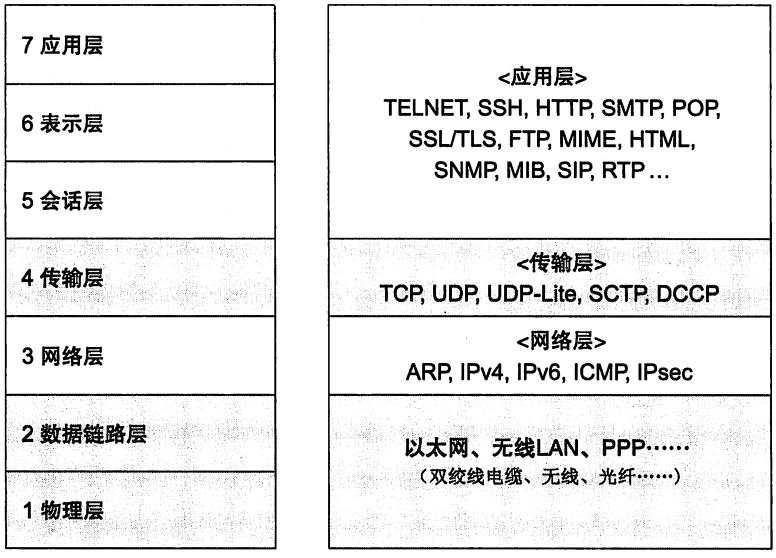
第5章

IP协议相关技术

IP（Internet Protocol）旨在让最终目标主机收到数据包，但是在这一过程中仅仅有IP是无法实现通信的。必须还有能够解析主机名称和MAC地址的功能，以及数据包在发送过程中异常情况处理的功能。此外，还会涉及IP必不可少的其他功能。

本章主要介绍作为IP的辅助和扩展规范的DNS、ARP、ICMP以及DHCP等协议。



5.1 仅凭IP无法完成通信

到第4章为止，主要介绍了网络通信中利用IP如何实现让数据包到达最终目标主机的功能，想必读者已经对此有所了解。

然而不知道大家有没有注意到，人们在上网的时候其实很少直接输入某个具体的IP地址。

在访问Web站点和发送、接收电子邮件时，我们通常会直接输入Web网站的地址或电子邮件地址等那些由应用层提供的地址，而不会使用由十进制数字组成的某个IP地址。因此，为了能让主机根据实际的IP包进行通信，就有必要实现一种功能——将应用中使用的地址映射为IP地址。

此外，在数据链路层也不使用IP地址。在以太网的情况下只使用MAC地址传输数据包。而实际上将众多IP数据包在网络上进行传送的就是数据链路本身，因此，必须了解发送端MAC地址。如果不知道MAC地址，那么通信也就无从谈起。

由此可知，在实际通信中，仅凭IP远远不够，还需要众多支持IP的相关技术才能够实现最终通信。

本章旨在介绍IP的辅助技术，具体包括DNS、ARP、ICMP、ICMPv6、DHCP、NAT等。还包括如IP隧道、IP多播、IP任播、质量控制（QoS）以及网络拥塞的显式通知和Mobile IP技术。

5.2 DNS

我们平常在访问某个网站时不使用 IP 地址，而是用一串由罗马字和点号组成的字符串。而一般用户在使用 TCP/IP 进行通信时也不使用 IP 地址。能够这样做是因为有了 DNS（Domain Name System）功能的支持。DNS 可以将那串字符串自动转换为具体的 IP 地址。

这种 DNS 不仅适用于 IPv4，还适用于 IPv6。

5.2.1 IP 地址不便记忆

TCP/IP 网络中要求每一个互连的计算机都具有其唯一的 IP 地址，并基于这个 IP 地址进行通信。然而，直接使用 IP 地址有很多不便之处。例如，在进行应用操作时，用户必须指定对端的接收地址，此时如果使用 IP 地址的话应用就会有很多不便之处。因为 IP 地址是由一串数据序列组成，并不好记▼。

▼电话号码也是一种数据序列。当人们搬家后不得不换一个新的号码时往往会感觉不好记。与此相比，由英文字母序列组成的电子邮件地址反倒比较容易记忆。

为此，TCP/IP 世界中从一开始就已经有了一个叫做主机识别码的东西。这种识别方式是指为每台计算机赋以唯一的主机名，在进行网络通信时可以直接使用主机名称而无需输入一大长串的 IP 地址。并且此时，系统必须自动将主机名转换为具体的 IP 地址。为了实现这样的功能，主机往往会利用一个叫做 hosts 的数据库文件。

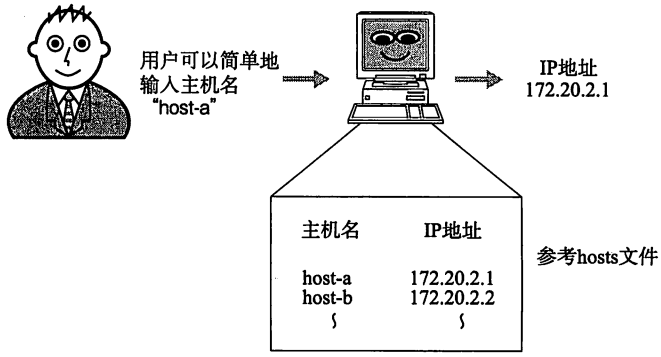


图 5.1 主机名与 IP 地址之间的转换

在互联网的起源 ARPANET 中，起初由互联网信息中心（SRI-NIC）整体管理一份 hosts 文件。如果新增一台计算机接入到 ARPANET 网或者已有的某台计算机要进行 IP 地址变更，中心的这个 hosts 文件就得更新，而其他计算机则不得不定期下载最新的 hosts 文件才能正常使用网络。

然而，随着网络规模的不断扩大、接入计算机的个数不断增加，使得这种集中管理主机名和 IP 地址的登录、变更处理的可行性逐渐降低。

5.2.2 DNS 的产生

在上述背景之下，产生了一个可以有效管理主机名和 IP 地址之间对应关系的系统，那就是 DNS 系统。在这个系统中主机的管理机构可以对数据进行变更和设定。也就是说，它可以维护一个用来表示组织内部主机名和 IP 地址之间对应关系的数据库。

在应用中，当用户输入主机名（域名）时，DNS会自动检索那个注册了主机名和IP地址的数据库，并迅速定位对应的IP地址▼。而且，如果主机名和IP地址需要进行变更时，也只需要在组织机构内部进行处理即可，而没必要再向其他机构进行申请或报告。

▼Windows和Unix中若想查找域名对应的IP地址，常用nslookup命令。输入"nslookup 主机名"时会返回对应的IP地址。

有了DNS，不论网络规模变得多么庞大，都能在一个较小的范围内通过DNS进行管理。可以说DNS充分地解决了ARPANET初期遇到的问题。就算到现在，当人们访问任何一个Web站点时，都能够直接输入主机名进行访问，这也要归功于DNS。

5.2.3　域名的构成

在理解DNS规范时，首先需要了解什么是域名。域名是指为了识别主机名称和组织机构名称的一种具有分层的名称。例如，仓敷艺术科学大学的域名如下：

```
kusa.ac.jp
```

域名由几个英文字母（或英文字符序列）用点号连接构成。在上述域名中最左边的“kusa”表示仓敷艺术科学大学（Kurashiki University of Science and the Arts）固有的域名。而“ac”表示大学（academy）或高等专科以及技术专门学校等高等教育相关机构。最后边的“jp”则代表日本（japan）。

▼持有域名的组织机构可以设置自己的子网，此时的子域名要介于主机名和域名之间。

在使用域名时，可以在每个主机名后面追加上组织机构的域名▼。例如，有pepper、piyo、kinoko等主机时，它们完整的带域名的主机名将呈如下形式：

```
pepper.kusa.ac.jp
piyo.kusa.ac.jp
kinoko.kusa.ac.jp
```

在启用域名功能之前，单凭主机名还无法完全管理IP地址，因为在不同的组织机构中不允许有同名的主机。然而，当出现了带有层次结构的域名之后，每一个组织机构就可以自由地为主机命名了。

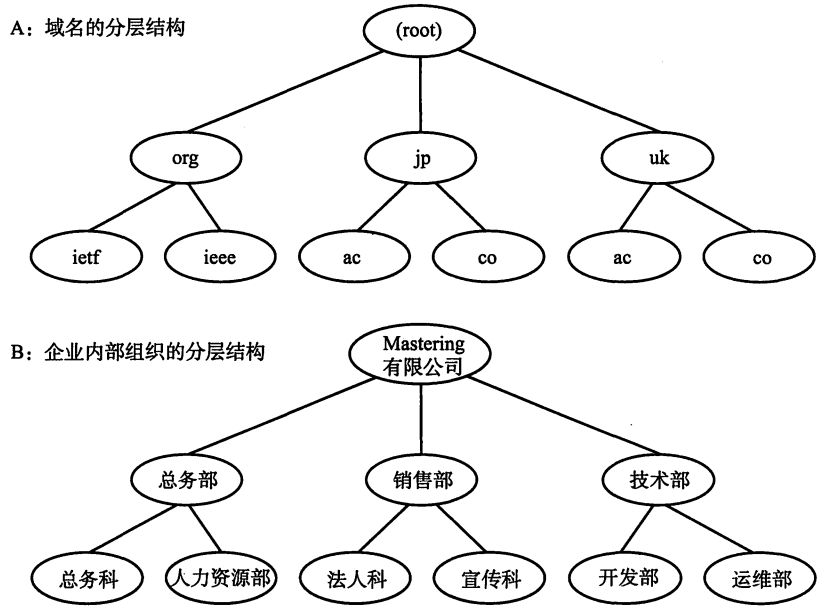
DNS的分层如图5.2所示。由于看起来像一颗倒挂的树，人们也把这种分层结构叫做树形结构。如果说顶点是树的根（Root），那么底下是这棵树的各层枝叶。顶点的下一层叫做第1层域名▼，它包括“jp（日本）”、“uk（英国）”等代表国家的域名▼，还包括代表“edu（美国教育机构）”或“com（美国企业）”等特定领域的域名▼。这种表示方法也非常类似于一个企业内部的组织结构图。

▼顶级域名（TLD：Top Level Domain）

▼国别顶级域名（ccTLD：country code TLD）

▼通用顶级域名（gTLD：generic TLD）

图 5.2
域名分层



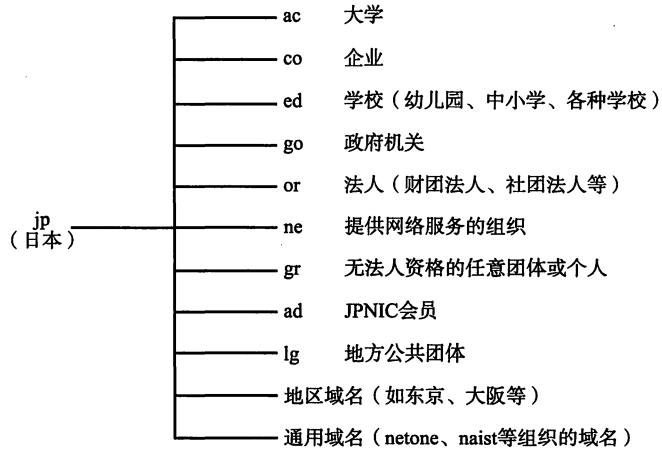
▼ jp 这个域名的登录管理和运维服务，从 2002 年 4 月 1 日起由日本的 JPRS 公司全权负责。

在 jp 的域名▼下，如图 5.3 所示，还可以有众多种类的域名。jp 往下第 2 层域名中不仅包括“ac”、“co”等表示不同组织机构的属性（组织类型）域名，还包括“tokyo”等表示地域的通用域名。甚至在使用属性（组织类型）域名或地域域名的情况下还可以有第 3 层域名。

▼ American Standard Code for Information Interchage 的缩写。是指用英文、数字以及“!”、“@”等字符表示的 7 比特编码。

很长时间以来域名都以 ASCII 字符编码▼表示，然而现在也逐渐开始使用日语等众多国家的文字表示。

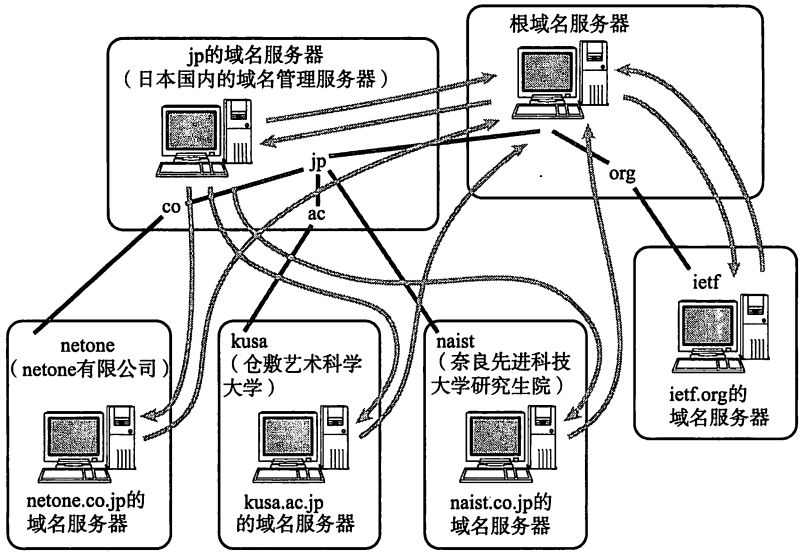
图 5.3
*.jp 域名



域名服务器

域名服务器是指管理域名的主机和相应的软件，它可以管理所在分层的域的相关信息。其所管理的分层叫做 ZONE。如图 5.4 所示，每层都设有一个域名服务器。

图5.4
域名服务器



- 各个域的分层上都有设有各自的域名服务器
- 各层域名服务器都了解该层以下分层中所有域名服务器的IP地址。因此它们从根域名服务器开始呈树状结构相互连接。
- 由于所有域名服务器都了解根域名服务器的IP地址，所以若从根开始按照顺序追踪，可以访问世界上所有域名服务器的地址。

根部所设置的DNS叫做根域名服务器。它对DNS的检索数据功能起着至关重要的作用▼。根域名服务器中注册着根以下第1层域名服务器的IP地址。以图5.4为例，根域名服务器中，注册了那些管理的域名服务器的IP地址。反之，如果想要新增一个类似jp或org的域名或修改某个已有域名，就得在根域名服务器中进行追加或变更。

▼根据DNS协议，根域名服务器可由13个IP地址表示，并且从A到M开始命名。然而，现在由于IP任播可以为多个节点设置同一个IP地址，为了提高容灾能力和负载均衡能力，根域名服务器的个数也在不断增加。关于IP任播，请参考5.8.2节。

类似地，在根域名服务器的下一层域名服务器中注册了再往下一层域名服务器的IP地址。根据每个域名服务器所管理的域名，如果下面再没有其他分层，就可以自由地指定主机名称或子网名称。不过，如果想修改该分层的域名或重新设置域名服务器的IP地址，还必须得在其上层的域名服务器中进行追加或修改。

因此，域名和域名服务器需要按照分层进行设置。如果域名服务器宕机，那么针对该域的DNS查询也就无法正常工作。因此，为了提高容灾能力，一般会设置至少两个以上的域名服务器。一旦第一个域名服务器无法提供查询时，就会自动转到第二个甚至第三个域名服务器上进行，以此可以按照顺序进行灾备处理。

所有的域名服务器都必须注册根域名服务器的IP地址。因为DNS根据IP地址进行检索时，需要从根域名服务器开始按顺序进行。关于根域名服务器IP地址相关的最新情况可以参考如下网站：

```
http://www.internic.net/zones/named.root
```

解析器（Resolver）

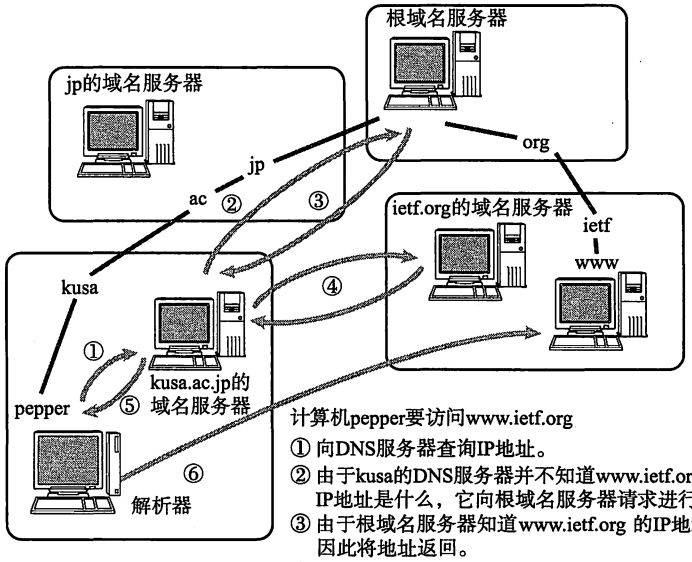
进行DNS查询的主机和软件叫做DNS解析器。用户所使用的工作站或个人电脑都属于解析器。一个解析器至少要注册一个以上域名服务器的IP地址。通常，它至少包括组织内部的域名服务器的IP地址。

5.2.4 DNS 查询

▼也叫做 query。

那么 DNS 查询▼的机制是什么呢？在此，以图 5.5 为例具体说明。图中 kusa. co. jp 域中的计算机想要访问网站 www. ietf. org，此时的 DNS 查询流程如图所示。

图 5.5
DNS 查询



▼该图中，不仅可以访问同一域中的域名服务器，还可以访问其他域的域名服务器。

解析器为了调查 IP 地址，向域名服务器▼进行查询处理。接收这个查询请求的域名服务器首先会在自己的数据库进行查找。如果有该域名所对应的 IP 地址就返回。如果没有，则域名服务器再向上一层根域名服务器进行查询处理。因此，如图所示，从根开始对这棵树按照顺序进行遍历，直到找到指定的域名服务器，并由这个域名服务器返回想要的数据。

▼缓存的时限可以在提供信息的域名服务上进行设置。

解析器和域名服务器将最新了解到的信息暂时保存在缓存里▼。这样，可以减少每次查询时的性能消耗。

5.2.5 DNS 如同互联网中的分布式数据库

前面提到 DNS 是一种通过主机名检索 IP 地址的系统。然而，它所管理的信息不仅仅是这些主机名跟 IP 地址之间的映射关系。它还要管理众多其他信息。具体可参考表 5. 1。

例如，主机名与 IP 地址的对应信息叫做 A 记录。反之，从 IP 地址检索主机名称的信息叫做 PTR。此外，上层或下层域名服务器 IP 地址的映射叫做 NS 记录。

在此特别需要指出的是 MX 记录。这类记录中注册了邮件地址与邮件接收服务器的主机名。具体可参考 8. 4 节的电子邮件说明。

表 5.1
DNS 的主要记录

类型	编号	内　容
A	1	主机名的 IP 地址（IPv4）
NS	2	域名服务器
CNAME	5	主机别名对应的规范名称
SOA	6	区域内权威记录起始标志
WKS	11	已知的服务
PTR	12	IP 地址反向解析
HINFO	13	主机相关的追加信息
MINFO	14	邮箱与邮件组信息
MX	15	邮件交换（Mail Exchange）
TXT	16	文本
SIG	24	安全证书
KEY	25	密钥
GPOS	27	地理位置
AAAA	28	主机的 IPv6 地址
NXT	30	下一代域名
SRV	33	服务器选择
*	255	所有缓存记录

5.3 ARP

只要确定了 IP 地址，就可以向这个目标地址发送 IP 数据报。然而，在底层数据链路层，进行实际通信时却有必要了解每个 IP 地址所对应的 MAC 地址。

5.3.1 ARP 概要

▼ Address Resolution Protocol

ARP▼ 是一种解决地址问题的协议。以目标 IP 地址为线索，用来定位下一个应该接收数据分包的网络设备对应的 MAC 地址。如果目标主机不在同一个链路上时，可以通过 ARP 查找下一跳路由器的 MAC 地址。不过 ARP 只适用于 IPv4，不能用于 IPv6。IPv6 中可以用 ICMPv6 替代 ARP 发送邻居探索消息▼。

▼请参考5.4.4节中的邻居探索。

5.3.2 ARP 的工作机制

那么 ARP 又是如何知道 MAC 地址的呢？简单地说，ARP 是借助 ARP 请求与 ARP 响应两种类型的包确定 MAC 地址的。

如图 5.6 所示，假定主机 A 向同一链路上的主机 B 发送 IP 包，主机 A 的 IP 地址为 172.20.1.1，主机 B 的 IP 地址为 172.20.1.2，它们互不知道对方的 MAC 地址。

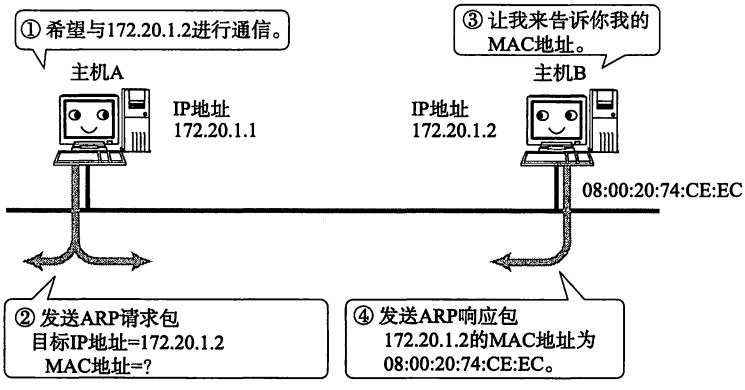


图 5.6 ARP 工作机制

主机 A 为了获得主机 B 的 MAC 地址，起初要通过广播发送一个 ARP 请求包。这个包中包含了想要了解其 MAC 地址的主机 IP 地址。也就是说，ARP 请求包中已经包含了主机 B 的 IP 地址 172.20.1.2。由于广播的包可以被同一个链路上所有的主机或路由器接收，因此 ARP 的请求包也就会被这同一个链路上所有的主机和路由器进行解析。如果 ARP 请求包中的目标 IP 地址与自己的 IP 地址一致，那么这个节点就将自己的 MAC 地址塞入 ARP 响应包返回给主机 A。

▼ ARP 请求包还有一个作用，那就是将自己的 MAC 地址告诉给对方。

总之，从一个 IP 地址发送 ARP 请求包以了解其 MAC 地址▼，目标地址将自己的 MAC 地址填入其中的 ARP 响应包返回到 IP 地址。由此，可以通过 ARP 从 IP 地址获得 MAC 地址，实现链路内的 IP 通信。

根据 ARP 可以动态地进行地址解析，因此，在 TCP/IP 的网络构造和网络通信中无需事先知道 MAC 地址究竟是什么，只要有 IP 地址即可。

如果每发送一个 IP 数据报都要进行一次 ARP 请求以此确定 MAC 地址，那将

会造成不必要的网络流量，因此，通常的做法是把获取到的MAC地址缓存▼一段时间。即把第一次通过ARP获取到的MAC地址作为IP对MAC的映射关系记忆▼到一个ARP缓存表中，下一次再向这个IP地址发送数据报时不需再重新发送ARP请求，而是直接使用这个缓存表当中的MAC地址进行数据报的发送。每执行一次ARP，其对应的缓存内容都会被清除。不过在清除之前都可以不需要执行ARP就可以获取想要的MAC地址。这样，在一定程度上也防止了ARP包在网络上被大量广播的可能性。

▼是指预见到同样的信息可能会再次使用，从而在内存中开辟一块区域记忆这些信息。

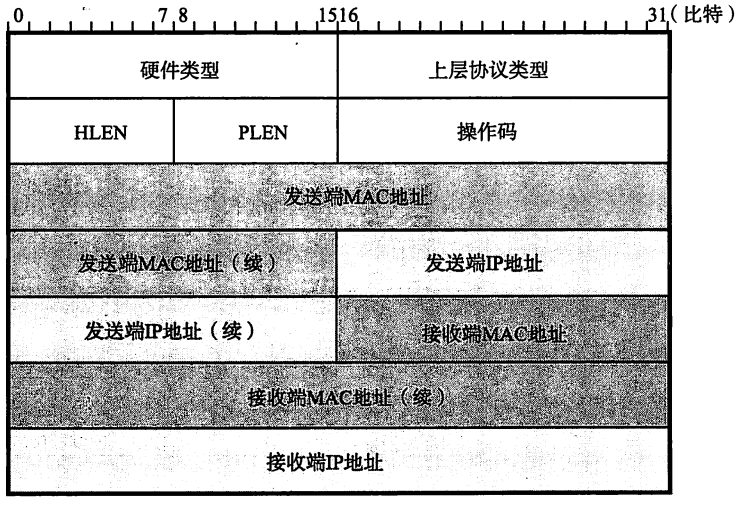
▼记录IP地址与MAC地址对应关系的数据库叫做ARP表。在UNIX或Windows中可以通过"arp-a"命令获取该表信息。

一般来说，发送过一次IP数据报的主机，继续发送多次IP数据报的可能性会比较高。因此，这种缓存能够有效地减少ARP包的发送。反之，接收ARP请求的那个主机又可以从这个ARP请求包获取发送端主机的IP地址及其MAC地址。这时它也可以将这些MAC地址的信息缓存起来，从而根据MAC地址发送ARP响应包给发送端主机。类似地，接收到IP数据报的主机又往往会继续返回IP数据报给发送端主机，以作为响应。因此，在接收主机端缓存MAC地址也是一种提高效率的方法。

不过，MAC地址的缓存是有一定期限的。超过这个期限，缓存的内容将被清除。这使得MAC地址与IP地址对应关系即使发生了变化▼，也依然能够将数据包正确地发送给目标地址。

▼尤其是在换网卡，或移动笔记本电脑、智能终端时。

图5.7
ARP包格式



HLEN：MAC地址长度=6（字节）
PLEN：IP地址长度=4（字节）

5.3.3　IP地址和MAC地址缺一不可？

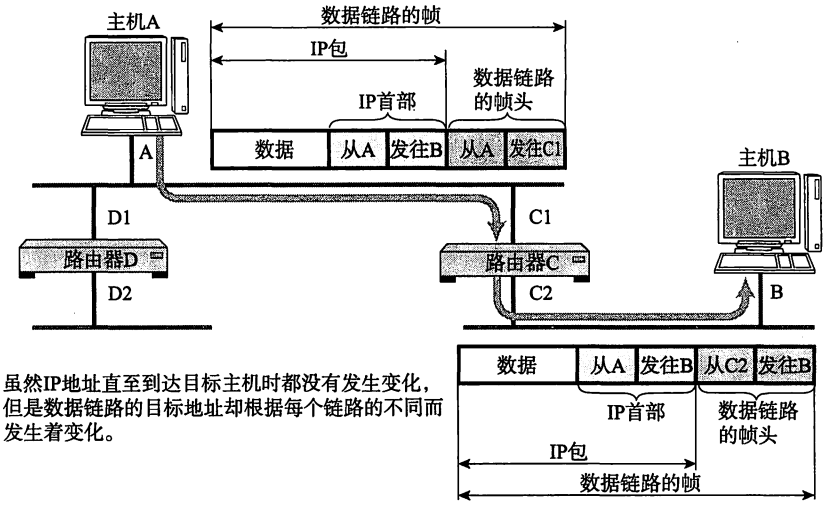
有些读者可能会提出这样的疑问："数据链路上只要知道接收端的MAC地址不就知道数据是准备发送给主机B的吗，那还需要知道它的IP地址吗？"

乍听起来确实让人觉得好像是在做多余的事。此外，还有些读者可能会质疑："只要知道了IP地址，即使不做ARP，只要在数据链路上做一个广播不就能发给主机B了吗？"那么，为什么既需要IP地址又需要MAC地址呢？

如果读者考虑一下发送给其他数据链路中某一个主机时的情况，这件事就不难理解了。如图5.8所示，主机A想要发送IP数据报给主机B时必须得经过路由

器 C。即使知道了主机 B 的 MAC 地址，由于路由器 C 会隔断两个网络，还是无法实现直接从主机 A 发送数据报给主机 B。此时，主机 A 必须得先将数据报发送给路由器 C 的 MAC 地址 C1。

图 5.8
MAC 地址与 IP 地址的作用不同



此外，假定 MAC 地址就用广播地址，那么路由器 D 也将会收到该广播消息。于是路由器 D 又将该消息转发给路由器 C，导致数据包被重复发送两次▼。

▼为了防止这种现象的出现，目前路由器可以做到将那些 MAC 地址成为了广播地址的 IP 数据报不进行转发。

在以太网上发送 IP 包时，“下次要经由哪个路由器发送数据报”这一信息非常重要。而这里的“下一个路由器”就是相应的 MAC 地址。

如此看来，IP 地址和 MAC 地址两者缺一不可。于是就有将这两个地址相关联的 ARP 协议▼。

▼为了避免这两个阶段的通信带来过多的网络流量，ARP 具有对 IP 地址和 MAC 地址的映射进行缓存的功能。有了这个缓存功能，发送 IP 包时就不必每次都发送 ARP 请求，从而防止性能下降。

最后，我们再试想一下，不使用 IP 地址，而是通过 MAC 地址连接世界上所有网络中所有的主机和节点的情况。仅仅凭一个 MAC 地址，人们是无法知道这台机器所处的位置的▼。而且如果全世界的设备都使用 MAC 地址相连，那么网桥在习得之前就得向全世界发送包。可想而知那将会造成多大的网络流量。而且由于没有任何集约机制，网桥就不得不维护一张巨大的表格来维护所学到的所有 MAC 地址。一旦这些信息超过网桥所能承受的极限，那将会导致网桥无法正常工作，也就无法实现通信了▼。

▼在使用 IP 地址的情况下，可以由网络部分充当提供位置的作用，对地址进行集约。

▼与之对应的 IP 地址路由控制表也将会变得无比庞大。

5.3.4 RARP

RARP（Reverse Address Resolution Protocol）是将 ARP 反过来，从 MAC 地址定位 IP 地址的一种协议。例如将打印机服务器等小型嵌入式设备接入到网络时就经常会用得到。

▼ Dynamic Host Configuration Protocol，具体请参考 5.5 节。DHCP 可以像 RARP 一样分配一个固定的 IP 地址。

平常我们可以通过个人电脑设置 IP 地址，也可以通过 DHCP▼ 自动分配获取 IP 地址。然而，对于使用嵌入式设备时，会遇到没有任何输入接口或无法通过 DHCP 动态获取 IP 地址的情况▼。

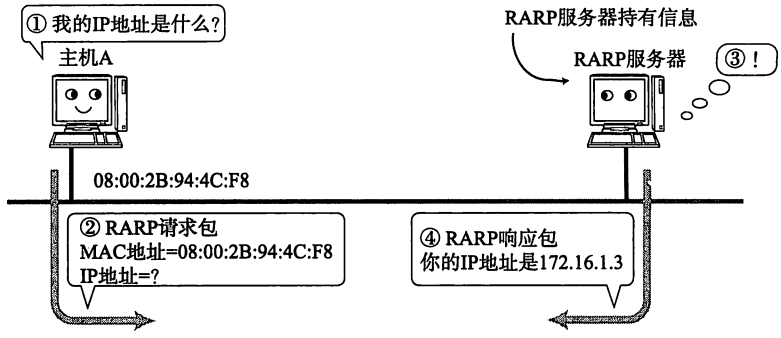
▼通过个人电脑连接这个嵌入式设备时虽然可以为其指定 IP 地址，但是用 DHCP 动态分配 IP 地址，有时会遇到无法知道所分配的 IP 是多少的情况。

在类似情况下，就可以使用 RARP。为此，需要架设一台 RARP 服务器，从而在这个服务器上注册设备的 MAC 地址及其 IP 地址▼。然后再将这个设备接入到网络，插电启动设备时，该设备会发送一条“我的 MAC 地址是 ***，请告诉我，我的 IP 地址应该是什么”的请求信息。RARP 服务器接到这个消息后返回类似于

▼使用 RARP 的前提是认为 MAC 地址就是设备固有的一个值。

“MAC地址为***的设备，IP地址为***”的信息给这个设备。而设备就根据从RARP服务器所收到的应答信息设置自己的IP地址。

图5.9
RARP



5.3.5　代理ARP

通常ARP包会被路由器隔离，但是采用代理ARP（Proxy ARP）的路由器可以将ARP请求转发给邻近的网段。由此，两个以上网段的节点之间可以像在同一个网段中一样进行通信。

在目前的TCP/IP网络当中，一般情况下用路由器连接多个网络时，会在每个网段上定义各自的子网，从而进行路由控制。然而，对于那些不支持设定子网掩码的老设备来说，不使用代理ARP，有时就无法更好地使用网络。

5.4 ICMP

5.4.1 辅助 IP 的 ICMP

架构 IP 网络时需要特别注意两点：确认网络是否正常工作，以及遇到异常时进行问题诊断。

例如，一个刚刚搭建好的网络，需要验证该网络的设置是否正确▼。此外，为了确保网络能够按照预期正常工作，一旦遇到什么问题需要立即制止问题的蔓延。为了减轻网络管理员的负担，这些都是必不可少的功能。

▼网络的设置可以包括很多内容，网线连好后涉及 IP 地址或子网掩码的设置、路由表的设置、DNS 服务器的设置、邮件服务器的设置以及代理服务器的设置等。而 ICMP 只负责其中与 IP 相关的设置。

ICMP 正是提供这类功能的一种协议。

ICMP 的主要功能包括，确认 IP 包是否成功送达目标地址，通知在发送过程当中 IP 包被废弃的具体原因，改善网络设置等。有了这些功能以后，就可以获得网络是否正常、设置是否有误以及设备有何异常等信息，从而便于进行网络上的问题诊断▼。

▼不过，ICMP 是基于尽力而为的 IP 上进行工作的，因此无法保证服务质量，而且在网络安全优先于便利性的环境里往往无法使用 ICMP，因此不宜过分依赖 ICMP。

在 IP 通信中如果某个 IP 包因为某种原因未能达到目标地址，那么这个具体的原因将由 ICMP 负责通知。如图 5.10，主机 A 向主机 B 发送了数据包，由于某种原因，途中的路由器 2 未能发现主机 B 的存在，这时，路由器 2 就会向主机 A 发送一个 ICMP 包，说明发往主机 B 的包未能成功。

ICMP 的这种通知消息会使用 IP 进行发送▼。因此，从路由器 2 返回的 ICMP 包会按照往常的路由控制先经过路由器 1 再转发给主机 A。收到该 ICMP 包的主机 A 则分解 ICMP 的首部和数据域以后得知具体发生问题的原因。

▼在 ICMP 中，包以明文的形式像 TCP/UDP 一样通过 IP 进行传输。然而，ICMP 所承担的功能并非传输层的补充，而应该把它考虑为 IP 的一部分。

ICMP 的消息大致可以分为两类：一类是通知出错原因的错误消息，另一类是用于诊断的查询消息。（如图 5.3）

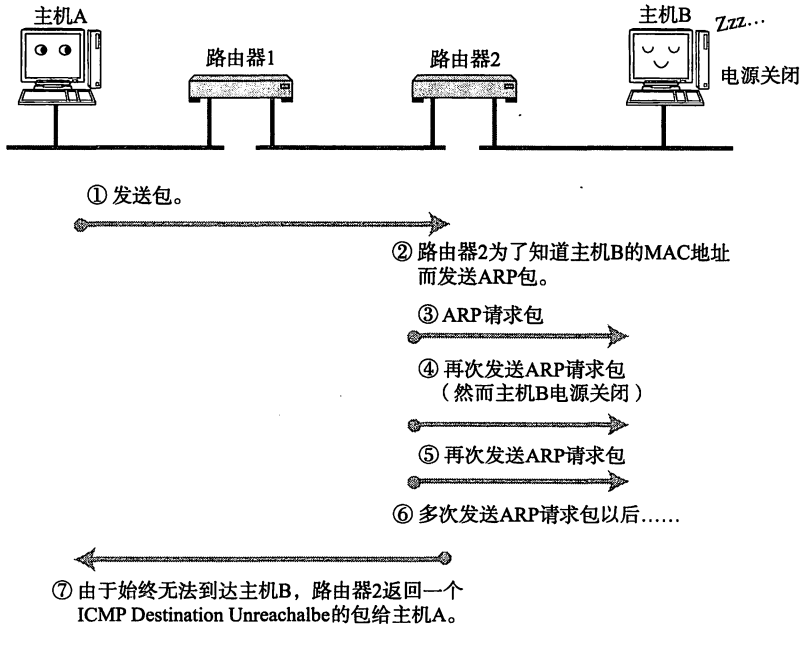


图 5.10
ICMP 无法到达的消息

表5.2
ICMP消息类型

类型（十进制数）	内　容
0	回送应答（Echo Reply）
3	目标不可达（Destination Unreachable）
4	原点抑制（Source Quench）
5	重定向或改变路由（Redirect）
8	回送请求（Echo Request）
9	路由器公告（Router Advertisement）
10	路由器请求（Router Solicitation）
11	超时（Time Exceeded）
17	地址子网请求（Address Mask Request）
18	地址子网应答（Address Mask Reply）

5.4.2　主要的ICMP消息

ICMP目标不可达消息（类型3）

IP路由器无法将IP数据包发送给目标地址时，会给发送端主机返回一个目标不可达（Destination Unreachable Message）的ICMP消息，并在这个消息中显示不可达的具体原因，如表5.3所示。

在实际通信当中经常会遇到的错误代码是1，表示主机不可达（Host Unreachable）▼，它是指路由表中没有该主机的信息，或者该主机没有连接到网络的意思。此外，错误代码4（Fragmentation Needed and Don't Fragment was Set）则用于前面4.5.3节介绍过的MTU探索。由此，根据ICMP不可达的具体消息，发送端主机也就可以了解此次发送不可达的具体原因。

▼自从不再有网络分类以后，Network Unreachable也渐渐不再使用了。

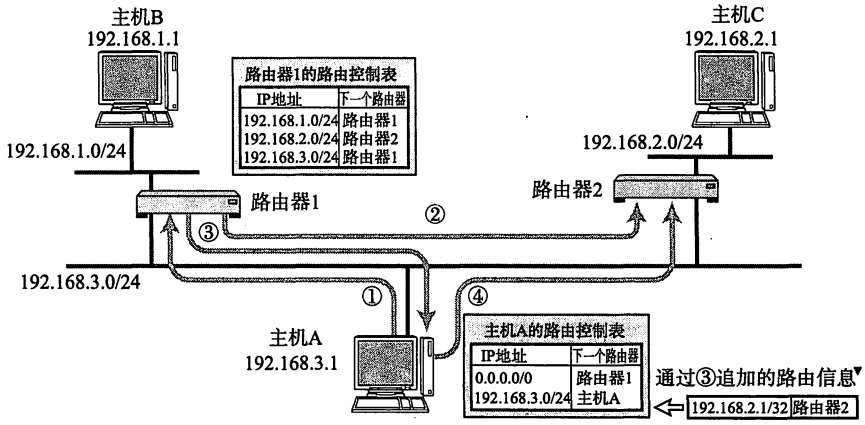
表5.3
ICMP不可达消息

错误号	ICMP不可达消息
0	Network Unreachable
1	Host Unreachable
2	Protocol Unreachable
3	Port Unreachable
4	Fragmentation Needed and Don't Fragment was Set
5	Soruce Route Failed
6	Destination Network Unknown
7	Destination Host Unknown
8	Source Host Isolated
9	Communication with Destination Network is Administratively Prohibited
10	Communication with Destination Host is Administratively Prohibited
11	Destination Network Unreachable for Type of Service
12	Destination Host Unreachable for Type of Service

ICMP 重定向消息（类型 5）

如果路由器发现发送端主机使用了次优的路径发送数据，那么它会返回一个 ICMP 重定向（ICMP Redirect Message）的消息给这个主机。在这个消息中包含了最合适的路由信息和源数据。这主要发生在路由器持有更好的路由信息的情况下。路由器会通过这样的 ICMP 消息给发送端主机一个更合适的发送路由。

图 5.11
ICMP 重定向消息



① 主机A要与主机C进行通信，此时主机A的路由控制表中没有192.168.2.0/24的记录，因此采用默认的路由发往路由器1。

② 路由器1知道192.168.2.0/24的子网在路由器2的后面，因此将包转发给路由器2。

③ 由于给192.168.2.1的包直接发送给路由器2效率会更高，因此路由器1发送一个ICMP重定向的包给主机A。

④ 主机A将这个路由信息追加到自己的路由控制表▼中，以备再次发送数据给主机C时使用路由器2而不是路由器1。

▼由于 ICMP 重定向消息中并不包含表示网络部分的子网掩码的长度，因此追加的路由信息为/32 的形式。

▼鉴于自动追加的信息要在一定期限之后删除，ICMP 的重定向消息也会在一定时间以后自动消除。

不过，多数情况下由于这种重定向消息成为引发问题的原因，所以往往不进行这种设置▼。

▼例如，不是发送端主机，而是途中某个路由器的路由控制表不正确时，ICMP 有可能无法正常工作。

ICMP 超时消息（类型 11）

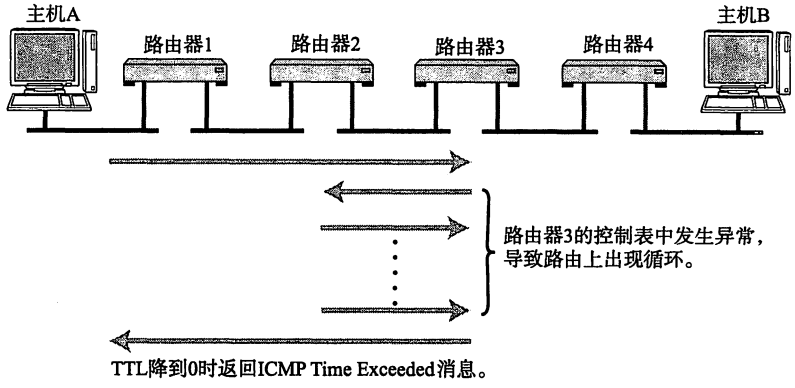
IP 包中有一个字段叫做 TTL（Time To Live，生存周期），它的值随着每经过一次路由器就会减 1▼，直到减到 0 时该 IP 包会被丢弃。此时，IP 路由器将会发送一个 ICMP 超时的消息（ICMP Time Exceeded Message，错误号 0▼）给发送端主机，并通知该包已被丢弃。

▼当 IP 包在路由器上停留 1 秒以上时减去所停留的秒数，但是现在绝大多数设备并不做这样的处理。

▼错误号 1 表示将被拆分包做重构处理时超时。

设置 IP 包生存周期的主要目的，是为了在路由控制遇到问题发生循环状况时，避免 IP 包无休止地在网络上被转发。此外，有时可以用 TTL 控制包的到达范围，例如设置一个较小的 TTL 值。

图5.12
ICMP 时间超过消息



方便易用的 traceroute

有一款充分利用 ICMP 超时消息的应用叫做 traceroute▼。它可以显示出由执行程序的主机到达特定主机之前历经多少路由器。它的原理就是利用 IP 包的生存期限从 1 开始按照顺序递增的同时发送 UDP 包，强制接收 ICMP 超时消息的一种方法。这样可以将所有路由器的 IP 地址逐一呈现。这个程序在网络上发生问题时，是问题诊断常用的一个强大工具。具体用法是在 UNIX 命令行里输入“traceroute 目标主机地址”即可。

关于 traceroute 的源代码可以参考以下网址：

```
http://ee.lbl.gov/
```

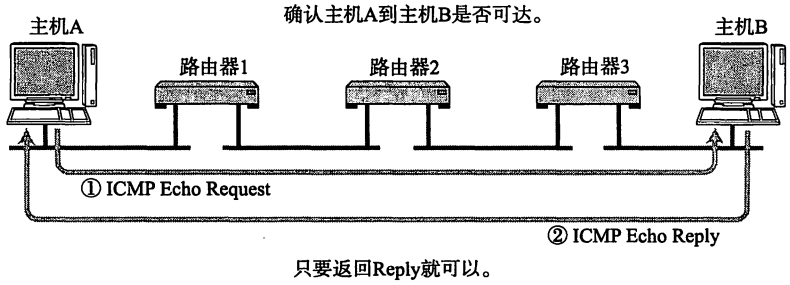
▼在 UNIX、MacOS 中是这个命令，而在 Windows 中对等的命令叫做 tracert。

ICMP 回送消息（类型 0、8）

用于进行通信的主机或路由器之间，判断所发送的数据包是否已经成功到达对端的一种消息。可以向对端主机发送回送请求的消息（ICMP Echo Request Message，类型 8），也可以接收对端主机发回来的回送应答消息（ICMP Echo Reply Message，类型 0）。网络上最常用的 ping 命令▼就是利用这个消息实现的。

▼ Packet InterNetwork Groper，判断对端主机是否可达的一种命令。

图5.13
ICMP 回送消息



5.4.3 其他 ICMP 消息

ICMP 原点抑制消息（类型 4）

在使用低速广域线路的情况下，连接 WAN 的路由器可能会遇到网络拥堵的问题。ICMP 原点抑制消息的目的就是为了缓和这种拥堵情况。当路由器向低速线路发送数据时，其发送队列的残存变为零而无法发送出去时，可以向 IP 包的源地址发送一个 ICMP 原点抑制（ICMP Source Quench Message）消息。收到这个消息的主机借此了解在整个线路的某一处发生了拥堵的情况，从而打开 IP 包的传输间隔。然而，由于这种 ICMP 可能会引起不公平的网络通信，一般不被使用。

ICMP 路由器探索消息（类型 9、10）

主要用于发现与自己相连网络中的路由器。当一台主机发出 ICMP 路由器请求（Router Solicitaion，类型 10）时，路由器则返回 ICMP 路由器公告消息（Router Advertisement，类型 9）给主机。

ICMP 地址掩码消息（类型 17、18）

主要用于主机或路由器想要了解子网掩码的情况。可以向那些目标主机或路由器发送 ICMP 地址掩码请求消息（ICMP Address Mask Request，类型 17），然后通过接收 ICMP 地址掩码应答消息（ICMP Address Mask Reply，类型 18）获取子网掩码的信息。

5.4.4 ICMPv6

ICMPv6 的作用

IPv4 中 ICMP 仅作为一个辅助作用支持 IPv4。也就是说，在 IPv4 时期，即使没有 ICMP，仍然可以实现 IP 通信。然而，在 IPv6 中，ICMP 的作用被扩大，如果没有 ICMPv6，IPv6 就无法进行正常通信。

尤其在 IPv6 中，从 IP 地址定位 MAC 地址的协议从 ARP 转为 ICMP 的邻居探索消息（Neighbor Discovery）。这种邻居探索消息融合了 IPv4 的 ARP、ICMP 重定向以及 ICMP 路由器选择消息等功能于一体，甚至还提供自动设置 IP 地址的功能▼。

▼ ICMPv6 中没有 DNS 服务器的通知功能，因此实际上需要与 DHCPv6 组合起来才能实现自动设置 IP 地址。

ICMPv6 中将 ICMP 大致分为两类：一类是错误消息，另一类是信息消息。类型 0～127 属于错误消息，128～255 属于信息消息。

表 5.4 ICMPv6 错误消息

类型（十进制数）	内　容
1	目标不可达（Destination Unreachable）
2	包过大（Packet Too Big）
3	超时（Time Exceeded）
4	参数问题（Parameter Problem）

表5.6
ICMPv6 信息消息

类型（十进制数）	内　容
128	回送请求消息（Echo Request）
129	回送应答消息（Echo Reply）
130	多播监听查询（Multicast Listener Query）
131	多播监听报告（Multicast Listener Report）
132	多播监听结束（Multicast Listener Done）
133	路由器请求消息（Router Solicitation）
134	路由器公告消息（Router Advertisement）
135	邻居请求消息（Neighbor Solicitation）
136	邻居宣告消息（Neighbor Advertisement）
137	重定向消息（Redirect Message）
138	路由器重编号（Router Renumbering）
139	信息查询（ICMP Node Information Query）
140	信息应答（ICMP Node Information Response）
141	反邻居探索请求消息（Inverse Neighbor Discovery Solicitation）
142	反邻居探索宣告消息（Inverse Neighbor Discovery Advertisement）

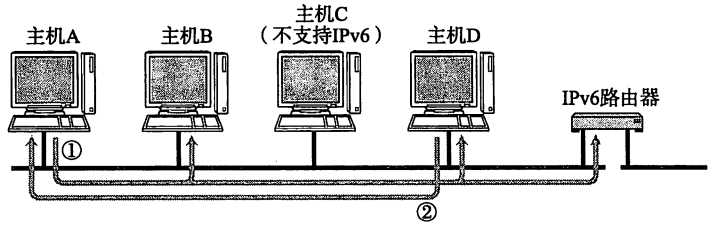
邻居探索

▼ IPv4 中查询 IP 地址与 MAC 地址对应关系用到的是 ARP。

▼ IPv4 中所使用的 ARP 采用广播，使得不支持 ARP 的节点也会收到包，造成一定的浪费。

ICMPv6 中从类型 133 至类型 137 的消息叫做邻居探索消息。这种邻居探索消息对于 IPv6 通信起着举足轻重的作用。邻居请求消息用于查询 IPv6 的地址与 MAC 地址的对应关系，并由邻居宣告消息得知 MAC 地址▼。邻居请求消息利用 IPv6 的多播地址▼实现传输。

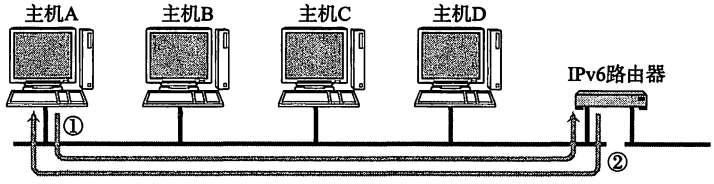
图5.14
IPv6 中查询 MAC 地址



① 以主机D为目标用多播发送邻居探索请求消息，查询主机D的MAC地址。
② 主机D通过邻居探索宣告消息将自己的MAC地址通知给主机A。

此外，由于 IPv6 中实现了即插即用的功能，所以在没有 DHCP 服务器的环境下也能实现 IP 地址的自动获取。如果是一个没有路由器的网络，就使用 MAC 地址作为链路本地单播地址（4.6.6 节）。而在一个有路由器的网络环境中，可以从路由器获得 IPv6 地址的前面部分，后面部分则由 MAC 地址进行设置。此时可以利用路由器请求消息和路由器宣告消息进行设置。

图 5.15
IP 地址的自动设置



① 通过路由器请求消息查询IP地址前面部分的内容。
② 通过路由器宣告消息通知IP地址后面部分的内容。

5.5 DHCP

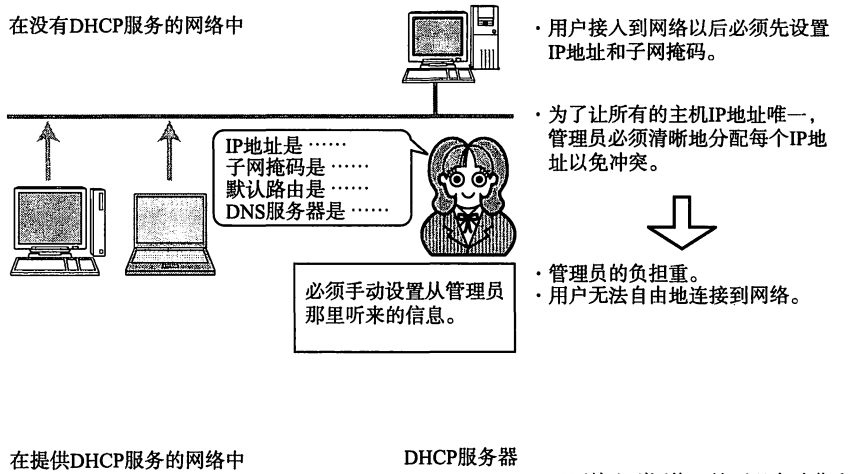
5.5.1 DHCP实现即插即用

如果逐一为每一台主机设置IP地址会非常繁琐的事情。特别是在移动使用笔记本电脑、智能终端以及平板电脑等设备时，每移动到一个新的地方，都要重新设置IP地址。

于是，为了实现自动设置IP地址、统一管理IP地址分配，就产生了DHCP（Dynamic Host Configuration Protocol）协议。有了DHCP，计算机只要连接到网络，就可以进行TCP/IP通信。也就是说，DHCP让即插即用▼变得可能。而DHCP不仅在IPv4中，在IPv6中也可以使用。

▼指只要物理上一连通，无需专门设置就可以直接使用这个物理设备。

图5.16
DHCP



5.5.2 DHCP的工作机制

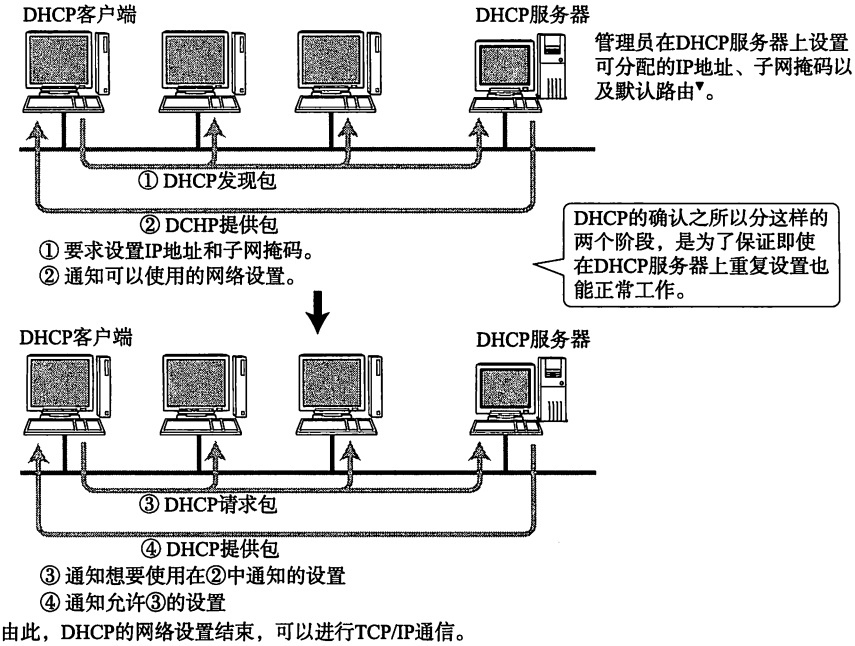
使用DHCP之前，首先要架设一台DHCP服务器▼。然后将DHCP所要分配的IP地址设置到服务器上。此外，还需要将相应的子网掩码、路由控制信息以及DNS服务器的地址等设置到服务器上。

▼很多时候用该网段的路由器充当DHCP服务器。

关于从DHCP中获取IP地址的流程，以图5.17为例简单说明的话，主要分为两个阶段▼。

▼在发送DHCP发现包与DHCP请求包时，DHCP即插即用的IP地址尚未确定。因此，DHCP发现包的目标地址为广播地址255.255.255.255，而源地址则为0.0.0.0，表示未知。

图 5.17
DHCP 的工作原理



▼ DHCP 在分配 IP 地址有两种方法。一种是由 DHCP 服务器在特定的 IP 地址中自动选出一个进行分配。另一种方法是针对 MAC 地址分配一个固定的 IP 地址。而且这两种方法可以并用。

使用 DHCP 时，如果 DHCP 服务器遇到故障，将导致无法自动分配 IP 地址，从而也导致网段内所有主机之间无法进行 TCP/IP 通信。为了避免此类问题的发生，通常人们会架设两台或两台以上的 DHCP 服务器。不过启动多个 DHCP 服务器时，由于每个服务器内部都记录着 IP 地址分配情况的信息，因此可能会导致几处分配的 IP 地址相互冲突▼。

▼为了避免这种地址重复的危险，可以在 DHCP 服务器上区分所要分配的地址。

为了检查所要分配的 IP 地址以及已经分配了的 IP 地址是否可用，DHCP 服务器或 DHCP 客户端必须具备以下功能：

- DHCP 服务器

  在分配 IP 地址前发送 ICMP 回送请求包，确认没有返回应答。

- DHCP 客户端

  针对从 DHCP 那里获得的 IP 地址发送 ARP 请求包，确认没有返回应答。

在获得 IP 地址之前做这种事先处理可能会耗一点时间，但是可以安全地进行 IP 地址分配。

5.5.3 DHCP 中继代理

家庭网络大多都只有一个以太网（无线 LAN）的网段，与其连接的主机台数也不会太多。因此，只要有一台 DHCP 服务器就足以应对 IP 地址分配的需求，而大多数情况下都由宽带路由器充当这个 DHCP 的角色。

相比之下，一个企业或学校等较大规模组织机构的网络环境当中，一般会有多个以太网（无线 LAN）网段。在这种情况下，若要针对每个网段都设置 DHCP

服务器将会是个庞大的工程。即使路由器可以分担 DHCP 的功能，如果网络中有不下 100 个路由器，就要为 100 个路由器设置它们各自可分配 IP 地址的范围，并对这些范围进行后续的变更维护，这将是一个极其耗时和难于管理的工作▼。也就是说将 DHCP 服务器分设到各个路由器上，于管理和运维都不是件有益的事。

▼ DHCP 服务器分配的 IP 地址范围，有时会随着服务器或打印机等固定 IP 设备的增减而不得不发生变化。

因此，在这类网络环境中，往往需要将 DHCP 统一管理。具体方法可以使用 DHCP 中继代理来实现。有了 DHCP 中继代理以后，对不同网段的 IP 地址分配也可以由一个 DHCP 服务器统一进行管理和运维。

这种方法使得在每个网段架设一个 DHCP 服务器被取代，只需在每个网段设置一个 DHCP 中继代理即可▼。它可以设置 DHCP 服务器的 IP 地址，从而可以在 DHCP 服务器上为每个网段注册 IP 地址的分配范围。

▼ DHCP 中继代理多数为路由器，不过也有在主机中安装某些软件得以实现的情况。

DHCP 客户端会向 DHCP 中继代理发送 DHCP 请求包，而 DHCP 中继代理在收到这个广播包以后再以单播的形式发给 DHCP 服务器。服务器端收到该包以后再向 DHCP 中继代理返回应答，并由 DHCP 中继代理将此包转发给 DHCP 客户端▼。由此，DHCP 服务器即使不在同一个链路上也可以实现统一分配和管理 IP 地址。

▼ DHCP 包中包含发出请求的主机的 MAC 地址。DHCP 中继代理正是利用这个 MAC 地址将包返回给了 DHCP 客户端。

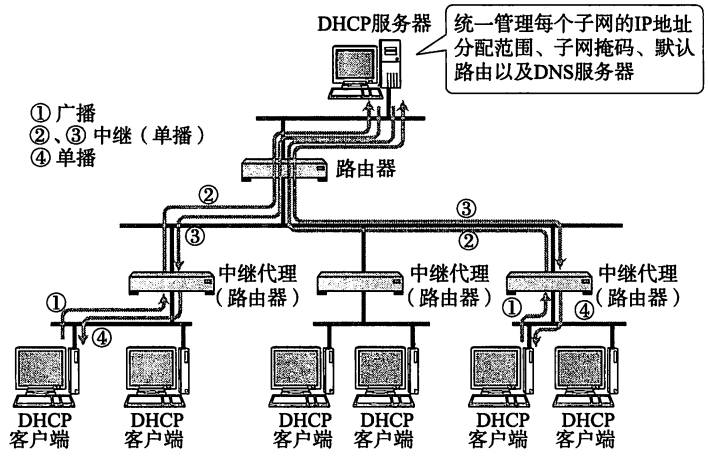


图 5-18
DHCP 中继代理

5.6 NAT

5.6.1 NAT 定义

NAT（Network Address Translator）是用于在本地网络中使用私有地址，在连接互联网时转而使用全局 IP 地址的技术。除转换 IP 地址外，还出现了可以转换 TCP、UDP 端口号的 NAPT（Network Address Ports Translator）技术，由此可以实现用一个全局 IP 地址与多个主机的通信▼。具体可参考图 5.19 和图 5.20 的构造。

▼通常人们提到的 NAT，多半是指 NAPT。NAPT 也叫做 IP 伪装或 Multi NAT。

NAT（NAPT）实际上是为正在面临地址枯竭的 IPv4 而开发的技术。不过，在 IPv6 中为了提高网络安全也在使用 NAT，在 IPv4 和 IPv6 之间的相互通信当中常常使用 NAT-PT▼。

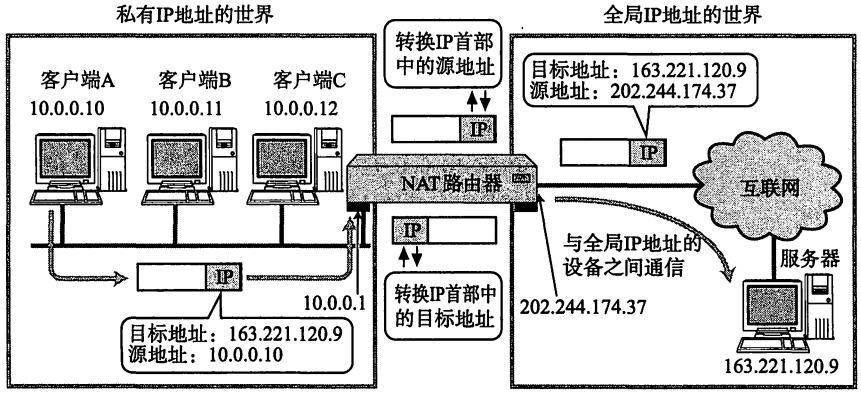
▼可参考 5.6.3 节。

5.6.2 NAT 的工作机制

如图 5.19 所示，以 10.0.0.10 的主机与 163.221.120.9 的主机进行通信为例。利用 NAT，途中的 NAT 路由器将发送源地址从 10.0.0.10 转换为全局的 IP 地址（202.244.174.37）再发送数据。反之，当包从地址 163.221.120.9 发过来时，目标地址（202.244.174.37）先被转换成私有 IP 地址 10.0.0.10 以后再被转发▼。

▼在 TCP 或 UDP 中，由于 IP 首部中的 IP 地址还要用于校验和的计算，因此当 IP 地址发生变化时，也需要相应地将 TCP、UDP 的首部进行转换。

图 5.19 NAT

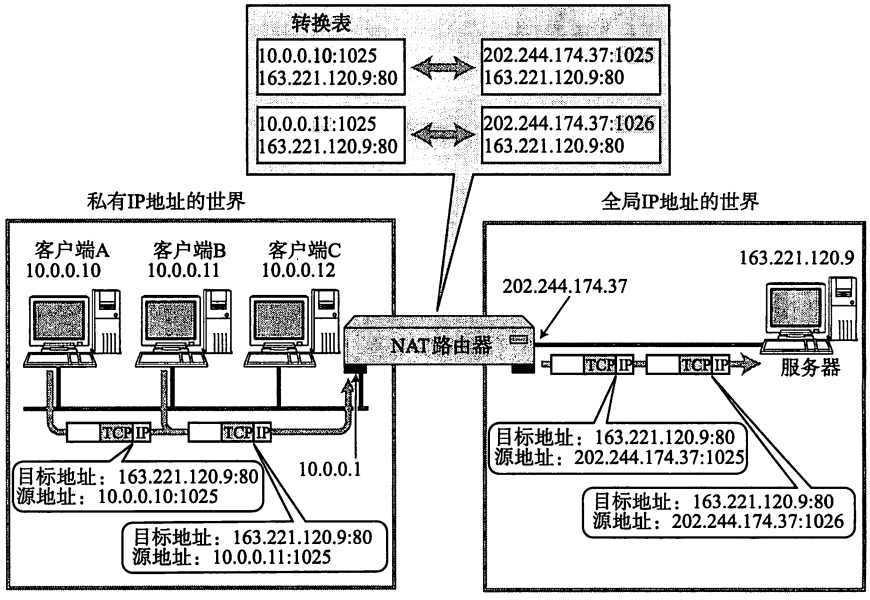


局域网内设置为私有IP地址，在与外部通信时被替换成全局IP地址。

在 NAT（NAPT）路由器的内部，有一张自动生成的用来转换地址的表。当 10.0.0.10 向 163.221.120.9 发送第一个包时生成这张表，并按照表中的映射关系进行处理。

当私有网络内的多台机器同时都要与外部进行通信时，仅仅转换 IP 地址，人们不免担心全局 IP 地址是否不够用。这时采用如图 5.20 所示的包含端口号一起转换的方式（NAPT）可以解决这个问题。

图 5.20
NAPT



*图中用"IP地址：端口号"标记。

关于这一点，第六章有更详细的说明。不过在此需要注明的一点是，在使用TCP或UDP的通信当中，只有目标地址、源地址、目标端口、源端口以及协议类型（TCP还是UDP）五项内容都一致时才被认为是同一个通信连接。此时所使用的正是NAPT。

图5.20中，主机163.221.120.9的端口号是80，LAN中有两个客户端10.0.0.10和10.0.0.11同时进行通信，并且这两个客户端的本地端口都是1025。此时，仅仅转换IP地址为某个全局地址202.244.174.37，会令转换后的所有数字完全一致。为此，只要将10.0.0.11的端口号转换为1026就可以解决问题。如图5.20所示，生成一个NAPT路由器的转换表，就可以正确地转换地址跟端口的组合，令客户端A、B能同时与服务器之间进行通信。

这种转换表在NAT路由器上自动生成。例如，在TCP的情况下，建立TCP连接首次握手时的SYN包一经发出，就会生成这个表。而后又随着收到关闭连接时发出FIN包的确认应答从表中被删除▼。

▼UDP中两端应用进行通信时起止时间不一定保持一致，因此在这种情况下生成转换表相对较难。

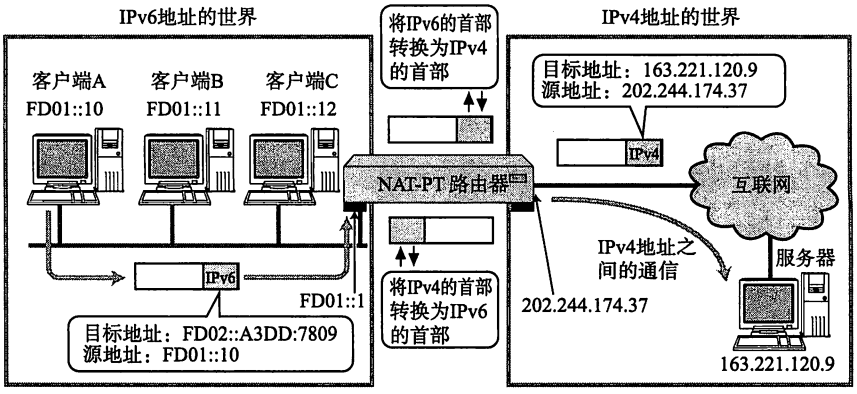
5.6.3　NAT-PT（NAPT-PT）

现在很多互联网服务都基于IPv4。如果这些服务不能做到在IPv6中也能正常使用的话，搭建IPv6网络环境的优势也就无从谈起了。

为了解决这个问题，就产生了NAT-PT（NAPT-PT）▼规范。NAT-PT是将IPv6的首部转换为IPv4的首部的一种技术。有了这种技术，那些只有IPv6地址的主机也就能够与IPv4地址的其他主机进行通信了。

▼PT是Protocol Translatio的缩写。严格来讲NAT-PT用来翻译IP地址，而NATP-PT则是用来翻译IP首部与端口号的。

图 5.21
NAT-PT



在局域网内设定成IPv6地址，与外部通信时改为IPv4地址。

NAT-PT 有很多形式，其中最让人们期待的当属结合 DNS 和 IP 首部替换的 DNS-ALG▼。不过，不论采用哪种形式，它们都避免不了下一小节所涉及的问题。

▼ ALG 是 Application Level Gateway 的缩写。

5.6.4 NAT 的潜在问题

由于 NAT（NAPT）都依赖于自己的转换表，因此会有如下几点限制：

- 无法从 NAT 的外部向内部服务器建立连接▼。
- 转换表的生成与转换操作都会产生一定的开销。
- 通信过程中一旦 NAT 遇到异常需重新启动时，所有的 TCP 连接都将被重置。
- 即使备置两台 NAT 做容灾备份，TCP 连接还是会被断开。

▼虽然可以指定端口号允许向内部访问，但是数量要受限于全局 IP 地址的个数。

5.6.5 解决 NAT 的潜在问题与 NAT 穿越

解决 NAT 上述潜在的问题有两种方法：

第一种方法就是改用 IPv6。在 IPv6 环境下可用的 IP 地址范围有了极大的扩展，以至于公司或家庭当中所有设备都可以配置一个全局 IP 地址▼。因为如果地址枯竭的问题得到解决，那么也就没必要再使用 NAT 了。然而，IPv6 的普及到现在为止都远不及人们的预期，前景不容乐观。

▼然而，如果不是所有设备都有 IPv6 的地址，其意义也就不大了。

另一种方法是，即使是在一个没有 NAT 的环境里，根据所制作的应用，用户可以完全忽略 NAT 的存在而进行通信。在 NAT 内侧（私有 IP 地址的一边）主机上运行的应用为了生成 NAT 转换表，需要先发送一个虚拟的网络包给 NAT 的外侧。而 NAT 并不知道这个虚拟的包究竟是什么，还是会照样读取包首部中的内容并自动生成一个转换表。这时，如果转换表构造合理，那么还能实现 NAT 外侧的主机与内侧的主机建立连接进行通信。有了这个方法，就可以让那些处在不同 NAT 内侧的主机之间也能够进行相互通信。此外，应用还可以与 NAT 路由器进行通信生成 NAT 表，并通过一定的方法将 NAT 路由器上附属的全局 IP 地址传给应用▼。

▼可以使用微软提供的 UPnP（Universal Plug and Play）规范。

如此一来，NAT 外侧与内侧可以进行通信，这种现象叫做“NAT 穿越”。于是 NAT 那个“无法从 NAT 的外部向内部服务器建立连接”的问题也就迎刃而解

了。而且这种方法与已有的IPv4环境的兼容性非常好，即使不迁移到IPv6也能通信自如。出于这些优势，市面上已经出现了大量与NAT紧密集合的应用▼。

然而，NAT友好的应用程序也有它的问题。例如，NAT的规范越来越复杂，应用的实现变得更耗时。而且应用一旦运行在一个开发者未预想到的特殊网络环境中时，会出现无法正常工作、遇到状况时难于诊断等问题▼。

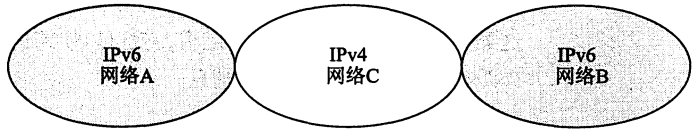
▼由此，IPv4的寿命又被延长，向IPv6的迁移也就放慢脚步了。

▼迁移到IPv6以后，系统会变得更为简单，因此它有着相当大的优势。如果同时使用IPv4和IPv6，会导致系统变得更为复杂。这对于系统开发、设计、运用等人员来说，是一件非常麻烦的事。

5.7 IP 隧道

在一个如图 5.22 所示的网络环境里，网络 A、B 使用 IPv6，如果处于中间位置的网络 C 支持使用 IPv4 的话，网络 A 与网络 B 之间将无法直接进行通信。为了让它们之间正常通信，这时必须得采用 IP 隧道的功能。

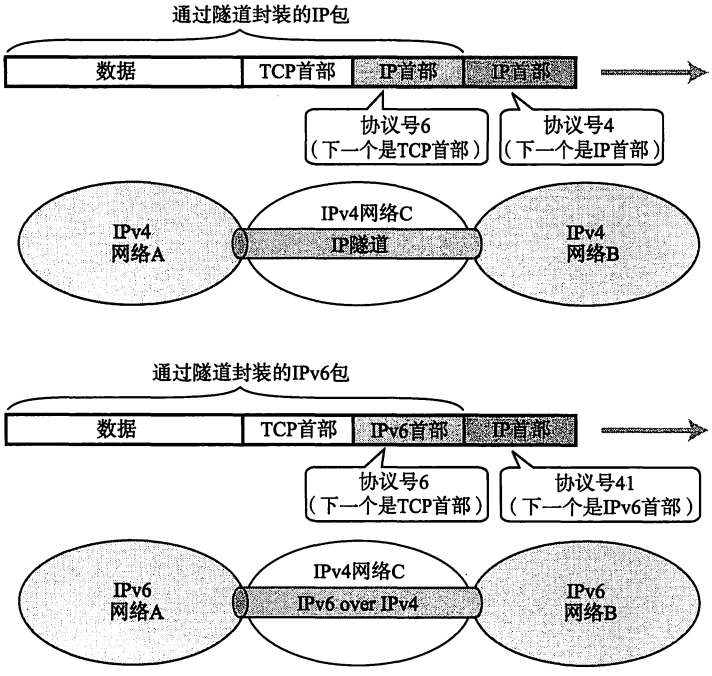
图 5.22
夹着 IPv4 网络的两个 IPv6 网络



IP 隧道中可以将那些从网络 A 发过来的 IPv6 的包统和为一个数据，再为之追加一个 IPv4 的首部以后转发给网络 C。

一般情况下，紧接着 IP 首部的是 TCP 或 UDP 的首部。然而，现在的应用当中“IP 首部的后面还是 IP 首部”或者“IP 首部的后面是 IPv6 的首部”等情况与日俱增。这种在网络层的首部后面继续追加网络层首部的通信方法就叫做“IP 隧道”。

图 5.23
IP 隧道



构造一个既支持 IPv4 又支持 IPv6 的网络是一项极其庞大的工程。在这种网络环境中，由于其路由表的量有可能会涨到平常的两倍，所以会给网络管理员增加不小的负担，而在路由器进行两种协议都要支持的设置也是相当费劲的事情。骨干网上通常使用 IPv6 或 IPv4 进行传输。因此，那些不支持的路由器就可以采用 IP 隧道的技术转发数据包，而对应的 IP 地址也可以在一旁进行统一管理。这就在一定程度上减轻了管理员的部分工作▼。此外，由于骨干网的设备上仅在一旁应对 IP 隧道即可，这也可以大量地减少投资成本。

▼隧道一旦设置有误，会导致数据包在网络上无限循环等严重问题。因此此处的设置需要极其谨慎。

- Mobile IP
- 多播包的转播
- IPv4网络中传送IPv6的包（6to4▼）
- IPv6网络中传送IPv4的包
- 数据链路帧通过IP包发送（L2TP▼）

▼指用IPv4包封装IPv6包的方式。IPv6的地址中包含全局6to4路由器（在IPv4网络入口）的IPv4地址。

▼将数据链路的PPP包用IP包转发的一种技术。

图5.24展示了一个利用IP隧道转发多播消息的例子。由于现在很多路由器上没有多播包的路由控制信息，多播消息也就无法穿越路由器发送信息。那么在这类环境当中，如果使用IP隧道，就可以使路由器用单播的形式发包，也就能够向距离较远的链路转发多播消息。

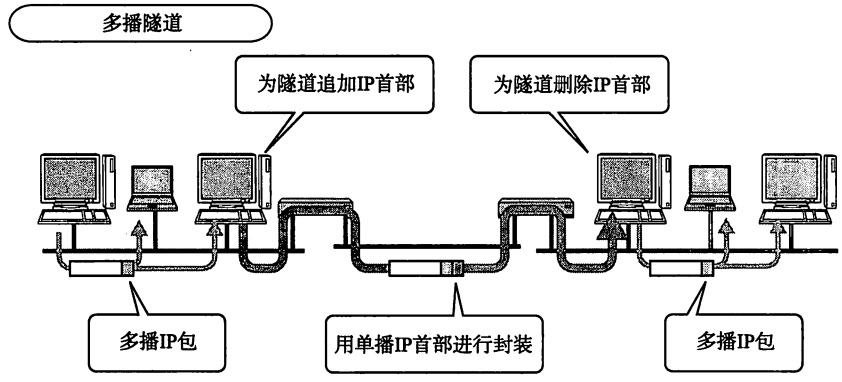


图5.24 多播隧道

5.8 其他 IP 相关技术

5.8.1 IP 多播相关技术

在多播通信中，确认接收端是否存在非常重要。如果没有接收端，发送多播消息将会造成网络流量的浪费。

而确认是否有接收端，要通过 MLD▼ 实现。它是 IPv4 中 IGMP▼ 和 IPv6 中 ICMPv6▼ 的重要功能之一。

IGMP（MLD）主要有两大作用：

1. 向路由器表明想要接收多播消息（并通知想接收多播的地址）。
2. 向交换集线器通知想要接收多播的地址。

首先，路由器会根据第 1 个作用，了解到想要接收多播的主机，并将这个信息告知给其他的路由器，准备接收多播消息。而多播消息的发送路径则由 PIM-SM、PIM-DM、DVMRP、DOSPF 等多播路由协议决定▼。

其次，第 2 个作用也被称作 IGMP（MLD）探听。通常交换集线器只会习得单播地址▼。而多播帧▼则跟广播帧一样不经过滤就会全部被拷贝到端口上。这会导致网络负荷加重，甚至给那些通过多播实现高质量图像传播的广播电视带来严重影响。

为了解决此类问题，可以采用作为第二个作用的 IGMP（MLD）探听。支持 IGMP（MLD）探听的交换集线器可以过滤多播帧，从而也能降低网络的负荷。

在 IGMP（MLD）探听中，交换集线器对所通过的 IGMP（MLD）包进行监控▼。由于从 IGMP（MLD）包中可获知多播发送的地址和端口，从而不会再向毫无关系的端口发送多播帧。这也可以减轻那些不接收多播消息的端口的负荷。

▼ Multicast Listener Discovery。多播监听发现。ICMPv6 的类型 130、131、132。

▼ Internet Group Management Protocol

▼关于 ICMPv6 的更多细节请参考 5.4.4 节。

▼关于单播路由协议可参考第 7 章。

▼通常交换集线器可以会习得发送端的 MAC 地址。而由于多播地址只用于目标地址，因此无法从包中习得。

▼指目标 MAC 地址是多播地址的意思。如图 3.5 所示，第 1 比特位为 1。

▼ IGMP（MLD）包由 IP（IPv6）的包进行传送，而非数据链层的包。支持 IGMP（MLD）的交换集线器不仅需要解析数据链路层的包，还得能够解析 IP（IPv6）和 IGMP（MLD）的包。之所以称为“探听”（snooping）也正是因为它需要监控“职责”以外的包。

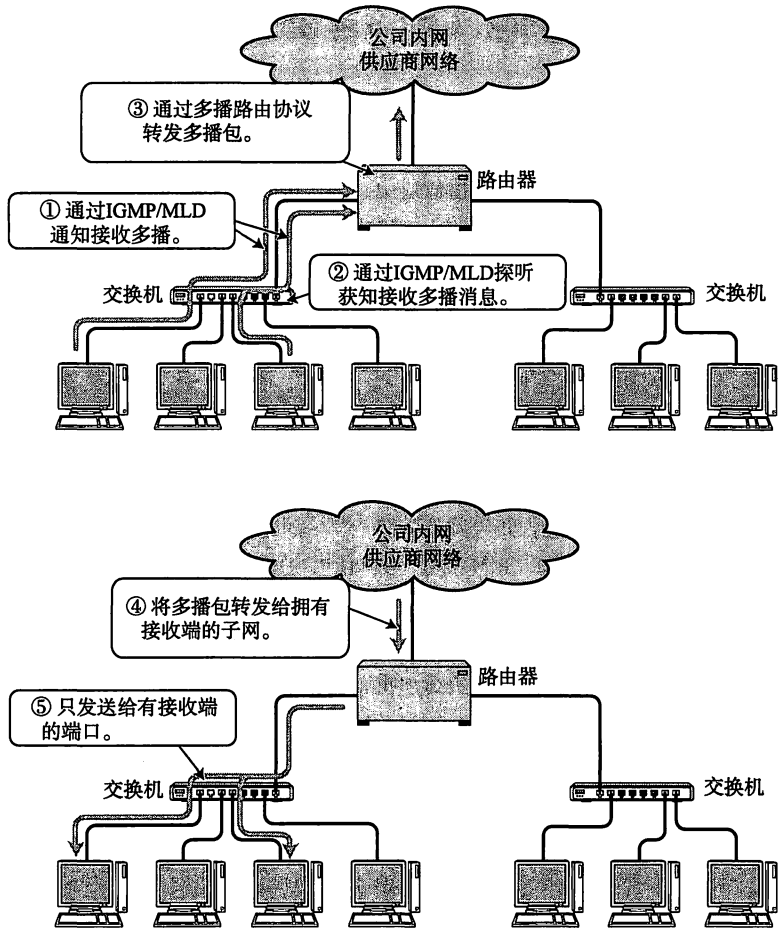


图5.25
基于IGMP（MLD）的多播实现

5.8.2　IP任播

IP任播主要用于报警电话110与消防电话119系统。当人们拨打110或119时，其接收电话并不是只有一个，而是可以拨打到一个区域管辖范围内的所有公安或消防部门。省、市、县、乡等不同级别的区域都各自设置着110与119的急救电话，而且数量极其庞大。

这种机制的实现，在互联网上就是IP任播。

IP任播是指为那些提供同一种服务的服务器配置同一个IP地址，并与最近的服务器进行通信的一种方法▼。它可适用于IPv4和IPv6。

▼选择哪个服务器由路由协议的类型和设置方法决定。关于路由协议的更多细节请参考第7章。

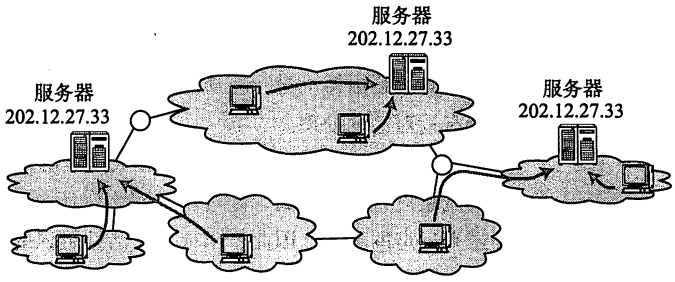
在IP任播的应用当中最为有名的当属DNS根域名服务器▼。DNS根域名服务器，出于历史原因，对IP地址的分类限制为13种类型。从负载均衡与灾备应对的角度来看，全世界根域名服务器不可能只设置13处。为此，使用IP任播可以让更多的DNS根域名服务器散布到世界的各个角落。因此，当发送一个请求包给DNS根域名服务器时，一个适当区域的IP地址也将被发送出去，从而可以从这个服务器获得应答。

▼可参考5.2.3节中的“域名服务器”一节。

IP任播机制虽然听起来非常方便，实际上也有不少限制。例如，它无法保证

将第一个包和第二个包发送给同一个主机。这在面向非连接的 UDP 发出请求而无需应答的情况下没有问题，但是对于面向连接的 TCP 通信或在 UDP 中要求通过连续的多个包进行通信的情况，就显得力不从心了。

图 5.26
IP 任播



IP任播中多个服务器设有同一个IP地址。当客户端发出请求时，可以由一个离客户端最近的服务器进行处理。

5.8.3 通信质量控制

通信质量的定义

近些年，IP 协议的实用性被认可，并应用于各种各样的通信领域中。IP 协议的设计和开发初衷是作为一个“尽力服务”的协议，是一款“没有通信服务质量保证”的协议。在“尽力服务”型的通信中，如果遇到通信线路拥堵的情况，可能会导致通信性能下降。这就好比在高速公路上，如果一下子有太多的车辆涌入高速，将会导致堵车，谁也无法确保何时能够达到目的地。“尽力服务”型网络中也存在此类问题。

通信线路上的拥塞也叫做收敛。当网络发生收敛时，路由器和集线器（交换集线器）的队列▼（Buffer）溢出，会出现大量的丢包现象，从而极端影响通信性能。这时如果正在访问 Web 页面，可能会出现点击任何链接都迟迟无法显示，或声音中断、视频画面停顿不前等现象。

▼ queue。等待队列。

近几年，特别是随着音频和视频服务对实时性要求的逐渐提高，在使用 IP 通信过程当中能够保证服务质量（QoS：Quality of Service）的技术受到了前所未有的追捧。

控制通信质量的机制

控制通信质量的工作机制类似于高速公路上的 VIP 通道。对于需要保证通信质量的包，路由器会进行特殊处理，并且在力所能及的范围之内对其进行优先处理。

通信质量包括带宽、延迟、时延波动等内容。路由器在内部的队列（缓存）中可以优先处理这些要求保证通信质量的包，有时甚至不得不丢弃那些没有优先级的包以保证通信质量。

为了控制通信质量，人们提出了 RSVP▼ 技术，它包括两个内容，一是提供点对点的详细优先控制（IntServ）另一个是提供相对较粗粒度的优先控制（DiffServ）。

▼ Resolution Reservation Protocol

IntServ

IntServ 是针对特定应用之间的通信进行质量控制的一种机制。这里的“特定

的应用”是指源IP地址、目标IP地址、源端口、目标端口以及协议号五项完全内容一致▼。

▼源端口与目标端口是TCP/UDPQN首部中的信息，具体可参考第6章。

IntServ所涉及的通信并非一直进行，只是在必要的时候进行。因此IntServ也只有在必要的时候才要求在路由器上进行设置，这也叫“流量设置”。实现这种流量控制的协议正是RSVP。RSVP中在接收端针对发送端传送控制包，并在它们之间所有的路由器上进行有质量控制的设定▼。路由器随后就根据这些设置对包进行有针对性的处理。

▼具体可以是带宽、延迟、时延波动（抖动）、丢包率，等。

不过RSVP的机制相对复杂，在大规模的网络中实施和应用比较困难。此外，如果流量设置要求过高，超过现有网络资源上限时，不仅会影响后续的使用，还会带来一定的不便。因此，出现了灵活性更强的DiffServ。

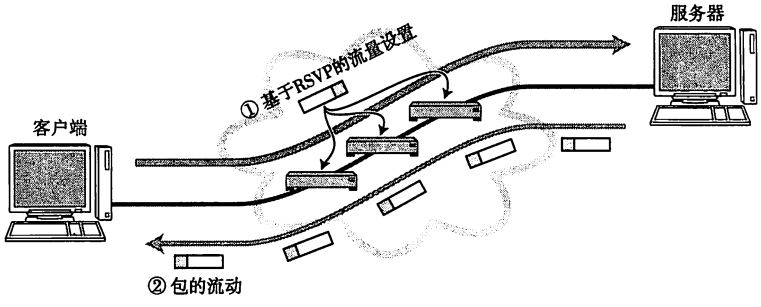


图5.27
RSVP中的流量设置

DiffServ

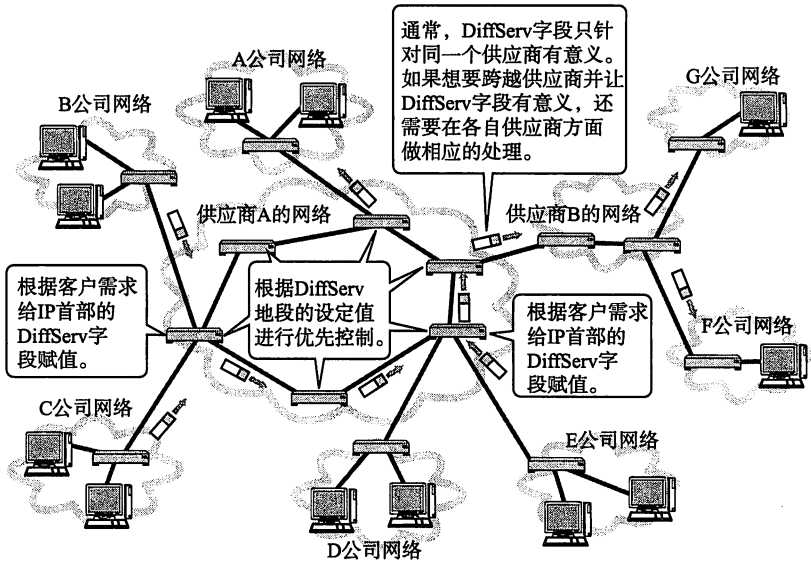
IntServ针对应用的连接进行详细的通信质量控制。相比之下，DiffServ则针对特定的网络进行较粗粒度的通信质量控制。例如，针对某个特定的供应商进行顾客排名，从而进行数据包的优先处理。

进行DiffServ质量控制的网络叫做DiffServ域。在DiffServ域中的路由器会对所有进入该域IP包首部中的DSCP▼字段进行替换。对于期望被优先处理的包设置一个优先值，对于没有这种期望的包设置无需优先的值。DiffServ域内部的路由器则根据IP首部的DSCP字段的值有选择性地进行优先处理。在发生网络拥塞时还可以丢弃优先级较低的包。

▼DSCP字段是IP首部TOS字段的替代。具体请参考4.7节。

IntServ中每进行一次通信都要设置一次流量设置。路由器也必须得针对不同流量进行质量控制，因此机制太过复杂，影响了实用性。而DiffServ则根据供应商的合约要求以比较粗粒度进行质量控制，机制相对简单，实用性较好。

图 5.28
DiffServ



5.8.4 显式拥塞通知

当发生网络拥塞时，发送主机应该减少数据包的发送量。作为 IP 上层协议，TCP 虽然也能控制网络拥塞，不过它是通过数据包的实际损坏情况来判断是否发生拥塞▼。然而这种方法并不能在数据包损坏之前减少数据包的发送量。

为了解决这个问题，人们在 IP 层新增了一种使用显式拥塞通知的机制，即 ECN▼。

ECN 为实现拥塞通知的功能，将 IP 首部的 TOS 字段置换为 ENC 字段，并在 TCP 首部的保留位中追加 CWR▼ 标志和 ECE▼ 标志。

通知拥塞的时候，要将当前的拥塞情况传达给那个发送数据包的源地址主机▼。然而，这个通知能不能发出去还是一个问题。而且，即使通知被发送出去，如果遇到一个不支持拥塞控制的协议▼，那么也就没有什么实质的意义。

因此，ECN 的机制概括起来就是在发送包的 IP 首部中记录路由器是否遇到拥塞，并在返回包的 TCP 首部中通知是否发生过拥塞。拥塞检查在网络层进行，而拥塞通知则在传输层进行，这两层的互相协助实现了拥塞通知的功能。

▼关于 TCP 拥塞控制请参考第 6 章。

▼ Explicit Congestion Notification，显式拥塞通知。

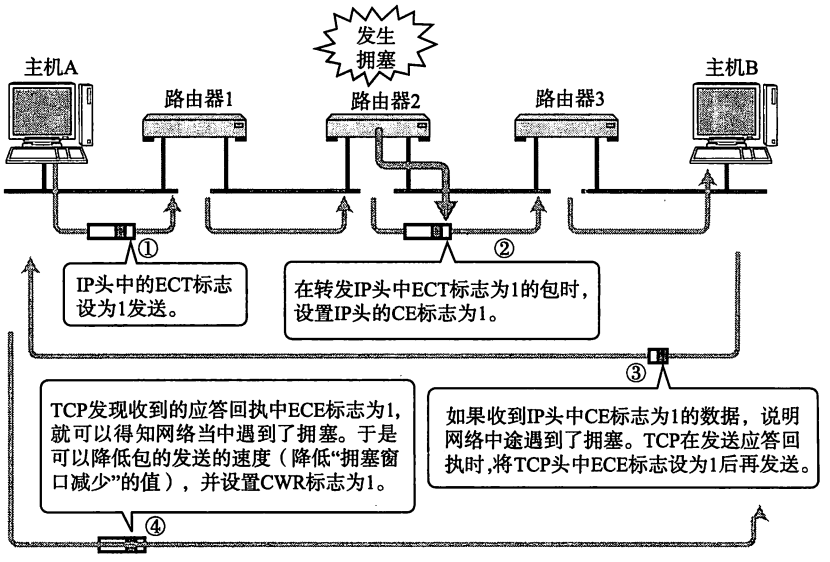
▼ Congestion Window Reduced，拥塞窗口减少。

▼ ECN-Echo

▼虽然 5.4.3 节介绍过的 ICMP 原点抑制消息正是由此产生的，但是实际上几乎从未被使用过。

▼例如使用 UDP 的通路等。

图5.29
拥塞通知



5.8.5　Mobile IP

Mobile IP 的定义

IP 地址由“网络地址”和“主机地址”两部分组成。其中“网络地址”表示全网中子网的位置，因此对于不同的地域它的值也会有所不同。

读者可以以智能手机和笔记本电脑等移动设备的情况做参考。通常，这些设备每连接到不同的子网，都会由 DHCP 或手动的方式分配到不同的 IP 地址。那么 IP 地址的变更会不会有什么问题呢?

与移动设备进行通信时，所连接的子网一旦发生变化，则无法通过 TCP 继续通信。这是因为 TCP 是面向连接的协议，自始至终都需要发送端和接收端主机的 IP 地址不发生变化。

在 UDP 的情况下也无法继续通信，不过鉴于 UDP 是面向非连接的协议，或许可以在应用层面上处理变更 IP 地址的问题▼。然而，改造所有应用让其适应 IP 地址变更不是件容易的事。

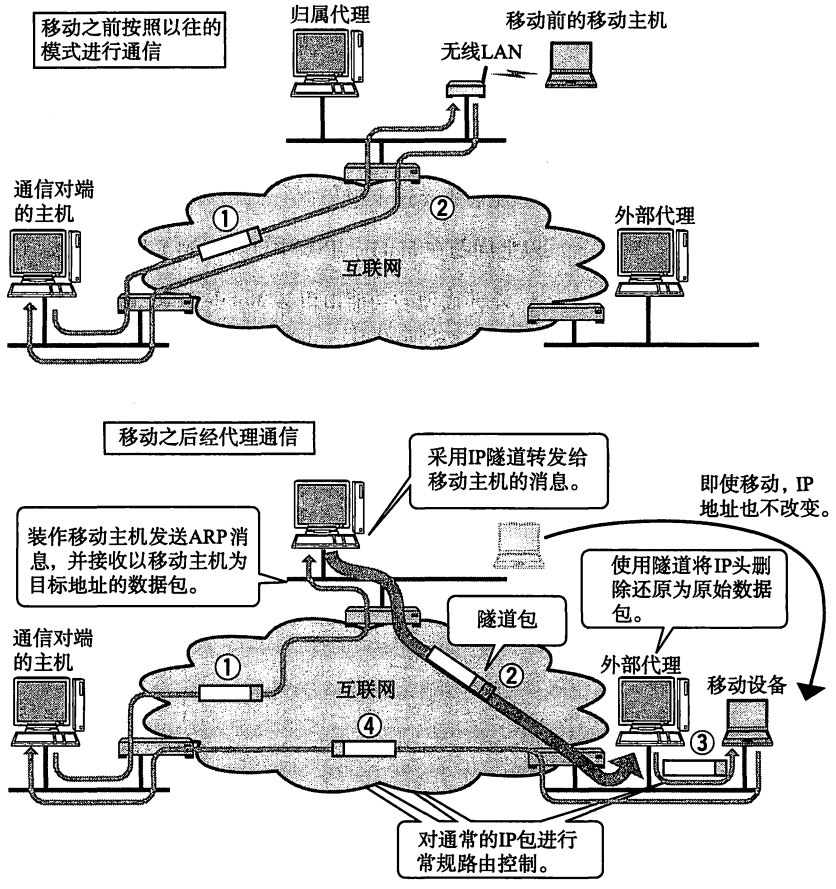
▼ TCP 的情况下，会断开 TCP 连接，不过通过修复等方法使应用上应对 IP 地址的变更也不是不可能的。

由此，Mobile IP 登上历史舞台。这种技术在主机所连接的子网 IP 发生变化时，主机 IP 地址仍保持不变。应用不需要做任何改动，即使是在 IP 地址发生变化的环境下，通信也能够继续。

IP 隧道与 Mobile IP

Mobile IP 的工作机制如图 5.30 所示。

图 5.30
Mobile IP



- 移动主机（MH：Mobile Host）
  是指那些移动了位置，IP 地址却不变的设备。在没移动的时候，所连接的网络叫做归属网络，IP 地址叫做归属地址。归属地址如同一个人的户籍，移动也不会改变地址。即使移动了也会被设置成所处子网中的 IP 地址。这种地址被称为移动地址（CoA：Care-of Address）。
- 归属代理（HA：Home Agent）
  处于归属网络下，可监控移动设备的位置，并转发数据包给移动主机。这很像注册户籍信息的政府机关。
- 外部代理（FA：Foreighn Agent）
  使用于支持移动主机的移动设备。所有需要接入网络的移动主机都需要它。

如图 5.30，Mobile IP 中的移动主机，在移动之前按照以往的模式进行通信，而移动之后则通过外部代理发送转发数据包向归属代理通知自己的地址。

从应用层看移动主机，会发现它永远使用归属地址进行通信。然而，实际上 Mobile IP 是使用转交地址转发数据包的。

Mobile IPv6

Mobile IP 中存在一些问题：

- 没有外部代理的网络不能通信。
- IP包呈三角形路径被转发因此效率不高。
- 为提高安全，一个域可以做这样的设置，即如果从自己的域向外部发送包的源地址不是本域在用的IP地址，则丢弃该包。而且这种设置已经越来越多。是因为从移动主机发给通信对端的IP包的源地址是归属地址，与另一个域的IP地址不符（如图5.30④中的IP包），因此目的地路由器可能会丢弃这个包▼。

▼为了避免该问题的发生，现在Mobile IP中移动主机向通信对端发送IP包时要经由归属代理，这也叫做双向隧道。事实上这种方式比三角形通路效率还低。

以上问题在Mobile IPv6中已经得到了相应的解决。

- 外部代理的功能由市县Mobile IPv6的移动主机自己承担。
- 考虑路径最优化，可以不用经过归属代理进行直接通信▼。
- IPv6首部的源地址中赋与移动地址，不让防火墙丢弃▼。

▼使用IPv6扩展首部中的"Mobility Header"（协议号135）。

▼使用IPv6扩展首部中的"目标地址选项"（协议号60）中的归属地址。

移动主机和通信对端的主机都需要支持Mobile IPv6▼才能使用以上所有功能。

▼由于IPv6的普及比较缓慢，今后支持IPv6的设备也支持Mobile IPv6的可能性非常高。