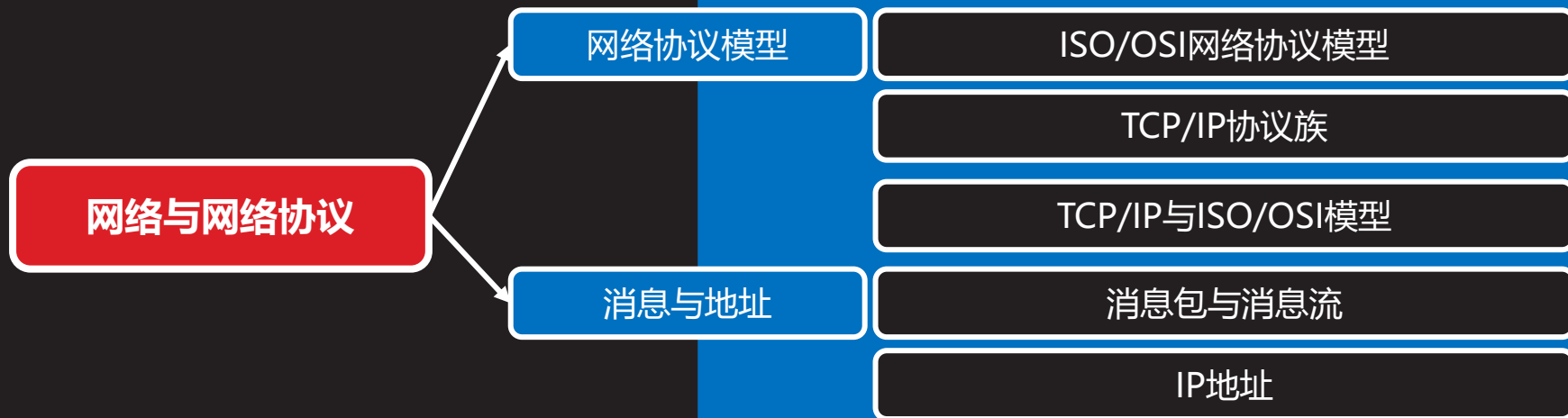


# Unix系统高级编程

网络和套接字

Unit23

# 网络与网络协议



# 网络协议模型



# ISO/OSI网络协议模型

- 什么是计算机网络？
  - 计算机网络，是指将地理位置不同的具有独立功能的多台计算机及其外部设备，通过通信线路连接起来，在网络操作系统、网络管理软件及网络通信协议的管理和协调下，实现资源共享和信息传递的计算机系统



# ISO/OSI网络协议模型 (续1)

- 什么是网络协议？
  - 网络协议是一种特殊的软件，是计算机网络实现其功能的最基本的机制。网络协议的本质就是规则，即各种硬件和软件必须遵循的共同守则。网络协议并不是一套单独的软件，它融合于其它所有的软件甚至硬件系统中，因此可以说协议在网络中无所不在



# ISO/OSI网络协议模型 (续2)

- 什么是协议栈?
  - 为了减少网络设计的复杂性，绝大多数网络采用分层设计的方法。所谓分层设计，就是按照信息的流动过程将网络的整体功能分解为一个个的功能层，不同机器上的同等功能层之间采用相同的协议，同一机器上的相邻功能层之间通过接口进行信息传递。各层的协议和接口统称为协议栈



# ISO/OSI网络协议模型 (续3)

- 描述计算机网络各协议层的一般方法是采用国际标准化组织 (International Standardization Organization, ISO)的计算机通信开放系统互连(Open System Interconnection, OSI)模型, 简称ISO/OSI网络协议模型

|                                  |       |                   |                 |
|----------------------------------|-------|-------------------|-----------------|
| 上层协议<br>定义网络<br>数据的格式及各种<br>网络应用 | 应用层   | 为用户的应用程序提供各种网络服务  | http/ftp/telnet |
|                                  | 表示层   | 将不同数据格式转换为一种通用格式  | ascii/jpeg/mpeg |
|                                  | 会话层   | 建立、管理和终止通信主机间的对话  | 安排访问次序          |
| 底层协议<br>定义数据<br>如何传输<br>到目的地     | 传输层   | 提供端到端的流量控制维持可靠传输  | TCP/UDP         |
|                                  | 网络层   | 路径选择、路由、IP寻址、建立连接 | IP/SPX、路由器      |
|                                  | 数据链路层 | 物理寻址、网络拓扑结构、错误检测  | MAC、网桥          |
|                                  | 物理层   | 高低电平、速率与距离、物理连接器  | HUB/中继器/线缆      |



# TCP/IP协议族

- TCP/IP不是个单一的网络协议，而是由一组具有层次关系的网络协议组成的协议家族，简称TCP/IP协议族
  - TCP：传输控制协议，面向连接，可靠的全双工的字节流
  - UDP：用户数据报协议，无连接，不如TCP可靠但速度快
  - ICMP：网际控制消息协议，处理路由器和主机间的错误和控制消息
  - IGMP：网际组管理协议，用于多播
  - IPv4：网际协议版本4，使用32位地址，为TCP、UDP、ICMP和IGMP提供递送分组服务
  - IPv6：网际协议版本6，使用128位地址，为TCP、UDP和ICMPv6提供递送分组服务





# TCP/IP协议族 (续1)

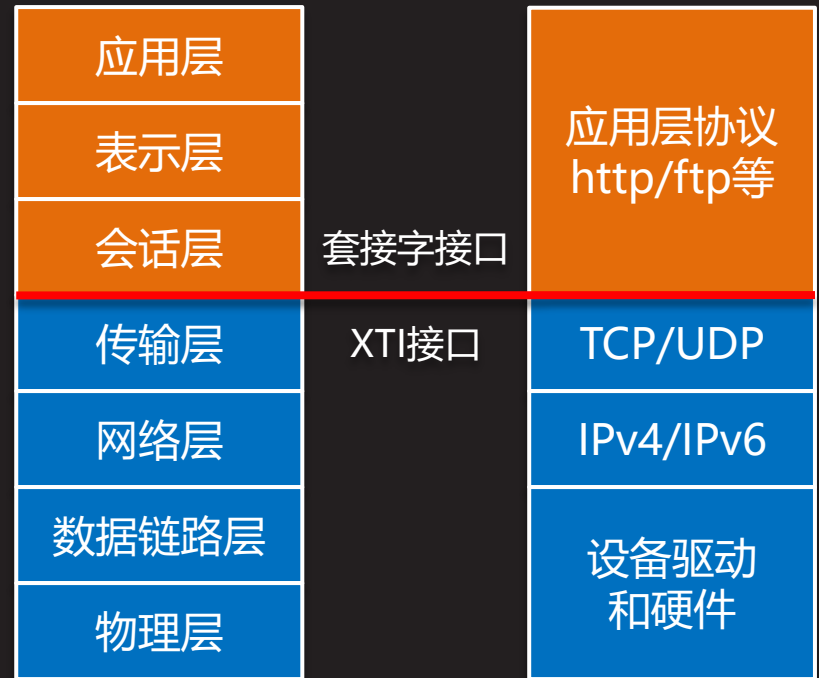
- TCP/IP不是个单一的网络协议，而是由一组具有层次关系的网络协议组成的协议家族，简称TCP/IP协议族
  - ARP：地址解析协议，把IPv4地址映射到硬件地址
  - RARP：逆地址解析协议，把硬件地址映射到IPv4地址
  - ICMPv6：网际控制消息协议版本6，综合了ICMP、IGMP和ARP的功能
  - BPF：BSD分组过滤器，为应用程序提供访问数据链路层的接口，由源自BSD的系统内核提供
  - DLPI：数据链路提供者接口，为应用程序提供访问数据链路层的接口，由源自SVR4的系统内核提供
- 通常所说的TCP、UDP的ICMP等协议都是工作在IP协议之上的，IP协议作为它们的基础协议为其提供服务支撑



# TCP/IP与ISO/OSI模型

- 在ISO/OSI网络协议模型的基础上，TCP/IP协议做了部分合并和简化，同时将网络编程的接口设定在传输层与会话层之间，这样做的理由有二

- 上三层与应用程序的业务逻辑(如数据包的组织与解析、收发的时机与次序等)密切相关，而与具体的通信细节(如收发分组、等待确认、分组排序、计算验证校验和、丢包重传等)关系不大；
- 下四层主要处理通信细节而与具体应用的业务逻辑无关



- 上三层通常构成用户进程，而下四层通常是系统内核的一部分

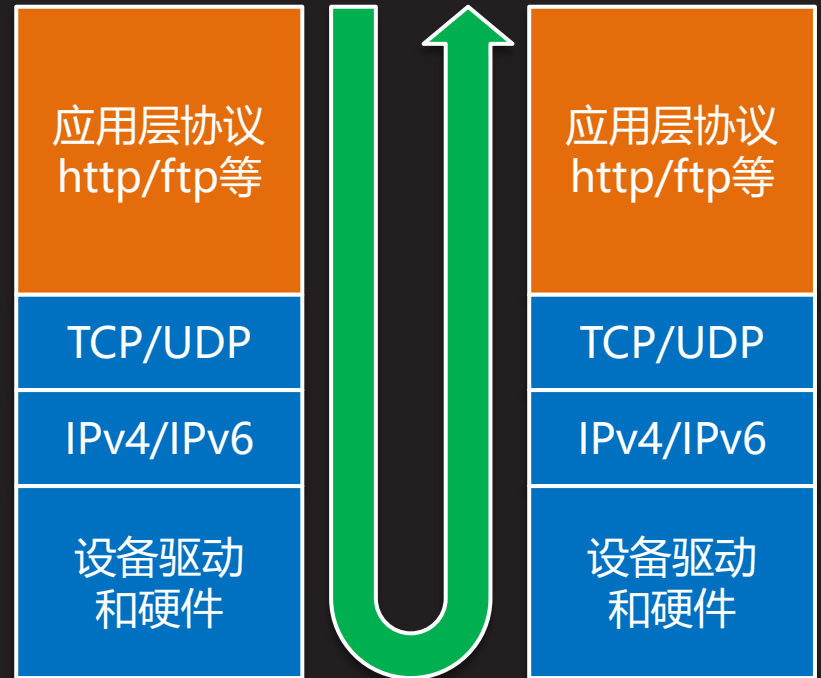


# 消息与地址



# 消息包与消息流

- 应用程序负责组织的通常都是与业务相关的数据内容，而要想把这些数据内容通过网络发送出去，就要将其自上向下地压入协议栈，每经历一个协议层，就会对数据做一层封包，每一层输出的封包都是下一层输入的内容，消息包沿着协议栈的运动形成了消息流
- 当从网络上接收数据时，过程刚好相反，消息包自下向上地流经协议栈，每经历一个协议层，就会对输入的数据解一层封包，经过层层解包以后，应用程序最终得到的将只是与业务相关的数据内容



# IP地址

- 什么是IP地址？
  - IP地址，全称网际协议地址(Internet Protocol Address)，是IP协议提供的一种统一的地址格式，为互联网上的每个网络和每台主机分配一个逻辑地址，借以消除物理地址差异性所带来的影响
- IP地址如何表示？
  - 在计算机内部，IP地址用一个32位无符号整数表示，如：0x01020304。如无特别说明本课程只讨论IPv4的情况
  - 人们更习惯使用点分十进制字符串表示，如：1.2.3.4。字符串形式的从左到右，对应整数形式的从高字节到低字节。注意这里所说的高低指的是数位高低而非地址高低



# IP地址 (续1)

- 什么是IP地址分级？
  - A级地址：以0为首的8位网络地址+24位本地地址
  - B级地址：以10为首的16位网络地址+16位本地地址
  - C级地址：以110为首的24位网络地址+8为本地地址
  - D级地址：以1110为首的32位多播地址
  - 例如：某台计算机的IP地址：192.168.182.48，写成整数形式：11000000 10101000 10110110 00110000，C级地址，网络地址：192.168.182.0，本地地址：48
- 什么是子网掩码？
  - IP地址 & 子网掩码 = 网络地址  
 $192.168.182.48 \& 255.255.255.0 = 192.168.182.0$



# 套接字

---

套接字

基本概念

编程接口

通信模式

绑定与连接

# 基本概念





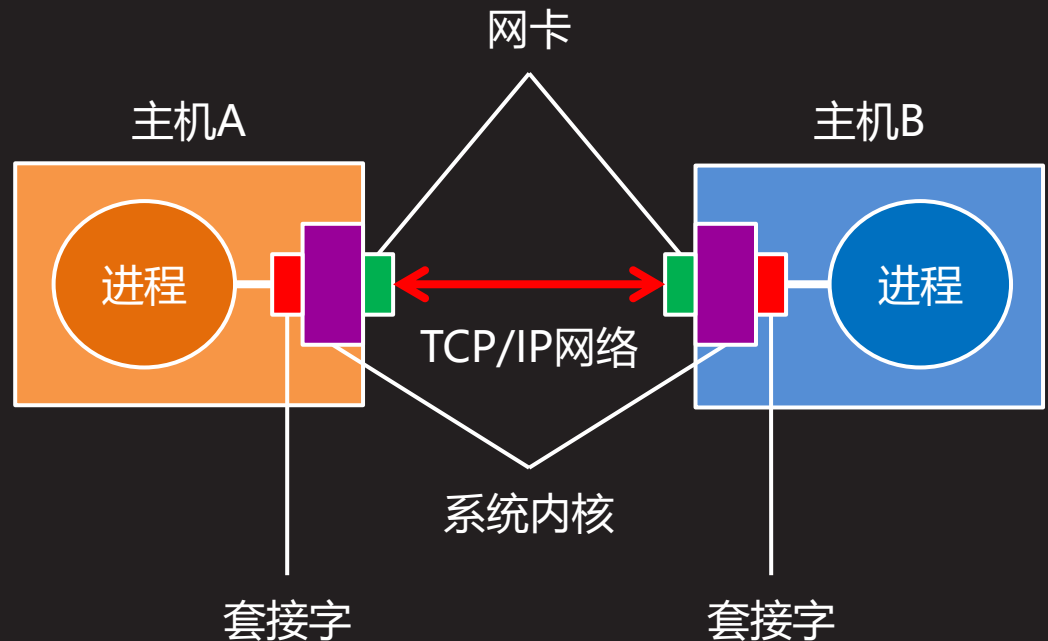


- 什么是伯克利套接字(Berkeley Socket)?
  - 美国加利福尼亚大学伯克利分校(University of California-Berkeley, UC Berkeley)于1983年发布4.2 BSD Unix系统。其中包含一套用C语言编写的应用程序开发库。该库既可用于在同一台计算机上实现进程间通信,也可用于在不同计算机上实现网络通信。当时的Unix还受AT&T的专利保护,因此直到1989年,伯克利大学才能自由发布他们的操作系统和网络库,而后者即被称为伯克利套接字应用编程接口(Berkeley Socket APIs)
  - 伯克利套接字接口的实现完全基于TCP/IP协议,因此它是构建一切互联网应用的基石。几乎所有现代操作系统都或多或少有一些源自伯克利套接字接口的实现。它已成为应用程序连接互联网的标准接口



# 编程接口 (续1)

- 什么是套接字?
  - 套接字(socket)的本意是指电源插座, 这里将其引申为一个基于TCP/IP协议可实现基本网络通信功能的逻辑对象
  - 机器与机器的通信, 或者进程与进程的通信, 在这里都可以被抽象地看作是套接字与套接字的通信
  - 应用程序编写者无需了解网络协议的任何细节, 更无需知晓系统内核和网络设备的运作机制, 只要把想发送的数据写入套接字, 或从套接字中读取想接收的数据即可



# 编程接口 (续2)

- 什么是套接字？
  - 从这个意义上讲，套接字就相当于一个文件描述符，而网络就是一种特殊的文件，面向网络的编程与面向文件的编程已没有分别，而这恰恰是Unix系统一切皆文件思想的又一例证

- 什么是套接字的异构性？

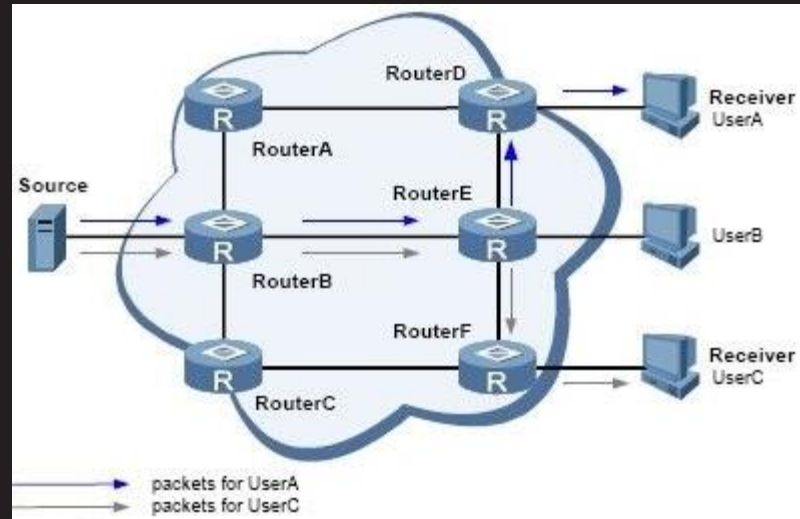
- 如前所述，套接字是对ISO/OSI网络协议模型中传输层及其以下诸层的逻辑抽象，是对TCP/IP网络通信协议的高级



封装，因此无论所依赖的是什么硬件，所运行的什么操作系统，所使用的是什么编程语言，只要是基于套接字构建的应用程序，只要是在互联网环境中通信，就不会存在任何障碍

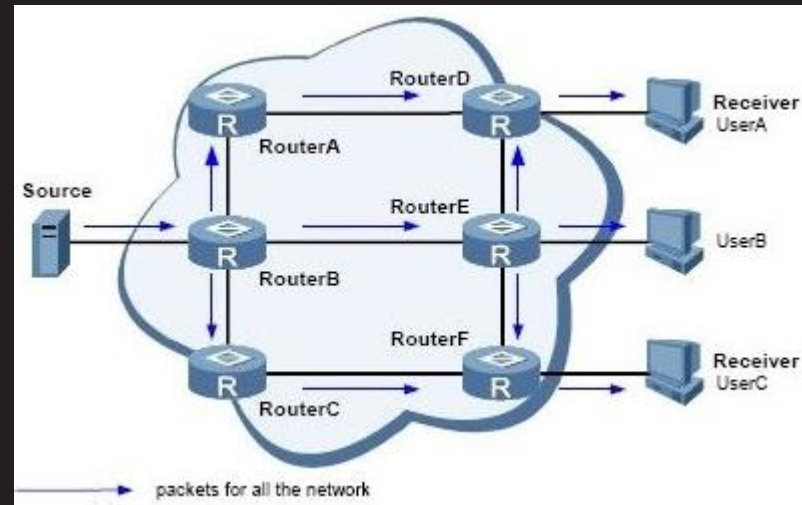
# 通信模式

- 单播模式
  - 每个数据包发往单个目的主机，目的地址指明单个接收者
  - 服务器可以及时响应客户机的请求
  - 服务器可以针对不同客户的不同请求提供个性化的服务
  - 网络中传输的信息量与请求该信息的用户量成正比，当请求该信息的用户量较大时，网络中将出现多份内容相同的信息流，此时带宽就成了限制传输质量的瓶颈



# 通信模式 (续1)

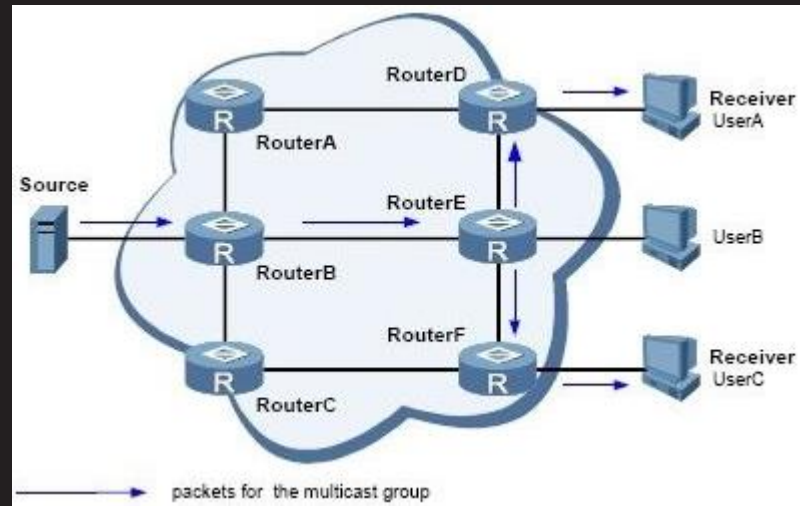
- 广播模式
  - 一台主机向网上的所有其它主机发送数据
  - 无需路径选择, 设备简单, 维护方便, 成本低廉
  - 服务器不用向每个客户机单独发送数据, 流量负载极低
  - 无法针对具体客户的具体要求, 及时提供个性化的服务
  - 网络无条件地复制和转发每一台主机产生的信息, 所有的主机可以收到所有的信息, 而不管是否需要, 网络资源利用率低, 带宽浪费严重
  - 禁止广播包穿越路由器, 防止在更大范围内泛滥



# 通信模式 (续2)

- 多播模式

- 网络中的主机可以向路由器请求加入或退出某个组，路由器和交换机有选择地复制和转发数据，只将组内数据转发给那些加入组的主机
- 需要相同信息的客户机只要加入同一个组即可共享同一份数据，降低了服务器和网络的流量负载
- 既能一次将数据传输给多个有需要的主机，又能保证不影响其它不必要的主机
- 多播包可以穿越路由器，并在穿越中逐渐衰减
- 缺乏纠错机制，丢包错包在所难免

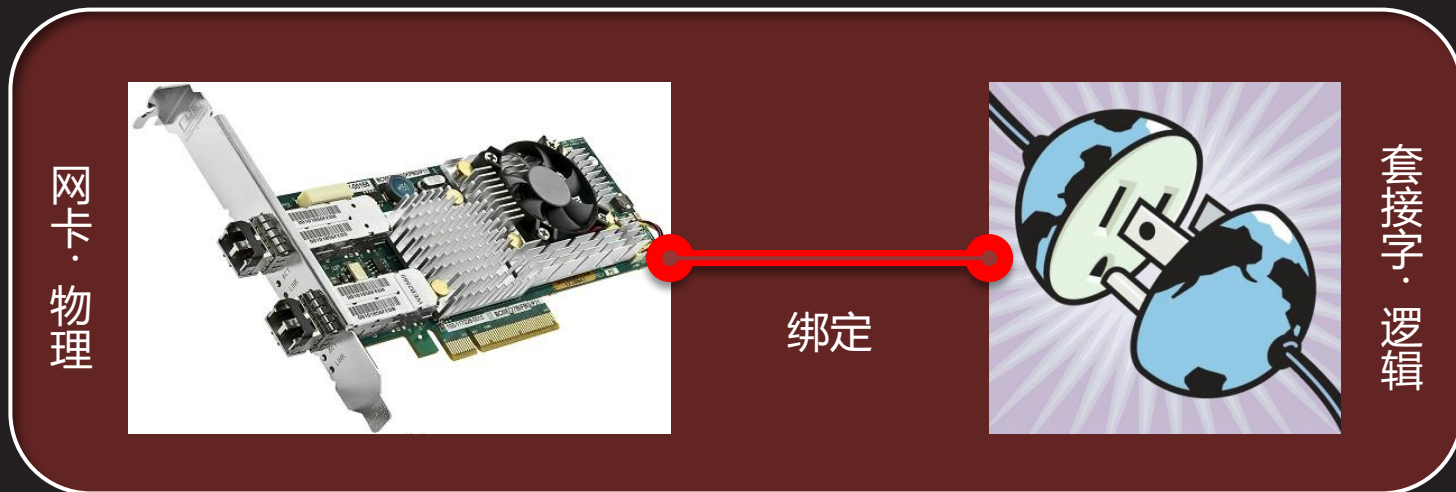




# 绑定与连接

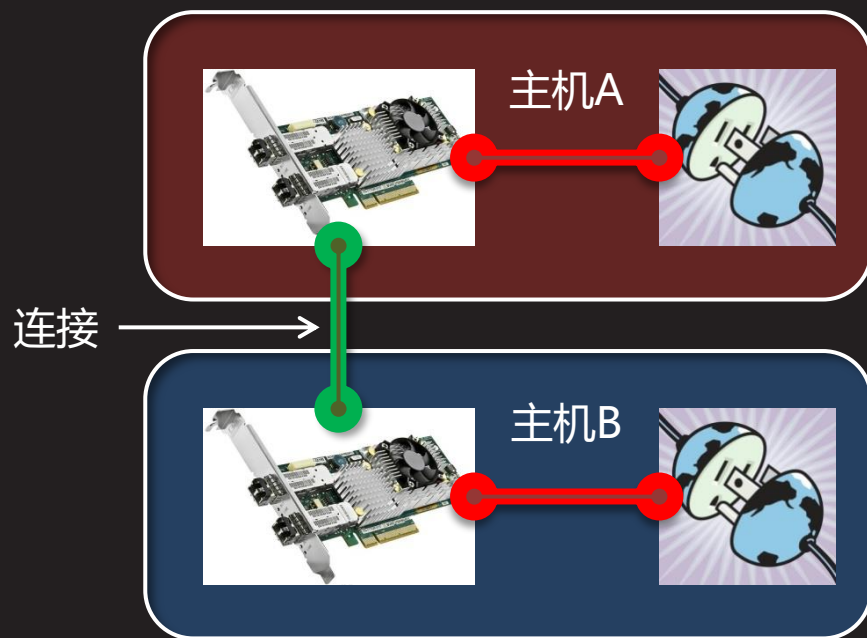
- 如前所述，套接字是一个提供给程序员使用的逻辑对象，它表示对ISO/OSI网络协议模型中传输层及其以下诸层的抽象。但真正发送和接收数据的毕竟是那些实实在在的物理设备。这就需要在物理设备和逻辑对象之间建立一种关联，使后续所有针对这个逻辑对象的操作，最终都能够反映到实际的物理设备上。建立这种关联关系的过程就叫做绑定

知识讲解



# 绑定与连接 (续1)

- 绑定只是把套接字对象和一个代表自己的物理设备关联起来。但为了实现通信还需要把自己的物理设备与对方的物理设备关联起来。只有这样才能建立起一种以物理设备为媒介的，跨越不同进程甚至机器的，多个套接字对象之间的联系。建立这种联系的过程就叫做连接





# 总结和答疑

