1. 概述

1. 介绍

TCP/IP协议族（protocol suite）源自上世纪60年代美国政府资助的分组交换网络研究，发展至至今，已经成为一种开放系统——允许各种性能、不同厂家、不同操作系统的计算机、移动设备（如手机、网络摄像头）等相互通信。当今世界范围的互联网*Internet*由几十亿台计算机和移动设备组成，而这几十亿台设备都是通过TCP/IP协议族实现相互通信，因此我们称TCP/IP是整个互联网大厦的基石，这点毋庸置疑。

本章对TCP/IP协议族做一个概述性的介绍，为后续章节铺设足够的背景知识。

2. 分层

网络协议不外乎都是通过层次结构（*layers*）实现，有时候也称为协议栈（*protocol stack*），每一层负责通信中的一部分工作，TCP/IP协议族也不例外，它是一个4层系统：

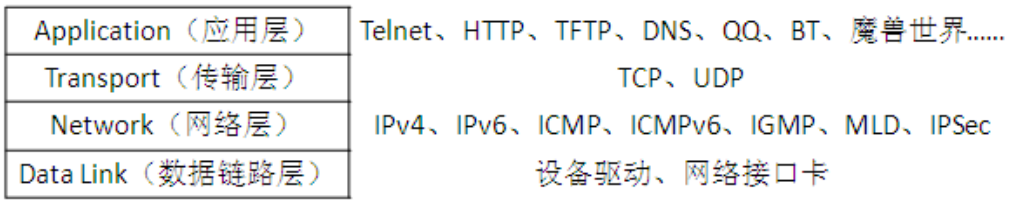


图1.1　TCP/IP协议族4层结构

从下往上每一层的作用如下：

1. 数据链路层通常简称为“链路层”，或者叫“网络接口层”，网络接口也就是我们通常所说或者所见的“有线网卡”、“无线网卡”、“ADSL拨号网卡”等等，而设备要让这些各式各样的网卡为设备效劳，设备需要这些网卡的驱动程序，驱动程序和网卡共同组成数据链路层，负责在不同的媒介（电话线、无线信号、以太网线、光纤）上处理物理信号硬件细节，以实现**数据在单一媒介两端之间传送**。如果我们说*network*，那指的是1条独立链路（一个独立的媒介），如果我们说*networking*或者*networks*，那就是指许2条以上独立链路已经通过相同协议连接起来，形成了网络互连（*internet*），如果全世界的网络互连都连接在一起，那么这个大一统的网络就是我们所说的互联网（*Internet*）。
2. 网络层也被称为“网络互连（*internet*）层”，**控制数据在网络中传输的路径**，计算机网络传输路径和我们熟知的交通网络有共同之处，有地址和寻址两部分工作，TCP/IP当前的地址系统就是我们所说的IPv4和IPv6地址，寻址则是为不同地址间找到一条合适的路径（由1条或若干条链路顺序连接而成）。ICMP（Internet Control Management Protocol）、ICMPv6等协议是一些配套的管理协议。
3. 传输层为**两个主机之间的应用程序提供数据传输服务**，为应用屏蔽网络层、链路层的差异。TCP/IP协议族中有两个使用广泛的传输层协议：TCP（Transmission Control Protocol）和UDP（User Datagram Protocol）。TCP提供**可靠的数据传输服务**，应用程序只需要将数

据交给TCP，TCP会负责数据分片发送、接收确认、数据重发等工作，应用程序可以不用管可靠传输是如何实现的；而UDP提供**简单的不可靠的传输服务**，可靠性需要应用程序自己来负责。如何使用传输层，需要根据不同的应用来分析。

4. 应用层处理**具体的应用细节**，如人机交互和数据处理，这一层是和广大网络使用者直接相关的，使用者可以不知道有传输层、网络层和链路层，但是不可能不知道哪些是有兴趣的应用：各种网站、各种下载工具、各种网游、各种即时聊天工具、各种专业网络工具如Telnet、FTP等。

下面举个例子，我们在1条局域网（最常见的以太网）链路上有2台主机，分别运行FTP服务器和FTP客户端，客户端从服务器中下载文件：

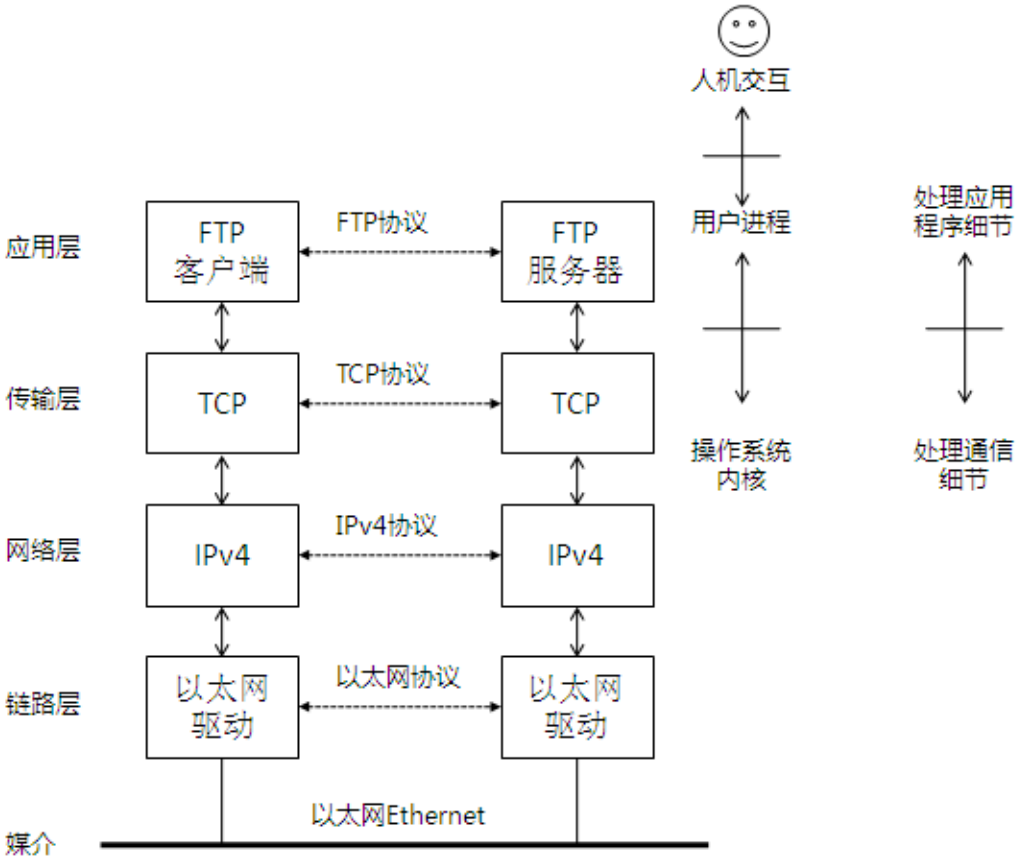


图1.2　2个主机在以太网上运行FTP

上图中，左边应用程序盒子标示成FTP客户端，右边为FTP服务器。网络应用程序大都如此设计：一端为服务器，一端为客户端，服务器为客户端提供某种类型服务，FTP提供的是文件共享与传输服务。

每个水平层之间都至少有1个协议在对等通信（同水平层之间通信），如传输层同时运行TCP和UDP协议。而每种协议只允许对等通信，如TCP协议允许2个TCP层通信，IP协议允许2个IP层通信。

在图1.2的右上角的注释表明，应用层是用户进程，主要负责人机交互和处理应用程序细节，甚至完全不知道数据是如何穿越网络的；而下3层则是通常由操作系统内核实现，这3层对于应用细节一无所知，只知如何在网络的两点间传输数据。互联网应用就是通过这种分工与合作完成的，下3层可以保证在各种各样网络设备通信中的一致性，而应用层可以为用户提供种类丰富、灵活的网络体验。

图1.2中，共有4种协议，FTP是应用层协议；TCP是传输层协议；IPv4是网络层协议；以太网协议则运行在链路层。整个TCP/IP协议族由许多协议组成，之所以叫TCP/IP是因为IP协议是必须的，TCP则是使用最广泛的，有时候TCP/IP协议族也叫IP协议族。

链路层和应用层的区别很好理解——链路层处理通信介质，而应用层处理某种用户进程。而链路层和传输层的区别可能就没那么好理解了，我们将通过一个由2条链路连接成的网络（这是最

简单的网络互连*internet*）来解释这种差别。

在介绍这个例子之前，先介绍不同的链路连接需要专门的设备，这和交通网络的十字路口有点类似，在网络中连接不通网络的设备有很多，通常是路由器*router*和网桥*bridge*。路由器大家比较耳熟，它不止连接链路还能连接不通IP，它以前的名字叫做网关*gateway*；网桥则是一系列链路连接设备的统称。现在流行的网络设备基本上都同时兼备网桥和路由器功能，但是从下面的例子分析，网桥和路由的区别：

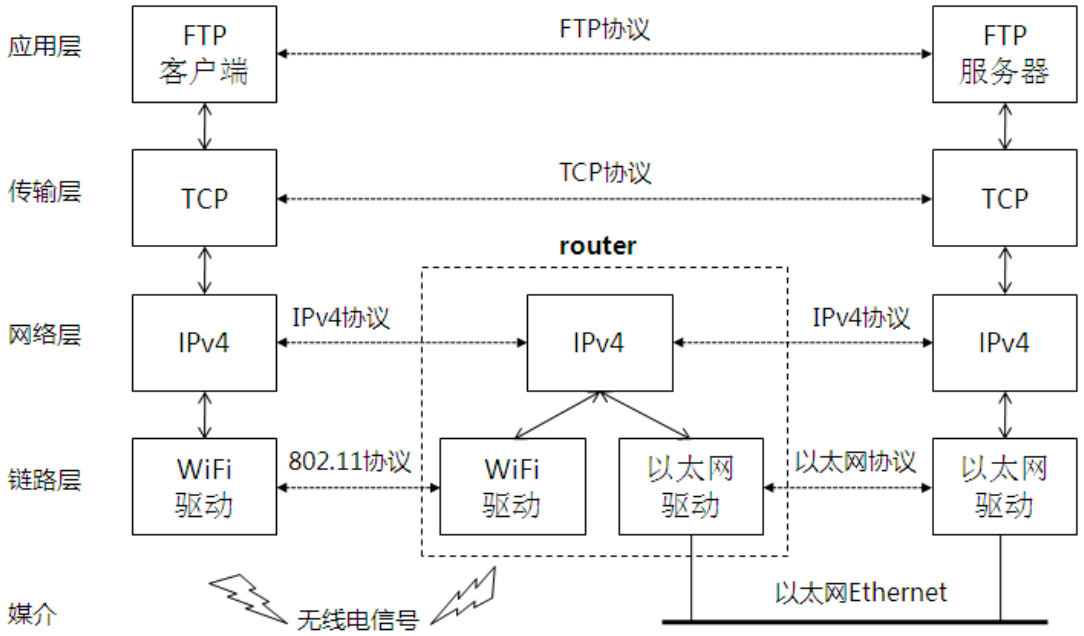


图1.3　通过路由器连接2条链路

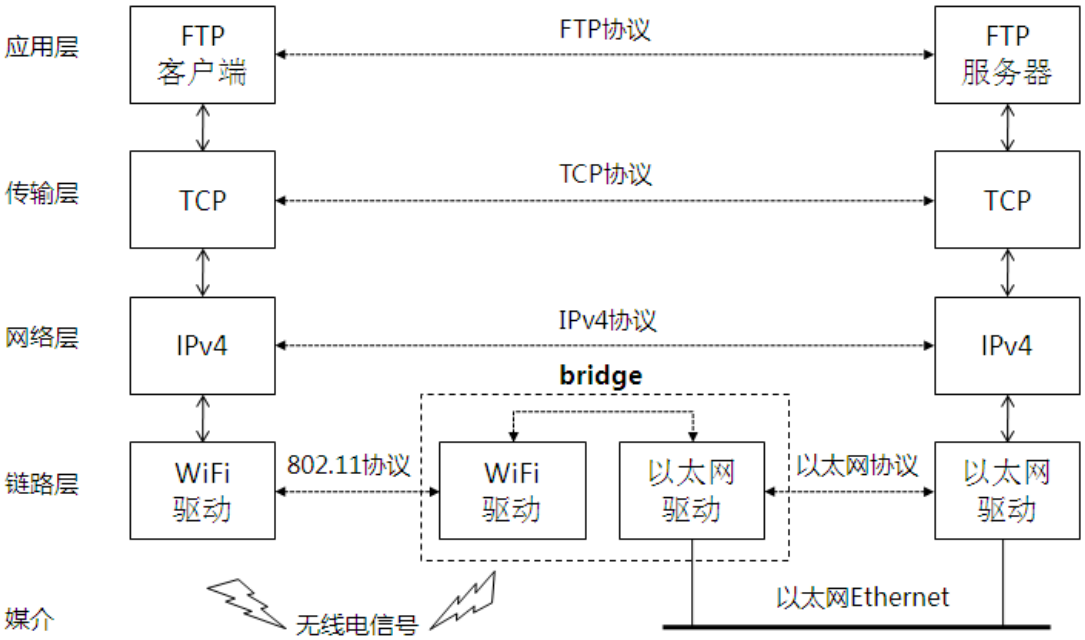


图1.4　通过网桥连接2条链路

图1.3表示的是通过路由器连接2条链路，图1.4则使用网桥连接2条链路（结合图1.2，在网络层看来这2条链路实际上是1条链路），路由和网桥的区别在于网桥不具备网络层协议处理能力，而路由器可以，这也是路由器为何比较昂贵的原因所在。如果网桥连接的链路类型一致（媒介一致），如都是以太网类型，对于这种以太网网桥，我们通常称为以太网交换机*switch*：

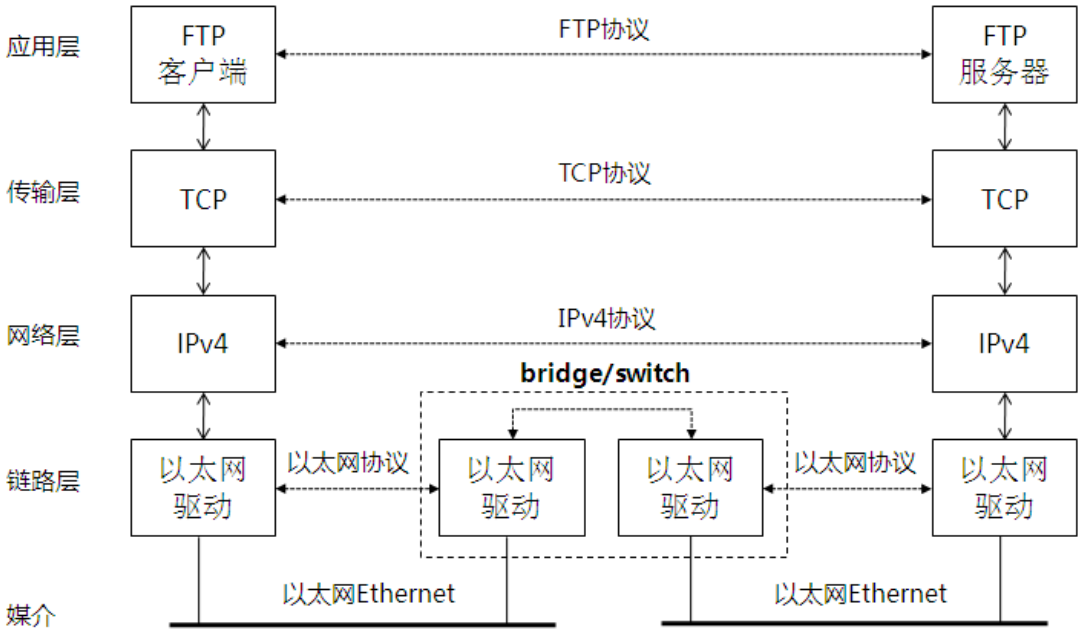


图1.5　使用以太网交换机连接两条以太网链路

在图1.3中，路由器连接着2条不同类型链路：WiFi和以太网，虽然例子中只有2台主机在通信，但实际上WiFi链路上的所有主机可以和以太网链路上的所有主机通信。从图1.3中我们也可以把2台通信主机称为端系统（*end system*），路由器称为中间系统（*intermediate system*）。应用层和传输层都使用端到端（*end-to-end*）协议，这2层也只需要部署在端系统中。网络层则是提供一种逐跳（hop-by-hop）协议，需要部署在所有端系统和中间系统上。

在TCP/IP协议族中，IP为数据源到数据目的提供的是不可靠传输服务（**无法确认数据是否到达目的或已经被丢弃**），这种尽力而为的传输模式没有任何保障数据一定可以从源传输到目的。而TCP则在IP之上提供可靠的传输服务（**对数据到达或丢失进行明确确认**），TCP是如何在不可靠的IP上提供可靠服务呢？它靠的是发送计时、等待接收确认、等待超时重发机制。TCP和IP的这种区别反映传输层和网络层各自不同的使命。

从图1.3中发现路由器连接拥有多个网络接口，但并不是拥有多个网络接口的设备都可以成为路由器，拥有多个网络接口的设备称为“多宿*multihomed*系统”，但如果路由器和多宿系统的根本区别在于：路由器可以将1个网络接口接收到的数据从另外1个网络接口转发出去。在网络中路由器也不需要是特殊的设备盒子，**多宿系统经过一些操作系统设置就可以变成路由器**。所以在称谓中，我们把网络角色分为主机、路由器、网桥，而“系统”则是它们的统称。

网络互连的目的之一就是为应用程序屏蔽各种物理层次细节，在图1.3~1.5中，尽管这种屏蔽效果不是那么明显，但我们可以发现应用层完全不在乎主机之间是否在同1条以太网链路还是在同1个无线频段中，也不考虑传输路径中存在1台路由器还是n台路由器。在图1.3~1.5中，路由器只有1台，但在当今规模的互联网中，两个应用程序需要传阅的路由器可能超过10台甚至是20台。正是这种屏蔽才使得网络互连变得强大而有用。

3. TCP/IP协议族分层

在前面介绍分层我们已经知道了一些协议，但TCP/IP协议族在不同层次中的协议还有很多，如图1.6所示。

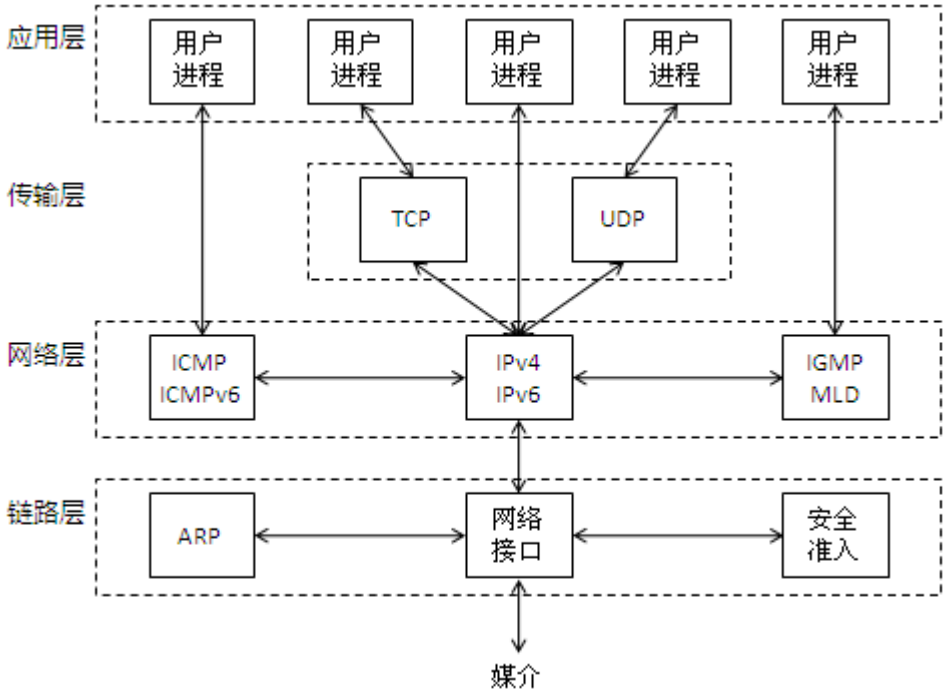


图1.6　TCP/IP协议族在不同层次的各种协议

TCP和UDP在传输层是占绝对多数的协议，它们都使用IPv4或IPv6（统称IP）作为网络层协议。尽管IP协议服务是不可靠的，TCP通过自身的机制实现了可靠传输，在**后续章节**会对TCP如何工作进行分析。

尽管TCP提供了可靠传输服务，但并不是所有用户应用程序都使用TCP作为传输层工具，UDP也有不少市场份额，UDP传输的数据单元被称为数据报*datagram*，1份数据报通常就是用户进程的信息单元。UDP和同层TCP最大不同在于UDP的传输服务是不可靠的。UDP和TCP在应用程序的分布上也有很大区别，更多面向用户的应用使用TCP，而UDP则衍生出了面向网络服务的应用。在**后续章节**也会对使用UDP的应用进程进行分析。

在网络层，IP是当仁不让的主力协议，所有的TCP、UDP信息在穿越网络时都需要被端系统和中间系统的IP层进行处理，**IP是TCP/IP协议族的核心，它的作用是为TCP/IP网络提供统一的地址格式和配套的寻址（寻路）服务**。图1.6中可以发现有的进程可以越过TCP、UDP，直接工作在IP上，虽然比较奇怪，但也并不罕见，如OSPF就是如此。如果有兴趣和能力的话，我们也可以在IP上实验一些新的传输层协议。IP协议目前分为两个版本——版本4和版本6，分别称为IPv4和IPv6，二者互不兼容，IPv4占据目前互联网的99.99%，而IPv6则拥有互联网的未来。在第三章中，我们会对IP进行详细介绍。

ICMP附属于IPv4、ICMPv6则附属于IPv6，它们通常用在端系统、中间系统之间传输IP差错控制和管理信息，端系统和中间系统可以通过它获得网络上存在的一些异常或错误。在**后续章节**会对ICMP及ICMPv6如何工作进行分析。

IGMP（Internet Group Management Protocol互联网组管理协议）附属于IPv4、MLD（Multicast Listener Protocol组播监听协议）附属于IPv6，它们用于组播服务之中，在**后续章节**会对组播及IGMP、MLD如何工作进行分析。

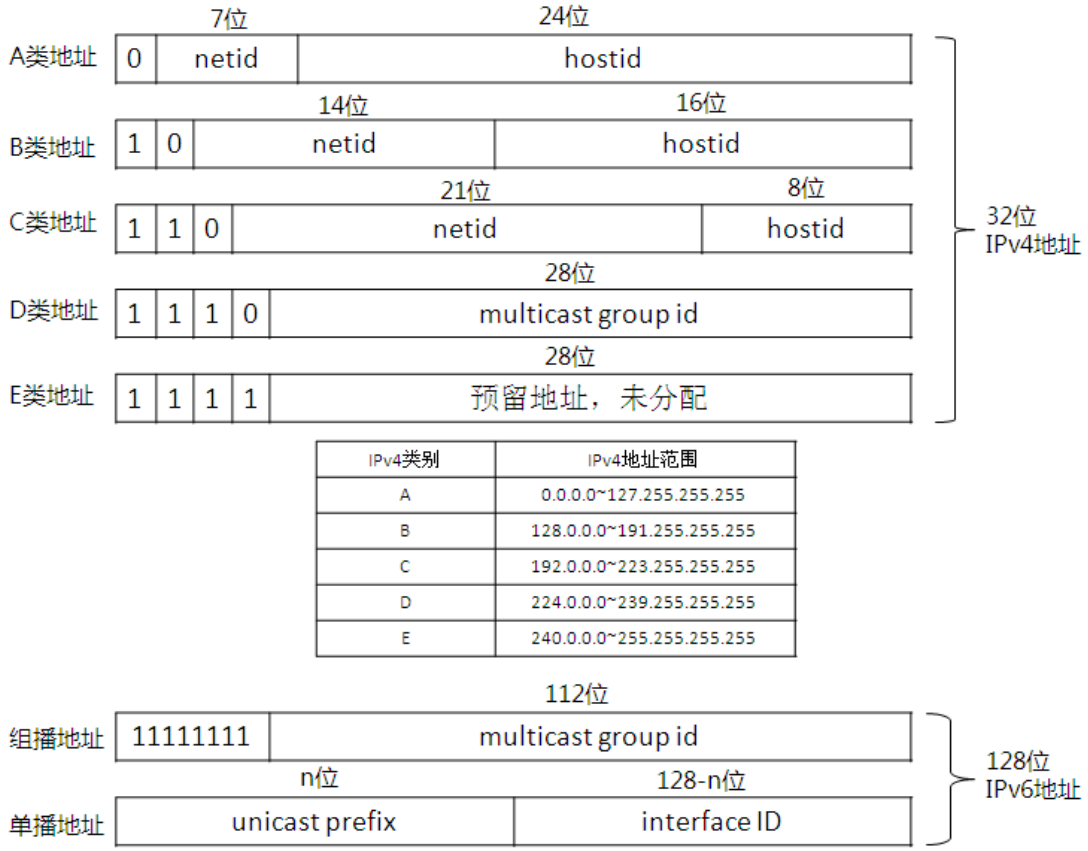
ARP（Address Resolution Protocol地址解析协议）附属于网络接口，ARP用于网络层地址和链路层地址之间的解析，在TCP/IP中，ARP主要为IPv4服务，它还有孪生协议RARP（Reverse Address Resolution Protocol逆向地址解析协议），但目前已经很稀有了。在**后续章节**会对ARP如何工作进行分析。

安全准入也是附属于网络接口，互联网的设计初衷是自由互联，但也由此带来一些安全问题，如何有效控制合法计算机连接到网络已经成了一项很重要的内容，安全准入正是这方面技术的统称，在**后续章节**会对安全准入如何工作进行分析。

4. 互联网地址（IP地址）

就像交通网络为每条公路、村庄、城镇命名一样，在TCP/IP世界中，要为每个接口分配统一格式的地址，这个地址就是互联网地址，也叫IP地址。为公路、村庄、城镇命名是为了制作交通地图，地图的作用是方便我们出行时规划交通路线。TCP/IP中的IP地址可以用于制作互联网地图，在这个地图上给定一个IP地址，规划出前往这个IP地址的网络传输路径；在TCP/IP中根据目的IP地址确定传输路径的过程我们称为“寻址”或者“路由”。

在TCP/IP中，我们为每个网络接口都会分配一个唯一的IP地址，IP地址在IPv4和IPv6并不一样，如图1.7所示。



IPv4类别	IPv4地址范围
A	0.0.0.0~127.255.255.255
B	128.0.0.0~191.255.255.255
C	192.0.0.0~223.255.255.255
D	224.0.0.0~239.255.255.255
E	240.0.0.0~255.255.255.255

图1.7　IPv4和IPv6地址分类

IPv4采用的是32位地址，为了方便表述，IPv4常常使用“点分十进制”表示，如一个C类地址202.101.172.35。在设置之初，IPv4地址被分为5类，图1.7中列出了每类地址范围。

IPv6地址采用的128位地址，这是为了满足日益枯竭的IPv4地址资源，IPv6地址采用“：分十六进制”表示，如1:2:3:4:5:6:7:8。IPv6地址并没有采用IPv4那种复杂的分类，总体上只分为组播地址和单播地址2类，内部结构则是动态的。

由于地址是分配给接口的，所以拥有多个接口的多宿系统会拥有多个IP地址——每个接口一个。由于每个接口拥有的地址是唯一的，所以必须由全球统一的地址分配机构统管全球IP地址的分配。这个机构叫“互联网信息中心”（*Internet Network Information Center*），简称为InterNIC，InterNIC处理各个组织的地址申请，然后分配IPv4地址中的netid和IPv6的unicast prefix部分，hostid和interface id由各个组织内部自行分配。

地址根据使用类型可以分为单播、组播两类，IPv4中还有特殊的广播地址（广播地址也是一种变相的组播），在**后续章节**会对IP地址获取、子网划分、路由和组播如何工作进行详细分析。

5. DNS域名系统

我们从上一节得知IPv4地址是32位长，IPv6地址则是128位长，在实际使用过程中，要记住这些

数字串不是那么容易的事，幸好，我们有DNS（*Domain Name System*），它的作用是将一个人熟记的域名（Domain Name）和特定的IP地址绑定起来，在使用过程中只需要记住域名即可，应用程序会自动根据用户提供的域名到DNS中查找绑定的IP地址，再与该IP地址进行通信。和InterNIC一样，DNS也是由全球统一机构管理的，以确保域名和IP地址的唯一绑定关系。
DNS除了提供根据域名查找IP地址服务外，还能提供根据IP地址查找域名的逆向服务。在**后续章节**会对DNS如何工作进行详细分析。

6. 数据封装

早起的通信系统是建立在书信基础上的，人把要表达的意思写在纸上，再把纸装到信封袋子里，把信封交给邮差，邮差完成投递，收信人要把信封拆开、看纸上写的内容，最终完成信息的接收。
在TCP/IP中，应用程序数据就相当于信纸上的字，应用程序数据在TCP/IP协议层从上往下传递过程中，每一层都会加上一些封装头部以实现信息在层与层之间对等交流，封装头部就相当于信封，如图1.8所示。

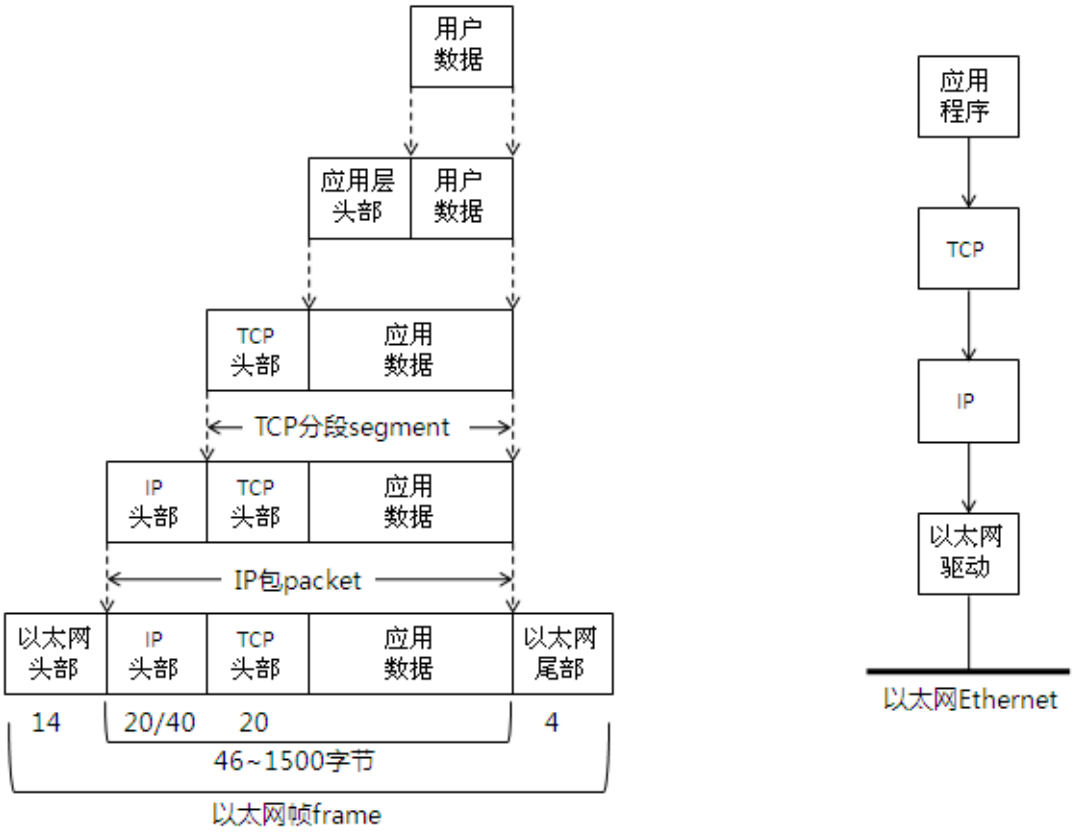


图1.8　协议栈从上往下的封装示意

1. 用户数据是用于人与人之间通信的，要在应用程序中传递，要先让应用程序进行处理变成应用数据；
2. 应用数据交给TCP层后，TCP会封装TCP头部，添加上TCP头部的应用数据被称为TCP分段*segment*，TCP分段是TCP层数据单元；
3. TCP分段交给IP层后，IP层会添加IP头部，添加IP头部的TCP分段被称为IP包*Packet*，之前也有把IP包称为数据报*datagram*的，由于数据报是UDP的称谓，为了以示区别，IP层数据单元统称为IP包；
4. IP包交给以太网驱动层后会被添加上以太网头部和尾部，称为以太网帧*frame*，由于链路层不止以太网一种，所以“帧”是链路层的数据单元，帧最后会编码成电流信号在以太网媒介中传递；
5. 从图1.8的下部可以得知各个封装头部采用的长度，如以太网头部长14字节，IPv4头部长

20字节，IPv6则达到40字节，TCP头部20字节，以太网尾部4字节，其中以太网头部和尾部之间数据长度最短为46字节，最长1500字节。

在图1.8中使用TCP做举例，我们也可以画出UDP封装的示意，差别在于UDP头部长度为8个字节，封装UDP头部的应用数据称为UDP数据报*datagram*。从图1.6可以得知TCP、UDP、ICMP、ICMPv6、IGMP、MLD都需要把数据发送给IP层处理，那么IP层如何区分上层数据是TCP、UDP还是ICMP发送的呢？它必须要有个标识字段用于区分上层协议，在IPv4头部有个字段叫*protocol*，IPv6中叫*next-header*，8位长，用于区分上层协议，如protocol=1表示上层是ICMP、6表示TCP、17表示UDP；next-header=58，表示IPv6封装上层为ICMPv6，6和17分别表示TCP和UDP。

类似的许多应用程序可以同时使用TCP或UDP，TCP、UDP也必须有相应标识来区分不同的应用程序，TCP和UDP都使用16位长的端口号*port-number*来区分，TCP、UDP头部都有源端口、目的端口两个字段。

网络接口也同时需要为ARP、IP和安全准入服务封装帧，以太网中用于区分上层协议的字段是16位长的类型*type*字段，如0x0800表示IPv4，0x86dd则表示IPv6。

7. 分用Demultiplexing

当目的主机的从以太网驱动接收到以太网帧时，开始从TCP/IP协议栈的底部往上走，每过一层协议就会将封装的头部拆除，每个协议都会检查封装头部的标识字段，以确定将解除封装的数据单元交给哪个协议，这个逐层解除封装的过程叫就做分用*demultiplexing*，上一节描述的逐步封装的过程称为复用*multiplexing*。图1.9解释了分用的过程。

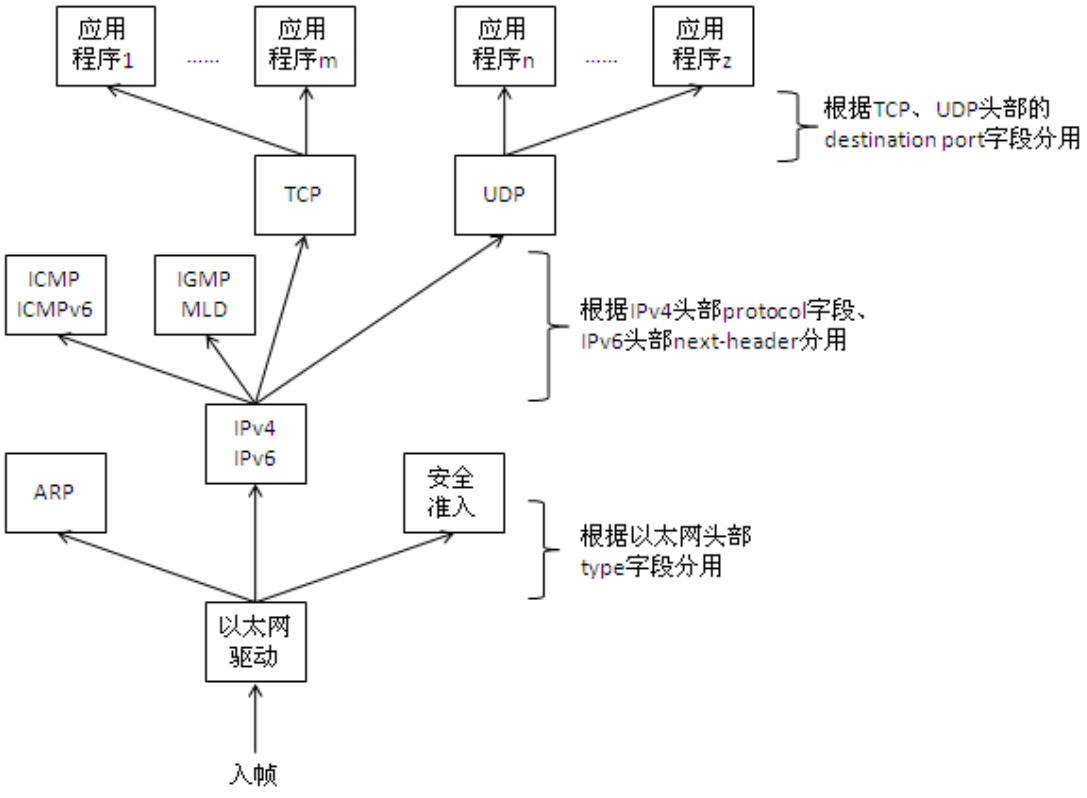


图1.9　以太网帧的分用过程

放置ICMP/ICMPv6、IGMP/MLD的位置对协议研究来说是一项挑战，在图1.6中，它们和IPv4、IPv6是同一层的附属协议，在图1.9中却变成上层协议，在协议栈角度它们的确和IPv4、IPv6属于网络层，但研究协议封装和解封装时，我们要意识到ICMP/ICMPv6、IGMP/MLD都是采用IPv4/IPv6封装的，因此放在IPv4/IPv6的上半层。

同样的还有ARP、安全准入与以太网驱动的关系，ARP和安全准入也都是采用以太网头部封装。而在后续章节介绍ARP时，我们会发现ARP和IPv4的关系，这也是为什么把ARP放置在

IPv4和以太网驱动之间的原因。需要我们认识的一点是，没有图能够完美地诠释所有协议之间的关系。
当我们详细介绍TCP、UDP时，我们会发现TCP、UDP的分用其实是基于destination port（目的端口）、source port（源端口）和source IP address（源地址），而不只是的目的端口。

8. 客户端-服务器（*client-server*）模型

大多数网络应用程序都是假设通信的一端是客户端，另外一端是服务器，服务器提供某些服务，等待客户端的访问。目前流行的BT、电驴等应用则跳出了客户端-服务器模型，许多客户端之间可以直接通信，这种模式称为对等（*peer-to-peer/P2P*）模式。
我们可以把服务器分为2类：重复*iterative*式和并发concurrent式，重复式服务器遵循如下步骤：

1. 等待客户端请求；
2. 处理客户端请求；
3. 向发送请求的客户端回应处理结果；
4. 回到步骤I1。

重复式服务器的问题在于I2步时间会比较久，此时其余的客户端请求都处于等待阶段，无法被处理。
而并发式服务器的步骤如下：

1. 等待客户端请求；
2. 启动一个新的服务器进程处理客户端请求，这占用更多的操作系统资源，当服务器处理完请求并给客户端以响应，该新服务器进程会终结，以释放资源；
3. 回到步骤C1。

并发式服务器的好处在于服务器催生新的进程来处理每个客户端的请求，也就是说每个客户端都有自己的服务器进程，因此并发式服务器可以同时处理多个客户端请求。
为何不在客户端进行分类呢，因为客户端并不能区分到底是与重复式服务器通信还是并发式服务器通信。
通常，TCP服务器都是并发式的，UDP服务器是重复式的，但也有一些特例，在后续章节介绍UDP服务器时会详细描述。

9. 对等（*Peer-to-Peer/P2P*）模型

对等模型是近几年兴起的一种通信模型，主要的通信是在客户端之间，服务器维护客户端的状态信息，为客户端之间通信牵线搭桥，如图1.10所示。

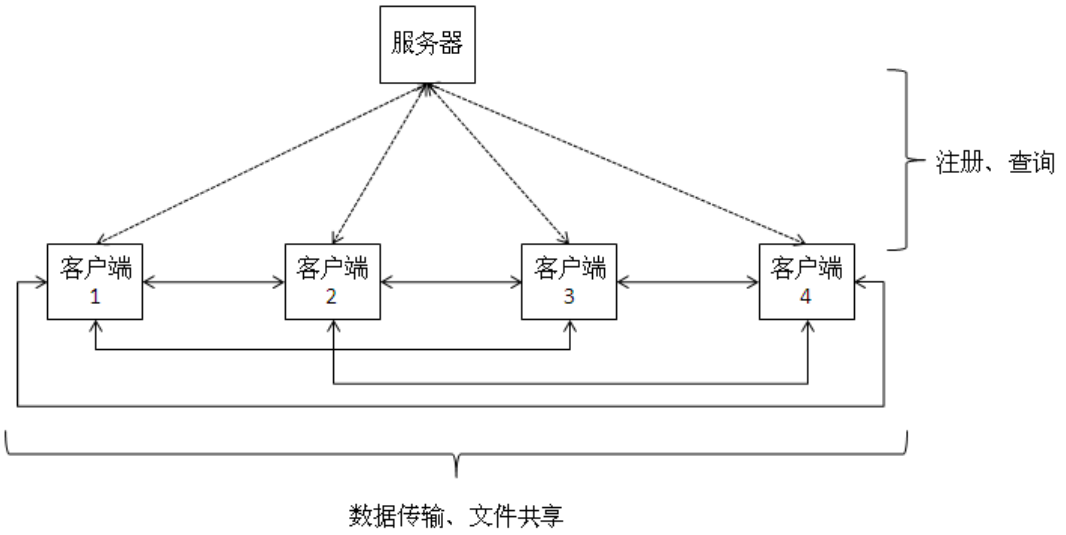


图1.10　对等通信模型
在对等通信中，服务器收集所有客户端的注册、共享文件列表信息，当客户端1需要某个文件时

会向服务器进行查询，服务器返回拥有该文件的客户端信息，这些客户端信息可能不止1个，如客户端2、3、4都拥有该文件，那么客户端1可以同时向客户端2、3、4发起数据请求，客户端2、3、4同时向客户端1传输数据。
在对等模型中，服务器的数据传输工作被许多客户端分担了，只需要进行一些简单的状态维护和检索，而客户端可以将同时从多处下载同一个文件的不同分片，再进行文件重组，传输效率也得到了极大的提升。使用这种模型的应用代表是*BitTorrent*、*eDonkey*等文件共享工具，另外还有一些在线视频网站也采用这种模型以提高传输效率。

10. 端口号

前面介绍分用时提到了TCP和UDP使用16位的端口号区分不同的应用程序，那么这些端口号是怎么确定的呢？
通常服务器使用所谓的熟知*well-known*端口，如FTP服务使用TCP 21端口、Telnet使用TCP 23端口、TFTP则使用UDP 69端口。TCP/IP中熟知端口号范围是1~1023，这些端口由互联网号码分配委员会（*Internet Assigned Numbers Authority IANA*）负责管理。
而客户端却不理睬使用什么端口，它只需要确认它使用的端口在主机上是唯一的，没有和别的应用程序冲突即可。客户端使用的端口被称为临时*ephemeral*端口，因为只有用户运行客户端程序的时候才需要一个端口，而服务器则一直需要开着端口等待客户端访问。
大部分操作系统给客户端预留的端口范围是1024~5000，大于5000的端口也是给一些服务器使用的，比如SIP使用的是5060端口。
在一些操作系统如Unix有个概念叫做保留*reserved*端口，只有一些拥有超级用户权限的进程才能使用预留端口，在Unix中，预留端口范围和熟知端口范围一致，从1~1023，其实从熟知端口的历史来说，它脱胎于Unix的保留端口。

11. 标准化过程

TCP/IP是一个开放的协议族，任何厂家和开发者只要遵守协议规范，都可以实现TCP/IP互联，那么这里有个疑问，TCP/IP的协议规范是由谁来制定呢？请看图1.11。

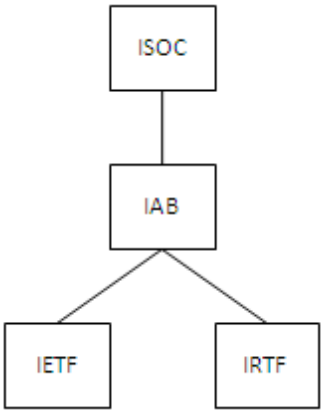


图1.11 TCP/IP标准化组织

1. ISOC（*Internet Society*）互联网协会的责任是支持并推动互联网不断向前发展，它视互联网为全球通信研究基础设置；
2. IAB（*Internet Architecture Board*）互联网架构委员会是一个技术监督和协调机构，由全球选拔的15名不同行业的志愿者专家组成，他们对互联网标准进行最后的编辑和技术审核，IAB率属于ISOC；
3. IETF（*Internet Engineer Task Force*）互联网工程项目组是一个面向近期标准的组织，它分为若干个领域（主要为应用、编址与路由、安全等，领域数量有可能会发生变化），IETF制定的规范往往会成为互联网标准，也就是我们常说的RFC（*Request For Comments*），IETF还有一个专门的组织叫IESG（*Internet Engineer Steering Group*）互联网工程指导小组负责帮助IETF主席进行标准化制定工作；

4. IRTF（*Internet Research Task Force*）互联网研究项目组和IETF正好相反，他们对互联网的长远发展进行研究。

IETF是距离所有互联网工程师和使用者最近的组织，他们制定的RFC是当今所有互联网应用和TCP/IP协议族都遵从的统一规范。IETF和IRTF在标准化组织里都率属于IAB。

12. RFC

在网络互连领域中，正式的标准通常是以RFC的名义进行发布，另外还有许多RFC并没有成为正式标准，而只是为了提供一些参考信息。RFC编号从1开始，数字越大表示该RFC越新，如RFC3513就要比RFC791新很多。RFC对内容的长度也没有要求，有的短至几页，有的超过了300页。

所有的RFC都可以在IETF的官方网站上获取，地址是www.ietf.org。

最新的RFC往往是搜索一些信息的入口点，因为最新的RFC会告诉你它更新了哪个RFC或替代了哪个RFC。下面是一些重要的RFC：

1. RFC1700，已分配的号码（*Assigned Number*），制定了互联网协议中的数字和常量，该RFC发表于1994年，作者是Reynolds和Postel，互联网中所有熟知的端口号都在该RFC中列出；
2. RFC1610，互联网正式协议标准（*Internet Official Protocol Standards*），作者也是Postel，该RFC叙述了各种互联网协议的标准化状况，每个协议的状态包括：标准、草案标准、提议标准、实验中、信息参考、历史标准，每个RFC都会注明当前的状态；除了状态，每个RFC还有需求级别：必须的、推荐的、可选择的、限制使用和不推荐；和RFC1700一样，RFC1610也有可能会被更新，请查看该RFC的开头部分的更新、替代信息；
3. 主机需求（*Host Requirements*）RFC，由RFC1122和RFC1123共同定义，它制定了一个互联网主机必须具备的一些特性，RFC1122对链路层、网络层和传输层进行了处理，RFC1123则处理了应用层，这2个RFC分别被许多RFC所更新，如RFC5966（在TCP实现DNS传输的要求）对RFC1123进行了更新，如果要对各层协议有更细致的研究，从这两个RFC出发将会是一个不错的选择，这2个RFC也列出了协议各个特性、实现细节的级别，如“必须”、“应该”、“可能”、“不应该”、“绝对不能”等；还有专门一个RFC1127，对这2个RFC开发过程中的讨论和结论进行了非正式的总结；
4. 路由器需求（*Router Requirements*）RFC，由RFC1009定义，当时路由器的名字还不叫*Router*，而是*Gateway*，该RFC和主机需求RFC类似，但是多了一些路由器独有特性的定义。

13. 网络互连与互联网

在图1.3中介绍了使用路由器连接两个网络实现网络互连，第1.4和1.10节中介绍了互联网地址和熟知端口号，它们是否有区别和联系呢？网络互连和互联网是两个概念，它们的英文单词却是一样的，网络互连叫*internet*（i是小写字母），而互联网则叫*Internet*（I是大写字母）：

1. internet指的是多个网络通过相同的协议互连，TCP/IP是网络互连使用最广泛的协议，所以大部分时候，internet指的就是使用TCP/IP连接多个网络，这个网络可大可小，可以是公有的也可以是私有的；
2. Internet特指通过TCP/IP协议实现全球网络互连的所有主机和网络设备，所以互联网Internet是网络互连internet的子集。

14. 测试网络

图1.12是本书中举例使用的测试网络：

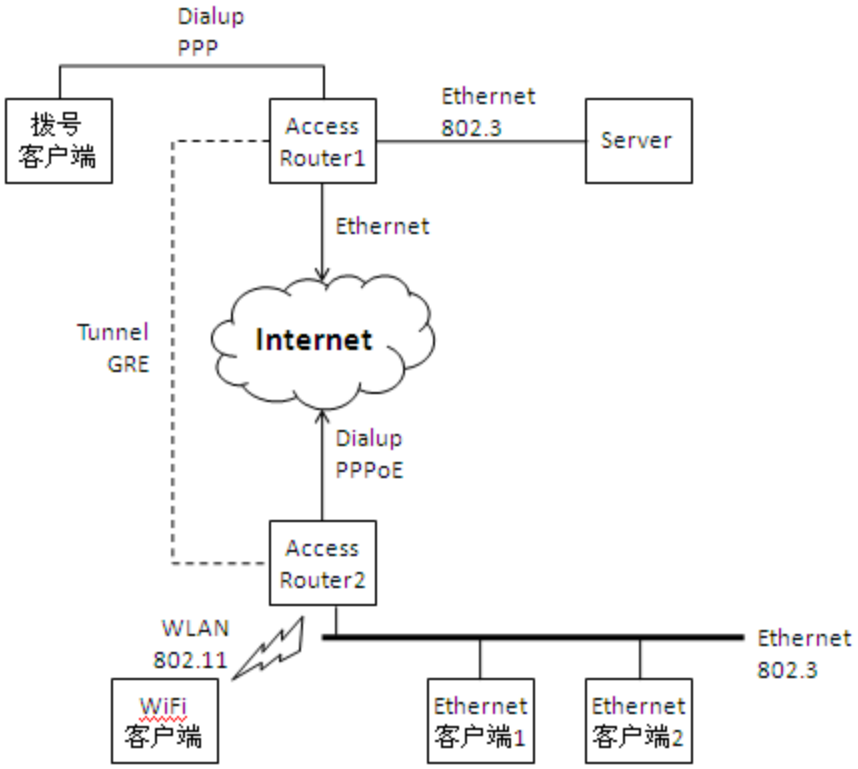


图1.12　测试网络

图中的互联网云表示整个互联网，而测试网络则是互联网的一部分，图中对每条网络链路使用的链路层协议进行了说明，还有一些复杂的链路层协议会在高级网络知识中提及。

15. 小结

本章对TCP/IP协议族进行了一个快速浏览，在后续的章节中会介绍更多术语和协议。TCP/IP协议族的4层分别是：链路层、网络层、传输层和应用层，对每一层的作用我们都做了专门介绍。网络层和传输层的根本区别是：网络层（IP）提供逐跳（hop-by-hop）服务，而传输层（TCP、UDP）提供的是端到端（end-to-end）服务。

网络互连internet是若干个网络通过某种共同协议连接，IP是最常见的网络互连协议，而互联网Internet是全球范围内使用IP互联而成的网络，它目前可能包含几十亿台电脑、手机、路由器、交换机，每一时刻都有几亿人通过这些设备在线自由地通信，互联网是有人类史以来最伟大的沟通方式。

TCP/IP网络互连需要使用接口连接到网络中，每个接口都有专门的IP地址，IP地址通常和电话号码一般难以记忆，也正如人们通过人名来保存电话号码一样，TCP/IP发明了主机名（hostname）来绑定IP地址，TCP/IP还有专门的DNS协议来解析指定主机名对应的IP地址。我们还介绍到了TCP、UDP的端口号，用于区分不同的应用程序，而服务器通常使用熟知well-known端口，而客户端则使用临时端口。

练习

1. 在图1.3、1.4、1.5中分别有几个network（链路）？在测试网络图1.12中又有几个？
2. 检查你PC的操作系统类型。
3. 查看你的PC或者手机有几个网络接口，分别什么类型，连接到什么网关？
4. 如果能看到PC的网关，检查一下这个网关的型号，有什么接口？
5. 查看你的PC或手机上的IP地址，分别属于哪个接口？

6. 检查一下和你的PC相同链路上还有没有其它PC？查看一下这些PC的操作系统类型和IP地址。
7. 如果你的PC同时拥有WiFi和以太网接口，它现在是多宿（multihomed）系统吗？如果不是，怎么让它变成是？
8. 看看你的PC或者手机是否可以连接到互联网，打开浏览器，是否可以登录www.ietf.org，并查看RFC1000，学习一下RFC这个术语是怎么产生的。
9. 如果可以登录互联网，打开你的浏览器，搜索引擎中输入常用端口或者well-known ports看看能搜索出什么结果。
10. 网桥是否是中间系统？判断一个设备是否是中间系统的依据是什么？
11. 使用netstat命令查看一下当前PC使用了哪些端口。
12. 通过netstat命令是否可以判断你主机名？是什么？
13. 尝试到www.wireshark.org下载并安装wireshark软件。