



# IPv6 – Servicios HTTP

## Objetivo

El objetivo de este laboratorio es presentar el funcionamiento en IPv6 del servidor HTTP Apache, uno de los más populares en la web. Esta práctica está dividida en dos experiencias: en la primera se muestra el funcionamiento básico de Apache2 en IPv6 y en la segunda se abordan las diferencias entre las configuraciones IPv4 e IPv6 de este mismo servidor.

Para la realización del presente ejercicio será utilizada la topología descrita en el archivo: **servicios-http1.imn**

## Introducción Teórica

Actualmente la Internet vive un momento de transición de IPv4 hacia IPv6 debido principalmente a la escasez de direcciones de esta primera versión del protocolo. Además de los cambios relacionados con las interfaces físicas de los dispositivos de red, es necesario realizar cambios en la forma de prestación de los servicios ya que deben ser adaptados para soportar nuevos formatos de direcciones y protocolos de comunicación específicos. En este sentido, y aunque existen otras técnicas, lo ideal es utilizar un equipo con doble pila que nos permita ejecutar los dos protocolos en forma simultánea.

En este laboratorio uno de los más populares servidores Web será utilizado para ejemplificar algunos de los cambios necesarios en la configuración del nuevo protocolo en este tipo de servidor. El servidor que utilizaremos, Apache2, es el servidor HTTP más utilizado en la web y soporta por defecto IPv6 desde su versión 2.0.

Aunque la configuración básica para habilitar el soporte IPv6 es en la mayoría de los casos bastante simple, es necesario prestar atención a las configuraciones más complejas, por ejemplo en los casos en que el servidor tiene VirtualServers ligados a direcciones IPv4 específicas, es necesario que la configuración se modifique de manera que también estén ligados a las direcciones IPv6 de los interfaces de red de la máquina.

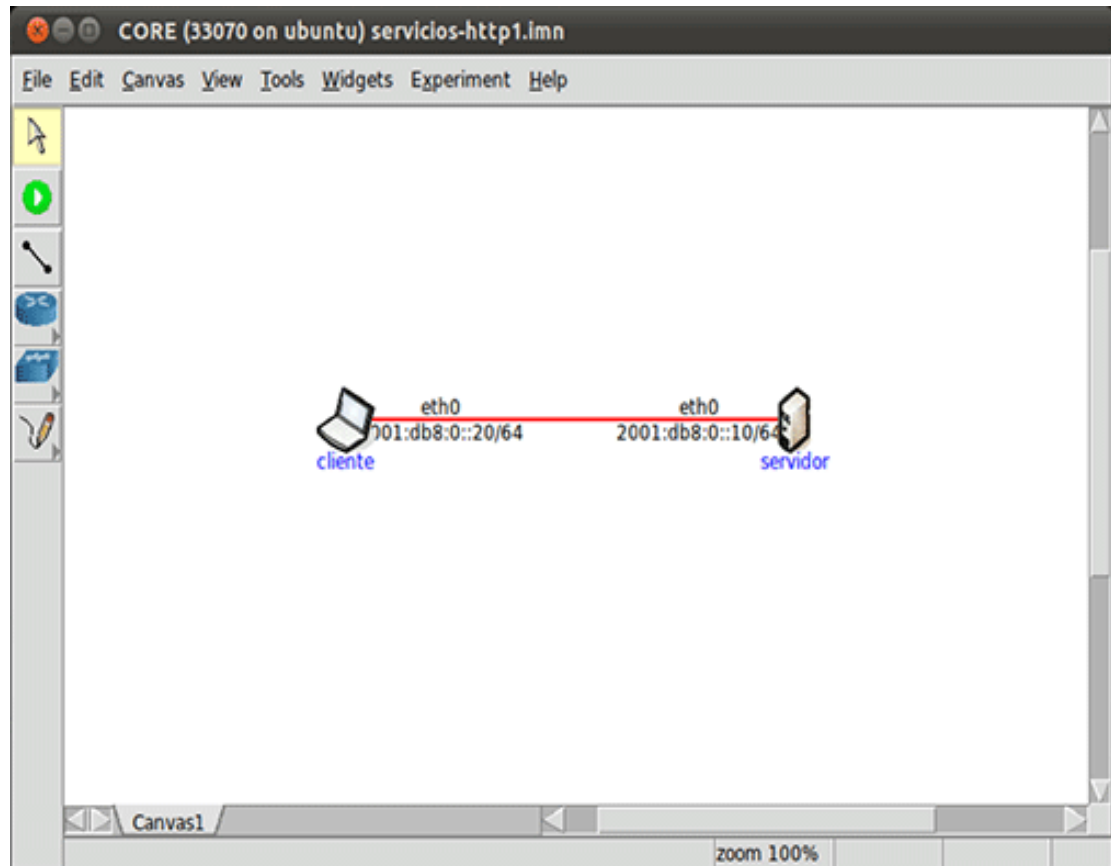
## Detalle del Experimento


### Funcionamiento Básico

1. En caso de que este utilizando la máquina virtual provista por LACNIC puede pasar directamente al paso nro. 2:
  - a. **Apache2** es un servidor HTTP bastante utilizado en la Web. Para instalarlo, alcanza con ejecutar el siguiente comando en una Terminal:

```
$ sudo apt-get install apache2
```

2. Inicie el CORE y abra el archivo “**servicios-http1.imn**” localizado en el directorio /home/core/Desktop/servicios/http, de la máquina virtual provista por LACNIC y la siguiente topología debe aparecer:



3. Analice la topología, observando las direcciones y comience el experimento:
  - Inicie la simulación realizando uno de los siguientes pasos:
    - Pulse el botón ; o
    - Utilice el menú Experiment > Start.
4. Inicie el servicio de apache2 en el ‘servidor’:
  - Abra una terminal en el ‘servidor’ con un doble-click.
  - Utilice el siguiente comando para iniciar el servidor apache2:

```
# /etc/init.d/apache2 start
```

El resultado debe ser:

```

CORE: servidor (console)
root@servidor:/tmp/pycore.55983/servidor.conf# /etc/init.d/apache2 start
* Starting web server apache2
apache2: apr_sockaddr_info_get() failed for servidor
apache2: Could not reliably determine the server's fully qualified domain name,
using 127.0.0.1 for ServerName
[ OK ]
root@servidor:/tmp/pycore.55983/servidor.conf# █

```

**\*OBS:** los errores mostrados ocurren por que los nombres de dominio y del servidor no fueron configurados para la maquina, a pesar de ello estos no afectan el funcionamiento del servicio para este experimento

5. Verifique que el servicio apache2 está activo y escuchando en el puerto 80 de las interfaces de red de la máquina 'servidor':

- Abra una terminal en el 'servidor' con un doble-click.
- Utilice el siguiente comando para verificar que el proceso apache2 está activo:

```
# ps aux
```

El resultado debe ser parecido al de la siguiente ventana:

```

CORE: servidor (console)
root@servidor:/tmp/pycore.55983/servidor.conf# ps aux
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root         1  0.0  0.1   2268    584 ?        S    03:24   0:00 /usr/sbin/vnode
root        29  0.0  0.2   5656    964 ?        Ss   03:24   0:00 /usr/sbin/sshd
root        30  0.0  0.3   3184   1660 pts/2    Ss+  03:28   0:00 /bin/bash
root        41  0.0  0.3   3184   1712 pts/7    Ss   03:36   0:00 /bin/bash
root        68  0.0  0.5   5784   2540 ?        Ss   03:37   0:02 /usr/sbin/apach
www-data    71  0.0  0.3   5556   1768 ?        S    03:37   0:00 /usr/sbin/apach
www-data    72  0.0  0.4  227188  2028 ?        Sl   03:37   0:00 /usr/sbin/apach
www-data    73  0.0  0.4  227188  2028 ?        Sl   03:37   0:00 /usr/sbin/apach
root       130  0.0  0.2   2528   1000 pts/7    R+   16:38   0:00 ps aux
root@servidor:/tmp/pycore.55983/servidor.conf# █

```

- Utilice el siguiente comando para verificar que el servicio apache2 está escuchando en el puerto 80 de todas sus interfaces de red:

```
# netstat -antup
```

El resultado debe ser:

```

CORE: servidor (console)
root@servidor:/tmp/pycore.55983/servidor.conf# netstat -antup
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
PID/Program name
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
29/sshd
tcp6       0      0 :::80                   :::*                    LISTEN
68/apache2
tcp6       0      0 :::22                   :::*                    LISTEN
29/sshd
root@servidor:/tmp/pycore.55983/servidor.conf#

```

**\*Obs:** El string “:::80” indica la utilización del puerto 80 de la dirección IPv6 [::/128] que es utilizado para mostrar que el servicio no está asociado a ninguna de las direcciones IPv6 del dispositivo, o sea que el servicio puede ser accedido en cualquiera de las direcciones de las interfaces de red de la máquina.

6. Verifique el funcionamiento del servidor HTTP a partir del ‘cliente’:

- Abra una terminal en el ‘cliente’ con un doble-click.
- Realice una solicitud HTTP GET al ‘servidor’:

```
# wget http://[2001:db8::10]/
```

El resultado debe ser:

```

CORE: cliente (console)
root@cliente:/tmp/pycore.55983/cliente.conf# wget http://[2001:db8::10]/
--2012-05-15 23:45:00-- http://[2001:db8::10]/
Connecting to 2001:db8::10:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 177 [text/html]
Saving to: `index.html'

100%[=====] 177      --.-K/s  in 0s

2012-05-15 23:45:00 (9.39 MB/s) - `index.html' saved [177/177]

root@cliente:/tmp/pycore.55983/cliente.conf#

```

- Vea que el archivo fue bajado correctamente:

```
# cat index.html
```

```

CORE: cliente (console)
root@cliente:/tmp/pycore.55983/cliente.conf# cat index.html
<html><body><h1>It works!</h1>
<p>This is the default web page for this server.</p>
<p>The web server software is running but no content has been added, yet.</p>
</body></html>
root@cliente:/tmp/pycore.55983/cliente.conf#

```

- Abra una terminal del servidor y verifique los logs generados por apache:



```
# cat /var/log/apache2/access.log
```

El resultado debe ser:

```
root@servidor:/tmp/pycore.55983/servidor.conf# cat /var/log/apache2/access.log
2001:db8::20 - - [14/May/2012:00:08:43 -0300] "GET / HTTP/1.0" 200 491 "-" "wget
/1.12 (linux-gnu)"
root@servidor:/tmp/pycore.55983/servidor.conf#
```

**\*Obs:** Note que la dirección registrada por el servidor Apache es una dirección IPv6. Eso es importante pues, en caso que el servidor utilice scripts u otras herramientas de análisis de logs, esas herramientas deberán ser capaces de analizar el formato de las direcciones IPv6.

7. Cierre la simulación del experimento:

- Ejecute el siguiente comando en una terminal del 'servidor' para detener el servicio de apache2:

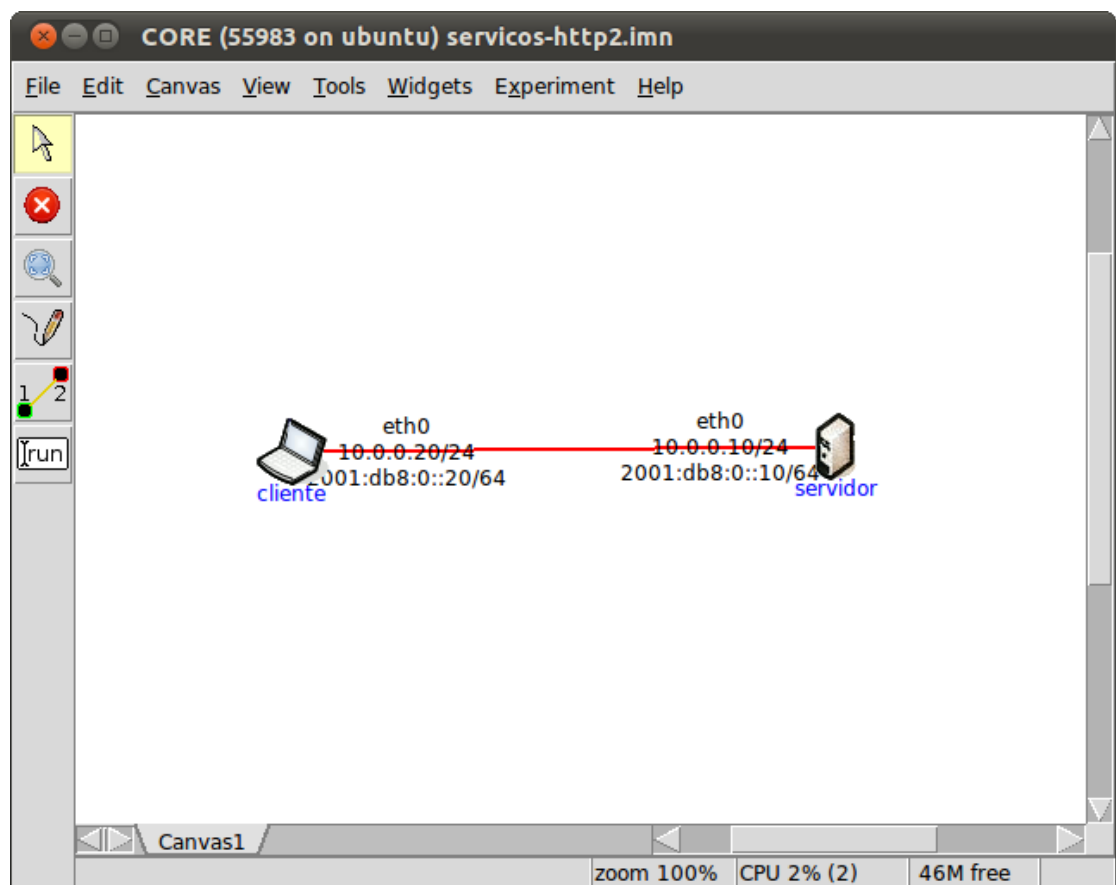
```
# /etc/init.d/apache2 stop
```


- Detenga la simulación de CORE :

- Pulse el botón ; o
- Utilice el menú Experiment > Stop.

## Experimento 4 – Configuración IPv6 de Apache

1. Esta segunda experiencia presenta algunas especificidades de la configuración de servidores Apache en redes IPv6. Para realizarla, también es necesaria la instalación del paquete “apache2” como fue indicado en el ítem 1.a. de la experiencia anterior en caso que no este usando la maquina virtual provista por LACNIC.
2. Inicie el CORE y abra el archivo “**servicios-http2.imn**” localizado en el directorio del desktop llamado “Servicios”, de la máquina virtual de LACNIC. La siguiente topología debe aparecer:



3. Verifique la configuración de los nodos de la topología:
  - Inicie la simulación ejecutando uno de los siguientes pasos:
    - Pulse el botón , o
    - Utilice el menú Experiment > Start.
4. Verifique que el servidor apache esta activo solamente vía IPv4:
  - Abra un terminal en el ‘cliente’ con un doble-click.



- Acceda al servidor HTTP ya configurado en la máquina 'servidor':

```
# wget 10.0.0.10
```

El resultado debe ser:

```
root@cliente:/tmp/pycore.55983/cliente.conf# wget http://10.0.0.10/
--2012-05-14 19:00:23-- http://10.0.0.10/
Connecting to 10.0.0.10:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 177 [text/html]
Saving to: `index.html'

100%[=====>] 177      --.-K/s   in 0s

2012-05-14 19:00:23 (26.5 MB/s) - `index.html' saved [177/177]

root@cliente:/tmp/pycore.55983/cliente.conf# █
```

**\*Obs:** Note que el archivo 'index.html' fue accedido correctamente.

- Ahora utilice la dirección IPv6:

```
# wget http://[2001:db8::10]/
```

El resultado debe ser:

```
root@cliente:/tmp/pycore.55983/cliente.conf# wget http://[2001:db8::10]/
--2012-05-14 20:21:09-- http://[2001:db8::10]/
Connecting to 2001:db8::10:80... failed: Connection refused.
root@cliente:/tmp/pycore.55983/cliente.conf# █
```

**\*Obs:** Note que el acceso fue denegado.

- Verifique ahora la configuración de apache en el 'servidor'. Abra una terminal en el servidor con un doble-click.
- Utilice el siguiente comando para verificar los servicios activos en la máquina:

```
# netstat -antup
```

El resultado debe ser:

```

CORE: servidor (console)
root@servidor:/tmp/pycore.55983/servidor.conf# netstat -antup
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
PID/Program name
tcp        0      0 10.0.0.10:80           0.0.0.0:*               LISTEN
43/apache2
tcp        0      0 0.0.0.0:22             0.0.0.0:*               LISTEN
104/sshd
tcp6       0      0 :::22                  :::*                     LISTEN
104/sshd
root@servidor:/tmp/pycore.55983/servidor.conf# █

```

**\*Obs:** Note que el puerto 80 del 'servidor' solo esta abierto para direcciones IPv4

5. En los siguientes pasos el servidor será configurado para escuchar también en el puerto 80 de su interface IPv6:

- En la terminal del servidor, abra el archivo “/etc/apache2/ports.conf”. Un editor de texto que puede ser utilizado es el nano. Para utilizarlo digite el siguiente comando en la terminal:

```
# nano /etc/apache2/ports.conf
```

- Ubique las siguientes líneas en el archivo:

```
NameVirtualHost 10.0.0.10:80
Listen 10.0.0.10:80
```

Note que estas líneas configuran el servidor apache para escuchar solamente en la interface IPv4. Por defecto el apache esta configurado para escuchar en todas las interfaces de red de la maquina donde esta instalado, esto se indica con las líneas de configuración (**Atencion! No cambie el archivo para incluir estas líneas**):

```
NameVirtualHost *:80
Listen 80
```

Sin embargo, la especificación de direcciones IP es bastante común en servidores de producción con mas de un sitio configurado. En esos casos, para que el sitio sea visto también en IPv6, las nuevas direcciones tienen que ser agregadas en la configuración.

- Entonces, debajo de la configuración de la dirección IPv4, **agregue** las lineas:

```
NameVirtualHost [2001:db8::10]:80
Listen [2001:db8::10]:80
```

**\*Obs:** Note que la dirección IPv6 debe estar entre paréntesis rectos. Utilice el comando ‘Ctrl+X’ seguido de ‘Y’ y luego ‘enter’ para salir y guardar las





modificaciones del archivo.

- Abra el archivo “/etc/apache2/sites-available/default” para modificar también la configuración del VirtualHost. Eso puede ser realizado con la utilización del editor de textos *nano*:

```
# nano /etc/apache2/sites-available/default
```

Modifique la línea:

```
<VirtualHost 10.0.0.10:80>
```

Con lo siguiente:

```
<VirtualHost 10.0.0.10:80 [2001:db8::10]:80>
```

Luego de esto guarde el archivo y cierre el editor de texto.

6. Reinicie el servicio apache2 ejecutando el siguiente comando para que el servidor se reinicie y toma la nueva configuración:

```
# /etc/init.d/apache2 restart
```

El resultado debe ser:

```
root@servidor:/tmp/pycore.55983/servidor.conf# /etc/init.d/apache2 restart
* Restarting web server apache2
apache2: apr_sockaddr_info_get() failed for servidor
apache2: Could not reliably determine the server's fully qualified domain name,
using 127.0.0.1 for ServerName
[Tue May 15 01:06:02 2012] [error] (EAI 2)Name or service not known: Failed to r
esolve server name for 10.0.0.10 (check DNS) -- or specify an explicit ServerNam
e
... waiting apache2: apr_sockaddr_info_get() failed for servidor
apache2: Could not reliably determine the server's fully qualified domain name,
using 127.0.0.1 for ServerName
[Tue May 15 01:06:03 2012] [error] (EAI 2)Name or service not known: Failed to r
esolve server name for 10.0.0.10 (check DNS) -- or specify an explicit ServerNam
e
[ OK ]
root@servidor:/tmp/pycore.55983/servidor.conf#
```

**\*OBS:** Los errores mostrados ocurren por que los nombres de dominio y del servidor no fueron configurados para la maquina, a pesar de ello estos no afectan el funcionamiento del servicio para este experimento.

- Ejecute nuevamente el comando *netstat*, tal cual lo indicado a continuación, para verificar que el servidor *apache2* está escuchando en el puerto 80 tanto en su dirección IPv4 como IPv6:

```
# netstat -antup
```

El resultado debe ser:

```

CORE: servidor (console)
root@servidor:/tmp/pycore.55983/servidor.conf# netstat -antup
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
PID/Program name
tcp        0      0 10.0.0.10:80           0.0.0.0:*               LISTEN
264/apache2
tcp        0      0 0.0.0.0:22             0.0.0.0:*               LISTEN
104/sshd
tcp6       0      0 2001:db8::10:80       :::*                    LISTEN
264/apache2
tcp6       0      0 :::22                  :::*                    LISTEN
104/sshd
root@servidor:/tmp/pycore.55983/servidor.conf# █

```

- Para confirmar el funcionamiento del servidor HTTP, abra una terminal en el 'cliente' y utilice los siguientes comandos:

```

# wget http://10.0.0.10/
# wget http://[2001:db8::10]/

```

El resultado debe ser:

```

CORE: cliente (console)
root@cliente:/tmp/pycore.55983/cliente.conf# wget http://10.0.0.10/
--2012-05-15 01:53:07-- http://10.0.0.10/
Connecting to 10.0.0.10:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 177 [text/html]
Saving to: `index.html.1'

100%[=====>] 177      --.-K/s  in 0s

2012-05-15 01:53:07 (9,52 MB/s) - `index.html.1' saved [177/177]

root@cliente:/tmp/pycore.55983/cliente.conf# wget http://[2001:db8::10]/
--2012-05-15 01:53:12-- http://[2001:db8::10]/
Connecting to 2001:db8::10:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 177 [text/html]
Saving to: `index.html.2'

100%[=====>] 177      --.-K/s  in 0s

2012-05-15 01:53:12 (11,3 MB/s) - `index.html.2' saved [177/177]

root@cliente:/tmp/pycore.55983/cliente.conf# █

```


7. Cierre la simulación del experimento:
  - Ejecute el siguiente comando en una Terminal del 'servidor' para detener la ejecución del servicio apache2:

```

# /etc/init.d/apache2 stop

```



- Detenga la simulación de CORE :
- Pulse el botón ; o
- Utilice el menú Experiment > Stop.