7 PROTOCOLOS DE TRANSPORTE

Objetivo

- Ofrecer una visión panorámica del transporte de datos de las aplicaciones a través de Internet.

Manual de clases

Tema 7 de:
PROTOCOLOS DE INTERNET

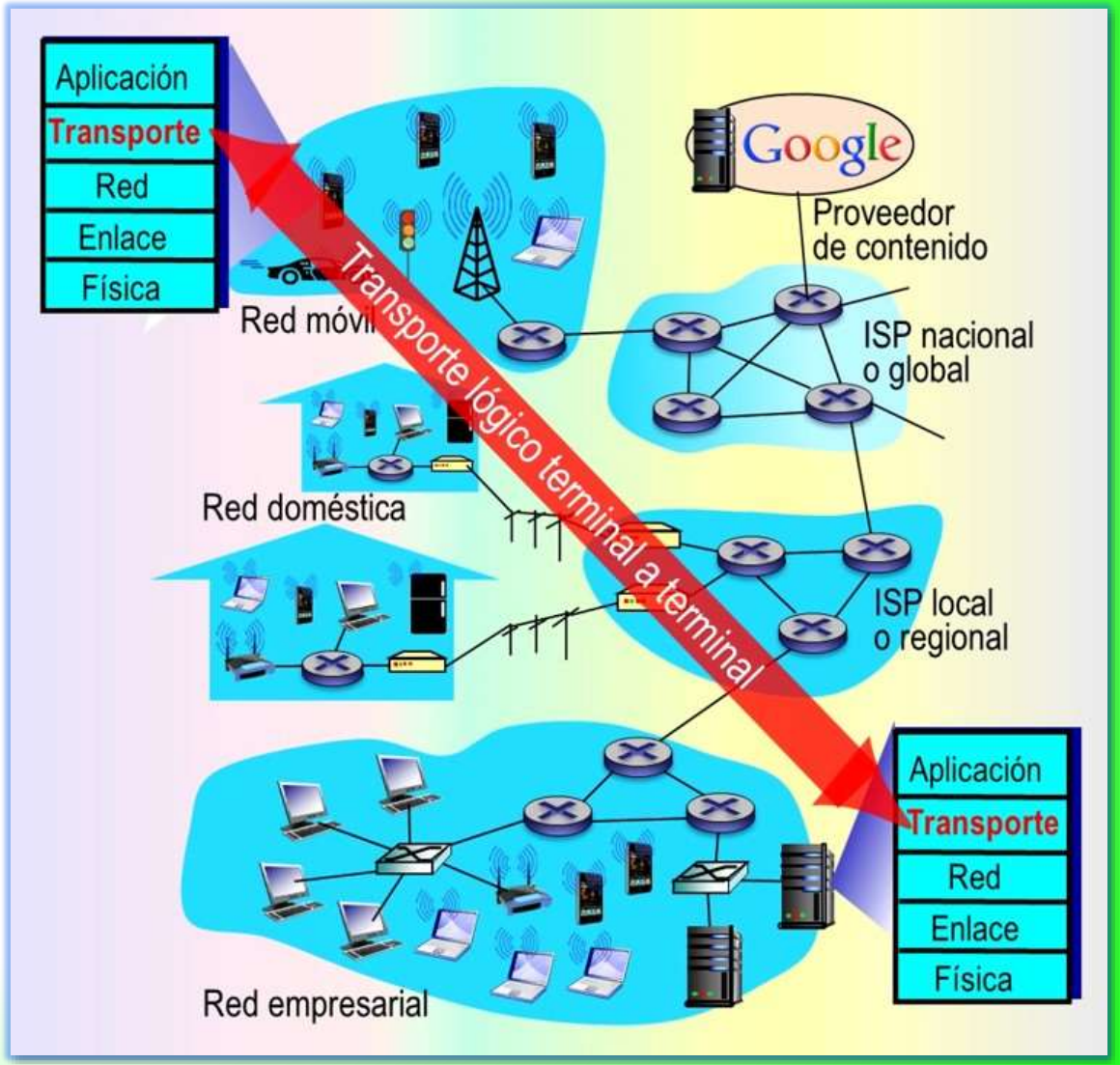
Edison Coimbra G.

Última modificación:
31 de agosto de 2022

PROTOCOLOS DE TRANSPORTE

La comunicación lógica (Kurose, 2017)

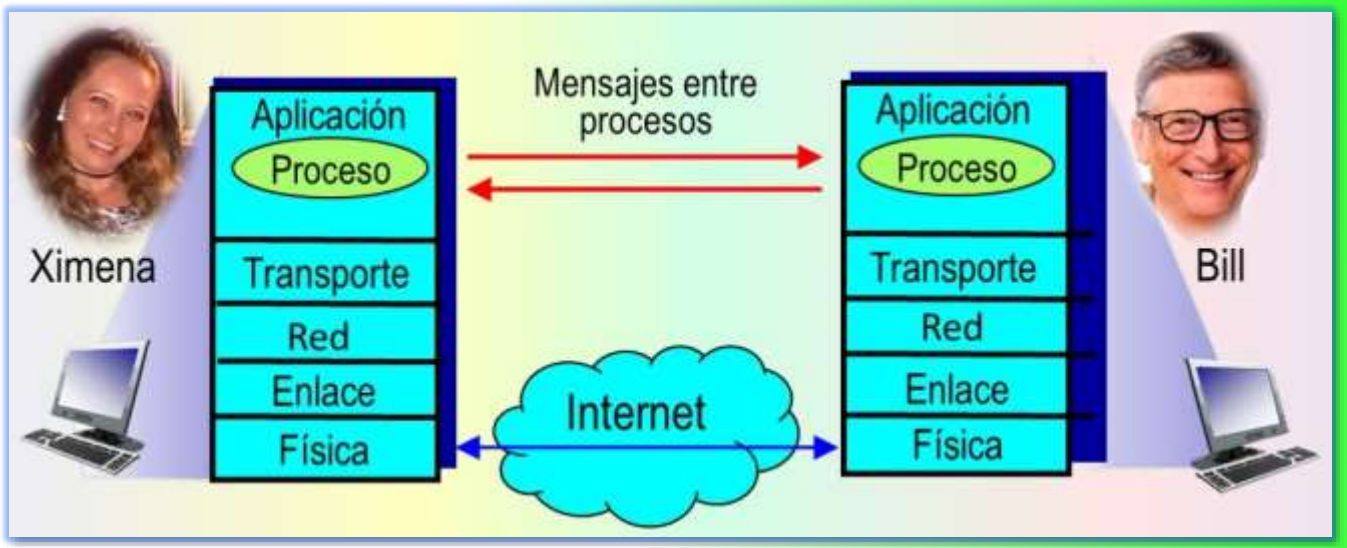
- **Un protocolo** de la capa de transporte proporciona una comunicación lógica entre procesos de aplicación que se ejecutan en host diferentes.
- ▶**Por comunicación lógica** se quiere decir que, desde la perspectiva de la aplicación, es como si los hosts que ejecutan los procesos estuvieran conectados directamente (vea la figura);
  - ✉En realidad, los hosts pueden encontrarse en puntos opuestos del planeta, conectados mediante routers y a través de un amplio rango de tipos de enlace.
- ▶**Los procesos de aplicación** utilizan la comunicación lógica proporcionada por la capa de transporte para enviarse mensajes entre sí, sin preocuparse por los detalles de la infraestructura utilizada para transportar estos mensajes.



PROTOCOLOS DE TRANSPORTE

Concepto de procesos

(Kurose, 2017)

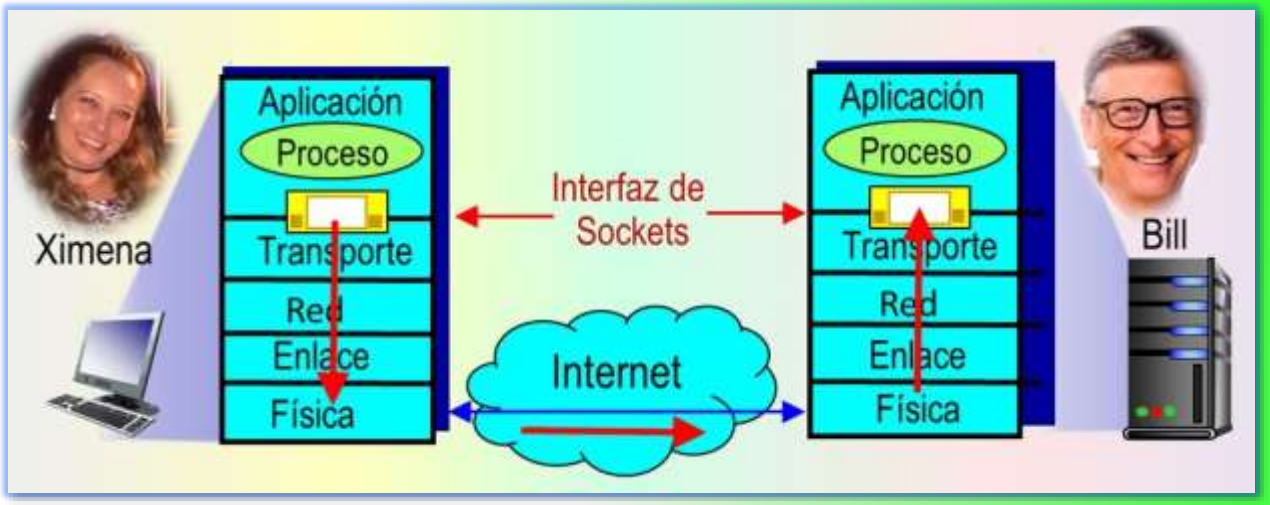


- **Antes de crear** una aplicación de red, se necesita disponer de conocimientos básicos sobre cómo se comunican entre sí los programas que se ejecutan en varios hosts. En la jerga de los sistemas operativos, los que se comunican no son programas, sino procesos.
- **Un proceso** puede interpretarse como un programa que se ejecuta dentro de un host.
  - ▶Si se ejecutan en el mismo host, se comunican entre sí mediante sistemas de comunicación interprocesos, aplicando reglas gobernadas por el sistema operativo del host.
  - ▶Si se ejecutan en host diferentes (con sistemas operativos potencialmente diferentes), se comunican entre sí intercambiando mensajes a través de la red.
- ▶**Un proceso emisor** crea y envía mensajes a la red.
- ▶**Un proceso receptor** recibe estos mensajes y responde devolviendo otros.
- **La figura ilustra** que los procesos que se comunican entre sí residen en la capa de aplicación de la pila de protocolos de cinco capas.

PROTOCOLOS DE TRANSPORTE

Interfaz entre el proceso y la red

(Kurose, 2017)



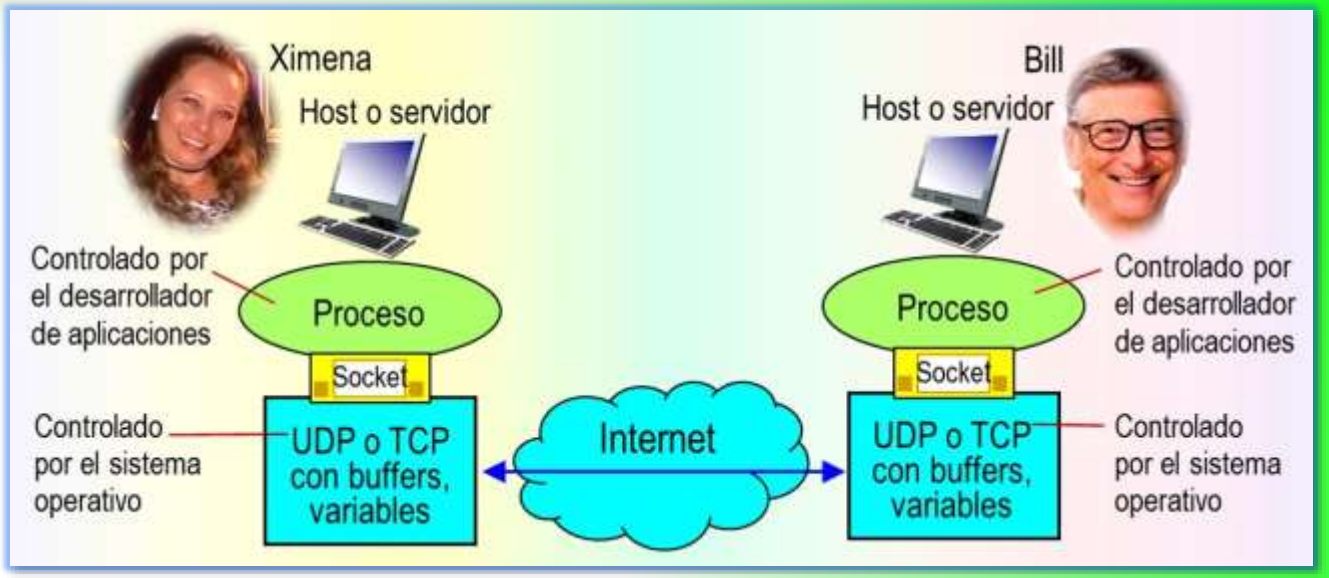
- **La mayoría de las aplicaciones** constan de parejas de procesos (cliente – servidor) que se comunican intercambiando mensajes.
- **Cualquier mensaje** enviado de un proceso al otro debe atravesar la red subyacente.
- **Un proceso** envía mensajes a la red y los recibe de la red a través una interfaz software denominada socket (enchufe).
- **¿Qué es un socket?** Una analogía para comprender el concepto de socket asociado a un proceso es la siguiente:
  - ▶Un proceso es análogo a una casa y un socket a la puerta de la casa. Cuando un proceso desea enviar un mensaje a otro que se ejecuta en otro host, envía el mensaje a través de su propia puerta (socket).
  - ▶El proceso emisor supone que existe una infraestructura de transporte al otro lado de la puerta que llevará el mensaje hasta la puerta del proceso de destino.
  - ▶Una vez que el mensaje llega al host de destino, éste pasa a través de la puerta (socket) del proceso receptor, actuando entonces el proceso receptor sobre el mensaje.

Comunicación entre procesos

PROTOCOLOS DE TRANSPORTE

Comunicación entre sockets

(Kurose, 2017)



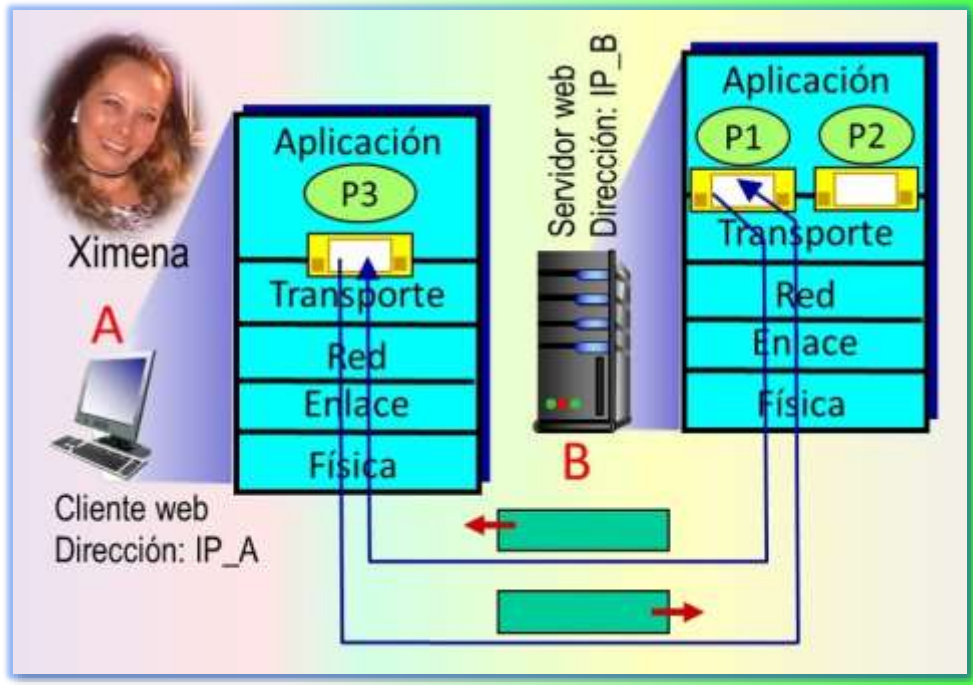
- **En la figura**, se ilustra la comunicación mediante sockets entre dos procesos que se comunican a través de Internet.
- **El protocolo de transporte** subyacente utilizado por los procesos puede ser UDP o TCP.
- **Un socket** es la interfaz entre la capa de aplicación y la de transporte de un host.
- **También se conoce** como Interfaz de programación de aplicaciones API entre la aplicación y la red, ya que el socket es la interfaz de programación con la que se construyen las aplicaciones de red.
  - ✉ Por ejemplo, un socket UDP se crea mediante la línea de código del programa Python:
    `clientsocket=socket(AF_INET,SOCKET_DGRAM)`
- **El desarrollador de la aplicación** tiene el control de todos los elementos del lado de la capa de aplicación del socket, pero no de los del lado de la capa de transporte, a excepción de:
  - ▶(1) La elección del protocolo de transporte que puede ser UDP o TCP y
  - ▶(2) Quizás la capacidad de fijar unos pocos parámetros, como los tamaños máximos del buffer y de segmento.
- **Una vez** que el desarrollador ha seleccionado un protocolo de transporte, la aplicación se construye utilizando los servicios de la capa de transporte proporcionado por dicho protocolo.

PROTOCOLOS DE TRANSPORTE

Direccionamiento de procesos

(Kurose, 2017)

- **Para enviar una carta** a un destino concreto, se necesita una dirección. De forma similar, para que un proceso que se está ejecutando en un host pueda enviar paquetes a otro proceso que se ejecuta en un host distinto, se necesita de una dirección del proceso receptor.
- **Para identificar** al proceso receptor se necesitan dos elementos de información:
  - ▶(1) La dirección IP del host.
  - ▶(2) Un identificador, llamado número de puerto, que especifique el proceso de recepción en el host de destino.
- **Por tanto,** además de conocer la dirección IP del host receptor, el host emisor también debe identificar al proceso receptor (o, para ser mas exactos, al socket receptor) que está ejecutándose en el host.
- **Esta información** es necesaria porque, en general, un host podría estar ejecutando varias aplicaciones de red a la vez.

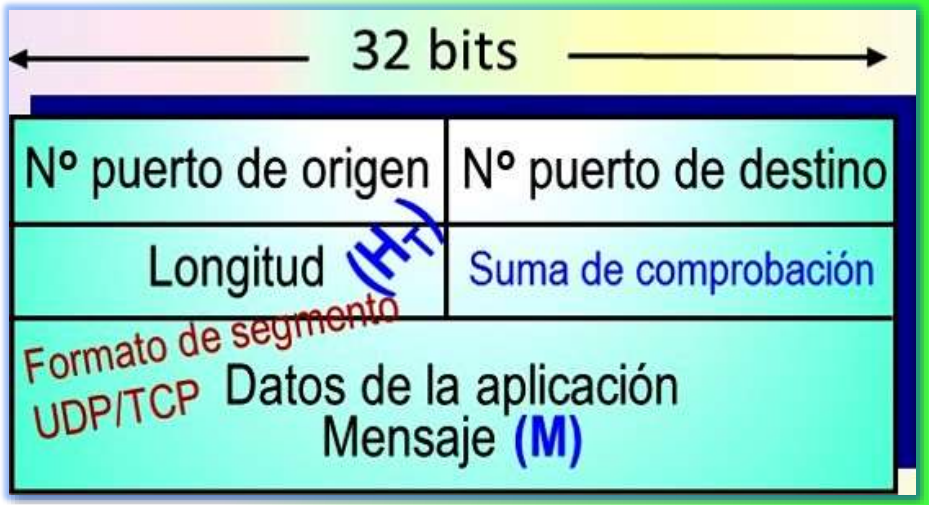


PROTOCOLOS DE TRANSPORTE

Números de puerto

(Kurose, 2017)

- **Cada segmento de transporte** tiene en su cabecera campos que identifican a un proceso, o más exactamente al socket al que tiene que entregarse el segmento (H_TM). Estos campos son el número de puerto de origen y el número de puerto de destino (ver el formato de segmento UDP/TCP de la figura).



- **Cada número de puerto** es un número en el rango de 0 a 65535 (16 bits).
  - ⊠Los números de puerto pertenecientes al rango de 0 a 1023 se conocen como números de puertos bien conocidos y son restringidos, lo que significa que están reservados para ser empleados por los protocolos de aplicación bien conocidos.
  - ⊠Por ejemplo HTTP utiliza el puerto 80, FTP utiliza el número de puerto 21, el proceso de servidor de correo SMTP se identifica mediante el puerto 25.
  - ⊠Al desarrollar una nueva aplicación, es necesario asignar un número de puerto a la aplicación.

PROTOCOLOS DE TRANSPORTE

Algunos números de puerto

(CISCO, 2016)

- **La tabla** muestra el número de puerto de los protocolos de aplicación bien conocidos y el protocolo de transporte subyacente.
  - ✉ Por ejemplo, el protocolo HTTP de la aplicación web utiliza los servicios del protocolo de transporte TCP, y el número de puerto del proceso es 80.

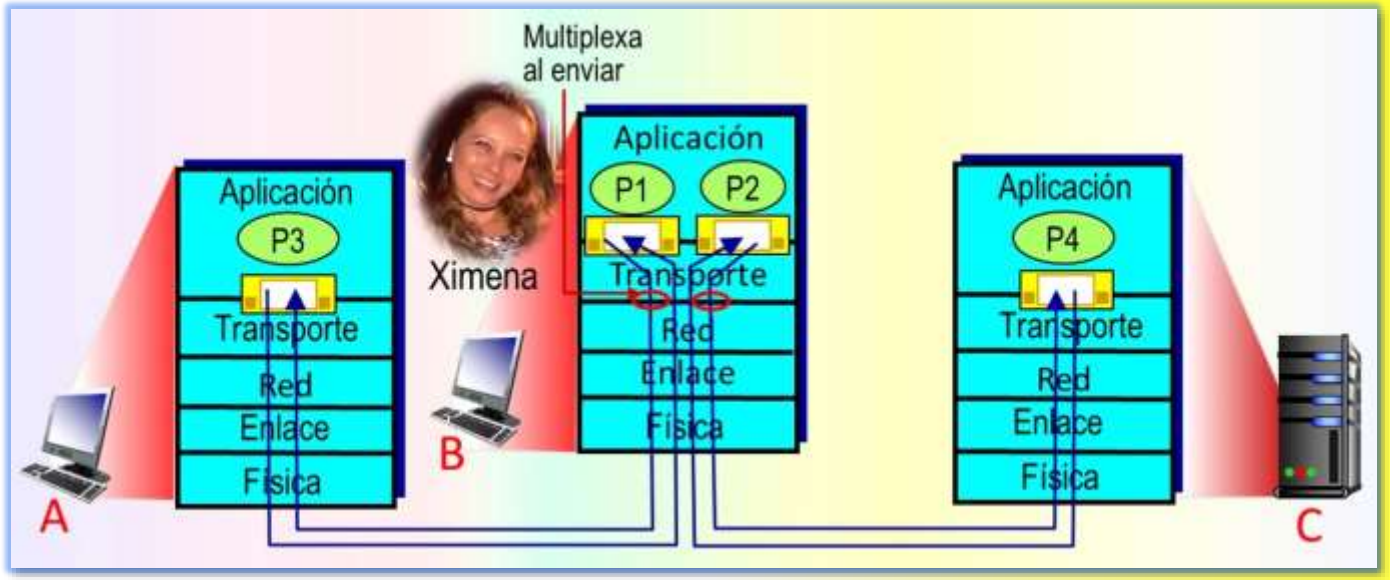
Número de puerto	Protocolo	Aplicación	Acrónimo
20	TCP	Protocolo de transferencia de archivos (datos)	FTP
21	TCP	Protocolo de transferencia de archivos (control)	FTP
22	TCP	Shell Seguro	SSH
23	TCP	Telnet	–
25	TCP	otocolo simple de transferencia de correo (Simple Mail Transfer Protocol)	SMTP
53	UDP, TCP	Servicio de nombres de dominios	DNS
67	UDP	Protocolo de configuración dinámica de host (servidor)	DHCP
68	UDP	Protocolo de configuración dinámica de host (cliente)	DHCP
69	UDP	Protocolo de transferencia de archivos trivial	TFTP
80	TCP	Protocolo de transferencia de hipertexto	HTTP
110	TCP	Protocolo de oficina de correos versión 3 (Post Office Protocol version 3)	POP3
143	TCP	Protocolo de acceso a mensajes de Internet (Internet Message Access Protocol)	IMAP
161	UDP	Simple Network Management Protocol	SNMP
443	TCP	Protocolo seguro de transferencia de hipertexto	HTTPS

PROTOCOLOS DE TRANSPORTE

¿En qué consiste la multiplexación?

- **Escenario**. Un escenario muy común es aquel en el que, por ejemplo, Ximena está frente a su PC y está descargando páginas web a la vez que ejecuta una sesión FTP y dos sesiones Telnet.
  - ✉Por tanto, tiene: cuatro procesos de aplicación en ejecución: dos procesos Telnet, un proceso FTP y un proceso HTTP.
- **Para explicar** el concepto de multiplexación, solo se considerará un proceso de aplicación y uno de Telnet.

(Kurose, 2017)



- ▶**1.Cada proceso** tiene un socket, por el que pasan los datos del proceso a la red.
  - ✉En este ejemplo, la capa de aplicación del host B de Ximena (como host emisor) entrega los datos de los procesos P1 y P2 directamente a los sockets correspondientes.
- ▶**2.Multiplexación** se denomina a la tarea que realiza la capa de transporte de reunir los fragmentos de datos (M) desde los diferentes sockets, de encapsularlos con la información de cabecera para crear los segmentos (H_TM) y de pasarlos a la capa de red.
  - ✉En este ejemplo, la capa de transporte del host B de Ximena recopila los datos salientes desde los sockets de los procesos P1 y P2, construye (multiplexa) los segmentos de transporte y los pasa a la capa de red con destino a los procesos P3 y P4 respectivamente.

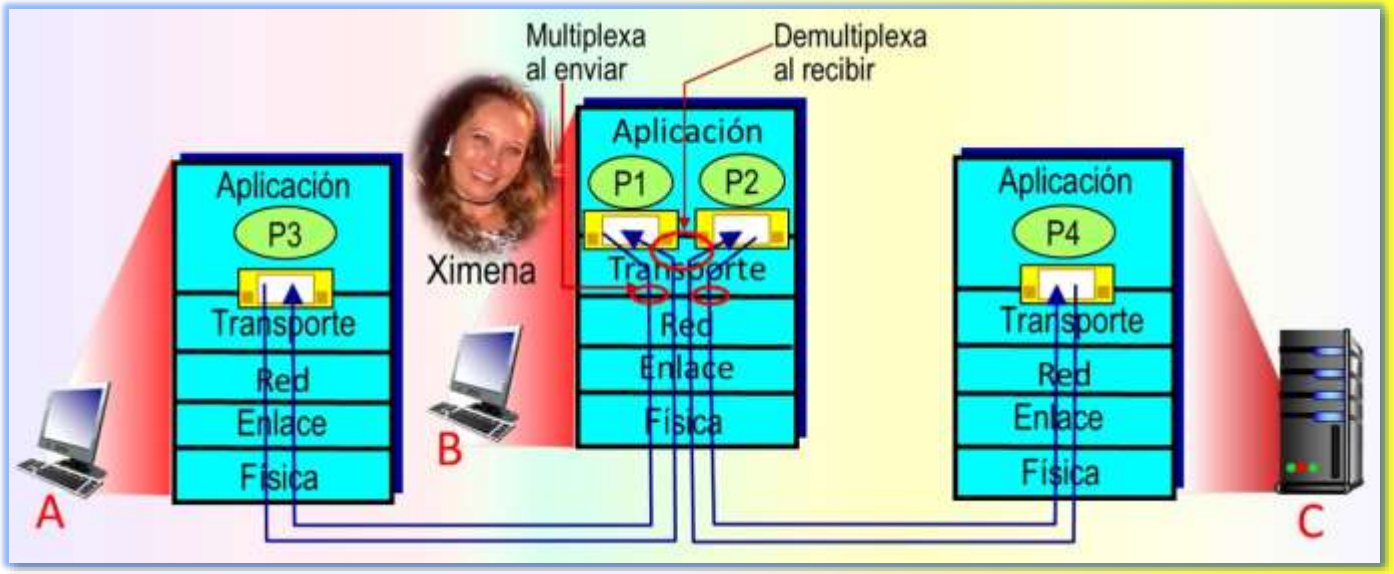
Multiplexación en la capa de transporte

PROTOCOLOS DE TRANSPORTE

¿En qué consiste la demultiplexación?

(Kurose, 2017)

- ▶**1.Cada proceso** tiene un socket, por el que pasan los datos de la red al proceso.
  - ✉En este ejemplo, la capa de transporte del host B de Ximena (como host receptor) no entrega los datos directamente a los procesos P1 y P2, sino a sus sockets correspondientes.
- ▶**2.Demultiplexación** se denomina a la tarea que realiza la capa de transporte de entregar los datos contenidos en los segmento de la capa de transporte a los sockets correctos.

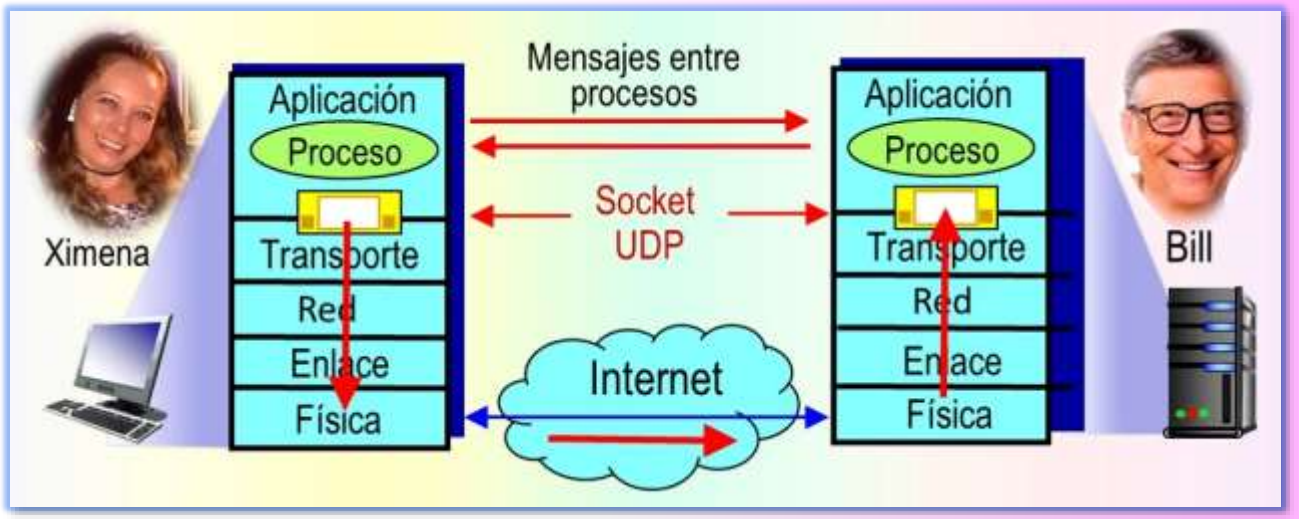


- Los segmentos de transporte (H_TM) que llegan son dirigidos a los socket apropiados, gracias a que cada segmento contiene los números de puerto de destino para identificar a los sockets receptores.
- El formato de este identificador depende de si se trata de un socket UDP o de un TCP.
- ✉En este ejemplo, la capa de transporte del host B de Ximena (como host receptor) examina los números de puerto contenidos en los segmentos que llegan desde los procesos P3 y P4 hacia P1 y P2 respetivamente y, a continuación, envía los segmento a los socket correspondientes.

PROTOCOLOS DE TRANSPORTE

Modelo de servicio proporcionado por UDP

(Kurose, 2017)



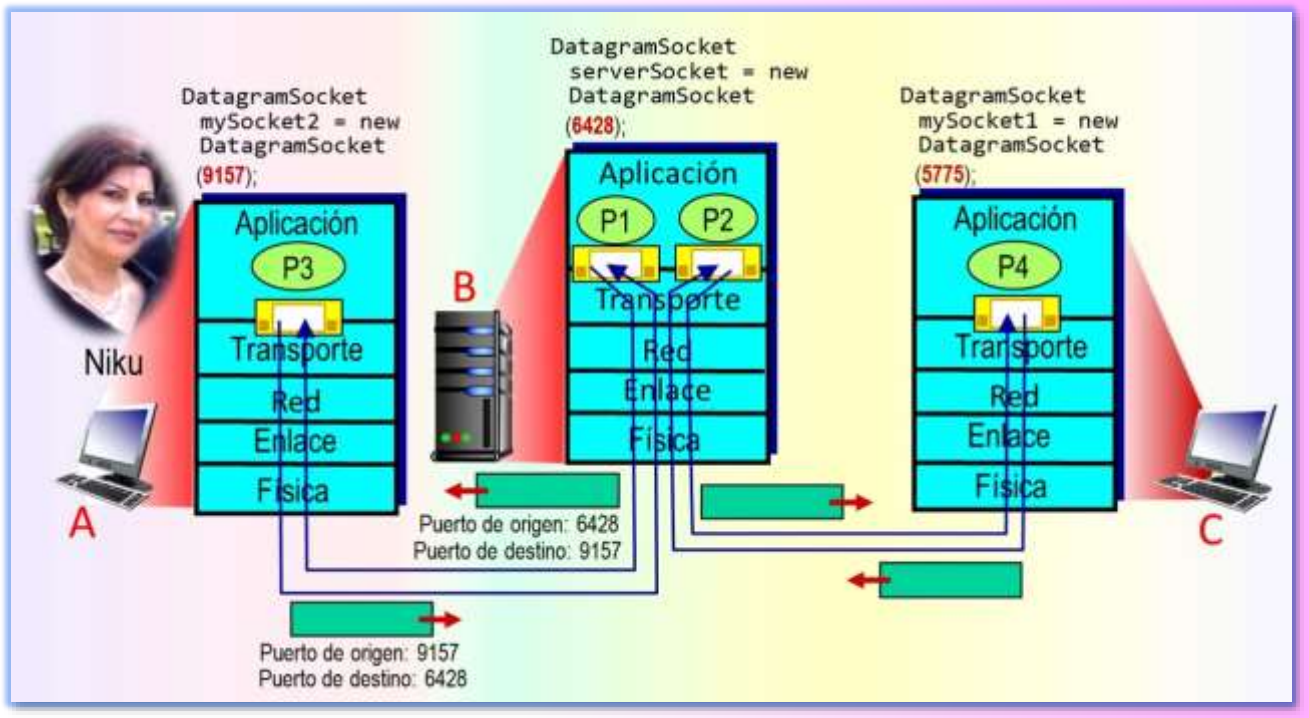
- **La responsabilidad principal** de UDP es ampliar el servicio de entrega de IP entre dos hosts a un servicio de entrega entre dos procesos que estén ejecutándose en los hosts.
- **Extender la entrega** host a host a una entrega proceso a proceso es lo que se denomina multiplexación y demultiplexación de la capa de transporte.
- **Los dos servicios básicos** de la capa de transporte ofrecidos por UDP son:
  - ▶**1.** Entrega de datos proceso a proceso.
  - ▶**2.** Comprobación de la integridad de los datos al incluir campos de detección de errores en las cabeceras de los segmentos. Suma de comprobación de Internet.
- **En particular**, al igual que IP, UDP es un servicio no fiable, que no garantiza que los datos enviados por un proceso lleguen intactos al proceso de destino.
- Nota.- El tráfico UDP no está regulado. Una aplicación que emplee el protocolo UDP puede enviar los datos a la velocidad que le parezca, durante todo el tiempo que quiera.

Transporte de datos con UDP

PROTOCOLOS DE TRANSPORTE

UDP Transporte sin conexión

(Kurose, 2017)



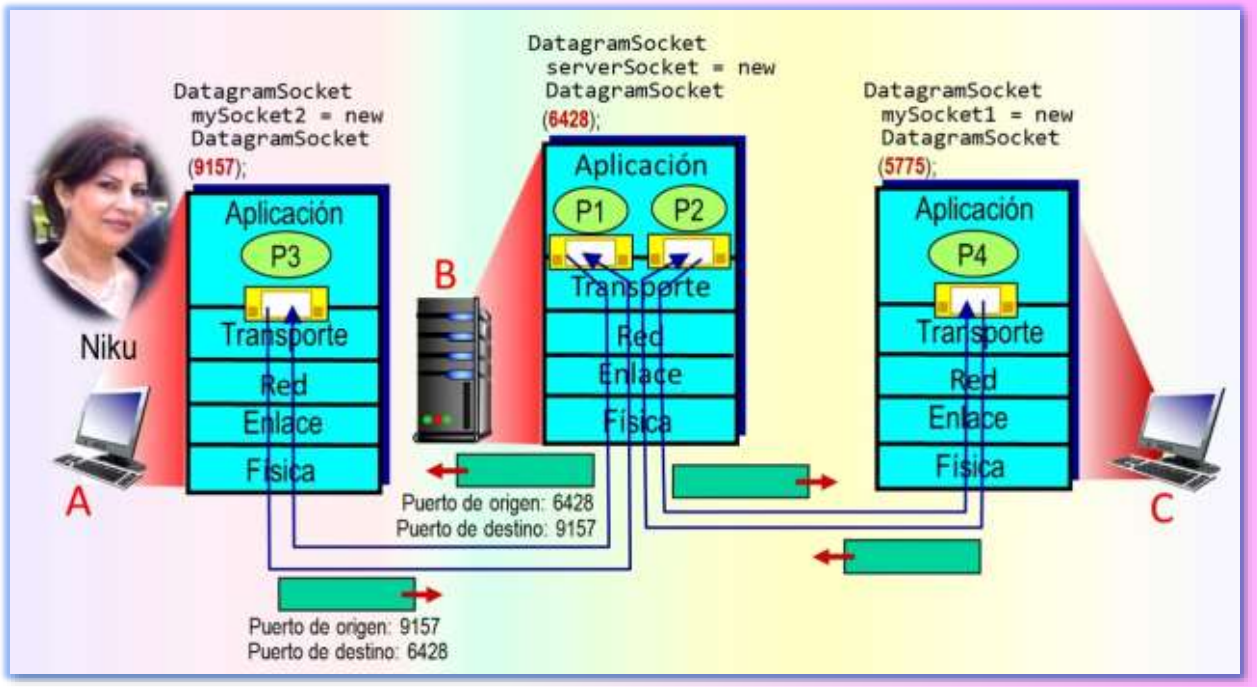
- ▶**1. Suponga** que el proceso P3 del host A de Niku desea enviar un fragmento de datos de una aplicación al proceso P1 del servidor B, para ello crea un socket UDP mediante la línea de código del programa Python: `clientSocket=socket(AF_INET,SOCK_DGRAM).`
- ▶**2. La capa de transporte** asigna automáticamente un número de puerto al socket UDP, comprendido en el rango 1024 a 65535 que actualmente no esté siendo utilizado en ese host por ningún otro puerto.
  - ✉Suponga que al proceso del host A de Niku se le asigna el puerto UDP 9157, y al proceso del servidor B el puerto UDP 6428.
  - ✉Si el desarrollador de la aplicación que escribe el código estuviera implementando el lado del servidor de un protocolo bien conocido, entonces tendría que asignar el correspondiente número de puerto bien conocido.
- ▶**3. La capa de transporte** del host A de Niku crea un segmento ($H_T M$) que incluye los datos de aplicación, el número de puerto de origen 9157 y el número de puerto de destino 6428.

Transporte de datos con UDP

PROTOCOLOS DE TRANSPORTE

UDP transporte sin conexión (cont.)

(Kurose, 2017)

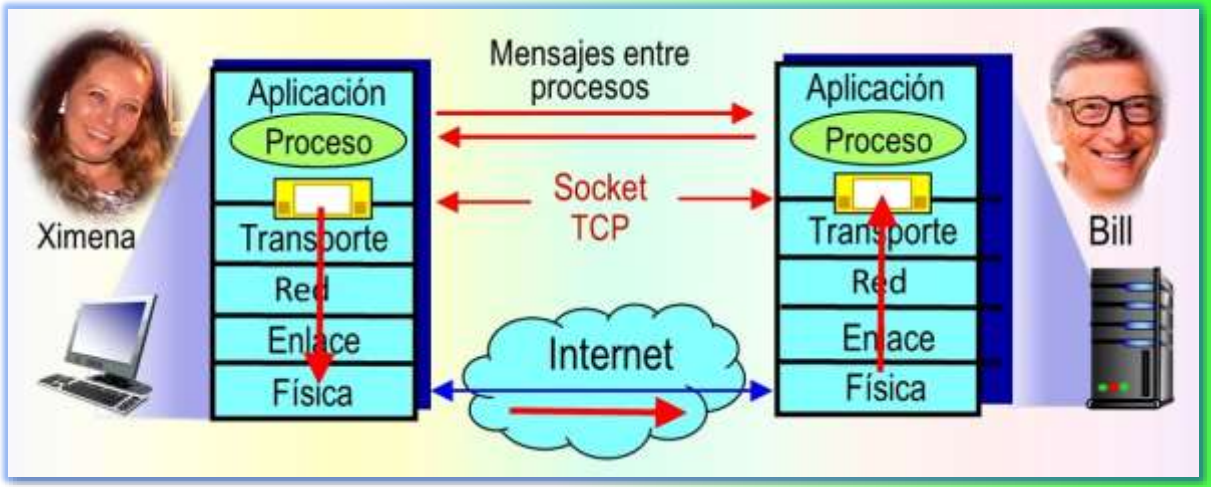


- ▶**4. La capa de transporte** en el host A de Niku pasa a continuación el segmento (H_TM) resultante a la capa de red. La capa de red encapsula dicho segmento en un datagrama IP (H_NH_TM) y hace el máximo esfuerzo por entregar el segmento al servidor receptor B.
- ▶**5. Si el segmento (H_TM)** llega al servidor receptor B, la capa de transporte examina el número de puerto de destino especificado en el segmento (6428) y entrega el segmento a su socket identificado por el puerto 6428.
  - ✉Observe que el servidor B podría estar ejecutando varios procesos, cada uno de ellos con su propio socket UDP y número de puerto asociado.
  - ✉A medida que los segmentos UDP (H_TM) llegan de la red, el servidor B dirige (demultiplexa) cada segmento al socket apropiado examinando el número de puerto de destino del segmento.
- **Es importante** observar que un socket UDP queda completamente identificado por una dupla que consta de una dirección IP de destino y un número de puerto de destino.
  - ✉En consecuencia, si dos segmentos UDP tienen diferentes direcciones IP y/o número de puerto de origen, pero la misma dirección IP de destino y el mismo número de puerto de destino, entonces los dos segmentos se enviaran al mismo proceso de destino a través del mismo socket de destino.

PROTOCOLOS DE TRANSPORTE

Modelo de servicio proporcionado por TCP

(Kurose, 2017)



- **La responsabilidad principal** de TCP es ampliar el servicio de entrega de IP entre dos hosts a un servicio de entrega entre dos procesos que estén ejecutándose en los hosts.
- **TCP ofrece** a las aplicaciones varios servicios adicionales a los básicos que entrega UDP:
  - ▶**1.** Entrega de datos proceso a proceso.
  - ▶**2.** Comprobación de la integridad de los datos al incluir campos de detección de errores en las cabeceras de los segmentos. Suma de comprobación de Internet.
  - ▶**3.** Transferencia de datos fiable. Para ello, utiliza técnicas de control de flujo, número de secuencia, mensajes de reconocimiento y temporizadores, que garantizan que los datos transmitidos por el proceso emisor sean entregados al proceso receptor, correctamente y en orden.
    - ✉ De este modo, TCP convierte el servicio no fiable de IP entre hosts en un servicio de transporte de datos fiable entre procesos.
  - ▶**4.** Mecanismos de control de congestión, que no es un servicio proporcionado a la aplicación invocante, sino un servicio que se presta a Internet como un todo. Los mecanismos de control de congestión de TCP evitan que cualquier conexión inunde con una cantidad tráfico excesiva los enlaces y routers existentes entre los hosts que se están comunicando.
    - ✉ TCP se esfuerza en proporcionar a cada conexión que atraviesa un enlace congestionado, la misma cuota de ancho de banda. Esto se consigue regulando la velocidad a la que los emisores de las conexiones TCP pueden enviar tráfico a la red.

PROTOCOLOS DE TRANSPORTE

Establecimiento de una conexión TCP

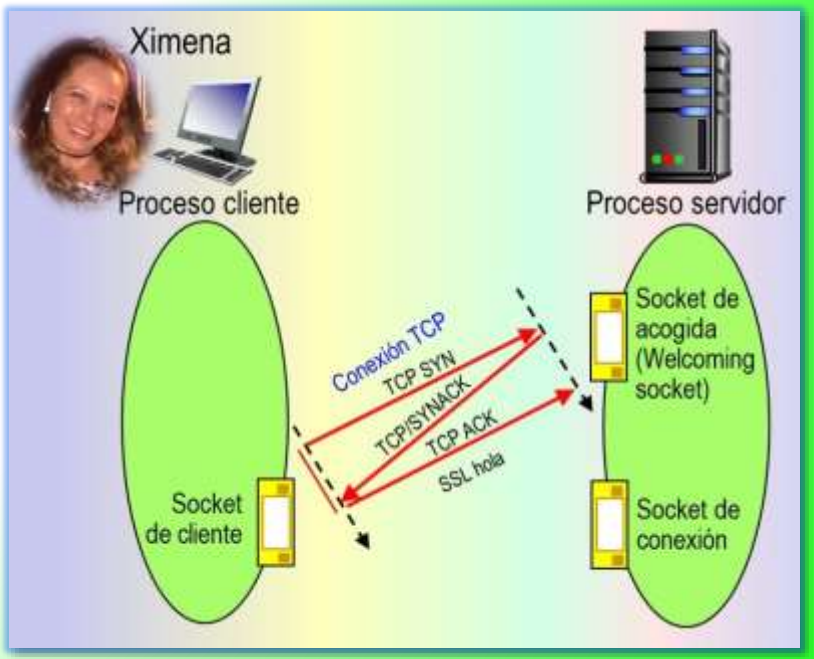
(Kurose, 2017)

- **Una sutil diferencia** entre un socket TCP y un socket UDP, es que en TCP se establecen conexiones identificadas por un conjunto de cuatro elementos: **(1)** dirección IP de origen, **(2)** número de puerto de origen, **(3)** dirección IP de destino, **(4)** número de puerto de destino. El establecimiento de la conexión tiene las siguientes fases.
- **Por tanto,** cuando un segmento TCP **(H_TM)** llega a un host procedente de la red, el host emplea los cuatro valores para dirigir (demultiplexar) el segmento al socket apropiado. En UDP, solo emplea dos, la dirección IP de destino y el número de puerto de destino.
- **▶1. Proceso de conexión**. La aplicación del servidor TCP tiene un "socket de acogida", que está a la espera de solicitudes de conexión procedentes de clientes TCP en el puerto 12000. El cliente TCP crea un socket, que incluye su puerto origen, y envía un segmento de establecimiento de conexión con las líneas de código:

```
clientSocket=socket(AF_INET,SOCKET_DGRAM)
.....clientSocket.connect((serverName,12000))
```

- **▶2. Creación del socket de conexión**. El proceso servidor, que recibe la solicitud de conexión, con el puerto 12.000, crea un nuevo socket, el socket de conexión.

```
connectionlionSocket, addr=serverSocket.accept()
```

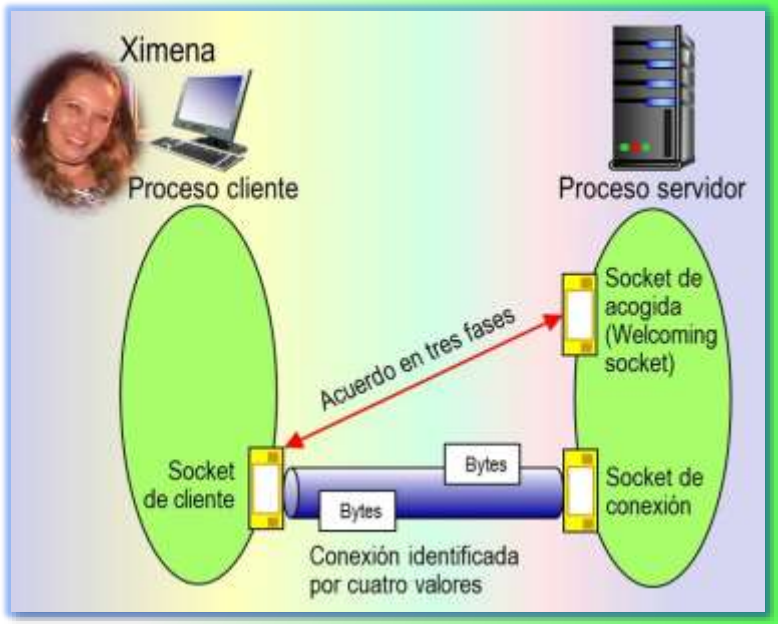


PROTOCOLOS DE TRANSPORTE

Establecimiento de una conexión TCP (cont.)

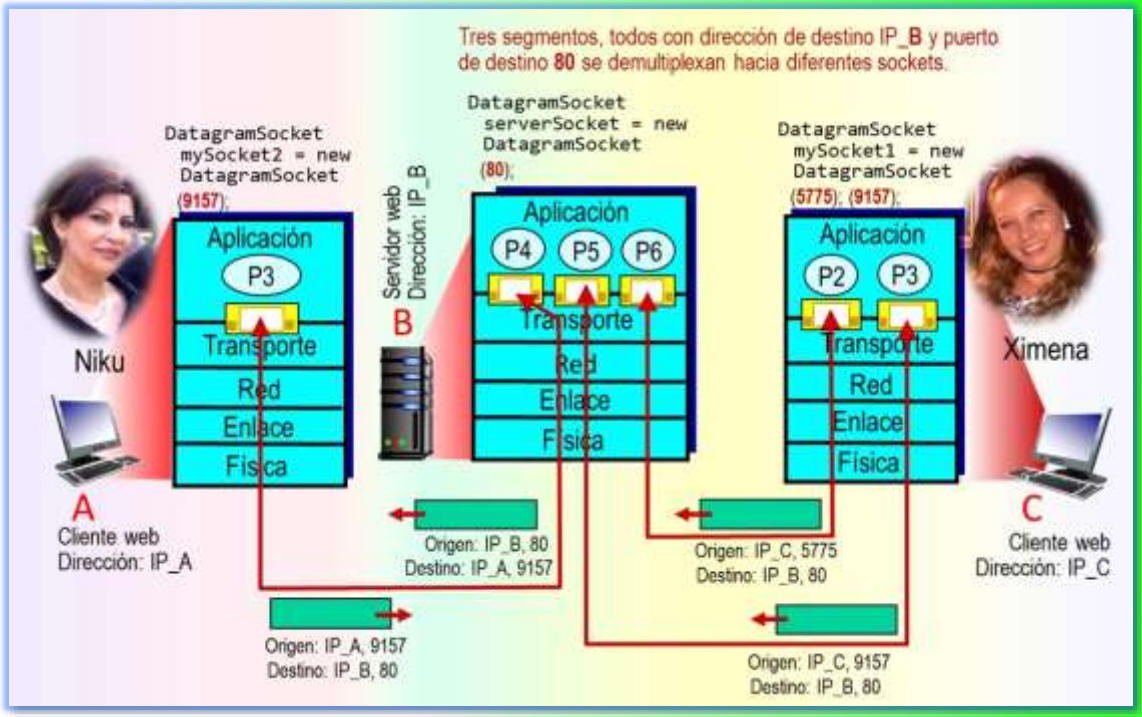
(Kurose, 2017)

- ▶**3.** Registro de identificadores de la conexión. A continuación, la capa de transporte en el servidor toma nota de los cuatro indicadores contenidos en el segmento de solicitud de conexión:
  - ▶ **(1)** La dirección IP del host de origen.
  - ▶ **(2)** El número de puerto de origen.
  - ▶ **(3)** La dirección IP del servidor de destino.
  - ▶ **(4)** El número de puerto de destino.
- ▶**4. Envío de datos**. El socket de conexión recién creado queda, entonces, identificado por cuatro valores. Así, todos los segmentos que lleguen y cuyo puerto de origen, dirección IP de origen, puerto de destino y dirección IP de destino se correspondan con estos cuatro valores serán enviados a este socket.
  - ✉Una vez establecida la conexión TCP, el cliente y el servidor podrán enviarse datos entre sí.



PROTOCOLOS DE TRANSPORTE

TCP transporte orientado a la conexión (Kurose, 2017)

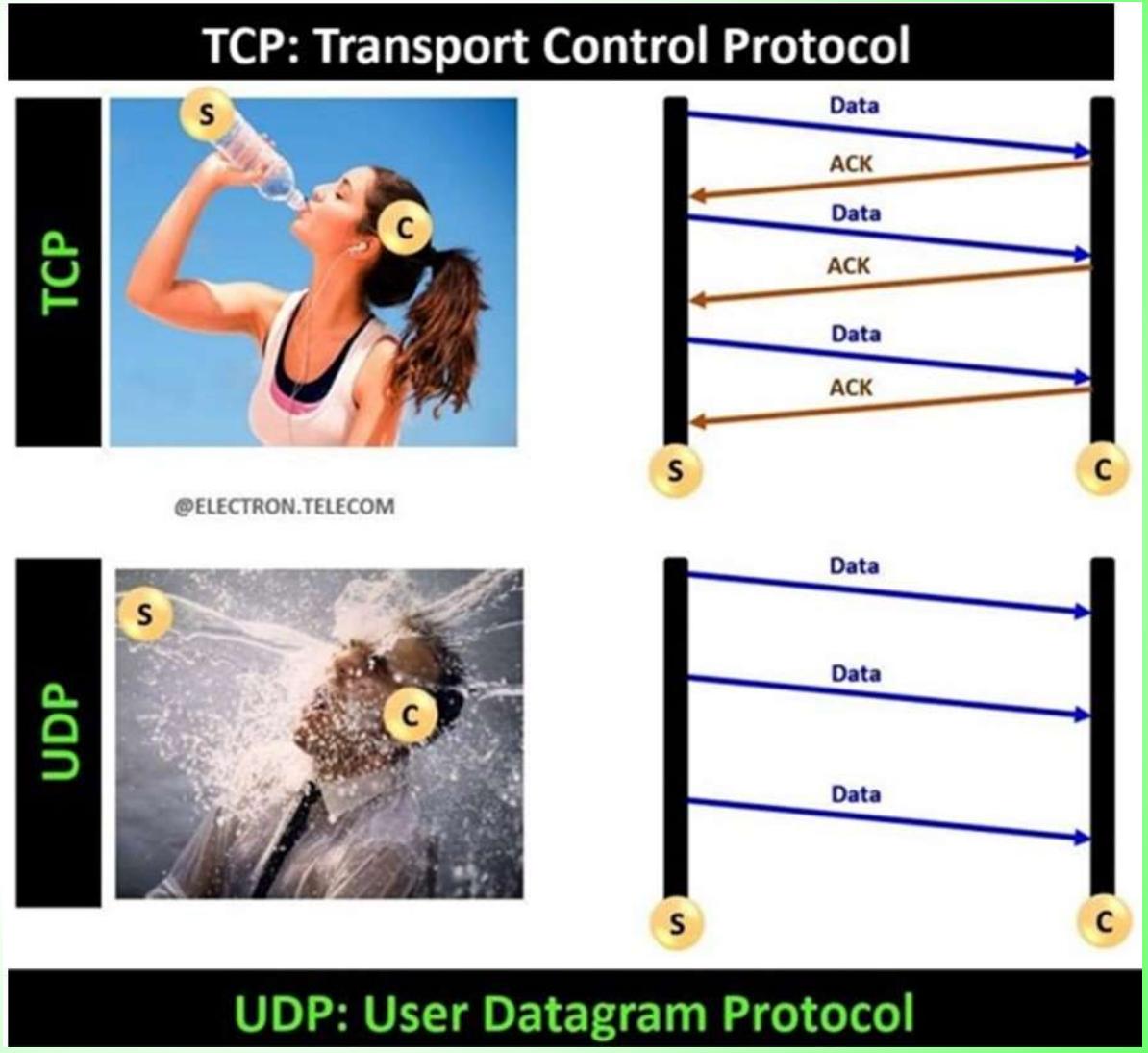


- **Ejemplo de conexión TCP**
- ▶1. El host servidor B puede dar soporte a muchos sockets TCP simultáneos, estando cada socket asociado a un proceso y cada socket identificado por su conjunto de cuatro elementos.
- ▶**2.** Cuando un segmento TCP ($H_T M$) llega al host servidor B, los cuatro campos (dirección IP de origen, puerto de origen, dirección IP de destino y puerto de destino) se utilizan para dirigir (demultiplexar) el segmento al socket apropiado. Esta situación se ilustra en la figura, en la que:
  - ✉Los hosts A y C y el servidor B tienen sus direcciones IP.
  - ✉El host A de Niku inicia una sesión HTTP con el servidor B (puerto 80) y asigna el número de puerto de origen 9157 a su conexión HTTP.
  - ✉El host C de Ximena inicia dos sesiones HTTP con el servidor B (puerto 80) y asigna dos números de puertos de origen diferentes (5775 y 9157) a sus dos conexiones HTTP.
- ▶**3.** Dado que el host A de Niku selecciona los números de puerto de origen independientemente de C, también puede asignar el número de puerto de origen 9157 a su conexión HTTP. Pero esto no es un problema: el servidor B todavía podrá demultiplexar correctamente las dos conexiones con el mismo número de puerto de origen, ya que tienen direcciones IP de origen diferentes.

Transporte de datos con TCP

PROTOCOLOS DE TRANSPORTE

Comparación entre TCP y UDP



PROTOCOLOS DE TRANSPORTE

Servicios que puede ofrecer un protocolo de la capa de transporte

(Kurose, 2017)

- **Muchas redes**, incluyendo Internet, proporcionan más de un protocolo de la capa de transporte. Cuando se desarrolla una aplicación, se tiene que elegir uno de los protocolos de la capa de transporte disponibles.
- **¿Cómo llevar** a cabo esta selección? Se tendrá elegir aquel protocolo que proporcione los servicios que mejor se adapten a las necesidades de la aplicación.
- **¿Cuáles son los servicios** que puede ofrecer un protocolo de la capa de transporte a las aplicaciones que lo invocan? Se pueden clasificar de forma bastante general según cuatro parámetros: transferencia de datos fiable, tasa de transferencia, temporización y seguridad.
  - ▶**1.** Transferencia de datos fiables. En una red de computadoras pueden perderse paquetes. Por ejemplo, un paquete puede desbordar el buffer de un router, o podría ser descartado por un host o un router después de comprobar que algunos de sus bits están corrompidos. En muchas aplicaciones (como el correo electrónico, la transferencia de archivos, el acceso remoto a hosts, la transferencia de documentos web y las aplicaciones financieras) la pérdida de datos puede tener consecuencias catastróficas.
    - ✉ Por tanto, para dar soporte a estas aplicaciones, es preciso hacer algo para garantizar que los datos enviados desde un terminal de la aplicación sean todos ellos entregados correcta y completamente al otro terminal de la aplicación.
    - ✉ Si un protocolo proporciona un servicio de entrega de datos garantizado, se dice que proporciona una transferencia de datos fiable. Un servicio importante que un protocolo de la capa de transporte puede potencialmente proporcionar a una aplicación es la transferencia de datos fiable proceso a proceso, es decir, de socket a socket.

PROTOCOLOS DE TRANSPORTE

Servicios que puede ofrecer un protocolo de la capa de transporte (cont.)

(Kurose, 2017)

- ▶**2.** Tasa de transferencia. El concepto de tasa de transferencia disponible, es la tasa a la que el proceso emisor puede suministrar bits al proceso de recepción a lo largo de una ruta de red.
  - ✉ Puesto que otras sesiones compartirán el ancho de banda a lo largo de la ruta de red y puesto que esas otras sesiones se iniciarán y terminarán aleatoriamente, la tasa de transferencia disponible puede fluctuar con el tiempo. Un protocolo que pueda proporcionar una tasa de transferencia disponible garantizada a una cierta velocidad especificada, resulta atractivo para muchas aplicaciones.
  - ✉ Por ejemplo, si una aplicación de telefonía por Internet codifica voz a 32 kbps, tendrá que enviar datos a la red y tendrá que entregar los datos a la aplicación receptora a esa velocidad. Si el protocolo de transporte no puede proporcionar esa tasa de transferencia, la aplicación tendrá que realizar la codificación a una velocidad menor o bien tendrá que renunciar, puesto que recibir a la mitad de la tasa de transferencia necesaria no tiene ninguna utilidad para esta aplicación de telefonía por Internet.
- ▶**3.** Temporización.- Un protocolo de la capa de transporte también puede proporcionar garantías de temporización. Al igual que con las tasas de transferencia garantizadas, las garantías de temporización también pueden darse de diversas formas. Un ejemplo de garantía podría ser que cada bit que el emisor empuja por su socket llegue al socket del receptor en no más de 100 milisegundos.
  - ✉ Un servicio así sería atractivo para las aplicaciones interactivas en tiempo real, como la telefonía por Internet, los entornos virtuales, la teleconferencia y los juegos multijugador, todas las cuales requieren restricciones de temporización muy estrictas sobre la entrega de datos para ser efectivas.

PROTOCOLOS DE TRANSPORTE

Servicios que puede ofrecer un protocolo de la capa de transporte (cont.)

(Kurose, 2017)

- ▶**4.** Seguridad.- Un protocolo de transporte puede proporcionar a una aplicación uno o más servicios de seguridad. Por ejemplo, en el host emisor, un protocolo de transporte puede cifrar todos los datos transmitidos por el proceso emisor, y en el host receptor puede descifrar los datos antes de entregarlos al proceso receptor
  - ✉ Un servicio así proporcionaría confidencialidad entre los dos procesos, incluso aunque los datos sean observados por intrusos de alguna manera entre los procesos emisor y receptor.
  - ✉ Un protocolo de transporte también puede proporcionar otros servicios de seguridad además del de la confidencialidad, como pueden ser mecanismos para garantizar la integridad de los datos y mecanismos de autenticación en el punto terminal.
- **En conclusión**, la primera decisión que tendrá que tomar un desarrollador de aplicaciones que crea una nueva aplicación de red para Internet, es si utilizar UDP o TCP. Cada uno de estos protocolos ofrece un conjunto diferente de servicios a las aplicaciones que los invocan.

PROTOCOLOS DE TRANSPORTE

(Kurose, 2017)

Relación entre la capa de transporte y la de red

- **Mientras** que un protocolo de la capa de transporte proporciona una conexión lógica entre procesos que se ejecutan en hosts diferentes, un protocolo de la capa de red proporciona una comunicación lógica entre hosts.
- **El protocolo** de la capa de red de Internet es el IP que proporciona una comunicación lógica entre hosts. El modelo de servicio de IP es un servicio de entrega de mejor esfuerzo. Esto quiere decir que IP hace todo lo que puede por entregar los segmentos entre los host que se están comunicando, pero no garantiza la entrega. En particular, no garantiza la entrega de los segmentos, no garantiza que los segmentos se entreguen en orden y no garantiza la integridad de los datos contenidos en los segmentos. Por estas razones, se dice que IP es un servicio no fiable.
- **Internet** pone a disposición de la capa de aplicación dos protocolos de la capa de transporte diferentes.
  - ▶Uno de ellos es el Protocolo de datagramas de usuario **UDP**, que proporciona a la aplicación que lo invoca un servicio sin conexión no fiable.
  - ▶El otro es el Protocolo de control de transmisión **TCP**, que proporciona a la aplicación que lo invoca un servicio orientado a la conexión fiable.
- **Al diseñar** una aplicación de red, el desarrollador tiene que especificar el uso de uno de estos dos protocolos de transporte, cuando crea los **sockets**.

PROTOCOLOS DE TRANSPORTE

Elección del protocolo de transporte

(Kurose, 2017)

- **La tabla** muestra algunas aplicaciones populares de Internet, sus protocolos de la capa de aplicación y sus protocolos de transporte subyacentes.
- **Para el correo electrónico**, el acceso remoto a terminales, la Web y la transferencia de archivos se ha elegido TCP porque ofrece un servicio de transferencia de datos fiable, garantizando que todos los datos lleguen finalmente a su destino.
- **La telefonía por Internet** (como Skype, WhatsApp) suele tolerar cierto grado de pérdidas, pero requiere una tasa de transferencia mínima para ser efectivas, los desarrolladores de aplicaciones suelen preferir ejecutarlas sobre UDP, evitando así el mecanismo de control de congestión de TCP y la mayor sobrecarga (bits que no forman parte de la carga útil) que los paquetes de datos tienen en TCP.

Selección del protocolo de transporte

Aplicación	Protocolo de la capa de aplicación	Protocolo de transporte subyacente
Correo electrónico	SMTP (RFC 5321)	TCP
Acceso remoto a terminal	Telnet (RFC 854)	TCP
Web	HTTP (RFC 2616)	TCP
Transferencia de archivos	FTP (RFC 959)	TCP
Flujos multimedia	HTTP (p.ej. YouTube)	TCP
Telefonía por Internet	SIP (RFC 3261), RTP (RFC 3550) o propietario (Skype, WhatsApp)	UDP o TCP

- ✉ Pero como muchos firewalls están configurados para bloquear el tráfico UDP (o la mayor parte del mismo), las aplicaciones de telefonía por Internet suelen diseñarse para usar TCP como solución alternativa, cuando falla la comunicación a través de UDP.

Referencias bibliográficas

PROTOCOLOS DE TRANSPORTE

Referencias bibliográficas

- CISCO (2015). *CCNA Routing and Switching. Introduction to Networks*. CISCO.
- CISCO (2016). *Introducción a las redes.* Madrid: Pearson Education, S.A.
- Forouzan, B. A. (2007). *Transmisión de datos y redes de comunicaciones.* Madrid: McGraw-Hill.
- Huawei Technologies (2020). *Basics of data communication networks*. Huawei.
- Kurose, J. Keith, R. (2017). *Redes de computadoras: un enfoque descendente*. Madrid: Pearson Education, S.A.

FIN

Tema 7 de:

PROTOCOLOS DE INTERNET

Edison Coimbra G.