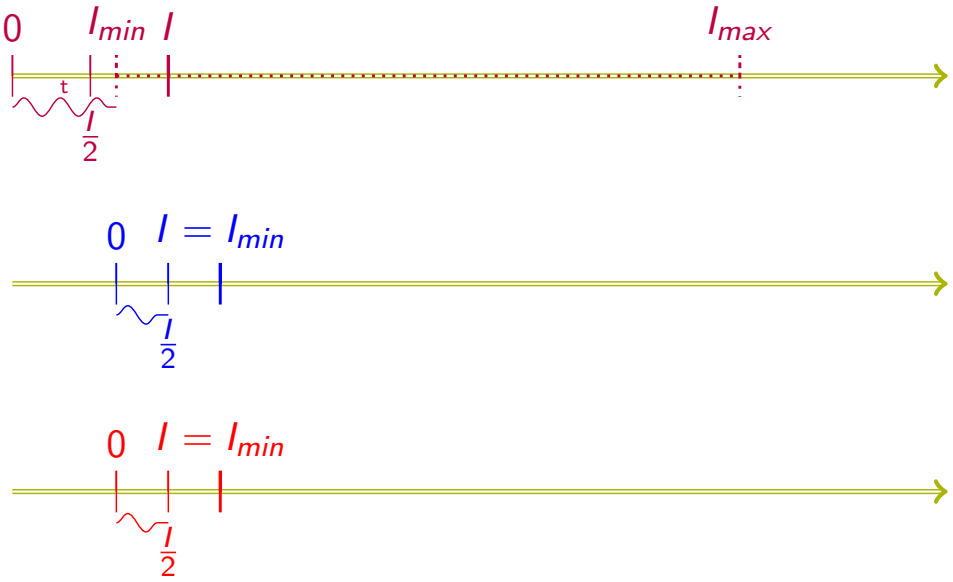
Trickle RFC 6202
RPL



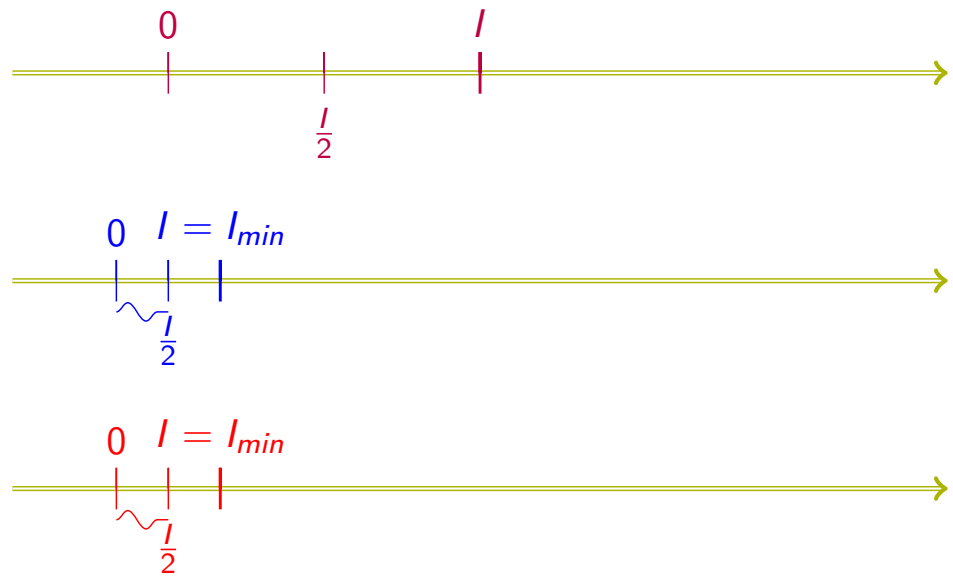


Trickle RFC 6202
RPL

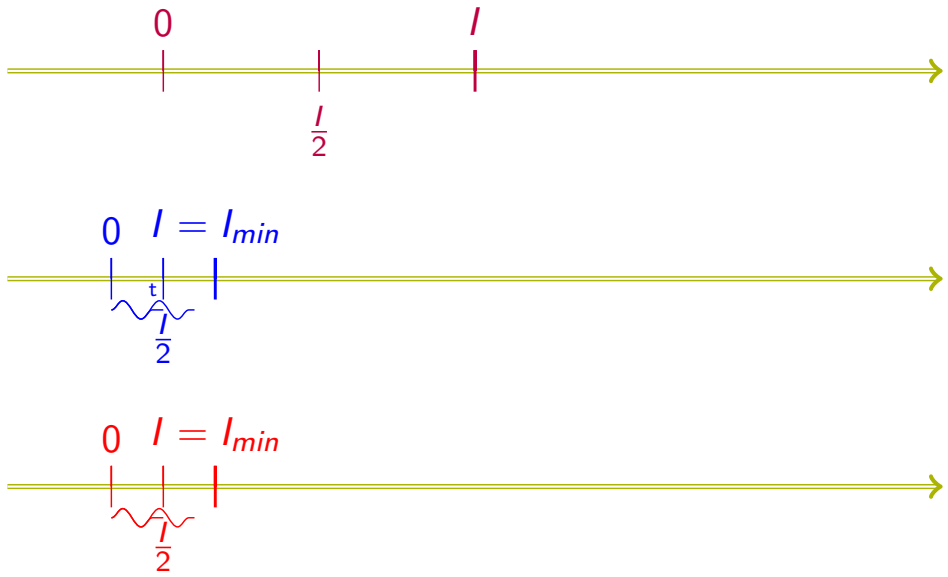




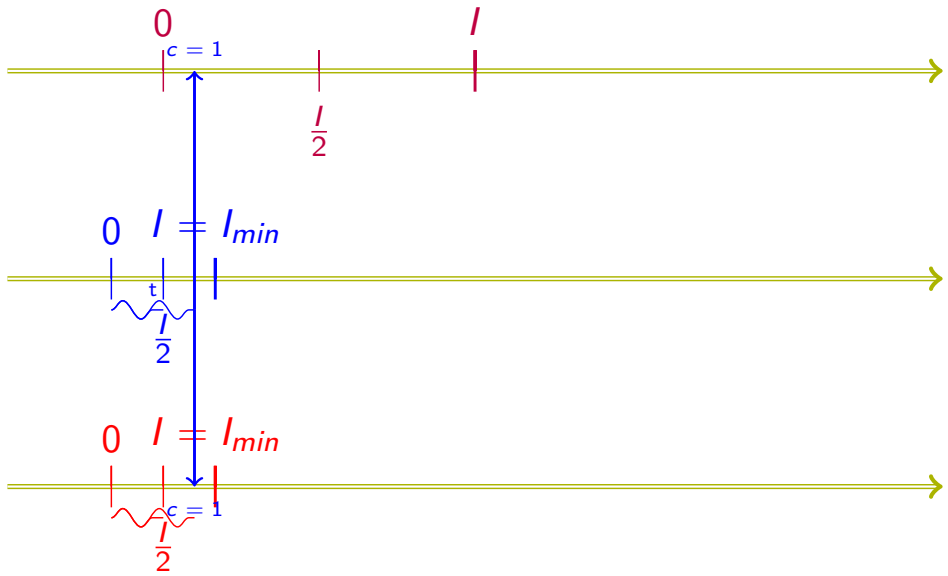
Trickle RFC 6202
RPL



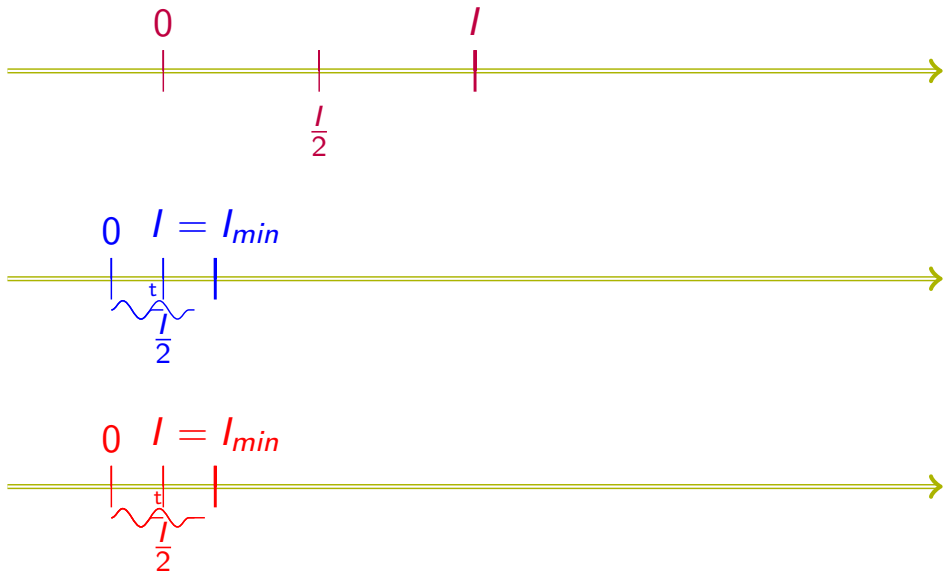
Slide 122 Laurent Toutain Filière 2
TELECOM Bretagne



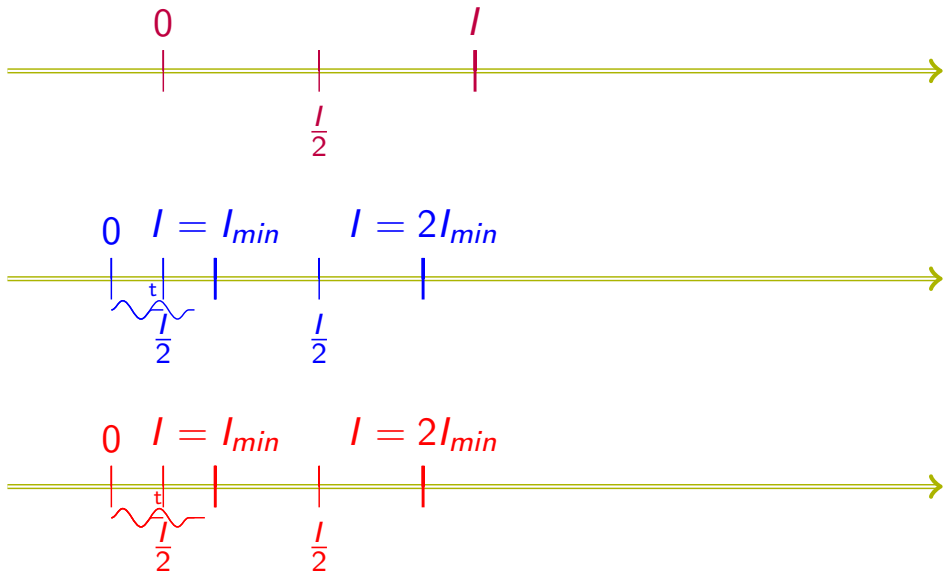
Trickle RFC 6202
RPL



Slide 122 Laurent Toutain Filière 2
TELECOM Bretagne



Trickle RFC 6202
RPL



DAO

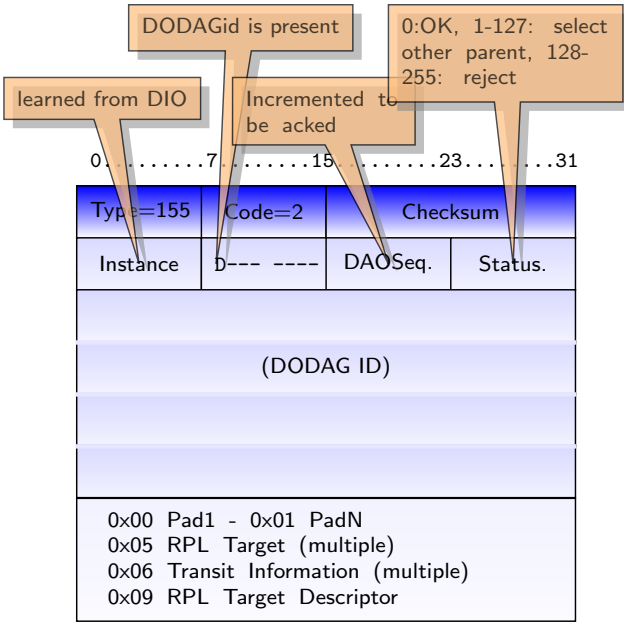
0.....7.....15.....23.....31

Type=155	Code=2	Checksum	
Instance	D--- ----	DAOSeq.	Status.
(DODAG ID)			

0x00 Pad1 - 0x01 PadN
0x05 RPL Target (multiple)
0x06 Transit Information (multiple)
0x09 RPL Target Descriptor

RPL

DAO



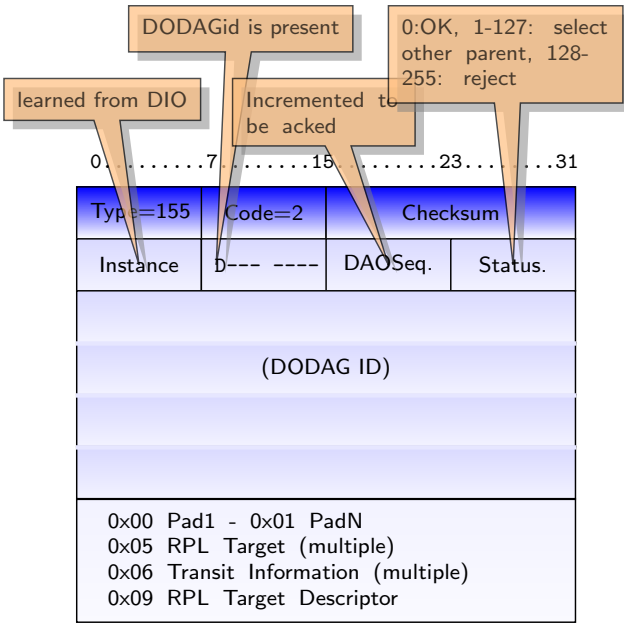
Slide 123 Laurent Toutain Filière 2

DAO

RPL

RPL Target (i.e. route or address)

0x05	Length (B)	Flags=0x00	Pref Length
Target Prefix			



DAO

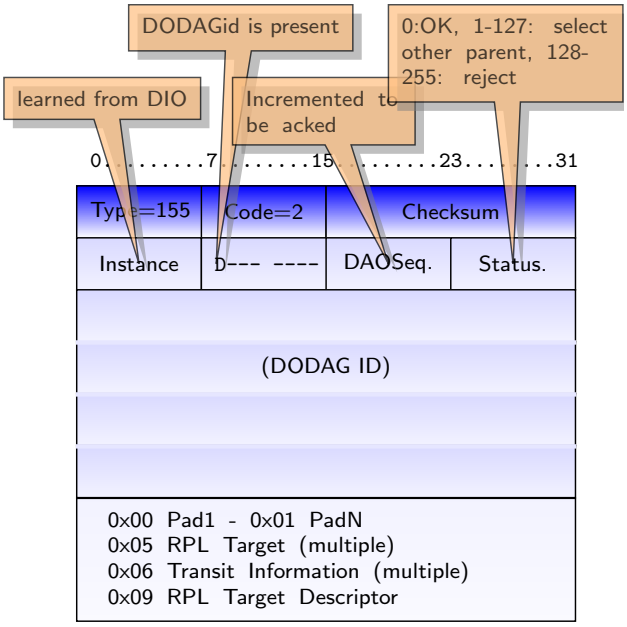
RPL

RPL Target (i.e. route or address)

0x05	Length (B)	Flags=0x00	Pref Length
Target Prefix			

Transit Information (i.e. parent)

0x06	Length (B)	E--- ----	Path ctrl
Path seq	Path Lifetime	Parent address	



DAO

RPL Target (i.e. route or address)

0x05	Length (B)	Flags=0x00	Pref Length
Target Prefix			

Transit Information (i.e. parent)

0x06	Length (B)	E--- ----	Path ctrl
Path seq	Path Lifetime	Parent address	

Target Descriptor

0x06	Length = 4	Descriptor
Descriptor (continued)		

DODAGid is present

0:OK, 1-127: select other parent, 128-255: reject

learned from DIO

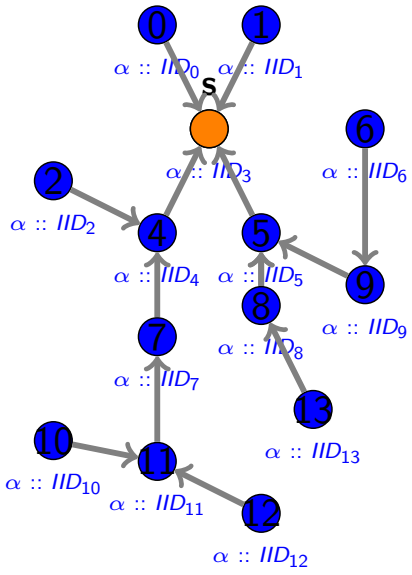
Incremented to be acked

0.....7.....15.....23.....31

Type=155	Code=2	Checksum	
Instance	D--- ----	DAOSeq.	Status.
(DODAG ID)			

0x00 Pad1 - 0x01 PadN
0x05 RPL Target (multiple)
0x06 Transit Information (multiple)
0x09 RPL Target Descriptor

DAO Storing Mode



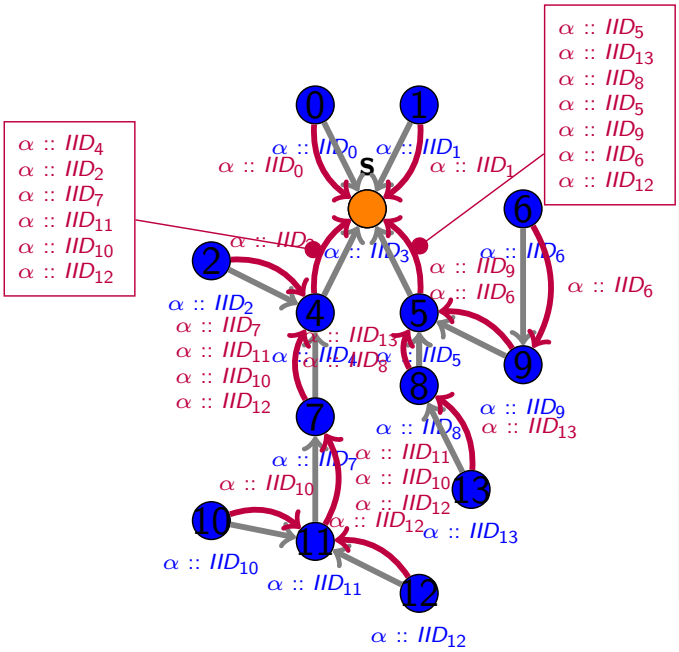
0.....7.....15.....23.....31

Type=155	Code=2	Checksum	
Instance	D--- ----	DAOSeq.	Status.
(DODAG ID)			

0x05 RPL Target (multiple)

DAO Storing Mode

RPL

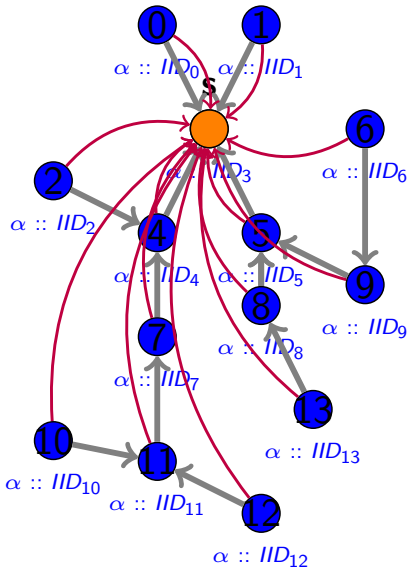


0.....7.....15.....23.....31

Type=155	Code=2	Checksum	
Instance	D--- ----	DAOSeq.	Status.
(DODAG ID)			
0x05 RPL Target (multiple)			

DAO Non-Storing Mode

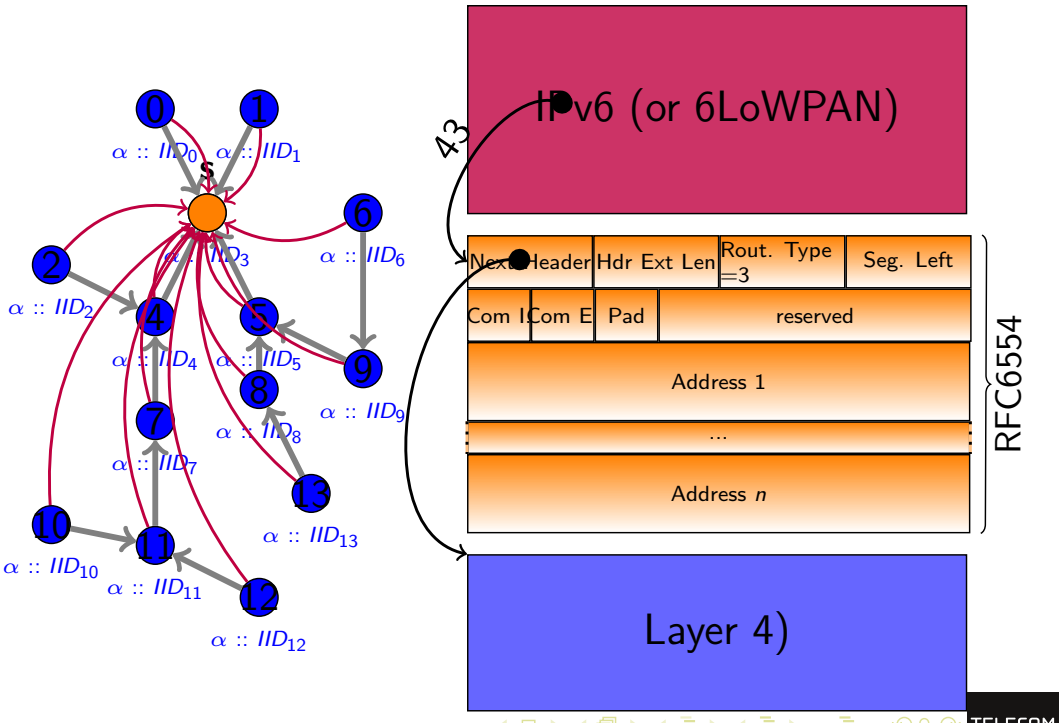
RPL



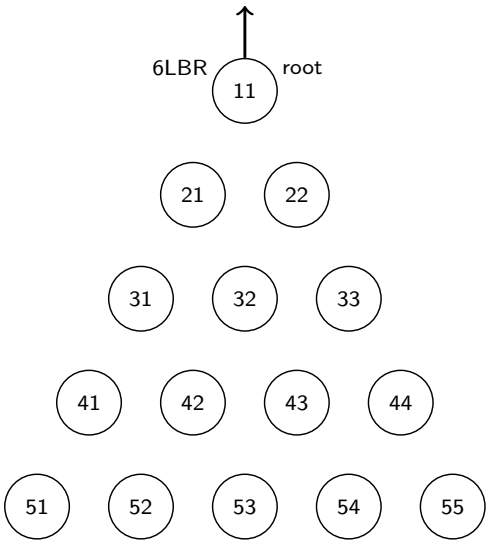
0.....7.....15.....23.....31

Type=155	Code=2	Checksum	
Instance	D--- ----	DAOSeq.	Status.
(DODAG ID)			
0x05 RPL Target 0x06 Transit Information			

DAO Non-Storing Mode

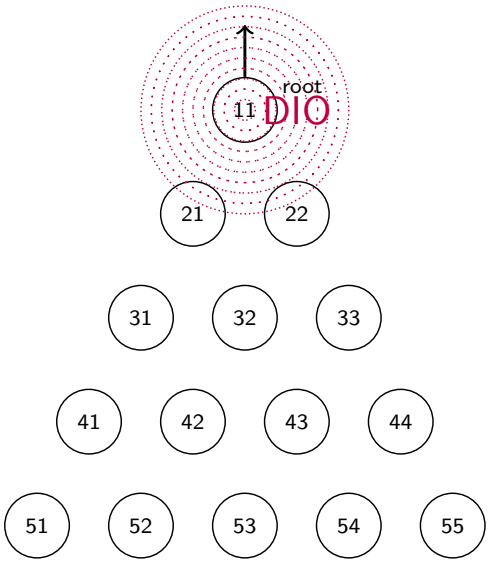


Upward traffic: DoDAG



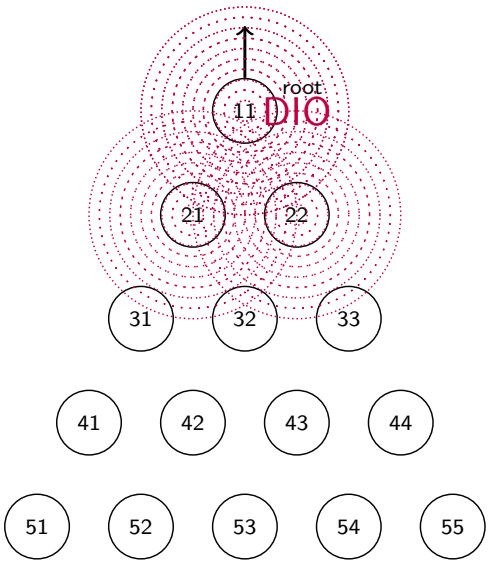
Upward traffic: DoDAG

RPL



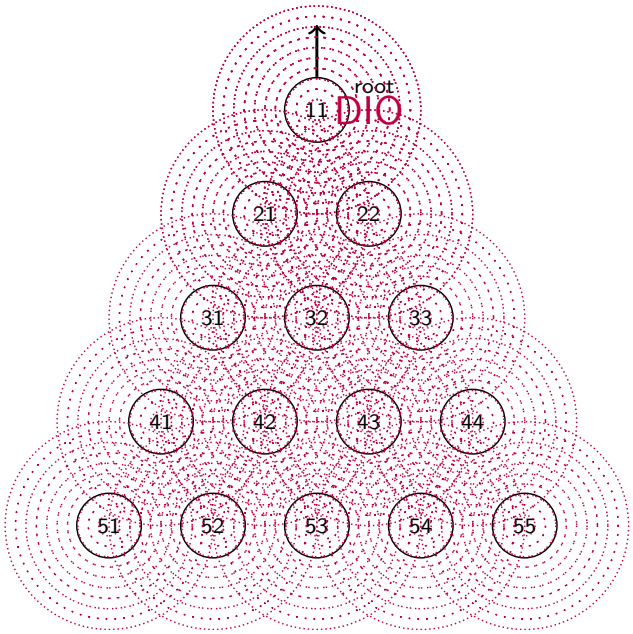
Upward traffic: DoDAG

RPL



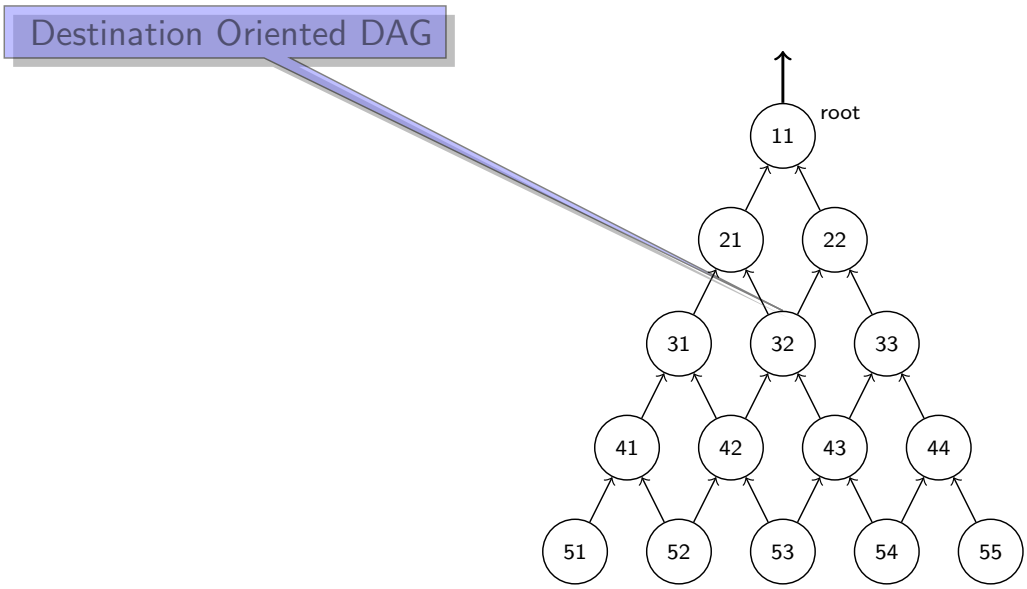
Upward traffic: DoDAG

RPL



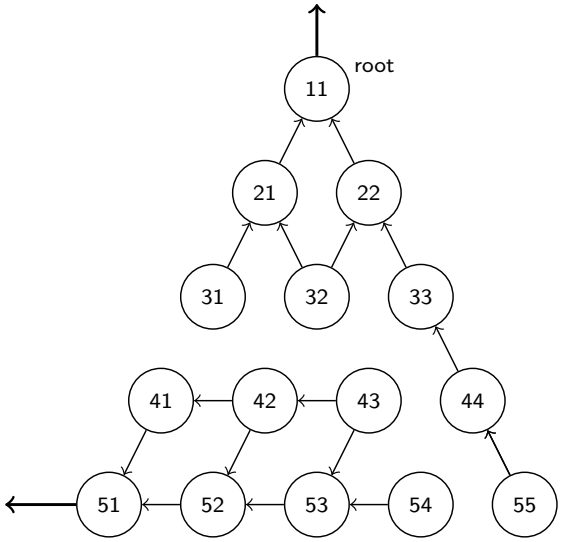
Upward traffic: DoDAG

RPL



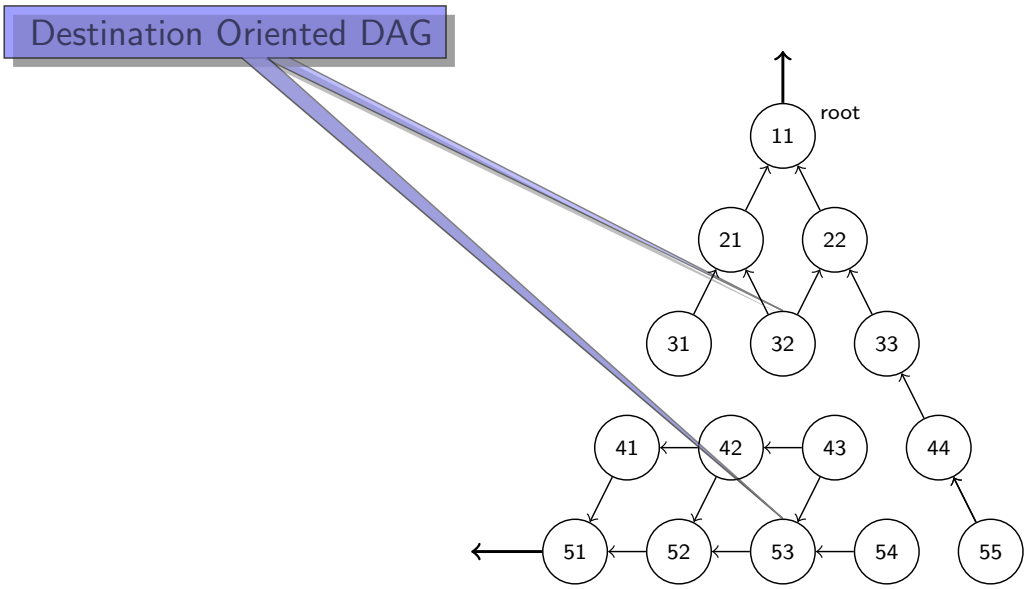
Upward traffic: DoDAG

RPL



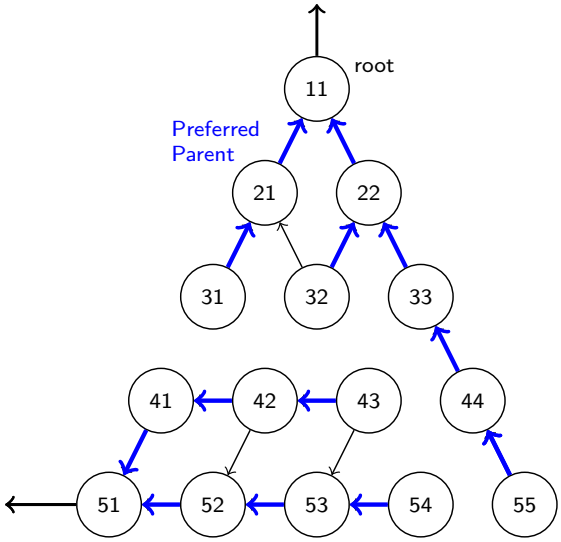
Upward traffic: DoDAG

RPL



Upward traffic: DoDAG

RPL



Upward traffic: DoDAG

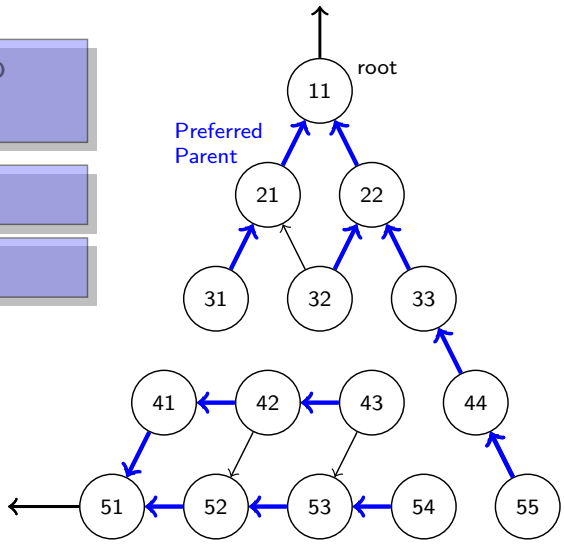
RPL

Objective Function is used to compute DoDAG:

- To compute rank

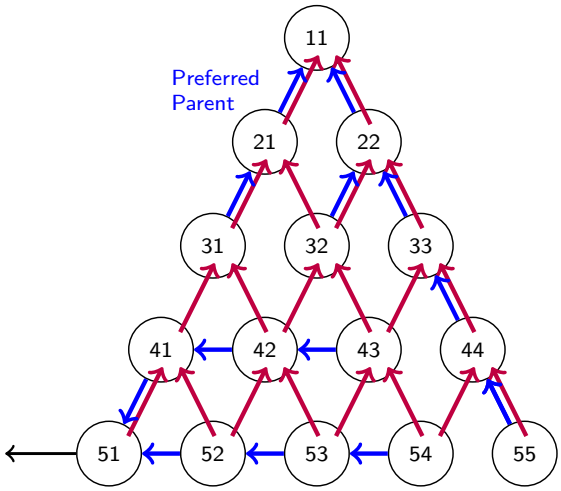
- To define preferred parent

A Node may never select a parent with a higher rank than itself
⇒ Avoid loops



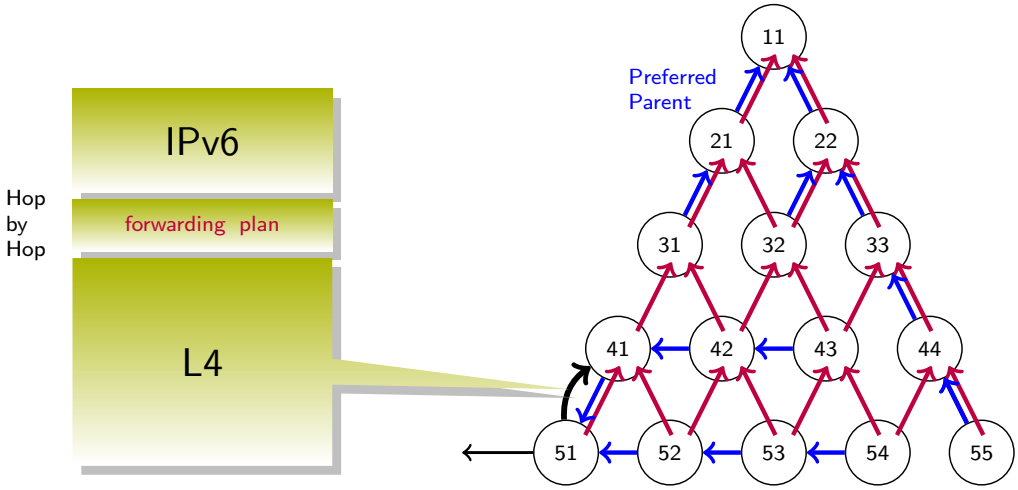
Upward traffic: DoDAG

RPL



Upward traffic: DoDAG

RPL



Metrics

Node Metrics	Link Metrics
Node State and Attributes Object Purpose is to reflects node workload (CPU, Memory…) "O" flag signals overload of resource "A" flag signal node can act as traffic aggregator	**Throughput Object** Currently available throughput (Bytes per second) Throughput range supported
Node Energy Object "T" flag: Node type: 0 = Mains, 1 = Battery, 2 = Scavenger "I" bit: Use node type as a constraint (include/ exclude) "E" flag: Estimated energy remaining	**Latency** Constraint - max latency allowable on path Metric - additive metric updated along path
Hop Count Object Constraint - max number of hops that can be traversed Metric - total number of hops traversed	**Link Reliability** Link Quality Level Reliability (LQL) 0=Unknown, 1=High, 2=Medium, 3=Low Expected Transmission Count (ETX) (Average number of TX to deliver a packet)
Object can be used as metric and/or constraint - metric can be additive/max/..	**Link Colour** Metric or constraint, arbitrary admin value

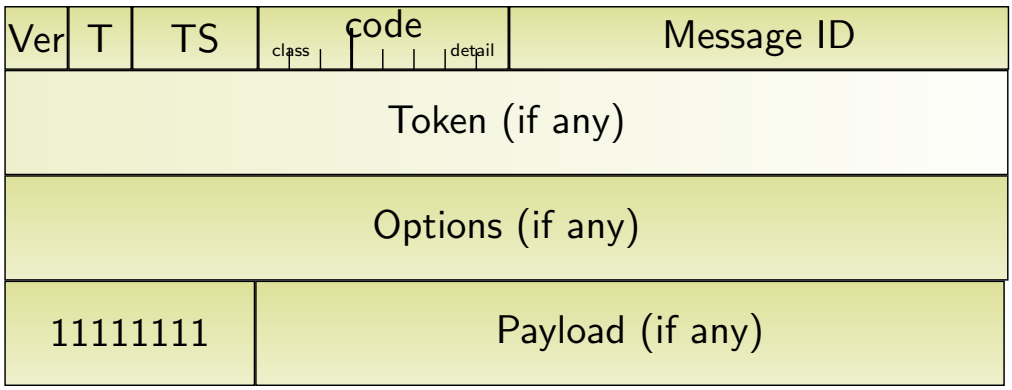
CoAP

- Constraint Application Protocol
- HTTP is too heavy for LoWPAN
  - TCP is not optimal (1 MSS windows)
  - Demand a lot of parsing to understand HTTP request
- CoAP is based on UDP and defines:
  - A simple message format easily processed by devices with limited processing resources,
  - A simple transport protocol to detect and correct packet losses,
  - A way to query or store information on a node using the REST paradigm,
  - A way to discover resources inside a LoWPAN network.
- CoAP includes a simple "transport" protocol

REST

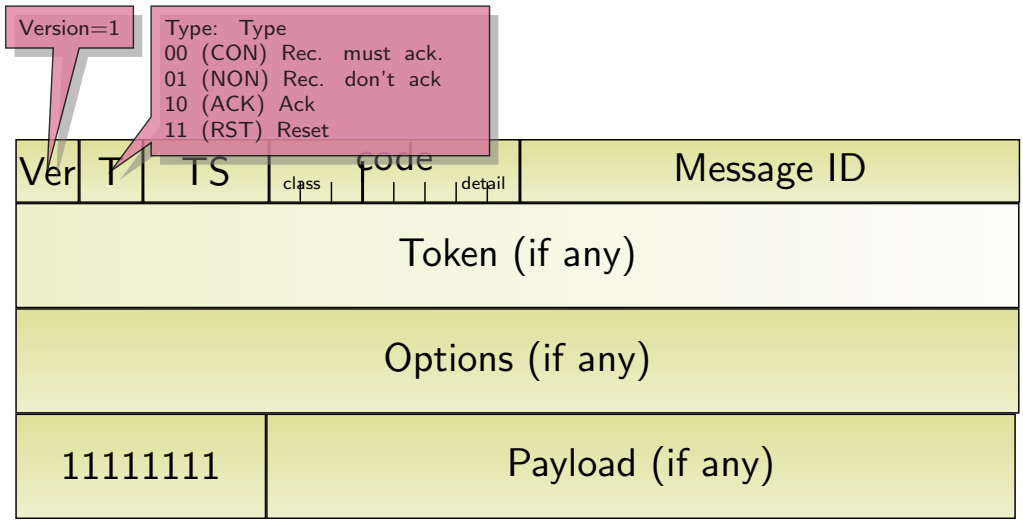
- Based on client/server, usually:
  - Server: Sensor/actuator having a value/action
  - Client: Internet host/Gateway requesting a value/action
- Defines 4 primitives:
  - GET: client request a resource value
  - PUT: client set assign a value to an already defined resource
  - POST: client create a resource and set a value
  - DELETE: client delete a resource
- Resources are identified by URI
- Requests are cachable

CoAP message format

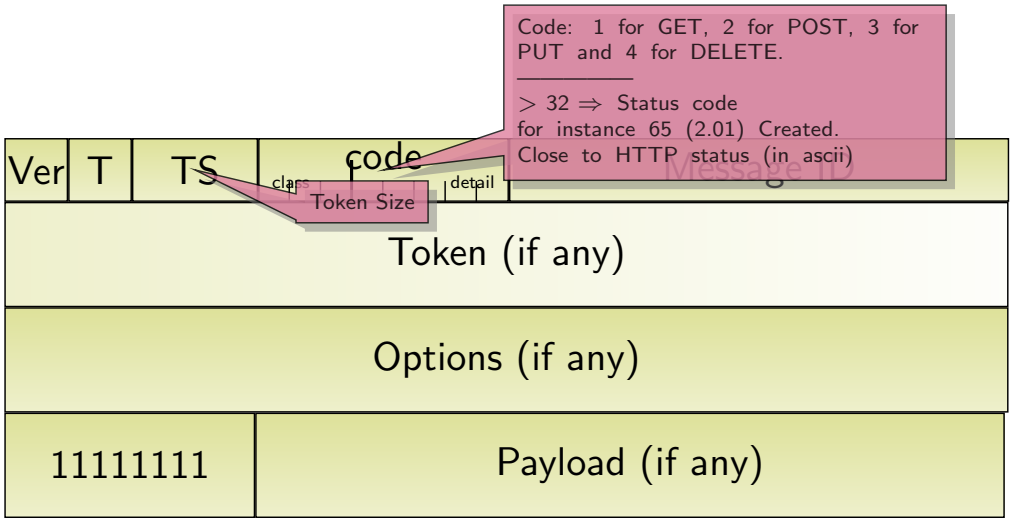


CoAP message format

CoAP

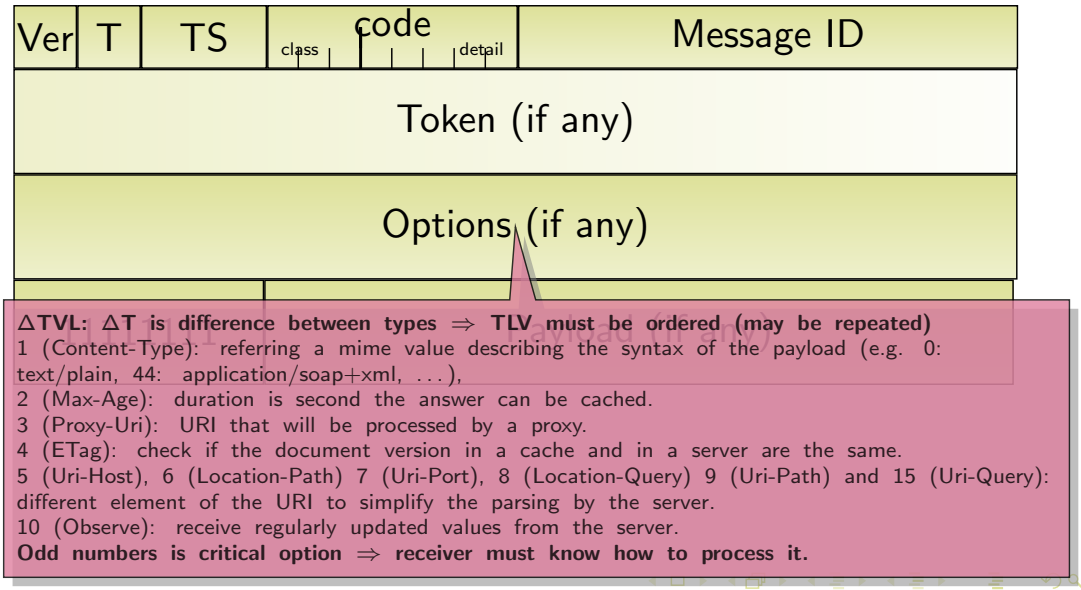


CoAP message format



CoAP message format

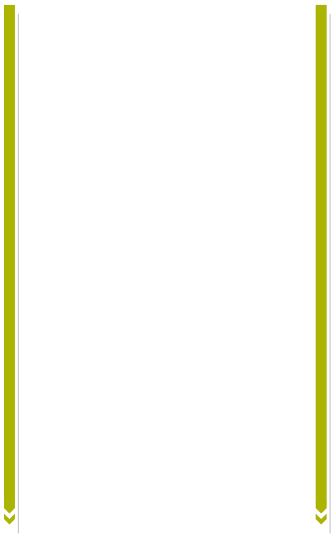
CoAP



Transport protocol

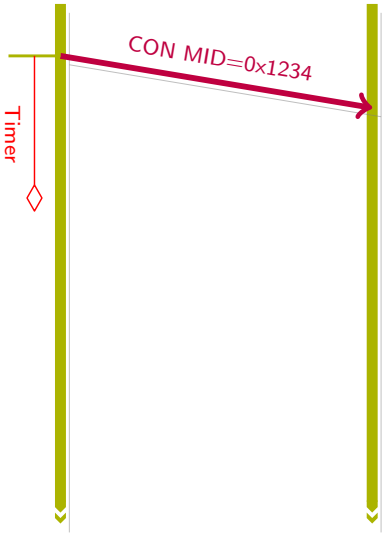
Simple case

CoAP



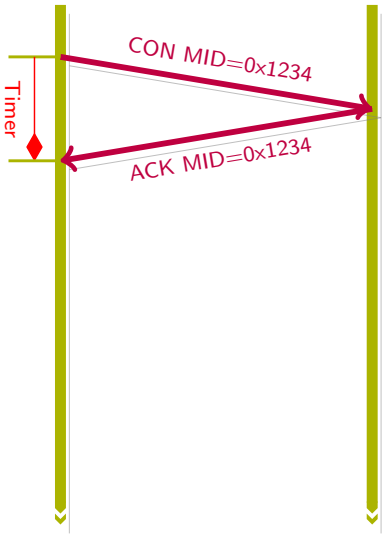
Simple case

CoAP



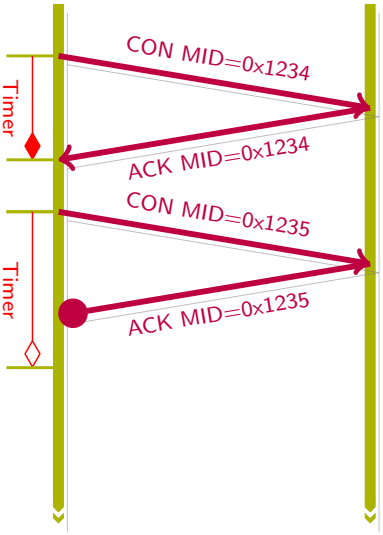
Simple case

CoAP



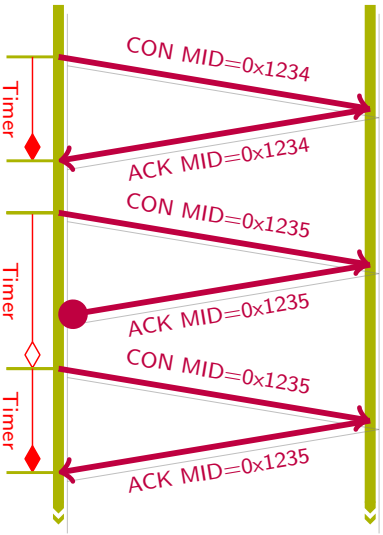
Simple case

CoAP

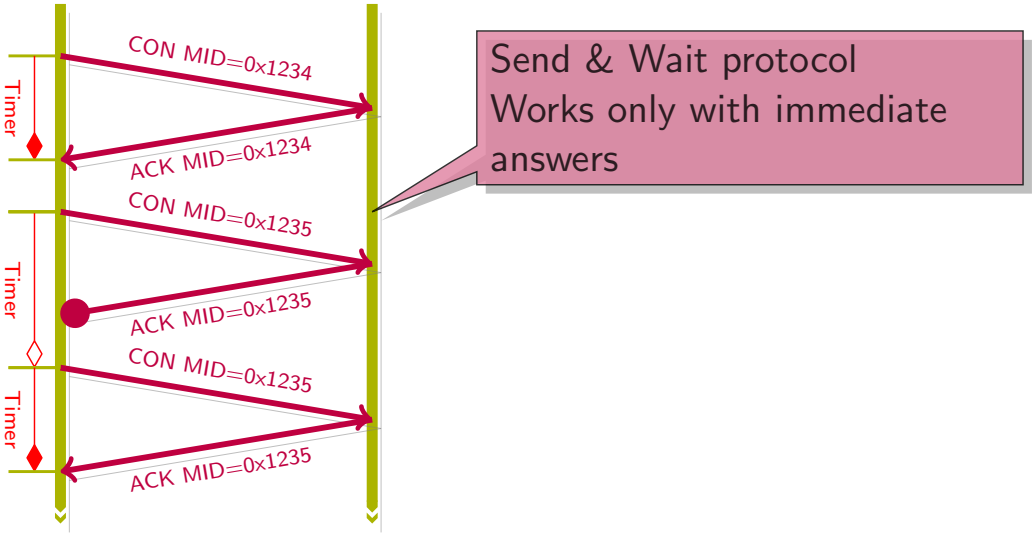


Simple case

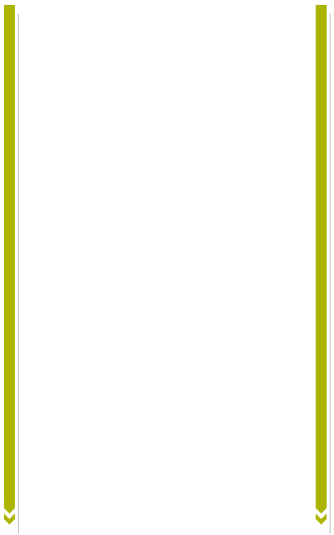
CoAP



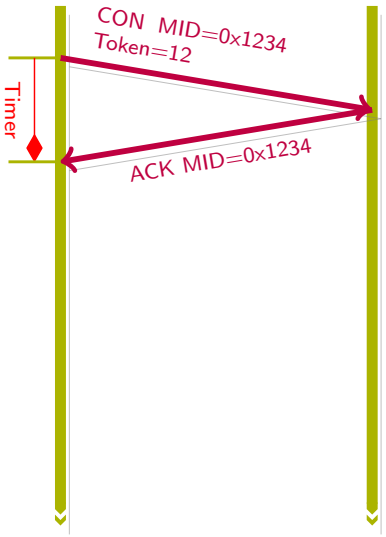
Simple case



Asynchronous exchanges: TLV Token

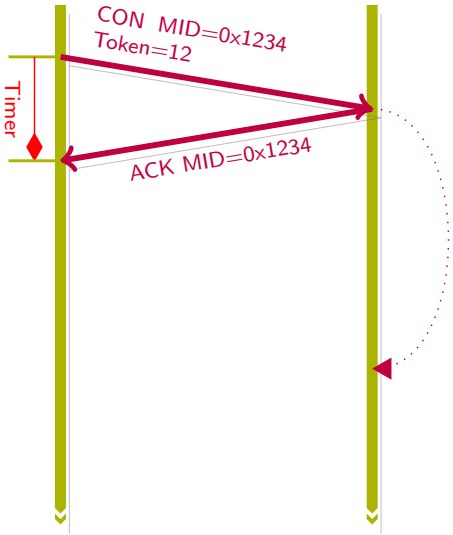


Asynchronous exchanges: TLV Token

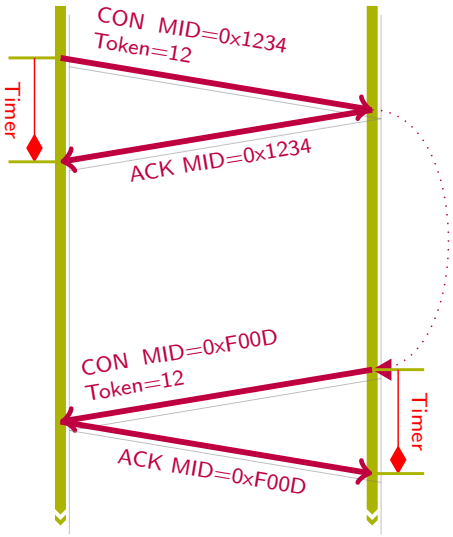


Asynchronous exchanges: TLV Token

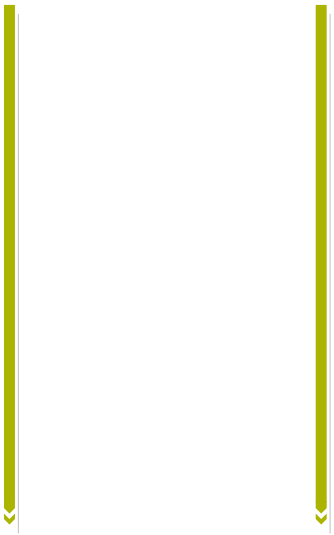
CoAP



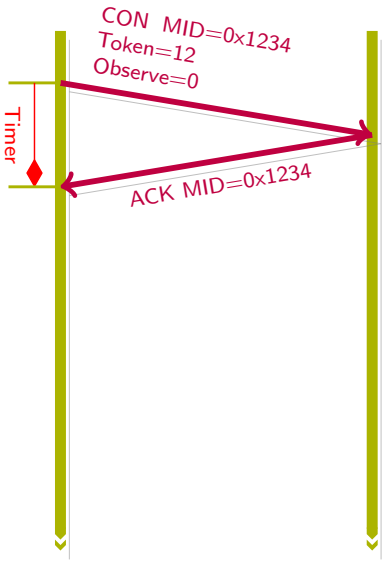
Asynchronous exchanges: TLV Token



Periodic exchanges: TLV Observe

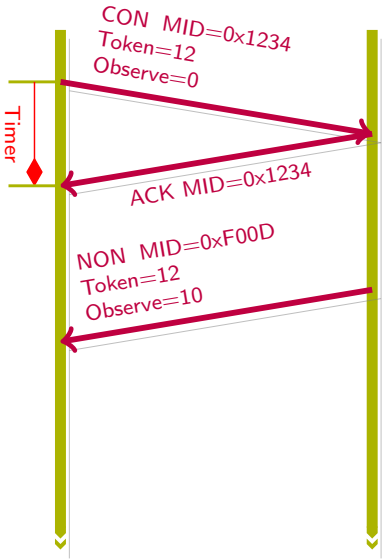


Periodic exchanges: TLV Observe

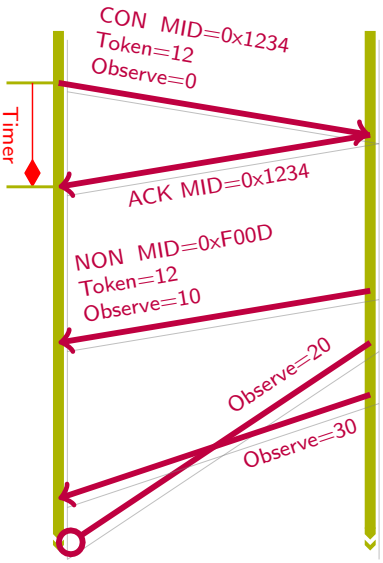


Periodic exchanges: TLV Observe

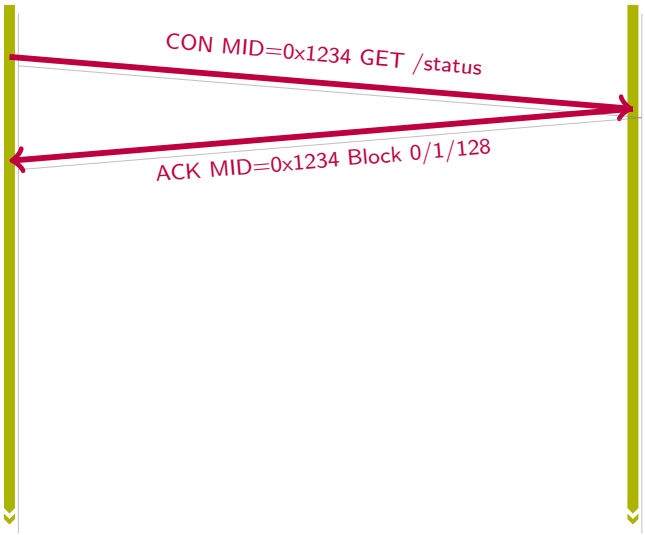
CoAP



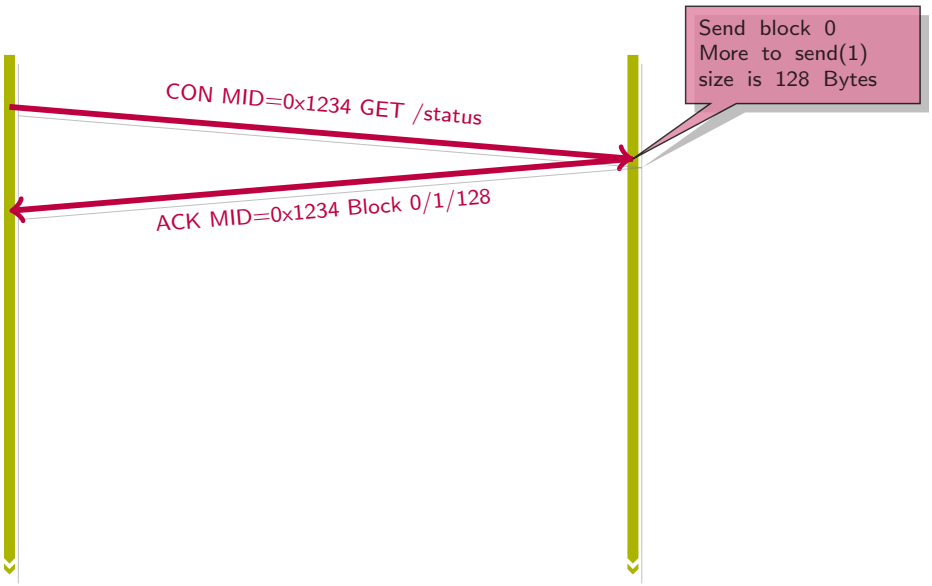
Periodic exchanges: TLV Observe



Large block Transfer exchanges: TLV Block

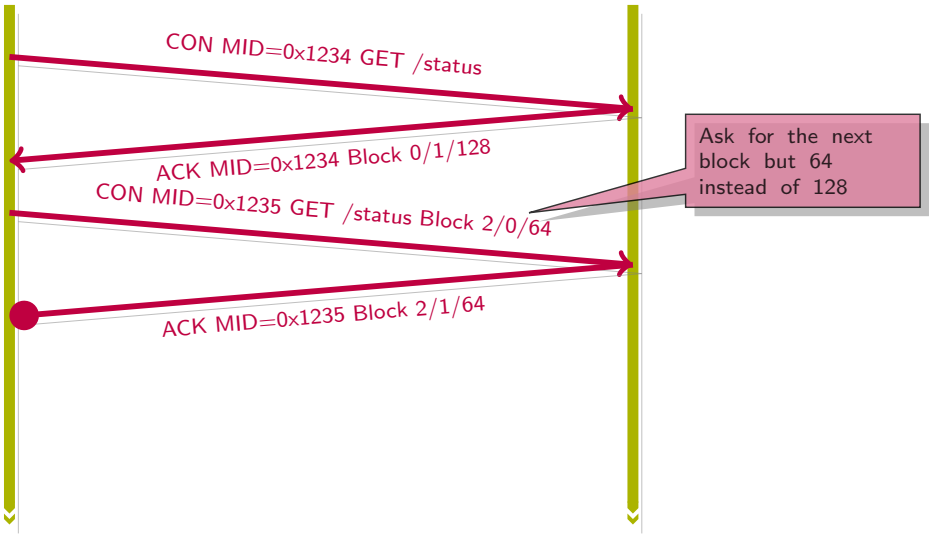


Large block Transfer exchanges: TLV Block



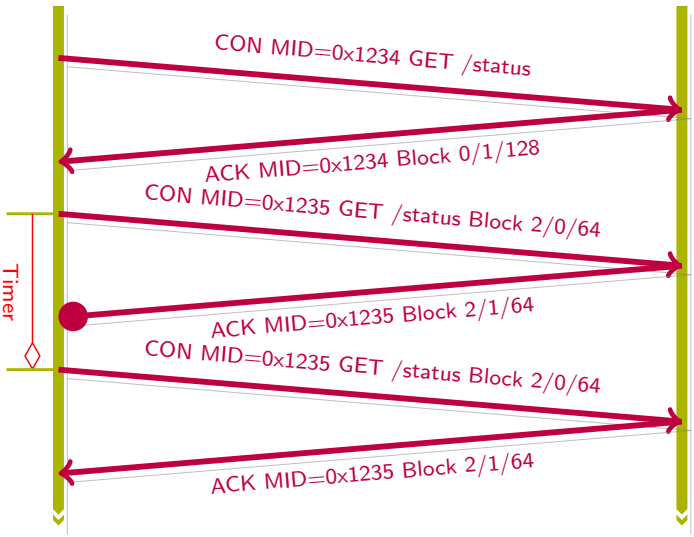
Large block Transfer exchanges: TLV Block

CoAP



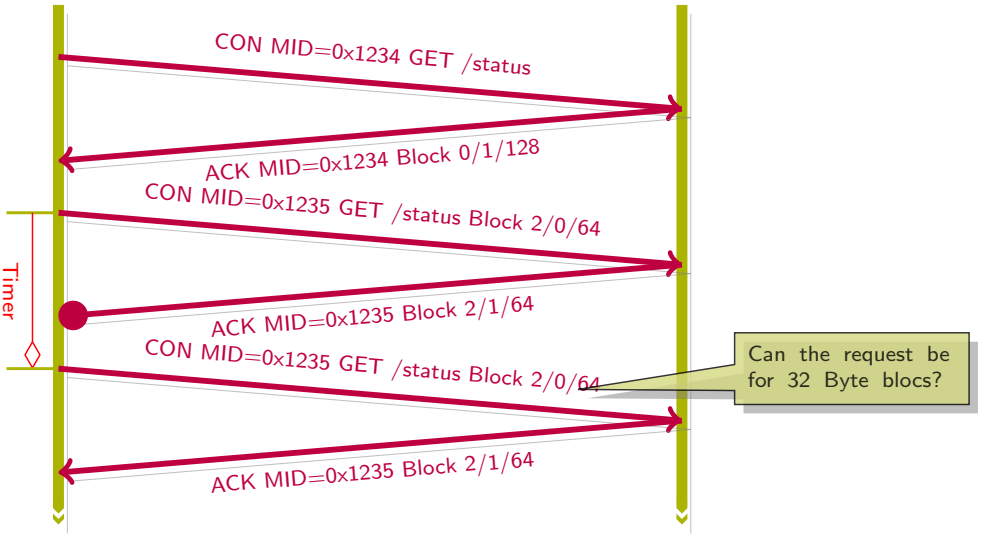
Large block Transfer exchanges: TLV Block

CoAP

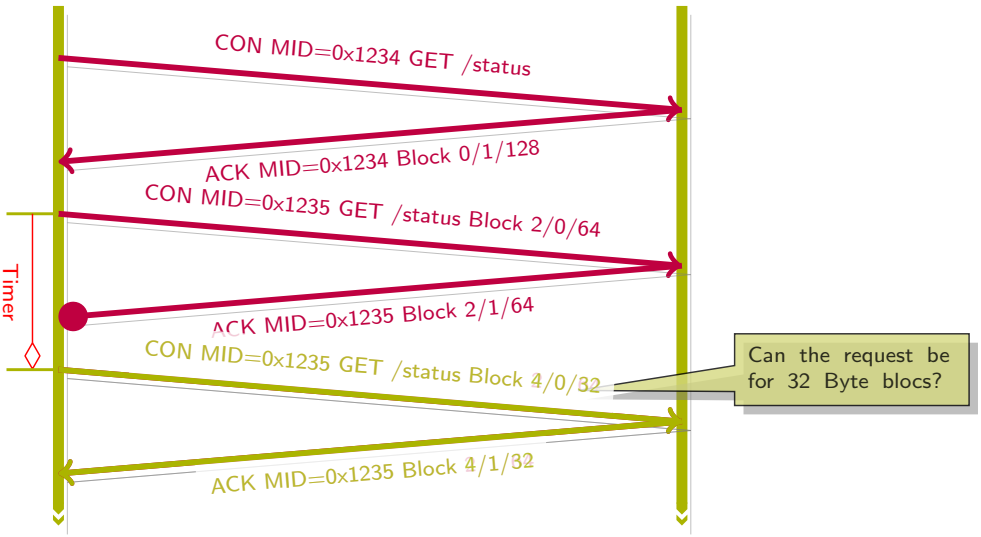


Large block Transfer exchanges: TLV Block

CoAP



Large block Transfer exchanges: TLV Block

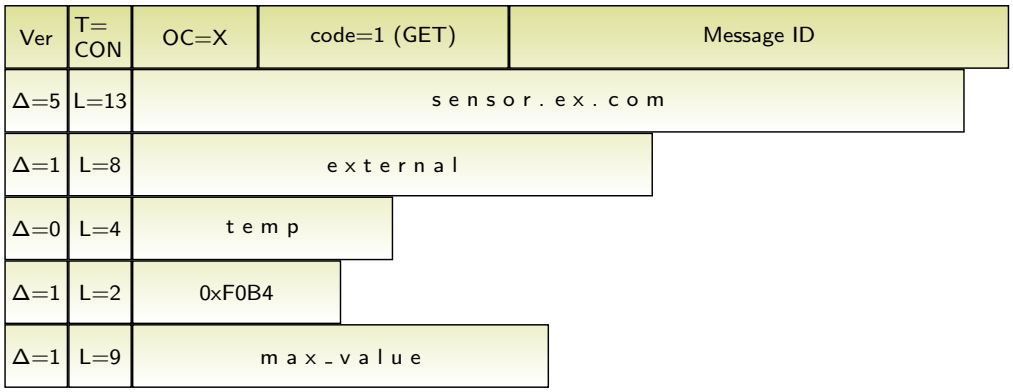


REST

REST Architecture

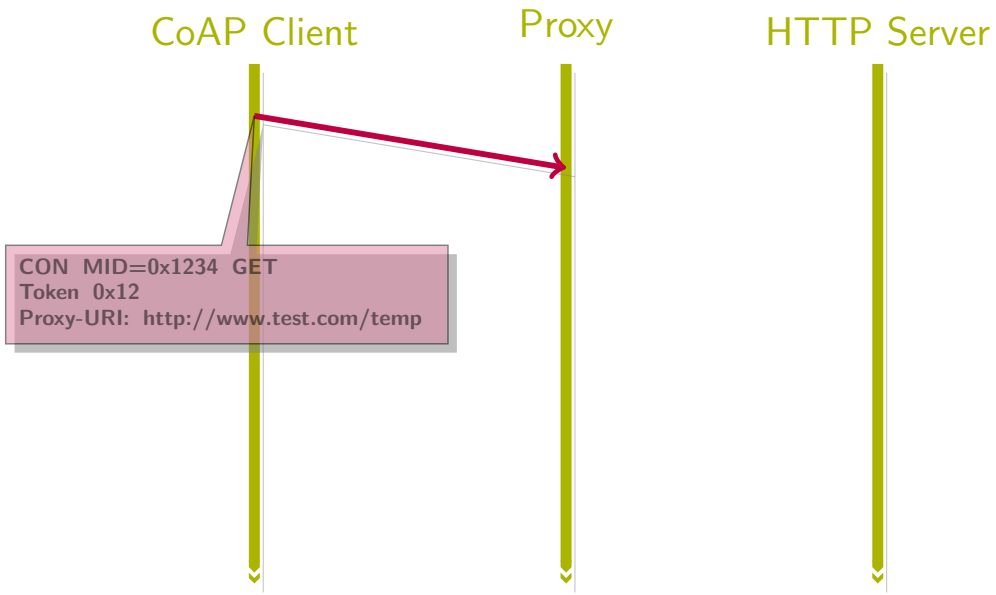
- 4 interactions:
  - GET (code 1 in CoAP header): The client requests the information located on the server
  - POST (code 2): The client creates and stores the information on a resource located in the server
  - PUT (code 3): The client stores information on a resource based on the server
  - DELETE (code 4): The client removes a resource on the server.
- URI
  - `coap://sensor.ex.com:61620/external/temp?max_value`

CoAP URI coding



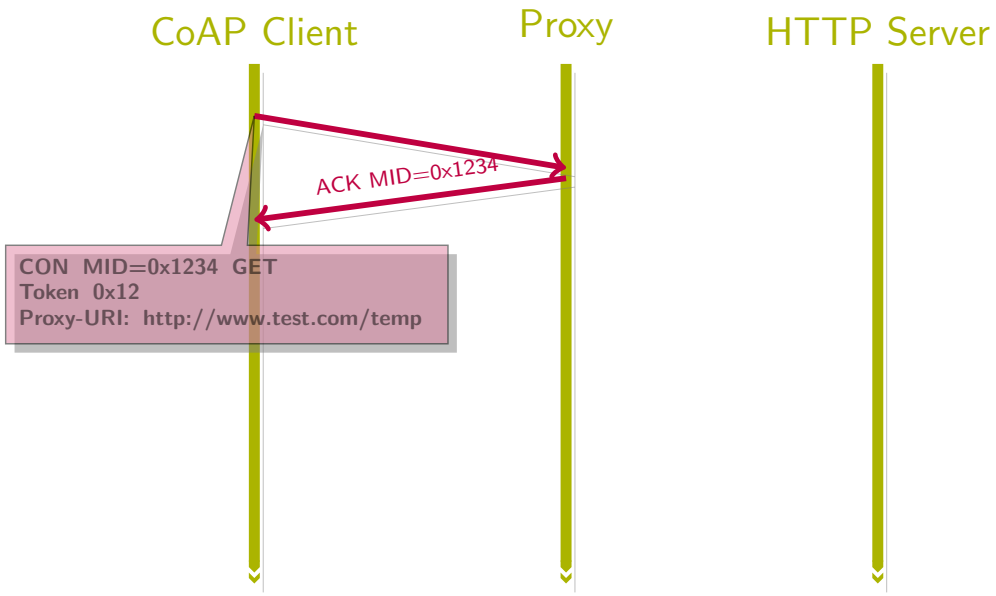
`GET coap://sensor.ex.com:61620/external/temp?max_value`

HTTP Mapping



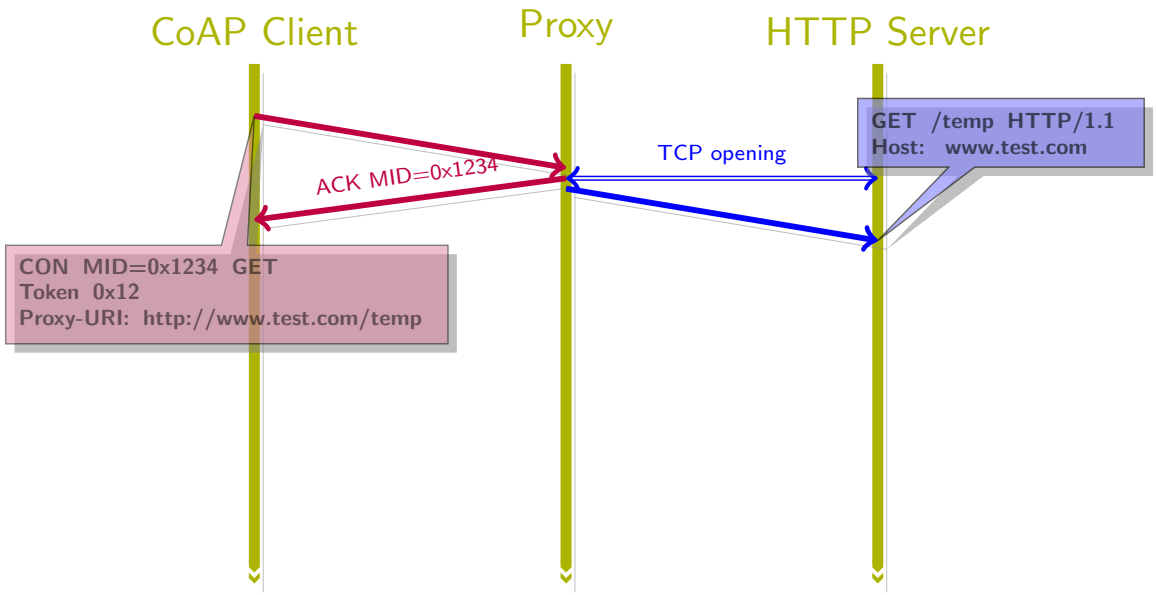
HTTP Mapping

CoAP



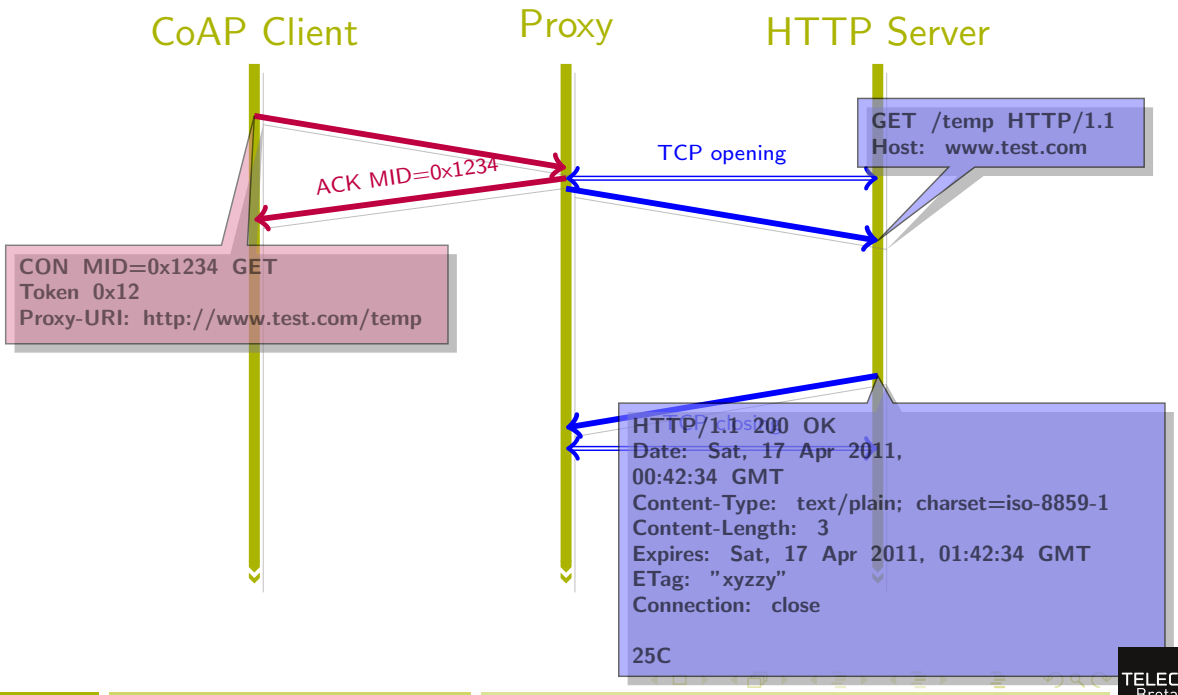
HTTP Mapping

CoAP

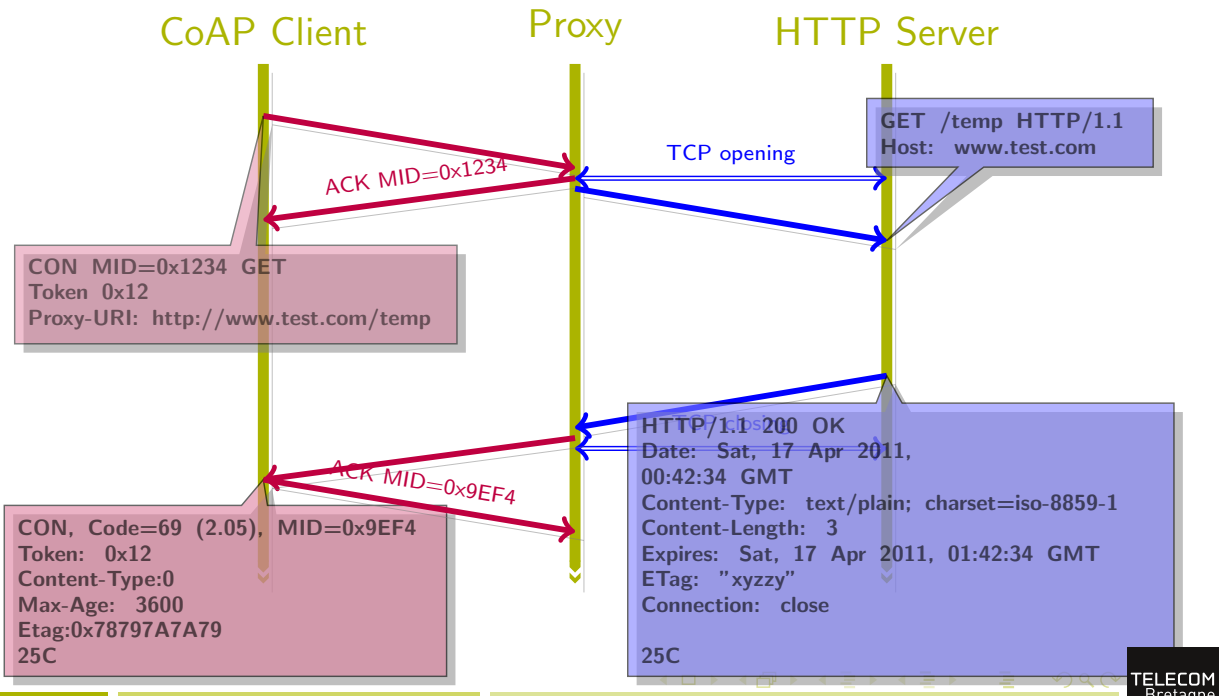


HTTP Mapping

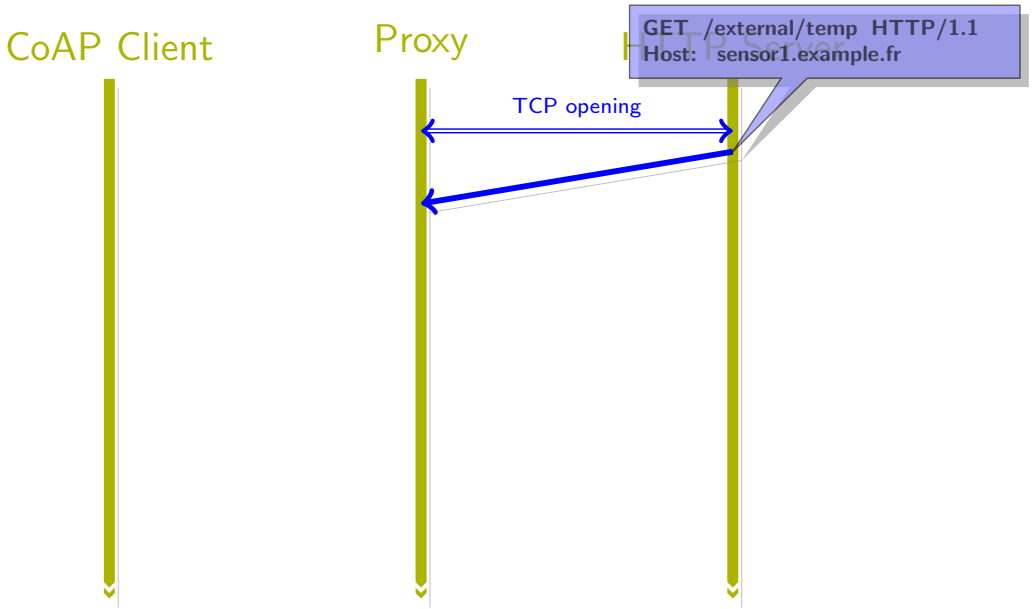
CoAP



HTTP Mapping

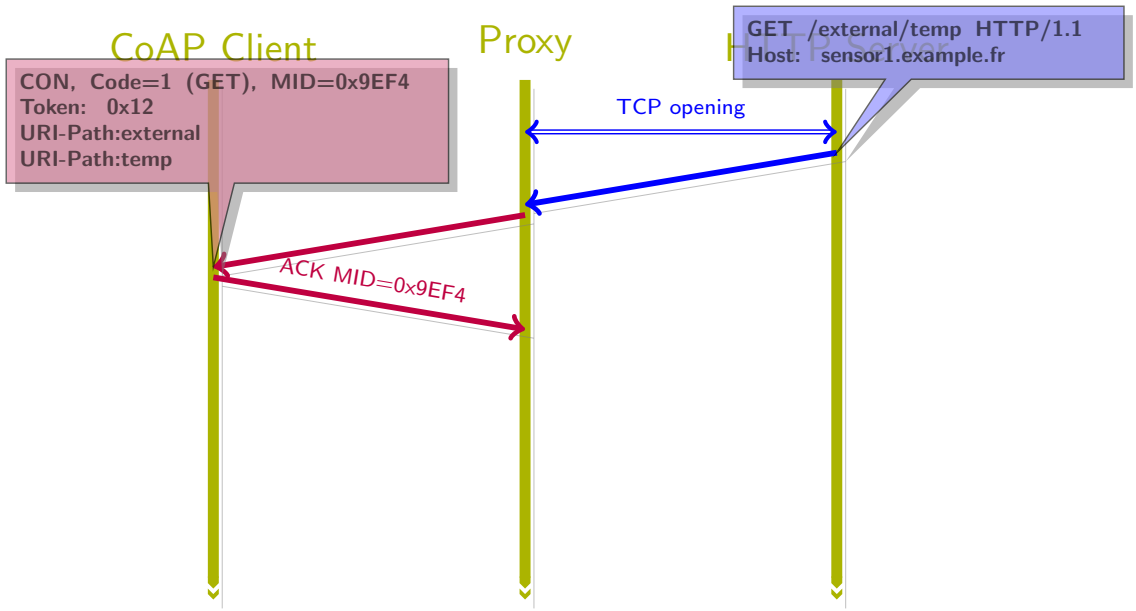


HTTP mapping



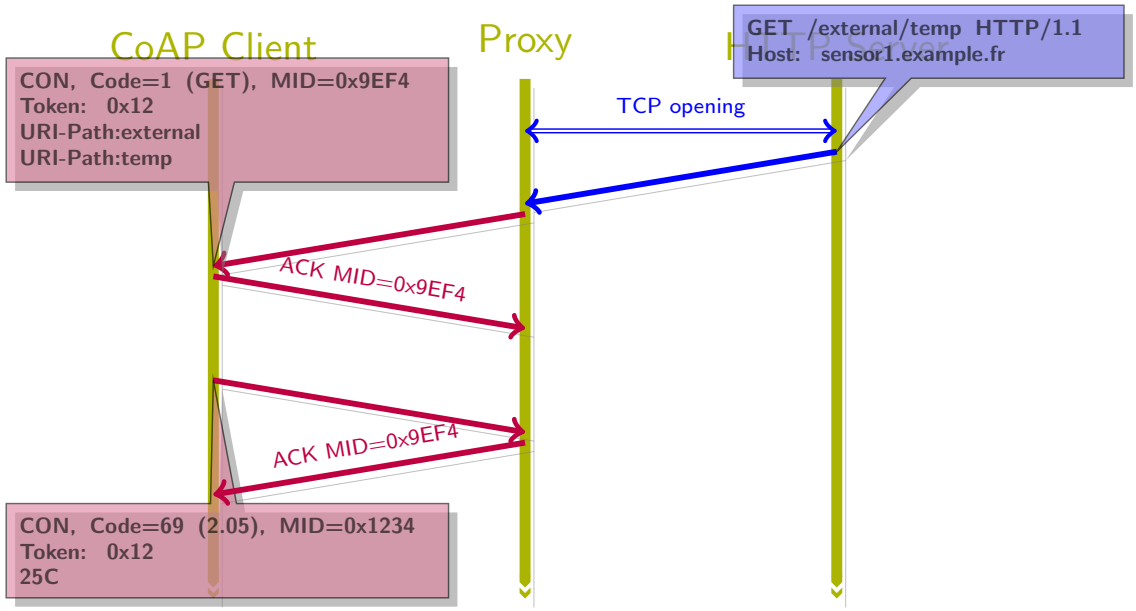
HTTP mapping

CoAP

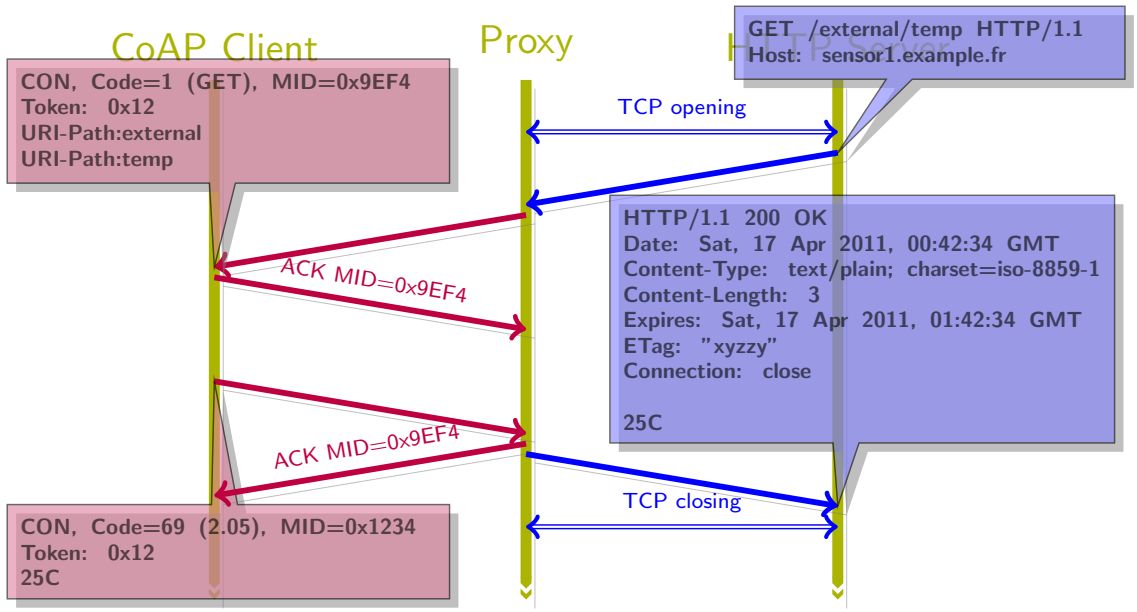


HTTP mapping

CoAP



HTTP mapping



Resource Discovery

Resource Discovery

- Discover CoAP server and resources managed by them:
  - Use multicast and DNS-SD (Service Discovery)
  - Avoid multicast and servers describe their resources using a standard representation.
    - Known as Web Linking (RFC 5988)
    - Link between resources.
    - Do not confuse with IPv6 links
  - Well known URI: `.well-known/core` (cf. RFC 5785)
    - `.well-known`: defined by RFC 5988
    - `core`: name of the working group
  - Response is structured following RFC 6690

Grammar

```
 Link = link-value-list
link-value-list = [ link-value *[ "," link-value ]]
link-value = "<" URI-Reference ">" *( ";" link-param )
link-param = ( ( "rel" "=" relation-types )
/ ( "anchor" "=" DQUOTE URI-Reference DQUOTE )
/ ( "rev" "=" relation-types )
/ ( "hreflang" "=" Language-Tag )
/ ( "media" "=" ( MediaDesc
/ ( DQUOTE MediaDesc DQUOTE ) ) )
/ ( "title" "=" quoted-string )
/ ( "title*" "=" ext-value )
/ ( "type" "=" ( media-type / quoted-mt ) )
/ ( "rt" "=" relation-types )
/ ( "if" "=" relation-types )
/ ( "sz" "=" cardinal )
/ ( link-extension ) )
```

Grammar

```
 Link = link-value-list
link-value-list = [ link-value *[ "," link-value ]]
link-value = "<" URI-Reference ">" *( ";" link-param )
link-param = ( ( "rel" "=" relation-types )
/ ( "anchor" "=" DQUOTE URI-Reference DQUOTE )
```

```
 </sensors>;ct=40;title="Sensor Index",
</sensors/temp>;rt="temperature-c";if="sensor",
</sensors/light>;rt="light-lux";if="sensor",
<http://www.example.com/sensors/t123>;anchor="/sensors/temp"
;rel="describedby",
</t>;anchor="/sensors/temp";rel="alternate"
```

```
/ ( "title" "=" quoted-string )
/ ( "title*" "=" ext-value )
/ ( "type" "=" ( media-type / quoted-mt ) )
/ ( "rt" "=" relation-types )
/ ( "if" "=" relation-types )
/ ( "sz" "=" cardinal )
/ ( link-extension ) )
```


CoAP

Grammar

List of resources (URI) on the node


```
 Link = li
link-value-list = [ link-value *[ "," link-value ]]
link-value = "<" URI-Reference ">" *( ";" link-param )
link-param = ( ( "rel" "=" relation-types )
/ ( "anchor" "=" DQUOTE URI-Reference DQUOTE )
/ ( "rev" "=" relation-types )
/ ( "hreflang" "=" Language-Tag )
/ ( "media" "=" ( MediaDesc
/ ( DQUOTE MediaDesc DQUOTE ) ) )
/ ( "title" "=" quoted-string )
/ ( "title*" "=" ext-value )
/ ( "type" "=" ( media-type / quoted-mt ) )
/ ( "rt" "=" relation-types )
/ ( "if" "=" relation-types )
/ ( "sz" "=" cardinal )
/ ( link-extension ) )
```


Slide 145 Laurent Toutain Filière 2

Grammar

```
 Link = link-value-list
link-value-list = [ link-value *[ "," link-value ]]
link-value = "<" URI-Reference ">" *( ";" link-param )
link-param = ( ( "rel" "=" relation-types )
/ ( "anchor" "=" DQUOTE URI-Reference DQUOTE )
/ ( "rev" "=" relation-types )
/ ( "hreflang" "=" Language-Tag )
/ ( "media" "=" ( MediaDesc
/ ( DQUOTE MediaDesc DQUOTE ) ) )
/ ( "title" "=" quoted-string )
/ ( "title*" "=" ext-value )
/ ( "type" "=" ( media-type / quoted-mt ) )
/ ( "rt" "=" relation-types )
/ ( "if" "=" relation-types )
/ ( "sz" "=" cardinal )
/ ( link-extension ) )
```

RFC 5988

RFC 6690

CoAP

Grammar


```
 Link = link-value-list
link-value-list = [ link-value *[ "," link-value ]]
link-value = "<" URI-Reference ">" *( ";" link-param )
link-param = ( ( "rel" "=" relation-types )
/ ( "anchor" "=" DQUOTE URI-Reference DQUOTE )
/ ( "rev" "=" relation-types )
/ ( "hreflang" "=" Language-Tag )
/ ( "media" "=" ( MediaDesc
/ ( DQUOTE MediaDesc DQUOTE ) ) )
/ ( "title" "=" quoted-string )
/ ( "title*" "=" ext-value )
/ ( "type" "=" ( media-type / quoted-mt ) )
/ ( "rt" "=" relation-types )
/ ( "if" "=" relation-types )
/ ( "sz" "=" cardinal )
/ ( link-extension ) )
```


relationship between two resources

RFC 5988

RFC 6690

Slide 145 Laurent Toutain Filière 2

Grammar

```
Link = link-value-list
link-value-list = [ link-value *[ "," link-value ]]
link-value = "<" URI-Reference ">" *( ";" link-param )
link-param = ( ( "rel" "=" relation-types )
/ ( "anchor" "=" DQUOTE URI-Reference DQUOTE )
/ ( "rev" "=" relation-types )
/ ( "hreflang" "=" Language-Tag )
/ ( "media" "=" ( MediaDesc
/ ( DQUOTE MediaDesc DQUOTE ) ) )
/ ( "title" "=" quoted-string )
/ ( "title*" "=" ext-value )
/ ( "type" "=" ( media-type / quoted-mt ) )
/ ( "rt" "=" relation-types )
/ ( "if" "=" relation-types )
/ ( "sz" "=" cardinal )
/ ( link-extension ) )
```

RFC 5988

RFC 6690

relationship between two resources

```
</sensors>;ct=40;title="Sensor Index",
</sensors/temp>;rt="temperature-c";if="sensor",
</sensors/light>;rt="light-lux";if="sensor",
<http://www.example.com/sensors/t123>;anchor="/sensors/temp"
;rel="describedby",
</t>;anchor="/sensors/temp";rel="alternate"
```


CoAP

Grammar

```
Link = link-value-list
link-value-list = [ link-value *[ "," link-value ]]
link-value = "<" URI-Reference ">" *( ";" link-param )
link-param = ( ( "rel" "=" relation-types )
/ ( "anchor" "=" DQUOTE URI-Reference DQUOTE )
/ ( "rev" "=" relation-types )
/ ( "hreflang" "=" Language-Tag )
/ ( "media" "=" ( MediaDesc
/ ( DQUOTE MediaDesc DQUOTE ) ) )
/ ( "title" "=" quoted-string )
/ ( "title*" "=" ext-value )
/ ( "type" "=" ( media-type / quoted-mt ) )
/ ( "rt" "=" relation-types )
/ ( "if" "=" relation-types )
/ ( "sz" "=" cardinal )
/ ( link-extension ) )
```

RFC 5988

RFC 6690

Human readable name

Slide 145 Laurent Toutain Filière 2 TELECOM Bretagne

Grammar

```
Link = link-value-list
link-value-list = [ link-value *[ "," link-value ]]
link-value = "<" URI-Reference ">" *( ";" link-param )
link-param = ( ( "rel" "=" relation-types )
/ ( "anchor" "=" DQUOTE URI-Reference DQUOTE )
/ ( "title" "=" quoted-string )
/ ( "title*" "=" ext-value )
/ ( "type" "=" ( media-type / quoted-mt ) )
/ ( "rt" "=" relation-types )
/ ( "if" "=" relation-types )
/ ( "sz" "=" cardinal )
/ ( link-extension ) )
```

RFC 5988

RFC 6690

```
</sensors>;ct=40;title="Sensor Index",
</sensors/temp>;rt="temperature-c";if="sensor",
</sensors/light>;rt="light-lux";if="sensor",
<http://www.example.com/sensors/t123>;anchor="/sensors/temp"
;rel="describedby",
</t>;anchor="/sensors/temp";rel="alternate"
```

Human readable name

Grammar

```
Link = link-value-list
link-value-list = [ link-value *[ "," link-value ]]
link-value = "<" URI-Reference ">" *( ";" link-param )
link-param = ( ( "rel" "=" relation-types )
/ ( "anchor" "=" URI-Reference DQUOTE )
/ ( "rev" "=" relation-types )
/ ( "hreflang" "=" Language-Tag )
/ ( "media" "=" ( MediaDesc
/ ( DQUOTE MediaDesc DQUOTE ) ) )
/ ( "title" "=" quoted-string )
/ ( "title*" "=" ext-value )
/ ( "type" "=" ( media-type / quoted-mt ) )
/ ( "rt" "=" relation-types )
/ ( "if" "=" relation-types )
/ ( "sz" "=" cardinal )
/ ( link-extension ) )
```

RFC 5988

RFC 6690

Resource Type:
- application-specific semantic type like "outdoor-temperature",
- ipso.
- "ucum:" Unified Code for Units of Measure

http://unitsofmeasure.org/ucum.html

- a URI referencing a specific concept in an ontology like "http://sweet.jpl.nasa.gov/2.0/phys.owl#Temperature"

Grammar


```
Link = link-value-list
link-value-list =[ "," link-value ]]
link-value = "<" URI-Reference ">" *( ";" link-param )
link-param = ( ( "anchor" "=" URI-Reference DQUOTE )
/ ( "rev" "=" relation-types )
/ ( "hreflang" "=" quoted-mt ) )
/ ( "media" "=" ( MediaDesc / ( DQUOTE MediaDesc DQUOTE ) ) )
/ ( "title" "=" quoted-string )
/ ( "title*" "=" ext-value )
/ ( "type" "=" ( type / quoted-mt ) )
/ ( "rt" "=" relation-types )
/ ( "if" "=" relation-types )
/ ( "sz" "=" cardinal )
/ ( link-extension ) )
```

RFC 5988

RFC 6690

Resource Type:
- application-specific semantic type like "outdoor-temperature",
- ipso.
- "ucum:" Unified Code for Units of Measure

http://unitsofmeasure.org/ucum.html

- a URI referencing a specific concept in an ontology like "http://sweet.jpl.nasa.gov/2.0/phys.owl#Temperature"

interface Description:
- opaque string

TELECOM Bretagne

Grammar

CoAP

Link = link-value-list
link-value-list =["," link-value]]
link-value = "<" URI-Reference ">" *(";" link-param)
link-param = (("anchor" "=" URI-Reference DQUOTE)
/ ("rev" "=" relation-types)
/ ("hreflang" "=" quoted-mt))
/ ("media" "=" (MediaDesc / (DQUOTE MediaDesc DQUOTE)))
/ ("title" "=" quoted-string)
/ ("title*" "=" ext-value)
/ ("type" "=" (type / quoted-mt))
/ ("rt" "=" relation-types)
/ ("if" "=" relation-types)
/ ("sz" "=" cardinal)
/ (link-extension))

RFC 5988

RFC 6690

Resource Type:
- application-specific semantic type like "outdoor-temperature",
- ipso.
- "ucum:" Unified Code for Units of Measure

http://unitsofmeasure.org/ucum.html

- a URI referencing a specific concept in an ontology like "http://sweet.jpl.nasa.gov/2.0/phys.owl#Temperature"

interface Description:
- opaque string

Approximative size

Slide 145 Laurent Toutain Filière 2

TELECOM Bretagne

IPSO Application framework

Function Set	Root Path	Resouce Type
Device	/dev	ipso.dev
General Purpose IO	/gpio	ipso.gpio
Power	/pwr	ipso.pwr
Load Control	/load	ipso.load
Sensors	/sen	ipso.sen
Light Control	/lt	ipso.lt
Message	/msg	ipso.msg
Location	/loc	ipso.loc
Configuration	/cfg	ipso.cfg

IPSO Application framework: Device

Type	Path	RT	IF	Type	Unit
Manufacturer	/dev/mfg	ipso.dev.mfg	rp	s	
Model	/dev/mdl	ipso.dev.mdl	rp	s	
Hardware Revision	/dev/mdl/hw	ipso.dev.mdl.hw	rp	s	
Software Version	/dev/md/sw	ipso.dev.mdl.sw	rp	s	
Serial	/dev/ser	ipso.dev.ser	rp	s	
Name	/dev/n	ipso.dev.n	p, rp	s	
Power Supply	/dev/pwr/#	ipso.dev.pwr	rp	e [0, 1, 2]	
Power Supply Voltage	/dev/pwr/#/v	ipso.dev.pwr.v	s	d	V
Time	/dev/time	ipso.dev.time	p, rp	i	s
Uptime	/dev/uptime	ipso.dev.uptime	s	i	s

IPSO Application framework: Sensor

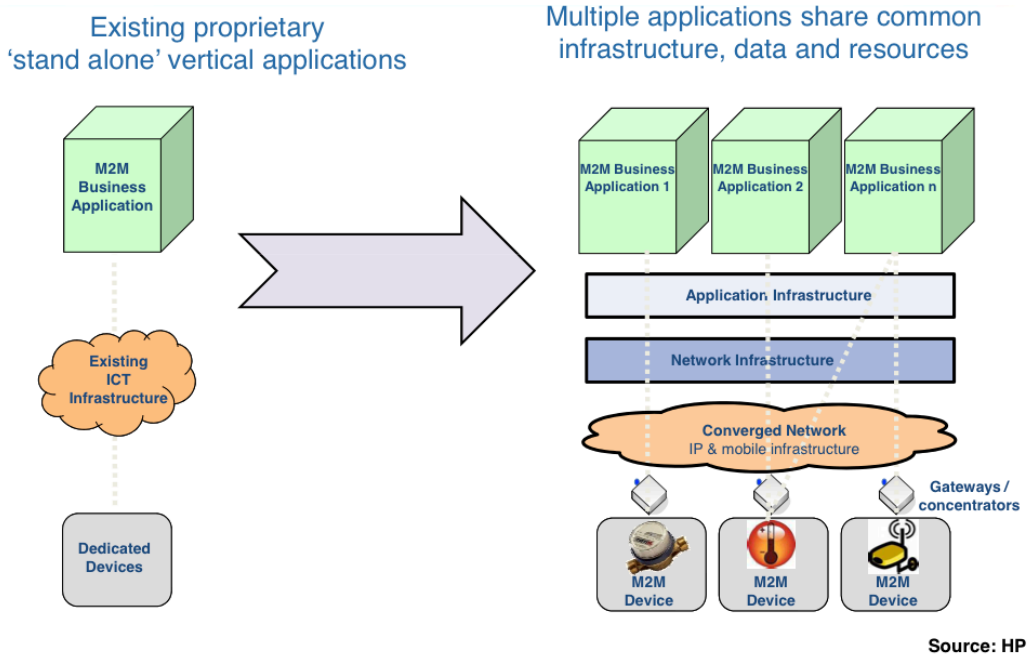
Type	Path	RT	IF	Type	Unit
Motion	/sen/#	ipso.sen.mot	s	i	
Motion Status	/sen/#/status	ipso.sen.mot.status	s	b	
Contact	/sen/#	ipso.sen.con	s	i	
Contact Status	/sen/#/status	ipso.sen.con.status	s	b	
Generic	/sen/#	ucum.unit	s	d	

Data representation

- plain/text
- application/senML+JSON

```
{"e":[
"n": "voltage", "t": 0, "u": "V", "v": 120.1 ,
"n": "current", "t": 0, "u": "A", "v": 1.2 ],
"bn": "urn:dev:mac:0024befffe804ff1/"
}
```

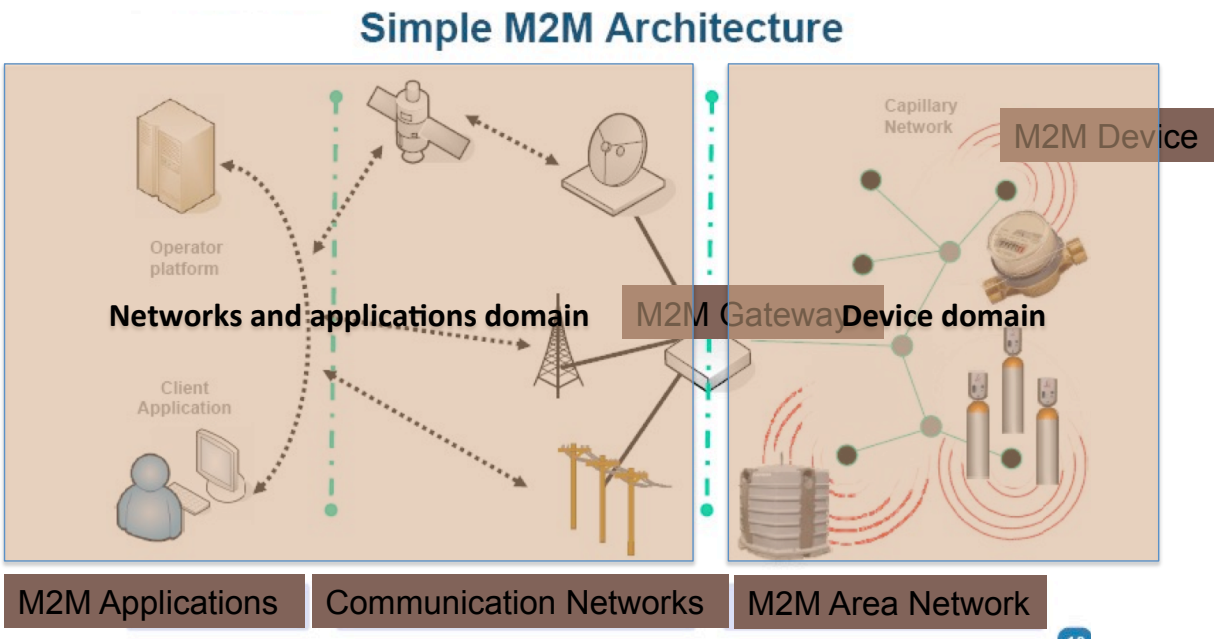
Inverting the pipes

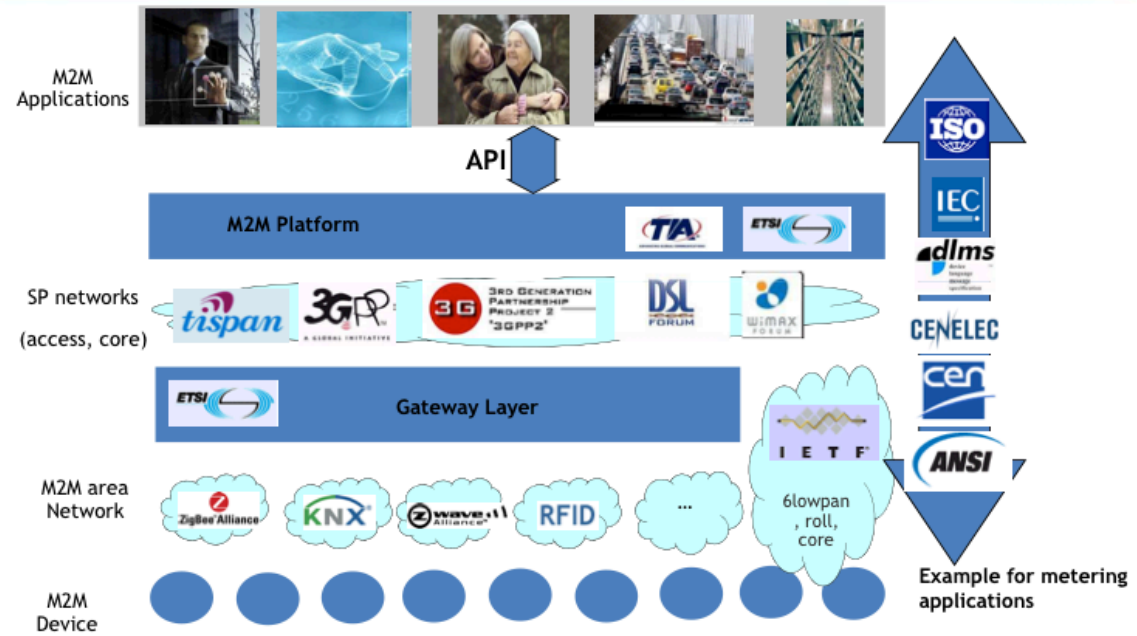


Outline

- ETSI M2M - introduction
  - Architecture
  - Terminology
    - How are things named (objects, APIs, interfaces)
  - Example framework mapping
- ETSI M2M – use case with lamps
  - Example

ETSI M2M – entities and domains

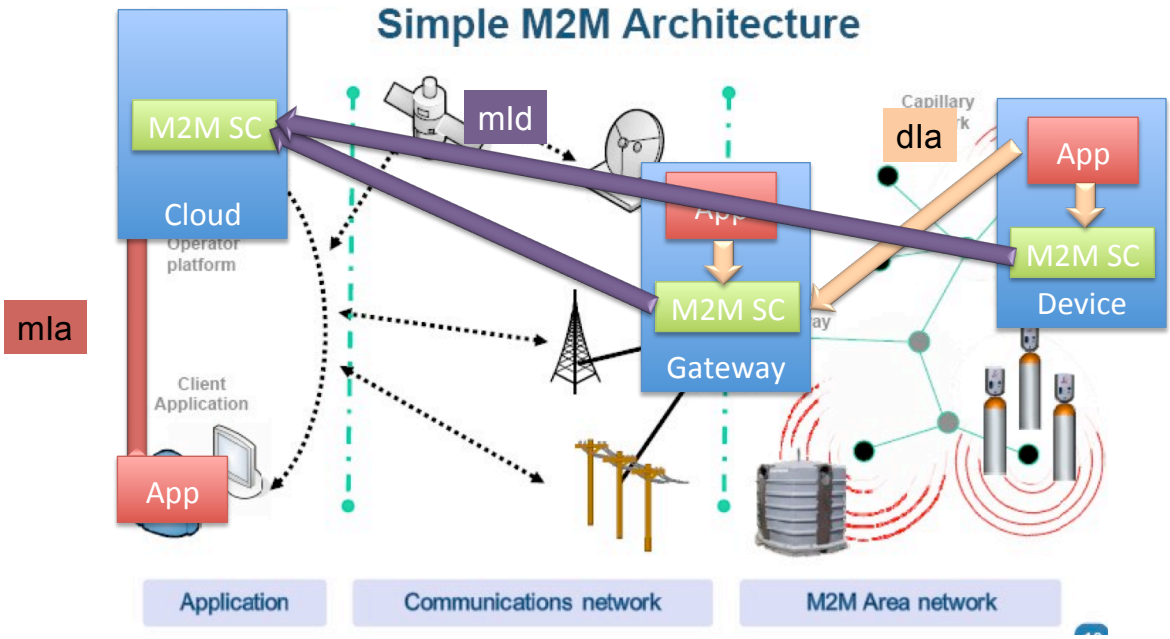




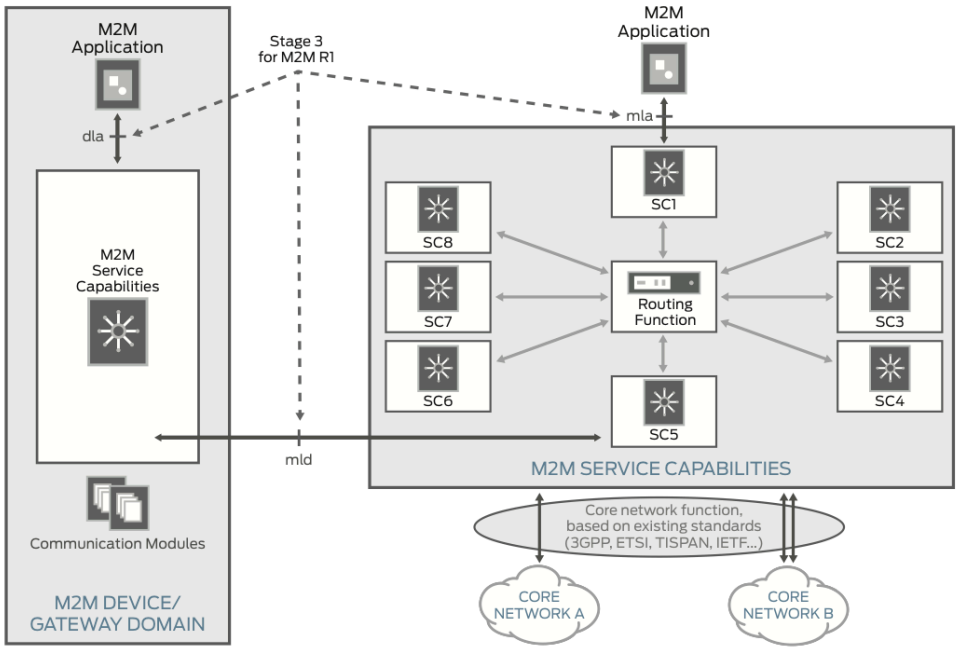
Source: ALU

Prefixes = **G**ateway, **D**evice or **N**etwork
application
middleware = Cloud = …
device
SC = Service Capabilities (think API)

ETSI M2M – reference points



ETSI M2M SC functional architecture framework

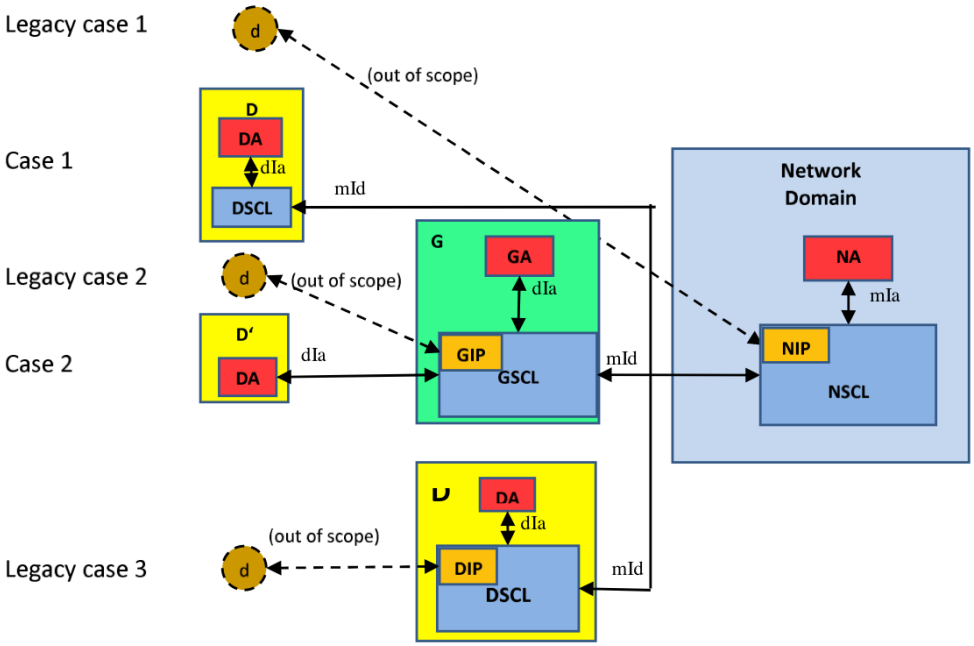


Service Capabilities (SCs)

x can stand for: N for network, G for gateway or D for device

- Application enablement (**xAE**) - provides a single API interface to applications
- Generic communication (**xGC**) - manages all aspects pertaining to secure-transport session establishment and teardown, as well as interfacing with bearer services provided by the core network
- Reachability, addressing, and repository (**xRAR**) - provides storage capability for state associated to applications, devices, and gateways and handles subscriptions to data changes
- Communication selection (**xCS**) provides network and network bearer selection for devices or gateways that are reachable via multiple networks or multiple connectivity bearers for example, WiFi, or GPRS.
- Remote entity management (**xREM**) - provides functions pertaining to device/gateway lifecycle management, such as software and firmware upgrade and fault and performance management
- Security (**xSEC**) implements bootstrapping, authentication, authorization, and key management. Interfaces with an M2M authentication server (MAS) to obtain authentication data.
- History and data retention (**xHDR**) - stores records pertaining to the usage of the M2M SCs. xHRD may be used for law enforcement purposes such as privacy
- Transaction management (**xTM**) - manages transactions
- Compensation broker (**xCB**) - manages compensation transaction
- Telco operator exposure (**xTOE**) - provides access, via the same API used to access the SCs, to traditional network operator services, such as SMS, MMS, and location.
- Interworking proxy (**xIP**) - allows a non-ETSI-compliant device to interwork with the ETSI standard

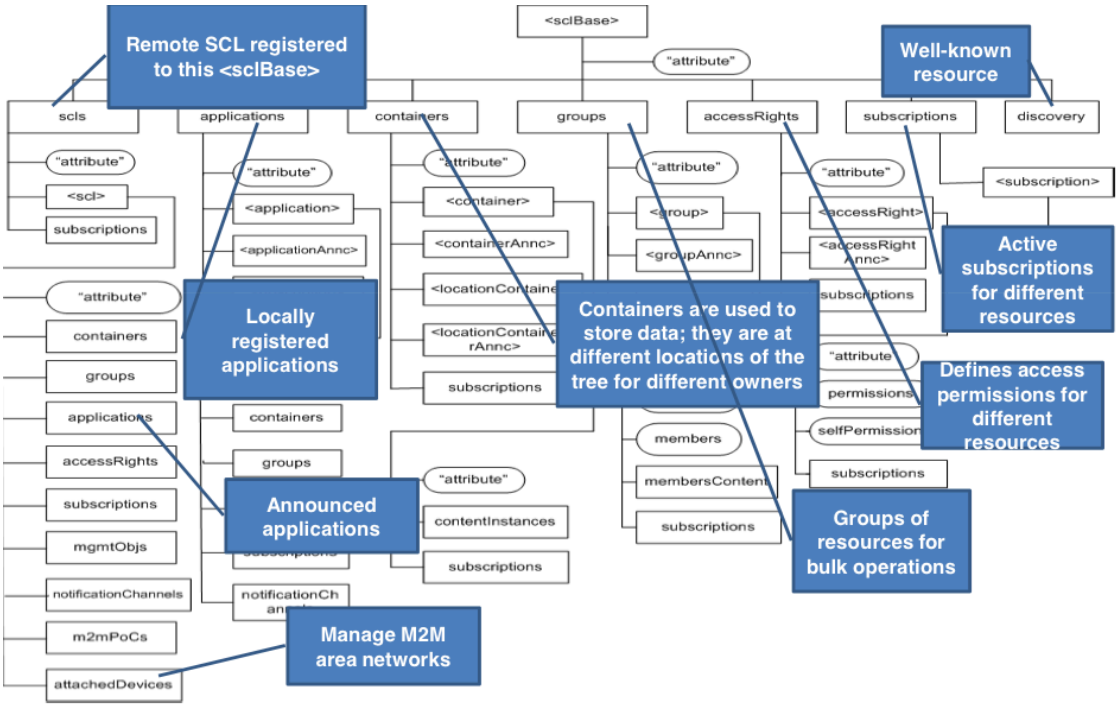
Mapping reference points to different deployment scenarios



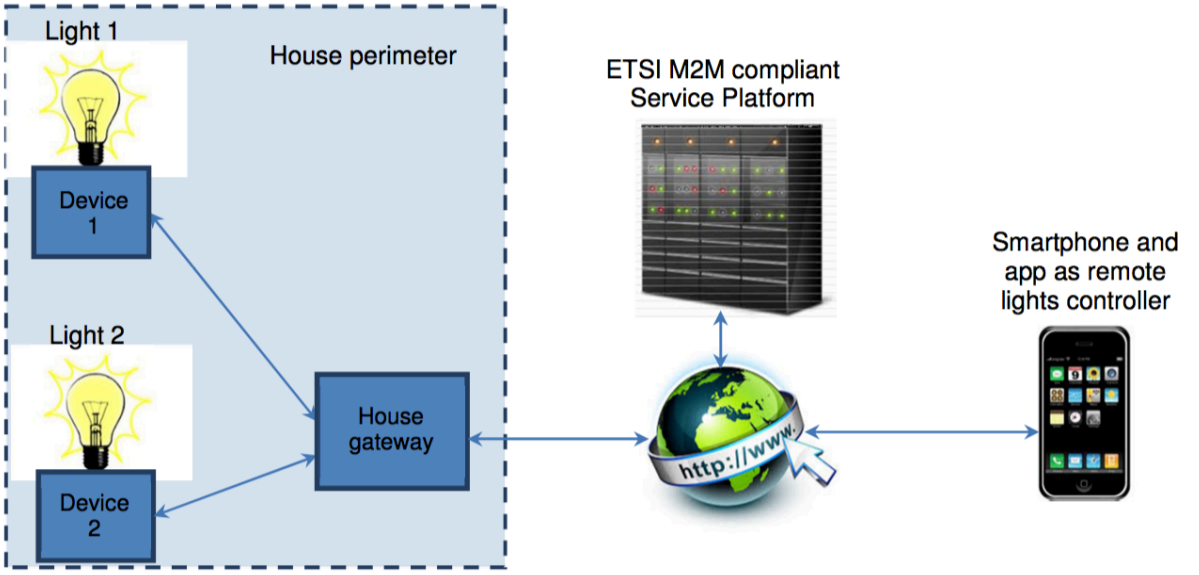
What about REST?

- ETSI M2M is RESTful
  - All “things” are resources manipulated with 4 simple actions – GET, PUT, POST, DELETE
  - A resource is *identified* uniquely via an URI
    - URI = string of type *schema:blahblahblah*
  - A resource has a *representation* of a valid Internet media type (e.g. JSON, XML)
  - All metadata is hypertext
- Think of every SCL as a book
  - It can be edited, and the only thing it can contain are “simple” values and references to other books
  - Only its not linear (pages are hierarchical – URI)

Resource tree overview



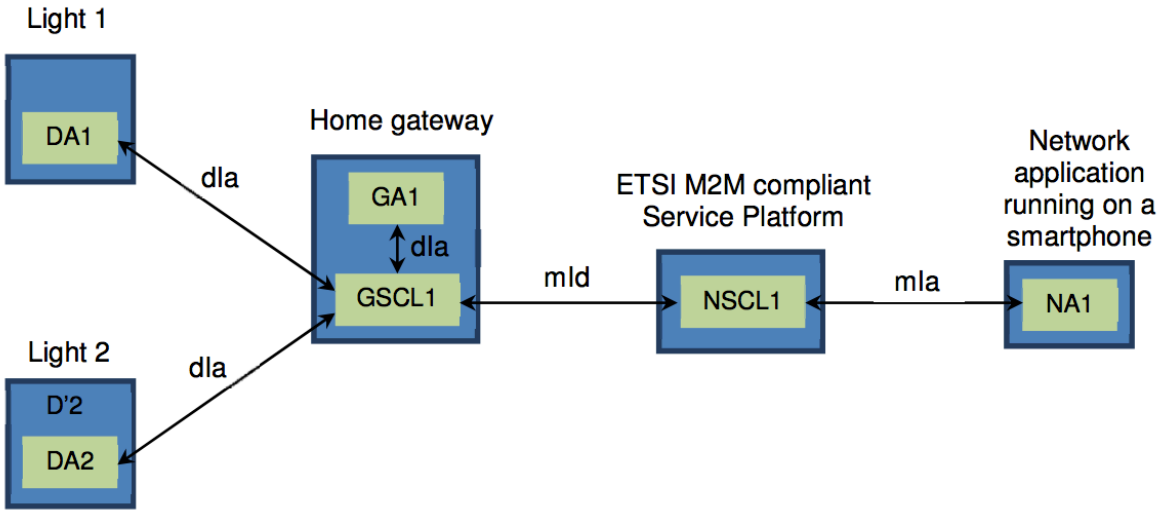
Use case – Lights



Source:

ETSI M2M application developers guideline

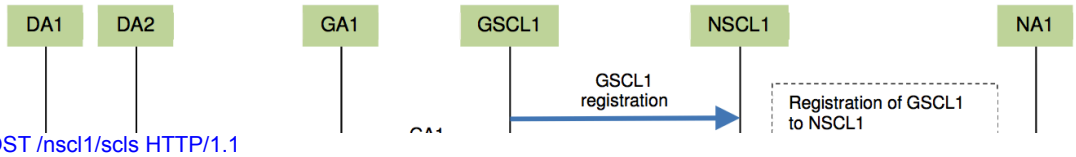
Mapping to ETSI M2M



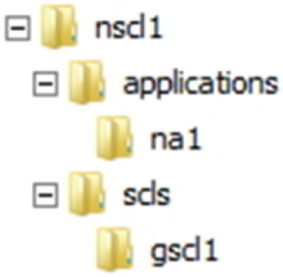
Reference points

- mIa
  - allows an application to access the M2M SCs in the network and applications domains
- dIa
  - allows an application residing in an M2M device to access the different M2M SCs in the same M2M device or in an M2M gateway
  - allows an application residing in an M2M gateway to access the different M2M SCs in the same M2M gateway
- mId
  - allows an M2M device or M2M gateway to communicate with the M2M SCs in the network and applications domain and vice-versa

1. Registration

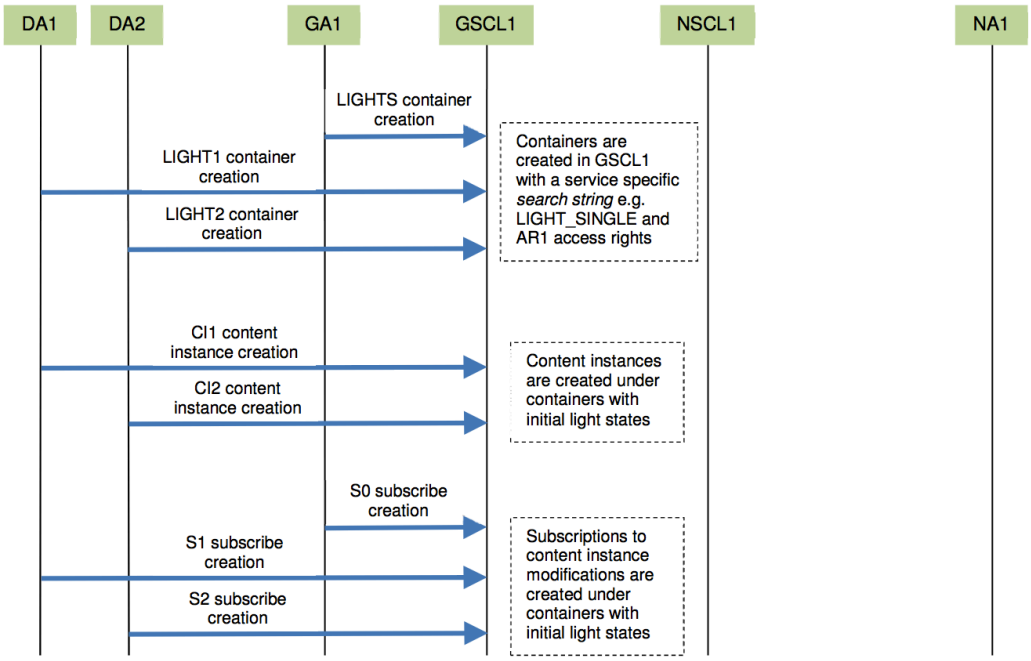



```
POST /nscl1/scls HTTP/1.1
Host: http://m2m.provider1.com:9000
From: gscl1
Content-Type: application/xml
Content-Length: 247

<xml>
<scl sclId="gscl1">
<expirationTime>2014-05-10T12:02:45.000+02:00</expirationTime> <searchStrings>
   <searchString>LIGHT_GSCL</searchString>
</searchStrings>
<link>http://m2m.provider2.com:9000/gscl1</link>
</scl>
</xml>

201 Created
Content-Location: http://m2m.provider1.com:9000/nscl1/scls/gscl1 Content
Content-Length: 0
```



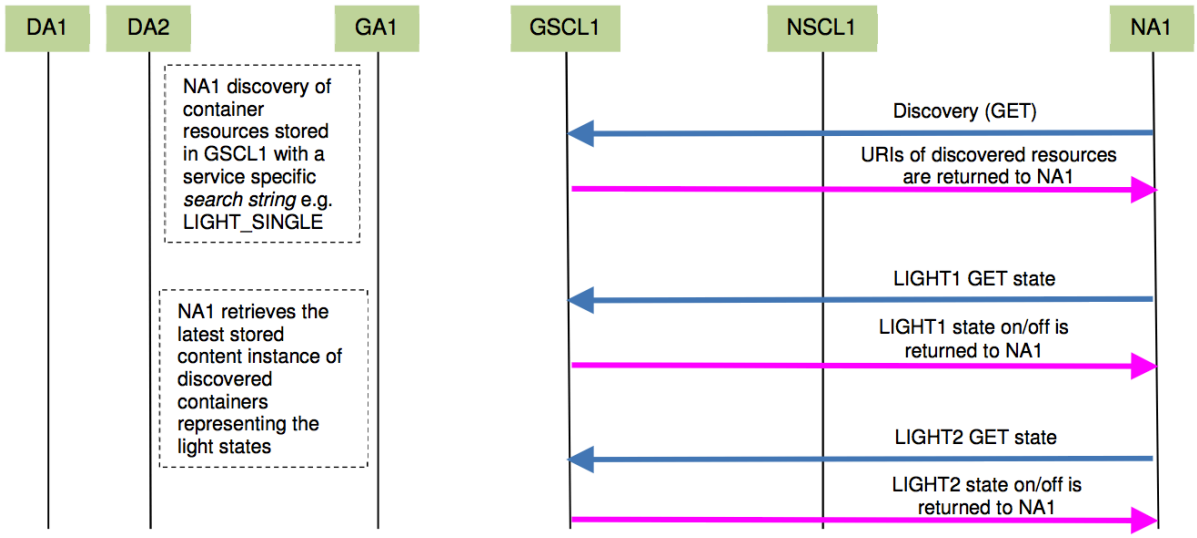
rights creation

2. Initial resource creation

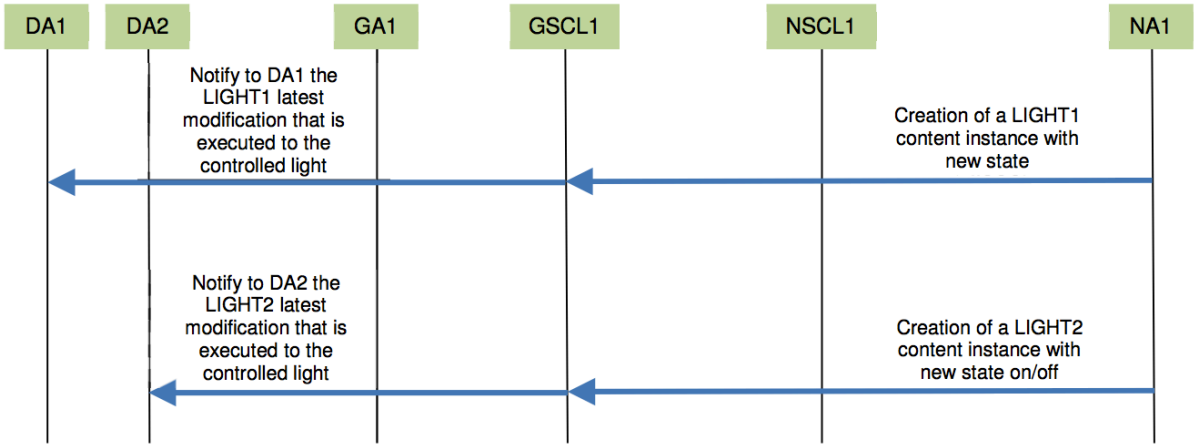


Resource tree for GSCL1 under sclBase/

3. Discovery and get state phase



4. Changing the state



Turning on a light from the phone

```
POST /gscl1/applications/da1/containers/LIGHT1/contentInstances HTTP/1.1
Host: http://m2m.provider2.com:9000
From: http://m2m.provider1.com:9000/applications/na1
Content-Type: application/xml
Content-Length: 117
<xml>
 <contentInstance>
<content> <textContent>ON</textContent>
  </content>
 </contentInstance>
</xml>

201 Created
Content-Location: http://m2m.provider2.com:9000/gscl1/applications/da1/
containers/LIGHT1/contentInstances/CI3 Content-Type: application/xml
Content-Length: 0
```


Reading the state of a light

```
GET /gscl1/applications/da1/containers/LIGHT1/contentInstances/latest HTTP/1.1
Host: http://m2m.provider2.com:9000
From: http://m2m.provider2.com:9000/applications/ga1
Accept: application/xml

200 Ok
Content-Type: application/xml
Content-Length: 458
<xml>
 <contentInstance id="http://m2m.provider2.com:9000/gscl1/applications/da1/
containers/LIGHT1/contentInstances/CI2">
<creationTime>2013-05-12T12:03:30.000+02:00</creationTime>
<lastModifiedTime>2013-05-12T12:03:30.000+02:00</lastModifiedTime>
<contentTypes>
<contentType>text/plain</contentType> </contentTypes> <contentSize>62</
contentSize> <content>
<textContent>OFF</textContent> </content>
 </contentInstance>
</xml>
```