



Taller: Mecanismos de Transición IPv6 (Práctica Transición)

Alvaro Vives, Consulintel
alvaro.vives@consulintel.es

Taller: Mecanismos de Transición IPv6
20-22 Junio 2011
Honduras Tegucigalpa



CLARA

This project is funded
by the European Union

A project implemented
by CLARA



1. Práctica Transición

1.1 Configuración red hosts

1.2 Apagón IPv4

1.3 Túneles



ConsultIntel The IPv6 Company



1.1 Configuración red hosts

Configuración red IPv6

- Se pretende dar una visión práctica de los mecanismos de autoconfiguración de hosts en una red IPv6
- Hay dos tipos de autoconfiguración: SLAAC y DHCPv6
- RA tienen dos bits: M y O
- El alumno practicará con servidores/clientes de RA y DHCPv6
- Se capturarán paquetes para verificar qué está pasando
- A tener en cuenta mal soporte de clientes DHCPv6



SLAAC (1)

- Instalar radvd:

```
# sudo apt-get install radvd
```

- Es necesario habilitar reenvío de paquetes IPv6:

```
[# sudo su]
```

```
# echo 1 > /proc/sys/net/ipv6/conf/all/forwarding
```

```
[# exit ]
```

- Para habilitar reenvío de forma permanente, se puede editar /etc/sysctl.conf y descomentar:

```
#net.ipv6.conf.all.forwarding=1
```

- El siguiente paso es **crear** el fichero de configuración /etc/radvd.conf (ver siguiente slide)

- Por último hay que lanzar el demonio:

```
# sudo /etc/init.d/radvd start
```



SLAAC (2)

- Fichero de configuración /etc/radvd.conf PCI:

```
interface eth1
{
    AdvSendAdvert on;
    MinRtrAdvInterval 3;
    MaxRtrAdvInterval 5;
    AdvHomeAgentFlag off;
    AdvManagedFlag off;
    AdvOtherConfigFlag off;

    prefix 2001:db8:1234:10i::/64 -----> CAMBIAR 10i !!!
    {
        AdvAutonomous on;
        AdvOnLink on;
        AdvRouterAddr off;
    };
};
```



SLAAC (3)

- Capturar paquetes con wireshark
- Filtrar paquetes icmpv6
- Comprobar en Ubuntu virtualizado y SO host (windows):
 1. Direcciones en interfaces
 2. Tabla rutas
- Deshabilitar autoconfiguración en interfaz de Ubuntu:

```
up echo 0 > /proc/sys/net/ipv6/conf/eth0/autoconf
```
- Reiniciar red:

```
# sudo /etc/init.d/networking restart
```
- Comprobar interfaz y rutas
- Parar radvd:

```
# sudo /etc/init.d/radvd stop
```



Ciente DHCPv6 (1)

- Instalar cliente DHCPv6:
`#sudo apt-get install dibbler-client`
- Elegir si lanzar el cliente al iniciar:

```
dibbler-client: DHCPv6 client

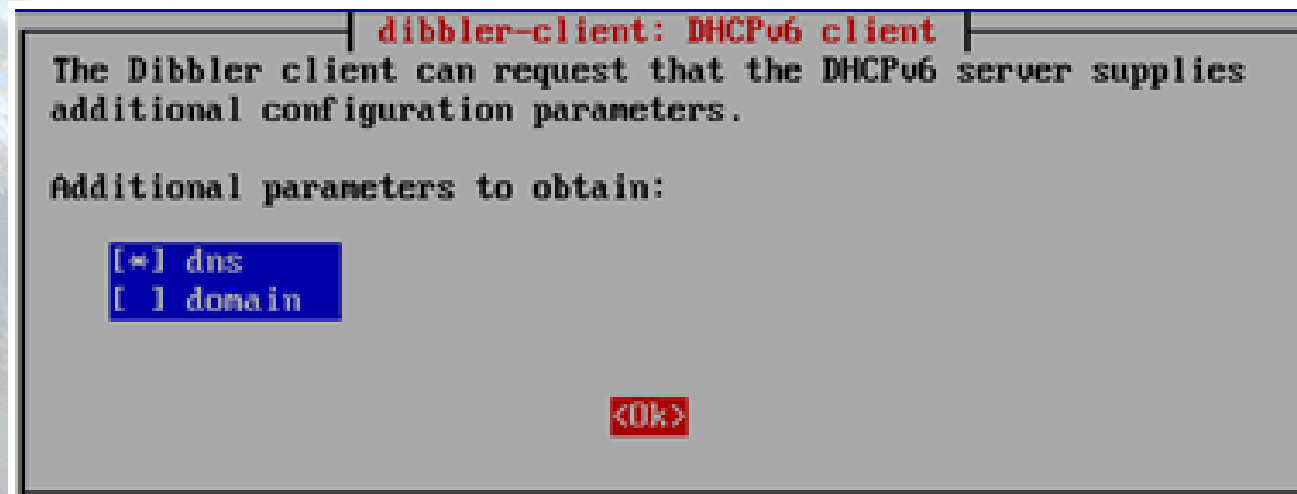
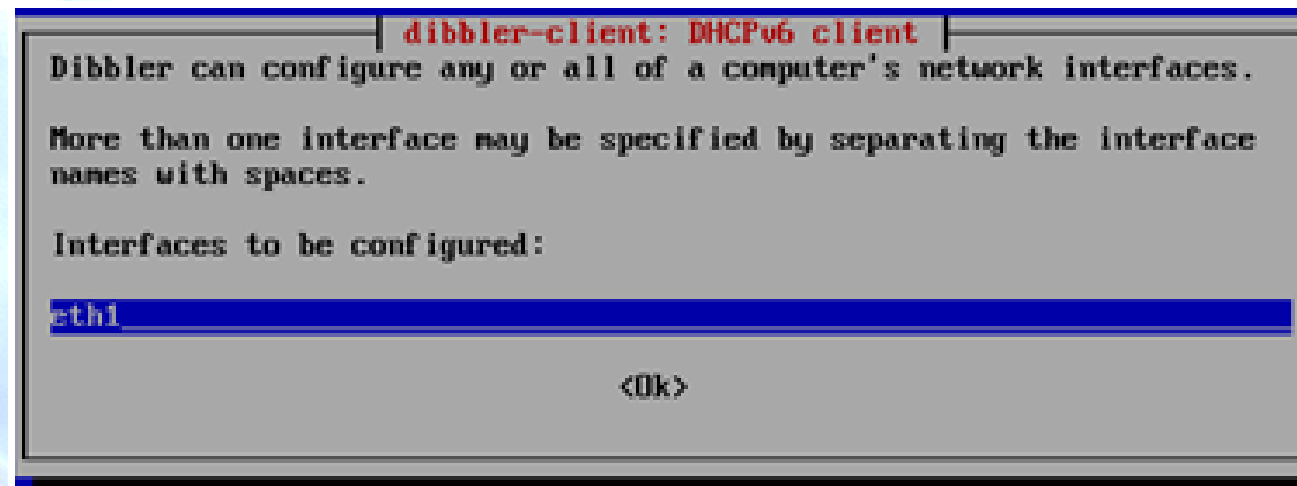
The Dibbler client can be configured to be launched when the system is
started. If you choose this option, this host will have a correct IPv6
setup after booting. Please ensure that a DHCPv6 server is available on
the network.

Should the Dibbler client be launched when the system starts?

<Yes> <No>
```


Ciente DHCPv6 (2)

- Configurar la interfaz adecuada y los parámetros a preguntar (luego se pueden cambiar):



Ciente DHCPv6 (3)

- Cuando sólo se quiere recibir el servidor DNS, esto es lo que se conoce como **stateless DHCPv6**
- El fichero /etc/dibbler/client.conf tendrá algo como:

```
iface eth0
{
  #ia
  option dns-server
  # option domain
}
```

- Iniciar cliente DHCPv6:

```
# sudo /etc/init.d/dibbler-client start
```

- Capturar paquetes DHCPv6
- Comprobar /etc/resolv.conf



Servidor DHCPv6 (1)

- Paso previo, configurar correctamente RAs:

```
AdvOtherConfigFlag on;
```

- Instalar servidor DHCPv6:

```
# sudo apt-get install dibbler-server
```

- Seleccionar si arrancar el servidor al iniciar:

```
dibbler-server: DHCPv6 server

The Dibbler server can be configured to be launched when the system is
started. If you choose this option, this node will act as a DHCPv6
server. It will provide IPv6 addresses and additional configuration
options to other nodes in the network.

Should the Dibbler server be launched when the system starts?

<Yes> <No>
```



Servidor DHCPv6 (2)

- Guardamos el fichero de configuración original:

```
#sudo mv /etc/dibbler/server.conf /etc/dibbler/server.conf.orig
```

- Editar la configuración del servidor en /etc/dibbler/server.conf:

```
stateless
```

```
iface eth1 {
```

```
    option dns-server 2001:db8:1:1::10i -> CAMBIA 10i
```

```
    option lifetime 1800
```

```
}
```

- Iniciar servidor DHCPv6, luego el radvd:

```
#sudo /etc/init.d/dibbler-server start
```

```
#sudo /etc/init.d/radvd start
```

- Se puede comprobar el funcionamiento del servidor en /var/log/dibbler/dibbler-server.log



Servidor DHCPv6 (3)

- Configurar ahora **stateful autoconfiguration**:
- Primero los RAs (/etc/radvd.conf):

```
AdvManagedFlag on;  
AdvOtherConfigFlag on;
```

- Configurar cliente DHCPv6 para que obtenga dirección y servidor DNS (/etc/dibbler/client.conf):

```
iface eth1  
{  
  ia  
  option dns-server  
  # option domain  
}
```



Servidor DHCPv6 (4)

- Configurar servidor DHCPv6 en /etc/dibbler/server.conf :

```
# stateless
iface eth1 {
    class {
        pool 2001:db8:1234:10i::/64 -> CAMBIA 10i
    }
    option dns-server 2001:db8:1:1::10i -> CAMBIA 10i
    option lifetime 1800
}
```

- Iniciar servidor DHCPv6, luego radvd:

```
# sudo /etc/init.d/dibbler-server start
# sudo /etc/init.d/radvd start
```



1.2 Apagón IPv4



ConsulIntel The IPv6 Company

Introducción

- Se simulará una red solo-ipv6 y los aspectos a tener en cuenta:
 1. Configuración DNS
 2. Problemas RA
 3. Acceso a contenido solo-IPv4

Configuración DNS (1)

- En una red solo-IPv6 la única forma de acceder al servicio DNS es sobre IPv6:
 - Windows XP no soporta tráfico DNS sobre IPv6
 - Se debe configurar la dirección(es) IPv6 de servidor DNS. Distintas opciones: manual, DHCPv6, RA
- Vamos a configurar RA para que configure un servidor DNS:
 - RFC6106 (IPv6 Router Advertisement Options for DNS Configuration), Nov. 2010
 - RDNSS daemon for Linux: <http://rdnssd.linkfanel.net>
 - rdnssd-win32: <http://sourceforge.net/projects/rdnssd-win32/>



Configuración DNS (2)

- Instalar rdnssd:

```
# sudo apt-get install rdnssd
```

- Guarda los servidores DNS vistos en RA en:

```
/var/run/rdnssd/resolv.conf
```

- Utiliza el script /etc/rdnssd/mege-hook para meclarlo con:

```
/etc/resolv.conf
```

- Iniciar rdnssd

```
# sudo /etc/init.d/rdnssd start
```

- Parar rdnssd

```
# sudo /etc/init.d/rdnssd stop
```



Configuración DNS (3)

- Configurar radvd para enviar info de rdns. En /etc/radvd.conf añadir:

```
RDNSS 2001:db8:1:1::1 2001:db8:1:1::2
{
    AdvRDNSSPreference 8;
    AdvRDNSSLifetime 30;
};
```

- Capturar RAs para ver opción RDNSS
- Comprobar si rdnsd ha configurado los servidores

Configuración DNS (4)

- **NOTA EXPLICATIVA:** Según man rdnssd:

When rdnssd uses a raw socket instead of the netlink kernel interface, it does not validate received Neighbor Discovery traffic in any way. For example, it will always consider Router Advertisement packets, whereas it should not if the host is configured as a router. When the netlink interface is used, such validation is done by the kernel.

- Cuando actuamos como router (`/proc/sys/net/ipv6/conf/all/forwarding = 1`) el kernel no pasara los RAs hacia el espacio de aplicaciones de usuario, donde funciona rdnssd
- Solución: `/proc/sys/net/ipv6/conf/all/forwarding = 0`
- radvd se quejará pero podremos ver como se configuran los RDNDSS anunciados
- Normalmente no irán juntos radvd y rdnssd



Problemas RA (1)

- Si hay múltiples nodos que anuncian prefijos mediante RA, puede producir DoS
(RFC6104, Rogue IPv6 Router Advertisement Problem Statement, Feb. 2011)
- Distintas soluciones:
 - RA-GUARD (RFC6105, IPv6 Router Advertisement Guard, Feb. 2011)
 - RAMOND: <http://ramond.sourceforge.net>
- Vamos a practicar con RAMOND



Problemas RA (2)

- Preparamos lo necesario:

```
# sudo apt-get install make gcc libapr1-dev libpcap0.8-  
dev libxml2-dev
```

- Instalamos ramond

```
# wget http://downloads.sourceforge.net/project/ramond/  
ramond/_0.5/ramond-0.5.tar.bz2
```

```
# bunzip2 ramond-0.5.tar.bz2
```

```
# tar -xvf ramond-0.5.tar
```

```
# cd ramond
```

```
# make
```

```
# cp ramond.conf.example ramond.conf
```



Problemas RA (3)

- Configuramos la MAC del router que vamos a permitir y el prefijo que puede anunciar:

```
<!-- All Routers Mac List -->
<mac-list name="ecs-routers">
  <entry>00:c0:26:50:21:15</entry>
</mac-list>

<!-- ECS PREFIX -->
<rule mac="ecs-routers" lifetime="0">
  <execute>./demo.pl native-router-killed</execute>
</rule>
<rule mac="ecs-routers" prefix="2001:db8:1:1::/64">
  <!-- do nothing -->
</rule>
```



Problemas RA (4)

- Ejecutamos ramond:

```
# sudo ./ramond
```

- Vemos los logs (prohibido):

```
2011-06-21-12-01-19: {3} -- listening --  
ramond received a forbidden router-advertisement for  
2001:db8:1234:100::/64  
the source address was fe80::a00:27ff:fe07:3b3d%eth0 with mac  
08:00:27:07:3b:3d
```

- Vemos los logs (prohibido):

```
2011-06-21-12-12-42: {3} -- listening --  
2011-06-21-12-12-45: {3} received a packet from  
fe80::a00:27ff:fe07:3b3d%eth0  
2011-06-21-12-12-45: {3} Matched rule 2
```



Acceso contenido solo-IPv4

- Una red con hosts solo-IPv6 no puede acceder a contenidos solo-IPv4
- Posibles soluciones:
 - Proxy IPv6-IPv4: por ejemplo configurar un proxy web para el acceso a web. El proxy será el que obtenga el contenido de las webs IPv4 y lo reenvíe por IPv6 a los hosts
 - NAT64/DNS64
- Vamos a practicar con NAT64/DNS64





1.3 Túneles

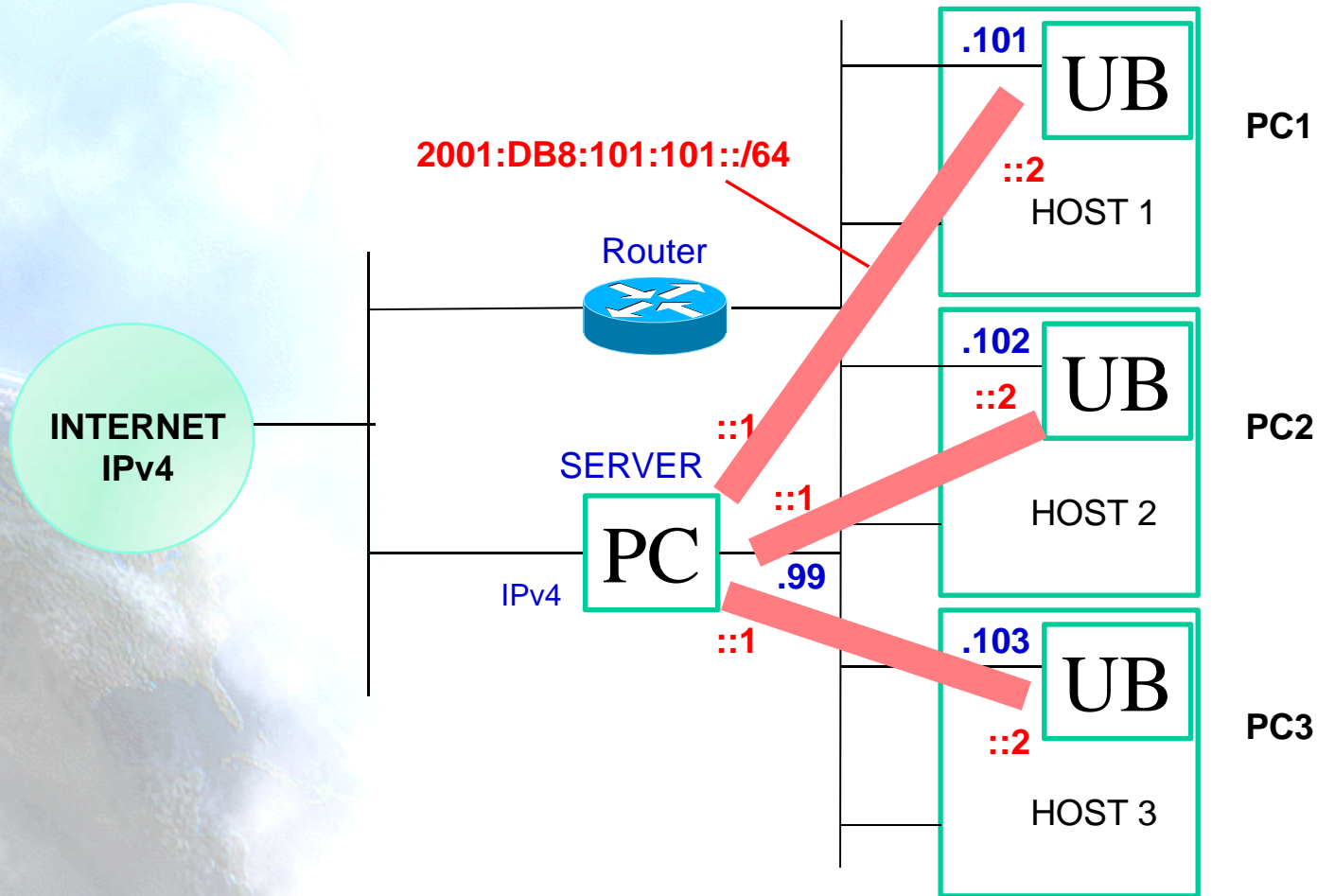


ConsultIntel The IPv6 Company

Introducción

- Suele suceder que nuestro ISP no ofrece IPv6 nativo, o si se realiza un evento, no haya conectividad IPv6 allí
- Una solución rápida y sencilla es la creación de un túnel IPv6-in-IPv4 (6in4)
- Practicaremos con el establecimiento de un túnel

Topología



Configuración túnel 6in4 (1)

- Scripts de creación de túneles 6in4
 - Windows con netsh desde ventana comandos

```
C:\>netsh interface ipv6 add v6v4tunnel "Tunel01"  
Direccion_IPv4_local Direccion_IPv4_remota  
C:\> netsh interface ipv6 add address "Tunel01" Direccion_IPv6  
C:\> netsh interface ipv6 add route ::/0 "Tunel01"  
Direccion_gateway_IPv6 publish=yes
```
 - Linux/UNIX (desde ventana de comandos)

```
# sudo ip tunnel add Tunel01 mode sit remote  
Dir_IPv4_remota local Dir_IPv4_local ttl 255  
# sudo ip link set Tunel01 up  
# sudo ip addr add Dir_IPv6_local/124 dev Tunel01  
# sudo ip route add 2000::/3 dev Tunel01
```

Configuración túnel 6in4 (2)

- Comprobamos que hemos creado el tunel:

```
# ifconfig
Tunnel101 Link encap:IPv6-en-IPv4
Dirección inet6: 2001:db8:2:2::101/124 Alcance:Global
Dirección inet6: fe80::c0a8:1564/128 Alcance:Enlace
ACTIVO PUNTO A PUNTO FUNCIONANDO NOARP MTU:1480 Mátrica:1
Paquetes RX:0 errores:0 perdidos:0 overruns:0 frame:0
Paquetes TX:0 errores:0 perdidos:0 overruns:0 carrier:0
colisiones:0 long.colaTX:0
Bytes RX:0 (0.0 B) TX bytes:0 (0.0 B)
```

- Comprobamos ruta instalada:

```
# ip -6 route show

...
2000::/3 dev Tunnel101 metric 1024 mtu 1480 advmss 1420 hoplimit 0
...
```

Configuración túnel 6in4 (3)

- Se pueden capturar paquetes y ver como van encapsulados

No.	Time .	Source	Destination	Protocol	Info
20	15.815992	2001:db8:2:2::102	2001:db8:2:2::101	ICMPv6	Echo request
21	15.817727	2001:db8:2:2::101	2001:db8:2:2::102	ICMPv6	Echo reply
22	16.813904	2001:db8:2:2::102	2001:db8:2:2::101	ICMPv6	Echo request
23	16.815363	2001:db8:2:2::101	2001:db8:2:2::102	ICMPv6	Echo reply
24	17.811382	2001:db8:2:2::102	2001:db8:2:2::101	ICMPv6	Echo request
25	17.811609	2001:db8:2:2::101	2001:db8:2:2::102	ICMPv6	Echo reply

▶ Frame 20 (138 bytes on wire, 138 bytes captured)
▶ Ethernet II, Src: HewlettP_db:48:c3 (00:1c:c4:db:48:c3), Dst: CadmusCo_07:3b:3d (08:00:27:07:3b:3d)
▶ Internet Protocol, Src: 192.168.21.204 (192.168.21.204), Dst: 192.168.21.100 (192.168.21.100)
▶ Internet Protocol Version 6
▶ Internet Control Message Protocol v6

Configuración túnel 6in4 (4)

- Scripts de eliminación de túneles 6in4
 - Windows con netsh desde ventana comandos

```
C:\> netsh interface ipv6 del route ::/0 "Tunel01"
Direccion_gateway_IPv6
C:\> netsh interface ipv6 del address "Tunel01"
Direccion_IPv6
C:\> netsh interface ipv6 del int "Tunel01"
```
 - Linux/UNIX (desde ventana de comandos)

```
# sudo ip route del 2000::/3 dev Tunel01
# sudo ip addr del Dir_IPv6_local/124 dev Tunel01
# sudo ip link set Tunel01 down
# sudo ip tunnel del Tunel01 mode sit remote
Dir_IPv4_remota local Dir_IPv4_local ttl 255
```