

WUOLAH



Enriquem98_489139

www.wuolah.com/student/Enriquem98_489139



170

Apuntes Redes Tema 4.pdf

Resumen Tema 4 de Redes de Computadores



2º Redes de Computadores



Grado en Ingeniería Informática



**Escuela Técnica Superior de Ingenieros Informáticos
UPM - Universidad Politécnica de Madrid**



CUNEF POSTGRADO

La formación que necesitas para tu **futuro profesional**

⇒ **FINANZAS**

⇒ **DATA**

⇒ **DERECHO**

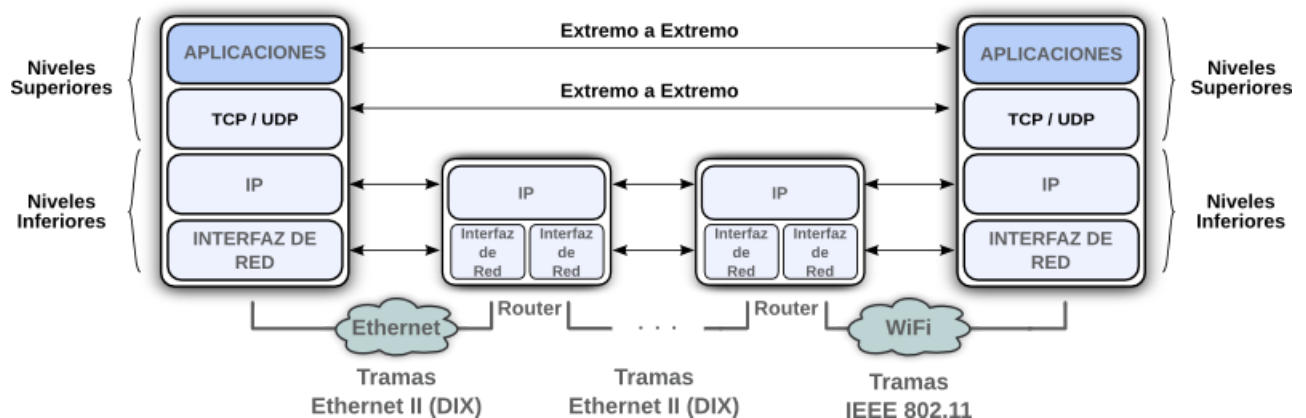
SCIENCE

www.cunef.edu

Tema 4

1.- El Nivel de Transporte de Datos

El **nivel de enlace** es responsable de la entrega de tramas entre dos sistemas vecinos en un mismo enlace (**comunicación sistema a sistema en un enlace**). A su vez, el nivel de red o nivel de Internet es responsable de la entrega de datagramas IP por Internet entre dos sistemas ya sean vecinos o no (comunicación sistema a sistema en Internet). Sin embargo, la comunicación real tiene lugar entre dos procesos de una aplicación (nivel de aplicación). Para ello, es necesario de un nivel de transporte que permita la comunicación entre procesos de aplicación.



Protocolo TCP

El **protocolo TCP**, ofrece extremo a extremo, fundamentalmente, un servicio orientado a conexión y, por tanto, **fiable y con control de flujo**, independientemente de las redes de comunicaciones y *routers* que hayan intervenido entre dos sistemas finales. Las unidades de datos del protocolo TCP se denominan **segmentos TCP**.

El protocolo de control de la transmisión TCP proporciona, al correspondiente proceso de aplicación, un servicio de **flujo de octetos** (*byte-stream*), **orientado a la conexión y fiable, multiplexado y dúplex**. Estos cuatro servicios fundamentales de TCP se desglosan a continuación:

- **Flujo de octetos** (*byte-stream*): Este servicio permite al proceso de aplicación (emisor) transmitir un flujo continuo de octetos a su entidad TCP emisora para que ésta los vaya recogiendo, numerando y agrupando en unidades de datos denominadas segmentos. La entidad TCP emisora calcula un **MSS** (*Maximum Segment Size*) de tal forma que los datagramas IP se correspondan con la MTU de la red de acceso. La entidad TCP receptora pasa al proceso de aplicación (receptor), exactamente, la misma secuencia de octetos y en el mismo orden (sin pérdidas ni duplicaciones).



**5 DÍAS DE FIESTA.
MÁS DE 1000 INVITADOS.
PLAYOTE, PERREO Y RON BARCELÓ.**



EL MOMENTO PARA VIVIR DESALIA ES AHORA
Participa en VIVEAHORADESALIA.COM

www.disfruta-de-un-consumo-responsable.com 37,5°. Promoción válida en España para mayores de 18 años hasta el 10 de mayo de 2019

- **Orientado a conexión** (conexiones lógicas): Este servicio permite, mediante la conexión entre la correspondiente pareja de sockets (cliente y servidor), que la entidad TCP emisora se ponga de acuerdo con la entidad TCP receptora para llevar a cabo todas las funciones de control de errores (fiabilidad) en la fase posterior de transferencia de datos.
 - **Control de errores:** Este servicio TCP abarca tanto el **control de los errores lógicos** (octetos de datos perdidos, desordenados y duplicados) **como físicos** (bits cambiados). Los **errores lógicos** se controlan mediante el uso de:
 - **Temporizadores:** El campo de datos de cada segmento tiene asociado un único temporizador y a su vencimiento (*timeout*), es decir, si el temporizador expira antes de que se confirmen todos los octetos de datos del segmento, se retransmite el segmento.
 - **Números de secuencia:** Se asignan a cada octeto transmitido (y nunca a los segmentos que transportan dichos octetos) un número de secuencia.
 - **Confirmaciones (*Acknowledgments* o **ACKs**):** Permiten a la entidad TCP emisora desactivar los temporizadores asociados a los octetos de datos transmitidos y, finalmente, cuando todos los octetos de datos de la ventana de transmisión se hayan confirmado, girar ésta para seguir enviando datos. El campo de datos de cada segmento de información tiene asociado una confirmación al igual que un temporizador.
 - Los **errores físicos** se controlan mediante dos procesos:
 - **Detección:** Mediante un mecanismo de suma de comprobación que se aplica a todo el segmento.
 - **Corrección:** A través de retransmisiones al vencimiento de los correspondientes temporizadores. La entidad TCP receptora elimina el segmento afectado y no envía una solicitud de retransmisión.
 - **Control de flujo:** Este servicio impide que una entidad TCP emisora “A” transmita más rápidamente de lo que otra entidad TCP receptora “B” es capaz de almacenar y procesar. Tanto “A” como “B” utilizan un protocolo de **ventana deslizante** para el control del flujo de segmentos de información.

- **Multiplexación:** Este servicio permite al protocolo TCP ofrecer un servicio multiplexado, o en paralelo, a los diferentes procesos de aplicación a través de los números de puerto que identifican a dichos procesos.
- **Full-dúplex:** Este servicio permite que la transmisión de información entre las dos entidades TCP sea bidireccional y simultánea.

Control de flujo

La **ventana de recepción (W_R)** controla la numeración de los octetos de datos almacenados en el **buffer de recepción**. Esta ventana define una lista de números de secuencia consecutivos que se corresponden con los octetos que en un momento dado el receptor puede aceptar. **Cualquier octeto cuyo número de secuencia esté fuera del rango esperado de números de secuencia, es automáticamente rechazado.**

Aparte de su propia ventana de recepción, toda entidad TCP dispone, también, de una **ventana de transmisión (W_T)** para controlar la numeración de los octetos de datos almacenados en el **buffer de transmisión**. Esta ventana define una lista de números de secuencia consecutivos que se corresponden con los octetos que en un momento dado el emisor ha enviado sin haber recibido confirmación. El **buffer de transmisión** es una memoria temporal en donde se guarda copia de los octetos contenidos en el campo de datos de cada segmento de información transmitido. Mientras la aplicación envía de forma continua los datos que quiere transmitir, la entidad TCP emisora recoge, numera y almacena estos datos en su **buffer de transmisión**. Seguidamente, la entidad TCP emisora agrupa los octetos de datos almacenados en trozos y les añade una cabecera creando los segmentos que pasa a la entidad IP del nivel inmediatamente inferior. **Si no se recibe una confirmación, al vencimiento del temporizador se retransmite la copia de los datos del correspondiente segmento.**

Aunque puede variar de una implementación TCP/IP a otra, generalmente, para que W_R y W_T estén debidamente **sincronizados y se lleve a cabo un correcto control de flujo, se tiene que cumplir lo siguiente:**

- Inicialmente y **en fase de establecimiento de la conexión**, W_R tiene que coincidir con el tamaño máximo del propio buffer de recepción. Posteriormente, **en fase de transferencia de datos**, W_R va variando puntualmente hasta el tamaño máximo en función de los octetos libres de su buffer de recepción. Se resalta que **el tamaño del buffer de recepción no tiene porqué ser igual del buffer de transmisión**. Para ello TCP dispone de un mecanismo adaptable de asignación de tamaño para su **buffer de transmisión** en función de la ventana de recepción indicada por el otro extremo en la fase de establecimiento de la conexión.
- Se pasan datos al proceso de aplicación **cuando se llena el buffer de recepción**. Salvo que el bit PSH (*Push*) de la cabecera TCP esté activado.

5x1=5, 5x2=10, 5x3=15
la tabla del cinco te la
sabes gracias a la
musiquita porque...



Si quieres ganar entradas dobles
para los mejores festivales

CONTROL
Feel make Feel

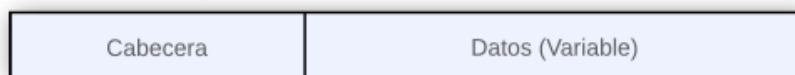
Síguenos

CONTROL
Feel make Feel

- El límite inferior de W_R se corresponde con el primer octeto de datos que se espera recibir después del último octeto pasado al proceso de aplicación.
- Se gira W_R (límites inferior y superior) cuando se pasan datos al proceso de aplicación. Se recuerda que esto último ocurre cuando se llena el *buffer* de recepción.
- Inicialmente y **en fase de establecimiento de la conexión**, W_T tiene que ser menor o igual que la W_R inicial del otro extremo y, por tanto, tiene que ser menor o igual que el tamaño máximo del *buffer* de recepción del otro lado de la comunicación. Posteriormente y **en fase de transferencia de datos**, W_T va variando puntualmente en función de la W_R indicada por el otro extremo.
- Se gira W_T (límites inferior y superior), y se eliminan las correspondientes copias en el *buffer* de transmisión, cuando se recibe confirmación de todos los octetos comprendidos entre el límite inferior y superior de W_T .
- El límite inferior de W_T **debe coincidir** con el límite inferior de W_R .
- El límite superior de W_T **debe coincidir** con el límite superior de W_R .
- **Si el límite superior de W_T es mayor que el límite superior de W_R** , entonces los octetos fuera de W_R **serán descartados**. Por tanto, no se puede girar el límite inferior de W_T antes de tiempo porque también se giraría su límite superior.

Un segmento TCP engloba dos tipos de información:

20 octetos sin opciones



Cabecera de control: Contiene la información de control que manejan las entidades TCP para llevar a cabo sus respectivas funciones. Tiene una **longitud mínima, por omisión, de 20 octetos (sin opciones de servicios adicionales) y máxima de 60 octetos (incluyendo opciones).**

Datos: Incluye la cabecera de información de control del correspondiente proceso de aplicación y los potenciales datos del mensaje de dicha aplicación (si existen). Este campo de datos se corresponde con lo que se entiende como Tamaño Máximo del Segmento (**MSS: Maximum Segment Size**). La entidad TCP emisora calcula un MSS de longitud variable de tal forma que los datagramas IP se correspondan con la MTU de la red de acceso. Para usar los trozos del tamaño adecuado es necesario que el protocolo TCP disponga en su diseño de un **temporizador interno** que le permita recoger una cantidad razonable de octetos de datos antes de crear el segmento de información. El proceso de aplicación envía, de forma continua, a su entidad TCP emisora los datos que quiere transmitir para que dicha entidad TCP recoja, numere y almacene estos datos en su buffer de envío y, finalmente, los agrupe en segmentos.

Bilbao
BBK Live

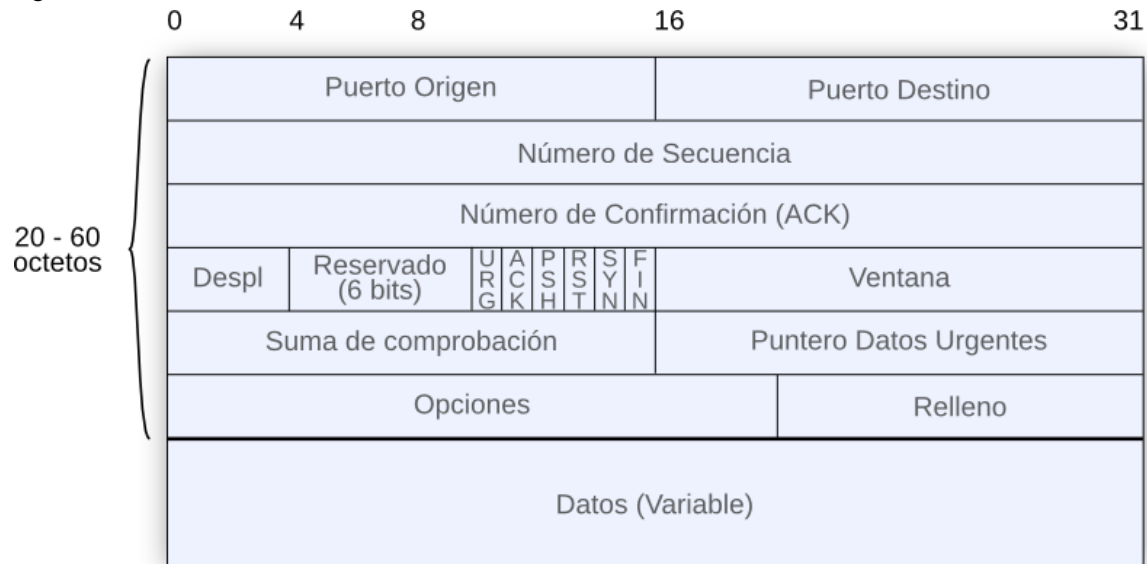
Festival de les Arts

CABO DE PLATA

WUOLAH

Posteriormente, la citada entidad TCP emisora, una vez se ha llenado su buffer de transmisión, toma un trozo de los datos almacenados y le añade una cabecera, creando un segmento. Finalmente, dicha entidad TCP emisora pasa el segmento a la entidad IP para que lo entregue como un único datagrama IP.

Los distintos campos que conforman la cabecera de información de control de TCP son los siguientes:



- **Puerto de Origen** (16 bits)
- **Puerto de Destino** (16 bits)
- **Número de Secuencia** (32 bits): Indica el primer octeto del campo de datos (si tiene un octeto o más) del segmento de información que se va enviar. Como el número de secuencia es de 32 bits, se trabaja en módulo 2^{32} .
- **Número de Confirmación o ACK** (32 bits): Indica el primer octeto del campo de datos del siguiente segmento de información que se espera recibir.
- **Desplazamiento de Datos (Data Offset) o Longitud de la Cabecera** (4 bits): Indica el número de bloques de cuatro octetos que ocupa la cabecera. Por omisión (sin opciones de servicios adicionales) tiene una longitud de 20 octetos (5 bloques o "0101" de 4 octetos). El tamaño máximo será de 60 octetos (15 bloques o "1111" de 4 octetos). Al igual que para el protocolo IP, éste es un campo necesario para reconocer el inicio del campo de datos de usuario, ya que el campo de opciones de servicios adicionales TCP es de longitud variable.
- **Reservado** (6 bits): Seis bits a cero reservados para un uso futuro.
- **URG (Urgent Pointer)** (1 bit): Bit urgente que si está activo indica que el campo Puntero de Datos Urgentes es un campo relevante que la entidad TCP receptora debe analizar debidamente.

- **ACK** (*ACKnowledgment*) (1 bit): Bit de confirmación que si está activo indica que el campo Número de Confirmación o ACK es un campo relevante que la entidad TCP receptora debe analizar debidamente.
- **PSH** (*Push*) (1 bit): Bit de empuje que define un mecanismo para proporcionar un servicio forzado de transferencia de octetos de datos, obligando a que la entidad TCP emisora transmita sin esperar a que se llene su *buffer* de transmisión. A su vez, esto evita que la entidad TCP receptora almacene temporalmente el segmento hasta llenar, a su vez, su *buffer* de recepción.
- **RST** (Reset) (1 bit): Bit de reinicio que si está activo indica a la entidad TCP receptora que abandone la comunicación debido a un error o a una situación anormal. Se usa como respuesta para rechazar una solicitud de establecimiento de una conexión TCP y para un cierre abrupto de ésta en cualquier momento posterior a la fase de establecimiento de la conexión.
- **SYN**(Synchronize) (1 bit): Bit de sincronización que si está activo indica que se está estableciendo una conexión TCP. Asimismo, sincroniza los números de secuencia empleados por las entidades TCP extremo a extremo. Combinándose con el bit ACK proporciona dos segmentos específicos de control TCP (sin datos de usuario).
 - **SYN = 1, ACK = 0: Solicitud de establecimiento de una conexión TCP.**
 - **SYN = 1, ACK = 1: Respuesta afirmativa (¡OK!) a la solicitud previa de establecimiento de una conexión TCP.**
- **FIN** (*Finalize*) (1 bit): Bit de finalización o liberación de la transmisión. La conexión se libera completamente cuando se transmite en cada sentido un segmento sin datos con dicho bit activado. Consecuentemente, un lado de la conexión se puede liberar y el otro seguir activo transmitiendo datos.
- **Ventana** (16 bits): Es la ventana deslizante de recepción (W_R) de la entidad TCP receptora. La ventana W_R puede ser de 1 octeto, 2 octetos hasta $2^{16}-1$ octetos como máximo (65.535 octetos o 16 “unos” en binario). Quiere esto decir, que el *buffer* de recepción es como máximo de 65.535 octetos. También, es válido un tamaño de 0 para indicar que se han recibido los octetos hasta el NÚMERO DE CONFIRMACIÓN (ACK) - 1 pero que el receptor necesita un tiempo extra de reposo y que de momento no desea recibir más datos. El permiso para enviar se transmite, posteriormente, con un segmento con el mismo NÚMERO DE CONFIRMACIÓN (ACK) y un campo VENTANA distinto de cero. En el establecimiento de la conexión cada entidad TCP define el tamaño máximo de la ventana que desea. Posteriormente, durante la fase de transferencia de datos,

este tamaño puede ir variando (se introducen nuevos valores en el campo de Ventana) hasta el máximo definido previamente en la anterior fase de establecimiento de la conexión.

- **Suma de Comprobación**(16 bits): Suma aritmética binaria o en módulo 2 (sin acarreo o suma OR exclusiva) de todos los bloques de 16 bits del segmento completo (cabecera y datos). El procedimiento de cálculo de la suma de comprobación es similar al utilizado para calcular la suma de comprobación de IP salvo por los dos siguientes puntos:
 - Cuando la longitud del segmento no es un múltiplo de 16 bits, el segmento se rellena de ceros hasta hacerlo múltiplo de 16 bits. Sin embargo, el segmento real que se ha de enviar no se modifica por lo anterior.
 - Se incluye una pseudocabecera al inicio del segmento cuando se calcula la suma de comprobación. Esta pseudocabecera, que tampoco se transmite, se crea en el origen y destino durante el proceso de cálculo. Dicho procedimiento le permite al módulo TCP, por un lado, identificar inmediatamente la conexión a la cual pertenece el segmento y, por otro lado, asegurarse de que el segmento ha alcanzado realmente la máquina de destino y el pertinente puerto.
- **Puntero de Datos Urgentes** (16 bits): Cuando el bit URG está activo, el valor de este campo de 16 bits denominado Puntero Urgente sumado al valor del campo Número de Secuencia, apunta al último octeto de los datos urgentes de control. Estos datos urgentes de control requieren un procesamiento especial por parte de la aplicación. Se asume que estos datos urgentes pueden ser avisos, interrupciones o datos de control (por ejemplo, caracteres de control o secuencias de escape) de la propia aplicación que pueden ir entremezclados en el *buffer* de recepción con los datos normales de ésta y se desean que aparezcan antes que dichos datos normales.
- **Opciones** (variable): Concebido para futuras mejoras o extensiones del protocolo TCP y que contiene opciones de servicios extras de dicho protocolo.
- **Datos** (variable): El protocolo TCP calcula un MSS de tal forma que los datagramas IP resultantes se correspondan con la MTU de la red de acceso.

5x1=5, 5x2=10, 5x3=15
la tabla del cinco te la
sabes gracias a la
musiquita porque...



Si quieres ganar entradas dobles
para los mejores festivales

CONTROL
Feel make Feel

Síguenos

CONTROL
Feel make Feel

Todas las opciones se especifican en la fase de establecimiento de la conexión (sólo en aquellos segmentos con el bit SYN = 1).

8 bits	8 bits	Variable
Tipo	Longitud	Datos Opción

Dichas opciones, que se van incorporando poco a poco en las diferentes implementaciones TCP, son las siguientes:

- **Opción 0 - Fin de la lista de Opciones:** Indica el final de la lista de opciones incluidas en el segmento. Es un único octeto (0000 0000) que sólo se incluye cuando el final de las opciones no coincide con el final de la cabecera TCP.
- **Opción 1 - Sin operación - Espacio de relleno:** Es un único octeto (0000 0001) que puede incluirse entre las opciones para que la longitud de la opción correspondiente sea un múltiplo de 4 octetos.
- **Opción 2 - Tamaño Máximo de Segmento (MSS: Maximum Segment Size):** Indica el número de octetos del campo de datos del segmento que la entidad TCP emisora del segmento que transporta esta opción, desea recibir para un procesamiento más óptimo. Esta es una de las opciones más relevantes y, por tanto, más utilizadas con el objetivo de permitir la escalabilidad de los protocolos TCP/IP. A través de esta opción, el protocolo TCP permite manejar diferentes tamaños de *buffers* y segmentos de información.
- **Opción 3 - Factor de Escala de Ventana:** Permite el uso de un tamaño mayor de Ventana (de recepción).
- **Opción 4 - Aviso de Confirmación Selectiva Permitida (SACK PERMITTED: Selective Acknowledgment Permitted):** Indica a la otra entidad TCP, en la fase de establecimiento de la conexión TCP (en un segmento SYN = 1), que puede usar la opción SACK en la fase de transferencia de datos. Si la entidad TCP receptora no ha recibido esta opción en el establecimiento de la conexión, no debe usar la opción Tipo 5 - SACK en la posterior fase de transferencia de datos.
- **Opción 5 - Confirmación Selectiva (SACK: Selective Acknowledgment):** Informa al emisor de bloques no contiguos de datos que han sido recibidos correctamente y, por tanto, qué octetos de datos se han perdido o no han llegado todavía a la entidad TCP receptora.
- **Opción 8 - Marca o sello de Tiempo (Timestamp):** Permite al emisor calcular el valor RTT (*Round Trip Time*: Tiempo de ida y vuelta de un segmento) para configurar sus temporizadores. Se resalta que durante el establecimiento de la

Bilbao
BBK Live

Festival de les Arts

CABO DE PLATA

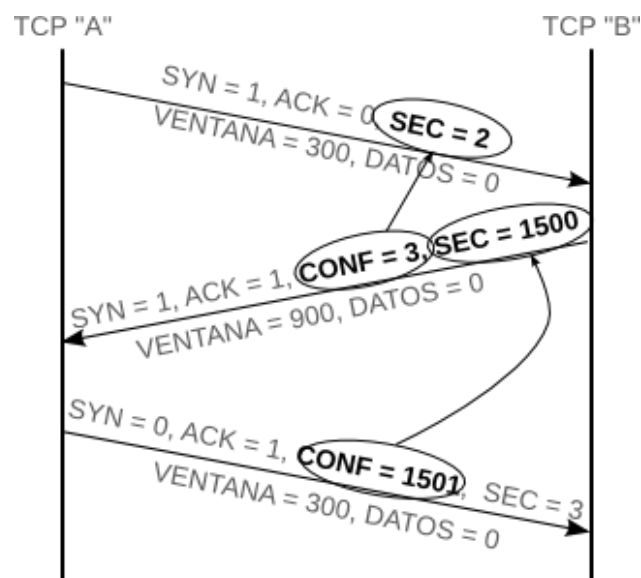
WUOLAH

conexión, el emisor puede enviar esta opción en el primer segmento con SYN = 1, ACK = 0 y puede hacer uso de dicha opción en otros segmentos siempre y cuando haya recibido respuesta con SYN = 1, ACK = 1.

Establecimiento de la conexión

El establecimiento de una conexión entre dos entidades TCP denominadas "A" y "B" se realiza mediante un procedimiento inicial de saludo o diálogo basado en el intercambio de tres segmentos de control (sin datos o Datos=0). Durante esta fase, se informa de los números de secuencia iniciales empleados y del tamaño máximo de la ventana de recepción.

En la siguiente figura se muestra un ejemplo de establecimiento de conexión. Sólo se indican los campos más significativos que aparecen en las correspondientes cabeceras de información de control.



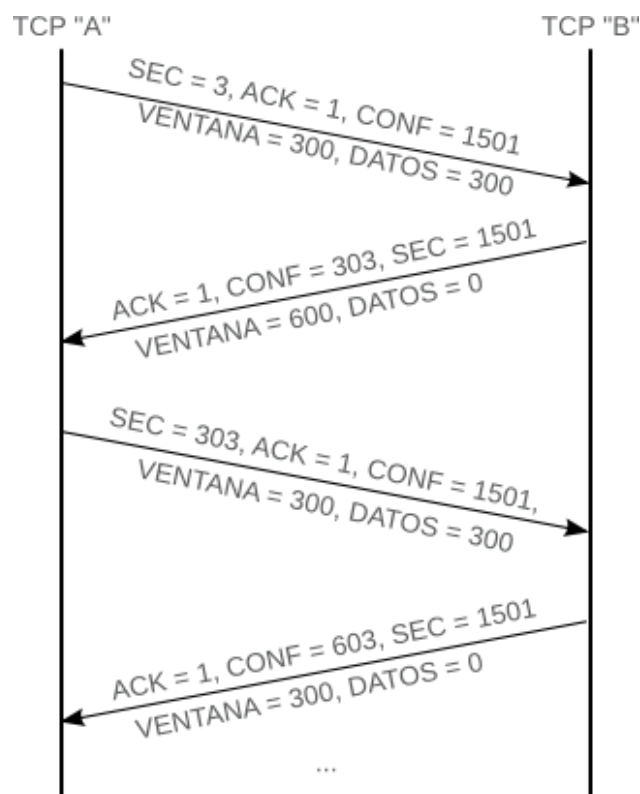
- **Primer segmento (SYN = 1, ACK = 0, SEC = 2,...):** Se usa para que la entidad TCP "A" se ponga de acuerdo con la entidad TCP "B" con el objetivo de llevar a cabo todas las funciones de control de errores y flujo en la fase posterior de transferencia de datos. Además, indica el número de secuencia (SEC = 2) que desea usar la entidad TCP "A" y el tamaño actual de su ventana (300 octetos).
- **Segundo segmento (SYN = 1, ACK = 1, CONF = 3, SEC = 1500,...):** Se usa para confirmar (CONF = 3) el número de secuencia "2" del otro extremo de la comunicación ("A"), y para indicar el número de secuencia (SEC = 1500) que va a emplear la entidad TCP "B". Por tanto, el primer octeto de datos, enviado por

“A”, se numerará con el valor 3. Asimismo, la entidad TCP “B” indica el tamaño de su ventana (900 octetos).

- **Tercer segmento (SYN = 0, ACK = 1, CONF = 1501,...):** Se usa para confirmar (CONF = 1501) el número de secuencia “1500” que va a utilizar el otro extremo de la comunicación. Por tanto, el primer octeto de datos, enviado por “B”, se numerará con el valor 1501. Asimismo, la entidad TCP “A” recuerda el tamaño de su ventana (300 octetos).

Transferencia de datos

En la siguiente figura se muestra una transmisión unidireccional de segmentos de información siempre de la entidad TCP “A” a la entidad TCP “B” en modo semidúplex una vez se ha llevado a cabo el establecimiento de la conexión. Se supone que en dicha transmisión no se producen errores y se dispone de una ventana máxima de recepción en “A” de 300 octetos y en “B” de 900 octetos.



- **Primer segmento de “A” a “B” (SEC = 3, ACK = 1,...):** La entidad TCP “A” transmite un primer segmento de información conteniendo 300 octetos en el campo de datos (DATOS = 300). El primer octeto de dicho campo de datos está

identificado con el número 3 (SEC = 3). Asimismo, confirma (CONF = 1501) el número de secuencia "1501" que va a utilizar "B" y recuerda el tamaño de su ventana (300 octetos).

- **Segundo segmento de "B" a "A" (ACK = 1, CONF = 303,...):** La entidad TCP "B" envía un segmento sin datos (DATOS = 0) y confirma todos los octetos recibidos (CONF = 303). Asimismo, "B" recuerda el número de secuencia (SEC = 1501) que va a utilizar y el tamaño actual de su ventana (600 octetos).
- **Tercer segmento de "A" a "B" (SEC = 303, ACK = 1,...):** La entidad TCP "A" transmite un tercer segmento de información conteniendo 300 octetos en el campo de datos (DATOS = 300). El primer octeto de dicho campo de datos está identificado con el número 303 (SEC = 303). Asimismo confirmar (CONF = 1501) el número de secuencia "1501" que va a utilizar "B" y recuerda el tamaño actual de su ventana (300 octetos).
- **Cuarto segmento de "B" a "A" (ACK = 1, CONF = 603,...):** La entidad TCP "B" envía un segmento sin datos (DATOS = 0) y confirma todos los octetos recibidos (CONF = 603). Asimismo, "B" recuerda el número de secuencia (SEC = 1501) que va a utilizar y el tamaño actual de su ventana (300 octetos).

5x1=5, 5x2=10, 5x3=15
la tabla del cinco te la
sabes gracias a la
musiquita porque...



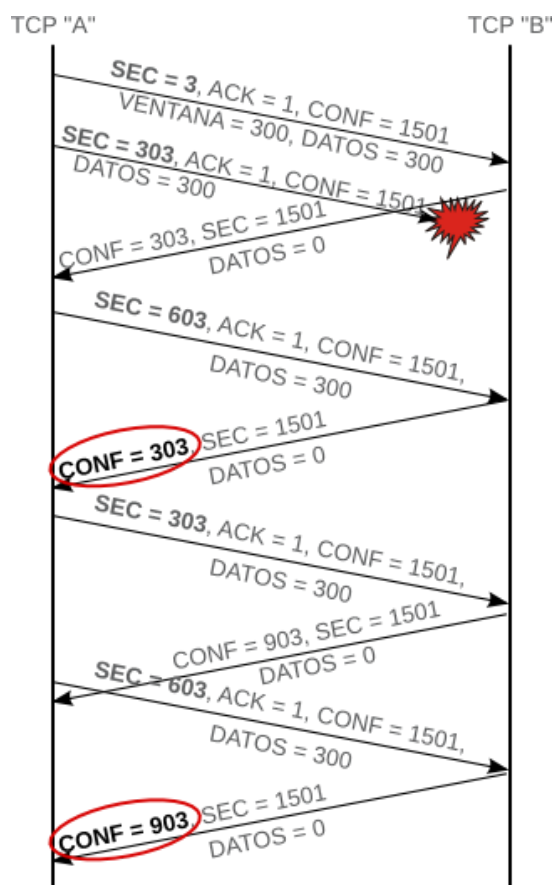
Si quieres ganar entradas dobles
para los mejores festivales

CONTROL
Feel make Feel

Síguenos

CONTROL
Feel make Feel

En la siguiente figura se muestra, en fase de transferencia de datos, una transmisión unidireccional de datos (siempre de la entidad TCP "A" a la entidad TCP "B"). Asimismo, se asume que en dicha transmisión se producen errores, disponiéndose de una ventana máxima de recepción en "A" de 300 octetos y en "B" de 900 octetos.



En este escenario, la entidad TCP "A" transmite 900 octetos (agrupados en tres segmentos) pendientes de confirmación. Así, el primer octeto está identificado con el número 3 (SEC = 3). Se asume que el segundo segmento se pierde por el camino. La entidad TCP "B" ante la llegada de los 300 primeros octetos procede a confirmarlos (CONF=303). La llegada a la entidad TCP "A" de la primera confirmación (CONF=303) produce una desactivación del temporizador. Cuando llega a la entidad TCP "B" el tercer segmento de información, vuelve a transmitir la anterior confirmación (CONF=303). Sólo al vencer los temporizadores (por no recibir las confirmaciones correspondientes) de los dos últimos segmentos de información enviados por "A", éstos se transmiten de nuevo.

Posteriormente, cuando "B" recibe el segundo segmento retransmitido por "A", procede a confirmar los octetos de datos de dicho segmento (CONF=903). El mismo procedimiento ocurre (CONF=903) ante la llegada otra vez a "B" del tercer segmento, el cual todavía estaba sin confirmar.

Bilbao BBk Live
Bilbao

Festival de les Arts

CABO DE PLATA

WUOLAH

Gestión de temporizadores

Con respecto a los valores que tienen que tener los temporizadores de espera de respuesta, no es lo mismo que el emisor y el receptor estén en una misma red de área local o dispersos geográficamente por Internet. La elección de un valor adecuado tiene una consecuencia directa en el funcionamiento eficiente de TCP:

- Si el temporizador es demasiado alto, el emisor esperará innecesariamente en ciertos casos por confirmaciones que nunca llegarán
- Si el temporizador es demasiado bajo se producirán retransmisiones innecesarias de segmentos que habían sido correctamente recibidos

La dificultad de manejar los temporizadores radica en que hay que determinar “aproximadamente” el periodo de tiempo existente desde que se envía un segmento hasta que se recibe su ACK, es decir, el tiempo de ida y vuelta (**RTT: Round Trip Time**). Dicho RTT se puede calcular haciendo uso de marcas de tiempo mediante la **opción número 8 de TCP**.

En este contexto, el protocolo TCP se enfrenta a las siguientes dificultades:

- Diferencias en capacidad y retardo de unas conexiones a otras.
- Oscilaciones debidas a la presencia de *routers* y situaciones de congestión que están fuera de su control.
- Aún en situaciones de control, los *routers* pueden tener largas colas de datagramas IP que atender, los enlaces pueden ser de diferentes velocidades y la ruta puede variar durante la conexión

Toda implementación TCP/IP debe disponer de un algoritmo adaptable de retransmisión muy dinámico (algoritmo de Karn) para calcular el RTT e ir adaptando los correspondientes temporizadores en función de los diferentes destinos. Básicamente, el proceso consiste en ir midiendo los RTTs desde que se envía un segmento de información hasta que se recibe su confirmación. Finalmente, el temporizador se obtiene mediante un promedio de las muestras de cada segmento transmitido más un determinado margen de seguridad.

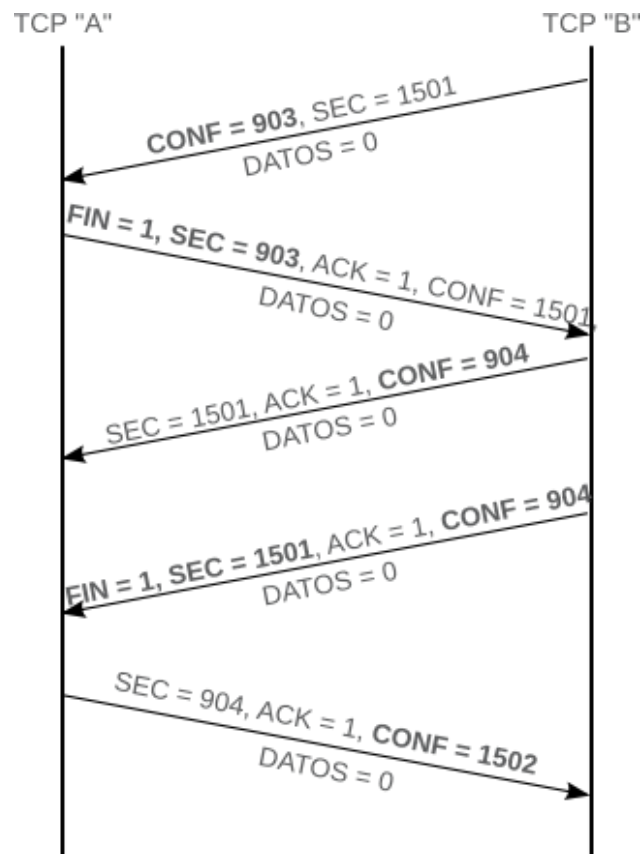
Liberación de una conexión

Una vez terminada la fase de transferencia de datos, se procede a liberar de forma ordenada la conexión previamente establecida. Para liberar cada lado de la conexión, es necesario haber recibido previamente una confirmación de todos los octetos enviados. Esta liberación ordenada implica un cierre independiente en cada dirección de la conexión mediante un proceso de diálogo similar al del establecimiento. Cualquiera de las partes puede solicitar la liberación de la conexión. Conceptualmente, este proceso suele ser el siguiente:

- TCP “A”: “He terminado” (FIN). “No tengo más datos que transmitir”.
- TCP “B”: “OK”.

- TCP "B": "Yo también he terminado" (FIN). "No tengo más datos que transmitir".
- TCP "A": "OK".

En la siguiente figura se muestra una finalización de una conexión. La conexión finaliza completamente cuando se transmite en cada sentido un segmento sin datos con el bit FIN activado.



- **Primer segmento de "A" a "B" (FIN = 1, SEC = 903, ...):** La entidad TCP "A" transmite un segmento sin datos con el bit FIN activado.
- **Segundo segmento de "B" a "A" (ACK = 1, SEC = 1501, CONF = 904, ...):** La entidad TCP "B" confirma (CONF = 904), con un segmento sin datos, la solicitud de liberación propuesta por el otro extremo "A" de la comunicación.
- **Tercer segmento de "B" a "A" (FIN = 1, SEC = 1501, ...):** La entidad TCP "B" transmite un segmento sin datos con el bit FIN activado.

- **Cuarto segmento de “A” a “B” (ACK = 1, CONF = n+1, ...):** La entidad TCP “A” confirma (CONF = 1502) la solicitud de liberación propuesta por el otro extremo “B” de la comunicación.
- **Las dos partes pueden iniciar la liberación simultáneamente.** En este caso, dicha liberación de la conexión se completa cuando una de las partes ha enviado una confirmación (ACK = 1, ...). Como no hay confirmaciones de confirmaciones, al transmitirse la última confirmación, **se lanza un temporizador** con un tiempo de estimación de la llegada de dicha confirmación al destino en función de los temporizadores de segmentos de información utilizados en la fase de transferencia de datos. Con este tiempo, **la entidad TCP que trasmite la última confirmación (ACK = 1) arranca un temporizador con un valor inicial dos veces el tiempo máximo de vida del segmento (MSL: Maximum Segment Lifetime).** Se entiende por **MSL**, el tiempo máximo que puede permanecer un segmento en la red antes de que sea descartado. Al vencer el citado plazo de espera se libera oficialmente la conexión. Con este temporizador se da tiempo a que llegue la confirmación (ACK = 1, ...) a su destino y a recibir potenciales segmentos retrasados u obsoletos para su eliminación inmediata. Asimismo, se asume que durante este tiempo **no se deben usar unos mismos números de secuencia como medida de protección contra segmentos retrasados u obsoletos.**
- **Cualquiera de las partes puede invocar una liberación abrupta,** es decir, finalizar en cualquier momento tanto en la fase de transferencia de datos como en la fase de liberación. Este tipo de terminación se produce cuando una de las partes ha detectado un problema irrecuperable. Dicha liberación se lleva a cabo mediante el **envío de uno o más segmentos de control con el bit RST (RESET) activado.** Esto causa que TCP descarte cualquier dato almacenado en el buffer de transmisión listo para su salida.
- Finalmente, se puede producir una liberación de la conexión TCP en donde una de las partes (“B”) transmite datos después de que la otra (“A”) haya enviado un segmento con el bit FIN activado. Por tanto, el lado de la conexión de “A” a “B” se cierra. Sin embargo, la conexión de “B” a “A” continúa abierta ya que “B” desea transmitir octetos de datos a la entidad “A”.

En la siguiente figura, la entidad TCP “A” envía un primer segmento sin datos con el bit FIN activado, SEC=903 y CONF=1501. Seguidamente, la entidad TCP “B”

5x1=5, 5x2=10, 5x3=15
la tabla del cinco te la
sabes gracias a la
musiquita porque...



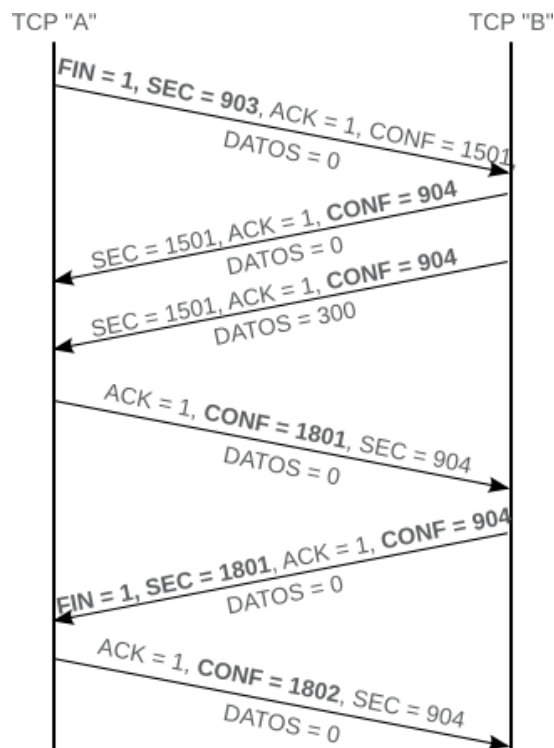
Si quieres ganar entradas dobles
para los mejores festivales

CONTROL
Feel make Feel

Síguenos

CONTROL
Feel make Feel

transmite un segmento con CONF=904. Posteriormente, "B" transmite un segmento de 300 octetos de datos, los cuales son confirmados por "A". Una vez la entidad TCP "B" termina de transmitir toda su información, procede a enviar un segmento sin datos con el bit FIN activado, SEC=1801 y CONF=904. Finalmente, la entidad TCP "A" transmite un último segmento con CONF=1802, completándose la liberación de la conexión TCP.



Bilbao BBk Live
Bilbao


Festival de les Arts

CABO DE PLATA

WUOLAH

Protocolo UDP

El protocolo UDP (User Datagram Protocol) ofrece, extremo a extremo, un **servicio no orientado a conexión y, por tanto, no fiable y sin control de congestiones en el nivel de transporte**. El protocolo UDP se utiliza en los siguientes escenarios:

- Aplicaciones interactivas (audioconferencias, videoconferencias, VoIP, etc.) y no interactivas (streaming de audio y vídeo, etc.) en tiempo real
- El intercambio de mensajes es muy escaso y los mensajes son cortos, por ejemplo, consultas al DNS
- Los mensajes se producen regularmente y no importa si se pierde alguno: SNMP, NTP, ...
- Los mensajes se envían en una RAL del tipo *Ethernet* (sin errores físicos): DHCP, SNMP, ...
- Para tráfico broadcast/multicast

Es un protocolo que proporciona dos servicios que no ofrece el protocolo IP:

- **Detección opcional, y sin recuperación, de errores físicos:** Se comprueba opcionalmente la integridad del datagrama UDP completo, es decir, cabecera y datos. Se recuerda que el protocolo IP sólo verifica la integridad de su cabecera de control. Para efectuar este servicio, el protocolo UDP utiliza un mecanismo de suma de comprobación similar al empleado por el protocolo IP.
- **Multiplexación (origen) y demultiplexación (destino) en función de los números de puerto.** Estas funciones principales, del protocolo UDP, proporcionan un servicio añadido al protocolo IP, el cual es capaz de entregar datagramas IP a una máquina determinada en Internet; pero incapaz de hacer lo propio con respecto a los procesos de usuario del nivel de aplicación. Para llevar a cabo dichas funciones, UDP utiliza un mecanismo basado en números de puerto.

Asimismo, y al contrario que con el protocolo TCP, la entidad del nivel de aplicación **debe pasar los datos bien delimitados**, es decir no ofrece un servicio de flujo de octetos (*byte-stream*).

Gracias a este protocolo se evita toda la sobrecarga añadida de enviar y recibir las múltiples unidades de datos necesarias para establecer y liberar una conexión.

Dado que los protocolos IP y UDP proporcionan servicios no orientados a conexión no fiables; en caso de que la aplicación requiera fiabilidad, y transporte rápido UDP, **será necesario implementar la fiabilidad en el nivel de aplicación**.

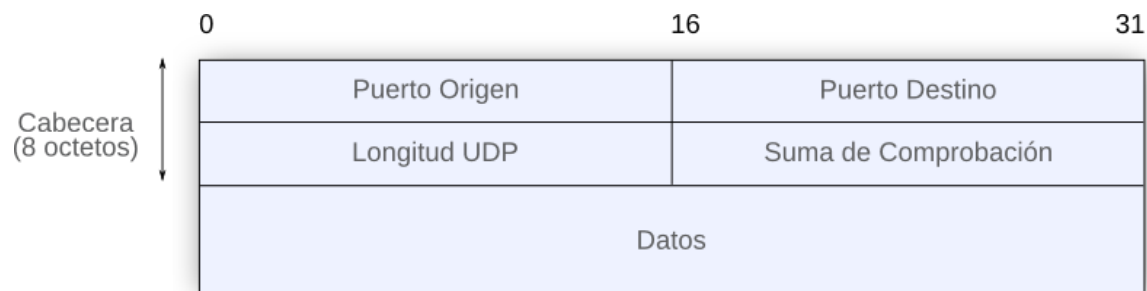
Formato de un datagrama UDP

Un datagrama UDP engloba dos tipos de información:



- **Cabecera de control:** Contiene la información que manejan las entidades UDP para llevar a cabo sus respectivas funciones.
- **Datos:** Incluye la cabecera de información de control de la correspondiente aplicación y los potenciales datos del mensaje de dicha aplicación (si existen). La entidad del nivel de aplicación debe pasar los datos en bloques bien delimitados ya que no existe un servicio de flujo de octetos (*byte-stream*) como en TCP.

Concretamente, los campos de esta cabecera de control son los siguientes:



- **Puerto Origen** (16 bits)
- **Puerto Destino** (16 bits)
- **Longitud UDP** (16 bits): Indica la longitud en octetos del datagrama UDP completo incluyendo la cabecera y los datos. Como mínimo un datagrama UDP tiene una longitud de 8 octetos. A pesar de que la máxima longitud de un datagrama UDP puede ser de 65535 octetos **no es común ver datagramas UDP mayores de 512 octetos**. Por otro lado, no sería necesario el campo Longitud UDP ya que en un paquete IP hay un campo longitud total y otro que define la longitud de la cabecera. Por tanto, si se resta el valor del segundo campo al primero, se puede deducir la longitud del datagrama UDP que está encapsulado en un datagrama IP. Sin embargo, se usa el campo Longitud UDP por una cuestión de eficiencia para la entidad UDP destino.

- **Suma de Comprobación**(16 bits): Suma aritmética binaria o en módulo 2 (sin acarreo o suma OR exclusiva) de todos los bloques de 16 bits del datagrama UDP completo (cabecera y datos). Si la longitud del datagrama no es múltiplo de 16 bits, el datagrama se rellena de ceros hasta hacerlo múltiplo de 16 bits. **Si se detecta un datagrama UDP con errores físicos se elimina y no se entrega al proceso de aplicación.** Su uso es **opcional**, es decir, **si una entidad UDP de origen no desea** calcular la suma de comprobación, **este campo debe contener sólo ceros**, de forma que la entidad UDP de destino sepa que la suma de comprobación no se ha calculado. Puede suceder que la entidad UDP de origen calcule la suma de comprobación y encuentre que **el resultado es cero**. En este caso se establece un **campo de suma de comprobación con todo a unos**. Al igual que en TCP, el procedimiento de cálculo es similar al utilizado por IP salvo por los dos siguientes puntos:

- Cuando la longitud del datagrama UDP no es un múltiplo de 16 bits, el datagrama se rellena de ceros hasta hacerlo múltiplo de 16 bits. Sin embargo, el datagrama real que se ha de enviar no se modifica por lo anterior.
- Se incluye una pseudocabecera al inicio del datagrama UDP cuando se calcula la suma de comprobación. Esta pseudocabecera que tampoco se transmite, se crea en el origen y destino durante el proceso de cálculo. Dicho procedimiento le permite al módulo UDP, por un lado, identificar inmediatamente la comunicación a la cual pertenece el datagrama UDP y, por otro lado, asegurarse de que dicho datagrama UDP ha alcanzado realmente la máquina de destino y el pertinente puerto.

5x1=5, 5x2=10, 5x3=15
la tabla del cinco te la
sabes gracias a la
musiquita porque...



Si quieres ganar entradas dobles
para los mejores festivales

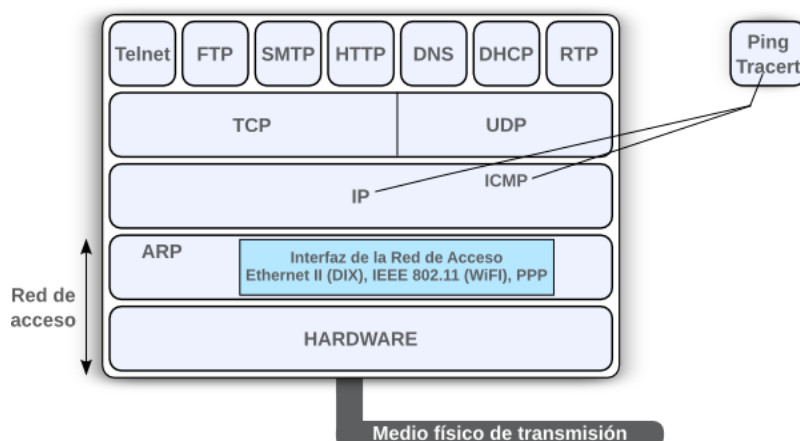
CONTROL
Feel make Feel

Síguenos

CONTROL
Feel make Feel

2.- Nivel de Aplicación:

En el nivel de aplicación de la arquitectura TCP/IP se podrían ejecutar los siguientes protocolos:



- **TELNET:** Ofrece un servicio de terminal remoto sobre TCP.
- **FTP:** Proporciona un servicio fiable de transferencia de ficheros sobre TCP.
- **SMTP:** Ofrece un servicio de envío de correo sobre TCP.
- **HTTP:** Proporciona un servicio sobre TCP de copiado de páginas en formato HTML para su visualización o interpretación local en la máquina del usuario.
- **DNS:** Ofrece un servicio de traducción de una dirección simbólica en su correspondiente dirección IP vía UDP.
- **DHCP:** Proporciona toda la información de configuración TCP/IP incluyendo una dirección IP para que cualquier sistema pueda crear, inmediatamente, su tabla de encaminamiento IP y acceder a una red TCP/IP o a Internet.

En este contexto, conviene distinguir entre servicios del nivel de aplicación que siguen o no el modelo cliente-servidor:

- Se rigen según un protocolo de comunicaciones del nivel de aplicación:
 - Telnet, FTP, SMTP, HTTP (Web), NFS, DNS, DHCP, etc.
- No se rigen según un protocolo de comunicaciones y sí en base a comandos:
 - PING, IPCONFIG (IFCONFIG en Unix), ARP, TRACERT (TRACEROUTE en Linux), NSLOOKUP, etc.

Bilbao BBK Live
Bilbao

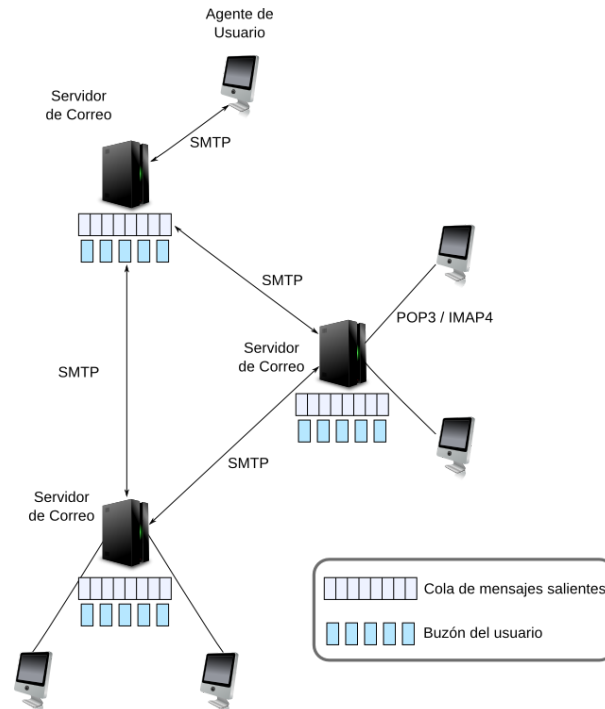
Festival de les Arts

CABO DE PLATA

WUOLAH

Servicio de correo electrónico

El correo electrónico es un servicio de aplicación TCP/IP basado en tres componentes fundamentales:

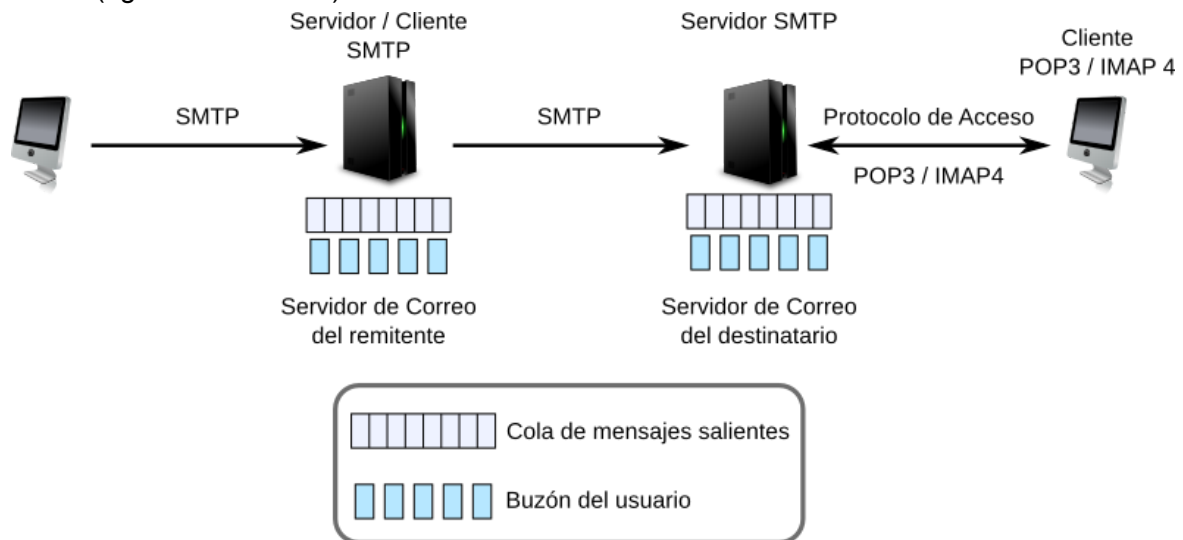


- **Agente de usuario** (por ejemplo, Microsoft Office Outlook). El agente de usuario del correo se corresponde con el propio sistema de correo en la máquina del usuario. Dispone, a su vez, de los siguientes elementos:
 - Un proceso cliente de correo SMTP
 - Un editor de texto
 - Un codificador/decodificador MIME
 - Un protocolo de acceso al correo POP3/IMAP4 para recuperar el correo desde el buzón del destinatario en su servidor de correo a un directorio de su disco duro.
- **Servidor de correo**: Se corresponde con el sistema de correo en el servidor de correo de la organización del usuario o en la red IP de su ISP. Dispone de los siguientes elementos:
 - Un proceso servidor de correo SMTP
 - Buzones de los usuarios
 - *Buffer* de mensajes salientes

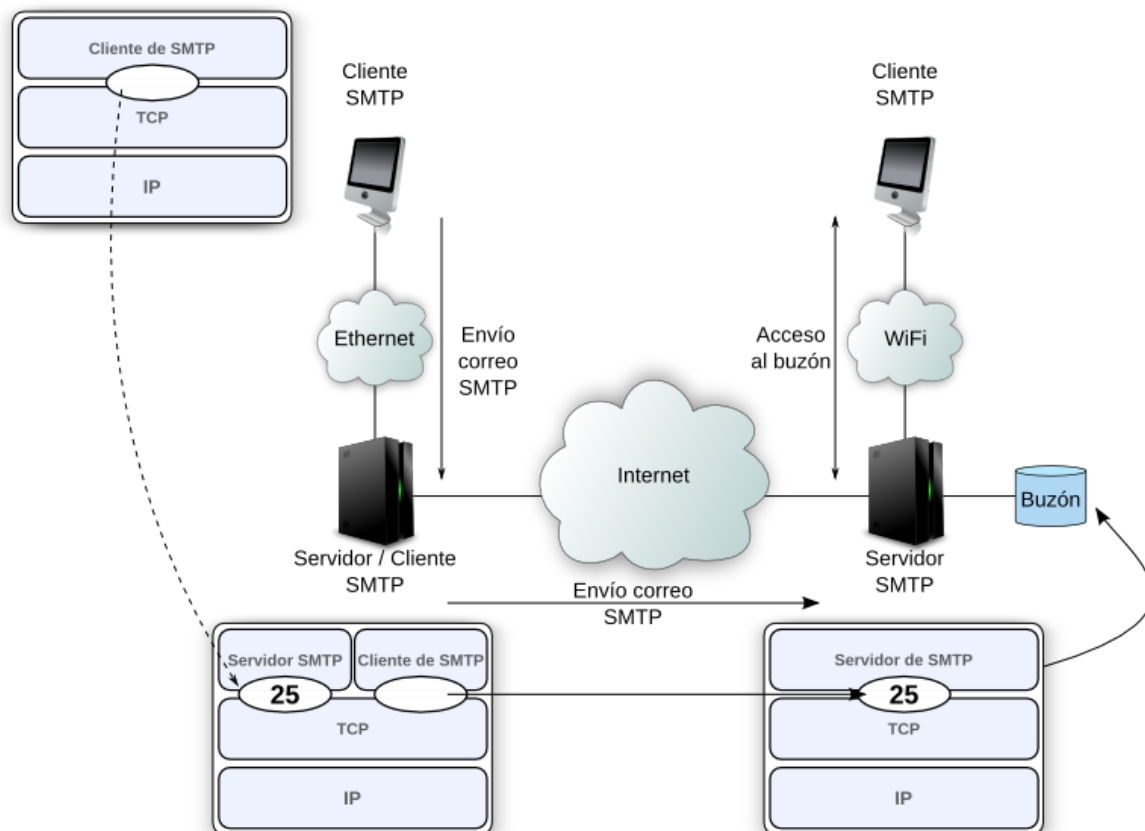
- **Protocolo de Envío de Correo SMTP** (*Simple Mail Transfer Protocol*): Permite enviar correo desde el cliente de correo SMTP de un agente de usuario a su servidor de correo SMTP por omisión. También permite enviar correo desde el servidor origen (del remitente) al servidor destino (buzón del destinatario).

Servicio de envío de correo: Protocolo SMTP

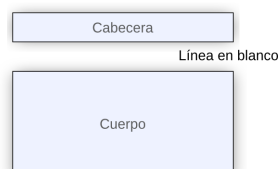
El protocolo **SMTP** (Simple Mail Transfer Protocol) proporciona un servicio de envío de correo electrónico por Internet. Este protocolo permite a un usuario enviar correo a otros usuarios aunque éstos no estén activos durante la comunicación, es decir, no estén conectados simultáneamente. Esto es así, ya que se adopta la idea del buzón electrónico, es decir, un directorio o carpeta particular donde los mensajes se depositan hasta que el receptor del mismo, se conecta y los recupera. Por regla general, cada usuario dispone de su propio buzón en la máquina servidora de correo de su organización (o de su ISP) donde se está ejecutando un proceso servidor SMTP. Asimismo, el usuario en su computadora dispone del correspondiente cliente SMTP que se ejecuta en el entorno de su aplicación de correo (agente de usuario).



Cuando un usuario (usuario1) compone un mensaje y decide transmitirlo al buzón de otro usuario (usuario2), el proceso cliente SMTP del primero, identificado con un número de puerto TCP (un número libre), transfiere el mensaje a su proceso servidor SMTP, identificado siempre en cualquier máquina servidora con el número de puerto 25 asociado al protocolo TCP. El proceso servidor SMTP se ejecuta en una máquina servidora de la organización del usuario emisor (o de su ISP) en la cual está su buzón. Dicho proceso servidor SMTP procede a analizar la dirección de correo del destinatario (usuario2@identificador_máquina). Por el identificador_máquina, en donde se ejecuta el proceso servidor SMTP, se da cuenta que el mensaje se debe transferir al buzón de otra máquina remota. El proceso servidor de SMTP del usuario1 se convierte en un proceso cliente de SMTP del proceso servidor de SMTP del usuario2. Seguidamente, el proceso servidor de SMTP del usuario 2 transfiere el mensaje al correspondiente buzón (usuario2).



El formato de texto de los mensajes de correo se basan en el estándar RFC-822 que define algunos campos de encabezado (To:, Cc:, Bcc:, From:, etc.), una línea en blanco y el cuerpo del mensaje. Sólo puede enviar mensajes en formato ASCII de 7 bits y no se puede utilizar para enviar ficheros que no contienen texto como son los ficheros binarios, de sonido o vídeo.



La extensión de correo electrónico multipropósito MIME (Multipurpose Internet Mail Extensions) permite enviar datos en formato no ASCII a través del correo electrónico. MIME transforma los datos en formato no ASCII en el lado emisor a datos ASCII y lo entrega al cliente de correo electrónico para que sean enviados a través de Internet. El mensaje en el lado receptor se transforma de nuevo a los datos originales.

La idea básica de MIME es continuar usando el formato RFC-822, pero agregando una estructura al cuerpo del mensaje y definir reglas de codificación para los mensajes no ASCII. Todo lo que hay que hacer es añadir software MIME en los programas emisores y receptores de correo.

5x1=5, 5x2=10, 5x3=15
la tabla del cinco te la
sabes gracias a la
musiquita porque...



Si quieres ganar entradas dobles
para los mejores festivales

CONTROL
Feel make Feel

Síguenos

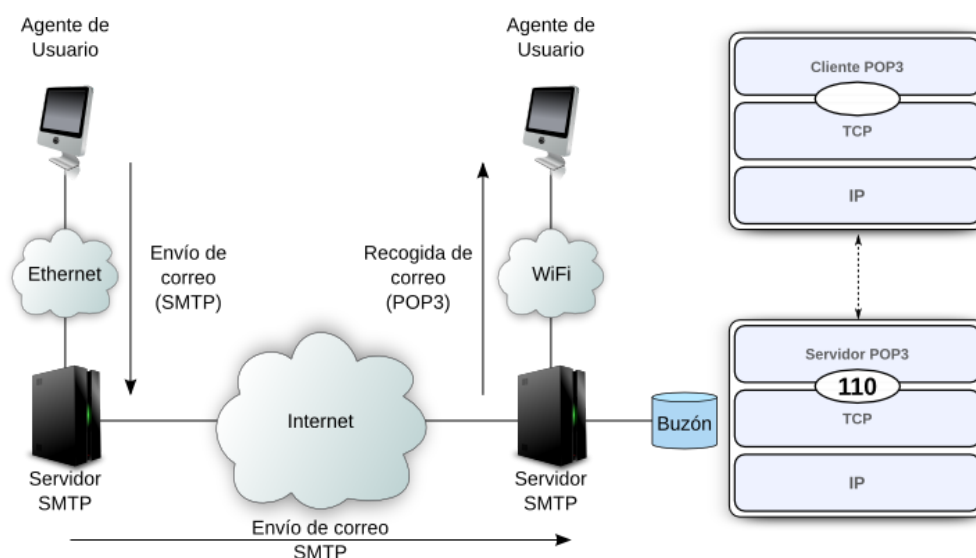
MIME define cinco nuevos encabezados de mensaje:

Encabezado	Significado
MIME - Version	Identifica la versión MIME
Content - Description	Cadena de texto que describe el contenido
Content - Id	Identificador único
Content - Transfer - Encoding	Cómo se codifica el mensaje para su transmisión
Content - Type	Naturaleza del mensaje

Servicio de recogida del correo: Protocolo POP3

El protocolo **POP3** (Post Office Versión 3) proporciona un servicio de **recogida de mensajes del buzón de correo**. El buzón de un usuario se encuentra en la máquina servidora de correo y no en su computadora. Por tanto, se necesita de un protocolo como POP3 que recoja el correo del buzón y lo traiga a un directorio de su disco duro.

El procedimiento consiste en que un proceso cliente POP3 identificado con un número de puerto TCP (un número libre) se comunica (para recoger y traer el correo) con el proceso servidor POP3 identificado siempre, en cualquier máquina servidora, con el número de puerto TCP 110. El proceso servidor POP3 se ejecuta en la máquina servidora de correo donde se ejecuta el proceso servidor SMTP. La aplicación de correo, que se ejecuta en la máquina del usuario, dispone de un proceso cliente SMTP y un proceso cliente POP3.



CONTROL
Feel make Feel

Bilbao BBK Live
Bilbao

Festival de les Arts

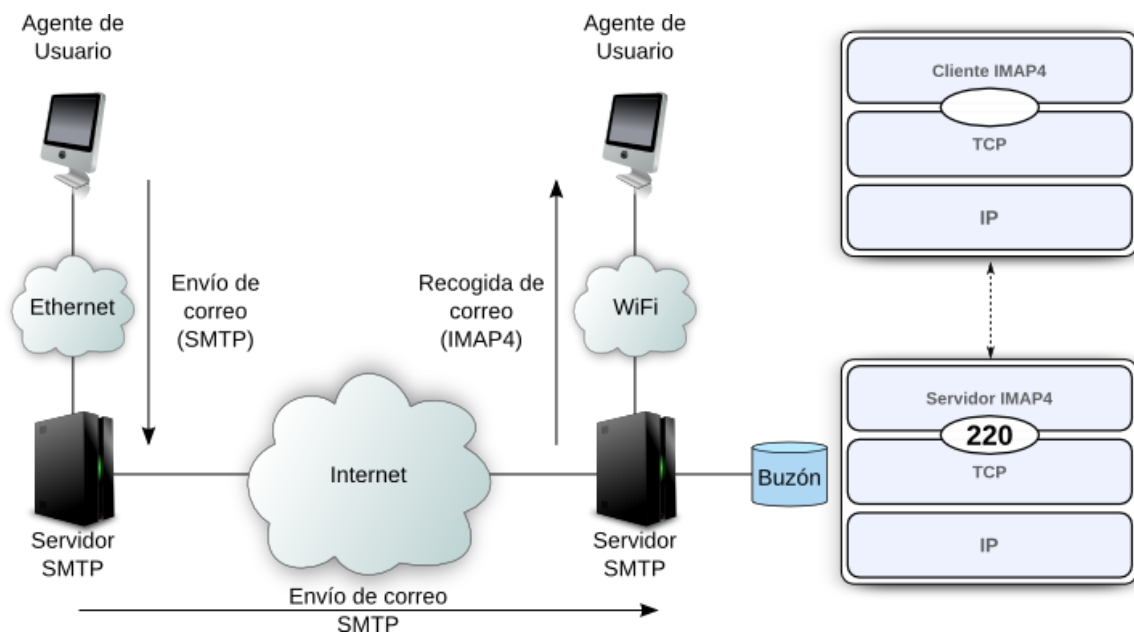
CABO DE PLATA

WUOLAH

Servicio de gestión del correo: Protocolo IMAPv4

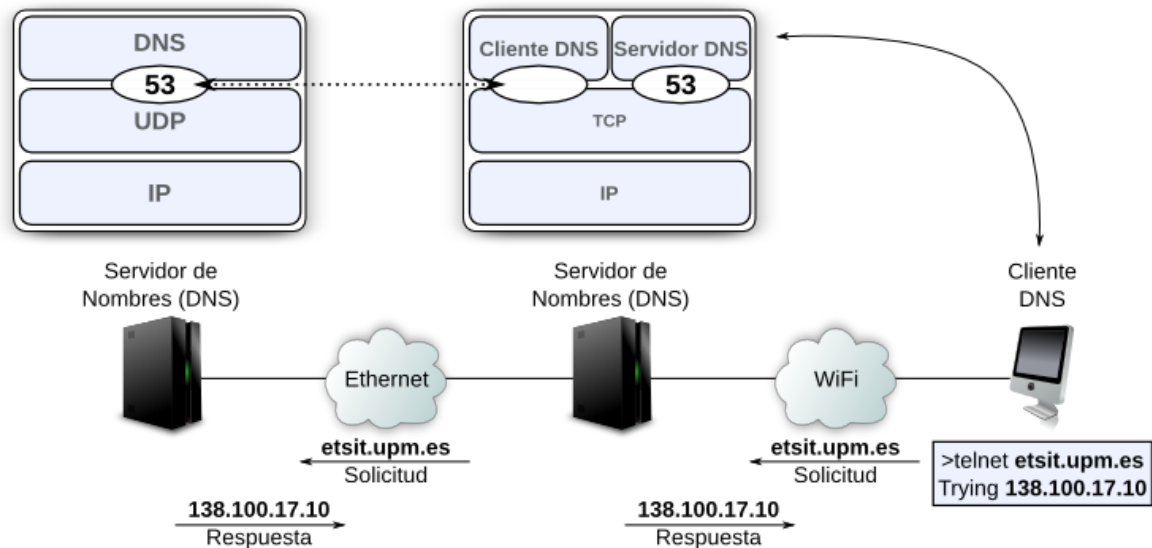
El protocolo **IMAP4** (Internet Message Access Protocol Rev 4) proporciona un servicio de **gestión de mensajes en el mismo buzón de correo**. Este protocolo permite al usuario **gestionar todo su correo en su propio buzón sin necesidad de recoger todos los mensajes y traerlos al disco duro de su máquina como se hacía por omisión con POP3**. El protocolo IMAP4 permite eliminar, clasificar y distribuir su correo en distintas carpetas dentro de la propia máquina servidora de correo. Asimismo, permite copiar o mover mensajes previamente seleccionados desde el buzón hasta su computadora. Es importante resaltar que el correo no se lee a través de la red sino que se trae vía IMAP4 de forma temporal a un directorio local en la máquina del usuario.

El procedimiento de trabajo de este protocolo consiste en que un proceso cliente IMAP4 identificado con un número de puerto TCP (un número libre) se comunica (para la gestión del correo) con el proceso servidor IMAP4 identificado siempre, en cualquier máquina servidora, con el número de puerto TCP 220. El proceso servidor IMAP4 se ejecuta en la máquina servidora de correo en donde se ejecuta el proceso servidor SMTP. La aplicación de correo, que se ejecuta en la máquina del usuario, dispone de un proceso cliente SMTP y un proceso cliente IMAP4.



Servicio de resolución de direcciones: Protocolo DNS

El protocolo DNS (Domain Name System) ofrece un servicio de traducción de una dirección simbólica en una dirección IP de máquina.



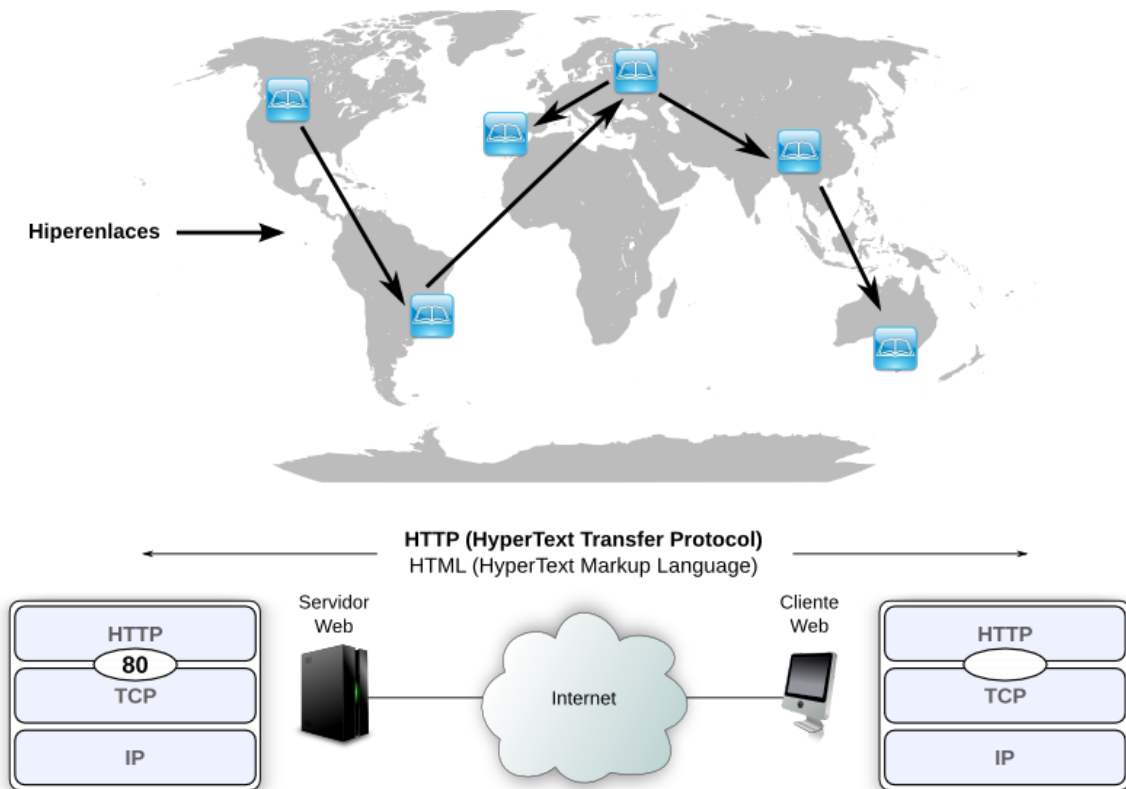
El procedimiento consiste en que un proceso cliente DNS identificado con un número de puerto UDP (un número) se comunica con el proceso servidor DNS identificado, siempre en cualquier máquina servidora, con el número de puerto UDP 53. El proceso cliente DNS se ejecuta en la máquina del usuario y el proceso servidor DNS en la máquina servidora de la organización (o del proveedor de servicios). Si el proceso servidor DNS, con el que interactúa directamente el proceso cliente DNS, no sabe traducir la dirección simbólica en su formato numérico; entonces, se transforma en un proceso cliente DNS de otro proceso servidor DNS de superior jerarquía en la estructura por niveles del servicio DNS existente por Internet. Y así se procederá hasta encontrar (o no, dando por concluida la fase de búsqueda) la asociación simbólica-numérica invocada. Una vez encontrado el formato numérico solicitado, éste se transmite por el mismo camino de invocaciones.

En una organización se puede encontrar dos tipos de servidores DNS:

- **Servidores DNS primarios** (*Primary Name Servers*): Almacenan la información de dominios en una base de datos local. Son responsables de mantener la información de los dominios actualizada, por lo que cualquier cambio en los datos o cualquier alta o baja de dominio debe ser comunicada a estos servidores.
- **Servidores DNS secundarios** (*Secondary Name Servers*): Se encuentran por debajo de los anteriores en la jerarquía, por lo que deben obtener de ellos los datos correspondientes a su zona de acción mediante un proceso de copia denominado "transferencia de zona". Además, estos servidores actúan como sistemas de seguridad, al mantener la información de forma redundante, con lo que si un servidor DNS tiene problemas, la información se puede recuperar desde otro. Además, evitan la sobrecarga del servidor principal, distribuyendo el trabajo entre distintos servidores situados estratégicamente con lo que se gana velocidad en las resoluciones.

Servicio World Wide Web (WWW)

El servicio **World Wide Web (WWW)** o simplemente Web es un inmenso conjunto de información diseminado por todo el mundo y enlazado entre sí. En concreto, WWW es una colección mundial de ficheros de texto y multimedia en código **HTML** (HyperText Markup Language) y ligados mutuamente a través de un sistema de documentos de hipertexto e hipermedia vía el protocolo HTTP (HyperText Transfer Protocol). Con el hipertexto e hipermedia, una palabra, frase, imagen, etc., pueden contener un hiperenlace con otro texto, audio, imagen y vídeo.



Desde el punto de vista tecnológico, el servicio Web es un servicio distribuido del tipo cliente-servidor en el que un cliente o navegador Web puede acceder a una página Web de inicio ofrecida como servicio desde un servidor Web. Sin embargo el servicio está distribuido sobre muchas localizaciones denominados sitios.

Cada sitio almacena uno o más documentos, denominados páginas Web. Cada página Web puede contener un enlace a otras páginas del mismo sitio o de otros sitios. Las páginas se pueden recuperar y visualizar utilizando navegadores Web. Un navegador consta fundamentalmente de un protocolo cliente que, generalmente, es HTTP y de un intérprete que puede ser HTML, Java o JavaScript, dependiendo del tipo de documento Web que se desea visualizar o interpretar.

5x1=5, 5x2=10, 5x3=15
la tabla del cinco te la
sabes gracias a la
musiquita porque...



Si quieres ganar entradas dobles
para los mejores festivales

CONTROL
Feel make Feel

Síguenos

CONTROL
Feel make Feel

Localizador de recursos URL

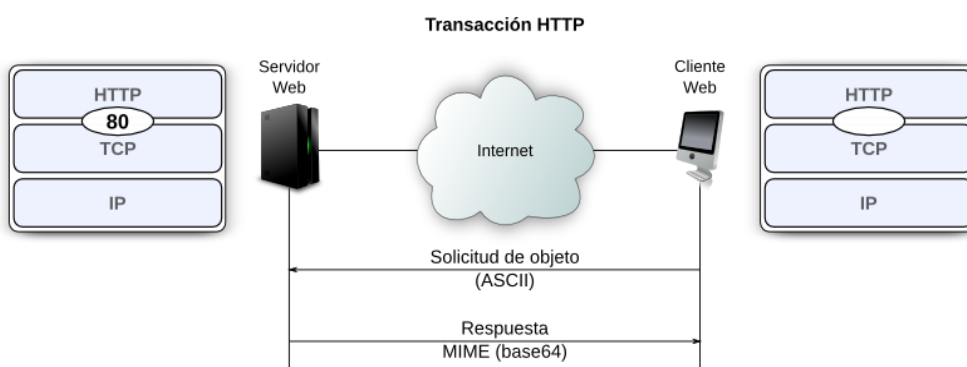
Un cliente que quiera acceder a una página Web necesita una dirección. Para facilitar el acceso a una página Web, el protocolo HTTP utiliza un localizador o **URL** (Uniform Resource Locator) que identifica de forma única a la página en Internet. El URL define cuatro aspectos:

protocolo://máquina:puerto/ruta

- **Protocolo:** lenguaje utilizado entre el cliente y el servidor para recuperar el documento. Se pueden utilizar diferentes protocolos para recuperar un documento. El más común es HTTP por ser el lenguaje nativo del servicio Web.
- **Máquina:** computadora en la que se localiza la información. Las páginas Web se almacenan, generalmente, en computadoras y las computadoras tienen un alias que, normalmente, comienzan con "www". Esto no es obligatorio, ya que la estación puede ser cualquier nombre dado a la computadora que almacena la página Web.
- **Puerto:** número entero que identifica al proceso servidor. **El URL puede contenerlo opcionalmente.**
- **Ruta:** nombre del fichero donde se encuentra la información. Puede contener separadores (/) para separar los directorios de los subdirectorios y ficheros.

Servicio de transacciones Web: Protocolo HTTP

El protocolo HTTP (HyperText Transfer Protocol) está basado en un modelo cliente-servidor orientado a transacciones. La versión actual de HTTP es la **1.1**. HTTP dispone de una **variante cifrada mediante SSL llamada HTTPS**. Para proporcionar fiabilidad, HTTP hace uso de TCP. Un cliente HTTP (navegador) abre una conexión y envía su solicitud al servidor Web, el cual responderá con el recurso solicitado si está disponible y si permite su acceso.



HTTP funciona como una combinación de los protocolos FTP y SMTP. Es similar a FTP debido a que transfiere ficheros y utiliza los servicios de FTP. Sin embargo, es mucho más sencillo que FTP debido a que sólo utiliza una conexión TCP. No hay una conexión de control diferente; sólo se transfieren datos entre el cliente y el servidor. A su vez, HTTP es similar a SMTP debido a que los datos transferidos entre el cliente y el servidor se parecen a mensajes SMTP. Además, **el formato de los mensajes es controlado por cabeceras MIME**. Los mensajes SMTP se almacenan y reenvían, pero los mensajes HTTP se entregan

Bilbao BBK Live

Festival de les Arts

CABO DE PLATA

WUOLAH

inmediatamente. Las órdenes del cliente al servidor se introducen en un mensaje de solicitud. El contenido del fichero solicitado u otra información se incluyen en un mensaje diferente.

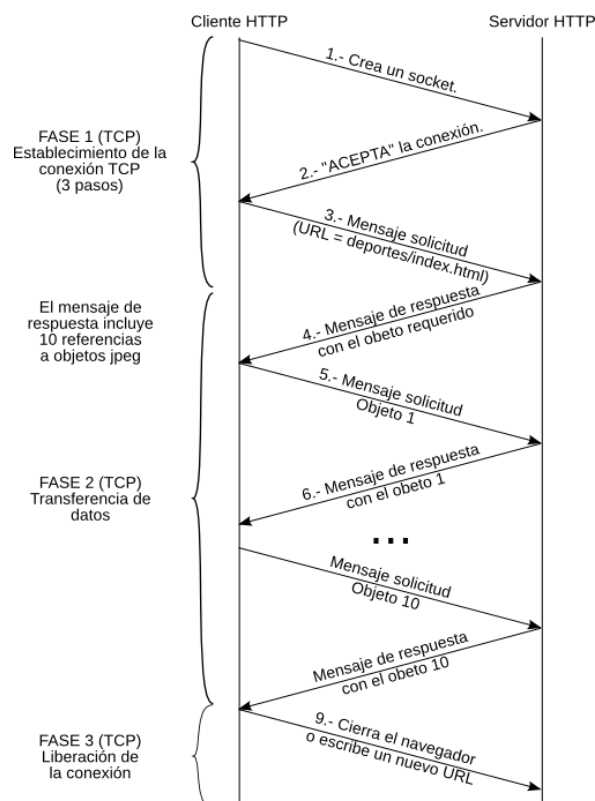
El procedimiento operativo consiste en que un proceso cliente HTTP identificado con un número de puerto TCP (un número libre) se comunica con el proceso servidor HTTP identificado siempre, en cualquier máquina servidora, con el número de **puerto TCP 80**. Un usuario que quiera disponer de su propio servidor Web (servidor Web privado, diferente del servidor Web del administrador), y no sólo de su página Web, arrancará dicho servidor en un número de **puerto diferente al 80**; generalmente, se suele poner el **8080**.

Características del protocolo HTTP 1.1

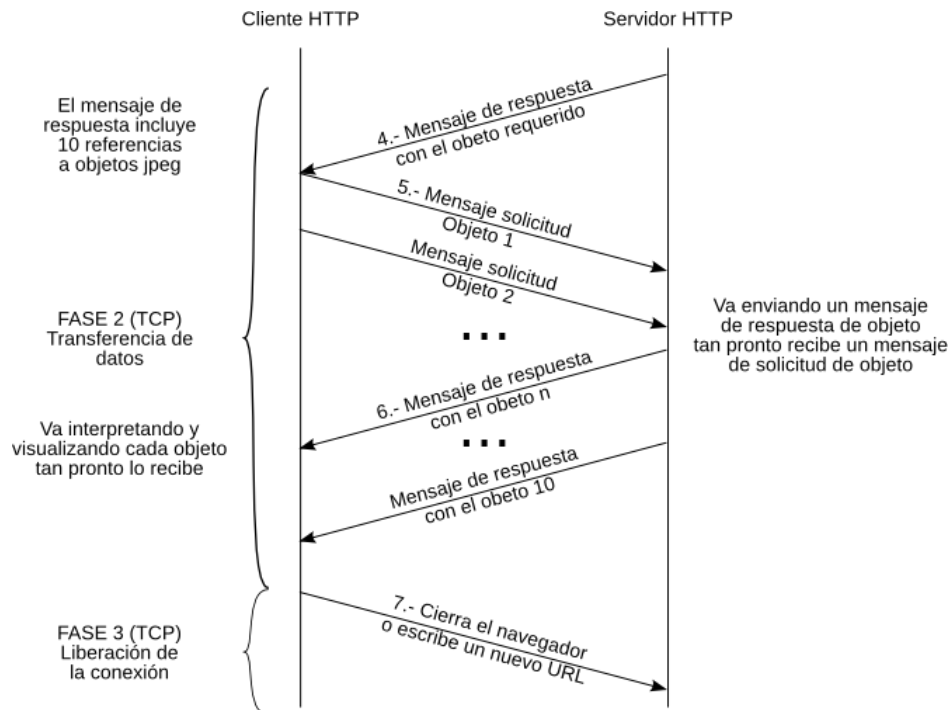
El protocolo HTTP 1.1 es un protocolo cliente-servidor orientado a transacciones. Para proporcionar fiabilidad, **HTTP hace uso de TCP**. Un cliente HTTP (navegador) abre una conexión y envía su solicitud al servidor Web, el cual responderá con el recurso solicitado si está disponible y se permite su acceso. Seguidamente, la conexión se cierra.

Las principales características de HTTP 1.1 son:

- **Persistente:** Soporta conexiones persistentes entre el cliente y el servidor. Se pueden enviar múltiples objetos por una única conexión TCP. Una típica página Web puede contener múltiples iconos, imágenes, etc., de tal manera que es necesario enviar múltiples solicitudes y, todo ello, en la misma conexión TCP.
- En HTTP 1.1 existen dos tipos de persistencia:
 - **Persistencia sin pipelining:** El cliente envía una nueva solicitud sólo cuando ha recibido el objeto de la anterior solicitud. Un RTT por cada objeto referenciado.



- **Persistencia con pipelining:** El cliente envía más de una solicitud tan pronto encuentra más de un objeto referenciado. Un solo RTT para todas las referencias. Por omisión en HTTP 1.1.



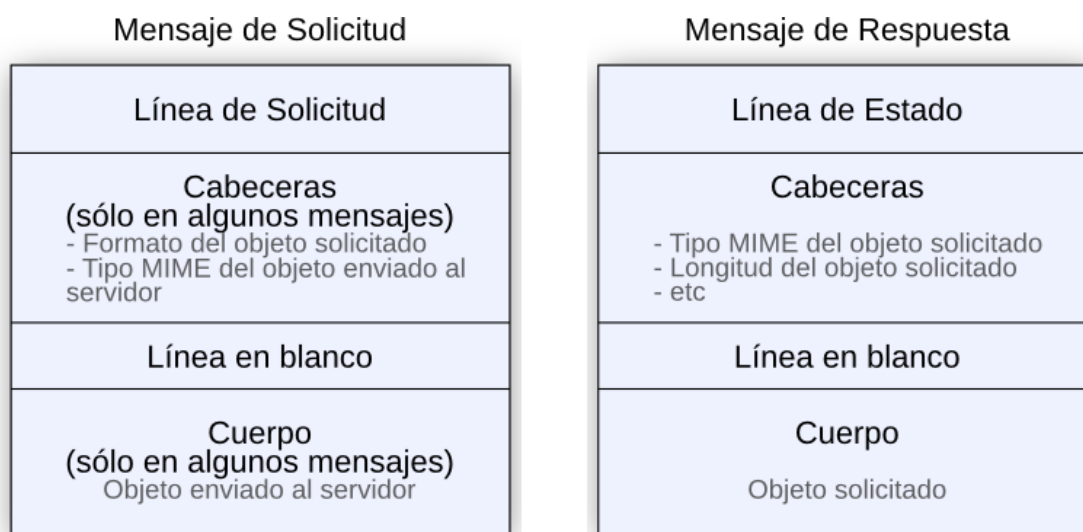
- **Sin estado:** El servidor HTTP no mantiene el estado o la información sobre las solicitudes (historial) de los clientes HTTP al cerrarse la conexión TCP. Para mantener el estado, el programador de la aplicación Web tendrá que gestionar el estado por encima de HTTP:

- **Reescritura de los URL:** Captura posterior de la información del usuario incrustada previamente por el servidor en cada URL.
- **Campos ocultos en la página HTML:** Captura posterior de la información del usuario incrustada por el servidor en campos no visibles HTML.
- **Cookies**("galletas"): Ficheros de texto o fragmentos de información (trozos de datos) diferentes que contienen las acciones del usuario para cada servidor Web visitado y que se almacenan en el disco duro del cliente, a través de su navegador, a petición del servidor Web. Esta información puede ser recuperada luego por el servidor en posteriores visitas para diferenciar usuarios y actuar de diferente forma dependiendo del usuario. Se utilizan para:
 - **Control de usuarios:** Cuando un usuario introduce su nombre de usuario y contraseña, se almacena una cookie para que no tenga que estar introduciéndolas por cada página del servidor
 - **Seguimientos de usuarios:** Estadísticas de usos, aficiones, cestas virtuales de compras, etc.

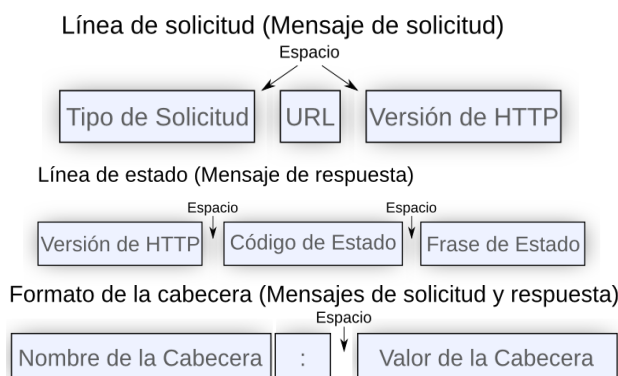
- **Personalización del sitio Web en función de los hábitos o preferencias del usuario:** Particularizar el aspecto en cuanto a presentación y funcionalidad.

Formato de un mensaje HTTP 1.1

El protocolo HTTP especifica el formato de los mensajes entre los clientes y los servidores Web. El formato de los mensajes HTTP 1.1 de solicitud y respuesta es similar en cuanto a formato. Un mensaje de solicitud consta de una línea de solicitud, una cabecera y en algunas ocasiones un cuerpo. A su vez, un mensaje de respuesta consta de una línea de estado, una cabecera y, en muchas ocasiones, un cuerpo con el objeto solicitado.



A su vez, el Tipo de Solicitud es el primer campo de la línea de solicitud de un mensaje de solicitud.



5x1=5, 5x2=10, 5x3=15
la tabla del cinco te la
sabes gracias a la
musiquita porque...



Si quieres ganar entradas dobles
para los mejores festivales

CONTROL
Feel make Feel

Síguenos

CONTROL
Feel make Feel

Existen diferentes tipos de solicitud en función de los métodos HTTP utilizados. Los tres métodos o solicitudes más relevantes son:

- **GET:** Método HTTP para solicitar un objeto, a veces, con parámetros que se encapsulan en el localizador URL e indican exactamente lo que el usuario necesita o busca. La mayoría de las solicitudes HTTP son mediante GET. Los parámetros pasados se codifican (pares de valores: nombre-valor) y son visibles en el URL.
- **POST:** Método HTTP para solicitar un objeto siempre con parámetros del usuario que no son visibles en el localizador URL. Típica solicitud HTTP cuando se hace clic en el botón de un formulario. Los parámetros se introducen en el cuerpo del mensaje (pares de valores: nombre-valor) y no son visibles en el localizador URL. El tamaño de los datos no está limitado como en GET.
- **PUT:** Método HTTP para enviar un objeto al servidor.

 **Bilbao
BBK Live**
Bilbao

 **Festival de les Arts**

CABO DE PLATA

WUOLAH