Java技术

第九章　网络通信

路　强

luqiang@hfut.edu.cn

合肥工业大学计算机与信息学院

几个时间节点

- 6月20日之前交作业4
- 6月30日晚上实验验收
- 7月05日之前交作业5和实验报告

（群文件中报告新模板、四个实验一个word）

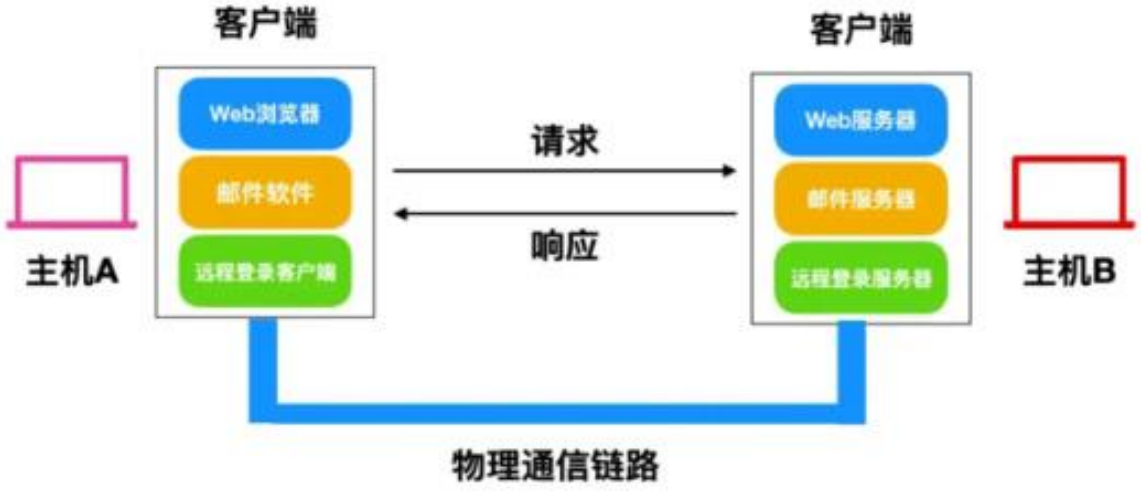
- 7月09日结课机考

本章学习提示

- **本章学习Java语言在网络通讯方面的应用**
  - **网络基本概念**
  - **java.net包中类的使用**
  - **简单的socket程序设计**

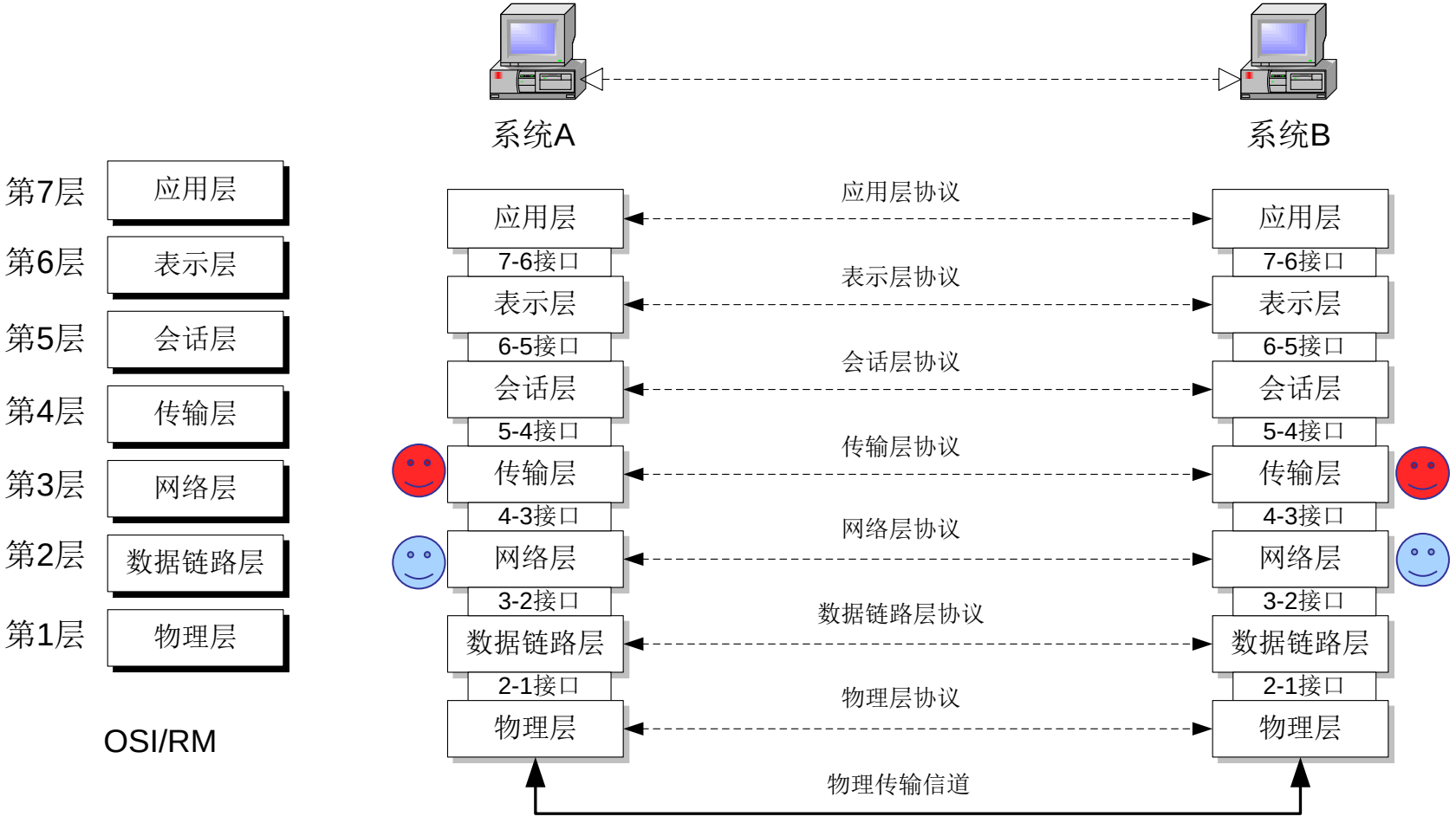
目　录

网络基础


客户端
客户端
Web浏览器
邮件软件
远程登录客户端
请求
响应
Web服务器
邮件服务器
远程登录服务器
主机A
主机B
物理通信链路

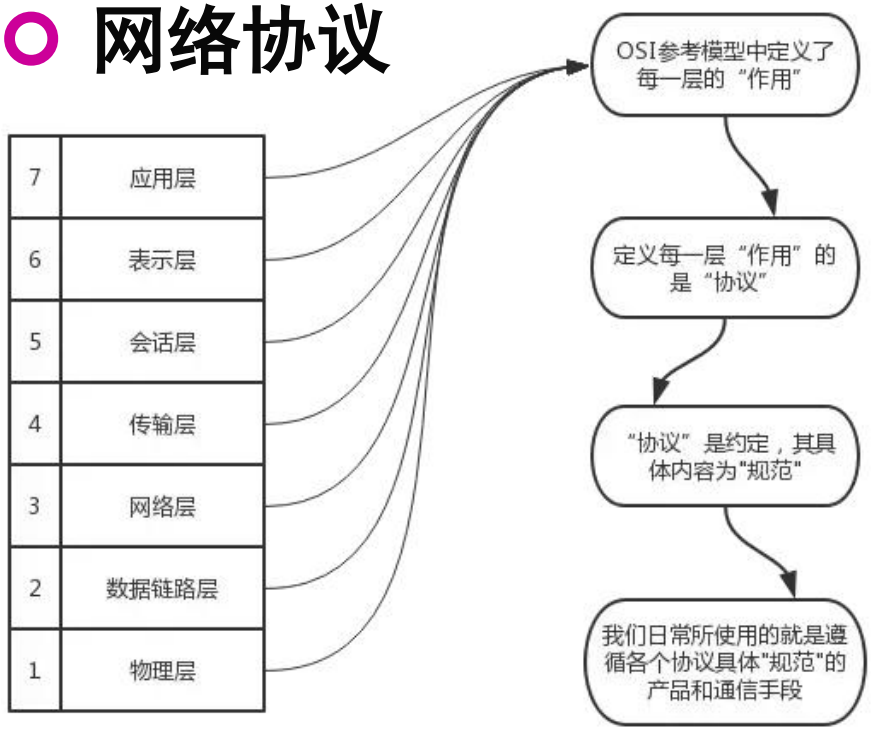
计算机网络的体系结构

OSI 体系结构



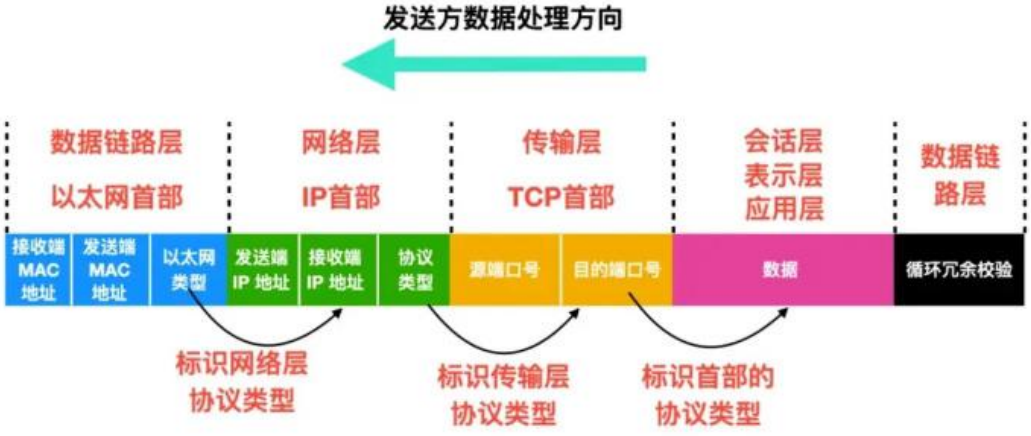
网络基础

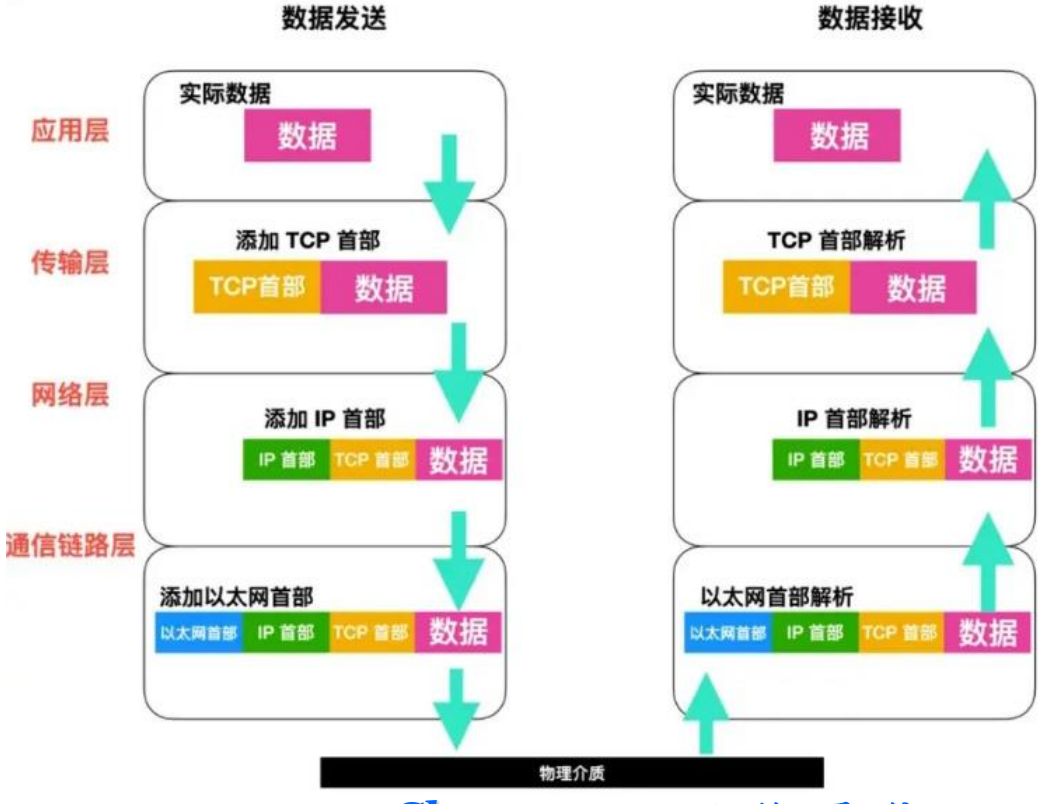
网络协议



OSI七层模型	TCP/IP概念层模型	功能	TCP/IP协议族
应用层	应用层	文件传输，电子邮件，文件服务，虚拟终端	TFTP，HTTP，SNMP，FTP，SMTP，DNS，Telnet
表示层		数据格式化，代码转换，数据加密	没有协议
会话层		解除或建立与别的接点的联系	没有协议
传输层	传输层	提供端对端的接口	TCP，UDP
网络层	网络层	为数据包选择路由	IP，ICMP，RIP，OSPF，BGP，IGMP
数据链路层	链路层	传输有地址的帧以及错误检测功能	SLIP，CSLIP，PPP，ARP，RARP，MTU
物理层		以二进制数据形式在物理媒体上传输数据	ISO2110，IEEE802，IEEE802.2

数据包、数据处理流程


发送方数据处理方向
数据链路层
以太网首部
网络层
IP首部
传输层
TCP首部
会话层
表示层
应用层
数据链路层
接收端 MAC 地址
发送端 MAC 地址
以太网类型
发送端 IP 地址
接收端 IP 地址
协议类型
源端口号
目的端口号
数据
循环冗余校验
标识网络层协议类型
标识传输层协议类型
标识首部的协议类型
接收方数据解析方向


数据发送
数据接收
应用层
传输层
网络层
通信链路层
实际数据
数据
添加 TCP 首部
TCP首部
添加 IP 首部
IP 首部
TCP 首部
添加以太网首部
以太网首部
TCP 首部解析
IP 首部解析
以太网首部解析
物理介质

网络基础

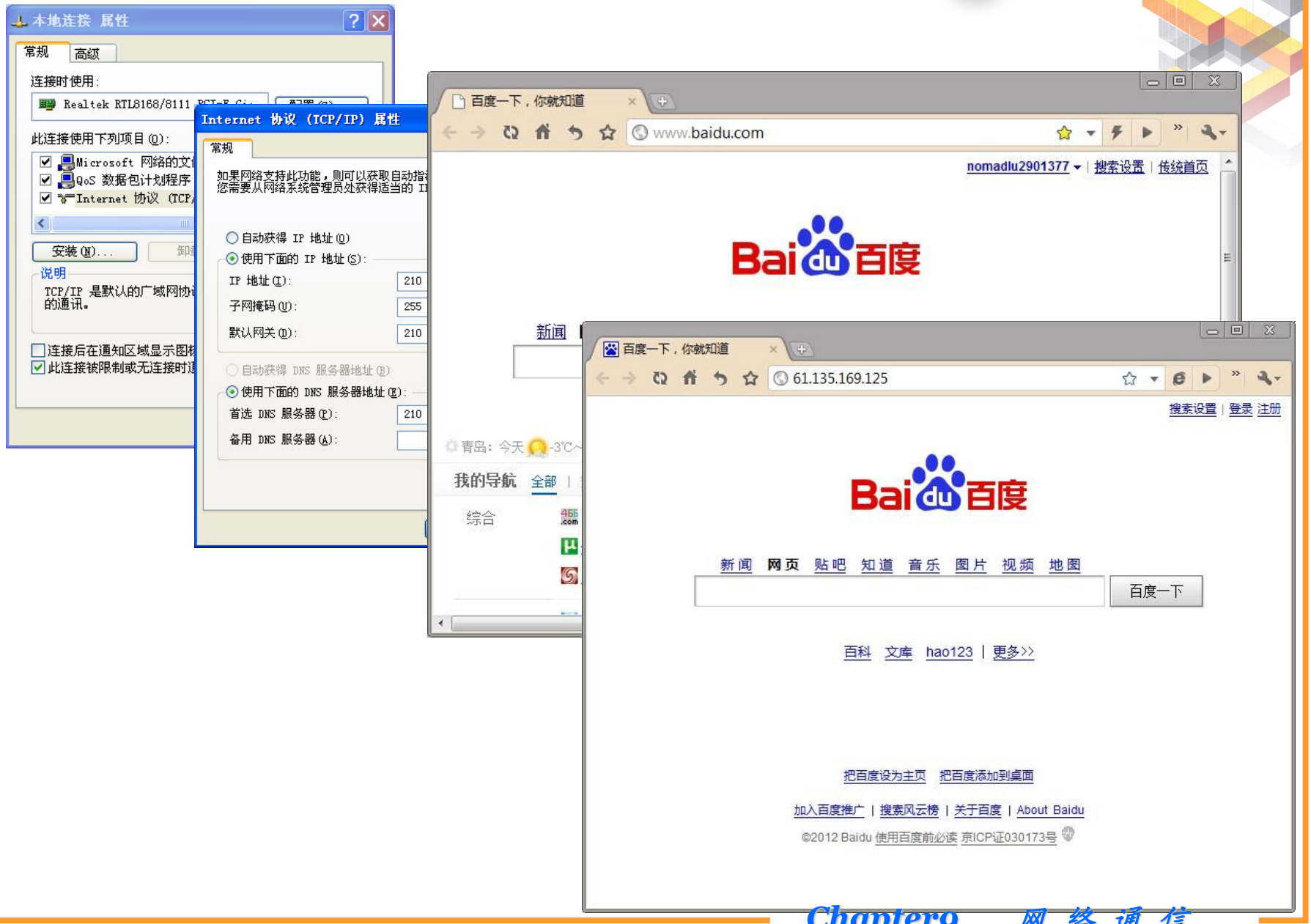
- 互联网上的计算机之间的通讯是通过TCP (Transport Control Protocol) 或UDP (User Datagram Protocol)协议。
- 下图所示：

应用协议 HTTP, SMTP, FTP, TELNET, SNMP
传输协议 TCP, UDP
网络协议 IP, ICMP, ARP
链路 链路层协议, 设备驱动

TCP协议

- **TCP**是Internet的基础和核心，**一个基于连接的协议**，提供两台计算机之间的**可靠的数据流**。
- 从网络通信的角度看，Internet是一个用TCP/IP协议把各个国家、各个部门、各种机构的内部网络连接起来的超级数据通信网。
- 为了支持点对点通信，每个节点要有一个像电话号码一样的唯一的地址称为**IP地址**，它是一个32位的二进制数。由于不便于记忆将此地址表示成4个十进制数，各取0-255的值。每个值之间用点“.”分隔。
- 为了进一步方便使用，人们给每个节点都起一个名字，把名字与IP地址建立一个对应关系，这就是域名系统**DNS**，凡域名空间有定义的域名都可以有效地转换成IP地址。

IP地址 和 域名之例


本地连接 属性
常规
高级
连接时使用:
Realtek RTL8168/8111
此连接使用下列项目(O):
Microsoft 网络的文
QoS 数据包计划程序
Internet 协议 (TCP
安装(N)...
说明
TCP/IP 是默认的广域网协
的通讯。
连接后在通知区域显示图
此连接被限制或无连接时
Internet 协议 (TCP/IP) 属性
常规
如果网络支持此功能，则可以获取自动指
您需要从网络系统管理员处获得适当的 I
自动获得 IP 地址(O)
使用下面的 IP 地址(S):
IP 地址(I):
210
子网掩码(U):
255
默认网关(D):
210
自动获得 DNS 服务器地址(B)
使用下面的 DNS 服务器地址(E):
首选 DNS 服务器(P):
210
备用 DNS 服务器(A):
百度一下，你就知道
www.baidu.com
nomadlu2901377
搜索设置
传统首页
Bai du 百度
新闻
青岛: 今天 -3℃~
我的导航
全部
综合
百度一下，你就知道
61.135.169.125
搜索设置
登录
注册
Bai du 百度
新闻
网页
贴吧
知道
音乐
图片
视频
地图
百度一下
百科
文库
hao123
更多>>
把百度设为主页
把百度添加到桌面
加入百度推广
搜索风云榜
关于百度
About Baidu
©2012 Baidu 使用百度前必读 京ICP证030173号

UDP协议

- 定义：

UDP是从一台计算机向另一台计算机发送称为数据报的独立数据包的协议，该协议并不保证数据报是否能正确地到达目的地。它是一个非面向连接的协议。

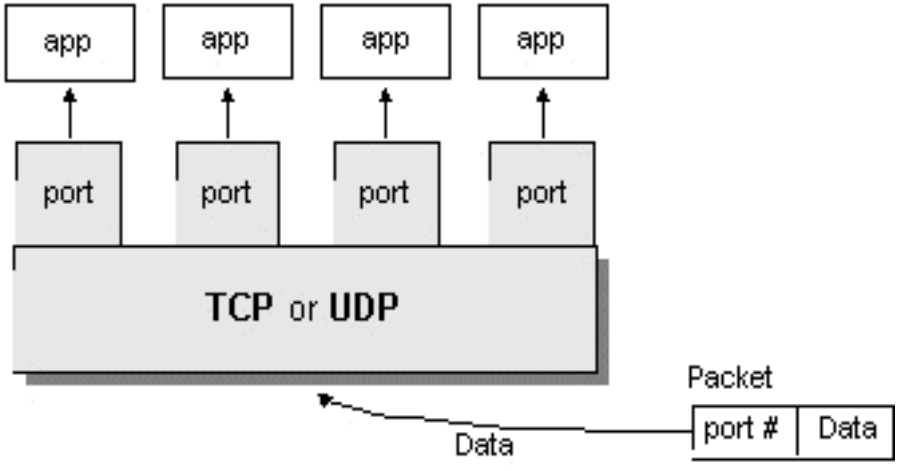
- QQ等IM软件，一般均采用UDP协议

客户机与服务器

- Internet 采用客户机/服务器方式访问资源，当用户在共享某个Internet资源时，有两个独立的程序协同服务。这两个程序运行在不同的计算机上，通常把提供资源的计算机叫做**服务器**，把使用资源的计算机叫做**客户机**。
- 由于Internet上用户往往不知道究竟是哪台计算机提供了资源，**客户机、服务器**指的是软件，即**客户程序**和**服务程序**
- 当用户使用Internet功能时，首先启动客户机，通过有关命令告知服务器进行连接以完成某种操作，而服务器则按照此要求提供相应的服务。

端口(PORT)

- 在互联网上传输的数据都包含有用来识别目的地的IP地址和端口号。IP地址用来标识网络上的计算机，而端口号用来指明该计算机上的应用程序。
- 端口号范围为0-65535, 其中低于1024的端口号保留给那些已定义的服务,用户不能使用。如FTP为21，TELNET 为23，HTTP为80。



Java与网络

- Java语言取得成功的领域之一就是网络

  (其他语言)数页代码-->(Java)一条语句

- TCP/IP(传输控制协议/网间协议)是internet的主要协议,定义了计算机和外设进行通信所使用的规则(应用层,传输层,网络层,链路层).
- 大多数基于internet的应用程序被看作TCP/IP协议的上一层.

  如：ftp, http, smtp, pop3, telnet, nntp等
- IP地址:TCP/IP网络中的每台计算机都有唯一的地址--IP地址.
- 在Java中,有一个用来存储internet地址的类叫**InetAddress**.

内容详见软件包 java.net

Java与网络2

- Java提供的网络功能有三大类: **URL, Socket, Datagram**.
- **URL**是三大功能中最高级的一种,通过URL Java程序可以直接送出或读入网络上的数据.
- **Socket**是传统网络程序最常用的方式,可以想象为两个不同的程序通过网络的通信信道.
- **Datagram**是更低级的网络传输方式,它把数据的目的纪录在数据包中,然后直接放在网络上.

目　录

URL 编程

- 在WWW上，每一信息资源都有统一且唯一的地址，该地址就叫URL (Uniform Resource Locator)，是WWW的统一资源定位标识符。
- URL由4 部分组成；**资源类型**、存放资源的**主机域名**、**资源文件名**和**端口**。

  例如， http://61.135.169.125:80/img/baidu_sylogo1.gif
  - http　　表示该资源类型是超文本信息
  - 61.135.169.125 (www.baidu.com)　　是百度的主机域名
  - 80　　表示HTTP协议的端口号
  - /img/baidu_sylogo1.gif　　表示资源文件名

HTTP协议默认端口号是80，FTP协议默认端口号是21

URL类及其构造方法

- 要使用URL进行通信，就要使用URL类创建其对象，通过引用URL类定义的方法完成网络通信。
- 创建URL类对象要使用java.net包中提供的java.net.URL类的构造方法。

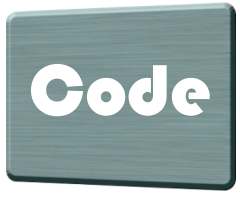
URL类的构造方法	功能说明
public URL(String str)	使用URL字符串创建URL对象
public URL(String protocol, String host,String file)	通过指定协议名、主机名、文件名，端口使用默认值，创建URL对象
public URL(String protocol, String host, String port,String file)	通过指定协议名、主机名、文件名和端口号，创建URL对象
public URL(URL content, String str)	通过在已知的URL路径上增加细节的办法创建URL对象

URL 类的方法

URL 类	功能说明
int getPort()	获得端口号,如果端口没有设置，返回-1
String getProtocol()	获得协议名，如果协议没有设置，返回null
String getHost()	获得主机名，如果主机没有设置，返回null
String getFile()	获得文件名，如果文件没有设置，返回null
Boolean equals(Object obj)	与指定的URL对象obj 进行比较，如果相同返回true,否则返回false
Final InputStreamOpenStream()	获得一个输入流，若获取失败，则抛出一个java.io.Exception异常
String toString()	将此URL对象转换为字符串的形式

URL程序例一


```
1.  //URLReader, 获取网页并按行打印
2.  import java.net.*;
3.  import java.io.*;
4.  public class Network_1{
5.      public static void main(String[] args) throws Exception{
6.          //URL数据-> InputStreamReader对象 -> BufferedReader对象  -> 程序处理
7.          URL hfut = new URL("http://www.baidu.com");
8.          BufferedReader in = new BufferedReader(
9.                                  new InputStreamReader( hfut.openStream() ) );
10.         String inputLine;
11.         //打印输出HTML
12.         while ( (inputLine = in.readLine() ) != null )
13.                     System.out.println(inputLine);
14.         //关闭缓冲区
15.         in.close();
16.     }
17. }
```



URLConnection类

- 使用URL类可以很简单地获得信息，但如果在获取到信息的同时还能向远程的计算机节点**传送信息**，就需要使用URLConnection类。
- 创建URLConnection类的对象，先创建一个URL对象，然后调用该对象的openConnection()方法就会返回一个对应URL地址的URLConnection对象。

内容详见
java.net.URLConnection类

```
//创建URL对象
URL url=new URL("http://www.baidu.com");
//创建链接通道
URLConnection connect=url.openConnection();
```

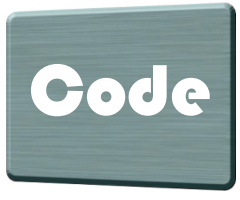
URLConnection类1

- 建立输入/输出流 **URLConnection类**不仅可以使用getInputStream()方法获得URL节点的信息，还可以采用getOutputStream()方法向URL节点处传送信息。
- 在建立URLConnection类对象的同时就已经在本级和URL节点之上建立了一条HTTP通道。
- HTTP是一个**连接协议**，发送信息之前要附加确认双方身份的信息。例:

```
DataInputStream datain=new
        DataInputStream(connect.getInputStream());
```

URL程序例二

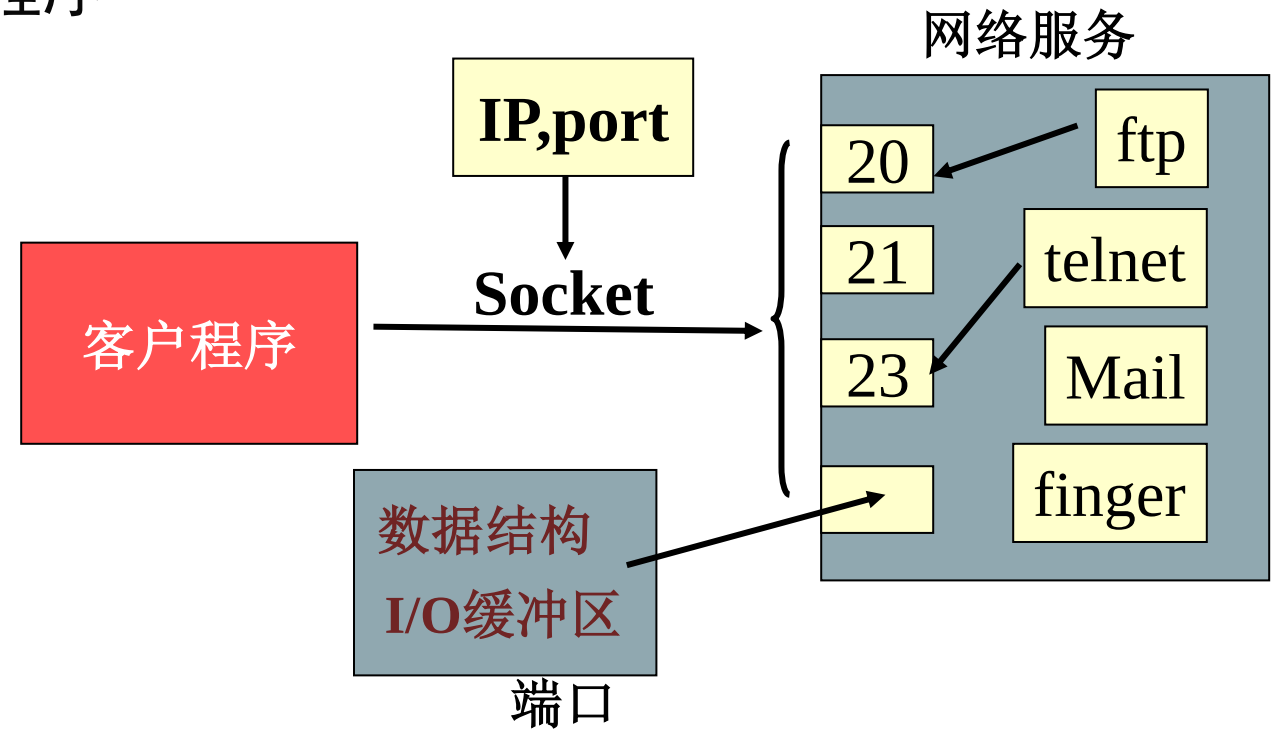

```
1.  //URLConnectionReader
2.  //使用java.net.URLConnection连接到互连网，获取网页并按行打印
3.  import java.net.*;
4.  import java.io.*;
5.  public class  Network_2 {
6.      public static void main(String[] args)  throws Exception {
7.     //URL对象-> URLConnection对象 -> InputStreamReader ->BufferedReader
8.          URL hfut = new URL("http://www.baidu.com");
9.          URLConnection uc = hfut.openConnection();
10.         BufferedReader in = new BufferedReader(
11.                                 new InputStreamReader( uc.getInputStream()) );
12.         String inputLine;
13.         while ((inputLine = in.readLine()) != null)
14.             System.out.println(inputLine);
15.         in.close();
16.     }
17. }
```



目　录

Socket网络编程

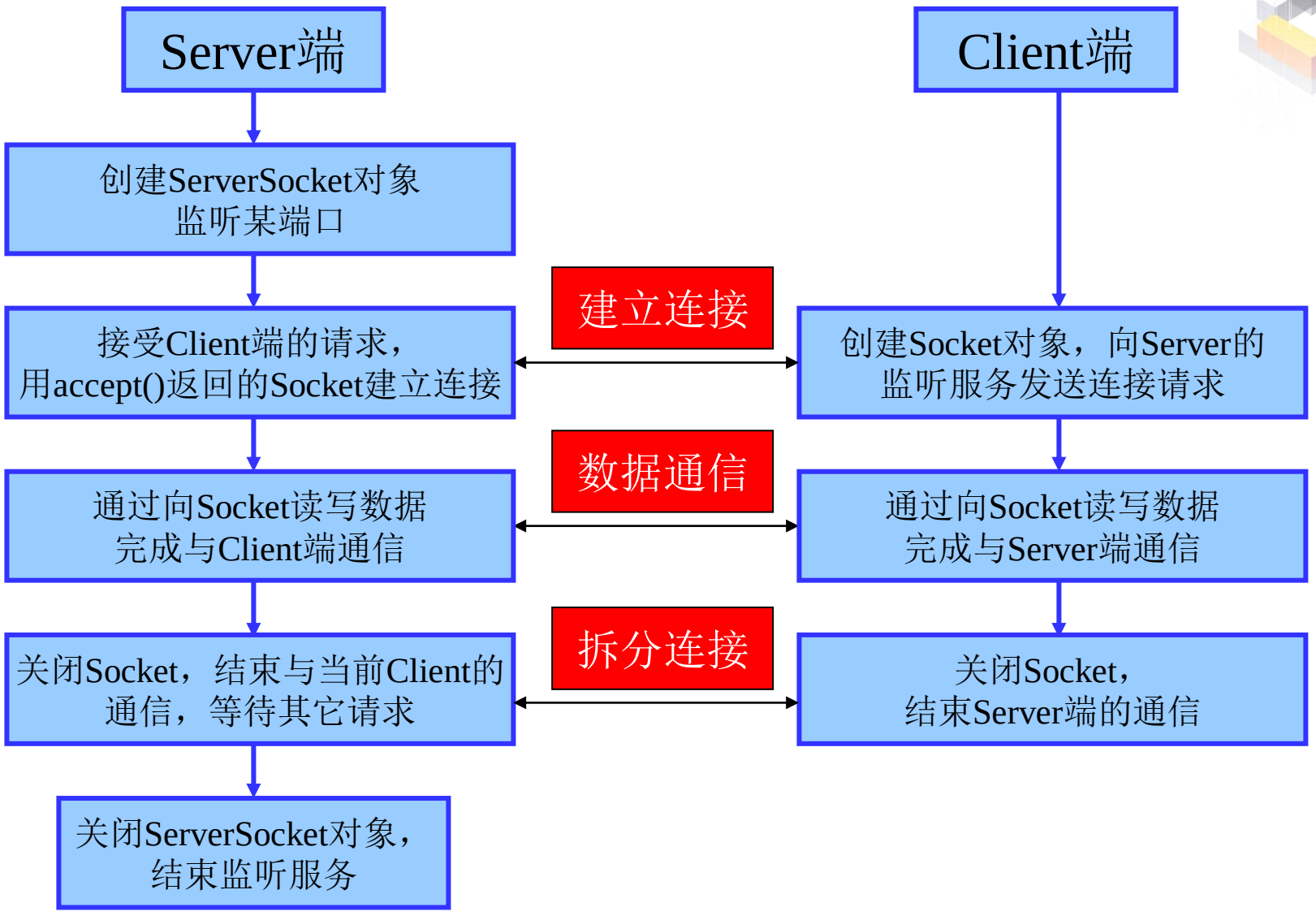
- 什么是Socket?
  - Socket是网络上运行的程序之间双向通信链路的最后终结点
  - IP与端口的组合得出一个套接字,可以完全分辨internet上运行的程序



Socket

- 在服务器端通过指定一个用来等待的连接的端口号创建一个 ServerSocket实例
- 在客户端通过规定一个主机和端口号创建一个 socket实例,连到服务器上
- ServerSocket类的accept方法使服务器处于阻塞状态,等待用户请求
- Socket类和ServerSocket是**基于TCP/IP协议**，端口必须在 0 到 65535 之间

Socket通信过程


Server端
Client端
创建ServerSocket对象
监听某端口
接受Client端的请求，
用accept()返回的Socket建立连接
建立连接
创建Socket对象，向Server的
监听服务发送连接请求
通过向Socket读写数据
完成与Client端通信
数据通信
通过向Socket读写数据
完成与Server端通信
关闭Socket，结束与当前Client的
通信，等待其它请求
拆分连接
关闭Socket，
结束Server端的通信
关闭ServerSocket对象，
结束监听服务

简单的服务器与客户的交互

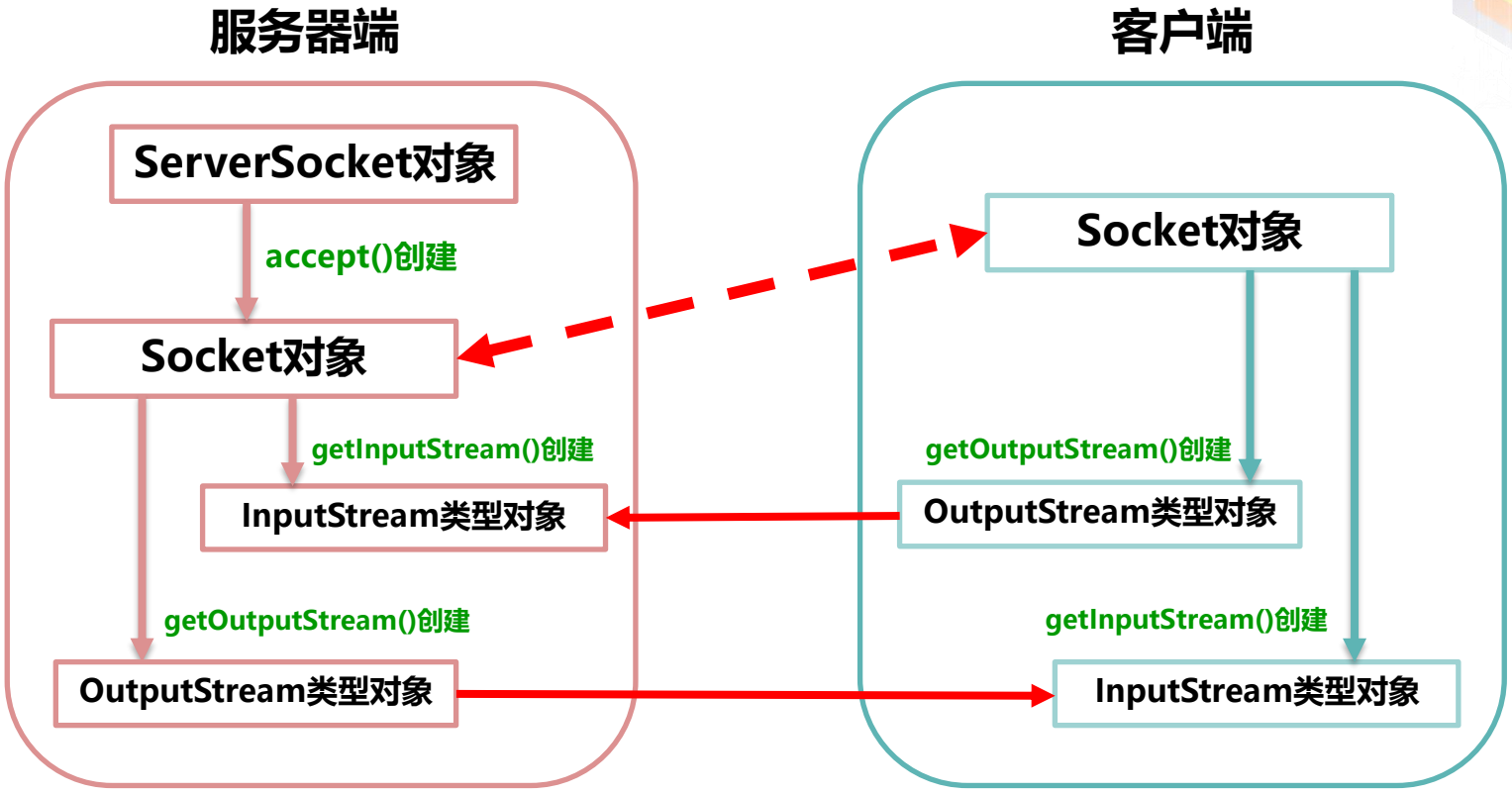
Network_3

服务器端： “EchoServer”

- ServerSocket：监听本机端口
- Socket：连接客户端
- PrintWriter ：向客户端输出数据
- BufferedReader：读取客户端传入的数据

客户端：“EchoClient”

- Socket: 连接服务器
- BufferedReader：读入本地用户命令行输入
- PrintWriter：向服务器传输数据

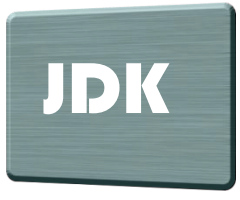

服务器端
客户端
ServerSocket对象
accept()创建
Socket对象
getInputStream()创建
InputStream类型对象
getOutputStream()创建
OutputStream类型对象
Socket对象
getOutputStream()创建
OutputStream类型对象
getInputStream()创建
InputStream类型对象

基于Socket的网络通信

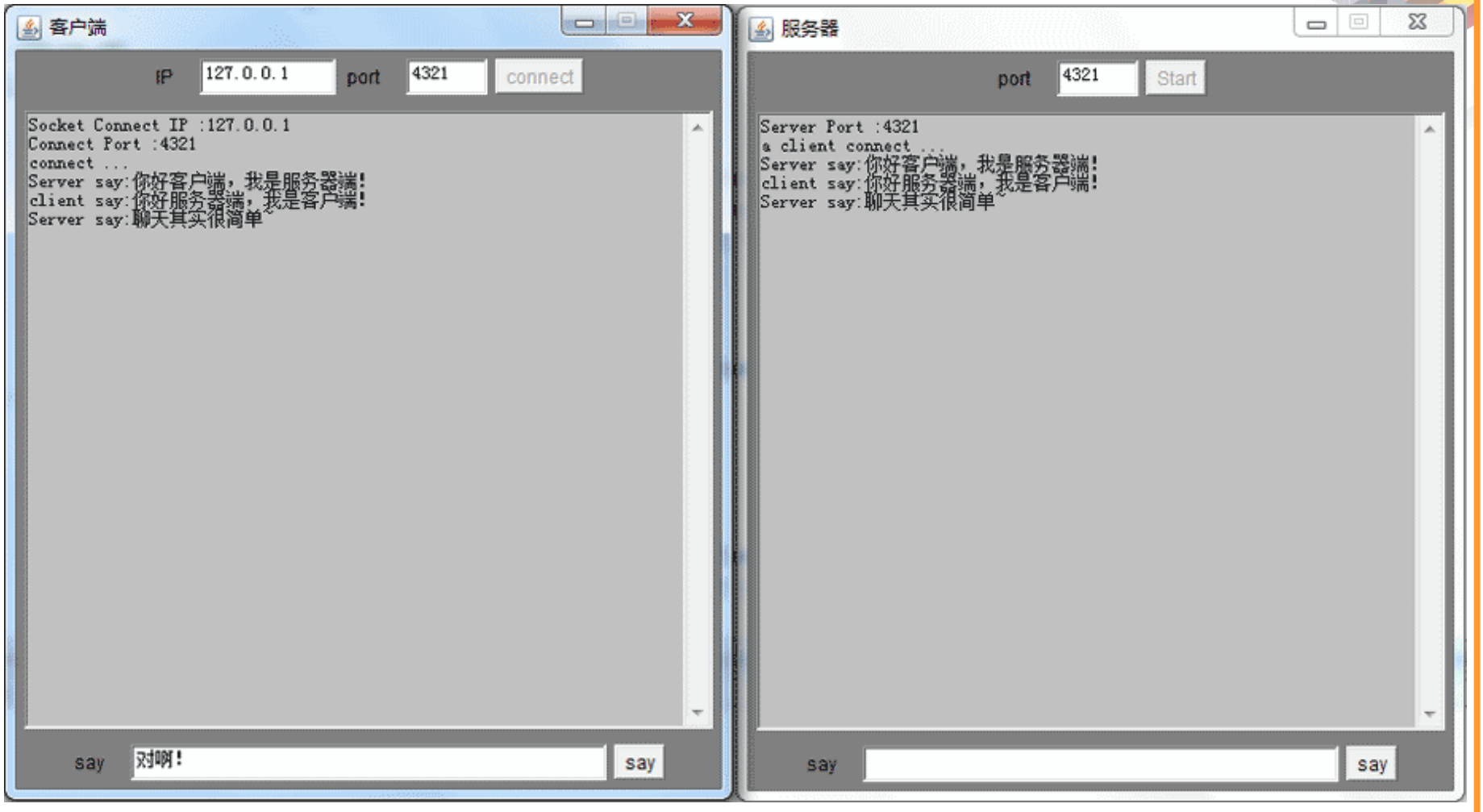
- 当程序需要建立网络连接时，必须有一台机器运行一个程序，随时等候连接，而另一端的程序则对其发出连接请求,同电话系统类似
- 建立连接的过程
  - 服务器端生成一个**ServerSocket实例对象**，随时监听客户端的连接请求
  - 客户端生成一个**Socket实例对象**，并发出连接请求
  - 服务器端通过accept()方法接收到客户端的请求后，开辟一个接口与之进行连接，并生成所需的I/O数据流
  - 通信都是通过一对InputStream()和OutputStream()进行的。通信结束后，两端分别关闭对应的Socket接口

Java 简单的C\S的交互

- 在Java中，基于TCP协议实现网络通信的类有两个：在客户端的Socket类和在服务器端的ServerSocket类。
- SeverSocket类的功能是建立一个Sever，并通过accept()方法随时监听客户端的连接请求



进一步的“聊天”程序



例： GUI界面的简单网络聊天程序

URL与 Socket 通信的区别

通信方式

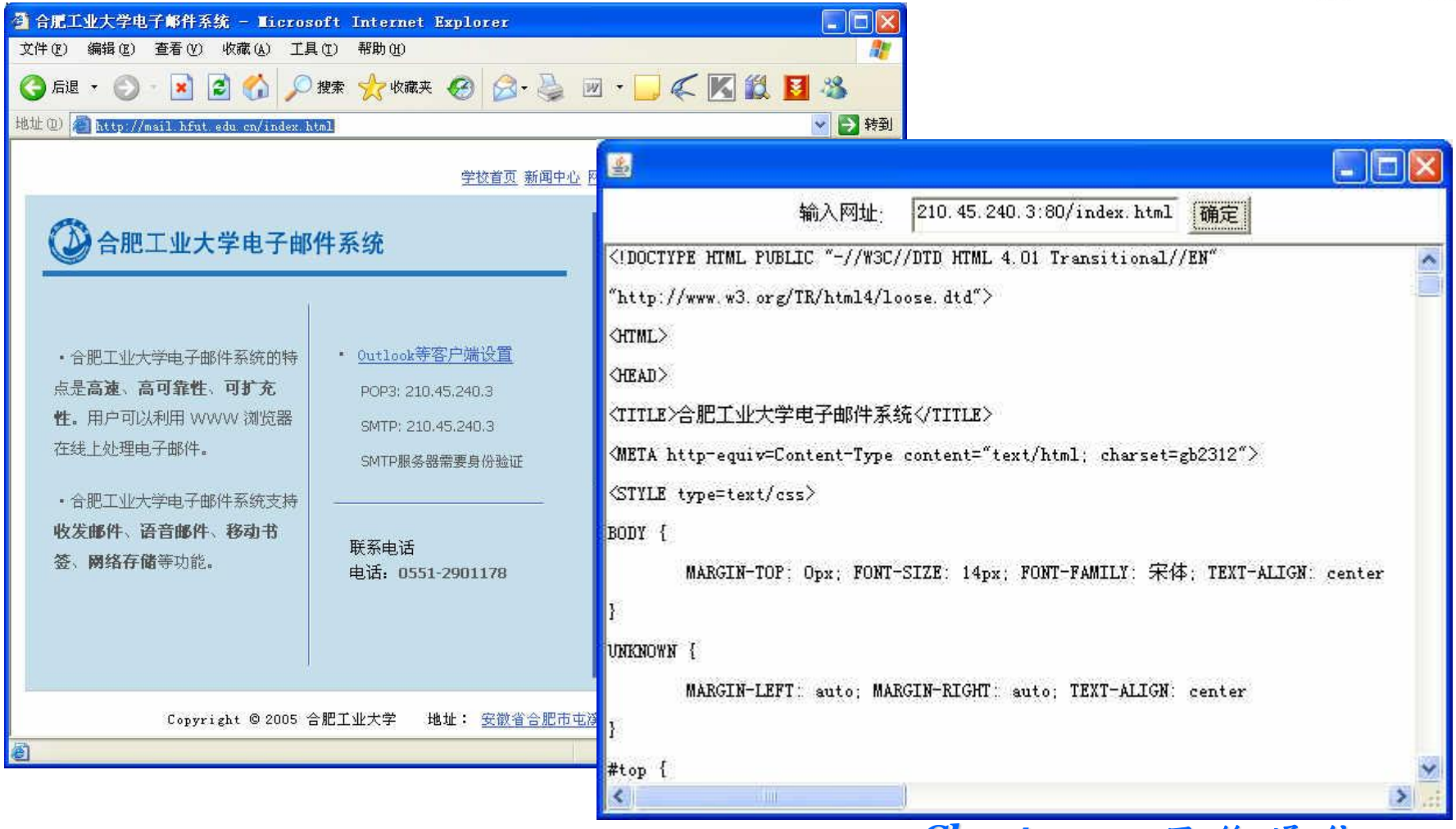
- **Socket**在服务器端运行通信程序，不停地监听客户端的连接请能够使通信发求，等待客户端请求服务当客户端提出请求时，马上连接并通信；
- **URL**进行通信时，**被动**等待客户端的请求。

连接数

- Socket服务器可以**同时与多个客户端**进行相互通信
- URL通信方式是服务器**只能与一个客户端**进行通信。

“GUI” 浏览器

- 基本的图形化URL程序 AWT + URL对象
- 访问 http://210.45.240.4:80/index.html



本章总结

本章学习Java语言在网络通讯方面的应用

- 网络基本概念
- java.net中类的使用
- 简单的Socket程序设计

作业5

作业5：见作业文档。


Thank You !