

Communication

Chapter 2

Based on the slides provided by Tanenbaum and Steen

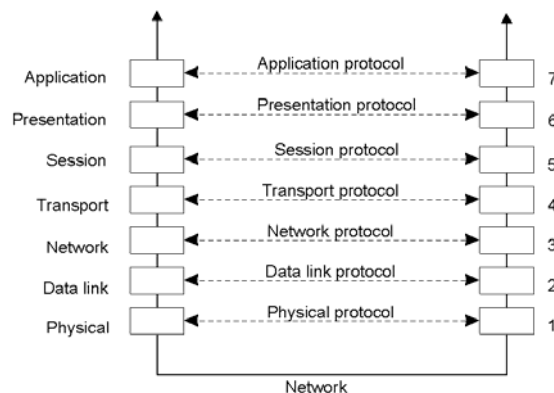
Communication

- Exchange of information between processes on different machines
- Communication through message passing is difficult, intricacies of message passing must be hidden from application developers
- Four most widely used communication models:
 - Remote Procedure Call (RPC)
 - Remote Method Invocation (RMI)
 - Message-Oriented Middleware (MOM)
 - Streams

6/9/2005

2

Layered Protocols



- Layers, interfaces, and protocols in the Open Systems Interconnection (OSI) model.
- Protocols are the rules about the format, content, and meaning of messages that communicating process must follow.

6/9/2005

3

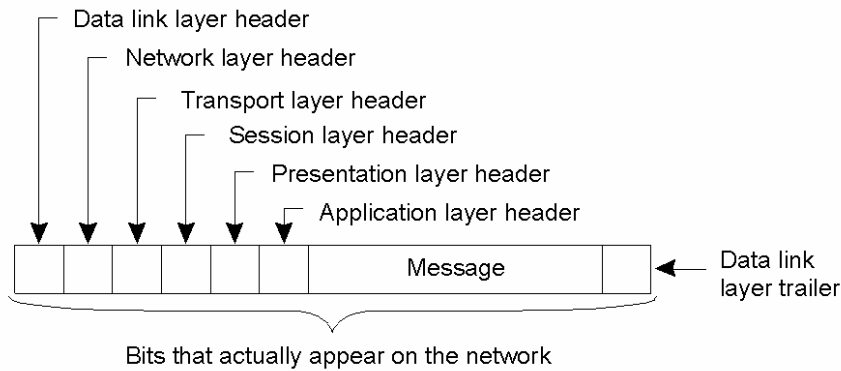
Types of Protocols

- Connection-oriented
 - Sender and receiver first establish connection and negotiate the protocol before exchanging data
 - At the end of communication the connection is terminated
 - Example: Telephone system
- Connectionless
 - Sender transmits data without any advance setup
 - Example: Postal mail system

6/9/2005

4

Sample Message Format



A typical message as it appears on the network.

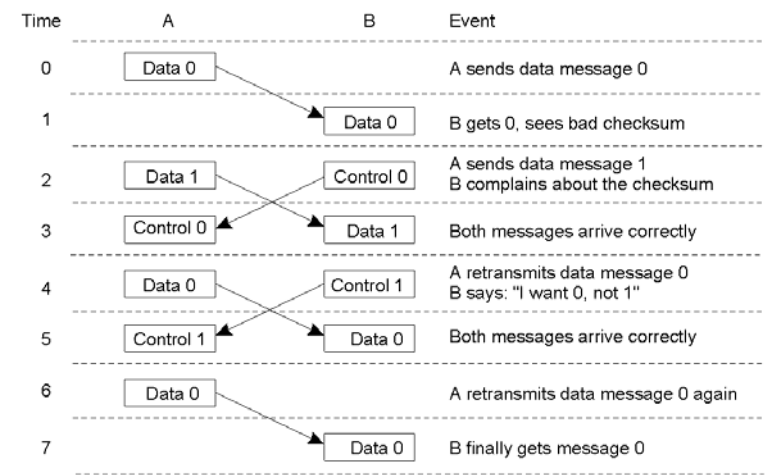
Physical Layer

- Sends 0s and 1s
- Deals with standardizing the electrical, mechanical, and signaling interfaces
- Example: RS-232-C standard for serial communication lines

Data Link Layer

- Provides mechanism to detect and correct errors
- Groups bits into units called frames and assigns a sequence number
- Adds a special bit pattern to the start and end of each frame along with a checksum
- Receiver recomputes the checksum and compares it with the checksum appended to the frame
- If they don't match receiver requests sender to retransmit

Data Link Layer



Discussion between a receiver and a sender in the data link layer.

Network Layer

- Chooses the best path for sending a message across a network – routing
- Shortest route may not always be the best route
- Most widely used network protocol is the connection-less Internet Protocol (IP)
- Messages in the network layer are called packets

6/9/2005

9

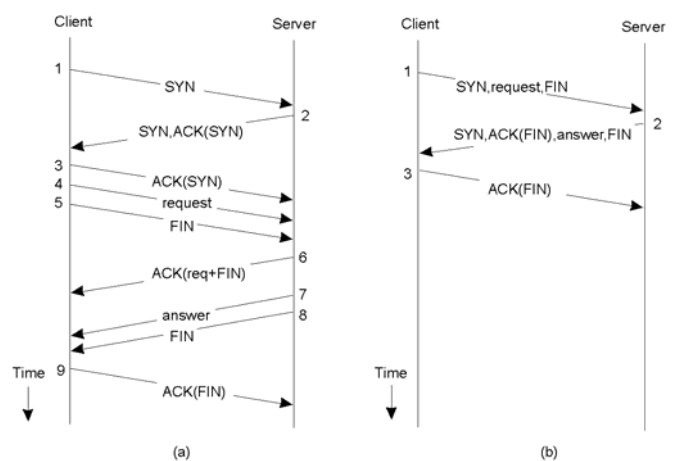
Transport Layer

- Provides an interface to the application developer to use the underlying network
- Provides end-to-end communication with reliable message deliver (does error recovery)
- Two common transport protocols:
 - Transmission Control Protocol (TCP) – connection oriented, adds overhead
 - Universal Datagram Protocol (UDP) – connection-less, works well on reliable networks

6/9/2005

10

Client-Server TCP

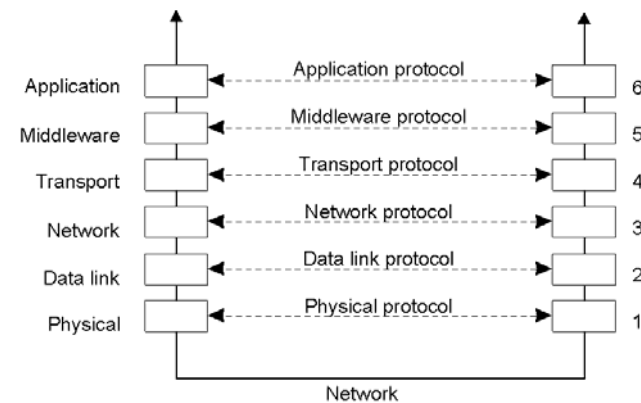


- a) Normal operation of TCP.
b) Transactional TCP.

6/9/2005

11

Middleware Protocols



An adapted reference model for networked communication.

6/9/2005

12

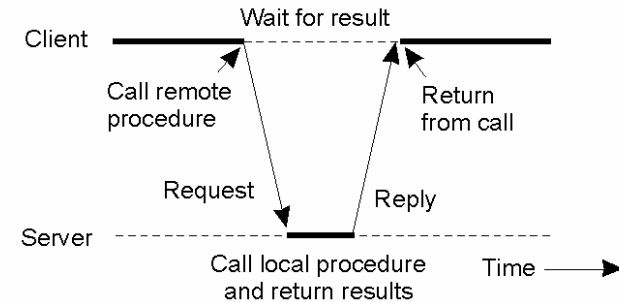
Remote Procedure Call

- Allows programs to call procedures located on other machines
- No message passing is visible to the programmer

6/9/2005

13

Client and Server Stubs



Principle of RPC between a client and server program.

6/9/2005

14

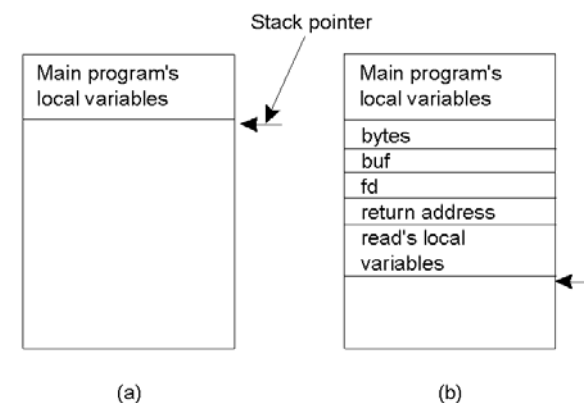
Steps of a Remote Procedure Call

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

6/9/2005

15

Conventional Procedure Call

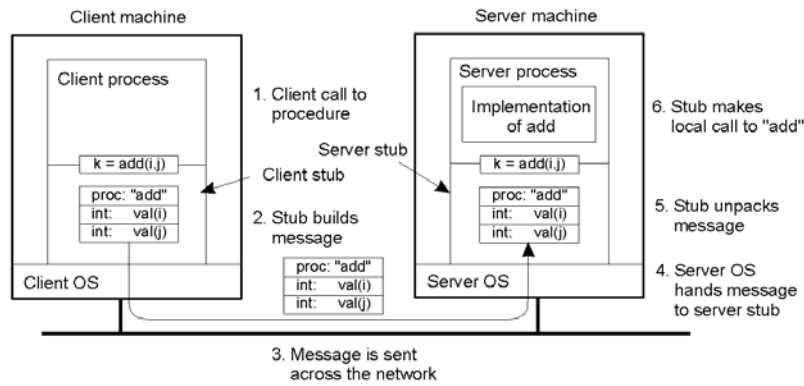


- a) Parameter passing in a local procedure call: the stack before the call to read
- b) The stack while the called procedure is active

6/9/2005

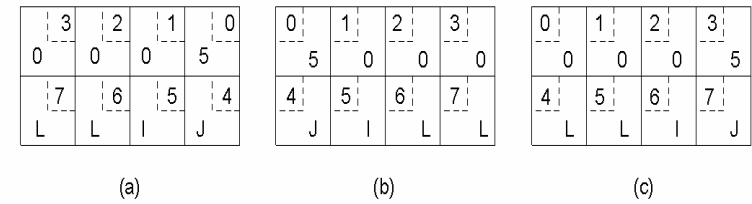
16

Passing Value Parameters (1)



Steps involved in doing remote computation through RPC

Passing Value Parameters (2)



- a) Original message on the Pentium
- b) The message after receipt on the SPARC
- c) The message after being inverted. The little numbers in boxes indicate the address of each byte

Parameter Specification and Stub Generation

- a) A procedure
- b) The corresponding message.

```
foobar( char x; float y; int z[5] )
{
  ...
}
```

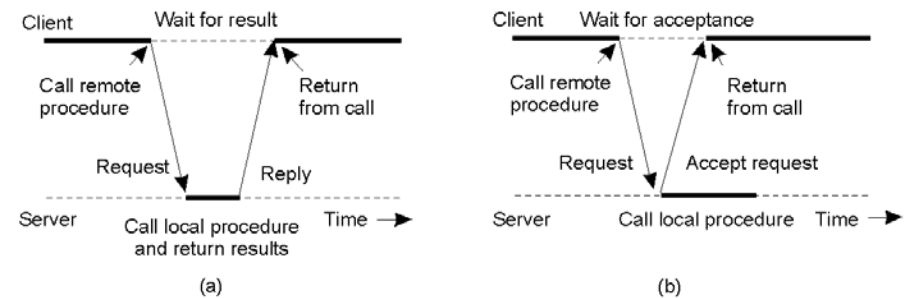
(a)

foobar's local variables	
	x
	y
	5
	z[0]
	z[1]
	z[2]
	z[3]
	z[4]

(b)

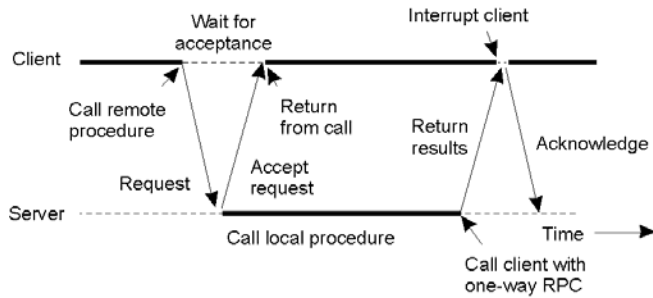
Message format definition, data representation are part of RPC protocol

Asynchronous RPC (1)



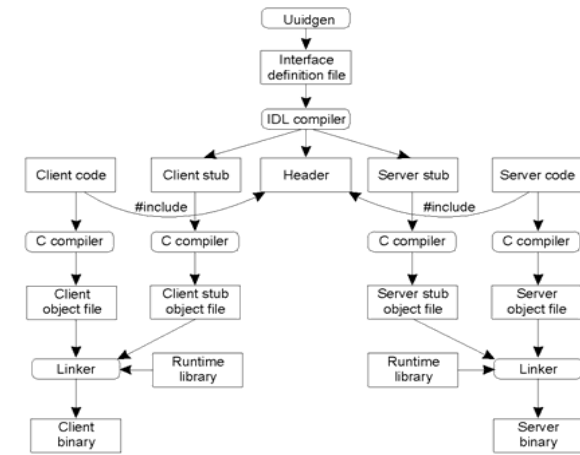
- a) The interconnection between client and server in a traditional RPC
- b) The interaction using asynchronous RPC

Asynchronous RPC (2)



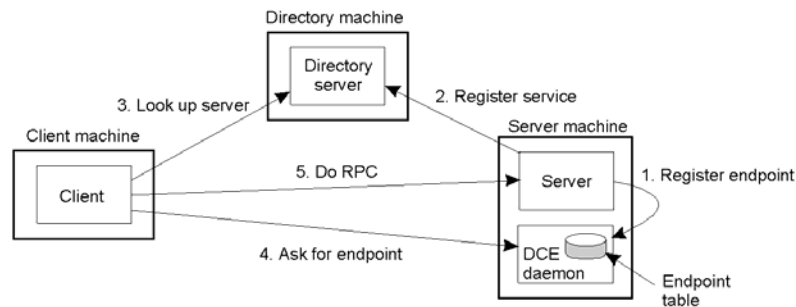
A client and server interacting through two asynchronous RPCs

Writing a Client and a Server



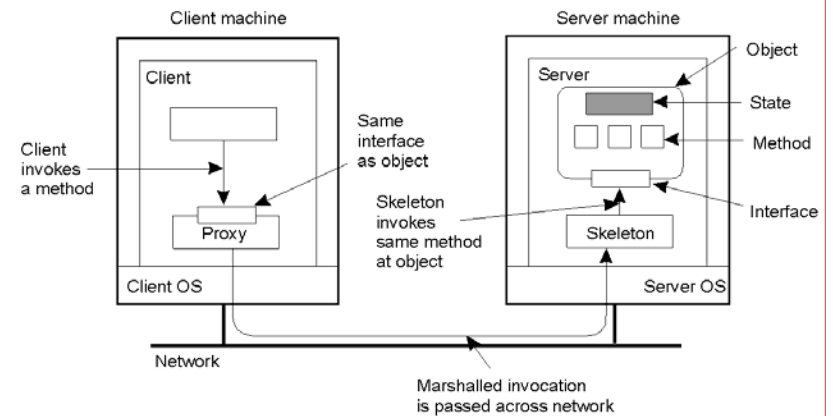
The steps in writing a client and a server in DCE RPC.

Binding a Client to a Server



Client-to-server binding in DCE.

Distributed Objects



Common organization of a remote object with client-side proxy.

Binding a Client to an Object

```
Distr_object* obj_ref;           //Declare a systemwide object reference
obj_ref = ...;                  // Initialize the reference to a distributed object
obj_ref-> do_something();        // Implicitly bind and invoke a method
```

(a)

```
Distr_object objPref;           //Declare a systemwide object reference
Local_object* obj_ptr;         //Declare a pointer to local objects
obj_ref = ...;                 //Initialize the reference to a distributed object
obj_ptr = bind(obj_ref);       //Explicitly bind and obtain a pointer to the local proxy
obj_ptr -> do_something();
```

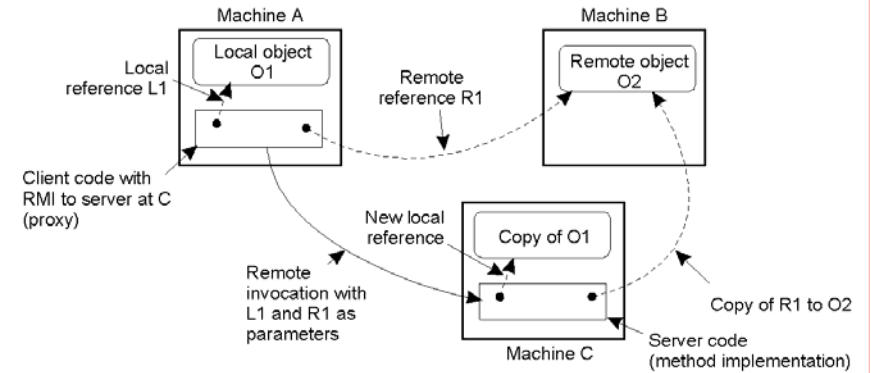
(b)

- (a) Example with implicit binding using only global references
- (b) Example with explicit binding using global and local references

6/9/2005

25

Parameter Passing



The situation when passing an object by reference or by value.

6/9/2005

26

Java RMI

See

<http://java.sun.com/developer/onlineTraining/rmi/RMI.html>