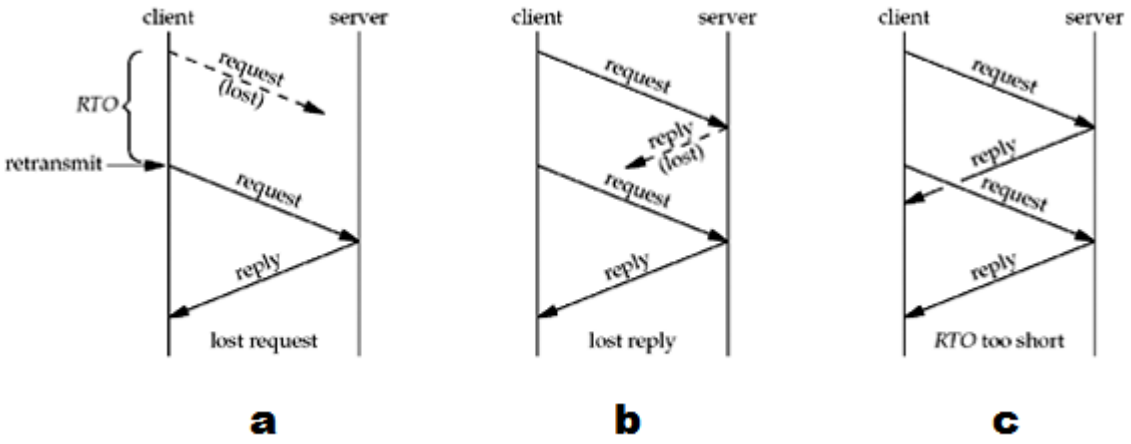
DESARROLLO DE SISTEMAS DISTRIBUIDOS

11 Ambigüedad de retransmisión

Elaborado por: Ukranio Coronilla

El algoritmo de Jacobson nos permite calcular el timeout RTO en cada solicitud del cliente. Aparentemente se tiene una solución óptima en los tiempos de retransmisión cuando existe pérdida de paquetes, sin embargo, puede ocurrir un problema conocido como **ambigüedad de retransmisión** y que se muestra a continuación.

En la figura **a** se puede observar que el mensaje de solicitud se pierde, pasa el RTO y el cliente vuelve a retransmitir. En la figura **b** el mensaje de respuesta proveniente del servidor se pierde, pasa el RTO y el cliente vuelve a retransmitir. Y en la figura **c** pasa el RTO y se retransmite la solicitud, sin embargo, ningún mensaje se ha perdido (ni solicitud ni respuesta) y por tanto se tiene una respuesta replicada provocando que se reciban datos duplicados.



Ejercicio 1:

Para verificar que este problema existe, intente transmitir un archivo grande con el cliente y servidor en la misma computadora. Agregue el código necesario para demostrar que efectivamente ocurre el problema de ambigüedad de retransmisión y no es ningún otro problema.

Una solución elegante fue propuesta por Jacobson, Braden, y Borman en 1992, y consiste en añadir en el mensaje de solicitud un **timestamp**.

Según Wikipedia timestamp es una secuencia de caracteres que permite identificar cuando ha ocurrido un evento, usualmente es una fecha y hora del día, y algunas veces es exacta hasta en una fracción de segundo.

El algoritmo consta de los siguientes pasos:

1.- Incluir en la solicitud del cliente un `timestamp` igual a la suma de la hora del cliente más el RTO estimado (se recomienda el tipo `struct timeval` porque incluye fecha y hora, así como precisión en el orden de microsegundos).

2.- El servidor debe copiar el `timestamp` en su mensaje de respuesta.

3.- Cuando llegue un mensaje de respuesta, el cliente deberá tomar la hora a la que llega el mensaje `trecepcion`. Y si

`trecepcion` <= `timestamp`

entonces se considera el mensaje como válido. En caso contrario tenemos una respuesta extemporánea, y por consiguiente se debe ignorar.

En el caso de que exista una respuesta extemporánea es importante no volver a enviar otra solicitud, sino simplemente esperar a la respuesta duplicada. De lo contrario no se resolverá el problema.

Ejercicio 2

Implementar el algoritmo para resolver el problema de ambigüedad de retransmisión. Finalmente haga las modificaciones necesarias para transmitir el archivo Linux.mp4 a la mayor velocidad posible. Anote sus resultados en el pizarrón, habrá un punto en el proyecto para el alumno que logre la mayor velocidad de transferencia entre dos computadoras.