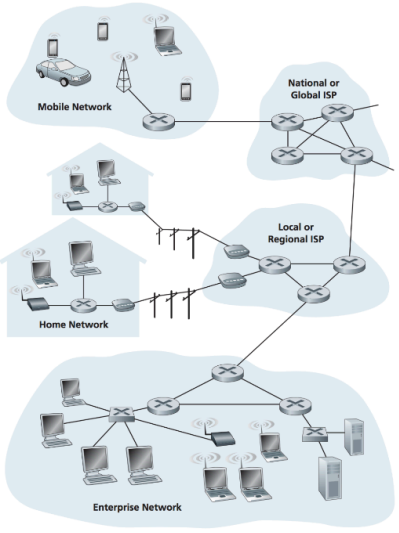
Redes de Computadores

Permitir que un proceso se **comunique** con otro.
¿Dónde se encuentra el otro proceso? ¿Qué medio utilizar para el envío? ¿Cómo se envía la información por medio de red? ¿A quién se le transmite el mensaje? ¿Cómo se encuentra el destinatario? ¿El destinatario está dispuesto a recibir el mensaje?



Internet
Dispositivos de origen/destino: **hosts.**
Dispositivos de intercambio: **routers, switches.**
Puntos de acceso: **modem, access point, cell tower.**
Proveedores de aceso: **Internet Service Provider (ISP).**
Unidades de Información: **Paquetes.**
Reglas para intercambio de paquetes: **Protocolos (**contrato de comunicación entre distintas partes acerca de la manera en que se producirá la comunicación).

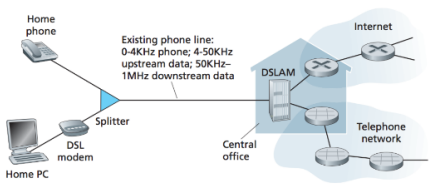
-> Transport Control Protocol (TCP)
-> Internet Protocol (IP)

Para que todos sigan el mismo protocolo se acuerdan **estándares: Internet Engineering Task Force (IETF) | Request For Comments (RFC).**
El internet es una plataforma para **aplicaciones distribuidas** (app con distintos componentes que se ejecutan en entornos separados, normalmente en diferentes plataformas conectadas a través de una red. Son de 2 niveles [cliente-servidor], tres niveles [cliente-middleware-servidor] y multinivel) que genera procesos que *interactúan* a través de la red. Estos utilizan **Application Programming Interface (API).**

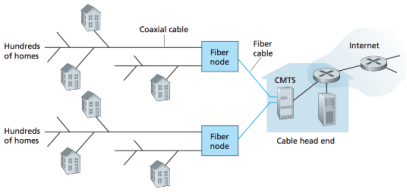
Hardware de Red

Comunicación ***end-to-end:*** Comunicación entre hosts. Algunos actúan como **clientes** y otros como **servidores.**
Clientes: Dispositivos livianos que solicitan servicios | Suelen estar en control del usuario.
Servidores: Dispositivos poderosos que proveen servicios | Suelen vivir en **data centers** (Centro de procesamiento de datos. Alberga un sistema de información de componentes asociados como telecomunicaciones)
El host puede **acceder** a la red de distintas maneras:

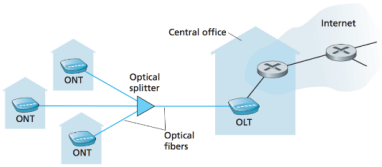
- **Digital Subscriber Line (DSL) |** 1.12Mbps downstream, 16 Mbps upstream
  - **DSL Módem** utiliza línea telefónica tradicional y convierte señales digitales en análogas
  - **DSL Access Multiplexer (DSLAM)** reconvierte señales análogas a digital.
  - Medio *broadcast p*rivado



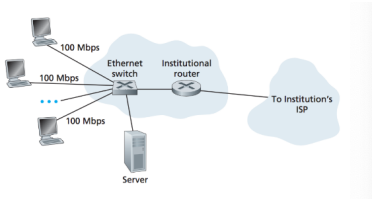
- **Cable |** 42.8Mbps downstream, 30.7 Mbps upstream
  - Utiliza infraestructura de TV por cable. Mezcla enlaces de fibra



coaxial (Hybrid Fiber Coaxial [HFC]).
  - Utiliza **cable modem** (modem especial diseñado para modular y demodular señal de datos sobre infraestructura de televisión por cable).
  - Medio *broadcast* compartido.
- **Fiber-to-the-Home (FTTH):** 1Mbps a 1Gbps
  - Optical Network Terminator (ONT).
  - Optical Line Terminator (OLT).
  - Conversión de señal óptica-eléctrica.



- **Dial - Up:** 2.4Kbps 56Kbps
  - Enlace a través de línea telefónica.
- **Satellite:** 1 Mbps
  - Enlace satelital para zonas rurales
- **Ethernet:** 10Mbps-10Gbps
  - Estándar de conexión por pares trenzados de cobre
  - Implementan **Local Area Network (LAN)** [Red que conecta los ordenadores en un área relativamente pequeña a través de líneas telefónicas y ondas de radio.



- **Wi-Fi:** 54Mbps, 150Mbps,…
  - Estándar de conexión inalámbrico de corto alcance.
  - Implementan **Wireless LAN (WLAN).**
- **Wide Area: 3G, 4G, LTE**

Medio Físico

Información se codifica en **bit.** Se transmiten como ondas:

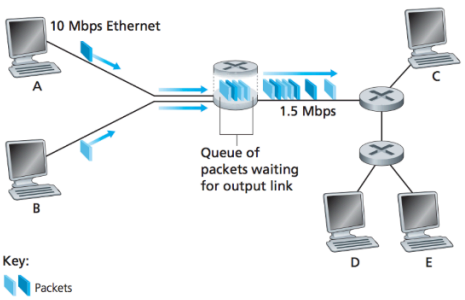
- **Medios Guiados:** Sólidos
  - P**ares trenzados de cobre:** Se comportan como una antena. Se categorizan según qué tan trenzados están. Recorren varios kilómetros sin atenuarse (~5 km). 10 Mbps a 10 Gbps. Se usa para telefonía, ethernet e implementación de LAN's.
  - **Cable coaxial de cobre:** Alta resistencia al ruido externo. Ancho de banda depende de la distancia. Fácil de modificar para insertar nodos nuevos. Medio compartido.
  - **Fibra Óptica:** Inmune a interferencia electromagnética. Cables hechos de fibra de vidrio. Alta velocidad de transferencia. Baja atenuación con la distancia (100 Kms). Tiene monomodo y multimodo dependiendo de los haces de luces.
- **Medios No Guiados:**
  - **Inalámbricos Terrestres:** Señal se debilita en el aire al propagarse. Se interfiere con otras señales. Propagación multicamino.
  - **enlace satelital**

Funcionamiento de Red

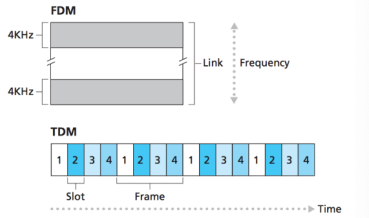
Transmisión en un enlace: **L** bits sobre un enlace de **R** bits/sec, demora **L/R.**

Packet Switching: Dispositivos de red intercambian paquetes. Dispositivo (router) almacén paquetes. Los examina para luego determinar el punto de reenvío (**forward)**. Para esto, usa **Tablas de reenvío (forwarding tables)**.

Las tasas de transmisión son determinadas por el enlace más lento. El router puede sufrir **congestión**. Si existen N enlaces iguales en la ruta, el **tiempo de transmisión (end-to-end delay)** = **N x L/R.**



Circuit Switching: Establecimiento previo de conexión entre origen-destino. Este enfoque es usado por **comunicación telefónica.** Es muy sensible a la saturación ante conexiones simultáneas. Si el enlace es de 1Mbps, con 4 conexiones simultáneas, cada una recibe hasta 250Kbps. Como se está usando un medio compartido es necesario mantener separada cada comunicación.



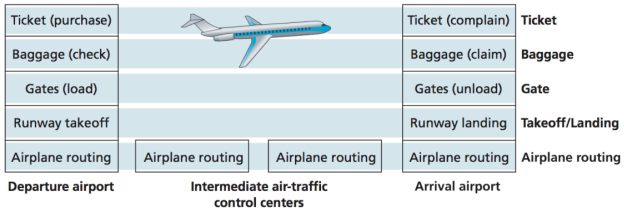
- **Frequency-Divison Multiplexing (FDM):** Banda de frecuencia (ancho de banda) para cada transmisión.
- **Time-Division Multiplexing (TDM):** División por tiempo entre emisores. Cada emisor tiene un turno para transmitir en todo el ancho de banda.

Conceptos de Software

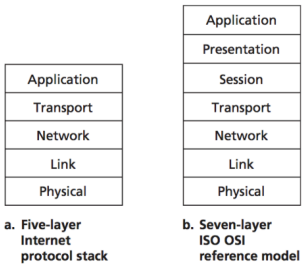
El Hardware soluciona la mitad de la tarea. Las redes se organizan en base a un modelo de **capas.** Estas, se comunican usando protocolos.

- **Servicio:** Conjunto de primitivas que provee una funcionalidad.
- **Capas:** Proveen servicios a otras capas.

El protocolo puede ser cambiado sin afectar al servicio.



Modelo Open System Interface (OSI): Modelo de referencia que define responsabilidades para cada capa. No define estándares, ni protocolos, ni implementaciones.



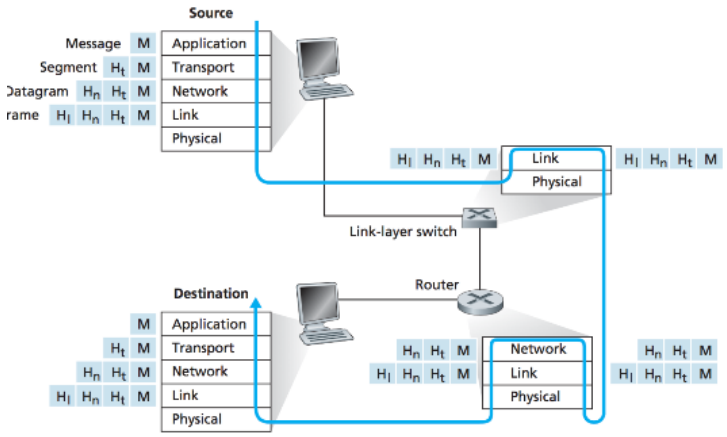
- **Aplicación:** Comunicación entre procesos (aplicaciones) mediante mensajes. Estas intercambian mensajes. HyperText Transfer Protocol (**HTTP**) | Simple Mail Transfer Protocol (**SMTP**)
  FIle Transfer Protocol (**FTP**) | Domain Name Service (**DNS**) | Internet Message Access Protocol (**IMAP**) | SSH, DHCP, POP, Telnet, XMPP, etc.

- **Presentación y Sesión:** Codificación, encriptación, serialización y luego establecimiento de sesión entre aplicaciones (autorización). De ser necesarias estas tareas se implementan en la aplicación. SOLO modelo OSI.

- **Transporte:** Transporte de mensajes entre **origen** y **destino**. Mensaje de capa de aplicación se transmite en **segmentos.** Transport Control Protocol (**TCP**) [Servicio *connection-oriented***,** garantías de entrega, control de flujo, control de gestión] | User Datagram Protocolo (**UDP**) [Servicio *connection-less,* sin garantías de entrega].

- **Red (Network):** Transporte de segmentos entre par de **hosts.** Los segmentos se transmiten en forma de **datagramas (paquetes).** Servicio de entrega desde origen a destino. Este último se busca en la red. Se genera un **encaminamiento (routing)** de segmento. Internet Protocol (**IP**) [Implementado en origen, destino y nodos intermedios, su hardware es el router] | Internet Control Message Protocolo (**ICMP**) [mensajes de control] | IGMP, OSPF [protocolos entre routers].

- **Enlace (Link) :** Transmisión de paquetes en cada **enlace**. Se transmiten en cada par de **hosts** de acuerdo a las características del enlace. Los paquetes de enlace se conocen como **frames.** Son confiables (reliability). Pueden proveer control de errores. Medium Access Control (**MAC**) [control de acceso al medio] | Ethernet | Wi-Fi | Hardware: switch, hub
- **Física:** Transmisión de frames en cada **enlace.** Transmisión de bits a través de un medio UTP (par trenzado de cobre) | Cable coaxial | Fibra Óptica | Medio inalámbrico



Capa de Aplicación

Arquitecturas:

- **Cliente-servidor:** Aplicaciones se construyen a través de procesos en distintos nodos que se comunican.
  - Los clientes se comunican con el servidor, nunca entre ellos.
  - El servidor posee una dirección fija y conocida. Este, además puede recibir múltiples solicitudes distintas.
  - Web | FTP | Telnet | SSH | email

- **Peer-to-Peer (P2P):** Aplicaciones se construyen en base a comunicación directa entre miembros (pares).
  - Facilita la auto-escalabilidad ya que los miembros pueden añadir capacidades a la red.
  - BitTorrent | Download accelerators | telefonía por internet (Skype) | IPTV
  - Las desventajas son que muchas conexiones son asimétricas (ISP Friendly) y tiene múltiples puntos de vulnerabilidad.

Socket (enchufe): Elemento de software que interactúa con la red. Es una interfaz para que el programador vea la red. Es un método para la comunicación entre un programa del cliente y un programa del servidor en una red. Es el punto final en una conexión.

- Es una puerta de entrada/salida al sistema
- Utiliza servicios de transporte (una capa más abajo).
- Solo existe en hosts de origen destino.
- Programador puede controlar parámetros del socket [Tipo de servicio (TCP/UDP) | Direcionamiento (IP + Puerto).
- Las aplicaciones imponen distintos requisitos al servicio de transporte.

Cuando tanto el cliente como el servidor poseen un socket, es posible mandar mensajes.

Se requiere definir:

- Tipo de servicio de transporte (UDP/TCP)
- Dirección del extremo opuesto (Dirección IP remota/Puerto Remoto)

La conexión TCP requiere "**Handshake"**, esto se refiere a que es necesario establecer conexión antes de enviar los datos. Luego, el servidor inicia una solicitud de conexión en un puerto conocido **del** servidor. Luego este debe tener un socket en ese puerto para luego aceptar la conexión y asignar un nuevo socket para la conexión.

HyperText Transfer Protocolo (HTTP): Cliente solicita y recibe archivos | Servidor recibe solicitudes y envía archivos.

Cookie: Archivo creado por un sitio web que contiene pequeñas cantidades de datos y que se envían entre emisor y receptor. El propósito es identificar al usuario almacenando su historial de actividad en un sitio web específico, de manera que luego se le pueda ofrecer el contenido más apropiado según sus hábitos. ¿Cuándo fue la última vez que un usuario entró a tal página?
Las cookies se almacenan en el cliente, luego estas se envían en conexiones posteriores. Luego el servidor usas las cookies para recordar el estado del cliente.

Proxy: Punto intermedio entre un ordenador conectado a internet y el servidor que está accediendo. Las solicitudes no se hacen directamente al servidor sino que al proxy. El **proxy web** es un dispositivo que posee una caché con la copia de las páginas web que se van visitando.

Content Delivery Network (CDN)[Red de Distribución de Contenidos]: Sistema de servidores distribuidos (red) que entregan páginas web y otros contenidos (audio y video) basados en la ubicación geográfica del usuario.

File Transfer Protocolo (FTP): Transferencia de archivos entre sistema local y remoto.

Domain Name System (DNS): Antes de iniciar la conexión TCP para el protocolo HTTP a URL. Utiliza una estructura jerárquica de dominios y servidores. Tecnología basada en una base de datos que sirve para **resolver nombres** en las redes, es decir, conocer la dirección IP de la máquina de donde está alojado el dominio al que queremos acceder.
3 partes: Cliente DNS | Servidor DNS | Zonas de autoridad
Al ingresar a una página lo que hacemos es hacer una petición al servidor DNS para que este nos retorne el IP de la página. Este si no tiene almacenado en su caché el IP de la página hace una petición al servidor encargado de la zona de autoridad. Este posee una tabla de datos en el que puede encontrar el IP, lo retorna y comienza la comunicación.
Cada sistema DNS almacena lo que se conoce como **mappings**, si no lo conoce lo solicita más arriba en la jerarquía. Se puede llegar a la raíz, las que existen 13 - de la A a la M- Para hacer consultas más altas en la jerarquía se hace a partir de recursividad o iteratividad.

Dominio: Nombre que identifica una máquina concreta.
Top Level Domain (TLD): Dominio de nivel superior -> TLDN (name) sería *com, net, cl, info, es.*
Tipos de DNS (nombre, valor, tipo, ttl) => A (nombre del host, dirección IP, A) | NS (dominio, IP host autoritario, NS) | CNAME (alias, nombre real, CNAME), MX (nombre, nombre de un mail server, MX).

Peer-To-Peer (P2P): No dependen de un servidor, tienen mejor escalabilidad y deben manejar pertenencia y abandono de miembros. **-> Distributed Hash Table:** Cada miembro posee una identificación única. Cada miembro se encarga de enrutar mensajes.

Capa de Transporte

Emisor: recibe un mensaje de capa de aplicación, lo divide en segmentos y solicita a la capa de red que envía los segmentos al receptor.
Receptor: Recibe segmentos, los ensambla para formar el mensaje y lo entrega a la capa de aplicación.
Dos protocolos -> **TCP** [Entrega confiable y ordenada. Control de flujo. Establecimiento de conexión]
-> **UDP** [Entrega no confiable y desordenada. Mejor esfuerzo]
Para identificar a qué proceso va cada mensaje - los que contienen segmentos - se utilizan los **puertos**, los que permiten identificar procesos dentro del nodo emisor o del nodo receptor. Este además puede **multiplexar** los mensajes que recibe.
El protocolo UDP demultiplexa sin conexión *<IP destino, puerto destino>* a partir de que el receptor identifica el puerto destino y pasa el segmento al proceso asociado al puerto destino.
En cambio, el protocolo TCP que necesita conexión *<IP Origen, puerto origen, IP destino, puerto destino>* se puede gestionar a partir de los servidores que son capaces de enviar solicitudes con múltiples conexiones. Bajo este protocolo se pueden asociar distintos sockets a distintos clientes.

User Datagram Protocol (UDP)

- Los segmentos pueden perderse, pueden llegar en distinto orden y se gestionan de manera independiente.
- No se establece conexión previa y no mantiene estado de conexión.
- A partir de este protocolo la aplicación puede controlar directamente el envío.
- No se demora en establecer conexión.
- No se reservan recursos.
- Menos overhead de tiempo y espacio.

Reliable Data Transfer (RDT): Protocolo que utiliza TCP para asegurar el envío de todos los paquetes y que sea de forma segura. Proveer transferencia confiable (reliable) sobre un medio no confiable (unreliable).
- Versión 1.0 Canal de transmisión confiable:
  - [Sender] Al recibir solicitud de envío, crea un paquete y envía.
  - [Reciever] Al recibir un paquete, extrae mensaje y lo pasa a aplicación.
  - **Problema**: Los paquetes pueden tener errores

- Versión 2.0 Protocolo Automatic Repeat reQuest (ARQ): Detección de errores y solicita retransmisión. Esto lo hace a partir de *acknowledgment (*ACK) y negative ACK (NAK).
- **Problema:** El sender no sabe si el reciever recibió correctamente o no.
- Versión 2.1 Agregar un **sequence number** que indica si se recibió o no.
- Versión 2.2 Eliminamos los NAK ya que cada ACK adjunta el sequence number, y solamente se continúa si este es 1.
- **Problema:** Se pueden perder paquetes.
- Versión 3.0 Timeout: para rentransmisión en el sender. Si no hay respuesta del receptor bajo una cierta cantidad de tiempo, se reenvía el paquete de datos.

Los protocolos se pueden mandar -> 1 a la vez.

-> enviar varios paquetes simultáneos en modo **pipelined**.

Para esto se necesita implementar:
- Números de secuencia incrementales.
- Tanto emisor como receptor necesitan *buffers.*

2 Enfoques:
- **Go-Back-N**
  - Emisor puede mantener hasta N paquetes sin ACK.
  - Receptor envía ACK luego de recibir el último paquete correctamente.
  - Emisor usa un timer para paquete más antiguo sin ACK, si este expira, se retransmite todo el grupo de paquetes sin ACK.
  -
- **Selective Repeat**
  - Emisor puede mantener hasta N paquetes sin ACK.
  - Receptor envía ACK para cada paquete individual.
  - Emisor usa un timer para paquete más antiguo sin ACK, si este expira, se retransmitesolo paquetes sin ACK.

Transport Control Protocolo (TCP)
- Protocolo de transmisión fiable.
- La transmisión es encadenada.
- Se hace un control de flujo a partir del tamaño de la ventana.
- Existen tanto buffers en emisor como en receptor.
- Existe un protocolo de establecimiento de conexión **“handshake”** antes de enviar paquetes de datos.
- Maximum Segment Size (**MSS**).
- ACK contiene el *sequence number* del siguiente byte que se espera. **Cumulative ACK’s.**
- Mensaje de respuesta también funciona como **ACK -> Piggybacked ACK.**

Retransmisión Rápida: Timeouts pueden tomar mucho tiempo por lo que bajo este método se envían productivamente segmentos probablemente perdidos, antes de que ocurra timeout, es decir, se mantiene enviándolos hasta que ocurra el ACK.

TCP Handshake Protocol:
1º Cliente envía segmento SYN al servidor
2º Servidor recibe SYN, y responde con SYN ACK
3º Cliente recibe SYN ACK, y responde con ACK

Capa de Red
Transmite a partir de paquetes o datagramas desde un emisor a un receptor. Existen dos problemas principales:
-**Direccionamiento**: Determinar la ubicación del nodo destino en la red. Acá cada host debe poseer una dirección única.
-**Enrutamiento:** Coordinar *routers* para que el paquete llegue al destino. Existen algoritmos para determinar rutas. Estas deben ser eficientes.

Router: Dispositivo de almacenamiento y reenvío: **store-and-forward**.
- **Forwarding:** Decisión local de un router. Cada router determina el próximo camino de un paquete mirando su *dirección destino*. Utilizan forwarding tables.
- **Enrutamiento**: Decisión distribuida entre routers. Con un algoritmo de routing llenan las tablas de forwarding.

Modos de conexión:

- **Virtual Circuit Network:** Esquema de red telefonica. Router mantiene estado de la conexión. Conexion-oriented. Garantiza ancho de banda durante la conexión [medida de datos y recursos de comunicación disponible o consumida expresada en bits/s]. Ancho de banda desperdiciado mientras no se transmitan datos. Poco flexible a congestiones.
- **Datagram Network:** Esquema de internet. Cada paquete lleva la dirección de destino. Routers reenvían de acuerdo a su forwarding table. Router no saben de caminos ni estados. Connectionless. No garantiza ancho de banda. Flexible ante congestión. Permite ocupar de mejor manera ancho de banda.

Foardwarding Table: Posee prefijos que almacenan los destinos para cada dirección posible.
Un mensaje encuentra su destinatario.a partir del paquete IP.
Cada paquete puede tener diferentes Maximun Transfer Unit (MTU).

Dirección IP
IPv4 = 32 bit.
Dotted-decimal notation. Grupo de 8 bit (0 a 255).

Subredes: Conjunto de nodos conectados a un router.
Primeros x bit definen la sub red. Últimos 32 bit definen el **host** dentro de la subred.
Ejemplo 223.1.1.40/24
Máscara = 255.255.255.0 o /24
Subred = 223.1.1.0
Host = 223.1.1.40
Acá a partir de la máscara podemos ver a cuántos IP podemos llegar, esto se obtiene haciendo 32 - 24 = 8. Este será la cantidad de bits para host. Ahora el tamaño de la subred es = 2^8 = 356-2= 254 hosts máximo.
La dirección del broadcast es 223.1.1.255

Cada router posee una dirección IP en cada salida.
Switches no poseen dirección IP.
Cada grupo de hosts conectados a un router forma una subred.

0.0.0.0 Dirección del host actual. Solo sirve como origen.
127.0.0.1 Localhost
ISP asigna bloques de direcciones a organizaciones y publican sus bloques. Las organizaciones pueden crear nuevas subredes dentro de su bloque.
ICANN Internet Corporation for Assigned Names and Numbers: Asigna direcciones a “regional Internet registries”

Direcciones dinámicas: Dynamic Host Configuration Protocol (DHCP) -> protocolo plug-and-play
(C) **DHCP discover message -> (S) DHCP offer message -> (C) DHCP request message -> (S) DHCP ACK message.**

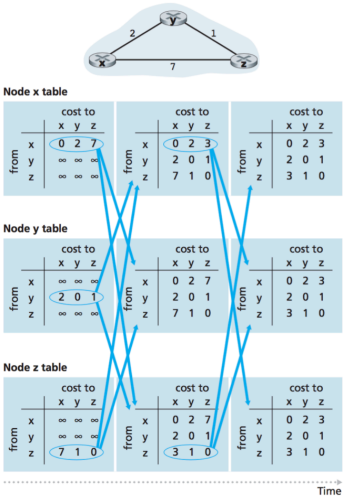
Network Address Translation (NAT): oficinas pequeñas y casas forman una LANs. Pero ISP provee cantidad limitada de IP’s. Este sistema permite multiplexar direcciones IP en redes privadas.

Internet Control Message Protocol (ICMP): Usado para mensajes de diagnóstico y control en capa de red. Se transmiten como *payload* en paquetes IP.

IPv6 = 64 bits.
Para poder recibir ambos tipos de IP se ocupan routers dual-stack.
Otra solución es **tunneling->** empaquetar los IPv6 a IPv4 cada vez que se pasa por IPv4.

Algoritmos de Routing: Conseguir que el paquete llegue desde un router de origen a un router de destino. Se modelan como un grafo. Los costos entre conexiones son ancho de banda, congestión, costo, distancia, etc.
Dos tipos:
-Centralizados: Nodos conocen la información completa de la red.
Link-State (LS): Cada nodo envía su info. De conectividad (vecinos y costos) a sus vecinos. Cuando todos tienen información de la topología, cada uno calcula las rutas más cortas -> Dijkstra.

-Decentralizados: Cálculo iterativo y distribuido entre nodos, sin conocer la topología completa.
Distance-Vector (DV): Se basa en ecuación de Bellman-Ford d(x,y) = min{c(x,v) + d(v,y)}. Cada nodo mantiene una tabla de la mejor ruta a sus vecinos. Además comparte su tabla con sus vecinos de tal forma que cuando ha sido propagado a todos los nodos, cada uno conoce las mejores rutas.
En este caso propagamos información de las rutas, mientras que en LS propagamos la información sobre los enlaces.



LInk State	Distance Vector
V nodos, E enlaces: O(VE) mensajes	Se ejecutan en paralelo y solo se propaga la ruta a los vecinos.
O(N^2) complejidad de cada nodo	Puede converger muy lentamente. Tiempo de convergencia indeterminado.
Ante errores, se propaga ausencia de enlace local	Ante errores, se propaga el error.

Routing a gran escala: No es factible almacenar tablas para todos los routers de internet. Internet se compone de **Sistemas Autónomos (AS)** interconectados.

Capa de Enlace

Frames, errores y control de flujo
La idea es transmitir frames a través de un enlace.
Servicios:
- Transforme un paquete en frames, y transmitirlo a través de UN enlace.
- Determinar quién puede usar el medio compartidor: **Médium Access Control (MAC)**
- Transferencia confiable en medios con alta tasa de errores.
- Detección y corrección de errores.
- **Control de flujo**

Detección de errores: Se agregan bits de redundancia que ayudan a detectar algunos errores.
-**Parity Check:** Errores de 1 bit.
-**Checksums:** Se suman para ver si calzan.
-**Cyclic Redundancy Check (CRC)**

MAC: Acceso al Medio
Problema de acceso al medio presente en medios compartidos (broadcast).
-**Time Division Multiple Access (TDMA)**: Acceso por turnos fijos.
-**Frequency-Divison Multiple Access (FDMA):** División por bandas de frecuencia.
-**Code Division Multiple Access (CDMA):** Permite que varios usuarios transmitan en la misma frequencia. Cad usuario posee un código de transmisión.
-**Random Access:** Cada nodo transmite cuando lo desea ocupando todo el ancho de banda disponible. Si dos transmiten al mismo tiempo se produce **colisión**. Debe existir un protocolo para las colisiones.
En caso de **colisiones: ALOHA particionado**, **ALOHA , CSMA**
- **Protocolos por turnos:**
  - **Master/slave**: master invita a los claves a transmitir (polling)
  - **Token:** Nodos se pasan un token. Solo el que tiene el token puede transmitir.

Hardware de Capa de Enlace: Switched LANs

Dirección MAC: Identificador único de 48 bits modificable por software. 24 bits para fabricante y 24 bit para NIC.

Address Resolution Protocol (ARP): Tabla ARP contiene asociaciones *<IP, MAC, TTL>*
Cada nodo (host, router, algunos switch) mantiene una tabla ARP.
TTL: Time-To-Live; india el tiempo que será recordada esa entrada.

¿Cómo comunicarse?
C quiere comunicarse con A, que no está en su tabla.
C envía mensaje **ARP Query** con IP de A y MAC.
Todos reciben el mensaje.
Solo A envía mensaje **ARP Response pacte** con su dirección MAC.
C guarda en su tabla ARP la asociación IP(A), MAC(A)

ETHERNET
Ampliamente usado para LAN: **coaxial | hub | switch**

L2 SWITCHES
Switch recibe frame y retransmite por el enlace apropiado, basado en **MAC**.
Filtering: determina si debe hacer **forward o drop.**
Forwoarding: Determina por cuál enlace transmitir.

Switch Table
Switch recibe frame en enlace x con destino MAC = D. Si no hay entrada para D, Broadcast para todos los enlaces, salvo x, si hay entrada para D con enlace x, descarta (**filtering).** Si hay entrada para D con enlace y != x, reenvía (**forwading).**

Los ciclos son malos para el broadcast por lo que se evitan con Minimum Spaning Tree's

Virtual LANS (VLAN)
Switches no proveen suficiente **traffic isolation.**