Transition Mechanism IPv4 -> IPv6

Sep 17 2019

Alejandro Acosta
alejandro \@ lacnic.net
@ITandNetworking

lacnic
www.lacnic.net

Contents

- Agenda
  - 1. Transition strategies
  - 2. Types of transition mechanisms
  - 3. Introduction to translation
  - 4. Native IPv6
  - 5. Dual stack
  - 6. IPv6-only
  - 7. NAT64/DNS64
  - 8. 464XLAT
  - 9. SIIT-DC

Transition strategies (I)

- We are talking about transitioning to IPv6 and coexistence with IPv4
- Two goals:

1. Achieve IPv6 connectivity
2. Provide a solution the shortage of IPv4 addresses

Transition strategies (II)

- **Three possible strategies** (in order of preference):

1. **Native IPv6:** Native IPv6 packet using the IPv6 header from origin to destination. Two options: **dual-stack** or **IPv6-only**.
2. **Tunneling**: Encapsulating one version of the IP protocol in another.
3. **Translation**: Necessary for communicating two hosts that “speak” only one (different) version of the IP protocol.

Hint: These strategies can be combined

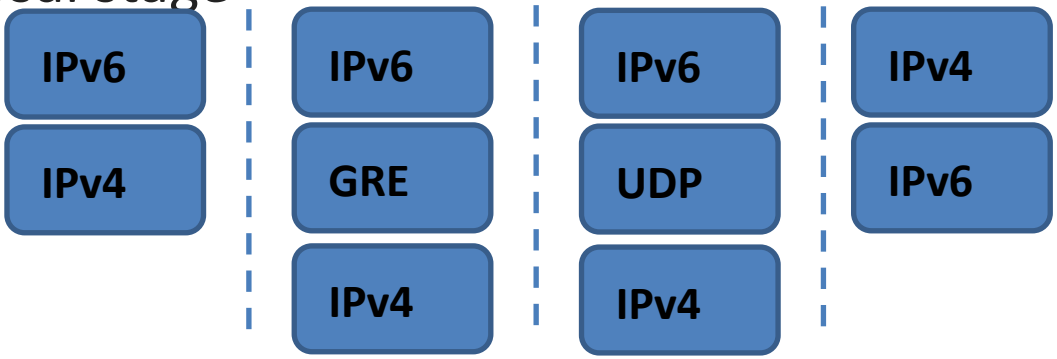
Types of transition mechanisms (I)

- **IPv6 transition mechanisms** (TM): Techniques used to allow IPv6 connectivity and, in some cases, to mitigate the shortage of IPv4 addresses.
- NAT44 and NAT444 (CGN) are not IPv6 TMs. They are mechanisms used to extend the life of public IPv4 addresses.
- We will talk about dual-stack and IPv6-only
- We will see examples of:

1. Tunneling
2. Translation

Types of transition mechanisms (II)

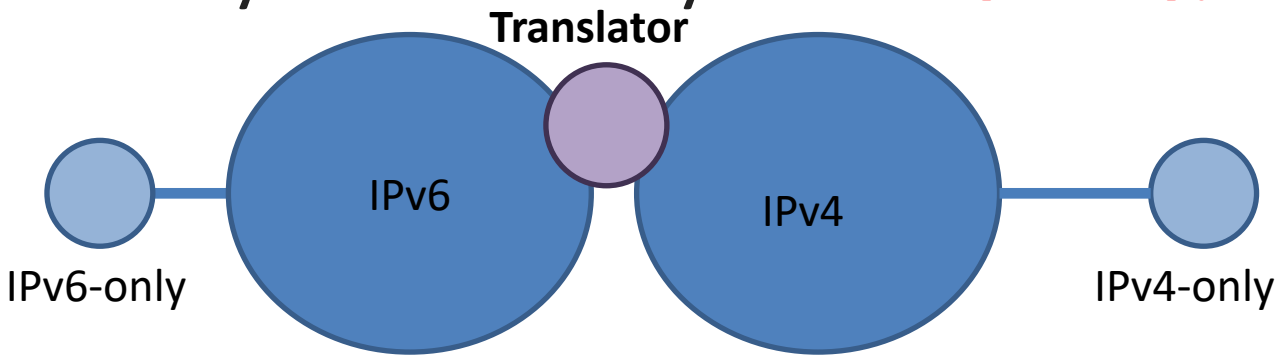
- **Tunnels:** To encapsulate one version of the IP protocol in another
- **Goal**: To traverse a network that does not support IPx
- Static vs. automatic tunnels
- Point-to-point vs. multipoint tunnels
- The most common transition mechanism
- There are many tunneling mechanisms, many of them deprecated or obsolete, others in the development or proposal stage



Types of transition mechanisms (III)

- When communication is necessary between an IPv6-only node and an IPv4-only node, in either direction:

1. IPv6-only → IPv4-only
2. IPv4-only → IPv6-only **OBSOLETE [RFC4966] (NAT-PT)**



Native IPv6

- **Native IPv6:** IPv6 without encapsulation or translation. The IPv6 header is used from origin to destination.
- **Advantages:** Final solution. IPv6 operation is solid and stable, with no need for future changes. The move to IPv6 is completed in a single stage

- Two options:

1. **Dual-stack**: adding native IPv6 to IPv4
2. **IPv6-only**: only native IPv6 on the network

Dual stack (I)

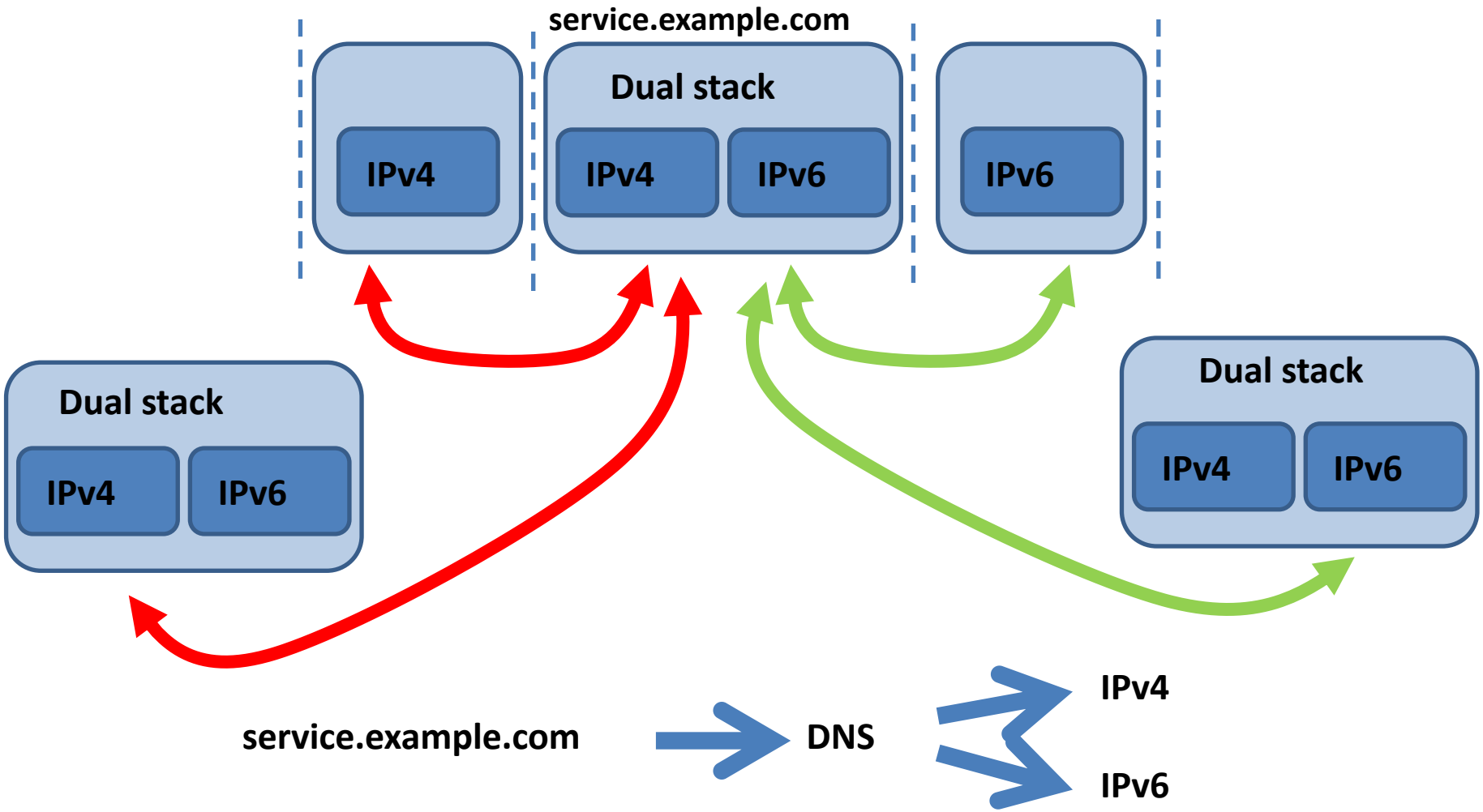
- Both IPv4 and IPv6 are available: the client chooses
- Advantage: Most IPv4-friendly option, allows a gradual transition to IPv6
- Default configuration places preference on IPv6 -> minor changes

Native IPv6
IPv6 with TM
IPv4

Native IPv6
IPv4
IPv6 with TM

- Coexistence with IPv4:
  - Data plane: from IPv4 to IPv6 ---> No problem
  - Control plane: both, IPv4 and IPv6 ---> More complex/expensive

Dual stack (II)



Dual stack (III)

- Solution: Happy Eyeballs [RFC8305] (subs RFC 6555)

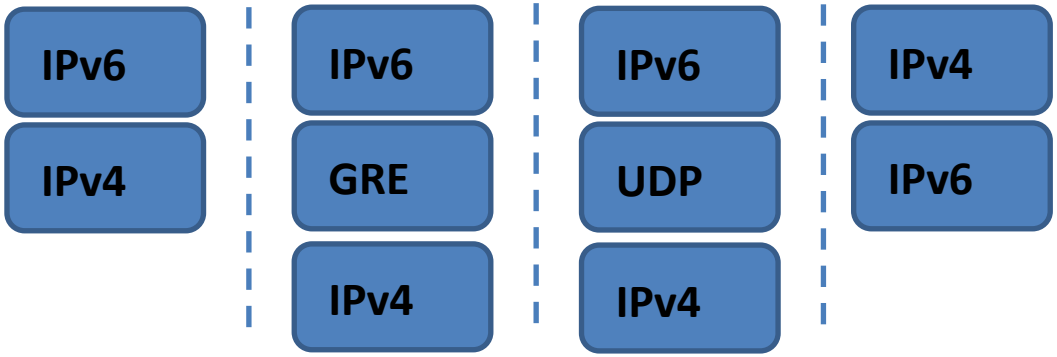
```
   DNS Server                    Client                    Server
       |                            |                          |
 1.    |<--www.example.com A?-----|                          |
 2.    |<--www.example.com AAAA?--|                          |
 3.    |---192.0.2.1-------------->|                          |
 4.    |---2001:db8::1----------->|                          |
 5.    |                            |                          |
 6.    |                            |==TCP SYN, IPv6===>X      |
 7.    |                            |--TCP SYN, IPv4------->|
 8.    |                            |<-TCP SYN+ACK, IPv4----|
 9.    |                            |--TCP ACK, IPv4------->|
10.    |                            |==TCP SYN, IPv6===>X      |
```

IPv6-only

- Reasons:
  - Not enough public IPv4 addresses
  - Easier to manage, monitor and configure
  - New IPv6-only services (WSN - 6Lowpan)
  - IPv6-only server farms with dual-stack front end
  - Implementations are valid “forever”
- What happens with IPv4-only content/services/hosts/applications?
  - DS-Lite: allows offering IPv4 to end users (tunneling)
  - NAT64/DNS64, 464XLAT: translation, incomplete solution
  - As more IPv6 content becomes available, the problem will not be as serious

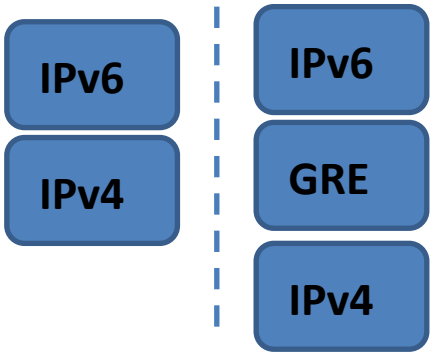
Introduction to tunneling

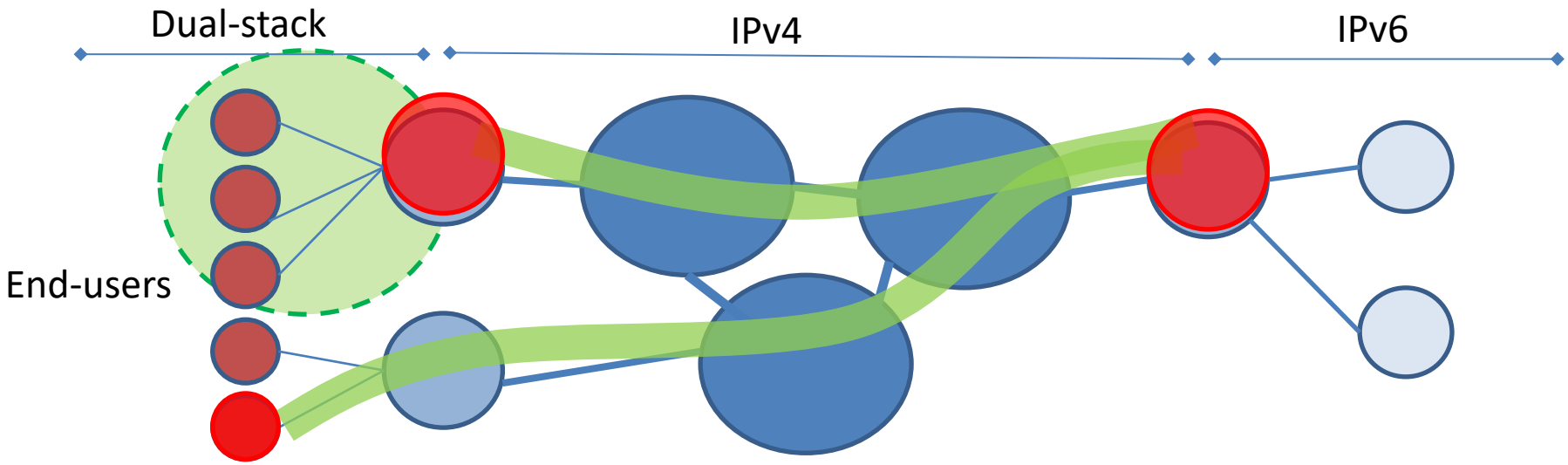
- **Tunnels:** Encapsulate one version of the IP protocol in another
- **Goal**: To traverse a network that does not support IPx
  - Static vs. automatic tunnels
  - Point-to-point vs. multipoint tunnels
  - Most common transition mechanism
  - There are many tunneling mechanisms, many of them deprecated or obsolete, others under development or in the proposal stage



Tunneling: 6in4

- **6in4**:
  - Encapsulates IPv6 (protocol 41) in IPv4
  - Static
  - Point-to-point
  - Manual





Tunneling: 6RD (I)

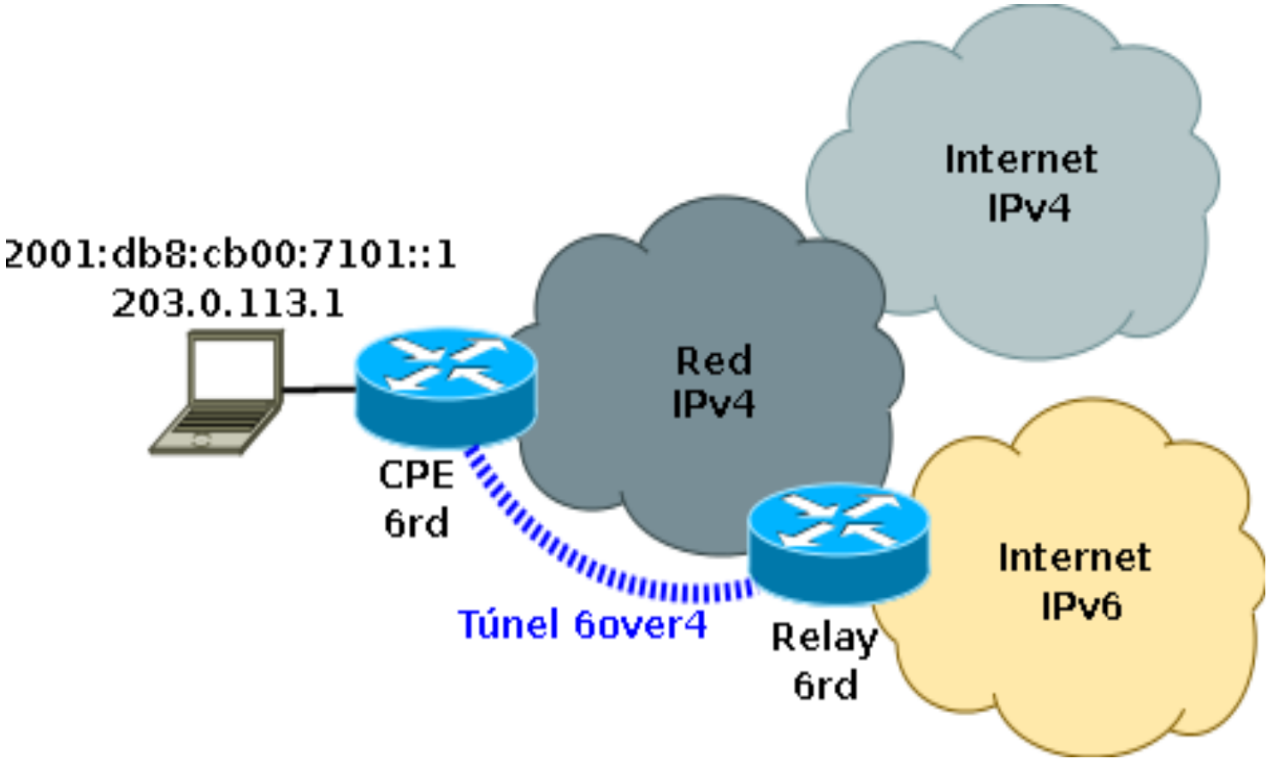
- Developed by "*Free*" a French ISP
  - Detailed in the rfc5569
  - Protocol specification rfc5969
- Developed in only 6 weeks
- Enables IPv6 connectivity for only-IPv4 networks
  - Must be supported by the client equipment (CPE)
- Depends on two components
  - **CPE 6rd**: interface between the operator and the user
  - **Relay 6rd**: interface between the IPv4 network and IPv6 network

lacnic

Tunneling: 6RD (II)

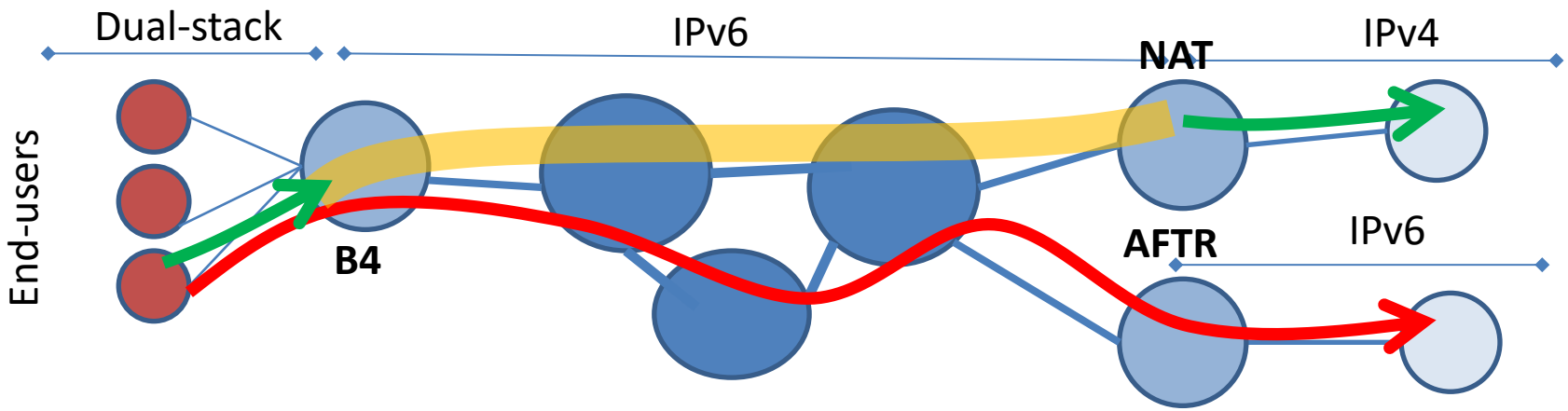
- Device characteristics
  - CPE
    - XDSL modem, cable modem, 3G modem, etc.
    - Modified software to support 6rd
    - Remote management is recommended
  - Relay 6rd
    - Encapsulate/unencapsulate IPv4 <-> IPv6 packets

Tunneling: 6RD (III)

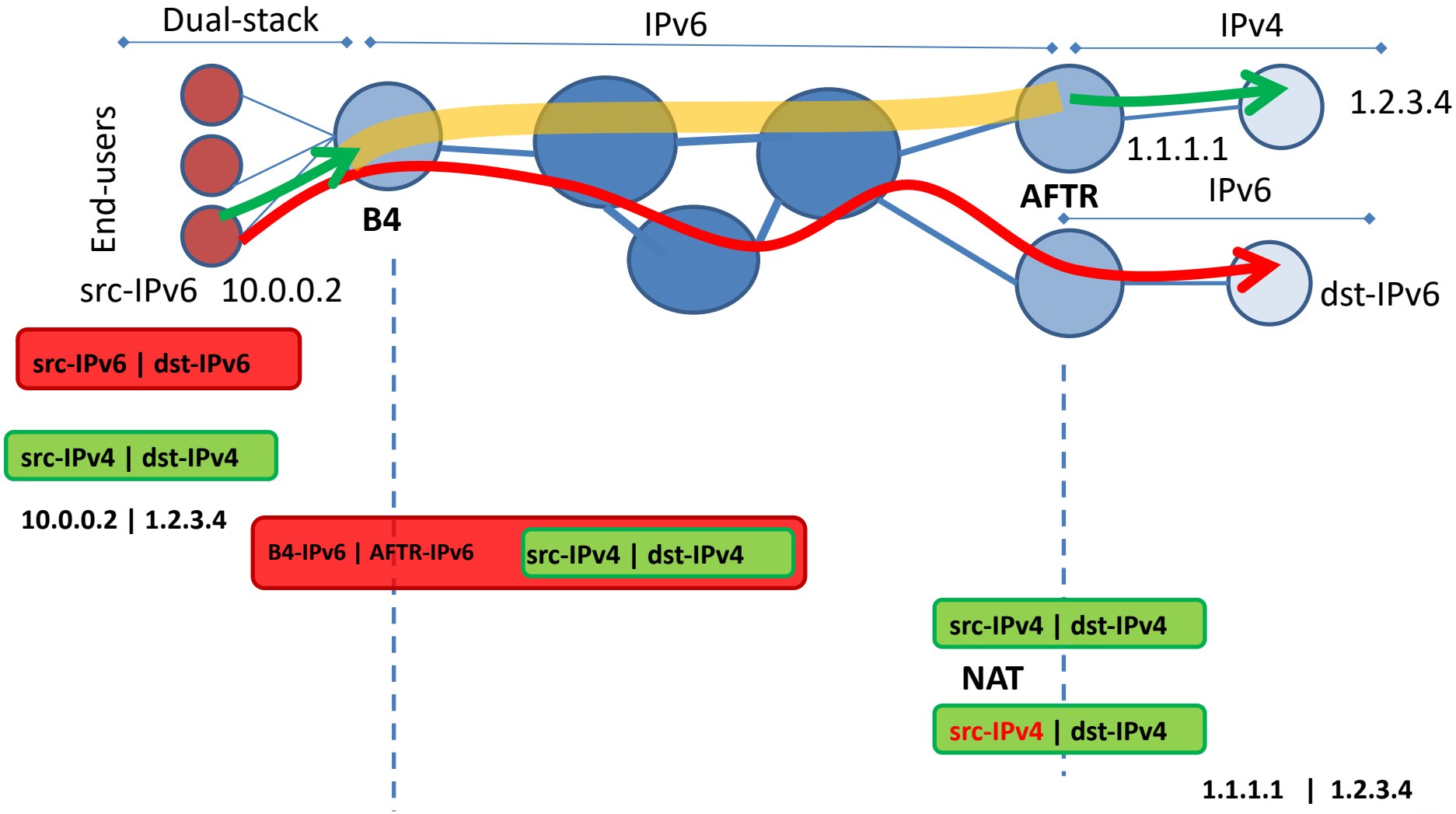


Tunneling: DS-Lite (I)

- DS-Lite:
  - Based on an IPv6 network: native IPv6 traffic
  - IPv4 traffic is encapsulated in IPv6 and sent to a “Big NAT”
  - Public IPv4 addresses are shared
  - Just one level of NAT
  - Name of the “Big NAT”: AFTR (Address Family Transition Router), CGN (Carrier Grade NAT) or LSN (Large Scale NAT)
  - “Client”: B4 (Basic Bridging BroadBand)



Tunneling: DS-Lite (II)



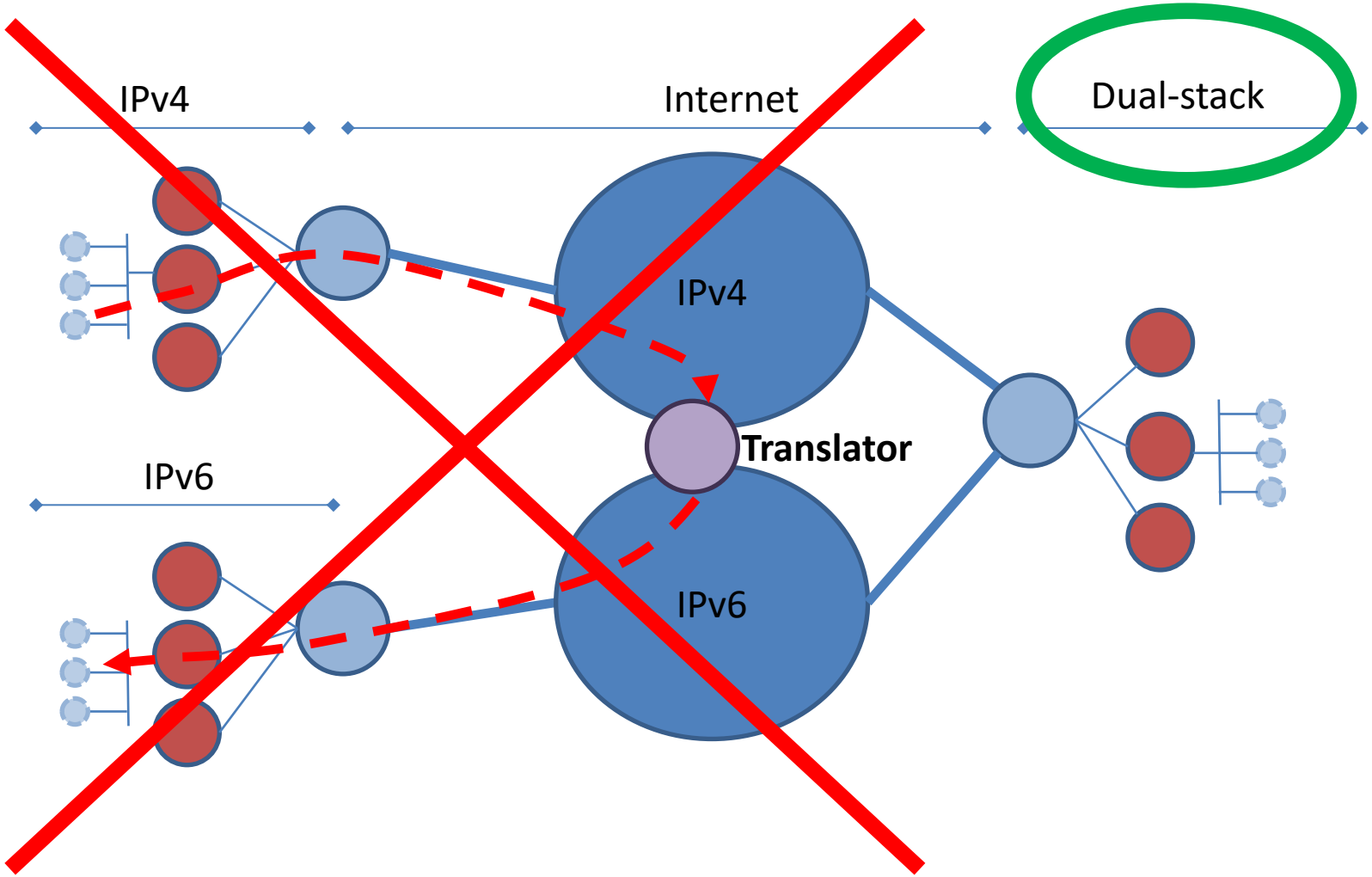
Introduction to translation (I)

- When an IPv6-only node must communicate with an IPv4-only node, in either direction:

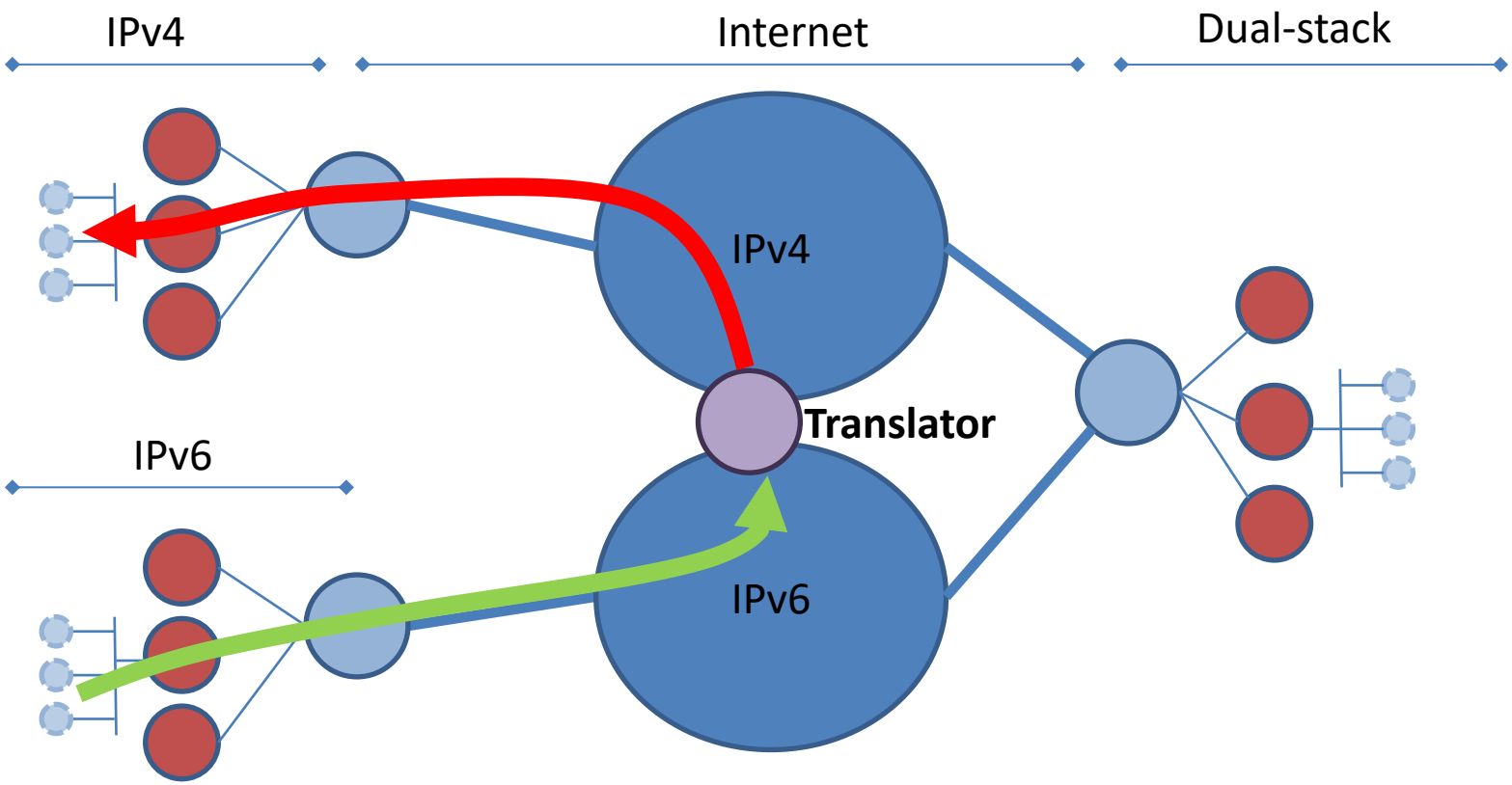
1. IPv6-only → IPv4-only
2. IPv4-only → IPv6-only **OBSOLETE [RFC4966] (NAT-PT)**

- Remember: Use only if dual-stack or tunneling mechanisms are not possible.

Introduction to translation (II)



Introduction to translation (III)



lacnic

NAT64/DNS64 (I)

- NAT64 only translates unicast TCP, UDP, and ICMP
- Users share public IPv4 addresses
- Automatic address translation using static information
- Uses a known prefix (64:ff9b::/96 or others)

```
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|PL| 0-------------32--40--48--56--64--72--80--88--96--104---------|
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|32|     prefix    |v4(32)         | u | suffix                    |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|40|     prefix        |v4(24)     | u |(8)| suffix                |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|48|     prefix            |v4(16) | u | (16)  | suffix            |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|56|     prefix                |(8)| u |  v4(24)   | suffix        |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|64|     prefix                    | u |   v4(32)      | suffix    |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|96|     prefix                                    |    v4(32)     |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
```

NAT64/DNS64 (II)

- IPv6-only nodes must believe that IPv4-only nodes are reachable via IPv6
- DNS64 creates false responses: automatically translates an IPv6 address based on the node's IPv4 address

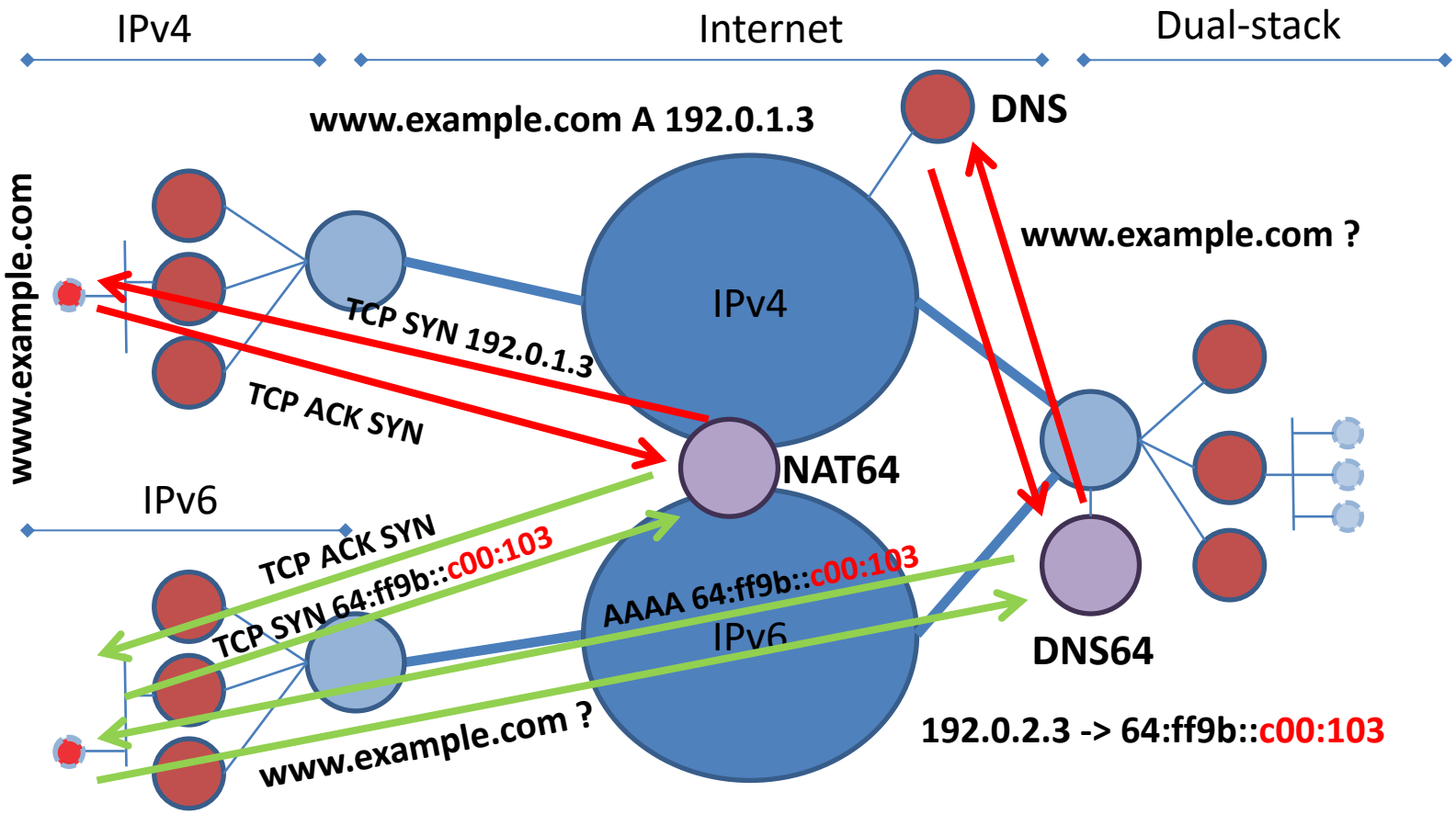
- NAT64/DNS64 are know as stateful NAT64 (v6 network -> v4 Internet)

vs.

- Stateless NAT64 -> without DNS64

1. IPv6 Internet to an IPv4 node
2. 1 to 1 translation

Stateful NAT64/DNS64



NAT64/DNS64 (III)

- **NAT64/DNS64 has certain limitations/issues**:
- Defined only for unicast TCP, UDP, and ICMP
- Applications using layer 3 information in the FTP application layer [RFC6384], SIP/H323 -> require ALG
- Need to add new “boxes”
- NAT64/DNS64 work well with DNS domain names. But what happens if an application tries to use an IP address? It doesn't work
- What happens with applications that don't support IPv6? They're out

464XLAT (I)

- NAT64/DNS64: Work with applications that support IPv6
- **What happens with an IPv4-only application running on an IPv6-only host?**
- **464XLAT**: stateful NAT64 (core) + stateless NAT64 (border)
- Only supports IPv4 in the client-server model, where the server has a global Pv4 address
- CLAT: customer-side translator
  - Algorithmically translates 1:1 private IPv4 addresses to global IPv6 addresses and vice versa
- PLAT: provider-side translator
  - Translates N:1 global IPv6 addresses to public IPv4 addresses and vice versa

464XLAT (II)

```
                                  +------+
                                  |  v6  |
                                  | host |
                                  +--+---+
                                     |
                                .---+---.
                               /        \
                              /   IPv6   \
                             |   Internet  |
                              \          /
                               `--------'
   UE / Mobile Phone                 |
+----------------------+        .---+---.                    .------.
| +----+    |          |       /        \     +------+     /        \
| | v6 +----+   +------+      / IPv6 PDP  \    |      |    /   IPv4   \
| +----+    |   |      |     |  Mobile Core +---+ PLAT +--+  Internet  |
|           +---+ CLAT +---+  \    GGSN   /    |      |    \          /
|           |   |      |      \          /     +------+     `---+---'
| +-----+   |   +------+       `-------'                       |
| | v4p +---+          |                                     +--+---+
| +-----+   |          |                                     | v4g  |
+----------------------+                                     | host |
                                                             +------+

        <- v4p -> XLAT <--------- v6 --------> XLAT <- v4g ->
```

v6 : Global IPv6
v4p : Private IPv4
v4g : Global IPv4
PDP : Packet Data Protocol
GGSN : Gateway GPRS Support Node

SIIT-DC

WHAT is SIIT-DC?

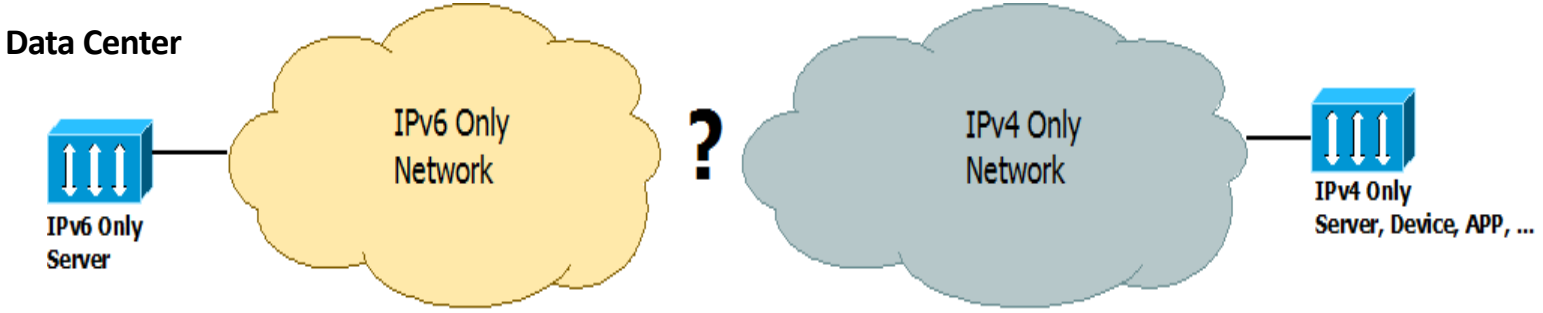
- Stateless IP/ICMP Translation for IPv6 Data Center Environments

What does SIIT-DC resolves?

- Servers in a DC (also worth to mention devices, appliances, aplications, hosts, etc.) **IPv6 Only that** requires connectivity to the **IPv4 Only world (and viceversa)**.

Why SIIT-DC?

- Evolution of DC: IPv4-Only → Proxies → DS → IPv6 Only



SIIT-DC

D) SIIT-DC

Arquitecture

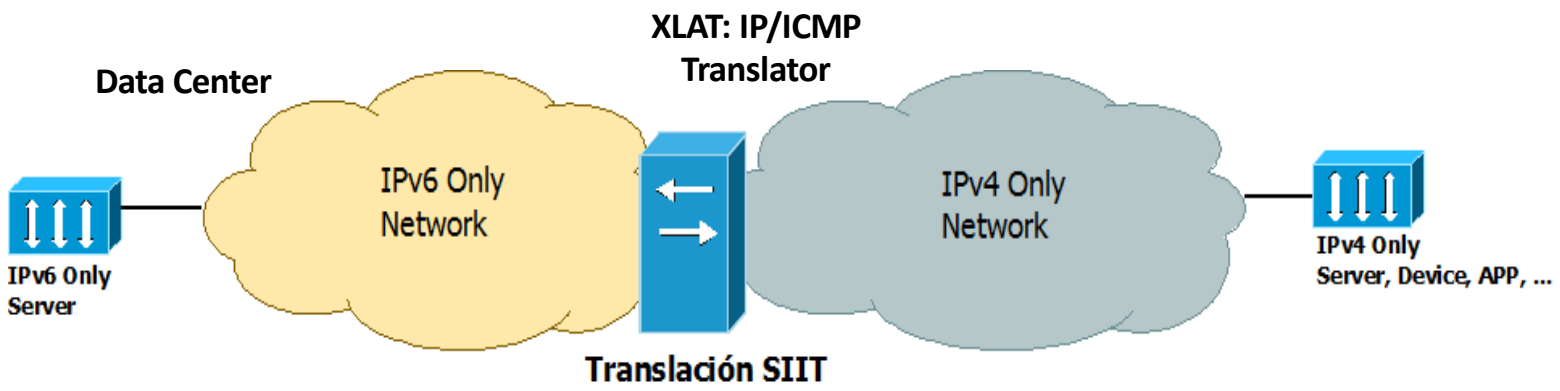
- Translation IPv4 to/from IPv6 and ICMPv4 t/f ICMPv6.
- SIIT-DC: RFC7755.
- Algorithm specified for translation IP/ICMP: RFC7915.
- Translation with EAMT.

Highligths

- Translation IP Header
- L4 payload is copied verbatim.
- 'Stateless'.
- Based in EAM Tables (Explicit Address Mapping – RFC7757).

Cons:

- Only Unicast, no IPv4 Options, no EH for IPv6.



SIIT-DC

D) SIIT-DC

SIIT with EAM (EAM: Explicit Address Mapping)

Example of translation

EAM Table

Entry	IPv4 Prefix	IPv6 Prefix
1	192.168.9.1	2001:db8:aaaa::
2	192.168.77.5/32	2001:db8:bbbb::7/128
3	**182.168.2.16/28**	**2001:db8:ccc::/124**
4	192.0.2.192/29	2001:db8:dddd::/64
5	190.10.2.224/29	2001:db8:eeee:8::/62
6	172.77.2.16/28	64:ff9b::/124

SIIT-EAM example of translations

IPv4/IPv6 a Transladar	IPv6/IPv4 Transladada
192.168.9.1	2001:db8:aaaa::
172.77.2.20	64:ff9b::4
2001:db8:eeee:9::	**190.10.2.225**

Translated IPv6 = EAM6 + (IPv4 to translate – EAM4)

Translated IPv4 = EAM4 + (IPv6 to translate – EAM6)

Questions?
Comment?

Alejandro Acosta
alejandro@lacnic.net
@ITandNetworking

lacnic
www.lacnic.net