

# Capítulo 1: Introducción

ELO322: Redes de Computadores

Agustín J. González

Este material está basado en:

- El material preparado como apoyo al texto *Computer Networking: A Top Down Approach Featuring the Internet*.

# ¿Para qué o cuál es el objetivo de una red?

- ❑ Para compartir recursos
- ❑ Recursos:
  - Datos
  - Aplicaciones
  - Equipos
  - Ejemplos:
    - Cursos on-line (Massive Open Online Courses)
    - Películas on-line (datos)

# Introducción

1.1 ¿Qué es la Internet?

1.2 Red periférica

1.3 Red central (core)

1.4 Red de acceso y medios físicos

1.5 Estructura de Internet e ISPs

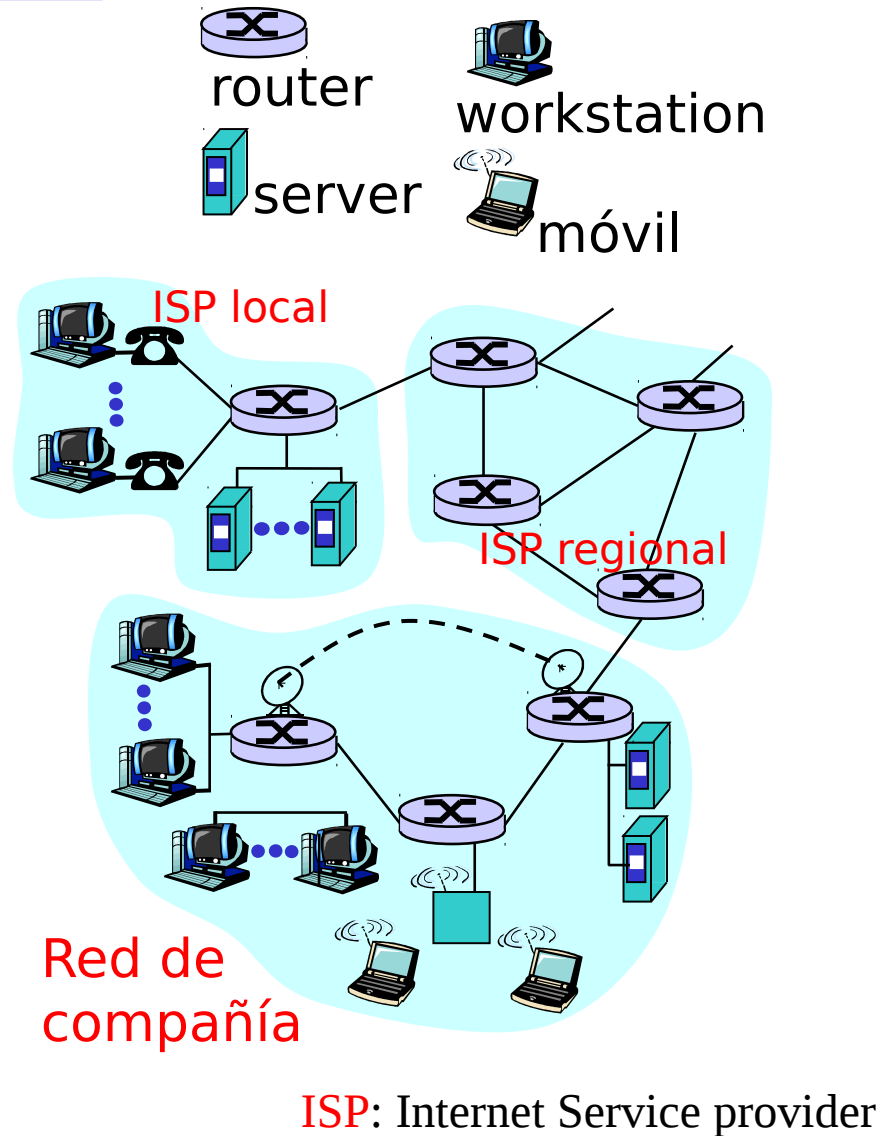
1.6 Retardos & pérdidas en redes de paquetes conmutados

1.7 Capas de protocolos, Modelo de servicio

1.8 Historia (lectura personal)

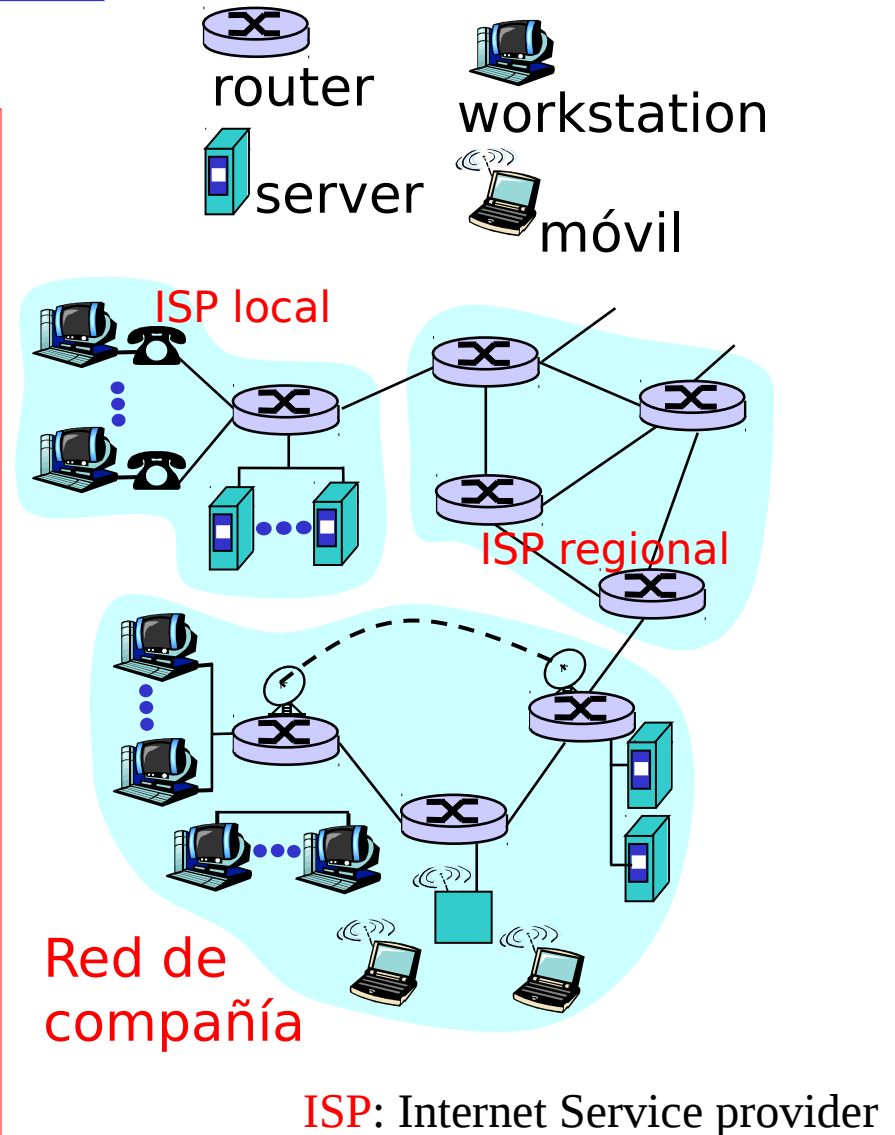
# ¿Qué es la Internet?

- ❑ Millones de dispositivos de cómputo conectados: *hosts = sistema terminal*
- ❑ Los hosts corren las *aplicaciones de red*
- ❑ *Conectados vía Enlaces de comunicaciones*
  - fibra, cobre, radio, satélite
  - Tasas de transmisión = *"ancho de banda (bandwidth)"*
- ❑ *routers*: re-envía paquetes (datos binarios)



# ¿Qué es la Internet?

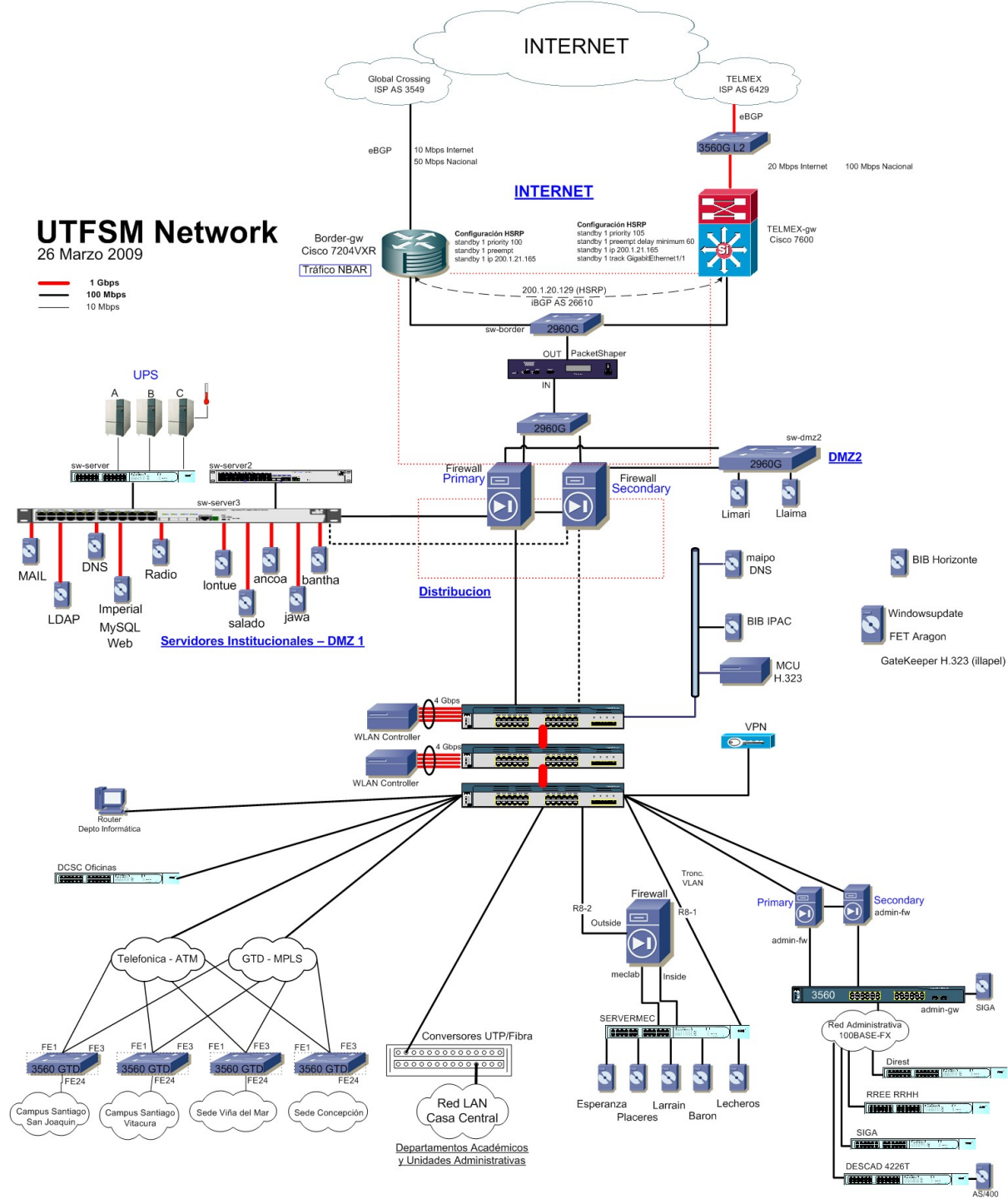
- ❑ *protocolos* controlan el envío, recepción de mensajes
  - e.g., TCP, IP, HTTP, FTP, PPP
- ❑ *Internet: “Red de redes”*
  - Débilmente jerárquica
  - Internet pública versus **intranet** privadas
- ❑ *Estándar en Internet*
  - RFC: Request for comments
  - IETF: Internet Engineering Task Force



# Red USM

Versión Marzo 2009

Ver:  
<http://www.dcs.c.utfsm.cl/>



# ¿Qué es la Internet?: sus servicios

- ❑ *Servicios de comunicación provistos a las aplicaciones*
  - Sin conexión, no confiable
  - Con conexión, confiable
- ❑ *Infraestructura de comunicación* permite aplicaciones distribuidas
  - e.g.: Web, email, juegos, e-commerce (comercio electrónico), peer-to-peer (Kazaa, eMule, WhatsApp), contenido (youtube, gmail, facebook)
- ❑ *El propósito de las redes* es el compartir recursos (datos, acceso a máquinas, etc).

# ¿Cómo se estructura y estudia las redes de Computadores?

- ❑ “Dividir para conquistar”
- ❑ La arquitectura se puede subdividir en capas.
- ❑ Capas de la arquitectura de Internet:

<b>Aplicación</b>
<b>Transporte</b>
<b>Red</b>
<b>Enlace de Datos</b>
<b>Física</b>



# ¿Qué es un protocolo?

## Protocolos humanos:

- ❑ “¿Qué hora es?”
- ❑ “Tengo una pregunta”

... mensaje específico es enviado

... acción específica es tomada cuando el mensaje u otros eventos son recibidos

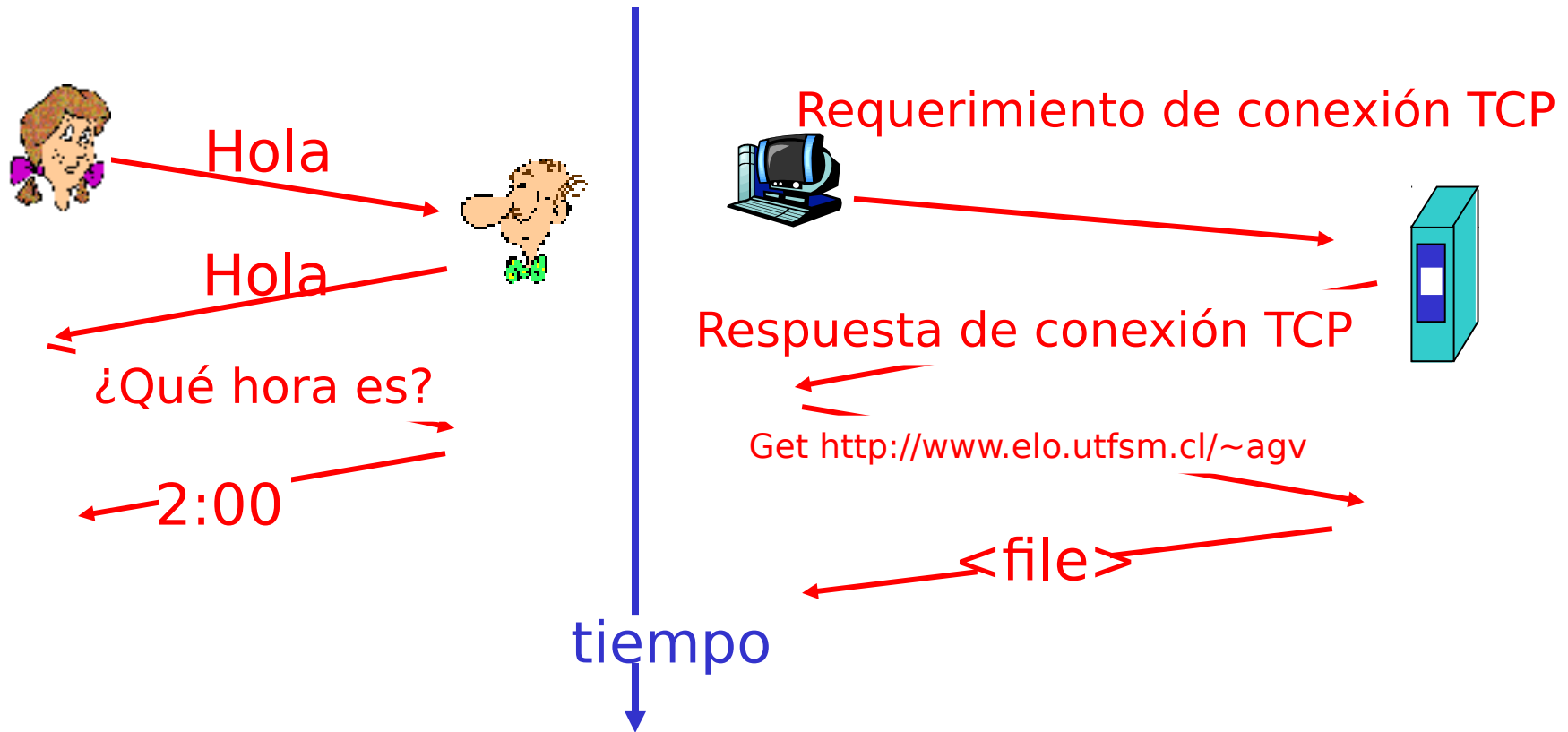
## Protocolos de red:

- ❑ Máquinas en lugar de humanos
- ❑ Todas las actividades de comunicación en Internet son gobernadas por protocolos

***Protocolo: conjunto de reglas que definen el formato y orden de mensajes enviados y recibidos entre entidades de la red, y las acciones tomadas al transmitir o recibir mensajes***

# ¿Qué es un protocolo?

Un protocolo humano y un protocolo en redes de computadores.



Q: Puede mencionar otros protocolos humanos?

# Introducción

1.1 ¿Qué es la Internet?

1.2 Red periférica

1.3 Red central (core)

1.4 Red de acceso y medios físicos

1.5 Estructura de Internet e ISPs

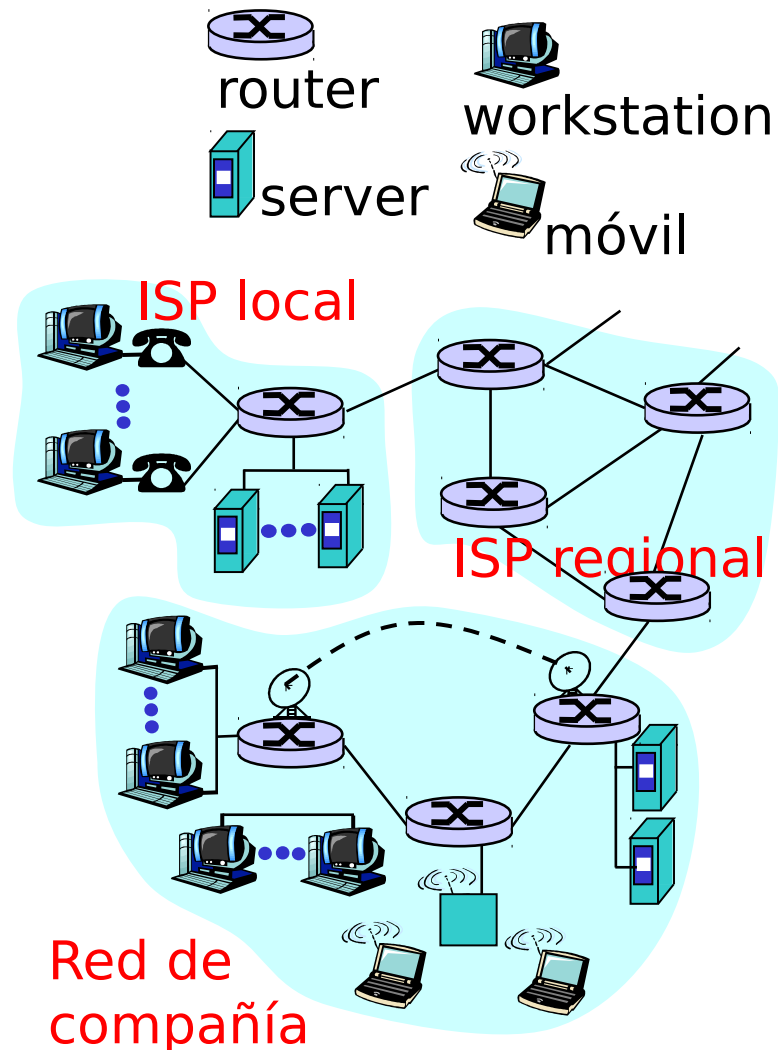
1.6 Retardos & pérdidas en redes de paquetes conmutados

1.7 Capas de protocolos, Modelo de servicio

1.8 Historia (lectura personal)

# Una mirada a la estructura de la red

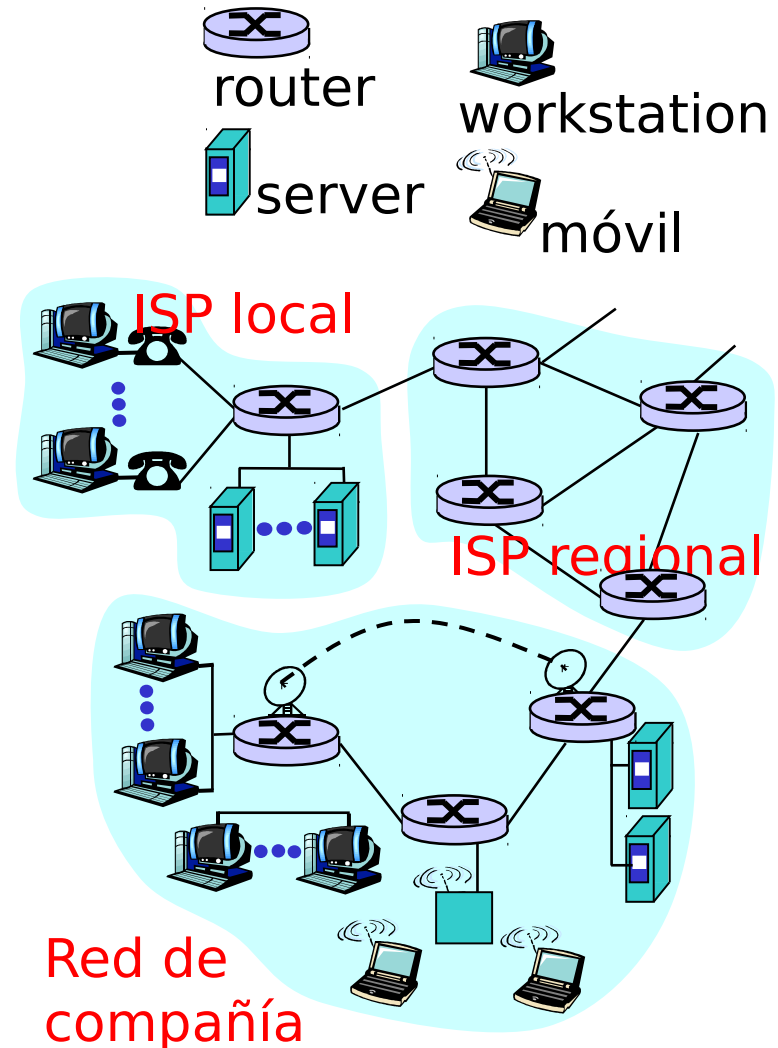
- ❑ **Red periférica (network edge):** aplicaciones y terminales (hosts)
- ❑ **Red Central (network core)**
  - Enrutadores (routers)
  - Red de redes
- ❑ **Redes de Acceso, medios de comunicación:** enlaces de comunicación (Ethernet, WiFi, 3G...)



Móvil: Laptop, Cellphone, etc

# Red Periférica (network edge)

- ❑ **Terminales (hosts):**
  - Corren programas/aplicaciones
  - E.g. Web, mail, chat
  - En la periferia de la red
- ❑ **Modelo cliente/servidor**
  - Terminales clientes piden servicios y los reciben de los servidores “always on”
- ❑ **Modelo peer-to-peer:**
  - Uso mínimo de servidores dedicados
  - E.g. Gnutella, KaZaA, otros
- ❑ **Modelos híbridos**
  - Mezcla de los dos previos



# Red periférica: servicio orientado a la conexión

- Objetivo: transferir datos entre sistemas terminales (hosts)
- ❑ *handshaking:* preparación para transferencia
    - Hola, hola en protocolos humanos
    - *Fija “estado”* en dos hosts comunicándose
    - **TCP - Transmission Control Protocol**

## Servicio TCP[RFC 793]

- ❑ *confiable, en orden,* transmisión de flujos de bytes
  - pérdidas: acuses de recibo y retransmisiones
- ❑ *Control de flujo:*
  - Transmisor no debe sobrecargar al receptor
- ❑ *Control de congestión:*
  - transmisor “baja tasa de envío” cuando la red está congestionada

# Red periférica: servicio sin conexión (UDP)

**Objetivo:** Igual que el previo! Transferencia de datos entre sistemas terminales (hosts)

## ❑ **UDP - User Datagram Protocol** [RFC 768]:

- Sin conexión
- Transferencia no confiable de datos
- Sin control de flujo
- Sin control de congestión

## Aplicaciones que usan TCP:

- ❑ HTTP (Web), SSH (Secure Shell), SMTP (email)

## Aplicaciones que usan UDP:

- ❑ streaming media, teleconferencia, DNS, Telefonía en Internet (la voz)

# Introducción

1.1 ¿Qué es la Internet?

1.2 Red periférica

1.3 Red central (core)

1.4 Red de acceso y medios físicos

1.5 Estructura de Internet e ISPs

1.6 Retardos & pérdidas en redes de paquetes conmutados

1.7 Capas de protocolos, Modelo de servicio

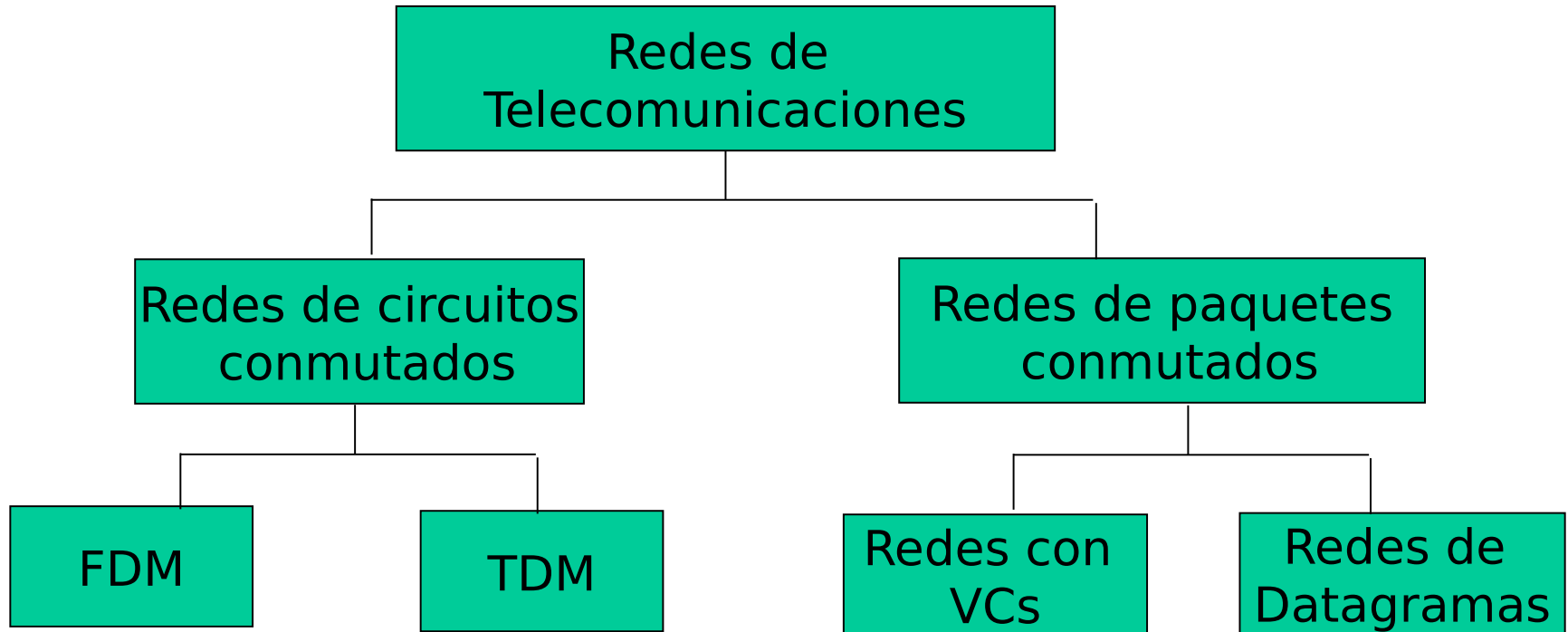
1.8 Historia (lectura personal)



# Red interna (network core)

- ❑ Malla de routers interconectados
- ❑ La pregunta fundamental: ¿Cómo se transfieren datos a través de las redes?
  - **Conmutación de circuitos (circuit Switching)**:  
Un circuito dedicado por cada “llamada”  
(e.g. red telefónica)
  - 
  - **Conmutación de paquetes (packet switching)**: datos enviados a través de la red en bloques discretos

# Taxonomía de redes



En Internet las aplicaciones envían paquetes.  
En su trayecto los paquetes pueden transitar por circuitos conmutados.

# Red interna: Conmutación de Circuitos

En este caso los recursos desde un terminal a otro son reservados al inicio de la llamada (transmisión de datos)

- ❑ Se reserva ancho de banda enlaces, capacidad en switches
- ❑ Los recursos reservados son dedicados, no compartidos.
- ❑ Capacidad garantizada
- ❑ Se requiere una configuración de la conexión (call setup) previa al envío.

# Red interna: Conmutación de circuitos

Recursos de la red  
(e.g., bandwidth)  
**dividido en  
“pedazos”**

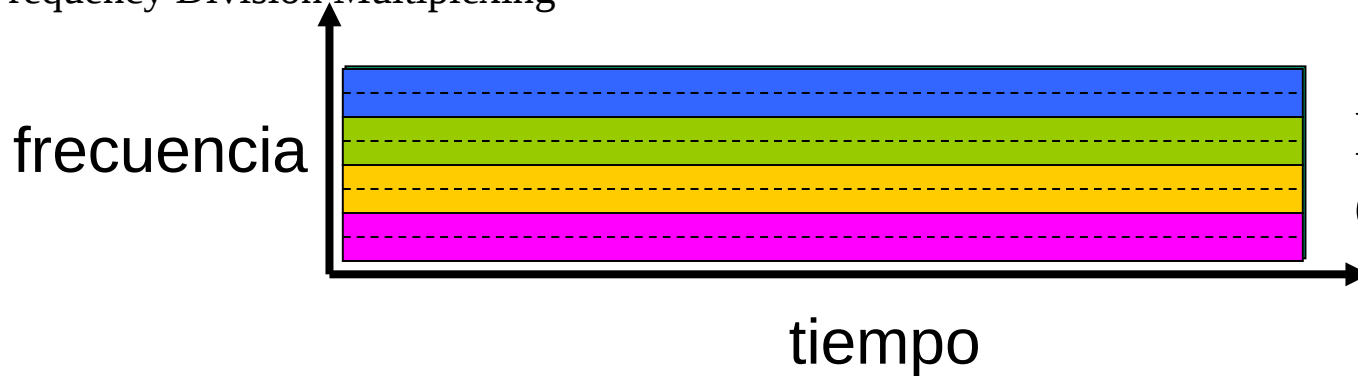
- ❑ Pedazos asignados a llamados
- ❑ Recurso es *inactivo (idle)* si no es usado por el dueño de la llamada (*no se comparte*)

- ❑ Dos formas para dividir los recursos en “pedazos”
  - División en frecuencia FDM (Frequency Division Multiplexing)
  - División en tiempo TDM (Time Division Multiplexing)

# Conmutación de circuitos: FDM y TDM

FDM (En redes ópticas: WDM)

Frequency Division Multiplexing



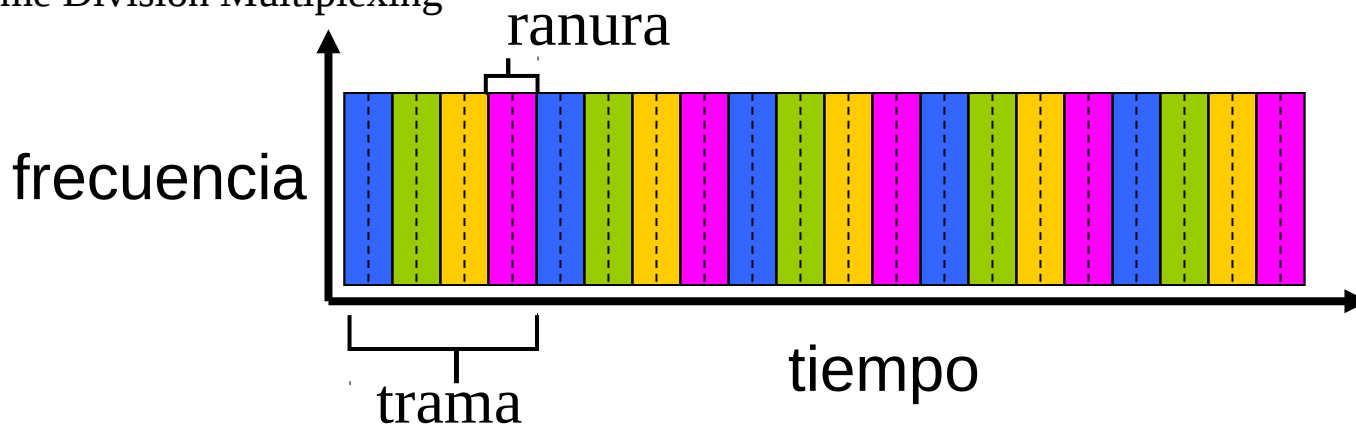
Ejemplo:

4 usuarios 

Ej. Radiodifusoras  
Canales en WiFi

TDM

Time Division Multiplexing



Ej.

\* Sala de clases  
\* Redes ópticas:  
SONET, SDH

# Ejemplo numérico

- ¿Cuánto tiempo toma enviar un archivo de 640.000 bits desde host A a host B por una red conmutada por circuitos?
  - Todos los enlaces desde A a B son de 1,536 Mbps
  - Cada enlace usa TDM con 24 ranuras
  - 500 msec son requeridos para establecer el circuito extremo a extremo
  - Suponga que no hay overhead (todos los bits del enlace transportan información)
  - Estimación, pues se desconoce qué ranura y su tamaño

**Resolver!**

# Red interna: Conmutación de paquetes

Cada flujo de datos extremo a extremo es dividido en *paquetes*

- ❑ Paquetes de usuarios A, B *comparten* los recursos de la red
- ❑ Cada paquete usa el bandwidth total.
- ❑ Recursos son usados según *son necesarios*

División del Bandwidth en "pedazos"

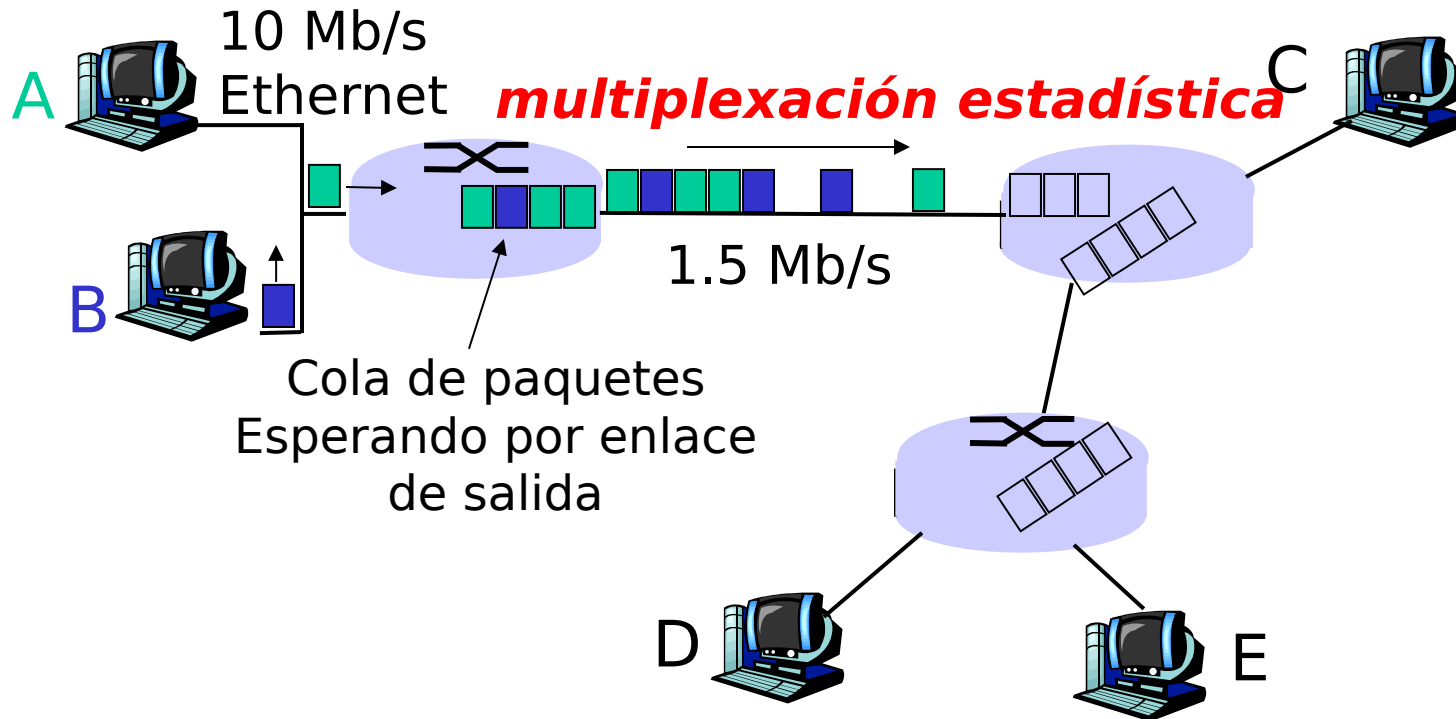
Asignación dedicada

Reservación de recursos

Contención de recursos:

- ❑ Demanda acumulada de recursos puede exceder cantidad disponible
- ❑ congestión: encolar paquetes, esperar por uso del enlace
- ❑ Almacenamiento y re- envío (store and forward): paquetes se mueven un tramo por vez
  - Nodo recibe paquetes completos antes de re- enviarlos

# Conmutación de Paquetes: Multiplexación Estadística



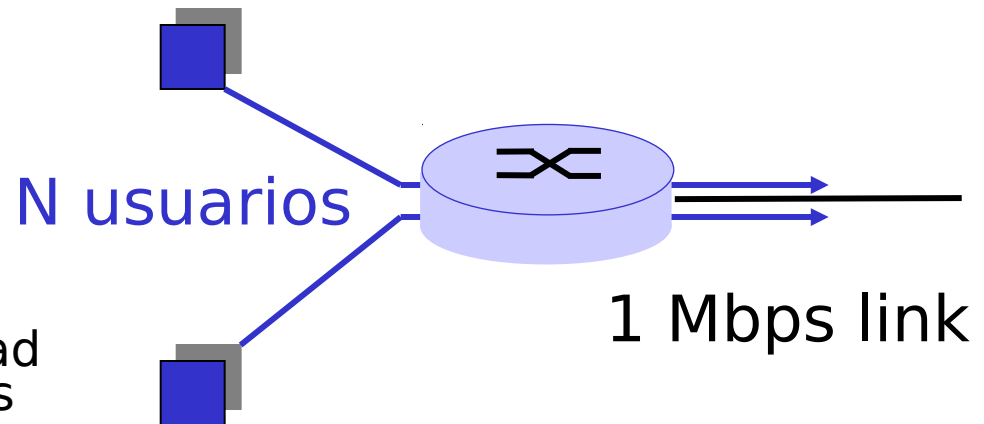
- ❑ Secuencias de paquetes de A y B no tienen patrón fijo ➔ ***multiplexación estadística***.
- ❑ Distinto a TDM donde cada host obtiene la misma ranura en la trama TDM.



# Conmutación de Paquetes versus conmutación de circuitos

Conmutación de paquetes permite que más usuarios usen la red!

- ❑ Enlace de 1 Mb/s
- ❑ Escenario: Cada usuario:
  - Usa 100 kb/s cuando están “activos”
  - Está activos 10% del tiempo
- ❑ Conmutación de circuitos:
  - 10 usuarios
- ❑ Conmutación de paquetes:
  - con 35 usuarios, la probabilidad de tener más de 10 activos es menor que .0004
  - Equivale a calcular la probabilidad de obtener más de 10 caras al lanzar 35 “monedas” donde cada “moneda” resulta cara con  $P=0.1$



# Conmutación de paquetes versus conmutación de circuitos

¿Es la conmutación de paquetes un ganador?

Packet switching

- ❑ Excelente para datos en ráfagas (de corta duración)
  - Se comparten los recursos
  - Más simple, no requiere establecimiento de llamado.
- ❑ Ante excesiva congestión: retardo de paquetes y pérdidas
  - Protocolos necesarios para transferencia de datos confiable y control de congestión
- ❑ Q: ¿Cómo proveer comportamiento tipo circuito?
  - bandwidth garantizado requerido en aplicaciones de audio y video
  - Aún un problema no resuelto (más adelante en el curso)

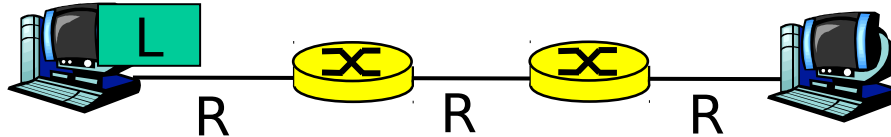
# Pregunta tipo certamen

En conmutación de paquetes mencione una ventaja de usar paquetes de tamaño grande y una ventaja de usar tamaño de paquete pequeño.

**Tamaño grande:** Como el encabezado es de tamaño fijo, el usar paquetes grandes aumenta la eficiencia de lo enviado: mayor proporción de bytes transportados corresponden a información.

**Tamaño pequeño:** Se reduce el retardo de las aplicaciones. No se requiere esperar acumular muchos datos para enviarlos.

# Conmutación de paquetes: almacenamiento y reenvío



- ❑ Demora  $L/R$  segundos transmitir (enviar) paquetes de  $L$  bits por el enlace de  $R$  bps
- ❑ El paquete completo llega al router antes que éste pueda ser transmitido sobre el próximo enlace: *store and forward*
- ❑ Retardo =  $3L/R$

## Ejemplo:

- ❑  $L = 7.5$  Mbits
- ❑  $R = 1.5$  Mbps
- ❑ retardo = 15 sec
- ❑ **OJO:** No se ha considerado tiempos de propagación ni de procesamiento en el conmutador.

# Redes de conmutación de paquetes: re- envío

- ❑ ***Objetivo:*** mover los paquetes a través de routers desde la fuente al destino
  - Estudiaremos varios algoritmos para seleccionar la ruta (más adelante: ruteo o enrutamiento)
- ❑ **Redes de datagramas:**
  - *Dirección de destino* en paquete determina próximo tramo
  - Las rutas pueden cambiar durante la sesión
  - analogía: conducción preguntando instrucciones
- ❑ **Redes de circuitos virtuales:**
  - Cada paquete lleva un rótulo (identificador del circuito, virtual circuit ID), el rótulo determina el próximo tramo
  - Camino fijo determinado *cuando se establece la llamada*, permanece fijo durante la llamada.
  - Analogía: Maratón con trazado definido.
  - *En este caso routers mantienen estado por cada llamada (= > mayor uso de memoria)*

# Pregunta tipo certamen

Liste los nombres de las capas de servicio del modelo TCP/IP ubicadas bajo la capa aplicación e indique cuál es la función de cada una de ellas.

**Transporte:** Llevar paquetes desde un proceso en la máquina origen a un proceso en la máquina destino.

**Red:** Rutear paquetes desde el computador origen al destino.

**Enlace:** Transferir paquetes desde un nodo a otro adyacente.

**Física:** Transferir bits a través de un enlace.