



Grado en Ingeniería Informática

PRÁCTICAS DE REDES

Autores:

Alejandro Merino Gómez (alejandromg@ubu.es)

Daniel Sarabia Ortiz (dsarabia@ubu.es)

*Dpto. de Ingeniería Electromecánica
Área de Ingeniería de Sistemas y Automática*

Versión 4.0

Fecha 13/07/2022 17:57

Esta obra está sujeta a la licencia Reconocimiento 4.0 Internacional de Creative Commons. Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by/4.0/>



Índice de las prácticas

Práctica 1.	Transmisión de datos. Señales en el dominio del tiempo y en la frecuencia	1
Práctica 2.	Transmisión de datos. Canales de transmisión	30
Práctica 3.	Transmisión de datos. Codificación, modulación digital y multiplexación	64
Práctica 4.	Instalación de herramientas para emulación y análisis de redes	97
Práctica 5A.	Arquitectura de red	127
Práctica 5B.	Redes en Linux	136
Práctica 6.	Redes LAN, Ethernet y VLAN	161
Práctica 7.	Enrutamiento estático y dinámico	199
Práctica 8.	NAT	211
Práctica 9.	DHCP	223
Práctica 10.	Análisis de TCP con Wireshark	234





Grado en Ingeniería Informática

REDES

PRÁCTICA 1

Transmisión de datos. Señales en el dominio
del tiempo y en la frecuencia

Docentes:

Alejandro Merino

Daniel Sarabia Ortiz

*Dpto. de Ingeniería Electromecánica
Área de Ingeniería de Sistemas y Automática*

Versión 2.1

Fecha 14/02/2022 9:30

*Esta obra está sujeta a la licencia Reconocimiento 4.0 Internacional de
Creative Commons. Para ver una copia de esta licencia, visite
<http://creativecommons.org/licenses/by/4.0/>*



Índice de contenidos

I	INTRODUCCIÓN	3
II	OBJETIVOS	4
III	CONTENIDOS ESPECÍFICOS DEL TEMA	5
1	Transmisión de datos	5
1.1	Modos de transmisión.....	6
1.2	Análisis de señales.....	7
1.3	Datos y señales.....	8
1.4	Factores que afectan al diseño de un sistema de comunicaciones.....	9
1.5	Dominio de la frecuencia	10
2	Instalación de Matlab	14
3	Uso de Matlab y Simulink	15
4	Estudio de señales en tiempo y frecuencia con Simulink	18
4.1	Señal analógica periódica	18
4.2	Varias señales analógicas periódicas y señal constante	22
4.3	Suma de señales periódicas analógicas	23
4.4	Señal digital periódica.....	25
4.5	Señal digital no periódica	26
4.6	Señal de audio.....	28
IV	BIBLIOGRAFÍA	29



I Introducción

El Hombre siempre se ha comunicado de una forma u otra. Con el avance de la tecnología, el proceso de la comunicación ha ido creciendo y mejorando los mecanismos utilizados, hasta llegar a lo que hoy conocemos y utilizamos. Toda comunicación lleva implícita la transmisión de información de un punto a otro pasando por una serie de procesos.

En las redes de comunicaciones y en particular en las redes de computadoras se dice que varios dispositivos (por ejemplo computadoras) están interconectados si pueden intercambiar información. Por tanto, las redes de computadoras están muy ligadas al concepto de transmisión de datos cuyo objetivo es la transmisión de información entre dos o más puntos.

Además, hemos visto que las redes se organizan en una pila de protocolos, en concreto en esta asignatura estudiaremos el modelo TCPI/IP que involucra 5 capas, siendo la inferior la capa o nivel físico, es decir, la que define como se envían los bits mediante señales y los mecanismos para conseguirlo: Mecánicos, eléctricos, de temporización, etc. Esta capa debe especificar, por ejemplo:

- ¿Cuántos voltios se emplean para representar un 1 y cuántos para un 0?
- ¿Cuántos nanosegundos dura un bit?
- ¿La transmisión se debe llevar a cabo en ambas direcciones al mismo tiempo?
- ¿Cómo se establece la conexión inicial y cómo se finaliza cuando ambos lados terminan?
- ¿Cuántos pines tiene un conector de red y para qué se utiliza cada uno?

También especifica cómo transmitir los bits a través de distintos medios físicos: mediante cable de cobre, fibra óptica, cable coaxial, ondas electromagnéticas, etc. Las propiedades de los medios físicos por los que se transmiten las señales determinan el comportamiento y calidad de la red: Rendimiento, latencia, tasa de error, etc.

En definitiva, el nivel físico son los cimientos sobre los que se construye cualquier red de comunicaciones y está estrechamente relacionada con la capa siguiente, la capa de enlace.

Evidentemente la transmisión de datos no se refiere únicamente a los sistemas de computadoras, pero hoy casi todos los sistemas de transmisión de información utilizan en algún punto computadoras o dispositivos electrónicos para facilitar esta tarea y además suelen formar parte de una red.

Por ello, en esta parte de la asignatura, abordaremos el concepto de la transmisión de datos y el modelo general de un sistema de comunicaciones.

Continuaremos con un concepto básico en las transmisiones de datos y por ende de las redes de computadoras, el concepto de señal como elemento que transporta información. Las señales pueden ser de muchos tipos, periódicas, no periódicas, analógicas, digitales, etc. Al final en una red de computadoras las computadoras están interconectadas enviando y recibiendo señales que son las que transportan la información.

Normalmente, estamos habituados a ver y describir las señales en función del tiempo, lo que se denomina descripción en el dominio del tiempo. Sin embargo, existe una descripción alternativa y complementaria que es la descripción en el dominio de la frecuencia. Estudiar una señal como función de la frecuencia nos permite caracterizar una señal de otra manera, fijándonos en propiedades que en una descripción temporal permanecen "ocultas" o al menos no son fácilmente reconocibles. Este tipo de propiedades, por ejemplo, el ancho de banda, son muy importantes en el estudio de los sistemas de transmisión de datos y condicionan el tipo de comunicación a realizar, por ejemplo si se usa una señal analógica o directamente una señal digital, si varios dispositivos pueden conectarse a la vez, si se va a utilizar una transmisión inalámbrica o a través de un cable de cobre o de fibra óptica, etc.



La herramienta matemática que permite hacer este estudio se denomina análisis de Fourier.

Finalmente, Matlab es una potente herramienta de cálculo matemático (y también de programación) que en principio resuelve problemas matemáticos de manera numérica. Matlab incorpora un entorno visual llamado **Simulink** que permite modelar, simular y analizar sistemas dinámicos (aquellos cuyas salidas varían con el tiempo) mediante **modelos** formados por diagramas de bloques interconectados.

Usaremos la toolbox de Simulink "**DSP System Toolbox**" para estudiar distintos tipos de señales tanto analógicas como digitales presentes en cualquier sistema de transmisión de datos. El estudio se realizará tanto en el dominio temporal como en el de la frecuencia obteniendo los correspondientes espectros de frecuencias. **Esta toolbox de Simulink también hay que instalarla junto a Matlab/Simulink.**

Todos los alumnos de la Universidad de Burgos disponen de licencia académica para utilizar Matlab de forma legal y trabajaremos con la versión **R2019b de Matlab**.

Matlab/Simulink poseen una potente ayuda, donde se describe el comportamiento de los bloques, su configuración y ejemplos de uso, se recomienda usarla ante cualquier duda.

Normalmente el guion de cada práctica incluirá:

- Una *descripción teórica* de los conceptos a estudiar.
- Un *esquema del modelo en Simulink* a realizar con las conexiones necesarias entre bloques.
- El *nombre de los bloques* que se van a usar.
- En qué *paleta* se encuentra cada bloque.
- Qué *parámetros* del bloque hay que configurar.
- Qué resultados se deben obtener y preguntas sobre los resultados relacionados con los conceptos teóricos a estudiar.

II Objetivos

- Conocer los principios de la transmisión de datos y el modelo general de un sistema de comunicaciones.
- Ser capaz de distinguir distintos tipos de señales, analógicas, digitales, periódicas, no periódicas y sus características en el tiempo, amplitud, frecuencia, fase, periodo.
- Clarificar la diferencia entre datos y señales. Las señales como representación de datos (información).
- Tener una noción básica del significado de la representación de una señal en el dominio de la frecuencia. Espectro de frecuencias de una señal.
- Ser capaz de distinguir distintos tipos de señales, analógicas, digitales, periódicas, no periódicas y sus características en la frecuencia. Ancho de banda de una señal.
- Aprender el uso básico de programación de modelos en Simulink.
- Ser capaz de generar señales analógicas y digitales en Simulink.
- Practicar y afianzar la mayoría de los conceptos anteriores usando Simulink y en particular la "DSP System Toolbox" de Simulink.



III Contenidos específicos del tema

1 Transmisión de datos

En las redes de comunicaciones se dice que varios dispositivos están interconectados si pueden intercambiar información. Por ello es interesante definir los siguientes conceptos.

La **transmisión de datos** consiste en el movimiento de información codificada, de un punto a uno o más puntos, mediante señales eléctricas o electromagnéticas.

Comunicación. Actividad asociada con el intercambio o distribución de información.

Información, concepto ambiguo:

- Según la RAE: 1. Acción y efecto de informar, 2. Oficina donde se informa sobre algo, 3. Averiguación jurídica y legal de un hecho o delito, ...
 Informar: 1. Enterar, dar noticia de algo, 2. Dicho de una persona o de un organismo, ...
- Conjunto organizado de datos procesados, que constituyen un mensaje que cambia el estado de conocimiento del sujeto o sistema que recibe dicho mensaje.
- La información es lo que se distribuye o intercambia con la comunicación.

Para que haya comunicación tiene que haber uno o varios emisores que generen los mensajes, y uno o varios receptores que los reciban y los interpreten.

- Sin receptores no hay comunicación.
- Es necesario interpretar los mensajes, porque si no, lo que nos llega no proporciona información, sino solo ruido.

Telecomunicación es la acción de comunicarse a distancia. Proviene del griego *tele* y del latín *comunicare*, es decir, "llevar información de un lugar a otro".

En particular, internet y las redes de computadoras, que son el tipo de redes que se van a estudiar en esta asignatura, no son más que un conjunto de dispositivos interconectados que intercambian información entre ellos. Por ejemplo, dos aplicaciones de gestión de correo electrónico ejecutándose en computadoras distintas en la que una recibe un email enviado por la otra. Otro ejemplo puede ser el acceso a una base de datos, acceso a una página web, comunicación digital, es decir, mantener una conversación de voz vía IP mediante dos aplicaciones (Skype, o similar) que se ejecutan en computadoras diferentes, etc.

El uso de las redes y el tipo de comunicación que se puede llevar a cabo es inmenso, pero en esta práctica y las siguientes nos vamos a centrar en los aspectos teóricos de la transmisión de datos y en cómo se realiza la comunicación en el nivel más bajo, es decir a nivel de señales eléctricas y electromagnéticas que son las responsables en último término de transmitir la información.

Un sistema de comunicaciones permite intercambiar información entre dos entidades y un **modelo completamente general de un sistema de comunicaciones** está formado por una fuente, un transmisor un sistema de transmisión o canal, un receptor y un destino, ver Figura 1:

- **Fuente.** El dispositivo que genera los datos a transmitir. Ejemplos: teléfono o un computador personal
- **Transmisor.** Normalmente los datos generados por la fuente no se transmiten directamente. El transmisor transforma y codifica la información, generando señales eléctricas o electromagnéticas susceptibles de ser transmitidas a través de algún sistema de transmisión. Por ejemplo, un módem convierte las cadenas de



bits generadas por un computador y las transforma en señales analógicas que pueden ser transmitidas a través de la red de telefonía.

- **Sistema de transmisión o canal.** Conecta el sistema origen con el sistema destino receptor. Puede ser cualquier medio de transmisión (cable de cobre, fibra óptica, cable coaxial, aire, etc.) y lo imponen los equipos que se conectan (tecnológicamente hablando). Por ejemplo, dos computadoras pueden conectarse mediante cable si disponen de tarjeta de red, pero también pueden comunicarse inalámbricamente, a través del aire, si disponen de tarjeta de red inalámbrica apropiada.
- **Receptor.** El receptor acepta la señal proveniente del sistema de transmisión y la transforma de tal manera que pueda ser manejada por el dispositivo de destino. Por ejemplo, un módem captará la señal analógica de la red o línea de transmisión y la convertirá en una cadena de bits.
- **Destino.** Toma los datos del receptor.

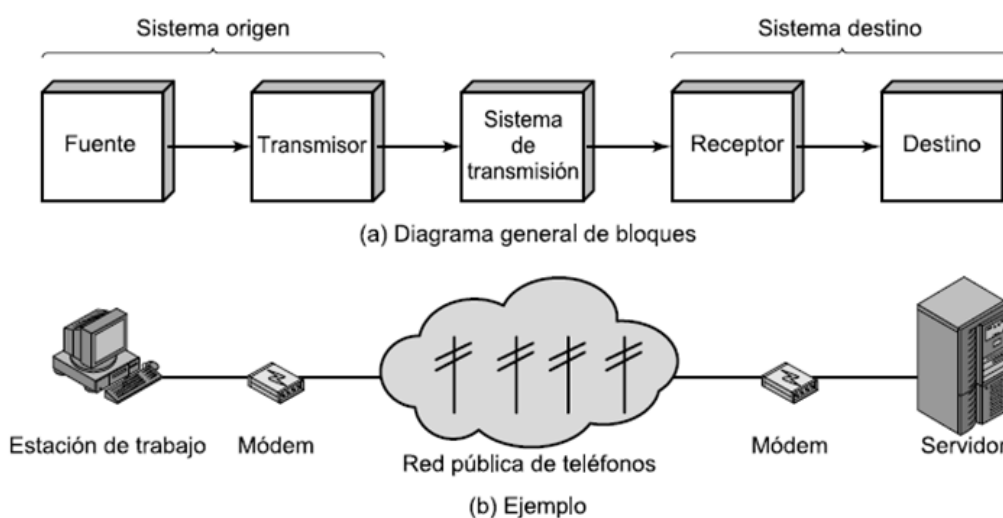


Figura 1. Modelo típico para las comunicaciones.

1.1 Modos de transmisión

Determina cómo es la transmisión entre dos dispositivos en cuanto al sentido de la transmisión:

- **Simplex.** La transmisión se realiza solo en un sentido, sin posibilidad de hacerlo en el opuesto. Una de las estaciones actúa siempre como emisor y la otra como receptor.
- **Half-duplex.** La transmisión se lleva a cabo alternativamente en uno u otro sentido, ambas estaciones pueden transmitir, pero no simultáneamente.
- **Full-duplex.** La transmisión es simultánea e independiente en ambos sentidos. Ambas estaciones pueden transmitir y recibir simultáneamente.

En full-dúplex el medio transporta señales en ambos sentidos al mismo tiempo. Tiene una eficiencia muy alta, pero exige unos terminales más complicados. La capacidad de transmitir en modo full-dúplex está condicionado por:

- Que el medio físico sea capaz de transmitir en ambos sentidos, por ejemplo empleo de frecuencias separadas (multiplexación en frecuencia) o empleo de cables separados.
- Que el sistema de transmisión sea capaz de enviar y recibir a la vez.
- El protocolo o norma de comunicación empleado por los equipos terminales.



1.2 Análisis de señales

Una señal es un fenómeno físico en el cual una o varias de sus características pueden variar para representar información. Se pueden clasificar de manera diferente las señales, por el número de valores que pueden tomar a lo largo del tiempo, por su periodicidad, por el rango (banda) de frecuencias que ocupa, etc.

Toda señal, considerada como función del tiempo, puede ser tanto analógica como digital:

- **Señales analógicas:** están representadas por funciones que pueden tomar un número infinito de valores en cualquier intervalo de tiempo. *Por ejemplo*, una onda electromagnética es una señal analógica que puede transmitirse a través de diferentes medios físicos, como aire, agua, vacío, un medio conductor, etc.
- **Señales digitales:** Son aquellas señales que están representadas por funciones que pueden tomar un número finito de valores en cualquier intervalo de tiempo. Su valor se mantiene constante durante un determinado intervalo de tiempo, tras el cual la señal cambia a otro valor constante. *Por ejemplo*, una secuencia de pulsos de tensión es una señal digital que se puede transmitir solo por un medio conductor.

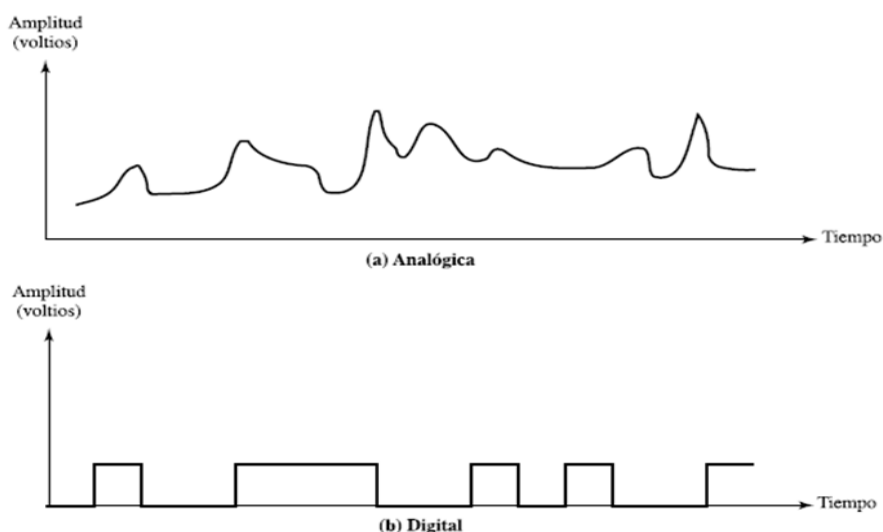


Figura 2. Señales a) analógica y b) digital.

También una señal puede ser periódica o no:

- Una señal **periódica** se caracteriza por tener un patrón que se repite a lo largo del tiempo. $s(t)$ es una señal periódica si es una función del tiempo que cumple $s(t) = s(t+T)$, donde t es el tiempo y T el período que indica cada cuánto tiempo se repite la señal y define la duración de un ciclo completo de la señal. El periodo se mide en segundos (s).
- Una señal que no cumpla la anterior definición es una señal **no periódica**.

La señal seno (Figura 3) es un ejemplo típico de señal periódica (y además analógica):

$$s(t) = A \operatorname{sen}(2\pi f_0 t + \phi) \quad (1)$$

Se caracteriza mediante tres parámetros:

- La amplitud A , es el valor máximo de la señal en el tiempo.
- La frecuencia f_0 que es el inverso del periodo T , es decir, $f = 1/T$ y se define como el número de ciclos completos que tienen lugar en una unidad de tiempo y sus unidades son hercios (Hz o s^{-1}). A veces se especifica la frecuencia angular w definida como $w=2\pi f$ expresada en rad/s.



- La fase ϕ , es una medida de la posición relativa de la señal dentro de un periodo de la misma y se mide en radianes (rad).

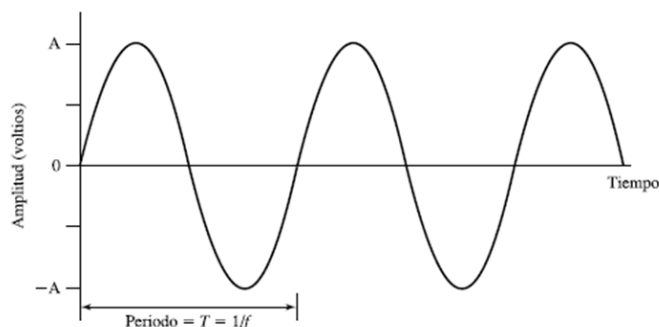


Figura 3. Señal seno de amplitud A, periodo T y fase 0 rad. Señal periódica y analógica.

Otro ejemplo de función periódica puede ser una onda cuadrada, ver Figura 4, de amplitud A y periodo T:

$$s(t) = \begin{cases} A, & (n)\frac{T}{2} \leq t < (n+1)\frac{T}{2} \\ -A, & (n+1)\frac{T}{2} \leq t < (n+2)\frac{T}{2} \end{cases} \quad \forall n = 0,1,2,3, \dots \quad (2)$$

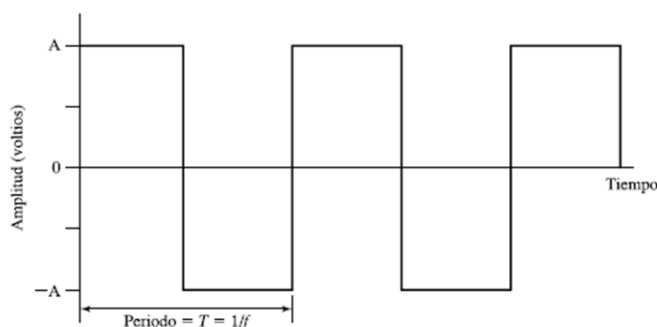


Figura 4. Onda cuadrada de periodo T. Señal periódica y digital.

1.3 Datos y señales

Dato es cualquier entidad capaz de transportar información y pueden ser analógicos o digitales:

- **Datos analógicos** pueden tomar valores en un intervalo continuo. Por ejemplo, el vídeo y la voz son valores de intensidad que varían continuamente.
- **Datos digitales** toman valores discretos a lo largo del tiempo. Por ejemplo, cadenas de texto, números enteros, pulsaciones del teclado, etc.

Las señales son representaciones eléctricas o electromagnéticas de los datos, por lo tanto, la transmisión de datos es **la comunicación de datos** mediante **la propagación** y el procesamiento de señales.

Los datos analógicos o digitales pueden ser representados tanto por señales analógicas como digitales, ver Figura 5. En esta asignatura, y en concreto en las prácticas siguientes, solo nos centraremos en la transmisión de datos digitales mediante señales analógicas, llamado modulación digital y transmisión de datos digitales mediante señales digitales.



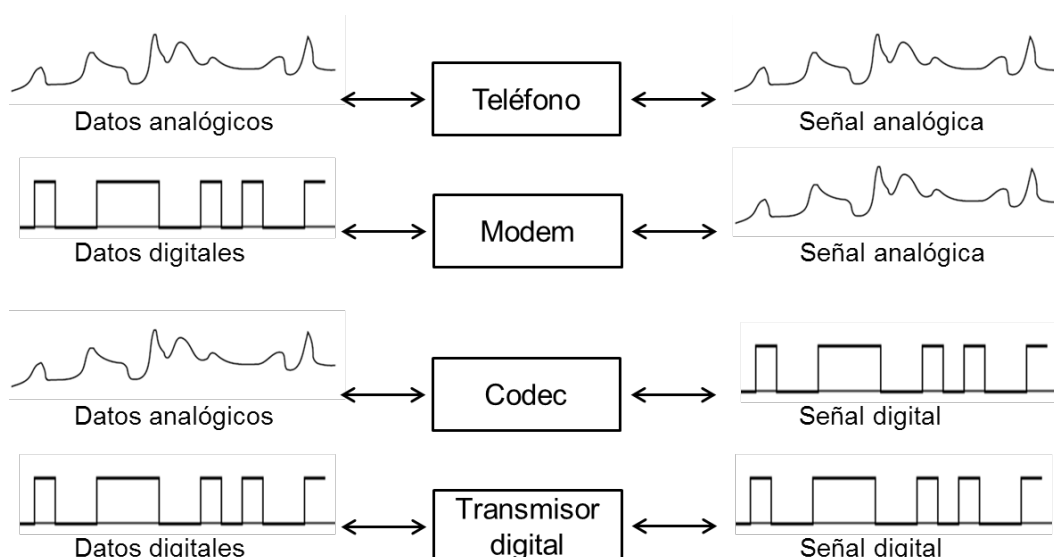


Figura 5. Formas en que los datos (información) pueden ser transmitidos mediante señales analógicas y digitales.

¿Dónde reside la información en una señal?

- **Señales analógicas:** La información va contenida en la propia forma de la onda en forma de amplitudes, frecuencias y fases. *Por ejemplo*, en una señal analógica que representa una señal sonora, la amplitud se corresponde con un sonido más o menos alto y la frecuencia se corresponde con un sonido más o menos agudo. Ver Figura 2 a).
- **Señales digitales:** La información está contenida en los pulsos codificados, es decir, en la secuencia concreta de ceros y unos transmitida y no en la forma de la onda. Por ejemplo, transmisión entre dos computadoras de la letra N codificada mediante código ASCII. Secuencia de ceros y unos: 01001110. Ver Figura 2 b).

1.4 Factores que afectan al diseño de un sistema de comunicaciones

Cuando se plantea el diseño de un sistema de comunicaciones hay una serie de aspectos que influyen:

1) Limitaciones tecnológicas

- Disponibilidad de software y hardware. Hay situaciones en las que se conoce un diseño óptimo para un determinado sistema, pero que ese diseño no se puede llevar a la práctica porque todavía no se ha desarrollado la tecnología o no es lo suficientemente rápida para implementarlo.
- Consumo de potencia. El compromiso entre coste y consumo siempre es un factor a tener en cuenta por los ingenieros. En cierta forma, este punto es un caso particular del apartado anterior.
- Tamaño de los componentes electrónicos. El tamaño de los componentes electrónicos es muy pequeño, pero también lo es el sitio donde deben ser colocados y cuanto más complejo es un circuito más aumenta su tamaño.



- 2) Estándares y regulaciones gubernamentales.** En comunicaciones es imprescindible la existencia de estándares que definan el funcionamiento de los equipos para permitir una correcta interoperación entre equipos procedentes de fabricantes diferentes. Además de las normas dictadas por los organismos de los diferentes países hay que tener en cuenta otro tipo de normas que son redactadas por los gobiernos.
- 3) Realidades comerciales.** A pesar de los esfuerzos de los ingenieros por el desarrollo de dispositivos cada vez más sofisticados y eficientes, al realidad dicta que el producto final es adquirido por sus características menos relevantes, si nos vamos al ejemplo de los teléfonos móviles vemos que en ocasiones se compran más por el tiempo que permiten hablar, su color o la promoción de ese momento que no por su calidad en la transmisión.

1.5 Dominio de la frecuencia

Es habitual representar las señales en el dominio temporal donde lo que normalmente representamos es la amplitud de una señal a lo largo del tiempo. Sin embargo, hay determinadas propiedades que no podemos observar en el dominio temporal pero que sí se pueden ver en el dominio de la frecuencia.

Ambas representaciones implican una misma realidad física, entonces ¿por qué usar el dominio de la frecuencia?

Para poder entender la interacción entre datos y la velocidad de transmisión, el tipo de modulación, la forma de las señales, ancho de banda, etc.

El método matemático más utilizado para el estudio en el dominio de la frecuencia de las señales es el "Análisis de Fourier" y a su vez hay dos disciplinas:

- Las **series de Fourier** que se utilizan para estudiar las señales periódicas.
- La **transformada de Fourier** que se usa para estudiar tanto señales periódicas como no periódicas.

Por tanto, para cada señal en el dominio del tiempo $s(t)$ existe una función $S(f)$ en el dominio de la frecuencia, que especifica las amplitudes de las frecuencias constitutivas de la señal.

Interpretación inicial

Imaginar que se quiere cubrir la superficie del cuadrado (S_C) de la Figura 6 con superficies circulares circunscritas y tangentes a todas las líneas.

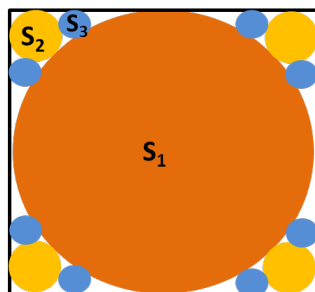


Figura 6. Reconstrucción de la superficie cuadrada mediante superficies circulares.

El resultado es que el área del cuadrado se puede poner como la suma de infinitas superficies circulares de diferentes tamaños, es decir cubriendo mas o menos superficie y por tanto más o menos importante en la superficie final, y además cada una de ellas con una frecuencia de aparición diferente, es decir,

$$S_C = S_1 \times 1 + S_2 \times 4 + S_3 \times 8 + \dots \quad (3)$$



Donde S_1 cubre la mayor parte de la superficie del cuadrado y aparece 1 vez, S_2 cubre menos parte de la superficie del cuadrado y aparece 4 veces y S_3 cubre menos parte y aparece 8 veces, etc. Claramente desde el punto de vista de cubrir la superficie del cuadrado la superficie 1 es más importante que la 3.

Series de Fourier

Una señal periódica en el tiempo $s(t)$ se puede representar como una suma de infinitas señales seno o coseno de diferente amplitud (a_n y b_n) y frecuencia f_n :

$$s(t) = \frac{a_0}{2} + \sum_{n=1}^{\infty} (a_n \cos(2\pi f_n t) + b_n \sin(2\pi f_n t)) \quad (4)$$

Donde f_1, f_2, f_3 , etc. son las frecuencias de oscilación de cada señal seno o coseno que forma la señal original $s(t)$. La amplitud de cada señal seno o coseno (a_n y b_n) indica lo importante que es esa señal en la señal original $s(t)$.

Las amplitudes a_n y b_n pueden representarse en función de cada frecuencia f_n a la que tienen lugar dando lugar a una representación en el dominio de la frecuencia, como se ve en la Figura 7.

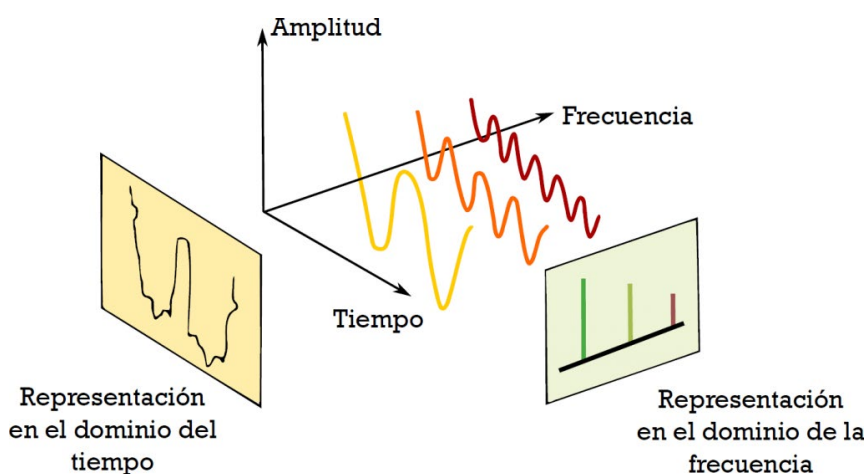


Figura 7. Descomposición de una señal $s(t)$ en suma de señales de diferente amplitud y frecuencia.

Transformada de Fourier

Se define (de manera muy informal) la transformada de Fourier de la función $s(t)$ definida en el tiempo:

$$S(w) = \int_{-\infty}^{+\infty} s(t) e^{-iwt} dt \quad (5)$$

Con w la frecuencia angular en rad/s y t el tiempo en segundos. El resultado es una función $S(w)$ cuya variable independiente es la frecuencia angular w y que además es una función compleja (puede tener parte real y parte imaginaria).

En vez de obtener la transformada de Fourier en función de la variable independiente frecuencia angular w , es más habitual expresarla en función de la variable independiente frecuencia f en Hz, es decir $S(f)$. Siendo necesario hacer cambios de variables en el procedimiento de cálculo matemático, recordad que $w = 2\pi f$, con w en rad/s y f en Hz.

Por ejemplo, la transformada de Fourier de la señal seno definida en (1) es:

$$S(f) = i \frac{A}{2} (\delta(f + f_0) - \delta(f - f_0)) \quad (6)$$



Nota 1. En este caso, la señal $S(f)$ es una función compleja (solo tiene parte imaginaria i).

Nota 2. La función $\delta(f+f_0)$ se denomina en matemáticas delta de Dirac (o función impulso) y significa que vale ∞ en $f = -f_0$ y 0 en el resto de valores de f . En el caso concreto de (6), la función vale $iA/2$ en $f = -f_0$.

Función	$s(t)$	$S(f)$
Seno	$s(t) = A \sin(2\pi f_0 t + \phi)$	$S(f) = i \frac{A}{2} (\delta(f + f_0) - \delta(f - f_0))$
Constante	$s(t) = A$	$S(f) = A \delta(f)$
Pulso		

Figura 8. Tabla con algunas transformadas de Fourier de varias señales típicas.

Espectro de frecuencias

Se define el espectro de frecuencia como el conjunto de frecuencias que existen en una señal. Como la señal $S(f)$ suele ser una función compleja, es decir que puede tener parte real y parte imaginaria, el espectro suele representar el módulo al cuadrado de $S(f)$, es decir, $|S(f)|^2$. Suele denominarse densidad de potencia espectral y su amplitud se da en watsios. La amplitud en el espectro indica lo predominante que es cada frecuencia f en la señal original en el tiempo $s(t)$. Por ejemplo, en la Figura 9 se muestra el espectro de frecuencias de una señal seno y de un pulso de tensión. En principio, desde un punto de vista matemático, el espectro se extiende para frecuencias negativas y positivas, pero desde un punto de vista físico, **solo tiene sentido la parte de frecuencias positivas** y es el que se va a analizar siempre.

El espectro de frecuencias se calcula matemáticamente (analíticamente) mediante la transformada de Fourier y numéricamente mediante algoritmos como FFT (Fast Fourier Transform) que es el algoritmo que se usa en Matlab.

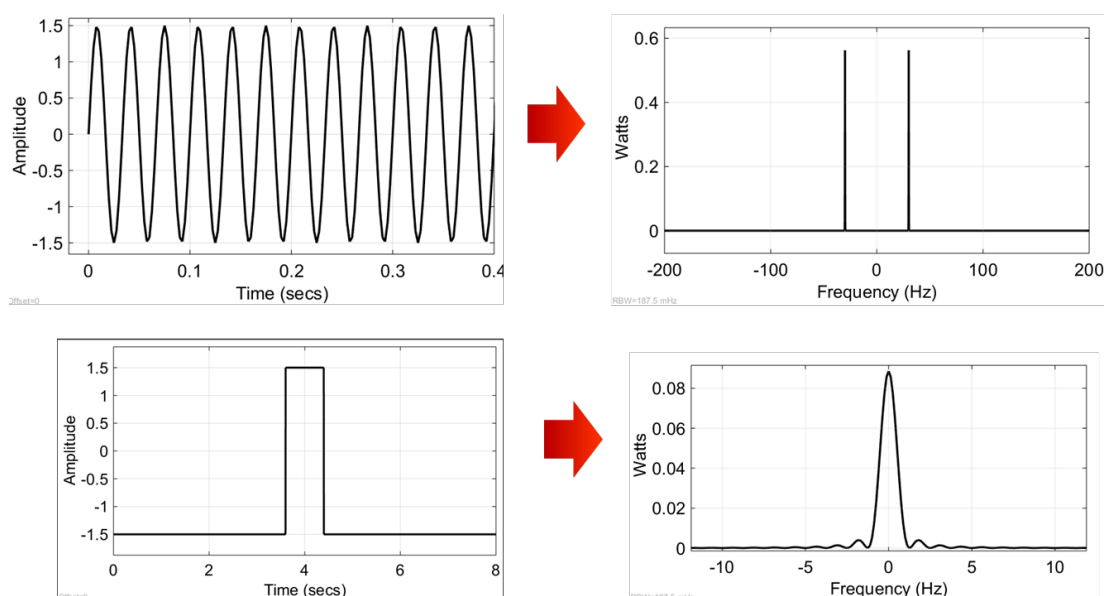


Figura 9. a) Señal seno y a la derecha su espectro de frecuencias. b) Señal pulso y su espectro de frecuencias a la derecha.



El **ancho de banda de una señal** da una idea de la extensión del contenido espectral significativo de la señal para frecuencias positivas, se mide en Hz, ver Figura 10:

- *Ancho de banda absoluto*, se corresponde con la anchura del espectro de frecuencias completo.
- *Ancho de banda relativo*, anchura del espectro de frecuencias donde se concentra la mayor parte de la energía de la señal. Es una **decisión completamente arbitraria**. En las prácticas vamos a calcularla como aquella franja de frecuencias cuyas componentes tengan más de un porcentaje fijo de potencia respecto de la componente con máxima potencia.

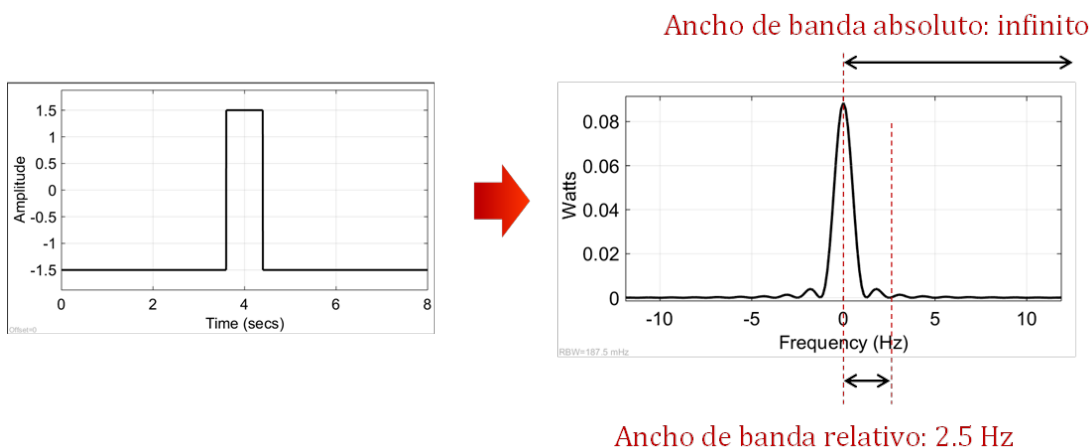


Figura 10. Espectro de frecuencias de una señal pulso y el ancho de banda relativo y absoluto.

Las señales pueden clasificarse según su contenido espectral en, ver Figura 11:

- **Señales limitadas en banda.** La anchura del espectro es *finita*. El ancho de banda coincide con el ancho de banda absoluto.
- **Señales no limitadas en banda.** La anchura del espectro es *infinita*, es decir, el ancho de banda absoluto es infinito, pero la mayor parte de la energía de la señal se concentra en una banda de frecuencias relativamente estrecha. Esta banda es precisamente el ancho de banda relativo.

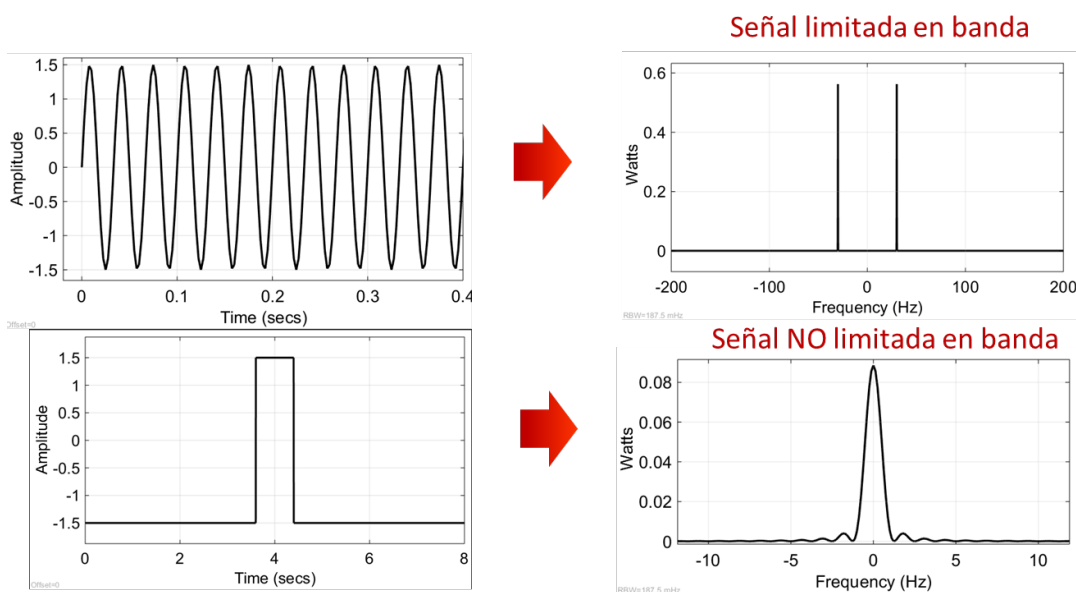


Figura 11. Señal limitada en banda y no limitada en banda.



Las señales también pueden clasificarse según donde se ubica su ancho de banda, ver Figura 12:

- **Señales banda base.** Son aquellas señales que presentan componentes de frecuencia entre 0 y un valor máximo f_m . Por ejemplo, las señales digitales
- **Señales pasa banda.** Son aquellas señales que ocupan un rango más alto de frecuencias y presentan componentes de frecuencia entre dos valores distintos de cero f_1 y f_2 .

Dando lugar a la **transmisión banda base**, cuando se transmiten señales banda base y **transmisión pasa banda** cuando se transmiten señales pasa banda.

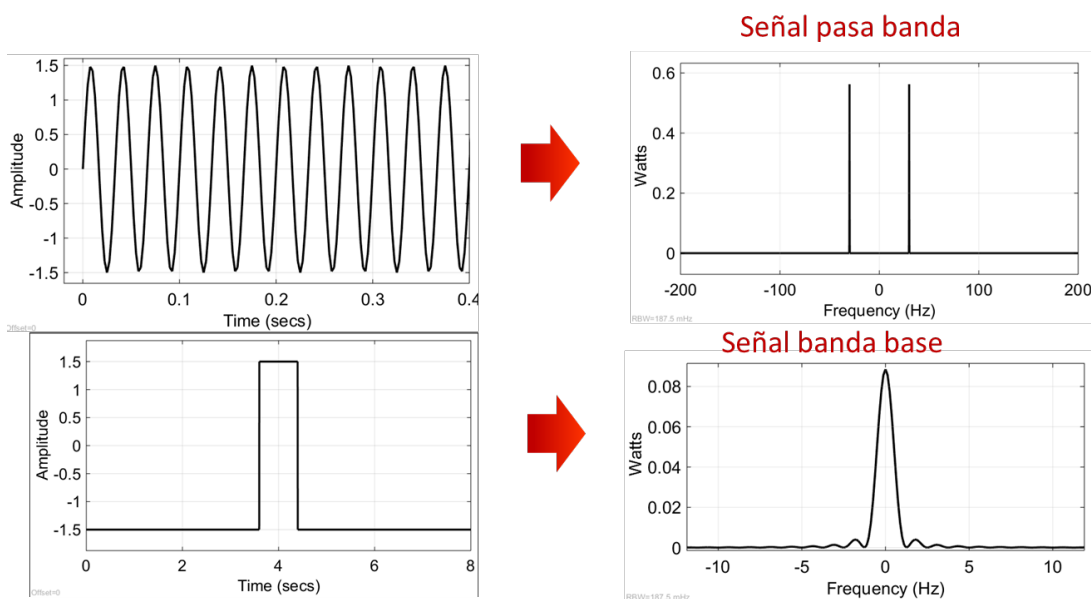


Figura 12. Señales pasa banda y banda base.

2 Instalación de Matlab

Puedes descargarte el manual de instrucciones de descarga e instalación de Matlab versión académica para la Universidad de Burgos en el siguiente enlace:

<https://www.ubu.es/servicio-de-informatica-y-comunicaciones/catalogo-de-servicios/software-tu-disposicion/software-disposicion-de-la-comunidad-universitaria/matlab>

Selecciona "Matlab para estudiantes" en el apartado ¿Cómo descargarlo? Ver Figura 13 y sigue las instrucciones del manual. Necesitarás autenticarte con tu usuario y password para acceder al manual.

Siguiendo las instrucciones del manual en algún momento se podrá elegir la versión de Matlab para descargar, elegir la **versión R2019b de Matlab**. También instalar la toolbox "**DSP System Toolbox**" entre todas las toolboxes que propone Matlab.

Importante: las prácticas se han probado en dicha versión y por tanto se corregirán en esa versión. Versiones antiguas (o más nuevas) de Matlab pueden dar resultados no esperados.



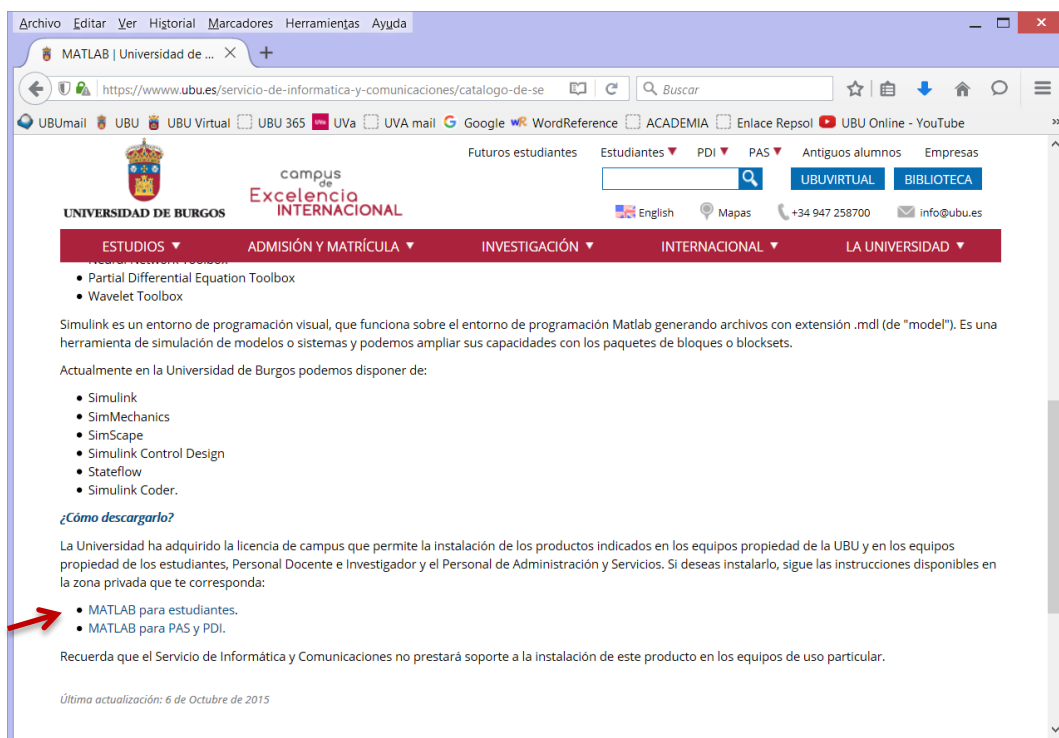


Figura 13. Página web de la UBU con la información necesaria para descargar Matlab.

3 Uso de Matlab y Simulink

Simulink es una extensión de Matlab para Modelar, Simular y Analizar sistemas dinámicos (aquellos cuyas salidas varían con el tiempo). Esto se hace mediante modelos formados por diagramas de bloques e interfaces gráficas de usuarios (GUIs).

Para el desarrollo de esta práctica y de varias siguientes usaremos bloques de la Toolbox de Simulink "DSP System Toolbox" y bloques estándares del propio Simulink.

Lo primero, al arrancar Matlab, es seleccionar el directorio actual de trabajo (Current Folder), es decir, el directorio en el que vamos a guardar nuestros ficheros, ver Figura 14. Esto se debe realizar siempre antes de comenzar a trabajar con Matlab/Simulink.

A continuación abriremos la herramienta Simulink pinchando en el icono "Simulink Library". Esto nos abre la paleta de Simulink en la que se encuentran los bloques estándares de Simulink y los bloques asociados a las toolboxes instaladas, en concreto los de la "DSP System Toolbox" ver Figura 15.

Para crear un modelo en blanco de Simulink o para cargar uno existente pinchar en los iconos correspondientes "New Model", ver Figura 15.



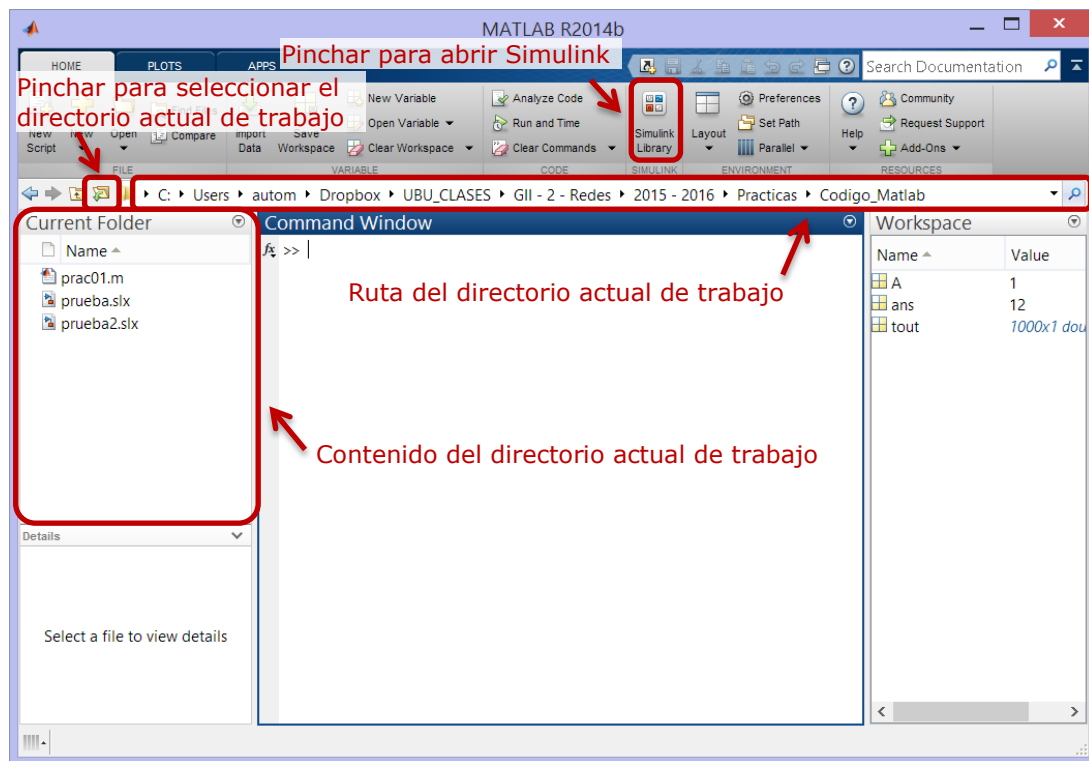


Figura 14. Aspecto de la interfaz de Matlab y selección del directorio actual de trabajo.

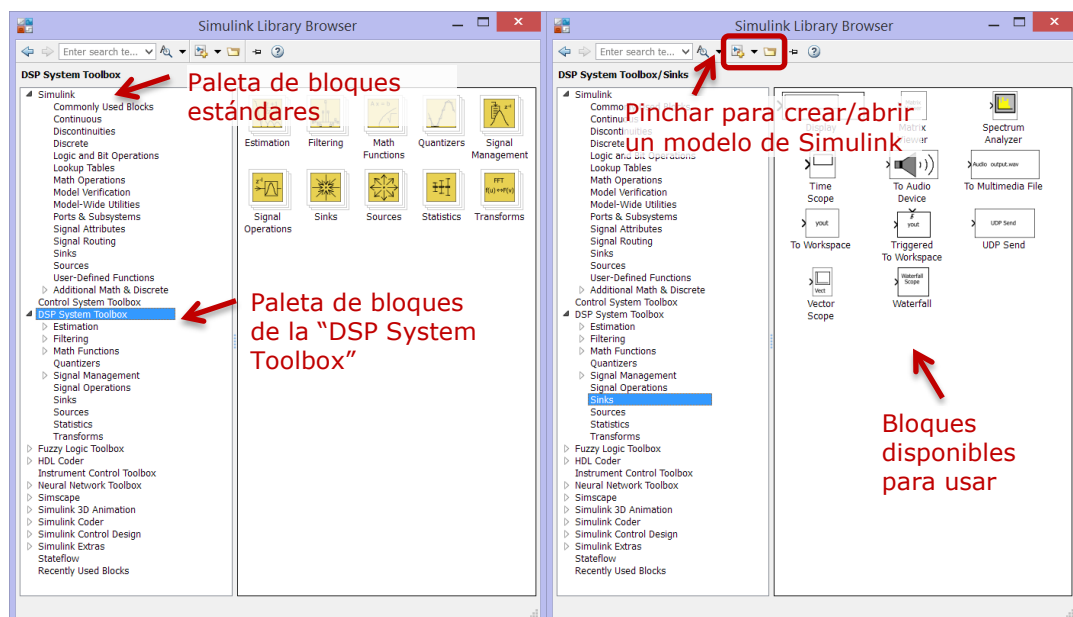


Figura 15. Paleta de Simulink.

Una vez creado un nuevo modelo se abrirá una ventana asociada a dicho modelo, ver Figura 16, que será donde programemos de manera visual nuestras prácticas simplemente arrastrando los bloques de la paleta de Simulink y conectándolos entre sí. Vamos a crear y grabar el modelo relacionado con el apartado 4.1 de esta práctica, por tanto grabamos el modelo con el nombre "Prac01_Signals_4_1.slx".

Importante: Tanto en Matlab como en Simulink el nombre de los ficheros NO PUEDE tener espacios en blanco ni empezar por un número. Se recomienda usar solo caracteres alfanuméricos estándares, evitando acentos, ñ, etc. La extensión de los ficheros de Matlab es ".m" y la de los modelos de Simulink es ".slx" (antiguamente era ".mdl").



Fijaos en la Figura 16 dónde están ubicados los iconos relacionados con el arranque y parada de la simulación (Run y Stop), el tiempo máximo de simulación y el porcentaje completado de la simulación, pues los usaremos más adelante.

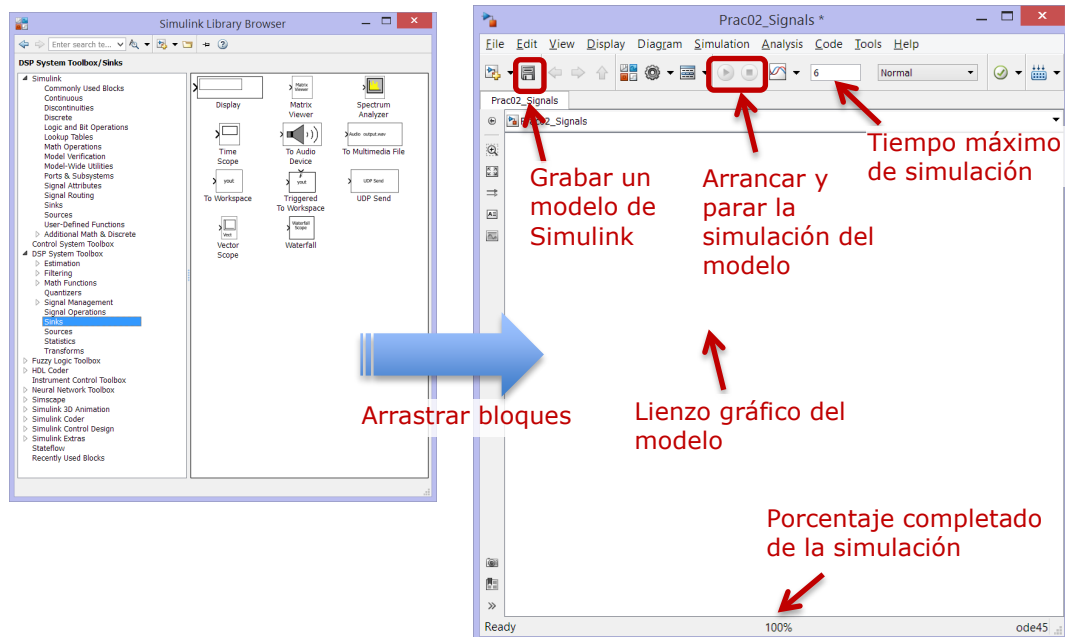


Figura 16. Modelo de Simulink guardado con el nombre "Prac01_Signals_4_1.slx".

Es conveniente a la hora de hacer prácticas incluir una pequeña descripción o al menos la relación del modelo con el apartado de la práctica para saber qué es lo que hace cada parte. Esta descripción puede incluirse en el bloque "Model Info" arrastrando dicho bloque de la paleta Simulink -> Model-Wide Utilities, ver Figura 17 y haciendo doble click en él. Escribir por ejemplo en el cuadro de texto:

"Práctica 01

Representación de señales en el tiempo y en la frecuencia

Apartado 4.1"

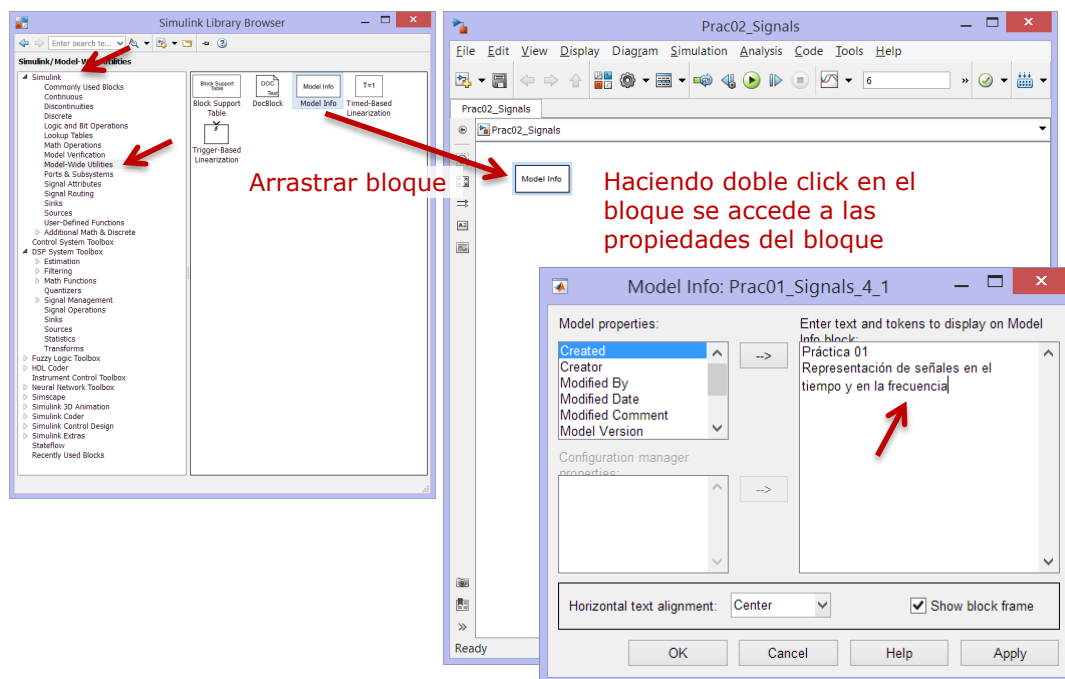


Figura 17. Insertar el bloque "Model Info" en nuestro fichero y cambiar el texto que se muestra.



El procedimiento para crear modelos siempre es el mismo:

1. Crear un nuevo modelo.
2. Arrastrar los bloques de la paleta correspondiente al modelo.
3. Cambiar parámetros del bloque (si es necesario) haciendo doble click en el bloque.
4. Unir bloques entre ellos, para ello, los bloques disponen de puertos de entrada y de salida que serán los que conectemos, es decir, un puerto de salida de un bloque con un puerto de entrada de otro bloque. Estos puertos vienen señalados en el bloque con una flecha.
5. Grabar el modelo.
6. Simular el modelo dando el valor al tiempo máximo que se va a simular el modelo y arrancando la simulación mediante el icono Play.
7. Cuando la simulación termine (100% completada), comprobar los resultados, normalmente en gráficas y figuras que iremos añadiendo.

Normalmente el guion de cada práctica incluirá:

- Una *descripción teórica* de los conceptos a estudiar.
- Un *esquema del modelo en Simulink* a realizar con las conexiones necesarias entre bloques.
- El *nombre de los bloques* que se van a usar.
- En qué *paleta* se encuentra cada bloque.
- Qué *parámetros* del bloque hay que configurar.
- Qué resultados se deben obtener y preguntas sobre los resultados relacionados con los conceptos teóricos a estudiar.

Matlab/Simulink poseen una potente ayuda, donde se describe el comportamiento de los bloques, su configuración y ejemplos de uso, se recomienda usarla ante cualquier duda.

4 Estudio de señales en tiempo y frecuencia con Simulink

4.1 Señal analógica periódica

Vamos a representar una señal periódica de tipo analógico en el dominio del tiempo y en el dominio de la frecuencia, ver Figura 18. Una señal *seno* es un tipo de señal de estas características.

El modelo lo grabaremos en un fichero de nombre "**Prac01_Signals_4_1.slx**".

En concreto representaremos una señal seno mediante el bloque "Sine Wave" con una amplitud (A) de 1.5, una frecuencia de oscilación (f) de 30 Hz y una fase (ϕ) de 0 rad y la simularemos durante 8 segundos (tiempo máximo de simulación).

Nota: En Simulink la frecuencia de la señal se da en rad/s y nosotros trabajamos normalmente en Hz, la transformación es $w = 2\pi f$, con w en rad/s y f en Hz.

El bloque "Time Scope" nos permite visualizar la señal seno respecto del tiempo.

El bloque "Spectrum Analyzer" nos permite visualizar la señal seno en el dominio de la frecuencia. Haciendo doble click en él se pueden visualizar los resultados y configurar las opciones de visualización, ver Figura 19. Este bloque realiza la Transformada de Fourier de la señal que le llega usando el algoritmo FFT (Fast Fourier Transform) que necesita un



número *finito* de muestras de la señal original. El número de muestras que se va a usar se especifica seleccionado "Window Length" y rellenando la caja de texto a su derecha, por ejemplo en 3200 muestras. Por otro lado, necesitamos muestras consecutivas de la señal continua original y esto se consigue usando previamente el bloque "Zero-Order Hold" con un periodo de muestreo de 0.0025 segundos.

Nota IMPORTANTE: El periodo de muestreo 0.0025 segundos se ha calculado como 8 segundos que vamos a simular la señal dividido entre 3200 muestras que queremos para calcular la Transformada de Fourier. El número de muestras que seleccionemos implica más o menos precisión/calidad en el cálculo de la transformada de Fourier.

Por tanto, en todos los ejemplos que se hagan se debe tener en cuenta el tiempo que se simula $T_{simulación}$, el periodo de muestreo T_s y el número de muestras N_s en el analizador del espectro.

Verificándose $T_s = T_{simulación}/N_s$. Normalmente N_s será siempre 3200, el tiempo de simulación se dará en el enunciado y por tanto habrá que calcular el periodo de muestreo T_s .

Nota: Será habitual usar el zoom en las gráficas para poder ver el detalle de los resultados, ver Figura 19.

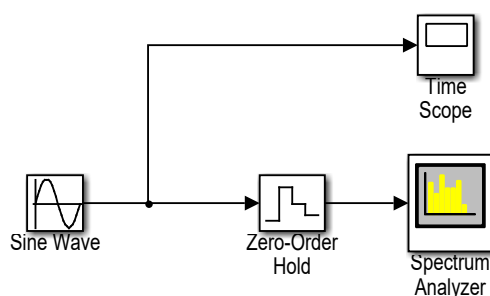


Figura 18. Modelo señal periódica analógica seno().

Tiempo máximo de la simulación: 8 segundos	
Nombre del modelo: Prac01_Signals_4_1.slx	
Bloque:	Sine Wave
Paleta:	Simulink -> Sources
Sine type:	Time based
Time:	Use simulation time
Amplitude:	1.5
Bias:	0
Frequency (rad/sec):	$2 \cdot \pi \cdot 30$
Phase (rad):	0
Sample time:	0.0025
Bloque:	Time Scope
Paleta:	DSP System Toolbox -> Sinks
Parámetros:	Ninguno
Bloque:	Zero-Order Hold
Paleta:	Simulink -> Discrete
Sample time:	0.0025
Bloque:	Spectrum Analyzer
Paleta:	DSP System Toolbox -> Sinks
Main Options:	Type: Power; Full Frequency Span; Window length: 3200
Trace Options:	Units: Watts
Parámetros:	Ver Figura 19



Pinchar para desplegar la configuración: Spectrum Settings

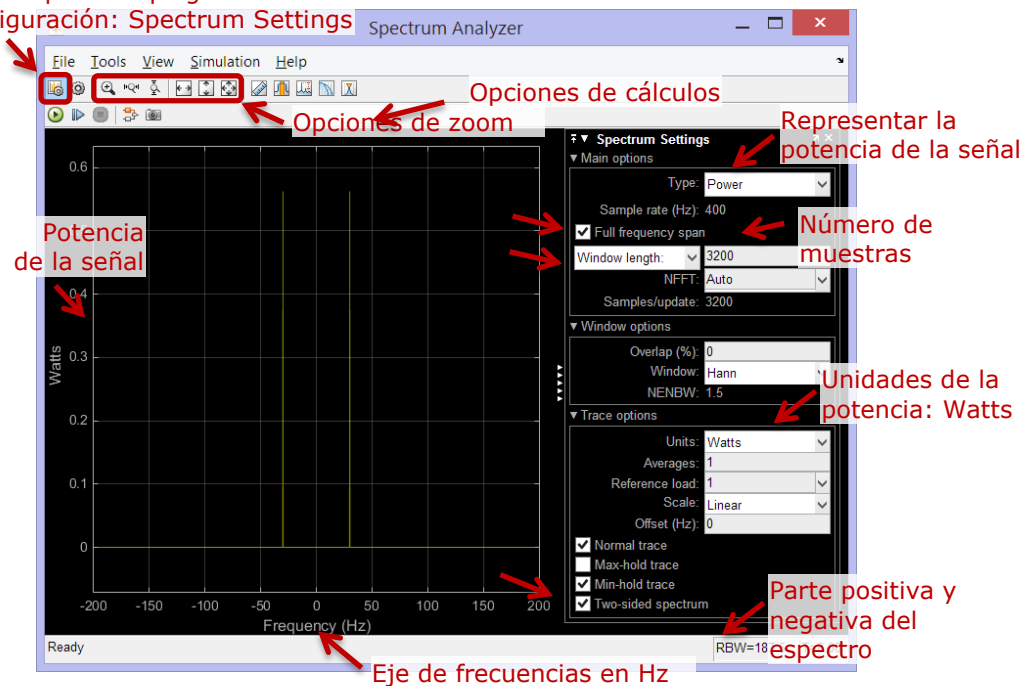


Figura 19. Visualización de espectros (señales en frecuencia).

Resultados

Como se aprecia en la Figura 20 a) la señal seno simulada tiene una amplitud de 1.5 y un periodo (T) de 0.0333 segundos, el periodo de la señal no es más que la inversa de la frecuencia de la señal 30 Hz y una fase de 0 radianes, la señal empieza en cero en el instante de tiempo 0.

El espectro de frecuencias, Figura 20 b), nos indica qué frecuencias contiene la señal seno, en este caso solo una en 30 Hz como era de esperar, pues así la habíamos definido. Esto significa que toda la energía de la señal está concentrada en dicha frecuencia. Esta señal NO contiene más frecuencias. Si descompusiéramos la señal seno mediante series de Fourier veríamos que la serie solo tiene un término y un único armónico en 30 Hz.

La amplitud de la señal en el espectro de frecuencias indica la potencia en watos que contiene la señal para cada frecuencia y tiene que ver, entre otras cosas, con la amplitud de la señal original, en este caso la potencia a 30 Hz es de unos 0.55 watos. La potencia total de la señal será la suma de todas las potencias para todas las frecuencias positivas.

Nota: Los espectros de frecuencias muestran los contenidos de la señal para frecuencias positivas y negativas (son siempre simétricos) pero a nosotros solo nos importa la parte positiva del espectro, pues son las frecuencias que tienen sentido físico. En la visualización de los espectros se puede seleccionar ver la parte negativa y positiva del espectro "Two-side spectrum" o solo la positiva, ver Figura 19.

Importante: Ambas representaciones, en el tiempo y en la frecuencia muestran información diferente de la misma señal. Ambas representaciones son unívocas, a una representación en el tiempo le corresponde una representación en la frecuencia y viceversa.



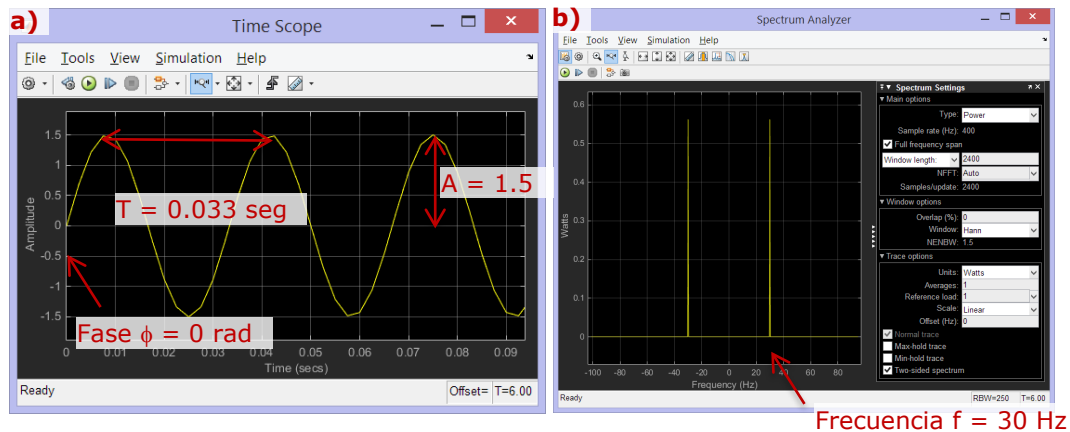


Figura 20. Resultados de la señal seno. a) En el tiempo. b) en la frecuencia.

Efecto de la fase

La fase de una señal juega un papel muy importante en algunos tipos de comunicaciones como veremos más adelante en la modulación digital, que permite por ejemplo conectarnos a Internet a través de un modem ADSL. A continuación vamos a ver el efecto de cambiar la fase de una señal periódica, para ello cambiamos la fase de la señal seno a $\pi/2$ rad, el resultado se muestra en la Figura 21 a) y el espectro de frecuencias se muestra en la Figura 21 b).

Nota. En el espectro de frecuencias se pierde la información relativa a la fase en la señal temporal original, ya que la fase es una característica temporal que no afecta al contenido en frecuencias de la señal.

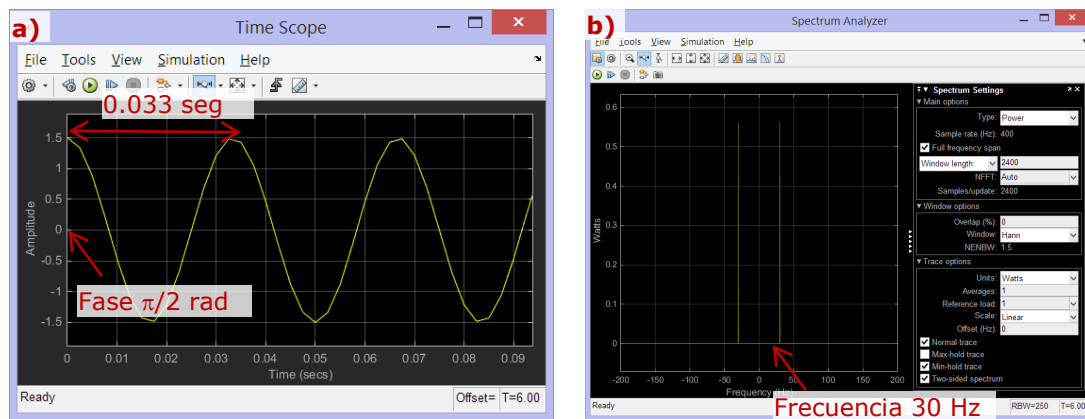


Figura 21. Resultados de la señal seno con una fase de $\pi/2$ rad. a) En el tiempo. b) en la frecuencia.

Efecto de la amplitud y potencia

La potencia del espectro de frecuencias está directamente relacionada, entre otras cosas, con la amplitud de la señal original en el tiempo, ver Figura 8. Por ejemplo para el seno, el espectro de potencias es:

$$S(f) = i \frac{A}{2} (\delta(f + f_0) - \delta(f - f_0)) \Rightarrow$$

$$|S(f)|^2 = \left(\frac{A}{2}\right)^2 (\delta(f + f_0) - \delta(f - f_0))^2 = \left(\frac{A}{2}\right)^2 (\delta(f + f_0) + \delta(f - f_0)) \quad (7)$$

Si la amplitud A es 1.5, entonces la potencia del espectro de frecuencias será $(1.5/2)^2 = 0.5625$, comprobarlo en el espectro de frecuencias de la Figura 20.



4.2 Varias señales analógicas periódicas y señal constante

En este ejercicio, ver Figura 22, vamos a trabajar con tres señales distintas, y las representaremos en el tiempo y en la frecuencia. Además, veremos cómo se puede representar más de una señal en una figura en Simulink, algo que será muy útil para realizar comparaciones. En concreto tenemos las siguientes 3 señales:

- Un seno de amplitud 1.5, frecuencia 30 Hz y fase 0 rad.
- Un seno de amplitud 3, frecuencia 10 Hz y fase 0 rad.
- Una señal constante de valor 1.5.

El modelo lo grabaremos en un fichero de nombre **"Prac01_Signals_4_2.slx"**.

El bloque "Time Scope" permite conectar directamente todas las señales que queramos (y visualizarlas a lo largo del tiempo) sin más que configurar el número de puertos de entrada. Esta configuración se realiza pinchando con el botón derecho del ratón en el bloque, seleccionando Signals & Ports -> Number of inputs ports -> 3.

Sin embargo, es necesario usar el bloque "Matrix Concatenate" para poder representar en el bloque "Spectrum Analyzer" (visualizarlas a lo largo de la frecuencia) más de una señal a la vez. La selección del número de entradas es un parámetro de configuración del bloque y se accede haciendo doble click en él.

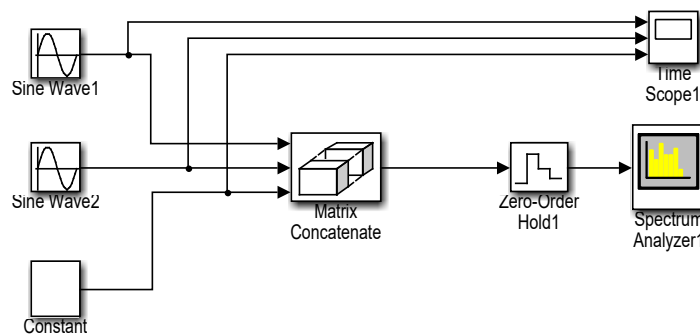


Figura 22. Varias señales periódicas analógicas y señal constante.



Tiempo máximo de la simulación: 8 segundos	
Nombre del modelo: Prac01_Signals_4_2.slx	
Bloque:	Sine Wave1
Paleta:	Simulink -> Sources
Sine type:	Time based
Time:	Use simulation time
Amplitude:	1.5
Bias:	0
Frequency (rad/sec):	$2 \cdot \pi \cdot 30$
Phase (rad):	0
Sample time:	0.0025
Bloque:	Sine Wave2
Paleta:	Simulink -> Sources
Sine type:	Time based
Time:	Use simulation time
Amplitude:	3
Bias:	0
Frequency (rad/sec):	$2 \cdot \pi \cdot 10$
Phase (rad):	0
Sample time:	0.0025
Bloque:	Constant
Paleta:	Simulink -> Sources
Constant value:	1.5
Bloque:	Time Scope1
Paleta:	DSP System Toolbox -> Sinks
Number of inputs ports:	3
(Pinchar con el botón derecho del ratón en el bloque y seleccionar Signals & Ports -> Number of inputs ports -> 3)	
Bloque:	Matrix Concatenate
Paleta:	DSP System Toolbox -> Math Functions -> Matrices and Linear Algebra -> Matrix Operations
Number of inputs:	3
Mode:	Multidimensional array
Concatenate dimensión:	2
Bloque:	Zero-Order Hold1
Paleta:	Simulink -> Discrete
Sample time:	0.0025
Bloque:	Spectrum Analyzer1
Paleta:	DSP System Toolbox -> Sinks
Main Options:	Type: Power; Full Frequency Span; Window length: 3200
Trace Options:	Units: Watts
Parámetros:	Ver Figura 19

Preguntas

- Visualizar las señales en el tiempo.
- Visualizar el espectro de frecuencias de las señales.
- ¿Qué frecuencias presenta cada señal? ¿Se puede distinguir una señal de otra por su espectro?
- ¿Qué ocurre con la potencia de cada señal? Comprobar visualmente como el valor de la potencia depende de la amplitud de la señal en el tiempo.

4.3 Suma de señales periódicas analógicas

En este apartado vamos a estudiar cómo es la señal suma de las dos señales seno del apartado anterior y su representación en el tiempo y en la frecuencia. Añadir al modelo anterior "Prac01_Signals_4_2.xlsx" los bloques que se muestran en la Figura 23.

El modelo lo grabaremos en un fichero de nombre "Prac01_Signals_4_3.slx".



El bloque "Add" es el que permite sumar varias señales de entrada, en este caso la señal "Sin Wave1" y la señal "Sin Wave2".

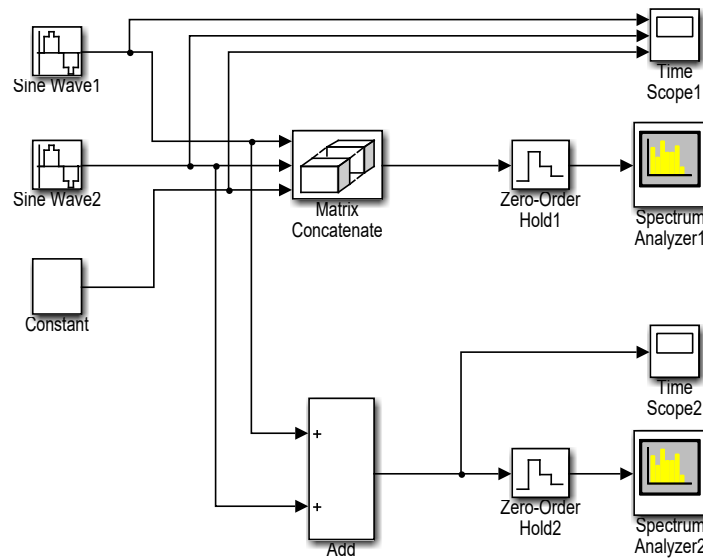


Figura 23. Estudio de una señal suma de dos señales seno.

Tiempo máximo de la simulación: 8 segundos	
Nombre del modelo:	Prac01_Signals_4_3.slx
Bloque:	Add
Paleta:	Simulink -> Math Operations
List of signs:	2
La configuración del resto de bloques es la misma que la vista anteriormente	

Preguntas

- Visualizar la señal suma en el tiempo.
- ¿Se pueden distinguir y caracterizar las señales que componen la señal suma por su representación en el tiempo?
- Visualizar el espectro de frecuencias de la señal suma.
- ¿Qué frecuencias presenta la señal suma? ¿Se pueden *distinguir mejor* las señales que componen la señal suma por su espectro de frecuencias?
- Clasificar todas las señales:

Señal	Seno 1	Seno 2	Constante	Señal suma de Seno 1 y Seno 2
Periódica / no periódica				
Ancho de banda absoluto				
Ancho de banda relativo				
Limitada en banda / no limitada en banda				
Banda base / pasa banda				



4.4 Señal digital periódica

En este ejercicio, ver Figura 24, vamos a trabajar con dos señales periódicas distintas, una analógica, un seno y una digital, una onda cuadrada y las representaremos en el tiempo y en la frecuencia. En concreto tenemos las siguientes 2 señales:

- Un seno de amplitud 1, frecuencia 1 Hz y fase 0 rad.
- Una onda cuadrada de amplitud 1, periodo 1 segundo y fase 0 segundos. Además el ancho del pulso lo definiremos como la mitad del periodo exactamente (50 % del periodo). Es decir 0.5 segundos la señal está a 1 y 0.5 segundos la señal está en 0.

El modelo lo grabaremos en un fichero de nombre **"Prac01_Signals_4_4.slx"**.

El bloque "Pulse Generator" permite generar señales periódicas cuadradas.

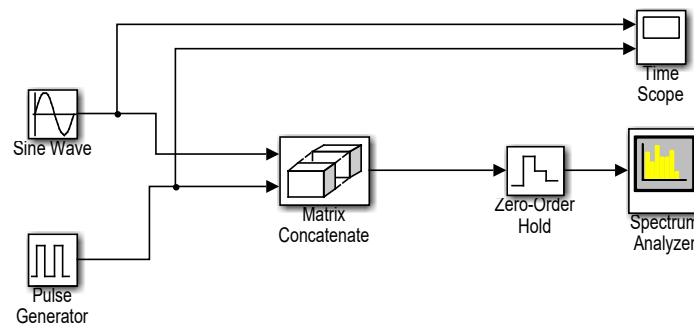


Figura 24. Señal digital periódica.

Tiempo máximo de la simulación: 8 segundos	
Nombre del modelo: Prac01_Signals_4_4.slx	
Bloque:	Sine Wave
Paleta:	Simulink -> Sources
Sine type:	Time based
Time:	Use simulation time
Amplitude:	1
Bias:	0
Frequency (rad/sec):	$2 \cdot \pi \cdot 1$
Phase (rad):	0
Sample time:	0.0025
Bloque:	Pulse Generator
Paleta:	Simulink -> Sources
Pulse type:	Time based
Time:	Use simulation time
Amplitude:	1
Period (sec):	1
Pulse Width (% of Period):	50
Phase delay (sec):	0
Bloque:	Matrix Concatenate
Paleta:	DSP System Toolbox -> Math Functions -> Matrices and Linear Algebra -> Matrix Operations
Number of inputs:	2
Mode:	Multidimensional array
Concatenate dimensión:	2
La configuración del resto de bloques es la misma que la vista anteriormente	

Preguntas

- Visualizar la señal analógica y la digital en el tiempo.
- Visualizar el espectro de frecuencias de la señal analógica y de la señal la digital. ¿Qué diferencias observas?



- c) ¿A qué se corresponde la frecuencia de 1 Hz en ambos espectros? Y ¿el resto de frecuencias?
- d) Clasificar todas las señales:

Señal	Seno	Onda cuadrada
Periódica / no periódica		
Ancho de banda absoluto		
Componente de frecuencia con mayor potencia		
Ancho de banda relativo. <i>Aquella banda de frecuencias cuyas componentes tienen más del 1% de potencia respecto de la componente de frecuencia de mayor potencia</i>		
Limitada en banda / no limitada en banda		
Señal Banda base / pasa banda		

- e) Si queremos transmitir **directamente** las dos señales ¿qué tipo de transmisión necesitaremos usar para cada una?

4.5 Señal digital no periódica

En este apartado, ver Figura 25, vamos a trabajar con una señal digital no periódica de 16 bits que transmite la siguiente secuencia de datos: [0 1 0 1 0 0 0 1 1 1 0 1 0 0 0 0]. Donde la duración de cada pulso es de 0.5 segundos.

Es importante porque la manera de generar dicha señal digital será la que usemos a lo largo de las prácticas siguientes.

En todo sistema de transmisión de señales digitales un concepto importante es la duración de un pulso, normalmente esta duración viene regida por el reloj de la tarjeta de red que se disponga para transmitir la señal digital. En nuestro modelo de generación de señales digitales, el reloj lo generaremos con un generador de pulsos.

El modelo lo grabaremos en un fichero de nombre **"Prac01_Signals_4_5.slx"**.

El bloque "Pulse Generator" permite generar señales periódicas cuadradas de amplitud 0.5, periodo 1 segundo, fase 0 segundos y con un ancho del pulso del 50 % del periodo total. Actuará como el reloj de la tarjeta de red de un PC.

El bloque "Triggered Signal from Workspace" tiene como entrada los datos digitales que se quieren transmitir, es decir la secuencia de ceros y unos anterior y como salida la señal digital correspondiente con la duración adecuada de cada pulso.

Importante. En el bloque "Triggered Signal from Workspace", la secuencia de datos [0 1 0 1 0 0 0 1 1 1 0 1 1 0 0 0] que se quiere transmitir se configura de una manera particular: el primer bit (en este caso el 0) se configura en el parámetro de configuración "Initial output" y el resto de la secuencia (15 bits) se configuran en el parámetro de configuración "Signal".



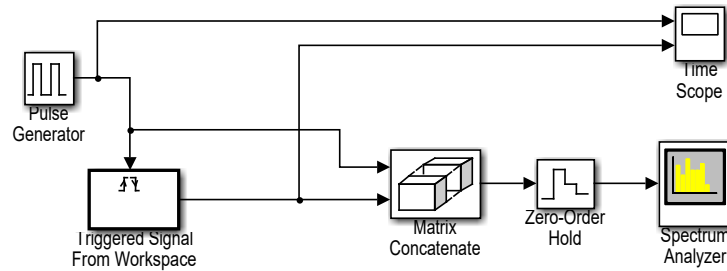


Figura 25. Señal digital no periódica.

Tiempo máximo de la simulación: 8 segundos	
Nombre del modelo:	Prac01_Signals_4_5.slx
Bloque:	Pulse Generator
Paleta:	Simulink -> Sources
Pulse type:	Time based
Time:	Use simulation time
Amplitude:	0.5
Period (sec):	1
Pulse Width (% of Period):	50
Phase delay (sec):	0
Bloque:	Triggered Signal From Workspace
Paleta:	DSP System Toolbox -> Signal Operations
Signal:	[1 0 1 0 0 0 1 1 1 0 1 1 0 0 0]
Trigger type:	Either edge
Initial output:	0
Samples per frame:	1
Form output:	Setting zero
La configuración del resto de bloques es la misma que la vista anteriormente	

Preguntas

- Visualizar las dos señales digitales en el tiempo, el pulso periódico y la señal digital.
- Afianzar el concepto de dato a transmitir y el concepto de señal transmitida. ¿Quién es cada una en este ejemplo?
- Visualizar el espectro de frecuencias del pulso periódico y de la señal la digital. ¿Qué diferencias observas?
- Clasificar todas las señales

Datos que se transmiten	
Señal	
Señal que se transmite	
Periódica / no periódica	
Ancho de banda absoluto	
Componente de frecuencia con mayor potencia	
Ancho de banda relativo. <i>Aquella banda de frecuencias cuyas componentes tienen más del 0.5% de potencia respecto de la componente de frecuencia de mayor potencia</i>	
Limitada en banda / no limitada en banda	
Señal Banda base / pasa banda	



4.6 Señal de audio

En este apartado vamos a trabajar con una señal de audio la visualización del espectro de frecuencias de dicha señal y el efecto de eliminar grupos de frecuencias. Como se ha visto anteriormente, el espectro de frecuencias nos indica las frecuencias que contiene la señal original y su importancia en dicha señal.

Se utilizará el código de Simulink "FFT_Imperial_March.slx", ver Figura 26, y el fichero de audio "The_Imperial_March.mp3" que debe ser copiado en la carpeta en la que esté el código de Simulink. Se reproducirán 20 segundos de música tras los cuales se visualizará el espectro de frecuencias del audio reproducido.

Se puede escoger entre reproducir la señal original, la señal original pero filtrando los agudos (eliminando frecuencias superiores a 2000 Hz) o la señal original filtrando los graves (eliminando frecuencia inferiores a 1000 Hz). Se recomienda habilitar los altavoces del equipo para comprobar los efectos del filtrado de frecuencias.

Importante. Pinchando en el bloque "From Multimedia File STEREO" se debe seleccionar la ruta donde se encuentre el fichero de audio mp3.

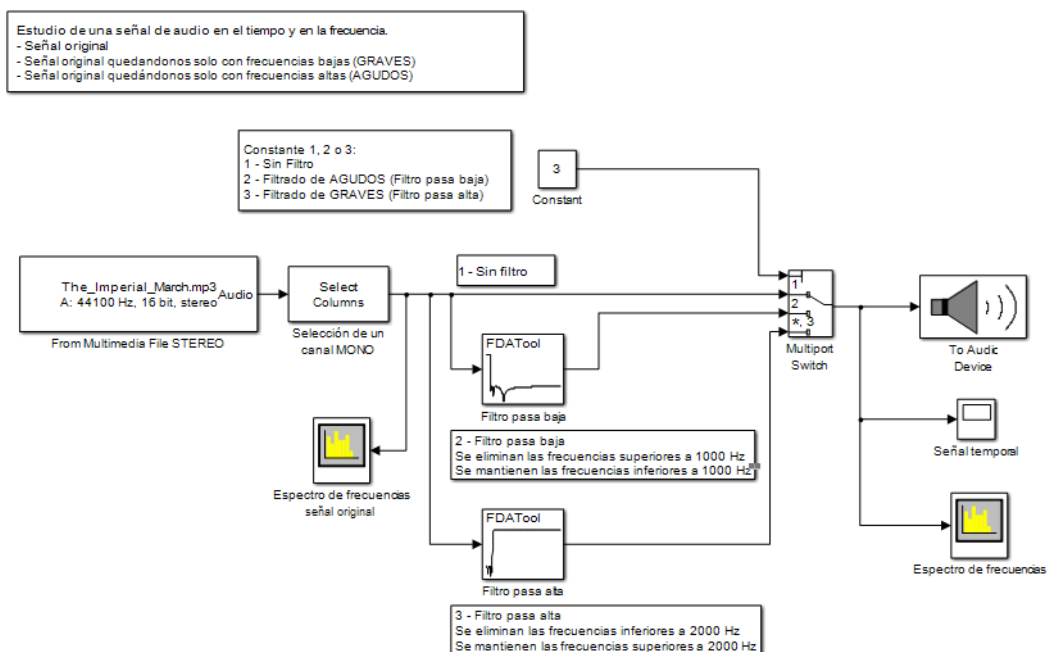


Figura 26. Señal de audio.

Preguntas

- Se está trabajando con una señal de audio, ¿en general cuál es el rango de frecuencias de una señal sonora (espectro audible por los humanos)?
- Clasifica la señal original (sin filtro) con la que se está trabajando

Periódica / no periódica	
Ancho de banda absoluto	
Componente de frecuencia con mayor potencia	
Ancho de banda relativo	
Limitada en banda / no limitada en banda	
Señal Banda base / pasa banda	



- c) Compara el ancho de banda relativo de la señal original con el ancho de banda del espectro audible por los humanos. ¿Qué ocurre? ¿Podrías justificar la diferencia?
- d) Prueba a eliminar las frecuencias superiores a 1000 Hz y compara los espectros de la señal original y de la señal filtrada, así como el efecto sonoro de filtrar.
- e) Prueba a eliminar las frecuencias inferiores a 2000 Hz y compara los espectros de la señal original y de la señal filtrada, así como el efecto sonoro de filtrar.
- f) Filtrando en el filtro **pasa baja**, seleccionar una frecuencia F_{pass} de 10000 Hz y pinchar en "Design Filter" volver a reproducir el fichero con la constante en 2. ¿Qué ha ocurrido con el ancho de banda de la señal? ¿Qué efectos tiene en el audio reproducido?
- g) A la vista de los resultados ¿qué se puede concluir?

IV Bibliografía

Stallings. Comunicaciones y redes de computadoras, Pearson. 2004





Grado en Ingeniería Informática

REDES

PRÁCTICA 2

Transmisión de datos. Canales de transmisión

Docentes:

Alejandro Merino

Daniel Sarabia Ortiz

*Dpto. de Ingeniería Electromecánica
Área de Ingeniería de Sistemas y Automática*

Versión 2.0

Fecha 08/02/2022 19:15

Esta obra está sujeta a la licencia Reconocimiento 4.0 Internacional de Creative Commons. Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by/4.0/>



Índice de contenidos

I	INTRODUCCIÓN.....	3
II	OBJETIVOS.....	3
III	CONTENIDOS ESPECÍFICOS DEL TEMA	4
1	Transmisión de datos	4
1.1	Velocidad en el movimiento de información.....	5
1.2	Dificultades en la transmisión.....	9
1.3	Ancho de banda de un canal.....	12
1.4	Transmisión en banda base.....	13
1.5	Capacidad de un canal.....	13
1.6	Transmisión analógica versus transmisión digital	14
1.7	Tipos de medios físicos de transmisión.....	15
1.8	Medios de transmisión guiados	17
1.9	Medios de transmisión no guiados.....	21
2	Estudio de transmisión en banda base con Simulink	24
2.1	Velocidad de transmisión y ancho de banda de la señal.....	24
2.2	Velocidad de transmisión y ancho de banda del canal.....	27
2.3	Atenuación de la señal recibida y ancho de banda de un canal	31
3	Problemas capacidad del canal.....	33
IV	BIBLIOGRAFÍA	34



I Introducción

Todas las señales acaban transmitiéndose por un medio físico que conecta el emisor y el receptor en un sistema de transmisión de datos. La naturaleza del medio físico afectará al tipo de transmisión que se quiera efectuar y condicionará el uso de un tipo u otro de señales que transporten la información entre el emisor y el receptor.

Además cualquier medio físico dificultará la transmisión añadiendo problemas a la transmisión en sí misma, lo que provocará que la señal que llega al receptor no sea exactamente igual a la señal transmitida por el emisor.

Usaremos la toolbox de Simulink **"DSP System Toolbox"** para estudiar distintos tipos de señales tanto analógicas como digitales presentes en cualquier sistema de transmisión de datos. El estudio se realizará tanto en el dominio temporal como en el de la frecuencia obteniendo los correspondientes espectros de frecuencias.

Todos los alumnos de la Universidad de Burgos disponen de licencia académica para utilizar Matlab de forma legal y trabajaremos con la versión **R2019b de Matlab**. Se recomienda usar dicha versión, aunque los códigos también funcionan en versiones superiores.

II Objetivos

- Conocer los principios de la velocidad de transmisión de información en sistemas de transmisión.
- Distinguir el concepto de velocidad de transmisión en baudios y velocidad de transmisión en bits por segundo. Tanto en señales analógicas como digitales.
- Distinguir la diferencia entre ancho de banda de una señal y ancho de banda de un canal.
- Estudiar la relación entre velocidad de transmisión y el ancho de banda de un canal.
- Distinguir entre la transmisión en banda base y sus características.
- Identificar la problemática de la atenuación en las transmisiones en banda base.
- Manejar unidades representadas en decibelios y conversión entre ellas.
- Calcular las capacidades de los canales, tanto en canales ideales como canales con ruido.
- Identificar y diferenciar distintos medios de transmisión en función del soporte físico, es decir, medios guiados y no guiados.
- Ser capaz de modelar canales de transmisión en Simulink.
- Practicar y afianzar la mayoría de los conceptos anteriores usando Simulink y en particular la "DSP System Toolbox" de Simulink.



III Contenidos específicos del tema

1 Transmisión de datos

La transmisión de datos (o transmisión de información) en un sistema de transmisión entre un transmisor y un receptor ocurre a través de un **medio físico de transmisión**, ver Figura 1, que se denomina **canal de comunicaciones**, o simplemente canal.

El estudio de los canales de comunicación puede abordarse de dos formas diferentes según las características técnicas consideradas:

- **Canal físico**, es el relacionado con las características físicas y eléctricas del sistema de comunicaciones. Se ocupa de los fenómenos relativos a la transmisión de señales. Usa como criterio de eficiencia la calidad de la señal recibida y trata de minimizar el efecto de los fenómenos de ruidos y distorsiones. Los canales físicos pueden estar constituidos por diferentes medios de comunicación.
- **Canal de información**, está relacionado con las especificaciones externas del sistema de telecomunicaciones. Se vincula con las técnicas relacionadas con la teoría de la información y de la codificación. Se ocupa de evaluar y permitir administrar adecuadamente los recursos del canal físico. El criterio de eficiencia es la velocidad de transmisión de la información y la calidad con la que es transportada.

Hay que recordar que las señales que transportan los datos pueden ser:

- **Analógicas**, es decir, ondas electromagnética que varían continuamente, que se pueden transmitir a través de un medio sólido o sin soporte sólido.
- **Digitales**, es decir, secuencias de pulsos de tensión que se pueden transmitir a través *sólo* de un medio conductor (soporte sólido).

Dependiendo de la forma de conducir la señal a través del medio, el medio de transmisión puede ser:

- **Guiado**: Necesitan un soporte sólido, cable de par trenzado, cable coaxial, fibra óptica.
- **No guiado**: No necesitan un soporte sólido: aire, agua, vacío.

Cada medio de transmisión presenta distintas características: Ancho de banda, retardos en la transmisión, atenuación, interferencias, costes, facilidad de instalación y mantenimiento, etc.

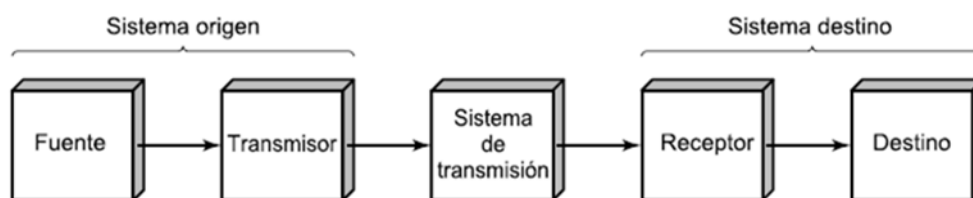


Figura 1. Modelo típico para las comunicaciones.

Cuando se usa una onda electromagnética para transportar datos, hay que tener claro que dependiendo de la frecuencia de la misma será de un tipo u otro.

Es decir, las ondas electromagnéticas se caracterizan por su frecuencia. Al conjunto de frecuencias cubierto por las ondas electromagnéticas se denomina espectro electromagnético, ver Figura 2. De esta manera podemos tener ondas electromagnéticas de radio, microondas, infrarrojos, luz visible, etc.



Por ejemplo, si se quiere usar fibra óptica deberemos transmitir señales *visibles*, que ocurren en un rango de frecuencias de 10^{14} y 10^{15} Hz. Otro ejemplo, el cable coaxial puede usarse para transmitir señales de *radio* que son señales caracterizadas por una frecuencia entre 10^4 y 10^9 Hz.

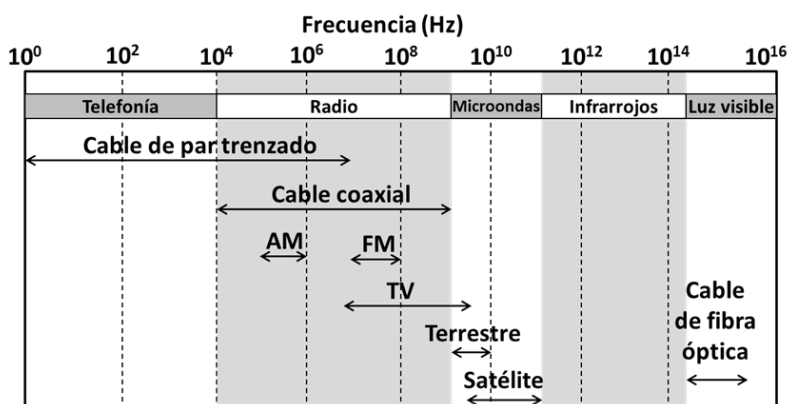


Figura 2. Espectro electromagnético para la transmisión de señales analógicas y los medios guiados y no guiados que se pueden usar.

1.1 Velocidad en el movimiento de información

La transmisión de datos lleva implícito un movimiento de información y por tanto es interesante definir la velocidad a la que se realiza.

Símbolo o elemento de señalización

Aquellos valores distintos de la señal que ocupa el intervalo más corto:

- Para una señal analógica, el símbolo es la frecuencia, fase o amplitud.
- Para una señal digital, el símbolo es la amplitud constante del pulso de tensión.

El intervalo más corto depende del tipo de señal:

- Para una señal analógica, el intervalo más corto es el periodo de la señal.
- Para una señal digital, el intervalo más corto es la duración (ancho) del pulso de tensión.

Velocidad en símbolos (V_s) o velocidad de modulación (V_m)

Es el número máximo de símbolos que se pueden transmitir en un segundo. Sus unidades son el baudio (N° símbolos / 1 seg)

Se utiliza como unidad el baudio, que es equivalente a un intervalo significativo por segundo, o sea, $V_m = N^\circ \text{ símbolos}/t$ baudios ($t =$ duración en segundos del intervalo significativo mínimo o intervalo más corto).

Velocidad de transmisión serie o régimen binario (V_t o R o C)

Es el número máximo de elementos binarios (bits) que pueden transmitirse en un segundo. Se mide en bps (bit/s).

Bit por segundo frente a baudio

Un error frecuente es utilizar el baudio como sinónimo de bit por segundo. La velocidad en baudios (baud rate), no debe confundirse con la tasa de bits. La velocidad en baudios de una señal representa el número de cambios de estado, o elementos de señalización, que la señal tiene en un segundo. Cada elemento de señalización transmitido puede transportar uno o más bits. La relación entre bit/s y baudios es:



$$V_t = V_m \log_2(n) \quad (1)$$

Dónde n es el número máximo de símbolos a transmitir. Sólo cuando el número máximo de símbolos diferentes a transmitir es 2 ($n=2$) coincide la velocidad de transmisión de datos en baudios y en bits por segundo. Importante, el logaritmo en la expresión (1) es en base 2.

Ejemplos de velocidad de la información en señales analógicas

Ejemplo 1. Considerar la señal periódica seno de la Figura 3, cuyo periodo es $T=1$ segundo (frecuencia = $1/1 = 1$ Hz) y de amplitud 2 voltios siempre. Por tanto el número de símbolos distintos es $n=1$ y si consideramos que cada vez que enviamos la señal con una amplitud de 2 v, implica la transmisión de un bit "0" de información, el intervalo más corto de esta señal periódica es el periodo $T = 1$ s y la velocidad en símbolos será $V_m = 1$ símbolo/s = 1 baudio.

Por tanto la velocidad de transmisión serie será $V_t = v_m \log_2(n) = 1 \times 0 = 0$ bps. Es decir transmitimos 8 bits "0" consecutivos. **ii Si no cambia algún símbolo no se transmite información !!**

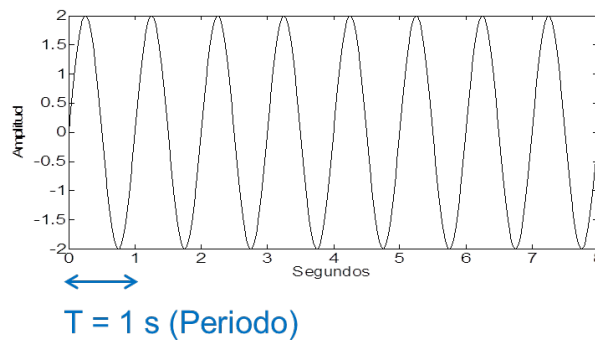


Figura 3. Señal seno de periodo 1 s y amplitud 2 v siempre.

Ejemplo 2. Ahora queremos transmitir información como una secuencia de 8 bits consecutivos [0 1 0 1 1 1 0 0] y usamos una señal seno de periodo $T = 1$ s (frecuencia = $1/1 = 1$ Hz) pero para transmitir un bit "0" usamos una amplitud de 2 v y para transmitir un bit "1" usamos una amplitud de 4 v, ver Figura 4 a). Ahora el número de símbolos diferentes es $n = 2$, el intervalo más corto será el periodo $T = 1$ s y por tanto la velocidad en símbolos será $V_m = 1$ símbolo/s = 1 baudio y la velocidad de transmisión serie será $V_t = v_m \log_2(n) = 1 \times 1 = 1$ bps.

Ejemplo 3. Ahora queremos transmitir información como una secuencia de 8 bits consecutivos [0 1 0 1 1 1 0 0] y usamos una señal seno de periodo $T = 1$ s (frecuencia = $1/1 = 1$ Hz) pero para transmitir dos bits "01" usamos una amplitud de 2 v, para transmitir "10" usamos una amplitud de 4 v, para transmitir "11" usamos 6 v y para transmitir dos bits "00" usamos una amplitud de 8 v, ver Figura 4 b). Ahora el número de símbolos diferentes es $n = 4$, el intervalo más corto será el periodo $T = 1$ s y por tanto la velocidad en símbolos será $V_m = 1$ símbolo/s = 1 baudio y la velocidad de transmisión serie será $V_t = v_m \log_2(n) = 1 \times 2 = 2$ bps.

Como se aprecia del ejemplo 2 y 3 que transmiten la misma información, la misma secuencia de 8 bits, aumentando el número de símbolos se puede aumentar la velocidad de transmisión de 1 bps a 2 bps y por tanto la del tiempo que dura la transmisión pasa de 8 segundos a 4 segundos.

Ejemplo 4. Ahora queremos transmitir información como una secuencia de 8 bits consecutivos [0 1 0 1 1 1 0 0] y usamos una señal seno de periodo $T = 0.5$ s (frecuencia = $1/0.5 = 2$ Hz) pero para transmitir un bit "0" usamos una amplitud de 2 v y para transmitir un bit "1" usamos una amplitud de 4 v, ver Figura 4 c). Ahora el número de



símbolos diferentes es $n = 2$, el intervalo más corto será el periodo $T = 0.5$ s y por tanto la velocidad en símbolos será $V_m = 2$ símbolo/s = 2 baudio y la velocidad de transmisión serie será $V_t = v_m \log_2(n) = 2 \times 1 = 2$ bps.

Como se aprecia del ejemplo 2 y 4 que transmiten la misma información, la misma secuencia de 8 bits, aumentando la frecuencia de la señal de 1 Hz a 2 Hz se puede aumentar la velocidad de transmisión de 1 bps a 2 bps y por tanto el tiempo que dura la transmisión pasa de 8 segundos a 4 segundos. Por eso cuando se usa la fibra óptica que transmite señales de luz de frecuencias de 1000 THz (10^{15} Hz) la velocidad de transmisión es mucho más grande que cuando se usa cables de cobre que permiten transmitir señales de frecuencias de 1 MHz (10^6 Hz).

En estos ejemplos hemos usado solo cambios en la amplitud como símbolos diferentes, pero en una señal analógica se puede usar también la frecuencia y la fase de la señal. Veremos esto más en detalle cuando hablemos de **modulación digital** en la práctica 3.

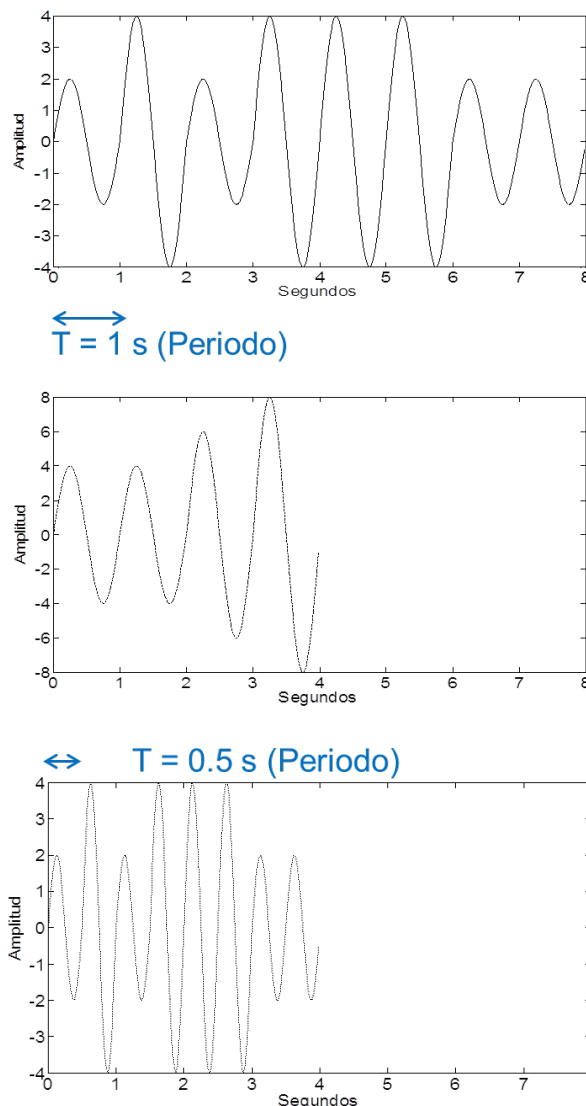


Figura 4. a) Señal analógica con velocidad de transmisión 1 bps y 1 baudio. b) Señal con velocidad de transmisión 2 bps y 1 baudio. c) Señal con velocidad de transmisión 2 bps y 2 baudios. Todas transmiten la secuencia de bits [0 1 0 1 1 1 0 0] como información.

Ejemplos de velocidad de la información en señales digitales

En las señales digitales, el símbolo posible solo es la amplitud del pulso y el intervalo más corto es la duración (anchura) del pulso. Por tanto aumentando el número de símbolos es posible aumentar la velocidad de transmisión en bps y también disminuyendo la duración del pulso es posible aumentar dicha velocidad de transmisión. El problema de disminuir



la duración del pulso es que aumenta considerablemente el ancho de banda de la señal y es posible que no pueda transmitirse por el correspondiente medio de transmisión: cable de par de cobre o cable coaxial que tienen un ancho de banda pequeño.

Veremos estos problemas después cuando introduzcamos el concepto de **ancho de banda de un sistema** y en la práctica 3 cuando hablemos de las técnicas de **codificación**.

Ejemplo 5. Queremos transmitir información como una secuencia de 8 bits consecutivos [0 1 0 1 1 1 0 0] usando una señal digital de 1 s de ancho de pulso y teniendo en cuenta que para transmitir un bit "0" usamos una amplitud de 0 v y para transmitir un bit "1" usamos una amplitud de 2 v, ver Figura 5 a). El número de símbolos diferentes es $n = 2$ (podemos transmitir dos amplitudes, 0 v y 2 v), el intervalo más corto será el ancho de pulso $T = 1$ s y la velocidad en símbolos será $V_m = 1$ símbolo/s = 1 baudio (en cada segundo solo transmitimos un único símbolo) y la velocidad de transmisión serie será $V_t = v_m \log_2(n) = 1 \times 1 = 1$ bps.

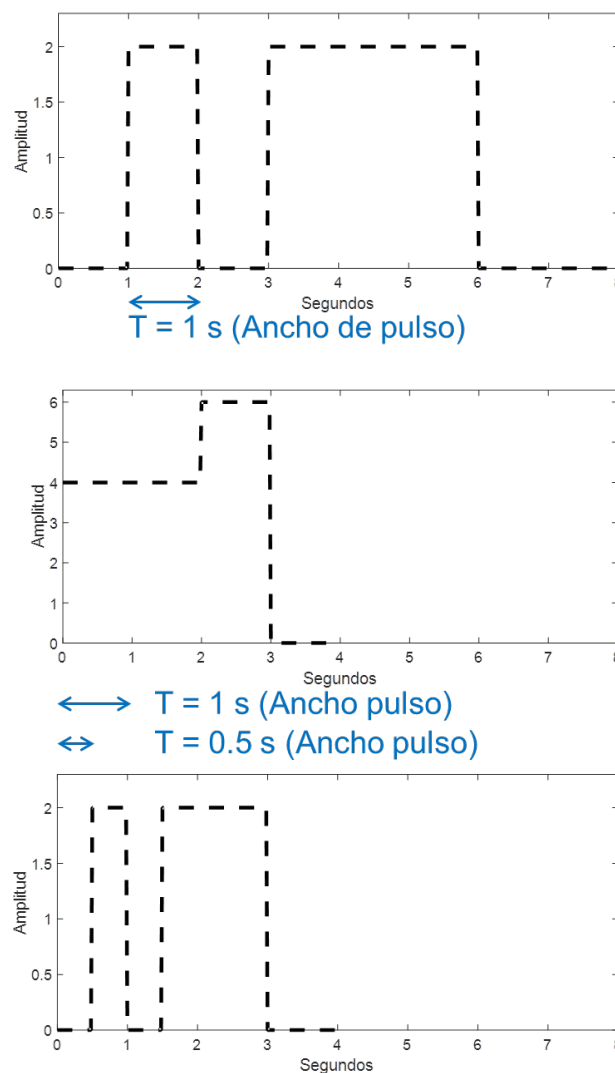


Figura 5. a) Señal digital con velocidad de transmisión 1 bps y 1 baudio. b) Señal con velocidad de transmisión 2 bps y 1 baudio. c) Señal con velocidad de transmisión 2 bps y 2 baudios. Todas transmiten la secuencia de bits [0 1 0 1 1 1 0 0] como información.

Ejemplo 6. Queremos transmitir información como una secuencia de 8 bits consecutivos [0 1 0 1 1 1 0 0] usando una señal digital de 1 s de ancho de pulso y teniendo en cuenta que para transmitir dos bits "00" usamos una amplitud de 0 v, para transmitir dos bits "01" usamos una amplitud de 2 v, para transmitir dos bits "10" usamos una amplitud de 4 v y para transmitir dos bits "11" usamos 6 v, ver Figura 5 b). El número de símbolos diferentes es $n = 4$ (podemos transmitir cuatro amplitudes, 0 v, 2 v, 4 v y 6 v), el



intervalo más corto será el ancho de pulso $T = 1$ s y la velocidad en símbolos será $V_m = 1$ símbolo/s = 1 baudio (en cada segundo solo transmitimos un único símbolo, una única amplitud) y la velocidad de transmisión serie será $V_t = v_m \log_2(n) = 1 \times 2 = 2$ bps.

Ejemplo 7. Queremos transmitir información como una secuencia de 8 bits consecutivos [0 1 0 1 1 1 0 0] usando una señal digital de 0.5 s de ancho de pulso y teniendo en cuenta que para transmitir un bit "0" usamos una amplitud de 0 v y para transmitir un bit "1" usamos una amplitud de 2 v, ver Figura 5 c). El número de símbolos diferentes es $n = 2$ (podemos transmitir dos amplitudes, 0 v y 2 v), el intervalo más corto será el ancho de pulso $T = 0.5$ s y la velocidad en símbolos será $V_m = 2$ símbolo/s = 2 baudio (en cada segundo se pueden transmitir dos símbolos, dado que cada símbolo tarda 0.5 s) y la velocidad de transmisión serie será $V_t = v_m \log_2(n) = 2 \times 1 = 2$ bps.

Como se aprecia del ejemplo 5 y 6 que transmiten la misma información, la misma secuencia de 8 bits, aumentando el número de símbolos de 2 a 4 se puede aumentar la velocidad de transmisión de 1 bps a 2 bps y por tanto el tiempo que dura la transmisión pasa de 8 segundos a 4 segundos.

En el caso de los ejemplos 5 y 7, disminuyendo el ancho del pulso, usando en ambos casos dos símbolos es posible aumentar la velocidad de transmisión de 1 bps 2 bps y por tanto el tiempo que dura la transmisión pasa de 8 segundos a 4 segundos.

1.2 Dificultades en la transmisión

En cualquier sistema de comunicaciones se debe aceptar que la señal que se recibe diferirá de la señal transmitida debido a dificultades sufridas en la transmisión, es decir las señales son *alteradas* durante la transmisión:

- En las señales analógicas y digitales se degrada la calidad de la señal.
- En las señales digitales se generan bits erróneos, un 1 binario se transformará en un 0 y viceversa.

Las dificultades más significativas son: Atenuación, distorsión por retardo, desvanecimiento o fading, rebotes en los cables o ecos y ruidos.

Decibelio (dB)

El decibelio es una unidad logarítmica muy utilizada que expresa la relación entre dos magnitudes S_1 y S_2 . $A_{dB} = 10 \log(S_1/S_2)$. **ii El logaritmo es en base 10 !!**

A veces se usa el decibelio para expresar una magnitud respecto de un valor de referencia, por ejemplo, la potencia de una señal en decibelios respecto de 1 watio: $P_{dBW} = 10 \log(P/1)$, donde P es la potencia media de la señal en watios y P_{dBW} es la potencia de la señal en decibelios-watio.

Otro ejemplo, para expresar tensiones respecto de 1 mv: $T_{dBmV} = 10 \log(T/1)$, donde T es la tensión de la señal en mv y T_{dBmV} es la tensión de la señal en decibelios-milivoltio.

Atenuación

Es la pérdida de potencia de la señal *a medida* que se propaga por el medio físico de transmisión, es decir, está relacionada con la distancia entre el emisor y el receptor. La atenuación N en decibelios se calcula como la relación entre la potencia de la señal transmitida P_T y la potencia de la señal recibida P_R , ambas en watios:

$$N_{dB} = 10 \log \frac{P_T}{P_R} \quad (2)$$

Si $N_{dB} > 0$ la señal se atenúa y si $N_{dB} < 0$ la señal se amplifica.



También puede relacionarse la atenuación N en decibelios con la amplitud de la señal transmitida A_T y la amplitud de la señal recibida A_R , ambas en voltios normalmente, ver Figura 6.

$$N_{dB} = 10 \log \frac{A_T}{A_R} \quad (3)$$

La atenuación también depende de las características físicas del medio:

- **Medios guiados:** La pérdida de potencia depende de la distancia d en metros entre el emisor y receptor y del grosor y material del cable dados por la constante α .

$$N_{dB} = 10 \log(\alpha d) \quad (4)$$

- **Medios no guiados:** La pérdida de potencia depende de la distancia d en metros entre el emisor y receptor y de la frecuencia f en Hz de la señal (¡también de las condiciones atmosféricas!!)

$$N_{dB} = 10 \log(4\pi d f)^2 \quad (5)$$

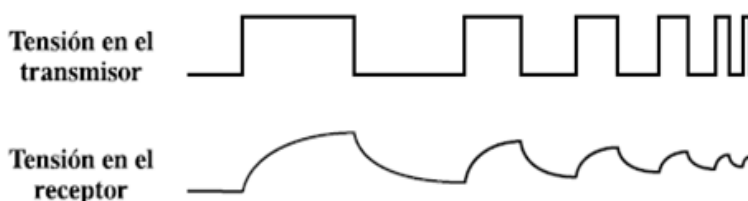


Figura 6. Atenuación en una señal digital.

Distorsión de retardo

Es propio de los medios guiados donde la velocidad de propagación de una señal varía con la frecuencia, por tanto, las distintas componentes de la señal llegarán al receptor en instantes diferentes de tiempo porque se propagan a diferente velocidad. El efecto es que unos datos se solaparán con los anteriores, es decir que habrá interferencias entre símbolos.

Desvanecimientos o fading

Es propio de los medios no guiados y consiste en la disminución de la relación señal ruido (S/N), ver apartado siguiente, debido principalmente a condiciones atmosféricas.

Rebotes en los cables o ecos

Se produce cuando en un circuito se produce un cambio en las características eléctricas de los conductores y parte de la onda transmitida se refleja, interfiriendo con la señal que viene en sentido contrario o incluso con ella misma después de varias reflexiones.

Ruido

Son señales no deseadas que se insertan en algún punto entre el emisor y el receptor. El ruido es el factor de mayor importancia de entre los que limitan las prestaciones de un sistema de comunicación. Entre los tipos de ruidos más habituales se encuentran el ruido térmico, el ruido impulsivo, la diafonía y el ruido de intermodulación.



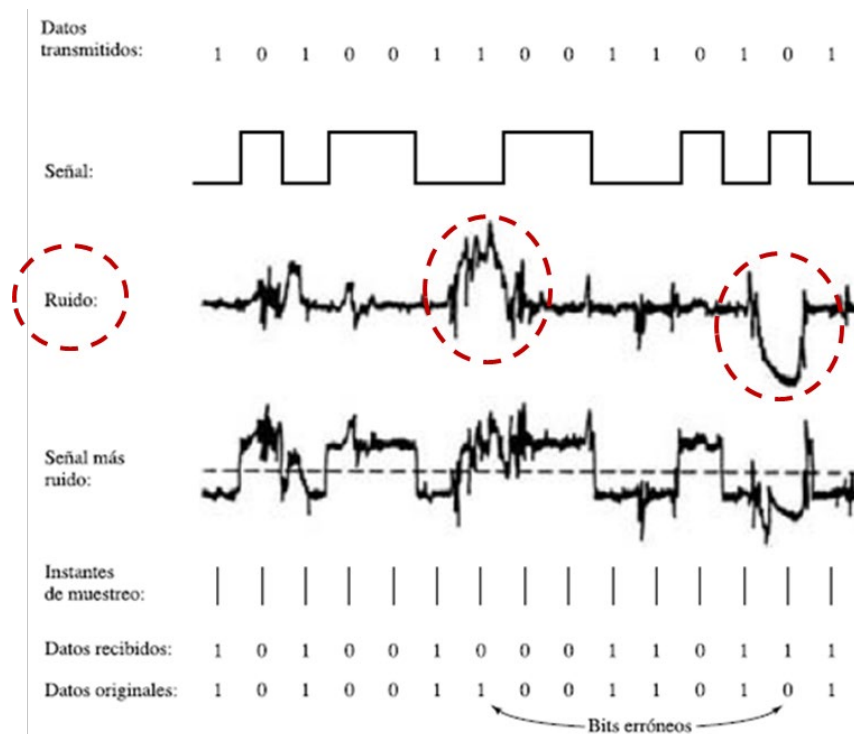


Figura 7. Efecto del ruido en una señal digital. En círculos rojos el efecto del ruido impulsivo ha provocado la recepción de bits erróneos.

Ruido térmico (blanco o gaussiano): es inherente al sistema de comunicaciones, por tanto no se puede eliminar y se produce por el movimiento aleatorio de los electrones en los conductores y demás componentes electrónicos pertenecientes al sistema de comunicaciones. En transmisión de señales digitales suele ser responsable de errores de bits aislados.

Se caracteriza por tener una distribución uniforme en frecuencia y porque depende de la temperatura.

Por tanto la potencia P (en watos) del ruido térmico presente en un canal de ancho de banda B (en Hz) a una temperatura T (en kelvins) es:

$$P = kTB \quad (6)$$

Donde k es la constante de Boltzmann $k = 1.3803 \cdot 10^{23}$ Julios / K.

Ruido impulsivo: es externo al canal utilizado. Aparece de forma discontinua, en intervalos irregulares de tiempo y con picos de corta duración, pero de gran amplitud. Es muy difícil localizar su origen. Suele tener un comportamiento aleatorio y se produce por perturbaciones electromagnéticas, tormentas atmosféricas, fallos o defectos en los sistemas de comunicación. En transmisión de señales digitales suele ser responsable de ráfagas de bits erróneos.

Relación señal ruido (S/N, signal to noise ratio): Indica la razón de la potencia de la señal (P_S , Signal) respecto a la potencia del ruido (P_N , Noise).

$$S/N = \frac{P_S}{P_N} \quad (7)$$

La relación S/N debe permanecer a un determinado nivel para mantener la señal de datos separada de la señal de ruido.

Es importante tener en cuenta que cuando se amplifica la señal, también se amplifica el ruido, por lo que la elección de la distancia entre los amplificadores para mitigar el efecto



de la atenuación es una decisión importante. Cuanto mayor es la relación S/N, mejor es el canal.

Normalmente la relación S/N es muy alta y se usa la unidad decibelio (dB) para representarla.

$$S/N_{dB} = 10\log\left(\frac{P_S}{P_N}\right) \quad (8)$$

Resumen

Para mitigar el efecto de la atenuación suelen utilizarse amplificadores (cuando se transmiten señales analógicas) o repetidores (cuando se transmiten señales digitales) cada cierta distancia que regeneran y retransmiten la señal digital antes de que se atenúe mucho. En definitiva, para conseguir una buena comunicación es necesario:

- Que la señal tenga suficiente potencia para que el receptor la detecte e interprete adecuadamente.
- Que la señal conserve un nivel suficientemente mayor que el ruido para ser recibida sin error.

1.3 Ancho de banda de un canal

Cuando una señal pasa a través de un medio físico siempre sufre una alteración de la forma de la señal. El ancho de banda de un sistema de transmisión es aquel rango de frecuencias (en Hz) transmitido sin ser fuertemente atenuado.

El ancho de banda de un canal es una propiedad física del medio físico, por ejemplo, tipo, anchura y longitud del cable (aunque a veces se añaden filtros software para disminuir "artificialmente" el ancho de banda de un sistema).

Por efecto de este ancho de banda:

- Las componentes de frecuencia de nuestra señal que están dentro de este rango sufren atenuaciones de hasta 3dB (es decir, la potencia de la señal disminuye).
- Las componentes de frecuencia que están fuera de este rango no son transmitidas, el canal actúa como un filtro para todas las frecuencias que están fuera de ese rango. Se dice que el canal se comporta como un filtro pasabanda.

Si el ancho de banda pasante fuese infinito, la señal no sufriría ninguna atenuación ni distorsión, pero en la práctica esto es totalmente imposible.

Además se cumple que cuanto *mayor es el ancho de banda, más información podemos enviar*, esto se traduce en la práctica en: Se pueden enviar más señales simultáneamente, o el ancho de banda de la señal que se envía es mayor y por lo tanto la precisión de la información en el receptor será mejor o la velocidad de transmisión de la señal sea mayor.

También se cumple que a menor ancho de banda mayor es el efecto de las distorsiones y atenuaciones.

Se puede hacer una representación espectral de la señal a transmitir y del canal por donde se va a transmitir. Si ambos espectros coinciden, la señal se puede transmitir tal cual por ese canal, si no coinciden, hay que transformar (modular) la señal antes de transmitirla.

Cuando se transmite información digital sobre señales analógicas se usan las técnicas conocidas como **modulación digital** y las veremos en la práctica 3.



Por ejemplo, para transmitir la señal de la izquierda de la Figura 8 que presenta un ancho de banda relativo entre 0 y 10 Hz mediante Bluetooth¹, que se basa en señales de radiofrecuencia cuyo ancho de banda del canal es 2.4 GHz a 2.5 GHz es necesario transformarla a frecuencias superiores antes de transmitirla.

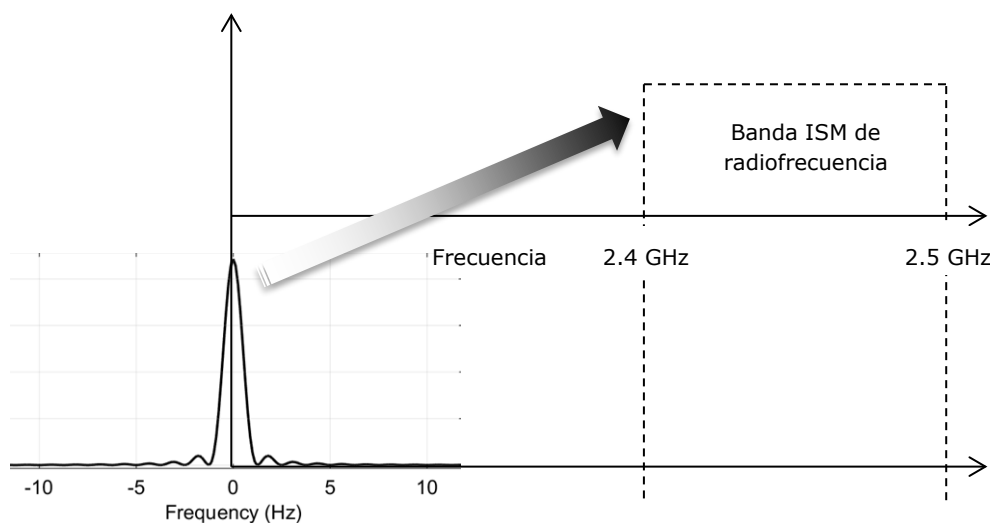


Figura 8. Transmisión mediante modulación digital de la señal.

1.4 Transmisión en banda base

Al contrario de lo que sucede con el proceso de modulación, en el que se realiza un desplazamiento del espectro de frecuencias de la banda base hacia frecuencias superiores, en la transmisión en banda base se preserva el espectro de frecuencia original utilizando una codificación especial para adaptar las señales a las líneas de transmisión.

En la transmisión en banda base se transmite información digital sobre señales digitales, por tanto, las señales se transmiten tal cual, lo que se hace es utilizar técnicas especiales para codificar la información que queremos transmitir, y una vez codificada no se altera. Estas técnicas se conocen como **codificación digital** y las veremos en la práctica 3.

Este tipo de transmisión es muy popular debido al bajo costo de los circuitos digitales y la flexibilidad de la aproximación digital.

Como ya se ha comentado mediante las técnicas de transmisión en banda base lo que se hace es codificar la información de una forma determinada que optimice determinadas características. En general, el principal problema que plantea una transmisión digital ON/OFF (secuencia de unos y ceros) es que su espectro de frecuencias tiene infinitas componentes por lo que se necesita transmitir hasta frecuencias muy altas. Si tenemos en cuenta que los medios físicos de transmisión tienen un ancho de banda finito y que todos ellos introducen una atenuación de 3dB, tendremos que para frecuencias altas habrá mucha distorsión.

1.5 Capacidad de un canal

La capacidad máxima de un canal es la proporción de información que se puede enviar por una línea. La capacidad depende del ancho de banda de la línea. Esto fue demostrado por Hartley en 1928.

¹ ISM (Industrial, Scientific and Medical) son bandas reservadas internacionalmente para uso no comercial de radiofrecuencia electromagnética en áreas industrial, científica y médica. En la actualidad estas bandas han sido popularizadas por su uso en comunicaciones Wi-Fi y Bluetooth.



Todas las fórmulas y desarrollos para el estudio de la capacidad de un canal y su relación con el ruido fueron desarrolladas por Hartley, Nyquist y en los años 40-50 por Shannon.

Capacidad de un canal sin ruido (canal ideal)

Teorema de Nyquist. Supongamos un canal ideal con un ancho de banda B (en Hz) y sin ruido alguno, la **capacidad máxima C** (en bits/s o bps) del canal se puede calcular como:

$$C = 2B \log_2 n \quad (9)$$

Donde n es el número de diferentes valores o niveles de tensión que se utilizan para codificar la señal (o número de estados posibles de señalización en una línea).

Por tanto, aumentando el número de niveles, podemos aumentar indefinidamente la capacidad de un canal, siempre y cuando no haya ruido.

Capacidad de un canal con ruido (canal real)

Teorema de Shannon. Supongamos un canal de comunicaciones cuyo ancho de banda es B (en Hz) y con ruido blanco, la **capacidad máxima C** (en bits/s o bps) del canal se puede calcular como:

$$C = B \log_2(1 + S/N) \quad (10)$$

Donde S/N es la relación señal ruido, la potencia de la señal dividido por la potencia del ruido, medida en watio/watio, **NO en decibelios**.

Al observar la relación anterior podría parecer que la capacidad de un canal puede crecer de forma infinita, ya que si aumentamos el ancho de banda de forma arbitraria podría parecer que la capacidad también aumentará. No obstante, el ruido blanco es un elemento inherente al sistema de comunicaciones y al aumentar el ancho de banda, el ruido blanco también aumenta, por eso la relación señal ruido disminuye y la relación total de la capacidad se mantiene constante.

El teorema anterior sólo considera ruido térmico (ruido blanco) pero no considera el ruido impulsivo, la atenuación ni la distorsión de retardo, por lo que en la práctica se consiguen razones mucho menores.

Teorema de Shannon/Hartley

Shannon demostró que todo canal de comunicaciones tiene una capacidad máxima.

TEOREMA: Cuando la velocidad de transferencia de datos V_t es menor que C , la probabilidad de que haya un error en la transmisión se aproxima a cero cuando se utilizan técnicas apropiadas para la codificación de la información.

TEOREMA: Si la velocidad de transmisión V_t es mayor que C , la tasa de error siempre será mayor que cero independientemente de la técnica de codificación empleada.

1.6 Transmisión analógica versus transmisión digital

La transmisión mediante señales analógicas es más fácil de implementar porque se actúa menos sobre la señal que se quiere transmitir.

La transmisión mediante señales digitales requiere de sistemas digitales más complejos (hardware y software) para transmitir la información.

La transmisión mediante señales digitales, es más económica que la analógica, a la vez de ser menos susceptible a las interferencias de ruido.

Las señales digitales sufren más con la atenuación que las señales analógicas.



1.7 Tipos de medios físicos de transmisión

Medio físico de transmisión: es el canal que permite la transmisión de información entre dos terminales en un sistema de transmisión

Los datos se propagan de un punto a otro mediante señales:

- Digitales, es decir, una secuencia de pulsos de tensión que se transmiten **solo** a través de un soporte sólido y además conductor.
- Analógicas, es decir, una onda electromagnética que varía continuamente y puede transmitirse sobre un medio sólido (conductor o no) o sobre un soporte no sólido

Dependiendo de la forma de conducir la señal a través del medio, el medio de transmisión puede ser:

- Guiado: Necesita un soporte sólido, cable de par trenzado(conductor), cable coaxial (conductor), fibra óptica (vidrio, no conductor).
- No guiado: No necesita un soporte sólido, aire, agua, vacío.

Las características y calidad de la transmisión dependen del tipo de señal y tipo de medio físico, pero para medios guiados, dependen fundamentalmente del medio físico y para medios no guiados dependen fundamentalmente de la señal.

Cada medio tiene distintas características: ancho de banda, retardos en la transmisión, atenuación, interferencias, costes, facilidad de instalación y de mantenimiento, etc.

Espectro electromagnético y señales analógicas

Cuando estamos transmitiendo señales analógicas, éstas son siempre señales electromagnéticas, por tanto el espectro electromagnético es la clasificación de las ondas electromagnéticas en función de su frecuencia de oscilación. En la Figura 9 se han representado el espectro electromagnético solo de aquellas señales usadas en comunicaciones y se puede ver también como diferentes tipos de ondas electromagnéticas pueden usar diferentes medios físicos de transmisión.

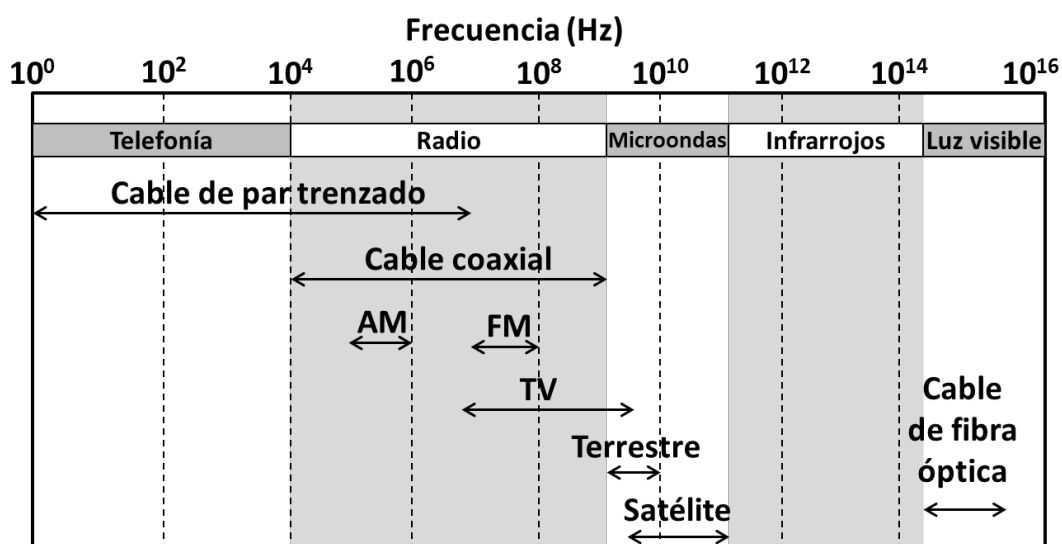


Figura 9. Espectro electromagnético de las señales usadas en comunicaciones.

Todos los fenómenos de la electricidad y el magnetismo derivan de la carga eléctrica. El mecanismo fundamental de la emisión de una onda electromagnética es la aceleración de una partícula cargada eléctricamente (electrón).



Siempre que una partícula cargada se acelera, radia energía. Dependiendo del fenómeno físico que intervenga y material del que provengan los electrones se generaran diferentes tipos de ondas electromagnéticas:

- Aceleración de electrones libres en la corteza de metales: Ondas de radio y microondas.
- Aceleración de electrones por salto entre bandas en semiconductores: Infrarrojos y luz visible.

Ondas de radio, microondas y corriente alterna

Se denomina corriente alterna (CA o AC) a la corriente eléctrica en la que la magnitud y el sentido varían cíclicamente. Cuando cualquier objeto conductor eléctrico (*metal*) conduce corriente alterna, genera una radiación electromagnética que se propaga en la misma frecuencia que la corriente.

Cuando una radiación electromagnética incide en un conductor eléctrico, hace que los electrones libres de su superficie oscilen, generándose de esta forma una corriente alterna cuya frecuencia es la misma que la de la radiación incidente.

Las ondas de radio son las ondas electromagnéticas que se transmiten por los medios guiados: cable de par de cobre trenzado y cable coaxial.

Este efecto se usa en las antenas, que pueden actuar como emisores o receptores de radiación electromagnética. Por tanto, las ondas de radio son ondas electromagnéticas que pueden transmitirse también por medios no guiados como el aire.

Sin embargo, las microondas son ondas electromagnéticas que se transmiten **solo** por medios no guiados.

Infrarrojos y luz visible

Cuando se aplica energía a un electrón de la banda de valencia en un *semiconductor*, éste puede pasar a la banda de conducción dejando el correspondiente hueco. Los electrones pueden caer desde el estado energético más elevado correspondiente en la banda de conducción a un hueco en la banda de valencia, liberando así energía, es decir generando una onda electromagnética, ver Figura 10.

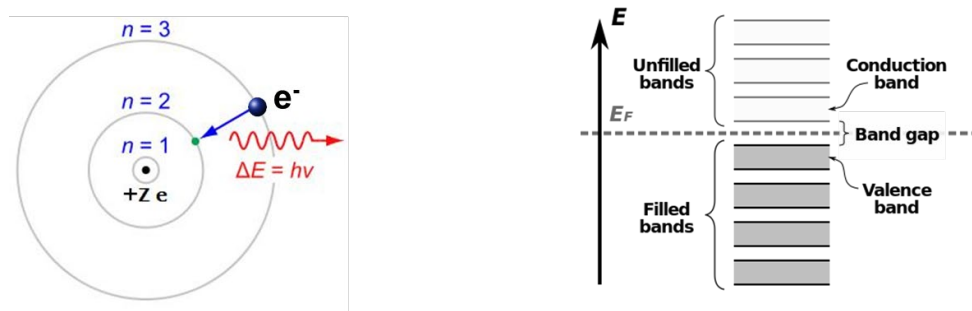


Figura 10. Generación de una onda electromagnética en un semiconductor.

- Los infrarrojos son ondas electromagnéticas que solo se transmiten por medios no guiados.
- La luz visible son ondas electromagnéticas que se pueden transmitir por medios guiados: fibra óptica.



1.8 Medios de transmisión guiados

Los medios de transmisión guiados están constituidos por un soporte material (un cable) que se encarga de la conducción (o guiado) de las señales desde un extremo al otro, los principales tipos son: *Par trenzado*, *cable coaxial* y *fibra óptica*.

Algunos como el par trenzado se usan para transmitir tanto *señales analógicas* como *señales digitales*, el resto, la fibra óptica y el cable coaxial se usan solo para transmitir *señales analógicas* (el cable coaxial se usó hace décadas para transmitir señales digitales, pero en la actualidad ya no), ver Figura 9.

Las principales características de los medios guiados son:

- El tipo de conductor utilizado.
- La velocidad máxima de transmisión.
- Las distancias máximas que puede ofrecer entre repetidores.
- La inmunidad frente a interferencias electromagnéticas.
- La facilidad de instalación.
- La capacidad de soportar diferentes tecnologías de nivel de enlace.

Cables de cobre

El telégrafo (1830) se basaba en un único hilo conductor, ver Figura 11, tendido sobre postes entre el emisor y el receptor. La energía que se propaga hacia arriba y hacia los lados se pierde, la que va hacia abajo se refleja en el suelo volviendo al hilo conductor y así sucesivamente, en este caso la Tierra actúa como un segundo conductor!!!

Los sucesivos rebotes hacen que las ondas portadoras de la información sean perceptibles a distancias razonables, los principales problemas de este medio de transmisión son:

- Es necesario usar ondas de muy baja frecuencia para que la Tierra actúe como plano conductor. Poca capacidad para transmitir información
- La Tierra es un conductor muy malo y la separación con el cable implica pérdidas importantes de energía. Es necesario transmitir a una potencia muy elevada
- Múltiples interferencias entre hilos contiguos, otros dispositivos, etc.

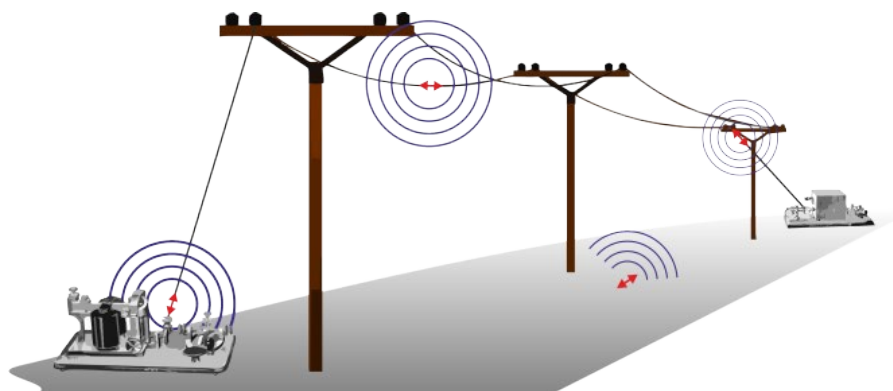


Figura 11. Esquema del telégrafo.

Más adelante surgió el cable bifilar que se basa en dos conductores en paralelo con un aislante (tira de plástico) entre los dos, que mejora el guiado de las ondas, ver Figura 12.

Usar dos cables conductores disminuye las pérdidas energéticas. Al manejar mejor la energía, el cable bifilar puede guiar ondas mucho más lejos que el cable único además usar dos cables conductores juntos permite aumentar el rango de frecuencias y por tanto aumentar la capacidad de transmisión. Sin embargo:



- A partir de pocos MHz las pérdidas de energía por calentamiento muy elevadas.
- Muy sensible a las interferencias de otras ondas electromagnéticas. Presencia de Diafonía (crosstalk).

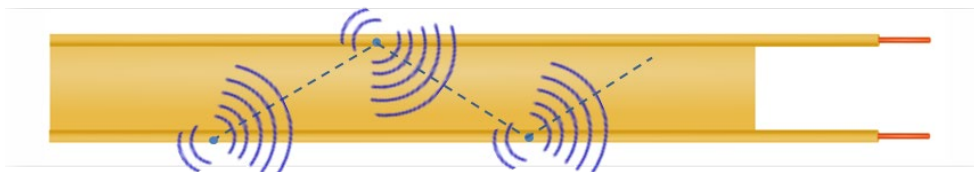


Figura 12. Cable bifilar.

Finalmente, en 1881, Alexander Graham Bell patentó una solución para combatir las interferencias, enrollando los dos hilos conductores el uno sobre el otro, envueltos ambos por un aislante. El cable pasa a llamarse par trenzado y es el que se usa hoy en día.

El efecto de cualquier interferencia se reparte casi por igual entre los dos hilos y entonces las dos perturbaciones prácticamente se anulan la una a la otra.

Cable de par trenzado

Es el medio más antiguo en el mercado (Figura 13) y en algunos tipos de aplicaciones es el más común. Consiste en dos alambres de cobre aislados y de un grosor de 1 milímetro. Se usa tanto para transmitir señales analógicas como digitales.

Un cable de par trenzado está formado por un grupo de pares trenzados recubiertos por un material aislante. Este cable se enrolla a su vez con otros cables. El ancho de banda del cable es entorno al MHz.

Normalmente el cable está formado por 4 pares que se identifican mediante colores:

- Par 1: Blanco – Azul / Azul
- Par 2: Blanco – Naranja / Naranja
- Par 3: Blanco – Verde / Verde
- Par 4: Blanco – Marrón / Marrón

Se conecta con otros dispositivos mediante conectores RJ (Registered Jack o "clavija registrada)

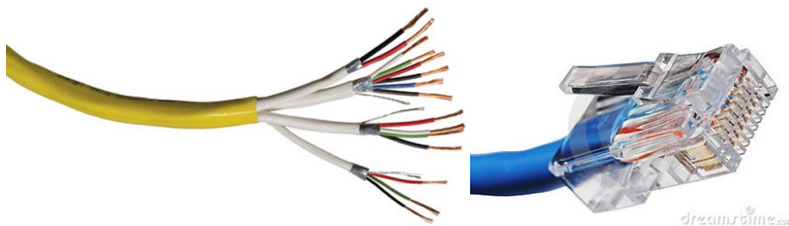


Figura 13. Cable de cobre de par trenzado y clavija RJ45.

Cable coaxial

Creado en la década de los 30. Se usa tanto para transmitir señales analógicas como digitales y posee dos conductores concéntricos:

- Uno central, llamado positivo, encargado de llevar la información.
- Uno exterior, de aspecto tubular, llamado malla o blindaje, que sirve como referencia de tierra y retorno de las corrientes.
- Entre ambos se encuentra una capa aislante llamada dieléctrico, de cuyas características dependerá principalmente la calidad del cable.
- Todo el conjunto suele estar protegido por una cubierta aislante.



La construcción y el blindaje del cable coaxial le confieren una buena combinación de ancho de banda alto y excelente inmunidad al ruido. El ancho de banda del cable es entorno a GHz.

Solían utilizarse en el sistema telefónico para las líneas de larga distancia y actualmente se utiliza en la televisión por cable.

El cable coaxial puede utilizarse como un medio compartido, es decir, una serie de sistemas terminales pueden estar conectados directamente al cable, recibiendo todos ellos lo que envíen los otros sistemas terminales.

Se conecta con otros dispositivos mediante conectores BNC, BNC-T (Bayonet Neill-Concelman, nombres de los diseñadores en 1951). Al final de línea ha de ponerse un terminador BNC.

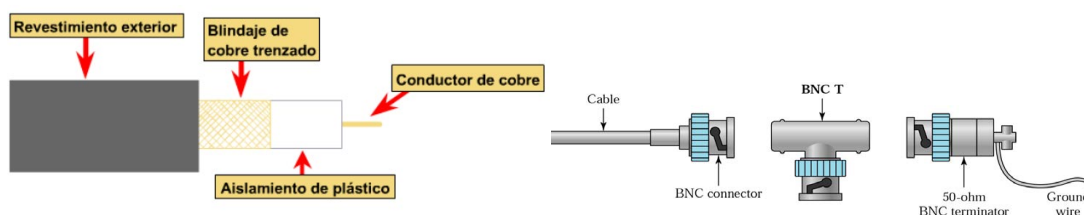


Figura 14. Cable coaxial y conectores BNC.

Fibra óptica

Los cables de fibra óptica son similares a los coaxiales (Figura 15):

- En el centro se encuentra el núcleo de vidrio a través del cual se propaga la luz.
- El núcleo está rodeado por un revestimiento de vidrio como índice de refracción menor que el del núcleo, con el fin de mantener toda la luz en este último.
- A continuación está una cubierta plástica delgada para proteger el revestimiento.
- Las fibras por lo general se agrupan en haces protegidas por una funda exterior.

El haz de luz queda completamente confinado y se propaga por el interior de la fibra con un ángulo de reflexión por encima del ángulo límite de reflexión total

Cada trayectoria que puede seguir un haz de luz en el interior de una fibra se denomina modo de propagación:

- **Fibra multimodo**, se propagan varios modos de luz. No todos los haces llegan a la vez. Cortas distancias, núcleo más grueso, es simple de diseñar y económico.
- **Fibra monomodo**, se propaga solo un modo de luz. Se reduce el diámetro del núcleo (8,3 a 10 micrones). Grandes distancias (400 km) y transmisión de altas tasas de información (decenas de Gbit/s).

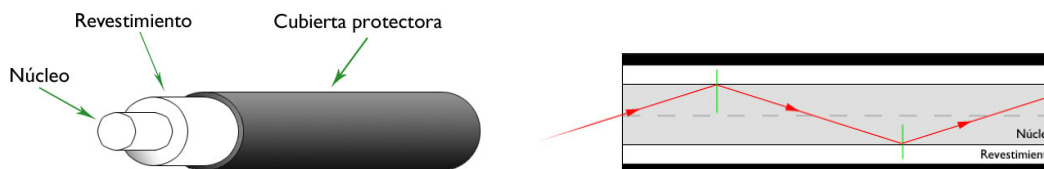


Figura 15. Fibra óptica.

Un sistema de transmisión óptico tiene tres componentes, Figura 16:

- La fuente de luz puede ser láser o un led, un pulso de luz indica un bit 1 y la ausencia de luz indica un bit 0
- El medio de transmisión es una fibra de vidrio.
- El detector genera un pulso eléctrico cuando la luz incide en él.



Al agregar una fuente de luz en un extremo de una fibra óptica y un detector en el otro, se tiene un sistema de transmisión de datos que acepta una señal eléctrica, la convierte y transmite mediante pulsos de luz y luego reconvierte la salida a una señal eléctrica en el extremo receptor.

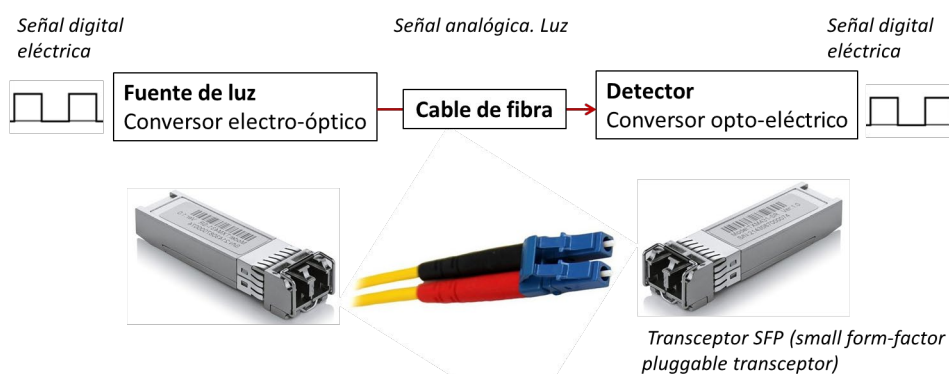


Figura 16. Sistema de transmisión óptico.

La fibra óptica es un medio flexible y de poco espesor que conduce pulsos de luz, representando cada pulso un bit.

- Un único cable de fibra óptica puede soportar velocidades de bit tremendamente altas, por encima de decenas o incluso centenas de gigabits por segundo.
- Ancho de banda entorno a THz.
- La fibra óptica es inmune a las interferencias electromagnéticas.
- Presenta una atenuación de la señal muy baja hasta una distancia de 100 kilómetros y es muy difícil que alguien pueda llevar a cabo un "pinchazo" en una de estas líneas.
- Medio de transmisión guiado a larga distancia preferido, especialmente para los enlaces transoceánicos. Muchas de las redes telefónicas para larga distancia utilizan hoy día exclusivamente fibra óptica. La fibra óptica también es el medio predominante en las redes troncales de Internet.
- Alto coste de los dispositivos ópticos, como transmisores receptores y conmutadores.
- Alto coste de mantenimiento.

Los conectores más comunes usados en la fibra óptica para redes de área local son los conectores ST, LC, FC Y SC. El conector SC (Set and Connect) es un conector de inserción directa que suele utilizarse en conmutadores Ethernet de tipo Gigabit, Figura 17.

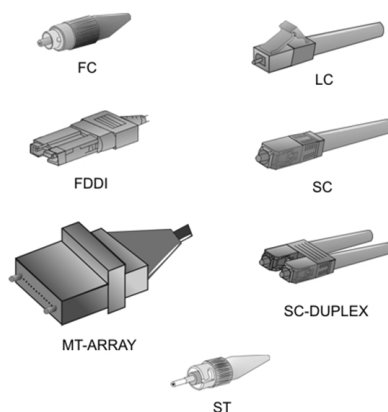


Figura 17. Conectores para cable de fibra óptica.



1.9 Medios de transmisión no guiados

Sirven para transmitir solo señales analógicas, Figura 9. Las señales se propagan libremente a través del medio. Los medios más importantes son el aire, agua y el vacío. Las señales que se usan fundamentalmente para la comunicación son *ondas de radio* y *microondas*. Tanto la transmisión como la recepción de información se lleva a cabo mediante antenas:

- **Transmisión**, la antena irradia energía electromagnética en el medio.
- **Recepción**, la antena capta las ondas electromagnéticas del medio que la rodea.

La configuración para las transmisiones no guiadas puede ser:

- **Direccional**, la antena transmisora emite la energía electromagnética concentrándola en un haz, por lo que las antenas emisora y receptora deben estar alineadas. Propio de altas frecuencias. Arquitectura punto-punto
- **Omnidireccional**, la antena transmisora emite en todas direcciones, pudiendo la señal ser recibida por varias antenas. Propio de bajas frecuencias. Arquitectura punto-multipunto

Generalmente, cuanto mayor es la frecuencia de la señal transmitida es más fácil confinar la energía en un haz direccional

Ondas de radio

Cubre el espectro de frecuencias entre 10 kHz y 1 GHz. Características:

- Omnidireccionalidad. Propagación en todas las direcciones.
- La atenuación depende de la distancia recorrida por la señal (pérdida de camino) y su frecuencia.
- La transmisión se ve afectada por condiciones climatológicas, lluvia, niebla, etc.
- Interferencias debidas a otras señales electromagnéticas domésticas.
- Interferencias por multitrayectorias, se generan otras trayectorias por la reflexión de la señal en la superficie terrestre, mar, otros objetos,... que interfieren con la señal original.
- Se reflejan en la ionosfera (entre 3 MHz y 0.3 GHz) por lo que pueden transmitirse más allá del horizonte ver Figura 18.
- Atraviesan bien los sólidos.

Se usa fundamentalmente para Radio comercial AM y FM y Televisión VHF y UHF.

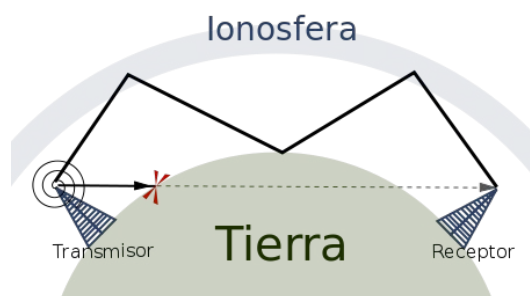


Figura 18. Transmisión de las ondas de radio más allá del horizonte.



Microondas

Cubre el espectro de frecuencias entre 1 GHz y 300 GHz. Características

- Omnidireccionalidad. Propagación en todas las direcciones.
- Direccionalidad. Es posible para frecuencias altas conseguir la propagación en una única dirección (un haz estrecho).
- La atenuación depende de la distancia recorrida por la señal (pérdida de camino) y su frecuencia.
- La transmisión se ve afectada por condiciones climatológicas, lluvia, niebla, etc.
- Interferencias debidas a otras señales de tipo microondas muy habituales hoy en día.
- Interferencias por multitrayectorias, se generan otras trayectorias debidas a la reflexión de la señal en la superficie terrestre, mar, otros objetos, etc. que interfieren con la señal original.
- No atraviesan bien los sólidos.

Tres amplios grupos para las comunicaciones (entre 1 GHz y 30 GHz):

- **Área local**, alcance de cientos de metros, usado en redes LAN inalámbricas. WiFi (2.4 GHz y 5 GHz) y Bluetooth (2.4 GHz).
- **Área extensa**, alcance de decenas de kilómetros, usado en redes Móviles. GSM (0.9-1.8 MHz), 3G (2.1 GHz), 4G (1.8 GHz y 2.6 GHz) y WiMAX (3.5 GHz).
- **Comunicaciones vía satélite**. Usan los rangos de frecuencia entre 1.5 GHz y 30 GHz.

En el caso de las comunicaciones vía satélite (Figura 19), éstas enlazan dos o más transmisores/receptores de microondas con base en la Tierra, que se conocen como estaciones terrestres. El satélite recibe las transmisiones en una banda de frecuencia, regenera la señal utilizando un repetidor y transmite la señal a otra frecuencia.

En este tipo de comunicaciones se emplean dos tipos de satélites:

- Satélites geoestacionarios (GEO), están permanentemente situados en el mismo punto por encima de la Tierra a 36000 km. Retardo de propagación de la señal de 280 milisegundos. Permiten el acceso a Internet a velocidades de cientos de Mbps. El rango de frecuencias usados se muestra en la Tabla 1.
- Satélites LEO (Low-Earth Orbiting), se colocan mucho más cerca de la Tierra y no se encuentran permanentemente en la misma posición, sino que giran alrededor de la Tierra y pueden comunicarse entre sí, así como con las estaciones terrestres. Para poder proporcionar una cobertura continua a un área, es preciso poner en órbita muchos satélites. Se espera que en un futuro se puedan utilizar para acceder a Internet.



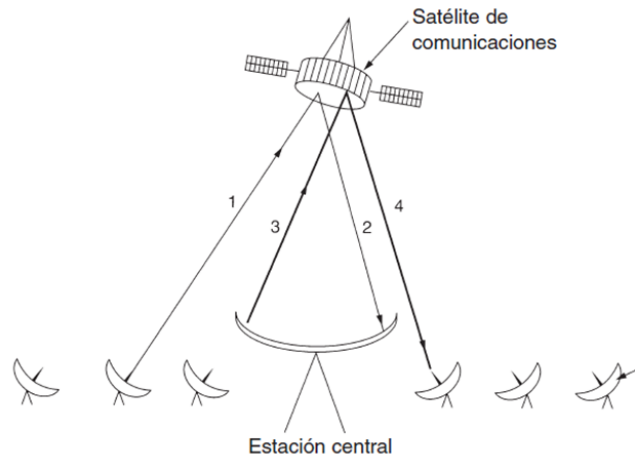


Figura 19. Comunicaciones vía satélite.

Tabla 1. Rango de frecuencias usadas en comunicaciones vía satélite.

Banda	Enlace bajada	Enlace subida	Ancho de banda	Problemas
L	1.5 GHz	1.6 GHz	15 MHz	Poco ancho de banda Congestionada
S	1.9 GHz	2.2 GHz	70 MHz	Poco ancho de banda Congestionada
C (*)	4 GHz	6.0 GHz	500 MHz	Interferencias con comunicaciones inalámbricas terrestres
Ku	11 GHz	14 GHz	500 MHz	La ondas son absorbidas por la lluvia
Ka	20 GHz	30 GHz	3500 MHz	La ondas son absorbidas por la lluvia Equipo caro

Infrarrojos

Cubre el espectro de frecuencias entre 300 GHz y 450 THz, ver Figura 9. Características:

- Direccionalidad. Se propagan solo en una única dirección. El emisor y el receptor deben estar completamente alineados.
- La atenuación depende de la distancia recorrida por la señal (pérdida de camino) y su frecuencia.
- No atraviesan sólidos.
- Apenas hay interferencias con otras señales electromagnéticas.

Se usa fundamentalmente para Comunicar pequeños dispositivos electrónicos, periféricos, etc. (319 THz).



2 Estudio de transmisión en banda base con Simulink

En este estudio con Simulink nos vamos a centrar en la transmisión de información digital mediante señales digitales, lo que se conoce como transmisión en banda base. En todos los casos usaremos la codificación digital más sencilla, es decir, al bit 0 que queramos transmitir le asignamos un voltaje de 0 voltios y al bit 1 que queremos transmitir le asignamos un voltaje distinto de 0 y positivo. En algunos casos le asignaremos 1 voltio y en otros 1.5 voltios.

Se proporciona los ficheros correspondientes de cada apartado para poder realizar la práctica y contestar a las preguntas correspondientes.

2.1 Velocidad de transmisión y ancho de banda de la señal

En este ejercicio, ver Figura 20, vamos a trabajar con tres pulsos de distintos periodos y las representaremos en el tiempo y en la frecuencia:

- Un tren de pulsos de amplitud 1 voltio y periodo de 1 segundo.
- Un tren de pulsos de amplitud 1 voltio y periodo de 0.5 segundos.
- Un tren de pulsos de amplitud 1 voltio y periodo de 0.25 segundos.

Usaremos la misma configuración del analizador de espectros (Spectrum Analyzer) usada en la práctica 1, para visualizar el espectro de las tres señales.

La duración de la transmisión será de 8 segundos y en todos los casos en la configuración del generador de pulsos "Pulse Generator" se usará un ancho de pulso del 50 %, es decir la mitad del periodo el pulso valdrá 0 y la otra mitad el pulso valdrá 1.

El modelo lo grabaremos en un fichero de nombre "**Prac02_2_1.slx**".

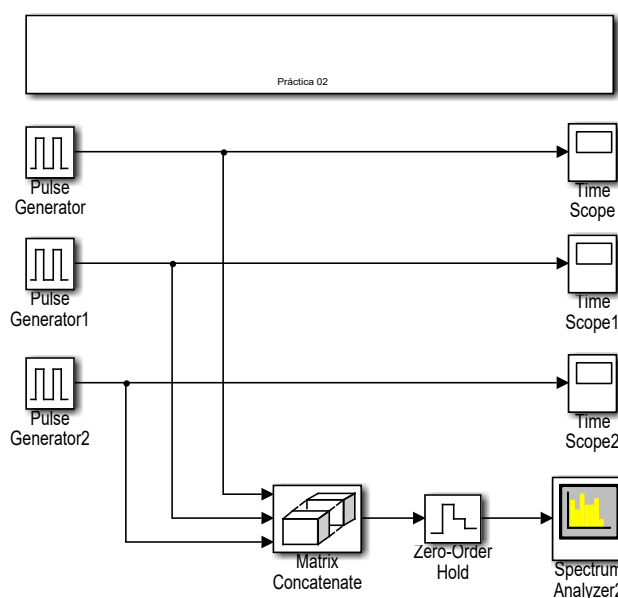


Figura 20. Modelo efecto de la velocidad de transmisión en el ancho de banda de la señal.



Tiempo máximo de la simulación: 8 segundos	
Nombre del modelo: Prac02_2_1.slx	
Bloque:	Time Scope, Time Scope1 y Time Scope2
Paleta:	DSP System Toolbox -> Sinks
Parámetros:	Ninguno
Bloque:	Pulse Generator
Paleta:	Simulink -> Sources
Pulse type:	Time based
Time:	Use simulation time
Amplitude:	1
Period (sec):	1
Pulse Width (% of Period):	50
Phase delay (sec):	0
Bloque:	Pulse Generator1
Paleta:	Simulink -> Sources
Pulse type:	Time based
Time:	Use simulation time
Amplitude:	1
Period (sec):	0.5
Pulse Width (% of Period):	50
Phase delay (sec):	0
Bloque:	Pulse Generator2
Paleta:	Simulink -> Sources
Pulse type:	Time based
Time:	Use simulation time
Amplitude:	1
Period (sec):	0.25
Pulse Width (% of Period):	50
Phase delay (sec):	0
Bloque:	Matrix Concatenate
Paleta:	DSP System Toolbox -> Math Functions -> Matrices and Linear Algebra -> Matrix Operations
Number of inputs:	3
Mode:	Multidimensional array
Concatenate dimensión:	2
Bloque:	Zero-Order Hold
Paleta:	Simulink -> Discrete
Sample time:	0.0025
Bloque:	Spectrum Analyzer
Paleta:	DSP System Toolbox -> Sinks
Main Options:	Type:Power; Full Frequency Span; Window length: 3200
Trace Options:	Units: Watts
Parámetros:	Ver práctica 1



Preguntas

- Visualizar las señales en el tiempo.
- Visualizar el espectro de frecuencias de las señales e identificar la frecuencia fundamental de cada señal.
- Rellenar la siguiente tabla:

Nota: En esta práctica y debido a que todas las señales son digitales y periódicas se ha escogido como definición (arbitraria) de ancho de banda relativo de la señal, aquella banda de frecuencias que contiene las 11 primeras componentes de frecuencia. Es decir, asumimos que estas 11 primeras componentes representan adecuadamente la señal, siendo las siguientes componentes de frecuencia no muy relevantes en la señal.

Señal	Pulso (periodo 1 seg.)	Pulso (periodo 0.5 seg.)	Pulso (periodo 0.25 seg.)
Periodo (s)	1	0.5	0.25
Frecuencia fundamental (Hz)			
Número total de bits transmitidos			
Número de símbolos diferentes			
Tipo de símbolos diferentes			
Número total de símbolos transmitidos			
Intervalo de tiempo más corto (s)			
Velocidad de transmisión serie (baudios)			
Velocidad de transmisión (bps)			
Ancho de banda absoluto (Hz)			
Ancho de banda relativo (Hz) considerando solo las 11 primeras componentes de frecuencia			

- Vista la tabla anterior ¿Cuál es la relación entre la velocidad de transmisión y el ancho de banda relativo de cada señal?
- Se desea transmitir dichas señales por un canal con ancho de banda de 0 a 3 Hz, ¿qué ocurrirá?



2.2 Velocidad de transmisión y ancho de banda del canal

En este apartado se pretende estudiar el efecto del ancho de banda del canal sobre la velocidad de transmisión y sobre la calidad de la señal transmitida. Para ello usaremos el modelo de la Figura 22 junto con tres señales digitales de prueba:

- Un tren de pulsos de amplitud 1 voltio y periodo de 1 segundo dado por el bloque "Pulse generator".
- Un tren de pulsos de amplitud 1 voltio y periodo de 0.25 segundos dado por el bloque "Pulse generator1".
- Un tren de pulsos de amplitud 1 voltio y periodo de 1 segundo dado por el bloque "Pulse generator2".

Además usaremos un componente filtro, que emula el comportamiento de un canal de transmisión (por ejemplo un cable de cobre) y como afecta a las señales transmitidas. Usaremos la misma configuración del analizador de espectros (Spectrum Analyzer) usada en la práctica 1, para visualizar el espectro de las señales, **excepto** que como en el estudio de transmisión de señales por canales solo nos interesan las frecuencias positivas, podemos desactivar la opción "Two-sided spectrum", ver Figura 21.

La duración de la transmisión será de 8 segundos y en todos los casos en la configuración del generador de pulsos "Pulse Generator" se usará un ancho de pulso del 50 %, es decir la mitad del periodo el pulso valdrá 0 y la otra mitad el pulso valdrá 1.

El modelo lo grabaremos en un fichero de nombre "Prac02_2_2.slx".

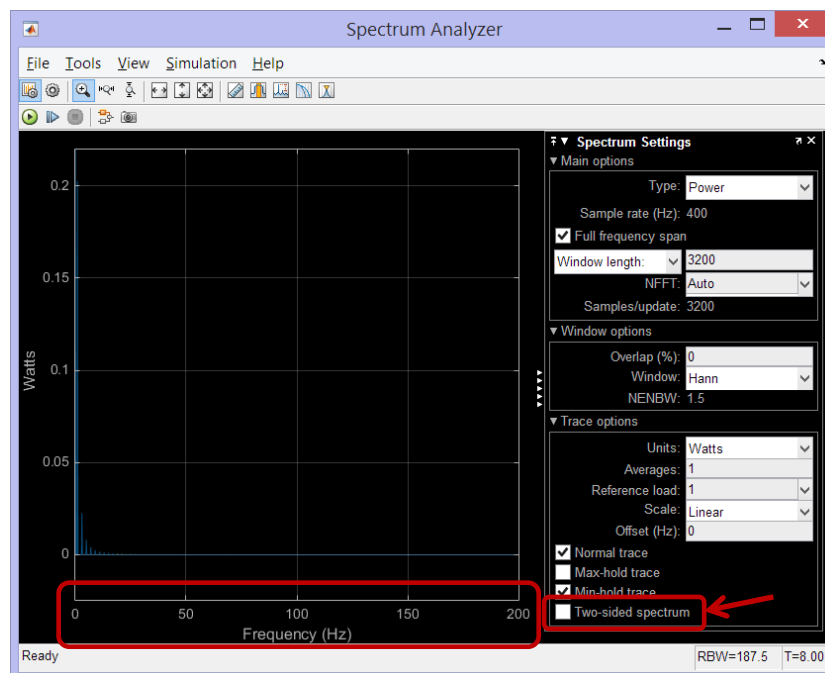


Figura 21. Configuración del analizador del espectro para visualizar solo frecuencias positivas, que son las únicas que tienen sentido físico.

Estudio inicial

Inicialmente usaremos el modelo superior de la Figura 22 para simular el comportamiento de un canal que deja pasar las bajas frecuencias y no deja pasar las altas frecuencias, usaremos un filtro analógico pasa baja. Un canal que se comporte de esta manera puede ser el *cable de cobre*. Usaremos por tanto el bloque en Simulink "Analog Filter Design", tipo "Lowpass" (pasa baja) y diseño "Chebyshev II" orden 8.



Importante: La frecuencia de corte, es decir la frecuencia a partir de la cual no se deja pasar componentes de frecuencia la fijaremos en 20 Hz (o $2\pi 20$ rad/s). Por tanto el ancho de banda del canal será de 20 Hz ($20 \text{ Hz} - 0 \text{ Hz}$).

Sobre este canal de transmisión transmitiremos en banda base el tren de pulsos de 1 segundo de periodo, dado por el bloque "Pulse generator". Esto significa que la velocidad de transmisión es de 2 bps y que la frecuencia fundamental de este tren de pulsos es 1 Hz.

Veremos en el espectro de frecuencias como el canal no deja pasar frecuencias de más de 20 Hz, es decir, hace cero todas las componentes de frecuencia superiores a 20 Hz en el receptor. También podemos ver, haciendo zoom en el espectro como el número de componentes de frecuencia que deja pasar es 11, esto da una idea de la calidad de la señal recibida, para ello comparar la señal transmitida el transmisor con la señal recibida en el receptor ambas en función del tiempo en el bloque "Time Scope".

Importante: Claramente la señal digital recibida **NO** es igual que la señal transmitida. Este efecto es debido al canal de transmisión y que en este caso hemos emulado con el bloque filtro.

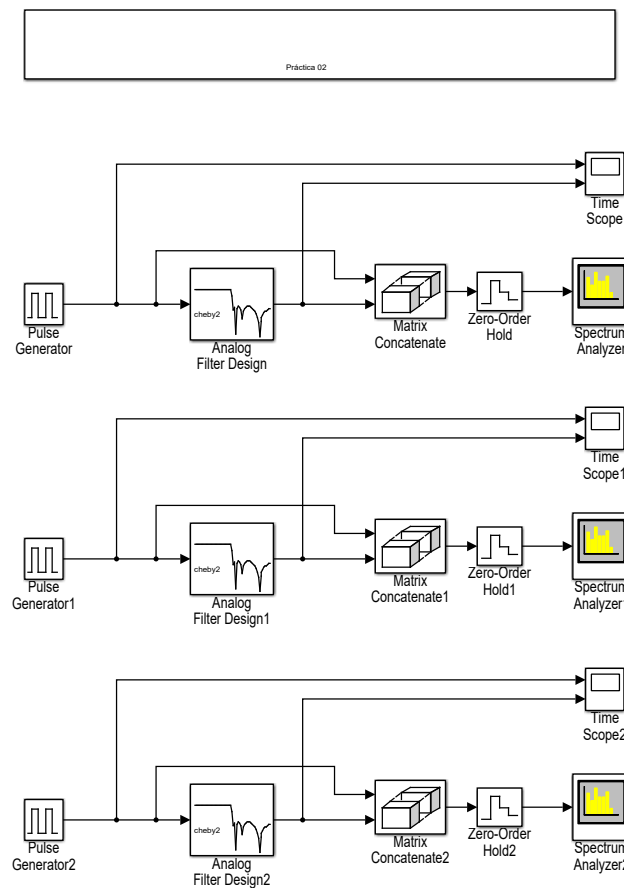


Figura 22. Modelo para la relación entre velocidad de transmisión y ancho de banda del canal. En orden de arriba abajo: Modelo 1, modelo 2 y modelo 3.



Tiempo máximo de la simulación: 8 segundos	
Nombre del modelo:	Prac02 _2_2.slx
Bloque:	Pulse Generator
Paleta:	Simulink -> Sources
Pulse type:	Time based
Time:	Use simulation time
Amplitude:	1
Period (sec):	1
Pulse Width (% of Period):	50
Phase delay (sec):	0
Bloque:	Pulse Generator1
Paleta:	Simulink -> Sources
Pulse type:	Time based
Time:	Use simulation time
Amplitude:	1
Period (sec):	0.25
Pulse Width (% of Period):	50
Phase delay (sec):	0
Bloque:	Pulse Generator2
Paleta:	Simulink -> Sources
Pulse type:	Time based
Time:	Use simulation time
Amplitude:	1
Period (sec):	1
Pulse Width (% of Period):	50
Phase delay (sec):	0
Bloque:	Analog Filter Design
Paleta:	DSP System Toolbox -> Filtering -> Filter Implementations
Design method:	Chebyshev II
Filter type:	Lowpass
Filter order:	8
Stopband edge frequency (rad/s):	$2\pi \cdot 20$
Stopband attenuation (db):	20
Bloque:	Analog Filter Design1 y Analog Filter Design2
Paleta:	DSP System Toolbox -> Filtering -> Filter Implementations
Design method:	Chebyshev II
Filter type:	Lowpass
Filter order:	8
Stopband edge frequency (rad/s):	$2\pi \cdot 80$
Stopband attenuation (db):	20
Bloque:	Matrix Concatenate, Matrix Concatenate1 y MatrixConcatenate 2
Paleta:	DSP System Toolbox -> Math Functions -> Matrices and Linear Algebra -> Matrix Operations
Number of inputs:	2
Mode:	Multidimensional array
Concatenate dimensión:	2
Bloque:	Zero-Order Hold, Zero-Order Hold1 y Zero-Order Hold2
Paleta:	Simulink -> Discrete
Sample time:	0.0025
Bloque:	Time Scope, Time Scope1 y Time Scope2
Paleta:	DSP System Toolbox -> Sinks
Number of inputs:	2



Estudio del aumento del ancho de banda del canal

Supongamos ahora que usamos otro canal de comunicaciones con un ancho de banda mayor, en este caso de 80 Hz, es decir, que solo deja pasar las componentes de frecuencia de la señal entre 0 y 80 Hz. Para ello usaremos los bloques de Simulink "Analog Filter Design1" y "Analog Filter Design2" usándose el primero para transmitir la señal digital Pulse "Generator1" y el segundo la señal digital "Pulse Generator2" (modelos inferiores de la Figura 22).

En ambos filtros se establece la frecuencia de corte en 80 Hz (o $2\pi 80$ rad/s) y el tipo de filtro será igual que el usado anteriormente, es decir, tipo "Lowpass" (pasa baja) y diseño "Chebyshev II" (orden 8).

Un cable de cobre que se comporte como un canal con mayor ancho de banda puede ser obtenido modificando las características físicas del mismo, como aumentando el blindaje del cable, el diámetro del cable de cobre o disminuyendo la distancia entre el emisor y receptor.

Preguntas

- a) Comprobar con los espectros de potencia y la representación en el tiempo como a mayor ancho de banda del canal se puede transmitir una señal a mayor velocidad pero conservando la calidad de la señal recibida.

Nota. Como en el ejercicio anterior podemos establecer que una calidad adecuada de la señal recibida es aquella que tiene 11 componentes de frecuencia.

- b) Comprobar con los espectros de potencia y la representación en el tiempo como a mayor ancho de banda del canal se puede transmitir una señal a una velocidad menor pero aumentando la calidad de la señal recibida.

Nota. Como en el ejercicio anterior podemos establecer que una calidad adecuada de la señal recibida es aquella que tiene 11 componentes de frecuencia.

- c) Rellenar la siguiente tabla con el resumen de los resultados:

	Modelo 1	Modelo 2	Modelo 3
Señal	Pulse Generator	Pulse Generator1	Pulse Generator2
Periodo (s)	1	0.25	1
Frecuencia fundamental (Hz)			
Número total de bits transmitidos			
Velocidad de transmisión serie (baudios)			
Velocidad de transmisión (bps)			
Ancho de banda absoluto (Hz)			
Ancho de banda relativo (Hz) considerando solo las 11 primeras componentes de frecuencia			
Ancho de banda del canal (Hz)			
Calidad de la señal recibida (Número de componentes de frecuencia)			



2.3 Atenuación de la señal recibida y ancho de banda de un canal

El segundo efecto de la transmisión por cualquier medio físico es la *atenuación*, eso quiere decir que la señal que se recibe está atenuada en las componentes de frecuencia que deja pasar el canal. Por encima de la frecuencia de corte, el canal no deja pasar componentes de frecuencia, como vimos en el apartado anterior.

El valor de la atenuación, en los medios guiados (cables) depende fundamentalmente de la distancia entre el emisor y el receptor, pudiendo ser de hasta 3 dB.

Nota: La relación entre la atenuación N medida en decibelios y la amplitud de la señal transmitida A_T y recibida A_R es $N_{dB} = 10 \log(A_T/A_R)$.

Para realizar el estudio de la atenuación vamos a considerar el modelo de Simulink de la Figura 23. En este caso vamos a transmitir una señal digital correspondiente a la siguiente secuencia de 16 bits: [0 1 0 1 0 0 0 1 1 1 0 1 1 0 0 0] y como codificación digital de esta secuencia consideraremos que un bit "1" se corresponde con 1.5 voltios y un bit "0" con 0 voltios. Esto lo podemos hacer con el bloque "Gain" de la figura.

La manera de generar la señal será como la usada en la práctica 1, es decir, el bloque "Pulse Generator" permite generar señales periódicas cuadradas de amplitud 0.5, periodo 1 segundo, fase 0 segundo y con un ancho del pulso del 50 % del periodo total. Actuará como el reloj de la tarjeta de red de un PC.

El bloque "Triggered Signal from Workspace" tiene como entrada los datos digitales que se quieren transmitir, es decir la secuencia de ceros y unos anterior y como salida la señal digital correspondiente con la duración adecuada de cada pulso.

Importante. En el bloque "Triggered Signal from Workspace", la secuencia de datos [0 1 0 1 0 0 0 1 1 1 0 1 1 0 0 0] que se quiere transmitir se configura de una manera particular: el primer bit (en este caso el 0) se configura en el parámetro de configuración "Initial output" y el resto de la secuencia (15 bits) se configuran en el parámetro de configuración "Signal".

Para simular el comportamiento del canal usaremos nuevamente un filtro analógico, bloque "Analog Filter Design", pero esta vez el diseño será "Chebyshev I", tipo de filtro "Lowpass", orden 8 con un nuevo parámetro que es "Passband ripple (db)", precisamente en este parámetro podemos especificar la atenuación en decibelios que queramos. Fijarla en **6 dB** (el doble de la atenuación que queramos $2 \times$ Atenuación, es como se define el filtro la atenuación).

Fijar inicialmente la frecuencia de corte del filtro pasabaja en 80 Hz (o $2\pi 80$ rad/s), es decir el ancho de banda del canal es de 80 Hz.

El modelo lo grabaremos en un fichero de nombre "**Prac02_2_3.slx**".

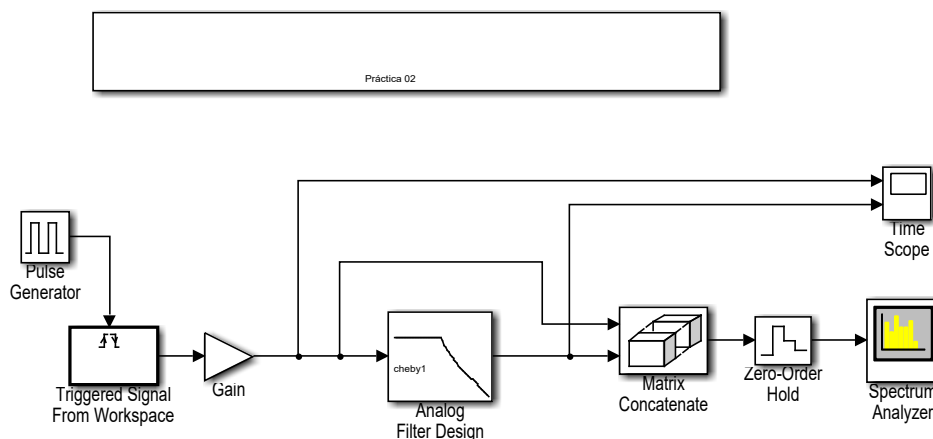


Figura 23. Modelo para estudiar el efecto del ancho de banda del canal y la atenuación de la señal recibida en la transmisión de señales digitales.



Tiempo máximo de la simulación: 8 segundos	
Nombre del modelo:	Prac02_2_3.slx
Bloque:	Pulse Generator
Paleta:	Simulink -> Sources
Pulse type:	Time based
Time:	Use simulation time
Amplitude:	0.5
Period (sec):	1
Pulse Width (% of Period):	50
Phase delay (sec):	0
Bloque:	Triggered Signal From Workspace
Paleta:	DSP System Toolbox -> Signal Operations
Signal:	[1 0 1 0 0 0 1 1 1 0 1 1 0 0 0]
Trigger type:	Either edge
Initial output:	0
Samples per frame:	1
Form output:	Setting zero
Bloque:	Gain
Paleta:	Simulink -> Math Operations
Gain:	1.5
Sample time:	-1
Bloque:	Analog Filter Design
Paleta:	DSP System Toolbox -> Filtering -> Filter Implementations
Design method:	Chebyshev I
Filter type:	Lowpass
Filter order:	8
Stopband edge frequency (rad/s):	$2\pi f$
	(*) La frecuencia de corte f la iremos cambiando
Passband ripple (db):	6
Bloque:	Time Scope
Paleta:	DSP System Toolbox -> Sinks
Number of inputs:	2
Bloque:	Matrix Concatenate
Paleta:	DSP System Toolbox -> Math Functions -> Matrices and Linear Algebra -> Matrix Operations
Number of inputs:	2
Mode:	Multidimensional array
Concatenate dimensión:	2
Bloque:	Zero-Order Hold
Paleta:	Simulink -> Discrete
Sample time:	0.0025
Bloque:	Spectrum Analyzer
Paleta:	DSP System Toolbox -> Sinks
Main Options:	Type:Power; Full Frequency Span; Window length: 3200
Trace Options:	Units: Watts
Parámetros:	Ver práctica 1

Preguntas

- Un bit "1" se transmiten como una señal de voltaje de 1.5 voltios. Calcular la amplitud de la señal digital que se recibe en el receptor para un bit "1" y que ha sido atenuada 3dB por el canal.
- Comprobar el resultado anterior gráficamente.
- Ir disminuyendo el ancho de banda del canal a 10 Hz, a 5 Hz y a 1 Hz. ¿Qué ocurre con la señal recibida en el receptor?
- Considerar la frecuencia de corte otra vez en 80 Hz, investigar las opciones del bloque "Spectrum Analyzer" y obtener el valor medio de la potencia de la señal transmitida en decibelios y calcular su equivalente en vatios. ¿Qué ocurre con la potencia de la señal que se recibe respecto de la potencia de la señal que se transmite?
- ¿Cómo se te ocurre aumentar la potencia de la señal recibida en el receptor? Dar los resultados de la potencia de la señal transmitida y recibida tanto en decibelios como en vatios.



3 Problemas capacidad del canal

- 1) Calcular la capacidad máxima de un **canal sin ruido**, de ancho de banda 3kHz para transmitir una señal digital codificada mediante dos valores (0 y 4.5 v).
- 2) Calcular la capacidad máxima de un **canal sin ruido**, de ancho de banda 3kHz para transmitir una señal digital codificada mediante 4 valores (0, 1.5, 3 y 4.5 v).
- 3) Dibujar el cronograma de la secuencia de datos 010111 para los dos casos anteriores, es decir, cuando se usa una canal sin ruido con ancho de banda de 3 kHz codificado mediante 2 valores o mediante 4 valores.
 - ¿Cuántos segundos dura la transmisión en cada caso?
 - ¿Cuántos símbolos diferentes se usan en cada caso?
 - ¿Cuál es el intervalo más corto de los símbolos en cada caso?
 - ¿Cuál es la velocidad de transmisión en baudios en cada caso?
- 4) Calcular la capacidad máxima de un **canal sin ruido**, de ancho de banda 6 kHz para transmitir una señal digital codificada mediante dos valores (señal binaria 0 y 4.5 V). Dibujar el cronograma de la secuencia de datos 010111 para la transmisión por este canal.
 - ¿Cuánto segundos dura la transmisión?
 - ¿Cuántos símbolos diferentes se usan?
 - ¿Cuál es el intervalo más corto de los símbolos?
 - ¿Cuál es la velocidad de transmisión en baudios?
- 5) Calcular la relación señal ruido (S/N) de una señal de 100 watos de potencia y de un ruido presente de 0.1 watos. Expresarla también en decibelios S/N_{dB} .
- 6) Calcular la capacidad máxima de un canal de ancho de banda 3 kHz **con ruido** que presenta una relación seña ruido de 40 dB.
- 7) Calcular la capacidad máxima de un canal de ancho de banda 1 MHz con ruido. Donde se sabe que la potencia de la señal en origen es de 100 watos, que la potencia de la señal ruido es de 0.01 watos y que la atenuación de las señal en función de la distancia está dada en la tabla siguiente:

km	Atenuación (dB)	Potencia señal origen Pso (w)	Potencia señal ruido Pr (w)	Potencia señal destino Psd (w)	S/N	S/N (dB)	Capacidad máxima del canal (bps)
0	0	100	0.01				
1	3	100	0.01				
2	6	100	0.01				



IV Bibliografía

Stallings. Comunicaciones y redes de computadoras, Pearson. 2004

A. Tanenbaum. Computer Networks, 5th edition, Pearson. 2011

G. Teodoro, J.E. Díaz Verdejo y J.M. López Soler. Transmisión de datos y redes de computadoras, Pearson, 2003





Grado en Ingeniería Informática

REDES

PRÁCTICA 3

Transmisión de datos. Codificación,
modulación digital y multiplexación

Docentes:

Alejandro Merino

Daniel Sarabia Ortiz

*Dpto. de Ingeniería Electromecánica
Área de Ingeniería de Sistemas y Automática*

Versión 2.0

Fecha 08/02/2022 19:21

*Esta obra está sujeta a la licencia Reconocimiento 4.0 Internacional de
Creative Commons. Para ver una copia de esta licencia, visite
<http://creativecommons.org/licenses/by/4.0/>*



Índice de contenidos

I	INTRODUCCIÓN	3
II	OBJETIVOS	3
III	CONTENIDOS ESPECÍFICOS DEL TEMA	3
1	Transmisión de datos	3
1.1	Codificación.....	4
1.2	Modulación digital.....	8
1.3	Multiplexación.....	13
2	Estudio codificación, modulación y multiplexación con Simulink	19
2.1	Codificación NRZ y Manchester	19
2.2	Modulación digital por desplazamiento de amplitud (ASK).....	22
2.3	Modulación digital por desplazamiento de frecuencia (FSK).....	25
2.4	Modulación digital por desplazamiento de fase (PSK).....	27
2.5	Multiplexación por división en frecuencia	29
IV	PROBLEMAS MULTIPLEXACIÓN	33
V	BIBLIOGRAFÍA	33



I Introducción

A diferencia de otras redes, como la de telefonía fija, en cualquier red de computadoras la información que se quiere transmitir entre sistemas terminales es de origen digital (secuencias de bits, es decir, unos y ceros consecutivos). Existen diferentes maneras de transmitir físicamente esta información digital, o bien usando señales digitales o bien usando señales analógicas como ya vimos en las prácticas anteriores. En el primer caso estaremos hablando de codificación y en el segundo de modulación digital.

En esta práctica estudiaremos en detalle distintas técnicas de codificación y modulación digital, junto con sus características, ventajas e inconvenientes.

Usaremos la toolbox de Simulink "**DSP System Toolbox**" y el estudio se realizará tanto en el dominio temporal como en el de la frecuencia obteniendo los correspondientes espectros de frecuencias.

Todos los alumnos de la Universidad de Burgos disponen de licencia académica para utilizar Matlab de forma legal y trabajaremos con la versión **R2019b de Matlab**.

II Objetivos

- Describir los dos mecanismos para transmitir información digital, usados en las redes de computadoras: codificación y modulación digital.
- Afianzar el concepto de transmisión en banda base asociado a la codificación y el concepto de transmisión pasa banda asociado a la modulación digital.
- Describir distintas técnicas de codificación. Ventajas e inconvenientes.
- Describir distintas técnicas de modulación digital. Ventajas e inconvenientes.
- Estudiar las características en la frecuencia de ambos mecanismos de transmisión de información digital y las diferencias fundamentales entre ellos.

III Contenidos específicos del tema

1 Transmisión de datos

La transmisión de datos (o transmisión de información) en un sistema de transmisión entre un transmisor y un receptor ocurre a través de un **medio físico de transmisión**, ver Figura 1, que se denomina **canal de comunicaciones**, o simplemente canal.

En este apartado nos vamos a centrar en la transmisión solo de datos digitales, puesto que son los que se manejan en cualquier red de computadoras. Estos datos pueden transformarse en señales digitales o en señales analógicas y después transmitir dichas señales.

La elección no será, casi nunca, una decisión del usuario, sino que vendrá determinada por el medio de transmisión a emplear.



Dos posibilidades para transmitir datos digitales:

- **Usando una señal digital.** Transmisión en banda base. Para obtener la secuencia que compone la señal digital a partir de los datos digitales se efectúa un proceso denominado *codificación* y los códigos que lo llevan a cabo se denominan *códigos en línea* (line codes).

Ejm. NRZ, NRZI, Manchester, AMI, etc.

- **Usando una señal analógica.** Transmisión pasabanda. Al proceso por el cual obtenemos una señal analógica a partir de unos datos digitales se le denomina *modulación digital*. El módem es el encargado de realizar dicho proceso.

Ejm. FSK, ASK, PSK, QAM, etc.

En ambos casos, ver Figura 1, el transmisor transmite la señal y el receptor debe realizar el proceso contrario para recuperar la información.

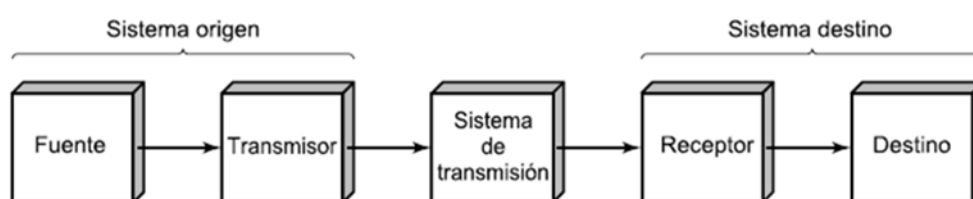


Figura 1. Modelo típico para las comunicaciones.

1.1 Codificación

Se considera transmisión en banda base cuando la información digital se transmite como señales digitales. En este tipo de transmisiones, como ya vimos en las prácticas anteriores, el canal por el que se transmita la señal digital debe tener un ancho de banda que comience en 0 Hz hasta una frecuencia f_1 suficiente para contemplar el ancho de banda relativo de la señal digital. Recordad que el ancho de banda absoluto de cualquier señal digital es infinito y el ancho de banda relativo es la banda de frecuencias donde se concentra la mayor parte de la energía de la señal.

Codificación NRZ (No Return to Zero)

Es el método que empleamos para representar la evolución de una señal digital en un cronograma. Cada bit ("0" o "1") se transmite (codifica) mediante un valor distinto de tensión. Por ejemplo, al bit "0" se le asigna 0 voltios y al bit "1" se le asigna 5 voltios, ver Figura 2, y esta señal digital es la que realmente se transmite por el canal.

Con esta codificación existen problemas importantes en la recuperación de la señal cuando hay muchos ceros y unos consecutivos. ¿Exactamente cuántos bits iguales consecutivos se han transmitido?, por ello se necesita usar una sincronización mediante relojes precisos en el emisor y en el receptor. Es decir que los relojes del emisor y del receptor estén perfectamente sincronizados.

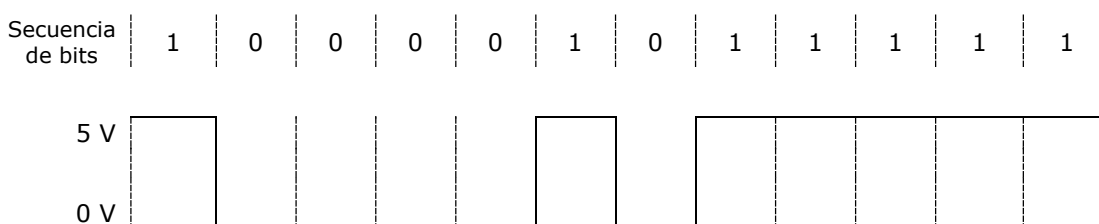


Figura 2. Codificación NRZ.



En NRZ es posible aumentar la velocidad de transmisión de datos usando más valores de tensión sin necesidad de aumentar el ancho de banda. Por ejemplo, con 4 valores de tensión (en vez de dos como antes) se pueden enviar 2 bits a la vez, ver Figura 3 y Figura 4. Dicho resultado era el esperado aplicando el teorema de Shannon para canales sin ruido que vimos en la práctica 2. Esto significa que se ha aumentado al doble la velocidad de transmisión y por tanto en la mitad de tiempo puedo enviar la misma información digital. El problema surge en el receptor, que antes debía distinguir solo entre dos niveles de tensión y ahora debe distinguir entre 4 niveles de tensión distintos.

Niveles de tensión	T	
	(bits/símbolo)	
4.5 V	1	1
3 V	1	0
1.5 V	0	1
0 V	0	0

Figura 3. Codificación con 2 bits por símbolo y por tanto con cuatro niveles de tensión.

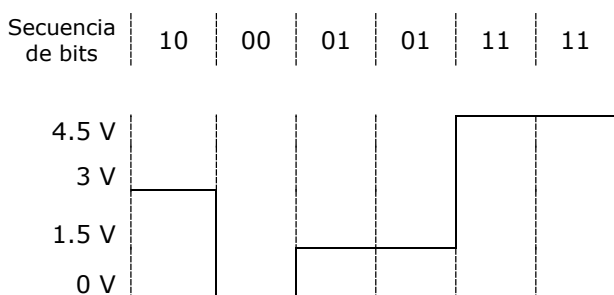


Figura 4. Codificación NRZ con cuatro niveles de tensión.

Codificación Manchester

En esta codificación se combina la señal del reloj del emisor con la secuencia de bits a transmitir mediante la operación lógica XOR, ver Figura 5. Cada bit no se representa como un nivel de tensión, sino como **transiciones o flancos** en mitad de la celda de bit.

Si la transición es positiva o flanco ascendente (de 0 a 5 voltios) implica que se ha transmitido un bit "0" y si la transición es negativa o flanco descendente (de 5 a 0 voltios) implica que se ha transmitido un bit "1".

De esta manera, la señal del reloj junto con la secuencia digital marca una transición.

En esta codificación, al existir frecuentes cambios en la secuencia de bits a transmitir, es muy fácil que el receptor se sincronice solo con los datos transmitidos.

En esta codificación, la señal que se transmite contiene información tanto de los bits a transmitir, es decir, la información digital que queríamos transmitir como información relacionada con el reloj del emisor. Esto implica que la señal que realmente se transmite tiene mucha más "actividad" pues contiene más información que la señal codificada mediante NRZ, comparar la Figura 6 con la Figura 2.

Este aumento en los cambios de los niveles de tensión implica que la señal que se transmite tiene un ancho de banda (relativo) mucho mayor que la correspondiente a la codificación NRZ y por tanto se necesita mayor ancho de banda del canal para transmitir correctamente una señal Manchester que una señal NRZ y esto a veces puede ser un problema.



Señal Reloj	Bit a transmitir	Bit transmitido (XOR)
0	0	0
0	1	1
1	0	1
1	1	0

Figura 5. Operación XOR entre la señal del reloj y cada bit a transmitir.

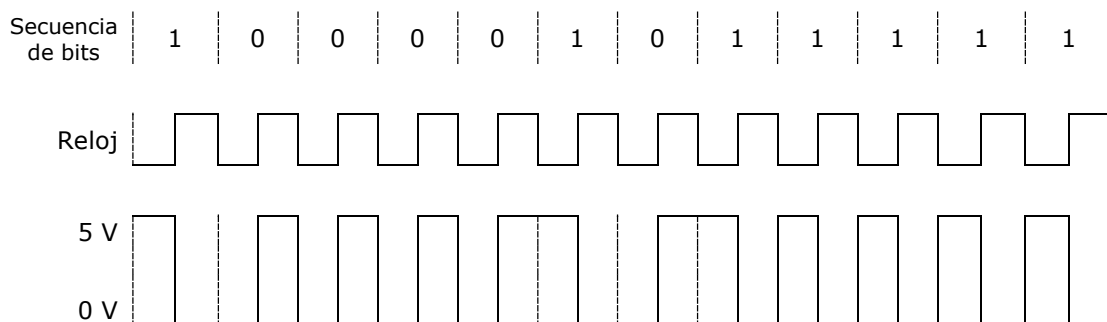


Figura 6. Codificación Manchester.

Codificación NRZI (Non Return to Zero Inverted)

Esta codificación incorpora indirectamente la señal del reloj en los bits transmitidos. Si se quiere transmitir un bit "0" se mantiene el nivel de la señal, si se quiere transmitir un bit "1" se cambia el nivel de la señal.

El bit "0" se representa como un nivel de tensión, pero, el bit "1" no se representa como un nivel de tensión, sino como **transiciones o flancos** (positivos o negativos dependiendo del caso) en mitad de la celda de bit. Esto significa que la conversión de la señal transmitida a secuencias de bits en el receptor será: si se detecta el mismo valor de tensión lo decodifica como un bit "0" y si se detecta un cambio de tensión, se decodifica como un bit "1".

En el caso de la Figura 7, se ha elegido transmitir en dos niveles de tensión (0 y 5 voltios). Si se quiere transmitir un bit "0" se mantiene el nivel de tensión que tuviera la señal (depende de momentos, en 0 voltios o en 5 voltios), si se quiere transmitir un bit "1" se cambia el nivel de tensión (si la señal estaba en 0 voltios se pasa a 5 voltios y si la señal estaba en 5 voltios se cambia a 0 voltios).

Necesita menos ancho de banda que la codificación Manchester con la ventaja de que muchos bits "1" consecutivos no producen problemas de sincronización, pues nunca se transmiten como un único valor de tensión, pero, muchos bits "0" consecutivos siguen produciendo problemas de sincronización, pues se transmiten con un valor de tensión, por ejemplo en la Figura 7, con 5 voltios en unos casos y en otros con 0 voltios.

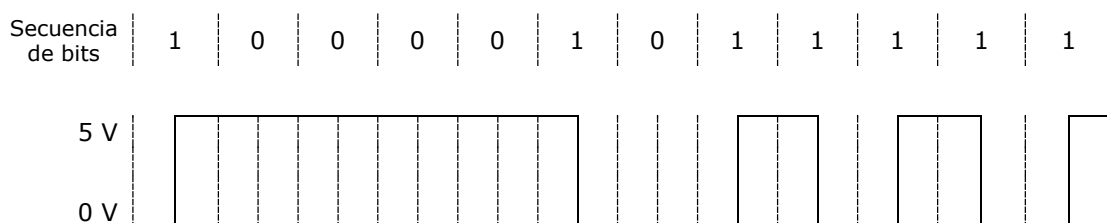


Figura 7. Codificación NRZI.



Para solucionar el problema de la transmisión de muchos bits "0" consecutivos se usa otra codificación combinada con la NRZI.

Esta nueva codificación se basa en agrupar bits consecutivos antes de ser codificados mediante NRZI en grupos más pequeños y asignar a cada grupo una secuencia de bits (codeword) con menos bits consecutivos iguales. El resultado de esta primera codificación es a su vez codificada con la codificación NRZI y el resultado final es la señal digital transmitida por el canal.

Por ejemplo, el código 4B/5B dado en la Figura 8, garantiza que nunca se van a transmitir más de 3 bits "0" consecutivos iguales. Para ello este código se basa en agrupar la secuencia original de bits en grupos de 4 bits (existen $2^4 = 16$ combinaciones o agrupaciones) y asignar a cada una de estas 16 agrupaciones una secuencia de 5 bits. Hay que fijarse que con 5 bits disponemos de $2^5 = 32$ combinaciones posibles y por tanto podemos seleccionar 16 de ellas con la idea de que nunca haya más de tres bits "0" consecutivos.

Data (4B)	Codeword (5B)	Data (4B)	Codeword (5B)
0000	11110	1000	10010
0001	01001	1001	10011
0010	10100	1010	10110
0011	10101	1011	10111
0100	01010	1100	11010
0101	01011	1101	11011
0110	01110	1110	11100
0111	01111	1111	11101

Figura 8. Código 4B/5B.

En la Figura 9, se ha codificado mediante el código 4B/5B la secuencia original de bits a transmitir [1 0 0 0 1 0 1 1 1 1 1], primeramente, realizando grupos de 4 bits [1 0 0 0], [0 1 0 1] y [1 1 1 1] y asignando a cada grupo las secuencias de 5 bits según el código 4B/5B de la Figura 8: [1 0 0 1 0], [0 1 0 1 1] y [1 1 1 0 1], en las que ya no aparecen más de tres bits "0" consecutivos. A continuación se codifica esta secuencia de bits mediante el código NRZI. Finalmente la señal que se transmite nunca implica más tres bits "0" consecutivos como valor de tensión 0 o 5 voltios y nunca se transmiten bits "1" consecutivos como un único valor de tensión.

Está claro que el inconveniente es que hay que realizar más operaciones matemáticas a los bits que se quieren transmitir y que por cada 4 bits originales se ha añadido un bit extra, por tanto, originalmente queríamos transmitir 12 bits y con esta combinación de codificaciones se transmiten 15 bits. Esto implica transmitir más información y por tanto más ancho de banda requiere nuestro canal de comunicaciones.

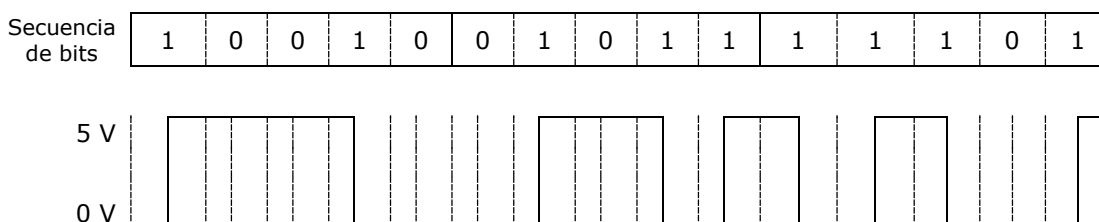


Figura 9. Codificación NRZI dónde previamente se ha realizado una codificación 4B/5B.

Codificación AMI (Alternate Mark Inversion)

La codificación AMI o bipolar es un código balanceado, es decir garantiza que la señal digital transmitida siempre tenga media cero. Todas las señales que presentan una media distinta de 0 a lo largo del tiempo tienen componentes importantes de frecuencia justo en 0 Hz, como vimos en las prácticas anteriores. Esto implica que si se quiere transmitir



directamente esa señal el medio físico por el que se trasmite debe dejar pasar frecuencias desde 0 Hz (incluido) hasta una frecuencia determinada. Esto es lo normal en transmisión en banda base usando niveles de tensión y transmitiendo los niveles de tensión en un cable de cobre. Sin embargo, muchos medios físicos de transmisión atenúan mucho más las señales que presentan media distinta de cero.

En la codificación bipolar se necesitan 3 niveles de voltaje para codificar los bits, normalmente un bit "1" se codifica mediante dos niveles de tensión +1 voltio o -1 voltio y un bit "0" se codifica mediante un nivel siempre de 0 voltios. La particularidad es que para transmitir un bit "1", el codificador alterna entre +1 V y -1 V, ver Figura 10.

De esta manera se consigue anular prácticamente la componente continua (el valor medio) de la señal eléctrica y además resuelve el problema de enviar largas secuencias de unos.

Sin embargo, no resuelve la cuestión de cómo evitar la pérdida de la señal de reloj cuando se envían largas secuencias de ceros, pero se puede combinar con códigos como el 4B/5B vistos antes.

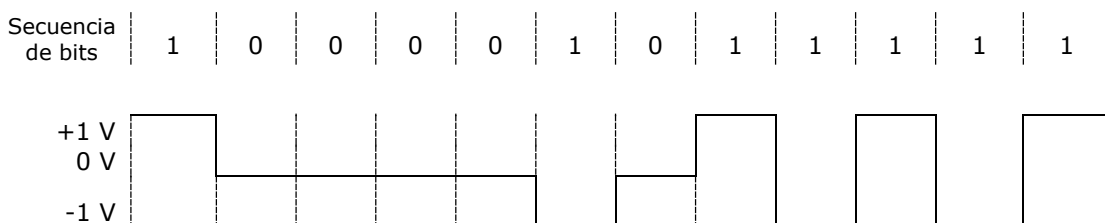


Figura 10. Codificación AMI.

1.2 Modulación digital

A menudo se quieren usar rangos de frecuencia que no empiezan en cero para transmitir información:

- Para evitar interferencias con otras señales.
- Para poder transmitir varias señales usando el mismo medio físico. Cada una en una banda de frecuencias distintas.
- En comunicaciones inalámbricas, transmitir a bajas frecuencias implica tamaños de antena enormes. No es útil para móviles, portátiles, etc.

Se denomina transmisión pasa banda cuando se elige una banda de frecuencias arbitraria para pasar una señal.

La capacidad del canal depende solo del ancho de la banda NO de la frecuencia absoluta.

El emisor puede desplazar una señal banda base que ocupa entre 0 y B Hz a una señal pasa banda que ocupe una banda entre S-B y S+B Hz sin que se cambie la información que transporta. El receptor volverá a desplazar la señal pasabanda a banda base para reconstruir mejor la señal.

La modulación digital consiste en modular una señal portadora pasabanda, es decir, cambiar su amplitud, frecuencia, fase o una combinación de ellas para representar la información digital que se quiera transmitir.

Modulación por desplazamiento de amplitud (ASK)

En la modulación ASK (Amplitude Shift Keying) se modifica la **amplitud** de la señal portadora en función del bit que se quiera transmitir: Usamos una señal (seno o coseno) de frecuencia f y fase ϕ constantes, pero si queremos transmitir un bit "1" usamos una



amplitud A_1 y si queremos transmitir un bit "0" usamos otra amplitud distinta A_0 , ver Figura 11

Este es el método utilizado en la fibra óptica, cuando se quiere transmitir un bit "1" se envía un pulso de luz (señal electromagnética de frecuencia entre 10^{14} y 10^{15} Hz) y una cierta amplitud A_1 y cuando se quiere transmitir un bit "0" no se transmite nada, es decir amplitud $A_0 = 0$.

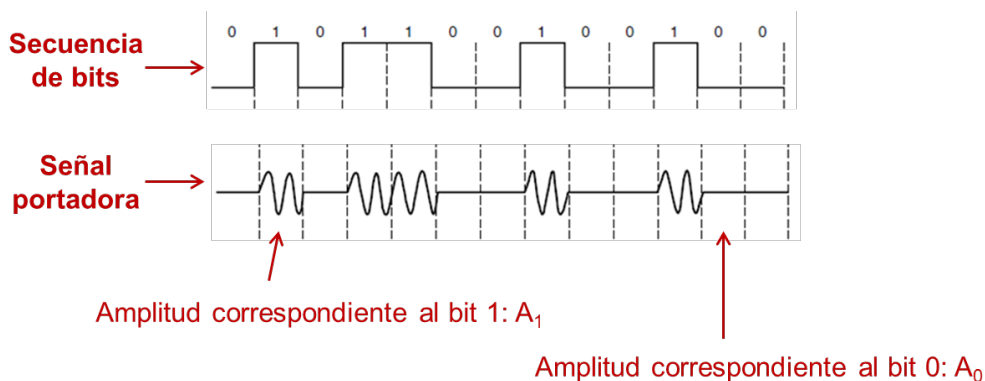


Figura 11. Modulación digital ASK.

Con este método, usar señales electromagnéticas de mayor frecuencia hace que la velocidad de transmisión se aumente considerablemente. Por ejemplo, si usamos una onda de radio de frecuencia 10^6 Hz, su periodo es de 10^{-6} s, es decir, podemos enviar un símbolo diferente (una amplitud) cada 10^{-6} segundos, en este caso un bit cada 10^{-6} segundos. Sin embargo, si usamos fibra óptica, es decir luz cuya frecuencia es de 10^{15} Hz, su periodo es de 10^{-15} s y podemos transmitir un bit cada 10^{-15} segundos. Es decir con la fibra óptica hemos aumentado la velocidad de transmisión en 10^9 , imil millones de veces!

Es importante recalcar que cuando se usa esta modulación ASK, el ancho de banda relativo de la señal analógica es exactamente el doble que el ancho de banda relativo de la señal equivalente digital equivalente si se usase la codificación NRZ para la secuencia de bits.

- **Ancho de banda relativo de la señal digital**, usando la codificación NRZ de la secuencia de bits: B (Hz). Es decir el ancho de banda relativo ocupa las frecuencias entre 0 Hz y B Hz. Señal banda base.
- **Ancho de banda relativo de la señal analógica**, usando modulación digital ASK: $2B$ (Hz). Ancho de banda centrado en la frecuencia f de la señal portadora, es decir, el ancho de banda relativo ocupa las frecuencias entre $f - B$ Hz y $f + B$ Hz. Señal Pasa banda.

Modulación por desplazamiento de frecuencia (FSK)

En la modulación FSK (Frequency Shift Keying) se modifica la **frecuencia** de la señal portadora en función del bit que se quiera transmitir: Usamos una señal (seno o coseno) de amplitud A y fase ϕ constantes, pero si queremos transmitir un bit "1" usamos una frecuencia f_1 y si queremos transmitir un bit "0" usamos otra frecuencia distinta f_0 , ver Figura 12.

Este era el método utilizado en los primeros módems de acceso telefónico.



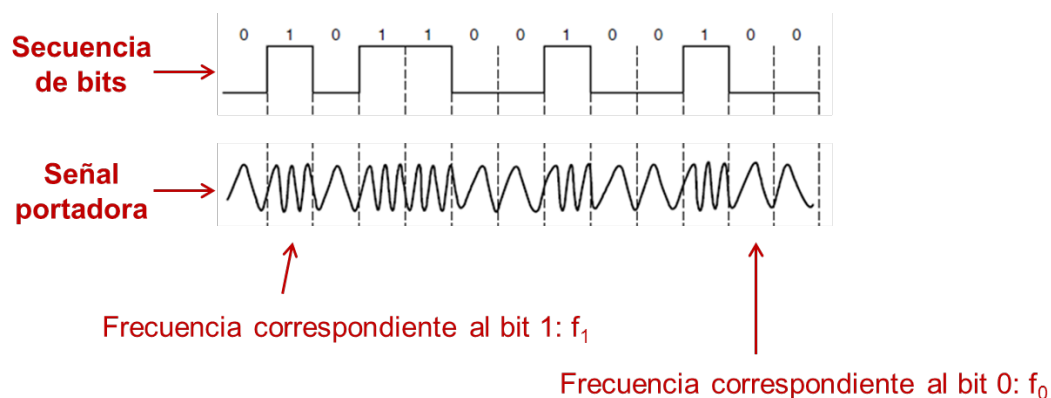


Figura 12. Modulación digital FSK.

Modulación por desplazamiento de fase (PSK)

En la modulación PSK (Phase Shift Keying) se modifica la **fase** de la señal portadora en función del bit que se quiera transmitir: Usamos una señal (seno o coseno) de amplitud A y frecuencia f constantes, pero si queremos transmitir un bit "1" usamos una fase ϕ_1 y si queremos transmitir un bit "0" usamos otra fase distinta ϕ_0 , ver Figura 13.

Este era el método utilizado en los primeros módems de acceso telefónico y actualmente en las redes LAN inalámbrica (IEEE 802.11), redes LAN inalámbrica de alta velocidad (IEEE 802.11g-2003) y en el cable coaxial para el tráfico de subida en redes de acceso mediante cable TV.

Es habitual representar esta modulación digital en un **diagrama de constelación**, Figura 14 a), que representa la fase y amplitud con la que se convierte un bit (o secuencia de bits). En el caso a) al bit "1" se le asocia una fase de 0 rad (0°) y al bit "0" se le asocia una fase de $\pi \text{ rad}$ (180°), es decir, la fase opuesta y ambos con amplitud constante A .

- El número máximo de símbolos distintos es $n = 2$ (dos fases distintas).
- Cada símbolo transmite un único bit.
- El número de símbolos que se transmiten en el intervalo más corto es 1 (una de las dos fases).
- El intervalo más corto de señalización es el inverso de la frecuencia f , es decir el periodo T , esto se cumple siempre en señales analógicas.

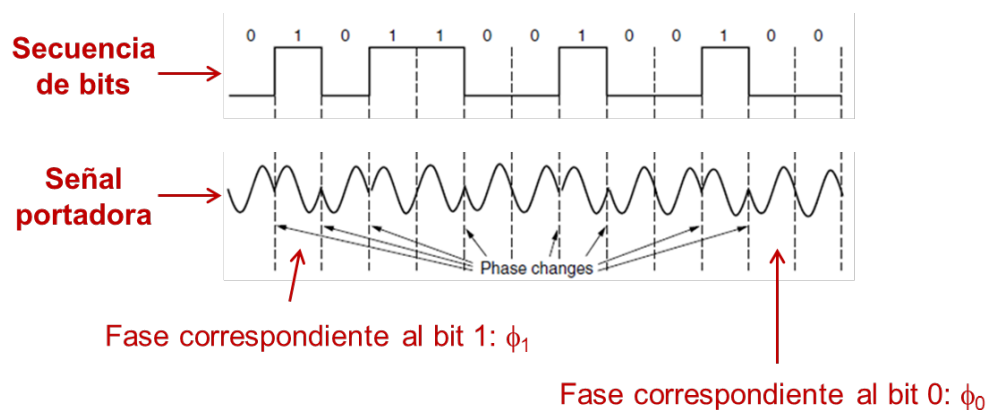


Figura 13. Modulación digital PSK.

Basándose en esta modulación existe una manera más eficiente de transmitir información digital, es decir aumentar la velocidad de transmisión sin cambiar el ancho de banda del canal, y es usar cuatro **fases** distintas para codificar con cada una de ellas 2 bits, a esta técnica se la denomina **QPSK (Quadrature Phase Shift Keying)** y su



diagrama de constelación se muestra en la Figura 14 b). A la secuencia de bits "00" se le asigna una señal de fase 0 rad (0°), a "01" una fase de $\pi/2 \text{ rad}$ (90°), a "10" una fase de $\pi \text{ rad}$ (180°) y a "11" una fase de $3\pi/2 \text{ rad}$ (270°). Todas las señales con la misma amplitud A y frecuencia f . De esta manera la velocidad de transmisión aumenta al doble *sin cambiar el tipo de onda electromagnética que se usa*, puesto que la frecuencia (que es lo que diferencia unas ondas electromagnéticas de otras) no cambia.

- El número máximo de símbolos distintos es $n = 4$ (cuatro fases distintas).
- Cada símbolo transmite $n_{\text{bits}} = 2 \text{ bits}$ ($2^{n_{\text{bits}}} = n$, por tanto $2^{n_{\text{bits}}} = 4$).
- El número de símbolos que se transmiten en el intervalo más corto es 1 (una de las cuatro fases).
- El intervalo más corto de señalización es el inverso de la frecuencia f , es decir el periodo T , esto se cumple siempre en señales analógicas.

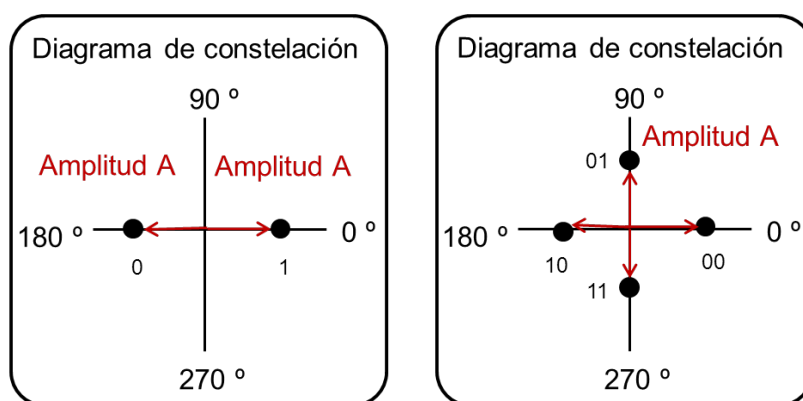


Figura 14. Diagramas de constelación para modulación digital a) PSK y b) QPSK.

Modulación de amplitud en cuadratura (QAM)

En la modulación digital QAM (Quadrature Amplitude Modulation) se modifica la **fase** y la **amplitud** la señal portadora en función de la secuencia de bits que se quiera transmitir. Usamos una señal (seno o coseno) de frecuencia f constante y depende de la modulación, Figura 15:

- **QAM-16.** Permite enviar 4 bits por símbolo usando 4 amplitudes y 12 fases distintas.
- **QAM-64.** Permite enviar 6 bits por símbolo usando 16 amplitudes y 48 fases distintas.

Actualmente se usa en modems ADSL y en las redes de cable coaxial para el tráfico de bajada. En concreto para QAM-16 podemos definir (ver Figura 16):

- El número máximo de símbolos distintos es $n = 16$ (combinación de 4 amplitudes distintas y 12 fases distintas).
- Cada símbolo transmite $n_{\text{bits}} = 4 \text{ bits}$ ($2^{n_{\text{bits}}} = n$, por tanto $2^{n_{\text{bits}}} = 16$).
- El número de símbolos que se transmiten en el intervalo más corto es uno (una combinación de una las cuatro amplitudes y una de las 12 fases).
- El intervalo más corto de señalización es el inverso de la frecuencia f , es decir el periodo T , esto se cumple siempre en señales analógicas.



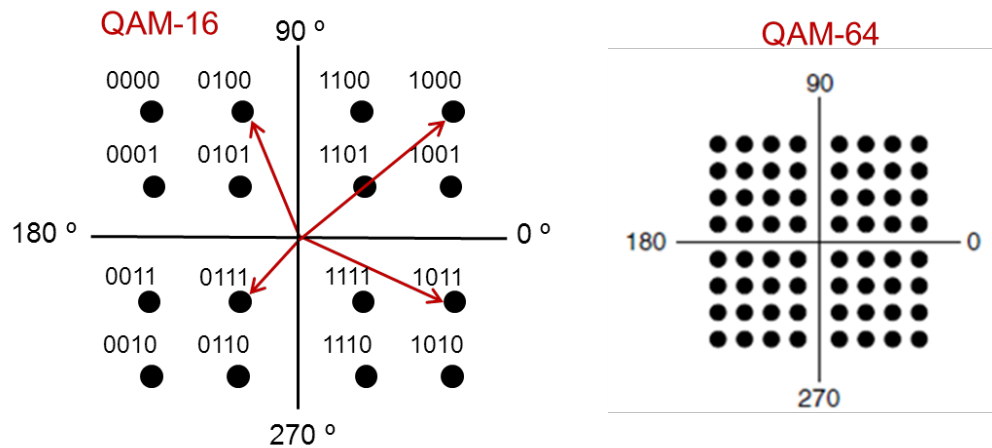


Figura 15. Diagramas de constelación para modulación digital a) QAM-16 y b) QAM-64.

Símbolo	Combinación fase y amplitud	Bits transmitidos
S_1	fase ₁ y A ₁	1001
S_2	fase ₂ y A ₂	1101
S_3	fase ₂ y A ₃	1000
S_4	fase ₃ y A ₄	1100
S_5	fase ₄ y A ₄	0100
S_6	fase ₅ y A ₂	0101
S_7	fase ₅ y A ₃	0000
S_8	fase ₆ y A ₁	0001
S_9	fase ₇ y A ₁	0011
S_{10}	fase ₈ y A ₂	0111
S_{11}	fase ₈ y A ₃	0010
S_{12}	fase ₉ y A ₄	0110
S_{13}	fase ₁₀ y A ₄	1110
S_{14}	fase ₁₁ y A ₂	1111
S_{15}	fase ₁₁ y A ₃	1010
S_{16}	fase ₁₂ y A ₁	0111

Figura 16. Modulación QAM-16. Número máximo de símbolos usados, definición de cada símbolo como combinación de una fase y una amplitud y bits transmitidos (asociados) a cada símbolo.

Para QAM-64 podemos:

- El número máximo de símbolos distintos es $n = 64$ (combinación de 16 amplitudes distintas y 48 fases distintas).
- Cada símbolo transmite $n_{\text{bits}} = 6$ bits ($2^{n_{\text{bits}}} = n$, por tanto $2^{n_{\text{bits}}} = 64$).
- El número de símbolos que se transmiten en el intervalo más corto es uno (una combinación de una de las 16 amplitudes y una de las 48 fases).
- El intervalo más corto de señalización es el inverso de la frecuencia f , es decir el periodo T , esto se cumple siempre en señales analógicas.

MODEM (MODULATOR-DEMODULATOR)

Dispositivo que convierte las señales digitales en analógicas (modulación) y viceversa (demodulación), permitiendo la comunicación entre computadoras a través de la línea telefónica convencional. Este aparato sirve para enviar la señal moduladora mediante otra señal llamada portadora. El propósito de la modulación es sobreponer señales en las ondas portadoras.



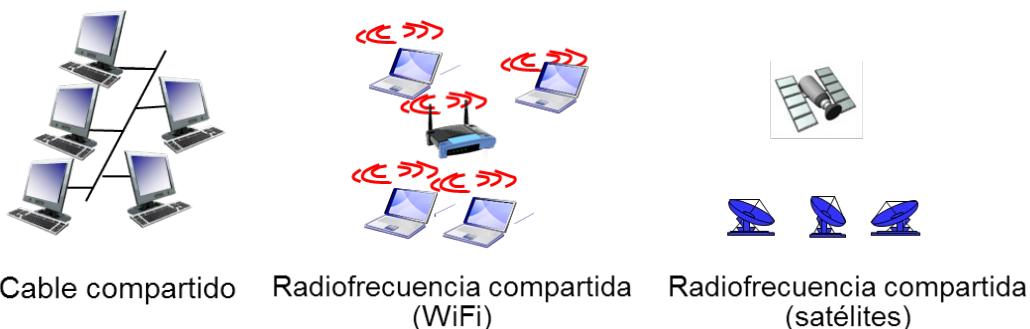
La señal moduladora constituye la información que se quiere transmitir y modifica alguna característica de la señal portadora de manera que se obtiene una señal, que incluye la información de la moduladora. Así el demodulador puede recuperar la señal moduladora original, es decir, quitando la portadora.

Se usa desde los años 60 para conectar computadoras y ha ido evolucionando (el hardware y el software) para conseguir las altas velocidades de transferencia actuales. Inicialmente se implementaba en los MODEM técnicas de modulación digital FSK y actualmente los MODEM ADSL implementan técnicas de modulación digital QAM64.

1.3 Multiplexación

La codificación y modulación digital permiten enviar una señal (digital y analógica respectivamente) que transporta información en bits. Sin embargo, en este apartado vamos a estudiar la **multiplexación** que permite usar varias señales a la vez sobre un mismo medio físico, es decir, permite compartir el medio físico entre varios dispositivos que se conectan a él, ver Figura 1, garantizando que cada dispositivo pueda enviar/recibir la información destinada a él sin interferir con el resto.

En concreto estudiaremos las técnicas de multiplexación siguientes, que se incluyen dentro de la categoría de *protocolos de particionamiento del canal*, pues se basan en dividir de alguna manera el ancho de banda del canal entre los distintos usuarios (nodos) interconectados: Multiplexación por división en frecuencia (FDM), multiplexación por división en tiempo (TDM), multiplexación por división en código (CDM) y multiplexación por división espacial (SDM).



Cable compartido Radiofrecuencia compartida (WiFi) Radiofrecuencia compartida (satélites)

Figura 17. Compartición de medio físico entre varios dispositivos.

Normalmente se aplica el nombre "multiplexado" para los casos en que un sólo dispositivo determina el reparto del canal entre distintas comunicaciones. Para los terminales de los usuarios finales, el multiplexado es transparente.

Sin embargo se emplea el término "control de acceso al medio" cuando son los terminales de los usuarios, en comunicación con un dispositivo que hace de nodo de red, los que deben usar un cierto esquema de comunicación para evitar interferencias entre ellos.

Una analogía posible para el problema del acceso múltiple sería una habitación (que representaría el canal) en la que varias personas desean hablar al mismo tiempo. Si varias personas hablan a la vez, se producirán interferencias y se hará difícil la comunicación.

Para evitar o reducir el problema, podemos seguir las siguientes estrategias:

- Hablar unos en tonos más agudos y otros en más graves de forma que sus voces se distingan. División por frecuencia.
- Hablar por turnos. División por tiempo.
- Hablar en idiomas distintos. Solo las personas que hablan el mismo idioma pueden entenderse, para el resto es "ruido". División por código.



- Dirigir sus voces en distintas direcciones de la habitación. División espacial.

Multiplexación por división en frecuencia

La multiplexación por división en frecuencias o FDM (Frequency-division multiplexing) consiste en dividir el espectro en bandas de frecuencia (o canales) y la señal de cada emisor se transmite en una banda concreta. Cada usuario está en posesión de su banda concreta. Entre cada banda es necesario añadir un "banda de guarda" que permite mantener separados los canales, a pesar de ello, existen siempre pequeños solapamientos entre componentes de señales, ver Figura 2.

El efecto en el dominio del tiempo de transmitir varias señales a la vez, pero en frecuencias distintas, es que se genera una señal aditiva de todas las individuales que se transmiten. Por tanto, el receptor solo tiene que utilizar tantos filtros pasabanda (solo dejan pasar la señal entre una frecuencia mínima y una frecuencia máxima) como bandas se haya dividido el espectro y de esa manera recuperar la señal de cada emisor.

En esta multiplexación, cada emisor solo dispone de un ancho de banda más pequeño que el ancho de banda total del medio de transmisión, pero lo puede utilizar todo el tiempo. Es una tecnología muy experimentada, fácil de implementar y se utiliza con modulaciones digitales, DSL, cable coaxial y en redes inalámbricas de área extensa (2G).

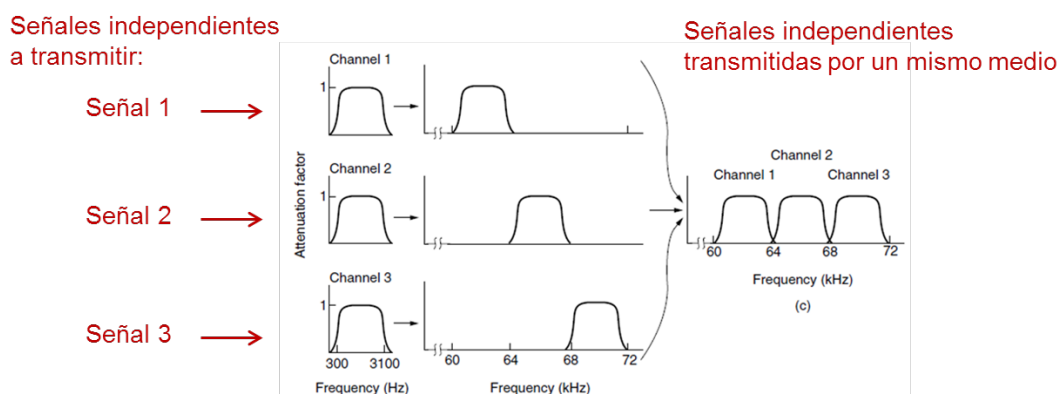


Figura 18. Multiplexación por división en frecuencia (FDM), para tres emisores, transmitiendo uno la señal 1, el segundo la señal 2 y el tercero la señal 3.

Multiplexación por división de longitud de onda

La multiplexación por división de longitud de onda o WDM (Wavelength Division Multiplexing) se basa en multiplexar varias señales sobre una sola fibra óptica mediante señales portadoras ópticas de diferente longitud de onda, usando luz procedente de un láser o de un LED.

Se refiere a una portadora óptica (descrita por su longitud de onda) mientras que la multiplexación por división de frecuencia generalmente se emplea para referirse a una portadora de radiofrecuencia (descrita por su frecuencia).

La longitud de onda λ (m) y la frecuencia f (Hz) son inversamente proporcionales, relacionadas a través de la velocidad de la onda v (m/s) (1).

$$\lambda = \frac{v}{f} \quad (1)$$

Recordad que la radiofrecuencia y la luz son formas de radiación electromagnética y que **todas** las ondas electromagnéticas se desplazan a la velocidad de la luz. Dicha velocidad es máxima en el vacío ($c = 300000 \text{ km/s}$), pero es inferior en otros medios, como puede ser el aire, el agua u otro material sólido.

La WDM puede ser de dos tipos:



- **Densa** (DWDM, "Dense" WDM): Se basa en dividir el espectro en muchos canales, es decir, en usar muchas longitudes de onda, y se usa para transmitir a largas distancias. Usa componentes más costosos y complejos, como láseres para generar la luz, pero a la vez este tipo de dispositivos permiten transmitir a más distancia y dividir el espectro de frecuencias en canales con anchos de banda mucho más pequeños (poder generar señales de luz de frecuencias muy juntas).
- **Ligera** (CWDM "Coarse" WDM): Se basa en dividir el espectro en pocos canales, es decir, en usar pocas longitudes de onda y se usa para transmitir a cortas distancias (entornos metropolitanos). El uso de dispositivos más baratos y sencillos, como leds para generar la luz, impide el transmitir a mayores distancias y dividir el espectro en muchos canales como en DWDM.

Se usa en las comunicaciones mediante fibra óptica. Los primeros sistemas WDM aparecieron en torno a 1985 y combinaban tan sólo dos señales, actualmente la tecnología DWDM permite transmitir hasta 160 canales a la vez, es decir, 160 señales ópticas independientes.

Multiplexación por división de tiempo

La multiplexación por división en tiempo o TDM (Time Division Multiplexing) es el tipo de multiplexación más utilizado en la actualidad, especialmente en los sistemas de transmisión digitales. En ella, el ancho de banda total del medio de transmisión es asignado a cada canal durante una fracción fija del tiempo total (intervalo de tiempo), es decir, cada emisor usa todo el ancho de banda del medio de transmisión pero solo en ciertos intervalos de tiempo, ver Figura 2.

Entre intervalos de tiempo es necesario añadir un "tiempo de guarda" que permite mantener los intervalos de tiempo separados y evitar solapamientos de las señales. Además:

- Requiere de algún algoritmo de "asignación de turnos".
- Requiere una sincronización estricta entre emisor y receptor.
- Tecnología simple y muy probada e implementada.
- Se utiliza con modulaciones digitales.
- Se utiliza en la conmutación de paquetes.

El receptor para recuperar la señal de cada emisor debe saber quién está transmitiendo durante cada intervalo de tiempo y después "juntar" las señales de todos los intervalos correspondientes a cada emisor. Para ello es necesaria una sincronización estricta entre los emisores y el receptor.

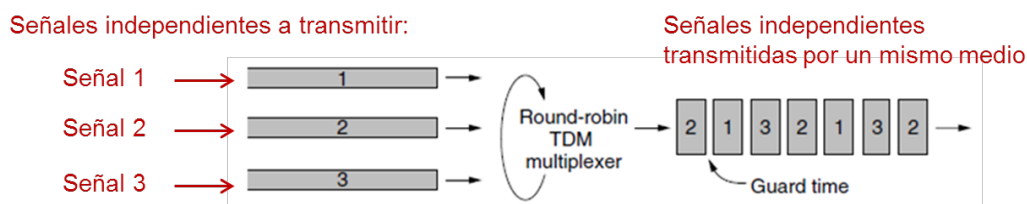


Figura 19. Multiplexación por división de tiempo (TDM), cada emisor usa todo el ancho de banda del medio de transmisión pero solo durante su intervalo de tiempo asignado.

Multiplexación por división de código

La multiplexación por división en código o CDM (Code Division Multiplexing) permite que varias señales de distintos usuarios compartan la misma banda de frecuencias todo el tiempo.

Se emplea una tecnología de espectro expandido y un esquema especial de codificación, por el que a cada transmisor se le asigna un código único, escogido de forma que sea



ortogonal respecto al del resto; el receptor capta las señales emitidas por todos los transmisores al mismo tiempo, pero gracias al esquema de codificación puede seleccionar la señal de interés si conoce el código empleado a pesar que todas las señales compartan la misma frecuencia y se envíen y reciban al mismo tiempo.

No es necesario emplear circuitería de filtrado en frecuencia (como en FDM), ni de conmutación de acuerdo con algún esquema temporal (como en TDM) para aislar la señal de interés. En CDM se reciben las señales de todos los usuarios al mismo tiempo y se separan mediante procesado digital.

Se usa en sistemas de comunicación por radiofrecuencia, tanto de telefonía móvil (3G, UMTS), transmisión de datos (WiFi), cable o navegación por satélite (GPS).

A cada estación o usuario se le asigna un único código de m -bits llamado secuencia de chips que es diferente para cada usuario, normalmente este código especial está formado por 64 o 128 bits. Por tanto para transmitir un bit de información cada emisor usará su código particular.

Para explicar el mecanismo de la multiplexación por división en código supongamos que varios emisores comparten el mismo canal y que todos ellos transmiten en banda base, es decir que usan una codificación especial asignando niveles de tensión a cada bit de información que quieran transmitir. El efecto en el tiempo de todas las señales digitales transmitidas a la vez es que la señal que recibe el receptor es la suma de todas las señales transmitidas por cada emisor (sumar en cada instante de tiempo todos los niveles de voltaje usados por todos los emisores).

En concreto, usaremos una secuencia con $m = 8$ bits y una codificación bipolar, es decir cada estación transmite una señal de tensión de -1 voltio, 0 voltios o 1 voltio.

Supongamos que a la estación A se le asigna la secuencia $A=(-1-1-1+1+1-1+1+1)$:

- Para transmitir el bit "1", A enviará su código asignado, $A=(-1-1-1+1+1-1+1+1)$.
- Para transmitir el bit "0", A enviará el negado de su código asignado, $\bar{A}=(+1+1+1-1-1+1-1-1)$.
- Cuando la estación A no quiera transmitir nada enviará 0 voltios.

Lo primero que hay que decidir cuando haya más de un usuario que quiere transmitir a la vez su señal digital, es qué código (secuencia de chips) se le asigna a cada uno.

La idea es garantizar que todos los códigos o secuencias de chips sean ortogonales entre sí, es decir que el producto interior de dos secuencias sea 0. Sea la secuencia A y la secuencia B ambas de m bits, estas dos secuencias son ortogonales entre sí, si se cumple (Código Walsh):

$$A \circ B = \frac{1}{m} \sum_{i=1}^m A_i B_i \quad (2)$$

Donde los productos del sumatorio se realizan elemento a elemento, es decir, elemento 1 de A por elemento 1 de B, elemento 2 de A por elemento 2 de B y así sucesivamente. Si dos secuencias son ortogonales entre sí, se cumplen las siguientes propiedades respecto del producto interior definido en (2):

$$\begin{aligned} A \circ \bar{B} &= 0 \\ A \circ A &= 1 \\ A \circ \bar{A} &= -1 \end{aligned} \quad (3)$$

Transmisión de varias señales a la vez



Supongamos que existen cuatro usuarios A, B, C y D que quieren transmitir a la vez y se escoge una multiplexación basada en código. A cada usuario se le asigna una secuencia A, B, C y D ortogonales entre sí:

$$\begin{aligned} A &= (-1 \ -1 \ -1 \ +1 \ +1 \ -1 \ +1 \ +1) \\ B &= (-1 \ -1 \ +1 \ -1 \ +1 \ +1 \ +1 \ -1) \\ C &= (-1 \ +1 \ -1 \ +1 \ +1 \ +1 \ -1 \ -1) \\ D &= (-1 \ +1 \ -1 \ -1 \ -1 \ -1 \ +1 \ -1) \end{aligned} \quad (4)$$

Durante la transmisión cada usuario puede:

- Transmitir un bit de información "1" mediante su código.
- Transmitir un bit de información "0" mediante el negado de su código.
- No transmitir nada (estar en silencio).

Hay que fijarse que al final los usuarios transmiten niveles de voltaje, -1 v, +1 v o 0 v.

Por ejemplo queremos que el usuario B y el C envíen un bit de información "1" a la vez, eso significa que cada uno enviará a la vez su código (mediante niveles de voltaje) y al hacerlo a la vez, el receptor recibirá la suma de ambos niveles de voltaje (la suma S_2 de las secuencias B y C):

$$S_2 = B+C = (-2 \ 0 \ 0 \ 0 \ +2 \ +2 \ 0 \ -2)$$

← Señal transmitida

↑ Información a transmitir por C (bit=1)

↑ Información a transmitir por B (bit=1)

Figura 20. Transmisión a la vez de un bit de información "1" del emisor B y C. Señal de voltaje realmente transmitida y recibida por el receptor (puede ser -2, -1, 0, 1 o 2 voltios).

Decodificación en el receptor de las señales transmitidas a la vez

El receptor debe conocer el código asignado a cada usuario y el receptor realiza el producto interior de la señal recibida con cada uno de los códigos de cada usuario, el resultado puede ser:

- **+1.** Un usuario concreto transmitió un bit de información "1".
- **-1.** Un usuario concreto transmitió un bit de información "0".
- **0.** Un usuario concreto no transmitió nada.

Siguiendo con el ejemplo anterior, la operación en el receptor para el mensaje recibido S_2 consistirá, ver Figura 21, en realizar los cuatro productos interiores, dando como resultado que A y D no enviaron nada, pues se obtiene un 0 y que B y C enviaron un bit de información "1" los dos, tal y como había ocurrido en realidad.

Emisores	Señal transmitida	Receptor. Decodificación de la Señal transmitida
$S_2 = B+C = (-2 \ 0 \ 0 \ 0 \ +2 \ +2 \ 0 \ -2)$ <p style="text-align: center;">↑ Información a transmitir por C (bit=1)</p> <p style="text-align: center;">↑ Información a transmitir por B (bit=1)</p>		$S_2 \cdot A = [2+0+0+0+2-2-2]/8 = 0$ $S_2 \cdot B = [2+0+0+0+2+2+0+2]/8 = 1$ $S_2 \cdot C = [2+0+0+0+2+2+0+2]/8 = 1$ $S_2 \cdot D = 0$

Figura 21. Decodificación del mensaje S_2 recibido en el receptor.



Resumen CDM

Si solo existe un emisor y este quiere enviar b bits de información digital, el hecho de usar CDM implica que cada bit de información digital se transmita mediante la secuencia de m bits asignada, por tanto, el número total de bits necesarios a enviar se aumenta a $m \cdot b$ bits. Si el usuario quisiera enviar la misma información digital en el mismo intervalo de tiempo, con CDM debería aumentar la velocidad de transmisión a $m \cdot b$ bits/s y por tanto necesitaría más ancho de banda. Así que el ancho de banda necesario se aumenta en m respecto a no usar CDM.

Por ejemplo, si el medio es compartido (un ancho de banda de 1 MHz) por 100 usuarios, pudiendo enviar cada uno 1 bit/Hz:

Usando FDM, cada usuario dispondría de un ancho de banda individual de 10 KHz y por tanto podría enviar 10 kbits de información digital.

Usando CDM, cada usuario dispone del total del ancho de banda, 1 MHz, es decir puede transmitir hasta 1 Mbits, pero cada uno tiene asignado una secuencia de 100 bits para transmitir un bit de información digital por tanto, en la práctica transmite solo 10 kbits de información digital.

En principio la multiplexación por división en código:

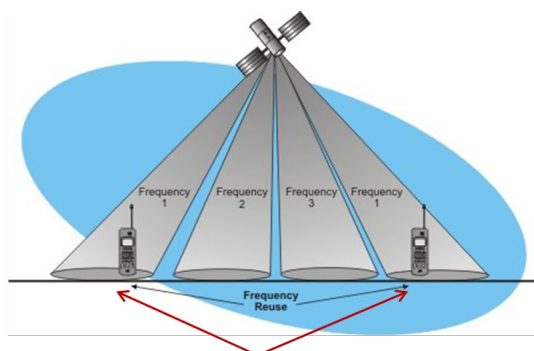
- Disponiendo de suficiente capacidad de cálculo, el receptor es capaz de escuchar a todos los emisores a la vez y decodificar sus secuencias en paralelo.
- Para canales no ruidosos, el número de emisores que pueden enviar a la vez información puede aumentar ilimitadamente sin más que aumentar la longitud de la secuencia de bits o secuencia de chips.
- Para 2^n emisores, el código Walsh puede obtener 2^n secuencias ortogonales.
- Necesidad de una sincronización perfecta entre emisores y receptores.

Multiplexación por división de espacio

La multiplexación por división espacial o SDM (Space Division Multiplexing) es una tecnología que segmenta el espacio en sectores utilizando antenas unidireccionales. Para ello es necesario localizar espacialmente a cada usuario. Se utiliza generalmente en comunicaciones por satélite y sólo es útil en combinación con FDM, TDM o CDM.

Esto significa que cada usuario puede usar todo el ancho de banda del medio de transmisión y durante todo el tiempo pues las señales transmitidas no interfieren entre ellas al estar orientadas a zonas del espacio distintas.

La futura tecnología de redes móviles 5G se basa en esta multiplexación.



Cada señal transmitida se dirige a una posición espacial determinada, pudiendo usarse señales en el mismo ancho de banda para usuarios distintos

Figura 22. Multiplexación por división de espacio (SDM).



2 Estudio codificación, modulación y multiplexación con Simulink

En este estudio con Simulink, primeramente vamos a comprobar como el usar una codificación Manchester que incorpora la información del reloj en los propios datos que se transmiten necesita de un ancho de banda mucho más elevado.

Después nos centraremos fundamentalmente en la transmisión de información digital mediante señales analógicas mediante las técnicas de modulación digital.

Finalmente veremos como es posible usar una técnica de multiplexación por división en frecuencia para sobre el mismo canal poder transmitir señales diferentes (a distintos destinatarios) sin que éstas interfieran entre ellas.

2.1 Codificación NRZ y Manchester

En este apartado se pretende comprobar cómo una codificación de tipo Manchester que incorpora en la señal que transmite la información del reloj del emisor, requiere de mayor ancho de banda del canal para ser transmitida respecto de una codificación NRZ.

Abrir el modelo almacenado en el fichero de nombre "Prac03_2_1.slx".

Vamos a considerar el modelo de Simulink de la Figura 23 para transmitir una señal digital correspondiente a la secuencia de 16 bits: [0 1 0 1 0 0 0 1 1 1 0 1 1 0 0 0], en el caso del modelo 1 usando una codificación NRZ y en el modelo 2 una codificación Manchester. Ambas codificaciones asumen que un bit "1" se corresponde con 1.5 voltios y un bit "0" con 0 voltios. Esto lo podemos hacer con los bloques "Gain" de la figura.

La manera de generar la señal será **similar** a la usada en la práctica 1, es decir, los bloques "Pulse Generator" permite generar señales periódicas cuadradas de amplitud **0.5**, periodo **0.5** segundos, fase **0** segundos y con un ancho del pulso del **50 %** del periodo total. Actuará como el reloj de la tarjeta de red de un PC.

Los bloques "Triggered Signal from Workspace" tienen como entrada los datos digitales que se quieren transmitir, es decir la secuencia de ceros y unos anterior y como salida la señal digital (entre 0 y 1) correspondiente con la duración adecuada de cada pulso.

Importante. En el bloque "Triggered Signal from Workspace", la secuencia de datos [0 1 0 1 0 0 0 1 1 1 0 1 1 0 0 0] que se quiere transmitir se configura de una manera particular: el primer bit (en este caso el 0) se configura en el parámetro de configuración "Initial output" y el resto de la secuencia (15 bits) se configuran en el parámetro de configuración "Signal".

Importante. Hay que fijarse que en esta práctica los pulsos del reloj tienen un **periodo de 0.5 segundos** (el 50% del periodo a 1 y el otro 50 % a 0) y que los bloques "Triggered Signal from Workspace" tienen el parámetro de configuración "Trigger type" en **"Rising edge"**, lo que significa que este bloque va a "sacar" un bit de la secuencia cuando se detecte un cambio de nivel positivo en la señal del reloj, es decir cada vez que la señal del reloj pase de 0 a 0.5.

Es un funcionamiento de un reloj de una tarjeta más realista.

Para el caso de la codificación Manchester además de lo anterior hay que incluir el código necesario antes de la transmisión definitiva de la señal digital, que se encargue de realizar la operación XOR entre la señal del reloj del emisor y la secuencia de bits que se quiere transmitir.

Importante. La mayor parte de los añadidos es para poder utilizar el bloque de operación entre bits "Bitwise Operator" para realizar la operación XOR. Este bloque solo permite señales de entrada de tipo booleana, devolviendo una señal booleana también. Sin embargo, los bloques "Pulse Generator1" y "Triggered Signal From Workspace1" devuelven valores de tipo real (double).



Por otro lado, el bloque "Gain1" solo acepta como entrada una señal de tipo real (double), así que hay que convertir el resultado booleano del XOR en double.

Los bloques "Data Type Conversion" y "Data Type Conversion1" permiten convertir la señal de entrada de tipo double en una señal de salida tipo booleana, TRUE (1) o FALSE (0), sin más que asignar el parámetro "Output data type" en "Boolean".

Finalmente, una vez hecha la operación XOR, el resultado booleano debe ser convertido a variable real (tipo double) con el bloque "Data Type Conversion2", asignando su parámetro "Output data type" en "double".

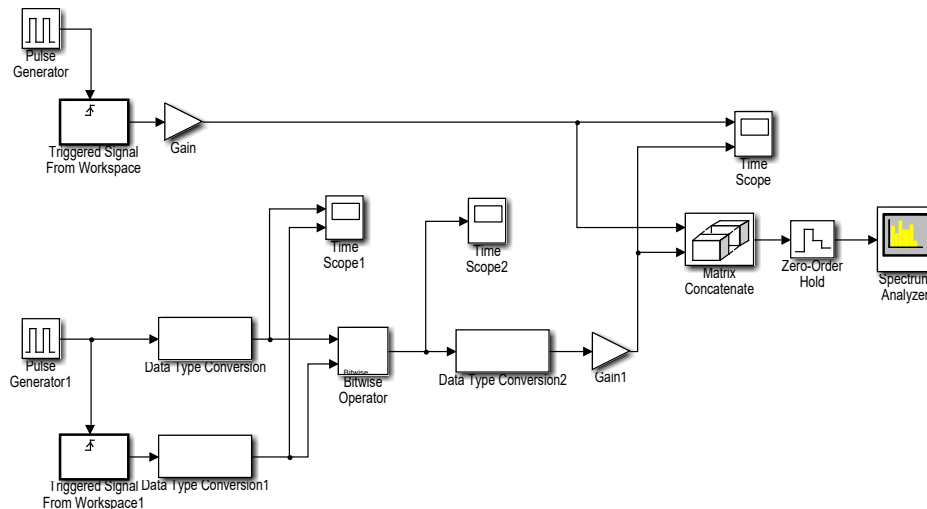


Figura 23. Modelo para la comparación entre la codificación NRZ y Manchester. En orden de arriba abajo: Modelo 1 (NRZ) y modelo 2 (Manchester).

Preguntas

- Comprobar con la representación en el tiempo cómo se ha realizado correctamente la codificación Manchester.
- Comparar en el tiempo cómo son las codificaciones NRZ y Manchester. Comprobar cómo la señal Manchester cambia de niveles de tensión de 0 voltios a 1.5 voltios muchas más veces que la codificación NRZ.
- Comparar los espectros de potencia de la señal codificada mediante Manchester y mediante NRZ. ¿Cuál es el efecto en la frecuencia de que la señal Manchester cambie de niveles de tensión mucho más frecuentemente?



Tiempo máximo de la simulación: 8 segundos	
Nombre del modelo:	Prac03_2_1.slx
Bloque:	Pulse Generator
Paleta:	Simulink -> Sources
Pulse type:	Time based
Time:	Use simulation time
Amplitude:	0.5
Period (sec):	0.5
Pulse Width (% of Period):	50
Phase delay (sec):	0
Bloque:	Pulse Generator1
Paleta:	Simulink -> Sources
Pulse type:	Time based
Time:	Use simulation time
Amplitude:	0.5
Period (sec):	0.5
Pulse Width (% of Period):	50
Phase delay (sec):	0.25
Bloque:	Triggered Signal From Workspace
Paleta:	DSP System Toolbox -> Signal Operations
Signal:	[1 0 1 0 0 0 1 1 1 0 1 1 0 0 0]
Trigger type:	Rising edge
Initial output:	0
Samples per frame:	1
Form output:	Setting zero
Bloque:	Triggered Signal From Workspace1
Paleta:	DSP System Toolbox -> Signal Operations
Signal:	[1 0 1 0 0 0 1 1 1 0 1 1 0 0 0]
Trigger type:	Falling edge
Initial output:	0
Samples per frame:	1
Form output:	Setting zero
Bloque:	Gain y Gain1
Paleta:	Simulink -> Math Operations
Gain:	1.5
Sample time:	-1
Bloque:	Matrix Concatenate
Paleta:	DSP System Toolbox -> Math Functions -> Matrices and Linear Algebra -> Matrix Operations
Number of inputs:	2
Mode:	Multidimensional array
Concatenate dimensión:	2
Bloque:	Zero-Order Hold
Paleta:	Simulink -> Discrete
Sample time:	0.0025
Bloque:	Data Type Conversion y Data Type Conversion1
Paleta:	Simulink -> Signal Attributes
Output data type:	boolean
Sample time:	-1
Bloque:	Bitwise Operator
Paleta:	Simulink -> Logic and Bit Operations
Operator:	XOR
Use bit mask:	Desactivado
Number of input ports:	2
Bloque:	Data Type Conversion2
Paleta:	Simulink -> Signal Attributes
Output data type:	double
Sample time:	-1
Bloque:	Spectrum Analyzer
Paleta:	DSP System Toolbox -> Sinks
Main Options:	Type:Power; Full Frequency Span; Window length: 3200
Trace Options:	Units: Watts
Parámetros:	Ver práctica 1
Bloque:	Time Scope, Time Scope1 y Time Scope1
Paleta:	DSP System Toolbox -> Sinks
Number of inputs:	2, y 1



2.2 Modulación digital por desplazamiento de amplitud (ASK)

En este apartado se va a estudiar la codificación digital ASK, es decir cuando interesa transmitir una secuencia digital mediante una señal analógica y en este proceso seleccionamos que el bit "0" se transmite como una señal de frecuencia 30 Hz, fase 0º y amplitud 0, y el bit "1" como una señal de frecuencia 30 Hz, fase 0º y amplitud 2 voltios.

Es decir, la diferencia en transmitir un bit "0" o un bit "1" está en la amplitud de la señal.

Abrir el modelo almacenado en el fichero de nombre "Prac03_2_2.slx".

Vamos a considerar el modelo de Simulink de la Figura 24 para transmitir una señal analógica (Modular mediante una señal portadora pasa banda) correspondiente a la secuencia de 16 bits: [0 1 0 1 0 0 0 1 1 1 0 1 1 0 0 0].

Importante. Hay que fijarse que en esta práctica los pulsos del reloj tienen un **periodo de 0.5 segundos** (el 50% del periodo a 1 y el otro 50 % a 0) y que los bloques "Triggered Signal from Workspace" tienen el parámetro de configuración "Trigger type" en "**Rising edge**", lo que significa que este bloque va a "sacar" un bit de la secuencia cuando se detecte un cambio de nivel positivo en la señal del reloj, es decir cada vez que la señal del reloj pase de 0 a 0.5.

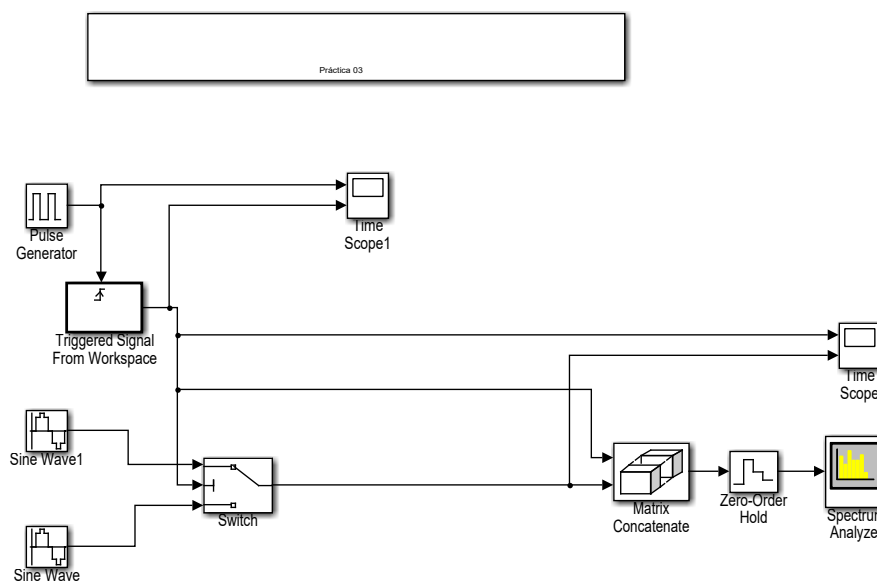


Figura 24. Modulación ASK.



Tiempo máximo de la simulación: 8 segundos	
Nombre del modelo: Prac03_2_2.slx	
Bloque:	Pulse Generator
Paleta:	Simulink -> Sources
Pulse type:	Time based
Time:	Use simulation time
Amplitude:	0.5
Period (sec):	0.5
Pulse Width (% of Period):	50
Phase delay (sec):	0
Bloque:	Triggered Signal From
Paleta:	DSP System Toolbox -> Signal Operations
Signal:	[1 0 1 0 0 0 1 1 1 0 1 1 0 0 0]
Trigger type:	Rising edge
Initial output:	0
Samples per frame:	1
Form output:	Setting zero
Bloque:	Sine Wave
Paleta:	Simulink -> Sources
Sine type:	Time based
Time:	Use simulation time
Amplitude:	0
Bias:	0
Frequency (rad/sec):	2*pi*30
Phase (rad):	0
Sample time:	0.0025
Bloque:	Sine Wave1
Paleta:	Simulink -> Sources
Sine type:	Time based
Time:	Use simulation time
Amplitude:	2
Bias:	0
Frequency (rad/sec):	2*pi*30
Phase (rad):	0
Sample time:	0.0025
Bloque:	Switch
Paleta:	Simulink -> Signal routing
Criteria:	u2 > Threshold
Threshold:	0
Sample time:	-1
Pestaña:	Signal Attributes -> Output data type: double
Bloque:	Matrix Concatenate
Paleta:	DSP System Toolbox -> Math Functions -> Matrices and Linear Algebra -> Matrix Operations
Number of inputs:	2
Mode:	Multidimensional array
Concatenate dimensión:	2
Bloque:	Zero-Order Hold
Paleta:	Simulink -> Discrete
Sample time:	0.0025
Bloque:	Spectrum Analyzer
Paleta:	DSP System Toolbox -> Sinks
Main Options:	Type:Power; Full Frequency Span; Window length: 3200
Trace Options:	Units: Watts
Parámetros:	Ver práctica 1
Bloque:	Time Scope y Time Scope1
Paleta:	DSP System Toolbox -> Sinks
Number of inputs:	2



Preguntas

- a) Visualizar la señal analógica en el tiempo.
- b) Visualizar el espectro de frecuencias de la señal analógica y de la señal original correspondiente a la secuencia de bits. Suponiendo que esta secuencia de bits se podría corresponder a la transmisión directamente mediante una señal digital NRZ. ¿Qué ha ocurrido con el espectro de la señal analógica transmitida?
- c) Clasificar todas las señales:

Señal	Secuencia de bits (Señal digital NRZ)	Señal analógica (ASK)
Señal periódica / No periódica		
Ancho de banda absoluto		
Componente de frecuencia con mayor potencia		
Ancho de banda relativo <i>Aquella banda de frecuencias cuyas componentes tienen más del 1% de potencia respecto de la componente de frecuencia de mayor potencia</i>		
Limitada en banda / no limitada en banda		
Banda base / pasa banda		
Ancho de pulso digital		No aplica
Duración de la transmisión		
Velocidad de transmisión bps		

- d) Cargar el fichero "Prac03_2_2d.slx" visualizar lo que ocurre (en tiempo y frecuencia) donde se ha usado la misma modulación ASK anterior, describir ahora que es lo que ocurre y rellenar la tabla siguiente:

Señal	Secuencia de bits (Señal digital NRZ)	Señal analógica (ASK)
Ancho de banda absoluto		
Componente de frecuencia con mayor potencia		
Ancho de banda relativo <i>Aquella banda de frecuencias cuyas componentes tienen más del 1% de potencia respecto de la componente de frecuencia de mayor potencia</i>		
Ancho de pulso digital		No aplica
Duración de la transmisión		
Velocidad de transmisión bps		



- e) Se está usando una señal analógica de frecuencia 30 Hz, a la vista de los resultados anteriores ¿se está consiguiendo transmitir a la máxima velocidad en bps que permite esta señal? ¿Cuál es la velocidad máxima en bps que se podría alcanzar? En esta situación, ¿cuánto duraría la transmisión de la misma secuencia de bits del resto de apartados?, ¿Qué ocurre con el ancho de banda relativo de esta nueva señal respecto de la original del apartado c)?

Modificar el modelo para conseguir la máxima velocidad y grabarlo en el fichero "Prac03_2_2e.slx".

2.3 Modulación digital por desplazamiento de frecuencia (FSK)

En este apartado se va a estudiar la codificación digital FSK, es decir cuando interesa transmitir una secuencia digital mediante una señal analógica y en este proceso seleccionamos que el bit "0" se transmite como una señal de frecuencia 10 Hz, fase 0º y amplitud 2 voltios, y el bit "1" como una señal de frecuencia 40 Hz, fase 0º y amplitud 2 voltios.

Es decir, la diferencia en transmitir un bit "0" o un bit "1" está en la frecuencia de la señal.

Abrir el modelo almacenado en el fichero de nombre "Prac03_2_3.slx".

Vamos a considerar el modelo de Simulink de la Figura 25 para transmitir una señal analógica (modular mediante una señal portadora pasabanda) correspondiente a la secuencia de 16 bits: [0 1 0 1 0 0 0 1 1 1 0 1 1 0 0 0].

Importante. Se usará el modelo del reloj igual que el del apartado 2.2.

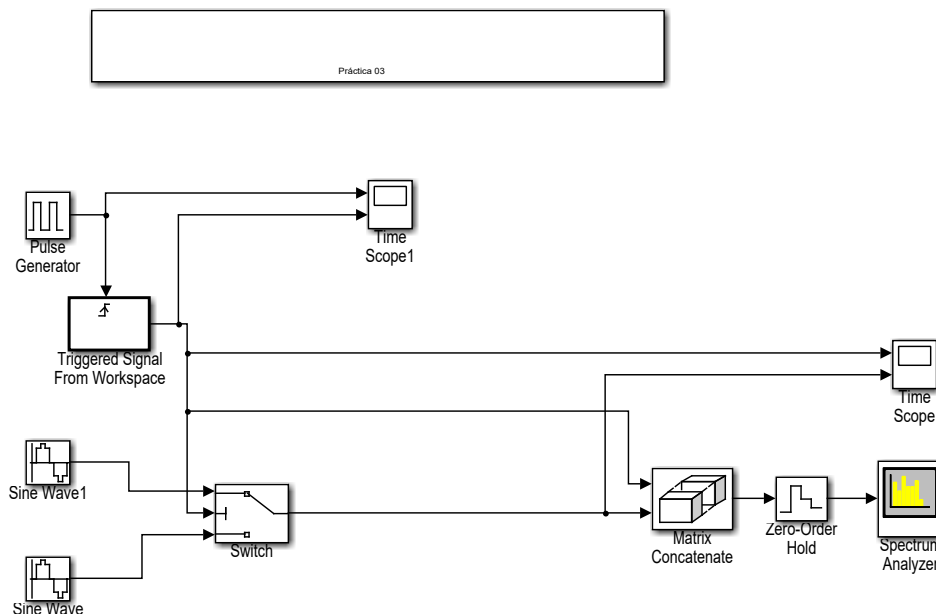


Figura 25. Modulación FSK.

Preguntas

- Visualizar la señal analógica en el tiempo.
- Visualizar el espectro de frecuencias de la señal analógica y de la señal original correspondiente a la secuencia de bits.

¿Qué ha ocurrido con el espectro de la señal analógica transmitida?

¿Aproximadamente cuál es el ancho de banda relativo de esta nueva señal analógica?



Tiempo máximo de la simulación: 8 segundos	
Nombre del modelo:	Prac03_2_3.slx
Bloque:	Pulse Generator
Paleta:	Simulink -> Sources
Pulse type:	Time based
Time:	Use simulation time
Amplitude:	0.5
Period (sec):	0.5
Pulse Width (% of Period):	50
Phase delay (sec):	0
Bloque:	Triggered Signal From
Paleta:	DSP System Toolbox -> Signal Operations
Signal:	[1 0 1 0 0 0 1 1 1 0 1 1 0 0 0]
Trigger type:	Rising edge
Initial output:	0
Samples per frame:	1
Form output:	Setting zero
Bloque:	Sine Wave
Paleta:	Simulink -> Sources
Sine type:	Time based
Time:	Use simulation time
Amplitude:	2
Bias:	0
Frequency (rad/sec):	2*pi*10
Phase (rad):	0
Sample time:	0.0025
Bloque:	Sine Wave1
Paleta:	Simulink -> Sources
Sine type:	Time based
Time:	Use simulation time
Amplitude:	2
Bias:	0
Frequency (rad/sec):	2*pi*40
Phase (rad):	0
Sample time:	0.0025
Bloque:	Switch
Paleta:	Simulink -> Signal routing
Criteria:	u2 > Threshold
Threshold:	0
Sample time:	-1
Pestaña:	Signal Attributes -> Output data type: double
Bloque:	Matrix Concatenate
Paleta:	DSP System Toolbox -> Math Functions -> Matrices and Linear Algebra -> Matrix Operations
Number of inputs:	2
Mode:	Multidimensional array
Concatenate dimensión:	2
Bloque:	Zero-Order Hold
Paleta:	Simulink -> Discrete
Sample time:	0.0025
Bloque:	Spectrum Analyzer
Paleta:	DSP System Toolbox -> Sinks
Main Options:	Type:Power; Full Frequency Span; Window length: 3200
Trace Options:	Units: Watts
Parámetros:	Ver práctica 1
Bloque:	Time Scope y Time Scope1
Paleta:	DSP System Toolbox -> Sinks
Number of inputs:	2



2.4 Modulación digital por desplazamiento de fase (PSK)

En este apartado se va a estudiar la codificación digital PSK, es decir cuando interesa transmitir una secuencia digital mediante una señal analógica y en este proceso seleccionamos que el bit "0" se transmite como una señal de frecuencia 10 Hz, fase 180° ($o \pi$ rad) y amplitud 2 voltios, y el bit "1" como una señal de frecuencia 10 Hz, fase 0° y amplitud 2 voltios.

Es decir, la diferencia en transmitir un bit "0" o un bit "1" está en la fase de la señal.

Abrir el modelo almacenado en el fichero de nombre "Prac03_2_4.slx".

Vamos a considerar el modelo de Simulink de la Figura 26 para transmitir una señal analógica (modular mediante una señal portadora pasabanda) correspondiente a la secuencia de 16 bits: [0 1 0 1 0 0 0 1 1 1 0 1 1 0 0 0].

Importante. Se usará el modelo del reloj igual que el del apartado 2.2.

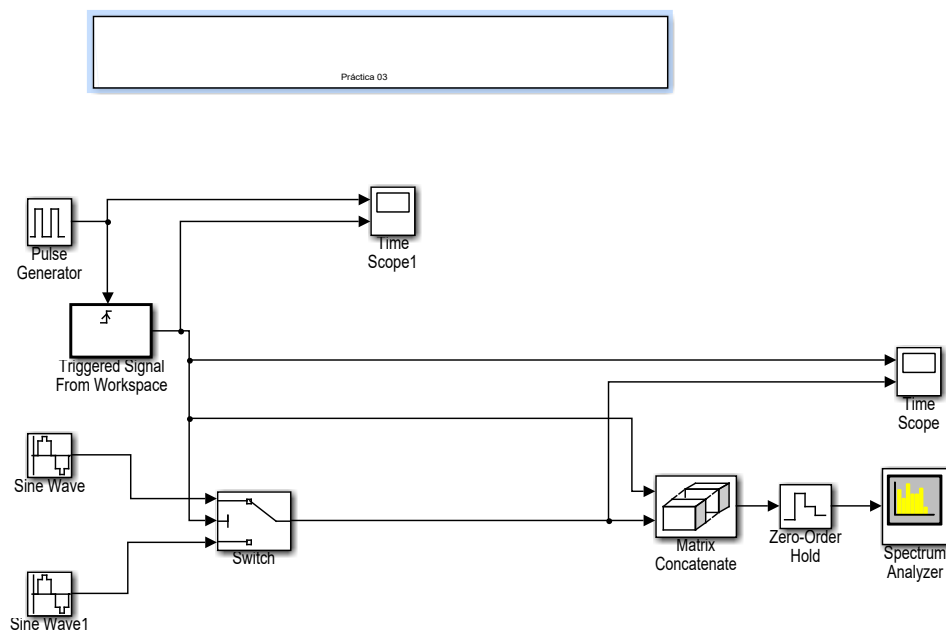


Figura 26. Modulación PSK.

Preguntas

- Visualizar la señal analógica en el tiempo.
- Visualizar el espectro de frecuencias de la señal analógica y de la señal original correspondiente a la secuencia de bits.
- Compara el espectro de la modulación ASK con el de la modulación PSK. ¿Qué observas? ¿A qué se pueden deber las diferencias?



Tiempo máximo de la simulación: 8 segundos	
Nombre del modelo:	Prac03_2_4.slx
Bloque:	Pulse Generator
Paleta:	Simulink -> Sources
Pulse type:	Time based
Time:	Use simulation time
Amplitude:	0.5
Period (sec):	0.5
Pulse Width (% of Period):	50
Phase delay (sec):	0
Bloque:	Triggered Signal From
Paleta:	DSP System Toolbox -> Signal Operations
Signal:	[1 0 1 0 0 0 1 1 1 0 1 1 0 0 0]
Trigger type:	Rising edge
Initial output:	0
Samples per frame:	1
Form output:	Setting zero
Bloque:	Sine Wave
Paleta:	Simulink -> Sources
Sine type:	Time based
Time:	Use simulation time
Amplitude:	2
Bias:	0
Frequency (rad/sec):	2*pi*10
Phase (rad):	0
Sample time:	0.0025
Bloque:	Sine Wave1
Paleta:	Simulink -> Sources
Sine type:	Time based
Time:	Use simulation time
Amplitude:	2
Bias:	0
Frequency (rad/sec):	2*pi*10
Phase (rad):	pi
Sample time:	0.0025
Bloque:	Switch
Paleta:	Simulink -> Signal routing
Criteria:	u2 > Threshold
Threshold:	0
Sample time:	-1
Pestaña:	Signal Attributes -> Output data type: double
Bloque:	Matrix Concatenate
Paleta:	DSP System Toolbox -> Math Functions -> Matrices and Linear Algebra -> Matrix Operations
Number of inputs:	2
Mode:	Multidimensional array
Concatenate dimensión:	2
Bloque:	Zero-Order Hold
Paleta:	Simulink -> Discrete
Sample time:	0.0025
Bloque:	Spectrum Analyzer
Paleta:	DSP System Toolbox -> Sinks
Main Options:	Type:Power; Full Frequency Span; Window length: 3200
Trace Options:	Units: Watts
Parámetros:	Ver práctica 1
Bloque:	Time Scope y Time Scope1
Paleta:	DSP System Toolbox -> Sinks
Number of inputs:	2



2.5 Multiplexación por división en frecuencia

En este apartado se pretende comprobar cómo una multiplexación por división en frecuencia puede ser usada para transmitir varias señales a la vez de varios emisores diferentes pudiendo el receptor saber perfectamente qué información/señal envió cada uno de los emisores.

Además, este tipo de multiplexación, para poder ser usada para transmitir **información digital** de varios emisores a la vez, es necesario combinarla con algún tipo de **modulación digital** en cada emisor individual. En concreto usaremos **la modulación digital por desplazamiento en amplitud (ASK)**, pudiendo utilizar el código usado en la práctica 3, apartado 2.2.

Abrir el modelo almacenado en el fichero de nombre "Prac03_2_5.slx".

El esquema general del proceso de comunicación se corresponde con el mostrado en la Figura 27, es decir:

- Dos emisores quieren transmitir información digital. Por ejemplo, dos computadoras.
- Un medio físico compartido por los dos emisores y el receptor. Por ejemplo, un cable coaxial.
- Un receptor que recibe las señales procedentes de los emisores.
- Cada emisor usa una técnica de modulación digital por desplazamiento de amplitud (ASK), generando cada uno una señal analógica a distinta frecuencia mediante el cable modem.
- En el cable coaxial compartido las señales analógicas de cada emisor se suman llegando dicha señal suma al receptor.
- El receptor que dispone de un cable modem recibe la señal analógica suma y la filtra usando dos filtros pasa banda, de esta manera es capaz de aislar cada una de las señales originales presentes en la suma.
- Una vez aisladas las señales de los dos emisores, el cable modem del receptor realiza la operación contraria de modulación, es decir, convierte cada señal analógica a datos digitales.



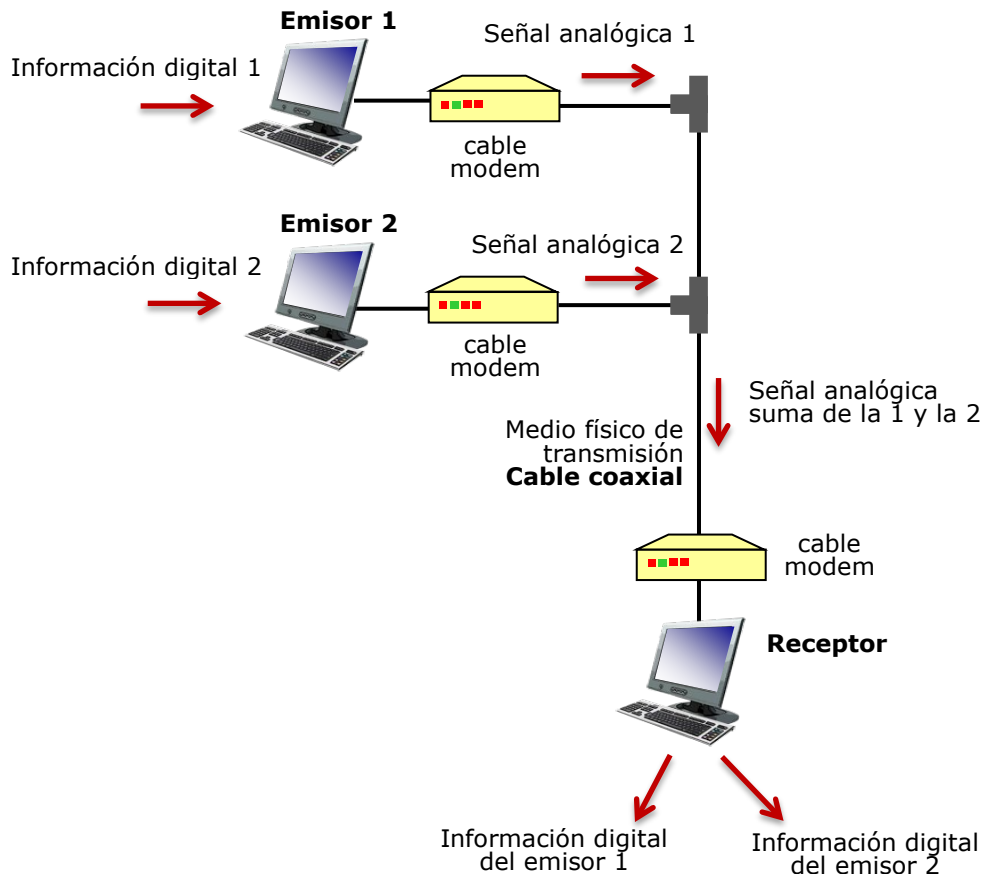


Figura 27. Esquema general de la multiplexación por división en frecuencia para la comunicación entre dos emisores y un receptor.

En particular, relacionado con la multiplexación:

- El ancho de banda completo del medio físico, es decir del cable coaxial, se divide en dos canales.
- Al emisor 1 se le asigna un canal de transmisión de ancho de banda 20 Hz (entre 10 Hz y 30 Hz).
- Al emisor 2 se le asigna un canal de transmisión de ancho de banda 20 Hz (entre 40 Hz y 60 Hz).
- Se deja una banda de guarda de 10 Hz entre ambos canales.

En particular, relacionado con la modulación digital (ASK) en los emisores:

- El emisor 1 usa una señal seno de frecuencia 20 Hz y amplitud 2 para transmitir un bit "1" y amplitud 0 para transmitir un bit "0".
- El emisor 2 usa una señal seno de frecuencia 50 Hz y amplitud 2 para transmitir un bit "1" y amplitud 0 para transmitir un bit "0".

En particular, relacionado con la información digital que quiere enviar cada emisor al receptor:

- El emisor 1 quiere enviar la secuencia de 16 bits [0 1 0 1 0 0 0 1 1 1 0 1 0 0 0 0] usando un reloj interno de periodo 0.5 segundos y amplitud 0.5 voltios.
- El emisor 2 quiere enviar la secuencia de 16 bits [0 1 0 1 0 0 1 1 0 1 1 1 0 1 0 1] usando un reloj interno de periodo 0.5 segundos y amplitud 0.5 voltios.



- En ambos casos, el comportamiento del reloj será la de extraer un bit de cada secuencia cuando se detecte un cambio de nivel positivo en la señal del reloj, es decir cada vez que la señal del reloj pase de 0 a 0.5 voltios, opción "Rising edge"

Filtrado de la señal en el receptor. El cable modem del receptor deberá usar dos filtros pasa banda para asilar cada señal, sabiendo que el filtro 1 se corresponderá con el emisor 1 y el filtro 2 con el emisor 2:

- **Filtro 1.** Tipo "Butterworth", "Filter order 8", "bandpass", frecuencia inferior 10 Hz y frecuencia superior 30 Hz. Es decir, solo va a dejar pasar la señal que tenga contenidos en frecuencia entre 10 Hz y 30 Hz, el resto de componentes las va a hacer 0.
- **Filtro 2.** Tipo "Butterworth", "Filter order 8", "bandpass", frecuencia inferior 40 Hz y frecuencia superior 60 Hz. Es decir, solo va a dejar pasar la señal que tenga contenidos en frecuencia entre 40 Hz y 60 Hz, el resto de componentes las va a hacer 0.

Vamos a considerar el modelo de Simulink de la Figura 28 para simular el comportamiento del esquema de transmisión dado en la Figura 27, junto con los correspondientes bloques para visualizar las señales en el tiempo y en la frecuencia, definidos de la misma manera que en las prácticas anteriores.

Finalmente, cualquier medio físico actúa sumando todas las señales que se le inyecten en él. Esto lo podemos simular con el bloque "add".

Simular el comportamiento durante **8 segundos**.

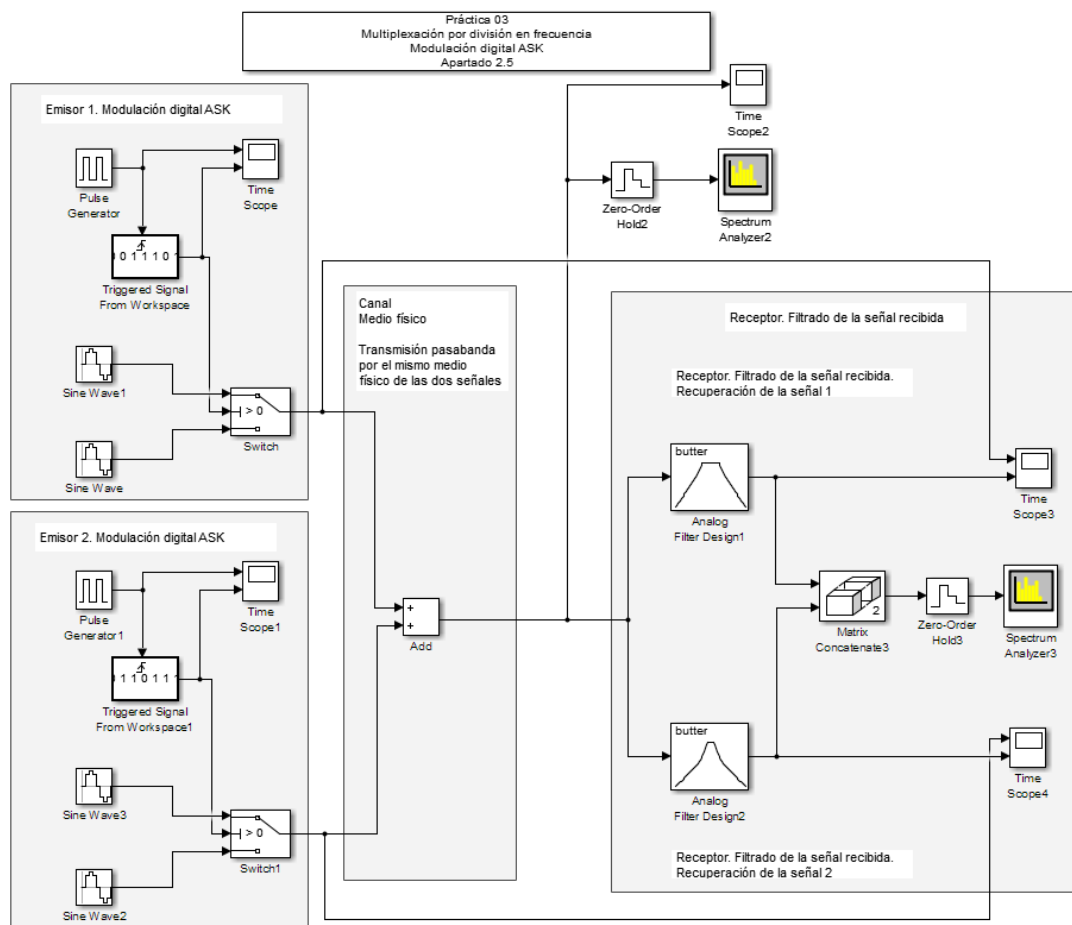


Figura 28. Modelo para simular la multiplexación por división en frecuencia. Formado por dos emisores y un receptor que comparten el mismo medio de transmisión.



Preguntas

- a) Comprobar el funcionamiento del modelo propuesto e identificar los elementos.
- b) Comprobar cómo es la señal en el tiempo que **llega** al receptor.
- c) Comprobar cómo es el espectro de frecuencias de la señal que **llega** al receptor. ¿Qué se puede concluir?
- d) Comparar la señal analógica aislada por el cable modem mediante el filtro 1 con la transmitida por el emisor 1. ¿Se ha podido recuperar la señal?
- e) Comparar la señal analógica aislada por el cable modem mediante el filtro 2 con la transmitida por el emisor 2. ¿Se ha podido recuperar la señal?
- f) ¿Qué efecto ves en las señales recuperadas?
- g) ¿Qué ocurre si en la multiplexación se asignan canales con ancho de banda **muy estrecho**, por ejemplo de solo 4 Hz, canal 1 entre 18 Hz y 22 Hz y canal 2 entre 48 Hz y 52 Hz? Justificar la respuesta.

Grabar el nuevo modelo en el fichero "**Prac03_2_5g.slx**".

- h) ¿Qué ocurre si en la multiplexación se asignan canales con ancho de banda **muy juntos**, por ejemplo 20 Hz, canal 1 entre 10 Hz y 30 Hz y canal 2 entre 20 Hz y 40 Hz? Asumir que la modulación digital del emisor 1 se realiza a 20 Hz y la del emisor 2 a 30 Hz. Justificar la respuesta.

Grabar el nuevo modelo en el fichero "**Prac03_2_5h.slx**".



IV Problemas multiplexación

- 1) Sean las siguientes cuatro secuencias de 8 bits (A, B, C y D), comprobar que todas las secuencias son ortogonales entre sí.

$$A = (-1 \ -1 \ -1 \ +1 \ +1 \ -1 \ +1 \ +1)$$

$$B = (-1 \ -1 \ +1 \ -1 \ +1 \ +1 \ +1 \ -1)$$

$$C = (-1 \ +1 \ -1 \ +1 \ +1 \ +1 \ -1 \ -1)$$

$$D = (-1 \ +1 \ -1 \ -1 \ -1 \ -1 \ +1 \ -1)$$

- 2) Eligiendo las secuencias anteriores A y B, comprobar que se cumplen las propiedades de la ortogonalidad respecto del producto interior vistas en la parte teórica.
- 3) Se dispone de 4 emisores A, B, C y D a los que se les asigna las secuencias del punto 1 anterior. Comprobar para cada uno de los 6 mensajes siguientes recibidos por el receptor y formados cada uno por un solo bit de información de cada emisor, que pueden ser decodificados adecuadamente en el receptor permitiendo conocer quién los envió.

$$S_1 = C$$

$$S_2 = B+C$$

$$S_3 = A+\overline{B}$$

$$S_4 = A+\overline{B}+C$$

$$S_5 = A+B+C+D$$

$$S_6 = A+B+\overline{C}+D$$

V Bibliografía

Stallings. Comunicaciones y redes de computadoras, Pearson. 2004





Grado en Ingeniería Informática

REDES

PRÁCTICA 4

Instalación de herramientas para emulación y análisis de redes

Docentes:

Alejandro Merino

Daniel Sarabia Ortiz

*Dpto. de Ingeniería Electromecánica
Área de Ingeniería de Sistemas y Automática*

Versión 3.8

Fecha 28/03/2022 12:57

*Esta obra está sujeta a la licencia Reconocimiento 4.0 Internacional de
Creative Commons. Para ver una copia de esta licencia, visite
<http://creativecommons.org/licenses/by/4.0/>*



Índice de contenidos

I	INTRODUCCIÓN	3
II	OBJETIVOS	4
III	INSTALACIÓN DE VIRTUALBOX	5
	1 Agregar máquinas virtuales en VirtualBox	5
	2 Clonación de máquinas virtuales en VirtualBox	7
	3 Instalación de UbuntuServer en VirtualBox	7
IV	INSTALACIÓN DE GNS3	10
V	INSTALACIÓN DE VMWARE Y GNS3 VM	12
	1 Instalación de VMware Workstation Player	12
	2 Descarga e instalación de la máquina virtual GNS3 para VMware	13
	3 Adición de máquinas virtuales en GNS3	13
	4 Adición de Cisco IOS en GNS3	14
	4.1 Cisco IOS para un Router	14
	4.2 Cisco IOS para un Switch	18
VI	INSTALACIÓN DE WIRESHARK	21
VII	EJERCICIO DE PRUEBA	22
VIII	SUBSANACIÓN DE PROBLEMAS DE INSTALACIÓN	26
	1 Máquina Virtual de GNS3	26
	2 Instalación VMware	28
	3 Problemas con la virtualización en GNS3	29
	4 Problemas con WireShark	30



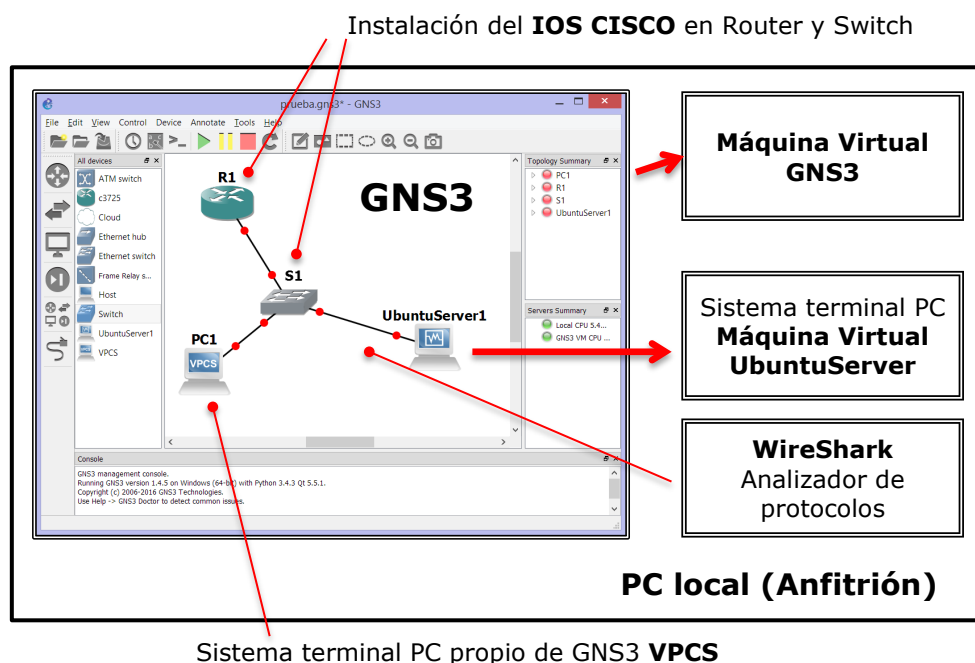
I Introducción

Para poder realizar prácticas de laboratorio utilizando una red lo más parecida posible a una red real, se van a utilizar emuladores de redes que nos van a permitir montar arquitecturas de red que, aun siendo sencillas, van a permitir ensayar las principales configuraciones existentes en una red real.

Esta emulación va a realizarse mediante el software **GNS3**, que permite emular el comportamiento de una red de computadores que incluya sistemas terminales tipo PC, routers y switches. Tanto los routers como los switches son dispositivos que ejecutan sistemas operativos y GNS3 permite instalar sistemas operativos reales de cualquier router o switch comercial. Nosotros usaremos los sistemas operativos de CISCO¹, **Cisco IOS** una de las empresas más importantes del sector.

En GNS3 se utilizarán dos tipos distintos de sistemas terminales. Por un lado, los llamados **VPCS** (Virtual PC Simulator) que permiten algunas operaciones básicas para comprobar la conectividad de una red. Por otro lado, se utilizarán **máquinas virtuales Linux**, en este caso con el sistema operativo **UbuntuServer**. Es decir que tendremos varias máquinas virtuales, cada una correspondiente a un PC comunicándose entre ellas, con routers, switches y otros sistemas terminales propios de GNS3 mediante la topología de red que configuremos en GNS3.

Además, para que el software GNS3 consuma menos recursos del equipo anfitrión, se usará una máquina virtual GNS3 para que parte del software GNS3 se ejecute en ella.



Finalmente, para poder analizar el tráfico de datos que intercambian los dispositivos que forman las redes que montemos, usaremos el software **WireShark**, que permite analizar los protocolos usados y la configuración de los paquetes transmitidos.

Por tanto, **para realizar las prácticas del curso es necesario instar y configurar** las diferentes herramientas que se van a utilizar, que es lo que se describe en esta sesión. Las comprobaciones del funcionamiento de todo lo instalado las iremos haciendo en las prácticas sucesivas.

¹ Cisco Systems es una empresa global con sede en San José (California, Estados Unidos), principalmente dedicada a la fabricación, venta, mantenimiento y consultoría de equipos de telecomunicaciones.



II Objetivos

Instalar y configurar el software de emulación de redes que se utilizará en las prácticas siguientes. A lo largo de este manual se explica cómo instalar y qué precauciones hay que tener con dicho software para el sistema operativo Windows (**probado en Windows 8.1 y Windows 10, ambos de 64 bits**), por lo que se recomienda leerlo atentamente.

Todo el software es gratuito y se puede descargar de las correspondientes páginas web, sin embargo, en el curso **suministramos todo el software necesario**, ver Tabla siguiente.

En cuanto a las dos máquinas virtuales Ubuntu Server, UbuntuServer_1 y UbuntuServer_2 se suministran ya creadas para VirtualBox, **no siendo necesario** la creación de una nueva a partir de la imagen del sistema operativo. Simplemente hay que agregar esas dos máquinas virtuales en Virtual Box.

Por tanto, será necesario:

- Instalar VirtualBox y VirtualBox Extension Pack.
- Agregar Máquinas Virtuales UbuntuServer_1 y UbuntuServer_2.
- Instalar VMware Workstation Player.
- Instalación de GNS3.
- Añadir IOS Routers y Switches en GNS3
- Instalación de Wireshark.

Software necesario (*)	Versión	Suministrado en las prácticas
VMware Workstation Player	16.1.0	VMware-player-16.1.0-17198959.exe
VirtualBox	6.1.16	VirtualBox-6.1.16-140961-Win.exe
VirtualBox Extension Pack	6.1.16	Oracle_VM_VirtualBox_Extension_Pack-6.1.16.vbox-extpack
Sistema Operativo Ubuntu Server	14.04.6 LTS	
Máquina Virtual Ubuntu Server ya creada para VirtualBox, incluye servidor web y servidor DHCP <i>Usuario: redes</i> <i>Password: redes</i>	14.04.6 LTS	UbuntuServer_1 VirtualBox.zip (.vbox) 14_04_6_LTS
Máquina Virtual Ubuntu Server ya creada para VirtualBox <i>Usuario: redes</i> <i>Password: redes</i>	14.04.4 LTS	UbuntuServer_2 VirtualBox.zip (.vbox) 14_04_4_LTS
GNS3	2.2.17	GNS3-2.2.17-all-in-one-regular.exe
Máquina Virtual GNS3 para VMware	2.2.17	GNS3.VM.VMware.Workstation.2.2.17.zip (GNS3 VM.ova)
IOS Routers y Switches en GNS3. Imagen del IOS del router C3725 (Cisco)		c3725-adventerprisek9-mz.124-15.T14.bin
WireShark	3.4.2	Wireshark-win64-3.4.2.exe
(*) Todo versiones para Windows 64 bits		



III Instalación de VirtualBox



VirtualBox es un software de Virtualización para arquitecturas x86 y AMD64/Intel64, que permite instalar sistemas operativos “invitados” dentro de un sistema operativo “anfitrión”. Está disponible como Software Open Source bajo licencia GNL v2.

La descarga de la aplicación puede realizarse desde la página web:

<https://www.virtualbox.org/>

Concretamente las descargas del software desde la web:

<https://www.virtualbox.org/wiki/Downloads>

En función del sistema operativo del que dispongamos se deberá instalar una versión u otra de VirtualBox, pero se recomienda instalar a partir de la versión **6.1.16**.

Aunque en estas prácticas no lo vamos a utilizar, puede ser útil descargar e instalar también el **VirtualBox Extension Pack** (all platforms, versión 6.1.16), que aparece en la misma página que la descarga de VirtualBox y que instala características adicionales en las máquinas virtuales como el reconocimiento de los dispositivos USB.

La instalación es bastante sencilla y basta con seguir los pasos que se indican durante el proceso de instalación.

1 Agregar máquinas virtuales en VirtualBox

En GNS3 vamos a conectar máquinas virtuales que emularán a distintos dispositivos en una red. Las máquinas que vamos a utilizar serán máquinas virtuales Ubuntu desde VirtualBox.

Ubuntu es un sistema operativo basado en GNU/Linux y que se distribuye como software libre. Vamos a utilizar Ubuntu server ya que no carga interfaz de usuario y será más ligero a la hora de ejecutar varias máquinas virtuales en el equipo.

Se suministran dos máquinas virtuales del servidor Ubuntu para VirtualBox:

- **UbuntuServer_1.** Máquina Virtual Ubuntu Server ya creada para VirtualBox que incluye un servidor web y un servidor DHCP ya configurados.
- **UbuntuServer_2.** Máquina Virtual Ubuntu Server ya creada para VirtualBox con la instalación mínima para funcionar en red.

Descomprimir el zip donde se encuentra cada máquina virtual en la ubicación de destino que se desee. Cada máquina virtual está formada por una carpeta con el nombre de la máquina que contiene:

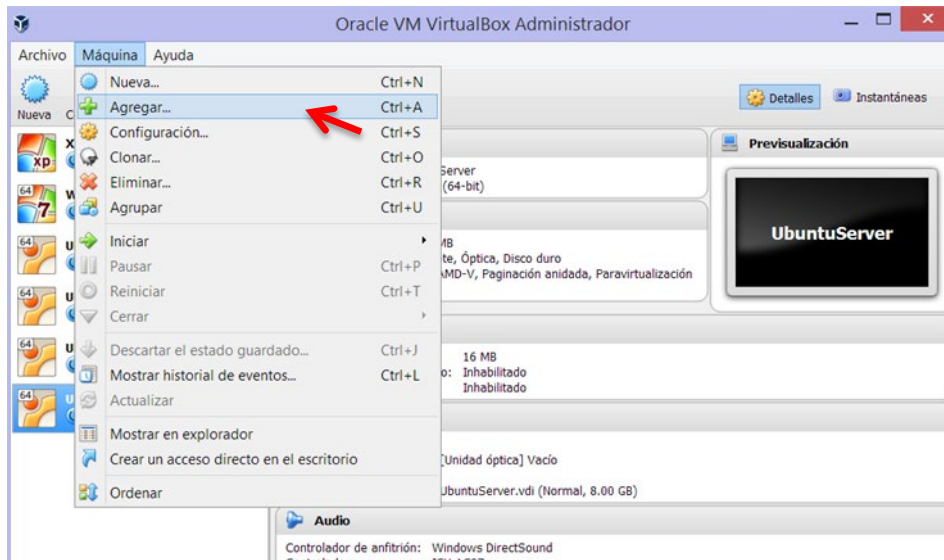
```

UbuntuServer_1
  Logs
  UbuntuServer_1.vbox
  UbuntuServer_1.vbox-prev
  UbuntuServer_1.vdi

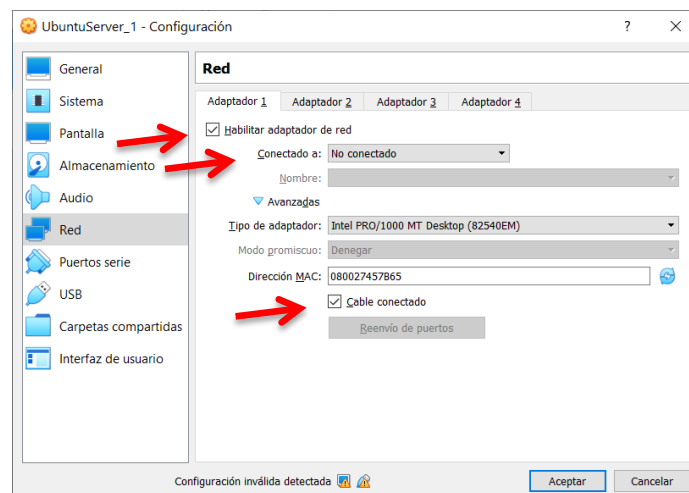
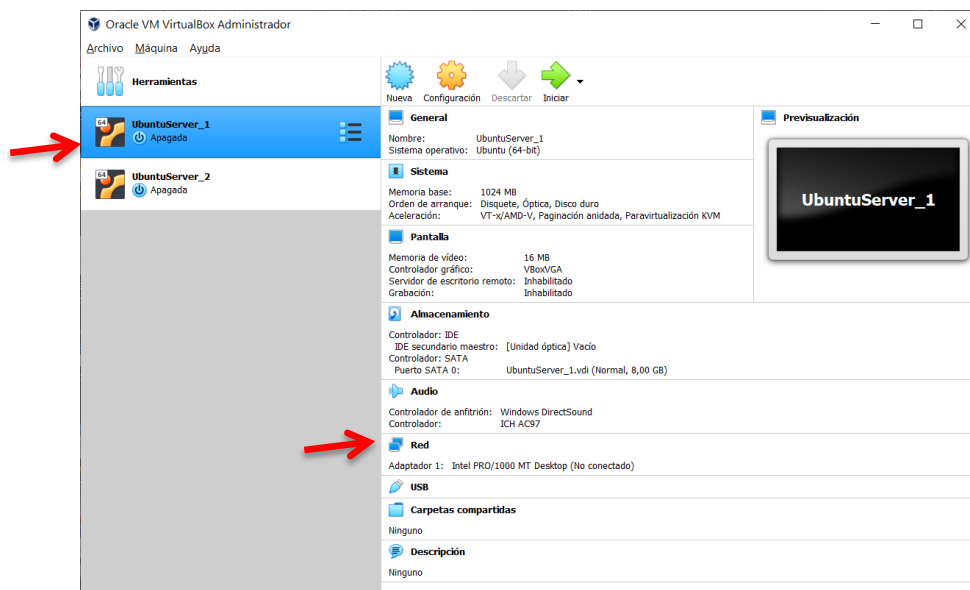
```

La extensión vbox se corresponde con máquinas virtuales para VirtualBox y la extensión vdi hace referencia a una imagen de disco de una máquina virtual en VirtualBox. A continua agregar directamente la máquina virtual a VirtualBox seleccionando “Añadir” o “Agregar” según la versión de Virtual Box y seleccionando “UbuntuServer_1.vbox”. Después proceder de la misma manera para la máquina UbuntuServer_2.





Importante. Para que funcione correctamente todo lo relacionado con redes hemos de configurar el menú de red de la máquina virtual Ubuntu, pinchar en red y seleccionar en la pestaña "Adaptador 1", **No conectado** en el desplegable "Conectado a" y además marcar **Cable conectado**.



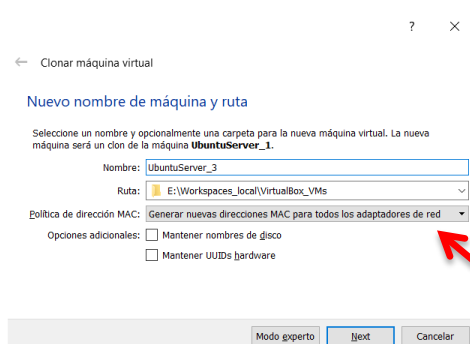
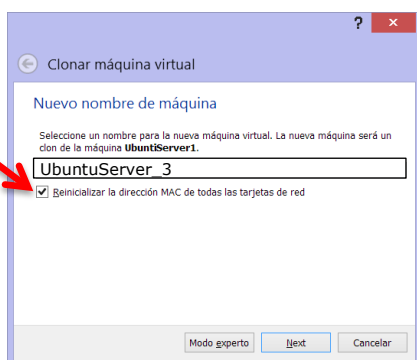
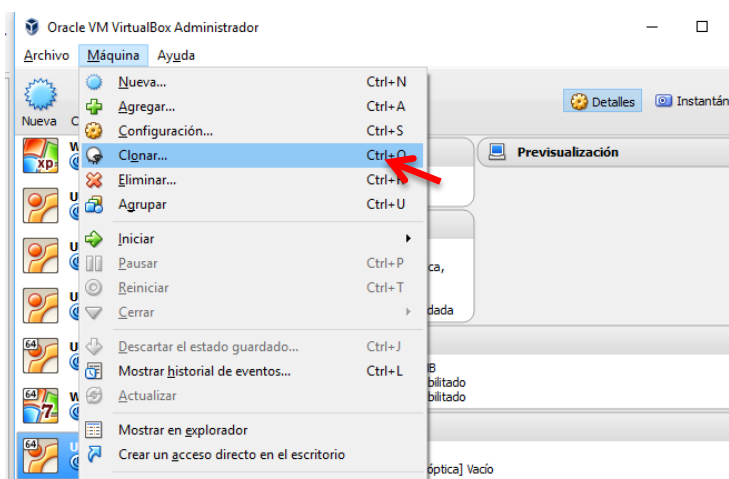
2 Clonación de máquinas virtuales en VirtualBox

Este paso no es necesario ya que se suministran las dos máquinas virtuales ya creadas en Virtual Box.

Una vez que se disponga de una máquina virtual del servidor Ubuntu en VirtualBox, se pueden realizar copias de la máquina virtual original seleccionando "Clonar", y dando otro nombre como UbuntuServer_3. De esta manera no hay que realizar la instalación del sistema operativo Ubuntu otra vez.

Nota. Para las prácticas **es suficiente** con tener las dos máquinas virtuales Ubuntu Server suministradas: UbuntuServer_1 y UbuntuServer_2. En todo caso si se quiere disponer de otra máquina virtual se recomienda clonar la UbuntuServer_2 ya que es la que ocupa el mínimo espacio y tiene lo mínimo para funcionar.

Importante. Si se hace una clonación hay que marcar la opción "Reinicializar la dirección MAC de todas las tarjetas de red" o en las versiones más actuales "Generar nuevas direcciones MAC para todos los adaptadores de red" para que cada máquina virtual Ubuntu tenga una dirección MAC diferente y no entren en conflicto unas con otras al conectarlas en la misma red.



3 Instalación de UbuntuServer en VirtualBox

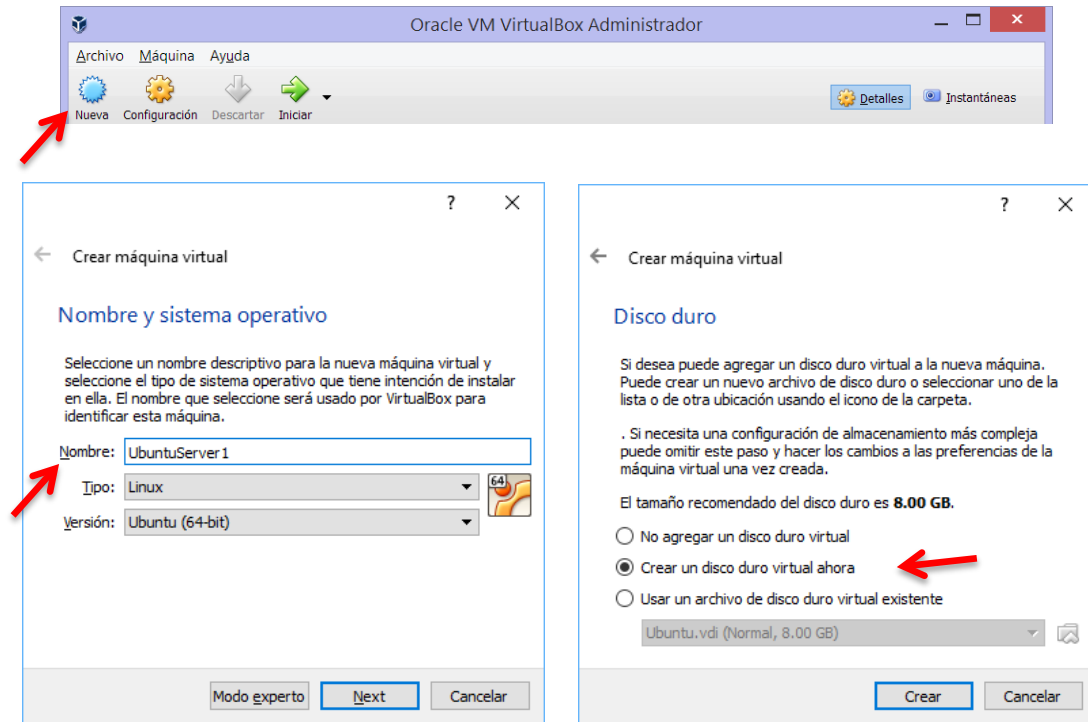
Este paso no es necesario ya que se suministran las dos máquinas virtuales ya creadas en Virtual Box.

Si se dispone de una imagen ISO del sistema operativo Ubuntu Server, por ejemplo suministrada en la página web:



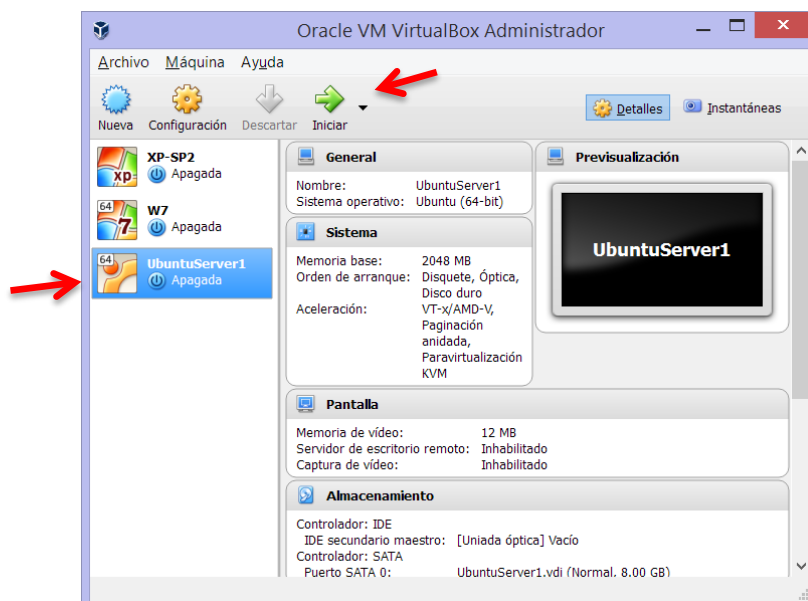
<https://www.ubuntu.com/download/alternative-downloads>

podemos descargar el sistema operativo **Ubuntu Server 14.04.6 LTS** contenido en la **ISO ubuntu-14.04.6-server-amd64.iso** y crear una nueva máquina virtual desde cero. Arrancamos VirtualBox y creamos una máquina Virtual nueva de nombre UbuntuServer1, pinchando en el icono "Nueva":

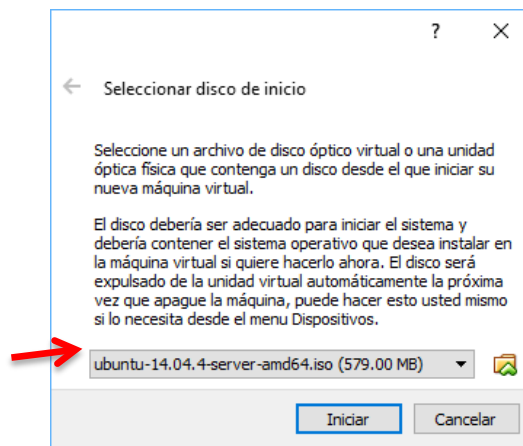


Seleccionamos las opciones que se dan **por defecto**. Si se dispone de un equipo con recursos limitados se puede reducir la cantidad de memoria que se asignará a la máquina.

Una vez creada la máquina, ésta aparecerá en el menú de la izquierda de VirtualBox, seleccionarla y pinchar el icono "Iniciar":



La máquina virtual se iniciará y aparecerá este menú, en el que debemos seleccionar el fichero "xxxxxx.iso" que hayamos descargado con el sistema operativo Ubuntu Server:

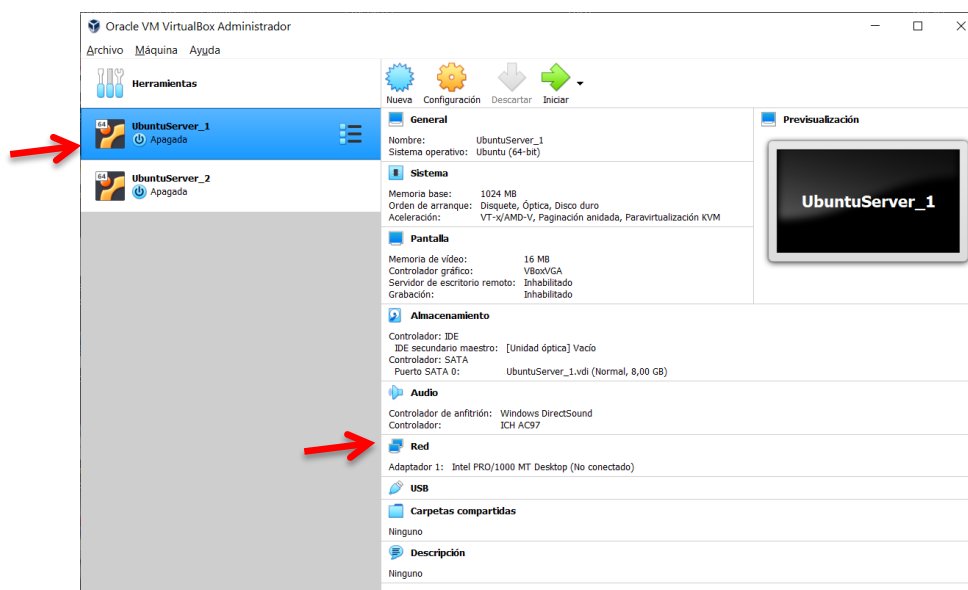


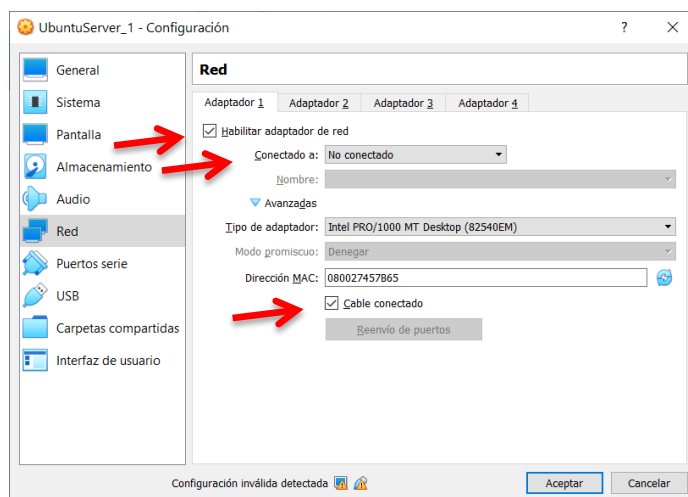
Una vez seleccionado se comienza la instalación de Ubuntu, se puede elegir el idioma español, tanto para el menú de instalación como para el sistema operativo, seleccionar el teclado en español y después instalar con todas las opciones por defecto.

En algún momento se pedirá Usar o no proxy. Nosotros no lo usaremos así que se dejará en blanco.

Al completar la instalación se reiniciará automáticamente la máquina virtual y puede que de un error: [FAILED] Failed unmounting /lib/modules. Presionar ENTER y continuar.

Importante. Una vez instalado el sistema operativo, y para que funcione correctamente todo lo relacionado con redes hemos de configurar el menú de red de la máquina virtual Ubuntu, pinchar en red y seleccionar en la pestaña "Adaptador 1", **No conectado** en el desplegable "Conectado a" y marcar "Cable conectado".





IV Instalación de GNS3



GNS3 (Graphical Network Simulator) es un emulador de redes que permite simular redes complejas. Permite combinar equipos virtuales y reales, emular software de cisco y analizar el tráfico que circula a través de los enlaces utilizando WireShark. Es software libre que puede descargarse, **previo registro**, de la página web:

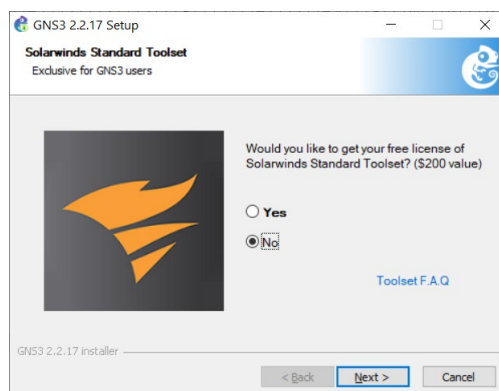
<https://www.gns3.com/>

Descargar la versión **2.2.17 (all in one) para Windows**.

En el siguiente enlace se describe el proceso de descarga e instalación de GNS3:

<https://docs.gns3.com/docs/>

Hay que seguir todos los pasos que se describen en el enlace y dejar todas las opciones por defecto, excepto que no es necesario instalar la aplicación "Solarwinds Response Time Viewer", por lo que se podría desmarcar de las aplicaciones a instalar.

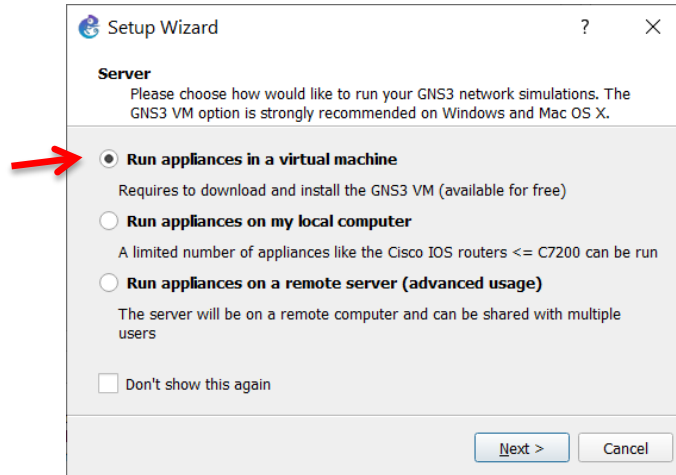


Al arrancar GNS3 por primera vez pregunta por el servidor en el que ejecutar las simulaciones de red, podemos seleccionar **Local GNS3 VM** o **Local**:

- **Run appliances in a virtual machine.** Significa que ciertas partes de la aplicación GNS3 se van a ejecutar en una máquina virtual (GNS3 VM.ova) que debe ser instalada en nuestro equipo y se explicará en el siguiente apartado.
- **Run appliances on my local computer.** Significa que toda la aplicación GNS3 se va a ejecutar directamente en nuestro equipo. En este caso no se debe instalar nada más.



- **Run appliances on remote server.** Significa que ciertas partes de la aplicación GNS3 se van a ejecutar en un servidor remoto, quedando la aplicación local de nuestro equipo simplemente como un cliente. El servidor debe ser instalado y configurado previamente.



Los desarrolladores de GNS3 recomiendan usar la primera opción, porque usa menos recursos de nuestro equipo y además se “toca” menos las configuraciones de las redes de nuestro equipo, evitando conflictos con otras aplicaciones. Esta es la opción que vamos a usar y la explicaremos en el punto siguiente.

Por otro lado, los desarrolladores de GNS3 también recomiendan usar VMware como software de Virtualización. **Por tanto, para poder continuar con la configuración de GNS3 (pantalla anterior) es necesario previamente instalar el VMware y descargar la máquina virtual de GNS3 para VMware.**



V Instalación de VMware y GNS3 VM

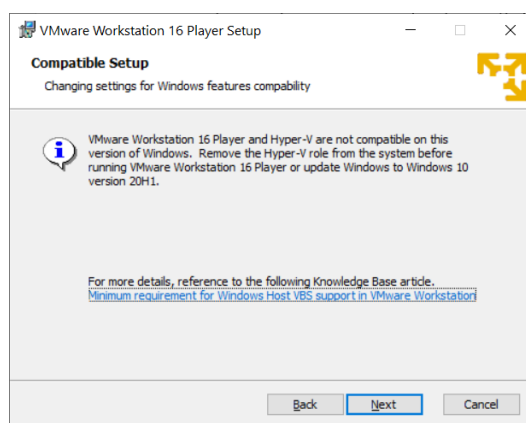
1 Instalación de VMware Workstation Player

VMware Workstation Player 16.1.0, puede descargarse del siguiente enlace:

<https://www.vmware.com/es/products/workstation-player/workstation-player-evaluation.html>

Aceptar en las diferentes ventanas de la instalación todas las opciones por defecto.

Nota 1. En ciertas versiones de Windows 10, Workstation Player 16 es incompatible con el software de virtualización Hyper V o con las credenciales de seguridad. Si esto ocurre os saldrá un mensaje similar al siguiente en la instalación.



A pesar de ello, se puede continuar con la instalación pero más adelante cuando se intente ejecutar la máquina virtual de GNS3 directamente desde Workstation Player dará un error.

El origen puede ser diferentes incompatibilidades y cada una de ellas tiene su solución, por lo que es posible que en algunos equipos funcione implementando solo una solución y en otros haya que hacer más pasos:

- Deshabilitar el aislamiento de núcleo en Windows Defender.
- Deshabilitar Hyper-V.
- Deshabilitar la guardia de credenciales desde las directivas de grupo de Windows 10.

En el siguiente enlace hay una pequeña guía para resolver el problema:

<https://www.softzone.es/2019/06/18/solucion-problema-vmware-device-credential-guard-windows-10/>

Y los siguientes son los enlaces oficiales de VMWare con soluciones al respecto:

<https://kb.vmware.com/s/article/76918>

https://kb.vmware.com/s/article/2146361?lang=en_US



2 Descarga e instalación de la máquina virtual GNS3 para VMware

Desde la página de GNS3 se puede descargar la máquina virtual de GNS3 "GNS3 VM.ova" que se ejecutará en VMware:

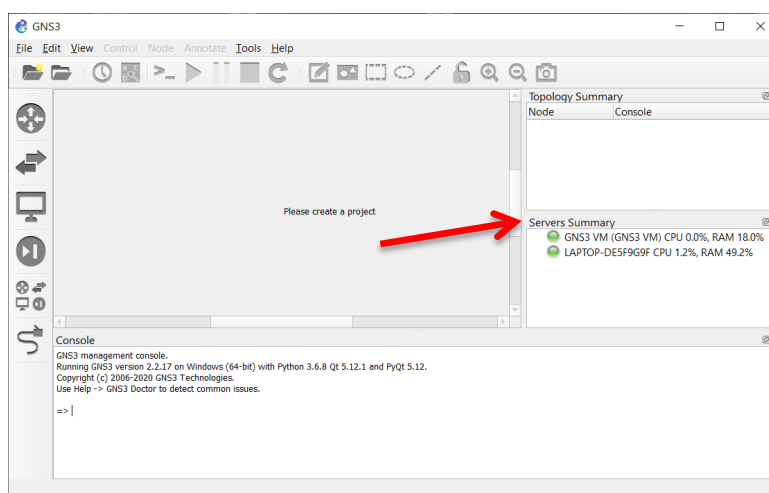
<https://gns3.com/software/download-vm>

Hay que recalcar que para cada versión de GNS3 existe una máquina virtual, es decir, en nuestro caso la versión de GNS3 es 2.2.17 y la de la máquina virtual de GNS3 debe ser la misma.

A partir de aquí, en el siguiente enlace puede seguirse el proceso de instalación de la máquina virtual de GNS3 en VMware:

<https://docs.gns3.com/docs/getting-started/setup-wizard-gns3-vm>

Si todo ha ido bien al arrancar GNS3 se debe arrancar a los pocos segundos y de manera automática la máquina virtual de GNS3 y en la aplicación debe aparecer en verde que está arrancada la máquina virtual según la figura siguiente.



3 Adición de máquinas virtuales en GNS3

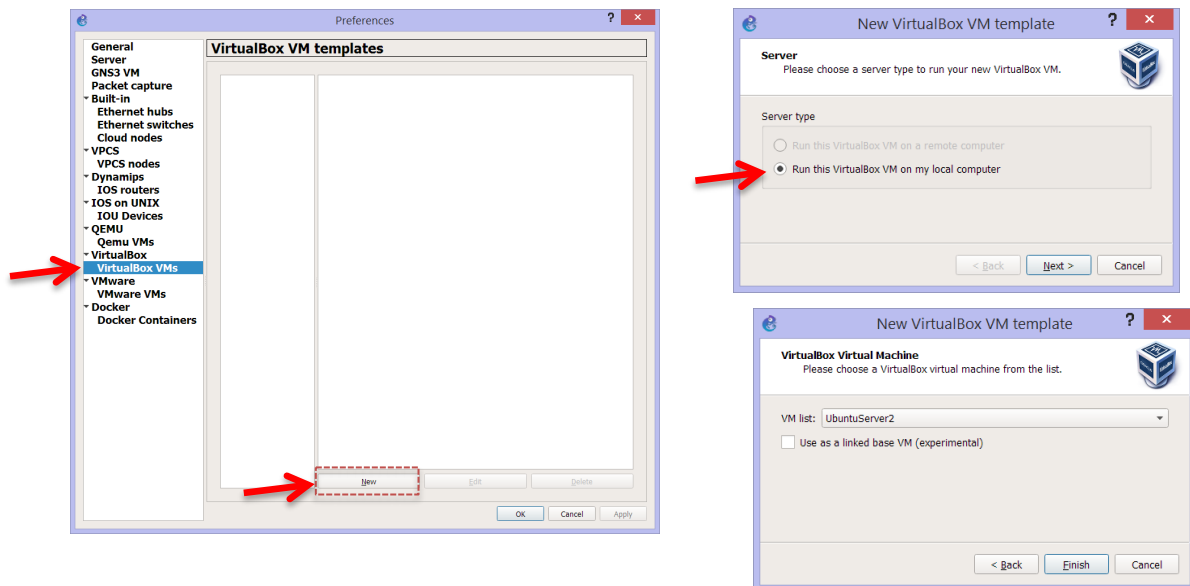
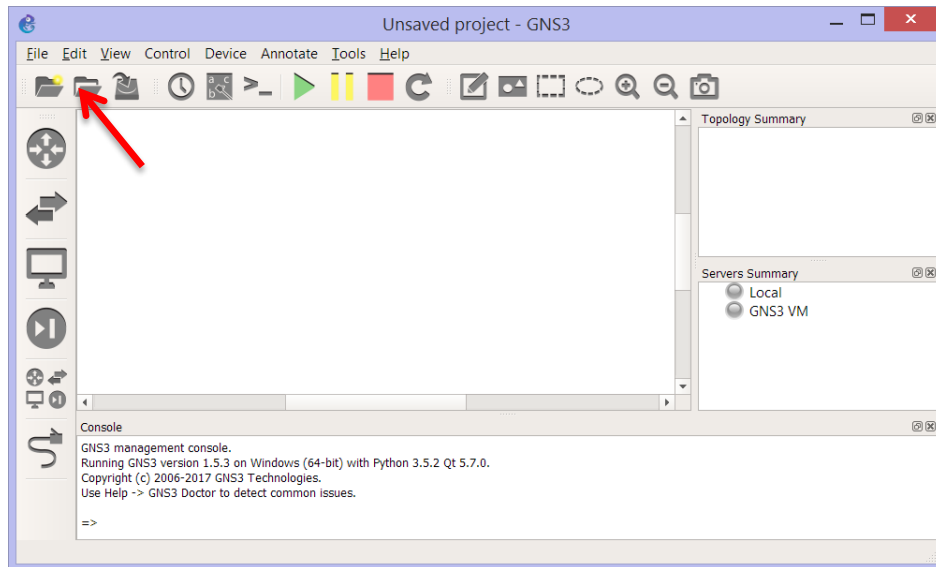
GNS3 tiene la opción de emular sistemas terminales de distintas formas, en nuestro caso vamos a trabajar con dos de esas alternativas. Utilizando simuladores de PC y utilizando máquinas virtuales:

- Los simuladores de PC son llamados **Virtual PC Simulators (VPCS)** y se utilizarán cuando no se necesite una configuración compleja del sistema terminal y simplemente se necesite un equipo que posea una determinada dirección.
- **Las máquinas virtuales** se utilizarán cuando se necesite probar acciones más complejas en los sistemas terminales y utilizaremos las máquinas virtuales **Ubuntu** creadas anteriormente.

Los VPCS aparecen ya por defecto en la interfaz.

Pero las máquinas virtuales será necesario añadirlas. Para añadir una máquina virtual en GNS3, seleccionamos Edit -> Preferences y en el desplegable añadimos la máquina virtual que queramos de las que se hayan creado anteriormente.





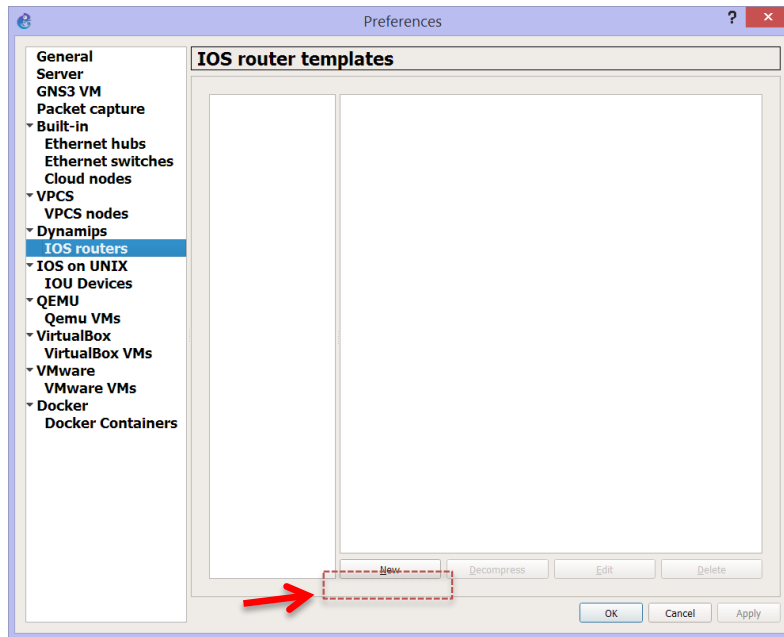
4 Adición de Cisco IOS en GNS3

En cualquier momento se pueden añadir nuevos routers de manera manual. GNS3 permite también virtualizar los IOS de los dispositivos Cisco. En nuestro caso se va a añadir un router y un switch genérico para hacer las prácticas. La imagen de este router/switch está disponible para la descarga en UBUVirtual.

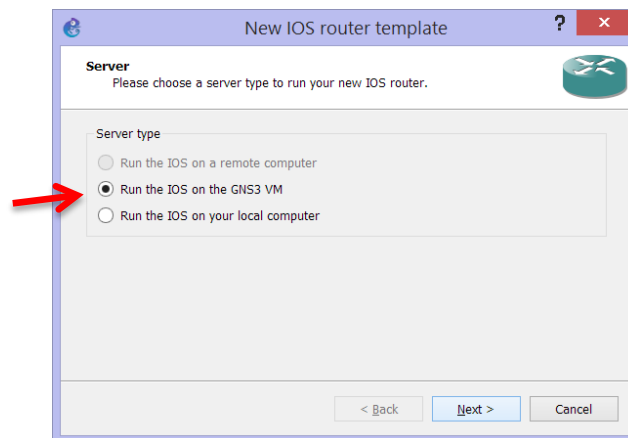
4.1 Cisco IOS para un Router

Para añadir un IOS para un router en GNS3, seleccionamos Edit -> Preferences y en la ventana siguiente seleccionar "IOS routers" y pinchar en "New".



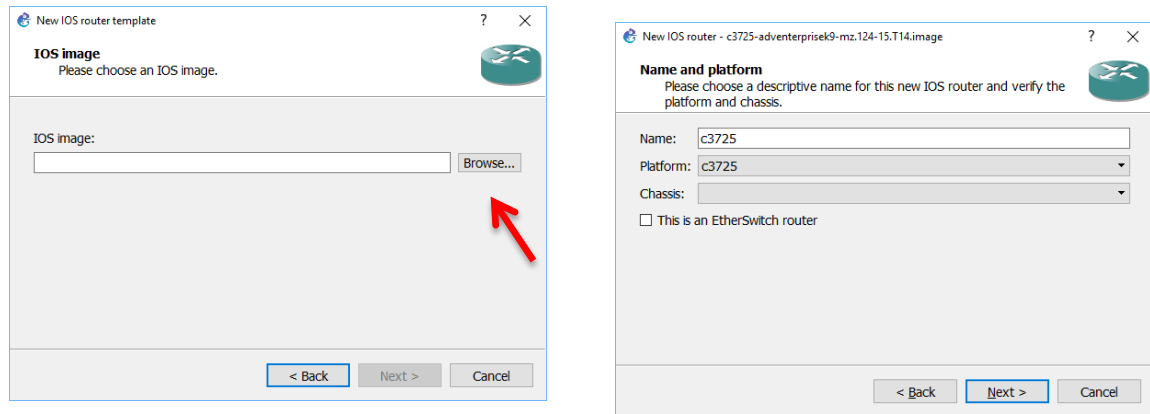


Seleccionar que el IOS se va a ejecutar en la máquina virtual GNS3 que ya instalamos anteriormente:

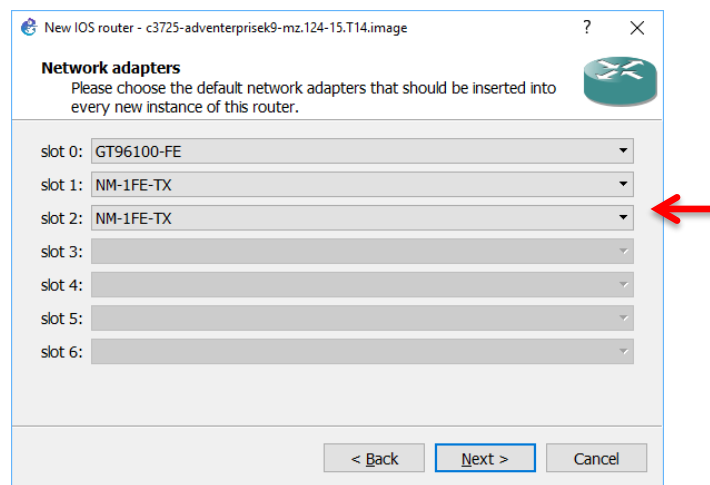


Seleccionar el fichero que contiene el sistema operativo de Cisco y darle el nombre por defecto:

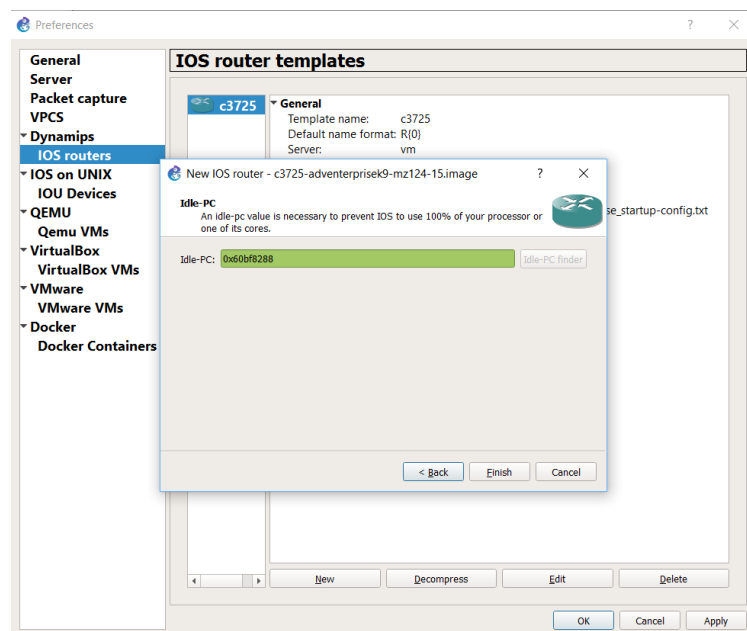




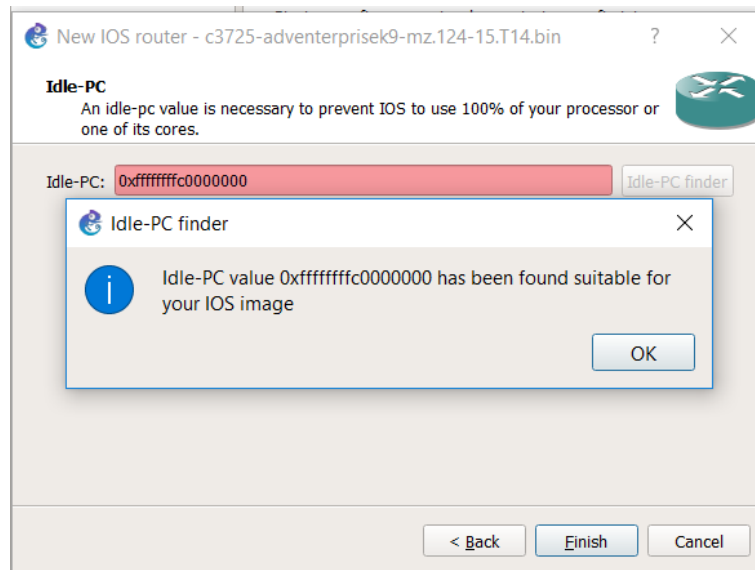
Elegir todas las opciones por defecto hasta la siguiente pantalla y elegir:



Una opción interesante para limitar el uso que hacen de la CPU las imágenes de routers y switches, es el uso de un Idle PC.

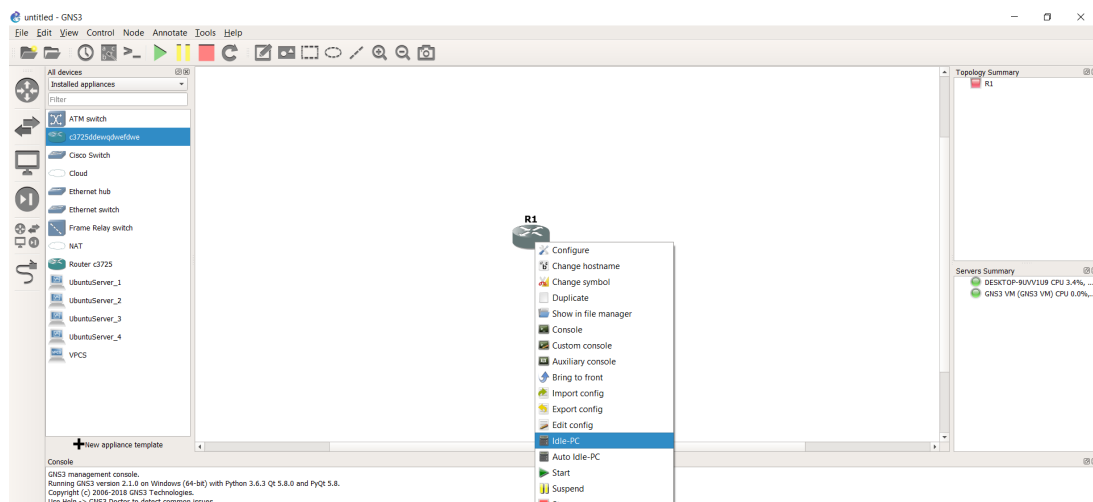


Si en la búsqueda del Idle-PC aparece el siguiente error:

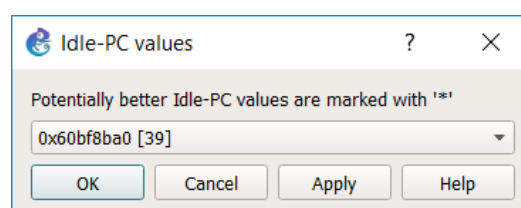


Configurar el equipo renunciando a la búsqueda del Idle-PC (dejando en blanco la caja de texto correspondiente a Idle PC) y seguir los siguientes pasos.

A continuación, en un proyecto de GNS3, utilizar el componente que se haya añadido y pinchando en el componente con el botón derecho seleccionar Idle-PC.



Después de un tiempo aparecerá una opción, la aplicamos y con esto quedará asignado este valor.



4.2 Cisco IOS para un Switch

A continuación, vamos a añadir un switch. Como switch utilizaremos el mismo IOS que en el caso anterior ya que pertenece a un router con capacidades de switch.

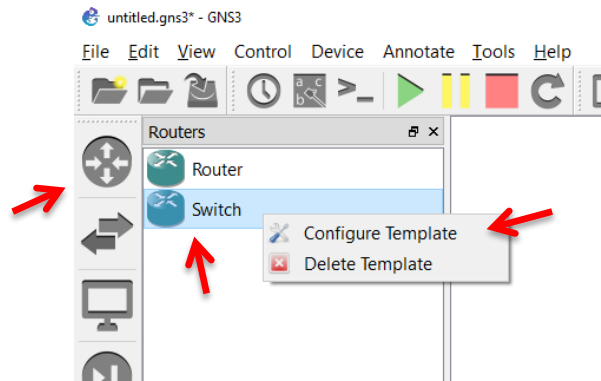
Seguiremos los pasos descritos con anterioridad para el caso del router. Será necesario en este caso cambiar el nombre, podemos poner:

Es necesario también escoger el módulo **NM-16ESW**, que nos proporcionará 16 puertos Ethernet:

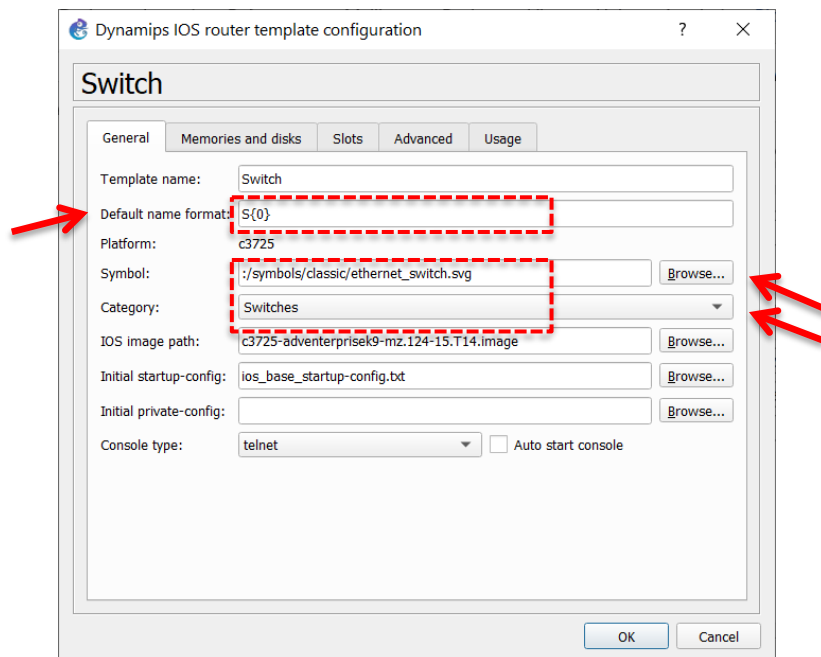
El resto de las opciones se seleccionan por defecto.

Posteriormente, para no confundir el router y el switch, se editará el template del switch, primero pinchamos en "Browse Routers" y después pinchamos con el botón derecho encima del símbolo del switch que acabamos de crear y seleccionamos: Configure Template.



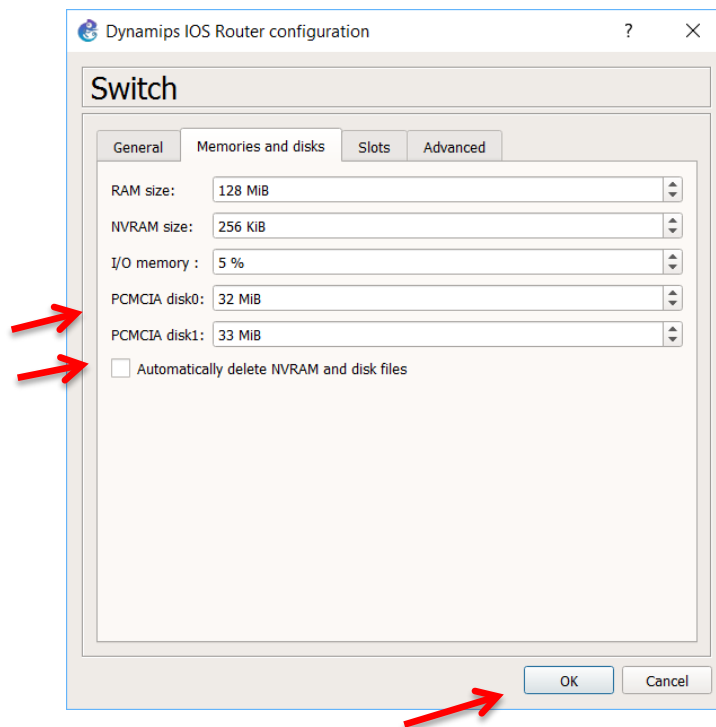


En el cuadro que aparece a continuación cambiamos el nombre por defecto S{0}, el símbolo (Ethernet Switch) y la categoría (Switches).



Además, en la pestaña "Memories and disks" se debe aumentar el tamaño de memoria en el disco tarjeta **PCMCIA a 32 MB**, se debe además **desmarcar la casilla "Automatically delete NVRAM and disk files"** y pinchar en OK para terminar:





Con esto ya podemos simular una red con GNS3!!!



VI Instalación de WireShark



WireShark es un analizador de protocolos (sniffer) utilizado para realizar análisis y solucionar problemas en redes de comunicaciones, para desarrollo de software y protocolos, y como una herramienta didáctica.

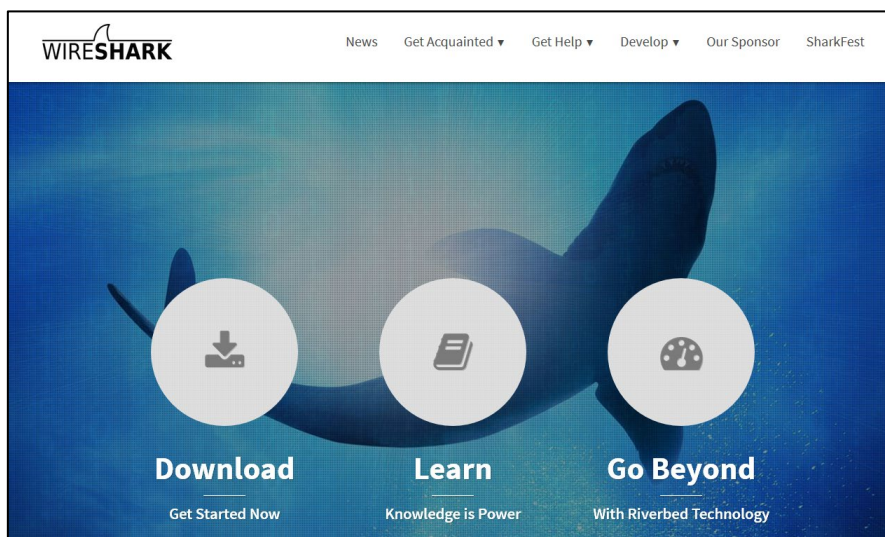
Permite examinar datos de una red viva o de un archivo de captura salvado en disco. Se puede analizar la información capturada, a través de los detalles y sumarios por cada paquete. WireShark incluye un completo lenguaje para filtrar lo que queremos ver

WireShark es software libre, y se ejecuta sobre la mayoría de los sistemas operativos Unix, Linux, así como en Microsoft Windows.

En principio **se instala automáticamente al instalar GNS3**, ver apartado anterior, pero si no, se puede descargar en la web:

<https://www.wireshark.org/>

Seleccionar download y después **la versión que haya más reciente (3.4.2)** y elegir el instalador adecuado para el sistema operativo que se tenga: Windows Installer (64 bits), Windows Installer (32 bits), etc.

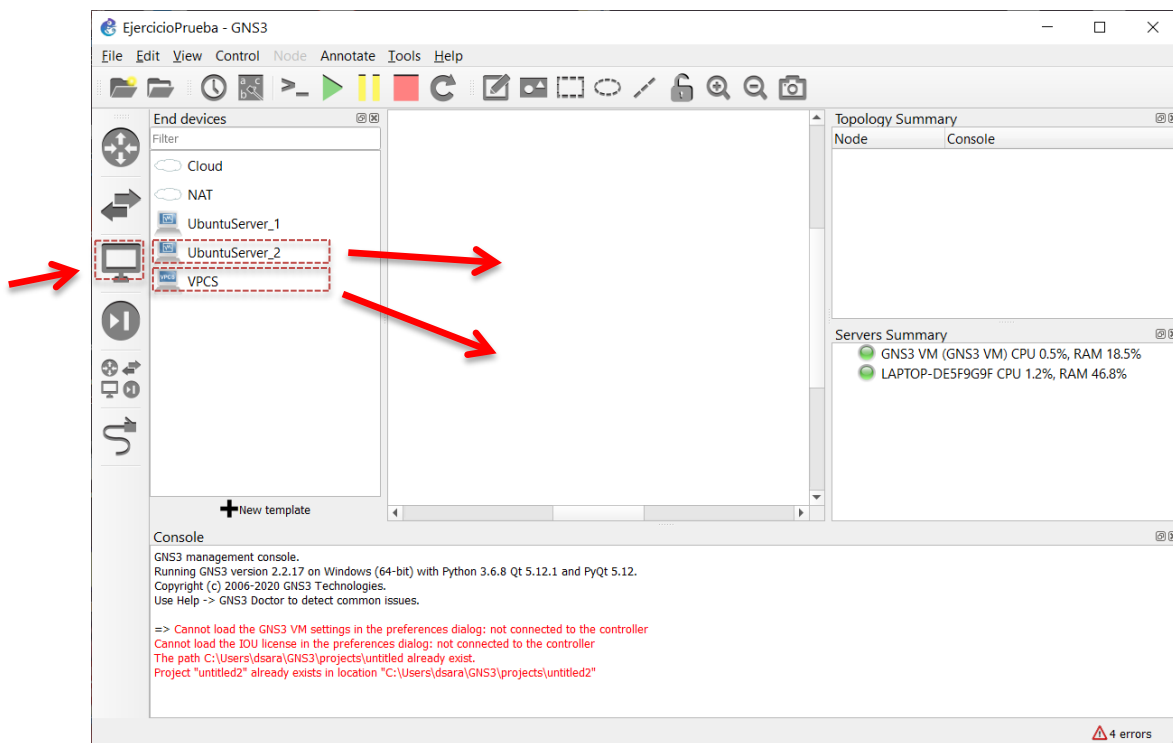


VII Ejercicio de prueba

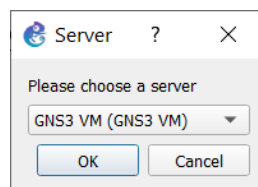
Para probar que la instalación se ha realizado correctamente se recomienda realizar la siguiente prueba.

1. Arrancar GNS3. Al arrancar GNS3 debe lanzarse la máquina virtual de VMware, tarda un poco en arrancar.
2. Crear un proyecto nuevo.

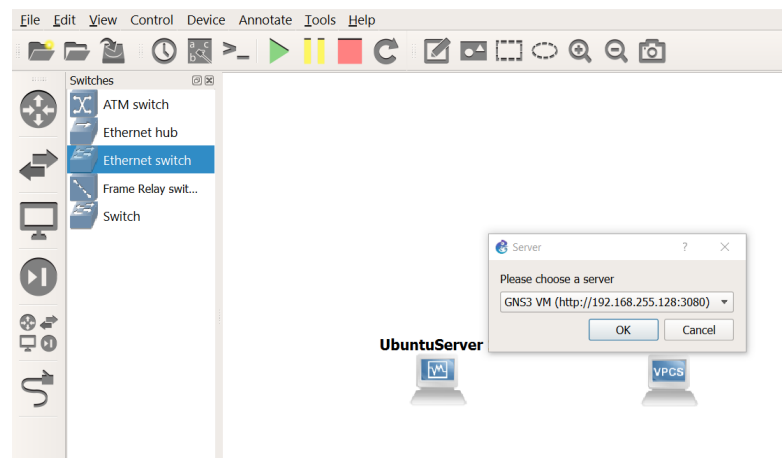
Importante: NO usar espacios en blanco o caracteres "raros" (ñ, tildes, etc.). Usar solo caracteres alfanuméricos y guion bajo.
3. Dentro de los dispositivos terminales arrastrar al área de trabajo una máquina UbuntuServer_2 y un Virtual PC.



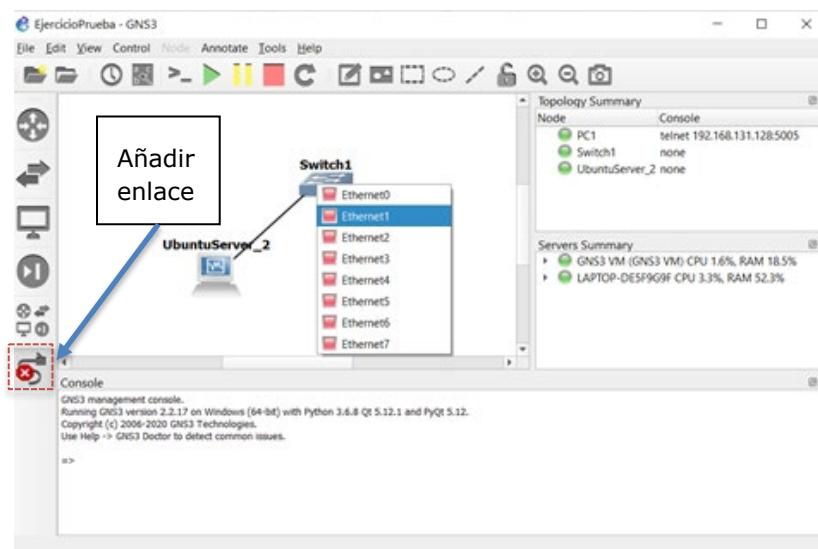
Al añadir el VPC os saldrá el siguiente diálogo, seleccionar que el VPC se ejecute en la máquina virtual de GNS3:



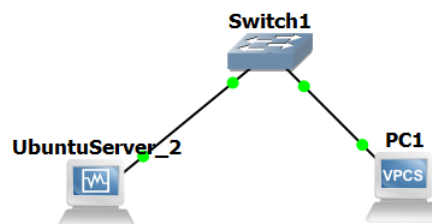
- Añadir también un Ethernet Switch (debe ser el que proporciona GNS3 **no el de Cisco**) y seleccionar que se ejecute en el servidor de GNS3 VM.



- Una vez arrastrados los componentes seleccionar el elemento conector y unir los terminales:



- Una vez realizada la conexión pulsar Play, deberá arrancar la máquina virtual Ubuntu y los enlaces aparecerán marcados en verde. El usuario de la máquina Ubuntu es *redes* y el password es *redes*.



- Configurar la dirección IP de la máquina Ubuntu escribiendo:

```
sudo ifconfig eth0 192.168.0.2 netmask 255.255.255.0
```



Puedes comprobar que la configuración se ha realizado correctamente escribiendo a continuación:

ifconfig

```

redes@redes:~$ sudo ifconfig eth0 192.168.0.2 netmask 255.255.255.0
redes@redes:~$ ifconfig
eth0      Link encap:Ethernet direcciónHW 08:00:27:c6:eb:7d
         Direc. inet:192.168.0.2 Difus.:192.168.0.255 Másc:255.255.255.0
         Dirección inet6: fe80::a00:27ff:fec6:eb7d/64 Alcance:Enlace
         ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST MTU:1500 Métrica:1
         Paquetes RX:6 errores:0 perdidos:0 overruns:0 frame:0
         Paquetes TX:11 errores:0 perdidos:0 overruns:0 carrier:0
         colisiones:0 long.colaTX:1000
         Bytes RX:384 (384.0 B) TX bytes:828 (828.0 B)

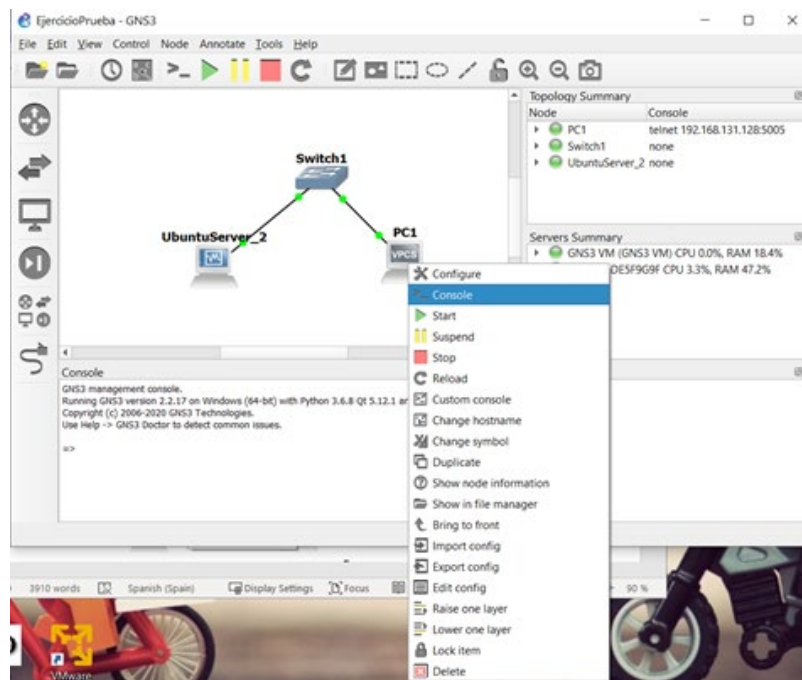
lo        Link encap:Bucle local
         Direc. inet:127.0.0.1 Másc:255.0.0.0
         Dirección inet6: ::1/128 Alcance:Amfitrión
         ACTIVO BUCLE FUNCIONANDO MTU:65536 Métrica:1
         Paquetes RX:30 errores:0 perdidos:0 overruns:0 frame:0
         Paquetes TX:30 errores:0 perdidos:0 overruns:0 carrier:0
         colisiones:0 long.colaTX:0
         Bytes RX:2248 (2.2 KB) TX bytes:2248 (2.2 KB)

redes@redes:~$

```

8. Configurar el VPC, para ello debes abrir la consola y escribir en ella:

ip 192.168.0.1/24



```

PC1 - PuTTY
Welcome to Virtual PC Simulator, version 0.8.1
Dedicated to Daling.
Build time: Apr 10 2019 16:35:00
Copyright (c) 2007-2015, Paul Meng (mirnshi@gmail.com)
All rights reserved.

VPCS is free software, distributed under the terms of the "BSD" licence.
Source code and license can be found at vpcs.sf.net.
For more information, please visit wiki.freecode.com.cn.

Press '?' to get help.

Executing the startup file

PC1> ip 192.168.0.1/24
Checking for duplicate address...
PC1 : 192.168.0.1 255.255.255.0

PC1>

```



9. Finalmente prueba a hacer ping entre los equipos. Deberían hacer ping sin problema.

```
PC1> ping 192.168.0.2

84 bytes from 192.168.0.2 icmp_seq=1 ttl=64 time=1.009 ms
84 bytes from 192.168.0.2 icmp_seq=2 ttl=64 time=2.099 ms
84 bytes from 192.168.0.2 icmp_seq=3 ttl=64 time=0.852 ms
84 bytes from 192.168.0.2 icmp_seq=4 ttl=64 time=0.699 ms
84 bytes from 192.168.0.2 icmp_seq=5 ttl=64 time=0.905 ms

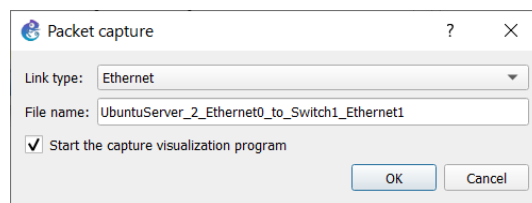
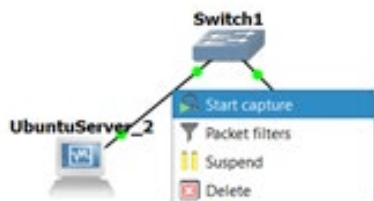
PC1>
```

```
redes@redes:~$ ping 192.168.0.1
PING 192.168.0.1 (192.168.0.1) 56(84) bytes of data:
64 bytes from 192.168.0.1: icmp_seq=1 ttl=64 time=0.903 ms
64 bytes from 192.168.0.1: icmp_seq=2 ttl=64 time=1.53 ms
64 bytes from 192.168.0.1: icmp_seq=3 ttl=64 time=1.01 ms
64 bytes from 192.168.0.1: icmp_seq=4 ttl=64 time=1.22 ms
64 bytes from 192.168.0.1: icmp_seq=5 ttl=64 time=0.886 ms

--- 192.168.0.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4007ms
rtt min/avg/max/mdev = 0.886/1.113/1.536/0.246 ms
redes@redes:~$ _
```

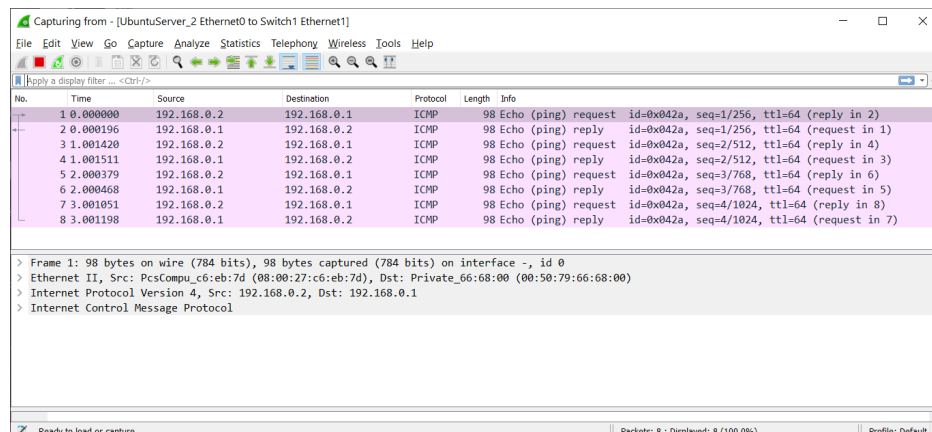
En ocasiones puede ocurrir que una vez instalada la máquina virtual de GNS3, al tratar de hacer ping entre equipos de una red que se haya creado, no sea posible conseguirlo a pesar de que todas las configuraciones sean correctas. Si esto ocurre, comprueba la dirección que aparece en Host Binding de GNS3, tal como se indica más arriba en Sección V2 Descarga e instalación de la máquina virtual GNS3 para VMware.

10. Pincha con el botón derecho del ratón en el enlace de conexión entre los dos equipos y selecciona "Start Capture", se arrancará automáticamente WireShark y aparecerá una lupa en el enlace seleccionado.



11. Desde la máquina Ubuntu hacer ping al VPC (ping 192.168.0.1).

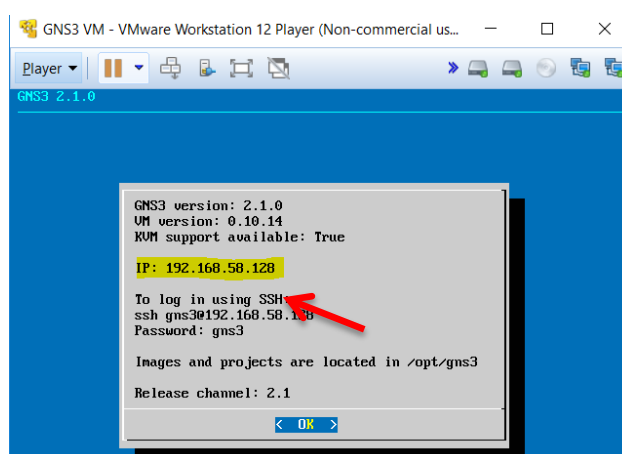
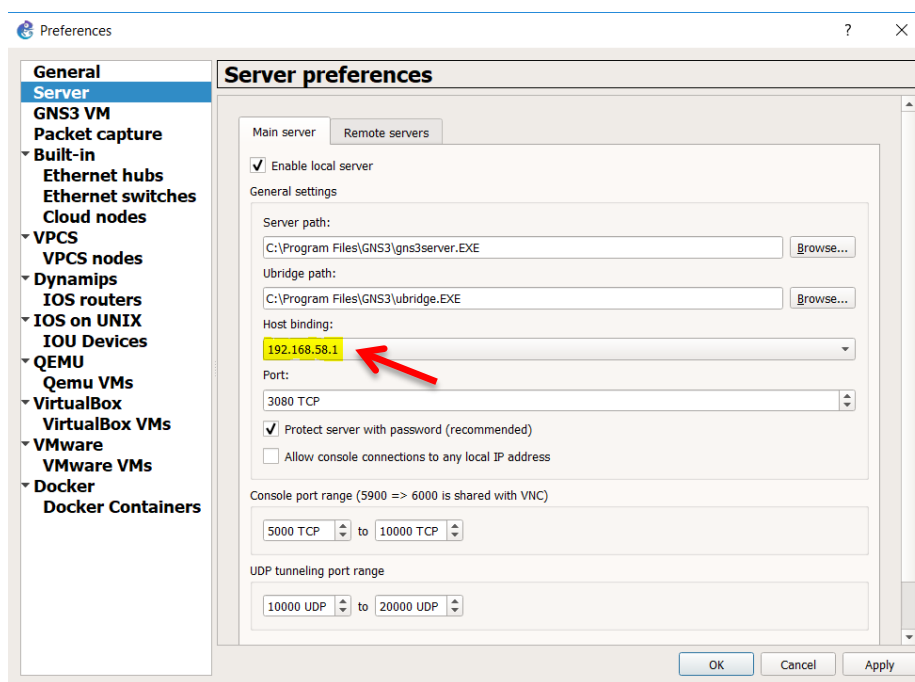
12. En WireShark deberemos ver las tramas correspondientes a ARP e ICMP.



VIII Subsanación de problemas de instalación

1 Máquina Virtual de GNS3

Problema 1. En ocasiones la dirección de la máquina virtual no pertenece a la subred de la dirección Host Binding en GNS3. Hay que tener cuidado con esto, ya que esto ocasiona que las máquinas que se ejecutan en modo local y las que se ejecutan la máquina virtual de GNS3 no se comuniquen entre ellas. Si no se puede encontrar una IP en la misma subred o hubiera algún problema en la conexión seleccionar "localhost" para "Host binding".



Una vez hecho esto, al arrancar GNS3, debe arrancar automáticamente la máquina virtual de GNS3.

Problema 2. Un problema que ocurre con el uso de la máquina virtual de GNS3 es que es necesario que habilitar en la Bios del equipo la tecnología de virtualización de Intel. Depende de la bios del equipo, la ubicación y el nombre de esta función puede variar. Si buscáis en la web:



Intel Virtualization Technology bios <nombre de la marca del equipo>

Es posible que encontréis un enlace que os explique cómo hacerlo.

Por ejemplo:

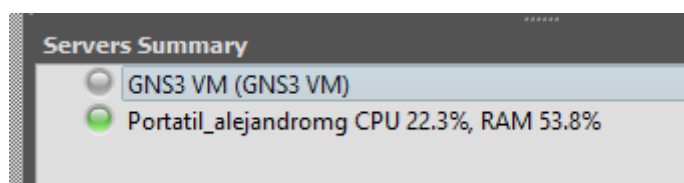
Intel Virtualization Technology bios Asus

Intel Virtualization Technology bios Lenovo

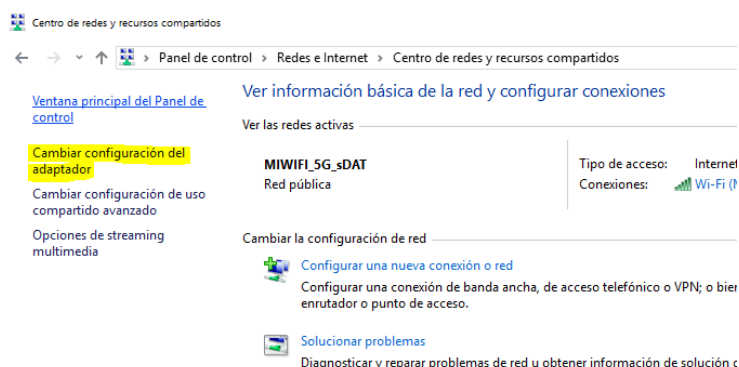
...

Una vez hecho esto, al arrancar GNS3, debe arrancar la máquina virtual de GNS3.

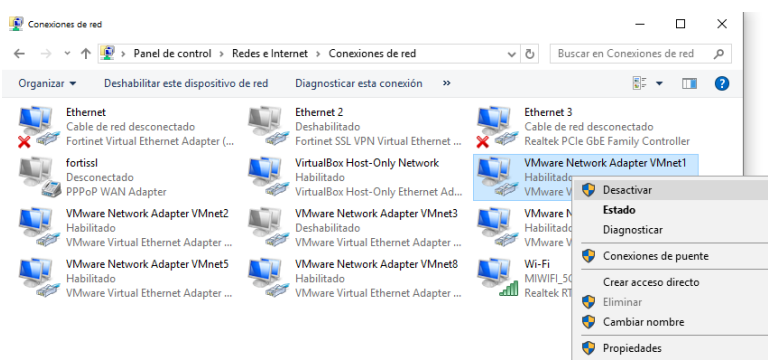
Problema 3. En las últimas versiones de Windows 10, es posible que la máquina Virtual de GNS3 arranque, pero la conexión con ella no sea posible, apareciendo el icono de la máquina virtual de GNS3 en gris.



Para solucionar este problema es necesario ir a Panel de Control->Redes e Internet->Centro de Redes y Recursos Compartidos, y pinchar en "Cambiar configuración del adaptador"



En ese momento podréis acceder a una pantalla con los adaptadores de red disponibles en vuestro equipo. Hay que seleccionar los adaptadores de red de VMWare (VMWare Network Adapter X), hacer clic con el botón derecho y Desactivar, esperar 3 o 4 segundos y activar de nuevo.



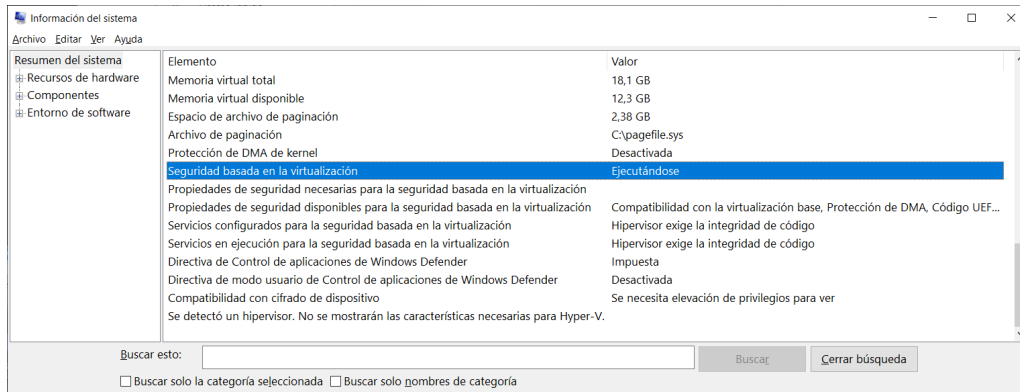
En principio es suficiente con hacerlo con el VMWare Network Adapter 1, si esto no funcionara o el número del adaptador de red asignado fuera distinto, hacer lo mismo con el resto de adaptadores.



2 Instalación WMware

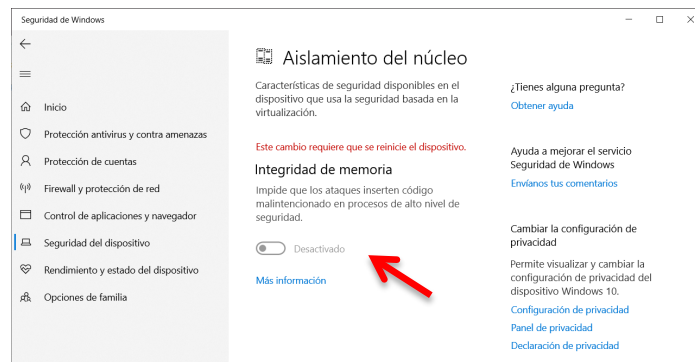
En principio en "Información del Sistema" la seguridad basada en virtualización debe estar "No habilitada", si es así probablemente el problema ya esté solucionado.

Si aparece "Ejecutándose" como en la figura siguiente, hay que seguir los pasos siguientes.



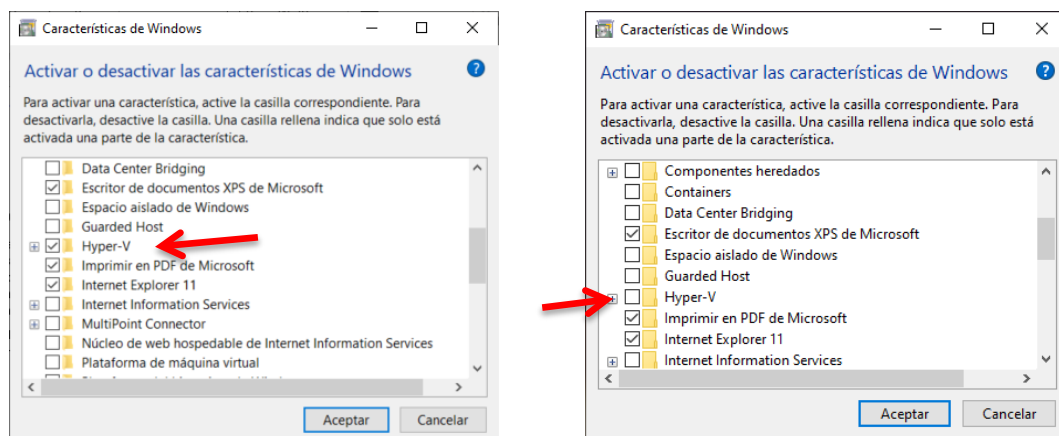
Deshabilitar el aislamiento de núcleo en Windows Defender

En "Seguridad de Windows" desactivar la "Integridad de memoria".



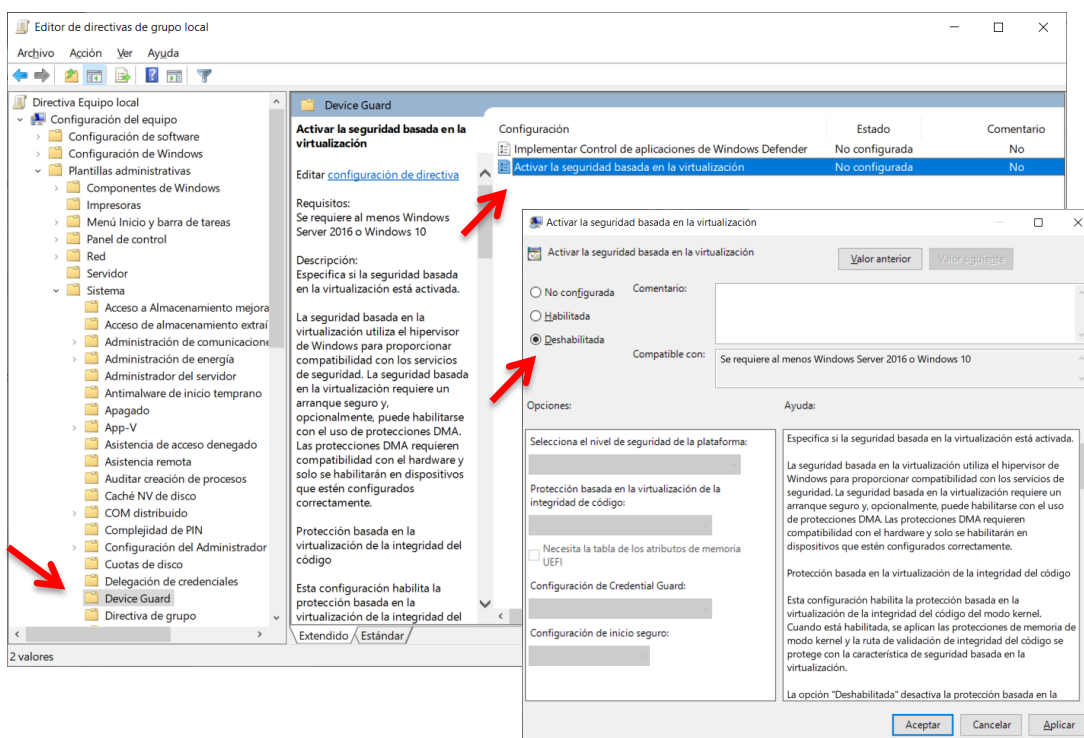
Deshabilitar Hyper-V

Básicamente es comprobar en "Características de Windows" si el Hyper-V está activado en nuestro sistema operativo, si lo está, hay que desactivarlo.



Deshabilitar la guardia de credenciales desde las directivas de grupo de Windows 10

Ejecutar el "Editor de directivas de grupo local" y en "Device Guard" seleccionar "Implementar Control..." y "Activar la seguridad ..." y marcar la opción Deshabilitada.

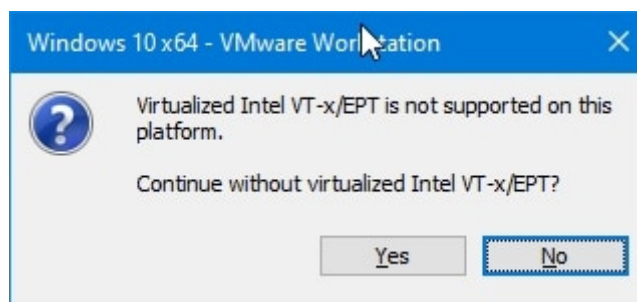


Probablemente sea necesario reiniciar el equipo después de cada paso anterior.

Finalmente volver a comprobar en "Información del Sistema" que la seguridad basada en virtualización esté "No habilitada" y si es así el problema estará solucionado. Ahora no dará ningún error al arrancar la máquina virtual de GNS3 desde Workstation Player.

3 Problemas con la virtualización en GNS3

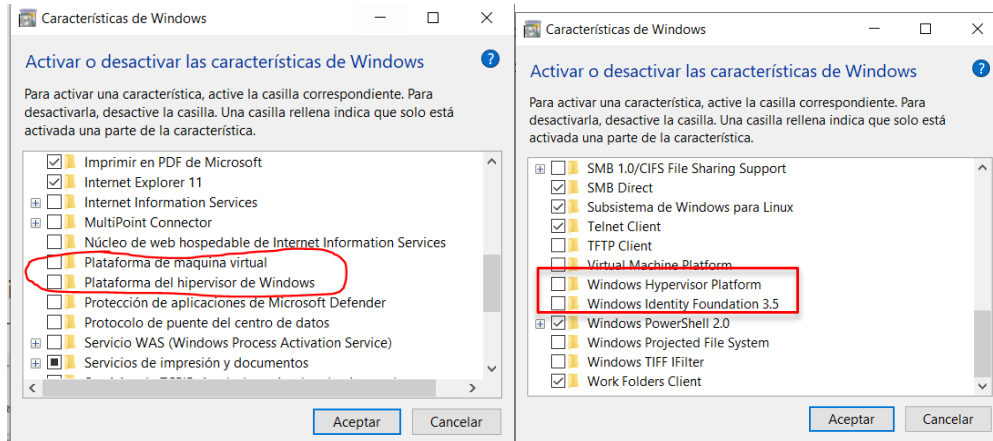
En algunos casos, al arrancar la máquina virtual de GNS3 en VMWare aparece el siguiente mensaje:



Para solucionar este error, es necesario deshabilitar las características de Windows:

- Plataforma de máquina virtual
- Plataforma del hipervisor de Windows.





4 Problemas con Wireshark

A veces cuando se quiere hacer una captura con Wireshark dentro de GNS3 aparece este mensaje de error: "End of file on pipe magic during open".

Es debido a que el proyecto de GNS3 se ha creado con un nombre que contiene algún carácter "raro" (tildes, espacios en blanco, ñ, etc.) y Wireshark no lo reconoce. Se debe cambiar el nombre del proyecto de GNS3 para que no salga este error.





Grado en Ingeniería Informática

REDES

PRÁCTICA 5A

Arquitectura de red

Docentes:

Alejandro Merino

Daniel Sarabia Ortiz

*Dpto. de Ingeniería Electromecánica
Área de Ingeniería de Sistemas y Automática*

Versión 1.1

Fecha 03/02/2022 15:01

Esta obra está sujeta a la licencia Reconocimiento 4.0 Internacional de Creative Commons. Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by/4.0/>



Índice de contenidos

I	INTRODUCCIÓN.....	3
II	OBJETIVOS	3
III	CONFIGURACIÓN DE RED EN WINDOWS	3
IV	COMANDOS DE DIAGNÓSTICO DE REDES	5
	1 Ping.....	5
	2 Traceroute o Tracert.....	7
	3 Comunicación PC y móvil.....	7
	4 Ejercicios.....	7
V	PROTOCOLOS Y WIRESHARK	8
VI	BIBLIOGRAFÍA	9



I Introducción

En esta práctica se pretende iniciarse en el concepto de arquitectura de redes empezando en sistemas Windows y conociendo como se configura un adaptador de red. También se mostrarán dos herramientas de diagnóstico de redes ping y tracert/traceroute que usaremos a lo largo de la asignatura para determinar si hay comunicación entre dos equipos al menos a nivel de la capa de red. Finalmente se mostrará el software WireShark que permite analizar los paquetes que se envían y reciben por un adaptador de red, pudiendo usar esta información para conocer características de la comunicación y protocolos usados en ella. También será una herramienta que usaremos a lo largo de todas las prácticas.

II Objetivos

- Aprender a configurar el adaptador de red en un sistema operativo Windows.
- Primeras nociones de direcciones IP y direcciones MAC.
- Uso de comandos básicos para diagnóstico de redes: Ping y tracert/traceroute.
- Introducir el uso de analizadores de paquetes WireShark.

III Configuración de red en Windows

La configuración de red en los sistemas operativos Windows se realiza en el Centro de redes y recursos compartidos que se accede desde el Panel de control y en Windows 10 en Red e Internet la configuración de Windows.

En esta ventana podemos ver las redes activas a las que se está conectado y también el tipo de acceso a dicha red. En el ejemplo se ve que se está conectado a la red de la ubu, que dicha red da acceso a Internet y que la conexión es a través de la red eduroam de tipo Wi-Fi (es decir se está usando la tarjeta de red Wi-Fi).

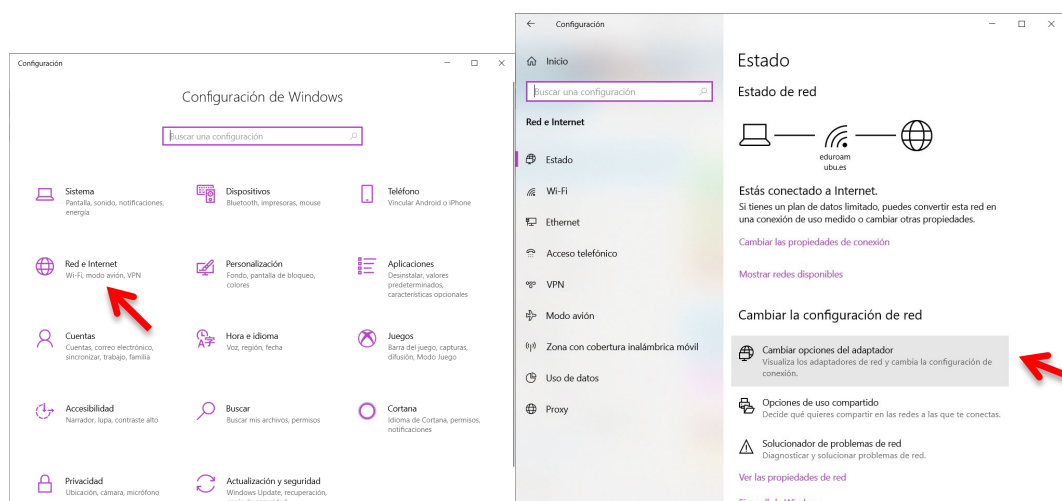


Figura 1. Centro de redes y recursos compartidos.

Pinchando en Cambiar configuración del adaptador podemos ver los adaptadores de red que disponemos para conectar el equipo. En el ejemplo se dispone de una tarjeta de red



Ethernet (adaptador Realtek PCIe GBE Family Controller) para conexión mediante cable y una tarjeta de red Wi-Fi (adaptador de red Adaptador de red Broadcom 802.11n) para conexión inalámbrica. Ambos adaptadores son reales, es decir son tarjetas de red reales montadas en el equipo. También se indica que la tarjeta de red Ethernet no está conectada con el cable.

En el ejemplo aparecen tres tarjetas de red que son virtuales, no hay hardware real montado y que han sido creadas por los hipervisores instalados en el equipo, en este caso VirtualBox y VMware.

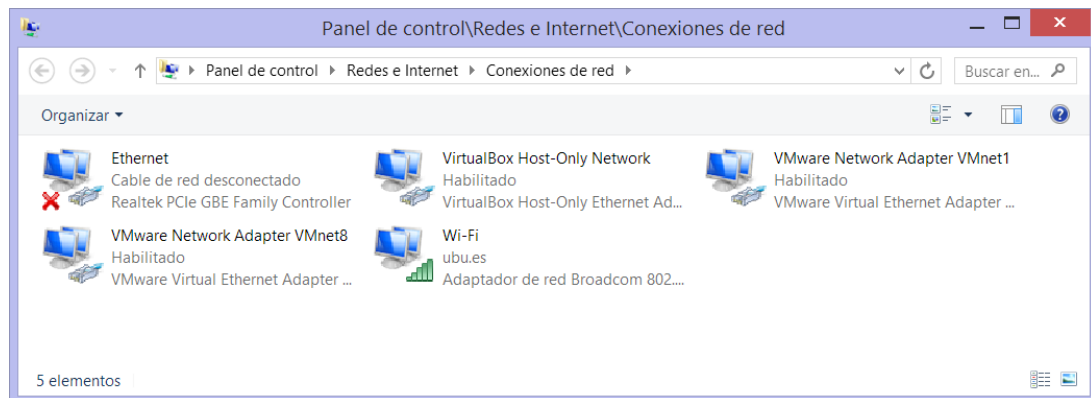


Figura 2. Conexión de redes.

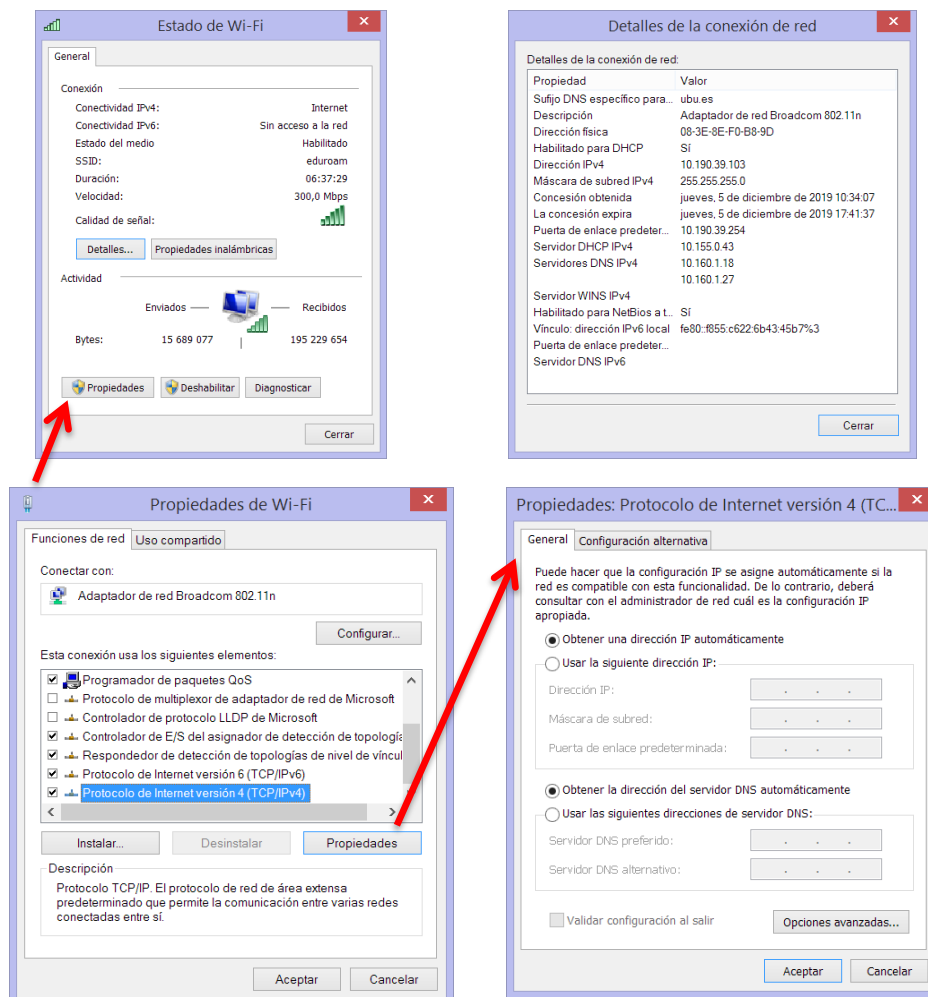


Figura 3. Propiedades del adaptador de red WIFI.



1. Comprobar en cada PC (en Windows) en el adaptador Ethernet o WIFI según el que se esté usando como está configurada la obtención de la IP.
2. En caso de ser dinámica
 - ¿Quién es el servidor DHCP?
 - ¿Qué IP, máscara de red y puerta de enlace ha suministrado el servidor DHCP?
3. Otra manera de ver la configuración de red es ejecutar la consola de comandos cmd y en ella usar el comando `ipconfig` o para obtener más información `ipconfig /all`:

C:\> ipconfig

```

Microsoft Windows [Versión 6.3.9600]
(c) 2013 Microsoft Corporation. Todos los derechos reservados.

C:\Users\autom>ipconfig

Configuración IP de Windows

Adaptador de LAN inalámbrica Conexión de área local* 3:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . . :

Adaptador de Ethernet Ethernet:

    Sufijo DNS específico para la conexión. . . : isa.cie.uva.es
    Vínculo: dirección IPv6 local. . . . . : fe80::81ed:5938:4de:f26b%4
    Dirección IPv4. . . . . : 192.168.1.128
    Máscara de subred. . . . . : 255.255.255.0
    Puerta de enlace predeterminada. . . . . : 192.168.1.1

Adaptador de LAN inalámbrica Wi-Fi:

    Sufijo DNS específico para la conexión. . . : ubu.es
    Vínculo: dirección IPv6 local. . . . . : fe80::f855:c622:6b43:45b7%3
  
```

IV Comandos de diagnóstico de redes

1 Ping

El objetivo de un ping es determinar si un host destino, identificado con una determinada IP, es accesible desde otro host, ver Figura 4. En esencia, Ping es un comando o una herramienta de diagnóstico que permite hacer una verificación del estado de una determinada conexión de un host local con al menos un equipo remoto contemplado en una red de tipo TCP/IP.

El funcionamiento del mecanismo es muy simple y puede ser de mucha ayuda. Se vale del envío de series de paquetes ICMP de solicitud (ICMP Echo Request) a la dirección IP de destino y recibir los correspondientes paquetes de respuesta (ICMP Echo Reply). Mediante esta utilidad puede diagnosticarse el estado, velocidad y calidad de una red determinada.

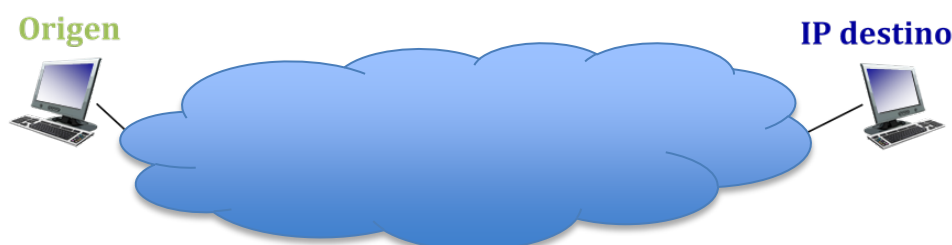


Figura 4. Comprobación de que un host destino es accesible desde un host origen.

Por medio del tiempo de espera de la respuesta a ese envío de información se determina el retraso o no de esa respuesta, lo que se conoce como **latencia**. Por ejemplo, si nos queremos conectar a un servidor, la latencia mide el tiempo de respuesta cuando enviamos una solicitud a dicho servidor.



El concepto es similar al empleado por submarinos al utilizar el sonar, en este caso el medio de transmisión no es el agua, sino las redes informáticas.

El tiempo de respuesta que tenga el comando Ping pudiera revelar lo que buscamos. Por ejemplo, un ping rápido o de baja latencia indica que hay una mejor conexión. Por el contrario, una mayor latencia puede ser síntoma de algún problema.

El comando ping se debe ejecutar **en la consola de Windows** (o de otro sistema operativo) seguido de la dirección IP del sistema de terminal de destino:

```
>ping 8.8.8.8
```

El comando ping entregará la siguiente información:

- **Tiempo de respuesta en milisegundos (ms).** O tiempo de latencia, indica cuánto tiempo necesita un paquete de datos para ir al ordenador de destino y volver.
- **Periodo de validez de los paquetes ICMP (Time to Live, TTL).** De manera general el TTL se ve disminuido en una unidad por cada nodo de red, en general un router, por el que pasa el paquete de datos. En este caso se habla de hops (saltos). Si el TTL disminuye a 0, entonces el router rechaza el paquete de datos. En el comando ping el (número máximo de saltos iniciales – TTL) indica por cuantos routers ha pasado el paquete de vuelta.

El valor inicial es de máximo 255. Son habituales implementaciones con un TTL inicial de 31, 63 o 127.

Si queremos saber cuál es el nombre del sistema terminal de destino asociada a la dirección IP:

```
>ping -a 8.8.8.8
```

Si queremos limitar el número máximo de saltos (TTL inicial) usar `-i` seguido del número máximo de saltos (en el ejemplo 3):

```
>ping -i 3 8.8.8.8
```

Uso del comando ping para diagnóstico:

- **Verificación de los protocolos TCP/IP en el emisor**

La ejecución de `ping localhost` (o `ping 127.0.0.1`) permite verificar si el conjunto de protocolos TCP/IP está correctamente instalado y en funcionamiento. Es enviado y respondido internamente por el propio equipo.

- **Verificación del adaptador de red**

Si ejecutamos ping a la IP del propio equipo, el comando es enviado a la red y recibido por el propio equipo, el cual envía la respuesta a la red y la recoge de ella. Esto permite verificar si la tarjeta de red está funcionando adecuadamente.

En Windows, la dirección IP del equipo (y otra información relevante sobre el adaptador de red) se muestra con el comando `ipconfig /all` ejecutado en la consola de Windows.

- **Verificación de la red local**

Si ejecutamos ping a una IP de un equipo de la misma red local podremos verificar si el cableado del equipo hacia la red (o si el adaptador inalámbrico) funciona correctamente.

Si ejecutamos ping a la IP de la puerta de enlace podremos verificar si el cableado general de la red funciona correctamente.

En Windows la dirección de la puerta de enlace o Gateway se muestra con el comando `ipconfig /all` ejecutado en la consola de Windows.



- **Verificación de la conexión a Internet**

Si ejecutamos `ping 91.198.174.194` (IP de Wikipedia) podremos verificar si la conexión a Internet está funcionando.

- **Verificación de los servidores DNS**

Siempre que no se haya modificado el fichero de hosts del equipo si ejecutamos `ping www.wikipedia.com` (o cualquier otra URL conocida) podremos verificar si están correctamente configuradas las IP de los servidores DNS.

2 Traceroute o Tracert

Otra utilidad de red bastante similar al ping es el *traceroute* en sistemas Linux o *tracert* en sistemas Windows que, además de enviar un paquete a un destino como hace ping, va mostrando la ruta que éste sigue, incluyendo otros datos de interés como los tiempos que tarda en cada salto y los hosts que visita hasta llegar al destino.

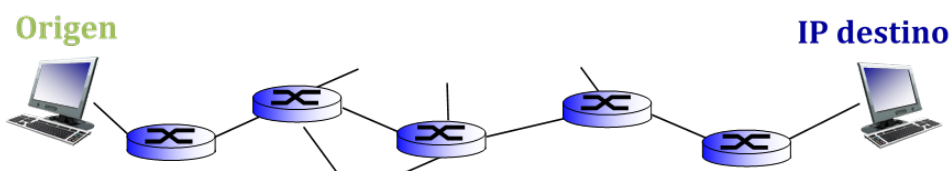


Figura 5. Comprobación de la ruta seguida por un paquete entre un host origen y destino.

En Linux, *traceroute* por defecto, aunque se puede cambiar, usa paquetes UDP (es decir usa el protocolo UDP de la capa de transporte). Sin embargo, en Windows, *tracert* envía paquetes de solicitud ICMP (es decir usa el protocolo ICMP de la capa de red) exactamente igual que el comando ping.

Para utilizar este comando en Windows debemos escribir en la consola de comandos `tracert` seguido de la dirección IP de destino:

```
>tracert 8.8.8.8
```

```
>tracert 91.198.174.194
```

También podemos escribir directamente `tracert` seguido de una dirección URL de destino:

```
>tracert www.wikipedia.com
```

3 Comunicación PC y móvil

Si se dispone en casa de un router inalámbrico se puede comprobar la comunicación entre un PC y un móvil o entre varios móviles haciendo ping de un equipo a otro.



Se puede descargar para Android la **APP Fing** que permite conectarse a una red móvil y hacer ping, traceroute, a los equipos que se quiera desde el móvil.

4 Ejercicios

1. Comprobar que los protocolos TCP/IP están correctamente instalados en el equipo de cada alumno.
2. ¿Podemos acceder a la IP 91.198.174.194 en solo 4 saltos? Comprobarlo de 2 maneras diferentes.
3. ¿Cuál es el nombre del equipo con IP 91.198.174.194?



4. Por cuantos routers pasa un paquete para llegar desde mi equipo a la IP 91.198.174.194.
5. Usando el comando ping y tracer a la IP 91.198.174.194 obtener el TTL inicial que se está usando en el comando ping.
6. En relación al adaptador de red usado para conectarse a internet, contestar:
 - Adaptador de red usado para conectarse a Internet:
 - Dirección MAC de mi adaptador (o dirección física):
 - Dirección IP de mi equipo:
 - Dirección IP de la puerta de enlace (gateway):
7. Comprobar que el adaptador de red está funcionando correctamente.
8. Comprobar que tenemos acceso al router de salida de nuestra red local.
9. Comprobar que tenemos acceso a internet.
10. Comprobar que funciona adecuadamente el servidor DNS.

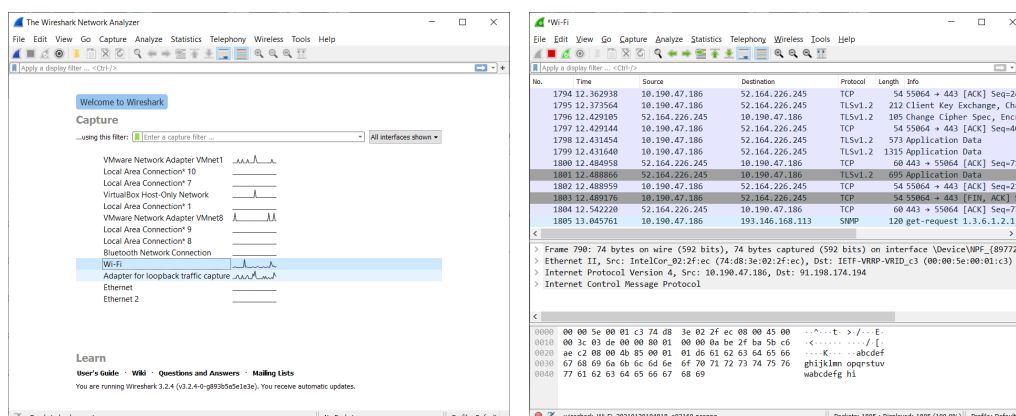
V Protocolos y Wireshark

Wireshark es un analizador de protocolos (sniffer) utilizado para realizar análisis y solucionar problemas en redes de comunicaciones, para desarrollo de software y protocolos, y como una herramienta didáctica.

Permite examinar datos de una red viva o de un archivo de captura salvado en disco. Se puede analizar la información capturada, a través de los detalles y sumarios por cada paquete. WireShark incluye un completo lenguaje para filtrar lo que queremos ver

Wireshark es software libre, y se ejecuta sobre la mayoría de los sistemas operativos Unix, Linux, así como en Microsoft Windows.

Arrancar WireShark, elegir el adaptador de red a capturar y comenzar la captura (botón derecho).



¿Tu ordenador se está comunicando hacia el exterior? ¿Por qué lo sabes?

Hacer un ping a la IP 91.198.174.194, ¿cuántos ping se han ejecutado en la pantalla de comandos? ¿Cuántos paquetes se han enviado y recibido (el protocolo de dichos paquetes es ICMP)? Localizarlos en Wireshark.



VI Bibliografía

<http://www.calculadora-redes.com/>





Grado en Ingeniería Informática

REDES

PRÁCTICA 5

Configuración de redes en Linux

Docentes:

Alejandro Merino

Daniel Sarabia Ortiz

*Dpto. de Ingeniería Electromecánica
Área de Ingeniería de Sistemas y Automática*

Versión 2.0

Fecha 08/02/2022 19:30

Esta obra está sujeta a la licencia Reconocimiento 4.0 Internacional de Creative Commons. Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by/4.0/>



Índice de contenidos

I	INTRODUCCIÓN.....	3
II	OBJETIVOS	3
III	DIRECCIONAMIENTO MAC Y DIRECCIONAMIENTO IP.....	3
	1 Direcciones IP	4
	2 Máscara de red	4
	3 Ejemplos IP y máscara de red	5
IV	CONTENIDOS ESPECÍFICOS DEL TEMA	8
	1 Actividades básicas de repaso para la gestión de archivos en Linux....	8
	2 Comandos básicos de gestión de redes en Linux	9
	2.1 ifconfig.....	9
	2.2 ping.....	10
	2.3 arp.....	11
	2.4 route.....	11
	2.5 Comando IP	11
	2.6 ip show	12
	2.7 ip link set.....	12
	2.8 ip addr.....	12
	2.9 ip route.....	12
	2.10 ip neigh.....	13
	3 Configuración permanente de la red.....	14
	4 Scripts	14
	4.1 Creación del archivo	14
	4.2 Edición del script.....	15
	4.3 Ejecución del script.....	15
	4.4 Creación de scripts más complejos.....	16
	5 Actividades.....	17
	5.1 Configuración de los VPCS.....	19
	5.2 Configuración UbuntuServer (ifconfig/ip).....	19
	5.3 Tabla ARP en UbuntuServer (arp)	20
	5.4 Configuración de la puerta de enlace (Gateway).....	21
	5.5 Direcciones permanentes.....	23
	5.6 Subredes IP.....	23
	5.7 Scripts	23
V	CONFIGURACIÓN DE RED ÚLTIMAS VERSIONES UBUNTU.....	24
VI	BIBLIOGRAFÍA	25



I Introducción

Los sistemas operativos con kernel Linux permiten gestionar y administrar redes así como proporcionar múltiples servicios de red; HTTP, DNS, FTP, etc. Por ello vamos a dedicar una práctica para introducir los comandos básicos de configuración de redes en máquinas Linux.

II Objetivos

- Introducir el uso del software de emulación de redes GNS3 que se va a usar en el resto de la asignatura y el analizador de paquetes WireShark.
- Introducir la capa de enlace y la capa de red del modelo TCP/IP, así como los conceptos básicos del direccionamiento MAC e IP para poder establecer una comunicación entre máquinas en redes de área local cableadas tipo Ethernet.
- Aprender los comandos básicos que nos permitan configurar la red en un sistema Linux.
- Ser capaz de realizar un script en Linux para automatizar la ejecución de algunos comandos de configuración de redes.
- Se suponen ciertos conocimientos básicos de los comandos de Linux para el manejo de archivos y directorios. En caso contrario se recomienda realizar las actividades opcionales.

III Direccionamiento MAC y direccionamiento IP

La pila de protocolos de Internet o modelo TCP/IP es un conjunto de protocolos de red en los que se basa Internet y que permiten la transmisión de datos entre computadoras. Actualmente es el modelo por excelencia usado en la mayoría de redes.

Se denomina modelo TCP/IP, de acuerdo a los dos protocolos más importantes que la componen, que fueron de los primeros en definirse, y que son los dos más utilizados: TCP (Transmission Control Protocol), Protocolo de Control de Transmisión e IP (Internet Protocol), Protocolo de Internet.

El modelo TCP/IP está formado por 4 niveles o capas:

- Nivel de aplicación.
- Nivel de transporte.
- Nivel de red.
- Nivel de enlace.

A lo largo de la asignatura y de otras prácticas se irá explicando en detalle el funcionamiento y las características de dichos niveles.

En esta práctica introductoria, el concepto básico para que dos máquinas en el entorno del modelo TCP/IP se puedan comunicar es que **dispongan de una dirección IP asignada**. Por máquinas a este nivel entendemos sistemas terminales o hosts (PCs, móviles, tabletas, servidores, etc.) y routers. La dirección IP es el direccionamiento que se usa en el nivel de red.

Conociendo la dirección IP de cualquier máquina es posible establecer una comunicación con ella.

Sin embargo, para que la comunicación realmente tenga lugar (por ejemplo en redes locales tipo Ethernet) se necesita conocer otro tipo de dirección, la denominada dirección MAC. Esta dirección es propia de la tarjeta de red del sistema terminal o del router y se asigna por el fabricante de tarjetas de red. La dirección MAC (llamada enlace o física o LAN o Ethernet) es el direccionamiento de los nodos a nivel de la capa de enlace, es decir de aquellos dispositivos que la implementan (hosts, routers y switches).



Está formada por 48 bits en notación hexadecimal: YY-YY-YY-YY-YY-YY (cada número Y representa 4 bits) y existen 2^{48} posibles direcciones. Ejemplo dirección MAC:

1A-2F-BB-76-09-AD

El conocimiento de esta dirección MAC **es necesaria pero transparente para el usuario**, es decir, un usuario que quiera comunicarse con otra máquina de destino deberá conocer simplemente la dirección IP de dicha máquina.

Las diferencias entre direcciones MAC y direcciones IP son:

- Las direcciones MAC tienen una estructura plana y nunca varía. La dirección MAC de la tarjeta de red es la misma independientemente de a que red esté conectada la máquina. Por tanto la MAC es portable.
- Las direcciones IP tienen una estructura jerárquica (una parte de red y una parte de host) y puede cambiar. Es necesario modificar la dirección IP cuando una máquina se mueve, es decir, cuando cambia la red a la que está conectado. Por tanto la IP no portable

Una posible analogía:

- La dirección MAC es similar al número del carnet de identidad de una persona. No cambia independientemente de a dónde se vaya a vivir esa persona
- La dirección IP es similar a la dirección postal de una persona. Debe modificarse cada vez que una persona cambia de domicilio

1 Direcciones IP

Las direcciones IP, en su formato IPv4, son números con una longitud de 32 bits, que identifican una determinada interfaz de red de un dispositivo que utilice el protocolo IP.

Las direcciones IP siguen una estructura jerárquica. En el caso de no existir subredes, cada dirección de 32 bits está compuesta por dos secciones.

- La sección que se corresponde con los bits superiores de la dirección, de longitud variable, identifica a la red.
- La sección que se corresponde con los bits inferiores se utiliza para identificar los equipos (hosts).



En la figura superior los i primeros bits se corresponden con los bits de la red, y los k últimos con los de los equipos. La porción correspondiente a la red es la misma para todos los equipos de una misma red.

Las direcciones IP suelen escribirse en notación decimal con puntos, para ello se dividen los 32 bits en cuatro grupos de 8 bits (1 byte). Por tanto, en formato decimal cada uno de los 4 bytes que componen la dirección IP tendrá un valor entre 0 y 255, por ejemplo:

128.67.202.35

2 Máscara de red

Como se indica anteriormente, los bits que se dedican a la dirección de red y a los equipos es variable. La cantidad de bits que se destinan a la red y los que se dedican a los equipos



se determina mediante la máscara de red. La máscara de red es un número binario que, al igual que una dirección IP, contiene 4 grupos de 8 bits. Por ejemplo:

```
11111111.11111111.11111111.11100000
```

Esta dirección se suele expresar, al igual que las direcciones IP en formato decimal con puntos. Por ejemplo, la dirección anterior sería:

```
255.255.255.224
```

Las máscaras, en formato binario, comienzan con un determinado número de unos y finalizan con ceros. El número de ceros indica los bits de una dirección IP que se destinan a los equipos. La máscara anterior, por ejemplo, nos estaría indicando que los últimos 5 bits de la dirección IP estarían dedicados a los hosts que hay en la red y los 27 primeros bits identifican la red.

El número de direcciones disponibles en la red es 2^k . Siendo k el número de ceros de la máscara (contados por la derecha). En el ejemplo anterior, dado que existen 5 ceros, el número de IPs posibles es, 2^5 , es decir 32.

El número de equipos que pueden ubicarse en la red puede determinarse fácilmente a partir de esto y será:

$$\text{Número de hosts} = 2^k - 2$$

Esto es así ya que:

- La primera dirección del rango de direcciones disponibles está destinada a identificar la subred. Se emplean en las tablas de enrutamiento para especificar los destinos. La dirección de red debe tener todos los bits de la subred a 0.
- La última dirección del rango se utiliza para la dirección de broadcast o dirección de difusión, que es la dirección a la que se envía un mensaje que se desea que llegue a todos los nodos de la red. La dirección de broadcast tiene todos los bits de la subred a 1.

En el ejemplo anterior, hay disponibles 32 direcciones IPs pero en esa red solo pueden ubicarse 30 equipos, ya que una dirección IP se reserva para identificar la red y otra se reserva para la dirección de broadcast.

La máscara de red también puede expresarse en formato punto decimal junto a la dirección IP, en la forma /X, siendo X el número bits utilizados para identificar a la red. A este número también se le llama **prefijo**. La dirección anterior con este formato se escribiría como: /27

3 Ejemplos IP y máscara de red

Ejemplo 1

Sea la dirección de red 192.168.35.0 con máscara 255.255.255.192

Los tres primeros campos al ser 255 serán todo unos en la máscara en binario y el cuarto campo al ser 192 se corresponderá con k ceros calculados como $256 - 192 = 2^k$, por tanto $k = 6$ ceros.

La máscara en binario quedará:

```
255.255.255.192 -> 11111111.11111111.11111111.11000000
```

Para obtener la máscara en forma de prefijo /X hay que tener en cuenta que la relación entre el prefijo X y el número de ceros es $32 - X = k$.

Por tanto $32 - X = 6$, lo que queda $X = 26$



Esta dirección puede escribirse como 192.168.35.0/26

Dispone de 26 bits para la red y por tanto de 6 bits para los hosts.

El número de IPs disponibles y hosts que se pueden asignar con esta máscara de red serán:

$$\text{Nº IPs} = 2^6 = 64$$

$$\text{Nº hosts} = 2^6 - 2 = 62$$

La **dirección de la subred** es la que tiene todos los bits de los equipos a 0 (siempre tienen que acabar en número par): 192.168.35.0

La **dirección de Broadcast** se corresponde con la última de las 64 direcciones del rango (la dirección de difusión siempre tiene que acabar en número impar): 192.168.35.63

El **rango disponible para los hosts** será: De la 192.168.35.1 a la 192.168.35.62.

Ejemplo 2

Sea la dirección de red 192.168.35.0/26

Para obtener la máscara en forma decimal o binario hay que tener en cuenta que la relación entre el prefijo X y el número de ceros es $32 - X = k$.

Por tanto $32 - 26 = k$, lo que queda $k = 6$

Es decir la máscara en binario tendrá todo unos y 6 ceros al final:

/26 -> 11111111.11111111.11111111.11000000

Para obtener la máscara en formato decimal hay que tener en cuenta que los tres primeros campos son todo unos en la máscara en binario y el cuarto campo contiene 6 ceros.

$$(11111111) = 256 - 2^0 = 255$$

$$(11000000) = 256 - 2^6 = 192$$

La máscara en decimal quedará:

11111111.11111111.11111111.11000000 -> 255.255.255.192

Ejemplo 3

Sea la dirección de red 192.168.35.0/18

Para obtener la máscara en forma decimal o binario hay que tener en cuenta que la relación entre el prefijo X y el número de ceros es $32 - X = k$.

Por tanto $32 - 18 = k$, lo que queda $k = 14$

Es decir la máscara en binario tendrá todo unos y 14 ceros al final:

/18 -> 11111111.11111111.11000000.00000000

Para obtener la máscara en formato decimal hay que tener en cuenta que los dos primeros campos son todo unos en la máscara en binario, el tercer campo contiene 6 ceros y el cuarto campo contiene 8 ceros.

$$(11111111) = 256 - 2^0 = 255$$

$$(11000000) = 256 - 2^6 = 192$$

$$(00000000) = 256 - 2^8 = 0$$

La máscara en decimal quedará:

11111111.11111111.11000000.00000000 -> 255.255.192.0

El número de IPs disponibles y hosts que se pueden asignar con esta máscara de red serán:



Nº IPs = $2^{14} = 16384$

Nº hosts = $2^{14} - 2 = 16382$

La **dirección de la subred** es la que tiene todos los bits de los equipos a 0 (siempre tienen que acabar en número par): 192.168.0.0

La **dirección de Broadcast** se corresponde con la última de las 16384 direcciones del rango (la dirección de difusión siempre tiene que acabar en número impar): 192.168.63.255

El **rango disponible para los hosts** será: De la 192.168.0.1 a la 192.168.63.254.

Ejemplo 4

Se dispone de un bloque direcciones IP lo suficientemente grande, por ejemplo:

157.126.20.0/24

Y se necesita asignar un bloque de direcciones para 20 equipos.

Para ubicar 20 equipos necesito por lo menos 5 bits dedicados a los hosts, ya que $2^5 - 2 = 30$, aunque vayan quedar sin utilizar 10 direcciones. ($2^4 - 2 = 14$ no son suficientes).

La máscara de red será por tanto:

255.255.255 y luego: 11100000 = 224 (255.255.255.224)

Luego, el bloque de direcciones que necesito para poder dar direcciones IP a esos 20 equipos será:

157.126.20.0/27

Rango de direcciones IP disponible para los equipos de la red será:

157.126.20.1 - 157.126.0.30

La dirección de broadcast será la última del bloque: 157.126.0.31



IV Contenidos Específicos del tema

1 Actividades básicas de repaso para la gestión de archivos en Linux

En la asignatura de Sistemas Operativos del grado se ha trabajado ya con sistemas operativos basados en Linux, por lo que se conocen ya los comandos básicos de gestión de ficheros. Se presentan aquí una serie de ejercicios básicos de repaso que se recomienda realizar a los alumnos que no hayan cursado la asignatura mencionada.

Actividades Opcionales

Actividad 1. Mostrar la carpeta en la que me encuentro actualmente (*comando pwd*).

Actividad 2. Desplazarme al directorio padre del fichero en el que me encuentro (*comando cd ..*).

Actividad 3. Desplazarme al directorio más alto de la estructura de directorios (*comando cd /*).

Actividad 4. Mostrar los ficheros que existen en la carpeta en la que me encuentro (*comando ls*)

Actividad 5. Dentro tu directorio home para tu usuario(*/home/redes*) Crea dos directorios d1 y d2 y sin moverte de carpeta otro directorio d3 dentro de d1 (*mkdir*).

Actividad 6. Sin moverte de directorio crea, dentro del directorio d3 (*comando cd*) un fichero vacío que se llame f1 (*gedit f1*).

Actividad 7. Copia este fichero a la carpeta d2 (*comando cp*).

Actividad 8. Muestra las propiedades del d2 (*comando ls -l*).

Actividad 9. Modificar los permisos del fichero (*comando chmod*) para permitir la lectura, escritura y ejecución por parte de cualquier usuario.

Actividad 10. Modificar los permisos del fichero para permitir que el propietario del fichero lo pueda leer, escribir y ejecutar, el grupo del propietario pueda leerlo, y el resto de los usuarios no pueda hacer nada.

http://www.ite.educacion.es/formacion/materiales/85/cd/linux/m1/permisos_de_archivos_y_carpetas.html



2 Comandos básicos de gestión de redes en Linux

En este curso utilizaremos la distribución de Linux Ubuntu. Ubuntu dispone de una interfaz gráfica de usuario desde la que es posible configurar las redes de manera intuitiva. Sin embargo, es interesante aprender a configurar las redes a través de línea de comandos, ya que se dispone de muchas más opciones de configuración y no se depende de una interfaz gráfica concreta.

En estas prácticas se utilizará GNS3 y máquinas virtuales Ubuntu Server para realizar distintas configuraciones básicas de red utilizando una serie de comandos que se enumeran a continuación.

2.1 ifconfig

Las interfaces de red permiten la conexión del núcleo con el hardware de red. En Linux es posible configurar estas interfaces mediante el comando *ifconfig*. En la configuración se puede activar o desactivar la interfaz y asignar a la interfaz una dirección IP y otros parámetros.

Las interfaces en una máquina pueden ser variadas, en nuestro caso vamos a trabajar únicamente con interfaces Ethernet que aparecen nombradas como *ethn* (eth0, eth1...)¹.

Si se ejecuta *ifconfig* sin parámetros proporciona información acerca de las interfaces **activas** en la máquina. Si se ejecuta con la opción *-a* mostrará todas las interfaces, estén activas o no.

Al listar las interfaces en una máquina aparece siempre la interfaz *lo*. Esta Interfaz se denomina interfaz de loopback y se utiliza cuando el destino de los datos que se envían es el propio equipo. Se utiliza para pruebas de diagnóstico y conectividad. Las direcciones de la interfaz de loopback son las correspondientes al rango 127.0.0.0/8

Hay que tener en cuenta que las configuraciones de red que se realicen mediante la ejecución de comandos son temporales y se perderán en el rearranque de la máquina. Para modificar la configuración de la red de manera que no se pierda consultar el apartado IV3.

La sintaxis del comando *ifconfig* es la siguiente:

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

Dónde:

<interfaz> es el nombre de la interfaz que se está configurando (eth0, eth1...).

<dir_ip> es la dirección ip de la interfaz que deseamos asignar a esa interfaz.

<netmask> es la máscara de red que se desea asignar a esa dirección IP. Por defecto, si no se especifica nada, se determinará una máscara de cnetlase A, B o C en función de la dirección IP que se haya escogido en *dir_ip*.

<dir_broadcast> Dirección de broadcast. La dirección de broadcast IPv4 es una dirección especial para cada red que permite la comunicación a todos los hosts en esa red. Para enviar datos a todos los hosts de una red, un host puede enviar un solo paquete dirigido a la dirección de broadcast de la red.

Un ejemplo de uso de *ifconfig* sería:

```
#ifconfig eth1 10.0.0.10 netmask 255.0.0.0 broadcast 10.255.255.255
up
```

¹ En las versiones más recientes de Ubuntu, las interfaces de red utilizan, para el caso de las tarjetas ethernet, los siguientes nombres *enpxsy* que indica que es una red ethernet, en el bus PCI *x* y en slot *y*.



Los comandos *ifconfig up* e *ifconfig down* activan y desactivan la interfaz determinada:

```
ifconfig <interfaz> up
ifconfig <interfaz> down
```

Por ejemplo:

```
#ifconfig eth1 down
```

Importante: Muchas de las operaciones de gestión de la red necesitan ejecutarse como superusuario, por lo tanto, en muchos casos será necesario escribir **sudo** delante de los comandos.

2.2 ping

El comando *ping* permite obtener un eco en los sistemas IPv4. Nos informa además del retardo de ida y vuelta de un paquete (RTT Round Trip Time).

La sintaxis del comando *ping* es la siguiente:

```
ping <dir_ip>[opciones]
```

Dónde:

<dir_ip> es la dirección IP de la que se desea obtener el eco.

Algunas de las opciones del comando *ping* son:

- b Hacer ping a una dirección broadcast.
- c *N* Se envían *N* paquetes.
- n Mostrar las direcciones de red como números.
- q Salida silenciosa. No se muestra nada excepto las líneas resumen al principio y al final.
- i Especifica un intervalo entre transmisiones sucesivas. Por defecto es un segundo.
- t Establece el Tiempo de Vida de los paquetes en segundos.

...

Ejemplos de uso de ese comando serían:

```
#ping 8.8.8.8
#ping 10.255.255.254 -b      #(ping de broadcast)
```



2.3 arp

El comando `arp` permite mostrar y modificar la tabla ARP de la máquina. La tabla ARP muestra las direcciones IP y su correspondiente MAC de los equipos de la subred en la que estamos.

La sintaxis del comando `arp` es la siguiente:

```
arp [-v|-n] [-Tipo_Hard] [-i if] -a [hostname]
arp [-v] [-i if] -d hostname [pub]
arp [-v] [-H type] [-i if] -s hostname hw_addr [temp]
```

Dónde:

- n* muestra las direcciones en formato numérico
- Tipo_hard*, tipo de hardware de la interfaz.
- i if* Selecciona una interfaz.
- d hostname* elimina todas las entradas del nodo hostname de la tabla ARP.
- s hostname hw_addr*, crea una entrada ARP para el equipo.

2.4 route

El comando `route` muestra y configura la tabla de enrutamiento.

La sintaxis del comando `route` es la siguiente:

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

Dónde:

- net/host* indica si el destino de la ruta es un host o una red.
- <*dir_ip*> es la dirección IP del host/red.
- <*dir_ip_gateway*> es la dirección IP de la puerta de enlace o Gateway.
- <*netmask*> es la máscara de red de la ruta que se quiere añadir.
- <*Interfaz*> es el nombre de la interfaz que se utilizará en esa ruta.

2.5 Comando IP

El paquete de Linux `Iproute2`, proporciona herramientas muy potentes para la administración de interfaces de red y conexiones. Este paquete incluye el comando `ip`, que es mucho más potente que `ifconfig` ya que incluye las mismas funcionalidades que `ifconfig`, `route` y `arp` y más.

Sintaxis:

```
ip [ OPCIONES ] OBJETO { COMANDO | help }
OPCIONES := { -V[ersion] | -s[tatistics] | -r[esolve] | -f[amily] {
inet | inet6 | ipx | dnet | link } | -o[neline] }
OBJETOS := { link | addr | addrlabel | route | rule | neigh | tunnel
| maddr | mroute | monitor }
```



2.6 ip show

Muestra dispositivos de red y su configuración.

`ip addr show` muestra más o menos la misma información que `ifconfig` sin parámetros.

`ip link show` muestra la parte de la información que se corresponde con el enlace.

`ip route show` muestra la parte de la información que se corresponde con el enrutado (si lo hay) y con la puerta de enlace.

2.7 ip link set

El comando `ip link` sirve para configurar los interfaces a nivel de enlace, con los parámetros `up` y `down` permite activar y desactivar una interfaz de red aunque tiene muchos más parámetros y posibilidades.

```
ip link set device up|down      (ifconfig device up|down)
```

2.8 ip addr

Permite configurar las direcciones de red

```
ip address {add|del} dir_ip {label alias | broadcast difusion} dev
interface
```

Ejemplos:

```
ip address add 192.168.1.4 dev eth0
ip address add 192.168.1.4/24 dev label eth0:0 dev eth0
ip address del 192.168.1.4 dev eth0
```

2.9 ip route

Si se dispone de un equipo con varias tarjetas de red, Linux posee varias instrucciones que permiten configurar rutas IPv4 de tal forma que nuestro equipo se comporte como un router. En el siguiente enlace puede encontrar información acerca de cómo hacerlo:

http://www.ite.educacion.es/formacion/materiales/85/cd/linux/m6/enrutamiento_en_linux.html

Sintaxis

```
ip route { add | del | change | append | replace | monitor } RUTA
```

add/del – añade/elimina una ruta

```
ip route add ${address}/${mask} via ${gateway} metric ${number}
ip route delete ${rest of the route statement}
```



Ejemplos:

```
ip route add 192.168.1.0/24 dev eth0
ip route add 192.168.2.0/24 via 10.0.1.1 metric 5
ip route delete 10.0.1.0/25 via 10.0.0.1
```

Ruta por defecto:

```
ip route add 0.0.0.0/0 via ${gateway}
ip route add default via ${gateway}
```

Ejemplos:

```
ip route add 0.0.0.0/0 via 10.0.0.1
ip route add default via 10.0.1.1
```

Otras opciones:

`ip route change` Cambia los parámetros de una ruta existente.

`ip route show` Muestra la lista de rutas.

Ejemplo:

```
ip route change 192.168.2.0/24 via 10.0.0.1
```

2.10 ip neigh

El comando `ip` también permite mostrar la tabla arp del equipo, para ello se utiliza el comando `ip neigh`.

```
ip neigh show #Mostrará la tabla ARP actual
```

También se pueden añadir, modificar o eliminar entradas de la tabla ARP:

Sintaxis:

```
ip neigh { add | del | change | replace } { ADDR [ lladdr LLADDR ] [
nud { permanent | noarp | stale | reachable } ] | proxy ADDR } [ dev
DEV ]
```

El siguiente ejemplo añade una entrada a la tabla ARP:

```
ip neigh add 172.16.0.2 lladdr 5B:FF:3F:B9:C6:C2 dev eth0
```



3 Configuración permanente de la red

Las configuraciones de red que se hacen utilizando ip o ifconfig, son temporales y se perderán cuando se produzca el reinicio del sistema.

Para que las modificaciones que se hagan en la configuración de red sean permanentes, es necesario modificar el archivo.

```
/etc/network/interfaces
```

Por tanto podemos editarlo tecleando:

```
sudo nano /etc/network/interfaces
```

Ahí podemos introducir la configuración de red, por ejemplo:

```
auto eth0      ---- Arranca la interfaz en el arranque

iface eth0 inet static
%%inet hace referencia a ipv4 y static a dirección estática
    address 172.16.1.50
    netmask 255.255.255.0
    gateway 172.16.1.254
    broadcast 172.16.1.255
```

Una vez salvada esta configuración debemos reiniciar el equipo (sudo reboot) y podemos ver cómo se ha mantenido la configuración de red.

Si quisiéramos dejar la configuración como DHCP, para por ejemplo tener acceso a internet en la máquina virtual deberemos escribir.

```
auto eth0
iface eth0 inet dhcp
```

4 Scripts

En Linux se pueden crear Scripts para ejecutar una serie de comandos que se ejecutarán de manera secuencial. Esto puede resultar útil para almacenar comandos en un fichero y poder ejecutarlos cuando deseemos.

Se va a mostrar a continuación cómo crear un script que configure las interfaces de red de los equipos y las rutas definidas en los apartados anteriores.

4.1 Creación del archivo

Desde un editor de texto cualquiera en Linux, podemos crear un fichero y lo vamos a llamar config_red.

Se puede crear un archivo directamente desde la consola escribiendo nano y el nombre del fichero a crear/abrir, por ejemplo:



```
nano config_red
```

Para no tener problemas con los permisos de acceso vamos a darle todos los permisos a todos los usuarios mediante el comando chmod.

```
chmod 777 /ruta_del_script/config_red
```

4.2 Edición del script

Una vez se ha creado el archivo se va a proceder a editar el script.

En la primera línea del script se debe indicar que shell que se utilizará. En nuestro caso vamos a utilizar el Shell bash, luego escribiremos.

```
#!/bin/bash
```

#! Se denomina Sha Bang y le indica al sistema que lo que se va a ejecutar una secuencia de comandos. A continuación podemos pasar a escribir los comandos de configuración de red necesarios. Por ejemplo podemos escribir:

```
ip address add 192.168.0.5/24 dev eth0
```

De esta forma cada vez que ejecutemos el script, se ejecutará ese comando.

4.3 Ejecución del script

Una vez finalizado el script lo guardaremos y podremos ejecutarlo desde la consola escribiendo:

```
./config_red
```



4.4 Creación de scripts más complejos

Vamos ahora a crear un script que nos permita ejecutar una serie de acciones en función de los argumentos que le pasemos al Script, para eso la estructura del Script debe ser la que sigue:

```
#!/bin/sh
# Script ejemplo para configuración de una dirección de red en la
# máquina
#
case "$1" in
conf)
echo "Asignando dirección de red.."
# Aquí comandos de configuración a ejecutar durante el arranque
# por ejemplo:
sudo ip address add 192.168.0.30/24 dev eth0
;;
desconf)
echo "Eliminando dirección de red.."
# Aquí comandos que se ejecutan al detener el servicio
# por ejemplo
sudo ip address del 192.168.0.30/24 dev eth0
;;
*)
echo "Modo de empleo: /etc/init.d/mi_script {start|stop}"
exit
;;
esac
exit 0
```

\$1 se refiere al primer argumento en la ejecución del script.

Para la **ejecución del script**, en este caso habrá que escribir por ejemplo:

```
./config_red conf
./config_red desconf
```

Para que se ejecuten las acciones correspondientes.



5 Actividades

Para realizar las actividades será necesario construir usando GNS3, la red de la Figura 1, que tiene dos máquinas virtuales Ubuntu y dos PCs virtuales VPCS de GNS3.

Las direcciones IPs y las máscaras de red iniciales serán las que se muestran en la figura para cada uno de los equipos.

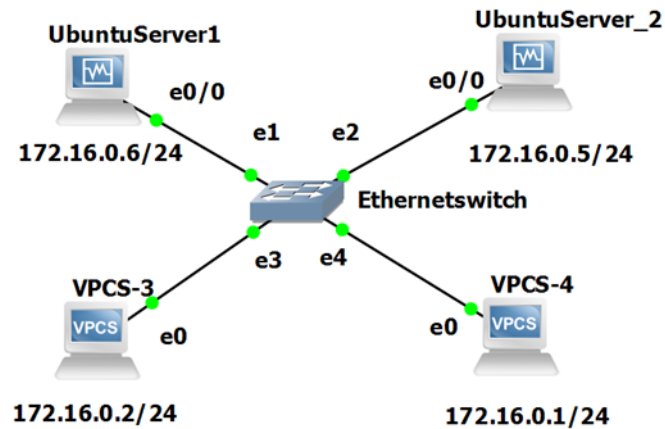
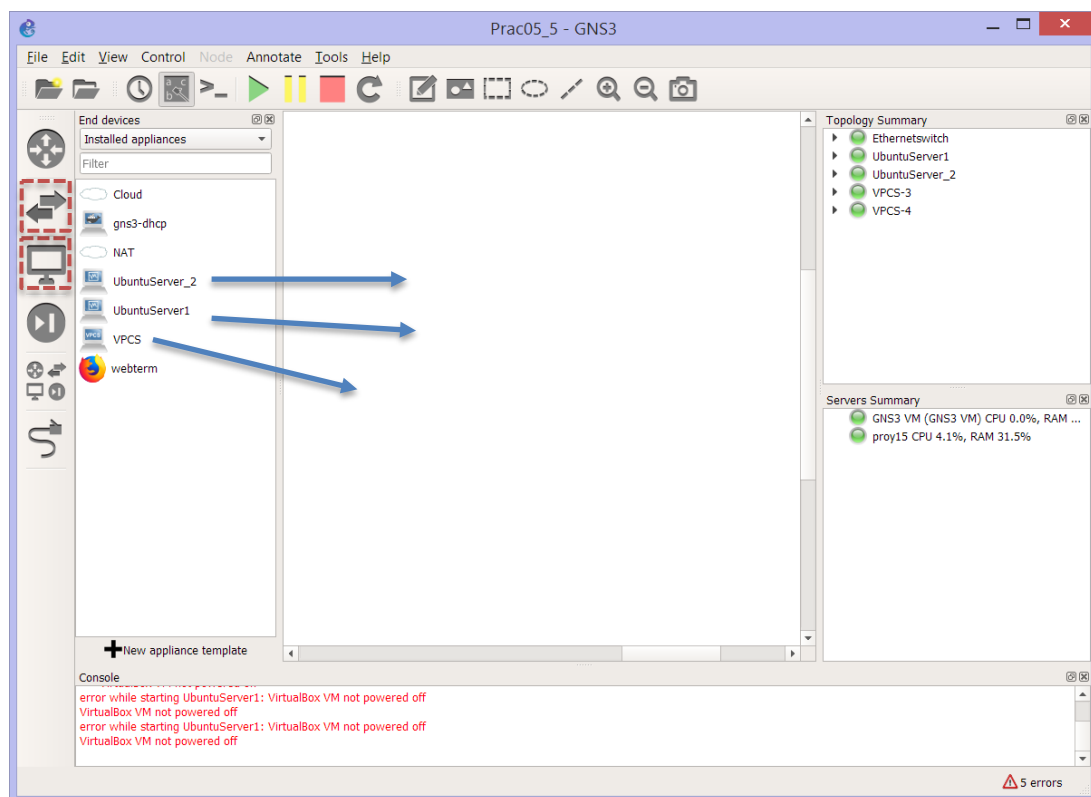


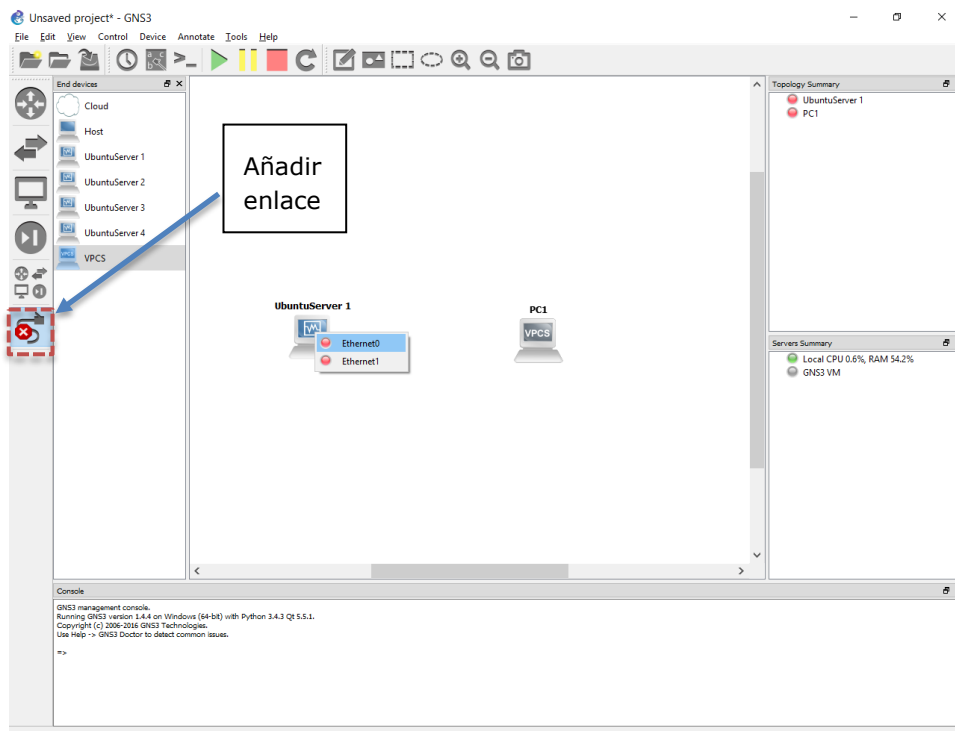
Figura 1. Configuración de red de la práctica.

Para configurar esta red es necesario arrancar GNS3 y después de dar un nombre al proyecto se muestran los dispositivos terminales:

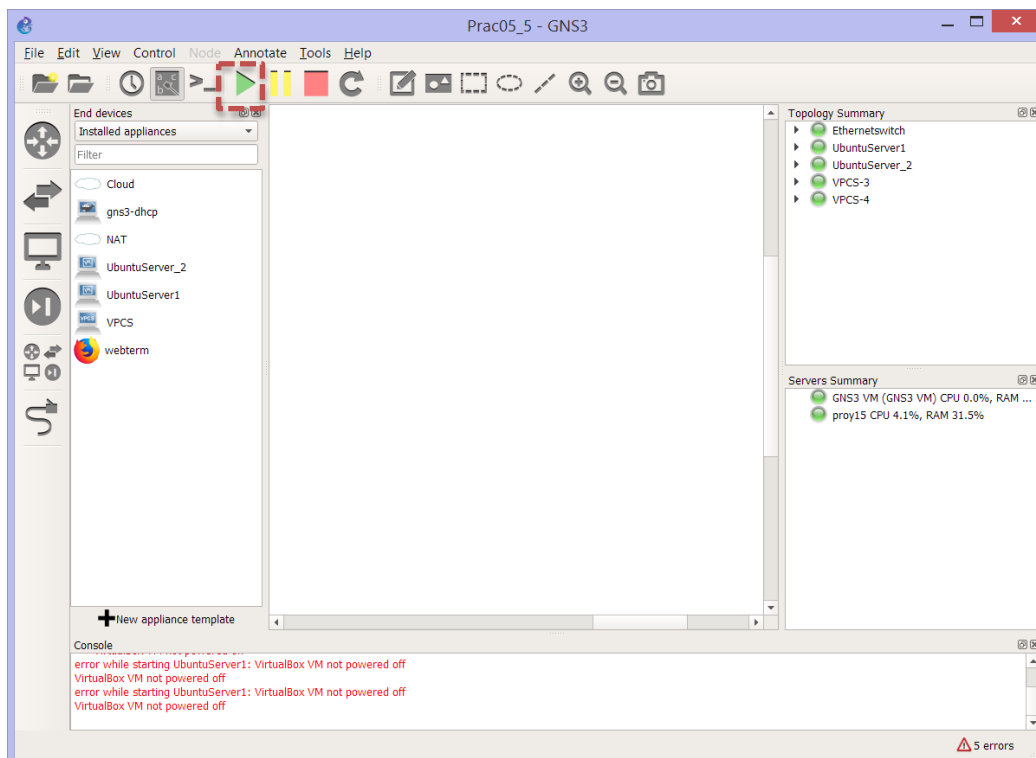


Colocar también el Ethernet switch, que será el que proporciona GNS3.

Una vez arrastrados los componentes seleccionar el elemento conector y unir los terminales:



Una vez realizada la conexión pulsar Play y deberán arrancar las máquinas virtuales Ubuntu.



a) Realizar el esquema de red suministrado en GNS3 y mostrar una captura de pantalla con el resultado.



5.1 Configuración de los VPCS

Para configurar los PCs virtuales y en general la mayoría de equipos que vienen por defecto en GNS3 se debe abrir la consola. La consola se abre, en modo de ejecución, haciendo doble click sobre el equipo o haciendo click con el botón derecho y seleccionando Console.

Para configurar la IP y máscara de red en los VPCSs se debe escribir:

```
VPCS-4>ip 172.16.0.1/24
```

Para comprobar que la asignación se ha hecho correctamente se puede escribir el comando:

```
VPCS-4>show
```

Para no perder la configuración realizada para ese equipo se debe escribir, donde nombre, es el nombre del equipo, por ejemplo VPCS-4:

```
VPCS-4>save nombre
```

De esta forma si se para y vuelve a arrancar la simulación no se pierde la configuración que sí que se perdería de otra forma.

- a) Configurar y mostrar la configuración de red del **VPCS-3** y del **VPCS-4**.

5.2 Configuración UbuntuServer (ifconfig/ip)

- a) Mostrar las interfaces activas en el equipo **UbuntuServer1** y **UbuntuServer_2**. Si únicamente aparece como activa la interfaz de loopback, ejecutar `ifconfig -a` para mostrar todas las interfaces y activar todas las interfaces Ethernet que aparezcan.
- b) Configurar las máquinas Linux: dirección IP y máscara de subred para que se cumpla lo siguiente
- La dirección IP y la máscara de red se corresponderá con el esquema de la figura.
 - Para la configuración de la IP de la máquina **UbuntuServer1** utilizar los comandos que proporciona `ifconfig` y para configurar la máquina **UbuntuServer_2** los que proporciona el comando `ip` (al usar `ip`, si ya hay una IP asignada y configurada, primero hay que borrarla y después asignar una nueva).
- c) Ejecutar el comando `ifconfig` sin parámetros para comprobar que se ha añadido la configuración en la máquina **UbuntuServer1** y el comando `ip addr show` para comprobar que se han añadido las configuraciones en la máquina **UbuntuServer_2**. Mostrar las capturas del resultado de esos comandos.
- d) Probar que es posible hacer ping entre ellos. Mostrar una captura del ping entre **UbuntuServer_2** y el **VPCS-4**.



- e) Partiendo del apartado anterior ¿Cuál es la dirección de broadcast de la red? ¿Es posible hacer ping desde el **UbuntuServer1** a la dirección de broadcast de la red?

Habilitar la respuesta a los pings de broadcast en el equipo **UbuntuServer_2**, tal como se indica abajo, y realizar nuevamente el ping de broadcast desde el equipo **UbuntuServer1**.

Para habilitar la respuesta a pings de broadcast, que por seguridad está deshabilitada por defecto en UbuntuServer_2, escribir:

```
sudo nano /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
```

y cambiar el 1 del fichero por un 0. Debería verse como, al hacer un ping de broadcast, se recibirán tantas respuestas como equipos conectados y que formen parte de la misma red. Sin embargo en GNS3 los VPCS no pueden responder a un ping de broadcast, está deshabilitado y no puede ser habilitado.

Mostrar una captura de un ping de broadcast después de haber modificado este fichero en el equipo **UbuntuServer_2**.

- f) Utilizando Wireshark, visualizar los paquetes que se capturan cuando hacemos ping entre dos equipos. Mostrar una captura con dichos paquetes de WireShark.

5.3 Tabla ARP en UbuntuServer (arp)

- a) Ejecuta el comando arp para mostrar la tabla ARP de la máquina **UbuntuServer_2**. Muestra una captura con el resultado de la ejecución de este comando.
- b) Elimina las entradas de la tabla ARP y vuelve a mostrar la tabla, utiliza los comandos proporcionados por ip neigh. Trata de que se complete la tabla nuevamente. ¿Qué hay que hacer para que se complete automáticamente la tabla?



5.4 Configuración de la puerta de enlace (Gateway)

Hasta ahora hemos configurado una serie de equipos que forman una red local (LAN) y hemos visto como configura la interfaz de red de equipos Linux y de equipos VPCs de GNS3 para que exista comunicación entre ellos. Sin embargo, si se quiere que los equipos de esta red local 1, ver figura siguiente Figura 2, puedan comunicarse con equipos que pertenezcan a otra red local o para conectarse a internet, hace falta **siempre** añadir un router. Donde una de las interfaces del router debe pertenecer siempre a la red local a la que se le quiera dar acceso al exterior (es decir que la IP de esta interfaz esté en el rango de IPs de la subred), en este caso la interfaz f0/0 y su IP 172.16.0.254 /24.

Precisamente esta dirección IP de esta interfaz del router se denomina **puerta de enlace** o **Gateway** y su valor debe ser añadida explícitamente en todas las máquinas de la red local a las que se quiera dar acceso al exterior.

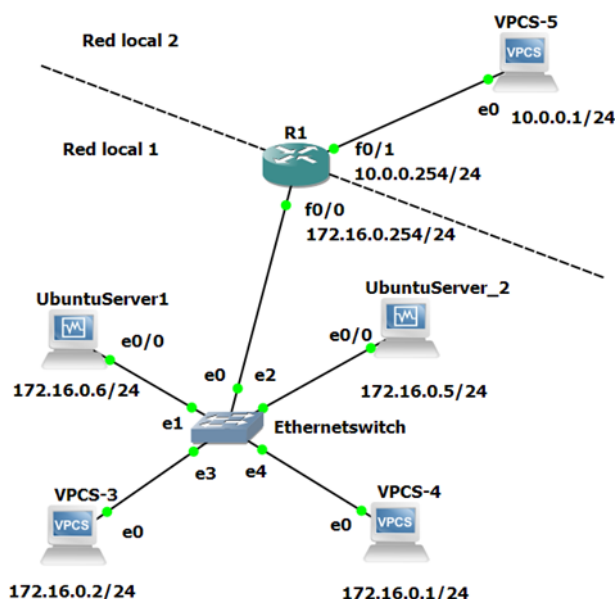


Figura 2. Configuración de red de la actividad 5.4.

En este apartado, vamos a añadir un **router R1** con dos interfaces:

- La interfaz f0/0 pertenece a la red local 1 con IP 172.16.0.254.
- La interfaz f0/1 con IP 10.0.0.254 que pertenecerá a la red local 2.

Esta segunda red local a su vez estará formada por un VPC de GNS3 (VPCS_5) con IP 10.0.0.1/24. La información de cada red local se muestra también en la siguiente tabla:

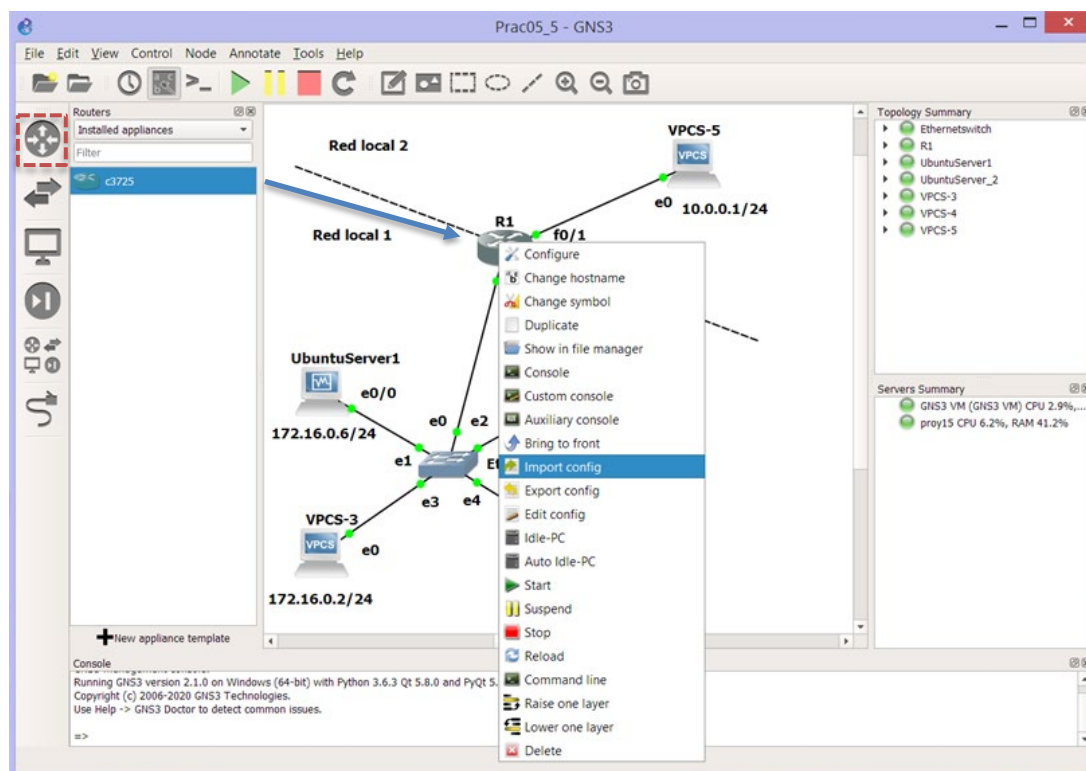
Redes	Equipos	Interfaz	IP	Máscara red	Puerta de enlace
Red local 1	UbuntuServer1	Ethernet0	172.16.0.6	255.255.255.0 /24	172.16.0.254
	UbuntuServer_2	Ethernet0	172.16.0.5	255.255.255.0 /24	172.16.0.254
	VPCS-3	Ethernet0	172.16.0.2	255.255.255.0 /24	172.16.0.254
	VPCS-4	Ethernet0	172.16.0.1	255.255.255.0 /24	172.16.0.254
	Router R1	f0/0	172.16.0.254	255.255.255.0 /24	No aplica
Red local 2	VPCS-5	Ethernet0	10.0.0.1	255.255.255.0 /24	10.0.0.254
	Router R1	f0/1	10.0.0.254	255.255.255.0 /24	No aplica



- a) Añadir al esquema el router R1 de cisco arrastrando el icono de del router c3725 al lienzo. Conectar la interfaz FastEthernet0/0 del router a la interfaz Ethernet0 del switch.

Añadir al esquema el VPCs de GNS3 y llamarlo VPCS-5. Conectar la interfaz FastEthernet0/1 del router a la interfaz Ethernet0 del VPCS-5.

Importar el fichero de configuración del router R1 "R1_i1_startup-config.cfg" pinchando con el botón derecho en el router y pinchando "Import config", seleccionar el fichero anterior. De esta manera el router queda configurado adecuadamente. En prácticas posteriores se verá como se realiza esta configuración.



- b) Configurar la interfaz de red del **VPCS-5** añadiendo la puerta de enlace de su red local y grabar su configuración para que no se pierda.

```
VPCS-5>ip 10.0.0.1/24 gateway 10.0.0.254
```

```
VPCS-5>save VPCS-5
```

- c) Configurar la interfaz de red del **VPCS-3** y **VPCS-4** añadiendo la puerta de enlace de su red local y grabar sus configuraciones para que no se pierdan.
- d) Configurar la puerta de enlace en los equipos **UbuntuServer1** y **UbuntuServer_2** y mostrar que la configuración es correcta.
- e) Comprobar y mostrar mediante un ping desde el **UbuntuServer1** al **VPCS-5** como es posible la comunicación entre redes locales distintas.
- f) Comprobar y mostrar mediante un ping desde el **VPCS-3** al **VPCS-5** como es posible la comunicación entre redes locales distintas.



5.5 Direcciones permanentes

- En la máquina **UbuntuServer_2** modificar el fichero `/etc/network/interfaces` y almacenar la configuración de red utilizada en las actividades anteriores. Mostrar el fichero modificado.
- Reiniciar la máquina **UbuntuServer_2** (`sudo reboot`) y comprobar que se ha mantenido la configuración de red.

5.6 Subredes IP

- Partiendo de las mismas IPs, configurar las máscaras de red de tal forma que los equipos **UbuntuServer1** y **UbuntuServer_2** formen una subred 2 y los equipos **VPCS-3** y **VPCS-4** formen otra subred diferente: subred 1.

Subred	Sistema terminal	IP	Máscara red	Rango IPs	Max. N° IPs
Subred 2	UbuntuServer1	172.16.0.6			
	UbuntuServer_2	172.16.0.5			
Subred 1	VPCS-3	172.16.0.2			
	VPCS-4	172.16.0.1			

- Una vez configuradas las subredes, hacer ping entre los equipos de la misma subred y entre equipos de distinta subred. Mostrar una captura de ambas pruebas.
¿Qué mensaje aparece cuando se intenta hacer ping entre equipos de distinta red?
- ¿Qué máscara habría que utilizar para volver a incluir los cuatro equipos en la misma subred y que se desperdicie el mínimo número de direcciones IP?
¿Cuál podría ser ahora la IP de la interfaz `f0/0` del router R1 conectado a la red local 1 para que esta red local tenga acceso al exterior?
¿Cuál sería ahora la puerta de enlace de todos los equipos de la red local 1?

5.7 Scripts

- Crear un script en el **UbuntuServer_2** llamado "config_red" en el que se haga la configuración de red usando el comando `ip` y que en función de un parámetro se puedan realizar las siguientes opciones:
 - Borrar la dirección IP que esté configurada.
 - Añadir una dirección IP fija (si no se ha eliminado la que existía previamente no funcionará).
 - Dar de baja un interfaz.
 - Activar una interfaz.

Con cualquier otro parámetro se dará un mensaje de aviso como el que se muestra en el ejemplo. Ejecutar el script y probar que funciona correctamente.

Escribir el código del script y capturas que muestren el funcionamiento del programa.



V Configuración de red últimas versiones Ubuntu

En las últimas versiones de Ubuntu aparecen algunos cambios respecto a lo que se ha mostrado en esta práctica.

En primer lugar, la identificación de las interfaces de red se ha modificado y han pasado de llamarse `eth0`, `eth1`, `wlan0`, `wlan1`... a direcciones del tipo `"enpxsy"` (p.e. `enp3s0`) para las interfaces ethernet o `"wlpxsy"` (p.e. `wlp1s0`) para las inalámbricas. Esta notación, aunque es algo más compleja, permite que la identificación sea predecible y se conserve en sucesivos arranques de una máquina.

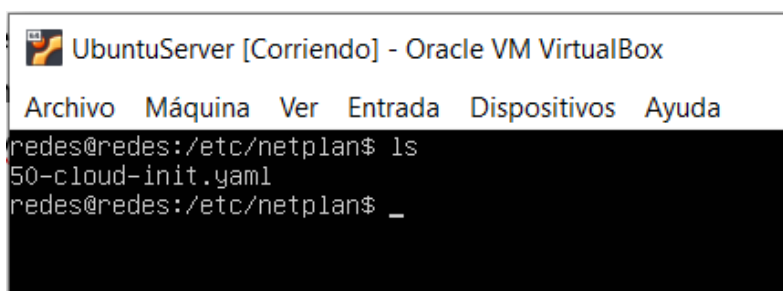
El significado de las letras es el siguiente. Las primeras letras identifican la interfaz:

- **en** se utiliza para identificar las interfaces de tipo Ethernet.
- **wl** se utiliza para identificar las interfaces inalámbricas.

Después de las letras aparece normalmente la siguiente identificación:

- **p** seguido de un número, identifica al número del bus PCI en el que se ubica la tarjeta de red.
- **s** seguido de un número, identifica el número de slot

Otro cambio que se ha introducido en las últimas versiones de Ubuntu, desde la 17.05, es la forma en la que se modifica la configuración estática permanente de las interfaces de red. El fichero `"etc/network/interfaces"` ya no está disponible. Ahora el fichero de configuración de red está en la ruta `"/etc/netplan"` (Figura 3) y tiene una extensión `.yaml`. Por ejemplo, **50-cloud-init.yaml**.

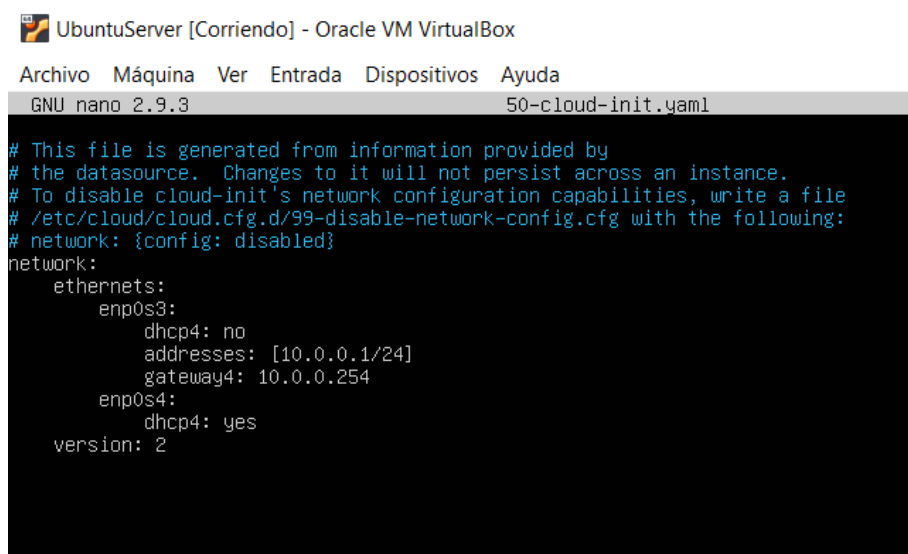


```

UbuntuServer [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
redes@redes:/etc/netplan$ ls
50-cloud-init.yaml
redes@redes:/etc/netplan$ _
  
```

Figura 3. Ubicación del fichero *.yaml.

La estructura de este fichero es tal y como se muestra en la Figura 4.



```

UbuntuServer [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
GNU nano 2.9.3 50-cloud-init.yaml
# This file is generated from information provided by
# the datasource. Changes to it will not persist across an instance.
# To disable cloud-init's network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}
network:
  ethernets:
    enp0s3:
      dhcp4: no
      addresses: [10.0.0.1/24]
      gateway4: 10.0.0.254
    enp0s4:
      dhcp4: yes
  version: 2
  
```

Figura 4. Estructura de la configuración de red en un fichero *.yaml.

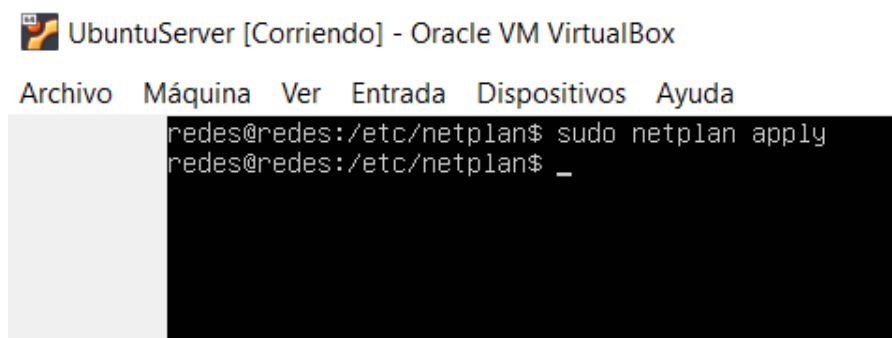


Hay que tener cuidado, porque este tipo de ficheros no admite tabulaciones y los espacios deben ser los correctos para que el archivo pueda interpretarse de manera apropiada. Cada indentación se corresponde con cuatro espacios.

Se muestran en este ejemplo las interfaces de red disponibles en la máquina. Los parámetros que aparecen son los siguientes:

- `dhcp4`: indica si la dirección IP se va a obtener mediante DHCP IPv4.
- `Addresses`: indica las posibles direcciones IP de la interfaz junto con su máscara en formato `/x`.
- `gateway4`: indica la dirección de la puerta de enlace en IPv4.
- `Versión`: hace referencia a la versión de YAML.

Una vez modificado este fichero a la configuración concreta deseada, se puede activar la misma ejecutando el comando `netplan apply` (Figura 5).



```
UbuntuServer [Corriendo] - Oracle VM VirtualBox
Archivo  Máquina  Ver  Entrada  Dispositivos  Ayuda
redes@redes:/etc/netplan$ sudo netplan apply
redes@redes:/etc/netplan$ _
```

Figura 5. Aplicación de una configuración de red con netplan.

VI Bibliografía

<http://www.computerhope.com/unix.htm>

<http://es.tldp.org/Manuales-LuCAS/GARL2/gar12/>

<https://echaleunvistazo.wordpress.com/2014/03/06/wireshark-en-ubuntu/>

<http://www.calculadora-redes.com/>





Grado en Ingeniería Informática

REDES

PRÁCTICA 6

Redes LAN, Ethernet y VLAN

Docentes:

Alejandro Merino

Daniel Sarabia Ortiz

*Dpto. de Ingeniería Electromecánica
Área de Ingeniería de Sistemas y Automática*

Versión 2.7

Fecha 30/03/2022 10:02

Esta obra está sujeta a la licencia Reconocimiento 4.0 Internacional de Creative Commons. Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by/4.0/>



Índice de contenidos

I	INTRODUCCIÓN.....	3
II	OBJETIVOS.....	3
III	COMANDOS BÁSICOS DEL IOS PARA DISPOSITIVOS CISCO	3
IV	ESTUDIO DE REDES LOCALES (LAN) CON GNS3.....	7
	1 Configuración de una red local punto a punto	7
	1.1 Estudio inicial.....	9
	1.2 Estudio del protocolo ARP y del protocolo ICMP	10
	2 Configuración de una red local (LAN) de tipo Ethernet.....	19
	2.1 Estudio inicial.....	20
	2.2 Funcionamiento del switch. Autoaprendizaje, filtrado y reenvío.....	21
V	ESTUDIO DE REDES VIRTUALES DE ÁREA LOCAL (VLAN).....	25
	1 Creación de una red LAN	25
	2 Creación de VLANs	26
	2.1 Enlaces Troncales.....	27
VI	ESTUDIO AVANZADO DE REDES LAN CON SOFTWARE DE CISCO ..	29
	1 Creación de una red LAN	29
	2 Creación de VLANs	30
	2.1 VLAN de administración.....	32
	2.2 Enlaces Troncales	33
	2.3 Enrutamiento entre VLANs.....	35
	2.4 Enrutamiento de VLANs con redes externas.....	37



I Introducción

Las LAN son redes localizadas en zonas geográficas pequeñas como pueda ser un edificio, un campus o una oficina.

Existen distintas arquitecturas LAN, pero las arquitecturas LAN cableadas más utilizadas están formadas por una serie de equipos terminales que van conectados a un conmutador de paquetes de la capa de enlace (switch) que está conectado a su vez al exterior a través de un conmutador de la capa de red (router). Es importante por tanto conocer los protocolos que permiten la comunicación entre los propios equipos de la LAN así como la posibilidad de crear subredes independientes que permitan aislar subredes de manera sencilla mediante las denominadas VLAN.

En la primera parte de la práctica se estudiarán el protocolo ARP y el funcionamiento de los switches (autoaprendizaje, inundación, filtrado y envío, así como su tabla MAC). En una segunda parte se estudiará la creación de VLANs y cómo establecer la comunicación entre ellas.

II Objetivos

- Introducir los comandos básicos del sistema operativo de dispositivos CISCO para su administración.
- Conocer el funcionamiento de los protocolos de la capa de enlace, en concreto el protocolo de resolución de direcciones (ARP).
- Reconocer los campos de la trama Ethernet y de algún otro paquete como el ICMP asociado al comando ping.
- Conocer el funcionamiento de los conmutadores de la capa de enlace, los switches, y sus funciones principales: autoaprendizaje, filtrado, reenvío e inundación.
- Conocer los comandos y las técnicas que nos permitan configurar una VLAN con enlaces troncales.
- Realizar enrutamiento entre VLANs.
- Aprender a configurar un router para dar soporte a una red local con VLANs.
- Ser capaz de configurar una red sencilla en el software de simulación/emulación GNS3.
- Ser capaz de utilizar herramientas de análisis de tráfico en redes, en este caso WireShark, interpretando el tráfico de paquetes, los protocolos utilizados, el valor de los campos de cada paquete, etc.

III Comandos básicos del IOS para dispositivos CISCO

Antes de comenzar la práctica de LAN y VLAN se van a mostrar los comandos básicos que poseen los dispositivos de CISCO y que se pueden utilizar desde la línea de comandos. Esta interfaz es la que aparece cuando nos conectamos físicamente a un dispositivo y también puede trabajarse con ella desde las consolas de GNS3.

Al entrar en la consola, se mostrará una interfaz de comandos como la siguiente:



```

S1
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html

If you require further assistance please contact us by sending email to
export@cisco.com.

Cisco 3725 (R7000) processor (revision 0.1) with 124928K/6144K bytes of memory.
Processor board ID FTX0945W0MY
R7000 CPU at 240MHz, Implementation 39, Rev 2.1, 256KB L2, 512KB L3 Cache
18 FastEthernet interfaces
DRAM configuration is 64 bits wide with parity enabled.
55K bytes of NVRAM.
1024K bytes of ATA System CompactFlash (Read/Write)
1024K bytes of ATA Slot0 CompactFlash (Read/Write)
Installed image archive

SETUP: new interface FastEthernet0/0 placed in "shutdown" state
SETUP: new interface FastEthernet0/1 placed in "shutdown" state
% There may not be enough space available to collect the complete crashinfo
% It would be advisable to have 280755 bytes free space on flash:crashinfo

Press RETURN to get started!

```

Si se pulsa *intro*, aparece el siguiente símbolo¹:

```
S2>
```

Este el símbolo que indica que estamos en **modo usuario**. En modo usuario es posible ver información relativa al dispositivo pero no modificarla.

Para ver los comandos disponibles en este modo, y en general siempre, se introduce un carácter interrogante "?":

```

S1>?
Exec commands:

 <1-99>      Session number to resume
 connect     Open a terminal connection
 disable     Turn off privileged commands
 disconnect  Disconnect an existing network connection
 enable     Turn on privileged commands
 exit       Exit from the EXEC
...

```

Para poder realizar cambios es necesario entra en modo privilegiado para ello debemos escribir *enable*, es posible que pida contraseña.

```

S1>enable
S1#

```

Para poder configurar el equipo debemos pasar al modo de **configuración global**, para ello escribimos: *configure terminal*

```
S1#configure terminal
```

¹ En GNS3, se arranca directamente en modo privilegiado, pero este no es el comportamiento normal de los equipos.



```
Enter configuration commands, one per line.  End with CNTL/Z.
S1(config)#
```

Los cambios en la configuración del dispositivo se almacenan en una memoria RAM que es volátil, por lo que se pierden en el arranque. Normalmente los dispositivos disponen una pequeña memoria no volátil (NVRAM), en la que se guarda la configuración que se carga durante el arranque.

Para visualizar la configuración existente en este momento, almacenada en la RAM, podemos escribir:

```
S1#show running-config
```

Para mostrar la memoria de arranque almacenada en la NVRAM, podemos escribir:

```
S1#show startup-config
```

Para almacenar la configuración actual en la memoria de arranque de tal forma que se pueda arrancar el dispositivo con la configuración actual:

```
S1#copy running-config startup-config
Destination filename [startup-config]? [enter]
Building configuration...
```

Cambiar el nombre del equipo:

```
S1(config)#hostname S1
S1(config)#
```

Para trasladarse hacia arriba en la estructura de directorios escribir `exit`.

Para **visualizar el estado de una interfaz** escribir `#show interface *` y `#show interfaces` para ver todas las interfaces. Por ejemplo:

```
S1#show interface FastEthernet0/1
S1#show interfaces
```

Para configurar una interfaz, es necesario escribir `interface` seguido del nombre de la interfaz, por ejemplo:

```
Router(config)#interface FastEthernet0/1
Router(config-if)#
```

En general, para hacer lo contrario de un comando hay que escribir *no* delante del comando, por ejemplo para dar de baja un interfaz es necesario escribir *shutdown* y para darlo de alta escribir *no shutdown*.

```
Router(config-if)#shutdown
```



```
%LINK-5-CHANGED: Interface FastEthernet0/1, changed state to
administratively down
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1,
changed state to down
%LINK-5-CHANGED: Interface FastEthernet0/1.1, changed state to down
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1.1,
changed state to down

Router(config-if)#no shutdown
%LINK-5-CHANGED: Interface FastEthernet0/1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1,
changed state to up
%LINK-5-CHANGED: Interface FastEthernet0/1.1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1.1,
changed state to up
```

En el siguiente enlace podéis encontrar más información:

<http://es.ccm.net/faq/2759-router-cisco-configuracion-basica>

A medida que vayamos haciendo prácticas iremos aprendiendo nuevos comandos.

La mayoría de los comandos pueden utilizarse usando únicamente las primeras letras del comando, siempre que el comando no sea ambiguo. Por ejemplo, puede escribirse:

```
Router#conf term
Router(config)#int F0/1
Router(config-if)#no sh
```



IV Estudio de redes locales (LAN) con GNS3

1 Configuración de una red local punto a punto

Se va crear una red local punto a punto, es decir una red entre dos sistemas terminales para estudiar el funcionamiento del protocolo ARP en los sistemas terminales y el protocolo ICMP. Para ello, crear un proyecto "Prac06_IV_01.gns3" en GNS3 con un sistema terminal de tipo **UbuntuServer** y un **PC de tipo virtual** conectados ambos mediante cable de par trenzado y tarjetas Ethernet.

Antes de arrancar los equipos pinchar con el botón de la derecha en la conexión entre los dos equipos y pinchar en Start capture, ver Figura 1, esto arrancará WireShark y nos permitirá analizar y capturar todo el tráfico entre los dos equipos, incluyendo los tipos de protocolos usados.

Cuando en un enlace se ha configurado una captura aparecerá una lupa encima del enlace, ver Figura 1, indicando así que los paquetes de datos que circulen por ese enlace se capturarán en WireShark.

Nota: Puede que la primera vez no se inicie WireShark automáticamente, en ese caso arrancar primero todos los equipos (pinchando en el icono Start, triángulo verde del menú superior) y después pinchar con el botón de la derecha en la conexión entre los dos equipos y pinchar en Start capture, ver Figura 1.

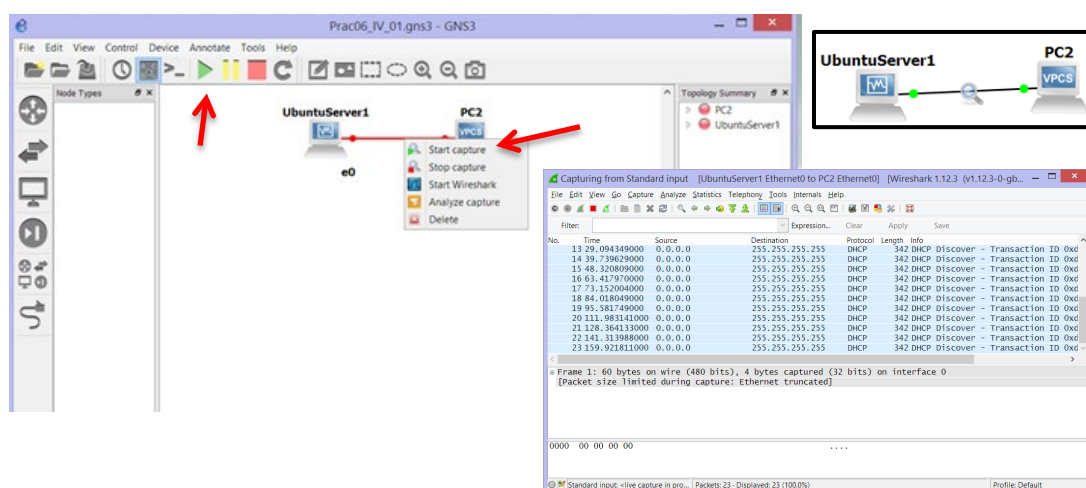


Figura 1. Red local punto a punto entre una máquina Linux (UbuntuServer1) y un pc virtual (PC2).

Después arrancar cada equipo individualmente (pinchando con el botón derecho sobre cada equipo y pulsando el Icono Start) o arrancando todos a la vez (pinchando en el icono start del menú principal, triángulo verde) y asignar las siguientes direcciones IP a cada equipo terminal:

- **UbuntuServer (UbuntuServer1).** La configuración del adaptador de red de un equipo Linux puede hacerse temporalmente o permanente tal y como se vio en la práctica anterior.

Usar la opción temporal: El nombre del adaptador de red Ethernet puede ser diferente en cada equipo: eth0, eth1, eth2, etc.

Asignar una IP estática: 192.168.0.1 con máscara de red 255.255.255.0.

```
UbutuServer> sudo ifconfig eth0 192.168.0.1 netmask 255.255.255.0 up
```

Comprobar que la configuración de la IP y de la máscara es correcta.

```
UbuntuServer> ifconfig
```



Usar la opción permanente: Si no se quiere perder la configuración de la IP del equipo con Ubuntu cuando se apague la máquina, almacenar la IP de manera permanente, modificando el fichero "interfaces". Puedes consultar cómo hacerlo en la práctica anterior.

```
UbuntuServer> sudo nano /etc/network/interfaces
```

- **PC Virtual (PC2).** Asignar una IP estática: 192.168.0.2 con máscara de red 255.255.255.0.

```
PC2> ip 192.168.0.2/24
```

Grabar la configuración en el PC virtual para que no se pierda al pararlo.

```
PC2> save PC2
```

Comprobar que la configuración de la IP y de la máscara es correcta.

```
PC2> show ip
```

Todos los comandos para configurar las tarjetas de red, tanto en sistemas Linux (UbuntuServer), como en los PCs virtuales se encuentran explicados en la práctica 5 "Comandos básicos de configuración de redes en Linux"

Nota 1: La conexión mediante cable de par trenzado de dos equipos se realizará mediante conexión *directa* siempre que uno de los equipos tenga capa de red y el otro no. En cualquier otro caso la conexión será cruzada.

Nota 2: La IP se usa para identificar una red y los equipos terminales y routers que pertenecen a ella. En este caso vamos a usar IPs estáticas de clase C (pertenecientes a IPv4, Protocolo de Internet versión 4) con valores posibles entre 192.0.0.0 y 223.255.255.254. IP = "xxx.xxx.xxx.yyy" Los tres primeros campos identifican a la red y el último campo a los terminales de esa red, en este caso una red puede contener hasta 254 hosts o routers.

La máscara de subred permite distinguir dentro de la dirección IP, los bits que identifican a la red y los bits que identifican al host. Realizando la operación AND entre la dirección IP y la máscara de red se obtiene la dirección de la red.

Nota 3: Para que dos terminales puedan comunicarse entre sí directamente es necesario que formen parte de la misma subred.

Nota 4: Recordad que la comunicación entre capas de red es sólo entre nodos que posean capa de red, es decir dirección IP.

Nota 5: Recordad que la comunicación entre capas de enlace es solo entre nodos físicamente adyacentes para lo cual es necesario conocer la dirección MAC del nodo adyacente.

Protocolo ARP. Protocolo a nivel de enlace que permite traducir direcciones IP en direcciones MAC. Cada nodo IP tiene una tabla ARP que contiene:

- Relación entre la dirección IP y la dirección MAC de los nodos que forman parte de la misma subred LAN.
- Tiempo de vida o TTL. Indica cuando se elimina la correspondencia entre IP y MAC en la tabla.
- El nodo emisor que quiere enviar un datagrama a una IP determinada, busca en su tabla ARP el equivalente MAC.

Paquete ARP: Puede ser un paquete ARP Request (petición ARP). Un nodo difunde (broadcast) un paquete ARP preguntando por la dirección MAC de un equipo con IP determinada.

Puede ser un paquete ARP Replay (respuesta ARP). Un nodo contesta a otro con la dirección MAC preguntada en un paquete previo ARP Request.



Protocolo ICMP: El Protocolo de Mensajes de Control de Internet o ICMP (Internet Control Message Protocol) es el sub protocolo de control y notificación de errores del Protocolo de Internet (IP). Pertenece a la capa de red y se usa para enviar mensajes de error, indicando por ejemplo que un servicio determinado no está disponible o que un router o host no puede ser localizado.

Las herramientas ping y traceroute usan el protocolo ICMP.

Paquete ICMP: Puede ser paquete ICMP Echo Request o paquete ICMP Echo Replay.

Echo Request (Petición eco) es un mensaje de control que se envía a un host con la expectativa de recibir de él un Echo Reply (Respuesta eco). Esto es conocido como Ping y es una utilidad del protocolo ICMP. Todo host debe responder a un Echo Request con un Echo Reply que contenga exactamente los mismos datos que el primero.

Comando ping: El comando ping permite comprobar la conexión entre las capas de red de dos nodos. Simplemente ejecutando ping en la pantalla de comandos seguido de la dirección IP del dispositivo de destino.

1.1 Estudio inicial

Actividad 1. Mostrar las direcciones MAC y direcciones IP de cada equipo.

Actividad 2. Mostrar la tabla ARP de ambos terminales y comprobar como inicialmente está vacía.

Actividad 3. Comprobar que la red está bien configurada y que la comunicación es posible haciendo un ping en la consola del UbuntuServer1 a la dirección IP del PC2.

Si se quiere parar la ejecución del comando ping en máquinas Linux es necesario pulsar Ctrl+c.

Actividad 4. Mostrar la tabla ARP de ambos terminales y comprobar como ahora está llena, es decir ya existe una entrada en la tabla ARP de cada nodo que relaciona su dirección IP con la su dirección MAC.



1.2 Estudio del protocolo ARP y del protocolo ICMP

Veamos en detalle el funcionamiento del protocolo ARP al intentar hacer un ping entre el equipo UbuntuServer1 y el PC2 cuando la tabla ARP del equipo Linux está vacía.

Primero borrar las tablas ARP de cada nodo:

- En UbuntuServer1 es necesario borrar una a una las entradas de la tabla ARP identificadas cada una por la dirección IP.

```
UbuntuServer1> sudo arp -d <IP a borrar>
```

- En el virtual Pc solo es necesario usar clear arp y borra toda la tabla.

```
PC2> clear arp
```

Comprobar que ahora las tablas ARP de ambos nodos están vacías otra vez.

Hacer un ping en la consola del UbuntuServer1 a la dirección IP del PC2 (192.168.0.2), parar la ejecución del comando ping, pulsando CTRL+c y visualizar la pantalla de WireShark, se debe obtener algo parecido a la Figura 2 mostrando la siguiente información:

1. *Lista de paquetes.* En esta tabla se muestran distinta información acerca de los paquetes capturados, las columnas que se muestran por defecto son:
 - a. Número de paquete
 - b. Tiempo de la captura
 - c. IPs de origen y destino
 - d. Protocolo
 - e. Longitud del paquete
 - f. Información adicional
2. *Detalles de los paquetes.* Se muestran los detalles de los paquetes capturados. En estos detalles se pueden encontrar toda la información de las cabeceras de las distintas capas de los protocolos en los que ha ido encapsulándose el paquete.
3. *Bytes de los paquetes.* Información real que está circulando por la red en formato hexadecimal.

En la parte superior de la pantalla podemos utilizar un filtro para visualizar solamente los paquetes que nos interese.

The screenshot shows the Wireshark interface with the following components highlighted by red arrows and labels:

- Filtro de paquetes por protocolo:** Points to the filter bar at the top.
- Lista de paquetes capturados (Packet List):** Points to the table of captured packets.
- Detalle de los paquetes (Packet details):** Points to the expanded details of the selected packet (Frame 154).
- Bytes del paquete (Packet Bytes):** Points to the raw hexadecimal data of the selected packet.

No.	Time	Source	Destination	Protocol	Length	Info
154	1007.968208000	CadmusCo_29:f0:8e	Broadcast	ARP	60	Who has 192.168.0.2? Tell 192.168.0.1
155	1007.968617000	Private_66:68:00	CadmusCo_29:f0:8e	ARP	60	192.168.0.2 is at 00:50:79:66:68:00
156	1007.968714000	192.168.0.1	192.168.0.2	ICMP	98	Echo (ping) request id=0x042d, seq=1/...
157	1007.968955000	192.168.0.2	192.168.0.1	ICMP	98	Echo (ping) reply id=0x042d, seq=1/...
158	1008.970408000	192.168.0.1	192.168.0.2	ICMP	98	Echo (ping) request id=0x042d, seq=2/...
159	1008.970735000	192.168.0.2	192.168.0.1	ICMP	98	Echo (ping) reply id=0x042d, seq=2/...

```

Frame 154: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
  Ethernet II, Src: CadmusCo_29:f0:8e (08:00:27:29:f0:8e), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
  Address Resolution Protocol (request)
  
```

```

0000 ff ff ff ff ff ff 08 00 27 29 f0 8e 08 06 00 01 .....
0010 08 00 06 04 00 01 08 00 27 29 f0 8e c0 a8 00 01 .....
0020 00 00 00 00 00 00 c0 a8 00 02 00 00 00 00 00 .....
0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
  
```

Figura 2. Captura con WireShark.

Esquema de funcionamiento

1. En UbuntuServer1

Capa 3. Capa de red

Se construye un datagrama ICMP Echo Request (Petición eco ICMP) asociado al comando ping con dirección IP de destino 192.168.0.2.

Comprueba que la IP de destino pertenece a la misma subred.

Capa 2. Capa de enlace

El protocolo ARP busca la relación de la dirección IP y MAC del destino en su tabla ARP y no la encuentra.

Almacena en el buffer el paquete ICMP (ping) en espera de conocer la MAC asociada a la dirección IP 192.168.0.2.

Construye un paquete ARP de consulta para la IP del destino 192.168.0.2. ¿Quién es el 192.168.0.2?

Capa 1. Capa física

Difunde (Broadcast) el paquete ARP de consulta por todos los nodos de la red mediante una trama Ethernet. En este caso solo hay uno.

Comprobar como es la trama Ethernet del paquete ARP de consulta en la captura de WireShark, ver Figura 3 y Figura 4, identificar los campos de la trama Ethernet, pinchando en Ethernet II:

- **Preámbulo.** Secuencia fija de ceros y unos para que el receptor sepa que comienza una trama Ethernet. Longitud 8 bytes. Este campo no se visualiza en WireShark, pero existe siempre.
- **MAC de destino.** En este caso al ser un paquete ARP de consulta para obtener la MAC de destino, se usa la MAC de difusión FF.FF.FF.FF.FF.FF. Longitud 6 bytes.
- **MAC del nodo origen.** La MAC de la máquina UbuntuServer1. Longitud 6 bytes.
- **Tipo.** Indica que protocolo de la capa de red se ha encapsulado en el campo de datos de la trama Ethernet. Como se está encapsulando un paquete ARP de consulta en el campo de datos, su valor debe ser 806. Longitud 2 bytes.
- **FCS (frame check sequence)** o secuencia de chequeo de la trama. Se usa el código CRC para detección de errores a nivel de capa de enlace. Longitud 4 bytes. Este campo no se visualiza en WireShark, pero existe siempre.
- **Campo de datos.** Dónde se encapsula el paquete de consulta ARP. Número de bytes variables, en este caso 46 bytes, 28 bytes que contienen el paquete ARP y 18 bytes rellenos con ceros (padding).

Comprobar como es el paquete ARP de consulta encapsulado en el campo de datos en la trama Ethernet, ver Figura 5, pinchando en Address Resolution Protocol, identificar los campos del paquete ARP:

- **Tipo de hardware (HTYPE).** Especifica el tipo de protocolo de enlace que se va a usar. El valor para Ethernet es 1. Longitud 2 bytes.
- **Tipo de protocolo (PTYPE).** Especifica el protocolo de interconexión de redes para las que se destina la petición ARP. Para IPv4, que es el tipo de IPs que estamos usando, tiene el valor 800. Longitud 2 bytes.
- **Longitud Hardware (HLEN).** Longitud de direcciones hardware utilizadas en la capa de enlace. En este caso la dirección MAC. En Ethernet el tamaño de direcciones es de 6 bytes, por ello el valor de este campo de longitud 1 byte es 6.
- **Longitud del Protocolo (PLEN):** Longitud de direcciones utilizadas en el protocolo de capa superior especificado en PTYPE. Usamos IPv4 cuyo tamaño de direcciones es de 4 bytes, el valor de este campo de longitud 1 byte es 4.



- **Operación (OP).** Especifica la operación que el emisor está realizando: 1 si es paquete ARP de consulta y 2 si es paquete ARP de respuesta. En este caso es 1 porque es una pregunta ARP. Longitud 2 bytes.
- **Dirección de hardware del remitente (SHA).** MAC del dispositivo de origen. En un paquete de consulta ARP es la MAC del equipo que pregunta, sin embargo en un paquete ARP de respuesta se almacena la MAC del equipo por el que se pregunta. Longitud 6 bytes.
- **Remitente dirección de protocolo (SPA).** En un paquete de consulta ARP es la IP del equipo que pregunta, sin embargo, en un paquete ARP de respuesta se almacena la IP del equipo por el que se pregunta. Longitud 4 bytes.
- **Dirección de hardware de destino (THA).** MAC del dispositivo de destino. Este campo se ignora en un paquete de consulta ARP pues no se conoce, sin embargo en un paquete ARP de respuesta se almacena la MAC del equipo que pregunta. Longitud 6 bytes.
- **Dirección de protocolo target (TPA).** IP del dispositivo de destino. Este campo en un paquete de consulta ARP contiene la IP del equipo por el que se pregunta su MAC, sin embargo en un paquete ARP de respuesta se almacena la IP del equipo que pregunta. Longitud 4 bytes.

2. En PC2:

Capa 1. Capa física

Se recibe la trama.

Capa 2. Capa de enlace

El nodo acepta la trama que llega pues es una trama de difusión MAC.

El nodo determina que es un paquete ARP de consulta.

El nodo rellena su tabla ARP con la dirección MAC y la IP de UbuntuServer1. Información que acaba de recibir.

El nodo construye un paquete ARP de respuesta a UbuntuServer1, ver Figura 6.

Fijarse como ahora la trama Ethernet ha cambiado los datos de la MAC, e IP de origen y destino. Ahora el PC2 es el origen y UbuntuServer1 el destino. El PC2 conoce ya la IP y la dirección MAC de UbuntuServer1, pues se acaba de actualizar la tabla ARP de PC2

Fijarse como en el campo de datos, este paquete es un paquete ARP de respuesta, Operación = 2 y que ahora el campo THA se ha rellenado con la dirección MAC del equipo PC2. Dirección MAC por la que se preguntaba.

Capa 1. Capa física

Se envía la trama con el paquete de respuesta ARP a UbuntuServer1.

3. En UbuntuServer1:

Capa 1. Capa física

Se recibe la trama del PC2 (paquete ARP respuesta de PC2).

Capa 2. Capa enlace

El nodo determina que es un paquete ARP respuesta del nodo PC2 con destino la MAC de UbuntuServer1. Acepta la trama.

Actualiza la tabla ARP con la MAC y la IP del PC2, por tanto, UbuntuServer1 ya tiene la dirección MAC de PC asociada a la IP 192.168.0.2.

Toma el paquete original ICMP que se quería enviar al 192.168.0.2 del buffer y lo encapsula en una trama Ethernet, ver Figura 7.



Ahora UbuntuServer1 conoce la relación entre la IP del PC2 y su MAC, así que ya sabe que MAC de destino debe poner en la trama Ethernet.

Fijarse ahora que el campo de tipo de la trama Ethernet dice 800, porque está encapsulando en su campo de datos un paquete o datagrama del protocolo IPv4.

Fijarse que el tipo en el paquete ICMP es 8, pues es un paquete Echo Request (Petición eco), ver Figura 8.

Capa 1. Capa física

Envía la trama con el paquete ICMP encapsulado a la dirección MAC de destino correspondiente al PC2.

4. En PC2:

Capa 1. Capa física

Recibe la trama de UbuntuServer1.

Capa 2. Capa enlace

Acepta la trama, pues la dirección MAC de destino coincide con su dirección MAC y quita las cabeceras y colas. Obtiene el datagrama ICMP Echo Request y lo pasa a la capa de red.

Capa 3. Capa de red

Acepta el datagrama ICMP, pues la dirección IP de destino coincide con su dirección IP.

Como ha recibido un paquete ICMP Echo Request, el nodo PC2 está obligado a contestar con un paquete ICMP Echo Replay.

El nodo construye un paquete ICMP Echo Replay con destino UbuntuServer1.

Fijarse que el tipo en el paquete ICMP es 0, pues ahora es un paquete Echo Replay, ver Figura 9.

Se pasa el datagrama ICMP Echo Replay a la capa de enlace.

Capa 2. Capa de enlace

El PC2 busca en su tabla ARP la relación entre la IP del UbuntuServer1 y su MAC, como la tiene sabe que MAC de destino debe poner.

Se encapsula el datagrama ICMP Echo Replay en una trama Ethernet con MAC de destino del UbuntuServer1.

Capa 1. Capa física

Transmite la trama a UbuntuServer1.

5. En UbuntuServer1:

Capa 1. Capa física

Recibe la trama del PC2.

Capa 2. Capa de enlace

Acepta la trama, pues la dirección MAC de destino coincide con su dirección MAC y quita las cabeceras y colas. Obtiene el datagrama ICMP Echo Replay y lo pasa a la capa de red.

Capa 3. Capa de red

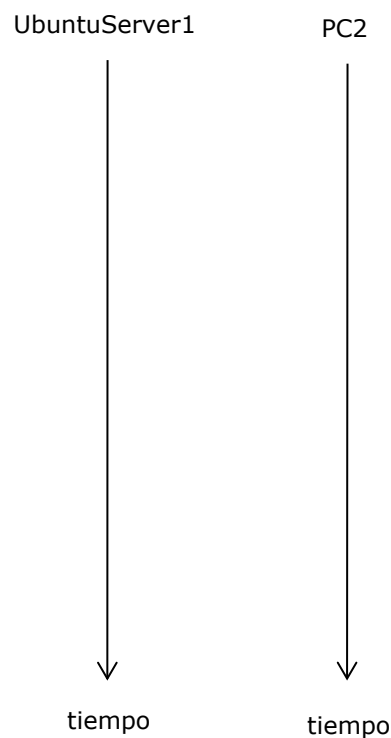
Acepta el datagrama ICMP Echo Replay, pues la dirección IP de destino coincide con su dirección IP.

Fin comunicación. ii Transmisión realizada correctamente !!



Preguntas

- Mostrar las direcciones MAC e IP de cada sistema terminal.
- Analizar con WireShark las tramas Ethernet (paquetes ARP e ICMP) enviados/recibidos asociados al ping realizado. Relacionar claramente el esquema de funcionamiento descrito anteriormente con el tráfico de paquetes por la red.
- Adjuntar los el fichero de WireShark con la captura del enlace (File-> Save).
- Completar el siguiente esquema temporal de la comunicación entre nodos con las tramas Ethernet transmitidas y relacionadas con el protocolo ARP y el protocolo ICMP, así como el tiempo en el que ocurren.



- Identificar los campos (mostrados en las tramas genéricas siguientes) de todas las tramas que han intervenido en la comunicación anterior y de que equipo a que equipo se envía cada trama.

TRAMA Nº						
MAC destino	MAC Origen	tipo	Paquete ARP consulta / respuesta			
			IP destino	MAC destino	IP origen	MAC origen

TRAMA Nº						
MAC destino	MAC Origen	tipo	DATAGRAMA IP ICMP request / replay			
			IP destino	IP origen	Datos	



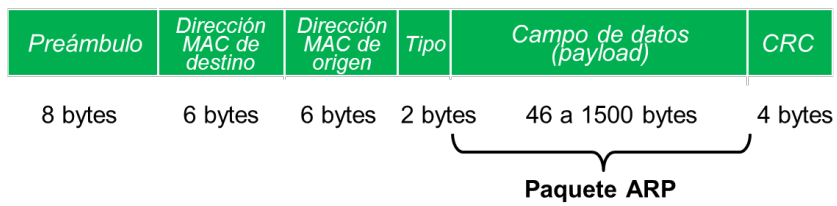
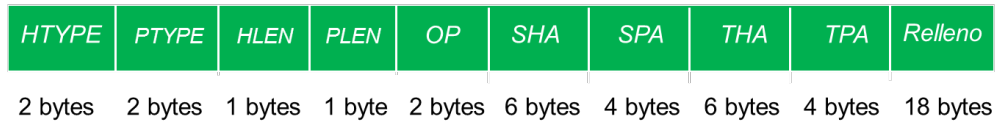
Trama Ethernet:**Paquete ARP:**

Figura 3. Trama Ethernet y campos de un paquete ARP encapsulado en una trama Ethernet.



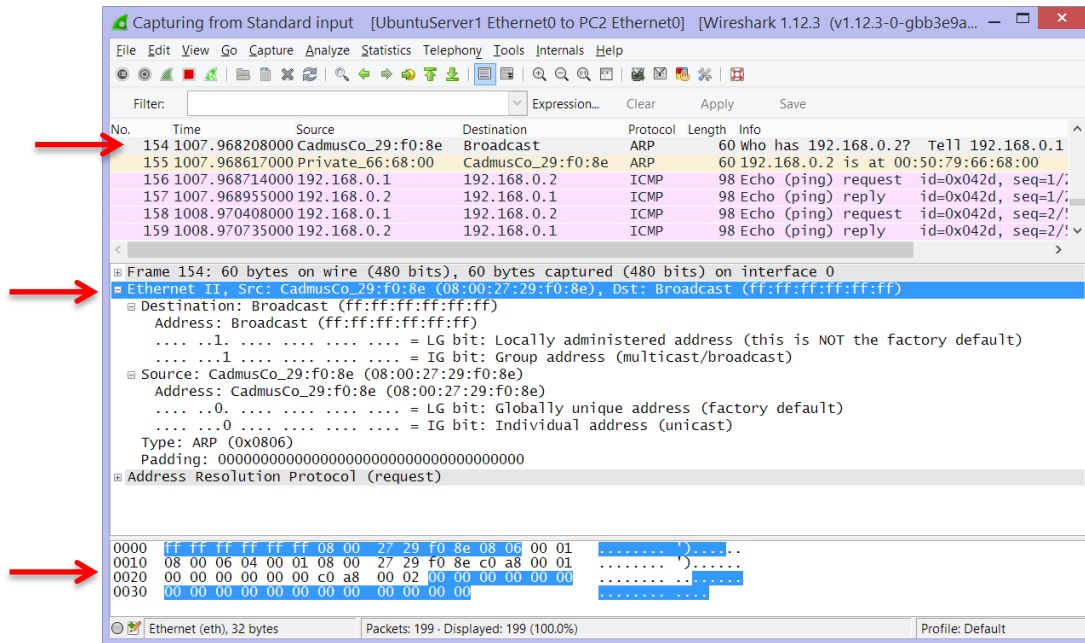


Figura 4. Detalle de los campos de la trama Ethernet en el que se encapsula un paquete de consulta ARP. Paquete transmitido de UbuntuServer1 a PC2.

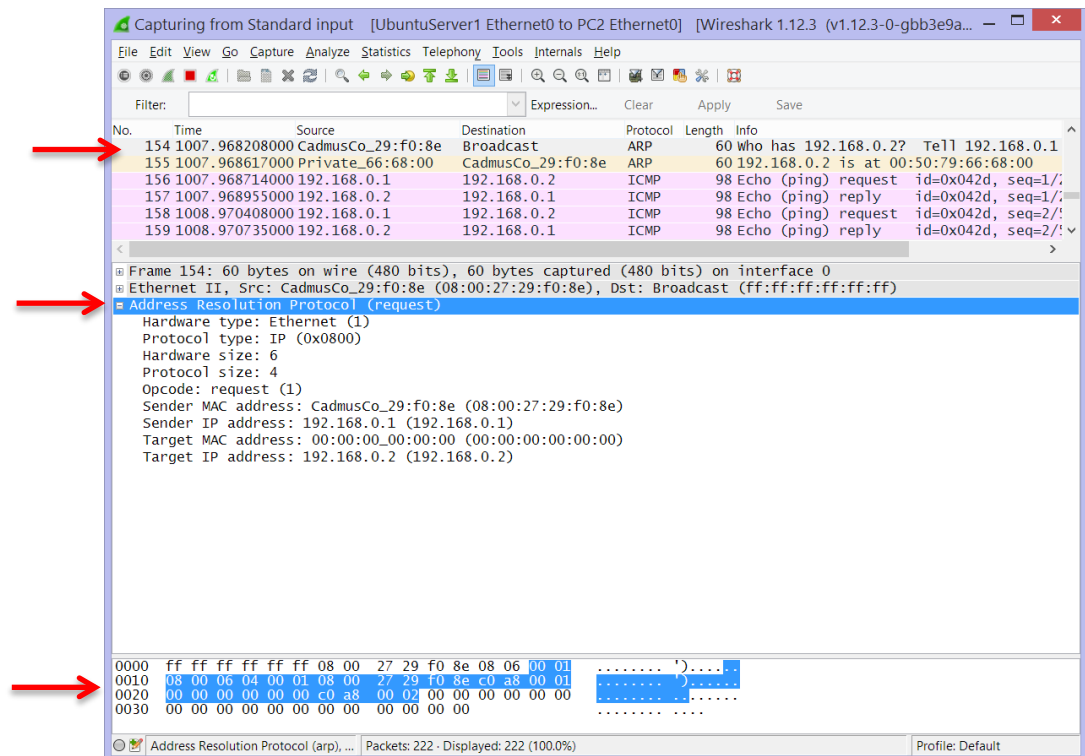


Figura 5. Detalle de los campos del protocolo ARP en un paquete de consulta ARP. Paquete transmitido de UbuntuServer1 a PC2.



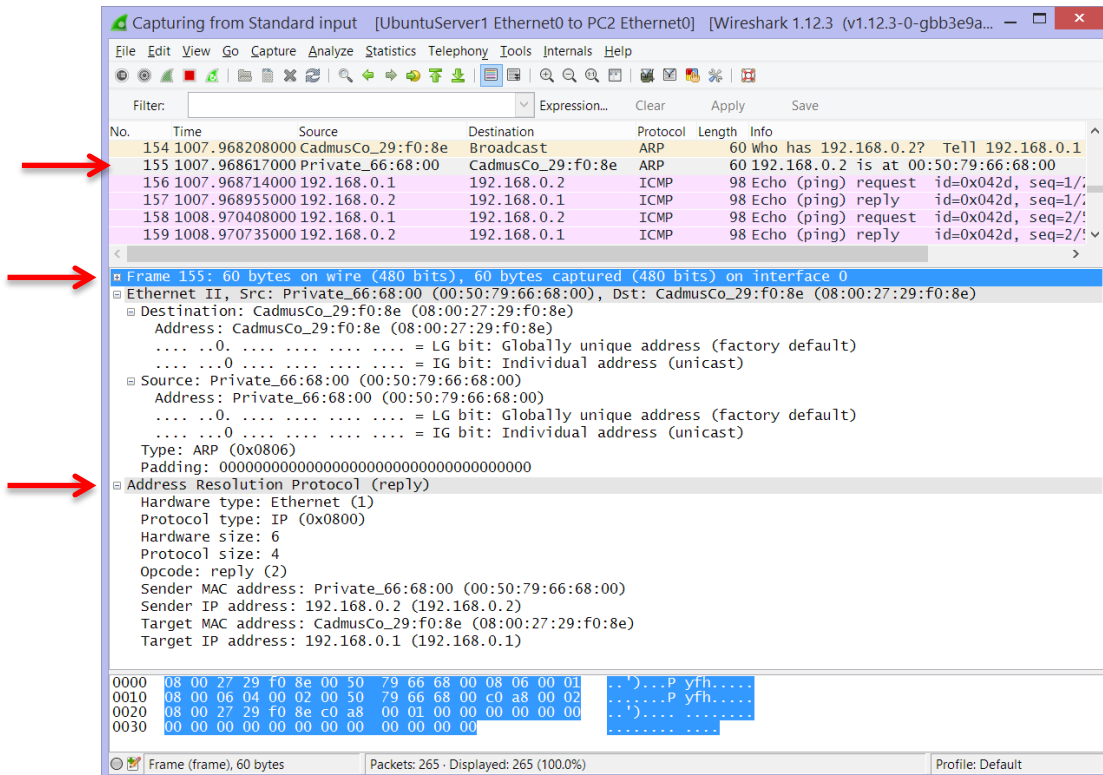


Figura 6. Detalle de los campos de la trama Ethernet en el que se encapsula un paquete de respuesta ARP. Paquete transmitido de PC2 a UbuntuServer1.

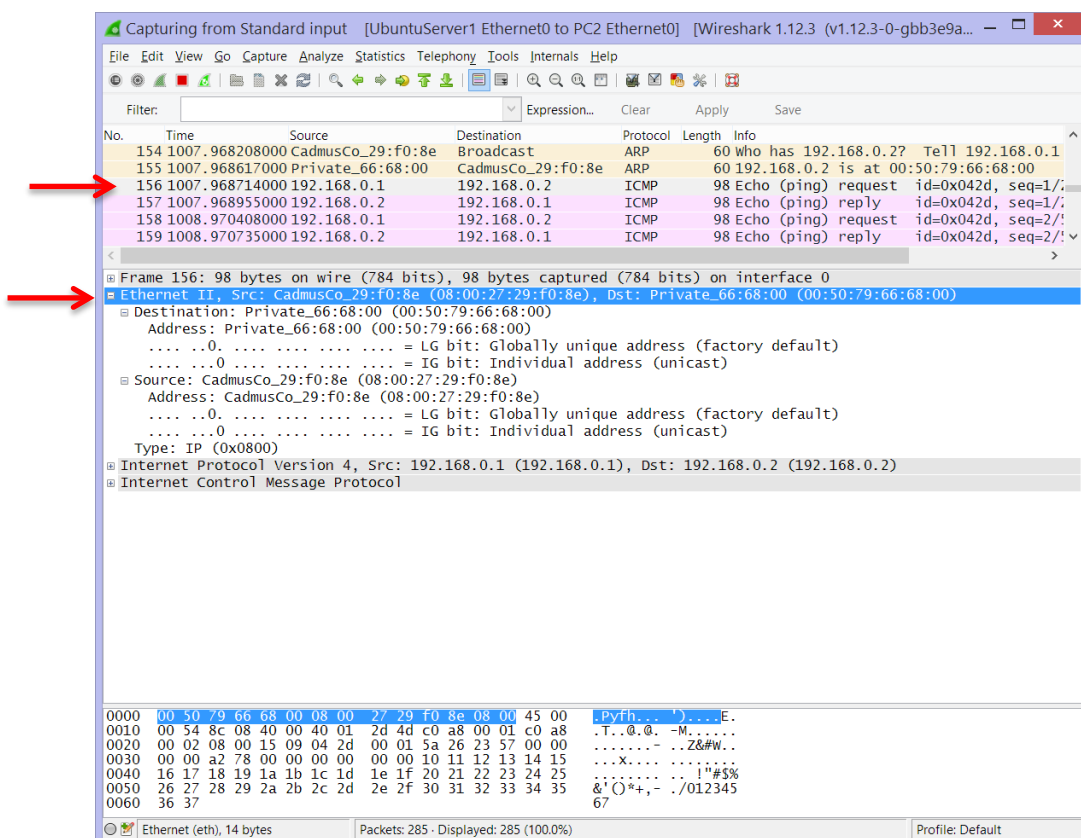


Figura 7. Detalle de los campos de la trama Ethernet en el que se encapsula un paquete Echo Request ICMP. Paquete transmitido de UbuntuServer1 a PC2.



The screenshot shows the Wireshark interface with the following details:

- Packet List:** Frame 156 is selected, showing an ICMP Echo (ping) request from 192.168.0.1 to 192.168.0.2.
- Packet Details:**
 - Ethernet II:** Src: CadmusCo_29:f0:8e (08:00:27:29:f0:8e), Dst: Private_66:68:00 (00:50:79:66:68:00).
 - Internet Protocol Version 4:** Src: 192.168.0.1 (192.168.0.1), Dst: 192.168.0.2 (192.168.0.2).
 - Internet Control Message Protocol:** Type: 8 (Echo (ping) request), Code: 0, Checksum: 0x1509 [correct], Identifier (BE): 1069 (0x042d), Identifier (LE): 11524 (0x2d04), Sequence number (BE): 1 (0x0001), Sequence number (LE): 256 (0x0100).
- Packet Bytes:** Shows the raw data in hexadecimal and ASCII, including the IP and ICMP headers.

Figura 8. Detalle de los campos del paquete Echo Request ICMP. Paquete transmitido de UbuntuServer1 a PC2.

The screenshot shows the Wireshark interface with the following details:

- Packet List:** Frame 157 is selected, showing an ICMP Echo (ping) reply from 192.168.0.2 to 192.168.0.1.
- Packet Details:**
 - Ethernet II:** Src: Private_66:68:00 (00:50:79:66:68:00), Dst: CadmusCo_29:f0:8e (08:00:27:29:f0:8e).
 - Internet Protocol Version 4:** Src: 192.168.0.2 (192.168.0.2), Dst: 192.168.0.1 (192.168.0.1).
 - Internet Control Message Protocol:** Type: 0 (Echo (ping) reply), Code: 0, Checksum: 0x1d09 [correct], Identifier (BE): 1069 (0x042d), Identifier (LE): 11524 (0x2d04), Sequence number (BE): 1 (0x0001), Sequence number (LE): 256 (0x0100), Response time: 0.241 ms.
- Packet Bytes:** Shows the raw data in hexadecimal and ASCII, including the IP and ICMP headers.

Figura 9. Detalle de los campos de la trama Ethernet en el que se encapsula un paquete Echo Reply ICMP. Paquete transmitido de PC2 a UbuntuServer1.



2 Configuración de una red local (LAN) de tipo Ethernet

Crear un proyecto llamado "**Prac06_IV_02.gns3**" en GNS3, ver Figura 10, con una red formada por tres sistemas terminales (un UbuntuServer y dos PCs virtuales) conectados a un conmutador (switch) mediante tarjetas Ethernet. Usar en este caso el **Switch de cisco** que se configuró en la práctica de instalación de las herramientas.

IMPORTANTE: Recordad que estamos utilizando un router con funcionalidades de switch para lo cual se le añadieron una serie de puertos FastEthernet. Por tanto **en esta práctica se deben usar los enlaces desde el FasEthernet1/0 al FasEthernet1/15**, ya que las interfaces FastEthernet0/0 y FastEthernet0/1, son las que se utilizarán para enrutamiento.

Conectar al switch:

- UbuntuServer1 por el enlace **FasEthernet1/1**.
- PC2 por el enlace **FasEthernet1/2**.
- PC3 por el enlace **FasEthernet1/3**.

Antes de arrancar los equipos pinchar con el botón derecho del ratón en el enlace:

- Entre el equipo con UbuntuServer1 y el switch.
- Entre el PC2 y el switch.
- Entre el PC3 y el switch.

Pinchar en Start capture para poder analizar el tráfico en dichos enlaces, se deben abrir tres ventanas de WireShark, cada una asociada a un enlace.

Después arrancar todos los equipos (pinchando con el botón derecho sobre los equipos y pulsando el Icono Start), asignar las siguientes direcciones IP a cada equipo terminal:

- **UbuntuServer (UbuntuServer1)**. Asignar una IP estática: 192.168.0.1 con máscara de red 255.255.255.0.
- **PC Virtual (PC2)**. Asignar una IP estática: 192.168.0.2 con máscara de red 255.255.255.0.
- **PC Virtual (PC3)**. Asignar una IP estática: 192.168.0.3 con máscara de red 255.255.255.0.

Comprobar que las interfaces del switch están activas. Si no lo estuvieran será necesario activarlas, por ejemplo:

```
S1(config)#configure terminal
S1(config)#interface FastEthernet1/1
S1(config-int)#no shutdown
```

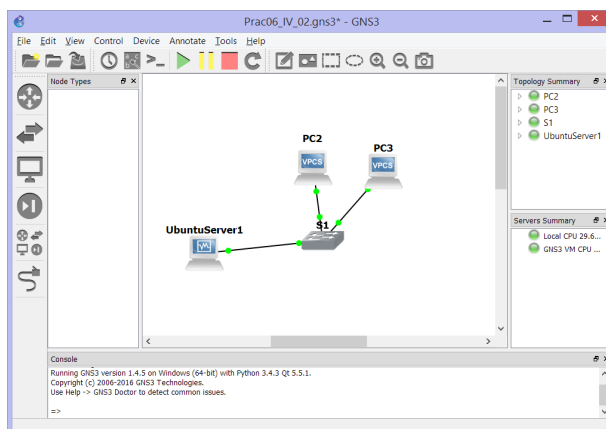


Figura 10. Red LAN propuesta.



2.1 Estudio inicial

Preguntas

- a) Mostrar las direcciones MAC e IP de cada sistema terminal.
- b) Mostrar la tabla MAC del switch y comprobar como inicialmente está vacía, si no borrarla, para configurar el switch es necesario abrir su pantalla de comandos, pinchar encima del switch con el botón derecho y seleccionar Console.
S1> enable
S1# show mac-address-table
Para borrar la tabla MAC
S1# clear mac-address-table
- c) Comprobar que la red está bien configurada y que la comunicación es posible haciendo un ping en la consola de UbuntuServer1 a la dirección IP del PC3.
- d) Mostrar la tabla MAC del switch y comprobar como ya hay dos entradas. ¿A qué interfaces se corresponden? ¿Falta algún terminal por asignar su MAC con la interfaz del Switch? ¿Por qué?
ii Cuidado que las entradas en la tabla del switch se borran automáticamente al poco tiempo !!
- e) Hacer dos ping consecutivos en la consola de UbuntuServer1 uno a la dirección IP del PC3 y otro a la dirección IP del PC2.
- f) Mostrar ahora la tabla MAC del switch y comprobar como ya hay tres entradas. ¿A qué interfaces se corresponden?
- g) ¿Es posible hacer un ping del PC1 al switch? ¿Por qué?
- h) ¿Las interfaces del switch tienen tabla ARP? ¿Por qué?



2.2 Funcionamiento del switch. Autoaprendizaje, filtrado y reenvío

En este apartado vamos a estudiar en detalle el funcionamiento de los switch en una red LAN, en concreto las funciones de autoaprendizaje, filtrado, reenvío y difusión o inundación, que permiten a un switch conocer que nodo (terminal) está conectado a que interfaz. Esta relación entre MAC del terminal con la interfaz del switch a la que está conectado dicho terminal se almacena en un switch en una tabla MAC.

Para estudiar solo el comportamiento del switch y como se rellena dicha tabla MAC, vamos a garantizar que la Tabla MAC del switch está vacía y que el terminal de UbuntuServer1 ya tienen su tabla ARP llena, por lo que no se activará el protocolo ARP para identificar la IP de destino con la MAC de destino.

Para estudiar el comportamiento haremos un ping desde UbuntuServer1 al PC3 (192.168.0.3), comprobando antes que se parte de la siguiente situación:

- **Tabla MAC del switch vacía:** Si no está vacía, la borraremos:

```
Switch> enable
```

```
Switch# clear mac-address-table
```

Comprobar que está vacía:

```
Switch# show mac-address-table
```

- **Tabla ARP del terminal UbuntuServer1 ya contiene cierta información.** Comprobar que tabla ARP de UbuntuServer1 contiene al menos la dirección IP y dirección MAC de PC3.

Si no la tiene añadirla manualmente:

```
UbuntuServer1> arp -s 192.168.0.3 <MAC de PC3>
```

Parar el comando ping (CTRL+c) en UbuntuServer1 después de haber hecho 3 o 4 pings y **parar y grabar cada captura de WireShark** en cada enlace (File -> Save). Identificar claramente el nombre del fichero con equipo origen y equipo destino.

Esquema de funcionamiento ping de UbuntuServer a PC3

1. En UbuntuServer1

Capa 3. Capa de red.

Red. Se construye un datagrama ICMP Echo Request (Petición eco ICMP) asociado al comando ping.

Comprueba que la IP de destino 192.168.0.3 pertenece a la misma subred.

Capa 2. Capa de enlace.

Se encapsula el datagrama ICMP en una trama Ethernet.

Existe una entrada en la tabla ARP que relaciona la IP de destino con la MAC de destino. Se añade dicha MAC a la trama Ethernet.

Capa 1. Capa física.

Transmite la trama con el paquete ICMP Echo Request al switch



2. En Switch (S1)

Capa 1. Capa física

Recibe la trama con el paquete ICMP Echo Request de UbuntuServer1.

Capa 2. Capa de enlace.

No existe entrada en la tabla MAC con la dirección MAC de UbuntuServer1 y la interfaz a la que está conectado UbuntuServer1. Almacena la MAC de UbuntuServer1 con el enlace por el que llegó (FastEthernet1/1) en la tabla MAC. *Función Autoaprendizaje.*

No existe entrada en la tabla MAC de la dirección MAC del PC3 y la interfaz conectada a PC3. El switch manda una copia de la trama que ha llegado por todas las interfaces distintas a la que llegó. *Función inundación o difusión.*

Capa 1. Capa física.

Transmite la trama con el paquete ICMP Echo Request por el resto de interfaces.

3. En PC2:

Capa 1. Capa física

Recibe la trama con el paquete ICMP Echo Request del switch.

Capa 2. Capa de enlace.

La dirección MAC del PC2 no se corresponde con la dirección MAC destinataria de la trama. La trama no va dirigida al PC2, la trama se descarta.

4. En PC3

Capa 1. Capa física

Recibe la trama con el paquete ICMP Echo Request del switch.

Capa 2. Capa de enlace.

Acepta la trama, pues la dirección MAC de destino coincide con su dirección MAC y quita las cabeceras y colas. Obtiene el datagrama ICMP Echo Request y lo pasa a la capa de red.

Capa 3. Capa de red

Acepta el datagrama ICMP Echo Request, pues la dirección IP de destino coincide con su dirección IP.

Como ha recibido un paquete ICMP Echo Request, el nodo PC3 está obligado a contestar con un paquete ICMP Echo Replay.

El nodo construye un paquete ICMP Echo Replay con destino UbuntuServer1.

Se pasa el datagrama ICMP Echo Replay a la capa de enlace.

Capa 2. Capa de enlace.

Se encapsula el datagrama ICMP Echo Replay en una trama Ethernet con MAC de destino UbuntuServer1.

El PC2 busca en su tabla ARP la relación entre la IP de UbuntuServer1 y su MAC, como la tiene sabe que MAC de destino debe poner.

Capa 1. Capa física.

Transmite la trama con el paquete ICMP Echo Replay al switch.



5. En Switch:**Capa 1. Capa física**

Recibe la trama del PC3 con destino MAC de UbuntuServer1.

Capa 2. Capa de enlace.

No existe entrada en la tabla MAC de la dirección MAC del PC3 y la interfaz conectada a PC3. Almacena la MAC de PC3 con el enlace por el que llegó (FastEthernet1/3) en la tabla MAC. *Función Autoaprendizaje.*

Capa 2. Capa de enlace.

Si existe en la tabla MAC una entrada que relaciona la interfaz de salida con la MAC de UbuntuServer1. Se envía la trama por el enlace correspondiente FastEthernet1/1. *Función reenvío.*

Capa 1. Capa física.

Transmite la trama con el paquete ICMP Echo Replay a UbuntuServer1.

6. En UbuntuServer1:**Capa 1. Capa física.**

Recibe la trama

Capa 2. Capa de enlace.

Acepta la trama, pues la dirección MAC de destino coincide con su dirección MAC y quita las cabeceras y colas. Obtiene el datagrama ICMP Echo Replay y lo pasa a la capa de red.

Capa 3. Capa de red

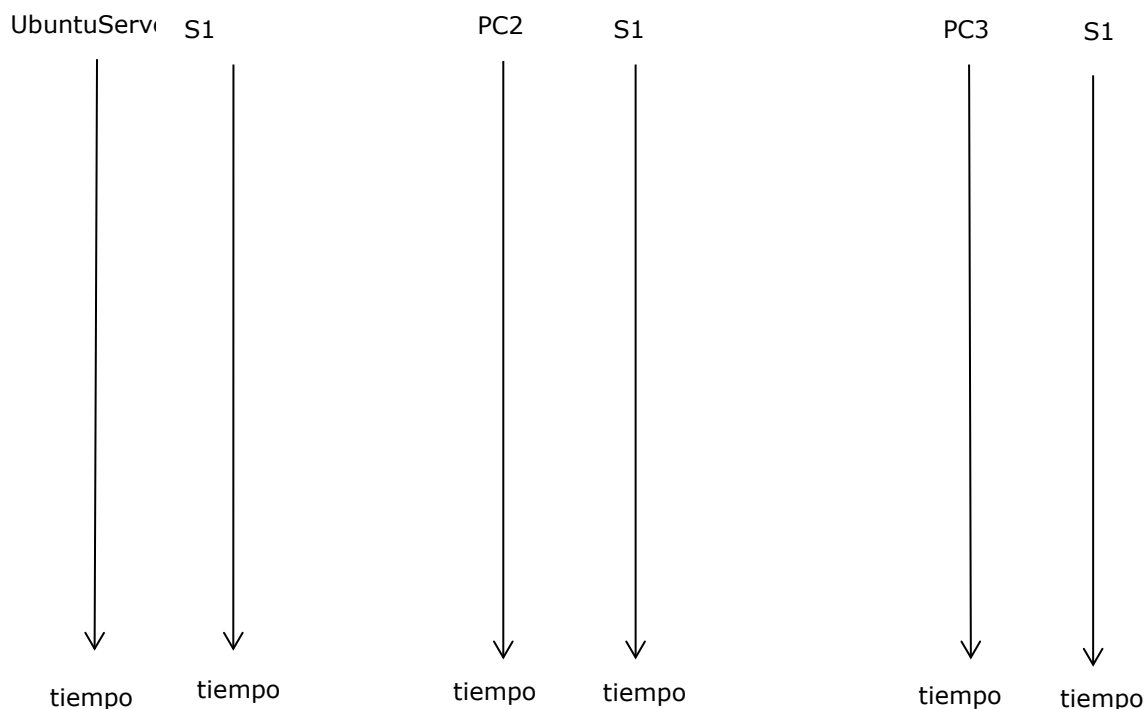
Acepta el datagrama ICMP Echo Replay, pues la dirección IP de destino coincide con su dirección IP.

Fin comunicación. ii Transmisión realizada correctamente !!



Preguntas

- Mostrar las direcciones MAC e IP de cada sistema terminal.
- Analizar con WireShark los paquetes ICMP enviados/recibidos por los tres enlaces asociados al ping realizado. Relacionar claramente el esquema de funcionamiento descrito anteriormente con el tráfico de paquetes por la red.
- Adjuntar los tres ficheros de WireShark con la captura de cada enlace (File-> Save).
- Rellenar un esquema temporal de la comunicación entre nodos con las tramas Ethernet transmitidas y relacionadas con el protocolo ARP y el protocolo ICMP entre el PC1 y el switch S1, entre el PC2 y el switch S1 y entre el PC3 y el switch S1.



- Identificar los campos (mostrados en las tramas genéricas siguientes) de todas las tramas que han intervenido en la comunicación anterior y de que equipo a que equipo se envía cada trama.

TRAMA Nº						
MAC destino	MAC Origen	tipo	Paquete ARP consulta / respuesta			
			IP destino	MAC destino	IP origen	MAC origen

TRAMA Nº						
MAC destino	MAC Origen	tipo	DATAGRAMA IP ICMP request / replay			
			IP destino	IP origen	Datos	

- ¿A qué se corresponde el protocolo CDP que te habrá salido en las capturas?



V Estudio de redes virtuales de área local (VLAN)

1 Creación de una red LAN

Actividad 1. Crear un proyecto llamado **"Prac06_V_01.gns3"**. En este fichero configurar, utilizando GNS3 una red con todos los equipos conectados a un switch con las IPs que aparecen en la Figura 11, todas con máscara 255.255.255.0 o /24. Usar en este caso el **Switch de GNS3 que trae por defecto (Ethernet Switch)**.

Equipo	Tipo	IP
PC1	VPCS	192.168.0.1/24
PC2	VPCS	192.168.0.2/24
PC3	VPCS	192.168.0.3/24
PC4	VPCS	192.168.0.4/24
PC5	VPCS	192.168.0.5/24
PC6	VPCS	192.168.0.6/24
Switch	Ethernet Switch (Switch de GNS3)	-

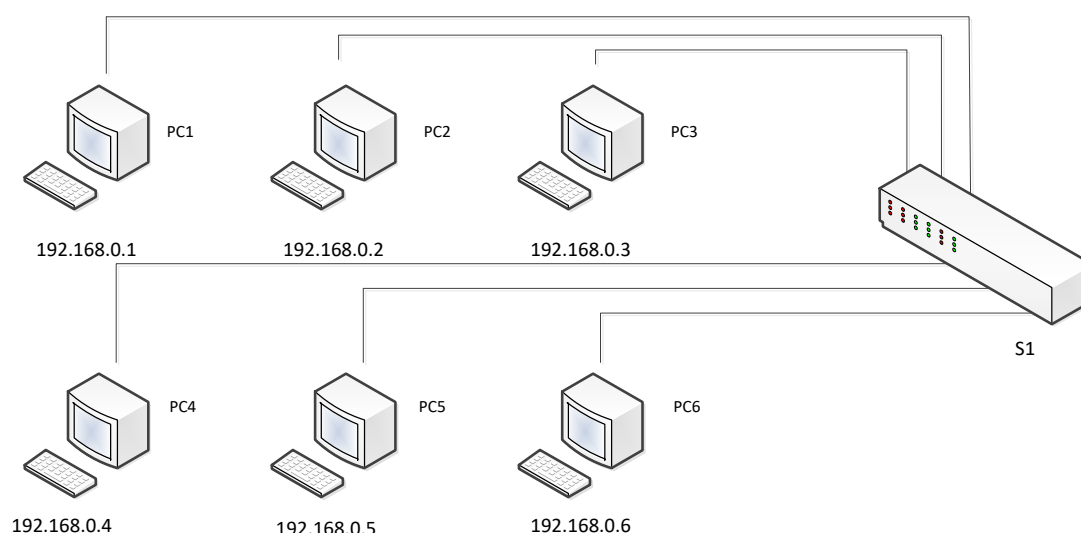


Figura 11. Red propuesta.

Actividad 2. Hacer ping desde cualquier equipo a otro. ¿Por qué se puede hacer ping? Mostrar una captura del resultado de hacer ping entre el PC1 y el PC6.

En ocasiones es necesario parar y rearrancar la simulación para conseguir el buen funcionamiento de la misma, por ello se recomienda salvar las configuraciones de todos los VPCS (PCs virtuales de GNS3).

Actividad 3. ¿En qué VLAN del switch aparecen conectados todos los equipos por defecto?

Los switches básicos de GNS3 (Ethernet Switch) no traen consola y no se puede visualizar su tabla MAC o su tabla VLAN. Solo se puede hacer doble click en el icono del switch y se muestra como está configurado.



2 Creación de VLANs

Con la división de los equipos en distintas VLANs, se consigue crear distintas redes de área local virtuales dentro de una misma red física. Esto flexibiliza en gran medida el uso de los conmutadores, limita los dominios de difusión y hace los sistemas más seguros.

Para crear VLANs en los switches de GNS3 se debe pinchar dos veces en el icono del switch, seleccionar el puerto y cambiar el número de la VLAN, ver Figura 12.

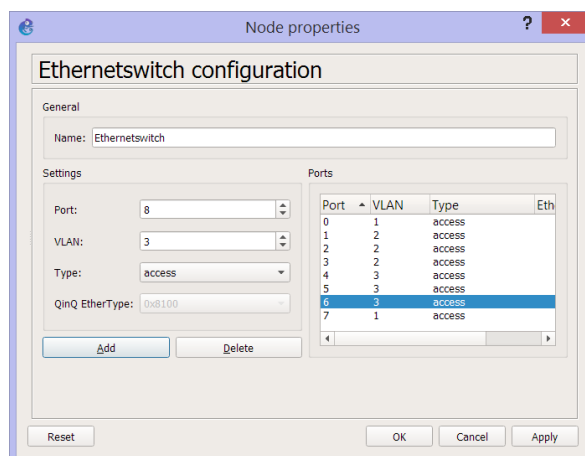


Figura 12. Configuración de VLANs en switches de GNS3.

Actividad 4. Guardar el proyecto anterior como "**Prac06_V_02.gns3**" y dividir el problema desarrollado en la actividad 1 en 2 VLANs (vlan 2 y vlan 3) tal y como se muestra en la Figura 13. Se utilizarán los mismos equipos que en la actividad anterior. Una vez hecha la subdivisión en distintas VLANs, tratar de hacer ping entre equipos de distintas redes. ¿Se puede hacer ping? ¿Por qué?

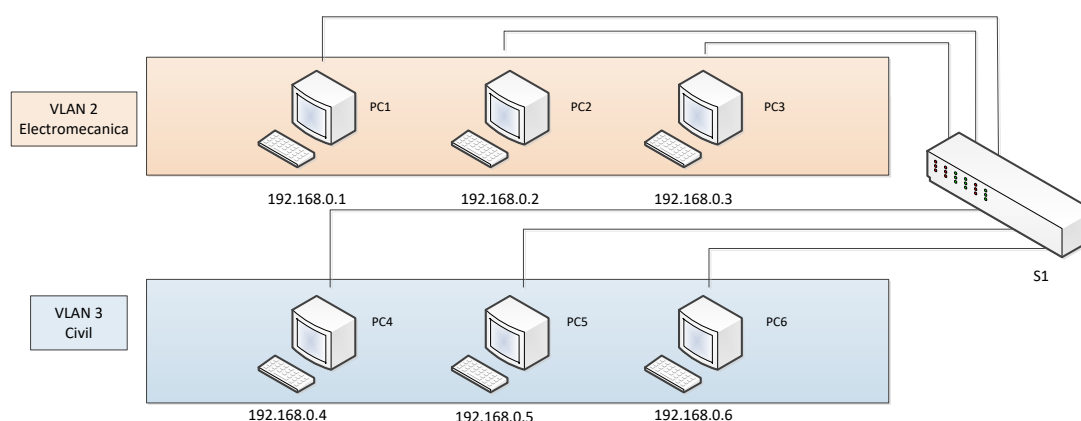


Figura 13. Red propuesta.

Actividad 5. Mostrar la configuración de VLAN del conmutador, donde veremos cómo se asigna a cada interfaz la VLAN a la que pertenece.



2.1 Enlaces Troncales

En ocasiones se dispone de distintos conmutadores y se quiere que algunos de los equipos conectados a distintos conmutadores pertenezcan a la misma red virtual. Para solucionar esto se recurre a la técnica de troncalización VLAN (Trunking).

Para ello, se define que determinados puertos del conmutador sean puertos troncales y se utilizan para comunicar los conmutadores entre sí. Los puertos troncales permiten el paso de las tramas de todas las VLANs.

Suponer que en este caso se dispone de 2 conmutadores de la capa de enlace (switches) cada uno conectado a una fila de ordenadores, pero que los dos primeros PCs de cada fila pertenecen a la VLAN de Electromecánica y el tercer PC a la VLAN de civil, según el esquema de la Figura 15.

Para definir un enlace como troncal en el switch de GNS3 se debe seleccionar dicho enlace haciendo doble click en el icono del switch y después en Type seleccionar dot1q, ver Figura 14.

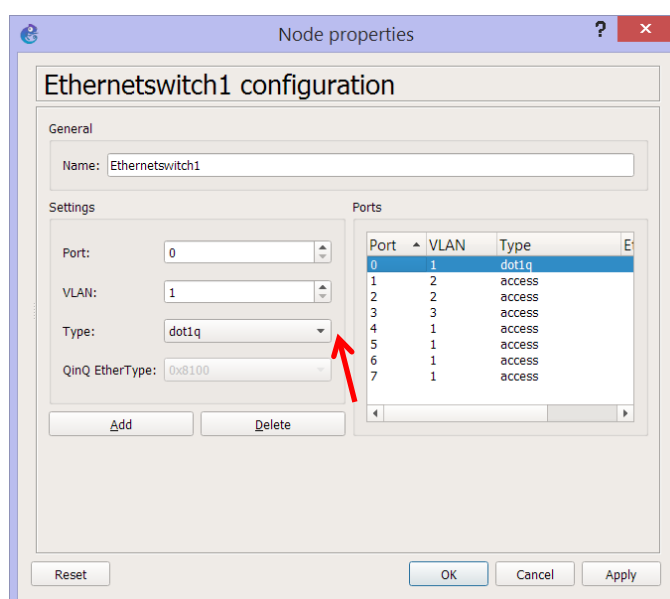


Figura 14. Configuración de enlaces troncales en switches de GNS3.

Actividad 7. Guardar el proyecto "Prac06_V_02.gns3" como "**Prac06_V_02_1.gns3**" y configurar las VLANs tal como se muestra en la Figura 15. Configurar los switches para permitir la troncalización. Probar mediante comandos ping la alcanzabilidad de los equipos de las distintas VLANs.

Mostrar la respuesta de ping desde el PC1 al PC6 y desde el PC1 al PC3.

Activar WireShark en el enlace troncal entre los dos switches e identificar los campos de la trama Ethernet 802.1 Q, ver Figura 16, cuando se hace ping entre el PC1 y el PC4, y cuando se hace ping entre el PC3 y el PC5.

Mostrar la configuración de las VLANs del switch S2.



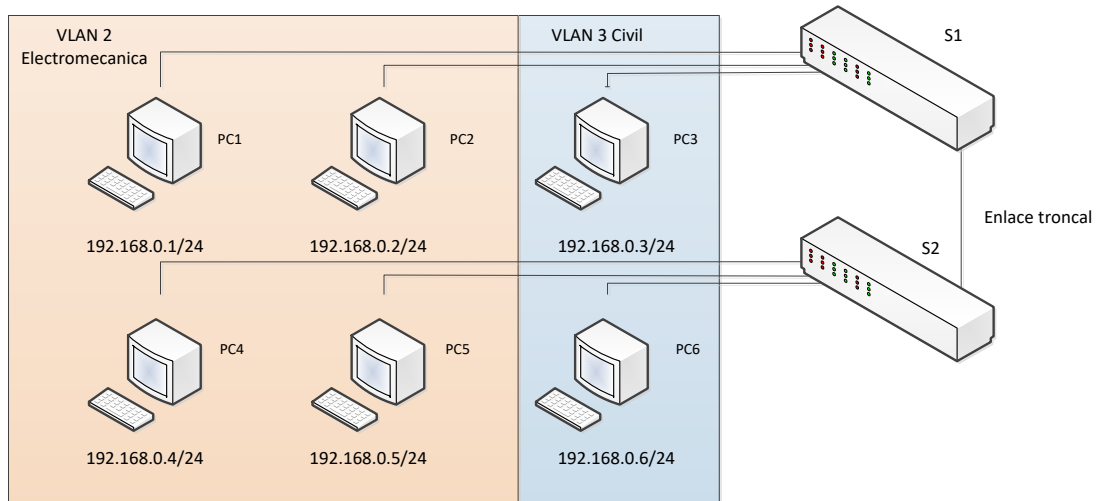


Figura 15. Red propuesta enlaces troncales.

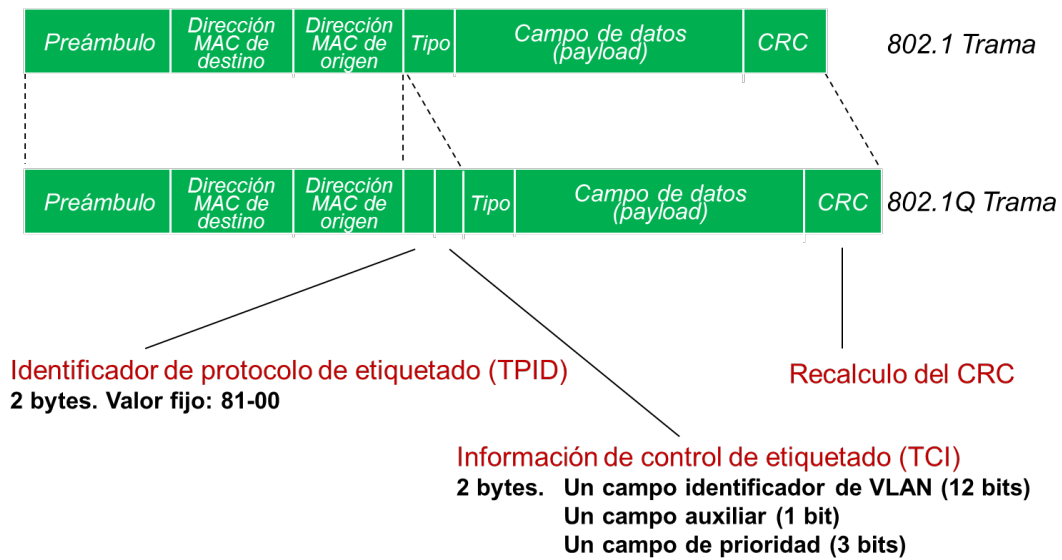


Figura 16. Trama Ethernet 802.1 y trama Ethernet 802.1Q ampliada para la comunicación mediante enlaces troncales.



VI Estudio avanzado de redes LAN con software de Cisco

En este apartado de la práctica se va a estudiar la misma red LAN y VLAN planteadas en la sección anterior V pero usando un Switch de Cisco que incluye un sistema operativo típico de un switch real con muchas más funcionalidades que el Switch básico de GNS3. Por tanto, se aprenderá a configurar un switch de Cisco para crear LANs y VLANs, así como la comunicación entre varias VLANs, la configuración y acceso remoto a switches y el enrutado de VLANs con el exterior a través de un router.

IMPORTANTE: Recordad que estamos utilizando un router con funcionalidades de switch para lo cual se le añadieron una serie de puertos FastEthernet. Por tanto, **en esta práctica se deben usar los enlaces desde el FasEthernet 1/0 al FasEthernet 1/15**, ya que las interfaces Fastethernet 0/0 y 0/1, se utilizan para enrutamiento.

Se recomienda guardar las configuraciones de los equipos y del switch para no perderlas cada vez que se detenga la emulación.

1 Creación de una red LAN

Actividad 1. Crear un proyecto llamado **"Prac06_VI_01.gns3"**. En este fichero configurar, utilizando GNS3 una red con todos los equipos conectados a un switch con las IPs que aparecen en la Figura 17, todas con máscara 255.255.255.0 o /24. Usar en este caso el **Switch de cisco** que se configuró en la práctica de instalación de las herramientas. Se puede partir del proyecto ya realizado anteriormente "Prac06_V_01.gns3", renombrarlo y sustituir el Switch de GNS3 por el Switch de Cisco.

Equipo	Tipo	IP
PC1	VPCS	192.168.0.1/24
PC2	VPCS	192.168.0.2/24
PC3	VPCS	192.168.0.3/24
PC4	VPCS	192.168.0.4/24
PC5	VPCS	192.168.0.5/24
PC6	VPCS	192.168.0.6/24
Switch	Switch (Switch de Cisco)	-

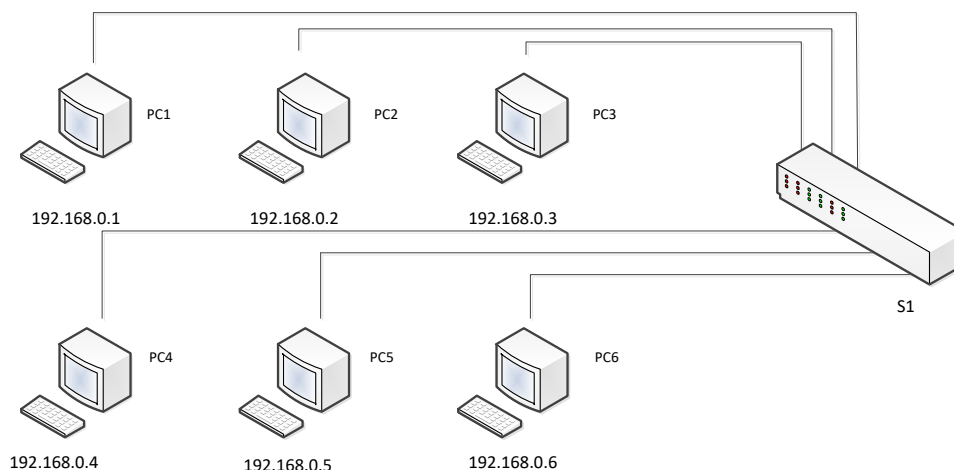


Figura 17. Red propuesta.



Actividad 2. Hacer ping desde cualquier equipo a otro. ¿Por qué se puede hacer ping? Mostrar una captura del resultado de hacer ping entre el PC1 y el PC6.

En ocasiones es necesario parar y rearrancar la simulación para conseguir el buen funcionamiento de la misma, por ello se recomienda salvar las configuraciones.

En caso de que no se pudiera hacer ping, comprobar que las interfaces del switch están activas. Si no lo estuvieran será necesario activarlas, por ejemplo:

```
S1(config)#configure terminal
S1(config)#interface FastEthernet1/0
S1(config-int)#no shutdown
```

Actividad 3. ¿En qué VLAN del switch aparecen conectados todos los equipos por defecto? Mostrar la tabla VLAN (show vlan-switch) del switch

¿Qué equipos han enviado tramas? ¿Por qué? Mostrar la tabla MAC (show mac-address-table) del switch e indicar a que interfaces del switch están conectados.

2 Creación de VLANs

Con la división de los equipos en distintas VLANs, se consigue crear distintas redes de área local virtuales dentro de una misma red física. Esto flexibiliza en gran medida el uso de los conmutadores, limita los dominios de difusión y hace los sistemas más seguros.

Se van a crear ahora 2 VLANs y se van a asignar los siguientes nombres y los siguientes equipos a esas VLANs.

- Electromecanica (vlan 2) -> Fila 1
- Civil (vlan 3) -> Fila 2

Los comandos para crear y nombrar las VLANs en los conmutadores CISCO, son por ejemplo:

```
S1# configure terminal
S1(config)# vlan 2
S1(config-vlan)# name Electromecanica
```

Para asignar las interfaces a las VLANs, se utilizarán los siguientes comandos (por ejemplo):

```
S1(config)#interface FastEthernet1/2
S1(config-if)#switchport access vlan 2
S1(config-if)#exit
S1(config)#interface FastEthernet1/5
S1(config-if)#switchport access vlan 3
```

En ocasiones se desactivan las VLANs cuando se reinicia GNS3, para volver a activarlas es necesario escribir:




```
S1(config)#vlan 2
S1(config-vlan)#state active
```

Las interfaces conectadas a las VLANs sí que se mantienen, por lo que no es necesario volver a configurarlas

Actividad 4. Guardar el proyecto anterior como **"Prac06_VI_02.gns3"** y dividir el problema desarrollado en la actividad 1 en las 2 VLANs descritas tal como se muestra en la Figura 18. Se utilizarán los mismos equipos que en la actividad anterior. Una vez hecha la subdivisión en distintas VLANs, tratar de hacer ping entre equipos de distintas redes virtuales. ¿Se puede hacer ping? ¿Por qué? Mostrar una captura del comando *show vlan-switch*.

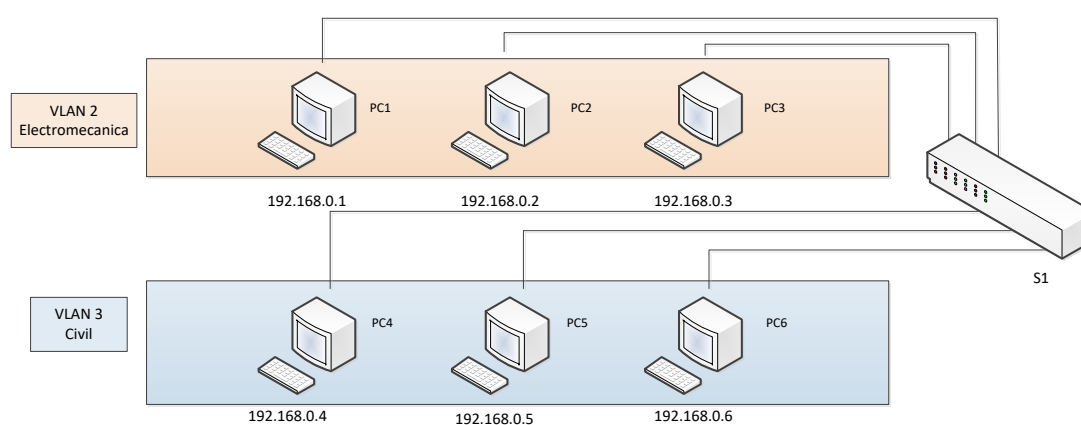


Figura 18. Red propuesta.

Actividad 5. Mostrar la tabla de direcciones MAC (*show mac-address-table*) del switch, donde veremos cómo se asigna cada interfaz con la MAC del equipo que está conectado.

```
S1
S1#show mac-address-table
Destination Address  Address Type  VLAN  Destination Port
-----
c201.07b9.0000      Self          1     Vlan1
0050.7966.6800      Dynamic       2     FastEthernet1/1
0050.7966.6802      Dynamic       2     FastEthernet1/3
S1#
```

El tiempo que se mantienen las asignaciones de la tabla mac en el switch es pequeño, por lo que se pierde rápidamente. Se puede aumentar este tiempo escribiendo:

```
S1(config)# mac-address-table aging-time 1000000
```

Esto asigna las tablas durante 1000000 s. No obstante, por alguna razón, en GNS3 no se mantienen las tablas MAC durante mucho tiempo, así que a pesar de fijar el tiempo de 1000000 s se borran rápidamente.

Para borrar la tabla escribir:

```
S1#clear mac-address-table
```



2.1 VLAN de administración

En los conmutadores es posible definir VLANs de administración, que nos permiten acceder a las capacidades de administración del Switch. Es necesario asignar una dirección IP a esta VLAN de forma que podamos acceder a la misma mediante Telnet, SSH, HTTP,... desde cualquier equipo conectado a las interfaces pertenecientes a esta VLAN (en nuestro caso accederemos vía Telnet). Para poder definir la interfaz de administración es necesario seguir los siguientes pasos:

1. Definir el número máximo de conexiones concurrentes permitido cuando accedemos vía telnet.

```
S1(config)#line vty 0 4
```

2. Configurar una contraseña de acceso.

```
S1(config-line)#pass 123456
S1(config-line)#login
S1(config-line)#exit
```

3. Definir la VLAN de administración, asignarle una dirección IP y darla de alta.

```
//Definir igual que se mostró con anterioridad la interfaz
S1(config)# vlan 99
S1(config-vlan)#name admin
S1(config)# interface vlan 99
S1(config-if)#ip address 192.168.20.20 255.255.255.0
S1(config-if)#no shutdown
```

4. Incluir las interfaces del switch que se desee que sean de administración en la VLAN.

```
S1(config-if)#interface FastEthernet1/15
S1(config-if)#switchport access vlan 99
```

5. Para visualizar que la vlan de administración (vlan 99) está correctamente configurada escribir:

```
S1#show ip interface vlan 99
```

Una vez configurada la VLAN de administración, es posible conectar un PC a la interfaz en la que se haya configurado la VLAN de administración. En nuestro caso utilizaremos una máquina virtual Ubuntu (los VPCS no permiten el uso de telnet). Desde ese PC se hará un telnet a la dirección IP por el puerto correspondiente. Por defecto el puerto que se utiliza es el puerto **23**. Por tanto desde este equipo remoto se podría configurar el switch.



```

ale.jandro@ubuntuserver:~$ telnet 192.168.20.20 23
Trying 192.168.20.20...
Connected to 192.168.20.20.
Escape character is '^]'.

User Access Verification

Password:
Router>

```

Nota: Hay que tener en cuenta que, por seguridad, no se permite que en las conexiones remotas el paso a modo privilegiado pueda hacerse sin contraseña. Si queremos poder acceder a la configuración en modo privilegiado del switch utilizando telnet, tenemos que habilitar una contraseña para el paso a modo privilegiado. Esto se hará escribiendo en el switch, Siendo 1234 la contraseña² seleccionada:

```
S1(config)#enable secret 1234
```

Actividad 6. Guardar el proyecto anterior como **"Prac06_VI_02_1.gns3"** y añadir al switch una VLAN de administración con el número 99. Incluir una interfaz del conmutador en esa VLAN y conectar un equipo a la misma tal cómo se muestra en la Figura 19, de manera que se pueda administrar el switch desde un equipo. Como equipo para probar la conexión telnet utilizar una máquina virtual Ubuntu.

Mostrar una captura de la conexión telnet desde el equipo remoto.

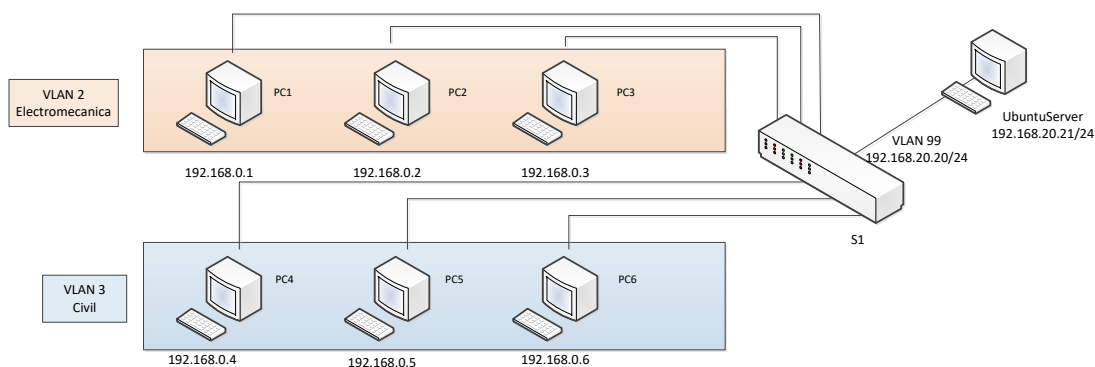


Figura 19. Red propuesta.

2.2 Enlaces Troncales

En ocasiones se dispone de distintos conmutadores y se quiere que algunos de los equipos conectados a distintos conmutadores pertenezcan a la misma red virtual. Para solucionar esto se recurre a la técnica de troncalización VLAN (Trunking).

Para ello, se define que determinados puertos del conmutador sean puertos troncales y se utilizan para comunicar los conmutadores entre sí. Los puertos troncales permiten el paso de las tramas de todas las VLANs.

Suponer que en este caso se dispone de 2 conmutadores de la capa de enlace (switches) cada uno conectado a una fila de ordenadores, pero que los dos primeros PCs de cada fila pertenecen a la VLAN de Electromecánica y el tercer PC a la VLAN de civil, según el esquema de la Figura 20.

Para definir un enlace como troncal, hay que escribir los siguientes comandos, por ejemplo para el conmutador S1 (lo mismo para el switch S2):

² Para más información: <http://aprenderedes.com/2006/08/configuracion-de-contrasenas-de-consola-auxiliar-y-telnet/>



```
S1(config)#interface FastEthernet1/0
S1(config-if)#switchport mode trunk
S1(config-if)#no shutdown
```

Se puede comprobar en el switch que interfaz está siendo usada como enlace troncal de la siguiente manera:

```
S1#show interfaces trunk
```

Nota: puede ser necesario reiniciar el switch, para ello previamente guardar la configuración para que no se pierda al reiniciarlo.

Actividad 7. Partir del proyecto anterior “Prac06_VI_02.gns3” y guardarlo como “Prac06_VI_02_2.gns3”. Configurar en los dos switches las VLANs tal y como se muestra en la Figura 20, además configurar en ambos switches el enlace común para permitir la troncalización, por ejemplo el FastEthernet1/0.

- Electromecanica (vlan 2) -> Switch1: PC1 y PC2. Switch 2: PC4 y PC5
- Civil (vlan 3) -> Switch1: PC3. Switch 2: PC6

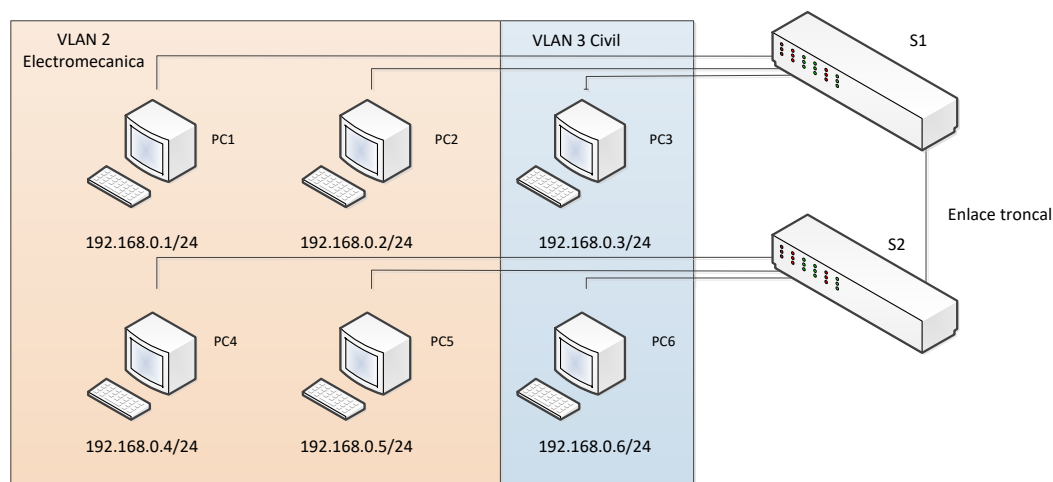


Figura 20. Red propuesta enlaces troncales.

Probar mediante comandos ping la alcanzabilidad de los equipos de las distintas VLANs.

Mostrar la respuesta de ping desde el PC1 al PC6 y desde el PC1 al PC3.

Activar WireShark en el enlace troncal entre los dos switches e identificar los campos de la trama Ethernet 802.1 Q, ver Figura 21, cuando se hace ping entre el PC1 y el PC4, y cuando se hace ping entre el PC3 y el PC5.

Mostrar la configuración de las VLANs del switch S2.



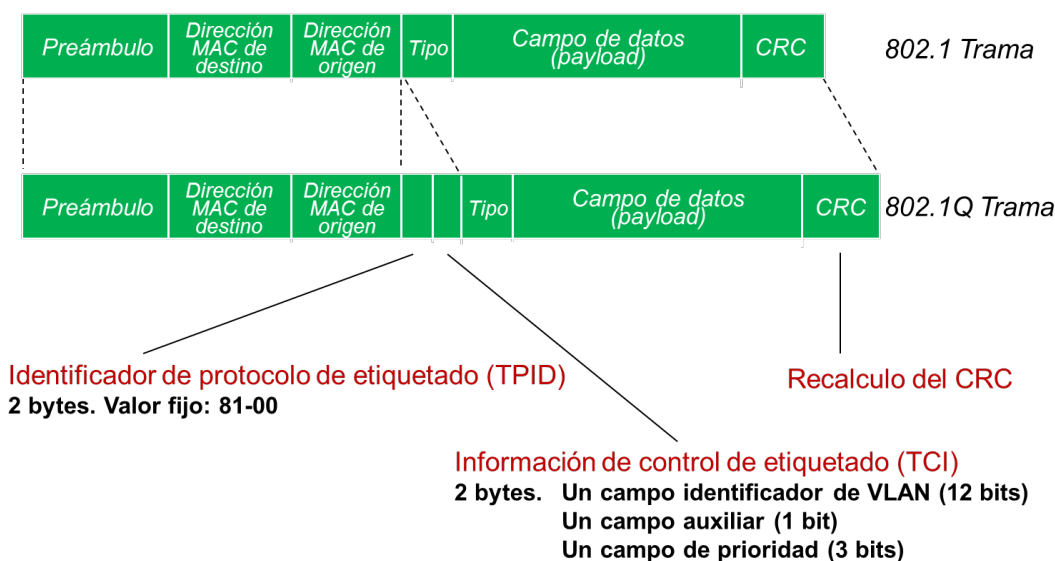


Figura 21. Trama Ethernet 802.1 y trama Ethernet 802.1Q ampliada para la comunicación mediante enlaces troncales.

2.3 Enrutamiento entre VLANs

Los switches tradicionales suelen trabajar únicamente a nivel de la capa de enlace, llamándose también conmutadores de nivel 2. Existen conmutadores de nivel 3, que soportan el enrutamiento entre subredes y VLANs, pero en esta práctica vamos a trabajar con conmutadores de nivel 2, que son los más habituales en redes pequeñas y son los que están disponibles en GNS3. Este tipo de conmutadores no permite el enrutamiento de los paquetes, por lo que los paquetes que procedan de una VLAN y tengan como destino otra VLAN de la subred, no llegarán al destino.

En ese caso, para poder comunicar dos VLANs diferentes, tenemos que colocar un router entre las mismas. El router se conecta a tantas interfaces del conmutador, como distintas VLANs queramos enrutar. A continuación, se configuran las interfaces del conmutador en las que está conectado el router, a las VLANs correspondientes.

En el ejemplo que se muestra en la Figura 22, se van a enrutar las VLANs del departamento de Civil y de Electromecánica. Habrá por tanto que conectar el router a dos interfaces del conmutador y hacer que una de ellas pertenezca a la VLAN de Civil y la otra a la de Electromecánica. A continuación, hay que asignar una IP a los enlaces del router que pertenezca a la subred de las diferentes VLANs. Por último, hay que **definir en los PCs la puerta de enlace** (la IP del router de primer salto). En este caso como sólo hay una posible ruta no es necesario hacer nada más en el router, si hubiera más de dos interfaces sería necesario configurar el enrutamiento.

Téngase en cuenta que ahora, dado que las interfaces del router deben pertenecer a distinta subred, es necesario también que VLANs estén en subredes diferentes, por lo que **hay que modificar las IPs de los equipos** en función de la subred a la que pertenezcan, tal y como se muestra en la Figura 22.

A continuación, se resumen paso a paso las acciones necesarias para la comunicación entre VLANs.

1. Configurar en cada equipo (VPCS) su IP y su puerta de enlace. Cada VLAN tiene que estar en una subred diferente.

Por ejemplo, en el PC1: ip 192.168.0.1/24 gateway 192.168.0.254

2. Añadir un router de Cisco al proyecto.
3. Conectar dos interfaces del router a dos interfaces del switch S1.



- Configurar las IPs de las interfaces del router.

Por ejemplo, para la interfaz conectada a la subred 192.168.0.0/24

```
R1#configure terminal
R1(config)#interface FastEthernet0/0
R1(config-if)#ip address 192.168.0.254 255.255.255.0
R1(config-if)#no shutdown
```

- Configurar en el switch las interfaces conectadas al router para que cada una pertenezca a la VLAN apropiada.

Actividad 8. Partir del proyecto "Prac06_V_02.gns3" (Enlaces troncales con switches de GNS3) y guardarlo como "Prac06_V_02_3.gns3" y configurar la red de la Figura 22 en GNS3, de forma que se permita la conexión utilizando un router intermedio de dos VLANs diferentes.

Los datos de cada una de las subredes que se utilizará son los siguientes:

Subred	VLAN	Puerta de enlace	Interfaz router	Interfaz Switch S1
192.168.0.0/24	VLAN 2 (Electromecánica)	192.168.0.254	f0/0	FastEthernet1/13
192.168.1.0/24	VLAN 3 (Civil)	192.168.1.254	f0/1	FastEthernet1/14

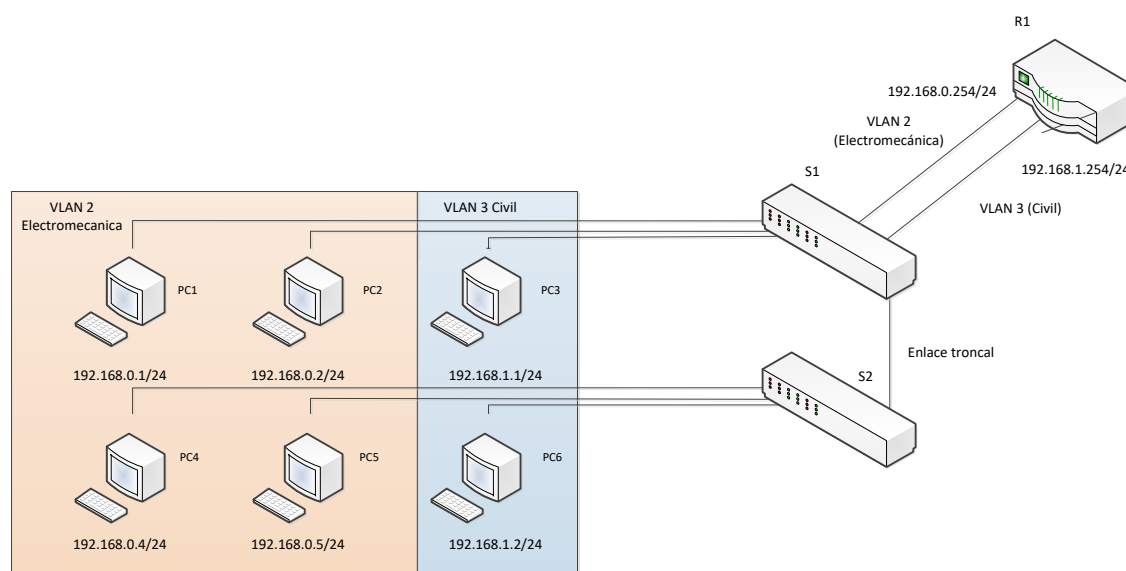


Figura 22. Enrutamiento entre VLANs.

Probar que se puede hacer ping entre los equipos de las dos VLANs. Mostrar una captura resultado de hacer ping entre el PC1 y el PC6 y entre PC1 y PC3. Usar el comando trace para comprobar por donde han ido los paquetes.

¿El router dispone de tabla ARP? ¿Por qué? Mostrar, si la tiene, la tabla ARP del router y explicar por qué contiene las entradas que aparecen.

Si se dispusiera de un conmutador con enrutamiento (conmutador de nivel 3), no sería necesario añadir el router que hemos utilizado anteriormente, el propio switch permitiría, asignando una IP a cada VLAN y activando el enrutamiento, comunicar diferentes VLANs.



2.4 Enrutamiento de VLANs con redes externas

Cuando en una subred existen distintas VLAN y se desea enviar paquetes hacia una red externa a través de routers, es necesario crear subinterfaces en la interfaz del router en la que esté conectada la subred con VLANs. Será necesario crear una subinterfaz (p.e. fa0/0.1, fa0/0.2...) por cada VLAN que se desee conectar al exterior. Al igual que en el caso anterior, cada VLAN debe estar en una subred diferente.

La dirección IP correspondiente a las subinterfaces en el router debe pertenecer al rango de IPs de la subred a la que pertenece. El router tendrá entonces una dirección IP distinta en cada subred y cada subinterfaz, que será la que se habrá que haber definido como dirección de puerta de enlace en los equipos de cada una de las VLANs.

Los comandos para definir las subinterfaces en el router serán los siguientes, por ejemplo, para la primera subinterfaz:

```
R1 (config) #interface FastEthernet0/0.1
R1 (config-subif) #encapsulation dot1Q 2
R1 (config-subif) #ip address 192.168.0.254 255.255.255.0
R1 (config-subif) #no shutdown
```

Con esto creamos la subinterfaz 1 de la interfaz FastEthernet 0/0, posteriormente se define el modo de encapsulación 802.1 Q y se asigna a la VLAN 2. Finalmente se define la dirección IP y máscara a la subinterfaz de este puerto. La encapsulación 802.1Q es una modificación del estándar Ethernet que añade tramas a la red Ethernet que permiten, entre otras cosas, identificar la VLAN a la que pertenece la trama.

Habría que repetir estos mismos comandos para cada subinterfaz de red del router.

Nota: Hay que tener en cuenta que el enlace que une el switch con el router debe permitir el tráfico de las distintas VLANs, luego debe configurarse también como **troncal**.

Actividad 9. Crear el proyecto "Prac06_VI_02_4.gns3" a partir del ejercicio anterior, eliminando el router y añadiéndolo de nuevo. Configurar la red de la Figura 23 con un router y un equipo externo (otro VPCS) con las direcciones IP de la figura. Configurar las subinterfaces del router, teniendo en cuenta que las subredes y subinterfaces del router son:

Subred	VLAN	Puerta de enlace	Subinterfaz router
192.168.0.0/24	VLAN 2 (Electromecánica)	192.168.0.254	f0/0.1
192.168.1.0/24	VLAN 3 (Civil)	192.168.1.254	f0/0.2
172.16.0.0/16	---	172.16.255.254	f0/1



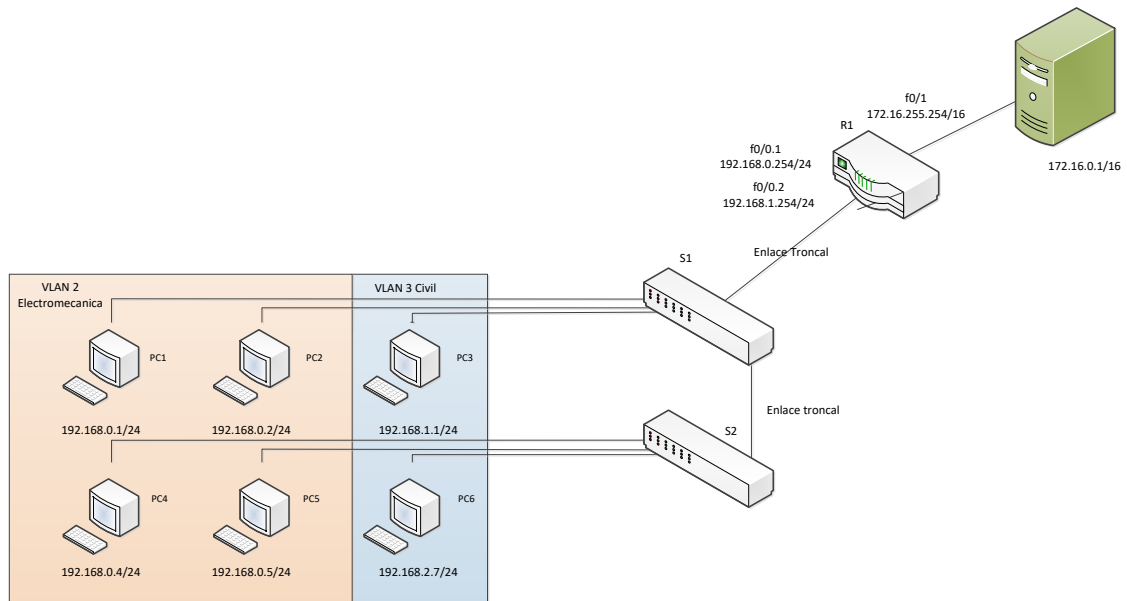


Figura 23. Conexión de VLANs con red externa.

Probar mediante comandos ping que desde cualquiera de las VLAN se puede conectar con el equipo de la red externa. Mostrar capturas con el resultado de hacer ping desde un equipo de la VLAN 2 y desde un equipo de la VLAN 3.





Grado en Ingeniería Informática

REDES

PRÁCTICA 7

Enrutamiento estático y dinámico

Docentes:

Alejandro Merino

Daniel Sarabia Ortiz

Dpto. de Ingeniería Electromecánica

Área de Ingeniería de Sistemas y Automática

Versión 1.4

Fecha 28/04/2021 20:18

Esta obra está sujeta a la licencia Reconocimiento 4.0 Internacional de Creative Commons. Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by/4.0/>



Índice de contenidos

I	INTRODUCCIÓN	3
	1.1 Enrutamiento estático.....	3
	1.2 Enrutamiento dinámico	3
II	OBJETIVOS	4
III	ENRUTAMIENTO ESTÁTICO	4
IV	ENRUTAMIENTO DINÁMICO	9
	1 Enrutamiento RIP	9
	2 Enrutamiento OSPF	10



I Introducción

Una de las funciones de la capa de red es el enrutamiento. El enrutamiento consiste en determinar el camino óptimo entre equipos terminales a través de una red.

Como se ha visto en la teoría, uno de los campos de las cabeceras IP de los paquetes que circulan por una red es la IP de destino. Cuando un router recibe un paquete, analiza el valor de la IP de destino contenida en la cabecera y consulta su tabla de reenvío para determinar por cuál de los enlaces que posee debe reenviar el paquete. En la Figura 1, el router 1 posee 4 enlaces y su tabla de reenvío indica por cual de esos enlaces debe enviar los paquetes en función de valor de IP de destino de la cabecera.

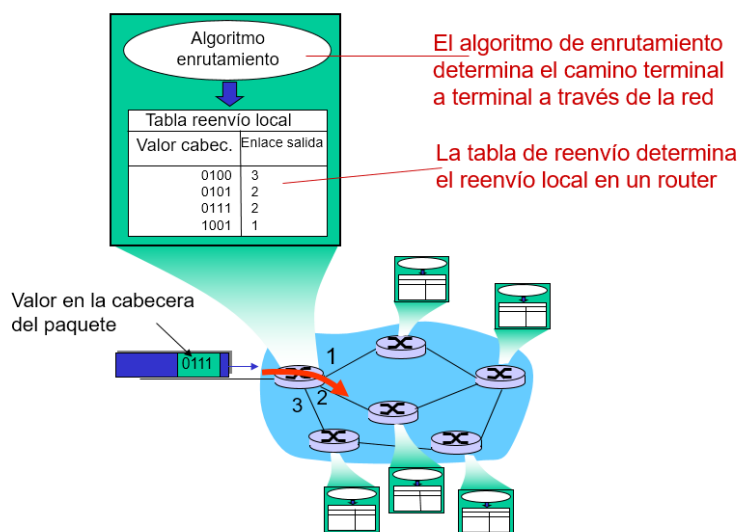


Figura 1. Enrutamiento y reenvío.

Pero, ¿Quién rellena la tabla de reenvío? Los encargados de completar esta tabla son los **algoritmos de enrutamiento**.

1.1 Enrutamiento estático

Una de las formas de direccionar paquetes en una red es hacerlo de forma manual. El administrador de red decide por qué camino deben circular los paquetes en función del destino y configura uno a uno los routers bajo su control para establecer las rutas.

Este tipo de configurar el enrutamiento tiene la ventaja de que es sencillo y consume poca CPU de los equipos, por lo que resultan apropiados para redes pequeñas o en combinación con el enrutamiento dinámico. Por otro lado, tienen la desventaja de requerir la supervisión e intervención de un operador.

1.2 Enrutamiento dinámico

En el caso del enrutamiento dinámico, es un algoritmo el que calcula las rutas y el que se encarga de completar las tablas de reenvío de los routers.

En el caso de internet los algoritmos más utilizados son:

- RIP: Routing Information Protocol
- OSPF: Open Shortest Path First
- IGRP: Interior Gateway Routing Protocol (Propietario de Cisco)



II Objetivos

- Aprender a configurar el enrutamiento estático y dinámico (RIP y OSPF) en una red con varios hosts y routers.
- Analizar el efecto de modificar el costo de los enlaces en un algoritmo de enrutamiento.

III Enrutamiento estático

En esta primera parte de la práctica se va a simular un enrutamiento estático utilizando GNS3.

Para ello se va a configurar una red como la que aparece en la Figura 2.

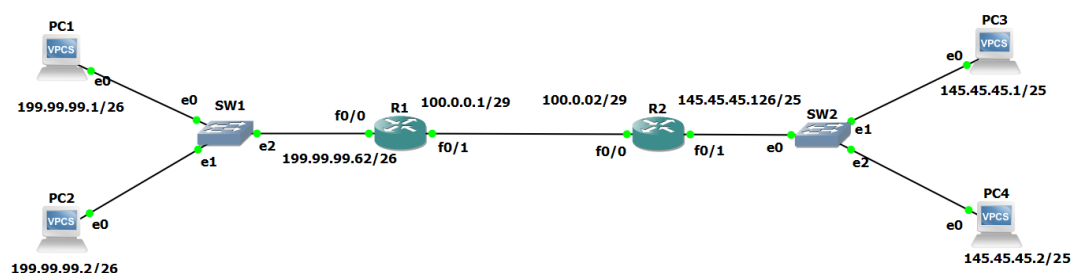


Figura 2.

Configuración de red de los equipos terminales

Nombre	Tipo	Dirección IP	Máscara	Puerta de enlace
PC1	VPCS	199.99.99.1	255.255.255.192	199.99.99.62
PC2	VPCS	199.99.99.2	255.255.255.192	199.99.99.62
PC3	VPCS	145.45.45.1	255.255.255.128	145.45.45.126
PC4	VPCS	145.45.45.2	255.255.255.128	145.45.45.126

Configuración de red de los switches

Nombre	
SW1	No es necesario configurar nada. Es suficiente con utilizar los switches que incorpora GNS3. No se recomienda utilizar los de Cisco ya que no se va a configurar nada en ellos y no tiene sentido sobrecargar la CPU.
SW2	



Configuración de red de los routers

Nombre	FasEthernet0/0	FasEthernet0/1
R1	199.99.99.62/26	100.0.0.1/29
R2	100.0.0.2/29	145.45.45.126/25

Configuración de los VPCS

Para configurar las IPs en los VPCSs debes escribir, por ejemplo:

```
PC1>ip 199.99.99.1/26 gateway 199.99.99.62
```

Puede observarse como en este caso, es necesario especificar la dirección de la puerta de enlace (gateway). Cuando un equipo vaya a enviar un mensaje a una dirección IP que no pertenece a su subred, lo que hará es enviarlo a la puerta de enlace, que es su conexión con el exterior, para que sea este el que lo envíe donde corresponda según su tabla de enrutamiento.

Para comprobar que la asignación se ha hecho correctamente se puede escribir el comando:

```
PC1>show
```

Para no perder la configuración realizada para ese equipo se debe escribir:

```
PC1>save
```

De esta forma si se para y vuelve a arrancar la simulación no se pierde la configuración que sí que perderías de otra forma.

Configuración de los routers.

En el caso de los routers, se deben configurar primero las direcciones IP de los enlaces.

```
R1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)# interface FastEthernet0/0
R1(config-if)#ip address 199.99.99.62 255.255.255.192
R1(config-if)#no shutdown
```

Una vez configuradas las interfaces se deben configurar las rutas estáticas. Las rutas se establecen determinando en cada router el camino por el que deben salir los mensajes para una determinada subred de destino.

Por ejemplo, para el router 1, se debe escribir:

```
R1(config)#ip route 145.45.45.0 255.255.255.128 100.0.0.2
```

Es decir, los mensajes que tengan como destino la subred 145.45.45.0 con máscara 255.255.255.128, tienen como dirección de primer salto (desde ese router) la 100.0.0.2.



Otra sintaxis posible es:

```
R1 (config) #ip route 145.45.45.0 255.255.255.128
FastEthernet0/1
```

Es decir, los mensajes que tengan como destino la subred 145.45.45.0/25, se deben enviar a través de la interfaz FastEthernet 0/1.

En este caso, como sólo hay una red de destino y una interfaz, sería suficiente con utilizar la ruta por defecto. La ruta por defecto define el interfaz de salida por la que se enviarán los mensajes que el router no sepa dónde direccionar. La ruta por defecto se definiría así:

```
R1 (config) #ip route 0.0.0.0 0.0.0.0 FastEthernet0/1
```

De la misma forma que en el caso de los VPCS, si no se desea perder la configuración de unos arranques a otros se puede salvar la configuración escribiendo:

```
R1#copy running-config startup-config
```

Es posible comprobar la configuración de las rutas escribiendo:

```
R2#show ip route
```

Se mostrarán las rutas configuradas en ese router con un código. Por el momento, al ser rutas estáticas, deben aparecer las redes directamente conectadas indicadas con la letra C y las redes accesibles por enrutamiento estático señaladas con una S. Por ejemplo:

```
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default, U - per-user static route
        o - ODR, P - periodic downloaded static route

Gateway of last resort is not set

C    100.0.2.0/29 is directly connected, FastEthernet0/1
S    145.45.0.0/25 [1/0] via 100.0.0.2
```

Se manera general, el significado de las entradas en las tablas de enrutamiento de Cisco, es el siguiente:

Imaginemos una línea del tipo:

```
S 200.0.4.0 [110/2] via 200.0.5.2, 00:01:08, FastEthernet1/0
```

La primera letra indica el mecanismo por el que se ha aprendido la ruta a esa dirección, de los algoritmos que hemos visto:

S: Static

R: RIP

O: OSPF



A continuación, se indica que la subred 200.0.4.0 está accesible a través del interfaz FastEthernet1/0 y el siguiente salto se corresponde con la interfaz con IP 200.0.5.2.

En el campo [110/2] se indican dos cosas, la distancia administrativa (el valor 110) y el coste al destino (el valor 2)

- **La distancia administrativa**, depende del algoritmo que se esté utilizando y se utiliza para dar prioridad a unos algoritmos de enrutamiento frente a otros cuando una ruta ha sido aprendida por distintos algoritmos y por tanto se superponen. Los valores por defecto de las distancias administrativas son, para los algoritmos vistos:
 - Estático: 1
 - OSPF: 110
 - RIP: 120

Esto quiere decir, que si existen distintas rutas a un destino se dará prioridad a las rutas estáticas, frente a las dinámicas y a las rutas aprendidas mediante OSPF frente a las aprendidas con RIP.

- **El costo** indica el coste hasta el destino, que como se indicó antes depende del ancho de banda del enlace. En los routers que usamos en GNS3, los costos son de 10, para las interfaces FastEthernet0/0 y FastEthernet0/1 y de 1 para las interfaces FastEthernet1/0 y FastEthernet2/0.

Si se desea eliminar alguna ruta de las que se han creado se puede hacer escribiendo **no** delante de la ruta a eliminar:

```
R1 (config) #no ip route 145.45.45.0 255.255.255.128 100.0.0.2
```

La configuración de los routers puede hacerse algo repetitiva y es propicia a cometer errores, para facilitar la tarea de escribir comandos, se recomienda escribir todos los comandos en un editor de texto, por ejemplo:

```
configure terminal
interface FastEthernet0/0
ip address 199.99.99.62 255.255.255.192
no shutdown
exit
exit
copy run start
```

Estos comandos pueden copiarse y pegarse todos a la vez en la interfaz de comandos, lo que facilita la mucho la tarea de escribirlos, ya que es posible reutilizarlos y detectar errores fácilmente.

Actividad 1. Crear el proyecto "Prac07_act1" y configurar la red que aparece en la Figura 2, configurando los dos routers que aparecen de manera estática.

- Hacer ping entre dos PCs de las dos subredes diferentes. Mostrar la captura de la salida de los pings.
- Escribir el comando: show ip route, para ver las tablas de enrutamiento de los routers. Mostrar la captura del resultado. Interpretar el resultado obtenido.



Si no se consigue hacer ping, una manera de tratar de averiguar dónde está el problema sin tener que revisar todos los equipos, es analizar la ruta que sigue un paquete desde el emisor al receptor. Con el comando *trace*, es posible conocer la ruta que sigue un paquete.

La sintaxis es:

```
trace HOST [OPTION ...]

Print the path packets take to the network HOST. HOST can be
an ip address or name.

Options:
  -P protocol      Use IP protocol in trace packets
                   1 - icmp, 17 - udp (default), 6 - tcp
  -m ttl           Maximum ttl, default 8
```

Por ejemplo:

```
PC6> trace 145.45.45.2 -P 1
trace to 145.45.45.2, 8 hops max (ICMP), press Ctrl+C to stop
 1  145.45.45.126  2.930 ms  9.772 ms  8.771 ms
```

Hay que tener en cuenta que cuando se configuran rutas estáticas, si en los routers no se proporciona el camino de vuelta de los mensajes, el comando *trace* no será capaz de devolver los mensajes de respuesta y se mostrarán asteriscos en vez de la IP de cada salto.

Actividad 2. Descargar de UBUVirtual el proyecto de GNS3 "Pract7_Plantilla.gns3project" en la que está configurada la red que se muestra en la Figura 3. Cargar la plantilla y guardarla como Prac07_Enrut_1_2. En esta plantilla están configuradas todas las direcciones IP, pero es necesario configurar estáticamente los routers tratando de que las rutas de ida y vuelta de los mensajes sean diferentes. Debe ser posible, al igual que antes, hacer ping entre las distintas subredes. Explicar las rutas que se han configurado para comunicar las distintas subredes.

- Mostrar una captura resultado de hacer ping entre el PC1 y el PC3, entre el PC2 y el PC5 y entre el PC3 y el PC6
- Mostrar las capturas de las tablas de enrutamiento de los cuatro routers.
- Utilizar el comando *trace* para mostrar tres rutas entre equipos de distintas subredes. Explicar los resultados obtenidos. Si aparecen asteriscos en alguna ruta explicar por qué aparecen y cómo podrían evitarse.



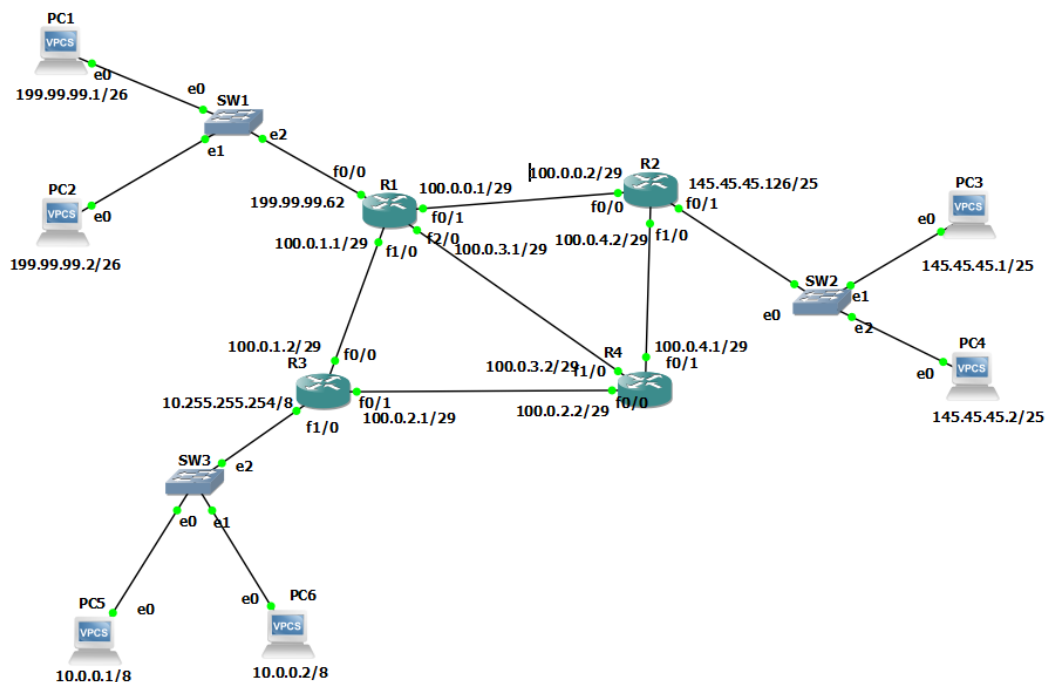


Figura 3

IV Enrutamiento dinámico

En el caso del enrutamiento dinámico, existen varios protocolos que pueden aplicarse. En esta práctica se van a estudiar los dos algoritmos más habituales RIP y OSPF.

1 Enrutamiento RIP

Si se desea aplicar enrutamiento RIP en un determinado sistema autónomo, lo primero que habrá que hacer es activar este tipo de enrutamiento, para ello se ejecutará el siguiente comando:

```
R1(config)#router rip
```

Una vez activado el protocolo RIP, es necesario especificar en cada uno de los routers las subredes que están conectadas a ese router. El router con esa información, aplicando el algoritmo de enrutamiento que le indiquemos, será capaz de encontrar por sí mismo las rutas óptimas. Por ejemplo, en el caso del router R1 habría que añadir las subredes que son adyacentes a ese router, que son:

- 199.99.99.0
- 100.0.0.0
- 100.0.1.0
- 100.0.3.0

Los comandos para realizar esto son los siguientes:

```
R1(config-router)#network 199.99.99.0
```



```
R1(config-router)#network 100.0.0.0
R1(config-router)#network 100.0.1.0
R1(config-router)#network 100.0.3.0
```

Si existieran rutas estáticas que se quiere eliminar, desde la interfaz de línea de comandos se puede escribir:

```
R1>enable
R1#show ip route
R1#configure terminal
R1(config)#no ip route 145.45.45.0 255.255.255.128
...
```

Se recomienda mediante el comando *show ip route*, comprobar, antes de eliminarlas, qué rutas estáticas están configuradas (aparecen etiquetadas con una S).

Actividad 3. Partiendo de la misma plantilla utilizada anteriormente configurar los routers para que utilicen el protocolo de enrutamiento dinámico RIP. Probar con esta nueva configuración si se puede hacer ping entre los diferentes equipos de las subredes. Almacenar el resultado en el proyecto "Prac07_act3".

- Mostrar una captura resultado de hacer ping entre el PC1 y el PC3, entre el PC2 y el PC5 y entre el PC3 y el PC6. (Es posible que se produzca un timeout en los primeros paquetes).
- Mostrar mediante una captura de pantalla la tabla de enrutamiento de los routers R2 y R3.
- Utilizando el comando *trace* en los VPCS determinar las rutas que siguen los paquetes para completar la siguiente tabla:

IP Origen	Salto 1	Salto 2	Salto 3	IP destino
199.99.99.1				145.45.45.2
199.99.99.2				10.0.0.1
10.0.0.1				199.99.99.1
10.0.0.2				145.45.45.2
145.45.45.1				199.99.99.2
145.45.45.2				10.0.0.2

¿Son lógicas las rutas que se obtienen?

¿Por qué en este caso no aparecen asteriscos en las rutas?

2 Enrutamiento OSPF

Si se desea utilizar el protocolo de enrutamiento OSPF, lo primero que habrá que hacer es activarlo, para ello, en cada router habrá que escribir:



```
R1(config)#router OSPF 1
```

A continuación, al igual que se hizo en el caso de RIP, se añaden las redes a las que está conectado el router, en este caso es necesario además incluir:

1. La máscara de red, en este caso se añade una máscara de red en formato wildcard o complementado (por ejemplo, para el caso de 255.255.255.0, la máscara wildcard será 0.0.0.255).
2. El área OSPF, en este caso, al tratarse de una red sencilla suponemos que sólo hay un área con el número 0.

Para R1 por ejemplo habría que añadir:

```
R1(config-router)# network 100.0.0.0 0.0.0.7 area 0
R1(config-router)# network 100.0.1.0 0.0.0.7 area 0
R1(config-router)# network 100.0.3.0 0.0.0.7 area 0
R1(config-router)# network 199.99.99.0 0.0.0.63 area 0
```

Para desactivar un protocolo de enrutamiento, será necesario escribir "no" delante del comando de activación. Por ejemplo, se existe enrutamiento RIP y se desea eliminarlo, se debe escribir el comando:

```
R1(config)#no router RIP
```

Coste de los enlaces

Vimos que en el protocolo RIP, el coste de los enlaces era siempre 1 y no se podía modificar. En el caso de OSPF, este coste sí que se puede variar. El costo de los enlaces en OSPF es por defecto inversamente proporcional al ancho de banda del enlace, es decir cuanto mayor ancho de banda, menor coste. En concreto el coste se calcula como un valor de referencia, que por defecto es 100 Mbps, partido por ancho de banda de la interfaz en bps:

$$\text{Coste} = 100\,000\,000 / \text{ancho de banda en bps}$$

En el caso de los routers que se utilizan en la práctica, las interfaces del slot 0 (GT96100FE: FasEthernet0/0 y FasEthernet0/1), tienen un ancho de banda de 10 Mbps, mientras interfaces de los slots 1 y 2 (NM-1FE-TX): FasEthernet1/0 y FasEthernet2/0), tienen un ancho de banda de 100 Mbps.

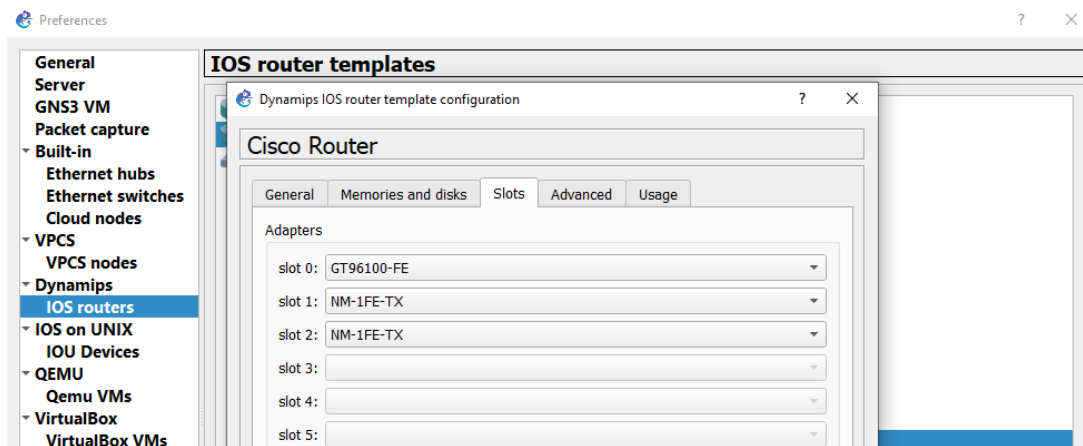


Figura 4. Interfaces FastEthernet en cada uno de los slots del router C3725.

El coste de las interfaces se puede modificar de manera manual, para ello se utiliza el comando *ip ospf cost* aplicado a la interfaz correspondiente, por ejemplo:

```
R1(config)#interface fa1/0  
R1(config-if)#ip ospf cost 65000
```

Esto provoca que el coste del enlace sea 65000 (el valor más alto posible)

Para comprobar el coste del enlace se puede escribir, por ejemplo:

```
R1#show ip ospf interface FastEthernet1/0
```

El coste aparece al final de la tercera línea.

Actividad 4. Partiendo de la plantilla suministrada ("Pract7_Plantilla.gns3project"), configurar el enrutamiento utilizando en este caso el protocolo OSPF. Almacenar el resultado en un proyecto nombre Prac07_Enrut_1_5. Mostrar las capturas con las tablas de enrutamiento de R1 y R4 que se obtiene.

Mostrar el resultado, utilizando el comando *trace*, de la ruta que siguen los paquetes entre el PC3 y el PC5. Explicar de manera cuantitativa, utilizando los valores de los costos que se pueden ver en las tablas de enrutamiento, por qué se sigue la ruta calculada entre el PC3 y el PC5.

Actividad 5. Modificar el costo de la interfaz f1/0 de R2, dándole un valor de 1000. Mostrar una captura con los comandos utilizados y mostrar el costo del enlace con el comando *show ip ospf interface f1/0*.

Mostrar, mediante capturas de pantalla del comando *trace*, el camino seguido por un paquete desde el PC3 al PC5 y a la inversa. Explicar por qué se siguen las rutas obtenidas.





Grado en Ingeniería Informática

REDES

PRÁCTICA 8

NAT

Docentes:

Alejandro Merino

Daniel Sarabia Ortiz

*Dpto. de Ingeniería Electromecánica
Área de Ingeniería de Sistemas y Automática*

Versión 1.1

Fecha 14/04/2020 11:35

Esta obra está sujeta a la licencia Reconocimiento 4.0 Internacional de Creative Commons. Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by/4.0/>



Índice de contenidos

I	INTRODUCCIÓN TEÓRICA.....	3
II	OBJETIVOS.....	5
III	CONTENIDOS ESPECÍFICOS DEL TEMA	5
	1 Configuración de NAT dinámico con GNS3	5
	2 Configuración de NAT estático con GNS3.....	8
	3 NAT con sobrecarga (PAT, Port Address Translation).....	9
IV	BIBLIOGRAFÍA	12



I Introducción teórica

Con la proliferación de las redes domésticas y de oficina pequeña (Small Office Home Office SOHO), se hizo necesario un mecanismo que flexibilizara la asignación de direcciones IP. En caso contrario, sería necesario que el ISP proporcionara bloques de direcciones IP a cada hogar o empresa pequeña, con el problema que supondría ampliar los equipos de la red fuera de ese bloque, el consumo excesivo de direcciones IP y la dificultad para los usuarios no expertos en la gestión de las direcciones.

Para solventar estos problemas, se ha popularizado la utilización de la traducción de direcciones NAT (Network Address Translation).

NAT es un procedimiento que se implementa en los routers y que permite traducir direcciones públicas en direcciones privadas, de forma que una o unas pocas direcciones públicas gestionen un gran número de direcciones privadas.

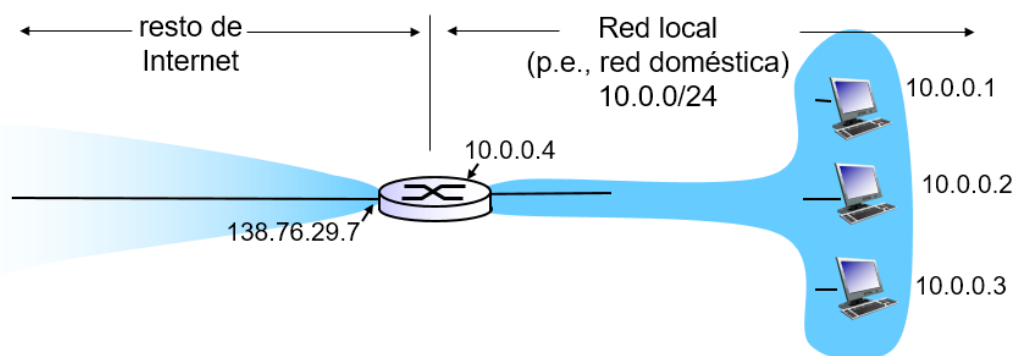


Figura 1. Esquema de una red que implementa NAT

En la Figura 10 se muestra una red que implementa NAT. En esta red, todos los datagramas que abandonan la red local tienen la misma dirección NAT IP: 138.76.29.7, con diferente número de puerto de origen. Dentro de la red local, las IPs son privadas en el rango 10.0.0.0/24. Por tanto, las direcciones privadas tienen que ser traducidas a direcciones públicas en el router.

Existen diferentes formas de configurar un traductor NAT, se puede hacer de manera manual, configurando de manera estática la traducción o de manera dinámica, de manera que la traducción sea automática, que es lo más usual.

La traducción se hace de la siguiente manera.

- En los datagramas salientes, se reemplaza la dirección IP privada de origen con su número de puerto, a la dirección IP pública con un nuevo número de puerto que no sea bien conocido¹.
- En el router se almacenarán en su tabla de traducciones NAT los pares:

Dirección IP de origen, número de puerto de origen -> dirección IP pública, nuevo número de puerto.

Los clientes/servidores remotos responderán a este datagrama usando la dirección IP pública con su nuevo número de puerto como dirección de destino

¹ El número de puerto es un campo utilizado en la capa de transporte para identificar procesos, es un campo de 16 bits, por lo que se pueden asignar un total de 65535 puertos. Existen números de serie, llamados bien conocidos, que son utilizados típicamente por protocolos de red, que habrá que evitar en las traducciones NAT. https://es.wikipedia.org/wiki/Anexo:Números_de_puertos_de_red.



- Cuando el router recibe un mensaje con destino a un equipo dentro de su red NAT, consulta su tabla y hace la traducción inversa, reemplazando la dirección IP pública con el nuevo número de puerto por la correspondiente dirección IP de privada, número de puerto almacenado en la tabla NAT.

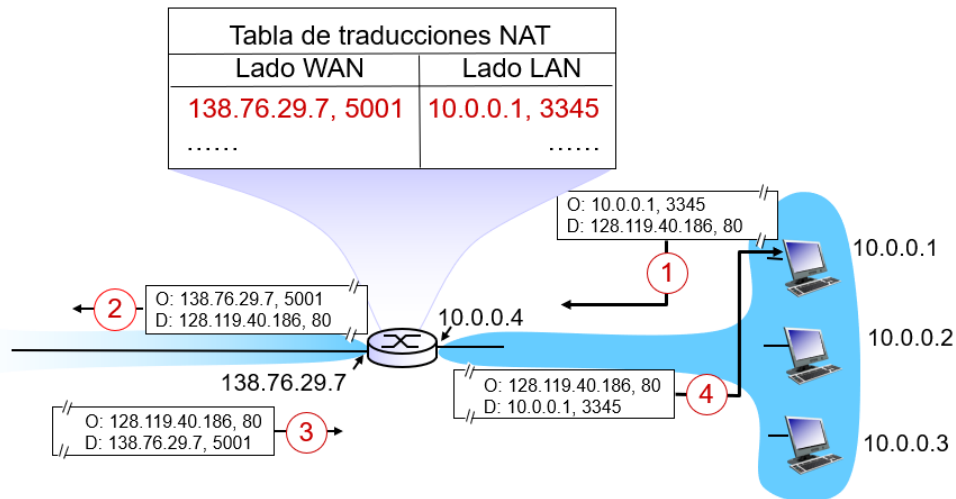


Figura 2. Ejemplo de traducción NAT.

En la figura 4 se muestra un ejemplo de traducción NAT.

1. Se envía un datagrama con dirección IP de origen 10.0.0.1 (dirección privada) y número de puerto 3345 y puerto de destino 128.119.40.186 número de puerto 80 (es un puerto que se corresponde al protocolo HTTP, se está haciendo una consulta web)
2. El datagrama llega al router y se reemplaza el campo de dirección IP de origen por la dirección IP pública del router (10.0.0.1 ->138.76.29.7), y el puerto de origen se reemplaza por un nuevo número de puerto (3345->5001). Esta traducción se almacena en el router.
3. Los datagramas de respuesta (la página web solicitada) se envían a la dirección IP/puerto traducidos, así el destino del mensaje de respuesta es: 138.76.29.7 5001.
4. El mensaje llega al router que busca el par 138.76.29.7 5001 en su tabla y hace la traducción inversa, volviendo al par IP/puerto original 10.0.0.1, 3345.

La traducción NAT presenta una serie de ventajas en las redes SOHO.

- No es necesario obtener un rango de direcciones del ISP: sólo una dirección IP para todos los dispositivos.
- Es posible modificar las direcciones de los dispositivos en la red local sin notificarlo al mundo exterior.
- Se puede cambiar de ISP sin cambiar las direcciones de la red local.
- Los dispositivos en la red local no son direccionables explícitamente ni visibles desde el mundo exterior (es un plus de seguridad).
- El campo del número de puerto tiene 16 bits lo que permite establecer 60 000 conexiones simultáneas con una única dirección IP WAN del router.

Aunque existen también algunas objeciones:



- Si se respeta la jerarquía de capas, los números de puerto deben usarse para direccionar procesos no direccionar hosts.
- Los routers están pensados para procesar paquetes sólo hasta la capa 3 y la modificación de los números de puerto pertenecen a la capa 4.
- Viola el enfoque terminal a terminal y los nodos intermedios no deberían modificar los números de puerto ni las direcciones IP
- La carestía de direcciones IP debería resolverse utilizando IPv6.

II Objetivos

- Comprender el objetivo de las redes NAT.
- Comprender las aplicaciones de las redes NAT.
- Estudiar las distintas maneras de configurar un traductor NAT.
 - NAT Dinámico
 - NAT Estático
 - NAT con sobrecarga.
- Implementar las configuraciones estudiadas en routers Cisco.
- Averiguar la visibilidad de los hosts en redes NAT.

III Contenidos específicos del tema

1 Configuración de NAT dinámico con GNS3

En el NAT dinámico, las direcciones IP internas de cada cliente de una LAN se asocian dinámicamente con cada IP externa (pública) en una relación 1 a 1. Si tenemos, como en el caso del ejemplo que se muestra a continuación, 3 direcciones internas necesitaremos 3 direcciones públicas. Si el número de IPs públicas es menor que el número de equipos que tratan de conectarse con el exterior, sólo lo conseguirán obtener una dirección los primeros equipos en tratar de conectarse. En el ejemplo, si un cuarto equipo solicitara la conexión y el pool de direcciones públicas estuviera agotado no podría conectarse con el exterior.

En NAT dinámico, cuando un equipo envía un mensaje con destino al exterior de la red local a través de un router que implementa NAT, el router, de manera dinámica, asigna una equivalencia entre la IP local del equipo que inicia la comunicación, con una de las IPs públicas disponibles. Esta equivalencia crea una entrada en la tabla NAT del router que se mantiene activa por defecto durante 24 h (en GNS 3 es mucho menor, este valor puede modificarse: <https://supportforums.cisco.com/document/28351/how-configure-nat-translation-timeoutb>).

En este caso se va a utilizar el ejemplo que se muestra en la Figura 3 cuya plantilla puede descargarse de UBUVirtual. La plantilla tiene ya configuradas las direcciones IP y las rutas en todos los equipos. En R1 no se incluye enrutamiento a la subred la 192.168.0.0/24, ya que **es una red privada que no se debe anunciar al exterior**.

En este ejemplo se van a traducir las direcciones privadas de los equipos internos desde la 192.168.0.x a las direcciones públicas 200.0.0.x.



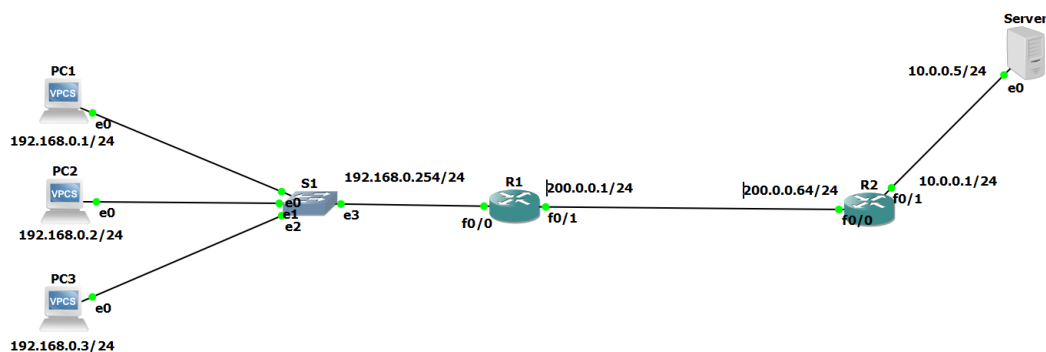


Figura 3. Esquema de red actividad 1.

Para realizar la configuración de NAT dinámico es necesario definir un pool (rango) de direcciones públicas y un pool de direcciones privadas.

Por ejemplo, se va a definir un pool de direcciones públicas y privadas de manera que se pueda realizar la asignación dinámica de las mismas:

Pool de direcciones públicas:

- 200.0.0.2 – 200.0.0.4

Rango de direcciones privadas:

- 192.168.0.0– 192.168.0.255

Para ello en R0 es necesario escribir:

```
R1(config)# ip nat pool nombre ip-comienzo ip-final netmask mascara
p.e.
R1(config)# ip nat pool pract8_act1 200.0.0.2 200.0.0.4 netmask
255.255.255.248
```

Donde nombre es el nombre con el que queremos que se almacene el pool de direcciones, ip-comienzo y final son la primera y la última dirección del pool y netmask es la máscara de la red a la que pertenecen las direcciones.

En segundo lugar, se crea la lista de acceso con las direcciones internas que vamos a convertir, por ejemplo:

```
R1(config)# access-list 1 permit 192.168.0.0 0.0.0.255
```

El comando *access-list* define una lista de control de acceso, en este caso la máscara se define como wildcard, es decir, al revés que en el caso de la definición de las redes. La lista de acceso tiene el valor 1 para el rango de IPs a los que se permite el acceso. En este caso la lista con el identificador 1 permite las direcciones de la 192.168.0.0 a la 192.168.0.255.

A continuación, se define NAT con el pool de direcciones que hemos definido

```
R1(config)# ip nat inside source list 1 pool pract8_act1
```

Por último, definimos el lado externo y el lado interno de la NAT

```
R1(config)# interface fastEthernet 0/0
R1(config-if)# ip nat inside
R1(config)# interface fastEthernet 0/1
```



2º Grado en Ingeniería en Informática

```
R1(config-if)# ip nat outside
```

Si se activa el modo debug para el Router 0.

```
R1# debug ip nat
```

Haciendo un ping desde uno de los PCs de la subred al servidor, podemos observar las traducciones que hace NAT.

```
Router#debug ip nat
IP NAT debugging is on
Router#
NAT: s=192.168.0.2->200.0.0.2, d=10.0.0.5 [23]
NAT*: s=10.0.0.5, d=200.0.0.2->192.168.0.2 [25]
```

Para desactivar el modo debug, escribir:

```
R1#no debug ip nat
```

También es posible también comprobar que la asignación de direcciones se ha hecho de forma correcta con el comando *show ip nat translations*.

```
R1#show ip nat translations
Pro Inside global Inside local Outside local Outside global
icmp 200.0.0.3:1 192.168.0.4:1 10.0.0.5:1 10.0.0.5:1
icmp 200.0.0.4:1 192.168.0.2:1 10.0.0.5:1 10.0.0.5:1
```

En esta tabla se muestran 4 campos:

- **Inside local:** Se muestra cómo se ve la dirección local de un equipo interno desde el interior de la red local.
- **Inside global:** Se muestra que dirección de un host interno vista desde el exterior.
- **Outside local:** Cómo se ve la dirección de un host externo desde mi red local.
- **Outside global:** Cómo se ve la dirección de un host externo desde la red externa. En este caso dado que no hay traducción para la red externa, la dirección será la misma tanto desde dentro como desde fuera de NAT.

La principal desventaja de NAT dinámico es que por cada equipo que se desee tener acceso a Internet se debe contratar una IP pública por lo que sólo se pueden mapear tantas IP privadas como IP públicas se tengan.

Actividad 1. Sobre la plantilla "Pract08_NAT_act1_soluc.gns3project", configurar NAT dinámico tal y como se describe con anterioridad.

- Enviar un paquete desde un equipo de la subred hasta el servidor y ver, utilizando el debug de nat, el cambio que se produce en la dirección de origen



y de destino del mensaje en el primer router. (El servidor es un VPCS al que se le ha cambiado el icono).

- Visualizar la tabla de traducciones NAT después del ping anterior.
- ¿Es posible hacer ping desde el Server0 a la dirección privada de los equipos de la subred detrás del router NAT? ¿Y a la dirección pública? ¿Por qué?
- Y a la inversa, ¿Es alcanzable el servidor externo desde los equipos de la LAN?
- Añadir un equipo más a la subred privada y tratar de hacer ping al servidor desde todos los equipos uno detrás de otro. ¿Es posible? ¿Por qué?

2 Configuración de NAT estático con GNS3

En NAT estático es posible asignar estáticamente una dirección IP pública única a una dirección IP privada. La asignación se realiza escribiendo directamente en la tabla NAT del router la equivalencia estática IP pública – IP privada. Esta configuración es muy útil para permitir, por ejemplo, el acceso desde Internet a un servidor que esté alojado en nuestra red interna, solventando el problema de NAT transversal, que no permite iniciar comunicación con el interior desde equipos que estén fuera de la red LAN.

En este caso, a la LAN del apartado anterior, con la NAT dinámica, se le va a añadir un servidor que debe ser accesible desde el exterior. Para ello le asignaremos una entrada estática en la tabla del traductor NAT Figura 4.

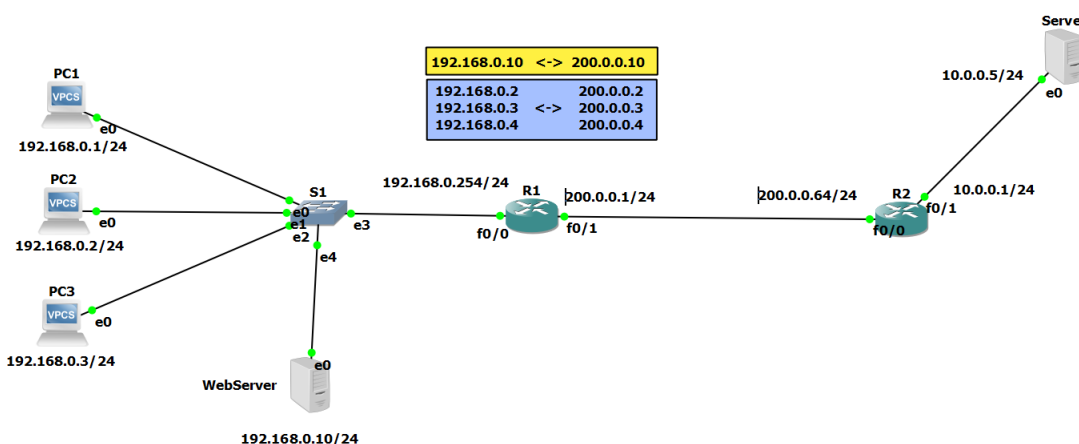


Figura 4. Red que implementa NAT dinámico (en azul) y estático (en amarillo).

La configuración de NAT estático es muy sencilla basta con escribir en el router R0 debemos los siguientes comandos:

Configuramos el mapeo NAT, que asigna la dirección privada a la pública

```
R1(config)#ip nat inside source static 192.168.0.10 200.0.0.10
```

Por último, definimos el lado externo y el lado interno de la NAT

```
R1(config)# interface fastEthernet 0/0
R1(config-if)# ip nat inside
R1(config)# interface fastEthernet 0/1
```



```
R1 (config-if) # ip nat outside
```

Actividad 2. Añadir al ejemplo anterior un servidor con una dirección fija 192.168.0.10 y configurar NAT estático para el servidor de la red LAN tal cual se muestra en el esquema mostrado en la Figura 4.

Comprobar mediante el comando *show IP NAT translations*, que la traducción NAT estática se ha agregado correctamente.

- Probar mediante el comando ping que es posible hacer ping desde el Server0 (externo a la red LAN) a la dirección pública del servidor de la LAN (WebServer).
- ¿Es posible hacer ping desde las direcciones privadas de los equipos de la LAN a la dirección privada del servidor de la LAN? ¿Y a la dirección pública?

3 NAT con sobrecarga (PAT, Port Address Translation)

NAT Con sobrecarga es el más usado en los hogares, ya que permite con una única dirección IP pública mapear varias direcciones IP privadas. La forma de distinguir el destino de los mensajes dirigidos a los distintos equipos dentro de la red doméstica, es a través de los números de puerto. Para ello el traductor NAT hace la siguiente traducción (ver Figura 5):

- *Datagramas salientes:* Para cada datagrama saliente, la dirección IP privada de origen y el número de puerto se traduce a una dirección IP pública y un nuevo número de puerto.
- *Datagramas entrantes:* Se traduce la dirección IP pública con el nuevo número de puerto se traduce a la dirección IP privada y el número de puerto original que está almacenado en la tabla NAT.

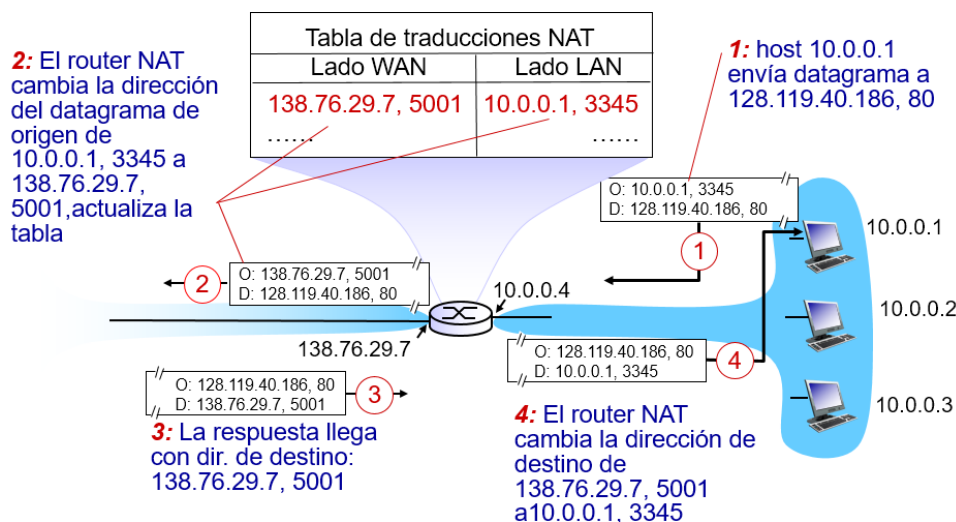


Figura 5. Traducciones NAT

En este caso se va a trabajar con la plantilla "Pract08_NAT_act3_plantilla.gns3project" que representa una red que utiliza enrutamiento RIP y se va a activar NAT en los equipos de la subred 192.168.0.0/24.



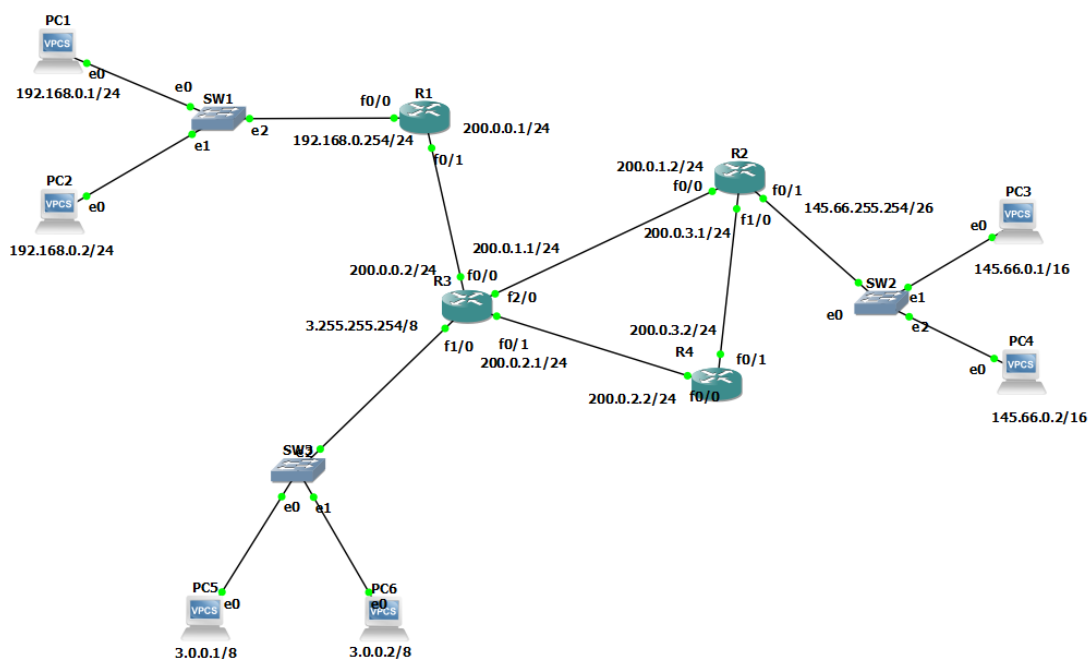


Figura 6. Red a la que se le aplicará NAT con sobrecarga.

Para configurar NAT con sobrecarga es necesario seguir los siguientes pasos.

En primer lugar, se crea la lista de acceso con las direcciones internas que se van a convertir:

```
R1(config)#access-list 1 permit 192.168.0.0 0.0.0.255
```

El comando *access list* define una lista de control de acceso con un identificador para el rango de IPs a los que se permite el acceso. La máscara es de tipo wildcard o complementado. En el ejemplo anterior la lista con el identificador 1 permite las direcciones de la 192.168.0.0 a la 192.168.0.255.

A continuación se define NAT con sobrecarga para la lista que hemos definido

```
R1(config)#ip nat inside source list 1 interface f0/1 overload
```

Esta sentencia indica que los paquetes recibidos en la interfaz interna permitidos por la lista de acceso 1, traducen la dirección de origen hacia la interfaz f0/0 con sobrecarga

Por último, se definen las interfaces internas y externas de NAT.

```
R1(config)#interface fastEthernet0/0
R1(config-if)#ip nat inside
R1(config-if)#exit
R1(config)#interface fastEthernet0/1
R1(config-if)#ip nat outside
```

Si se activa el debugueo de NAT puede verse que, si por ejemplo se hace ping desde del PC 1, se traduce a la única dirección pública del router.



```

R1#debug ip nat
IP NAT debugging is on
R1#
NAT: s=192.168.0.2->200.0.0.1, d=10.0.0.3 [1]
NAT: s=192.168.0.2->200.0.0.1, d=10.0.0.3 [2]
NAT*: s=10.0.0.3, d=200.0.0.1->192.168.0.2 [1]
NAT: s=192.168.0.2->200.0.0.1, d=10.0.0.3 [3]
NAT*: s=10.0.0.3, d=200.0.0.1->192.168.0.2 [2]
NAT: s=192.168.0.2->200.0.0.1, d=10.0.0.3 [4]
NAT*: s=10.0.0.3, d=200.0.0.1->192.168.0.2 [3]

```

PAT tiene una serie de ventajas como que se pueden conectar N estaciones privadas a Internet utilizando solamente una IP pública, además es posible ocultar la cantidad real de direcciones privadas dificultando la tarea de un posible atacante.

Actividad 3. Abrir la plantilla "Pract08_NAT_act3_plantilla.gns3project" y configurar NAT con sobrecarga siguiendo los pasos descritos, aplicando NAT a la subred privada 192.168.0.0/24.

De manera previa a la configuración de NAT con sobrecarga, será necesario **reprogramar el enrutamiento RIP para eliminar de la tabla de enrutamiento la red 192.168.0.0.**

- Comprobar mediante el comando *debug IP NAT* y visualizando la tabla NAT de R1, que las traducciones se realizan correctamente. Comprobar cómo solo se utiliza una dirección IP de salida, pero con diferente número de puerto.
- Probar mediante ping la conectividad entre la subred 192.168.0.0 y el resto de subredes y a la inversa. ¿Puede hacerse ping desde la subred 192.168.0.0 al resto de las subredes? ¿Y a la inversa? ¿Por qué?
- ¿Por qué es necesario eliminar la red 192.168.0.0 del enrutamiento RIP?



IV Bibliografía

Redes de Computadoras. Un enfoque descendente, 5º edición, Jim F. Kurose & Keith W. Ross, Pearson, 2010.

http://www.cisco.com/c/en/us/td/docs/security/asa/asa80/configuration/guide/conf_gd/cfgnat.html





Grado en Ingeniería Informática

REDES

PRÁCTICA 9

DHCP

Docentes:

Alejandro Merino

Daniel Sarabia Ortiz

*Dpto. de Ingeniería Electromecánica
Área de Ingeniería de Sistemas y Automática*

Versión 1.4

Fecha 03/05/2021 12:10

Esta obra está sujeta a la licencia Reconocimiento 4.0 Internacional de Creative Commons. Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by/4.0/>



Índice de contenidos

I	INTRODUCCIÓN.....	3
II	OBJETIVOS.....	3
III	CONTENIDOS ESPECÍFICOS DEL TEMA	3
	1 Análisis del protocolo DHCP con Wireshark.	3
	2 DHCP en Linux.....	5
	3 Configuración de un router como un servidor DHCP	9
IV	BIBLIOGRAFÍA	11



I Introducción

DHCP es un protocolo que permite a un host obtener su dirección IP de manera dinámica. Esta dirección IP es proporcionada por un servidor DHCP. Esto permite reutilizar direcciones, de forma que sólo las mantiene mientras hay conexión, dar soporte a usuarios móviles que quieren unirse a una red, etc. Con ello se consigue la gestión automática de las direcciones y un ahorro de direcciones IP.

II Objetivos

- Aprender a configurar un servicio DHCP tanto en un equipo terminal como en un router.
- Configurar un servidor DHCP en un equipo Linux.
- Comprender el protocolo DHCP y las transacciones que tienen lugar durante la obtención de una dirección IP mediante DHCP.

III Contenidos específicos del tema

1 Análisis del protocolo DHCP con Wireshark.

En este apartado se va a repasar lo visto en teoría acerca del protocolo DHCP utilizando la herramienta Wireshark. Para ello se va a realizar una solicitud DHCP en Windows y se van a analizar con Wireshark los paquetes que se envían y reciben a través de la red.

1. Abrir la interfaz de comandos de Windows (Inicio ->cmd) (Es necesario ejecutar la interfaz de comandos **como administrador**)
2. Abrir Wireshark y comenzar a capturar los paquetes de la conexión de área local (o wifi, según la interfaz desde la que se realice la captura).
3. Desde la interfaz de comandos liberar el direccionamiento IP que haya en ese momento escribiendo el comando:

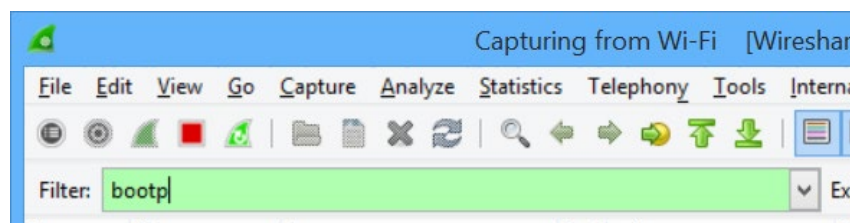
```
ipconfig /release
```

4. Solicitar una dirección IP al servidor DHCP escribiendo el comando:

```
ipconfig /renew
```

5. Volver a renovar la dirección IP escribiendo de nuevo ipconfig /renew.
6. Parar Wireshark y analizar los paquetes del protocolo DHCP. Para filtrar los paquetes pertenecientes a DHCP filtrar utilizando bootp (DHCP procede de un protocolo que se llama BOOTP):





Actividad 1. Seguir los pasos descritos anteriormente para obtener, utilizando Wireshark, los mensajes que se intercambian en el protocolo DHCP. Adjuntar el fichero en el que aparezcan las transacciones de liberación solicitud y renovación de una dirección IP mediante DHCP.

¿A qué nivel de la capa de protocolos pertenece DHCP?

¿Qué protocolo de la capa de transporte se utiliza para transportar los mensajes DHCP?

- **Liberación de una dirección IP (release)**

¿Cuántos pasos se necesitan?

¿Cuáles son las direcciones de origen y de destino? ¿Con qué se corresponden?

- **Obtención de una dirección IP (renew)**

Completar la siguiente tabla:

Nombre de la etapa	IP origen	IP destino	Puerto origen	Puerto destino	ID de la transacción

¿Cuántos y cuáles son los pasos que utiliza DHCP hasta que se determina una dirección IP?

¿Se ha utilizado algún agente retransmisor?

¿Qué tiempo de arrendamiento se ha concedido a la dirección IP suministrada?

¿Cuál es la dirección IP del servidor DHCP?

¿Qué dirección IP y qué máscara me oferta el servidor DHCP en la fase de oferta?

- **Renovación de una dirección IP cuando ya se tiene una (renew por segunda vez)**

¿Cuántos y cuáles son los pasos que utiliza DHCP para la renovación de una IP?



Protocolo IP

Identificar los campos del datagrama IP en los mensajes capturados por el protocolo y comprobar que coinciden con los campos de los datagramas IP estudiados en teoría.

Completar la siguiente tabla, con los campos del datagrama IP, utilizando la información de la transacción ACK del protocolo DHCP.

Versión del protocolo	Long. Cabecera	Tipo de servicio	Longitud datagrama
Identificador fragmentación		Flags fragmentación	Desplazamiento
TTL	Capa superior	Suma de comprobación	
IP Origen			
IP Destino			

2 DHCP en Linux

2.1 Configuración del servicio DHCP en una máquina Ubuntu

En esta sección, se describe cómo configurar un servicio DHCP en una máquina Virtual Ubuntu. En el caso de esta práctica, **no es necesario hacer esto ya que la máquina virtual UbuntuServer1 proporcionada ya está configurada** para proporcionar este servicio.

Conexión de la máquina virtual a internet

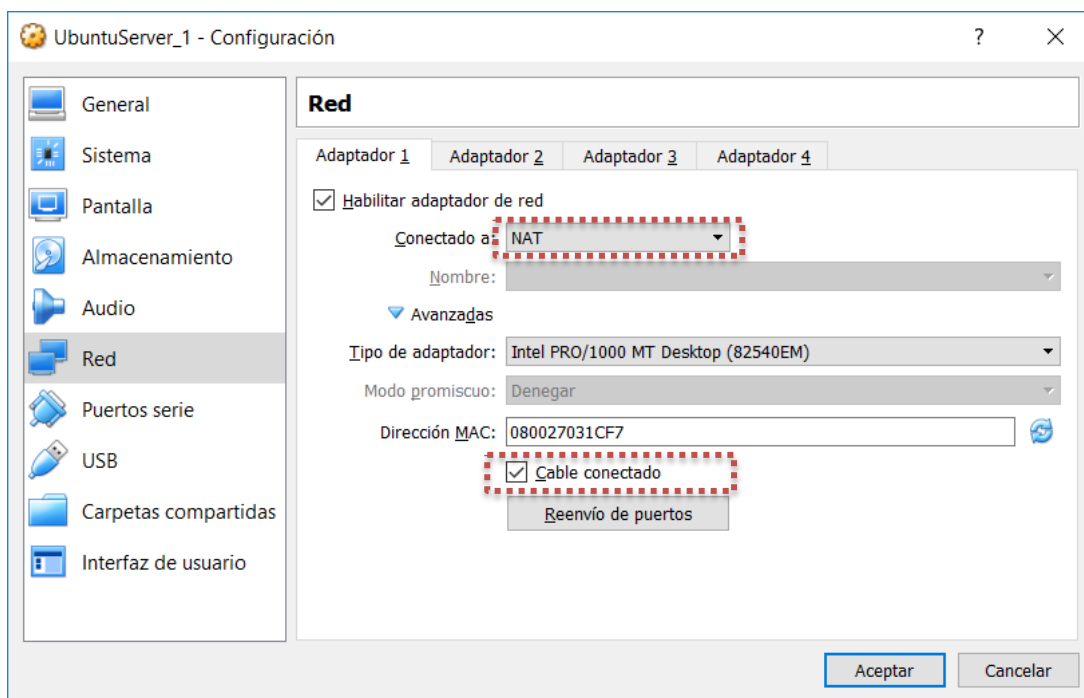
Antes de comenzar es necesario que la máquina Ubuntu que se vayan a utilizar como servidor DHCP tenga conexión a internet, ya que necesita descargar algunos paquetes. Para ello es necesario configurar la red de la máquina virtual.

Esto se realizará en VirtualBox, dentro de la configuración de la red.

Conectado a: **NAT**

Se selecciona, en opciones avanzadas: **Cable conectado**





A continuación, será necesario modificar la configuración permanente de la máquina virtual y establecerla como DHCP.

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet dhcp
```

De esta forma nuestra máquina Ubuntu tendrá acceso a internet y será posible descargar los paquetes de instalación necesarios.

Recuerda que tienes que **reiniciar la máquina Ubuntu** para que tome la configuración permanente modificada.

Instalación de paquetes para la configuración de DHCP

En este apartado se va a estudiar cómo configurar un servidor DHCP en un equipo con un sistema operativo Linux.

Lo primero que se hará será instalar el servidor DHCP, para ello hay que escribir:

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

En ocasiones es necesario realizar la actualización del sistema Operativo. Esto puede hacerse escribiendo:

```
sudo apt-get update
```



Una vez se finalice con la instalación descrita, se deben deshacer los cambios hechos en la red de la máquina virtual en VirtualBox, para poder trabajar en GNS3.

Configuración del servidor DHCP

Una vez instalado el servidor hay que proceder a dar una dirección IP al servidor DHCP. Para ello se abre el archivo que contiene la configuración IP del equipo en Ubuntu escribiendo:

```
sudo nano /etc/netplan/50-cloud-init.yaml
```

y se modifica escribiendo la configuración IP del servidor DHCP:

```
network:
  ethernets:
    enp0s3:
      dhcp4: no
      addresses: [192.168.10.50]
      netmask: [255.255.255.0]
      gateway4: 192.168.10.254
  version: 2
```

Hay que tener en cuenta que el servidor DHCP no se asigna una dirección IP así mismo, por lo que debe tener una dirección estática y que esté fuera del rango de direcciones que luego asignaremos a los clientes DHCP.

A continuación, debe indicarse al servidor a través de qué interfaces debe escuchar las peticiones DHCP. Para ello editamos el archivo `isc-dhcp-server` escribiendo:

```
sudo nano /etc/default/isc-dhcp-server
```

Y editamos la última línea, donde pone:

```
INTERFACES=""
```

En nuestro caso sólo existe un enlace por el que recibir peticiones luego escribimos:

```
INTERFACES="enp0s3" #según esté en vuestro equipo.
```



A continuación, es necesario configurar el servidor DHCP. La configuración del servidor aparece en el archivo: `/etc/dhcp/dhcpd.conf`

Editamos este archivo de tal forma que quede como el que sigue:

```
#Configuration for an internal subnet.
subnet 192.168.10.0 netmask 255.255.255.0 {
    range 192.168.10.10 192.168.10.20;
    option subnet-mask 255.255.255.0;
    option routers 192.168.10.254;
    default-lease-time 600;
    max-lease-time 7200;
}
```

En la configuración se le indica el rango de direcciones IP que va a asignar, la dirección de la puerta de enlace, dirección de difusión y tiempos de arrendamiento de las direcciones. Se salva la configuración del servidor y se reinicia la máquina:

```
sudo reboot
```

En ese momento, el resto de PCs dentro del dominio de este equipo obtendrán una dirección IP de manera dinámica utilizando nuestro equipo como servidor DHCP.

2.2 Ejemplo de servicio DHCP con máquinas Ubuntu

En esta sección se va a probar el servicio DHCP proporcionado por la máquina Ubuntu. Para ello se va a construir una red como la que se muestra en la Figura 1.

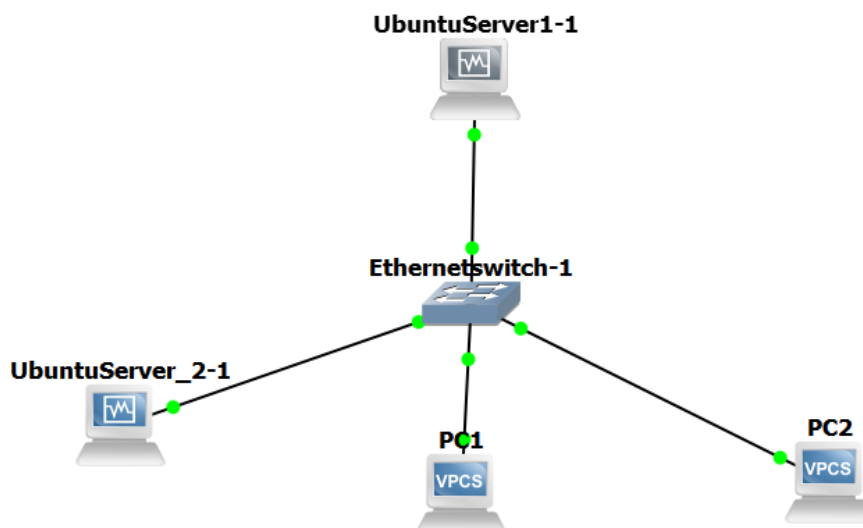


Figura 1. Esquema para la prueba de la configuración de DHCP



En este esquema el equipo UbuntuServer1 es el servidor DHCP y proporciona direcciones IP al resto de los equipos (UbuntuServer_2, PC1 y PC2). Las direcciones IP que se proporcionan están en el rango 192.168.0.1-192.168.0.50, la máscara que se suministra es la 255.255.255.0 y la puerta de enlace que será la 192.168.0.254.

En el equipo Ubuntu que desee obtener la dirección IP vía DHCP será necesario tener configurada DHCP en la configuración permanente de la interfaz (/etc/network/interfaces)

```
auto eth0
iface eth0 inet dhcp
```

En las máquinas Ubuntu se puede liberar la dirección escribiendo:

```
$ sudo dhclient -r eth0
```

- Se puede solicitar una nueva dirección IP vía DHCP mediante el comando:

```
$ sudo dhclient eth0
```

En los VPCS, para obtener una dirección IP mediante DHCP se debe escribir en la consola *dhcp*.

```
VPCS-1> dhcp
```

La liberación de la conexión puede conseguirse escribiendo *dhcp -x* y la renovación escribiendo *dhcp -r*.

Si se ha realizado todo correctamente todos los equipos tendrán una dirección IP asignada automáticamente por el servidor DHCP.

Actividad 2. Reproducir el esquema representado en la Figura 1. Guardar el fichero con el nombre Prac09_Act02.

Mostrar una captura del resultado mostrando la configuración de cada uno de los equipos.

Mostrar el resultado de escribir los comandos:

- `sudo dhclient eth0 -v`: en la máquina Ubuntu
- `dhcp`: en uno de los VPCS.

Modificar la configuración del servidor DHCP para que solo proporcione 2 direcciones IP. ¿Qué ocurrirá en este ejemplo cuando un cuarto equipo haga una solicitud de una dirección IP?

3 Configuración de un router como un servidor DHCP

Además de en un servidor, también es posible configurar el servicio DHCP en un router. Para probar esto, se va a partir del ejemplo anterior y se va a reemplazar el equipo UbuntuServer_1 por un router, tal y como se muestra en la Figura 2.



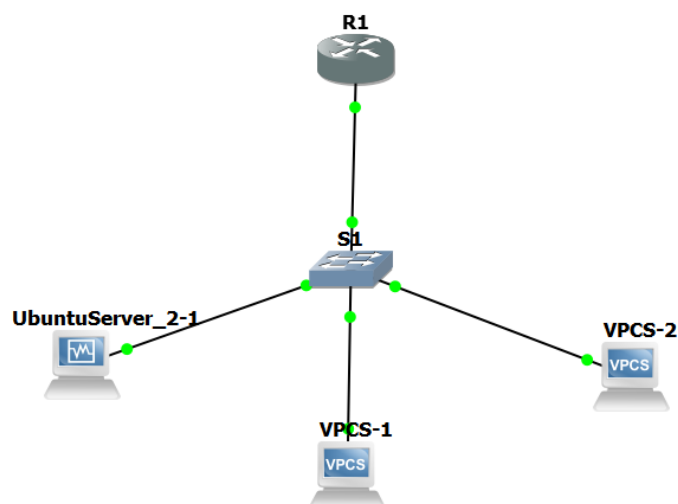


Figura 2. DHCP en un router.

Antes de comenzar con la configuración de DHCP será necesario configurar la dirección de la interfaz del router:

```
Router>enable
Router#configure terminal
Router(config)#interface fastethernet0/0
Router(config-if)#ip address 192.168.0.1 255.255.255.248
Router(config-if)#no shutdown
```

A continuación, se configura el servicio DHCP. Se define el nombre del grupo:

```
Router(config)#ip dhcp pool DHCP_1
```

El comando anterior habilita el modo de configuración de DHCP. Ahora definiremos el rango de direcciones a entregar mediante DHCP y la dirección de la puerta de enlace:

```
Router(dhcp-config)#network 192.168.0.0 255.255.255.248
Router(dhcp-config)#default-router 192.168.0.1
```

En este ejemplo, se utiliza una subred con máscara 255.255.255.248, por lo que podremos asignar un máximo de 6 direcciones IP. Teniendo en cuenta que el router utiliza una dirección, solo podremos asignar direcciones a 5 equipos. Es recomendable que la dirección IP del router esté excluida de las direcciones IP que se asignan, para ello se puede escribir:

```
Router(config)#ip dhcp excluded-address 192.168.0.1
```

Una vez realizada la configuración se recomienda almacenarla escribiendo



```
Router#copy run start
```

Podemos comprobar que se ha asignado la conexión:

```
Router#show ip dhcp binding
```

Si quisiéramos se puede desactivar el servidor DHCP:

```
Router(config)# no service dhcp
```

Actividad 3. Reemplazar sobre la actividad anterior el servidor DHCP Ubuntu por un router Cisco y guardarla como "Prac09_Act03". Seguir los pasos descritos con anterioridad para incorporar el servicio de DHCP en el router. Probar que la asignación dinámica de direcciones funciona en el resto de los equipos.

Explicar el procedimiento seguido y mostrar una captura de pantalla con la red construida en GNS3 y otra captura con el resultado de ejecutar el comando show ip dhcp binding en el router.

IV Bibliografía

Redes de Computadoras. Un enfoque descendente, 5º edición, Jim F. Kurose & Keith W. Ross, Pearson, 2010.

<https://www.linuxfordevices.com/tutorials/ubuntu/dhcp-server-on-ubuntu>

http://www.cisco.com/en/US/docs/ios/12_4t/ip_addr/configuration/guide/htdhcpre.html





Grado en Ingeniería Informática

REDES

PRÁCTICA 10

Análisis de TCP con Wireshark

Docentes:

Alejandro Merino

Daniel Sarabia Ortiz

*Dpto. de Ingeniería Electromecánica
Área de Ingeniería de Sistemas y Automática*

Versión 1.4

Fecha 12/05/2022 11:23

Esta obra está sujeta a la licencia Reconocimiento 4.0 Internacional de Creative Commons. Para ver una copia de esta licencia, visite <http://creativecommons.org/licenses/by/4.0/>



Índice de contenidos

I	OBJETIVOS	3
II	ANÁLISIS ESTADÍSTICO DE TCP	3
III	DESARROLLO DE LA PRÁCTICA:	9
	1 Análisis de los paquetes capturados.....	9
IV	BIBLIOGRAFÍA	12



En esta práctica se comenzará con una explicación de las herramientas estadísticas de las que dispone WireShark para analizar el protocolo del protocolo TCP, posteriormente se propondrá un ejercicio práctico en el que se analizarán de manera práctica y real los aspectos prácticos más importantes analizados en la teoría.

I Objetivos

- Comprender las fases del establecimiento y cierre de una conexión TCP.
- Reconocer en un segmento TCP sus campos y su significado.
- Identificar en una transmisión TCP los números de secuencia y comprender su significado.
- Identificar retransmisiones y retransmisiones rápidas y analizar su funcionamiento.
- Calcular el RTT en una transmisión TCP.
- Diagnosticar el estado de una red a partir del análisis de los diagramas tiempo secuencia.

II Análisis estadístico de TCP

Wireshark dispone de algunas herramientas que permiten hacer diagnósticos de red y analizar el rendimiento de las transmisiones.

Para el análisis de una comunicación usando el protocolo de transporte TCP, resulta muy útil el análisis que se obtiene de las distintas gráficas que proporciona Wireshark.

Para analizar un flujo TCP, es necesario seleccionar un paquete correspondiente al flujo que se desea analizar y seleccionar Statistics->TCP Stream Graphs ->...

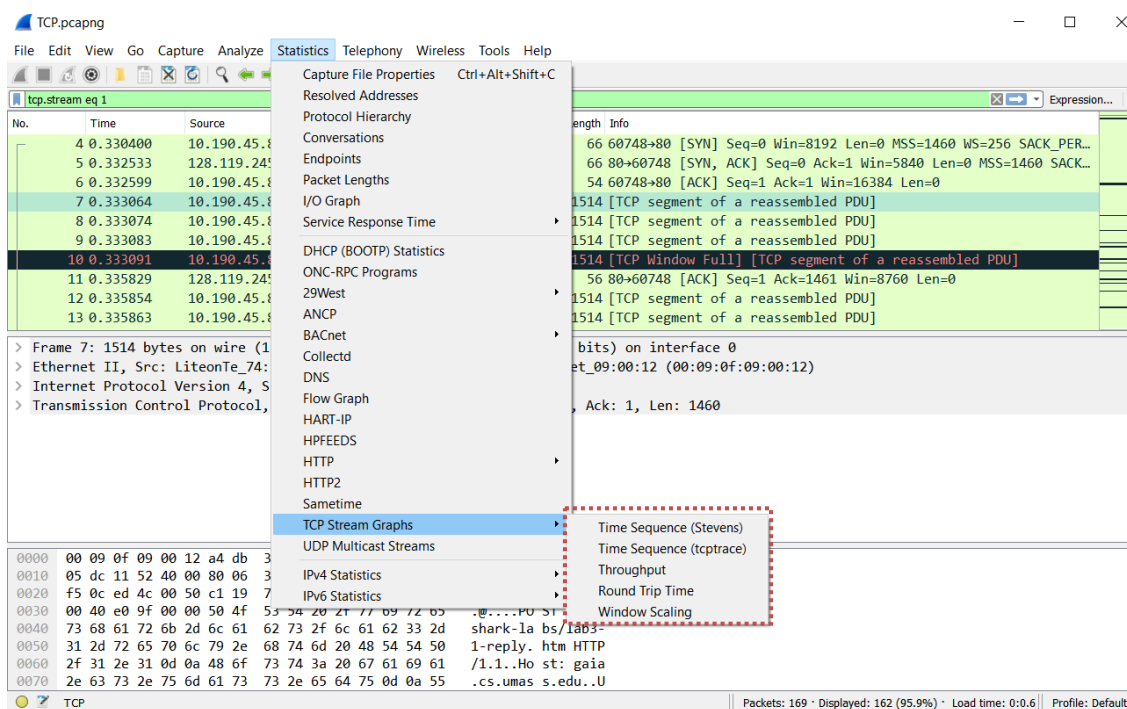


Figura 1. Ubicación de las Stream Graphs en WireShark

Hay varias gráficas que se pueden obtener:

Gráfica Time Sequence (Stevens): Esta gráfica muestra la evolución del número de secuencia con el tiempo, Figura 7. Se muestra el número de secuencia (en Bytes) en el eje de las Y, y el tiempo en el eje de las X. Dado que los números de secuencia representan bytes transmitidos, la pendiente de esta gráfica permite determinar la velocidad de la transmisión. A partir de los datos podemos ver también si hay retransmisiones, zonas en las que se interrumpe la transmisión, etc.

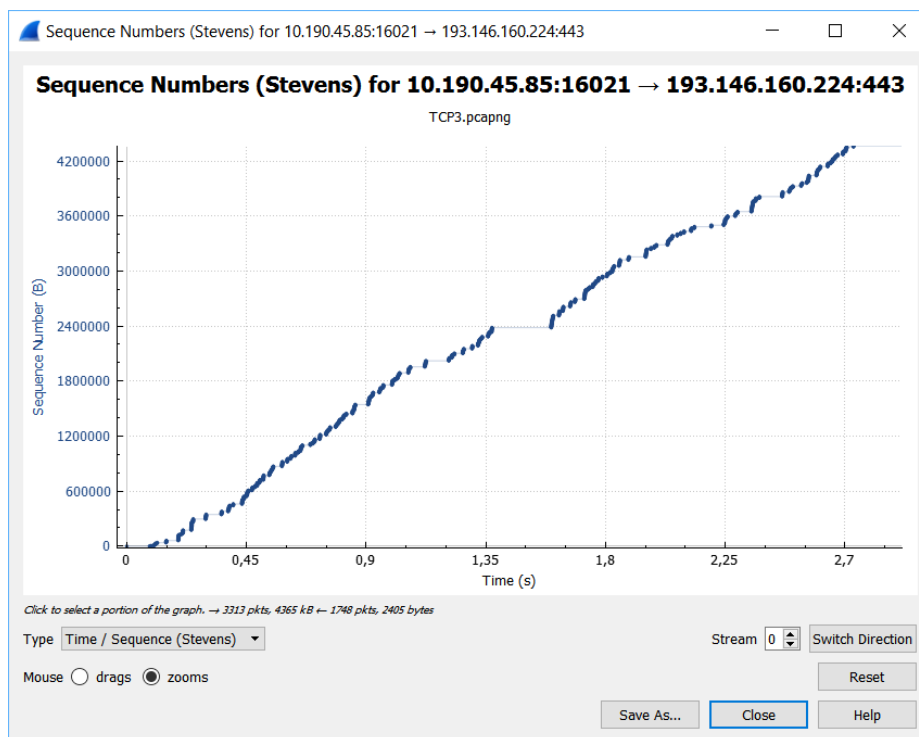


Figura 2. Gráfica de números de secuencia en WireShark

Gráfica Time Sequence (tcp trace): En esta gráfica (Figura 8), al igual que en el caso anterior se muestra la evolución del número de secuencia respecto al tiempo, pero esta gráfica muestra información adicional.



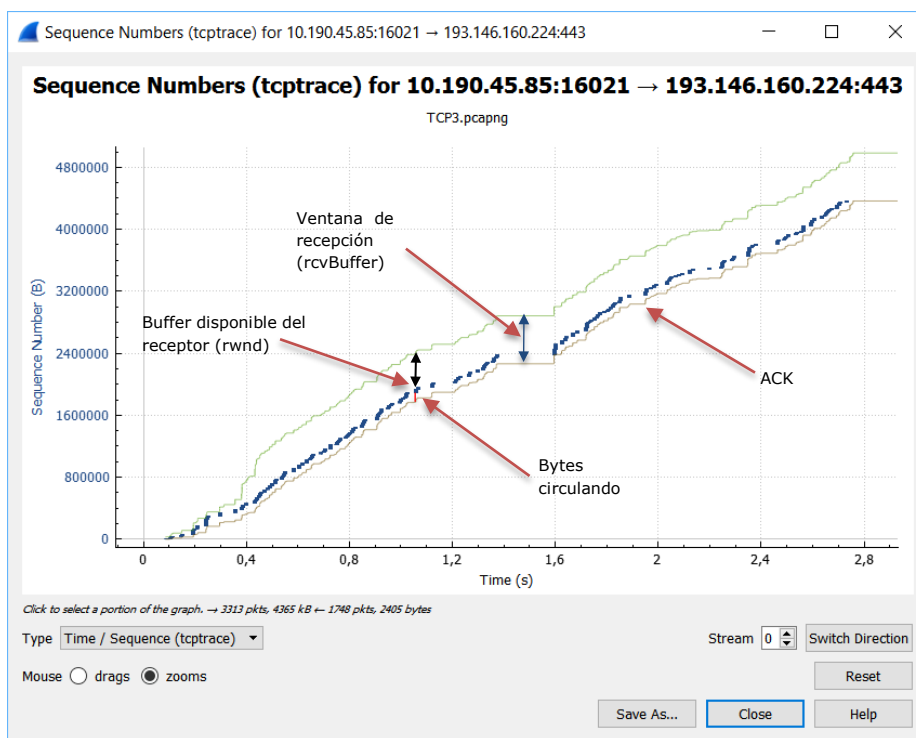


Figura 3. Gráfica TCP trace en Wireshark

La línea central en color azul representa el número de secuencia de los paquetes y sería equivalente a la mostrada en la gráfica anterior.

La línea gris, representa el último número de secuencia que fue reconocido por el receptor en cada instante de tiempo. Es decir, en un instante de tiempo se han enviado una serie de bytes y de ellos hay algunos que ya han sido reconocidos (los indicados en la línea gris) y otros que faltan por reconocer. Los bytes no reconocidos son los bytes que están en ese momento circulando por la red (Bytes in flight). Estos bytes pueden calcularse en la gráfica, mediante la resta entre el número de secuencia enviado en un instante determinado y el número de reconocimiento en ese instante. Wireshark proporciona además este valor calculado (los valores que no se obtienen directamente del paquete si no que son calculados, se muestran en Wireshark se muestran entre corchetes), Figura 9.

```

310 1.592387  10.190.45.85  193.146.160.224  TCP  1514 [TCP
311 1.592409  10.190.45.85  193.146.160.224  TCP  1514 [TCP
312 1.592431  10.190.45.85  193.146.160.224  TCP  1514 [TCP
313 1.592453  10.190.45.85  193.146.160.224  TCP  1514 [TCP
... ..0.. ..0.. = ECN-Echo: Not set
... ..0.. ..0.. = Urgent: Not set
... ..1.. ..0.. = Acknowledgment: Set
... ..0.. ..0.. = Push: Not set
... ..0.. ..0.. = Reset: Not set
... ..0.. ..0.. = Syn: Not set
... ..0.. ..0.. = Fin: Not set
[TCP Flags: .....A....]
Window size value: 256
[Calculated window size: 65536]
[Window size scaling factor: 256]
Checksum: 0x3799 [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
[SEQ/ACK analysis]
  [RTT: 0.085157000 seconds]
  [Bytes in flight: 11680]
  [Bytes sent since last PSH flag: 5840]
[Reassembled PDU in frame: 319]
TCP segment data (1460 bytes)

```

Figura 4. Bytes que están circulando

La línea superior verde muestra el valor de la ventana de recepción disponible por el receptor calculada por el emisor. Este cálculo lo realiza el emisor sumando el número de reconocimiento en el instante actual más el último valor de la ventana de recepción anunciada por el receptor (rcvBuffer), ver Figura 4. Este tamaño de ventana se corresponde con el tamaño total del buffer que está disponible en el receptor en ese



momento. El tamaño de este buffer se corresponde con el espacio en memoria que el receptor destina a la transmisión y es gestionado por el sistema operativo del receptor, por lo que puede aumentar o disminuir durante una transmisión en función de la disponibilidad de memoria en el equipo en cada momento.

Por tanto, la diferencia entre el valor de la ventana de recepción disponible en el receptor calculada por el emisor (línea verde) y el número de secuencia (línea azul), indica los bytes que pueden enviarse sin saturar el buffer del receptor (Figura 10). Por tanto, si estas dos líneas se solapan, el número de bytes sin reconocer es igual al valor de la ventana anunciada por el receptor en el último mensaje que envió al emisor. Esto significa que los bytes que están circulando tienen la capacidad de desbordar al receptor. En esta situación el cliente considera que el buffer del receptor estará lleno e interrumpe el envío de segmentos hasta que reciba algún reconocimiento que permita seguir con la transmisión (Figura 11).

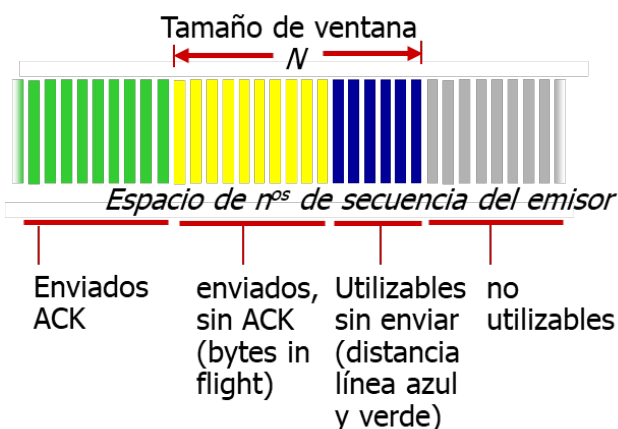
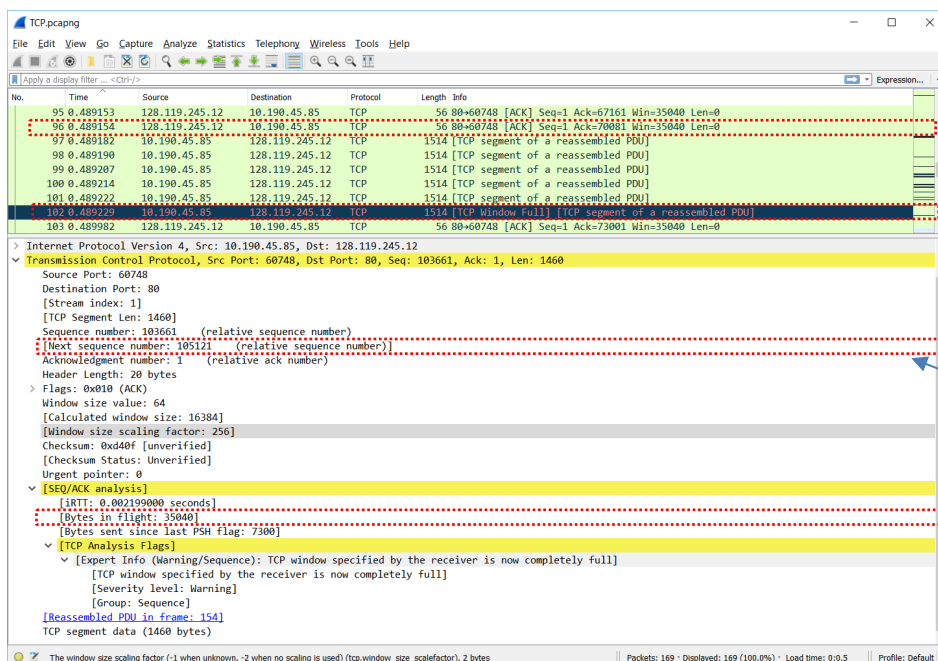


Figura 5. Esquema de ventana deslizante en un emisor TCP



Tamaño del ventana anunciado por el receptor: 35040

Evento de buffer de recepción llena

Solape: El número de secuencia (línea azul) es igual al último ACK reconocido + ventana de recepción anunciada por el receptor (línea verde):
70081 + 35040 = 105521

El número de bytes enviados no reconocidos es igual al tamaño del buffer anunciado por el receptor.

Figura 6. Evento de ventana de recepción llena en una captura de WireShark

Gráficamente se observa en la Figura 12:



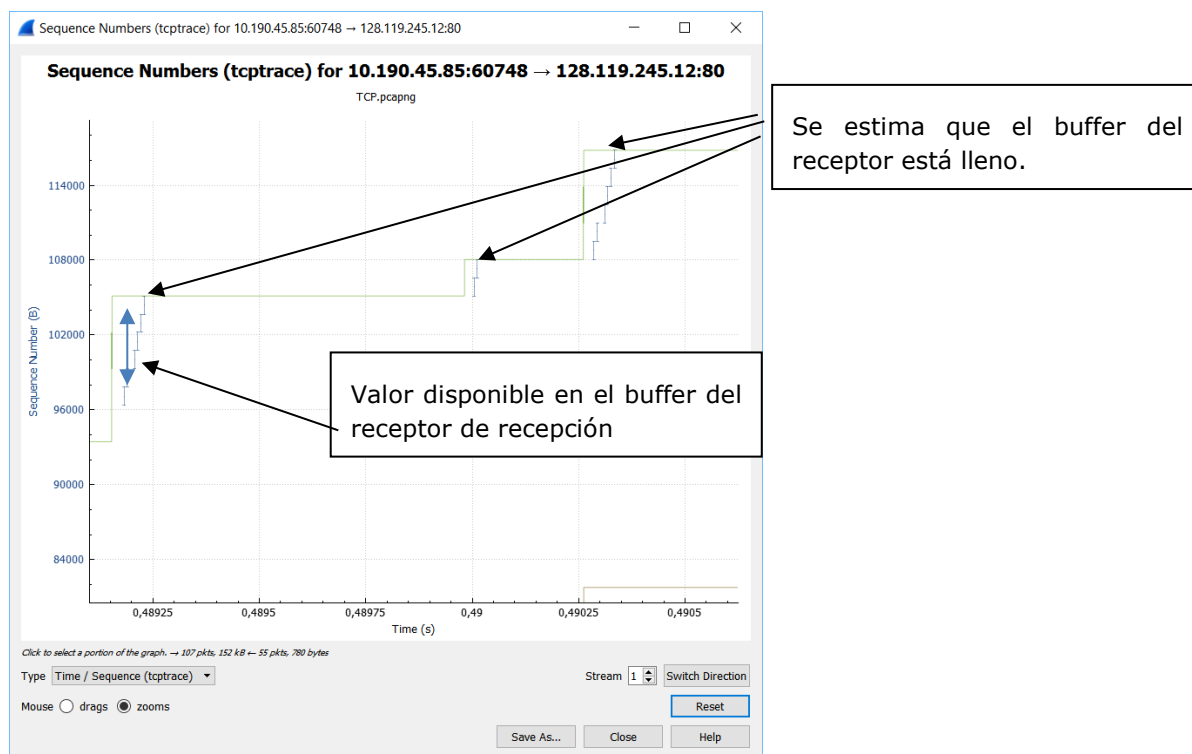


Figura 7. Ventana de recepción llena en la gráfica TCP trace

En transmisiones en las que la línea verde y la azul están separadas, no existirán problemas de ventana de recepción llenas.

La diferencia en horizontal entre las líneas azul y gris representa el tiempo de ida y vuelta de los segmentos, ya que ambos se corresponden con el mismo valor del eje que representa al número de secuencia. Por tanto, cuando existe mucha distancia en horizontal entre la línea azul y la gris, esto indica que están tardando mucho tiempo en recibirse los reconocimientos de los segmentos, por tanto, habrá muchos segmentos circulando y será más probable que se llene la ventana de recepción. Esta distancia en horizontal se puede observar directamente en la gráfica Round Trip Time que se explica a continuación.

Gráfica Round Trip Time (RTT). En esta gráfica se muestra el tiempo de ida y vuelta de los segmentos enviados. Este tiempo se calcula como la diferencia entre el tiempo que transcurre desde que se envía un segmento hasta que se recibe el ACK de ese segmento (Figura 12 y Figura 13).



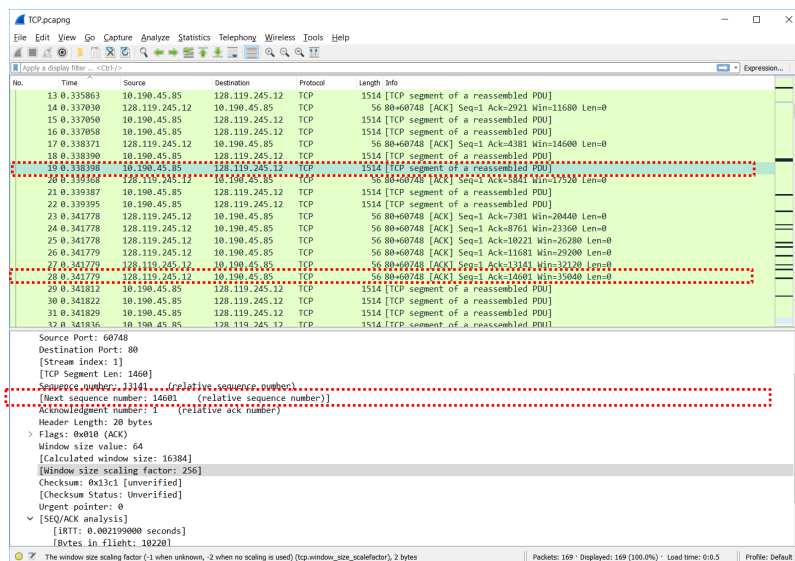


Figura 8. Cálculo del RTT en Wireshark

En la captura anterior podemos ver cómo el segmento 19, recibe la confirmación por parte del segmento 28. El tiempo de ida y vuelta será:

$$0.341779 - 0.338398 = 0.003381$$

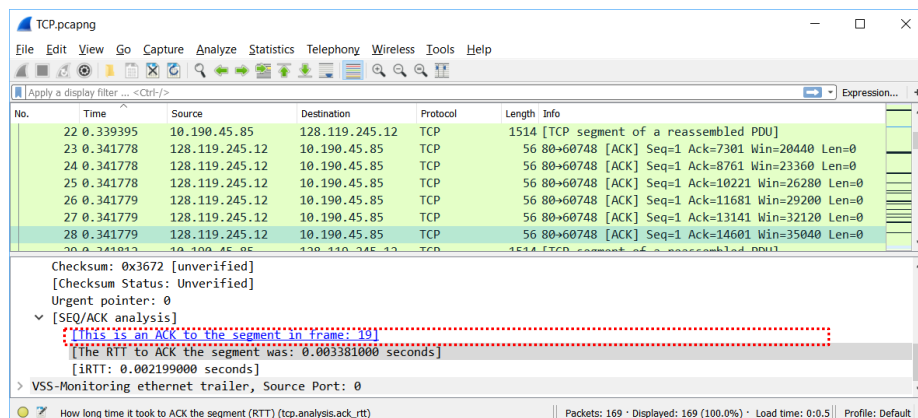


Figura 9. Información de reconocimiento calculada por Wireshark

En la **gráfica del rendimiento (Throughput)** es posible analizar el rendimiento de la transmisión. Usualmente se comienza con una velocidad pequeña que crece exponencialmente (arranque lento). Este crecimiento se produce hasta que se alcance el ancho de banda de la línea o hasta que se limite la velocidad por algún problema en la transmisión. Este límite puede venir impuesto por el valor de la ventana de recepción, que limita la cantidad de segmentos que podemos enviar, o por la ventana da congestión. La velocidad de la transmisión se reduce también cuando se producen timeouts o reconocimientos duplicados. En esta gráfica (Figura 15), los puntos azules representan el tamaño de los segmentos transmitidos y la línea marrón, es el cálculo que se hace de la velocidad de transmisión dividiendo el número de bytes enviados por el RTT. También puede calcularse a partir de la pendiente de la recta tiempo secuencia.



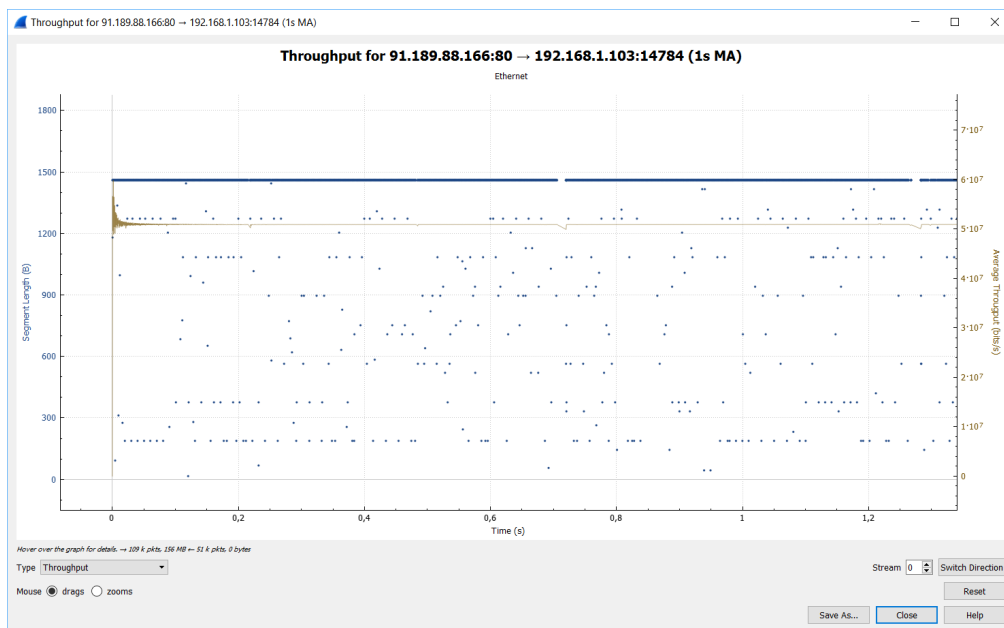


Figura 10. Gráfica de rendimiento en WireShark

III Desarrollo de la práctica:

Una vez vistos los principales conceptos teóricos de TCP, se va a analizar una comunicación TCP utilizando Wireshark.

Para poder analizar el tráfico TCP, se va a descargar y volver a subir un fichero en UBUVirtual. Se analizará el tráfico TCP cuando se produce la subida del fichero. Se recomienda no tener ninguna aplicación que haga uso de la red, para tratar de tener el registro de paquetes lo más limpio posible.

Pasos a seguir:

- Entrar en UBUVirtual y descargar el fichero "Fichero de descarga Práctica 10" que aparece en el tema 5.
- A continuación, se ha creado una tarea: "Subir aquí fichero de la práctica 10" para que pueda subirse el fichero que se ha descargado. Entrar en la tarea, pero **no pulsar aún la subida del fichero**.
- Arrancar Wireshark.
- Comenzar la captura de paquetes con Wireshark, por la interfaz por la que se esté utilizando.
- Inmediatamente después comenzar la subida del fichero descargado anteriormente.
- Una vez finalizada la subida detén rápidamente la captura de paquetes con Wireshark.

1 Análisis de los paquetes capturados.

La manera más rápida de filtrar los elementos correspondientes a la transmisión TCP que nos interesa es tratar de localizar el stream correspondiente a la transmisión TCP de la subida del fichero, para ello, localiza un paquete correspondiente a la transmisión (puedes identificarlo por las IPs de origen y destino y porque habrá muchos paquetes similares).



Una vez localices el paquete haz click sobre le mismo con el botón derecho del ratón y selecciona: Follow->TCP Stream (Figura 16)

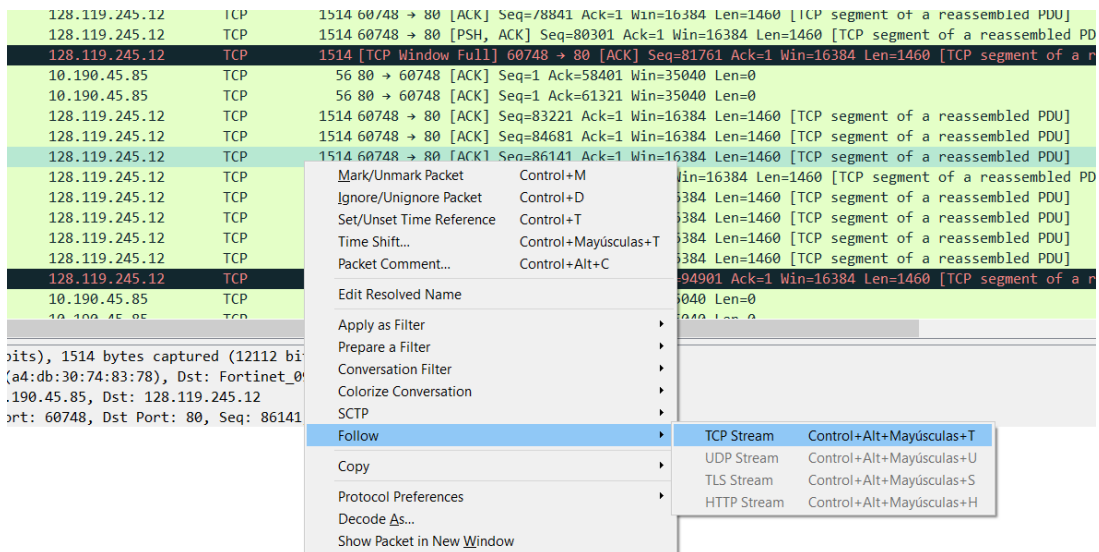


Figura 11. Obtención del Stream de interés en WireShark

Si no eres capaz de localizar el stream, trata de acotar la búsqueda filtrando los segmentos TCP y añadir también en el filtro las IPs de origen y destino, (Figura 17):

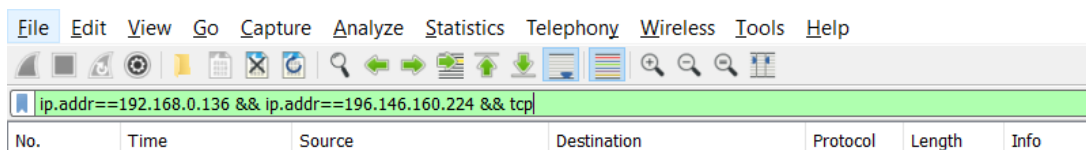


Figura 12. Filtros por IP y protocolo TCP en WireShark

Actividades propuestas.

Establecimiento y cierre de la conexión

1. Indica el stream correspondiente a la transmisión TCP en la que se ha subido el fichero solicitado.
2. Localiza las tres fases correspondientes al establecimiento de la conexión y rellena la siguiente tabla:

	Bit SYN	BIT ACK	Nº sec.	Nº rec	IP origen	IP destino	Puerto origen	Puerto destino
FASE 1								
FASE 2								
FASE 3								

¿Cuál es el tamaño mínimo del tamaño de la ventana de recepción anunciada por el receptor para la transmisión?



¿Cómo varía el tamaño durante la transmisión?

3. Localiza las tres/cuatro fases correspondientes al fin de la conexión:

	Bit FIN	BIT ACK	Nº secuencia	Nº reconoc.	IP origen	IP destino
FASE 1						
FASE 2						
FASE 3						
FASE 4						

Estructura de los paquetes TCP

4. Identificar los campos del paquete TCP visto en teoría.

Paquete seleccionado:

Número de secuencia:

Siguiente número de secuencia: ¿Cómo se calcula?

Número de reconocimiento:

Longitud de la cabecera:

Ventana de recepción:

Puntero de datos urgentes:

5. El cálculo de la carga útil del mensaje se calcula restando al tamaño del mensaje en bytes los tamaños de las cabeceras Ethernet, IP y TCP. Localizar las longitudes de las cabeceras del segmento seleccionado en el ejemplo anterior y comprobar que el tamaño del mensaje es igual a la carga útil más la suma de las cabeceras.

¿Cuánto vale la cabecera Ethernet?

¿Cuánto vale la cabecera IP?

¿Cuánto vale la cabecera TCP?

¿Cuál es la carga útil?

¿Cuál es el tamaño del mensaje?

6. ¿Son coherentes, en los mensajes TCP, la longitud del segmento con el tamaño del mensaje, los números de secuencia y siguiente número de secuencia? ¿Porqué?

7. Identificar 4 segmentos consecutivos TCP del fichero que se ha subido y los ACK del receptor correspondientes a cada segmento. Identificar los tiempos de envío de los segmentos y recepción de los paquetes para calcular el RTT.

	Número de paquete	Tiempo de ida	Nº paquete reconocimiento	Tiempo de vuelta	RTT
Segmento 1					
Segmento 2					
Segmento 3					
Segmento 4					



¿Se producen reconocimientos acumulativos?

8. Estimar los RTT estimados que se utilizarán para el cálculo del Timeout utilizando la fórmula:

$$i. \text{ RTTEstimado} = (1 - \alpha) * \text{RTTEstimado} + \alpha * \text{RTTMuestra}$$

	RTT	RTT estimado
Segmento 1		
Segmento 2		
Segmento 3		
Segmento 4		

Análisis de la transmisión

En los ejercicios siguientes es importante escoger el sentido de la transmisión que va desde vuestro equipo al servidor de UBUVirtual, que es en el sentido en el que ha habido transferencia de ficheros.

9. Representar gráficamente el tiempo frente al número de secuencia (*Time-Sequence-Graph(Stevens)*), incluir una captura de la gráfica y contestar a las siguientes preguntas:

¿Cuántos bytes se han transmitido?

¿Aparecen retransmisiones?, ¿Por qué se puede saber?

¿Qué representaría la pendiente en el gráfico?

¿Aparece alguna retransmisión rápida?

Calcule la velocidad aproximada de la transmisión.

¿LA velocidad de la transmisión ha sido estable?

10. Analizar también la gráfica Tiempo secuencia (tcptrace style).

¿Qué conclusiones puedes sacar de la gráfica que obtienes?

¿Cómo ha sido la transmisión?

¿Obtienes en algún momento ventana de recepción 0? ¿A qué puede ser debido?

¿Qué es lo que limita la velocidad de la transmisión?

IV Bibliografía

https://www.wireshark.org/docs/wsug_html_chunked/ChAdvTCPAnalysis.html

<https://packetbomb.com/understanding-the-tcp-trace-time-sequence-graph-in-wireshark/>

<https://www.packetsafari.com/blog/2021/10/31/wireshark-tcp-graphs/>

<https://blog.nipraas.com/2020/07/basic-tcp-analysis-with-wireshark-part-1.html>



Redes de Computadoras. Un enfoque descendente, 5º edición, Jim F. Kurose & Keith W. Ross, Pearson, 2010.

