

Redes de Computadoras  
*Guión de Prácticas*

Vicente González Ruiz  
Depto de Arquitectura de Computadores y Electrónica  
vruiz@ual.es  
<http://www.ace.ual.es/~vruiz/docencia>

January 15, 2008

# Índice

|          |   |          |
|----------|---|----------|
| <b>1</b> | <b>El laboratorio de redes</b>                                      | <b>2</b> |
| 1.1      | Hardware . . . . .  | 2        |
| 1.2      | Software . . . . .  | 4        |
| 1.2.1    | Windows . . . . .   | 4        |
| 1.2.2    | Fedora Core Linux . . . . .   | 4        |
| 1.2.3    | Debian Linux del PC virtual . . . . .                               | 4        |
| 1.3      | Funcionamiento básico del <i>VMware</i> . . . . .                   | 5        |
| 1.3.1    | El PC virtual . . . . .   | 5        |
| 1.3.2    | Usuarios . . . . .  | 5        |
| 1.3.3    | Conectividad . . . . .  | 6        |
| <b>2</b> | <b>Atenuación de las señales</b>                                    | <b>7</b> |
| 2.1      | ¿Qué es una señal? . . . . .  | 7        |
| 2.2      | ¿Cuándo transporta información una señal? . . . . .                 | 7        |
| 2.3      | ¿Atenuación? . . . . .  | 7        |
| 2.4      | ¿Qué es la relación señal/ruido? . . . . .                          | 8        |
| 2.5      | ¿Qué es una señal digital? . . . . .                                | 8        |
| 2.6      | La digitalización de señales . . . . .                              | 8        |
| 2.6.1    | Digitalización de señales digitales . . . . .                       | 9        |
| 2.7      | Espectro de frecuencias de una señal . . . . .                      | 10       |
| 2.8      | ¿Cómo es el espectro de frecuencias de una señal digital? . . . . . | 12       |
| 2.9      | ¿Cómo afecta la atenuación de las señales a su espectro? . . . . .  | 12       |
| 2.10     | El filtrado de señales . . . . .                                    | 13       |

---

## List of Examples

|   |   |       |   |
|---|---|-------|---|
| 1 | a | ..... | 2 |
|---|---|-------|---|

# Práctica 1

## El laboratorio de redes

Las prácticas de redes de computadoras son realizadas en el laboratorio de redes. Dedicaremos esta primera práctica a conocer el hardware y el software del que disponemos.

### Example 1 (a)

ejemplo

---

### 1.1 Hardware

En el laboratorio existen 25 computadoras con las siguientes características:

- \* Procesador: AMD Athlon 64 bits 3500
- \* Memoria RAM: 1GB
- \* Disco Duro: 80GB ATA
- \* Adaptadores de red: DLINK DFE-528TX, Linksys NC100, ALi M5263
- \* Tarjeta de Video: ATI RADEON 9250
- \* Tarjeta de Sonido: Integrada en placa base Realtek AC'97

Como se deduce del tipo de los adaptadores de red que tienen instalados las computadoras (hosts), la red del laboratorio es una red Ethernet. Además, como utiliza conmutadores (switches) Ethernet, podemos decir que se trata de una Ethernet conmutada (hay diferentes tipos de redes Ethernet). Finalmente, si seccionáramos uno de los cables que unen los adaptadores de red de los hosts con los conmutadores, veríamos que se están utilizando enlaces fabricados con pares trenzados de cobre UTP, categoría 5. En Ethernet estos enlaces son full-duplex (hay pares para enviar datos y otros para recibir datos, que funcionan de forma simultánea).

La topología de la red se muestra en la Figura 1.1.

La tasa de transmisión entre un host y su conmutador directamente conectado es de 100 Mbps y la tasa de transmisión entre los conmutadores es de 100 Mbps.

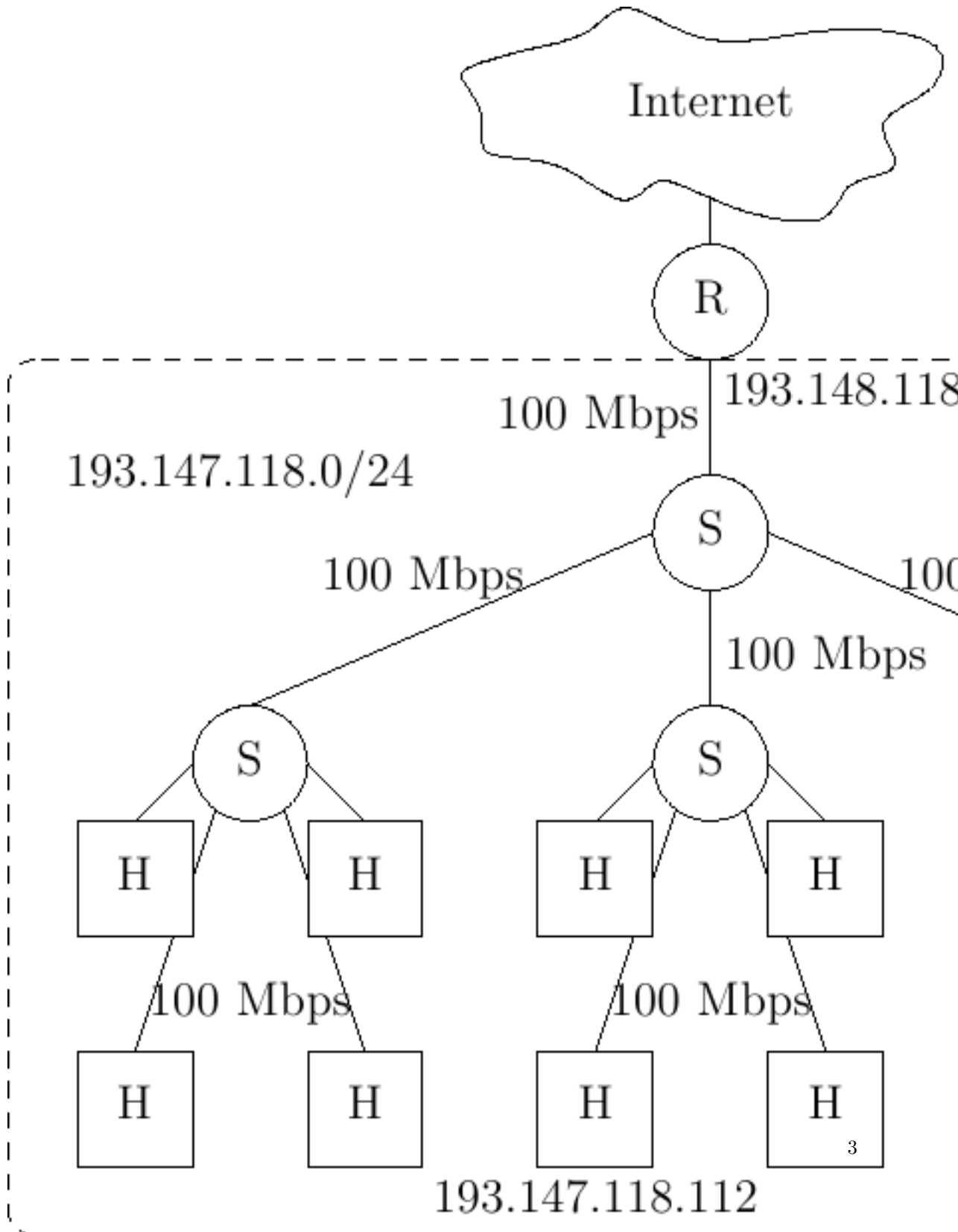


Figure 1.1: Topología física del Laboratorio de Redes.

## 1.2. SOFTWARE

---

Por tanto, la tasa de transmisión entre un host del laboratorio y el resto de Internet es a lo sumo, 100 Mbps.

### 1.2 Software

Cada computadora tiene instalados (al menos) dos sistemas operativos: Microsoft Windows XP y Linux. Ambos permiten conectarse a Internet mediante la pila de protocolos TCP/IP. Esto significa que seremos capaces de comunicarnos con otros usuarios de la red si el software de red ha sido convenientemente configurado.

#### 1.2.1 Windows

Utilizaremos Windows para las tareas típicas de navegar por Internet, descargar programas, etc. Para acceder a la computadora utilizaremos la cuenta:

**Usuario:** redes

**Contraseña:** redes

**Taller 1.1:** Si está consultando este documento en un ordenador del laboratorio usando Windows, ya habrá accedido a su cuenta de usuario. Si no es así, hágalo y si existe algún problema, comuníquese al profesor.

#### 1.2.2 Fedora Core Linux

Como una alternativa a Windows, podremos utilizar Linux para acceder a la computadora y realizar las prácticas. Los datos del usuario que deberemos utilizar son:

**Usuario:** redes

**Contraseña:** redes

**Taller 1.2:** Si está consultando este documento en un ordenador del laboratorio usando Linux, ya habrá accedido a su cuenta de usuario. Si no es así, hágalo y si existe algún problema, comuníquese al profesor.

#### 1.2.3 Debian Linux del PC virtual

Para realizar la mayoría de las prácticas de redes utilizaremos Linux, Debian Linux, en concreto.

Debido a que es necesario realizar muchas acciones como administrador del sistema, existe un peligro potencial de estropear el buen funcionamiento del sistema operativo. Por otra parte, en algunas prácticas es interesante poder utilizar más de una computadora. Aunque esto podría ser posible físicamente, lo más cómodo es utilizar un software de emulación de PC que permita ejecutar en una misma computadora cuántos PC's virtuales sean necesarios.

### 1.3. FUNCIONAMIENTO BÁSICO DEL VMWARE

---

Estos dos problemas son resueltos utilizando VMware (<http://www.vmware.com>). Dicho software permite definir una computadora virtual e instalar en ella los programas que queramos. Nosotros vamos a utilizar una distribución Debian Linux (<http://www.debian.org>) instalada en un PC virtual.

## 1.3 Funcionamiento básico del VMware

VMware consta de muchos programas de los cuales, para realizar las prácticas, sólo necesitaremos el VMware Player (<http://www.vmware.com/download/player/>). Este programa es de libre distribución.

VMware Player permite ejecutar el PC virtual sin necesidad de usar ninguna otra utilidad extra. Para definir un PC virtual se utiliza el programa VMware Workstation (<http://www.vmware.com/download/ws/>) que sí es de pago.

### 1.3.1 El PC virtual

Con el objetivo de acelerar las prácticas lo máximo posible se ha creado un PC virtual y se le ha instalado Debian Linux. Dicho PC virtual puede descargarse de esta URL: <http://www.ace.ual.es/~vruiiz/docencia/redes/debian>.

Como puede apreciarse se trata de una colección de ficheros que contienen el contenido del PC virtual. Dicho PC tiene las siguientes características:

- \* Procesador: El mismo que el host físico
- \* Memoria RAM: 256 MB
- \* Disco Duro: 8 GB BusLogic BT-958 SCSI
- \* Adaptador de red: Advanced Micro Devices [AMD] 79c970 [PCnet32 LANCE]
- \* Tarjeta de Video: VMware Inc [VMware SVGA II] PCI Display Adapter
- \* Tarjeta de Sonido: La misma que el host físico

**Taller 1.3:** Realice los siguientes pasos:

1. Descargue e instale el programa VMware Player (<http://www.vmware.com/download/player/>) (si no estuviera ya instalado). Seleccione el sistema operativo adecuado en su caso.
2. Descargue el PC virtual. Recuerde que tiene que descargarse todos los ficheros que parecen en la anterior URL.
3. Ejecute el PC virtual.

### 1.3.2 Usuarios

La instalación de Linux Debian que descargamos tiene creado sólo un usuario: el usuario root con password toor. El usuario root en linux es el usuario administrador del sistema. Posee todos los privilegios posibles y por supuesto, usándolo sin conocimiento hay un peligro potencial de estropear el sistema operativo. **Por tanto, usaremos la “cuenta de root” sólo cuando sea necesario.**

**Taller 1.4:** Como acabamos de indicar, nos identificaremos en el sistema como `root` sólo cuando sea imprescindible. Una de estas situaciones se da durante la creación de una nueva cuenta de usuario (véase el Apéndice ??).

Acceda al PC virtual como administrador y cree una cuenta de usuario. Anote el nombre de usuario (`login`) y clave de acceso (`password`) para poder usar esta cuenta más adelante.

#### 1.3.3 Conectividad

El Player puede configurarse de tres formas diferentes dependiendo del nivel de conectividad que queremos dotar a los PC's virtuales. Las alternativas son:

1. *Bridged*: El PC virtual es dotado con una IP real, perteneciente a la red a la que pertenece el host donde se ejecuta el PC virtual (host huésped). Esto sólo es posible hacerlo si disponemos de dicha IP.
2. *NAT*: El PC virtual accede a la red a través de un *NAT box*. La IP usada es, por tanto, privada y asignada dinámicamente por un servidor de DHCP instalado en el host huésped (el que ejecuta Windows, por ejemplo).
3. *Host Only*: En este caso el PC virtual no puede conectarse a ningún otro host, independientemente de que sea real o virtual.

**Taller 1.5:** Para comprobar si hay acceso a Internet desde el PC virtual, actualice los paquetes instalados en él (véase el Apéndice ??). Nótese que esta acción sólo puede realizarse como superusuario (usuario `root`).

**Cuestión 1.1:** ¿Qué tasa de transmisión a lo sumo vamos a obtener entre dos hosts del laboratorio? Razone su respuesta.

**Cuestión 1.2:** Descargue un archivo de Internet usando su navegador. ¿Cuál ha sido la tasa de descarga? Explique cómo y por qué ha realizado dicho cálculo.

**Cuestión 1.3:** ¿Cuántos conmutadores, a lo sumo, cruzará un paquete de datos transmitido desde un host del laboratorio hasta cualquier otro host del laboratorio? Razone su respuesta.

**Cuestión 1.4:** ¿Cuántos conmutadores, como mínimo, han atravesado el archivo que se ha descargado? Considere la NAT box del VMware como un conmutador más. Razone.

## Práctica 2

# Atenuación de las señales

### 2.1 ¿Qué es una señal?

Una señal [3] es cualquier perturbación medible de las características físicas de un medio (el que sirve de transmisión).

### 2.2 ¿Cuándo transporta información una señal?

Las señales suelen ser impredecibles para el receptor. Cuando una señal es recibida y además ésta es impredecible de alguna manera, la señal aporta información al receptor. También se dice, en este caso, que disminuye la incertidumbre del receptor [4].

### 2.3 ¿Atenuación?

Todos los medios de transmisión atenúan las señales que transportan. Esta atenuación produce un decremento en la energía que transporta la señal y por tanto, cuanto más alejados están el emisor y el receptor, menor sería dicha energía.

La cantidad de energía que se pierde durante la transmisión depende del tipo de medio\* y de la frecuencia de la señal. Generalmente, las señales de mayor frecuencia se atenúan más que las señales de menor frecuencia [5]. Por este motivo, por ejemplo, cuando nos acercamos a un lugar donde hay un concierto de música escuchamos primero los sonidos de más baja frecuencia (el bombo de la batería) y posteriormente el resto de instrumentos.

---

\*En los medios lineales (como pueda ser un cable) la pérdida es lineal con la distancia recorrida. En los medios de difusión (como es el aire), la pérdida es exponencial con la distancia.

## 2.4 ¿Qué es la relación señal/ruido?

Los medios de transmisión nunca están totalmente libres de señales indeseables (*ruido*) que pueden interferir en el receptor con la señal que nosotros intentamos transmitir. En adelante, llamemos a esta señal (la que nosotros deseamos transmitir), *señal de datos*.

A la relación entre la energía de señal de datos y la energía del resto de señales que llegan también al receptor (y que perturban la recepción de la señal de datos) es lo que se conoce como *relación señal-ruido* o SNR (Signal-Noise Ratio). Dicho de otra forma:

$$\text{SNR} = \frac{\text{Energía de la señal de datos}}{\text{Energía de la señal de ruido}}$$

## 2.5 ¿Qué es una señal digital?

Las *señales digitales* son aquellas que transportan información digital [2]. Estas señales se diferencian de las analógicas en que sólo unos determinados valores de la señal tienen un significado en concreto. Por ejemplo, una señal digital puede oscilar entre 0 y 5 voltios. El primer voltaje indicaría que se transmite la información "0" y el segundo que se transmite la información "1". Un voltaje de 2.5 voltios no significa nada.

**Cuestión 2.1:** Discuta si el grado de abertura de una puerta es una información digital o analógica.

## 2.6 La digitalización de señales

Las señales (independientemente de si representan fuentes de información analógicas o digitales) necesitan digitalizarse antes de ser procesadas por un sistema digital (una computadora, por ejemplo). El proceso al que deben ser sometidas se conoce como *digitalización* y consiste básicamente en registrar el valor aproximado de la señal cada cierto intervalo de tiempo.

El número de muestras capturadas durante la digitalización está directamente relacionado con la máxima componente de frecuencia de la señal. Dicha relación es expuesta por el *Teorema del Muestreo Uniforme* de esta forma:

**Si capturamos  $2n$  muestras/segundo de una señal entonces la máxima componente de frecuencia registrada de dicha señal tiene  $n$  Hercios.**

La digitalización implica normalmente una pérdida irreversible de parte de la información que transporta la señal, por dos motivos:

1. Si la señal no está limitada en banda por mucho que aumentemos la frecuencia de muestreo no será posible registrar todas sus componentes de frecuencia.

2. Si la señal no es digital, es imposible utilizando un número finito de bits representar cada uno de sus estados.

Evidentemente, las señales digitales digitalizadas sólo adolecen del primer problema.

### 2.6.1 Digitalización de señales digitales

Una forma sencilla de crear versiones digitales de señales digitales consiste en replicar tantas veces como sea preciso cada uno de los distintos valores que toma la señal digital. Esto es lo que realiza el programa <http://www.ace.ual.es/~vruiz/docencia/redes/practicas/sampler.c> (véase el Apéndice ??). Nótese que si la señal no fuera digital este proceso no sería tan simple.

**Cuestión 2.2:** Razone por qué no sería tan simple.

En la entrada de `sampler.c`, cada número real (representado físicamente como un número en punto flotante de 32 bits de precisión) indica un estado de la señal digital. La salida es simplemente una replicación de cada uno de estos estados.

**Cuestión 2.3:** ¿No cree que sobran bits para representar cada muestra? ¿Cuántos cree que serían realmente necesarios? ¿Por qué cree que se han usado números reales?

Para generar el fichero de entrada para `sampler.c` podemos usar el programa <http://www.ace.ual.es/~vruiz/docencia/redes/practicas/ascii2float.c> (véase el Apéndice ??).

Finalmente, para averiguar el contenido de un fichero `.float` podemos utilizar el programa <http://www.ace.ual.es/~vruiz/docencia/redes/practicas/float2ascii.c> (véase el Apéndice ??).

**Taller 2.1:** 1. Ejecute el PC virtual (<http://www.ace.ual.es/~vruiz/docencia/redes/debian>). El resto de pasos se realizarán en él.

2. Descargue el fichero <http://www.ace.ual.es/~vruiz/docencia/redes/practicas/sources.tar.gz>.
3. Descomprima el fichero anterior. Se creará un directorio llamado `sources`. En ese directorio encontraremos todos los programas que vamos a utilizar.
4. Compile los fuentes en C. Situándonos en el directorio `sources`, escribir:  

```
make all
```
5. Asegurarse de que los scripts poseen permiso de ejecución para el dueño del fichero (aparece una "x" en la cuarta posición comenzando desde la izquierda cuando escribimos el comando `ls -l` en el directorio en el que hemos descargado el script).

**Taller 2.2:** Ejecute:

## 2.7. ESPECTRO DE FRECUENCIAS DE UNA SEÑAL

---

```
ascii2float << EOF | float2ascii
```

y escriba una serie de números reales separados por la tecla <Enter>. Cuando no desee introducir más números escriba EOF y pulse finalmente <Enter>. Nota: con el redirector de flujo de entrada << provocamos que todo lo que tecleemos se convierta en la entrada estándar del programa sobre el que estamos realizando la redirección, hasta que no escribimos el código de fin de entrada (en este caso la cadena EOF). Nota 2: el redirector de flujo | provoca que la salida estándar del programa que queda a la izquierda se la entrada estándar del programa que queda a la derecha.

## 2.7 Espectro de frecuencias de una señal

Las señales pueden representarse en el dominio del tiempo o en el dominio de la frecuencia [1]. En el primer dominio, la señal queda especificada por el valor de la señal a lo largo del tiempo. En el segundo, la señal queda definida por un conjunto (en principio infinito) de funciones sinusoidales puras (senos y cosenos) que sumadas entre sí generan dicha señal. A la contribución relativa de cada una de estas señales sinusoidales es a lo que se conoce como *espectro de frecuencias de una señal* o *espectro de Fourier*.

El espectro de Fourier  $S(f)$  de la señal  $s(t)$  es una función compleja y por tanto, tiene una parte real y otra imaginaria. Para tener una idea de la distribución energética del espectro de  $S(f)$  suele representarse su módulo o magnitud que se calcula como:

$$|S(f)| = \sqrt{\text{Re}(S(f))^2 + \text{Im}(S(f))^2}$$

donde  $\text{Re}(S(f))$  es la parte real de  $S(f)$  y  $\text{Im}(S(f))$  es la parte imaginaria de  $S(f)$ .

Para calcular  $|S(f)|$ , siendo  $s(t)$  una señal digitalizada, es posible utilizar la DFT (Discrete Fourier Transform) mediante el programa <http://www.ace.ual.es/~vruiz/docencia/redes/practicas/spectrum> (véase el Apéndice ??).

**Taller 2.3:** Ejecute:

```
ascii2float << EOF > signal.float && spectrum_analyzer signal.float
```

e introduzca una señal con un número par de muestras. A la salida vamos a obtener el módulo del espectro de frecuencias de la señal introducida. Nota: el símbolo > dirige la salida estándar del programa que queda a la izquierda sobre el fichero que aparece a la derecha. Nota 2: los símbolos && representan el operador de ejecución condicional. Significa que, sólo si el comando que queda a la izquierda del operador se ejecuta con éxito, entonces el shell ejecutará el comando que queda a la derecha.

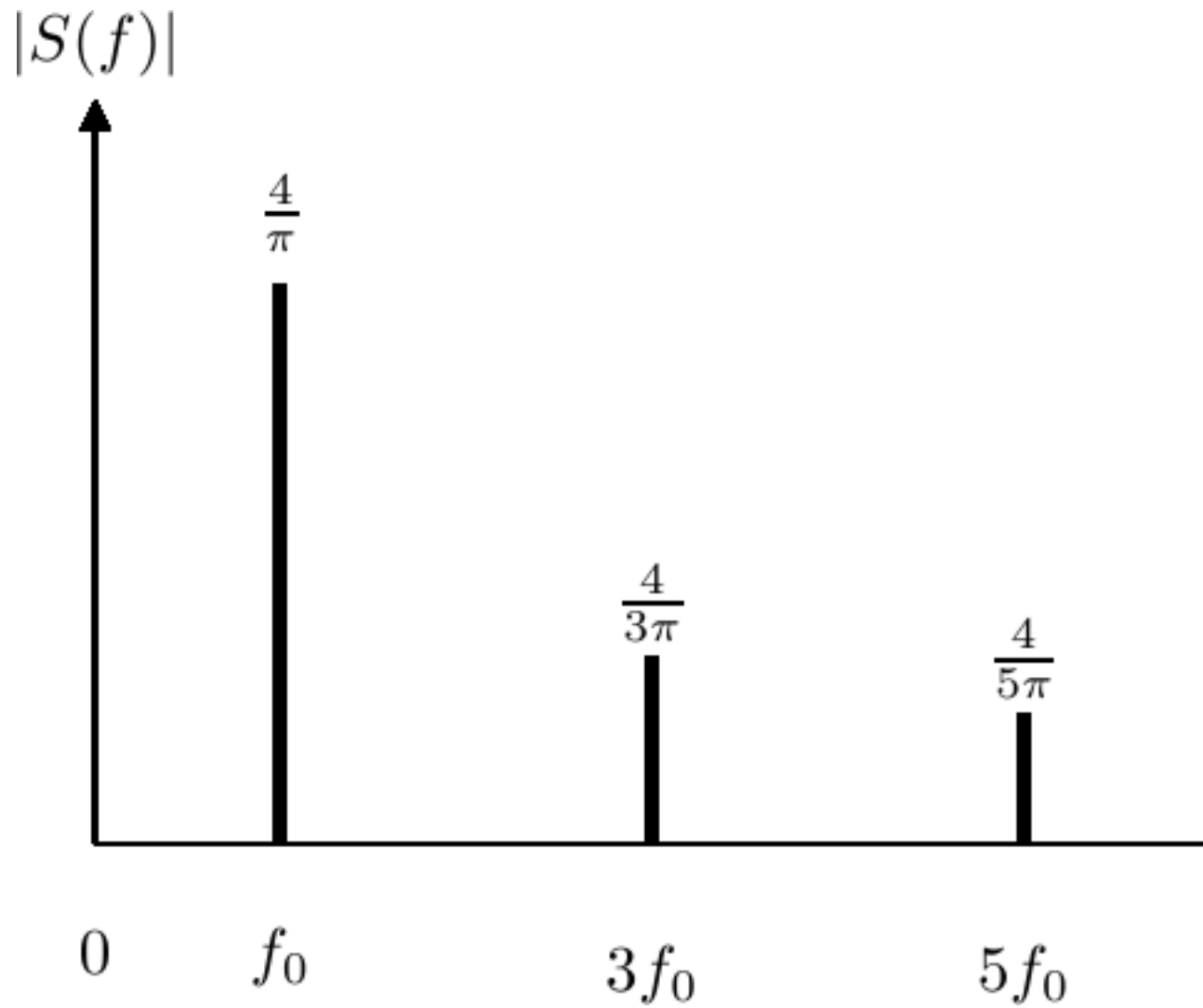


Figure 2.1: Espectro de frecuencias de una señal digital  $s(t)$ .  $1/f_0$  es el tiempo que perdura un bit de información en la señal digital (periodo de un bit).

## 2.8 ¿Cómo es el espectro de frecuencias de una señal digital?

El espectro de una señal digital ideal es discreto y aunque acumula la mayor parte de la energía en las bajas frecuencias también ocupa un *ancho de banda* infinito (véase la Figura 2.1 y la teoría de la asignatura de redes). Esto implica que todos los medios van a filtrar de alguna manera las señales digitales transmitidas.

Podemos visualizar el módulo del espectro de una señal digital utilizando el script [http://www.ace.ual.es/~vruiz/docencia/redes/practicas/draw\\_signal.sh](http://www.ace.ual.es/~vruiz/docencia/redes/practicas/draw_signal.sh) (véase el Apéndice ??).

**Taller 2.4:** Con este programa visualizar el módulo del espectro de la señal "01" es tan simple como:

```
# Visualizando el espectro de la señal 01
rm -f signal.txt # Borramos el fichero
echo "-1.0" >> signal.txt # El valor -1 representa el bit 0
echo "1.0" >> signal.txt # El valor 1 representa el bit 1
ascii2float < signal.txt | sampler 128 > signal.float
spectrum_analyzer signal.float > spectrum.txt
draw_signal.sh spectrum.txt "Espectro de la señal 01"
```

Como puede apreciarse en la ventana que ha debido aparecer, cada coeficiente de Fourier representa el punto de una polilínea. Modifique el script `draw_signal.sh` para que en lugar de unir los puntos del fichero de entrada usando líneas, se pinten barras verticales. Para saber cómo hacer esto exáctamente, ejecute el programa `gnuplot` y escriba en el intérprete `help plot`. Luego pulse la tecla `<q>`, y cuando pregunte `Subtopic of plot:` escriba `with`. Aparecerá un mensaje de ayuda con los argumentos que el comando `plot` acepta para pintar las gráficas. Luego use un editor de ficheros ASCII para modificar el script `draw_signal.sh`. Cuando haya conseguido mostrar una gráfica semejante a la que se muestra en la Figura 2.1 modifique de nuevo el script `draw_signal.sh` para dejarlo con su contenido original.

## 2.9 ¿Cómo afecta la atenuación de las señales a su espectro?

Como ya hemos indicado, los medios de transmisión tienden a atenuar más las altas frecuencias. Esto significa, modelando de una forma bastante pobre el proceso, que las componentes de alta frecuencia de las señales digitales van a ser eliminadas (van a ser cero). El espectro de una señal digital atenuada será semejante al que se presenta en la Figura 2.1, aunque a partir de una determinada banda de frecuencia  $i \cdot f_0$ , no poseerá energía. En otras palabras, si  $s(t)$  es la señal original y  $\hat{s}(t)$  es la

señal filtrada (atenuada), se cumple que:

$$\hat{S}(f) = \begin{cases} S(f) & \text{si } f < i \cdot f_0 \\ 0 & \text{en caso contrario} \end{cases}$$

## 2.10 El filtrado de señales

Un filtro de señales es un sistema físico que modifica la potencia relativa de las distintas componentes de frecuencia de dichas señales [2]. En concreto, un filtro paso bajo es aquel que deja pasar las bajas frecuencias y elimina las altas, a partir de una determinada frecuencia  $f_c$ . En la Figura 2.2 podemos ver la *función de transferencia de un filtro paso bajo*<sup>†</sup>. A la frecuencia  $f_c$  se le conoce como *frecuencia de corte del filtro*.

El programa [http://www.ace.ual.es/~vruiz/docencia/redes/practicas/low\\_pass.filter.c](http://www.ace.ual.es/~vruiz/docencia/redes/practicas/low_pass.filter.c) (Apéndice ??) permite filtrar una señal, dejando pasar la banda de frecuencias más bajas y eliminando el resto.

**Taller 2.5:** Como ejemplo, vamos a eliminar la mitad de las altas frecuencias de la señal "01" cuando la transmitimos a una tasa de bit (bit-rate) de 1 bit/segundo. Utilizaremos una precisión de 128 muestras/bit en nuestro experimento. Para simplificar todo lo posible el proceso ha sido diseñado el script [http://www.ace.ual.es/~vruiz/docencia/redes/practicas/filtrado\\_01.sh](http://www.ace.ual.es/~vruiz/docencia/redes/practicas/filtrado_01.sh) cuyo contenido se muestra a continuación:

```
#!/bin/sh

#
# Filtrado de la señal 01
#

#
# Creamos la señal
#
rm -f 01_signal.txt # Borramos el fichero
echo "-1.0" >> 01_signal.txt # El valor -1 representa el bit 0
echo "1.0" >> 01_signal.txt # El valor 1 representa el bit 1
ascii2float < 01_signal.txt | sampler 128 > 01_signal.float

#
# Visualizamos su espectro
#
spectrum_analyzer 01_signal.float > 01_spectrum.txt
draw_signal.sh 01_spectrum.txt "Espectro de la señal 01"
```

---

<sup>†</sup>Que no es más que la respuesta del filtro paso bajo (la señal generada por el filtro paso bajo) representada en el dominio de la frecuencia cuando a la entrada colocamos una función impulso (una función que siempre vale 0 excepto durante un intervalo de tiempo muy pequeño).

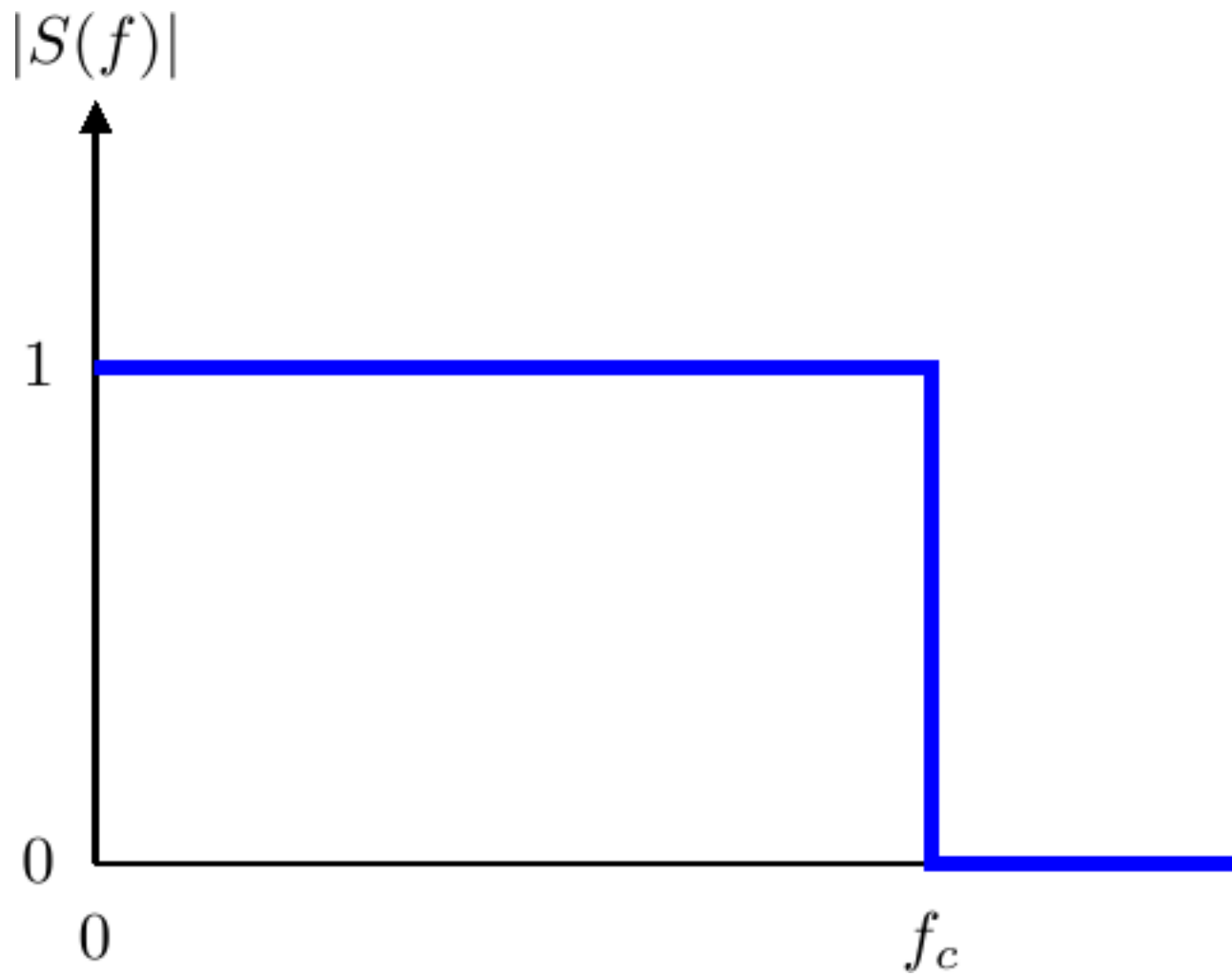


Figure 2.2: Función de transferencia de un filtro paso bajo.

## 2.10. EL FILTRADO DE SEÑALES

---

```
#
# filtramos la señal
#
low_pass_filter 0.5 01_signal.float > 01_filtered.float

#
# Visualizamos la señal filtrada
#
float2ascii < 01_filtered.float > 01_filtered.txt
draw_signal.sh 01_filtered.txt "Señal 01 filtrada"

#
# Visualizamos el espectro de la señal filtrada
#
spectrum_analyzer 01_filtered.float > 01_filtered_spectrum.txt
draw_signal.sh 01_filtered_spectrum.txt "Espectro de la señal 01 filtrada"
```

Ejecute el script `filtrado_01.sh` escribiendo:

```
filtrado_01.sh
```

Si todo ha ido bien debería aparecerle una ventana con una curva semejante a la que se muestra en la Figura 2.3.

**Cuestión 2.4:** ¿Por qué el número de coeficientes de Fourier (número de componentes de frecuencia) es la mitad que el número de muestras introducidas?

**Cuestión 2.5:** ¿Qué controla el parámetro 128 que toma el programa `sampler` en el script que visualiza el espectro de la señal “01” desde el punto de vista del espectro de la señal? Es decir, ¿en qué varía el espectro de la señal cuando variamos dicho parámetro?

**Cuestión 2.6:** Repita el último experimento cuando se transmiten dos bits/segundo y eliminando todas las componentes de frecuencia por encima de 8 Hz (modifique el script `filtrado_01.sh`). ¿La señal recibida tiene ahora una mayor “calidad” (se parece más a la original)? ¿Por qué o por qué no? Explique las modificaciones que ha realizado en el script para llevar a cabo este experimento.

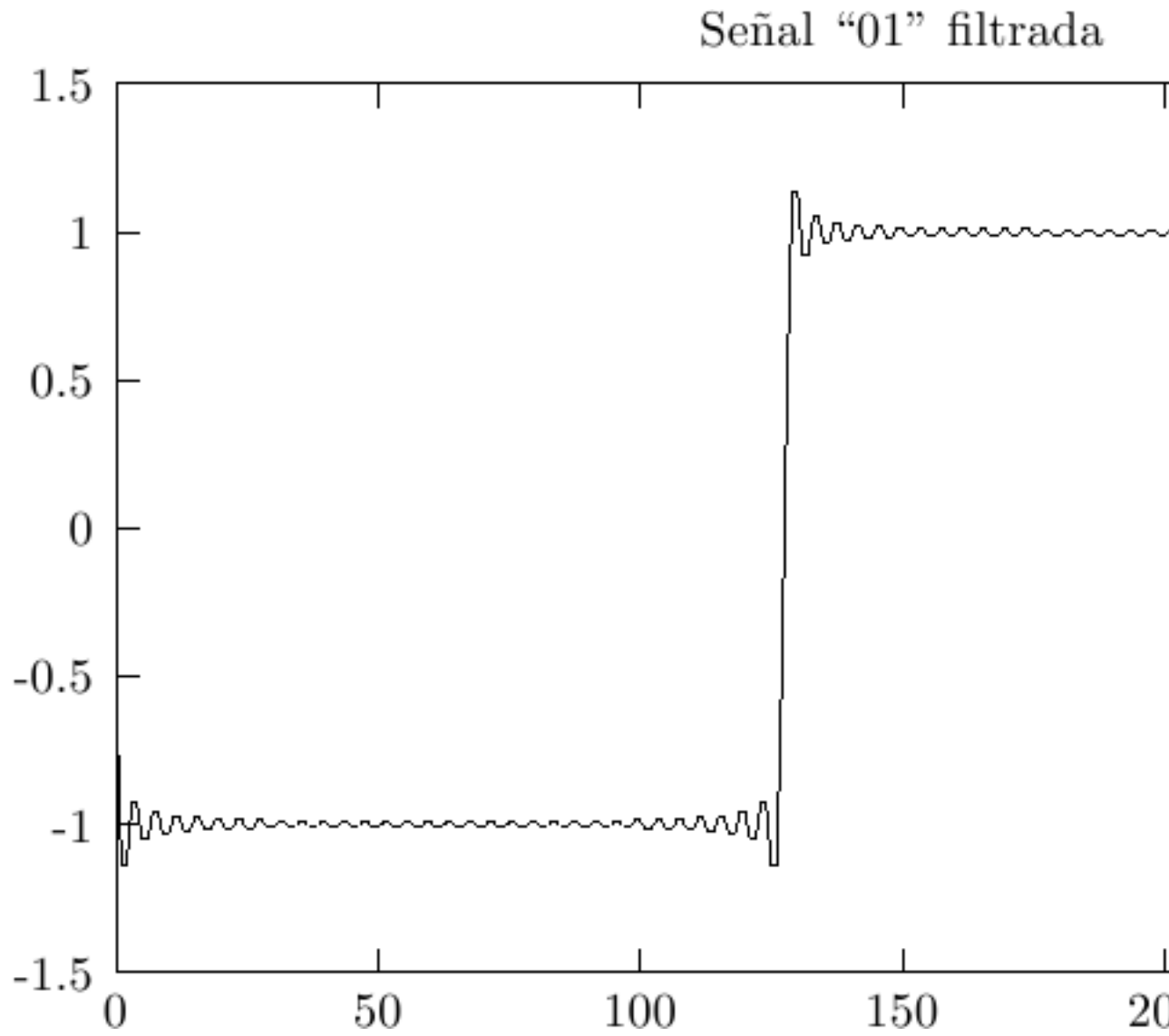


Figure 2.3: La señal "01" transmitida a razón de 1 bit/segundo cuando se han eliminado la mitad de las altas frecuencias.

# Bibliography

- [1] Bhagwandas Pannalal Lathi. *Introducción a la Teoría y Sistemas de Comunicación*. Limusa Noriega Editores, 1994.
- [2] Alan V. Oppenheim and Ronald W. Schaffer. *Discrete-time signal processing*. Prentice Hall, 1999.
- [3] Alan V. Oppenheim, Alan S. Willsky, and S. Hamid Nawab. *Señales y Sistemas (2a edición)*. Prentice Hall, 1997.
- [4] Claude E. Shannon. *The Mathematical Theory of Communication*. University of Illinois Press, 1963.
- [5] William Stallings. *Comunicaciones y Redes de Computadores (7a Edición)*. Prentice Hall, 2004.