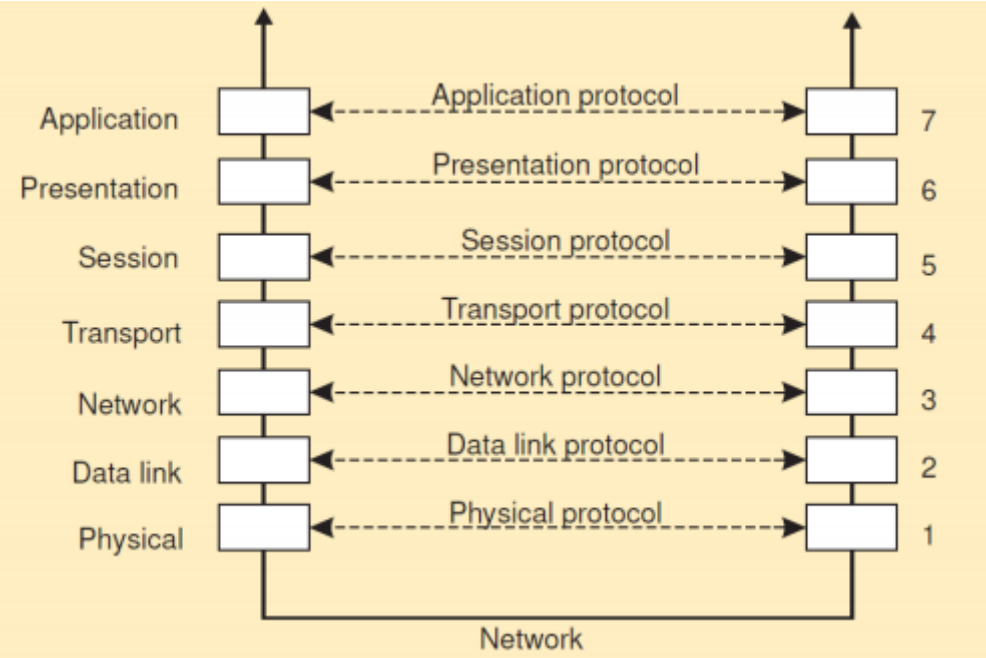
4 分布式系统通信

Author： 中山大学 17数据科学与计算机学院 YSY

https://github.com/ysyisyourbrother

分层协议

基本网络模型



底层通信

物理层： 包含发送数据的规约和实现，负责发送和接收端的传输；

数据链路层：将发送数据整理成“帧”，并且提供验错和流控的功能

网络层：描述数据包如何在网络中的计算机之间如何路由；

在很多分布式系统中，**最底层的接口 其实是网络层。**

传输层

对于大部分分布式系统来说，传输层提供实际的通信功能

TCP：面向连接、可靠的、面向流的通信

UDP：不可靠的（尽力而为）数据报文通信

中间件层

中间件提供通用的服务和协议，可用于支撑很多不同的应用

通信类型

持久通信

提交传输的消息一直由通信中间件存储，直到该消息被传送给接收方为止。因此对发送方而言，没有必要持续运行应用程序，对接收方也是。电子邮件系统就是一个典型的持久通信

瞬时通信

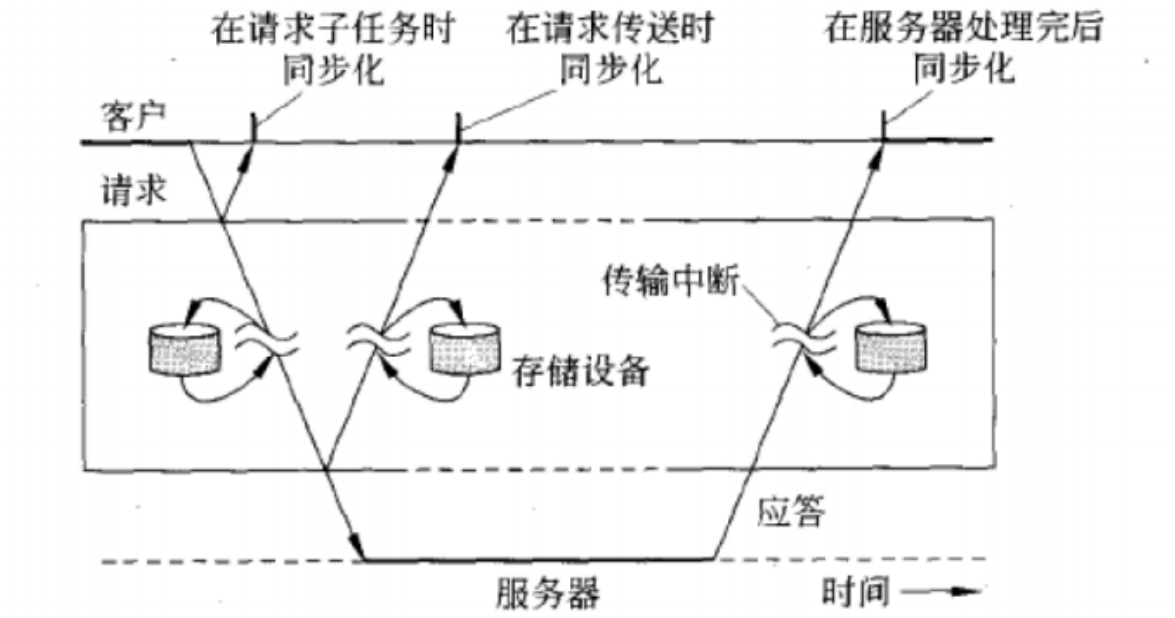
通信系统只有在发送和接收应用程序正在运行的时候才存储消息。如果传输中断或接收方掉线，中间件就会把消息丢弃。比如路由器如果无法将数据包传到下一个路由器，就会丢弃该消息。

异步通信

发送方在提交了要传输的数据后就立刻继续执行。消息由中间件临时储存起来

同步通信

1. 发送方被阻塞直到知道其请求被接受。 同步发生时刻：请求提交时
2. 发送方为同步化，直到其请求被传送给目标接收方 同步发生时刻：请求传输时
3. 通过让发送方一直等到其请求被完全处理，知道接收方返回了一个响应，就可以实现同步化 同步发生时刻： 请求处理完后



举例：客户端/服务器通信

客户/服务器计算系统一般是基于瞬态、同步的通信方法：

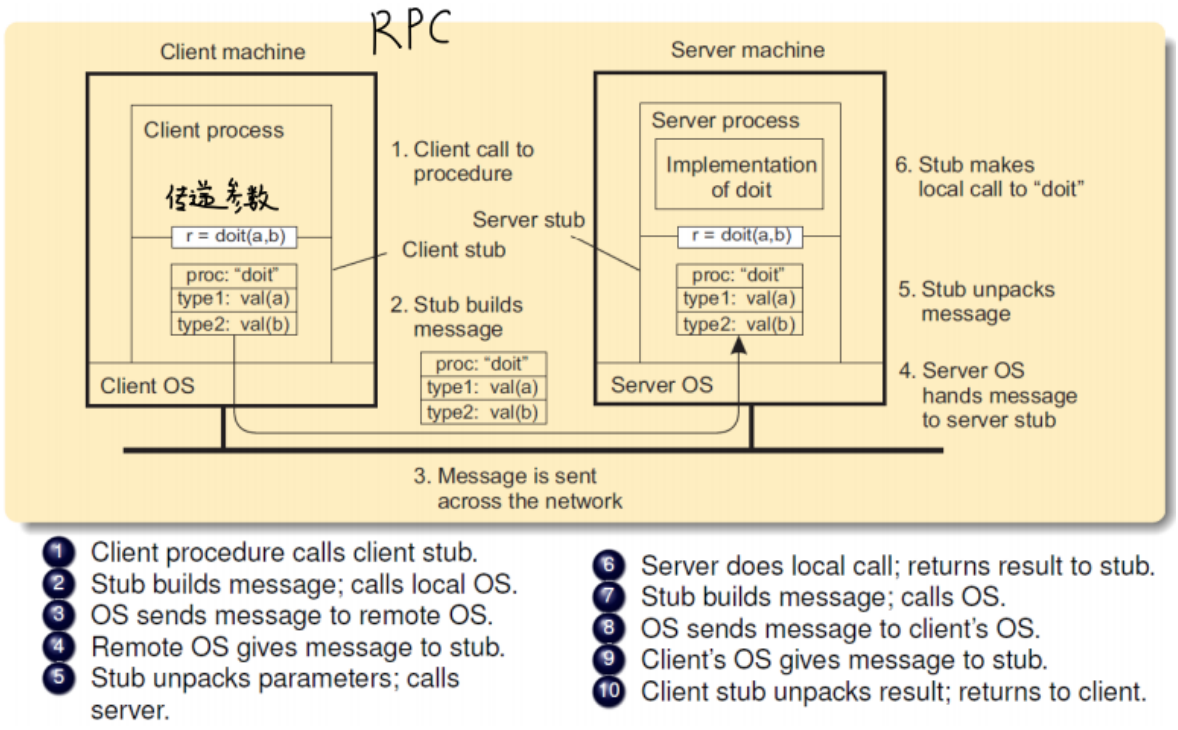
1. 客户和服务器在通信时必须处于活跃的状态
2. 客户端发出请求后，被阻塞直到收到应答
3. 服务器只是等待到来的请求，然后处理这些请求

不适用于邮件传输系统

远程过程调用

机器A上的进程调用机器B上的进程，A上的调用进程被挂起，而B上的被调用进程开始执行。调用方通过使用参数将信息传送给被调用方，然后可以通过传回的结果得到信息。

编程人员看不到任何消息传递的过程



RPC参数传递

客户端和服务器可能有不同的数据表示

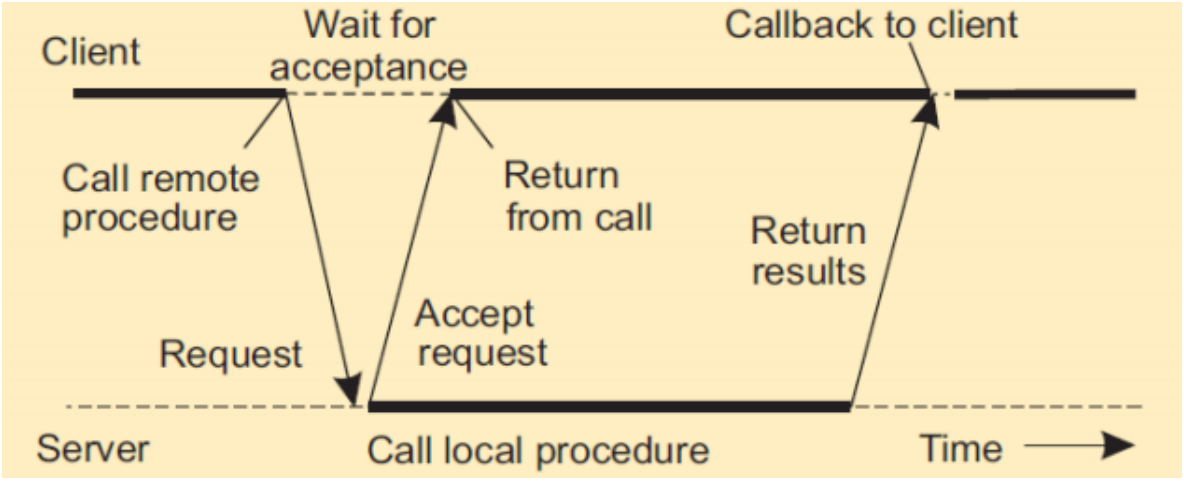
客户端和服务器应具有一致的编码机制：

1. 基本的数据类型如何表示
2. 复杂的数据类型如何表示

客户端和服务器需要正确地解释消息，并将其转换成与机器无关的表 达形式

异步RPC

摒弃严格的请求-响应行为，但是让客户端连续运行而不需要等待服 务器返回信息



面向消息的通信

远程过程调用有助于隐藏分布式系统中的通信，但也存在缺点：

1. 必须保证服务器正在运行
2. 客户端会发生同步阻塞的问题

面向消息的瞬时通信

Berkeley套接字

原语	含义
socket	创建新的通信端点
bind	将本地地址附加到套接字上
listen	宣布已准备好接受连接
accept	在收到连接请求之前阻塞调用方
connect	主动尝试确立连接
send	通过连接发送数据
receive	通过连接接收数据
close	释放连接

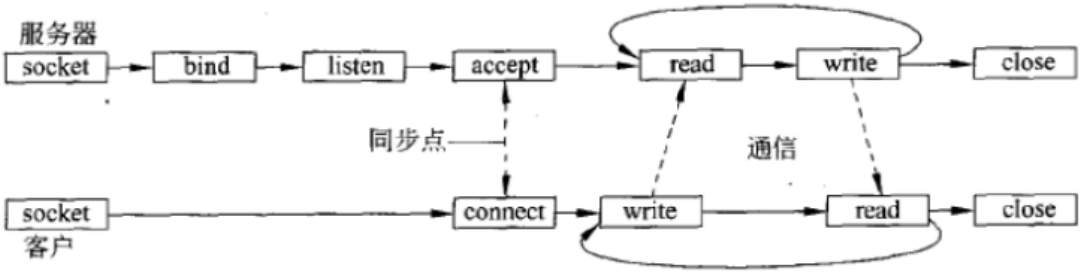


图 4.15　使用套接字的面向连接通信模式

消息传递接口MPI

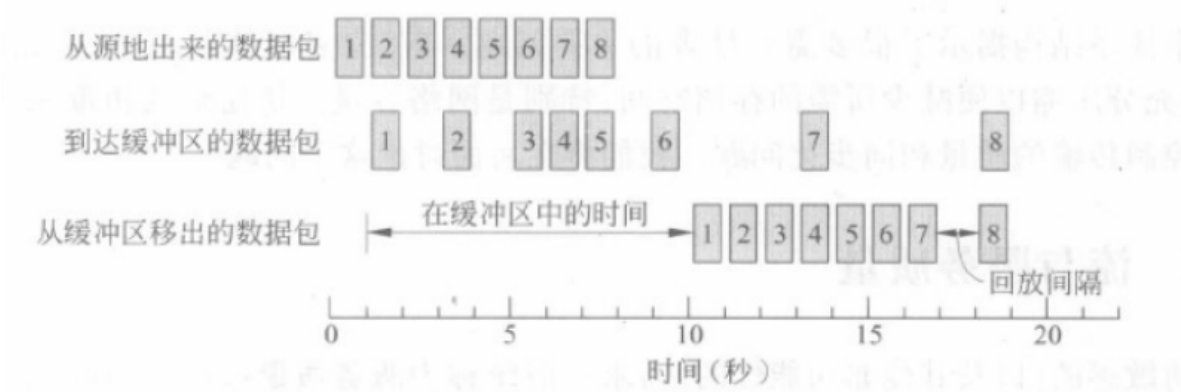
套接字的缺点：

1. 灵活性比较差，提供的功能简单
2. 套接字使用通用的协议栈（TCP/IP）进行网络通信；不适 用于专用协议

Operation	Description
MPI_bsend	Append outgoing message to a local send buffer
MPI_send	Send a message and wait until copied to local or remote buffer
MPI_ssend	Send a message and wait until transmission starts
MPI_sendrecv	Send a message and wait for reply
MPI_isend	Pass reference to outgoing message, and continue
MPI_issend	Pass reference to outgoing message, and wait until receipt starts
MPI_recv	Receive a message; block if there is none
MPI_irecv	Check if there is an incoming message, but do not block

面向流的通信

数据流是数据单元的序列，可以应用于离散的媒体，也可以应用于 连续媒体



多播通信

随着点对点技术，尤其是结构化覆盖网络管理的出现，通信通路的创建更简单了。由于点对点结局方法通常是部署在应用层，因此已引入了多种应用层多播技术。

应用层多播

其本质是将分布式系统组织成一个覆盖网络，然后利用这个网络 分发数据

覆盖网络的构建方法

1. 组织成树，导致每对节点之间只有唯一的路径
2. 组织成网络结构，每个节点都有多个邻居节点 （健壮性高）

Gossip数据通信

对传播信息来说，一种技术是依靠感染行为。研究人员观察了疾病是如何在人群中扩散的，于是进行了很长时间调查，看看是否可以开发些简单技术在超大型分布式系统中进行信息扩散。

消息传播模型

具有要传播到其他结点数据的称为：**已感染的**

还没有接受到数据的节点称为：**易受感染的**

不会传播其数据的已更新结点称为：**已隔离的**

Anti-entropy （反/逆熵）模型

结点P随机选取另一节点Q，然后与Q交换更新信息。

1. P只是把它自己的更新信息发出给Q，即为push的方法
2. P只是从Q那里获得更新信息，即为基于pull的方法
3. P和Q相互发送更新信息给对方，基于push-pull的方法

只基于push的方法是一个不好的选择。在基于push的方法中，更新信息只能由已感染的结点来传播。如果已经感染的结点很多，选择一个易感染的结点的概率就小，到后面速度会比较慢

只基于pull的方法，易感染的结点主动从感染结点出接受信息更新

如果只有一个结点是已经感染的，使用反熵模型的任一种方法都可以很快的传播。但push-pull方法最好，如果将每个节点作为发起者与随机选择的一个结点进行至少一次的信息交换，定义为一轮，则单个信息更新到所有节点需要$O(log(N))$轮。