DISTRIBUTED SYSTEMS
Principles and Paradigms
Second Edition
ANDREW S. TANENBAUM
MAARTEN VAN STEEN

Chapter 4
Communication

1

Layered Protocols (1)

Figure 4-1. Layers, interfaces, and protocols
in the OSI model.

2

Layered Protocols (2)

Data link layer header

Network layer header

Transport layer header

Session layer header

Presentation layer header

Application layer header

Message

Data link layer trailer

Bits that actually appear on the network

Figure 4-2. A typical message as it appears on the network.

3

Middleware Protocols

Application	Application protocol	6
Middleware	Middleware protocol	5
Transport	Transport protocol	4
Network	Network protocol	3
Data link	Data link protocol	2
Physical	Physical protocol	1

Network

Figure 4-3. An adapted reference model
for networked communication.

4

Types of Communication

Figure 4-4. Viewing middleware as an intermediate (distributed) service in application-level communication.

Tanenbaum & Van Steen, Distributed Systems: Principles and Paradigms, 2e, (c) 2007 Prentice-Hall, Inc. All rights reserved. 0-13-239227-5

5

Conventional Procedure Call
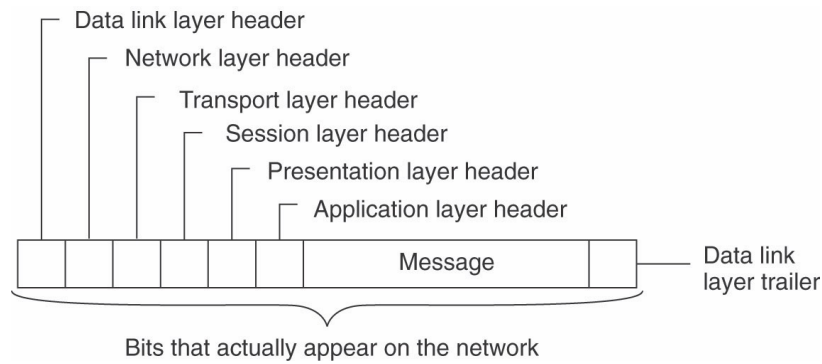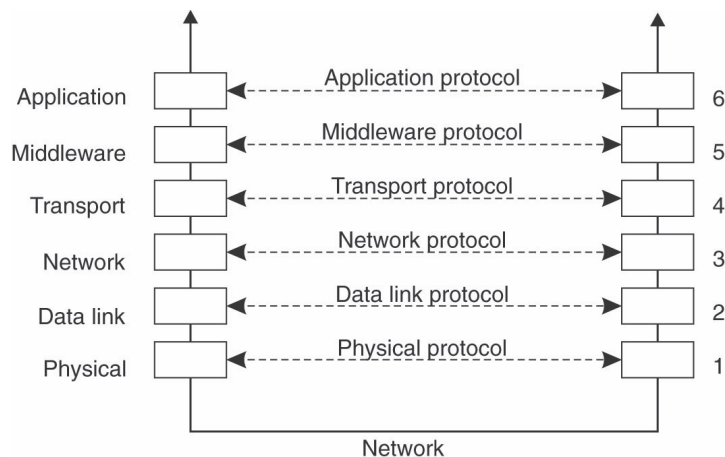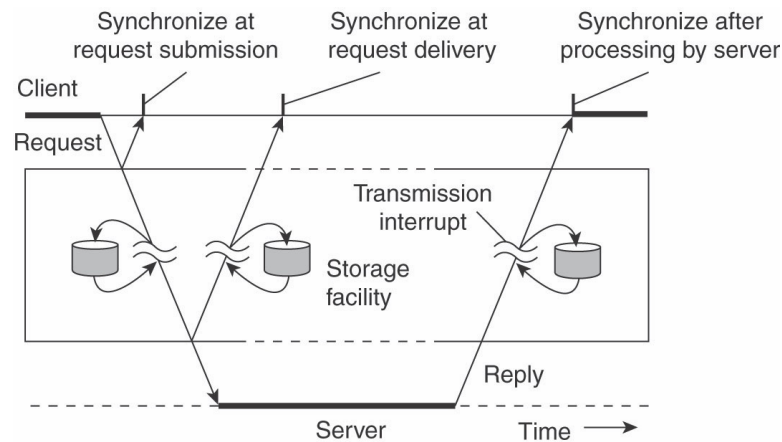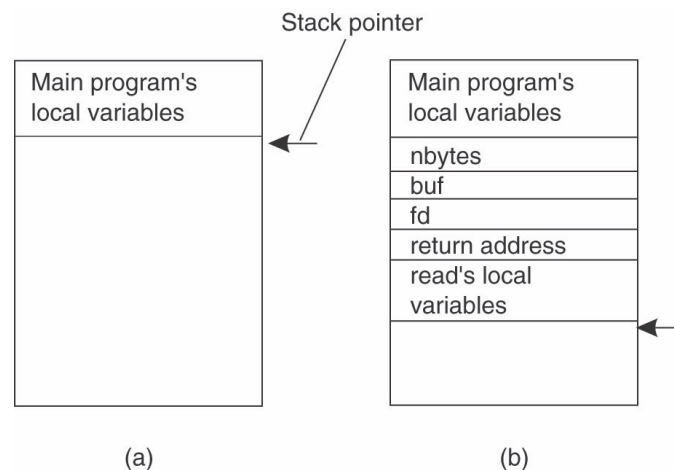

Figure 4-5. (a) Parameter passing in a local procedure call: the stack before the call to read. (b) The stack while the called procedure is active.

Tanenbaum & Van Steen, Distributed Systems: Principles and Paradigms, 2e, (c) 2007 Prentice-Hall, Inc. All rights reserved. 0-13-239227-5
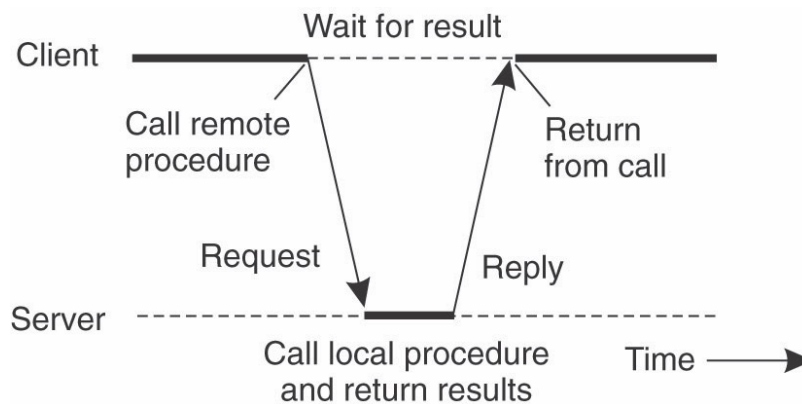
6

Client and Server Stubs

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

7

Remote Procedure Calls (1)

A remote procedure call occurs in the following steps:

1. The client procedure calls the client stub in the normal way.
2. The client stub builds a message and calls the local operating system.
3. The client's OS sends the message to the remote OS.
4. The remote OS gives the message to the server stub.
5. The server stub unpacks the parameters and calls the server.

Continued …

8

Remote Procedure Calls (2)

A remote procedure call occurs in the following steps (continued):

6. The server does the work and returns the result to the stub.
7. The server stub packs it in a message and calls its local OS.
8. The server's OS sends the message to the client's OS.
9. The client's OS gives the message to the client stub.
10. The stub unpacks the result and returns to the client.

9

Passing Value Parameters (1)
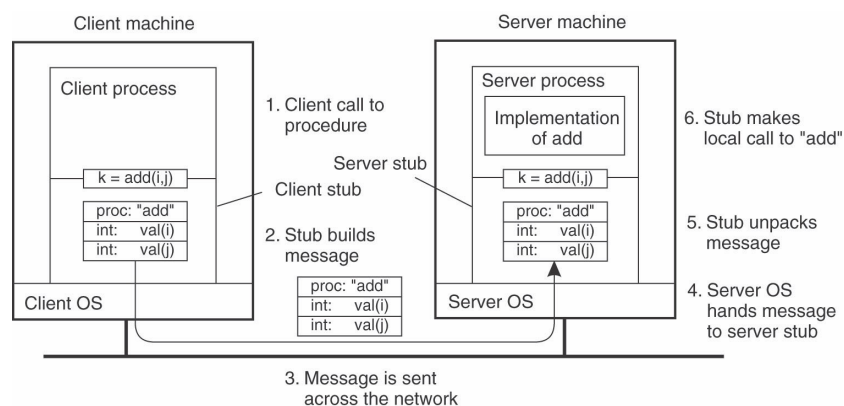
Figure 4-7. The steps involved in a doing a remote computation through RPC.

10

Passing Value Parameters (2)

(a)

Figure 4-8. (a) The original message on the Pentium.

11

Passing Value Parameters (3)

(b)

Figure 4-8. (a) The original message on the SPARC.

12

Passing Value Parameters (4)

0 :	1 :	2 :	3 :
0	0	0	5
4 :	5 :	6 :	7 :
L	L	I	J

(c)

Figure 4-8. (c) The message after being inverted. The little numbers in boxes indicate the address of each byte.

13

Parameter Specification and Stub Generation

```
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)

Figure 4-9. (a) A procedure. (b) The corresponding message.

14

Asynchronous RPC (1)

(a)

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

15

Asynchronous RPC (2)

(b)

Figure 4-10. (b) The interaction using asynchronous RPC.

16

Asynchronous RPC (3)

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

Tanenbaum & Van Steen, Distributed Systems: Principles and Paradigms, 2e, (c) 2007 Prentice-Hall, Inc. All rights reserved. 0-13-239227-5
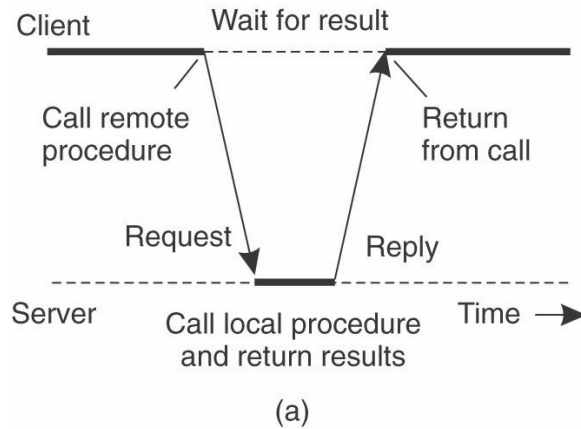
17

Writing a Client and a Server (1)
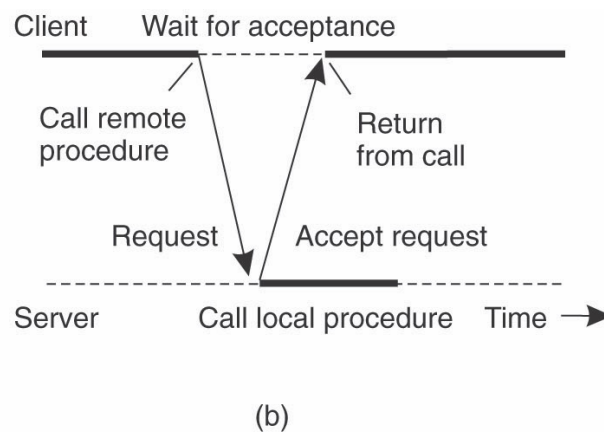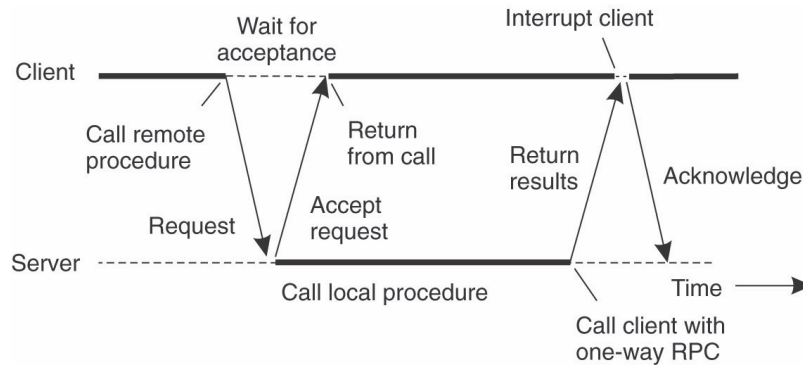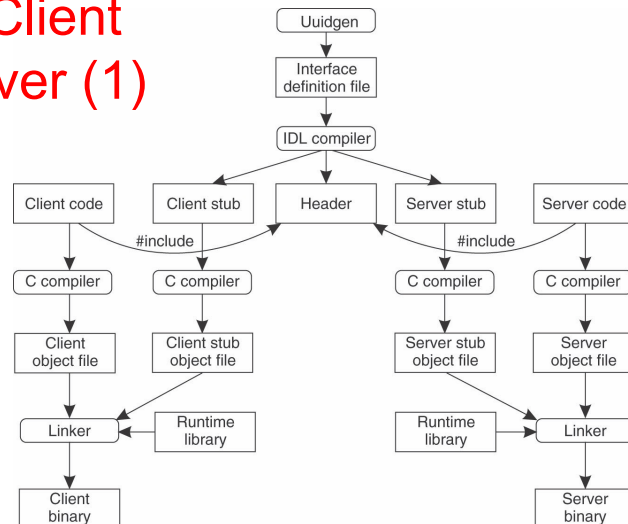
Figure 4-12. The steps in writing a client and
a server in DCE RPC.

Tanenbaum & Van Steen, Distributed Systems: Principles and Paradigms, 2e, (c) 2007 Prentice-Hall, Inc. All rights reserved. 0-13-239227-5

18

Writing a Client and a Server (2)

Three files output by the IDL compiler:

- A header file (e.g., interface.h, in C terms).
- The client stub.
- The server stub.

19

Binding a Client to a Server (1)

- Registration of a server makes it possible for a client to locate the server and bind to it.

- Server location is done in two steps:
  1. Locate the server's machine.
  2. Locate the server on that machine.

20

Binding a Client to a Server (2)
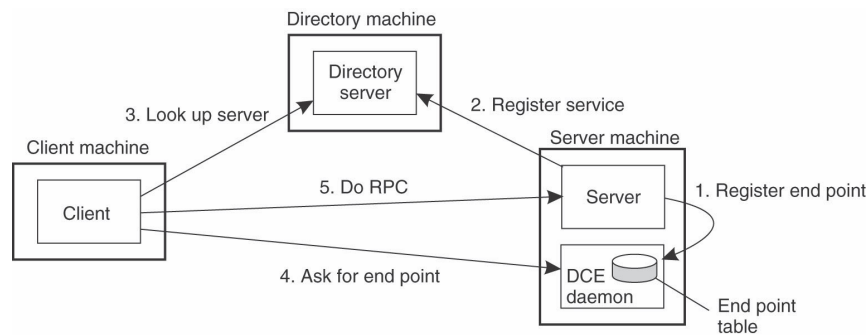
Figure 4-13. Client-to-server binding in DCE.

21

Berkeley Sockets

Primitive	Meaning
Socket	Create a new communication end point
Bind	Attach a local address to a socket
Listen	Announce willingness to accept connections
Accept	Block caller until a connection request arrives
Connect	Actively attempt to establish a connection
Send	Send some data over the connection
Receive	Receive some data over the connection
Close	Release the connection

Figure 4-14. The socket primitives for TCP/IP.

22