

第二章 分布式系统的通信

进程间的通信是一切分布式系统的基础，它基于底层网络提供的底层消息传递机制

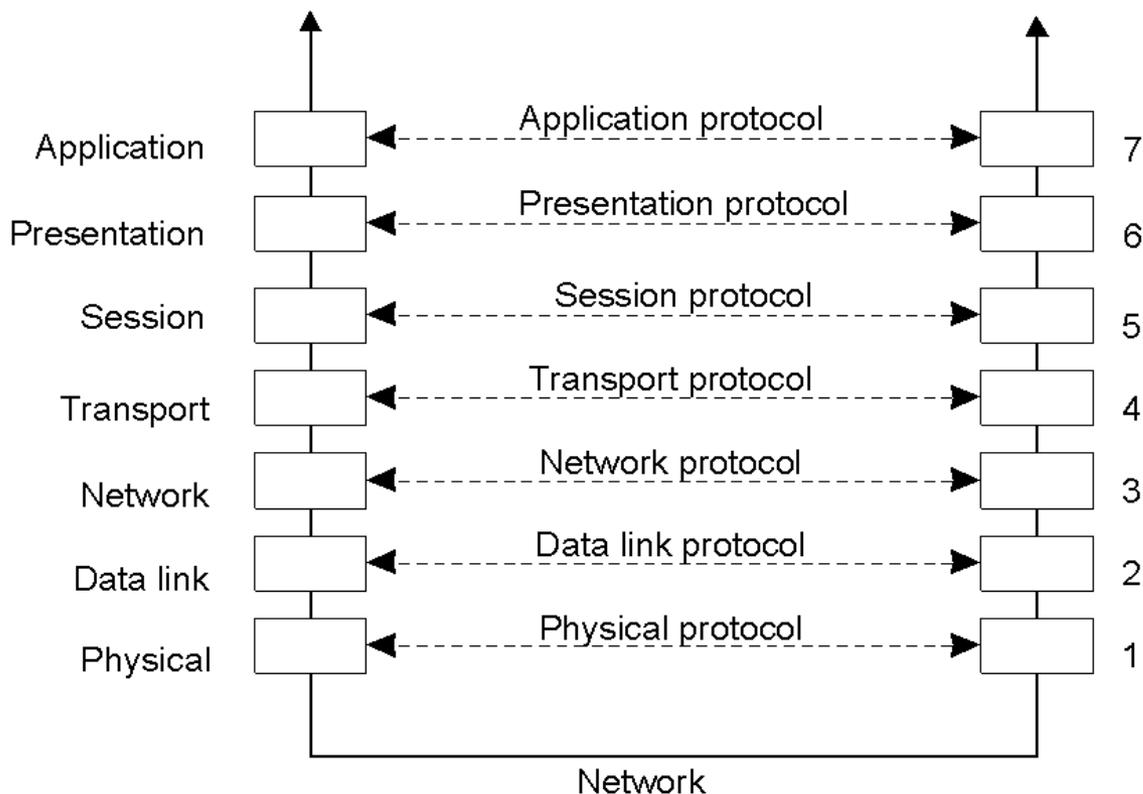
- 分层协议
- 远程过程调用
- 远程对象调用
- 面向消息的通信

层次协议 (1)

- OSI 模型中的层、接口和协议

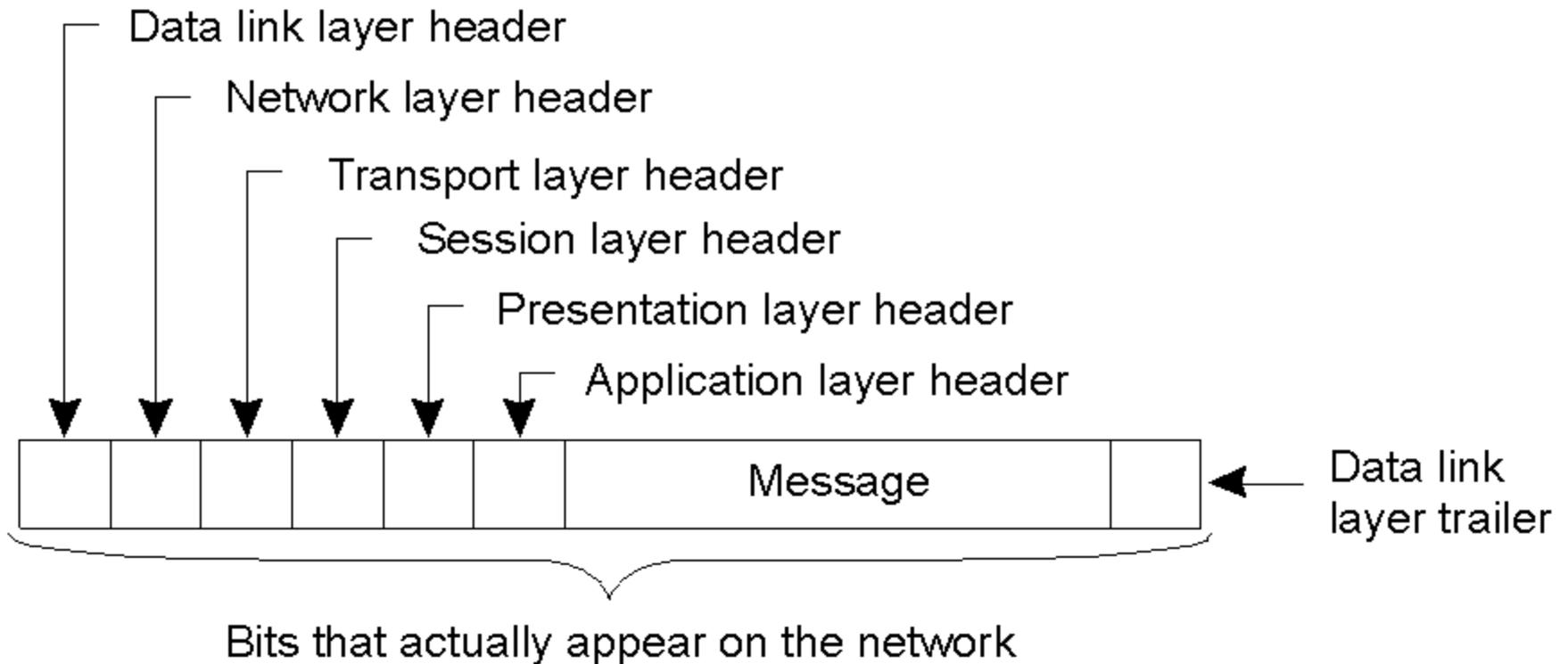
必须在不同层次制订多种协议，包括从位传输的底层细节到信息表示的高层细节：

- 0, 1 的电压表示
- 消息的结束位
- 检测消息的丢失或损坏及其处理
- 数值、字符串及其它数据项的长度和表示方法
- 面向连接的协议：电话
- 无连接的协议：邮箱



层次协议(2)

- 在网络上传输的典型消息



远程过程调用

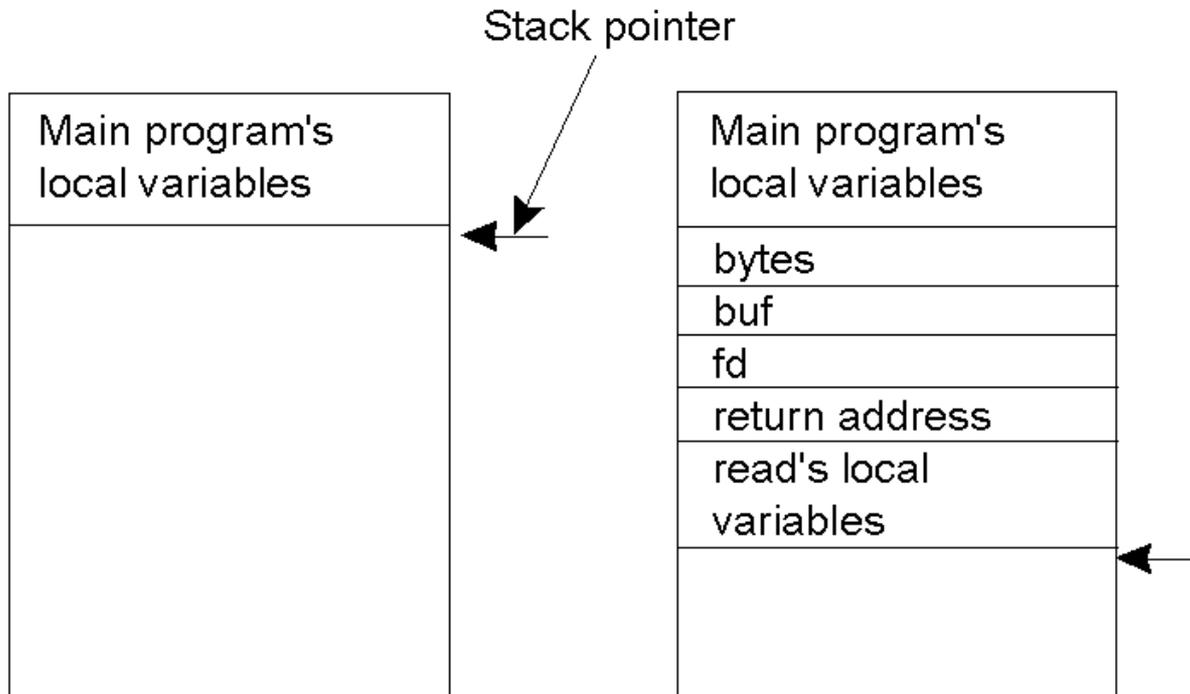
(Remote Procedure Call)

RPC是分布式系统通信处理的事实标准

- 常规过程调用
- 客户存根和服务端存根
- 参数传递

常规过程调用

- a) `Count=read(fd,buf,nbyte)`
- b) 本地过程调用中的参数传递：调用`read`前的堆栈状态
- c) 过程调用执行时的堆栈状态



(a)

(b)

客户存根和服务端存根

- 客户和服务端间的RPC原理

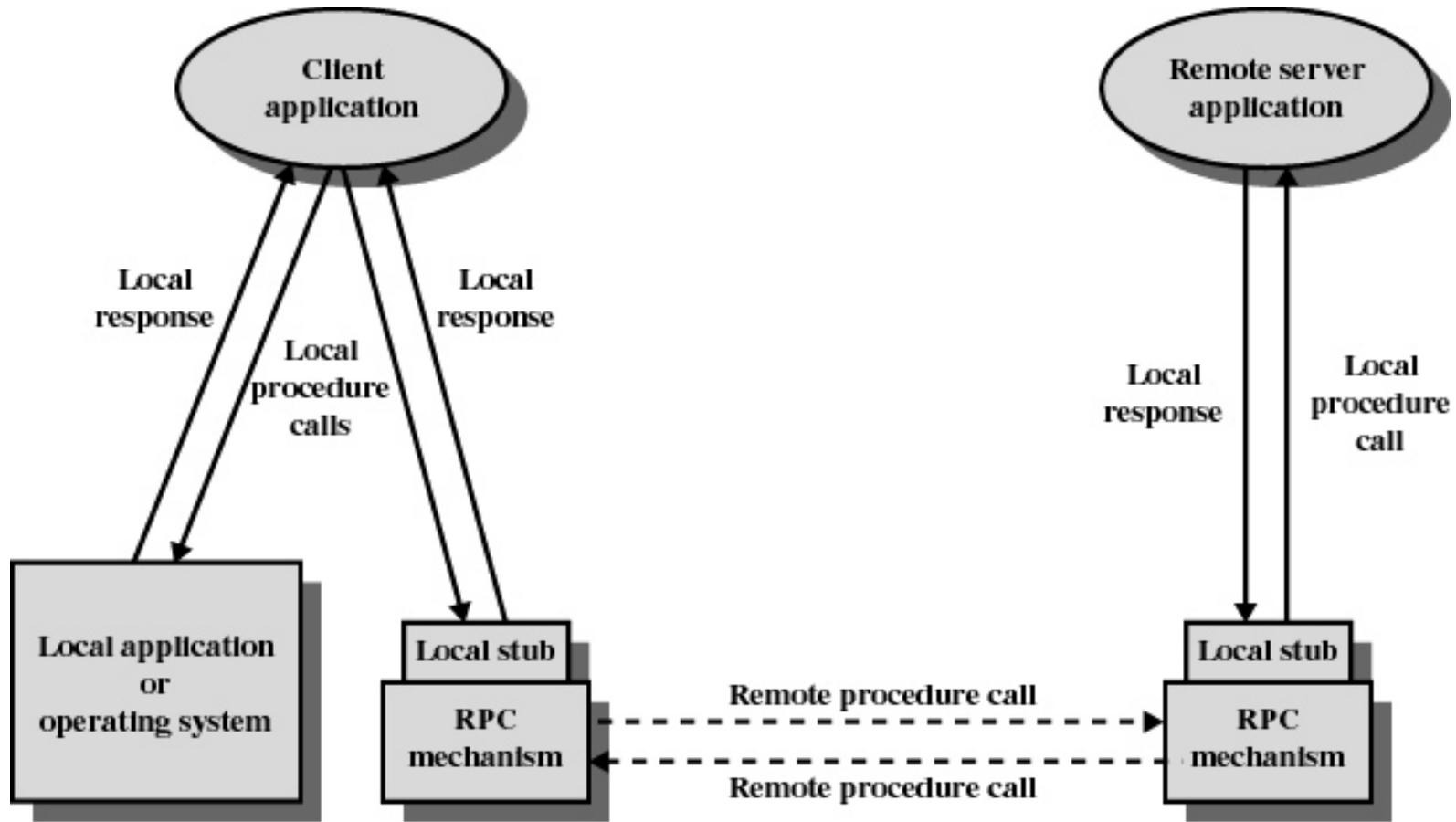


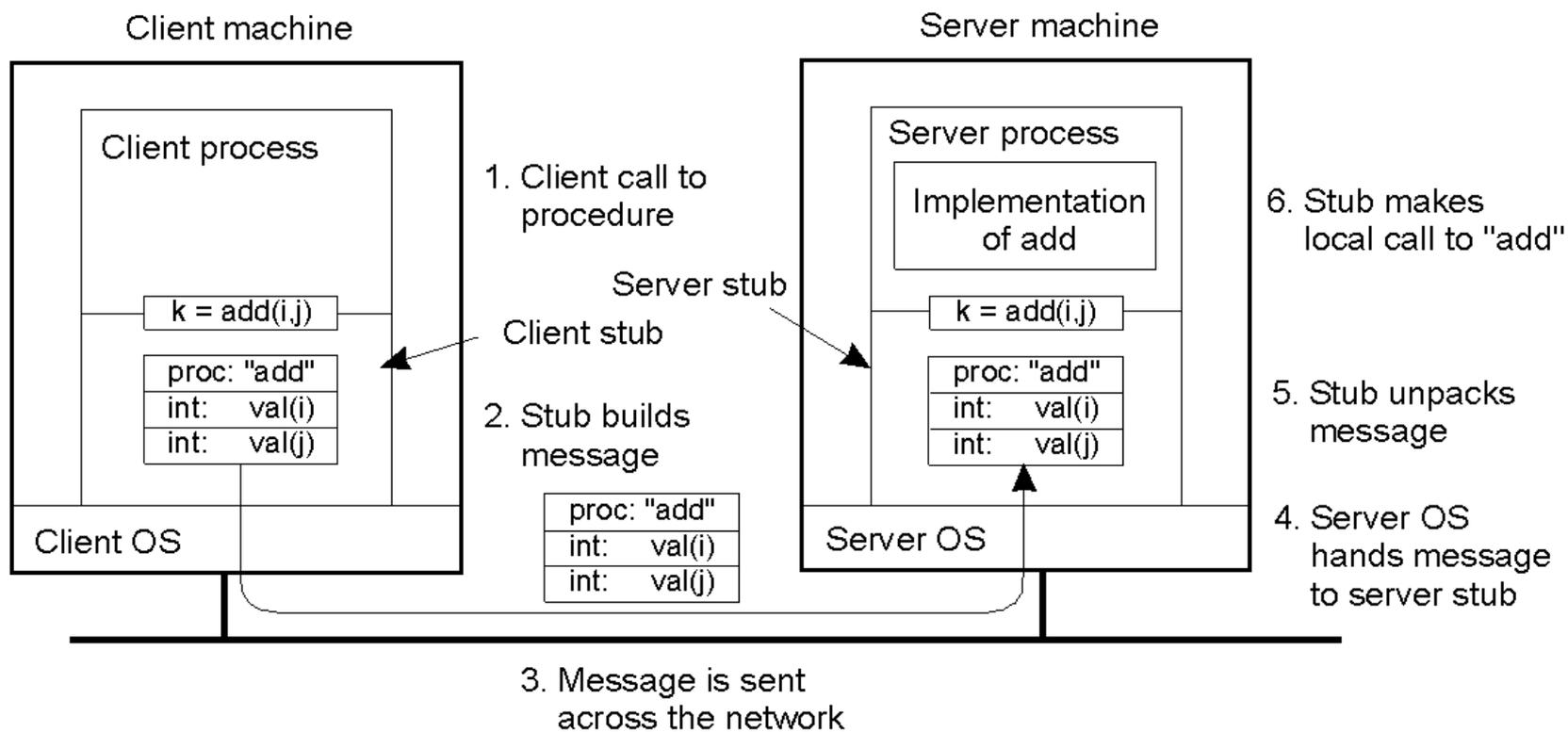
Figure 13.13 Remote Procedure Call Mechanism

远程过程调用步骤

1. 客户过程以正常的方式调用客户存根
2. 客户存根生成一个消息，然后调用本地操作系统
3. 客户端操作系统将消息发送给远程操作系统
4. 远程操作系统将消息交给服务器存根
5. 服务器存根将参数提取出来，然后调用服务器
6. 服务器执行要求的操作，操作完成后将结果返回给服务器存根
7. 服务器存根将结果打包成一个消息，然后调用本地操作系统
8. 服务器操作系统将含有结果的消息发送回客户端操作系统
9. 客户端操作系统将消息交给客户存根
10. 客户存根将结果从消息中提取出来，返回给调用它的客户过程

参数传递-传递值参(1)

- 通过RPC进行远程计算的步骤



传递值参 (2)

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

(a)

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

(b)

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

(c)

a) Pentium上的原始消息

b) SPARC收到的消息

c) 进行逆转后的消息

传递引用参数

- 对于简单数组和结构：使用复制-还原代替引用调用
- 很难传递一般意义的指针：如复杂图形的指针

参数说明

RPC双方必须就交换的格式达成一致

- 一个过程
- 相应的消息

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

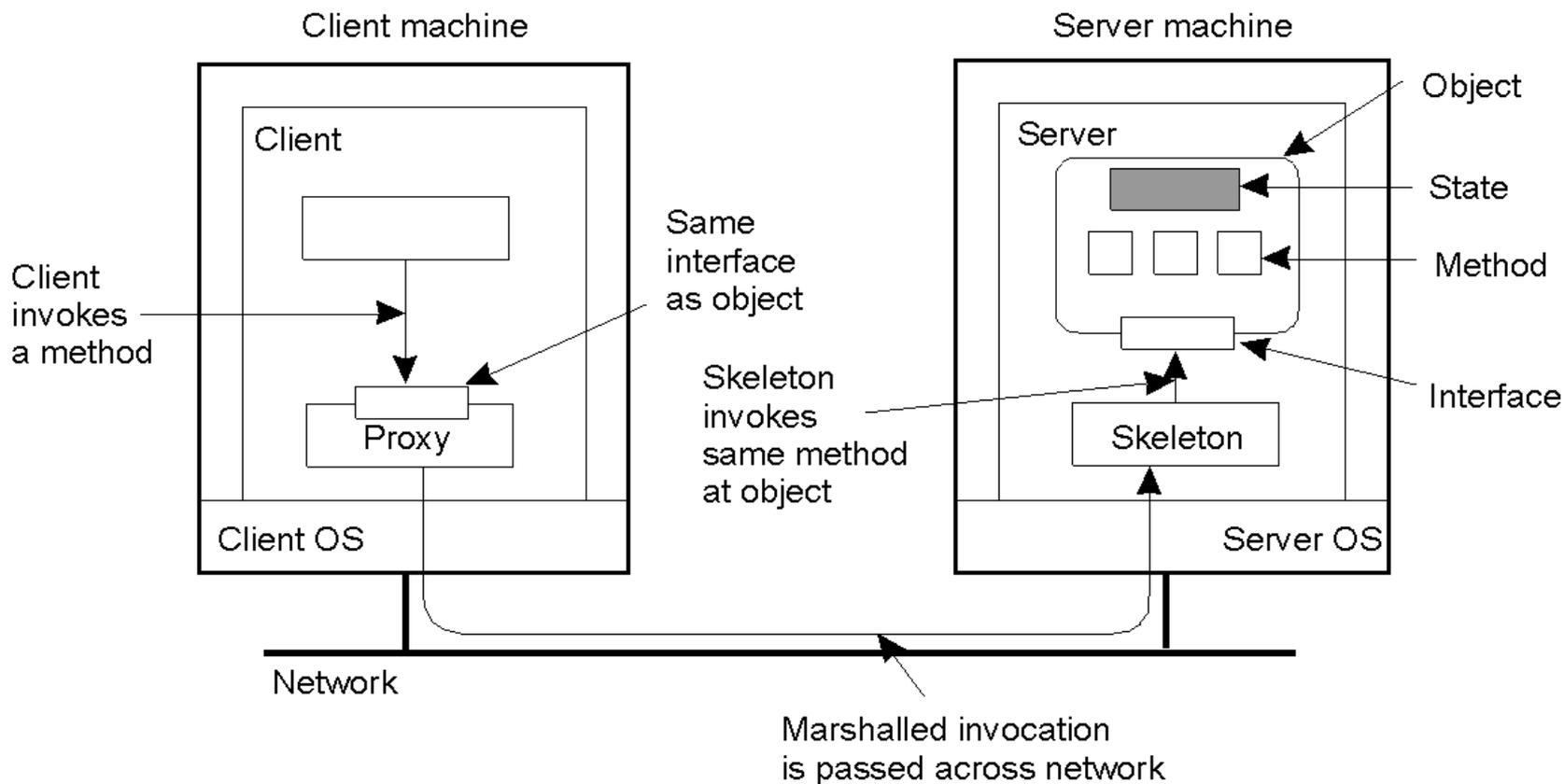
(a)

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

(b)

远程对象调用

- 使用客户端代理的远程对象的一般组织结构



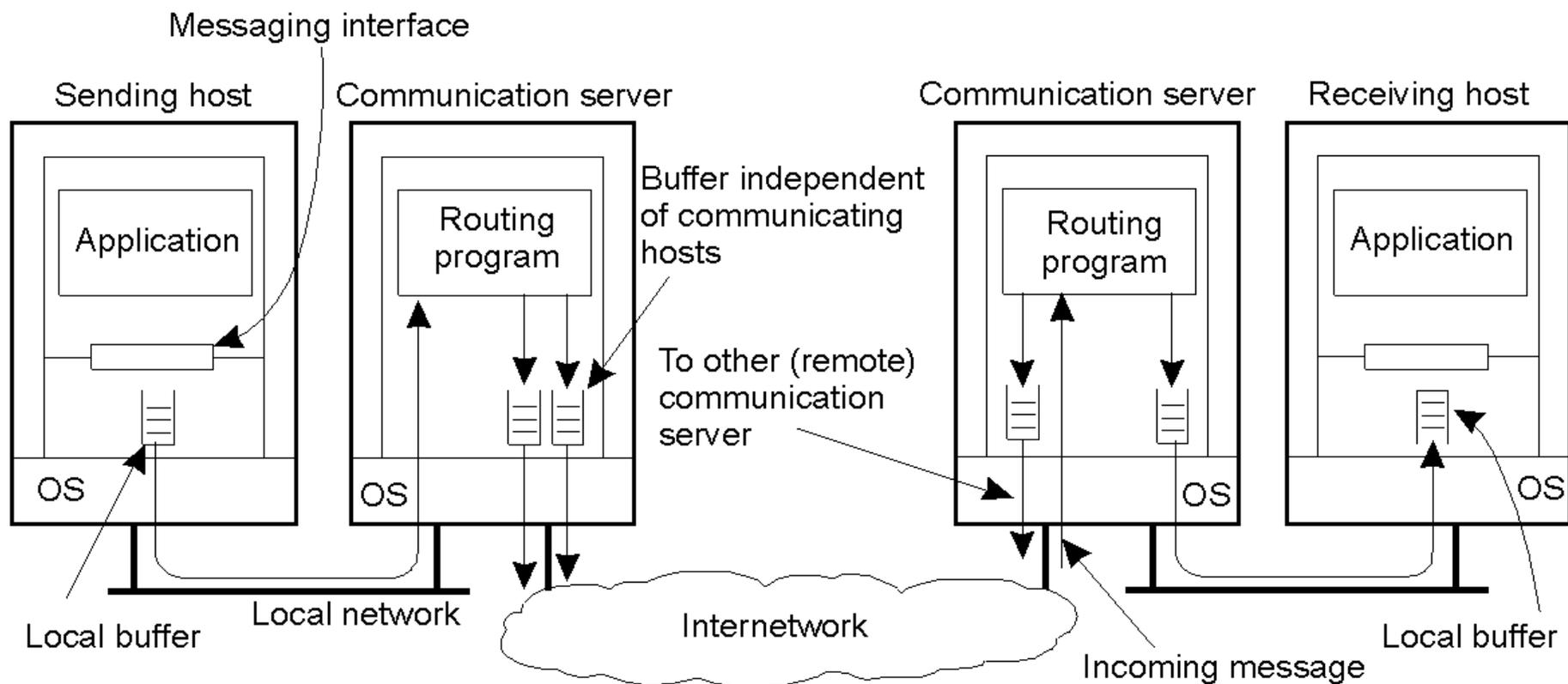
面向消息的通信

当远程过程调用和远程对象调用不适用时，需要面向消息的通信。

- 消息中的持久性和同步性
- 面向消息的暂时通信
- 面向消息的持久通信

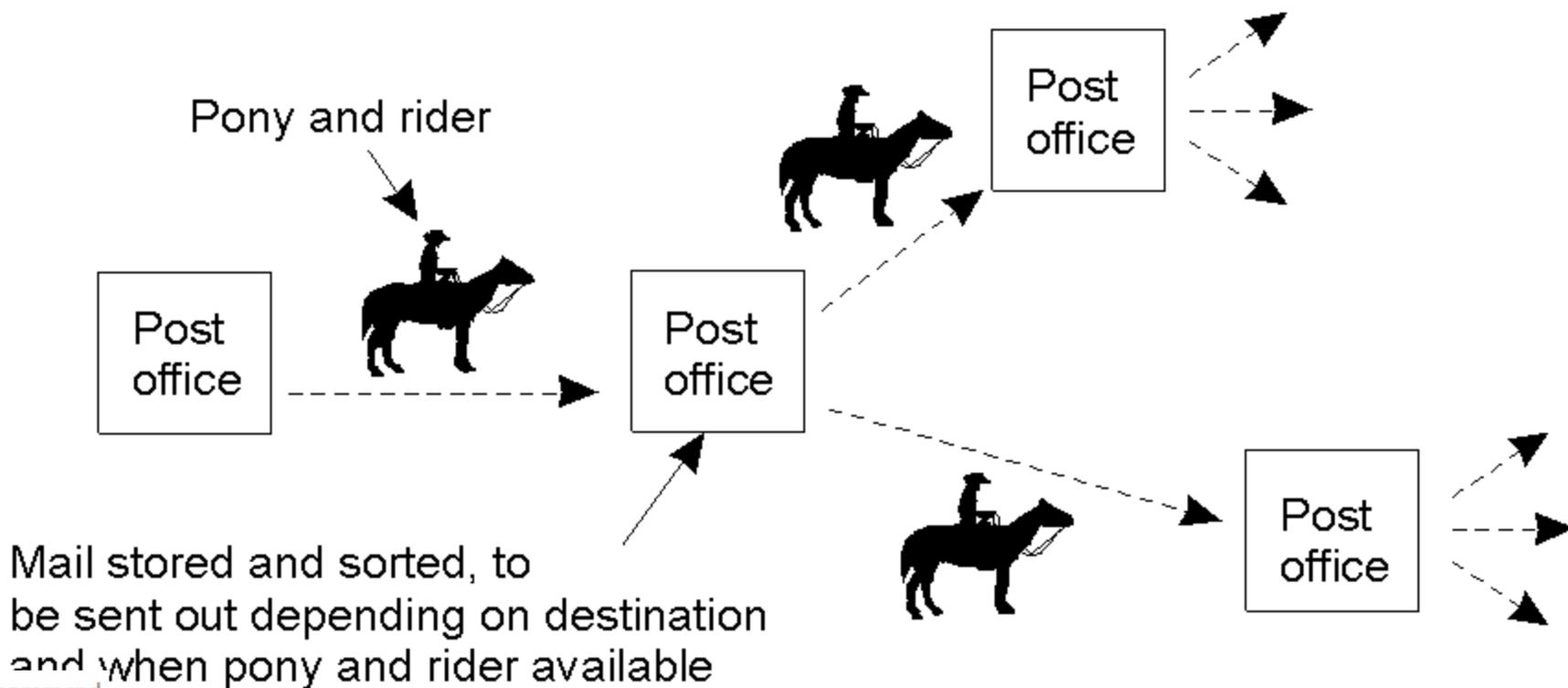
消息中的持久性和同步性(1)

- 通信系统的通用结构



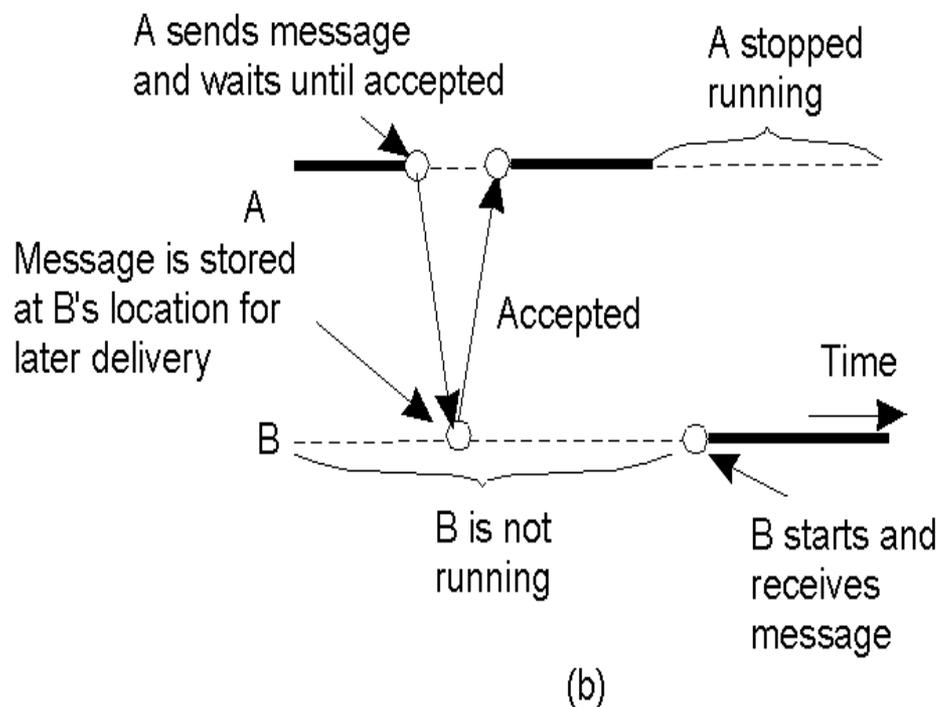
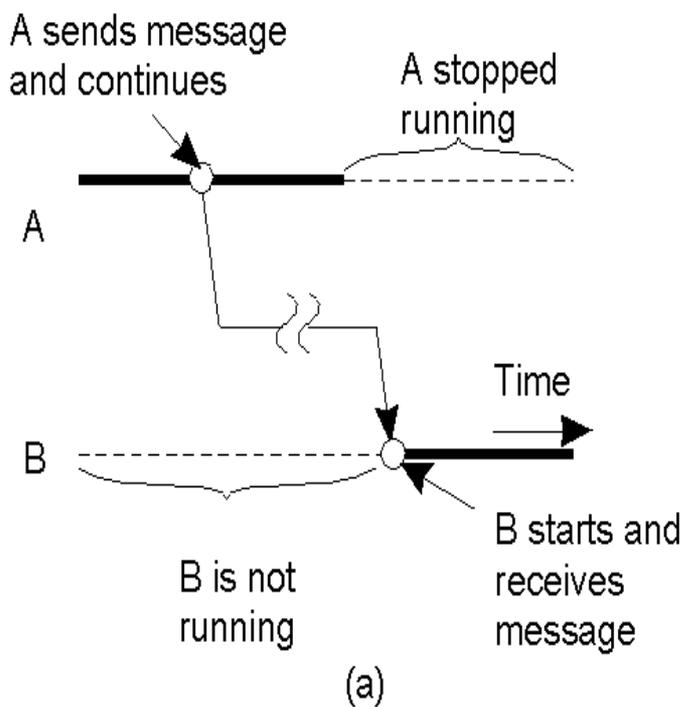
消息中的持久性和同步性(2)

- 在持久通信中，通信双方不必保持运行
- 在暂时通信中，通信系统只在发送者和接收者运行时存储消息

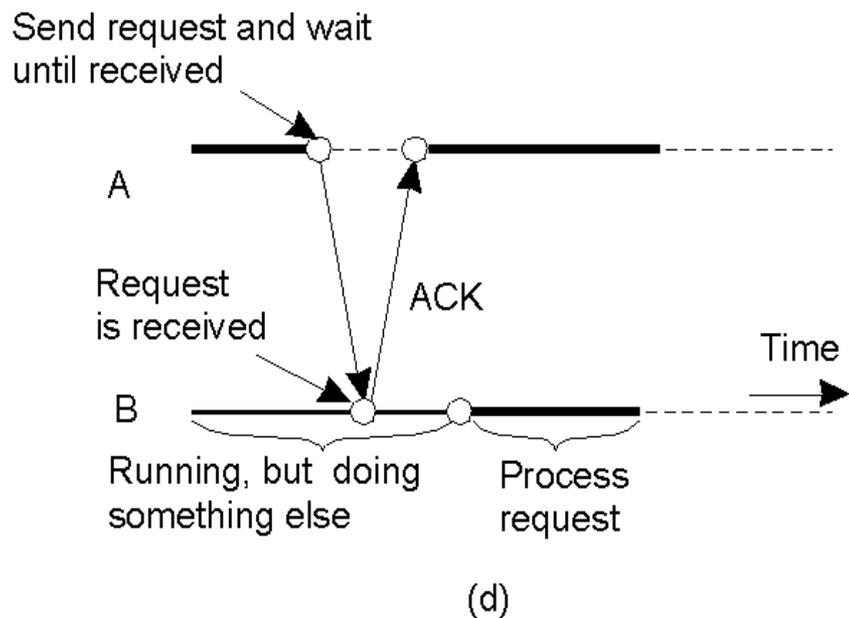
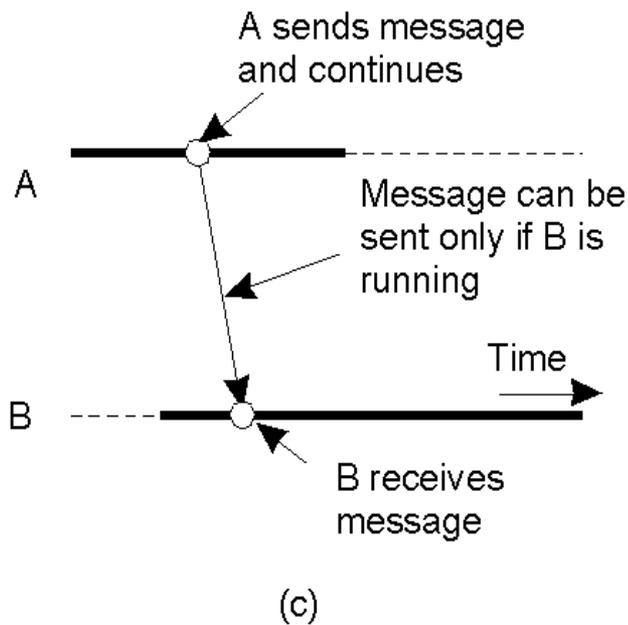


消息中的持久性和同步性(3)

- a) 持久**异步通信**: 提交消息后立即执行其他程序, 电子邮件
- b) 持久**同步通信**: 提交消息后会被阻塞, 直到消息已到达并存储在接收主机

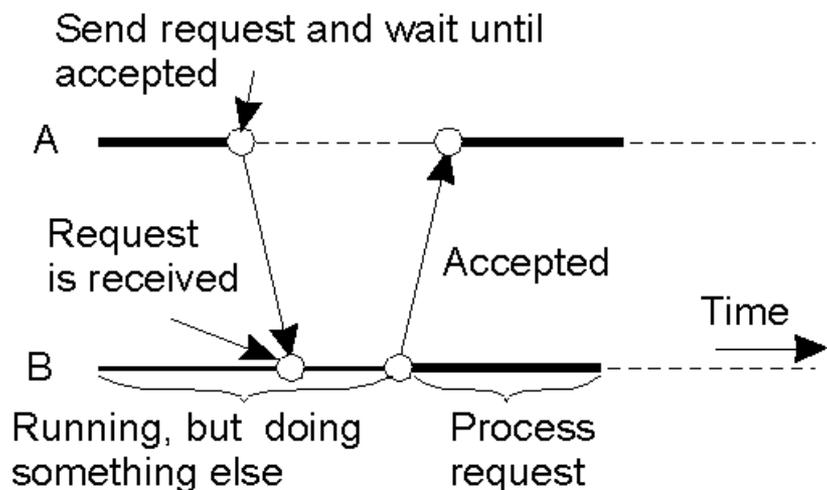


消息中的持久性和同步性(4)

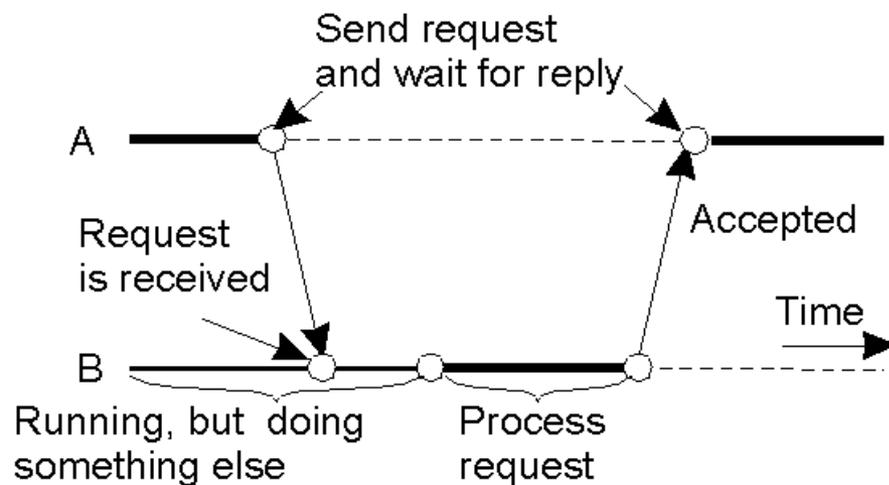


- c) 暂时异步通信
- d) 基于接收的暂时同步通信

消息中的持久性和同步性(5)



(e)



(f)

- e) 基于交付的暂时通信
- f) 基于响应的同步通信

面向消息的暂时通信

不提供消息的中介存储，实时性要求高
(几秒甚至几毫秒)

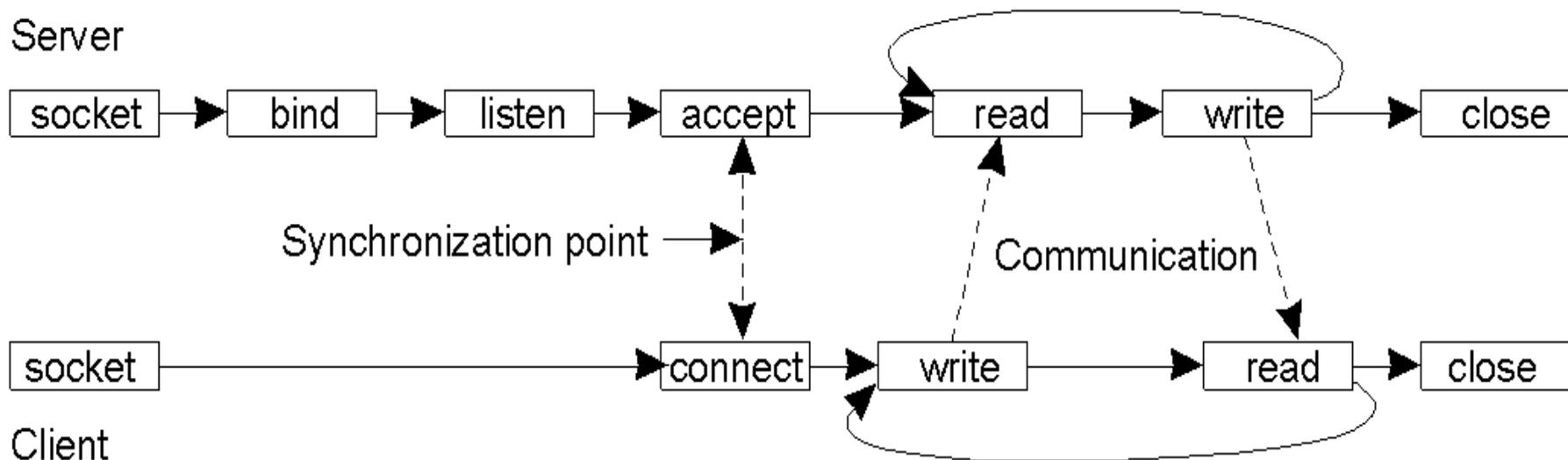
- **Berkeley Sockets**
- **Message-Passing Interface**

Berkeley Sockets (1)

- TCP/IP套接字原语

原语	含义
Socket	创建新的通信端点
Bind	将本地地址附加 (attach) 到套接字上
Listen	宣布已准备好接受连接
Accept	在收到连接请求前阻塞调用方
Connect	主动尝试建立连接
Send	通过连接发送数据
Receive	通过连接接收数据
Close	释放连接

Berkeley Sockets (2)



- 使用套接字的面向连接通信模式

The Message-Passing Interface (MPI)

- 适用于高速互连网络（**COW, MPP**），为高性能并行应用程序设计，有更高级的特性：缓冲、同步
- **MPI**中的基本消息传递原语

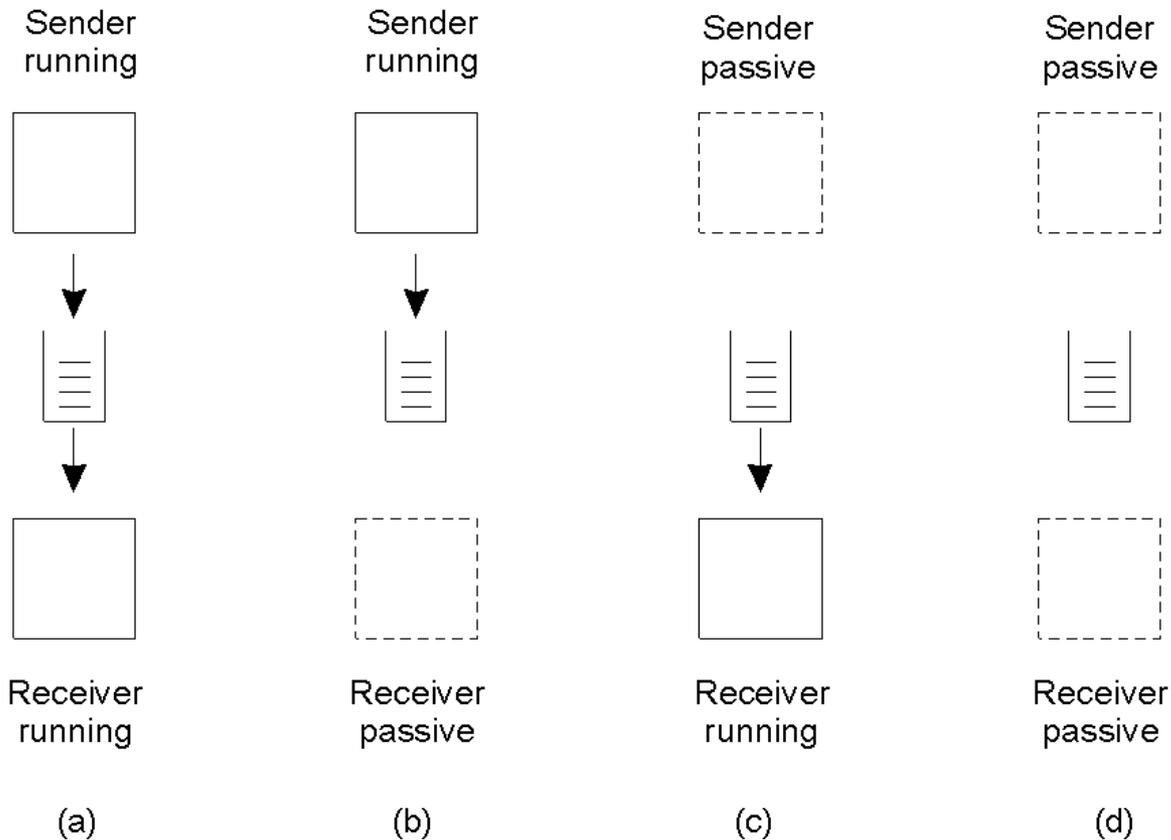
原语	含义
MPI_bsend	将消息追加到本地发送缓冲区中
MPI_send	发送消息，并等待到消息复制到本地或远程缓冲区为止
MPI_ssend	发送消息，并等待到对方开始接收为止
MPI_sendrecv	发送消息，并等待到受到应答消息为止
MPI_isead	传递要发送消息的引用，然后继续执行
MPI_issend	传递要发送消息的引用，并等待到对方开始接收为止
MPI_recv	接受消息，如果不存在等待的消息则阻塞
MPI_irecv	检查是否有输入的消息，但无论有无消息都不会阻塞

面向消息的持久通信

- 提供消息的中介存储，实时性要求低（几分钟）
- **Message-Queuing system**消息队列系统
 - 应用程序通过在特定队列中插入消息来进行通信
 - 只保证发送者的消息最终被放置到接收者的队列中，并不保证时间，也不保证消息被读取。

Message-Queuing Model (1)

- 使用队列的松耦合通信的四种组合方式



Message-Queuing Model (2)

- 消息队列系统中队列的基本接口

∴

原语 含义

Put 将消息追加到指定队列

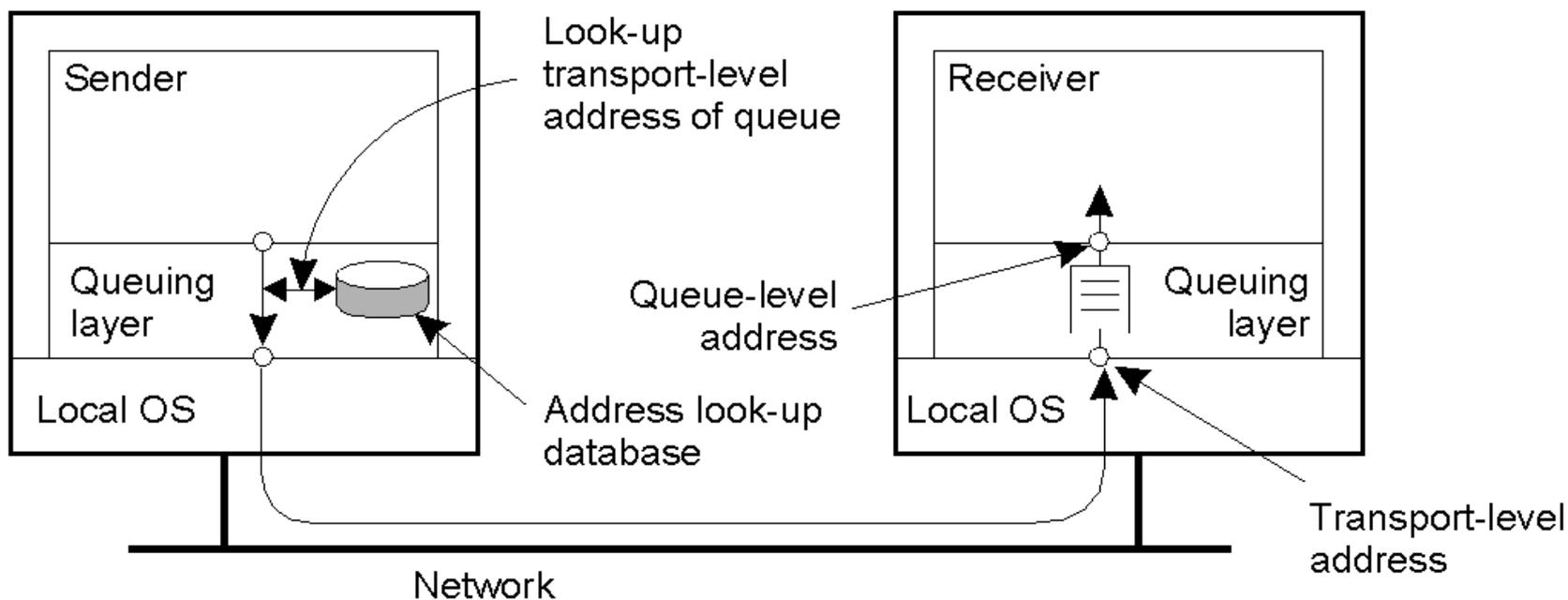
Get 调用进程阻塞，直到指定队列非空，取出第一个消息

Poll 察看指定队列的消息，取出第一个消息，不阻塞调用进程

Notify 注册一个处理程序，在有消息进入指定队列时调用该处理程序

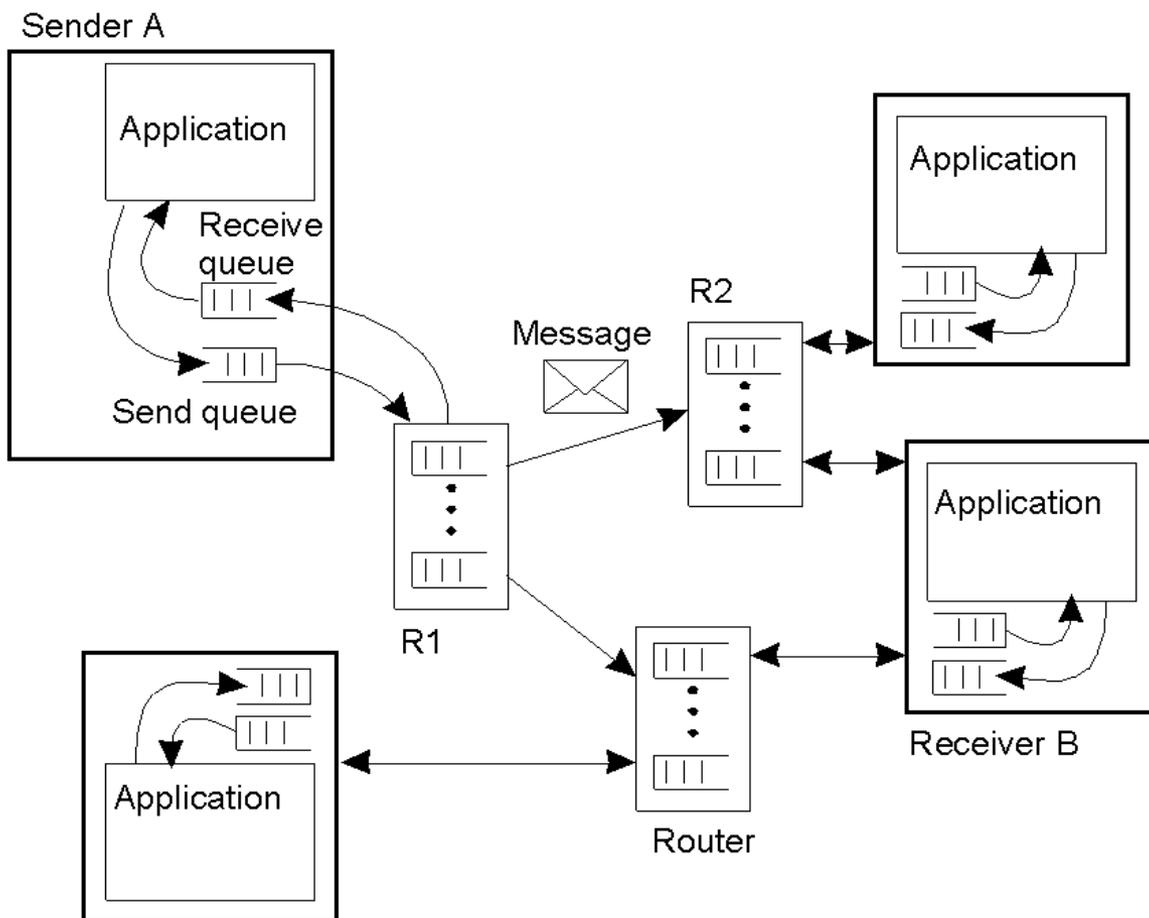
消息队列系统的通用体系结构(1)

- 源队列，目的队列
- 队列级编址与网络级编址间的关系

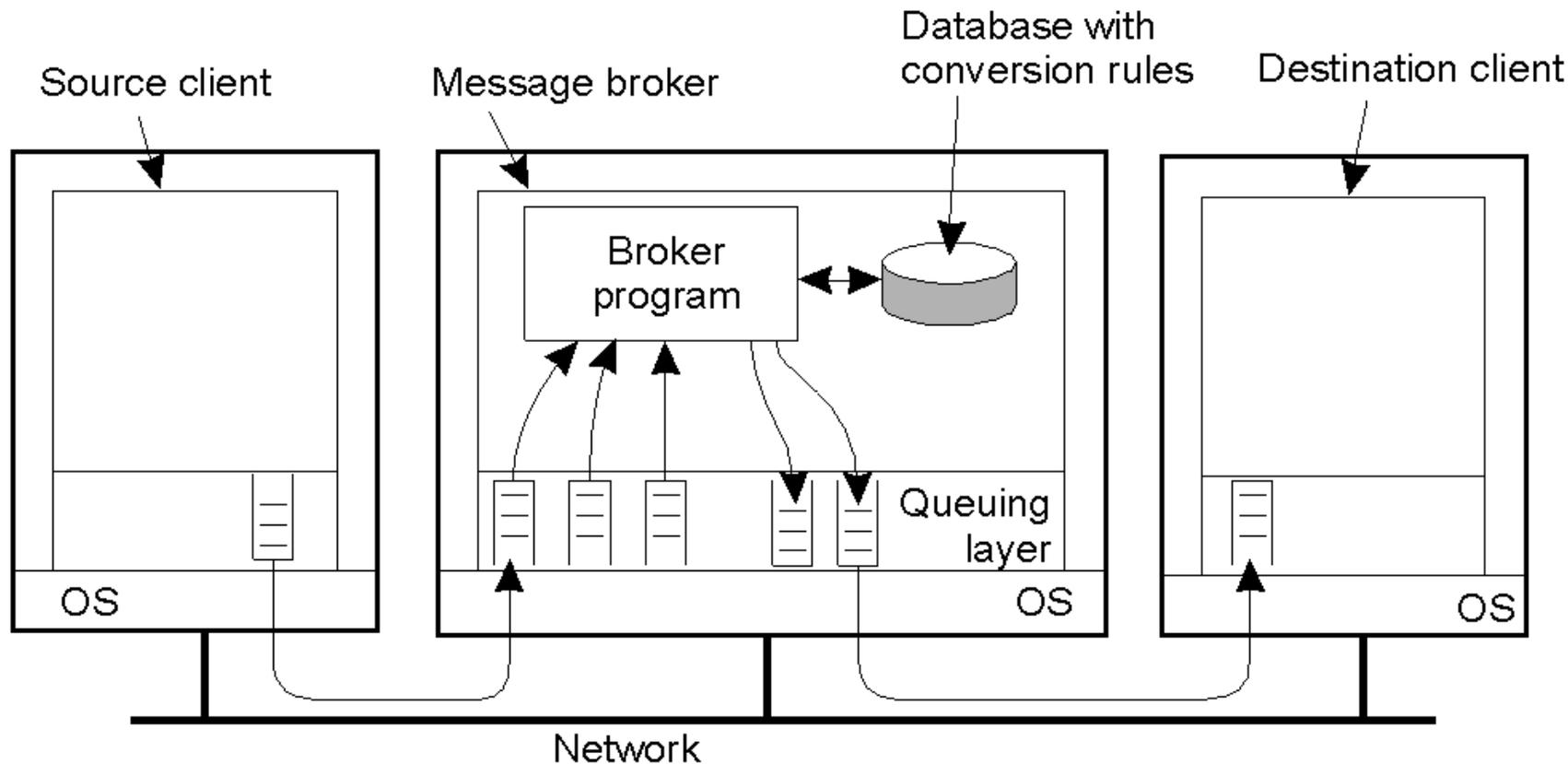


消息队列系统的通用体系结构(2)

- 含有路由器的消息队列系统的通用体系结构



(消息转换器) Message Brokers



小 结

- 分层协议
- 远程过程调用
- 远程对象调用
- 面向消息的通信

习题

1. **SPARC**芯片使用**32**位长的字，以最高有效字节的格式存储。而**486**的存储格式为最低有效字节优先。如果从**SPARC**传送整数**2**到**486**，那么在**486**上看到的数值是多少？
2. 假设一个空**RPC**（**0**字节数据）需时间**1.0 ms**，每增加**1KB**数据，时间增加**1.5 ms**。如果要从文件服务器读**32 KB**的数据，请计算一次读取**32 KB**数据的**RPC**所需的时间和**32**次读取**1KB**数据的**RPC**所需的时间？