

1、IP 地址 (InetAddress)

2、端口号

3、InetAddress类

4、网络协议

1. 概念

2. 作用

3. 五元组

4. 协议分层

1. 通信协议分层的思想

2. 分层的作用

3. OSI 七层模型

4. TCP/IP五层 (或四层) 模型 (TCP/IP协议簇)

5. 网络设备所在分层

6. 网络分层对应

7. 封装和分用

1、IP 地址 (InetAddress)

1. IP地址主要用于标识网络主机、其他网络设备（如路由器）的**网络地址**。简单说，IP地址用于**定位主机的网络地址**。

2. 本地回环地址 (hostAddress): 127.0.0.1 主机名 hostName localhost

3. IP 地址 分类方式 1: IPV4 和 IPV6

1. IPV4: 4 个字节组成，4 个 0~255 。大概42 亿，30亿都在北美，亚洲 4 亿。2011 年初已经用尽。以点分十进制表示，如 192.168.0.1

2. IPV6: 128 位 (16 个字节), 写 成 8 个无符号整数，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

4. IP 地址 分类方式 2 : 公网地址(万维网使用) 和 私有地址(局域网使用) 。 192.168.开头的就是私有地址，范围即为 192.168.0.0 ~192.168.255.255，专门为组织机构内部使用

5. 特点：不易记忆

2、端口号

1. 在网络通信中，IP地址用于标识主机网络地址，端口号可以标识主机中发送数据、接收数据的进程。简单说：**端口号用于定位主机中的进程**。

2. 不同的进程有不同的端口号

3. 被规定为一个16位的整数，范围：0~65535。

4. 端口分类：

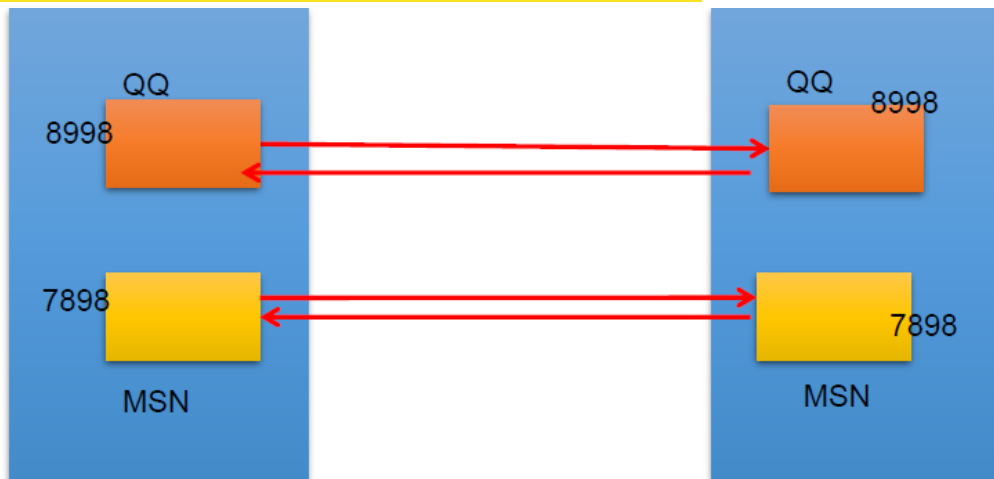
1. 公认端口：0~1023。被预先定义的服务通信占用（如：HTTP占用端口80，FTP 占用端口21，Telnet占用端口23）

2. 注册端口：1024~ 40151。分配给用户进程或者应用程序。（如：Tomcat 占用端口8080，MySQL 占用端口3306，Oracle 占用端口 1521 等）

3. 动态端口：49152~65535。

5. 注意事项：两个不同的进程，不能绑定同一个端口号，但一个进程可以绑定多个端口号。

6. 端口号与 IP 地址的组合得出一个网络套接字：Socket。



3、InetAddress类

1. Internet 上的主机有两种方式表示地址

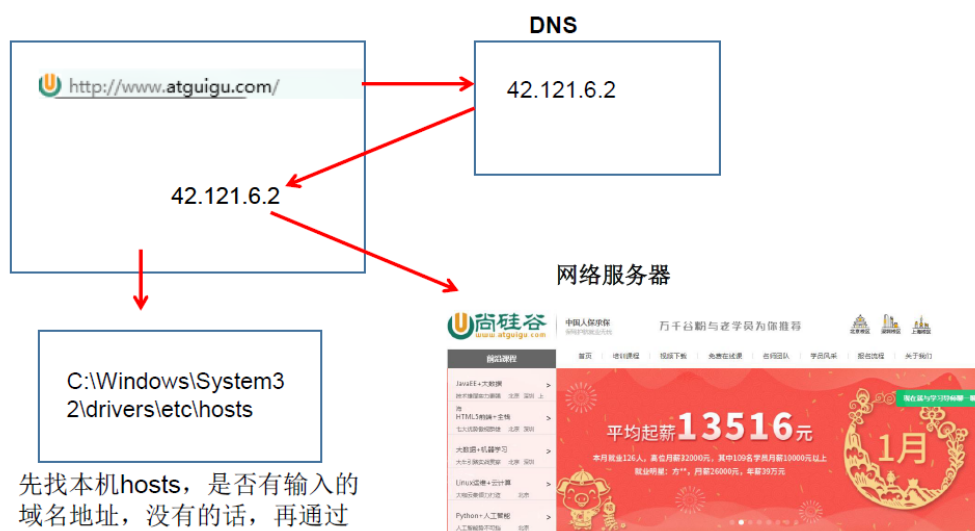
1. 域名：www.baidu.com

2. IP地址：202.108.35.210

2. InetAddress 类主要表示 IP 地址，两个子类：Inet4Address、Inet6Address。

3. InetAddress 类对象含有一个 Internet 主机地址的域名和IP地址：www.baidu.com 和 202.108.35.210

4. 域名容易记忆，当在连接网络时输入一个主机的域名后，域名服务器 (DNS) 负责将域名转化成 IP 地址，这样才能和主机建立连接。-----域名解析



让天下没有难学的技术

5. InetAddress 类没有提供公共的构造器，而是提供了如下几个静态方法来获取InetAddress 实例：

- a. public static InetAddress getLocalHost()
- b. public static InetAddress getByName(String host)

6. InetAddress 提供了如下几个常用的方法：

- a. public String getHostAddress(): 返回 IP 地址字符串（以文本表现形式）。
- b. public String getHostName()：获取此 IP 地址的主机名
- c. public boolean isReachable(int timeout): 测试是否可以到达该地址

```
1 public class InetDaaressTest {
2     public static void main(String[] args) throws UnknownHostException {
3         InetAddress address = InetAddress.getByName("www.baidu.com");
4
5         System.out.println(address);
6         // 获取 InetAddress 对象所含的域名
7         System.out.println(address.getHostName());
8         // 获取 InetAddress 对象所含的 IP 地址
9         System.out.println(address.getHostAddress());
10        System.out.println();
11        // 获取本机的域名和 IP 地址
12        InetAddress address1 = InetAddress.getLocalHost();
13        System.out.println(address1);
14    }
15 }
```

4、网络协议

1. 概念

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。只有遵守这个约定，计算机之间才能相互通信交流。通常由三要素组成：

- 1. 语法：即数据与控制信息的结构或格式；
- 2. 语义：即需要发出何种控制信息，完成何种动作以及做出何种响应；
语义主要用来说明通信双方应当怎么做。用于协调与差错处理的控制信息。
- 3. 时序：即事件实现顺序的详细说明。

时序定义了何时进行通信，先讲什么，后讲什么，讲话的速度等。比如是采用同步传输还是异步传输。

协议（protocol）最终体现为在网络上传输的**数据包**的格式

2. 作用

计算机之间的传输媒介是光信号和电信号。通过 "频率" 和 "强弱" 来表示 0 和 1 这样的信息。要想传递各种不同的信息，就需要约定好双方的数据格式

知名协议的默认端口：

系统端口号范围为 0 ~ 65535，其中：0 ~ 1023 为知名端口号，这些端口预留给服务端程序绑定广泛使用的应用层协议，如：

- 1 2端口：预留给SSH服务器绑定SSH协议
- 2 21端口：预留给FTP服务器绑定FTP协议
- 3 23端口：预留给Telnet服务器绑定Telnet协议
- 4 80端口：预留给HTTP服务器绑定HTTP协议
- 5 443端口：预留给HTTPS服务器绑定HTTPS协议

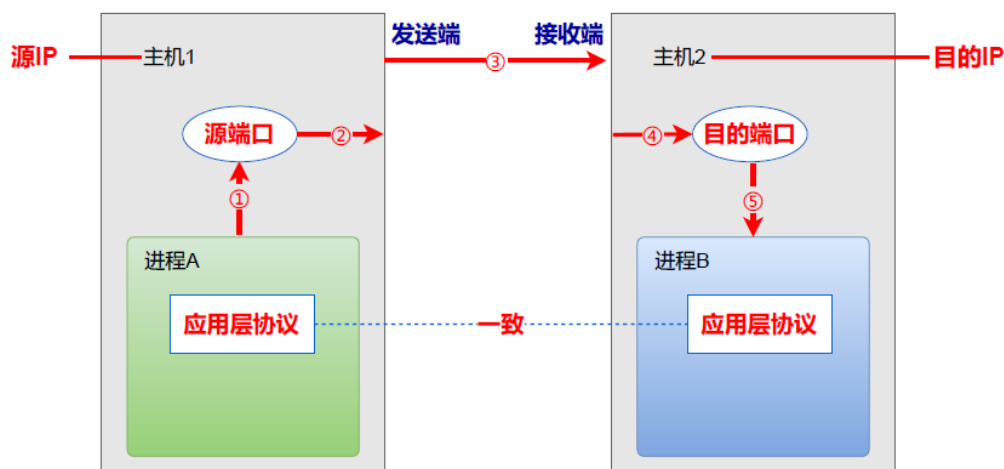
需要补充的是：

以上只是说明 0 ~ 1023 范围的知名端口号用于绑定知名协议，但某个服务器也可以使用其他 1024 ~ 65535 范围内的端口来绑定知名协议。

3. 五元组

在TCP/IP协议中，用五元组来标识一个网络通信：

1. 源IP：标识源主机
2. 源端口号：标识源主机中该次通信发送数据的进程
3. 目的IP：标识目的主机
4. 目的端口号：标识目的主机中该次通信接收数据的进程
5. 协议号：标识发送进程和接收进程双方约定的数据格式



可以在cmd中，输入 netstat -ano 查看网络数据传输中的五元组信息：

如果需要过滤（一般是通过端口号或进程PID过滤），可以使用 netstat -ano | findstr 过滤字符串

4. 协议分层

1. 通信协议分层的思想

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。各层互不影响，利于系统的开发和扩展。

2. 分层的作用

分层最大的好处，类似于面向接口编程：定义好两层间的接口规范，让双方遵循这个规范来对接。

在代码中，类似于定义好一个接口，一方为接口的实现类（提供方，提供服务），一方为接口的使用类（使用方，使用服务）：

- 对于使用方来说，并不关心提供方是如何实现的，只需要使用接口即可
- 对于提供方来说，利用封装的特性，隐藏了实现的细节，只需要开放接口即可，即可以很方便的对其中的某层/某些协议进行灵活的替换。

这样能更好的扩展和维护，如下图：



3. OSI 七层模型

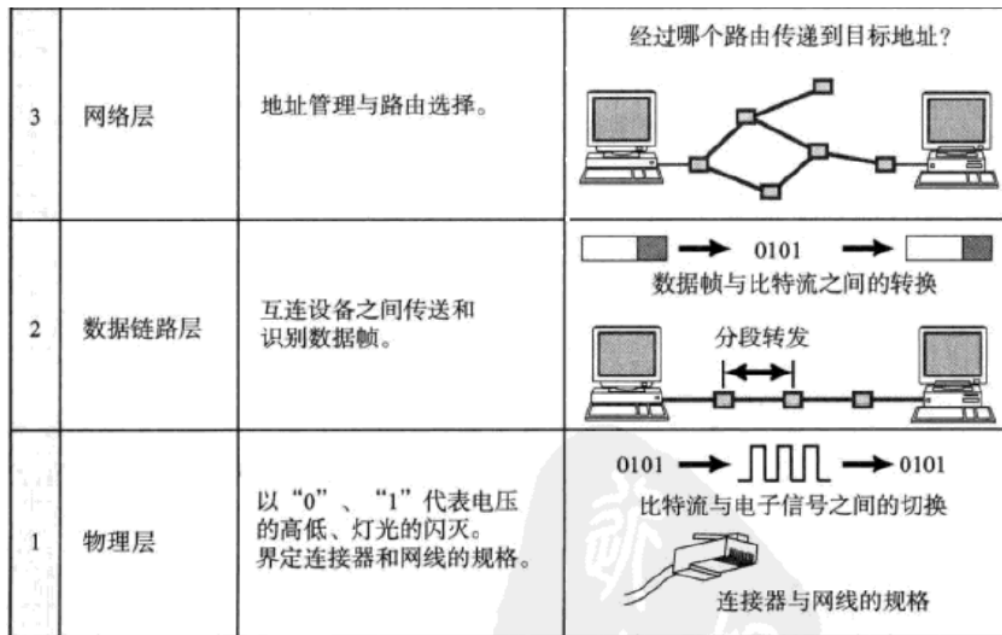
OSI：即 Open System Interconnection，开放系统互连

- OSI 七层网络模型是一个逻辑上的定义和规范：把网络从逻辑上分为了7层。
- OSI 七层模型是一种框架性的设计方法，其最主要的功能就是帮助不同类型的主机实现数据传输；
- 它的最大优点是将服务、接口和协议这三个概念明确地区分开来，概念清楚，理论也比较完整。通过七个层次化的结构模型使不同的系统不同的网络之间实现可靠的通讯。



OSI 七层模型划分为以下七层：

	分层名称	功 能	每层功能概览
7	应用层	针对特定应用的协议。	<p>针对每个应用的协议</p> <p>电子邮件 ↔ 电子邮件协议</p> <p>远程登录 ↔ 远程登录协议</p> <p>文件传输 ↔ 文件传输协议</p>
6	表示层	设备固有数据格式和网络标准数据格式的转换。	<p>接收不同表现形式的信息，如文字流、图像、声音等</p>
5	会话层	通信管理。负责建立和断开通信连接（数据流动的逻辑通路）。管理传输层以下的分层。	<p>何时建立连接，何时断开连接以及保持多久的连接？</p>
4	传输层	管理两个节点之间的数据传输。负责可靠传输（确保数据被可靠地传送到目标地址）。	<p>是否有数据丢失？</p>



OSI 七层模型既复杂又不实用：所以 OSI 七层模型没有落地、实现。

实际组建网络时，只是以 OSI 七层模型设计中的部分分层，也即是以下 TCP/IP 五层（或四层）模型来实现。

4. TCP/IP五层（或四层）模型（TCP/IP协议簇）

1. 传输层协议中有两个非常重要的协议：

- 传输控制协议 TCP(Transmission Control Protocol)
- 用户数据报协议 UDP(User Datagram Protocol)

2. TCP/IP 以其两个主要协议：**传输控制协议（TCP）和网络互联协议（IP）而得名**，实际上是一组协议，包括多个具有不同功能且互为关联的协议。

3. IP(Internet Protocol) 协议是**网络层的主要协议，支持网间互连的数据通信**。

4. TCP/IP 协议模型从更实用的角度出发，形成了高效的**五（四）层体系结构**，即**物理链路层、IP 层、传输层和应用层**。

5. TCP/IP通讯协议采用了5层的层级结构，每一层都呼叫它的下一层所提供的网络来完成自己的需求。

应用层：**负责应用程序间沟通**，如简单电子邮件传输（SMTP）、文件传输协议（FTP）、网络远程访问协议（Telnet）等。我们的网络编程主要就是针对应用层。

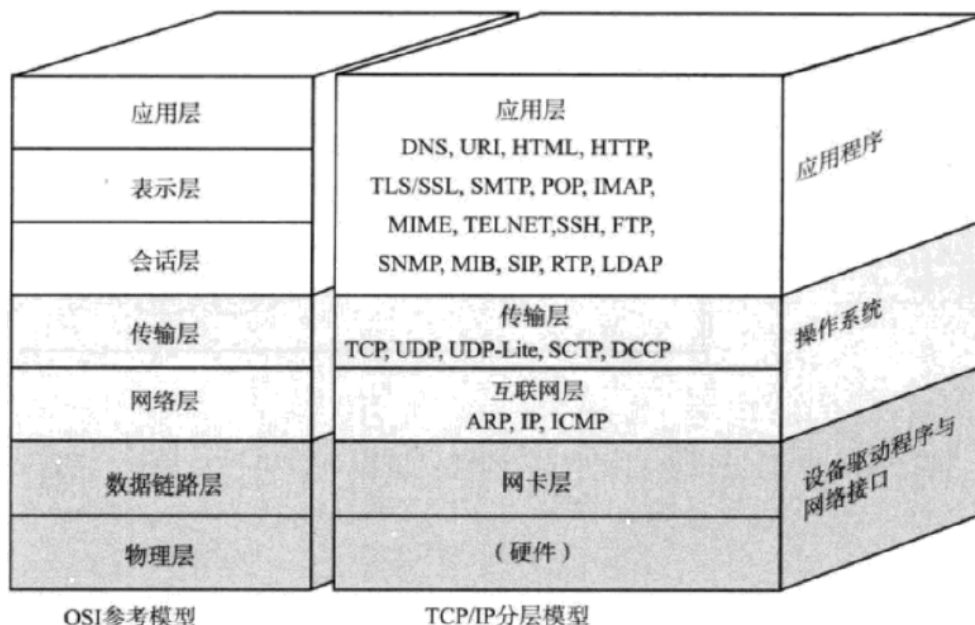
传输层：**负责两台主机之间的数据传输**。如传输控制协议（TCP），能够确保数据可靠的从源主机发送到目标主机。（只关注起点和终点，不关注具体的通信细节）

网络层：**负责地址管理和路由选择**。例如在IP协议中，通过IP地址来标识一台主机，并通过路由表的方式规划出两台主机之间的数据传输的线路（路由）。路由器（Router）工作在网路层。（**针对网络上的任意两台主机，完成一个通信路径的规划，以及具体传输**）

数据链路层：**负责设备之间的数据帧的传送和识别**。例如网卡设备的驱动、帧同步（就是说从网线上检测到什么信号算作新帧的开始）、冲突检测（如果检测到冲突就自动重发）、数据差

错校验等工作。有以太网、令牌环网，无线LAN等标准。交换机（Switch）工作在数据链路层。
(实现两个相邻设备之间的数据通信)

物理层：负责光/电信号的传递方式。（硬件设备，网络通信的基础设施）



物理层我们考虑的比较少。因此很多时候也可以称为 **TCP/IP四层模型**。

5. 网络设备所在分层

1. 对于一台**主机**，它的操作系统内核实现了从**传输层到物理层**的内容，也即是TCP/IP五层模型的下四层；
2. 对于一台**路由器**，它实现了从**网络层到物理层**，也即是TCP/IP五层模型的下三层；
3. 对于一台**交换机**，它实现了从**数据链路层到物理层**，也即是TCP/IP五层模型的下两层；
4. 对于集线器，它只实现了物理层；

注意我们这里说的是传统意义上的交换机和路由器，也称为二层交换机（工作在TCP/IP五层模型的下两层）、三层路由器（工作在TCP/IP五层模型的下三层）

6. 网络分层对应

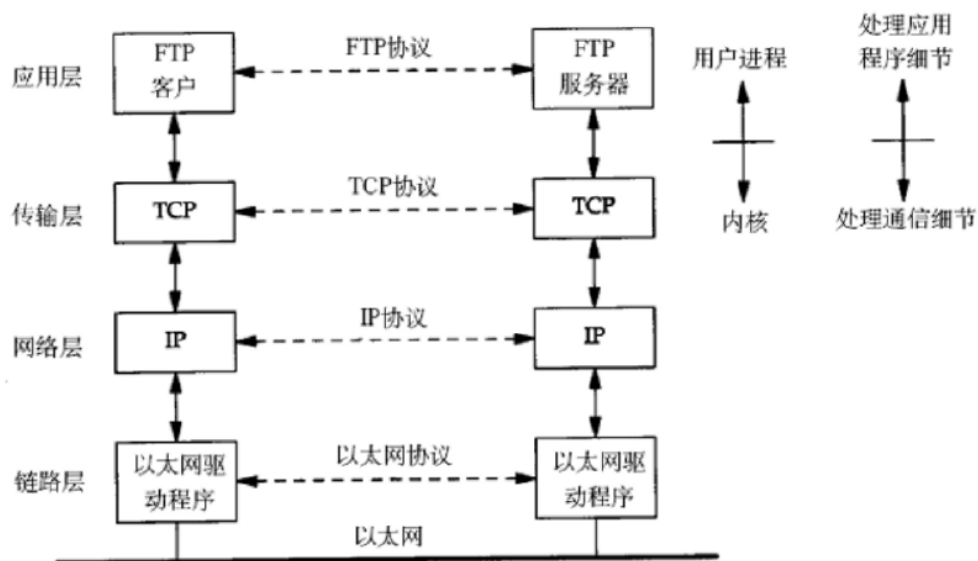
网络数据传输时，经过不同的网络节点（主机、路由器）时，网络分层需要对应。

以下为同一个网段内的两台主机进行文件传输：

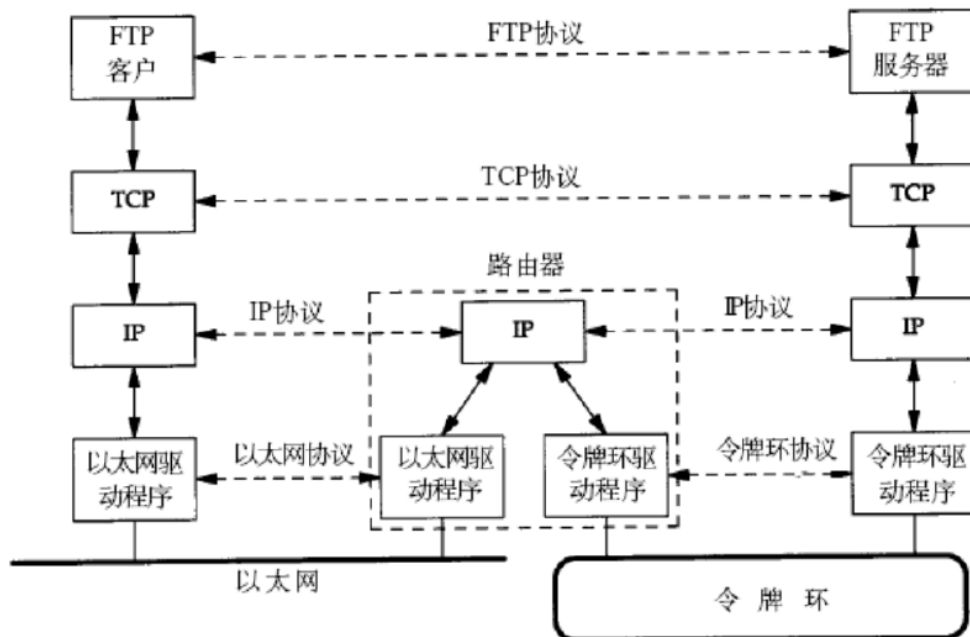
应用层	Telnet、FTP和e-mail等
传输层	TCP和UDP
网络层	IP、ICMP和IGMP
链路层	设备驱动程序及接口卡

两台计算机通过TCP/IP协议通讯的过程如下所示

TCP/IP通讯过程

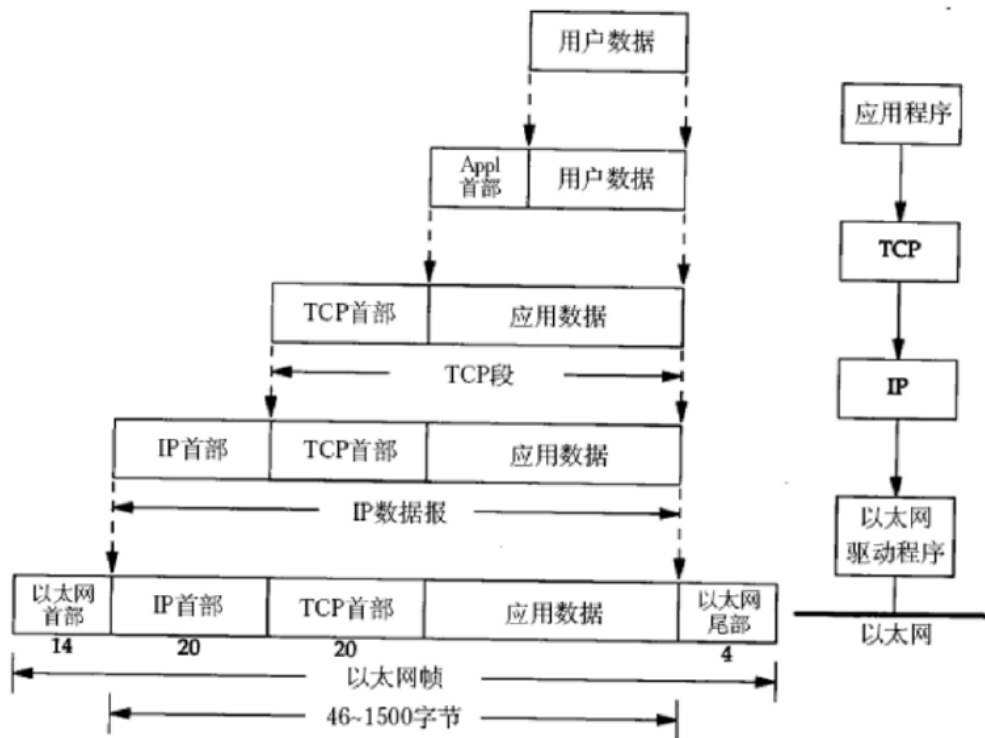


以下为跨网段的主机的文件传输：数据从一台计算机到另一台计算机传输过程中要经过一个或多个路由器



7. 封装和分用

1. 不同的协议层对数据包有不同的称谓，在传输层叫做段(segment)，在网络层叫做数据报(datagram)，在链路层叫做帧(frame)。



下图为数据分用的过程：

