Universidad de Alcalá

Departamento de Automática

REDES DE COMPUTADORES
Laboratorio

Anexo-Práctica 1:
COMANDOS BÁSICOS PARA LA CONFIGURACIÓN DEL NIVEL IP EN UNA RED DE SISTEMAS UNIX

Grado en Ingeniería Informática
Curso 2018/19

1. LA INTERFAZ `loopback`

La primera interfaz que es necesario activar al configurar el nivel IP es la interfaz `loopback`, o bucle de retorno, la cual puede ser entendida como una especie de "cortocircuito" en la pila de protocolos. Es decir, los *datagramas* que se envían a esta interfaz no abandonan nunca la máquina, sino que retornan inmediatamente al nivel IP que los envía. El nombre estándar de la interfaz `loopback` es `lo` y su dirección IP es `127.0.0.1` pues usa la red de clase A 127 y el nombre `localhost`.

El tráfico enviado a la dirección de bucle de retorno no aparece nunca en ninguna red adjunta. Un aspecto especial de esta interfaz es que el tráfico enviado desde un *host* a sí mismo se redirecciona a la dirección del bucle de retorno. La interfaz de bucle de retorno se utiliza normalmente para establecer conexiones de red entre aplicaciones basadas en el paradigma cliente-servidor hospedadas en la misma máquina. Esto reduce el tráfico en la red local e incluso permite que dichas aplicaciones funcionen en ausencia de una red física. La interfaz `loopback` se debe configurar siempre (independientemente de que la máquina esté conectada a una red o no), ya que es útil, no sólo como ejemplo o como método de pruebas, sino porque algunas aplicaciones la utilizan como modo normal de operación. La mayor parte de sistemas operativos que incluyen la pila de protocolos TCP/IP configuran por defecto esta interfaz en el proceso de instalación del sistema.

2. COMANDOS BÁSICOS DE CONFIGURACIÓN DE RED EN SISTEMAS UNIX

2.1 COMANDO `ifconfig`

Este comando descubre una interfaz al nivel de red (IP) del *kernel* del Sistema Operativo. Esto implica asignar una dirección IP a la placa de red y activarla. Una vez que se ha "pinchado" una tarjeta de red Ethernet en un ordenador deberá procederse a informar al sistema de los parámetros de utilización de la misma, es decir, incluir mediante una serie de *scripts* las funcionalidades de inicio, paro y configuración de su dispositivo de interfaz. Los *scripts* de red en Linux utilizan el comando `ifconfig` (*Interface Configuration*) que se usa precisamente para todo lo indicado anteriormente: arranque de la interfaz de red, paro del dispositivo y configuración de los parámetros de red.

El comando `ifconfig` permite a `root` configurar (o reconfigurar) las interfaces de red. Los usuarios distintos de `root` pueden invocar también al comando con fines informativos. Para ello deben especificar el camino completo (`/sbin/ifconfig`). Si se ejecuta sin parámetros `ifconfig` presenta información acerca de la configuración y funcionamiento de las interfaces de red activas en ese momento. Por tanto, `ifconfig` es la primera herramienta que debe usarse cuando se desee conectar un ordenador a una red. La sintaxis del comando es la siguiente:

```
#ifconfig <interfaz> <dir_ip> netmask <máscara> broadcast <dir_broadcast> up
```

- En `interfaz` debe proporcionarse el nombre de la interfaz de red que se desea configurar. Generalmente el nombre de interfaz se forma a partir de un nombre de manejador (*driver*) seguido de un número de unidad. El nombre de manejador para redes Ethernet es `eth` y las unidades comienzan a numerarse a partir de `0` hasta el número de interfaces existentes del mismo tipo menos uno (`eth0, eth1,` etc.).
- `dir ip` es la dirección IP que se desea asignar a la interfaz de red.
- `máscara` establece la máscara de red que se desea utilizar para la interfaz. Si no se proporciona este valor, la máscara de red tomará por defecto el valor predeterminado para direcciones de redes clase A, B o C en función de la dirección IP que se haya especificado en `dir ip`.

- `dir broadcast` al proporcionar una dirección de *broadcast* con la opción `broadcast`, se indica a la interfaz de red que se desea que habilite el modo de difusión *broadcast* dirigido y que contemple dicha dirección. La dirección de *broadcast* dirigida a una red se determina a partir de la dirección IP de cualquiera de los equipos pertenecientes a dicha red y su máscara de red, ya que se forma a partir de la dirección de red poniendo '1's en la parte de dirección correspondiente a los equipos.

Suponer que se acaba de instalar una tarjeta Ethernet en el ordenador de nombre `puesto8` del laboratorio. Si Linux ha reconocido la tarjeta y no aparece ningún problema de hardware, para configurar la tarjeta se podría usar la siguiente línea de comando:

```
#ifconfig eth0 158.49.98.8 broadcast 158.49.98.63 netmask 255.255.255.192
```

Esto significa que mediante el comando `ifconfig` le estamos indicando al sistema que la dirección IP de la interfaz `eth0` es `158.49.98.8;` que su máscara de red es `255.255.255.192;` y que su dirección de *broadcast* es la `158.49.98.63.`

Si a continuación se utiliza el comando `ifconfig` sin ningún parámetro se mostrará en pantalla toda la información sobre las interfaces de red disponibles actualmente en nuestro sistema, como puede observarse en el siguiente ejemplo:

```
#ifconfig

eth0      Link encap:Ethernet  HWaddr 00:05:1C:0F:19:2E
          inet addr:158.49.98.8  Bcast:158.49.98.63  Mask:255.255.255.192
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:14 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:8 dropped:0 overruns:0 carrier:16
          collisions:0 txqueuelen:100
          RX bytes:1098 (1.0 Kb)  TX bytes:0 (0.0 b)
          Interrupt:11 Base address:0xec00
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:24 errors:0 dropped:0 overruns:0 frame:0
          TX packets:24 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:2016 (1.9 Kb)  TX bytes:2016 (1.9 Kb)
```

donde, `eth0` es la interfaz de red disponible en ese momento y a continuación se observan todos los valores actuales (dirección IP, dirección *broadcast*, máscara de red, etc.). `lo` muestra el estado de la interfaz de `loopback`.

También puede utilizarse el comando de la siguiente forma para activar o desactivar la interfaz:

```
ifconfig eth0 down

ifconfig eth0 up
```

Este comando es más completo que lo expuesto hasta este momento. Existen otros parámetros importantes que hay que tener en cuenta a la hora de configurar un dispositivo de red.

2.2 COMANDO `route`

Permite añadir y/o borrar entradas en la tabla de enrutamiento y además muestra su contenido. El comando `route print` muestra una lista con las rutas actualmente conocidas por IP para el equipo. `route add` se utiliza para añadir rutas a la tabla de rutas, y `route del` se usa para borrar rutas de la tabla. Todos los nombres simbólicos usados para el destino se consultan en el archivo de la base de datos de la red `/etc/networks`. Los nombres simbólicos se consultan en el archivo de la base de datos de nombres de equipos `/etc/hosts`. Si no se puede resolver un nombre simbólico en base al contenido de estos ficheros se recurrirá al sistema de nombres de dominio DNS (Domain Name System). Por defecto, `route` muestra la tabla de rutas intentando determinar nombres simbólicos. Si se desea que muestre direcciones IP en lugar de nombres simbólicos, deberá utilizarse con la opción `-n`. Para añadir a la tabla de rutas una ruta hacia una red concreta se utilizará el siguiente formato:

```
#route add -net <dir_ip> netmask <máscara> [gw <dir_ip_gateway>]dev <interfaz>
```

en el que los argumentos tienen el siguiente significado:

- `dir ip`. `route` puede utilizarse para establecer rutas estáticas hacia equipos y redes vía una interfaz de red una vez que se ha configurado esta última mediante la utilidad `ifconfig`. La mayor parte de las veces se utilizará para establecer rutas hacia redes, por lo que debe utilizarse con la opción `-net`, con lo que `dir ip` se interpretará como una dirección de red para la que se desea establecer una ruta. Si el sistema de nombres de dominio DNS está configurado y funcionando correctamente, puede utilizarse un nombre simbólico en lugar de una dirección IP. Si se desea establecer una ruta a un equipo, deberá utilizarse la opción `-host`.
- `máscara` especifica la máscara de red para la ruta que se desea añadir. Durante el proceso de encaminamiento de cada paquetes IP se realizará una operación lógica AND entre la dirección IP destino contenida en su cabecera y esta máscara. Si el resultado de dicha operación coincide con la `dir ip`, se utilizará esta entrada de la tabla de rutas para realizar el encaminamiento de dicho paquete IP. Si no se indica una máscara de red, se utilizará una de las máscaras por omisión para las clases de direcciones A, B y C en función de la clase a la que pertenezca `dir_ip`.
- `dir_ip_gateway`. Cuando se utiliza la entrada que se está creando en la tabla de rutas para enviar un paquete IP, dicho paquete se encaminará a través del *gateway* indicado en esta opción. El *gateway* debe ser accesible, por lo que, previamente, deberá haberse establecido una ruta hacia el mismo (por ejemplo utilizando el comando `route add` con la opción `-host`). Si se especifica la dirección IP de una de las interfaces de red locales, se utilizará ésta para decidir la interfaz hacia la que se encaminarán los paquetes. Es opcional, sólo aparece cuando es necesario encaminar hacia un *gateway*.
- En `interfaz` debe proporcionarse el nombre de la interfaz de red que se desea utilizar en esta ruta. Generalmente el nombre de `interfaz` se forma a partir de un nombre de manejador (`eth`) seguido de un número de unidad (0, 1, 2, …).

El siguiente ejemplo añade una entrada en la tabla de enrutamiento en la sintaxis que admite el sistema:

```
#route add default gateway 158.49.98.62
```

En el ejemplo anterior no se proporciona la máscara ni la interfaz. Esta información no es necesaria si ya se ha configurado una de las interfaces del ordenador que está en la misma red.

El comando `route [-n]` muestra el contenido de la tabla de *routing*. Con la opción `-n` se muestran las direcciones IP en forma numérica. Por ejemplo:

```
#route -n
Kernel IP routing table
Destination  Gateway  Genmask             Flags MetricRef   Use   Iface
158.49.98.0  0.0.0.0  255.255.255.255.0   U     0      0      0     ethO
127.0.0.0    0.0.0.0  255.0.0.0           U     0      0      0     lo
```

En algunos sistemas UNIX el comando `route` sólo permite añadir entradas en la tabla de enrutamiento. En éstos el contenido de la tabla de enrutamiento se muestra con el comando `netstat –nr`

2.3 COMANDO `arp`

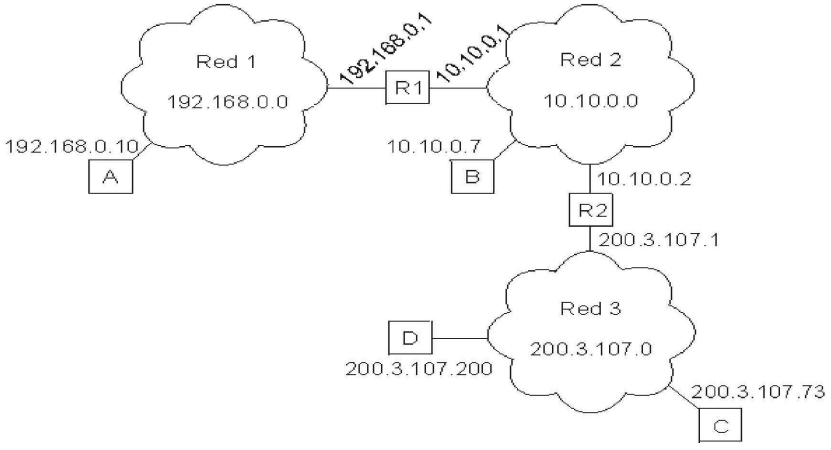
Todos los ordenadores de una misma red comparten el mismo medio físico, por lo que debe de existir un identificador único para cada equipo o, mejor dicho, para cada tarjeta de red. Cuando se envían datos en una red local hay que especificar claramente a quien van dirigidos. Esto se consigue mediante la dirección MAC (Médium Access Control), un número compuesto por 12 dígitos hexadecimales que identifica de forma única a cada dispositivo Ethernet. La dirección MAC se compone de 48 bits. Los 24 primeros bits identifican al fabricante del hardware y los 24 bits restantes corresponden al número de serie asignado por el fabricante, lo que garantiza que dos tarjetas no puedan tener la misma dirección MAC ya que direcciones MAC duplicadas causarían problemas en la red.

El comando `arp` Permite ver y modificar manualmente la tabla que mantiene el módulo ARP (Address Ressolution Protocol). En esta tabla están las correspondencias entre las direcciones IP y las direcciones hardware. Las invocaciones más normales de este comando son:

`#arp –a`	Muestra la tabla ARP
`#arp –s dir_IP dir_Hard`	Asigna la dir ección hardware `dir_Hard` a la dirección IP `dir_IP`
`#arp –d dir_IP`	Borra la entrada `dir_IP` de la tabla de rutas.

Cada ordenador almacena una tabla de direcciones IP y direcciones físicas. Cada vez que ARP formula una pregunta y le responden, inserta una nueva entrada en su tabla.

La primera vez que C, en la siguiente figura, envíe un mensaje a D tendrá que difundir previamente una pregunta ARP. Sin embargo, las siguientes veces que C envíe mensajes a D ya no será necesario realizar nuevas preguntas puesto que C ya habrá almacenado en su tabla la dirección física de D. Sin embargo, para evitar incongruencias en la red debido a posibles cambios de direcciones IP o adaptadores de red, se asigna un tiempo de vida de cierto número de segundos a cada entrada de la tabla. Cuando se agote el tiempo de vida de una entrada, ésta será eliminada de la tabla.



Las tablas ARP reducen el tráfico de la red al evitar preguntas ARP innecesarias. Pensemos ahora en distintas maneras para mejorar el rendimiento de la red. Después de una pregunta ARP, el destino conoce la dirección IP y la dirección física del origen. Por lo tanto, podría insertar la correspondiente entrada en su tabla. Pero no sólo eso, sino que todas las estaciones de la red que escuchan la pregunta ARP: podrían insertar también las correspondientes entradas en sus tablas. De esta forma se va construyendo la tabla dinámicamente. Como es muy probable que otros ordenadores se comuniquen en un futuro con el primero, así se habrá reducido el tráfico de la red optimizando su rendimiento.

Lo explicado en los párrafos anteriores sirve para comunicar dos ordenadores conectados a la misma red. Si no estuviesen conectados a la misma red, sería necesario atravesar uno o más *routers* hasta llegar al *host* destino desde el origen. Si el ordenador origen no tiene en su tabla la dirección física del router, formulará una pregunta ARP solicitando al router su dirección física. Estos pasos se van repitiendo para cada red hasta llegar al ordenador destino.

Si se quieren conocer las direcciones hardware de otros equipos de la red, se hará uso de la *cache arp* de nuestro equipo, mediante el comando `arp -a`. Este comando muestra la relación IP/MAC de los equipos que la *cache* tiene almacenados en ese momento. Para poder comunicar con un equipo de la red debe conocerse previamente su dirección MAC.

Si se desea obtener la dirección Ethernet de un ordenador primero se le puede hacer un `ping` (como se explica en el apartado siguiente). De esta forma se almacenará la dirección MAC en la *cache* y después mediante `arp -a` se podrá obtener su dirección MAC.

2.4. COMANDO `ping`

Este comando puede entenderse como un sonar, que permite verificar si una interfaz está al alcance del nivel de red, y medir el retardo de ida y vuelta (RTT *Roun Trip Time*) que hay hasta el nodo destino. El comando `ping` envía periódicamente un paquete a la dirección que se da como parámetro que provoca la respuesta del destino.

Para detener `ping` hay que hacer `Ctrl-C`. El comando se utiliza generalmente para testear aspectos de la red, como comprobar que un sistema está encendido y conectado; esto se consigue enviando a dicho equipo paquetes ICMP (de tipo `ECHO/SMALL> REQUEST`), tramas que causarán que el núcleo del sistema remoto responda con paquetes ICMP, pero esta vez de tipo `ECHO/SMALL> RESPONSE`. Al recibirlos, se asume que la máquina está encendida.
En el siguiente ejemplo se comprueba si es posible acceder al ordenador de dirección IP 158.49.98.12, además de otra información estadística del tráfico de paquetes.

```
#ping 158.49.98.12

PING 158.49.98.12 (158.49.98.12): 56 data bytes
64 bytes from 158.49.98.12:   icmp seq=0   ttl=255   time=0.6 ms
64 bytes from 158.49.98.12:   icmp seq=1   ttl=255   time=0.6 ms
64 bytes from 158.49.98.12:   icmp seq=2   ttl=255   time=0.6 ms
^C
---   158.49.98.12 ping statistics  ---
3 packets transmitted,  3 packets received,  0% packet loss
round-trip  min/avg/max = 0.6/0.6/0.6  ms
```

El campo `ttl` indica el tiempo de vida de los paquetes enviados. Un `ttl=255` significa que el paquete puede atravesar hasta 255 *routers* en su camino hasta el *host* destino. Cada *router* por el que pase irá disminuyendo en una unidad el valor del campo `ttl`, y cuando llega a 0 el paquete se descarta, enviándose al origen un mensaje ICMP de tipo 11 (*Time Exceeded*). Esto se hace para no tener en la red paquetes indefinidamente.

También puede hacerse `ping` a una interfaz del mismo ordenador, como por ejemplo el `loopback` que se muestra en el ejemplo siguiente:

```
#ping 127.0.0.1

PING 127.0.0.1 (127.0.0.1): 56 data bytes
64 bytes from 127.0.0.1:   icmp seq=0   ttl 64   time=0.1 ms
64 bytes from 127.0.0.1:   icmp seq=1   ttl 64   time=0.0 ms
^C
---   127.0.0.1 ping statistics  ---
2 packets transmitted,  2 packets received,  0% packet loss
round-trip  min/avg/max = 0.0/0.0/0.1  ms
```

Puede hacerse también `ping` a la dirección de *broadcast* de una interfaz para saber qué otros ordenadores están conectados a la misma red:

```
#ping 158.49.98.255

PING 158.49.98.255 (158.49.98.255): 56 data bytes
64 bytes from 127.0.0.1:     icmp seq=0   ttl=64   time=0.1 ms
64 bytes from 158.49.98.6:   icmp seq=0   ttl=255  time=0.6 ms  (DUP!)
64 bytes from 158.49.98.2:   icmp seq=0   ttl=255  time=0.7 ms  (DUP!)
…
^C
---   158.49.98.255 ping statistics  ---
1 packets transmitted,1 packets received,+12 duplicates 0% packet loss
round-trip  min/avg/max = 0.1/1.9/7.1  ms
```

Los DUP indican que se ha recibido más de un paquete de respuesta duplicada a un mismo `ping`.

2.5. EL FICHERO `/etc/hosts`

Para evitar el incómodo uso de las direcciones IP, los sistemas UNIX permiten asignar nombres (a modo de etiquetas) a las interfaces. Para ello se usa el fichero `/etc/hosts` cuyo contenido podría ser algo del estilo de lo mostrado en el siguiente ejemplo:

```
#more /etc/hosts
127.0.0.1     localhost
158.49.98.12  puesto12
158.49.98.23  puesto23
```

El uso de nombres en lugar de direcciones IP facilita la tarea de comunicarse con los ordenadores de la red.