

第一部分 IP基础知识

第1章 为何要升级IP

我们都知道岁月的流逝并不会使一些美好的事物消失。但不幸的是，一些现在看来不错的事物并不意味着能够永远使用下去——无论它现在是多么的辉煌，它或者将会过时，或者将被开发殆尽，总会有新鲜的事物遮盖它原有的光芒。而当这种好的事物已经成为基础设施的一部分的时候，对它的维护变得非常重要，而了解何时对它进行升级以及如何以最少的混乱、最低的代价进行升级则显得尤其重要。

IP第4版作为网络的基础设施而广泛地应用在 Internet和难以计数的小型专用网络上，这就是著名的IPv4。IPv4是一个令人难以置信的成功协议，它可以把数十个或数百个网络上的数以百计或数以千计的主机连接在一起，并已经在全球 Internet上成功地连接了数以千万计的主机。IP协议诞生于70年代中期，可以用几种不同方式表示 IP的存在时间，本章以及本书的其他部分中将有更详细的描述。但是，就像被过度使用的桥或高速公路一样，IPv4已经走到了尽头并且必须马上升级。

本章将讨论以下议题：

- 什么是IPv4，为什么它如此重要？
- IPv4中存在哪些问题，它为什么需要升级？
- 为什么我们现在就需要修补 IP而不是等到将来？
- IP的升级对于用户、网络操作员、管理者和供货商究竟有哪些影响？

在本书中，IP用来指网际协议的各个版本，IPv4是指1998年及早些时候使用的IP。IPv6指的是由Internet工程任务组(IETF)制订的用来取代IPv4的新的IP版本，该协议公布在最近发表的IETF的RFC文档中。

1.1 IP的影响

元素、化合物及服务已经融入了我们(以及我们的父辈和祖父辈)的日常生活，但IP与此不同，在我们的印象中，它的使用还远不能像使用电力或道路网络一样的熟悉和必不可少。即便如此，无论是个人计算机产品还是大型主机产品，对于 IP的支持实质上已经成为新的计算机硬件、软件或网络设备最普遍的功能之一。网际协议及其相关协议已经取得了 IBM和苹果、微软、网景、Sun、Novell、康柏、莲花及所有其他主要计算机厂商的共识。本节将介绍以下问题：

- 究竟什么是IP？
- IP可以应用在哪些地方？
- 有多少人、多少计算机和网络在使用IP？

- 如果IP发生变化，我们能够预计到什么？

1.1.1 什么是IP

IP解决的最根本的问题是如何把网络连接在一起，也就是把计算机连接在一起，而且除了其他计算机的网络地址之外，这些连接起来的计算机无需了解任何的网络细节。这就有以下三个要求：首先，每个连接在“网络的网络”上的计算机必须具有唯一的标识；其次，所有计算机都能够与所有其他计算机以每个计算机都能识别的格式进行数据的收发；最后，一台计算机必须能够在了解另一计算机的网络地址后把数据可靠地传至对方，而无需了解对方计算机和网络的任何细节。IP实现了上述目标。详细的介绍参见第2章，本节将进行扼要(可能是非常简单)的介绍。

所谓“网络的网络”就是互连网络(internetwork)，也被简称为互联网(internet)。全球Internet与它们的区别在于它的第一个字母是大写的I。最近，内联网(intranet)逐渐取代互联网用于指称使用TCP/IP的机构网络。

TCP/IP网络协议集基于一个四层的网络互联模型来连接任意两个系统。最底层是物理层，位于物理层之上的是数据链路层，用于在网络媒体(如以太网电缆或无线发送器)上传输计算机格式的数据。这一层协议使得连接在同种媒体上的两个系统可以通信，但不能与未连接在同一媒体上的系统通信。换言之，所有连接在办公室的以太网集线器上的PC机之间可以在数据链路层直接进行通信，但也只有连接在该集线器上的计算机才能彼此通信。

在数据链路层，数据被发送到与计算机的网络接口相关联的地址。这意味着每个将计算机连接到网络的设备都有一个类似于序列号的地址：对该连接设备这个地址通常是唯一的，每个设备“侦听”目的地址与自己的地址相同的数据包。如果一个系统没有连接到特定网络上的设备，它就不能与网络上的其他系统在数据链路层上直接通信。

不在同一个物理网络上的系统不能在数据链路层直接进行通信的部分原因，在于连接在不同的网络上的计算机往往使用不同的协议。例如：使用令牌环网的计算机无法理解以太网上传输的数据。另一个原因是链接不同链路层协议的网络需要特殊类型的系统，这种系统被称为网关(gateway)。网关是一个同时连接两个或更多运行不同协议网络的计算机，它可以将来自一种数据链路层协议的数据翻译成另一种协议。但即便有了网关，仍然需要一些其他的办法来连接异构的网络。

数据链路层的上一层被称为网际层，正是在这一层，位于不同物理网络上的设备可以进行通信。每一个接口被分配了一个网际层地址，这个地址在连接在该互连网络上的所有系统中具备唯一性(使用IP连接到网络上的系统通常称为主机)。所有连接在同一个互连网络上的主机可以理解这些地址，并可以在必要时使用各种方法将这些地址与数据链路层的地址进行映射。路由器正是在Internet层发挥作用的：这些系统(也可以是网络协议网关)连接在两个或更多的网络上，并由连接到这些网络上的所有主机使用，以向远端网络上转发数据包。

一个需要全球唯一地址的网络示例是电话系统：每个电话用户必须具备一个唯一的电话号码。随着电话网络的扩展和用户数量的增加，电话公司用增加交换局和地区号来加长电话号码的做法并不少见。与电话号码不同，虽然IP地址也是由数字组成，但它既不能多于也不能少于32位。正如在美国使用的10位电话号码把电话用户的数量

限制在了 10^{10} 之内，32位地址限制了Internet的地址数量不能超过 2^{32} ，即接近于40亿。与电话号码一样，真正可用的地址少于理论值(在Internet地址中更少)，这主要是由于一些号码被保留或具备了特殊意义。地址空间的限制是IPv4的根本问题，本书将进一步讨论这个问题。

当一台主机需要向另一台主机发送数据时，它将检查目的主机的Internet地址。如果该地址与自己连接在同一物理网络上，则发送端主机简单地通过数据链路层将数据包发送至目的地。在这种情况下，以太网上的发送端主机将通过以太网传输直接到达目的主机。

但是，如果发送端主机发现目的方主机与自己不是连接在同一物理网络上，那么发送端主机将把数据发给与自己连接在同一个物理网络上的路由器。然后，该路由器判断数据的地址是否属于与自己直接连接的网络。如果是，该路由器将简单地把数据交给目的主机；如果该数据的地址不属于与自己直接连接的网络，该路由器将把数据转发给连接在其他网络上的路由器。如此继续，如果一切顺利的话，直到将数据最终交给与目的主机在同一物理网络上的路由器为止。

其他TCP/IP网络互联操作在传输层和应用层上完成。在传输层上，数据在通信系统的实际进程之间移动；在应用层上，数据在应用自身之间移动。这些层以及网络层将在第2章中详细讨论。

1.1.2 IP应用在哪些地方

许多年以来，只有在大学或研究机构的网络中才能找到IP的应用。而IP的商用产品直到80年代后期、90年代初期才出现，即使这样，这些产品仍被定位为专用产品。直到1995年，TCP/IP才被普遍引入到个人计算机产品中，因为从那时起，Novell和微软开始选择IP作为联网协议来支持其打印和文件服务的网络传输。

这意味着正在使用IP的不仅包括每个连接到Internet的计算机，还包括所有使用这些网络操作系统来访问机构资源的所有计算机，而不论这些计算机是否连接到Internet。

从手提式电脑到功能强大的超级计算机，目前使用的所有计算机几乎都支持IP。另一方面，IP也越来越多地用于连接其他设备，从而可以任意地使用网页浏览客户机访问内置网页服务器以实现家用电器和安全系统的远程控制。

使用IP的网络除了Internet之外还包括称作内联网的公司网络，其规模可以从一个办公室中连接在一起的几台主机到分布在全球范围内的所有分支机构的数以万计的主机。IP网的另一个特例是外联网(extranet)，它是出于某个共同目标在实体间提供安全连接的专用IP网。例如，外联网可用于把不同公司的成员连接成一个工作组或把需要传递订货和执行信息的商业伙伴连接起来(如需了解更多的关于外联网的信息，参见作者的另一本书《Extranet Design and Implementation》(SYBEX,1997))。

从计算机硬件和软件到家庭娱乐产品、移动电话，甚至支持无线Internet连接的汽车，这些支持IP的产品的数量体现出IP对于当今世界的通信基础设施的重要性。

1.1.3 有多少人在使用IP

“有多少人在使用连接到IP网络的系统”是一个复杂的问题。对于运行IP和连接到Internet的网络数量，曾经一度有一个简单的计算方法：可以根据由不同网络授权机构指派的网络地址数量作出判断。可这种方法并不能保证其正确性，因为它忽略了那些运行IP但没有连接

Internet的网络。

如今情况更加复杂, 因为一个网络地址可以分为子网, 由使用同一个 Internet服务供应商 (ISP) 的多个机构共享。与此类似, 还有很多机构在连接到 Internet时采用了网络地址翻译技术 (这种方法在内部主机 /网络与外部主机 /网络间使用网关作为媒介, 网关把只有内部网才能识别的内部网络地址翻译为外部网络能够识别的地址), 在官方的统计数据中同样也不会包含这种网络。以上两种技术将在后续章节进一步讨论, 使用这两种技术可以连接更多网络, 但同时加大了准确统计 Internet网络数量的难度。

即使网络数量的准确统计成为可能, 每个网络中包含的主机也并非都可计数。今天, 网络中越来越多的公司使用地址翻译技术和防火墙技术把公司的资源隐藏起来, 任何一个网络所连接的主机名字和数量对于公司外部的人都很难辨别和获得。

最后, 想要统计出通过这些系统访问 IP网络的个人用户数量更是难上加难。大型主机和超级计算机可能有数以百计甚至更多的用户, 同时还有一些用户使用多台计算机。结果是必须推测每台计算机上的平均用户数量, 实际的数目可能是高达每台计算机 300个用户也可能低至每台计算机 1/3个用户。

根据过去十年中令人目瞪口呆的增长速度, 研究分析者提出了一些不同的估计, 但毋庸置疑的是, Internet上肯定有数千万台主机, 且使用 IP的个人用户数量有一亿甚至更多。

1.1.4 当IP发生变化时会产生哪些影响

正如你看到的一样, IP的升级将影响许多人和机构。当从 IPv4向IPv6转变时, 可能会发生一些事情, 而这些都是需要网络管理员来应付。首先, 可能没有任何变化: 没有软件 /硬件升级、服务不变、一切不变, 只要网络管理员选择不进行任何升级或只升级与 Internet的连接。相反, 也可能有很大变化, 许多新的网络软件需要分发和配置, 新的应用需要安装和升级, 升级时出现的故障需要应付; 此外, 升级还会给用户、机构和网络管理员带来显著的好处。

过渡方案以及不同的过渡策略将在第 12章中进一步讨论。

1.2 IPv4的局限性及其缺点

在当前计算机工业飞速发展的步伐下, 指出 IPv4的局限性和缺点如同指出小汽车和卡车的内燃机是有缺陷的动力源一样。IP的确是一个非常强壮的协议, 并已经证明了它能够连接小至几个节点, 大至 Internet上难以计数的主机。为交通工具选择动力源时, 只要能像汽油机或柴油机一样提供动力, 任何人都可以使用包括电能、太阳能或是风能作为上路动力而不会影响别人, 与此不同的是, IP的升级将对所有使用IP的人产生重大影响。

TCP/IP的工程师和设计人员早在 80年代初期就意识到了升级的需求, 因为当时已经发现 IP地址空间随着 Internet的发展只能支持很短的时间。本节将介绍 IP必须升级的原因以及可以同时改进之处, 其中包括:

- 地址空间的局限性: IP地址空间的危机由来已久, 并正是升级的主要动力。
- 性能: 尽管IP表现得不错, 一些源自 20年甚至更早以前的设计还能够进一步改进。
- 安全性: 安全性一直被认为是由网络层以上的层负责, 但它现在已经成为 IP的下一个版本可以发挥作用的地方。
- 自动配置: 对于 IPv4节点的配置一直比较复杂, 而网络管理员与用户则更喜欢“即插即

用”，即：将计算机插在网络上然后就可以开始使用。IP主机移动性的增强也要求当主机在不同网络间移动和使用不同的网络接入点时能提供更好的配置支持。

1.2.1 IP地址空间危机

Internet经历了核爆炸般的发展，在过去的10到15年间，连接到Internet的网络数量每隔不到一年的时间就会增加一倍。但即便是这样的发展速度，也并不足以导致90年代后期IP地址的匮乏。

IP地址为32位长，经常以4个两位十六进制数字表示，也常常以4个0至255间的数字表示，数字间以小数点间隔。每个IP主机地址包括两部分：网络地址，用于指出该主机属于哪一个网络(属于同一个网络的主机使用同样的网络地址)；主机地址，它唯一地定义了网络上的主机。这种安排一方面是IP协议的长处所在，另一方面也导致了地址危机的产生。

由于IPv4的地址空间可能具有多于40亿的地址，有人可能会认为Internet很容易容纳数以亿计的主机，至少几年内仍可以应付连续的倍增。但是，这只适用于IP地址以顺序化分布的情况，即第一台主机的地址为1，第二台主机的地址为2，依此类推。通过使用分级地址格式，即每台主机首先依据它所连接的网络进行标识，IP可支持简单的选路协议，主机只需要了解彼此的IP地址，就可以将数据从一台主机上转移至另一台主机。这种分级地址把地址分配的工作交给了每个网络的管理者，从而不再需要中央授权机构为Internet上的每台主机指派地址。到网络外的数据依据网络地址进行选路，在数据到达目的主机所连接的路由器之前无需要了解主机地址。

通过中央授权机构顺序化地为每台主机指派地址可能会使地址指派更加高效，但是这几乎使所有其他的网络功能不可行。例如，选路将实质上不可行，因为这将要求每个中间路由器去查询中央数据库以确定向何处转发包，而且每个路由器都需要最新的Internet拓扑图获知向何处转发包。每一次主机的地址变动都将导致中央数据库的更新，因为需在其中修改或删除该主机的表项。

IP地址被分为五类，只有三类用于IP网络，这三类地址一度被认为足以应付将来的网络互联。A类地址只有126个，用于那些最大的实体，如政府机关，因为它们连接着最多的主机：理论上最多可达一千六百万台。B类地址大约16 000个，用于大型机构，如大学和大公司，理论上可支持超过65 000台主机。C类网络超过两百万个，每个网络上的主机数量不超过255个，用于使用IP网络的其他机构。

更小的公司，某些只有几台主机，它们对于C类地址的使用效率很低；而大型机构在寻找B类地址时却发现越来越难；那些幸运地获得A类地址的少数公司很少能够高效地使用它们的一千六百万个主机地址。这导致了在过去几年中一直使用的网络地址指派规程陷入了困境，在试图更有效地分发地址空间的同时，还要注意保存现有的未指派地址。与此同时，一些解决地址危机的办法开始得以广泛使用，其中包括无类域间选路(CIDR)、网络地址翻译和使用非选路网络地址。

IP寻址将在第2章中详细解释，而与IPv4地址短缺相关的问题、挑战和临时解决方案将在第3章中讨论，IPv6的地址空间将在第6章中详细介绍。

1.2.2 IP性能议题

IP刚开始时，从各方面看就像一个实验品，其主要目的在于为在异种网络间进行数据的

可靠、健壮和高效传输探索最佳机制，从而实现不同计算机的互操作。在很大程度上 IP 实现了此目标，但这并不意味着 IP 可以继续实现这些目标，也不意味着在对 IP 进行修改后而不能更好。在过去的几年中，很明显不仅 IP 有改进余地，同时新的开发也导致修改 IP 的呼声越来越高。在这次升级中考虑了最大传输单元、最大包长度、IP 头的设计、校验和的使用、IP 选项的应用等议题。针对这些议题已经提出了专门建议并已引入 IPv6 中，这将有利于提高 IPv6 的性能并改进 IPv6 作为继续高速发展的网络的基础的能力。

与 IPv4 性能相关的问题和挑战将在第 3 章中讨论，而 IPv6 的解决方法将在第 7 章和第 8 章中详细解释。

1.2.3 IP 安全性议题

刚开始时连入 Internet 的都是侧重于研究与开发的机构，至少其中的研究人员互相间了解各自的名声，而它们与军队和政府的密切关系也保证了安全性不是一个主要问题。更重要的是，很久以来人们认为安全性议题在网络协议栈的低层并不重要，应用安全性的责任仍交给应用层。在许多情况下，IPv4 设计为只具备最少的安全性选项，而 IPv6 的设计者们已在其中加入了安全性选项来强力支持 IP 的安全性。

IPv6 安全性的增强无疑将改进虚拟专用网 (VPN) 的互操作性。IPv6 的安全性特性中包括数据的加密与对于所传输的加密数据和未加密数据进行的身份验证。这些功能也许将被证实是有价值的，但其价值(和功效)将主要体现在政治上而不是技术上。

IPv4 的安全性议题将在第 3 章中有所介绍，而 IPv6 对于安全和身份验证的解决方案将在第 9 章中详细解释。

1.2.4 自动配置

在 IP 还很年轻时，大部分计算机被放在雕花地板的房间里且其价格超过了大多数人一年(甚至更长时间)的收入。这些系统无法搬到任何其他地方去：它们年复一年地放在一个房间或建筑物中，它们与 Internet 的连接基本上是静态的，极少改变。那时也没有 ISP，它们通过电话公司提供的线路来链接至其他网络或 Internet 骨干网。

现在事情有了些变化。有数百个 ISP 可供选择，如果对于用户系统与网络间的选路和转发没有影响的话，理论上用户可以在不同的 ISP 间切换，从而更好地利用不同的速率和服务。同样，越来越多用户的工作方式要求网络服务具有更大的移动性。他们可能在家中使用一个或多个系统，可能在世界各地使用所携带的膝上型或笔记本电脑，也可能使用办公室中的任何一部电脑。更复杂的事情在于，这些人可能不只受雇于一个雇主，也可能为多个雇主工作。即便是同一个人使用同一部计算机，该计算机也会频繁地升级或售出。

随着工作和计算机对于移动性要求的与日俱增，IP 也必须做出一些改变以适应这种需求。针对这个问题，IPv4 已经有了一些改变，动态主机配置协议 (DHCP) 可以允许系统在启动甚至只在需要时才通过服务器获取其正确和完整的 IP 网络配置。目前，主机(无论是移动的还是固定的)仍然依赖于到网络的单点连接。当用户携带笔记本电脑出差时，只需给其 ISP 打一个电话就可以恢复连接能力。如果该 ISP 不能提供区域外的免费长途号码，就需要打长途电话来拨入该 ISP。

但是，还可以进行更多的改进，IPv6 应该能够旁路到单一 ISP 的静态连接，让用户系统能

够检测到最近的网络网关并通过它进行连接。目前尚不清楚这个功能如何实现，这里暂不讨论，但IPv6将可能实现该功能，其技术细节将在第11章中解释。

1.3 紧迫感

对IP地址体系结构不足的官方认可可以参见1991年发布的RFC 1287，其中定义了IP在成长过程中遇到的问题。至少从1992年就已开始了网络地址的定量分配，那时候对新的B类地址提出了要求，而不足以使用B类地址的中型机构开始接受成块的C类地址(参见RFC 1366和RFC 1466)。

与最后一分钟(或更晚)才开始的为2000年问题所做的努力不同，IPv6的升级工作体现了多年来许多专职工程师和计算机科学家的努力。他们已完成的工作使Internet和其他IP网络继续高效地发挥作用并保持多年的增长。

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络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：

- 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。

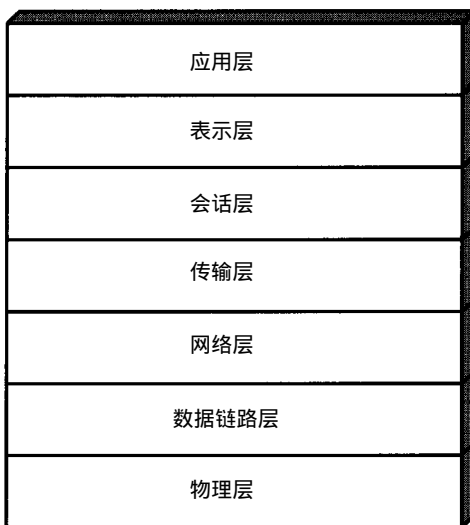


图2-1 网络互联的OSI模型提供了系统在网络上进行互操作的7个不同层

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，因为层越少就越

意味着交互越少，自然也就意味着连网实现更加高效。

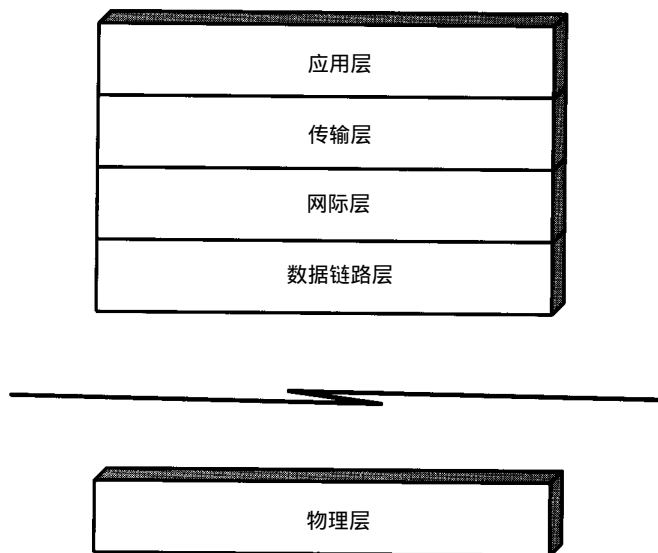


图2-2 Internet模型只用四层实现了无缝的、可互操作的网络互联

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，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。

版本	头长度	服务类型	数据报长度	
数据报ID			分段标志	分段偏移值
生存期	协议		校验和	
源IP地址				
目的IP地址				
IP选项(需要时填充)				
数据报的数据部分				
净荷				

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。

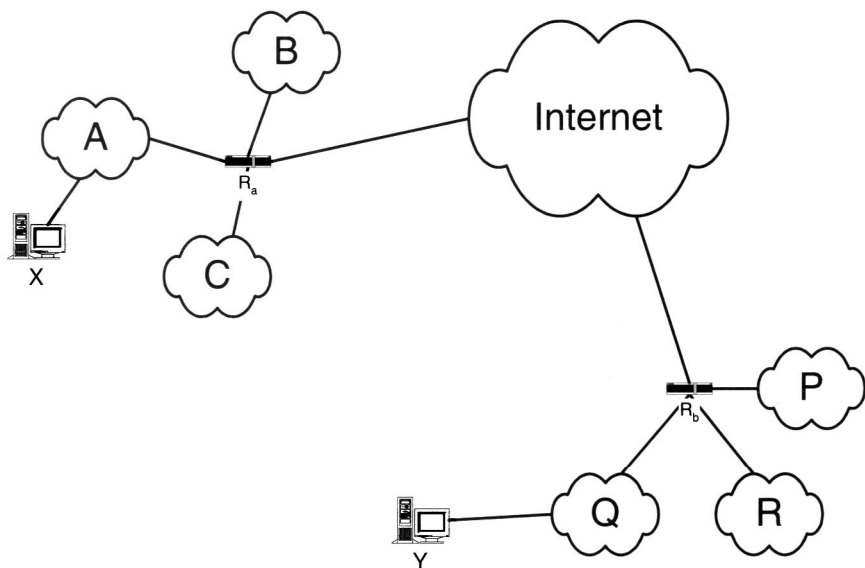


图2-4 IP路由工作过程

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。

第3章 IPv4的问题

本章主要讨论 IPv4 中存在的问题。虽然 IPv4 已经取得了令人难以置信的成功，但是仍有一些值得改进的地方。其中最显眼和最值得注意的可改进之处在于其地址空间的大小。其他议题与性能及 IP 头字段的设计和使用相关。本章还将讨论安全性、性能和管理控制等议题。

3.1 修改还是替换

考虑到 IPv4 存在的时间，它确实工作得不错。那为什么还要用其他的東西来替换它呢？毕竟如果把 IPv4 替换掉的话，网络中的所有系统均需要升级。升级到最新的微软 Windows 易如闲庭信步，但 IPv4 的升级对于大型组织来说，简直就是一场恶梦。我们讨论的网络可能包括十亿甚至更多的遍布全球的系统，上面运行着不知道多少种不同版本的 TCP/IP 连网软件、操作系统和硬件平台。要求对其中所有系统同时进行升级是不可想象的。

那么有没有办法可以避免 IP 升级可能带来的纷乱和不幸呢？答案是也许有，也许没有。这取决于对新协议的需求程度。换句话说，如果协议的唯一问题仅仅在于地址的匮乏，通过使用诸如后面所讨论的划分子网、网络地址翻译或无类域内选路等现有工具和技术，也许可以使该协议在相当长的时间内仍可继续工作。但是，这种权宜之计不可能长期有效，实际上，这些技术已经使用了很多年，如果不实现对 IP 的升级，它们最终将阻碍未来 Internet 的发展，因为它们限制了可连接的网络数和主机数。

本章还将讨论 IPv4 的其他问题，除了地址缺乏的问题外，还包括更普遍的扩展性问题、管理问题、选路困难、服务的改进和服务质量特性的交付以及安全性问题。

最后，拥有多年 IPv4 工作经验的工程师们作出的决定是替换而不是修补 IPv4。我们知道 IPv4 中哪些工作良好，哪些只是可以工作，哪些可以工作得更好。现在的情形不是用未知量来取代已知量。IPv6 的设计者们将这个新协议建立在 IPv4 的基础上，沿用 IPv4 工作良好的部分，改进可以工作的部分，去掉影响性能和功能的部分，另外还增加了当前特别需要的功能。

本节的其余部分讨论目前用于解决 IPv4 缺点的一些方法，然后讨论 IPv4 向 IPv6 升级的协议过渡的含义。

协议的补丁和扩展

IPv4 面临的最紧迫的问题是地址空间的大小问题，主要研究方向也定位在如何减少地址空间的浪费并提高使用效率上。其他议题，包括选路、网络管理、配置及 IPv4 扩展选项有时也与地址空间有关。

1. IPv4 选路

在互联网或内网上传输的 IPv4 包必须从一个网络选路到另一个网络以到达其目的地。选路协议可以使用动态机制来确定路由，但是所有选路最终依赖于某个路由器查看不同路由的列表并确定正确的路由。选路表包含网络的列表和连接到这些网络的接口的列表。路由器查看包，确定包所在的网络（或该网络可能在的网络），然后把包发送到适当的网络接口。

现在的关键在于路由表的长度将随着网络数量的增加而变长。而路由表越长，路由器在表中查询正确路由的时间就越长。如果只需要了解 10个、100个或1000个网络，这不是问题。但是诸如现在的 Internet，拥有大量的网络，在骨干路由器上通常携带超过 11万个不同网络地址的显式路由，此时选路变成了一场恶梦。

选路议题影响到性能，它对互联网增长的影响远比地址空间的匮乏更紧迫。IPv4地址可能在5年内使用殆尽，但如果不使用分级寻址来集聚和简化选路，Internet的性能可能在最近甚至现在就变得不可接受。

2. 划分子网

对子网的合理使用将增加地址使用的效率，但它对于效率的改进是有限的。如果了解原因的话，先考虑原来的网络地址分配方式：一个机构可以申请到一个 A、B或C类地址。如果能够证明自己需要相当数量的主机地址，机构也许能获得一个 B类地址；否则，获得的将会是一个C类地址。无论申请人的网络中的主机是 200台、20台还是2台，他们都将获得一个C类地址，这样就占用了 254个主机地址。如果他们能够使权威机构确信他们确实需要一个 B类地址的话，即便他们只有 1000台主机，他们仍将获得完整的 B类地址，这样一来又占用了 65 534 个主机地址。

获得这些地址后，从外部发往网络内任一处的业务流都在一个路由器接口处理，该路由器将把这些数据重新选路到本机构内的目的地。这种体系结构意味着用户可以按照自己的愿望来设计网络。图 3-1 中显示了两种方案。两个网络都连接到 Internet，但 C类网在本机构内只提供了一个网络的连接能力，而 B类网络把机构划分成三个子网，通过内部路由器彼此连接，并通过第二个路由器连接到 Internet。

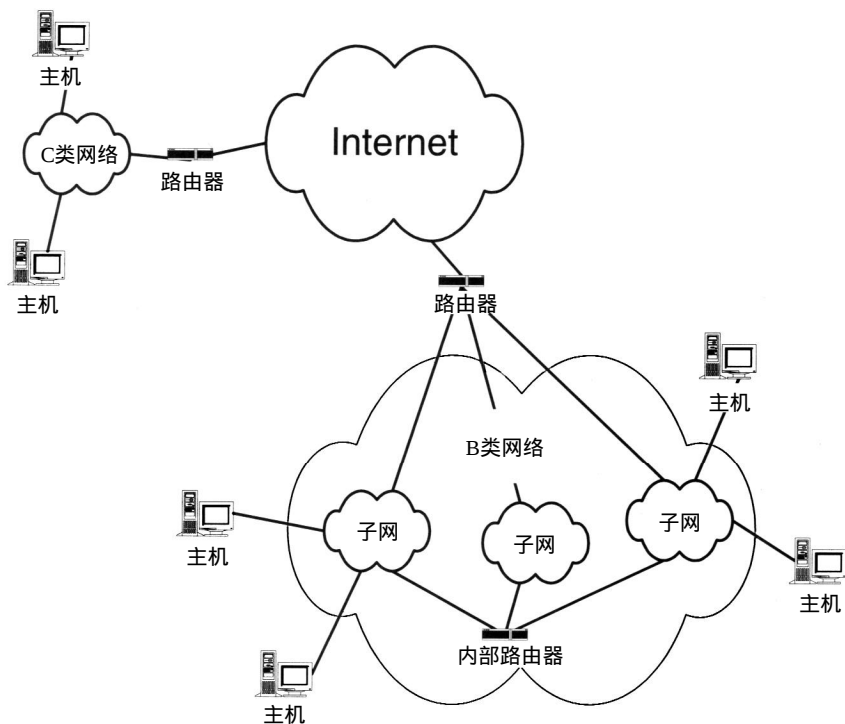


图3-1 子网有助于组织网络业务流，并改善网络地址使用的效率

当本地网络媒体在网络大小或连接的主机数量上到达极限时,就需要划分子网,同时它也可用来反应机构的体系结构。图中没有明确显示出子网不必在同一个建筑物内或同一个城市内。路由器重定向数据可以经过本地连接,也可以经过长途数据通信链路。这意味着一个机构可以与不同的分支、操作单位或子公司一起共享一个网络地址。

划分子网的问题在于它只适用于某种特定规模的机构——或者是C类网络或者是B类网络。例如,一个大型机构使用一个B类地址的网络有以下优点:使用8位子网掩码(换句话说,从B类地址的16位主机地址中借用8位)意味着可以有256个子网,其中每个子网可以最多有254个主机。如果使用9位子网掩码,还可以把子网数量倍增至512,当然每个子网上的主机数量降到了126。通过增加或减少位数量,可以很好的调整子网结构使之适应整个机构的体系结构。

对于需要B类地址的用户,不幸的是,除非是已经有了B类网络,否则目前很难得到B类地址。地址授权机构现在将C类地址成块分配给ISP,并通过它们再重新分配给用户。一个C类网络最多只能处理254台主机(绝对是最大值),如果划分子网后,其主机数将进一步减少。因此,划分子网的C类网络可以用于包含不超过8(或16)个子网的小公司,同时每个子网上的主机数量少于30个(或14个)。即便如此,这两种配置还是把网络能够容纳的主机数量限制为不能超过210个,降低了地址分配的效率。

子网有助于在一个机构内组织其业务流,同时可以使来自外部源的数据报的选路更简单。外部源无需知道目的子网的任何情况,因为所有的子网是在同一个网络地址下,且所有去往该网络中任何地址的数据报都要首先经过一个路由器,然后由该路由器决定把数据向哪个子网发送。

划分子网的有趣特点在于可以对一个已经划分子网的网络进一步划分子网。在图3-2中,一个B类网络被分为三层。第一层的路由器连接在Internet上,没有子网操作。但是,在该机构内,Internet路由器意识到有4位用于子网。这意味着最多可能有16个子网;那些子网中的任何一个都可以像图中所示一样进一步划分子网。在这个例子中,它们每一个都为最低层的子网又使用了4位,但机构内部的不同组可以选择不同的方法来分配其地址。例如,一个组具有多个小组但每个小组中的主机数量较少,可以使用6位作为子网,此时使得子网掩码的总长度达到10位;而另一个具有较少分支但每个小组都比较大,因此只使用了3位,从而使总的子网掩码达到7位。

3. 无类域间选路

无类域间选路(CIDR)技术有时也被称为超网,它把划分子网的概念向相反的方向作了扩展:通过借用前三个字节的几位可以把多个连续的C类地址集聚在一起。换句话说,就像所有到达某个B类地址的数据都将发给某个路由器一样,所有到达某一块C类地址的数据都将被选路至某个路由器上。

称做无类选路的原因在于它使得路由器可以忽略网络类别(C类)地址,并可以在决定如何转发数据报时向前再多看几位。另外一个与子网划分不同的特点在于,对于外部网络来说,子网掩码是不可见的;而超网路径的使用主要是为了减少路由器上的路由表项数。例如,一个ISP可以获得一块256个C类地址。这可以认为与B类地址相同,只不过前3位不是10x而是110。有了超网后,路由器可设定为包含地址块的前16位,然后把地址块作为有8位超网的一条路由来处理,而不再是为其中包含的每个C类地址处理最多可能256项路由。由于ISP经常负责为他们的客户的网络提供路由,于是他们获得的通常就是这种地址块,从而所有发往其客

户网络的数据可以由ISP的路由器以任何一种方式选路。

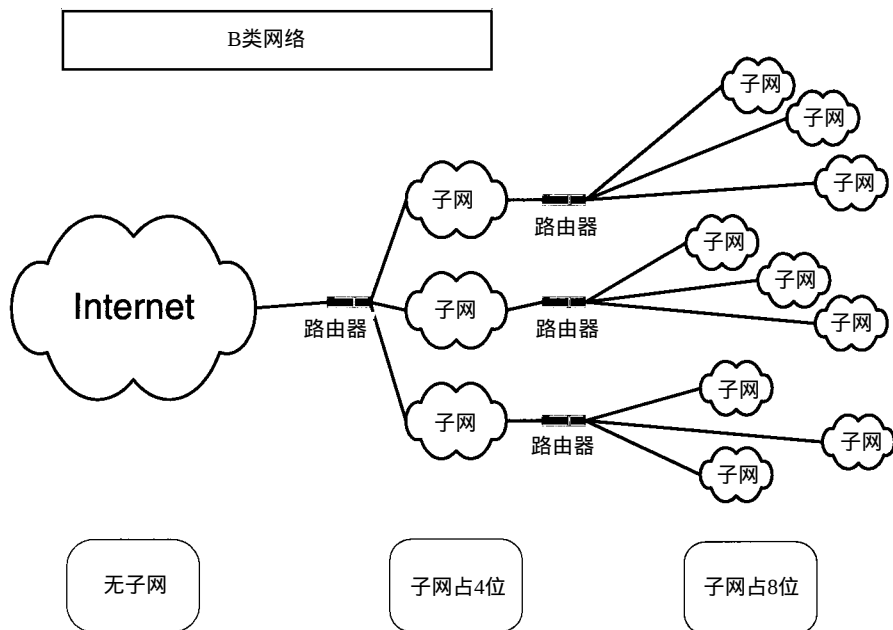


图3-2 子网可以进一步划分，产生更复杂的网络

由于B类网络相对缺乏和C类网络相对富余，这种把C类地址捆在一起的方法对于中等规模的机构来说很有用。此外，CIDR还缩短了路由表，这大大增加了选路的效率。但是，虽然CIDR增强了网络地址分配的效率，可它却并不能增加IPv4下总的主机数量，因此这只是一种短期解决办法而不是对于IPv4问题的长期解决方案。

4. 网络地址翻译

网络向外泄露的信息越少，网络的安全性就越高。对于TCP/IP网络来说，这意味着可能需要在内部网络和外部网络间设立一个防火墙，由它来接收所有请求。既然内部主机与外部主机失去了直接联系，那么IP地址就无所谓全球唯一，换句话说，如果内部主机不需要由Internet上的主机直接寻址，那么就可以为它们任意选择一个IP地址。实际上，许多与Internet没有任何联系的机构采用的就是这种方法。但当他们确实需要把二者连接在一起时，就必需对所有主机重新编号。

曾经有一段时间，许多公司无论是否打算连接Internet，都急于先申请到一段全球唯一的地址，因为这样可以使它们将来不必为主机重新编号。但是，随着专用IP网络的发展，为避免减少可分配的IP地址，有一组IP地址被拿出来专门用于专用IP网络。任何一个专用IP网络均可以使用包括一个A类地址(10.0.0.0)、16个B类地址(从172.16.0.0到172.31.0.0)和256个C类地址(从192.168.0.0到192.168.255.0)在内的任何地址。同时正如RFC 1918中的定义，把这些专用网络连接到公用网络的路由器不转发该网络上的任何数据。

网络地址翻译(NAT)在专用网络和公用网络之间的接口实现，该系统(一般是防火墙或路由器)了解专用网络上所有主机的地址，并将其翻译为可访问的公用网络地址，这样所有的内部主机就可以与外部主机通信。

虽然这种办法对于提高IP地址的分配效率有所帮助，但是网络设计人员在决定一个网络

是否使用 NAT 之前必须非常小心，要先确定其是否适用。对于那些永远不需要与其他网络合并或直接访问公用网络的网络，NAT 很适合。潜艇上的 IP 网络就可能非常适合使用专用地址：它不太可能与另一艘潜艇上的网络合并在一起，也不太可能需要直接连接到其他网络或公用网络。如果两个以上使用专用网络地址的网络需要合并，例如两个使用专用 IP 地址的银行要把他们的 ATM 机合并，那么最终形成的网络很可能需要进行重新编号以避免 IP 地址的冲突。

NAT 为一些小型机构提供了一种自己管理其地址空间的简单方法，无需依赖于地址授权机构为他们现在及将来的需要来分配足够的地址空间。NAT 还使得一些机构可以非常快速和灵活地定义临时地址或真正的专用网络地址。与 CIDR 不同，NAT 确实提供了一种可以真正减少 IP 地址需求的办法，尽管它使用起来有很大随意性，并且在重新对专用 IP 网络编址时将花费较长的时间和昂贵的代价。

5. 网络管理与配置

设计 IPv4 和大多数其他 TCP/IP 应用协议集的目的都不是易于使用。例如，原始的文件传送协议 (FTP) 依靠的就是非常神秘的请求和响应代码，并使用了类似天书般的命令。提到这一点的原因在于：实际上设计这些显然很复杂的命令和控制机制的目的是实现跨平台的标准化，并简化对理解这些协议的软件的访问。一个使用 IPv4 的系统必须使用一组特别复杂的参数来进行正确的配置，其中一般包括：主机名、IP 地址、子网掩码、默认路由器及其他（根据应用而有所不同）。这种做法很复杂，意味着进行这些配置的人必须理解所有这些参数，或者至少由真正理解它的人来提供这些参数。这意味着将一个系统连接到 IPv4 网络将十分复杂、非常耗时且代价高昂。

启动协议 (BOOTP) 是将主机连接到网络的简化过程的第一步。这个相对比较简单 的协议为只具备极少配置信息的主机（通常是一个简单的终端）提供了到 BOOTP 服务器获取其 IP 配置信息的方法。由于它只提供了将 IP 地址及其他配置信息与链路层地址（例如以太网卡地址）映射的方法，故它并不足以解决所有问题。要使用 BOOTP 管理 100 台主机，则必须为每台主机指定其 IP 地址。

地址管理和主机配置提出了至少两大问题：首先，如果配置主机很困难，将耗费钱财；其次，如果无论是否已连接，均为每个主机捆绑一个 IP 地址，这将浪费地址。如果可以使主机的配置变成即插即用，那将是一种好方法。即，只需把系统插到网络上，它将自动配置。在多个主机间共享 IP 地址也是一种好方法，如果有 100 台主机，在任意时刻同时上网的主机数不超过一半，那么只需使用 50 个 IP 地址让它们共享即可。

试图解决这些问题的结果是：在 BOOTP 框架之上构造了另一个称为动态主机配置协议 (DHCP) 的协议。它使用的仍然是客户机/服务器模型，与 BOOTP 一样，客户机可以使用 DHCP 来向服务器查询配置信息。但是，DHCP 更具灵活性，因为它可以随着 IP 地址的分配方法的不同而提供不同的配置信息。地址的分配有以下三种机制：

- 自动分配。主机申请 IP 地址，然后获得一个永久地址，可在每次连接网络时使用。
- 手工分配。服务器根据网络管理员提供的表格为每个主机分配一个特定 IP 地址。无论主机是否需要，这些地址都将被保留。
- 动态分配。服务器按照先来先服务的方法分配 IP 地址，主机在一个特定时间范围内使用该 IP 地址，然后该地址“借用”期满。

无论是自动分配还是手工分配都可能使得 IP 地址分配效率很低。自动分配可能占用 IP 地址，如果一个机构的主机数量多于用户数量，使用这种方法将占用与主机数量相同的地址。手工分配意味着网络管理员必须为每个主机配置一个 IP 地址，而不管其连接网络的时间是一个小时还是一年。动态分配使得可以在一个大的用户数量的前提下共享少量的 IP 地址。

不幸的是，DHCP 由于其与状态相关的特性而无法实现真正的即插即用。用户不得不建立一个了解其主机的 DHCP 服务器，并且要使支持 DHCP 的主机了解最近的 DHCP 服务器。真正的即插即用，其实是移动性问题的一部分，而这正是 IPv4 不能支持的。下面我们会看到，IPv4 不具备支持移动性和网络管理能力，这也增强了升级到 IPv6 的呼声。

6. 服务类型

IP 使用的是包交换网络体系结构。这意味着包可以使用许多不同的路由到达目的地。这些路由的区别在于：有的代价比较高，有的吞吐量比较大，有的延时比较小，还有的可能会比其他的更可靠。第 2 章讨论的 IPv4 服务类型 (TOS) 字段，允许应用程序告诉 IP 如何处理其业务流。一个需要大吞吐量的应用，如 FTP，可以强制 TOS 为其选择具有更大吞吐量的路由；一个需要更快响应的应用，如 Telnet，可以强制 TOS 为其选择具有更小延时的路由。

这确实是一个好想法，但却从来没能在实际应用中真正实现。一方面，这需要选路协议彼此协作，除提供基于开销的最佳路由外还要提供可选路由的延时、吞吐量和可靠性的数值。另一方面，还需要应用开发者实现一个功能，使其可以提出可能影响性能的服务请求。TOS 是一种选择，如果用户认为低延时对于其应用最重要，则应用的吞吐量或可靠性将受到影响。

7. IP 选项

第 2 章中曾提到，IPv4 头包含了一个可变长的选项字段。IP 选项用于指示一些特殊的功能。在最初的规范中没有定义这些选项，但最终增加了关于安全性和选路功能的选项。选路选项中包括一个记录路由的功能，让每个处理包的路由器都将自己的地址记录到该包中，另一个时间戳功能让每个路由器在包中记录自己的地址和处理包的时间。另外还有源选路选项：“宽松源选路”指明包在发往其目的地的过程中必须经过的一组路由器，而“严格源选路”则指定了该包只能由列出的路由器处理。

IP 选项的问题在于它们是特例。大多数 IP 数据报不包括选项，并且厂商按不包括选项的数据报来优化路由器。IP 头如果不包括选项，则 5 字节长，易于处理，尤其是在路由器设计优化了对这种头的处理之后。对于路由器的销售而言，性能是关键，且由于大部分数据报不支持 IP 选项，因此路由器往往把这种包作为特例，搁置起来，只有在不会影响路由器总体性能时才加以处理。

尽管使用 IPv4 选项有很多好处，但由于它们对于性能的影响已使得它们很少使用。

8. IPv4 安全性

很长时间以来，都认为安全性不是网际层的任务。在这种情况下，安全性意味着对净荷数据的加密。其他安全性概念还包括对净荷的数字签名、密钥交换、实体的身份验证和资源的访问控制。这些功能一般由较高层处理，通常是应用层，有时是传输层。例如，广泛应用的安全套接字层 (SSL) 协议由 IP 之上的传输层处理，而应用相对较少的安全 HTTP (SHTTP) 则由应用层处理。

最近，随着虚拟专用网 (VPN) 软件和硬件产品的引入，安全隧道协议和机制有所扩展。这些产品通常会对于一个 IP 数据报流加密，即把这些包本身作为另一些 IP 数据报的净荷。IP 数据报

可想象为一个包装好的盒子，里面还包含着一个盒子，在小盒子中还包含另一个盒子。最小的盒子中包含的是应用数据，下一个盒子中是传输层数据，而最外面的盒子包含 IP 数据。实际上隧道的工作方法就是把一个 IP 盒子放在不同地址信息的另一个 IP 盒子中。

隧道协议，如微软的点到点隧道协议 (PPTP)，首先对 IP 数据报加密，然后打包，再发送到隧道上。

所有这些关于 IP 安全性的办法都有问题。首先，在应用层进行加密使很多信息被公开。尽管应用层数据本身是加密的，携带它的 IP 数据仍会泄露参与处理的进程和系统的信息。在传输层加密要好一些，并且 SSL 为 Web 的安全性工作得很好，但它要求客户机和服务器应用程序都要重写以支持 SSL。隧道协议也工作得不错，但却被缺乏标准的问题所困扰。

IETF 的 IP 安全性 (IPsec) 工作组一直致力于设计一种机制和协议来同时保证 IPv4 和 IPv6 业务流的安全性。虽然已有一些基于 IP 选项的关于 IPv4 安全性的机制，但在实际应用中并不成功。IPsec 的目标是使这些工具可用，并在 IPv6 中集成更加完整的安全性。

3.2 过渡还是不过渡

毫无疑问，IPv4 需要一些改变以使得它能够在网络协议的发展中得以继续生存。增长中的网络正在消耗有限的 IP 地址空间资源，这一简单问题意味着地址空间必须扩充。前一节中列举了一些可以帮助 IPv4 延长生命的著名方法，但是众所周知那不过是临时的办法。现在已经清楚，地址问题并不是 IPv4 中存在的唯一问题：网络越多意味着路由表越大，同时还导致路由器的性能下降。同样，难以实现 IPv4 选项意味着这些选项中实现的功能对用户不可用。

考虑一下如果只是简单地倍增 IP 地址的长度而不修改协议的其他部分，将会发生什么。所有的 TCP/IP 协议栈将需要同时被更新。落在后面的人将丧失与 Internet 连接的能力。尽管这种改变已经相对简单，但是由于错误配置而导致的系统瘫痪仍将产生巨大影响。对于任何人来说这种改变的代价都是巨大的，因为它意味着使用 IPv4 的所有机构都需要定位系统中的每一台主机，对于拥有许多用户和主机的大型机构，这绝不是一件简单的事情。更复杂的是，那些系统中有许多是比较老的或过时的甚至是已经废弃的系统，在这些系统上运行的网络软件可能已经过期并且没有人再提供支持。

任何对于现有系统进行升级的请求都可能导致混乱。对于 IPv4 的修补，无论是临时加入一个补丁还是用另一个重新设计的协议来替换，都将导致混乱。既然与其他方法相比，升级不会带来更多的痛苦，那么在可能存在一个更强健的修补方案时，何必再使用一个一个单独的补丁呢？

IPv6 协议规范在 1995 年底提交 IETF 并获得批准。软件厂商最早在 1996 年就已经开始提供 IPv6 网络协议栈的测试版。1997 年，测试产品和实验性的 IPv6 骨干网 (6BONE) 已经就位，但是到了 1998 年升级的势头缓慢下来。无论如何，最近还无法确定一个明确的“交割”日期。相反，将逐渐出现向 IPv6 过渡 (后续章节讨论)，IPv4 与 IPv6 将共存并交互。

向 IPv6 的过渡最有可能从高端而不是从最终用户开始，即一些机构和 ISP 可能会先在其骨干网络中首先实现 IPv6。即便如此，这些机构中也会有一部分会先去解决两千年问题，从而进一步降低过渡的速度。但是不管怎么说，只要应用开发者开始递交基于 IPv6 的新产品，那么向 IPv6 发展的速度就将大大加快。尽管与两千年问题相比，该问题在最终期限上具有较大的灵活性，IPv6 的计划也不容拖延太久。

第4章 通向IPng之路

IPng, 也称为下一代 IP, 目前并没有完全定型, 它将与之前的几个建议一起随时间的推移而发展。本章回顾了升级 IPv4所做的种种努力, 并将讨论这些努力对于 IPv6的形成有哪些影响。

4.1 概念的诞生

90年代初期, 很显然 Internet已经超出了协议所能控制的范围。1991年12月发布的 RFC 1287, 其标题是“未来的 Internet体系结构”, 其中列出了 1991年1月的 Internet Activities Board(IAB, 后来被称为 Internet Architecture Board)和 Internet Engineering Steering Group(IESG)会议上确定的发展方向, 其中包括对于 Internet将来的基本估计和 Internet协议中需要改进的最重要的领域。

4.1.1 对Internet将来的估计

以下给出了四个预测, 这些预测是针对未来五到十年中网络互联的趋势的分析。对于将来的网络环境达成一致认同有利于做出正确规划。预测如下:

- TCP/IP协议集将在一定时间内与 OSI模型并存。国际标准化组织 (ISO)开发了开放系统互联结构(参见第2章中对于 OSI七层模型的讨论)。尽管 TCP/IP在市场上获得了成功并赢得了广泛的接受, 但 OSI模型将在一定时间内继续发挥巨大的影响。
- Internet将变得更加复杂, 需要与种类繁多的不同网络技术协同工作。即, 与仅仅依靠一种或几种网络连接介质不同, 将会有越来越多的网络连接介质诞生并长时间使用。
- 对于 Internet的访问将由许多承载商一起提供, 其中包括提供多种网络的公用和专用供应商。换句话说, 多种不同类型的单位, 包括公司、政府机构、教学组织和公用事业的网络, 既可以通过电信业务供应商也可以通过个人拥有的网络进行连接。
- Internet需要能够支持多达十亿个网络的互联, 具体数量可能在一千万到一百亿个网络之间。

我们从 1998年开始对这些预测做一个回顾就可以发现它们工作得的确不错。其中最重大的预测是 OSI模型将继续保持其重要性, 虽然近年来 OSI模型在 IP中已应用很少, 但它并没有消失。随着新的网络技术, 如异步传送模式 (ATM)和 xDSL(对于不同的数字用户环路服务的统称)逐渐普及, Internet确实变得越来越复杂。同时网络连接的多样性也正如预测一样增长起来。

虽然 IPv4地址空间不够已经是不争的事实, 但需要互联的网络的具体数量目前仍不是很清楚。一方面, 在十亿个网络这种数量下, 可以为世界上的每一个公司和单位分配一个网络; 另一方面, 随着计算机的普及和费用的逐渐降低, 每个人拥有至少一个网络已经成为一种需求, 这就导致了世界上需要有至少一百亿个网络以用于个人。在某些不可见的因素影响下, 甚至可能会产生需要至少一万亿个网络的请求。

4.1.2 Internet发展中需要考虑的领域

1991年1月的IAB/IESG会议上还产生了另一个清单，其中列出了未来结构调整中最重要的领域。制定这份清单的目的在于定义将来的开发力量应当集中在哪些方面。其中包括：

- 选路与寻址问题。
- 多协议体系结构。
- 安全性体系结构。
- 流量控制和状态。
- 现代应用。

下面将讨论这些领域和它们的开发努力以及其他问题。

1. 寻址与选路

地址空间毫无疑问已经是一个问题，而路由表的膨胀也值得密切注意。RFC中指出，调用一个包含5000至7000项的路由表的时间将逐步影响快速发展的网络的性能。RFC 1287的作者指出，不仅IPv4的地址空间将被消耗殆尽，而且在那个时刻到来之前可能IPv4的路由算法已经无法适应如此大数量的网络。他们还指出，源与目的地之间的多路由可能会导致服务类型(TOS)的变化，并将需要一些机制来控制路由的选择。

通过一些机制对网络路由进行集聚的方法已被建议作为路由爆炸的一个可能的解决方案。通过某些办法在大的域之间划定边界将有助于提高路由效率。另一个建议是用一些高效的算法来进行路由的计算，同时在路由器上利用一些办法以某些特殊的路由方法保持特殊业务流的状态。

对于寻址方案可能的修改包括使用现有的32位地址作为一个非全球唯一的标识。即，在网络中不互通的部分间地址可以重用。例如，把全球分为几个不同的域，这使得一个主机地址在每个域中都可以被使用一次，而域间的互操作将通过协议网关在数据进行域间切换时重写其地址而进行。

另一种寻址方案是只增加主机地址字段的长度，并集成一个管理域作为网络地址的一部分。第三种方案是使用让路由器将主机地址与管理域映射的连接策略扩展主机地址字段，并将整个字段作为一个非层次地址空间。

2. 多协议体系结构

对于互操作的OSI传输与TCP/IP业务流的支持被认为是需要进一步开发的一个重要方面。这是因为直到1991年，Internet的连接仍然意味着一个主机必须具备一个Internet地址。如果没有一个IP地址并且没有运行IP，那么将不能上网。这种观点在1991年有了一些变化，RFC 1287的作者建议连接可以通过电子邮件网关或者更简单一些，通过某些应用来访问Internet。例如，NetWare网络上的用户可以在他们的系统上使用诸如网页浏览器和电子邮件客户机之类的Internet应用，但同时使用网络互联包交换(IPX)协议在他们本地的Novell Netware网络上传输数据。

实际上，1990年大多数硬件和软件供应商放弃竞争，而改为接受TCP/IP作为网络互联协议集。甚至1998年Novell公司开发的Netware网络操作系统也属于TCP/IP产品。

更重要的是，至少从事后看来，TCP/IP可以包含或借鉴其他协议。而互操作性，尤其是应用之间而不是低层之间的互操作性，则被认为是一件好事。

3. 安全性体系结构

国防部对于重点研究和开发工作的投资导致了 IP 的产生,这也意味着 IP 中天生就具有国防安全性方面的考虑。但是,商用 Internet 在安全性需求上与军队的网络有所不同,并且向一个协议集中添加安全性要比从头建设一个安全性协议难得多。

安全服务的一个建议是根据不同的用户名 (在 X.500 协议中定义的一个 OSI 架构) 进行身份验证并加以访问控制。同时还提出了关于一致性的强制措施,其中包含了一些方法来防止传输过程中数据被修改以及对于传输源的欺骗和抵御重播攻击 (一个窃听者把从经过身份验证的用户处偷来的数据再次传输)。其他的服务包括保密性 (对数据进行加密)、不可再现性 (使用数字签名来防止发送方拒绝承认发送了某段数据) 和通过拒绝对于某些服务的攻击以实现保护。

在该 RFC 中提出的其他安全性问题还包括路由器 / 网关上的协议过滤以及对于密钥的管理和存储的加密。

4. 流量控制与状态

IPv4 是一个无连接协议,但一些进程——例如语音和图像——需要一定程度的流量控制以正常工作。一个图像流必须以一个相对稳定的速率到达其目的地,不能太快也不能太慢,否则它的工作将不正常。RFC 1287 中定义了一些包排队机制以进行必要的流量控制,同时还定义了一些保持不同流的状态的方法来拓展 IP 使之更加适用于实时传输。

请注意在 IPv4 中定义了服务类型 (TOS) 域,但 TOS 不仅没有得到广泛应用而且现在连如何实现都尚不清楚。

5. 高级应用

RFC 的作者建议,与其考虑如何提出新的应用,还不如改进和简化开发新的和高级应用所需的过程,因为这样将带来更大的创造力。作为一个开始,他们建议创造用于不同类型数据的通用数据格式,尤其是文本、图像与图形、音频与视频、工作站显示与数据类等。对于开发高级应用而言,另一个重要的方面在于数据交换的机制。建议的机制中包括:存储转发业务、全球文件系统、进程间通信、数据广播和访问数据库的标准方法等。

4.2 第一回合

到 1994 年为止,已经出现了一些可以作为 IPv4 继承者的提案。IETF 在 1994 年考虑的三个主要提案其实在 1992 年便已开始成形。RFC 1347, 含有更多地址的 TCP 和 UDP (TUBA), 是其中之一。TUBA 是一个简单的 Internet 寻址和选路协议,可以认为是简单地用 OSI 网络互联协议和无连接网络协议 (CLNP) 替换了 IP。CLNP 中使用了网络服务访问点 (NSAP) 地址,该地址可以是任意长度,但通常为 20 字节,从而提供了足够的地址空间。除此之外,使用 CLNP 还可以帮助 IP 和 OSI 间进行会聚,并同时消除了建立一个完整的新协议的要求。

另一个提案在 1992 年以 IPv7 出现,并在 1993 年的 RFC 1475 中有更加详细的描述,其标题为“TP/IX: 下一代的 Internet”。目前不太清楚 TP/IX 的含义,根据 Christian Huitema 在《IPv6: The New Internet Protocol》(Prentice Hall PTR, 1998) 一书中的解释,该名字主要是为了表达该 RFC 的作者, Robert Ullman, 在修改 IP 的同时升级 TCP 的愿望。TP/IX 使用 64 位地址,并在分级结构中加入了位于各单位之上的寻址层以用于管理。IPv7 的 8 字节地址中有 3 字节用于管理域,3 字节用于各单位的网络,另 2 字节用于标识主机。IPv7 包头在对 IPv4 的包头进行简化的同时,也加入了转发路由标识符,使得中介路由器可以根据它来决定如何处理数据报。例如,可以根据在转发路由标识符中与该路由相关的值 (如吞吐量或价值) 来选择特定的路由,

也可以根据它来把数据放在特定的业务流中或把数据与一台移动的主机联系起来——这意味着，一台主机可以在从一个网络搬到另一网络上的同时保持其 TCP 连接。TP/IX 不仅对 TCP 和 UDP 协议作了修改，它同时还包括了一个叫做 RAP 的新的选路协议。

TP/IX 后来演变成了 RFC 1707 中定义的另一个协议，CATNIP：Internet 通用体系结构。但是 CATNIP 除了保持了 IPv7 的设计理念外似乎与 TP/IX 的共同点不多。为了提供一个通用的体系结构，CATNIP 标准中允许三种应用最广泛的体系结构：TCP/IP、OSI 和 IPX 的使用，并讨论了如何为下一代 IP 集成一个更具竞争力的标准。它的目标是使得所有业已存在的系统在各个主机均无需修改、地址无需变化、软件无需升级的情况下可以继续互通。通过允许使用不同的网络体系，CATNIP 将对实际基础设施的影响降到最小，但是，这也意味着需要通过增加一层的复杂性来实现真正的互联互通。

第三种提案某些时候被称为 IP 中的 IP，或 IP Encaps (即 IP 封装)。在这个提案中，IP 包含两层：一层用于全球骨干网络，而另一层用于比较有限的范围。在有限范围内仍然使用 IPv4，但骨干网络中使用不同地址的新的一层。后来这种提案不断演变并与其他协议相融合从而产生了简单增强 IP (SIPP)。RFC 1710 (简单增强 IP 白皮书) 对 SIPP 的历史做出了如下解释：

SIPP 工作组代表的是进行 IPng 开发的三个不同 IETF 工作组的进展。第一组被称为 IP 地址封装 (IPAE)，由 Dave Crocker 和 Robert Hinden 领导。该小组提出了对 IPv4 的一些扩展以携带更多的地址。他们的主要工作是研究过渡的办法。

后来 Steve Deering 提出了一个由 IPv4 发展而来的新建议，被称为简单 IP (SIP)。在 Steve Deering 和 Christian Huitema 的领导下成立了一个工作组进行基于该建议的研究。SIP 中包括 64 位地址，一个经过简化的包头及单独的扩展头中的选项。两个小组经过长期的互相接触达成了共识，即 IPAE 与 SIP 间有许多通用的元素，并且 IPAE 开发的过渡机制可以应用于 SIP，最终两个小组决定合并来集中他们的力量。新的 SIP 工作组的领导是 Steve Deering 和 Robert Hinden。

与 SIP 相并行，Paul Francis (即 Paul Tsuchiya) 也成立了一个工作组来开发 P IP (Pip)。Pip 是一个基于新的结构的 IP。Pip 背后的推动力在于引入了新的 IP，而且赋予了该机会非常重要的新的特征。Pip 支持以 16 位为单位的变长地址，地址间通过标识符进行区分，能够支持选择承载商、移动性和高效转发。其中包括了与 IPAE 类似的过渡机制。

经过 Pip 和 SIP 工作组领导层之间大量的讨论，他们逐渐意识到 Pip 中大量的先进特征可以由 SIP 在不修改基本的 SIP 协议且保持 IPAE 过渡机制的前提下完成。本质上使得各种协议的最佳特征都可以被保持。正是基于这一点，两个组决定合并以集中力量。新的协议被称为简单增强 IP (SIPP)。合并后的小组的主席是 Steve Deering，Paul Francis 和 Robert Hinden。

简而言之，SIPP 对 IPv4 进行了以下改动：

- 选路和寻址扩展：SIPP 定义了 64 位地址，倍增了 IPv4 地址空间。目的是为了在分层结构中提供更多级别，在每一层中可以分别完成各自的选路。另一个功能是加入了群地址，定义网络拓扑中的不同地区。SIPP 以 64 位为单位的地址扩展，及群地址均使得更大的地址空间成为可能。
- IP 头的简化：SIPP 去掉一些 IPv4 头字段的内容，同时对结构进行了重组使之有助于提高路由的效率。

- 对于选项实现的改进：SIPP使用了更加灵活的方法对IP选项进行编码和实现。
- 服务质量：SIPP使得对属于特定业务流的数据级进行标记成为可能。主机可以要求对于这些业务流采用特殊的处理，这一点对于像音频和视频传输这种需要实时递交的业务非常有用。
- 身份验证和保密：SIPP中加入了关于身份验证、数据一致性和保密性的内容。

SIPP是来自不同小组的人们共同工作的结果。在已完成的建议中包括了许多有趣的新方法，同时与升级IPv4而不是从头建立一个新协议的目标相差不远。值得注意的是其选路方法与IPv4很类似，仍然使用了CIDR来增加灵活性并提高选路的性能。另外一个重要的功能在于允许根据不同的选择标准(包括性能、开销、供应商对于流量的策略等等)对供应商进行选择。其他的选路扩展还包括对于移动主机的支持以及自动重新寻址和扩展寻址。

一个值得注意的机制是SIPP对于IP选项的处理：与把IP选项作为IP头的基本组成部分不同，SIPP中把IP选项与包头的主要部分进行了隔离。该选项头，如果有的话，将被放在包头后的数据报中并位于传输层协议头之前。使用这种方法后，路由器只有在必要的时候才会对选项头进行处理，这样一来就提高了对于所有数据进行处理的性能。

RFC 1710同时提供了SIPP技术规范的技术概述和对于协议必要性的证明和描述。如果只是为了知道IPv6来自何方的话，它也值得一读，因为SIPP经过一些修改之后，被IESG接受作为IPng的基础。

4.3 拾遗

RFC 1752(关于IP下一代协议的建议)于1995年发布。这是一份令人着迷的文件因为其中非常明确地指出了在IPv4的候选继任协议中哪些是我们需要的，哪些是我们已拥有的。在它的小结中，RFC 1752的作者指出了IPng将来的样子：

本协议提案中包括一个拥有分级地址结构的简化的头结构，从而允许进行严格的路由集聚，并且足以应付Internet在可见的将来产生的需求。本协议中也包括包一级的身份验证和加密功能以及即插即用的自动配置功能。这个设计改变了IP头选项的编码方式，从而在提高性能的同时增加了在将来引入新的选项的灵活性。它还包括了对业务流进行标记的能力。

包含许多特殊建议的列表的第五项指出，IPng将基于SIPP的128位地址。该RFC中的其他部分为Internet研究小组解决IPv4中的问题提供了非常好的历史资料，同时也提供了对于三个竞争者：TUBA、CATNIP和SIPP的详细分析。该RFC检查了每个提案，并讨论了它们如何满足(或不能满足)需要，同时也提供了这些提案经评审后的结果。

所有三种提案都在某种程度上获得了赞扬，并都可以在最终的提案中找到各自的影子。例如，SIPP中并不包括一个强壮的过渡方案或可以被全盘接受的自动配置的机制，因此在该标准中引入了TUBA在这些方面的方案。同时SIPP中所有值得自豪的方案中也有些未被采纳：如地址扩展的概念被认为是实验性过强并将为IPng的工作引入风险，同时其64位地址被128位地址取代以适应任何没有预见到的情况。

RFC 1752描述的建议中包括了多种与IPng及相关协议的实际设计相关的进一步的任务。SIPP及其他协议可以被认为是只是一个起点，尤其是当IPng强壮到可以长年为Internet服务的时候就更是如此。

4.4 IPv6，第一回合

最早的描述IPv6及其支持的协议的RFC标准(RFC 1883~1887)于1996年早期发表，其中有一些可参见本书的附录B中。但是，它们当时并没有全部完成且其后加入了多种附件和少许修改。附录A中包括与IPng和IPv6相关的RFC的列表。下一章将描述IPv6的基础，第6章则提供了基于这些和后来的协议标准的更多的细节。

4.5 IPv6，第二回合

到1998年夏末为止，新的IPv6 RFC获得了发表的批准。其中尤其值得注意的是，RFC 2373(IPv6的寻址体系结构)替换了RFC 1883；RFC 2374(一种IPv6可集聚全球单播地址格式)替换了RFC 2073。其他允许发表的新的RFC描述了ICMPv6、IPv6中的邻居发现和无状态自动配置。

第二部分 IPv6细节

第5章 IPv6的成型

本章介绍了IPv4的更新，描述了新的协议头中各字段及IPv6的地址空间，着重介绍了IPv6中包含的变化和新特性。IPv4拥有两个“帮助”协议：Internet控制报文协议(ICMP)和Internet组管理协议(IGMP)。主机和路由器使用这些协议来报告IP层差错及执行其他功能、诊断等。IPv6中使用的是对ICMP进行升级后的ICMPv6协议，ICMPv6中最初包含了IGMP的功能，但现在看来这些功能可能由IGMPv2来完成。

本章第一小节描述了IPv6协议的基本框架并介绍了RFC 1883(IPv6技术规范)和其他后续标准(到1998年9月还没有分配RFC号码)中定义的IPv6头字段、选项和扩展。本章第二小节概述了RFC 1885(用于IPv6的Internet控制报文协议(ICMPv6)的技术规范)中定义的ICMPv6。IGMPv2在RFC 2236(Internet组管理协议第2版)中定义，并且与IPv4和IPv6均有关联。

IPv6的地址方案将在第6章中介绍，第7章对IPv6的选项和扩展头有更详细的介绍。第8章将探讨IPv6的选路。第9章将进一步讨论IPv6中的安全性和身份验证问题，第10章将介绍升级到IPv6对IP的上层和下层协议造成的影响。

5.1 IPv6

对IPv4的升级最早在两个RFC中进行了定义。RFC 1883中描述的是协议本身，而RFC 1884介绍的是IPv6的地址结构。现在RFC 1884已经被RFC 2373所替代，1998年夏天IETF批准了一个草案来替换RFC 1883。从32位地址到128位地址的变化代表了一个重大的转变，但如何制定和分配IPv6地址直到1998年秋天也没有定论。第6章将对于IPv6的地址有更详细的介绍。本节只介绍真正的IPv6协议中最重要的改变而不讨论地址细节。

5.1.1 变化概述

IPv6中的变化体现在以下五个重要方面：

- 扩展地址。
- 简化头格式。
- 增强对于扩展和选项的支持。
- 流标记。
- 身份验证和保密。

对于IP的这些改变对IAB于1991年制定的IPv6发展方向中的绝大部分都有所改进。IPv6的扩展地址意味着IP可以继续增长而无需考虑资源的匮乏，该地址结构对于提高路由效率有所帮助；对于包头的简化减少了路由器上所需的处理过程，从而提高了选路的效率；同时，改进对头扩展和选项的支持意味着可以在几乎不影响普通数据包和特殊包选路的前提下适应更

多的特殊需求；流标记办法为更加高效地处理包流提供了一种机制，这种办法对于实时应用尤其有用；身份验证和保密方面的改进使得 IPv6 更加适用于那些要求对敏感信息和资源特别对待的商业应用。

1. 扩展地址

IPv6 的地址结构中除了把 32 位地址空间扩展到了 128 位外，还对 IP 主机可能获得的不同类型地址作了一些调整。就像在第 6 章中将要详细介绍的一样，IPv6 中取消了广播地址而代之以任意点播地址。IPv4 中用于指定一个网络接口的单播地址和用于指定由一个或多个主机侦听的组播地址基本不变。

2. 简化的包头

IPv6 中包括总长为 40 字节的 8 个字段(其中两个是源地址和目的地址)。它与 IPv4 包头的不同在于，IPv4 中包含至少 12 个不同字段，且长度在没有选项时为 20 字节，但在包含选项时可达 60 字节。IPv6 使用了固定格式的包头并减少了需要检查和处理的字段的数量，这将使得选路的效率更高。

包头的简化使得 IP 的某些工作方式发生了变化。一方面，所有包头长度统一，因此不再需要包头长度字段。此外，通过修改包分段的规则可以在包头中去掉一些字段。IPv6 中的分段只能由源节点进行：该包所经过的中间路由器不能再进行任何分段。最后，去掉 IP 头校验和不会影响可靠性，这主要是因为头校验和将由更高层协议 (UDP 和 TCP) 负责。

3. 对扩展和选项支持的改进

在 IPv4 中可以在 IP 头的尾部加入选项，与此不同，IPv6 中把选项加在单独的扩展头中。通过这种方法，选项头只有在必要的时候才需要检查和处理。下面和第 7 章将对此有更多的讨论。

为便于说明，考虑以下两种不同类型的扩展部分：分段头和选路头。IPv6 中的分段只发生在源节点上，因此需要考虑分段扩展头的节点只有源节点和目的节点。源节点负责分段并创建扩展头，该扩展头将放在 IPv6 头和下一个高层协议头之间。目的节点接收该包并使用扩展头进行重装。所有中间节点都可以安全地忽略该分段扩展头，这样就提高了包选路的效率。

另一种选择方案中，逐跳 (hop-by-hop) 选项扩展头要求包的路径上的每一个节点都处理该头字段。这种情况下，每个路由器必须在处理 IPv6 包头的同时也处理逐跳选项。第一个逐跳选项被定义用于超长 IP 包(巨型净荷)。包含巨型净荷的包需要受到特别对待，因为并不是所有链路都有能力处理那样长的传输单元，且路由器希望尽量避免把它们发送到不能处理的网络上。因此，这就需要在包经过的每个节点上都对选项进行检查。

4. 流

在 IPv4 中，对所有包大致同等对待，这意味着每个包都是由中间路由器按照自己的方式来处理的。路由器并不跟踪任意两台主机间发送的包，因此不能“记住”如何对将来的包进行处理。IPv6 实现了流概念，其定义如 RFC 1883 中所述：

流指的是从一个特定源发向一个特定(单播或者是组播)目的地的包序列，源点希望中间路由器对这些包进行特殊处理。

路由器需要对流进行跟踪并保持一定的信息，这些信息在流中的每个包中都是不变的。这种方法使路由器可以对流中的包进行高效处理。对流中的包的处理可以与其他包不同，但无论如何，对于它们的处理更快，因为路由器无需对每个包头重新处理。下一节中将对流和

流标记有更详细的讨论。

5. 身份验证和保密

RFC 1825(IP的安全性体系结构)描述了IP的安全性体系结构,包括IPv4和IPv6。它发表于1995年8月,目前正在进行修改和更新。1998年3月发表了一个更新版 Internet草案。IP安全性的基本结构仍然很坚固,且已经进行了一些显著的改变和补充。这个体系结构以及它在IPv6中如何实现,都将在第9章介绍。

IPv6使用了两种安全性扩展:IP身份验证头(AH)首先由RFC 1826(IP身份验证头)描述,而IP封装安全性净荷(ESP)首先在RFC 1827(IP封装安全性净荷(ESP))中描述。

报文摘要功能通过对包的安全可靠性的检查和计算来提供身份验证功能。发送方计算报文摘要并把结果插入到身份验证头中,接收方根据收到的报文摘要重新进行计算,并把计算结果与AH头中的数值进行比较。如果两个数值相等,接收方可以确认数据在传输过程中没有被改变;如果不相等,接受方可以推测出数据或者是在传输过程中遭到了破坏,或者是被某些人进行了故意的修改。

封装安全性提供机制,可以用来加密IP包的净荷,或者在加密整个IP包后以隧道方式在Internet上传输。其中的区别在于,如果只对包的净荷进行加密的话,包中的其他部分(包头)将公开传输。这意味着破译者可以由此确定发送主机和接收主机以及其他与该包相关的信息。使用ESP对IP进行隧道传输意味着对整个IP包进行加密,并由作为安全性网关操作的系统将其封装在另一IP包中。通过这种方法,被加密的IP包中的所有细节均被隐藏起来。这种技术是创建虚拟专用网(VPN)的基础,它允许各机构使用Internet作为其专用骨干网络来共享敏感信息。

5.1.2 包头结构

在IPv4中,所有包头以32位为单位,即基本的长度单位是4个字节。在IPv6中,包头以64位为单位,且包头的总长度是40字节。IPv6协议为对其包头定义了以下字段:

- 版本。长度为4位,对于IPv6,该字段必须为6。
- 类别。长度为8位,指明为该包提供了某种“区分服务”。RFC 1883中最初定义该字段只有4位,并命名为“优先级字段”,后来该字段的名称改为“类别”,在最新的IPv6 Internet草案中,称之为“业务流类别”。该字段的定义独立于IPv6,目前尚未在任何RFC中定义。该字段的默认值是全0。
- 流标签。长度为20位,用于标识属于同一业务流的包。一个节点可以同时作为多个业务流的发送源。流标签和源节点地址唯一标识了一个业务流。在RFC 1883中这个字段最初被设计为24位,但当类别字段的长度增加到8位后,流标签字段被迫减小长度来作补偿。
- 净荷长度。长度为16位,其中包括包净荷的字节长度,即IPv6头后的包中包含的字节数。这意味着在计算净荷长度时包含了IPv6扩展头的长度。
- 下一个头。这个字段指出了IPv6头后所跟的头字段中的协议类型。与IPv6协议字段类似,下一个头字段可以用来指出高层是TCP还是UDP,但它也可以用来指明IPv6扩展头的存在。
- 跳极限。长度为8位。每当一个节点对包进行一次转发之后,这个字段就会被减1。如果

该字段达到 0，这个包就将被丢弃。 IPv4中有一个具有类似功能的生存期字段，但与 IPv4不同，人们不愿意在 IPv6中由协议定义一个关于包生存时间的上限。这意味着对过期包进行超时判断的功能可以由高层协议完成。

- 源地址。长度为 128 位，指出了 IPv6包的发送方地址。
- 目的地址。长度为 128 位，指出了 IPv6包的接收方地址。这个地址可以是一个单播、组播或任意点播地址。如果使用了选路扩展头（其中定义了一个包必须经过的特殊路由），其目的地址可以是其中某一个中间节点的地址而不必是最终地址。

图5-1中显示了IPv6头的格式。下一节中提供了IPv6头与IPv4头字段间更加详细的比较。

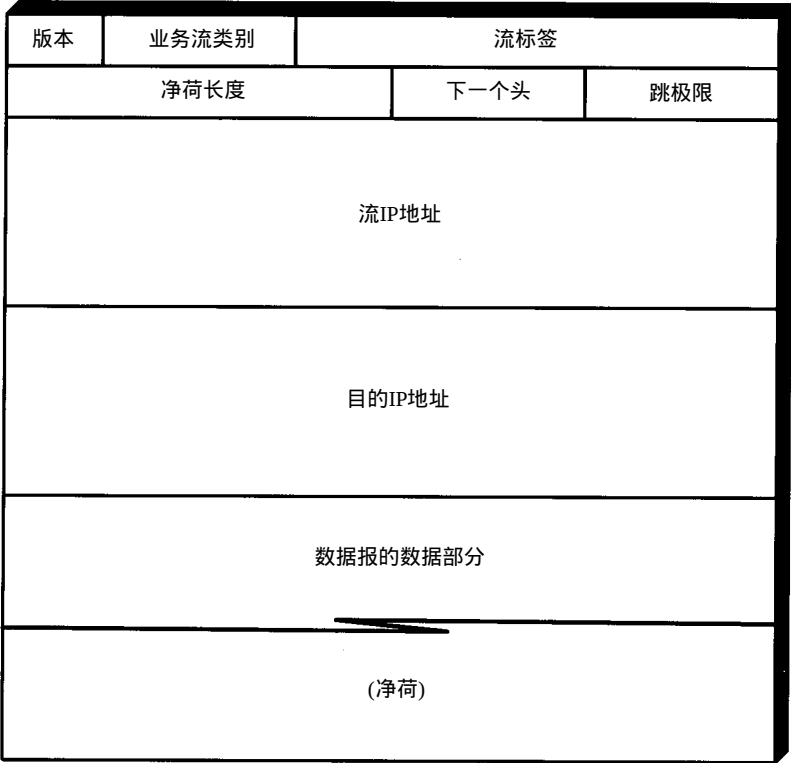


图5-1 Ipv6头比IPv4头(参见图2-3)简单得多

5.1.3 IPv4与IPv6的比较

先回顾一下图 2-3中定义的IPv4头。尽管这些头字段中有一些与 IPv6头类似，但其中真正完全保持不变的只有第一个字段，即版本字段，因为在同一条线路上传输时，必须保证 IPv4和IPv6的兼容性。下一个字段，即包头长度，则与 IPv6无关，因为 IPv6头是固定长度，IPv4中需要这个字段是因为它的包头可能在 20字节到40字节间变化。

服务类型字段与 IPv6的流类别字段相似，但 TOS的位置比该字段要落后一些，而且在具体实现中也没有广泛应用。下一个字段是数据报长度，后来发展成了 IPv6中的净荷长度。IPv6的净荷长度中包含了扩展头，而 IPv4数据报长度字段中则指明包含包头在内的整个数据报的长度。这样一来，在 IPv4中，路由器可以通过将数据报长度减去包头长度来计算包的净荷长度，而在IPv6中则无须这种计算。

后面的三个字段是数据报 ID、分段标志和分段偏移值，它们都用于 IPv4 数据报的分段。由于 IPv6 中由源结点取代中间路由器来进行分段 (后面将有更多关于分段的内容)，这些字段在 IPv6 中变得不重要，并被 IPv6 从包头中去掉了。

而生存期字段，正如上面所述，变成了跳极限字段。生存期字段最初表示的是一个包穿越 Internet 时以秒为单位的存在时间的上限。如果生存期计数值变为 0，该包将被丢弃。其原因是包可能会存在于循环路由中，如果没有方法让它消失，它可能会一直选路 (或者直到网络崩溃为止)。在最初的规范中要求路由器根据转发包的时间与收到包的时间的差值 (以秒为单位) 来减小生存期的值。在实际情况中，大部分路由器都设计为每次对该值减 1，而不是计算路由器上真正的处理时间。

协议字段，如前所述，指出在 IPv4 包中封装的高层协议类型。各协议对应的数值在最新版本 RFC (现在是 RFC 1700) 中可以查到。这个字段后来发展成为 IPv6 中的下一个头字段，其中定义了下一个头是一个扩展头字段还是另一层的协议头。

由于如 TCP 和 UDP 等高层协议均计算头的校验和，IPv4 头校验显得有些多余，因此这个字段在 IPv6 中已消失。对于那些真的需要对内容进行身份验证的应用，IPv6 中提供了身份验证头。

IPv6 中仍然保留了 32 位的 IPv4 源地址和目的地址，但将它们扩展为 128 位。而 IP 选项字段则已经彻底消失，取而代之的是 IPv6 扩展头。

5.1.4 流标签

IPv4 通常被描述为无连接协议。就像任何一个包交换网络一样，IPv4 设计为让每个包找到自己的路径以到达其目的地。每个包都分别处理，而结果是两个从相同数据源发往相同目的地的包可以采用完全不同的路由来穿越整个网络。这对于适应网络突发事件来说是个好办法，因为突发事件意味着任何一条路由都可能在任何时间出现故障，但只要两主机间存在某些路由则可以进行数据的交互。

但是，这种方法的效率可能不太高，尤其是当包并不是孤立的，且实际上是两个通信系统间的业务流的一部分时。进一步考虑一个包流从一台主机发往另一主机时在它所经过的路径上将发生的事情：每个中间路由器对每个包的处理将导致在链路上轻微地增加延时。对于类似文件传输或终端仿真之类的大部分传统 Internet 应用，延时只会带来一点不方便而已，但对于一些提供互操作的音频和视频应用而言，即使只是增加一点点延时也会显著降低服务质量。

对每个 IPv4 包均进行单独处理带来的另一个问题在于难以把特定的业务流指定到较低代价的链路上。例如，电子邮件的传输优先级不高，并且不是实时应用，但 IPv4 网络管理员却没有简单的办法来标识这些包，把它们传输到较低开销的 Internet 链路，并为实时应用保留较高开销的链路。

IPv6 中定义的流的概念将有助于解决类似问题。IPv6 头字段中的流标签把单个包作为一个系列源地址和目的地址相同的包流的一部分。同一个流中的所有包具有相同的流标签。

5.1.5 业务流类别

最早有关 IPv6 的 RFC (1883) 中定义了 4 位优先级字段，这意味着每个包可能具备 16 个优先级中的一个。但是，经过多次讨论后这个字段的名称改为“类别”，且长度也扩大到了 1 字节。

在最新的关于RFC 1883的Internet修订草案中，名字又被改为“业务流类别”。

IPv6类别字段的数值及如何正确使用还有待定义。使用IPv4服务类型字段和使用IPv6类别的实验最终必将为此带来有用的结果。使用业务流类别的目的在于允许发送业务流的源节点和转发业务流的路由器在包上加上标记，并进行除默认处理方法之外的不同处理。一般来说，在所选择的链路上，可以根据开销、带宽、延时或其他特性而对包进行特殊的处理。

虽然在IPv6的实现中很可能需要并建议高层协议为它们的数据指定一个特定的业务流等级，但这些实现中可能也允许中间路由器根据实际情况修改这个值。

5.1.6 分段

IPv6的分段只能由源节点和目的节点进行，这样就简化了包头并减少了用于选路的开销。逐跳分段被认为是一种有害的方法。首先，它在端到端的分段中将产生更多的分段。此外在传输中，一个分段的丢失将导致所有分段重传。IPv6的确可以通过其扩展头来支持分段，但是如下所述，了解IPv4分段如何工作将有助于了解IPv6中为什么要进行改变。

在IPv4中，当一个没有分段的包由于太长而无法沿着发送源到目的地的网络链路进行传输时，就需要进行包的分段。举例来说，一个源节点可以创建一个长度为1500字节的包，并把它向Internet上的某个远端目的地发送。这个包通过源节点的本地以太网到达该节点的默认路由器。然后路由器通过其链路把数据发到Internet上，这条链路可能是到一个ISP的点到点连接。在Internet中的某处或离目的节点较近的某处，可能有条网络链路无法处理这样一大块的数据。在这种情况下，使用该网络链路的路由器将不得不把1500字节的数据报分割成许多不超过下一个网络的最大传输单元(MTU)的分段。因此，如果假设下一个链路可以处理的包长度不能超过1280字节的话，路由器将把最初的一个包分割为两个。第一个包的长度为1260字节，留下的20字节用于IPv4头。第二段的长度就是剩余数据的长度，240字节，另外再用20字节作为另一个IPv4头。

IPv4中的分段由沿途的中间路由器根据需要进行。进行分段的路由器根据需要修改包头并在其中包含进最初的数据报标识，同时还须正确地设置分段标志和分段偏移值。当目的节点收到由此产生的分段包之后，该系统必须根据每个分段包的IPv4头后的分段数据重组最初的包。

在使用了分段之后，不论中间的网络是什么类型，不同类型网络上的节点都可以互操作，源节点无需了解任何有关目的节点网络的信息，同时也无需了解它们之间的网络信息。这一一直被认为是一个不错的特性，由于不需要节点或路由器存储信息或记录整个Internet的结构，从而Internet可以获得很好的扩展性。但另一方面，它也为路由器带来了性能方面的问题，对IP包进行分段消耗了沿途路由器和目的地的处理能力和时间。了解IP数据报标识、计算分段偏移值、真正把数据分段以及在目的地进行重装都会带来额外的开销。

问题在于对于任何一个指定的路由器，虽然源节点能够了解链路的MTU是多大，但却没有办法事先知道整个路径的MTU。路径MTU是源节点和目的节点之间在不分段时可以沿着该路由穿越任何网络的最大包长。

然而，目前有两种方法可以减少或消除对于分段的需求。第一种方法可用在IPv4中，它使用一种叫做“路径MTU发现”的方法。通过这种方法，路由器可以向目的地发送一个包来报告该路由器上链路的MTU值。如果包到达了一条必须对其进行分段的链路，负责分段的路

由器将使用ICMP回送一个报文来指出分段路由器上链路的 MTU 值。这种过程可以重复进行直到路由器确定路径 MTU 为止。(后面将有对 ICMP 的进一步讨论。)

另一种减少分段需求的方法是要求所有支持 IP 的链路必须能够处理一些合理的最小长度的包。换句话说,如果一个链路的 MTU 超过 20 字节,那么所有的节点都必须准备产生可观数量的分段包。另一方面,如果能够提出所有网络链路都可以适应的某个合理的长度,并把它设置为允许包长度的绝对最小值,那么就可以消灭分段。

IPv6 中实际上同时使用了上面两种方法。在最初的 RFC 中,IPv6 规定每个链路支持的 MTU 最小为 576 字节。那么这些包的净荷长度将是 536 字节,另外 40 字节用于 IPv6 头。由于 RFC 1883 发表于 1995 年,后来产生了很多关于更大的 MTU 的争论。在 Huitema 提出的报告(参见《IPv6:新的 IP》第 2 版,Prentice-Hall)中,建议值为 1997,Steve Deering 则正在促使将 MTU 值改为 1500 字节。在最新的于 1997 年 11 月发表的 Internet 草案中,MTU 值被设为 1280 字节。很明显,关注的焦点在于:倡导较短 MTU 的人希望那些不能支持较长 MTU 的网络不会被完全丢弃,而倡导较长 MTU 的人不希望为照顾小部分接近于废弃的网络而使得整个 Internet 的性能下降。

为了对较短的 MTU 进行一些弥补,IPv6 标准中强烈推荐所有 IPv6 节点都支持路径 MTU 发现。路径 MTU 发现最早出现在 RFC 1191 中,其中使用了分段标志中的“不能分段”来要求中间路由器在发现包太长时返回一个 ICMP 出错报文。

路径 MTU 发现的 IPv6 版本在 RFC 1981(IPv6 的路径 MTU 发现)中描述。这是对原有的 RFC 1191 的升级,但其中加入了一些改变使之可以工作在 IPv6 中。其中最重要的是,由于 IPv6 头中不支持分段,因此也就没有“不能分段”位。正在执行路径 MTU 发现的节点只是简单地自己的网络链路上向目的地发送允许的最长包。如果一条中间链路无法处理该长度的包,尝试转发路径 MTU 发现包的路由器将向源节点回送一个 ICMPv6 出错报文。然后源节点将发送另一个较小的包。这个过程将一直重复,直到不再收到 ICMPv6 出错报文为止,然后源节点就可以使用最新的 MTU 作为路径 MTU。

这里需要注意,有一些实例并没有实现路径 MTU 发现。例如,使用最小 IPv6 实现来进行远程网络启动的终端只是简单地使用 576 字节的路径 MTU。从源节点到目的节点的 IPv6 分段,作为一个扩展头来实现,将在下一节中讨论。

5.1.7 扩展头

IPv4 选项的问题在于改变了 IP 头的大小,因此更像一个“特例”,即需要特别的处理。路由器必须优化其性能,这意味着将为最普遍的包进行最佳性能的优化。这使得 IPv4 选项引发一个路由器把包含该选项的包搁置一边,等到有时间的时候再进行处理。

IPv6 中实现的扩展头可以消灭或至少大量减少选项带来的对性能的冲击。通过把选项从 IP 头中搬到净荷中,路由器可以像转发无选项包一样来转发包含选项的包。除了规定必须由每个转发路由器进行处理的逐跳选项之外,IPv6 包中的选项对于中间路由器而言是不可见的。

可用的选项

除了减少 IPv6 包转发时选项的影响外,IPv6 规范使得对于新的扩展和选项的定义变得更加简单。在需要的时候可能还会定义其他的选项和扩展。本节仅列出已定义的扩展,而对于扩展头和选项的使用在第 7 章中将有更详细的讨论,安全性头将在第 9 章讨论。RFC 1883 中为

IPv6 定义了如下选项扩展：

- 逐跳选项头。此扩展头必须紧随在 IPv6 头之后。它包含包所经路径上的每个节点都必须检查的选项数据。由于它需要每个中间路由器进行处理，逐跳选项只有在绝对必要的时候才会出现。到目前为止，已经定义了两个选项：巨型净荷选项和路由器提示选项。巨型净荷选项指明包的净荷长度超过 IPv6 的 16 位净荷长度字段。只要包的净荷超过 65 535 字节(其中包括逐跳选项头)，就必须包含该选项。如果节点不能转发该包，则必须回送一个 ICMPv6 出错报文。路由器提示选项用来通知路由器，IPv6 数据报中的信息希望能够得到中间路由器的查看和处理，即使这个包是发给其他某个节点的（例如，包含带宽预留协议信息的控制数据报）。
- 选路头。此扩展头指明包在到达目的地途中将经过哪些节点。它包含包沿途经过的各节点的地址列表。IPv6 头的最初目的地址是路由头的一系列地址中的第一个地址，而不是包的最终目的地址。此地址对应的节点接收到该包之后，对 IPv6 头和选路头进行处理，并把包发送到选路头列表中的第二个地址。如此继续，直到包到达其最终目的地。
- 分段头。此扩展头包含一个分段偏移值、一个“更多段”标志和一个标识符字段。用于源节点对长度超出源端和目的端路径 MTU 的包进行分段。
- 目的地选项头。此扩展头代替了 IPv4 选项字段。目前，唯一定义的目的地址选项是在需要时把选项填充为 64 位的整数倍。此扩展头可以用来携带由目的地节点检查的信息。
- 身份验证头(AH)。此扩展头提供了一种机制，对 IPv6 头、扩展头和净荷的某些部分进行加密的校验和的计算。
- 封装安全性净荷(ESP)头。这是最后一个扩展头，不进行加密。它指明剩余的净荷已经加密，并为已获得授权的目的地节点提供足够的解密信息。

5.2 ICMPv6

IP 节点需要一个特殊的协议来交换报文以了解与 IP 相关的情况。ICMP 正好适用于这种需求。在 IPv4 升级到 IPv6 的过程中，ICMP 也经历了一定的修改。ICMPv6 在 RFC 1885 中定义。ICMP 报文可以用来报告错误和信息状态，以及类似于包的 Internet 探测(Ping)和跟踪路由的功能。

IGMP 一开始就包含在 ICMPv6 规范中，并且在 1997 年 11 月发表的 RFC 2236 中得到更新，1998 年初秋，IGMP 第 3 版也开始了讨论。IGMP 可以用来支持组播传输，它为主机提供了向本地路由器报告其属于某个组播组的方法。

ICMPv6 报文

ICMP 报文的产生来源于一些错误情况。例如，如果一个路由器由于某些原因不能处理一个 IP 包，它就可能会产生某种类型的 ICMP 报文，并直接回送到包的源节点，然后源节点将采取一些办法来纠正所报告的错误状态。例如，如果路由器无法处理一个 IP 包的原因是由于包太长而无法将其发送到网络链路上，则路由器将产生一个 ICMP 错误报文来指出包太长，源节点在收到该报文后可以用它来确定一个更加合适的包长度，并通过一系列新的 IP 包来重新发送该数据。

RFC 1885 中定义了以下报文类型(没有包括该文档中定义的有关组的报文)：

- 目的地不可达。
- 包太长。
- 超时。
- 参数问题。
- 回声请求。
- 回声应答。

下面将详细介绍这些报文。

1. 目的地不可达

这个报文由路由器或源主机在由于除业务流拥塞之外的原因而无法转发一个包的时候产生。这种错误报文有五个代码，包括：

- 0：没有到达目的地的路由。这个报文在路由器没有定义 IP包的目的地路由时产生，路由器将采用默认路由来发送无法利用路由器的路由表进行转发的包。
- 1：与目的地的通信被管理员禁止。当被禁止的某类业务流欲到达防火墙内部的一个主机时，包过滤防火墙将产生该报文。
- 2：不是邻居。当使用 IPv6选路扩展头并严格限定路由时，将使用这个代码。当列表中的下一个目的地与当前正执行转发的节点不能共享一个网络链路时，将会产生该报文。
- 3：地址不可达。这个代码指出在把高层地址解析到链路层（网络）地址时遇到了一些问题，或者在目的地网络的链路层上去往其目的地时遇到了问题。
- 4：端口不可达。这种情况发生在高层协议（如DP）没有侦听包目的端口的业务量，且传输层协议又没有其他办法把这个问题通知源节点时。

2. 包太长

当接收某包的路由器由于包长度大于将要转发到的链路的 MTU，而无法对其进行转发时，将会产生包太长报文。该 ICMPv6错误报文中有一个字段指出导致该问题的链路的 MTU值。在路径MTU发现过程中这是一个有用的错误报文。

3. 超时

当路由器收到一个跳极限为 1的包时，它必须在转发该包之前减小这个数值。如果在路由器减小该数值后，跳极限字段的值变为 0(或者是路由器收到一个跳限制字段为 0的包)，那么路由器必须丢弃该包，并向源节点发送 ICMPv6超时报文。源节点在收到该报文后，可以认为最初的跳限制设置得太小(包的真实路由比源节点想象的要长)，也可以认为有一个选路循环导致包无法交付。

在“跟踪路由”功能中这个报文非常有用。这个功能使得一个节点可以标识一个包在从源节点到目的节点的路径上的所有路由器。它的工作方式如下：首先，一个去往目的地的包的跳极限被设置为 1。它所到达的第一个路由器将跳减少极限，并回送一个超时报文，这样一来源节点就标识了路径上的第一个路由器。然后如果该包必须经过第二个路由器的话，源节点会再发送一个跳极限为 2的包，该路由器将把跳极限减小到 0，并产生另一个超时报文。这将持续到包最终到达其目的地为止。同时源节点也获得了从每个中间路由器发来的超时报文。

4. 参数问题

当IPv6头或扩展头中的某些部分有问题时，路由器由于无法处理该包而会将其丢弃。路由器的实现中应该可以产生一个 ICMP参数错误报文来指出问题的类型（如错误的头字段、无

法识别的下一个头类型或无法识别的 IPv6 选项), 并通过一个指针值指出在第几个字节遇到这种错误情况。

5. ICMPv6 回声功能

ICMPv6 中包含了一个与错误情况无关的功能。所有 IPv6 节点都需要支持两种报文: 回声请求和回声应答。回声请求报文可以向任何一个正确的 IPv6 地址发送, 并在其中包含一个回声请求标识符、一个顺序号和一些数据。尽管二者都是可选项, 但回声请求标识符和顺序号可以用来区分对应不同请求的响应。回声请求的数据也是一个选项, 并可用于诊断。

当一个 IPv6 节点收到一个回声请求报文后, 它必须回送一个回声应答报文。在应答中包含相同的请求标识符、顺序号和最初的请求报文中携带的数据。

ICMP 回声请求/应答报文对是 ping 功能的基础。ping 是一个重要的诊断功能, 因为它提供了一种方法来决定一个特定的主机是否与其他一些主机连接在相同的网络上。

第6章 IPv6寻址

本章在介绍IPv6寻址之前，首先介绍一些与使用IP寻址来标识和定位IP网络上的节点相关的问题。多年以来，IP地址被认为是可以在IP网络上最终唯一并持久的节点标识符。近年中，尤其是随着下一代IP技术的发展，对于IP地址的这种观点正在改变。如果我们仍像过去20年中所使用的方法来分配网络和节点地址，那将是一种不必要和低效的办法。

本章在介绍了RFC 2373(IPv6寻址体系结构)中描述的IP寻址体系结构之后，将首先介绍一些与IP寻址相关的议题。然后将介绍几种可能的地址分配方法。本章将IPv6寻址分成了以下几个部分：128位地址的结构和命名及IPv6地址的不同类型(单播、组播和泛播)。

IPv6的设计者们可以只是简单地在IPv4寻址体系结构中扩大地址空间。但是这样一来将使我们丧失一个改进IP的巨大机会。对于整个寻址体系结构的修改所带来的巨大机会，不仅体现在提高地址分配的效率上，同时也体现在提高IP选路性能上。本章将介绍这些改进，第8章对于IPv6选路议题将有更加详细的介绍。而地址分配、移动网络技术和自动配置将在第11章中有详细讲解。

RFC 2373于1998年7月发表，并废弃了最早于1995年12月发表的RFC 1884(IPv6寻址体系结构)。其中大部分变化源自最初的RFC发布后的两年半中被认为是必需需要进行澄清、更正和修改之处。

6.1 地址

IPv4与IPv6地址之间最明显的差别在于长度：IPv4地址长度为32位，而IPv6地址长度为128位。RFC 2373中不仅解释了这些地址的表现方式，同时还介绍了不同的地址类型及其结构。IPv4地址可以被分为2至3个不同部分(网络标识符、节点标识符，有时还有子网标识符)，IPv6地址中拥有更大的地址空间，可以支持更多的字段。

IPv6地址有三类、单播、组播和泛播地址。下一节将对此作更详细的介绍。单播和组播地址与IPv4的地址非常类似；但IPv6中不再支持IPv4中的广播地址，而增加了一个泛播地址。本节介绍的是IPv6的寻址模型、地址类型、地址表达方式以及地址中的特例。

6.1.1 地址表达方式

IPv4地址一般以4部分间点分的方法来表示，即4个数字用点分隔。例如，下面是一些合法的IPv4地址，都用十进制整数表示：

10.5.3.1

127.0.0.1

201.199.244.101

IPv4地址也时常以一组4个2位的十六进制整数或4个8位的二进制整数表示，但后一种情况较少见。

IPv6地址长度4倍于IPv4地址，表达起来的复杂程度也是IPv4地址的4倍。IPv6地址的基本

表达方式是 X:X:X:X:X:X:X:X，其中 X 是一个 4 位十六进制整数 (16 位)。每一个数字包含 4 位，每个整数包含 4 个数字，每个地址包括 8 个整数，共计 128 位 ($4 \times 4 \times 8 = 128$)。例如，下面是一些合法的 IPv6 地址：

```
CD CD:910A:2222:5498:8475:1111:3900:2020
```

```
1030:0:0:0:C9B4:FF12:48AA:1A2B
```

```
2000:0:0:0:0:0:0:1
```

请注意这些整数是十六进制整数，其中 A 到 F 表示的是 10 到 15。地址中的每个整数都必须表示出来，但起始的 0 可以不必表示。

这是一种比较标准的 IPv6 地址表达方式，此外还有另外两种更加清楚和易于使用的方式。

某些 IPv6 地址中可能包含一长串的 0 (就像上面的第二和第三个例子一样)。当出现这种情况时，标准中允许用“空隙”来表示这一长串的 0。换句话说，地址

```
2000:0:0:0:0:0:0:1
```

可以被表示为：

```
2000::1
```

这两个冒号表示该地址可以扩展到一个完整的 128 位地址。在这种方法中，只有当 16 位组全部为 0 时才会被两个冒号取代，且两个冒号在地址中只能出现一次。

在 IPv4 和 IPv6 的混合环境中可能有第三种方法。IPv6 地址中的最低 32 位可以用于表示 IPv4 地址，该地址可以按照一种混合方式表达，即 X:X:X:X:X:d.d.d.d，其中 X 表示一个 16 位整数，而 d 表示一个 8 位十进制整数。例如，地址

```
0:0:0:0:0:0:10.0.0.1
```

就是一个合法的 IPv4 地址。把两种可能的表达方式组合在一起，该地址也可以表示为：

```
::10.0.0.1
```

由于 IPv6 地址被分成两个部分——子网前缀和接口标识符，因此人们期待一个 IP 节点地址可以按照类似 CIDR 地址的方式被表示为一个携带额外数值的地址，其中指出了地址中有多少位是掩码。即，IPv6 节点地址中指出了前缀长度，该长度与 IPv6 地址间以斜杠区分，例如：

```
1030:0:0:0:C9B4:FF12:48AA:1A2B/60
```

这个地址中用于选路的前缀长度为 60 位。

6.1.2 寻址模型

IPv6 寻址模型与 IPv4 很相似。每个单播地址标识一个单独的网络接口。IP 地址被指定给网络接口而不是节点，因此一个拥有多个网络接口的节点可以具备多个 IPv6 地址，其中任何一个 IPv6 地址都可以代表该节点。尽管一个网络接口能与多个单播地址相关联，但一个单播地址只能与一个网络接口相关联。每个网络接口必须至少具备一个单播地址。

这里有一个非常重要的声明和一个非常重要的例外。这个声明与点到点链路的使用有关。在 IPv4 中，所有的网络接口，其中包括连接一个节点与路由器的点到点链路（用许多拨号 Internet 连接中），都需要一个专用的 IP 地址。随着许多机构开始使用点到点链路来连接其分支机构，每条链路均需要其自己的子网，这样一来消耗了许多地址空间。在 IPv6 中，如果点到点链路的任何一个端点都不需要从非邻居节点接受和发送数据的话，它们就可以不需要特殊的地址。即，如果两个节点主要是传递业务流，则它们并不需要具备 IPv6 地址。

为每个网络接口分配一个全球唯一的单播地址的要求阻碍了 IPv4地址的扩展。一个提供通用服务的服务器在高需求量的情况下可能会崩溃。因此，IPv6地址模型中又提出了一个重要的例外：如果硬件有能力在多个网络接口上正确地共享其网络负载的话，那么多个网络接口可以共享一个 IPv6地址。这使得从服务器扩展至负载分担的服务器群成为可能，而不再需要在服务器的需求量上升时必须进行硬件升级。

下面将要讨论的组播和泛播地址也与网络接口有关。一个网络接口可以具备任意类型的多个地址。

6.1.3 地址空间

RFC 2373中包含了一个IPv6地址空间“图”，其中显示了地址空间是如何进行分配的，地址分配的不同类型，前缀(地址分配中前面的位值)和作为整个地址空间的一部分的地址分配的长度。图6-1显示了IPv6地址空间的分配。

分配	前缀(二进制)	占地址空间的百分率
保留	0000 0000	1/256
未分配	0000 0001	1/256
为NSAP分配保留	0000 001	1/128
为IPX分配保留	0000 010	1/128
未分配	0000 011	1/128
未分配	0000 1	1/32
未分配	0001	1/16
可集聚全球单播地址	001	1/8
未分配	010	1/8
未分配	011	1/8
未分配	100	1/8
未分配	101	1/8
未分配	110	1/8
未分配	1110	1/16
未分配	1111 0	1/32
未分配	1111 10	1/64
未分配	1111 110	1/128
未分配	1111 1110 0	1/512
链路本地单播地址	1111 1110 10	1/1024
站点本地单播地址	1111 1110 11	1/1024
组播地址	1111 1111	1/256

图6-1 RFC 2373定义的IPv6地址空间的分配

在IPv6地址分配中需要注意几点。首先，在 RFC 1884中，地址空间的四分之一被用于两类不同地址：八分之一是基于供应商的单播地址，而另八分之一是基于地理位置的单播地址。人们希望地址的分配可以根据网络服务供应商或者用户所在网络的物理位置进行。基于供应商的集聚，正如它最初的名字一样，要求网络从提供 Internet接入的供应商那里得到可集聚的 IP地址。但是，这种方法对于具有距离较远的分支机构的大型机构来说并不是一种完美的解

决办法，因为其中许多分支机构可能会使用不同的供应商。基于供应商的集聚将为这些大单位带来更多的IP地址管理问题。

Steve Deering提议把基于地理位置的地址分配方法作为 SIP(SIPP的前身，在第4章中有介绍)中的一种办法。这些地址与基于供应商的地址不同，以一种非常类似 IPv4的方法分配地址。这些地址与地理位置有关，且供应商将不得不保留额外的路由器来支持 IPv6地址空间中可集聚部分外的这些网络。

ISP实际上并不赞同这个解决方案，因为管理基于地理位置的寻址将大大增加复杂性（和花费）。另一方面，难以对基于供应商的地址进行配置和重配置也引起许多对基于供应商的分配方案的反对。如果没有广泛使用基于 IPv4自动配置方案(如DHCP)，那么所有机构的网络将会存在巨大的管理问题。尽管 IPv6对于自动配置功能有着更好的支持，但并没有将地理位置的分配方法最终融合进去。

注意，绝大部分的地址空间并没有分配，地址分配的第一部分被保留了下来。图 6-1中所列出的地址类型以及保留分配的一些地址将在下一节中讨论。

6.2 地址类型

如上所述，IP地址有三种类型：单播、组播和任意点播。广播地址已不再有效。RFC 2373中定义了三种IPv6地址类型：

- 单播：一个单接口的标识符。送往一个单播地址的包将被传送至该地址标识的接口上。
- 泛播：一组接口(一般属于不同节点)的标识符。送往一个泛播地址的包将被传送至该地址标识的接口之一(根据选路协议对于距离的计算方法选择“最近”的一个)。
- 组播：一组接口(一般属于不同节点)的标识符。送往一个组播地址的包将被传送至有该地址标识的所有接口上。

这三种地址类型将在下面进行更详细的论述。

6.2.1 广播路在何方

广播地址从一开始就为 IPv4网络带来了问题。广播被用来携带去向多个节点的信息或被那些不知信息来自何方的节点用来发出请求。但是，广播可能将为网络性能设置障碍。同一网络链路上的大量广播意味着该链路上的所有每个节点都必须处理所有广播，其中绝大部分节点最终都将忽略该广播，因为该信息与自己无关。把广播在子网之间进行转发将导致更多的问题，因为路由器上将充斥着这种业务流。

IPv6对此的解决办法是使用一个“所有节点”组播地址来替代那些必须使用广播的情况，同时，对那些原来使用了广播地址的场合，则使用一些更加有限的组播地址。通过这种方法，对于原来由广播携带的业务流感兴趣的节点可以加入一个组播地址，而其他对该信息不感兴趣的节点则可以忽略发往该地址的包。广播从来不能解决信息穿越 Internet的问题，如选路信息，而组播则提供了一个更加可行的方法。

6.2.2 单播

单播地址标识了一个单独的 IPv6接口。一个节点可以具有多个 IPv6网络接口。每个接口

必须具有一个与之相关的单播地址。单播地址可被认为包含了一段信息，这段信息被包含在128位字段中：该地址可以完整地定义一个特定的接口。此外，地址中数据可以被解释为多个小段的信息。但无论如何，当所有的信息被放在一起后，将构成标识一个节点接口的128位地址。

IPv6地址本身可以为节点提供关于其结构的或多或少的信息，这主要根据是由谁来观察这个地址以及观察什么。例如，节点可能只需简单地了解整个128位地址是一个全球唯一的标识符，而无须了解节点在网络中是否存在。另一方面，路由器可以通过该地址来决定，地址中的一部分标识了一个特定网络或子网上的一个唯一节点。

例如，一个IPv6单播地址可看成是一个两字段实体，其中一个字段用来标识网络，而另一个字段则用来标识该网络上节点的接口。在后面讨论特定的单播地址类型时还会看到，网络标识符可被划分为几部分，分别标识不同的网络部分。IPv6单播地址功能与IPv4地址一样受制于CIDR，即，在一个特定边界上将地址分为两部分。地址的高位部分包含选路用的前缀，而地址的低位部分包含网络接口标识符。

最简单的方法是把IPv6地址作为不加区分的一块128位的数据，而从格式化的观点来看，可把它分为两段，即接口标识符和子网前缀。RFC 2373中表示的格式见图6-2。接口标识符的长度取决于子网前缀的长度。两者的长度是可以变化的，这取决于谁对它进行解释。对于非常靠近寻址的节点接口（远离骨干网）的路由器可用相对较少的位数来标识接口。而离骨干网近的路由器，只需用少量地址位来指定子网前缀，这样，地址的大部分将用来标识接口标识符。下面要讨论的是可集聚的单播地址，它的结构更为复杂。

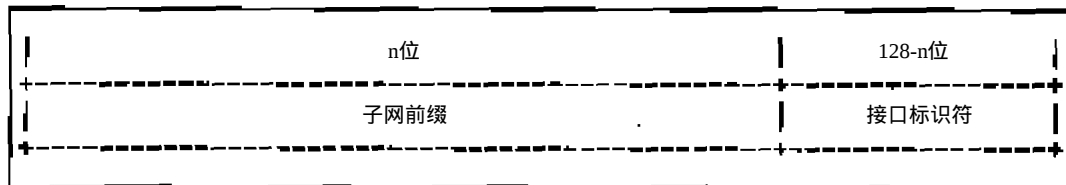


图6-2 RFC 2373中定义的IPv6单播地址的简单格式

IPv6单播地址包括下面几种类型：

- 可集聚全球地址。
- 未指定地址或全0地址。
- 回返地址。
- 嵌有IPv4地址的IPv6地址。
- 基于供应商和基于地理位置的供应商地址。
- OSI网络服务访问点(NSAP)地址。
- 网络互联包交换(IPX)地址。

6.2.3 单播地址格式

RFC 1884给出了几种通用的不同类型的IPv6地址。给NSAP和IPX分配的地址、基于OSI网络和NetWare地址都无缝地包含在IPv6体系结构中。分别占八分之一的地址空间的基于供应商和基于地理位置分配的地址组成了一批可分配的地址。链路本地和站点本地地址提供了10型网络地址转换的网络统一不变的版本。

然而，RFC 2373 改变和简化了 IPv6 的地址分配。其中之一是取消了基于地理位置的地址分配，基于供应商的单播地址改变成可集聚全球单播地址。从名字的改变上就可看出，对于基于供应商的地址，允许前面定义的集聚以及基于交换局的新型集聚。这也反映了一种更平衡的地址分类。NSAP 和 IPX 地址空间仍然保留着，且八分之一的地址分配给可集聚地址。另外，除了组播地址和某类保留地址外，IPv6 地址空间的其余部分都是未分配的地址，为将来的发展预留了足够的空间。

1. 接口标识符

在 IPv6 寻址体系结构中，任何 IPv6 单播地址都需要一个接口标识符。接口标识符非常像 48 位的介质访问控制 (MAC) 地址，MAC 地址由硬件编码在网络接口卡中，由厂商烧入网卡中，而且地址具有全球唯一性，不会有网卡具有相同的 MAC 地址。这些地址能用来唯一标识网络链路层上的接口。

IPv6 主机地址的接口标识符基于 IEEE EUI-64 格式。该格式基于已存在的 MAC 地址来创建 64 位接口标识符，这样的标识符在本地和全球范围是唯一的。RFC 2373 包括的附录解释了如何创建接口标识符。有关 IEEE EUI-64 标准更多的信息，请访问 IEEE 标准网点：<http://standards.ieee.org/db/oui/tutorials/EUI64.html>。

这些 64 位接口标识符能在全球范围内逐个编址，并唯一地标识每个网络接口。这意味着理论上可多达 2^{64} 个不同的物理接口，大约有 1.8×10^{19} 个不同的地址，而且这也只用了 IPv6 地址空间的一半。这至少在可预见的未来是足够的。

2. 可集聚全球单播地址

本章已经提到了基于供应商的集聚，它的概念还会在第 8 章中再次提到。可集聚全球单播地址是另一种类型的集聚，它是独立于 ISP 的。基于供应商的可集聚地址必须随着供应商的改变而改变，而基于交换局的地址则由 IPv6 交换实体直接定位。由交换局提供地址块，而用户和供应商为网络接入签订合同。这样的网络接入或者是直接由供应商提供，或者通过交换局间接提供，但选路通过交换局。这就使得用户更换供应商时，无需重新编址。同时也允许用户使用多个 ISP 来处理单块网络地址。

可集聚全球单播地址包括地址格式的起始 3 位为 001 的所有地址 (此格式可在将来用于当前尚未分配的其他单播前缀)。地址格式化为图 6-3 所示的字段。

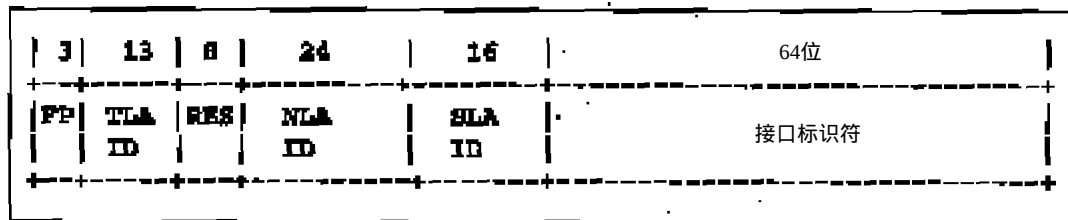


图6-3 RFC 2373中定义的IPv6全球可集聚单播地址格式

图中包括下列字段：

- FP 字段：IPv6 地址中的格式前缀，3 位长，用来标识该地址在 IPv6 地址空间中属于哪类地址。目前该字段为“001”，标识这是可集聚全球单播地址。
- TLA ID 字段：顶级集聚标识符，包含最高级地址选路信息。这指的是网络互连中最大的选路信息。目前，该字段为 13 位，可得到最大 8192 个不同的顶级路由。

- RES字段：该字段为8位，保留为将来用。最终可能会用于扩展顶级或下一级集聚标识符字段。
- NLA ID字段：下一级集聚标识符，24位长。该标识符被一些机构用于控制顶级集聚以安排地址空间。换句话说，这些机构(可能包括大型ISP和其他提供公网接入的机构)能按照他们自己的寻址分级结构来将此24位字段切开来。这样，一个实体可以用2位分割成4个实体内部的顶级路由，其余的22位地址空间分配给其他实体(如规模较小的本地ISP)。这些实体如果得到足够的地址空间，可将分配给它们的空间用同样的方法再子分。
- SLA ID字段：站点级集聚标识符，被一些机构用来安排内部的网络结构。每个机构可以用与IPv4同样的方法来创建自己内部的分级网络结构。若16位字段全部用作平面地址空间，则最多可有65 535个不同子网。如果用前8位作该组织内较高级的选路，那么允许255个高级子网，每个高级子网可有多达255个小子网。
- 接口标识符字段：64位长，包含IEEE EUI-64接口标识符的64位值。

现在很清楚，IPv6单播地址能包括大量的组合，甚至超过了将来RFC可能会指定的显式字段。不论是站点级集聚标识符，还是下一级集聚标识符都提供了大量空间，以便某些网络接入供应商和机构通过分级结构再子分这两个字段来增加附加的拓扑结构。

3. 特殊地址和保留地址

在第一个1/256 IPv6地址空间中，所有地址的第一个8位：0000 0000被保留。大部分空的地址空间用作特殊地址，这些特殊地址包括：

- 未指定地址：这是一个“全0”地址，当没有有效地址时，可采用该地址。例如当一个主机从网络第一次启动时，它尚未得到一个IPv6地址，就可以用这个地址，即当发出配置信息请求时，在IPv6包的源地址中填入该地址。该地址可表示为0:0:0:0:0:0:0:0，如前所述，也可写成::。
- 回返地址：在IPv4中，回返地址定义为127.0.0.1。任何发送回返地址的包必须通过协议栈到网络接口，但不发送到网络链路上。网络接口本身必须接受这些包，就好像是从外面节点收到的一样，并传回给协议栈。回返功能用来测试软件和配置。IPv6回返地址除了最低位外，全为0，即回返地址可表示为0:0:0:0:0:0:0:1或::1。
- 嵌有IPv4地址的IPv6地址：有两类地址，一类允许IPv6节点访问不支持IPv6的IPv4节点，另一类允许IPv6路由器用隧道方式，在IPv4网络上传送IPv6包。这两类地址将在下面进行讨论。

4. 嵌有IPv4地址的IPv6地址

不管人们是否愿意，逐渐向IPv6过渡已成定局。这意味着IPv4和IPv6节点必须找到共存的方法。当然两个不同IP版本最明显的一个差别是地址。最早由RFC 1884定义，然后被带入RFC 2373中，IPv6提供两类嵌有IPv4地址的特殊地址。这两类地址高阶80位均为0，低价32位包含IPv4地址。当中间的16位被置为FFFF时，则指示该地址为IPv4映象的IPv6地址。图6-4显示了这两类地址结构。

IPv4兼容地址被节点用于通过IPv4路由器以隧道方式传送IPv6包。这些节点既理解IPv4又理解IPv6。IPv4映象地址则被IPv6节点用于访问只支持IPv4的节点。这两类地址还将在第12章中讨论。

5. 链路本地和站点本地地址

对于不愿意申请全球唯一性的 IPv4 网络地址的一些机构，通过采用网络 10 型地址对 IPv4 网络地址进行翻译，可以为这些机构提供一个选项。位于机构之外，但由机构使用的路由器不应该转发这些地址，但是不能阻止转发这些地址，也不能区分这些地址和其他有效的 IPv4 地址。可以相对容易地配置路由器，使其能转发这些地址。

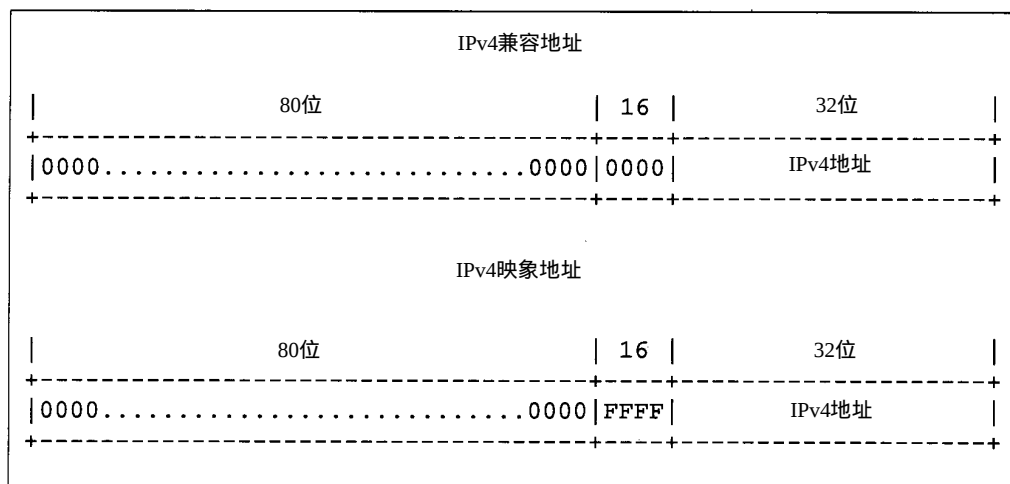


图6-4 RFC 2373定义的嵌入IPv4地址的IPv6地址

为实现这一功能，IPv6 从全球唯一的 Internet 空间中分出两个不同的地址段。图 6-5，源自 RFC 2373，显示了链路本地和站点本地地址的结构。

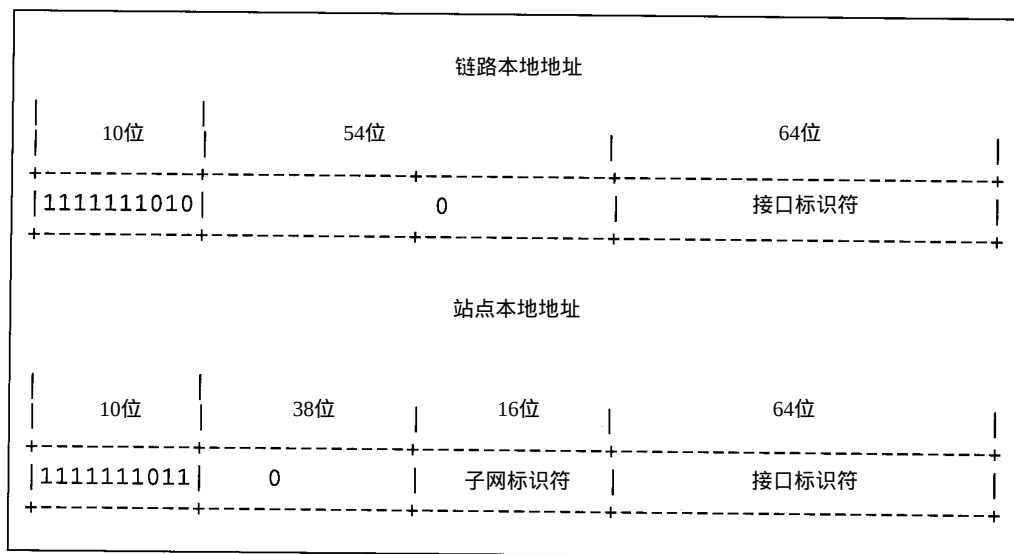


图6-5 RFC 2373中指定的链路本地和站点本地网络地址

链路本地地址用于单网络链路上给主机编号。前缀的前 10 位标识的地址即链路本地地址。路由器在它们的源端和目的端对具有链路本地地址的包不予处理，因为永远也不会转发这些包。该地址的中间 54 位置成 0。而 64 位接口标识符同样用如前所述的 IEEE 结构，地址空间的这部分允许个别网络连接多达 $(2^{64}-1)$ 个主机。

如果说链路本地地址只用于单个网络链路的话,那么站点本地地址则可用于站点。这意味着站点本地地址能用在内联网中传送数据,但不允许从站点直接选路到全球 Internet。站点内的路由器只能在站点内转发包,而不能把包转发到站点外去。站点本地地址的 10位前缀与链路本地地址的 10位前缀略有区别,然后后面紧跟一连串“0”。站点本地地址的子网标识符为16位,而接口标识符同样是 64位基于IEEE地址。

6. NSAP和IPX地址分配

IPng的目标之一是要统一整个网络世界,使 IP、IPX和OSI网络间能进行互操作。为了支持这种互操作性,IPv6为OSI和IPX各保留了 1/128地址空间。在本书写作时,IPX地址格式尚未精确定义;NSAP地址分配的描述见RFC 1888(OSI NSAP和IPv6)。对OSI和NSAP的讨论已超出本书范围,感兴趣的读者可以在 RFC中找到更完整的论述。

6.2.4 组播

像广播地址一样,组播地址在类似老式的以太网的本地网中特别有用,在这种网中,所有节点都能检测出线路上传输的所有数据。每次传输开始时,每个节点检查其目的地址,如果与本节点接口地址一致,节点就拾取该传输的其余部分。这使节点拾取广播和组播传输相对比较简单。如果是广播,节点只要侦听,无须做任何决定,因此简单。对组播来说,稍复杂一些,节点要预订一个组播地址,当检测出目的地址为组播地址时,必须确定是否是节点预定的那个组播地址。

IP组播就更为复杂。一个重要的原因是 IP并不是不加鉴别就将业务流放在 Internet上转发至所有节点,这是 IP成功之处。如果要这样做的话,它将迫使大多数甚至所有连接的网络屈服。这就是为什么路由器不应该转发广播包的原因。不过,对组播而言,只要路由器以其他节点的名义预订组播地址,就能有选择地转发它。

当节点预订组播地址时,它声明要成为组播的一个成员。于是任何本地路由器将以该节点的名义预订组播地址。同一网络上的其他节点要发送信息到该组播地址时,IP组播包将被封装到链路层组播数据传输单元中。在以太网上,封装的单元指向以太网组播地址;在其他用点对点电路传输的网络上(如ATM),通过其他某些机制将包发送给订户,通常通过某类服务器将包发送给每个订户。从本地网以外来的组播,用同样方法处理,只是传递给路由器,由路由器把包转发给预订节点。

1. 组播地址格式

IPv6组播地址的格式不同于 IPv6单播地址,采用图 6-6所示的更为严格的格式。组播地址只能用作目的地址,没有数据报把组播地址用作源地址。

地址格式中的第 1个字节为全“1”,标识其为组播地址。回顾图 6-1,组播地址占了 IPv6地址空间的整整 1/256。组播地址格式中除第 1字节外的其余部分,包括如下三个字段:

- 标志字段:由 4个单个位标志组成。目前只指定了第 4位,该位用来表示该地址是由 Internet编号机构指定的熟知的组播地址,还是特定场合使用的临时组播地址。如果该标志位为“0”,表示该地址为熟知地址;如果该位为“1”,表示该地址为临时地址。其他3个标志位保留将来用。
- 范围字段:长 4位,用来表示组播的范围。即,组播组是只包括同一本地网、同一站点、同一机构中的节点,还是包括 IPv6全球地址空间中任何位置的节点。该 4位的可能值为

0~15，见图6-7。

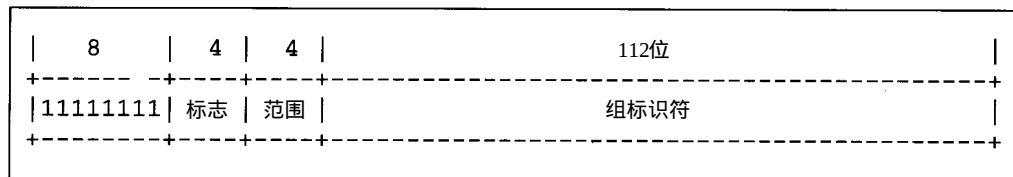


图6-6 RFC 2373中指定的IPv6组播地址格式

- 组标识符字段：长 112 位，用于标识组播组。根据组播地址是临时的还是熟知的以及地址的范围，同一个组播标识符可以表示不同的组。永久组播地址用指定的赋予特殊含义的组标识符，组中的成员既依赖于组标识符，又依赖于范围。

十六进制	十进制	值
0	0	保留
1	1	节点本地范围
2	2	链路本地范围
3	3	(未分配)
4	4	(未分配)
5	5	站点本地范围
6	6	(未分配)
7	7	(未分配)
8	8	机构本地范围
9	9	(未分配)
A	10	(未分配)
B	11	(未分配)
C	12	(未分配)
D	13	(未分配)
E	14	全球范围
F	15	保留

图6-7 RFC 2373中指定的IPv6组播范围值

所有IPv6组播地址以FF开始，表示地址的第1个8位为全“1”。目前，因为标志的其余位未定义，所以地址的第3个十六进制数字若为“0”，则表示熟知地址；若为“1”，则表示临时地址。第4个十六进制数字表示范围，可以是未分配的值或保留的值，见图6-7。

2. 组播组

IPv4已具备使用组播的应用，由于这种应用将同样的数据发送到多个节点，例如，电视会议或财经新闻及股票行情的发布，因而需要高带宽。用分配的组播地址和组播范围进行组合，可以表现出多种含义，并用在其他应用上。一些早期注册的组播地址，包括成组的路由器、DHCP服务、音频和视频服务以及网络游戏服务，详情请参阅 RFC 2375(IPv6组播地址分配)。

考虑组播组标识符为“所有DHCP服务器”时可能发生的情况。用组标识符 1:3来代表这个组。用2表示链路本地范围(本地网络链路)，则IPv6组播地址为FF02:0:0:0:0:0:1:3。该地址可解释为：链路本地范围内的所有DHCP服务器，即，所有DHCP服务器在同一网络上。如果将范围改为站点本地，那么该地址的意思变为“同一站点上的所有DHCP服务器”。

保留的组播组标识符可用于扩展范围字段。如果范围字段值为 1, 表示组标识符所指定的所有特定类型的服务器只包括本地节点上的服务器。如果范围字段值为 2, 除了包括本地节点上的服务器外, 再加上连接到同一网络的其他所有服务器。例如, 只有当一个网络时间协议 (NTP) 服务器运行在本地节点上时, 用组标识符标识范围值为 1 的该服务器将具有一个激活的成员; 如果范围值增至 2, 则包括连接到同一网络的运行一个 NTP 服务器的任何节点; 如果范围值增至 8, 它将包括运行在整个机构的所有 NTP 服务器; 如果范围值增至 E (十进制为 14), 它将包括互联网上任何地点的所有 NTP 服务器。

另一方面, 对于临时组播地址的组标识符, 在它们自己的范围以外没有意义。全球范围的临时组播组和链路本地的组, 即使它们可能有相同的组标识符, 也没有任何关系。

6.2.5 泛播

组播地址在某种意义上可以由多个节点共享。组播地址成员的所有节点均期待着接收发给该地址的所有包。一个连接 5 个不同的本地以太网网络的路由器, 要向每个网络转发一个组播包的副本 (假设每个网络上至少有一个预订了该组播地址)。泛播地址与组播地址类似, 同样是多个节点共享一个泛播地址, 不同的是, 只有一个节点期待接收给泛播地址的数据报。

泛播对提供某些类型的服务特别有用, 尤其是对于客户机和服务器之间不需要有特定关系的一些服务, 例如域名服务器和时间服务器。名字服务器就是个名字服务器, 不论远近都应该工作得一样好。同样, 一个近的时间服务器, 从准确性来说, 更为可取。因此当一个主机为了获取信息, 发出请求到泛播地址, 响应的应该是与该泛播地址相关联的最近的服务器。

1. 泛播地址的分配及其格式

泛播地址被分配在正常的 IPv6 单播地址空间以外。因为泛播地址在形式上与单播地址无法区分开, 一个泛播地址的每个成员, 必须显式地加以配置, 以便识别泛播地址。

2. 泛播选路

了解如何为一个单播包确定路由, 必须从指定单个单播地址的一组主机中提取最低的公共选路命名符。即, 它们必定有某些公共的网络地址号, 并且其前缀定义了所有泛播节点存在的地区。比如一个 ISP 可能要求它的每一个用户机构提供一个时间服务器, 这些时间服务器共享单个泛播地址。在这种情况下, 定义泛播地区的前缀, 被分配给 ISP 作再分发用。

发生在该地区中的选路由共享泛播地址的主机的分发来定义的。在该地区中, 一个泛播地址必定带有一个选路项: 该选路项包括一些指针, 指向共享该泛播地址的所有节点的网络接口。上述情况下, 地区限定在有限范围内。泛播主机也可能分散在全球 Internet 上, 如果是这种情况的话, 那么泛播地址必须添加到遍及世界的所有路由表上。

第7章 IPv6扩展头

本章讨论IPv6扩展头的含义、工作方式及与IPv4扩展头的区别，着重解释扩展头的顺序、使用方法，并讨论巨型报文、逐跳选项、目的地址选项、选路和分段头的使用。在第9章将对安全性头(身份验证头和封装安全性净荷头)进一步讨论。

7.1 扩展头

第5章介绍了一种新的IPv6扩展头，它作为简化的IPv6头，由工作在无选项方式的大多数网络业务流所采用，同时它提高了网络对确实需要选项的包的处理能力。以下扼要重述第5章的内容，这种新的IPv6扩展头包括：

- 逐跳选项头：此扩展头必须紧随在IPv6头之后，它包含包所经路径上的每个节点都必须检查的可选数据。到目前为止，只定义了一个选项：巨型净荷选项。该选项指明，此包的净荷长度超出了IPv6的16位净荷长度字段。只要包的净荷(包括逐跳选项头)超出65 535字节，就必须包含该选项。如果节点不能转发此包，则必须返回一个ICMPv6出错报文。
- 选路头：此扩展头指明包在到达目的地途中将经过的特殊的节点。它包含包沿途经过的各节点的地址列表。IPv6头的最初目的地址不是包的最终目的地址，而是选路头中所列的第一个地址。此地址对应的节点接收到该包后，对IPv6头和选路头进行处理，然后将包发送到选路头列表中的第二个地址。如此继续，直至该包到达最终目的地。
- 分段头：此扩展头包含一个分段偏移值、一个“更多段”标志和一个标识字段，用于源节点对长度超出源端和目的端间路径MTU的包进行分段。
- 目的地选项头：此扩展头包含只能由最终目的地节点所处理的选项。目前，只定义了填充选项，将该头填充为64位边界，以备将来所用。
- 身份验证头(AH)：此扩展头提供了一种机制，对IPv6头、扩展头和净荷的某些部分进行加密的较验和计算。
- 封装安全性净荷(ESP)头：这是最后一个扩展头，不进行加密，它指明剩余的净荷已经加密，并为已获得授权的节点提供足够的解密信息。

除了理解上述扩展头的功能之外，还有必要了解这些扩展头的使用方法、工作情况以及将来如何用于扩展IPv6。下面一节将描述这些扩展头的正确用法，后续小节将详细解释每个扩展头的工作过程，与安全性相关的扩展头的内容参见第9章。

7.2 扩展头的用法

将IPv4选项合并到标准IPv4头比较复杂。IPv4头最短为20字节，最长为60字节，附加数据包含IPv4选项，必须由路由器翻译以对IP包进行处理。这种方法有两个影响：其一，路由器实现时往往对附加选项的包进行分流处理，因此导致处理效率降低；其二，由于选项导致性能下降，应用开发者倾向于不使用选项。

使用IPv6扩展头，可以在不影响性能的前提下实现选项。开发者可以在必要时使用选项，

而无须担心路由器会对带扩展选项的包区别对待，除非是设置了选路扩展头或逐跳选项。即使设置了这两个选项，路由器仍可以进行必要的处理，比使用 IPv4选项容易。

7.2.1 扩展头的标识

所有的IPv6头长度都一样，并且看起来几乎相同，唯一的区别在于下一个头字段。在没有扩展头的IPv6包中，此字段的值表示上一层协议。即，若IP包中含有TCP段，则下一个头字段的8位二进制值是6(源自RFC 1700(已指派号码))；若IP包中含有UDP数据报，这个值就是17。表7-1中列举了下一个头字段的某些值。

下一个头字段值指明是否有下一个扩展头及下一个扩展头是什么，因此，IPv6头可以链接起来，从基本的IPv6头开始，逐个链接各扩展头。这种头连接链的构成见图7-1。图中第一个IPv6包没有扩展头；第二个包有选路扩展头，其后为TCP头和包的其余部分；最后一个包有更复杂的头链，IPv6头后面有分段扩展头，然后是身份验证扩展头，后接ESP扩展头，最后是TCP头和包的其余部分。

表7-1 IPv6下一个头字段的一些可能值，用以指明扩展头

下一个头字段值	描 述
0	逐跳头
43	选路头(RH)
44	分段头(FH)
51	身份验证头(AH)
52	封装安全性净荷(ESP)
59	没有下一个头
60	目的地选项头

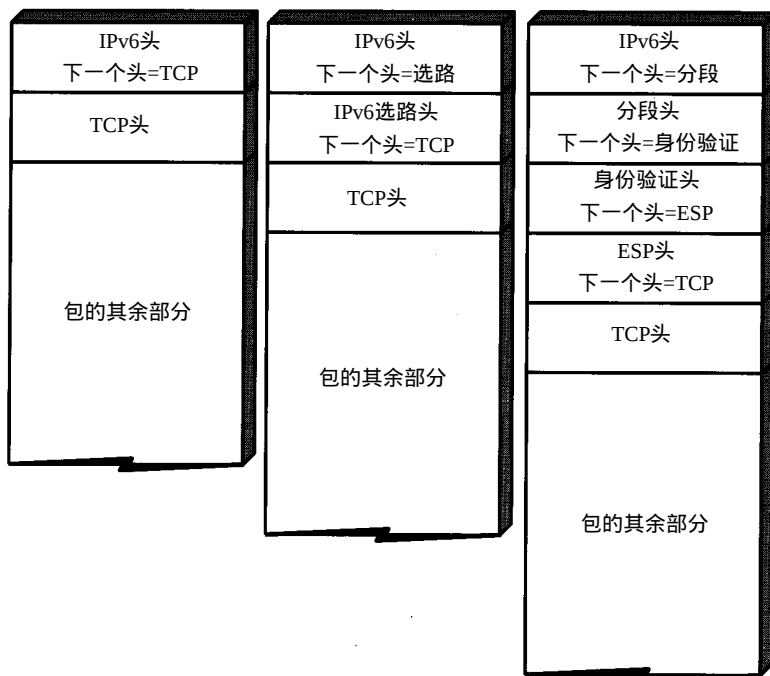


图7-1 三个不同的IPv6包：第一个包没有扩展头，第二个包有一个选路扩展头，第三个包有三个扩展头

7.2.2 扩展头的顺序

一个IPv6包可以有多个扩展头，但是，只有一种情况允许同一类型的扩展头在一个包中多次出现，而且各扩展头在链接时有一个首选顺序。RFC 1883规定，扩展头应该依照如下顺序：

- (1) IPv6头。
- (2) 逐跳选项头。
- (3) 目的地选项头(应用于IPv6目的地址字段的第一个目的地和选路头中所列的附加目的地中)。
- (4) 选路头。
- (5) 分段头。
- (6) 身份验证头。
- (7) ESP头。
- (8) 目的地选项头(当使用选路头时，仅应用于包的最终目的地)。
- (9) 上层头。

从以上顺序可知，在同一个IP包中只有目的地选项扩展头可以多次出现，并且仅限于包中包含选路扩展头的情况。

上述顺序并不是绝对的。例如，前面已提及，在包的其余部分要加密时，ESP头必须是最后一个扩展头。同样，逐跳选项优先于所有其他扩展头，因为每个接收IPv6包的节点都必须对该选项进行处理。

7.2.3 建立新的选项

扩展头必须通过IPv6头的下一个头字段来确认。这意味着由于这个字段为8位，最多只能有256个不同值。即使将来该字段的可能取值的个数有所减少，也必须支持上一层头的所有可能值。即，该值不仅对扩展头进行标识，还标识着封装在IP包内的所有其他协议。因此，目前已经指派了很多值，未指派的值相当有限。

IPv6用于扩展头的某些协议标识符沿自IPv4，例如身份验证头和ESP头。到目前为止，已指派了很多扩展头，但也允许通过逐跳选项扩展头和目的地选项扩展头来建立新的选项。

除了为下一个头字段保存协议值以外，通过使用这些选项头扩展，很容易健壮地实现新选项。如果使用一个全新的头类型来发送IP包，若目的节点支持新的头类型，则一切顺利；反之，如果新的头类型对目的节点是未知的，则目的节点只能丢弃该包。另一方面，所有的IPv6节点都必须支持逐跳选项扩展头、目的地选项扩展头以及一些基本选项(参见下节)。此时，如果目的节点收到带有目的地选项扩展头的包，即使不支持该扩展头中的选项，它也能够响应。即，这些选项可以向接收节点请求适当的响应，即使接收节点对选项并不理解。例如，选项可能是“做X，如果不理解X，就丢弃此包”这样的形式，或者可以是“做X，如果不理解X，就跳过此选项，并完成对扩展头的处理”。选项也可以请求目的节点发回一个ICMP出错报文，以指明目的节点不理解此选项。

7.2.4 选项扩展头

逐跳选项扩展头和目的地选项扩展头可以包含特定的选项。RFC 1883中定义了两个填充

选项，用于确保扩展头字段符合边界要求。即，如果选项使用 3 个 8 位字段后接一个 32 位字段，就必须插入(即填充)附加的 8 位，以确保在越过一个 32 位字边界时，选项中的 32 位字段不会被拆开。图 7-2 给出了该过程。如果无需填充，则只定义一个功能选项，即逐跳选项扩展头中使用的巨型净荷选项。

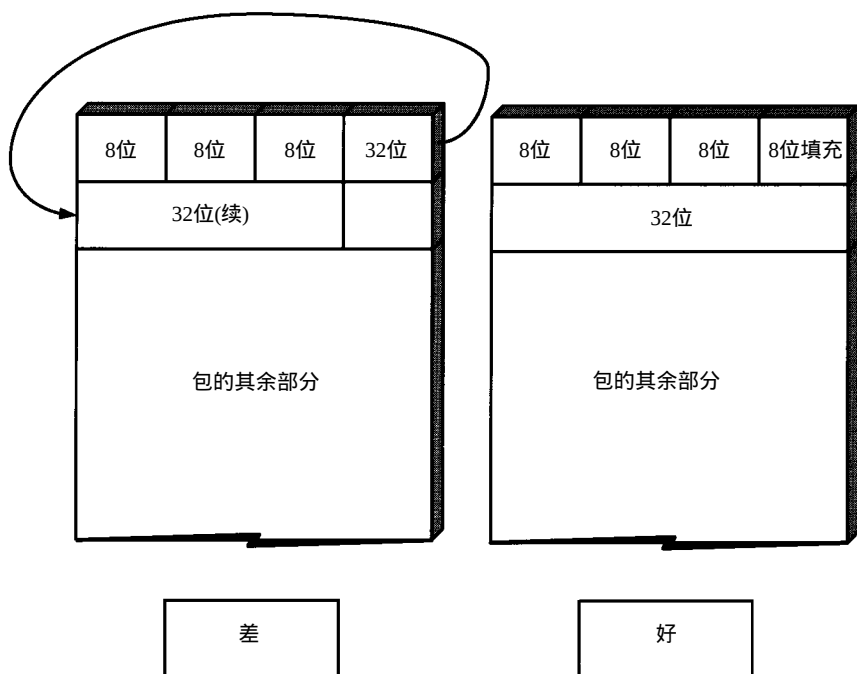


图7-2 选项头可能需要填充，以保证字段在越过32位字边界时不会被拆开

所有的选项扩展头——逐跳选项扩展头和目的地选项扩展头都有类似的帧格式，见图 7-3。很简单，这些扩展头只有两个预定义的字段：下一个头字段和头扩展长度字段。所有 IPv6 头都包含下一个头字段。头扩展长度字段占 8 位，指明该选项头的长度。该长度以 8 字节为单位，不包含扩展头的第一个 8 字节，即如果选项扩展头只有 8 字节长，该字段值即为 0。该字段限制了扩展头最多为 2048 字节。扩展头的其余部分为该扩展头所包含的选项。

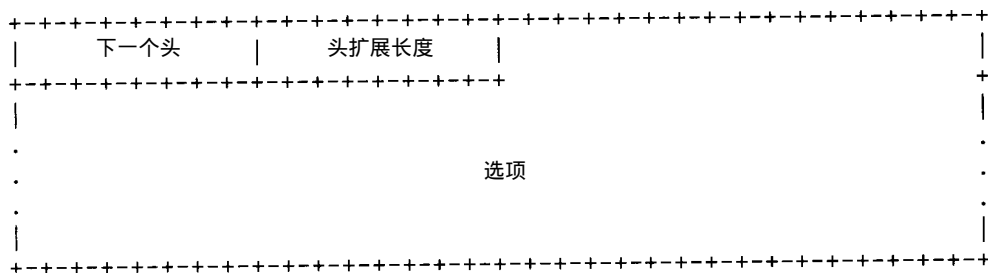


图7-3 RFC 1883中定义的标准选项头格式

7.2.5 选项

IPv6选项包含如下三个字段：

- 选项类型：该字段为 8 位标识符，指明选项的类型。即使目的节点不能够识别选项，也可以由该字段的前 3 位编码翻译出选项的类型。
- 选项数据长度：该字段为 8 位整数，表示选项数据字段的长度。该字段最大值为 255。
- 选项数据：该字段包含选项特定的数据，最大长度为 255 字节。

选项类型字段的前 2 位表示目的节点在不能识别特定的选项时应该采取的动作，共有如下四种选项类型：

- 00：忽略此选项，完成对扩展头其余部分的处理。
- 01：丢弃整个包。
- 10：丢弃包，不论该包的目的地地址是否是组播地址，都向该包的源地址发送一个 ICMP 报文。
- 11：丢弃包，如果该包的目的地地址是单播地址或任意点播地址（即非组播地址），则向该包的源地址发送一个 ICMP 报文。

选项类型的第 3 位指明在包从源地址到目的地址的传送过程中，选项数据的值是否可以改变。若为 0，则不允许改变；若为 1，则选项数据是可变的。

逐跳选项扩展头和目的地选项扩展头都包含的相同选项是两个填充选项：填充选项 1 和填充选项 N。填充选项 1 很特别，它只有 8 位，全部置为 0，没有选项数据长度字段和其他选项数据。

而填充选项 N 是由前面的四种选项类型之一来标识的，它使用多个字节来填充扩展头。如果扩展头需要 N 字节填充，则选项数据长度字段值为 N-2，即选项数据字段占 N-2 个字节，全部置为 0。再加上 1 字节的选项类型字段、1 字节的选项数据长度字段，一共填充了 N 字节。

7.3 逐跳选项

从源节点到目的节点的路由上的每个节点，即每个转发包的路由器都检查逐跳选项中的信息。到目前为止，只定义了一个逐跳选项：巨型净荷选项。图 7-4 描述了 RFC 1883 所定义的使用巨型净荷选项的逐跳扩展头。

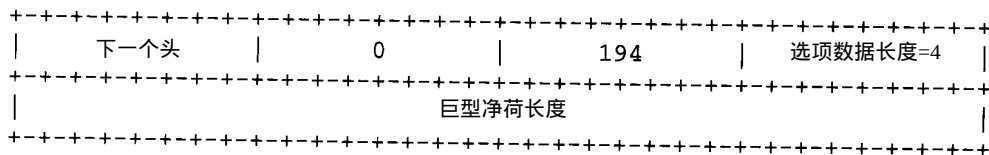


图7-4 RFC 1883中定义的包含巨型净荷选项的逐跳扩

展头，允许IPv6包中的净荷超过65 535字节

与其他选项扩展头相同，前两个字段指明了下一个头协议和扩展头的长度（此时，由于整个选项只有 8 位，扩展头长度的字段值为 0）。巨型净荷选项从扩展头的第三个字节开始。第三个字节为扩展头类型，其值为 194；第四个字节，即巨型净荷选项数据长度的值为 4。选项的最后一个字段为巨型净荷长度，指明包括逐跳选项扩展头在内，IP 包中所包含的实际字节数，但不包括 IPv6 头。

只有沿途每个路由器都能够处理时，节点才能使用巨型净荷选项来发送大型 IP 包。因此，该选项在逐跳扩展头中使用，要求沿途的每个路由器都必须检查此信息。

巨型净荷选项允许 IPv6 包净荷长度超过 65 535 字节，最多可以为 $2^{32} - 1$ 字节，超过了 40 亿

字节。如果使用该选项，要求 IPv6 头的 16 位净荷长度字段值必须为 0，扩展头中的巨型净荷长度字段值不小于 65 535。如果不满足这两个条件，接收包的节点应该向源节点发送 ICMP 出错报文，通知有问题发生。此外还有一个限制：如果包中有分段扩展头，就不能同时使用巨型净荷选项，因为使用巨型净荷选项时不能对包进行分段。

7.4 选路头

选路头代替了 IPv4 中所实现的源选路。源选路允许用户指定包的路径，即到达目的地沿途必须经过的路由器。在 IPv4 源选路中，使用 IPv4 选项，对用户指定的中间路由器的个数有一定限制：带扩展的 IPv4 头有 40 个附加字节，最多只能填入 10 个 32 位地址。此外，由于路径上的每个路由器都必须处理整个地址列表，而不论该路由器是否在列表中，因而对源路由包的处理很慢。

IPv6 定义了一个通用的选路扩展头，有两个字段，各占 1 字节：选路类型字段和剩余段数段。其中选路类型字段表示所使用的选路头的类型；而剩余段数段表示扩展头的其余部分所列出的附加路由器的个数，这些路由器是在到达最终目的地的途中包必须经过的。扩展头的其余部分为类型特定的数据，与选路头类型相关。RFC 1883 中定义了一种类型，即类型 0 选路头。

类型 0 选路扩展头解决了 IPv4 源选路的主要问题。只有列表中的路由器才处理选路头，其他路由器则不必处理。而且列表中最多可以指定 256 个路由器。对选路头的操作过程如下：

- 由源节点构造包必须经过的路由器的列表，并构造类型 0 选路头，头中包括路由器的列表、最终目的节点地址和剩余段数，剩余段数（8 位整数）指明在包向目的节点交付之前所必须经过的特定路由器的数目。
- 源节点发送包时，将 IPv6 头的目的地址设置为选路头列表中的第一个路由器的地址。
- 该包一直转发，直到到达路径中的第一站，即 IPv6 头的目的地址（选路头列表中的第一个路由器），只有该路由器才检查选路头，沿途的中间路由器都忽略选路头。
- 在第一站和所有后续其他站，路由器检查选路头以确保剩余段数与地址列表一致。若剩余段数的值等于 0，则表示此路由器节点实际上是该包的最终目的地，节点将继续对包的其他部分进行处理。
- 假定此节点不是该包的最终目的地，它将自己的地址从 IPv6 头的目的地址字段取出，并以选路头列表中的下一个节点地址来替代。同时，节点将剩余段数字段的值减 1。然后将包发送往下一站。列表中的其他节点重复此过程，直到包到达最终目的地。

RFC 1883 对类型 0 选路头的定义中，在剩余段数字段后保留了一个字节，并增加了 24 位严格/宽松位映射字段。该字段将 24 个标志映射到最多 24 个中间路由器，由此源节点可以指定使用严格选路还是宽松选路。严格选路不允许经过列表中不包含的中间路由器，而宽松选路则允许。目前没有采纳该方案，剩余段数字段之后的整个 32 位都作为保留位。未使用严格/宽松位映射字段表示头中所列举的路由器个数只受限于 8 位的剩余段数字段，当然也表示在类型 0 选路头中不能使用严格选路。

7.5 分段头

IPv6 只允许源节点对包进行分段，简化了中间节点对包的处理。而在 IPv4 中，对于超出本地链路允许长度的包，中间节点可以进行分段。这种处理方式要求路由器必须完成额外的

工作，并且在传输过程中包可能被多次分段。当一个节点要发送的包对于本地链路的单个数据传送单元来说太大时，就需要分段。例如，以太网允许传送的 MTU 为 1500 字节，要发送一个 4000 字节的 IP 包，如果不分成三段，每段均小于 1500 字节，就无法在以太网链路上传送。前方有些链路可能具有更小的 MTU，比如 576 字节，这种链路上的路由器就必须将已经分成 1500 字节的 IP 包分段，再次分成更小的段。

IPv4 中的分段很令人烦恼，它使得中间节点和目的节点都必须增加处理分段的必要开销。通过使用路径 MTU 发现机制，源节点可以确定源节点到目的节点之间的整个链路中能够传送的最大包长度，从而可以避免中间路由器的分段处理。RFC 1883 规定最小的 MTU 为 576 字节，但在将用来代替 RFC 1883 的文档草案中，最小的 MTU 要求已增加到 1280 字节，并建议将链路配置为应该至少可以传送 1500 字节长的包。

上述规定表明，源节点可以发送长达 1280 字节的包，而不必顾虑这些包会被分段。长达 1500 字节的包也很可能不被分段。但是，IPv6 规范建议所有节点都执行路径 MTU 发现机制，并只允许由源节点分段。换言之，在发送任意长度的包之前，必须检查由源节点到目的节点的路径，计算出可以无需分段而发送的最大长度的包。如果要发送超出此长度的包，就必须由源节点进行分段。

在 IPv6 中，分段只发生在源节点，并使用分段头来表示。RFC 1883 中规定的帧格式如图 7-5 所示。分段头字段包括：

- 下一个头字段：此 8 位字段对所有的 IPv6 头是共同的。
- 保留：此 8 位字段目前未用，设置为 0。
- 分段偏移值字段：与 IPv4 的分段偏移值字段很相似。此字段共 13 位，以 8 字节为单位，表示此包(分段)中数据的第一个字节与原来整个包中可分段部分的数据的第一个字节之间的位置关系。换言之，若该值为 175，表示分段中的数据从原包的第 1400 字节开始。
- 保留字段：此 2 位字段目前未用，设置为 0。
- M 标志：此位表示是否还有后续字段。若值为 1，表示后面还有后续字段；若值为 0 则表示这是最后一个分段。
- 标识字段：该字段与 IPv4 的标识字段类似，但是为 32 位，而在 IPv4 中为 16 位。源节点为每个被分段的 IPv6 包都分配一个 32 位标识符，用来唯一标识最近(在包的生存期内)从源地址发送到目的地址的包。

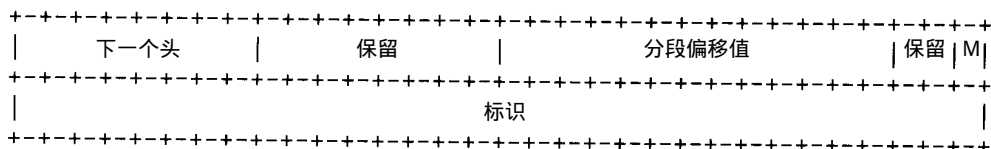


图7-5 RFC 1883中定义的IPv6分段扩展头字段

整个 IPv6 包中只有部分可以被分段，可分段的部分包括：净荷和只能在到达最终目的地时才处理的扩展头。对于 IPv6 头和在发往目的节点的途中必须由路由器处理的扩展头，如选路头或逐跳选项头，则不允许进行分段。

7.6 目的地选项

类似逐跳选项头，目的地选项头提供了一种随着 IPv6 包来交付可选信息的机制。其余的

扩展头选项，如分段头、身份验证头和 ESP头，都是每次出于某一个特定的理由而定义的，而目的地选项扩展头则是允许为目的节点而定义的新选项。目的地选项将使用前面所描述的构造选项的格式。

到目前为止，除了前面提到的填充选项，在已发布的 RFC中尚未定义任何目的地选项，但是Internet草案中定义了一些和移动 IP相关的选项，具体内容参见第 11章。

第8章 IPv6选路

本章首先讨论寻址和地址分配对 IP 网络的影响，然后讨论 IPv6 选路与 IPv4 选路的区别，重点介绍 IPv6 选路协议，也将讨论与选路相关的不同传输类型——单播、任意点播和组播。

8.1 地址对 IP 网络的影响

追溯至 70 年代后期、80 年代初期，IP 刚诞生的时候，几乎无人想到 IP 和 Internet 会发展为上万个不同网络、数千万个主机的规模。在描述早期 IP 实现的文档 RFC814(名字、地址、端口和路由)中，只使用了 32 位地址中的 8 位来标识网络。即，这些互联网络最多只支持 256 个网络。即使较复杂的实现也使用比较简单的寻址机制，即单个网络使用一个选路表项来指定，每个网络内部的单个主机使用一个主机表项来指定。

主机名和网络域名与主机地址和网络地址是通过简单的表链接到一起的。如果一个主机的网络地址改变了，例如由于网络重构而导致地址变化，就必须更新相关的表。如果一个网络域的地址改变，也必须更新选路表。主机地址的变化只需要在主机所在的域内进行更新，而网络地址的变化还需要对外部路由器的表进行更新。通过使用域名系统 (DNS) 服务器可以简化这种情形，而 DNS 还有待充分地规范和实现。在 DNS 的支持下，节点可访问 DNS 服务器以查询与主机名字相对应的网络地址。因此应用程序无需考虑 IP 地址，除非主机名所对应的 IP 地址可能改变。

然而，使用 IP 地址作为主机或节点的全球唯一标识已经有很长的历史，而且暂时还很稳定。即，不仅每个 IP 主机和网络都是通过唯一地址标识的，而且在一段时间内，该地址将保持不变。直到 90 年代中期，这种方案的效果一直很好。当 Internet 作为一种通信媒体，大规模地提供给各机构和个人访问，如同使用电信业务一样，此时 IP 地址的使用和分发也随之发生了变化。此前，大多数使用 IP 和 Internet 的公司直接向负责编址的授权机构申请网络地址和网络域，直接负责自己的 Internet(或 Internet 的前身，如 NSFNet 或 ARPANet) 连接，或与某些专业网络厂商(如 Bolt, Beranek 和 Newman，即 BBN) 合作负责。

但是，当 Internet 进入商用之后，情况就发生了变化。尤其是随着负责编址的授权机构对地址进行严格管理，单独的机构不再直接控制其 IP 地址。这些授权机构把编址任务交给 ISP 来代理，并且与 CIDR 共同使用，这样就可以对路由进行集聚。由于选路表的膨胀，集聚路由成为一个重要特性。

这种趋势导致 IP 寻址发生了巨大变化。首先，若一个机构改变了其 ISP，可能必须要随之改变其网络地址。其次，由于对 IP 地址的控制更加严格，一个有 500 个节点的机构可能只能得到 255 个节点的地址空间。本章将介绍一些与 IP 寻址机制相关的 IP 寻址分支，以及这些分支与 IPv6 选路的关系。

8.1.1 标识符和定位符

RFC 2101(目前 IPv4 地址行为)发布于 1997 年 2 月，该文档描述了 IPv4 地址的使用如何随时

间的推移而变化。它的要点在于对标识符和定位符的使用进行了区分。文档中将标识符定义为“两台主机的通信会话的整个生存期内使用的位串，用于对其中一台主机相对于另一台进行标识”。即，在用于Internet通信时，标识符看起来类似源主机的IP地址。而定位符被定义为“用于对某个特定包必须交付的位置进行标识的位串，例如它可用于在Internet拓扑中对目的主机所连接的位置进行定位”。即，定位符看起来类似目的主机的IP地址。

因此，标识符用于标识源端，而定位符用于标识目的端。这样做很直观，也很合理，主机IP地址既可以用作标识符，也可用作定位符。但是给予定位功能（即发现目的地）的优先权高于标识功能（即了解数据的源头）。即，与能够准确了解包的源头却不能交付该包相比，能够首先交付包然后再找出其源头更重要。

RFC 2101的作者指出，对于标识符和定位符的要求有两个重要区别：一是唯一性，二是持久性。

首先讨论唯一性。对于通信节点双方来说，标识符必须是唯一的，即各节点之间进行通信时，其标识符都必须唯一。有唯一合法IP地址的主机能够通过识别有唯一合法IP地址的任何其他主机，且连接到同一个互联网的所有此类主机都是唯一的。而另一方面，对于相互通信的路由器而言，定位符仅在某些情况下要求是唯一的。即，在同一选路域内，定位符必须唯一，但在不同的选路域内，定位符可以重叠。例如，一个路由器可以将10号网络连接到其他网络，但不能将两个或多个10号网络互相连接，否则即使规范没有禁止转发10号网络的包，路由器也不知道该向哪条链路上发送目的地址为10号网络的包。

现在考虑持久性。标识符的生存期要比定位符长。标识符至少要保持到两个节点间的通信结束。如果在通信过程中，一个节点的标识符有所改变，另一个节点则无法对后续包正确寻址。而另一方面，定位符只在相关的选路机制需要时才起作用。即，对于在节点通信过程中定位符改变的情况，路由器有能力进行处理。

目前，尽管定位符和标识符大多来源于节点的IPv4地址，但两者还是有不同的属性，且其理想化特性不同而且不一致。例如，理想的标识符只在节点初次安装到网络上时分配一次，其后永远不变。理想的标识符与一个节点相捆绑，并且只捆绑到一个节点，不能再重新使用或重新分配，这样就可以一直将该标识符与该节点相链接，而不会链接到其他节点。总之标识符的功能是将节点作为数据源进行标识。

而另一方面，定位符用于确定包必须向何处发送，它不需要持续很长时间，但是它应该描述在网络拓扑中节点实际所处的位置。这样，如果主机在网络拓扑中的位置由于某种原因而改变，定位符也随之改变。例如，如果主机从一个网络中迁移到另一个网络中，在理想情况下，其定位符应该改变。同样，如果主机所连接的网络重新编号，主机的定位符也随之变化。

RFC 2101的作者已注意到，不论作为定位符或标识符，IP地址都不理想。由于IP地址不再是全球唯一的，例如网络号10代表了共享同一网络和主机地址的相当大一部分IP节点，因此它不是理想的标识符。同时IP地址缺乏持久性的情况越来越多，因此它更不适合作为标识符。对于依靠DHCP来分配临时IP地址的网络，今天这个IP地址由一个节点使用，明天可能由另一个节点使用。

同样，IP地址作为定位符也有其不足。其一，网络号10无法说明此节点在互联网中的位置。其二，由于历史原因，网络地址无法说明该网络与其他网络的位置关系。当然随着越来越

越多的网络路由使用 CIDR 进行集聚, 这种情况发生了一些变化。某一 CIDR 块内的网络地址通常由负责该块的机构来处理。但是, 对于 B 类网络, 或在 CIDR 广泛应用之前已分配地址的网络, 其地址无法说明此网络在 Internet 中的位置。其三, 如果改变 ISP, 网络拓扑随之发生变化, 但是除非该机构对网络重新编号, 否则网络地址无法反映这种拓扑的变化, 而重新编号又使 IP 地址作为标识符的稳定性受到影响。

8.1.2 地址分配、无缝互操作和网络拓扑

RFC 2008(Internet 选路的不同地址分配策略的含义)发布于 1996 年 10 月, 该文档提出了有关 IP 选路的一些问题, 并描述了地址分配的“当前最好惯例”。此文档的基本前提是对“地址借出”方法的研究, 相对于传统的“地址所有权”方法, 该方法极大地改善了性能和扩展性。

换言之, 该 RFC 鼓励能够采用地址借出方法的机构使用由其 ISP 分配的 IP 地址, 一旦机构改变其 ISP, 地址也要随之变化。这意味着该机构是向 ISP 暂借其 IP 地址, 而 ISP 负责为客户集聚业务。通过集聚, ISP 只需维护更少的路由, 且 Internet 的整个扩展性得以改善。但是, 集聚也意味着如果机构决定改变 ISP, 就必须改变其 IP 地址, 以便新的 ISP 可以对其路由进行集聚。

另一种方法——地址所有权方法导致了地址表的急剧膨胀。但数据流是由其他节点导向一个特定的 IP 地址时, 即在使用 IP 地址作为标识符时, IP 地址还是有很好的实用价值。

然而, 使用 IP 地址作为标识符将导致很多问题。首先, 在处理网络业务、升级或改变节点功能方面, 用户因此损失了相当多的灵活性。使用 DNS, 用户可以将一个逻辑名字(如 www.loshin.com)捆绑到一个地址, 该地址可能随时间变化。例如, 用户很容易将其 web 站点的捆绑从已过时的 80486 微机上移到第三方 web 呈现供应商所运行的高端 SMP 服务器上, 用户只需将其逻辑名的 DNS 映射从一个 IP 地址改变为另一个 IP 地址。

灵活性的用途很大。在上例中, Internet 呈现供应商需要将用户的域名映射为自己服务器的 IP 地址, 需要为用户个人系统分配新的主机名。更重要的是, 必须要求 IP 应用程序只使用逻辑节点名, 而不能使用 IP 地址, 这样这些应用才能在 IPv4 和 IPv6 链路上无缝互操作。如果应用程序只涉及逻辑节点名, 就可以采用其他方法来实现节点名和节点地址的映射。有关取决于 IPv6 的协议这方面及其他方面的论题将在第 10 章讨论。

使用地址借出方法的最大优点是 IP 地址可以反映出网络拓扑。在图 8-1 中, ISP B 为 Acme 公司分配了一个网络地址, 这样该公司就可以通过 ISP B 连接到 ISP Q, 再通过 ISP Z 连接到 Internet。如果所有的 IP 地址都借给用户, 网络的性能和可扩展性就可以获得显著提高。对于要发送给 Acme 公司的包, 图中的源节点知道首先要选路到 ISP Z, 对于链接到 ISP Z 左边的所有网络, 都使用该路由。如图所示, 在 ISP Z 只有三条路由, 分别连接到其客户 ISP P、ISP Q 和 ISP R。同样, ISP Q 也只需要有三条路由, 分别连接到其客户 ISP A、ISP B 和 ISP C。

Acme 公司从 ISP B 借用 IP 地址, 如果改变 ISP, 就必须重新编号。例如, 该公司认为 ISP A 能提供更好的服务且价格更低。此时, ISP Q 就必须改变其选路表以呈现 Acme 公司的新地址, 但是对于已经向 Acme 公司发送的包, 更高层的 ISP 和 Internet 内另一侧的路由器仍然通过 ISP Z 进行选路。Acme 公司可以决定自己需要只来自上一层 ISP 的高层服务, 此时需要改变更多的路由。但是, 该公司应该注意到地址借出能够显著改善扩展能力和性能, 代价是一旦改变 ISP, 或其 ISP 改变了上级 ISP, 就必须对该公司的网络重新进行编号。RFC 2008 的作者指出如果使用某种 NAT 或应用网关, 可以无需对内部主机重新编号。他们还指出, 集聚和地址借出的目

的不应是构造尽可能小的选路表，而应是减缓现有选路表的增长速度，以确保额外的增长不会影响到选路的性能。

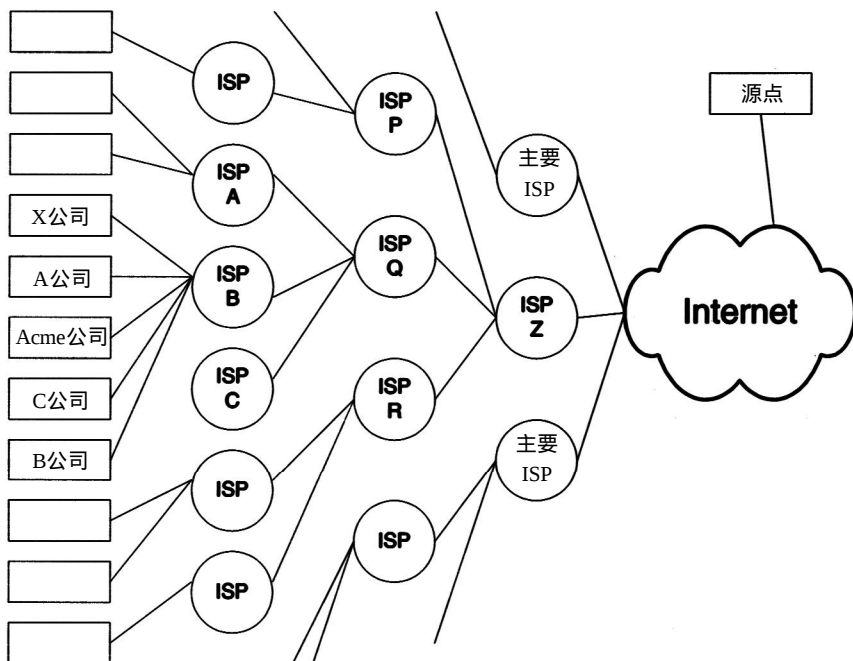


图8-1 IP路由集聚和地址借出使IP地址能反映Internet拓扑

本节提出的问题和IPv4寻址机制密切相关，但是这些RFC的写作都是以向IPv6升级为前提的，指出IPv4的可以改进之处有助于说明IPv6寻址从何处及如何改进。

8.2 选路问题

目前好像几乎人人都知道 IPv4网络地址即将耗尽。另一个问题却不是如此显而易见，即非默认路由器，或者是列出Internet上所有路由的路由器，即在Internet骨干网上或骨干网附近、因而必须知道全部路由的路由器，它们如何处理日益庞大的路由表。路由表中必须列出到达所有独立网络的路由，因此 CIDR广受欢迎。使用 CIDR，一个上述骨干路由器可以用一个涵盖8位地址空间的CIDR路由代替256个C类网络的256条路由。所有的256条路由可以经由一个Internet访问供应商来选路，因此 CIDR可以显著减少映射到Internet所需要的路由数目。

IPv6没有IPv4中的地址类别的概念。不论 A、B、C类地址的存在对于IPv4如何有用，长期以来这种分类都是对地址的浪费，对于网络地址体系结构，子网或超网能力好像用处更多。而且出于选路目的，IPv6地址可以积累起来，理论上有很大潜力可以显著地减少非默认选路表的大小。

当然，这种高度集聚的体系结构也有缺点，即一旦一个机构改变其供应商，就必须对网络重新编号。同样，多宿主网络可能引起更多的问题。实际上，基于供应商的 CIDR模式集聚方法的反对者把这个问题称为“专制”，他们已经提出了替代方案。很显然，这些替代方案在IPv6中没有采纳，但是这些方案有助于使自动配置和供应商移动性成为 IPv6过渡策略的关键部分。自动配置和供应商移动性的相关机制将在第11章中介绍。

看起来IPv6选路协议和IPv4选路协议好像没有显著的不同,这一点也许会很令人吃惊。毕竟IPv6寻址体系结构自身将显著改进选路效率,并减少非默认选路表的大小,因此选路算法和协议只需要进行极少的修改便可取得更好的执行效果。为支持 IPv6,对这些协议所做的修改大部分都与如何处理较长的IPv6地址有关。

IP选路协议

IP选路协议实质上可以分为链路状态协议和矢量距离(或路径矢量)协议两类,也可以按照内部选路和外部选路来分类。这两种分类方法看起来很简单,但足以满足本书的要求。

1. 内部选路和外部选路

内部选路和外部选路的概念对 Internet 的结构非常重要。这两个概念与 Internet 以及相连的互联网络之间的交互方式密切相关。例如,某个公司的内联网通过一条链路与 Internet 相连,该内联网与全球 Internet 之间的全部业务流都经由该链路来传送。如果这条链路中断,内联网就不再具有外部连接能力。这种类型的网络称为自治系统(AS),因为网络内部的一切都由单一的管理机构来管辖。这种系统的自治性体现在如果想要访问系统内的任何节点,全球 Internet 路由器只需要了解一条路由。同样,AS 内的任何节点可以使用默认路由来向 AS 之外的任何节点发送包。默认路由用于标识链接该 AS 与全球 Internet 的路由器。

内部选路参与 AS 内部包的选路。换言之,在相对小型的互联网络内的选路,所谓小型是相对于全球 Internet 而言。AS 内部的路由器保持的路由表相对较小,这些路由表包含到 AS 内部的子网和网络的路由,如果包要寻址的网络的地址没有明确列在路由表中,那么表中也包含一条或多条此时要选的默认路由。

另一方面,外部选路发生在 AS 环境之外。骨干路由器不能有默认路由。作为骨干路由器,如果要正确地发挥其作用,它就必须了解每个目的网络的显式路由。这意味着它的选路表将非常庞大,由此可见,使用诸如 CIDR 之类的集聚机制对于改善骨干选路性能极为重要。

如下所述,所有的选路协议都使用了链路状态选路算法和距离矢量选路算法中的一些要素。目前的 IPv4 外部路由器依靠 RFC 1771 中定义的边界网关协议 4(BGP-4),这是一种支持 CIDR 的距离矢量选路协议。尽管最初有一些评论(参见 Huitema 所著《IPv6 The New Internet Protocol》)认为 BGP 非常适用于 IPv4 的 32 位地址,但完全不适于 IPv6,但是 BGP-4 似乎仍将用于 IPv6 中的外部选路。有一个 Internet 草案中描述了为能够正确处理 IPv6 单播地址(链路本地地址、站点本地地址和全球地址)而进行的某些扩展,RFC 2283 中指出,经过上述扩展和支持多协议选路所进行的相对少量的修改,BGP-4 将能够处理 IPv6 外部选路。

另一个重要的外部选路协议是域间选路协议(IDRP),该协议源自 ISO/OSI 的努力。IDRP 在描述实质上规模无限的网络和支持网络地址体系结构方面提供了更大的灵活性。有些人认为 IDRP 是更好的 IPv6 外部选路协议,但是随着 Internet 的进一步发展,IDRP 是否能继续保持其重要性,这一点还存在质疑。

2. 链路状态和矢量距离协议

通常矢量距离协议较简单。选路信息协议(RIP)就是一个重要的矢量距离协议,该协议很简单,但是,因为它要求互联网络中的每个路由器都要周期性地向网络中所有其他路由器广播自己的选路信息,故该协议有一定的局限性。正如其名称所示,每个路由器所广播的信息包括路由和表示该路由长度的整数的列表。见图 8-2,网络中的路由器 1 到网络 A 的路径距离是

1跳，到网络B、D、F的路径距离是2跳，到网络C和E的路径距离是3跳。图中路由器2到网络B是单跳路径，到网络A是一条2跳路径。

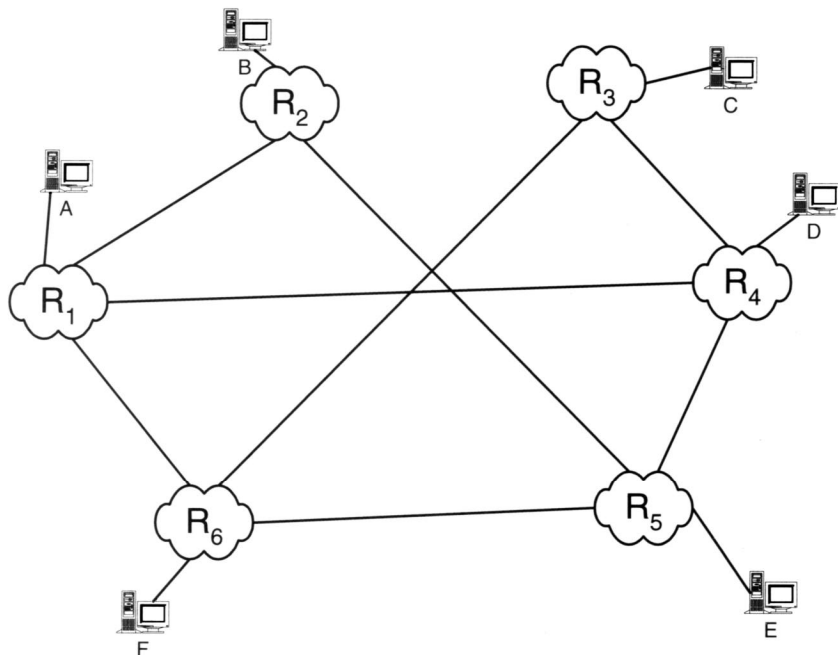


图8-2 路由器使用RIP链接而成的一个简单互连网络

使用这个广播信息，路由器2得知任何要发往网络A的包应该向路由器1选路，因为其他路由器没有通告到网络A的更短路径。同样，路由器2知道应该避免把发往网络A的包转发给路由器5，因为该路由器与网络A的距离为3跳。路由器2也清楚路由器3与网络A的距离为3跳，与自己的距离也为3跳，因此如果经由路由器3来转发给网络A将需要6跳。

显而易见，RIP有一些缺陷。首先，该协议“噪音”很大。每个路由器都频繁地发送状态信息，默认情况下每30秒就发送一个报文，这样随着互连网络规模的扩充、路由器数目的增加，整个网络的业务流将急剧增长。其次，由于RIP定义的限制，它只能支持不超过16跳的互连网络，即，如果路由的长度超过16跳，就不能使用RIP来选路。虽然RIP头可以支持的跳长度多达232-1，但由于选路收敛问题，开发者们在早期就确定RIP应该限制在16跳之内。所谓选路收敛问题，是指在不正确的路由通过网络传播之前，有关连通性问题的报文无法转播报给所有的路由器。

选路协议有必要允许包的选路独立于网络拓扑。这意味着源节点不必在内存中保留Internet的拓扑结构，也能够向网络中的任何目的地发送包。中间路由器应该了解网络的连通性，以便正确地转发包，但是它们也不必了解整个网络结构，只需要了解本地部分。因此，RIP之类的协议使得路由器能够获得来自其他路由器的有关它们的连接状态的通知。设计IP的目的是使其工作在既不完全可靠也不完全冗余的网络基础设施中。以图8-2为例，这意味着如果路由器6失效，来自网络E的包仍然可以发送到网络A，不是通过路由“路由器5-路由器6-路由器1”，而是通过替代路由“路由器5-路由器2-路由器1”。

问题是，如果路由器1到网络A的链接失效，网络A和互连网络之间就不再有连通性，即它被从互连网络切断了。但是，路由器1直接收到有关其他路由器与网络A之间的链路的信

息。所有这些路由至少是 2 跳，对于这些信息路由器 1 予以忽略，因为它自身到网络 A 只有 1 跳。为保证协议有效，必须矫正此问题及类似问题，但是相应地增加了复杂度。

上述问题并不说明 RIP 无用，对于小型或中型互联网（又称为内联网）而言，对网络带宽的限制并不重要，因而 RIP 还是很重要的选路协议。但是，对于大型内联网，RIP 不适用；当有很多不同路由需要考虑时，对于选路骨干业务流，RIP 也不适用。

针对矢量距离方法的缺陷，诸如开放的最短路径优先（OSPF）协议之类的链路状态协议得以大量引入。采用这种方法，路由器不是周期性地向所有其他路由器通知自己的所有路由，而是只通告自己的直接链路。见图 8-2，路由器 1 通告自己和网络 A、路由器 2、路由器 4 及路由器 6 有直接连接。其他的路由器也通告它们的直接连接，这样所有的路由器根据这些通告的报文就可以产生合适且理想的路由。如果路由器 5 得知与自己直接连接的路由器 2 又连接到路由器 1，而路由器 1 直接连接到网络 A，那么它就可以产生一条从网络 E 到网络 A 的路由。路由器只在连接改变或其他路由器询问时才发出通告，这样使用诸如 OSPF 之类的协议会减少与选路相关的噪声。由于没有 RIP 的跳限制，也不会在路由器间产生大量业务流，因而 OSPF 能够支持较大型网络。但是，OSPF 比 RIP 要复杂得多。对多层次的支持和对基于服务类型选路的支持也是 OSPF 的重要特性。

选路协议也可允许更多的选路信息。例如，路由器可以根据可用带宽、延时、甚至价格对某些链路分配不同的值。这样，在图 8-2 中，对于要发送给网络 A 的业务流，如果到路由器 4 的链路在某方面看起来更优（例如更短、更快或价格更低），路由器 5 可能就倾向于通过路由器 4 来选路。

3. IPv6 对选路协议的更新

如上所述，在为适合 IPv6 地址和地址范围而进行简单修改之后，BGP-4 和 IDRP 很可能继续用作 IP 的外部选路协议。同样，在 RFC 2080 (IPv6 的 RIPng) 中描述的 RIPng 与现有协议很相似。OSPF 版本 2 最近作为 Internet 标准在 RFC 2328 中定义，而用于 IPv6 的 OSPF 版本还在定义中。预计用于 IPv6 的 OSPF 将保留 OSPF 的概貌，包括基本特性和功能，并使之适合于处理 IPv6 选路。

第9章 IPv6身份验证和安全性

很多年来，人们一直在争论 IP 层是否需要身份验证和安全性及相关的用法问题。本章将讨论如何在 IPv6 中通过身份验证头 (AH) 和封装安全性净荷 (ESP) 头来实现身份验证和安全性，包括安全密码传输、加密和数据包的数字签名。但在探讨 IPv6 的安全性头之前，本章将首先介绍 IP 安全性体系结构以及在 IPv6 中该体系结构可能实现的部分。该体系结构在 RFC 1825 (IP 的安全性体系结构) 中首次进行了描述。

9.1 为 IP 增加安全性

IPv4 的目的只是作为简单的网络互通协议，因而其中没有包含安全特性。如果 IPv4 仅作为研究工具，或者在包括研究、军事、教育和政府网络的相对严格的辖区中作为产品型网络协议而使用，缺乏安全性并不是一个严重的缺陷。但是，随着 IP 网络在商用和消费网络中的重要性与日俱增，攻击所导致的潜在危害将具有空前的破坏性。

本节主要包括：

- 人们已经为 IP 定义的安全性目标。
- 这些目标如何满足。
- 这些目标和相关论题如何在 IP 中定义。

下一节将介绍 IP 的安全性体系结构 (又称为 IPsec) 本身的细节以及为完成上述目标而安装的一些工具。

应注意，RFC 1825 以及后续文档中所定义的 IPsec 提供的是 IP 的安全性体系结构，而不是 Internet 的安全性体系结构。两者的区别很重要：IPsec 定义了 IP 层使用的安全性服务，对 IPv4 和 IPv6 都可用。如果在适当的 IPv4 选项格式中实现 AH 和 ESP 头，IPv4 也可以使用这种安全性功能，只是在 IPv6 中更容易实现。

9.1.1 安全性目标

对于安全性，可以定义如下三个公认的目标：

- 身份验证：能够可靠地确定接收到的数据与发送的数据一致，并且确保发送该数据的实体与其所宣称的身份一致。
- 完整性：能够可靠地确定数据在从源到目的地传送的过程中没有被修改。
- 机密性：确保数据只能为预期的接收者使用或读出，而不能为其他任何实体使用或读出。

完整性和身份验证经常密切相关，而机密性有时使用公共密钥加密来实现，这样也有助于对源端进行身份验证。

AH 和 ESP 头有助于在 IP 上实现上述目标。很简单，AH 为源节点提供了在包上进行数字签名的机制。AH 之后的数据都是纯文本格式，可能被攻击者截取。但是，在目的节点接收之后，可以使用 AH 中包含的数据来进行身份验证。

另一方面，可以使用 ESP 头对数据进行加密。ESP 头之后的所有数据都进行了加密，ESP 头为接收者提供了足够的数据以对包的其余部分进行解密。

Internet 安全性(实际上任何一种安全性)的问题在于很难创建安全性，尤其是在开放的网络中，包可能经过任意数量的未知网络，任一个网络中都可能包嗅探器在工作，而任何网络都无法察觉。在这样的开放环境中，即使使用了加密和数字签名，安全性也将受到严重的威胁。对 IP 业务流的攻击也包括诸如侦听之类，致使从一个实体发往另一个实体的数据被未经授权的一个实体所窃取。此外，IP 安全性还应该解决下列安全性威胁：

- 否认服务攻击：即实体使用网络传送数据，致使某个授权用户无法访问网络资源。例如，攻击者可能使某主机淹没于大量请求中，从而致使系统崩溃；或者重复传送很长的 e-mail 报文，企图以恶意业务流塞满用户或站点带宽。
- 愚弄攻击：即实体传送虚报来源的包。例如，有一种愚弄攻击是由攻击者发送 e-mail 报文，报头的“From:”指明该报文的发信人是美国总统。那些在在包头携带错误源地址的攻击则更加阴险。

密钥处理问题则更加复杂。为使身份验证和加密更可靠，IP 安全性体系结构要求使用密钥。如何安全地管理和分配密钥，同时又能正确地将密钥与实体结合以避免中间者的攻击，这是 Internet 业界所面临的最棘手的问题之一。这种中间者的攻击是指，攻击者(假设为 C)将自己置于两个通信实体(假设为 A 和 B)之间，拦截 A 和 B 之间传送的所有数据，冒充 A 把数据重新发送给 B，也冒充 B 把数据重新发送给 A。如果 C 能够以类似 B 的公共密钥进行身份验证，从而让 A 确认它就是 B，同样也让 B 误以为它就是 A，那么 A 和 B 就会误认为他们之间的传送是安全的。

IPsec 本身不能使 Internet 更加安全。本章只提出与 Internet 安全性相关的几个最迫切的问题。对 Internet 安全性的细节感兴趣的读者，请参考本书作者的另一本书《Personal Encryption Clearly Explained》(AP Professional, 1998)，书中讨论了加密、数字签名和 Internet 安全性问题。

9.1.2 RFC 1825 及建议的更新

RFC 1825 于 1995 年 8 月发布，共有 22 页；其第 5 版修改草案完成于 1998 年 5 月，已经达到 66 页。安全性的正确实现要求认真考虑细节问题，这是对原 RFC 进行扩充的主要原因。更新后的文档在最终发布时，在关于如何实现所有的 IP 协议(包括 ICMP 和组播)方面将提供更多的细节，同时将更详细讨论密钥管理相关问题和安全性关联问题。

9.2 IPsec

IPsec 的目标是提供既可用于 IPv4 也可用于 IPv6 的安全性机制，该服务由 IP 层提供。一个系统可以使用 IPsec 来要求与其他系统的交互以安全的方式进行——通过使用特定的安全性算法和协议。IPsec 提供了必要的工具，用于一个系统与其他系统之间对彼此可接受的安全性进行协商。这意味着，一个系统可能有多个可接受的加密算法，这些算法允许该系统使用它所倾向的算法和其他系统协商，但如果其他系统不支持它的第一选择，则它也可以接受某些替代算法。

IPsec 中可能考虑如下安全性服务：

- 访问控制。如果没有正确的密码就不能访问一个服务或系统。可以调用安全性协议来控制密钥的安全交换，用户身份验证可以用于访问控制。
- 无连接的完整性。使用IPsec，有可能在不参照其他包的情况下，对任一单独的IP包进行完整性校验。此时每个包都是独立的，可以通过自身来确认。此功能可以通过使用安全散列技术来完成，它与使用检查数字类似，但可靠性更高，并且更不容易被未授权实体所篡改。
- 数据源身份验证。IPsec提供的又一项安全性服务是对IP包内包含的数据的来源进行标识。此功能通过使用数字签名算法来完成。
- 对包重放攻击的防御。作为无连接协议，IP很容易受到重放攻击的威胁。重放攻击是指攻击者发送一个目的主机已接收过的包，通过占用接收系统的资源，这种攻击使系统的可用性受到损害。为对付这种花招，IPsec提供了包计数器机制。
- 加密。数据机密性是指只允许身份验证正确者访问数据，对其他任何人一律不准。它是通过使用加密来提供的。
- 有限的业务流机密性。有时候只使用加密数据不足以保护系统。只要知道一次加密交换的末端点、交互的频度或有关数据传送的其他信息，坚决的攻击者就有足够的信息来使系统混乱或毁灭系统。通过使用IP隧道方法，尤其是与安全性网关共同使用，IPsec提供了有限的业务流机密性。

通过正确使用ESP头和AH，上述所有功能都有可能得以实现。目前，人们使用了很多密码功能，在下一节中将对予以简要描述。后续节将扼要描述密钥管理基础设施。

9.2.1 加密和身份验证算法

由于对安全性的攻击方法多种多样，设计者很难预计到所有的攻击方法，因此设计安全性算法和协议非常困难。普遍为人接受的关于安全性方法的观点是，一个好的加密算法或身份验证算法即使被攻击者了解，该算法也是安全的。这一点对于Internet安全性尤其重要。在Internet中，使用嗅探器的攻击者通过侦听系统与其连接协商，经常能够确切了解系统使用的是哪一种算法。

与Internet安全性相关的重要的密码功能大致有5类，包括对称加密、公共密钥加密、密钥交换、安全散列和数字签名。

1. 对称加密

大多数人都熟知对称加密这一加密方法。在这种方法中，每一方都使用相同的密钥来加密或解密。只要掌握了密钥，就可以破解使用此法加密的所有数据。这种方法有时也称作秘密密钥加密。通常对称加密效率很高，它是网络传送大量数据中最常用的一类加密方法。

常用的对称加密算法包括：

- 数据加密标准(DES)。DES首先由IBM公司在70年代提出，已成为国际标准。它有56位密钥。三重DES算法对DES略作变化，它使用DES算法三次加密数据，从而改进了安全性。
- RC2、RC4和RC5。这些密码算法提供了可变长度密钥加密方法，由一家安全性动态公司，RSA数据安全公司授权使用。目前网景公司的Navigator浏览器及其他很多Internet客户端和服务端产品使用了这些密码。

- 其他算法。包括在加拿大开发的用于 Nortel 公司 Entrust 产品的 CAST、国际数据加密算法 (IDEA)、传闻由前苏联安全局开发的 GOST 算法、由 Bruce Schneier 开发并在公共域发表的 Blowfish 算法及由美国国家安全局开发并用于 Clipper 芯片的契约密钥系统的 Skipjack 算法。

安全加密方法要求使用足够长的密钥。短密钥很容易为穷举攻击所破解。在穷举攻击中，攻击者使用计算机来对所有可能的密钥组合进行测试，很容易找到密钥。例如，长度为 40 位的密钥就不够安全，因为使用相对而言并不算昂贵的计算机来进行穷举攻击，在很短的时间内就可以破获密钥。同样，单 DES 算法已经被破解。一般而言，对于穷举攻击，在可预测的将来，128 位还可能是安全的。

对于其他类型的攻击，对称加密算法也比较脆弱。大多数使用对称加密算法的应用往往使用会话密钥，即一个密钥只用于一个会话的数据传送，或在一次会话中使用几个密钥。这样，如果会话密钥丢失，则只有在此会话中传送的数据受损，不会影响到较长时期内交换的大量数据。

2. 公共密钥加密

公共密钥加密算法使用一对密钥。公共密钥与秘密密钥相关联，公共密钥是公开的。以公共密钥加密的数据只能以秘密密钥来解密，同样可以用公共密钥来解密以秘密密钥加密的数据。这样只要实体的秘密密钥不泄露，其他实体就可以确信以公共密钥加密的数据只能由相应秘密密钥的持有者来解密。尽管公共密钥加密算法的效率不高，但它和数字签名（参见后续讨论）均是最常用的对网络传送的会话密钥进行加密的算法。

最常用的一类公共密钥加密算法是 RSA 算法，该算法由 Ron Rivest、Adi Shamir 和 Len Adleman 开发，由 RSA 数据安全公司授权使用。RSA 定义了用于选择和生成公共 / 秘密密钥对的机制，以及目前用于加密的数学函数。

3. 密钥交换

开放信道这种通信媒体上传送的数据可能被第三者窃听。在 Internet 这样的开放信道上要实现秘密共享难度很大。但是很有必要实现对共享秘密的处理，因为两个实体之间需要共享用于加密的密钥。关于如何在公共信道上安全地处理共享密钥这一问题，有一些重要的加密算法，是以对除预定接受者之外的任何人都保密的方式来实现的。

Diffie-Hellman 密钥交换算法允许实体间交换足够的信息以产生会话加密密钥。按照惯例，假设一个密码协议的两个参与者实体分别是 Alice 和 Bob，Alice 使用 Bob 的公开值和自己的秘密值来计算出一个值；Bob 也计算出自己的值并发给 Alice，然后双方使用自己的秘密值来计算他们的共享密钥。其中的数学计算相对比较简单，而且不属于本书讨论的范围。算法的概要：Bob 和 Alice 能够互相发送足够的信息给对方以计算出他们的共享密钥，但是这些信息却不足以让攻击者计算出密钥。

Diffie-Hellman 算法通常称为公共密钥算法，但它并不是一种公共密钥加密算法。该算法可用于计算密钥，但密钥必须和某种其他加密算法一起使用。但是，Diffie-Hellman 算法可用于身份验证。Network Associates 公司的 PGP 公共密钥软件中就使用了此算法。

密钥交换是构成任何完整的 Internet 安全性体系都必备的。此外，IPsec 安全性体系结构还包括 Internet 密钥交换 (IKE) 及 Internet 安全性关联和密钥管理协议 (ISAKMP)。在后续章节中将讨论这些标准和其他相关标准。

4. 安全散列

散列是一定量数据的数据摘要的一种排序。检查数字是简单的散列类型，而安全散列则产生较长的结果，经常是 128 位。对于良好的安全散列，攻击者很难颠倒设计或以其他方式毁灭。安全散列可以与密钥一起使用，也可以单独使用。其目的是提供报文的数字摘要，用来验证已经收到的数据是否与发送者所发送的相同。发送者计算散列并将其值包含在数据中，接收者对收到的数据进行散列计算，如果结果值与数据中所携带的散列值匹配，接收者就可以确认数据的完整性。

常用的散列方法由 RSA 数据安全公司提出，包括 MD2、MD4 和 MD5 报文摘要函数。安全散列算法 (SHA) 是由美国国家标准和技术协会 (NIST) 所开发的标准摘要函数。散列可以单独使用，也可以和数字签名一起使用。

5. 数字签名

前面提到的公共密钥加密依赖于密钥对，而数字签名则依靠公共密钥加密的特性，即允许数据以实体密钥对中的秘密密钥来加密，以公共密钥来解密。发送者首先对于要签名的数据进行安全散列计算，然后对结果使用秘密密钥加密。而接收者首先进行相同的散列计算，然后对发送者所附加的加密值进行解密。如果两次计算的值能够匹配，接收者就可以确信公共密钥的主人就是对报文签名的实体，且报文在传送中并没有被修改。

RSA 公共密钥加密算法可以用于数字签名。签名实体为待签名的数据建立散列，然后以自己的密钥对散列加密；证实实体则对接收到的数据进行相同的散列计算，使用签名实体的公共密钥对签名解密，并且比较所得的两个值。如果散列与解密的签名相同，则数据就得到证实。

数字签名有如下几种含义：

- 如果签名得到证实，说明所接收到的报文在从签名到接收的一段时间内未经任何改动。
- 如果不能证实签名，则说明或者是报文在传送过程中受到了破坏或篡改，或者是签名计算错误，又或者是签名在传送过程中被破坏或篡改。在上述任何情况下，未得到证实的签名并不一定是坏事，但是要求对报文重新签名并重传，以便最终能为接收者所接受。
- 如果签名得到证实，意味着与公共密钥相关联的实体是对报文签名的唯一实体。换言之，与公共密钥关联的实体不能否认自己的签名，这是数据签名的重要特性，称为不可抵赖。

还有其他机制可以实现数据签名，而 RSA 是其中应用最广泛的，并且已在大多数 Internet 产品中实现。

9.2.2 安全性关联

安全性关联是 (SA) IPsec 的基本概念。安全性关联包含能够唯一标识一个安全性连接的数据组合。连接是单方向的，每个 SA 由目的地址和安全性参数索引 (SPI) 来定义。其中 SPI 是对 RFC 1825 修改后的 Internet 草案中所要求的标识符，它说明使用 SA 的 IP 头类型，如 AH 或 ESP。SPI 为 32 位，用于对 SA 进行标识及区分同一个目的地址所链接的多个 SA。进行安全通信的两个系统有两个不同的 SA，每个目的地址对应一个。

每个 SA 还包括与连接协商的安全性类型相关的多个信息。这意味着系统必须了解其 SA、与 SA 目的主机所协商的加密或身份验证算法的类型、密钥长度和密钥生存期。

9.2.3 密钥管理

如何管理密钥是 Internet 安全性专业人士面临的最复杂的问题之一。密钥管理不仅包括使用密钥协议来分发密钥，还包括在通信系统之间对密钥的长度、生存期和密钥算法进行协商。Internet 工作组和研究团体对此已进行了大量工作，但是由于尚未达成一致，目前还没有发表任何 RFC。

Internet 安全性关联密钥管理协议 (ISAKMP) 为密钥的安全交换定义了整个基本构架。ISAKMP 实际上是一个应用协议，协议中定义了用于系统之间协商密钥交换的不同类型报文，它在传输层使用 UDP。

但是 ISAKMP 只是特定机制所使用的框架，而没有定义实际完成交换的机制和算法。这些年来在不同的建议中定义了大量的交换机制，通常以 Diffie-Hellman 密钥交换为基础。主要的提案包括：

- Photuris。此提案基于 Diffie-Hellman 算法，但增加了要求，即要求节点首先发送一个 cookie(一个随机数)，然后服务器给予应答，这样减少了否认服务攻击的威胁(否认服务攻击是由攻击者伪造源地址而导致的)。Photuris 也要求通信各方都必须对协商好的密钥签名，以减少中间者攻击的危害(所谓中间者攻击，是指某个攻击者对系统的 Alice 冒充自己是 Bob，又对另一个系统的 Bob 冒充自己是 Alice)。
- Sun 公司的 Internet 协议的简单密钥管理 (SKIP)。SKIP 也是以 Diffie-Hellman 密钥交换为基础，但是它并不要求通信各方使用随机数来计算其密钥，而是要求使用静态的密码表。各方查找密码表中的秘密值，然后基于查到的秘密值来计算，并传送所算出的值。
- OAKLEY。此机制与 Photuris 有某些相似特性，但在不考虑否认服务攻击的情形下，它提供不同的密钥交换模式。

1998 年秋，基于 OAKLEY 和 SKEME (Internet 的安全密钥交换机制)，Internet 密钥交换最终在 Internet 密钥交换规范中得以定义。

读者应该注意到，人工密钥管理也是一个重要选项，而且在很多情况下是唯一的选项。人工方法要求个人单独交付密钥，并使用密钥来配置网络设备。即使在开放标准已经充分确定并且实现之后，人工密钥管理仍将继续是一个重要选择，对于商业产品尤其如此。

9.2.4 实现 IPsec

IP 层安全性用于保护 IP 数据报。它不一定要涉及用户或应用。这意味着用户可以愉快地使用应用程序，而无需注意所有的数据报在发送到 Internet 之前，需要进行加密或身份验证，当然在这种情形下所有的加密数据报都要由另一端的主机正确地解密。

这样就引入了如何实现 IPsec 的问题，有如下三种可能方法：

- 将 IPsec 作为 IPv4 栈或 IPv6 栈的一部分来实现。这种方法将 IP 安全性支持引入 IP 网络栈，并且作为任何 IP 实现的一个必备部分。但是，这种方法也要求对整个实体栈进行更新以反映上述改变。
- 将 IPsec 作为“栈中的一块”(BITS)来实现。这种方法将特殊的 IPsec 代码插入到网络栈中，在现有 IP 网络软件之下、本地链路软件之上。换言之，这种方法通过一段软件来实现安全性，该软件截获从现有 IP 栈向本地链路层接口传送的数据报，对这些数据报进行必要的安全性处理，然后再交给链路层。这种方法可用于将现有系统升级为支持 IPsec

的系统，且不要求重写原有的IP栈软件。

- 将IPsec作为“线路的一块”(BITW)来实现。这种方法使用外部加密硬件来执行安全性处理功能。该硬件设备通常是作为一种路由器使用的IP设备，或者更确切一些，是安全性网关，此网关为位于它后面的所有系统发送的IP数据报服务。如果这样的设备只用于一个主机，其工作情况与BITS方法类似，但如果一个BITW设备为多个系统服务，实现相对要复杂得多。

上述各种方法的差别不在于字面上，而在于它们的适用情况不同。要求高级别安全性的应用最好使用硬件方法实现；而如果系统不具备与新的IPsec兼容的网络栈，应用最好选择BITS方法。

9.2.5 隧道模式与透明模式

本书在后续章节中讨论移植策略时，还将涉及协议隧道概念。而对于IP安全性，隧道同样重要。见图9-1，两个系统建立了SA，以便在Internet上安全地通信。其中一个系统产生网络业务流，经过加密或者签名，然后发送给目的系统。而在接收方，首先对收到的数据报进行解密或者身份验证，把净荷向上传送给接收系统的网络栈，由使用数据的应用进行最后的处理。两个主机之间的通信如同没有安全性头一样简单，而且数据报实际的IP头必须要暴露出来以便在Internet选路，因此这种方法称为使用SA的透明模式。

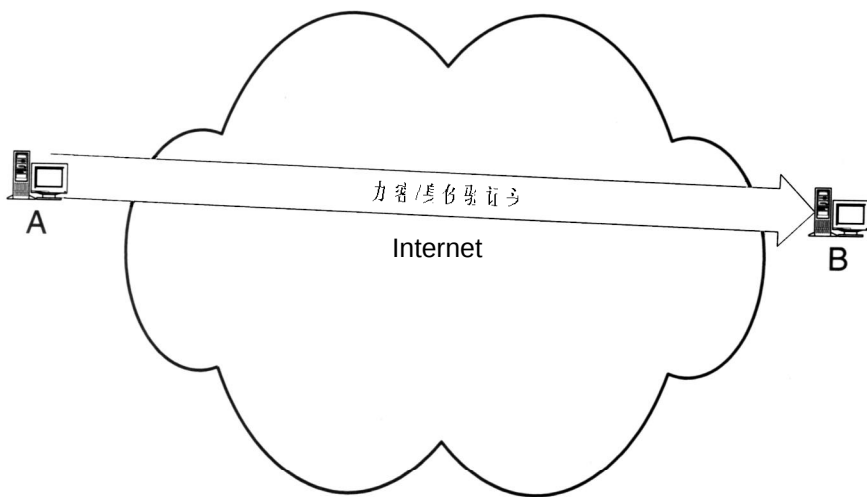


图9-1 一对主机使用IPsec进行透明通信

SA也可以用来将安全IP以隧道方式通过互连网络。见图9-2，来自系统A的所有IP包首先转发到安全性网关X，由X建立一条跨越Internet、目的地为安全性网关Y的隧道，由Y对经隧道方式传来的数据拆包并转发。安全性网关Y可能将包转发给本地互连网络内的任一主机B、C或D，也可能转发给外部主机，如M。这取决于源主机如何为这些包定向。如果SA目的节点是安全性网关，则称为隧道关联。即，隧道传送既可以在两个安全性网关之间进行（见图9-2），也可以在正规节点和安全性网关之间进行。因此，图9-2中的主机M可以与安全性网关X或Y建立隧道连接，M所发送的数据报首先传送给安全性网关，然后经过网关解密或身份验证之后，再进行正确地转发，由此可见这是一种隧道方式。

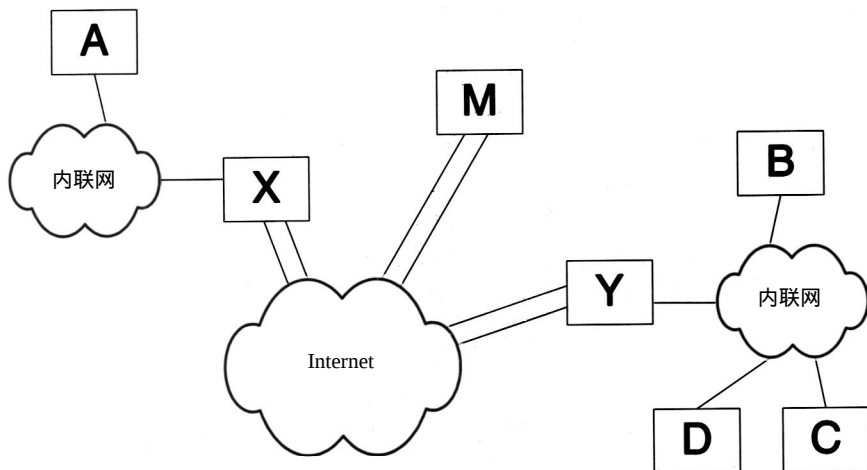


图9-2 IP安全性隧道

9.3 IPv6安全性头

如前所述，IPsec安全性服务完全通过AH和封装安全性净荷(ESP)头相结合的机制来提供，当然还要有正确的相关密钥管理协议。RFC 1826(IP身份验证头)中对AH进行了描述，而ESP头在RFC 1827(IP封装安全性净荷(ESP))中描述。上述RFC及IP安全性体系结构RFC仅仅是解决安全性问题的第一步。IPsec工作组各成员正继续对这些扩展头的规范进行改进，这些文档的当前草案的篇幅几乎是原RFC的两倍。这些草案保留了原RFC的语言和意图，并进行了扩充，对包头及其功能的描述更加完整，综合性更强。

各安全性头可以单独使用，也可以一起使用。如果一起使用多个扩展头，AH应置于ESP头之前，这样，首先进行身份验证，然后再对ESP头净荷解密。使用IPsec隧道时，这些扩展头也可以嵌套。即，源节点对IP包进行加密和数字签名，然后发送给本地安全性网关，该网关则再次进行加密和数字签名，然后发送给另一个安全性网关。

AH和ESP头既可以用于IPv4，也可以用于IPv6，这一点很重要。本节将讨论这些安全性扩展头在IPv6中如何使用，对于IPv4，这些扩展头作为选项加在正常的IPv4头中。

9.3.1 身份验证头

AH的作用如下：

- 为IP数据报提供强大的完整性服务，这意味着AH可用于为IP数据报承载内容验证数据。
- 为IP数据报提供强大的身份验证，这意味着AH可用于将实体与数据报内容相链接。
- 如果在完整性服务中使用了公共密钥数字签名算法，AH可以为IP数据报提供不可抵赖服务。
- 通过使用顺序号字段来防止重放攻击。

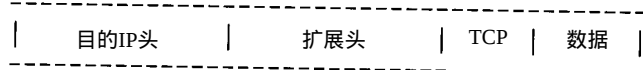
AH可以在隧道模式或透明模式下使用，这意味着它既可用于为两个节点间的简单直接的数据报传送提供身份验证和保护，也可用于对发给安全性网关或由安全性网关发出的整个数据报流进行封装。

1. 语义

IPv6中的AH与其他扩展头一起使用时，必须置于那些将由中间路由器处理的扩展头之后，及那些只能由数据报目的地处理的扩展头之前。这意味着 AH应置于逐跳扩展头、选路扩展头或分段扩展头之后。根据不同情况，AH可在目的地选项扩展头之前，也可在其后。

在透明模式中，AH保护初始IP数据报的净荷，也保护在逐跳转发中不变化的部分 IP头，如跳极限字段或选路扩展头。图9-3中显示了在透明模式中，当计算和增加AH时，IP数据报的变化情况。图中的目的地选项头也可以置于AH之前。对于目的IP地址和扩展头，仅在逐跳转发它们不发生变化的情况下，才能得到保护。

计算AH之前的数据报



插入AH之后的数据报

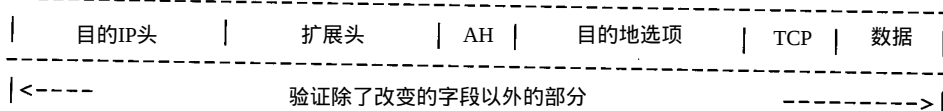
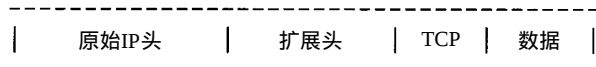


图9-3 在透明模式中为IP数据报增加AH

当AH用于隧道模式中时，使用方法与上不同。图9-4表明了其中的区别。初始的目的IP地址与整个初始IP数据报一起，封装在全新的IP数据报中，该数据报再发送到安全性网关。因此，整个初始IP数据报以及传送中不变的封装IP头部分都得以保护。

原始IP数据报



使用隧道方式向安全性网关发送的IP数据报(GW)

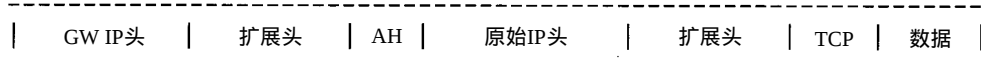


图9-4 在隧道模式中为IP数据报增加AH

2. AH字段

图9-5表示了AH的格式和各字段。与所有的IPv6扩展头一样，第一个字段是8位的下一个头字段，它表示后续的扩展头协议。其他字段包括：

(1) 净荷长度。此8位字段指明AH的整个长度，其值以32位字为单位，并减去2。正如初始的定义，AH包含64位，其余部分为身份验证数据(参见后续内容)。因此净荷长度字段只指出身份验证数据以32位字为单位的长度。加入序列号字段(参见后续内容)后，此值等于身份验证数据加上序列号字段的长度。

(2) 保留。净荷长度字段之后的16位为将来使用而保留。目前，此16位必须全部置为0。

(3) 安全性参数索引(SPI)。此32位字段是一个任意数。与目的IP地址和安全性协议一起使用，SPI是AH使用的SA的唯一标识。若SPI值为0，则表示只用于本地而不予传送；值1~255被Internet分配号码授权机构(IANA)保留作将来使用。

(4) 序列号。此32位字段是一个必备的计数器，由发送者插入IP头，但不一定由接收者使

用。从0开始，每发送一个数据报，该计数器增1，这可用于预防重放攻击。若接收者使用此字段来对抗重放攻击，对于序列号与已收到的数据报相同的数据报，接收者将予以丢弃。这意味着若计数器重新开始循环，即已经接收到 2^{32} 个数据报，则必须协商新的 SA。否则，一旦计数器重新置位，接收系统将丢弃所有的数据报。

(5) 身份验证数据。此字段包含完整性检查值 (ICV)，这是 AH 的核心。其内容的长度必须是32位的整数倍，为满足这个条件，其中可能包含填充字段。下节将讨论该值的计算。

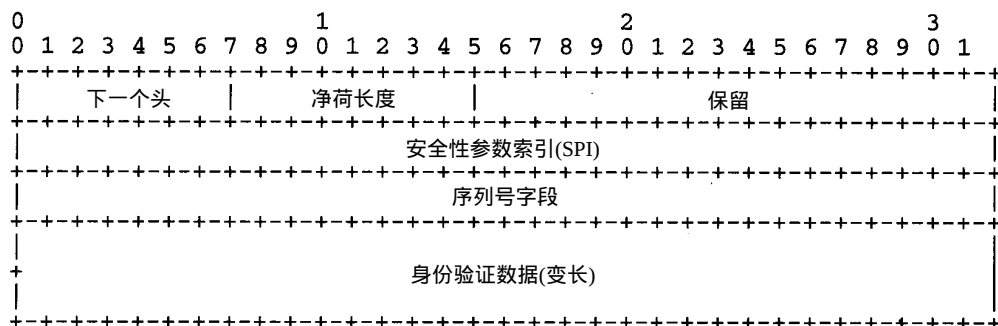


图9-5 AH格式和字段

3. 计算完整性检查值

对于如何计算ICV以及使用什么机制来计算，RFC 1826的描述比较模糊。实际上，术语“完整性检查值”在该文档中并没有出现，而是出现在将要代替 RFC 1826的后续草案中。预期适当的身份验证算法将导致ICV的产生。建议的算法包括：

- 报文身份验证代码(MAC)，然后对其结果用适当的对称加密算法(如DES)进行加密。
- 安全散列功能，如MD5或SHA的更新版SHA-1。

按照标准的约定，预计 AH 的任何实现将必须支持 MD5 和 SHA-1 密钥散列。身份验证数据针对整个 IP 数据报净荷以及 IP 头的不变部分或可预测部分来计算。

9.3.2 封装安全性净荷头

ESP头被用于允许IP节点发送和接收净荷经过加密的数据报。更确切一点，ESP头是为了提供几种不同的服务，其中某些服务与 AH 有所重叠。ESP头提供的服务包括：

- 通过加密提供数据报的机密性。
- 通过使用公共密钥加密对数据来源进行身份验证。
- 通过由 AH 提供的序列号机制提供对抗重放服务。
- 通过使用安全性网关来提供有限的业务流机密性。

ESP头可以和 AH 结合使用。实际上，如果 ESP 头不使用身份验证的机制，建议将 AH 和 ESP 头一起使用。

1. 语义

ESP头必须跟随在去往目的节点所途经的中间节点需要处理的扩展头之后，ESP头之后的数据都可能被加密。实际上，加密的净荷是作为 ESP 头的最后一个字段(参见后续内容)。

与 AH 类似，ESP 既可用于隧道模式，也可用于透明模式。在透明模式中，如果有 AH，IP 头以及逐跳扩展头、选路扩展头或分段扩展头都在 AH 之前，其后跟随 ESP 头。任何目的地选

项头可以在 ESP 头之前，也可以在 ESP 头之后，或者 ESP 头前后都有，而 ESP 头之后的扩展头将被加密。

在很多方面，仅仅是常规数据报带着加密净荷从源端传送到目的端。某些情况下，适合在透明模式中使用 ESP。但是，这种模式使攻击者有可能研究两个节点之间的业务流，留意正在通信的节点、节点之间交换的数据量、交换的时间等。所有这些信息都可能为攻击者提供有助于对通信双方进行攻击的信息。

类似前面描述的 AH 的情形，使用安全性网关是一种替代方法。安全性网关可以直接与节点连接，也可以链接到另一个安全性网关。单个节点可以在隧道模式中使用 ESP，即加密所有出境包，并封装到单独的 IP 数据报流中，再发送给安全性网关。然后网关解密业务流，并重新将原始 IP 数据报发往目的地。

使用隧道模式时，ESP 头对整个 IP 数据报进行封装，并作为 IP 头的扩展将数据报定向到安全性网关。ESP 头与 AH 的结合也有几种不同方式，例如以隧道方法传送的数据报可能有透明模式的 AH。

2. 字段

ESP 头与其他扩展头不同。其一，下一个头字段的位置接近 ESP 头的末端。其二，ESP 头之前的扩展头将其下一个头字段值置为 50，以指明随后是 ESP 头。ESP 头的其余部分将可能包括如下字段：

- 安全性参数索引(SPI)。与上节提到的 AH 中的 32 位 SPI 值相同。通信节点使用该值来指出 SA，SA 用于确定数据应如何加密。
- 序列号。32 位，从 0 开始，每发送一个数据报，该值加 1。如前所述，序列号可用于防御重放攻击，在循环用完所有 2^{32} 个值之前，必须建立新的 SA。
- 净荷数据。此字段长度可变，它实际上包含数据报的加密部分以及加密算法需要的补充数据，例如初始化数据。
- 填充。头的加密部分(净荷)必须在正确的边界终止，因此有时需要填充。
- 填充长度。此字段指明净荷数据所需要填充的数据量。
- 下一个头：此字段像其他 IPv6 扩展头中的字段一样操作，但是它不位于扩展头的开始，而是靠近扩展头末端。
- 身份验证数据。此字段是一个 ICV，它对除身份验证数据本身之外的整个 ESP 头进行计算。这种身份验证计算是可选的。

3. 进行封装

预计一个兼容的 ESP 实现至少要求支持 DES 加密和 SHA-1 身份验证。它也可以支持其他算法，但支持上述两个算法是最低要求。

第10章 相关的下一代协议

在TCP/IP协议集内，与IP直接或间接互操作的协议包括各种应用层协议、链路层协议以及TCP和UDP。本章将探讨如何对这些协议进行修改或者是否必须修改以适应IPv6。

10.1 协议的层次

回顾第2章，TCP/IP网络技术依赖于层次概念。在每一层，两个实体可以彼此通信，使用相邻下层来封装其数据。应用协议通常从一个节点到另一个节点来定义应用程序彼此通信的方式。来自应用层的数据由传输层协议进行封装，然后再封装在网际层协议内，最后封装在链路层协议中。

要理解IPv6对其他层协议的影响，重要的是要理解这些协议如何使用IP。由于如此众多的系统要依靠大量的TCP/IP网络技术和应用协议，重要的是对IP的升级不一定要对IP的上层或下层协议进行广泛的升级。因此，除IPv4之外，大多数现有TCP/IP应用、软件和硬件只需要进行少量修改，就可以和IPv6一起工作。

10.1.1 应用层

万维网(WWW)和e-mail是当今使用最广泛的应用。WWW和e-mail客户必须指向Internet上的服务器才能工作。传统上这些客户能够接受节点的主机名或其IP地址。在使用域名时，可调用域名系统(DNS)来获得与主机名对应的IP地址，然后在传输层和网际层使用。

对于简单的应用，很容易使其与IPv6一起工作：可以重写软件，使其能接受和正确处理IPv4地址和IPv6地址；或者要求只能按主机名来访问。前一种方法保留了应用直接对节点寻址的能力，但相对而言比较复杂；而后一种方法只是去掉了大多数用户不使用甚至不需要的功能。

但是，考虑到对IPv6提供的安全性、服务质量或其他特性的需要，有些应用希望使用IPv6服务，这样就需要更广泛的更新。

10.1.2 传输层

大多数情况下，IP地址与应用层协议无关，但是与传输层协议却有很大关系。UDP和TCP的伪头都使用了源IP地址和目的IP地址，而且TCP电路是由源节点和目的节点的IP地址和端口号来定义的。如果要与IPv6互操作，至少要修改UDP和TCP，以适应128位IPv6地址。这意味着UDP和TCP需要识别IPv6地址，并能正确计算伪头。对于TCP，其实现还必须能够管理基于IPv6地址的电路。

在第一个IPv6 RFC发布之后，出现了一些顾虑，即需要TCPng来补充IPng。目前TCP在处理移动节点时有一点问题：确定TCP电路需要源节点和目的节点的IP地址。如果在TCP交互期间，一方或双方的IP地址有所改变，则电路的标识就会出现错误。移动节点从一个网络地址向另一个网络地址转换时就会出现这种情况，例如，火车或汽车上的节点使用无线网络接入，

或连接到网络的节点在夜间为获得更好的费率而改变 ISP 的情形。

这种问题的产生是由于 TCP 至少在目前还没有机制能允许在连接中改变 IP 地址。如果一个节点收到的 TCP 段中的源 IP 地址与此 TCP 电路在建立时协商的地址不同, 该节点将认为这个 TCP 段是属于另一个电路的。这意味着移动 IP 目前还不能支持激活的 TCP 电路从一个网络地址向另一个网络地址转换。

TCPng 的问题比简单地允许 TCP 连接支持网络地址转换要复杂许多。问题在于支持这样的地址转换将导致安全性漏洞: 攻击者很容易冒充从一个网络向另一个网络转换的节点, 如同授权的节点从一个网络向另一个网络转换一样。解决这样的问题将要求对 TCP 进行重大的升级, 即需要引入机制使节点在其 IP 地址改变时能向其他节点证明自己。

目前, 如果移动 IP 在 TCP 连接的中间切换网络, 它必须在切换之后重新协商连接。某种意义上, 对于支持移动主机的无缝互操作, TCPng 是很必要的。

10.1.3 链路层

与上面层相比, 诸如以太网和 ATM 之类的链路层协议由于 IPv6 的升级而受到的影响很小。这是由于这些协议只是将上层数据报封装到链路层帧中。但这并不说明 IPv6 对链路层协议毫无影响。例如, ATM 使用类似点到点电路来跨越网络传送数据, 对于需要将 IPv6 包交付多个节点的服务, ATM 需要格外注意。关于 ATM over IP 的详细信息参见 RFC 1680 (IPng 对 ATM 服务的支持) 和 RFC 1932 (IP over ATM: 一个框架文档)。也可参见 IP over ATM 工作组的 Internet 草案, 其中一些草案涉及 Ipv6 over ATM 地址。

可能受 IPv6 影响的链路层问题还包括路径 MTU 发现 (参见第 5 章) 及地址解析协议 (ARP), 这些协议需要修改以支持 128 位 IPv6 地址。

10.2 IPv6 域名系统扩展

Internet 应用程序能够很容易使用, DNS 是一个重要因素: 它使名字很方便地映射到 IP 地址。DNS 使用分级的名字空间, 每一级都有一些服务器帮助将名字映射为地址。主机名可能是诸如 “host.organization.com” 的形式, 表示主机 host 在域 organization.com 中。如果 organization.com 内的节点要查找 host, 就查询本地 DNS 服务器, 该服务器保持着 organization.com 中的主机的名字和地址信息, 它将简单地查找 host, 并以 host 对应的 32 位 IP 地址来回答节点的请求。

如果 organization.com 之外的节点需要 host.organization.com 的 IP 地址, 它将查询自己的本地 DNS 服务器, 这个本地服务器必须查询保持 .com 网络域信息的上一级服务器, 然后该上级服务器将请求导向 organization.com 域的 DNS 服务器, 由这个服务器最终响应, 将所请求的 IP 地址发送给查询者的本地 DNS 服务器, 再由该本地 DNS 服务器将信息传递给发出请求的节点。

到目前为止一切顺利。但是 DNS 最初被设计为用于处理 32 位 IPv4 地址。RFC 1886 (支持 IPv6 的 DNS 扩展) 描述了为使 DNS 支持 IPv6 而进行的必要的修改。此 RFC 篇幅很短, 它扼要陈述了为使 DNS 适用于 IPv6 而进行的三处修改:

- 建立新的资源记录类型 (称为 AAAA 记录类型), 以将名字映射为 128 位的 IPv6 地址。IPv4 资源记录使用 A 类记录类型。
- 建立新域, 即 .IP6.int, 用于增补 IPv6 主机地址以支持基于地址的查找, 即请求节点想了解 IPv6 地址对应的域名。IPv4 地址也有类似设施, 即 .in-addr.arpa。

- 必须修改现有的DNS查询，使之不仅能定位或处理 IPv4地址，同样也能处理 IPv4和IPv6地址共存的情况。

10.3 地址解析协议和邻居发现

IPv6不再执行地址解析协议 (ARP)或反向地址解析协议 (RARP)。在IPv4中，这些协议用于计算IP地址与本地链路网络地址的关联，换言之，以以太网为例，这些协议将节点的以太网 MAC地址链接到IP地址。这些协议的必要性在于，节点要计算出将 IP包使用链路层发往同一本地子网的哪一个节点。

ARP简单易行，它可在以太网和任一使用 48位MAC地址的网络媒体上执行，也可用于任意长度的MAC地址。在IPv6中没有继续使用ARP有如下原因：首先，ARP依赖于IPv6和使用组播的ICMPv6报文。这意味着，没有必要为使用ARP的每个不同类型网络都重新构造ARP，任一支持IPv6和组播的节点应该也支持邻居发现。对组播的支持很重要，在链路层更是如此。和广播一样，组播在诸如以太网之类的支持多路同时访问同一媒体的网络上很容易实现。但是，对于所谓的非广播多址接入 (NBMA)网络，例如ATM和帧中继，组播则很难处理。这些NBMA网络依赖于电路而非包，要求为将接收组播信息的每个节点都建立一条单独的电路，这导致组播更加复杂。但是只要有机制能提供组播功能，这些网络上的节点也能够支持邻居发现，而无需显式建立ARP之类的服务。

RFC 1970(IPv6的邻居发现)中描述了邻居发现机制，它提供了几种不同用途，包括下列方面的支持：

- 路由器发现。即帮助主机来识别本地路由器。
- 前缀发现。节点使用此机制来确定指明链路本地地址的地址前缀以及必须发送给路由器转发的地址前缀。
- 参数发现。此机制帮助节点确定诸如本地链路 MTU之类的信息。
- 地址自动配置。用于IPv6节点自动配置(见第11章)。
- 地址解析。替代了ARP和RARP，帮助节点从目的IP地址中确定本地节点(即邻居)的链路层地址。
- 下一跳确定。可用于确定包的下一个目的地，即，可确定包的目的地是否在本地链路上。如果在本地链路，下一跳即是目的地；否则，包需要选路，下一跳即是路由器，邻居发现可用于确定应使用的路由器。
- 邻居不可达检测。邻居发现可帮助节点确定邻居(目的节点或路由器)是否可达。
- 重复地址检测。邻居发现可用于帮助节点确定它想使用的地址在本地链路上是否已被占用。
- 重定向。有时节点选择的转发路由器对于待转发的包而言并非最佳。这种情况下，该转发路由器可以对节点进行重定向，以将包发送给更佳的路由器。例如，节点将发往Internet的包发送给为节点的内联网服务的默认路由器，该内联网路由器可以对节点进行重定向，以将包发送给连接在同一本地链路上的Internet路由器。

邻居发现通过定义特殊的ICMP报文类型来执行，这些报文包括：

- 路由器通告。要求路由器周期性地通告其可用性，以及用于配置的链路和Internet参数(见第11章)。这些通告包含对所使用的网络地址前缀、建议的逐跳极限值及本地MTU的

指示，也包括指明节点应使用的自动配置类型的标志。

- 路由器请求。主机可以请求本地路由器立即发送其路由器通告。路由器必须周期性发送这些通告，但是在收到路由器请求报文时，不必等待下一个预定传送时间到达，而应立即发出通告。
- 邻居通告。节点在收到邻居请求报文的请求或其链路层地址改变时，发出邻居通告报文。
- 邻居请求。节点发送邻居请求报文来请求邻居的链路层地址，以验证它先前所获得并保存在高速缓存中的邻居链路层地址的可达性，或者验证它自己的地址在本地链路上是唯一的(见第11章)。
- 重定向。路由器发送重定向报文以通知主机，对于特定目的地自己不是最佳路由器。

路由器通过组播来发送其路由器通告报文，这样同一链路上的节点可以构造自己的可用默认路由器列表。

邻居发现也可以用于实现其他目标，包括：

- 链路层地址变化。对同一网络，节点可以有多个接口，如果节点得知自己的链路层地址改变，就可以通过发送几个组播包来将其地址改变通知其他节点。
- 入境负载均衡。应注意，接受大量业务流的节点可能有多个网络接口，使用邻居发现，所有这些接口都可以用一个 IP 地址来代表。通过让路由器在发送其路由器通告包时省略源链路层地址，可以实现路由器负载均衡。此时，查找该路由器的节点每次想要发送包给该路由器时，都必须执行邻居发现，而该路由器就可以选择接受包的链路层接口来响应此节点。
- 任意点播地址。正如第 6 章所述，任意点播地址表示单播地址的集合，发送给该任意点播地址的包将交付给这些地址中的任一个。通常任意点播地址用于标识提供同样服务的节点集，即，将包发送给一个任意点播地址的节点并不在意由节点集中的哪一个来响应。因为任意点播地址的多个成员都可能响应对其链路层地址的请求，邻居发现机制要求节点应预计到可能收到多个响应，并能正确地处理。
- 代理通告。如果一个节点不能正确响应邻居发现请求，邻居发现机制允许用另一个节点来代表该节点。例如，一个代理服务器可以代表移动 IP 节点(见第 11 章)。

第11章 自动配置和移动IP

首先考虑全球分布最广泛的网络——电话系统的情形。如果每次买一个新电话，都要进行配置，以便能够使用指派到家中的电话号码，那么使用这样的网络将很不方便；如果每次把一个分机从房间移动到另一个房间时，都要重新进行配置，那么使用这样的网络也将非常不方便。很多情况下，IPv4网络也要求如上所述的人工配置：安装一台新计算机或其他连接设备都要求有人来手工配置地址和其他网络信息。过去十年内情况有所改进，但 IP配置仍然是个问题。

IPng所陈述的最重要目标之一是支持“即插即用”——不需要任何人工干预，就有可能将一个节点插入IPv6网络并在网络中启动。此目标与计算工业的趋势一致：1983年，个人计算机的买主必须安装操作系统及所有需要的应用软件；1998年，大多数新系统都预先安装了操作系统及应用软件。新系统在网络就绪方面的变化也显而易见。1983年，如果要连接到网络，用户必须购买并安装网卡或MODEM，然后对特定的应用进行配置；1998年，大多数系统都包括了已配置好、可以马上使用的内置MODEM及(或)网卡。

要使网络计算像使用电话一样方便，还需要改进用户友好性。在以太网网卡上设置硬件开关对工程师和科学家们可能很容易，但机械师、医生、旅行社以及其他非技术专业人士却不能忍受为使其系统运行或连接到Internet而浪费几小时或几天。

11.1 IPv6的即插即用

IPv6使用两种不同机制来支持即插即用网络连接。第一种机制的示例是启动协议(BOOTP)，后来又设计了动态主机配置协议(DHCP)，允许IP节点从特殊的BOOTP服务器或DHCP服务器获取配置信息。但是这些协议支持所谓的“状态自动配置”，即服务器必须保持每个节点的状态信息，并管理这些保存的信息。不论对于为许多个人用户服务的ISP，还是雇员经常在各部门间流动的大型机构，DHCP都是IPv4网络配置的重要工具。

11.1.1 状态自动配置与无状态自动配置

DHCP的问题在于，作为状态自动配置协议，它要求安装和管理DHCP服务器，并要求接受DHCP服务的每个新节点都必须在服务器上进行配置。很简单，DHCP服务器保存着它要提供配置信息的节点列表，如果节点不在列表中，该节点就无法获得IP地址。DHCP服务器还保持着使用该服务器的节点的状态，因为该服务器必须了解每个IP地址使用的时间，以及何时IP地址可以进行重新分配。

状态自动配置的问题在于，用户必须保持和管理特殊的自动配置服务器以便管理所有“状态”，即所容许的连接及当前连接的相关信息。对于有足够资源来建立和保持配置服务器的机构，该系统可以接受；但是对于没有这些资源的小型机构，工作情形较差。至少对于大多数个人或小型机构，无状态自动配置是较好或较容易的解决方案。这种机制允许个人节点能够确定自己的IP配置，而不必向服务器显式请求各节点的信息。

实际上,至少在理论上且进行了某些假定的情况下,无状态自动配置规程相对容易实现。首先,如果使用IEEE EUI-64链路层地址(见第6章),用户就可以确信自己的主机ID是唯一的。因此,节点要完成的工作是确定自己的链路层地址并计算出EUI-64地址,然后确定自己的IPv6网络地址。向最近的路由器询问是确定网络地址的一种方法,这就是IPv6中无状态自动配置的实现方式(见下一节)。

最后,根据IPv6中的定义,状态自动配置和无状态自动配置可以共存并可一起操作。后续章节中将涉及正在制订中的DHCP的更新版本,称为DHCPv6。两种类型自动配置方法的合作比单独使用其中一种更易于实现互联网络连接的即插即用。例如,使用无状态自动配置,节点可以很快确定自己的IP地址,而且一旦获得此信息,它就可以与DHCP服务器交互以获得所要求的其他网络配置值。实际上,DHCPv6很可能要依靠IPv6无状态自动配置来简化某些情况下的状态配置。

如果使用无状态自动配置要简单很多,那么为什么还要使用状态自动配置呢?此问题的答案取决于构造网络的机构的要求。无状态自动配置对得到IP地址的节点提供最低程度的监视。任一节点可以连接到链路,通过路由器向能实现无状态自动配置的节点发出的通告来获知网络和子网信息,并构造有效的链路地址。但是,如果有DHCP服务器的支持,机构可以更紧密地控制网络可配置的节点。只有由网络管理员明确授权的节点才能通过DHCP服务器来配置。

11.1.2 IPv6无状态自动配置

RFC 1971(IPv6无状态地址自动配置)中描述了IPv6的无状态自动配置。该RFC还在更新,大多数修改是对原规范的澄清或细化,例如对潜在的路由器否认服务攻击的处理方法等。无状态自动配置过程要求节点采用如下步骤:首先,进行自动配置的节点必须确定自己的链路本地地址(如IEEE EUI-64地址);然后,必须验证该链路本地地址在链路上的唯一性;最后,节点必须确定需要配置的信息。该信息可能是节点的IP地址,或者是其他配置信息,或者两者皆有。如果需要IP地址,节点必须确定是使用无状态自动配置过程还是使用状态自动配置过程来获得。

无状态自动配置要求本地链路支持组播,而且网络接口能够发送和接收组播。完成自动配置的节点首先将其链路本地地址(如IEEE EUI-64地址)追加到链路本地前缀(在第6章中讨论,见图6-1)之后。这样,节点就可以开始工作:它可以使用IPv6与同一网络链路上的其他节点通信,只要同一链路没有其他节点使用与之相同的EUI-64地址,该节点的IPv6地址就是可用的。

但是,在使用该地址之前,节点必须先证实起始地址在本地链路是唯一的,即,节点必须确定同一链路上没有其他节点使用与之相同的EUI-64地址。大多数情况下不会出现这个问题,大多数使用网络接口卡(如以太网适配器或令牌环适配器)的节点都有唯一的48位MAC地址;而对于通过点到点链路连接的节点,链路上只有一个端节点。但是,其他网络媒体可能没有唯一的MAC地址,某些网络接口卡也可能错误地使用了它们无权使用的MAC地址。此时,节点必须向它打算使用的链路本地地址发送邻居请求报文。如果得到响应,试图自动配置的节点就得知该地址已为其他节点所使用,它必须以其他方式来配置。

如果没有路由器为网络上的节点服务,即,如果本地网络孤立于其他网络,则节点必须寻找配置服务器来完成其配置;否则,节点必须侦听路由器通告报文。这些报文周期性地发

往所有主机的组播地址(见第6章),以指明诸如网络地址和子网地址等配置信息。节点可以等待路由器的通告,也可以通过发送组播请求给所有路由器的组播地址来请求路由器发送通告。一旦收到路由器的响应,节点就可以使用响应的信息来完成自动配置。

11.1.3 BOOTP和DHCP

1985年,BOOTP首先在RFC 951(自举协议)中描述,该协议的最初目的是允许工作站向本地服务器询问他们自己的IP地址、某服务器主机的地址以及自举执行文件的名称。对于某些应用,如无盘工作站从网络服务器上装入全部软件的情况,BOOTP的功能足够了;但是对于很多其他应用,如将个人机连接到IPv4网络,BOOTP的功能不足。其问题在于除BOOTP提供的信息之外,PC的TCP/IP网络软件需要更多的信息,如主机名字、域名、子网掩码及DNS服务器地址等。

到1993年,描述DHCP的RFC 1531(动态主机配置协议)得以发布。以BOOTP的报文结构为基础,DHCP增加了一些机制,用于传送将主机连接到IPv4所需要的全部IP配置信息。DHCP的功能与IPv4网络地址相对不足有关。例如,得到一个C类网络地址的机构最多有254个地址来分配给用户。对于某些应用,这些地址很充足,例如一个花店中,连网的计算机不太可能超过几十台。但是,对于有数百名雇员的企业,尤其如果连接到网络的计算机超过250台时,就可能产生地址分配的问题。

DHCP增加了在有限时期内向节点分配地址的能力。这意味着,对于较大型网络,如果同时连接的节点数不超过254,C类网络是足够的。ISP很愿意使用DHCP,因为这样他们所服务的用户数不再受限于他们控制的地址数,只要在任一时刻只有一部分用户连接到该IP,其他用户就不会浪费额外的未使用的IP地址。

11.1.4 DHCPv6

当然,与处理IP地址的其他协议相同,DHCP必须升级以支持IPv6地址。DHCPv6正在制订中。很显然,它不仅是为支持更长地址而进行的表面更新,由于IPv6中增加了无状态自动配置,对于DHCPv6,使用这个新能力将很有好处。

使用无状态自动配置,节点至少自动拥有了本地连接能力,DHCP也不再是提供一些其他基本配置参数所必需的方法。默认路由器不是配置的一部分,因为通过侦听路由器通告,任何节点都可以自行确定自己的默认路由器(见第10章)。

新的DHCP能支持各种新特性,例如:

- 配置动态更新DNS的能力,可以反映网络当前状态。
- 地址非难,即地址分配即将失效的状态,该机制可用于对网络进行动态重新编号(见下节)。

11.2 移动网络技术

一直到近期,几乎所有的网络设备都是在原地静止的。计算机,甚至个人计算机,都很大且不经常移动。近几年来,不仅是笔记本计算机,而且包括手提计算机、个人数字助手(PDA)都显著增加,甚至蜂窝电话和寻呼机也都可以支持IP。目前的问题在于,不论设备平常是通过有线媒体或无线媒体连接到网络,当设备移动时,如果不论移动设备实际上在何处,

其他设备都能够以同一个IP地址来访问该设备，这将是方便的。

要实现这一点却非常困难，因为节点移动时，可能必须连接到使用不同IP地址的不同网络。移动IP在RFC 2002(IP移动性支持)中描述。目前，此RFC还在进行修改和更新以支持IPv6。任何情况下，移动IP都应支持节点从一个网络向另一个网络移动，即“宏观移动性”，而不仅仅是支持“微观移动性”，例如像蜂窝电话一样，从一个蜂窝向另一个蜂窝切换无线连接。

11.2.1 IPv4中的移动IP

正如RFC 2002中所述，移动IP使用移动代理的概念。为移动主机指派一个一直可达的主地址。当主机位于正常驻地时，它使用自己的主地址连接到本地网络，所有的协议都按正常方式操作；而移动代理通常是常规路由器，它作为外地代理，在移动主机离开其驻地网络时像一种邮件领取部一样使用。移动代理也可以作为主代理，处理传送给移动主机的信息。

当移动节点离开驻地时，可以按照下列方法使用移动IP(如IPv4所述)来连接到网络：

(1) 外地代理和主代理周期性地发出报文，表明它们的可用性。移动主机也可以主动请求此信息。这些通告以ICMP路由器通告为基础，为移动节点提供足够的信息，使其能够确定它是在自己的驻地网络还是在外地网络中。

(2) 如果移动节点确定自己目前连接到驻地网络，就如同非移动主机一样工作。

(3) 但是，如果该节点确定自己是在外地网络中，则它将从外地网络获得“关照地址”。该地址是当移动主机在外地网络中时，可到达移动主机的临时地址。移动主机可以使用外部机制(如DHCP)来获得在外地网络上的有效地址，或者它也可以使用移动代理指定的某个地址，该地址就称为外地代理关照地址。此时，对于所服务的任何移动节点，移动代理使用同一个境内地址，并将进入网络的包转发给正确的节点。

(4) 一旦移动主机拥有可在外地网络上寻址的某类地址，通过发送报文，它将该地址注册到其主代理，实际报文的内容类似“如果你收到发给我的主地址的包，请转发到这个地址”。

(5) 这样，一旦主代理知道对于发给移动节点的包应向何处转发，它就把这些包拦截下来，并进行封装，以IP隧道方式发送到移动节点提供的关照地址。如果该关照地址是一个配置的关照地址，则由外地代理来接收封装的IP包，拆包并转发给移动节点；如果该关照地址是在外地网络上分配给移动节点的单独IP地址，移动节点就可以接收到带封装的IP包，自己进行拆包。

如果外地网络上的移动节点要发送包，则无需进行特殊操作，这些主机将继续使用其主地址为包的源地址，对这些包也无需进行任何特殊处理。

11.2.2 IPv6中的移动IP

相对而言，移动IPv6将更易于实现和使用。首先，在IPv6中，在无状态自动配置或使用DHCPv6的状态自动配置的支持下，获得关照地址的过程更加简单。正因如此，IPv6中没有外地代理关照地址，而只有配置的关照地址。其次，应该有可能使用IPv6的各种特性来改进移动节点的操作。例如，主代理可以使用邻居发现的代理通告来截获发给移动节点的IPv6包。对于通过目的地选项来将地址更新与地址相捆绑的路由优化，节点也应该有基本的支持。

移动IPv6中包含的另一个新特性是：即使在移动节点的常规主代理不可达的情况下，移动节点也有能力和驻地网络建立联系。移动节点可以向驻地网络中为主代理保留的地址发送任意点播包，结果任何可用的主代理将把自己的选项通知移动节点。

第三部分 IP过渡和应用

第12章 IP过渡策略

一旦IPv6投入应用，看起来网络中的所有的主机都必须升级。对于要处理包含成千上万个主机的全球公司网络的网络管理者而言，这种挑战很令人沮丧。但是，实际情况并非如此，研究向IPv6过渡的人士正在致力于IPv6的设计及IPv6所支持的协议和机制，以实现得体的渐进的升级。如果能有条理地、明智地进行现有网络向IPv6的升级，升级的影响可能较小。本章将讨论目前已提出的平滑过渡策略。

在RFC 1933(主机和路由器向IPv6过渡的机制)、RFC 2185(向IPv6过渡的选路问题)、RFC 2071(网络重新编号概观：为何需要及需要什么)以及RFC 2072(路由器重新编号指导)等文档中都涉及有关向IPv6过渡的讨论。还有一些有关向IPv6过渡和商业环境中向IPv6升级的Internet草案正在制订中。

向IPv6过渡必定是渐进的。考虑到目前已经有大量网络和节点连接到Internet，人们无法接受大量的切换形式的升级。这种升级要求网络管理员为Internet上的每个主机和路由器都找到并安装新版本的网络软件，考虑到目前有很多不同的平台运行着IPv4，这种做法将很难实现。

与此相似，随着网络厂商和开发者逐渐将IPv6引入不同的平台，随着网络管理者逐渐确定自己所需要的IPv6功能，向IPv6过渡也将是一个相对缓慢的过程。预计IPv4和IPv6将长期共存，也许将永远共存。大多数过渡策略都依靠协议隧道的两路方法，即至少在最初，将来自IPv6岛的IPv6包封装在IPv4包中，然后在广泛分布的IPv4海洋中传送。经过过渡的早期阶段，越来越多的IP网络和设备将支持IPv6。但即使在过渡的后期阶段，IPv6封装仍将提供跨越只支持IPv4的骨干网和其他坚持使用IPv4的网络的连接能力。另一路策略是双栈方法，即主机和路由器在同一网络接口上运行IPv4栈和IPv6栈。这样，双栈节点既可以接受和发送IPv4包，也可以接受和发送IPv6包，因而两个协议可以在同一网络中共存。

12.1 IPv6协议隧道方法

见图12-1，隧道方法用于连接处于IPv4海洋中的各孤立的IPv6岛。此方法要求隧道两端的IPv6节点都是双栈节点(见下节)，即也能够发送IPv4包。将IPv6封装在IPv4中的过程与其他协议封装相似：隧道一端的节点把IPv6数据报作为要发送给隧道另一端节点的IPv4包中的净荷数据，这样就产生了包含IPv6数据报的IPv4数据报流。在图12-1中，节点A和节点B都是只支持IPv6的节点。如果节点A要向B发送包，A只是简单地把IPv6头的目的地址设为B的IPv6地址，然后传递给路由器X；X对IPv6包进行封装，然后将IPv4头的目的地址设为路由器Y的IPv4地址；若路由器Y收到此IPv4包，则首先拆包，如果发现被封装的IPv6包是发给节点B的，Y就将此包正确地转发给B。

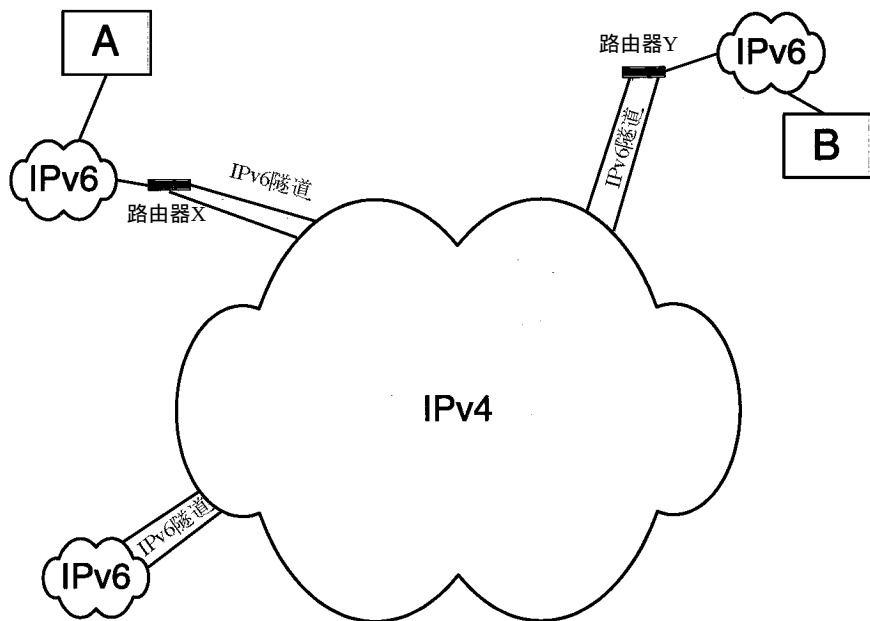


图12-1 通过在双栈IPv4/IPv6路由器之间使用隧道连接，
IPv6网络孤岛的链接可以跨越IPv4海洋

12.1.1 与IPv4兼容的IPv6地址

第6章介绍了包含IPv4地址的IPv6地址。这些地址有两类：IPv4兼容地址和IPv4映射地址。IPv4兼容地址是指在128位地址中，高阶的96位全部为0，而最后的32位包含IPv4地址(见图6-4)。能够自动将IPv6包以隧道方式在IPv4网络中传送的IPv4/IPv6节点将使用这些地址。

双栈节点则对于IPv4包和IPv6包都使用相同的地址。只支持IPv4的节点向双栈节点发送包时，使用双栈节点的IPv4地址；而只支持IPv6的节点则使用双栈节点的IPv6地址，即将原IPv4地址填充0后成为128位。总之，这类节点可以作为路由器链接IPv6网络，采用自动隧道方式穿越IPv4网络。该路由器从本地IPv6网络接收IPv6包，将这些包封装在IPv4包中，然后使用IPv4兼容地址发送给IPv4网络另一端的另一个双栈路由器。如此继续，封装的包将通过IPv4网络群转发，直至到达隧道另一端的双栈路由器，由该路由器对IPv4包拆包，释放出IPv6包并转发给本地的IPv6主机。

12.1.2 配置隧道和自动隧道

配置隧道和自动隧道的主要区别在于：只有执行隧道功能的节点的IPv6地址是IPv4兼容地址时，自动隧道才是可行的。在为执行隧道功能的节点建立IP地址时，自动隧道方法无需进行配置；而配置隧道方法则要求隧道末端节点使用其他机制来获得其IPv4地址，例如采用DHCP、人工配置或其他IPv4的配置机制。

12.1.3 IPv6隧道类型

可以作为隧道端点的节点有几种不同的组合类型，图12-2描述了这些不同隧道的操作情

形。图中的互联网络由三个网络、两个路由器和两台主机组成，它使用了如下几种不同的隧道类型。但是，为了区别这些不同类型的隧道，根据所演示的隧道类型，图中的实体可能是只支持IPv4、只支持IPv6或者IPv4/IPv6双栈。

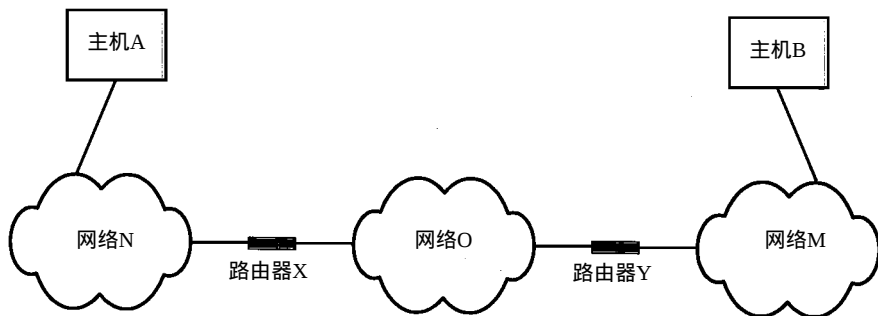


图12-2 IPv6隧道的不同类型

不同的隧道类型包括：

- 路由器-路由器隧道。路由器 X和路由器 Y使用隧道方式来传送经过网络 O的包，而网络 O只支持IPv4。主机A可以透明地将IPv6包发送给主机B，这两个主机都不必考虑中间插入的IPv4网络(即网络O)。这种情况下，主机A和主机B都是只支持IPv6的节点。
- 路由器-主机隧道。此时网络 M只支持IPv4，但主机B同时运行IPv4和IPv6，网络的其他部分都只支持IPv6。这种情况下，隧道传送发生在路由器 Y和主机B之间。在网络的其他部分，IPv6包可以自由传送。但是路由器 Y必须将IPv6包封装在IPv4包中，以便通过只支持IPv4的网络M。
- 主机-主机隧道。假设此时只有主机 A和主机B同时支持IPv4和IPv6，而网络的其他部分都只支持IPv4。这种情况下，隧道传送发生在主机 A和主机B之间。对于发往主机 B的IPv6包，主机A必须把它们封装在IPv4包中，以便由只支持IPv4的路由器来运载。
- 主机-路由器隧道。假设此时主机 A和路由器X为双栈节点，网络 N只支持IPv4，而网络的其他部分都只支持IPv6。这种情况下，主机 A仅对发往路由器 X的IPv6包采用隧道方式；一旦通过了只支持IPv4的网络N，路由器X就对这些通过隧道传送的包拆包，然后按正常方式通过IPv6网络转发。

12.2 IPv4/IPv6双栈方法

正如2000问题的幽灵所表现出来的，传统系统的坚固性被高估了。很长时间内，IPv4仍将存在，即使一些网络或连网世界的其余部分已升级为IPv6。到那时，升级系统将需要保持与IPv4系统的互操作能力。随着时间的推移，互操作的负担将由早期的实现者承担转为由传统系统的维护者来承担。任何情况下，同时支持IPv4和IPv6的系统都是必要的。

双栈节点并不是一个新概念。例如，许多公司主机既支持到Internet的连接能力，也支持连接到使用早期版本的Novell Netware(在Netware 5中，IP已代替IPX作为纯网络层协议)的公司LAN。这些主机已经支持两种根本不同的网络栈。到Internet的连接能力通过TCP/IP协议栈来提供，而到Netware的连接能力则通过IPX栈来提供。链路层接收到数据段并拆开，段头指明数据报是发给TCP/IP栈还是发给IPX栈，然后将该包传递给正确的栈处理。

双栈节点

IPv4/IPv6双栈节点与其他类型的多栈节点的工作方式相同。链路层接收到数据段，拆开并检查包头。如果 IPv4/IPv6 头中的第一个字段，即 IP 包的版本号是 4，该包就由 IPv4 栈来处理；如果版本号是 6，则由 IPv6 栈处理。

最简单的双栈工作是只支持 IPv4 和 IPv6，但不支持隧道方式。对于大多数节点，尤其是如果这些节点的 Internet 应用软件都已升级为同时支持 IPv4 和 IPv6，这种功能足够。因此，如同用于访问 IPv4 网络服务一样，同一应用也能够用于访问本地 IPv6 网络服务。节点可以与任何 IPv4 节点或 IPv6 节点互操作，但只限于与其有连接能力的网络。在图 12-3 的示例中，可以与双栈节点 D 互操作的节点包括：网络 A 和网络 B 中的 IPv4 节点或 IPv6 节点、网络 M 中的所有 IPv4 节点，但 D 不能和网络 C 中的节点互操作。网络 C 是严格的 IPv6 网络，从网络 A 到网络 C 没有 IPv6 路径。链接网络 A 和网络 M 的路由器只支持 IPv4，因此无法通过网络 M 向网络 C 转发 IPv6 包。

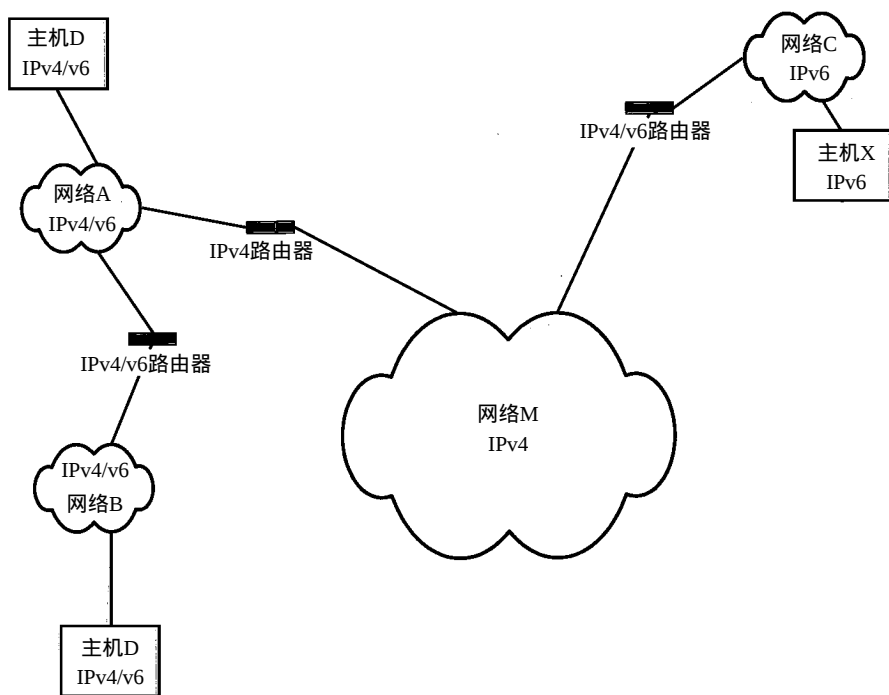


图12-3 根据是否能在IPv4网络中按隧道方式转发IPv6包，双栈节点、路由器和网络提供不同程度的互操作性

支持隧道方式的双栈节点增加了在 IPv4 网络上进行互操作的能力，而无需额外的 IPv6 路由器。在 IPv4 网络上以隧道方式传送 IPv6 包使图 12-3 中的示例得以改变。例如，如果节点 D 能在 IPv4 上以隧道方式传送 IPv6 包，则它可使用本地 IPv4 路由器将包转发给网络 C。如果节点同时支持自动隧道，则可实现无缝操作；否则需要某些链接配置。

12.3 IPv6 地址分配

如果用户需要 IPv4 网络地址，通常必须和 ISP 协商方案，ISP 将按照 CIDR 类型地址集聚来

分配地址块。IPv4网络地址最终由Internet分配号码授权机构(IANA)来控制。但是,如果用户需要IPv6地址,事情就不是这样简单。正如在RFC 1881(IPv6地址分配管理)中的定义,IANA将IPv6地址空间块指派给区域或其他类型的登记机构,这些机构再将较小块地址空间分配给网络供应商或其他子机构,然后子机构依次将地址分配给请求IPv6地址的商业公司、机构或个人。

但是,直到1998年秋,这些分配还尚未开始实施,而IPv6地址的精确格式也尚未确定。如果用户需要正式分配的全球唯一的IPv6网络地址,就必须等到IANA开始分配地址空间的时候。同时,用户可以选择孤立于其他网络来运行自己的IPv6网络,使用自己分配的内部网络地址。由于IPv6支持无状态自动配置,对错误编号的IPv6网络重新进行编号的负担远没有对错误编号的IPv4网络重新编号那么繁重,但是也需要付出代价来重新配置路由器,因而不可避免地会遇到麻烦。

目前已产生了一个简单的替代方法。下一节将讨论的6BONE是一个全球的IPv6网络,用于IPv6产品的测试和预生产。

12.4 6BONE

1996年,在第一个IPv6标准为IETF接受并作为RFC发布之后不久,就产生了6BONE网络。大型互联网需要有骨干结构来链接完全不同且广泛分布的网络,而作为现有IPv4基础构架之上的虚拟结构,6BONE使用隧道方式来链接参与的网络。

到1998年9月,6BONE在35个国家中已有参与者,连接了至少200个站点。6BONE打算在IPv6产品实现广泛的商业推广之前,用于测试或获取IPv6的经验。

为将站点连接到6BONE,需要如下设备:

- 支持IPv6并连接到6BONE的路由器。
- 支持IPv6以建立用户自己的IPv6网络的工作站。
- 连接到6BONE的途径,这可解释为找到已连接到6BONE的其他机构,并建立到这些机构的站点的连接;
- 支持IPv6 AAAA记录的DNS服务器。

第13章介绍了一些已经实现IPv6的厂商,但是连接到6BONE的最佳方法是从访问6BONE站点:<http://www.6bone.net>开始。

通过向majordomo@isi.edu发送报文,用户可以订阅6BONE邮件列表,该报文的全部内容 is: subscribe 6bone。

第13章 IPv6解决方案

尽管进展缓慢，但 IPv6 产品正逐渐向市场发展。本章重点介绍现有及即将面市的支持 IPv6 的产品，并解释这些产品的使用方法及其对网络专业人员的用途。

13.1 需要支持IPv6的产品

连接到 IPv6 网络的所有设备必须支持 IPv6，即作为 IPv6 系统使用的 IPv4 必须升级。简单说，这些设备分为两大类：网络主机和网络路由器。网络主机可以是个人计算机或大型计算机系统。不论是哪类设备，至少 IP 栈要升级到 IPv6。

但是，要注意 IPv4 数据可以使用协议隧道技术来通过 IPv6 网络，此时，如果用户不要求 IPv6 的特性，可以暂时不对主机升级。但路由器必须升级到 IPv6，尤其是将 IPv4 网络连接到 IPv6 网络的路由器必须实现 IPv6。

13.2 正在开发IPv6产品的公司

自从第一批 IPv6 RFC 发布以来，我就经常非正式地向网络厂商询问他们关于 IPv6 的产品计划。大多数情况下，我得到的回答是：“我们正在进行这方面的工作，但是还没有为市场做好准备”。这种回答已经足以表明，在实验网络和研究实验室以外，只有极少数的场合可以使用 IPv6。

正规的主机 IPv6 产品，尤其是对个人计算机，还是有一些产品可以选择。FTP 软件公司的 Secure Client(安全客户)3.0 于 1997 年首次面市，是支持 IPv6 的完整 TCP/IP 协议栈产品。大多数厂商仍然只提供 IPv6 的测试版或研究版。例如，微软公司只有支持 NT 平台的实验 IPv6 栈。Sun、伯克利软件设计公司(BSDI)、DEC 及其他公司都正在研制支持 IPv6 的产品。IBM 公司为其 AIX 4.2 操作系统提供支持 IPv6 的原型版。Linux 操作系统的 IPv6 版本也已经出现，它以美国海军研究实验室(NRL)的实现为基础。

路由器厂商的进展稍好。例如，Bay 公司的 BayRS(Bay 选路服务)12.0 版选路产品已支持 IPv6。日立公司早在 1997 年 6 月就已开始销售 IPv6 路由器，而 DEC 在 1997 年 3 月发布了其 IPv6 AlphaServer 产品。另一方面，有些公司目前还不能提供 IPv6 产品。3Com 公司宣称他们已准备好向 IPv6 升级。Cisco 公司也正向 IPv6 项目投入相当大的研究力量，但是好像还不能将 IPv6 支持引入目前的生产线。

表 13-1 列出了上述各公司及其他公司 IPv6 产品的相关网络站点，感兴趣的读者请访问相关网络厂商站点以了解更新信息。

Sun 公司提供了一个出色的站点，介绍支持 IPv6 产品的最新信息，其 URL 为：<http://playground.sun.com/pub/ipng/html/ipng-implementations.html>。

表 13-1 IPv6 厂商信息的网页

公司/机构	网 页	描 述
3Com 公司	http://www.3com.com/nsc/ipv6.html	3Com 一直活跃于 IPv6 研究和开发领域

(续)

公司/机构	网 页	描 述
Bay公司	http://www.baynetworks.com	Bay公司的站点提供了相当多的 IP 信息，既包括一般信息，也包括 Bay 的生产线信息。1998 年中期后，要通过北方电信的站点来访问 Bay
伯克利软件设计公司	http://www.bsdi.com	计划在 1998 年支持 IPv6 升级到 BSDI Internet 服务器
Cisco 公司	http://www.cisco.com/warp/public/732/ipv6/index.html	Cisco 是另一个活跃于 IPv6 研究和开发的路由器厂商
DEC	http://www.digital.com/info/ipv6	基于 64 位 alpha 的服务器软件
FTP 软件公司	http://www.ftp.com	另一个 TCP/IP 产品厂商 NetManage 宣布，计划于 1998 年 6 月收购 FTP 软件公司
日立公司	http://www.hitachi.co.jp/Prod/comp/net-work/nr60e.htm	此页提供了日立公司的 IPv6 路由器的信息
IBM 公司	http://www.ibm.com	AIX 4.3 现在支持 IPv6
LinuxFAQ	http://www.bieringer.de/linux/IPv6/default.html	此 URL 指向频繁更新的 Linux/IPv6 FAQ
Mentat 公司	http://www.mentat.com/ipv6.html	Mentat TCP 支持 IPv6，为苹果、惠普、摩托罗拉、Stratus 和其他硬件厂商所许可
Novell 公司	http://www.novell.com	NetWare 5.0 允许以最小冲击方式向 IPv6 升级
Process 软件公司	http://www.process.com/ipv6/	Process 公司正在为 VMS 操作系统开发 IPv6 解决方案
Sun 公司	http://playground.sun.com/pub/solaris2-ipv6/html/solaris2-ipv6.html	Sun 公司是领先的 IPv6 厂商之一，其主页提供了 Solaris IPv6 原型，也提供了其他 IPv6 信息页面(见上文)

13.3 对IPv6的期待

1996 年，IPv6 仿佛马上就要面市。对 IPv4 地址耗尽的预测看似前景黯淡，厂商仿佛也热情很高，他们想象这将是实现越区销售、升级现有客户并销售新产品的机遇。但是到 1998 年中期，情况发生了变化。许多乐观的厂商对 IPv6 产品可用性的预测没有成为现实，而只是尴尬地保留在这些厂商的网页上。对于 IPv6，用户依然既不了解也不关注。

至于网络管理者，当然，他们了解 IPv4 的问题，但是与大多数 IT 专业人士所面临的 2000 年问题相比，IP 升级问题显得无足轻重。尤其是当前有很多问题需要解决的时候，而且这些问题更加紧迫，例如 2000 年兼容性问题不容许拖延，相比之下在未来几年之内 IPv4 向 IPv6 升级问题的优先级较低。

可以设想，如果得知一颗相当于法国国土大小的流星将在 1999 年 12 月 31 日午夜撞击地球，对于解决可能在未来 5 或 10 年内某一时刻发生的交通信号控制问题，谁还会关心呢？尤其是前者将使数亿人丧生，而后者只是使几百万人感觉不便。

虽然无法从 2000 年问题所需要的资金中分一杯羹，IPv6 将来还是很可能投入应用。尽管在北美很容易获得 IPv4 地址，但其他地区的网络管理者们都在为 IP 地址的严重短缺而苦恼。

熟悉IPv6的北美网络专业人士还不急于支持 IPv6，许多人还寄希望有更好的技术，或至少是不同的技术来代替 IPv4。随着自动配置的改进和用户需求的增加，IPv6很可能进入应用，但是在IPv6得以大规模应用之前，还有很多工作要完成。加强对 IPv4 升级问题的理解将有助于促进未来十年内IPv6的发展。

附录A 与IPv6有关的RFC索引

Internet RFC(Request for Comment)已成为传统的传播消息的媒介,它不仅用来发布Internet标准,同时还发布任何与Internet有关的文件资料及其协议。按数字大小排列(也与日期先后相对应)的本附录列表是一个RFC的选集,直接或间接地讨论了与从IPv4向IPv6迁移相关的议题。列表反映了到1998年9月为止发布的所有与IPv6有关的RFC。若读者想得到最新发布的RFC,请检索在线文件库。附录B给出了几个最重要的与IPng有关的RFC。

编 号	标 题
1029	对于以太网型的多局域网系统中的地址解析更有效的故障容错方法
1287	未来的Internet体系结构
1338	超网:一种地址分配和集聚策略
1366	IP地址空间管理指南
1367	IP地址空间管理指南的日程计划
1375	对新的IP地址类别的建议
1454	下一版本IP提案的比较
1466	IP地址空间管理指南
1467	CIDR在Internet中使用的状况
1519	无类域间选路(CIDR):一种地址分配和集聚策略
1550	征求下一代IP(IPng)白皮书
1560	多协议Internet
1563	文本/丰富的MIME内容类型
1629	Internet中OSI NSAP的分配指南
1667	IPng建模和仿真需求
1668	IPng统一的选路需求
1669	IPng标准的市场生命力
1670	对IPng工程考虑的输入
1671	向IPng过渡和其他考虑的白皮书
1673	电力研究机构评论IPng
1674	蜂窝移动电话工业界如何看待IPng
1675	对IPng安全性的关注
1680	IPng对ATM服务的支持
1683	IPng中的多协议互操作性
1686	IPng需求:有线电视工业界观点
1687	一个大公司用户对IPng的观点
1688	IPng移动性的考虑
1700	指派的号码
1702	在IPv4网络上通用选路封装
1705	向最终目标迈进:IPng的问题
1707	CATNIP: Internet公共体系结构
1715	地址分配效率比例系数H
1719	为IPng定方向
1726	选择下一代IP(IPng)的技术准则
1744	Internet地址空间管理的观察报告
1752	对IP下一代协议的建议

(续)

编 号	标 题
1768	对CLNP组播的主机组扩展
1770	对由发送者指引的多目的地分发的IPv4选项
1771	边界网关协议4 (BGP-4)
1809	IPv6中流标记字段的用法
1810	MD5性能报告
1814	地址唯一性令人满意
1825	IP的安全性体系结构
1826	IP身份验证头
1827	IP封装安全性净荷(ESP)
1860	IPv4可变长度子网表
1878	IPv4可变长度子网表
1881	IPv6地址分配管理
1883	IPv6技术规范
1884	IPv6寻址体系结构
1885	用于IPv6的 Internet 控制报文协议(ICMPv6)的技术规范
1886	支持IPv6的DNS扩展
1887	一种IPv6单播地址分配的体系结构
1888	OSI NSAP 和 IPv6
1897	IPv6测试地址分配
1917	呼吁Internet社区向IANA返回未用的IP网络(前缀)
1924	IPv6地址的一种紧凑的表示方法
1933	IPv6主机和路由器的过渡机制
1953	用于IPv4 1.0的Ipsilon流管理协议技术规范
1954	加标记的IPv4流在ATM Data Link Ipsilon 1.0上的传输
1955	IPNG中为Internet选路和寻址的新方案(ENCAPS)
1970	IPv6的邻居发现
1971	IPv6无状态地址自动配置
1972	在以太网上传输IPv6包的一种方法
1981	IPv6的路径MTU发现
2002	IP移动性的支持
2019	在FDDI网络上传输IPv6包的一种方法
2022	ATM网络UNI 3.0/3.1 对组播的支持
2023	IPv6 over PPP
2030	用于IPv4、IPv6和OSI的简单网络时间协议(SNTP)第4版
2036	Internet中A类地址空间部分使用的观察报告
2050	Internet注册IP分配指南
2071	网络重新编号概要：为何需要及需要什么？
2072	路由器重新编号指导
2073	基于供应商的IPv6单播地址格式
2080	IPv6用的RIPng
2081	RIPng协议适用性陈述
2101	目前IPv4地址状况
2107	Ascend 隧道管理协议(ATMP)
2121	影响MARS群规模的论点
2126	TCP顶端的ISO传送服务(ITOT)
2133	IPv6基本套接字接口扩展
2147	TCP和UDP over IPv6巨型报
2175	MAPOS 16——SONET/SDH上具有16位寻址的多访问协议

(续)

编 号	标 题
2176	IPv4 over MAPOS第1版
2185	向IPv6过渡的选路问题
2202	HMAC-MD5和HMAC-SHA-1测试实例
2205	资源预留协议(RSVP)——第1版功能性技术规范
2207	用于IPSEC业务流的RSVP扩展
2236	Internet组管理协议第2版
2240	域名分配的合法依据
2283	为BGP-4用的多协议扩展
2292	用于IPv6的新式的套接字 API
2300	Internet正式协议标准
2328	OSPF第2版
2344	为移动IP用的反向隧道
2353	IP网络中的APPN/HPR——APPN实施者研讨会总结文件
2365	行政管理范围内的IP组播
2373	IPv6寻址体系结构
2374	IPv6可集聚全球单播地址格式
2375	IPv6组播地址指派

附录B RFC精选

有数十个以某种或其他形式表示的与 IPv6 相关的 RFC。附录 A 就包含了本书写作时可以得到的 RFC 的索引。本附录精选了一些 RFC，它将阐明 IPv6 的发展，通过这些建议的标准本身来探索某些可能的解决方案，以反复了解 IPv4 存在的问题。

RFC 1287 是最早的 RFC 之一，它描述了未来的 Internet。1991 年由 IETF 和 IAB 的一些头头们编写。该文件定义了升级 Internet 体系结构的一些问题和方法。引入的某些概念是非常有远见的，直至最近才广为流传，例如：“基于 IP” 连网来定义 Internet 的概念是陈旧的，作者建议未来的 Internet 应更有益地称为“基于应用”的。

写于 1993 年的 RFC 1454，比较了为下一代 Internet 协议所写的现有的提案。将该文件所得出的结论与最终放入 IPv6 标准中的解决方案进行比较，会很有意思。写于 1994 年的 RFC 1671，对过渡到 IP 新版本将会引发的问题提供了一个考虑周到的见解。由 Christian Huitema 起草的 RFC 1715，介绍了一个有意义的测量方法，可用来测量一个网络地址体系结构分配地址的效率。

本附录还包括了 RFC 2373 和 2374，它们分别为 IPv6 寻址体系结构的标准跟踪技术规范 and IPv6 可集聚单播地址格式的标准跟踪技术规范。为了试图在本书中介绍更多的信息和清楚地解释起见，简化和跳过了某些材料。对书中涉及到的 RFC，读者可对照其详细材料。在此要提出的是 1998 年底以前描述 IPv6 协议技术规范 RFC 1883 将被一个新的 RFC 所替代。与其将一个即将被废弃的 RFC 包括进去，不如建议读者在线寻找最新的 Internet 文件 (Internet 草案或 RFC)。

本书包括了一定数量的相关的 RFC。极力推荐读者通过下面列出的 Internet 上的 RFC 库地址来核对这些文件：

<http://www.pmg.lcs.mit.edu/rfc.html>

<http://www.csl.sony.co.jp/rfc/>

<http://www.cis.ohio-state.edu/Excite/AT-rfcsquery.html>

<http://info.internet.isi.edu/ls/in-notes/rfc/files>

<http://www.nexor.com/public/rfc/index/rfc.html>

如果上述这些地址无效或使用起来不方便，建议读者使用你最爱用的搜索引擎来检索。

RFC 1287 未来的Internet体系结构

网络工作组

RFC: 1287

D.clark

MIT

L.Chapin

BBN

V. Cerf

CNRI

R. Braden

ISI

R. Hobby

UC Davis

1991年12月

提示

这是一个提供信息的 RFC，它讨论 Internet 体系结构未来可能演变的重大方向，及走向期望目标的建议步骤。它是提供给 Internet 社区讨论和评议用的。本文是为 Internet 社区提供信息，而不是指定 Internet 标准，本文的分发不受限制。

目录

1. 绪论	[1]
2. 选路与寻址	[3]
3. 多协议体系结构	[5]
4. 安全性体系结构	[7]
5. 业务流控制与状态	[9]
6. 现代应用	[10]
7. 参考文献	[11]
附录A 建立步骤	[12]
附录B 组成员	[15]
安全性考虑	[16]
作者地址	[16]

1. 绪论

1.1 Internet体系结构

作为 TCP/IP 协议集之后的巨大计划，Internet 体系结构在 70 年代后期由一个网络研究小组^[1, 2, 3, 4]开发并测试。80 年代初期，体系结构中加入了若干重要的特性，如子网化、自治系统和域名系统^[5, 6]。最近又加入了 IP 组播^[7]。

在本体系结构框架内，Internet 工程任务组 (IETF) 一直以极大的活力和效率为 Internet 策划、

定义、推广、测试和标准化协议进行不懈的工作。已完成的三个特别重要的领域是选路协议、TCP性能与网络管理。同时，Internet基础设施继续以惊人的速度增长。自1983年1月ARPANET第一次从NCP转换成TCP/IP时，Internet的销售商、管理员、专家和研究人员一直以极大的努力坚强地工作，为他们的成功而奋斗不息。

定义Internet体系结构的一组研究人员形成了Internet活动董事会(IAB)的初始成员。IAB由1981年DARPA建立的一个技术顾问组发展成为Internet总技术和策略监督实体。IAB的成员年年有变化，以便更好地体现Internet社区中需求和议题的变化，及反映Internet的国际化，但仍旧保持协议体系结构制定的关系。

IAB创建了IETF，为Internet实施协议的开发和工程设计。为了管理不断发展的IETF活动，IETF主持在IETF内建立了Internet工程指导组(IESG)。IAB和IESG密切配合，共同批准IETF内开发的协议标准。

过去几年中，对基础体系结构有着不断增加的严峻考验的迹象，大部分由于Internet持续不断的增长引起。对这些问题的讨论经常反映在许多主要的发送文件清单中。

1.2 假设

对当前Internet体系结构中的问题，解决的优先次序取决于人们对TCP/IP与OSI协议集未来关系的观点。一种观点是让TCP/IP夭折在成功之中，然后转换到OSI协议。然而，许多在Internet协议产品和服务上花过大力气并获得成功的人们，急于要在已有的框架内尝试解决新的问题。而且，有些人相信OSI协议将会遇到许多同样类型的问题。

为了着手解决这些问题，IAB和IESG于1991年1月联合组织了一天有关Internet体系结构议题的讨论会。这次会议的框架是由Dave Clark(见附录A中的幻灯片)整理的。关于TCP/IP与OSI协议的关系和未来方向问题上的讨论生气勃勃、富有挑战，有时还有激烈的争论。会议的重要成果是在下一个5~10年涉及网络世界的下述四个基本假设上达成了共识。

(1) TCP/IP和OSI协议集将在一个长时期内共存。

OSI协议集引入的背后是强有力的政治和市场力量，以及以某些技术优势作后盾。然而，TCP/IP牢固确立的市场位置意味着在可预见的未来非常可能继续使用。

(2) Internet将继续包括各种各样的网络和服务，永远不会是由一个单网络技术构成的。

实际上接到Internet上的网络技术和特性的范围在下一个十年还将增加。

(3) 商用和专用网也将加入，但不能期望公共通信提供全部服务。将会形成公用网和专用网、公共通信与专用线路混用的局面。

(4) Internet体系结构要能达到 10^9 个网络的规模。

Internet的规模历史性地呈指数增长，在将来的某个时候估计可能会饱和，但预测到什么时候饱和，差不多和预测未来的经济一样容易。在任何情况下，负责工程设计的需要考虑一个有能力扩展到最坏情况规模的体系结构。指数9是比较模糊的数字，估计在7~10之间变化。

1.3 开始一个规划过程

IAB和IESG会议的另一个成果是在体系结构进化中的下列五个最重要的领域上形成了共识。

(1) 选路与寻址

这是一个最急需解决的体系结构的问题，因为它直接关系到Internet继续成功增长的能力。

(2) 多协议体系结构

Internet正在朝着广泛的既支持TCP/IP又支持OSI协议集的方向迈进。对两个协议集的支持带来了技术难题,需要有一个计划,也就是一个体系结构来增加成功的机会。人们开玩笑地把这个领域看成是:“为了造福人类,问题变得更艰难”。

Clark观察到转发网关(如邮件网关)在Internet运行中是非常有生命力的,但是它不属于体系结构或规划的一部分。该组成员讨论了围绕包含这样的网关的部分网络连接来建立体系结构的可能性。

(3) 安全性体系结构

在设计Internet体系结构时,虽然考虑到了军事上的安全性,可是现代安全性议题是非常广泛的,它也包括商业上的需求。还有,经验表明除非一开始就把它建立到体系结构中去,否则是很难在协议集内再加入安全性。

(4) 数据流控制及状态

Internet将扩展以支持如语音和视频这样的“实时”应用。这就需要网关中有新的包排队机制(数据流控制)和附加的网关状态。

(5) 现代应用

随着基础的Internet通信机制的成熟,需要不断革新和标准化,以创建新形式的应用。

IAB和IESG于1991年6月再次在SDSC召开三天的会议,讨论这5个课题。这次会议多少有点反常,被称为“体系结构的再处理”,召集在一起开会,表明有坚强的决心朝着规划体系结构的进化迈出第一步。除了IAB和IESG以外,由32人组成的小组,包括了研究指导组(IRSG)的成员及少量的特邀客人。会议的第二天,分成5个组讨论,每个组讨论1个领域的问题。附录B列出了成员名单。

本文件是从这些组的主席报告中收集得到的。该材料在亚特兰大召开的IETF会议上介绍过,同时发表在会议记录中^[8]。

2. 选路与寻址

为了应付Internet预期的增长和功能的演变,IP寻址和选路结构需要改变。人们预测:

- Internet将用完IP网络地址的某些地址类,如B类地址。
- 尽管该地址空间当前已被子分和管理,Internet将用完全部32位IP地址空间。
- IP网络号的总数将增长到一定时刻,就连较好的选路算法也不再能完成基于网络号的选路。
- 为了允许适应不同的TOS和策略,从一个源到一个目的地需要多个路由。这将需要新的应用和多种多样的转运服务来推动。源或源的代理,必须控制路由的选择。

2.1 建议的方法

处理这些事情所需方法有通用的约定。

(1) 必须改变寻址方案,使网络号集聚成较大的单位,以此作为选路的基础。自治系统或行政管理域(AD)就是一种集聚的例子。

集聚将完成若干目标:确定采用策略的范围,控制选路部件数,以及为网络管理提供部件。有些人认为如一个嵌套的AD那样,可进一步组合一些集聚。

(2) 必须提供某些有效的方法来计算公共路由,以及某些通用的方法来计算“特定”的路由。

特定路由的通用方法将由“源路由”指定的形式来建立路由。

会上,对期望AD如何集聚或选路协议如何组织来处理集聚边界,尚未达成完全一致的意见。可能使用一个非常通用的方案(参考文献Chiappa),可是某些人倾向于一个更受限制的方案,并定义期望的网络模型。

为了处理地址空间耗尽的问题,必须要么扩展地址空间,要么在网的不同部分重用 32位地址字段。下一节将描述几种可能的地址格式。

或许更重要的问题是如何向新的方案迁移。所有迁移计划都需要某些路由器(或者Internet内的其他部件)能重写包头,以适应只处理老格式或者只处理新格式的主机。除非格式变换能够进行算法上的推理,迁移本身需要在变换元素中建立某种状态。

我们并不计划对体系结构进行一系列“小”的改变。从现在起,我们将着手实施一个计划,以便能渡过地址空间耗尽的难关。比起 Internet社区近期承担的任务而言,这是一系列更长的规划行动,但迁移问题需要一个漫长的研制周期,同时很难发现有效的方法来处理某些更直接的问题。诸如B类地址的耗尽问题,从某种意义上讲,就其本身而论,不用长时间。因此,一旦我们着手进行一项变更的计划,就要求全部替代当前的 32位全球空间。(如果有非常巧妙、能很快应用、而又留有发展空间的想法浮现出来的话,本结论总是可以被修订的。这并不意味着我们不鼓励对于短期行动的创新性设想。但需要指出的是即使小的变更也要花长时间去推广应用)。

仅有地址空间变换是不够的。同时还需要提供一个规模可伸缩的选路体系结构以及能更完善地管理Internet的工具。建议的方法就是把AD作为选路的集聚单位。我们已经有部分方法来实现。IDRP能实现这一功能。BGP的OSI版本(IDRP)也能实现。BGP改进后也可实现。另外需要的附加设施是要有一张网络号到AD的映象表。

为了若干原因(特定的路由和地址变换以及计费和资源分配),我们将从“无状态”网关模型做起,在该模型中仅把预先计算好的路由存放在网关中,然后发展成另一个模型,该模型至少在某些网关中每个连接有状态。

2.2 扩展的IP地址格式

扩展的IP地址格式有三个比较好的选择。

(1) 用同样位数不同含义的地址字段代替 32位地址字段。由此地址的唯一性只是在某个较小的区域(一个AD或者一个集聚的AD)内,而不在全球范围。当包穿越边界时,边界上的网关重写包地址。

问题:(a)必须找到并重写包内的地址;(b)主机软件需作修改;(c)必须用某些方法建立地址映象。

本方案是Van Jacobson的研究成果,也可参见Paul Tsuchiya为NAT所作的工作。

(2) 将32位地址字段扩展至64位(或其他值),用以保持一个全球主机地址和主机所在的AD。

这样的选择方案提供一个从主机到作为选路根据的AD的烦琐的映射。普通路由(是指基于目的地址而不需考虑源地址的选路)可直接从包地址中得到,正如目前进行的,不需要任何事先建立过程。

(3) 将32位地址字段扩展至64位(或其他值),并用该字段作为“平面”主机标识符。需要时,用建立连接来为路由器提供从主机标识符到AD的映射。

这64位地址如以太网地址一样，可用来简化主机标识符的分配问题。

所有以上选择方案作为迁移的一部分，都需要一个地址重写模块。第二和第三方案 IP头需要改变，所以主机软件也要随之改变。

2.3 建议的行动

建议采取下列行动：

(1) 时间表。

对于上面提出的各种问题，要编制出一个估计的详细时间表，又要对一个新的寻址 /选路体系结构编制出开发和推广应用的相应时间表。用这些时间表作为根据来评价用于变革的各种提案。这是IETF的任务。

(2) 新地址格式。

探索下一代地址格式的可选方案并提出一个迁移计划。特别是要构造一个作地址映射的网关样机。要理解这个任务的复杂性，以便指导我们思考有关迁移的方案选择。

(3) 基于AD的选路。

采取步骤做出作为选路基础的网络集聚 (AD)。特别是要为映射网络号到 AD的一张全球映射表探索若干可选方案。这是 IETF的事情。

(4) 基于策略的选路。

基于策略的选路要继续当前的工作。有下列明确的目标：

- 寻求方法以控制指定策略的复杂性 (这是一个人类的接口议题，而不是算法复杂性议题)。
- 充分了解在网关中保持连接状态的议题。
- 充分了解连接状态建立议题。

(5) 进一步集聚的研究。

作为研究活动，探索如何将 AD集聚到仍然较大的选路元素中。

- 考虑体系结构应定义 AD的“角色”，还是集聚的“角色”。
- 考虑用一个万能的选路方法，还是在 AD和集聚以内和以外用不同的选路方法。

现有的计划如 DARTnet 工程项目帮助解决这些议题中的几个：如网关内的状态、状态建立、地址映射和计费等。研究开发界的其他试验也承担本领域的研究。

3. 多协议体系结构

改变Internet以支持多协议集引起以下三个特殊的体系结构问题：

- 如何正确地定义 Internet？
- 如何设计支持多个又不论何种协议集的 Internet？
- 是为部分还是过滤了的网络设计连通性？
- 如何在体系结构中加入能明显地支持应用的网关？

3.1 什么是“Internet”？

如果不首先确定我们认为的 Internet是什么或者应该是什么的话，要想建设性地处理“多协议Internet”议题将是非常困难的。我们要把“Internet”和“Internet社区”区别开来，前者是由通信系统组成的，而后者是一群人和组织。大部分人接受后者的松散定义，即“认为他

们自己是Internet社区的一部分”。Internet本身这种“社会学的”定义似乎是没有用的。

不久以前，Internet被定义为IP网络连通性(IP和ICMP过去是、现在仍然是唯一“需要”的Internet协议)。如果我 ping 你，你能 ping 我，那么我们都在Internet上，同时，Internet令人满意的工作定义可构想为IP对话系统的接近可过渡的最后结果。这样的Internet模型是简单的、统一标准的，或许最重要的是可测试的。IP网络连通性模型可清楚地判别系统是否“在Internet上”。

随着Internet的增长及其使用的技术已广泛的被商界接受，对一个系统“在Internet上”的含义已经有所改变，应当包括：

- 具有部分IP网络连通性，受限于策略过滤器的任何系统。
- 运行TCP/IP协议集，不管是否从Internet的其他部分实际上可接入的任何系统。
- 能交换RFC 822邮件，无需邮件网关的干预或邮件对象的转换的任何系统。
- 有e-mail连通到Internet，不论是否需要邮件网关或邮件对象转换的任何系统。

对Internet的这些定义仍是基于原始的网络连通性概念，只是“栈的向上移动”。

在此，基于有区别的统一概念，提出Internet的新定义：

- “老的”Internet概念：以IP为基础，组织的原则是IP地址，也就是一个公共的网络地址空间。
- “新的”Internet概念：以应用为基础，组织的原则是域名系统和目录，也就是一个公共的(虽然必定是多形式的)应用名字空间。

这就告诉我们，“连接的状况”概念传统上是与IP地址(通过网络号)紧密联系在一起的，应该代之以与存放在分布式Internet目录中的名字和相关标识信息联系在一起的。Internet基于名字的定义意味着一个大得多的Internet社区，以及一个更为动态(和不可预测的)可运转的Internet。对Internet体系结构的争论，是基于在很宽的范围内对未来可能发展的适应性，而不至于局限于原来的设想。

3.2 基于过程的多协议Internet模型

与其制订一个特殊的“多协议Internet”，接受一个预先确定特定协议数量的体系结构，倒不如建议采用一个面向过程的Internet模型，它可以适应不同的协议体系结构，符合传统的“能工作”原则。

面向过程的Internet模型，作为一个基本前提，主张不包括稳定状态“多协议Internet”的体系结构。最基本的驱动Internet进化的力量不是推动它朝多协议多样化发展(虽然事实上永远不可能达到)。要说明的是Internet发展的趋势是向同质性进化，作为最“热动稳定”状态，下面描述一个新的基于过程的Internet体系结构的四个部分：

第1部分：核心Internet体系结构。

这是传统的基于TCP/IP的体系结构。是Internet进化的“磁铁中心”，公认(1)同质性仍然是处理互连网多样化的最好方法；(2)IP网络连通性仍然是Internet的最佳基本模型(在全球Internet中，不论IP无处不在的实际状态是否是一个现实)。

一开始，Internet体系结构只包括第1部分。然而Internet的成功在于它超出了原来的设想。无处不在和高度统一，对极大地丰富Internet“基因库”作出了贡献。

新Internet体系结构增加的两个部分扩大了Internet的广度和深度。

第2部分：链路共享。

传输媒体、网络接口及低层链路协议等物理资源是由多个非交互的协议集所共享的。这部分体系结构被认为是必须且适合于共存的，但不涉及到互操作性；被称为 *ships in the night*(S.I.N.)。

当然，共存的协议集实际上不是纯粹孤立的；在真正的 Internet 系统中，S.I.N. 会引发管理、无冲突、协调和公平性等议题。

第3部分：应用互操作性。

虽然缺乏互连普遍性(即“基础栈”的互操作性)，但只要能在 Internet 系统的不相邻社区之间安排应用的基本语义能以传递信息，仍然可能获得普遍的应用功能。这可以通过应用转发站，或者由用户代理，对不同的由共同语义表示的应用服务提供一致的虚访问方法来完成。

体系结构的这一部分，强调了 Internet 的最终作用是作为应用间的通信基础，而不是它本身的结局。在一定程度上，使一个应用群体和他们的用户能够从一个基础协议集过渡到另外一个，而不会发生难以接受的功能丢失，这可被称为“过渡起动器”。

将第2和第3部分加入到原始的 Internet 体系结构中，充其量是一件好坏半掺的事情。虽然大大增加了 Internet 的广度和 Internet 社区的规模，但也会引入复杂性、价格和管理等重大问题，同时还可能会出现功能的丢失(特别对第3部分而言)。第2和第3部分不可避免地背离了第1部分所表示的同质性，但这是我们所不希望的。为了扩展 Internet 广度，某些功能丢失了，还要承受附加的系统复杂性和成本。而在一个完美的世界中，应该不需付出这些代价就可换得 Internet 的进化和扩展。

目前有一种趋势，Internet 的进化倾向于第1部分表示的同质性体系结构，而不是第2和第3部分表示的折衷的体系结构。第4部分表达了这种趋势。

第4部分：混合/集成。

第4部分认识到可以从不同的 Internet 协议体系结构中集成类似的元素以形成混合体，以便减少 Internet 系统的多样化和复杂性。同时也认识到可以影响已存在的 Internet 基础设施以便 Internet 吸收“新东西”，并把已建立的 Internet 的测试、评价和应用实践融入到“新东西”中去。

本部分表达了 Internet 的发展趋势，作为一个系统，试图回到原来由第1部分统一的体系结构所表示的“美好的状态”。虽然 Internet 将永远不会在未来的任意时刻回到统一的状态，但这是一个对 Internet 进化起作用的力量。

按照这个动态的进程模型，在 TCP/IP 栈上通过 RFC 1006 运行 X.400 邮件，集成 IS-IS 选路，传送网关以及对 IP 和 CLNP 协议的单个共同后继协议的开发，都是很好的例子。在第1部分主张的“磁场”影响下，参照第4部分混合的动态，它们显示了背离第2和第3部分的非统一性，而走向更好的同质性。

4. 安全性体系结构

4.1 哲学准则

Internet 安全性体系结构开发的主题是简单、可测试性、可信度、采用的技术和安全周界标识。

- 安全性比协议和密码保密措施要求更多。
- 安全性体系结构和策略应该简单且容易理解。复杂性会引起错误理解和不良的实现。
- 实现应该是可测试的，以便确定是否满足了策略。

- 我们认为使任何安全性体系结构运行的硬件、软件和人是可以信任的。假设安全性策略实施的技术设备至少和个人计算机及工作站具有同样的能力。我们不需要能力差的部件受到自保护(但可能会用诸如链路级密码编码设备进行外部补救)。
- 最后, 认定安全性有效保护的周界是最根本的。

4.2 安全性周界

有4种可能的安全性周界: 链路级、网络 / 子网级、主机级和进程 / 应用级。每种施加不同的需求, 能够接纳不同的技术, 并能对何种系统部件可以被认为是有效的做出不同的假设。

隐私强化邮件是一种进程级安全系统的例子, 另一个例子是为SNMP提供身份验证和保密。主机级安全性一般是在主计算机的通信口上用一个外部安全机制。网络或子网安全性则应用从子网到“外部”的网关/路由器上的外部安全性能力。链路级安全性采用传统的点对点或媒体级(如以太网)密码编码机制。

关于网络/子网安全性保护存在许多开放的问题, 不单是主机级(端/端)安全性方法与网络/子网级安全性方法之间存在潜在的不匹配, 而且网络级保护也不能处理安全性周界内出现的威胁。

在进程级采用保护, 假定基础的程序和操作系统机制是可以信任的, 不会由于使用了相应的安全机制而妨碍应用程序。当安全性周界在系统体系结构中向下移至链路级, 就要做有关安全威胁的许多假设, 以便得出这样的论点, 就是在一个特定周界的实施是有效的。例如, 如果只有链路级使用加密编码, 我们可以假设来自外部的攻击, 只通过通信线, 那么主机、交换机和网关实际上是被保护的, 同时人和所有部件中的软件都是可以信任的。

4.3 期望的安全性服务

如果在系统的应用级和较低级实现选定和非选定的接入控制, 则需要可验证的正规的名字。除此之外, 还需要实施完整性(防修改、防欺骗、防重放), 保密性和防止否认服务。在某些情况下, 可能还需要防止报文传输的否认或防止秘密信道。

已经有一些标准部件用以建立 Internet 安全系统。可以采用密码算法(如DES、RSA、El Gama1和其他可能的公共密钥和对称密钥算法), 也可以采用如MD2和MD5的散列函数。

根据OSI的意义需要可鉴别的名字, 并且为了便于人们了解标识符和目录服务, 非常需要一个指派标识符以及广泛使用目录服务的基础设施。把公共密钥与可鉴别的名字捆绑在一起, 并把能力和许可与可鉴别的名字捆绑在一起的认证概念, 具有很多优点。

在路由器/网关级, 采用地址和协议过滤器及其他配置控制, 能有助于形成一个安全性系统。把建议的OSI安全性协议3(SP3)和安全性协议4(SP4)作为Internet安全性体系结构的可能要素, 要给予认真的考虑。

最后, 必须看到, 在未实施安全存储的PC或笔记本电脑系统上, 安全地存储秘密信息(诸如一个公共密钥对的秘密部分), 还没有好的解决方案。

4.4 建议的行动

建议采取下列行动:

- (1) 安全性参考模型。

需要建立一个Internet的安全性参考模型，并迅速地得到开发。该模型应该建立目标周界，并用文件形式建立安全性体系结构目标。

(2) 隐私强化邮件(PEM)。

对于隐私强化邮件，最关键的步骤看来是建立：(1)认证生成和管理基础设施；(2)X.500目录服务以提供通过可鉴别的名字访问公共密钥。在推广使用本系统时，还需要对专利方面的限制和出口限制给予认真关注。

(3) 分布式系统安全性。

对分布式系统的应用，不论是简单的客户机/服务器系统还是复杂的分布式计算环境，都需要检查安全设施。例如，对授予与可鉴别的名字捆绑在一起的许可/能力的认证的实用性应受到检查。

(4) 主机级安全性。

对面向主机的安全性，应当对SP4予以评估，SP3也在考虑之列。

(5) 应用级安全性。

不论是为了服务的直接实用性(如PEM.SNMP身份验证)，还是为了获得能够形成Internet安全性体系结构精华的有价值的实际经验，都应该实施应用级安全性服务。

5. 业务流控制与状态

目前的Internet平等地处理所有的IP数据报。每个数据报对同一连接、同一应用、同一应用类别、同一用户类，不论它和其他包有任何关系，都是独立地转发的。虽然在IP头中定义了服务类型位和优先权位，但通常都没有实施，事实上还不清楚如何去实施它们。

众所周知，未来的Internet需要支持尽力而为所不能满足的大量应用，如电视会议的包图像和语音。为了处理实时业务流，要求在路由器中有以附加的状态来控制业务流控制机制。

5.1 假设和原则

- 假设：Internet需要为业务流的特定子集支持性能保证。

遗憾的是对术语“性能”、“保证”或“子集”，远不能给出精确的定义。研究仍需要对这些问题的回答。

- 默认的服务将继续是当前无服务保证的“尽力而为”数据报分发服务。
- 路由器机制可分割为两部分：(1)转发路径；(2)发生在后台的控制计算(如选路)。

转发路径必须高度优化，有时由硬件辅助完成，因此相对而言很昂贵，而且难于变更。运行在转发路径上的业务流控制机制，是由发生在后台的选路和资源控制计算创建的状态来控制的。在改变路由器的转发路径时，最多起动一次，所以最好一开始就使它正确。

- 新的扩展必须运行在一个高度异质的环境中。在该环境中，某些部分将永远不支持保证。对一个路径上的某些段(如高速局域网)，即使当显式资源预留不用时，“超配给”(即超过容量)也会对实时业务给予满足要求的服务。
- 组播分发或许是最根本的。

5.2 技术议题

需要解决的技术议题，包括：

(1) 资源建立。

为了支持实时业务流，从源到目的地的路由上的路由器中需要预留资源。该新的路由器状态应该是“硬”的(如建立连接)还是“软”的(即缓冲的状态)？

(2) 资源捆绑与路由捆绑。

选择从源到目的地的一条路由传统上是由一个动态选路协议来完成的。资源捆绑和选路可以重叠在单个复合进程中，或者也可以基本上独立地完成。这就要求在复杂性和效率之间折衷考虑。

(3) 另一组播模型。

IP组播用一个逻辑寻址模型，在该模型中，目标地址本身与一个组联系。在 ST-2 中，一个组播会话中的每个主机在它的建立包中包括一系列显式目标地址。每一种方法都有优点和缺点。当前还不十分清楚对 n 路电视会议而言，哪个会占优势。

(4) 资源建立与行政管理域间的选路。

不论倾向于哪种资源保证，必须保持穿越一条任意的端对端并包括多个 AD 的路由。因此，任何资源建立机制必须与包含在 IDPR 中的路由建立机制平稳地配合。

(5) 计费。

资源保证子集(“类别”)可以是自然的计费单位。

5.3 建议的行动

此处所谓的行动是指对上面列出的技术议题的进一步研究，紧随其后的是相应协议的开发和标准化。DARTnet，DARPA 研究测试床网络，在本研究中将起重要作用。

6. 现代应用

人们不禁要问“我们想要何种基于网络的应用，为什么现在还没有？”很容易列出一张潜在应用的大表，其中许多都将基于客户机/服务器模型。然而问题中更有意思的是：“为什么还没有人来做呢？”回答是：方便应用程序编写的工具尚不存在。

首先，对于许多将用于穿越网络的数据术语，需要一套公共交换格式。定义了公共交换格式后，还需便于开发应用程序移动数据的工具。

6.1 公共交换格式

为使信息有意义，应用程序必须知道它们要交换的信息的格式。考虑下面的格式类型：

(1) 文本——文本是最标准的，但今天的国际性 Internet 还需要有除了 USASCII 以外的字符集。

(2) 图像——当进入“多媒体时代”，图像变得越来越重要，但需要对如何在信息包中表示图像信息取得一致。

(3) 图形——和图像一样，矢量图形信息需要一个共同的定义。有了定义的格式才能交换类似结构蓝图的细节。

(4) 视频——先要知道从网络上来的视频信息的格式，才能在工作站上运行视频窗口。

(5) 模拟音频——当然，人们需要的是伴有声音的视频，但这样的格式应该可以表示所有类型的模拟信号。

(6) 显示——我们打开工作站上的窗口，并打开另一个人的工作站上的窗口，给它显示与研究项目有关的某些数据，所以需要有一个通用的窗口显示格式。

(7) 数据对象——对进程间的通信，类似整数、实数、串等数据的格式需要一致。

这些格式的相当一部分正在由几个标准组定义。我们需要为 Internet 的每一类取得一种一致的格式。

6.2 数据交换方法

应用程序将需要下列的数据交换方法：

(1) 存储转发。

不是每个人所有时间都在网上。需要一个标准手段向有时连在网上的主机提供信息流，也就是需要一个通用的存储转发服务。组播也应包括在这一服务中。

(2) 全球文件系统。

在网上，大部分数据访问可以被分解成单个文件访问。如果有一个真正的全球文件系统，那就能访问Internet上的任何文件(假定被许可的话)。你曾经需要用FTP吗？

(3) 进程间通信。

对一个真正的分布式计算环境，需要通过一些手段使进程在网络上能通过一个标准方法来交换数据。这样的需求包括 RPC、API等。

(4) 数据广播。

许多应用程序要求发送同样的信息到其他许多主机，因此需要一个标准且高效的方法来完成这功能。

(5) 数据库访问。

对于好的信息交换，需要为访问数据库指定一个标准方法。全球文件系统能使你获得数据，但数据库访问方法将告诉你有关它的结构和内容。

上述许多项正在由其他组织着手拟订，但对Internet的互操作性，还需要在方法上取得一致。

最后，现代应用需对本文中两个较早领域的问题寻找解决方案。从业务流控制与状态领域而言，应用需要发送实时数据的能力。这意味着数据能在一个确定的时间范围内分发。从安全性领域而言，应用也需要全球身份验证和访问控制系统。今天的 Internet由于缺乏可信度和安全，失去了许多有用的应用。这要求在明天的应用中得到解决。

7. 参考文献

- [1] Cerf, V. and R. Kahn, "A Protocol for Packet Network Intercommunication," IEEE Transactions on Communication, May 1974.
- [2] Postel, J., Sunshine, C., and D. Cohen, "The ARPA Internet Protocol," Computer Networks, Vol. 5, No. 4, July 1981.
- [3] Leiner, B., Postel, J., Cole, R., and D. Mills, "The DARPA Internet Protocol Suite," Proceedings INFOCOM 85, IEEE, Washington DC, March 1985. Also in: IEEE Communications Magazine, March 1985.
- [4] Clark, D., "The Design Philosophy of the DARPA Internet

Protocols", Proceedings ACM SIGCOMM '88, Stanford, California, August 1988.

- [5] Mogul, J., and J. Postel, "Internet Standard Subnetting Procedure", RFC 950, USC/Information Sciences Institute, August 1985.
- [6] Mockapetris, P., "Domain Names - Concepts and Facilities", RFC 1034, USC/Information Sciences Institute, November 1987.
- [7] Deering, S., "Host Extensions for IP Multicasting", RFC 1112, Stanford University, August 1989.
- [8] "Proceedings of the Twenty-First Internet Engineering Task Force", Bell-South, Atlanta, July 29 - August 2, 1991.

附录A 设定步骤

幻灯片1

Internet向何处去？
体系结构的选择
IAB/IESG -- 1990年1月
David D. Clark

幻灯片2

设定讨论的课题

目的：

- 为IAB、IESG及Internet社区建立一个理解的共同框架。
- 了解要解决的问题集
- 了解为我们敞开的解决方案的范围
- 得出某些结论或“总结论”。

幻灯片3

若干声明——我的见解

两个不同的目标：

- 使建立Internet成为可能。
- 定义Internet的一套协议。

声明：这些目标有非常不同的含义。协议只不过是一种手段，然而是一种有力的手段。

声明：如果Internet获得成功及增长，就将需要专门的设计。这就需要至少另一个十年的继续努力。

声明：不加控制的增长将会导致混乱。

声明：从根本上解决问题看来是走向成功的唯一方法。从上向下命令是无力的。

幻灯片4

报告提纲

(1) 问题空间和解决方案空间。

(2) 一系列专门问题——供讨论用。

(3) 回到顶层问题——供讨论用。

(4) 行动计划——供总体讨论用。

设法从技术研究中将功能需求分离出来。

了解我们是如何受到问题空间和解决方案空间的限制。

是否体系结构除了协议以外别无其他？

幻灯片5

问题空间是什么？

选路与寻址：

大到什么程度，采用何种拓扑结构及选路模型？

逐渐变大：

用户服务；主机和网络采用何种技术？

Internet的舍弃：

计费、控制的使用和修复故障。

新服务：

视频？事务处理？分布计算？

安全性：

终端节点还是网络？ 路由器还是转发器？

幻灯片6

限定解决方案的空间

从当前的状态能迁移到多远？

- 我们能改变IP头吗(除了OSI外)？
- 我们能以命令方式改变主机的需求吗？
- 我们能管理一个长期迁移目标吗？——始终如一的方向与多种多样的目标、资金来源。

我们能接受网络级的连通性吗？

- 转发将来会被抛弃吗？
- 安全性以及变换是一个关键议题。
- 需要一个基于转发的体系结构吗？

如何能够和必须管理Internet？

- 我们能管理或者限制网络的连通性吗？

研究开发什么协议？一个还是多个？

幻灯片7

多协议Internet

“把问题想得难一点对人类有好处。”

我们是迁移、互操作还是容忍多协议？

- 不是所有的协议集在同一时期都有同样的功能范围。
- Internet需要特定的功能。

声明：基本的矛盾(非宗教性的或恶意的)：

- 满足Internet积极进取的需求。
- 处理OSI迁移。

结论：一种协议必定为主导，其他协议必定为辅助。我们什么时候“切换”到 OSI？
请考察本文下面的每张幻灯片。

幻灯片8

选路和寻址

什么是Internet的目标规模？

- 如何将地址和路由联系起来？
- 拓扑模型是什么？
- 什么是可能的解决方案？

选路要求什么样的策略范围？

- BGP和IDRP是两个解答。问题在那儿？
- 固定类别或可变路径？
- 源控制的选路是最低要求。

如何无缝地支持移动主机？

- 新地址类，再捆绑到本地地址，用DNS吗？

是否要推动Internet组播？

幻灯片9

逐渐变大——一个老题目

(寻址与选路在前一张幻灯片上。)

在下一个十年中需要什么样的用户服务？

- 我们能否构筑一个计划？
- 需要体系结构方面的改变吗？

是否有更好的处理速度、包大小等范围的需求。

- 是否取消分段策略？

我们将支持什么主机范围(如UNIX 环境)？

幻灯片10

处理舍弃

Internet是由独立管理和控制的部分组成的。

为网络收费需要什么支持？

- 体系结构不隐含按容量收费、重记帐和为丢失包付费。
- 是否需要控制以提供记帐标识符或选路？

需求：必须支持有控制共享的链路。(简单的形式是基于链路标识符的类别)。

- 如何一般化？

对故障隔离是否更加需要？(我投赞成票！)

- 我们如何能找到可以交谈的经理们？
- 我们需要主机上的服务吗？

下载

幻灯片11

新服务

要支持视频和音频吗？是实时吗？百分比多少？

- 需要计划从研究结果得到什么，什么样的质量？
- 向供货商交底的目标日期。

我们能“更好”地支持事务处理吗？

- TCP 能做吗？VMTP 呢？介绍呢，还是刹车？

哪些象样的应用即将出笼？

- 分布计算——它真的将发生吗？
- 信息网络技术吗？

幻灯片12

安全性

能坚持说终端节点是唯一防线吗？

- 在网络内部我们能做什么？
- 能要求主机做什么？

能容忍转发器或安排它们的结构吗？

能找到一个更好的方法来构筑安全性边界吗？

需要全球身份验证吗？

有新的主机需求吗？

- 登录。
- 身份验证。
- 管理接口。电话号码或访问点。

附录B 组成员

第1组：选路与寻址

Dave Clark, MIT [Chair]
Hans-Werner Braun, SDSC
Noel Chiappa, Consultant
Deborah Estrin, USC
Phill Gross, CNRI
Bob Hinden, BBN
Van Jacobson, LBL
Tony Lauck, DEC.

第2组：多协议体系结构

Lyman Chapin, BBN [Chair]
Ross Callon, DEC
Dave Crocker, DEC
Christian Huitema, INRIA
Barry Leiner,
Jon Postel, ISI

第3组：安全性体系结构

Vint Cerf, CNRI [Chair]

Steve Crocker, TIS

Steve Kent, BBN

Paul Mockapetris, DARPA

第4组：业务流控制与状态

Robert Braden, ISI [Chair]

Chuck Davin, MIT

Dave Mills, University of Delaware

Claudio Topolcic, CNRI

第5组：现代应用

Russ Hobby, UCDavis [Chair]

Dave Borman, Cray Research

Cliff Lynch, University of California

Joyce K. Reynolds, ISI

Bruce Schatz, University of Arizona

Mike Schwartz, University of Colorado

Greg Vaudreuil, CNRI.

安全性考虑

安全性议题在第4节讨论。

作者地址

David D. Clark

Massachusetts Institute of Technology

Laboratory for Computer Science

545 Main Street

Cambridge, MA 02139

Phone: (617) 253-6003

EMail: ddc@LCS.MIT.EDU

Vinton G. Cerf

Corporation for National Research Initiatives

1895 Preston White Drive, Suite 100

Reston, VA 22091

Phone: (703) 620-8990

EMail: vcerf@nri.reston.va.us

Lyman A. Chapin

Bolt, Beranek & Newman

Mail Stop 20/5b

150 Cambridge Park Drive

Cambridge, MA 02140

Phone: (617) 873-3133

EMail: lyman@BBN.COM

Robert Braden

USC/Information Sciences Institute

4676 Admiralty Way

Marina del Rey, CA 90292

Phone: (310) 822-1511

EMail: braden@isi.edu

Russell Hobby

University of California

Computing Services

Davis, CA 95616

Phone: (916) 752-0236

EMail: rdhobby@ucdavis.edu

RFC 1454 下一版本IP提案的比较

网络工作组

RFC : 1454

T.Dixon

RARE

1993年5月

提示

本文为Internet社区提供信息，不指定 Internet 标准，它的分发不受限制。

摘要

本文是经过少许编辑后重印的 RARE 技术报告(RTC(93)004)。

下面是当前 IP 的三个主要替代提案的特点的简短总结。本文并不打算作为详尽的或最终的文本(最后给出简要的参考文献目录以提供更多的信息源)，但可作为讨论这些提案时的参考，由 RARE 和 RIPE 来协调。应该承认这些提案本身是“推动目标”的，它反映了在华盛顿举行的第 25 届 IETF 会议的意见是完全正确的。Ross Callon 和 Paul Tsuchiya 对原始草案的评议也包括在内。有一个时期，术语 IPv7 用来指 IP 的下一个版本，但该术语与一个特别提案有关，所以现在用术语 IPng 来标识下一代 IP。

在个别讨论提案前，本文先对为解决问题和达到特定目标的机制作一般性的讨论。

1. 为何当前的IP能力不足？

该问题已经由 ROAD 小组研究并阐述过，此处简述如下：

- IP B 类地址空间耗尽。
- IP 地址空间会全部耗尽。
- 地址分配的非分级结构导致平面的选路空间。

虽然 IESG 对于新的 IP 要求比简单选路和寻址议题更深入一步，但正是这些议题使扩展当前协议成为不实际的选择。因而，对提出的各种协议的大部分讨论和开发集中在这些专门问题上。

对这些问题的近期补救办法，包括使用 CIDR 提案(CIDR 允许以 C 类网络的集聚来选路)以及以发挥 CIDR 优势的方式分配 C 类网络地址的分配策略。支持 CIDR 的选路协议有 OSPF 和 BGP4。以上这些都不是新 IP(IPng)必须具备的先决条件，但是必须延长当前 Internet 的生存期，以满足长期解决方案工作的要求。Ross Callon 指出为延长 IP 生存期有其他选择，他的一些想法已被列在 TUBA 清单中。正在考虑可使 Internet 进一步增长的长期提案。这些提案的时间进度如下：

- 12月15日提出作为 RFC 的议题选择准则。
- 2月12日两个可互操作的实施就绪。
- 2月26日第二个提案的草案文件就绪。

有雄心的目标是在 1993 年 3 月在哥伦比亚举行的第 26 届 IETF 会议上能作出提案贯彻应用的决定。

当前可选的候选对象有：

- PIP(P IP——一个全新的协议)。
- TUBA(具有大地址的TCP/UDP——用ISO CLNP)。
- SIP(简单IP——具有较大地址和较少选项的IP)。

Robert Ullman有一个更好的提案，不过我对它了解不多。与每个提案候选对象相联系的是过渡计划，但大都独立于提案本身且包含的元素可分开采用，即使对 IPv4，也还要延长当前的设备和系统的生存期。

2. 提案具有的共同点

2.1 较长的地址

所有的提案都为较长的地址字段做了准备，不仅增加了可寻址系统的数量，也方便了路由集聚的地址分级分配。

2.2 基本原理

提案也起源于世界性的“选路实现”观点——也就是说集中在网络内的选路内部部件而不是集中在终端用户或应用看得见的网络服务上。这或许是不可避免的，尤其是给生产可互操作的设备的时间非常紧。然而在第25届IETF会议上少数真正的用户代表显然不高兴，因为他们支持最终必须采用新的主机设备。

提案中有一个内置的假设，就是 IPng企图成为一个环球协议：也就是同一网络层协议将可用在同一局域网上的主机之间、主机和路由器之间、同一自治域的路由器之间和不同自治域的路由器之间。在定义分开的“接入”协议和“远程”协议上有某些优点，这在需求中没有排除。尽管这是Internet内少有的重要变革机会，但要求加速开发和低风险导致提案数不断递增，而不是从根本上变革到经过很好验证的已有的技术上。

一个未进一步陈述的假设是体系结构的目标定在单个连接的主机。目前，要设计允许主机有多个接口，并和单个连接的主机相比，可从增加的带宽和可靠性中获益（是地址属于接口而不属于主机的缘故）的IPv4网络很困难。正如这些文件中提到的，倾向于拓扑是否存在限制。已经认定不一定是PIP或TUBA提案的制约，但是相信这是一个议题，到现在为止还没有出现在相当的准则中。

2.3 源选路

已有的IPv4对源指定路由已有保证，然而很少用，（有人要反驳我吗？）部分原因是由于需要了解直至路由器级的网络的内部结构。源路由通常是需要使用的，当用户根据策略，要求源和目的地之间的业务倾向或强令通过特殊行政管理域时，源路由也可被行政管理域内的路由器用来指定通过特殊的逻辑拓扑。源指定的选路需要一些性质不同的部件：

- (1) 根据技术规范中源的策略来选择路由。
- (2) 路由的选择要与其策略相适应。
- (3) 用已标识的路由对业务流做标记。
- (4) 为已加标记的业务流相应地选路由。

这些步骤不是完全独立的。在这种方法中，第 (3) 步标识的路由可能会约束前面步骤中能被选择的路由种类。目的地不可避免地、或者通过告知准备接受的策略，或者通过一个协商过程，加入到源路由的技术规范中去。

所有提案都是通过在每个包中加一串直接地址 (或许部分地指定) 来标记源路由。没有规定一个主机取得指定这些直接地址所需信息的过程 (这个阶段不完全不合理，但期望有更多的信息)。这些决定的负面后果是：

- 根据必须指定的直接地址数，包头会变得很长 (虽然当前有指定的机构或想象该机构只指定直接地址的重要部分)。
- 如果个别的直接地址不再可达到，源路由可能必须周期性地重新指定。

正面影响是：

- 域间路由器不必了解策略，只是机械地跟随源路由。
- 路由器不必存储标识路由的上下文，因为信息被指定在每个包头中。
- 路由服务器可定位在网络的任何地方，只要主机知道如何找到它们。

2.4 封装

封装是将一个网络层包封装到另一个包中，以使有效的包能直接通过一条路径，否则就不能到达能移去最外面包的路由器，并指引结果包到它的目的地。封装需要：

- (1) 在包中有一指示位，以指示它包含另一个包。
- (2) 路由器具备这样的功能，它能在收到一个包后，移去封装并再启动包转发进程。

所有提案都支持封装。由源进行的合适的封装可能会获得源选路的效果。

2.5 组播

所有提案都能协调在地址规范许可的多种范围的组播。Internet 范围的组播尚待进一步研究。

2.6 分段

所有提案都支持中间路由器对包的分段，然而最近的一些讨论，主张从提案中取消该机制，而改为使用 MTU 发现过程以避免中间分段。这样的决定实际上排除在网络上使用报文计数字列编号的传输协议 (如 OSI 传输协议)，只有用字节计数并确认的协议 (如 TCP) 才能在一个连接激活期间处理 MTU 还原。OSI 传输协议可能不会特别地与 IP 界有关联，但是它可能与提供多协议服务的供应商有关联，但是应该注意到对于 IPng 支持的服务类型的影响。

2.7 包的生存期

IPv4 中的“生存期”(TTL) 字段在每种情况下，作为一个简单的段计数被重新计算，很大程度上以实施方便为基础。虽然老的 TTL 很大程度上以这种方式实现，但它以服务于体系结构为目的，在网络中为一个包的生存期设置了一个上限。如果该字段作为一个跳计数而重新计算，那么必定对网络中包的最大生存期有其他的技术规范，所以源主机能保证网络层分段标识符和传输层序列号，当存在混淆危险时，从来不会有重用的危险。事实上，有三个分开

的议题：

- (1) 防止选路形成回路(由跳计数解决)。
- (2) 限制网络层包的生存期(需要,但目前为止未指定),支持传输层的设想。
- (3) 允许源对包设置更多限制(例如在拥塞情况下丢弃老的实时业务流,让位给新的业务流,这是一个选项,到目前为止还没作规定)。

3. 提案略为提及的内容

3.1 资源预留

应用日益要求确定的带宽和传输延时,两者对实时视频和音频传送都是必须的。这样的应用需要过程向网络指出它们的需求以及必要的资源预留。这样的过程在某种程度上类似于源路由选择。

- (1) 源提出需求的技术规范。
- (2) 确认需求能被满足。
- (3) 用需求来标记业务流。
- (4) 为已加标记的业务流相应地选路由。

按照同样资源需求规定路线发送的业务流有时也称其为流。流的标识需要一个建立过程,人们可能设想与建立源路由使用同样的过程,但两者是有区别的,表现在:

- 在一条路径上的所有路由器必须同意并参予资源预留。
- 由此在每个路由器中相对直接地保持前后关系和短的流标识。
- 在失效时,网络可选择重定路由。

每个提案用各种方法来携带流标识,然而这是目前十分超前的研究。没有确定建立机制。实际上预留资源过程是一个高层次的问题。源选路和资源预留间的交互作用,还需进一步试验:虽然两者性质截然不同,实现的制约也不一样,但两个不同的机制将使得在选择路由时,既要满足策略,又要满足性能指标,变得困难。

3.2 地址分配策略

在IPv4中,地址与系统捆绑在一起是长期的。且在多数情况下,能与DNS名字互相交换使用。默许地接受地址和一个特殊系统的联系,在IPng中可能更为短暂。提案之一的PIP是使系统的标识和它的地址之间有区别,并允许捆绑能暂时改变。没有提案规定地址生存期的限制,也未规定地址分配方式必须受特殊协议的约束。例如,由IPng提供的较大的地址空间中分配分级地址的高位部分,可以选择是根据与地理位置相关方式,还是参照服务供应商方式。基于地理位置的地址是不变的,也易于分配,但意味着在分配区域内有重新退化到“平面”地址的危险,除非采取确实的拓扑上的限制。基于供应商的地址分配会造成地址改变(如果供应商改变)或多个地址(如果多个供应商)。移动主机(依赖于基础技术)不论是基于地理位置还是基于供应商方案都会出现问题。

对地址分配方案以及对地址生存期的影响没有严密的提案,假设捆绑名字到地址的已有的DNS模型仍然有效是不可能的。

值得提出的是,在地址分配机制和可能采用的自动配置方法之间有交互作用。

3.3 自动配置

对当前 IP 服务用户最大的担忧是维护基本配置信息的管理工作，诸如为主机分配名字和地址，并要保证信息正确地反映在 DNS 中。部分问题是由不良的实施造成的（或者盲目相信 vi 和 awk 是网络管理工具）。不过许多问题通过使过程更自动化而得到减轻。这些可能性（有些是互斥的）有：

- 分配主机地址使用相对恒定的值，如 LAN 地址。
- 在子网内定义一个动态地址分配协议。
- 定义“通用地址”，通过使用它，主机不需预先配置便可达本地服务器（DNS、路由服务器等）。
- 通过检索 DNS，主机便能确定它们的名字。
- 当主机配置改变时，由主机更新它们的名字/地址捆绑。

当很多提案提及某些以上的可能性时，选择合适的解决方案在一定程度上依赖于地址分配策略。同时，动态配置引起某些困难的理性和实际议题（确切地说，地址的作用是什么？从什么意义上来讲，当一个主机地址改变时，还是同一个主机吗？如何处理 DNS 映射的动态变化，又如何对它们进行身份验证）。

提案小组会发现大部分问题在他们讨论范围以外。当定义和选择 IPng 的候选者时，像“系统”这样的议题没有很好的讨论，看来是一种疏忽。参加者意识到了这一点，看来即使做了决定，某些观念还会在更多的读者范围内重新研究。

然而 IP 不可能在非技术环境中对有专利权的连网系统（如 Netware、AppleTalk）产生影响，在体系结构中或供应商都没有严格地采用自动配置。我坚信在人们头脑中对如何解决这些议题有想法，只是没有写在纸上而已。

3.4 应用接口/应用协议改变

一些公共应用协议（如 FTP、RPC 等）已经确定专门传送 32 位 IP 地址，无疑还有其他标准的和专用的协议。也有许多应用简单地把 IP 地址当成 32 位整数来处理。甚至用 BSD 套接字试图不透明地处理地址的一些应用，也不明白如何分析语法或打印长地址（即使套接字结构大到足够容纳它们）。

因此，每个提案需要指定机制，以便当变换发生时，允许已存在的应用程序和接口运行在新的环境下。对于 TCP 和 IPng（也能运行 IPv4），有一个程序设计接口参考技术规范是有用的，它允许开发者现在就开始改变应用程序。从指定过渡机制的所有提案中，就能推断出已存在的应用兼容性。现在还没有迹象表示一个新的接口技术规范独立于所选的协议。

3.5 DNS 改变

显然，必须要有能支持新的、长地址的名字到地址的映射服务。所有提案都认为这种服务应该由具有合适定义的新资源记录的 DNS 来提供。关于为响应某种查询，用返回“A”记录信息的合适性，以及什么信息该首先请求的讨论正在进行。在为建立正确地址所必须的询问次数和由于返回非期望信息破坏已存在的执行过程之间存在着折衷。

为自动配置使用 DNS 和寻址方案反向转换的规模的讨论很热烈，但没有实质性进展。

4. 提案中没有真正提到的

4.1 拥塞避免

IPv4中路由器用“源抑制”控制报文，向源指示拥塞并有可能不久会丢失包。TUBA/PIP有一“遭受拥塞”位，它给目的地提供类似的信息。但这些技术规范都没有提供如何使用这些设施的详细说明。因而近年来有许多研究分析实体，他们建议这样的设施不仅可以用来报告拥塞(向传输协议提供信息)，也可减少通过网络层的延迟。每种提案都提供某种形式的拥塞信令，但没有一个指定它使用的机制(或分析该机制实际上是否可用)。

作为网络服务的用户，目前大约有30%的丢弃率，且仅在500英里内来回时间就多至2秒。我对某些提案感兴趣的是网络服务在额外负载的情况下性能下降得很少。

4.2 移动主机

移动主机的一个特征是相对快地移动它们的物理位置和到网络拓扑的连接点。显然对寻址和选路是重要的(不管是地理的还是拓扑的)。到目前为止，没有解决方案的详细技术规范，看来这是一个认识问题。

4.3 计费

IESG的选择准则只要求提案不会阻止为审计和收费目的而进行的信息收集，因此没有提案考虑潜在的计费机制。

4.4 安全性

“网络层安全性议题有待进一步研究”，最好每个候选者都能扩展以显示能提供一定程度的安全性，例如对抗地址欺骗。当资源分配特性允许某些主机为特殊应用要求大量可用带宽时，这一点特别重要。

值得提出的是，提供某种程度的安全性意味着在网络内人工配置安全信息，还必须考虑与自动配置目标的关系。

5. 提案的不同点

每个提案互不相同，正像不同于IPv4一样，原理差别虽小，但会产生重大影响(地址规模的扩充，原理上仅是一个小的差别)。主要的特性差别是：

- PIP

PIP有一个创新的头格式，从而简化了分级、策略和虚电路选路。头中也有一个“含糊”的字段，它的语义在不同的行政管理域可以有不同的定义，它的使用和解释在穿越边界时协商解决，还没有指定控制协议。

- SIP

SIP提供了“最小者”方法——从IPv4头中移去所有不常用的字段，并将地址长度扩展到64位。控制协议基于对ICMP的修改。该提案有处理效率高和易于熟悉的优点。

- TUBA

TUBA基于CLNP(ISO 8473)和ES-IS(ISO 9542)控制协议。TUBA是考虑为了TCP和UDP能在CLNP网络上运行。倾向于TUBA的主要论点是认为能处理网络层协议的路由器已经存在,可扩展的地址提供了宽范围的可供“未来验证”的余量,同时是一个标准和产品会聚的机会。

5.1 PIP

PIP包头包含了一个指令集,供路由器中的转发处理器完成对包的某些动作。在传统协议中,某些字段的内容隐含某些动作。PIP为源端编写指引包通过网络选路的小“程序”提供了灵活性。

PIP地址长度实际上不受限制:网络拓扑分级的每一级成为地址的一部分,同时地址随网络拓扑改变而改变。在完全分级的网络拓扑中,每级所需的选路信息数量可以非常小。因而在实际上,分级的级数将更多地由商界和实用因素来决定,而不是受任何特定的选路协议的制约。一个明显优点是地址的高位部分在本地交换时可以省略,低位部分在源路由中可以省略,减少了主机系统需要知道的拓扑信息数量。

这里有一个假设,就是 PIP地址易于改变,所以为了标识,给系统指定另一个参数 PIP标识符。不清楚该参数有何用途,它不能同等地受到 DNS名字的服务(更加紧凑,但同样不需要携带在每个包中,但需要一个额外的检索)。因此,提出了这样的问题:两个潜在可通信的主机系统如何找到可使用的正确的地址。

PIP最复杂的部分莫过于某些头字段的意义是由特定域中相互之间的合同来确定的。专门处理设施的语义(如排队优先级)是全球登记的,但实际使用和包头中为这些设施申请的编码在不同的域中可以是不同的。在两个域之间用不同编码的边界路由器必须从一种编码映射成另一种。因为路由器和其他域在物理上不一定是相邻的,而是通过“隧道”,因此一个路由器必须了解的潜在编码规则数十分大。相对于更熟悉的“选项”而言,虽然用这样的方案可以节省包头的空间,但是协商这些设施使用和编码的复杂性导致成本增加,以及在每个域边界上对包的再编码,这些才是关心的主题。虽然主机为它们的本地域有可能“预编译”编码规则,还是存在许多潜在的实施上的困难。

虽然PIP在三个提案中提供了最大的灵活性,但对于“希望可用”的情况还需进行更多工作,使其潜在的优点和缺点能暴露得更具体。

5.2 SIP

SIP是一个简单而具有较大地址和较少选项的 IP。它的主要优点是甚至比 IPv4更容易处理。它的主要缺点是:

- 如果32位地址不够的话,那么 64位地址在可预见的未来是否就够了,还远远不清楚。
- 虽然在头字段中有少量“保留”位,但 SIP支持新特性的扩展不明显。真是没有其他什么可说的!

5.3 TUBA

ISO CLNS的特征相当有名,协议与 IPv4有很强的文化上的相似性,然而有 20字节供网络层寻址。除了谬误的(不是这儿发明的)偏见之外,反对 TUBA的主要争论在于 TUBA太像 IPv4了。除了更大、更灵活的地址外,别无其他贡献。采样试验证明路由器能高效地处理非常长

的地址,但同时长的头很少不给网络带宽带来负面影响。

对建议的控制协议(ISO 9542)有下列异议:

- 根据我以前的经验,如果要合理地容纳大的局域网,路由器发现主机的过程将是低效的,而且会消耗路由器资源,同时在主机上需要十分精确的时间分辨力。TUBA支持者建议,根据最近的经验,ARP不适用了,但是我想本议题还需要检验。
- 重定向机制实际上是基于LAN地址,而不是网络地址,意思是本地路由器将复合的选路决定交给同一LAN上其他路由器。同样,重定向方案(如IPv4中的方案)重定向到网络地址会造成不必要的额外跳数。要分析哪个解决方案比较好,依赖于构筑的情况。客观地说,该协议的路由器发现部分提供了一个其他提案所没有的机制。通过该机制,主机能定位就近的网关,并能自动配置它们的地址。

6. 过渡计划

为使“老”主机能与“新”主机对话,显然需要一个过渡:

- (1) IPng主机也能用IPv4,或
- (2) 通过一个中间系统转换。

或者:

- (1) 系统间的基础设施有能力承载IPng和IPv4,或
- (2) 网络的某些部分用隧道或转换方法将一个协议附在另一个协议中。

各种提案拥护的过渡计划只是简单地将上述方案进行组合。经验表明,不管选择那个协议,事实上以上情况都会发生。

隧道/转换过程的一个问题是必须携带在穿过网络中的IPv4隧道时的附加信息(外加地址部分),这可以在数据封装在IPv4包前加一个附加的“头”来实现,或者通过将信息编码作为新IPv4选项类型来实现。

在前一种情况下,可能要正确地映射出错报文会有困难,因为原始包在返回前被截取;后一种情况,包有被丢弃的危险(因为IPv4选项不是自描述的,新的选项可能无法通过IPv4路由器)。这就是为了支持IPng隧道方法而引入IPv4的“新”版本的理由。

另一个替代方案(在该方案中,IPng主机有两个栈,基础设施可以支持、也可以不支持IPng或IPv4)当然需要一个机制来解决用哪个协议做试验。

7. 随意评议

这是Internet协议中发生的首次根本性改变。因为Internet是一个可管理的实体,它的发展是与美国政府合同紧密联系的。或许IETF/IESG/IAB组织结构不可避免地无法管理如此大幅度的改变,但希望提议的新结构在促进共识上获得更大成功。值得注意的是许多觉察到的OSI过程问题(如进步慢,在琐事上派别内争,聚焦在最低层共同特性解决方案上,缺乏对终端用户的考虑等),用它们来处理IPng是危险的,同时关注着由网络设计的广泛参与所带来的困难会到什么程度。

三个主要提案在IPng上很少有实质性的差别,但选择IPng的竞争过程如不成功就是失败。在这方面,选择过程的结果没有什么特别意义,但在过程本身中为了修复Internet工程过程的社会和技术凝聚力,或许是必要的。

8. 更多的信息

提案的主要讨论清单如下：

TUBA: tuba@lanl.gov
PIP: pip@thumper.bellcore.com
SIP: sip@caldera.usc.edu
General: big-internet@munari.oz.au

(Requests to: <list name>-request@<host>)

各种提案的Internet草案和RFC，仍能在惯常场合找到。

安全性考虑

安全性议题未在本文中讨论。

作者地址

Tim Dixon
RARE Secretariat
Singel 466-468
NL-1017AW Amsterdam
(Netherlands)

Phone: +31 20 639 1131 or + 44 91 232 0936
EMail: dixon@rare.nl or Tim.Dixon@newcastle.ac.uk

RFC 1671 向IPng过渡和其他考虑的白皮书

网络工作组

RFC : 1671

类别 : 信息类

B.Carpenter

CERN

1994年8月

提示

本文为Internet社区提供信息，不指定任何种类的Internet标准。本文的分发不受限制。

摘要

本文是响应RFC 1550而向IETF IPng领域提交的文件。本文的发布并不意味着IPng领域接受文中所表达的任何思想。评议请提交给big-internet@munari.oz.au邮件列表。

总结

本白皮书在所选领域勾画了IPng某些通用需求。下面表示的是逐级过渡的需求：

- (1) 在网络的每级和每层实现互通。
- (2) 包头转换被认为是有害的。
- (3) 共存。
- (4) IPv4到IPng地址映射。
- (5) 双栈主机。
- (6) 域名系统(DNS)。
- (7) 智能双栈代码。
- (8) 智能管理工具。

接受某些关于物理和逻辑组播的论点，并建议需要一个IPng在ATM上运行的模型。最后，本文建议的策略选路、计费和安全防火墙等需求，需要所有IPng包携带所涉及事务处理类型的踪迹，以及它们的源和目的地址。

过渡和发展

显然过渡需要几年的时间，同时网络中的每个站点不得不决定它自己的阶段过渡计划。只有那些最小的站点可能在ISP的压力下，考虑一步到位（“标志日”）的过渡。此外，一旦决定采用IPng，那么Internet和所有用Internet协议集的专用网在下一十年（或更长）的活动，将受到IPng发展的强大影响。用户站点注视着决策，是否和他们过去所看到的在改变程序设计语言或操作系统时所用的同样方法来改变IPv4。向IPng转变可能不是必然的结果。他们主要担心是，改变是否能使成本和影响生产的风险减到最低。

这样的担心立刻对IPng过渡和发展的模型产生了强大的约束。这些约束中的某些列在下面，并对每种约束赋予简短的解释。

术语“IPv4主机”是一个和今天的主机运行同样内容，而没有维护版本及配置改变的主机。“IPng主机”是一个运行IP新版本，经过重新配置的主机。它类似于路由器。

1. 网络的每级和每层实现互通

这是主要约束。计算机系统、路由器和应用软件厂商肯定不会协调他们产品的发布日期。用户将继续运行他们的老设备和软件。因此，IPv4和IPng主机和路由器的任何组合必须能互通(即加入到UDP和TCP会话中)。一个IPv4包必须能找到从任何IPv4主机到其他任何IPv4或IPng主机的路径，反之亦然，穿过IPv4和IPng路由器的混合路径，IPv4主机无需修改。IPv4路由器无需修改可与IPng路由器互通。另外，一个“明白”IPv4但还“不明白”IPng的应用软件包必须能在运行IPv4的计算机系统上运行，并和IPng主机通信。例如，欧洲的一个老PC机应该可访问美国的NIC服务器，即使NIC服务器运行的是IPng，且北美的选路机制只是部分地变换过。或者某个公司某个部门的一个C类网络应该保持对运行IPng的公司服务器完全的访问，尽管C类网络内部什么也没有改变。

(并不要求一个只能在IPv4上运行的应用程序到一个IPng主机上运行。因此，我们承认某些主机一直要等到所有它们的应用程序是IPng兼容后才能升级。换句话说，我们承认某种程度上API要改变。然而，即使这样的放松，还是有争议的，甚至有些厂商要求在IPng主机上严格保持IPv4 API。)

2. 包头转换被认为是有害的

该作者相信在任何过渡情况下，要求IPv4和IPng间动态包头转换将会造成几乎是不可克服的实际困难：

(1) 可以认为IPng功能将是IPv4功能的一个超集。然而，协议间的成功转换要求被转换的两个协议的功能事实上应该相同。为此，应用需要知道它们什么时候通过IPng API和设在网络中某处的转换器与IPv4主机互通，以便只用IPv4功能。这是不现实的约束。

(2) 转换器的管理对大的站点而言是完全行不通的，除非转换机制是完全隐蔽和自动的。特别是任何转换机制要求为每个主机中的表格(如DNS表或路由器表)人工地保持专门标志以指示需要转换，这样做是完全不可能进行管理的。在一个有几千台运行多种操作系统的主机的站点上，主机在不同软件版本上前进或后退，使得继续用这种方法来不断跟踪所需要的这些标志的状态是不可能的。整个Internet的多样化，将会导致混乱、复合的失效模式和困难的诊断。特别是不可能遵守(1)的约束。

实践中为了避免混乱，对转换所需要的知识、所涉及的站点将决不泄漏，并且如果还没有有这样的知识的话，当需要时，应用程序不能将其本身限制在IPv4功能上。

为了避免混淆，此处所讨论的包头转换和地址转换(NAT)不是同一件事。本文不讨论地址转换。

本文不详细处理性能议题，但转换的另一个明显缺点是带来必然的开销。

3. 共存

Internet基础设施(不论是公用的还是专用的)必须允许IPv4和IPng在同一路由器和同一物理路径上共存。

为了在不要求主机步调一致地更新，及不使用转换器的情况下，网络基础设施能更新至IPng，共存是必须的。

值得注意的是，这种需求并不强制使用有关公共的或是分离的方法来进行选路。作为共存机制，也不排斥使用封装。

4. IPv4到IPng地址映射

人们必须明白过渡期间会遇到什么问题。虽然IPng地址的自动配置可能是所期望的目的，如

果在给定的站点上,IPv4和IPng地址之间有一个可选的简单映射,那么过渡的管理就会大大简化。

因此,IPng地址空间应包含IPv4地址的映射,这样(如果一个站点或服务供应商愿意做的话)一个系统的IPv4地址能机械地被转换成IPng地址,大多数倾向于加一个前缀。对每个站点而言,前缀不一定是相同的,可能至少是服务供应商指定的。

这并不意味着这种地址映射会用作动态转换(虽然有可能是),或将IPv4选路嵌入IPng选路内(虽然有可能是)。主要目的是简化网络运行者的过渡规划。

顺便指出,这样的需求实际上没有假设IPv4地址是全球唯一的。在建立IPv4和IPng选路域与分级之间的关系上也没有太多帮助。没有理由设想它们之间是1:1对应关系。

5. 双栈主机

无转换的逐步过渡是很难想象的,除非大部分主机同时能运行IPng和IPv4。如果A想和B(IPng主机)以及和C(IPv4主机)交谈,于是A或者B必须能运行IPv4和IPng两者。换句话说,所有运行IPng主机必须仍能运行IPv4。只能运行IPng的主机在过渡期间是不允许的。

这样的需求并不意味着IPng主机真的有两种完全分离的IP实现(双栈和双API),但是表现出好像是分离的。封装是兼容的(即两个栈中的一个可为另一个封装包)。

显然,对双栈主机的管理,由于上面提到的地址映射而简化了。除了IPv4地址以外,只有站点前缀必须配置(人工或动态地)。

在双栈主机中,即使IPng API和IPv4 API是作为一个单个实体实现的,但在逻辑上是可区别的。应用程序将从API得知它们在用IPng还是IPv4。

6. DNS

双栈要求隐含了DNS必须给IPng主机回答IPv4和IPng两者的地址,或将两者编码在一起的单个回答。

如果在DNS中,一个主机附属于一个IPng地址,但该主机实际上还没有运行IPng,犹如在IPng空间中出现一个黑洞——见下一点。

7. 智能双栈代码

双栈代码可从DNS得到两个地址,用哪一个呢?多年的过渡期间Internet将包含黑洞。例如,从IPng主机A到IPng主机B路上某处有时(不可预测)会遇到只运行IPv4的路由器,它将丢弃IPng包。同样,DNS的状态也不一定与现实是一致的。DNS声称知道IPng地址的主机可能在一特别的时刻并没有运行IPng,因此到那个主机的IPng包在传递时将被丢弃。知道一个主机具有IPv4和IPng两种地址并没有给出有关黑洞的信息。对此必须有一种解决方案,这方案不能依赖于人工保持的信息。(如果这个不解决,双栈方法是不会好于转换方法的。)

8. 智能管理工具

过渡期间需要一整套管理工具。为什么IPng路由不同于IPv4路由?如果要转换的话,该发生在何处?何处有黑洞?(宇宙学家喜欢同样的工具。)今天的主机是否真正有IPng能力?

组播

众所周知,IPng必须支持组播应用,体系结构上一个明显的规则是:不论是LAN还是WAN线路,组播包不应在同一线路上经过两次。如果做不到这点,则意味着同时组播的事务处理最大数量会减半。

LAN上的IPv4的一个负面特征是:轻率地使用物理广播包,诸如ARP(各种非IETF盲目模

仿者)协议。在大的 LAN 上,这将导致一系列不希望有的后果(经常是由于差的产品或差的用户,而不是协议设计本身造成的。)如有可能的话,体系结构上明显的规则是改用单播(或最坏情况,用组播)来代替物理广播。

ATM

网络工业界正在 ATM 上大量投资。没有 IPng 提案似乎是可取的(从获得管理部门批准的意见上),除非它是“ATM 兼容”的,也就是要有一个如何运行在 ATM 网络上清晰的模型。虽然不马上需要一个像 RFC 1577 那样十分详细的文件,但必须显示该基本模型是能工作的。

类似的论点同样可用于 X.25、帧中继、SMDS 等,但 ATM 是当前呼声最高的。

策略选路与计费

遗憾的是,这不能被忽略,许多人对此感兴趣。基金代理希望业务流流经提供基金的线路,且在以后他们要知道有多少业务流。计费信息也可用作网络规划和反向付费的根据。

所以 IPng 及它的选路过程允许根据详细的源和目的地址来指定业务流的途径。(作为一个例子,从 MIT 物理系输出的业务流和任何其他系输出的业务流可能通过不同的路由到 CERN。)

满足该需求的一个简单途径是坚持 IPng 必须支持基于供应商的寻址和选路方案。

业务流的计费要求同样的详细程度(甚至更详细,例如 ftp 的业务流是多少, www 的业务流又是多少)。

两者都需要花费时间和金钱,并且不仅影响 IP 层,所以 IPng 不该回避它们。

安全性考虑

公司网络运行者和校园网络运行者曾受到过好几次安全问题的困扰,他们对待此事比许多协议专家更认真。实际上,许多公司网络运行者希望在向 IPng 过渡中,作为一个比其他任何议题更为紧迫的议题,安全性能得以改善。

因为 IPng 估计是一个数据报协议,限制了它为端到端的安全性所能做的工作, IPng 必须允许路由器中有比 IPv4 更有效的防火墙。特别需要基于源和目的地址以及事务处理类型的高效的业务流阻挡。

看来需要同样的特征允许策略选路和详细计费来改善防火墙的安全性。讨论这些特征的细节超出了本文范围,但是看来不大可能在边界路由器中限制实现细节。为了检查有害的业务流,允许基于策略的源选路和/或允许详细计费,包必须携带某些三重验证的踪迹(源、目的地、事务处理)。可能所有 IPng 可以在每个包中以某种格式携带源和目的地的标识符,但是标识事务处理的类型或甚至于个别的事务处理,是一个额外需求。

声明和致谢

以下是个人观点,未必代表我老板的观点。

近几年来, CERN 已经通过三个网络的过渡(由 John Gamble 解决的 IPv4 重编号, 由 Mike Gerard 解决的 AppleTalk 从阶段 I 向阶段 II 的过渡, 以及由 Denise Heagerty 解决的 DECnet 阶段 IV 向 DECnet/OSI 选路的过渡)。如果没有从他们那儿获取知识,我是不能写出这个文件的。我也

从许多人，特别是从 IPng董事会的各个成员的讨论或作品中，获益非浅。多位董事会成员及波音公司的Bruce L Hutfless提出了意义帮助阐明本文。不过意见是我本人的，并非董事会全体成员的共同意见。

作者地址

Brian E. Carpenter
Group Leader, Communications Systems
Computing and Networks Division
CERN
European Laboratory for Particle Physics
1211 Geneva 23, Switzerland

Phone: +41 22 767-4957
Fax: +41 22 767-7155
Telex: 419000 cer ch
Email: brian@dccw.cern.ch

RFC 1715 地址分配效率比例系数H

网络工作组

C.Huitema

RFC : 1715

INRIA

类别 : 信息类

1994年11月

提示

本文是为Internet社区提供信息，并不指定任何类的Internet标准。本文的分发不受限制。

摘要

本文是响应RFC 1550而向IETF IPng领域提交的文件。本文的发布并不意味着IPng领域接受文中所表达的任何思想。评议可提交给作者或发邮件至 sipp@sunroof.eng.sun.com邮件列表。

目录

- 1. 地址分配的效率 [1]
- 2. 估计合理的系数H值 [1]
- 3. 评估提出的地址计划 [2]
- 4. 安全性考虑 [2]
- 5. 作者地址 [2]

1. 地址分配的效率

IPng争论的实质性部分集中于地址长度的选择。一个重复的概念是“分配效率”，参加讨论的大部分人表示，效率是网络中有效的系统数对最大理论值之比。例如，32位的IP寻址计划理论上超过70亿个系统，而目前DNS中记录有大约3500万个地址，说明效率为0.05%。

但是这种经典评估是误导，因为它没有考虑分级的级数。例如IP地址至少分为三级：网络、子网和主机。为了排除这些相关性，建议对效率系数用一个对数尺度：

$$H = \log(\text{目标数}) / \text{可用位数}$$

系数H不至于太依赖于分级的数量。例如：设想在两级之间选择，每级用8位编码。若单级则用16位编码。如果在每个8位级平均分配100个元素，或在单个16位级平均分配10 000个元素，我们将获得同样的效率。

为便于心算，以下用的是以10为底的对数。当数变大时，人们习惯于用10的指数来表示。这样可以说“IPng能编号 1×10^{15} 个系统”。如果遵循这样的单位选择，H就在0与理论最大值0.30103($\log 2$)之间变化。

2. 估算合理的系数H值

我们并不指望在实际中获得系数值为0.3。关心的问题是推断该值为合理的期望值。我们可以试着从已有的编号计划来评估它。特别感兴趣的是考虑计划打破时，即当人们被迫对电话号码增加数字时，或者计算机地址增加位数时。我手头有若干这样的数字：

- 当电话号码数达到一个门限 1.0×10^7 时，所有法国的电话号码加1位数字，由8位加到9位数。对数值为7，位数大约为27(一个十进制数大约为3.3位)。则系数就是0.26。

- 扩展美国电话系统的区域号，使其为 10 位数，可以有 $1.0E+8$ 个用户。对数值为 8，位数为 33 位，系数大约为 0.24。
- 扩展 Internet 地址长度，从 32 位至某一值。当前 32 位网上大约有 300 万个主机。 $3.0E+6$ 的对数大约为 6.5，这样得出的系数为 0.2。我们相信 32 位还足够用好几年。如果主机数乘 10，系数升至 0.23。
- 扩展 SITA 7 号码地址的长度。按照他们的文件，在他们的网络中，大约有 64 000 个可编址的点，分散在 1200 个城市，180 个国家中。一个上限情况提供 5 位码供寻址用，造成效率为 0.14。这是一种极端情况，因为 SITA 在它的分级中用固定长度的令牌。
- 全球连通的物理 / 空间科学 DECnet 网 (阶段 IV) 到 15 000 个节点时停止增长 (即新节点隐藏)，在 16 位空间中给出系数为 0.26。
- 在 46 位空间中，大约有 2 亿个 IEEE 802 节点，给出系数为 0.18。然而，这个号码空间没有饱和。

从以上例子，可以推测出效率系数通常在 0.14 到 0.26 之间。

3. 评估提出的地址计划

用反向计算，可得到网络中寻址的设备总数：

	悲观的估计 (0.14)	乐观的估计 (0.26)
32 位	$3 E+4 (!)$	$2 E+8$
64 位	$9 E+84$	$E+16$
80 位	$1.6E+11$	$2.6 E+27$
128 位	$8 E+17$	$2 E+33$

数字对于为什么有些人认为 64 位 “不够”，而另一些人则认为 “有足够余量” 解释得很好。根据分配效率，或者远低于目标，或者远高于目标。我的观点是 128 位足足有余。甚至我们假设效率最低，仍有超过 $1.E+15$ 台 Internet 主机的冗余估计。

同时值得提出的是，如果我们给网络贡献 80 位，并为 “缺少自动配置的服务器” 提供 48 位，在悲观的情况下，仍能编号多于 $E+11$ 个网络；要达到 $E+12$ 个网络，只要取效率系数为 0.15。

这就是为什么我认为 128 位在下一 30 年中是完全安全的解释。必须包括在地址分配内的制约程度，显示出与今天我们所知道如何做的是非常一致的。

4. 安全性考虑

安全性议题不在本文中讨论。

5. 作者地址

Christian Huitema
INRIA, Sophia-Antipolis
2004 Route des Lucioles
BP 109
F-06561 Valbonne Cedex
France

Phone: +33 93 65 77 15
EMail: Christian.Huitema@MIRSA.INRIA.FR

RFC 2373 IPv6寻址体系结构

网络工作组

RFC : 2373

撤销 : 1884

分类 : 标准跟踪

R.Hinden

诺基亚公司

S.Deering

Cisco公司

提示

本文为Internet社区指定一个Internet标准跟踪协议，并请求为改进进行讨论和提出建议。对标准化状态和本协议的状况，请参考“Internet正式协议标准”(STD.1)最新版本。本文的分发不受限制。

版权声明

本文件全部版权属于The Internet Society (1998)。

摘要

本技术规范定义IPv6^[IPv6]的寻址体系结构。本文件包括IPv6寻址模型、IPv6地址的文字表示、IPv6单播地址、任意点播地址和组播地址的定义以及IPv6节点需要的地址。

目录

1. 概述	[2]
2. IPv6寻址	[2]
2.1寻址模型	[2]
2.2地址的文本表示	[3]
2.3地址前缀的文本表示	[3]
2.4地址类型表示	[4]
2.5单播地址	[5]
2.5.1接口标识符	[5]
2.5.2未指定的地址	[6]
2.5.3回返地址	[6]
2.5.4嵌有IPv4地址的IPv6地址	[6]
2.5.5NSAP地址	[7]
2.5.6IPX地址	[7]
2.5.7可集聚全球单播地址	[7]
2.5.8本地使用的IPv6单播地址	[7]
2.6任意点播地址	[8]
2.7组播地址	[9]
2.7.1预定义的组播地址.....	[10]
2.7.2新IPv6组播地址的分配.....	[10]

2.8节点需要的地址	[11]
3. 安全性考虑	[11]
附录A 创建基于EUI 的接口标识符	[11]
附录B 文本表示的ABNF描述	[13]
附录C RFC 1884的变化.....	[13]
参考文献	[14]
作者地址	[15]
版权声明	[15]

1. 概述

本技术规范定义了IPv6的寻址体系结构。包括当前定义的IPv6^[IPv6]地址格式的详细描述。

作者衷心感谢Paul Francis, Scott Bradner, Jim Bound, Brian Carpenter, Matt Crawford, Deborah Estrin, Roger Fajman, Bob Fink, Peter Ford, Bob Gilligan, Dimitry Haskin, Tom Harsch, Christian Huitema, Tony Li, Greg Minshall, Thomas Narten, Erik Nordmark, Yakov Rekhter, Bill Simpson和Sue Thomson所做的努力。

2. IPv6寻址

IPv6地址为接口和接口组指定了128位的标识符。有三种地址类型：

- 单播。一个单接口有一个标识符。发送给一个单播地址的包传递到由该地址标识的接口上。
- 任意点播。一般属于不同节点的一组接口有一个标识符。发送给一个任意点播地址的包传送到该地址标识的、根据选路协议距离度量最近的一个接口上。
- 组播。一般属于不同节点的一组接口有一个标识符。发送给一个组播地址的包传递到该地址所标识的所有接口上。

在IPv6中没有广播地址，它的功能正在被组播地址所代替。在本文中，地址内的字段给予一个规定的名字，例如“用户”。当名字后加上标识符一起使用(如“用户ID”)时，则用来表示名字字段的内容。当名字和前缀一起使用时(如“用户前缀”)则表示一直到包括本字段在内的全部地址。

在IPv6中，任何全“0”和全“1”的字段都是合法值，除非特殊地排除在外的。特别是前缀可以包含“0”值字段或以“0”为终结。

2.1 寻址模型

所有类型的IPv6地址都被分配到接口，而不是节点。一个IPv6单播地址属于单个接口。因为每个接口属于单个节点，多个接口的节点，其单播地址中的任何一个可以用作该节点的标识符。所有接口至少需要有一个链路本地单播地址(见2.8节额外需要的地址)。一个单接口可以指定任何类型的多个IPv6地址(单播、任意点播、组播)或范围。具有大于链路范围的单播地址，对这样的接口是不需要的，也就是从非邻居或者到非邻居的这些接口，不是任何IPv6包的起源或目的地。这有时适用于点到点接口。对这样的寻址模型有一个例外：

如果处理多个物理接口的实现呈现在Internet层好像一个接口的话，一个单播地址或一组单播地址可以分配给多个物理接口。这对于在多个物理接口上负载共享很有用。

目前的IPv6延伸了IPv4模型，一个子集前缀与一条链路相关联。多个子集前缀可以指定给同一链路。

2.2 地址的文本表示

用文本串表示的IPv6地址有三种规范形式：

(1) 优先选用的形式为x:x:x:x:x:x:x:x，其中x是8个16位地址段的十六进制值。

例如：

FEDC : BA98 : 7654 : 3210 : FEDC : BA98 : 7654 : 3210

1080 : 0 : 0 : 0 : 8 : 800 : 200C : 417A

个别字段中前面的0可以不写，但是每段必须至少有一位数字 ((2)中描述的情形除外)。

(2) 在分配某种形式的IPv6地址时，会发生包含长串0位的地址。为了简化包含0位地址的书写，指定了一个特殊的语法来压缩0。使用“::”符号指示有多个0值的16位组。“::”符号在一个地址中只能出现一次。该符号也能用来压缩地址中前部和尾部的0。

用下面的例子来说明：

1080:0:0:0:8:800:200C:417A

单播地址

FF01:0:0:0:0:0:0:101

组播地址

0:0:0:0:0:0:0:1

回返地址

0:0:0:0:0:0:0:0

未指定地址

可用下面的压缩格式表示：

1080::8:800:200C:417A

单播地址

FF01::101

组播地址

::1

回返地址

::

未指定地址

(3) 当谈到IPv4和IPv6节点这样一个混合环境时，有时更适合于采用另一种表示形式：

x:x:x:x:x:x:d.d.d.d,其中x是地址中6个高阶16位段的十六进制值，d是地址中4个低价8位段的十进制值(标准IPv4表示)。举例说明：

0:0:0:0:0:0:13.1.68.3

0:0:0:0:0:FFFF:129.144.52.38

写成压缩形式为：

::13.1.68.3

::FFFF.129.144.52.38

2.3 地址前缀的文本表示

IPv6地址前缀的表示方式和IPv4地址前缀在CIDR中的表示方式很相似。一个IPv6地址前缀可以表示为如下的形式：

IPv6地址/前缀长度

其中，IPv6地址是2.2节中表示的任何形式的IPv6地址。而前缀长度是组成前缀的十进制值，说明地址最左边的连续的地址位的长度。

例如，60位长的前缀12AB00000000CD3(十六进制)可用下面的合法格式来表示：

12AB:0000:0000:CD30:0000:0000:0000:0000/60

12AB::CD30:0:0:0/60

12AB:0:0:CD30::/60

但是，下面的表示方式是不合法的。

12AB:0:0:CD3/60 在任何一个16位段的地址块中，可以省略前部的0。但不能省略尾部的0。

12AB::CD30/60 /左边的地址会展开成 12AB:0000:0000:0000:0000:0000:CD30

12AB::CD3/60 /左边的地址会展开成 12AB:0000:0000:0000:0000:0000:0CD3

当书写节点地址和它的子网前缀两者时，可以组合成如下表示：

节点地址：

12AB:0:0:CD30:123:4567:89AB:CDEF

和它的子网号：

12AB:0:0:CD30::/60

可以缩写成为：

12AB:0:0:CD30:123:4567:89AB:CDEF/60

2.4 地址类型表示

一个IPv6地址的具体类型是由地址的前面几位来指定的。包含这前面几位的可变长度字段称为格式前缀(FP)。这些前缀的初始分配如下：

分 配	前缀(二进制)	占地址空间的百分率
保留	0000 0000	1/256
未分配	0000 000	11/256
为NSAP地址保留	0000 001	1/128
为IPX地址保留	0000 010	1/128
未分配	0000 011	1/128
未分配	0000	11/32
未分配	0001	1/16
可集聚全球单播地址	001	1/8
未分配	010	1/8
未分配	011	1/8
未分配	100	1/8
未分配	101	1/8
未分配	110	1/8
未分配	1110	1/16
未分配	1111 0	1/32
未分配	1111 10	1/64
未分配	1111 110	1/128
未分配	1111 1110 0	1/512
链路本地单播地址	1111 1110 10	1/1024
站点本地单播地址	1111 1110 11	1/1024
组播地址	1111 1111	1/256

注：1. 未指定地址(见2.5.2节)、回返地址(见2.5.3节)，和嵌入IPv4地址的IPv6地址(见2.5.4节)的分配在格式前缀空间0000-0000以外。

2. 除了组播地址(1111 1111)外，格式前缀空间001到111，在EUI-64格式中都要求必须有64位接口标识符。参见2.5.1节中的定义。

这样的分配方案支持可集聚地址、本地用地址和组播地址的直接分配，并有保留给 NSAP 地址和 IPX 地址的空间。其余的地址空间留给将来用。可用于已有使用的扩展（如附加可集聚地址等）或者新的用途（如将定位符和标识符分开）。地址空间的 15% 是初始分配的，其余 85% 的地址空间留作将来使用。

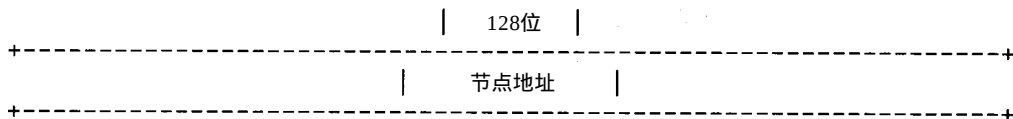
单播地址和组播地址是由地址的高阶字节值来区分的：值为 FF(1111 1111) 标识一个地址为组播地址，其他值则标识一个地址为单播地址。任意点播地址取自单播地址空间，和单播地址在语法上是无法区分的。

2.5 单播地址

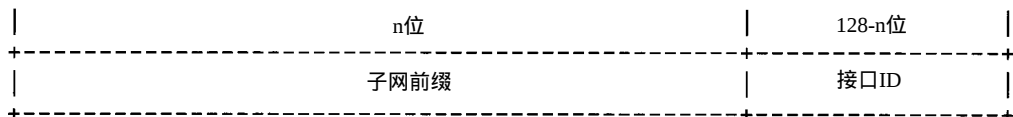
IPv6 单播地址是用连续的位掩码集聚的地址，类似于 CIDR 的 IPv4 地址。

IPv6 中的单播地址分配有多种形式，包括全部可集聚全球单播地址、NSAP 地址、IPX 分级地址、站点本地地址、链路本地地址以及运行 IPv4 的主机地址。将来还可以定义另外的地址类型。

IPv6 节点对 IPv6 地址的内部结构可能知之甚多或知之甚少，这是由节点的作用决定的（例如，主机还是路由器）。在最简单的情况下，节点把单播地址（包括它本身）看成是无内部结构的、如下图所示的 128 位地址。



一个稍完善但仍很简单的主机可能还知道它所连接的链路的子网前缀，在这种场合下，不同地址可能有不同值。



更完善的主机可能知道单播地址中其他分级边界。虽然一个非常简单的路由器可能对 IPv6 单播地址的内部结构一无所知，但为了运行选路协议，路由器对一个或多个分级边界要有更为普遍的知识。知道边界随路由器不同而不同，是由路由器在选路分级中所处的位置决定的。

2.5.1 接口标识符

在 IPv6 单播地址中接口标识符用来标识链路的接口。标识符在该链路上应是唯一的。也可能在较宽范围内是唯一的。在许多情况下，一个接口标识符与该接口的链路层地址相同。在一个单节点上，同一个接口标识符可以用在多个接口上。

在一个单节点的多个接口上，用同样的接口标识符不会影响接口标识符的全球唯一性，或由接口标识符创建的每个 IPv6 地址的全球唯一性。

在许多格式前缀中（见 2.4 节），接口标识符要求 64 位长，并构成 IEEE EUI-64 格式。基于 EUI-64 的接口标识符，当全球令牌可用时（如 IEEE 48 位 MAC），具有全球范围的意义。当全球令牌不可用时（如串行链路、隧道终点等），则只具有本地范围的意义。当由 EUI-64 形成接口标识符时，若 u 位（IEEE EUI-64 术语中称全球/本地位）置 1，则表示全球范围；若 u 位置 0，则表示本地范围。一个 EUI-64 标识符的头三个字节的二进制表示如下所示。

0	0 0	1 1	2
0	7 8	5 6	3
+-----+-----+-----+-----+			
cccc ccug cccc cccc cccc cccc			
+-----+-----+-----+-----+			

按Internet标准中的位序，其中u是全球/本地位，g是个体/团体位，c是公司标识符。“附录A 创建基于EUI-64接口标识符”为不同的基于EUI-64接口标识符的创建提供了实例。

当形成接口标识符时，使用u位的动机是当硬件令牌不可用，即在串行链路、隧道终点等情况下，便于系统管理员人工配置本地范围标识符。另一种方法是用0200:0:0:1、0200:0:0:2等形式代替十分简单的::1、::2等形式。

在IEEE EUI-64标识符中使用全球/本地位的目的是为了将来技术的发展能利用具有全球范围的接口标识符所带来的好处。

形成接口标识符的细节定义在IP over<link>技术规范中，诸如IP over Ethernet^[ETHER]、IP over FDDI^[FDDI]等。

2.5.2 未指定地址

地址0:0:0:0:0:0:0:0称为未指定地址。它不能分配给任何节点。意思是没有这个地址。它的一个应用示例是初始化主机时，在主机未取得自己的地址以前，可在它发送的任何IPv6包的源地址字段放上未指定地址。

未指定地址不能在IPv6包中用作目的地址，也不能用在IPv6选路头中。

2.5.3 回返地址

单播地址0:0:0:0:0:0:0:1称为回返地址。节点用它来向自身发送IPv6包。它不能分配给任何物理接口。可以设想它正在与一个虚拟接口相关联(如回返接口)。

发送到单节点外的IPv6包回返地址必须用作源地址。具有一个目的地址为回返地址的包不应发出单节点之外，IPv6路由器也不会转发这样的包。

2.5.4 嵌有IPv4地址的IPv6地址

IPv6过渡机制^[TRAN]包括一种技术，使主机和路由器能在IPv4选路基础设施上动态地以隧道方法传送IPv6包。使用该技术的IPv6节点要指定特殊的IPv6单播地址，它在低阶32位上携带IPv4地址。这种地址类型称其为“与IPv4兼容的IPv6地址”，并具有下面的格式：

	80位	16	32位	
+-----+-----+-----+-----+-----+				
0000.....0000 0000	IPv4地址			
+-----+-----+-----+-----+-----+				

第二种类型的IPv6地址嵌有IPv4地址。该地址用来表示只支持IPv4，而不支持IPv6的节点的IPv6地址。这种地址类型称为“与IPv4映射的IPv6地址”，并具有下面的格式：

	80位	16	32位
+-----+-----+-----+-----+-----+			
0000.....0000 FFFF	IPv4地址		
+-----+-----+-----+-----+-----+			

2.5.5 NSAP地址

NSAP地址到IPv6地址的映射定义在[NSAP]中。对于已经规划或应用OSI NSAP寻址计划，并希望应用IPv6或向IPv6过渡的网络实现者，该文件应该重新设计成IPv6寻址计划来满足他们的需要。另外还定义了一套机制，用来在IPv6网络中支持OSI NSAP寻址。如果需要这种支持的话，则必须要有这样的机制。该文件还定义了OSI地址格式内IPv6地址的映射，这应该是必需的。

2.5.6 IPX地址

IPX地址到IPv6地址的映射表示如下：

7	121位
+-----+	+-----+
0000010	待定
+-----+	+-----+

本草案的定义、动机和使用正在研究中。

2.5.7 可集聚全球单播地址

全部可集聚全球单播地址定义在[AGGR]中。设计这样的地址格式为了既支持基于当前供应商的集聚，又支持被称为交换局的新的集聚类型。其组合使高效的选路集聚可用于直接连接到供应商和连接到交换局两者的站点上。站点可以选择连接到两种类型中的任何一种集聚点。

IPv6可集聚全球单播地址格式如下所示：

3	13	8	24	16	64位
+-----+	+-----+	+-----+	+-----+	+-----+	+-----+
FP	TLA	RES	NLA	SLA	接口ID
	ID		ID	ID	
+-----+	+-----+	+-----+	+-----+	+-----+	+-----+

其中，001(FP)用于可集聚全球单播地址的格式前缀(3位)；TLA ID为顶级集聚标识符；RES保留将来用；NLA ID为下一级集聚标识符；SLA ID为站点级集聚标识符；INTERFACE ID为接口标识符。

在[AGGR]中，还规定了内容、字段长度和分配规则。

2.5.8 本地用IPv6单播地址

规定了链路本地和站点本地两种类型的本地使用单播地址。链路本地地址用在单链路上，而站点本地地址用在单站点上。链路本地地址格式表示如下：

10位	54位	64位
+-----+	+-----+	+-----+
1111111010	0	接口ID
+-----+	+-----+	+-----+

设计链路本地地址的目的是为了用于诸如自动地址配置、邻居发现或无路由器存在的单链路的寻址。路由器不能将带有链路本地源地址或目的地址的任何包转发到其他链路上去。

站点本地地址具有下面的地址格式：

10位	38位	16位	64位
11111111011	0	子网ID	接口ID

站点本地地址的设计目的是为了用于无需全球前缀的站点内部寻址。

路由器不应转发站点外具有站点本地源或目的地址的任何包。

2.6 任意点播地址

IPv6任意点播地址是分配给一般属于不同节点的多个接口。根据这个特性，发送给任意点播地址的包，总是发送到具有该地址并按照选路协议测得距离为最近的接口。

任意点播地址从单播地址空间分配而来，可用任何一种规定的单播地址格式。这样，任意点播地址和单播地址在语法上是无法区别的。当一个单播地址分配给多个接口时，如果把它转为任意点播地址，那么被分配该地址的节点，必须显式地配置，以便知道这是一个任意点播地址。

对于任何已分配的单播地址，有一个最长的地址前缀 P 用于标识拓扑地区。在该地区中，所有接口均属于该任意点播地址。在由 P 标识的区域内，任意点播组的每个成员，被告知在选路系统中作为一个独立实体（通常称之为“主机路由”）。在 P 标识的区域以外，任意点播地址可以集合在前缀 P 的选路通告中。

在最坏情况下，一个任意点播组的前缀 P 可以是 0 前缀，那组成员可能没有拓扑位置。在这种情况下，任意点播地址在整个 Internet 中，必须被告知作为一个分离的选路实体，这就为可以支持多少这样的全球任意点播组，带来严格的规模限制。因此，期望支持全球任意点播组似乎是不可能的或者说是非常受限制的。

任意点播地址的用途之一是标识一组路由器，该组路由器是属于提供 Internet 服务的一个组织的。这样的地址在 IPv6 选路头中可用作直接地址，造成包的传递通过一个特定的集聚或集聚系列。其他可能的用途是标识连到一个特定子网的一组路由器，或者标识提供入口到一个特定选路域的一组路由器。

Internet 任意点播地址在广泛传播及随意使用方面经验不多，然而已知使用它们所带来的复杂性和麻烦却很普遍 [ANYSCT]。在获得更多的经验，并对一些问题有一致的解决方案之前，IPv6 任意点播地址的下列限制始终存在。

- 任意点播地址不能用作 IPv6 包的源地址。
- 任意点播地址不能指定给 IPv6 主机，只能指定给 IPv6 路由器。

要求的任意点播地址

预定的子网路由器任意点播地址，其格式如下：

n位	128-n位
子网前缀	0000000000000000

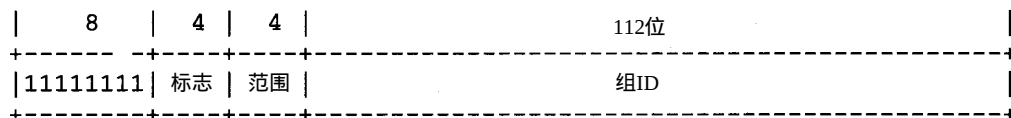
在任意点播地址中，子网前缀用来标识一条特定链路。对于接口标识符置 0 的链路上的一个接口，其任意点播地址和单播地址语法上是相同的。

发送给子网路由器任意点播地址的包会传递到子网上的一个路由器。与子网有接口的所有路由器需要支持子网路由器任意点播地址。

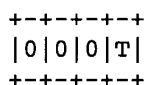
子网路由器任意点播地址企图用在某些应用场合，即一个节点需要和远程子网上一组路由器中的一个进行通信的场合。例如当移动主机要和一个位于本子网的移动代理通信的场合。

2.7 组播地址

IPv6组播地址是一组节点的标识符。一个节点可以归属于任意数量的组播组。组播地址具有下面的格式：



地址开始的 1111 1111 标识该地址为组播地址。标志由 4 位组成：



前面3位为保留位，初始设置为0。

T=0指示一个永久分配的(熟知的)组播地址，由全球 Internet 编号机构进行分配。

T=1指示一个非永久分配(临时)的组播地址。

4位的组播范围值用来限制组播组的范围。该字段的可能值如下表。

值	描 述	值	描 述
0	保留	8	组织本地范围
1	节点本地范围	9	(未分配)
2	链路本地范围	A	(未分配)
3	(未分配)	B	(未分配)
4	(未分配)	C	(未分配)
5	站点本地范围	D	(未分配)
6	(未分配)	E	全球范围
7	(未分配)	F	保留

组标识符字段标识给定范围内的组播组，可以是永久的，也可以是临时的。

永久分配的组播地址，意思是独立于范围值。例如，如果为 NTP 服务器组指定一个组标识符为 101(十六进制)的永久组播地址，于是：

FF01:0:0:0:0:0:0:101 意指在同一节点上的所有 NTP 服务器。

FF02:0:0:0:0:0:0:101 意指在同一链路上的所有 NTP 服务器。

FF05:0:0:0:0:0:0:101 意指在同一站点上的所有 NTP 服务器。

FF0E:0:0:0:0:0:0:101 意指 Internet 上的所有 NTP 服务器。

非永久分配的组播地址仅在给定范围内才有意义。例如，在某个站点由非永久的站点本地组播地址 FF15:0:0:0:0:0:0:101 标识的组与一个不同站点中使用同一个组标识符的组没有关系，与不同范围内使用同一个组标识符分配非永久地址的组也没有关系，与具有同一个组标识符的永久组也没有关系。

组播地址在 IPv6 包中不能用作源地址或出现在任何选路头中。

2.7.1 预定义的组播地址

下面为熟知的预定义的组播地址：

保留的组播地址：

```
FF00:0:0:0:0:0:0:0  
FF01:0:0:0:0:0:0:0  
FF02:0:0:0:0:0:0:0  
FF03:0:0:0:0:0:0:0  
FF04:0:0:0:0:0:0:0  
FF05:0:0:0:0:0:0:0  
FF06:0:0:0:0:0:0:0  
FF07:0:0:0:0:0:0:0  
FF08:0:0:0:0:0:0:0  
FF09:0:0:0:0:0:0:0
```

上面列出的是保留的组播地址，且永远不能分配给任何组播组。

所有节点地址：

```
FF0A:0:0:0:0:0:0:0  
FF0B:0:0:0:0:0:0:0  
FF0C:0:0:0:0:0:0:0  
FF0D:0:0:0:0:0:0:0  
FF0E:0:0:0:0:0:0:0  
FF0F:0:0:0:0:0:0:0
```

上面列出的组播地址标识了范围1(节点本地)或范围2(链路本地)内的所有IPv6节点的组。

所有路由器地址：

```
FF01:0:0:0:0:0:0:1  
FF02:0:0:0:0:0:0:1
```

以上的组播地址标识了范围1(节点本地)、范围2(链路本地)或范围5(站点本地)内的所有IPv6路由器的组。

```
FF01:0:0:0:0:0:0:2  
FF02:0:0:0:0:0:0:2  
FF05:0:0:0:0:0:0:2
```

请求节点地址：FF02:0:0:0:0:1:FFXX:XXXX

上面的组播地址是从节点的单播和任意点播地址计算而得的。取单播或任意点播地址的低24位，并将其附加到前缀FF02:0:0:0:0:1:FF00::/104上形成一个请求节点组播地址，其范围在FF02:0:0:0:0:1:FF00:0000至FF02:0:0:0:0:1:FFFF:FFFF之间。

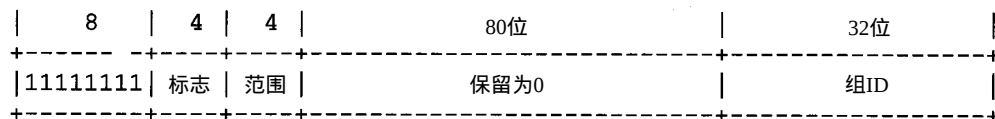
例如，对应IPv6地址4037::01:800:200E:8C6C的请求节点组播地址是FF02::1:FF0E:8C6C。IPv6地址差别仅在高位，譬如，由于与不同的集聚相关联的多个高位前缀，将映射到同一个请求节点地址，因此减少了一个节点必须加入的组播地址数。

对每个指定的单播和任意点播地址，一个节点需要计算并加入相关的请求节点组播地址。

2.7.2 新IPv6组播地址的分配

目前将IPv6组播地址映射到IEEE 802 MAC地址的方法是用IPv6组播地址的低阶32位来创建

MAC地址。值得提出的是令牌网有不同的处理方法，定义见[TOKEN]。32位组标识符将生成唯一的MAC地址。由于新IPv6组播地址应当分配，所以组标识符总是在低阶32位上，如下图所示：



尽管将永久IPv6组播组数限制在 2^{32} ，但在将来不可能成为极限。如果将来必须要超过这个限度，组播仍然能工作，只是处理稍慢而已。

其他IPv6组播地址的定义和注册由IANA^[MASGN]完成。

2.8 节点要求的地址

主机需要识别下面的地址以辨识它自身：

- 它的每个接口的链路本地地址。
- 分配的单播地址。
- 回返地址。
- 所有节点的组播地址。
- 每一个分配的单播和任意点播地址的请求节点组播地址。
- 主机所属的所有其他组的组播地址。

主机需要识别的所有地址，要求路由器都能识别，路由器还要能识别用来识别其本身的下列地址：

- 配置路由器工作的接口所用的子网路由器任意点播地址。
- 完成路由器配置要用的所有其他任意点播地址。
- 所有路由器组播地址。
- 路由器归属于所有其他组的组播地址。

在实现中应该预定义的地址前缀包括：

- 未指定地址。
- 回返地址。
- 组播前缀(FF)。
- 本地用前缀(链路本地和站点本地)。
- 预定义的组播地址。
- IPv4兼容的前缀。

实现时除非专门配置(如任意点播地址)，应假设所有其他地址均为单播地址。

3. 安全性考虑

IPv6寻址文件对Internet基础设施的安全性没有任何直接影响。IPv6包身份验证的定义见[AUTH]。

附录A 创建EUI-64接口标识符

根据特定链路或节点的特性，有不少方法可以创建EUI-64接口标识符。本附录介绍了其中的某些方法。

A.1具有EUI-64标识符的链路或节点

将一个EUI-64标识符转换成一个接口标识符，只需改变 u 位的值。例如，一个全球唯一的EUI-64标识符具有下面的形式。

0	1 1	3 3	4 4	6
0	5 6	1 2	7 8	3
+-----+-----+-----+-----+-----+				
ccccc0gccccccc cccccccmmmmmmmm mmmmmmmmmmmmmmmmm mmmmmmmmmmmmmmmmm				
+-----+-----+-----+-----+-----+				

其中，c位是分配给公司的标识符；0是全球/本地位的值，此处指本地范围；m是生产商选择的扩展标识符。IPv6接口标识符的形式如下：

0	1 1	3 3	4 4	6
0	5 6	1 2	7 8	3
+-----+-----+-----+-----+-----+				
ccccc1gccccccc cccccccmmmmmmmm mmmmmmmmmmmmmmmmm mmmmmmmmmmmmmmmmm				
+-----+-----+-----+-----+-----+				

唯一改变的是转变全球/本地位的值。

A.2具有IEEE 802 48位MAC地址的链路或节点

[EUI64]规定了从一个IEEE 48位MAC标识符创建一个EUI-64标识符的方法。就是将十六进制表示的两个字节 OxFF和OxFE插入到48位MAC地址中间(公司标识符与厂商配给的标识符之间)。下面的例子是一个具有全球范围的48位MAC地址。

0	1 1	3 3	4
0	5 6	1 2	7
+-----+-----+-----+			
ccccc0gccccccc cccccccmmmmmmmm mmmmmmmmmmmmmmmmm			
+-----+-----+-----+			

其中，c位是分配给公司的标识符；0是指示全球范围的全球/本地位值；g是个体/团体位；m是生产厂选择的扩展标识符。这样，接口标识符便具有下面的形式。

0	1 1	3 3	4 4	6
0	5 6	1 2	7 8	3
+-----+-----+-----+-----+-----+				
ccccc1gccccccc ccccccc11111111 11111110mmmmmmmmmm mmmmmmmmmmmmmmmmm				
+-----+-----+-----+-----+-----+				

当接口或节点上IEEE 802 48位MAC地址可用时，由于它们具备的可用性和唯一性特性，就可以用它来实现创建接口标识符。

A.3具有非全球标识符的链路

有许多链路类型，当多个接入时，例如包括 LocalTalk和Arcnet,就无全球唯一的链路标识符。创建EUI-64格式化标识符的方法是取链路标识符(如LocalTalk 8位节点标识符)，并在其前面填充0。下面就是一个具有十六进制值 Ox4F的LocalTalk 8位节点标识符生成的接口标识符的例子。

0	1	3	4	6
0	5	1	7	3
0000000000000000	0000000000000000	0000000000000000	000000001001111	

注意其中的全球/本位置为0，以指示本地范围。

A.4 无标识符的链路

有一些链路无任何类型内置标识符。最普遍的就是一些串行链路和配置的隧道。为链路选择的接口标识符必须是唯一的。

当一条链路上无内置标识符可用时，最好是用从另一个接口的，或分配给节点本身的，一个全球接口标识符。使用这种方法就不会有连接同一链路的同一节点的其他接口会用同样的标识符。

如果在链路上无全球接口标识符可使用时，就需要创建一个本地范围接口标识符。唯一的要求就是在该链路上是唯一的。有许多可能的方法用来选择一条链路唯一的接口标识符，包括如下方法：

- 人工配置。
- 生成随机数。
- 节点串行号(或其他节点特殊令牌)。

链路唯一接口标识符的生成方法应该使一个节点启动后或者接口从节点中删除或加入时都不应该有变化。

合适算法的选择，取决于链路和实现。形成接口标识符的细节规定在相应的 IPv6 over <link>技术规范中。强烈建议在任何自动算法中要实现冲突检测算法。

附录B 文本表示的ABNF描述

本附录定义了ABNF^[ABNF]中的IPv6地址及前缀的文本表示，仅供参考用。

```
IPv6address = hexpart [ ":" IPv4address ]
IPv4address = 1*3DIGIT "." 1*3DIGIT "." 1*3DIGIT "." 1*3DIGIT
IPv6prefix = hexpart "/" 1*2DIGIT

hexpart = hexseq | hexseq "::" [ hexseq ] | "::" [ hexseq ]
hexseq = hex4 * ( ":" hex4 )
hex4 = 1*4HEXDIG
```

附录C 对RFC 1884的修改

对RFC 1884(IPv6寻址体系结构)作了如下的修改：

- 增加了一个描述文本表示的ABNF的附录。
- 澄清了链路唯一标识符在自举或其他接口重新配置后不会改变。
- 阐述了评议后的地址模型。
- 改变了集聚格式术语，以便和集聚草案一致。
- 增加了在同一节点上，接口标识符可用于多个接口的文字说明。
- 增加了定义新组播地址的规则。

- 增加了创建基于EUI-64接口标识符的描述过程。
- 增加了定义IPv5前缀的标记方法。
- 用一个长的前缀改变请求节点组播的定义。
- 增加了站点范围所有路由器组播地址。
- 规定可集聚全球单播地址用001格式前缀。
- 将010(基于供应商的单播)和100(保留为地理上的)格式前缀改成未指定的格式前缀。
- 增加了对单播地址的接口标识符的定义部分；对单播地址增加了接口标识符定义的选择。要求在格式前缀范围内使用EUI-64以及在EUI-64中置全球/本地范围位的规则。
- 更新了NSAP文本部分以反映RFC 1888的工作。
- 删去协议特定的IPv6组播地址(如DHCP)，并参考了IANA中的定义。
- 删去了“单播地址例子”部分，变成OBE。
- 增加了新参考文献，并更新了参考文献。
- 对少量文字说明进行了澄清和改进。

参考文献

- [ABNF] Crocker, D., and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, November 1997.
- [AGGR] Hinden, R., O'Dell, M., and S. Deering, "An Aggregatable Global Unicast Address Format", RFC 2374, July 1998.
- [AUTH] Atkinson, R., "IP Authentication Header", RFC 1826, August 1995.
- [ANYCST] Partridge, C., Mendez, T., and W. Milliken, "Host Anycasting Service", RFC 1546, November 1993.
- [CIDR] Fuller, V., Li, T., Yu, J., and K. Varadhan, "Classless Inter-Domain Routing (CIDR): An Address Assignment and Aggregation Strategy", RFC 1519, September 1993.
- [ETHER] Crawford, M., "Transmission of IPv6 Packets over Ethernet Networks", Work in Progress.
- [EUI64] IEEE, "Guidelines for 64-bit Global Identifier (EUI-64) Registration Authority", <http://standards.ieee.org/db/oui/tutorials/EUI64.html>, March 1997.
- [FDDI] Crawford, M., "Transmission of IPv6 Packets over FDDI Networks", Work in Progress.
- [IPv6] Deering, S., and R. Hinden, Editors, "Internet Protocol, Version 6 (IPv6) Specification", RFC 1883, December 1995.
- [MASGN] Hinden, R., and S. Deering, "IPv6 Multicast Address Assignments", RFC 2375, July 1998.
- [NSAP] Bound, J., Carpenter, B., Harrington, D., Houldsworth, J., and A. Lloyd, "OSI NSAPs and IPv6", RFC 1888, August 1996.

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [TOKEN] Thomas, S., "Transmission of IPv6 Packets over Token Ring Networks", Work in Progress.
- [TRAN] Gilligan, R., and E. Nordmark, "Transition Mechanisms for IPv6 Hosts and Routers", RFC 1993, April 1996.

作者地址

Robert M. Hinden

Nokia

232 Java Drive

Sunnyvale, CA 94089

USA

Phone: +1 408 990-2004

Fax: +1 408 743-5677

EMail: hinden@iprg.nokia.com

Stephen E. Deering

Cisco Systems, Inc.

170 West Tasman Drive

San Jose, CA 95134-1706

USA

Phone: +1 408 527-8213

Fax: +1 408 527-8254

EMail: deering@cisco.com

版权声明

本文件全部版权属于 The Internet Society(1998)。

本文件及其译文可以复制并对外提供，可以部分或全部编著、复制、出版、分发与其有关的评议、解释和有助于实施的派生著作，没有任何限制，要求在复制文件和派生著作中包括上述版权警告及本节版权声明内容。但是，本文件的内容不允许做任何形式的修改，诸如删除版权警告或者关于 Internet Society 或其他 Internet 组织的介绍，除非为了开发 Internet 标准或者翻译成英语以外的其他语言的需要，即使在这种情况下，也仍然必须遵循 Internet 标准过程中确定的版权程序。

上述许可是永久性的，不会由 The Internet Society、他的继任者或转让者予以废除。

本文件及其提供的信息以“现状”为基础，The Internet Society 与 IETF 否认所有的保证、明示或暗示、包含但并不限于任何保证所含信息的使用，将不会侵犯具有特殊目的的商用性或适用性的任何权利或隐含的保证。

RFC 2374 IPv6可集聚全球单播地址格式

网络工作组

RFC 2374

撤销：2073

类别：标准跟踪

R. Hinden

Nokia

M. O'Dell

UUNET

S. Deering

Cisco

1998年7月

提示

本文为Internet社区指定一个Internet标准跟踪协议，并请求为改进进行讨论和提出建议。对标准化状态和本协议的状况请参考“Internet正式协议标准”(STD-1)最新版本。本文的分发不受限制。

版权声明

本文件全部版权属于The Internet Society (1998)。

1. 引论

本文定义了可用于Internet上的IPv6可集聚全球单播地址格式。本文定义的地址格式与IPv6协议^[IPv6]以及“IPv6寻址体系结构”^[ARCH]是一致的。它的设计是为了推进规模可伸缩的Internet选路。

本文件取代了RFC 2073(基于供应商的IPv6单播地址格式)。RFC 2073成为了历史文件。可集聚全球单播地址格式是对RFC 2073某些方面的改进。主要的改变包括删去了对路由集聚、EUI-64接口标识符的支持，对供应商和交换局集聚的支持，公共和站点拓扑的分割以及新集聚术语等所不需要的注册位。

2. IPv6地址概述

IPv6地址是为接口和接口组指定的128位标识符。有三类地址：单播、任意点播和组播。本文专门定义单播地址类。

在本文中，地址内的字段，赋予如“子网”这样的专门名字。当名字与其后的名词“标识符”一起使用时(如“子网标识符”)，被称为名字字段的内容。当名字与名词“前缀”一起使用时(如“子网前缀”)，则表示包括本字段在内的所有左边的寻址位。

IPv6单播地址的设计使Internet选路系统在不需要了解IPv6地址内部结构的情况下，在任意位边界上，使用一个最长的前缀匹配“算法”，就可作出包的转发决定。IPv6地址的结构是指派和分配用的。唯一的例外就是要在单播和组播地址之间加以区别。

IPv6地址的特定类型由地址的前几位指出。包含这前几位的可变长度字段叫做格式前缀(FP)。

本文为可集聚全球单播地址定义地址格式，其格式前缀为001(二进制)。其他格式前缀也

可以采用同样的地址格式，只要这些格式前缀是标识 IPv6 单播地址的。只是本文只定义了这一种格式前缀而已。

3. IPv6 可集聚全球单播地址格式

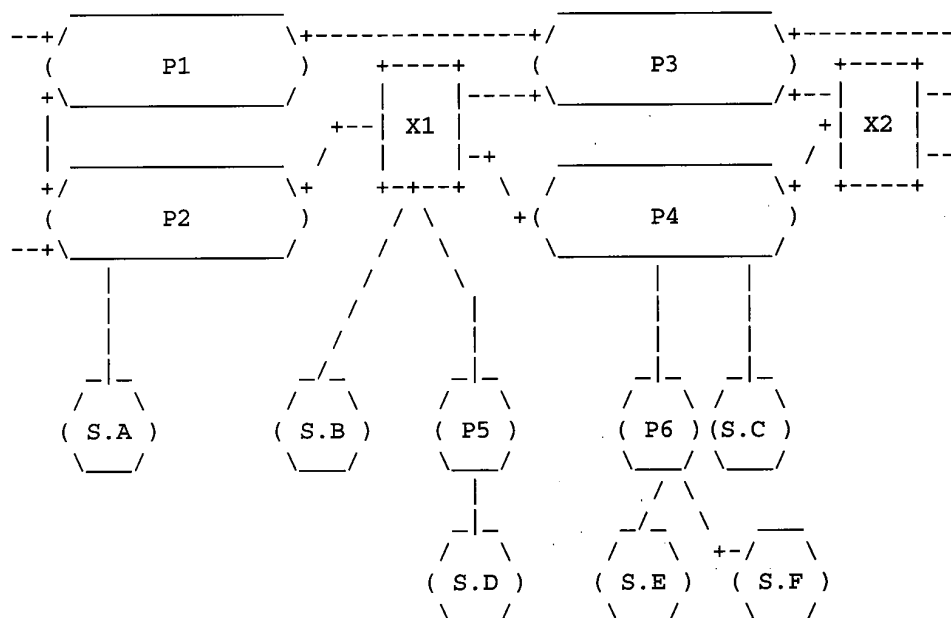
本文为 IPv6 可集聚全球地址格式的分配定义一种地址格式。作者相信这地址格式会广泛用于连到 Internet 的 IPv6 节点。设计该地址格式时，考虑到既支持当前基于供应商的集聚，也支持新的基于交换局的集聚。其组合既允许直接连接到供应商的站点能高效率地选路集聚，也允许连接到交换局的站点能高效率地选路集聚。站点可以选择连接到两者中的任一个集聚实体。

当该地址格式的目的是支持基于交换局的集聚（除了当前基于提供商的集聚外）时，它的总路由集聚特性与交换局无关。只有用基于供应商的集聚，才能提供效率高的路由集聚。

可集聚地址安排成一个三层次的分级结构：

- 公用拓扑。
- 站点拓扑。
- 接口标识符。

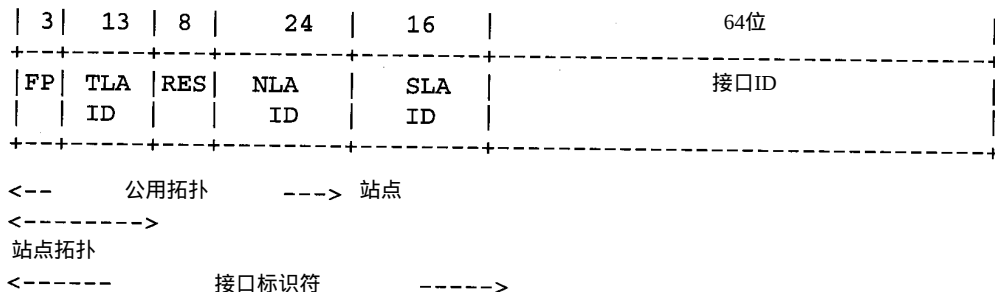
公用拓扑是提供公用 Internet 传送服务的供应商和交换局群体。站点拓扑是本地的特定站点或组织，它不提供到本站点以外节点的公用传送服务。接口标识符是标识链路上的接口。



正如上面图所表示的可集聚地址格式，其目的是支持长途供应商（如图中 P1、P2、P3、P4），交换局（如图中 X1 和 X2），多级供应商（如图中 P5 和 P6）和用户（如图中 S.x）。交换局（不像目前的 NAPs、FIXes 等）将分配 IPv6 地址。连接到这些交换局的组织，也要从一个或多个长途供应商那里预订（直接、间接地通过交换局等）长途服务。这样做可使寻址与长途转运供应商无关。这使得在改换长途供应商时，无需给它们的组织重新编号。组织也能成为多家的，也就是通过交换局连到一个以上的长途供应商，而不需要从每个长途供应商处获得地址前缀。用于此类供应商的选择及移植性的机制不在本文中讨论。

3.1 可集聚全球单播地址结构

可集聚全球单播地址格式表示如下：



其中，FP为格式前缀(001)；TLA ID为顶级集聚标识符；RES保留为将来用；NLA ID为下一级集聚标识符；SLA ID为站点级集聚标识符；INTERFACE ID为接口标识符；

下面分别给出IPv6可集聚全球单播地址格式的每一部分的说明。

3.2 顶级集聚标识符

顶级集聚标识符(TLA ID)是选路分级结构中的最高级。非默认路由器必须为每个激活的TLA ID保留一个路由表项，同时也许还有为TLA ID提供选路信息的附加项。附加项的目的是为它们的特定拓扑优先选路，但是所有级的选路拓扑，必须使提供给非默认路由表的附加项数量最少。

这样的寻址格式支持8192(2^{13})个TLA ID。要增加TLA ID的数量可以向右扩展TLA字段到保留字段，或者在另外的格式前缀上使用此格式。

关系分配TLA ID的议题，超出了本文范围，将在正在进行准备的文件中说明。

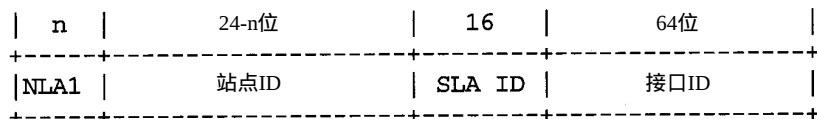
3.3 保留字段

保留字段留作将来用，当前必须置成0。

保留字段可留作TLA和NLA字段扩展时用。见第4节的讨论。

3.4 下一级集聚标识符

下一级集聚标识符被得到一个TLA ID的机构用来创建寻址分级结构和标识站点。该机构可以指定NLA ID字段的前n位，用来创建适合于它的网络的寻址分级结构。该字段的其余部分用来标识它愿为之服务的站点。表示如下：



每个得到一个TLA ID的机构可以有24位NLA ID空间。NLA ID空间使每个机构能够为相当于目前IPv4 Internet能够支持的总网络数几乎一样多的组织提供服务。

得到TLA ID的机构，也支持他们自己站点ID空间中的NLA ID。这就允许得到TLA ID的机构，能给提供公用传送服务的机构提供服务，也能给不提供公用传送服务的机构提供服务。

得到 NLA ID 的机构，也可以选择用他们的站点 ID 空间去支持其他的 NLA ID。这种情况表示如下：

n	24-n位	16	64位
-----	-----	-----	-----
NLA1	站点ID	SLA ID	接口ID
-----	-----	-----	-----
m	24-n-m	16	64位
-----	-----	-----	-----
NLA2	站点ID	SLA ID	接口ID
-----	-----	-----	-----
o	24-n-m-o	16	64位
-----	-----	-----	-----
NLA3	站点ID	SLA ID	接口ID
-----	-----	-----	-----

对一个特定的 TLA ID，设计 NLA ID 位的安排，留给负责该 TLA ID 的机构去做。同样，设计下一级 NLA ID 位的安排，由前面一级 NLA ID 负责。在此建议分配 NLA 地址空间的机构用类似于[RFC2050]中的“慢启动”分配过程。

设计 NLA ID 分配计划，要在选路集聚效率和灵活性之间进行权衡。创建分级结构允许较大集聚数，从而使得路由表较小。平面 NLA ID 的分配能使分配容易和连接灵活，但使得路由表较大。

3.5 站点级集聚标识符

SLA ID 字段被单个机构用来创建他自己的本地寻址分级结构与标识子网。除了每个机构有一个数量很大的子网以外，类似于 IPv4 中的子网。16 位的 SLA ID 字段支持 65 535 个单个子网。

机构可以选择他们的 SLA ID 为平面路由（如在 SLA 标识符之间不创建任何逻辑关系，这会使得路由表较大），或者在 SLA ID 字段中，创建一个两级或多级分级结构（使路由表较小）。后一种情况表示如下：

n	16-n	64位
-----	-----	-----
SLA1	子网	接口ID
-----	-----	-----
m	16-n-m	64位
-----	-----	-----
SLA2	子网	接口ID
-----	-----	-----

构成 SLA ID 字段所选择的方法，由个别机构负责。

在这种地址格式下支持的子网数，除了最大的机构之外，对其他所有机构应该是足够的。对于需要更多子网的组织，可以和它获得 Internet 服务的机构商量，以获得附加的站点标识符，从而用来创建更多的子网。

3.6 接口ID

接口标识符用来标识一条链路上的接口。对链路来说，应该是唯一的。也可以在一个更

宽的范围内是唯一的。许多情况下,一个接口标识符与接口的链路层地址相同,或者根据接口的链路层地址而得的。用在可集聚全球单播地址格式中的接口标识符要求 64位长,并能构成IEEE EUI-64格式^[EUI-64]。这些标识符,当全球令牌(如IEEE 48位MAC)可用时,具有全球范围意义;当全球令牌不可用时(如串行链路、隧道终点等),则只具有本地范围意义。u位(在IEEE EUI-64术语中称为全球/本地位)在EUI-64标识符中必须根据[ARCH]的规定,正确地置位以指示是全球还是本地范围。

创建基于EUI-64接口标识符的过程定义见[ARCH]。形成接口标识符的细节,规定在相应的IPv6 over<link>技术规范中,诸如IPv6 over Ethernet^[ETHER], IPv6 over FDDI^[FDDI]等。

4. 技术动机

在可集聚的地址格式中,字段长度的设计选择需要满足许多技术需求。这些将在下面段落中介绍。

顶级集聚标识符的长度是13位。可有8192个TLA ID。选择这样的长度,可使Internet上顶级路由器的非默认路由表,能在当前的选路技术且合理地留有余量的情况下,保持有限的范围。

因为非默认路由器为优化 TLA内部路径和 TLA之间的路径,还要含有大量的长的前缀,所以保留余量是重要的。

重要的议题不仅是非默认选路表的长度问题,还有拓扑的复杂性决定了当计算一个转发表时,路由器必须考察非默认路由的拷贝数。当前 IPv4的实践是通常一个前缀要通过不同的路径通告15次。

Internet拓扑的复杂性将来还可能增加。重要的是 IPv6非默认选路应支持更大的复杂性以及巨大的Internet。

应该提出的是,在写本文时(1998年春),作者作了一个比较,IPv4非默认路由表包含大约50 000个前缀,表示可能支持大于8192个的路由。现在争论的问题是在当前的选路技术下,是否IPv4目前支持的前缀数已经足够多了。一些需要认真考虑的议题是路由稳定性以及供应商不支持所有顶级前缀的情况。技术上要求挑选 TLA ID的长度,在考虑合理余量的情况下,低于IPv4所具有的。

选择TLA ID字段为13位是出于工程的综合考虑。位数太少将不足以支持足够的顶级组织,位数太多将会超过合理协调的能力。为了处理前面所提到的议题,用当前的选路技术考虑一个合理的余量是合适的。

如果将来选路技术改进到在非默认路由表中能支持大量的顶级路由,那么如何加大 TLA标识符,就有两种选择:第一种是扩大 TLA ID字段占用保留字数,这将使TLA ID数大约增加二百万个;第二种途径是为这样的地址格式分配另一个格式前缀(FP)。或者将这两种途径组合,使TLA ID数量大大地增加。

保留字段的长度为8位,是为了使TLA ID字段和NLA ID字段有大的增长余地。

下一级集聚标识符的长度为24位。如果用平面结构的话,可容纳大约1600万个NLA ID。如果分级使用的话,合成起来大致等效于IPv4的地址空间(假定平均网络规模为254个接口)。如果NLA ID将来需要更多的空间,那么可以将NLA ID扩展到保留字段来协调。

站点级集聚标识符字段的长度是16位。每个站点可支持65 535个子网。本字段长度的设计目标,对除了最大组织以外的所有组织是足够的。对于需要更多子网的组织,可以和它获

得Internet服务的机构商量,以得到附加的站点标识符,从而用来创建更多的子网。

站点集聚标识符字段是固定长度,这是为了强制标识特定站点的所有前缀,具有同样的长度(即48位)。这样会方便站点在拓扑中的移动(如变更ISP以及接到多个ISP的多家站点)。

接口标识符字段为64位。选择这个长度是为了满足[ARCH]中指定的需求,以支持基于EUI-64接口标识符。

致谢

作者对Thomas Narten, Bob Fink, Matt Crawford, Allison Mankin, Jim Bound, Christian Huitema, Scott Bradner, Brian Carpenter, John Stewart和Daniel Karrenberg的评论和建设性意见表示衷心的感谢。

参考文献

- [ALLOC] IAB and IESG, "IPv6 Address Allocation Management", RFC 1881, December 1995.
- [ARCH] Hinden, R., "IP Version 6 Addressing Architecture", RFC 2373, July 1998.
- [AUTH] Atkinson, R., "IP Authentication Header", RFC 1826, August 1995.
- [AUTO] Thompson, S., and T. Narten., "IPv6 Stateless Address Autoconfiguration", RFC 1971, August 1996.
- [ETHER] Crawford, M., "Transmission of IPv6 Packets over Ethernet Networks", Work in Progress.
- [EUI64] IEEE, "Guidelines for 64-bit Global Identifier (EUI-64) Registration Authority", <http://standards.ieee.org/db/oui/tutorials/EUI64.html>, March 1997.
- [FDDI] Crawford, M., "Transmission of IPv6 Packets over FDDI Networks", Work in Progress.
- [IPv6] Deering, S., and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 1883, December 1995.
- [RFC2050] Hubbard, K., Kouters, M., Conrad, D., Karrenberg, D., and J. Postel, "Internet Registry IP Allocation Guidelines", BCP 12, RFC 1466, November 1996.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

安全性考虑

IPv6寻址文件对Internet基础设施安全性无任何直接影响。IPv6包的身份验证定义见[AUTH]。

作者地址

Robert M. Hinden
Nokia

232 Java Drive
Sunnyvale, CA 94089
USA

Phone: 1 408 990-2004
EMail: hinden@iprg.nokia.com

Mike O'Dell
UUNET Technologies, Inc.
3060 Williams Drive
Fairfax, VA 22030
USA

Phone: 1 703 206-5890
EMail: mo@uunet.uu.net

Stephen E. Deering
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA

Phone: 1 408 527-8213
EMail: deering@cisco.com

版权声明

本文件全部版权属于 The Internet Society(1998)。

本文件及其译文可以复制并对外提供。可以部分或全部编著、复制、出版、分发与其有关的评议、解释和有助于实施的派生著作，没有任何限制，但要求在复制文件和派生著作中包括上述版权警告及本节版权声明内容。但是，本文件的内容不允许做任何形式的修改，诸如删除版权警告或者关于 Internet Society 或其他 Internet 组织的介绍，除非为了开发 Internet 标准或者翻译成英语以外的其他语言的需要，即使在这种情况下，也仍然必须遵循 Internet 标准过程中确定的版权程序。

上述许可是永久性的，不会由 The Internet Society、他的继任者或转让者予以废除。

本文件及其提供的信息以“现状”为基础，The Internet Society 与 IETF 否认所有的保证、明示或暗示、包含但并不限于任何保证所含信息的使用，将不会侵犯具有特殊目的的商用性或适用性的任何权利或隐含的保证。