Communication

Content

2.1 Layered Protocols, Interprocess communication (IPC): MPI, Remote Procedure Call (RPC), Remote Object Invocation, Remote Method Invocation (RMI)

2.2 Message Oriented Communication, Stream Oriented Communication, Group Communication

RPC

- RPC is a protocol that one program can use to request a service from a program located in another computer on a network without having to understand the network details.
- A procedure call is also called as function call or subroutine call.
- RPC uses the client server model.

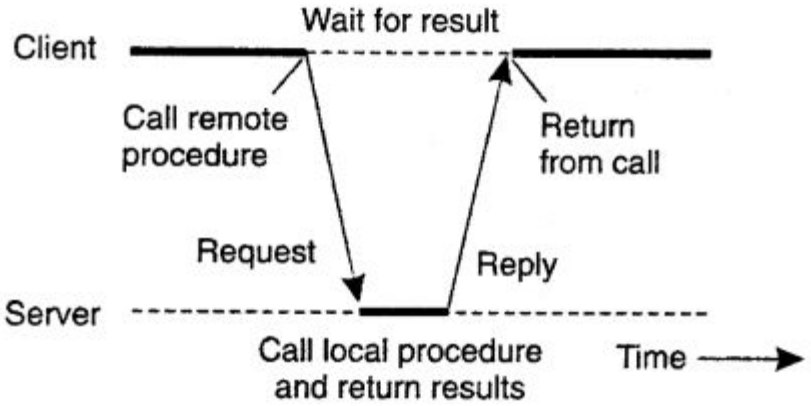


Figure 4-6. Principle of RPC between a client and server program.

RPC

It includes five elements:

1. The client
2. The client stub
3. The RPC runtime
4. The server stub
5. The server

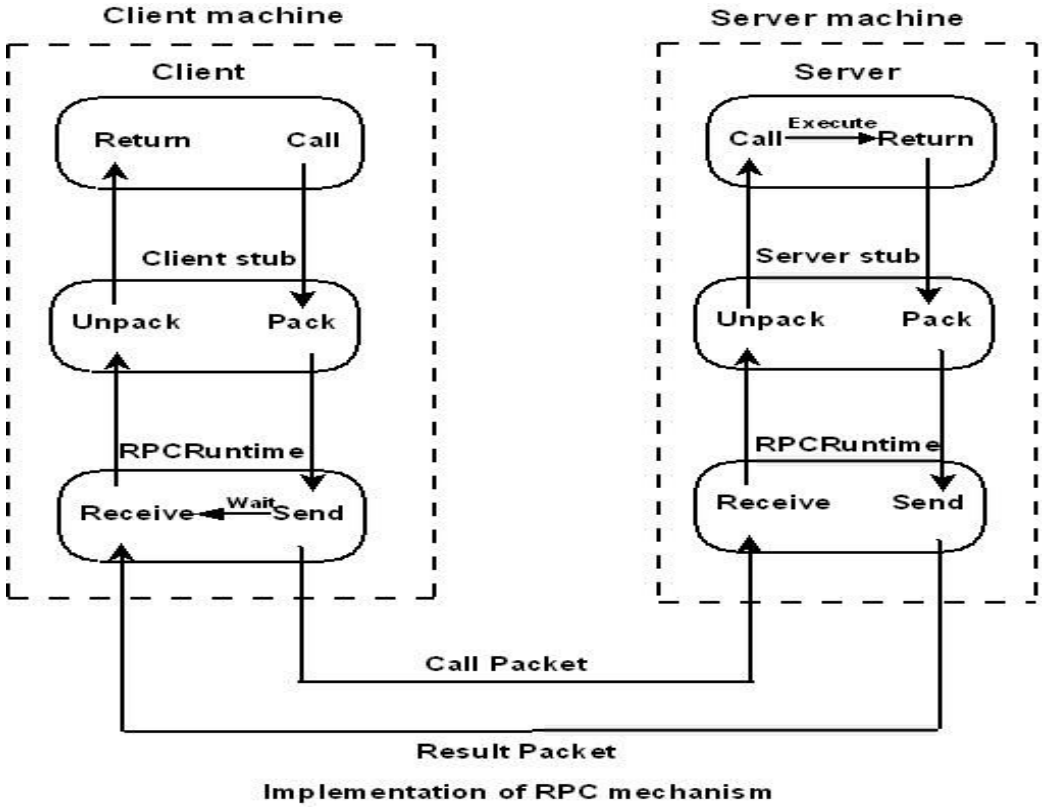
1. Client

- A Client is a user process which initiates a RPC
- The client makes a normal call that will invoke a corresponding procedure in the client stub.

RPC

2. Client Stub

- Client stub is responsible for the following two tasks:
- On receipt of a call request from the client, it packs specifications of the target procedure and arguments into a message and asks the local RPC Runtime to send it to the server stub.
- On receipt of the result of procedure execution, it unpacks the result and passes it to the client.

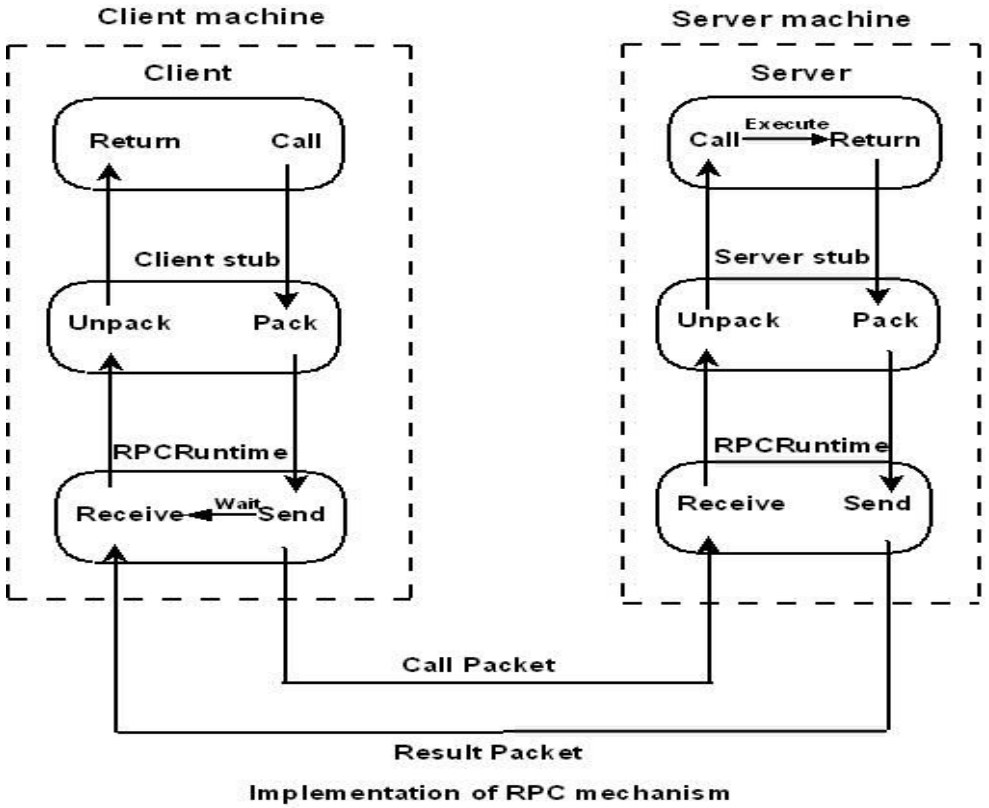


Implementation of RPC mechanism

RPC

3. RPC Runtime

- Transmission of messages between Client and the server machine across the network is handled by RPC Runtime.
- It performs Retransmission, Acknowledgement, Routing and Encryption.
- RPC Runtime on Client machine receives messages containing result of procedure execution from server and sends it client stub as well as the RPC Runtime on server machine receives the same message from server stub and passes it to client machine.

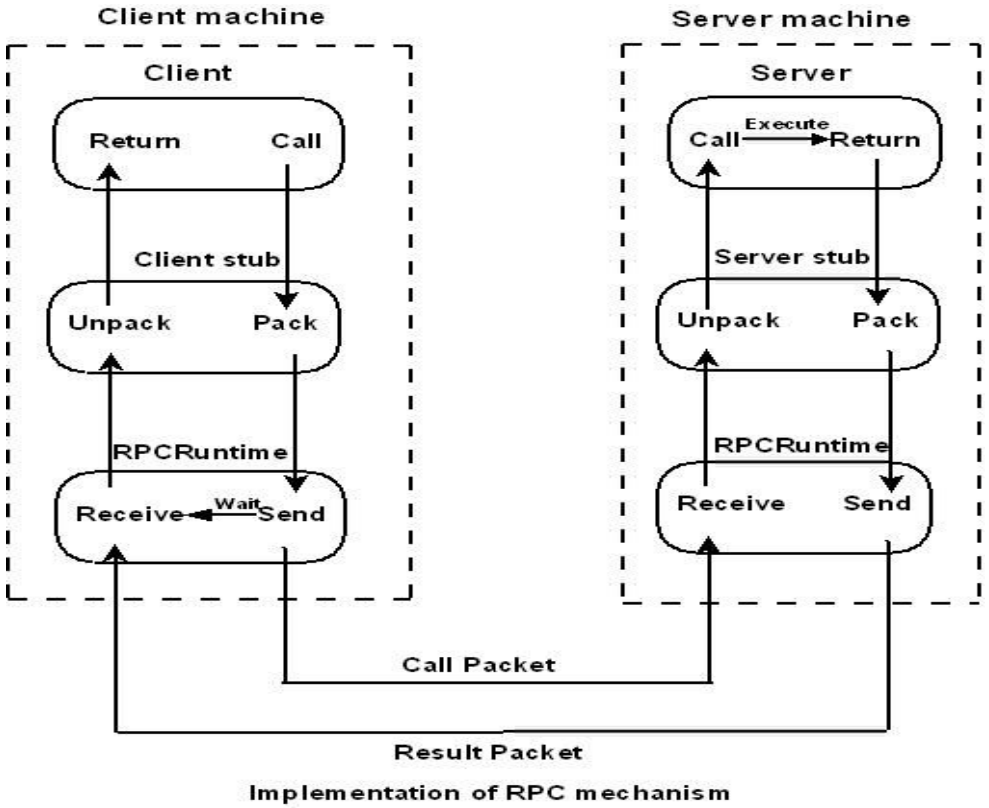


Implementation of RPC mechanism

RPC

4. Server Stub

- Server stub is similar to client stub and is responsible for the following two tasks:
- On receipt of a call request message from the local RPC Runtime, it unpacks and makes a normal call to invoke the required procedure in the server.
- On receipt of the result of procedure execution from the server, it packs the result into a message and then asks the local RPC Runtime to send it to the client stub.

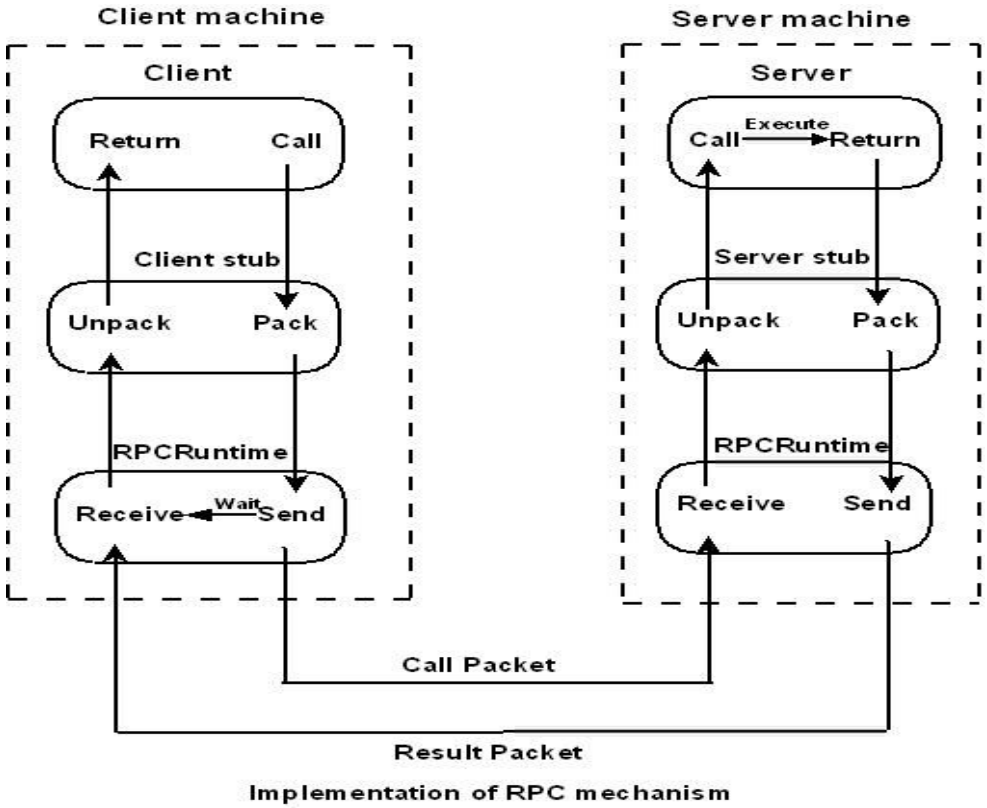


Implementation of RPC mechanism

RPC

5. Server

- When a call request is received from the server stub, the server executes the required procedure and returns the result to the server stub.



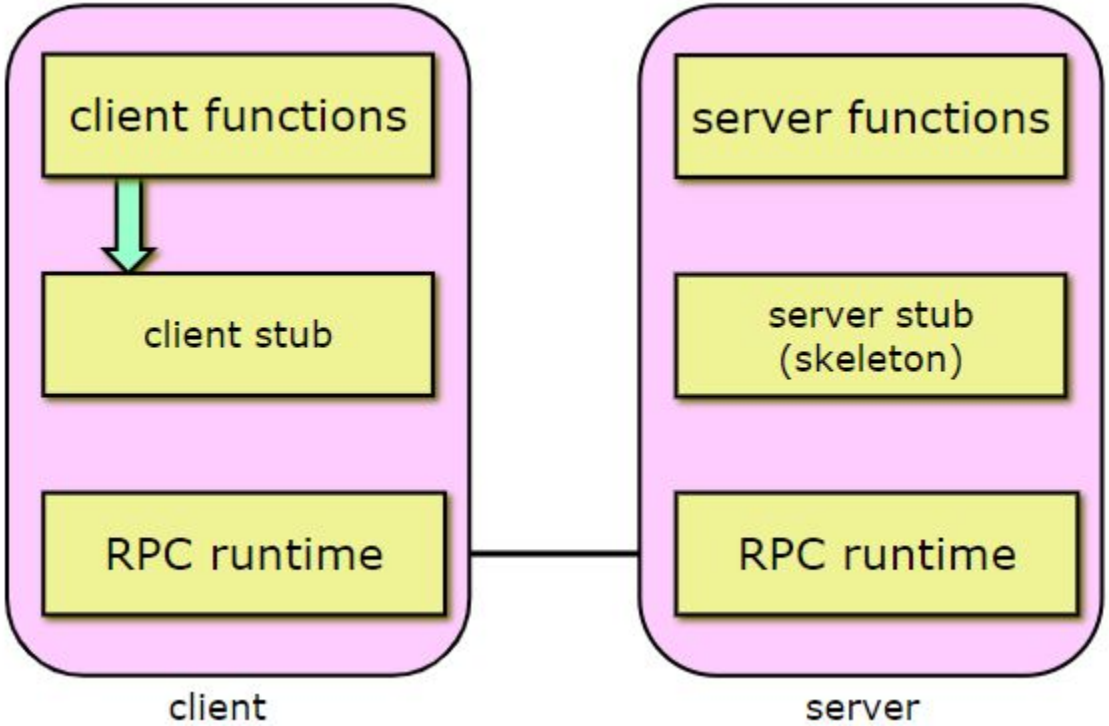
Implementation of RPC mechanism

Steps of RPC

1. Client procedure calls client stub in normal way.
2. Client stub builds message, calls local OS.
3. Client's OS sends message to remote OS
4. Remote OS gives message to server stub
5. Server stub unpacks parameters, calls server
6. Server does work, returns result to the stub
7. Server stub packs it in message, calls local OS
8. Server's OS sends message to client's OS
9. Client's OS gives message to client stub
10. Stub unpacks result, returns to client

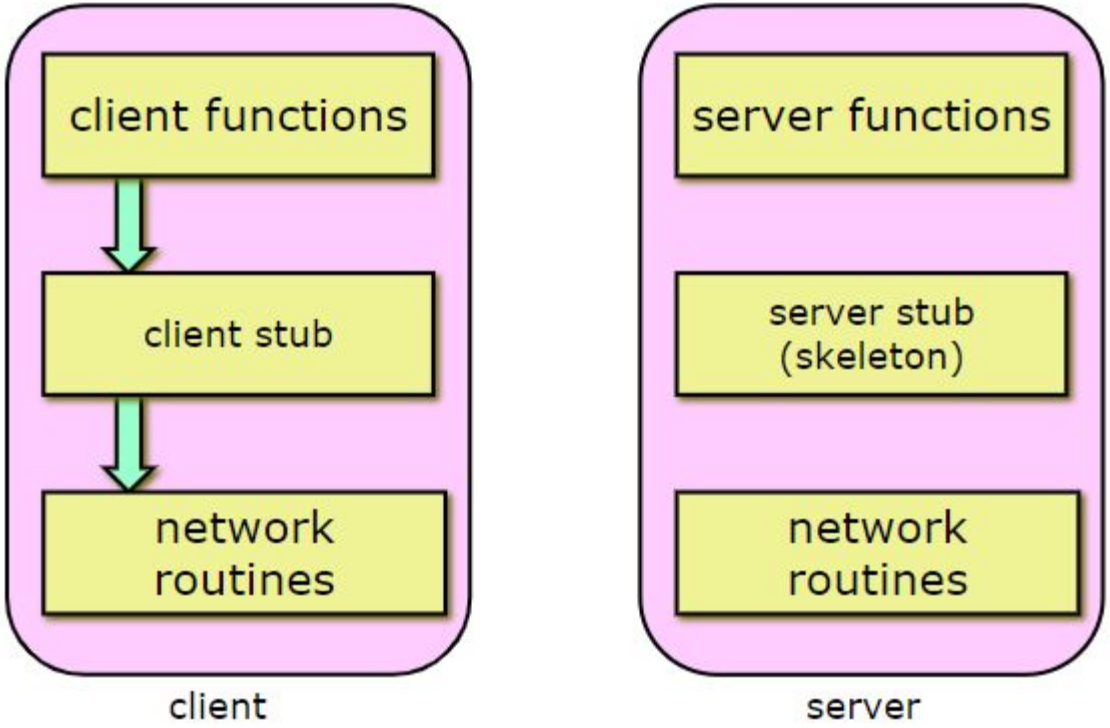
Steps of RPC

1. Client procedure calls client stub in normal way.



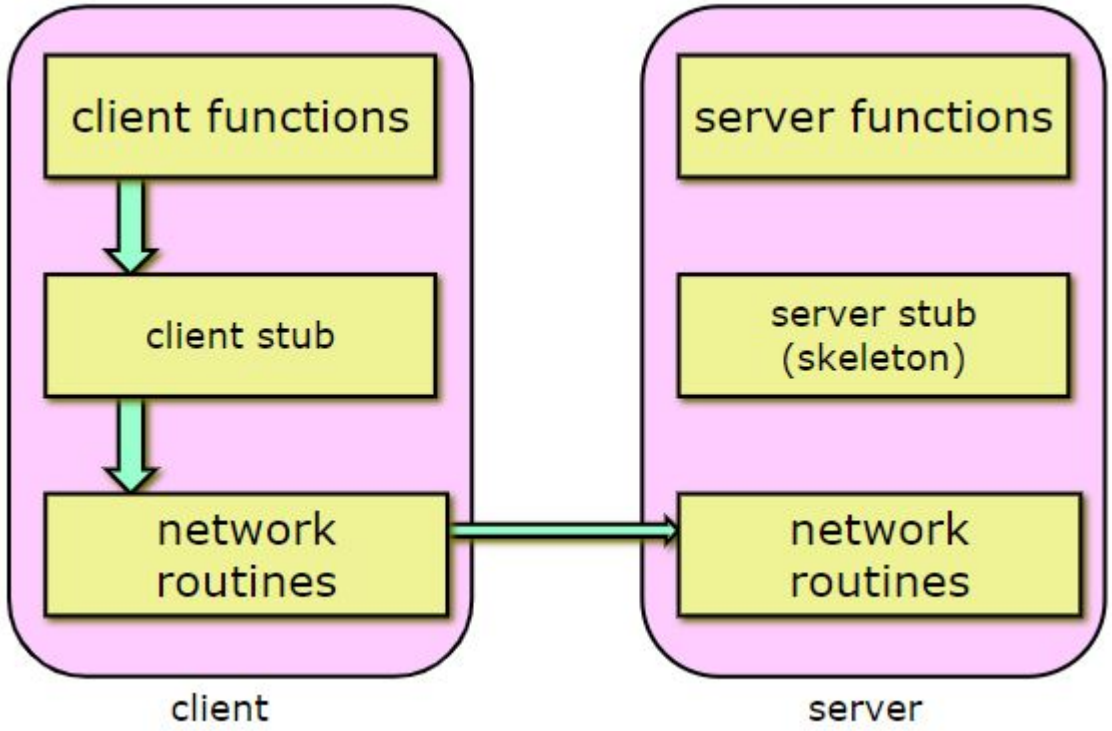
Steps of RPC

2. Client stub builds message, calls local OS.



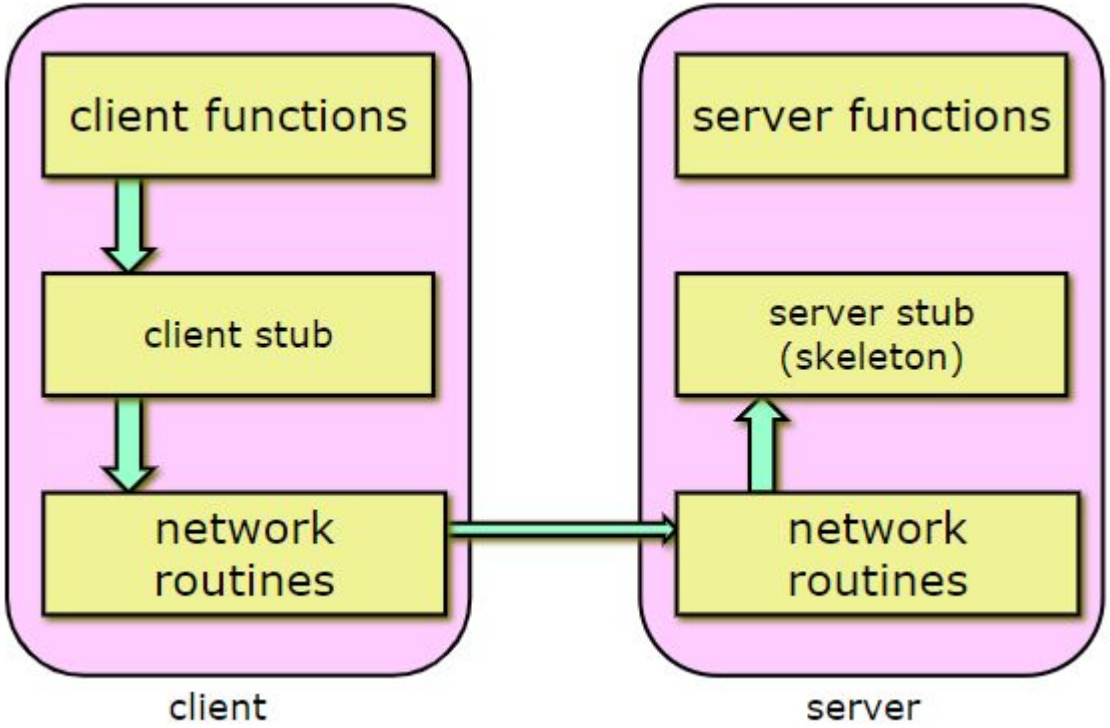
Steps of RPC

3.Client's OS sends message to remote OS



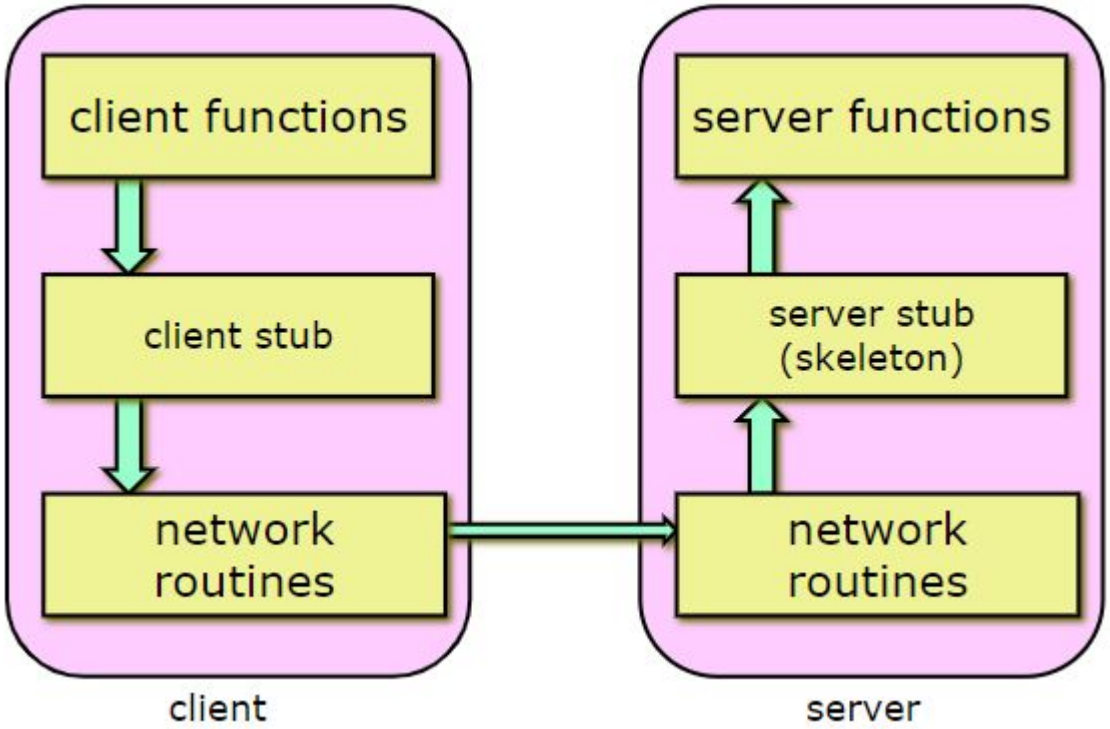
Steps of RPC

4. Remote OS gives message to server stub



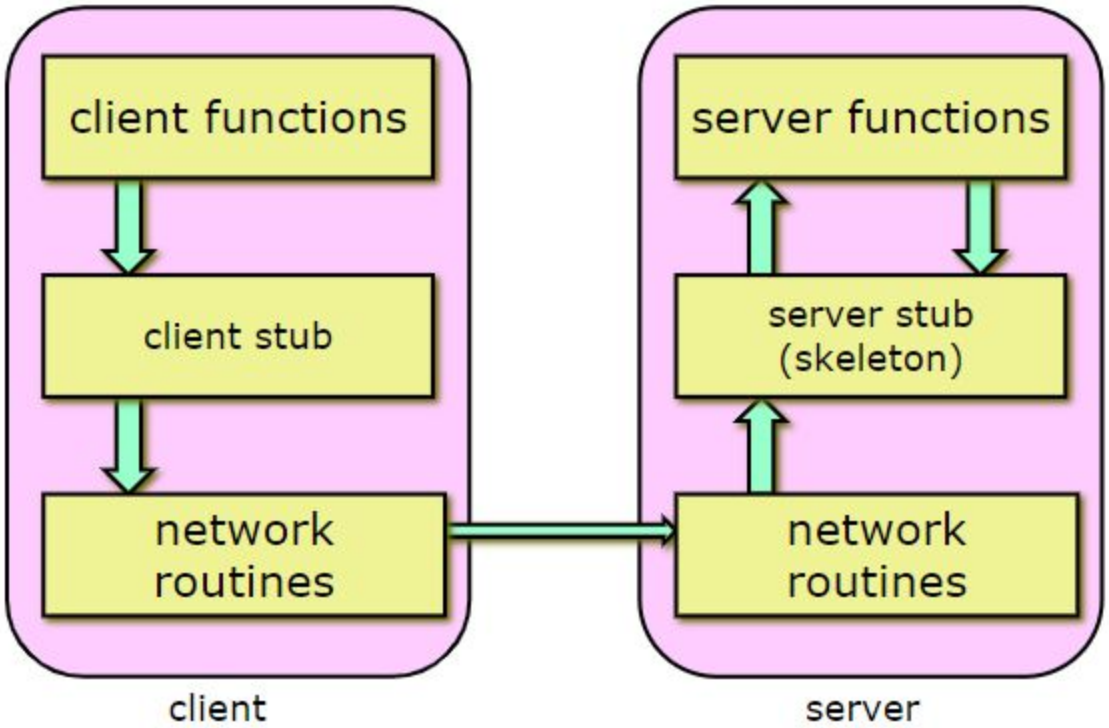
Steps of RPC

5. Server stub unpacks parameters, calls server



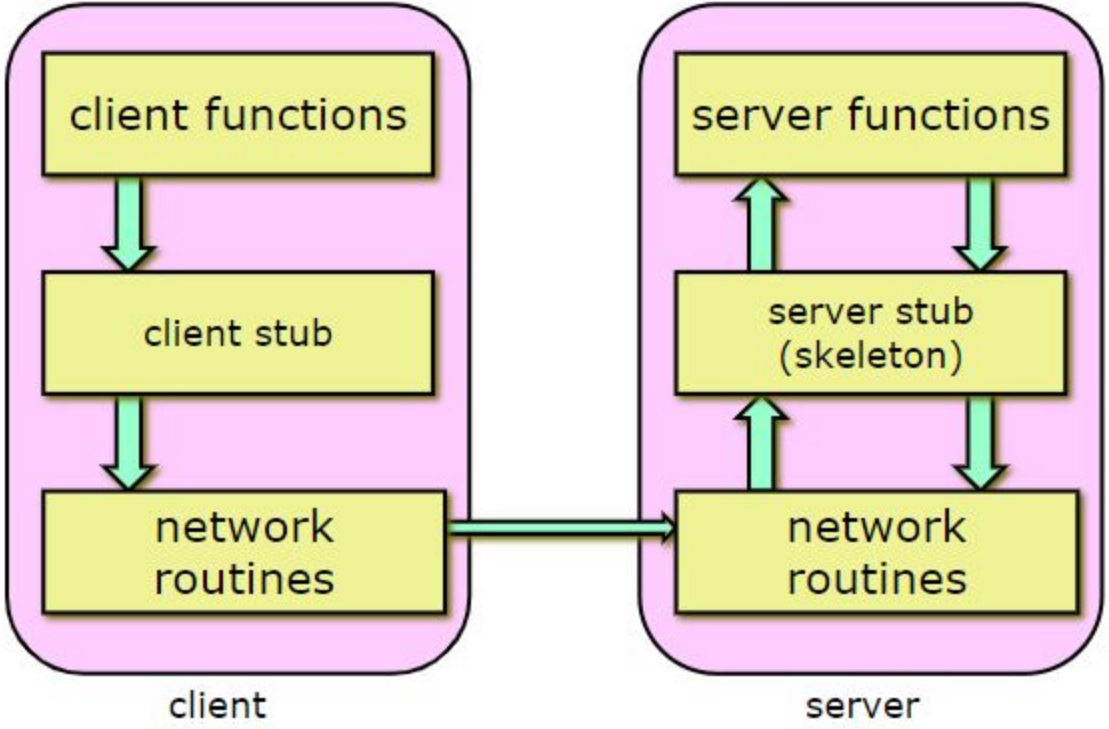
Steps of RPC

6. Server does work, returns result to the stub



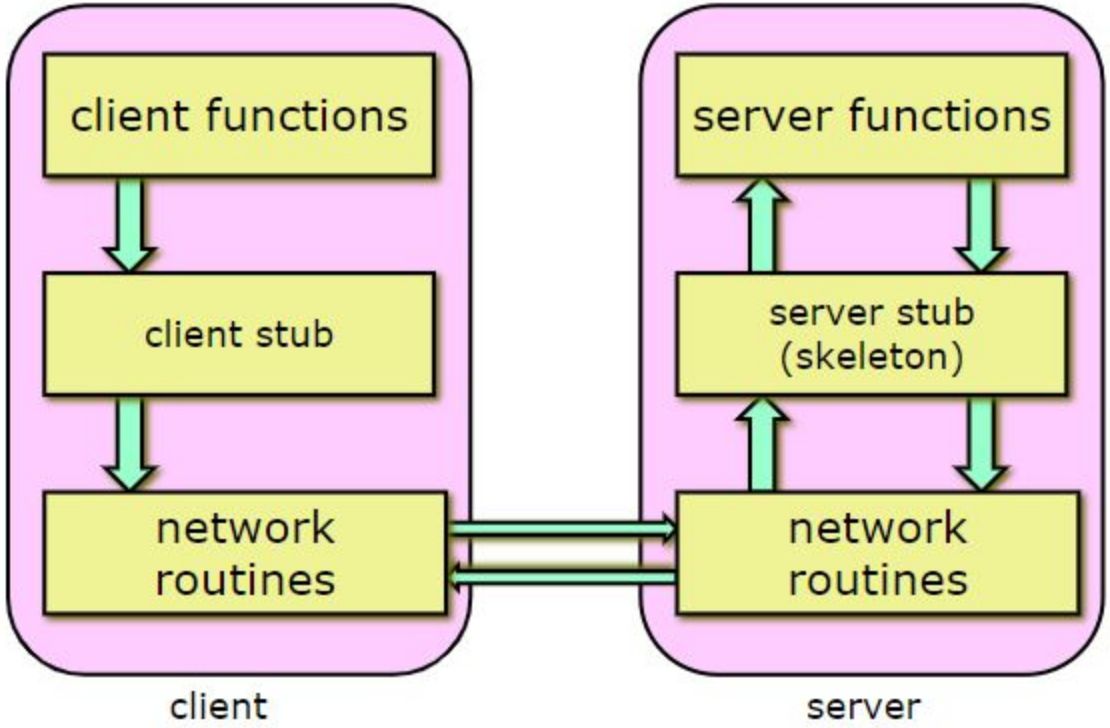
Steps of RPC

7. Server stub packs it in message, calls local OS



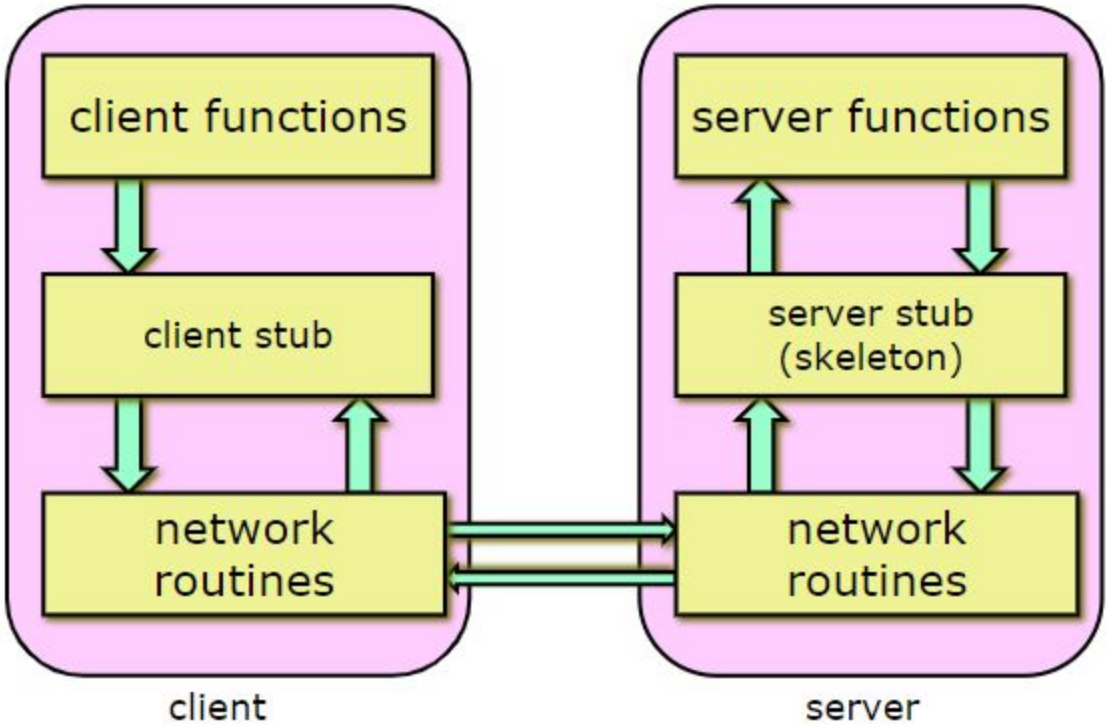
Steps of RPC

8. Server's OS sends message to client's OS



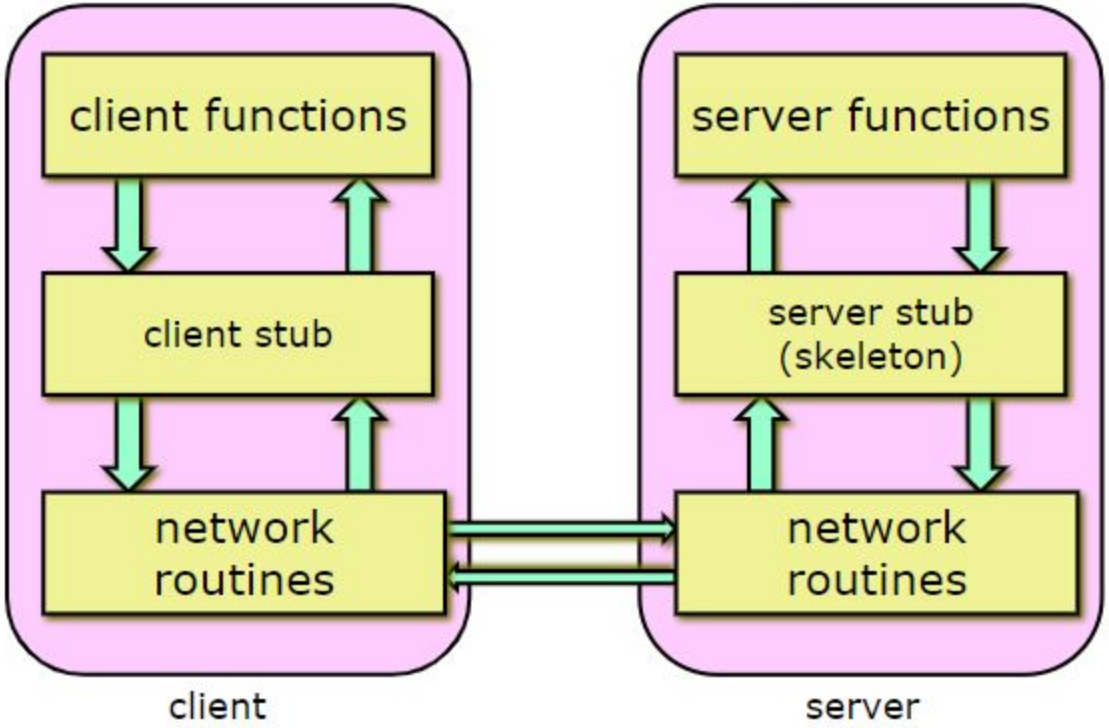
Steps of RPC

9. Client's OS gives message to client stub



Steps of RPC

10. Stub unpacks result, returns to client



Example of RPC

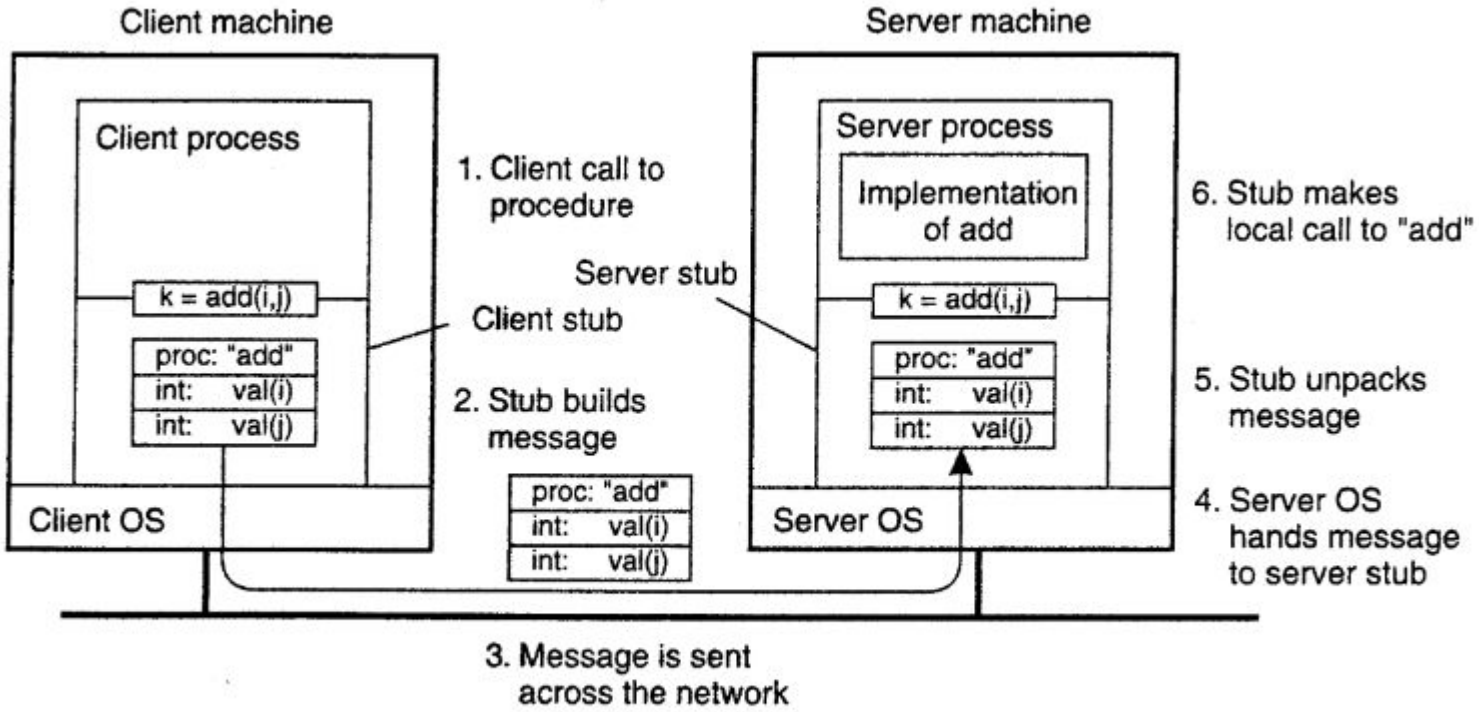


Figure 4-7. The steps involved in a doing a remote computation through RPC.

Parameter Passing

In RPC, the parameter are passed in two ways - Pass by value and Pass by reference.

- In pass by value, the actual parameters and their data types are copied in to the stack and passed to the called procedure.
- In pass by reference, the pointer to the data is passed instead of value to the called procedure at the server side.

It is very difficult to implement as the server needs to keep the track of the pointer to the data at the client’s address space.

Asynchronous RPC

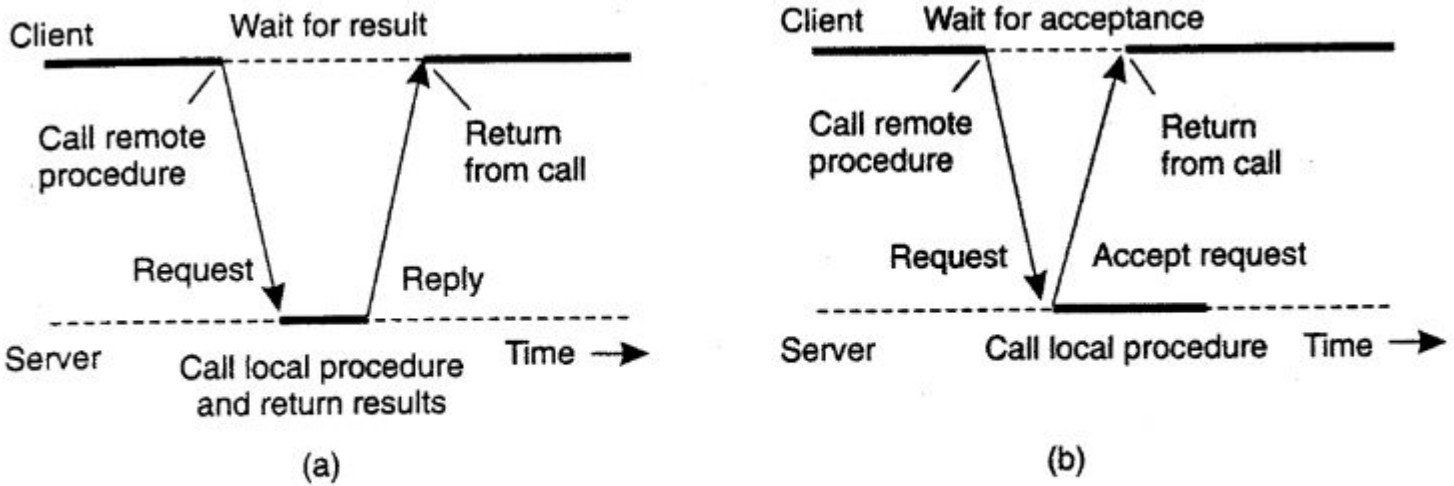


Figure 4-10. (a) The interaction between client and server in a traditional RPC. (b) The interaction using asynchronous RPC.

Asynchronous RPC

- As in conventional procedure calls, when a client calls a remote procedure, the client will block until a reply is returned.
- This strict request-reply behavior is unnecessary when there is no result to return, and only leads to blocking the client while it could have proceeded and have done useful work just after requesting the remote procedure to be called.
- Examples of where there is often no need to wait for a reply include: transferring money from one account to another, adding entries into a database, starting remote services, batch processing, and so on.
- To support such situations, RPC systems may provide facilities for what are called asynchronous RPCs, by which a client immediately continues after issuing the RPC request.
- With asynchronous RPCs, the server immediately sends a reply back to the client the moment the RPC request is received, after which it calls the requested procedure.
- The reply acts as an acknowledgment to the client that the server is going to process the RPC.
- The client will continue without further blocking as soon as it has received the server's acknowledgment. Figure below shows how client and server interact in the case of asynchronous RPCs.

Asynchronous RPC

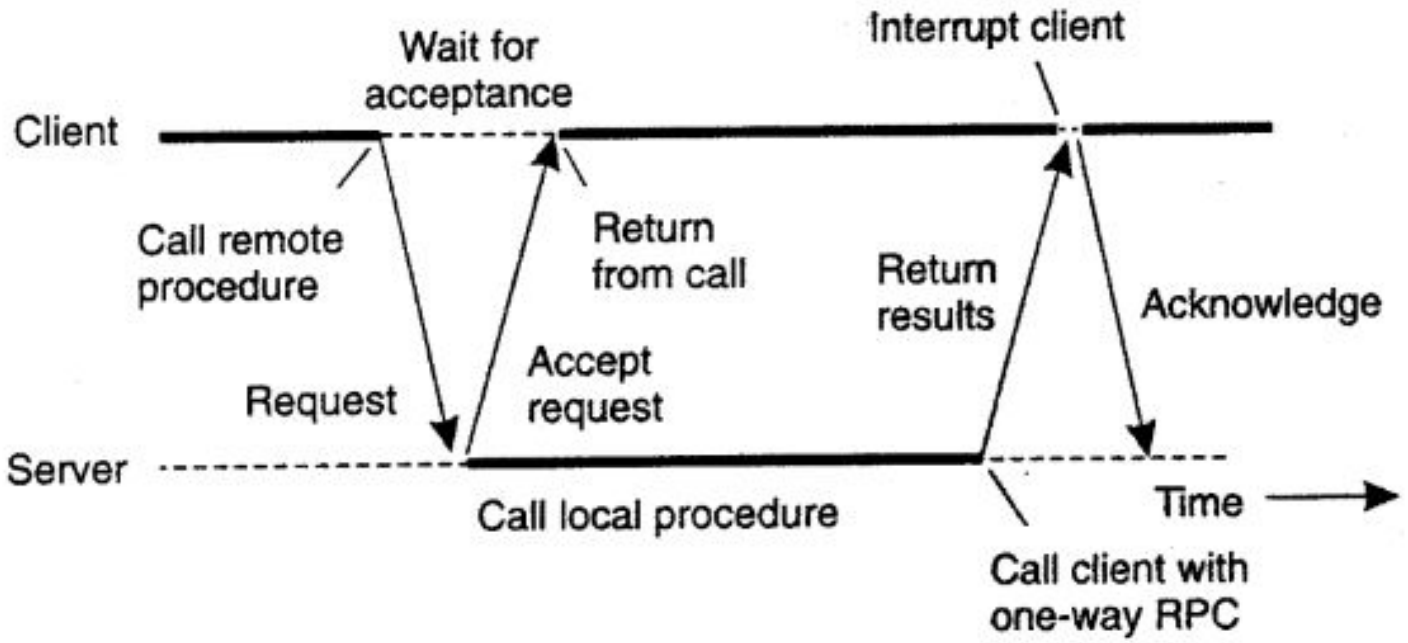


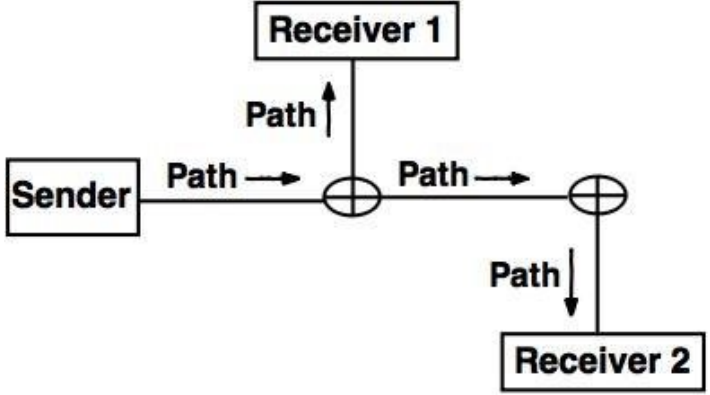
Figure 4-11. A client and server interacting through two asynchronous RPCs.

RSVP

An example of how RSVP can be used in a video conferencing application

- A user wants to initiate a video conference with multiple participants and requires a certain amount of bandwidth and other network resources to ensure smooth transmission of video and audio data.
- The user's device sends an RSVP PATH message to the network, requesting the necessary resources for the video conference.
- The network responds with an RSVP RESV message, reserving the requested resources for the duration of the video conference.
- The video conference takes place, with the reserved resources ensuring the smooth and uninterrupted transmission of data between all participants
- In this example, RSVP is used to reserve network resources (the video conference) among multiple participants. This ensures that the video conference can take place without any interruptions or degradation in quality due to a lack of network resources.

RSVP



Path messages

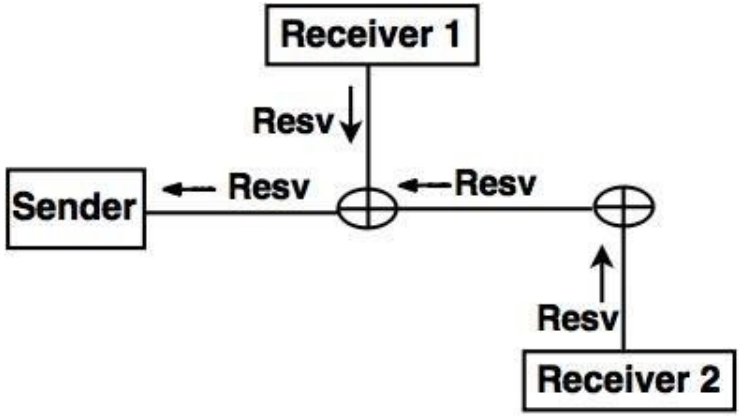
The two important types of RSVP messages are:

1. Path messages:

The receivers in a flow make the reservation in RSVP, but the receivers do not know the path traveled by the packets before the reservation. The path is required for reservation To solve this problem the RSVP uses the path messages.

A path message travels from the sender and reaches to all receivers by multi-casting and path message stores the necessary information for the receivers.

RSVP



Resv Messages

The two important types of RSVP messages are:

2. Resv messages:

After receiving path message, the receiver sends a Resv message. The Resv message travels to the sender and makes a resource reservation on the routers which supports for RSVP.