Capítulo 2: Capa Aplicación: Principios de las aplicaciones

ELO322: Redes de Computadores
Agustín J. González

Este material está basado en:

- Material del texto *Computer Networking: A Top Down Approach, by* Jim Kurose and Keith Ross

Capítulo 2: Capa Aplicación

- 2.1 Principios de las aplicaciones de red
- 2.2 Web y HTTP
- 2.3 Correo Electrónico
  - SMTP, POP3, IMAP
- 2.4 DNS
- 2.5 P2P para archivos compartidos
- 2.6 Video streaming y redes de distribución de contenidos
- 2.7 Programación de sockets con UDP y TCP

Capítulo 2: Capa Aplicación

Objetivos:

- Veremos los aspectos conceptuales y de implementación de los protocolos de aplicación
  - Modelo de servicio de la capa transporte
  - Paradigma cliente-servidor
  - Paradigma peer-to-peer (entre pares)
  - Redes de distribución de contenidos
- Aprendizaje de protocolos examinando protocolos de aplicación populares
  - HTTP
  - SMTP / POP3 / IMAP
  - DNS
- Programación de aplicaciones de red
  - API de sockets

Algunas aplicaciones de red

- E-mail
- Web
- Mensajería instantánea
- Login remoto
- Compartición de archivos P2P
- Juegos de red multi-usuarios
- Reproducción de clips de video almacenados (YouTube, Hulu, Netflix)
- Voz sobre IP (e.g. Skype)
- Conferencias de video en tiempo real
- Redes sociales
- Buscadores ...
- .....

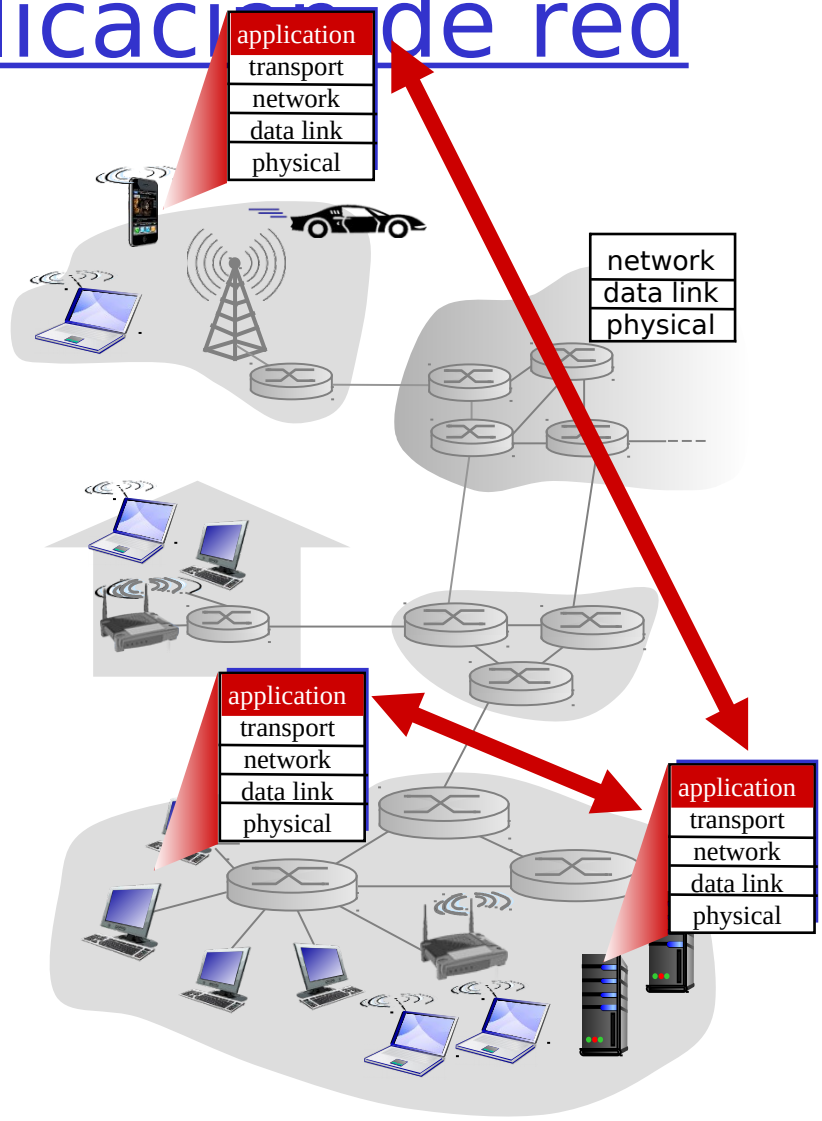
Creación de una aplicación de red

Tú creas programas que

- Corren en diferentes sistemas
- Se comunican por la red.
- Ej. Web: Programa del servidor Web se comunica con el programa del navegador

No requiere hacer código para los dispositivos en la red interna

- Dispositivos internos de la red (routers, switches) **no corren** aplicaciones de usuarios
- Aplicaciones en el periferia permite desarrollos y propagación rápidos

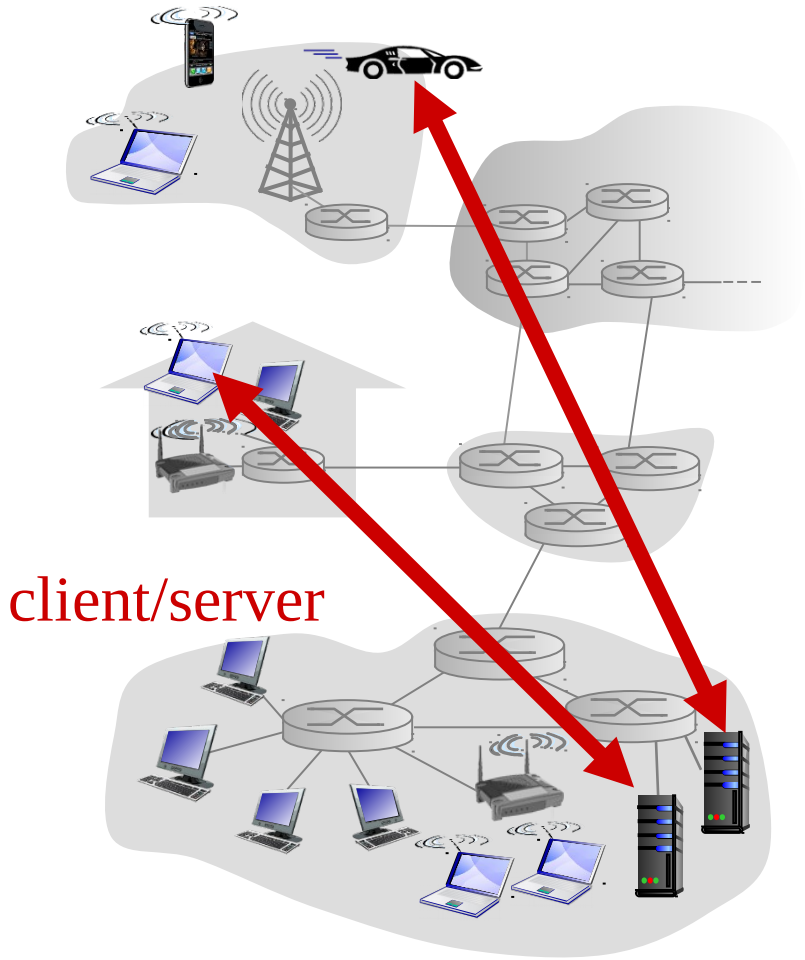


Arquitecturas de Aplicación

Posibles arquitecturas para las aplicaciones:

- Cliente-servidor
- Peer-to-peer (P2P)

Arquitectura Cliente-servidor



Servidor:

- Computador siempre on
- Dirección IP permanente
- Granja de servidores por **escalamiento**

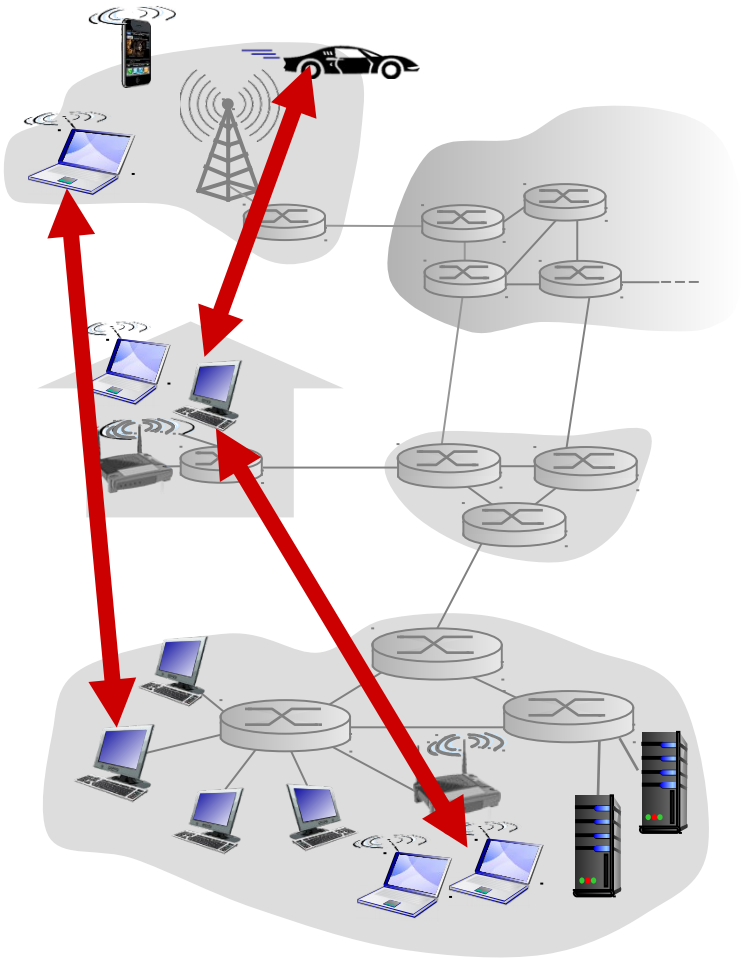
Cliente:

- Se comunica con servidor
- Puede conectarse intermitentemente
- Puede tener direcciones IP dinámicas (no estática)
- No se comunican directamente entre sí (dos clientes puros)

Escalabilidad: es la habilidad de atender más clientes sin perder calidad. Ej. Radio

Arquitectura P2P Pura

- No hay servidor siempre on
- Sistemas terminales arbitrarios se comunican directamente
- Pares requieren servicios de otros pares, y a cambio proveen el servicio a otros
  - Autoescalable: un par nuevo trae demandas y nueva capacidad de servicio
- Pares se conectan intermitentemente y cambian sus direcciones IP
  - Difícil de administrar



Procesos que se comunican

Proceso: es un programa corriendo en un computador.

- Dentro de la máquina dos procesos se comunican usando comunicación entre procesos (definida por Sistema Operativo).
- Procesos en diferentes hosts se comunican vía intercambio de mensajes

Definiciones

Proceso Cliente:
proceso que inicia la comunicación

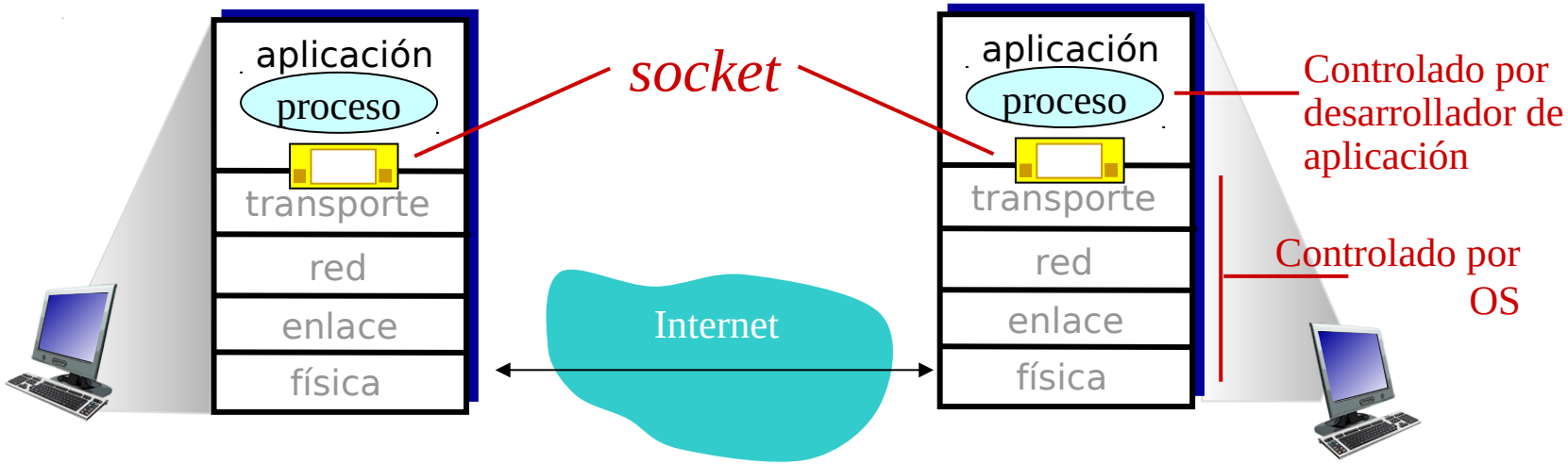
Proceso servidor:
proceso que espera ser contactado

- Nota: Aplicaciones con arquitectura P2P tienen procesos clientes y procesos servidores

Sockets

Analogía (Python):
f = open('workfile', 'w')

- Un proceso envía/recibe mensajes a/desde su socket
- socket es un punto de comunicación entre dos partes (análogo a una puerta)
  - Proceso transmisor envía mensajes por un socket
  - Proceso transmisor confía en la infraestructura de transporte al otro lado de la puerta, la cual lleva los mensajes al socket en el proceso receptor



Direccionamiento de procesos

- Para que un proceso reciba un mensaje, éste debe tener una identificación
- Un terminal/host tiene al menos una dirección IP única de 32 bits.
- Q: ¿Es suficiente la dirección IP para identificar un proceso en un host?

  Respuesta: No, muchos procesos pueden estar corriendo en el mismo host (= computador).

- El identificador de proceso en Internet incluye la dirección IP (host) y un número de puerto (port) asociado con el proceso en ese host.

  El puerto es un número positivo y de 16 bits.
- Ejemplo Servidor web ELO:
  - Dirección IP: 200.1.17.10
  - Número de puerto: 80

Viendo las conexiones de nuestra máquina

- En Linux y Windows netstat (Network Statistic):
  - netstat -t para ver conexiones TCP
  - netstat -u para ver conexiones UDP

- Hasta aquí sabemos:
  - Relación entre aplicación y proceso
  - Necesidad de los puertos
  - Mecanismo de software usado para pedir servicios a capa transporte
- Ahora estudiaremos algunos protocolos.

Protocolos de capa aplicación definen:

- Tipos de mensajes intercambiados, e.g., mensajes de requerimiento y respuesta
- Sintaxis de los mensajes: los campos en los mensajes & cómo éstos son delimitados.
- Semántica de los campos, i.e, significado de la información en los campos
- Reglas para cuándo y cómo los procesos envían y responden a mensajes

Protocolos de dominio público:

- Definidos en RFCs
- Permite inter- operatividad
- Ej: HTTP (WEB), SMTP (email)

Protocolos propietarios:

- Secreto industrial de una empresa
- Ej: Skype

¿Qué servicios de la capa transporte necesita una aplicación?

Confiabilidad en la entrega (Sin pérdida de datos)

- Algunas aplicaciones (e.g., transferencia de archivos, web) requieren transferencia 100% confiable
- otras (e.g., audio) pueden tolerar pérdida

Retardo

- algunas Aplicaciones (e.g., Telefonía en internet, juegos interactivos) requieren bajo retardo para ser “efectivas”

Tasa de datos (“throughput”)

- algunas aplicaciones (e.g., multimedia) requieren una tasa mínima de datos para ser “efectivas”
- otras (“aplicaciones elásticas”) hacen uso de la tasa que obtengan (e.g. correo electrónico).

Seguridad

- Encriptación, integridad de datos, ...

Requerimientos de servicios de transporte de aplicaciones comunes

Aplicación	Tolera Pérdidas?	Tasa de datos	Sensible a Tiempo?
file transfer	no	elastic	no
e-mail	no	elastic	no
Web documents	no	elastic	no
real-time audio/video	sí	audio: 5kbps-1Mbps video:10kbps-5Mbps	yes, 100’s msec
stored audio/video	sí	Igual al de arriba	yes, few secs
interactive games	sí	few kbps up	yes, 100’s msec
text messaging	no	elastic	yes and no

Servicios de dos protocolos de capa de transporte en Internet

Servicio TCP

(Transmission Control Protocol):

- *Ofrece Transporte confiable* entre proceso Transmisor (Tx) y Receptor (Rx)
- *Tiene Control de flujo:* Tx no sobrecargará al Rx
- *Tiene Control de congestión:* el Tx se frena cuando la red está sobrecargada
- *No provee:* garantías de retardo, ancho de banda mínimos, seguridad
- *Es Orientado a la conexión* establecer conexión (setup) requerido entre procesos cliente y servidor antes de transferencia

Servicio UDP

(User Datagram Protocol):

- Transferencia de datos no confiable entre proceso Tx y Rx.
- No provee: confiabilidad, control de flujo, control de congestión, garantías de retardo, ancho de banda, seguridad, establecimiento de conexión

Q: ¿Por qué existe UDP?

Aplicaciones Internet: aplicación, protocolo de transporte

Aplicación	Protocolo capa aplicación	Protocolo de transporte que lo sustenta
e-mail	SMTP [RFC 2821]	TCP
remote terminal access	Telnet [RFC 854]	TCP
Web	HTTP [RFC 2616]	TCP
file transfer	FTP [RFC 959]	TCP
streaming multimedia	HTTP (e.g. YpuTube) RTP [RFC 1889]	TCP or UDP
Internet telephony	SIP, RTP, proprietary (e.g. skype)	TCP or UDP

Dando seguridad a TCP

TCP & UDP

- No encriptan
- Passwords en texto legibles son enviadas a través de Internet

SSL (Secure Socket Layer)

- Provee una conexión TCP con encriptación
- Da integridad de datos
- Los puntos extremos se autentican

SSL está en capa aplicación

- Aplicaciones usan bibliotecas SSL, que usan servicios TCP

SSL socket API

- Password son enviadas encriptadas antes de ser pasada a TCP y atraviesan así Internet.
- Tema fuera de este curso.
- (API: Application Programming Interface)

Capítulo 2: Capa Aplicación

- 2.1 Principios de las aplicaciones de red
- 2.2 Web y HTTP
- 2.3 Correo Electrónico
  - SMTP, POP3, IMAP
- 2.4 DNS
- 2.5 P2P para archivos compartidos
- 2.6 Video streaming y redes de distribución de contenidos
- 2.6 Programación de sockets con UDP y TCP