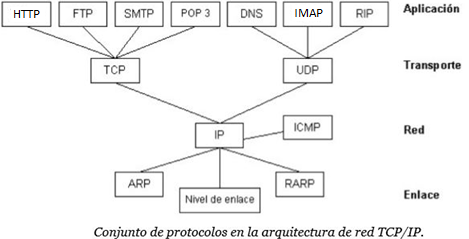
***Unidad 11. Protocolos de capa de Transporte.***

***Tema 1: Protocolo TCP, (Protocolo de Control de Transporte).***

***Tema 2: Protocolo UDP, (Protocolo de Usuario).***



***Tema 1: Protocolo TCP, (Protocolo de Control de Transporte).***

La función que cumple el protocolo TCP, es básicamente la de efectuar una transportación confiable y bidireccional de los datos. Fue creado en 1974 por Vint Cerf y Robert Kahn.

Muchos programas dentro de una red de datos compuesta de computadoras, pueden usar TCP para crear conexiones entre ellos a través de las cuales puede enviarse un flujo de datos. El protocolo garantiza que los datos serán entregados en su destino sin errores y en el mismo orden en que se transmitieron. También proporciona un mecanismo para distinguir distintas aplicaciones dentro de un mismo Host de Destino, a través del concepto de Puerto.

TCP da soporte a muchas de las aplicaciones más populares de Internet, por ejemplo navegadores, intercambio de archivos, etc., y también a los protocolos de aplicación como HTTP, DNS, FTP, SMTP, POP3, IMAP, etc.

***Funciones de TCP.***

En la familia de protocolos TCP/IP, TCP es la capa intermedia entre el protocolo de internet, (IP), y las aplicaciones. Habitualmente, las aplicaciones necesitan que la comunicación sea fiable y, dada que la capa IP aporta un servicio de datagramas no fiable, (sin confirmación), TCP añade las funciones necesarias para prestar un servicio que permita que la comunicación entre dos sistemas se efectúe libre de errores, sin pérdidas y con seguridad.

Los servicios provistos por TCP corren en los computadores, (hosts), de ambos extremos de una conexión, no en las redes. Por lo tanto, TCP es un protocolo para manejar conexiones de extremo a extremo. Ya que TCP establece una sesión de extremo a extremo, a estas conexiones se las llama ***circuitos virtuales.***

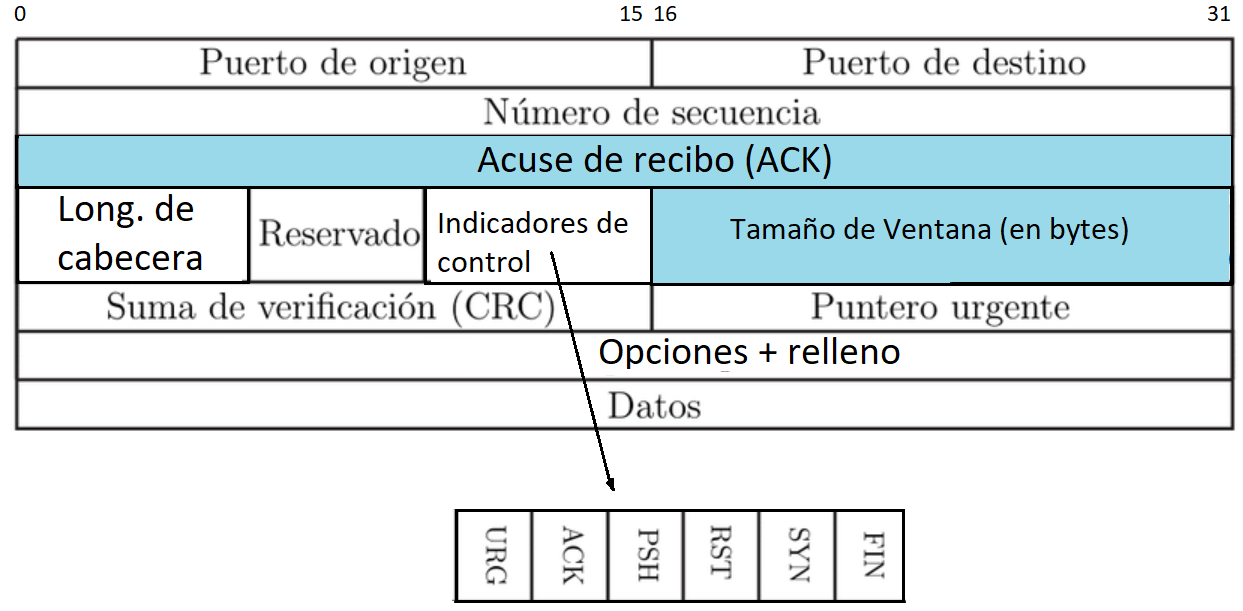
***Descripción de las funciones de TCP:***

1. ***Orientado a conexión:*** Dos computadores establecen una conexión para intercambiar datos. Los sistemas de los extremos se sincronizan para manejar el flujo de paquetes, y adaptarse a la congestión de la red.
2. ***Operación Full-Duplex:*** Una conexión TCP es un par de circuitos virtuales, cada uno en una dirección. Sólo los dos sistemas finales sincronizados pueden usar la conexión.
3. **Error Checking**: Es una técnica de Checksum usada para verificar que los paquetes no estén corruptos.
4. ***Acknowledgements:*** Al recibir uno o más paquetes, (segmentos), el receptor regresa un acknowledgement, (reconocimiento), al transmisor indicando que recibió los segmentos. Si los segmentos no son notificados, el transmisor puede reenviarlos o terminar la conexión si considera que el receptor no está ya en conexión.
5. ***Control de Flujo:*** Si el transmisor está desbordando el buffer del receptor por transmitir demasiado rápido, el receptor descarta paquetes. Los acknowledgement fallidos indican al transmisor para bajar la tasa de transferencia o dejar de transmitir.
6. ***Servicio de recuperación de Paquetes:*** El receptor puede pedir la retransmisión de un paquete. Si el paquete no es notificado como recibido, (ACK), el transmisor envía de nuevo el segmento.

Los servicios confiables de entrega de datos son críticos para aplicaciones tales como transferencias de archivos, (FTP, por ejemplo), servicios de bases de datos, proceso de transacciones y otras aplicaciones de misión crítica en las cuales la entrega de cada paquete debe estar garantizada.

Nota: el paquete de datos en la capa de transporte se llama ***segmento*** de datos.

***Formato de TCP:*** La cabecera se compone de palabras de 32 bits.



Los campos resaltados en celeste, son configurados por la estación receptora.

* ***Puerto Origen.*** Representa el nombre de la aplicación que envía los datos.
* ***Puerto Destino.*** Representa el nombre de la aplicación que recibe los datos.
* ***Número de Secuencia***. TCP enumera cada segmento, de esta forma el Host Destino puede confirmar que todos los segmentos llegaron y con el número de secuencia puede ordenarlos para reconstruir el archivo.
* ***Número de Acuse de Recibo.*** Es utilizado por el equipo receptor para reconocer que los segmentos han llegado con éxito. Este número será el número de secuencia del segmento siguiente que el Host Receptor está listo para recibir.
* ***Longitud.*** Expresa la longitud de la cabecera del segmento TCP.
* ***Reservados.*** No se utilizan y se establecen en “0”.
* ***Indicadores de Control.*** TCP utiliza seis indicadores de control. Cada indicador está encendido si se establece en “1”, y está apagado si se establece en “0”.

El orden de los indicadores de control es:

***U***rgente – ***A***cuse de recibo – ***P***ush – ***R***esetear – ***S***incronizar – ***F***inalizar

***Urgente:*** Toma el valor “1” indicando que el campo Puntero Urgente es válido.

***Acuse de recibo:*** Toma el valor “1” indicando que el Número de Acuse de Recibo es válido.

***Push:*** Toma el valor “1” indicando que el proceso TCP del Destino debe pasar los datos que tenga almacenados a la capa de aplicación, sin esperar a recibir más datos.

***Resetear:*** Toma el valor “1” indicando que termina la conexión sin esperar respuesta.

***Sincronizar:*** Toma el valor “1” para solicitar el establecimiento de una conexión.

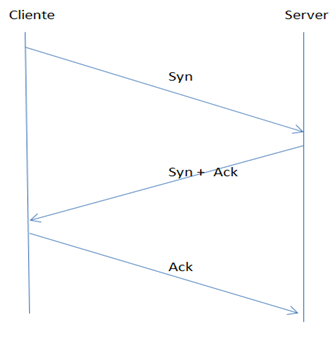
***Finalizar:*** Toma el valor “1” para indicar que se solicita la finalización de una conexión. El paquete que lo lleva activo indica que es el último segmento a enviar.

* ***Ventana o tamaño de ventana:*** Indica el tamaño del buffer de memoria que tiene disponible el Host Receptor para recibir segmentos TCP.
* ***Suma de Comprobación***: Es un valor obtenido de realizar una comprobación de redundancia cíclica, CRC, utilizando la cabecera y los datos del segmento TCP, para confirmar que no hay ninguna alteración en el mismo.
* ***Puntero Urgente.*** Si el indicador de control Urgente está activado en “1”, este campo es válido, e indica dónde acaban los datos urgentes.
* ***Opciones y Relleno.*** Raramente algunas opciones pueden agregarse en el campo Opciones. El campo Relleno sirve para completar la palabra de 32 bits de la cabecera de TCP.
* ***Datos:*** Este campo es de los datos reales que quieren transmitirse.

***Establecimiento de la conexión TCP.***

Para establecer la conexión TCP se usa un procedimiento llamado ***negociación en tres pasos.*** Para la desconexión se usa un procedimiento similar.

Durante el establecimiento de la conexión, se utilizan los siguientes indicadores de control: Sincronizar, (syn), y Reconocimiento, (ack).



El Host Origen, (Cliente), que quiere establecer una sesión, envía un segmento TCP al Host Destino con el Indicador Syn en “1”, encendido.

El Host Receptor, (Server), responde con un segmento TCP con el Indicador Syn y Ack en “1”, encendido.

Para completar el establecimiento de la conexión el Host Origen debe enviar un segmento TCP con el Indicador Ack en “1”, encendido.

El establecimiento de la conexión TCP queda así completado y puede comenzar el envío de los datos.

***Tamaño de ventana TCP.***

Aclaremos mejor el concepto del campo “Tamaño de Ventana”. El tamaño de la ventana de recepción TCP del Host Destino, (Server), es la cantidad de datos, (en bytes), que pueden ser contenidos dentro del buffer de memoria asignado por el Host Destino, para almacenar los datos de los segmentos TCP recibidos.

El Host Origen recibe el “Tamaño de Ventana” que le define el Host Destino, y así queda limitada la cantidad de datos, (bytes), que puede enviar hasta llenar el buffer de memoria asignado por el Host Destino o sea el Tamaño de Ventana del mismo.

Cuando el buffer se llena, el Host Destino envía el “Tamaño de Ventana” igual a cero, es decir no tiene más lugar para ningún dato más. Esto detiene el envío de datos por parte del Host Origen.

Luego el Host Destino entrega los datos que tiene en el buffer a la Aplicación que corresponda, y entonces libera nuevamente el Buffer de memoria, indicando al Host Origen el nuevo valor de su buffer o “Tamaño de Ventana”, prosiguiendo entonces la transferencia de datos.

***Escalado de ventana.***

El campo TCP de “Tamaño de Ventana” está limitado a 16 bits, es decir que el tamaño máximo de ventana es de 216 = 65.536 bytes.

Si queremos aumentar el tamaño de la ventana debe hacerse durante la negociación de tres pasos que constituye el comienzo de la conexión. En ella se utiliza una opción denominada escalado de ventana TCP, (TCP window scale), que permite incrementar el tamaño máximo de ventana desde 65.536 bytes a 1 Gigabyte.

***Ventana deslizante.***

La ventana deslizante es un proceso dirigido al control de flujo de datos que existe entre emisor y receptor, mediante el cual el receptor indica al emisor su disponibilidad para recibir datos.

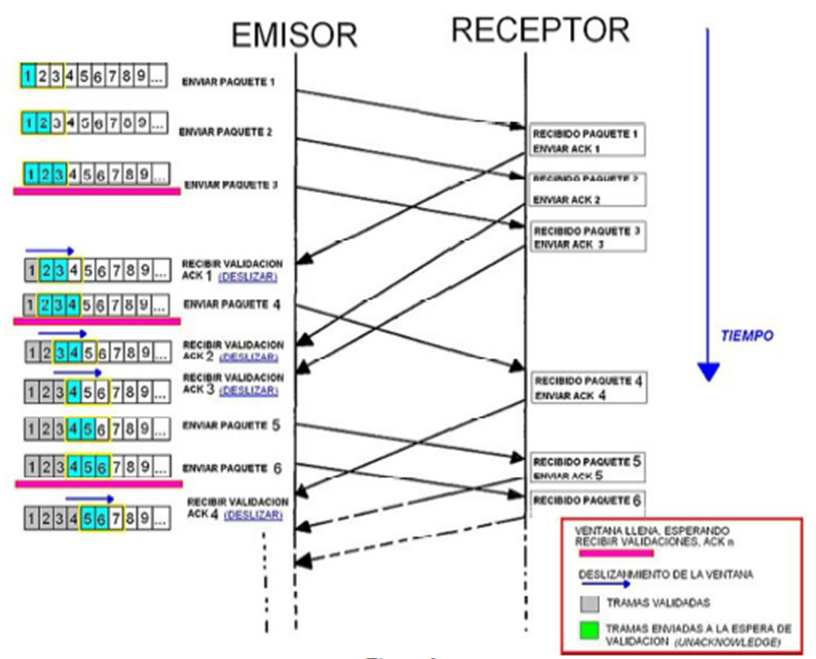
Este proceso es necesario para no inundar al receptor con segmentos TCP. El receptor al recibir datos debe procesarlos, y si no realiza esto a la misma velocidad que el emisor los envía, se verá saturado de datos, y parte de ellos se pueden perder.

Para evitar tal situación, la ventana deslizante controla el ritmo de envíos del emisor al receptor.

El proceso de ventana deslizante permite al emisor transmitir múltiples segmentos TCP, antes que el receptor le confirme la recepción de los segmentos TCP. Tal confirmación se llama validación, y consiste en el envío de mensajes denominados Acuse de Recibo, (Acknowledge), del receptor al emisor. Esta validación contiene el número del siguiente segmento TCP, que espera recibir el receptor.

Con esta indicación el emisor es capaz de distinguir el número de los envíos realizados con éxito.

Ejemplo de Ventana Deslizante de 3 segmentos TCP. Es decir el Emisor puede enviar hasta 3 segmentos TCP sin esperar validación alguna. Luego que envió estos tres segmentos TCP, debe esperar los Acknowledges correspondientes para deslizar la ventana y continuar con el envío de segmentos TCP. El Acknowledge se expresa en el campo de Acuse de Recibo Número. En él, el receptor responde al emisor, indicando que ha recibido correctamente el segmento TCP cuyo número envía en el campo “Acuse de Recibo Número”.



Otro ejemplo. Si el tamaño de ventana deslizante se fija en 8, el emisor puede transmitir hasta 8 segmentos sin recibir validación de ninguno de ellos, figura (a). Luego de esto esperará una confirmación de recepción procedente del receptor, sin enviar ningún paquete más.

Una vez que el emisor reciba una validación del primer segmento TCP que envió, perteneciente al número 1 de la ventana deslizante de 8, la ventana se desplazará abarcando 8 posiciones, (su ancho de ventana definido), pero en este caso desde la 2 a la 9, y enviará el segmento TCP número 9, continuando a la espera de recibir más confirmaciones, para poder seguir deslizándose y enviando lo segmentos TCP siguientes, figura (b).

Si el protocolo TCP esperase la validación de cada trama el proceso de transmisión sería lentísimo. El proceso de ventana deslizante hace que exista una continua transmisión, mejorando el desempeño de la misma.



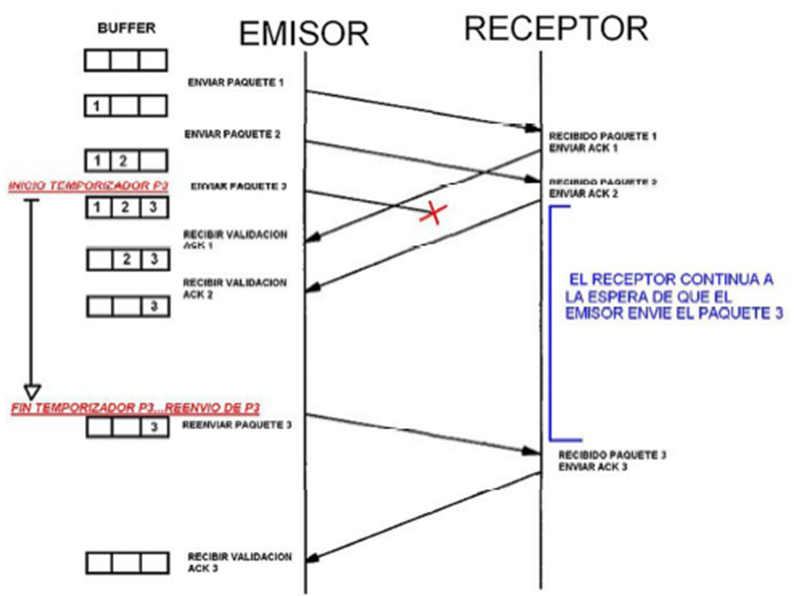
El emisor deberá guardar en un buffer de memoria, todas aquellas tramas enviadas y no validadas, (Unacknowledge packets), por si necesitase retransmitirlas. Sólo puede borrarlas del buffer al recibir su validación procedente del receptor, y deslizar así la ventana una unidad más.

El tamaño del buffer del emisor debe ser igual o mayor al tamaño de la ventana.

Así el buffer del emisor podrá almacenar temporalmente, todas las tramas enviadas sin validar.

A cada uno de los segmentos TCP pertenecientes al buffer del emisor, que fueron enviados y no validados, se les asigna un temporizador. El temporizador o contador es el límite de tiempo de espera, para recibir la validación de un determinado segmento TCP.

Si el segmento TCP se pierde en el camino, el emisor nunca recibirá la validación. De esta manera el temporizador expira, tomando el emisor la decisión de reenviar el segmento TCP asignado ese temporizador.



***Ventana de recepción del receptor.***

El receptor posee un buffer de memoria que corresponde al tamaño de la ventana fijado. En este buffer el receptor recibe un conjunto de segmentos TCP, que le llegan desordenados.

El buffer del receptor es la lista que tiene el receptor con los números de secuencia consecutivos de los segmentos TCP que puede aceptar.

El receptor dispone de un buffer de igual tamaño que su ventana de recepción, para almacenar temporalmente todos los segmentos TCP de una sesión TCP y ordenarlos.

El receptor es capaz de reconocer un segmento TCP erróneo, mediante el campo de Control de Redundancia Cíclica, CRC, y cuando detecta un segmento erróneo, porque se han producido errores en la transmisión, simplemente lo descarta. Y al no enviar el reconocimiento de su recepción, el emisor volverá a enviarlo, cuando se consuma el tiempo del temporizador asociado a él.

***Otra estrategia para la recuperación de errores.***

Estrategia de rechazo selectivo, (repetición selectiva, selective repeat).

El receptor descarta únicamente el segmento erróneo, y acepta los que llegan detrás almacenándose en el buffer de memoria de recepción. Al detectar el segmento TCP erróneo envía un código NACK n, (trama n errónea y no validada), al emisor para indicarle la situación, procediendo el emisor a reenviarle únicamente este segmento TCP erróneo.

Este segmento TCP se encontraría en el buffer de memoria del emisor, a la espera de ser validada. El receptor al recibir la retransmisión correcta del anterior segmento erróneo, lo almacena en el buffer con el resto de los segmentos recibidos y los ordena, para continuar con el proceso de entrega a la aplicación que corresponda.

***Tema 2: Protocolo UDP, (Protocolo de Usuario).***

El Protocolo UDP, protocolo de usuario, (User Datagram Protocol), es un protocolo de la capa de transporte de la familia de protocolos TCP/IP, y permite el envío de paquetes a través de la red, sin que se haya establecido previamente una conexión.

Tampoco tiene confirmación, ni control de flujo, por lo que los paquetes pueden adelantarse unos a otros, perderse en el camino, etc., ya que no hay confirmación de entrega o recepción.

Su uso principal es para protocolos como DHCP, BOOTP, DNS, y demás protocolos en los que el intercambio de paquetes puede volver a realizarse si no se completó el proceso, y esto no acarrea ningún inconveniente.

También son muy utilizados en la transmisión de audio y video en tiempo real, donde no es posible realizar retransmisiones, por los estrictos requisitos de retardo que se tiene en estos casos, resultando más importante la recepción de la transmisión, que garantizar el hecho de que lleguen absolutamente todos los bytes. Esto es porque no hay tiempo para enviar de nuevo los paquetes perdidos, cuando se está escuchando a alguien, o viendo un video en tiempo real.

Cuando TCP y UDP circulan por la misma red, TCP pasa a segundo lugar, para dejar a los datos en tiempo real, usar la mayor parte del ancho de banda.

El problema es que ambos son importantes para la mayoría de las aplicaciones, por lo cual, encontrar el equilibrio entre ambos es crucial.

***Descripción técnica.***

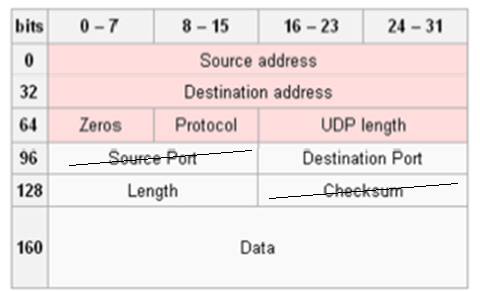
En la familia de protocolos de Internet, el UDP proporciona una sencilla interfaz entre la capa de red y la capa de aplicación. UDP no otorga garantías para la entrega de sus mensajes, y sólo tiene una suma de verificación en la cabecera y la carga útil.

La cabecera de UDP, está compuesta por una pseudo-cabecera IP, con los siguientes campos:

* IP origen.
* IP destino.
* Campo de ceros.
* Protocolo UDP.
* Longitud del paquete UDP.

Luego sigue la cabecera propiamente dicha de UDP, que consta de los siguientes campos:

* Puerto Origen. Que es opcional. Si no se utiliza debe ser puesto a cero.
* Puerto Destino. Es importante porque define a qué aplicación entregar los datos.
* Longitud. Que indica el tamaño en bytes del paquete UDP, incluidos los datos.
* Chequeo. Que es opcional en IPv4, aunque se utiliza normalmente. En IPv6 es obligatorio.



Seudo-cabecera

Cabecera UDP

La pseudo-cabecera es completada, y de los cuatro campos de la cabecera de UDP son utilizados dos en IPv4, Puerto Destino y Longitud incluidos los datos.

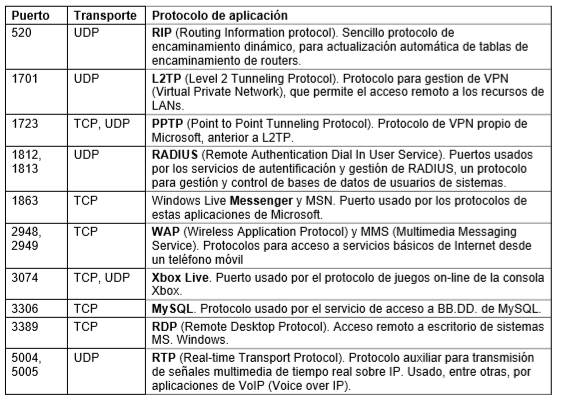
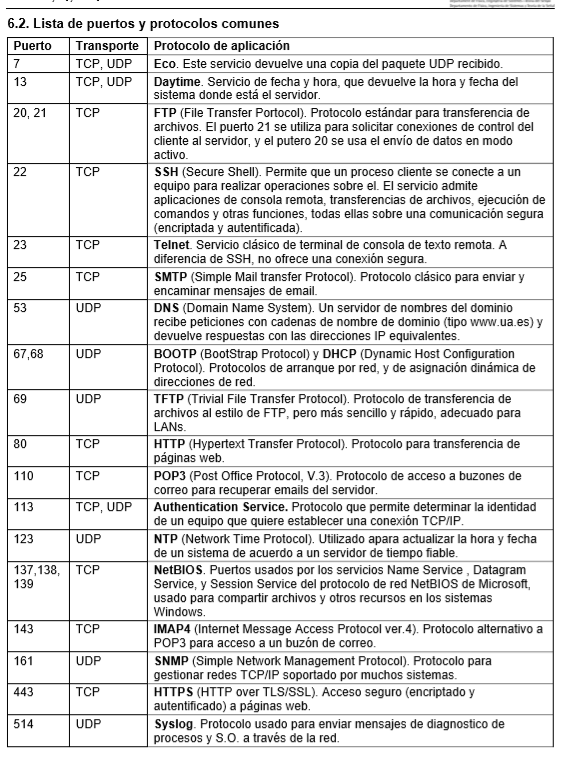
En IPv6, además de estos dos campos, es obligatorio el campo de Checksum.

***Puertos.***

Los puertos 1 a 1023 se llaman puertos “bien conocidos”, y están asignados a aplicaciones reconocidas.

Los puertos 1024 a 49.151 son puertos registrados también.

Los puertos 49.152 a 65.535 son puertos efímeros, y son utilizados como puertos temporales, sobre todo por los clientes al comunicarse con los servidores.



Fin de Unidad 11.