Unidad 3 / Escenario 5

Lectura fundamental

Protocolos de transporte

Contenido

Palabras clave: capa de transporte, multiplexación, demultiplexación, orientado y no a conexión, TCP y UDP.

La capa de transporte, sin importar si es confiable o no, debe implementar la comunicación lógica entre los procesos de dos máquinas diferentes. Para lograr esto, la capa de transporte debe estar en capacidad de identificar, para un paquete de red dado, a qué proceso se debe entregar (demultiplexación) y así mismo debe estar en capacidad de tomar los datos de diferentes procesos y enviarlos por un solo canal adicionándole los datos necesarios para que del otro lado se pueda entregar al proceso adecuado (multiplexación).

A continuación se tratarán principalmente dos protocolos de transporte TCP y UDP, de una manera general con el fin de aclarar conceptos sobre los mismos.

1. Protocolos de transporte

Los protocolos de transporte que predominan en la actualidad son: UDP (User Datagram Protocol), y TCP (Transport Control Protocol).

1.1. UDP (User Datagram Protocol)

UDP permite entregar segmentos de datos desde un proceso en una máquina emisora a otro en una máquina receptora, sin necesidad de que estas establezcan una conexión previamente. Entonces, muchos de nosotros nos podríamos preguntar ¿No es esa la labor de IP? La respuesta es no, porque un paquete IP es entregado al Host correspondiente, mas no al proceso. Para poder entregar el datagrama al proceso correspondiente se deben adicionar números de puerto a los paquetes UDP. Nótese que el proceso descrito anteriormente es la demultiplexación de la que hablábamos.

UDP es normalmente utilizado en procesos en los que la pérdida de algún paquete no causa errores en la aplicación, tal como es el caso de la transmisión de datos multimedia, ya que en caso de perder un paquete de sonido o video probablemente el oído o el ojo humano ni siquiera lo noten.

2. Número de secuencia y ACK

La red tampoco garantiza que los paquetes enviados se recibirán en el orden correspondiente, ya que puede suceder que estos tomen rutas diferentes. Por esta razón, es de suma importancia incluir un número de secuencia el cual identifique de manera única el paquete que se está enviando, este número de secuencia se incrementa de acuerdo con la cantidad de información (bytes) que sean transmitidos por el emisor.

También puede suceder que el paquete nunca llegue. Para solucionar este inconveniente se debe agregar un mecanismo de acuse de recibo (ACK), el cual también deberá especificar el próximo número de secuencia esperado para evitar confusión en la confirmación. Teniendo en cuenta que los paquetes que incluyen los ACK también pueden perderse en tránsito, el emisor debe esperar un tiempo para recibir dicha confirmación, en caso de no recibirla, este debe reenviar el paquete. En caso de que un receptor reciba un paquete duplicado (por su número de secuencia) simplemente debe descartarlo.

Para obtener mayor claridad sobre el manejo de los números de secuencia y número de ACK revisemos los siguientes escenarios: 2.1 Transmisión normal (sin pérdidas ni retransmisiones)

2.1. Transmisión normal (sin pérdidas ni retransmisiones)

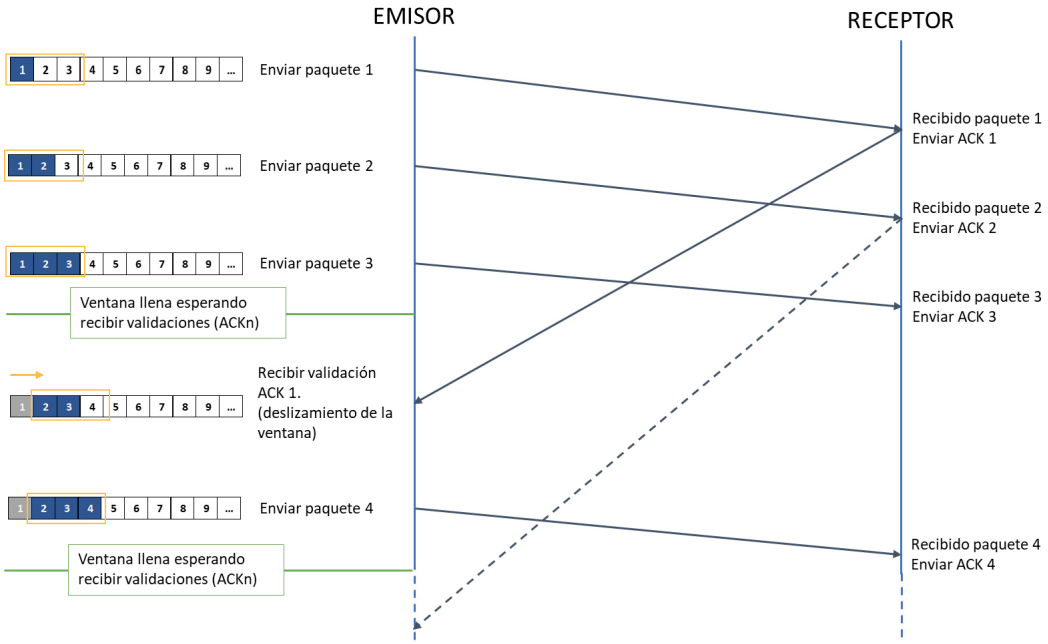

Figura 1. Transmisión normal (sin pérdidas ni retransmisiones)

Fuente: elaboración propia

Sobre la figura anterior vale la pena hacer las siguientes observaciones:

- El número de ACK corresponde al próximo byte que se espera recibir, mas no al último recibido.
- Se observa que los números de secuencia de cada uno de los host son incrementados en 1 (en este caso particularmente, debido a que se envía un único byte en cada mensaje). Sin embargo, en caso de enviar más de un byte el número de secuencia del emisor de sebe incrementar respectivamente.
- El número de secuencia del emisor y del receptor son diferentes, salvo una improbable colisión en la generación de los números aleatorios iniciales (sobre lo cual se hablará más adelante).

Ahora consideremos un escenario no tan ideal.

2.2. Escenario con pérdida de un paquete de acuse de recibo (ACK):

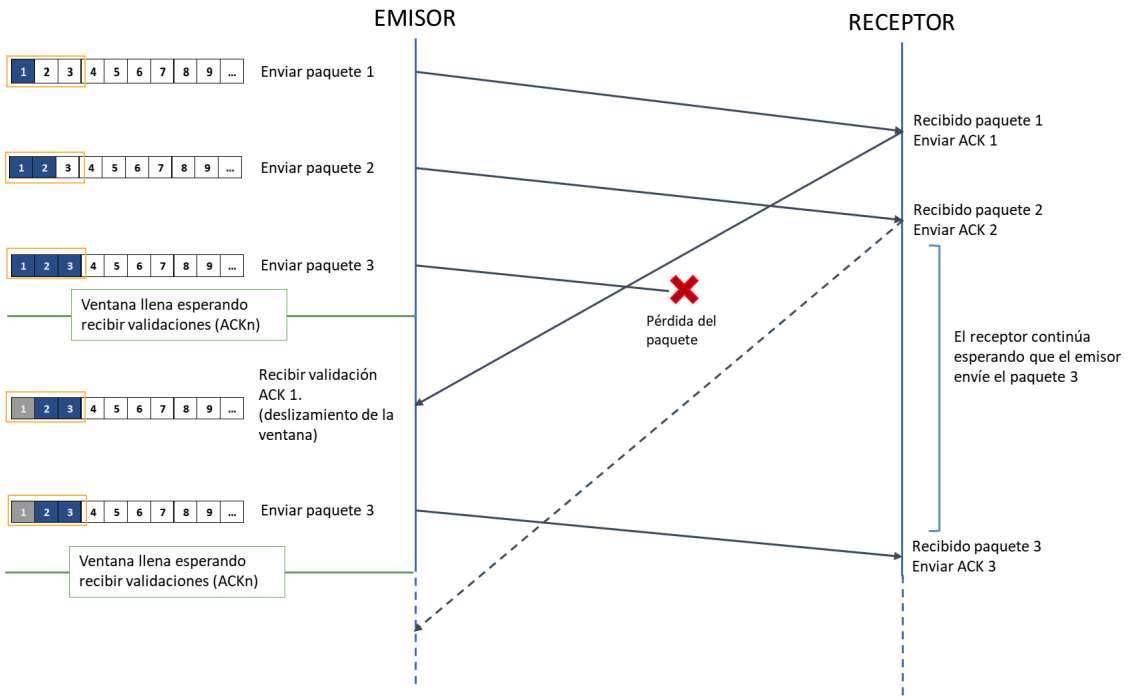

Figura 2. Escenario con pérdida de un paquete de acuse de recibo (ACK)

Fuente: elaboración propia

Este escenario contempla la pérdida de un paquete de confirmación. Considerando esto, vale la pena resaltar lo siguiente:

- El número de secuencia del emisor se incrementa en 8 debido a que envía 8 bytes de datos.
- El número de confirmación (ACK) enviado por el Host B corresponde al próximo byte que espera recibir (92+8 = 100).
- El emisor espera un tiempo para recibir la confirmación del paquete con número de secuencia 92. Una vez se cumple este tiempo, lo retransmite.
- El recepetor recibe un paquete duplicado, ante lo cual reenvía la confirmación ya que puede concluir que está demorada o se ha perdido en la red.

2.3. Apertura de la conexión TCP (HandShake)

La creación de los números de secuencia se hace al momento de establecer una conexión TCP, dicha conexión se establece usando paquetes con los flags SYN y ACK, los cuales muestran solicitud de sincronización y acuse de recibo de un paquete respectivamente. A continuación se expone el establecimiento de una conexión como ejemplo:

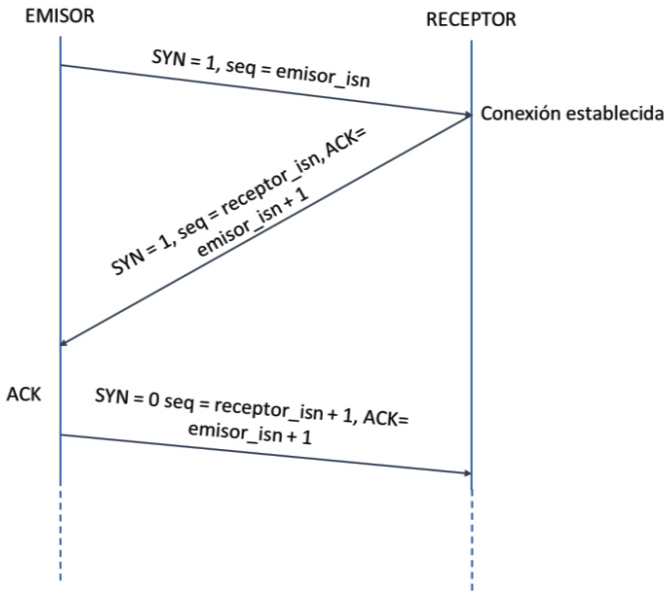

Figura 3. Apertura de la conexión TCP (HandShake)

Fuente: elaboración propia.

En la figura anterior SYN=1 indica que el bit correspondiente al flag de SYN se encuentra prendido. Analicemos los tres pasos de negociación de la conexión:

- El emisor envía un paquete con SYN = 1, para indicarle al receptor que se está solicitando el establecimiento de una conexión y con un número de secuencia asignado por él de manera aleatoria (client_isn).
- El receptor responde con el bit de SYN prendido, inicializando su número de secuencia con un número aleatorio (server_isn) y enviando el bit de ACK prendido con número de ACK correspondiente al próximo número de secuencia que se esperaría recibir por parte del cliente (client_isn+1).
- El emisor responde con un paquete con el bit de ACK prendido. Incrementa su número de secuencia en 1 y envía el paquete con este número de secuencia (server_isn+1) y con número de ACK correspondiente al siguiente byte que espera, es decir, server_isn+1.

Es importante aclarar que cada extremo escoge el número de secuencia inicial y estos deben ser generados de manera aleatoria para evitar que un atacante que se encuentre en la misma red no pueda adivinar los números de secuencia e interceptar o modificar la conversación.

2.4. Ventana (TCP)

La eficiencia de un protocolo que envía paquete por paquete esperando confirmación de cada uno es muy baja. Por esta razón, se introduce el pipeline en TCP. Este concepto consiste en enviar varios paquetes, permitiendo que haya más de un paquete pendiente por recibir confirmación, este conjunto de paquetes consecutivos que se encuentran en espera se llama ventana. Cada nodo en TCP tiene dos ventanas:

- **Ventana de envío:** Contiene el rango de bytes que se han enviado para los cuales se está esperando confirmación.
- **Ventana de recepción:** Muestra al emisor el rango de bytes que está dispuesto a procesar el nodo actuando como receptor; si esta ventana tiene una longitud de cero la comunicación full - dúplex (en ambos sentidos) se vuelve unidireccional.

3. Métodos de retransmisión

3.1. Selective Repeat (Repetición selectiva)

Las principales características de este método son las siguientes:

- El emisor puede llegar a tener hasta N paquetes sin recibir confirmación.
- El receptor debe confirmar cada uno de los paquetes que recibe
- El emisor debe verificar el tiempo límite "timeout" de cada paquete enviado, en caso de que este tiempo expire se reenvía solamente el paquete correspondiente.

3.2. Go back - N

Las principales características de este método son las siguientes:

- El emisor puede tener hasta N paquetes sin confirmar.
- El emisor mantiene el tiempo de espera límite o "timeout" para el paquete más viejo que no se ha confirmado.
- El receptor envía ACK (confirmaciones) de manera acumulativa, es decir, solo envía una confirmación cuando antes de este paquete no hay ningún espacio vacío.
- En caso de que el tiempo límite del paquete más viejo se cumpla, se deben reenviar todos los paquetes a partir de este punto ya que el protocolo no permite verificar cuál fue el paquete que hizo falta.

Los protocolos de transporte NO se ejecutan en los dispositivos de red, por el contrario, deben ser implementados en los extremos haciendo posible que la complejidad de la red se maneje ahí, el objetivo de esto es alivianar la carga de los dispositivos de red y así aumentar su eficiencia.

Referencias

Kurose, Jim, K. R. *Computer Networking: A Top Down Approach Featuring the Internet* 3rd edition. Addison - Wesley, 2004.

Santos, G. M. (2014). *Sistemas telemáticos*. Madrid, ES: RA-MA Editorial.

INFORMACIÓN TÉCNICA

FACULTAD DE
INGENIERÍA, DISEÑO
E INNOVACIÓN

Módulo: Telecomunicaciones

Unidad 3: Funcionamiento de la capa de transporte

Escenario 5: La capa de Transporte

Autor: John Alirio Olarte Ramos

Asesor Pedagógico: Juan Felipe Marciales

Diseñador Gráfico: Karim Gaitán

Asistente: María Avilán