Telefonica

WIRESHARK

Protocolo TCP

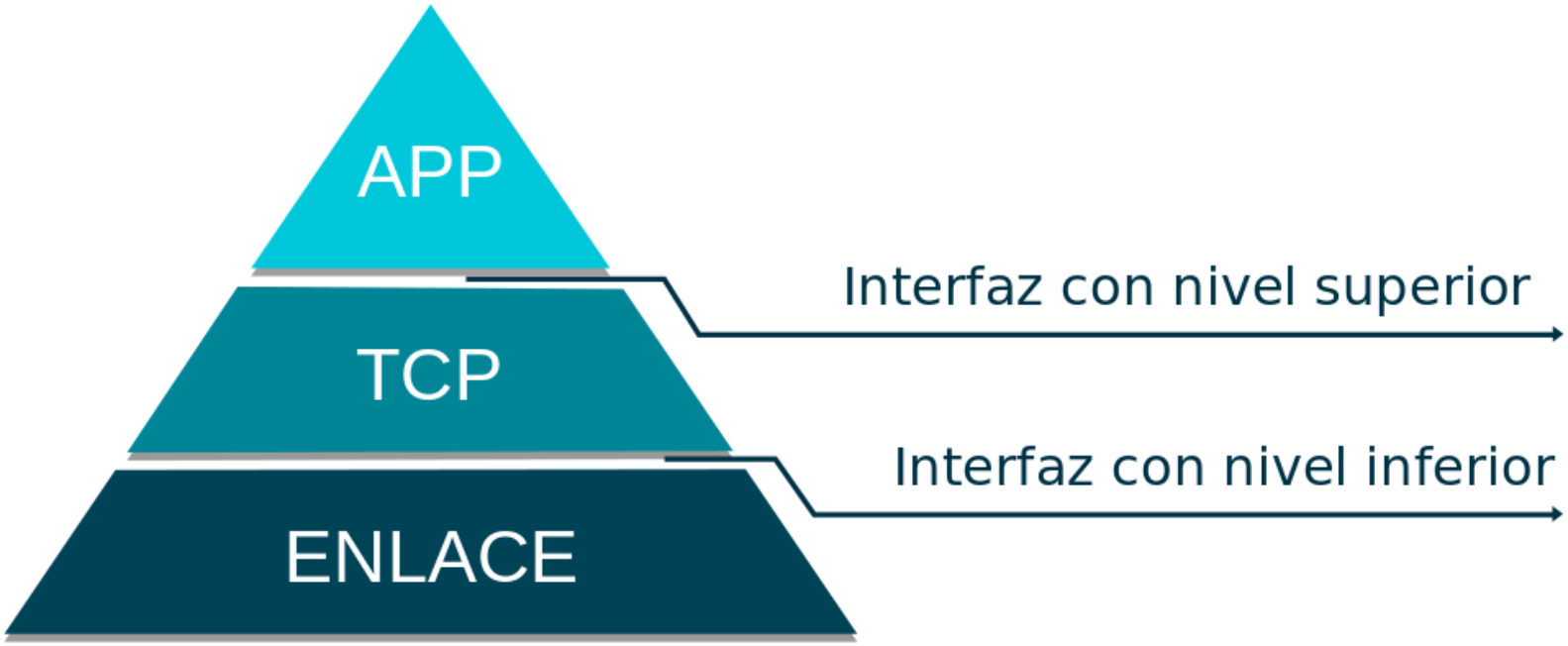
Protocolo TCP

Introducción

Protocolo TCP

Introducción

TCP es un protocolo orientado a la conexión entre dos extremos, fiable y diseñado para encajar en una jerarquía de protocolos



Protocolo TCP

Introducción

Es tarea de TCP, asegurar que los datos, se entregan bien, en secuencia y sin errores. Para poder operar sobre un entorno de Internet menos fiable, ofrece:

- Transferencia básica de datos
- Fiabilidad
- Control de flujo
- Multiplexamiento
- Conexiones
- Prioridad y seguridad

Protocolo TCP

Introducción

Un flujo de datos enviado sobre una conexión TCP, se entrega de forma fiable y ordenada al destino, para ello, necesita

- Números de secuencia
- Acuses de recibo
- Retransmisiones
- Control de flujo

Protocolo TCP

Introducción

Una conexión, empieza llamando a la función OPEN con los argumentos de un puerto local y una dirección de conector remoto

Los procedimientos para establecer conexiones, utilizan el bit de control SYN e involucran un intercambio de tres mensajes

La conexión queda establecida, cuando los números de secuencia quedan sincronizados

Protocolo TCP

Introducción

TCP, almacena los datos del usuario emisor y los envía en segmentos según su propia conveniencia, siempre que no se invoque la función **PUSH**

- PUSH, envía al destino, todos los datos que tiene hasta ese momento

Cuando el receptor, ve el indicador PUSH, no debe esperar más datos antes de pasarlos a la aplicación receptora

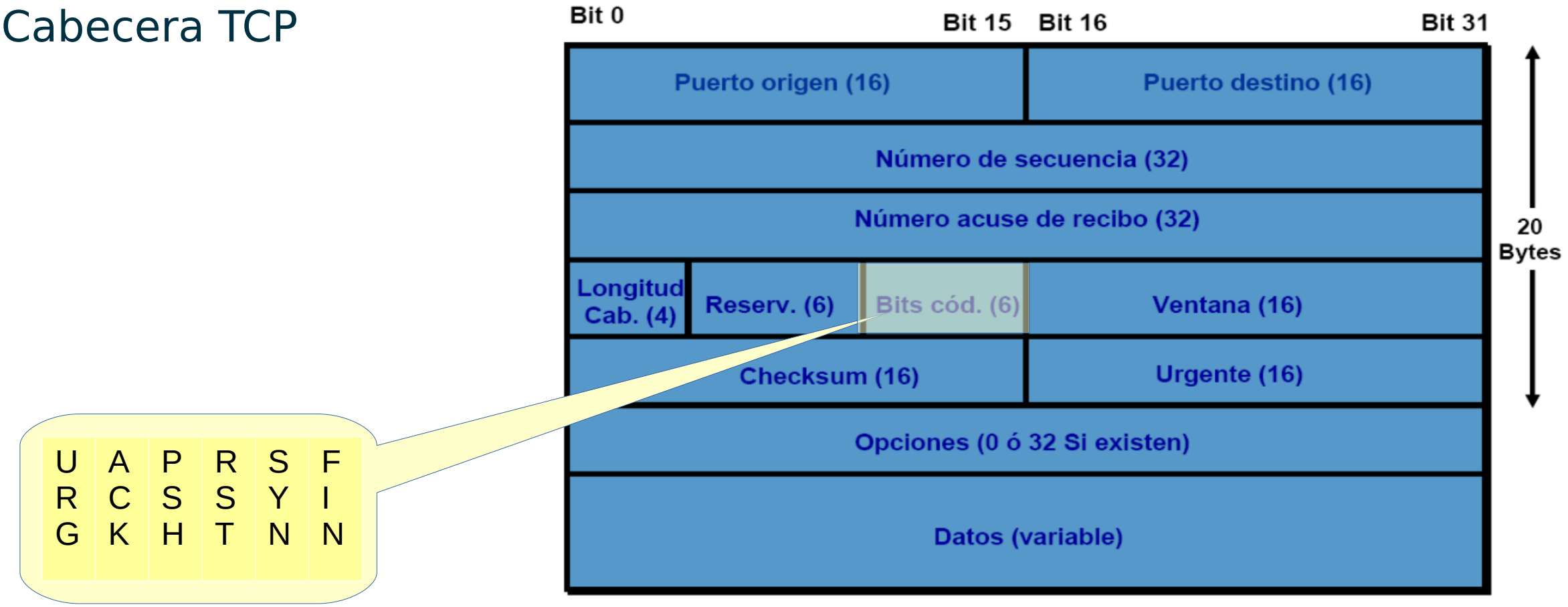
Protocolo TCP

Especificación funcional

Protocolo TCP

Especificación funcional

Cabecera TCP



Protocolo TCP

Especificación funcional

El mantenimiento de una conexión TCP, requiere el almacenamiento y seguimiento de varias variables, almacenadas en un registro de conexión

- Dirección de conector local y remoto
- Valores de seguridad y prioridad de la conexión
- Puntero al bufer de envío y recepción del usuario
- Putero a la cola de retransmisión
- Variables rlacionadas con los números de secuencia de envío y recepción

Protocolo TCP

Especificación funcional

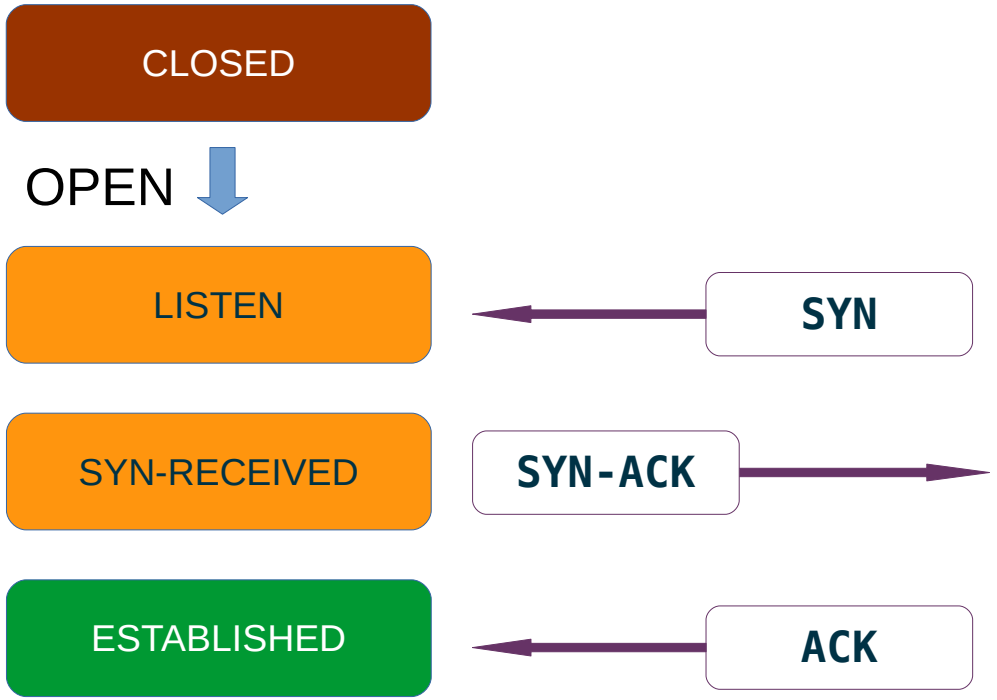
Una conexión TCP, progresa de acuerdo a una serie de estados y en función de respuesta a eventos

- OPEN, CLOSE
- SEND, RECEIVE
- ABORT
- STATUS
- Llegadas de segmentos con indicadores (SYN, ACK,, RST, FIN)
- La expiración de plazos de tiempo

Protocolo TCP

Especificación funcional

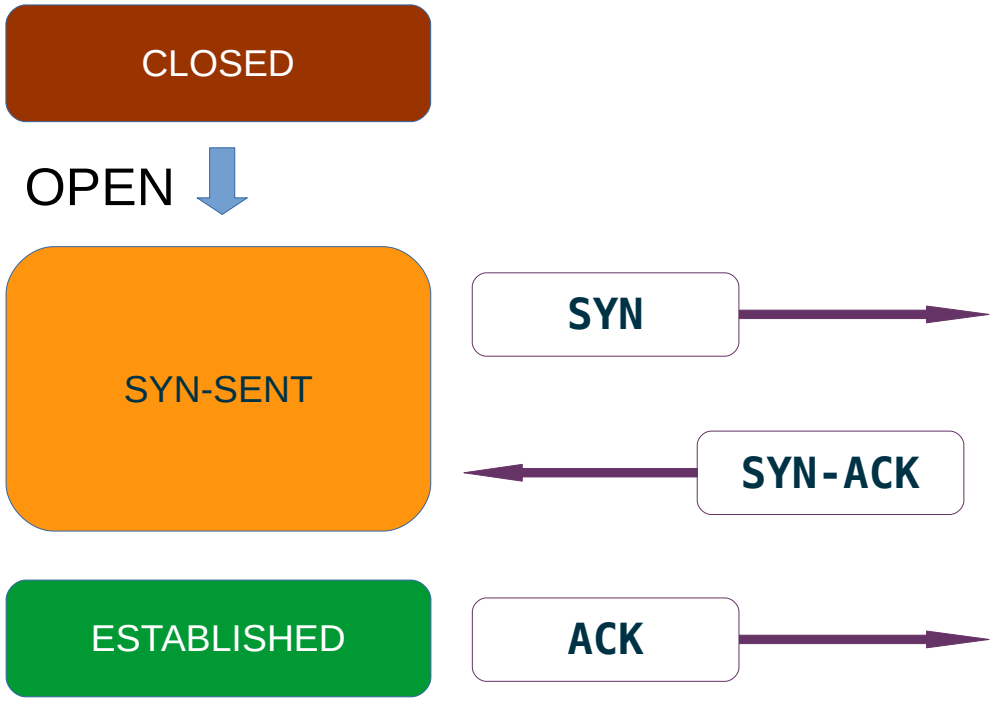
Etados en el establecimiento de una conexión **PASIVA**



Protocolo TCP

Especificación funcional

Etados en el establecimiento de una conexión **ACTIVA**



Protocolo TCP

Especificación funcional

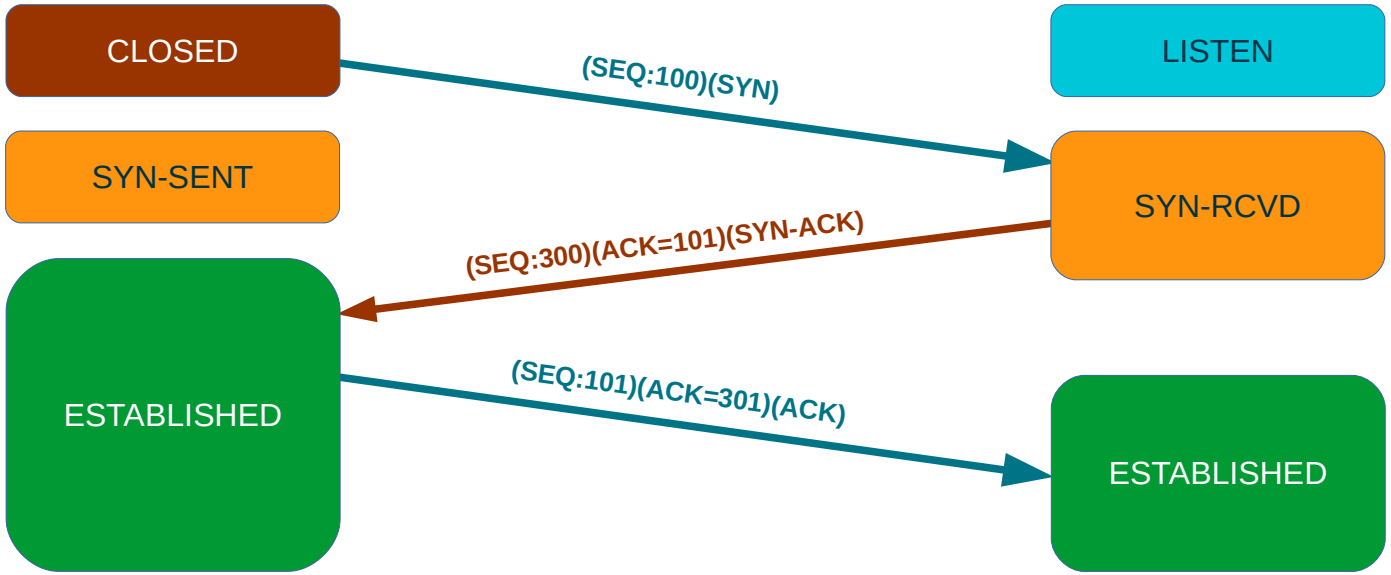
Recordamos que:

- Todo octeto de datos enviado por una conexión TCP, tiene un número de secuencia y debe ser validado por un acuse de recibo
- El acuse de recibo, es acumulativo, y una vez un segmento ha sido validado, es eliminado del bufer de retransmisión
- Para que una conexión quede establecida, los dos TCP deben sincronizar sus números de secuencia

Protocolo TCP

Especificación funcional

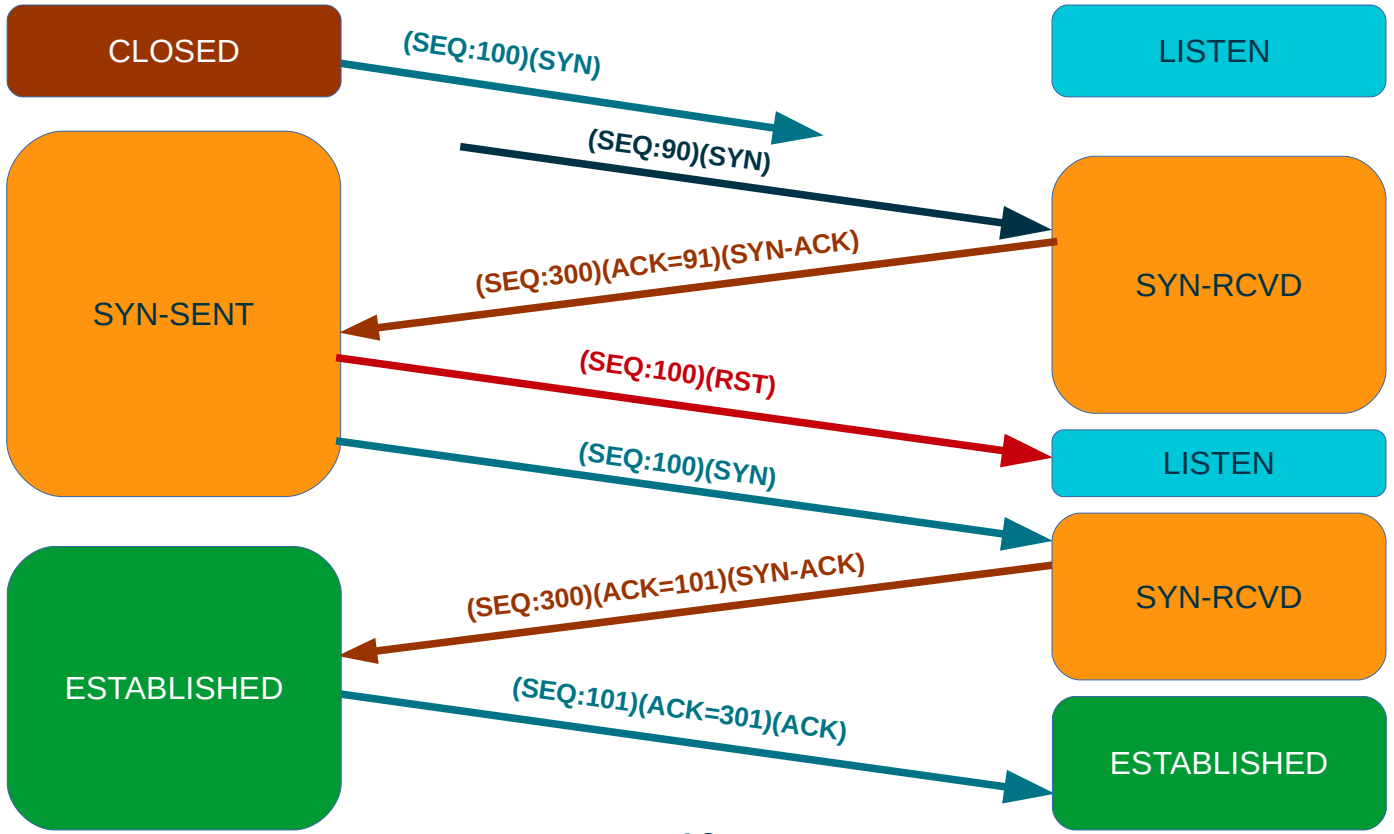
Establecimiento normal de una conexión en tres pasos, en caso de caida, existe un tiempo de silencio



Protocolo TCP

Especificación funcional

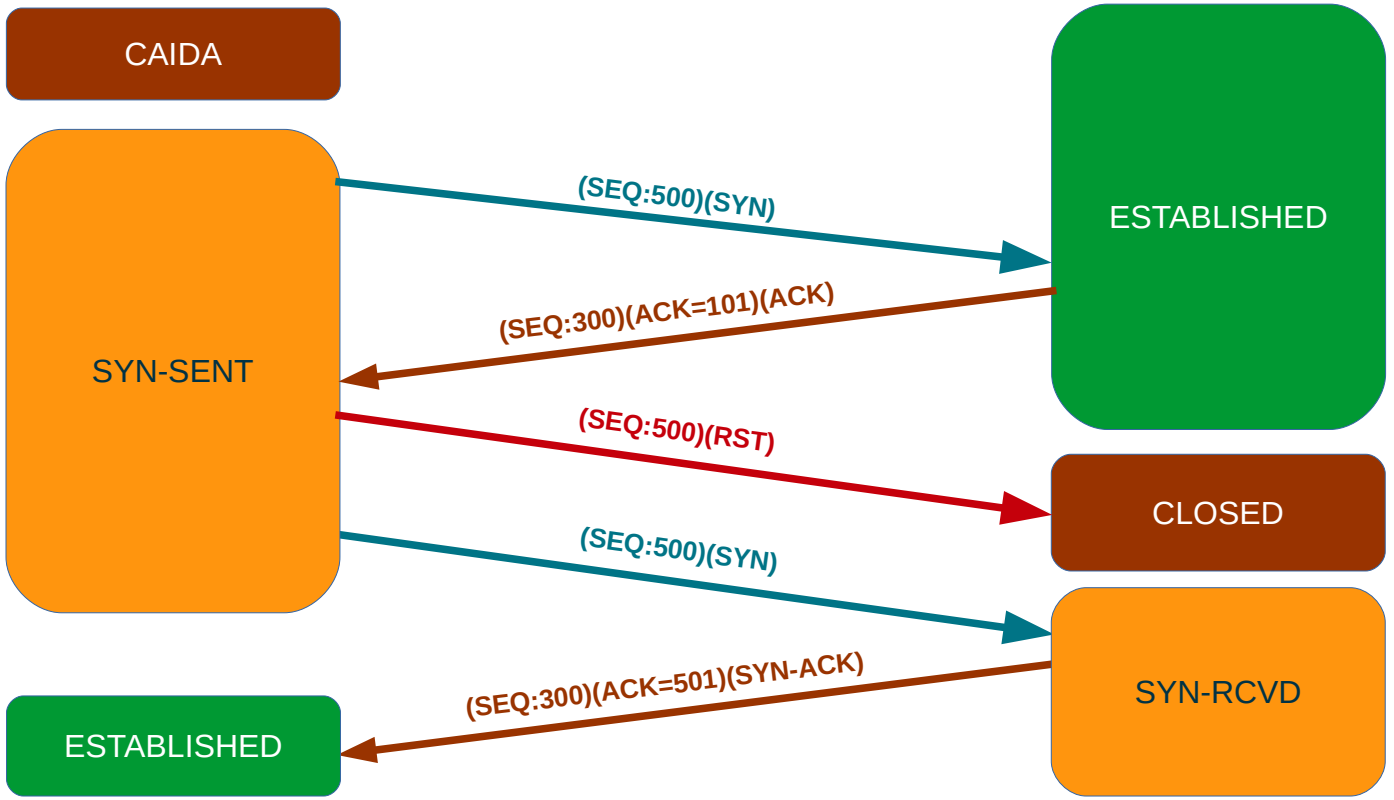
La llegada de un segmento SYN duplicado, hace creer al receptor, que está en un roceso de una conexión simultánea



Protocolo TCP

Especificación funcional

Una conexión, se dice que está medio abierta, si uno de los TCP ha cerrado la conexión por una caída



Protocolo TCP

Especificación funcional

Un RESET, es una señal TCP que se envía con el fin de comnicar al receptor, que queremos romper la comunicación

Como regls general, se envia un RESET siempre que llegue un segmento que no está destinado a la conexión en curso

- Si la conexión no existe y se recibe cualquier segmento
- En estado no sincronizado, se recibe un segmento que confirma uno no enviado
- En estado sincronizado, cualquier segmento inaceptable, provoca el envío de un segmento sin datos, con el número de secuencia y ACK, que espera recibir

Protocolo TCP

Especificación funcional

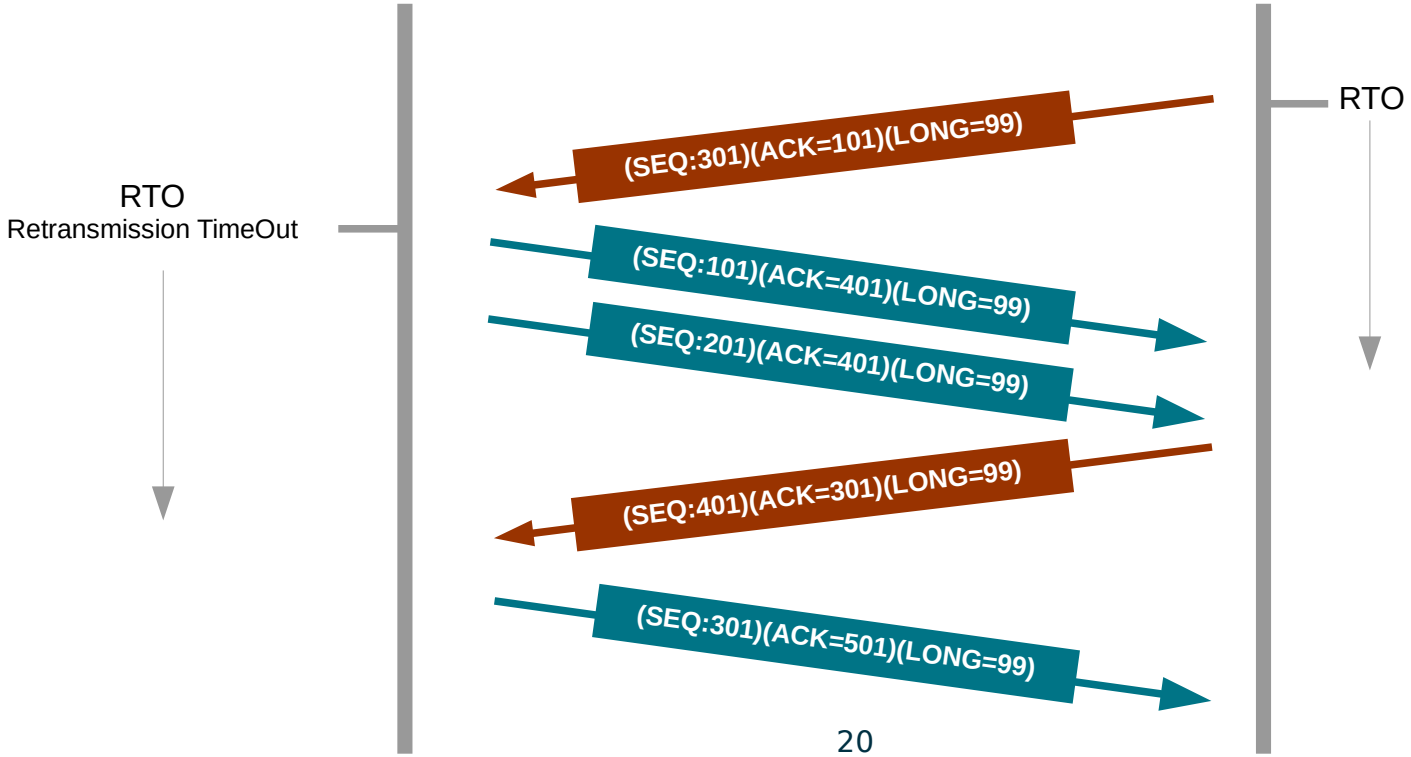
Problemas en el establecimiento de una conexión

- Un SYN respondido con RST, busque problemas en el servidor
- Triple SYN sin respuesta, problema con un firewall o una aplicación que no responde

Protocolo TCP

Especificación funcional

Una vez establecida la comunicación, los datos se transmiten mediante el intercambio de segmentos



Protocolo TCP

Especificación funcional

Una conexión se cierra, cuando un usuario llama a la función CLOSE, puede continuar recibiendo, hasta que el otro extremo cierre con otro CLOSE

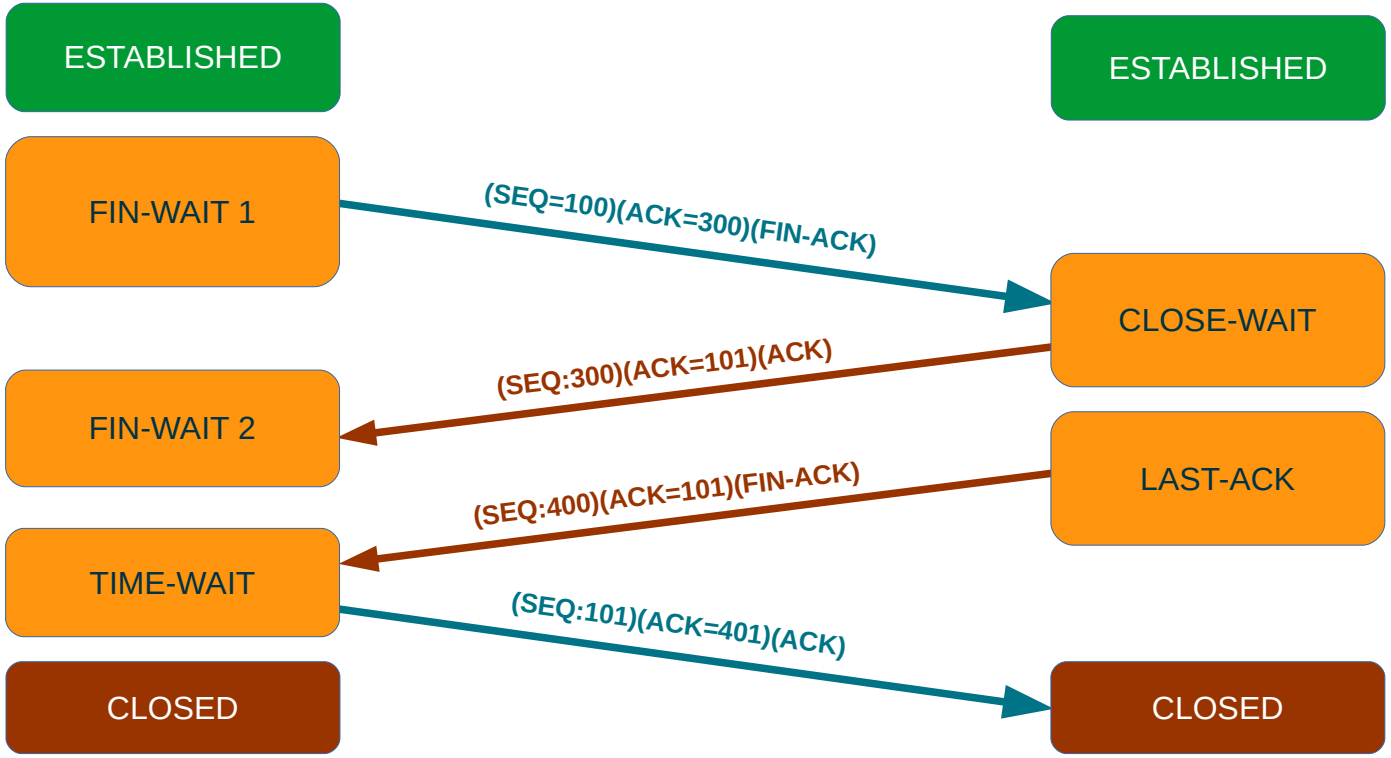
Existen tres casos

- El usuario inicia el cierre de la conexión con CLOSE
- El TCP remoto inicia el cierre enviando un FIN
- Ambos usuarios cierran simultáneamente

Protocolo TCP

Especificación funcional

Cierre normal de una conexión



Protocolo TCP

Laboratorio 1

Ver el intercambio demensajes para el establecimiento de una conexión (handshake) y su finalización

¿Qué le pasa en el cierre al último archivo?

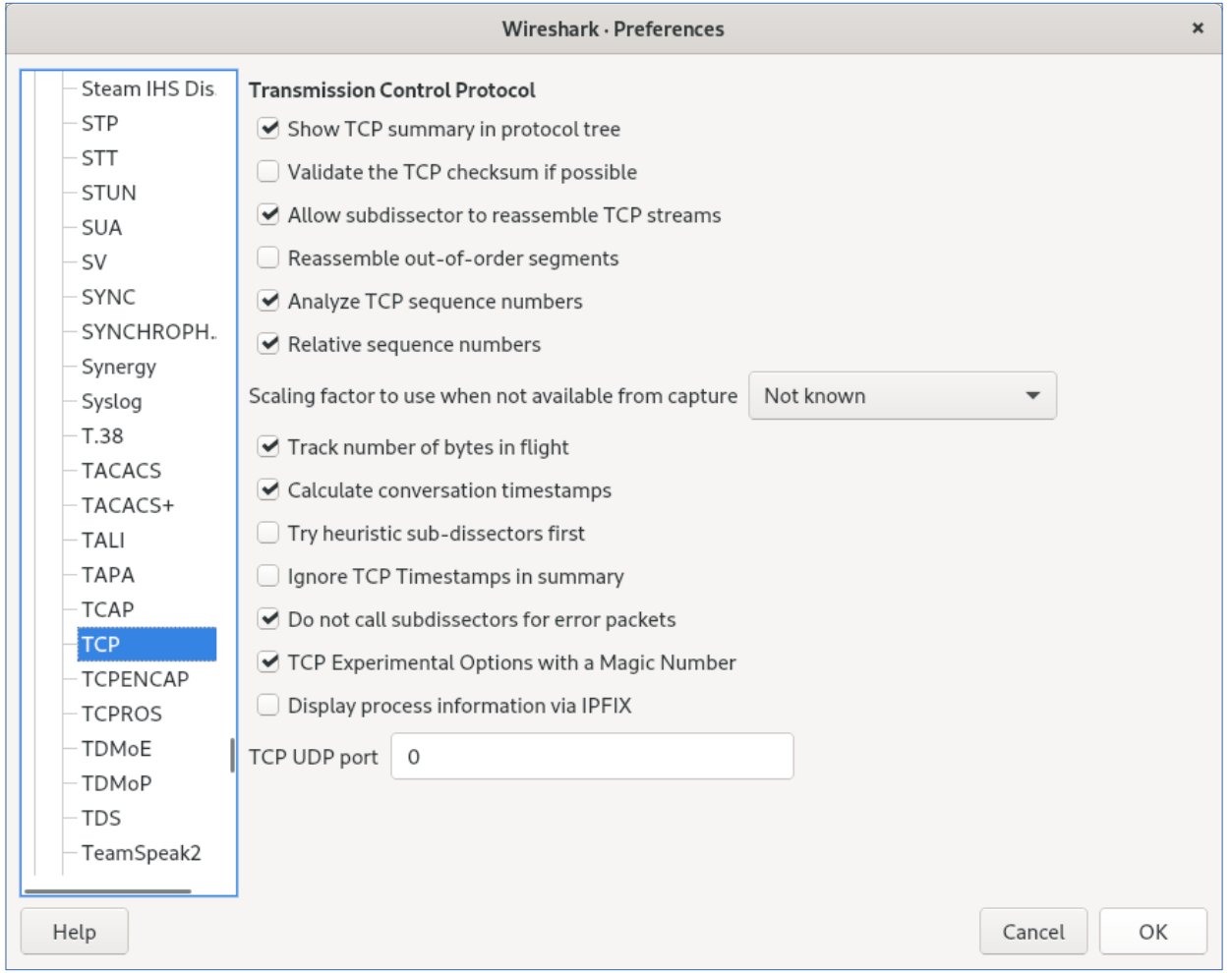
Protocolo TCP

TCP y Wireshark

Protocolo TCP

TCP y Wireshark

Parámetros del protocolo TCP



Protocolo TCP

TCP y Wireshark

Parámetros del protocolo TCP

- Los números de secuencia, son elegidos por el proceso y son dificiles de seguir (el estándar TCP no establece ninguna regla de elección)
- El Timestamp, intenta resolver la sensibilidad de TCP a las variaciones del retardo. El rfc1323, adopta la solución
  - El remitente, pone un timestamp en cada segmento que envía
  - El receptor, refleja estas marcas en segmentos ACK

Protocolo TCP

Control de flujo

Protocolo TCP

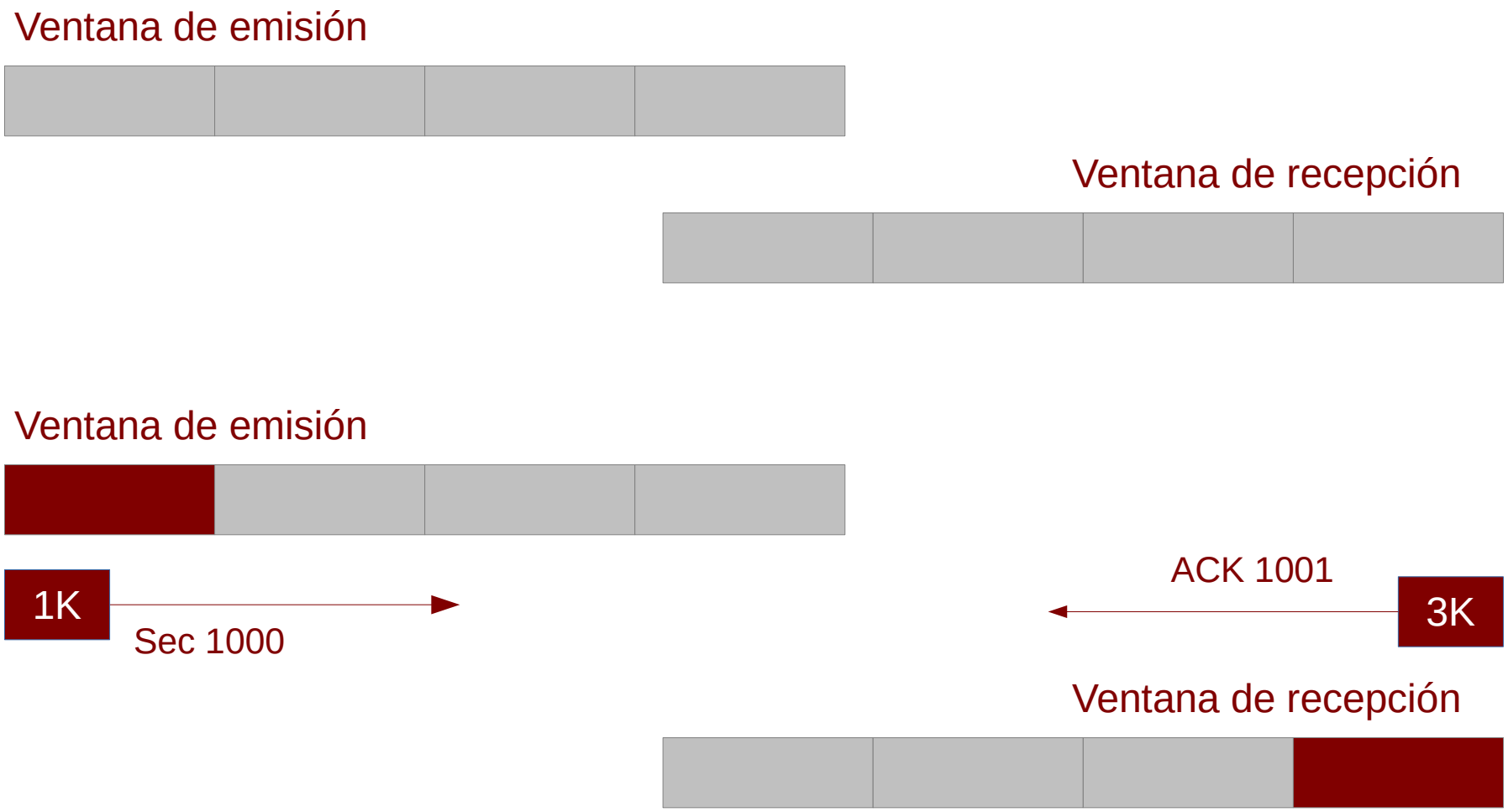
Control de flujo

El mecanismo de control de flujo, establece durante la conexión el espacio de recepción disponible y durante la comunicación, anuncia los datos que puede aceptar en ese momento (ventana de recepción)

- Si llegan más datos de los que pueden ser aceptados, serán descartados
- Cuando el receptor tiene una ventana de tamaño 0 y llega un segmento, debe enviar un ACK con su próximo número de secuencia y su tamaño de ventana actual

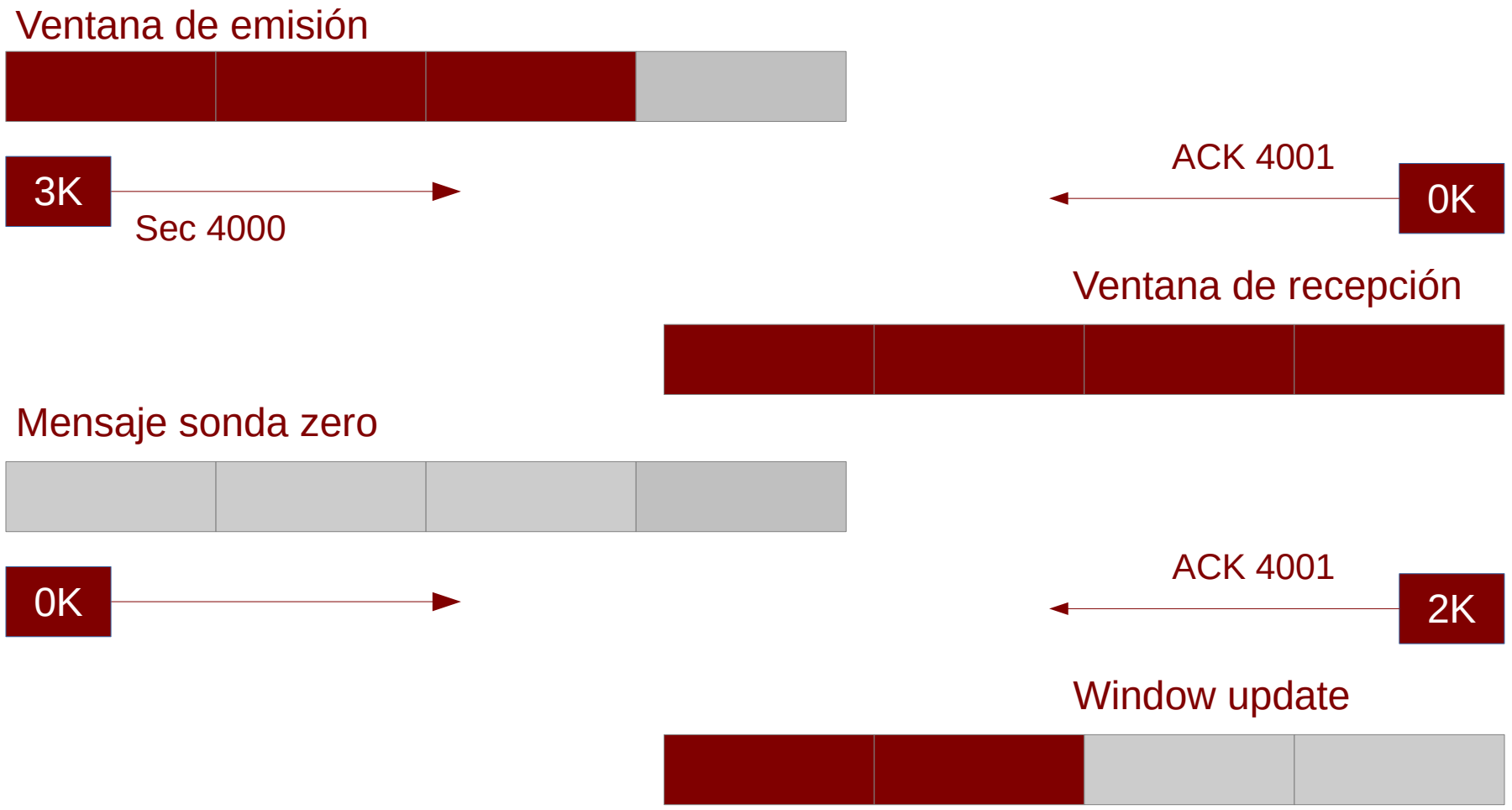
Protocolo TCP

Control de flujo



Protocolo TCP

Control de flujo



Protocolo TCP

Control de flujo

Disponer de una ventana pequeña, provoca que los datos sean transmitidos en segmentos pequeños

Para mejorar la transmisión, hay clientes que realizan ciertas técnicas

- Que el receptor retrase la actualización hasta que el tamaño de la ventana sea de al menos el 20% o 40% del máximo psible
- Que el emisor evite el envío de mensajes pequeños hasta que la ventana sea lo bastante grande para enviar datos

Protocolo TCP

Control de flujo

TCP ventana zero, se produce cuando un receptor, anuncia un tamaño de ventana 0, que le indica al remitente que deje de enviar datos

- Puede ser debido a un equipo con problemas de memoria o capacidad de proceso
- Una aplicación que consume demasiada memoria

Protocolo TCP

Control de flujo

Mensaje **TCP sonda zero**, es enviado por el emisor para ver si todavía existe condición de ventana zero en recepción, si sigue igual, el emisor, doble el temporizador antes de enviar una sonda de nuevo

- Podemos utilizar gráficos TCP para ver problemas de rendimiento

Protocolo TCP

Control de flujo

El tamaño de ventana es de 16 bits, lo que permite anunciar un tamaño máximo de 65535 bytes

Se puede calcular el throughput para un tamaño de ventana TCP con la fórmula

$$Troughput = \frac{TCP\ WindowSize}{RTT}$$

Para una ventana de 65535 bytes en una ruta con RTT de 100 ms, el throughput, es de 5,24 Mbps

Protocolo TCP

Control de flujo

Para que tamaños de ventana más grandes alojen rutas de alta velocidad, la especificación **rfc1323**, define en las opciones de cabecera TCP, un ajuste de escala

- **Window size**, indica ul multiplicador con el tamaño de ventana, para notificar el receptor un bufer de mayor tamaño
- Esta opción, sólo se envía en segmentos SYN durante el proceso de conexión

Protocolo TCP

Laboratorio 2

Problemas de ventana TCP

Estudiar el mecanismo de control de flujo

Comprobar el multiplicador para el tamaño de ventana

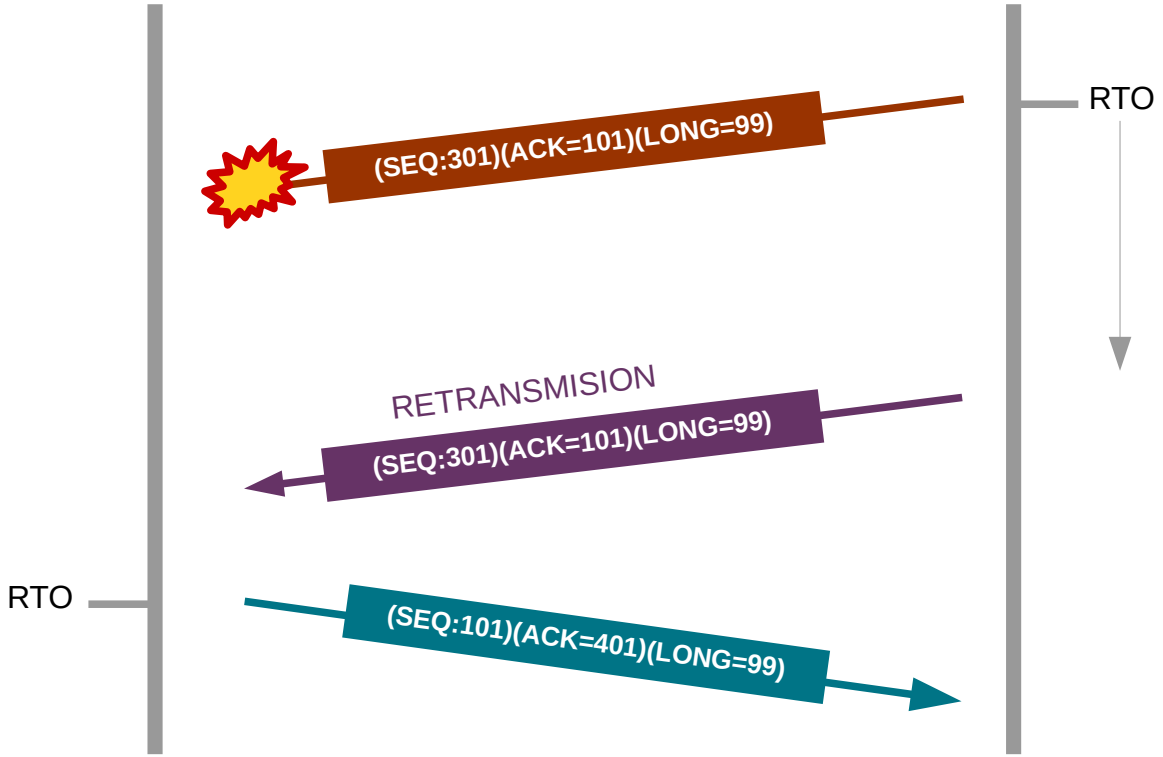
Protocolo TCP

Retransmisiones

Protocolo TCP

Retransmisiones

TCP utiliza la retransmisión (tras un tiempo de espera) para asegurar la entrega de cada segmento



Protocolo TCP

Retransmisiones

Control de errores, retransmisiones. Un exceso de retransmisiones puede volver lenta la red

- Las retransmisiones ocurren cuando un paquete no ha llegado o un reconocimiento no ha llegado a tiempo
- La primera retransmisión, se produce al expirar un temporizador (RTO), las siguientes, duplican el tiempo hasta alcanzar un máximo de 5 retransmisiones
- Si se recibe 3 ACKs duplicados, no hace falta que expire RTO

Protocolo TCP

Retransmisiones

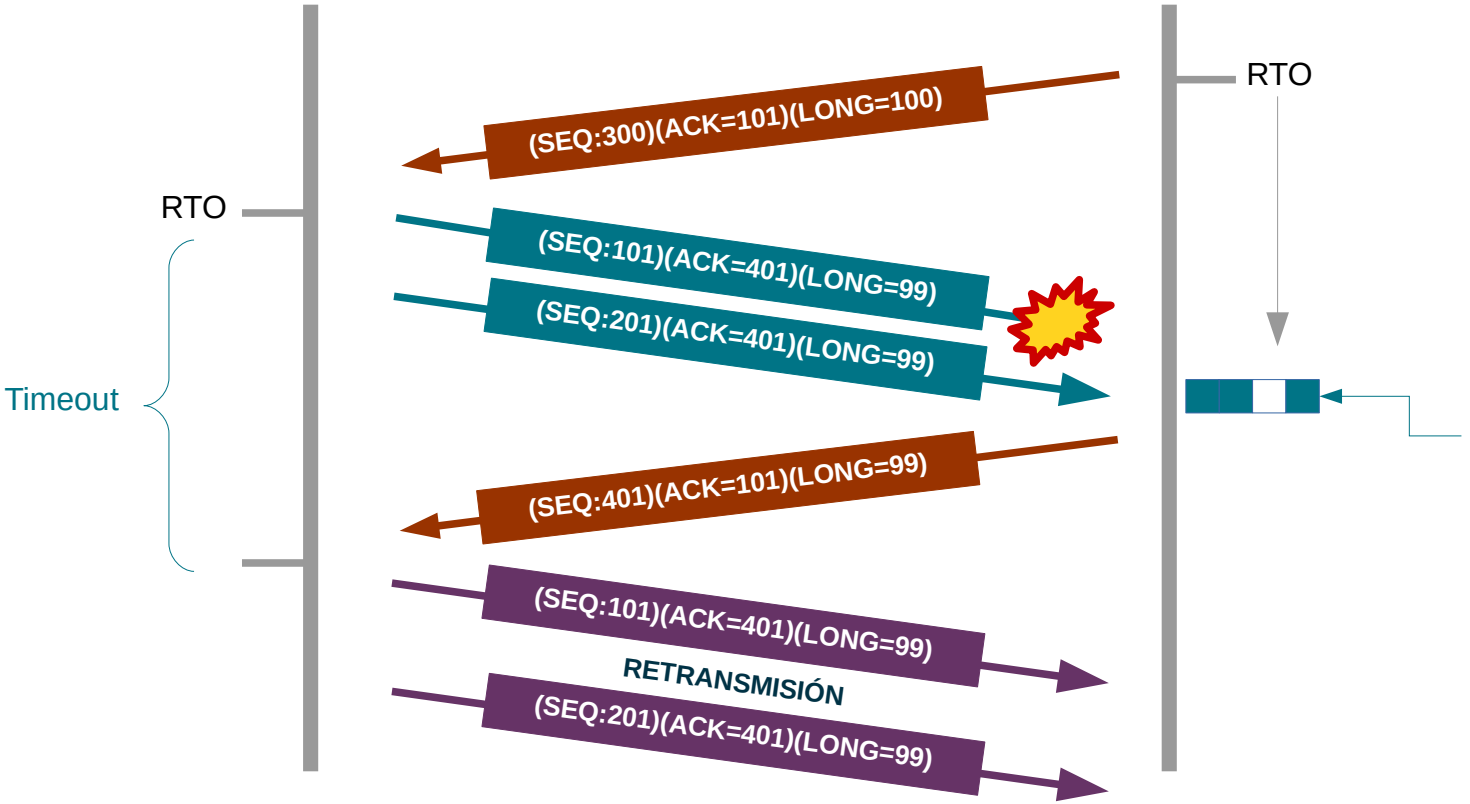
La elección del tiempo de vencimiento del temporizador de retransmisión, está basada en los retardos observados en la red

Los retardos, pueden variar dinámicamente, por tanto, los temporizadores, deben adaptarse a esta situación y se recalculan mediante diversos algoritmos

Protocolo TCP

Retransmisiones

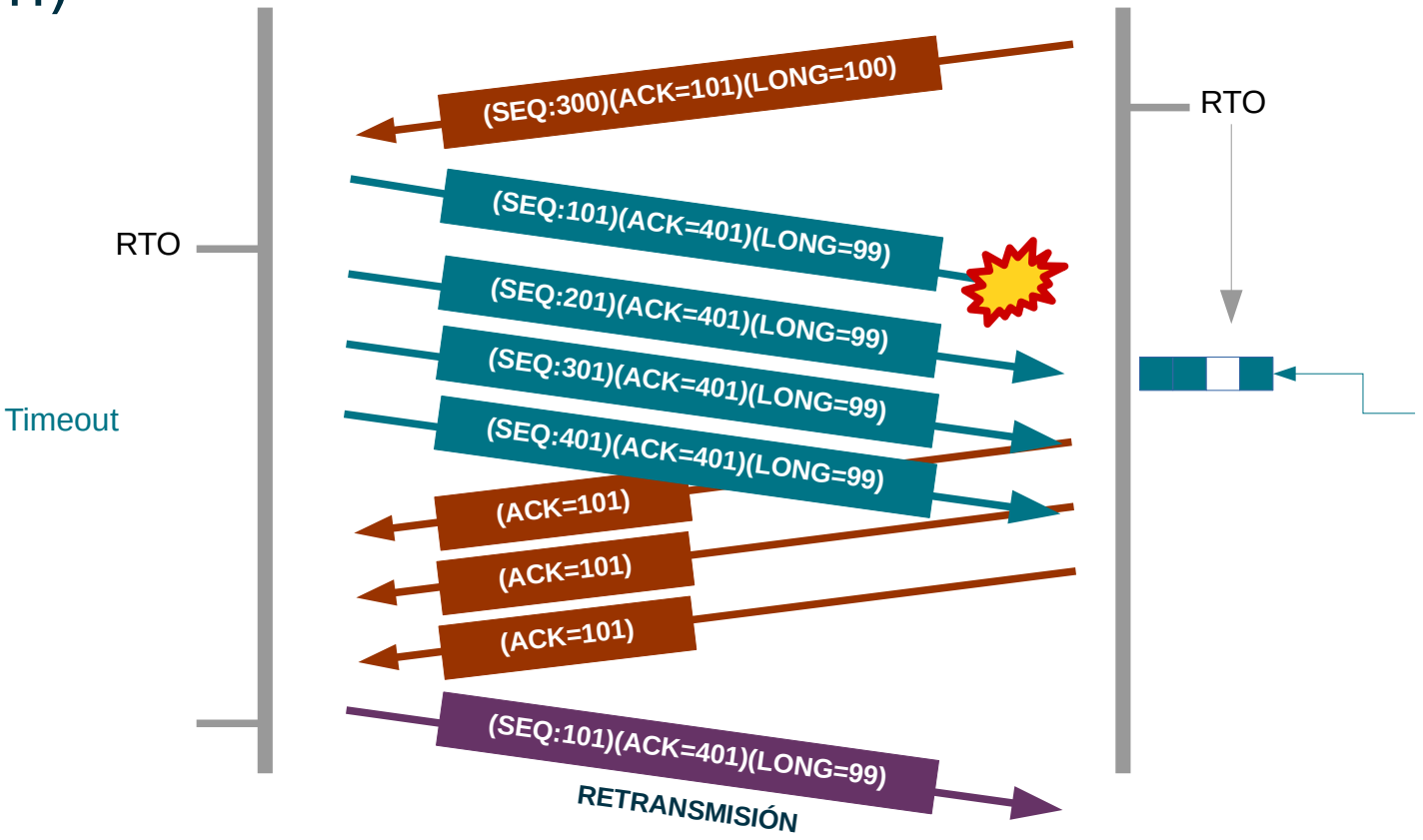
Perdida de un segmento y vencimiento de RTO



Protocolo TCP

Retransmisiones

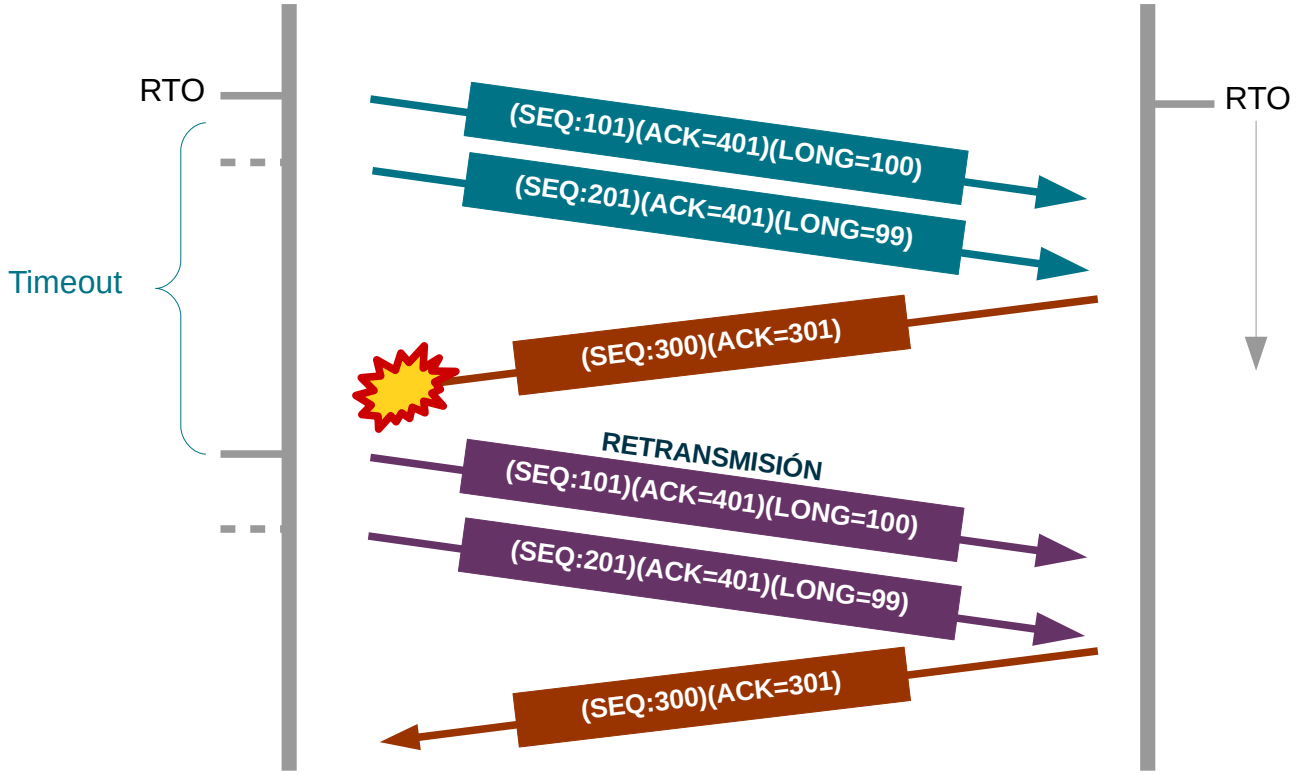
Perdida de un segmento y recepción de 3 ACKs seguidos (Fast retransmission)



Protocolo TCP

Retransmisiones

Perdida de un ACK y vencimiento de RTO



Protocolo TCP

Retransmisiones

Después de 5 retransmisiones seguidas, la conexión se considera perdida

- Se envía un SYN intentando establecer una nueva conexión
- No se envía SYN y el usuario, deberá volver a correr la aplicación

Protocolo TCP

Retransmisiones

En consecuencia, las retransmisiones son un comportamiento natural, siempre que no hayan demasiadas

- Ubicar el problema en la dirección IP, conexión o aplicación concreta
- Comprobar si es debido a la perdida de paquetes, a un servidor o a una aplicación lenta
- Comprobar si hay variaciones en el retardo

Protocolo TCP

Laboratorio 3

Retransmisiones TCP

Estudiar el mecanismo de retransmisiones y comprobar el temporizador entre ellas

¿Qué se supone que debe pasar?

Protocolo TCP

Selective ACK

Protocolo TCP

Selective ACK

En las opciones de la cabecera TCP, puede haber la siguiente opción

- **SACK** (selective ACK), permite reconocer paquetes específicos, ambas partes tienen que ponerse de acuerdo

Protocolo TCP

Selective ACK

PCP SACK, usa los reconocimientos para que el emisor, tome decisiones inteligentes respecto a las retransmisiones. Utiliza dos opciones

- **SACK** permited, utilizada para habilitar la opción SACK en el establecimiento de la conexión
- SACK, utilizado para informar de los bloques de datos recibidos que no son contiguos

Protocolo TCP

Selective ACK

Funcionamiento de SACK

Protocolo TCP

Laboratorio 4

Retransmisiones selectivas TCP

Comprobar el funcionamiento de SACK

Protocolo TCP

ACKs duplicados

Protocolo TCP

ACKs duplicados

Cuando tenemos variaciones en el retardo, también se pueden esperar retransmisiones

Una demora, puede ocurrir debido a

- La inestabilidad de una línea
- Una aplicación saturada o ineficiente
- Un equipo sobrecargado

Protocolo TCP

ACKs duplicados

En la mayoría de los casos, ACKs duplicados, se deben a la alta latencia, las variaciones del retardo o un host lento

- Uno o dos ACKs duplicados, implican desorden en la recepción
- Tres ACKs duplicados, implica la perdida del paquete y el emisor, vuelve a enviar el paquete (retransmisión rápida)

Protocolo TCP

Laboratorio 5

ACKs duplicados

Comprobar cual es el motivo para que se produzca una retransmisión

Protocolo TCP

Latencia y retardos

Protocolo TCP

Latencia y retardos

La latencia, es una medida utilizada para definir retardo de tiempo,esta, puede ser causada por

- Problemas a a lo largo de una ruta
- Problemas en el cliente
- Poblemas en el servidor

Protocolo TCP

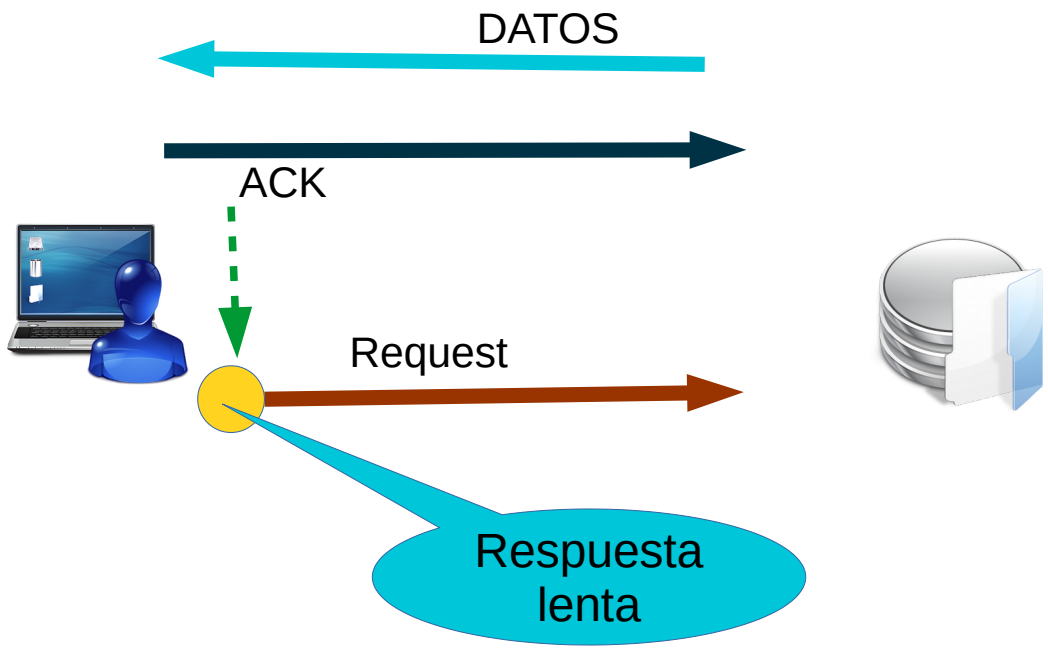
Latencia y retardos

La latencia en ruta, se conoce como **Round Trip Time** (RTT, tiempo de ida y vuelta)

Protocolo TCP

Latencia y retardos

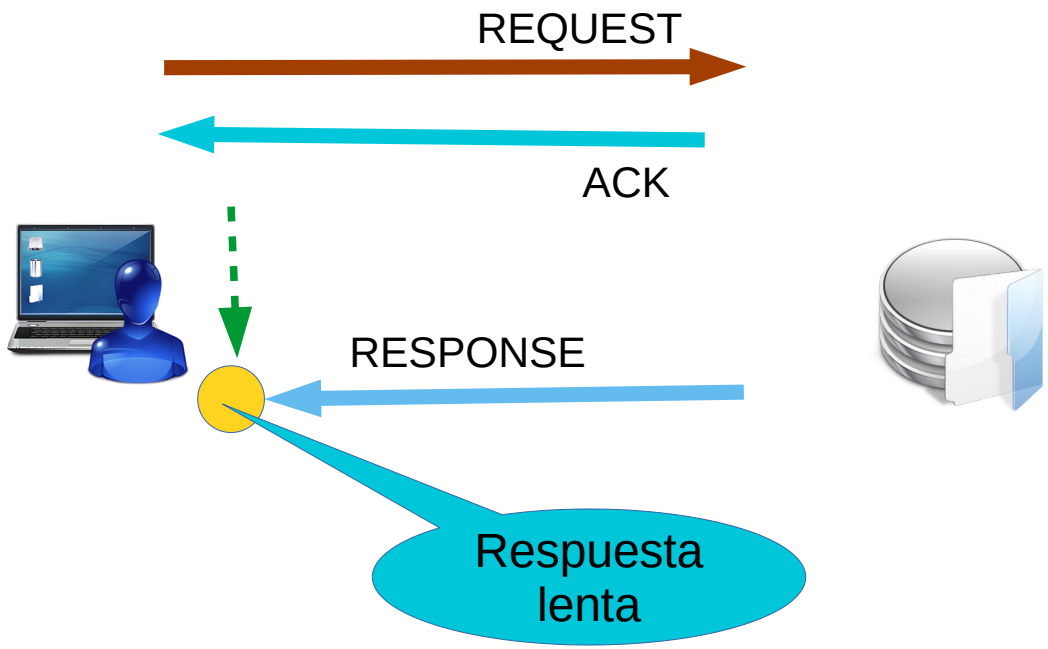
La latencia en ruta, se conoce como **Round Trip Time** (RTT, tiempo de ida y vuelta)



Protocolo TCP

Latencia y retardos

La latencia de servidor, se produce cuando tarda en responder a la recepción de una solicitud



Protocolo TCP

Latencia y retardos

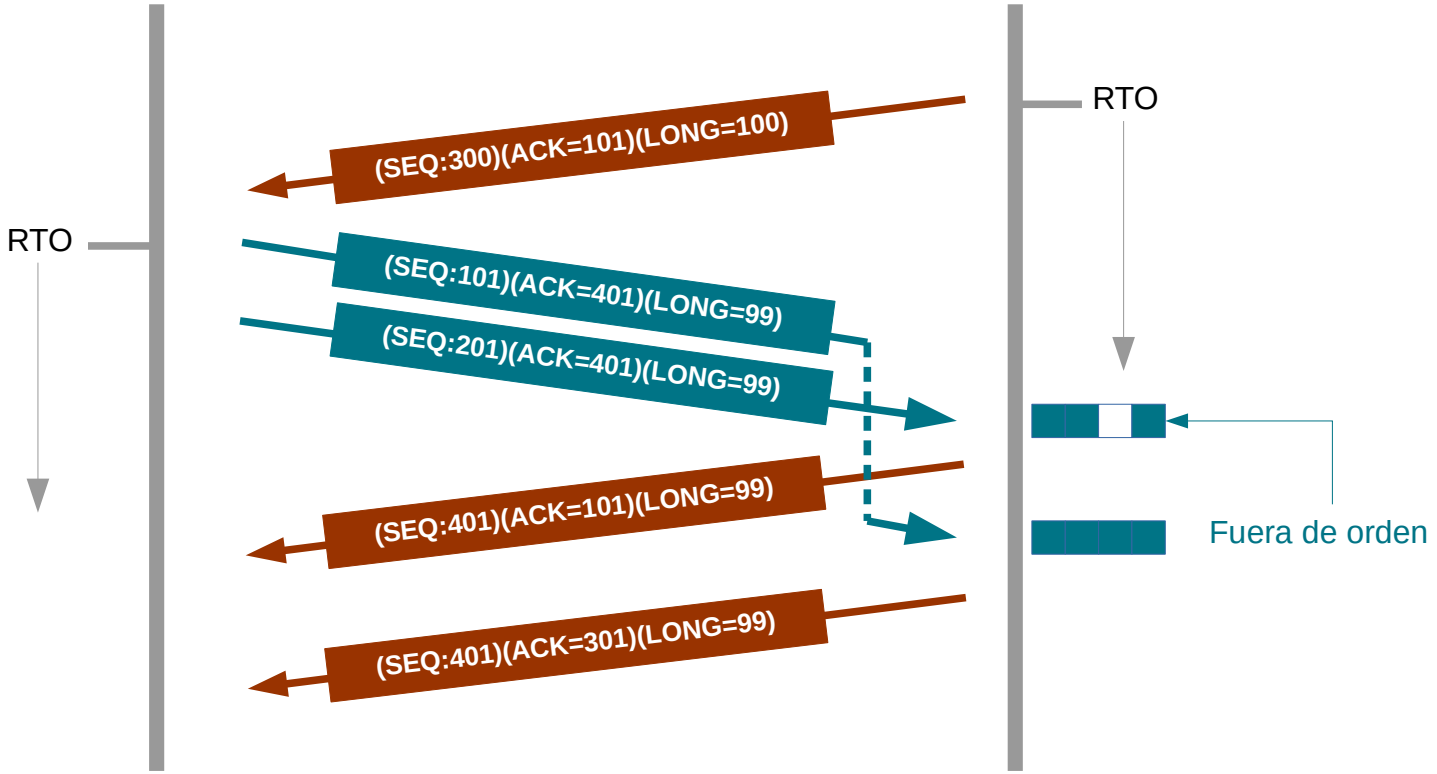
Podemos encontrarnos perdidas de paquetes y segmentos fuera de orden

- Puede ser debido a diferentes rutas de origen al destino o a perdida de paquetes
- Puede ser que Wireshark, no las haya capturado
  - Tráfico muy pesado
  - La máquina desde la que se toman las capturas no es muy potente
  - Los switchs pueden descartar paquetes
  - Captura de datos en una red inalámbrica

Protocolo TCP

Latencia y retardos

Recepción fuera de orden



Protocolo TCP

Latencia y retardos

Los motivos para perder un paquete

- Congestión en un router
- Error en una línea
- Error en un paquete

Protocolo TCP

Latencia y retardos

El ajuste por defecto de la columna Time, es desde el comienzo de la captura y en segundos

Para seleccionar el tiempo desde el final de un paquete hasta el final del siguiente, seleccione

- **View → Time Display Format → Seconds since previous displayed packet**

Protocolo TCP

Latencia y retardos

Detectar la latencia con una columna **TCP Delta**, una vez que se ha habilitado la opción “**Calculate Conversation Timestamp**”

Creamos una columna desde el parámetro “**Time since previous frame**” de TCP

Protocolo TCP

Latencia y retardos

¿Qué retrasos se pueden considerar como normales?

- Solicitudes de archivos “ico”
- Paquetes SYN
- Paquetes FIN, FIN-ACK, RST
- Paquetes GET
- Consultas DNS

No.	Time	TCP Delta ▲	Source	Destination	Protocol	Info
471	20.675453000	18.30530200(	24.6.173.220	173.194.79.82	HTTP	GET /svn/trunk/image/p-expand.gif HTTP/1.1
470	20.675046000	18.30340000(	24.6.173.220	173.194.79.82	HTTP	GET /svn/trunk/image/16x16/User.png HTTP/1.1
458	20.611551000	18.22969800(	24.6.173.220	173.194.79.82	HTTP	GET /svn/trunk/image/throbber.gif HTTP/1.1

Protocolo TCP

Laboratorio 6

Trabajar con Timestamp para comprobar retrasos entre tramas

Ajustar tiempo a Seconds since previous displayed packet

Añadir una columna Delta Time

Protocolo TCP

Debilidades TCP

Protocolo TCP

Debilidades TCP

Antes de la planificación de un ataque, es necesario conocer el objetivo que se va a atacar y para ello, hay que obtener información

- Utilización de herramientas de administración
- Descubrimiento de usuarios
- Información de dominio
- Cadenas identificativas

Protocolo TCP

Debilidades TCP

La exploración de puertos, permite el reconocimiento de los servicios ofrecidos en los equipos encontrados

Técnicas de exploración de puertos

- TCP connect scan
- TCP SYN scan
- TCP FIN scan
- TCP xmas tree scan
- TCP NULL scan

Protocolo TCP

Debilidades TCP

Para UDP, se envían datagramas sin ninguna información en el campo de datos, si está cerrado, se recibe ICMP.port_unreachable

Protocolo TCP

Debilidades TCP

Nmap, implementa la gran mayoría de técnicas asociadas a la exploración de puertos y permite descubrir información de los servicios encontrados

Protocolo TCP

Debilidades TCP

Ejemplos de nmap

```
nmap -sP 192.168.101.10 → Sólo ping
nmap -sS 192.168.101.10 → TCP SYN
nmap -sT 192.168.101.10 → TCP Connect
nmap -sU 192.168.101.10 → UDP
nmap -sN 192.168.101.10 → TCP Null
nmap -sF 192.168.101.10 → TCP FIN
nmap -sA 192.168.101.10 → TCP ACK
nmap -sX 192.168.101.10 → TCP Christmas Tree
```

Protocolo TCP

Debilidades TCP

La forma en la cual se muestra el estado de los puertos es

- **OPEN**, indica que la máquina remota está en estado **LISTEN**
- **CLOSED**, no hay ninguna aplicación escuchando en ese puerto
- **FILTERED**, indica que un firewall, está bloqueando el acceso a ese puerto

Protocolo TCP

Debilidades TCP

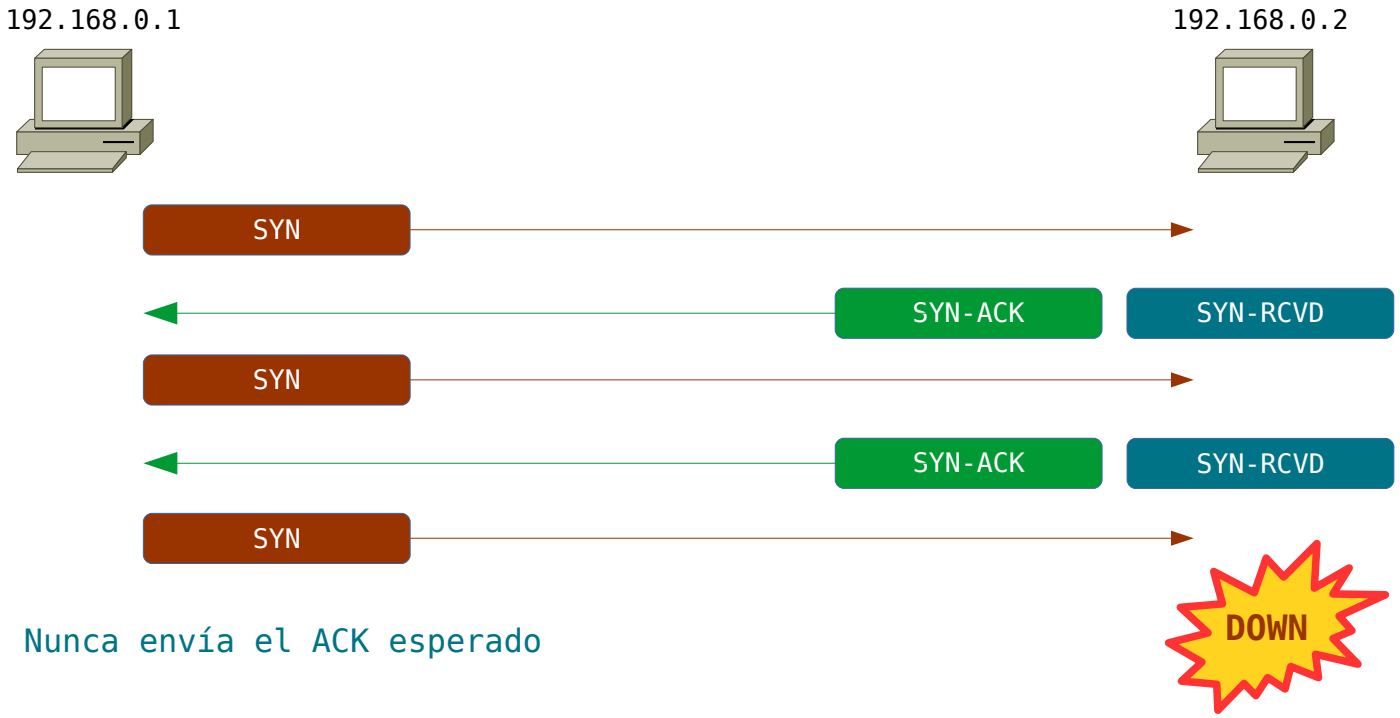
Defiimos denegación de servicio (DoS), como la imposibilidad de acceder a un recurso o servicio po parte de un usuario legítimo

- TCP SYN flooding
- Connect flood

Protocolo TCP

Debilidades TCP

Ataques de denagación de servicio TCP SYN flooding, envían masivamente paquetes de establecimiento de conexión (SYN), el destino, se queda sin recursos



Protocolo TCP

Debilidades TCP

Connection flod, los servicios TCP orientados a conexión (Telnet, HTTP, SSH, etc.), tienen un limite máximo de conexiones simultáneas soportadas, cuando ese límite se alcanza, cualquier conexión nueva, es rechazada

Las conexiones iran caducando por inactividad, el atacante, sólo tiene que intentar nuevas conexiones

Protocolo TCP

Debilidades TCP

Un ataque DdoS es aquel en el que una multitud de sistemas cooperan entre sí para atacar a un único objetivo, causando una denegación de servicio

Se observan garn acantidad de segmentos con el flag SYN, que no reciben respuesta

Protocolo TCP

Laboratorio 7

Estudio de un escaneo de puertos realizado con la herramienta nmap. ¿Que puertos estan abiertos?


Telefónica