**Llamadas a procedimientos remotos**

Los sockets los utilizamos en los sistemas cliente servidor para enviar y recibir mensajes de una maquina local a otra. En 1984 Birrell y Nelson crearon un mecanismo para permitir que los programas llamaran procedimientos en otras máquinas. El mecanismo se le llamo procedimiento remoto (RPC). La forma de hacer que las llamadas a procedimientos remotos funcionen está en la creación de funciones que le hacen parecer al usuario que la llamada es realmente local. La función se parece a la función que el usuario intenta llamar pero ésta realmente contiene código para enviar y recibir mensajes a través de una red.

El RPC tiene 2 ventajas la primera es para el programador, que ahora puede utilizar el procedimiento la semántica de llamadas duras y la otra es que la escritura de aplicaciones distribuidas se simplifican porque el RPC oculta todo el código de red en funciones de código auxiliar. Dicho esto entonces, los programas de aplicación no tiene que preocuparse por los detalles como los sockets, etc. Ahora usando el modelo de referencia OSI, RPC es un servicio de capa de presentación, surgen varios problemas que citare a continuación:

Enviar los parámetros: Pasar por valor es sencillo, solo copie el valor en el mensaje de red). Pasar por referencia es difícil; no tiene sentido pasar una dirección a una máquina remota.

Como se representan los datos: Con RPC, una máquina remota puede tener un orden de bytes diferente, diferentes tamaños de números enteros y una representación de punto flotante diferente. El problema se resolvió en la suite de protocolos IP obligando a todos a usar big endian1ordenamiento de bytes para todos los campos de 16 y 32 bits en los encabezados.

A que nos unimos: Necesitamos ubicar un host remoto y el proceso adecuado (puerto o transporte dirección) en ese host.

¿Qué pasa cuando las cosas salen mal?: los programas que utilizan el RPC deben estar preparadas para probar el fallo de una llamada de procedimiento remoto o atrapar una excepción.

Para habilitar el uso de llamadas a procedimientos remotos con los lenguajes como C o java que no manejan el concepto de RPC, se utiliza una solución comúnmente adoptada es proporcionar un compilador separado que genere las funciones de stub de cliente y servidor y la información se escribe en un lenguaje de definición de interfaz.

**Primera generación de llamadas a procedimiento remoto**

Sun Microsystem fue de los primeros sistemas en proporcionar bibliotecas RPC y un compilador, lo desarrollaron a principios de la década de 1980 como parte de su archivo de computación en red abierta (ONC).

Sun tenía un compilador que tomaba la definición de una interfaz de procedimiento remoto y generaba las funciones de stub de cliente y servidor. Pero antes de ejecutar este compilador, el programador debía tener la definición de la interfaz.

**Segunda generación de llamadas a procedimiento remoto**

DCOM (Distributed COM) se introdujo con Windows NT 4.0 en 1996 y es una extensión del modelo de objetos componentes para permitir que los objetos se comuniquen entre máquinas. La función de DCOM es permitir el acceso a objetos COM remotos, un proceso que necesita crear un objeto necesitaría suministrar la red el nombre del servidor, así como el ID de clase. DCOM proporciona una capacidad de denominación de objetos llamada apodo. Cada postura de un objeto puede crear su propio apodo y devolvérselo al cliente. Se admiten varios tipos de apodos como el apodo de archivo, que utiliza el tipo de archivo (p. Ej., ".Doc") para determinar el objeto apropiado. El apodo de URL abstrae el acceso a las URL a través de protocolos de Internet en una interfaz COM y el nombre de clase: se usa junto con otros nombres para anular el mecanismo de búsqueda de ID de clase.