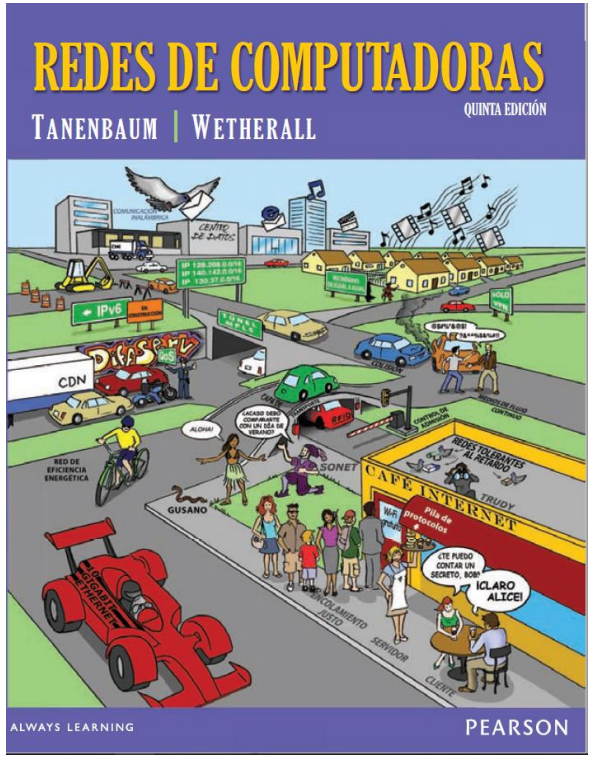
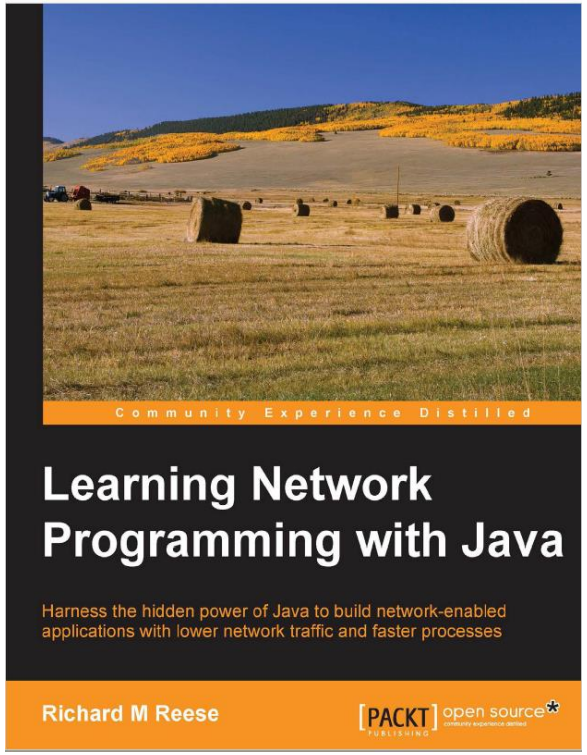
REDES DE COMPUTADORES Y LABORATORIO

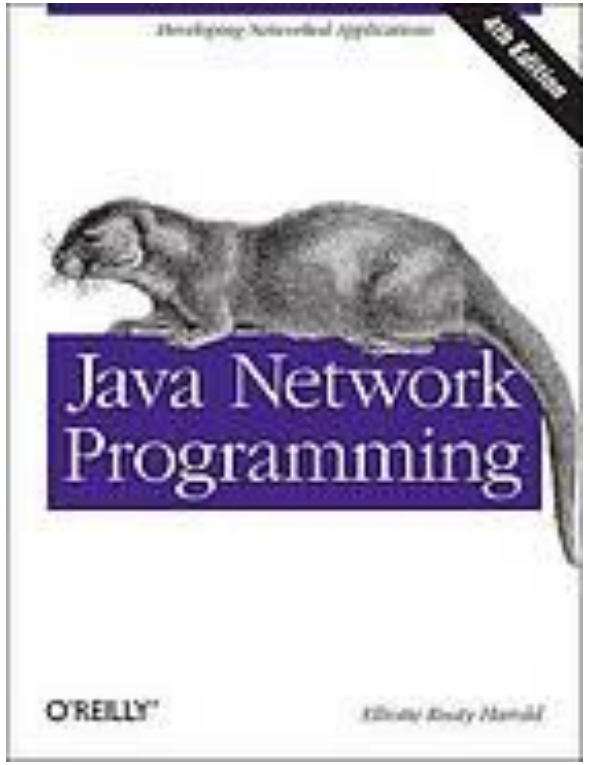
Christian Camilo Urcuqui López, MSc

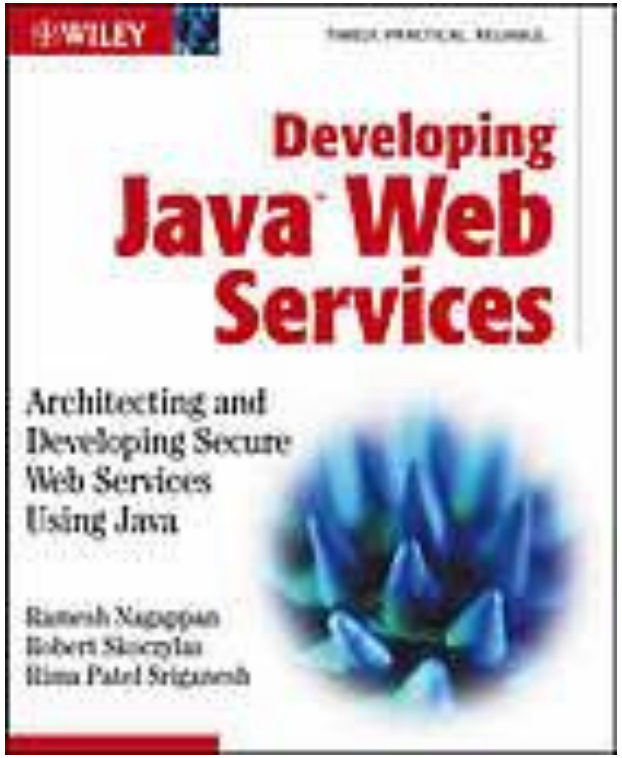
UNIVERSIDAD ICESI | A OTRO NIVEL

BIBLIOGRAFÍA









COMPETENCIAS

- Describir UDP.
- Describir TCP.

LOS PROTOCOLOS DE TRANSPORTE DE INTERNET: TCP

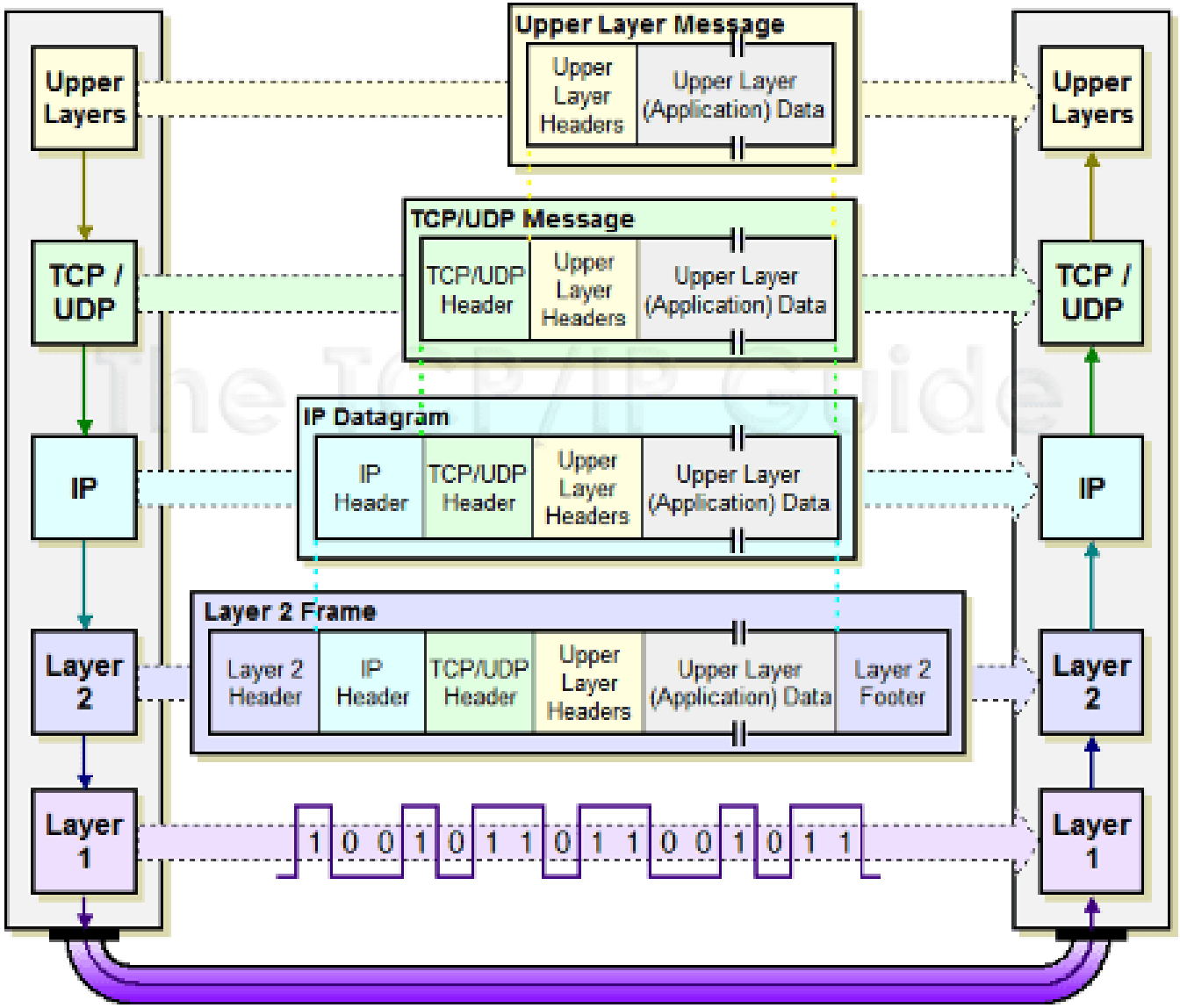
TCP (PROTOCOLO DE CONTROL DE TRANSMISIÓN

- Se diseño con la finalidad de proporcionar un flujo de bytes confiable de extremo a extremo a través de una interred no confiable. Roadmap RFC
- El servicio TCP se obtiene al hacer que tanto el servidor como el receptor creen puntos terminales, llamados **sockets.**
- Cada socket tiene un número (dirección) que consiste en la dirección IP del host y un número de 16 bits que es local para ese host, llamado **puerto**

Puerto	Protocolo	Uso
20, 21	FTP	Transferencia de archivos.
22	SSH	Inicio de sesión remoto, reemplazo de Telnet.
25	SMTP	Correo electrónico.
80	HTTP	World Wide Web.
110	POP-3	Acceso remoto al correo electrónico.
143	IMAP	Acceso remoto al correo electrónico.
443	HTTPS	Acceso seguro a web (HTTP sobre SSL/TLS).
543	RTSP	Control del reproductor de medios.
631	IPP	Compartición de impresoras.

Figura 6-34. Algunos puertos asignados.

La capa IP no ofrece garantía de que los datagramas se entregarán de manera apropiada y en el orden correcto; es trabajo de TCP incorporar los mecanismos que permitan tener un buen desempeño y confiabilidad.



TCP (PROTOCOLO DE CONTROL DE TRANSMISIÓN

- Todas las conexiones TCP son *full dúplex* y punto a punto.
- Una conexión TCP es un flujo de bytes, no un flujo de mensajes.

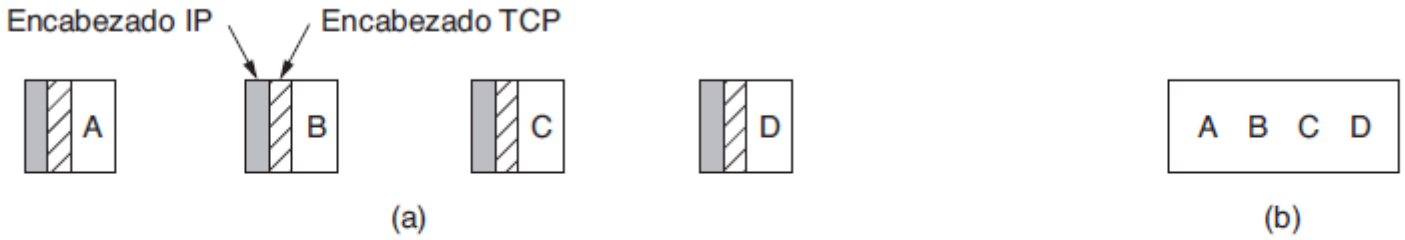


Figura 6-35. (a) Cuatro segmentos de 512 bytes que se envían como diagramas IP separados. (b) Los 2 048 bytes de datos que se entregan a la aplicación en una sola llamada READ.

- Cuando una aplicación pasa datos a TCP, éste decide entre enviarlos de inmediato o almacenarlos en el búfer. La bandera PUSH (PSH) sirve para forzar la salida de datos.

TCP (PROTOCOLO DE CONTROL DE TRANSMISIÓN

- URGENT es otra bandera que le indica a TCP que deje de acumular datos y transmita de inmediato todo lo que tiene para esa conexión (se utiliza cuando de un usuario presiona CTRL-C para interrumpir un cálculo remoto).
- Cada byte de una conexión de TCP tiene su propio número de secuencia de 32 bits.

TCP (PROTOCOLO DE CONTROL DE TRANSMISIÓN

- Un **segmento TCP** consiste en un encabezado fijo de 20 bytes, es decir, 160 bits (más una parte opcional), seguido de cero o más bytes de datos. **El software de TCP decide qué tan grandes deben ser los segmentos**
- Hay dos límites que restringen el tamaño del segmento.
  1. Cada segmento debe caber en la carga útil de 65515 bytes del IP.
  2. Cada segmento debe caber en la MTU (Unidad Máxima de Transferencia) del emisor y receptor con el fin de evitar fragmentación.

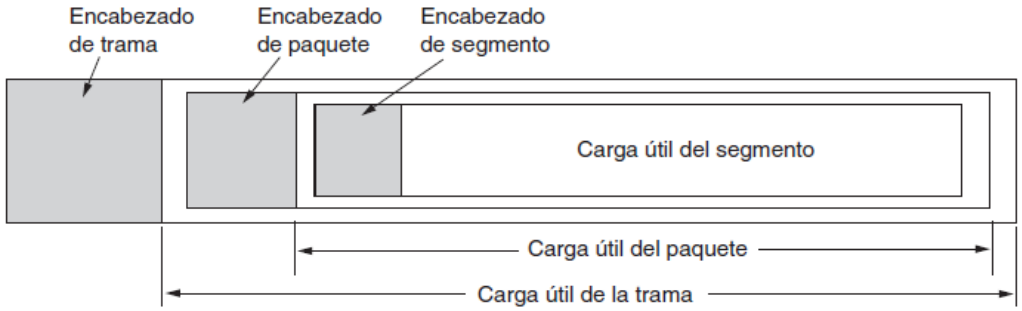


Figura 6-3. Anidamiento de segmentos, paquetes y tramas.

TCP (PROTOCOLO DE CONTROL DE TRANSMISIÓN

- Las implementaciones modernas de TCP realizan el descubrimiento de MTU de la ruta mediante mensajes de error de ICMP con el fin de encontrar la ruta más pequeña

- 5 tupla: protocolo (TCP), dirección IP destino y origen, puerto destino y origen.
- *CWR y ECE* para indicar congestión
- *SYN* se usa para establecer conexiones.
- *RST* se utiliza para reestablecer la conexión.
- *FIN* libera la conexión.

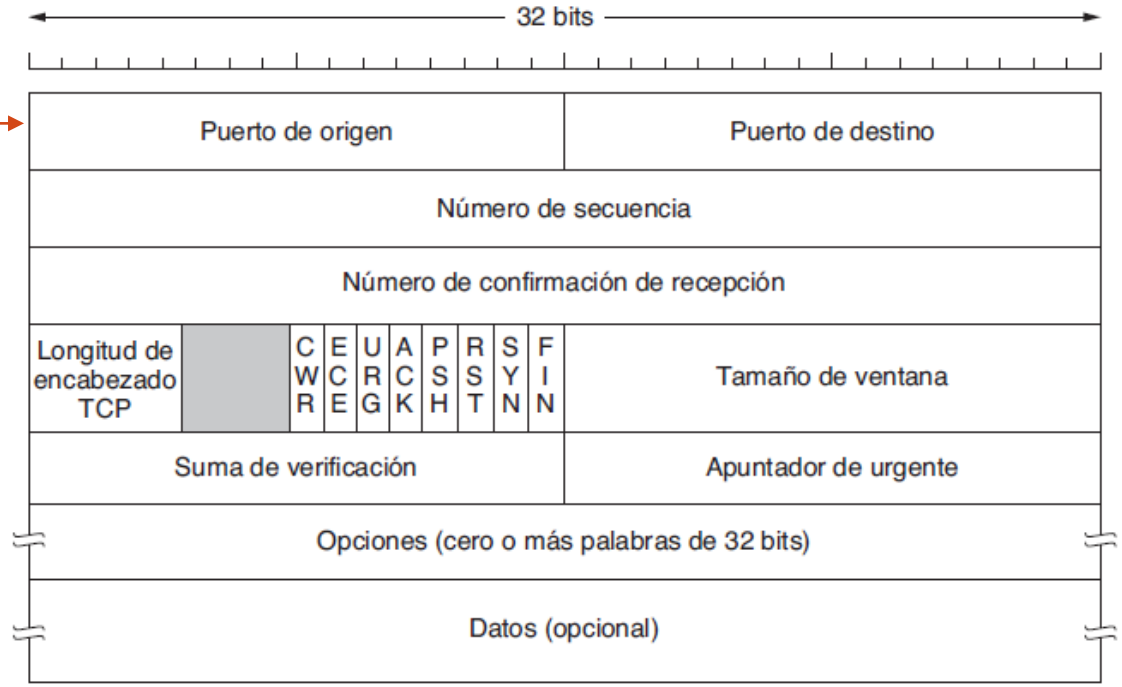


Figura 6-36. El encabezado TCP.

TCP (PROTOCOLO DE CONTROL DE TRANSMISIÓN

- El control de flujo en TCP se maneja mediante una ventana deslizante de tamaño variable.
  - El campo *Tamaño de ventana* indica la cantidad de bytes que se pueden enviar.
  - Un campo de Tamaño de ventana de 0 es válido e indica que se han recibido los bytes hasta *Número de confirmación de recepción*
  - *-1* para cuando el receptor no ha tenido oportunidad de consumirlos y ya no desea más.

- *Suma de verificación* para agregar confiabilidad.
- *Opciones* agrega las características adicionales que no están cubiertas por el encabezado normal.
  - MSS (Tamaño Máximo de Segmento) que un host permite aceptar.
  - Tamaño de escala permite al emisor y al receptor negociar un factor de escala de ventana al inicio de la conexión.

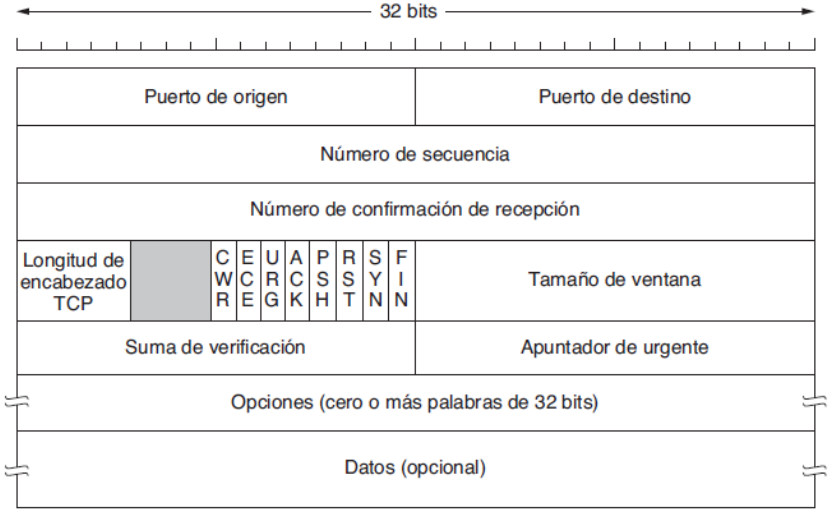
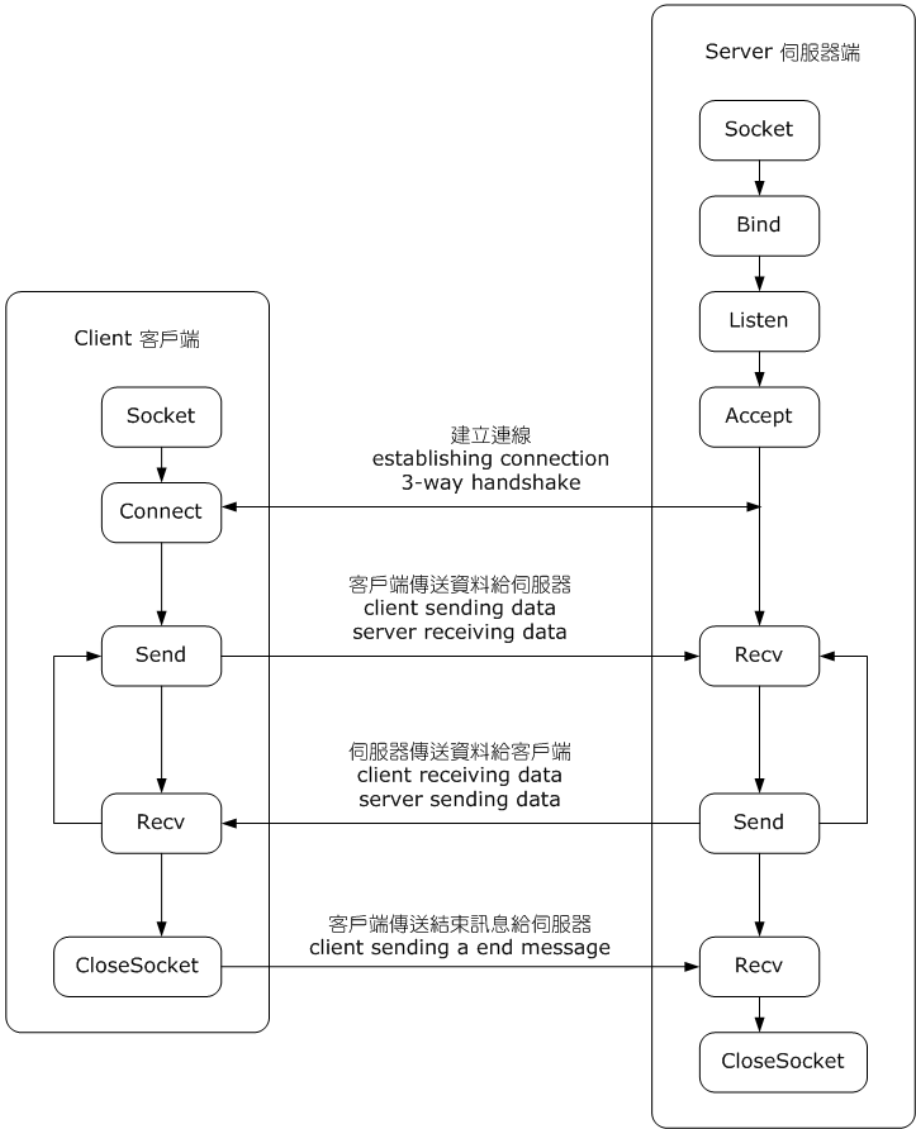


Figura 6-36. El encabezado TCP.

TCP Socket 基本流程圖
TCP Socket flow diagram



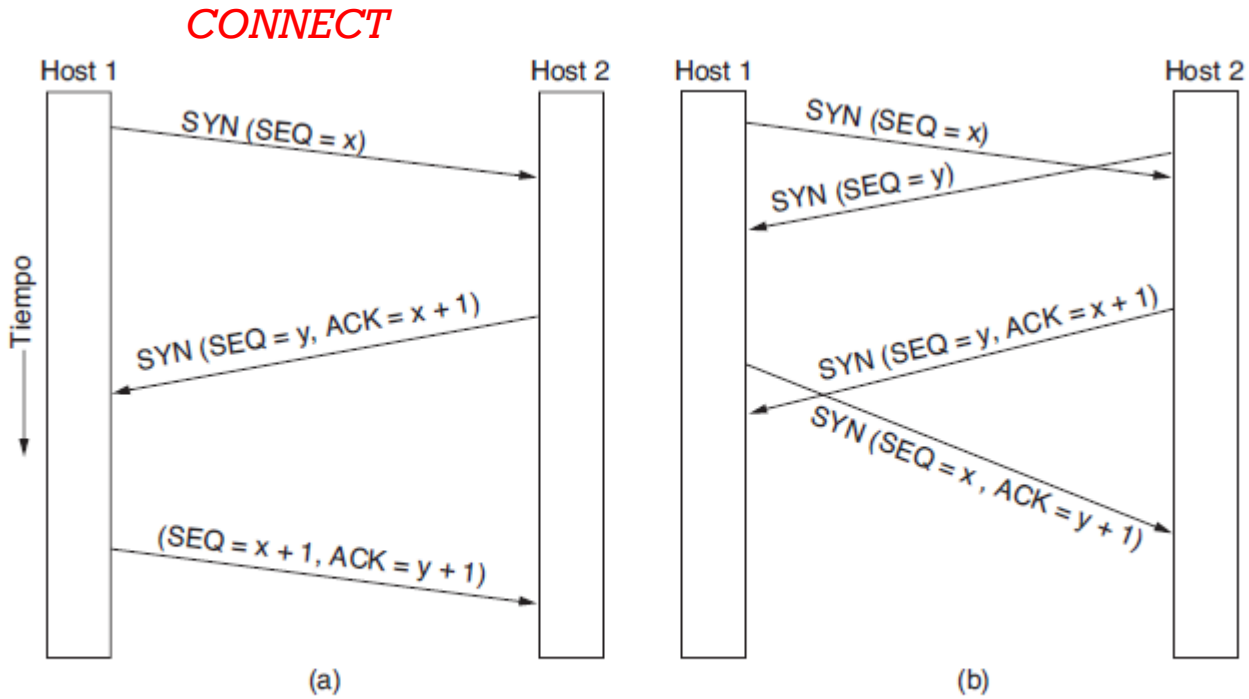


Figura 6-37. (a) Establecimiento de una conexión TCP en el caso normal. (b) Establecimiento de una conexión simultánea en ambos lados.

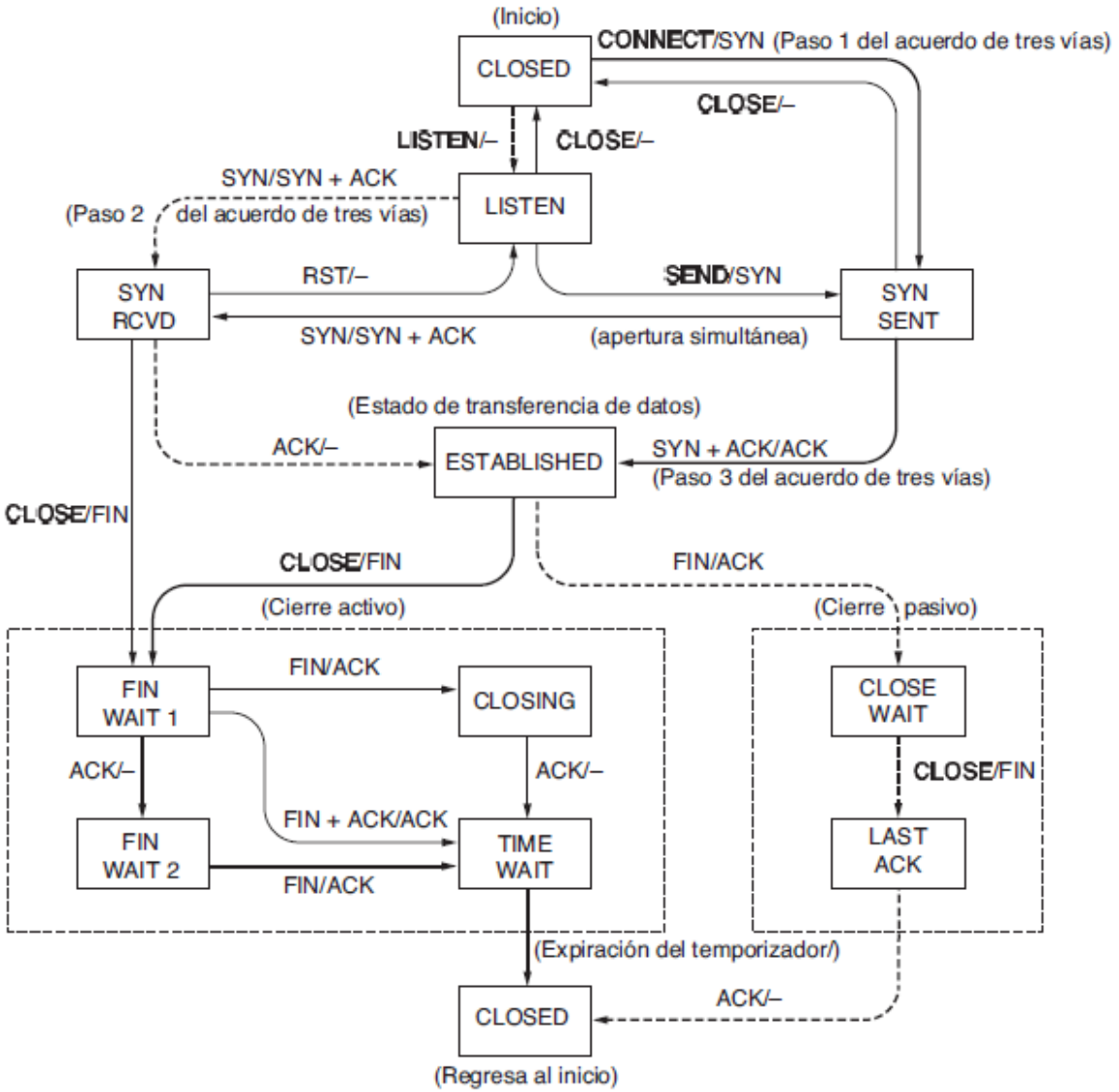
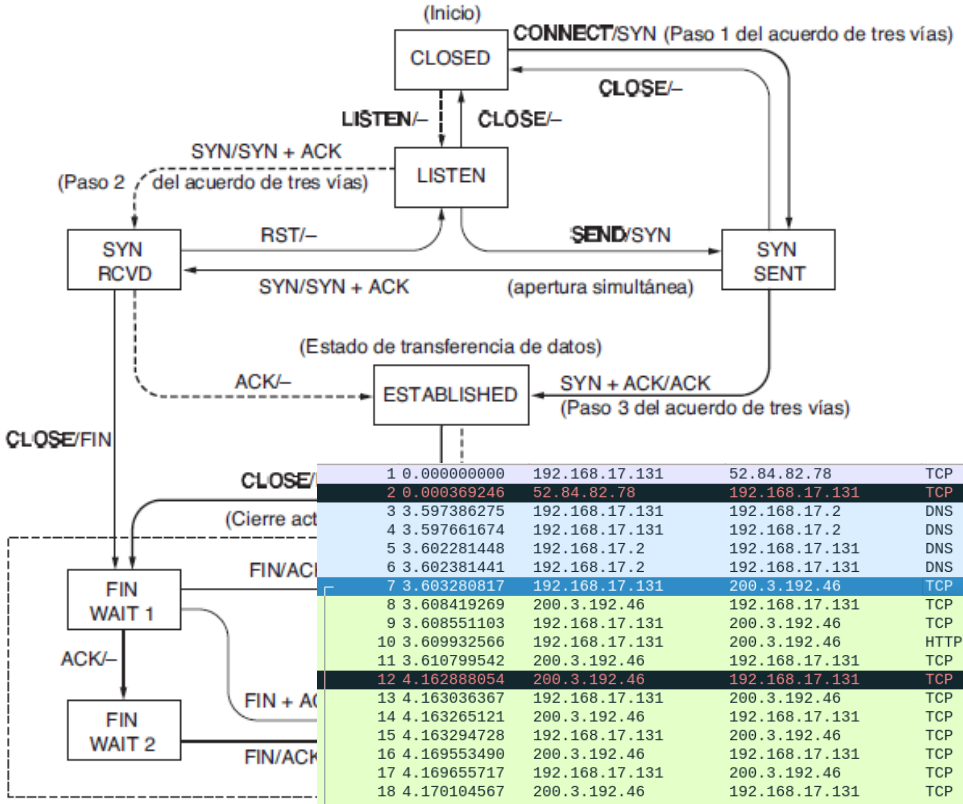


Figura 6-39. Máquina de estados finitos para administrar las conexiones TCP. La línea continua gruesa es la trayectoria normal para un cliente. La línea punteada gruesa es la trayectoria normal para un servidor. Las líneas delgadas son eventos no usuales. Cada transición se etiqueta con el evento que la ocasiona y la acción resultante, separada por una diagonal.



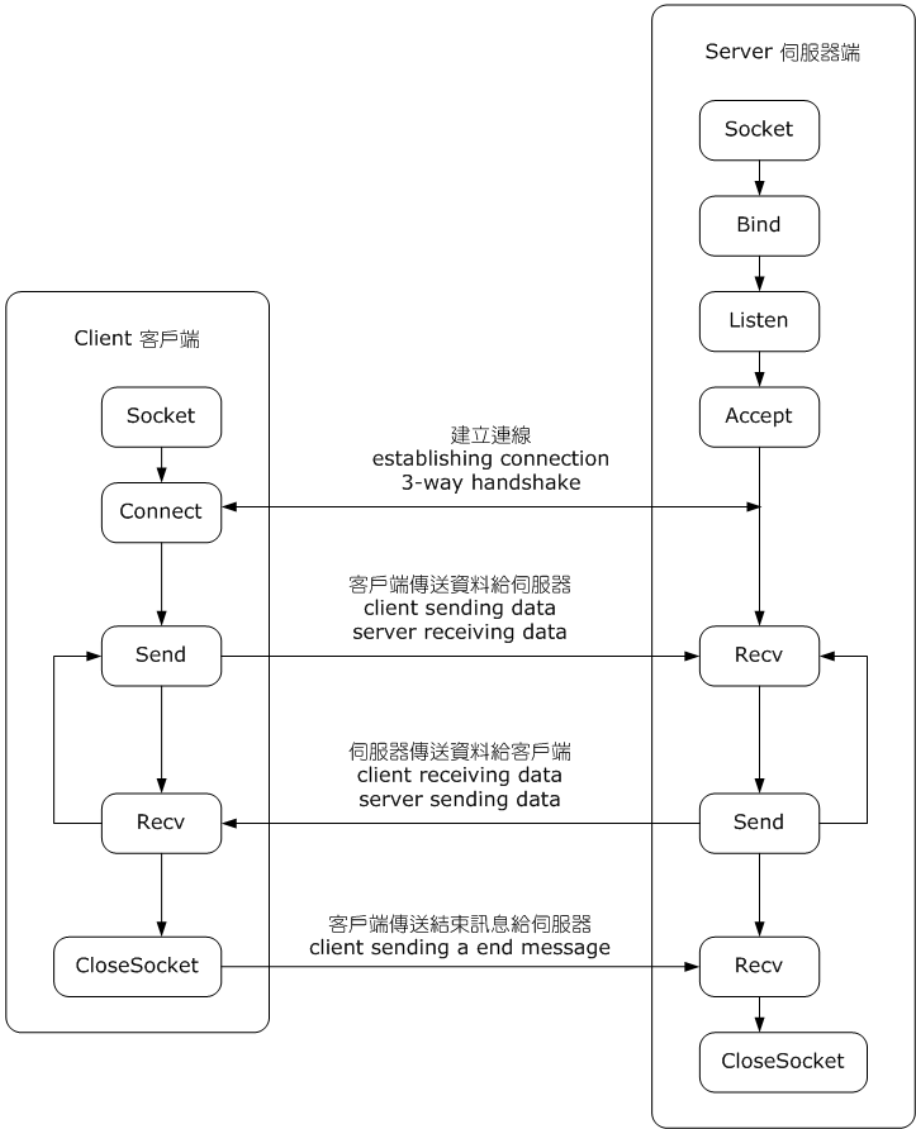
1	0.000000000	192.168.17.131	52.84.82.78	TCP	54	41020 → 443 [ACK] Seq=1 Ack=1 Win=65535 Len=0
2	0.000369246	52.84.82.78	192.168.17.131	TCP	60	[TCP ACKed unseen segment] 443 → 41020 [ACK] Seq=1 Ack=2 Win=64240 Len=0
3	3.597386275	192.168.17.131	192.168.17.2	DNS	76	Standard query 0xaeb6 A www.icesi.edu.co
4	3.597661674	192.168.17.131	192.168.17.2	DNS	76	Standard query 0xd535 AAAA www.icesi.edu.co
5	3.602281448	192.168.17.2	192.168.17.131	DNS	126	Standard query response 0xd535 AAAA www.icesi.edu.co SOA kodos
6	3.602381441	192.168.17.2	192.168.17.131	DNS	111	Standard query response 0xaeb6 A www.icesi.edu.co A 200.3.192.46 NS kodos
7	3.603280817	192.168.17.131	200.3.192.46	TCP	74	57204 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=582435103 TSecr=0 WS=128
8	3.608419269	200.3.192.46	192.168.17.131	TCP	60	80 → 57204 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
9	3.608551103	192.168.17.131	200.3.192.46	TCP	54	57204 → 80 [ACK] Seq=1 Ack=1 Win=29200 Len=0
10	3.609932566	192.168.17.131	200.3.192.46	HTTP	370	GET / HTTP/1.1
11	3.610799542	200.3.192.46	192.168.17.131	TCP	60	80 → 57204 [ACK] Seq=1 Ack=317 Win=64240 Len=0
12	4.162888054	200.3.192.46	192.168.17.131	TCP	29254	[TCP Window Full] 80 → 57204 [PSH, ACK] Seq=1 Ack=317 Win=64240 Len=29200 [TCP segment of a reassembled PDU]
13	4.163036367	192.168.17.131	200.3.192.46	TCP	54	57204 → 80 [ACK] Seq=317 Ack=29201 Win=64240 Len=0
14	4.163265121	200.3.192.46	192.168.17.131	TCP	3559	80 → 57204 [PSH, ACK] Seq=29201 Ack=317 Win=64240 Len=3505 [TCP segment of a reassembled PDU]
15	4.163294728	192.168.17.131	200.3.192.46	TCP	54	57204 → 80 [ACK] Seq=317 Ack=32706 Win=64240 Len=0
16	4.169553490	200.3.192.46	192.168.17.131	TCP	64293	80 → 57204 [PSH, ACK] Seq=32706 Ack=317 Win=64240 Len=64239 [TCP segment of a reassembled PDU]
17	4.169655717	192.168.17.131	200.3.192.46	TCP	54	57204 → 80 [ACK] Seq=317 Ack=96945 Win=64240 Len=0
18	4.170104567	200.3.192.46	192.168.17.131	TCP	64293	80 → 57204 [PSH, ACK] Seq=96945 Ack=317 Win=64240 Len=64239 [TCP segment of a reassembled PDU]
19	4.170143670	192.168.17.131	200.3.192.46	TCP	54	57204 → 80 [ACK] Seq=317 Ack=161184 Win=65535 Len=0
20	4.170466272	200.3.192.46	192.168.17.131	HTTP	13600	HTTP/1.1 200 OK (text/html)
21	4.170499231	192.168.17.131	200.3.192.46	TCP	54	57204 → 80 [ACK] Seq=317 Ack=174731 Win=65535 Len=0
22	4.171320679	192.168.17.131	200.3.192.46	TCP	54	57204 → 80 [FIN, ACK] Seq=317 Ack=174731 Win=65535 Len=0
23	4.171805001	200.3.192.46	192.168.17.131	TCP	60	80 → 57204 [ACK] Seq=174731 Ack=318 Win=64239 Len=0
24	4.288429458	192.168.17.131	200.3.192.46	TCP	74	57206 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=582435788 TSecr=0 WS=128

```
▸ Frame 7: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
▸ Ethernet II, Src: Vmware_b3:d3:87 (00:0c:29:b3:d3:87), Dst: Vmware_e3:c6:70 (00:50:56:e3:c6:70)
▸ Internet Protocol Version 4, Src: 192.168.17.131, Dst: 200.3.192.46
▾ Transmission Control Protocol, Src Port: 57204, Dst Port: 80, Seq: 0, Len: 0
    Source Port: 57204
    Destination Port: 80
    [Stream index: 1]
    [TCP Segment Len: 0]
    Sequence number: 0    (relative sequence number)
    Acknowledgment number: 0
    1010 .... = Header Length: 40 bytes (10)
  ▾ Flags: 0x002 (SYN)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Nonce: Not set
      .... 0... .... = Congestion Window Reduced (CWR): Not set
      .... .0.. .... = ECN-Echo: Not set
      .... ..0. .... = Urgent: Not set
      .... ...0 .... = Acknowledgment: Not set
      .... .... 0... = Push: Not set
      .... .... .0.. = Reset: Not set
    ▸ .... .... ..1. = Syn: Set
      .... .... ...0 = Fin: Not set
      [TCP Flags: ··········S·]
    Window size value: 29200
```

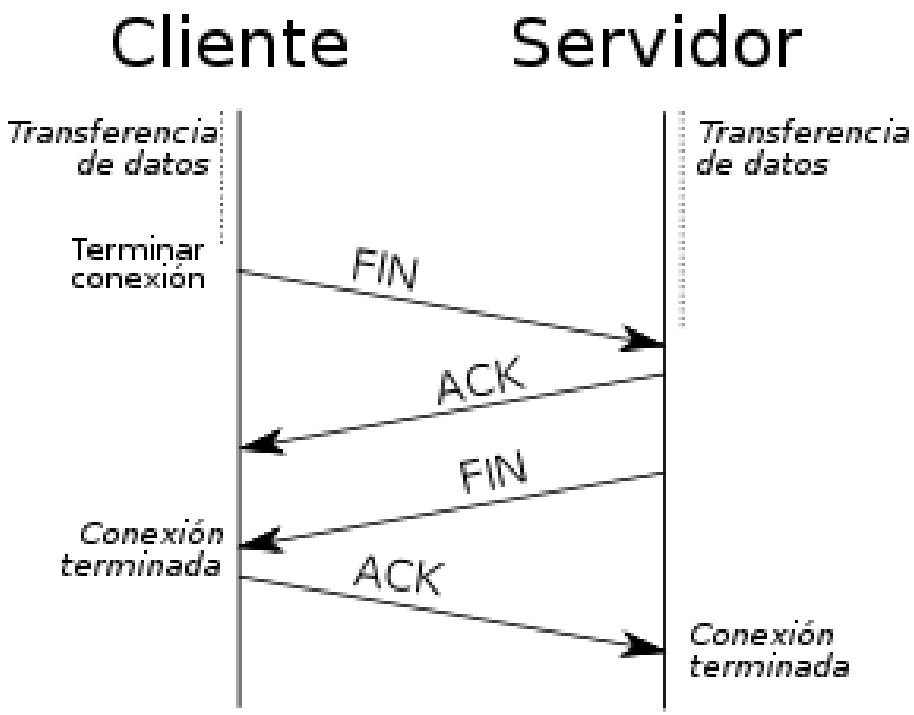
Figura 6-39. Máquina de estados finitos p
normal para un cliente. La línea punteada g
no usuales. Cada transición se etiqueta con

UNIVERSIDAD ICESI | A OTRO NIVEL

TCP Socket 基本流程圖
TCP Socket flow diagram



DISCONECT



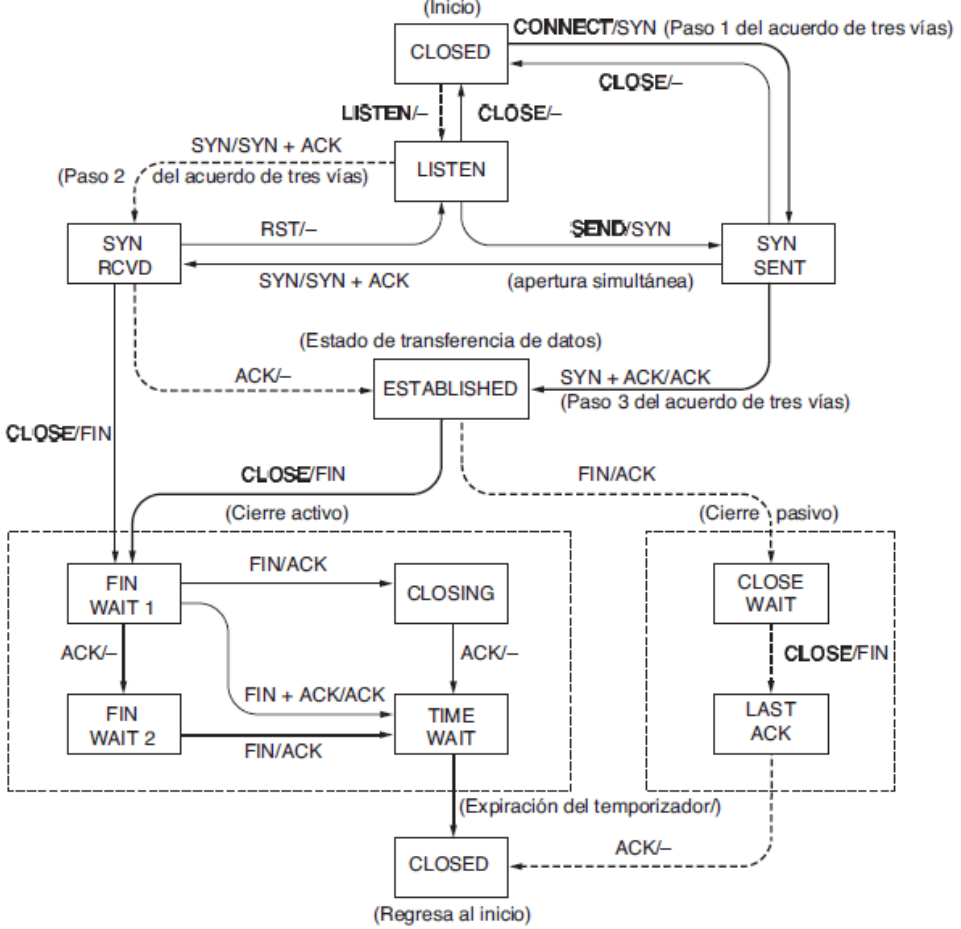


Figura 6-39. Máquina de estados finitos para administrar las conexiones TCP. La línea continua gruesa es la trayectoria normal para un cliente. La línea punteada gruesa es la trayectoria normal para un servidor. Las líneas delgadas son eventos no usuales. Cada transición se etiqueta con el evento que la ocasiona y la acción resultante, separada por una diagonal.

```
Win=64240 Len=0
=517 Win=64240 Len=38664 [TCP segment of a reassembled PDU]
Win=64240 Len=0
=517 Win=64240 Len=21612 [TCP segment of a reassembled PDU]
 Win=64240 Len=0
k=517 Win=64240 Len=14696 [TCP segment of a reassembled PDU]
 Win=64240 Len=0
k=517 Win=64240 Len=31317 [TCP segment of a reassembled PDU]
 Win=65535 Len=0
k=517 Win=64240 Len=6831 [TCP segment of a reassembled PDU]
 Win=65535 Len=0
k=517 Win=64240 Len=64239 [TCP segment of a reassembled PDU]
 Win=65535 Len=0
k=517 Win=64240 Len=1674 [TCP segment of a reassembled PDU]
 Win=65535 Len=0
k=517 Win=64240 Len=39580 [TCP segment of a reassembled PDU]
 Win=65535 Len=0
k=517 Win=64240 Len=6879 [TCP segment of a reassembled PDU]
 Win=65535 Len=0
k=517 Win=64240 Len=8718 [TCP segment of a reassembled PDU]
 Win=65535 Len=0
k=517 Win=64240 Len=7724 [TCP segment of a reassembled PDU]
 Win=65535 Len=0
k=517 Win=64240 Len=27727 [TCP segment of a reassembled PDU]
 Win=65535 Len=0
k=517 Win=64240 Len=7971 [TCP segment of a reassembled PDU]
 Win=65535 Len=0
k=517 Win=64240 Len=6282 [TCP segment of a reassembled PDU]
 Win=65535 Len=0
k=517 Win=64240 Len=10219 [TCP segment of a reassembled PDU]
 Win=65535 Len=0
k=517 Win=64240 Len=32688 [TCP segment of a reassembled PDU]
 Win=65535 Len=0
k=517 Win=64240 Len=64239 [TCP segment of a reassembled PDU]
```

```
508 10.405029558  192.168.17.131  200.3.192.46      TCP      54 57482 → 80 [ACK] Seq=517 Ack=431614 Win=65535 Len=0
509 10.405360324  200.3.192.46    192.168.17.131    TCP   25890 80 → 57482 [PSH, ACK] Seq=431614 Ack=517 Win=64240 Len=25836 [TCP segment of a reassembled PDU]
510 10.405399714  192.168.17.131  200.3.192.46      TCP      54 57482 → 80 [ACK] Seq=517 Ack=457450 Win=65535 Len=0
511 10.407244350  200.3.192.46    192.168.17.131    TCP   59533 80 → 57482 [PSH, ACK] Seq=457450 Ack=517 Win=64240 Len=59479 [TCP segment of a reassembled PDU]
512 10.407294636  192.168.17.131  200.3.192.46      TCP      54 57482 → 80 [ACK] Seq=517 Ack=516929 Win=65535 Len=0
513 10.409103234  200.3.192.46    192.168.17.131    TCP   64293 80 → 57482 [PSH, ACK] Seq=516929 Ack=517 Win=64240 Len=64239 [TCP segment of a reassembled PDU]
514 10.409182573  192.168.17.131  200.3.192.46      TCP      54 57482 → 80 [ACK] Seq=517 Ack=581168 Win=65535 Len=0
515 10.409493784  200.3.192.46    192.168.17.131    TCP    8946 80 → 57482 [PSH, ACK] Seq=581168 Ack=517 Win=64240 Len=8892 [TCP segment of a reassembled PDU]
516 10.410489994  192.168.17.131  200.3.192.46      TCP      54 57482 → 80 [ACK] Seq=517 Ack=590060 Win=65535 Len=0
517 10.417655384  200.3.192.46    192.168.17.131    HTTP  27939 HTTP/1.1 200 OK  (text/css)
518 10.417773834  192.168.17.131  200.3.192.46      TCP      54 57482 → 80 [ACK] Seq=517 Ack=617946 Win=65535 Len=0
519 10.418843237  192.168.17.131  200.3.192.46      TCP      54 57482 → 80 [FIN, ACK] Seq=517 Ack=617946 Win=65535 Len=0
520 10.419155006  200.3.192.46    192.168.17.131    TCP      60 80 → 57482 [ACK] Seq=617946 Ack=518 Win=64239 Len=0
```

TCP SYN (STEALTH) SCAN (-SS)

- https://nmap.org/book/synscan.html

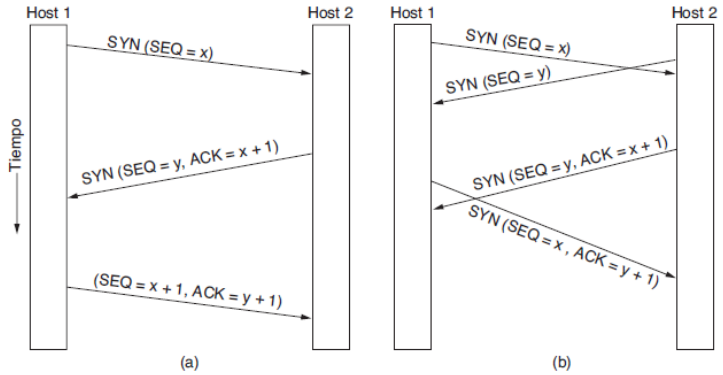
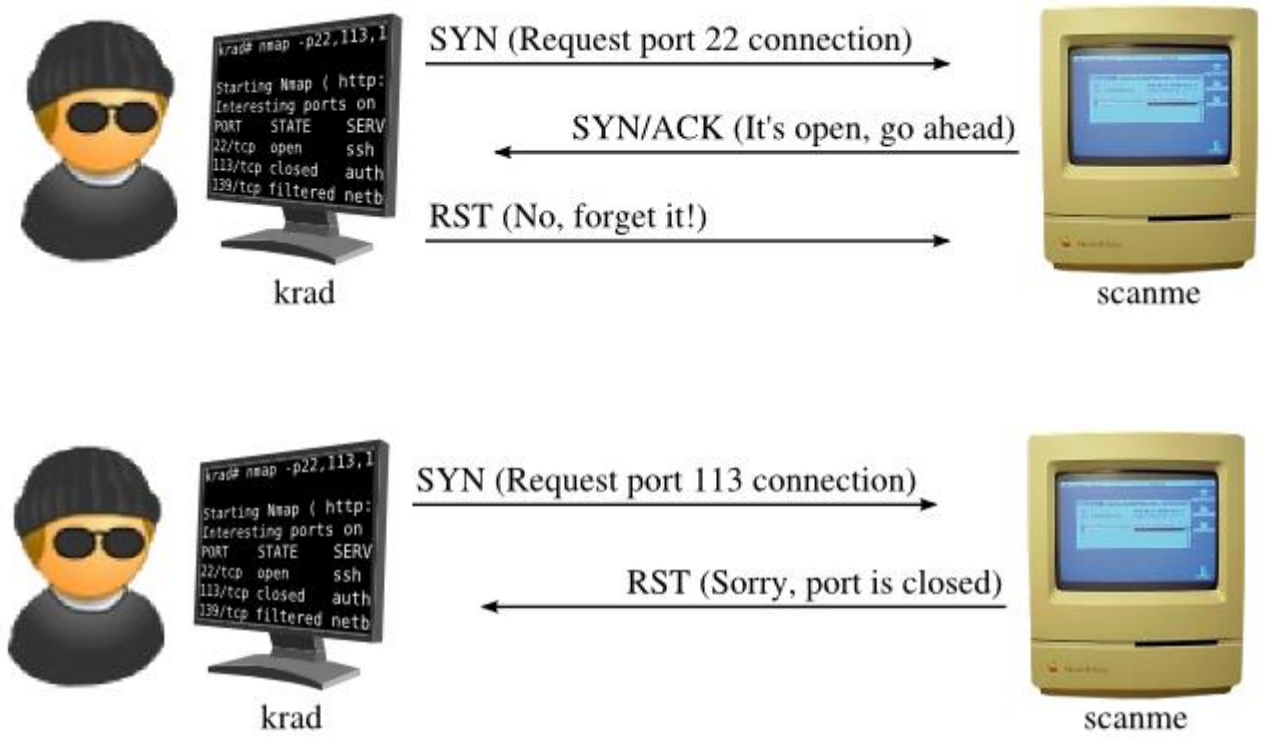


Figura 6-37. (a) Establecimiento de una conexión TCP en el caso normal. (b) Establecimiento de una conexión simultánea en ambos lados.



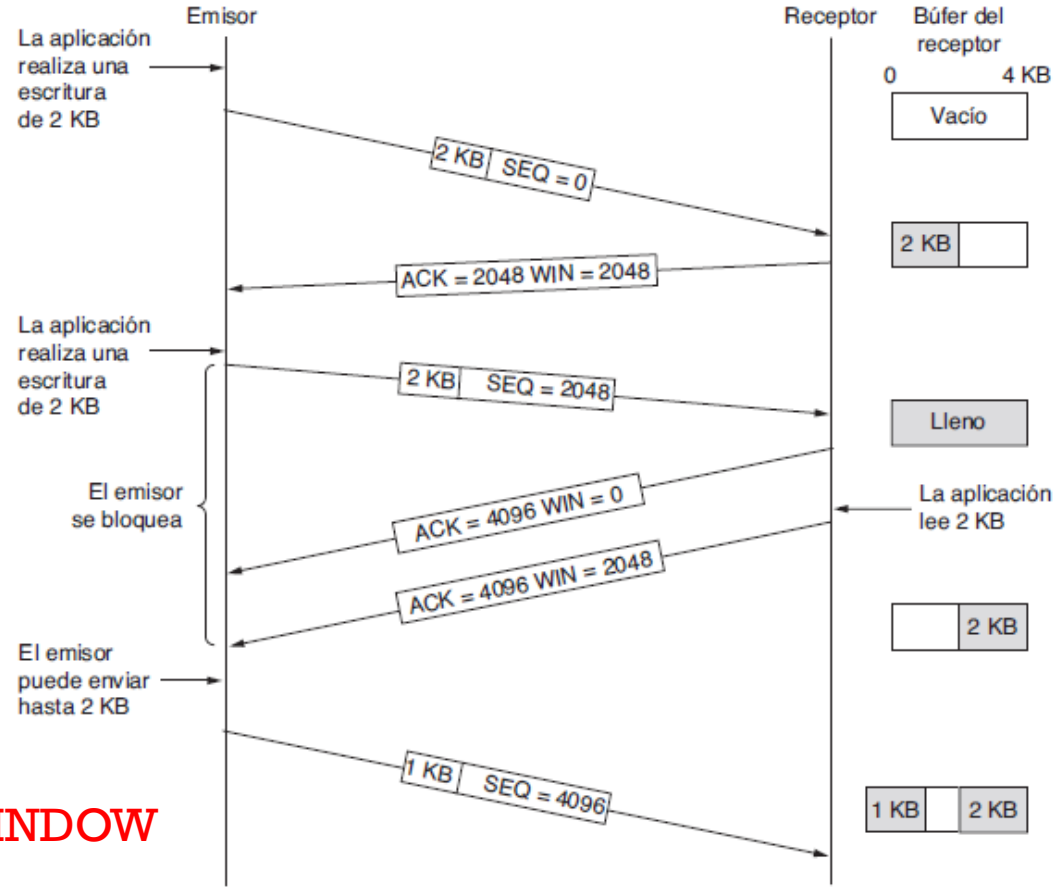


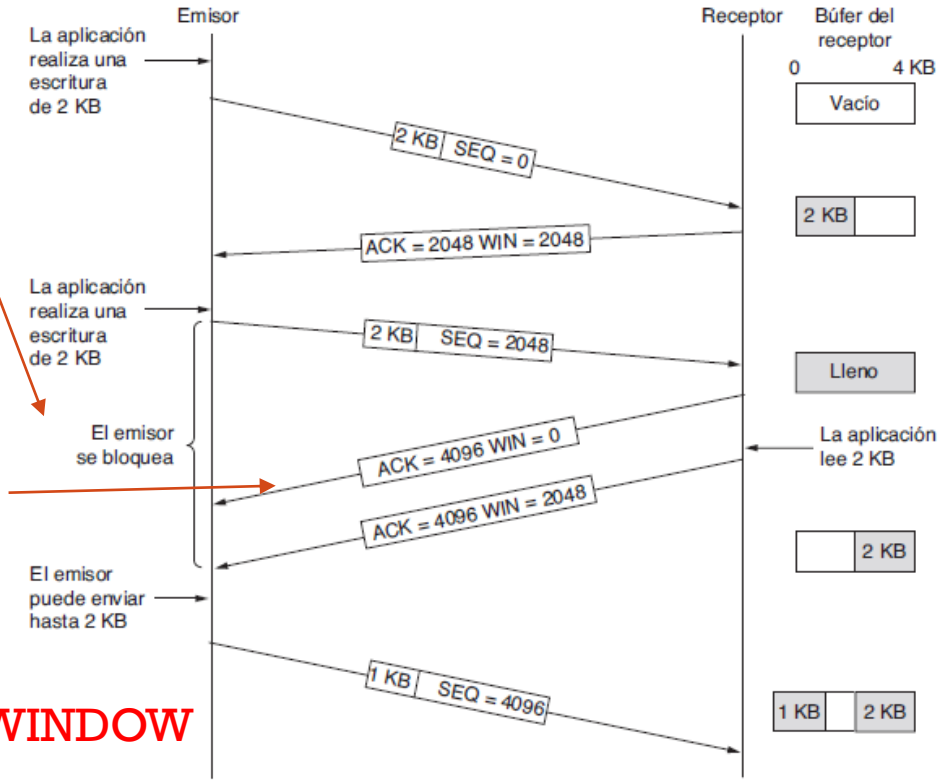
Figura 6-40. Administración de ventanas en TCP.

WIN= WINDOW

VENTANA DESLIZANTE DE TCP

- La ventana de 1 TCP separa los aspectos de la confirmación de la recepción correcta de los segmentos y la asignación del búfer en el receptor.

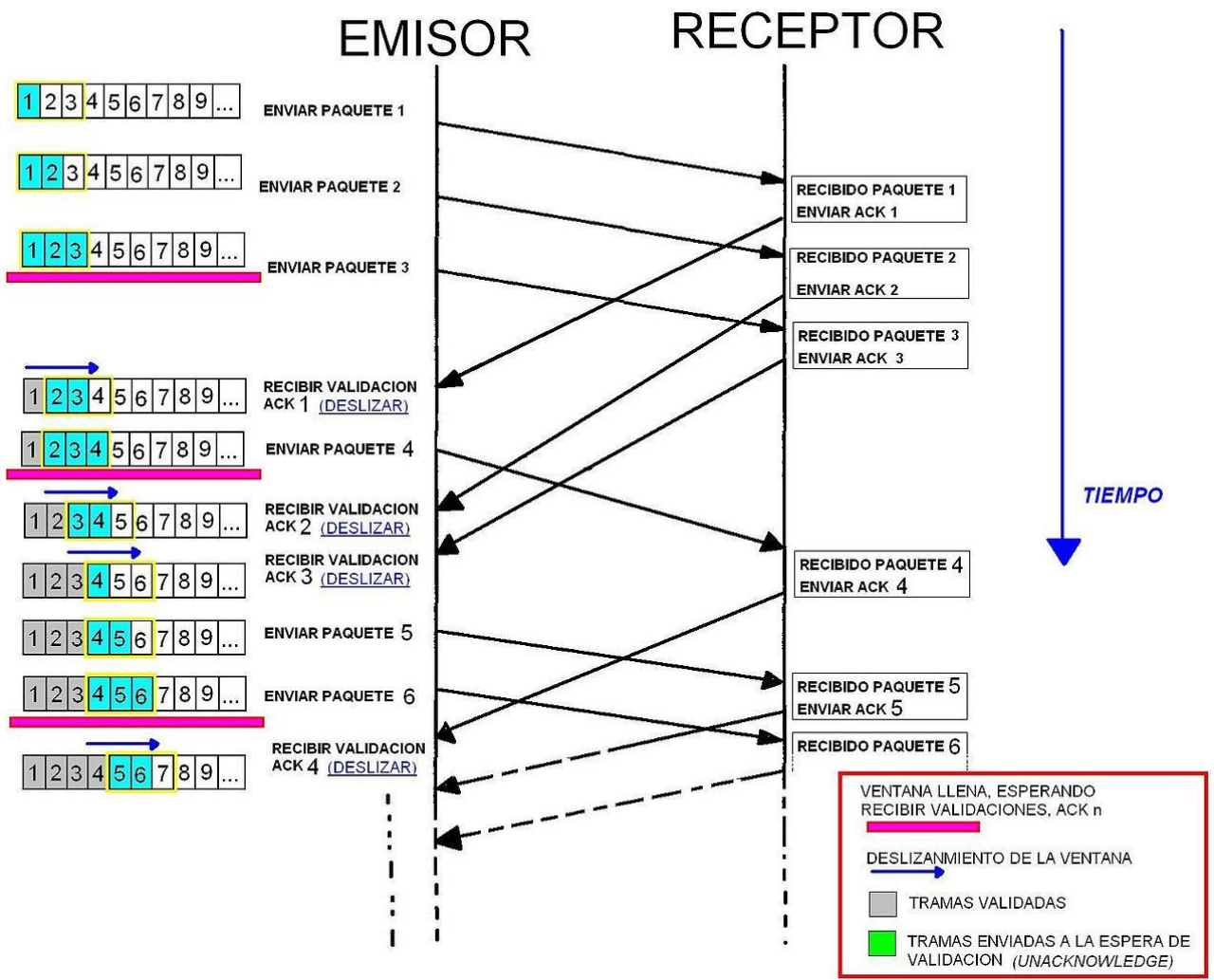
- Cuando la ventana es 0, el emisor no puede enviar segmentos, excepto en dos situaciones:
  - Datos urgentes (URG) para finalizar el proceso en ejecución.
  - El emisor envía un segmento de 1 byte para que el receptor vuelva a anunciar el siguiente byte esperado y el tamaño de la ventana **(sonda de ventana).**



WIN= WINDOW

Figura 6-40. Administración de ventanas en TCP.

VENTANA DESLIZANTE DE TCP



VENTANA DESLIZANTE DE TCP

- Cuando el ancho de banda escasea y la aplicación que utiliza TCP aplica una suma de segmentos elevada (por ejemplo, Telnet y SSH), existe un mecanismo **(confirmaciones de recepción con retardo)**

que permite retrasar las confirmaciones de recepción y las actualizaciones de ventana hasta 500 mseg, con el objetivo de reducir la carga impuesta en la red por el receptor. Pero este mecanismo es ineficiente para varios paquetes cortos (por ejemplo, paquetes de 41 bytes que contienen 1 byte de datos).

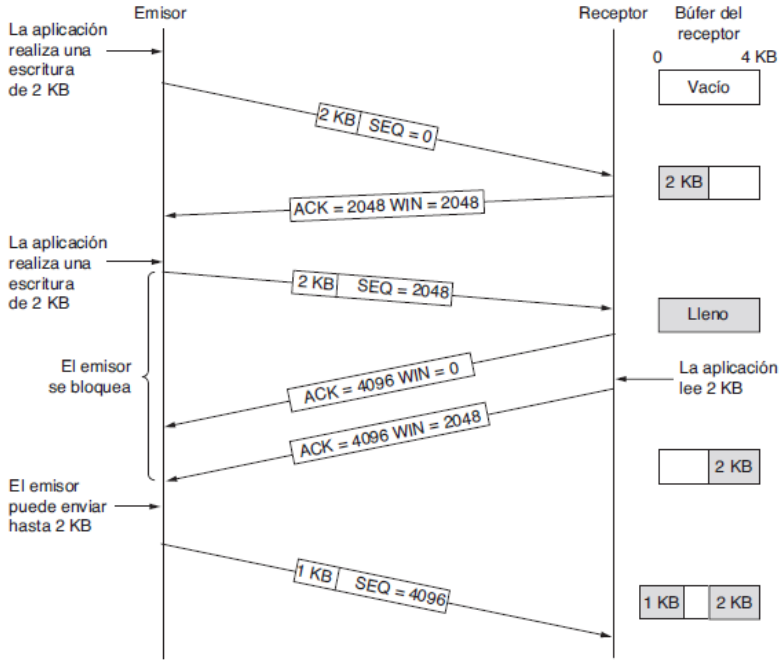


Figura 6-40. Administración de ventanas en TCP.

EL ALGORITMO DE NAGLE

- Cuando llegan datos en pequeñas piezas al emisor, sólo se envía la primera pieza y el resto se almacena en búfer hasta que se confirma la recepción del byte pendiente. Después se envían todos los datos del búfer en un segmento TCP y nuevamente comienzan a almacenarse en búfer los datos hasta que se haya confirmado la recepción del siguiente segmento. **Esto significa que sólo puede haber un paquete corto pendiente en cualquier momento dado.**
- **Es bueno deshabilitarlo cuando hay aplicaciones de gran flujo rápido de paquetes, por ejemplo, los juegos interactivos. Desactivar la opción *TCP_NODELAY***

https://youtu.be/fS3PW5yOttk

SÍNDROME DE VENTANA TONTA

- Este síndrome ocurre cuando se pasan datos a la entidad TCP **emisora en bloques grandes**, pero una aplicación interactiva del lado **receptor lee datos sólo a razón de 1 byte a la vez**.
- **La solución de Clark** es **evitar que el receptor envíe una actualización de ventana para 1 byte**. En cambio, se le obliga a esperar hasta tener disponible una cantidad decente de espacio, y luego lo anuncia. El receptor no envía una actualización de ventana sino hasta que pueda manejar el tamaño máximo de segmento que anuncio al establecerse la conexión.

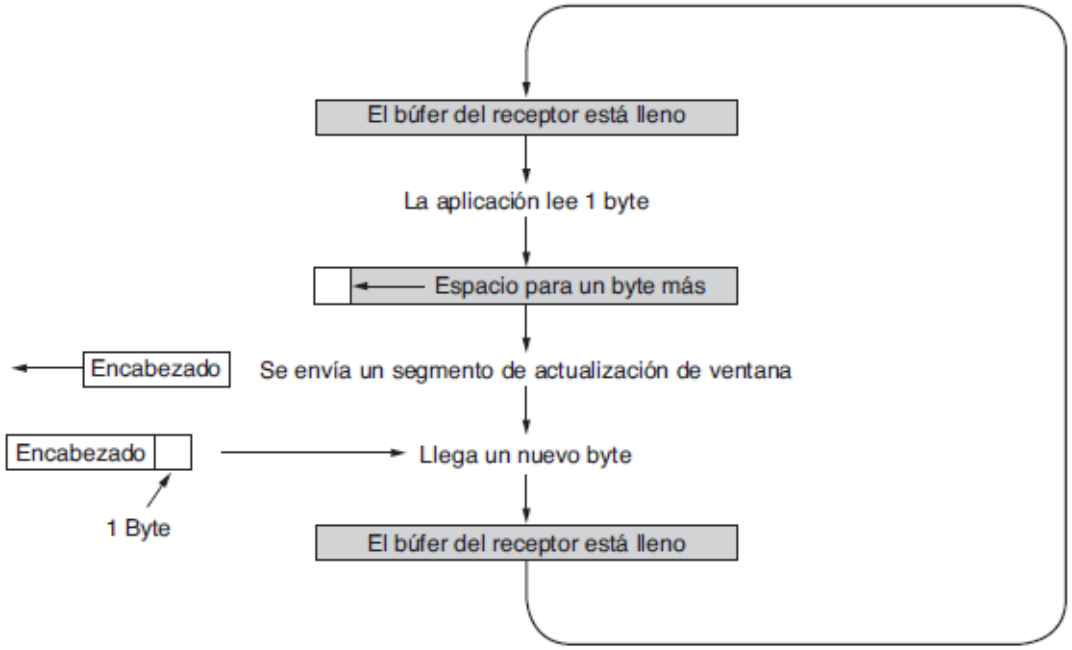


Figura 6-41. Síndrome de ventana tonta.

CLARK Y NAGLE

El algoritmo de Nagle y la solución de Clark al síndrome de ventana tonta son complementarios. **Nagle trataba de resolver el problema causado por la aplicación emisora que entregaba datos a TCP, 1 byte a la vez. Clark trataba de resolver el problema de que la aplicación receptora tomara los datos de TCP, 1 byte a la vez.** Ambas soluciones son válidas y pueden operar juntas. **El objetivo es que el emisor no envíe segmentos pequeños y que el receptor no los pida.**

EL FUTURO DE TCP

- Existen distintas versiones de los algoritmos clásicos, en especial para el control de la congestión y la robustez ante los ataques informáticos. Dos aspectos a tener en cuenta:
  - TCP no proporciona la semántica de transporte que desean todas las aplicaciones, por ejemplo el control de los limites en los mensajes o registros.
    - SCTP (Protocolo de Control de Transmisión de Flujo, *Stream Control Transmision Protocol*). RFC 4960
    - SST (Transporte Estructurado de Flujo, *Structured Transmision Protocol).*
  - Control de la congestión. TCP se basa en las pérdidas de paquetes como señal de congestión.

LOS PROTOCOLOS DE TRANSPORTE DE INTERNET: UDP

UDP (USER DATAGRAM PROTOCOL)

- Proporciona una forma para que las aplicaciones envíen datagramas IP encapsulados sin tener que establecer una conexión. RFC 768.
- UDP es un protocolo simple para interacciones cliente/servidor y multimedia.
- UDP trasmite segmentos que consisten en un encabezado de 8 bytes seguido de la carga útil.

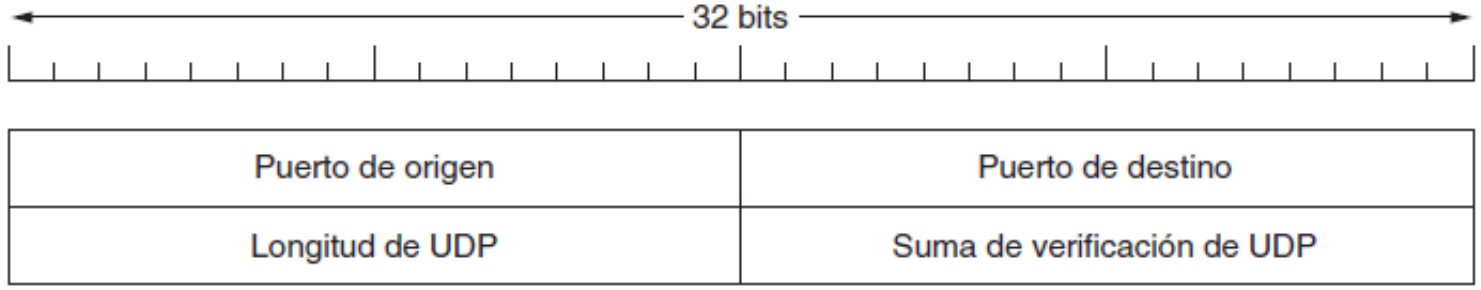
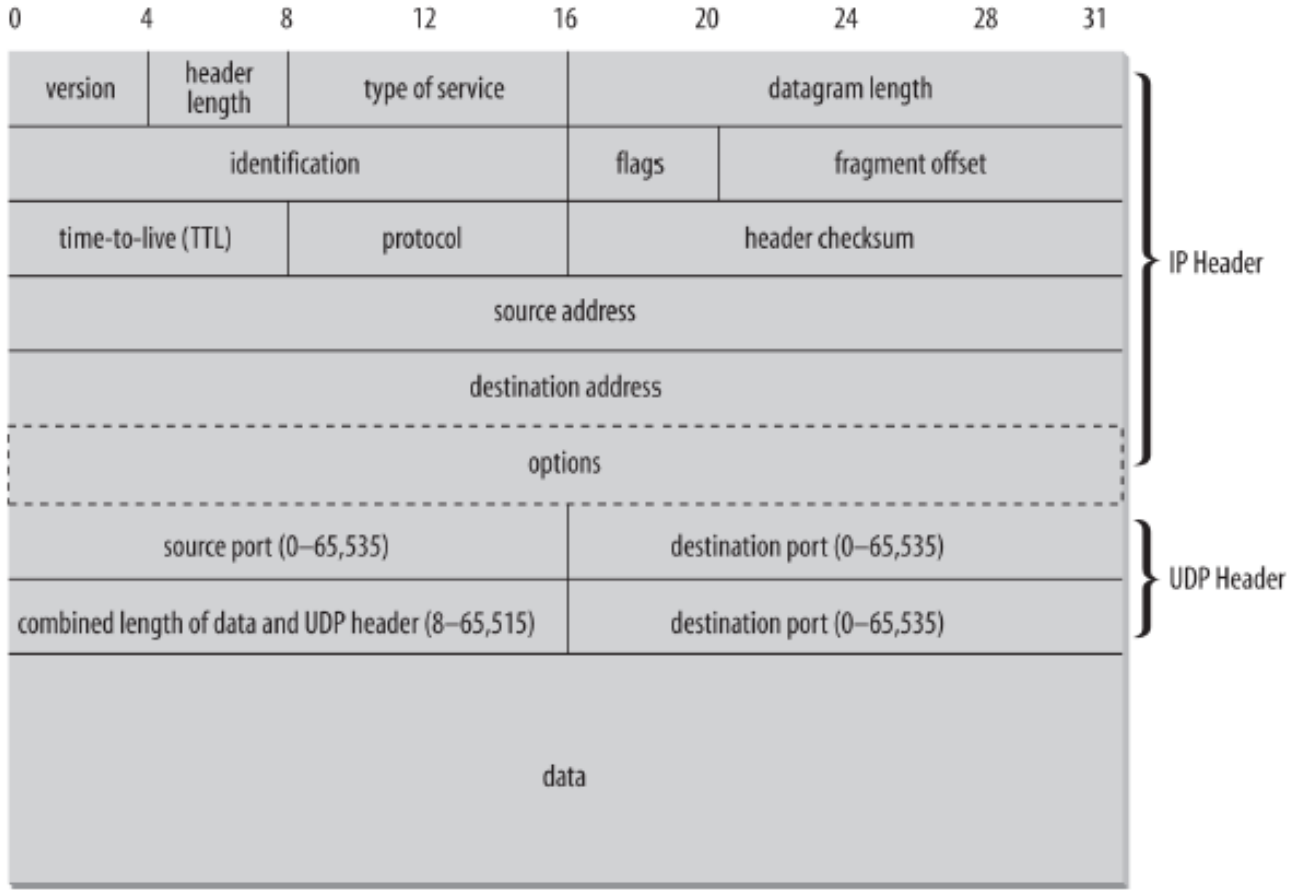


Figura 6-27. El encabezado UDP.

- La longitud incluye el encabezado de 8 bytes y los datos. La longitud mínima es de 8 bytes y la máxima es de 655515 bytes.

UDP (USER DATAGRAM PROTOCOL)



LLAMADA A PROCEDIMIENTO REMOTO

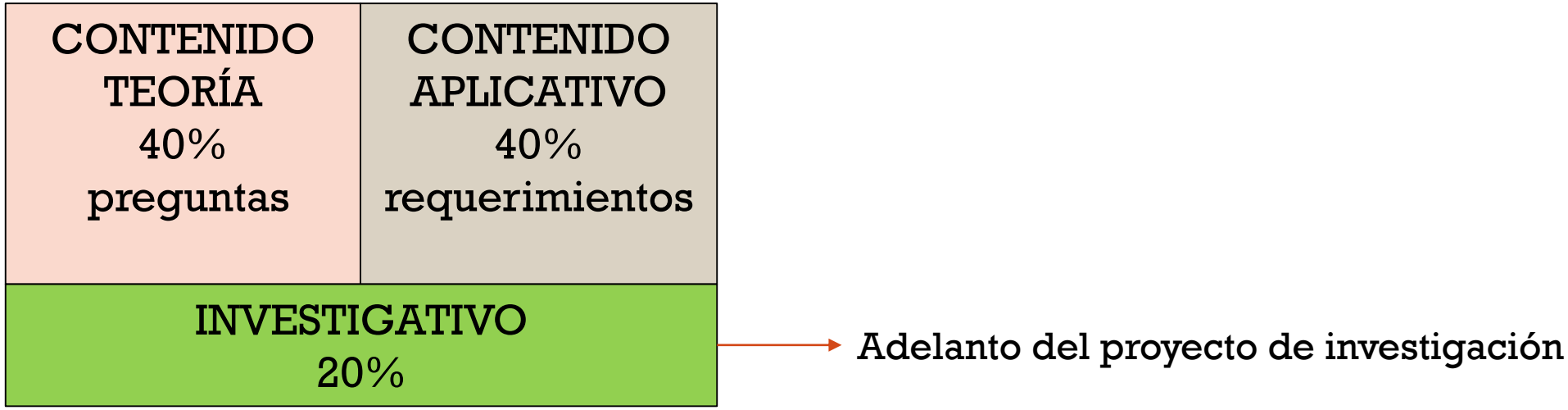
- Cuando un proceso en la máquina 1 llama a otro procedimiento de la máquina 2, el proceso invocador en 1 se suspende y la ejecución del procedimiento invocado se lleva a cabo en la máquina 2.
- Se puede transportar información del proceso invocador al proceso invocado en los parámetros, y se puede regresar información en el resultado del procedimiento. El paso de mensajes es transparente para el programador de la aplicación. Está técnica se conoce como **RPC (Llamada a Procedimiento Remoto).**
- **Es un protocolo de gran avance sobre los sockets**

SESIONES RESTANTES

Próxima semana

- Sustentación de practico I – Entrega el domingo
  - Sesión 20
- Avances del proyecto de investigación ()
  - Sesión 21, no hay clase – resolución de dudas
  - Sesión 22, revisión de avances
- ¿Parcial II – Sesión 23? ¿O después de semana santa?

PARCIAL NÚMERO 2



Material utilizado	1. Arboleda, L. (2012). Programación en Red con Java. 2. Harold, E. (2004). Java network programming. " O'Reilly Media, Inc.". 3. Tanenbaum, A. S. (2003). Redes de computadoras. Pearson educación. 4. Reese, R. M. (2015). Learning Network Programming with Java. Packt Publishing Ltd.
Actividades DESPUÉS clase	A1. Secciones 7.1 y 7.2

REFERENCIAS

1. https://www.google.com.co/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwju5o62x-vdAhXjtlkKHaouDcAQjRx6BAgBEAU&url=http%3A%2F%2Feltallerdelbit.com%2Fdireccionamiento-ip%2F&psig=AOvVaw3E_T5IpV-ANtL0eEQbHtkg&ust=1538700259199893
2. https://www.cisco.com/c/dam/en/us/support/docs/ip/dynamic-address-allocation-resolution/19580-dhcp-multintwk-4.gif
3. https://en.wikipedia.org/wiki/Berkeley_sockets#/media/File:InternetSocketBasicDiagram_zhtw.png
4. https://buildingautomationmonthly.com/what-is-the-tcp-ip-stack/
5. https://upload.wikimedia.org/wikipedia/commons/thumb/9/9d/Ventana_deslizante_2.JPG/1280px-Ventana_deslizante_2.JPG