



Implementado SIIT y NAT64 utilizando Jool

Mérida, marzo 2020



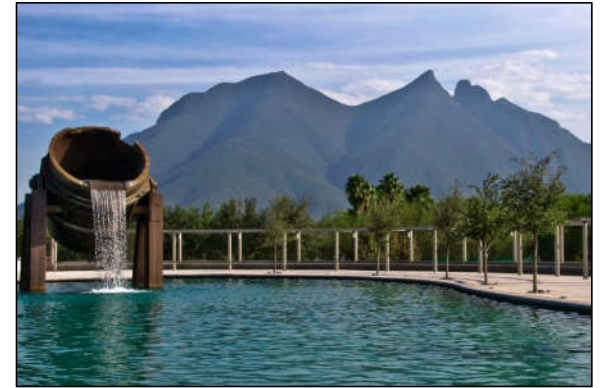
Pacto entre damas y caballeros



Todos estamos aquí para
aprender



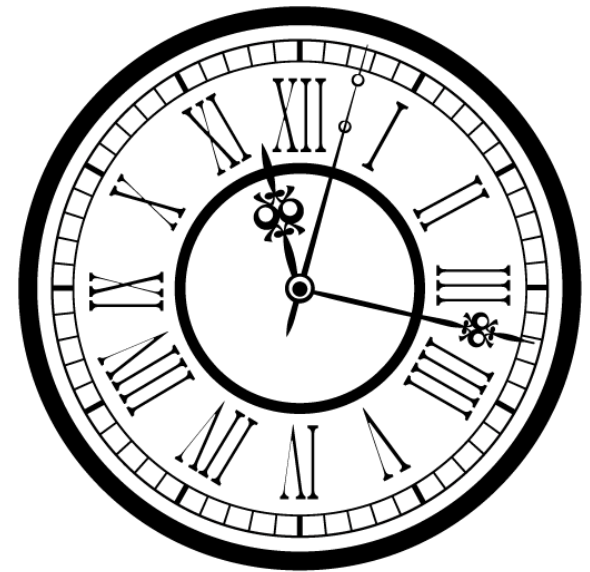
Todos participamos en la
plática



Perdonamos y aceptamos el
acento norteño del instructor

Agenda

- ¿Quiénes somos y qué hacemos?
- ¿Cuál es el problema?
- Mecanismos de transición
 - SIIT Tradicional
 - SIIT-EAM
 - NAT64
 - DNS64
 - 464XLAT
 - SIIT-DC
 - SIIT-DC-DTM
- Jool
 - Instalación
 - Configuración
 - Prácticas



¿Quiénes somos y qué hacemos?

Jorge Cano

- > 18 años trabajando para NIC México
- Inicié como desarrollador de software
- Después como arquitecto de software
- Actualmente como líder del laboratorio de innovación e investigación



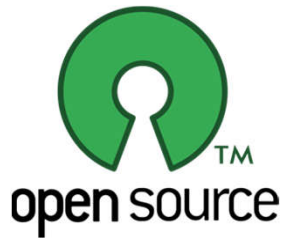
Laboratorio de Investigación

Equipo

- 4 Ingenieros de tiempo completo

Objetivo

- Investigar e innovar en nuevas tecnologías que ayuden al crecimiento y desarrollo de Internet



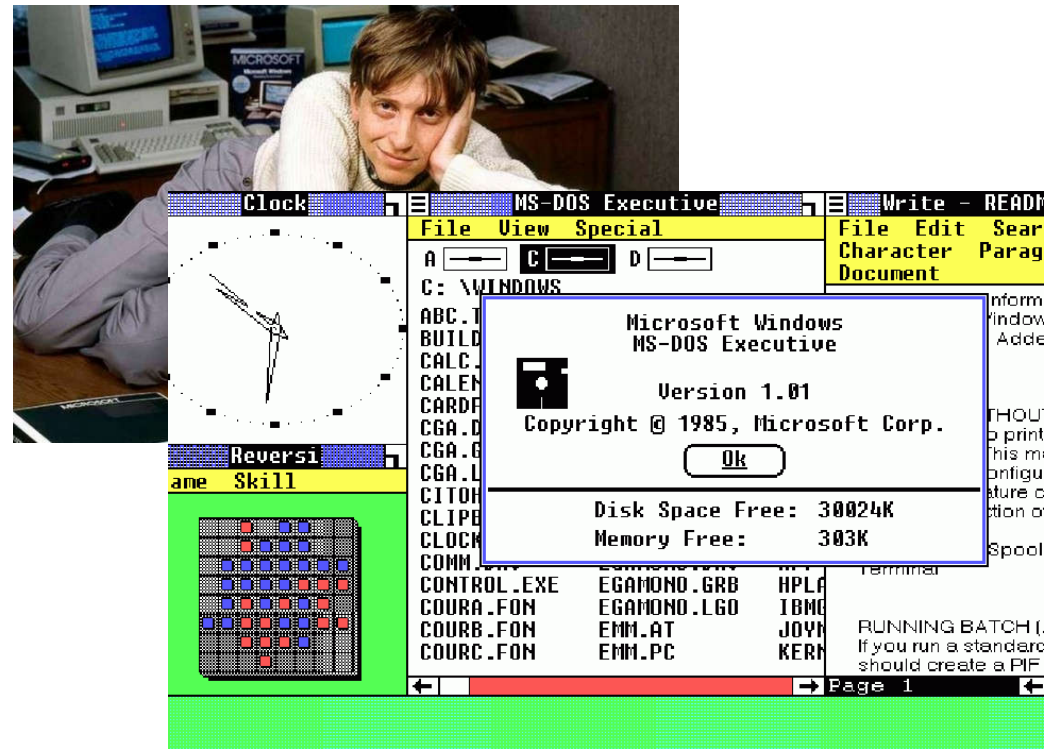
Historia

IPv4 (Internet Protocol) fue implementado por primera vez en 1983.

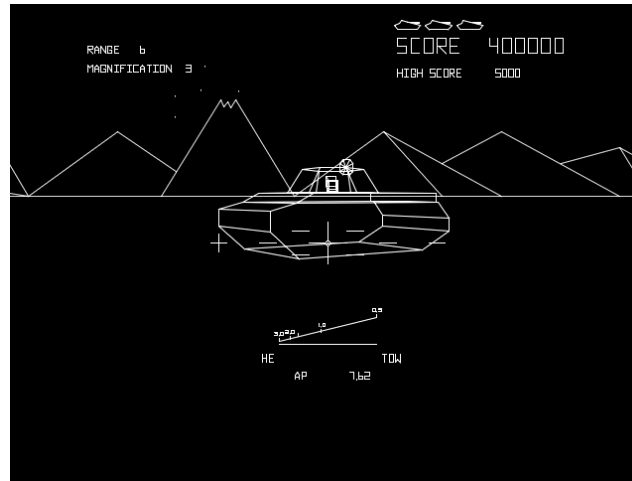


<https://www.speakertv.com/artsfeatures/997111/>

Historia



Historia



HP 800 Series \$113,500 USD
8 MB de RAM, expandible a 24MB.

Historia



¿Cuál es el problema?

RIR	Bloques /22 de IPv4 disponibles
ARIN	Agotados
AFRINIC	~ 26
LACNIC	~ 14
RIPE NCC	Agotados
APNIC	~ 29



Fuente: Guangliang Pan, APNIC

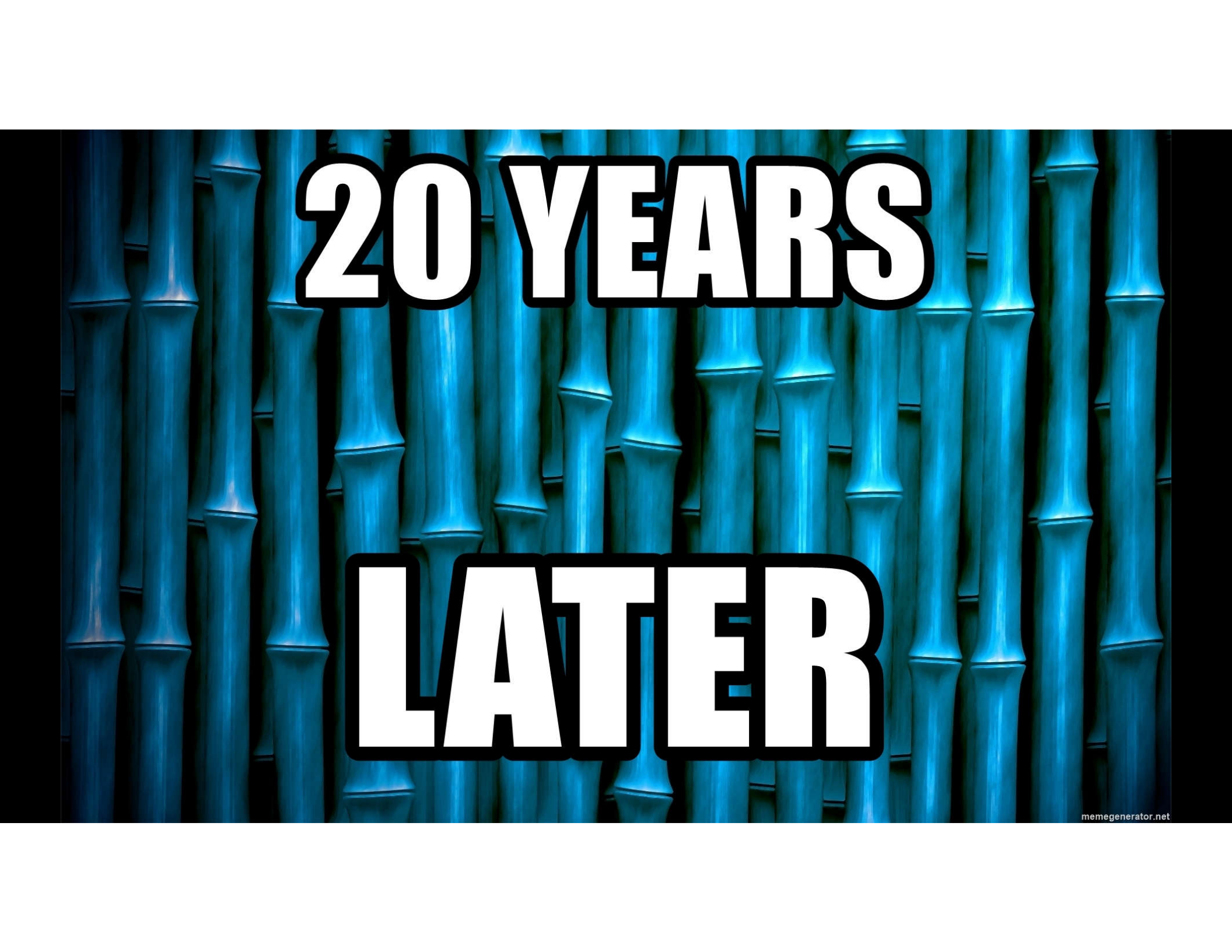
<https://blog.apnic.net/2019/12/13/keep-calm-and-carry-on-the-status-of-ipv4-address-allocation/>

¿Cuál es el problema?

IPv6 fue publicado como Draft Standard en diciembre de 1998.

El plan era utilizar en conjunto IPv4 e IPv6, por un corto período de transición y después sólo IPv6.



The background of the image consists of numerous vertical bamboo stalks. The stalks are rendered in a monochromatic blue and black color scheme, with the blue being a vibrant, slightly cyan-like hue. The lighting creates highlights and shadows on the bamboo's surface, emphasizing its segmented texture. The stalks are arranged in a dense, slightly irregular pattern, filling the entire frame.

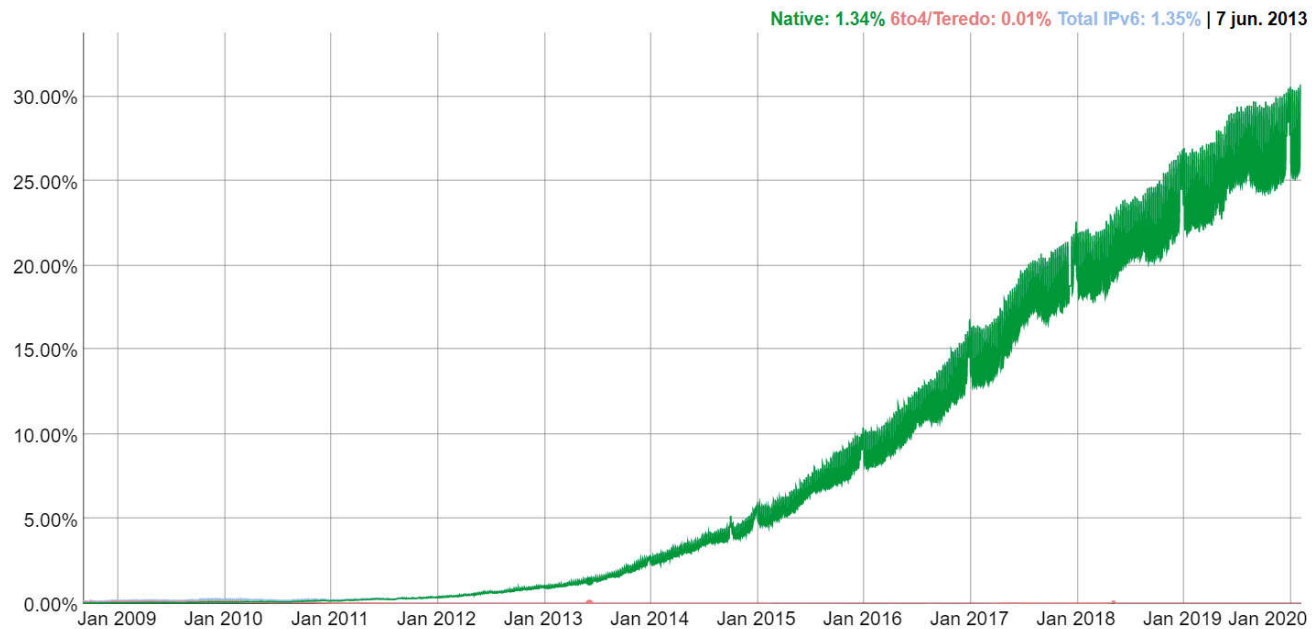
20 YEARS

LATER

¿Cuál es el problema?

IPv6 Adoption

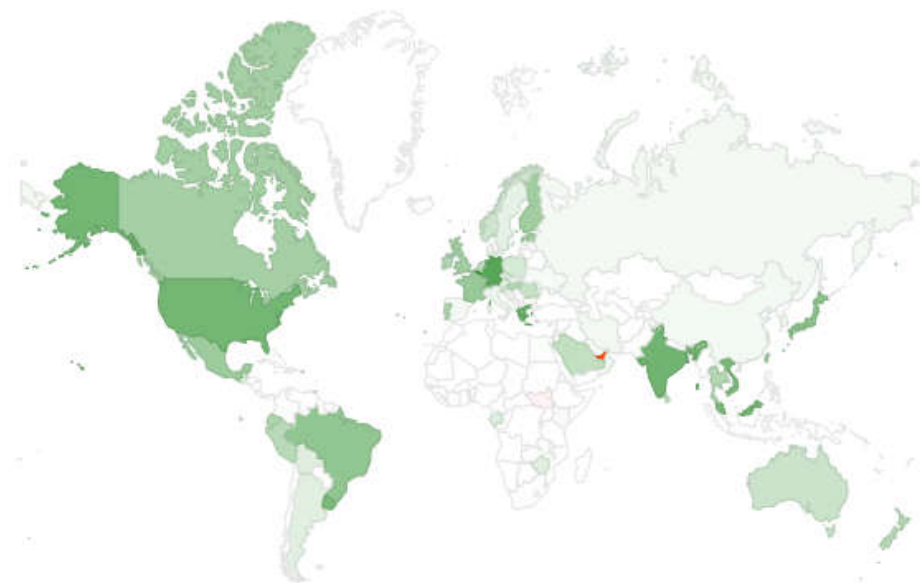
We are continuously measuring the availability of IPv6 connectivity among Google users. The graph shows the percentage of users that access Google over IPv6.



Fuente: <https://www.google.com/intl/en/ipv6/statistics.html>

¿Cuál es el problema?

Per-Country IPv6 adoption



Fuente: <https://www.google.com/intl/en/ipv6/statistics.html>

¿Cuál es el problema?

Agotamiento de IPv4



IPv4 e IPv6 son incompatibles

Snapshots at jasonlove.com



It's a good thing Chuck raised his voice, because Pedro understood loud English.

IPv4 vs. IPv6

IPv4		IPv6
32-bit	Tamaño	128-bit
Decimal 192.0.2.76	Formato	Hexadecimal 2001:0db8:0234:ab00:0123:4567:8901:abcd
192.0.2.0/24	Prefijo	2001:db8:234::/48
2^{32} ~4,294,967,296	Cantidad	2^{128} ~340,282,366,920,938,463,374,607,431,768,211,456

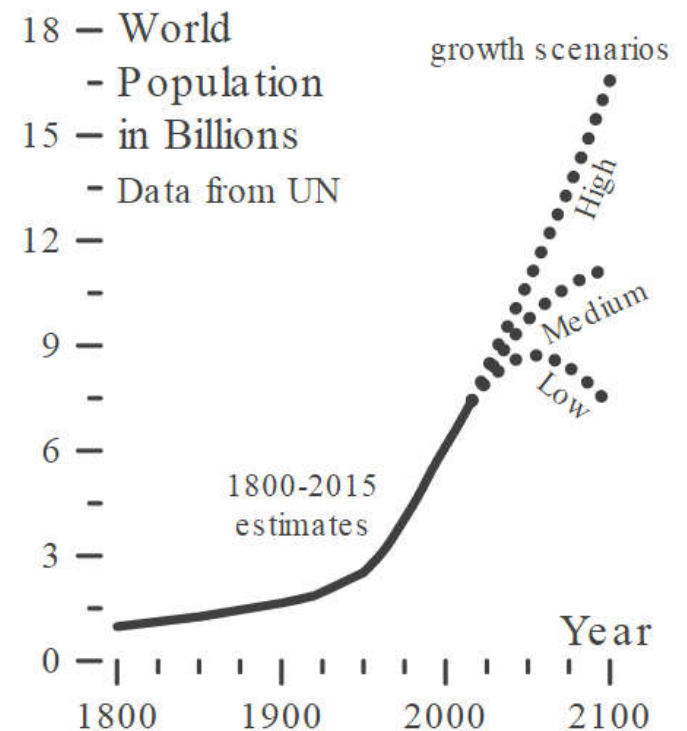
Número de posibles direcciones de IPv6:

340 sextillones 282 mil 366 quintillones 920 mil 938 cuatrillones 463 mil 463 trillones 374 mil 607 billones 431 mil 768 millones 211 mil 456 .

¿Cuándo se agotará IPv6?

- Actualmente somos 7.7 mil millones de personas en el mundo.
- Tomando el escenario en el que la tasa de crecimiento no disminuye (1.1% anual).
- Y dando a cada persona en el mundo un bloque /96

El agotamiento de direcciones IPv6 sería en el año 6021.



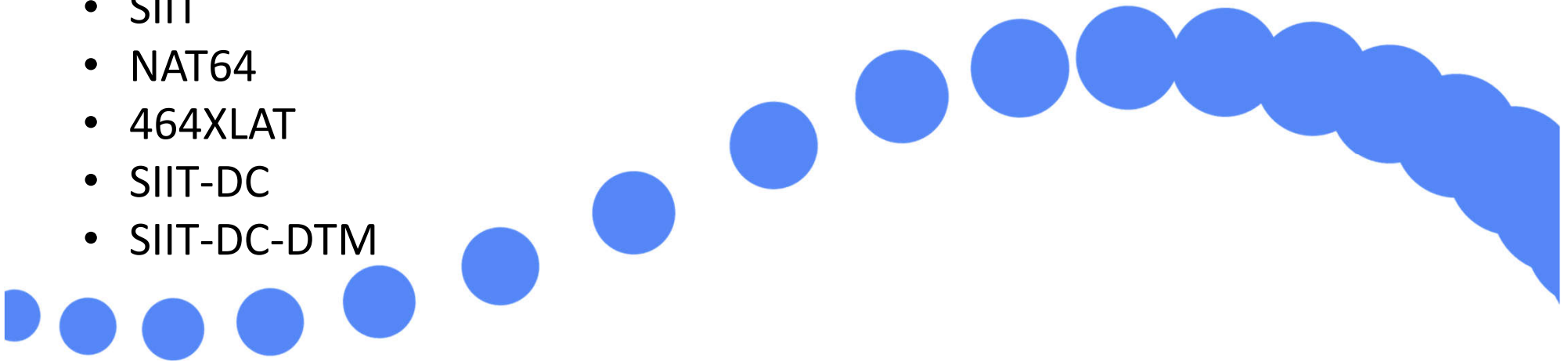
Mecanismos de Transición

Operando ambas versiones

- Dual Stack

Traducción

- SIIT
- NAT64
- 464XLAT
- SIIT-DC
- SIIT-DC-DTM



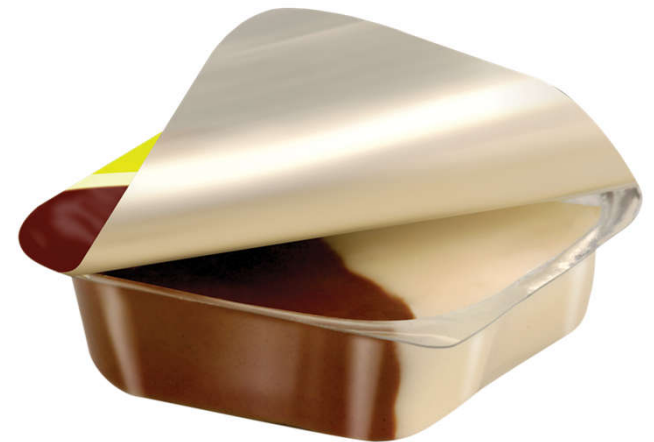
Dual Stack

Ventajas

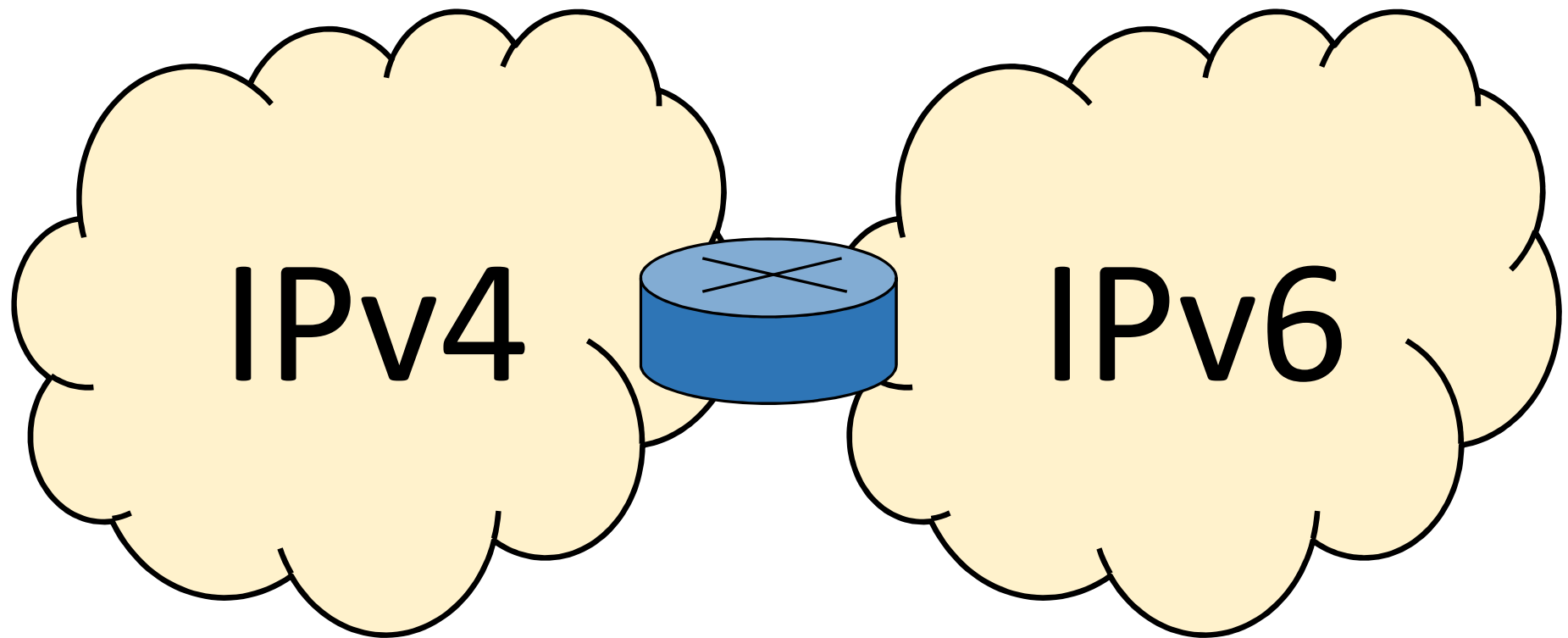
- Soporte para aplicaciones y servicios legados que solo soportan IPv4

Desventajas

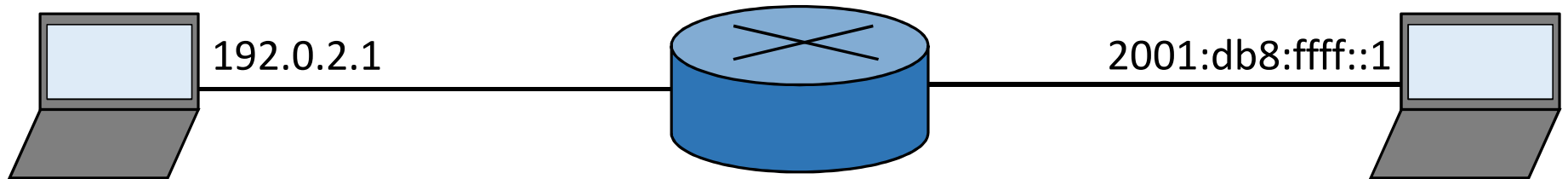
- Agotamiento de direcciones IPv4
- Carrier-grade NAT (NAT444) es costoso
- Doble mantenimiento



Traducción de direcciones IP



Traducción de direcciones IP



MAC	01:01:01:01:01:01
	02:02:02:02:02:02
IP	192.0.2.1
	10.0.0.4
PORT	6123
	80
DATA	Hola

	03:03:03:03:03:03
	04:04:04:04:04:04
	2001:db8:aaaa::1
	2001:db8:ffff::1
	6123
	80
	Hola

Traducción de direcciones IP

IPv4

Version, IHL	TOS	Total Length
Identification		Flags, fragment offset
TTL	Protocol	Header Checksum
Source Address		
Destination Address		
Payload		

IPv6

Version	Traffic class	Flow label	
Payload length		Next header	Hop limit
Source Address			
Destination Address			
Payload			

Traducción de direcciones IP

IPv4

Version, IHL	TOS	Total Length
Identification		Flags, fragment offset
TTL	Protocol	Header Checksum
Source Address		
Destination Address		
Payload		

IPv6

Version	Traffic class	Flow label		
Payload length		Next header	Hop limit	
Source Address				
Destination Address				
Next Header	Reserved	Frag Offset	Res	M
Identification				
Payload				

Traducción IP/ICMP

Traducción de headers:

- RFC 7915

Traducción de direcciones:

- RFC 6052 (Prefijo simple – tradicional)
- RFC 6146 (Stateful NAT64)
- RFC 6791 (ICMP errors)
- RFC 7757 (Explicit Address Mappings)



Traducción IP

SIIT

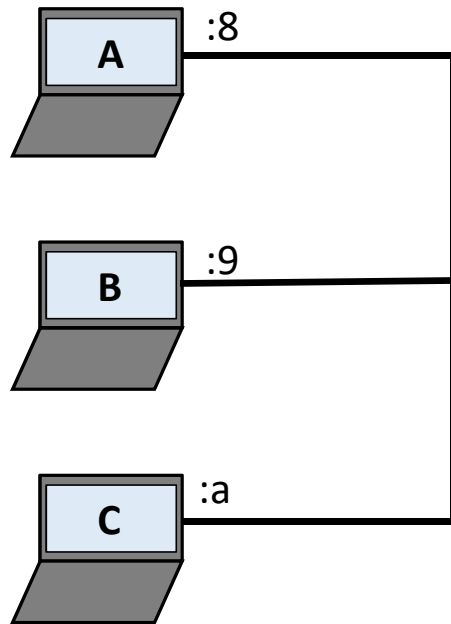
- “Stateless NAT64”
- Traductor de capa 3

“NAT64”

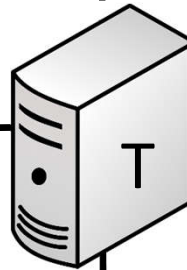
- Stateful NAT64
- Traductor de capas 3 y capas 4

SIIT-EAM

2001:db8:6::/96

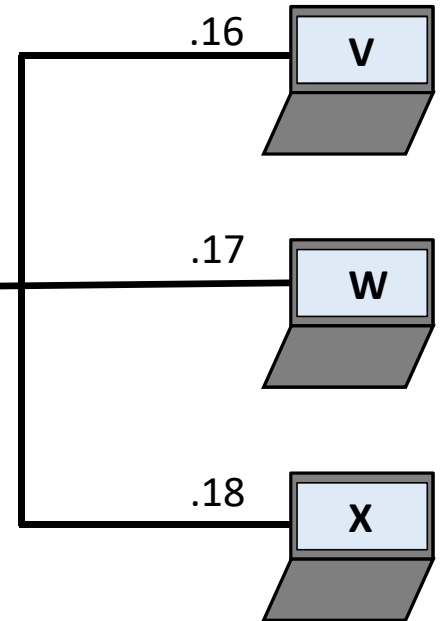


:1



.1

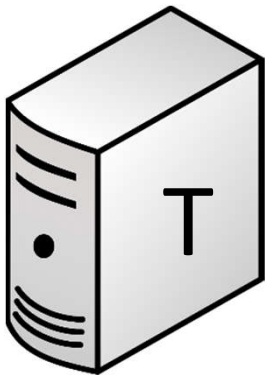
192.0.2.0/24



SIIT-EAM

Voy a engañar a los nodos IPv4.
Ellos van a creer que los nodos
2001:db8:6::/120 se llaman 10.0.0.0/24.

También voy a engañar a los nodos de IPv6.
Ellos van a creer que los nodos 192.0.2.0/24
se llaman 2001:db8:4::/120.



2001:db8:6::/120  **10.0.0.0/24**

- /120 y /24 indican **prefijo**.
- 10.0.0.0/24  **10.0.0.0**
- 2001:db8:6::/120  **2001:0db8:0006:0000:0000:0000:0000:0000**
- La idea de EAM es reemplazar prefijos.
- Por ejemplo, la dirección **10.0.0.5**:
 - Quitamos el prefijo de IPv4: .5
 - Agregamos el prefijo de IPv6: **2001:db8:6::5**
 - Done

2001:db8:6::/120  **10.0.0.0/24**

2001:db8:6::1

10.0.0.1

2001:db8:6::2

10.0.0.2

2001:db8:6::3

10.0.0.3

...

2001:db8:6::9

10.0.0.9

2001:db8:6::a

10.0.0.10

2001:db8:6::b

10.0.0.11

...

2001:db8:6::ef

10.0.0.254

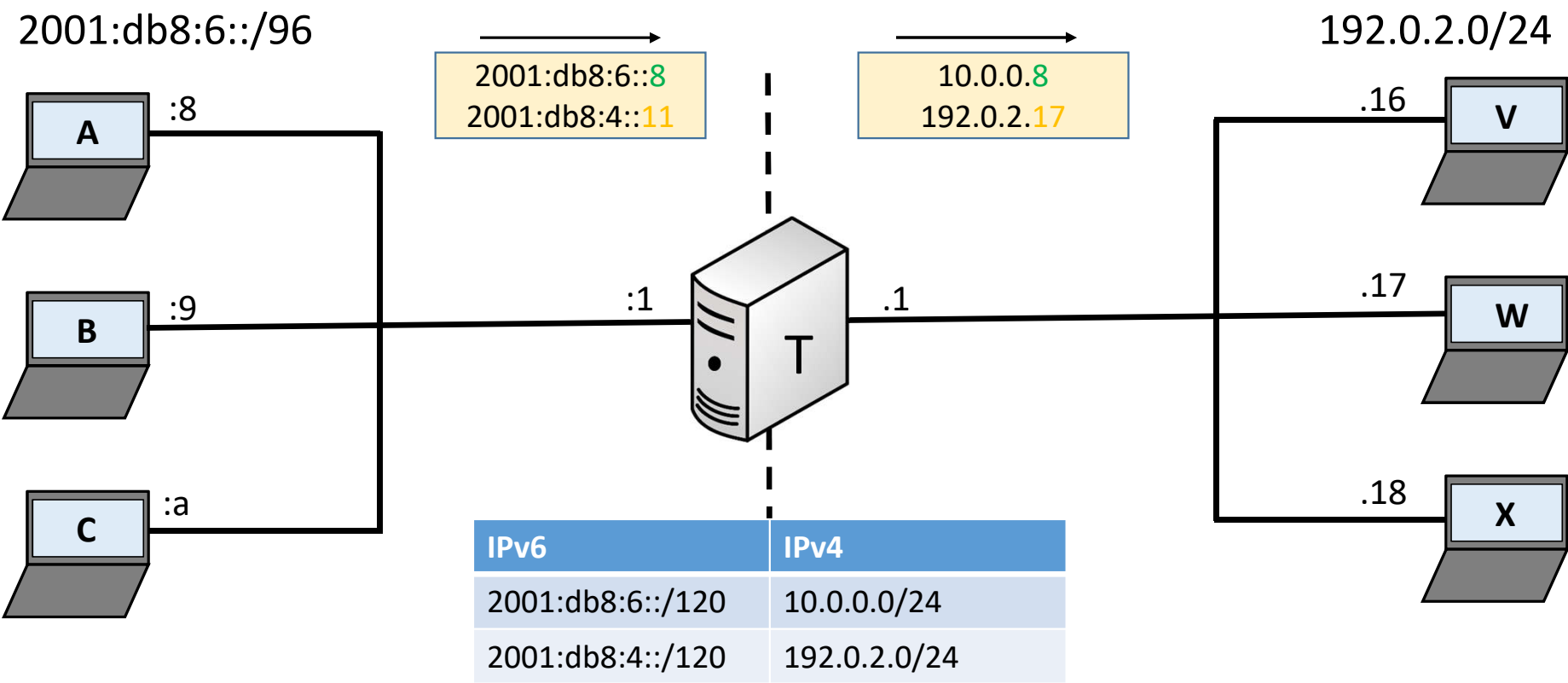
2001:db8:6::ff

10.0.0.255

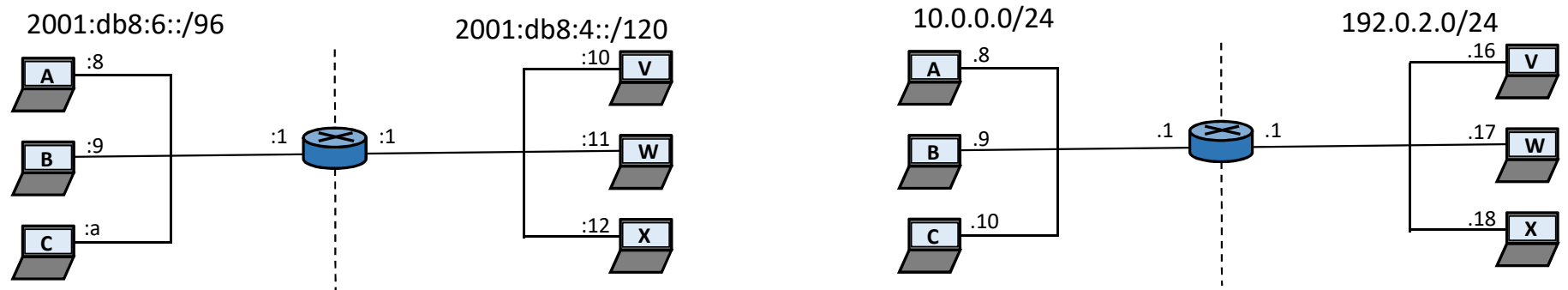
Tabla EAM

IPv6	IPv4
2001:db8:6::/120	10.0.0.0/24
2001:db8:4::/120	192.0.2.0/24

SIIT-EAM



SIIT-EAM



IPv6	IPv4
2001:db8:6::/120	10.0.0.0/24
2001:db8:4::/120	192.0.2.0/24

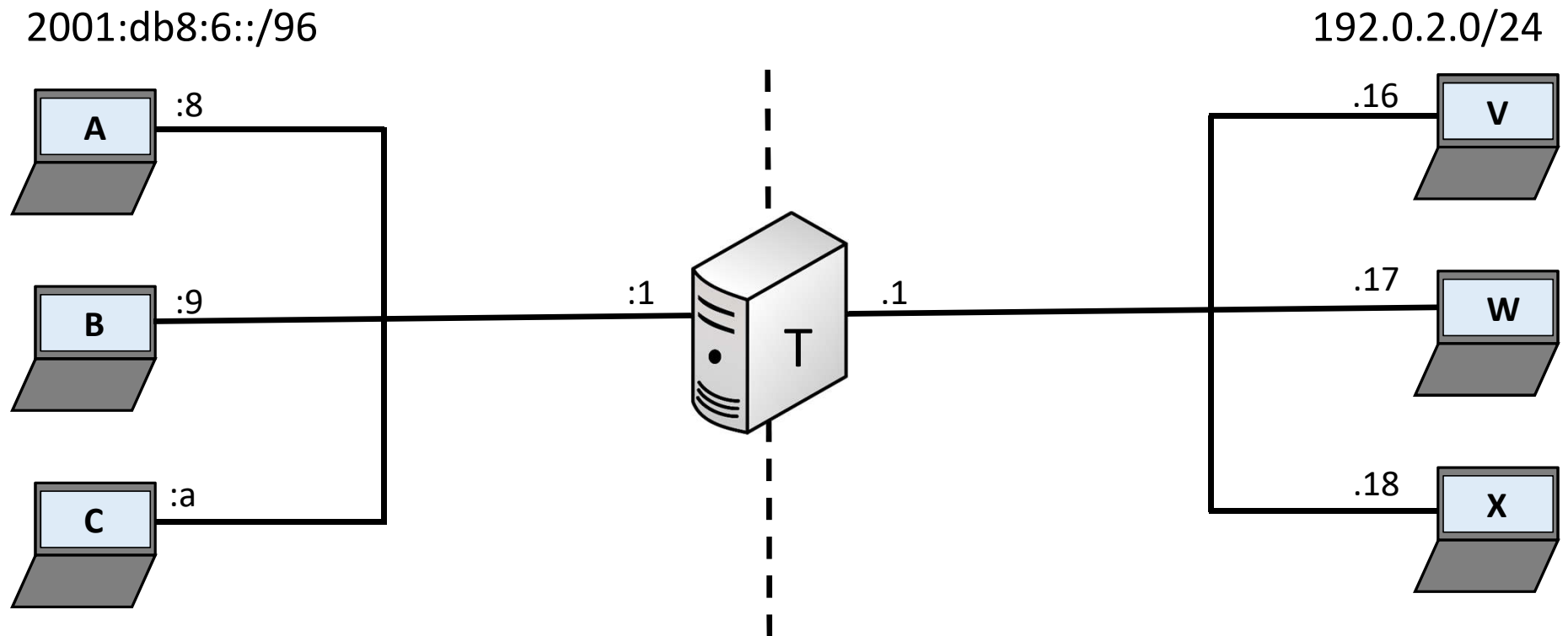
Tabla EAM (simplificada)

IPv6	IPv4
2001:db8:6::1/128	10.0.0.6/32
2001:db8:4::d5/128	192.0.2.10/32
2001:db8:aaaa:bbbb::cccc/128	198.51.100.55/32
2001:db8:1234:5678::ef/128	10.0.0.5/32

SIIT-tradicional

- SIIT-EAM **reemplaza** prefijos de ambos protocolos (IPv6/4).
 - Se basa en una tabla, por lo que podemos tener cualquier número de prefijos.
- En contraste, SIIT-tradicional solamente **agrega** y **quita un** prefijo de IPv6.
 - Es más restrictivo.

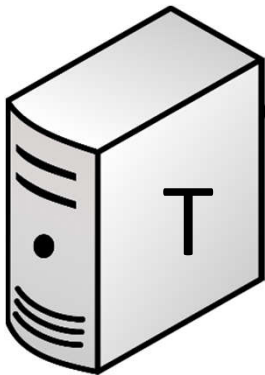
SIIT-tradicional



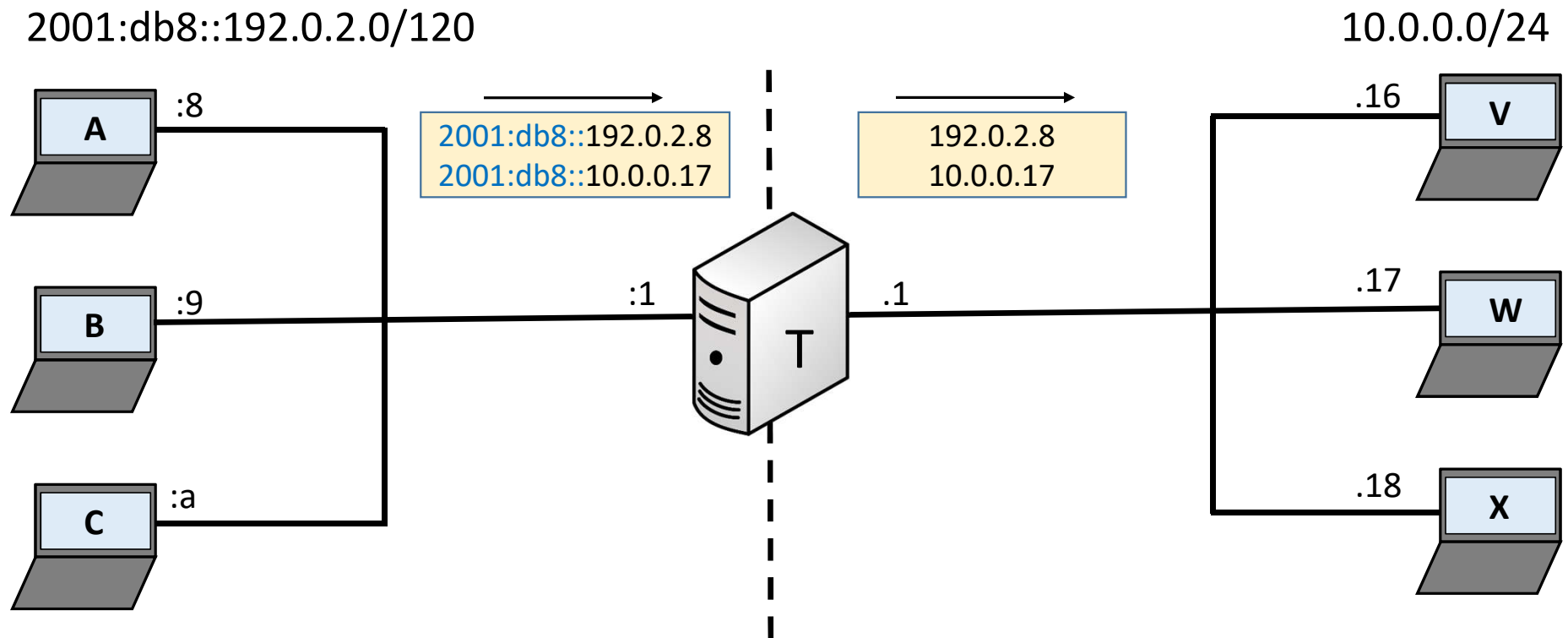
SIIT

Cuando me toque traducir un paquete de 4 a 6
voy a **agregar** el prefijo 2001:db8::/96.

Cuando me toque traducir de 6 a 4, voy a
quitar el prefijo 2001:db8::/96.

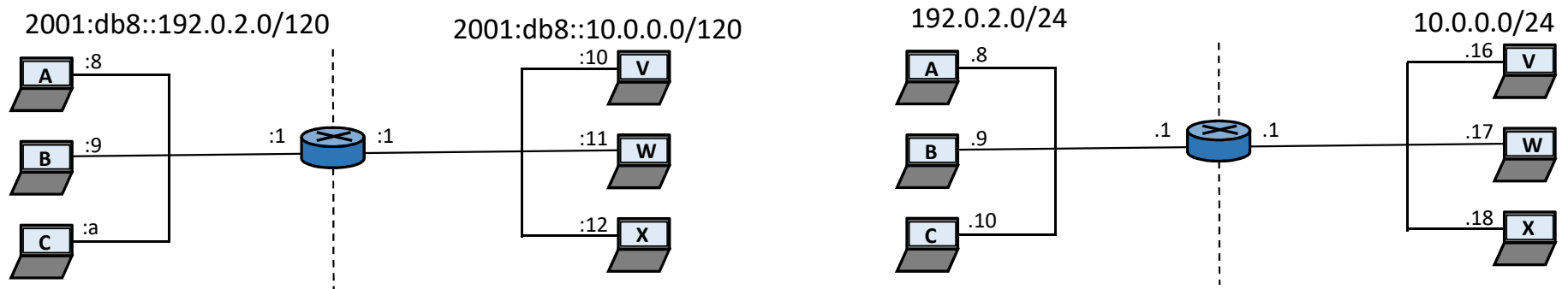


SIIT-tradicional



- Toda la dirección IPv4 sobrevive.
- Todos los nodos de IPv6 que quieran hablar con IPv4 necesitan el prefijo y una dirección IPv4 válida dentro.

SIIT-tradicional



Jool

- SIIT y NAT64
- Linux (kernels 3.2 en adelante)
- Libre uso y de código abierto
- Desarrollado en C

<https://jool.mx>



Instalar Jool

- Módulos de Kernel
- Aplicaciones de configuración y control



Instalar Jool

Ejemplo: Instalar Jool en Debian 10

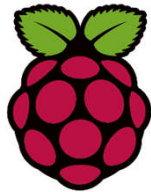
Documentación completa y ejemplos

<https://jool.mx>



debian

OpenWrt
Wireless Freedom



CentOS



alpine
Linux



ubuntu

Instalar Jool

1. Revisar versión del kernel

```
$ /bin/uname -r  
4.19.0-6-amd64
```

2. Build Essentials

```
# apt-get install build-essential pkg-config
```

3. Kernel headers

```
# apt-get install linux-headers-$(uname -r)
```

4. Development Library and Headers para libnl-genl-3

```
# apt-get install libnl-genl-3-dev
```



Instalar Jool

5. Netfilter xtables Library development

```
# apt-get install libxtables-dev
```

6. Descargar Jool

```
$ wget https://github.com/NICMx/Jool/releases/download/v<version>/jool_<version>.tar.gz  
$ tar -xzf jool_<version>.tar.gz
```

7. Instalar Jool

```
# dkms install jool-<version>
```

8. Instalar herramientas de usuario

```
$ cd jool-<version>  
$ ./configure  
$ make  
# make install
```



jcano@debian

```
OS: Debian GNU/Linux 10 (buster) x86_64
```

Host: VirtualBox 1.2

```
Kernel: 4.19.0-6-amd64
```

Uptime: 22 secs

```
Packages: 377 (dpkg)
```

```
Shell: bash 5.0.3
```

```
Terminal: /dev/tty1
```

```
CPU: Intel i7-3520M (1) @ 2.890GHz
```

GPU: VMware SVGA II Adapter

```
Memory: 57MiB / 987MiB
```



```
jcano@debian:~$
```

Taller Jool

Opción 1: Live USB

- Arrancar su computadora personal desde la memoria USB
- Es posible que sea necesario desactivar la validación EUFI en el BIOS

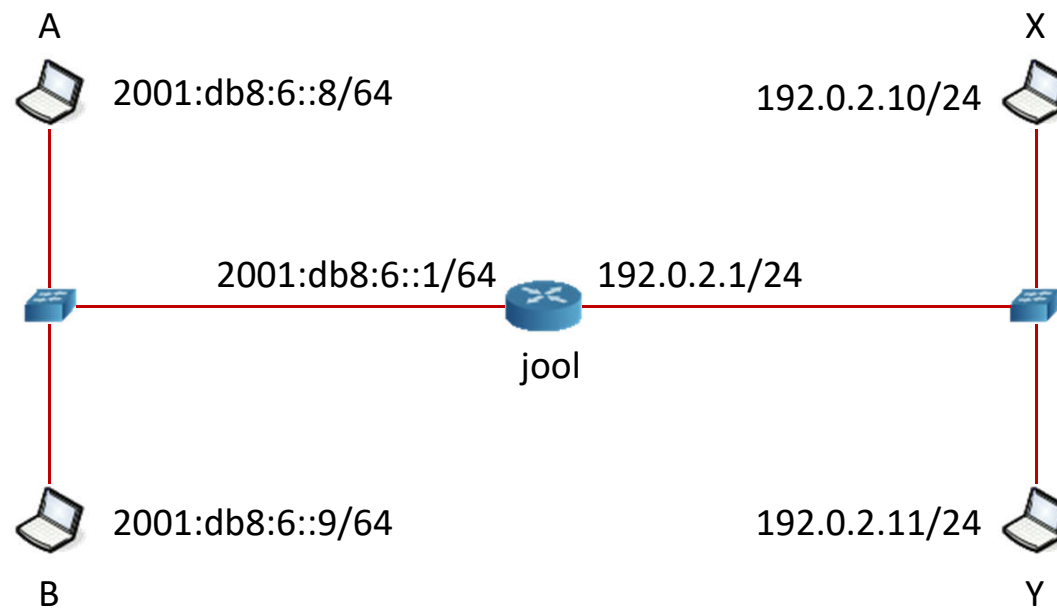
Opción 2: VirtualBox

1. Instalar VirtualBox
2. Importar la máquina virtual del taller de la memoria USB
3. Ejecutar la máquina virtual en VirtualBox





Práctica 1: SIIT



Práctica 1: SIIT

1. Nodo Jool:

```
user@jool:~# jool_siit instance add --netfilter
user@jool:~# jool_siit eamt add 2001:db8:6::/120 198.51.100.0/24
user@jool:~# jool_siit eamt add 2001:db8:4::/120 192.0.2.0/24
user@jool:~# jool_siit eamt display
```

IPv6 Prefix	IPv4 Prefix
2001:db8:4::/120	192.0.2.0/24
2001:db8:6::/120	198.51.100.0/24

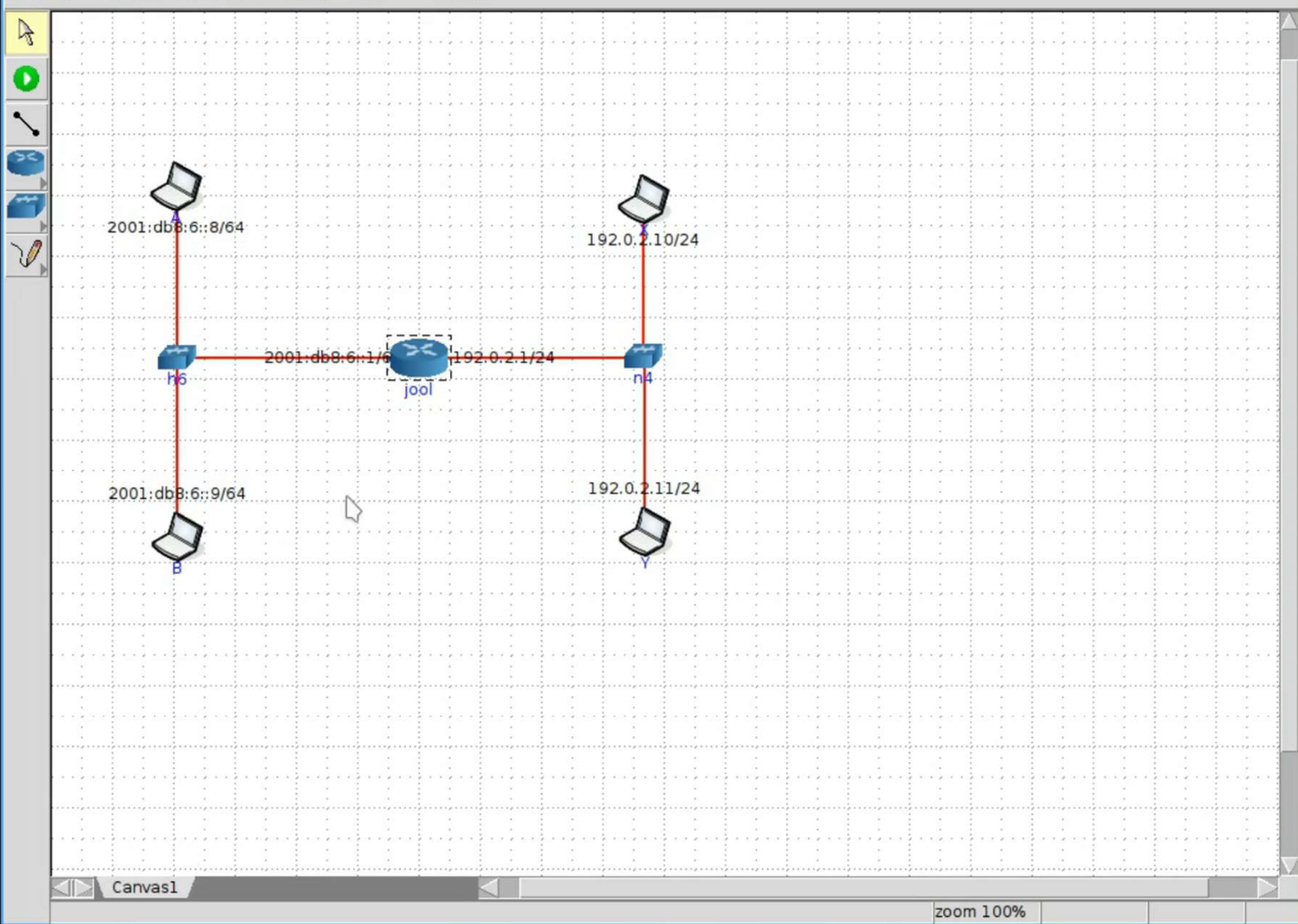
Práctica 1: SIIT

1. Nodo A:

```
user@A:~# ping6 -c 4 2001:db8:6::9  
user@A:~# ping6 -c 4 2001:db8:6::1  
user@A:~# ping6 -c 4 2001:db8:4::A  
user@A:~# ping6 -c 4 2001:db8:4::B
```

2. Nodo X:

```
user@X:~# ping -c 4 192.0.2.11  
user@X:~# ping -c 4 192.0.2.1  
user@X:~# ping -c 4 198.51.100.8  
user@X:~# ping -c 4 198.51.100.9
```



SIIT-tradicional y SIIT-EAM

1. ¿Soluciona el problema de coexistencia?

Sí, permite que dispositivos de IPv4 e IPv6 puedan comunicarse entre sí.

2. ¿Soluciona el problema de agotamiento de IPv4?

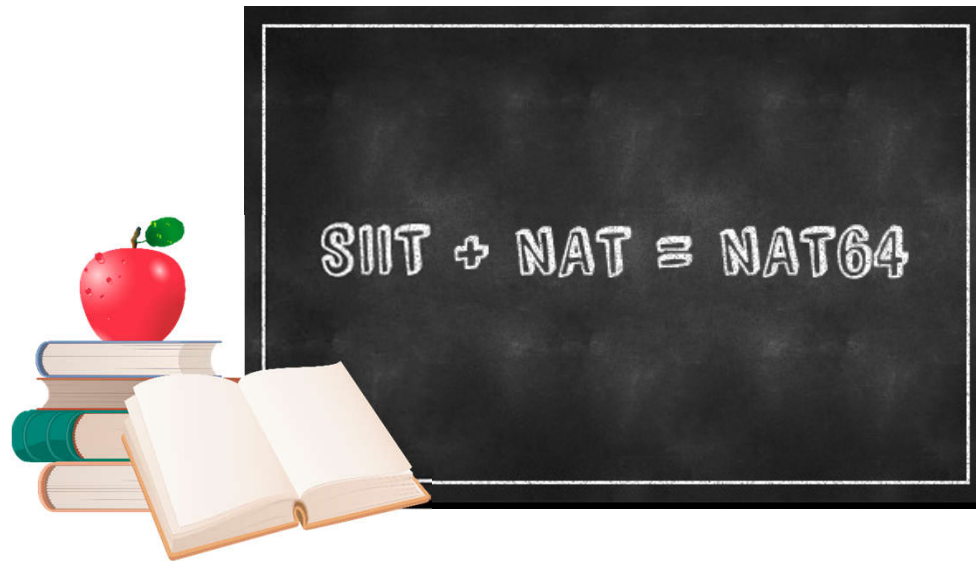
No, porque se necesita una dirección IPv4 por cada IF



© Bethesda Softworks

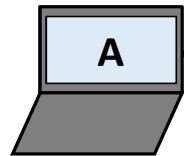
NAT64

NAT64 combina un SIIT y un NAT para lograr que “n” direcciones de IPv6 puedan mapearse a unas cuantas direcciones IPv4.

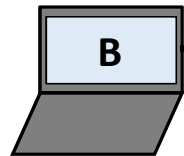


NAT

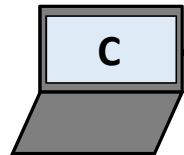
192.168.0.0/24



.8

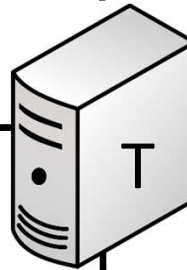


.9



.10

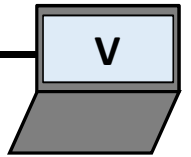
.1



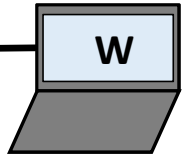
.1

10.0.0.0/24

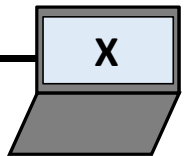
.16



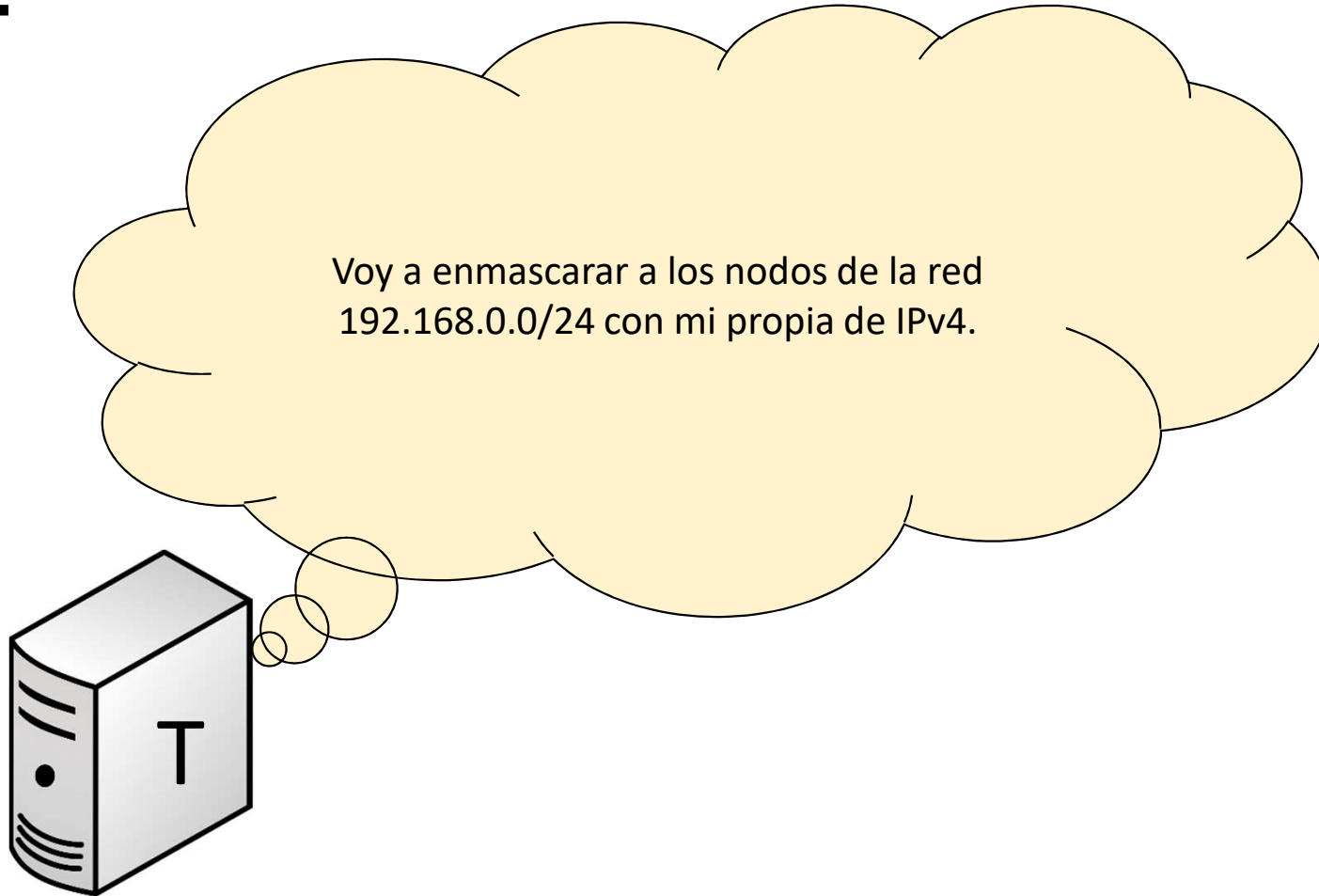
.17



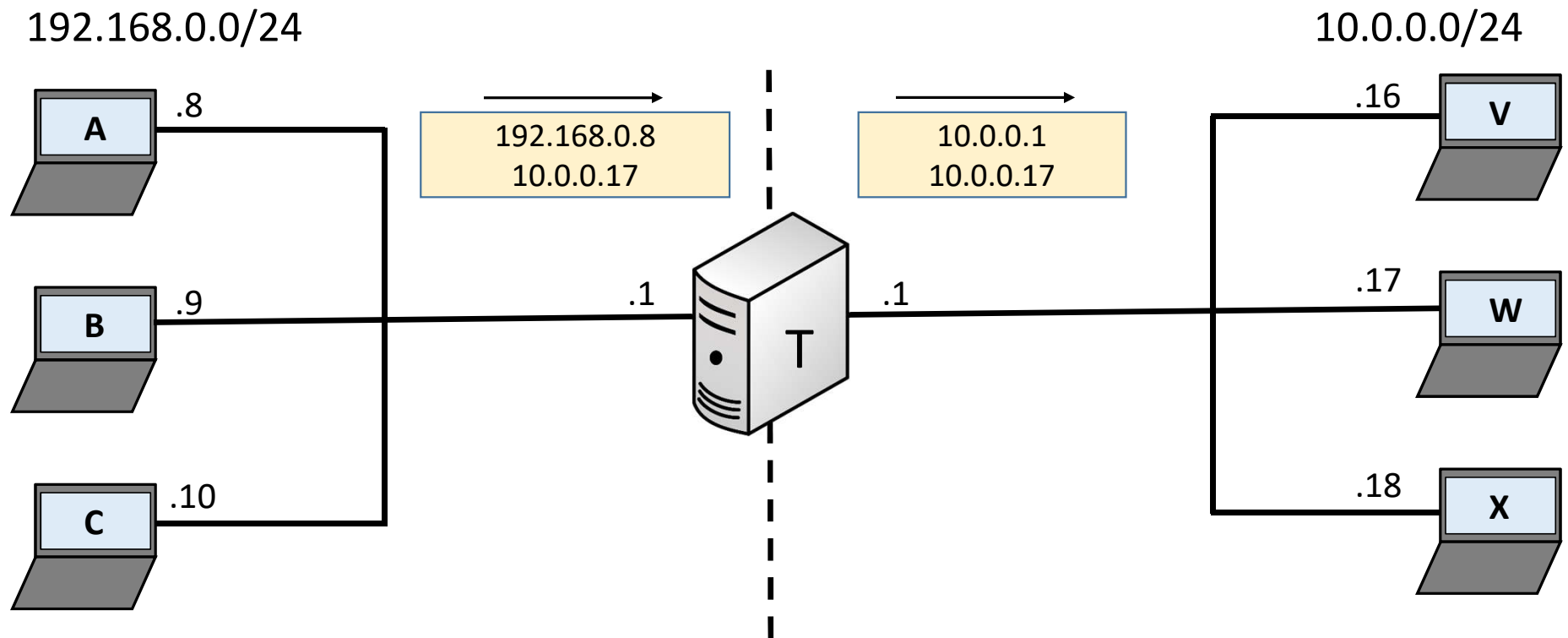
.18



NAT

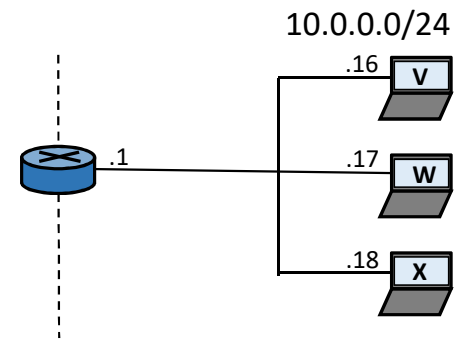
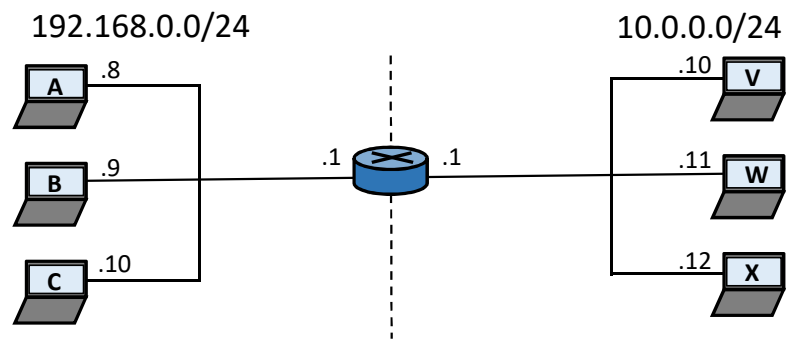


NAT

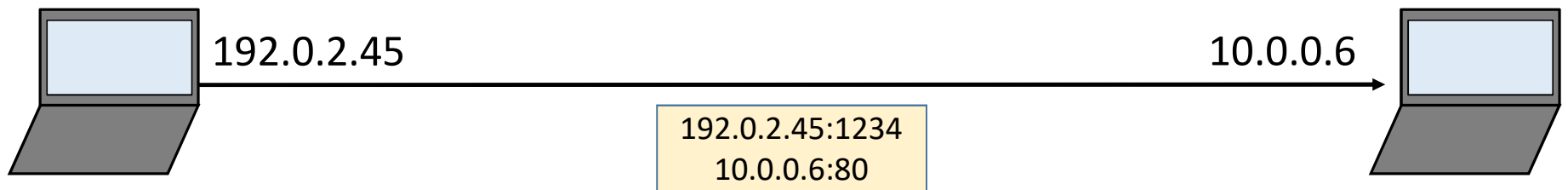


Soluciona el problema de agotamiento de direcciones porque hace que A, B, C y T tengan la misma dirección a los ojos de cualquier otra red.

NAT

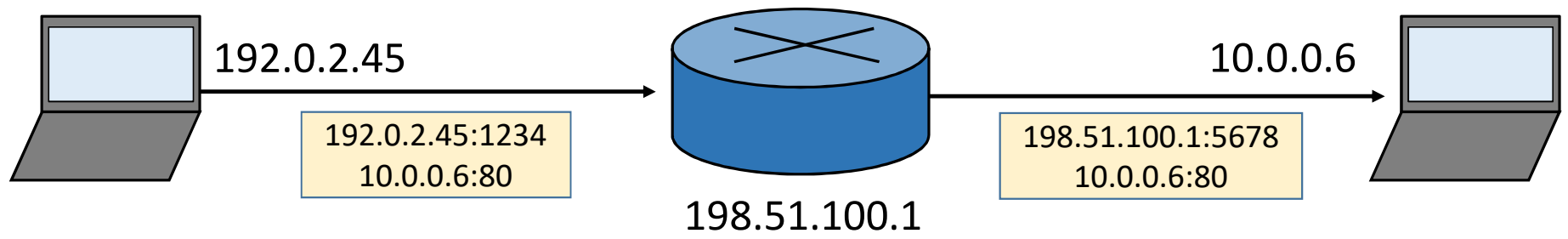


Sockets



Socket	Cliente	Servidor
Dirección	192.0.2.45	10.0.0.6
Puerto	1234	80

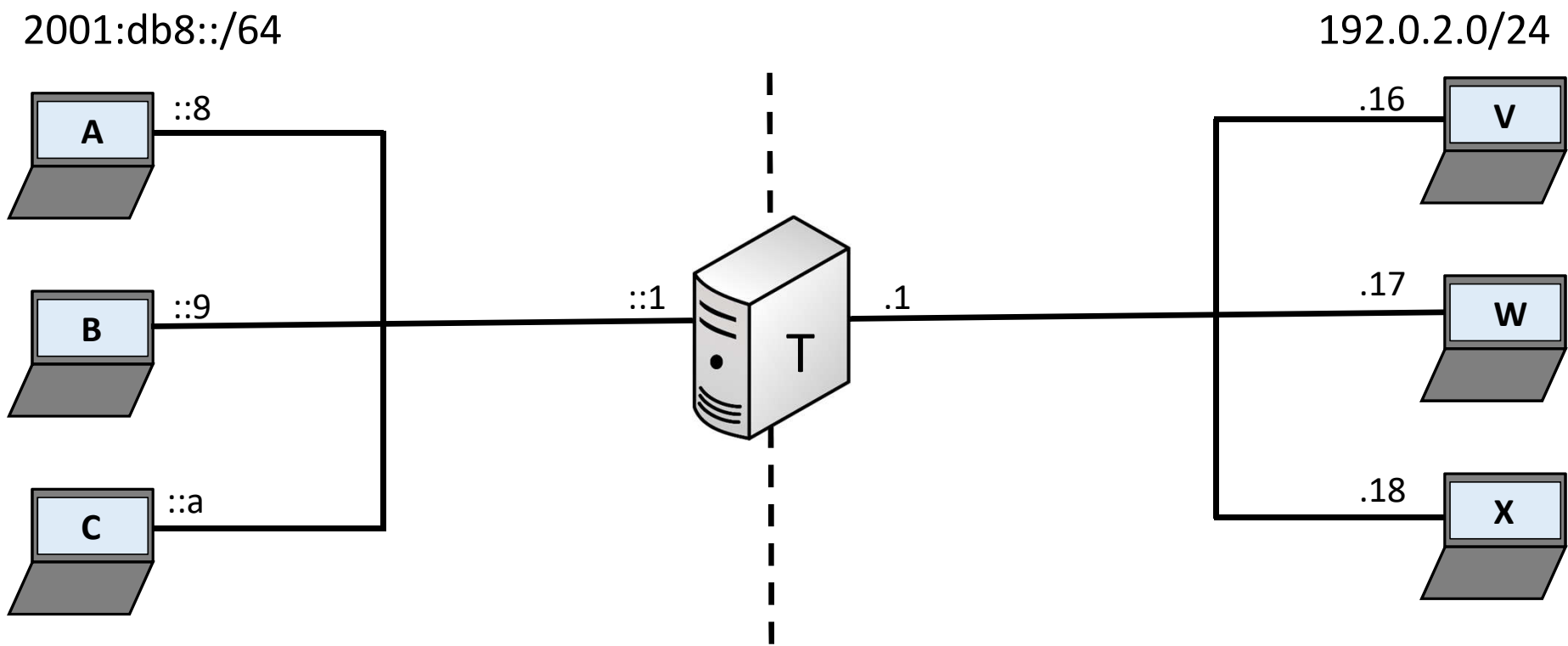
Sockets, NAT



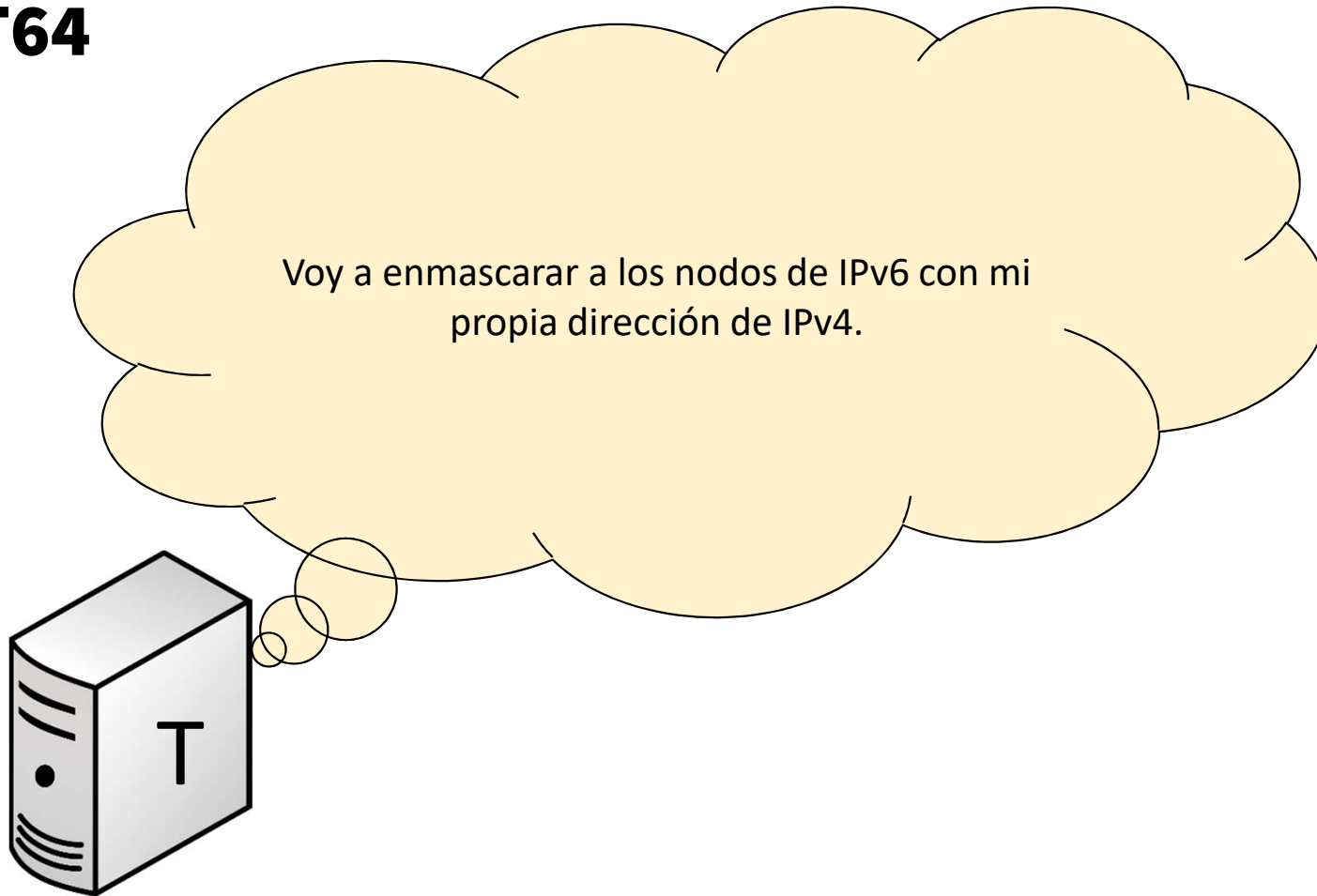
Socket	Cliente	Servidor
Dirección	192.0.2.45	10.0.0.6
Puerto	1234	80

Socket	Cliente	Servidor
Dirección	198.51.100.1	10.0.0.6
Puerto	5678	80

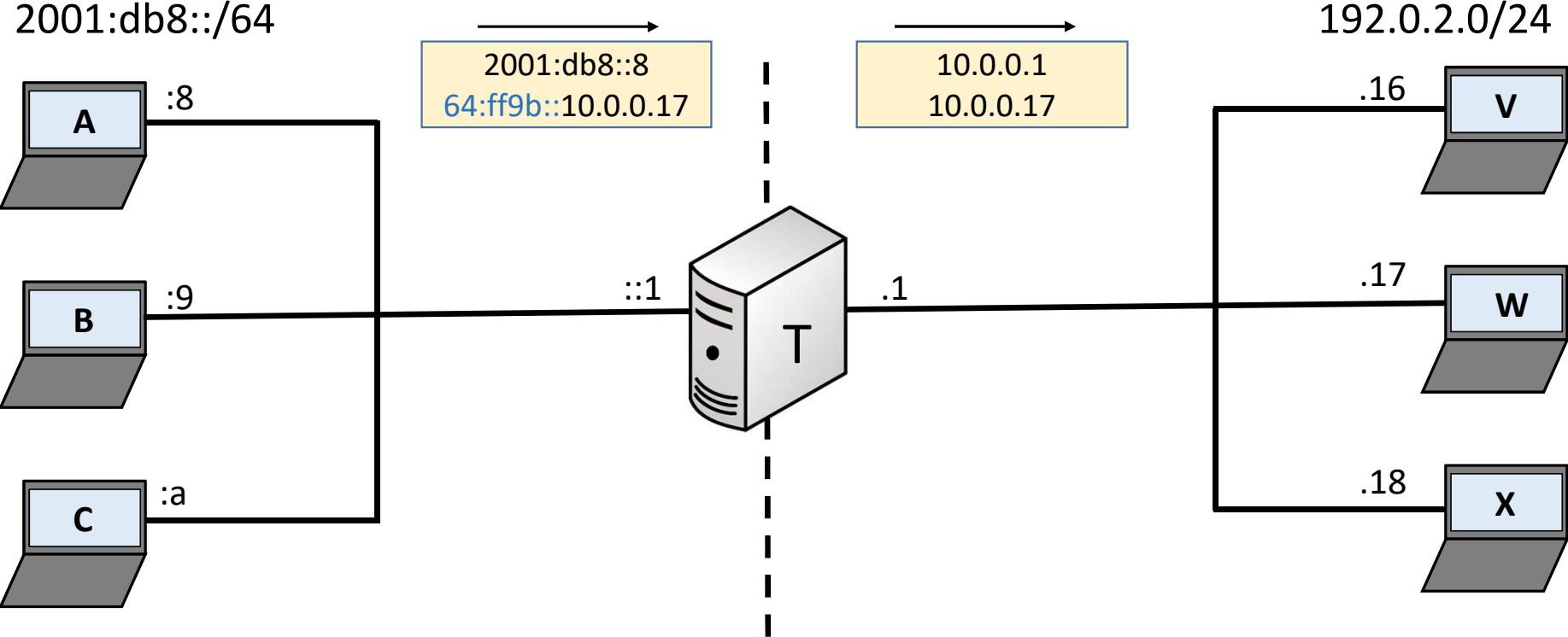
NAT64



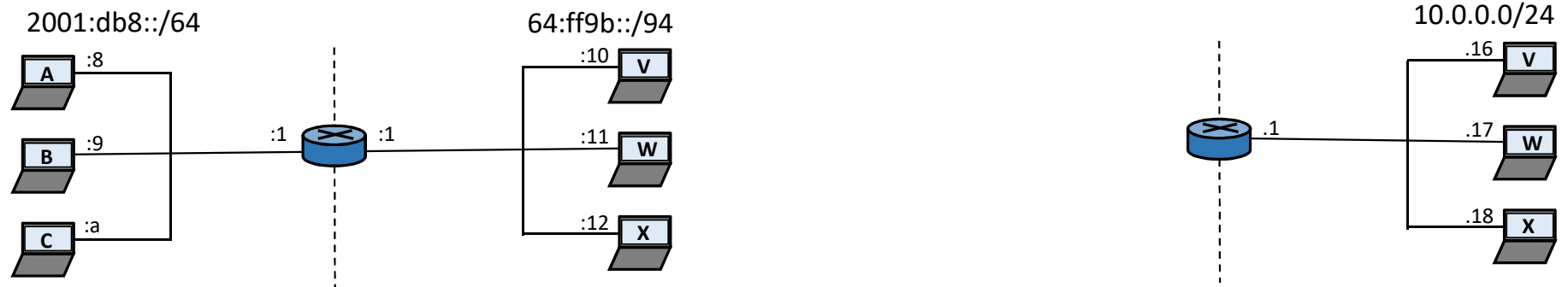
NAT64



NAT64



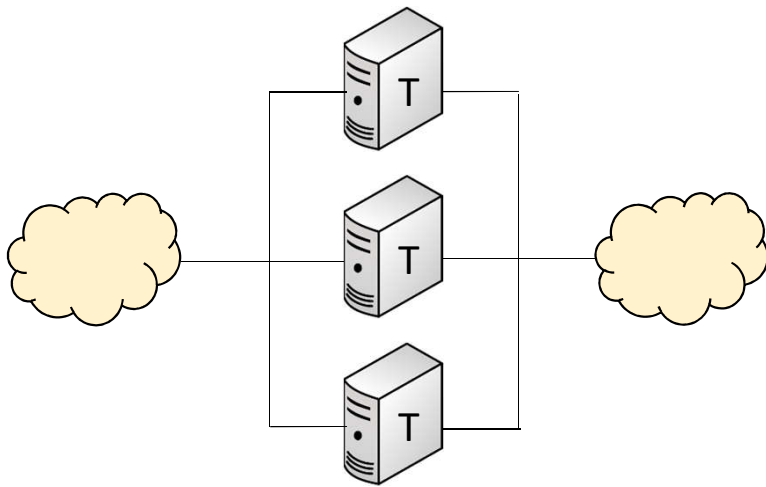
NAT64



SIIT vs NAT64

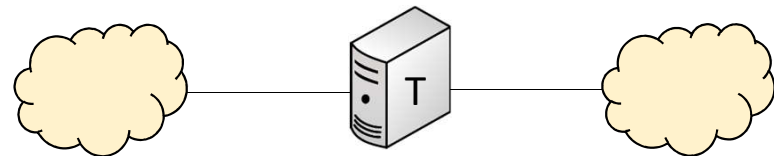
SIIT

- Stateless



NAT64

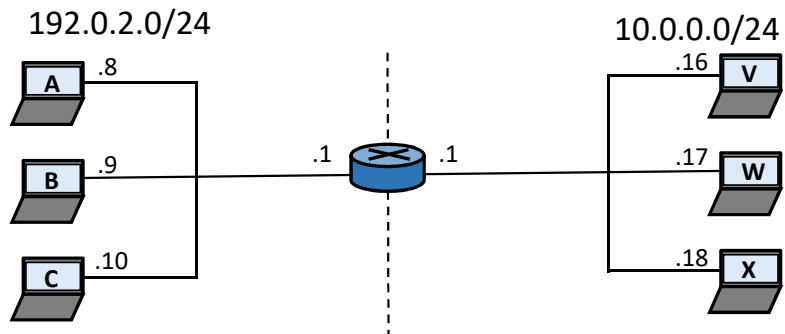
- Stateful



SIIT vs NAT64

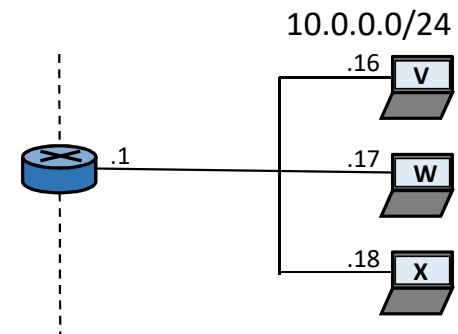
SIIT

- Transparente



NAT64

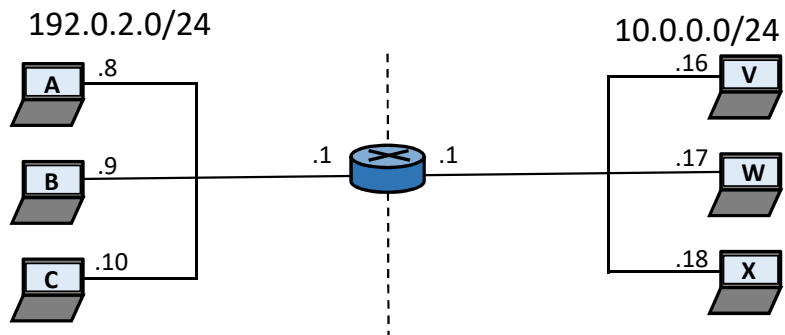
- No transparente



SIIT vs NAT64

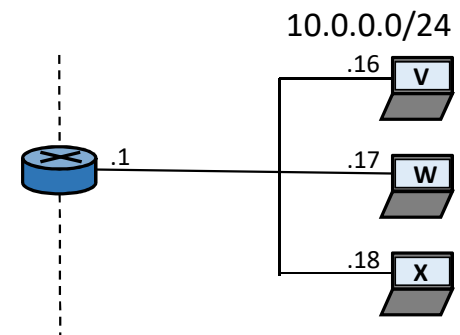
SIIT

- 1 a 1



NAT64

- 1 a N



SIIT vs NAT64

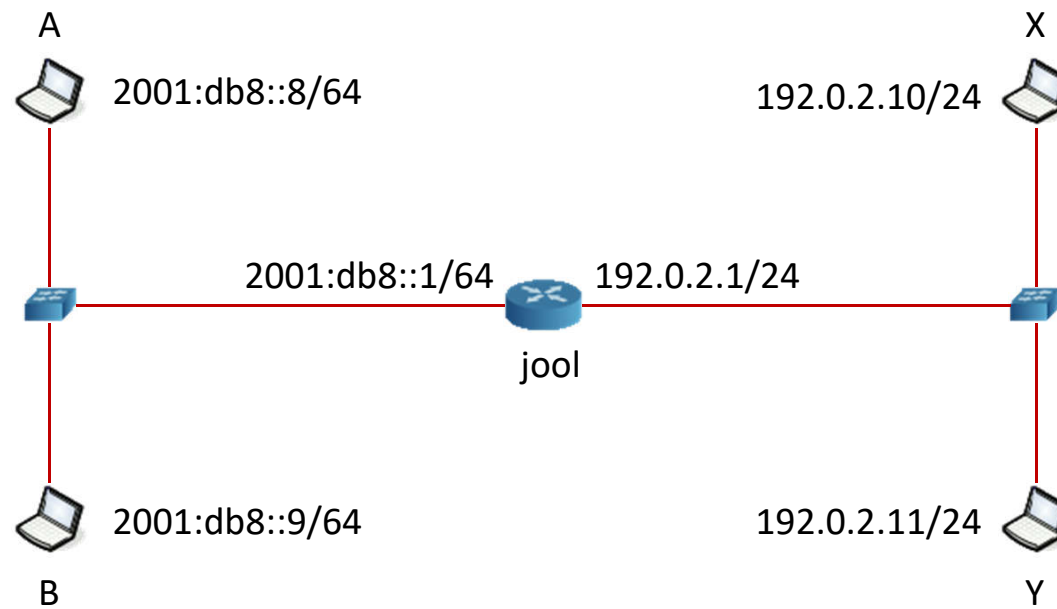
SIIT

- TCP, UDP y otros
- Capa 3 solamente
- La comunicación puede
Iniciar desde v4 y v6

NAT64

- TCP y UDP
- Capa 3 y 4
- La comunicación inicia
normalmente desde v6

Práctica 2: NAT64



Práctica 2: NAT64

1. Nodo Jool:

```
user@jool:~# jool instance add --netfilter --pool6 64:ff9b::/96
```

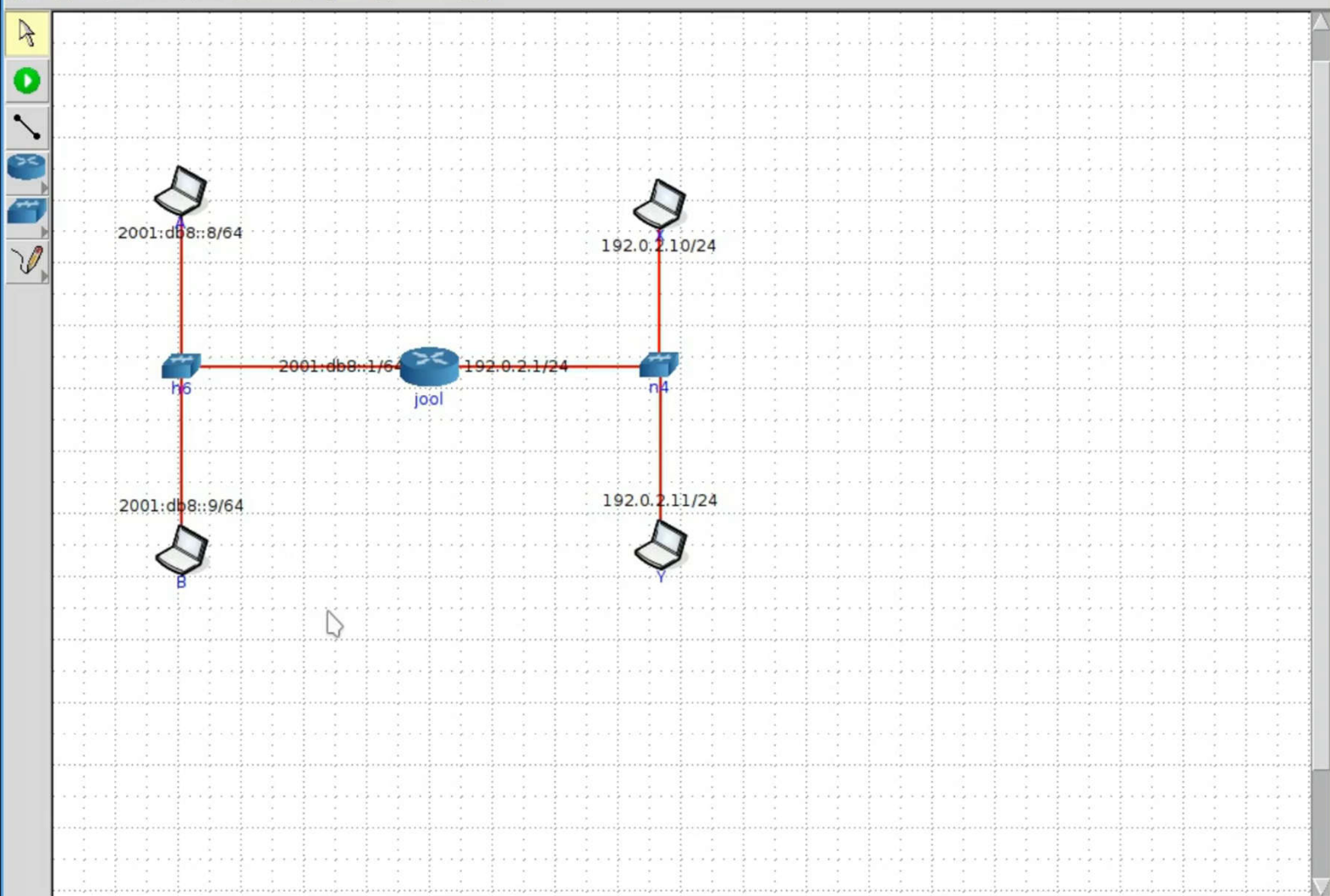
2. Nodo A:

```
user@X:~# ping6 -c 4 2001:db8::9
```

```
user@X:~# ping6 -c 4 2001:db8::1
```

```
user@X:~# ping6 -c 4 64:ff9b::192.0.2.10
```

```
user@X:~# ping6 -c 4 64:ff9b::192.0.2.11
```



Canvas1

Cleanup completed in 0 seconds.

zoom 100%

NAT64

1. ¿Soluciona el problema de coexistencia?

Sí, permite que dispositivos de IPv4 e IPv6 puedan comunicarse entre sí.

2. ¿Soluciona el problema de agotamiento de IPv4?

Sí, porque una o varias direcciones IPv4 son compartidas por múltiples direcciones IPv6.

3. ¿Entonces NAT64 es la solución optima?

No, porque no le puedes pedir a los usuarios que agreguen un prefijo cada que quieran tener acceso a IPv4



© Bethesda Softworks

Nombres de Dominio

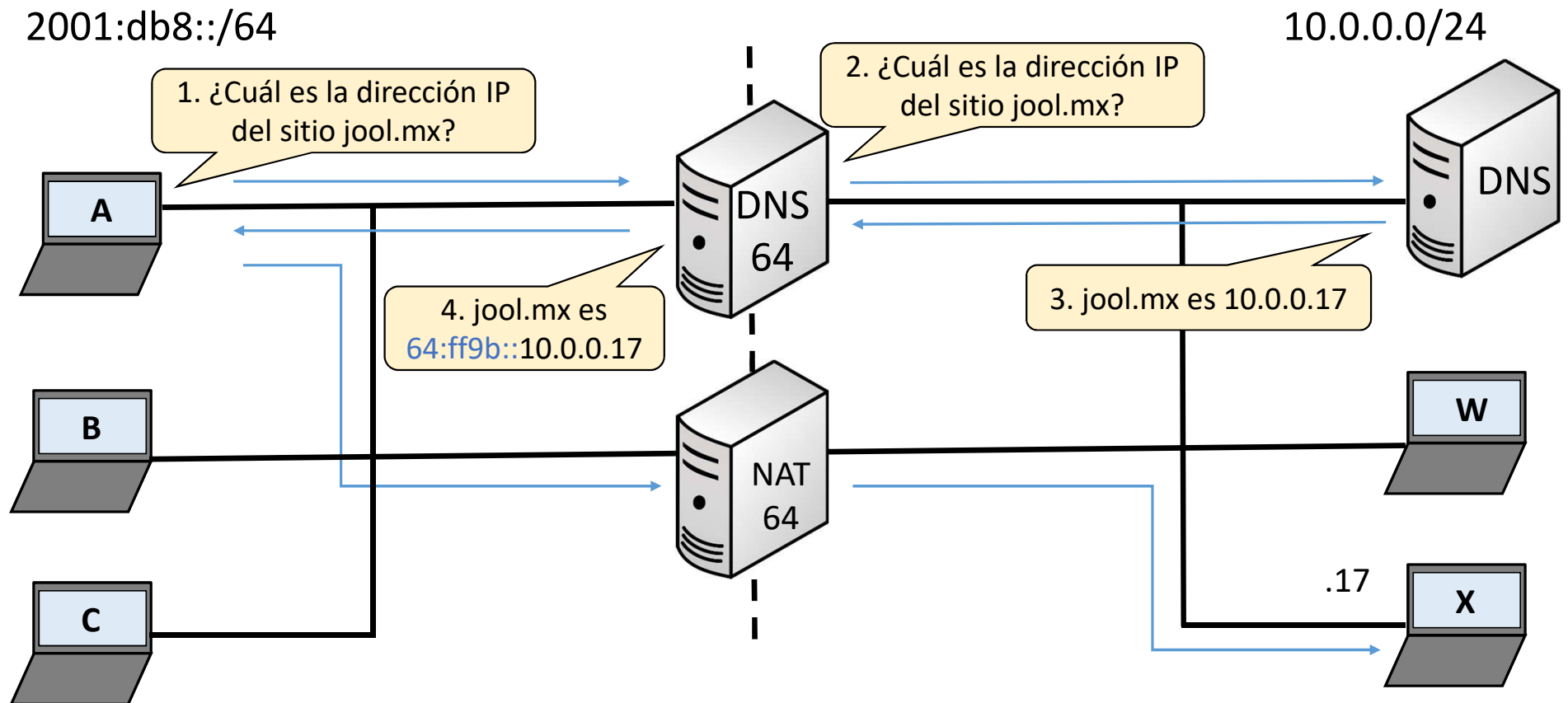
- Son un identificador en Internet.
- Son más fáciles de recordar.
- Son más descriptivos de la empresa o servicio ofrecido.
- Las aplicaciones desarrolladas con nombres de dominio son independientes a la infraestructura.



DNS



DNS64



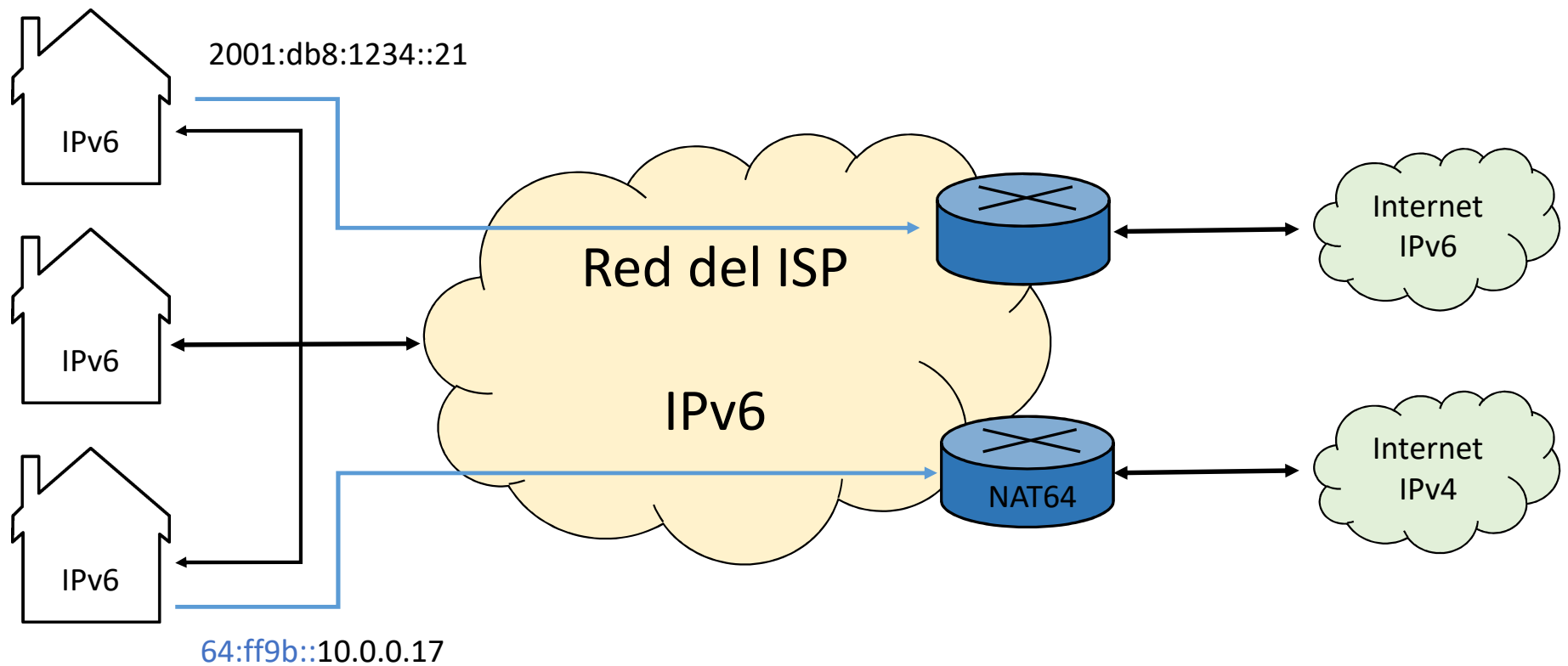
DNS64 (BIND)

```
options {  
    (...)  
  
    # BIND no escucha en IPv6 por defecto.  
    listen-on-v6 { any; };  
  
    # Aquí se le indica a BIND el prefijo que  
    # el NAT64 está agregando y quitando.  
    dns64 64:ff9b::/96 {  
        # Opciones  
    };  
};
```

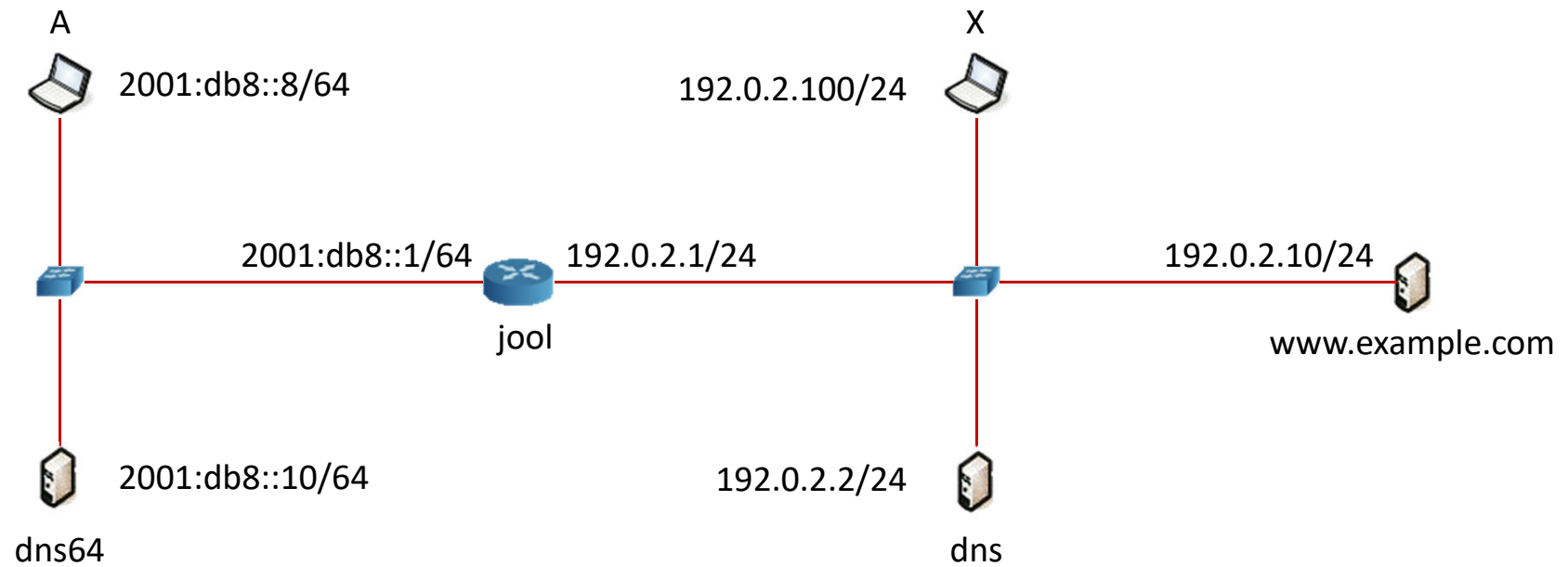
DNS64 (BIND)

```
options {  
    (...)  
  
    # BIND no escucha en IPv6 por defecto.  
    listen-on-v6 { any; };  
  
    # Aquí se le indica a BIND el prefijo que  
    # el NAT64 está agregando y quitando.  
    dns64 64:ff9b::/96 {  
        clients { any; };  
        mapped { any; };  
        exclude { 0::/3; 4000::/2; 8000::/1; 2001:db8::/32; };  
        break-dnssec no;  
    };  
};
```

NAT64 (vida real)



Práctica 3: NAT64 y DNS64



Práctica 3: NAT64 y DNS64

1. Nodo dns64:

Editar el archivo `/etc/bind/named.conf`

```
options {  
    ...  
    forwarders { 64:ff9b::192.0.2.2; };  
    listen-on-v6 { 2001:db8::10; };  
    dns64 64:ff9b::/96 {};  
};
```

2. Nodo jool:

```
user@jool:~# jool instance add --netfilter --pool6 64:ff9b::/96
```

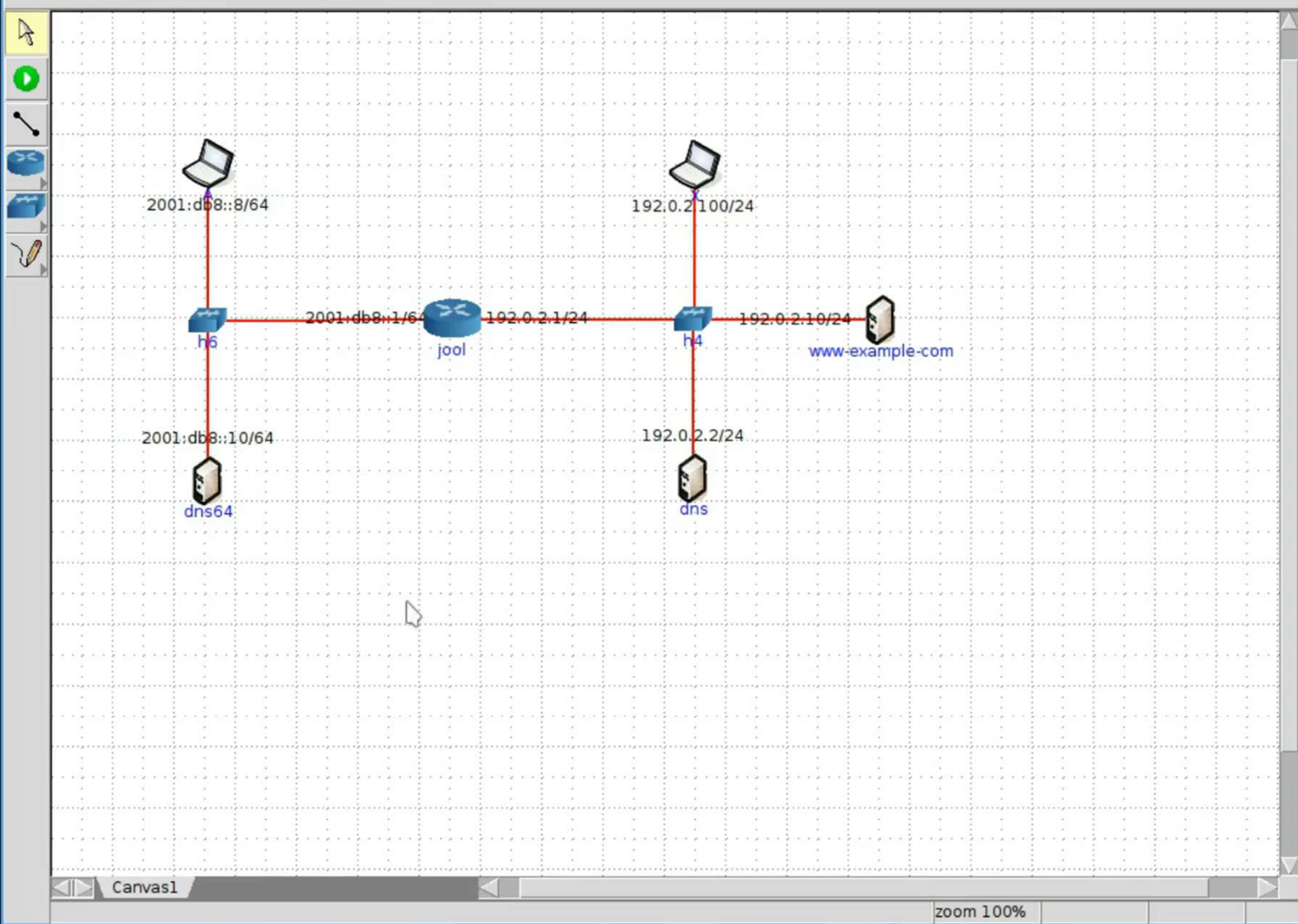
Práctica 3: NAT64 y DNS64

1. Nodo X:

```
user@X:~# ping -c 4 www.example.com  
PING www.example.com (192.0.2.10) ...
```

2. Nodo A:

```
user@A:~# ping6 -c 4 www.example.com  
PING www.example.com (64:ff9b::c000:20a) ...
```



NAT64 / DNS64

1. ¿Soluciona el problema de coexistencia?

Sí, permite que dispositivos de IPv4 e IPv6 puedan comunicarse entre sí.

2. ¿Soluciona el problema de agotamiento de IPv4?

Sí, porque una o varias direcciones IPv4 son compartidas por múltiples direcciones IPv6.

3. ¿Entonces NAT64 / DNS64 es la solución optima?

No, porque algunas aplicaciones o dispositivos legados no soportan IPv6 u operación en IPv6-Only.



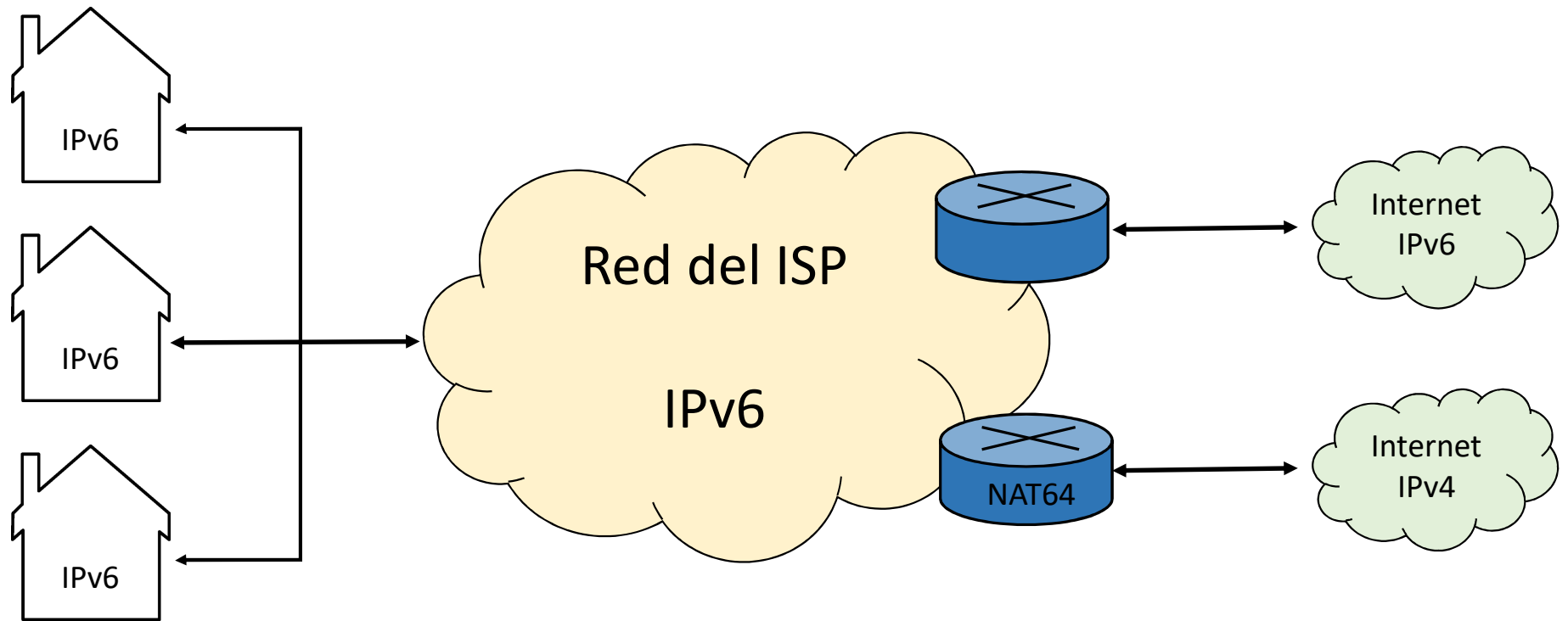
© Bethesda Softworks

Desventajas de NAT64

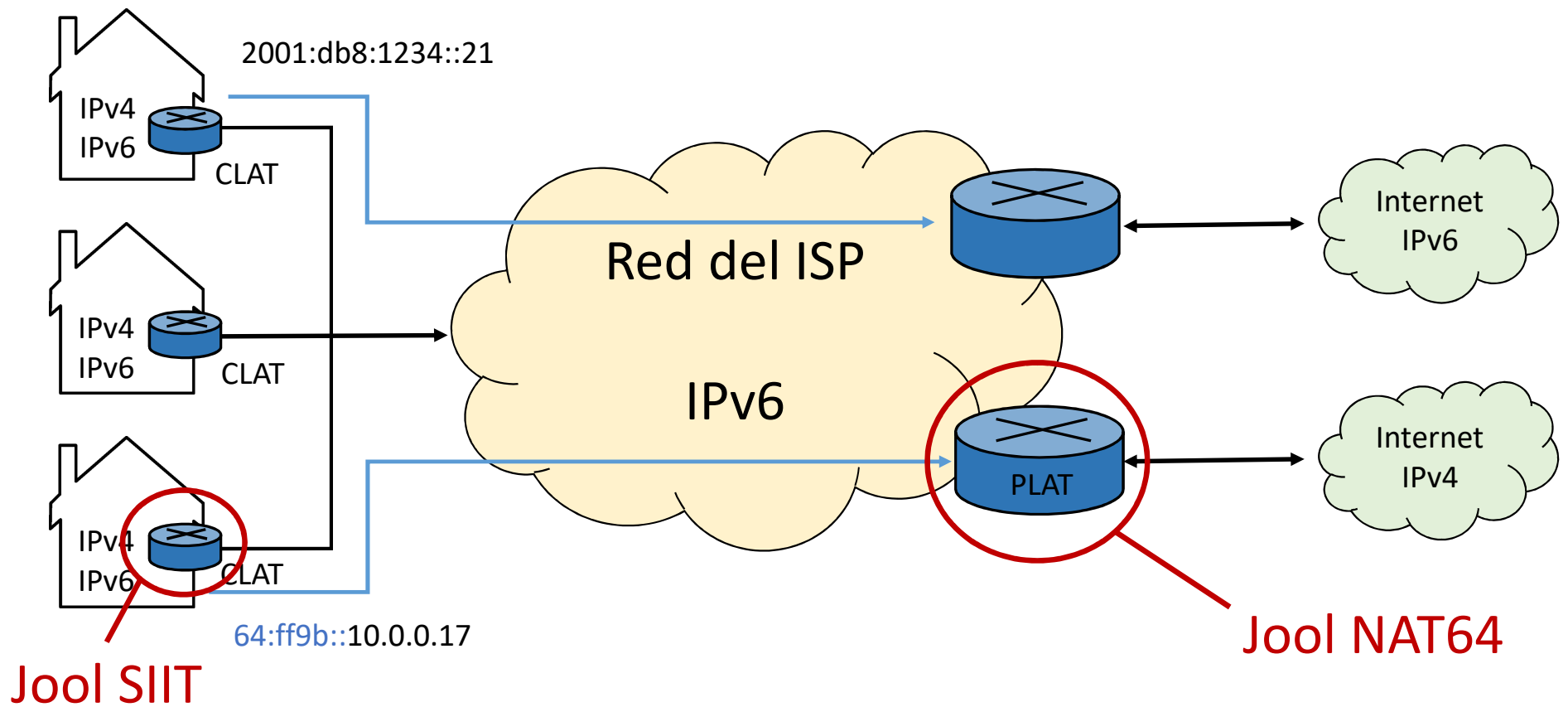
1. Dispositivos antiguos que NO soportan IPv6
2. Aplicaciones legadas que NO soportan IPv6
3. Aplicaciones que utilizan literales
`<a href="http://192.0.2.1/index.html">`
4. DNS64 rompe DNSSEC



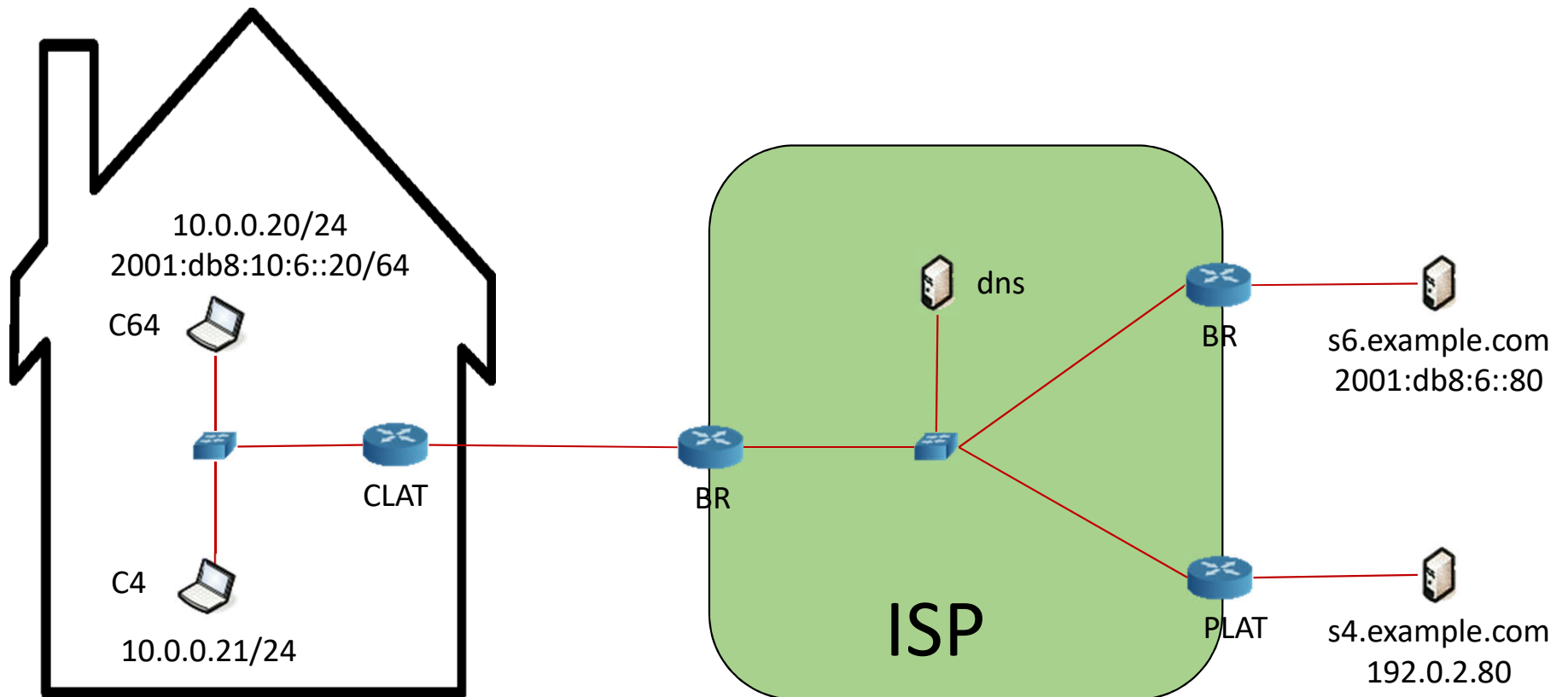
NAT64



464XLAT



Práctica 4: 464XLAT



Práctica 4: 464XLAT

1. Nodo PLAT:

```
# jool instance add --netfilter --pool6 64:ff9b::/96
```

2. Nodo CLAT:

```
# jool_siit instance add --netfilter --pool6 64:ff9b::/96
```

```
# jool_siit eamt add 2001:db8:10:4::/120 10.0.0.0/24
```

Práctica 4: 464XLAT

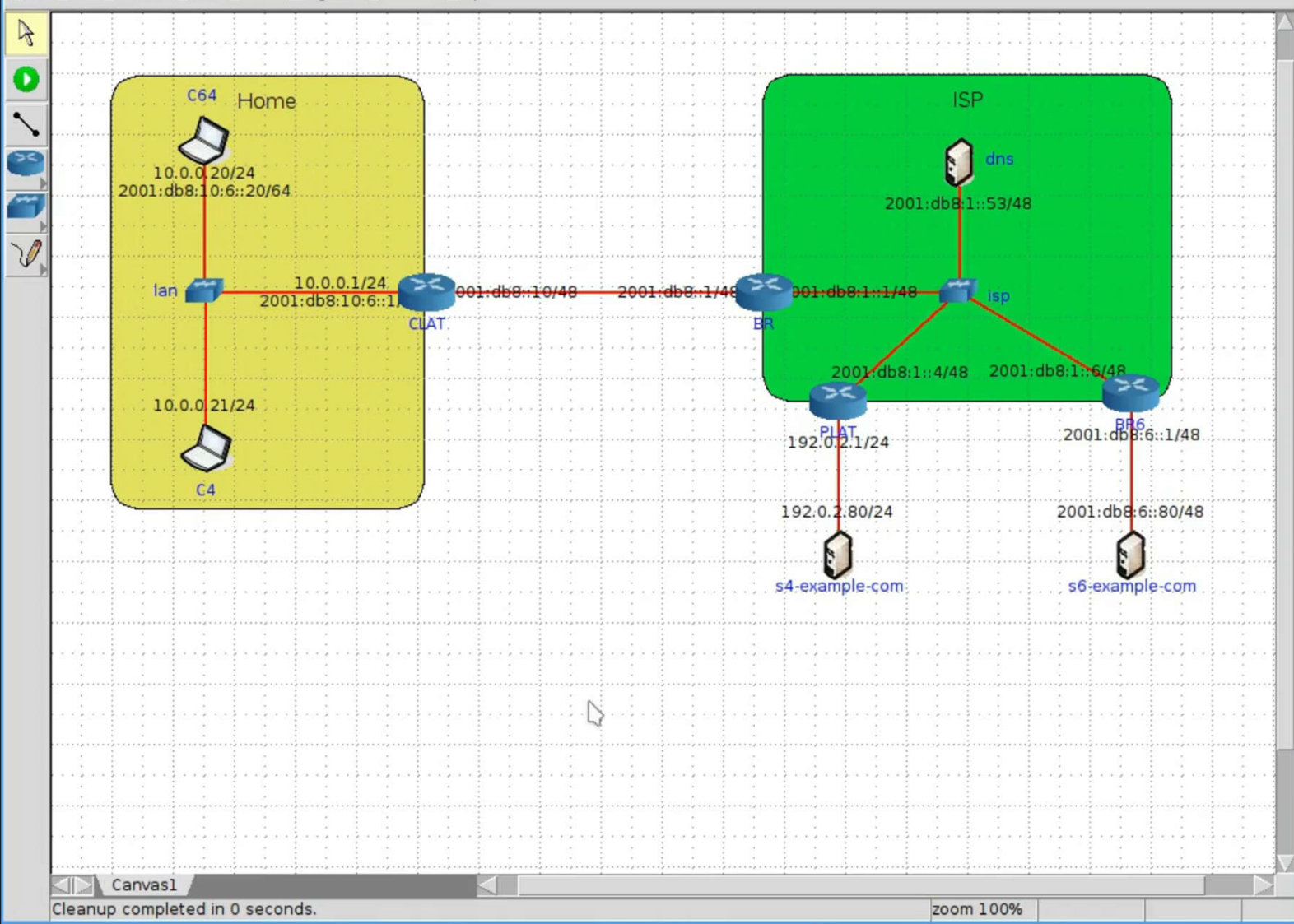
1. Nodo C64:

```
user@C64:~# ping -c 4 s6.example.com  
PING s6.example.com (2001:db8:6::80) ...
```

```
user@C64:~# ping -c 4 s4.example.com  
PING s4.example.com (64:ff9b::c000:250) ...
```

2. Nodo C4:

```
user@C4:~# ping -c 4 s4.example.com  
PING s4.example.com (192.0.2.80) ...
```



464XLAT

1. ¿Soluciona el problema de coexistencia?

Sí, permite que dispositivos de IPv4 e IPv6 puedan comunicarse entre sí.

2. ¿Soluciona el problema de agotamiento de IPv4?

Sí, porque una o varias direcciones IPv4 son compartidas por múltiples direcciones IPv6. Y en el caso de las direcciones IPv4 en el hogar/oficina son IPv4 privadas.

3. ¿Entonces 464XLAT soluciona las desventajas de NAT64?

Sí, porque los dispositivos y aplicaciones legadas reciben una IPv4 y el resto opera en Dual Stack resolviendo también el problema de las literales. Y al ser el DNS64 opcional, se puede omitir para evitar romper DNSSEC.

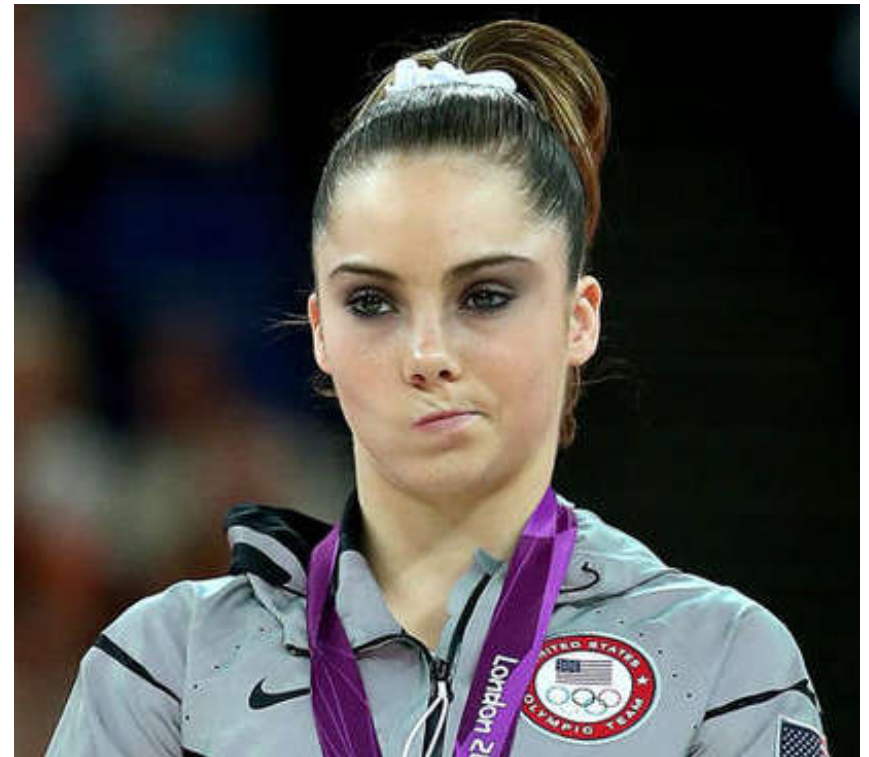


© Bethesda Softworks

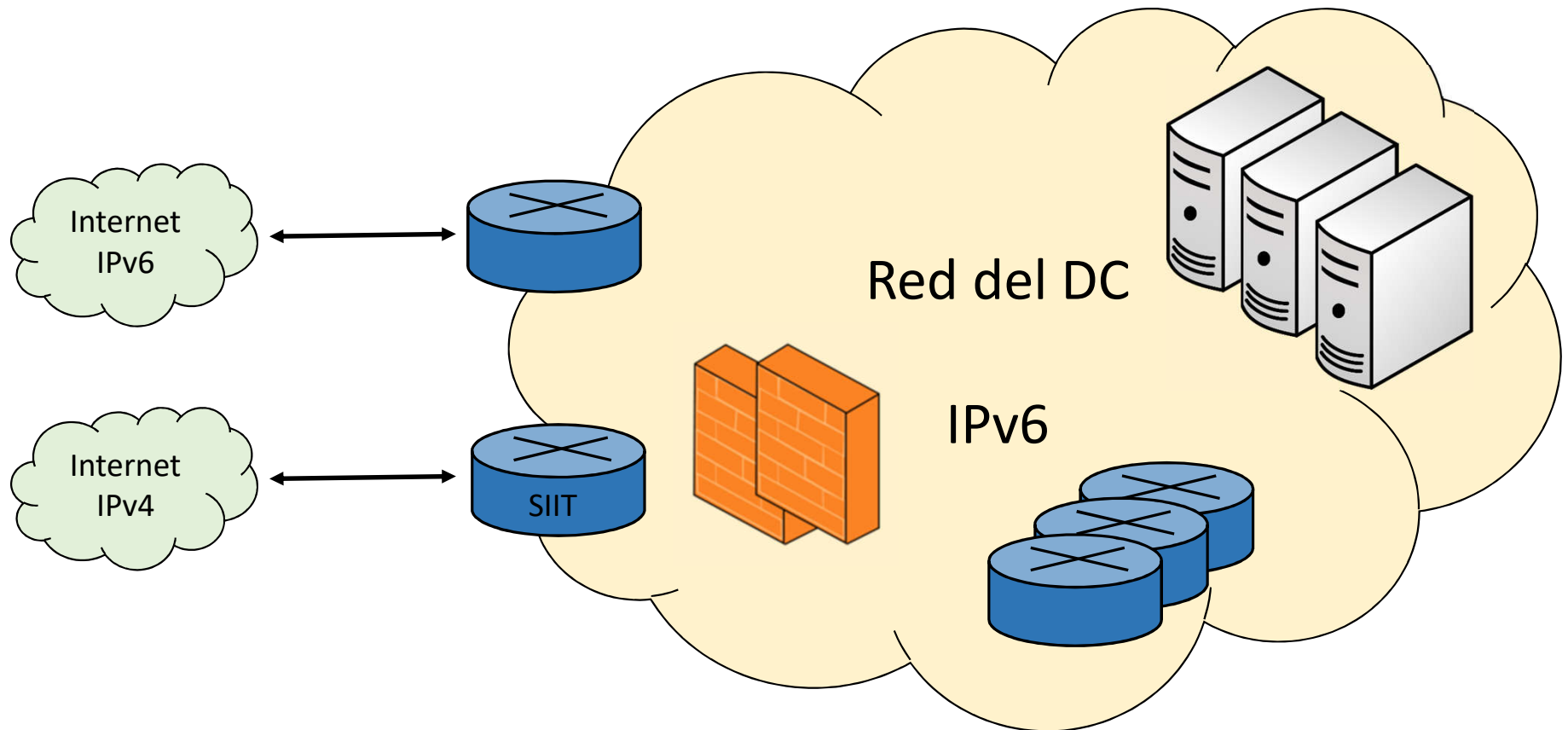
Pero... no soy un ISP

Soy un administrador de sistemas que está a cargo de un grupo de servidores que dan servicio de contenidos y aplicaciones de Internet.

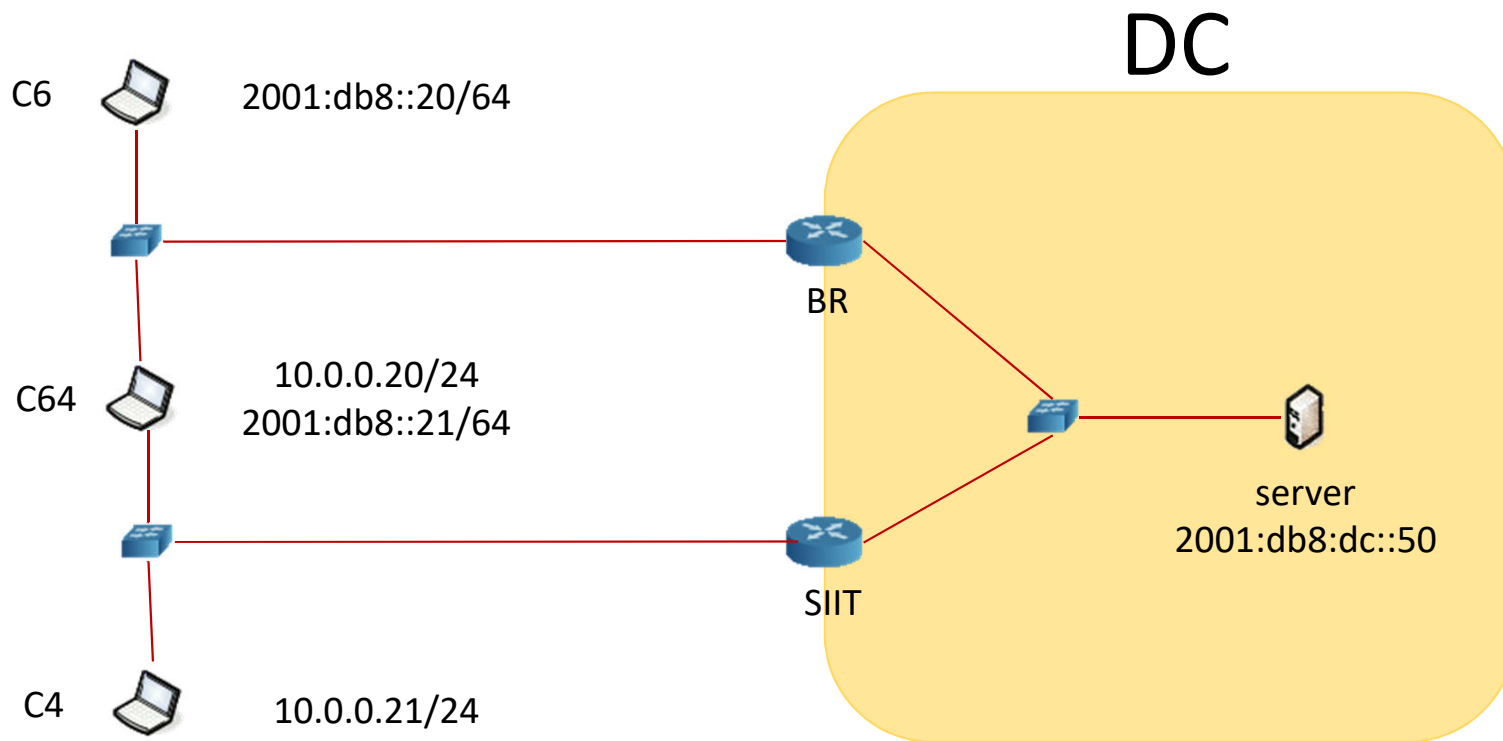
¿Existen esquemas de transición para mi caso?



SIIT-DC



Práctica 5: SIIT-DC



Práctica 5: SIIT-DC

1. Nodo SIIT:

```
user@SIIT:~# jool_siit instance add --netfilter --pool6 2001:db8:46::/96  
user@SIIT:~# jool_siit eamt add 192.168.0.80/32 2001:db8:dc::50/128
```

2. Nodo C4:

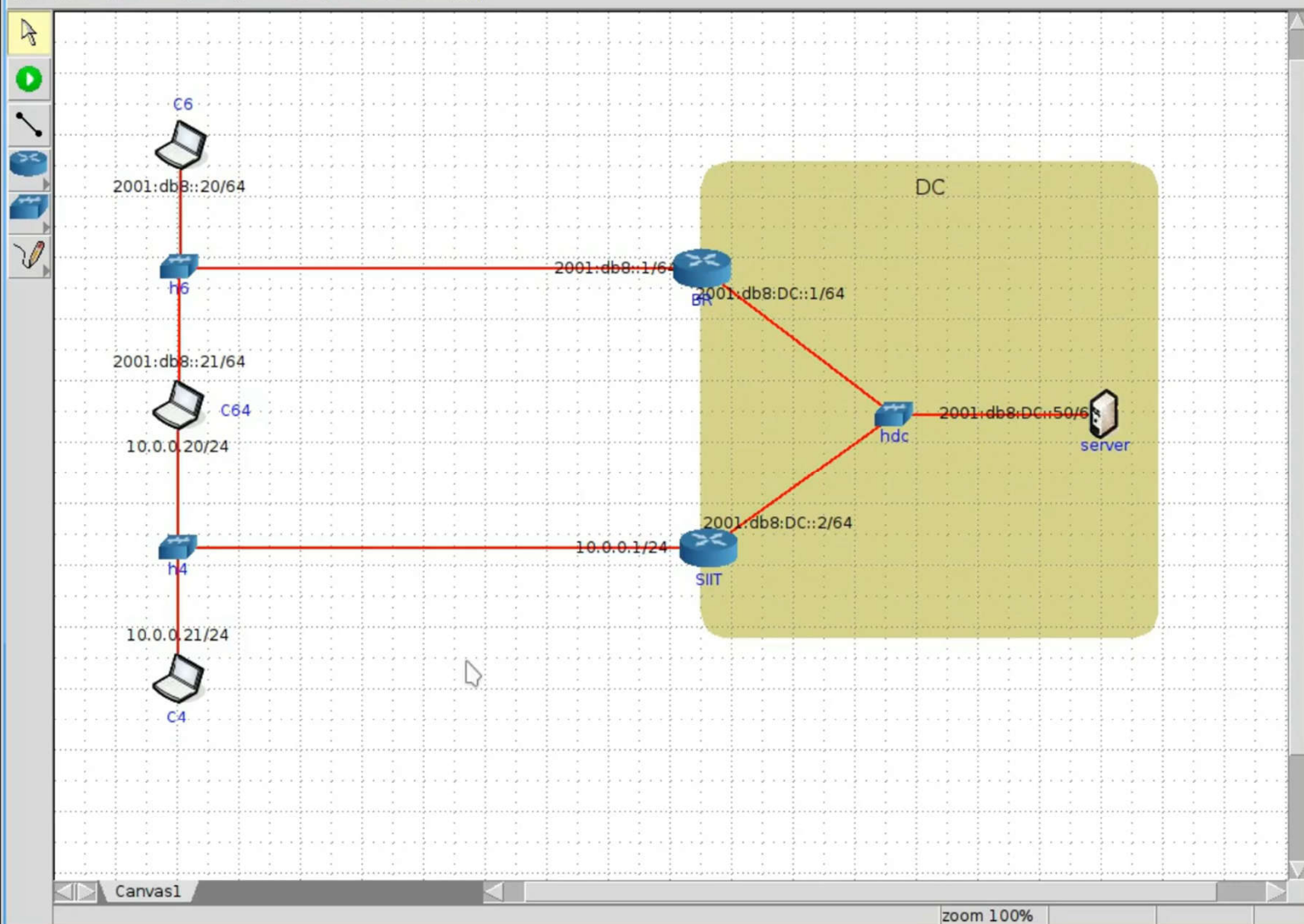
```
user@C4:~# ping 192.168.0.80
```

3. Nodo C6:

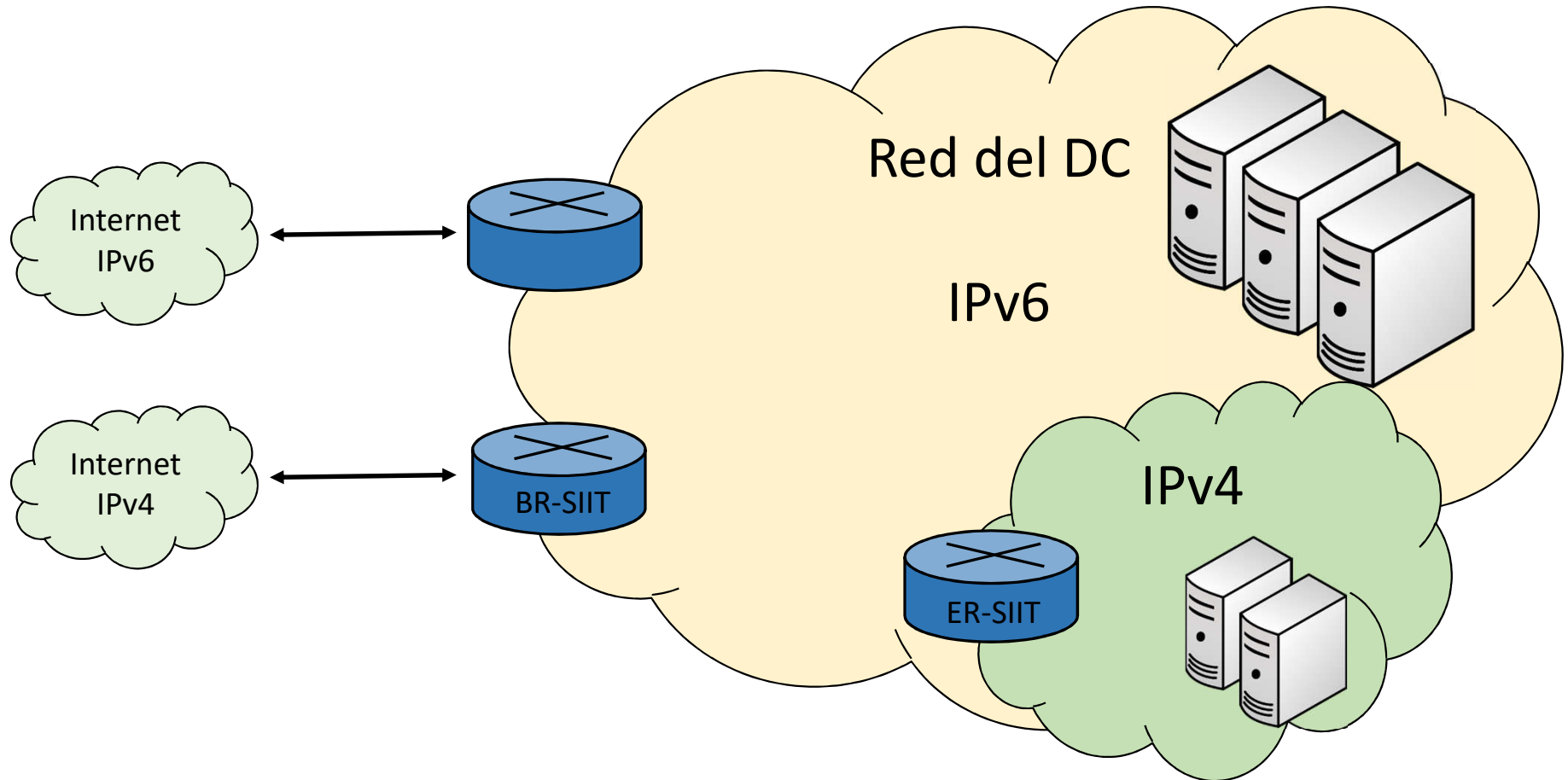
```
user@C6:~# ping 2001:db8:dc::50
```

4. Nodo C64:

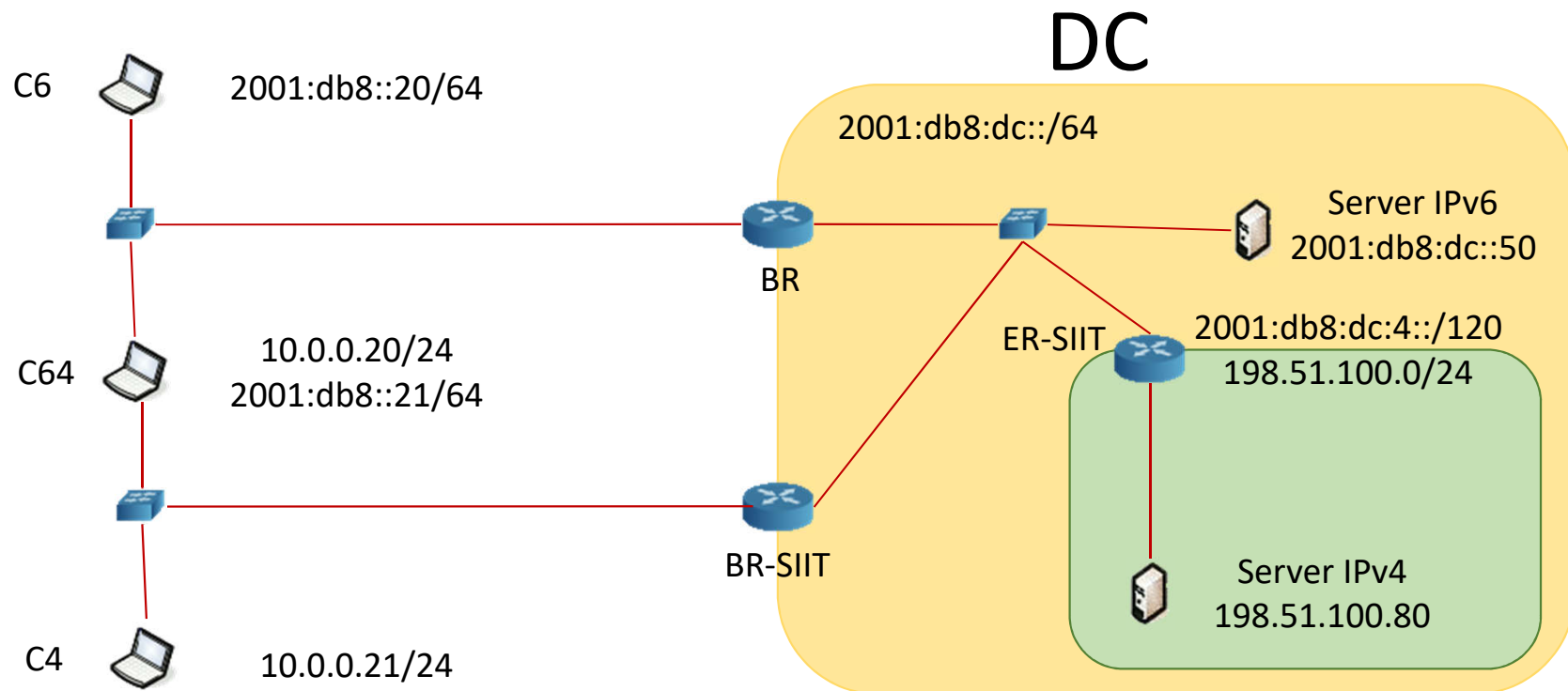
```
user@C64:~# ping 192.168.0.80  
user@C64:~# ping 2001:db8:dc::50
```



SIIT-DC: Dual Translation Mode



Práctica 6: SIIT-DC-DTM



Práctica 6: SIIT-DC-DTM

1. Nodo BR-SIIT:

```
user@BR-SIIT:~# jool_siit instance add --netfilter --pool6 2001:db8:46::/96
user@BR-SIIT:~# jool_siit eamt add 192.168.0.80/32 2001:db8:dc::50/128
user@BR-SIIT:~# jool_siit eamt add 192.168.0.81/32 2001:db8:dc:4::50/128
```

2. Nodo ER-SIIT:

```
user@ER-SIIT:~# jool_siit instance add --netfilter --pool6 2001:db8:46::/96
user@ER-SIIT:~# jool_siit eamt add 198.51.100.0/24 2001:db8:dc:4::/120
```

3. Nodo C4:

```
user@C4:~# ping 192.168.0.80
user@C4:~# ping 192.168.0.81
```

Práctica 6: SIIT-DC-DTM

1. Nodo C6:

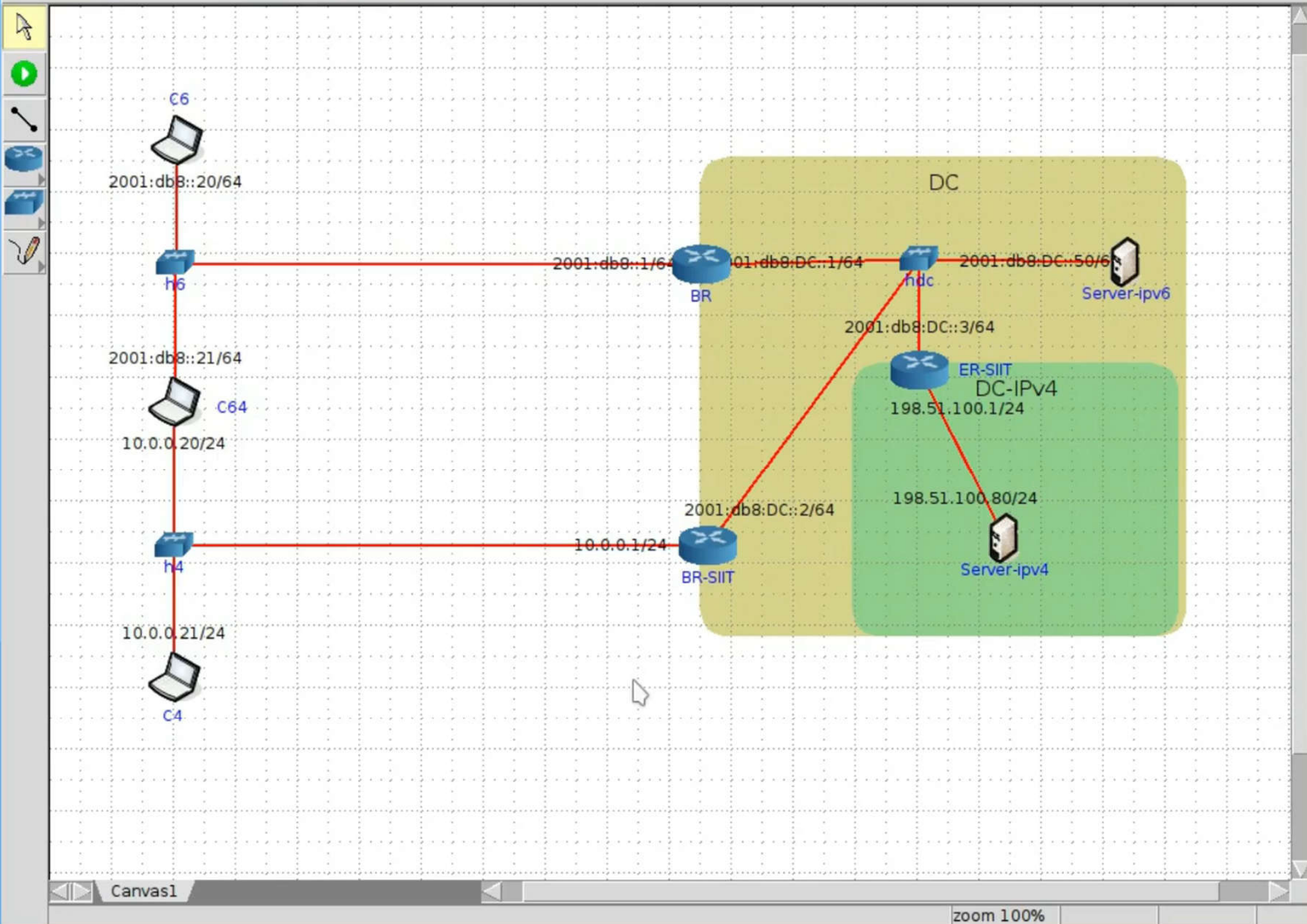
```
user@C6:~# ping 2001:db8:dc::50
```

2. Nodo C64:

```
user@C64:~# ping 192.168.0.80
```

```
user@C64:~# ping 192.168.0.81
```

```
user@C64:~# ping 2001:db8:dc::50
```



Jool fuera del simulador

- Cargar los módulos en el kernel.

```
$ modprobe jool
```

```
$ modprobe jool_siit
```

- Apagar offloads

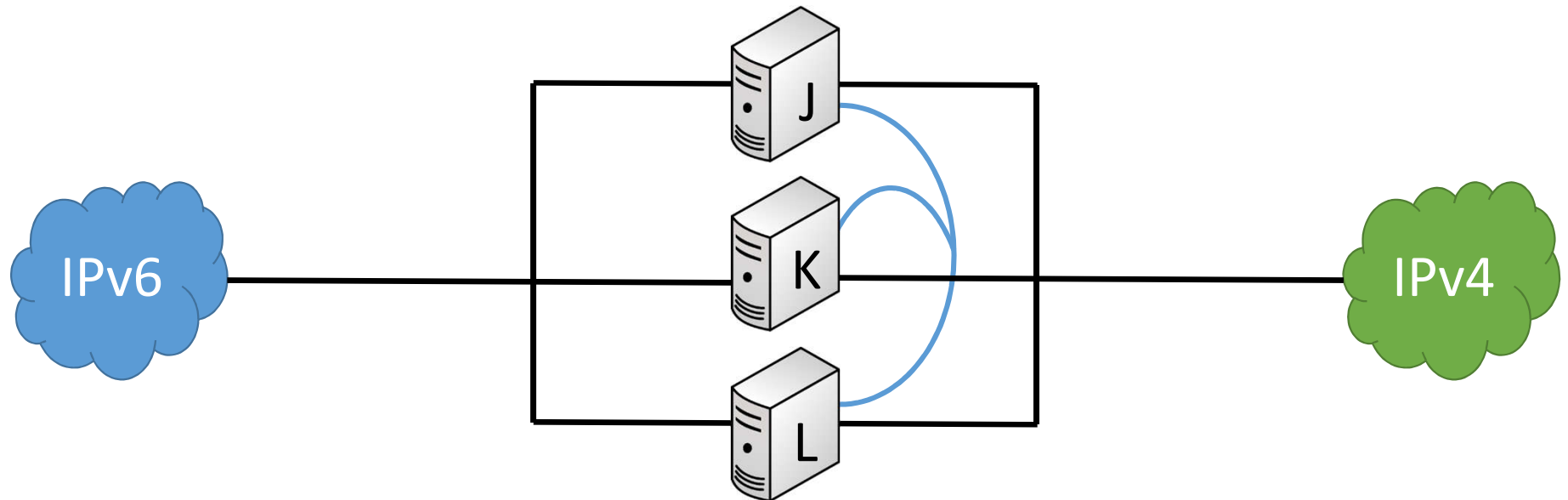
```
$ ethtool --offload [your interface here] lro off
```

```
$ ethtool --offload [your interface here] gro off
```

(Si se usa Jool en máquinas virtuales, es recomendable también apagar los offloads de la máquina host)

Alta disponibilidad

Es posible tener más de una instancia de Jool que comparten las sesiones para tener un esquema de alta disponibilidad.



Para más información: <http://jool.mx/en/session-synchronization.html>

Instancias

- Es posible tener más de una instancia de Jool corriendo al mismo tiempo.

Formato:

```
$ jool instance add <etiqueta> ...
```

Ejemplos:

```
$ jool instance add "ejemplo" ...
```

```
$ jool instance add "otra" ...
```

```
$ jool instance add ...
```

- Si no se asigna una etiqueta a la instancia, se asigna una llamada "default".

Frameworks

A partir de la versión 4.0 de Jool, se tiene disponible dos frameworks en los que Jool puede operar:

- Netfilter

Corresponde a la funcionalidad tradicional que Jool tiene desde la versión 3.X.

- Iptables

Permite utilizar Jool como un plug-in más de iptables y poder definir reglas de tráfico.

Cuidado:

- Netfilter Jool sigue la política: “Si no lo sé traducir, lo regreso al kernel y él decidirá que hacer”.
- Iptables Jool sigue la política: “Todo lo que hizo match a una regla es mío y si no lo sé traducir lo descarto”.

Frameworks

Ejemplo de Jool usando iptables:

```
$ jool instance add "example" --iptables --pool6 64:ff9b::/96
```

```
$ ip6tables -t mangle -A PREROUTING -d 64:ff9b::/96 -j J00L --instance "example"
```

```
$ iptables -t mangle -A PREROUTING -d 203.0.113.1 -p tcp --dport 61001:65535 -j J00L --instance "example"
```

```
$ iptables -t mangle -A PREROUTING -d 203.0.113.1 -p udp --dport 61001:65535 -j J00L --instance "example"
```

```
$ iptables -t mangle -A PREROUTING -d 203.0.113.1 -p icmp -j J00L --instance "example"
```

Bitácora de traducción

Jool puede guardar en la bitácora todas las traducciones que realiza.

```
$ jool global update logging-bib true
```

```
$ dmesg
```

```
[ 312.493235] alpha 2015/4/8 16:13:42 (GMT) - Mapped 2001:db8::5#19945 to 192.0.2.2#8208 (UDP)
[ 373.724229] alpha 2015/4/8 16:14:23 (GMT) - Mapped 2001:db8::5#46516 to 192.0.2.2#7369 (UDP)
[ 468.675528] alpha 2015/4/8 16:15:38 (GMT) - Forgot 2001:db8::5#19945 to 192.0.2.2#8208 (UDP)
```

Bitácora de sesiones

Jool también puede guardar en la bitácora todas las sesiones que son creadas y destruidas.

<date / time> - <action> session <IPv6 node>|<IPv6 representation of IPv4 node>|<IPv4 representation of IPv6 node>|<IPv4 node>|<Protocol>

```
$ jool global update logging-session true
```

```
$ dmesg
```

```
[ 3238.087902] alpha 2015/4/8 17:1:47 (GMT) - Added session 1::5#47073|64:ff9b::c000:205#80|192.0.2.2#63527|192.0.2.5#80|TCP
[ 3238.099997] alpha 2015/4/8 17:1:47 (GMT) - Added session 1::5#47074|64:ff9b::c000:205#80|192.0.2.2#42527|192.0.2.5#80|TCP
[ 3478.498559] alpha 2015/4/8 17:5:48 (GMT) - Forgot session 1::5#47073|64:ff9b::c000:205#80|192.0.2.2#63527|192.0.2.5#80|TCP
[ 3478.499758] alpha 2015/4/8 17:5:48 (GMT) - Forgot session 1::5#47074|64:ff9b::c000:205#80|192.0.2.2#42527|192.0.2.5#80|TCP
```

Archivo de configuración

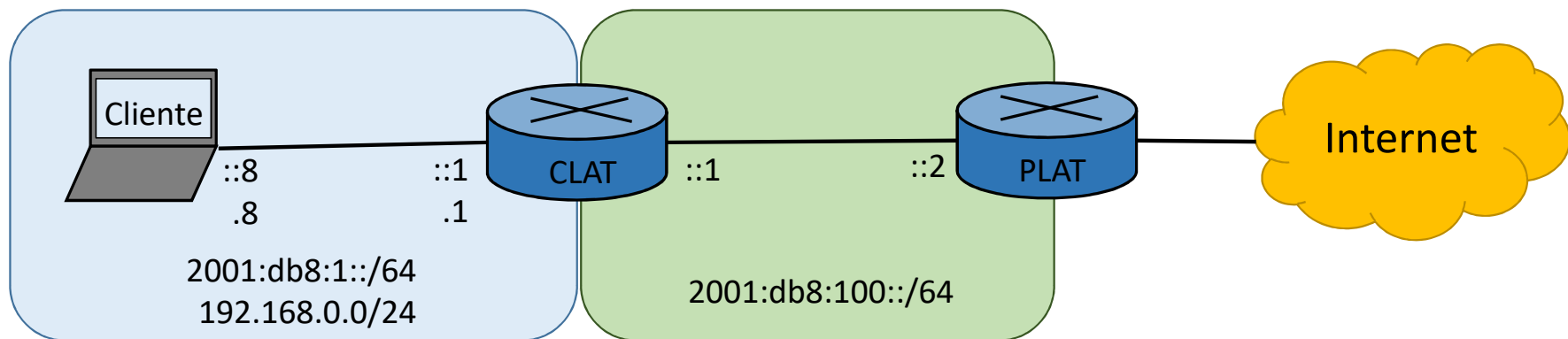
- Configuración por medio de un archivo en formato JSON.
- La aplicación de la configuración es de forma atómica.

```
# jool_siit file handle <archivo de configuración>  
# jool      file handle <archivo de configuración>
```

Archivo de configuración

```
{ "comment": "Configuración SIIT de ejemplo",  
  "instance": "ejemplo",  
  "framework": "netfilter",  
  "global": {  
    "pool6": "64:ff9b::/96",  
  },  
  "eamt": [  
    { "ipv6 prefix": "2001:db8:1::/128", "ipv4 prefix": "192.0.2.0" },  
    { "ipv6 prefix": "2001:db8:2::/120", "ipv4 prefix": "198.51.100.0/24" }  
  ]  
}
```

Práctica 7: 464XLAT (VirtualBox)



Práctica 7: 464XLAT (VirtualBox)

1. Nodo PLAT:

```
# jool instance add --netfilter --pool6 64:ff9b::/96
```

2. Nodo CLAT:

```
# jool_siit instance add --netfilter --pool6 64:ff9b::/96
```

```
# jool_siit eamt add 192.168.0.0/24 2001:db8:2::/120
```



Desempeño

- Medir el desempeño de Jool todavía es un “Trabajo en Progreso”
- Pero tenemos algunos puntos de referencia que podemos reportar:
 - Respuesta de la comunidad
 - Prueba de T-Rex

Respuesta de la comunidad

- En general hemos recibido buenos comentarios de la comunidad sobre el desempeño de Jool.
- Se han abierto algunos bugs sobre el desempeño, pero la solución siempre ha resultado ser un ajuste ajeno a Jool.

Respuesta de la comunidad

[Nuestros SIIT-BRs] están prácticamente ociosos. Están traduciendo aproximadamente 100Mb/s de casi exclusivamente tráfico web. De hecho, el hardware es bastante viejo; Sun X4170s con 2x quad-core Intel L5520 CPUs. Menos de la cuarta parte de un solo CPU se usa para todo el sistema (así que no solamente Jool), los 7.75 restantes están libres.

Otra cosa que vale la pena mencionar es que dado que SIIT es stateless, puedes escalar horizontalmente fácilmente. Solo agrega más máquinas traductoras con configuración idéntica, y corre balanceo de carga entre ellas.

En general alcanzas los límites de memoria de bus antes que los del CPU.

<https://mail-lists.nic.mx/pipermail/jool-list/>

Prueba de T-Rex

Prueba de carga natural implementada por Cisco. La variante “Advance Stateful Support” soporta NAT64.

<https://trex-tgn.cisco.com>

CentOS 7 Dell R630 con dual Xeon E5-2680 v4 (28 cores en total), 64 GB de RAM y NIC Intel X710.

Prueba de T-Rex

-Per port stats table

ports	0	1

opackets	14987324	41611624
obytes	1776772320	56340402998
ipackets	41610306	14987530
ibytes	57170300382	1477031674
ierrors	0	0
oerrors	0	0
Tx Bw	23.74 Mbps	752.99 Mbps

-Global stats enabled

Cpu Utilization : 4.4 % 35.0 Gb/core

Platform_factor : 1.0

Total-Tx : **776.73 Mbps**

Total-Rx : **783.78 Mbps**

Total-PPS : 94.55 Kpps

Total-CPS : 2.78 Kcps

Expected-PPS : 0.00 pps

Expected-CPS : 0.00 cps

Expected-BPS : 0.00 bps

Active-flows : 6665 Clients : 256 Socket-util : 0.0413 %

Open-flows : 1665014 Servers : 65536 Socket : 6665 Socket/Clients : 26.0

drop-rate : **0.00 bps**

current time : 603.5 sec

test duration : 0.0 sec

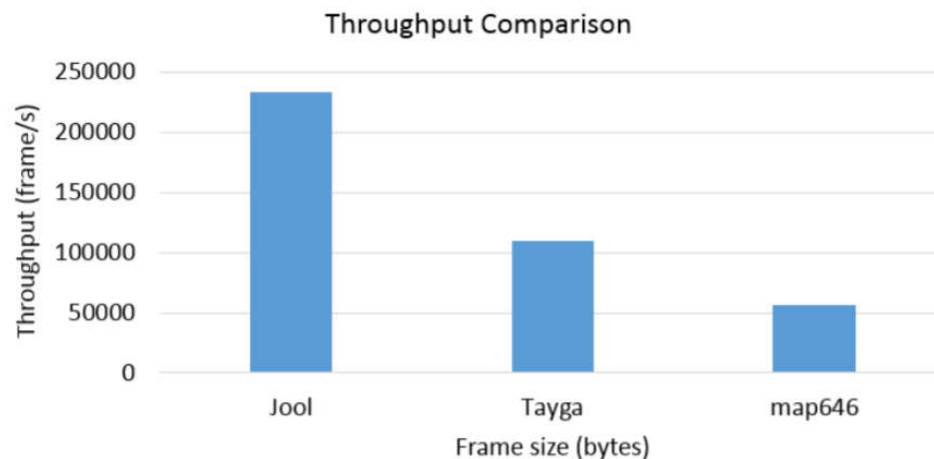
- ~780 Mbps
- 0 drop rate
- ~1% utilización de CPU (el 4.4% de la izquierda se refiere al cliente, no al traductor).

- Jool es 100% CPU, de modo que no parece estar imponiendo un significativo overhead

Respuesta de la comunidad

Benchmarking Stateless NAT64 Implementations with a Standard Tester

Gábor Lencse



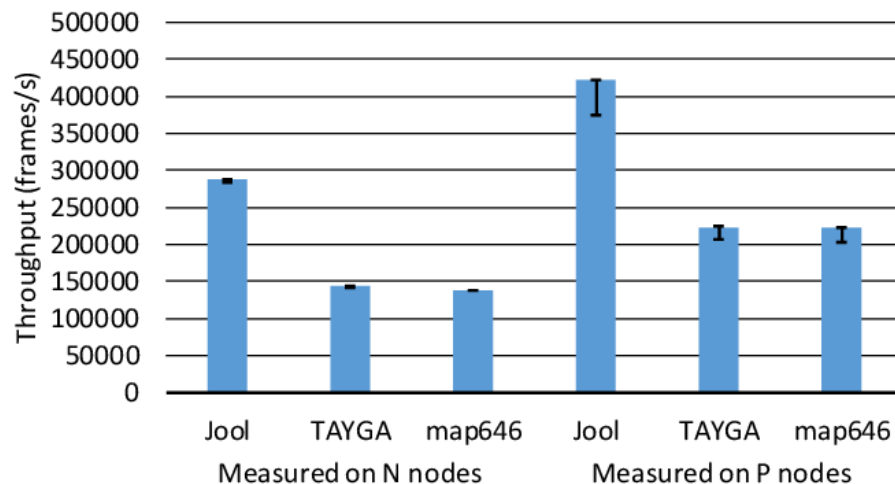
“We have found that Jool showed both the highest throughput and the most consistent behavior in the frame loss tests, thus we recommend Jool for new deployments. We hope that our results may contribute to the global deployment of the IPv6 protocol.”

<https://www.semanticscholar.org/paper/Benchmarking-Stateless-NAT-64-Implementations-with-Tester-Lencse/03c7f161472c2235421fd9e26eab9bc407855d50>

Respuesta de la comunidad

Performance Analysis of SIIT Implementations: Testing and Improving the Methodology

Gábor Lencse, Keiichi Shima



“We have performed the throughput and frame loss rate benchmarking tests required by RFC 8219 to analyze the performance of Jool, TAYGA and map646 and found that the performance of Jool scaled up well with the number of active CPU cores, and also significantly outperformed TAYGA and map646.”

<https://www.semanticscholar.org/paper/3-Benchmarking-Measurements-with-a-Stateless-NAT-64-Lencse-Shima/1a08ee61413ef5402c59da45c3e419f69ed0e451>

Jool en el mundo

nat64check

Check: <http://www.example.com/>



<https://nat64check.go6lab.si/>

<https://nat64check.ipv6-lab.net/>



NAT64 public test in Go6lab

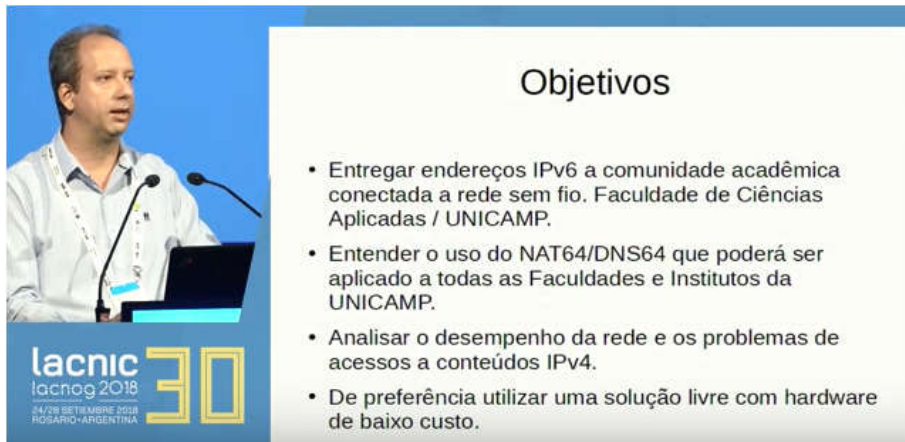
<https://go6lab.si/>

Jool en el mundo

Entrega de IPv6-only al usuario final utilizando NAT64

Henri Alves Godoy

Universidade Estadual de Campinas



Objetivos

- Entregar endereços IPv6 a comunidade acadêmica conectada a rede sem fio. Faculdade de Ciências Aplicadas / UNICAMP.
- Entender o uso do NAT64/DNS64 que poderá ser aplicado a todas as Faculdades e Institutos da UNICAMP.
- Analisar o desempenho da rede e os problemas de acessos a conteúdos IPv4.
- De preferência utilizar uma solução livre com hardware de baixo custo.

lacnic
lacnog 2018
24/28 SETEMBRO 2018
ROSARIO • ARGENTINA

https://youtu.be/-dZP_mUVe-Y

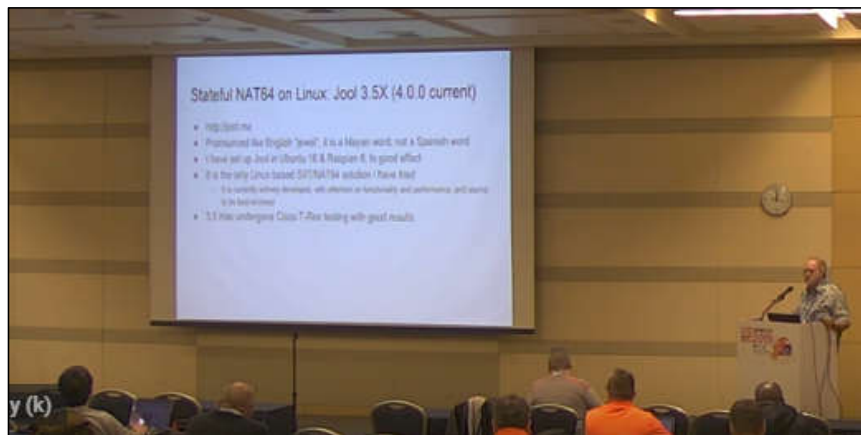


Jool en el mundo

The IPv6-Only Network: Building Networks with DNS64/NAT64/464XLAT

Alan Whinery

University of Hawaii



<https://youtu.be/5l19jNUsFg8>



Jool en el mundo

NAT64

IPv6アドレスとしてアクセスする先がわかったので、このアドレスを用いて通信をする。

利用したのは<https://www.jool.mx/>にあるjoolである。

joolのインストール

1. カーネルのモジュールとして用意されているので、もしかするときちんとカーネルを作る必要があるかもしれない。
2. カーネルの条件としては、IPv4. IPv6は当然であるがNetFilter関係も最低限有効にしておかないとmakeできない。
3. 展開して、modの中でmakeするだけ
4. `make modules_install`でモジュールjoolがインストールされる。

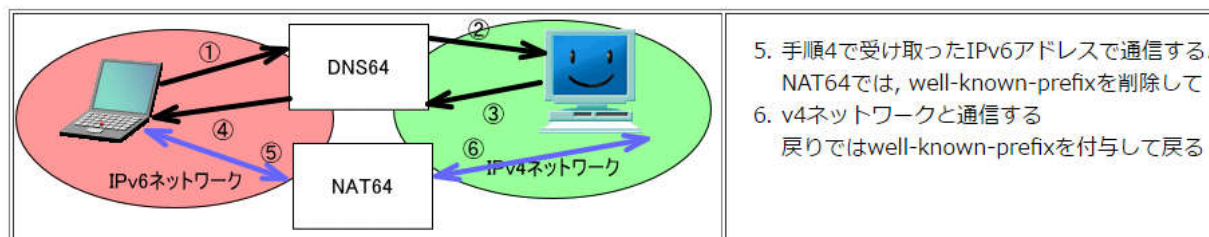
joolの起動

モジュールとして用意してあるので

- `modprobe jool pool4=v4側のアドレス`

で起動する。pool6もあるが標準ではDNS64で利用するwell-known-prefix 64:ff9b::/96なので、通常は変更する必要はない

- 新しいバージョンのjoolではpool6を明示的に指定する必要がある指摘をいただきました(2015/11/12)



Jool en el mundo

NAT64

いよいよ本題の NAT64 です。OpenBSD なら 標準機能の PF が NAT64 に対応していますが、Linux の場合にはサードパーティのソフトウェアをインストールする必要があります。ざっと調べた限り

Li

か

の

し

【

Jo

常

イ

応

ん

っ

Japanese - detected	English
ざっと調べた限り Linux なら「Jool」を使うのが一番良さそうです。Jool は Netfilter と連携して NAT64 を実現するカーネルモジュールで、NIC Mexico（メキシコの国別インターネットレジストリ）が開発しています。	Roughly examined as long as there is to use a Linux " Jool " is the most good. Jool is a kernel module to achieve a NAT64 in conjunction with Netfilter, NIC Mexico (Mexico's National Internet Registry) has been developed .

Open in Google Translate

レパッケージを使うことにします。

<http://www.niksula.hut.fi/~mhienka/Rpi/linux-headers-rpi/> にラズパイ用のカーネルヘッダの deb パッケージが公開されているので、カーネルバージョンに対応するパッケージをダウンロードしてインストールします。

Jool en el mundo

	French – detected	English	
Pour la mise en place d'un tel routeur Ecdysis ne fait seulement que 50% de ce que la RFC du nat64 demande, fonctionne juste de IPv6 à IPv4, simple et rapide a mettre en place mais dans un stade de développement presque a l'arrêt, pour l'inverse je parlerai de Jool qui est capable de faire les deux et qui sont encore en phase de développement pour respecté a 100% les RFC .	Pour la mise en place d'un tel routeur une simple Debian suffit avec les modules noyaux Jool.	For the establishment of such a router simply enough with Jool Debian kernel modules .	rapide a metre en place mais dans un stade de développement presque a l'arrêt, pour l'inverse je parlerai de Jool qui est capable de faire les deux et qui sont encore en phase de développement pour respecté a 100% les RFC .
Prérequis: la possession d'un serveur Une machine Routeur Une machine Client je passe la partie configuration	Ecdysis ne fait seulement que 50% de ce que la RFC du nat64 demande, fonctionne juste de IPv6 à IPv4, simple et rapide a mettre en place mais dans un stade de développement presque a l'arrêt, pour l'inverse je parlerai de Jool qui est capable de faire les deux et qui sont encore en phase de développement pour respecté a 100% les RFC .	Ecdysis does only 50 % of what the RFC NAT64 request, works just of IPv6 to IPv4 , simple and fast meter was in place but in a stage of development has almost stopped, for the reverse I talk about Jool that is able to do both and are still in the development phase to comply 100% RFC .	rapide a metre en place mais dans un stade de développement presque a l'arrêt, pour l'inverse je parlerai de Jool qui est capable de faire les deux et qui sont encore en phase de développement pour respecté a 100% les RFC .
1. Installation de Jool Un Kernel Linux en			rapide a metre en place mais dans un stade de développement presque a l'arrêt, pour l'inverse je parlerai de Jool qui est capable de faire les deux et qui sont encore en phase de développement pour respecté a 100% les RFC .

Jool en el mundo

Implementeringer

- sten, port oversætter til Windows & UNIX-baserede systemer.

Danish – detected ▾	↔	English ▾	🔊
Jool, en implementering til Linux af RFC6146, som er udviklet af Monterrey Institute of Technology og NIC Mexico		Jool , an implementation for Linux of RFC6146 , which is developed by the Monterrey Institute of Technology and NIC Mexico	

- IVI IPv4 / IPv6 pakke oversættelse implementering som en Linux-kernen patch
- Microsoft Forefront Unified Access Gateway, en reverse proxy og VPN-løsning, der implementerer DNS64 og NAT64
- BIND, Berkeley Internet Name Domain DNS-server, implementerer DNS64 siden version 9.8

Jool en el mundo

5.4. NAT64 implementációk stabilitása és teljesítménye

A témával kapcsolatos DNS64 implementáció stabilitásának állapotát vizsgáljuk.

Először rögzített tesztelési módszert, utána közzéadjuk és értelmezzük a mérési eredményeinket.

Hungarian – detected

Vizsgálataink elvégzése után találtunk rá a Linux alatti Jool [58] állapottartó (stateful) NAT64 implementációra, amelynek a teljesítményét és stabilitását a későbbiekben szintén tervezzük megvizsgálni.

English

After completing our study we found him in the Linux Jool [58] stateful NAT64 implementation with of performance and stability To accomplish this we also plan to investigate

en [56] (egy cícióval foglalt NAT64 4 implementált Jool [58] bilitását a későbbiekben kidol-

5.4.1. Korábbi tudományos eredmények

Let's play a game



Contacto

Página Oficial	https://jool.mx
Repositorio del código	https://github.com/NICMx/Jool
Discusión pública y noticias	jool-list@nic.mx
Noticias	jool-news@nic.mx
Discusión privada	jool@nic.mx

¿Preguntas?



¡Muchas Gracias!

