第2章　TCP/IP网络互联简介

本章简单介绍了TCP/IP网络互联，总结了TCP/IP的基本理论和实践基础。主要侧重于当前的IP版本 —— IPv4及其工作方式，其中包括IP寻址和IP头。对于不熟悉TCP/IP的读者，本章可作为对TCP/IP的浓缩介绍；而对于那些有这方面经验的读者，本章可作为一个整理思路的过程。

2.1　网络互联问题

简单的网络把两个或更多的计算机用同一网络媒体连接在一起，网络媒体可以是线路、无线频率或任何其他通信媒体。对此网络中的每个系统都必须唯一标识，否则一个系统无法与另一系统通信：除下面的注释所提到的传输之外，所有传输都必须明确地寻址到一个特定系统，且所有传输都必须含有可识别的源地址以便其响应（或出错报文）能够正确地返回发送者。

广播和组播

有时候某些传输可以一次寻址到多个目的系统。这种传输可能是网络中所有系统均可接收的广播(broadcast)，通常用于管理目的。广播报文使用特殊的广播地址作为其目的地址，而网络中的所有主机都要侦听来自广播地址的报文。

另一种可以被多个系统接收的地址类型称为组播(multicast)。如果某个系统预订了某个组播地址，该系统将侦听发给该组播地址的传输数据。对于有多个系统感兴趣且只有这些系统感兴趣的信息，就可以使用组播地址作为其目的地址。换句话说，那些没有预订组播地址的系统将不会注意这些组播传输。

在一个简单网络中可以用以下几种方法为主机设定地址：

- 从1(或其他数字)开始，对所有主机连续编号。
- 为每台主机随机指派地址。
- 每台主机使用一个全球唯一值。

以上每种方法均有其缺点。如果该网络不与其他网络合并，则为主机连续编号的方法没有问题。但实际上，各部门间的网络经常需要合并，整个机构也是如此。而使用随机地址的方法则带来了特定网络中或合并的网络间的唯一性问题。最后，每台主机使用全球唯一值的方法解决了各种环境中的地址重复问题，但需要一个中央授权机构来发放地址。

主机、节点和路由器

不同的硬件系统可以通过IP网络连接起来，这些硬件系统包括：

- 节点：即实现IP的任何设备。
- 路由器：即可以转发并非寻址到自己的数据的设备。换句话说，路由器可以接收发往其他地址的包并进行转发，这主要是由于路由器连接着不止一个物理网络。
- 主机：即非路由器的任何节点。

实际上，对于绝大部分网络接口设备，有一个授权机构来确保每个接口设备制造商使用自己的地址范围，从而可以保证每个设备具备一个唯一号码。这意味着网络中的数据可以直接定向到与网络中每个系统使用的网络硬件接口关联的地址。

这就从根本上解决了简单网络中的问题：如果一个系统欲向其他系统发送数据，它只需要将目的主机与目的主机的网络地址关联，创建包含待发数据的网络传输单元，然后通过自己的网络接口传送。不论网络媒体使用什么机制来交付数据，目的主机都能接收到。

增加复杂性

上述网络类型 —— 局域网(LAN)在本地网络中可以很好地工作。换句话说，只要所有主机都连接在同一网络媒体上， LAN将工作得很好。在实践中，这意味着单个网络中能够连接的主机数量有一个上限。这个上限通常与媒体的一些物理特性有关，例如：网络中能够承载的数据容量的最大值(带宽)、物理电缆两端间的最大传输距离等。总之，局域网通常局限于连接同一建筑物或小型校园中的数百台主机，无线网络或一些使用卫星技术的网络可以有更大的范围但仍将受限于其带宽。

随着个人计算机在许多企业内的普及，那些超过数百名员工的机构或者人数很少但不只一个建筑物或有多个分支部门的机构，发现局域网并不足以解决其连网问题。将网络 (例如部门或分支机构网络)链接为一个机构互联网络的方法成为必不可少。

如果企业中的所有网络都是同一种类型，如以太局域网，则网络互联的实现很容易。连接局域网的方法之一是使用网桥：网桥将侦听两个网络上的业务流，如果发现有数据欲从一个网络传送到另一网络，它将该数据重传至目的网络。但是，链接较多局域网的复杂的互联网络很难处理：要求链接 LAN的设备能够了解每个系统的地址和网络位置。即便是同一地点和同一网络上的系统，随着系统数量的增加，也将导致对业务流进行跟踪和选路的任务变得非常艰难。

当然，这种情况要求指定地点的所有网络都使用相同的媒体。实际上，其部门已长期使用网络的机构往往发现其网络上不止一种网络媒体，通常包括以太网、令牌环和其他媒体。各种网络媒体上传输的数据在格式上很可能有这样或那样的不同，这就意味着如果连接在不同网络上的系统要进行互操作，则在发送之前要了解目的地的网络类型并按照对方要求的格式来构造数据。此外，还需要一些中间系统对数据进行正确选路，并在必要时把数据转换为正确的物理格式，以适于在可能差别很大的网络媒体上传输。

试想一下一个具有许多不同分部、分支机构和部门的大企业的情形。每个 LAN都需要了解企业中任何地方的结构变化，从而正确地为数据选路。试想一下如果不是企业而是政府机关遇到这种问题将会是多么地难以处理，而当网络互联扩展至企业之间以及成为众所周知的全球Internet后问题则变得更加严重。

上述解决方案过于简单，不足以解决任意大型网络上的选路问题，更不用说解决 Internet的问题。人们很需要一个不同的解决办法：它必须能够使连接在不同网络上的不同系统，彼此之间只需知道对方的互联网地址就可以进行无缝互操作。下一节将介绍这种解决方案。

2.2 分层网络互联模型

上述连网模型假定所有网络通信发生在连接到网络的系统之间，但并没有指出这些系统

间是如何通信的。换句话说，它假定所有数据简单地按照本地网络的格式在接口上传送，而并没有讨论这些数据的格式如何。通过详细说明数据如何在使用它的个人或程序间进行转移，可以把无缝互操作的问题分解为更易于管理的部分。

当数据从一个系统传输至另一系统时，其分离的过程模型通常称为协议栈。该协议栈被用在不同层中。协议的实现也称为协议栈，它表示数据将在哪一层处理以及数据如何在相邻上下层间传递。

2.2.1 OSI模型

开放系统互连(OSI)通常作为基本参考模型，最初用于表示网络互联的通用模型。如图 2-1所示，它的七个层表示互操作系统间通信的不同级别。自下而上，这些层包括：

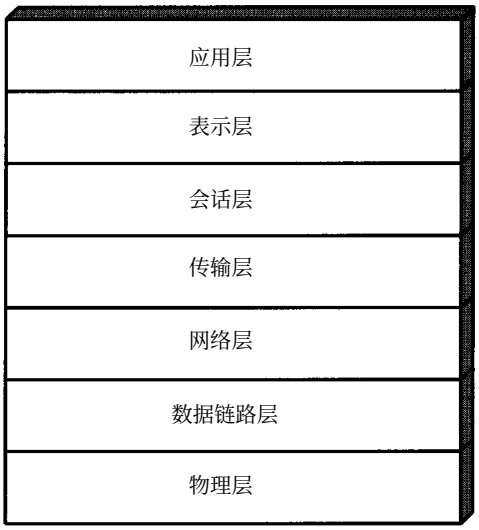

图2-1 网络互联的OSI模型提供了系统在网络上进行互操作的7个不同层

- 物理层。代表数据转移时的真正媒体。系统通过物理层彼此间发送原始电脉冲或其他合适的信号。在这一层，系统间的通信通过与物理媒体的连接得以实现。
- 数据链路层。增加了协议，用于解释物理媒体上传输的数据，其中包括可靠性和重传等功能。在这一层，系统间的通信通过直接连接到网络的实际网络接口来实现。
- 网络层。提供协议使得系统之间可以通信，它把系统而不仅仅是网络接口连接到一起。正是在这一层，通信被认为发生在系统间而不只是在网络接口间。这一层需要考虑如何在位于两个不同网络的两个不同节点间传送数据。
- 传输层。提供协议使得一个系统的进程连接到另一个系统的进程成为可能。换句话说，在这一层，运行在一台主机上的两个不同程序可以各自连接到不同主机上运行的不同程序。
- 会话层。处理连接的流和定时。正是在这一层，管理连接的实际结构——不论发送方是否在发送数据而接收方是否在接收数据。
- 表示层。在这一层，不同的系统将自己的数据翻译为彼此都能接受和理解的格式。在完全不同的系统上运行的程序必须使用所有系统都能理解的标准格式，而这种翻译就发生在这一层。
- 应用层。定义实际程序如何使用网络交互。例如，某个网络程序的应用协议可以定义来自用户的输入类型或远端设备响应的输出类型。

2.2.2 Internet模型

那些构造实际网络的网络互联研究者们发现可以使用只有四层的网络模型来提供所有功能。如图2-2所示，Internet模型把网络的层进行了压缩，使网络互联更简单，因为层越少就意

味着交互越少，自然也就意味着连网实现更加高效。

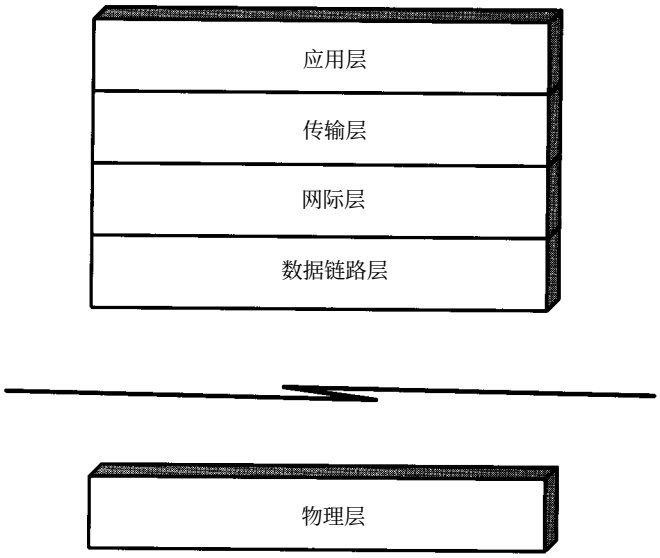

图2-2 Internet模型只用四层实现了无缝的、可互操作的网络互联

虽然在某些情况下这些层看来与OSI分层模型类似，但其中确实有一些差异。从最底层开始，主要的差异首先在于，Internet模型中把物理层作为独立的一层舍弃了。这可能是由于实现者假定在数据链路层发送和接收的数据是由物理媒体传递的。其次，网络层变成了网际层，使得通过网络把系统链接在一起的需求变得更加明显。传输层中包含了会话层的大部分功能，而应用层中则包含了表示层的大部分功能。

理解这些层如何工作将帮助我们理解IP连网是如何工作的，因为在Internet模型中通信系统在哪一层交互更加清晰：

- 数据链路层(又称为网络接口层)。连接在同一网络上的系统彼此之间可以通信。在这一层上通信的系统不一定相同，因为两个不同网络上的系统不能直接在这一层通信，而在其他层通信的系统则要保持一致。
- 网际层(又称为网络层)。系统通信的层次。这一层的数据传输单元在地址信息之后包含一些净荷数据。换句话说，数据可视为仅仅是从源系统发送到目的系统。两个系统可以用多种不同的方法交互，但是至少在这一层，可将来自不同的应用层交互的数据仅仅视为具备相同的源地址和目的地址，而无需立即进行区分。
- 传输层。进程间通信的层次。正是在这一层，两个通信系统间可以具有多个业务流 (参见上一段)。
- 应用层。用户(无论是个人还是程序)间通过网络应用进行交互的层次。

本书主要考虑发生在网际层的事情，而对于其他层只考虑在修改网际层协议后会受到影响的部分。

2.2.3 封装

要理解Internet中各层间的交互方法并实现无缝互操作，有必要先理解“封装”的概念。在某种意义上，如果一块数据以某种方式打包以便传输，这时就发生了连网中的封装。理解

封装在Internet模型中工作方式的最好办法是简单地跟踪协议栈中的流程。

考虑如下示例，一个应用允许一台主机上的客户可以向位于另一主机上的服务器发出查询。从客户端应用开始，用户输入一个查询。在应用层进行封装的第一步是将该应用层的协议数据单元(见注释)中的查询打包。该PDU中包含了数据，并用有关如何处理数据的信息将该数据“包起来”。这些信息包括：远端主机上的目的应用的逻辑名、地址或其他指针，以及下一层(传输层)正确处理该包所需的必要信息。

> 协议数据单元(PDU)特指协议对一块数据打包的方式。不同的协议以不同的名字来指称这一块数据。例如，以太网和其他数据链路层协议称之为帧；IP称之为IP数据报或包。对于通常协议或未知协议，PDU主要指的就是这些数据包。PDU中通常包括头(通常位于PDU的开始有时也可能位于最后)和净荷数据，数据可以在头被去掉后使用。PDU指的是一块数据的命名方式，而不是真正的数据块，该数据块通常被称为报文。

在传输层，简单地将从应用层传递来的包作为位串，并在加上头后交给网际层。进程使用端口来发送和接收数据， TCP/IP的传输层在头中加入了目的端口号和源端口号 (与其他项一起)，并把新打包的数据交给协议栈的下一层——网际层的协议。

网际层软件从传输层接收该报文，查看目的 IP地址，然后决定对该数据如何操作。但不管怎样都将加上包含实际源主机和目的地主机网络地址的网际层头，然后将整个包交给协议栈中的下一层——数据链路层的协议。这一步比较棘手：如果 IP网络软件确定数据的目的地是在同一网络上的另一系统，则在数据链路层将包寻址到目的地。但是，目的地在其他网络的数据仍必须以与源主机在同一物理网络上的某个系统为目的地，该数据没有其他的出路。

上面忽略的一个因素是称为路由器的系统。这是一个多宿主机，它同时连接在两个或多个物理网络上，并通过程序设计为可以将包转发到远端网络上。这意味着当有数据发往远端网络时，IP软件会指定数据链路层以与源主机在同一物理网络的路由器地址作为该数据的目的地址。网际层的源地址和目的地址保持不变，但是如果目的主机在外部网络上，数据链路层的目的地址将与目的主机不同。

现在继续跟踪数据在协议栈中向上传递的过程。当数据链路层报文到达其目的地时，接收系统将去掉其数据链路层头并检查其网际层头。如果该头中的地址与接收主机地址相同，将继续去掉该头并将数据上交传输层。但是，如果目的地址与接收主机地址不同，或者接收主机是一个路由器，将重新对该报文打包并转发至适当的网络。

当传输层获得该消息时，它将去掉头并将净荷上交给适当的应用。应用层在去掉头后对数据进行处理。在数据离开发送方之后直至到达接收方之前，低层操作的协议不对数据中的净荷进行处理。虽然可能有这样那样的完整性检查，除了高层提供的头之外，低层协议无需查看其他部分的数据。这种机制使得连接在不同网络上的不同主机可以进行无缝互操作。只要所有的中间系统能正常操作，且只要两个端系统使用的应用软件可以互操作，系统类型、网络体系结构或系统的物理输出与此无关。

2.3 IP

1981年完成的RFC 791定义了当前使用的IP。但是，从那时起又有许多RFC阐明并定义了IPv4寻址议题、在某种特定网络媒体上运行的 IP以及IPv4的服务类型位(TOS)。感兴趣的读者如果想了解20年前定义的IP协议，可以参考RFC 791。该协议的工作主要是定义了在处理数据

时可以应用的简单规则、帮助处理数据的一组头以及寻址机制。在此进行一些扼要解释。

2.3.1 IP寻址

IP地址体系结构依靠高度结构化的地址，地址空间由其长度 (32位)决定。所有IP地址均包括32位或4个字节，IP领域也常使用术语八位组 (octet)。这些地址被分为不同类，其中定义了如何对地址进行处理。还有一些地址具有特殊含义。

1. IP地址结构

IP地址是等级地址，通常从左到右读，高阶位 /字节即是最高有效位 /字节。举例说明，地址前几位说明地址所属的地址类；前几个字节说明该地址所属的网络。最低有效字节 (或位)将地址限定为特定的主机。这种结构意味着向网络外选路时可以忽略单个主机而只需跟踪整个网络的位置。

32位地址被分为两部分：第一部分是网络地址，第二部分是本地地址。在本地网络外，只有网络地址是重要的；而在本地网络内，因为所有主机都连接在同一个本地网络上，只有本地地址是重要的而网络地址则无关紧要。

IP网络地址分发给多个机构，由机构自己为机构内部主机分配本地地址。这意味着某个特定网络内的本地地址可能没有全部分配出去。这样就削减了总数为232的地址空间的可用地址数。

2. IP地址分类

最初IP地址分为三类：A、B和C，用于为不同类别网络上的主机编号。后来在 IP组播成为标准后又加入了第四类地址，称为D类，但该地址即不能用于单个主机也不能用于特定网络。A、B、C类地址渐渐被称作单播 (unicast)，意味着其中每个地址只标识单个主机，且来自 /发往某个单播地址的数据是从一个主机发往另一个主机的。 D类地址用于组播传输，意味着可以有多于一台的主机接收发给某组播地址的数据，但组播传输仍然是由单个主机发起。

检查IP地址的前几位将有助于对地址进行分类。 IP地址的分类如下：

- A类地址第一(高阶)位为0，网络由后续的七位定义。故第一个八位组用于网络地址而其余的三个八位组用于每个网络中的主机地址。这意味着最多有 2^7即128个网络地址组合，而地址中剩余的24位可用于主机地址，这意味着可以有 2^{24}即16 777 216个唯一主机标识符(真正的最大值会有一点减少，参见后续讨论)。这意味着A类地址可以由第一个八位组的值来确定。任何一个0到127间的网络地址均是一个A类地址。
- B类地址前两位为10，网络由后续14位定义。故前两个八位组用于网络地址而其余的两个八位组用于每个网络中的主机地址。这意味着最多有 2^{14}即16 384个网络地址组合，而每个网络中的主机数不能超过 2^{16}即65 536(真正的最大值会有一点减少，参见后续讨论)。这意味着B类地址可以由第一个八位组的值来确定。任何一个 128到191间的网络地址是一个B类地址。
- C类地址前三位为110，网络由后面的21位定义。故前三个八位组用于网络地址而其余的一个八位组用于每个网络中的主机地址。这意味着最多有 2^{21}即2 097 152个网络地址组合，而每个网络中的主机数不能超过 2^8即256(真正的最大值会有一点减少，参见后续讨论)。这意味着C类地址可以由第一个八位组的值来确定。任何一个 192到223间的网络地址是一个C类地址。
- D类地址前四位为1110。组播中不使用网络地址的概念，因为任何网络上的主机无论是

否在同一网络上均可接收组播。这意味着最多有2^{28}即268 435 456个组播地址组合，而所有组播地址可以由第一个八位组的值来确定。任何一个第一个八位组在 224到239间的网络地址是一个组播地址。

- E类地址前五位为11110。在IPv4地址中保留该地址。

3. 特殊地址

由于有一些网络地址有特殊含义，导致可分配的网络地址的总数进一步减少。下列地址不能分配给实际的网络：

- 第一个八位组是127的地址(如127.0.0.1)定义为回返地址。这个约定是必要的。对于所有发往回返地址的数据，网络栈将视为传输给自己的数据，尽管数据沿网络栈向下传递，并没有真正发送到网络媒体上。这种方法允许主机通过其网络接口与自己通信，这对于测试很有用。
- 地址中的主机部分为全 1的地址是广播地址。网络上的所有主机都将接收以广播地址为目的地址的数据(参见后续关于广播的更详细的讨论)。
- 全0的地址表示本网络或本主机。换句话说，一个表示特定网络的 A类地址若主机部分为全0表示在此特定网络上的本主机。同样，网络地址为全 0(如0.0.121.1)表示在本网络上的特定主机。

这些限制减少了可用的网络和主机地址。回返地址占用了一个 A类网络地址，否则127将是最高阶的A类地址。同样，对于全0地址(0.0.0.0)的保留又减少了一个A类地址。因此，有效的A类网络局限于第一个八位组为1~126，而不是0~127，即只有126个可能的A类地址。

保留地址也影响到每个网络上的唯一主机地址的数量。网络上的最大主机数变成了 2^n-2，而不是2^n，对于A类，n=24；B类，n=16；C类，n=8。全0或全1地址分别保留下来，以用于本主机或广播地址。虽然这并没有显著的减少 A类和B类地址的数量，但却把 C类地址的数量从256减少到了254。这种地址丢失在网络划分为子网时变得更加严重 (子网将在后面讨论)。

4. 广播

定义广播是为了提供一种机制使得网络上的所有主机可以接受同一条消息。广播很有用。它允许一台主机把某种变化通知网络上的所有其他主机。例如，服务器通过发送广播来通告自己的状态变化。另外，一些主机在不知把数据向哪里传输时也可以使用广播。例如，工作站在不知道服务器的名字和地址时可以广播一个请求来寻找服务器。

虽然广播地址已经存在，但 IPv6将不实现广播地址。广播的主要问题在于对网络性能的负面影响。虽然在一个类似以太网的基带网络上广播产生的业务量不比单播多，但在其他配置中它的确导致了一些问题。扼要地说，对于诸如 ATM之类在虚电路上传输的网络，广播很麻烦；在机构的互联网中广播必须经过路由器，这也会产生问题。广播的另一问题在于虽然它通常只与一小部分主机有关，却增加了每台主机必须处理的业务量。广播在网络中的消失将在第6章中详细讨论。

5. 子网

整个IP地址空间按等级组织，外部选路基于网络地址的第一部分进行，内部选路则由网络地址的所有者负责。这种方法使得路由表更加简单、高效。但是，处理 32位地址空间和24位(A类网络)地址空间甚至16位(B类网络)地址空间的选路是有区别的，该区别在于是路由表太大以至于无法处理，还是仅仅是路由表太大以至于无法处理。由于大多数物理网络只能处

理几百台主机的连接，A类或B类地址的所有者需要设计它们的内部体系结构。

划分子网正是对该问题的解决办法。子网允许网络管理者对其地址空间分级组织。在没有划分子网的网络中，路由器严格地按照网络类型来解释网络地址。如果第一个八位组指出是一个A类地址，路由器将忽略其他三个八位组，因为它们代表的是 A类地址的主机地址。但是，当划分子网后，网络上的主机将掩盖地址的主机部分中的一部分，并将被掩盖的部分作为子网。换句话说，如果把 A类网络的第二个八位组划分为子网，路由器将把 A类网络地址和主机部分中的第一个八位组组合作为两个八位组的网络地址。

划分子网的原因有以下几个。首先，它允许系统管理员按照自己的需要组织网络地址空间。其次，在该网络之外子网是不可见的。发给 A类网络上主机的数据报总会到达进入该机构的同一个路由器，发送方无需了解 (或关心)该数据进入目的机构的网络后将发生的事情。

即使在所有主机连接在同一个 LAN的情况下仍可以划分子网，但如果网络上有不同的LAN(或网段)，子网就更加重要。一个包含多个网段的互联网如果不划分子网将很难使用，甚至在某些情况下不可用。这样中继器、网桥、网关和路由器都将无法发挥最佳性能。由于目前大部分IP网络地址是C类地址，而C类地址很难高效地划分子网，因此这将会导致一些问题。C类地址中划分子网的缺点将在第 3章中详细讨论，很简单，对于全 0地址和全1地址的保留限制了C类地址划分子网后的每个子网的主机数量。

2.3.2 IP头

IP数据报非常简单，就是在数据块(称为净荷)的前面加上一个包头。IP数据报中的数据(包括包头中的数据)以32位(4字节或4个八位组)的方式来组织。图2-3中展示了IP头字段的排列。从中可以看出，所有IP数据报头最小长度是5个字(20字节)，如果有其他选项的话，包头可能会更长。

图2-3 IPv4头包括12个不同字段

1. IP头字段

IPv4头字段包括：

- 版本：这个4位字段指明当前使用的IP版本号。这是要处理的第一个字段，因为接收方必须了解如何解释包头中的其余部分。
- 头长度：IPv4的头长度的范围从5个4字节字到15个4字节字。头长度指明头中包含的4字节字的个数。可接受的最小值是5，最大值是15(意味着包头有60字节长而选项占了其中40个字节)。
- 服务类型：这8位中只有前4位用来作为IP路由器的服务类型(TOS)请求。一个TOS位表示对如何处理数据报的优先选择：延时、吞吐量、可靠性或代价。在请求中把延时位置位意味着需要最小的延时；把吞吐量位置位意味着需要最大的吞吐量；把可靠性位置位意味着需要最高的可靠性。 TOS在IPv4中的应用并不广泛，其原因将在第 3章中讨论。由于通常对于路由没有选择余地，这些只是要考虑的建议，这些位由高层应用协议自动设置为合适的值。例如，远程网络会话要求最小延时，而文件传输要求最大吞吐量。
- 数据报长度：指的是包括包头在内的整个数据报的长度。该字段为 16位，限定了IP数据报的长度最大为65 536字节。这个字段的必要性在于IP中没有关于“数据报结束”的字符或序列。网络主机可以使用数据报长度来确定一个数据报的结束和下一个数据报的开始。
- 数据报ID：这个唯一的16位标识符由产生它的主机指定给数据报。发送主机为它送出的每个数据报产生一个单独ID，但数据报在传输的过程中可能会分段，并经过不同的网络而到达目的地。分段后的数据报都共享同一个数据报 ID，这将帮助接收主机对分段进行重装。
- 分段标志：3位分段标志位中的第一位未用，其他两位用于控制数据报的分段方式。如果“不能分段(DF)”位设为1，意味着数据报在选路到目的地的过程中不会分段传输。如果数据报不分段就无法选路，试图分段的路由器将丢掉该数据报并向源主机发送错误报文。如果“更多段(MF)”位设为1，意味着该数据报是某两个或多个分段中的一个，但不是最后一段。如果MF位设为0，意味着后面没有其他分段或者是该数据报本来就没有分段。接收主机把标志位和分段偏移一起使用，以重组被分段的数据报。
- 分段偏移值：这个字段包含 13位，它表示以8字节为单位，当前数据报相对于初始数据报的开头的位置。换句话说，数据报的第一个分段的偏移值为 0；如果第二个分段中的数据从初始数据报开头的第800字节开始，该偏移值将是100。
- 生存期：这个8位字段指明数据报在进入互联网后能够存在多长时间，它以秒为单位。生存期(TTL)用于测量数据报在穿越互联网时允许存在的秒数。其最大值是 255，当TTL到达0时，包将被网络丢弃。设定TTL的本意是让每个路由器计算出处理每个数据包所需的时间，然后从TTL中把这段时间减去。实际上，数据报穿越路由器的时间远小于 1秒，因此路由器厂商在实现中采用了一个简单的减法：即在转发数据报时把 TTL减1。在实践中，TTL代表的是数据报在被丢弃前能够穿越的最大跳数。
- 协议：指明数据报中携带的净荷类型，主要标识所使用的传输层协议：一般是 TCP连接或UDP数据报。
- 头校验和：IPv4中不提供任何可靠服务，此校验和只针对包头。计算校验和时，把包头

作为一系列16位二进制数字(校验和本身在计算时被设为0)，并把它们加在一起，然后对结果取补码。这保证了头的正确性但并没有增加任何传输可靠性或对IP的差错检查。

- 源/目的IP地址：这些是源主机和目的主机的实际的32位(4个八位组)IPv4地址。

2. IP选项

顾名思义，IP选项是可选的且不经常使用，而且它们在IPv6中的形式根本不同。在IPv4中，IP选项主要用于网络测试和调试。

可用的选项大多与选路有关。例如，有的选项允许发送方指定数据报必须经过的路由，换句话说，定义了由哪些路由器来处理该数据报。还有的选项要求中转路由器记录其IP地址为数据报打上时间戳。一些选项，尤其是指出数据报必须经过哪些IP地址的报文要求在选项后附加一些数据。

指定路由、记录路由器或增加时间戳等选项增加了IP头的长度。如果使用的话，IP选项会以没有间隔字符的方式串在一起，如果它们的结尾不在字边界，即字节数不是4字节的整数倍，还将会加上填充数据。正如上述对头长度字段的描述，选项字段可以包括不超过40字节的选项和选项数据。IPv4的选项将在第3章中详细描述。

2.3.3 数据报的转移

理解数据报的转移过程意味着要理解IP寻址方案和IP数据报头字段。发送数据报的IP主机为数据报建立的IP头中包含自己的地址作为源地址，并包含目的主机IP地址。当这个数据报沿着网络协议栈到达链路层后，链路层必须确定向“同一个本地网络”上哪一台主机发送。换句话说，即便目的地在另一个网络上，数据报也必须发送给与发送方主机在同一个网络上的主机。

发送主机将检查目的地址。如果在同一个IP网络和子网上，该主机将使用地址解析协议(ARP)向本地网络发送广播，并把IP地址映射到链路层(如以太网)地址，然后将该数据报封装到数据链路层帧中并直接发送到目的地。但是，如果目的地在不同的网络或子网上，发送者必须确定向何处发送数据，使之可以转发到正确的网络。

这就是路由器的作用。发送方主机了解本地主机，也了解路由器。一般来说，一个子网上有一个或两个路由器用来转发包。发送主机把IP数据报(由初始发出，目的地址为最终目的地)封装在链路层帧中，该帧直接发给默认路由器，由此路由器把该帧拆开并检查IP数据报头。首先，它将检查版本号，IPv4中只允许该字段为版本4。它还将继续处理头字段中的其他部分，递减生存期字段并重新计算包头校验和。

路由器还会检查目的地址以确定它是否属于路由器直接连接的任一本地网络。如果是，路由器将使用ARP确定目的地的数据链路层地址，然后把该数据报封装在数据链路层帧中发送。如果不属于该路由器直接连接的任何网络，则将数据报转发给另一个路由器。继续此过程，直到数据报到达其目的网络为止。

图2-4中展示了这个工作过程。图中包含有两个不同机构，它们均连接在Internet上，且各自有三个网络。每个网络连接到一个路由器上，每个路由器同时连接三个网络和Internet。当主机X向主机Y发送数据时，该数据将首先被发送到网络A上以到达路由器A。当路由器A收到该数据报后，此路由器将该数据报拆开，确定其目的地不在与自己连接的任何网络(A、B或C)上。然后此路由器将该数据报转发到另一个路由器上(在本例中位于Internet中某处)，该路

由器将继续通过Internet转发数据报直至到达路由器B为止。一旦路由器B收到该数据报，该路由器拆包后发现其目的地址在自己的一个本地网络上，于是这个路由器使用 ARP来查询网络以确定正确的数据链路层地址并将数据发送至该主机。

每个路由器都修改包中的生存期和头检验和。如果在发送者和接收者之间数据报必须分段，中间路由器还要修改数据报 ID和分段偏移值。在原始数据报过大而无法穿越一个中间网络的时候，这种情况就可能发生。

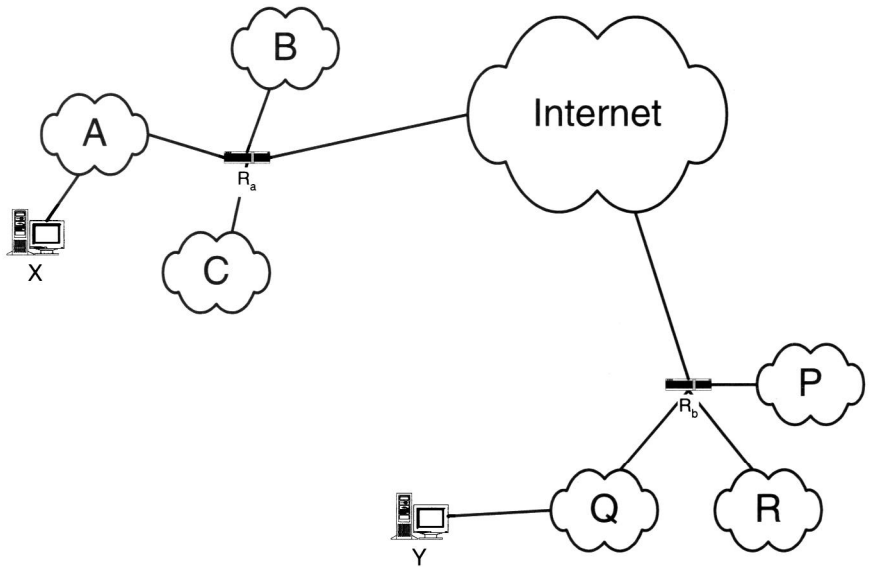

图2-4 IP路由工作过程

2.4 ICMP

IP使用Internet控制报文协议(ICMP)为路由器提供机制，以便向要求通信路径信息或路由可达性状态信息的其他路由器或主机提供这些信息。 ICMP还有其他功能，包括为其他节点通告当前时间和所用子网掩码的请求提供响应。 ICMP向其他节点提供的信息非常有用，其中包括：

- 通知节点目的地不可达。
- 发送关于特定路由或路由器的差错或状态信息。
- 对可达节点状态的请求/应答。
- 关于超时(生存期终止)数据报的通知。

ICMP是一个非常简单的协议，它使用四个字段来完成这些功能。 ICMPv6为支持IPv6的重要特性 —— 邻居发现而进行了扩展，这一点将在第 10章中讨论。

2.5 选路、传输和应用协议

关于选路、传输和应用协议的详细内容最好留给其他教材，其中包括我的《 TCP/IP Clearly Explained》(AP Professional,1997)。第8章关于选路协议的讨论将侧重于如何进行更新和修改以更加适应传输协议和应用协议。第 10章中将讨论对传输协议、应用协议及其他协议

的修改。本小节仅提供对这些主题的简单介绍。

2.5.1 选路协议

选路协议帮助定义用于确定路由器向哪里转发包和如何了解跟踪路由的规则。路由器可以使用多种不同协议与转发包的其他路由器通信，这些协议包括边界网关协议 (BGP)、选路信息协议(RIP)和开放最短路径优先(OSPF)协议等。这些协议使得路由器可以响应网络、链路和路由器状态的变化，这一点正是IP能够在大型网络上支持任意节点间连接能力的关键功能。

2.5.2 传输协议

网络层IP定义了互联网上特定节点间通信的规则，传输层协议则定义了在同一个互联网的一个或几个主机的特定进程间通信的规则。通常和 IP一起使用的传输层协议主要是传输控制协议(TCP)和用户数据报协议(UDP)。这两个协议对于IP连网很重要，但在与IPv6一起使用时它们不必有大的改动。这两个协议中与 IPv6相关的部分将在第10章中讨论，本节只进行简单介绍。

1. 传输控制协议

TCP提供了在两个端点的进程间建立虚电路的机制，这意味着一个 TCP虚连接如同在系统间承载数据的全双工电路。由于 TCP中提供了进程间数据的可靠传输，因此被称为可靠协议，它还提供了根据当前网络状态来优化传输性能的机制。这意味着，在所有数据均可收到和确认的情况下，传输速率可以逐渐增加。延时将导致发送主机在收到进一步的确认前降低发送速率。

TCP通常用于交互式应用，尤其是用于诸如web之类的某些数据接收差错将影响正常工作能力的应用中。 TCP使用了“ 三次握手 ”机制来建立电路，所有的电路都使用正式中止。除多种校验和及其他可靠性功能外，这种连接方式增加了使用 TCP的开销并导致其效率低于UDP。

2. 用户数据报协议

UDP是一个相当简单的协议。它几乎只使用源和目的信息，主要用于简单的请求 /响应式结构的应用中。它不可靠，即没有任何控制能确定 UDP数据报是否已被接收。它是无连接的，即在主机间传输数据时，不需要任何类型的电路。 UDP的无连接特性使得UDP可以向广播地址发送数据；而TCP则不同，它要求特定的源地址和目的地址。

2.5.3 应用协议

实质上所有寻址问题均已在传输层 (指定给节点上运行的特定进程的地址或端口号)和网络层(标识特定网络上的节点的 IP地址)处理。应用协议，例如超级文本传输协议 (HTTP)，就无须考虑寻址问题并因此无须为使用 IPv6而进行大的改变。 IP应用如何与IPv6网络栈一起工作将在第10章中讨论。