

Tema I
Introducción a los SSTT
Servicios de información y
aplicaciones en red

Servicios Telemáticos – 3º Grado de en Ingeniería Informática

Objetivos

- ▶ Introducir una **visión general** de la arquitectura de Internet
- ▶ Asentar los conceptos de **servicios, protocolos y procesos**

Referencias

- ▶ Capítulo 1.1-1.3, 1.5 y Capítulo 2.1
- ▶ J. F. Kurose, K.W. Ross. *Computer Networking, A top-down approach featuring the Internet.* 6th Ed.

¿Qué es Internet? (I)



PC



server



wireless laptop



smartphone

- ▶ Millones de dispositivos interconectados:

- ▶ *Hosts (anfitrión) = end systems*

- ▶ Ejecutan *aplicaciones de red*

- ▶ *Enlaces de comunicación*

- ▶ Fibra, cobre, radio, satélite

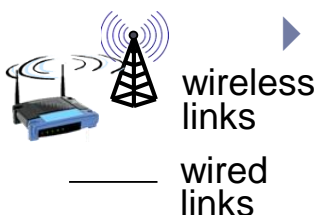
- ▶ Ratio de transmisión:

- ▶ *ancho de banda*

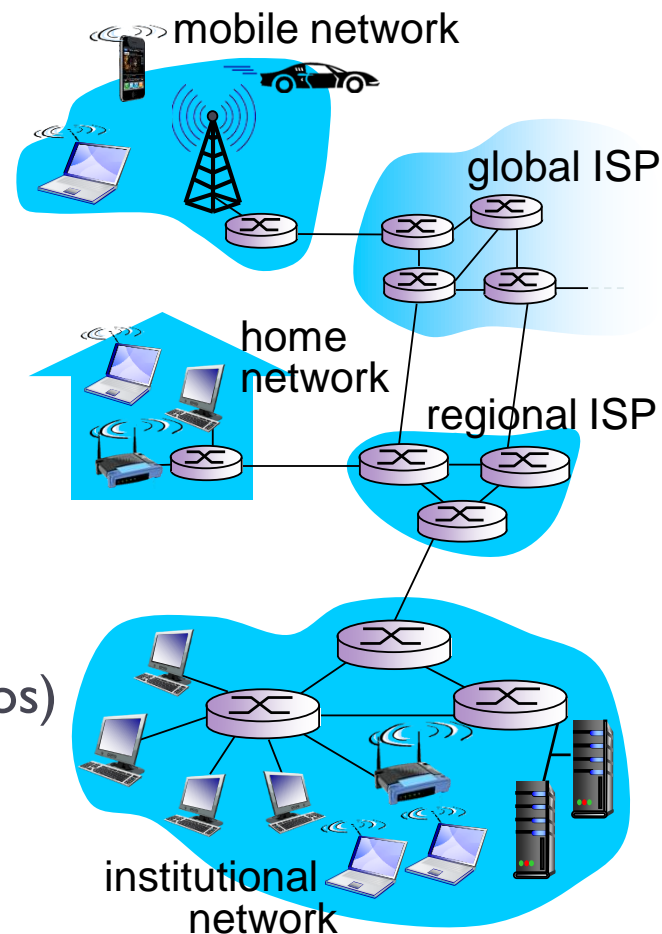
- ▶ *Conmutación de paquetes:*

- ▶ reenvía paquetes (bloques de datos)

- ▶ *routers y switches*

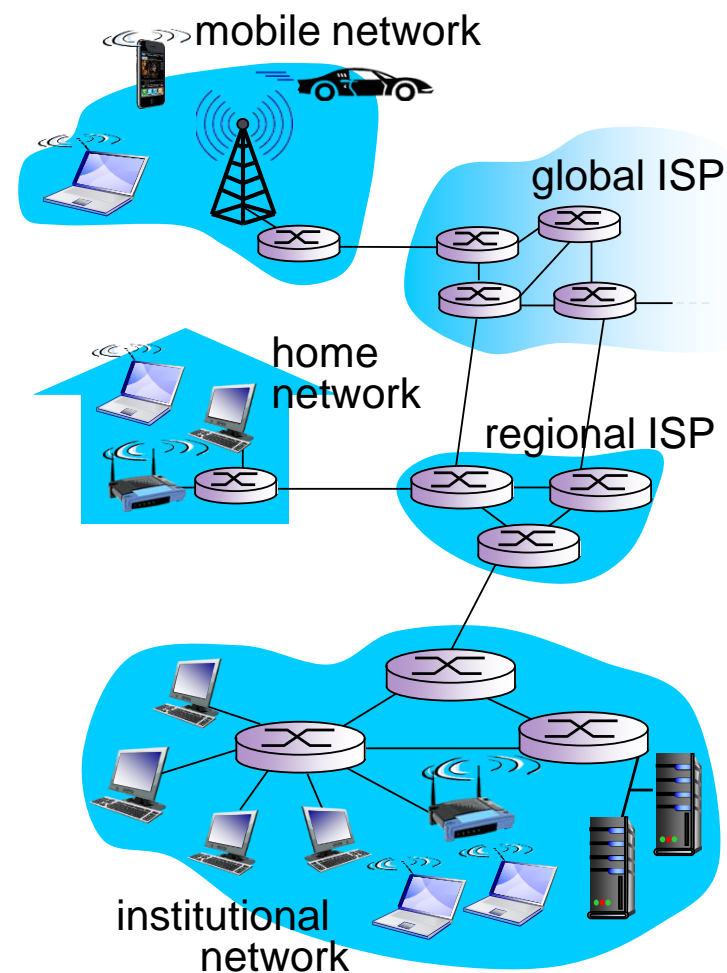


router



¿Qué es Internet? (II)

- ▶ *Internet: “red de redes”*
 - ▶ ISPs interconectados
- ▶ *protocolos* para enviar y recibir mensajes: p.ej., TCP, IP, HTTP, Skype, 802.11
- ▶ *Internet standards*
 - ▶ RFC: Request for comments
 - ▶ IETF: Internet Engineering Task Force



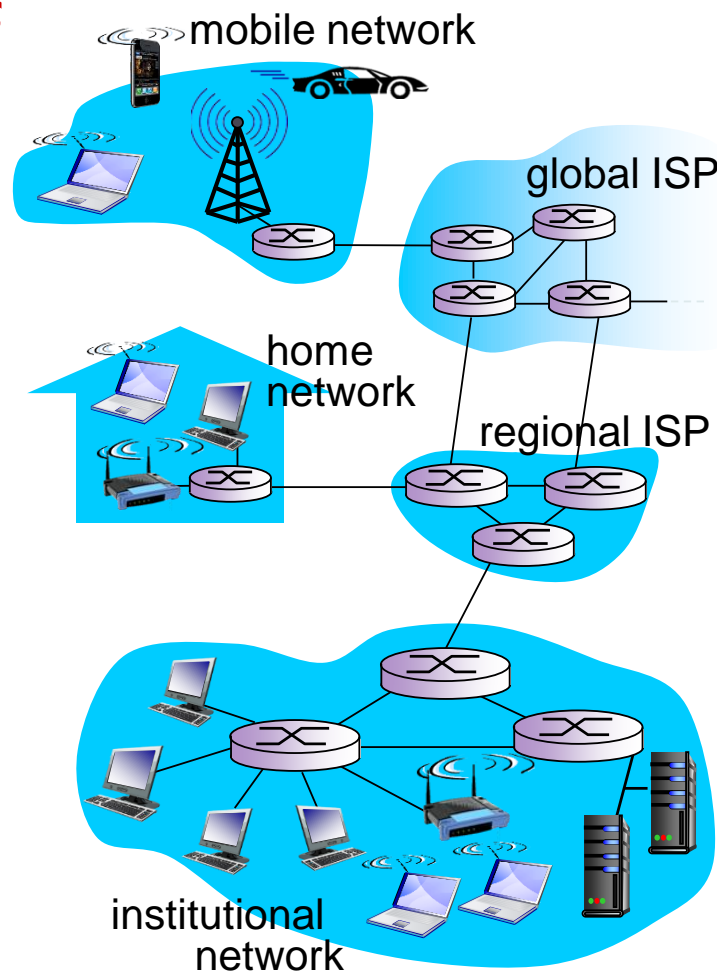
¿Qué es Internet?: Visto como un servicio

► *Infraestructura que ofrece servicios a aplicaciones:*

- Web, VoIP, email, games, e-commerce, social nets, ...

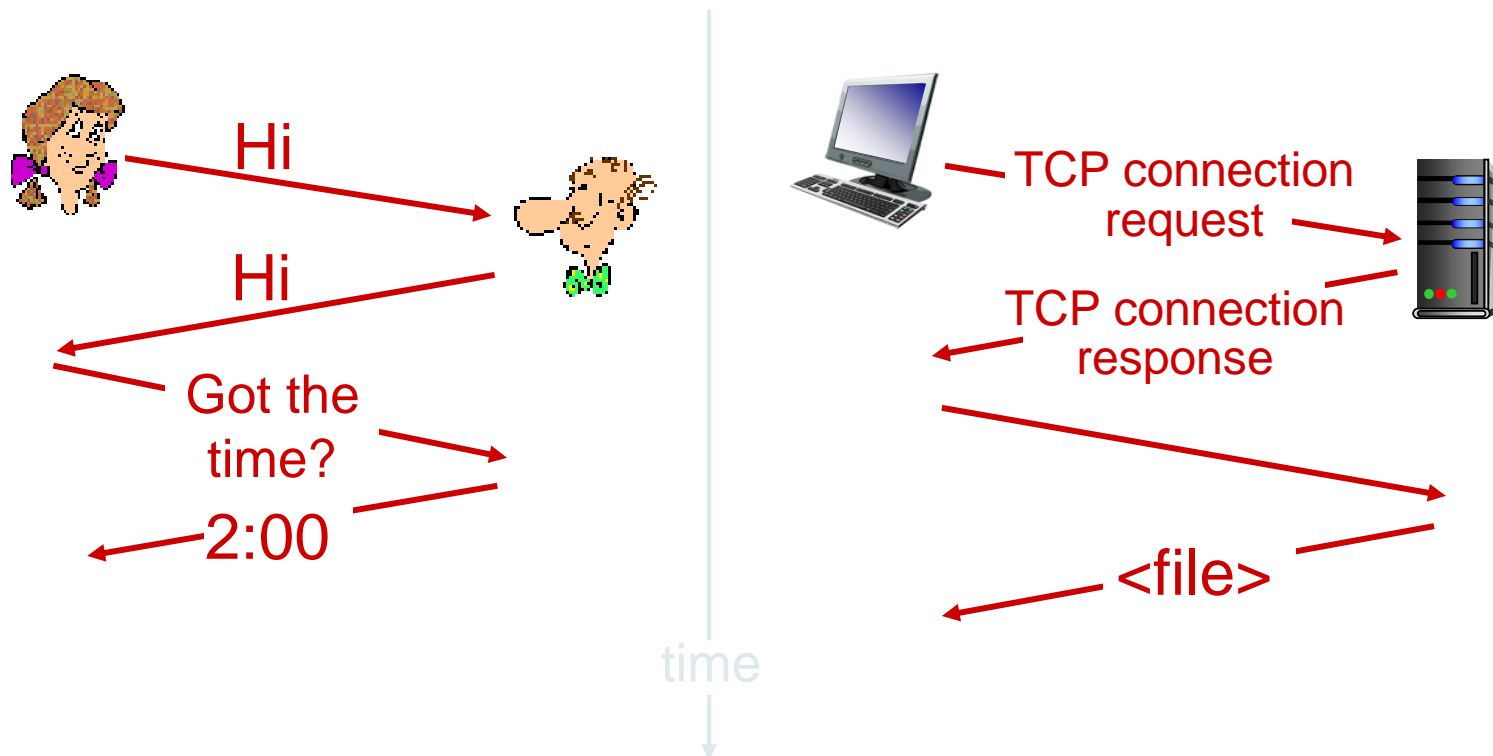
► *Ofrece interfaz de programación para apps*

- Permite conectar a las aplicaciones a Internet
- Ofrece opciones de servicios



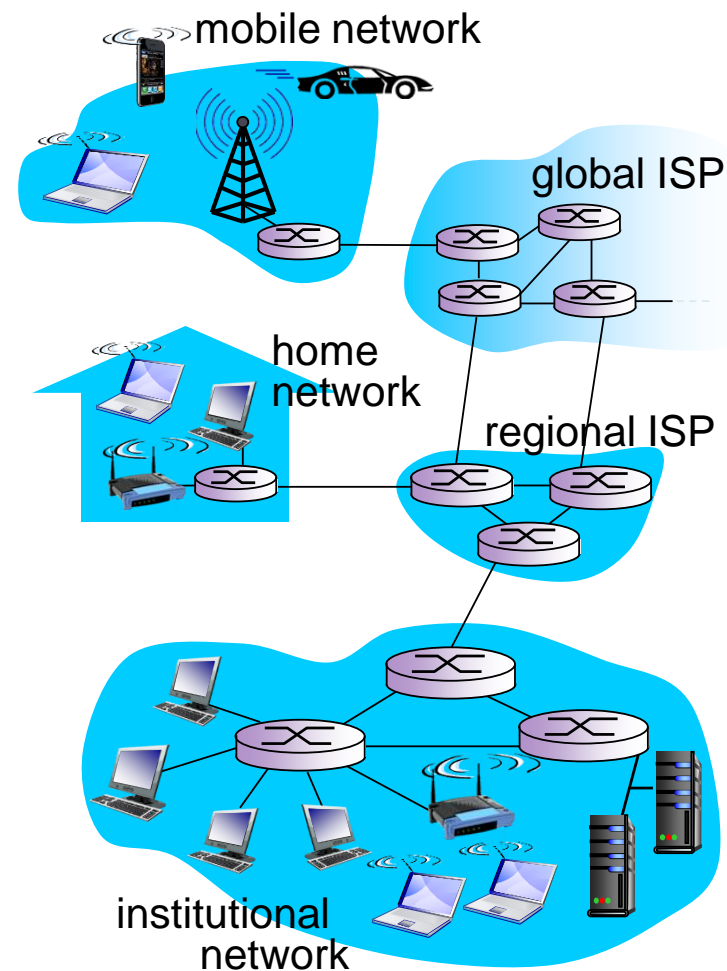
¿Qué es un protocolo?

Protocolo Humano vs Protocolo en Internet



Estructura de la red

- ▶ **Extremo de la red:**
 - ▶ hosts: clientes y servidores
 - ▶ Servidores habitualmente en centros de datos
- ▶ **Redes de acceso, medio físico:** enlaces de comunicación con cable cable o sin cable (wireless)
- ▶ **Núcleo de red:**
 - ▶ Routers interconectados
 - ▶ Red de redes



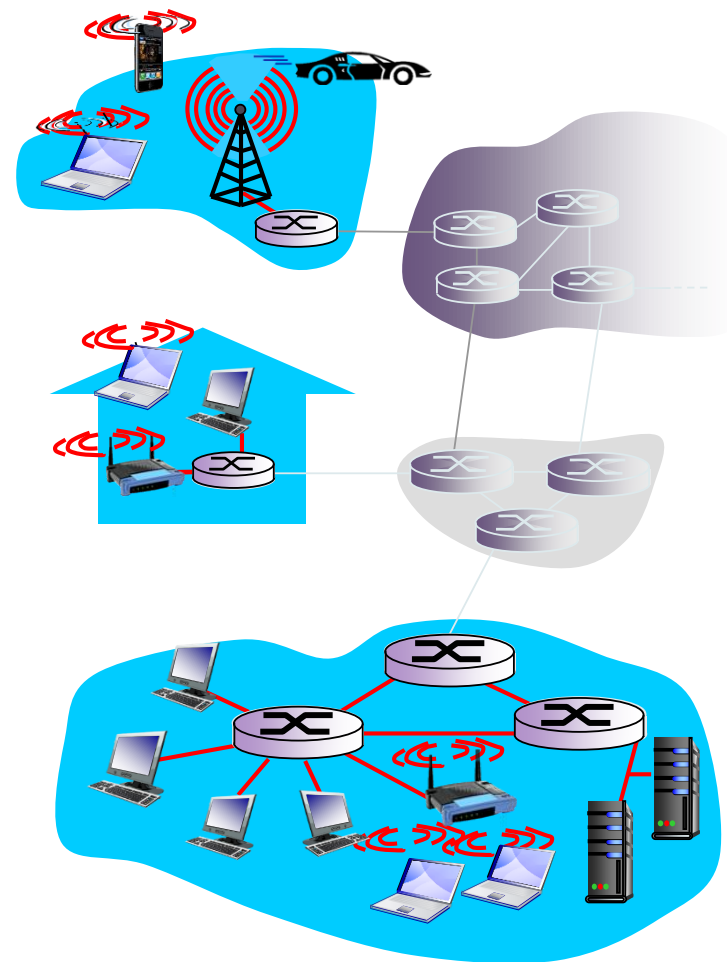
Redes de acceso y medio físico

Q: ¿Cómo conectar sistemas finales al edge router?

- ▶ Acceso residencial
- ▶ Acceso Institucional networks (school, company)
- ▶ Acceso a la red móvil

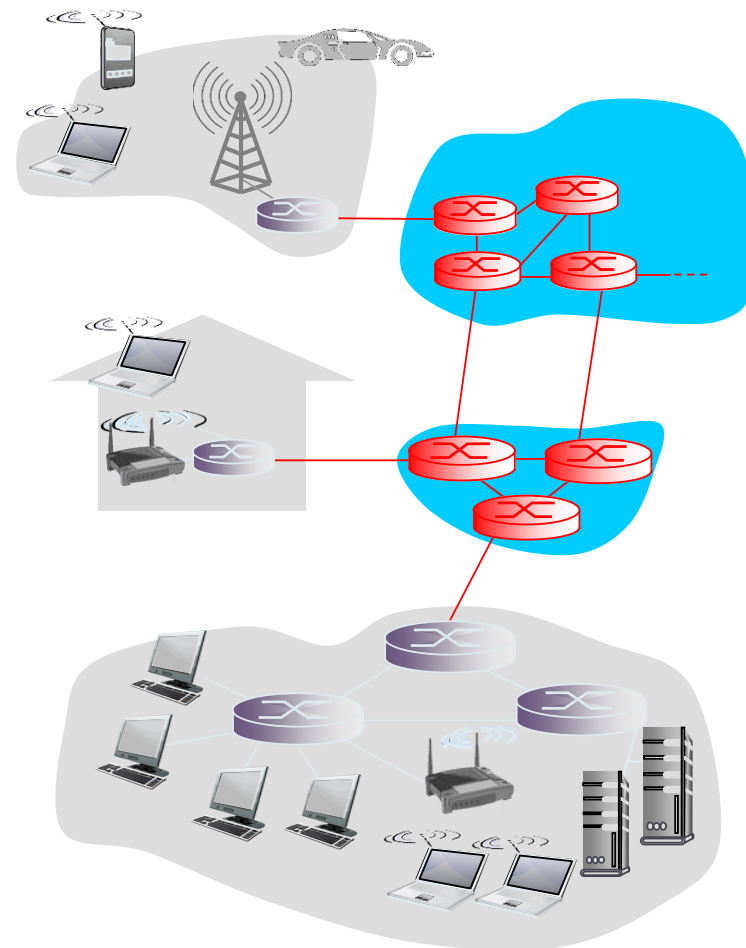
A tener en cuenta:

- ▶ Ancho de banda (bits por segundo) de la red de acceso
- ▶ ¿Compartido o dedicado?



El núcleo de la red

- ▶ Malla (mesh) de routers interconectados
- ▶ **Conmutación de paquetes: los hosts dividen mensajes del nivel de aplicación en paquetes más pequeños**
 - ▶ Los paquetes son enrutados de un router al siguiente a través de enlaces entre la fuente y el destino

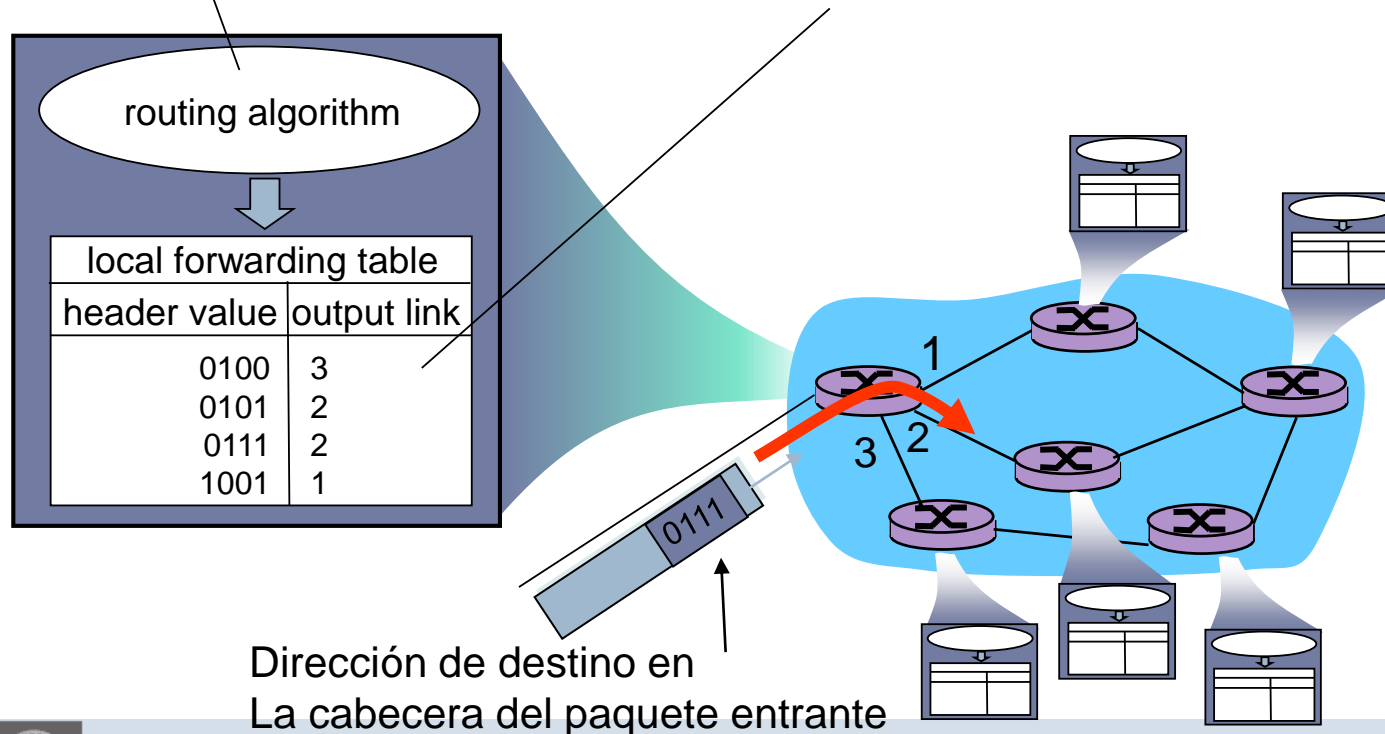


Dos funciones del núcleo de red

routing: determina en base a la fuente y el destino de los paquetes el siguiente salto

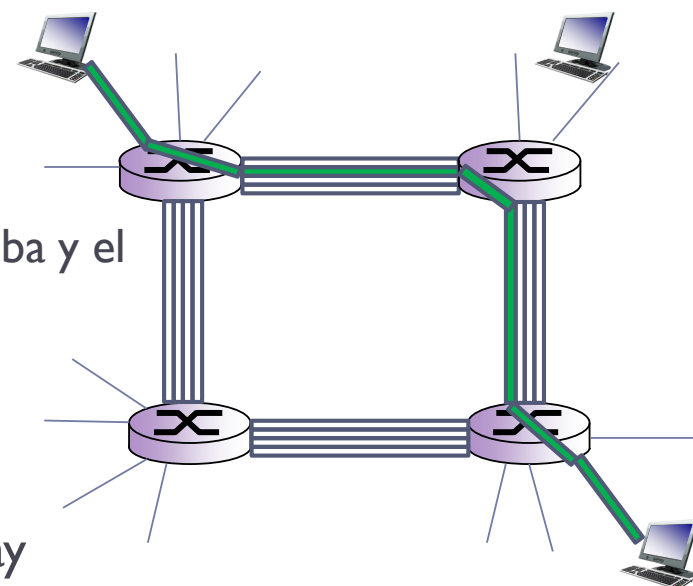
- *Algoritmos de enrutamiento*

forwarding: mueve paquetes de la entrada de un router a la interfaz de salida para el siguiente router



Alternativa: Conmutación de circuitos

- ▶ **Recursos end-end reservados para una “llamada” entre la fuente y el destino**
 - ▶ En el diagrama hay enlaces para 4 circuitos
 - ▶ La llamada obtiene el 2º circuito en el enlace de arriba y el 1º circuito en el enlace de la derecha.
- ▶ **Recursos dedicados: no se comparten**
 - ▶ Rendimiento (garantizado)
 - ▶ El segmento del circuito se mantiene idle si no hay llamada (*no se comparte*)
- ▶ Se usa en redes telefónicas



Com. de paquetes vs com. de circuitos

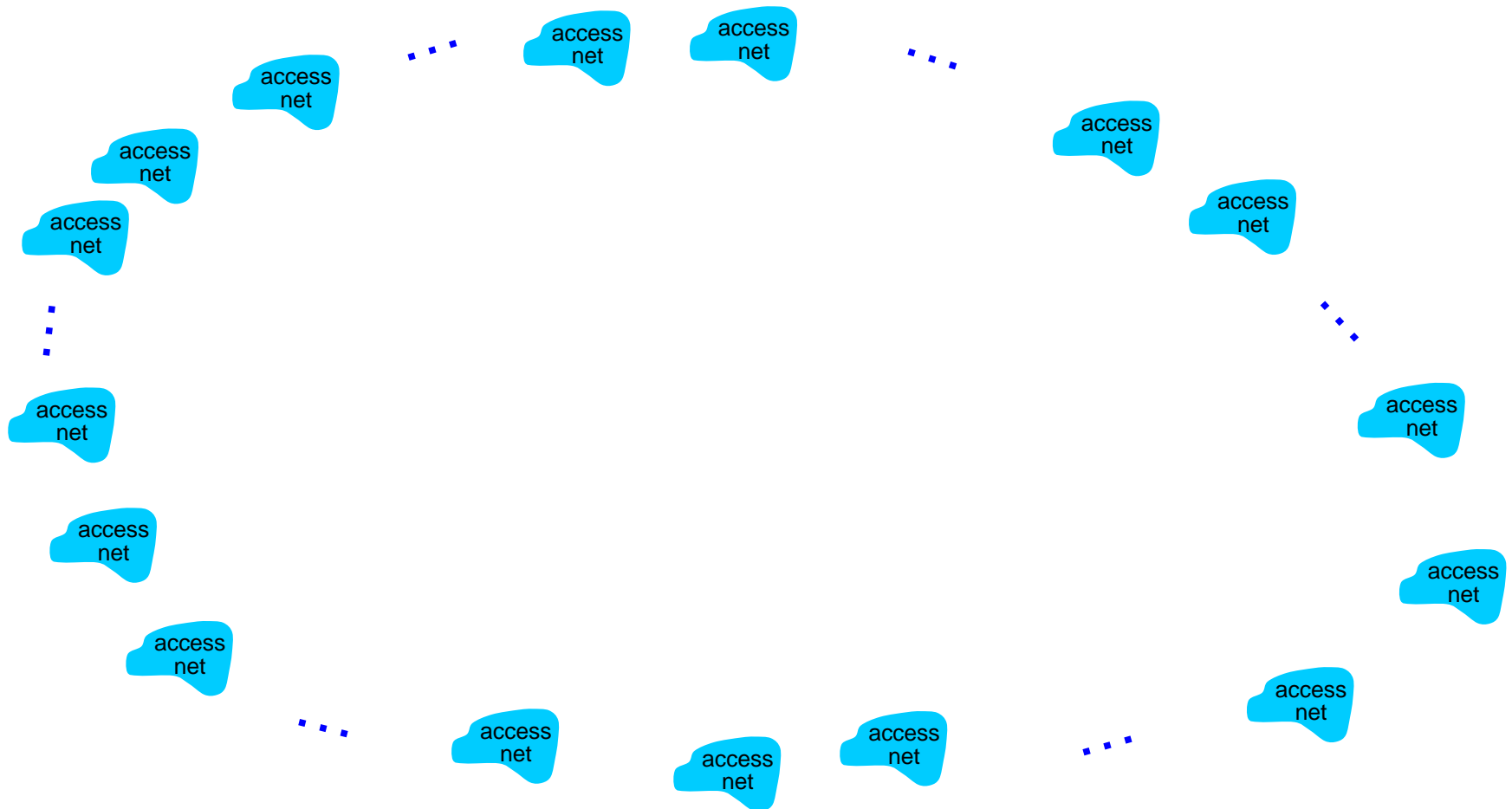
- ▶ Tráfico a ráfagas (com. de paquetes)
 - ▶ Compartición de recursos
 - ▶ Más simple, sin establecimiento de llamada
- ▶ **Posibilidad de congestión excesiva:** retardos en los paquetes y pérdidas (Com. de paquetes)
 - ▶ Protocolos para transferencia confiable de datos, control de congestión
- ▶ **Q: ¿Cómo suministrar un comportamiento como conmutación de circuitos?**
 - ▶ Ancho de banda que garantiza las necesidades de aplicaciones de audio/video
 - ▶ Sin resolver todavía

Estructura de Internet: Red de redes

- ▶ Sistemas finales (hosts) se conectan a Internet vía un Internet Service Providers (ISP) de acceso
 - ▶ Ej: ISP residencial, de una empresa, la Universidad.
 - ▶ Los ISPs de acceso deben estar interconectados también.
 - ▶ Así que dos hosts pueden enviarse paquetes entre ellos.
- ▶ La red de redes resultante es muy compleja
 - ▶ La evolución ha sido dirigida por temas **económicos** y **políticas nacionales**.
- ▶ Veamos la estructura interna de Internet.

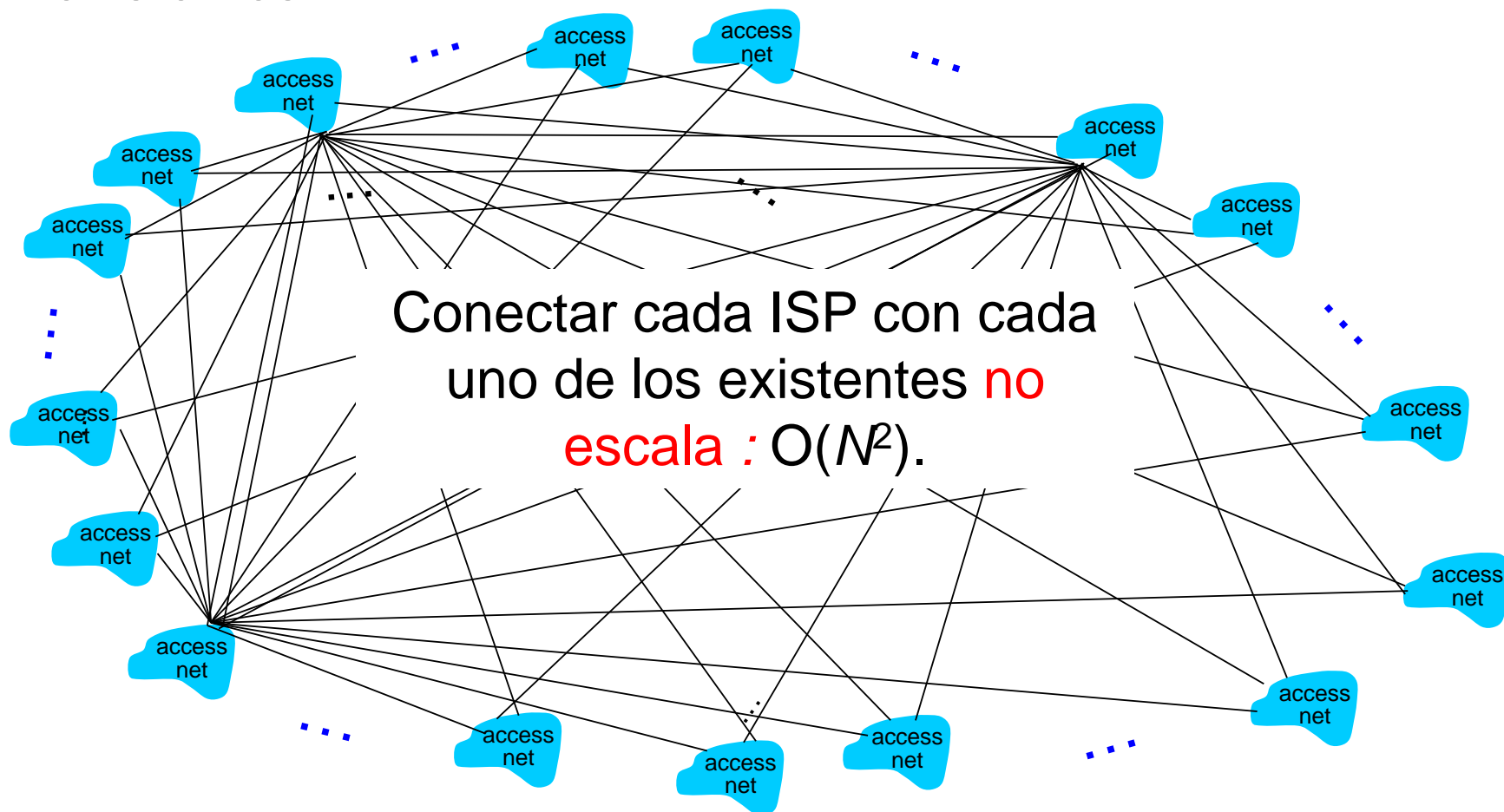
Estructura de Internet: red de redes

- **Q:** ¿Dado que hay millones de ISPs de acceso, ¿cómo interconectarlos entre ellos?



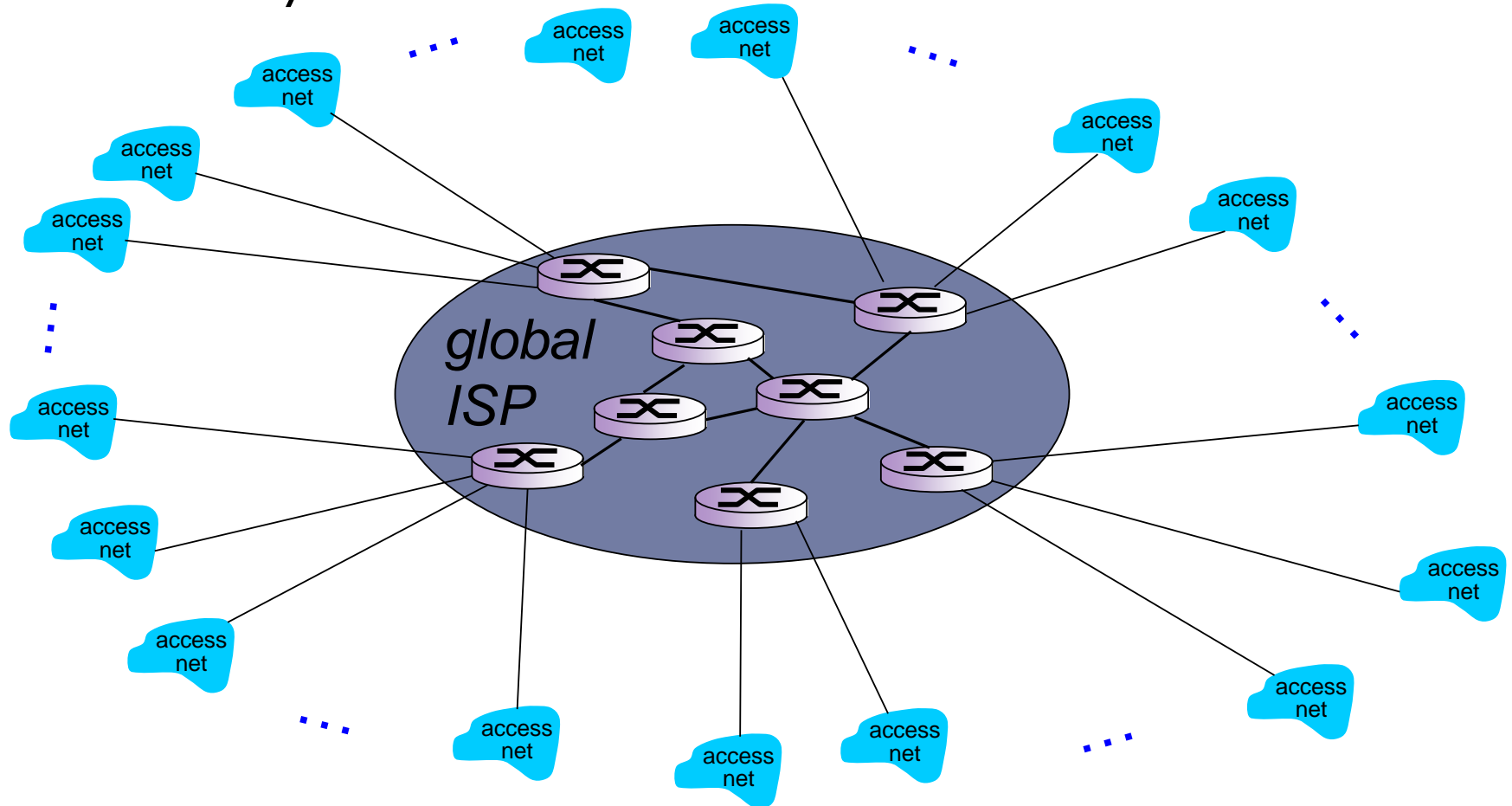
Estructura de Internet: red de redes

- *Opción:* Conectar cada ISP de acceso con cada uno de los existentes



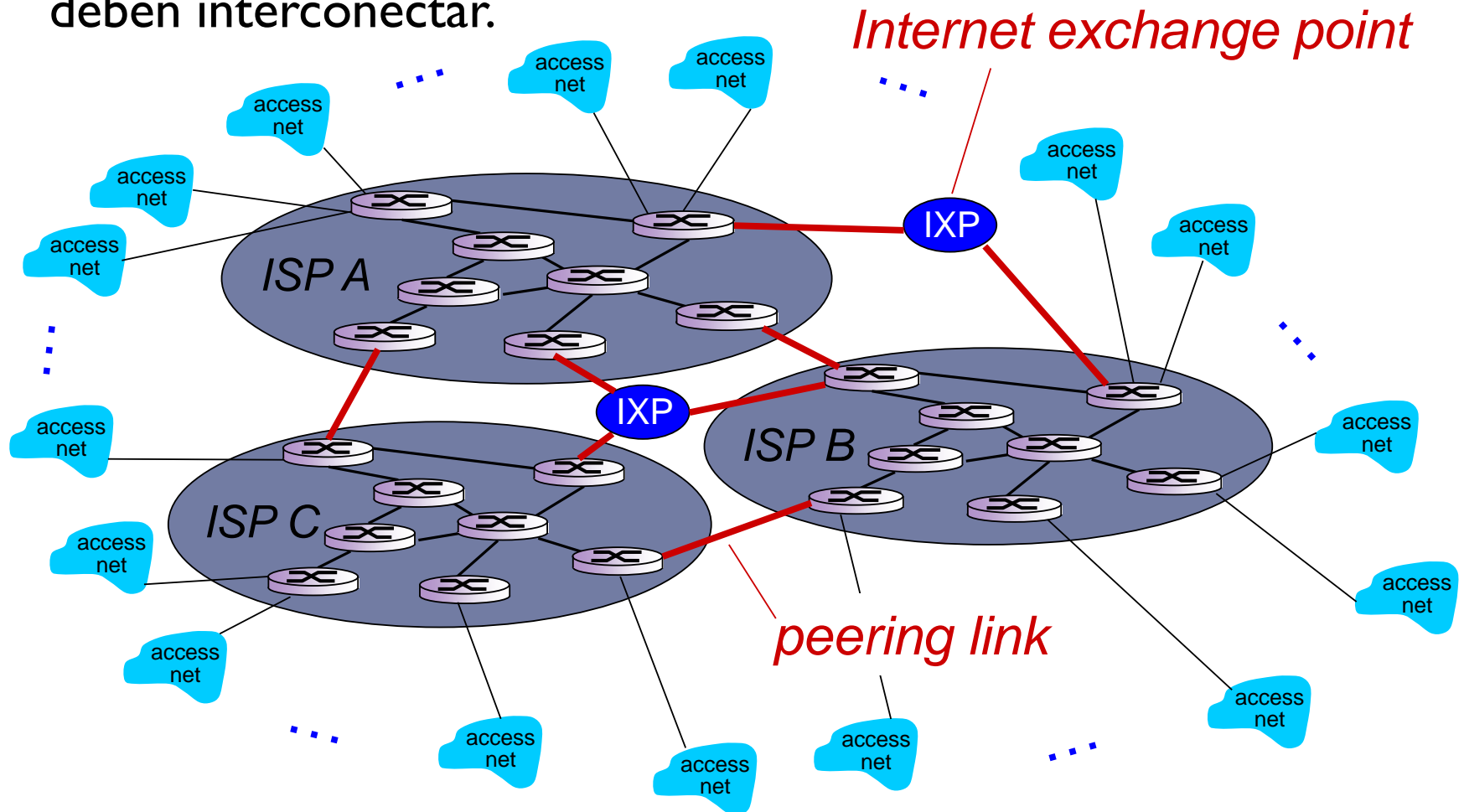
Estructura de Internet: red de redes

- **Opción:** Conectar cada ISP de acceso con un ISP global de tránsito.
ISP **Cliente** y el **Proveedor** ISPs tienen un acuerdo económico.



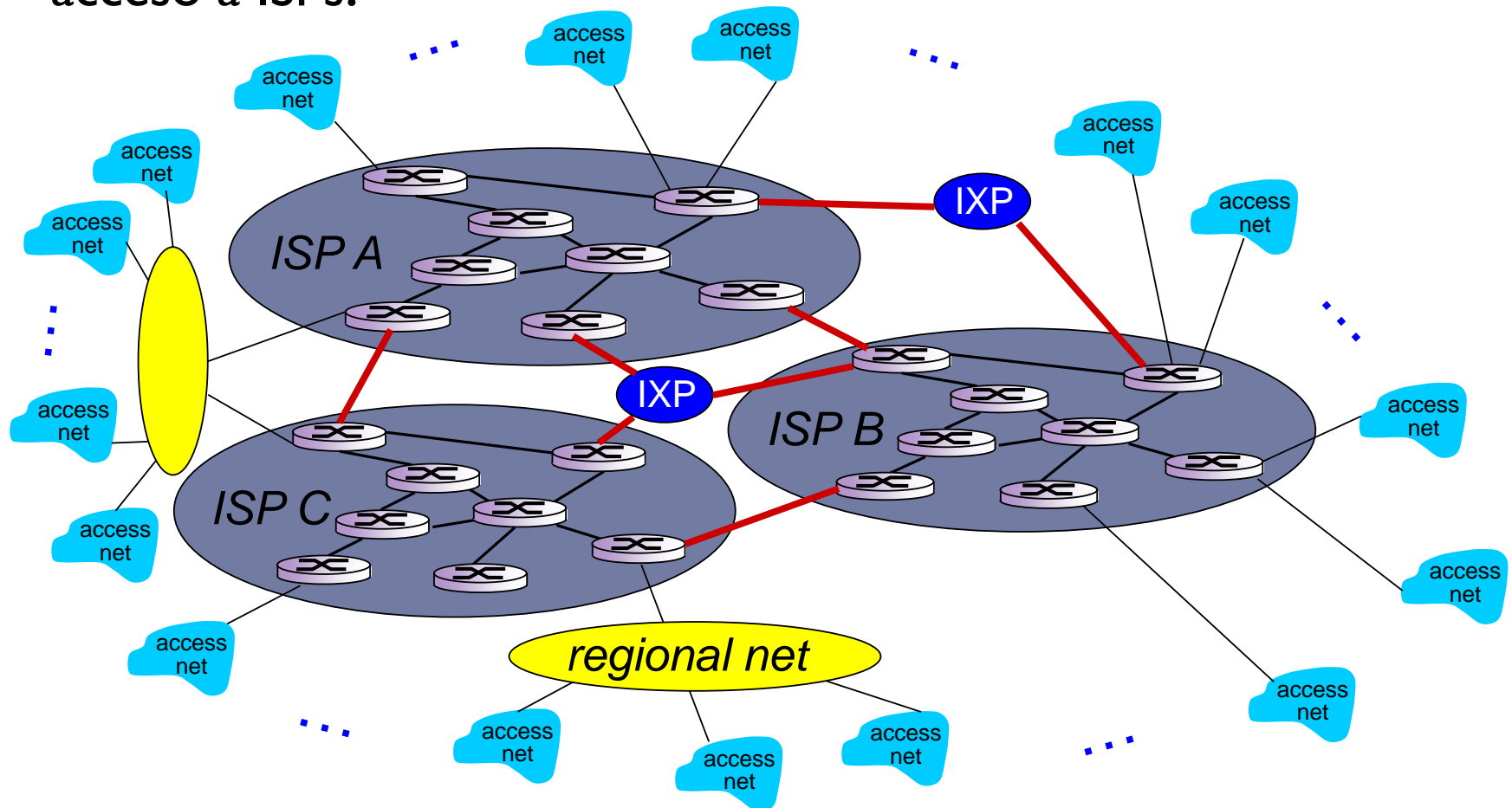
Estructura de Internet: red de redes

- ▶ Pero si un ISP global es viable, habrá competidoresque se deben interconectar.



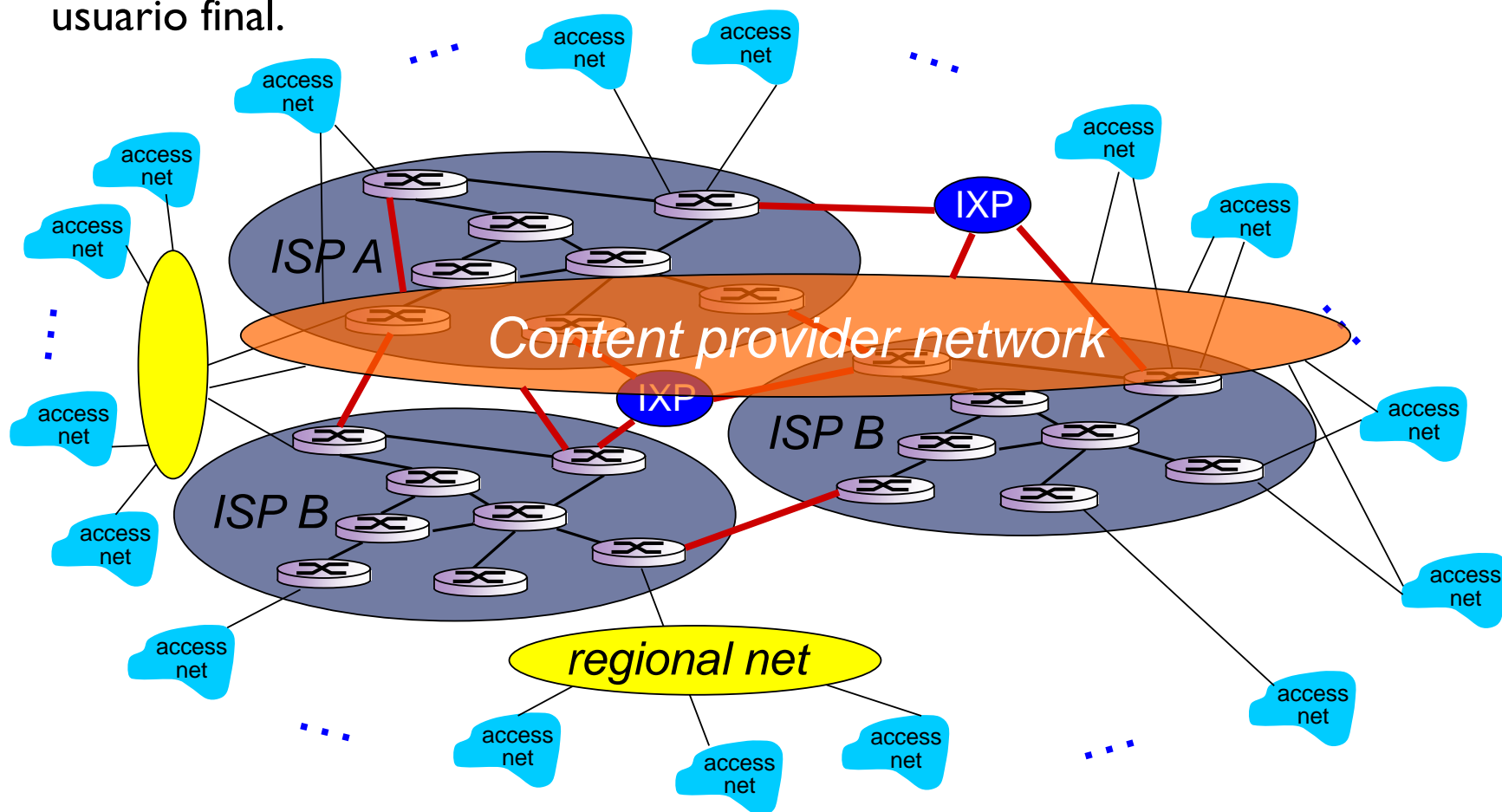
Estructura de Internet: red de redes

- ... y redes regionales puede aparecer para conectar redes de acceso a ISPs.

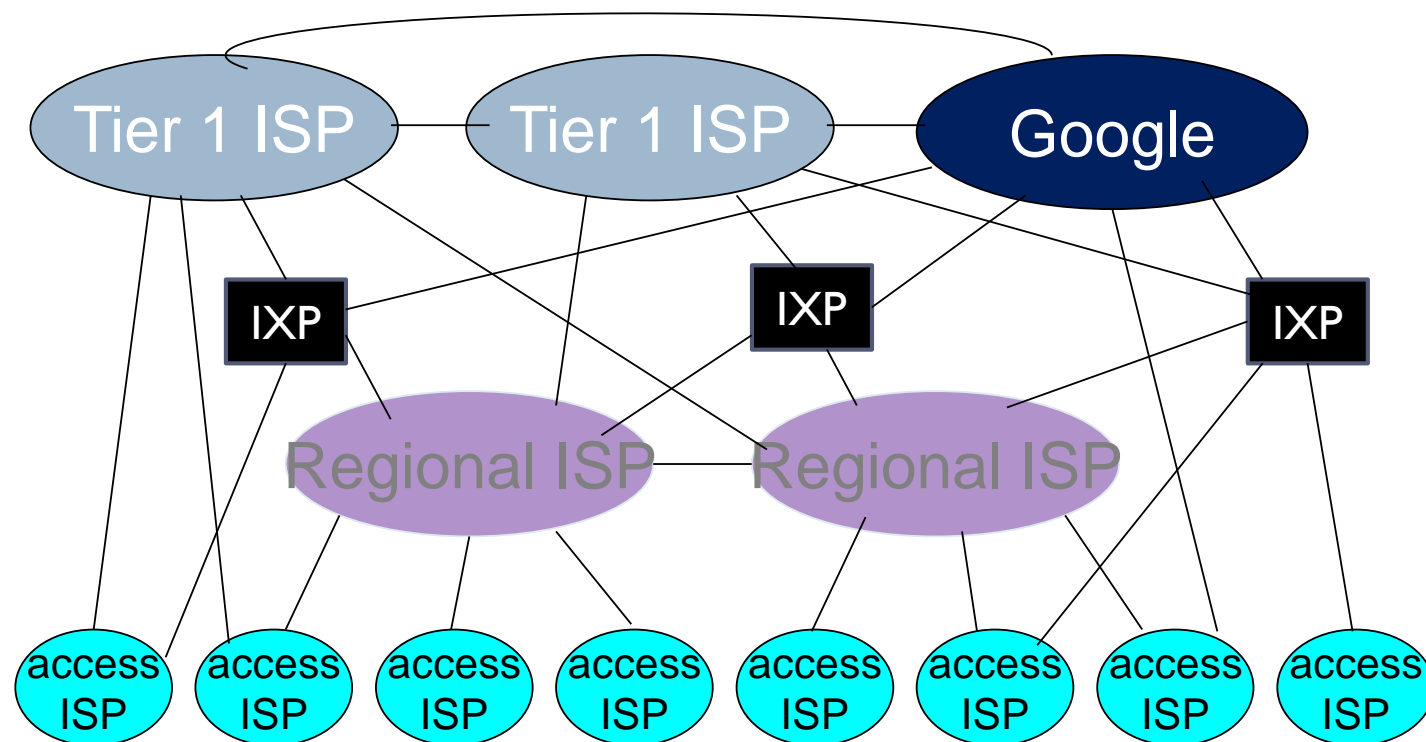


Estructura de Internet: Red de redes

- ... y redes de proveedores de contenido (e.g., Google, Microsoft, Akamai) puede montar su propia red para ofrecer servicios, y contenido cercano al usuario final.

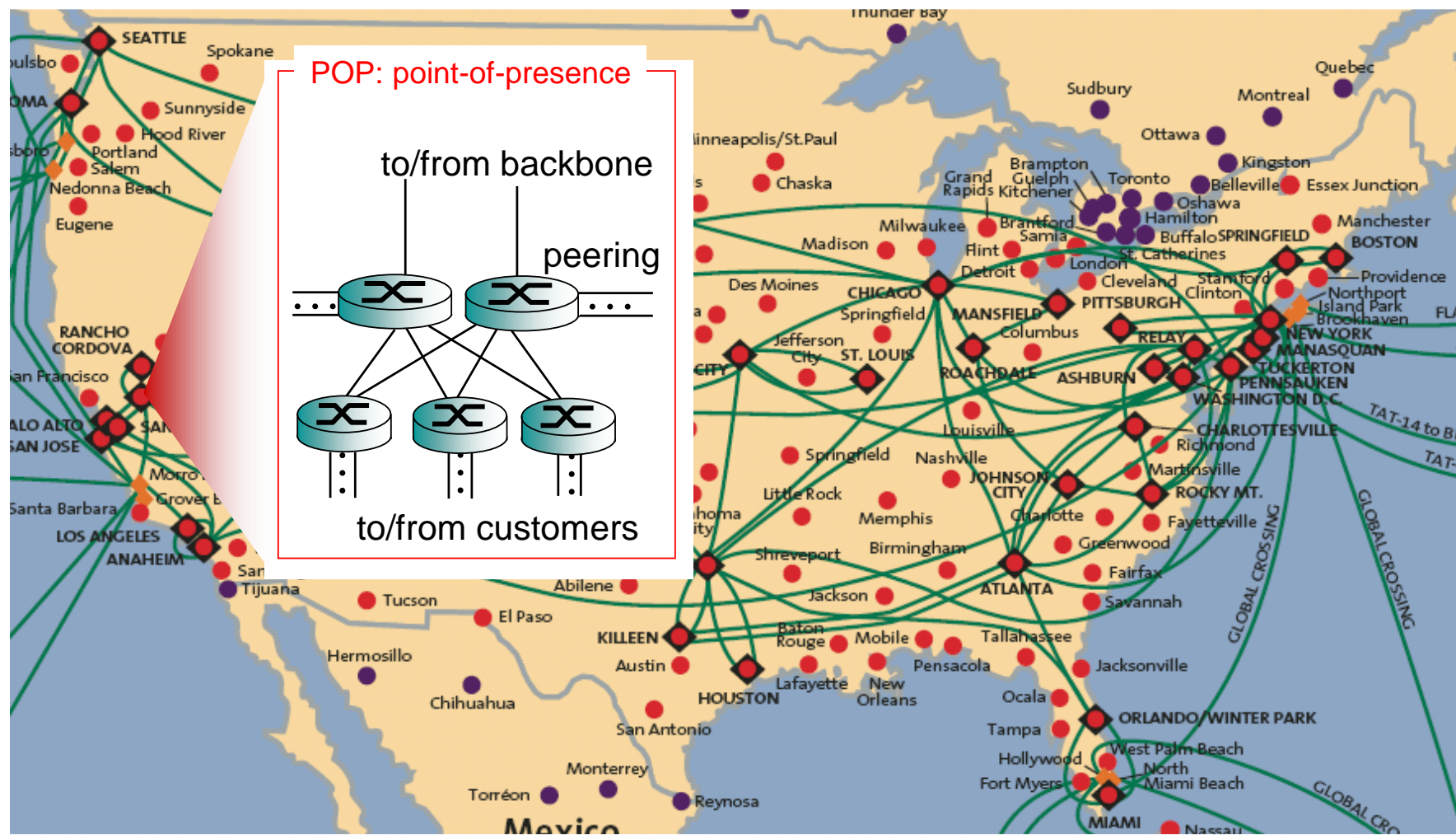


Estructura de Internet: Red de redes



- ▶ En el centro: # pequeño de grandes redes bien interconectadas.
 - ▶ “tier-I” ISPs comerciales (e.g., Level 3, Sprint, AT&T, NTT), cobertura nacional e intern.
 - ▶ Redes de proveedores de contenidos (e.g., Google): redes privadas que conectan sus centros de datos a Internet (con frecuencia, no usan los tier-I ni los regionales)

Ejemplo: Tier-I Sprint



Niveles en los protocolos de Internet

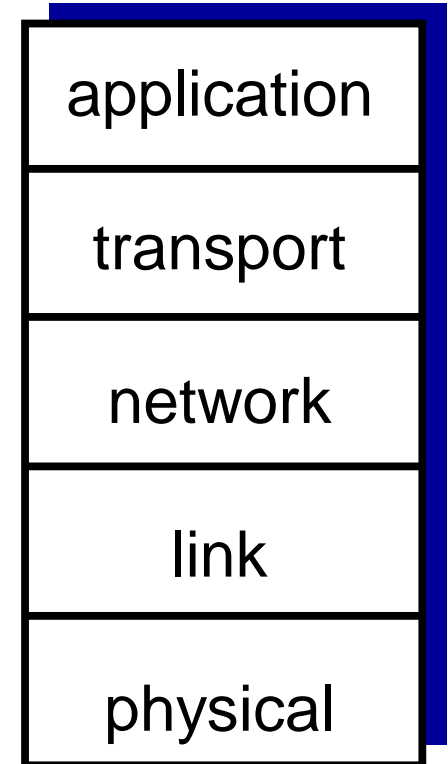
- ▶ *Redes son complejas, con muchos componentes:*
 - ▶ hosts
 - ▶ routers
 - ▶ Enlaces de varios tipos
 - ▶ Aplicaciones
 - ▶ Protocolos
 - ▶ hardware, software
- ▶ *Pregunta:*
 - ▶ ¿Hay alguna esperanza de organizar la estructura de la red?

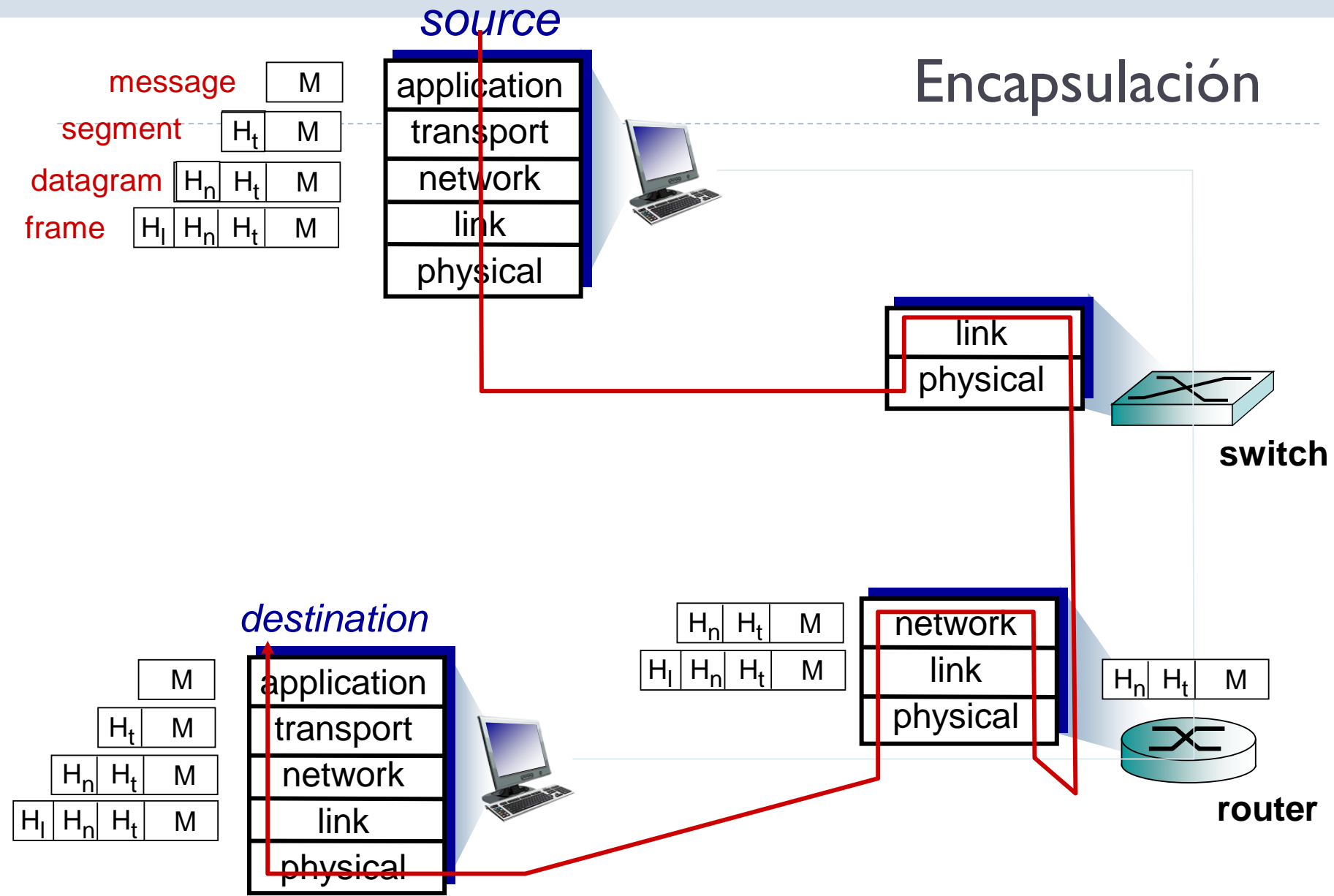
¿Por qué niveles?

- ▶ Para tratar con sistemas complejos:
 - ▶ Estructura explícita que permite la identificación, relación de componentes de sistemas complejos
 - ▶ *Modelo de referencia por capas* (layers)
- ▶ Modularización facilita el mantenimiento, actualización de los sistemas
 - ▶ El cambio de la implementación del servicio de una capa es transparente al resto del sistema
 - ▶ P. ej., cambio en TCP no afecta a IP

Pila de protocolos de Internet

- ▶ **Aplicación:** para soportar aplicaciones de red (nuestra asignatura)
 - ▶ FTP, SMTP, HTTP
- ▶ **Transporte:** transferencia de datos entre procesos de hosts diferentes
 - ▶ TCP, UDP
- ▶ **Red:** enrutamiento de datagramas de la fuente al destino
 - ▶ IP, routing protocols
- ▶ **Enlace:** transferencia de datos entre elementos de red vecinos (p.ej. conectados al mismo switch)
 - ▶ Ethernet, 802.11 (WiFi), PPP
- ▶ **Físico:** bits “en el cable”





source

Encapsulación

switch

router

destination



Seguridad en la red

- ▶ **Campos de la seguridad en red:**
 - ▶ ¿Cómo los chicos malos pueden atacar nuestras ordenadores en red?
 - ▶ ¿Cómo podemos defendernos?
 - ▶ ¿Cómo diseñar arquitecturas que sean inmunes a ataques?
- ▶ **Internet no se diseñó con mucha seguridad en mente**
 - ▶ *Visión original:* “un grupo de usuarios que confiaban los unos con los otros” 😊
 - ▶ **IMPORTANTE:** Consideraciones de seguridad en todos los niveles
- ▶ **Asignatura: ¡seguridad necesaria para proteger las comunicaciones de servicios telemáticos!**

La capa de *Aplicación*

el objetivo de esta asignatura

- ▶ aspectos conceptuales y de implementación de protocolos de aplicaciones en red
 - ▶ modelos de servicio de capa de transporte
 - ▶ paradigma cliente-servidor
 - ▶ paradigma peer-to-peer
- ▶ aprender sobre protocolos examinando protocolos de nivel de aplicación conocidos
 - ▶ HTTP
 - ▶ FTP
 - ▶ SMTP / POP3 / IMAP
 - ▶ DNS
- ▶ crear aplicaciones en red
 - ▶ socket API

Algunas aplicaciones en red

- ▶ e-mail
- ▶ web
- ▶ mensajes de texto
- ▶ login remoto
- ▶ compartición de ficheros P2P
- ▶ juegos en red multiusuario
- ▶ retransmisión (streaming) de video almacenado (YouTube, Hulu, Netflix)
- ▶ Voz sobre IP (e.g., Skype)
- ▶ video conferencia en tiempo real
- ▶ redes sociales
- ▶ motores de búsqueda
- ▶ ...
- ▶ ...

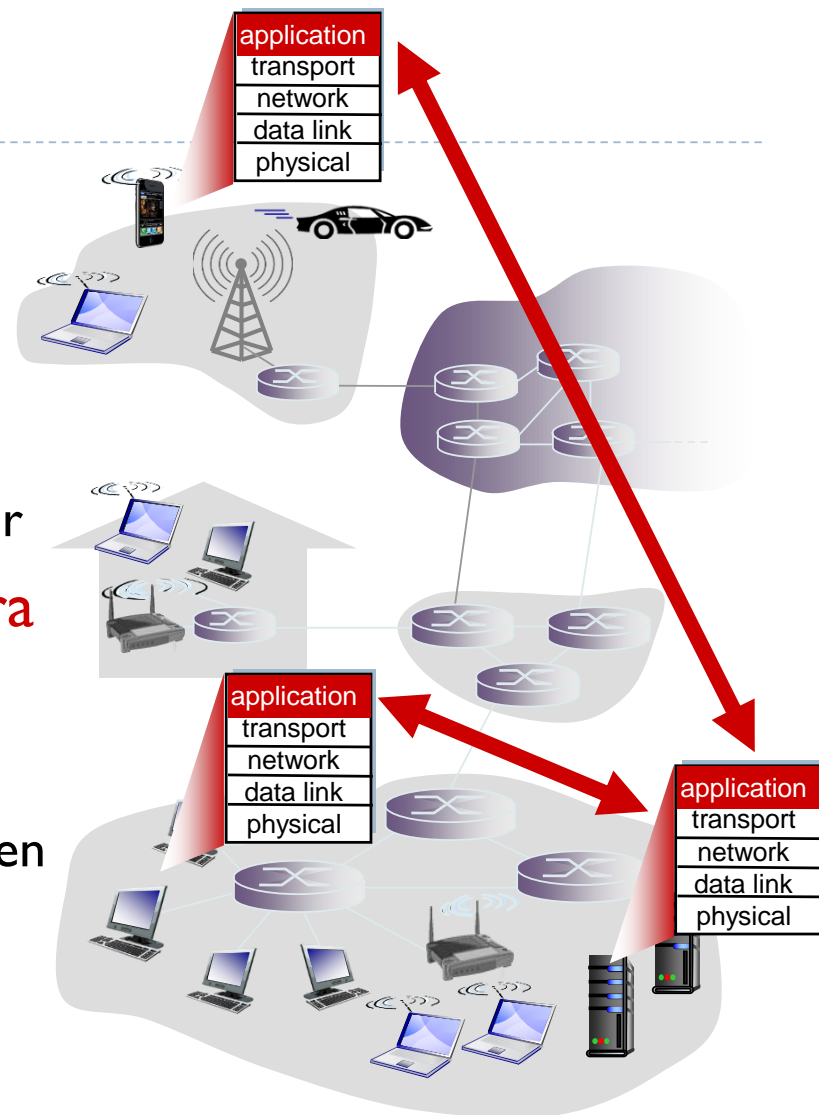
Creando una nueva app

escribir programas que:

- ▶ se ejecutan en (diferentes) *sistemas finales*
- ▶ se comunican a través de la red
- ▶ ej., software de servicio web que se comunica con el software de un navegador

no es necesario escribir software para dispositivos de nucleo de red

- ▶ no ejecutan aplicaciones de usuario
- ▶ las aplicaciones en sistemas finales permiten un rápido desarrollo y propagación

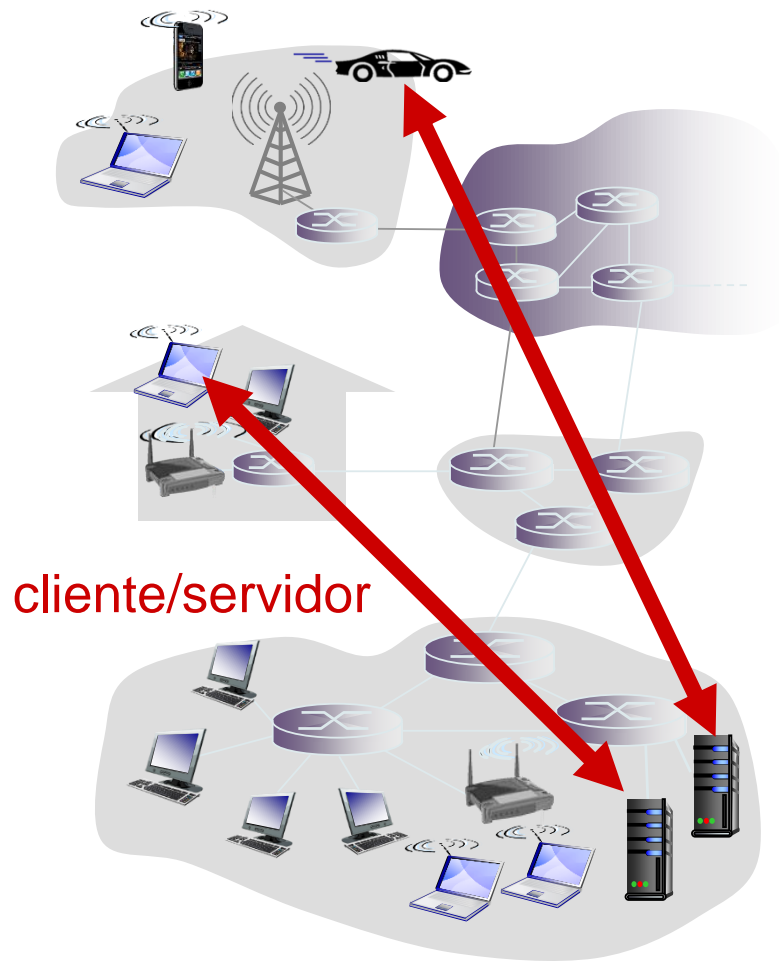


Arquitecturas de las aplicaciones

Posibles estructuras de aplicaciones:

- ▶ cliente-servidor
- ▶ peer-to-peer (P2P)

Arquitectura cliente-servidor



Servidor:

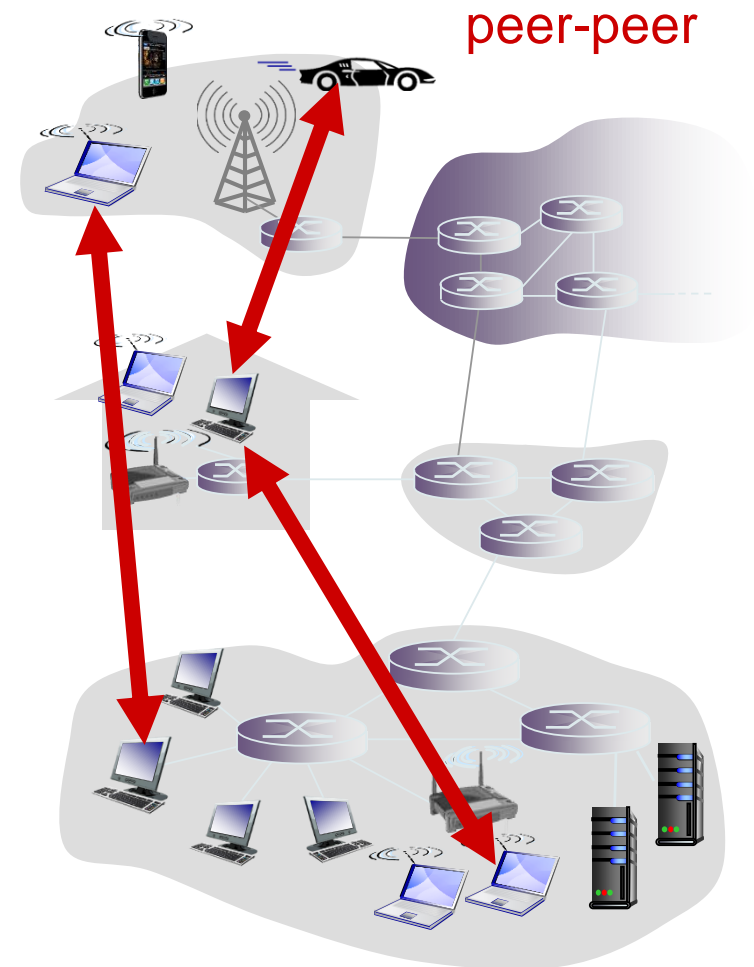
- ▶ siempre activo y en-línea
- ▶ dirección IP permanente
- ▶ en centro de datos por escalabilidad

Clientes:

- ▶ se comunican con el servidor
- ▶ pueden estar intermitentemente conectados
- ▶ pueden tener una IP dinámica
- ▶ no se comunican directamente entre ellos

Arquitectura P2P

- ▶ *no* hay servidor siempre en-línea
- ▶ sistemas finales se comunican directm. de modo arbitrario
- ▶ equipos solicitan un servicio desde otros equipos, y ofrece el servicio en respuesta a otros equipos
 - ▶ *auto-escalable*— nuevos equipos añaden nueva capacidad al servicio, así como nuevas demandas del servicio
- ▶ equipos están intermitentem. conectados y cambian de IP
 - ▶ gestión compleja



Comunicación de procesos

- proceso:** programa en ejecución en un host
- ▶ en el mismo host, dos procesos se comunican usando **comunicación inter-proceso** (definida por el SO)
 - ▶ procesos en diferentes hosts se comunican mediante el intercambio de **mensajes**

clientes, servidores

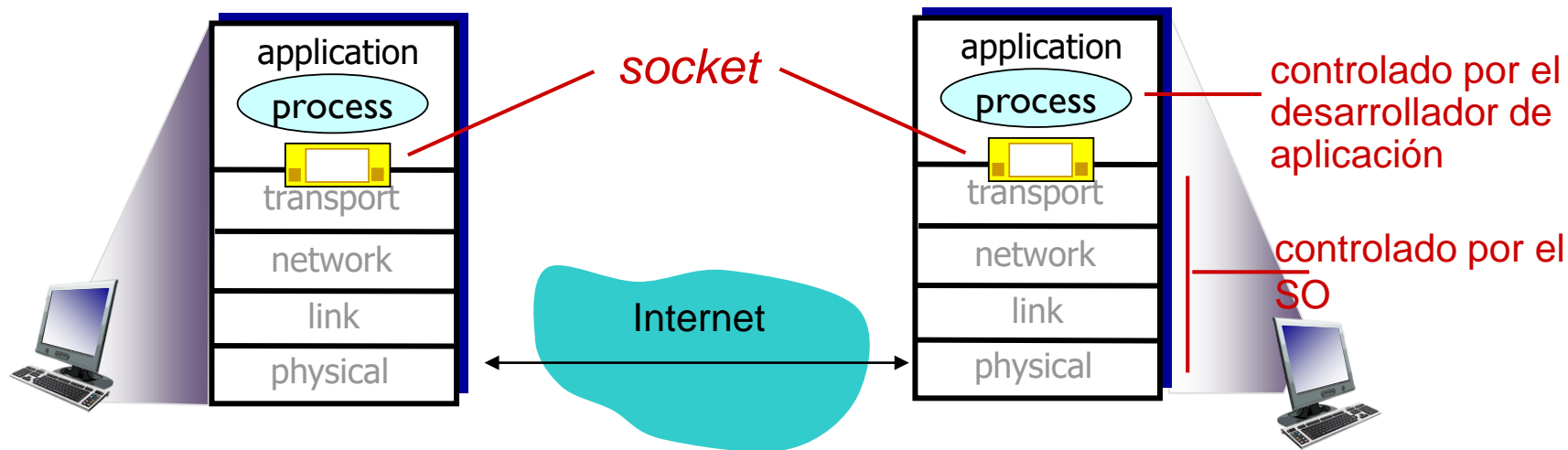
proceso cliente: proceso que inicia una comunicación

proceso servidor: proceso que espera ser contactado

- ❖ además: aplicaciones en arquitecturas P2P tienen procesos clientes y servidores

Sockets

- ▶ Los procesos **envían/reciben** mensajes **hacia/desde** su socket
- ▶ Un socket es análogo a una puerta
 - ▶ proceso que envía “empuja” mensajes de puertas a fuera
 - ▶ proceso que envía confía en la infraestructura de transporte al otro lado de la puerta para entregar el mensaje al socket del proceso que recibe



Direccionamiento de procesos

- ▶ para recibir mensajes el proceso debe tener un *identificador*
- ▶ un host tiene una dirección IP única de 32-bit
- ▶ P: ¿Es suficiente la IP de un host para identificar un proceso?
 - R: no, en un host se pueden ejecutar muchos procesos simultáneamente
- ▶ *identificador* incluye tanto *dirección IP* como *números de puerto* asociados con el proceso en el host
- ▶ ejemplo de números de puerto:
 - ▶ servidor HTTP: 80
 - ▶ servidor de correo: 25
- ▶ para enviar un mensaje al servidor web `gaia.cs.umass.edu` :
 - ▶ *dirección IP*: 128.119.245.12
 - ▶ *número de puerto*: 80

Un protocolo de nivel de aplicación define ...

- ▶ **tipos de mensajes intercambiados**

- ▶ ej.: consulta, respuesta

- ▶ **sintaxis de los mensajes**

- ▶ qué campos hay en el mensaje y cómo se delimitan

- ▶ **semántica del mensaje**

- ▶ significado de la información en los campos

- ▶ **reglas**

- ▶ cuándo y cómo los procesos envían y reciben mensajes

protocolos abiertos:

- ▶ definidos en RFCs
- ▶ permiten interoperabilidad
- ▶ ej., HTTP, SMTP

protocolos propietarios:

- ▶ ej., Skype

¿Qué servicio de transporte necesita una aplicación?

integridad de datos

- ▶ algunas aplicaciones (ej., transferencia de ficheros, transferencia web) requieren 100% de fiabilidad
- ▶ otras aplicaciones (ej., audio) pueden tolerar algunas pérdidas

rendimiento

- algunas aplicaciones (ej., multimedia) requieren un rendimiento alto para ser “eficaces”
- algunas aplicaciones (“aplicaciones elásticas”) hacen uso del rendimiento que tienen disponible

temporización

- ▶ algunas aplicaciones (ej., telefonía por Internet, juegos interactivos) requieren un retardo bajo para ser “eficaces”

seguridad

- ❖ cifrado, integridad de datos...

Req. servicio de transporte: apps comunes

Application	Data Loss	Throughput	Time-Sensitive
File transfer/download	No loss	Elastic	No
E-mail	No loss	Elastic	No
Web documents	No loss	Elastic (few kbps)	No
Internet telephony/ Video conferencing	Loss-tolerant	Audio: few kbps—1 Mbps Video: 10 kbps—5 Mbps	Yes: 100s of msec
Streaming stored audio/video	Loss-tolerant	Same as above	Yes: few seconds
Interactive games	Loss-tolerant	Few kbps—10 kbps	Yes: 100s of msec
Instant messaging	No loss	Elastic	Yes and no

Servicios de protocolos de transporte en Internet

servicio TCP:

- ▶ *transporte confiable* entre los procesos emisor y receptor
- ▶ *control de flujo*: servidor no “satura” al cliente
- ▶ *control de congestión*: se ralentiza al emisor cuando la red está sobrecargada
- ▶ *no ofrece*: temporización, rendimiento mínimo garantizado, seguridad
- ▶ *orientado a conexión*: se debe establecer una conexión inicial entre los procesos emisor y receptor

servicio UDP:

- ▶ *transferencia de datos no confiable* entre los procesos emisor y receptor
- ▶ *no ofrece*: transporte confiable, control de flujo, control de congestión, temporización, rendimiento garantizado, seguridad, establecimiento de conexión

P: ¿por qué? ¿por qué existe UDP?

Aplicaciones en Internet y protocolo de transporte

Application	Application-Layer Protocol	Underlying Transport Protocol
Electronic mail	SMTP [RFC 5321]	TCP
Remote terminal access	Telnet [RFC 854]	TCP
Web	HTTP [RFC 2616]	TCP
File transfer	FTP [RFC 959]	TCP
Streaming multimedia	HTTP (e.g., YouTube)	TCP
Internet telephony	SIP [RFC 3261], RTP [RFC 3550], or proprietary (e.g., Skype)	UDP or TCP

Seguridad con TCP

- ▶ TCP & UDP
 - ▶ no hay cifrado
 - ▶ passwords en claro enviados sobre sockets cruzan Internet en claro
- ▶ SSL
 - ▶ ofrece cifrado a TCP
 - ▶ integridad de datos
 - ▶ autenticación de los extremos
- ▶ SSL está en la capa de aplicación
 - ▶ Aplicaciones usan librerías SSL, que “hablan” con TCP
- ▶ API para sockets SSL
 - ▶ passwords en claro enviados sobre sockets SSL cruzan Internet cifrados

Bibliografía

- ▶ **Capítulos 1.1-1.3, 1.5 y 2.1.**
 - ▶ J. F. Kurose, K.W. Ross. *Computer Networking, A top-down approach featuring the Internet*. 6th Ed.

- ▶ Parte de estas transparencias son copyright 1996-2014 J.F. Kurose, K.W. Ross All Rights Reserved. Last update: Nov. 14, 2014 Comments welcome: kurose at cs.umass.edu