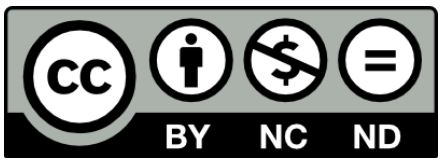
Fundamentos de Redes

Los Del DGIIM, losdeldgiim.github.io

Doble Grado en Ingeniería Informática y Matemáticas
Universidad de Granada



Fundamentos de Redes

Los Del DGIIM, `losdeldgiim.github.io`

Irina Kuzyshyn Basarab
José Juan Urrutia Milán

Granada, 2023-2024

Índice general

1. Introducción a los fundamentos de redes

Objetivos

- Conocer y comprender los principios básicos de las comunicaciones.
- Entender el diseño funiconal en capas de las redes y los conceptos y terminología fundamentales involucrados.
- Comprender desde un punto de vista teórico-conceptual el modelo de referencia OSI y su correspondencia con el modelo de capas usado en Internet.

1.0.1. Introducción

La arquitectura lógica de Internet está diseñada por capas. Veamos el model TCP/IP (también veremos el modelo OSI):

Aplicación que hace uso de la red
Transporte **(TCP/UDP)**
Red **(IP)**
Enlace
Física

Tabla 1.1: Modelo de capas del protocolo TCP/IP

Las dos últimas capas están implementadas en Hardware y las tres primeras en Software, también llamado **NOS** (Network Operative System). En la asignatura veremos las capas altas.

1.1. Sistemas de comunicación y redes

Definición 1.1 (Sistema de comunicación)**.** Es una infraestructura (hw + sw) que permite el intercambio de información. Un sistema típico es el siguiente:

- Tenemos una fuente y un transmisor en un mismo equipo (que es el que va a mandar la información). La fuente genera la información y el transmisor adapta la información al medio.

- Después tenemos el canal de comunicación, el cual produce errores: ruidos, interferencias, diafonías (cuando hay muchos cables en paralelo juntos, puede suceder que la información de un cable se meta en otro).

- Al final tenemos un receptor y el destino (en un mismo equipo). El receptor adapta la información para el destino y éste espera los datos a recibir.

1.1.1. Motivación para usar redes

Para verlo, vamos a hablar de la primera red de comunicaciones que era una red de telefonía móvil. Cada usuario contaba con su línea de teléfono, que conectaba con una central de conmutación local, luego regional y luego nacional, la cual debía conectarse con la central local del destino. Se usaba conmutación de circuitos.

- Inicialemnte se creaba un camino físico juntando cables, llamado circuito.

- Era ineficiente porque no se está hablando todo el tiempo, y los tiempos de silencio el circuito se desaprovecha.

- Era un problema de seguridad el mal funcionamiento de una central, pues dejaba a miles de telefónos sin servicio.

Si ahora pensamos en ordenadores (o equipos más generales, móviles, PCs, portátiles, móviles...) en vez de móviles y cambiamos las centrales de conmutación por routers, contamos con muchísimos caminos para conectar dos ordenadores, haciendo más segura la red.

Ahora ya no tenemos un camino físico, sino que son los routers quienes deciden por dónde enviar los paquetes y en qué momento. Los routers tienen colas, lo que genera algo de retardo, pero tiene la ventaja de que se usa mucho mejor el canal y hay más seguridad, pues hay más caminos que solo uno.

De una red esperamos:

- Autonomía.

- Interconexión.

- Intercambio de información con eficacia y transparencia.

Definición 1.2 (Red)**.** Sistema de comunicación con sistemas finales o terminales autónomos (con capacidad para procesar información) que facilita el intercambio eficaz y transparente de información. Concretamente tenemos:

- **Hosts:** sistemas finales o terminales autónomos. Los que transmiten y reciben datos.

- **Subred:** infraestructura para el transporte de información, formada por líneas de transmisión y nodos o elementos de conmutación: routers y switches.

En cuanto a medios de transmisión, originalmente se usaban cables de pares (pensado para transmitir 4kHz, la media de la voz humana), luego cables coaxiales, que mejoraron mucho; y fibra óptica que puede transmitir sin interferencias, por lo que es el mejor medio guiado existente. Los cables trenzados son para distancias más cortas, Ethernet por ejemplo.

1.1.2. Topologías de redes

En bus: es la más sencilla, pero como el medio es común, todos intentan acceder y se producen colisiones.

En anillo: un círculo en el que tenemos los distintos computadores. Es similar al bus pues el medio es compartido. Una versión habitual es **token ring**, testigo de anillo. Como el medio es común, se van pasando el testigo evitando así colisiones.

En estrella: todos están conectados a un centro principal, típicamente un switch.

A diferencia del bus, en este caso cada cable es independiente del resto. Si un PC pone algo en una toma el resto no lo escuchan. Cada línea tienen una cola para guardar a dónde enviar los paquetes y el switch tiene un procesador que coge los paquetes de dichas colas y los envían a las salidas. Es una topología mucho más segura por el hecho de no compartir el medio.

En árbol: típica en redes empresariales. Se suele estructurar en tres niveles:

- Primer nivel: red troncal.
- Segundo nivel: red de división.
- Tercer nivel: red de acceso.

Los equipos de primer y segundo nivel suelen ser switches. Pueden aparecer ciclos en el árbol. Ethernet no tiene ningún mecanismo para evitar que un paquete se mueva en círculo, lo que echa la red abajo.

El protocolo STP hace que cualquier topología quiten enlaces redundantes que forman bucles.

Mallada: todos los nodos están conectados entre sí por medios independientes. Es muy fiable, si se cae un enlace tienes más caminos para llegar a tu destino. Pero no es escalable, si metemos un nodo n-ésimo, hay que meter $n-1$ enlaces. Pero para redes pequeñas está bien. Dentro de una empresa la red troncal puede seguir esta topología para evitar caídas importantes.

Híbrida: se usa una mezcla de todas. Es la más utilizada.

Definición 1.3 (CSMA/CD)**.** Ethernet para compartir un medio común como un bus utiliza CSMA/CD: acceso múltiple sintiendo la portadora (Carrier Sense). CD viene de "detecta colisiones": si lo que hay en el cable no es lo que ha puesto (si hay más ruido), da error.

Definición 1.4 (CSMA/CA)**.** Wifi usa este protocolo, no puede escuchar el medio, primero escucha que no haya nadie, envía el mensaje y recibe confirmación. Si no recibe confirmación, hay colisión.

1.1.3. Clasificación de redes

Según tamaño y extensión:

- LAN (Local Area Network). Red de área local, suele ser el mismo edificio.
- MAN (Metropolitan Area Network). Red de área metropolitana, para conectar un campus o una ciudad.
- WAN (Wide Area Network). Red de área extensa, redes desponibles en todo el país, como las redes telefónicas.
- PAN (Personal Area Network). Red de área personal, todo lo que puede tener una persona, relojes, portátiles, cascos...

Según tecnología de transmisión:

- Difusión: lo que pone alguien en el medio le llega a todos. HUB.
- Punto a punto: Sólo estoy unido a un nodo. Switch.

Según el tipo de tranferencia de datos:

- Simple: solo transmite o recibe. Por ejemplo los TDT (para que una televisión analógica reciba señal digital).
- Half-duplex: transmite y recibe pero no simultáneamente. Por ejemplo el WIFI, aunque cambia muy rápido, por lo que no nos damos cuenta.
- Full-duplex: transmite y recibe simultáneamente. Por ejemplo Ethernet.

1.1.4. Nomenclatura típica en figuras (Iconos)

HUB: es un concentrador: permite centralizar los nodos de una red de computadoras, se implementa mediante un bus.

Bridge: funciona como un switch, pero uniendo tecnologías distintas, funciona a nivel 2.

Switch: tiene muchas bocas y tiene LAN. Necesitan un router para conectarse a otro switch.

Router: tiene pocas bocas y lo que hace es conectar distintas redes.

Cortafuegos: bloquea el acceso no autorizado, permitiendo el autorizado.

NAT: traducción de direcciones de red.

Switch multicapa: (no entra) todas las bocas de un switch son de la misma LAN, pero esto a veces no nos interesa. Podemos hacer redes virtuales (VLAN) dividiendo un mismo switch en varias redes, permitiéndonos esto conectar dos switches distintos con la misma VLAN. Pero esto no nos permite movernos por distintas redes en el mismo switch, habría que pasar por un router, ya que necesitamos movernos a nivel de red para cambiar de red. Esto sí se puede hacer con un switch multicapa.

1.2. Diseño y estandarización de redes

La idea principal que se sigue al diseñar redes es solucionar los problemas en capas. Se estandarizan Modelos de Referencia (no son implementaciones, solo una referencia): definición de las capas y las funcionalidades de cada una. Los principios que se siguen son que las funcionalidades distintas tienen que estar en distintas capas y minimizar el flujo de información entre las capas.

A continuación detallamos los problemas que tiene que resolver una red por capas:

A nivel físico: hay que ver como transmitir los datos. Hay distintos tipos de codificaciones para enviar bits de información.

Capa de enlace: se encarga de los mecanismos de acceso al medio. Si hay un medio común, antes de transmitir datos tiene que asegurarse de que ningún equipo está transmitiendo. Suele seguir dos protocolos:

- MAC, control de acceso al medio.
- LLC, control de acceso lógico para las primeras retransmisiones. Si algún paquete llega mal, retransmite varias veces.

Capa de red: una vez llegado a este punto, se asume que no han habido colisiones en la comunicación. Esta capa se encarga principalmente de:

- El direccionamiento: saber dar una dirección y tener un identificador dentro de la red.
- El encaminamiento: saber cómo llegar al destino.

Capa de transporte: se encarga de recuperar los paquetes que en la capa de enlace no se ha podido, es la capa de la fiabilidad.

- Corrige errores.
- Gestiona la congestión.
- Control de flujos: si hay un receptor más lento que el emisor, debe decirle al emisor que disminuya la velocidad de emisión, para adecuarse a la del receptor.

Además se encarga de la multiplexación de datos: mediante puertos (los veremos más adelante) le indica al SO a qué aplicación corresponde cada paquete.

Capa de aplicación: los clientes y los servidores deben buscar alguna forma de comunicarse.

1.2.1. Modelo OSI vs TCP/IP

Aplicación
Presentación
Sesión
Transporte
Red
Enlace
Física

Tabla 1.2: Modelo OSI.

Aplicación
Transporte
Red
Red subyacente

Tabla 1.3: Modelo TCP/IP.

El modelo OSI fue propuesto por la ISO y el TCP/IP por el Internet Engineering Task Force. Las tres primeras capas del modelo OSI se corresponden con la capa de aplicación, y las dos última con la red subyacente, que es la parte física. Esta última en el modelo TCP/IP depende un poco de la tecnología está implementada de una forma u otra, pero la comunicación con la capa de red no puede variar, pues la capa de red si está estandarizada.

- Las capas físicas solo se encargan de hacer la primera conexión.
- La capa de red, salto a salto se encarga de llegar al destino, usando routers y sus tablas de encaminamiento.
- Una vez hecho el encaminamiento, las capas superiores son solo de los extremos, son los computadores de los extremos los que se comunican.
- Por tanto, los computadores tienen las 5 capas (en el caso de TCP/IP) y los routers solo las 3 más bajas.

1.3. Terminología, conceptos y servicios

En la Figura 1.1 vemos el camino que siguen los datos que le manda un emisor a un receptor. Cada capa le pone una cabecera a lo que le llega de arriba. A este proceso se le llama encapsulamiento. La capa física por su parte manda señales eléctricas en vez de bits.

Por otra parte, cuando la información va subiendo por las capas del modelo, cada capa le quita la cabecera correspondiente y ve si tiene que hacer algo.

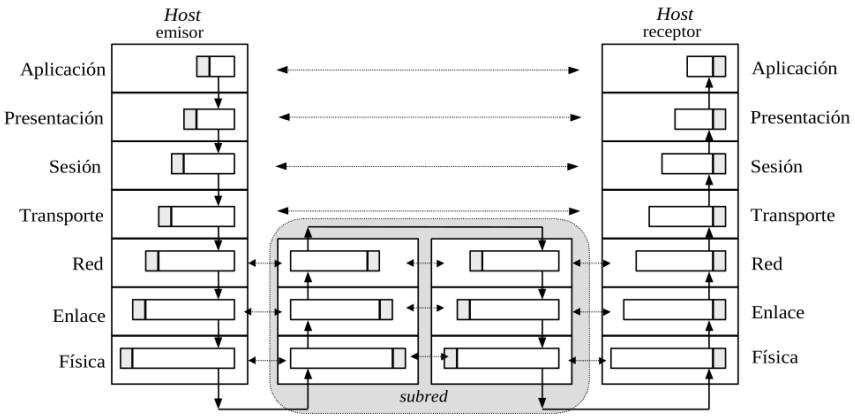


Figura 1.1: Comunicación real frente a comunicación virtual.

Las flechas continuas es por donde realmente pasa la información y las discontinuas representan la comunicación virtual u horizontal, que en breves definiremos.

Las capas del mismo tipo en distintos equipos hablan directamente en sentido abstracto por medio de otras capas. Aunque solo se habla entre las capas del mismo tipo se usan recursos que nos proporcionan otras capas adyacentes.

1.3.1. Definiciones importantes

SDU: unidad de datos de servicio. Son los datos de una determinada capa.

PDU: unidad de datos de paquete. SDU + cabecera.

Comunicación real: o vertical. Es por donde realmente va la información.

Comunicación virtual: u horizontal. Es la comunicación entre capas del mismo tipo en distinto equipo. No es una comunicación directa, sino a través de otras capas.

Entidades pares: entidades iguales, las que hablan entre sí.

Protocolo: cómo se comunican las entidades pares, los paquetes que se mandan.

Interfaz: cómo hablan las entidades adyacentes.

Servicio: lo que proporciona una capa a otra.

Capa proveedora/usuaria del servicio: una capa es proveedora de la de arriba y usuaria de la de abajo.

Pila de protocolos: los protocolos de las distintas capas.

Arquitectura de red: modelo de referencia + pila de protocolos.

1.3.2. Retardos en la comunicación

Supongamos que queremos transmitir un paquete entre dos equipos (terminales), por medio de otro, un router. Veamos los retardos que tenemos en la comunicación.

- Para empezar, tenemos que saber el tamaño del paquete: en Ethernet un paquete típico son $1500\text{Bytes} \cdot 8^{\text{bits}}/_{\text{Byte}}$. Luego el tiempo que se tarda en transmitir el paquete sale como resultado de dividirlo entre los bps. Esto es el **tiempo de transmisión** que depende exclusivamente de la tarjeta de red.

- Una vez transmitido, hay un retardo, que se corresponde con el **tiempo de propagación** $\left({}^{\text{distancia}}/_{\text{velocidad de transmisión}}\right)$, que es el tiempo que que ocurre entre que se escribe el primer bit y que llega el primer bit al siguiente equipo. La velocidad de transmisión inalámbrica es a la velocidad de la luz y la de un cable suele rondar $^2/_3$ de la velocidad de la luz. En los cables se pierde un bit por cada millón de bits, y suele deberse a colas llenas.

- Cuando el paquete llega a un equipo intermedio, en nuestro caso el router; éste lo mete en una cola hasta que pueda procesarlo. El tiempo que está el paquete en dicha cola depende de la situación del equipo. Luego el tiempo que está el paquete en dicho equipo es el **tiempo de cola** más el **tiempo de procesamiento** (del orden de milisegundos) más el **tiempo de acceso al medio**.

- Ahora se envía el paquete al equipo destino, el tiempo de propagación, pese a calcularse de forma análoga al anterior, probablemente no coincida debido a distintas distancias, tarjetas de red....

1.3.3. Tipos de servicios

Relacionado con el nivel de transporte, hay dos clasificaciones importantes: orientadas a conexión y a fiabilidad.

- Orientado a conexión (SOC): antes de mandar un paquete comprueba que el otro equipo esté encendido.

- Orientados a la fiabilidad: se asegura de que todo funcione bien (que todos los bits de un archivo estén bien). Además, si algo falla la conexión se termina.

  Para tener un protocolo fiable, contamos con los siguientes mecanismos:

  - Control de conexión. Ser fiable implica ser orientado a conexión.
  - Control de errores.
  - Control de congestión. Hablamos de congestión cuando las colas de los routers se llenan y empiezan a descartar paquetes.
  - Control de flujo.
  - Entrega ordenada. Si se envían muchos paquetes, estos deben llegar en orden.

  Algunos protocolos que estudiaremos más adelante:

- TCP es un servicio orientado a conexión y fiable.
- UDP es un servicio no orientado a conexión y no es fiable. La finalidad de este protocolo es ser rápido.

1.4. Internet: topología y direccionamiento

Internet tiene dos cosas importantes:

- Los protocolos de comunicación.
- Cómo se gestiona Internet. Las direcciones IP son únicas en todo el mundo (salvo algunas, ya lo veremos).

1.4.1. Organización topológica

Los operadores se establecen según la siguiente jerarquía:

- **Tier 3**: los más cercanos a los usuarios, ISPs (Internet Service Provider).
- **Tier 2**: necesitan pasar por una Tier 1 para llegar a toda Internet y ofrecen servicios de conectividad a operadores de Tier 3.
- **Tier 1**: los que componen la estructura troncal de Internet. Están todos comunicados entre sí y hay como mínimo en dos continentes.

Hay dos tipos de relaciones entre operadores:

- **Tránsito:** conexiones entre distintos tier. Por ejemplo un tier 3 paga a un tier superior para enviar datos.
- **Peering:** conexiones entre el mismo tier.

Antiguamente, para que un ISP de un país hable con otro del mismo país había que ir hasta EEUU por falta de recursos. Más tarde se pusieron puntos neutros en cada país para comunicar operadores dentro de un mismo país.

1.4.2. Red Iris

Es la red española para investigación. Todas las universidades públicas y centros de investigación están conectados a ella.

La red se divide según autonomía y por otro lado tiene conexiones externas con la red científica europea.

1.4.3. Direccionamiento por capas

- Enlace: depende de la tarjeta de red. Nesesitamos conocer la dirección MAC del siguiente punto. Las direcciones MAC son de la siguiente forma: AA:BB:CC:DD:EE:FF. Son en teoría únicas en todo el mundo. Pero realmente se ponen aleatorias para evitar seguimientos (puesto que si sabemos la dirección MAC de una tarjeta podemos hacer seguimiento de paquetes).
- Red: direcciones IP: A.B.C.D. Las públicas son únicas en todo el mundo, las privadas no.
- Transporte: direccionamiento a través de puertos, que identifican a qué proceso va un determinado paquete.
- Aplicación: Nombres de dominio mediante DNS.

2. Seguridad en redes

Objetivos

- Comprender la importancia de la seguridad en las comunicaciones y aprender cómo desplegar mecanismos básicos de seguridad en redes de computadores e Internet.
- Conocer los aspectos de seguridad en redes: confidencialidad, autentificación, no repudio, integridad y disponibilidad.
- Entender los conceptos básicos de la seguridad en redes, como el uso de algoritmos de clave secreta, de clave pública, intercambio de claves...
- Comprender qué son los certificados digitales y las autoridades de certificación, y los diferentes mecanismos que se pueden implementar con certificados.
- Conocer algunos de los principales protocolos de comunicación seguros, como TLS e IPSec, y los mecanismos que lo utilizan.

2.1. Introducción

Una red de comunicaciones es **segura** cuando se garantizan todos los aspectos de seguridad, con lo que no hay protocolos ni redes 100 % seguras. Definamos brevemente los aspectos de seguridad que vamos a estudiar:

- **Confidencialidad / privacidad:** cuando transmitimos algo a un receptor queremos que solo dicho receptor sea capaz de ver el mensaje. Se consigue con el cifrado.
- **Autenticación:** las entidades son quien dicen ser. Se consigue con algoritmos de Reto-Respuesta o doble cifrado.
- **No repudio o irrenunciabilidad:** no podemos renunciar de haber participado en una transacción, es una prueba legal ante un juez. Se consigue con la firma deigital o con el doble cifrado con certificado, pero ha de haber una entidad fiable.
- **Integridad:** que los datos no sean manipulados por el camino. Se consigue con funciones hash o compendios (resúmenes).
- **Disponibilidad:** el sistema mantiene las prestaciones de los servicios independientemente de la demanda. (No se ve en la asignatura).

Debe haber seguridad en todos los niveles de la red, el grado de seguridad lo fija el punto más débil.

Definición 2.1 (Ataque de seguridad)**.** Cualquier acción intencionada o no que menoscaba cualquiera de los aspectos de seguridad.

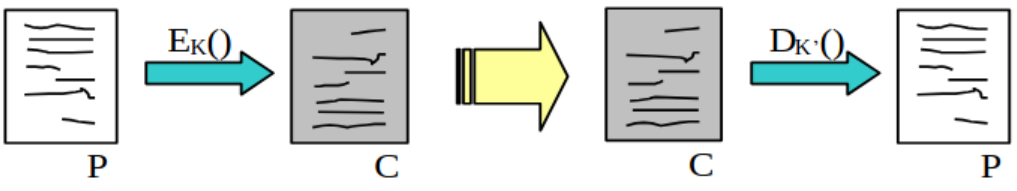
- **Sniffing:** escuchar comunicaciones, por ejemplo Wireshark.
- **Snooping (phishing):** suplantar a alguien.
- **Man in the Middle:** se pone alguien en medio de dos equipos que se comunican y intercepta todos los mensajes que se transmiten.
- **Distributed Denial of Service (DDoS):** mandar muchas peticiones hasta conseguir que el servicio deje de funcionar. Distributed si los ataques vienen de muchos sitios.
- **Malware:** software malicioso. Troyanos, gusanos, spyware, backdoors, rootkits, keyloggers, ransomware (se encriptan todos o parte de los datos y se pide un recate de los datos).

Los mecanismos de seguridad que vamos a estudiar son:

- Cifrado (simétrico y asimétrico)
- Autentificación con clave secreta (reto-respuesta)
- Intercambio de Diffie-Hellman (establecimiento de clave secreta)
- Funciones Hash (Hash Message Authentification Code)
- Firma digital
- Certificados digitales

2.2. Cifrado

Se trata de un procedimiento para garantizar la confidencialidad. Tenemos un texto plano a transmitir (P) y lo ciframos con una función E_k, que dará lugar a un texto cifrado (C), el cual se mandará a través del canal de comunicaciones. Llegará al otro extremo y será descifrado con una función $D_{k'}$ de desencriptado. Los algoritmos de cifrado y descifrado normalmente son conocidos y la dificultad reside en las claves k y k'.



Veremos dos tipos de algoritmos de cifrado:

- Cifrado simétrico (clave secreta), misma clave distintas funciones.
- Cifrado asimétrico (clave privada y clave pública), distinta clave misma función.

2.2.1. Cifrado simétrico

Se llama simétrico porque se usa la misma clave para cifrar y para descifrar. Esto significa que es una clave secreta que comparten los dos.

DES, (Data Encryption Standard)

Un algoritmo que se suele usar se basa en realizar permutaciones y funciones XOR encadenadas. Al final lo que obtenemos es una sustitución, por lo que con la misma entrada el resultado siempre será el mismo. Por tanto, DES no es más que un esquema de sustitución que usa palabras de 56 bits. Para arreglar la posibilidad de descifrarlo, se utiliza un esquema reentrante, donde la salida de aplicar una transformación se usa para la creación del cifrado de la siguiente palabra a cifrar. Se cifran palabras de 64 bits. De este modo, quien recibe el mensaje codificado necesita conocer la última entrada usada para codificar y podrá aplicar el proceso inverso.

DES doble y 3DES

Son mejoras de DES que proporcionan robustez al algoritmo. Se toman dos claves k_1 y k_2 y para cifrar se toma una función E y su inversa D y se concatenan E_{k_1}, D_{k_2}, E_{k_1} y para descifrar se concatenan D_{k_1}, E_{k_2}, D_{k_1}.

IDEA

Usa la misma idea que DES pero con claves de 128 bits, lo que aumenta la complejidad de violación del protocolo.

2.2.2. Cifrado asimétrico

Cada usuario A tiene una clave pública K_{pub_A} y una clave privada K_{pri_A} distintas. Conociendo la pública no es posible conocer la privada, por lo que la pública la conocen todos pero la privada solo la conoce A.

$$C = K_{pub_A}(P) \rightarrow P = K_{pri_A}(C)$$

$$C = K_{pri_A}(P) \rightarrow P = K_{pub_A}(C)$$

Esta es la forma de funcionar de cualquier cifrado con clave pública. Además hay una correspondencia biunívoca entre las claves públicas y privadas.

RSA

Es un algoritmo para sacar las claves del cifrado asimétrico.

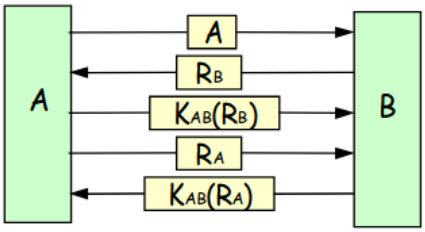
1. Elegimos p y q primos grandes ($> 10^{100}$)
2. $n = p \cdot q$ y $z = (p-1) \cdot (q-1)$
3. Elegimos d primo relativo de z
4. Calculamos e tal que $e \cdot dmodz = 1$
5. $K_pub = (e, n)$ y $K_{pri} = (d, n)$ de tal forma que:
   - $C = P^e modn$
   - $P = C^d modn$

2.3. Autenticación

Pongámonos en el supuesto de que dos equipos, A y B, quieren autenticarse. Lo más sencillo es que cada uno tenga una BBDD con el usuario y la clave que comparten con el otro y que cada uno le mande a otro su usuario y la clave y que el servidor confirme si son correctos o no. Esto es vulnerable pero se usa en muchos servicios. Se hacen algunas mejoras de este método como por ejemplo que el envío de información se haga a través de túneles cifrados.

2.3.1. Reto-respuesta

Tenemos como antes una BBDD con la clave que comparten A y B. Supongamos ahora que A quiere conectarse con B. Lo primero que hace es enviarle su identidad, lo que no es un dato sensible. Lo que ahora manda B es un reto, un número aleatorio. A calcula X, un cifrado con la clave compartida del reto. B cifra el reto también, X' y cuando A le responde comprueba que X=X' y así autentica a A. Ahora falta autenticar a B, y esto se hace de la misma forma.

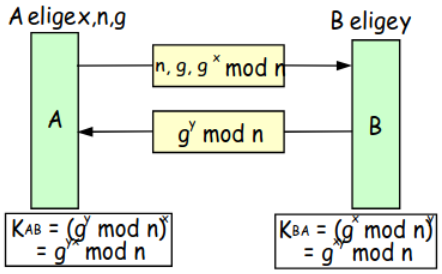


Este sistema tiene algunas vulnerabilidades:

- **Ataque por repetición:** el atacante escucha por mucho tiempo y va guardando las respuestas, hasta que se repita un reto y así puede responder. **Solución:** que el reto no se pueda repetir (NONCE), Ej: fecha + hora.
- **Ataque por reflexión:** el atacante antes de responder manda el mismo reto, espera a la respuesta y la reenvía. **Solución:** usar dominios de claves disjuntas.

2.3.2. Intercambio de Diffie-Hellman

Permite establecer una clave secreta entre dos entidades a través de un canal no seguro.



Este sistema es vulnerable al ataque man-in-the-middle:
El atacante se pone en medio e intercepta el mensaje de A, y le manda su propia clave a B, y esa misma clave se la responde a A, y espera que B le responda. De esta forma, hace de mensajero invisible, A y B ni siquiera saben que están escuchando sus mensajes.

2.4. Funciones Hash

Son funciones de forma que dada una palabra P, nos da una palabra a modo de resumen o compendio de los datos, R. P puede ser de cualquier longitud, R, sin embargo, suele ser de longitud fija y ademas la función es unidireccional, irreversible, es decir, no se puede obtener P a partir de R. Dado un mensaje P, se envía P junto con su hash.

Para que no se pueda modificar el mensaje a P' e incluir su resumen se tienen varias alternativas:

- En el resumen podemos meter la clave que se comparte. A este hash (P + K_{AB}) se le suele llamar MAC (Message Authentification Code).
- Cifrar el hash con la clave compartida.

Así conseguimos integridad del mensaje y por otra parte autenticación con los MAC.

MD5 (Message Digest)

Dada una palabra nos da un resumen de 128 bits. El algoritmo trabaja sobre bloques de 512 bits, por lo que si la palabra no tiene un número de bits múltiplo de 512 rellenamos con 100...0 hasta que sea congruente con 448módulo 512, y se le añade un campo de longitud de 64 bits. A continuación se divide el mensaje en bloques de 512 bits y hacemos un procesamiento secuencial por bloques, es decir, que cada salida sirve como entrada para la siguiente caja MD5.

SHA (Secure Hash Algorithm)

Funciona igual que MD5 pero los resúmenes son de 160 bits.

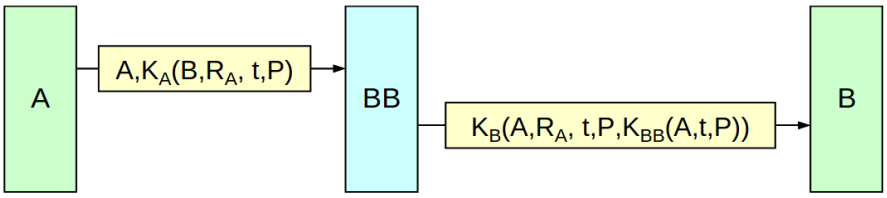
2.5. Firma digital y certificados digitales

Una **firma digital** intenta ser un sustituto de una firma escrita para poder garantizar el **no repudio** en nuestras acciones en Internet. Con ellas conseguimos:

- Autenticación por parte del receptor de la identidad del emisor.
- No repudio por parte del firmante.
- El emisor obtiene garantía de no falsificación. (Integridad).

2.5.1. Firma con clave secreta o Big Brother (BB)

El protocolo consiste en usar una entidad en la que todos los usuarios confían y que vigilará las transacciones de todos los usuarios. Es una especie de man-in-the-middle. Si A quiere enviar un mensaje a B, el BB formará parte de la comunicación haciendo de puente que guardará cada una de las transacciones realizadas.



El BB tiene una clave compartida con todos. A le manda a BB:

A: Identificador.

B: El destinatario.

R_A**:** Un resumen para dar integridad.

t: el instante de tiempo.

P: texto plano, el mensaje a enviar.

Todo menos el identificador va cifrado con la clave que comparte A con BB. El mensaje es recibido y reenviado por BB añadiéndole algunos detalles.

$K_{BB}(A, t, P)$**:** Esta clave solo la tiene el BB lo que prueba ante un juez quién ha hecho dicha transacción, en qué momento y el contenido de la transacción.

$K_B(\ldots)$**:** Además todo va cifrado con la clave de B para confidencialidad.

2.5.2. Firma digital con clave asimétrica o Doble cifrado

Supongamos que A le quiere mandar un mensaje a B. La idea se basa en lo siguiente:

- $k_{pri_A} \equiv$ autenticación, solo A ha podido cifrarlo.
- $k_{pub_B} \equiv$ confidencialidad, solo B podrá descifrarlo.

De esta manera, juntando las dos obtenemos autenticación + confidencialidad: $k_{pub_B}(K_{pri_A}(P))$ (puede cifrarse al revés, no es relevante).

Sin embargo todo esto no garantiza el **no repudio**, puesto que nada nos garantiza que A sea el dueño de la clave. Para garantizarlo necesitamos los certificados digitales que deben ser emitidos por autoridades de certificación, que son entidades fiables.

Definición 2.2 (Autoridades de certificación (AC))**.** Entidad que garantiza la asociación entre identidad y claves.

Certificado digital

El usuario obtiene sus claves pública y privada, envía una solicitud, firmada digitalmente, a la AC indicando su identidad y clave pública. La AC comprueba la firma y emite el certificado: este tiene los datos que a continuación detallaremos y va firmado digitalmente por la clave firmada de la AC con objeto de que el certificado no pueda falseficarse. Campos de un certificado X.509:

- Identidad del usuario.
- Su clave pública.
- La AC que lo ha emitido.
- Periodo de validez.
- Algunos datos más como la versión del certificado, el número de serie...

2.6. Protocolos seguros

La seguridad se divide en dos tipos:

- **Perimetral:** uso de *firewalls*, *sistemas de detección de intrusiones* o *sistemas de respuesta*.
- **Seguridad en protocolos:** consiste en usar protocolos para garantizar seguridad.
  - Capa de aplicación: PGP o SSH.
  - Capa de sesión (entre aplicación y transporte): TLS, SSL.
  - Capa de red: IPSec.

2.6.1. PGP (Pretty Good Privacy)

Es un protocolo de correo electrónico seguro.

Emisor:
- R = MD5(P)
- FD = $Kpr_A(R)$
- Z = ZIP(FD + P)
- C = $IDEA_K(Z)$ + $Kpu_B(K)$
- M = B64(C)

Receptor:
- C = $B64^{-1}(M)$
- K = $Kpr_B(Kpu_B(K))$
- Z = $IDEA_K^{-1}(IDEA_K(Z))$
- FD + P = $ZIP^{-1}(Z)$
- R = $Kpu_A(FD)$
- R' = MD5(P)
- R' = R ??

El emisor hace un resumen del mensaje, lo firma; esto lo comprime junto con el mensaje, cifra esa compresión con IDEA y una clave de sesión K (generada solo para esa sesión), que manda cifrada con la clave pública del receptor y esto lo codifica con base 64 (esto no tiene nada que ver con seguridad). El resultado de esto último es lo que le manda al destinatario. El receptor simplemente tiene que hacer el proceso inverso para sacar el mensaje y con el resumen comprobar que no se ha modificado.

Con este proceso conseguimos:

- Confidencialidad: el mensaje va cifrado.
- Integridad: gracias al resumen.
- Autenticación: gracias al cifrado con la clave privada.
- No repudio: solo si hay un certificado digital.

2.6.2. TLS, SSL

Se usan para muchos protocolos (HHTPS, IMAPS, SSL-POP). TLS fue el original y SSL se acabó popularizando más. Hacen más o menos lo mismo, pero no son compatibles. Lo que hacen en esencia es crear túneles cifrados.

SSL

No es un protocolo sino una familia de ellos.

- SSL Record Protocol: encapsula los protocolos y ofrece un canal seguro con privacidad, autenticación e integridad. Lo que se hace es que cuando voy a mandar datos, los parto en fragmentos, cada fragmento lo comprimos, hago un resumen y cifro datos + resumen. Y esto último es lo que se transmite, encapsulado en un paquete TCP.
- SSL Handshake Protocol: se negocia el algoritmo de cifrado, la función hash, autentica al servidor, el cliente genera claves de sesión (temporales) con el algoritmo de Diffie-Hellman o aleatorias y que irán cifradas con la clave pública del servidor.
  El problema del hombre en medio se resuelve autenticando al servidor. Así, si el mensaje que me llega con $g^y modn$ no viene del servidor no lo tomo.

- SSL Assert Protocol: informa sobre errores en la sesión.
- Change Cipher Spec Protocol: para notificar cambios en el cifrado.

2.6.3. IPSec

Su objetivo es garantizar autenticación, integridad y opcionalmente privacidad a nivel IP. Crea túneles unidireccionales.

Definición 2.3 (Túnel)**.** Es un sitio donde entra un paquete y a la salida tendremos exactamente el mismo paquete. Para ello encapsulamos el paquete dentro de otro paquete. Si el túnel va cifrado (la parte de los datos va cifrada) entonces el túnel es seguro.

Son tres procedimientos:

1. Establecimiento de una "Asociación de seguridad": con el objetivo de establecer una clave secreta (Diffie-Hellman), con la previa autenticación. Es simplex. Vulnera el carácter NO orientado a conexión.
2. Garantizar la autenticación e integridad de los datos mediante las "Cabeceras de autenticación".
3. (Opcional) Garantizar la privacidad de los datos mediante el protocolo de "Encapsulado de seguridad de la carga".

Dos tipos de túneles:

- Modo transporte: la asociación se hace extremo a extremo entre el host origen y destino.
- Modo túnel: la asociación se hace entre dos routers intermediarios. Útil por ejemplo si una empresa quiere comunicar dos sucursales, en vez de comunicar cada dos trabajadores, se comunican mediante esos routers intermediarios y solo hacen falta dos túneles (para ambas direcciones).

3. Relaciones de Problemas

3.1. Introducción

Ejercicio 3.1.1. Explique brevemente las funciones de cada una de las capas del modelo de comunicación de datos OSI.

El modelo de comunicación de datos OSI cuenta con 7 niveles o capas:

1. Capa física: Se encarga de la parte física de la transmisión de los datos. Encontramos distintas formas de codificar los bits para su envío.

   Realiza funciones adicionales, como la codificación del canal.

2. Capa de enlace: Se encarga de los mecanismos de acceso al medio. En caso de haber un medio compartido por varios dispositivos, debe encargarse de no transmitir datos cuando otro medio lo está haciendo y de hacerlo cuando el canal se encuentre libre.

   En esta capa nos encontramos con los protocolos MAC y LLC.

3. Capa de red: Se encarga principalmente del direccionamiento de equipos (saber dar una dirección y tener un identificador dentro de la red) y del encaminamiento de datos (saber cómo mandar los paquetes al destinatario).

4. Capa de transporte: Se encarga principalmente de la fiabilidad de las comunicaciones:

   - Corrección de errores.
   - Manejar el congestionamiento de la red.
   - Controlar el flujo de datos (reducir velocidades si el receptor no es capaz de adecuar la velocidad de recibo a la de envío).
   - Realizar la multiplexación de los datos (ya que un mismo equipo puede tener varias aplicaciones que estén recibiendo datos a la vez).

5. Capa de sesión.

6. Capa de presentación[1].

7. Capa de aplicación: Se encarga de decidir qué datos envía a qué equipo, así como de interpretar los datos recibidos por otros equipos.

[1] No se han mencionado en clase las funcionalidades de las capas de presentación ni de sesión.

Ejercicio 3.1.2. Si la unidad de datos de protocolo en la capa de enlace se llama trama y la unidad de datos de protocolo en la capa de red se llama paquete, ¿son las tramas las que encapsulan los paquetes o son los paquete los que encapsulan las tramas? Explicar la respuesta.

Son las tramas las que encapsulan a los paquetes, ya que son las capas inferiores (en este caso, la de enlace) las que encapsulan la información de las capas superiores (en este caso, la de red) para su envío.

De esta forma, los paquetes son la PDU (*Protocol Data Unit*) de la capa de red, que se convierte en el SDU (*Servie Data Unit*) de la capa de enlace, la cual añade su cabecera al mismo convirtiéndolo en su PDU.

Ejercicio 3.1.3. Averigüe qué son los sistemas de representación de datos "*Little Endian*" y "*Big Endian*". ¿Puede un host que utilice representación *Little Endian* interpretar mensajes de datos numéricos provenientes de un host que utilice representación *Big Endian* y viceversa? Discuta la respuesta.

Sí que puede, para ello, debe haber un determinado protocolo que permita indicar qué codificación llevan los datos en binario. De esta forma, en alguna parte de la cabecera de los paquetes enviados, debe haber un bit que indique si los valores numéricos que se envían estén en *Big Endian* o en *Little Endian*.

Ejercicio 3.1.4. Cuando se intercambia un fichero entre dos hosts se pueden seguir dos estrategias de confirmación. En la primera, el fichero se divide en paquetes que se confirman individualmente por el receptor, pero el fichero en conjunto no se confirma. En la segunda, los paquetes individuales no se confirman individualmente, es el fichero entero el que se confirma cuando llega completo. Discutir las dos opciones.

Suponiendo que enviamos n paquetes de datos, la primera forma envía de vuelta al emisor n paquetes de confirmación, uno por cada paquete. De la segunda forma, el receptor espera a unir todos los paquetes en un fichero completo (y a verificar que no se ha perdido ningún paquete del mismo) para enviar el mensaje de verificación.

De la segunda forma se envían menos mensajes de verificación al emisor, por lo que la posibilidad de congestión de red por paquetes de verificación es menor. Sin embargo, en caso de que un paquete no consiga llegar o llegue en mal estado, no será hasta el final del envío de todos los paquetes que el receptor no genere el mensaje al emisor, por lo que en caso de errores en la comunicación, hay un mayor tiempo en la comunicación, al tener que esperar a que el receptor tenga todos los paquetes. Además, en el caso de error, el receptor no informará de qué paquete ha llegado mal, por lo que deberá pedir al emisor que reenvíe todos los paquetes de nuevo.

Resumiendo, ambas estrategias de confirmación tienen sus pros y sus contras. Dependiendo de la situación (si queremos mayor velocidad en la comunicación o si queremos menor saturación de red), puede interesarnos una u otra.

Ejercicio 3.1.5. ¿Para qué sirve el programa *ping*? ¿y el programa *traceroute*?

El programa *ping* usa el protocolo ICMP para enviar un paquete a un equipo, el cual tratará de responder con un paquete de confirmación de recepción del primer paquete. Sirve para comprobar la conexión y el buen funcionamiento de la red existente entre dos equipos. También sirve para calcular empíricamente la latencia de la conexión.

Por otra parte, el programa *traceroute* sirve para consultar todos los nodos intermedios por los que pasan los paquetes que salen de un emisor y llegan a un receptor, junto con la latencia de cada salto. Se usan varios paquetes ICMP con un valor creciente del campo TTL (*Time To Live*) para que cada salto intermedio devuelva un paquete de error ICMP con el valor del TTL que ha recibido. De esta forma, el emisor puede saber cuántos saltos intermedios hay entre él y el receptor, así como la latencia de cada uno de ellos.

Ejercicio 3.1.6. ¿Qué protocolos de un paquete puede cambiar un router? ¿En qué circunstancias?

Puede cambiar el TTL (*time to life*) del paquete[2].

Ejercicio 3.1.7. Averigue qué ISPs operan en España.

Algunos de los ISPs (*Internet Service Providers*) que operan en España son:

- Movistar.
- Vodafone.
- Orange.
- Jazztel.
- MásMóvil.
- Yoigo.
- Digi.

Ejercicio 3.1.8. ¿Qué es una aplicación cliente-servidor? ¿y una aplicación *peer-to-peer*?

Una aplicación cliente-servidor es una aplicación que depende de otra que probablemente esté en un equipo remoto (llamado servidor) para su funcionamiento.

Un ejemplo de aplicación cliente-servidor es una página web: tenemos aplicaciones que se ejecutan en local en cada equipo que accede a una determinada url. Dicha aplicación solicita datos a una aplicación que se encuentra en un equipo remoto, la cual proporciona datos (por ejemplo, accediendo a una base de datos) como respuesta a los datos solicitados por la aplicación que se ejecuta en cada equipo de forma local.

[2]Todavía no se ha mencionado en clase nada más acerca de esto.

Por otra parte, una aplicación *peer-to-peer*[3] es una aplicación que se distribuye entre varios equipos (que pueden estar muy lejanos entre sí) de forma que todas las aplicaciones tienen la misma relevancia en el buen funcionamiento del sistema.

Ejercicio 3.1.9. Describa brevemente la diferencia entre un *switch*, *router* y un *hub*.

Para responder a la pregunta, usamos además información que hemos aprendido en el Tema 2:

- Un switch es un nodo en una red que permite conectar tantos equipos como deseemos (normalmente, estos tienen 48 bocas de entrada RJ45 en el caso de conectar los equipos por ethernet) a una red. Funcionan a nivel de enlace, luego no tienen una dirección IP asociada.

- Un router es un nodo en una red que permite conectar redes distintas entre sí. Para ello, disponen de distintas tarjetas de red, cada una asociada a una red que se encuentra conectada al router. Disponen por tanto de varias direcciones IP, una por cada red a la que se conecta. Funciona a nivel de red.

  Presenta en su interior la tabla de enrutamientos, que permite el encaminamiento en la capa de red. Cuenta además con el NAT, que permite traducir direcciones de IP privadas a públicas y viceversa.

- Un *hub* es un nodo en una red que permite implementar la difusión. Se trata de un conjunto de bocas ethernet que internamente funcionan como un bus. Cada vez que un paquete se envía por una de las bocas, este es reenviado a todas las demás bocas, por lo que todos los equipos conectados al *hub* reciben el paquete.

Ejercicio 3.1.10. ¿Qué diferencia, en el contexto de una red de computadores, existe entre la tecnología de difusión y la tecnología punto-a-punto?

La tecnología de difusión permite enviar un paquete desde un equipoo y hacer que este sea recibido por el resto de equipos que estén conectados a la misma red (o hacer llegar estos a equipos en distintas redes). Es cada dispositivo el que decide si el paquete es para él o no.

Por otra parte, la teconología punto-a-punto permite el envío de paquetes desde un equipo a otro usando un medio directo, por lo que el destino está implícito desde que se envía el paquete. Esta tecnología es más rápida y segura, pero su escalabilidad es mucho menor.

Ejercicio 3.1.11. Un sistema tiene una jerarquía de protocolos de n capas. Las aplicaciones generan mensajes de M bytes de longitud. En cada capa se añade una cabecera de h bytes. ¿Qué fracción del ancho de banda de la red se llena con cabeceras? Aplique el resultado a una conexión a 512 kbps con tamaño de datos de 1500 bytes y 4 capas, cada una de las cuales añade 64 bytes cabecera. ¿Qué velocidad real de envío de datos resulta?

[3]En español, podemos pensar en "entre pares".

Debemos sumar a los M bytes iniciales que proporcionan las aplicaciones n veces (la capa de aplicación también incluye una cabecera) h bytes, por lo que la longitud de los mensajes que de verdad se envían es de $n \cdot h + M$ bytes. Por tanto, por cada $n \cdot h + M$ bytes enviados, $n \cdot h$ de ellos son de cabeceras:

$$\frac{n \cdot h}{n \cdot h + M} \cdot 100 \ \% \text{ de ancho de banda que se llena de cabeceras}$$

Si ahora partimos de 1500 bytes iniciales y añadimos 4 veces (una por capa) 64 bytes, estamos en realidad enviando mensajes de longitud:

$$4 \cdot 64 + 1500 = 256 + 1500 = 1756 \text{ bytes}$$

Por tanto, el ($^{256}/_{1756} = 0{,}145786$) 14.58 % de la red se emplea para enviar cabeceras, luego se aprovecha el ($1 - 0{,}145786 = 0{,}854214\,\%$) 85.42 % de la red para el envío de datos.

Si enviamos paquetes a una velocidad de 512kbps, en realidad estaremos enviando datos a una velocidad real de:

$$0{,}854214 \cdot 512 = 437{,}357568 \text{ kbps}$$

Ejercicio 3.1.12. Clasifique como de *difusión* o *punto a punto* cada uno de los siguientes sistemas de transmisión:

1. Radio y TV: Difusión, ya que es un emisor (en este caso, una cadena de televisión o radio) que difunde paquetes a cualquiera que tenga sintonizado dicho canal.

2. Redes inalámbricas (WLAN): Difusión, ya que cualquier equipo puede conectarse a la red y recibir los paquetes que se envían de forma inalámbrica.

3. ADSL: Punto a punto, ya que la conexión se establece mediante un cable. Usa en medio único.

4. Redes de cable: Puede implementar ambas tecnologías, ya que puede ser punto a punto (si cada equipo tiene su propio cable) o de difusión (si todos los equipos comparten el mismo cable).

5. Comunicaciones móviles (por ejemplo, GSM, UMTS, ...): Difusión, ya que (de nuevo) cualquier equipo puede recibir los paquetes que se envían por la red.

Ejercicio 3.1.13. Clasifique los siguientes servicios como orientados a conexión/no orientados a conexión y confirmados/sin confirmación. Justifique la respuesta.

Recordamos que los medios orientados a conexión son aquellos que comprueban si el receptor está disponible antes de enviar la información, y que los confirmados son aquellos que confirman la recepción del mensaje.

1. Correo postal ordinario: No es orientado a conexión (ya que no comprobamos anteriormente que el destinatario esté disponible, simplemente enviamos la carta) y es sin confirmación, ya que tras enviar la carta nada nos garantiza que el destinatario nos responda, pese a pedirlo explícitamente.

2. Correo certificado: No es orientado a conexión por la misma razón que el correo normal. Sin embargo, es confirmado, ya que al recibir el destinatario la carta debe firmar para que el emisor sea consciente de que la carta ha sido recibida.

3. Envío y recepción de fax.

4. Conversación telefónica. Es orientado a conexión, puesto que no se puede establecer una llamada si la otra persona no coge el teléfono. Además, es confirmado, ya que la otra persona ha de responder para que se produzca una comunicación.

5. Domiciliación bancaria de recibos.

6. Solicitud de certificado de empadronamiento. Es no orientado a conexión, ya que no se comprueba si el destinatario está disponible antes de enviar la solicitud. Además, es con confirmación, ya que el ayuntamiento envía un documento que certifica que el solicitante está empadronado.

Ejercicio 3.1.14. ¿Cuál es el tiempo necesario en enviar un paquete de 1000 Bytes, incluidos 50 Bytes de cabecera, por un enlace de 100 Mbps y 10Km? ¿cuál es el tiempo mínimo desde que se envía hasta que se recibe confirmación? ¿qué relación hay entre este tiempo y los temporizadores en, por ejemplo, las capas de enlace y transporte?

En primer lugar, hemos de calcular el retardo de transmisión T_t, que es el tiempo que se tarda en enviar el paquete por el enlace. Para ello, tenemos que:

$$T_t = 10^3\,\mathrm{B} \cdot \frac{8\,\mathrm{b}}{1\,\mathrm{B}} \cdot \frac{1\,\mathrm{s}}{100 \cdot 10^6\,\mathrm{b}} = 80 \cdot 10^{-6}\,\mathrm{s} = 80\,\mu\mathrm{s}$$

Por otra parte, el retardo de propagación T_p es el tiempo que se tarda en enviar el paquete por los 10Km de cable. Para ello, suponiendo que la velocidad de transmisión es $^2/_3c = 2 \cdot 10^8\,\mathrm{m/s}$, tenemos que:

$$T_p = 10 \cdot 10^3\,\mathrm{m} \cdot \frac{1\,\mathrm{s}}{2 \cdot 10^8\,\mathrm{m/s}} = 50 \cdot 10^{-6}\,\mathrm{s} = 50\,\mu\mathrm{s}$$

Por tanto, el tiempo total que se tarda en enviar el paquete es de $T_t + T_p = 130\,\mu\mathrm{s}$.

Veamos ahora el tiempo mínimo desde que se envía hasta que se recibe confirmación. Además de los tiempos anteriores, hemos de tener en cuenta el tiempo de procesamiento del paquete, el retardo de transmisión del paquete de confirmación y el retardo de propagación del paquete de confirmación. Tenemos que:

- No se proporciona información del tiempo de procesamiento del paquete. No obstante, en los dispositivos modernos, este retardo es de varios órdenes de magnitud menor que los otros, por lo que se puede considerar despreciable.

- El retardo de propagación del paquete de confirmación es el mismo, puesto que la distancia recorrida es la misma.
- EL retardo de transmisión sí difiere, puesto que el tamaño del paquete de confirmación difiere. Normalmente, estos solo incluyen una cabecera, por lo que este tiempo, notado por T_{ACK}, es:

$$T_{\text{ACK}} = 50\,\text{B} \cdot \frac{8\,\text{b}}{1\,\text{B}} \cdot \frac{1\,\text{s}}{100 \cdot 10^6\,\text{b}} = 4 \cdot 10^{-6}\,\text{s} = 4\,\mu\text{s}$$

  Un temporizador de control de flujo en capa de enlace o de transporte debe ser suficientemente mayor a este tiempo mínimo para evitar un re-envío inmediato de paquetes ante cualquier eventualidad mínima en la red, como un retardo en las colas (mayor retardo de procesamiento) por un cierto nivel de congestión.

Por tanto, el tiempo total mínimo que se tarda en enviar el paquete y recibir confirmación es de:

$$T_{\text{total}} = T_t + T_p + T_{\text{ACK}} + T_p = 80\,\mu\text{s} + 50\,\mu\text{s} + 4\,\mu\text{s} + 50\,\mu\text{s} = 184\,\mu\text{s}$$

3.2. Capa de red

Ejercicio 3.2.1. Estime el tiempo involucrado en la transmisión de un mensaje de datos para la técnicas de conmutación de circuitos (CC) y de paquetes mediante datagramas (CDP) y mediante circuitos virtuales (CPCV) considerando los siguientes parámetros:

- M: longitud en bits del mensaje a enviar.
- V: velocidad de transmisión de las líneas en bps.
- P: longitud en bits de los paquetes, tanto en CPD como en CPCV.
- H_d: bits de cabecera de los paquetes en CPD.
- H_c: bits de cabecera de los paquetes en CPCV.
- T: longitud en bits de los mensajes intercambiados para el establecimiento y cierre de conexión, tanto en CC como en CPCV.
- N: número de nodos intermedios entre las estaciones finales.
- D: tiempo de procesamiento en segundos en cada nodo, tanto en CC como en CPD y en CPCV.
- R: retardo de propagación, en segundos, asociado a cada enlace, en CC, en CPD y en CPCV.

Ejercicio 3.2.2. Un mensaje de 64 kB se transmite a lo largo de dos saltos de una red. Ésta limita la longitud máxima de los paquetes a 2 kB y cada paquete tiene una cabecera de 32 bytes. Las líneas de transmisión de la red no presentan errores y tienen una capacidad de 50 Mbps. Cada salto corresponde a una distancia de 1000 km. ¿Qué tiempo se emplea en la transmisión del mensaje mediante datagramas?

Ejercicio 3.2.3. Suponga que una red de datagramas usa cabeceras de H bytes y que una red de paquetes de circuitos virtuales utiliza cabeceras de h bytes. Determine la longitud M de un mensaje que se consigue transmitir más rápido haciendo uso de la técnica de conmutación de circuitos virtuales que mediante la de datagramas. Suponga que los paquetes tienen la misma longitud en ambas redes y que los retardos de procesamiento son idénticos

Ejercicio 3.2.4. Una aplicación audiovisual en tiempo real hace uso de conmutación de paquetes para transmitir voz a 32 kbps y vídeo a 64 kbps a través de la conexión de red de la figura **??**. Se consideran paquetes de voz e información de audio con dos longitudes distintas: 10 ms y 100 ms. Cada paquete tiene además una cabecera de 40 octetos.

a. Encuentre para ambos casos el porcentaje de bits suplementarios que supone la cabecera.

b. Dibuje un diagrama temporal e identifique todas las componentes del retardo extremo a extremo en la conexión anterior. Recuerde que un paquete no puede ser transmitido hasta que esté completo y que no se puede retransmitir hasta que no se haya recibido completamente. Suponga despreciables los errores a nivel de bit.

c. Evalúe todas las componentes del retardo de las que se dispone suficiente información. Considere las dos longitudes de paquete aceptadas. Suponga que la señal se propaga a una velocidad de 1 km/5 microsegundos y considere dos velocidades para la red troncal: 45 Mbps y 1,5 Mbps. Resuma el resultado para los cuatro posibles casos en una tabla con cuatro entradas.

d. ¿Cuál de las componentes anteriores implica la existencia de retardos de cola?

Ejercicio 3.2.5. Imagine una situación donde hay cinco routers RA-RE. RA, RB y RC se conectan cada uno a una red local A, B y C, siendo cada router única puerta de enlace de cada red. RA, RB y RD están conectados entre sí a través de un switch. RC, RD y RE están conectados entre sí a través de un switch. RE conecta a Internet a través de la puerta de acceso especificada por el ISP. Especifique tablas de encaminamiento en los routers. Asigne a voluntad las direcciones IP e interfaces necesarias.

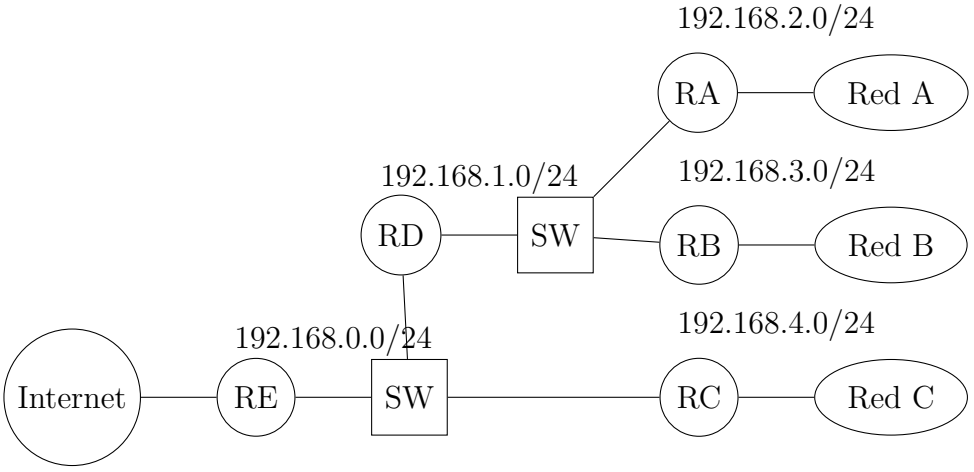


Figura 3.1: Situación del ejercicio 5

En primer lugar, asignaremos las direcciones IP y las interfaces de cada router. Cada router presenta una conexión por la izquierda y otra por la derecha, por lo que sólo usaremos dos interfaces de cada router, luego asociaremos dos direcciones IP a cada router.

Por comodidad, asociaremos las conexiones de la derecha de cada router a la interfaz *ether0*, y las conexiones de la izquierda de cada router a la interfaz *ether1*.

Una vez añadidas las interfaces, procederemos a asocciar direcciones IP a cada interfaz de cada router:

- RA:
  - Tendrá IP 192.168.2.1 en la red 192.168.2.0/24.

  - Tendrá IP 192.168.1.2 en la red 192.168.1.0/24.

- RB:
  - Tendrá IP 192.168.3.1 en la red 192.168.3.0/24.
  - Tendrá IP 192.168.1.3 en la red 192.168.1.0/24.
- RD:
  - Tendrá IP 192.168.1.1 en la red 192.168.1.0/24.
  - Tendrá IP 192.168.0.2 en la red 192.168.0.0/24.
- RC:
  - Tendrá IP 192.168.4.1 en la red 192.168.4.0/24.
  - Tendrá IP 192.168.0.3 en la red 192.168.0.0/24.
- RE:
  - Tendrá IP 192.168.0.1 en la red 192.168.0.0/24.
  - Su IP en la red que le conecta con Internet la proveerá el ISP.
- Suponemos que RE se conecta a Internet a través de un router con IP 33.33.33.33 en la red 33.33.0.0/16.

Procedemos ahora a rellenar las tablas de encaminamiento de cada router:

Red destino	Máscara	Siguiente salto	Interfaz
192.168.2.0	255.255.255.0	*	ether0
192.168.1.0	255.255.255.0	*	ether1
192.168.3.0	255.255.255.0	192.168.1.3 (RB)	ether1
0.0.0.0	0.0.0.0	192.168.1.1 (RD)	ether1

Tabla 3.1: Tabla de encaminamiento para RA.

Red destino	Máscara	Siguiente salto	Interfaz
192.168.3.0	255.255.255.0	*	ether0
192.168.1.0	255.255.255.0	*	ether1
192.168.2.0	255.255.255.0	192.168.1.2 (RA)	ether1
0.0.0.0	0.0.0.0	192.168.1.1 (RD)	ether1

Tabla 3.2: Tabla de encaminamiento para RB.

Red destino	Máscara	Siguiente salto	Interfaz
192.168.4.0	255.255.255.0	*	ether0
192.168.0.0	255.255.255.0	*	ether1
192.168.0.0	**255.255.252.0**	192.168.0.2 (RD)	ether1
0.0.0.0	0.0.0.0	192.168.0.1 (RE)	ether1

Tabla 3.3: Tabla de encaminamiento para RC.

Donde hemos agrupado la Red A (192.168.2.0/24), B (192.168.3.0/24) y la red 192.168.1.0/24 en la superred 192.168.0.0/22. Notemos que dentro de las direcciones de la superred se encuentran las direcciones de la forma 192.168.0.x, que no se encuentran en dicha superred. Sin embargo, tenemos una entrada específica para dichas direcciones, con una máscara de mayor prioridad (más 1s), por lo que no tenemos problema[4]

Red destino	Máscara	Siguiente salto	Interfaz
192.168.1.0	255.255.255.0	*	ether0
192.168.0.0	255.255.255.0	*	ether1
192.168.2.0	255.255.255.0	192.168.1.2 (RA)	ether0
192.168.3.0	255.255.255.0	192.168.1.3 (RB)	ether0
192.168.4.0	255.255.255.0	192.168.0.3 (RC)	ether1
0.0.0.0	0.0.0.0	192.168.0.1 (RE)	ether1

Tabla 3.4: Tabla de encaminamiento para RD.

Red destino	Máscara	Siguiente salto	Interfaz
192.168.0.0	255.255.255.0	*	ether0
33.33.0.0	255.255.0.0	*	ether1
192.168.4.0	255.255.255.0	192.168.0.3 (RC)	ether0
192.168.0.0	**255.255.252.0**	192.168.0.2 (RD)	ether0
0.0.0.0	0.0.0.0	33.33.33.33 (Router ISP)	ether1

Tabla 3.5: Tabla de encaminamiento para RE.

Donde hemos vuelto a usar la superred 192.168.0.0/22 que engloba a Red A, B y 192.168.1.0/24.

Ejercicio 3.2.6. Asigne las direcciones de subred en la siguiente topología a partir de 192.168.0.0 para minimizar el número de entradas en las tablas de encaminamiento, asumiendo que en las redes LAN puede haber hasta 50 PCs.

[4]Si no tuviéramos dicha entrada, tendríamos un problema, ya que si mandamos un paquete a 192.168.0.26, por ejemplo, iría a la superred que hemos definido pero algún router se daría cuenta de que no sabe llegar a 192.168.0.0/24.

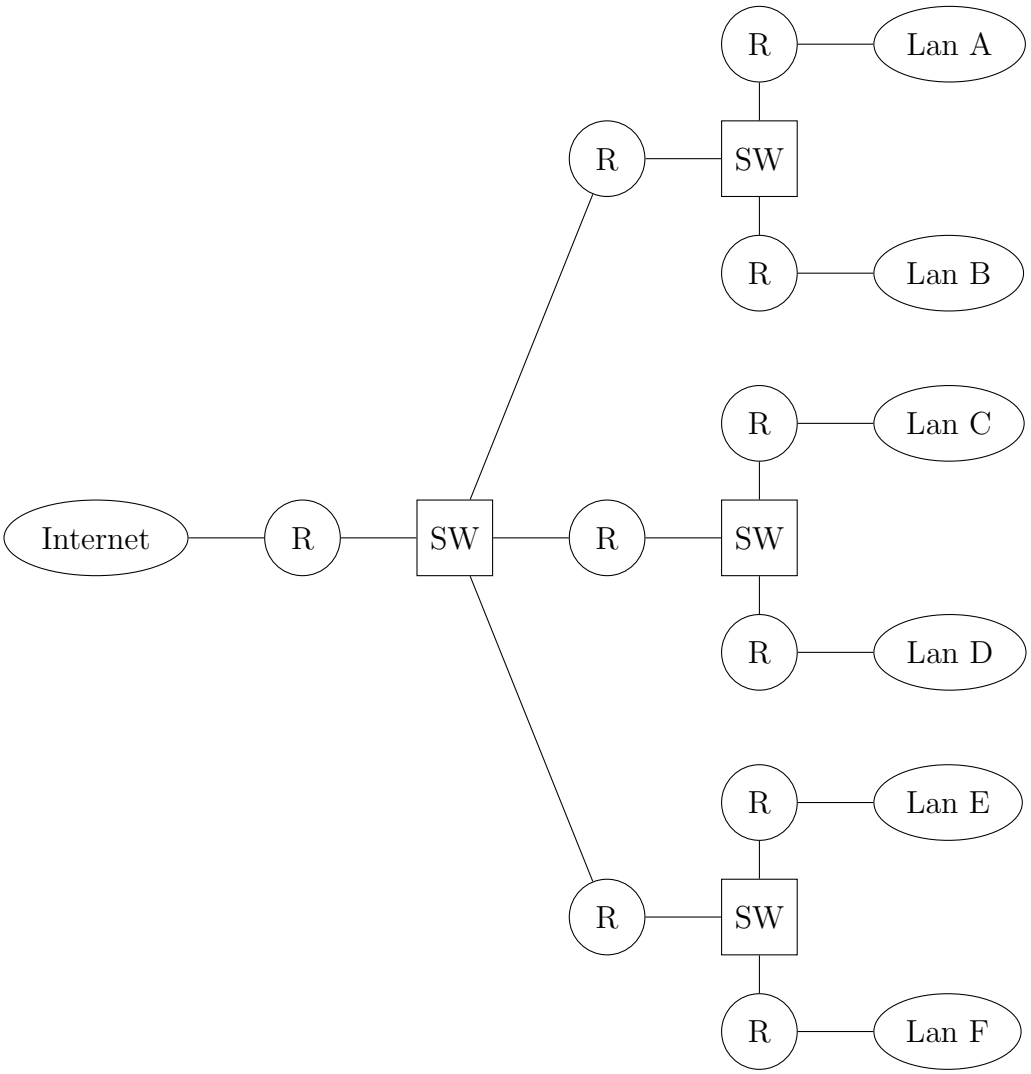


Figura 3.2: Situación del ejercicio 6

Ejercicio 3.2.7. Un datagrama de 4020 bytes pasa de una red Token Ring con THT 8 ms (MTU 4400) a una Ethernet (MTU 1500) y después pasa por un enlace PPP con bajo retardo (MTU 296). Si ese mismo datagrama pasara directamente de la red Token Ring al enlace PPP (sin pasa por la red Ethernet) ¿habría alguna diferencia en la forma como se produce la fragmentación? Especifique en ambos casos los fragmentos obtenidos.

Ejercicio 3.2.8. ¿Cómo podría utilizar ICMP para hacer una estimación de la latencia entre dos entidades finales? ¿Y para estimar la latencia de un enlace en particular entre dos routers?

Ejercicio 3.2.9. Considere la subred de la figura. Se utiliza el algoritmo de encaminamiento de vector distancia, habiéndose recibido en el encaminador C los siguientes vectores de encaminamiento: desde B (5, 0, 8, 12, 6, 2), desde D (16, 12, 6, 0, 9, 10) y desde E (7, 6, 3, 9, 0, 4). Los retardos medidos a B, D y E son, respectivamente, 6, 3 y 5. ¿Cuál es la nueva tabla de encaminamiento de C? Indique la línea de salida y el retardo esperado.

Ejercicio 3.2.10. Considere la red mostrada en la figura 3.3, en la que se representan 4 nodos unidos con enlaces. En cada enlace se indica el retardo sufrido por los mensajes al atravesarlo. Los nodos utilizan un protocolo de encaminamiento dinámico de tipo distribuido en el que la métrica está basada en el retardo. Se pide lo siguiente:

a. Escriba las tablas de encaminamiento para todos los nodos de la red una vez haya pasado el tiempo suficiente para que dichas tablas se construyan de forma estable.

b. Considere que los nodos envían y actualizan sus tablas cada 5 segundos, siendo la primera actualización en $t = 0$ s. Suponga que, en $t = 12$ s., el enlace BD pasa a tener un retardo de 3 s. ¿Cuál será el encaminamiento desde el nodo A hasta el nodo B cuando las tablas se estabilicen de nuevo?

c. ¿En qué instante comenzará dicho encaminamiento a funcionar? Justifique su respuesta explicando qué sucederá desde $t = 12$ s. hasta dicho instante y también después del mismo.

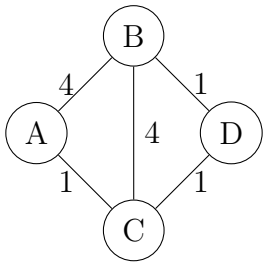


Figura 3.3: Grafo para el ejercicio 10.

Ejercicio 3.2.11. En la topología de red adjunta se indica la capacidad, en Kbps, de las líneas de transmisión entre los distintos nodos intermedios. Considérese al respecto que los enlaces son *full-duplex* y que la velocidad es la misma para cada uno de los sentidos. Por otra parte, la tabla anexa especifica el tráfico, en paquetes/segundo, entre cada par de nodos. Además, en cursiva se indica la ruta (secuencia de nodos) seguida en la transmisión. Teniendo en cuenta todo lo anterior, determine el retardo medio en el envío de un paquete sobre la red global.

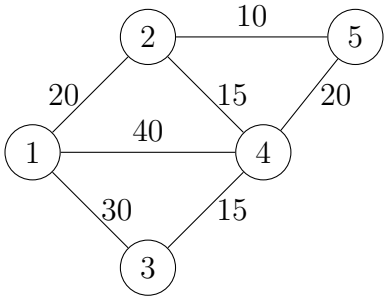


Figura 3.4: Grafo para el ejercicio 11.

		Nodo destino				
		1	2	3	4	5
Nodo origen	1		2-*12*	3-*13*	1-*14*	2-*145*
	2	2-*21*		4-*243*	2-*24*	2-*25*
	3	3-*31*	4-*342*		3-*34*	5-*345*
	4	1-*41*	2-*42*	3-*43*		1-*45*
	5	2-*541*	2-*52*	5-*543*	1-*54*	