Introducción a conceptos de redes en GNU/Linux

Margarita Manterola y Maximiliano Curia

Junio 2003

1. Introducción

Todos los días, cuando usamos internet, nos comunicamos a diversos puntos del planeta. Cada vez que leemos nuestro correo, navegamos en la web o entramos en un chat estamos utilizando un número inmenso de computadoras, todas interconectadas entre sí.

Todo esto lo hacemos usando un protocolo de comunicación entre partes de la red. Se denomina **protocolo** a cualquier lenguaje que puede ser utilizado por computadoras, switchs, routers y demás elementos de red para comunicarse entre sí.

El protocolo que se utiliza más comúnmente en Internet es el TCP/IP que, en realidad, es la suma de dos protocolos distintos (TCP e IP).

TCP/IP forman un protocolo tan flexible que permiten la comunicación de una multitud de plataformas y una enorme cantidad de computadoras. Millones de usuarios de todo el planeta acceden cada día a los servicios de red que proveen estos protocolos.

2. Conceptos básicos de TCP/IP

Hablaremos a continuación de los conceptos relacionados exclusivamente con el protocolo IP, y en gran parte con el protocolo TCP. Estos conceptos son generales, más allá del sistema operativo. Además, nos pueden ser útiles para poder comprender el funcionamiento de otros protocolos.

En particular, hablaremos del la versión del protocolo IP llamada *ipv4*, que es la versión que se utiliza actualmente en Internet. Existe un prototipo para constituir una nueva versión (la ipv6), pero todavía se encuentra en experimentación.

Por otro lado, el protocolo TCP utiliza al protocolo IP como base y se usa para una gran variedad de servicios de red (web, mail, ftp, etc). Otros protocolos, similares al TCP son el UDP (utilizado para consultas a DNS, por ejemplo) y el ICMP (utilizado para hacer `ping`).

2.1. Dirección IP

Toda máquina que está en una red de computadoras debe identificarse de alguna manera que la haga distinguible de las otras computadoras. Esta identificación se hace mediante un número. Y dependerá de los valores correspondientes a la red a la que pertenece.

El número, llamado **dirección IP** (Internet Protocol) - o a veces simplemente **IP**, está compuesto por cuatro bytes. Generalmente lo representamos por cuatro números (menores a 256) separados por puntos. Por ejemplo: **200.45.34.59**.

De los 4,29 giga de direcciones disponibles con este formato, existen unas cuantas que se encuentran destinadas para usos particulares o privados.

- Las dirección 127.0.0.1 se encuentra reservada para identificar a la computadora local. Por añadidura, todo el rango de IPs que comienza con 127 también identifica a la computadora local.
- El rango de direcciones 192.168.0.0 - 192.168.255.255 se encuentra reservado para utilizarlo en redes privadas menores a 255 computadoras.
- El rango 172.16.0.0 - 172.16.255.255 se utiliza para redes privadas compuestas por un número menor a 65500 computadoras.
- El rango 10.0.0.0 - 10.255.255.255 se encuentra reservado para utilizarlo en redes privadas de hasta 16 millones de computadoras
- Existen muchos otros rangos que también están reservados para usos especiales, de manera que el gigantesco número de IPs disponibles se ve empequeñecido.

Llamamos redes **privadas** a aquellas cuyas direcciones IP no son válidas para utilizar en Internet. Estas dirección son válidas únicamente dentro de la red a la que esté conectada la computadora.

Mientras que la red **pública** es Internet, y las direcciones IPs **públicas** son todas las direcciones que son válidas para usar en Internet.

En los orígenes de Internet, había una sobreabundancia de direcciones IP, de modo que grandes cantidades de estas direcciones fueron otorgadas a algunas entidades (como ciertas universidades, o empresas que aportaron al desarrollo de Internet). En la actualidad, el número de direcciones IP disponibles para ser utilizadas es muy pequeño, de modo que se han convertido en un recurso muy preciado y, por lo tanto, muy caro.

2.2. Hosts

Además de tener una dirección IP que las identifica, las computadoras suelen tener algún nombre un poco más amigable, que permite identificarlas dentro del lenguaje humano.

En el archivo **/etc/hosts**, podemos asignar, a cada dirección IP, uno o varios nombres. La sintaxis es muy sencilla, por ejemplo:

```
192.168.21.10    amadeus.amadeus  amadeus
192.168.21.1     erwin.amadeus    erwin
192.168.21.7     peperina.amadeus peperina
192.168.21.13    freak.amadeus    freak
```

2.3. Subredes

Acompañando la dirección IP habrá también una *máscara de subred* (netmask), que nos indica qué parte de la dirección es interna de la red a la que está conectada la computadora, y qué parte representa a la red en sí.

Por ejemplo, si la dirección IP es 192.168.35.65 y la máscara de subred es 255.255.255.0, nuestra computadora se identifica con el número 65 dentro de nuestra red, que a su vez utiliza el rango 192.168.35.0 al 192.168.35.255. Este rango también se puede escribir 192.168.35.0/24.

La denominación de *máscara* indica justamente esta capacidad de separar lo que pertenece a la red y lo que es específico de la computadora local. Es decir, escribiendo 255 en binario tendremos 11111111 (8 unos). Y si escribimos 255 tres veces, tendremos 24 unos. Esto indica que los primeros 24 bits de nuestra dirección IP identifican a la red y los últimos 8 a nuestra máquina dentro de la red.

Otro ejemplo: la dirección IP 10.1.130.210 podría tener la máscara 255.0.0.0, que nos indica que nuestra red es 10.0.0.0/8 y el resto de los números son la identificación de esa computadora dentro de su red.

Las redes suelen clasificarse según el rango de IPs válido que posean. Se agrupan en *Clases*

Clase C serán las redes que tengan una máscara 255.255.255.0, es decir, las redes que tengan un máximo de 255 computadoras.

Clase B serán las redes que tengan la máscara 255.255.0.0.

Clase A serán las redes que tengan la máscara 255.0.0.0.

2.4. Ruteos

Un grupo de computadoras que pertenezcan a una misma red pueden comunicarse entre sí directamente. Lo usual será que se encuentren conectadas a través de un hub o switch. O también, puede darse la situación de que tengamos dos computadoras conectadas entre sí con un cable de una a otra.

Pero cuando queremos interconectar redes debemos poseer un equipo que se encargue de rutear la información de una red a la otra. Se denomina **ruta** de un determinado paquete de información al destino que debe asignarse a ese paquete según una serie de criterios (la dirección de origen, el puerto de origen, la dirección de destino, el puerto de destino, el protocolo utilizado (TCP, UDP, ICMP), etc). Esto es lo que sucede en el caso de Internet.

Por ejemplo, tomemos el caso de un usuario que desea acceder, a través de Internet, al sitio www.google.com. Para ello, su computadora se comunicará con la del ISP (Internet Service Provider), que a su vez se comunicará con otra computadora, ubicada en el exterior, que por su parte se comunicará con una serie de equipos hasta llegar a la computadora que corresponde a google.

Para ver el trayecto que realizan los paquetes de información al transmitirse por la red, podemos utilizar el comando `traceroute`. Por ejemplo, para ver el caso de google que mencionamos, debemos ejecutar
`traceroute www.google.com`.

Para poder interconectar redes, necesitamos tener al menos un equipo que exista en las dos redes (puede ser una computadora común, o también un equipo especialmente armado para esto, llamado **router**). Este equipo será el que actuará de nexo entre ambas. En ese equipo existirá una configuración de las rutas que deben seguir los paquetes de información dentro de cada red. A este equipo lo llamamos **gateway** (puerta de enlace).

Las estaciones que pertenezcan a las diversas redes deben tener configurado cuál es el gateway correspondiente para acceder a la otra red.

Tanto para realizar la configuración como para ver el estado actual de las tablas de ruteo, en GNU/Linux utilizamos el comando `route`. Si lo ejecutamos sin parámetros nos muestra la tabla de ruteo actual. Veremos la misma tabla, pero sin resolución de nombres, si ejecutamos `route -n`.

La ruta con el nombre de *default*, será la ruta por la cual se dirigirán los paquetes cuando no coincidan con ninguna otra ruta. Es por esta ruta por la que normalmente se realizarán nuestras conexiones a Internet.

2.5. Firewall

Así como existe un equipo que se encarga de que los paquetes pasen de una red a otra, puede existir un equipo que se encarga de que los paquetes **no** pasen de una red a otra. Se trata de un **firewall**, cuya función principal es proteger a las computadoras de posibles ataques externos.

Además, en las empresas se suele utilizar estos equipos para impedir que los empleados accedan a un determinado servicio (ICQ, MSN, etc).

Muchas veces, cuando una computadora con GNU/Linux es el gateway de una red, esa misma computadora funciona como firewall. Esto se debe a que cualquier sistema GNU/Linux posee ambas funcionalidades integradas directamente en el kernel del sistema, y la configuración es muy estable y relativamente sencilla.

2.6. Proxy

Se denomina **proxy** a una computadora que almacena las páginas más visitadas en una memoria caché, de manera que se haga más rápido el acceso a esas páginas.

Según la configuración de la red, puede suceder que la única forma de acceder a Internet sea a través del proxy. Esto se puede dar en dos situaciones:

- cuando no hay ningún gateway que interconecte la red interna con la externa, y el proxy funciona como el único gateway existente;
- cuando existe un gateway, pero también un firewall que bloquea la utilización de la red externa para un determinado grupo de computadoras, pero no para el proxy.

Sin embargo, esto no tiene porqué ser así. Aún si el firewall permite la utilización de la red externa, es posible elegir utilizar el proxy para acceder a Internet, ya que al tener las páginas previamente cargadas en memoria, se hace mucho más rápida la navegación.

Este fue el objetivo de la creación de los proxies, aunque en la actualidad, muchas veces se los utiliza como gateway, impidiendo de esta manera la utilización de **toda** Internet, permitiendo el acceso únicamente al servicio web.

3. Servicios

Existen muchos servicios que se utilizan a través de redes. Algunos de los más difundidos son el acceso web y el e-mail. Pero existen, literalmente, miles de servicios que podríamos utilizar dentro de una red.

Muchos de estos servicios están especificados en el archivo **/etc/services**. Este archivo define un nombre para conexiones de red en determinado puerto, con determinado protocolo. Los protocolos, por otro lado, se definen en el archivo **/etc/protocols**.

Para ver en qué puertos está escuchando la computadora que estemos utilizando en este momento, se puede utilizar el comando `netstat -nl`. (La n es para que no resuelva los nombres, y la l para que muestre los puertos de escucha (listening)).

Otro comando útil para saber en qué puertos correspondientes a qué servicios están habilitados es `nmap localhost`. Además de **localhost** (que es el nombre que identifica a nuestra computadora), podemos utilizar la dirección IP, o el nombre correspondiente, de cualquier máquina para saber qué puertos tiene abiertos esa computadora y a qué servicios corresponden.

3.1. DNS

El servicio de DNS (Domain Name Server), es el que nos permite traducir nombres (por ejemplo: www.gnu.org) a direcciones IP (199.232.41.10, en este caso).

Nació por la necesidad de poder expandir y centralizar el archivo **/etc/hosts** que se encuentra en cada una de las computadoras. En un principio, este archivo había que actualizarlo cada vez que una computadora se agregaba a la red. Esto implicaba un importante trabajo de mantenimiento para tener actualizado el listado de estaciones.

A medida que la red (Internet) fue creciendo, fue necesario delegar ciertas tareas de administración para hacer más fácil la vida de los administradores de las redes. Para eso se creó el servicio de DNS, uno de los primeros protocolos distribuidos y uno de los servicios más utilizados.

Para ver cómo funciona este servicio, usaremos el comando `nslookup`. Este comando nos provee de una consola interactiva con la que podemos efectuar distintas consultas. Empezaremos por una sencilla.

```
nslookup

> set q=any
```

```
> ar
Server:         192.168.21.13
Address:        192.168.21.13#53

Non-authoritative answer:
(...)
ar      nameserver = athea.ar.
ar      nameserver = ctina.ar.

Authoritative answers can be found from:
(...)
ar      nameserver = athea.ar.
ar      nameserver = ctina.ar.
```

Las respuestas *non-authoritative* son respuestas que hacen los DNS Servers a partir de lo que tienen almacenado (caché). Existe un grupo de DNS Servers que son las raíces de este árbol. Lo usual es que cada servidor nos diga dónde tenemos que seguir preguntando, hasta llegar a la raíz.

Para hacer esto, debemos indicarle a la aplicación, cuál es el servidor al que debe efectuar la consulta. Por ejemplo, teniendo en cuenta la respuesta anterior, podemos buscar la dirección IP de `athea.ar`:

```
> athea.ar
(...)
Non-authoritative answer:
Name:   athea.ar
Address: 200.16.98.2
(...)
```

Y con esta dirección fijar el servidor a utilizar en las próximas consultas.

```
> server 200.16.98.2
Default server: 200.16.98.2
Address: 200.16.98.2#53
```

Una vez fijado el servidor podemos efectuar una nueva consulta.

```
> com.ar
Server:         200.16.98.2
Address:        200.16.98.2#53

com.ar
        origin = athea.ar
```

```
        mail addr = noc-ar.atina.ar
        serial = 2003050701
        refresh = 21600
        retry = 3600
        expire = 1728000
        minimum = 21600
com.ar  nameserver = ns.uu.net.
com.ar  nameserver = ns1.retina.ar.
com.ar  nameserver = athea.ar.
com.ar  nameserver = ctina.ar.
com.ar  nameserver = merapi.switch.ch.
com.ar  nameserver = relay1.mecon.ar.
```

Ahora buscaremos la dirección de un dominio en particular.

```
> gnuservers.com.ar
Server:         200.16.98.2
Address:        200.16.98.2#53

Non-authoritative answer:
*** Can't find gnuservers.com.ar: No answer

Authoritative answers can be found from:
gnuservers.com.ar       nameserver = ns2.aktiv-assekuranz.com.
gnuservers.com.ar       nameserver = ns1.aktiv-assekuranz.com.
```

En este caso, el servidor no poseía dentro de su caché la dirección IP del dominio que buscamos. Para obtener el resultado correcto, debemos hacer la búsqueda de ns1.aktiv-assekuranz.com, yendo de nuevo al server donde empezamos las preguntas. Una vez que obtenemos esa dirección (200.68.69.99), podremos ejecutar la consulta en ese server.

```
> server 200.68.69.99
> gnuservers.com.ar
Server:         200.68.69.99
Address:        200.68.69.99#53

gnuservers.com.ar
        origin = gnuservers.com.ar
        mail addr = root.gnuservers.com.ar
        serial = 2003041208
        refresh = 604800
        retry = 86400
        expire = 2419200
```

```
        minimum = 604800
gnuservers.com.ar       nameserver = aktiv-assekuranz.com.
gnuservers.com.ar       nameserver = aleli.aktiv-assekuranz.com.
Name:   gnuservers.com.ar
Address: 200.42.58.163
gnuservers.com.ar       mail exchanger = 0 mail.gnuservers.com.ar.
```

El trabajo que acabamos de hacer, normalmente es realizado por el servicio del DNS que tengamos configurado en el archivo **/etc/resolv.conf**. Es decir, cuando nosotros hacemos una consulta a algún DNS, es este equipo el que se encarga de preguntar a los otros DNS del mundo hasta encontrar la respuesta a nuestra consulta.

Para pedirle al DNS que tenemos configurado que busque la información correspondiente a una determinada dirección, podemos utilizar el comando `dig gnuservers.com.ar`. Si queremos seleccionar un DNS en particular para hacer la pregunta, podemos escribir `dig @200.42.0.108 gnuservers.com.ar`

3.2. Telnet

El servicio de telnet es uno de los servicios más básicos y más antiguos, anterior incluso al servicio de DNS. Nos permite ingresar a un servidor remoto, con un usuario y contraseña, y acceder a ese servidor como si estuviéramos conectados desde una terminal. Normalmente este servicio funciona sobre el puerto 23 de TCP.

Se trata de un servicio muy antiguo y bastante obsoleto, ya que no utiliza ningún tipo de encriptación, y la información (incluyendo la contraseña del usuario) se transmite en texto plana, que algún usuario mal intencionado podría capturar.

Con el programa `tcpdump` pueden verse los paquetes que pasan por una placa de red, un módem o cualquier otro dispositivo de red. Con este programa, por ejemplo, es posible ver los paquetes de telnet que se transmiten sin encriptación. Sin embargo, es necesario ser super usuario para poder utilizarlo.

Sin embargo, el cliente de telnet (en GNU/Linux `telnet`) nos permite interactuar con otros servicios TCP de otros servidores de manera sencilla. Es decir, es posible acceder a una computadora (o un equipo cualquiera) utilizando diversos protocolos y puertos. Esto se debe a que el cliente de telnet utiliza el protocolo TCP en su forma más básica.

En cada caso será necesario conocer las palabras claves del protocolo para poder hacer las consultas que se desee.

3.3. Web

El servicio de www es uno de los más utilizados en Internet actualmente. Normalmente se lo utiliza a través de un programa especialmente pensado para ello (*browser* o *navegador*). El protocolo que se utiliza en este servicio es el HTTP (*HyperText Transfer Protocol*. Este servicio normalmente se utiliza en el puerto 80. El browser debe conocer tanto el protocolo HTTP, como el lenguaje HTML en que recibe el contenido que debe mostrar por pantalla.

Como se dijo antes, es posible utilizar el cliente de telnet para observar de forma detallada el accionar de este protocolo.

```
# telnet www.fi.uba.ar 80
GET /
```

En el caso de la página de www.fi.uba.ar, al efectuar esta consulta obtenemos una página muy sencilla que lo que hace es indicarle al navegador que debe pedir otra página (index.php). Una vez que hemos obtenido el resultado, la sesión se cierra, esto es parte del protocolo HTTP 1.0. Haciendo nuevamente la consulta, podremos obtener la página deseada.

```
# telnet www.fi.uba.ar 80
GET /index.php
```

Por otro lado, si queremos obtener una página a través de un proxy, será necesario conectarnos al puerto del proxy, que normalmente se encuentra en el puerto 8080, y hacer una consulta un poco distinta, incluyendo el protocolo a utilizar para el pedido.

```
# telnet proxy.fi.uba.ar 8080
GET http://www.google.com.ar HTTP/1.0
```

En este caso, al tener que especificarle el protocolo, también debemos escribir **ENTER** dos veces. Esto se debe a que en la siguiente línea sería posible especificar un nuevo pedido.

3.4. FTP

Otro servicio muy utilizado en Internet es el de FTP, *File Transfer Protocol*, que se utiliza para bajar o subir archivos a un determinado servidor. Este servicio normalmente se utiliza en el puerto 21.

Dos comandos que pueden existir en un sistema GNU/Linux pueden servir para aprender cómo funciona este protocolo: `ftp` y `lftp`. Ambos son clientes de ftp de consola, el segundo bastante más poderoso que el primero.

A continuación se explican algunos de los comando básicos de ftp. Dado que este protocolo nació dentro de computadoras UNIX, y aún hoy la mayoría de los servidores de FTP utilizan UNIX o GNU/Linux, los comandos que se usan son muy similares a los de GNU/Linux.

open host abre una conexión. En el caso del programa **ftp**, pregunta el usuario y contraseña inmediatamente, en otros programas, será necesario utilizar **user nombre** para que luego nos pregunte la contraseña.

ls lista los archivos en el servidor remoto.

cd directorio cambia de directorio.

put archivo sube el archivo al servidor.

get archivo baja el archivo del servidor.

mput archivos sube varios archivos al servidor (acepta meta-caracteres de shell para coincidir con varios archivos).

mget archivos baja varios archivos del servidor (también acepta meta-caracteres de shell).

Además de estos, en **lftp** existen muchos otros comandos para realizar tareas más complejas, como subir o bajar directorios completos con todos sus subdirectorios (**mirror**).

3.5. SSH

Una versión mejorada del servicio de telnet es el servicio de **ssh**, *Secure Shell.* Permite, al igual que el telnet, ingresar a un servidor remoto y utilizarlo del mismo modo que una terminal.

La diferencia principal reside en que el ssh utiliza encriptación para enviar la información, de modo que el usuario una seguridad apreciablemente mayor de la información que se transfiere a través de Internet.

El comando a usar en GNU/Linux para utilizar este servicio es simplemente `ssh`, que normalmente se conecta al puerto 22 del servidor remoto. Como para el resto de los servicios, también existen clientes de ssh para otros sistemas operativos.

Otros comandos relacionados con ssh son **scp** y **sftp**. El comando `scp` permite copiar archivos en forma segura entre dos computadoras diferentes, mientras que el comando `sftp` es un reemplazo del FTP, la diferencia reside en que utiliza encriptación mientras que FTP no.

3.6. Correo electrónico

En el envío y recepción del correo electrónico participa más de un protocolo. El protocolo SMTP es el que se utiliza para **enviar** emails, mientras que los protocolos POP e IMAP pueden utilizarse para **recibir** emails.

El protocolo SMTP funciona en el puerto 25, un servidor de SMTP recibe mensajes en ese puerto. El POP funciona en el puerto 110, un servidor POP envía mensajes desde ese puerto. El IMAP es una mejora sobre el protocolo POP, que funciona en el puerto 143, un servidor IMAP permite leer los mensajes en ese puerto.

3.6.1. SMTP

Se reproduce a continuación una sesión de SMTP con el servidor mail.fi.uba.ar, desde una computadora que no se encuentra dentro de la red de la Facultad de Ingeniería.

```
#telnet mail.fi.uba.ar 25

Trying 157.92.49.3...
Connected to srv-pc2.fi.uba.ar.
Escape character is '^]'.
220 srv-pc2.fi.uba.ar ESMTP Sendmail 8.11.6/8.11.6;
Sat, 10 May 2003 17:54:22 -0300
HELO ulises
250 srv-pc2.fi.uba.ar Hello h066060006167.netizen.com.ar [66.60.6.167]
(may be forged), pleased to meet you
MAIL FROM: marga@marga.com.ar
250 2.1.0 marga@marga.com.ar... Sender ok
RCPT TO: ngiaved@fi.uba.ar
250 2.1.5 ngiaved@fi.uba.ar... Recipient ok
DATA
354 Enter mail, end with "." on a line by itself
Subject: Prueba.

Esto es una prueba.
```

```
.
250 2.0.0 h4AKt9H02006 Message accepted for delivery
QUIT
```

Los comandos de SMTP utilizados son estándar y funcionan con cualquier servidor SMTP. El comando **HELO** se utiliza para comenzar la conexión. Una vez que se ha comenzado la conexión se pueden enviar varios mensajes. El comando **MAIL FROM:** indica la dirección que está enviando el mensaje, mientras que el comando **RCPT TO:** indica la dirección que va a recibir el mensaje. El comando **DATA** indica que comienza a transmitirse el contenido del mensaje, que termina con un punto.

Es interesante notar que el tema del mensaje (*Subject*) es parte del contenido del mensaje, es una tarea del programa que se utilice para leer el correo separar esta información del cuerpo del mensaje.

En este caso, el servidor SMTP está configurado para recibir correo de cualquier dirección, solamente para direcciones dentro de fi.uba.ar, si se intenta enviar un mensaje a una dirección que no se encuentre dentro del dominio, el servidor indicará que no se permite hacer *relay*: Relaying denied. IP name possibly forged [66.60.6.167]".

Es decir, **relay** es enviar un mensaje desde una dirección cualquiera, a otra dirección cualquiera. Lo usual es que los servidores SMTP permitan hacer relay únicamente a las computadoras que se encuentran en un rango de IPs determinado, que las identifica como computadoras que se encuentran dentro del mismo dominio que el servidor.

Esto significa que el servidor acepta dos tipos de mensajes únicamente:

- Los que tienen como destinatario una dirección @fi.uba.ar (normalmente, estos mensajes los enviarán otros programas de SMTP, en algún otro lugar del mundo).

- Los que se envían desde una dirección IP **dentro** del dominio de fi.uba.ar (estos serán los mensajes que enviaríamos con un cliente de correo normal, configurando mail.fi.uba.ar como nuestro SMTP, siempre que estemos dentro de la facultad).

Se reproduce a continuación otra sesión de envío de email, pero esta vez desde dentro de un servidor de la facultad de ingeniería. Notar la diferencia en la respuesta al HELO.

```
telnet mail.fi.uba.ar 25
```

```
Trying 157.92.49.3...
Connected to mail.fi.uba.ar.
Escape character is '^]'.
220 srv-pc2.fi.uba.ar ESMTP Sendmail 8.11.6/8.11.6;
Sat, 10 May 2003 18:19:15 -0300
HELO aleph
250 srv-pc2.fi.uba.ar Hello aleph.fi.uba.ar [157.92.49.1],
pleased to meet you
MAIL FROM: marga@marga.com.ar
250 2.1.0 marga@marga.com.ar... Sender ok
RCPT TO: gnu@marga.com.ar
250 2.1.5 gnu@marga.com.ar... Recipient ok
DATA
354 Enter mail, end with "." on a line by itself
Esto es otra prueba.
.
250 2.0.0 h4ALJqH03701 Message accepted for delivery
QUIT
221 2.0.0 srv-pc2.fi.uba.ar closing connection
Connection closed by foreign host.
```

3.6.2. IMAP

Entre las mejoras que incluye el protocolo IMAP están el manejo de carpetas directamente en el servidor y la posibilidad de ver los encabezados de los mensajes sin tener que ver todo el contenido.

De este modo, el usuario puede leer sus mensajes directamente del servidor, sin importar desde dónde los lea. También es posible utilizar este protocolo como si fuera POP, es decir, bajando todos los mensajes del servidor y almacenándolos en la computadora local.

3.7. NFS

El servicio NFS (Network File-System) es la forma más usual con la que se comparten archivos o, mejor dicho, sistemas de archivos en ambientes UNIX, es un servicio muy rápido que no sufre el tener un gran número de conexiones abiertas para distintos clientes, y tiene un básico manejo de seguridad (basado en las direcciones de las máquinas). No es un sistema seguro, los archivos se transmiten sin encriptación a través de la red y asume que los números (UID y GID) asignados a los usuarios serán los mismos en toda la red.

3.8. Samba

El protocolo que utiliza Microsoft Windows para compartir archivos e impresoras se llama SMB. En GNU/Linux podemos utilizar este protocolo utilizando las herramientas Samba.

Por ejemplo si queremos montar el directorio compartido Música de la maquina Ponce en el directorio tmp de nuestro home pondríamos:

```
smbmount //Ponce/Música ~/tmp
```

Esto nos pedirá la contraseña del directorio compartido para poder realizar la operación. Muchas veces no se necesita una contraseña y alcanza con presionar ENTER una vez más. Es posible utilizar el parámetro `-o guest`, para decirle que no tiene contraseña y que no se necesite el ENTER adicional.

3.9. NIS

Es un protocolo destinado a exportar información y configuraciones a diversas máquinas de la red, como por ejemplo, los usuarios y contraseñas. Modifica y regula en gran parte el funcionamiento de la red, ya que, en vez de tener cada máquina una configuración local tendrán que preguntarle al servidor de NIS cada vez que se necesita esa configuración.