

UTN-FRBA-Dto.Sistemas Redes de Información

Unidad 5: Protocolos TCP/IP Clase 2: Transporte

Fuentes: Stallings, cap. 18 y 20-Comer, vol.1
TCP/IP Guide
Versión: 1

Capa 4: Transporte

- Hay sólo dos protocolos para distintas aplicaciones
- Transmission Control Protocol (TCP)
 - Orientado a conexión
 - RFC 793
- User Datagram Protocol (UDP)
 - Sin conexión
 - RFC 768

2

Modelo Cliente Servidor

- Usado para el procesamiento distribuido
- Una aplicación (Cliente) pide servicios a otra (Servidor)
- El cliente usa un port conocido, mientras que el cliente usa uno provisorio
- Se crea un socket (par dirección/port)
- La aplicación servidor:
 - debe estar arrancada
 - debe autenticar el acceso del cliente
 - puede aceptar múltiples pedidos

3

Interfaz de socket

- Los protocolos están embebidos en el sistema operativo
- Las aplicaciones deben usar una API (*Application Program Interface*) para acceder a los protocolos
- La norma TCP/IP define funciones pero no da nombres ni argumentos, y cada sistema operativo define los detalles
- La interfaz de socket apareció en el BSD de Unix y fue adaptada en el Windows Sockets
- Los pasos son:
 - Crear un socket
 - Definir propiedades
 - Transferir datos haciendo lectura y escritura del socket
 - Cerrar el socket

4

TCP

- TCP (Transmission Control Protocol) es el más utilizado en Internet
- Hace una entrega confiable, para lo cual antes se usaba el Reconocimiento Activo con Reconfirmación:
 - Cuando envía un paquete, el trasmisor arranca un timer
 - Cuando recibe el paquete, el receptor manda un mensaje de recibo
 - Si el trasmisor no lo recibe dentro del tiempo, retransmite el paquete
- Desperdicia ancho de banda porque debe esperar reconocimiento antes de enviar otro paquete
- Se soluciona con el método de ventanas deslizantes que aprovecha al máximo el canal

5

Uso de ports

- TCP usa número de ports para identificar la aplicación
- La conexión entre dos terminales tiene cuatro identificaciones
 - Dirección fuente
 - Port fuente
 - Dirección destino
 - Port destino
- Un mismo port puede ser compartido por varias conexiones
- Modo de enlace TCP:
 - Pasivo abierto: un servidor espera pedidos
 - Activo abierto: un cliente inicia pedido

6

Asignación de número de port

- TCP y UDP asignan números de port típicos para cada aplicación que transportan
- Los números bajos (hasta 1.023) son universales (conocidos) y los usan los servidores
- Los números medios (hasta 49.151) deben ser registrados ante el IANA
- Los números altos (hasta 65.535) son privados y los usan los clientes en forma dinámica mientras dura la sesión

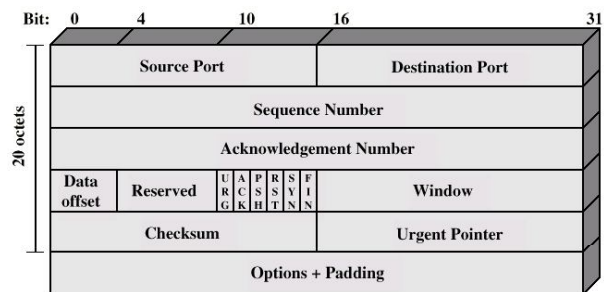
7

Servicios TCP

- Provee una comunicación confiable entre extremos
- El mensaje cruza redes confiables y no confiables
- Dos formas de tratar casos especiales:
 - Data stream push
 - El usuario pide transmitir todos los datos hasta el flag push
 - El receptor los entrega de la misma forma
 - Evita la espera hasta llenar el buffer
 - Urgent data signal
 - Indica que están llegando datos urgentes
 - La aplicación decide cómo proceder

8

Encabezamiento TCP



9

- El N° de secuencia es aleatorio para permitir varias sesiones, y cada extremo lo elige al comienzo
- La ventana indica los octetos que puede recibir
- Significado de los FLAGS:
 - URG : VALIDO CAMPO DE PUNTERO URGENTE (ENTREGAR HASTA ESTE PUNTO)
 - ACK : VALIDO CAMPO ACK.
 - PSH : REALIZAR OPERACION PUSH Y ENTREGAR DATOS
 - RST : INICIO DE LA CONEXION.
 - SYN : SINCRONIZAR NUMERO SECUENCIA.
 - FIN : FIN FLUJO DE BYTES DEL TX.

10

Items Pasados a IP

- TCP pasa algunos parámetros al IP
 - Precedencia
 - Retardo normal/bajo
 - Flujo normal/alto
 - Confiabilidad normal/alta
 - Seguridad

11

Mecanismos TCP

- Establecimiento de la conexión
- Transferencia de datos
- Terminación de la conexión

12

Establecimiento

- Normalmente la inicia el cliente
- Forma de establecer la conexión
 - Se establece entre pares de puertos
 - Un puerto origen puede conectarse a múltiples destinos

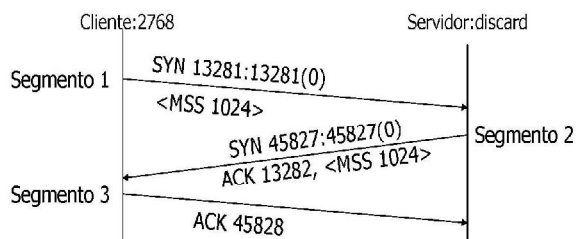
13

Proceso de establecimiento

- El emisor (cliente) envía un segmento SYN (segmento 1) especificando:
 - el N° de puerto del servidor al que quiere conectarse
 - el N° de secuencia inicial.
- El servidor responde con su propio segmento SYN que contiene el N° de secuencia inicial del servidor (segmento 2).
- También asiente (ACK) el SYN del cliente + 1 (los mensajes SYN consume un N° de secuencia).
- El cliente asiente el SYN del servidor con un N° de ACK igual al N° de secuencia del servidor + 1 (segmento 3).

14

Esquema de Establecimiento



15

- Transferencia de datos
 - Es una secuencia de octetos
 - Octetos numerados módulo 2^{23}
 - Control de flujo por asignación de créditos por el número de octetos
 - Datos en *buffers* en transmisor y receptor
- Datos para entrega urgente a aplicaciones:
 - Se pone en uno el bit "Urgente"
 - Se envía octeto apuntando el puntero de datos urgente
- Retransmisión rápida
 - Para casos especiales cambia el clock
 - Transmisor avisa con tres ACK duplicados
 - Además baja ventana de congestión a la mitad

16

- Terminación de la conexión
 - Cierre ordenado
 - Usuario manda primitiva *CLOSE*
 - Transporte pone flag *FIN* en el último segmento
 - Usuario termina abruptamente con primitiva *ABORT*
 - Transporte abandona intentos de transmitir o recibir
 - Manda segmento *RST*

17

- Se intercambian 4 segmentos para cerrar una conexión.
- Una conexión TCP es full-duplex y cada dirección se cierra independientemente.
- Cada extremo envía un FIN cuando a finalizado el envío de datos
- El otro extremo puede continuar enviando datos (half-close).
- El extremo que envía el primer FIN realiza el cierre activo, y el otro extremo el cierre pasivo.
- Cualquiera de los dos extremos puede iniciar el cierre.

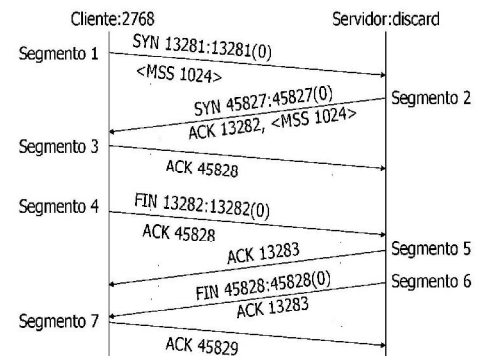
18

Protocolo de finalización de conexión:

- El cliente finaliza la aplicación: envía un FIN (segmento 4) con el número de secuencia correspondiente (cierra el flujo de datos cliente a servidor).
- El servidor responde con un ACK (segmento 5) del n° de secuencia + 1 (los mensajes FIN consumen un n° de secuencia).
- A continuación, el servidor envía un FIN (segmento 6).
- El cliente confirma la recepción del FIN, con un ACK del n° de secuencia recibido + 1 (segmento 7).

19

Esquema de Finalización



20

Timeout

- Tiempo que ha de transcurrir para que un cliente tratando de establecer una conexión indique que dicha conexión no puede ser establecida.
- No se especifica el motivo en las aplicaciones estándares.
- Este timeout varía de unas implementaciones a otras.
- En el caso del UNIX BSD y derivados, este timeout vale 75 segundos en los que se suelen mandar entre 3 y 5 paquetes de establecimiento.
- Funciona en base a ticks de 500 mseg.

21

Segmentos de reset

- Un segmento es de Reset cuando se activa en la cabecera TCP el flag RST.
- Se activa el bit de Reset en una conexión TCP cuando el paquete que ha llegado no parece, en principio, estar relacionado con la conexión a la que está referido el paquete.
- Las causas de generar un paquete con este bit para una conexión TCP pueden ser varias:
 - Intento de conexión a un puerto no existente
 - Abortar una conexión
 - Respuesta ante conexiones semi-abiertas

22

Control de flujo

- El receptor controla el flujo avisando al transmisor el tamaño del buffer libre medido en bytes
- Cuando se libera el espacio manda un reconocimiento
- TCP hace checksum incluyendo encabezamiento IP
- Retransmisiones
 - En los enlaces hay retardo variable
 - Esperar mucho retarda transmisión
 - Esperar poco desperdicia ancho de banda
 - Se mide el RTT (*round trip time*) del enlace para ajustar espera
 - Como el RTT puede variar, se usa un promedio ponderado entre el valor medido anteriormente y el nuevo (*smoothing*)

23

- Para medir el retardo se verifica la recepción del ACK, que acepta datos, no una transmisión
- Asociando erróneamente el ACK a la primera o última transmisión causa diferencias en la optimización y baja rendimiento
- Algoritmo de Partridge/Karn ignora las retransmisiones
- Control de flujo:
 - El receptor indica su ventana, que es la cantidad de bytes que puede recibir
 - Hay un valor mínimo declarado de la ventana
 - El transmisor puede bajar la ventana por debajo del valor declarado si detecta congestión

24

Algoritmo de Jacobson/Karel

- Si hay congestión se generan retardos, se pierden tramas y hay retrasmisiones, por lo que reduce la ventana
- La ventana de congestión será el mínimo entre el valor declarado y el calculado
- En tráfico normal la ventana de congestión es igual a la del receptor
- Si se pierde un segmento, se reduce la ventana a la mitad, y sigue sucesivamente así hasta un mínimo
- Arranque lento: cuando inicializa usa la ventana mínima y la incrementa en un segmento cada vez que recibe un ACK

25

- Cuando detecta congestión reduce la ventana a la mitad y la define como umbral
- Durante la recuperación usa arranque lento hasta llegar al umbral
- Por encima del umbral incrementa la ventana de congestión en un segmento por vez
- Los algoritmos reproducen el comportamiento en la realidad y van ajustando los parámetros ante los cambios

26

UDP

- *User datagram protocol*
- RFC 768
- Servicio sin conexión para aplicaciones
 - No es confiable
 - No garantiza entrega ni duplicación
- Simple
 - Reducido encabezamiento
 - Cada operación de salida genera un único datagrama
- Se aplica en gestión de red
 - opera en capa 4
 - no es orientado a la conexión
- No se ocupa de asegurar la entrega de paquetes
 - la integridad de los mensajes es problema de la aplicación

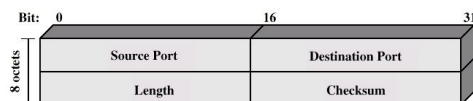
27

Usos de UDP

- Recolección de datos
- Difusión periódica de datos
- Pedidos y respuestas
- Tráfico escaso (como DNS)
- Aplicaciones de tiempo real que no esperan ACK
- Medio altamente confiable

28

Encabezamiento UDP



- N° de puerto son distintos que TCP
- Longitud UDP es el largo de cabecera y los datos (redundante con información de cabecera IP)

29

Cálculo del checksum

- Para calcular el checksum del encabezamiento se arma un pseudo encabezamiento con información del datagrama IP:

0	8	16	31
DIRECCION IP DE LA FUENTE			
DIRECCION IP DEL DESTINO			
CERO	PROTOCOLO	LONGITUD UDP	

30

Comparación entre protocolos

- | • UDP | TCP |
|-------------------------------|----------------------|
| envía datagramas | envía segmentos |
| no confiable | confiable |
| checksum opcional | checksum requerido |
| sin conexión | con conexión |
| para LAN | para LAN y WAN |
| sin control de flujo | con control de flujo |
| múltiples intercambios | sólo punto a punto |
| unicast, multicast, broadcast | sólo unicast |
- El 95% del tráfico de Internet usa TCP