IPv6 Training

KENIC-AFRINIC IPv6 Workshop

17th – 20th June 2008

César Olvera (cesar.olvera@consulintel.es)
Jordi Palet (jordi.palet@consulintel.es)
Alvaro Vives (alvaro.vives@consulintel.es)

Agenda

1. Basic Introduction to IPv6
2. Header Formats & Packet Size Issues
3. Addressing
4. ICMPv6, Neighbor Discovery & DHCPv6
5. Applications
6. IPv6 DNS
7. Security/firewalling
8. Transition and Coexistence
9. Mobility
10. Routing

IPv6 Tutorial

1. Basic Introduction to IPv6

deploy

ConsultinTel
Consultores Integrales en Telecomunicaciones

Agenda

1.1. IPv6 history
1.2. Benefits of IPv6

1.1. IPv6 history

Why a New IP?

Only *compelling* reason: more addresses!

- for billions of new devices,
  e.g., cell phones, PDAs, appliances, cars, etc.
- for billions of new users,
  e.g., in China, India, etc.
- for “always-on” access technologies,
  e.g., xDSL, cable, ethernet-to-the-home, etc.

IPng Requirements

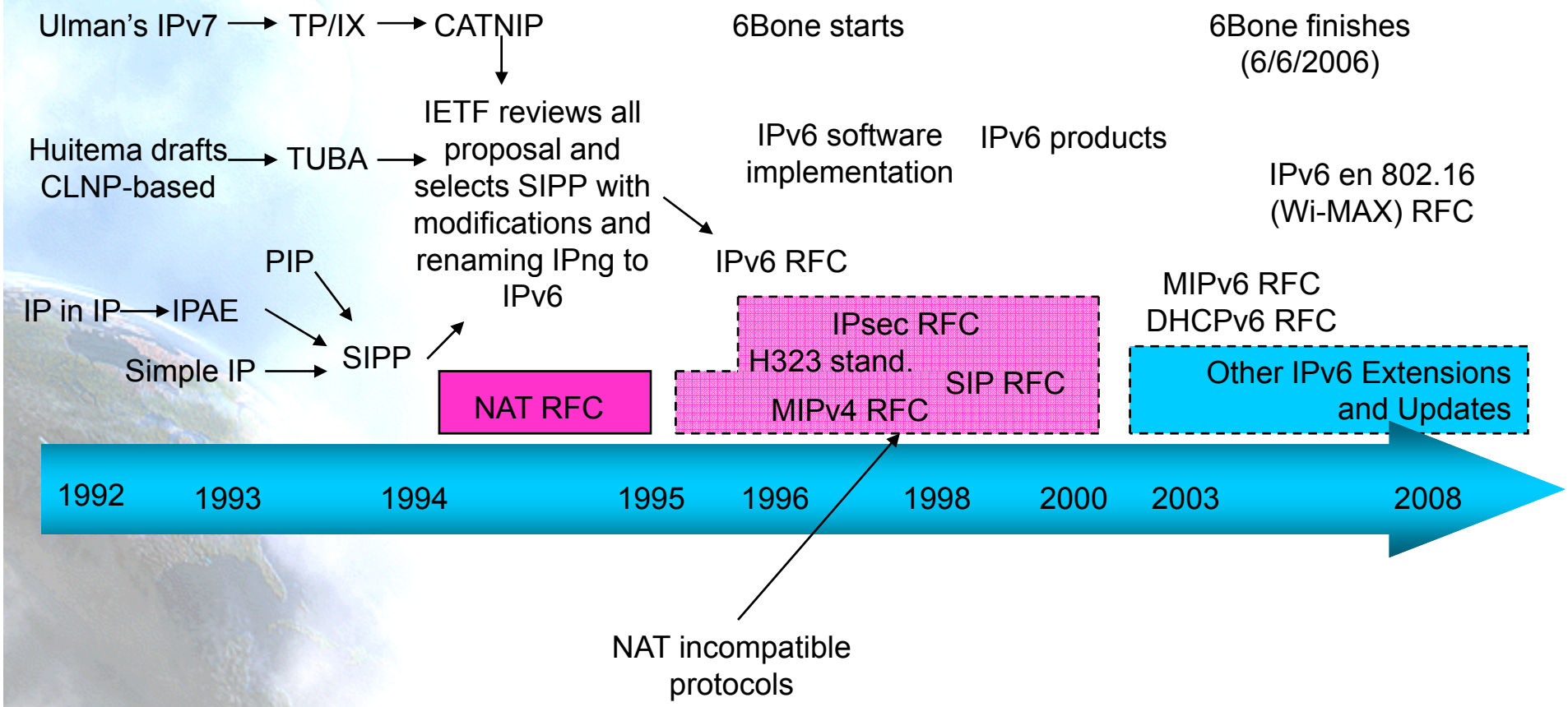
- November, 1991
  - IETF formed a working group to examine the issue and to consider possible solutions
- July, 1992
  - IETF determined that it was essential to begin to create a next-generation Internet Protocol (IPng)

- IPng had to work out both issues
  - Support for large address space
  - Support for addressing schemes based on aggregating hierarchies
- But also new requirements were stated to improve the IPv4 deficiencies
  - Security (both authentication and encryption)
  - Plug-and-play network autoconfiguration
  - Improve QoS facilities
  - Support for mobility

Candidates for IPng

- Creation and selection of the new protocol is under the IETF umbrella
- Between 1992 and 1994 there were seven candidatures, but by spring 1994 only three remained
  - CATNIP (Common Architecture for the Internet)
    - Designed as a "convergence protocol," integrating IP, Novell's IPX, and the network layer protocol of the OSI suite.
  - SIPP (Simple Internet Protocol Plus)
    - An evolution from the current IP (IPv4) and interoperable with it
  - TUBA (TCP and UDP with Bigger Addresses)
    - A proposal to adopt the OSI network layer (CLNP) as the new Internet network layer
- By July, 1994 the IETF selected SIPP as protocol that should become IPng
  - The SIPP documents were the basis for working on IPng
  - The SIPP working group disappeared to be integrated into the IPng working group
- Key aspects of SIPP to be chosen
  - Transition aspects from IPv4 to IPng
    - Long period with both IPv4 and IPng protocols coexisting
    - Some nodes never will upgrade to IPng
    - New IPng nodes could used old IPv4-only network to transport IPng packets (tunneling)
    - No need for a flag-day
- Later on, the IPng working group was officially renamed as IPv6 working group

IPng Time-line



But Isn't There Still Lots of IPv4 Address Space Left?

- ~ Half the IPv4 space is unallocated
  - if size of Internet is doubling each year, does this mean only one year's worth?!
- No, because today we deny unique IPv4 addresses to most new hosts
  - we make them use methods like NAT, PPP, etc. to share addresses
- But new types of applications and new types of access need unique addresses!

Why Are NAT's Not Adequate?

- They won't work for large numbers of "servers", i.e., devices that are "called" by others (e.g., IP phones)
- They inhibit deployment of new applications and services
- They compromise the performance, robustness, security, and manageability of the Internet

Why Was 128 Bits Chosen as the IPv6 Address Size?

- Some wanted fixed-length, 64-bit addresses
  - easily good for 10^{12} sites, 10^{15} nodes, at .0001 allocation efficiency (3 orders of mag. more than IPng requirement)
  - minimizes growth of per-packet header overhead
  - efficient for software processing
- Some wanted variable-length, up to 160 bits
  - compatible with OSI NSAP addressing plans
  - big enough for autoconfiguration using IEEE 802 addresses
  - could start with addresses shorter than 64 bits & grow later
- Settled on fixed-length, 128-bit addresses
  - (340,282,366,920,938,463,463,374,607,431,768,211,456 in all!)

What Ever Happened to IPv5?

0–3		unassigned
4	IPv4	(today's widespread version of IP)
5	ST	(Stream Protocol, not a new IP)
6	IPv6	(formerly SIP, SIPP)
7	CATNIP	(formerly IPv7, TP/IX; deprecated)
8	PIP	(deprecated)
9	TUBA	(deprecated)
10-15		unassigned

1.2. Benefits of IPv6

Incidental Benefits of Bigger Addresses

- Easy address auto-configuration
- Easier address management/delegation
- Room for more levels of hierarchy and for route aggregation
- Ability to do end-to-end IPsec (because NATs not needed)

Incidental Benefits of New Deployment

- Chance to eliminate some complexity, e.g., in IP header
- Chance to upgrade functionality, e.g., multicast, QoS, mobility
- Chance to include new enabling features, e.g., binding updates

Summary of Main IPv6 Benefits

- Expanded addressing capabilities
- Server-less autoconfiguration (“plug-n-play”) and reconfiguration
- More efficient and robust mobility mechanisms
- Built-in, strong IP-layer encryption and authentication
- Streamlined header format and flow identification
- Improved support for options / extensions

IPv6 Tutorial

2. Header Formats & Packet Size Issues

Agenda

2.1. Terminology
2.2. IPv6 Header Format
2.3. Packet Size Issues
2.4. Upper-Layer Protocol Issues
2.5. Jumbograms

2.1. Terminology

RFC2460

- Internet Protocol, Version 6: Specification
- Changes from IPv4 to IPv6:
  - Expanded Addressing Capabilities
  - Header Format Simplification
  - Improved Support for Extensions and Options
  - Flow Labeling Capability
  - Authentication and Privacy Capabilities

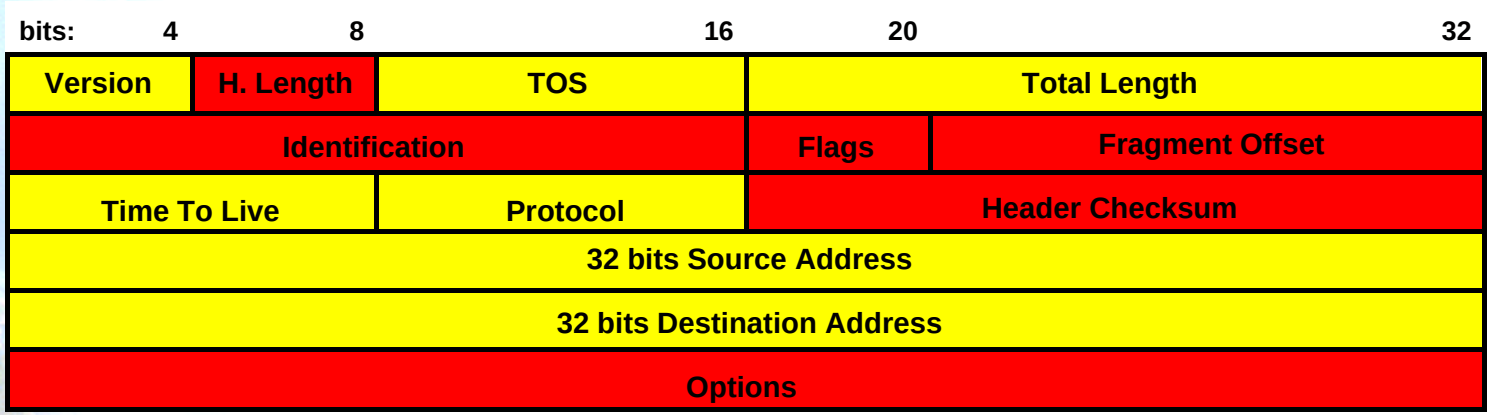
Terminology

- **Node:** Device that implements IPv6
- **Router:** Node that forwards IPv6 packets
- **Host:** Any node that isn't a router
- **Upper Layer:** Protocol layer immediately above IPv6
- **Link:** Communication Facility or Medium over with nodes can communicate at the link layer
- **Neighbors:** Nodes attached to the same link
- **Interface:** A node´s attachment to a link
- **Address:** An IPv6-layer identification for an interface or a set of interfaces
- **Packet:** An IPv6 header plus payload
- **Link MTU:** Maximum Transmission Unit
- **Path MTU:** Minimum link MTU of all the links in a path between source and destination node's

2.2. IPv6 Header Format

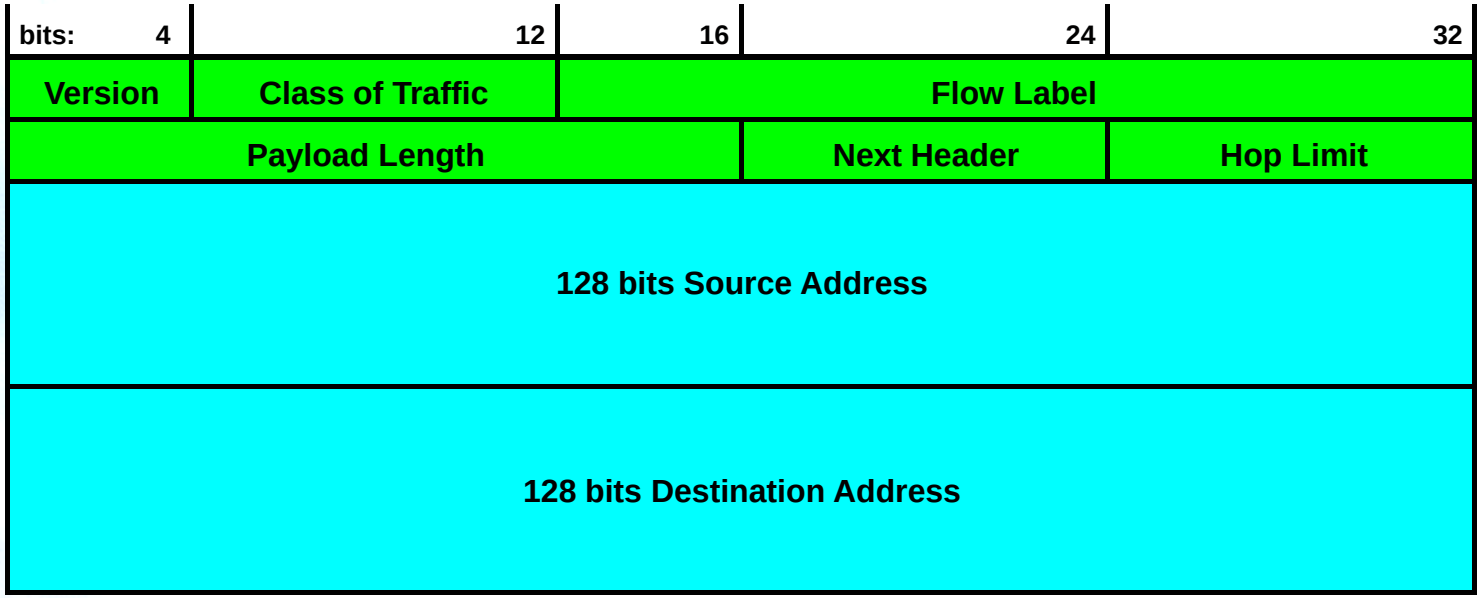
IPv4 Header Format

- 20 Bytes + Options (Max. 40 Bytes)
  - Variable Length: from 20 to 60 Bytes



IPv6 Header Format

- From 12 to 8 Fields (40 bytes)



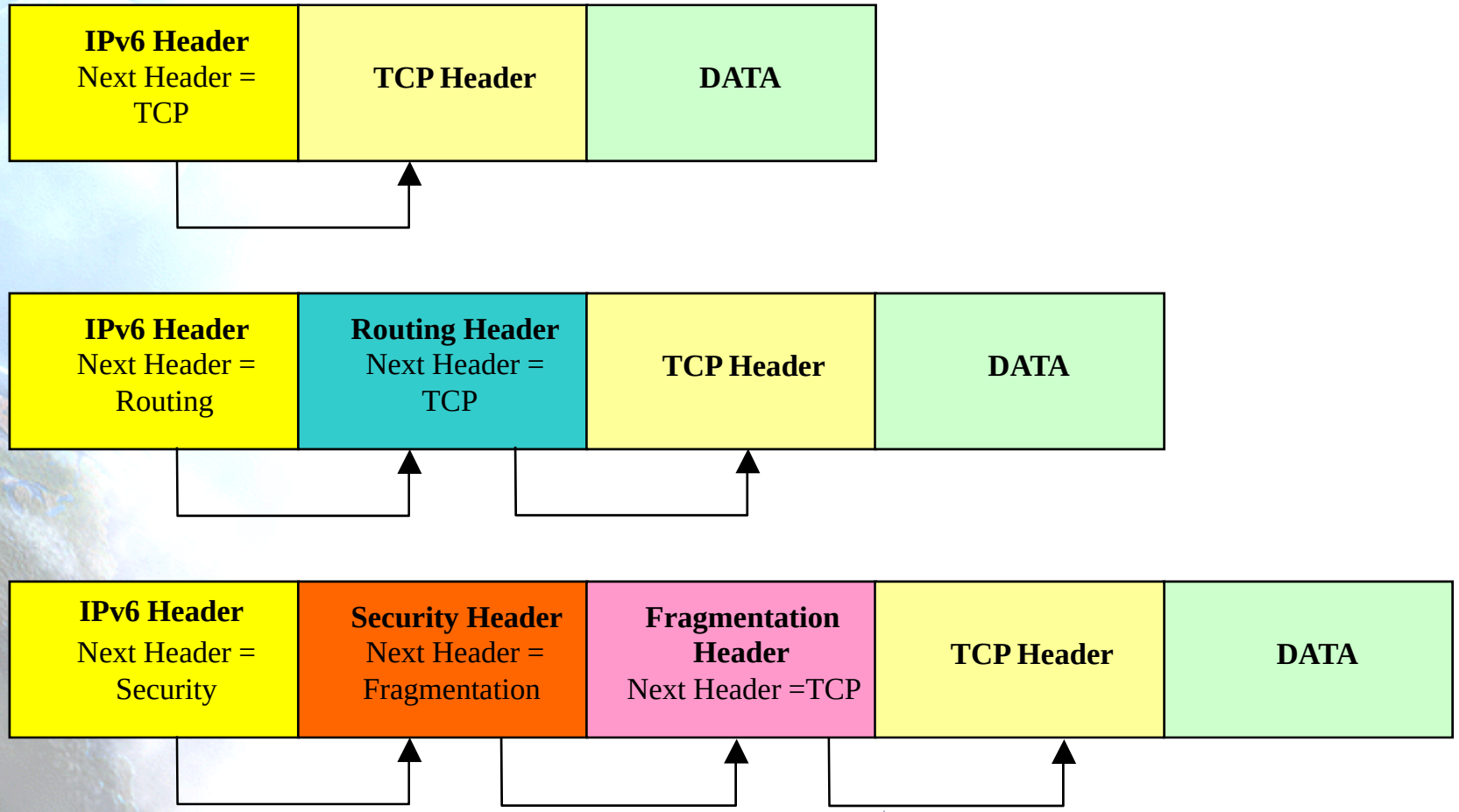
  - Avoid checksum redundancy
  - Fragmentation end to end

Summary of Header Changes

- 40 bytes
- Address increased from 32 to 128 bits
- Fragmentation and options fields removed from base header
- Header checksum removed
- Header length is only payload (because fixed length header)
  - Include length count of present extension headers
- New Flow Label field
- TOS -> Traffic Class
- Protocol -> Next Header (extension headers)
- Time To Live -> Hop Limit
- Alignment changed to 64 bits

Extension Headers

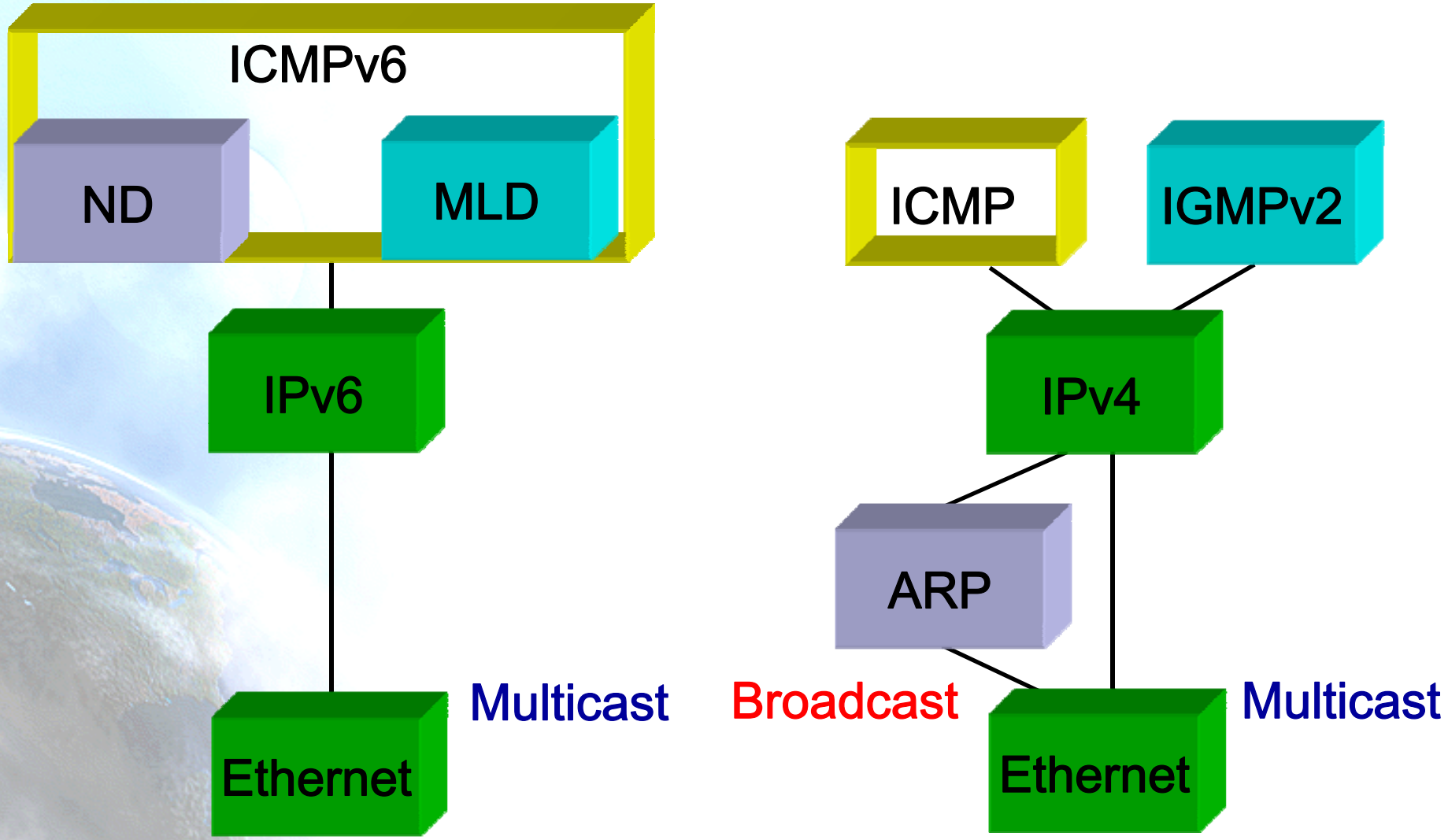
- "Next Header" Field



Extension Headers Goodies

- Processed Only by Destination Node
  - Exception: Hop-by-Hop Options Header
- No more “40 byte limit” on options (IPv4)
- Extension Headers defined currently (to be used in the following order):
  - Hop-by-Hop Options (0)
  - Destination Options (60) / Routing (43)
  - Fragment (44)
  - Authentication (RFC4302, next header = 51)
  - Encapsulating Security Payload (RFC4303, next header = 50)
  - Destination Options (60)
  - Mobility Header (135)
  - No next header (59)
    - TCP (6), UDP (17), ICMPv6 (58)

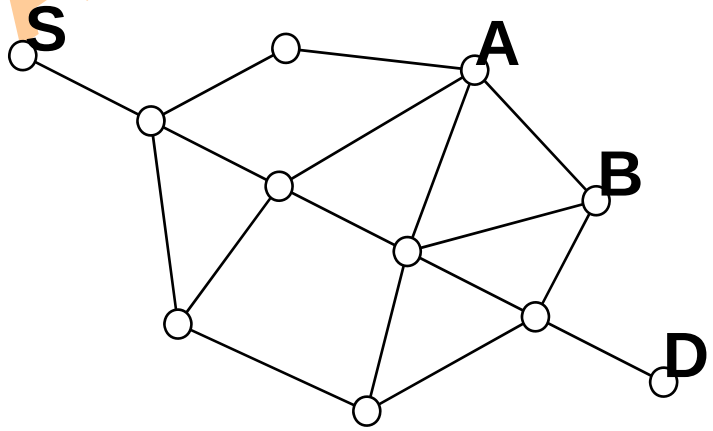
Control Plane IPv4 vs. IPv6



Example: Using the Routing Header

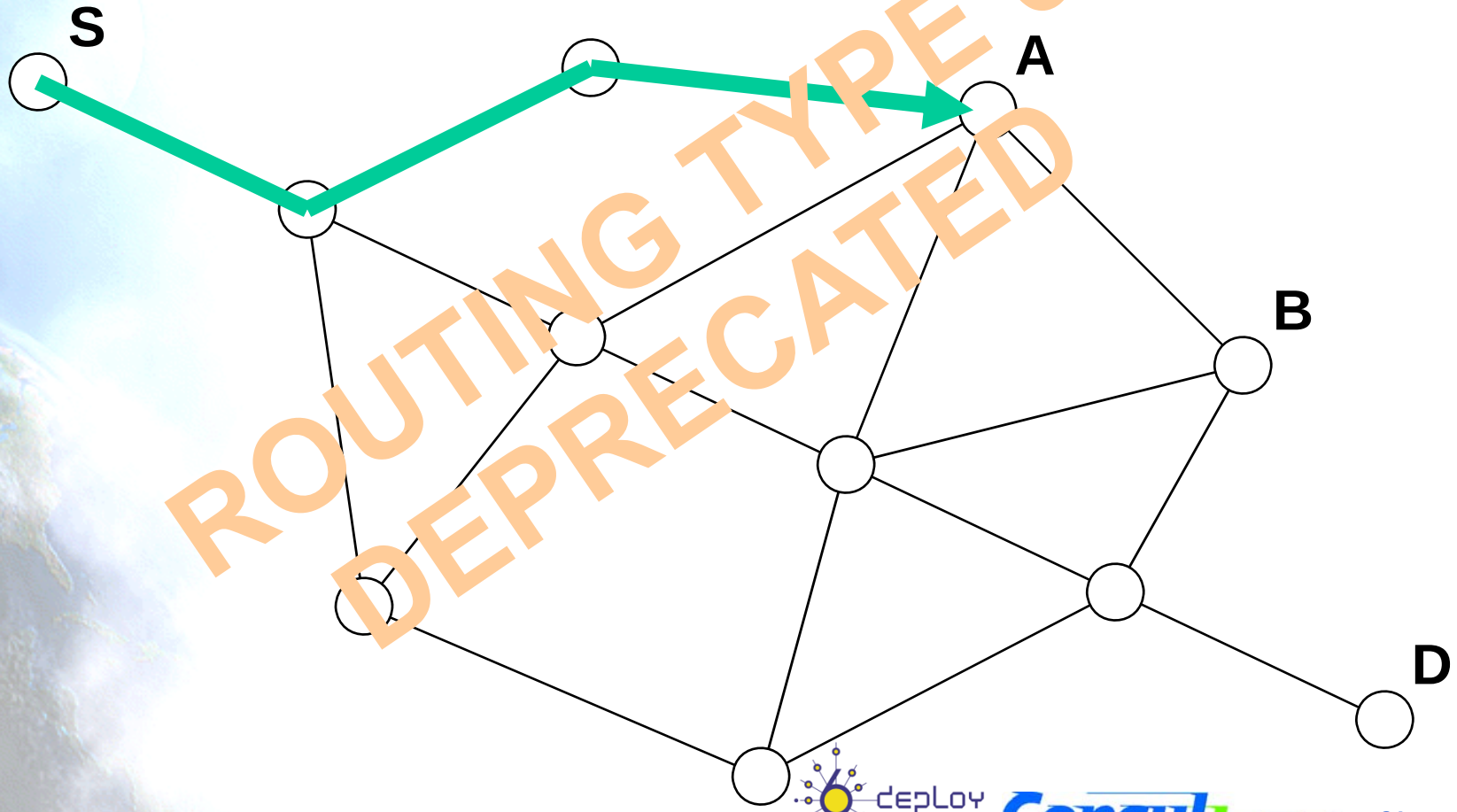
8 bits	8 bits unsigned	8 bits	8 bits unsigned
Next Header	**H. Ext. Length**	**Routing Type = 0**	**Segments Left**
Reserved = 0			
Address 1			
Address 2			
…			
Address n			

- Next Header value = 43
- A type 0 routing header, where:
  - Source Node: S
  - Destination: D
  - Intermediate Nodes: A & B



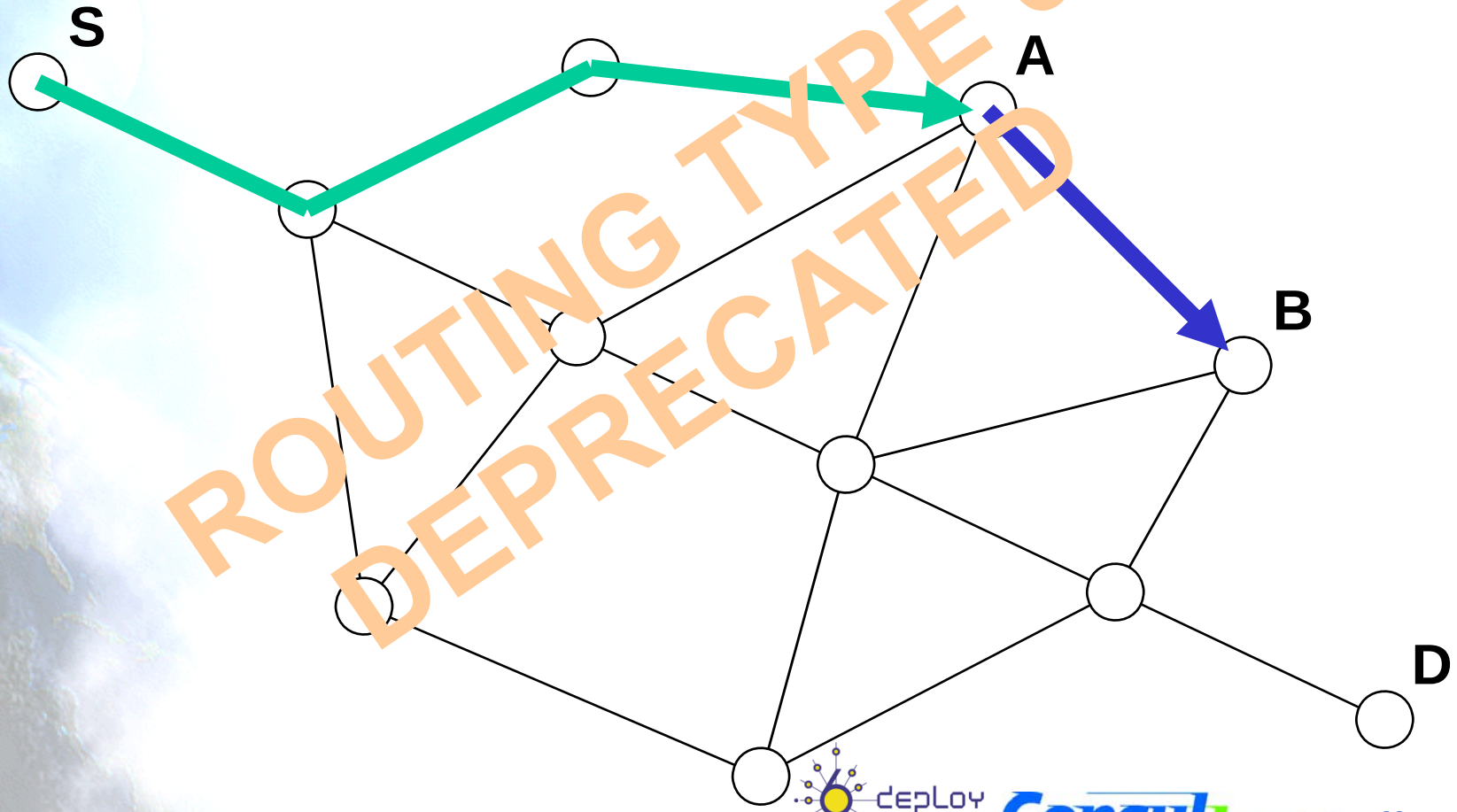
Example: Headers when S to A

IPv6 Base Header	Routing Header
Source Address = S Destination Address = A	H. Ext. Length = 4 Segments Left = 2 Address 1 = B Address 2 = D



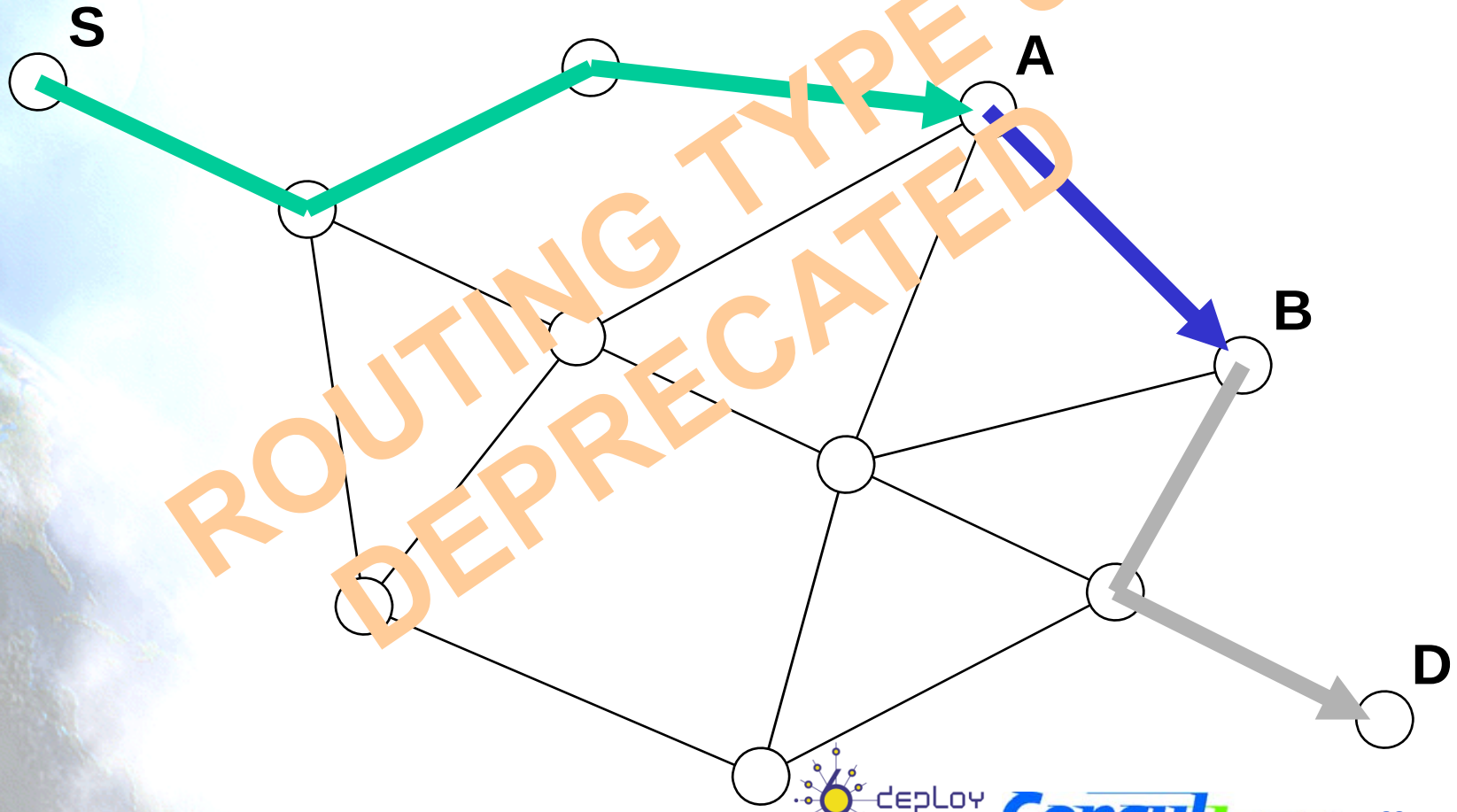
Example: Headers when A to B

IPv6 Base Header	Routing Header
Source Address = S Destination Address = B	H. Ext. Length = 4 Segments Left = 1 Address 1 = A Address 2 = D



Example: Headers when B to D

IPv6 Base Header	Routing Header
Source Address = S Destination Address = D	H. Ext. Length = 4 Segments Left = 0 Address 1 = A Address 2 = B

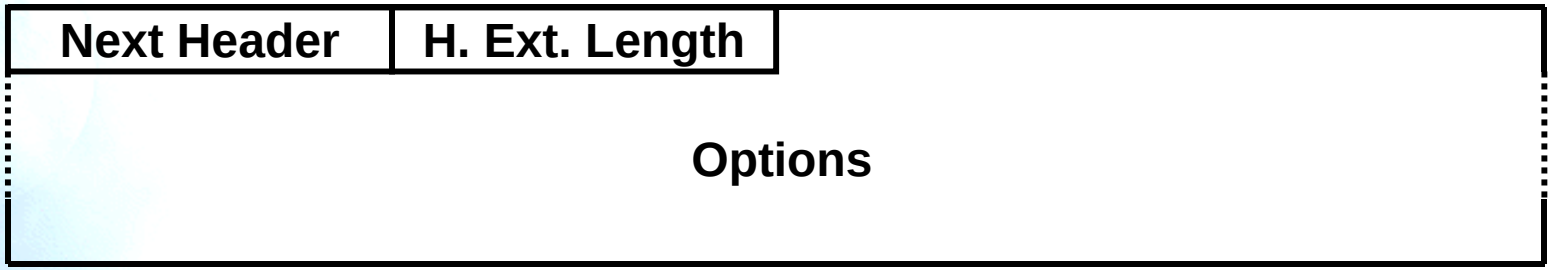


Routing Header Type 0 (deprecated)

- Routing Header Type 0 is deprecated as for RFC5095 because it can be used for traffic amplification which could lead to a DoS attack.
- The attack is based on the following:
  - IPv6 packets could be directed through specific nodes
  - Nodes could appear more than once
  - Traffic could be sent between two nodes more than once
  - All the path among the two nodes is affected
- The risk is enough to deprecate this header type
  - Nodes should proceed as specified in section 4.4 of RFC2460 for routing header of unknown type
- The use of this header is not allowed for “benign” uses
  - In the future another routing header type could be specified for these uses
- Only type 0 is affected by the deprecation, other ones, like type 2 used for Mobility are still valid

Hop-by-Hop & Destination Options Headers

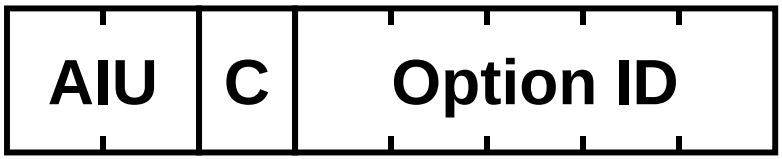
- "Containers" for variable-length options:



- Where Options =

- Next Header values:
  - 0 for Hop-by-Hop Options Header
  - 60 for Destination Options Header

Option Type Encoding



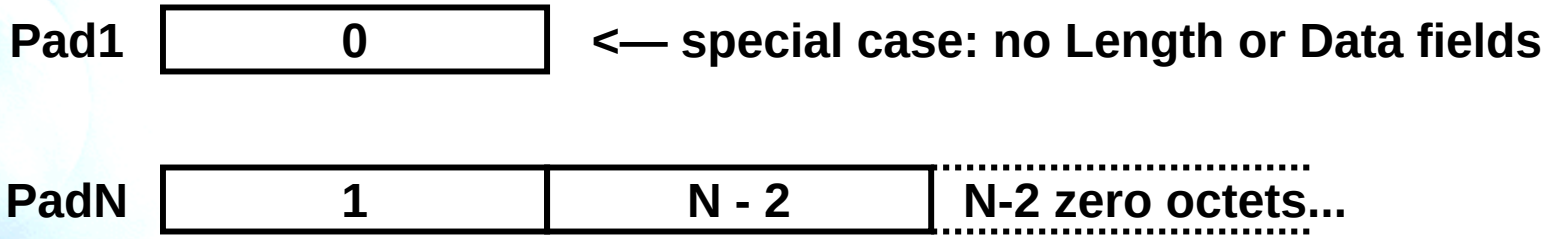
AIU — action if unrecognized:

00 — skip over option
01 — discard packet
10 — discard packet &
send ICMP Unrecognized Type to source
11 — discard packet &
send ICMP Unrecognized Type to source
only if destination was not multicast

C — set (1) if Option Data changes en-route
(Hop-by-Hop Options only)

Option Alignment and Padding

Two Padding Options:



- Used to align options so multi-byte data fields fall on natural boundaries
- Used to pad out containing header to an integer multiple of 8 bytes

Fragment Header

- Used by an IPv6 Source to send a packet larger than would fit in the path MTU to its destination.
- In IPv6 the Fragmentation is only performed by source nodes, not routers.
- Next Header value = 44

8 bits	8 bits	13 bits unsigned	2 bits	1 bit
Next Header	**Reserved = 0**	**Fragment Offset**	**Res. = 0**	**M**
Identification				

- Original Packet (unfragmented):

Unfragmentable Part	**Fragmentable Part**

Fragmentation Process

- The Fragmentable Part of the original packet is divided into fragments, each, except possibly the last ("rightmost") one, being an integer multiple of 8 octets long. The fragments are transmitted in separate "fragment packets“

Unfragmentable Part	1st Fragment	2nd Fragment	...	Last Fragment

- Fragment Packets:

Unfragmentable Part	Fragment Header	1st Fragment

Unfragmentable Part	Fragment Header	2nd Fragment

• • •

Unfragmentable Part	Fragment Header	Last Fragment

2.3. Packet Size Issues

Minimum MTU

- Link MTU:
  - A link's maximum transmission unit, i.e., the max IP packet size that can be transmitted over the link
- Path MTU:
  - The minimum MTU of all the links in a path between a source and a destination
- Minimum link MTU for IPv6 is 1280 octets vs. 68 octets for v4
- On links with MTU < 1280, link-specific fragmentation and reassembly must be used
- On links that have a configurable MTU, it's recommended a MTU of 1500 bytes

Path MTU Discovery (RFC1981)

- Implementations are expected to perform path MTU discovery to send packets bigger than 1280 octets:
  - for each destination, start by assuming MTU of first-hop link
  - if a packet reaches a link in which it can't fit, will invoke ICMP "packet too big" message to source, reporting the link's MTU; MTU is cached by source for specific destination
  - occasionally discard cached MTU to detect possible increase
- Minimal implementation can omit path MTU discovery as long as all packets kept ≤ 1280 octets
  - e.g., in a boot ROM implementation

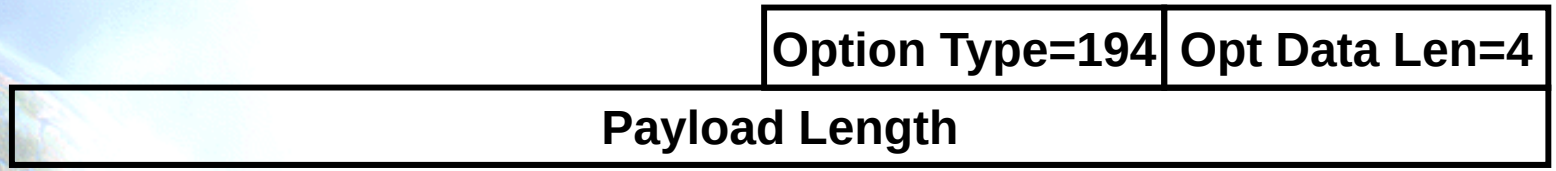
Fragment Header

Next Header	Reserved	Fragment Offset	0 0 M
Original Packet Identifier			

- Though discouraged, can use IPv6 Fragment header to support upper layers that do not (yet) do path MTU discovery
- IPv6 fragmentation & reassembly is an end-to-end function; routers do not fragment packets en-route if too big, instead, they send ICMP “packet too big”.

Maximum Packet Size

- Base IPv6 header supports payloads of up to 65,535 bytes (not including 40 byte IPv6 header)
- Bigger payloads can be carried by setting IPv6 Payload Length field to zero, and adding the “jumbogram” hop-by-hop option:



- Can’t use Fragment header with jumbograms (RFC2675)

2.4. Upper-Layer Protocol Issues

Upper-Layer Checksums

- Any transport or other upper-layer protocol that includes the addresses from the IP header in its checksum computation must be modified for use over IPv6, to include the 128-bit IPv6 addresses instead of 32-bit IPv4 addresses.
- TCP/UDP “pseudo-header” for IPv6:

Source Address	
Destination Address	
Upper-Layer Packet Length	
zero	Next Header

- ICMPv6 includes the above pseudo-header in its checksum computation (change from ICMPv4). Reason: Protect ICMP from misdelivery or corruption of those fields of the IPv6 header on which it depends, which, unlike IPv4, are not covered by an internet-layer checksum. The Next Header field in the pseudo-header for ICMP contains the value 58, which identifies the IPv6 version of ICMP.

Maximum Packet Lifetime

- IPv6 nodes are not required to enforce maximum packet lifetime.
- That is the reason the IPv4 "Time to Live" field was renamed "Hop Limit" in IPv6.
- In practice, very few, if any, IPv4 implementations conform to the requirement that they limit packet lifetime, so this is not a “real” change.
- Any upper-layer protocol that relies on the internet layer (whether IPv4 or IPv6) to limit packet lifetime ought to be upgraded to provide its own mechanisms for detecting and discarding obsolete packets.

Maximum Upper-Layer Payload Size

- When computing the maximum payload size available for upper-layer data, an upper-layer protocol must take into account the larger size of the IPv6 header relative to the IPv4 header.
- Example: in IPv4, TCP's MSS option is computed as the maximum packet size (a default value or a value learned through Path MTU Discovery) minus 40 octets (20 octets for the minimum-length IPv4 header and 20 octets for the minimum-length TCP header). When using TCP over IPv6, the MSS must be computed as the maximum packet size minus 60 octets, because the minimum-length IPv6 header (i.e., an IPv6 header with no extension headers) is 20 octets longer than a minimum-length IPv4 header.

Responding to Packets Carrying Routing Headers

- When an upper-layer protocol sends one or more packets in response to a received packet that included a Routing header, the response packet(s) must not include a Routing header that was automatically derived by "reversing" the received Routing header UNLESS the integrity and authenticity of the received Source Address and Routing header have been verified (e.g., via the use of an Authentication header in the received packet).

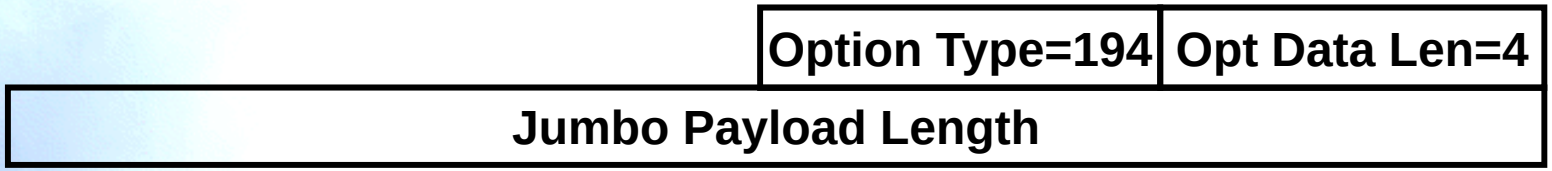
2.5. Jumbograms

IPv6 Jumbograms RFC2675

- “Jumbogram" is an IPv6 packet containing a payload longer than 65,535 octets
- Jumbogram
  - Is relevant only to IPv6 nodes that may be attached to links with a link MTU > 65,575 octets (65,535 + 40 of the IPv6 header)
  - Need not be implemented by IPv6 nodes that do not support attachment to links with such large MTUs
- RFC2675 describes the IPv6 Jumbo Payload option
  - Also provides the means of specifying such large payload lengths
  - And describes the changes needed to TCP and UDP to make use of Jumbograms

IPv6 Jumbo Payload Option

- Jumbo Payload option is carried in an IPv6 Hop-by-Hop Options header, immediately following the IPv6 header
- Format:



- Jumbo Payload Length Field
  - 32-bit unsigned integer
  - Length of the IPv6 packet in octets, excluding the IPv6 header but including the Hop-by-Hop Options header and any other extension headers present
  - Must be greater than 65,535.

UDP Jumbograms

- The 16-bit field of the UDP header limits the total length of a UDP packet (that is, a UDP header plus data) to no greater than 65,535 octets

- RFC2675 specifies the modification of UDP to relax that limit:
  - UDP packets longer than 65,535 octets may be sent by setting the UDP Length field to zero, and letting the receiver derive the actual UDP packet length from the IPv6 payload length
  - Note that, prior to this modification, zero was not a legal value for the UDP Length field, because the UDP packet length includes the UDP header and therefore has a minimum value of 8

TCP Jumbograms

- There is no length field in TCP header, then is nothing limiting the length of an individual TCP packet. However
  - MSS value that is negotiated at the beginning of the connection limits the largest TCP packet that can be sent
  - Urgent Pointer cannot reference data > 65,535 bytes
- Proposed Solutions
  - When determining what MSS value to send
    - if MTU of the directly attached interface minus 60 is > = to 65,535, then set MSS to 65,535
    - When an MSS value of 65,535 is received, it is to be treated as infinity
    - The actual MSS is determined by subtracting 60 from the value learned by performing Path MTU Discovery over the path to the TCP peer
  - The Urgent Pointer problem could be fixed by adding a TCP Urgent Pointer Option. However, since it is unlikely that applications using Jumbograms will also use Urgent Pointers, a less intrusive change similar to the MSS change will suffice

IPv6 Tutorial

3. Addressing

Agenda

3.1. Types of Addresses
3.2. Prefix and Representation
3.3. Unique Local IPv6 Addresses
3.4. Interface IDs
3.5. Multicast Addresses
3.6. Other Considerations

3.1. Types of Addresses

Address Types (RFC4291)

Unicast (one-to-one)
- global
- link-local
- site-local (deprecated)
- Unique Local (ULA)
- IPv4-compatible (deprecated)
- IPv6-mapped

Multicast (one-to-many)

Anycast (one-to-nearest)

Reserved

Some Special-Purpose Unicast Addresses (RFC5156)

- The unspecified address, used as a placeholder when no address is available:

  0:0:0:0:0:0:0:0 (::/128)

- The loopback address, for sending packets to itself:

  0:0:0:0:0:0:0:1 (::1/128)

3.2. Prefix and Representation

Text Representation of Addresses

“Preferred” form:	2001:DB8:FF:0:8:7:200C:417A
Compressed form:	2001:DB8:0:0:0:0:0:43
	becomes 2001:DB8::43
IPv4-compatible:	::13.1.68.3 (deprecated in RFC4291)
IPv4-mapped:	::FFFF:13.1.68.3

URL: http://[2001:DB8::43]:80/index.html

Address Type Prefixes

Address Type	Binary Prefix	IPv6 Notation
Unspecified	00…0 (128 bits)	::/128
Loopback	00…1 (128 bits)	::1/128
Multicast	1111 1111	FF00::/8
Link-Local Unicast	1111 1110 10	FE80::/10
ULA	1111 110	FC00::/7
Global Unicast	(everything else)	
IPv4-mapped	00…0:1111 1111:IPv4	::FFFF:IPv4/128
Site-Local Unicast (deprecated)	1111 1110 11	FEC0::/10
IPv4-compatible (deprecated)	00…0 (96 bits)	::IPv4/128

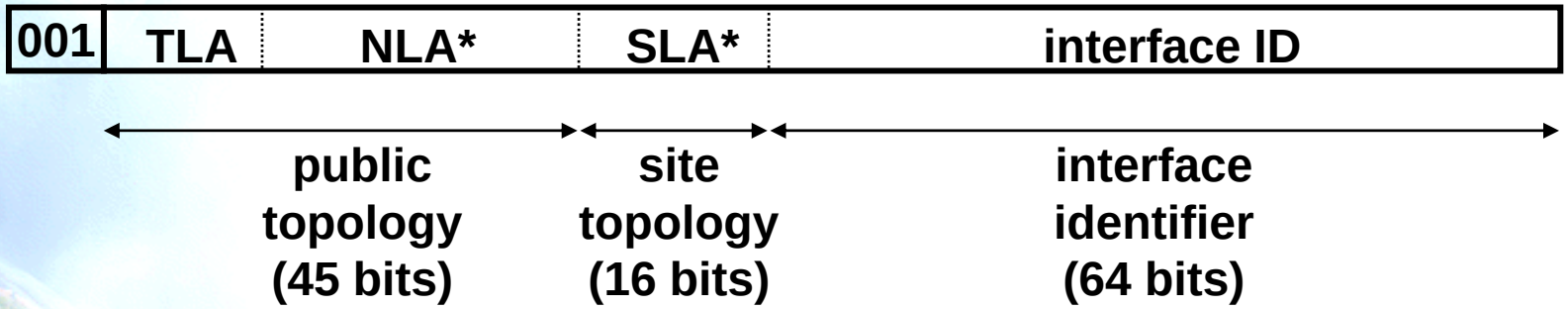
- **Anycast** addresses allocated from unicast prefixes

Global Unicast Prefixes

Address Type	Binary Prefix
IPv4-compatible	0000...0 (96 zero bits) (deprecated)
IPv4-mapped	00…0FFFF (80 zero+ 16 one bits)
Global unicast	001
ULA	1111 110x (1= Locally assigned) (0=Centrally assigned)

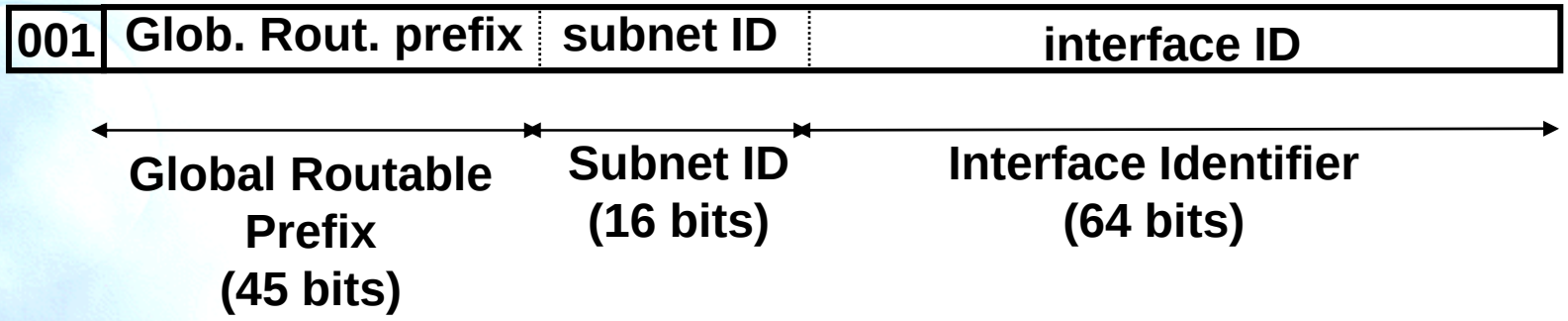
- **2000::/3** prefix is being allocated for Global Unicast, all other prefixes reserved (approx. 7/8ths of total)

Aggregatable Global Unicast Addresses (RFC2374) (Deprecated)



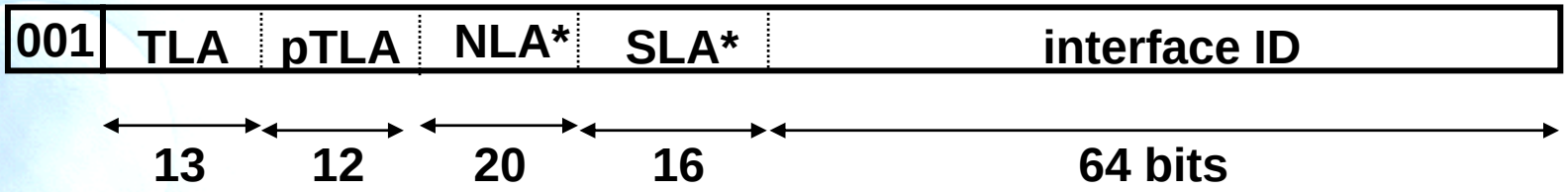
- TLA = Top-Level Aggregator
  NLA* = Next-Level Aggregator(s)
  SLA* = Site-Level Aggregator(s)
- all subfields variable-length, non-self-encoding (like CIDR)
- TLAs may be assigned to providers or exchanges
- Obsoleted by RFC3587: IPv6 Global Unicast Address Format

Global Unicast Addresses (RFC3587)



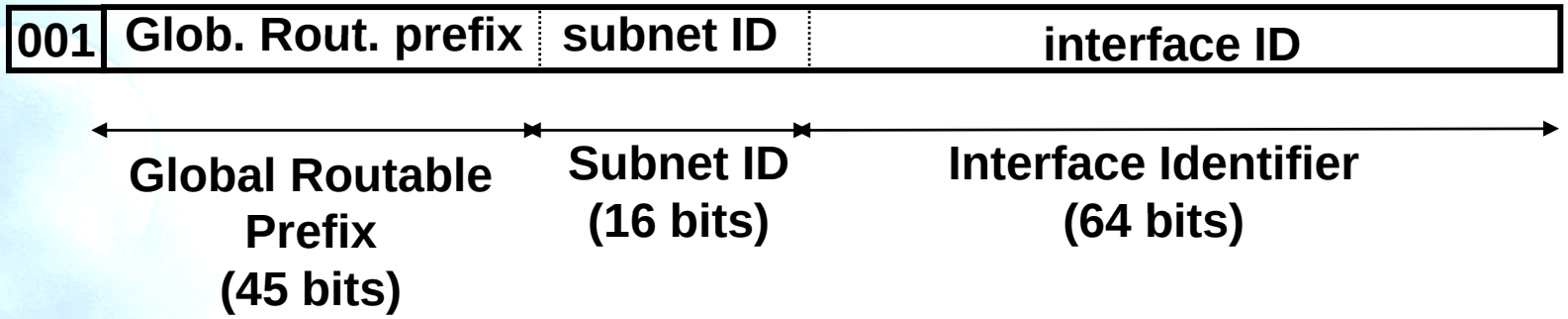
- The global routing prefix is a value assigned to a zone (site, a set of subnetworks/links)
  - It has been designed as an hierarchical structure from the Global Routing perspective
- The subnetwork ID, identifies a subnetwork within a site
  - Has been designed to be an hierarchical structure from the site administrator perspective
- The Interface ID is build following the EUI-64 format

Global Unicast Addresses for the 6Bone (Obsoleted 6/6/6 – RFC3701)



- 6Bone: experimental IPv6 network used for testing only
- TLA 1FFE (hex) assigned to the 6Bone
  - thus, 6Bone addresses start with 3FFE:
  - (binary 001 + 1 1111 1111 1110)
- Next 12 bits hold a “pseudo-TLA” (pTLA)
  - thus, each 6Bone pseudo-ISP gets a /28 prefix
- Not to be used for production IPv6 service

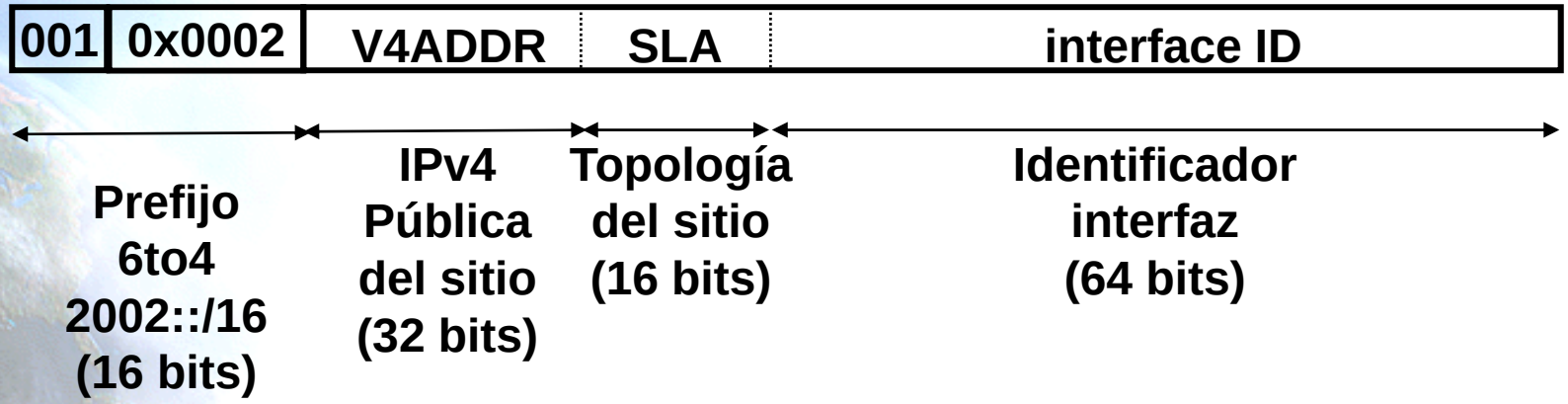
Global Unicast Addresses for Production Service



- LIRs receive by default /32
  - Production addresses today are from prefixes 2001, 2003, 2400, 2800, etc.
  - Can request for more if justified
- /48 used only within the LIR network, with some exceptions for critical infrastructures
- /48 to /128 is delegated to end users
  - Recommendations following RFC3177 and current policies
    - /48 general case, /47 if justified for bigger networks
    - /64 if one and only one network is required
    - /128 if it is sure that one and only one device is going to be connected

6to4 Addresses (RFC3056)

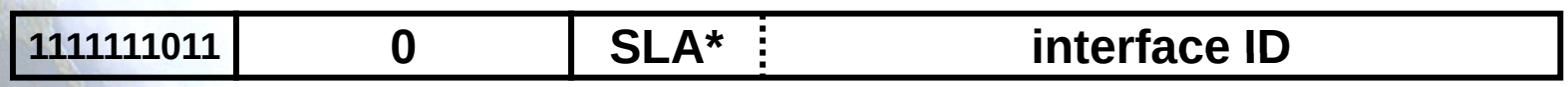
- RFC3056: Connection of IPv6 Domains via IPv4 Clouds
- Assigned Prefix: **2002::/16**
- To assign to sites **2002:V4ADDR::/48**



Link-Local & Site-Local Unicast Addresses

Link-local addresses for use during auto-configuration and when no routers are present (**FE80::/10**):

Site-local addresses for independence from changes of TLA / NLA* (**FEC0::/10**): (deprecated RFC3879)



Documentation Prefix

- Prefix Assigned: **2001:DB8::/32**

- RFC3849: IPv6 Address Prefix Reserved for Documentation
- Unicast IPv6 prefix reserved for examples in:
  - –RFCs
  - –Books,
  - –Documents
  - –etc.

3.3. Unique Local IPv6 Addresses

Unique Local IPv6 Unicast Addresses

IPv6 ULA (RFC4193)

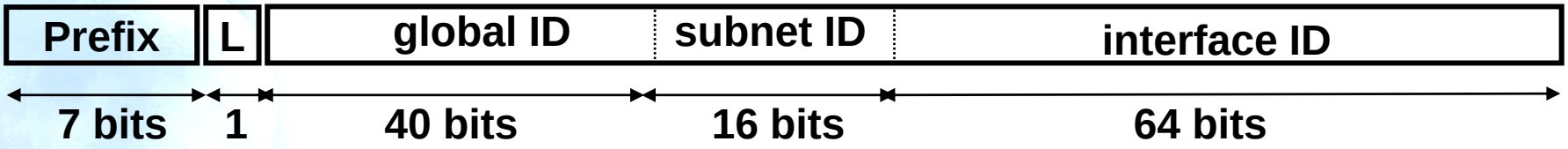
- Globally unique prefix with high probability of uniqueness
- Intended for local communications, usually inside a site
- They are not expected to be routable on the Global Internet
- They are routable inside of a more limited area such as a site
- They may also be routed between a limited set of sites
- Locally-Assigned Local addresses
  - vs Centrally-Assigned Local addresses

IPv6 ULA Characteristics

- Well-known prefix to allow for easy filtering at site boundaries
- ISP independent and can be used for communications inside of a site without having any permanent or intermittent Internet connectivity
- If accidentally leaked outside of a site via routing or DNS, there is no conflict with any other addresses
- In practice, applications may treat these addresses like global scoped addresses

IPv6 ULA Format

- Format:



- FC00::/7 Prefix identifies the Local IPv6 unicast addresses
- L = 1 if the prefix is locally assigned
- L = 0 may be defined in the future (in practice used for centrally assigned prefixes)
- ULA are created using a pseudo-randomly allocated global ID
  - This ensures that there is not any relationship between allocations and clarifies that these prefixes are not intended to be routed globally

Centrally Assigned Unique Local IPv6 Unicast Addresses (1)

- Centrally-Assigned Local addresses
  - vs Locally-Assigned Local addresses
- Latest Draft:
  - draft-ietf-ipv6-ula-central-02.txt
  - June 2007
  - It defines the characteristics and requirements for Centrally assigned Local IPv6 addresses in the framework defined in IPv6 ULA – RFC4193

Centrally Assigned Unique Local IPv6 Unicast Addresses (2)

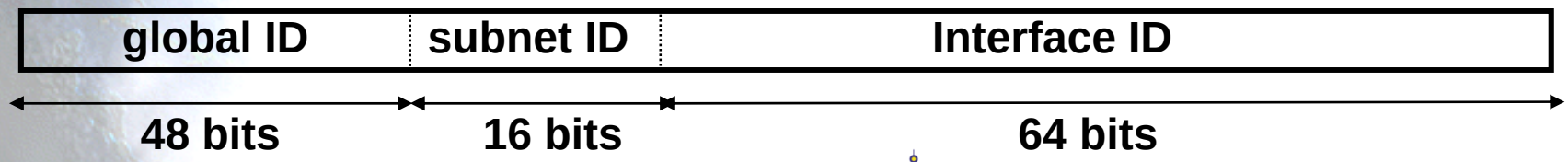
- The major difference between both assignments:
  - the Centrally-Assigned is uniquely assigned and the assignments are registered in a public database.
- It is recommended that sites planning to use Local IPv6 addresses use a centrally assigned prefix as there is no possibility of assignment conflicts. Sites are free to choose either approach.
- The allocation procedure for creating global-IDs for centrally assigned local IPv6 addresses is setting L=0. Remember that the allocation procedure for locally assigned local IPv6 addresses is thru L=1, as is defined in RFC4193.
- More info on RIR's policies for centrally-assigned ULAs assigments:
  - http://www.arin.net/meetings/minutes/ARIN_XVIII/ppm2_transcript.html#anchor_3
  - http://www.arin.net/meetings/minutes/ARIN_XIX/ppm1_notes.html

3.4. Interface IDs

Interface IDs

The lowest-order 64-bit field of unicast addresses may be assigned in several different ways:

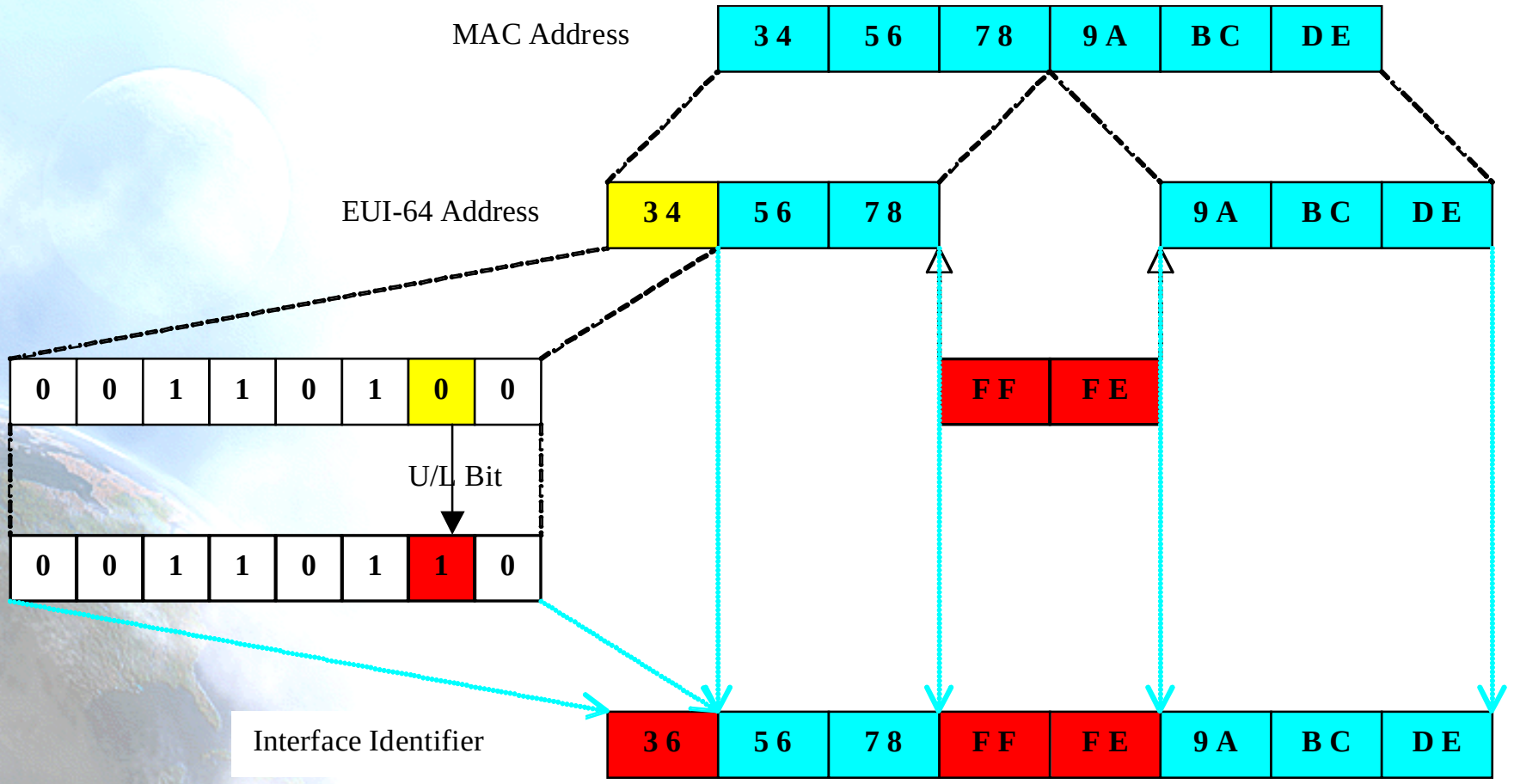
- auto-configured from a 48-bit MAC address (e.g., Ethernet address), expanded into a 64-bit EUI-64
- assigned via DHCP
- manually configured
- auto-generated pseudo-random number (to counter some privacy concerns)
- possibly other methods in the future



IPv6 in Ethernet

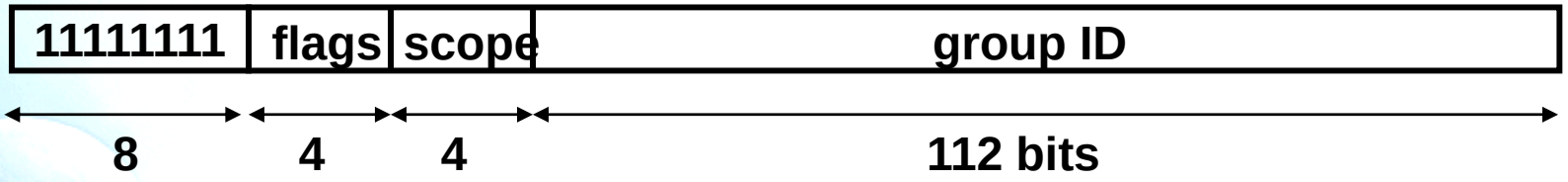
48 bits	48 bits	16 bits	
Ethernet Destination Address	Ethernet Source Address	1000011011011101 (86DD)	IPv6 Header and Data

EUI-64


MAC Address
3 4
5 6
7 8
9 A
B C
D E
EUI-64 Address
3 4
5 6
7 8
F F
F E
9 A
B C
D E
0 0 1 1 0 1 0 0
U/L Bit
0 0 1 1 0 1 1 0
Interface Identifier
3 6
5 6
7 8
F F
F E
9 A
B C
D E

3.5. Multicast Addresses

Multicast Addresses



- Flags: **0RPT**: The high-order flag is reserved, and must be initialized to 0.
  - T: Transient, or not, assignment
  - P: Assigned, or not, based on network prefix
  - R: Rendezvous Point Addr. embedded, or not
- Scope field: 1 - Interface-Local
  2 - link-local
  4 - admin-local
  5 - site-local
  8 - organization-local
  E - global

(3,F reserved)(6,7,9,A,B,C,D unassigned)

3.6. Other Considerations

IPv6 Node Required Addresses

- IPv6 Host required addresses:

1. Link-Local address for each interface.
2. Any additional Unicast and Anycast addresses that have been configured for the node's interfaces (manually or automatically).
3. The loopback address.
4. The All-Nodes multicast addresses (FF01::1, FF02::1).
5. The Solicited-Node multicast address for each of its unicast and anycast addresses.
6. Multicast addresses of all other groups to which the node belongs.

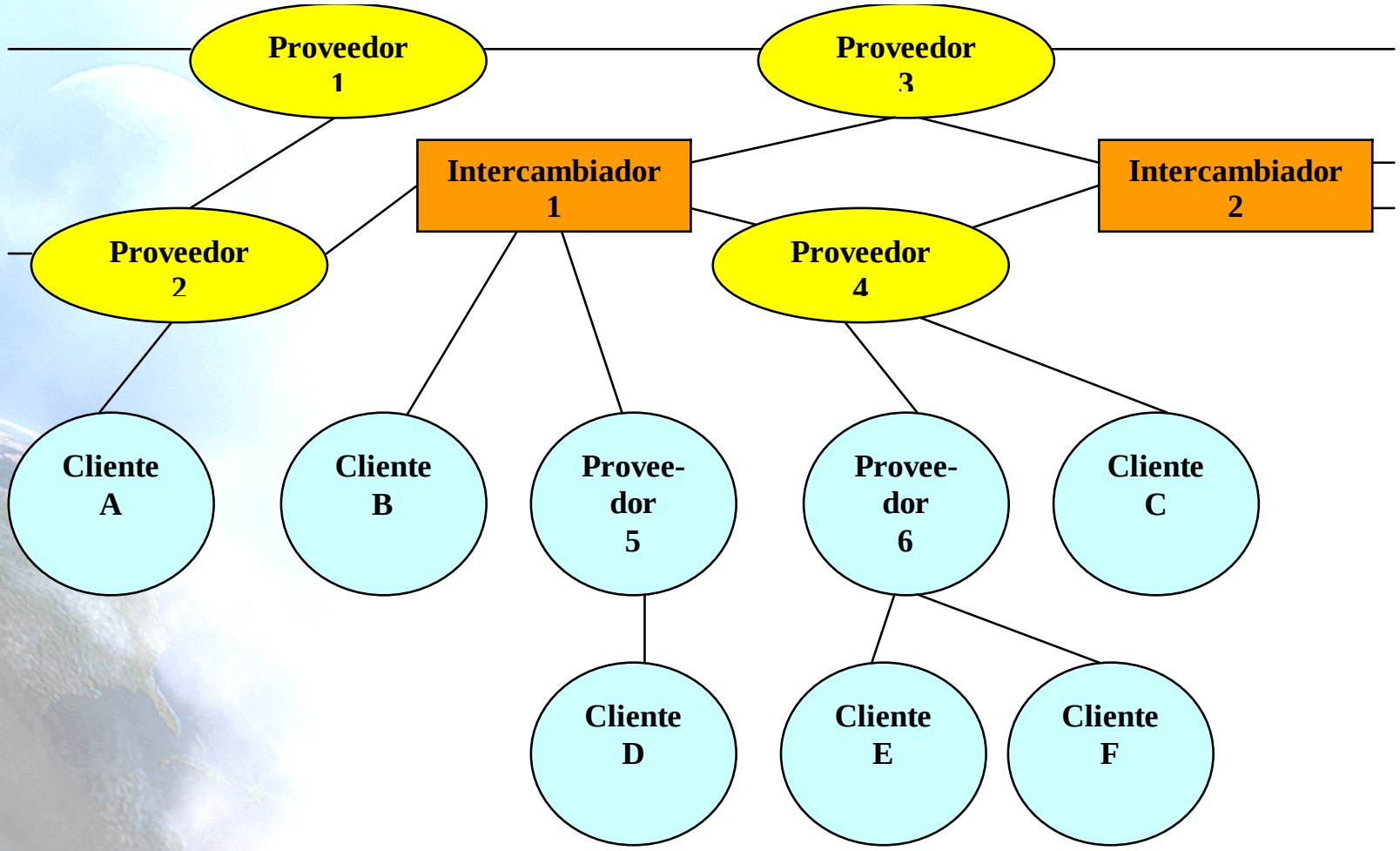
- IPv6 Router required addresses: Host +:

1. The Subnet-Router Anycast addresses for all interfaces for which it is configured to act as a router.
2. All other Anycast addresses with which the router has been configured.
3. The All-Routers multicast addresses (FF01::2, FF02::2, FF05::2).

Address Aggregation

- The aggregatable address format was designed to support long distance transit providers, exchanges, lower level providers and clients.
- Exchanges could provide IPv6 addresses. This model is not supported with IPv4.
  - Organizations connected to exchanges could receive connectivity services from the exchange or from one ore more transit providers.
- In getting addresses from an exchange, transit provider independence is obtained.
  - This way it is easy to change from one transit provider to another without having a renumbering problem. This is one of the aims of IPv6.

Aggregation Scheme



Addressing Plan (1)

- The aims of the addressing plan are:
  - Assign addresses from a prefix obtained from a RIR
  - The assignation is for the different existent and future networks
- The following criteria could be used (RFC3711 and best practices):
  - Assign a /64 prefix to all internal IPv6 networks. This is required for autoconfiguration of IPv6 addresses (Unicsat and/or Anycast).
  - End users, including residential (DSL, FTTx, etc.) and corporate (enterprises, ISPs, Universities, etc.) ones could receive a /48 prefix.
    - This allows for 2^{16} (65536) IPv6 networks using a /64 prefix

Addressing Plan (2)

- Assigning enough addresses to have 65536 /64 networks could look like being too much, but there are some reasons for this:
  - The future NGN deployments will ease the implementation of advanced services like VoIP, IPTV, etc., that could require specific networks for each final user.
  - New and unimaginable applications and/or services could appear in the future, in fields like home automation or ambient intelligence. These could require different networks for each device/service.

Addressing Plan (3)

- To assign addresses to links there are two approaches:
  - Use a specific **address pool** to use in the native IPv6 point-to-point links or IPv6-in-IPv4 tunnels between the CPE and provider's BRAS. And use another ***/48* prefix pool** for end user assigment.
  - Another approach, that seeks to ease the assignation and numbering process, is to use the first /64 prefix from the /48 prefix assigned to the end user. This /64 prefix is used on the point-to-point link between CPE and BRAS (draft-palet-v6ops-point2point). This way less operational effort is needed and the addressing plan could have a "flat" structure.

Addressing Plan (4)

- For the addressing plan the different networks that could eventually implement IPv6 should be taken into account, these can include:
  - Networks that could implement native IPv6 from the first moment
  - Networks that could implement native IPv6 in the middle/long-term
  - IPv6 transition services
- The main objective is not to have to change the addressing plan in the future, i.e., when massive IPv6 deployment is carried out

Addressing Plan (5)

- The sub-networks that could be considered in an addressing plan are:
  - Internet/BGP Interdomain routing
  - Core Network/Intradomain routing
  - Management Network.
  - Basic Network Services
  - DNS
  - NTP
  - Web Caches.
  - Corporative Intranet.
  - WiFi.
  - Remote Access (VPNs, etc.).
  - Data Center.
  - IPv6 Transition Mechanisms (tunnels, etc.).
  - End users/ISPs.
  - ADSL-FTTH End users.
  - GPRS/3G Users.

Addressing Plan (6)

- Following is an addressing plan example using a /32 prefix. With this prefix more than 50,000 end users could obtain a /48 prefix.
- The /32 prefix is divided into /38 prefixes that are grouped to be assigned to considered networks defined following these rules:
  - Networks that are independent from each other
  - Networks with similar topology
  - Leave spare /38 prefixes to give flexibility in case of network growths

Addressing Plan (7)

- Based on the above, an example with 6 groups, to which assign /38 prefixes is:
  1. Core networks and internal networks
     - Routing
     - Basic Services
     - Internal Networks
     - WiFi
     - Links
     - Mobility
     - Data Center
  2. Tunnels
  3. Corporative Clients and ISPs
  4. Residential Users (ADSL-FTTH)
  5. GPRS/3G
  6. Free Prefixes

Addressing Plan (8)

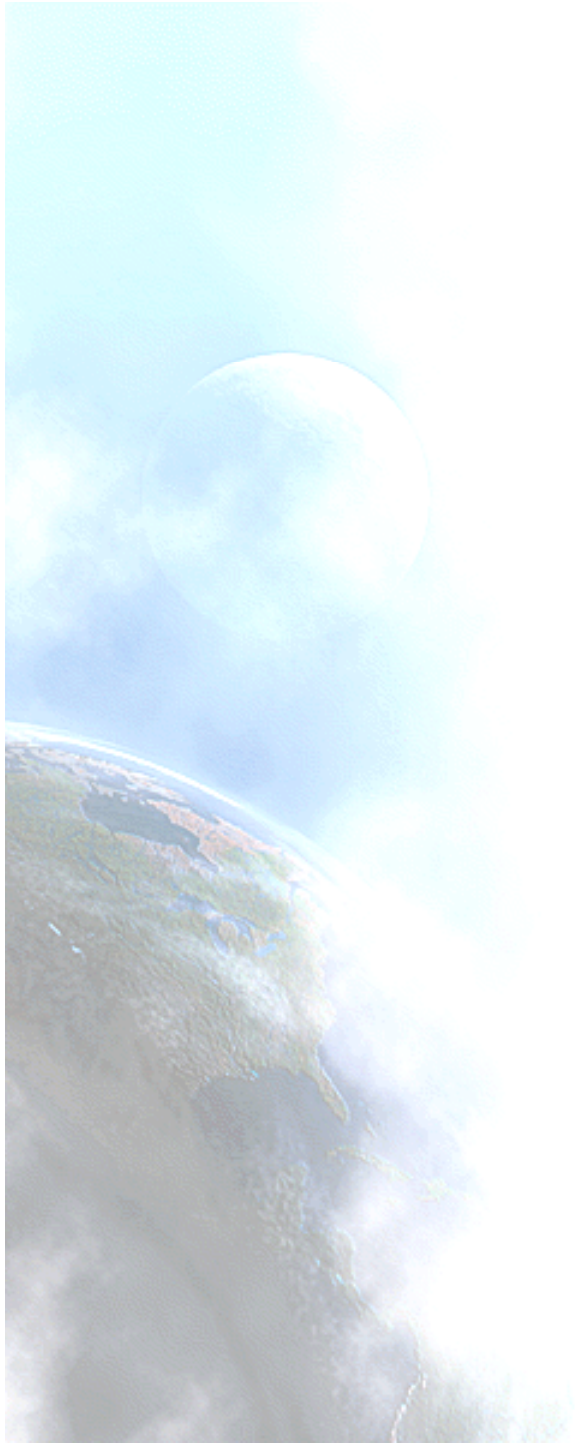
#	Prefix	Category	Number of Prefixes	Prefix Length
0	2001:DB8:0000::/38	Routing, Basic Services, Internal Networks, WiFi, Links, Mobility, Data Center		
1	2001:DB8:0400::/38	Free	1	/38
2	2001:DB8:0800::/38	Tunnels		
	2001:DB8:0C00::/38 2001:DB8:1000::/38	Free	2	/38
5	2001:DB8:1400::/38	Corporative Clients and ISPs	1.024	/48
6	2001:DB8:1800::/38	Corporative Clients and ISPs	1.024	/48
7	2001:DB8:1C00::/38	Corporative Clients and ISPs	1.024	/48
	2001:DB8:2000::/38 ... 2001:DB8:3C00::/38	Free	8	/38
16	2001:DB8:4000::/38	ADSL-FTTH Users	1.024	/48
	Hasta	ADSL-FTTH Users	1.024	/48
35	2001:DB8:8C00::/38	ADSL-FTTH Users	1.024	/48
	2001:DB8:9000::/38 2001:DB8:9400::/38 2001:DB8:9800::/38	Free	3	/38
39	2001:DB8:9C00::/38	GPRS/3G	67.108.864	/64
	2001:DB8:A000::/38 2001:DB8:A400::/38	Free	2	/38
42	2001:DB8:A800::/38	GPRS/3G	1.024	/48
	Hasta	GPRS/3G	1.024	/48
61	2001:DB8:F400::/38	GPRS/3G	1.024	/48
	2001:DB8:F800::/38 2001:DB8:FC00::/38	Free	2	/38
		Total prefijos /38 Libres	18	

IPv6 Tutorial

4. ICMPv6, Neighbor Discovery & DHCPv6

Agenda

4.1. ICMPv6
4.2. Neighbor Discovery
4.3. Secure Neighbor Discovery
4.4. Autoconfiguration
4.5. DHCPv6
4.6. Router Renumbering



4.1. ICMPv6

ICMPv6 (RFC4443)

- IPv6 uses the Internet Control Message Protocol (ICMP) as defined for IPv4 (RFC792)
- Some changes for IPv6: ICMPv6
- Next Header value = 58
- ICMPv6 is used by IPv6 nodes to report errors encountered in processing packets, and to perform other internet-layer functions, such as diagnostics (ICMPv6 "ping")
- ICMPv6 is an integral part of IPv6 and MUST be fully implemented by every IPv6 node

ICMPv6 Messages

- Grouped into two classes:
  - Error messages
  - Informational messages

bits 8	16	32
Type	Code	Checksum
Message Body		

- Error messages have a zero in the high-order bit of their message Type field values (message Types from 0 to 127)
- Informational messages have message Types from 128 to 255

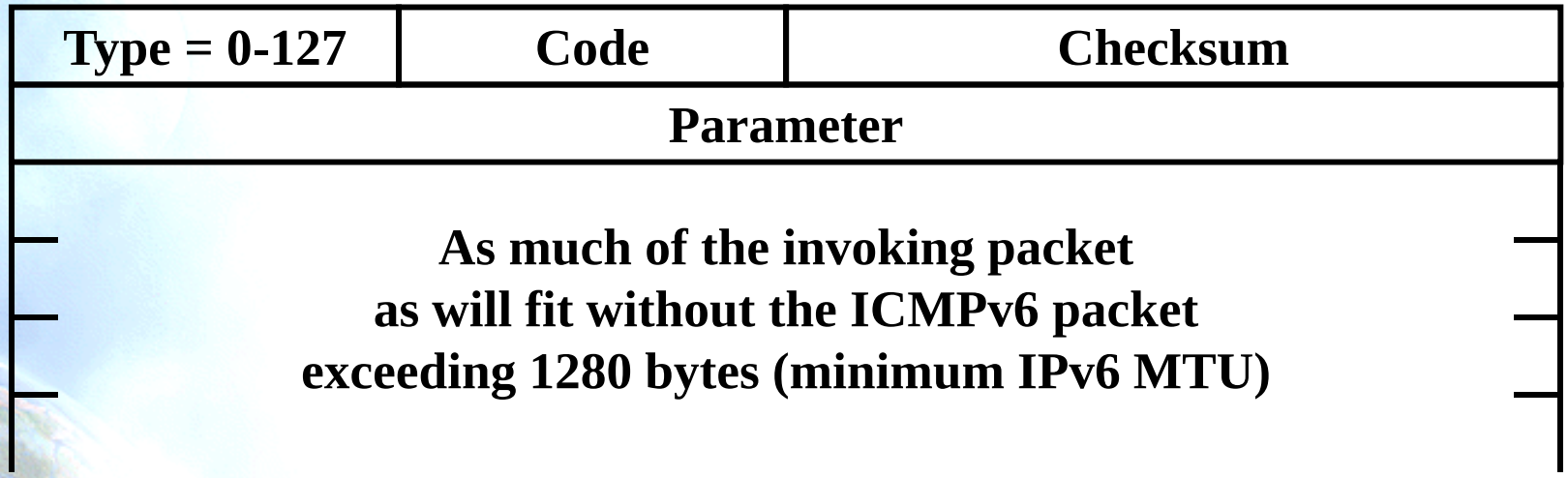
Message Source Address Determination

- A node that sends an ICMPv6 message has to determine both the Source and Destination IPv6 Addresses in the IPv6 header before calculating the checksum.
- If the node has more than one unicast address, it must choose the Source Address of the message as follows:

a) Message responding to a message sent to one of the node's unicast addresses, then Reply Source Address = Same Address.

b) Message responding to a message sent to any other address (multicast group address, anycast address implemented by the node or unicast address not belonging to the node) then, the Source Address of the ICMPv6 packet MUST be a unicast address belonging to the node. The address SHOULD be chosen according to the rules that would be used to select the source address for any other packet originated by the node, given the destination address of the packet. However, it MAY be selected in an alternative way if this would lead to a more informative choice of address reachable from the destination of the ICMPv6 packet.

ICMP Error Messages

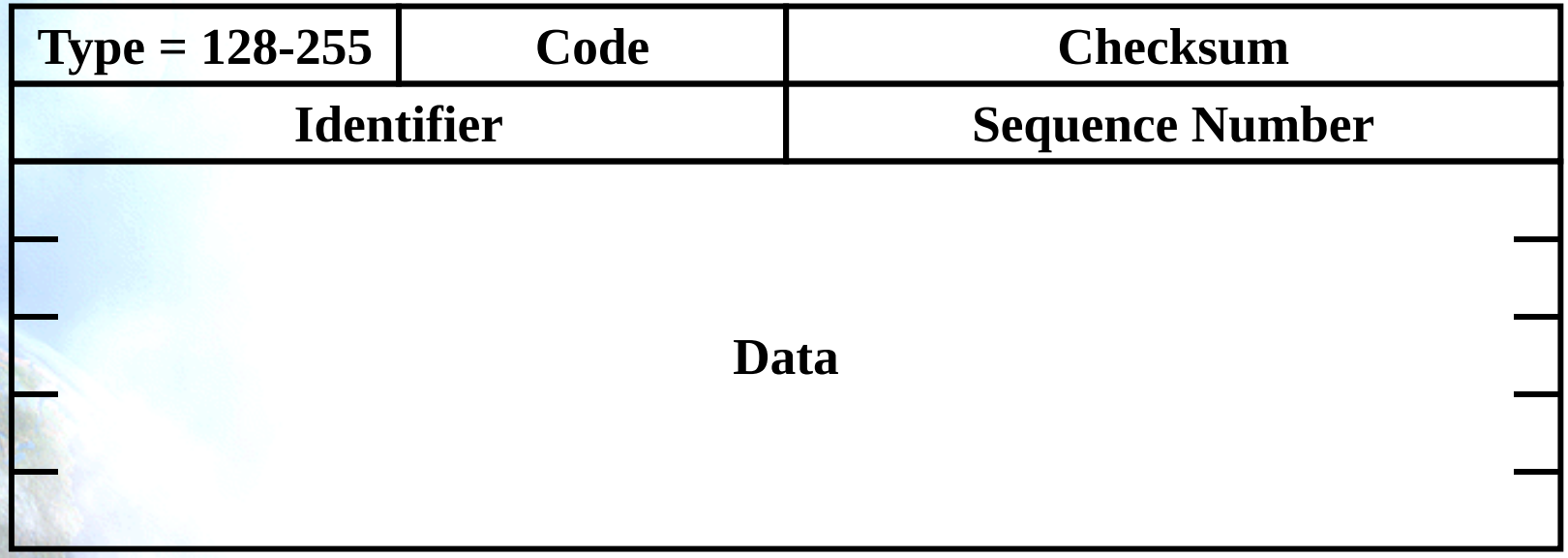


ICMP Error Messages Types

- Destination Unreachable (type = 1, parameter = 0)
  - No route to destination (code = 0)
  - Communication with destination administratively prohibited (code = 1)
  - Beyond scope of source address (code = 2)
  - Address Unreachable (code = 3)
  - Port Unreachable (code = 4)
  - Source address failed ingress/egress policy (code = 5)
  - Reject route to destination (code = 6)
- Packet Too Big (type = 2, code = 0, parameter = next hop MTU)
- Time Exceeded (type = 3, parameter = 0)
  - Hop Limit Exceeded in Transit (code = 0)
  - Fragment Reassembly Time Exceeded (code = 1)
- Parameter Problem (type = 4, parameter = offset to error)
  - Erroneous Header Field (code = 0)
  - Unrecognized Next Header Type (code = 1)
  - Unrecognized IPv6 Option (code = 2)

ICMP Informational Messages

- Echo Request (type =128, code = 0)
- Echo Reply (type = 129, code = 0)



- Multicast listener discovery messages:
  - Query, report, done (like IGMP for IPv4)

4.2. Neighbor Discovery

ND (RFC4861)

- Defines the Neighbor Discovery (ND) protocol for IPv6
- Nodes (hosts and routers) use Neighbor Discovery to determine the link-layer addresses for neighbors known to reside on attached links and to quickly purge cached values that become invalid
- Hosts also use Neighbor Discovery to find neighboring routers that are willing to forward packets on their behalf
- Nodes use the protocol to actively keep track of which neighbors are reachable and which are not, and to detect changed link-layer addresses
- ND enables the autoconfiguration mechanism in IPv6

Interaction Between Nodes

- Defines mechanism to solve:
  - Router Discovery
  - Prefix Discovery
  - Parameter Discovery
  - Address Autoconfiguration
  - Address Resolution
  - Next-hop Determination
  - Neighbor Unreachability Detection (NUD)
  - Duplicate Address Detection (DAD)
  - First-Hop Redirect

New ICMP Packet Types

- ND defines 5 packet types:
  - Router Solicitation (RS)
  - Router Advertisement (RA)
  - Neighbor Solicitation (NS)
  - Neighbor Advertisement (NA)
  - Redirect

Router Advertisements

- On multicast-capable links, each router periodically multicasts a Router Advertisement packet
- A host receives Router Advertisements from all routers, building a list of default routers
- A separate Neighbor Unreachability Detection algorithm provides failure detection
- Router Advertisements contain a list of prefixes used for on-link determination and/or autonomous address configuration
- Router Advertisements allow routers to inform hosts how to perform Address Autoconfiguration

Comparing with IPv4

- IPv6 ND protocol corresponds to a combination of IPv4 protocols: ARP, ICMP Router Discovery, ICMP Redirect en IPv4, plus something more (NUD).
- ND provides a multitude of improvements over the IPv4 set of protocols:
  - Router Advertisements carry link-layer addresses; no additional packet exchange is needed to resolve the router's link-layer address.
  - Router Advertisements carry prefixes for a link; there is no need to have a separate mechanism to configure the "netmask".
  - Router Advertisements enable Address Autoconfiguration.
  - Redirects contain the link-layer address of the new first hop; separate address resolution is not needed upon receiving a redirect.
  - The use of link-local addresses to uniquely identify routers (for RA and Redirect messages) makes it possible for hosts to maintain the router associations in the event of the site renumbering to use new global prefixes.
  - By setting the Hop Limit to 255, ND is immune to off-link senders that accidentally or intentionally send ND messages. In IPv4, off-link senders can send both ICMP Redirects and RA messages.

Router Advertisement Format

Bits 8	16			32
Type = 134	**Code = 0**			**Checksum**
Cur Hop Limit	**M**	**O**	**Reserved = 0**	**Router Lifetime**
Reachable Time				
Retrans Timer				
Options …				

- Cur Hop Limit: default value that should be placed in the Hop Count field of the IP header for outgoing IP packets.
- M: 1-bit "Managed address configuration" flag.
- O: 1-bit "Other stateful configuration" flag.
- Router Lifetime: 16-bit unsigned integer.
- Reachable Time 32-bit unsigned integer.
- Retrans Timer 32-bit unsigned integer.
- Possible Options: Source link-layer address, MTU, Prefix Information, Flags Expansion (RFC5175)

Router Solicitation

- At Start-up, hosts send Router Solicitations in order to prompt routers to generate Router Advertisements quickly.
- Sent to all routers multicast address (link scope).

Bits 8	16	32
Type = 133	**Code = 0**	**Checksum**
Reserved = 0		
Options …		

- Possible Options: Source link-layer address.

Neighbor Solicitation

- Nodes send Neighbor Solicitations to request the link-layer address of a target node while also providing their own link-layer address to the target.
- Neighbor Solicitations are multicast when the node needs to resolve an address and unicast when the node seeks to verify the reachability of a neighbor.

Bits 8	16	32
Type = 135	**Code = 0**	**Checksum**
Reserved = 0		
Target Address		
Options …		

- Target Address: IP address of the target of the solicitation. It MUST NOT be a multicast address.
- Possible Options: Source link-layer address.

Neighbor Advertisement

- A node sends Neighbor Advertisements in response to Neighbor Solicitations and sends unsolicited Neighbor Advertisements in order to (unreliably) propagate new information quickly.

Bits 8			16	32
Type = 136			**Code = 0**	**Checksum**
R	**S**	**O**	**Reserved = 0**	
Target Address				
Options …				

- Flags:
  - **R**: **Router Flag**=1 indicates that the sender is a router.
  - **S**: **Solicited Flag**=1 the advertisement was sent in response to a NS.
  - **O**: **Override Flag**=1 indicates that the advertisement should update caches.
- Target Address: For solicited advertisements, the Target Address field in the Neighbor Solicitation message that prompted this advertisement. For an unsolicited advertisement, the address whose link-layer address has changed. The Target Address MUST NOT be a multicast address.
- Possible Options: Target Link-Layer Address (MAC of the Tx).

Redirect

- Routers send Redirect packets to inform a host of a better first-hop node on the path to a destination.
- Hosts can be redirected to a better first-hop router but can also be informed by a redirect that the destination is in fact a neighbor.

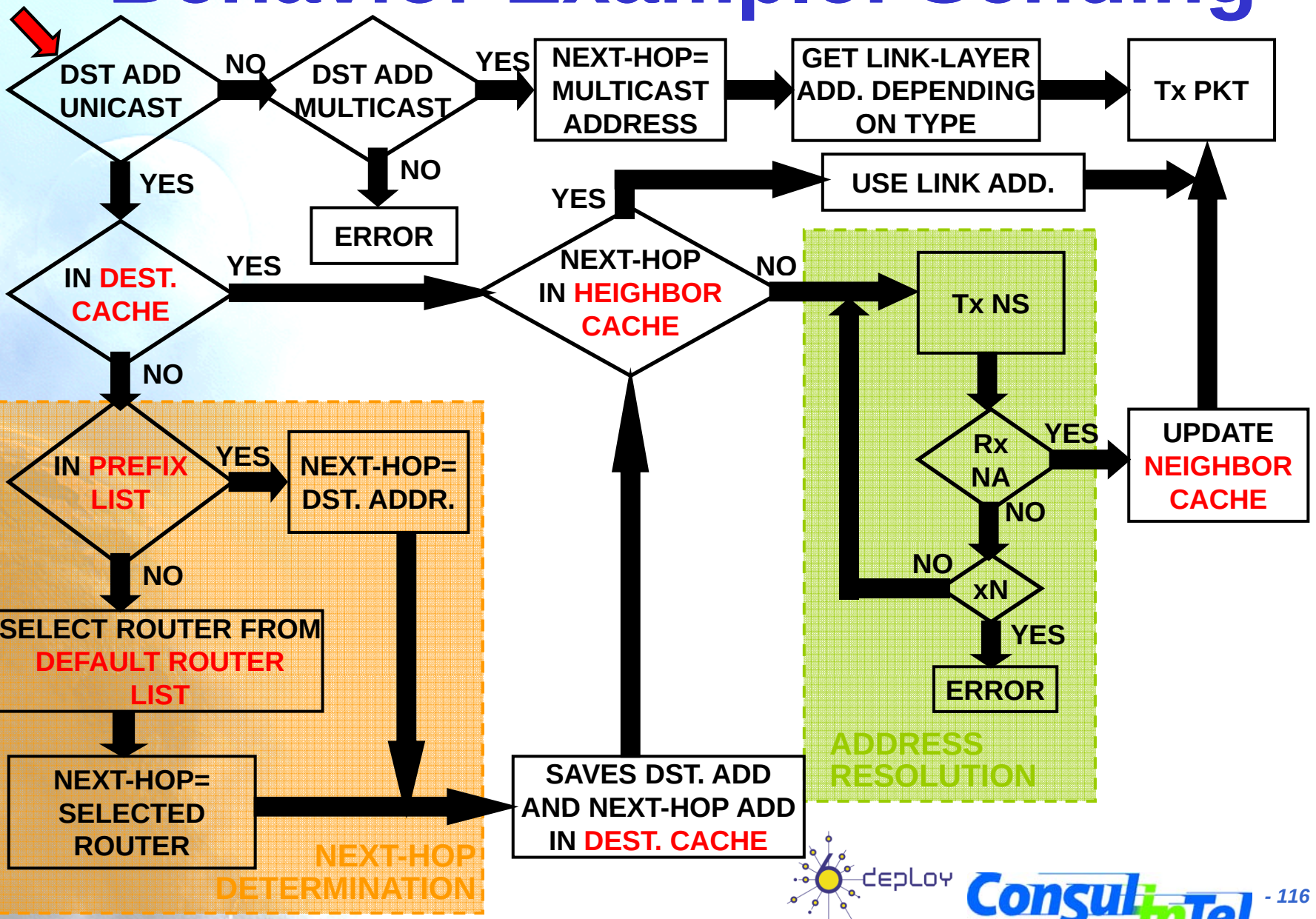
Bits 8	16	32
Type = 137	**Code = 0**	**Checksum**
Reserved = 0		
Target Address		
Destination Address		
Options …		

- Target Address: IP address that is a better first hop to use for the ICMP Destination Address.
- Destination Address: IP address of the destination which is redirected to the target.

Protocol Behavior Example (1)

- **Neighbor Cache**: Neighbors to which traffic have been sent recently. Indexed by the on-link unicast IP address. Each entry contains: link-layer address, if it is router/host, NUD information (reachability state, etc.).
- **Destination Cache**: Maps destination address with next hop. Addresses to which a packet has been sent recently.
- **Prefix List**: Contains link's prefix list. Based on the RAs, from where also the valid lifetime is obtained.
- **Default Router List**: List of routers where off-link packets should be sent. Each entry point to an entry in the Neighbor Cache and has a valid lifetime obtained from the RA (router lifetime).

Behavior Example: Sending



Default Router Preferences and More-Specific Routes [RFC4191]

Bits 8	16			32
Type = 134	**Code = 0**			**Checksum**
Cur Hop Limit	**M**	**O**	**H| PRF| Rsvd**	**Router Lifetime**
Reachable Time				
Retrans Timer				
Options …				

- [RFC4191] describes an optional extension to Router Advertisement messages for communicating default router preferences and more-specific routes from routers to hosts.
- PRF (Default Router Preference) = 01 High
  = 00 Medium (default)
  = 11 Low
  = 10 Reserved (MUST NOT be sent)
- Also a new **Route Information Option** is defined, also with Prf (Route Preference) a 2-bit signed integer (same values are used).

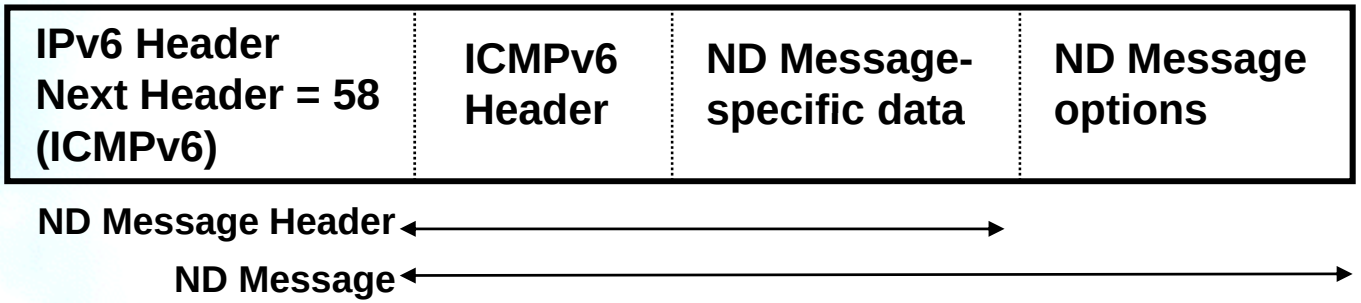
4.3. Secure Neighbor Discovery

Secure Neighbor Discovery (SEND) - RFC3971

- IPv6 nodes use the Neighbor Discovery Protocol (NDP) to:
  - Discover other nodes on the link
  - Determine their link-layer addresses to find routers
  - Maintain reachability information about the paths to active neighbors
- NDP is vulnerable to various attacks if it is not secured
- RFC3971 specifies security mechanisms for NDP
  - Unlike those in the original NDP specifications, these mechanisms do not use IPsec
  - SEND is applicable in environments where physical security on the link is not assured (such as over wireless) and attacks on NDP are a concern
- Implementations are available only for linux ans *BSD:
  - E.g., http://www.docomolabs-usa.com/lab_opensource.html

SEND Elements

- The NDP messages follow the ICMPv6 message format
- An actual NDP message includes
  - an NDP message header
    - ICMPv6 header
    - ND message-specific data
  - and zero or more NDP options, which are formatted in the Type-Length-Value format



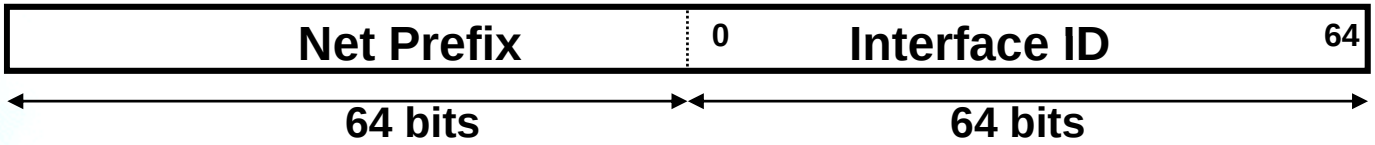
- To secure the NDP, a set of new Neighbor Discovery options is introduced and used to protect NDP messages
- RFC3971 introduces
  - SEND's Neighbor Discovery options
  - An authorization delegation discovery process
  - An address ownership proof mechanism
  - And requirements for the use of these components in NDP

SEND's Behavior and CGAs

- SEND is based in the use of CGAs (Cryptographically Generated Addresses)
- A CGA is an IPv6 address generated following RFC3972:
  - RFC3972 describes a Method for binding a public signature key to an IPv6 address in the context of SEND protocol
  - A pair of public-private keys are generated in the node using SEND
  - The interface ID is generated by computing a cryptographic one-way hash function from a node's public key and auxiliary parameters
  - CGAa are IPv6 addresses with the interface ID generated using this method
  - Then, the messages sent from an IPv6 address can be protected by attaching the public key and auxiliary parameters and by signing the message with the corresponding private key
  - The binding between the public key and the address can be verified by re-computing the hash value and by comparing the hash with the interface identifier
  - The protection works without a certification authority or any security infrastructure
  - Nodes using SEND MUST use CGAs
- The node using SEND is authenticated by
  - The use of the public key used to generate the CGA
  - Signing the message using its private key

IPv6 CGA Format

- RFC3972 considers
  - IPv6 addresses: leftmost 64 bits = subnet prefix, rightmost 64 bits = interface ID
  - Bits of the interface ID are numbered starting from bit zero on the left
  - A CGA has a security parameter (Sec) that determines its strength against brute-force attacks
    - Sec parameter is a 3-bit unsigned integer, and it is encoded in the three leftmost bits (i.e., bits 0 - 2) of the interface ID
    - i.e. Sec = (interface identifier & 0xe000000000000000) >> 61



- The CGA is associated with a set of parameters that consist of a public key and auxiliary parameters
  - Two hash values Hash1 (64 bits) and Hash2 (112 bits) are computed from the parameters
- A CGA satisfies the following two conditions:
  - The first hash value, Hash1, equals the interface identifier of the address. Bits 0, 1, 2, 6, and 7 (i.e., the bits that encode the security parameter Sec and the "u" and "g" bits from the standard IPv6 address architecture format of interface ID (RFC3513)) are ignored in the comparison
  - The 16*Sec leftmost bits of the second hash value, Hash2, are zero
  - The above definition can be stated in terms of the following two bit masks:
    Mask1 (64 bits) = 0x1cffffffffffffff
    Mask2 (112 bits) = 0x00000000000000000000000000000000 if Sec=0,
    0xffff0000000000000000000000000000 if Sec=1,
    0xffffffff00000000000000000000000 if Sec=2,
    0xffffffffffff000000000000000000 if Sec=3,
    0xffffffffffffffff0000000000000000 if Sec=4,
    0xffffffffffffffffff00000000000000 if Sec=5,
    0xffffffffffffffffffffff0000000000 if Sec=6, and
    0xffffffffffffffffffffffff if Sec=7

- Then, a CGA is an IPv6 address for which the following two equations hold:
  Hash1 & Mask1 == interface identifier & Mask1
  Hash2 & Mask2 == 0x00000000000000000000000000000000

SEND's Neighbor Discovery Options (1)

- CGA (Cryptographically Generated Addresses) option, is used to carry the public key and associated parameters
  - CGA are used to make sure that the sender of a Neighbor Discovery message is the "owner" of the claimed address
  - A public-private key pair is generated by all nodes before they can claim an address
  - RFC3971 also allows a node to use non-CGAs with certificates that authorize their use. The details of such use are beyond the scope of this specification and are left for future work
- Format:

Bits 8	Bits 16	24	32
Type = 11	**Length**	**Pad Length**	**Reserved**
CGA Parameters			
Padding			

SEND's Neighbor Discovery Options (2)

- RSA (RSA Encryption Standard) Signature option, is used to protect all messages relating to Neighbor and Router discovery
  - Public key signatures protect the integrity of the messages and authenticate the identity of their sender
  - The RSA Signature option allows public key-based signatures to be attached to NDP messages
- Format:

Bits 8	Bits 16	32
Type = 12	**Length**	**Reserved**
Key Hash (128-bit)		
Digital Signature (PKCS#1 v1.5 signature)		
Padding		

SEND's Neighbor Discovery options (3)

- Timestamp and Nonce Options are introduced in order to prevent replay attacks
  - The Timestamp option offers replay protection without any previously established state or sequence numbers. For example, It can be used when Neighbor and Router Discovery messages are sent in some cases to multicast addresses
  - The Nonce option protects the messages used in solicitation-advertisement pairs
    - Nonce is an unpredictable random or pseudo-random number generated by a node and used exactly once. In SEND, nonces are used to assure that a particular advertisement is linked to the solicitation that triggered it

Authorization Delegation Discovery process (1)

- Background
  - NDP allows a node to configure itself automatically based on information learned shortly after connecting to a new link
  - It is particularly easy to configure "rogue" routers on an unsecured link, and it is particularly difficult for a node to distinguish between valid and invalid sources of router information, because the node needs this information before communicating with nodes outside of the link
  - As the newly-connected node cannot communicate off-link, it cannot be responsible for searching information to help validate the router(s)
  - However, given a certification path, the node can check someone else's search results and conclude that a particular message comes from an authorized source
  - In the typical case, a router already connected beyond the link can communicate if necessary with off-link nodes and construct a certification path
- Certification path
  - SEND Protocol mandates
    - A Certificate format
    - Introduces two new ICMPv6 messages used between hosts and routers
  - They allow the host to learn a certification path with the assistance of the router

Authorization Delegation Discovery process (2)

- Authorization Model
  - To protect Router Discovery, SEND requires that routers be authorized to act as routers
  - This authorization is provisioned in both routers and hosts
  - Routers are given certificates from a trust anchor, and the hosts are configured with the trust anchor(s) to authorize routers
  - This provisioning is specific to SEND and does not assume that certificates already deployed for some other purpose can be used
- The Authorization for routers is two fold:
  - Routers are authorized to act as routers
    - The router belongs to the set of routers trusted by the trust anchor
    - All routers in this set have the same authorization
  - Optionally, routers may also be authorized to advertise a certain set of subnet prefixes
    - A specific router is given a specific set of subnet prefixes to advertise; other routers have an authorization to advertise other subnet prefixes
    - Trust anchors may also delegate a certain set of subnet prefixes to someone (such as an ISP) who, in turn, delegates parts of this set to individual routers

Certificate Format (1)

- The certification path of a router terminates in a Router Authorization Certificate that authorizes a specific IPv6 node to act as a router
  - Because authorization paths are not a common practice in the Internet, the path MUST consist of standard Public Key Certificates (PKC)
  - The certification path MUST start from the identity of a trust anchor shared by the host and the router. This allows the host to anchor trust for the router's public key in the trust anchor
  - Note that there MAY be multiple certificates issued by a single trust anchor.
- Router Authorization Certificates
  - Are X.509v3 certificates (RFC3280)
  - They SHOULD contain at least one instance of the X.509 extension for IP addresses (RFC3779)

Certificate Format (2)

- Example of a certification path.
  - Suppose that isp_group_example.net is the trust anchor. The host has this certificate:

Certificate 1:
Issuer: isp_group_example.net
Validity: Jan 1, 2004 through Dec 31, 2004
Subject: isp_group_example.net
Extensions:
IP address delegation extension:
Prefixes: P1, ..., Pk
... possibly other extensions ...
... other certificate parameters ...

  - When the host attaches to a link served by router_x.isp_foo_example.net, it receives the following certification path:

Certificate 2:
Issuer: isp_group_example.net
Validity: Jan 1, 2004 through Dec 31, 2004
Subject: isp_foo_example.net
Extensions:
IP address delegation extension:
Prefixes: Q1, ..., Qk
... possibly other extensions ...
... other certificate parameters ...

Certificate 3:
Issuer: isp_foo_example.net
Validity: Jan 1, 2004 through Dec 31, 2004
Subject: router_x.isp_foo_example.net
Extensions:
IP address delegation extension:
Prefixes R1, ..., Rk
... possibly other extensions ...

... other certificate parameters ...

4.4. Autoconfiguration

Autoconfiguration

- The standard specifies the steps a host takes in deciding how to autoconfigure its interfaces in IPv6.
- The autoconfiguration process includes creating a link-local address and verifying its uniqueness on a link, determining what information should be autoconfigured (addresses, other information, or both), and in the case of addresses, whether they should be obtained through the stateless mechanism, the stateful mechanism, or both.
- IPv6 defines both a stateful and stateless address autoconfiguration mechanism.
- Stateless autoconfiguration requires no manual configuration of hosts, minimal (if any) configuration of routers, and no additional servers.

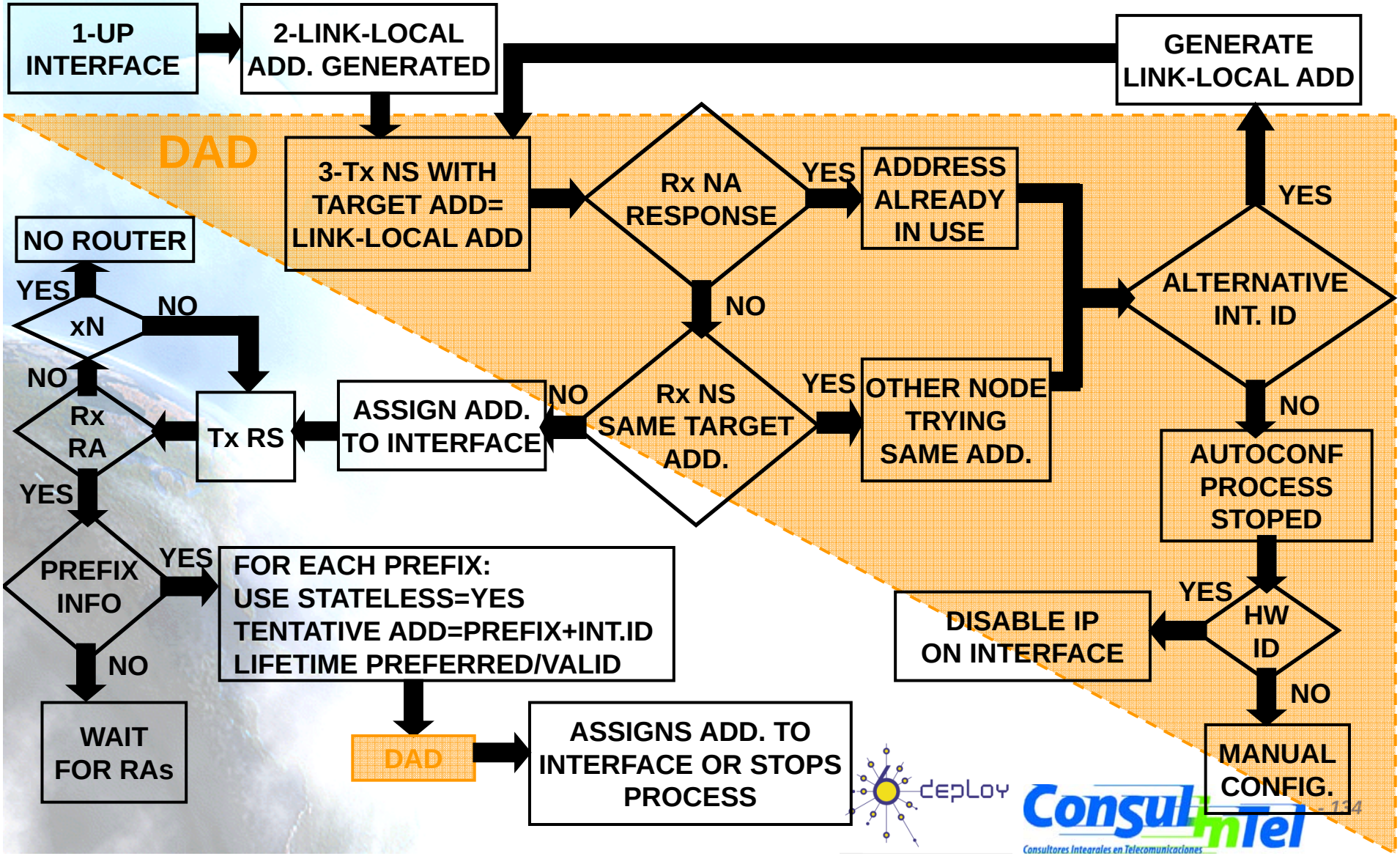
Stateless or Serverless Autoconfiguration (RFC4862)

- Stateless mechanism allows a host to generate its own addresses using a combination of locally available information and information advertised by routers.
- **Routers advertise prefixes** that identify the subnet(s) associated with a link.
- **Hosts generate an "interface identifier"** that uniquely identifies an interface on a subnet, locally generated, e.g., using MAC address.
- An address is formed by combining the both.
- In the absence of routers, a host can only generate link-local addresses.
- Link-local addresses are sufficient for allowing communication among nodes attached to the same link.

Benefits of Stateless Autoconfiguration

- There is no need of manual configuration of each individual machine before connecting them to the network.
- Small sites consisting of a set of machines attached to a single link should not require the presence of a DHCPv6 server or router for communicating. Link-local addresses would be used.
- A large site should not require the presence of a DHCPv6 server for address configuration.
- Facilitate the renumbering of a site's machines through the leasing lifetime and the assignment of multiple addresses to the same interface.

Stateless Autoconfiguration Behavior



Prefix Information Option Format

Bits 8	16	24	32		
Type = 3	**Length = 4**	**Prefix Length**	**L**	**A**	**Reserved1 = 0**
Valid Lifetime					
Preferred Lifetime					
Reserved2 = 0					
Prefix					

- **L**(1bit): **on-link flag**=1 indicates that this prefix can be used for on-link determination.
- **A**(1bit): **autonomous address-configuration flag**=1 indicates that this prefix can be used for stateless address autoconfiguration.
- **Valid Lifetime**: Time in secs. that the prefix is valid for the purpose of on-link determination. Also used in stateless address autoconfiguration.
- **Preferred Lifetime**: Time in secs. that addresses generated from the prefix via stateless address autoconfiguration remain preferred.
- **Prefix** (128 bits): IP Address or prefix of an IP address.

Stateful Autoconfiguration or DHCPv6 (RFC3315)

- Hosts obtain interface addresses and/or configuration information and parameters from a server.
- Servers maintain a database that keeps track of which addresses have been assigned to which hosts.
- Stateless and stateful autoconfiguration complement each other.
- Both stateful and stateless address autoconfiguration may be used simultaneously.
- The site administrator specifies which type of autoconfiguration to use through the setting of appropriate fields in Router Advertisement messages.

Address Life Time

- IPv6 addresses are leased to an interface for a fixed (possibly infinite) length of time, that indicates how long the address is bound to an interface.
- When a lifetime expires, the binding (and address) become invalid and the address may be reassigned to another interface elsewhere in the Internet.
- To handle the expiration of address bindings gracefully, an address goes through two distinct phases while assigned to an interface.
  - Initially, an address is "preferred", meaning that its use in arbitrary communication is unrestricted.
  - Later, an address becomes "deprecated" in anticipation that its current interface binding will become invalid.

Duplicate Address Detection

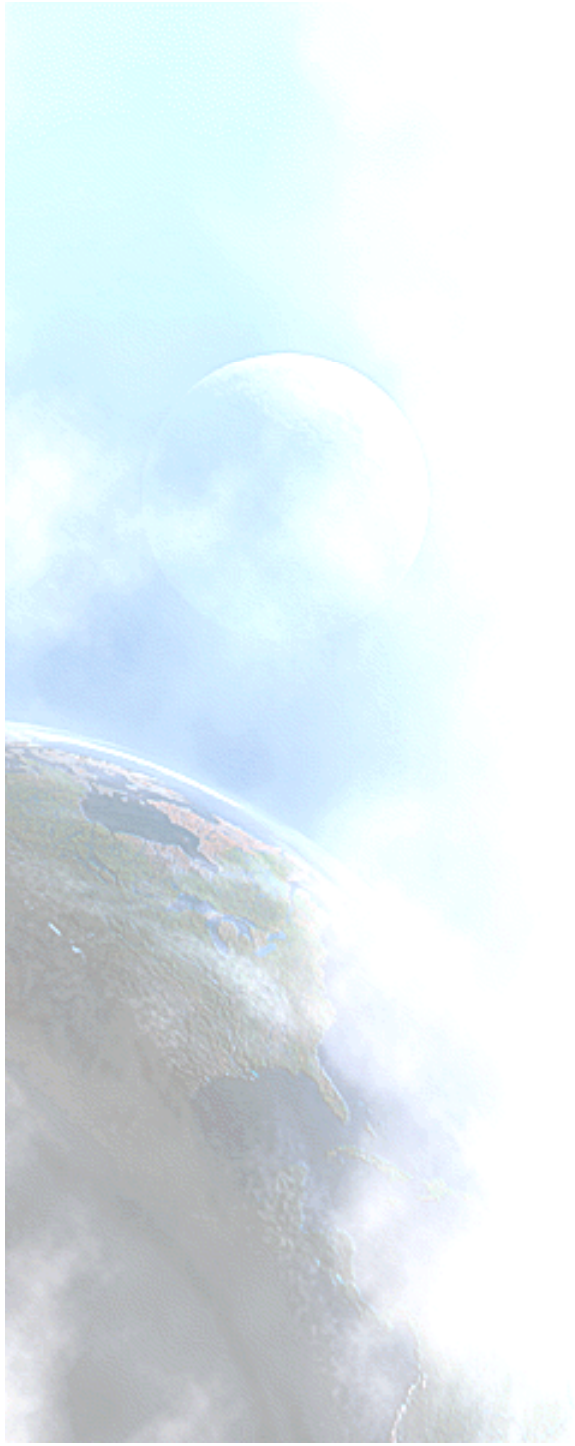
- To insure that all configured addresses are likely to be unique on a given link, nodes run a "duplicate address detection" algorithm on addresses before assigning them to an interface.
- The Duplicate Address Detection algorithm is performed on all addresses, independent of whether they are obtained via stateless or stateful autoconfiguration.
- The procedure for detecting duplicate addresses uses Neighbor Solicitation and Advertisement messages.
- Since host autoconfiguration uses information advertised by routers, routers will need to be configured by some other means. However, it is expected that routers will generate link-local addresses using the same mechanism.
- Routers are expected to successfully pass the Duplicate Address Detection procedure on all addresses prior to assigning them to an interface.

DNS Server Configuration Using Stateless Autoconfiguration (1)

- There were two ways of configuring DNS servers on the node:
  - Manually
  - Using DHCPv6 or DHCPv4 (in case of duals-stack nodes)
- In some environments this could be a problem:
  - Need to use two protocols in IPv6 (Stateless Autoconfguration and DHCPv6)
  - Delay on obtaining the DNS server address when DHCP is used
  - In wireless environments, where the node frequently change of network, it is not possible to use manual configuration or the DHCP delay could be too much
- Recently a new way of configuring DNS servers was introduced (RFC5006), the Recursive DNS Server (RDNSS) Option for the RA.
  - It could be used together with DHCPv6.

DNS Server Configuration Using Stateless Autoconfiguration (2)

- Works the same as for routers and prefix learning using RFC4862 ND: IPv6 Sateless Address Autoconfiguration.
- With the RDNSS option the nodes learn, with only one message exchange:
  - Prefix to be used for autoconfiguration
  - Recursive DNS Servers
- If, in addition to the RDNSS option, DHCPv6 is used, then the “O” Flag should be set in the RA
- Two options to configure the RDNSS option on the routers:
  - Manually
  - Automatically, being a DHCPv6 client



4.5. DHCPv6

DHCPv6 (RFC3315 – RFC4361)

- DHCPv6 is a client-server-based UDP protocol designed to reduce the IPv6 nodes management cost in those environments whereby control of IPv6 address allocation is required and/or more control than the one provided by the stateless mechanism about the provision of network parameters is needed
- DHCP reduces the cost of ownership by centralizing the management of network resources such as IP addresses, routing information, OS installation information, directory service information, and other such information on a few DHCP servers, rather than distributing such information in local configuration files among each network node
- DHCPv6 provides a superset of features, and benefits from the additional features of IPv6 and freedom from BOOTP -backward compatibility constraints

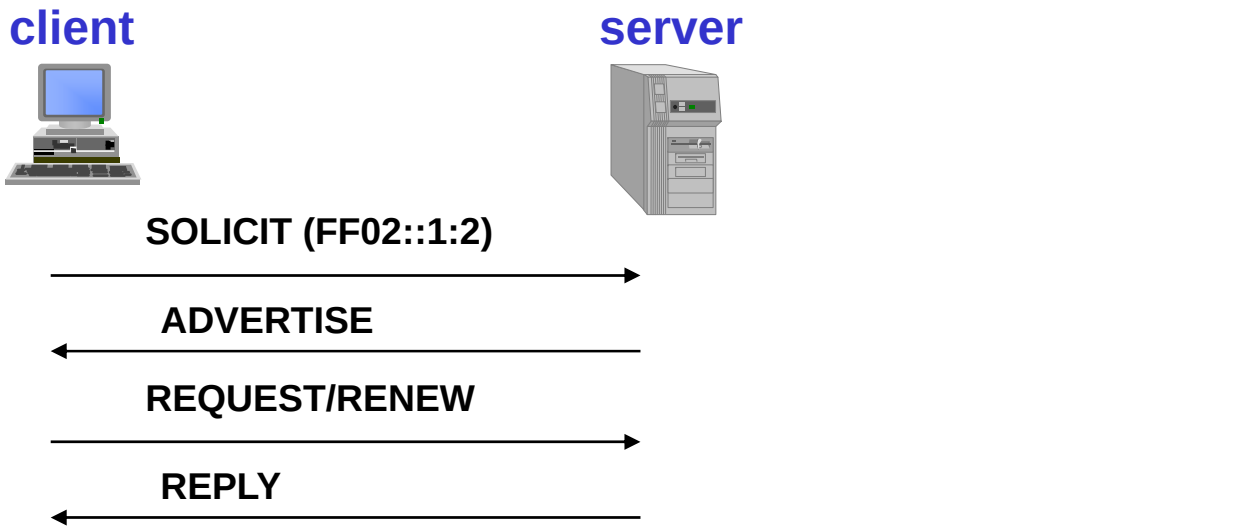
Goals of DHCPv6

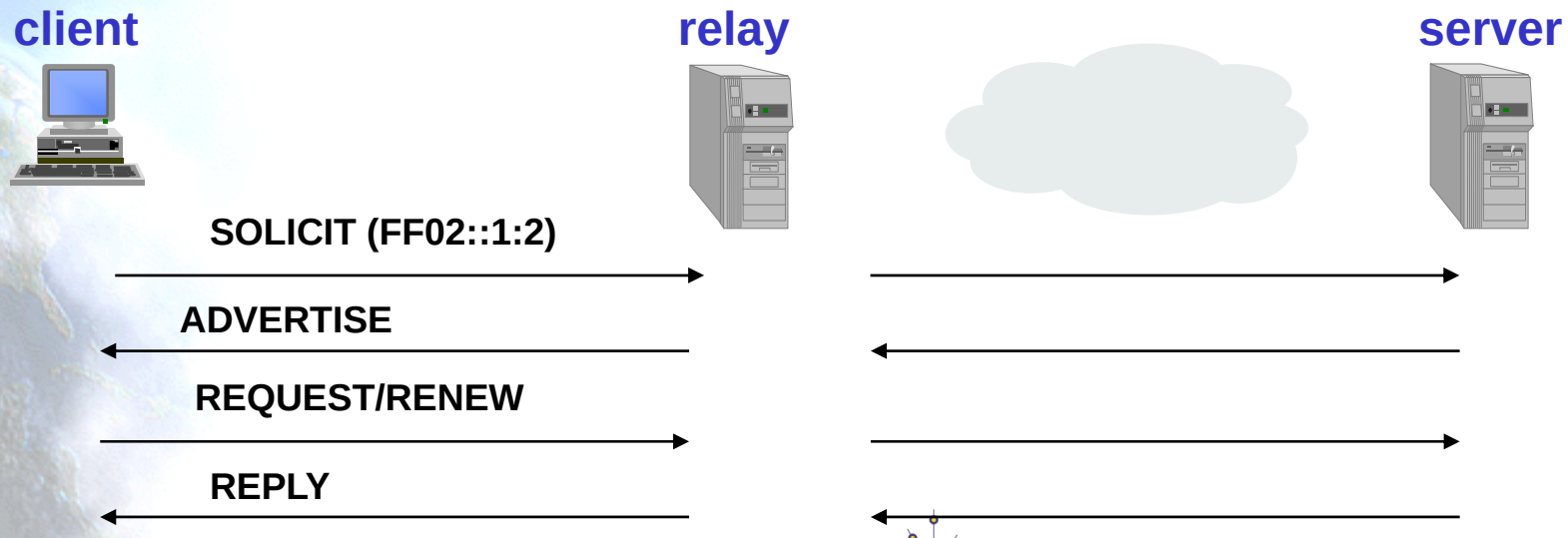
- DHCP is a mechanism rather than a policy. Network administrators set their administrative policies through the configuration parameters they place upon the DHCP servers in the DHCP domain they're managing. DHCP is simply used to deliver parameters according to that policy to each of the DHCP clients within the domain
- DHCP is compatible with IPv6 stateless autoconf
- DHCP does not require manual configuration of network parameters on DHCP clients, except in cases where such configuration is needed for security reasons. A node configuring itself using DHCP should require no user intervention
- DHCP does not require a server on each link. To allow for scale and economy, DHCP must work across DHCP relays
- DHCP coexists with statically configured, non-participating nodes and with existing network protocol implementations
- DHCP clients can operate on a link without IPv6 routers present
- DHCP will provide the ability to renumber network(s) when required by network administrators
- A DHCP client can make multiple, different requests for configuration parameters when necessary from one or more DHCP servers at any time
- DHCP will contain the appropriate time out and retransmission mechanisms to efficiently operate in environments with high latency and low bandwidth characteristics

DHCPv6 Details

- UDP ports are
  - Clients listens to 546
  - Server and relays listen to 547
- Address for DHCPv6 relay agent and servers
  - FF02::1:2 (link local scope)
  - FF05::1:3 (site scope only for servers)
- DHCP messages
  - SOLICIT
  - ADVERTISE
  - REQUES
  - CONFIRM
  - RENEW
  - REBIND
  - REPLY
  - RELEASE
  - DECLINE
  - RECONFIGURE
  - INFORMATION-REQUEST
  - RELAY-FORW
  - RELAY-REPL
- Each message can carry one or more DHCP options
  - Domain-list
  - DNS-server
  - IA-NA, etc.
- DHCP Unique Identifier (DUID)
  - servers use DUIDs to identify clients for the selection of configuration parameters and in the association of IAs with clients
  - clients use DUIDs to identify a server in messages where a server needs to be identified

Basic DHCPv6 Example





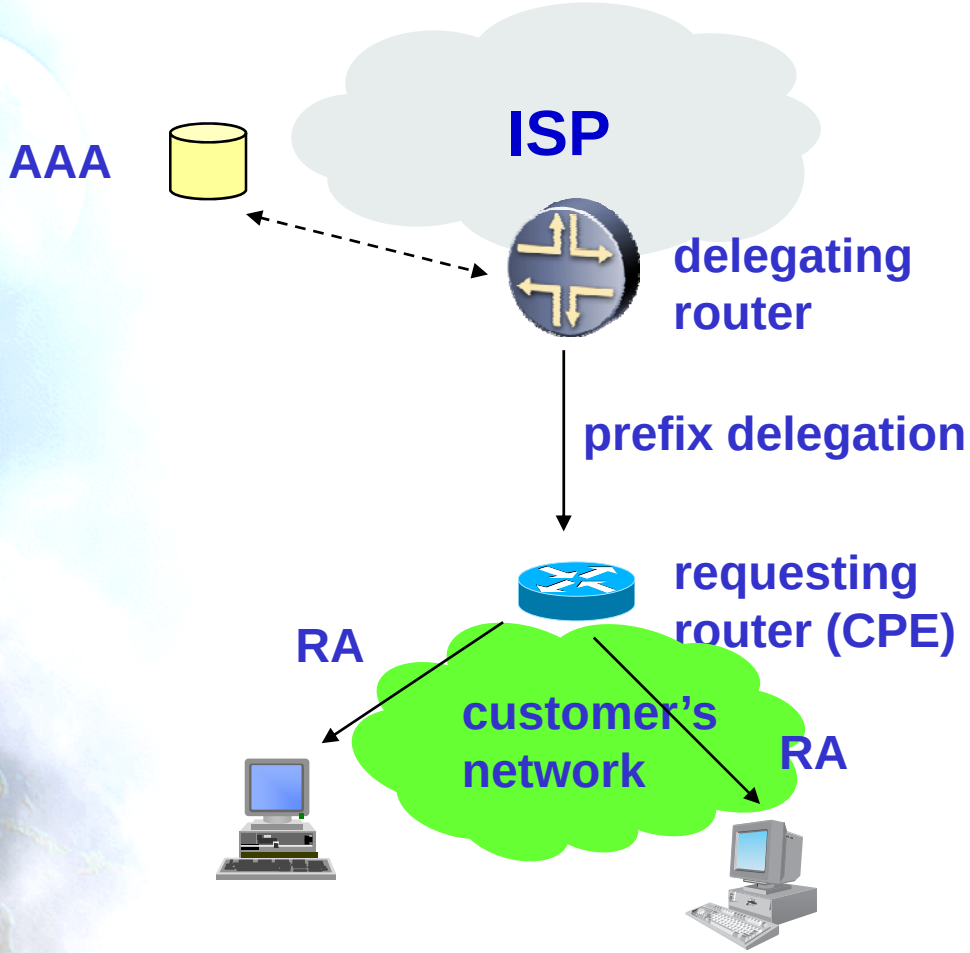
DHCPv6-PD (RFC3633)

- It provides an automated mechanism for the delegation of IPv6 prefixes to authorized requesting routers
- Delegating router does not require knowledge about the topology of the networks to which the requesting router is attached
- Delegating router does not require other information aside from the identity of the requesting router to choose a prefix for delegation
  - for example a ISP to assign a prefix to a CPE device acting as a router

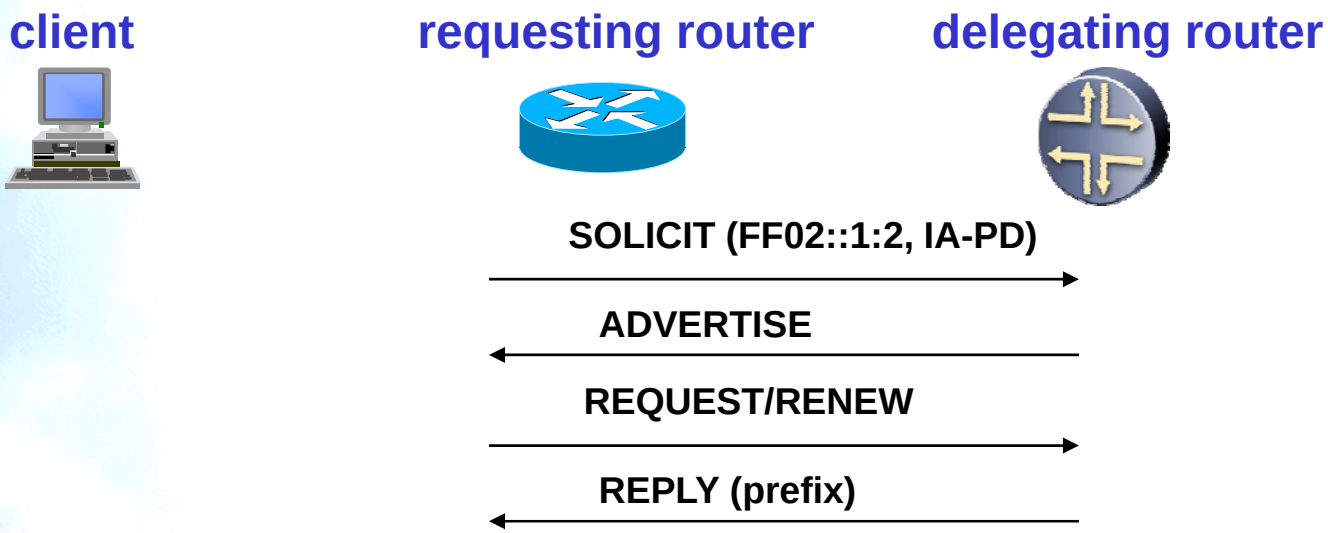
DHCPv6-PD Details

- Requesting router (RR) authentication is needed
- Profile for a RR could be stored in AAA server
- Delegated prefix could be gotten from either:
  - the customer's profile stored in the AAA server
  - prefix pool
- The delegated prefixes have lifetime as IPv6 address in DHCPv6
- DHCPv6-PD doesn't provide a way to propagate the delegated prefix through the customer's network
  - ::/64 prefixes form the delegated prefix are assigned in the RR according to the configured policy
- DHCPv6 relay agents could also be used as in DHCPv6

Network architecture for DHCPv6-PD



Basic DHCPv6-PD Example



New User Features with DHCPv6

- Configuration of Dynamic Updates to DNS.
- Address deprecation, for dynamic renumbering.
- Relays can be preconfigured with server addresses, or use of multicast.
- Authentication.
- Clients can ask for multiple IP addresses.
- Addresses can be reclaimed using the Reconfigure-init message.
- Integration between stateless and stateful address autoconfiguration.
- Enabling relays to locate off-link servers.

4.6. Router Renumbering

RFC2894

- IPv6 Neighbor Discovery and Address Autoconfiguration make initial assignments of address prefixes to hosts.
- These two mechanisms also simplify the reconfiguration of hosts when the set of valid prefixes changes.
- The Router Renumbering ("RR") mechanism allows address prefixes on routers to be configured and reconfigured almost as easily as the combination of Neighbor Discovery and Address Autoconfiguration works for hosts.
- Provides a means for a network manager to make updates to the prefixes used by and advertised by IPv6 routers throughout a site.

Functional Overview

- Router Renumbering Command packets contain a sequence of Prefix Control Operations (PCOs).
- Each PCO specifies an operation, a Match-Prefix, and zero or more Use-Prefixes.
- A router processes each PCO, checking each of its interfaces for an address or prefix which matches the Match-Prefix.
- Applied for every interface on which a match is found.
- The operation is one of ADD, CHANGE, or SET-GLOBAL to instruct the router to respectively add the Use-Prefixes to the set of configured prefixes, remove the prefix which matched the Match-Prefix and replace it with the Use-Prefixes, or replace all global-scope prefixes with the Use-Prefixes.

Thanks !

Contact:

- Jordi Palet Martínez (Consulintel): jordi.palet@consulintel.es
- Alvaro Vives Martínez (Consulintel): alvaro.vives@consulintel.es

6DEPLOY Project

http://www.6deploy.org

The IPv6 Portal:

http://www.ipv6tf.org