

11_网络编程（Socket编程）

CONTENTS

- 基本概念
 - 计算机网络
 - 网络通信协议
 - TCP协议和UDP协议
 - IP地址和端口号
 - URL统一资源定位符
 - Socket套接字
- TCP编程
 - 实现用户登录功能
- UDP编程
 - 实现在线客服系统



本章技能点列表

技能点名称	难易程度	认知程度	重要程度
计算机网络	易	了解	*
ISO/OSI模型和TCP/IP协议栈	易	记忆	*
TCP与UDP区别和联系	中	理解	**
URL统一资源定位符	易	理解	**
IP地址和端口号	中	理解	***
Socket套接字	易	理解	**
TCP编程	难	应用	**
UDP编程	中	应用	**



计算机网络

- 计算机网络定义

- 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

- 计算机网络定义分析

- 主干：计算机网络是计算机系统
- 网络功能：资源共享 信息传递
- 网络组成：
 - 网络硬件：计算机 外部设备 通信线路 （连接）
 - 网络软件：网络操作系统 网络管理软件 网络通信协议 （管理和协调）

- 计算机网络分类1（按照规模）

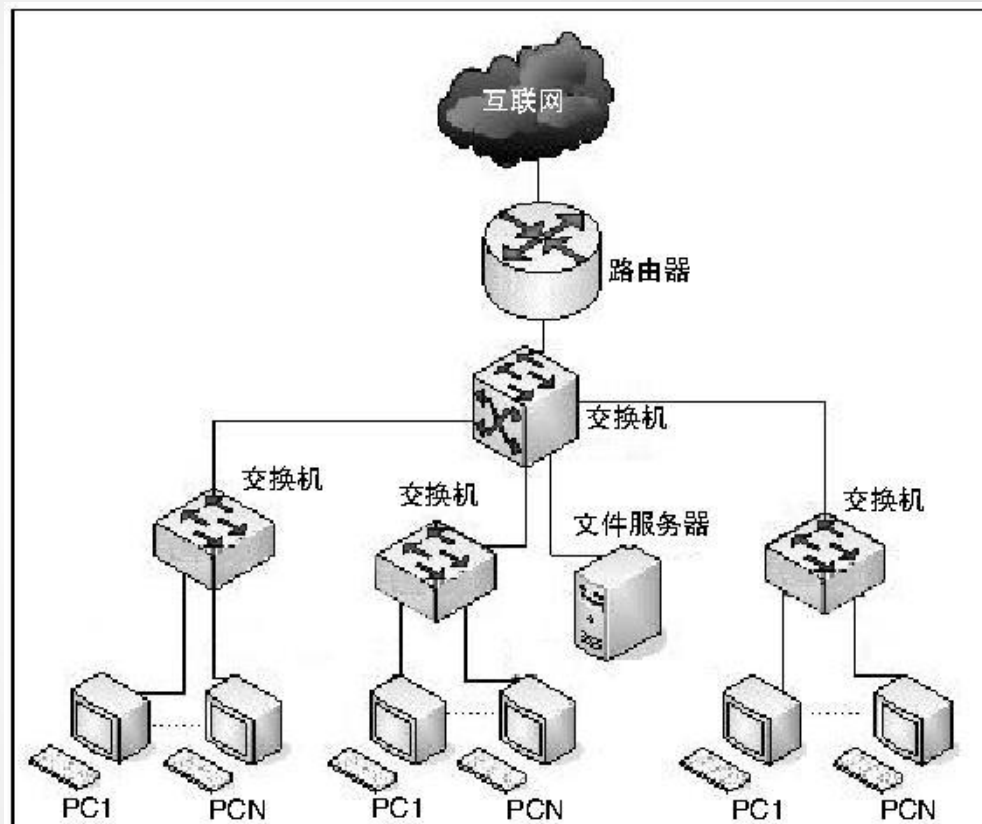
- 局域网LAN
- 城域网MAN
- 广域网WAN

- 计算机网络分类2（传输介质）

- 同轴电缆网络
- 双绞线网络
- 光纤网络
- 卫星网络

- 计算机网络分类3（拓扑结构）

- 星形网络
- 总线网络
- 环状网络





网络通信协议

- 网络通信协议

- 计算机网络中实现通信必须有一些约定即通信协议，对速率、传输代码、代码结构、传输控制步骤、出错控制等制定标准
- 好比公路交通规则，学生守则

- 问题：网络协议太复杂

- 计算机网络通信涉及内容很多，比如指定源地址和目标地址，加密解密，压缩解压缩，差错控制，流量控制，路由控制，如何实现如此复杂的网络协议呢？

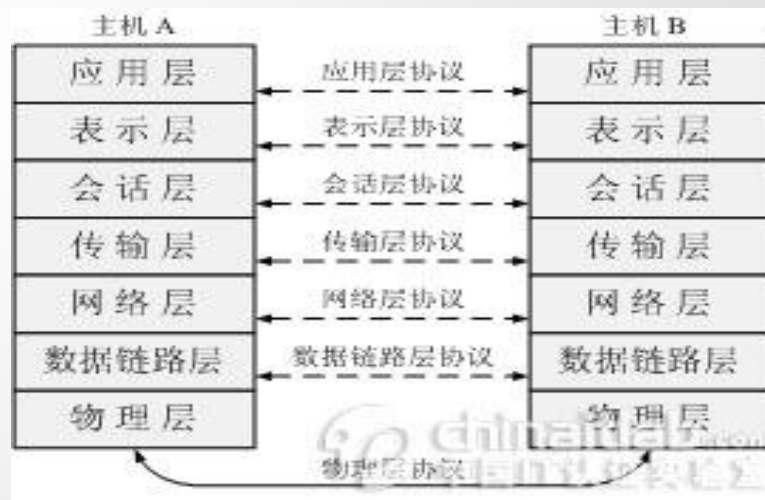
- 解决方案：分层

- 由于结点之间联系很复杂，在制定协议时，把复杂成份分解成一些简单的成份，再将它们复合起来。最常用的复合方式是层次方式，即同层间可以通信、上一层可以调用下一层，而与再下一层不发生关系。



网络通信协议

- 网络通信协议的分层
 - 名义上标准：ISO/OSI参考模型
 - 事实上标准：TCP/IP协议栈（Internet使用的协议）

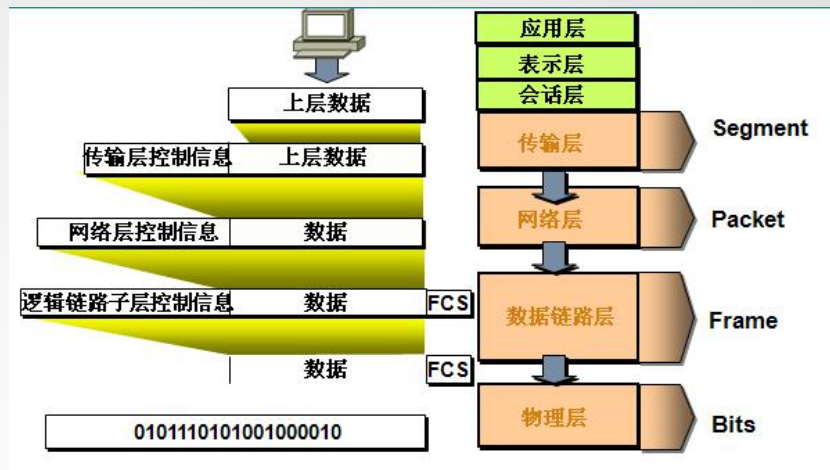




网络通信协议

- 数据封装与拆封

- 1、封装（发送数据）



- 2、拆封（接收数据）





网络通信协议

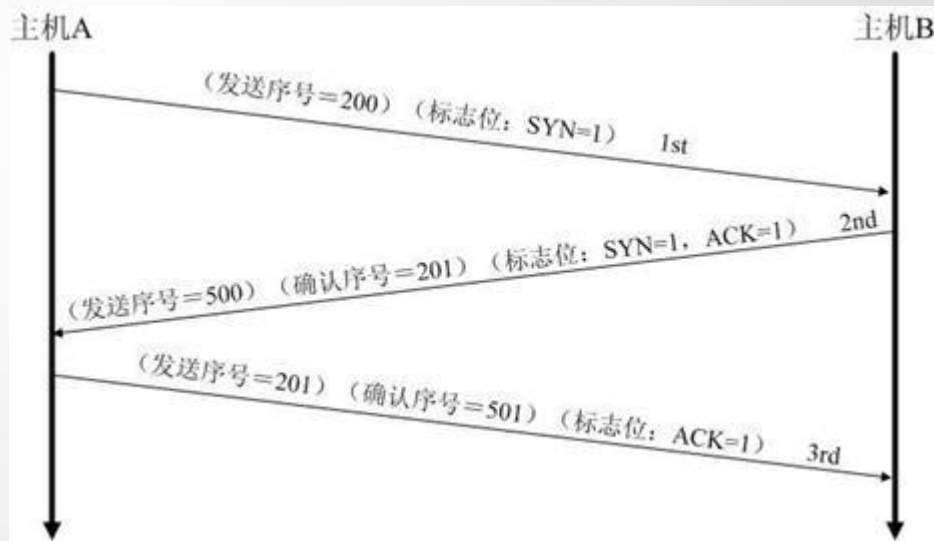
- TCP/IP协议栈
 - 网络层主要协议IP协议
 - 传输层主要协议TCP和UDP

OSI		TCP/IP协议集
应用层	应用层	Telnet, FTP, SMTP, DNS, HTTP 以及其他应用协议
表示层		
会话层		
传输层	传输层	TCP, UDP
网络层	网络层	IP, ARP, RARP, ICMP
数据链路层	网络接口	各种通信网络接口（以太网等） （物理网络）
物理层		



TCP协议和UDP协议

- TCP(transfer control protocol)
 - 一种面向连接（连接导向）的、可靠的、基于字节流的运输层（Transport layer）通信协议。
- 特点
 - 面向连接
 - 点到点的通信
 - 高可靠性:三次握手
 - 占用系统资源多、效率低
- 生活案例
 - 打电话
- 应用案例：
 - HTTP FTP TELNET SMTP





TCP协议和UDP协议

- UDP(User Datagram Protocol)
 - 一种无连接的传输层协议，提供面向事务的简单不可靠信息传送服务
- 特点：
 - 非面向连接，传输不可靠，可能丢失
 - 发送不管对方是否准备好，接收方收到也不确认
 - 可以广播发送
 - 非常简单的协议，开销小
- 生活案例：
 - 发送短信 发电报
- 应用案例：
 - DNS SNMP



IP地址与端口

- IP地址
 - 用来标志网络中的一个通信实体的地址。通信实体可以是计算机，路由器等。
- IP地址分类
 - IPV4：32位地址，以点分十进制表示，如192.168.0.1
 - IPV6：128位（16个字节）写成8个16位的无符号整数，每个整数用四个十六进制位表示，数之间用冒号（：）分开，如：3ffe:3201:1401:1280:c8ff:fe4d:db39:1984
- 特殊的IP地址
 - 127.0.0.1 本机地址
 - 192.168.0.0--192.168.255.255私有地址，属于非注册地址，专门为组织机构内部使用。



IP地址与端口

- 端口:port
 - IP地址用来标志一台计算机，但是一台计算机上可能提供多种应用程序，使用端口来区分这些应用程序。
 - 端口是虚拟的概念，并不是说在主机上真的有若干个端口。通过端口，可以在一个主机上运行多个网络应用程序。
 - 端口范围0---65535,16位整数
- 理解IP和端口的关系
 - IP地址好比每个人的地址（门牌号），端口好比是房间号。必须同时指定IP地址和端口号才能够正确的发送数据
 - IP地址好比为电话号码，而端口号就好比为分机号。
- 端口分类
 - 公认端口 0—1023 比如80端口分配给WWW，21端口分配给FTP
 - 注册端口 1024—49151 分配给用户进程或应用程序
 - 动态/私有端口 49152--65535



IP地址与端口

- InetAddress类
 - 封装计算机的ip地址,没有端口

//使用getLocalHost方法创建InetAddress对象

```
InetAddress addr = InetAddress.getLocalHost();
```

```
System.out.println(addr.getHostAddress()); //返回: 192.168.1.110
```

```
System.out.println(addr.getHostName()); //输出计算机名
```

//根据域名得到InetAddress对象

```
addr = InetAddress.getByName("www.163.com");
```

```
System.out.println(addr.getHostAddress()); //返回 163服务器的ip:61.135.253.15
```

```
System.out.println(addr.getHostName()); //输出: www.163.com
```

//根据ip得到InetAddress对象

```
addr = InetAddress.getByName("61.135.253.15");
```

```
System.out.println(addr.getHostAddress()); //返回 163服务器的ip:61.135.253.15
```

```
System.out.println(addr.getHostName()); //输出ip而不是域名。如果这个IP地址不存在或DNS  
服务器不允许进行IP地址和域名的映射, getHostName方法就直接返回这个IP地址。
```



IP地址与端口

- InetAddress
 - 包含端口，用于socket通信的

//包含端口

```
InetSocketAddress socketAddress = new InetSocketAddress("127.0.0.1",8080);  
InetSocketAddress socketAddress2 = new InetSocketAddress("localhost",9000);  
System.out.println(socketAddress.getHostName());  
System.out.println(socketAddress2.getAddress());
```

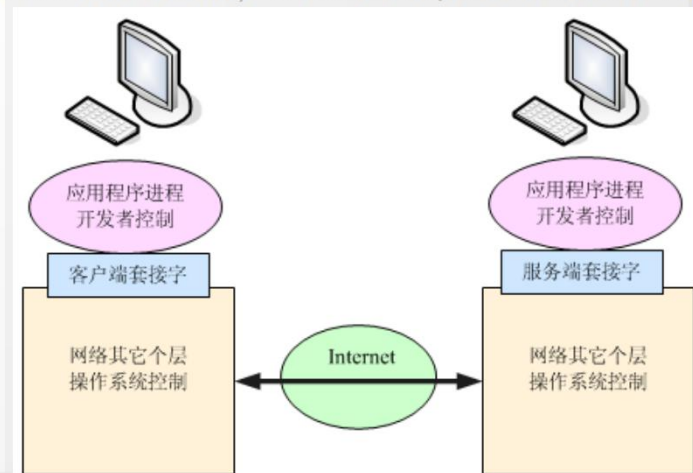
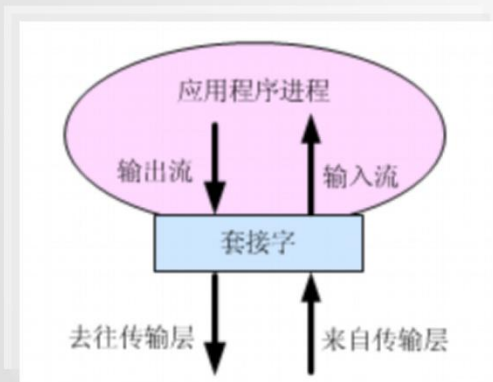


- URL (Uniform Resource Locator)
 - 统一资源定位符,由4部分组成: 协议、存放资源的主机域名、端口号和资源文件名。
- URL是指向互联网“资源”的指针。
 - 资源可以是简单的文件或目录,也可以是对更为复杂的对象的引用,例如对数据库或搜索引擎的查询。

```
URL u = new URL("http://www.baidu.com:80/index.html#aa?cansu=bjsxt");
System.out.println("获取与此url关联的协议的默认端口: "+u.getDefaultPort());
System.out.println("getFile: "+u.getFile()); //端口号后面的内容
System.out.println("主机名: "+u.getHost()); //www.google.cn
System.out.println("路径: "+u.getPath()); //端口号后, 参数前的内容
System.out.println("端口: "+u.getPort()); //存在返回80. 否则返回-1
System.out.println("协议: "+u.getProtocol());
System.out.println("参数部分: "+u.getQuery());
System.out.println("锚点: "+u.getRef());
URL u = new URL("http://www.abc.com/aa/");
URL u2 = new URL(u, "2.html"); //相对路径构建url对象
System.out.println(u2.toString()); //http://www.abc.com/aa/2.html
```


- 套接字Socket

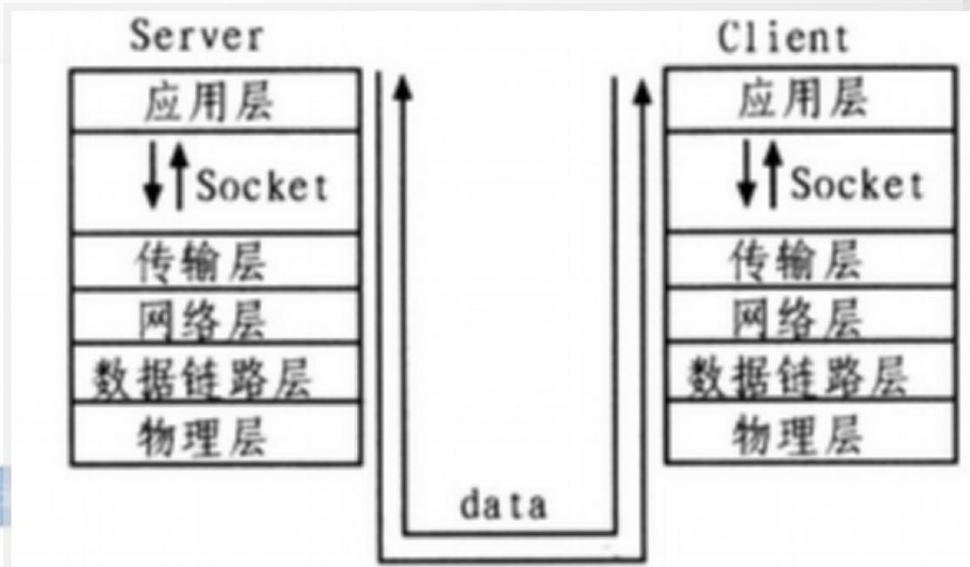
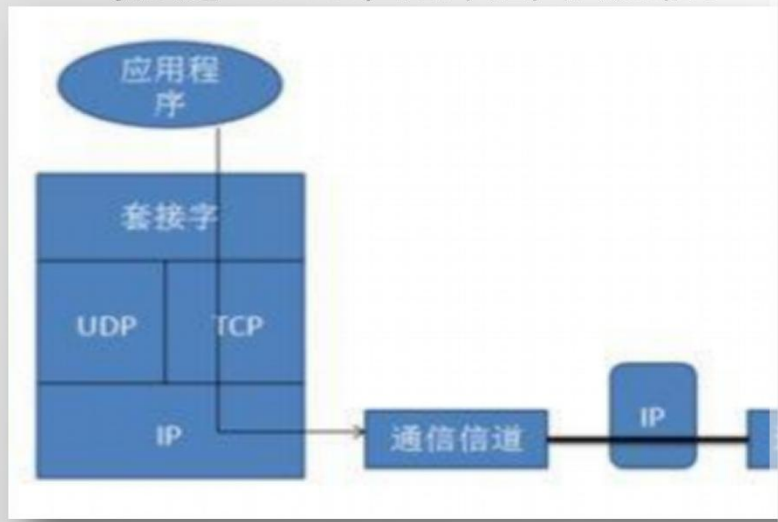
- 我们开发的网络应用程序位于应用层，TCP和UDP属于传输层协议，在应用层如何使用传输层的服务呢？在应用层和传输层之间，则是使用套接字来进行分离。
- 套接字就像是传输层为应用层开的一个小口，应用程序通过这个小口向远程发送数据，或者接收远程发来的数据；而这个小口以内，也就是数据进入这个口之后，或者数据从这个口出来之前，是不知道也不需要知道的，也不会关心它如何传输，这属于网络其它层次的工作。





套接字Socket

- Socket实际是网络传输层供给应用层的编程接口。传输层则在网络层的基础上提供进程到进程间的逻辑通道，而应用层的进程则利用传输层向另一台主机的某一进程通信。Socket就是应用层与传输层之间的桥梁
- 使用Socket编程可以开发客户机和服务器应用程序，可以在本地网络上进行通信，也可通过Internet在全球范围内通信。





套接字Socket

- 生活案例1:
 - 如果你想写封邮件发给远方的朋友，如何写信、将信打包，属于应用层。信怎么写，怎么打包完全由我们做主；
 - 而当我们把信投入邮筒时，邮筒的那个口就是套接字，在进入套接字之后，就是传输层、网络层等（邮局、公路交管或者航线等）其它层次的工作了。我们从来不会去关心信是如何从西安发往北京的，我们只知道写好了投入邮筒就OK了。





套接字Socket

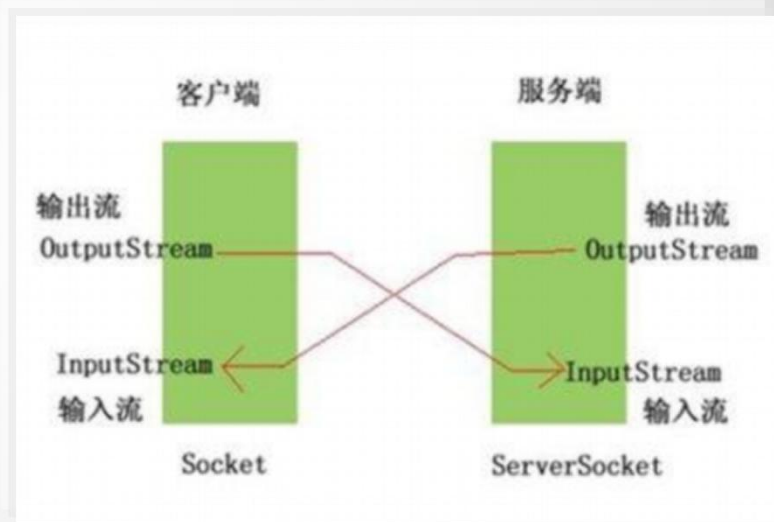
- 生活案例2:
 - 如果你想发货给国外，你只要把货物放入集装箱，然后交给码头就可以了。发送什么货物，货物如何打包，完全有你做主。
 - 码头就是套接字，剩下的事情就交给港口和货运公司处理就行了，具体细节我们无需了解。





TCP编程

- 需求：完成网络登录功能：
 - 用户输入用户名密码，服务器给出登录成功或失败的提示
- 分析
 - 使用基于TCP协议的Socket网络编程实现
 - TCP协议基于请求-响应模式
 - 在网络通讯中，第一次主动发起通讯的程序被称作客户端(Client)程序
 - 第一次通讯中等待连接的程序被称作服务器端(Server)程序
 - 利用IO流实现数据的传输



- TCP通信原理（详细步骤）

- 服务器创建ServerSocket，在指定端口监听并处理请求
- 客户端创建Socket，向服务器发送请求





- 网络登录功能分解

- 单向：客户端向服务器端发送字符串，服务器获取字符串并输出
- 双向：服务器端给出客户端反馈，客户端得到反馈并输出
- 对象：客户端向服务器端发送User对象，服务器端获取对象并输出
- 多线程：服务器接收多个客户端的请求，并给出反馈
每个客户请求开启一个线程



• 网络登录功能总结

- 服务器创建ServerSocket，在指定端口监听并处理请求；
- ServletSocket通过accept() 接收用户请求并返回对应的Socket，否则一种处于监听等待状态，线程也被阻塞
- 客户端创建Socket，需要指定服务器的ip和端口号，向服务器发送和接收响应
- 客户端发送数据需要输出流(写)，客户端获取反馈数据需要输入流(读)
- 服务端反馈数据需要输出流(写)，服务端获取请求数据需要输入流(读)
- 一旦使用ServerSocket和Socket建立了网络连接后，网络通信和普通IO流操作并没有太大区别
- 网络通信输出流建议使用DataOutputStream和ObjectOutputStream,与平台无关，输入流相应使用DataInputStream和ObjectInputStream
- 如果是字符串通信也可以使用BufferedReader和PrintWriter，简单方便



UDP编程

- 需求：完成在线咨询功能：
 - 客户和咨询师在线交流
- 分析
 - 使用基于UDP协议的Socket网络编程实现
 - 不需要利用IO流实现数据的传输
 - 每个数据发送单元被统一封装成数据包的方式，发送方将数据包发送到网络中，数据包在网络中寻找他的目的地。
- UDP基本概念
 - DatagramSocket：用于发送或接收数据包
 - DatagramPacket：数据包



作业

- TCP/IP协议栈
- IP地址和端口
- TCP和UDP的联系区别
- TCP编程的主要步骤
- UDP编程的主要步骤