Estudios de Informática, Multimedia y Telecomunicación

Redes y Aplicaciones Internet
Reto 2: PEC 1 – Primera Prueba de Evaluación Continua

- Hay que entregar la solución en un fichero preferiblemente pdf, utilizando la plantilla adecuada, en el registro de evaluación continua.
- La fecha límite de entrega es el 20 de marzo del 2022

Respuestas

1. Haz un esquema gráfico del acceso a Internet que tienes en vuestra casa o en la empresa, identificando los elementos principales (*hosts*, *routers*, *switches*...). ¿Qué tipo de acceso es? Describe sus características diferenciales.

   Respuesta libre, donde se tienen que identificar los elementos del extremo de la red (equipos (*hosts*), enrutadores (*routers*), conmutadores (*switches*), ...) y el tipo de acceso (DSL, cable, FTTH, …).

2. Describe las tres tecnologías inalámbricas distintas que utilizas durante el día y sus características. Si puedes elegir entre varias tecnologías, ¿por qué prefieres una sobre otra?

   Hay tres tecnologías inalámbricas en Internet muy populares hoy en día:

   a) Wifi (802.11) En una LAN inalámbrica, los usuarios inalámbrico transmiten/reciben paquetes a/desde una estación base (es decir, un punto de acceso inalámbrico) en un radio de unas decenas de metros.
   La estación base suele estar conectada a Internet por cable y, por lo tanto, sirve para conectar los usuarios inalámbricos con la red por cable.

   b) Redes de acceso inalámbrico de área amplia 3G y 4G. En estos sistemas, los paquetes se transmiten a través de la misma infraestructura inalámbrica utilizada para la telefonía celular, por lo cual la estación base es gestionada por un proveedor de telecomunicaciones.
   Esto proporciona acceso inalámbrico a los usuarios en un radio de decenas de kilómetros de la estación base.

   c) Bluetooth. El Bluetooth es una especificación industrial por las Redes de Ámbito Personal (PAN, *Personal Area Network*) inalámbrico, que básicamente quiere decir que sirve para conectar los dispositivos que se traen encima o están a una distancia próxima.
   Con el Bluetooth, se puede obtener una forma de conectar e intercambiar información entre dispositivos como ordenadores de bolsillo, teléfonos móviles, ordenadores portátiles, ordenadores, impresoras y cámaras digitales a través de una forma segura, de bajo coste a través de ondas de radio.

3. ¿Por qué dos Proveedores de Internet (*Internet Service Providers*, ISP) del mismo nivel jerárquico a menudo se conectan directamente entre sí? ¿Cómo gana dinero un Punto de Intercambio de Internet (*Internet Exchange Point*, IXP)?

Si los dos ISP no hacen *peering* entre sí, cuando se envían tráfico entre ellos lo tienen que enviar a través de un ISP proveedor (intermediario), al cual tienen que pagar para transportar el tráfico. Al establecer un *peering* entre ellos directamente, los dos ISP pueden reducir los pagos a sus ISP proveedores. Un Punto de Intercambio de Internet (IXP) (normalmente en un edificio independiente con sus propios conmutadores) es un punto de encuentro donde varios ISP pueden conectarse y hacer *peering*.

Un ISP gana el dinero cobrando a cada uno de los ISP que se conectan al IXP una cuota relativamente pequeña, que puede depender de la cantidad de tráfico enviado o recibido desde el IXP.

4. La tarea de enviar información de un equipo a otro a través de diferentes redes se divide en 5 funcionalidades diferentes y complementarias, que se implementan como una serie de capas en la torre de protocolos (*protocol stack*) TCP/IP.
   a) Explica brevemente el mecanismo de Encapsulación (*Encapsulation*), tanto en emisión como en recepción.
   b) Explica brevemente (1-5 líneas para cada una) la funcionalidad de las capas Física, Enlace (de datos), Red, Transporte y Aplicación del modelo TCP/IP.
   c) Para cada capa, lista también el nombre que se da a sus paquetes de datos.
   d) Para cada capa, lista algún(os) protocolo(s) significativo(s).

   a)
   Toda capa entrega información a la capa inferior en emisión. Esta información incluye una Cabecera (*Header*), al inicio, con información que esta capa quiere intercambiar con la misma capa del siguiente equipo remoto que la implementa. Después de la cabecera van los Datos (*Payload*), que es la información que proviene de la capa superior para ser entregada a la capa superior del equipo remoto. Hay que tener en cuenta que los datos, a la vez, suelen incluir una cabecera de la capa superior, y datos de 2 capas por encima.
   En recepción, cada capa recibe la información, analiza la Cabecera, y si todo es correcto entrega los Datos a la capa superior.

   b, c, d)
   Capa física
   - El trabajo de la capa física es mover los bits individuales dentro de la trama desde un nodo al siguiente. Los protocolos de esta capa también dependen del enlace y del medio de transmisión (por ejemplo, cable de cobre de par trenzado o fibra óptica monomodo).
   - En este caso son los bits los datos que se desplazan por el enlace.
   - Por ejemplo, Ethernet tiene muchos protocolos de capa física: uno para el par trenzado de cobre, otro para el cable coaxial, otro para la fibra, etc.

   Capa de enlace (de datos)
   - Para trasladar un paquete desde un nodo (host o *router*) hasta el nodo siguiente de la ruta, dentro de la misma red, se recorre a los servicios de la capa de enlace.
   - Nos referiremos a los paquetes de la capa de enlace como tramas.
   - Algunos ejemplos de protocolos de capa de enlace son Ethernet y WiFi.

   Capa de red
   - La capa de red de Internet es la responsable de pasar de un host a otro, pasando de una red a otra mediante una serie de enrutadores (*routers*) entre el origen y el destino.
   - Los paquetes de la capa de red se conocen como datagramas.
   - La capa de red incluye el protocolo IP (y también el protocolo ICMP).

   Capa de transporte

- La capa de transporte de Internet transporta los mensajes de la capa de aplicación entre los puntos finales/extremos de la aplicación.
- Nos referiremos a un paquete de la capa de transporte como segmento (TCP) o datagrama de usuario (UDP).
- En Internet hay dos protocolos de transporte, TCP (fiable y orientado a conexión) y UDP (no fiable), cualquiera de los cuales puede transportar mensajes de la capa de aplicación.

Capa de aplicación
- Es donde residen las aplicaciones de red y los protocolos de la capa de aplicación.
- Nos referiremos al paquete de información de la capa de aplicación como un mensaje.
- Incluye muchos protocolos, como el protocolo HTTP (que permite la solicitud y transferencia de documentos web), SMTP (que permite la transferencia de mensajes de correo electrónico) o FTP (que permite la transferencia de ficheros entre dos sistemas finales).

5. Responde brevemente a las siguientes preguntas:

a) ¿Por qué hacen falta mecanismos de conmutación (*switching*)?
b) ¿Qué diferencia hay entre una red de conmutación de circuitos y una red de conmutación de paquetes?
c) ¿Qué ventaja tiene una red de conmutación de circuitos sobre una red de conmutación de paquetes?
d) ¿Por qué hacen falta mecanismos de multiplexado en un enlace?
e) ¿Qué diferencia hay entre el Multiplexado por división en el tiempo (TDM, *Time-division multiplexing*) y el Multiplexado por división en la frecuencia (FDM, *Frequency-division multiplexing*)?
f) ¿Qué ventajas tiene el TDM sobre el FDM en una red?

a) Los mecanismos de conmutación (*switching*) hacen falta porque la red no es totalmente conectada (los equipos no están conectados todos con todos), sino que están conectados a través de un número limitado de enlaces conectados entre ellos mediante conmutadores. Por lo tanto, hace falta un mecanismo que establezca las conexiones individuales entre las vías de transmisión o los circuitos de comunicación entrantes y salientes para establecer la comunicación deseada entre dos puntos.

b) Una red de conmutación de circuitos puede garantizar una cierta cantidad de ancho de banda de extremo a extremo durante la duración de una llamada. La mayoría de redes de conmutación de paquetes actuales (incluida Internet) no pueden garantizar el ancho de banda de extremo a extremo.

c) En las redes de conmutación de circuitos, los recursos necesarios a lo largo de un trayecto (buffers, velocidad de transmisión del enlace) para facilitar la comunicación entre los sistemas finales se reservan para la duración de la sesión de comunicación entre los sistemas finales.
En las redes de conmutación de paquetes, estos recursos no se reservan; los mensajes de una sesión utilizan los recursos bajo demanda y, por lo tanto, pueden tener que esperar (es decir, hacer cola) para acceder a un enlace de comunicación.

d) El Multiplexado hace falta para transmitir varias señales en un solo canal para establecer un enlace multicanal.

e) Con FDM, el espectro de frecuencias de un enlace se reparte entre las conexiones establecidas. En concreto, el enlace dedica una banda de frecuencias a cada conexión durante su duración.
En un enlace TDM, el tiempo se divide en tramas de duración fija, y cada trama se divide en un número fijo de ranuras de tiempos. Cuando la red establece una conexión mediante un

enlace, la red dedica un intervalo de tiempo a cada trama a esta conexión. Estas franjas están dedicadas al uso exclusivo de esta conexión, con una franja de tiempo disponible para utilizar (en cada trama) para transmitir los datos de la conexión.

f) FDM requiere un "sofisticado" *hardware* analógico para desplazar la señal a las bandas de frecuencia adecuadas.

6. Existen 4 tipos de retraso (*delay*). Para cada uno de ellos:
   a) Nómbralo y explícalo brevemente.
   b) ¿En qué parte del camino aplican?: medio de transmisión o nodos.
   c) ¿De qué parámetro(s) depende(n)?:
      i. La distancia del enlace
      ii. La velocidad de transmisión del enlace/la red
      iii. La velocidad de propagación del medio
      iv. La congestión del enlace/la red
      v. La "potencia" del *hardware* del nodo
   d) La dependencia anterior, ¿es de forma proporcional o inversamente proporcional?

Retraso de procesamiento
- El tiempo necesario para examinar la cabecera del paquete y determinar hacia dónde dirigirlo es parte del retraso de procesamiento. El retraso de procesamiento también puede incluir otros factores, como el tiempo necesario para comprobar si hay errores en nivel de bits al paquete que se produjeron al transmitir los bits del paquete desde el nodo ascendente al enrutador (*router*) A. Los retrasos de procesamiento en los *routers* de alta velocidad suelen ser del orden de microsegundos o menos. Después de este procesamiento nodal, el *router* dirige el paquete a la cola que precede el enlace con el *router* B.
- Se produce en los nodos.
- Inversamente proporcional a la "potencia" del nodo (cuanto más potente, menos retraso).

Retraso a la cola
En la cola, el paquete experimenta un retraso de cola mientras espera ser transmitido al enlace. La duración del retraso de cola de un paquete específico dependerá del número de paquetes que lleguen antes y que estén a la cola y esperando ser transmitidos al enlace. Si la cola está vacía y no se está transmitiendo ningún otro paquete, el retraso de la cola de nuestro paquete será cero. En cambio, si el tráfico es intenso y hay otros muchos paquetes en espera de ser transmitidos, el retraso de la cola será elevado. El número de paquetes que un paquete que llega puede esperar encontrar es una función de la intensidad y la naturaleza del tráfico que llega a la cola. A la práctica, los retrasos de cola pueden ser del orden de microsegundos a milisegundos.
- Se produce en los nodos.
- Directamente proporcional a la congestión del enlace/la red.

Retraso de transmisión
Suponiendo que los paquetes se transmiten por orden de llegada, como es habitual en las redes de conmutación de paquetes, nuestro paquete sólo se puede transmitir después de que se hayan transmitido todos los paquetes que han llegado antes que él. Denotamos la longitud del paquete por L bits, y la velocidad de transmisión del enlace desde el *router* A al *router* B por R bits/seg. Por ejemplo, para un enlace Ethernet de 10 Mbps, la velocidad es R = 10 Mbps; para un enlace Ethernet de 100 Mbps, la velocidad es R = 100 Mbps. El retraso de transmisión es L/R. Es el tiempo necesario para empujar (es decir, transmitir) todos los bits del paquete al enlace. En la práctica, los retrasos de transmisión suelen ser del orden de microsegundos a milisegundos.
- Se produce en el medio de transmisión.
- Inversamente proporcional a la velocidad de transmisión del enlace/la red.

Retraso de propagación
Cuando un bit se introduce en el enlace, se tiene que propagar hasta el *router* B. El tiempo necesario para propagarse desde el principio del enlace hasta el *router* B es el retraso de propagación. El bit se propaga a la velocidad de propagación del enlace. La velocidad de propagación depende del medio físico del enlace (es decir, fibra óptica, cable de cocer de par trenzado, etc.) y es al rango de:

2 * 10^8 metros/segundo a 3 * 10^8 metros/segundo

que es igual, o algo menos, que la velocidad de la luz. El retraso de propagación es la distancia entre dos *routers* dividida por la velocidad de propagación. Es decir, el retraso de propagación es d/s, donde d es la distancia entre el *router* A y el *router* B y s es la velocidad de propagación del enlace. Cuando el último bit del paquete se propaga hasta el nodo B, este y todos los bits anteriores del paquete se almacenan al *router* B. El proceso completo continúa entonces con el *router* B realizando ahora el reenvío. En las redes de área extensa, los retrasos de propagación son del orden de milisegundos.
- Se produce en el medio de transmisión.
- Es directamente proporcional a la distancia del enlace e inversamente proporcional a la velocidad de propagación del medio.

7. Contesta a las siguientes preguntas:
   a) Explica la diferencia entre Reenvío (*Forwarding*) y Encaminamiento (*Routing*).
   b) ¿Qué es una red con servicio de mejor esfuerzo (*best-effort*)?
   c) ¿Crees que un servicio con red de mejor esfuerzo puede ofrecer lo suficiente para servicios como la reproducción en continuo (*streaming*) de vídeo o conferencias en tiempo real? Justifica brevemente la respuesta.

a)
El reenvío (*forwarding*) se refiere a la acción local del enrutador (*router*) de transferir un paquete desde una interfaz de enlace de entrada a la interfaz de enlace de salida apropiada. El reenvío tiene lugar a escalas de tiempos muy cortas (normalmente unos pocos nanosegundos), por lo cual suele implementarse en hardware.

El enrutamiento (*routing*) hace referencia al proceso de toda la red que determina las rutas de extremo a extremo que los paquetes toman desde el origen hasta el destino. El encaminamiento tiene lugar en escalas de tiempos mucho más largas (normalmente segundos) y, suele implementarse en software.

b)
El servicio de mejor esfuerzo (*best-effort*) es un servicio en que se compromete a tratar el tráfico de la red todo lo bien que pueda en cada momento, pero sin garantizar ningún parámetro de calidad constante.
No se garantiza que los paquetes se reciban en la orden en que se enviaron, ni siquiera se garantiza lo entrega final. Tampoco se garantiza el retraso de extremo a extremo ni el ancho de banda mínimo.

c)
Curiosamente, el modelo de servicio básico de Internet es de mejor esfuerzo, que combinado con un ancho de banda adecuado, ha demostrado ser más que “bueno” para permitir una increíble gama de aplicaciones, incluyendo servicios de vídeo en *streaming* como Netflix y aplicaciones de voz y vídeo sobre IP, y de conferencias en tiempo real como Skype o Zoom.

8. Para que una aplicación que se encuentra en un equipo de una red se pueda comunicar con otra aplicación que se encuentra en otro equipo de otra red hacen falta 3 “direcciones”: Dirección física (p. ej. dirección MAC), Dirección de red (p. ej. dirección IP), y Puerto.
   Contesta a las siguientes preguntas:
   a) Busca y explica para los 3 tipos de direcciones MAC, IP y Puerto: 1) con qué finalidad se usan/qué identifican y 2) en la cabecera de qué capa se utilizan.
   b) ¿Cuántos bits tienen las 3 direcciones? Explica la notación que se utiliza en cada caso.
   c) Busca qué son las 2 partes, Identificador de Red (NetId) e Identificador de equipo (HostId), de una dirección IP. ¿En qué posición de la dirección se encuentran? ¿Qué relación tienen las direcciones de 2 equipos (o *router*) que se encuentran en la misma red?

   a)
   - Dirección física (p. ej. dirección MAC): dirección única para cada tarjeta de red dentro de Internet, y por tanto también dentro de la red. Permite comunicarse de forma eficiente dentro de una misma red. Capa de enlace de datos.
   - Dirección de red (p. ej. dirección IP): dirección única de un ordenador dentro de todo Internet. Identifica una red + un equipo dentro de esta red. Capa de red.
   - Puerto: “dirección” única de una aplicación dentro del ordenador (multitarea). Identifica una aplicación de otra dentro del ordenador. Capa de Transporte.

   b)
   Cada dirección IP tiene 32 bits de longitud (equivalentes a 4 bytes), por lo tanto hay un total de 2^32 (o aproximadamente 4.000 millones) de direcciones IP posibles.
   Las direcciones MAC tienen 48 o 64 bits.
   En cuanto a los puertos, son números de 16 bits (0 hasta 65,535).

   Las direcciones IP suelen escribirse en la llamada notación decimal con puntos (*dotted-decimal*), donde cada uno de los 5 bytes de la dirección se escribe en su forma decimal y se separa con un punto de los otros bytes de la dirección.
   Por ejemplo, considerar la dirección IP 193.32.216.9. El 193 es el equivalente decimal de los primeros 8 bits de la dirección; el 32 es el equivalente decimal de los segundos 8 bits de la dirección, y así sucesivamente.
   Así, la dirección 193.32.216.9 en notación binaria es 11000001 00100000 11011000 00001001.

   Las direcciones MAC se representan en notación de dos puntos hexadecimales (*colon hexadecimal*), que es representar cada uno de los 6 o 8 octetos en notación hexadecimal, separados por dos puntos (‘:’).
   P. ej., una dirección de 48 bits: 00:00:5e:00:53:af

   Los puertos se representan directamente en decimal.

   c)
   El Identificador de red o NetID es el fragmento de la dirección IP que nos dice a qué red pertenece el host.

   El Identificador de equipo, HostId, es el fragmento de una dirección IP que identifica exclusivamente un host en una red TCP/IP específica.

   El NetId son los (n) primeros bits de una dirección IP, mientras que el HostId son los (32-n) bits últimos de la dirección.

   Todos los equipos de la misma red tienen el mismo NetId, pero diferente HostId.

9. Contesta a las siguientes preguntas:
   a) Busca qué son las direcciones de red (o direcciones IP) públicas y las privadas.
   b) ¿Por qué hacen falta las direcciones privadas si ya tenemos las públicas?
   c) En general, el enrutador (*router*) particular de una casa, ¿a cuántas redes está conectado? ¿Cuántas direcciones IP tiene y de qué tipo es/son (pública o privada)?
   d) En casa muchos de nosotros tenemos las mismas direcciones IP 192.168.1.0/24 ("192.168.1.x"), y no tenemos problemas para encaminar los paquetes. Di cual es el mecanismo que lo hace posible, y explica brevemente su funcionamiento.

a)
Las direcciones privadas (o locales, internas) son las que se utilizan dentro de las redes locales (LAN).

Una dirección IP pública es una dirección IP que se usa para acceder en Internet. Las direcciones IP públicas pueden ser encaminadas en Internet, a diferencia de las direcciones privadas.

b)
Todos los dispositivos podrían utilizar direcciones IP públicas, pero como causa del mecanismo de su distribución muchas no se pueden usar, llegó un momento en que no se podían asignar direcciones públicas a todos los equipos que lo desearan.

Por este motivo, se decidió utilizar direcciones privadas (o locales, internas) dentro de las redes locales (LAN).

Dentro de la red se utilizan direcciones privadas ("192.18.1.x"), y los puertos de los equipos que envían los datos, y fuera la red se utiliza la dirección pública del enrutador (*router*), y un puerto disponible de este enrutador.

c)
Hace falta 1 dirección IP para cada una de las redes donde se está conectado, y como el enrutador está conectado a 2 redes (la de casa y la del proveedor de internet), tiene 2 direcciones: 1 dirección privada de la red de casa y 1 típicamente pública (a pesar de que últimamente también hay de privadas) de la red del proveedor).

d)
En casa no tenemos problemas porque el enrutador se encarga de gestionarlo, mediante la Traducción de direcciones de red (NAT, *Network Address Translation*).

Con NAT, el enrutador modifica los datagramas (dirección IP) y segmentos/datagramas de usuario (puerto) que salen y entran, sustituyendo la dirección y puerto origen privados (de la red de casa) por direcciones y puertos públicos en los que salen, y la dirección y puerto de destino, de público a privado, en los que salen en els que surten de los que entran.

1 dirección IP privada + 1 puerto privado <-> 1 dirección IP púbica (del enrutador) + 1 puerto "disponible"

10. Compara la funcionalidad (tanto las similitudes como las diferencias) de los protocolos de transporte UDP y TCP. ¿Para qué tipo de transmisiones es más adecuado un protocolo o el otro?

Se dice que TCP está orientado a la conexión porque antes de que un proceso de aplicación pueda empezar a enviar datos a otro, los dos procesos primero tienen que "hacer un apretón de manos" entre sí, es decir, tienen que enviar algunos segmentos preliminares entre sí para

establecer los parámetros de la transferencia de datos subsiguiente.
UDP no está orientado a la conexión.

Algunas aplicaciones son más adecuadas para UDP por las razones siguientes:
- Menor retraso/Control más fino a nivel de aplicación sobre qué datos se envían y cuándo. Las aplicaciones en tiempo real suelen requerir una velocidad de envío mínima, no quieren retrasar demasiado la transmisión de segmentos y pueden tolerar una cierta pérdida de datos, el modelo de servicio de TCP no se ajusta especialmente bien a las necesidades de estas aplicaciones.
Bajo UDP, luego que un proceso de aplicación pasa datos a UDP, este empaquetará los datos dentro de un segmento UDP y pasará inmediatamente el segmento a la capa de red. TCP, por otro lado, tiene un mecanismo de control de la congestión que estrangula el emisor TCP de la capa de transporte cuando uno o más enlaces entre los hosts de origen y destino se congestionan excesivamente. TCP también continuará reenviando un segmento hasta que el destino acuse recibo del mismo, independientemente del tiempo que tarde la entrega fiable. Estas aplicaciones pueden utilizar UDP e implementar, como parte de la aplicación, cualquier funcionalidad adicional que haga falta más allá del servicio de entrega de segmentos sin complicaciones de UDP.
- No se establece la conexión. TCP utiliza un apretón de manos de tres vías antes de empezar a transferir datos. UDP simplemente se lanza sin ningún tipo de preliminares formales. Así, UDP no introduce ningún retraso para establecer una conexión. Esta es probablemente la razón principal por la cual DNS funciona sobre UDP en lugar de TCP: el DNS sería mucho más lento si funcionara sobre TCP. HTTP utiliza TCP en lugar de UDP, puesto que la fiabilidad es fundamental para las páginas web con texto (a pesar de que existe el protocolo QUIC, *Quick UDP Internet Connection*, utilizado al navegador Chrome de Google, que utiliza UDP como protocolo de transporte subyacente e implementa la fiabilidad en un protocolo de capa de aplicación sobre UDP.
- No hay estado de conexión. TCP mantiene el estado de la conexión en los sistemas finales. Este estado de conexión incluye las memorias intermedias de recepción y envío, los parámetros de control de la congestión y los números de secuencia y acuse de recibo para implementar el servicio de transferencia de datos fiable de TCP y proporcionar el control de la congestión. En cambio, UDP no mantiene el estado de la conexión y no rastrea ninguno de estos parámetros. Por eso, un servidor dedicado a una aplicación concreta puede soportar normalmente muchos más clientes activos cuando la aplicación se ejecuta sobre UDP en vez de TCP.
- Pequeña sobrecarga de la cabecera del paquete. El segmento TCP tiene 20 bytes de sobrecarga de cabecera a cada segmento, mientras que el UDP sólo tiene 8 bytes de sobrecarga.

UDP: Aplicaciones en tiempo real; protocolos de 1 solicitud/1 respuesta, por ejemplo, DNS
TCP: Protocolos que requieren fiabilidad de transmisión (sin pérdida de datos), por ejemplo, HTTP o correo electrónico (SMTP, IMAPv4, POP3).