

CURSO:

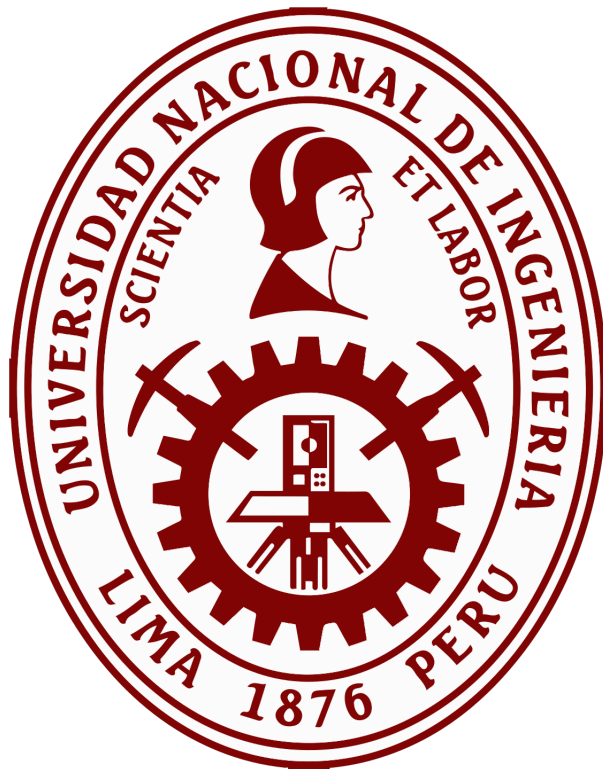
ADMINISTRACIÓN DE REDES

ALUMNO:

CARLOS RENATO CHAUCA PRINQUE

TAREA:

LABORATORIO 7.2: DDNS EN SUBREDES



ÍNDICE

1) Introducción.....	página 3
2) Configuración de los clientes.....	página 4
3) Configuración del servidor.....	página 5
4) Pruebas.....	página 10
5) Link del video.....	página 12
6) Referencias.....	página 12

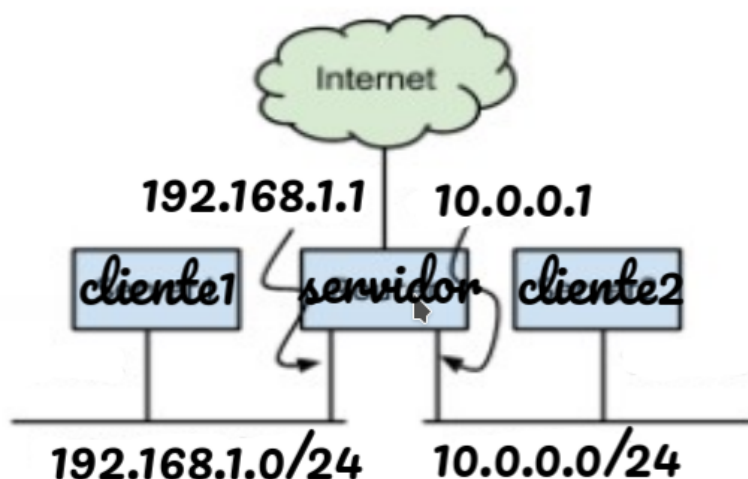
1) INTRODUCCIÓN:

Hasta el momento hemos trabajado en los laboratorios anteriores con un servidor DHCP y un servidor DNS, ahora explicaremos de manera resumida que viene a ser un servidor DDNS.

SERVIDOR DDNS:

En palabras sencillas, el servidor DDNS viene a ser la union de un DHCP y un DNS. Al trabajar con un servidor DNS se configuraba de tal modo que las IP's de los clientes fuesen establecidas de manera fija o estática, esto representa un problema cuando se tratan de clientes los cuales reciben IP's de manera dinámica, puesto que bien podrían tener una IP ahora y luego podrían poseer otra si éstas recibieran una IP de un servidor DHCP, ahí es donde entra el servidor DDNS, el cual solo vendría a ser un servidor DNS dinámico indicando que las IP's se asignarán de manera dinámica y así poder asociar correctamente el nombre del cliente y la IP con la que éste está trabajando de momento.

Nuestra arquitectura debería ser de acuerdo a la siguiente imagen:



Se entiende que el servidor poseerá 2 interfaces además de la NAT, las cuales serán `enp0s8` (para la subred1) y `enp0s9` (para la subred2), ignorando el mal dibujo se observa que los 2 clientes no poseen una IP dado que la recibirán de manera dinámica por el servidor.

2) CONFIGURACIÓN DE LOS CLIENTES:

Su configuración es realmente sencilla ya que el servidor DDNS se encargará prácticamente de todo.

Se modificará el fichero `/etc/hostname` con el nombre propio de cada cliente:

```
adr@cliente1:~$ cat /etc/hostname
cliente1
```

```
adr@cliente2:~$ cat /etc/hostname
cliente2
```

Los ficheros de interfaces de red deben poseer una asignación dhcp en ambos.

Configuración dhcp del cliente1

```
adr@cliente1:~$ cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto enp0s3
iface enp0s3 inet dhcp
```

Configuración dhcp del cliente2

```
adr@cliente2:~$ cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto enp0s3
iface enp0s3 inet dhcp
```

Por último se configurará (luego de reiniciar el servicio de red `'/etc/init.d/networking restart'`) el fichero `resolv.conf`.

Nombre de dominio que buscará el cliente1

```
adr@cliente1:~$ cat /etc/resolv.conf
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver 192.168.1.1
search fc.uni.edu
```

Nombre de dominio que buscará el cliente2

```
adr@cliente2:~$ cat /etc/resolv.conf
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver 10.0.0.1
search fic.uni.edu
adr@cliente2:~$
```

Los nombres de dominio del fichero `resolv.conf` dependerá con que nombre de dominio estará trabajando el alumno, en mi caso trabajo con `'fc.uni.edu'` para la subred 1 y `'fic.uni.edu'` para la subred 2 respectivamente.

3) CONFIGURACIÓN DEL SERVIDOR:

Antes de comenzar con todas las configuraciones se deben instalar los servicios con los que vamos a trabajar, bind9 para DNS y isc-dhcp-server para DHCP.

```
sudo apt-get install bind9 isc-dhcp-server
```

3.1) Configuración de las interfaces:

Se deben asignar 2 IP's de manera estática para identificar a las 2 subredes con las que se trabajarán.

```
dR: ~
adr@AdR:~$ cat /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto enp0s3
iface enp0s3 inet dhcp

auto enp0s8
iface enp0s8 inet static
address      192.168.1.1
network      192.168.1.0
netmask      255.255.255.0
broadcast    192.168.1.255

auto enp0s9
iface enp0s9 inet static
address      10.0.0.1
network      10.0.0.0
netmask      255.255.255.0
broadcast    10.0.0.255
```

3.2) Configuración del servicio DHCP:

Este servicio se configura en el fichero '/etc/dhcp/dhcpd.conf'.

Configuración del servicio dhcp para la subred 1

```
subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.50 192.168.1.100;
    option domain-name "fc.uni.edu";
    option domain-name-servers 192.168.1.1;
    option routers 192.168.1.1;
    ddns-domainname "fc.uni.edu";
}

zone fc.uni.edu {
    primary 192.168.1.1;
}

zone 1.168.192.in-addr.arpa {
    primary 192.168.1.1;
}
```

Configuración del servicio dhcp para la subred 2

```
subnet 10.0.0.0 netmask 255.255.255.0 {  
    range 10.0.0.50 10.0.0.100;  
    option domain-name "fic.uni.edu";  
    option domain-name-servers 10.0.0.1;  
    option routers 10.0.0.1;  
    ddns-domainname "fic.uni.edu";  
}  
  
zone fic.uni.edu {  
    primary 10.0.0.1;  
}  
  
zone 0.0.10.in-addr.arpa {  
    primary 10.0.0.1;  
}
```

Se establecerá a cada zona el argumento 'primary', cuya referencia especifica la dirección IP del servidor cuya información de zonas se actualizará (es decir, no trabajará de manera estática).

Argumentos para el funcionamiento del servicio DDNS

```
ddns-updates on;  
ddns-update-style interim;  
ignore client-updates;
```

La línea ddns-updates on activa las actualizaciones dinámicas.

La línea ddns-update-style indica el tipo de actualización dinámica.

Todo esto deberá estar incluido dentro del fichero de configuración '/etc/dhcp/dhcpd.conf'.

Se podrá comprobar si todo esta correcto con el comando 'dhcpd -t'

3.3) Configuración de las zonas de dominio:

Se establecerán 2 zonas directas (para fc.uni.edu y fic.uni.edu) y 2 inversas de las anteriores ya mencionadas.

```
//zona directa del nombre de dominio fc.uni.edu
zone "fc.uni.edu" {
    type master;
    file "/etc/bind/db.fc.uni.edu";
    journal "directa1.jnl";
    allow-update {192.168.1.1;};
};

//zona inversa del nombre de dominio fc.uni.edu
zone "1.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/db.1.168.192.revzone";
    journal "inversa1.jnl";
    allow-update {192.168.1.1;};
};

//zona directa del nombre de dominio fic.uni.edu
zone "fic.uni.edu" {
    type master;
    file "/etc/bind/db.fic.uni.edu";
    journal "directa2.jnl";
    allow-update {10.0.0.1;};
};

//zona inversa del nombre de dominio fic.uni.edu
zone "0.0.10.in-addr.arpa" {
    type master;
    file "/etc/bind/db.0.0.10.revzone";
    journal "inversa2.jnl";
    allow-update {10.0.0.1;};
};
```

Observamos que a cada zona se han añadido 2 líneas comparadas al laboratorio 7.1 de DNS, en este caso se explicará para que sirve cada una de éstas.

- 'journal <name>.jnl;' : éste archivo de extensión .jnl se usa para que el servidor DDNS escriba los cambios respectivos en el archivo de zona cuando un cliente adquiere una nueva dirección IP.
- 'allow-update { IP; }' : Ingresa la IP del servidor en el que se permiten las actualizaciones dinámicas.

Obs: Se omitirán los pasos de chequeo de zona ya que se han visto en el laboratorio anterior, se podrán comprobar aun así si se desea con el comando 'sudo named-checkconf'.

3.4) Configuración de las zonas directas e inversas:

Se tomarán como archivos de referencia los que ya hemos visto en el laboratorio anterior, los cuales son /etc/bind/db.local para las zonas directas y /etc/bind/db.127 para las zonas inversas.

La configuración debería ser como se muestra en las siguientes imágenes:

Configuración de la zona directa e inversa del dominio fc.uni.edu

```
AdR: ~
adr@AdR:~$ cat /etc/bind/db.fc.uni.edu
;
; BIND data file for local loopback interface
;
$TTL      604800
@          IN      SOA      servidor.fc.uni.edu. root.fc.uni.edu. (
                        2          ; Serial
                        604800     ; Refresh
                        86400      ; Retry
                        2419200    ; Expire
                        604800 )   ; Negative Cache TTL
;
;
@          IN      NS       servidor.
localhost  IN      A        127.0.0.1
servidor   IN      A        192.168.1.1
adr@AdR:~$ cat /etc/bind/db.1.168.192.revzone
;
; BIND reverse data file for local loopback interface
;
$TTL      604800
@          IN      SOA      servidor.fc.uni.edu. root.fc.uni.edu. (
                        1          ; Serial
                        604800     ; Refresh
                        86400      ; Retry
                        2419200    ; Expire
                        604800 )   ; Negative Cache TTL
;
;
@          IN      NS       servidor.
1          IN      PTR      servidor
```

Configuración de la zona directa e inversa del dominio fic.uni.edu

```
AdR: ~
adr@AdR:~$ cat /etc/bind/db.fic.uni.edu
;
; BIND data file for local loopback interface
;
$TTL      604800
@          IN      SOA      servidor.fic.uni.edu. root.fic.uni.edu.
                        2          ; Serial
                        604800     ; Refresh
                        86400      ; Retry
                        2419200    ; Expire
                        604800 )   ; Negative Cache TTL
;
;
@          IN      NS       servidor.
localhost  IN      A        127.0.0.1
servidor   IN      A        10.0.0.1
adr@AdR:~$ cat /etc/bind/db.0.0.10.revzone
;
; BIND reverse data file for local loopback interface
;
$TTL      604800
@          IN      SOA      servidor.fic.uni.edu. root.fic.uni.edu.
                        1          ; Serial
                        604800     ; Refresh
                        86400      ; Retry
                        2419200    ; Expire
                        604800 )   ; Negative Cache TTL
;
;
@          IN      NS       servidor.
1          IN      PTR      servidor
```


Se podrá comprobar si los ficheros fueron sobrescritos correctamente con los siguientes comandos:

- `sudo named-checkzone fc.uni.edu /etc/bind/db.fc.uni.edu` (para la zona directa de fc.uni.edu)
- `sudo named-checkzone 1.168.192 /etc/bind/db.1.168.192.revzone` (para la zona inversa de fc.uni.edu)
- `sudo named-checkzone fic.uni.edu /etc/bind/db.fic.uni.edu` (para la zona directa de fic.uni.edu)
- `sudo named-checkzone 0.0.10 /etc/bind/db.0.0.10.revzone` (para la zona inversa de fic.uni.edu)

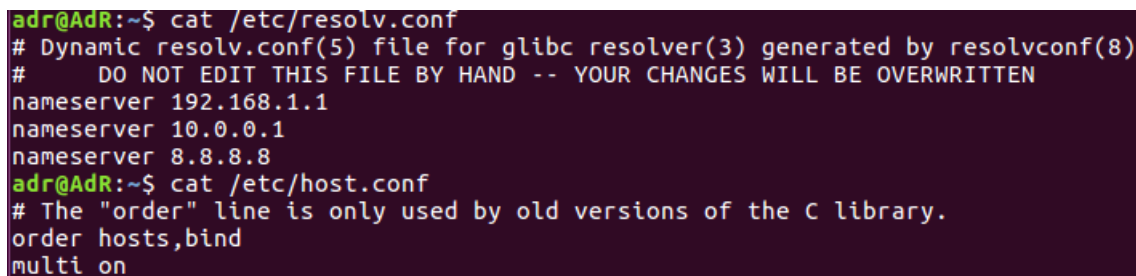
Ahora si podemos reiniciar tanto el servicio DHCP como el servicio DNS

```
sudo /etc/init.d/isc-dhcp-server restart
```

```
sudo /etc/init.d/bind restart
```

3.5) Configuración del resolv.conf:

Por ultimo es necesario añadir al fichero resolv.conf el nameserver con el que se trabajará, como trabajaremos con 2 subredes yo añadiré ambas así al momento de buscar en si mismo buscará primeramente a los nombres de dominio con los que hemos trabajado, por ultimo añadiré los DNS públicos de google.



```
adr@AdR:~$ cat /etc/resolv.conf
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver 192.168.1.1
nameserver 10.0.0.1
nameserver 8.8.8.8
adr@AdR:~$ cat /etc/host.conf
# The "order" line is only used by old versions of the C library.
order hosts,bind
multi on
```

añadiendo a esto tiene que quedar claro que en el fichero '/etc/host.conf' no se modificará, ya que trabajara primeramente buscando una respuesta de los hosts (donde tanto en los clientes como en el servidor no debe haber nada configurado en el fichero '/etc/hosts'), luego del servicio bind a donde es que queremos que busque, caso contrario no tendría sentido trabajar con el servicio bind que es el propósito del laboratorio.

Una vez hecho esto debemos verificar que exista conexión entre los 2 clientes, si no es así será necesario activar el forwarding con el comando: 'sudo sysctl net.ipv4.conf.all.forwarding=1' y enrutar manualmente los 2 clientes con el comando 'route add -net <red de destino> netmask 255.255.255.0 gw <ip de origen>'.
Ahora si pasemos a las pruebas.

4) PRUEBAS:

Asignación de IP's de manera dinámica a los 2 clientes:

Se estableció un rango de IP's entre 192.168.1.50:100 para la subred 1

```
adr@cliente1:~$ ifconfig
enp0s3  Link encap:Ethernet  direcciónHW 08:00:27:6a:4c:aa
        Direc. inet:192.168.1.50  Difus.:192.168.1.255  Másc:255.255.255.
        Dirección inet6: fe80::a00:27ff:fe6a:4caa/64 Alcance:Enlace
        ACTIVO DIFUSIÓN MULTICAST  MTU:1500  Métrica:1
        Paquetes RX:126 errores:0 perdidos:0 overruns:0 frame:0
        Paquetes TX:387 errores:0 perdidos:0 overruns:0 carrier:0
        colisiones:0 long.colaTX:1000
        Bytes RX:16316 (16.3 KB)  TX bytes:42867 (42.8 KB)
```

Se estableció un rango de IP's entre 10.0.0.50:100 para la subred 2

```
adr@cliente2:~$ ifconfig
enp0s3  Link encap:Ethernet  direcciónHW 08:00:27:b7:4f:39
        Direc. inet:10.0.0.50  Difus.:10.0.0.255  Másc:255.255.255.0
        Dirección inet6: fe80::a00:27ff:feb7:4f39/64 Alcance:Enlace
        ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST  MTU:1500  Métrica:1
        Paquetes RX:44 errores:0 perdidos:0 overruns:0 frame:0
        Paquetes TX:294 errores:0 perdidos:0 overruns:0 carrier:0
        colisiones:0 long.colaTX:1000
        Bytes RX:6348 (6.3 KB)  TX bytes:48516 (48.5 KB)
```

Verificamos que hay conexión entre los 2 clientes de diferentes subredes con un ping.

```
adr@cliente1:~$ ping 10.0.0.50
PING 10.0.0.50 (10.0.0.50) 56(84) bytes of data.
64 bytes from 10.0.0.50: icmp_seq=1 ttl=63 time=111 ms
64 bytes from 10.0.0.50: icmp_seq=2 ttl=63 time=1.98 ms
64 bytes from 10.0.0.50: icmp_seq=3 ttl=63 time=2.03 ms
64 bytes from 10.0.0.50: icmp_seq=4 ttl=63 time=2.03 ms
^C
--- 10.0.0.50 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3007ms
rtt min/avg/max/mdev = 1.982/29.484/111.885/47.574 ms
adr@cliente1:~$ ping 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=1.02 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.867 ms
^C
--- 10.0.0.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 0.867/0.945/1.023/0.078 ms
```

```
adr@cliente2:~$ ping 192.168.1.50
PING 192.168.1.50 (192.168.1.50) 56(84) bytes of data.
64 bytes from 192.168.1.50: icmp_seq=1 ttl=63 time=3.20 ms
64 bytes from 192.168.1.50: icmp_seq=2 ttl=63 time=1.96 ms
64 bytes from 192.168.1.50: icmp_seq=3 ttl=63 time=1.90 ms
^C
--- 192.168.1.50 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2008ms
rtt min/avg/max/mdev = 1.907/2.358/3.206/0.600 ms
adr@cliente2:~$ ping 10.0.0.1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=1.11 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=1.11 ms
^C
--- 10.0.0.1 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1003ms
rtt min/avg/max/mdev = 1.112/1.114/1.117/0.033 ms
adr@cliente2:~$
```

Por último ejecutamos el comando nslookup desde las 3 máquinas virtuales para ver que el servicio DDNS se está ejecutando correctamente.

Ejecución del comando nslookup en el servidor

```
adr@AdR:~$ nslookup 10.0.0.50
Server:      192.168.1.1
Address:     192.168.1.1#53

50.0.0.10.in-addr.arpa  name = cliente2.fic.uni.edu.

adr@AdR:~$ nslookup cliente1.fic.uni.edu
Server:      192.168.1.1
Address:     192.168.1.1#53

Name:   cliente1.fic.uni.edu
Address: 192.168.1.50

adr@AdR:~$ nslookup servidor.fic.uni.edu
Server:      192.168.1.1
Address:     192.168.1.1#53

Name:   servidor.fic.uni.edu
Address: 192.168.1.1

adr@AdR:~$ nslookup 10.0.0.1
Server:      192.168.1.1
Address:     192.168.1.1#53

1.0.0.10.in-addr.arpa  name = servidor.0.0.10.in-addr.arpa.
```

Ejecución del comando nslookup en el cliente1

```
adr@cliente1:~$ nslookup cliente2.fic.uni.edu
Server:      192.168.1.1
Address:     192.168.1.1#53

Name:   cliente2.fic.uni.edu
Address: 10.0.0.50

adr@cliente1:~$ nslookup 10.0.0.50
Server:      192.168.1.1
Address:     192.168.1.1#53

50.0.0.10.in-addr.arpa  name = cliente2.fic.uni.edu.

adr@cliente1:~$ nslookup servidor.fic.uni.edu
Server:      192.168.1.1
Address:     192.168.1.1#53

Name:   servidor.fic.uni.edu
Address: 10.0.0.1

adr@cliente1:~$ nslookup 192.168.1.1
Server:      192.168.1.1
Address:     192.168.1.1#53

1.1.168.192.in-addr.arpa  name = servidor.1.168.192.in-addr.arpa.

adr@cliente1:~$
```

Ejecución del comando nslookup en el cliente2

```
adr@cliente2:~$ nslookup cliente1.fic.uni.edu
Server:      10.0.0.1
Address:     10.0.0.1#53

Name:   cliente1.fic.uni.edu
Address: 192.168.1.50

adr@cliente2:~$ nslookup 192.168.1.50
Server:      10.0.0.1
Address:     10.0.0.1#53

50.1.168.192.in-addr.arpa  name = cliente1.fic.uni.edu.

adr@cliente2:~$ nslookup servidor.fic.uni.edu
Server:      10.0.0.1
Address:     10.0.0.1#53

Name:   servidor.fic.uni.edu
Address: 10.0.0.1

adr@cliente2:~$ nslookup servidor.fic.uni.edu
Server:      10.0.0.1
Address:     10.0.0.1#53

Name:   servidor.fic.uni.edu
Address: 192.168.1.1
```

5) LINK DEL VIDEO:

https://drive.google.com/open?id=1Z44acudpKXpAQ0fKaUHutHNTg3nW_EoQ

6) REFERENCIAS:

<http://manpages.ubuntu.com/manpages/bionic/man1/dyndns.1.html>

<https://es.slideshare.net/DiegoMontiel7/servidor-dns-dinamico-en-ubuntu-server-1604-1>

<http://techadminblog.com/setup-linux-dynamic-dns-server-2/>

<https://www.dnsknowledge.com/tutorials/centos-tutorials/bind-9/howto-setup-dynamic-dn>