



网络基础



java.net包



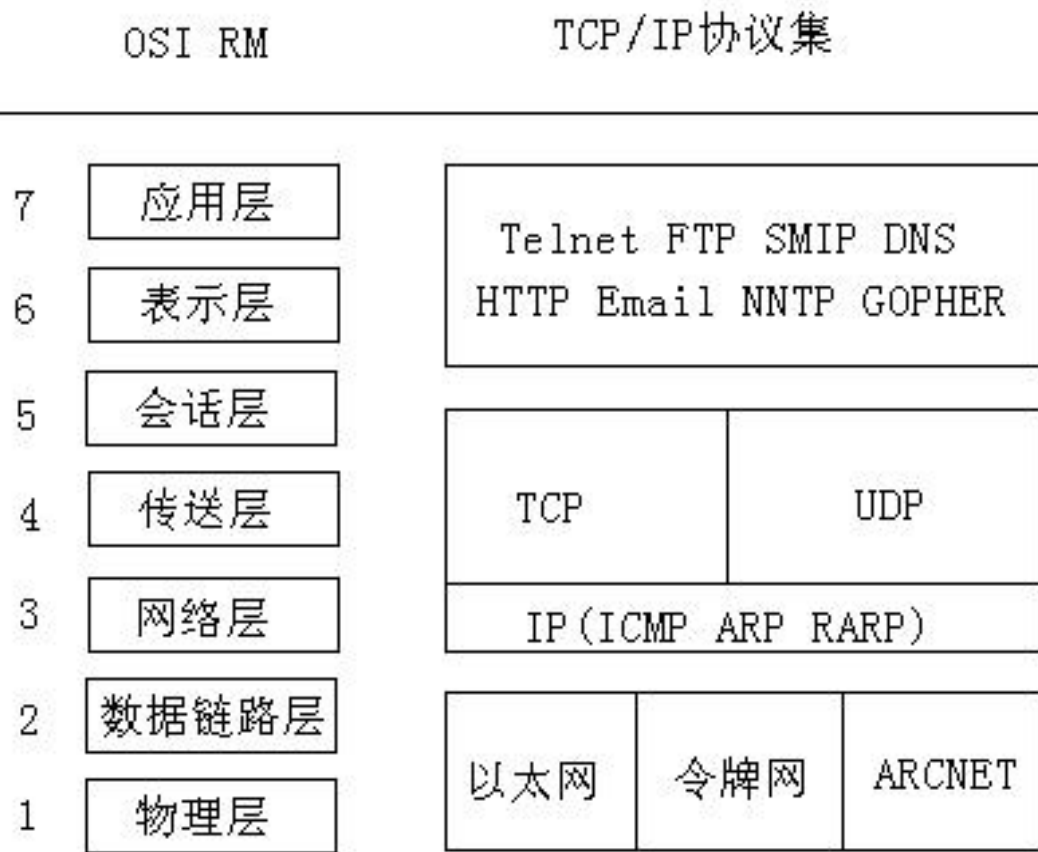
IP组播编程

- **Java具有支持Internet和WWW等的完整软件包。使用Java语言可以非常容易地完成网络程序设计，这一独特的优点是Java风行世界的原因之一。**
 - **在java.net包中的类是针对Internet网设计的，即Internet网是使用这些类的硬件基础。**
 - **Internet网提供的服务不断增加，如电子邮件、远程文件传输服务(ftp)、远程登录(Telnet)、网络新闻、网络浏览(WWW服务)、网络查询(WAIS、Gopher和Archie等)和电子公告牌BBS等成为最常用、最受欢迎的服务项目。**
-

- **Internet网成功的关键是它具有把不同网络互连起来的能力，它的这种互连能力体现在它使用的一组网络协议TCP/IP上。**
 - **TCP/IP这个协议名称可分为两部分：TCP是指“传输控制协议”（即Transport Control Protocol），IP指“互连网协议”（Internet Protocol）。**
-

分层模型

- **网络接口层：**与物理网络打交道的物理网络部分，它相当于OSI的物理层和数据链路层，负责接收和发送数据帧。
- **网际层：**与OSI的网络层相对应，针对网际环境设计的，具有更强的网际通信和互连能力。主要处理Internet网络中大量的广域网和局域网互连时的数据交换。

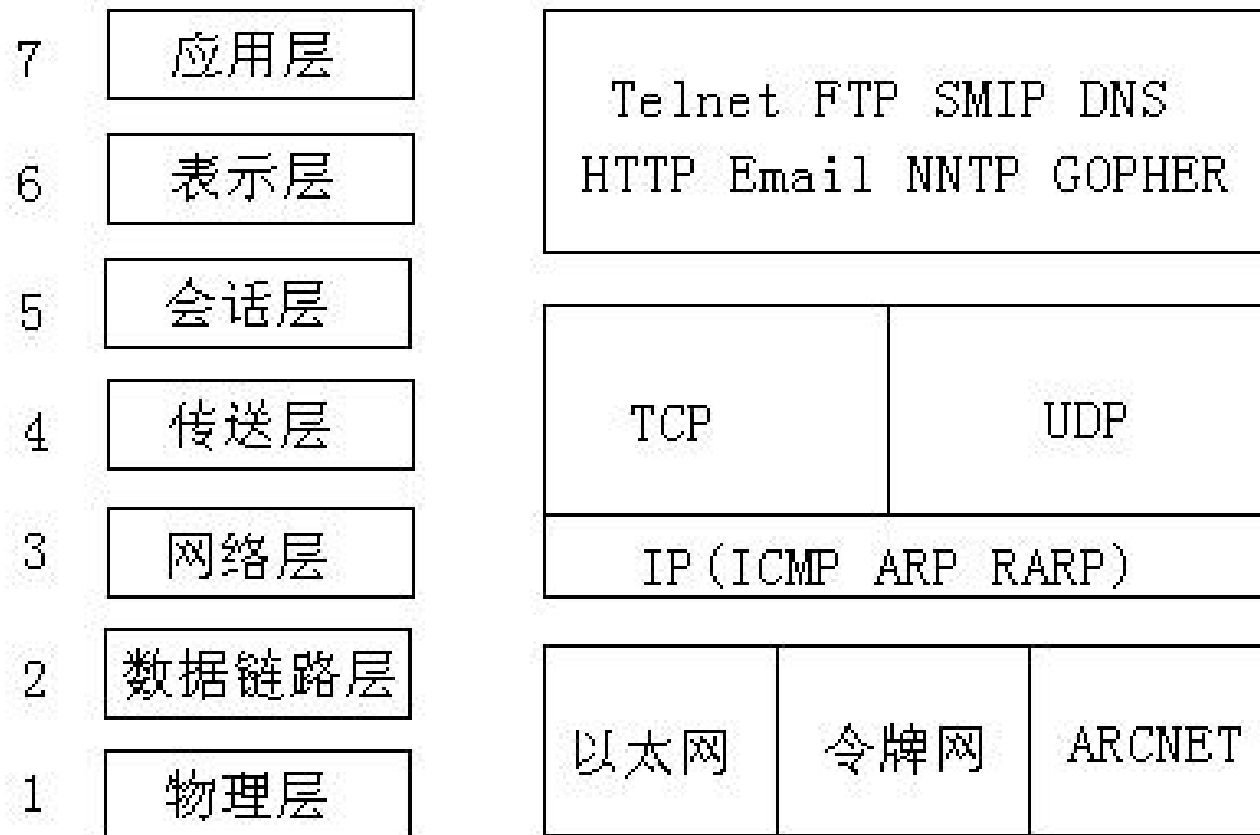


TCP/IP与OSI对照图

分层模型

OSI RM

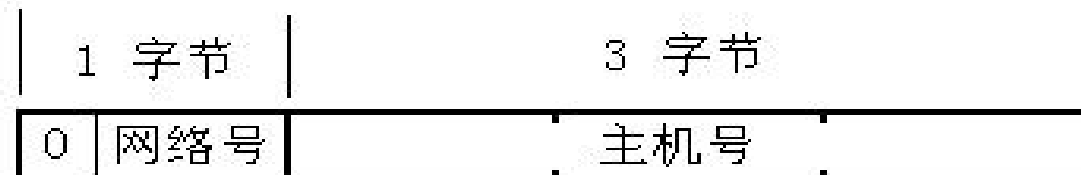
TCP/IP协议集



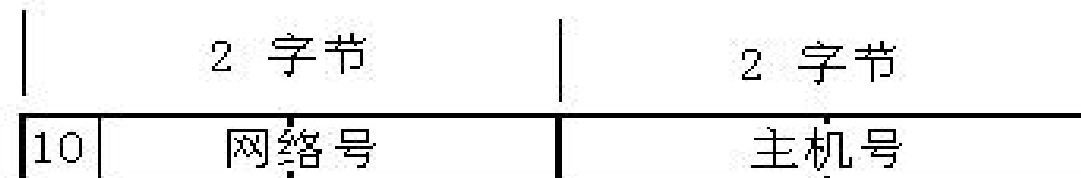
TCP/IP与OSI对照图

IP地址

- TCP/IP协议为每台主机分配一个唯一的32位网际地址，或称为IP地址。
- 32位的IP地址在书写时分成四个部分，每一个部分用"."分隔，其取值范围为0至255。例：202.112.14.181
- IP地址分为：网络号和主机号(netid, hostid)，网络号定义为主机号为全0的网际地址。
- 根据网际上网络的规模，把IP地址分为三类：A类，B类，C类。



A类地址



B类地址



C类地址

IP地址分类示意图

域名系统DNS

- **DNS (Domain Name System, 域名系统)**，因特网上作为域名和IP地址相互映射的一个分布式数据库，能够使用户更方便的访问互联网，而不用去记住能够被机器直接读取的IP数串。
 - **host.subdomain1.[subdomain2...].rootdomain**
 - **通过主机名，最终得到该主机名对应的IP地址的过程叫做域名解析（或主机名解析）。DNS协议运行在UDP协议之上，使用端口号53。**
-

套接字(socket)

- 网络上的两个程序通过一个双向的通信连接实现数据的交换，这个连接的一端称为一个socket。
- Socket的英文原义是（电力系统的）“插座”，在网络中用于描述IP地址和端口。在Internet上的主机一般同时提供几种服务，每种服务都绑定到一个端口上。客户软件将插头插到不同编号的插座（ip和端口），就可以得到不同的服务。
- 如果一个进程要通过网络向另一个进程发送数据，只需简单地写入与socket相关联的输出流。一个进程通过从与socket相关联的输入流读来读取另一个进程所写的数据。
- Java中套接字使用流模型，一个socket包括两个流：一个输入流和一个输出流。

客户端/服务器模型

- 即Client/Server(客户机/服务器)结构，通过任务合理分配到Client端和Server端，降低了系统通讯开销，可充分利用两端硬件环境优势。
 - C/S的优点是能充分发挥客户端 P C 的处理能力，很多工作可以在客户端处理后再提交给服务器。
 - 缺点主要有：
 - 适用于局域网环境可连接用户数有限；
 - 客户端需要安装专用的客户端软件；
 - 对客户端的操作系统一般会有限制。
-



网络基础



java.net包



IP组播编程

java.net包

- **java.net包可以根据下列功能进行分组：**
 - **Internet寻址(InetAddress类和URL类)**
 - **TCP/IP面向连接服务类(Socket类和ServerSocket类)**
 - **UDP/IP无连接服务类(DatagramPacket类和DatagramSocket类)**
 - **MIME内容类型处理器(ContentHandler类和URLStreamHandler类)**
 - **WWW相关类(URLConnection类和URLStreamHandler类)**
-

➤ InetAddress类和URL类:

- InetAddress类提供有关从域名地址查询IP地址的方法，类中没有构造器。
TCP/IP面向连接服务类(Socket类和ServerSocket类)。
- getAddress() 返回4个字节的IP地址，是4个整数，有可能是负数，需要进行转换处理，才能正确显示。
- getHostName() 返回被查询主机域名地址，即生成InetAddress对象时使用的字符串参数。
- getHostAddress() 返回IP地址，与getAddress不同，它是用字符串表示的IP 地址，IP地址用十进制数表示，即"%d.%d.%d.%d"。

➤ **URL类支持对WWW服务器的访问：使用这个类可以方便地访问WWW服务器上的资源。**

➤ **URL类构造器：**

```
public URL(String protocol,String host,int port,  
            String file);
```

```
public URL(String protocol,String host,String file)  
            throws MalformedURLException;
```

```
public URL(String spec)throws MalformedURLException;
```

```
public URL(URL context, String spec)  
            throws MalformedURLException;
```

java.net包

- URL的四个构造器能够很方便地生成URL对象。
- 其中：
 - protocol: 协议
 - host: 主机域名
 - port: 端口号
 - file: 文件名
 - spec: 完整的URL地址字符串
 - context是spec内容的补充。
- MalformedURLException是创建URL对象可能抛出异常。

Socket类和ServerSocket类*

- **Socket类用于支持客户机，ServerSocket类用于支持服务器，它们一起提供完整的TCP/IP连接服务功能。**
 - **编写服务程序，要求首先建立客户机和服务器的TCP连接，要建立TCP连接必须涉及二个端口：服务器的端口和客户机的端口。**
 - **客户机上使用的端口是临时端口，不必应用程序指定，由系统自动分配，在通讯结束后被销毁。**
 - **服务器的端口应由应用程序选择。**
-

ServerSocket类用于支持服务器

➤其构造器定义如下:

➤ `public ServerSocket(int port)`

➤ `public ServerSocket(int port,int backlog)`

➤ `public ServerSocket(int port,int backlog, InetAddress bindAdd)`

➤其中:

➤ `port`: 表示服务器守候的端口号;

➤ `backlog`: 连入服务器的队列长度, 即允许同时连入服务器的客户机数目, 缺省值为50。

Socket类用于支持客户机

➤构造器定义如下:

public Socket(String host, int port)

public Socket(InetAddress address,int port)

public Socket(String host,int port, InetAddress localAddr,int localPort)

public Socket(InetAddress address,int port, InetAddress localAddr,int localPort)

➤其中:

➤host:连接的服务器

➤port:服务器端口

➤address: 服务器的IP地址

➤localPort:客户机端口

连接过程

▪ Server

```
ServerSocket server = new  
ServerSocket(8001);
```

```
Socket socket = server.accept();
```

wait.....

connected.....

```
InputStream input =  
socket.getInputStream();
```

```
input.read();
```

▪ Client

.....

.....

```
Socket socket = new  
Socket(192.168.1.1, 8001)
```

connected.....

```
OutputStream output =  
socket.getOutputStream();
```

```
output.write();
```

协议

- **所谓协议，就是通信双方事先规定的信息交换规则：先运行服务程序，使它处于守候状态，再运行客户程序建立连接，如果连接成功，服务程序发送数据，然后进行交互通讯，即收到一条信息后则应该发送一条应答信息。**
- **结束通讯，在双方通讯中，发送中止信息后中止两方的通讯程序。**

DatagramPacket类和DatagramSocket类

- **DatagramPacket类用来实现一个无连接包的传送服务。 每条消息从一台机器发送到到另一台机器的选径仅依赖于这个包所包含的信息。 多个包从一台机器发送到另一台机器可能经过不同的路径，并且以任意顺序到达。**
 - **构建器：**
 - **DatagramPacket(byte[] buf, int length, InetAddress addr, int port)：**从Buf数组中，取出Length长的数据创建数据包对象，目标是Addr地址，Port端口。
 - **DatagramPacket(byte[] buf, int offset, int length, InetAddress address, int port)：**从Buf数组中，取出Offset开始的、Length长的数据创建数据包对象，目标是Addr地址，Port端口。
-

DatagramPacket类和DatagramSocket类

- **DatagramSocket类表示用来发送和接收数据报包的套接字。**
- **数据报套接字是包投递服务的发送或接收点。每个在数据报套接字上发送或接收的包都是单独编址和路由的。从一台机器发送到另一台机器的多个包可能选择不同的路由，也可能按不同的顺序到达。**
- **在DatagramSocket上总是启用UDP广播发送。为了接收广播包，应该将 DatagramSocket绑定到通配符地址。在某些实现中，将 DatagramSocket绑定到一个更加具体的地址时广播包也可以被接收。**

IP组播

- **IP 组播是指在IP 网络中将数据包以尽力传送best-effort 的形式发送到网络中的某个确定节点子集，这个子集称为组播组 multicast group。IP组播的基本思想是源主机只发送一份数据，这份数据中的目的地址为组播组地址。组播组中的所有接收者都可接收到同样的数据拷贝，并且只有组播组内的主机目标主机可以接收该数据。网络中其它主机不能收到组播组用D 类IP地址 224.0.0.0 239.255.255.255 来标识。**
- **IP 组播技术有效地解决了单点发送多点接收的问题，实现了IP 网络中点到多点的高效数据传送，能够大量节约网络带宽、降低网络负载。**

思考问题

1

在Java中网络编程类有哪些？

2

Java中如何实现网络编程的？

3

在C/S模式中如何实现TCP/IP通信的？