路由器新技术丛书

路由器原理与设计

白建军　卢泽新　编著

人 民 邮 电 出 版 社

内 容 提 要

本书从路由器的基本原理出发，结合工程实践和国际上路由器技术的发展现状与趋势，深入浅出地介绍了高性能路由器设计的一系列关键技术和实现难点。主要内容包括：高性能路由器的体系结构、高速接口、系统软件、路由体系及路由协议栈等方面的设计与实现技术，同时对路由表问题、高级交换技术以及路由器的安全体系和 QoS 作了深入介绍。最后还给出了当前国际上一些主流高端路由器产品的主要技术指标。

本书内容详尽，是一本当前少有的详细介绍路由器设计技术的书籍，既适合于具备一定网络基础知识的网络工程师、高等院校网络专业师生阅读，也可供网络数据产品研发及相关人员参考。

路由器新技术丛书

路由器原理与设计

◆ 编　　著　白建军　卢泽新

责任编辑　梁　凝

◆ 人民邮电出版社出版发行　　北京市崇文区夕照寺街 14 号

邮编　100061　　电子函件　315@ptpress.com.cn

网址　http://www.ptpress.com.cn

读者热线　010-67180876

北京汉魂图文设计有限公司制作

印刷厂印刷

新华书店总店北京发行所经销

◆ 开本：787×1092　1/16

印张：

字数：　千字　　2002 年 6 月第 1 版

印数：1 –　000 册　　2002 年 6 月北京第 1 次印刷

ISBN 7-115-10208-2/TN · 1857

定价：　元

本书如有印装质量问题，请与本社联系　电话：（010）67129223

前　言

随随着 Internet 的迅猛发展，一个专门为解决 Internet 骨干网运营商所面临的特殊问题的网络设备——路由器随之出现。

传统的 IP 网络在传输实时业务时的主要瓶颈在于路由器速度太慢，时延和时延抖动太大，不能满足 QoS 要求，因此研制核心路由器的任务显得非常重要。传统路由器采用软件实现路由识别、计算和包转发，但即使是昂贵的高档路由器的包转发速度也远远低于网络交换机的速度。自 1997 年下半年以来，一些公司开始陆续推出采用硬件专用电路(ASIC)进行路由识别、计算和转发的新型 Internet 骨干路由器。

骨干路由器不仅能够集合带宽及业务的吞吐量，而且还要具备丰富的软件功能及控制特性。为能同时实现增加带宽及丰富的软件功能，新型路由系统的设计者必须在设计中采用以前只有在交换机中才具有的转发功能。但是，相对于为固定长度数据包提供线速性能的直接交换而言，实现应用最长匹配查询的不定长度数据包线速路由与转发要复杂得多。特别是，这种数据包的处理已不能通过基于微处理器或微处理器辅助的方式来实现，而必须使用一种基于随 Internet 环境变化而改变其路由策略的专用集成电路芯片（ASIC）的方式完成。这种新型的 Internet 骨干路由器所面临的系统上的挑战也是非常复杂的。它们不仅要能适应现有的 OC-12 及 OC-48 速率的骨干网和相关的 Intra-POP（Intra Point of Presence，内部层现网络）结构，而且还要能够支持具有 OC-48 速率和 10Gbit/s 以太网 Intra-POP 结构的 OC-192 速率的骨干网络。另外，新的路由系统必须要加速 Internet 从“尽力服务”模式向最基础的可靠性服务的方向转变。Internet 用户希望能够像使用公共电话网络那样，在任何需要通信的时候，听到“拨号音”，便可得到高质量的服务。

本书以高端路由器技术为主，介绍路由器的原理与设计，全书共分 10 章。第 1 章从路由器技术基础开始谈起，介绍了路由器的基本组成、工作机制、路由器的分类以及高端路由器的关键组成。在第 2 章中，介绍了路由器技术的发展与当前主要的路由器实现标准。第 3 章概括阐述了路由器实现过程中的一些关键技术。从第 4 章开始具体介绍实现细节，包括报文交换体系结构、路由器软件系统、高速硬件接口设计技术以及路由协议的设计与实现技术。作为一个独立部分，第 8 章着重阐述了 Internet 骨干网的路由表问题，这不仅仅是因为随着 Internet 的急速增长，路由表容量对路由器路由转发的性能影响越来越严重，而且也因为路由表的查找与维护也是路由器设计过程中的一个难点。最后两章概要介绍了高级交换技术、路由器安全以及 QoS 技术，这些技术必将在未来的高端路由器中成为主流，也是当前网络领域的研究重点。由于路由器技术，特别是高端路由器技术涉及的领域繁多，比如路由器中实现组播以及光路由等新技术，因此在一本书里不可能给出完整的介绍和阐述。我们只希望本书能够为网络技术的研究与普及作出微薄贡献，其他未能涉及的领域请参阅相关书籍和网络资源。

本书参考了大量的网络资源以及学术论文，还有很多信息来源于论坛，国际上一些网络

设备公司的技术支持中心、一些国际性标准化中心组织的公开文档。另外，作者在国家“863-300”核心路由器项目的测试和验收期间，与 Agilent 中国公司的梁勇高级工程师、信息产业部电信传输研究所的卢彧、袁琦、武静、魏亮、吴江等同志、大唐电信集团的沈祺工程师、巨龙公司的骆璐以及网通集团的吕东芳等工程师的讨论，为本书积累了大量的素材，在此对他们一一表示感谢。

本书由白建军组织编写，第 6 章由张锐执笔，其余各章由白建军执笔，最后由卢泽新教授审阅全书并定稿。王乐春、朱珂、张小哲、张明杰等博士生也为本书提出了很多建议。书中大部分图表都由张锐绘制。

由于水平所限，加上时间仓促，书中欠妥乃至错误之处在所难免，恳请广大读者批评指正。意见、建议和批评请发至电子邮件：jianjun_bai@163.net，作者将不胜感激。

作者

2002 年 3 月

于国防科大计算机学院

目　录

第 1 章　路由器技术概述

互联网络是一个被网络界广泛接受的术语，它表示相互连接的许多网络的集合。每一个独立的网络都有它自己的网络号，此网络号在特定的互联网络中必须是惟一的。路由器的基本作用是根据从网络协议获悉的有关信息，控制通过互联网络的通信量。

在一个有几百台、甚至几千台计算机连在一起的互联网络中，必须有一些约定的方式供这些设备相互访问和通信。随着网络规模的增大，让每一台计算机记住互联网络上其他所有计算机的地址是不切实际的，因此必须有一些机制来减少每台计算机为实现与其他所有计算机通信而维护的信息量。已使用的机制是将一个互联网络分成许多独立但互相连接的网络，这些网络本身可能又被分为许多子网（见图 1-1）。连接这些独立网络的任务可以交给被称为路由器的专用计算机来完成。使用这种方法，网络上的计算机只需记住互联网络中的独立网络的地址，而不需记住网络上的每一台计算机的地址。

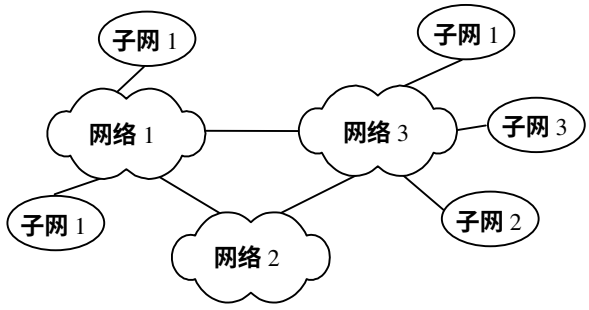


图 1-1　网络和子网的连接

1.1　引　　言

通信网络是由一些系统和节点组成的集合，这些系统和节点负责传输连接在通信网络上的用户之间的信息。在一个网络中主要定义两种系统：端系统和中间系统。端系统是支持端用户应用或者服务的设备，中间系统是连接多个网络并允许这些网络的端系统相互之间进行通信的设备。

简单来说，路由器就是一个中间系统，它主要用来连接两个或者多个网络，这些网络可能是同构的也可能是异构的。一个路由器简单到可以只是一台有两块或更多网络接口卡的微机，从接收接口进来的数据报，经过处理，转发到适当的发送接口。路由器是负责在网络层

对 IP 数据包进行转发的主要设备，它还负责对 IP 数据包进行灵活的路由选择，把数据逐段向目的地转发。它掩盖了下层网络的细节，使各类网络都在 IP 上达到统一。

我们通过描述互联网上的计算机如何相互寻址来引入路由器。说明路由器的作用的最好的类比是邮局服务系统。当邮寄一封信时，需要提供公寓号码、街区名称和号码、城镇和省名。在计算机术语中，发送信息时，需要提供应用端口号、主机号、子网号和网络号。

核心的概念是当邮局接收到发往另一个城镇的信件时，邮政人员首先将它发送到目的城镇所在的分局。从那里，这封信被交给负责特定街区的某个邮递员。最终，这封信被投递到目的地。

计算机网络也采用相似的过程。发往互联网络的信息首先被送到与目的网络相连的路由器。路由器实际上起着这个网络的分发中心的作用，它把信息送到目的子网。最后此信息被送到目的主机的目的端口上。

图 1-2 给出了一个简单的互联网络的系统，其中由路由器将不同的网络连接起来。图中的网络 2、4 上有主机，而网络 5、6、7 上没有。网络 5、6、7 仅仅是与局域网或广域网上的路由器相连。在此网络中，主机 1 和 2 必须配置成同样的网络号（本例中为 4）。除此之外，与同一网络相连的路由器接口必须配置成同样的网络号。

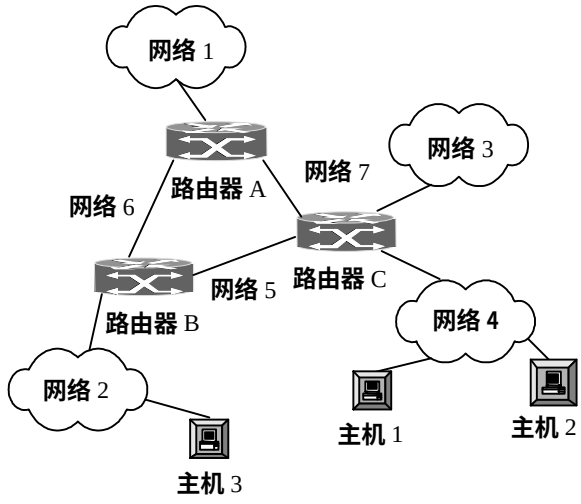


图 1-2　一个简单的网络连接

从本质上说，路由器的作用是将报文分组从一个网络路由到另一个网络。这句话暗含着两个含义。第一，一个路由器的多个接口不能被配置成相同的网络号（但是，通过使用子网掩码，可以在同一路由器的不同接口上配置相同的网络号，但是有不同的子网号）；第二，由于每条路由带有一个目的网络号，故在缺少目的网络号的情况下路由器不能转发路由报文。

1.2　路由器的组成及其功能

路由器工作在 OSI 参考模型的网络层（如图 1-3 所示），完成不同网络之间的数据存储、

分组和转发。它可以根据报文来传输数据，完成网络层路由和转发任务。由于在两个不同网络的网络层之间按报文传输数据时，需要改变两个不同类型网络报文中的第二层地址，即决定在网络之间数据传输时的路由去向，所以称之为“路由器”。

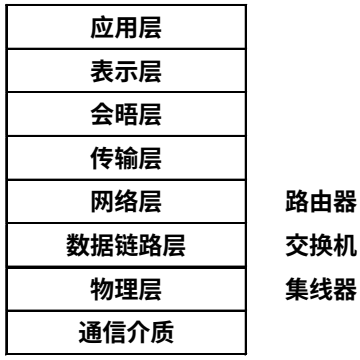


图 1-3　路由器在 OSI 参考模型中的位置

路由器的基本用途是连接多个逻辑上分开的网络，必须具有判断网络地址和选择路径的功能，能够在多个网络互联环境中建立灵活的连接，并可用完全不同的数据分组和介质访问方法连接各种子网。路由器只接受源站或其他路由器的信息，属于网络层的一种互联设备，它不关心各子网使用的硬件设备，但要求运行与网络层协议相一致的软件。

虽然路由器可以支持多种协议（例如 TCP/IP、IPX/SPX、AppleTalk 等协议），但是在我国绝大多数路由器运行 TCP/IP 协议。本书假定路由器主要运行 TCP/IP 协议簇（实际情况也是如此）。目前网络层使用的协议为 IPv4，因为这是最流行的网络层协议，它所涉及的概念与其他网络层协议是类似的。

1.2.1　路由器的基本组成

路由器通常连接两个或多个由 IP 子网或点到点协议标识的逻辑端口，至少拥有 1 个物理端口。路由器根据收到的数据包中的网络层地址和路由器内部维护的路由表决定输出端口以及下一跳地址，并且重写链路层数据包头实现转发数据包。路由器通常动态维护路由表来反映当前的网络拓扑，通过与网络上其他路由器交换路由和链路信息来维护路由表。

我们先以一个路由器模型为基础，介绍路由器的一般组成，后面还会有更详细的描述。一个典型的路由器主要由四部分组成：输入端口、输出端口、交换网络和路由处理器。如图 1-4 所示。

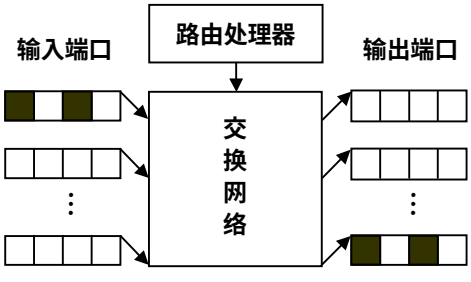


图 1-4　路由器基本组成

1．输入端口

输入端口在线卡上，是物理链路的连接点也是报文的接收点。输入端口的设计遵守物理链路设计标准，完成的功能主要包括：

（1）数据链路层帧的拆封。

（2）在一些路由器的设计中，输入端口拥有一个转发表（是主处理器路由表的一个简单映射），它可以直接在转发表中执行路由查找并把数据报送往输出端口，从而减主处理器的交换负担。早期的路由器都是简单地把所有接收到的数据报由输入接口送给主处理器进行处理，以至于主处理器的处理能力成为整个路由器性能的瓶颈。

（3）为了提供 QoS，输入端口也可能根据预先指定的策略把接收的报文进行分类。

（4）有时，输入端口还要运行数据链路层协议（如 SLIP、PPP 等）或者网络层的协议（如 PPTP）。

（5）把一些特定的控制数据报（利用 RIP、OSPF、BGP 等路由信息数据报）送给路由处理器，由主处理器更新路由表。

2．输出端口

输出端口的主要功能包括排队和缓冲管理，在交换网络以比接口速度更快的速率传送数据报时，这种功能显得尤其重要。输出端口使用复杂的调度算法（例如：RED 算法、WFQ 算法等）实现 QoS 功能，另外输出端口还要执行与输入端口类似的功能——数据报的封装和支持物理链路协议。

3．交换网络

交换网络完成输入端口和输出端口之间的互联。交换网络有从简单的 Crossbar 交换到复杂的 Perfect shuffle、Clos 交换等许多实现方法，现在常用的主要有三种：总线交换、共享存储交换和互联网络交换。

最简单的交换方式是总线交换：当一个数据报到达输入端口时，通过一个共享总线直接被送往输出端口而不用路由处理器的干预，一旦发现总线忙，数据报就会在输入端口排队。这种方式的不足在于因为总线是共享的，路由器的交换能力受限于共享总线的带宽。

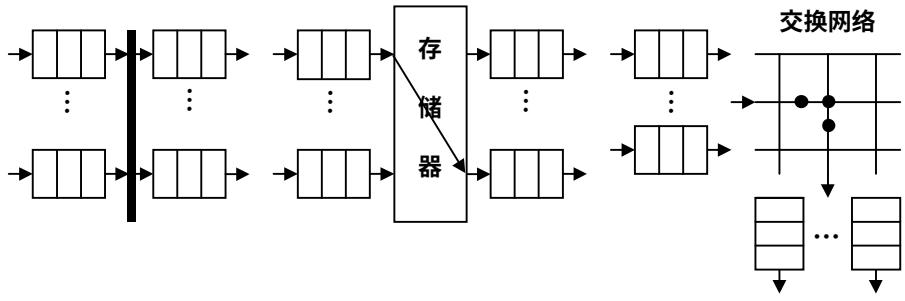
共享存储交换使用一个共享的存储器完成报文的交换，它没有独立的路由处理器，路由器的主 CPU 本身完成路由转发。当输入端口接收到数据报时，它首先中断主 CPU，将接收到的数据报从输入端口缓冲送到共享存储器，处理器经过路由表的查找为该数据报选择合适的输出端口，最后将该数据报送给输出端口。Cisco 8500 系列路由器使用的就是这种共享存储的交换方式。这种方法的不足是交换速度受限于访存速度。

互联网络交换把输入端口和输出端口用多条可用的交换路径连接起来，从而消除了共享总线带宽所带来的限制。一种典型的互联网络交换是 Crossbar。如图 1-5 中所示，Crossbar 通过交换网络把 N 个输入接口和 N 个输出接口连接起来，Crossbar 可以被看作一个由 $N \times N$ 个交叉点连接成的 $2N$ 条总线，当一个交叉点处于“可用”状态时，它所连接的两条总线之间就可以进行数据传输。Cisco 12000 系列路由器就是用这种互联方式进行交换，达到 60Gbit/s 的交换能力

4．路由处理器

路由处理器运行系统软件和各种路由协议，实现维护路由表和计算转发表等功能。其部分功能既可以用软件实现，也可以用硬件实现。对路由处理器的软件与硬件分工的优化也是

影响路由器性能的主要因素。



（a）总线交换　　（b）共享存储交换　　（c）互联网络交换

图 1-5　典型交换网络

可见，只要具有了以上四个基本要素，就可以实现路由器的基本功能。如何把这四个基本要素合理地结合起来，使路由器具有高性能的同时也具有低的价格一直是路由器体系结构开发所关注的问题。随着新硬件的不断上市，硬件成本也不断下降，原来是高不可攀的体系结构实现技术，现在已经有了实现的可能。同时由于网络的作用越来越受到重视，用户也愿意支付较高的投资来消除目前网络的瓶颈，从而大幅度提高整个网络的性能。

1.2.2　路由器的硬件组成及初始化过程

1．硬件组成

目前市场上有大量各种类型的路由器产品。尽管这些产品在处理能力和所支持的接口数上有所不同，但它们都使用一些核心的硬件部件。图 1-6 展示了路由器的关键部件，其中 CPU（或微处理器）的类型、ROM 与 RAM 的大小以及 I/O 端口的数目和介质转换器根据产品的不同会有些相应的变化，但每一个路由器都有图中所示的各个硬部件。通过分析各硬件的功能，就能从整体上了解路由器的工作方式及其所提供的功能。

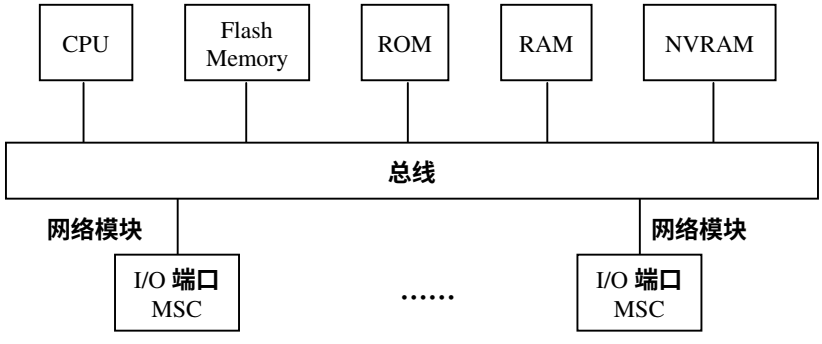


图 1-6　路由器的关键部件

（1）中央处理单元

中央处理单元（Central Processing Unit，CPU），或者称为微处理器，负责执行组成路由器操作系统（OS）的命令和通过控制台（Console）和 Telnet 连接输入的命令。因此，CPU

的处理能力与路由器的处理能力直接相关。

（2）闪存

闪存（Flash Memory）是一种可擦写、可编程的 ROM。在许多路由器上，闪存作为一种选择性的硬部件，负责保存操作系统的映像（Image）和路由器的微代码（Micro Code）。因为修改闪存无需更换和移动芯片，在需要定期修改存储内容的情况下，其代价低，且使用方便。只要空间允许，用户可以在闪存中存储多个操作系统的映像。这项功能对于测试新的映像十分有用。路由器的闪存还能通过使用普通文件传输协议（Trivial File Transfer Protocol，TFTP）将操作系统的映像加载到另一个路由器上。

（3）只读存储器

只读存储器（Read Only Memeory，ROM）中所包含的代码执行加电检测，这一点与许多微机所执行的加电自检（Power On Self-Test，POST）相同。ROM 中的启动程序还负责加载操作系统软件。尽管许多路由器需要软件升级时，只能通过更换路由器系统板上的 ROM 芯片才能做到，但另一些路由器可能使用不同类型的存储方式来保存操作系统。

（4）随机存取存储器

随机存取存储器（Random Access Memory，RAM）用来保存路由表，进行报文缓存等。在许多流量流向一个通用接口时，报文可能不能直接输出到该接口，此时 RAM 可以提供报文排队所需的空间。在设备操作期间，RAM 还能提供保存路由器配置文件所需的存储空间。当 RAM 为缓存 ARP 信息提供空间时，它能减少 ARP 的流量，并增强多个网络连接到路由器时的传输能力。路由器关电时，RAM 中的内容被清除。

（5）非易失的 RAM

非易失的 RAM（Nonvolatile RAM，NVRAM）在路由器关电时，仍能保持其内容。通过在 NVRAM 中保存配置文件的一个副本，路由器在电源故障时可以快速地恢复。使用 NVRAM 后，路由器就不再需要硬盘或软盘来保存其配置文件。因此，路由器中没有移动部件可以延长各部件的使用寿命。计算机系统的许多故障都是因为对移动部件的拆装，如硬盘故障等引起的。

因为路由器没有硬盘和软盘，所以通常的方法是将配置文件存储在一台微机（Personal Computer，PC）上，这样，用户可以使用文本编辑器方便地进行修改。配置文件可以通过网络使用 TFTP 直接加载到 NVRAM 中。当通过路由器来加载配置时，路由器相当于一个客户机，而文件所在的 P C 相当于一台服务器。这意味着用户要在 P C 上安装 TFTP 服务器软件来操作 P C，以便将文件在计算机上进行移入或移出。

（6）输入/输出端口和特定介质转换器

输入/输出端口（Input/Output Port，I/O 端口）是报文进出路由器的连接装置。每一个 I/O 端口都连到一个特定介质转换器（Media-Specific Converter, MSC）上，MSC 提供物理接口到特定类型介质，如以太网、令牌环网、RS-232 连接或 POS（Packet Over SONET）接口等。

根据路由器的数据流，当路由器从网络上接收到数据后，报文的第 2 层报文头被丢掉，并进入 RAM 中。此时，CPU 检查路由表，以决定报文的输出端口及报文的封装方式。这种处理过程称为进程式交换模式（Process Switching Mode）。在进程式交换模式中，每一个报文都必须被 CPU 处理，CPU 要查询路由表并决定往何处发送报文。

Cisco 路由器还有一种交换模式称为快速交换（Fast Switching）。在快速交换模式中，

路由器维护着一个内存缓冲区，其中包含目的 IP 地址与下一跳的接口信息。路由器通过存储以前从路由表中获取的信息而创建这样的缓存。去往特定目标的第一个报文需要 CPU 查询路由表。一旦得到路由信息，在决定去往特定目标的下一跳时，该信息就已经存储在快速交换缓存中。当新的发往该目标的报文到达时，就不再需要查询路由表。这种机制使得路由器交换报文的速度提高，并且路由器 CPU 的负载也会相应降低。

快速交换的变种存在于特殊的硬件结构中——它只在高端路由器模型中才存在。但快速交换的变种与快速交换两者的原理在本质上是一样的，对于所有的交换模式都需要一个包含从目的地址到接口映射的缓存。

还有一种交换模式称为网络流交换（Net-flow Switching），这种方式不仅缓存目的 IP 地址，还缓存源 IP 地址和上层的 TCP 或 UDP 端口等。

对于快速交换还有一些要点需要指出。首先，对路由表或 ARP 缓存的任何修改都会导致快速交换缓存中的内容被强制清除。该动作一般发生在拓扑结构发生改变时，此时快速交换缓存要重建。而且，快速交换缓存中的路由项会随着路由表中内容的改变而改变。快速交换缓存中的路由项与路由表中的相应项要能够匹配。

例如，如果路由器有一个到网络 10.1.1.0/24 的路由，路由器将会缓存目标 10.1.1.0/24。如果路由器只有一个到网络 10.1.1.0/ 16 的路由，路由器将会缓存目标 10.1.0.0/ 16。如果路由表中没有网络或子网的路由项，路由器就使用缺省路由，并使用缺省主网络屏蔽码，同时将缓存目的 10.0.0.0/8。这种方式只工作在对特定目标只有一个路由的情况下。如果有多个代价相同的路径，且无缺省路径，路由器就会缓存整个 32 位的目标。

例如，若目的 IP 地址为 10.1.1.1，而路由器中有两条到 10.1.1.0/24 网络的路由，则系统将缓存 10.1.1.1/32，并将其匹配成第一跳。下一个去往 10.1.1.0/24 网络的路由，如 10.1.1.2/32，将会缓存并匹配成第二跳。如果还有第三条代价相同的路径，则 10.1.1.0/24 网络上的下一个目标将成为第三跳，等等。注意，上述的情形只有在没有缺省路由的情况下才会发生。如果路由器要使用缺省路由来发送报文，则将缓存主网络号，而不是整个 32 位地址。

本质上，路由器使用一种称为轮询（round-robin）的方法缓冲每一个目标的下一跳地址，这意味着路由器的负载基于每一个目的站点进行了平衡。也就是说，因为快速缓存中包含了目的节点与接口的映射，一旦缓存中缓冲了某个表项，其他去往相同目的地的报文就可以使用缓存中的表项内容。路由器不会在快速交换缓存中缓冲同一目的结点的多个接口。

进程交换模式下，路由器基于每个报文进行负载平衡。因为没有快速交换缓存，每一个报文都要经过查询才能发送到后续的接口。如果有多条路径，这种方式就能使得网络的负载更加平衡，但路由器的 CPU 负载却增加了，从而降低了路由器传递报文的速度。

2．路由器初始化过程

打开路由器的电源，路由器执行一系列预定义的操作。路由器所执行的附加操作还依赖于具体设备以及对该设备进行的配置。要了解路由器的初始化过程，首先要分析设备加电时所发生的主要事件。

图 1-7 以流程图方式说明路由器初始化过程所执行的主要功能。路由器加电时，首先执行一系列诊断性测试，以验证处理器、存储器和接口电路能否正常工作。由于该过程是在加

电时进行的，所以通常称之为加电自检（Power-On Self-Test，POST）。

完成 POST 之后，路由器执行启动装载程序。装载程序的主要功能是初始化，并将操作系统映像的一个拷贝拷入主存中。在加载映像以前，装载程序必须首先判断操作系统映像的位置。操作系统映像可能位于 Flash Memory 中或者 ROM 中，甚至可以在网络上。为了决定操作系统映像的位置，启动装载程序要检查路由器的配置寄存器。配置寄存器的值可以通过硬件跳线或软件来设置，这依赖于路由器的模型。寄存器的设置指示操作系统的位置，并定义其他设备的功能，如路由器对来自控制台的 break 键如何响应，以及在控制台终端上是否显示诊断消息等。

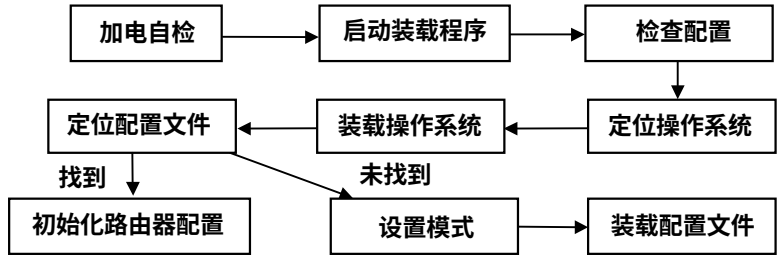


图 1-7　路由器的初始化过程

在大多数情况下，引导寄存器将设置成 2，即路由器将查看配置文件，以决定引导命令。如果什么也没找到，路由器将引导 Flash Memory 中的第一个映像。如果在 Flash Memory 中找不到有效的操作系统映像，或者找不到 Flash Memory，那么路由器使用广播地址发送一个用于请求操作系统映像的 TFTP 请求，以装载 TFTP 服务器上的映像。

一旦配置寄存器检测完成，启动装载程序就知道从何处可以找到操作系统的映像，并将其加入路由器的 RAM 中。操作系统装载完成之后，启动装载程序从 NVRAM 中寻找以前创建和存储的配置文件。

如果找到了配置文件，则配置文件将会被加载到内存中逐行执行，这样路由器就变成可操作的了，且工作是按照以前定义的网络环境进行的。如果以前创建的 NVRAM 文件并不存在，则操作系统将执行一个预定义的、以问题驱动方式进行的配置过程，称为设置对话（Setup Dialog）。一旦操作员完成设置对话，配置信息将存储在 NVRAM 中，以便下一次初始化过程中进行缺省装载。可以通过设置配置寄存器来让路由器忽略 NVRAM 中的内容。该特征用于路由器的密码恢复，此时管理员可以忽略配置文件中的内容。

1.2.3　路由器的功能

根据 RFC 1812，路由器要涉及到链路层（包括 PPP，ARP 等）、Internet 层（包括 IP，ICMP、IGMP 等）、传输层（包括 TCP、UDP，主要是实现一些其他应用协议时需要的功能）以及应用层（包括各种路由协议、BOOTP、SNMP、DVMRP 等）。所有的路由器必须完成两个基本功能：路由处理（routing process）和包转发处理（packet forwarding process）。

路由处理的主要任务是收集网络拓扑信息并形成转发表。包转发处理是根据转发表将报文从输入端口转发到适当的输出端口。

总的来说，路由器通常实现下列基本功能：

■　实现 RFC 规定的各种基本协议，如：IP、TCP、UDP、ICMP 等互联网协议。

■　连接到两个或多个数据包交换的网络，对每个连接到的网络，实现该网络所要求的功能。比如：把 IP 数据包封装到链路层帧或从链路层帧中取出 IP 数据包、按照该网络所支持的最大数据包大小发送或接收 IP 数据包，该大小是网络最大传输单元（Maximum Transmission Unit，MTU）。

■　将 IP 地址与相应网络的链路层地址相互转换，例如将 IP 地址转换成以太网硬件地址。

■　接收及转发数据包，必要时将数据包分段。在收发过程中实现缓冲区管理、拥塞控制以及公平性处理。出现差错时，辨认差错并产生 ICMP 差错及必要的差错消息。丢弃生存时间（TTL）域为 0 的数据包。按照路由表信息，为每个 IP 数据包选择下一跳目的地。

■　支持至少一种内部网关协议（Interior Gateway Protocol，IGP），与同一自治域中的其他路由器交换路由信息及可达性信息，支持外部网关协议（Exterior Gateway Protocol，EGP），与其他自治域交换拓扑信息，当前主要是边界网关协议 BGP-4（Border Gateway Protocol）。动态维护网络路由信息以符合网络当前的拓扑结构，从而能根据路由数据信息为每一个 IP 数据包选路。

■　提供网络管理和系统支持机制，包括存储/上载配置、诊断、升级、状态报告、异常情况报告及控制等。

■　实现网络支持的流量控制和差错指示。支持网管功能，包括网络调测、异常报告以及状态报告等。

1.2.4　路由器与网桥的比较

典型情况下，路由器用于将地理上分散的网络连接在一起，使得将大量计算机连接到一起成为可能。在路由器流行之前，通常使用网桥来达到同样的目的。网桥在小规模网络中表现出色，但在大环境中，就出现了问题。网桥要记住网络上所有独立的计算机的地址。用网桥将大量计算机连接在一起的问题就在于网桥不能理解网络号，因此在网络上任何地方生成的广播都将被发送到网上的每一个地方。

1.3　路由器分类

当前路由器分类方法各异，各种分类方法有一定的关联，但是并不完全一致。

从能力上分，路由器可分高端路由器、中端路由器和低端路由器。各厂家划分并不完全一致。通常将背板交换能力大于 40Gbit/s 的路由器称为高端路由器，背板交换能力在 40Gbit/s 以下的路由器称为中低端路由器。以市场占有率最大的 Cisco 公司为例，Cisco 12000 系列为高端路由器，7500 以下系列路由器为中低端路由器。

从结构上分，路由器可分为模块化结构与非模块化结构。通常中高端路由器为模块化结构，低端路由器为非模块化结构。

从功能分，路由器可分为通用路由器与专用路由器。一般所说的路由器为通用路由器。

专用路由器通常为实现某种特定功能对路由器接口、硬件等作专门优化。例如接入服务器用于接入拨号用户，增强 PSTN 接口以及信令能力；VPN 路由器用来增强隧道处理能力以及硬件加密；宽带接入路由器强调宽带接口数量及种类。

从性能上分，路由器可分为线速路由器以及非线速路由器。通常线速路由器是高端路由器，能以媒体速率转发数据包；中低端路由器是非线速路由器。但是一些新的宽带接入路由器也有线速转发能力。

从路由器在网络中的位置划分，路由器可分为核心路由器、企业级路由器和接入路由器。

路由器分类方法还有很多，并且随着路由器技术的发展，可能会出现越来越多的分类方法。这里我们主要从网络设备的位置这一角度详细介绍各种路由器。

如图 1-8 所示，从路由器在网络中的位置以及网络层次的角度，路由器可分为核心路由器、企业级路由器与接入路由器。尽管这种分三个不同层次安排的结构非常有道理，但它并不意味着企业级路由器不能用作接入路由器或核心路由器。同样，在某些情况中，接入路由器平台也可用作核心路由器或分发层路由器。

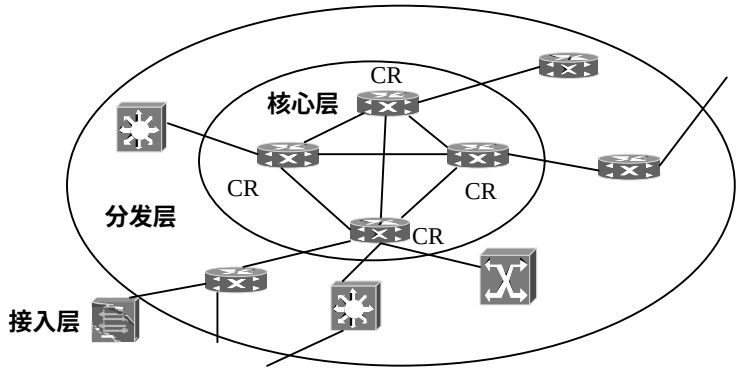


图 1-8　路由器分层网络

1.3.1　核心路由器

核心路由器位于网络中心，也称为骨干路由器。通常使用高端路由器当作核心路由器。核心路由器要求具有快速的包交换能力与高速的网络接口，通常是模块化结构。这些路由器与其他核心路由器相连，为目的地之间提供穿越骨干的多条路径。

核心路由器承担着企业级路由器之间广域网的大部分流量。核心路由器通常通过几个高速接口进行配置，如图 1-9 所示。

Packet Over SONET 技术是核心路由器提供高速核心路由的一种常用方法。在广域网（WAN，Wide Area Network）和城域网（Metropolitan Area Network）中，建立在同步光纤网络（Synchronous Optical Network, SONET）环上并使用 OC-3、OC-12、OC-48 以及 OC-192 连接的骨干很普遍。Packet over SONET 允许在 SONET 网络上直接传递 IP 报文而不必使用 ATM。这就大大刺激了那些没有使用 ATM，但对高速高带宽的骨干有需求的公司。使用 Packet Over SONET 作为骨干只需对路由器投资即可。相比较而言，ATM 不仅要求对路由器进行投资，还要对交换设备投资。

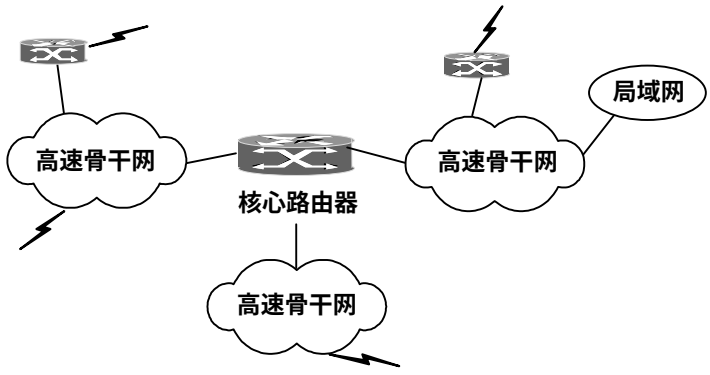


图 1-9　核心路由器

1.3.2　企业级路由器

企业级路由器主要用来连接中型的园区网和企业级网络。为了满足互操作性需求，企业级路由器不仅要支持 IP 协议，还要支持诸如 IPX、Vines 等其他多种协议。同时，随着各种业务需求的不断增长，企业级路由器也要支持防火墙、QoS、安全策略以及 VPN 等。

例如，在图 1-10 中，企业级路由器在一个校园环境中表现为一个核心路由器，但对一个建筑物而言，它又表现为一台接入路由器。或者只把企业级路由器作为一个区域或校园的企业级路由器，仅仅管理核心层和接入层之间的数据传输。

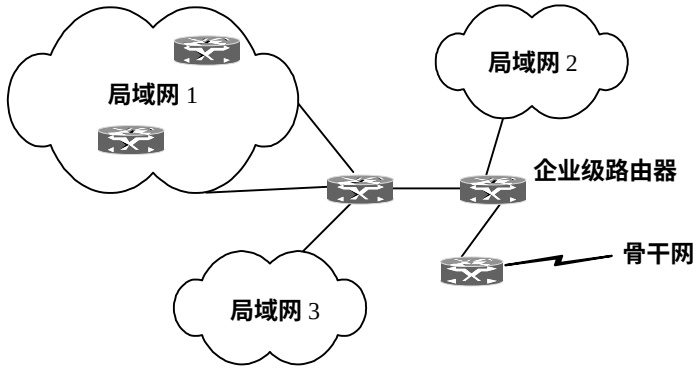


图 1-10　企业级路由器

1.3.3　接入路由器

接入路由器位于网络边缘，通常使用中低端路由器。接入路由器主要负责把端用户和小单位连接到网络服务商（Internet Service Provider，ISP）。这种路由器的复杂性主要在于它要支持各种接入技术（如高速调制解调器、ADSL 等）和各种链路层协议（如 PPTP、IP Sec 等），同时随着 ADSL 及其他高速接入技术的发展，还要支持各种异构网络和更高速的端口。图 1-11 给出了一种接入路由器的使用示例。

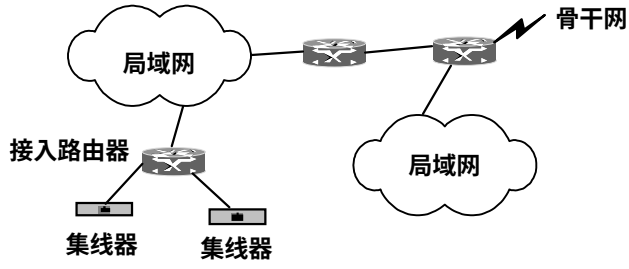


图 1-11　接入路由器

1.4　路由器的关键性能指标

对不同级别的路由器，要求实现的功能以及要达到的性能都不一样，首先列举一般路由器所要达到的一些关键性能指标。

（1）背板能力：通常指路由器背板容量或者总线能力。

（2）吞吐量：指路由器的包转发能力。

（3）丢包率：指路由器在稳定的持续负荷下由于资源缺少在应该转发的数据包中不能转发的数据包所占比例。

（4）转发时延：指需转发的数据包最后一比特进入路由器端口到该数据包第一比特出现在端口链路上的时间间隔。

（5）路由表容量及查找速度：指路由器运行中可以容纳的路由数量。

（6）可靠性和稳定性：指路由器可用性、无故障工作时间和故障恢复时间等指标。

（7）路由协议的一致性、互操作性和健壮性。

而对于高速路由器来说，对各种性能指标要求更高。目前，高速路由器的背板交换能力应达到 40Gbit/s 以上，同时系统即使暂时不提供比 OC-1102/STM-64 接口速度快的接口，也必须有能力在将来无需对现有接口卡和通用部件升级的情况下提供对高速接口的支持。在设备处理能力方面，当系统满负荷运行时，所有接口应该能够以线速处理短包，如 40 字节、64 字节，同时，高速路由器的交换矩阵应该能够无阻塞地线速处理所有接口的交换，且与流量的类型无关。

以下将逐一介绍高速路由器的关键性能指标。

1.4.1　吞吐量

吞吐量指路由器的包转发能力。吞吐量与路由器端口数量、端口速率、数据包长度、数据包类型、路由计算模式（分布或集中）以及测试方法有关，一般泛指处理器处理数据包的能力。高速路由器的包转发能力至少应达到 20Mbit/s 以上。吞吐量主要包括两个方面：

1．整机吞吐量

整机指设备整机的包转发能力，是设备性能的重要指标。路由器的工作在于根据 IP 包头或者 MPLS 标记选路，因此性能指标是指每秒转发包的数量。整机吞吐量通常小于路由器所有端口吞吐量之和。

2．端口吞吐量

端口吞吐量指端口的包转发能力，它是路由器在某端口上的包转发能力。通常采用两个相同速率测试接口。一般测试接口可能与接口位置及关系相关，例如同一插卡上端口间测试的吞吐量可能与不同插卡上端口间吞吐量值不同。

1.4.2　背板能力

背板是输入与输出端口间的物理通路。背板能力是路由器的内部实现。传统路由器采用共享背板，但是作为高性能路由器会不可避免地遇到拥塞问题，其次也很难设计出高速的共享总线，所以现有高速路由器一般采用可交换式背板的设计。背板能力能够体现在路由器吞吐量上。背板能力通常大于依据吞吐量和测试包长所计算的值，但是只能在设计中体现，一般无法测试。

1.4.3　丢包率

丢包率是指路由器在稳定的持续负荷下，由于资源缺少而不能转发的数据包在应该转发的数据包中所占的比例。丢包率通常用来衡量路由器在超负荷工作时的性能。丢包率与数据包长度以及包发送频率相关，在一些环境下，可以在加上路由抖动或大量路由后进行测试模拟。

1.4.4　时延以及时延抖动

时延是指数据包第一个比特进入路由器到最后一个比特从路由器输出的时间间隔。该时间间隔是存储转发方式工作的路由器的处理时间。时延与数据包长度和链路速率都有关，通常在路由器端口吞吐量范围内测试。时延对网络性能影响较大，作为高速路由器，在最差情况下，要求对 1518 字节及以下的 IP 包时延均都小于 1ms。

时延抖动指时延变化。数据业务对时延抖动不敏感，所以该指标通常不作为衡量高速路由器的重要指标。对 IP 上除数据外的其他业务，如语音、视频业务，该指标才有测试的必要性。

1.4.5　突发量能力

突发量是指以最小帧间隔发送最多数据包不引起丢包时的数据包数量。该指标用于测试路由器缓存能力。具有线速全双工转发能力的路由器，该指标值无限大。

1.4.6　路由表容量

路由器通常依靠所建立及维护的路由表来决定包的转发。路由表能力是指路由表内所容纳路由表项数量的极限。由于在 Internet 上执行 BGP 协议的路由器通常拥有数十万条路由表项，所以该项目也是路由器能力的重要体现。一般而言，高速路由器应该能够支持至少 25 万条路由，平均每个目的地址至少提供 2 条路径，系统必须支持至少 25 个 BGP 对等体以及至少 50 个 IGP 邻居。

1.4.7　服务质量

1．队列管理机制

队列管理控制机制通常指路由器拥塞管理机制及其队列调度算法。常见的方法有 RED、

WRED、WRR、DRR、WFQ、WF2Q 等。

排队策略：

- 支持公平排队算法。
- 支持加权公平排队算法。该算法给每个队列一个权（weight），由它决定该队列可享用的链路带宽。这样，实时业务可以确实得到所要求的性能，非弹性业务流可以与普通（Best-effort）业务流相互隔离。
- 在输入/输出队列的管理上，应采用虚拟输出队列的方法。
- 拥塞控制:
- 必须支持 WFQ、RED 等拥塞控制机制。
- 必须支持一种机制，用该机制可以为不符合其业务级别 CIR/Burst 合同的流量标记一个较高的丢弃优先级，该优先级应比满足合同的流量和尽力而为的流量的丢弃优先级高。
- 在有可能存在输出队列争抢的交换环境中，必须提供有效的方法消除头部拥塞。

2．端口硬件队列数

通常路由器所支持的优先级由端口硬件队列来保证。每个队列中的优先级由队列调度算法控制。

1.4.8 网管能力

网管是指网络管理员通过网络管理程序对网络上资源进行集中化管理的操作，包括配置管理、记账管理、性能管理、差错管理和安全管理。设备所支持的网管程度体现设备的可管理性与可维护性，通常使用 SNMPv2 协议进行管理。网管粒度指示路由器管理的精细程度，如管理到端口、到网段、到 IP 地址、到 MAC 地址等粒度。管理粒度可能会影响路由器转发能力。

1.4.9 可靠性和可用性

1．设备的冗余

冗余可以包括接口冗余、插卡冗余、电源冗余、系统板冗余、时钟板冗余、设备冗余等。冗余用于保证设备的可靠性与可用性，冗余量的设计应当在设备可靠性要求与投资间折衷。路由器可以通过 VRRP 等协议来保证路由器的冗余。

2．热插拔组件

由于路由器通常要求 24 小时工作，所以更换部件不应影响路由器工作。部件热插拔是路由器 24 小时工作的保障。

3．无故障工作时间

该指标按照统计方式指出设备无故障工作的时间，一般无法测试，可以通过主要器件的无故障工作时间计算或者大量相同设备的工作情况计算。

4．内部时钟精度

拥有 ATM 端口做电路仿真或者 POS 口的路由器互联通常需要同步。在使用内部时钟时，其精度会影响误码率。在高速路由器技术规范中，高速路由器的可靠性与可靠性规定应达到以下要求：

- 系统应达到或超过“99.999%”的可用性。

- 无故障连续工作时间：MTBF>10 万小时。
- 故障恢复时间：系统故障恢复时间 < 30 mins。
- 系统应具有自动保护切换功能。主备用切换时间应小于 50ms。
- SDH 和 ATM 接口应具有自动保护切换功能，切换时间应小于 50ms。
- 要求设备具有高可靠性和高稳定性。主处理器、主存储器、交换矩阵、电源、总线仲裁器和管理接口等系统主要部件应具有热备份冗余。线卡要求 m+n 备份并提供远端测试诊断功能。电源故障能保持连接的有效性。
- 系统必须不存在单故障点。

1.5　高性能路由器的关键组成

从体系结构上看，自从路由器产生以来，路由器已经经历了多代变化。但本质上还是一台特殊的专门执行报文转发和协议处理的计算机。从功能上看，路由器与计算机还是有较大的区别。这种区别虽然在低档路由器或在路由器发展的初期阶段表现得并不突出，但到了网络系统的规模、速度、种类、应用都已发生巨大变化的今天，这些网络系统本身的变化当然要导致作为网络核心的路由器的体系结构发生巨变。特别是在最新的高档路由器上，这些技术表现得尤为突出。

作为网络中心的核心路由器，与其它类型路由器的主要区别就是在体系结构及其主要组成模块上。如图 1-12 所示核心路由器主要由 4 部分组成：高速接口子系统、报文转发子系统、交换开关子系统和软件子系统，其中，软件子系统又包括操作系统、路由协议、网络管理等。

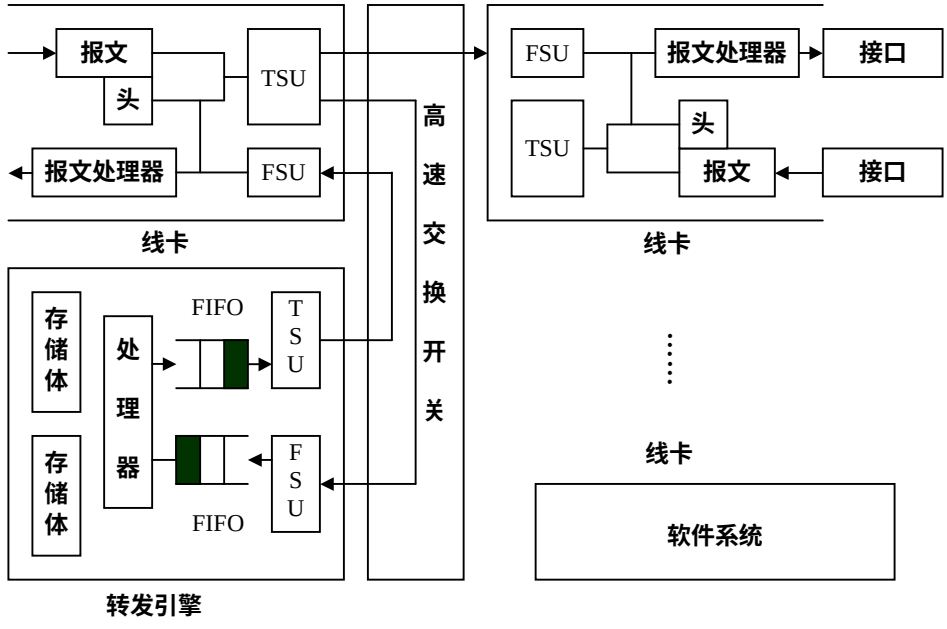


图 1-12　核心路由器的主要组成

1.5.1 高速接口子系统

路由器接口用作将路由器连接到网络，可以分为局域网接口及广域网接口两种。局域网接口主要包括以太网(10M、100M 和 1000M 以太网)、令牌环、令牌总线、FDDI 等网络接口。广域网主要包括 E1/T1、E3/T3、DS3、通用串行口（可转换成 X.21 DTE/DCE、V.35 DTE/DCE、RS 232 DTE/DCE、RS 4410 DTE/DCE、EIA530 DTE）ATM 接口、POS 接口（OC-3、OC-12、OC-48 以及 OC-1102）等网络接口。

当前路由器接口技术较成熟，难点在于高密度接口板的设计与制作和高速接口（大于/等于 10Gbit/s）的实现。

1.5.2 报文转发子系统

核心路由器的报文转发子系统有几种实现形式：线卡负责转发、独立的转发引擎（路由器的各个线卡公用，一般为 1 块或大于一块）负责转发以及在总线性的结构中由主处理器进行转发。

转发任务一般是根据一定的规则在路由表或转发表中查找需要转发的目的地址，如果找到，就把报文转发给该路由表项或者是转发表项所对应的输出端口。目前，高性能路由器中主要是使用独立的转发引擎实现报文转发。

路由器转发引擎需要处理许多任务，例如报文头的封装和拆封、更新每个报文头的 TTL 域、按特定的服务类型将报文分类到不同的队列中等等。占据转发处理大部分时间的主要工作则是从转发表中搜索合适匹配前缀的下一跳信息，即 IP 路由查找。IP 路由查找主要是根据报文的目的 IP 地址进行查找。如果要扩展转发引擎的功能，就应该根据 IP 报文的更多信息，作出转发决策。例如，可以考虑在转发引擎中实现如下功能：支持组播的转发功能，或者基于报文优先级的转发功能，进行 QoS 分组以及防火墙过滤（其中包括基于源和目的 IP 地址的硬件过滤）。

1.5.3 交换开关子系统

通常 IP 路由器从功能上分为两部分。一是数据功能部分，包括前向决策、背板和输出链路调度，目的在于尽可能快地使数据通过路由器。由于处理频率很高，因此常由硬件实现数据功能。二是控制功能，包括路由表信息的改变以及系统配置与管理等内容，由于处理频度低，因此常由软件实现。

不难看出，提高路由性能的关键在于提高路由器的数据交换能力。其中前向决策实现的功能是当数据到来时通过查表得到目的路由，然后将路由信息写入报文，并且同时修改时间戳以及重新计算校验和。背板实现了输入与输出的物理链路连接，使得报文顺利通过背板到达输出端口。对于目的端口冲突的报文需要队列缓存机制，当队列空间不足时会出现报文丢弃或替换。输出链路调度是指对到达输出端口的报文进行规划调度以适应不同服务的要求。

传统路由器的输出部分多采用 FIFO 的规则来保持报文自然的输出顺序，为满足更广泛的服务需要，目前很多先进路由器则将待发的数据报分成不同的流或不同的优先级类，然后

通过精心安排报文调度来满足不同服务的需求。

1.5.4　软件子系统

核心路由器的软件可以说是整个系统的灵魂，它决定着系统的整体性能。一般来说路由器的软件系统主要包括：操作系统软件、协议软件（底层链路协议、传输协议、路由协议）、网络管理软件以及部分应用软件。关键的几个软部件是：操作系统、配置文件、协议软件。图 1-13 给出了一种路由器软件系统的组成（Ip Infusion 公司的路由器软件产品）。

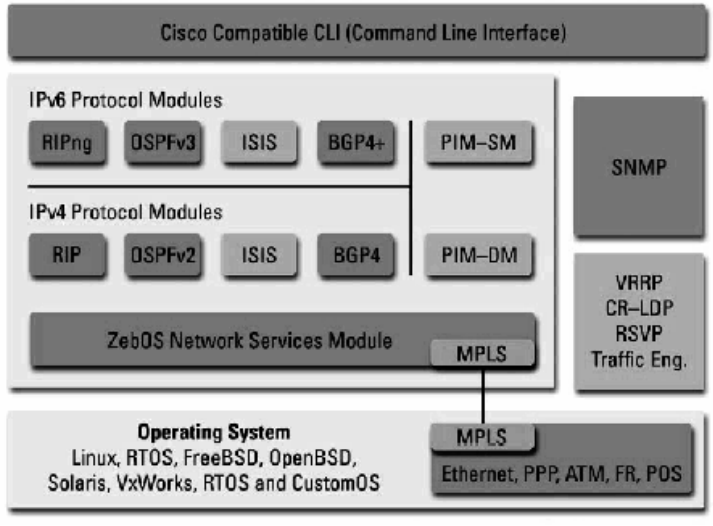


图 1-13　一种核心路由器的软件组成

1．操作系统

操作系统有多种选择，常用的是 VxWorks、Future、Linux、FreeBSD、OpenBSD 等。通常是把各种软件编译成一种操作系统映像。

操作系统映像由启动装载程序定位，这基于对配置寄存器的设置。映像被定位之后，将被加载到内存中的低地址部分。操作系统映像包括一系列的例程，这些例程支持在设备之间传输数据、管理各种网络功能、修改路由表以及执行用户命令等。

2．配置文件

路由器的第二个主要软部件是配置文件。该文件由路由器管理员创建，所包含的语句被操作系统用来执行各种操作系统功能。例如，在配置文件中，可以用语句定义一个或多个访问表，并告诉操作系统将不同的访问表应用于不同的接口上，从而对流经路由器的报文提供一定程度的控制。虽然配置文件中已经定义了影响路由器操作的各个功能，但实际执行这些操作的还是操作系统。操作系统翻译并执行配置文件中的语句。

配置文件中的语句以 ASCII 码存储，所以其内容可以在路由器的控制台终端或远程终端上显示。当用户在连接到网络的微机上创建或者修改配置文件，并使用 TFTP 将该文件上载到路由器上时，这一点很重要。这是因为使用文本编辑器或者字处理工具通常会在保存的文本中嵌入控制字符，而路由器则不认识这些字符。这样，当使用文本编辑器或字处理工具创建或者修改配置文件时，要将文件保存为 ASCII 文本（.txt）文件。一旦配置文件被保存，它

就被存储在 NVRAM 中，并且在路由器每次初始化时加载到内存的高端地址部分。

3．协议软件

核心路由器支持的网络协议很广泛。在这些协议中传输控制协议/互联网协议（Transmission Control Protocol/Internet Protocol，TCP/IP）使用得最多。

TCP/IP 协议栈主要包含以下协议：

- IP 访问列表（IP access list）。
- IP 安全选项（IP Security Option, IPSO）。
- IP 计费（IP accounting）
- 简单网络管理协议（Simple Network Management Protocol, SNMP）。
- 串行线路接口协议（Serial Line Interface Protocol, SLIP）。
- 地址解析协议（Address Resolution Protocol, ARP）。
- 反向地址解析协议（Reverse Address Resolution Protocol, RARP）。
- 域名服务系统（Domain Name System, DNS）。
- Internet 控制报文协议（Internet Control Message Protocol, ICMP）。
- Internet 组管理协议（Internet Group Management Protocol, IGMP）。
- 用户数据报协议（User Datagram Protocol, UDP）。
- Telnet。
- TN3270。
- 简单文件传输协议（Trivial File Transfer Protocol, TFTP）。
- 采用 MD5（Message Digest 5）算法的路径认证。
- IP 访问控制列表（Access Control List, ACL）的违背日志。
- 基于策略的路由。

以及最主要的几种 IP 路由协议：

- 路由信息协议（Routing Information Protocol, RIP）。
- RIP-2。
- BGP-4，边界网关协议版本 4。
- 协议无关的组播（Protocol Independent Multicast, PIM）。
- 中间系统到中间系统（Intermediate System-Intermediate System, IS-IS）。
- 下一跳路由协议（Next Hop Routing Protocol, NHRP）。

鉴于篇幅所限，本书在后面的章节只对 TCP/IP 协议栈中几种主要的路由协议给予详细介绍，这几种路由协议是目前各种路由器产品，特别是核心路由器必须实现的。

第 2 章　路由器技术的发展与相应的国际标准

传统的 IP 网络在传输实时等业务时的主要瓶颈在于路由器速度太慢，时延和时延抖动太大，不能满足 QoS，因此研制高速路由器的任务显得非常重要。传统路由器采用软件实现路由识别、计算和包转发,但即使是昂贵的高档路由器的包转发速度也不到 1Mbit/s,远远低于网络交换机的速度。自 1997 年下半年以来，一些公司开始陆续推出采用硬件专用电路(ASIC)进行路由识别、计算和转发的新型高速路由器。

随着设计技术的进步，各种标准化工作也在同步进行。包括 Cisco、Juniper 在内的各大网络设备提供商以及 IETF 等国际标准化组织与团体，纷纷参与路由器的标准化工作。大多数技术都已经形成 RFC 文档，虽然有一些技术还不太成熟，但几个大的厂家使用的技术也成了潜在的标准。

本章首先介绍路由器发展过程中出现的各种主流技术，然后介绍伴随着技术的发展，标准化工作主要取得了哪些主要成绩。

2.1　路由器的发展

路由器发展的不同阶段，先后出现了不同的主流技术，包括路由器的体系结构、接口技术、路由协议、网络管理等技术。这些技术都随着时代的变迁被升级或者是被替代。

2.1.1　传统路由器

传统路由器通常是基于总线和集中处理器结构的，其处理能力一般是几十万个包／秒，最大的吞吐能力约 1Gbit/s 左右。在处理一个 IP 包时必须首先接受并缓存整个包，然后解析报头的各个字段，再根据 IP 地址查找路由表转发 IP 包。如果 IP 包有差错，还要进行差错处理。这个过程约需要几百甚至几千微秒。

速度瓶颈主要在于路由表查询、输入输出调度选择等。另一方面，网络规模的进一步增大需要路由器支持大量的端口，然而一般路由器只支持 10 个端口，即使大型路由器也只能支持 50 个端口。为了克服端口的限制，常采用的堆叠式配置，无论在成本上还是在性能上代价都很高。同时在纯路由器网络中，每一个端口即使是内部中间端口都需要占用 IP 地址。

当用户数量急剧增加时，路由器性能也随之而下降，这时路由器虽然可以保障优先级较高的数据传输，但由于它采用无连接的 IP 协议，因而不能让服务质量（带宽、优先级等）与

商业上的优先级对应起来。

规模极大的路由器网络大多采用分层结构，并在大的节点采用可堆叠式配置，以支持大量用户接入。由于 IP 包在沿途的每一个路由器上都需要进行排队和协议处理，因此分层路由器结构和堆叠式配置使得 IP 包需要经过更多的路由器数，这将导致传输时延增加和性能下降。当前 Internet 所使用的 IPv4 协议对实时业务、灵活的路由机制、流量控制和安全性能的支持不够，地址资源也短缺。

因此传统路由器的工作模式，甚至它的结构必须改良，否则将严重制约未来高速网络的发展。

2.1.2 现代路由器技术

1997 年下半年以来，一些公司开始陆续推出采用专用集成电路（ASIC）进行路由识别、计算和转发的新型路由器。由于其速度很快，像导线传输一样，被称为线速路由器或高速路由器。

高速路由器一般由 I/O 线路卡、交换核心和路由处理器组成。每个端口接一个 I/O 线卡，由 ASIC 构成，负责数据包的进、出排队和路由识别；交换核心负责数据包的转发；路由处理器负责路由计算、QoS 等。这种高速路由器的结构与第二层网络交换机相似，它兼有第三层路由包转发功能和第二层交换功能。

目前 G 比特级高速路由器已经投入商用，商品化的 G 比特级高速路由器的交换速度一般为 10～60Gbit/s，包转发速度为 10～60Mbit/s，其延时和延时抖动为微秒数量级，在网络中即使通过多级高速路由器，其积累的延时仍可以保持在可接受的范围内。此类产品的杰出代表是 Cisco 公司的 12000 系列 G 比特级高速路由器以及国产“银河玉衡”核心路由器。

Cisco 的 12000 系列 G 比特级高速路由器（GSR）是 Cisco 公司新一代的 Internet 路由产品成员。它的目标是扩展 Internet 和企业主干网，使其达到 OC-3/STM-1、OC-12/STM-4、OC-48/STM-16 以及 OC-192/STM-64 的速度水平。GSR 可支持不同速度级，在 5-60Gbit/s 的速度范围内，为 IP 数据报提供了 G 级的带宽交换能力。具有高级阻塞管理能力和多信道广播以及服务质量 QoS 特性；与现有设备兼容。GSR 的高速分布式路由体系结构与最新的交换核心相结合，能以千兆比特的速度提供传统的 3 层路由服务。

2.1.3 T 比特路由器

随着密集波分复用（DWDM）系统的开发，网络的传输带宽以惊人的速度增长。网络带宽的增长很有可能远远超过摩尔定律所描述的芯片运算速度增长的幅度，使得制约网络设计、服务质量、高级网络应用的因素是计算机而不是网络本身。这一点将深刻地影响未来的通信网络设备的设计。

当前的 G 比特级高速路由器所采用的体系结构还不能支持未来新一代网络的性能需求。如果使用的光纤接口速度超过了路由器本身的容量，那么路由器将无法将光纤提供的原始带宽全部有效地转换为可用带宽。高速路由器需要提供 T 级的交换速度才能够经济有效地伸缩到更高的接口速度。服务供应商需要 T 级高速路由器，它能够为今天的 OC-3、0C-12、OC-48 和 OC-192 提供路由和 QoS 功能，而同时具有将来支持更高速度的能力。几个路由器厂商正在加紧研制 T 比特的高速路由器，Avici 已经生产出了样机。

与 G 比特高速路由器的结构不同，为了达到 T 比特路由速度，T 比特路由器一般采用并行或大规模并行计算技术。一般 T 比特路由器都是由多个交换路由卡组成，每块卡的功能与 G 比特路由器的整体功能相似，将交换功能也置于卡中。这样，T 比特高速路由器的背板功能是随插卡的数量而扩充的，每增加一块模板，总体背板交换能力增加 60Gbit/s～120Gbit/s，总体背板能力因而可达 Tbit/s 级。

如 Avici 公司的 TSRT 交换路由器，由 20 块交换路由卡组成，每块卡采用专用硬件构成，其交换机构的交换速度为 70Gbit/s，总体交换速度为 1.4Tbit/s。Pluris 公司的 TNST 级路由交换机采用大规模并行技术，共有 16K 个处理机结点，每个处理机结点支持 155Mbit/s 的端口，采用通用器件构成，总交换速度为 5Tbit/s，它支持 1000 个 2.5Gbit/s 的 STM16 端口。

目前已经实现的 T 级交换路由器（TSR），可以将 16 个 OC-192（即 10Gbit/s ATM/SDH）接入，通过 TSR 汇集，在一根光纤上以 WDM 原理传输，可达到 320Gbit/s 的高带宽。随着 21 世纪的到来，可以相信这个数字将会和目前的 10Gbit/sOC-192ATM 一起，以四的倍数增长。

T 级高速路由器的适应性和可扩充性，从根本上解决了网络设备的能力限制和不断迅速增长的 Internet 需求之间的矛盾。同时由于将 G 比特高速路由器、交换机都压缩到一块模板上，成本得以大幅下降。

2.1.4　未来光交叉连接路由器

由于光通信技术的发展和用户带宽需求的增长，多种业务在多个层次进行复用，日益表现出处理开销增加，网络管理复杂，难以适应用户要求等现象，如 IP/ATM/SDH/Optical 的组网方式或 IP/SONET/Optical 的组网方式，就需要布置 IP 网络设备、ATM 网络设备、SDH 网络设备和光网络设备等。人们希望新一代 Internet 技术只要部署 IP 设备和必要的光设备，高速路由器便可直接进入国家骨干网，成为国家核心网。也就是 IP 网络和协议能够与光网络紧密结合，由原来支持 SDH/SONET 到主要支持 IP，实现光 Internet。为了达到这一点，需要新的技术，如新的光接口、新的网络控制协议、Lucent 的 SDL 技术、Cisco 的 DPT 技术和 OIF(Optical Internet Forum)正在研究的技术。同时，DWDM 及其相关技术的飞速发展和商用化，光互联网已经成为未来 Internet 发展的一个重要趋势和方向。光互联网的一个重要问题是如何解决 IP over DWDM，最为重要的设备便是需要有路由功能的光交叉连接设备（OXC），可以称这种路由器为光交叉连接路由器。图 2-1 描述了一种基于光交叉连接的智能路由器结构。

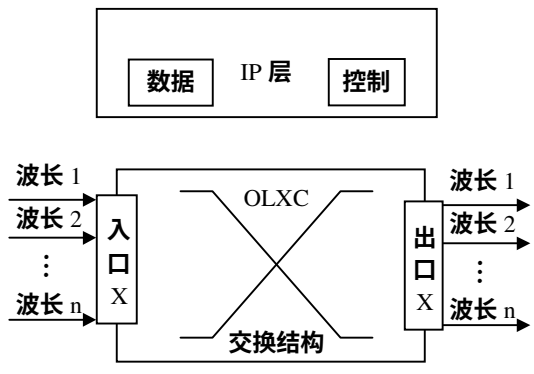


图 2-1　一种基于光交叉连接的路由器结构

它包括电路由器的功能和光层交叉连接的功能。IP 层负责完成电域的一些较为复杂的控制管理功能，包括：寻址、路由、全网拓扑发现、业务量工程、服务质量控制、对光资源的管理（光路的建立和波长的分配）、网络故障恢复等。光层交叉连接器（OLXC）负责连接入口和出口之间的高速光通道，它通过交换式的光通路来提供高速动态的连接。IP 层拥有光层的资源，对之进行分配和管理，负责各光交叉连接路由器之间信令和控制消息的处理，而光层只需要负责简单的传输即可。

2.1.5 主动网络路由器

随着主动网络技术的发展和进一步成熟，主动网络路由器的需求不可避免。这种路由器需要提供可编程的能力，能够执行分组里携带的代码，或者根据每个分组里提供的特殊信息对分组进行不同的路由选择或转发处理。主动式路由器可以在现有路由器硬件结构的基础上添加数据分组主动处理功能而构成。图 2-2 描述了一种高性能主动式路由器体系结构。

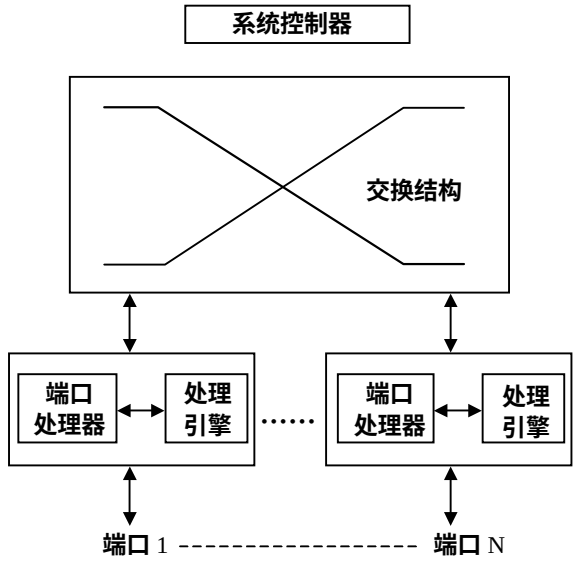


图 2-2 一种主动式路由器体系结构

每个物理接口上包括一个端口处理器（Port Control，PC）和一个处理引擎（Process Engine，PE）。PC 完成对分组的分类、缓存和调度，主要由一个分组分类和排队的处理芯片来完成；PE 通过一个主动处理芯片来完成对分组的个性化处理，主动处理芯片通常包括一个或者几个带动态随机存储器的处理器，处理器之间、处理器与队列控制器之间、处理器与片外动态随机存储器之间通过主动处理芯片上的 I/O 通道进行通信。可编程的 PE 可以用现在已经基本商用的网络处理器（Network Processor，NP）实现。NP 是一种芯片级的系统设计，通常把处理器、内存、输入输出做在一个单独的 ASIC 芯片上。NP 可以实现从数据链路层到应用层的编程处理，可以提供更多的个性化、智能化、主动式的服务。

2.1.6 未来的发展

目前，对高性能路由器的要求一般有：线速报文处理、高速交换背板、高密度高速接

口、电信级可靠性（99.999%）、稳定可靠的路由软件、大容量路由表、支持一定的 QoS（如支持流量工程）等等。基于这些要求，目前的骨干路由器一般采用多处理器（分布式）式结构或大规模并行式结构，主要问题是路由表的快速查表和高速的报文转发及对 QoS 的支持。

提高路由查找性能的关键是减少查找次数，即减少完成一次查找所要访问的存储器次数，与此同时为了降低成本应尽量减少存储器容量，算法本身也应该便于硬件实现。针对上述考虑目前有多种实现方案。这些方案都在寻求查找速度与存储器容量间的折衷方法。

传统保存路由表的数据结构是树。为了提高路由表查询的速度，可以对数据结构和算法进行优化。采用压缩树的方法来缩小路由表的大小，从而可将路由表放在处理器的一级 Cache 中。也可以通过扩大使用的内存来存储路由表。Stanford 大学的 Gupta P, Lin S, McKeown 等人提出了一种使用两个表的方案：把 32 位的目的地址分为前 24 位和后 8 位两部分。长度小于等于 24 位的所有可能的路由前缀都存在第一个表中（2 的 24 次方项）；对于要查找的路由前缀，若长度小于等于 24 位，则在第一个表中直接查找，大于 32 位的，通过前 24 位在第一个表中找到一项，将该项和后 8 位一起在第二个表中查找。根据统计，骨干网路由器路由前缀 99.93%是不大于 24 位的，因此这种方法的效果较好。因为绝大多数路由查找只需一次访存，最坏也只需两次访存。另外，还有提出利用通用计算机的两极 cache 来减少访存时间的方法。

解决报文高速转发的交换结构目前主要有共享内存、交叉开关。共享内存的实现简单，也可以达到较高的吞吐率（如 Juniper 公司的 M40 就使用了共享内存的方法），但可扩展性较差。交叉开关的难度是调度，还可能发生阻塞。第一种阻塞是线路头部阻塞，这可以通过 VOQ(virtual output queuing)技术：在每个输入端都对所有的输出端排队来解决。另两种阻塞分别是输入阻塞和输出阻塞，这通常是通过增大交换开关的交换速度来解决的。

根据输入输出和交换结构的相对速度不同，排队可分为输入排队和输出排队两种。如果交换结构的带宽大于所有的输入端口的带宽的总和，那么报文在输出端排队，是输出排队类型的，否则报文在输入端排队，是输入排队类型的。输出排队的调度算法要求输出端的接口速率是输入端的 N 倍（N 是端口数），因此目前多以输入排队的算法来取得高性能，而且输入排队的实现较简单。

对 QoS 的支持也是难点。对 QoS 的支持通常使用基于 WFQ 的调度算法，而目前所有的基于 WFQ 的调度算法都要求使用输出排队或共享内存，而高性能又要求输入排队和交叉开关；这个矛盾还没有得到很好的解决。Stanford 大学的 MicKeow 等学者正在研究一种可扩展的并行的很好地支持 QoS 的 T 级路由器——Fork Join 路由器，在输出和输入端都进行排队，以期能很好地支持 QoS。

目前，路由器主要有三种发展趋势：

（1）越来越多的功能以硬件方式来实现，具体表现为 ASIC 芯片使用得越来越广泛；

（2）放弃使用共享总线，而使用交换背板，即开始普遍采用交换式路由技术；

（3）并行处理技术在路由器中运行，极大地提高了路由器的路由处理能力和速度。

此外，高级交换体系结构、高速接口、大容量路由表、安全、抗抖动能力、支持更高级的 QoS 和交换协议以及全光网络中的光交换都是未来高速路由器的发展方向。

2.2 路由器设计标准

IETF 于 1995 年制定了 IPv4 路由器的总体技术要求，对路由器所涉及的各层的技术都给出了统一的标准。随着技术的不断发展和应用的不断更新，有些技术标准还处于逐渐发展和完善的过程，目前业界还没有统一的设计与实现标准，主要是根据当前大的生产厂商所用技术为主。

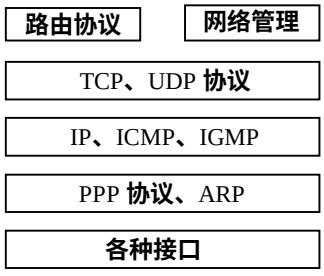


图 2-3 路由器各层模型

在路由器中，要涉及到链路层（包括 PPP、ARP 等）、Internet 层（包括 IP，ICMP，IGMP 等）、传输层（包括 TCP、UDP，主要是实现其他一些应用协议时需要的功能）以及应用层（包括各种路由协议，如 BOOTP、SNMP、DVMRP 等），如图 2-3 所示。这里我们简要介绍 Internet 层以下的标准，IP 层上的主要有 TCP、UDP、各种路由协议以及应用协议。后面的章节对几种路由协议都给与了详细介绍，应用协议不是本书重点，因此不再赘述。在书后的附录里我们给出了 IP 路由器各个层次在实现时应该遵循的标准 RFC 列表。

IETF 要求的标准有以下几种等级：

- 必须（Must）：毋庸置疑，一定要照办；
- 必须实现（Must Implement）：要实现，但缺省状态下不一定要激活；
- 决不能（Must Not）：绝对不能这么做；
- 应该（Should）：一般情况下要这么做，但可能存在特殊情况可以不这么做；
- 应该实现（Should Implement）：一般情况下要这么做，但可能存在特殊情况可以不这么做。实现后缺省状态下不一定要激活；
- 不应该（Should Not）：一般情况下不可这么做，但可能存在特殊情况可以这么做；
- 可以（May）：可选项。

2.2.1 链路层实现要求

1．链路层与 Internet 层的接口

链路层应该向 Internet 层提供数据包的源物理地址。链路层必须传送给 Internet 层的信息包括：

- IP 数据包。
- IP 数据包中的数据长度。
- 收到数据包的物理接口标识。
- 数据包的目的物理地址的分类（链路层单播、组播、广播）。

Internet 层应该向链路层发送数据包的链路层优先级，Internet 层必须发给链路层的信息包括：

- IP 数据包。
- IP 数据包的长度。
- 目的物理接口。
- 下一跳 IP 地址。

2．地址解析协议（ARP）

路由器实现 ARP 协议时必须遵循 RFC 1122 规范。链路层绝不能因为 ARP 缓存中没有目的地址的对应 ARP 项，而向 IP 层发送“目的地址不可到达”信息，它应在发送 ARP 请求并等待应答的同时，把少量的 IP 报文暂时排列等候。若这样仍无效，才可向 IP 层发送“目的地址不可到达”。路由器绝不能接受任何声明另一主机或路由器的物理地址为广播或组播地址的 ARP 应答。

3．MTU

每个逻辑接口的 MTU 必须在合法的取值范围内可配置。路由器绝不能认可那些会导致发送帧长超出链路层协议所许可的最大帧长度的 MTU 配置。但是，应该能够接收帧长等于、甚至大于 MTU 的包。

4．以太网和 802.3 并存

连接到 10M 以太网的路由器必须与 RFC 1122 中的以太网的要求相一致并且应该无条件的一致。

5．PPP（Point-to-Point Protocol）

实现点到点或通用串行接口的路由器必须实现 PPP（但缺省状态下可不激活）。路由器上的通用串行接口必须支持 PPP。可以允许线路被配置为使用非 PPP 的点到点线路协议。点到点接口应该在激活时缺省使用 PPP 协议，或者要求在激活前配置链路层协议。通用串行接口在激活之前也应该配置所用的链路层协议。

PPP 协议族规范详见 RFC 1332、RFC 1333、RFC 1334、RFC 1172、RFC 1661 以及后来由发布的一些扩展。一条 PPP 链路在工作之前首先应用链路控制协议（Link Control Protocol，LCP）协商链路的配置并进行链路的测试，然后用 IP 控制协议（Internet Control Protocol, IPCP）协商在 PPP 上传送的网络层协议配置。这些过程结束后才可以在 PPP 链路上传送 IP 数据包。LCP 选项之一是告诉对方本地能够接受什么样的机制，而不是告诉对方本地希望发送什么。

（1）LCP 选项（Link Control Protocol Options）

LCP 提供多种可协商选项，包括地址和控制域压缩、协议域压缩、异步字符映射、最大接收单元（Maximum Receive Unit，MRU）、链路质量监控（Link Quality Monitoring，LQM）、“magic number”（幻数，用于环路检测）、密码认证协议（Password Authentication Protocol, PAP）、询问握手认证协议（Challenge Handshake Authentication Protocol，CHAP）和 32 位帧校验序列（Frame Check Sequence，FCS）。路由器可以在同步或异步链路上使用地址/控制域压缩和协议域压缩。能够接受这些压缩选项的路由器也必须能够接受未压缩的 PPP 包头信息。但并不表示要发送带压缩的报文。

普通的 PPP 报文头包含地址域、控制域、协议域。其中地址域为 0xFF（广播），控制域为 0x03，表示“无编号信息”（Unnumbered Information），协议域占 2 个字节，说明数据区域的内容，协议域压缩后为 1 个字节。

路由器必须为异步 PPP 链路协商 ACCM（Asynchronous Control Character Map，ACCM），但不为同步链路协商 ACCM。若路由器在同步链路上收到 ACCM 协商请求，它回送确认（ACK），然后忽略掉。还应该协商 MRU 和 LQM。即使路由器协商后得到的 MRU 小于 1500 字节，它也必须能够接收 1500 字节的帧。

路由器应该实现并协商用于回路检测的幻数（Magic Number）选项，支持认证选项（PAP，和/或，CHAP），以及支持 16 比特 CRC 帧校验序列。可以支持 32 比特 CRC 帧校验序列。

（2）IP 控制选项（IP Control Protocol Options）

路由器可以提供 IP 地址协商，也必须接受对端对 IP 地址协商的拒绝。

6．接口测试

路由器必须提供允许寻路软件决定物理接口或永久虚拟链路（Permanent Virtual Circuit，PVC）可用性的机制，以及允许寻路软件判断物理接口线路质量的机制。对于因管理操作而产生的物理接口变为可用/不可用，以及因任何原因引起的链路级接口变为不可用/可用时，路由器必须能够通知寻路软件。

2.2.2 网络层 IP 协议实现要求

网络层的协议主要有 IP，ICMP 和 IGMP 等协议。ICMP 和 IGMP 的报文要放到 IP 数据报中传送，因此从结构上看它们位于 IP 协议之上，但实际上它们和 IP 协议一起构成完整的网络层协议族。

本节和后面一节主要介绍 Ipv4 路由器在实现 IP 和 ICMP 协议时应该遵循的基本规则，网络层协议栈的其他协议实现规范这里也不再重复，有兴趣的读者可以参考 RFC 1812。

1．IP 协议

路由器必须实现 RFC 791 定义的 IP 协议以及其必选的扩展，这些扩展包括：子网（RFC 950）、IP 广播（RFC 922）、无类域间路由（RFC 1912，CIDR）以及 RFC 1122 中规定的 IP 包分段/重组、IP 组播等。

（1）IP 地址

IP 地址分为 5 类，分别为 A 类、B 类、C 类、D 类和 E 类地址，其中 E 类地址为保留地址，D 类地址为组播地址。A，B，C 类地址作为通用单播地址网络前缀，在现在的 CIDR 环境中它们的分类已经没有意义。组播地址有 28 位，可以是永久性的也可以是暂时的。永久组播地址由 IANA 负责分配，而临时组播地址是动态分配给临时组的。成员关系是通过 IGMP 协议动态确定的。

在讨论单播地址时，地址的表示方式为二元组（<网络前缀>，<主机号>）。网络前缀由管理者分配，这样才能保证该值在设备所连接路由域中是惟一的。对于网络前缀和主机号有一些特殊的保留值，这些保留值不能被单个用户使用。

除了初始化过程，路由器产生的数据报的源地址必须使用它的某个 IP 地址（不能是组播或广播地址）。路由器必须能够接收和处理广播目的地址的报文，把该报文使用的广播地址当作寻址到某个路由器一个 IP 地址。收到任何源地址错误的 IP 报文时，均默默丢弃。源地址检查由 IP 层或（适当的时候）由传输层协议完成。

IP 路由器应该满足 RFC 1112 对主机关于 IP 组播的要求，并支持所有直连网络上的本地 IP 组播。如果已经存在 IP 组播地址到链路层地址的映射，就使用该映射地址。可以提供配置选项，设置为使用链路层广播地址。在点到点连接和所有其它接口上，组播被封装为链路

层广播。对本地 IP 组播的支持包括：产生组播数据报、加入组播组、接收组播数据报、退出组播组，即支持 RFC 1112 全部内容，包括 IGMP。

除了上述规则以外，还要进行地址检查。地址检查包括源地址的有效性检查以及无效地址的过滤。

■　源地址的有效性检查：实现基于在收到数据报的源地址和逻辑接口转发表之间的对比功能。如果实现了这种功能，当路由器发现收到数据振的接口不是向该数据报中的源地址转发的接口时，它将数据报丢弃。

■　无效地址的过滤：有一些 IP 源地址是无效的（在 IP 地址分类时有相应的规定），路由器必须对这些无效地址进行过滤。路由器不应该转发具有无效 IP 源地址或网络前缀为 0 的源/目的地址的数据报。除了环回测试外，不应该转发具有网络前缀为 0 的源/目的地址的数据报。路由器可以具有一个开关允许网络管理关闭这种检查，但在缺省状态下必须进行这种检查。

如果由于无效地址的过滤将丢弃数据报，路由器应该记录数据报的源地址、目的地址，如果问题是由源地址引起的，还要记录收到该数据报的物理接口、发送数据报的主机或路由器的链路层地址。

（2）数据报的分段与重组

数据报要在物理网络上传送，它的长度必须符合物理网络对数据段长度的要求。因此路由器必须支持 IP 数据报分段（详见 RFC 791）。对 IP 数据报进行分段时，应该使分段数最小化。如果数据报的长度过大，就必须首先被分为几段，然后才发往物理网络进行传送，这个过程就称为数据报分段。具体的分段策略请参见 RFC 1812。

为了消除或减少数据报的分段，最好能事先预知从源到目的地路径上的 MTU，RFC 1191 描述了动态获取 MTU 的技术。当路由器产生 IP 数据报时它使用 RFC 1191 中描述的方案限制数据振的长度。

除了支持 IP 数据报分段，路由器还必须能够重组收到的 IP 数据报。在转发一个 IP 数据报之前，路由器绝对不能对其进行重组。也有一些特殊情况需要路由器重组需要转发的数据报。

2．IP 数据报的包头

在 IP 包头有两个未用比特，一个在 TOS 中，一个在标志字段中。路由器绝对不能在自己产生的报文中将这些未用比特置为 1，并且也不能因为接收到的报文中未用比特为 1 而丢弃该报文，即不检查未用比特的值。

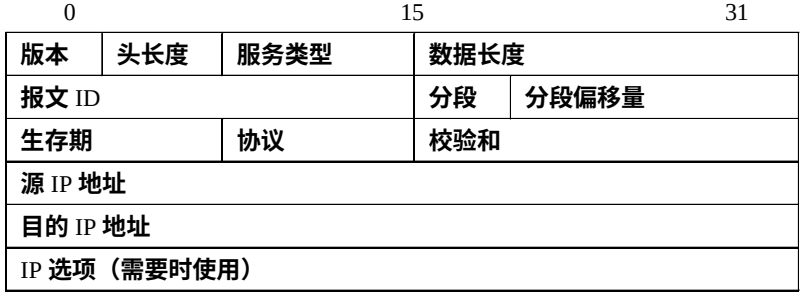


图 2-4　IP 报文头

（1）TOS（Type of Service，服务类型）

IP 头中 TOS 字节分为 3 部分：优先级（高 3 位）、TOS 位（4 位）、保留位（最低位）。更多关于 TOS 及其用途的讨论参见 RFC 1349。

当路由器决定如何转发数据报时应该考虑数据报包头中的 TOS 字段。路由器必须为它的路由表中的每一条路由维持一个 TOS 值，若路由器从一个不支持 TOS 的路由协议得到路由，此路由的 TOS 设为 0。

在为一个目的地址选择路由时，路由器选用和下面的算法等价的一种算法：

- 路由器在它的路由表中查找去往目的地址的所有可用路由。
- 若没有路由，路由器将数据报丢弃，原因为目的地址不可达。
- 若有多个路由的 TOS 和数据报的 TOS 匹配，路由器根据策略选择最佳路由。
- 否则路由器重复上一步，除非找到 TOS 为 0 的路由
- 若上几步，路由器未找到路由，它将丢弃该数据报，原因为目的地址不可达，返回的 ICMP 差错代码为 11 或 12。

（2）TTL（Time to Live，生存期）

TTL 有 2 个功能：限制 TCP 片的存活时间、终止互联网络路由环。需要特别注意，路由器只能在转发数据报时检查 TTL 值。路由器绝对不能产生或转发 TTL=0 的数据报。也不能仅仅因为接收到的数据报的 TTL 为 0 或 1 而将其丢弃，如果该数据报是发给路由器自身的，它就是有效的，因而路由器必须尝试接收。

对于路由器产生的报文，IP 层必须提供一种机制使得传输层可以设置每一个数据报的 TTL 字段。如果使用固定的 TTL 值，那它就必须是可配置的，且数值应该超过 Internet 的典型“直径”（即最长可能路径），现在建议此数值超过 Internet 直径的两倍，以适应 Internet 的增长（穿过美国的报文通常要经过 15～20 个路由器，缺省 TTL 要超过 40，建议为 64）。

尽管 TTL 的单位为秒，但也有跳数的概念。每个处理数据报的路由器必须至少将 TTL 值减 1。如果数据报在本路由器停留时间超过 1 秒，路由器可以按实际的秒数减少 TTL 的值。TTL 超时会导致报文被路由器丢弃。高层协议可能会希望设置 TTL 值以实现 Internet 资源的“扩展环”搜索。这种方式被某些诊断工具使用，如使用 IP 组播来试图定位某种类型的“最近”路由器。传输协议也会定义自己的最大数据报存活时间的 TTL 值。

若 TTL 的值已经减为 0，必须丢弃该报文。对于发往组播地址的数据报文，若路由器预测到下游的某个路由器将会把 TTL 减为 0，它就必须将此数据报丢弃，以更有效地实现 IP 组播的扩展环搜寻算法（RFC 1112）；但对发往单播和广播地址的数据报，路由器绝对不能丢弃。

（3）IP 优先级

路由器的优先级处理基本机制是基于优先级的资源分配。包括基于优先级的排队拥塞控制、链路层的优先级选择等。路由器也可以为它所产生的路由，使用的管理策略以及流量控制策略选择优先级。

① 基于优先级的排队服务

路由器应该实现基于优先级的排队服务。若实现了基于优先级的排队服务，路由器就必须提供配置选项在 IP 层抑制这种机制。可以使用严格优先级排队策略以外的其他流量管理机制，但必须是可配置的，也就是说可以通过配置使用严格优先级排队策略。

② 低层优先级映射（lower layer precedence mapping）

使用优先级排队的路由器必须实现低层优先级映射。对于使用低层优先级映射的路由器：

■　如果链路层定义了链路层优先级，路由器必须具有将 IP 优先级映射为链路层优先级的机制。

■　必须提供为所有的 IP 流量选择缺省链路层优先级的配置选项。

■　应该能为每一个接口配置特定的、非标准的从 IP 优先级到链路层优先级的映射。

③ 优先级处理

对于一个路由器，无论它是否使用优先级排队策略，它都要遵循以下规则：

■　必须接受并处理所有优先级的输入流量，除非管理配置另有规定。

■　可以使用有效性过滤机制从管理上限制特定流量源对优先级等级的使用。如果使用了这种机制，它绝对不能过滤下述 ICMP 差错报文：目的地址不可达、重定向、超时和参数差错。

■　可以使用 cutoff 机制以允许路由器可被设置为拒绝或丢弃低于某一优先级的话务。这种功能可以通过管理动作或与实现相关的启动机制来激活，但必须有配置选项能取消这种无人为干预的启动机制。当数据报由于这种 cutoff 机制而被丢弃时，应该发送 ICMP 目的地址不可达（代码 15）的差错报文，除非配置选项禁止这么做。

■　绝对不能改变不是它产生的报文的优先级。

■　应该能为它所支持的路由或管理协议配置不同的优先级值。

■　必须对和链路层优先级相关的错误指示做出恰当的反应，若一个数据包由于链路层因优先级的原因不能接受它而被丢弃，应该发送 ICMP 目的地址不可达（code15）差错报文，除非配置选项不允许这样做。

（4）包头校验和

路由器必须检查包头校验和，丢弃包头校验和错误的数据包。如果路由器仅仅改变了 IP 数据包的 TTL 值，那么可以使用递增加 1 的方法来更新 IP 包头校验和，以减少因路由器操作而导致未发现的校验和差错的可能性，详见 RFC 1071。

（5）包头选项

接收方对报文中 IP 选项的处理顺序并没有做规定。应该实现修改后的安全选项，详见 RFC 1108。路由器支持源站选路选项，提供配置选项使得它可以丢弃所有的源站选路数据包，但缺省时绝对不能使能此功能。路由器必须可作为源站选路的终点。

通常情况下，对一个源站选路的数据包的正确回应是遍历相同的路由。路由器必须提供途径使得传输层和应用层协议可以反转接收到的数据包的源路由。当路由器不知策略的限制时，它将反转后的源路由插入产生的数据包中。路由器绝不能产生包含多个源站路由的数据包。当路由器产生源站路由选项时，必须正确填写此选项，即使它是通过将一个错误的包含了源站地址的路由记录反转得到的。

IP 报文头还有其他的一些选项，这里就不一一赘述。

3．包头有效性检查

在路由器处理数据报之前，进行下面的包头有效性检查，其中的任何一项失败都将导致数据报被丢弃，不再做进一步的处理，但应该记录差错。

■　链路层报告的 IP 数据报的长度不小于 20 字节。

- IP 的校验和正确。
- IP 的版本号为 4。
- IP 的首部长度字段不小于 20 字节。
- IP 的总长度字段必须足够大，至少应足以容纳 IP 首部。

上面的规定并不排除路由器支持其他版本 IP 的可能性。如果路由器能够区分不同版本的 IP，那它不能将其他版本的数据报视为错误。另外，检查链路层报告的 IP 数据报的长度时，它应该至少和 IP 数据报首部中的总长度字节相等。如果检查发现链路层可能截除过 IP 数据报，此数据报被丢弃，记录此错误，发送 ICMP 参数差错报文，指针指向 IP 总长度字段。

最后，如果 IP 数据报的目的地址不是路由器的地址，那么路由器应该确认数据报中不含严格源站选路和记录路由选项。如果此检查失败，路由器记录此错误，发送 ICMP 参数差错报文，指针指向数据报的 IP 目的地址。

4．路径 MTU 发现

为了消除分片或使分片最小化，要知道从源地址到目的地址的路径 MTU，路径 MTU 是该路径中各跳 MTU 中的最小值。RFC 1191 提出一种动态发现任意一条互联网络路径 MTU 的技术。一条路径若所经过的路由器中有些不支持该技术，则可能无法发现正确的路径 MTU，但会选择尽可能正确的路径 MTU。

当路由器产生一个数据报时，使用 RFC 1191 中提到的方案来限制数据报大小。若到目的地址的路由是通过支持路径 MTU 的路由协议学习到的，则通过路由协议得到的路径 MTU 作为路径 MTU 的初值和最大值。

5．子网

路由器在它的接口配置和路由数据库中都支持变长网络前缀。特定环境中，可能需要支持子网划分，这些子网实现互联的路径不属于该被子网分割的网络的一部分，即支持不连续的子网。

6．转发算法

本小节的目的并不是要求具体实现中必须精确地遵从下面的算法。相反，这一部分只是指出各个步骤之间的顺序关系：

- 路由器在做基于包头内容的任何处理之前必须检查 IP 包头。
- 在处理一些选项时，需要插入路由器的 IP 地址或者它的标示，因而在发送数据报的接口确定之前无法完成这些选项的处理。
- 在确定数据报是否要发给路由器自身之前，路由器不能检查和减少 TTL。
- 需要本地交付给路由器的数据报的首部绝对不能以任何方式改变（除了可能需要插入时间戳外）。因而在判定数据报是否要本地交付之前不能更新 IP 数据报的首部。

（1）通则

下面的转发算法对所有的数据报都适用，无论是单播、组播还是广播。

第一步：从链路层收到 IP 数据报。

第二步：进行包头有效性检查，除了策略中描述的要进行本地交付的分段外，不进行分段的重组。

第三步：进行大部分的包头选项处理。正如下面的“ 下一跳地址的决定 ”部分所描述的那样有一些选项的处理需要在路由决定之后进行。

第四步：路由器检查 IP 数据报的目的地址，以决定要对数据报进行的处理。有三种可能：

- 数据报是发给路由器的，需要本地交付；如果需要，进行数据报的重组。
- 数据报不是发给路由器的，需要排队等待转发。
- 数据报需要同时进行本地交付和转发。

（2）单播

数据报的本地交付在 RFC 1122 中已经有详尽的描述，下面仅讨论需要转发的数据报。如果目的地址是一个单播地址，则：

第五步：路由器通过查路由表决定转发数据报的下一跳地址，同时也决定发送数据报的接口。

第六步：路由器检查是否允许转发数据报。数据报的源地址和目的地址必须通过地址有效性检查，并且通过包过滤。

第七步：路由器检查并减少 TTL。

第八步：完成任何在第三步中无法完成的选项处理。

第九步：进行必要的分段（见分组的分段与重组部分），这一步在第五步之后进行，就决定了同一数据报的不同分段都从同一个接口发送。

第十步：路由器决定下一跳的链路层地址，这一步依赖于链路层。

第十一步：路由器将数据报封装为恰当的链路层帧，并送到相应的队列进行排队。

第十二步：若需要，路由器发送重定向 ICMP。

（3）组播

路由器应该支持 IP 组播数据报，它可以使用静态组播路由或者动态组播路由。组播和单播的不同之处有：

- IP 组播通常需要同时基于源地址和目的地址同时进行。
- IP 组播使用扩展环搜索（Expanding Rings Search）。
- IP 组播作为链路层组播转发。
- 从不为 IP 组播数据报发送 ICMP 差错报文。

下面是 IP 组播的步骤：

第一步：根据数据报首部中的源和目的地址路由器判定是否从正确的接口收到数据报。若不是，将数据报丢弃，不做进一步的处理。判定的算法依赖于组播路由查找算法，在一种最简单的算法——RPF——中，正确的接口就是向源地址回送数据报时使用的接口。

第二步：基于数据报的源和目的地址，路由器决定发送数据报的接口。为了实现扩展环搜索，为每一个出接口定义一个最小 TTL 值，路由器将向每一个最小 TTL 值不大于数据报的 TTL 值的出接口发送一份数据报的拷贝，分别利用下面的步骤。

第三步：将 TTL 值减少。

第四步：完成通则中第三步不能完成的所有选项处理。

第五步：完成所有必要的分段。

第六步：路由器决定链路层封装所用的链路层地址，这一部分依赖于具体的链路层。在 LAN 上使用链路层组播或广播地址。

第七步：将数据报封装为链路层帧，并送到合适的队列排队。

路由器应该满足 RFC 1122 中 IP 组播对主机的要求，同时支持在相连的所有网络上的本

地 IP 组播。当定义了从 IP 组播地址到链路层地址的映射时，使用此映射，也可以配置成使用链路层广播地址代替之。在点到点链路上和其他所有接口上组播都被封装为链路层广播。支持本地 IP 组播包括产生组播数据报、加入组播组、接收组播数据报和退出组播组。这就意味这要支持 RFC 1112 的所有内容。

（4）广播

① 链路层广播的转发

在多数链路层协议中（除 PPP），IP 数据报的封装格式使得接收者可以根据链路层协议首部将组播、广播和单播区分开。这一部分的规则仅适用于可以将组播和广播区分开的链路层协议。路由器不转发任何接收到的作为链路层广播的数据报，除非它对应有 IP 组播地址。如果是对应有 IP 组播地址，我们可以认为是由于缺乏有效的组播服务而使用了链路层广播。

路由器绝对不能转发任何接收到的作为链路层组播的数据报，除非它的目的地址为 IP 组播地址。同时将丢弃（不做任何进一步的处理）所有从链路层收到的未定义 IP 组播和广播地址的数据报。当路由器发送作为链路层广播的数据报时，IP 目的地址必须是一个合法的 IP 广播或组播地址。

② IP 广播的转发

IP 广播地址主要分为两类：有限广播（Limited Broadcast）和定向广播（directed broadcast）。定向广播又可以分为特定网络前缀的定向广播（network-prefix-directed broadcast）、子网定向广播和某一网络的全子网广播。广播的分类依赖于广播地址和路由器对目的网络的子网结构的理解，同一广播，不同的路由器对它的理解可能不同。

7．本地交付的判定

当路由器收到一个数据报时，它要决定数据报是要本地交付、要进行转发还是两者的组合。路由器必须使用下面的规则决定使用这三种方式的哪一种：

（1）若收到的数据报含有尚未终止的源站路由选项，并且选项指针指向的下一跳地址不是路由器任一地址，路由器将不考虑下面的规则转发之（不进行本地交付）。

（2）下面的情况下只进行本地交付，不进行转发：

- 数据报的目的地址和路由器的某一地址刚好匹配
- 数据报的目的地址为有限广播地址{-1，-1}
- 数据报的目的地址为不能进行转发的 IP 组播地址（如 244.0.0.1 等），并且至少有一个和收到该数据报的物理接口相关联的逻辑接口是目的组播群组的成员。

（3）下面的情况下将进行本地交付和转发：

- 数据报的目的地址为 IP 广播地址，它指向（addresses）路由器的至少一个逻辑接口，但不指向和收到该数据报的物理接口相关联的任何一个逻辑接口。
- 数据报的目的地址为允许转发的 IP 组播地址并且至少有一个和收到该数据报的物理接口相关联的逻辑接口是该组播群组的成员。

（4）若收到的数据报的目的地址是除有限广播之外的其他广播地址并且指向至少一个和收到该数据报的物理接口相关联的逻辑接口，数据报将被本地交付。该数据报也被转发，除非传送该数据报的链路使用的 IP 封装使用了不区分广播和单播的封装形式。

（5）剩下的其他情况路由器都将进行数据报的转发。

8．下一跳地址的决定

当路由器要转发数据报时，它要检查是可以将数据报直接发送到目的地还是要发往另外一个路由器进行转发。在后一种情况下就需要路由器决定下一跳的地址。

下一跳选择进程的目标是在 FIB（Forwarding Information Base）中为数据报选择一条最佳路由。从概念上讲路由的查找就是按某种裁剪规则（Pruning Rules）在路由表中选择一个路由。在为一个数据报查找下一跳路由时，遵守以下规则，这些规则必须按下面的顺序应用于 FIB：

■　基本匹配：丢弃通往 IP 目的地址之外的其他地址的路由。具体地讲，就是将路由赋予两个特性，一个是 route.dest，路由的目的地址，一个是 route.length，标明路由的网络前缀长度。IP 目的地址为 ip.dest，按照此规则，路由表中的路由除了最高 route.length 位的 route.dest 和 ip.dest 相同的那些路由，其实的都将被抛弃。

■　最长匹配：在完成基本匹配后，路由器在剩余的路由中选择网络前缀最长的那些，丢弃其他的。

■　Weak TOS：每一个路由都有一个 TOS 属性，在选择路由时选择那些 TOS 值和数据报的 TOS 值相同的路由，如果没有，则选择 TOS 为 0 的路由。

■　Best Metric：每一个路由都有一个 metric 属性和 domain id，在选择路由时，若路由的 domain 相同，则选择 metric 优的那一个。

■　Vendor Policy（制造商采用的特定策略）：上面的策略通常不能满足实际需要，因而制造商们制定特定的策略弥补上面策略的不足。

基本匹配和最长匹配裁剪规则概括了几种特定路由的处理。这些路由按照下面的优选顺序被选用：

（1）主机路由（Host Route）。

（2）分级网络前缀路由（Hierarchical Network Prefix Route）：也就是到达某一个网络的路由，在 FIB 中可能有多条通往同一个网络的路由，在选择时按前缀长度递减的顺序选择。

（3）缺省路由（Default Route）。

如果上面的裁剪规则使用完后没有找到正确的路由，则必须将数据报丢弃并发送适当的 ICMP 差错报文。

在路由查找的最后可能还有几个路由剩余，这时制造商可有其他的策略选择其中的一条路由。其中的一种方法便是负荷分担。但它在某些情况下可能有利，有些情况下可能不利，好的设计者应该提供方法取消这种功能。

9．包过滤和转发控制

为了提供安全性或/和限制网络某一部分的流量，路由器提供包过滤功能。实现包过滤的路由器应该可以配置以决定是转发全部数据包还是根据数据包的源和目的地址前缀转发，或者根据报文的其他属性转发。

支持过滤功能应该提供配置，从而允许路由器具有：

■　准许转发的列表，是准许转发的报文定义的规定。

■　不许转发的列表，是不允许转发的报文定义的规定；

上面的‘报文定义’指的是报文的源和目的地址网络前缀，也可以包括其他的，如 IP 协议类型、TCP 端口号等。除了地址对，路由器允许使用传输/应用层的源和目的端口组合，也允许将数据包丢弃不做任何进一步的处理。路由器发送 ICMP 目的地址不可达报文（code13）

用于对丢弃的报文的应答，同时对丢弃的数据包进行计数并有选择地记录丢弃的数据包。

路由器还应该提供配置功能决定某个物理接口是否可以工作，实现转发控制。对于被 disabled 设为不可用状态的接口有以下选择：

- 丢弃从该接口接收到的除发给路由器以外的数据包，不做任何进一步的处理。
- 绝对不能从该接口发送除路由器自身产生的以外的数据包。
- 绝对不能通过路由协议发布需要通过该接口的路由

10．状态变化

路由器可能有多种状态，应正确地处理它们。

（1）路由器停止转发

当路由器停止转发后，它绝对不能再发布除第三方路由以外的路由信息。但是可以继续接收和使用它的域内的其他路由器的路由。如果转发数据库仍保留，路由器不能停止其中的路由定时器。如果路由器记住了其他路由器发来的路由，也不能停止对这些路由的定时器。它必须丢弃那些定时器已经到时的路由。在不进行转发时，路由器就和一台主机一样，必须遵循对主机的规定。

路由器可以发送 ICMP 目的地址不可达报文，但不应该发送重定向报文。

（2）路由器开始转发

当路由器开始转发时，它加速向通常和它交换路由信息的路由器发送路由信息。

（3）某一个接口失败或被 disable

当某一个接口失败或被 disable 时，此时路由器清除并停止公告它的转发数据库中的使用该接口的路由。同时 disable 所有使用该接口的静态路由。若路由器学到或记住了其他的通往同一目的地的路由和 TOS，它必须为它们选择最佳的替代路由。

这个状态下，路由器使用适当的 ICMP 目的地址不可达或重定向报文应答因接口不可用造成的报文丢失。

（4）某一接口被使能

接口从不能用变为能用之后，路由器必须重新使能使用该接口的所有静态路由。如果路由器学习到了使用该接口的路由时，就拿它和其他的路由进行对比，选择最佳路由。路由器应该加速向通常和它交换路由信息的路由器发送路由信息。

2.2.3 网络层 ICMP 协议实现要求

ICMP 协议为 IP 协议的辅助协议，它完成诊断及差错控制功能。它的报文分为差错控制报文和查询报文两大类。前者包括目的地址不可达、重定向、源站抑制、超时以及参数差错报文 后者包括回送请求/应答、时间戳、地址掩码以及路由器查找报文。路由器必须支持 ICMP 协议。这里简要介绍各种情况下，发送的 ICMP 报文类型。

1．目的地址不可达

路由器因没有到目的地址的路由而无法转发数据包时，产生 1 个码 0（网络不可达）的目的不可达报文。若路由器有到该目的地址的路由，但 TOS 既不是缺省值 0000，也不是该寻址包所要求的 TOS 值，则产生 1 个码 11（因 TOS 网络不可达）的目的不可达报文。

若收到发给路由器直连网络中 1 个主机的包，而路由器确定没有到目的主机的路径时，路由器产生 1 个码 1（主机不可达）的目的不可达报文。若该目的主机存在于直连网络上，

但不是时 TOS 值不是缺省的 0000，也不是由于 TOS 值不是该寻值所要求的值，而是其他原因导致的不能转发，则产生 1 个码 12（因 TOS 主机不可达）的目的不可达报文。

当路由器无法转发或投递 IP 数据报时，它必须能够向源站发送目的不可达报文，并在报文的代码（Code）字段注明无法投递的原因。

2．重定向

当和源主机直连的路由器发现源主机在使用一条非优化路由时，它向源主机发送重定向报文。

如果给路由器本身发起的数据报选择路由时，路由器运行了路由协议，或者路由器和转发该数据报所经接口的转发功能均处于激活状态时，路由器可以忽略 ICMP 重定向。这与对 lnternet 主机的要求是不同的。

路由器决不能产生对“网络”及“网络和服务类型”的重定向报文。但它必须能够产生对“主机”的重定向报文，并产生对“主机和服务类型”的重定向报文。下面三个条件全部满足时，才可以发送重定向报文：

（1）收到报文的物理端口就是要转发报文的端口。

（2）报文的源 IP 地址和下一跳的 IP 地址在同一个逻辑网络上。

（3）报文不含‘源路由’选项。

路由器绝对不能使用从 ICMP 重定向报文学到的路由进行 IP 数据报转发，但当它失去路由功能时它可以表现得和主机一样。

3．源站抑制

路由器不应该产生源站抑制报文，因为这种方式的作用很弱。但是，一旦产生源站抑制报文，就必须有能力限制该类报文的生成速率。路由器可以忽略所接收到的任何源站抑制报文。

4．超时

当路由器要转发的数据报的 TTL 为 0 时，它必须向源主机发送超时报文。当路由器收到 TTL 为 0 的数据报或在重组目的地址为它本身的数据报时，它必须遵循 RFC 1122 的规定。

5．参数差错

对于其他 ICMP 差错报文没有覆盖到的错误，路由器产生 ICMP 参数差错报文向源站报错。“参数差错”报文必须原样不变地带有触发该差错报文的 IP 数据报的头部，或包含指针域所指字节的 IP 选项。

RFC 1122 中定义了参数差错报文的 1 个新变量：code=1，所需选项丢失，用于军事通信中的安全选项丢失。

6．回送请求/应答

路由器必须实现 ICMP 回送（Echo）服务器功能。也就是说当路由器收到回送请求时它要回送应答。同时也必须准备好接收、重组和响应最大长度至少为 576 和所直连网络的 MTU 的 ICMP 回送请求报文。回送功能可以选择不响应发给 IP 广播或组播地址的回送请求。

路由器提供配置选项：当激活时，路由器忽略所有回送请求，不做任何处理。若提供该功能，缺省状态必须为允许回送响应。路由器必须实现发送回送请求/接收回送响应的用户/应用层接口，所有回送响应报文传递给该接口。

回送响应中的 IP 源地址必须和相应的回送请求中 IP 目的地址相同。ICMP 回送请求中的

数据完全包含在回送响应中。如果收到的回送请求中包含记录路由和/或时间戳，该选项被更新为包含当前路由器，并被完整地加入到回送响应中的 IP 头中。这样，所记录的路由才是这个路径。

如果收到带有源路由选项的回送请求，该选项中的路由顺序必须被反转并作为回送响应报文的源路由选项，除非路由器知道该选项会妨碍响应报文的发送。

7．时间戳

路由器可以实现时间戳及其应答功能，时间戳的数值最好为自 0:00 开始的微秒级标准时间。一旦实现了此功能，则：

- 回答每一个时间戳请求，且设计为最小延时
- 收到目的地址为 IP 广播或组播地址的时戳请求报文时，可以丢弃，不做任何处理
- 应答报文使用的源地址必须为相应时间戳请求报文的目的地址
- 当路由器收到的时间戳请求报文中包括源路由选项时，在应答报文中将这些路由反转顺序并作为应答报文的源路由选项，除非路由器知道这样做会妨碍应答报文的发送
- 当路由器收到的时间戳请求报文中包括记录路由和/或时间戳选项时，在应答报文中应该将把当前路由器加入后的更新选项作为应答报文的 IP 选项。
- 如果路由器提供发送时间戳请求的应用层接口，那么当时间戳应答到来时，将其发送到 ICMP 用户接口。

8．地址掩码请求/应答

路由器必须支持接收 ICMP 地址掩码请求报文，并且应答 ICMP 地址掩码应答报文。路由器对每个逻辑接口提供配置选项，以定义路由器是否为该接口提供 ICMP 地址掩码应答。在路由器知道确切的地址掩码前决不能应答地址掩码请求。如果一个物理接口对应有多个逻辑接口，并且这些逻辑接口的地址掩码不同，那么路由器决不能应答从这个端口来的源地址为 0.0.0.0 的 ICMP 地址掩码请求。

路由器检查它所收到的所有地址掩码应答，以判断其中包含的内容是否与它已知的地址掩码信息匹配，如果地址掩码应答表现为出错，就记录此地址掩码和发送者的 IP 地址。绝对不能使用 ICMP 地址掩码应答中的内容决定正确的地址掩码。

当主机启动时，若路由器未工作，主机可能无法得到地址掩码，因此在配置了自己的地址掩码后，路由器可以在它的每一个逻辑接口上广播一个无原由的（gratuitous）ICMP 地址掩码应答。

9．路由器广播和请求

在支持 IP 组播或广播寻址的相连网络上，IP 路由器必须支持 RFC 1256 中规定的 ICMP 路由器查找协议中关于路由器的部分。实现这些规定时，必须包括为路由器规定的所有配置变量以及这些变量的缺省配置。不要求路由器实现 ICMP 路由器查找协议中的主机部分，但在 IP 转发无效时可能对操作有用（如，作为主机工作时）。

10．一些通用规则

若收到未知的 ICMP 报文类型，路由器将其发往 ICMP 用户接口（如果有）或直接将其丢弃，不做任何处理。当产生 ICMP 报文时，路由器必须初始化 TTL。ICMP 应答报文中的 TTL 绝对不能从触发此应答的数据包中取出。

在长度不超出 576 字节的前提下，ICMP 差错报文应该包含尽可能多的原始报文（Original

Datagram)。返回的 IP 头（以及用户数据）必须和所收到的报文一致。通常在检测到错误（如 TTL 小于 0）之前转发时，会对收到的 IP 数据包头部做修改（如 TTL 减 1，更新选项等），而在发送 ICMP 差错报文时，应将 IP 报文头恢复到原始状态，也就是说 ICMP 差错报文所包含的必须是原始报文头，除非路由器没有被要求这么做。

除非另有规定，路由器产生的 ICMP 报文的源地址必须是发送此报文的物理端口的一个 IP 地址，如果没有 IP 地址，用路由器的 route-id 代替。

ICMP 差错报文的 TOS 域应该和触发此报文的数据报的 TOS 域一致，除非这样的设置会导致该 ICMP 报文因无法寻路到目的地而被立即丢弃。如果是这样的话，该 ICMP 差错报文的 TOS 必须设为正常（即 0）。ICMP 应答报文的 TOS 也应该设为和收到的请求报文的 TOS 一致。

接收到以下类型的报文时，绝对不能发送 ICMP 差错报文：

- ICMP 差错报文。
- IP 头确认错误的 IP 数据包，除非有特别规定，允许发送 ICMP 差错报文。
- 目的地址为 IP 组播或广播地址的报文。
- 链路层广播或组播的报文。
- 源地址网络前缀为 0 或无效源地址。
- 分段包中非第一分段的其他段（如，分段偏移非 0 的包）。

发送源抑制报文的路由器必须有能力限制报文的生成速率。也应该有能力限制其它 ICMP 差错报文的发送速率。速率限制参数作为路由器配置中的一部分应该可设置。如何实现速率限制由设计者决定。

第 3 章　路由器关键技术

网络带宽的迅速增长对骨干路由器提出了新的要求。传统路由器转发 IP 包的能力有限，已经无法满足现代高速网络的需求。近年来，国际上对高速路由器技术的研究也日益活跃。高性能路由器是下一代高速网络的核心，自 1997 年第一个吉比特级的路由器面世以来，高速、高性能路由器迅速成为市场瞩目的热点。在世纪更迭的今天，巨大的市场推动力进一步促进了路由器技术的高速发展。

本章对路由器的关键技术作了简要介绍，其中有些技术在后续章节还有进一步的阐述。首先分析路由器体系结构的发展演变，然后对高速路由器设计的几个重要问题进行了探讨。

3.1　体系结构及调度技术

路由器的基本结构包括交换背板、转发判断、输出链路调度、转发路由表管理等几部分，其核心是背板设备。随着 Internet 的不断发展，原有路由器所用的共享式背板 20Gbit/s 的带宽已经无法满足网络的需求，新的高容量的背板结构不断涌现，像采用纵横制交换器的交换式背板、采用三维交换矩阵结构的交换式背板和基于光纤的分布交换矩阵等，其中最大的分布交换矩阵能提供上百太比特的容量。

目前，人们正致力于开发高速、高性能、高吞吐量、低成本的新一代路由器，以满足网络不断发展的需求。新一代路由器内部结构所展现出的主要发展趋势为：第一，越来越多地使用基于硬件的交换和分组转发引擎。CMOS 集成技术的提高使很多功能可以在专用集成电路（ASIC）芯片上实现，原来由软件实现的功能现在可由硬件以更快的速度和更低的成本完成，大大提高了系统性能。第二，向并行处理的方向发展，逐渐抛弃易造成拥塞的共享式总线，采用交换背板结构。第三，进一步发展在光纤连接上进行的线速选路技术，实现吉、太比特速率，为 Internet 过渡到全光网奠定了基础。

3.1.1　第一代单总线单 CPU 结构路由器

最简单的路由器可以由一台普通计算机实现，只需要加载特定的服务软件，并增加一定数量的网络接口卡即可实现。最初的路由器产品是由一台完成特定服务的计算机实现，只是根据应用要求在硬件上进行了部分优化。

最初的路由器采用了传统计算机体系结构，包括共享中央总线、中央 CPU、内存及挂在共享总线上的多个网络物理接口。如 Cisco2501 路由器就是第一代路由器的典型代表，其中 CPU 是 Motorola 的 68302 处理器，具有一个 AUI 以太网接口和两个广域网接口。第一代单

总线单 CPU 路由器的结构如图 3-1 所示。

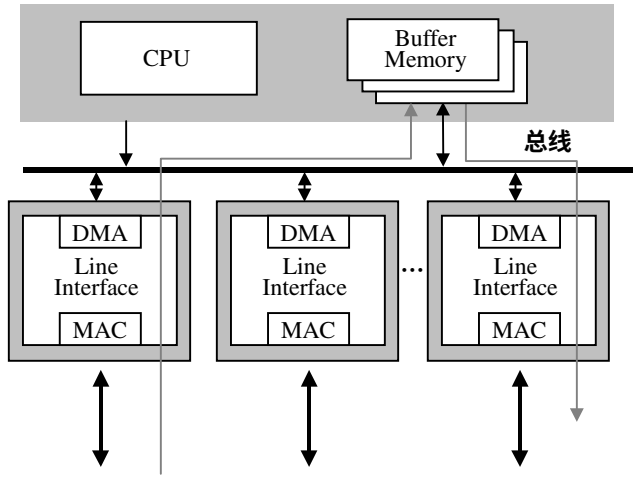


图 3-1　第一代单总线单 CPU 路由器结构

中央 CPU 完成除所有物理接口之外的其他所有功能，数据包从一个物理接口接收进来，经总线送到中央 CPU 中做转发决定处理，然后又经总线送到另一个物理接口发送出去。这种单总线单 CPU 路由器的主要局限是处理速度慢，一颗 CPU 完成所有的任务，从而限制了系统的吞吐量。另外，系统容错性也不好，CPU 若出现故障容易导致系统完全瘫痪。但该结构的优点是系统价格低。目前的边缘路由器基本上都是这种结构。

传统路由器的控制（信令）与转发（数据）是一体的，软件实现上必须涉及两个功能：Routing & Forwarding。路由器完成两项基本功能：

（1）数据转发功能：对每一个经过路由器的用户数据进行处理。这些处理包括接收报文、路由预处理、查表、交换、输出等。

（2）控制功能：周期性地完成对网络路由控制的操作。这些控制操作包括周期性地与相邻路由器交换路由表，以及系统配置和管理等。

为了提高路由器的性能，根据路由器的基本功能划分，在体系结构方面发生巨大变化，用多个 CPU 完成路由器的功能，CPU 按照功能可以分为两类，一种是主 CPU，另一种是多个副 CPU。

3.1.2　第二代单总线多 CPU 结构路由器

根据多个 CPU 的功能不同，第二代路由器有多种形式：最简单的是采用主从两个 CPU 的结构代替了原来仅一个 CPU 的结构，因而较大地降低了 CPU 的负荷，提高了处理速度。路由器的两颗 CPU 为非对称主从式关系结构，其中一颗 CPU 负责通信链路层的协议处理。另一颗 CPU 则作为主 CPU 负责网络层以上的处理，主要包括转发决定、路由算法和配置控制等计算工作。像这种单总线主从 CPU 结构的典型设备有 3Com 公司的 NetBuilder2 路由器等。

主从 CPU 结构路由器的后继是单总线对称式多 CPU 结构路由器。该路由器可以说改善了单 CPU 体系结构中的主要限制，因为它开始采用简单的并行处理技术，即做到在每个接口处都有一个独立的 CPU，专门单独负责接收和发送本接口的数据包，管理接收发送队列、查

询路由表做出转发决定等。而主控 CPU 仅完成路由器配置控制管理等非实时功能。这种体系结构的优点是本地转发/过滤数据包的决定由每个接口处理的专用 CPU 来完成，对数据包的处理被分散到每块接口卡上。该结构路由器的主要代表有北电的 Bay BCN 系列，其中大部分接口 CPU 采用的是性能并不算高的 Motorola 60MHz 的 MC68060 或 33MHz 的 MC68040。第二代单总线多 CPU 路由器的结构如图 3-2 所示。

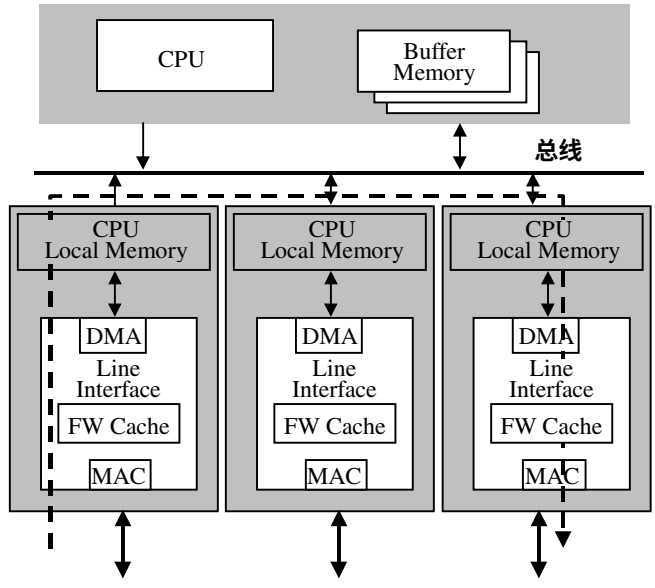


图 3-2　第二代单总线多 CPU 路由器结构

在每一块接口卡上都有本地的 CPU，该种体系结构提供更多的并行性，本地的 CPU 可以完成本地的转发控制，这种机制可以是报文直接发向输出接口，因此，每一个报文只通过总线一次，这样就增加了系统的吞吐率。核心 CPU 只需要维护每一个线卡 CPU 使用的转发路由表，以及整个核心系统的管理功能。这样该体系结构只受两个因素限制：

- 转发处理是由软件实现，因此受限于 CPU 的速度。
- 共享总线的使用，使得每次只有一个报文可以在两个线卡间转发。

针对以上的两个限制条件，对第二代路由器进行了不断改进，过渡阶段使用的体系结构有多总线多 CPU 结构路由器，以及路由+交换的路由器等。多总线多 CPU 结构路由器典型之作为 Cisco7000 系列，在 Cisco7000 中共有 3 类 CPU 和 3 条总线，分别是接口 CPU、交换 CPU、路由 CPU、CxBUS、dBUS、SxBUS。

路由+交换的路由器是指：

- 增加体积较小 Cache，可以做到硬件快速查找。
- 第一个报文通过软件转发，软件查表的同时更新硬件的局部转发 cache。属于该流的其他后续报文，硬件通过查 cache 交换。
- 随着 Internet 规模的扩大，路由器端口增加，主干路由器每个端口上同时存在的流的数目增加，cache 的命中率下降。经过中间过渡体系结构的摸索，路由器发展为第三层交换和交叉开关为体系结构特征的第三代产品。

3.1.3　第三代交叉开关体系结构的 GSR 路由器

在交叉开关体系结构路由器中，数据直接从输入端经过交叉开关流向输出端。它采用交叉开关结构替代共享总线，这样就允许多个数据包同时通过不同的线路进行传送，从而极大地提高了系统的吞吐量，使得系统性能得到了显著提高。系统的最终交换带宽仅取决于中央交叉阵列和各模块的能力，而不是取决于互连线自身。就目前来看，这种方案是高速核心路由器的最佳方案。同时基于交叉开关设计则有更好的可扩展性能。

对第三层 IP 报文在千兆位以太网或 2.5Gbit/s POS 上实现线速率路由转发，利用软件满足该要求是不现实的。第三层路由利用交换技术全部使用硬件完成路由转发。这就是 GSR 路由器，GSR 是指千兆位交换式路由器。交换式路由器将控制与转发相对分离，使软件得以专注于控制路由。该类路由器一般只支持 TCP/IP 协议。其中的控制部分只运行路由协议和管理，不强调实时性要求。目前对于交换式路由器，如果需要获得丰富的附加功能，将丧失部分硬件转发的性能。第三代交叉开关体系结构 GSR 路由器如图 3-3 所示。按照整个路由器中是否存在核心 CPU 又把交叉开关路由器分为两类：主从分布控制交换式路由器体系结构和全分布控制交换式路由器体系结构。全分布控制交换式路由器体系结构如图 3-4 所示。

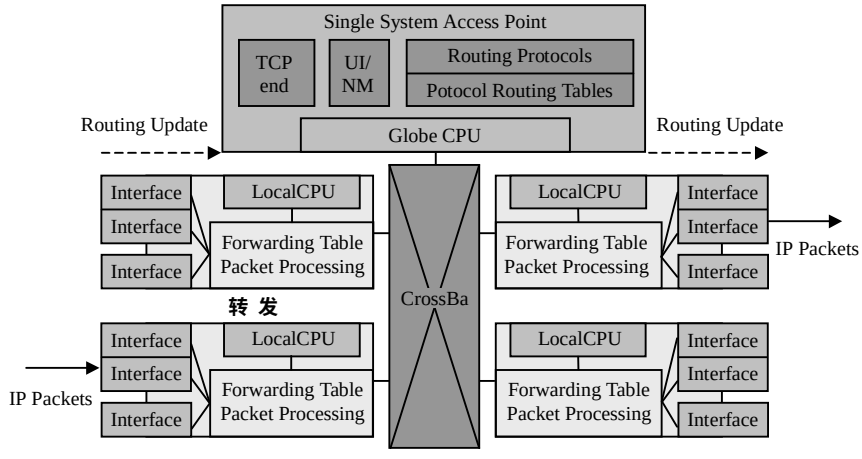


图 3-3　第三代交叉开关体系结构的 GSR 路由器

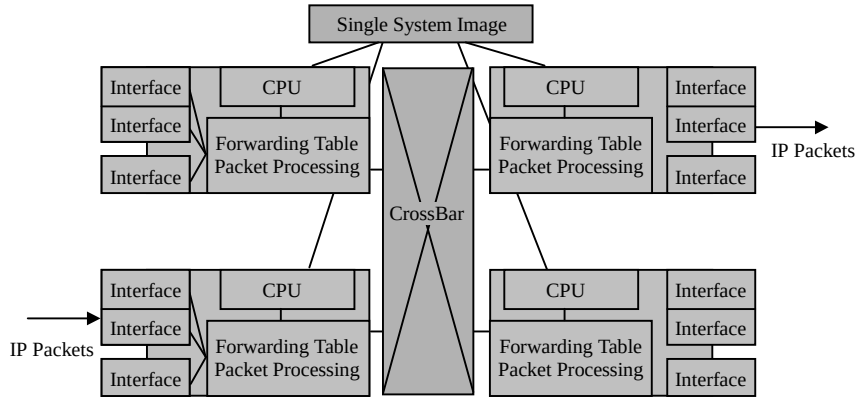


图 3-4　全分布路由器体系结构

由于存在全局核心 CPU，因此该体系结构含有单一系统访问点（single system access point），从而简化了系统设计。同时该体系结构便于系统的适度扩展。在软件方面，部分底层的软件功能也能够达到适度并行。

Cisco12000 就是采用该种设计体系结构。该体系结构在功能上最多可以达到 16 端口 2.5Gbit/s 的 POS 的线速率路由转发。

由于该类路由器中没有核心的 CPU，所有接口卡上都有功能完全等同的 CPU，因此该体系结构的路由器有非常好的可扩展性。同时软件全并行处理，有效地提高系统的性能。但是该体系结构存在着其特定结构实现的难点：

- 全系统的单一映象技术。
- 路由协议的分布处理与单一映象技术。
- 网络管理的分布处理与单一映象技术。
- 系统管理与配置的一致性与单一映象技术等

这些技术和计算机体系结构的一致性问题非常相似，可以借鉴多种计算机体系结构的技术来实现全分布式路由器。

在第三代路由器中，CrossBar 是关键的部件，但是 CrossBar 的交换阵列扩展性很低，要想实现大规模的第三代路由器是非常困难的。同时随着光通信技术的迅猛发展，10Gbit/s 的光信号已经出现，因此要对第三代路由器的体系结构进行重新设计。因此人们在第三代全分布控制交换式路由器体系结构的基础上提出了第四代路由器的概念。它是适应 10Gbit/s 的通信速度的核心路由器。

3.1.4 第四代多级交换路由器

一般路由器只能容纳 8 到 16 个线卡（美国 NEBS 对线卡堆槽宽的要求是不超过 19，从而也限制了线卡的数目只能在 16 个左右）。由于 WDM 技术的发展，单个光纤上复用的信道数急剧增多，对线卡的数目要求也越来越大，因此人们又提出了基于大规模并行或集群式的路由器，这又被称为第四代路由器。第四代路由器和大型计算机的 MPP 结构非常相似。图 3-5 给出了一种可

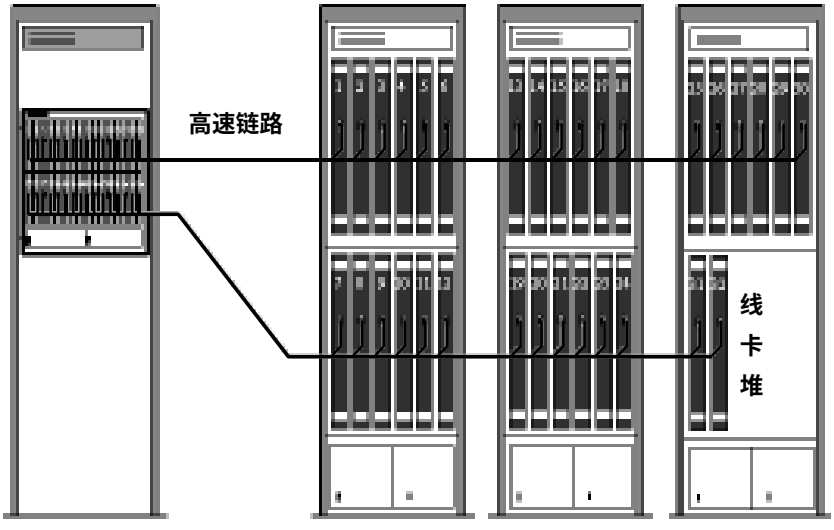


图 3-5 第四代多级交换路由器

扩展集群式的路由器，将线卡堆和交换核心物理上可以分隔得很远，各个线卡堆通过高速链路接到交换核心，线卡堆可根据需要增加，线卡堆和交换核心间通过专门的协议通信。

另一种方案是采用多个完整的高速路由器互连，形成大规模并行式路由器的结构。在这种结构中每个节点本身也是高速路由器，同时提供了外接的光纤卡。多个节点可以通过高速光纤形成 n 维超立方体结构连接，使得每两个线卡间都有多条链路，多个节点对外而言就跟一个路由器一样地工作。

这两种方案的共同特点是可扩展性好，具有很好的容错性，并且容易安放，可以按照需要形成高速乃至超高速的路由器（T 到几百 T 级）。

第四代路由器体系结构要达到的指标是明确的：

- 性能：每块主板具有每秒钟超过 6.4 Terabits 的光交换能力。
- 支持多服务类型：对本地 IP 报文路由，以及同时对本地 ATM 信元交换具有相同功效。
- 低延迟：确保低于 10 microseconds 的延迟。
- QoS：对 IP CBR 支持线速率 QoS，对 ATM CBR 到 IP CBR 映射支持线速率 QoS。
- 多路广播：支持多路广播队列和实现对多媒体大长度报文的高效路由。
- 高可靠性：模块化体系结构，NEBS 一致和载波级可靠性包括软硬件支持的热插拔等。

未来的路由器体系结构的实现需要在以下几方面进行突破：

- 多级交换：流控算法、丢弃算法、路由算法。
- 10Gbit/s 接口：OC-192、10G 以太网。
- 自动保护切换技术：小于 50ms。
- 线速处理 IP 报文：线速报文分类、线速路由查表、线速报文调度。
- QoS 支持：WFQ、RED 等算法的硬件实现。
- 可编程性。
- LVDS：600～800Mbit/s，1～2 米、11.架苣诓炕チ。
- VSR：10Gbit/s，80～200 米、并行光纤传输、机架之间互连。
- 高可靠性设计：机架之间互连 99.999%的可靠性、抗电磁干扰、热备份、热插拔、抗震动以及在温度和湿度变化时性能的稳定性等。

可以明确地说，下一代路由器如果按照常规的体系结构来实现是不能达到第四代路由器的要求的。将来市场上的超级路由器无疑会是吉比特速率的网络互连设备，同时突破了目前路由器实现方法所产生的限制，将多个路由器通过直接连接达到很高的可扩展性和高性能，或者多个路由器结合成“ ganging ”结构。这种结构将产生许多新的问题同时会恶化原有问题。多个设备进行聚合成簇将不能跟上指数级的性能需求，并且缺少必须的关键的体系结构基础支持，以适应对视频和数字信号的统一集中处理。目前迫切需要的是为网络服务提供商提供突破目前核心交换技术的限制，因为现在网络服务提供商的规模已远远超过当前解决方案所能承受的规模限制。

网络的应用要求呼唤新一代路由器的出现，新一代的路由器的出现等待体系结构的突破。

3.2 高速硬件接口

在路由器的设计中，各种接口设备的设计至关重要。因为如果没有高效的、快速的接口设备，路由器就无法正常工作，接口就会成为制约路由器性能的瓶颈。另外核心的交换设备和中央控制设备一般只有一个模式，而路由器所连接的网络可能是多种多样的。为了使网络类型对路由器是透明的，就要使接口能够将各种网络的数据无差别地传输给核心设备，同时还不能成为瓶颈。

3.3 系 统 软 件

路由器技术中最核心的技术是软件技术。路由软件是最复杂的软件之一。有些路由软件运行在 UNIX 操作系统上，有些路由软件运行在嵌入式操作系统上，甚至有些软件为提高效率，本身就是操作系统。全球最大的路由器生产厂家 Cisco 公司曾一度宣称是一个软件公司，可见路由器软件在路由器技术中所占的重要地位。

路由器系统软件一般实现路由协议功能、查表转发功能和管理维护等功能。由于互联网规模庞大，运行在互联网上路由器中的路由表非常巨大，可能包含几十万到几百万条路由，查表转发工作可想而知非常繁重。在高端路由器中上述功能通常由 ASIC 芯片硬件实现。

路由软件高复杂性的另一方面体现在高可靠性、高可用性以及健壮性。实现路由软件的功能并不复杂，在免费共享软件中我们甚至可以得到路由协议和数据转发的实现源码。但是难点在于需要该软件每年 365 天，每天 24 小时高效可靠地运行。

在路由器研制过程中，可以通过购买商用源码等形式迅速实现路由器的软件系统。但是通常认为路由器软件需要一年甚至两年的时间来稳定。路由器的软件系统跟硬件的体系结构有很密切的关系，集中处理路由器的系统软件体系相对比较简单，但效率不高。而分布式处理结构的系统软件就比较复杂，一般都用主从模块化的方法实现。

3.4 路 由 协 议

Internet 以及任何其它类型的通信网络都是由一些系统和节点组成的集合，这些系统和节点负责传输连接在通信网络的用户之间的信息。在一个网络中主要定义两种系统：端系统和中间系统。端系统是支持端用户应用或者服务的设备，中间系统是连接多个网络并允许这些网络的端系统之间相互进行通信的设备。

以 Internet 为例，从 OSI 参考模型的协议层级的观点看，为了掩盖网络层以下各种不同具体网络实现的差异，各主机之间从网络层以上都遵循 TCP/IP 协议族，从而实现网络层以上的互连。所有的上层协议（IP 层以上的，如 TCP、UDP）及应用层协议（如 SMTP、SNMP）都是直接或间接地建立在 IP 这种不稳定的无连接的数据报协议之上的，IP 数据报是整个

Internet 数据传输的基本单位。因而，对于路由器来说，最重要的便是实现 IP 协议报文的处理，也就是说能够根据收到的 IP 数据报的头信息进行正确的处理和转发，这种处理和转发是根据各种路由协议的机制实现的。路由协议决定 IP 数据报转发的关键因素，其中包括路由表查找、路由表建立、路由表维护和路由表更新。路由协议结构如图 3-6 所示。

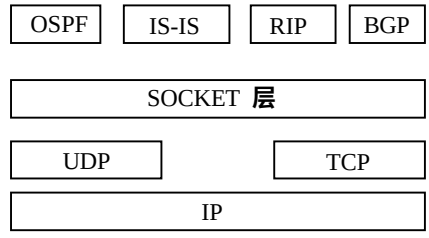


图 3-6　路由协议结构

所谓路由技术，简单而言就是采用一种或多种策略，为数据报文从源地址到目的地址的转发选择一条或几条合适的路径。通过路由器相互间的通信，每个路由器都建立一个路由表，存放网络的路由转发信息。当路由器转发数据分组时，首先根据数据分组的目的地址查找路由表表项，然后按表项中提供的相应下一站的地址对该数据分组进行转发。

实现路由功能也是对路由器最根本的要求，它直接涉及到 IP 数据报路由的正确与否、效率高低，以至整个网络的稳定性。

按应用范围的不同，路由协议可分为两类：在一个 AS 内的路由协议称为内部网关协议 IGP（Interior Gateway Protocol），AS 之间的路由协议称为外部网关协议 EGP（Exterior Gateway Protocol）。路由结构与自站系统示于图 3-7。

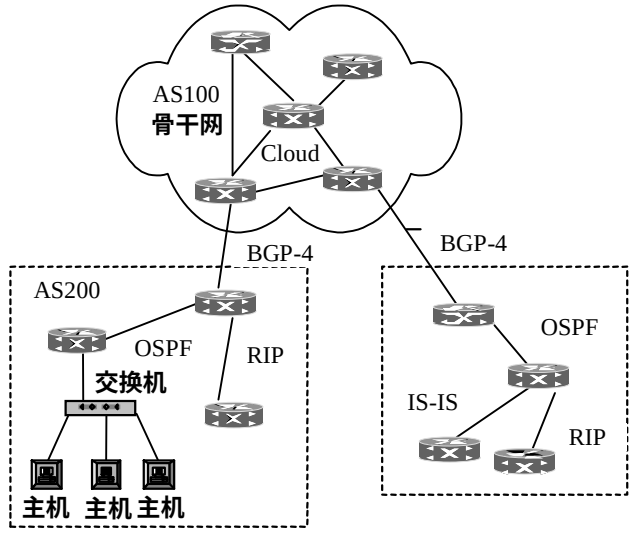


图 3-7　路由结构与自治系统

目前，开放的路由协议主要包含 RIP/RIPv2、OSPF、IS-IS 和 BGP4，另外 Cisco 的 IGRP、

EIGRP 也是使用较多的路由协议。其中 RIP 协议、IGRP 协议、EIGRP 协议采用距离向量算法，IS-IS 协议和 OSPF 协议采用链路状态算法。它们作为域内路由协议，一般用在自治系统内部，在 AS 内部计算以及交换网络可达性消息。对于小型网络，采用基于距离向量算法的路由协议易于配置和管理，且应用较为广泛。对于大型网络，采用链路状态算法的 IS-IS 和 OSPF 较为有效，并且得到了广泛的应用。IS-IS 与 OSPF 在质量和性能上的差别并不大，但 OSPF 更适用于 IP，较 IS-IS 更具有活力。IETF 始终在致力于 0SPF 的改进工作，其修改节奏要比 IS-IS 快得多。这使得 OSPF 正在成为应用广泛的一种路由协议。现在不论是传统的路由器设计，还是已经形成标准的 MPLS（多协议标记交换），均将 OSPF 视为必不可少的路由协议。

外部网关协议最初采用的是 EGP。EGP 是为一个简单的树形拓扑结构设计的。随着越来越多的用户和网络加入 Internet，EGP 的不足渐渐暴露出来。为了摆脱 EGP 的局限性，IETF 边界网关协议工作组制定了标准的边界网关协议 BGP，目前使用的版本是 BGP-4。BGP-5 和 BGP++ 也正在形成与开发之中，主要是增加了支持 IP v6 的功能。BGP-4 协议基于距离向量，是当前 AS 间路由协议的唯一选择。通常 BGP 交换大量网络可达性消息，是 IP 骨干网上的重要协议。

路由协议的实现与路由器软件要求相似，需要实现高可靠、高稳定、健壮性以及安全性。这些协议在后面还会有更进一步的介绍。

3.5 高层交换与 MPLS

随着 Internet 的发展，新业务不断出现，网络数量和规模不断增长，给路由技术提出了新的要求。在过去的 2O 多年中，路由技术和交换技术都得到了充分的发展。但随着网络流量的高速增长，路由器的处理速度低下以及交换机无法适应众多新业务等问题也日益暴露出来。在这种情况下，多种多层交换技术被提出。多层交换基本实现方法均是将 IP 层的智能路由能力与第二层的快速交换能力融合在一起。

与交换机相比，路由器显然能够提供构成企业网安全控制策略的一系列存取控制机制。但是，因为路由器对任何数据包都有一个“拆打”过程，即使是同一源地址向同一目的地址发出的所有数据包，也要重复相同的过程。这导致路由器在高负载情况下不可能具有很高的吞吐量，也成为网络潜在瓶颈之一。端到端传输的性能和服务质量要求对所有连网设备的负载进行细致的均衡，以保证客户机与服务器之间数据平滑地流动。第二层与第三层交换技术，甚至是第四层交换，在解决局域网和互联网络的带宽及容量问题上发挥了很好的作用。这些技术试图减轻传统路由器带来的性能瓶颈，在企业网流量分布偏离 80/20 规则且大多数流量必须跨越子网边界时显得越来越重要。大多数高层交换技术可以总结为“路由一次，交换多次”，或者是基于高性能硬件的线速路由器。

在第九章有对第三层交换更进一步的阐述。

3.6 QoS

Internet 在过去几年所取得的巨大成就和未来所蕴涵的巨大发展潜力几乎没有人怀疑。当

人们在思考未来 Internet 的发展时，如何在 IP 网络上保证用户信息传输的质量就成为一个不容忽视的重要问题。为解决这一问题，QoS（Quality of Service，服务质量）技术便应运而生。更高的带宽及将音频、视频和数据集成在同一种网络媒介上的技术的出现，大大推动了对支持保证服务的多媒体应用的需要。QoS 就是保证服务质量的一种技术，在路由器上实现 QoS 是网络技术发展的一种必然趋势。

QoS 是指一个网络能够利用各种各样的基础技术向选定的网络通信提供更好的服务的能力。这些基础技术包括：帧中继（Frame Relay）、异步传输模式（Asynchronous Transfer Mode，简写为 ATM）、以太网和 802.1 网络、SONET 以及 IP-路由网络。

数据网络利用率的增长使得出现了将各种数据通信量（音频、视频、数据）放置于提供数据服务的底层基础之上的趋势。在试图合并音频、视频、数据网络时，网络工程师面临的障碍是不同类型的通信需要不同级别的网络服务。

路由器的 QoS 能力主要体现在：

- 队列管理机制：队列管理控制机制通常指路由器拥塞管理机制以及队列调度算法。常见的方法有 RED、WRED、WRR、DRR、WFQ、WF2Q 等。
- 端口硬件队列数：通常路由器中所支持的优先级由端口硬件队列来保证。每个队列中的优先级由队列调度算法控制。
- QoS 分类方式：指路由器可以区分 QoS 所依据的信息。最简单的 QoS 分类可以基于端口。同样路由器也可以依据链路层优先级（802.1Q 中规定）、上层内容（TOS 字段、源地址、目的地址、源端口、目的端口等信息）来区分包优先级。
- 分类业务带宽保证：体现路由器是否能对各种业务等级作带宽保证。该指标可以由队列调度算法等方式实现。
- RSVP：RSVP 是资源预留协议，用于端到端路径上资源的预留。使用软状态刷新，是流驱动工作方式。该协议一般不能在大规模全国范围网络上运行。但是通常路由器支持该协议，一些著名厂商使用该协议用于 MPLS。
- IP Diff Serv：区分服务是对 IP 服务质量分级，是对 QoS 的一种简化。
- CAR 支持：CAR 是指承诺接入速率，是一种接入控制。按照与用户签订的协议，对超出承诺速率的数据包做不同处理：丢弃或标记；又称为标记颜色。

3.7 安　　全

由于 Internet 是无中心网，网络之间可以自由通信、自由交互、自由往来，网络安全问题更加突出。在人们追求服务质量的同时，不得不考虑另外一个问题：路由器转发我们的数据时是否是安全的？的确，目前所用的路由器（包括国产的和国外的）几乎都是普通路由器和不具有加密功能的“安全路由器”。这些路由器有的只有路由功能而没有安全、加密功能；有的只增加了包过滤功能，系统在面对各种各样的攻击时非常脆弱。如何保证网络设备自身的安全以及存储在其上或通过其传输的敏感信息的安全，就成为必须解决的问题。

路由器作为内外网之间数据通信的核心设备，其本身的安全性能的重要性不言而喻。强化路由器本身的安全防范往往可以收到事半功倍的作用，保证路由器的安全，也可以说是在

路由器上实现一些安全机制成为网络技术领域一个不可回避的重要问题。安全路由器需要解决的主要问题包括：

■ 防止非授权人员的入侵：非授权人员，从路由器的操作系统入手，从“后门”侵入路由器，从而更改路由表或窃取有用信息，是需要特别加以防范的。

■ 对路由信息和 IP 数据报的加密保护：路由器是网络的转接设备，它不断地接收和发送路由信息和 IP 数据报。因此，路由信息和敏感的 IP 数据报的安全保护就成为极其重要的问题。安全路由器应当具有对路由信息和敏感 IP 数据报的加密保护功能。它涉及加密算法、密钥、数据完整性和数字签名等问题。

■ 防止虚假路由信息的接收和路由器的非法接入：发送虚假路由信息，使路由器路由混乱、阻塞，从而导致网络瘫痪是黑客可用的手段之一,黑客也常常将“自制路由器”接入到网络之中。安全（加密）路由器应当对路由器具有鉴别功能，能够阻止路由器的非法接入。在网络中需要解决和保证虚假路由信息的辨认，对虚假路由信息拒收。

■ 加密算法的选择以及密钥的管理：出于对国家信息安全的考虑和对国外安全产品是否留有“陷门”的忧虑，并遵守国家有关政策、法规，设备应当选择我国有关部门批准的加密算法。在进行加密作业时，则尽量做到一次一密。密钥是加密作业中最活跃的因素，所谓“ 一次一密 ”，简言而之，就是每次加密所用的密钥互不相同。因此，从某种意义上说，保密的关键是密钥，它应考虑以下几个方面：密钥种子、密钥的随机性；密钥的种类和层次；密钥长度；密钥生成算法的复杂度；密钥（含密钥生成算法及密钥自身）保护；密钥的存储、传输、更换和废除以及密钥的管理方式等。

■ 数据完整性验证：严密的设计应当考虑 IP 数据报完整性验证，当然，也可以对传送的密钥做完整性验证，或者两者都做。在某些应用环境或考虑到某些因素的情况下，也可以不进行数据完整性验证。

■ 数字签名：这是对发报者的确认，也是发报者自身严肃性的体现。采用加密技术做数字签名，对很多作业来说是需要的。但同数据完整性验证一样，在某些应用环境或考虑到某些因素的情况下，也可以不进行数字签名。

已经有一些安全技术来增强路由器的安全，比如：访问列表、IPSec、一些加密算法、MPLS 的一些机制、隧道技术和 VPN 等等。但这些技术也只能说是增强路由器的安全特性，任何技术都保证不了路由器的绝对安全。从目前的实现以及未来的发展趋势来看，IPSec 和 MPLS 将会是将来提高路由器安全特性的研究重点。我们会在最后一章深入解释 IPSec 如何为路由器提供安全保护。

IPSec 是一个协议套件，它提供了一种标准机制并定义了一套默认的、强制实施的算法，为 IP 及上层协议提供安全保证。IPSec 的组件如图 3-8 所示。假若想增加新的算法，其过程是非常直接的，不会对共通性造成破坏。为保障 IP 数据报的安全，IPSec 定义了一个特殊的方法，它规定了要保护什么通信、如何保护它以及通信数据发给何人，从而可保障主机之间、网络安全网关（如路由器或防火墙）之间或主机与安全网关之间的数据包的安全。由于受 IPSec 保护的数据报本身不过是另一种形式的 IP 包，所以完全可以嵌套提供安全服务，同时在主机之间提供像端到端这样的验证，并通过一个通道，将那些受 IPSec 保护的数据传送出去。

使用某种 IPSec 协议：封装安全载荷（ESP）或者验证头（AH）对 IP 数据报或上层协议进行保护。其中，AH 可证明数据的起源地、保障数据的完整性以及防止相同数据包的不断

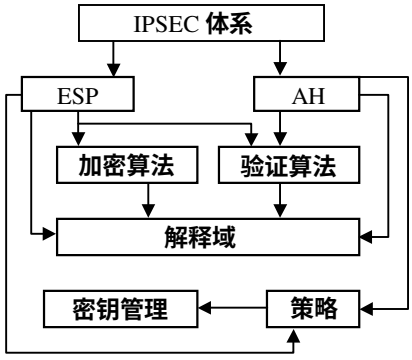


图 3-8　IPSec 的组件

重播。ESP 则更进一步，除具有 AH 的所有能力之外，还可选择保障数据的机密性，以及为数据流提供有限的机密性保障。AH 或 ESP 所提供的安全保障完全依赖于它们采用的加密算法。针对一致性测试，以及为保证不同实施方案间的互通性，定义了一系列需要强制实行的加密算法。这些算法可提供常规性质的安全保障，然而加密技术的最新进展以及摩尔定律的连续证明（观察家认为每隔 18 个月，计算能力便增加一倍），使得默认的加密算法不适合高度密集的数据，也不适合需要必须超长时期保密的数据。

目前，市场上一些所谓的安全路由器一般都是在普通路由器上内嵌或者是外挂一个安全模块，这个安全模块会实现一些加密算法等，但 IPSec 还没有能够实现。这种路由器严格来说还不能认为是安全路由器，因为它们没能从根本上解决或者是实现应用于路由器系统的安全机制。

将加密模块放在路由器里，路由器与加密模块集成为一体与路由器外挂加密机的外挂式体系结构有一些区别。

■　保密性：一般来说，内嵌式路由器结构较外挂式路由器结构好。路由器大都设有多个广域网接口，外挂式结构解决多个广域口的保密问题比较复杂，内嵌式相对来说要简单一些；另外，外挂式的加密在路由器外部进行，内嵌式的加密是在路由器内部进行的；第三，内嵌式比外挂式较为容易实现一次一密。

■　灵活性：内嵌式可以由用户自行设置要加密的客户机和（或）服务器，而外挂式结构则对经过路由器的所有客户机和服务器的 IP 数据报都要加密；从密钥管理来看，内嵌式亦较外挂式优越，在设置分布式密钥管理机制情况下，可不专设密钥管理中心。

■　经济性：由于内嵌式是将加密模块集成到路由器里，而不是像外挂式那样，做成一台独立的机器，因此，成本较低；第二，设计成分布式密钥管理时，可不加配密钥管理中心，系统集成比较经济。

■　效率：内嵌式安全（加密）路由器的加密模块集成在安全(加密)路由器里，可以将加密模块做成硬件，和路由器进行内部交互（这种交互是比较多的）连接，加密速度较快；另外，采用分布式密钥管理机制，不用与密钥管理中心进行交互。因此，内嵌式的效率一般来说要比外挂式高。

第 4 章　路由器交换结构设计

路由器是局域网与局域网、局域网与广域网之间的通信枢纽，是确保各网络之间通信畅通的重要网络设备，其在网络互连设备中的地位正如大型机在计算机界中的地位一样。

高性能计算机体系结构技术经过多年的研究，已相当的成熟，积累了大量的经验和技术。在路由器的研制中，特别是高速路由器的研制过程中遇到了与大型机研究过程中非常相近的技术问题。在当前硬件支持的能力下希望能够借鉴计算机体系结构的经验，通过路由器体系结构的创新大幅度提高路由器性能。我们说路由器本质上还是一台特殊的专门执行协议处理的计算机。因此这种借鉴计算机体系结构的思想是完全合理和可行的。

本章主要讲述路由器报文交换体系结构的主要发展道路，以及新一代路由器所面临的技术难点。在路由器的发展过程中共出现了 4 种典型的体系结构：单总线单 CPU 交换、多个 CPU 共享总线、主从分布式多 CPU 共享总线交换和主从分布式多 CPU Crossbar 交换。在上一章已经介绍过路由器体系结构的发展、分类以及各种典型技术的特点。本章主要是从软件的角度，以 Cisco 系列路由器产品的报文交换结构为例，介绍路由器设计过程中，在报文交换方式上所作的考虑。

4.1　交换体系结构

随着高性能网络的发展，上网人数不断增多，网上用户交互信息量不断增大，同时用户硬件设备性能的增强使得数据传输率也在不断提高，这三个方面的原因导致了互联网上的数据流量正以每月超过 30%的速度增长，网络带宽需求的膨胀给交换路由设备带来巨大压力。

路由器的首要任务是在一个网络段与另一个网络段之间交换报文。如果说调度程序和存储器管理程序是路由器系统基础结构的话，那么其交换体系结构就是路由器基础的基础。路由器系统所采用的交换方法和结构基本上体现了路由器完成其首要任务的好坏程度。交换操作本身是非常简单的：

- 报文来到一个接口。
- 路由器检查该报文的目的地址，并和已知目的地址列表中的地址相比较。
- 如果有匹配项，报文就被转发到适当的接口。
- 如果没有匹配的，报文就被丢弃。

实际上，真正需要解决的问题并不是如何交换，而是如何快速交换。因为报文交换是一个数据密集型的操作而不是计算密集型的操作。加快交换操作的速度并不仅仅是使用一个快速的 CPU 那样简单。其它的因素，例如 I/O 总线性能和数据存储速度，也会对交换的性能有

很大的影响。路由器系统设计者所面临的挑战就是在有限可用的 CPU、I/O 总线和存储技术的条件下，创建出最高性能的报文交换系统。

随着路由网络范围和数量的增长，系统开发者一直在坚持不懈地寻求解决这种性能挑战的更好方法，这也导致系统交换方法的不断重新设计和改善。以 Cisco 系统的 IOS 为例，刚开始开发 IOS 时，只有一种交换方法，就是今天所熟知的进程交换。后来又发布了改进的交换方法。其中有些是基于特定硬件的优化，有些是使用在许多平台上都能很好工作的软件技术。Cisco IOS 在发展过程中采用的交换方法主要有：进程交换、快速交换、自治交换、硅交换、引擎交换、最优交换、分布式快速交换、Cisco 快速转发（Cisco Express Forwarding，CEF）、分布式 Cisco 快速转发（Distributed Cisco Express Forwarding，DCEF）等等。其中进程交换、快速交换、最优交换和 Cisco 快速转发代表了不同时期的主流技术。

Cisco 路由器的每种交换路径的特性都依赖于一个路由是否使用缓冲、缓冲是如何被构建或如何被访问的以及实际的交换发生在什么样的上下文中（进程或中断）。进程交换不缓冲任何信息，只是在一个进程上下文中进行报文的交换。快速交换把转发报文必需的可达性信息和 MAC 头部信息缓冲进一个 radix 树中，并在中断期间进行报文的交换。最优交换把可达性信息和 MAC 头部缓冲进一个多叉树(mrtee)中，也是在中断期间进行报文的交换。CEF 交换把可达性信息缓冲到一个 mtrie 中，而把转发报文需要的 MAC 头部缓部进一个邻接表中。CEF 也是在中断期间交换报文。

我们利用一些 IP 路由的例子说明交换方法，但这些方法也同样能应用于其他的网络协议，比如：IPX 和桥接等。虽然所用的基础结构经常独立于各种协议（例如：IP 和 IPX 就有独立的快速缓冲），但交换结构的内容是类似的，而且每种协议的交换方法都基本相同。

4.1.1　进程交换

进程交换是 IOS 最先采用的交换方法，基本上它是采用强制手段来交换报文。虽然进程交换性能优化程度最低，而且要占用大量的 CPU 时间，但它独立于平台。这使它广泛地应用于所有基于 Cisco IOS 的产品中。进程交换也提供了许多其他交换方法不具有的一些能力，比如通信量负载共享。

进程交换的基本工作过程：

（1）网络接口侦听到线缆上有报文需要处理。

（2）接口硬件收到该报文并把它传向 I/O 存储器。

（3）网络接口向主处理器发出中断，告诉它在 I/O 存储器中有一个报文正等待处理，这个中断称为接收中断。

（4）IOS 的中断处理程序检查报文头部信息（封装类型、网络层头部等等），得知它是一个 IP 报文，随后就把该报文放进输入队列中，等待合适的交换进程处理。

（5）实际的报文转发操作开始决定把收到的报文转发到哪个接口。首先查看路由表，看其中有没有到该报文目的的地址的路由。如果有，就从路由表入口项获得下一跳的地址（路径中下一个路由器或最后的目的地址）。随后查看 ARP 缓冲，以获得构造一个新的下一跳的介质访问控制头部所需要的信息。然后构建一个新介质访问控制头部，把输入报文中的旧介质访问控制头覆盖掉。最后，报文就排队等待到外部网络的网络接口。

（6）输出网络接口硬件侦听到有一个报文等待输出时，它把该报文从 I/O 存储队列中取

出并发送到网络上。一旦输出网络接口硬件结束传输该报文，它就中断主处理器，告知它报文已经被传输出去。IOS 随后就更新它的输出报文计数器，并释放先前该报文所占的 I/O 存储空间。

进程交换的优点是它具有报文负载共享能力。当有多个路由（路径）要到达同一目的地时，每个报文负载共享提供了一个相对简单的、把通信量路由到多条链接的方法。如果有多个路径存在，被进程交换的报文会根据每条路径的路由度量自动地在可用的路径之间分布。

但是，虽然报文通信量共享非常有利于多条链接上的通信量负载平衡，它确实还存在一个不足：它会导致到达目的地的报文乱序。这在有许多可用路由情况下是非常可能发生的，乱序报文的处理会非常严重地影响终端的性能。

进程交换还有一个最重要的不足就是交换速度慢以及由此引起的报文丢失。进程交换需要为每个报文查看路由表。随着路由表的增长，查看路由表的时间也会相应增长，因此交换的时间也增长。递归路由更是需要额外的时间查看路由表（为了解决递归），从而增长了整个查看路由表的时间。

查看路由表时间也增加主处理器的使用时间。虽然这种影响在那些只有很少路径的小网络中并不明显，但大的网络会拥有几百甚至几千条路由。对于这样的网络，路由表的大小会很严重地影响主处理器的利用率以及路由等待时间（报文进入和流出路由器之间的时间间隔）。

另一个影响进程交换速度的主要因素是存储器数据传输时间。在有些平台上，报文被交换以前，交换进程需要把接收的报文从 I/O 存储器中拷贝到另外的存储区。路由处理完以后，报文被传输出去以前又要拷回到 I/O 存储区。存储器数据拷贝操作非常占用 CPU 时间，因此在那些平台上，进程交换的性能非常低。

4.1.2 快速交换

快速交换充分利用了缓冲技术。路由器在缓冲中保存了所预料的内容：目的地址前缀、前缀长度、输出接口、下一跳 IP 地址以及一个 MAC 头部。所有交换报文到一个特定地址必须的数据都包含在了一个入口项中。

缓冲通常是指把一个大的数据集合的一个经常使用的子集存储在一个可以快速访问的本地存储区。例如，计算机可能会把磁盘上的一个文件经常被使用的部分做一个拷贝放在内存中，或者 CPU 为了提高其性能而预取一些指令到一个非常快的相连存储区域。在 IOS 中，快速缓冲用来保存关于报文可达性／接口／MAC 头部拷贝的非常简单的一种数据结构。

为了理解快速缓冲如何工作，考虑前面讨论的进程交换。对进程交换做一点修改，查找完报文的下一跳、输出接口以及 MAC 头部数据之后，再添加一步：把这些信息保存到特殊数据结构里面，这个特殊结构允许快速访问任何基于目的 IP 地址的缓冲入口项。这个数据结构就是快速缓冲。随着时间的推移，在缓冲内部就存储了很多经常用到的目的 IP 地址。

快速交换的基本工作过程：

（1）网络接口侦听到线缆上有报文需要处理。

（2）接口硬件收到该报文并把它传向 I/O 存储器。

（3）网络接口向主处理器发出中断，告诉它在 I/O 存储器中有一个报文正等待处理，这个中断称为接收中断。

（4）IOS 的中断处理程序检查报文头部信息（封装类型、网络层头部等等），得知它是一

个 IP 报文，中断处理软件直接查询快速缓冲，看它里面是否存有到达这个目的地址输出接口的指针。如果有，主处理器就通知输出接口硬件，在 I/O 存储器里已有一个报文准备好传送，然后就终止中断以让其他进程占用 CPU。如果没有，就按进程交换的方法查找。

（5）输出网络接口硬件侦听到有一个报文等待输出时，它把该报文从 I/O 存储队列中取出并发送到网络上。一旦输出网络接口硬件结束传输该报文，它就中断主处理器，告知它报文已经被传输出去。IOS 随后就更新它的输出报文计数器，并释放先前该报文所占的 I/O 存储空间。

实际上，只要缓冲入口项存在，就没有调度任何进程参与快速交换。随着快速缓冲的引入，IOS 在很短的中断期间就能完成整个报文的交换操作。缓冲使 IOS 能够从相对简单的转发一个报文的任务中分离出资源密集的路由决策任务。这样，快速缓冲引入了“路由一次，转发多次”的概念。

快速缓冲入口项在进程交换时构建。因为进程交换构建缓冲入口项，即使能快速交换，到任何目的地址的第一个报文还是将会被进程交换。只要缓冲入口项存在，以后到该目的地址的报文就可以快速交换了。

在少数路由改变、网络大体上稳定并且通信量倾向于在一个特定的目的地址子集之间流动的情况下，用快速交换方法能很好地工作。然而在有些网络环境中，特别是 Internet 框架下，这样的情况很少。在那些环境中，网络条件能引起很大数量的缓冲丢失（没有缓冲入口项匹配报文的情况），从而也导致很大数量的报文进行进程交换。在一定的条件下也可能会导致缓冲颠簸，因为在缓冲里没有所有需要入口项的空间而引起旧的入口项连续被新入口项覆盖。

缓冲是用特殊数据结构实现的，这个数据结构允许任何记录都能用最短的时间搜索到。对于主路由表来说，搜索时间会随着表的大小成比例地增长。而对于缓冲数据结构来说，因为它们的组织方式，其搜索时间可以非常小而且可以完全连续地不考虑总的入口项数目。

IP 快速缓冲用 Hash 表数据结构实现。如图 4-1 所示。在 Hash 表中，每一个 IP 地址前缀都指向表中一个特定的地方。一个特定的 Hash 表入口项是按搜索到的 32 位 IP 地址的前 16 位与后 16 位进行 XOR 操作而找到的。XOR 计算的结果指向期望的 Hash 表的一个位置，称为一个 Hash 桶（Hash bucket）。每一个 Hash 桶包含一个缓冲入口项，其中包括一个为下一跳预先构建的 MAC 头部。每一次 Hash 计算并不总是产生一个惟一的 Hash 桶地址。这种多于一个 IP 地址指向相同 Hash 位置的情况称为冲突。当发生冲突时，IOS 把冲突的缓冲入口项链接到一起，放入一个最多能存放 6 个入口项的桶中。用这种方法，缓冲中不多于 6 个的入口项为了发现一个特定的匹配而需要被搜索。

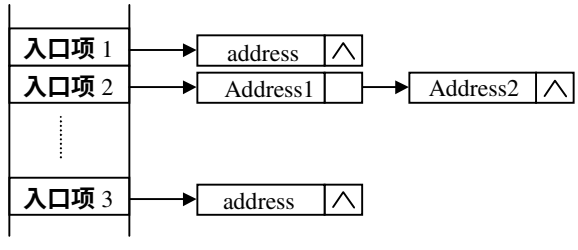


图 4-1　Hash 表

后来，在 IOS 的更高版本中，用一个称为 2 叉 Radix 树（2-way radix tree，2 叉树的一种形式）的数据结构来代替 Hash 表。如图 4-2 所示。在这种结构中，MAC 头部仍作为缓冲的一部分来保存。同 Hash 表一样，Radix 树也是用来提高获取记录数据时间的另一种特殊数据结构，信息被存储在一个基于二进制的、表示关键字（唯一标识每一数据集合的域）的一种树形结构中。

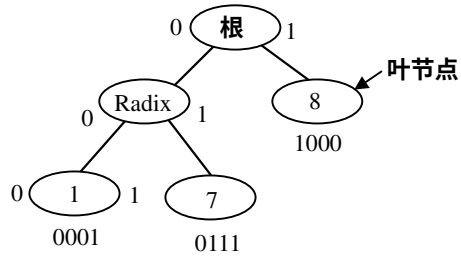


图 4-2　2 叉 Radix 树

树的每一层分枝都是基于数字内的二进制数字。例如，为了存储或查找，数 8，就要从二进制 1000 的第 1 个数字 1 开始。第 1 个数字 1 导致选择右边的分枝，从而到达树的一个节点。发现这个节点没有任何后继分枝，所以就比较节点里的数字和要搜索的数字是否匹配。这个例子中，恰好匹配。

而要查找或存储数 7，就又从根重新开始。在 7 的二进制表示（0111）中第 1 个数字是 0，所以选择左边的分枝。发现所遇到的节点有后继分枝，于是又根据二进制表示的第二个数字 1 向下查找，这次选择的是右分枝。

快速缓冲存储 IP 地址前缀时，有一个不足：不可能重叠缓冲入口项。解决这个局限性的一个简单方法就是为每一个目的主机构建一个缓冲入口项。但这不太现实，有许多原因，包括要花费许多构建这么多入口项所需要的处理以及缓冲存储所有这些信息需要的存储空间。为了解决这个不足，IOS 根据下面的规则构建入口项：

- 如果目的地址可以直接到达，就用 32 位长的前缀缓冲它。
- 如果存在多条费用相同的路径到达这个目的地址，也用 32 位长的前缀缓冲它。
- 如果它是一个超网，就用超网的前缀缓冲。
- 如果它是一个没有子网的主网，就用主网的前缀缓冲。
- 如果它是一个没有子网的主网，就用这个主网中最长的前缀长缓冲。

另外，在快速交换中还要解决递归问题。解决递归问题是在建立缓冲入口项时进行，而不是进行报文交换时。所以路由缓冲包含到达被缓冲的目的地址的实际下一跳 MAC 头部和输出接口数据。这就使缓冲入口项与路由表入口项和 ARP 缓冲（或其他的 MAC 层表）入口项断开了联系。

因为递归问题是在构建缓冲入口项时解决的，所以在快速缓冲与路由表或 ARP 缓冲之间就没有了直接的联系。那么，缓冲又怎样与原始表中的数据保持同步呢？最好的办法就是当主表里任何相应数据发生改变时，就使缓冲入口项失效，或删除。

IOS 会定期地让快速缓冲时的入口项老化。为了确定缓冲没有越界，也为了周期性地让入口项与 ARP 表再同步，快速缓冲的一小部分会每分钟失效一次。当可用存储空间大于一定

的界限时，缓冲老化进程就每分钟随意地使缓冲入口项总数的 1/20 失效。如果存储空间少于一定的界限时，缓冲老化进程就开始更疯狂地老化入口项——每分钟让缓冲入口项总数的 1/5 失效。

不像进程交换，快速交换并不支持报文负载共享。这个限制可能导致一些两个目的地址之间存在多条路径的情况下链接的不平衡使用。原因就是在快速度交换时必须让发生的路由和转发实现分离。缺乏负载均衡决策的机制是许多网络设计者担心的一个方面。作为这种担心的回应，新的交换方法现在开始支持负载平衡，从而帮助克服这个问题。其中一种新方法就是 Cisco 快速转发（Cisco Express Forwarding，CEF），在后面讨论。

4.1.3　最优交换

与快速交换一样，最优交换也是在短短的一个中断期间完成整个报文交换任务。最优交换基本上是带有优化缓冲的快速交换。最优交换与快速交换的主要区别就是它们访问路由缓冲的方法不同。最优交换软件还巧妙地利用了特定的处理器体系结构，而快速交换代码则是普通的代码，没有为任何特定处理器进行优化。不像快速交换那样通用，只有 IP 协议能使用最优交换。

最优交换缓冲通过一个 256 叉多叉树（256-way multiway tree，mtree）访问快速缓冲，图 4-3 是一个 256 叉树的示例。路由作为一个结点集合存储，每一个节点有 256 个后继分枝，预构建的 MAC 头部存储在节点中。虽然这种方式能提供比快速缓冲的二叉 radix 树更快的搜索，但它也存在与快速缓冲一样的局限性。

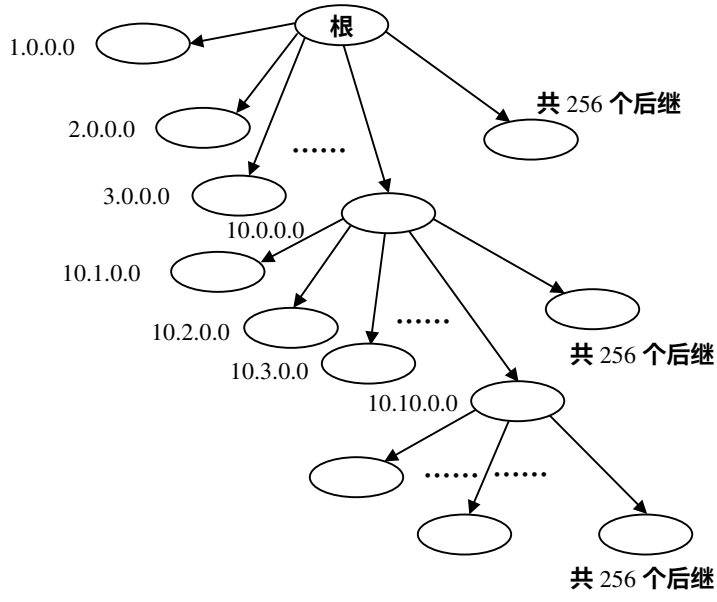


图 4-3　一个 256 叉树的示例

最优交换在许多方面仍然与快速交换类似，包括：缓冲入口项在第一个报文用进程交换到目的地址时构建、随着路由表和其他缓冲信息的改变，缓冲入口项也会老化和失效、仍然严格地根据目的地进行负载共享、仍用相同的规则构建到每一个特殊目的地址的缓冲入口项

等等。最优缓冲 Mtree 并不从所有节点开始，每一个节点的许多后继分枝都是空的。随着报文的进程交换缓冲就被构建。并被填充，然而可能所有的节点都被填充了，也从没有遇到一次最优缓冲Mtree。

4.1.4 快速转发 CEF

Cisco 快速转发是 IOS 最新的、速度最快的交换方法。像快速交换那样，CEF 交换也是在一个处理器中断期间用缓冲入口项进行整个交换操作。两者主要的区别是缓冲入口项构建的时刻不同。快速交换在发送到目的地址的第 1 个报文进行进程交换时会构建一个缓冲入口项，而 CEF 不是这样。CEF 表直接从路由表构建，而邻接表从 ARP 构建，这些表在任何报文交换以前已经构建好。

设计 CEF 是为了解决快速交换的最主要的不足：缺乏对重叠缓冲入口项的支持；因为在路由缓冲与其基础的表之间没有或只有很少的更正，路由表或 ARP 表的任何变化都会引起大部分路由缓冲失效；为了构建路由缓冲入口项，到任何给定目的地址的第一个报文都要被进程交换以及在一些情况下的低效负载平衡。

这些不足在普通的企业网络中不会引起问题，因为路由不经常改变且路由表不是特别大。然而在 Internet 中，它们确实是一个严重的问题。Internet 的路由器有非常巨大的路由表，到 2001 年底就有 130，000 条路由且在不断增长。同时，路由表也在不断变化，从而引起缓冲入口项不断失效。事实上，因为缓冲失效的太频繁而引起通过 Internet 的大部分通信量被进程交换。CEF 就是针对这些环境而被特别设计的，主要是为了提高路由的性能。

不像快速交换那样构建一个路由表和 MAC 地址表的缓冲子集合，CEF 构建它自己的一个镜像路由表和 MAC 地址表的结构。CEF 维持两种主要的结构，它们可以一起被看成 CEF 的“ 快速缓冲 ”：CEF 表和邻接表。邻接表包含用来直接连接下一跳的 MAC 层报文头部数据。邻接表与 ARP 表、帧中继映射表以及其他类型的表的数据一起存放。

CEF 表可以看成路由表的一个简化版本，它用一个 256 叉的 mrtee 数据结构来优化可用性。一个 mrtee 数据结构把实际的数据存储在树结构本身里面。例如：在最优交换 mtree 缓冲里，用来转化报文的 MAC 头部数据实际上存储在 mtree 内部。而在一个 mtree 数据结构中，树结构用来定位所需要的数据，而数据本身却存储在别的地方。在一个 256 叉 Mtree 结构中，每个节点最多能有 256 个后继分枝。在 CEF 表中，每一个孩子（或连接）表达一个 32 位 IP 地址中的一个８位组。

CEF 表直接从路由表构建，而邻接表从 ARP 构建，这些表在任何报文交换以前已经构建好。因为在任何报文交换以前 CEF 就预先构建好了缓冲，所以每一个刚接收的又要发送到一个可到达的目的地址的报文都能在接收中断期间把它转发。从而完全没有必要先进行一次进程交换才能让一个缓冲入口项建好，这种预建好的缓冲极大地提高了拥有大路由表的路由器的路由性能。使用一般的快速交换时，在路由缓冲入口项构建以前，IOS 会被进程级的通信量淹没。而有了 CEF 后，这种大量的进程级负载都消除了，而且也防止了 IOS 在网络路由较多时被进程级交换瓶颈阻塞。

把可达性/接口数据与 MAC 层头部数据分开放入两个数据结构还有其他的优点：因为报文交换所使用的表直接与它们的信息源相联，缓冲就不再需要一个老化进程。CEF 数据结构里的入口项永远不会老化，路由表和 ARP 缓冲的任何变化都会很容易地在 CEF 数据结构中

得到反映。同时，有了 CEF，当路由表和 ARP 缓冲发生变化时，再也不需要让很大数量的入口项失效。

CEF 交换的通信量负载共享可以是基于每一个源／目的地址对（缺省的设置）的形式，也可以是基于每个报文的形式。基于源地址／目的地址对的通信量负载共享解决了快速交换中的一些问题。CEF 是怎样实现它的通信量共享负载的呢？CEF 表中的入口项不仅指向邻接表中的一个入口项，也可以指向一个负载共享(Load share)数据结构。作为一种 CEF 表搜索结构，当 CEF 交换方法发现一个负载共享入口项时——不是一个邻接表入口项，它就用源地址/目的地址来决定使用负载共享表中的哪个入口项。而负载共享入口项就指向一个包含转发一个报文所需的 MAC 头部信息和其他信息的一个邻接表入口项。

4.2　早期的报文交换体系结构

最简单的交换方式是共享总线交换：当一个数据报到达输入端口时，通过一个共享总线直接被送往输出端口而不用路由处理器的干预，一旦发现总线忙，数据报就会在输入端口排队。这种方式的不足在于因为总线是共享的，路由器的交换能力受限于共享总线的带宽。

后来又提出一种共享存储交换：使用一个共享的存储器完成报文的交换，它没有独立的路由处理器，路由器的主 CPU 本身完成路由转发。当输入端口接收到数据报时，它首先中断主 CPU，将接收到的数据报从输入端口缓冲拷贝到共享存储器，处理器经过路由表的查找为该数据报选择合适的输出端口，最后将该数据报拷贝给输出端口。Cisco 8500 系列路由器使用的就是这种共享存储的交换方式。这种方法的不足就是交换速度受限于访存速度。

本节首先以 Cisco 早期的一种共享总线交换设计方法为例，介绍共享总线交换的特点与原理。随后再介绍两种存储共享的交换设计方法。

4.2.1　共享总线交换

在上一节所讲的快速交换和进程交换中，接收接口硬件都是先把接收到的报文拷入 I/O 存储区，然后再中断主处理器开始交换报文。无论是中断期间还是在一个专门的交换进程中，主处理器都得进行所有的实际交换工作。接口硬件只是从介质接收或向介质发送报文。同时，报文交换也只是主处理器必须完成的许多任务中的一个，因此并不是所有的处理能力都用于报文交换。

共享总线控制器提供了另外一种处理方法。这种方法不是让主处理器上运行的 IOS 进行所有的交换，而是在根本不占用主处理器的情况下，由共享总线控制器完成大部分交换工作。只要共享总线卡上有接收接口和发送接口，共享总线控制器就能自动地完成快速交换。这种能力提供了一种新的交换持性，称为自治交换。

共享总线交换引擎主要基于一个微编码处理器，这种微处理器允许几个操作在一个周期内同时执行。不像用于通常目的的处理器，如 M68K，共享总线处理器有一个特别设计的支持数据交换操作的精简专用命令集。这个处理器封装在一个特殊接口卡上，该接口卡还包含为处理器命令所用的存储器（称为控制存储器）、快速报文存储器以及多路总线和共享总线之间的接口。这个卡就是交换引擎，也被称为共享总线控制卡。这两个新的特性，

共享总线和共享总线控制卡，导致共享总线路由器的产生。图 4-4 给出了一种总线体系结构示意图。

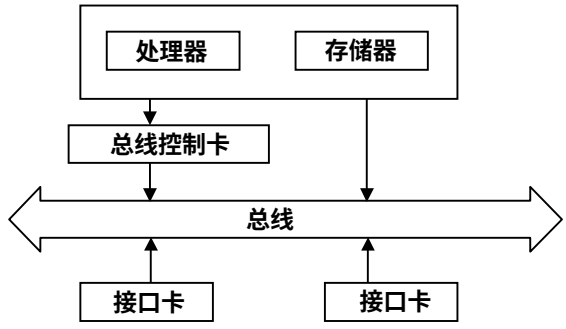


图 4-4　共享总线交换示意图

其中，处理器（CPU）负责运行 IOS 并执行进程交换以及在多路总线接口之间进行快速交换。处理器也快速交换那些从共享总线接口接收但不能自主交换的报文。存储器负责存储 IOS 运行映像，系统运行所需的各种数据以及缓冲的报文。总线控制器在共享总线接口之间运行优化的非 IOS 微代码进行快速交换，这个处理器在共享总线控制卡上。而快速报文存储器是共享总线控制卡上位于共享总线交换处理器与共享总线接口卡之间的双端口高速存储器，主要用于存储从共享总线接口接收或向共享总线接口发送的报文。

自治交换同快速交换的工作机制基本上相同，只是在不同的处理器上进行。它由接收中断触发，并用一个本地缓冲查找转发信息。事实上，自治交换用的快速缓冲的 Hash 表结构与以前版本一样，IOS 用同样的方法在处理交换期间维护主快速缓冲和本地交换处理器缓冲。每次向主缓冲里添加一个入口项或删除一个入口项，缓冲也会添加或删除一个入口项。虽然自治交换和快速交换差不多，但自治交换并没有快速交换支持的协议多，自治交换只支持 IP、IPX 和桥接协议，另外它处理缓冲遗失的策略也与快速交换不相同。

在快速交换中，如果一个报文被接收，但没有快速缓冲入口项让它使用了，该报文就会进入进程交换队列。而在自治交换中，如果没有缓冲入口项供报文使用，该报文就会被送往主处理器进行可能的快速交换。如果报文是发送到另外一个共享总线接口，则当该报文仍然在共享总线控制器里时，主处理器就能对它进行快速交换。然而，如果报文要发送到一个多路总线接口，整个报文就必须从快的多路总线拷贝到一个系统缓冲区中才能进行进程交换。虽然，共享总线控制器也有一个多路总线接口，但不能把该接口直接用于自治交换。

4.2.2　共享存储交换

共享存储交换使用一个共享的存储器完成报文的交换，它没有独立的路由处理器，路由器的主 CPU 本身完成路由转发。

图 4-5 给出了一种典型的存储共享体系结构。可以看到整个体系结构是非常简单的，只包括三个主要的部件：处理器（CPU）、存储器（DRAM）和接口控制器，其中处理器和接口控制器通过数据总线连接到存储器上。

存储共享路由器的处理器是它的隐藏在背后的核心。它运行 IOS 并进行报文交换，负责

运行那些平台支持的所有交换方法。处理器的类型取决于其运行的平台。例如 Cisco 1600 和 2500 系列路由器就采用 Motorola 68000 系列的 CPU，而 4500 和 4700 系列路由器则采用 MIPS RISC CPU。

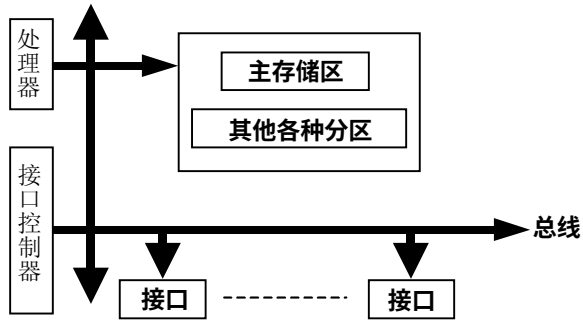


图 4-5　典型的存储共享体系结构

存储共享平台使用动态随机访问存储器（Dynamic Random Access Memory，DRAM）来保存路由器的大部分数据。DRAM 包含所有的路由表、高速缓冲、IOS 数据、报文缓冲区，在很多系统中 DRAM 还包含 IOS 本身的代码。

IOS 把可用的 DRAM 分成两个逻辑存储类型：本地存储区和 I/O 存储区。大多数情况下，本地存储区放在主区里。在存储共享平台上，主区用来存储路由表、缓冲、一般的 IOS 数据结构以及 IOS 实时运行代码，但不存储报文缓冲区。报文缓冲区存储在 I/O 存储区，这也经常称为存储共享（Shared memory），因为在存储器与介质接口之间共享了该存储区。

在许多系统中，为本地存储段所用的 DRAM 与为 I/O 存储段所用的 DRAM 在物理上是分开的，通常用两个不同的存储体。这些系统中，IOS 只是把一个存储体中所有可用的存储空间分配给相应的存储区：一个体给 I/O 存储区，另一个体给主存储区。然而，在 Cisco 1600 和 Cisco 2500 中，I/O 存储段空间和本地存储段空间都是从一个 DRAM 体分配。

IOS 创建两个存储池来管理 DRAM 区内存储空间的分配与回收。这两个存储池分别称为处理器池和 I/O 池。处理器存储池从本地存储段的子区存储空间中创建，I/O 存储池从 I/O 存储段的 I/O 存储区创建。I/O 池的大小与 I/O 存储区的大小关系非常密切。因为主区的其他部分必须为 IOS 数据、BSS 段以及在其他许多平台上 IOS 实时运行映像本身保留，处理器存储池只是主区的一个子区。

接口控制器负责向物理介质接口传送或从其上接收报文。它们都有自己的处理器，称为介质控制器，但这些介质控制器并不进行任何交换也不进行任何 IOS 处理操作。接口控制器主要进行介质控制操作以及在网络介质接口和 I/O 存储区之间移动报文。根据平台的不同，接口控制器可以是可以插拔的模块也可以是嵌到系统主板上的一个固定部件。

1．存储共享路由器的报文缓冲机制

IOS 有一个称为系统缓冲区的报文缓冲区集合，它主要是用于进程交换报文。在存储共享路由器上 IOS 也使用系统缓冲区，但是用于不同的目的。IOS 把系统缓冲区用于所有的报文交换方式，而不单单是进程交换。除标准的公用缓冲池以外，存储共享平台上的 IOS 还创建了一些私有系统缓冲池和为接口控制器所用的特殊缓冲结构，这种结构称为环。

私有缓冲池同公用缓冲池一样用于报文交换，但它会尽力阻止接口缓冲匮乏。在没有私

有缓冲池的情况下，所有接口必须通过竞争才能从公用缓冲池得到缓冲区，这就增加了缓冲区竞争的可能性（这会降低性能），也增加了一个接口占有一个池中所有的其他接口所需的缓冲的可能性。而有了私有缓冲池，每一个接口都被分配给一定数量的缓冲区供它专用，从而使缓冲区竞争达到最小的程度。尽管大多数接口都有自己专用的缓冲池，但一些低速接口没有，比如异步接口。

私有缓冲池是静态的，而且在 IOS 系统初始化期间就分配给了一定数量的缓冲区，这些私有池也不能根据需要创建新的缓冲区。如果确实需要一个缓冲区而私有池中又没有可用的了，IOS 就转回公用缓冲池为该接口分配一个与其 MTU 大小匹配的缓冲区。

除了公用缓冲池和私有缓冲池以外，IOS 还在 I/O 存储区内创建了一些称为环的特殊缓冲区控制结构。IOS 和接口控制器用这些环控制那些用于从介质接收和向介质发送报文的缓冲区。环实际上是一种普通的控制结构，它被许多类型的介质控制器用来管理存储正被接收或等待被发送的扼文的存储空间。环本身包含一些指向 I/O 存储区其它地方个别的一些报文缓冲区的特定介质控制单元。IOS 创建这些环作为介质控制器的代表，并与介质控制器一起管理。

每一个接口都有一对环：用于接收报文的接收环和用于发送报文的发送环。这些环的大小都是固定的，其大小受几个因素的影响。接收环的大小依赖于特定接口，并由介质控制器规格决定，而发送环的大小独立于介质控制器规格和接口上配置的队列类型。

分配给接收环的报文缓冲区数目是固定的，等于环的大小。接收环的报文缓冲区一开始从接口的私有缓冲池分配，而在操作期间它们可能被来自公用缓冲池或私有缓冲池的缓池区替换掉。分配给发送环的缓冲区数目可以从 0 到发送环可以拥有的最大值。发送环的报文缓冲区可以来自交换报文源接口的接收环，如果报文是 IOS 发出的，发送环报文缓冲区也可以来自公用池。在它们缓冲的数据被发送后，这些缓冲区都被发送环释放回它们的原来所属的缓冲池。

2．存储共享路由器的报文交换机制

存储共享路由器的 IOS 支持四种交换方法：进程交换、CEF 交换、快速交换以及 IOS 交换。报文的交换共分为三个阶段：

（1）接收报文阶段

（2）交换报文阶段

（3）发送报文阶段

在接收报文阶段，接口介质控制器检测到网络介质上有一个报文并把它拷入一个由接收环第一空闲单元所指的缓冲（也就是说，介质控制器所属的下一个缓冲区）。随后介质控制器把报文缓冲区又归还给处理器并向处理器请求一次中断，它并没有必要等待 CPU 的响应，而是继续把输入的报文接收到接收环的其他缓冲区中。这时介质控制器能继续把输入报文接收到接收环中，因此有可能在处理器处理完环中所有新的缓冲区以前，介质控制器把接收环填满，这种情况称为超限运行。当发生超限运行时，所有输入的报文都会被抛弃直至处理器能赶上需求。然后 CPU 响应接收中断，它试图从输入环中移走新填充的缓冲区，随后再从接口的私有池把输入环填满。

在交换报文阶段，接收环被重新填满以后，CPU 开始真正交换报文。这个操作包括确定该报文的下一跳信息和重写进报文的 MAC 头部信息。IOS 试图用配置在该接口的快速交换

方法交换报文。在存储共享系统中，它首先试着使用 CEF 交换（如果配置的话），然后才是快速交换，最后如果实在没有其他方法，才采用进程交换。接收中断期间，IOS 就试图使用 CEF 表（首选方法）或快速交换缓冲（其次）来进行一次交换决策。如果 CEF 交换和快速交换都失败了，IOS 就会采用进程交换。报文被放入适当进程的输入队列（例如：一个 IP 报文会被放入 IP 输入进程的队列中），随后接收中断返回。最后报文交换进程开始运行，交换该报文并根据需要重写 MAC 头部。注意这时报文还没有从它最初的缓冲区中删除。报文被交换后，IOS 继续走到进程交换的发送阶段。

在发送报文阶段，如果报文被 CEF 交换或快速交换，那么在接收中断上下文期间，IOS 就会检查在输出接口的输出队列里是否还有报文。如果已经有报文在接口的输出队列中，IOS 就把本次接收的报文仍放入输出保持队列中，而不是直接放入发送环，这样做是为了减少报文乱序的可能性。如果输出保持队列是空的，IOS 就把报文放入输出接口的发送环中，这通过把报文缓冲区连接到发送环描述器来完成。随后接收中断返回，继续处理。如果发送环已满，报文被放进输出保持队列中，接收中断返回。如果输出保持队列已满，报文被抛弃，输出抛弃计数器加１，接收中断返回。

发送阶段，IOS 在把任何 CEF 或快速交换的报文放在发送环以前，IOS 首先检查输出保持队列是否已经为空，这样会减少发送报文乱序的可能性。

如果报文被进程交换，它就被放在输出接口的输出队列里。假如输出队列已满，报文就会被抛弃同时输出抛弃计数器加１。随后 IOS 试图在发送环的输出接口寻找一个空闲描述器。如果确实有一个空闲描述器，IOS 就从输出保持队列移走报文并把缓冲区连接到发送环。如果没有空闲描述器（也就是说，发送环已满），IOS 就一直把报文留在输出保持队列，直至介质控制器从环中发送一个报文并使一个描述器空闲。输出接口介质控制器周期性地轮询它的发送环是否有报文需要发送。一旦介质控制器检测到一个报文，就把它拷到网络介质上并向处理器引发一个发送中断。最后，IOS 响应发送中断，断开报文缓冲区到发送到环的连接，并把缓冲区归回给它原来所属的缓冲池。

4.2.3　基于存储片的交换

前面讨论的报文缓冲机制都有一个共同特点，就是先形成由各种大小的缓冲区组成的缓冲池，然后为每个报文分配一个缓冲区。这些缓冲区经常被称为连续的缓冲区，因为整个报文的内容都放在一段连续存储空间的一个缓冲区内。缓冲区的大小或者是为了适应接口上允许的最大的报文长度，或者是根据一个固定的计划进行划分。

虽然连续的缓冲区易于管理，但它们并不提供存储器的高效利用。使用过大的缓冲区或者缓冲区利用的不平衡都会使连续缓冲机制浪费存储空间。在 IOS 版本 11.1 中，Cisco 引进了一个新的缓冲机制，称为存储片缓冲（particle buffering），它主要是用来避免连续缓冲区的低效使用。不是所有平台都支持存储片缓冲，它主要是应用于 Cisco 2600 系列、Cisco 3600 系列、Cisco 7500 VIP、Cisco 7200 系列以及 7000 的派生产品，例如：Cisco 7100 和 Cisco 6400 NRP。本节主要介绍存储片缓冲是如何用于报文交换的。存储片缓冲提供了一种高效缓冲各种大小报文的方法，这种方法使存储空间浪费达到最小。自从在 Cisco IOS 11.1 中引入存储片缓冲后，它就成了 Cisco IOS 实际的缓冲机制，并且现在许多基于 IOS 的 Cisco 平台也正在被设计成能够使用它。

下面介绍 Cisco IOS 对存储共享结构缓冲区管理的一种改进机制。

存储片缓冲用一种分散—聚集（Scatter-Gather）的机制进行报文缓冲存储区的管理。存储片缓冲机制不为缓冲区分配一块连续的存储区，而是分配一些不连续的（分散的）存储区，称为存储片。随后再把它们链接（聚集）在一起，组成一个逻辑报文缓冲区，称为一个存储片缓冲区。有了这种机制，一个缓冲区就能够跨越在多个物理缓冲区之间。

当在接口上收到报文时，系统就构建一个存储片缓冲。也就是说，把各个存储片链接到一起。这种方法同那些使用前不用聚集的连续缓冲区不同，也不像那些连续缓冲区。这种方法并不存在预先创建的存储片缓冲区组成的缓冲池。但是，IOS 怎么知道为每个缓冲区加入多少存储片泥？IOS 维持一些由存储片组成的存储片池，当收到报文时，它就从池中取出一些存储片来创建报文缓冲区。那些池中的存储片是基本的构造块，而存储片池就提供构造报文缓冲区所需的“原材料”。

在一个给定的池内，所有的存储片的大小都相同。所以池中的任何空闲片都可以用来构建缓冲区，而不用再考虑空闲片的大小。这种统一性简化了存储片的管理，并有助于高效地利用存储器。根据平台的不同，存储片的大小主要是有一种尺寸，也有一些例外——有的平台也可以拥有一个二级存储片组成的池，但通常是只有一种主要的存储片大小。当使存储空间的浪费最小化时，也应考虑让存储片的长度足够大从而能存放平均长度的报文，典型的是 512 字节。这样，许多报文的缓冲区就可以只由一个存储片构成。

存储片实际并不是仅仅由包含报文数据的存储空间组成，还需要一些额外的开销，以附加单元的形式保存关于报文和把分散的存储片链接在一起的信息。每一个存储片缓冲区都包含一个链接一个或多于一个存储片的报文头部，而一个存储片由存储片头部及相应的存储片数据块组成。表 4-1 详细说明了每一个组成单元：

表 4-1　　存储片的划分

单元名称	含　义
报文头	与连续缓冲区里的报文头部相同，它包含缓冲区中报文类型及大小的信息，还包含一些指向报文中特殊域的指针。报文头部也包含到第一个存储区的链接。
存储片头	是一个控制块，它包含一个到存储片数据块的链接和一个到缓冲区下一个存储片（如果有的话）的链接。还包含一些其他的信息，比如：数据块中数据的字节数以及本储存片所属池的信息。
存储片数据块	包含实际数据的一片存储空间，它仅仅是一个缓冲区。

IOS 采用存储共享平台上连续报文缓冲区的池管理策略来管理存储片池。为每一个接口创建一个私有静态存储片池，然后再创建一个所有接口和进程共享的动态公共池，称为标准池。在私有池用完它们自己的存储片的时候，私有池就会把公共池当作一个求助的对象。

在大多数系统中，还创建另外一个公共存储片池，称为快速交换池。它包含一些 128 字节长的小存储片。这些小碎片用在快速交换期间预先向报文写数据的时候——例如，在报文中，重写一个更大的介质头部。为了写数据，IOS 必须在一个已经被链接在一起的存储片缓冲区头部再插入一个新的存储片。因为要预写的数据通常很小，所以就使用这个特殊池中的存储片。

存储片缓冲区提高了存储器的利用率，但也增加了操作它的程序的复杂性。例如，使用存储片报文的域就可能会在各存储片池边界被任意分割。所以报文交换就必须适应这种可能性。

对大部分协议来说，所有的快速交换方法都可以处理存储片缓冲区。然而，进程交换则不行。因为进程交换增加了交换的复杂性，故 Cisco 决定不让进程交换处理存储片缓冲区。进程交换仍然要把报文放在连续的缓冲区里。因此，用存储片缓冲的报文排队等待进程交换时，必须通过一个称为合并的进程把存储片缓冲区转换为一个连续的缓冲区。

合并基本上是把一个报文缓冲区中所有存储片的内容一个个拷贝到一个适当大小的连续缓冲区里。根据平台的不同，这个任务可以由主 CPU 完成，也可以由一个单独的 DMA 引擎完成。当这些过程结束时，所产生的新报文看起来跟原来的一样（同样的报文头），只是它驻留在存储器中一片物理上连续的存储块里。

4.3　高性能交换

目前解决报文高速转发的交换结构主要有共享内存交换和互联网络交换。共享内存的实现简单，也可以达到较高的吞吐率（如 Juniper 公司的 M40 就使用了共享内存的方法），但可扩展性较差。而互联网络交换把输入端口和输出端口用多条可用的交换路径连接起来，消除了共享总线带宽所带来的限制。一种典型的互联网络交换是 Crossbar，也称为交叉开关，如图 4-6 所示。Crossbar 通过交换网络把 N 个输入接口和 N 个输出接口连接起来，Crossbar 可以被看作一个由 N*N 个交叉点连接成的 2N 条总线，当一个交叉点处于“可用”状态时，它所连接的两条总线之间就可以进行数据传输。Cisco 12000 系列路由器就是用这种交换结构达到 60Gbit/s 的交换能力。

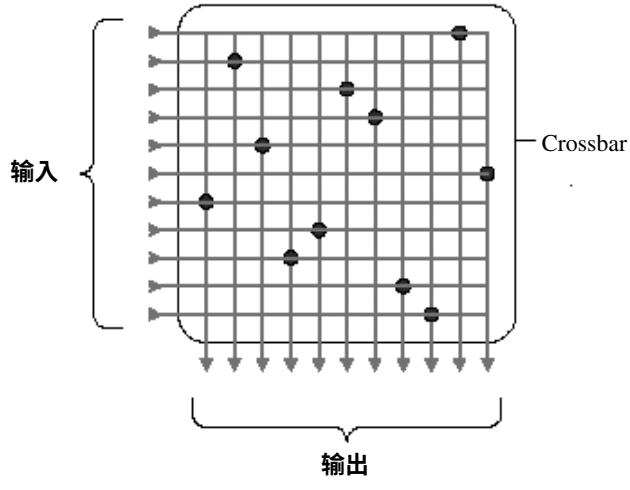


图 4-6　Crossbar

在交叉开关体系结构路由器中，数据直接从输入端经过交叉开关流向输出端。它采用交

叉开关结构替代共享总线，这样就允许多个数据包同时通过不同的线路进行传送，从而极大地提高了系统的吞吐量，使得系统性能得到了显著提高。系统的最终交换带宽仅取决于中央交叉阵列和各模块的能力，而不是取决于互连线自身。就目前来看，这种方案是高速核心路由器的最佳方案。同时基于交叉开关设计则有更好的可扩展性能。

对第三层 IP 报文在千兆位以太网或 2.5G POS 上实现线速率路由转发，利用软件满足该要求是不现实的。第三层路由利用交换技术全部使用硬件完成路由转发。交换式路由器将控制与转发相对分离，使软件得以专注于控制路由。该类路由器一般只支持 TCP/IP 协议。其中的控制部分只运行路由协议和管理，不强调实时性要求。目前对于交换式路由器，如果需要获得丰富的附加功能，将丧失部分硬件转发的性能。按照整个路由器中是否存在核心 CPU 又把交叉开关路由器分为两类：主从分布控制交换式路由器体系结构和全分布控制交换式路由器体系结构。主从分布控制交换式路由器由于存在全局核心 CPU，因此该体系结构含有单一系统访问点，从而简化了系统设计。同时该体系结构便于系统的适度扩展。在软件方面，部分底层的软件功能也能够达到适度并行。Cisco12000 就是采用该种设计体系结构。这种体系结构在功能上最多可以达到 16 端口 2.5G 的 POS 的线速率路由转发。

在第三代路由器中，Crossbar 是关键的部件，但是 Crossbar 的交换阵列扩展性很低，要想实现大规模的第三代路由器是非常困难的。同时随着光通信技术的迅猛发展，10Gbit/s 的光信号设备已经出现，因此要对第三代路由器的体系结构进行重新设计。人们在第三代全分布控制交换式路由器体系结构基础上提出了第四代路由器的概念。它是适应 10Gbit/s 的通信速度的核心路由器。

4.3.1 Crossbar 交换

前面已经提及，最早的路由器结构是基于计算机的结构。它由一条共享总线、中心 CPU、存储器和线卡组成。输入／输出端口功能由线卡完成，线卡提供 MAC 层功能并与外部链路相连。每个输入分组均通过共享总线送往 CPU，完成转发决定，然后再次通过共享总线送往输出端口。其性能受两大因素限制：

- 中心 CPU 的处理能力，因为搜索路由表非常耗时。
- 共享总线的性能，因为每个分组均需两次通过共享总线。

为解决第一个性能瓶颈问题，一些路由器厂商使用多 CPU，让多 CPU 并发，每个 CPU 仅处理部分输入流量，但是每个分组依然需要通过共享总线两次。

不久路由器结构发生了一次重要变化，出现了带有智能线卡的路由器结构。每个接口提供路由 Cache 和处理能力，转发决定在接口中完成，每个分组仅需在从输入端口传送至输出端口时通过共享总线一次。路由缓冲是中心 CPU 路由表的高速缓存。但是，由于 CPU 性能不可能跟上物理链路容量的增长速度，也就不可能为来自每条输入链路的每秒几百万个分组完成转发决定，所以在每个接口增加一个具有特别用途的 ASIC（专用集成电路）取代 CPU 进行转发决定、管理队列和仲裁对共享总线的访问。

但是共享总线一次只允许一个分组从输入端口传送至输出端口，共享总线的性能瓶颈依然存在，为消除第二个性能瓶颈，将共享总线替换为交叉开关，这样多线卡之间可以同时通信。带有交换式背板的路由器结构是新一代路由器结构的典型代表。

在带有交换式背板的路由器中，智能线卡对于输入分组主要完成接收、IP 路由查找和分

组分类操作，确定其输出端口、NH 地址和对分组欲进行的操作，对于输出分组主要完成 QoS 调度和发送；交叉开关的主要功能是将分组从一个端口线速交换到另一个端口，并可同时对多个端口并发操作；路由 CPU 主要完成系统的控制，包括路由表的更新、MIB 库的管理和错误处理等。

Crossbar 算法有很多实现方法，比如支持组播的 EISP 算法、支持优先级调度的 OSP 算法和支持带宽预约的 RISP 算法。Cisco 12000 的调度算法是基于斯坦福大学的一个设计。可以在斯坦福大学 Nick Mckeown 的论文" Fast Switched Backplane for a Gigabit Switched Router "中找到交换结构的详细描述。物理上，Cisco 12000 路由器的交换结构可由 5 个交换结构卡组成。系统中存在着两种交换结构卡：时钟调度卡和交换结构卡，时钟调度卡是交换结构卡的超集。

4.3.2　Cisco 12000 系列路由器交换结构

在 Internet 流量快速增长的同时，对可靠性也有了更高的要求，因为人们开始应用互联网及电子邮件进行商业活动。为了适应流量及可靠性的需求，Cisco 开发了吉比特交换路由器（GSR）--Cisco 12000 系列路由器。其结构示意如图 4-7 所示。GSR 有以下四种配置：

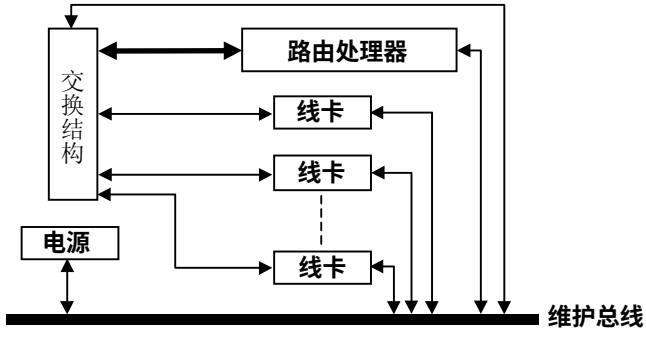


图 4-7　Cisco 12000 的结构示意图

- 12008——8 槽背板，40-Gbit/s 交换结构。
- 12012——12 槽背板，60-Gbit/s 交换结构。
- 12016——16 槽背板，80-Gbit/s 交换结构，可升级到 320-Gbps。
- 12416——16 插槽机柜，总交换容量高达 320Gbit/s，每槽容量为 20Gbit/s(10Gbit/s 全双工)。

其体系结构的重要组成部分有：交换结构、维护总线（Maintenance Bus，MBUS）、路由处理器和线卡（Line Cards）。

1．交换结构

Cisco 12000 采用交叉开关结构为背板。交叉开关结构本质上是 2N 条总线（N 是连到交换结构的 LC 数量），用 N＊N 个交叉点连接在一起。

如果一个交叉点运行，则通过该交叉点连接的两个 LC 被连通并可以相互通信。经过交叉开关结构的发送单元是固定大小的报文，被称为 Cisco 单元，与可变大小报文比较，Cisco

单元更易于调度。在放入结构之前，报文被分割成单元，并在发送之前由输出端 LC 重组。Cisco 单元为 64 字节长，其中 8 字节头部，48 字节负载，8 字节 CRC。

尽管交叉开关结构输出允许多个 LC 同时发送、接收数据，仍可能（实际上很可能）两个卡同时想发送数据到同一个输出端卡上。为解决这类问题，设计者必须选择在给定结构周期中哪个 LC 发送及哪个 LC 接收。物理上，这种路由器的交换结构可由 5 个交换结构卡组成。如图 4-8 所示。

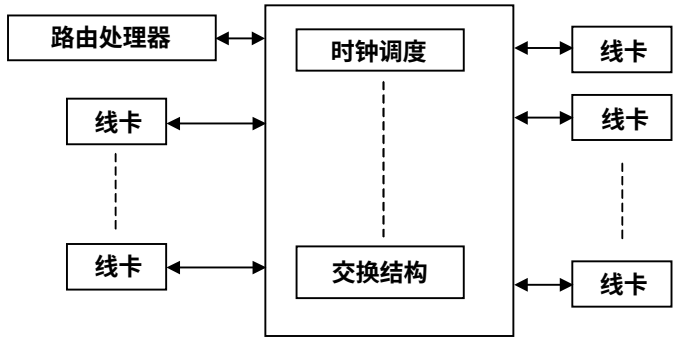


图 4-8　Cisco 12000 交换结构

GSR 交换结构可以在两种模式下运行：全带宽和 1／4 带宽。

为指出配置的交换结构所能提供的带宽，必须记住每交换卡每个槽能提供 1.25-Gbit/s 的全双工带宽。在全带宽模式下，有 5 个结构卡——两个 CSC 及三个 SFC 可用。而第二个 CSC 处于备用模式，提供时钟及调度冗余，因此，只有 4 个交换卡被实际使用以通过结构发送数据。每个交换卡应该用 4 去乘 1.25-Gbit/s，然后用该结果乘以安装的 LC 数目（本平台上的最大数目）。在 1/4 带宽结构设置下，始于 LC 的信息流被分段成相同大小的单元，然后经由唯一的连接发往活动的 CSC。

12016 可以升级到增强型交换结构，该结构是基于每交换结构卡每槽可提供 5-Gbit/s 全双工带宽的 64X64 纵模结构。在全带宽模式下，一个增强型交换结构每槽提供 20-Gbit/s(5-Gbit/s*4)带宽，它要求支持 OC-192 线卡。增强型交换结构不在本书中讨论。

2．维护总线

维护总线（MBUS）是连接路由处理器、LC 开关结构卡、电源及风扇的 1Mbit/s 串行总线，用于系统维护。当路由器启动后，路由处理器通过 MBUS 来发现系统中其它卡，启动 LC，获得版本信息、环境信息及一般的维护信息。数据流不经 MBUS 发送，它们是通过交换传输，MBUS 是专门用于控制各组成部分的。

3．吉比特路由处理器 GRP

吉比特路由处理器，是系统的中枢。GRP 运行路由协议，计算转发表，创建 CEF 表与邻接表，并将它们经由交换结构分布到系统中所有 LC 中。

GRP 的 CPU 使用与 Cisco 7500 的 RSP4 相同的 R5000 处理器。CPU 主要负责执行路由协议及保存 CEF 表的主拷贝，该拷贝用于下载到执行交换的 LC 上。主存储器（DRAM）最大可达 256MB，用于存储 IOS 代码及全部数据结构。CSAR SRAM（512KB）存储器用于将从交换结构到来的单元重组为报文。以太网控制器是为超带宽管理设计的。通信量不应在该

接口与 LC 接口间交换。

4．线卡

Cisco GSR 中大多数报文的交换是在线卡 LC 上进行的。线卡由三部分组成：

■　物理层接口模块（PLIM）：是特殊介质硬件模块（ATM，POS，快速以太网），它终止了与物理层的联系。

■　交换结构接口：交换结构接口准备报文，这些报文经过交换结构发送至目的 LC。

■　报文转发引擎：负责交换报文。有三种报文转发引擎，分别称为：引擎 0，引擎 1，引擎 2。线卡按报文转发引擎分类。引擎 0 是第一个报文转发引擎，用在早期的 GSR LC 上，引擎 1 的 LC 增进了报文交换性能。引擎 2 的 LC 更进一步增进报文转发性能并在硬件上支持 QoS 特性。

第 5 章　高端路由器系统软件设计

说起操作系统，一般人们最熟悉的是那些最流行的、使用最广泛的计算机操作系统，比如：UNIX、Linux、MS-DOS、Microsoft Windows 98/2000 等等，这些都是很知名的操作系统。然而如果说起网络操作系统，可能会有许多人不知道。这是很自然的，因为很多人根本就没有直接接触过网络操作系统，只知道在路由器或其他网络设备中运行着软件，而说不清楚这种软件到底是属于什么类别。

即使那些知道网络操作系统，甚至直接使用网络操作系统的人，也常常认为网络操作系统只不过是在路由器上运行的一种软件而已。其实，对于一个网络来说，真正在维护网络各种可用特性的是各种各样的网络操作系统，比如：Cisco 路由器的 IOS、Juniper 路由器的 JUNOS、Pluris 路由器的 Tera OS 等等。

确实，IOS、JUNOS、Tera OS 并不运行字处理软件或其他的应用程序，但实际上它仍是一种操作系统：一种特殊的、用来交换数据报文的操作系统。为全面理解传统路由器软件结构在今天的服务提供商网络中所面临的挑战，很有必要去了解这些软件系统的源头和发展。最初，路由器软件结构假设底层的硬件仅有一个 CPU，这块 CPU 不仅负责提供实时的包转发，同时也要提供路由计算，建立路由更新，管理用户界面和支持网络管理。这种对单一 CPU 的依赖，决定了传统路由器操作系统被开发和实现的框架。

本章介绍高性能路由器中系统软件的一些知识，包括它的组成模块、实现方法等等。首先阐述系统软件体系结构的组成，随后概要介绍系统软件的几种实现方法，而对于具体的实现细节，鉴于篇幅和其他各方面的原因，在这里并不作深入介绍，有兴趣的读者可以参考相应的资料以及附录中给出的网络资源。

5.1　软件体系结构

路由器软件系统结构的很大一部分都致力于使报文交换更加迅速、更加高效。虽然组成软件系统的组件中有许多也是组成通常所说的操作系统的组件，但这些组件都包括为系统特别设计的一些关键的功能。首先从一些基本的操作系统概念和术语开始。这些概念和术语，对理解软件系统体系结构非常有用。

5.1.1　概述

作为一个计算机系统或者路由器系统来说，操作系统主要完成两个主要功能：硬件抽象和资源管理。硬件抽象给上层的软件——包括应用软件、协议软件、支撑软件等等提供一个

接口，这个接口在硬件与它们的应用程序之间。有了这样的接口，软件开发者就不用再关心硬件的复杂性，实际上对硬件的编程只需一次（在操作系统内部），然后大家都可以共享。对于计算机操作系统来说，硬件抽象主要是提供对诸如外设、存储器、时钟等其他硬件资源的抽象；而对于路由器来说，硬件抽象主要是提供对各种类型的线卡、系统的交换背板等其他硬件资源的抽象。

管理计算机资源主要包括管理 CPU 周期、存储器、磁盘空间等等。通过操作系统的有效管理，这些资源就可以被各种应用共享。像硬件抽象功能一样，在操作系统内部构造一个资源管理体系，从而每一个应用的开发者都不必再为自己程序编写资源管理代码。目前，所实现的路由器大部分都是单 CPU 的，因此这里的叙述主要是针对单 CPU 的系统。可能有些方法和策略在多 CPU 的系统中就不需要了，但这并不影响这些方法的基础性。

1．资源管理和多任务调度

单线程操作系统在同一时刻只允许一个程序运行，而现在更多的操作系统支持在同一时刻管理多个程序。在同一时刻运行多个程序称为多任务，而支持多任务的操作系统一般称之为多任务操作系统。为多任务操作系统编写的应用程序通常包括多个能在同一时刻运行的、独立的任务。这些小的子程序一般称之为线程，它们组成了一个程序内的单线的指令序列。每个线程都含有自己的一个 CPU 寄存器的集合，称为上下文。线程也能和同一程序内部的其他线程共享存储地址空间。

一组共享相同的地址空间、完成同一个任务且一起控制一组操作系统资源的线程称为一个进程。在支持虚存的操作系统和 CPU 中，每一个进程都在各自独立的地址空间和上下文中运行，它们的地址和寄存器内容并不交叉。因为一个处理器在同一时刻只能运行一个程序的指令，因此操作系统必须判断某一时刻应该执行哪一个程序的指令集合（即哪一个线程）。决定究竟该执行哪个进程的机制称为进程的调度。进程调度通常由操作系统的一部分核心程序来完成，这部分核心程序称为系统内核。

根据操作系统所采用的优化方式的不同，操作系统可以使用不同的线程调度策略，比如 FIFO 调度、优先级调度、抢先调度、时间片轮转式调度等等。不同的应用（批处理、交互、事务性、实时及其他类型的应用）拥有不同的 CPU 使用特性，同时这些应用的整体性能也受所采用的调度策略影响。

最简单的调度策略就是根据每一个线程请求处理的先后来分配处理器供之使用，先来的先用，直至执行完一个再执行下一个，这种方法也称为 FIFO（先进先出）直至运行完调度。FIFO 的优点是它比较容易实施，开销非常低，而且比较公平——所有的线程都是平等的，先来的先服务。这种调度方法比较适合于批处理应用以及一些需要顺序处理，而且处理完马上就会结束的事务性应用。但是，这种调度方式不适合交互性和实时性很强的应用。因为交互性应用需要很快地被调度，对 CPU 需求周期也比较短，只有这样它才能很快地把执行结果及时反馈给用户或服务其他外部设备。

对于交互性和实时性很强的应用，一个可行的解决方法就是为每个线程赋于不同的优先级。给那些急需访问 CPU 的线程，比如实时应用的线程，赋于较高的优先级；而给不太关键的线程，比如批处理应用的线程，赋于较低的优先级。具有高优先级的线程可以跳到任务队列的头部，迅速地访问 CPU。如果有几个相同优先级的线程同时到达，则按它们到达的顺序调度（就像基本的 FIFO 策略），这种调度机制就称为直至运行完带有优先级的调度。

虽然这种调度策略比基本的 FIFO 有所改进，但它有一个不足：这种方法很容易使某一个线程长期独占 CPU，从而不太适合于交叉的或实时的应用。具有高优先级的线程可能会一直等在一个已经开始运行、优先级比较低的线程后面，直至它运行完。要解决这个问题，一个可行的方法就是暂时挂起或抢先一个正在运行的线程，从而其他的线程可以使用 CPU。

路由器的系统软件一般都是使用这种抢先调度。抢先调度会根据不同的任务优先级分配 CPU 资源的使用。有意地挂起一个线程而调度另外的线程称为抢先。采用抢先而不是直至运行完的调度策略称为抢先调度。采用这种调度策略的操作系统称为抢先多任务操作系统。抢先主要是基于内核通过一个上下文转换（Context Switch）来周期性地改变现在运行的线程而实现的。上下文转换的触发可以是一个系统时钟（每一个线程都分到一个时钟长度），也可以是一个调用内核本身的函数。当一个上下文转换被触发时，内核就选择下一个线程访问 CPU，而被抢先的线程则被放于队列尾，并根据调度策略等待下一次被调度的机会。

当操作系统内核停止一个线程使用 CPU，而调度另一个线程访问 CPU 时，一个上下文转换就发生了。换句话说，当计算机改变它正执行的任务时，就发生了上下文转换。如果根据 CPU 占用时间来评价的话，上下文转换是非常费时的，因为所有正在运行的线程所使用的寄存器都得被保存，然后再重新为新的线程使用。上下文是被抢先的线程得知它在哪里被停止的基础，也是即将运行的线程得知它上次运行到哪里的基础。

2．存储器管理

操作系统还必须管理系统的存储器资源。一般情况下，操作系统会把存储器分为不同的区，分别用来存储实际的计算机指令（代码）、数据变量和堆。堆是存储器中的一段，进程可以在堆中动态地使用和释放存储空间。

一些操作系统还提供了一种让进程能够寻址比存储器实际存储空间还大的方法，这种方法对应的一个概念称为虚拟存储器，简称为虚存。有了虚存，系统就能把它的存储器扩展到二级存储，例如一个磁盘驱动器，而且这对进程是透明的。操作系统使用一个硬件特性来创建虚存（有的处理器上有），称为存储器映射单元（Memory Map Unit，MMU）。MMU 根据内容实际驻留的存储区把对存储器的寻址请求重映射到物理存储器（RAM）或二级存储器（磁盘）。MMU 也允许对一些地址范围进行保护（标志为只读）或置为全部不能被映射。

虚存还有另外一个好处，在支持虚存的操作系统内，MMU 可以被编程为每一个进程创建一个独立的地址空间。这样每一个进程就可以拥有完全属于自己的一个存储空间，同时还可以阻止其他的进程访问。然而，虽然虚存有许多好处，但它也不是无偿拥有的。虚存的应用会对资源有更大的需求，同时也会有性能上的损失（有些是非常严重的）。因此，正像即将看到的，IOS 并不使用一个完全的虚存机制。

3．中断

中断是一个硬件特性。它引起 CPU 暂时停止现在运行的指令序列而把控制权交给另外一个特殊程序。那个特殊的程序称为中断处理程序，它执行一定的操作来响应引发中断的事件，处理完后再返回被中断的指令。中断通常由外部硬件产生，比如一个介质控制器需要被注意，但也可能由 CPU 自身产生。操作系统通过提供一系列处理各种类型的中断的处理程序来支持中断。

5.1.2 路由器系统软件的组成

最初，路由器操作系统被设计成较小的、内嵌进早期路由器的一种操作系统。那时，路

由器本身被看成一种硬件设备——在硬件和软件之间没有很大的差异。实际上，最初的网络操作系统只是指运行于路由器的“一种操作系统”。

随着路由式的网络逐渐被接受，支持路由器的协议逐渐增多，同时对其他功能的需求也逐渐增加。网络界的工程师们通过向路由器软件中加入各种新的特性来响应这种需求，从而导致今天的多功能路由和桥接软件集于一身的网络操作系统的出现。有趣的是，虽然系统软件的功能增加了许多，但它基本的体系结构却没怎么改变。

以 Cisco 的 IOS 为例，简单介绍一下路由器系统软件的基本组成。目前 Cisco 公司的 12000 系列高性能路由器中仍然使用 IOS，只不过是功能多了一些，基本的体系结构并没有改变很多。同其他操作系统相比，IOS 有一个非常简单的结构。像其他较小的、内嵌的系统一样，IOS 被设计成比较简洁的、驻留在存储器且受限于原始平台的一种系统。

早期的路由器只有很有限的存储器让软件和数据（例如，路由表）共享，而且为了限制可执行映像的大小，IOS 只提供基本的服务。在设计过程中，速度是一个主要考虑的方面。为了最大限度地增加路由器快速交换报文的能力，开发者作了许多努力来设计操作系统，使之具有最小的操作性开销，同时允许 CPU 使用最大的带宽进行报文交换。例如线程间的存储保护机制。

IOS 并没有其他许多操作系统所具有的安全保护措施，因为它们会带来很大的 CPU 和存储器开销。总的来说 IOS 主要是为了追求速度，因此在错误保护方面有所牺牲。

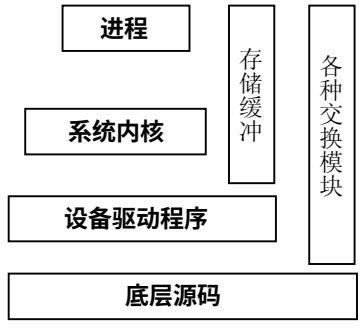


图 5-1　IOS 体系结构

IOS 主要有五个主要组成部分，这也是一般的高端路由器软件系统通用的一种结构，其体系结构如图 5-1 所示。各模块的功能列于表 5-1。

表 5-1　　高端路由器软件系统组成

模 块 名 称	功　　能
进程	执行一定任务的独立线程和相关的数据，如系统维护、交换报文、应用路由协议以及网络管理。
内核	为 IOS 的其余部分提供基本的系统服务，例如存储管理和进程调度。它为进程提供了硬件资源（CPU 和存储器）的管理。
报文缓冲	用来存放将要被交换的报文，主要是全局存储缓冲以及相应的管理函数。
设备驱动程序	控制网络接口硬件设备及其它外部设备（如：Flash Card，闪存卡）的函数。设备驱动程序接口位于 IOS 进程、IOS 内核和硬件之间。它们也同时与快速交换软件有接口。
交换软件	各种高度优化的报文交换函数，例如快速交换、最优交换、快速转发等等。

1．进程

路由器操作系统的进程与其它操作系统里的单线程基本相同。每一个进程有且只有一个线程组成，都有自己的栈空间、CPU 上下文，并且都能控制诸如存储器和控制台设备的资源。为了使开销最小，路由器操作系统并不采用进程间的存储保护，在上下交换期间不执行任何存储器管理。因此，虽然每个进程都有分配给自己的存储空间，但其他的进程也同样能访问该空间。

Cisco IOS 采用带优先级的运行直至结束策略调度可执行进程。开始，这种没有抢先的模型对一个要迅速处理输入报文的操作系统来说不是一个好的选择。在某种意义上说，这是对的，IOS 交换需要快速改善其处理模型的实时反映限制。这种模型也有许多优点，从而使它比较适合于支持在外部保持关键交换路径的进程，例如，协调的多任务通常会使线程间的上下文交换更少，同时也减少了 CPU 总的调度开销；对编程者更简单，因为编程者能控制一个进程将在哪里被挂起，这就很容易将上下文交换限制在共享数据会被修改的地方，从而减少了偏向某一进程和进程间死锁的可能性。

除了在软件系统被中断期间，系统可以在任何时候创建和终止一个进程，一个进程可以被内核（系统初始化时）或被其它运行的进程创建或终止。这里所说的中断是指硬件中断。当 CPU 被中断时，它暂时挂起现在运行的线程，并开始运行相应的中断处理程序，在 CPU 运行中断处理程序期间不能创建新的进程。系统中有一个专门的进程负责创建多个进程，它就是分析程序 Parser。分析程序是一个解释系统配置和可执行命令的函数集合，它在系统初始化期间被内核唤醒，也可以被为控制台和 Telnet 会话提供命令行接口（CLI，Command-line Interface）的进程唤醒。

任何时候用户键入一个命令或从一个文件读出一个配置命令行，分析程序都会解释该内容并采取相应的动作。一些配置命令可能只是一个值的集合，如 IP 地址，而有的命令可能比较复杂，如路由命令或事件监视命令，有些命令也会引起分析程序创建新的进程，如在路由器上启动一个协议。

在路由器系统中，进程的生存期要经历几个阶段。图 5-2 给出了它们的生存阶段及相应的状态。各阶段进程的状态列于表 5-2。

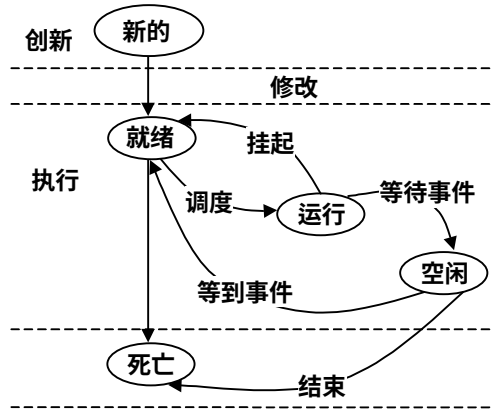


图 5-2　进程生命周期

表 5-2　　路由器进程的生存周期

阶 段 名 称	解　　释
创建阶段	一个新进程被创建时，它会收到自己的栈区并进入“新的”(new）状态。然后进入修改状态。如果没有必要修改，进程就进入执行状态。
修改阶段	跟其他的许多操作系统不一样，系统在进程被创建时，并不自动向进程传送启动参数或为之分配一个控制台，因为它假定大多数进程不需要这些资源。如果一个进程确实需要两个之中任何一项资源的话，创建它的线程可以修改该进程从而为之分配资源。
执行阶段	一个进程被成功地创建和修改后，它转向准备状态并进入到执行阶段。在这个阶段，一个进程能够访问 CPU 并且运行。这时，进程处于准备状态，或者运行状态，或者空闲状态。处于准备状态的进程是在等待访问 CPU 并准备开始执行指令；处于运行状态的进程是正在控制 CPU 并执行着指令；而一个空闲的进程则是处于睡眠状态，在它有资格运行以前等待一个外部事件发生。 当一个进程被调度运行时，它从准备状态过渡到运行状态。在没有抢先的多任务调度下，一个被调度的进程一直在 CPU 上运行直至被挂起或终止。一个进程有两种被挂起的方法：它可以自己向内核明确地表示它要放弃 CPU，转入准备状态，等待下一次轮到自己时再运行；也可以因为等待一个外部事件的发生而挂起。进程开始等待一个外部事件时，内核会隐含地把它转入空闲状态，从而被挂起，继续等待外部事件发生。事件发生后，内核再把它转入到准备状态，于是该进程等待轮到它时再运行。
终止阶段	终止阶段是进程生存周期的最后一个阶段。当一个进程完成它的功能并关闭（称为自终止）或者另外一个进程杀死它时，就进入终止阶段。当一个进程被杀死或自终止时就进入死亡状态。被终止掉的进程一直保持死亡状态，不被激活，直到内核宣布收回它所占的全部资源。内核也可能在进程终止时记录关于进程栈的统计信息。当该进程的资源被重分配时，它就转出死亡状态，彻底从系统中消失。

调度进程时，采取一个优先级机制。一般，路由器系统共有 4 个优先级别。分别是：

- 最高：为基本的、用来解决资源分配问题的系统进程保留。
- 高：分配给那些提供快速反应的进程，比如从网络接口收到报文的进程。
- 中：缺省分配给大多数的 IOS 进程。
- 低：分配给那些提供辅助性任务的进程，比如：登录信息进程。

进程创建时，根据进程的目的，它会被分配一个优先级别。这种特性是静态的，也就是说，一旦创建时进程被分配了一个优先级，该级别就永远不再改变。根据进程对系统重要程度的不同，优先级机制使进程具有不同的访问 CPU 的权力。然而，Cisco 的 IOS 并没有采用抢先机制，所以优先级级别比较高的进程也不能中断优先级级别比较低的进程。这种机制只不过是给予优先级级别较高的进程更多访问 CPU 的机会，这也将会在后面讲述内核调度操作时看到。在路由器系统中，有些进程是必须存在的。它们基本上都是用来执行内务处理或为其他进程提供服务。

2．系统内核

当在操作系统上下文中使用“内核”一词时，“内核”经常构成这样一幅操作系统核心

的“映像”：核心在一个特定的 CPU 保护模式下运行并管理系统资源。虽然路由器系统软件内核也帮助管理系统资源，但它的体系结构与其他操作系统的核心有所不同。路由器系统软件内核并不是一个单独的单元，相反它是一个作为对等体，而不是管理者，与系统的其它部分相连的组件及函数组成松散集合。并不存在特殊的内核模式。路由器系统软件的每一部分，包括内核，都是以客户模式在 CPU 上运行并拥有对系统资源的完全访问权。

内核负责调度进程、管理存储器、为自陷和中断处理程序提供服务例程、维护时钟以及处理软件异常。

（1）进程调度程序

调度程序用一系列代表每个进程状态的进程队列来管理系统中所有的进程。队列中存有该进程处于某个状态时的上下文信息。随着调度程序把该进程的上下文从一个进程队列转移到另一个进程队列，进程的状态也做相应变化。总的说来，一共有 6 种进程队列：

- 空闲队列：包含那些处于活动状态但等待一个事件发生后才能运行的进程。
- 死亡队列：包含那些已经终止但在其被系统完全删除前必须重新声明其所占资源的进程。
- 准备队列：包含有资格即将运行的进程。共有 4 种准备队列，每一个都有相应的进程优先级：关键、高、中和低优先级的准备队列。

当一个正在运行的进程被挂起，调度程序重新获得 CPU 的控制权，然后采用一种算法从任一个队列中选取下一个即将运行的进程。

调度算法的思想很简单。调度程序首先检查优先级级别为关键的准备队列里正在等待的进程，让这些进程都有一次运行的机会。所有的优先级级别为关键的进程都运行过一次后，调度程序检查优先级级别为高的队列。如果该队列里面没有进程准备好，调度程序就直接检查级别为中的队列。否则，调度程序就把高优先级的进程从队列中移出，并让它们运行。每运行完一个高优先级的进程，调度程序会检查优先级级别为关键的队列，如果其中有准备好的进程，则把它们运行完之后再运行高优先级的进程。当所有高优先级的进程都拥有运行的机会后，调度程序跳过中优先级和低优先级的进程队列再重新开始。

在确定没有高优先级的进程等待运行的情况下，调度程序检查中优先级的进程队列。如果其中没有进程准备好运行，就直接检查低优先级的队列。否则，调度程序就把中优先级进程移出并让之运行。每运行完一个中优先级的进程，调度程序会检查是否存在任何准备好的高优先级进程，如果有，则先让它们运行完（这个过程中又会检查是否有级别为关键的进程），然后才调度下一个中优先级的进程运行。当所有的中优先级的进程都有机会运行后，调度程序会跳过低优先级进程队列，又从最初重新开始。调度程序在连续跳过 15 次低优先级进程队列后才会继续往下走，这是一种安全机制：防止低优先级的进程被饿死。

确信没有任何准备好的中优先级或高优先级的进程等待运行（或者低优先级进程队列已经被忽略过 15 次）之后，调度程序开始检查低优先级队列。如果其中有准备好的进程，就把它们移出并运行。在每两个低优先级进程运行中间，又会检查是否存在任何中优先级的进程准备好，如果有则让它运行，然后才开始运行下一个低优先级进程（同样，这个过程中也会检查是否有高优先级和级别为关键的进程）。最后调度程序返回第一步重新开始

上述的调度算法与“跑表”的操作类似。把秒看作级别为最高（关键）的进程，把分看作高优先级的进程，把小时看作中优先级的进程而把日看作低优先级的进程。秒针要走过表

盘上所有的秒，分针要走过表盘上每一分，时针要走过每一小时，包括每一个走过完整的秒的分针，等等。

与跑表一样，调度算法也有一个内置的重置特性。算法在确定没有高优先级的进程等待运行之后才开始检查中优先级队列。一旦发现有高优先级的进程，它就又从头开始，检查级别为关键的进程，从而高优先级进程就像到达第 1 小时而被重置为 0 的跑表。

可能会出现这种情况：一个进程长时间一直独占 CPU，这样系统本身就没有 CPU 时间来运行其他程序。为了减少一直在运行的进程的影响，IOS 采取一种进程监视时钟来允许调度程序定期地决定正在运行的进程是否继续运行下去。这个特性和抢先并不一样，它只是一种阻止系统因为一个进程占用全部 CPU 时间而变得反应迟钝或完全死掉的一种机制。如果一个进程将要被挂起（例如：它运行的时间太长了），调度程序也能强行终止该进程。

当调度程序允许一个进程占用 CPU 时，它就会为该进程分配一个监视时钟。经过一段预先设置好的时间，缺省为 2 秒，如果该进程仍在运行，时钟就会超时，调度程序重新获取控制权。

（2）存储管理程序

存储管理程序把整个物理存储器映射成一个大的、连续的虚拟地址空间。即使系统没有应用完全的虚存机制，但是当需要创建虚拟地址空间的时候，也会用到 CPU 的 MMU。为了减少开销，内核并不进行任何存储页的调度或交换，所以虚拟地址空间受限于实际的物理存储器的大小。

系统把连续的地址空间分为不同的存储区域，这些存储区域称为区。区也和各种类型的物理存储器相符合，例如在一个给定的路由器上，SRAM 用来存储报文，DRAM 则可能用来存储软件和数据。把存储器分成不同的区有利于各种类型的存储器分组，从而软件就不必关心每一种平台上特定存储器的类型。

内核的存储管理程序负责管理路由器系统所有可用的存储空间，包括含有系统映像本身的存储空间。一般的实现中，存储管理程序大体上可以分为三个独立单元：区管理程序、池管理程序和组块管理程序。区管理程序定义和管理一个平台上的各种存储区；池管理程序负责存储池的创建以及池内存储块的分配与收回；组块管理程序管理特别分配的、包含多个固定大小子块的存储块。

区管理程序负责维护所有的存储区。它提供能允许系统软件其它部分创建区并设置区属性的服务。它也允许系统软件其它部分查询可用的区，例如决定一个平台上可用的整个存储器的容量。存储区被分为 8 类，存储区也可以按“父－子”关系进行嵌套分类。虽然对嵌套的深度没有限制，实际上只有 1 级的嵌套可用。如表 5-3 所示。

表 5-3　　Cisco 路由器存储区分类

存储区分类	特　　性
本地（Local）	运行时间数据结构和本地堆栈，一般是 DRAM。
Iomem	CPU 和网络介质通过一个数据线共享，一般是 SRAM。
快速（Fast）	快速存储器，例如 SRAM，为特定目的和关键任务所用。
I Text	可执行的 IOS 代码。

续表

I Data	初始化变量。
I BSS	没有初始化的变量。
PCI	PCI 总线存储器，为 PCI 总线上所有的设备可用。
闪存（Flash）	这个存储区可以用来存储以闪存运行或从 RAM 运行的 IOS 映像。也常被用来存储路由器配置和其它数据的备份，比如：崩溃数据。一般来说，文件系统经常存储在闪存区域。

IOS 使用存储池管理可用的空闲存储区域，这些存储池也是通常意义上的堆。池管理程序是内核中比较重要的一个单元。正像调度程序为进程分配 CPU 一样，它负责为进程分配存储器空间。所有的进程都必须直接或间接地通过池管理程序为它们分配存储空间。每次进程都是通过标准系统函数唤醒池管理程序来分配或收回存储容量。

每个池都是一个存储块（Memory Block）的集合，这些块根据需要可以被分配也可以被收回。存储池在区外构成并由内核管理。通常一个池对应一个特定的区，但这也不是必需的。一个存储池可以从存储器扩展的几个区来构造，它允许分配存储器并重新声明各种各样的区，从而达到高效访问。

池管理程序通过维持各池内空闲的存储块列表进行工作。刚开始每一个池只是含有一个和池空间一样大的块。随着池管理程序不断响应对存储空间的请求，最初的空闲块变得越来越小。同时，进程也可以释放存储空间给池，这样就生成了许多各种大小的、不连续的空闲块。这种现象称为存储碎片现象。

随着空闲块返回到一个池，池管理程序把它们的大小和起始地址加进一个记录小容量块的列表中。当一个进程需要从池中申请存储容量时，池管理程序首先从和该进程申请的容量大小相当的空闲列表开始搜索。这个方法有助于把那些已经回收的、容量大小与请求匹配的块的分配，从而提高了存储器的利用效率。如果在一个最佳匹配的列表中没有可用的块，池管理程序就会查找更大容量的列表直至找到一个可用的空闲块。如果在一个更大容量列表中找到了一个可用的块，池管理程序就会把该块分开，把没用的那部分放入相应较小的空闲块列表中。

池管理程序通过合并那些物理上相邻的块来控制碎片问题。当一个块被收回且它与一个原来已经存在的空闲块相邻，池管理程序就把它们两个合并成一个大的块，然后放入相应容量的块列表中。

虽然池管理程序提供了一种非常高效的方法管理那些任意大小存储块的集合，但它要为每个块的管理花费 32 个字节的存储开销。这个开销对那些拥有几百个大块的池来说是微不足道的，但对那些有几千个小块的池来说，就变得比较大了。作为一种优化方法，内核提供另一种存储管理程序，称为组块管理程序，来管理那些由很多小块组成的大池而不必付出相应的开销。

组块管理程序并不创建额外的、带有各种容量大小列表的存储池。而是管理一个大小固定的块的集合，这些大小固定的块是对那些从某个标准存储池中分配出来的大块的又一次细分。从某种意义上说，组块管理程序可以被看作是一个子存储池管理程序。

这个策略一般这样实施：一个进程向一个特定的存储池请求分配一个大存储块，然后进程再申请组块管理程序把分配到的大块再分成由一系列较小的、容量固定的组块组成的集合。随后再根据需要，利用组块管理程序对那些组块进行分配和回收。这种方法的优点就是只有 32 字节的开销（申请大块时），池管理程序也不必再负担成千上万小碎片的分配和回收，所以池中碎片的潜在开销被减少。

即使系统采取各种优化策略来实现存储器资源的高效管理，但是仍然存在一些不可忽视的问题。前面的叙述基于假设一旦进程请求分配存储容量，它的请求总被满足。然而，可能不经常是这样。存储器资源同 CPU 资源一样也是有限的，经常会出现没有足够的存储容量来满足请求的情况。当一个进程向池管理程序申请存储空间而其请求却没被满足时会发生一些不可想象的情况，最坏的情况就是系统死机。

基本上，有效的存储空间的分配请求没被满足的原因有两个：没有足够可用的存储空间或者有足够的空间，但是都是小而不连续的碎块，它们都没能达到申请的容量大小。

因存储空间不够而导致请求失败的情况经常发生，因为随着网络流量的增加和规模的扩大，没有足够的存储空间来支持系统中所有活动进程的情况是有可能发生的，比如路由表的大规模增长使系统的存储器被占完，这对于骨干网高速路由器来说是一个致命的缺陷。当这种情况发生时，只有两种选择：给路由器增加存储器或减少系统配置的特性、接口等等直至问题解决。在一些情况下（极少数的），存储器不足也可能因为某个进程中存在一个称为内存遗漏的缺陷而发生。内存遗漏是指这样一种现象：一个进程一直连续分配到存储块，但从不释放或者只释放一部分，随着系统的运行，最后所有的可用存储空间都被它占用。这种情况一般是由于软件缺陷而引起的。

另外一种原因就是系统存在严重的存储碎片问题。最大的可用连续存储块都不能达到进程所请求块的大小。发生这种情况的原因就是当许多小的存储块被分配，然后，存储池又以释放后不能把它们合并成大块的方式被释放。随着这种趋势的继续，池中大的存储块都被分成小的碎片直至最后只剩下小的碎片。当所有大的块都用完的时候，存储分配失败就发生了。

一般都是把内核的池管理程序设计成维持自行修复存储池来避免碎片问题。大多数情况下，这种设计都会正常工作，然而有时一些分配现象可能会使这种设计方法失败。

3．报文缓冲管理

在路由报文时，像任何存储转发操作一样，系统必须有一个地方存放那些即将被路由出去的数据。典型的策略就是当进行报文交换操作时，创建一个存储缓冲来保持流入的报文。因为路由报文的能力是整个路由器软件体系结构的核心，一般系统中会包含一个特殊的单元来管理那些缓冲，在 IOS 中，这个单元被称为缓冲池管理程序。IOS 就是用这个单元为交换操作创建和管理一系列连续的报文缓冲池，池中的缓冲统一被看成系统缓冲区。

缓冲池管理程序提供了管理由各种大小缓冲区组成的集合（或池）的一种很方便的方法。虽然它能被用来管理各种类型的缓冲池，但主要是用来管理报文缓冲池。

报文缓冲池用从那些可用存储池中分配的存储空间来创建。为了创建一个池，报文缓冲管理程序向内核的存储池管理程序申请一个存储块，然后再把申请到的块分为许多缓冲区。随后报文缓冲管理程序就构建一个所有空闲缓冲区组成的列表，并跟踪这些缓冲区的分配和释放情况。报文缓冲池可以分为私有池或公共池。公共池能被任何系统进程所用，而私有池是为一个特殊的使用它们的进程集合所创建（并且只为这些进程所知）。

报文缓冲池可以是动态的也可以是静态的。静态池创建时有固定的缓冲区数——以后也不能再增加。动态池创建时有一个特定的最小缓冲区基数，称之为永久缓冲区，但是以后根据需要还可以再添加或删除池里的缓冲区。使用动态缓冲池时，如果池管理程序接到一个申请缓冲池的请求，而当时池是空的，它就会试图扩展缓冲池并马上满足请求。如果在请求的上下文里，缓冲池不能扩展，就不能满足请求，但随后还会在池管理程序背景进程里扩展缓冲池。

4．设备驱动程序

操作系统的一个主要功能就是在硬件平台与在硬件平台上运行的软件之间提供一种硬件抽象，这种硬件抽象的一般实现方法就是提供设备驱动程序。硬件抽象也是操作系统的一部分，而且有了硬件抽象，相对应的那些硬件单元也成为整个系统的一部分。

路由器系统软件包含许多硬件设备驱动程序，如：闪存、NVRAM 驱动程序，但最重要的是它的网络接口设备驱动程序（对线卡的驱动）。网络接口驱动程序，在处理报文流入或流出接口时，为接口提供了较高的智能操作。每一个驱动程序包括两个主要的单元：控制单元和数据单元。控制单元负责管理设备的状况和状态。数据单元负责所有通过设备的数据流操作，同时也包含报文交换操作逻辑。设备驱动程序同报文交换函数有非常密切的关系，我们在讨论路由器的报文交换结构时已经提及过。

接口驱动程序通过一个称为接口描述块（Interface Descriptor Block，IDB）的特殊控制结构与系统的其他部分相连。IDB 包含指向设备驱动程序函数的入口项和关于设备状况和状态的数据，例如：IP 地址、接口状态以及一些在 IDB 中出现的其他域的报文统计信息等等。系统软件在为每一个线卡维护一个 IDB 的同时，也为线卡上的每个接口维持一个 IDB。

5.2 路由器系统软件的实现方法

其实，路由器系统软件并没有严格意义上的定义。从系统的角度看，路由器软件系统包括操作系统、协议软件、驱动软件以及一些网络管理和一些其他的功能模块。这里所说的路由器系统软件主要是指路由器的操作系统。路由器的操作系统有多种实现方式，最简单的可以是在一台微机上运行 Linux 系统，再加入相应的协议包就可以当作一台路由器使用，这时可以把 Linux 看作这种路由器的操作系统。而目前大多数路由器厂商使用嵌入式实时操作系统（如 Psos、VxWorks 等），Cisco 公司使用的是自己开发的专用操作系统 Cisco IOS，这也是本书多次引用当作实例的一种实现方法。

本小节主要介绍路由器系统软件的一种实现方法——嵌入式实时操作系统 VxWorks，有关 VxWorks 的详细情况请访问风河公司的网站：www.windriver.com 或者中国地区代理商奥索公司网站 www.autosoft.com.cn。

路由器软件系统结构的很大一部分都致力于使报文交换更加迅速、更加高效。虽然组成系统的各部分组件中有许多也是通常所说的操作系统的组件，但这些组件都包括为系统特别设计的一些关键的功能。

当前的计算机网络广泛采用了各种广域网技术，并且建立在形形色色的局域网媒体之上，因此，一个局域网必须通过网络互连设备与其它网络进行通信。这些网络互连设备，如

路由器，就是一种特殊的嵌入式系统。它的操作系统不仅仅具有嵌入式操作系统的基本功能，而且具有强大的网络通信、网络路由、网络管理功能，我们称之为网络实时操作系统。目前，实时操作系统已经由以前的嵌入式系统的统一实体的结构演变为最低层是一个实时内核的层次结构。多任务、抢占调度、快速上下文切换、低中断延时和快速灵活的通信机制是对现代实时内核的标准要求。

5.2.1　对实时系统的要求

计算机早期开发的操作系统的最原始的结构形式是一个统一的实体(monolithic)。在这样的系统中提供的不同功能的模块，如处理器管理、内存管理、输入输出等，通常是独立的。然而它们在执行过程中并不考虑其他正在使用中的模块，各个模块都以相同的时间粒度运行。由于现代实时环境需要许多不同的功能，以及在这样的环境中存在的并发活动所引起的异步性和非确定性，操作系统变得更加复杂。所以早期操作系统的统一结构的组织已经被更加精确的内部结构所淘汰。

操作系统最好的内部结构模型是一个层次性的结构，最低层是内核。这些层次可以看成为一个倒置的金字塔，每一层都建立在较低层的功能之上。内核仅包含一个操作系统执行的最重要的低层功能。正像一个统一结构的操作系统，内核提供了在高层软件与下层硬件之间的抽象层。然而，内核仅提供了构造操作系统其他部分所需的最小操作集。

实时操作系统内核需满足许多特定的实时环境提出的基本要求。总的来说，对一个实时内核的要求主要包括：

- 多任务：由于真实世界的事件的异步性，能够运行许多并发进程或任务是很重要的。多任务提供了一个较好的对真实世界的匹配，因为它允许对应于许多外部事件的多线程执行。系统内核通过将 CPU 分配给这些任务来获得并发性。
- 抢占调度：真实世界的事件具有继承的优先级，在分配 CPU 的时候要注意到这些优先级。基于优先级的抢占调度，任务都被指定了优先级，在能够执行的任务（没有被挂起或正在等待资源）中，优先级最高的任务被分配 CPU 资源。换句话说，当一个高优先级的任务变为可执行状态，它会立即抢占当前正在运行的较低优先级的任务所占有 CPU 资源。
- 快速灵活的任务间的通信与同步：在一个实时系统中，可能有许多任务作为一个应用的一部分执行。系统必须提供这些任务间的快速且功能强大的通信机制。内核也要提供为了有效地共享不可抢占的资源或临界区所需的同步机制。
- 方便的任务与中断之间的通信：尽管真实世界的事件通常作为中断方式到来，但为了提供有效的排队、优先化和减少中断延时，我们通常希望在任务级处理相应的工作。所以需要在任务级和中断级之间存在通信。
- 性能边界：一个实时内核必须提供最坏情况的性能优化，而非针对吞吐量的性能优化。我们更期望一个系统能够始终以 50 微秒执行一个函数，而不期望系统平均以 10 微秒执行该函数，但偶尔会以 75 微秒执行它。
- 特殊考虑：由于对实时内核的要求的增加，必须考虑对内核支持不断增加的复杂功能的要求。这包括多进程处理和对更新的、功能更强的处理器结构（如：RISC）的支持。

目前，许多商用化的内核支持的功能远强于上面所列的要求。在这方面，它们不是真正的内核，而更像一个小的统一结构的操作系统。因为它们包含简单的内存分配、时钟管理，

甚至一些输入输出系统调用的功能。

5.2.2 VxWorks

WindRiver 公司的 Tornado II 开发平台极大地缩短了嵌入式开发者开发产品的时间。Tornado II 的一个部件（Tornado 工具）包含了一个功能强大的核心套件和一些可选的交叉开发工具和组件。Tornado II 的其他集成部件包含 VxWorks 实时系统，实时系统是一种在目标机处理器上执行的高性能的、可裁剪的实时操作系统。

VxWorks 操作系统是一种功能最全的，现在可以获得的独立于处理器的实时系统。然而，VxWorks 是带有一个相当小的真正微内核的层次结构。内核仅提供多任务环境、进程间通信和同步功能。这些功能模块对 VxWorks 在较高层次提供丰富性能的要求有足够的支持。

它包括一个微内核、强大的网络支持、文件系统和 I/O 管理、C++支持的各种模块。与此同时 VxWorks 还支持超过 320 家的合作伙伴公司的第三方产品。通常内核操作对于用户是不可见的。应用程序为了实现需要内核参与的任务管理和同步使用一些系统调用，但这些调用的处理对于调用任务是不可见的。应用程序仅链接恰当的 VxWorks 例程（通常使用 VxWorks 的动态链接功能），就像调用子程序一样发出系统调用。这种接口不像有些系统需要一个笨拙的跳转表接口，用户需要通过一个整数来指定一个内核功能调用。

VxWorks 是一个具有可伸缩、可裁剪、高可靠性，同时适用于所有流行的 CPU 平台的实时操作系统。所谓可伸缩性是指 VxWorks 提供了超过 1800 个应用编程接口（API）供用户自行选择使用；所谓可裁剪性是指用户可以根据自己的应用需求对 VxWorks 进行配置，产生具有各种不同功能集的操作系统映像；所谓可靠性是指可以胜任一些诸如飞行控制这样的关键性任务。总体来说，VxWorks 具有以下主要特征：

- 高性能的微内核设计：VxWorks 的微内核具有全部实时特性，包括迅速的多任务调度，中断支持以及同时支持抢占式调度和时间片轮转调度。与此同时，该微内核还具有系统负载小，对外部事件的响应时间确定等特点。VxWorks 还提供广泛的任务间通信机制，包括共享内存、消息队列、Sockets、远程过程调用（RPC）和信号。同时，还提供三种信号量：二进制信号量、计数信号量、互斥信号量。
- 可裁剪的运行软件：VxWorks 在设计之初就具有可裁剪特性，使得开发者可以对操作系统的功能、大小进行增减，从而为自己应用程序提供更多的系统资源。例如：在深层嵌入式应用中，可能操作系统只有几十 K 的存储空间，而对于一些高端的通信应用，几乎所有的操作系统功能都可能需要。这就要求开发者能够从 100 多个不同的功能选项中生成适用于自己应用的操作系统配置。这些独立的模块既可以用于产品中，也可省去。利用 Tornado II 的工程项目管理工具，可以十分轻松地对 VxWorks 的各种功能选项进行增减。
- 丰富的网络支持：VxWorks 是第一个集成标准 TCP/IP 网络功能的实时操作系统。到目前为止，VxWorks 的 TCP/IP 协议支持最新的 Berkeley 网络协议，如：IP、IGMP、CIDR、TCP 等。同时，Wind River 公司还提供一些可选的网络协议产品，如 Wind Net SNMP v1、2 等。另外，Wind River 公司还与许多第三方厂家一起，提供各种完备的网络解决方案，例如：用于广域网的 ATM、SMDS、帧中继（FR）、ISDN、7 号信令（SS7）、X25 和 V5 等网络协议；IPX/SPX、AppleTalk、SNA 等局域网协议；RMON、CMIP/GDMO 和用于对分布式网络

进行基于 Web 的网络管理的解决方案；用于分布式计算环境的 CORBA 产品。

■　POSIX1003.1b 兼容：VxWorks 支持 POSIX 1003.1b 规范以及 1003.1 规范的基本系统调用。包括：进程原语、文件目录、I/O 原语、语言服务以及目录管理。另外，VxWorks 还遵循 POSIX 1003.1b 实时扩展标准，包括：异步 I/O、计数信号量、消息队列、信号、内存管理(页面锁定)以及调度控制。

■　BSP 移植：Wind River 公司提供大量的预制的支持许多商业主板及评估板的 BSP。同时，VxWorks 的开放式设计以及高度的可移植性使得用户在使用不同的目标板进行开发时，所做的移植工作量非常小。到目前为止，Wind River 公司能够提供超过 200 个 BSP，当用户在为自己的目标板开发 BSP 时，可以从 Wind River 公司的标准 BSP 中选一个最接近的来加以修改。

■　提供操作系统可选附件：为了扩展 VxWorks 的功能，Wind River 公司还提供了一些可选附件，包括 BSP 开发工具包，支持 Flash 文件系统的 TrueFFS 组件；用于虚拟存储管理的 VxVMI 组件；用于支持多处理器的 VxMP 组件和 VxFusion 组件；以及各种图形方面的组件。

表 5-4　　列出了 VxWorks 技术上的特点。

微　内　核	
高效的任务管理	支持多任务，任务数没有限制。同时支持抢占式调度和时间片轮转调度。快速、确定的上下文切换。256 个任务优先级。
快速、灵活的任务间通信	具有优先级继承特点的二进制计数以及互斥信号量；消息队列；POSIX 的管道、计数信号量、消息队列、信号；Socket；共享内存（Shared Memory）。
高度可裁剪性	开发者可以对操作系统的功能、大小进行增减，从而为自己应用程序提供更多的系统资源。
其他	增量连接和加载组件、快速、确定的中断响应、优化的浮点支持、动态内存管理、系统时钟以及定时器支持。
网　络　支　持	
BSD 4.4 网络	BSD 4.4 TCP/IP 网络。
网络协议和传输协议	IP、IGMP、CIDR、TCP、UDP、ARP。
路由协议	RIPv.1/v.2、WindNet OSPFv.2（可选）、WindNet STREAMS SVR4（可选）。
Sockets	标准的 Berkeley Sockets，以及“零拷贝”Sockets。
网管	WindNet SNMP v.1/v.2c 以及 MIB 编译器（可选）。
其他	SLIP、CSLIP、PPP。 BOOTP、DNS、DHCP、TFTP。 FTP、rlogin、rsh、telnet。 NFS、ONC RPC。

续表

快速、灵活的 I/0 和本地文件系统	
I/O	POSIX 异步 I/0 以及目录管理、SCSI 支持。
文件系统	MS-DOS 兼容的文件系统、Raw 文件系统、Flash 文件系统（TrueFFS，可选）、ISO 9660 CD-ROM 文件系统、PCMCIA 支持。
目标机开发特性	
兼容性	完全 ANSI C 兼容，同时支持一些增强的 C++特性；POSIX 1003.1 b 兼容。
调试	交互的 C 命令解释器、完全的符号调试、强大的性能监视功能、动态链接、超过 1800 个系统调用、 通过以太网、串口、ICE 的系统级调试。

5.2.3 JUNOS 软件系统

软件的结构决定了系统将被如何设计及不同组成部分间如何进行连接和相互操作。许多关心路由协议和路由器配置的网络专家并不会花费时间去分析提供系统运行基础的下层结构。软件结构在决定网络的控制，稳定性，性能，可管理性，和复杂软件系统的可扩展性上将作为一个重要的角色。因为传统路由系统是在假设它们将支持对时间敏感的任务（即，包转发）的前提下被开发的，设计者必须要开发一个具有实时操作系统单元的运行环境。即使意识到普通的操作系统可以提供极高的系统可靠性和稳定性，但同时他们也知道，一个多进程系统中所固有的由于内存管理及工作的重复而导致的性能低效性。这些传统路由系统的设计人员决定，他们将通过合并代码和避免重复的方式来增强整个系统的性能，即使这意味着将要建立一个单一的，非模块化的代码库。

随着我们步入 Internet 的黄金时代，基于实时的，非结构化代码库的传统路由软件结构在支持快速出现的新功能和 Internet 核心网所需的稳定运行上，都显得力不从心。现在，在高性能光接口上实时地转发业务要求配置基于硬件的转发引擎。因此，下一代的路由软件不再需要对包转发和高级系统功能之间的资源竞争进行处理。基于硬件的转发引擎的效率，允许路由软件运行在一个能够提供更高的可靠性、可伸缩性、有效性，和可为敏感的重要任务应用提供高性能的普通操作系统环境之中。

Juniper 网络公司专为高性能和高速增长的 Internet 服务提供商开发出了一套 JUNOS Internet 软件。本小节讨论 Juniper 网络公司的 JUNOS 软件系统总体设计和一些相关的性能，JUNOS 软件已准备好服务于现在的 Internet，并对将来的发展奠定了坚实的基础。

当对一个路由软件进行评价时，对其软件结构、路由协议、策略定义语言、流量工程能力、用户界面、系统安全性，和网络管理性能的检验是非常重要的。上述的每一个特性都将决定该软件可以为 ISP 提供成功进入 Internet 下一阶段成长所需控制的能力。

1．源于 FreeBSD 的 JUNOS

FreeBSD 为开发支持高速增长的 Internet 下一代路由体系提供了基础。FreeBSD 是为在普通的英特尔处理器上运行而特别设计的。它非常稳定，并继承了那些从 20 世纪 70 年代初

期便开始工作在 Internet 上的早期产品的成熟的网络功能，它包含了一个非常优秀的代码库以支持内核、文件系统、用户计费和安全性。

但是，Juniper 网络公司加强并重写了一部分 FreeBSD，因为它最初是被设计运行在主机系统上的，只支持几种有限的网络接口。而另一方面，路由器具有更多种类的物理接口和子接口，并具有一个更大的路由表。另外，大部分的网络代码被去除或以行业级的工具来代替，以支持来自 Internet 的巨大压力。Juniper 网络公司的工程队伍具有多年的 Internet 路由经验，并且在不受传统路由体系约束的情况下，从最底层开始开发和设计路由体系。这就意味着，他们可以优化其数据结构，计划大量的虚电路，并对巨型路由表的存储及查询进行设计。他们可以自由地使用最新开发出的支持流量工程和不同服务等级的技术，着力设计合适的用户界面和强有力的策略定义语言。

2．软件体系

Juniper 网络公司软件体系的最基础的设计思路是将控制功能与包转发功能分离。路由引擎管理系统的路由和控制功能，并运行从 FreeBSD 得到的内核。包转发引擎在硬件上运行，专门用于包转发。这两项功能的完全分离，使路由器可以提供高性能和高可靠性的操作系统。

包转发引擎负责提供所有的包转发功能。这代表了进入系统的业务量的 99%。包转发引擎由几个可处理所有类型的数据包并专用于 Internet 核心网的 ASIC 组成。路由引擎负责执行整个系统的路由处理。它通过一个能够提供充足计算周期的处理器，在任何网络环境下提供所需的功能。路由引擎运行着经过修改的 FreeBSD 版本，使其能够在高负荷条件下稳定运行。

路由协议、接口管理、机箱管理、SNMP 管理、系统安全性，和用户界面都作为一个子系统在操作系统内协调工作。每个程序执行一个独立的进程，完全在自己的内存保护下运行。这便减少了一些失控应用干扰其它的应用或内核的机会。JUNOS 路由表包含了从相邻路由器获得的和静态配置的路由信息。转发表是从路由表得到的，它包含一个用于协调带有输出端口的报头或标记的 IP 报头和多协议标记交换（MPLS）标记的索引。包转发引擎使用转发表中的内容而不是路由表中的内容决定进行转发。

3．JUNOS 体系的优势

基于将控制与转发平台分离的路由体系的实现，使 Juniper 可以在普通操作系统之上运行路由引擎。这是一个非常关键的设计特点，它使 JUNOS 具有很高的可靠性、可维护性和性能。

■　保护内存确保运行的可靠性：每个用户进程都在其自己的保护内存空间中运行。这样便确保了一个子系统故障不会影响其它在保护内存下执行的子系统的运行。在这些独立的操作之间，Juniper 网络为内部进程通信建立了整齐的、良好定义的接口。这种结构使软件系统具有很好的可靠性。

■　专门为 ISP 网络设计：Juniper 网络公司专注于为高速增长的 Internet 骨干网提供产品。这意味着 Juniper 网络公司的产品不会运行在企业环境下，也不致为满足不同特殊应用需求而建立非结构化的代码库。

■　增强的网络稳定性：Juniper 网络公司的软件体系使得传输的包永远不会通过路由引擎进行处理。对于那些被要求送至路由引擎的业务、链接保持和路由协议更新将被赋予最高的优先权，以确保无论系统的负载情况如何，邻接永远不会断开。这种控制业务的优先权防止了网络中的级联故障，因为它确保了无论系统发生什么情况，链接和路由邻接都一直保持运行。

第 6 章　高速接口设计技术

传统的路由器通常是基于总线和集中处理器结构，其处理能力一般是几十万个包/秒，最大的吞吐能力约 1 个 Gbit/s 左右，而 SDH 的接口速率通常为 STM-4（622Mbit/s）、STM-16（2.5Gbit/s），传统的路由器显然不能适应于 POS。Internet 骨干网上业务量的激增对路由器的处理能力及容量提出了更高的要求，随之也产生了许多千兆级路由器的设计，如 Cisco 的 7500 系列、12000 系列路由器，Ascend 的 GRF 系列路由器，Lucent 的 PacketStar 6400 系列路由器。这些千兆级路由器抛弃了传统的总线/背板加集中处理器的结构，代之以高性能的专用或通用的交换矩阵，有些甚至直接采用了 ATM 交换矩阵；同时将原来集中在中央处理器的智能尽量分散至各个接口处理模块，希望通过高速缓存和其它的路由预处理手段加速数据包的转发，如 Cisco 的 VIP 结构。经过一系列结构上的改进，路由器的吞吐量有了很大的提高。但它们毕竟还是传统意义上的路由器，还依据路由表进行数据转发，这仍然没有脱离传统路由器的局限性——对庞大路由表的检索。

在路由器的设计中，各种接口设备的设计至关重要。因为如果没有高效的、快速的接口设备，路由器就无法正常工作，接口就会成为制约路由器性能的瓶颈。另外核心的交换设备和中央控制设备一般只有一个模式，而路由器所连接的网络可能是多种多样的。为了使网络类型对路由器是透明的，就要使接口能够将各种网络的数据无差别地传输给核心设备，同时还不能成为瓶颈。现在路由器主要采用以太网和 POS 两种接口，本章也分以太网和 POS 两种接口介绍它们的设计。

6.1　以太网接口设计

以太网实际上是对 IEEE 802.3 协议的一种实现。IEEE 802.3 协议适用于 CSMA/CD 局域网。其工作原理是：当站点希望传输时，它就侦听传输电缆，如果线路正忙，就等到线路空闲为止，否则就立即传输。如果两个或多个站点同时在空闲的电缆上开始传输，它们就会冲突。于是所有冲突站点中止传输，等待一个随机时间后，再重复上述过程。以太网就是根据 802.3 协议建立的网络，这种网络模型在局域网中的工作效率比它在广域网中高，所以现在局域网大多采用以太网设计模型。以太网一般分为千兆以太网和 100/10 兆以太网，这在后面将会具体介绍。

6.1.1　以太网的基本原理

早期以太网传输速率为 10Mbit/s。随着网络技术飞速发展，多媒体应用越来越多，对网络的需求也越来越大，尤其是在服务器端上，100Mbit/s 的速度已不能满足要求。于是 Gigabit

Ethernet 诞生了。就如同 Fast Ethernet 的起源一样，Gigabit Ethernet 也必须要能够向下兼容 Fast Ethernet 以及 Ethernet。目前中大型企业新一代的区域网络规划中，Gigabit Ethernet 普遍使用在区域网络的骨干上，并以光纤介质为主流。在铜线（UTP）Gigabit 部分，短期内则还不会像 100baseTX 那样快速延伸至桌面。

1．以太网

以太网是一个共享的网络，这意味着许多用户同时在访问网络，而访问权的授予是根据先来先用的原则。

以太网的基本特征是采用一种称为载波监听多路访问/冲突检测 CSMA/CD（Carrier Sense Multiple Access/Collision Detection）的共享访问方案，即多个工作站都连接在一条总线上，所有的工作站都不断向总线上发出监听信号，但在同一时刻只能有一个工作站在总线上进行传输，而其他工作站必须等待其传输结束后再开始传输。例如，当用户 A 要传送信息时，该用户计算机发出请求到它的内部网络接口卡（Network Interface Card，NIC）。NIC 用一个电子“耳”去倾听网上是否有其他计算机正在占用网络。如果电子“耳”没有倾听到其他占用，NIC 就将信息发送出去。如果用户 B 正在发送信息，用户 A 的 NIC 便保持等待，直到用户 B 的信息发送完毕再发。如果用户 A 和 B 恰巧同一时刻试图发送，便会发生冲突。如果发生这种情况，各方 NIC 便都保持等待并启动一个随机时钟发生器，它为用户 A 或用户 B 选择不同的时间去重新尝试传送。冲突检测方法保证了只能有一个工作站在电缆上传输。

以太网协议包含数据链路层协议和物理层协议。

2．快速以太网

100Base-T 快速以太网是从 10Base-T 以太网标准发展而来的。它保留了以太网的帧结构和 CSMA/CD 协议，使 10Base-T 和 100Base-T 站点间进行数据通信时不需要进行协议转换。随着桌面计算机和应用程序功能的增加，网络用户日益增加，所产生的数据量也越来越大，所以网络带宽就成为一个瓶颈问题。在一个时期内，100Base-T 较好地解决了这一问题。快速以太网能显著提高工作站和服务器的传输带宽，从而可以安全地增大网络上的负载。

3．千兆以太网

千兆以太网技术开始于 1996 年，目前已完成了大量的工作，并已经能够使用。千兆以太网技术的主要目的是为骨干网络提供 1Gbit/s 的带宽，并为现有快速以太网提供自然升级的办法，同时要尽可能地利用现有的网络管理工具和相应的培训。

在局域网中为了维持直径为 200m 的最大冲突区域、最小 CSMA/CD 载波时间，以太网时间片已从目前的 512 比特扩展到 512 字节（4096 比特），最小信息包大小仍为 64 字节。载波扩展特性在不修改最小包尺寸的条件下解决了 CSMA/CD 固有的时序问题。虽然这些改变可能会影响到小信息包的性能，然而这种影响已经被 CSMA/CD 算法中称作信息包突发传送的特性所抵消。

为了适应千兆以太网产品进入市场，标准中 1000Base-SX、1000Base-LX、1000Base-CX 版本都能适应目前经过时间考验的 Fiber Channel 技术。即采用 8B10 NRZ（不归零制）编码方式，提供 1.25Gbaud 的有效波特率，因此能提供全速 1Gbit/s 的数据速率。

1000Base-SX 系列采用低成本短波的 CD（compact disc）或 VCSEL 发送器；

1000Base-LX 系列使用相对昂贵的长波激光器；

1000Base-CX 系列则打算在配线间使用短跳线电缆把高性能服务器和高速外围设备连接

起来；

1000Base-T 系列是支持大量已安装的 5 类布线系统的新设计。并且为了克服 5 类线的缺陷，运用了复杂的数字信号处理 DSP 技术。1000Base-T 在传输中使用了全部 4 对双绞线并工作在全双工模式下，因此新增加的参数例如回波损耗以及远端串扰（FEXT）等就显得重要起来。这种设计采用 PAM-5（5 级脉冲放大调制）编码在每个线对上传输 250 Mbit/s。双向传输要求所有的四个线对收发器端口必须使用混合磁场线路。因为无法提供完美的混合磁场线路，所以无法完全隔离发送和接收电路。任何发送与接受线路都会对设备发生回波。因此，要达到要求的 10^{-10} 的错误率（BER）就必须抵消回波。1000Base-T 无法对频率集中在 125MHz 之上的频段进行过滤，但是使用扰频技术和网格编码能对 80MHz 之后的频段进行过滤。为了解决五类线在如此之高的频率范围内因近端串扰（在 PSNEXT 的情况下）而受到的限制，应该采用合适的方案来抵消串扰。使用网格编码可以增强抗干扰能力，这一结论对上面的所有情况都适用。

千兆以太网的最值得称赞的是继承了传统以太网技术价格便宜的优点。千兆技术仍然是以太网技术，采用与 10M 以太网相同的帧格式、帧结构、网络协议、全/半双工工作方式、流控模式以及布线系统。由于该技术不改变传统以太网的桌面应用、操作系统，因此可与 10M 或 100M 的以太网很好地配合工作。升级到千兆以太网不必改变网络应用程序、网管部件和网络操作系统，能够最大程度地保护投资，因此该技术的市场前景十分看好。Gigabit Ethernet 支持的网络类型，如表 6-1 所示。

表 6-1　　千兆以太网支持的网络类型

传 输 介 质	距　离
1000BaseCX Copper STP	25m
1000BaseT Copper Cat 5 UTP	100m
1000BaseSX Multi-mode Fiber	500m
1000BaseLX Single-mode Fiber	3000m

千兆以太网技术有两个标准：IEEE802.3z 和 IEEE802.3ab。IEEE802.3z 制定了光纤和短程铜线连接方案的标准，目前已完成了标准制定工作。IEEE802.3ab 制定了五类双绞线上较长距离连接方案的标准。

IEEE802.3z 标准被定义为支持应用在建筑物内垂直主干多模光纤和园区内主干单模光纤的 100BASE-LX，其链路长度分别是 550 米和 3 千米，支持应用在较短垂直主干和水平布线多模光纤的 1000BASE-SX，其链路长度是 260 米。802.3z 同时还定义了应用在室内铜质屏蔽电缆（150）的 1000BASE-CX，其链路长度 205 米。802.3 以 1.25G 波特率使用 8B/10B 编码，从而获得 1Gbit/s 的数据传输速率。

IEEE802.3ab 标准被定义为支持四对 5 类非屏蔽双绞线（UTP）的 1000BASE-T，其链路长度是 100 米。它提供半双工（GSMA/CD）和全双工 1000Mbit/s 的以太网服务。1000BASE-T 的拓扑准则和自动协商机制与 100BASE-TX 所用的相同。这样不仅简化了将其与传统以太网逐步集成的任务，还为生产 100Mbit/s 和 1000Mbit/s 双速 PHY 提供可能。后者确保了 1000BASE-T 设备能够“后退到”100BASE-TX 的操作，100BASE-TX 是 100BASE-T 技术中

的一种它是 100BASE-T 中传输速率最高的技术，因此为升级系统提供一种灵活方法。

CSMA/CD 工作描述：

- 要发送数据的站监听媒体信道；
- 如果媒体信道空闲，则发送；
- 如果媒体有载波（忙），则继续对信道进行监听。一旦发现空闲，便立即发送；
- 开始发送数据，并监听信道；
- 如果在发送过程中检测到碰撞，则停止自己的正常发送，转而发送一短暂的干扰信号（jam），强化碰撞，使 LAN 上所有站都能知道出现了碰撞；
- 根据回退策略（二进制指数回退算法）进行下一次尝试；
- 如果在发送过程中没有检测到碰撞，则发送成功。

4．以太网与上层网络层的关系

通过以太网传输高层网络协议的数据需要解决两个问题：报文的封装和地址的映射。

（1）报文封装

完整的高层数据报文（包含信息头和信息体）被封装在以太帧（Ethernet、802.3 LLC 或 802.3 LLC with SNAP）的数据段，如果高层数据报文长度大于数据段最大长度（1500、1496 或 1492 字节），则需要分段（发送方）/重组（接收方）；如果高层数据报文长度小于数据段的最小长度，则需要添加相应的补丁（pad），以凑足最小的 64 字节的 MAC 帧。

（2）地址映射

以太网帧以以太网地址寻址（确定目的地），而高层数据报文以高层协议地址寻址，这样就有一个地址映射转换的问题。以在以太网上运行 TCP/IP 为例，一个 IP 节点要向另一个 IP 节点发送 IP 报文，它必须首先知道对方的以太网地址，然后将自己的 IP 报文封装在一个发往对方的以太帧中发送。ARP（地址解析协议）专门用来完成从 IP 地址到以太地址的翻译。我们以一个例子来说明 ARP 的工作过程：

假设在一个以太网中有两台 IP 主机：HOST1（以太地址为 MAC1，IP 地址为 IP1）和 HOST2（以太地址为 MAC2，IP 地址为 IP2），HOST1 要向 HOST2 发送数据，HOST1 通过 ARP 协议来解析 HOST2 的 MAC 地址。HOST1 首先发送一个以太广播帧，其中封装了一个 ARP 请求报文，ARP 请求报文中包含有 HOST1 的 IP 地址、MAC 地址和要解析的 IP2。网上所有站点都接收到该广播帧，只有 HOST2 发现该报文是解析自己的 MAC 地址，HOST2 首先记录下 HOST1 的（IP1-MAC1）地址关联，然后发送一个 ARP 响应报文给 MAC1，ARP 响应报文包含 HOST2 的 IP 地址和 MAC 地址。HOST1 收到 HOST2 发送的 ARP 响应报文，记录下 HOST2 的（IP2-MAC2）地址关联，然后将 IP 报文封装在以太帧中发往 MAC2。

吉比特以太网是建立在以太网标准基础之上的技术。千兆以太网和大量使用的以太网与快速以太网完全兼容，并利用了原以太网标准所规定的全部技术规范，其中包括 CSMA/CD 协议、以太网帧（即数据包格式）、全双工、流量控制以及 IEEE 802.3 标准中所定义的管理对象。作为以太网的一个组成部分，千兆以太网也支持流量管理技术，它保证在以太网上的服务质量，这些技术包括 IEEE 802.1P 第二层优先级、第三层优先级的 ToS 编码位、特别服务和资源预留协议（RSVP）。

吉比特以太网还利用 IEEE 802.1Q 虚拟局域网支持、第四层过滤和千兆位的第三层交换。吉比特以太网原先是作为一种交换技术设计的，采用光纤作为上行链路，用于楼宇之间的连

接。之后，在服务器的连接和骨干网中，吉比特以太网获得广泛应用。由于采用第五类非屏蔽双绞线的千兆以太网标准业已出台，吉比特以太网正在进入中小企业用户。

6.1.2 具体接口设计

以太网接口由接口相关部分和接口无关部分组成。接口相关部分主要由接口协议芯片组成，主要有千兆接口和 100 兆/10 兆自适应接口，它们的结构关系如图 6-1 所示。

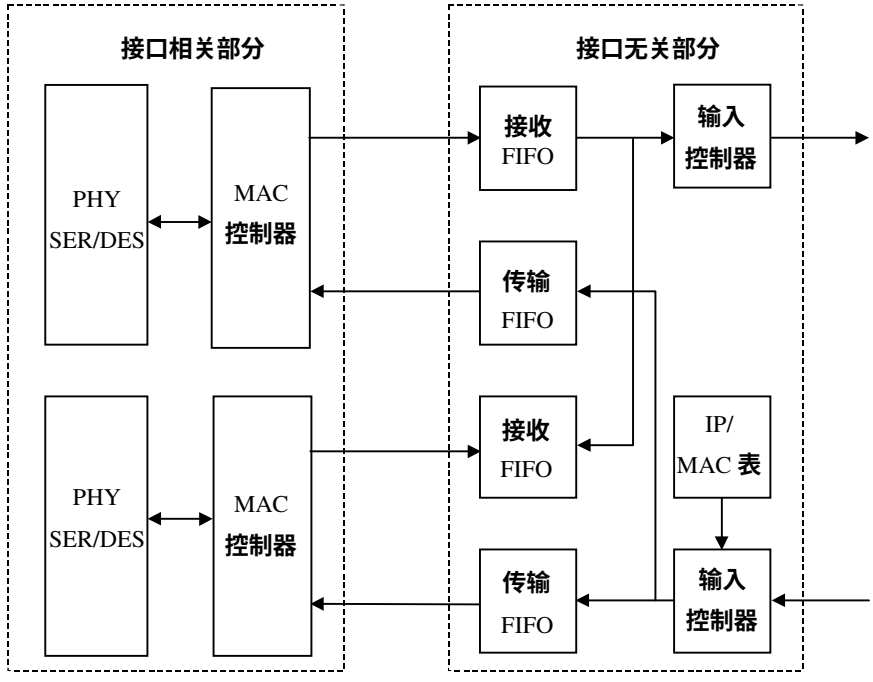


图 6-1 以太网接口结构

1．千兆接口相关部分

吉比特以太网接口相关部分主要由物理层收发装置和 MAC 层协议处理芯片组成。物理层收发装置主要是光电转换接口，它外接光纤接收和发送数据。因为光纤数据经常是一位宽，所以在物理层接口和 MAC 处理芯片之间一般要设置一个串并转换器。将一位宽的信号转换成多位宽的信号。

这 3 部分的连接，如图 6-2 所示。

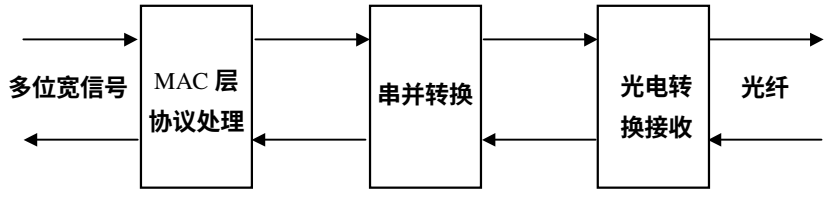


图 6-2 吉比特以太网组成

（1）MAC 层接口芯片

一般这种芯片包括三个部分：系统接口、物理层接口和寄存器控制接口。系统接口分为接收和发送两部分，接收通路和发送通路是独立的。

（2）转发器

吉比特以太网模块的收发转换功能主要就是实现字节流的串/并转换。此外还提供时钟产生/提取和帧构造等功能。

（3）光电转换器

主要用于将接收光信号转为电信号以及将电信号转为光信号送出。

2．千兆报文预处理

报文数据在系统接口和接口无关部分之间传递时，需要作以下预处理工作：

■　位宽转换：一般系统接口的数据宽度与路由器内部的数据宽度是不一样的。因此需要设计位宽转换器。

■　论询：因为有若干个以太网端口相互独立，它们的数据都汇总到一条输入路径或一条输出路径，因此无论在接收过程还是在发送过程中，都涉及到一个仲裁问题。考虑到效率和易实现性，设计采用轮询机制。

3．寄存器接口

软件对 MAC 层协议芯片中的寄存器的使用通过这个接口来进行。

4．10M/100M 接口相关部分

10/100M 接口相关部分的结构和 1000M 接口相关部分的结构是很相似的，只是所采用的芯片不同。

（1）MAC 子层

MAC 子层的构成，从功能模块上分为七个部分：系统接口、MAC、FIFO、PHY 接口、管理接口 MI、寄存器接口以及寄存器。其中系统接口、寄存器、PHY 接口、管理接口 MI 是公共部分，MAC、FIFO、寄存器接口相互独立，由每个通道各自享有。每个接口的设计都要依据所选择的芯片来完成。

（2）物理层

物理层一般由 4 个物理通道组成。每个物理通道都实现了标准的物理层功能。从功能模块上来看，可分为以下几个部分：控制接口、编码/解码器、扰波/解扰器、时钟、数据恢复、双绞线/光纤收发接口、自动协商模块及管理接口 MI。MI 由 4 个物理通道共同享有。

5．10M/100M 报文预处理

10M/100M 接口的报文预处理和 1000M 接口的报文预处理在位宽转换、轮询及寄存器接口等方面是相同的。但还要依据具体所选择的芯片来设计。

6．接口无关部分

接口无关部分包括接收/发送 FIFO、输入控制器、输出控制器、IP/MAC 表等。不管外部接口是 1000M 还是 10M/100M，它们的逻辑功能都是一样的。

（1）接收/发送 FIFO

该 FIFO 的宽度要和路由器内部数据宽度相同。

（2）输入控制器功能描述

输入控制器接收以太网端口来的数据，经 MAC 控制器处理后，得到数据链路层数据。分析链路层报文头，根据分析结果，决定是否滤掉 MAC 头（必要时再滤掉链路控制层的头），

得到 IP 报文，重新对齐边界并加上带外控制信息后按内部数据宽度“内部数据宽度”可以是 36 位或 72 位。为 1 组的格式写到母卡的接收 FIFO 中。

（3）输出控制器功能描述

输出控制器读取母卡发送 FIFO 中的报文，若是 IP 报文（包含带外控制信息），则根据 MAC 索引地址查 IP/MAC 表获得目的 MAC 地址，将目的 MAC 地址、源 MAC 地址、类型/长度加到 IP 报文的前面，并在重新对接边界之后根据二头中的目的子端口号按内部数据宽度为一组的格式写到发送 FIFO 中。若为直通报文则直接根据二头中的目的子端口号按内部数据宽度为一组的格式写入发送 FIFO 中。在此过程中，还要对数据进行校验，若发现有错，将错误状态寄存器中的相应位置“1”。输出控制器还提供线卡上寄存器访问接口和 IP/MAC 表更新的 CPU 接口。图 6-3 是输出控制器的逻辑结构。

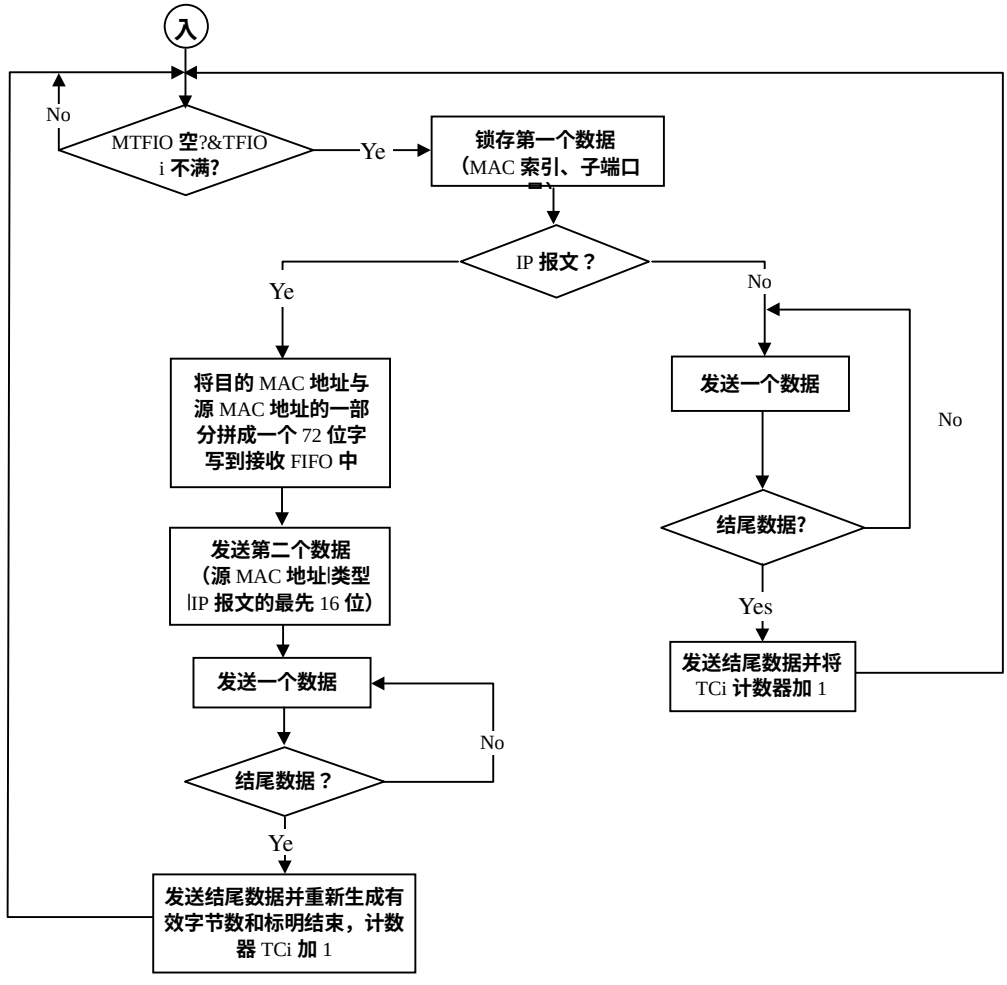


图 6-3　输出控制器的逻辑结构

（4）IP/MAC 表操作

发送过程中，IP 报文类型的数据需要加上目的 MAC 地址、源 MAC 地址及长度/类型域，以组成完整的链路层数据。目的 MAC 地址和源 MAC 地址存放在 IP/MAC 表中，IP/MAC 表

由双端口 RAM 构成，软件通过访问寄存器接口完成 IP/MAC 表的更新。输出控制器读 IP/MAC 表，获取目的 MAC 地址和源 MAC 地址。IP/MAC 表的地址索引由母卡在报文的第一拍数据中指明。非 IP 报文类型的数据即直通数据，无须处理可直接送往路由器外。

6.2　POS 接口设计

POS 的全称是 Packet OVER SONET/SDH。SONET 即同步光纤网络，它与 SDH 即同步数字分级结构只有微小的差别。因为 SONET 芯片越来越便宜，计算机 SONET 接口板可能会得到更加广泛的使用，这样各公司就可以更容易地将它们的计算机通过特殊的租用线路直接连接到电话网的核心。SONET 系统由交换机、多路复用设备和中继器构成，相互之间都由光纤连接。基本的 SONET 帧速率是每 125 μs 的 810 帧。由于 SONET 是同步的，因此不论是否有数据，帧都被发送出去。8000 帧/s 与数字电话系统中使用的 PCM 信道的抽样频率完全一样。

PPP 即点到点协议，PPP 能处理错误检测，支持多种协议，在连接时期允许商议 IP 地址，允许身份验证，以及在 SLIP 上所作的许多其他改进。PPP 成就了 3 件事：第一，明确地划分一帧的尾部和下一帧的头部的成帧方式。这种帧格式也处理错误检测工作。第二，当线路不再需要时，挑出这些线路，测试它们，商议抉择，并仔细地再次释放链路控制协议。这个协议被称为链路控制协议 LCP。第三，用独立于所使用的网络层协议的方法来商议使用网络层的哪些选项。对于每个所支持的网络层来说，所选择的方法有不同的网络控制协议。这在后面都将详细介绍。

6.2.1　POS 技术概述

POS 通过 SDH 直接承载 PPP 封装之后的 IP 数据包（IP/SDH），它与 SDH 间接地承载经过 ATM 协议的 IP 数据包（IP/ATM/SDH）相比有许多优点（以下将 RFC1483、RFC1577、LANE、MPOA 等 IP-ATM 相结合的模型统称为 IP/ATM 模型）。

POS 最主要的优点就是：高效地利用带宽资源，这对于广域骨干网是极为重要的。对于宽带广域网来说，POS 能够提供比基于 ATM 的网络高出 25%~30%的带宽。

具体说来：以 IP 层的数据包为始到映射入 SDH 净荷区为止，POS 方案需要把 IP 包封装入 PPP-HDLC 帧中，然后将 PPP 帧直接映射入 SDH 的净荷区；而基于 ATM 的方案，需要首先将 IP 包进行 RFC1483 的封装，最终成为 53 字节的信元，这时才能够映射入 SDH 净荷中（当然 ATM 论坛也提出裸信元的传送方式，但对于长途传输通常还是经过 SDH 网）。前者相当于每 48 字节净荷就有 5 字节开销，而后者仅相当于 1 500 字节（PPP 帧的缺省信息域长度）净荷才有 7 字节开销（PPP 帧的开销）。以 SDH 中 STM-4（OC-12）为例，其传输速率是 155.520Mbit/s，除去通道开销、段开销、线路开销后的可用信息速率是 149.760Mbit/s。经过 ATM 再映入 SDH，用户可用速率为 135.6Mbit/s；而直接将 PPP 帧映入 SDH，用户可用速率为 149.064Mbit/s。前者的效率仅为 90.54%，而后者为 99.54%。这也就是说对于 STM-64（2.5Gbit/s）的速率，基于 ATM 的方案将浪费约 235M 的带宽，相当于 5 个 T3 信道。以上还没有考虑 ATM 同 IP、SDH 相重复的 OAM 开销，以及 IP/ATM 模型引入的额外的协议开销所占用的带宽。所以从效率的角度讲，POS 方案无疑比基于 ATM 方案的带宽利用率要高得多。

目前全球 Internet 巨大的规模本身就证明了基于 IP 协议的路由器网络是具有极好扩展性的网络：路由器间动态地交换路由信息，使数据能够避开有故障的链路；采用 CIDR、VLSN 技术使目的地址到达信息可以在被概括后扩散；分级管理的体系和自治域，使系统间既相互独立又相互联系。POS 方案实际上仍是通过路由器组建网络，只不过路由器升级到千兆级，路由器间的连接也由专线升级到 SDH 的速率等级，因此继承了扩展性良好的优点。

事实上，单纯就 ATM 协议而言，它比 IP 协议具有更好的扩展性：ATM 的网间互连协议 PNNI 是迄今为止最复杂的路由/信令协议，具有结构化、等级化的特点，采取源路由的技术更能够提供许多先进的特性，如 QoS 路由、路由的折回等等。但是问题在于历史没有给予 ATM 机会来拓展全面的纯 ATM 网络；恰恰相反，众多业务在 IP 层上的汇聚，使 ATM 不得不以 IP 协议承载者的身份出现在数据网络中，从而出现了许多 IP-ATM 相结合的模型。从 RFC1577 到 LANE、MPOA，从 PAR 到 I-PNNI，面向连接的 ATM 始终试图无缝地支持非面向连接的 IP，但却暴露出许多问题。且不说功能上的重复和模型的复杂性所导致的性能问题，大多 IP/ATM 模型存在扩展性问题，这是 ATM 交换机中资源的有限性导致的。处理器资源、内存资源、VPI/VCI 资源的有限，直接导致以下的事实：一个 ATM 物理端口上不可能高速地建立大量的 SVC。而 CLIP、LANE、MPOA 等模型中，路由器间必须建立全网状网的 PVC 连接；而且运行过程中还会出现大量的直通 SVC，增加了资源的紧张，引入了时延。IP 到 ATM 的地址解析是一一对应的，由于二者的地址空间都很大，会出现巨大的地址映射表，难于维护和检索，特别是在 IP 路由的汇集处，这个缺点非常明显。所以由于 IP/ATM 模型的扩展性问题，导致纯路由器网络的规模和灵活性要好于基于 ATM 的 IP 网络。

众所周知，为了将 ATM 与应用广泛的 IP 协议相结合，提出过许许多多标准的、专用的模型，如最早的 RFC1483、FRC1577、LANE 和后来的 MPOA，以及 IP Switch、ARIS、CSR 等等。这些模型为了在 ATM 协议上运行非面向连接的 IP，额外提出了许多配置协议、地址解析协议，引入了协议上、管理上的开销。而且两种不同属性事物的结合给网络的建设、配置和故障的定位带来了许多新的问题。而 POS 只是传统路由器模型的自然延伸，不需要引入新的概念，模型简洁实用。而且 POS 直接依靠了 SDH/SONET 体系所具有的高可靠性。SDH 通常采用环形拓扑，并且具有自动保护倒换（APS）机制，在光纤线路出现节点故障时能够快速自愈。

特别值得一提的是，由 AT&T 和 BT 联手成立的全球合资公司 GlobalVenter 在其技术白皮书中明确表示其骨干网络将把 IP 协议直接建立于 SDH/DWDM 之上，仅在有必要时才用 ATM 兼容传统的应用（如承载帧中继），两大电信巨人的选择无疑向人们暗示了 POS 试图和 IP/ATM 分庭抗礼的趋势。并且加拿大的宽带实验网工程 CAnetII 也在 1998 年 10 月上旬的会议上，明确提出建立“光 Internet（Optical Internet）”的口号，其旨在将 IP/ATM/SDH、IP/SDH、IP/DWDM 统一到一个基于大容量光纤传输系统的网络上。这些说明，对以提供 IP 业务为主的服务商而言，POS 和未来基于 DWDM 的 IP 网络将是其最佳的选择。

因为 POS 是通过 SDH 提供的高速传输通道直接传送 IP 分组。POS 定位于电信运营级（carrier scale）的数据骨干网，其网络主要由大容量的高端路由器经由高速光纤传输通道连接而成。POS 实际上并不是一种全新的模型，而是对传统 IP 网络概念的顺延，它完全兼容传统的 IP 协议体系，只是在物理通道上借助 SDH 提供点到点物理连接，从而使速率提高到 Gbit/s。POS 模型主要涉及两个问题：数据的封装和高速路由器。

SONET/SDH 是物理层的协议，负责在信道上透明传送比特流；IP 是网络层的协议，负

责数据包由源到宿的寻址和路由。根据 OSI 七层模型，二者之间还需要一个链路层协议来进行帧级的定位和纠错。由于 SONE/SDH 是点对点的传输通道组成的，所以采用 PPP 作为链路层的协议顺理成章。

对于 PPP 协议，我们并不陌生。PPP 协议是针对点到点链路通信设计的，它常被用在短距离直接、租用线和拔号 Modem 中，作为其链路层的协议。根据 RFC1661，PPP 协议包括三部分：对多种协议数据的封装方法；用于建立、配置、检测链路的链接扩展协议（LCP）；用于建立、配置不同网络层协议的网络扩展协议（NCP）。对于 POS 而言，主要涉及的是 PPP 协议中有关数据封装的部分。PPP 协议的封装方案效率高，适于在 SONET/SDH 通道上采用。POS 的数据封装过程很简单，分为两步：将 IP 包封装入 PPP 帧中；将 PPP 帧放入 SDH 的虚容器。

RFC1661 仅定义了一种一般的格式封装多协议数据，具体的定帧方案是由其它协议完成的。RFC1662 提供了面向比特、面向字节的同步链路和在 8 比特组异步链路中定帧、纠错的能力。PPP 协议通常采用 HDLC 帧的封装。

POS 接口卡称为线卡（line card），它主要完成 Packet Over Sonet（POS）报文的收发并完成 PPP 协议处理。Line Card 由数据接收和数据发送两部分组成。对于数据接收部分，Line Card 从物理链路接收光信号，经过光电转换以后，对数据进行组帧、SONET/SDH 头剥离、HDLC 头剥离等处理形成 PPP 帧，最后对 PPP 帧进行处理，将处理后的 IP 报文或者做了标记的 PPP 报文送母板。对于数据发送过程，Line Card 从母板接收 PPP 报文或者 IP 报文，经过 HDLC 封装、SONET/SDH 封装、并串转换以后送光纤收发器经电光转换后发送出去。

POS Line Card 主要完成以下功能：

- 完成 Packet Over Sonet 报文的收发、PPP 协议处理；
- 支持本地 CPU 对 Line Card 的初始化、配置、错误监控和调试诊断等；
- 支持板级和系统级边界扫描的导通测试；
- 支持基于逻辑分析仪和示波器的手工调试。

在实现中，要求 Line Card 对母板透明。

6.2.2　POS Line Card 组成及数据处理过程

POS Line Card 结构框图如图 6-4 所示。POS Line Card 由三个功能块组成，它们是光纤收发器、SONET/HDLC 协议处理器和 PPP 协议处理器。

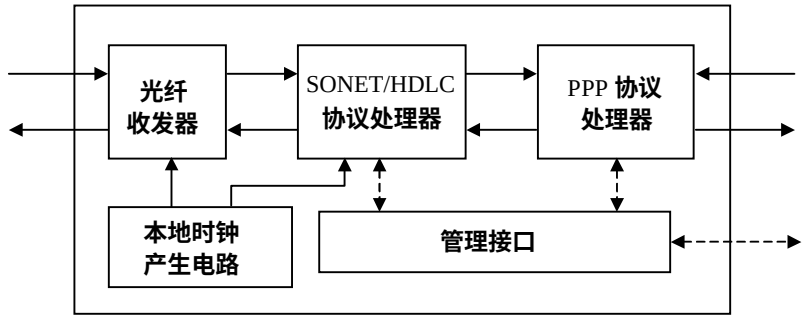


图 6-4　POS Line Card 结构框图

光纤收发器主要完成光信号接收、光电转换和电光转换、光信号的发送。SONET/HDLC

协议处理器分接收和发送两部分：接收部分对接收的串行位流进行处理，最终形成 PPP 帧；发送部分对接收的 PPP 帧进行处理，最终形成串行位流。PPP 协议处理器也分成接收和发送两个部分：接收部分对接收的 PPP 报文进行分类处理，64-bit 字对齐并添加带外信息后送母板；发送部分根据接收报文类型的不同，剥离带外信息并做出相应的处理后送 SONET/HDLC 协议处理器。

1．POS Line Card 工作过程

对于数据接收处理，Line Card 从物理链路上接收 POS 数据流，通过光电转换器将光信号转换成电信号以后，恢复出系统时钟和系统数据送 SONET/SDH 接收器。SONET/SDH 接收器将接收到的数据流进行串并转换，形成 16bit 的数据送 SONET/HDLC 协议芯片。SONET/HDLC 协议芯片接收到 16bit 的 STS 信号以后，经过帧和 TOH/SOH 定位、指针解释、TOH/SOH 和 POH 解析，剥离出 SONET/SDH-SPE/VC，然后从 SONET/SDH-SPE/VC 中剥离出 PPP 帧。Line Card 接着检查 PPP 帧的协议域，如果 PPP 帧封装的是 IP 报文，剥离 PPP 头，字对齐并加带外信息以后送母板；如果 PPP 帧封装的不是 IP 报文，将 PPP 报文做出标记并加带外信息后送母板。

数据发送处理过程和数据接收处理过程完全相反。Line Card 从母板接收数据报文，如果报文是直通报文，剥离报文带外信息后直接将报文送 SONET/HDLC 协议芯片；如果报文是 IP 报文，剥离报文带外信息并进行 PPP 帧封装以后送 SONET/HDLC 协议芯片处理。SONET/HDLC 协议芯片接收到发送报文数据以后，将 PPP 帧封装成 HDLC 帧，添加 POH 以及固定填充字节以后形成 SONET/SDH-SPE/VC，再将 SONET/SDH-SPE/VC 添加 TOH/SOH 以后形成 SONET/SDH 帧。SONET/HDLC 协议芯片将 SONET/SDH 帧以 16bit 为单位交 SONET/SDH 发送器。SONET/SDH 发送器将并行数据流经并串转换后将位流数据送光纤发送器。光纤收发器将位流数据进行电光转换后通过光纤发送出去。结构框图如图 6-5 所示。

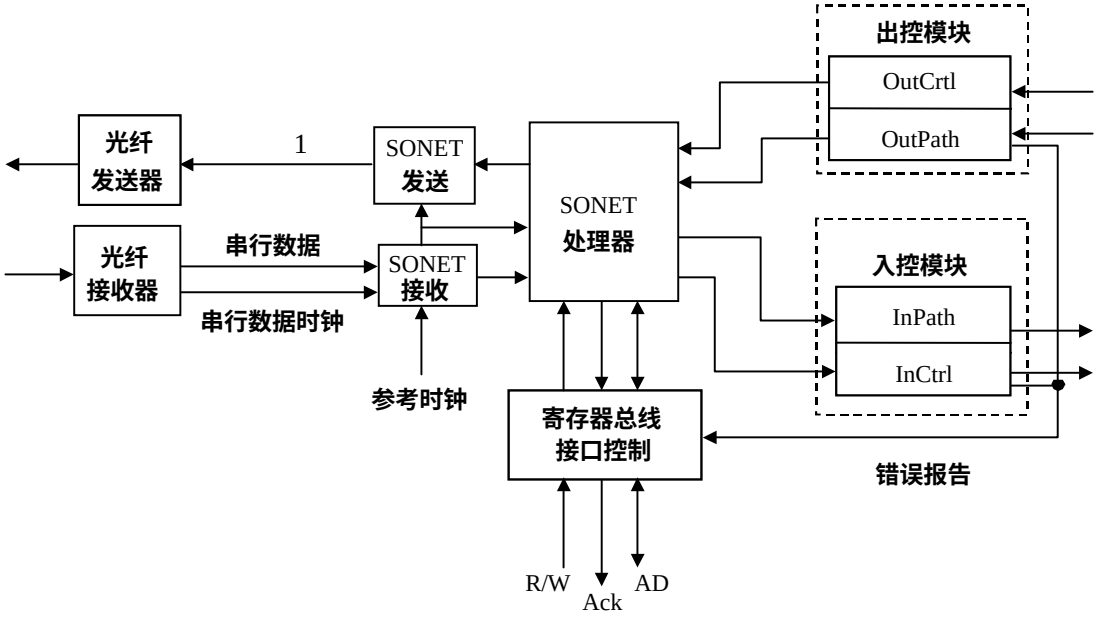


图 6-5　光纤收发器结构框图

10G、2.5G、622M 和 155M POS 线卡的区别主要在于所选择的 SONET 处理器以及接收和发送芯片的选择不同，基本的结构都是一样的。光纤发送器将电信号转换为光信号发出，而光纤接收器接收光信号并转换成电信号；SONET 接收器和发送器负责将数据转换成 SONET 帧和将 SONET 帧转换成数据；SONET 处理器负责对 SONET 帧进行协议处理；入控模块主要完成接收报文处理、字节对齐、错误检测、带外信息生成和写 FIFO 控制等；出控模块主要完成读 FIFO 控制、发送报文处理、字节对齐、错误检测和带外信息过滤等；管理接口模块主要完成双向 CPU 总线和微处理器接口之间桥接匹配，供 CPU 对内部寄存器的读写和本卡的设置、管理等。

2．链路层接收控制

接收控制部分主要完成的工作有：

- 数据接收；
- 协议类型判断；
- 字对齐，带外信息的生成；
- 数据的发送，以报文为单位，轮流写入两个接收 FIFO 中。

接收过程如下：接收控制每拍从 SONET 处理器接收数据流，对数据进行奇偶校验，若校验错，向 CPU 发中断。检测其输出的帧控制信号，根据帧控制信号的意义做出相应的处理。

如果帧控制信号指示该拍数据是 PPP 帧的第一拍，则检查 PPP 帧的协议域，如果 PPP 帧封装的是 IP 报文，则清除接收直通标志位，剥离 PPP 帧头，接收下一拍数据后进行字节边界对齐，生成带外信息（报文头标识），然后将生成了带外信息的数据送母板。如果 PPP 帧封装的不是 IP 报文，则置位接收直通标志位，生成带外信息，然后将生成了带外信息的数据送母板。接收控制工作流程如图 6-6 所示。

如果帧控制信号指示该拍数据不是 PPP 帧的第一拍，也不是帧的最后一拍，则检查接收直通标志位。如果接收直通标志位被置位，则生成带外信息（奇偶校验码和中间报文标识），然后将生成了带外信息的数据送母板。如果接收直通标志位为零，与上一拍数据剩余字节进行字边界对齐，生成带外信息，然后将生成了带外信息的数据送母板。

如果帧控制信号指示该拍数据是 PPP 帧的最后一拍，则检查接收直通标志位。如果接收直通标志位被置位，检查该拍数据中的有效字节数，如果不够 8 个字节，用填充数据补足 8 个字节，对齐字边界，然后生成带外信息（报文尾标识），最后将生成带外信息的数据送母板。如果接收直通标志位为零，检查该拍数据中的有效字节数，如果该拍有效字节数与上一拍数据剩余字节之和超过 8 个字节，则进行字边界对齐，生成带外信息（奇偶校验码和中间报文标识），然后将生成了带外信息的数据送母板。剩余的字节用填充数据补足 8 个字节，对齐字边界，生成带外信息（该拍有效字节数和报文结束标识），最后将生成带外信息的数据送母板。

3．链路层发送控制

发送控制部分主要完成的工作有：

- 数据接收；
- PPP 报文封装；
- 字对齐，帧控制信号的生成；

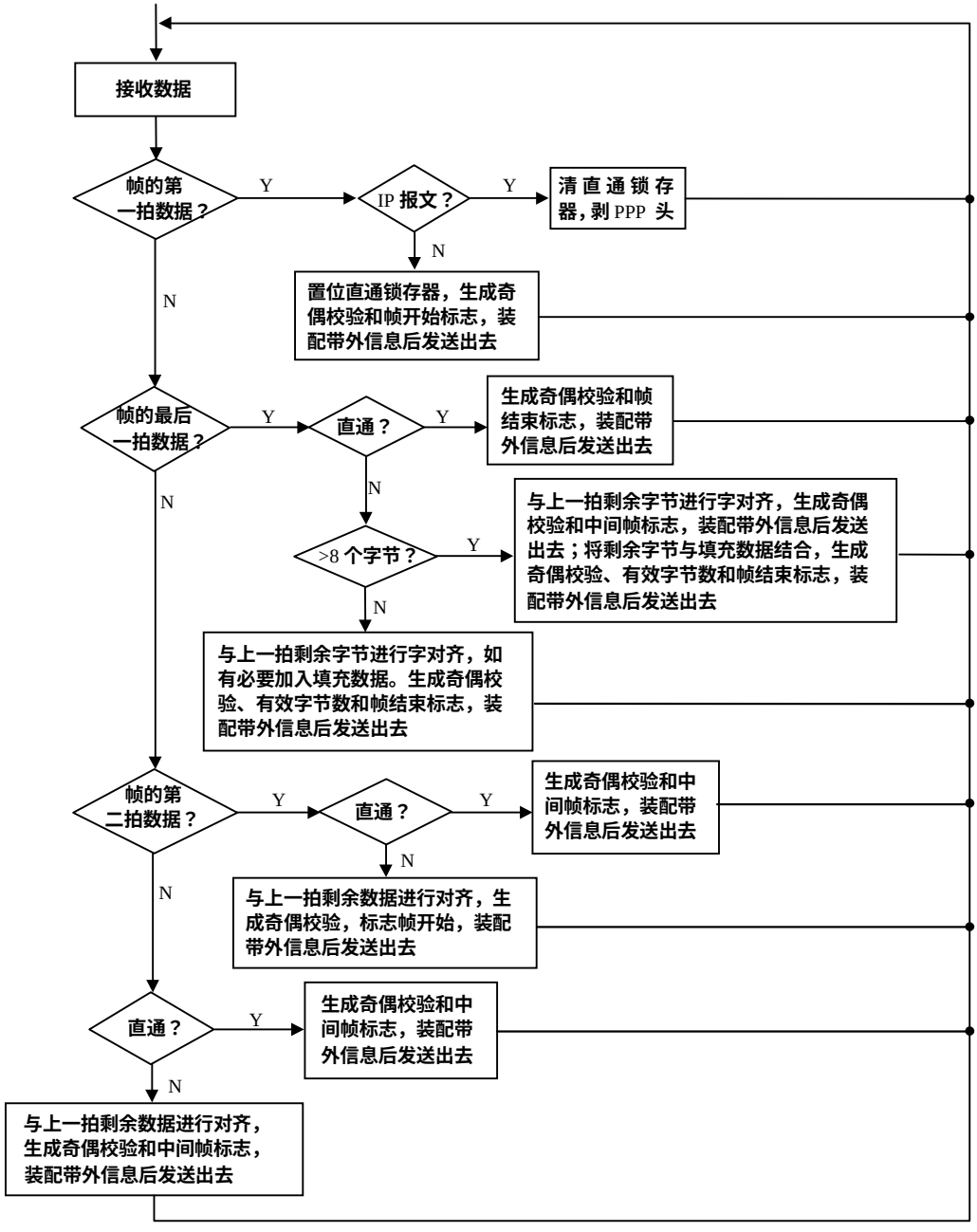


图 6-6　接收控制工作流程

■　**数据的发送。**

发送控制部分工作过程如图 6-7 **所示。**

发送控制每拍从母板接收数据，对其中的有效数据作作奇偶校验，将校验结果与带外信息中的校验位作比较，如果校验错，置位错误寄存器的“ 奇偶校验错 ”位（高有效），向 CPU **发中断。检测数据中的其他带外信息，根据带外信息的意义做出相应的处理。**

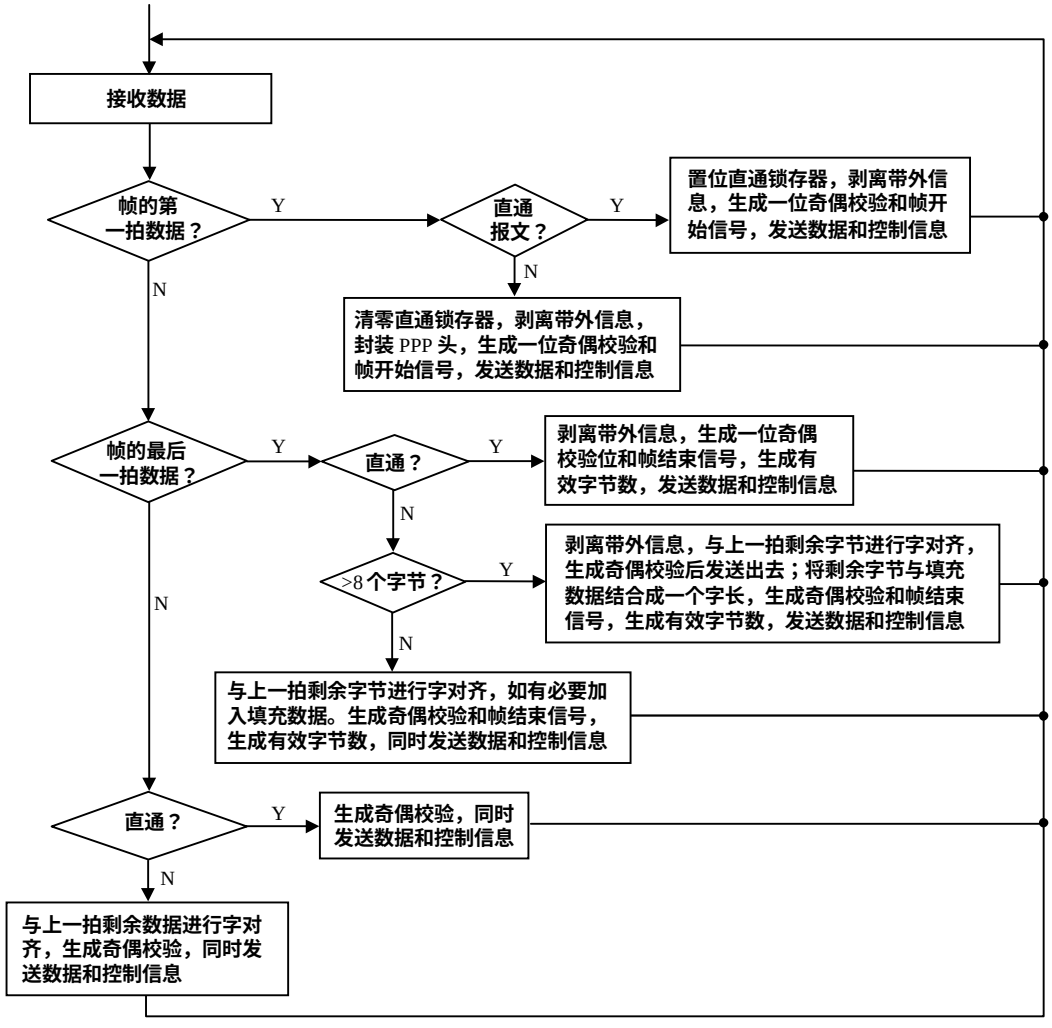


图 6-7　发送控制流程

如果带外信息指示该拍数据是接收帧的第一拍，则检查带外信息的直通位，如果带外信息的直通位为零，则清除发送直通标志位，去掉带外信息，添加 PPP 帧头，进行字节边界对齐，产生一位校验位，产生帧开始信号，然后将去掉了带外信息的数据以及帧开始信号和校验位同时送 SONET 处理器。如果带外信息的直通位非零，则置位发送直通标志位，去掉带外信息，对有效数据做一位奇偶校验，产生帧开始信号，然后将去掉了带外信息的数据以及帧开始信号和校验位送 SONET 处理器。

如果带外信息指示该拍数据不是接收帧的第一拍，也不是接收帧的最后一拍，则检查直通标志位。如果直通标志位被置位，去掉带外信息，对有效数据做一位奇偶校验，然后将去掉了带外信息的数据以及校验位送 SONET 处理器。如果带外信息的直通位为零，去掉带外信息，与前一拍剩余的字节进行字边界对齐，产生一位校验位，然后将去掉了带外信息的数据以及校验位同时送 SONET 处理器。

如果带外信息指示该拍数据是接收帧的最后一拍，则检查发送直通标志位。如果发送直

通标志位被置位，产生帧结束信号，产生一位奇偶校验位，检查该拍数据中的有效字节数，将有效字节数、帧结束信号、奇偶校验位和数据送 SONET 处理器。如果直通标志位为零，检查该拍数据中的有效字节数，如果该拍有效字节数与上一拍数据剩余字节之和超过 8 个字节，去掉带外信息，与前一拍剩余的字节进行字边界对齐，产生一位校验位，然后将去掉了带外信息的数据和校验位送 SONET 处理器；锁存最后剩余的有效字节数，将最后剩余的字节用填充数据补足 8 个字节，产生帧结束信号，产生一位奇偶校验位，将有效字节数、帧结束信号、奇偶校验位和数据送 SONET 处理器。如果该拍有效字节数与上一拍数据剩余字节数之和不超过 8 个字节，锁存有效字节数，去掉带外信息，与前一拍剩余的字节进行字边界对齐，将用填充数据补足 8 个字节，产生帧结束信号，产生一位奇偶校验位，将有效字节数、帧结束信号、奇偶校验位和数据送 SONET 处理器。

4．管理接口控制器

管理接口部分主要功能：

- 微处理器接口控制；
- GPIO 接口控制；
- 中断控制；
- 板级寄存器控制。

管理接口控制器与 SONET 处理器的接口主要包括微处理器接口、GPIO 接口、TOH/SOH 接口以及测试接口。微处理器接口主要完成对 SONET 处理器的寄存器和板级寄存器的操作。GPIO 接口控制主要完成对 GPIO/LED/Overwrite 接口的操作控制。TOH/SOH 接口主要是完成对 TOH/SOH 寄存器的读写，由微处理器接口完成读写。测试接口完成 SONET 处理器的边界扫描操作。

中断控制主要完成 SONET 处理器产生的中断、奇偶校验错中断等的控制。与 SONET 处理器的接口包括在微处理器接口中。

板级寄存器主要有：

- 线卡版本号寄存器：线卡版本号。
- 线卡类型寄存器：线卡类型（2.5G 卡、1000M 卡、100/10M 卡）。
- 协议协商寄存器：存储相邻路由器封装 PPP 报文的协商结果。
- 错误寄存器：最低位为发送奇偶校验错误位，当发送数据奇偶校验错时置位该寄存器（高有效），且在错误屏蔽寄存器的该位没有被置位时向 CPU 发中断。次低位为接收奇偶校验错误位，当接收数据奇偶校验错时置位该寄存器（高有效），且在错误屏蔽寄存器的该位没有被置位时向 CPU 发中断。第三位为 SONET 处理器中断位，当 SONET 处理器发生中断时，置位寄存器的次低位且在错误屏蔽寄存器的该位没有被置位时向 CPU 发中断。
- 错误屏蔽寄存器：最低位为发送奇偶校验错屏蔽位，当置位该位时，即使错误寄存器的该位被置位也不向母板发出中断；次低位为接收奇偶校验错屏蔽位，当置位该位时，即使错误寄存器的该位被置位也不向母板发出中断；第三位为 SONET 处理器中断屏蔽位，当置位该位时，即使错误寄存器的该位被置位也不向母板发出中断。

总之，对于路由器的线卡设计，设计方案务必要详尽，对于所选用的芯片的型号、寄存器的设置、各部分的数据宽度、缓冲队列的设计等都要有明确的设计。特别是对每个模块的工作流程以及状态机都要预先计划好，不然很可能造成设计中的漏洞而导致返工。

第 7 章　路由协议的设计与实现

路由器工作离不开路由表，路由器通过查找路由表得知去往目的地的数据包下一跳送往何处，路由协议就是用来维护路由表的一种协商机制。在网络规模还不是很大，并且网络变化很少的情况下，使用手工配置静态路由显得很有效。管理者只需保留一张关于网络的表格，并在新的网络加入或删除一个网络时，更新该表格。从此路由器出发，去往每一个目的网段的数据包下一跳送往何处由工作人员来决定，并且配置进路由表中。随着互联网规模的不断扩大，人工改变路由的方式耗时而且容易带来错误，人工维护整个网络的路由表已不能满足网络对于路由表的需要。网络运营需要路由表能真实快速地反映整个网络的情况。当网络发生变化时，路由表要快速得到调整使数据包尽可能地走最佳网络路径到达目的地。

路由协议动态维护路由器中的路由表。它通过收集和发送报文获得网络上的每个拓扑结构的变动，并根据这些变动迅速调整自己的路由表，使得路由器在转发数据包时选择最佳的路径。

本章着重介绍 IPv4 路由器上必须实现的几个路由协议，包括 RIP 协议、OSPF 协议和 BGP-4 协议。

7.1　路 由 结 构

Internet 是由自治系统（Autonomous System，AS）集合构成的，每个自治系统包括了处于一个机构管理之下的若干网络和路由器。BGP（Border Gateway Protocol，边界网关协议），就是为 TCP/IP 网络设计的用于自治系统之间的路由协议。该协议的基本功能是与其它 BGP 自治系统交换网络可达性信息（Network Layer Reachable Information，NLRI），这种可达性信息包含了通往目标所要穿越的自治系统记录，利用这些信息，系统就可以构建一个无环的自治系统连接图，并把形成的外部路由信息重发布给内部网关协议 IGP。路由结构示意如图 7-1 所示。

BGP 可以说在一定程度上综合了距离向量和链路状态算法的优点，是一种路径向量协议。被称为路径向量协议的原因在于 BGP 路由信息中包含着 AS 号的一个序列，这个序列指明了路由经过的路径。利用这个信息可建立起各 A S 的连接图，从而避免路由循环。

我们已经知道在单个 AS 中使用内部网关路由协议（RIP，OSPF 等）执行路由功能。BGP 的引入使实现自治系统间无环路由信息交换更容易，并且能够通过无类域间路由选择（CIDR）来控制路由表的扩展。设计 BGP 也是为了通过使用自治系统来提供 Internet 的一个结构化的清晰视图。

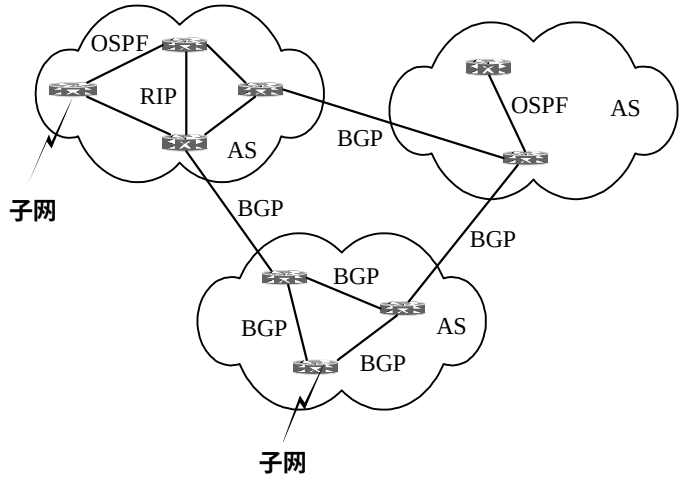


图 7-1　路由结构示意图

从某种意义上说，Internet 可成为一个大的 OSPF 网络。参与 Internet 的所有组织将会不得不采用同样的管理策略。而通过把 Internet 划分成多个自治系统，我们就能创建一个由许多更小的更易于管理的小网络组成的大的网络。在这些被称为自治系统的更小的网络内部，可以采用各组织自己的规则和管理策略。每个 AS 都由 Internet 注册机构分配一个唯一的数来标识。

IETF 的 Inter-Domain Routing 工作组分别在 1989 年公布了 BGP 协议的版本一（BGP-1），1990 年公布了版本二（BGP-2），1993 年公布了版本三（BGP-3），目前最新版本为 1995 年制订的 BGP-4，BGP 在不断发展的过程中逐渐成为 Internet 路由体系结构的基础。我们主要是以 BGP-4 为主，阐述 BGP 的原理、关键特征、实现及其在 Internet 中的应用。

路由协议根据使用的范围分为内部路由协议和外部路由协议。在 20 世纪 80 年代早期，Internet 还仅限于初始的 Arpanet、其卫星网 Satnet 和一系列通过网关直接连接到这些网络的局域网。由于网络不断增长，需要采用一种层次结构。因此，Internet 被分割成了一系列的“自治系统”。一个自治系统实际上是在同一个机构下管理的一系列路由器和网络。在自治系统内部运行的路由协议被称为内部路由协议，而在自治系统之间运行的路由协议被称为外部路由协议。

内部路由协议又根据使用的原理和算法不同而进行划分，例如 RIP、OSPF、IGRP，外部路由协议也有 EGP、BGP 等。

路由协议的稳定实施是成功管理一个网络的重要因素。稳定和高性能是内部网关协议（IGP）在服务提供网络内管理业务流量的基本性能。外部网关协议（EGP）的牢固性和可伸缩性对于在不同商网络之间的链接和控制是非常重要的。一些网络专家们在评价路由协议实施的可靠性时，将互操作性作为一个单独的因素。他们很关心软件是否符合 IETF 标准和软件在多个厂家产品环境中的操作情况。互操作性是每个厂商都应满足的一个重要元素，但它只应作为诸多需要仔细衡量的部分中的一个因素。还有其它一些关键的元素隐藏在软件之下，并不会被评价者立即发现。但是，正是这些隐藏的因素在决定路由协议在 Internet 中执行的能力时起重要的作用。

区别路由协议的关键要素是稳定性和可扩展性。稳定性和可扩展性并不因故障而发生；它们必须在项目一开始就被设计到软件体系当中去。

稳定性主要关心在大型网络中承受运行压力和连续长时间无故障工作的能力。对于任何路由协议的实施，都有许多设计要点在决定系统稳定性的过程中起重要作用：

- 软件工程师在预测和编写对不同类型故障进行响应的代码时的预见，其包括协议错误，如畸形包，意外的对等关系，如传输过多的请求/更新，和在网络感到压力时 CPU 资源被耗尽。
- 开发者提供正确调节器的技巧，以使路由器适应多种不同情况。
- 软件工程师在编写代码时自发地遵循爱因斯坦的格言“使事情尽可能地简单，但不过于简单”。这样可产生一个易于理解的，快速的和稳定的代码库。

扩展性主要关心网络实现与不断扩张的网络环境同步成长的能力。有许多因素在决定路由协议实现过程中的可伸缩性起着重要作用：

- 支持的最大端口数。
- 路由表查询的速度。
- 路由表中可存储的最大路由数。
- 每个路由器可支持的最大 OSPF 或 IS-IS 邻接数或 BGP 对等体数。
- 路由器链接状态表中可存储的最大 OSPF 的 LSA 数或 IS-IS 的 LSP 数。
- 允许网络管理员方便有效地控制输入、输出和修改大量路由信息用的策略控制语言的能力。

7.2　RIP 协议设计

RIP（Routing Information Protocol，路由信息协议）是应用较早、使用较普遍的内部网关协议，适用于小型同类网络，是典型的距离向量（distance-vector）协议。RIP 在两个文档中正式定义：RFC 1058 和 1723。RFC 1058（1988）描述了 RIP 的第一版实现。RFC 1723（1994）定义了 RIP v2，像 RIP 版本 1 一样，仍是基于经典的距离向量路由算法的。RIPv2 中加入了一些现在的大型网络中所要求的特性，如认证、路由汇总、无类域间路由，和变长子网掩码（VLSM）。这些高级特性都不被 RIP 版本 1 支持。

RIP 的主要特点归纳如下：

- 它是一个距离向量路由协议。
- 路由选择的度量值是跳步数。
- 最大允许的跳步数是 16。
- 缺省情况下每隔 30 秒钟广播一次路由更新。
- 它有在多条路径上进行负载平衡的功能。

7.2.1　RIP 协议简介

RIP 是一个用于路由器和主机间交换路由信息的距离向量协议，它基于 Bellham-Ford 算法（距离向量），此算法 1969 年被用于计算机路由选择。Xerox 首先于 1970 年开发出今天为大家所熟知的 RIP 协议，作为 Xerox Networking Services XNS 协议族的一部分。

尽管 RIP 协议有技术限制（后面还会讲到），RIP 还是一种被广泛应用于同构网络的内部网关协议。其广泛使用源于广泛使用的 4BSD UNIX 上的 Berkeley 分布路由软件。路由软

件用 RIP 给本地网络上的机器提供路径选择和可达信息。TCP/IP 最早用 RIP 提供局域网的路由信息，最后用 RIP 提供广域网的路由信息。

RIP 使用广播用户数据报协议（User Datagram Protocol ,UDP）数据报文的方式把路由表项发送到相邻路由器。因为 RIP 使用 UDP 作为其发送机制，所以发送到相邻路由器的路由表更新不能得到保证。路由器间 RIP 表项的发送缺省是在路由器初始启动后 30 秒。当一个路由器在到另一个已经活动的路由器的连接上变成活动时，这种路由器的“公布”也会出现在路由器之间。

使用 RIP 的路由器期待在 180 秒之内从邻接路由器获得更新。如果在这段时间内没有收到邻接路由器的路由表更新，则去往未更新路由器的网络路由被标识为不可用，强制把 ICMP 网络不可到达消息返回给通过未更新路由器而连接的资源请求者。一旦接收更新定时器到达 240 秒，未更新路由器的路由表项将被从路由表中移去。路由器现在接收到的要到达通过未更新路由器连接的报文，可以被重定向到此路由器的缺省网络路径上。“缺省路由”可以通过 RIP 学习或是用缺省 RIP 度量定义。目的网络在路由表中没有找到的报文被重定向到定义缺省路由的接口上。

1．度量值和管理距离

RIP 使用的衡量不同路由的价值的度量值是“跳步数”。跳步数是一条路由要经过的路由器的数目。一个直接相连的网络的跳步数是零，不可达网络的跳步数是 16。这非常有限的度量值使得 RIP 不能作为一个大型网络的路由协议。如果一个路由器有一个缺省的网络路径，那么 RIP 就通告一条路由，它链接着路由器和一个虚网络 0.0.0.0。网络 0.0.0.0 并不存在，但是 RIP 把 0.0.0.0 当作是一个实现缺省路由的网络。

所有直接连接接口的跳数为零，考虑图 7-2 所示的路由器和网络。虚线是广播 RIP 报文的方向，路由器 A 通过发送广播到 N1，通告它与 N2 之间的跳数是 1（发送给 N1 的广播中通告它与 N1 之间的路由是无用的）。同时也通过发送广播给 N2，通告它与 N1 之间的跳数为 1。同样，R2 通告它与 N2 的度量为 1，与 N3 的度量为 1。如果相邻路由器通告它与其他网络路由的跳数为 1，那么我们与那个网络的度量就是 2，这是因为为了发送报文到该网络，我们必须经过那个路由器。在我们的例子中，R2 到 N1 的度量是 2，与 R1 到 N3 的度量一样。

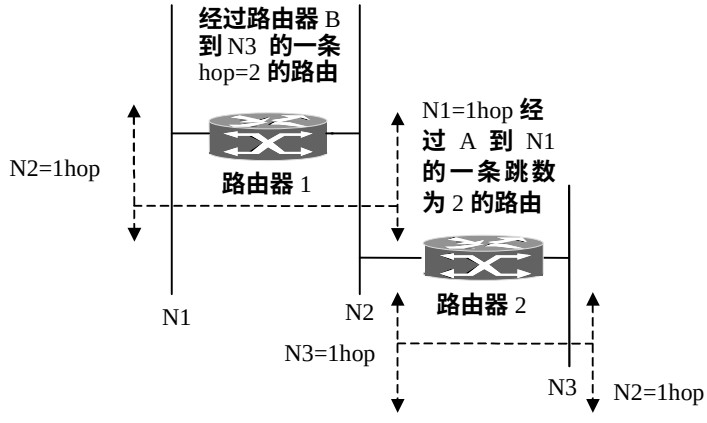


图 7-2　度量值和管理距离

由于每个路由器都发送其路由表给邻站，因此，可以判断在同一个自治系统 A S 内到每个网络的路由。如果在该 AS 内从一个路由器到一个网络有多条路由，那么路由器将选择跳数最小的路由，而忽略其他路由。跳数的最大值是 15，这意味着 RIP 只能用在主机间最大跳数值为 15 的 AS 内。度量为 16 表示无路由到达该 IP 地址。

2．定时器

RIP 协议定义了四种定时器：更新定时器、无效定时器、保持定时器和刷新定时器。

更新定时器（以秒为单位）设置路由器发送更新信息的速度，默认值是 30 秒。

无效定时器（以秒为单位）设置路径被认为无效的时间间隔，指明经过多少秒钟一条路由将被认为是无效的。如果某条路径在常规更新信息中不出现，就启动该定时器，默认值是 180 秒。它至少是更新定时器的值的三倍。当没有更新报文刷新路由时，这条路由将成为无效路由。这时，路由进入一种保持状态，即路由被标记并通告为不可达。但是路由器仍然可以使用这条路由发送报文。

保持定时器（以秒为单位）设置拒绝好的路由信息的间隔时间，指明在多少秒之内将搁置提供更好的路由的信息。它的思想是保证每个路由器都收到了路径不可达信息，而且没有路由器发出无效路径信息。默认值是 180 秒。它应该至少是更新定时器的值的三倍。当有一个更新报文到达，指明一个路由不可达时，这条路由就进入保持状态，并被标记、通告为不可达。但是，这条路由仍然可以用来发送报文。当保持计时器超时，如果有别的路由器更新报文到达，这条路由可能就不再是不可达的了。

刷新定时器（以秒为单位）设置路径从路由表中删除必须等待的时间。默认值是 240 秒。它的值应该比无效定时器的值大。如果它的值更小，那么正常的保持定时器间隔时间就不能保证，就会造成在保持定时器间隔时间之内接受一条新的路由。

3．协议流程

RIP 使用的 UDP 端口号是 520。一旦在某个接口上配置好 RIP 协议并设置相应的版本，协议就按如下流程工作：

■　初始化：在启动一个路由守护程序时，它先判断启动了哪些接口，并在每个接口上发送一个请求报文，要求其他路由器发送完整路由表。在点对点链路中，该请求是发送给其他终点的。如果网络支持广播的话，这种请求是以广播形式发送的。目的 U D P 端口号是 520（这是其他路由器的路由守护程序端口号）。这种请求报文的命令字段为 1，但地址系列字段设置为 0，而度量字段设置为 1 6。这是一种要求另一端完整路由表的特殊请求报文。

■　接收到请求。如果这个请求是刚才提到的特殊请求，那么路由器就将完整的路由表发送给请求者。否则，就处理请求中的每一个表项：如果有连接到指明地址的路由，则将度量设置成路由表中该路由的度量值，否则将度量置为 16（度量为 16 是一种称为“无穷大”的特殊值，它意味着没有到达目的的路由）。然后发回响应。

■　接收到响应。使响应生效，可能会更新路由表。可能会增加新表项，对已有的表项进行修改，或是将已有表项删除。

■　定期选路更新。每过 30 秒，所有或部分路由器会将其完整路由表发送给相邻路由器。发送路由表可以是广播形式的（如在以太网上），或是发送给点对点链路的其他终点的。

■　触发更新。每当一条路由的度量发生变化时，就对它进行更新。不需要发送完整路由表，而只需要发送那些发生变化的表项。每条路由都有与之相关的定时器。如果运行 RIP

的系统发现一条路由在 3 分钟内未更新，就将该路由的度量设置成无穷大（16），并标注为删除。这意味着已经在 6 个 30 秒更新时间里没收到通告该路由的路由器的更新了。再过 60 秒，将从本地路由表中删除该路由，以保证该路由的失效已被传播开。

4．存在的问题

这种方法看起来很简单，但它有一些缺陷。首先，RIP 没有子网地址的概念。例如，如果标准的 B 类地址中 16 bit 的主机号不为 0，那么 RIP 无法区分非零部分是一个子网号，或者是一个主机地址。有一些实现中通过接收到的 RIP 信息，来使用接口的网络掩码，而这有可能出错。

其次，在路由器或链路发生故障后，需要很长的一段时间才能稳定下来。这段时间通常需要几分钟。在这段路由建立的时间里，可能会发生路由环路。在实现 RIP 时，必须采用很多微妙的措施来防止路由环路的出现，并使其尽快建立。RFC 1058 [Hedrick 1988a]中指出了很多实现 RIP 的细节。采用跳数作为路由度量忽略了其他一些应该考虑的因素。同时，度量最大值为 15，则限制了可以使用 RIP 的网络的大小。

RIP 同任何距离向量路由选择协议一样，都有一个缺陷：路由器不知道网络的全局情况。路由器必须依靠相邻路由器来获取网络的可达信息。由于路由选择更新信息在网络上传播慢，距离向量路由选择算法有一个慢收敛问题，这个问题将导致不一致性产生。RIP 使用以下机制减少因网络上的不一致带来的路由选择环路的可能性：计数到无穷大、水平分割、破坏逆转更新、保持计数器和触发更新。

（1）计数到无穷大问题

RIP 允许最大跳数为 15。大于 15 的目的地被认为是不可达。这个数字限制了网络大小的同时也防止了一个叫做计数到无穷大的问题，如图 7-3 所示。

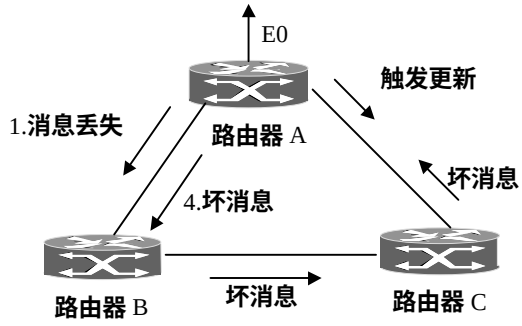


图 7-3　记数到无穷大

（1）路由器 A 的以太网接口 E0 变为不可用后，产生一个触发更新送往路由器 B 和路由器 C。这个更新信息告诉路由器 B 和路由器 C，路由器 A 不能够到达 E0 所连的网络，假设为网络 A。这个更新信息传输到路由器 B 被推迟了（CPU 忙、链路拥塞等），但到达了路由器 C。路由器 C 从路由表中去掉到该网络的路径。

（2）路由器 B 仍未收到路由器 A 的触发更新信息，并发出它的常规路由选择更新信息，通告网络 A 以 2 跳的距离可达。路由器 C 收到这个更新信息，认为出现了一条新路径到网络 A。

（3）路由器 C 告诉路由器 A 它能以 3 跳的距离到达网络 A。

（4）路由器 A 告诉路由器 B 它能以 4 跳的距离到达网络 A。

（5）这个循环将进行到跳数为无穷，在 RIP 中定义为 1 6。一旦一个路径跳数达到无穷，它将声明这条路径不可用并从路由表中删除此路径。

由于计数到无穷大问题，路由选择信息将从一个路由器传到另一个路由器，每次段数加一。路由选择环路问题将无限制地进行下去，除非达到某个限制。这个限制就是 RIP 的最大跳数。当路径的跳数超过 15，这条路径就从路由表中删除。

（2）水平分割

水平分割规则如下：路由器不向收到某条路由到来的方向回传此路由。当打开路由器接口后，路由器记录路由是从哪个接口来的，并且不向此接口回传此路由。路由器应该可以对每个接口关闭水平分割功能，因为水平分割在非广播多路访问 hub-and-spoke 环境下（NBMA，non broadcast mutilple access）有不利影响。在图 7-4 中，路由器 A 通过一个中间网络连接路由器 B、C 和 D。如果在路由器 A 启用水平分割，路由器 C、D 将收不到路由器 A 通过接口 E0 从路由器 B 来的路由信息。

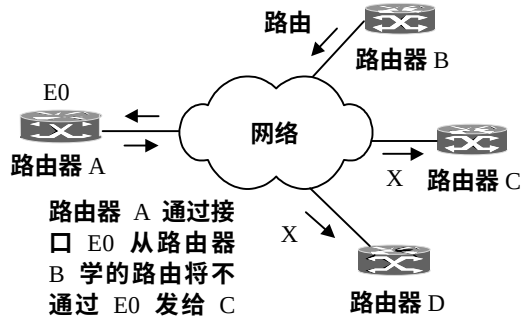


图 7-4　水平分割

（3）毒性逆转

水平分割是路由器用来防止把一个接口得来的路由又从此接口传回导致的问题的方案。水平分割方案忽略在更新过程中从一个路由器获取的路由又传回该路由器。有毒性逆转的水平分割的更新信息中包括这些路径，但这个处理过程把这些路径的度量设为 16(无穷)。通过把跳数设为无穷并把这条路径告诉源路由器，有可能立刻解决路由选择环路。否则，不正确的路径将在路由表中驻留到超时为止。毒性逆转的缺点是它增加了路由更新的的数据大小。

在图 7-5 中，路由器 A 通过一个中间网络连接路由器 B、C 和 D。如果在路由器 A 启用

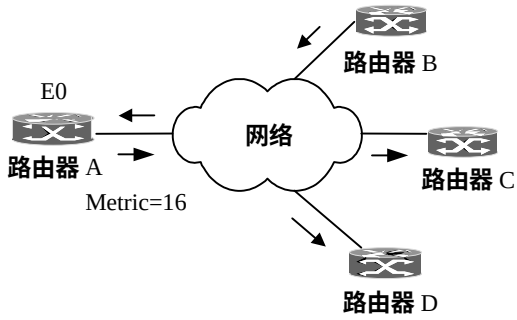


图 7-5　毒性逆转

毒性逆转，路由器 C、D 会收到路由器 A 通过接口 E0 从路由器 B 学来的路由信息，但是这些路由信息的度量值为 16。

（4）保持定时器

保持定时器防止路由器在路径从路由表中删除后一定的时间内接受新的路由信息。它的思想是保证每个路由器都收到了路径不可达信息，而且没有路由器发出无效路径信息。仍然在图 7-3 中，由于路由更新信息被延迟，路由器 B 向路由器 C 发出错误信息。使用保持计数器这种情况将不会发生，因为路由器 C 将在 180 秒内不接受通向网络 A 的新的路径信息。到那时路由器 B 将存储正确的路由信息。

（5）触发更新

有破坏逆转的水平分割将任何两个路由器构成的环路打破。三个或更多个路由器构成的环路仍会发生，直到无穷（16）时为止。触发式更新想加速收敛时间。当某个路径的度量改变了，路由器立即发出更新信息，路由器不管是否到达常规信息更新时间都发出更新信息。

7.2.2 RIP 协议的实现

1．报文格式

RIP 使用两种报文传输路由信息：更新报文和请求报文。它们的报文格式一样，如图 7-6 所示。

命令（1-5）	版本	必须为 0
路由项 1 的地址族		必须为 0
路由项 1 的 IP 地址		
必须为 0		
必须为 0		
路由项 1 的距离		
路由项 2 的地址族		必须为 0
路由项 2 的 IP 地址		
必须为 0		
必须为 0		
路由项 2 的距离		
……		

图 7-6 RIP 报文格式

命令域表示该报文是请求报文还是响应报文。请求报文要求路由器发送其路由表的全部或部分。响应报文可以是主动提供的周期性路由更新或对请求的响应。大的路由表可以使用多个 RIP 报文来传递信息。版本号域指明使用的 RIP 版本，此域可以通知不同版本的不兼容。地址族标志（AFI）指明使用的地址族。RIP 设计用于携带多种不同协议的路由信息。每个项都有地址族标志来表明使用的地址类型，IP 的 AFI 是 2。地址域指明该项的 IP 地址。度量域表示到目的的过程中经过了多少跳数（路由器数）。有效路径的值在 1 和 15 之间，16 表示不可达路径。

报文格式中，“0”域是为了向后兼容旧的 RIP 协议，0 域说明现在的 RIP 不支持旧版本所有的私有特性，比如：traceon 和 traceoff 机制，已经被 RFC 1058 抛弃，但为了向后兼容，RFC 1058 在 RIP 报文中为其保留了空间，但却要求这些空间必须为 0。当收到的报文中这些域不是 0 时，就会被简单地抛弃。

在一个 IP RIP 报文中最多可有 25 个 AFI、地址和 metric 域，即一个 RIP 报文中最多可含有 25 个地址项。

RIP v2 的报文格式如图 7-7 所示，RIP v2 规范（RFC1723）允许 RIP 报文包含更多的信息，并提供了简单的认证机制。

命令	版本	未使用
地址族标识符		路由标签
IP 地址		
子网掩码		
下一跳		
度量值		

图 7-7　RIP v2 报文格式

命令域表示该报文是请求报文还是响应报文。请求报文要求路由器发送其路由表的全部或部分。响应报文可以是主动提供的周期性路由更新或对请求的响应。大的路由表可以使用多个 RIP 报文来传递信息。版本域指明使用的 RIP 版本，在实现 RIP v2 或进行认证的 RIP 报文中，此值为 2。地址族标志（AFI）指明使用的地址族。RIP 设计用于携带多种不同协议的路由信息。每个项都有地址族标志来表明使用的地址类型，IP 的 AFI 是 2。如果第一项的 AFI 为 0xFFFF，该项剩下的部分就是认证信息。带有认证信息的 RIP 报文格式如图 7-8 所示。目前，唯一的认证类型就是简单的口令。路由标记提供区分内部路由（由 RIP 学得）和外部路由（由其它协议学得）的方法。IP 地址指明该项的 IP 地址。子网掩码包含该项的子网掩码，如果此域为 0，则该项不指定子网掩码。下一跳指明下一跳的 IP 地址。度量值表示到目的地址过程中经过了多少跳数（路由器数），有效值在 1 和 15 之间，16 表示不可达路径。

命令	版本	未使用
0xFFFF		验证类型
验证信息		

图 7-8　带有认证信息的 RIP 报文格式

2．距离向量算法

V-D 算法是距离向量算法的缩写。它是基于 R.E.Bellman 的最短路径算法及 Ford 和 Fulkerson 最先提出的分布式算法的。V-D 算法的描述如下：

- 设 *N* 为节点数，*M* 为链路数。
- 设 L 为一个大小为 *M* 的链路表，其中 L[l].m 为链路的量度值，L[l].s 为链路的源站点，L[l].d 为链路的目的站点。
- 设 *D* 为一个[*N*,*N*]的矩阵，其中 *D*[*i*,*j*]为从 i 到 j 的距离。
- 设 *H* 为一个[*N*,*N*]的矩阵，其中 *H*[*i*,*j*]为报文从 i 到 j 路由的距离。

算法如下：

（1）初始化所有 $D[i,j]$,若 $i=j$,则 $D[i,j]=0$,否则,$D[i,j]$为无穷大。将所有的 $H[i,j]$都初始化为-1。

（2）对于所有链路 l，和所有目的站点 k：令 i=L[l].s, j=L[l].d, 计算出 $d=L[l].m+D[j,k]$。

（3）若 d小于 $D[i,k]$, 则更新 $D[i,k]=d$; $H[i,k]=l$。

（4）若至少有一个 $D[i,k]$被更新，则重复步骤（2）。否则，计算结束。

在上面网络的情况中，我们看到，在计数到无穷大的过程中，网络的中间状态极度混乱，大量报文循环发送，链路拥塞，并且广播报文本身也会由于拥塞而丢失。但由于节点广播其距离向量表是一个随机过程，因此这种慢收敛是不可避免的，且有可能随时发生。

虽然现在有很多种针对这一问题的补救措施，如分割范围、使用信息源抑制法、毒性逆转法及触发更新法等。但不幸的是，这些技术能解决一些问题，却又带来了一些新的问题。例如，在许多路由器共享一个公共网络的结构中采用触发更新技术的情况下，一个广播就能改变这些路由器的选路表，引发新一轮的广播。如果第二轮广播改变了路由表，它又会引发起更多的广播，这就产生了雪崩。使用广播技术和使用抑制技术防止慢收敛问题，可使 V-D 算法的应用工作效率极低。广播要耗费大量宝贵的带宽资源。即使不出现广播雪崩现象，所有机器周期性地进行广播意味着网络流量随着路由器数目的增加而增加。

3．RIP v2

尽管 RIP v2 使用了和 RIP v1 相同的基本算法，它还是有几个新的特征，包括外部路由标记、子网掩码、多个下一跳地址、以及认证。较重要的改变是从 RFC 1388 中去掉了域（domain）字段，对这个字段如何使用由于没有公认的方法，所以决定保留这个字段以为将来扩展。

（1）外部路由标记

路由标记有可能用来传播从 EGP 中获取的信息。关于这个域内容的解说超出了本节范畴。无论如何，它有可能被用到，例如，传播 EGP AS 号。

（2）子网掩码

内含子网掩码是进一步开放 RIP 协议的本意。子网掩码信息可以使得 RIP 在多样的环境下更有用，允许在网络上使用可变的网络掩码。子网掩码是实现由 CIDR 所提出的无类路由所必须的。

（3）多个下一跳地址

在使用多个路由协议的环境中，支持下一跳地址使 RIP 能够最优化路由。例如，连同其他 IGP 一起在一个网络上使用 RIPv2，一个路由器运行两种协议，那么路由器可向其他进行 RIP v2 的路由器指出，到给定目的地址的下一跳，有比它自己有更优的路由。

（4）认证机制

从 RIP v1 到 RIP v2，一个很重要的改进是增加了认证机制。本来，它同是由 OSPF 提出的扩展机制。现在，一般使用文本口令用于认证。更加复杂的认证机制也可以很容易地引入。

（5）组播 Multicasting

RIP v2 使用组播地址发布路由，而不是广播。IP 组播地址的使用可能减轻不运行路由协议主机的负担，它也允许 RIP v2 使用一些信息而不让 RIP v1 获取。这可以防止它曲解 RIP-2

的路由信息，因为 RIP-1 不支持子网掩码。

7.3　OSPF 协议设计

Internet 最早使用 RIP 动态路由协议。RIP 协议适合小型网络系统。但是在网络数目增多时存在一些问题。并且 RIP 协议还存在无限计数和收敛速度慢等不足。后来开始在 Internet 使用链路状态协议。所谓链路是指路由器接口，所以 OSPF 也可以称为接口状态路由协议。状态是指接口的参数，包括接口是开启还是关闭、接口的 IP 地址、子网掩码、接口所连的网络，以及使用路由器的网络连接的相关代价。

7.3.1　OSPF 协议简介

1988 年 IETF 开始制定新的链路状态协议，即 OSPF 协议，并且在 1990 年形成标准。OSPF 是动态路由协议，能快速检测到网络拓扑结构变化，并很快计算新的路由表。现在 OSPF 是 Internet 上的主要内部网关协议。所谓内部网关协议（IGP，Interior Gateway Protocol），即在一个自治系统（AS，Autonomous System）内部运行的协议。自治系统是指在一个管理机构下运行的网络，其内部可以使用自己的路由协议。AS 之间则使用外部网关协议（EGP）来交换路由信息。

OSPF 把一个 AS 分成多个区域（AREA），每个区域内包括一组网络。OSPF 区域如图 7-9 所示。一个区域内的路由器之间交换所有的信息（链路状态），而对于同一个 AS 内的其他区域内的路由器隐藏它的详细拓扑结构。这种分级结构可以减少路由信息的流量，并且简化路由器计算。每个运行 OSPF 的路由器维护其所在的 AS 的链路状态数据库。通过这个数据库，使用 Dijkstra 的最短路径算法，每个路由器可以构造一个以其自己为根的到该 AS 内部各个网络的最短路径树。路由器之间使用 OSPF 协议相互交流这些拓扑信息。

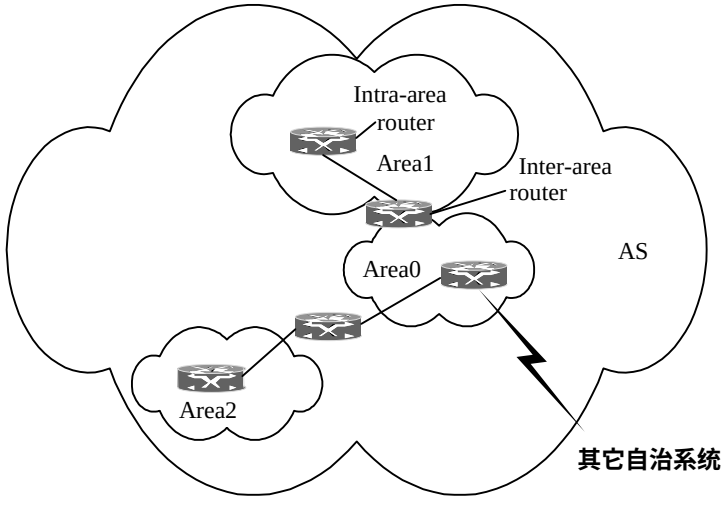


图 7-9　OSPF 区域

OSPF 使用 IP 协议中的服务类型（TOS）参数控制传输报文的服务质量。对于不同服务类型的报文来说，它们具有不同的服务质量参数，如延迟，带宽以及丢失率等。OSPF 根据不同的报文类型计算不同的路径。当两点间有多条成本相同的路径时，OSPF 在路由表中保留并轮流使用这些路径，提高网络带宽的利用率。OSPF 支持各种灵活的 IP 子网配置方式，由 OSPF 传播的路由都有目的和掩码两部分，所以同一个网络内的不同子网可以有不同长度的掩码，即变长子网掩码（VLSM）。数据包被路由到最长前缀匹配之处。主机被认为是全部匹配的子网。

为了确保路由器信息可靠地进行交换，OSPF 路由器间要相互认证（Authentication），只有可信的路由器才能加入到路由信息的交换中来。OSPF 可以采用多种认证方式，不同区域内的认证方式可以不同。这样，一些区域可能采用更加严格的认证手段。OSPF 还可以有效地使用从 EGP 得到的路由信息，在本 AS 内传播。

表 7-1 列出了 OSPF 协议的一些主要术语。

表 7-1　　OSPF 主要概念

名　称	含　义
Router ID	用于标志每个路由器的 32 位数。
Interface（接口）	路由器和具有唯一 IP 地址和子网掩码的网络之间的连接，也称为(Link)。
Neighboring Routers 相邻路由器	带有到公共网络的接口的路由器，由 Hello 协议动态发现并维护。
Adjacency 毗邻	仅毗邻的路由器之间才能直接交换链路状态公告 LSA。直接相连的路由器并不一定就具有毗邻关系。
DR：指定路由器	Designated Router，在每一个广播和 NBMA 网络中，都有一个 DR 用于向公共网络传播链路状态信息。通常，当一个路由器的接口开始工作时，它首先检查是否在此网络上有一个 DR。如有，则接受。
BDR 后备指定路由器	Backup Designated Router，在 DR 故障时很快且平稳地接替 DR。
Internal Router：区域内部路由器	所有与此路由器相连的网络都属于同一区域。
ABR 区域边界路由器	Area Border Router，在不止一个 OSPF 区域内有接口的路由器.
ASBR 自治系统边界路由器	Autonomous System Border Router，一个 OSPF 路由器，它连接到另一个 AS，或者在同一 AS 中，但运行不同于 OSPF 的 IGP。
LSA 链路状态公告	Link State Advertisement，由不同类型的路由器产生 5 种 LSA。
Stub network	只有一个出口路径（路由器）的网络。
Backbone Area 主干区域	一个特殊的 OSPF 区域，称为区域 0 或 0.0.0.0（Area ID）。由所有的 ABR 及其他不属于任何 Area 的路由器组成。位于主干区域上的路由器必须是连续的，即它们中的任意两个路由器都能只需通过主干区域相互通信。

7.3.2　OSPF 的关键特征

作为一个链路状态路由协议，OSPF 有其自身的特点。本节介绍 OSPF 使用的选路算法、

协议流程以及它的主要数据组织。

1．SPF 算法（Shortest Path First，最短路径优先算法）

SPF 算法是 E.W.Dijkstra 提出的，用来计算在一个图中从一个源节点到所有其它节点的最短路径的算法。因为链路状态路由算法使用了 SPF 算法的分布式计算最短路径的思想，因此，链路状态路由算法通常又被称为“SPF 算法”。但是，链路状态路由算法实际上又不一定非要要求用 Dijkstra 算法，只要能有效地计算出最短路径，其它的算法同样允许被应用于链路状态路由算法中即通常讲的“SPF 算法”。

SPF 算法把网络中的节点分为两个集合：已求值节点集合 E（包含全部的最短路径已知的节点）和剩余节点集合 R，另外它还包含一个有序路径列表 O。该算法工作过程如下：

（1）初始化集合Ｅ，使之只包含源节点Ｓ，并初始化集合Ｒ，使之包含所有其它节点。初始化路径列表Ｏ，使其包含一段从Ｓ起始的路径。这些路径的长度值等于相应链路的量度值，并以递增顺序排列列表Ｏ。

（2）若列表Ｏ为空，或者Ｏ中第１个路径长度为无穷大，则将Ｒ中所有剩余节点标注为不可达，并终止算法。

（3）首先寻找列表Ｏ中的最短路径Ｐ（这就是取名“最短路径优先”的原因）。从Ｏ中删除Ｐ。设Ｖ为Ｐ的最终节点。若Ｖ已在集合Ｅ中，继续执行步骤２。否则，Ｐ为通往Ｖ的最短路径。将Ｖ从Ｒ移至Ｅ。

（4）建立一个与Ｐ相连并从Ｖ开始的所有链路构成的侯选路径集合。这些路径的长度是Ｐ的长度加上与Ｐ相连的长度。将这些新的链路插入有序表Ｏ中，并放置在其长度所对应的等级上。继续执行步骤（2）。

容易看出，路径总数等于网络中的链路数。这个算法相当于对这些路径进行排序。如果恰当地执行插入操作，那么此算法的量级是 O(M.logM)。其中Ｍ为链路数。

通常的链路状态路由算法的原理是各个网络节点不必交换通往目的站点的距离，而只需维护一张网络拓扑结构图，在网络拓扑结构发生变化时及时更新即可。利用这些图和 Dijkstra 最短路径算法可以计算出比距离向量协议更为精确的路由。实际上，尽管路由的计算是分布式的，但其结果与集中方式计算出来的一样精确。

网络拓扑结构图以数据库的形式存在，数据库中的每条记录都代表网络的一条链路。例如图 7-10 所示的简单网络采用表 7-2 的链路状态数据库表示各种数据。

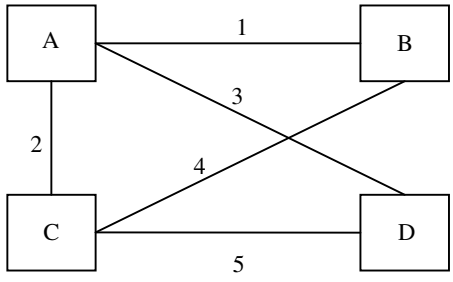


图 7-10　简单网络拓扑图

表 7-2　　链路状态数据库内容

从 节 点	到 节 点	经过链路	距　离
A	B	L1	1
A	C	L2	1
A	D	L3	1
B	A	L1	1
B	C	L4	1
C	A	L2	1
C	B	L4	1
C	D	L5	1
D	A	L3	1
D	C	L5	1

数据库中的距离值可以不是１，只要使用的量度一致，则每条直接相连的链路的距离值可以不同。这就保证了有些链路虽然都是直接相连的但会优先。当然，还可以从另一个角度比如说最小费用的角度去考虑 X.25 链路比 FDDI 优先。设定不同的量度，就可以得到不同优先指数的最短路径表。

之所以大部分的网络专家都喜欢链路状态协议，而不喜欢对距离向量进行改进，这是有相当的理由的。因为基于链路状态的 SPF 算法具有以下优点：

- 迅速无环路的收敛性。
- 支持精确的度量值，而且如果需要，可支持多重度量值。
- 支持通往一个目的站点的多重路径。
- 区分不同的外部路由。

当网络中某条链路状态发生变化时，交换扩散协议会迅速把此项变化传遍全网中每个需要接受此项刷新的节点，每个节点根据此变化重新计算路由表。SPF 算法可以保证即使网络拓扑结构中无论有多少环路都不影响计算路由过程的快速准确，计算结果无环路且路由对于网络上链路的敏感性很高。另外，此种算法还可以支持不同的度量值，可同时支持最大吞吐量、最短时延、最低费用和最高可靠性等路由表。SPF 算法还支持通往一个目的站点的多重等价路由和区分不同类型的自治系统外部路由。

2．协议工作流程

（1）当一个路由器启动后，它首先初始化路由协议数据结构，然后路由器等待下层协议给出的接口开始工作的指示。

（2）接口开始工作，路由器便从该接口发出 Hello 报文，以寻求与邻居建立邻接关系，如双方可以建立邻接关系，则路由器之间开始同步链路状态数据库，等这些工作完成后，双方便正式建立了邻接关系。

（3）邻接关系建立后，路由器就可以了解到它的各个接口连接的路由器或网络情况，从而可以形成路由器链路通告。另外通过邻接关系的建立，multi_access 网络的 DR 就可以了解到该网络中都有那些路由器，从而可以形成网络链路通告。这两种通告都是构造区域拓扑不可缺少的信息。

（4）在协议工作过程中路由器必须发送路由器链路通告，如该路由器为 DR，它还必须发送网络链路通告，如路由器为区域边界路由器，路由器要将接收自其它区域（包括骨干区域）的路由信息整理后（形成汇总链路通告）送入本区域，以上这些信息仅在一个区域内传播。对于由 AS 边界路由器产生的 AS 外部链路通告，AS 内的每个路由器都要获知，也就是说，该信息要传播到 AS 的每个角落。

（5）路由器根据所收到的路由器链路通告和网络链路通告导出本区域的拓扑图，计算出最短路径树，形成本区域内的路由信息，然后再利用汇总链路通告计算出到其它区域的网络或到区域边界以及 AS 边界路由器的路由，最后利用 AS 外部链路通告算出通往 AS 之外网络的路由。

（6）以上链路通告都是周期性发送的，如果路由器接收到新的或与以前不同的链路通告，便对路由表进行更新。

3．路由表的计算

以本区域的链路状态数据库作为输入，路由器通过下面的算法一步一步计算出路由表。在每一步，路由器必须利用到该数据库中的某一单个信息块（例如由某一个路由器产生的 router-LSA）。如果通过这个查找过程返回的 LSA 的 LS age 等于 MaxAge，则这一个 LSA 不应被用来计算路由表，就像认为本次查找过程失败了一样。

建立路由表的过程可以分为以下五个步骤：

第一步，使当前的路由表无效。新的路由表从头开始计算，旧的路由表保存起来用来找出哪些路由项已经变化。

第二步，对每个与之相连的区域建立最短路径树，计算区域间（intra-area）路径。特别是，所有目的类型（Destination Type）为 ABR（区域边界路由器）的路由项被计算出来了。这一步可以分为两部分：首先在建立最短路径树时只考虑路由器和可传递网络，其次考虑把 stub networks 加进去。在区域最短路径树的计算过程中，该区域的 Transit Capability 也被计算出来，第四步要用到此参数。

第三步，通过相关的 Summary-LSA 计算出区域之间的路径项。如果此路由器与多个区域相连（也就是说它是一个 ABR），则只检查骨干区域。

第四步，如果此路由器是一个 ABR 且它与一个或一个以上的可传递网络相连（也就是说，它与非骨干区域相连且此区域的 Transit Capability =1），则检查该区域的 Summary-LSA，看通过上面第（2）、（3）步计算出的路径是否可以因通过本区域而得到优化。

第五步，通过检查 AS-external-LSA 计算到区域外的路径。在以上第（2）至（4）步中，已经计算出产生 AS-external-LSA 的 ASBR。

由于以上计算产生的路由项的变化，OSPF 协议可能产生更进一步的计算。例如，由于区域内的路径项发生变化，ABR 将会产生新的 Summary-LSA。

下面给出（2）～（5）步的详细过程。

（1）计算一个区域的最短路径树

路由器以它自己为根计算最短路径树，此阶段共划分两步：第一步只考虑路由器和 Transit

Networks，用 Dijkstra 算法，根据数据库中这一部分 LSA 计算出一个树。第二步通过计算 stub networks 形成树的叶子。

先大致描述一下计算的第一阶段（主要是 Dijkstra 算法，前面已经介绍了 Dijkstra 算法的主要过程），然后给出第二阶段的详细步骤。

在第一阶段的每一次循环，都有一个候选节点列表，从根部到此节点的路径已经被找到，但不一定是最短路径。但是，在这些候选节点中，对于具有到树根最短距离的节点则可以肯定这条路径是最短的，因此，这条节点被加到最短路径树上，并从候选节点列表中去掉。然后检查此节点的相邻节点,找出需要增加或修改的候选节点列表上的节点。完成后，此算法再接着循环。当候选节点列表中不再有节点时，最短路径树就建立起来了。

建立最短路径树的第二阶段是要将 stub networks 加到树上去。在此阶段，所有路由器节点又被全部检查一遍。那些在第一阶段中被认为不可到达的节点不被考虑。对每个可达路由器节点 V，检查数据库中的路由器 LSA，对在此 LSA 中的每个 stub networks 要做以下计算：

计算从根到此节点的距离 D。比较此距离 D 与当前到此 stub network 的最低费用（通过检查此 stub network 的当前路由表项）。如果 D 大，那就检查 LSA 中下一个 stub network 。

如果计算到这一步，那就是要更新此 stub network 的路由表项。它要计算使用此 stub network 的 next hops。它的输入是目的地（本 stub network）和它的父节点（路由器节点）。①如果距离 D 和现在的费用相同，只需简单地将这些 next hops 加到路由表项的 next hops 即可。在这种情况下，路由表项中已经存在 Link State Origin，如果此 Link State Origin 是 router-LSA，并且它的 Link State ID 比节点 V 的路由器 ID 要小，则将 Link State Origin 设置为 V 的路由器 LSA。②D 小，则重新将路由表项的费用设置为 D，next hops 全部更新为新计算的 next hops。Link State Origin 设为节点 V 的 router-LSA。然后计算下一个 stub nextwork。

对所有在第二阶段新增和被修改过的路由表项，相关的区域应设为区域 A，路径类型应设为 Intra-Area。当所有可达的 router-LSA 被检查完后，第二阶段就完成了。此时，所有区域间的路径都被计算出。

（2）计算 next hops

每个 next hop 由转发 IP 包到目的地的下一个接口和它的路由器 IP 地址决定。每当发现一个更短的路径时就要计算 next hop。这在计算最短路径树的两个阶段都有可能发生。

其他各种路径的计算方法请参考 RFC 2178。

7.3.3 OSPF 协议的实现

作为一种链路状态路由协议，OSPF 为所有网络中的节点同时维护一张完整的网络拓扑结构图，并按该图计算出全部的最短路由。而网络拓扑结构图保存在一个数据库中，数据库中的每一项代表网络中的一条链路。

虽然当今 IP 网络大多通过简单的点到点链路建立起来，但通常还包括以太网、令牌环网或 FDDI 网，甚至有时还是建立在虚电路网络上的，如 X.25 和 ATM。除此之外，当今 IP 网络的规模也十分庞大，因此，OSPF 协议设计时除了提供普通链路状态协议所具有的特点外，还提供了以下特殊的支持：

- 区分主机和路由器。
- 广播型网络，例如以太网或 FDDI。

■　非广播型网络，例如 X.25 或 ATM。

在当今网络中，一个十分常见的模式是 IP 主机连接在一个局域网上，此局域网再通过一个路由器连到大的广域网上。当在数据库中描述这种链路时，OSPF 使用一个特殊的概念“末梢网络链路”（Stub-Network Link），用以表示一个网段上只有一台路由器的网络拓扑结构。OSPF 通过使用这一崭新的名词来把主机和路由器区分开。如果一个网段上存在多个路由器在同时工作，则此网络链路就不能成为末梢网络链路。

为了减少网络的流量开销，OSPF 协议还划分出了广播型网络和非广播型网络。广播型网络指物理上是广播型的网络，如以太网，FDDI 网络等。非广播型网络指物理上不是广播型，但一个网段上又同时有多于两台路由器在工作的网络，如 X.25 和 ATM 等网络。但是点到点的网络不属于非广播型网络。在 OSPF 协议中点到点链路和虚连接是单独考虑的。划分网络链路类型是为了充分利用网络物理上的特性，用最少的网络流量，从而实现 OSPF 协议的正常运行。

当自治系统非常大时，网络拓扑数据库的内容就比较多。一旦网络中某一链路状态发生变化，就会引起整个网络中每个节点都重新计算一遍自己的路由表，从而影响路由协议的性能。因此，OSPF 把自治系统划分为多个域，每个域包括多个网段。每个域内部维持一张惟一的拓扑结构图，域边界路由器把所连的各个域的内部路由总结后在域间扩散。这样，当网络中的某条链路状态发生变化时，此链路所在的域中的每个路由器重新计算整个路由表，而其它域中的路由器只需修改其路由表中的相应条目而无须重新计算整个路由表，节省了计算路由表的时间。

OSPF 路由协议的内容可分为交换扩散协议、网络拓扑结构数据库内容的维护以及路由表的计算三部分。网络拓扑数据库中的每一条内容通过交换扩散协议与相邻的节点达到同步，从而使整个网络中每个节点维持的网络拓扑数据库同步。每个节点当得知网络拓扑结构图发生变化时，重新计算整个或部分路由表以达到路由的迅速更新，使网络拓扑的变化及时地反映在相应的路由表中。路由表的计算在前面已经作介绍，这里不再赘述。在引入 OSPF 的各种报文类型以后，我们主要介绍一下交换扩散协议和网络拓扑结构数据库的维护。

1．报文类型

OSPF 的报文分为 5 种：

■　呼叫（Hello）报文：用来发现相邻的路由器，建立毗邻关系等。通过周期性地发送呼叫报文，呼叫协议还可以用于确定邻居路由器是否还在工作。在广播和 NBMA 网络中，被指定的路由器的选取也要使用呼叫协议。

■　数据库描述（Database Description）报文：任务是描述路由器的链路状态数据库的容量，并且是形成邻接的第一步。数据库描述报文通过投票响应方式发出，一个路由器被指定为主机，其它的被指定为从机，主机发出数据库选票，从机通过发出数据库描述器包来发出应答。

■　链路状态请求（Link State Request）报文：从毗邻的路由器要求 LSA 的信息。链路状态请求报文是 OSPF 的第三类报文，一旦整个数据库使用数据库描述报文来与路由器交换，路由器将比较它邻居的数据库和自己的数据库。此时，路由器也许会发现邻居的数据库在某些部分比自己的更先进。如果这样，路由器将会要求这部分使用链路状态请求报文。

■　链路状态更新（Link State UpDate）报文：发送新的 LSA，路由器使用扩散技术来传递 LSA。LSA 有很多类（路由器、网络、概括和外部等），这些都会在后文中详细介绍。

■　链路状态确认（Link State Acknowledgement）报文：确认链路状态更新报文。这种应答使 OSPF 的扩散过程更可靠。

（1）公共报文头

所有 OSPF 报文都直接在 IP 上封装且 IP 中的协议类型为 89，都有共同的 24 字节首部，如图 7-11 所示。

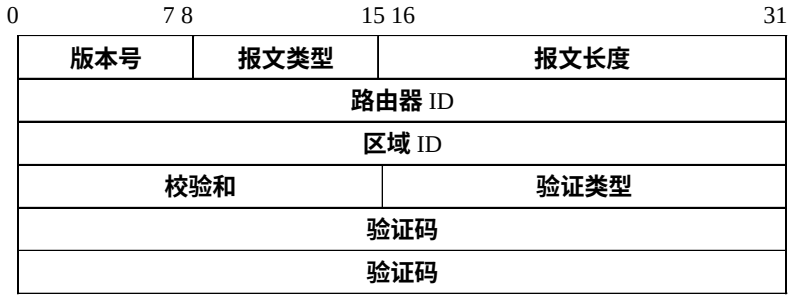


图 7-11　OSPF 报文公共报文头

版本编号设置为２，用来指示 OSPF 的当前版本。报文类型是指 OSPF 报文的类型，共有五种类型的报文，如表 7-3 所示。

表 7-3　　报文的五种类型

报文类型编号	报文类型描述
1	Hello 报文
2	数据库描述报文
3	链路状态请求报文
4	链路状态更新报文
5	链路状态确认报文

报文长度是 OSPF 报文中的字节数。路由器 ID 是所选择的用来标识路由器的 IP 地址。每个路由器都有一个 32 位的标识，此标识在整个自治系统中要做到惟一。它来惟一地代表某个路由器。通常路由器的 ID 用路由器所有接口参数中最大或最小的 IP 地址来代表。区域 ID 是区域的标识。每个区域也有一个 32 位的标识且此标识必须在全自治系统惟一。全为 0 的标识保留给主干区域使用。通常选择一个 IP 网络号作为一个区域的标识。校验和按包括 OSPF 公共报头的整个报文来计算，除去 8 个字节的验证域外。所使用的计算校验和的算法与传统的 IP 报文计算校验和算法一样。验证类型只定义了 2 种标准格式，如表 7-4 所列。

表 7-4　　OSPF 报文的两种验证类型

类　　型	类 型 描 述
0	无验证
1	简单密码验证

在简单密码验证的情况中，验证域是一个８字节的密码。每一个网段可被分配一个密码，

同一网段内，验证密码必须一致，同一域内验证类型必须一致。

（2）呼叫（Hello）报文

呼叫（Hello）报文格式如图 7-12 所示。

图 7-12　Hello 报文格式

除了 Hello 间隔（16 位）、选项域（8 位）和优先权（8 位）之外，其他所有域都是 32 位的。网络掩码是接口所对应的子网掩码。在没有子网路由选择机制的情况下，对应于 A 类网络，它设置成十六进制数值 FF000000，对应于 B 类网络设置成 FFFF0000，对应于 C 类网络设置成 FFFFFF00。

路由器每隔一个“ Hello 间隔 ”发送一个 Hello 报文，内容包括链路上指派路由器的地址（若还没有指派路由器则为 0），以及备份的地址（或若还没有备份指派路由器的则为 0）。还包括一个邻居列表，表示在最近“ 死亡间隔 ”（dead-interval）秒数内从这些邻机那里收到一个 Hello 报文。Hello 间隔与死亡间隔都是由管理员设置的链路参数。

（3）数据库描述报文

数据库描述报文格式如图 7-13 所示。

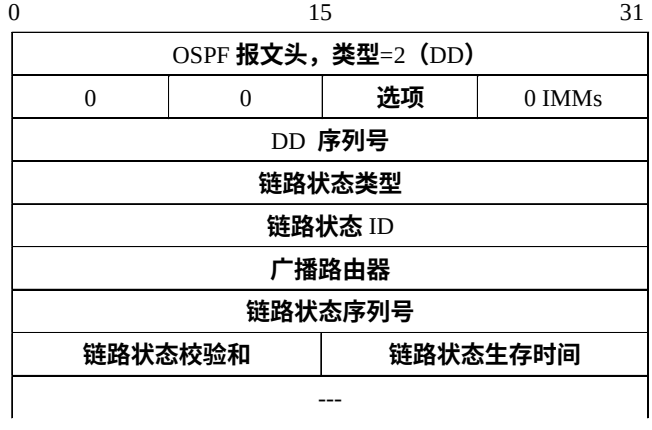


图 7-13　数据库描述报文

交换协议使用“数据库描述报文”从路由器中选择“主”(master)和“从”(s1ave)。当这些角色达成一致之后，2 个路由器就交换它们的数据库描述信息，并且各自都列出在下一阶段被请求的记录。

在相邻路由器建立起毗邻关系后就开始交换此包。跟在 OSPF 报文头后面的第 1 个 32 位字中，除了“选项”字节（与 Hello 报文中的类似）和 3 个位（分别被称为 I （初始化，29)，M（未完，30）与 MS（主- 从，31)）外，全为空。下一个 32 位字是数据库描述（DD）报文的序列号。报文的内容是一组链路状态记录描述：类型、1D、广播路由器、序列号、校验和和生存时间，但没有任何链路状态记录的内容。

（4）数据库请求报文

数据库请求报文格式如图 7-14 所示。

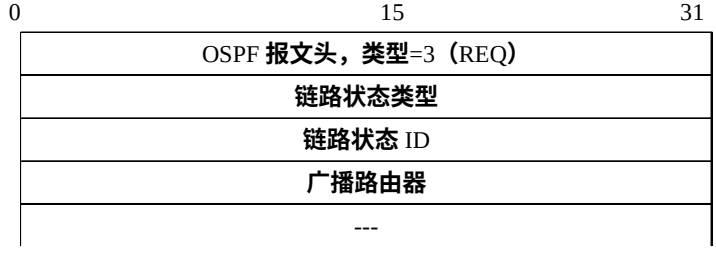


图 7-14 数据库请求报文

数据库请求报文包含一组链路状态记录标识符。每个记录由 3 个 32 位字进行描述：记录类型、记录标识符和广播路由器标识符。当收到这类请求时，路由器将使用与扩散新的记录数值相同的过程来发送一组链路状态更新。每当接收到更新，这个记录描述就从请求记录列表中删除。

（5）数据库更新报文

当一条链路发生状态变化时，与该链路相对应的路由器将发布新版本的链路状态广告。每个记录都由广播路由器标识、链路状态标识与链路状态类型的组合来标识，并有一个序列空间的序列号。这些更新还可以在响应链路状态请求报文时发送。它们被装载在链路状态更新报文之中，如图 7-15 所示。

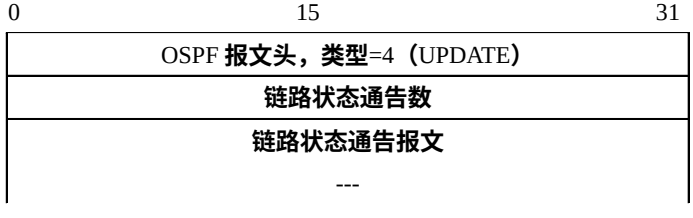


图 7-15 数据库更新报文

跟在 OSPF 头后面的是通告数，接着是链路状态通告报文本身。对于每个通告，序列号都要与本地数据库中的数值相比较。如果这是一个新值，则通告将被安排在其它所有接口上进行传送。不论怎样，应该向传送这一更新报文的路由器确认这个广告。为保证扩散过程可靠性，作为一种尝试，该路由器会以一定的间隔重传它的更新信息，直到接收到确认为止。

（6）链路状态确认报文

通告一般由链路状态确认报文进行确认，链路状态确认报文格式如图 7-16 所示。

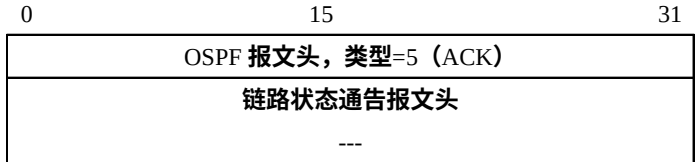


图 7-16　链路状态确认报文

每个确认报文可包含一定数量的通告报头，和交换协议期间在数据库描述报文中被传送的那些报文极其类似。既然一个确认报文可以确认多个通告，就应当延迟它的传送，以便将许多链路状态确认组装到一个报文中去。为了防止不必要的重传，这个延迟必须很短。

2．建立毗邻关系（Adjacency）

在以太网等共享网络上，如果让任何两台路由器之间都相互交换 LSA 信息，会占用许多带宽，并且很难保持路由器之间状态的一致性。OSPF 规定,在多访问网络中,要选出一台指定路由器(DR)和一台备份指定路由器 BDR。DR 和网络上其它的路由器交换 LSA 信息，并代表它所在的网络的链路状态公告 LSA。备份指定路由器 BDR 则在指定路由器 DR 出现故障时快速转变为 DR，保证网络的正常运转。

毗邻（Adjacent）是一种逻辑关系，仅毗邻的路由器之间才能直接交换链路状态公告 LSA。直接相连的路由器并不一定就具有毗邻关系。OSPF 规定，仅在如下路由器之间建立毗邻关系：

- PPP 网络和虚拟链路两端的路由器之间；
- 指定路由器 DR 和同一网上的所有其它路由器之间；
- 备份指定路由器 BDR 和同一网上的所有其它路由器之间。

OSPF 协议运行后，试图与相邻的路由器建立毗邻关系。它定期向各个网络接口（包括虚拟的网络接口）发送 Hello 报文。在 Hello 报文中，包含它自己的 ID（即某一接口的 IP 地址）、优先权（用于选择 DR）、已知的 DR、BDR 和相邻路由器表。接收到 Hello 报文的路由器如果发现自己在对方的相邻路由器表中，这表明双方都收到了对方的 Hello 报文。在多访问网络上，根据优先权值，各路由器选择自己网络上的 DR。因为算法是固定的，所以各路由器选择的 DR 和 BDR 也是一致的。

3．建立完整的数据库

建立了毗邻关系的路由器间相互交换各自的 LSA 数据库内信息的过程，称为数据库的同步。LSA 数据库中包括许多 LSA。每一个 LSA 代表某一个路由器的局部信息，由头部和实例（Instance）两部分报文组成。LSA 头部有 LSA 类型、广播路由器的 ID、序号等信息。图 7-17 为 LSA 头部的格式。

链路状态记录有好几种类型，但所有记录都具有相同的 LSA 头部。

“广播路由器”（Advertising Router）由其多个 IP 地址之中的那个被选定为该路由器的“OSPF 标识符”的地址来标识。“生存时间”（Age）是一个 16 位的无符号整数，用来指示首次做链路状态记录广播之后所经过的秒数。

LSA 头部的 LS 类型、链路状态 ID 和广播路由器可以惟一地标示一个 LSA。每当路由

器发送新的 LSA，其序列号都增加 1。路由器接收到一个新的 LSA，可以根据 LSA 头中的类型、链路状态 ID 和广播路由器，判断它自身的数据库中是否有相同的 LSA，并且根据 LSA 序列号判断更新哪一个 LSA。

LS 生存时间	选项	LS 类型
链路状态 ID		
广播路由器		
LS 序列号		
LS 校验和	长度	

图 7-17　链路状态广告报头

毗邻的两台路由器之间数据库同步的过程如下：假设有两台路由器 A 和 B 刚建立起毗邻关系，路由器 A 和 B 将相互发送数据库描述报文。在数据库描述报文中包括多个 LSA 的头部。如果路由器 B 在路由器 A 发送的报文中发现其中的一些 LSA 头代表的 LSA 在它自身的数据库中不存在，或者收到的 LSA 比它拥有的 LSA 更新，则路由器 B 把该 LSA 头部加入到自己的 LSA 请求表中。然后路由器 B 向路由器 A 发送链路状态请求报文，要求得到 LSA 具体的信息（即实例）。在请求报文的数据部分，放的是多个（LS 类型，LS 标志，广播路由器）三元组，表示要求更新的 LSA。

路由器 A 收到 B 的 LSA 请求报文后，将向 B 发送 LSA 更新报文。在该报文的数据部分，是路由器 B 所请求的 LSA 的完整信息。路由器 B 检查收到的每一个更新报文，把收到的新的 LSA 从 LSA 请求表中删除。同时向路由器 A 发出 LSA 更新的确认包。如果路由器 B 的 LSA 请求表为空，则表明两者的数据库达到一致，即同步成功。OSPF 对每个链路状态更新报文发送链路状态确认报文，保证数据库描述报文的可靠传输。

路由器在其链路状态发生变化或收到其他路由器发送的 LSA 更新报文后，路由器也要向毗邻的路由器主动发送 LSA 更新报文，以便其他路由器尽快更新其拓扑数据库。链路状态公告 LSA 分为 5 类，其发送者及包含的信息如表 7-5 所示。

表 7-5　　5 种 LSA 及其含义

LSA 类型	发 送 者	内　　容
路由器 LSA	所有路由器	描述它自身的各个接口的类型、地址、掩码等信息。
网络 LSA	DR	描述 DR 所在网络的类型以及上面的所有路由器的 ID。
网络汇总 LSA	ABR	描述到区域外的网络的路由信息。
ASBR 汇总 LSA	ABR	描述到 ASBR 的路由信息。
AS 外部 LSA	ASBR	描述到 AS 外部的网络的路由信息。

通过路由器 LSA 和网络 LSA 可以构造一个区域的网络拓扑。不过，只有属于同一个区域的路由器之间才交换路由器 LSA 和网络 LSA。如果一台路由器同时属于多个区域（即 ABR），则它同时要维护多个区域的网络拓扑结构。虽然 LSA 的种类很多，我们只讨论 4 种

主要的 LSA。

（1）路由器 LSA

在一个区域中的每个路由器都产生一个路由器 LSA。路由器 LSA 汇总了所有来自广播路由器的链路。路由器 LSA 只在其所属的区域中扩散。

链路 ID 指的是 OSPF 路由器的 ID，其内容开始是一个 32 位字，指明链路数以及路由器类型，接下来是一组链路描述，如图 7-18 所示。

<table>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>E</td><td>B</td><td>0</td><td>链路数</td></tr>
<tr><td colspan="10">链路 ID</td></tr>
<tr><td colspan="10">链路数据</td></tr>
<tr><td colspan="2">类型</td><td colspan="2">#TOS</td><td colspan="6">TOS 0 度量值</td></tr>
<tr><td colspan="2">TOS=x</td><td colspan="2">0</td><td colspan="6">TOS x 度量值</td></tr>
<tr><td colspan="2">TOS=y</td><td colspan="2">0</td><td colspan="6">TOS y 度量值</td></tr>
<tr><td colspan="10">---</td></tr>
<tr><td colspan="2">TOS=z</td><td colspan="2">0</td><td colspan="6">TOS z 度量值</td></tr>
</table>

图 7-18　路由器链路状态

第 1 个字节的位 6 与位 7，被称为 E 和 B，分别表示路由器是一个区域边界路由器（E，外部）还是一个边界路由器（B，边缘）。每个链路都有链路 ID、链路数据和链路类型，类型可以取 3 种数值：

- 如果此链路是通往另一路由器的点到点链路，则链路 ID 就是这个路由器的 OSPF 标识符，而且链路数据是这个路由器的接口 IP 地址。
- 如果此链路连接到中转网络，则链路 ID 就是这个指派路由器接口 IP 地址，而且链路数据是这个路由器接口的 IP 地址。
- 如果此链路连接到一个末梢网络，则链路 ID 就是这个 IP 网络或子网的编号，而且链路数据是相应网络或子网的掩码。

所有广播都必须包含链路的默认 TOS（0）度量值，但还有可能包含多种（#TOS）TOS 度量值。每种 TOS 度量值都有 TOS 编号（与 IP 报文中出现的数值相同）和相对应的量度值。

（2）网络 LSA

网络 LSA 广播由中转网络（广播型或非广播型）中的指派路由器发布，其链路状态 1D 为 IP 接口的 ID，如图 7-19 所示。

<table>
<tr><td>网络掩码</td></tr>
<tr><td>附属路由器</td></tr>
<tr><td>---</td></tr>
<tr><td>附属路由器</td></tr>
</table>

图 7-19　网络链路状态

报文的内容包括 32 位网络或子网的掩码，接着是所有的附属路由器（Attached Router）。

更确切他说，是所有与指派路由器建立邻接关系的路由器 OSPF 标识符，报文中没有“路由器数量”域，因为这可以从报文内容的长度推导出来。

（3）汇总 LSA

网络汇总 LSA（链路状态类型＝3）与边界路由器的汇总 LSA（链路状态类型＝4）都是由区域边界路由器发布。尽管这些路由器可能会发布多个汇总 LSA，但它们也不会像路由器 LSA 那样将多个汇总 LSA 装到单个报文中去，而是为每个目的站点单独发送一份。链路状态 ID 是 IP 网络或子网的编号（类型＝3），或者是边界路由器的 IP 地址（类型＝4）。报文内容是一个 32 位掩码，再跟一组度量值，见图 7-20 所示。

网络掩码		
TOS=0	0	TOS 0 度量值
TOS=x	0	TOS x 度量值

TOS=z	0	TOS z 度量值

图 7-20　汇总链路状态

掩码是网络或子网的掩码，如果是一个边界路由器，则是十六进制数值 FFFFFFFF。TOS 度量值列表与路由器链路的度量值相同，并且总是以 TOS 0 度量值的度量值开始。其内部没有“TOS 数”域，因为这可以由内容的长度推导出来。

（4）AS 外部 LSA

外部 LSA（链路状态类型= 5）由自治系统边界路由器发布。至于汇总链路，每个记录恰好都有一个目的站点广播。链路状态 ID 是目的站点的 IP 网络或子网编号。报文内容有一个 32 位掩码，后跟一组度量值，见图 7-21 所示。

网络掩码		
E，TOS=0	0	TOS 0 度量值
外部路由标签（0）		
E，TOS=x	0	TOS x 度量值
外部路由标签（x）		

E，TOS=z	0	TOS z 度量值
外部路由标签（z）		

图 7-21　外部链路状态

这里的 TOS 度量值列表与路由器链路中的有 2 点不同：TOS 域本身在位置 0 处包含一个 E（外部）位，而且度量值后还跟有一个 32 位“外部标签”。

外部路由通过边界路由器使用“外部网关”协议（例如 EGP 或 BGP）获取，这些协议没有必要提供与 OSPF 度量值相对应的度量统计值，设置 E 位是为了指明该 TOS 的度量值与内部度量值不可比，并且应该看作“大于任何内部路由”。当 E 应为 0 时，度量值可以加入

到内部路径的开销中，以计算经过边界路由器到达目的站点的路径的开销。“外部路由标签”是一个 32 位域，由边界路由器用来交换有关路由信息，OSPF 不对它进行检查。

4．指定路由器的选取

所有的多路访问网络都有两个或更多的附属路由器，它们会选出一个 DR。DR 的设计思想是使邻接的数目减少，这些邻接是网络中必要的。为了让 OSPF 使路由器能够交换路由信息，必须要有一个邻接。如果没有使用 DR，在一个多路访问网络中的每台路由器都需形成一个与其它路由器的邻接（因为链路状态数据库在邻接中是同时产生的），这个过程将会产生 N - 1 个邻接关系。DR 的使用就是为了减少必须维持的邻接数目。邻接的减少也减少了路由协议交通的容量，以及拓扑数据库的大小。

在一个多路访问网络中的所有路由器，只向 DR 和 BDR 发送路由信息，同时 DR 负责把这些信息扩散到所有邻接路由器去，并以网络的名义产生一个网络链接通告。如果 DR 失效，BDR 将代替它工作。

DR 与 BDR 的选举通过 Hello 报文来进行，但是由于物理网络的区别，选举的过程和要求也不一样。OSPF 有 4 种网络类型或模型（广播式、非广播式、点到点和点到多点），根据网的类型不同，OSPF 工作方式也不同。

（1）OSPF 的网络类型

广播网络：在广播网络中 Hello 报文完全可以动态地发现所有的邻居、DR 和 BDR，因此不需任何特殊的配置信息。

非广播网络：Hello 协议的正确工作需要一些配置信息的帮助。在这种网络中每个合格的路由器（可能成为 DR 的路由器，即路由器的优先级不为 0）都有一个包括所有该网络上的路由器的列表，并且该列表还要标出哪些路由器可以成为 DR。另外合格路由器在非广播网络上的接口一开始工作，它便一直向所有其它合格路由器发送 Hello 报文(这主要是为了发现 DR)，也就是说，不管路由器是不是 DR 或 BDR，任何两个合格路由器总是在交换 Hello 报文，这样做的主要目的是为了保证 DR 选举算法的正确性。另外一旦路由器自身被选为 DR 或 BDR，它就也开始向网络中的所有其它路由器发送 Hello 报文。如果是不能成为 DR 或 BDR 的路由器（优先级为 0），它也必须周期性地向 DR 和 BDR 发送 Hello 报文（如果已存在的话），并向接收到 Hello 报文的其它合格路由器发送响应 Hello 报文。

点到点网络：点到点网络类型是串行口的缺省类型，它没有使用帧中继简化或者被作为子接口的点到点型，一个子接口是一种定义接口的逻辑方式，同样的物理接口能被分成多个逻辑接口，这个概念的产生是为了处理在 NBMA 网中的水平分割问题。点到点模型中，既没有 DR 也没有 BDR，相连的路由器直接形成邻接。并且，每个点到点链路都要求一个分开的子网。

点到多点式网络：点到多点式网络可以被装配到使用配置成多点访问的任何接口。在这种网络中没有 DR，不需要定义邻居，因为额外的 LSA 被用来传播邻居路由器连接。整个网络使用一个子网。

（2）接口状态机

DR 与 BDR 的选举还与接口的状态紧密相关，因此这里先介绍一下接口状态机。接口状态机的转换图如图 7-22 所示。

接口状态被保存在接口数据结构中，共有七种状态，如表 7-6 所示。

影响接口状态发生变化的事件，列于表 7-7。

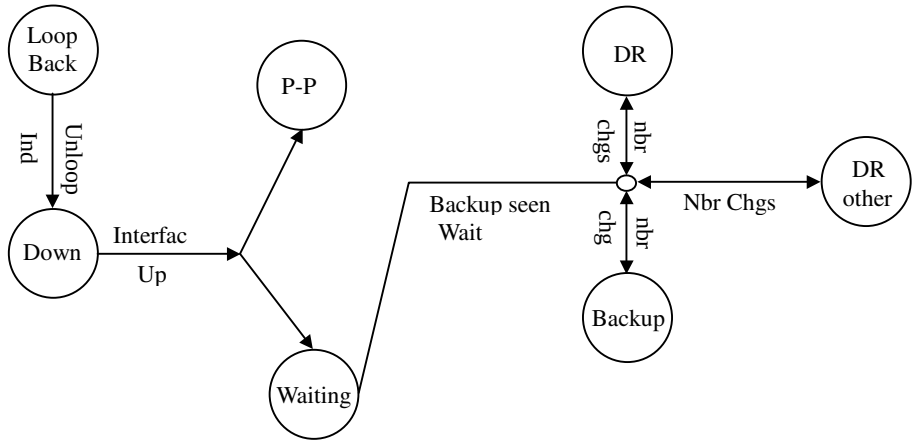


图 7-22　OSPF 接口状态机

表 7-6　　接口数据结构

状　态	含　义
Down	接口的初始状态，处于该状态时，下层协议指出接口不可用。
Loopback	处于该状态时，接口送往网络的所有信息都被回送。Loopback 分为软件和硬件实现两种情况。
Waiting	处于该状态时，路由器监视它所收集到的 Hello 报文，这个状态的目的是为了收集网上关于 DR 和 BDR 的信息，以便于下一步对 DR 和 BDR 的选择。
Point-to-point	接口处于这种状态说明与接口连接的或者是一个物理 point-to-point 网络，或者是一个虚拟链路。
DR Other	说明在该接口网络上，路由器本身既不是 DR，也不是 BDR。
Backup	说明在该接口网络上，路由器本身被选为 BDR。
DR	说明在该接口网络上，路由器本身被选为 DR。

表 7-7　　影响接口状态发生变化的事件

事　件	含　义
Interface UP	下层协议指出接口开始工作，这个事件将使接口根据接口网络的特性而把接口状态推进到 Point-to-point、Waiting 或 DR Other。
Wait Timer	当接口进入 Waiting 状态时，计时器会被启动，当该计时器超时，该事件被触发，接口则开始计算 DR 和 BDR 以决定下一步的状态选择（DR、BDR，或者 DR Other）。
Backup seen	路由器已检测到该接口网络是否存在 BDR(通过两种途径：接收到一个来自邻居的声称自己是 BDR 的 Hello 报文；或者接收到一个声称自己是 DR 的邻居的 Hello 报文，该报文指出不存在 BDR)，该事件的处理与 Wait Timer 相同。这里的邻居是指已建立双向通信的邻居路由器，即路由器自身出现在邻居路由器 Hello 报文的邻居列表中。

续表

事　件	含　义
Neighbor Change	一些邻居状态的变化，将会引起路由器重新计算 DR 和 BDR。这些变化包括：与邻居建立双向通信关系，即邻居已进入 2-WAY 或更高级的状态、退出了与邻居的双向通信，即邻居已进入 Init 或更低级的状态、某个建立了双向通信关系的邻居刚刚声称自己为 DR 或 BDR（检测 Hello 报文即可）、某个建立了双向通信关系的邻居不再声称自己为 DR 或 BDR、某个建立了双向通信关系的邻居改变了所声称的优先级。
Loop Ind	接口接收到一个来自网络管理或下层协议的指示，该指示要求接口进入 Loop Back 状态。
Unloop Ind	接口接收到一个来自网络管理或下层协议的指示，该指示要求接口退出 Loop Back 状态，这时接口会进入 Down 状态。
Interface Down	下层协议指出接口不再工作，接口进入 Down 状态。

（3）DR 和 BDR 的选举算法

DR 和 BDR 的选举算法由接口状态机激活。在这里，假设进行计算的路由器为 X。

首先，路由器 X 先检查它的邻居列表（即那些已与它建立了双向通信关系的邻居，X 本身也包括在内），从中选出那些可以成为 DR 的路由器（即优先级不为 0），然后执行下列步骤：

① 记下当前 DR 和 BDR 的标识，这个记录将被用于以后的比较（DR，BDR 是否发生变化）。

② 计算新的 BDR。首先检查列表并排除已经声称自己为 DR 的路由器，这些路由器不能被选为 BDR，然后查看是否存在声称自己为 BDR 的路由器，如果有，则选择其中优先级最高的作为 BDR，如果最高优先级的侯选路由器有多个，则选择其中 Router ID 最高的作为 BDR。如果没有声称自己是 BDR 的路由器，则在剩余的路由器中选择具有最高优先级的作为 BDR。同样出现多个相同最高优先级的路由器时，选择 Router ID 最高的。如排除已声称为 DR 的路由器后，没有路由器可选则网络中没有 BDR。

③ 计算新的 DR。如果有一个或多个路由器声称自己为 DR，则从中选择优先级最高的作为 DR，出现多个相同最高优先级的路由器时，选择 Router ID 最高的。如果没有声称自己为 DR 的路由器，则将确定的 BDR 作为 DR。

④ 如果路由器本身就是新选的 DR 或 BDR，或现在不再作为 DR 或 BDR，则计算须重复第二步和第三步的计算。这将保证没有任何路由器同时声称自己既是 DR，又是 BDR。

⑤ 路由器根据计算的结果进入相应的状态（DR，BDR，或者 DR Other），如路由器为 DR 或 BDR 则执行相应的功能。

⑥ 如所连接的网络是非广播性的网络，并且路由器刚成为 DR 或 BDR，则路由器开始向其它不能成为 DR 的路由器（优先级为 0）发送 Hello 报文。

⑦ 如前面的计算引起了 DR 或 BDR 的改变，那么与接口相关的邻接关系可能需要进行修改，一些关系需要被建立，而另一些可能需要被打破，因此路由器要对每个已建立双向通信关系的邻居触发 Adj OK（见邻接关系）事件，重新检查所建立的邻接关系是否合格。

5．小结

OSPF 是一个非常复杂的协议，实现 OSPF 要本着以下四个原则：数据结构具有可扩展性、各模块划分独立且接口明确、程序不依赖于任何网络拓扑结构、参数配置简单。在设计数据结构时，考虑到 OSPF 处理的中心内容为拓扑结构图，而网络中的拓扑结构图是没有任何定式的，对于 OSPF 程序来说，每个组最多有多少个区域，每个区域最多有多少个接口，每个接口最多有多少个邻居都是未知的，都是随网络拓扑结构的变化而变化的。

7.4 BGP 协议设计

Internet 的路由选择技术比较复杂，根据网络结构和路由器所处位置的不同，一般把路由协议大致分为自治系统之间的外部网关协议（EGP）和自治系统内的内部网关协议(IGP)。BGP 协议是为 TCP/IP 网络设计的自治系统间的路由协议，即边界网关协议。随着 Internet 的迅速发展，网络拓扑的日趋复杂，多个自治系统间通信的要求越来越高，BGP 协议也愈来愈显得重要。

BGP 的设计经历了四个阶段，其中第四版本 RFC 1771 于 1995 年 3 月出版，是最新的版本，即 BGP-4， BGP-4 是目前 Internet 上使用的外部路由协议。

7.4.1 BGP 协议简介

BGP-4 是自治系统之间的路由协议。它的主要功能就是与其它的 BGP 系统交换网络层可达性信息（NLRI，Network Layer Reachable Information）。网络层可达性信息 NLRI 中包含了可达性信息所经过的自治系统列表，从而构造了一个自治系统连接图，以避免路由环路，同时也使得基于自治系统级别的策略控制成为可能。BGP-4 协议提供了系列新的机制，用来支持无类别域间选路。这些机制包括用 IP 地址前缀来消除网络类别的概念，以及引入新的属性支持路由聚合等。

BGP-4 运行在 TCP 协议上。在 TCP 端口建立 TCP 连接后，BGP 就将全部的路由信息传播出去，以后只有在路由信息发生变化时才传播路由信息。BGP 不需要周期性地刷新路由表。它通过周期性地发送 keepalive 报文来确定连接的存在，如果有错误发生，便发送 notification 报文，并断掉该连接。

一般地，通过 BGP 协议直接通信的路由器称为 BGP 发言人（BGP Speaker）。BGP 不仅指定了路由信息如何在不同的自治系统上的 BGP 发言人间传播，也规定了在同一自治系统上的 BGP 发言人间路由信息的交换。

1．相关概念

在详细讨论 BGP 的复杂细节之前，对关键的术语和概念有一个清楚的理解，其中有些是可互换使用的。

自治系统 AS：从选路的角度来说，拥有同一选路策略、在统一的技术管理机构下的一系列路由器和网络称为一个 AS（自治系统）。一个 AS 的管理相对其它的 AS 而言，有独立而统一的内部路由策略，对外呈现一致的路由信息。BGP 用 AS 号来识别是否为同一个 AS。A S 号由 Internet 注册机构分配。

BGP 发言人：通过 BGP 协议进行直接通信的路由器称为 BGP 发言人。对一个指定的

BGP 发言人，和它进行通信的其他的 BGP 发言人，被称为 peer（对等体）。若该 peer 和指定的 BP 发言人在不同的 AS，称为外部 peer，若在同一个 AS，称为内部 peer。两个路由器之间的相邻连接，也称为对等体连接。

EBGP 和 IBGP：BGP 不仅指定了路由信息如何在不同 AS 内的 BGP 发言人之间的传播，而且也规定了在同一 AS 内的 BGP 发言人间的路由信息交换，以在对其他自治系统运行 EBGP 的边界路由器间传递信息。那么在同一 AS 内的 BGP 发言人运行的 BGP 协议称为 IBGP，而在不同 AS 之间运行的 BGP 协议称为 EBGP。EBGP 和 IBGP 间的区别表现为每个对等体如何处理从其他对等体来的路由更新，以及不同的 BGP 属性在外部与内部链路上相比的传送方式。

CIDR：无类域间路由 CIDR 是用于解决 Internet 路由器的 IP 路由表爆炸性增长，以及抑制 I P 地址空间的耗尽的一种机制。CIDR 是一种地址分配方案，它在 BGP 中消除了网络类的概念。在 CIDR 中，一个 IP 网络由一个前缀表示，前缀是 IP 地址中最左边相邻的有效位，其余为 0。如：172、26、12、1 的前缀可以为 172、0、0、0 或 172、26、0、0。

网络层可达性信息（Network Layer Reachability Information，NLRI）：BGP 通过 NLRI 支持无类别域间路由。NLRI 是 BGP 更新报文的一部分，用于列出可到达的目的地的集合。BGP 更新报文中的 NLRI 域，包含二元组<长度，前缀>。长度是掩码中的位数，前缀就是 IP 地址。这两个合起来代表网络号。例如，网络 10.0.0.0/8 在 BGP 更新报文中会用< 8 , 10.0.0.0> 标识。

2．BGP-4 关键特征

BGP 认为 IGP 路由协议完成自治系统内的路由，它并不对自治系统内的路由作任何假设。特别是，BGP 不需要所有自治系统运行同样的内部路由协议。它对底层的网络拓扑没有任何限制，通过 BGP 的 UPDATE 报文交换的信息已经足够用来构建一个自治系统连接图，有了自治系统连接图就能够消除路由环，同时也能够在自治系统级应用路由策略。

BGP 协议的关键特征主要体现在两个概念：路径属性（PATH Attributes）和网络层可达性信息的聚合。

（1）路径属性

BGP 属性是一个用来纪录特定路由信息的参数集合，这些参数包括：路径属性、一个路由的参考度、一个路由的下一跳信息和聚合信息等等。这些属性主要是用于 BGP 过滤以及路由决策的过程。路径属性使 BGP 具有了很好的灵活性和可扩充性，它可以分为众所周知（well-known）的路径属性（又分为众所周知的强制属性和众所周知的自由属性）和可选（optional）路径属性（又分为可选可传递的属性和可选不可传递的属性）。可选属性允许在一些运行 BGP 协议的路由器之间进行试验而不影响网络的其他部分。

一个最重要的路径属性就是 AS-PATH，它由网络层可达性信息经过的各个自治系统的自治系统号组成。AS-PATH 可以直接消除路由信息环，同时它也是一个用于基于策略路由的强有力的多功能机制。BGP-4 通过使 AS-PATH 包含一系列自治系统号以及一个链表扩充了 AS-PATH 的属性，这种扩展的格式使广播从比较详细的路由中聚合出来的聚合路由变为可能。

每个路径属性以三元组<属性类型，属性长度，属性值>的形式表示。其中，属性类型是 2 字节的字段，包括 1 字节属性标记和 1 字节属性类型代码。路径属性分为 4 类：公认必遵、公认自决、可选过渡及可选非过渡。

4 种属性类别的意义，列于表 7-8。

表 7-8　　BGP-4 属性类型

序号	属 性 名 称	含　义
1	公认必遵（Well-Known Mandatory）	在 UPDATE 报文里必须包含的属性，必须能被所有的 BGP 实现识别。ORIGIN 属性、AS_PATH 属性和 Next_Hop 属性就是三个这种属性。
2	公认自决（Well-Known Discretionary）	能被所有的 BGP 实现识别的属性，但在 UPDATE 报文中可发可不发。
3	可选过渡（Optional Transitive）	若 BGP 工具不能识别可选属性，就查看该属性标记。若标记置位，表明属性是过渡的，那么 BGP 工具接受该属性，并向前传送给其他发言人。
4	可选非过渡（Optional Non-transitive）	当可选属性未被识别，而且过渡标记未置位时（即非过渡的），该属性应被忽略，不传递给其他 BGP 对等体。

属性长度记录属性的长度值，若数据扩展位为 0，那么长度用 1 个字节表示，否则用 2 个字节表示。属性值依据不同的属性类型，有不同的定义如表 7-9 所示。

表 7-9　　BGP-4 的各种属性

序号	属　性	含　义
1	ORIGIN 属性	值域大小是 1 字节。依据不同的属性类型，属性值域的内容，有不同的定义： 0：IGP； 1：EGP； 2：INCOMPLETE。
2	AS_PATH 属性	值域由一系列的 AS 路径段组成。每个 AS 路径段由一个三元组<路径段类型，路径段长度，路径段值>表示。
3	NEXT_HOP 属性	值域大小为 4 个字节，记录的是下一跳的 IP 地址。
4	MED 属性	值域大小位 4 个字节，是一个可选非过渡属性。
5	LOCAL_PREF 属性	值域大小也为 4 个字节，用于区分到同一目的地的各个路由优先程度的。
6	ATOMIC_AGGREGATOR 属性	长度域值为 0，所以不存在值域。
7	AGGREGATOR 属性	域大小为 6 个字节，它包含形成该聚合路由的最后一个 AS 号（2 字节）和形成该路由的 BGP 发言人的 IP 地址（4 字节）。

（2）路由聚合

BGP-4 支持无类域间路由 CIDR，CIDR 通过超网机制消除了 IP 类型编址的概念。超网常用于 B 类和 C 类地址。例如，具有子网掩码 255.255.255.0 的 C 类地址 192.168.200.0 使用带有 255.255.0.0 子网掩码的 192.168.0.0，从而变成一个超网。为减少路由更新报文的大小而使用 CIDR 超网是很有用的，这样也减少了路由表的大小。

通过传播 IP 网络地址最常用的那些位，BGP-4 允许网络地址聚合。聚合就是把几条不同路由组合成一条路由来广播，聚合可作为决策过程的一部分，从而减少广播路由的信息量，公布一个单一路由用于建立超网的路由表。图 7-23 所示为一个路由聚合的示例。

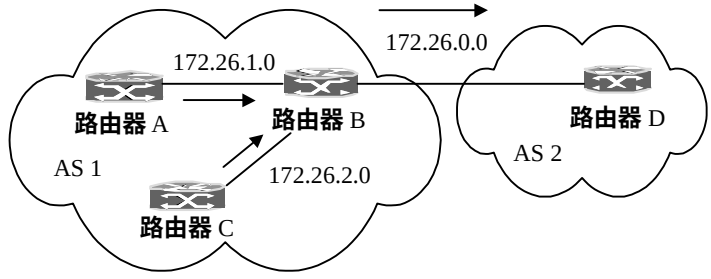


图 7-23　聚合路由

当路由具有属性 ORIGIN、AS_PATH、ATOMIC_AGGREGATE 或 AGGREGATOR 时，只要聚合的各个路由的属性 MED 和 NEXT_HOP 相同，就能进行聚合。具有不同属性类型代码的路由不能聚合。

具有相同属性类型代码的路由，根据以下规则进行聚合：

ORIGIN 属性：如果在要聚合的路由当中，至少有一条路由的 ORIGIN 属性值为 INCOMPLETE，那么该聚合路由的 ORIGIN 属性值也必须是 INCOMPLETE；否则，若至少有一条路由的 ORIGIN 属性值是 EGP 时，那么聚合路由的 ORIGIN 属性值应是 EGP。其它情形下，聚合路由的 ORIGIN 的属性值是 IGP。

AS_PATH 属性：如果要聚合的路由具有相同的 AS_PATH 属性，则聚合路由的 AS_PATH 属性也是同样的。为了便于聚合 AS_PATH 属性，我们为每个 AS 的 AS_PATH 属性建立一个二元组模型：<属性类型，属性值>。属性类型指一个路径段的 AS 所属的类型（如 AS_SEQUENCE，AS_SET），属性值即 AS 号。

若要聚合的路由具有不同的 AS_PATH 属性，则聚合该属性时，需满足以下所有条件：

① 聚合后的 AS_PATH 属性中的所有值为 AS_SEQUENCE 的二元组，应当包含在要聚合的每个初始路由的 AS_PATH 中。

② 聚合后的 AS_PATH 中的所有值为 AS_SET 的二元组，至少应包含在初始路由集中的某一个路由的 AS_PATH 中（属性值可以是 AS_PATH 或 AS_SEQUENCE）。

③ 如果在聚合后的路由中，值为 AS_SEQUENCE 的二元组 X 在二元组 Y 之前，那么在初始集合中的每个包含 Y 的路由中的 AS_PATH 中，X 也应在 Y 之前。

④ 不论什么类型，在聚合的 AS_PATH 路由中，不允许出现具有相同 AS 号的二元组。

当然在具体实现聚合时，可以选择任何算法，只要满足上述条件。

这里介绍一种较简单的聚合算法：

① 首先在要聚合的路由集中，找出每个路由的 AS_PATH 都共有的最长的一组 AS 序列，并将此序列作为聚合 AS_PATH 属性的主序列；

② 将其余的二元组的属性类型设为 AS_AET，然后添加到聚合 AS_PATH 属性序列中；

③ 若聚合后的 AS_PATH 中有多个具有相同 AS 号的二元组（不考虑属性类型），那么删

除这些二元组中类型为 AS_SET 的二元组，最后只保留一个二元组。

在 RFC1771 文档的附录中还介绍了另一种聚合算法，该算法适应性更好，且支持更复杂的配置策略，有兴趣的读者可以参考原文档。

ATOMIC_AGGREGATE 属性：如果在要聚合的路由中，至少有一条路由具有 ATOMIC_AGGREGATE 属性，那么聚合后的路由也必须有该属性。

AGGREGATOR 属性：聚合时，忽略要聚合的路由的所有 AGGREGATOR 属性。

3．BGP-4 的路由算法

BGP 所采用的算法是一种路径向量算法（path vector），路径向量算法既不能被归类为纯距离向量（distance vector）算法，也不能被归类为纯链路状态（link state）算法。BGP 在它的 AS-PATH 路径属性中包含一个完整的 AS 路径从而构建一个网络拓扑图，这就使它比较类似于链路状态算法。而在对等体之间仅仅交换当前所用的路由又使它比较类似于距离向量算法。

同这种路径向量算法结合在一起，为了节省带宽和处理能力，BGP-4 使用“递增式”（incremental）的更新机制，这种机制在最开始交换完整的路由信息以后，在对等体之间就只交换关于那条路由的更新信息。递增式更新机制需要对等体之间有比较可靠的传输，为了满足这一要求，BGP 使用 TCP 作为它的传输协议。除了递增式的更新机制，BGP-4 还引进了路由聚合的概念，从而关于一组网络的信息可以表示为一个路由。

BGP-4 是一个自包含协议，也就是说它不仅规定了如何在属于不同自治系统的 BGP 对等体之间交换路由信息，还规定了如何在属于同一个自治系统的对等体之间交换路由信息。BGP-4 不需要一个自治系统内所有的路由器都参与 BGP 路由，只有连接本自治系统与邻接自治系统的边界路由器才参与 BGP 路由，限制运行 BGP 协议的路由器的数目也正是实现伸缩性的一种方法。为了实现 BGP 和 IGP（包括：IGRP、RIP、OSPF 以及 EIGRP 等）的完美结合，BGP-4 支持交换静态配置的外部路由，还可以通过路由重发布将 IGP 路由公布出去。

下面介绍在正常情况下，BGP 协议消耗的链路带宽、路由器存储空间以及路由器 CPU 周期，同时也描述了 BGP 的扩展性以及相关的一些限制。

4．链路带宽开销和 CPU 开销

在初始的 BGP 连接建立以后，对等体之间就立即交换完整的路由信息集合。目前，主要 ISP 的核心路由器一般都有到达每个可达 IP 地址的一条路由，在 2001 年 1 月对路由器 UPDATE 数据库的分析表明，将近有 120000 条 IPV4 地址前缀[8]。这样，每一个 BGP 路由器都必须维持一个完全的路由表，并且当路由变化时，向它的每一个邻居发送到每个前缀的最佳路由。随着网络的不断快速增长，这个数目也在不断增长。统计表明，一个骨干网上的核心路由器每天要从它的每一个邻居收到 5000 个 UPDATE 报文[8]。

为减少边界路由器所保持和交换的 NLRI 条目数，引入了 CIDR（Classless Inter-Domain Routing，无类域间路由）和“超网”（Supernet）的概念。超网描述了一个基于有类网络的具有双重功能的聚合，BGP-4 通过广播少数几个大的聚合块而不是广播许多小的有类网络信息，来减少路由器所要维持的 NLRI 条目数。如果我们简单地把各个聚合块列为它们各自有类的网络信息，那么就可以把剩余的空间看成为将来的扩展所作的保留。评价 BGP-4 聚合是否成功的最好尺度就是采样现在 Internet 上 NLRI 条目数，并把这个数字与没有应用 BGP-4 以前的数字进行比较。

在 2001 年 12 月，路由器拥有全部 Internet 直连的 NLRI 入口项的个数为 130000。BGP

开发小组认为这个数字还会以每个月增长 4000 条的速度递增。然而应用 BGP-4 的聚合以后，这个增长速度就会大大减慢，采样表明，大概只有 6000 项[8]。

BGP 消耗的带宽和 CPU 周期不依赖于 NLRI 条目数，而是 Internet 的稳定性。如果 Internet 是稳定的，则 BGP 消耗的带宽和 CPU 周期，只是由于交换 BGP 的 keepalive 报文引起的（keepalive 报文只在 BGP 对等体之间交换，推荐的交换频率是每 30 秒一次）。实际上 keepalive 报文非常短（只有 19 个字节），消耗的带宽是大约 5bits/s，基本不消耗任何处理能力，这种开销可以忽略。如果 Internet 不稳定，则只有可达性信息的改变（这是由于网络的不稳定造成的）才在对等体之间交换。当每个 UPDATE 报文只包含一个单个网络的更新信息时，UPDATE 报文的开销最大。应该指出的是，在实际应用中，路由的改变非常依赖于 AS-PATH 属性，也就是说，改变的路由一般都有共同的 AS-PATH 属性。在这种情况下，多个网络信息就可以归为一个单一的 UPDATE 报文，从而大大提高了带宽和 CPU 的利用率。

因为在稳定的状态下，BGP 协议消耗的带宽和 CPU 周期仅仅依赖于 Internet 的稳定性，但丝毫不依赖于组成 Internet 的各个自网络的数目，所以假设各个自治系统之间连接的总体稳定性可以被控制的话，BGP 协议在带宽和路由器 CPU 周期上的开销就具有了非常好的可伸缩性。Internet 的快速增长使网络的稳定性成为一个非常重要的问题，但 BGP 本身并没有为 Internet 带来任何不稳定因素。通过对网络的观察发现，现在网络的不稳定大多是因为在各个自治系统内部使用不恰当的内部路由引起的。

5．存储开销

目前从总的网络数来看，平均的 AS 距离增长的速度非常慢，如果我们假设 Internet 中网络数目的增长速度大大高于每一个路由器平均的对等体数目，则 BGP 协议在存储开销方面的伸缩性同 Internet 中的网络数目成线性增长的关系。

BGP-4 的存储开销依赖于底层的 IP 协议以及 IP 协议所采用的地址分配机制，在有更加灵活的地址分配机制的情况下，它会有更好的伸缩性。随着无类域间路由、超网和聚合等概念的引入，BGP-4 在存储方面的限制逐渐减少。同时，BGP 在存储上支持自治系统分层也只需要很少的附加条件，这一类的分层，同更加灵活的地址分配机制一起，都能够为 BGP 协议所用，从而提供了无限的伸缩性。

7.4.2　协议操作机制

前面已经介绍过，BGP 是自治系统间的路由协议，它的主要功能是和其他 BGP 发言人交换网络可达性信息。一个 BGP 发言人是任何运行 BGP 协议的设备。

BGP 使用 TCP 作为它的传输协议（端口 179），TCP 为 BGP 协议提供可靠的数据传输。形成一个 TCP 连接的两个 BGP 路由器，称为邻居或者对等体。一旦传输连接形成，两个对等路由器交换报文确认连接参数。在这一步，路由器交换 BGP 版本号、AS 号、保持时间、BGP 标识和其他可选参数等信息。如果对等体间有任何一个参数不一致，就会有发送 Notification 报文，从而对等体关系建立不成功。

如果对等路由器都同意这些参数，则整个 BGP 路由表通过 UPDATE 报文交换路由信息。UPDATE 报文包含了经过每个系统的 NLRI 列表，以及每个路由的路径属性。路径属性包含了诸如路由源（ORIGIN）之类的信息和优先权的高低。路径属性将会在本章后面详细讨论。

BGP 路由表在 BGP 连接的过程中对每个对等体都是有效的。如果有路由报文发生了变化，邻居路由器使用增量的更新（报文）来传递这个信息。BGP 并不要求刷新路由信息。如果没有路由变化产生，BGP 对等体之间仅交换 Keep Alive 报文，Keep Alive 报文被周期性地发送以确保连接是保持有效的。

1．运行于 TCP 之上

BGP 协议是运行在 TCP 之上的一个应用，它通过可靠的传输协议来交换信息。这样 BGP 协议本身不需对分段、重组、重传、确认及序列号等问题进行处理。BGP 除了自己的认证机制外，还可以利用 TCP 的认证机制。

BGP 使用 TCP 端口号 179 建立 TCP 连接。

2．报文交换

运行 BGP 时，BGP 发言人首先通常在 TCP 端口 179 等待连接，同时和其配置的对等体主动建立连接，一旦传输层连接建立后，双方就各自发送 OPEN 报文协商连接参数。参数确认后，BGP 发言人之间才可以交换路由信息。

初始的数据流是全部的 BGP 路由表，之后，只有在路由信息发生变化时，才发送更新信息。为确保连接的存在，BGP 发言人周期性地发送 KeepAlive 报文。当有错误和特殊情况发生时，便发送 notification 报文，并关闭连接。

BGP 一共有四种类型的报文：OPEN 报文、UPDATE 报文、Keep Alive 报文、Notification 报文。

3．安全考虑

除了运行于可靠的传输层上之外，BGP 还采取了一些其它的安全措施：

建立 TCP 连接的 BGP 对等体必须在本地被配置。也就是说，当一个 BGP 发言人与其它的 BGP 发言人交换路由信息时，它必须是被其它的 BGP 发言人配置了的 BGP 对等体，否则其它的 BGP 发言人不接受其发起的 TCP 连接。

AS 号必须配置且互相匹配。当两个 BGP 发言人建立了 TCP 连接，在交换 OPEN 报文后，如果发现报文中的 AS 号和配置的不同，那么便关闭 TCP 连接，不再交换其它的路由报文。

4．路由取消机制

BGP 协议提供了路由取消机制。BGP 发言人可以通知它的对等体原先自己发布的路由已不再使用。BGP 通过下面的三种方法表明该路由服务已被取消：

- 把原先发布的路由的 IP 前缀放在 UPDATE 报文中 WITHDRAWN ROUTES 域中进行通告。
- 通告一条具有相同网络可达信息的替换路由信息。
- 关闭 BGP 发言人之间的连接。

5．路由信息库

BGP 协议的所有路由存储在其路由信息库中。BGP 协议提出了一个路由信息库概念模型，该模型指出 BGP 的路由信息库由三部分报文成：

- Adj-RIBs-In：记录的是本地 BGP 发言人从它的对等体获得的未经处理的路由信息，该路由信息可作为决策过程的输入；
- Loc-RIB：记录的是通过本地 BGP 发言人的决策过程后的路由信息；

■　Adj-RIBs-Out：记录的是通过本地 BGP 发言人的 UPDATE 报文通告给特定（配置的）对等体的路由信息。

总的来看，Adj-RIBs-In 包含了从对等节点广播到本地 BGP 发言人的无特权的路由选择信息；Loc-RIB 包含了从本地 BGP 决策进程中收集到的路由；Adj-RIBs-Out 依靠本地发言人的更新信息组织路由，广播到特定的对等节点。

需要指出的是，尽管理论模型上区分 Adj-RIBs-In、Loc _RIB、Adj-RIBs-Out 它们之间的区别，但这三个信息库仅仅是一个概念模型，而协议真正实现时，不必一定保留三个独立的路由信息备份，也就是说协议实现可以不受制于协议规范。

6．IBGP 的连通性

如果一个 AS 内有到其它 AS 的多个连接，那么它可能需要多个 BGP 发言人。为了避免在 AS 内产生路由环，BGP 不向 IBGP 对等体通告从其他 IBGP 对等体得知的路由。因此，在 AS 内保持一个完整的 IBGP 闭合网是很重要的，这就是说，AS 中的每一个 BGP 路由器必须与该 AS 中的所有其他 BGP 路由器建立 BGP 对话。

7．版本协商

BGP 发言人通过多次打开 BGP 连接协商协议的版本号，如果两个对等体支持的版本不同，那么选择相同版本中的最高版。

如果一个带有错误码的 OPEN 报文建立连接的试图失败，而且错误子码不支持版本号，那么 BGP 发言人用它尝试过的可用版本号，版本号通过 notification 报文广播给对等体，声明本地系统支持这个版本号。如果两个对等支持一个或多个普通版本，那么这将允许它们迅速决定最高的普通版本。为了支持 BGP 版本协商，未来的 BGP 版本必须要保留 OPEN 报文和 notification 报文。

7.4.3　BGP 协议的实现

BGP-4 的主要功能是与其它的 BGP 系统交换网络层可达性信息。网络层可达性信息 NLRI 中包含了可达性信息所经过的自治系统列表，从而构造了一个自治系统连接图，以避免路由环路，同时也使得基于自治系统级别的策略控制成为可能。BGP-4 协议提供了系列新的机制，用来支持无类别域间选路。这些机制包括用 IP 地址前缀来消除网络类别的概念，以及引入新的属性支持路由聚合等。

在具体实现时，除了要考虑 BGP-4 规范所规定的各种规则以外，还要考虑其互操作性、稳定性以及可扩展性。另外，因为 BGP-4 运行在 Internet 骨干网上，其路由表的容量和性能直接影响着整个网络的稳定性和性能，因此其各种数据结构，特别是路由表的组织，也是要考虑的一个重要因素。

1．报文处理

BGP 报文的传送是在可靠的传输协议之上进行的。只有当整个报文都收到之后，BGP-4 才对报文进行处理。报文的最大长度是 4096 字节，最小长度是 19 字节，即只包含 BGP 头部信息，没有数据域。

BGP 共有四种报文类型：OPEN 报文、UPDATE 报文、notification 报文、keepalive 报文，这四种报文有一个公共的报文头部。

（1）报头格式

每个报文都有固定长度的报头，长度为 19 字节。在报头后，依据报文类型不同，可以有数据部分，或者没有。报头格式如图 7-24 所示。

图 7-24 BGP-4 报文头格式

标记域长度为 16 字节，通常情况下的值为全 1。如果 OPEN 报文带有认证信息（这是一个可选参数），那么其它的报文类型中的 Marker 值是经过加密的。加密的算法由认证机制来规定。标记域用来检测一对 BGP 对等体间的同步，并鉴别进来的 BGP 报文。

长度域是 2 字节的无符号整数，它标识包含头部在内的整个报文的字节长度。该域的值必须在 19 和 4096 之间。

类型域是 1 字节的无符号整数，它标识报文的类型。报文的类型定义如下：

1：OPEN 报文

2：UPDATE 报文

3：notification 报文

4：keepalive 报文

（2）OPEN 报文

在传输协议连接建立之后，两边发送的第一个消息是 OPEN 报文。如果 OPEN 报文可以接受，需要发回一个 keepalive 报文来确认 OPEN 报文。一旦确认了 OPEN 报文，即可开始交换 UPDATE，KEEKPALIVE 和 notification 报文。除了固定长度的报头外，OPEN 报文格式如图 7-25 所示。

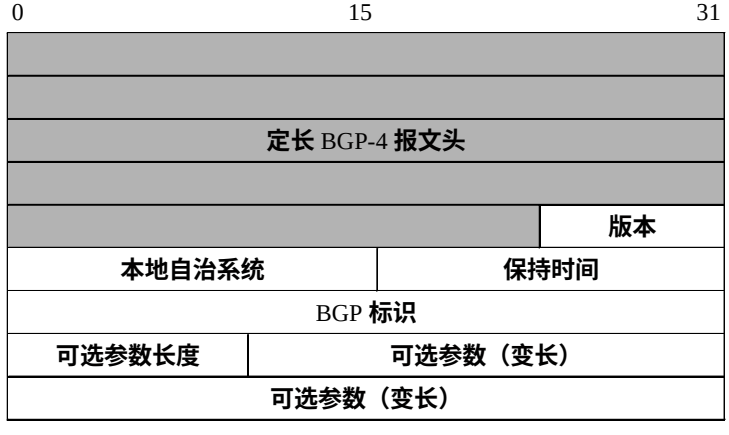


图 7-25 OPEN 报文格式

其各个域的描述如表 7-10 所示。

表 7-10　　OPEN 报文各域

域　名	含　义
版本	1 字节的无符号整数，表示 BGP 协议的版本号，目前的 BGP 版本号是 4。
本地自治系统	2 字节无符号整数，指出 BGP 路由器的 AS 号码。
保持时间（Hold Timer）	2 字节无符号整数，记录发送端所设定的保持定时器的数值。保持时间是指两个相继的 keepalive 或 UPDATE 报文接收之间消耗的以秒计算的最大时间值。BGP-4 路由器与它的对等体协商过程中，把保持时间设置为两者保持时间较小的那个值。保持时间可以是零，此时，保持计时器和 keepalive 计时器永不被复位，连接被认为总是在工作中。建议的最小保持时间是 3 秒。
标识符	4 字节的无符号整数，表示发送端的 ID。BGP 的标识符在启动时确定。BGP 发言人将分配给它的一个 IP 地址设为自己的标识符。
可选参数长度	1 字节的无符号整数，表示以字节为单位的可选参数字段的总长度。长度为 0，表示没有可选参数出现。
可选参数	一个可变长度字段，表示 BGP 对等体协商期间使用的一系列可选参数。每个参数是一个三元组，由参数类型、参数长度及参数值三部分表示。这三部分的长度分别为 1 字节、1 字节和可变长。

（3）UPDATE 报文

BGP 协议的核心是路由更新。路由更新包括 BGP 用来组建无循环的网络结构所需的所有信息。更新报文包含的基本部分是：网络层可达信息（NLRI）、路径信息和撤销路由。一条 UPDATE 报文可以向它的对等体发布一条可行性路由，或者撤销多条不可行路由，也可以同时发布一条可行性路由和撤销多条不可行路由。UPDATE 报文除了包含固定长度的报头外，还包含其他字段，如图 7-26 所示。

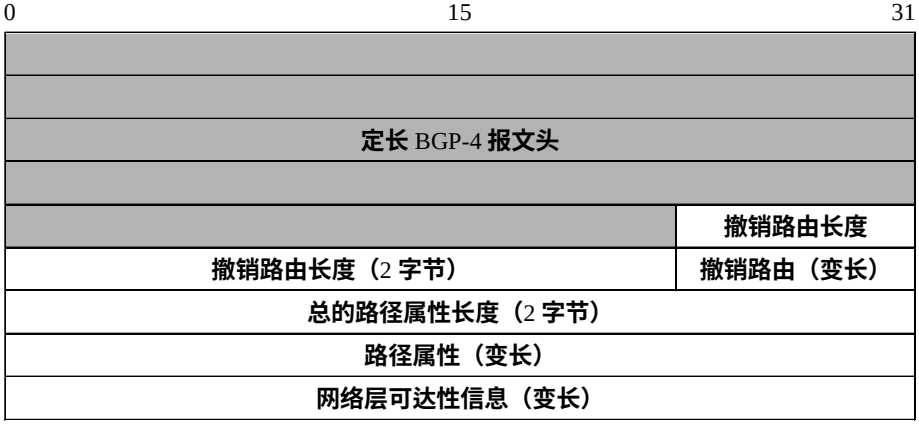


图 7-26　BGP-4 UPDATE 报文

各个字段描述如表 7-11 所示。

表 7-11　　UPDATE 报文各域

域　名	含　义
撤销路由长度	2 字节的无符号整数，表示以字节计数的整个撤销路由域的长度。
撤销路由	长度是可变的。用二元组<长度，前缀>的形式来表示。长度是 1 个字节，指 IP 地址前缀的长度。前缀是可变长度，该域由 IP 地址前缀加上足够的填充位组成，使得该域大小为字节的倍数。撤销路由提供了那些不可行的，或不再服务的需要从 BGP 路由表中撤销的选路更新列表。
总的路径属性长度	2 字节的无符号整数，标识路径属性域的字节长度。
路径属性	此可变长字段包含了与网络层可达性信息字段中前缀相关联的 BGP 属性列表。路径属性给出了正被通告的前缀信息，如优先级或者前缀源等。这些信息用于过滤及路由选择过程。
网络层可达性信息 NLRI	长度是可变的，包含了网络可达信息的 IP 地址前缀列表。它的表示方法与前面的撤销路由的表示一样，以二元组<长度，前缀>的方式表示。长度范围指示出在 IP 地址前缀的比特长度。0 指示出这个前缀和所有的 IP 地址相匹配（包括前缀、它本身和 0 字节）。前缀范围包括 IP 地址前缀，这个 IP 地址前缀后面会跟着足够的尾随的比特，作为这个字节边界的结束范围。需要注意的是尾随的比特值是不相关的。

UPDATE 报文的最小长度为 23 字节，其中 19 字节为固定的报文头长度、2 字节为不可行路由长度、2 字节为总路径属性长度（不可行路由长度的值为 0，总路径长度值为 0）。一个 UPDATE 报文可以广播至少一条具有多个路径属性的路由。

一个 UPDATE 报文可以列出从服务器上撤回的多条路径。每一个这样的路由被它的目的地（由 IP 前缀表示）识别，即在 BGP 发言人的背景里确切地定义路由，BGP 发言人与之前广播的路由相连。也可以只广播从服务器上撤回的路由，它不包括路径属性或者网络层可到达信息。相反地，它可以只广播一条可行路由，即撤回路由范围不需要被提出。

（4）keepalive 报文

BGP 不用任何基于传输协议保持机制决定对等体是否可到达，而是利用 keepalive 报文使保持时钟不超时。keepalive 是周期性地在对等体间交换的报文，依靠它判断这些对等体之间的邻居关系是否仍然存在。keepalive 报文是 19 个字节的 BGP 报文头，后面没有数据。

keepalive 报文以保证在保持时间内不溢出的速率发送。建议的速率是保持时间间隔的三分之一，即 30 秒。如果保持时间间隔为零，则不用发送周期性的 keepalive 报文。keepalive 消息的发送频率不应该超过每两秒一次。

（5）notification 报文

任何时候检测到错误，BGP 就会发出 notification 报文，并且在发送出此报文后立即关闭相对应邻居之间的 BGP 连接。除了定长的 BGP 报文头，notification 报文包含其他的几个字段，如图 7-27 所示。

其中错误码是 1 字节的无符号整数，标识 notification 的类型。错误子码也是 1 字节的无符号整数，是对每个错误类型中更具体的错误记录。数据域根据错误码和错误子码而定，用来诊断发送 notification 报文的原因。

错误码定义如表 7-12 所示。

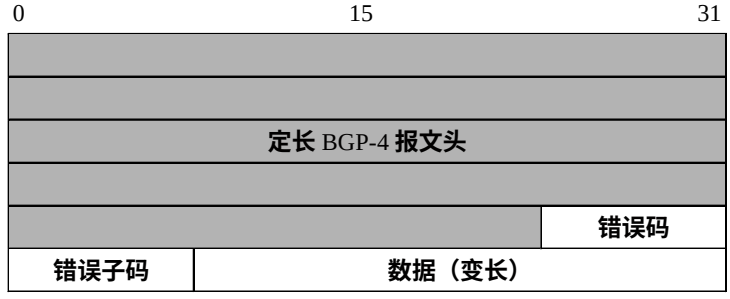


图 7-27　notification 报文格式

表 7-12　　notification 报文错误码

错　误　码	符 130 号 名 称。
1	报文头错误。
2	OPEN 报文错误。
3	UPDATE 报文错误。
4	保持定时器超时。
5	FSM 错误。
6	终止。

错误子码（Error subcode）给出了更具体的错误信息。每个错误码可以有一个或多个相关的错误子码。如表 7-13 所示。

表 7-13　　notification 报文错误子码

错　误　码	错 误 子 码	出 错 含 义
1：报文头错	1	连接未同步。
	2	错误报文长度。
	3	错误报文类型。
2：OPEN 报文错误	1	不支持的版本号。
	2	对端 AS 号错。
	3	非法的 BGP 标示符。
	4	不支持的选项参数。
	5	认证失败。
	6	不支持的保持时间长度。

续表

错 误 码	错 误 子 码	出 错 含 义
3：UPDATE 报文错误	1	畸形属性链表。
	2	不认识的公认属性。
	3	缺少公认属性。
	4	属性标志错。
	5	属性长度错。
	6	无效 ORIGION 属性。
	7	AS 路由环路。
	8	无效的 NEXT-HOP 属性。
	9	可选参数错。
	10	无效网络域。
	11	畸形 AS_PATH。
其它	无	无

变长的数据域用来诊断 notification 的原因。数据域内容依赖于具体的错误码和错误子码。

2．状态机

真正深入了解并掌握一个协议，必须对其各个状态之间的变化以及变化的条件非常熟悉。本节介绍 BGP-4 协议的状态机。以有限状态机（FSM）的方式描述 BGP 的操作，即通过 FSM 定义的状态，观察概括 BGP 的操作。需要指出的是，BGP 的 FSM 的各种状态及事件是针对每个 BGP 对等体来说的。

BGP-4 协议共有 6 种状态，13 种事件。分别列于表 7-14 和表 7-15。

表 7-14　　BGP-4 协议 6 种状态

序 号	状　态	含　义
1	Idle	BGP 最开始处于空闲状态。
2	Connect	BGP 正在等待传输层协议连接的完成。
3	Active	BGP 试图通过启动 TCP 连接获得一个对等体。
4	OpenSent	正等待它的对等体发来 OPEN 报文。此状态下，BGP 已经向该对等体发送过 OPEN 报文。
5	OpenConfirm	BGP 在收到 OPEN 报文后，等待对等体发送 keepalive 或 notification 报文。
6	Established	对等协商的最后阶段，在这个阶段，BGP 开始与对等体交换 UPDATE 数据包。

表 7-15　　BGP-4 协议 13 种事件

序 号	事　　件	含　　义
1	BGP Start	BGP 启动。
2	BGP Stop	BGP 停止。
3	BGP Transport connection open	BGP 传输连接打开。
4	BGP Transport connection closed	BGP 传输连接关闭。
5	BGP Transport connection open failed	BGP 传输连接打开失败。
6	BGP Transport fatal error	BGP 传输重大错误。
7	Connect Retry timer expired	Connect Retry 定时器过期。
8	Hold Timer expired	Hold 定时器过期。
9	Keep Alive timer expired	Keep alive 定时器过期。
10	Receive OPEN message	收到 OPEN 报文。
11	Receive keepalive message	收到 keepalive 报文。
12	Receive UPDATE messages	收到 UPDATE 报文。
13	Receive notification message	收到 notification 报文。

下面，对 BGP 状态机的各种状态之间的转换给予详细描述。

（1）Idle 状态（空闲状态）

最初 BGP 处于空闲状态。在该状态下，BGP 拒绝接受任何传输层连接请求，对对等体不分配任何资源。

当发生启动（Start）事件（由系统或操作员发起）时，本地系统初始化所有 BGP 资源、启动 Connect Retry 定时器、初始化到对等体的传输层连接，同时侦听远端的对等体是否主动和它建立连接。一旦侦听到连接请求，就变换它的状态为 Connect。Connect Retry 定时器的值由本地的策略决定如何配置，但要大到足够允许 TCP 初始化。

如果 BGP 发言人发现有错误，就关闭连接并回到 Idle 状态。BGP 通过等待启动事件的产生来脱离 Idle 状态，而对其他任何事件都将忽略。协议指出启动事件也可以自动产生，但是需要注意的是，为了避免持续的 BGP 错误造成的不良结果，对一个以前由于错误而回到 Idle 状态的对等体，不要立即产生启动事件，且连续产生启动事件的时间间隔应指数增加。协议建议初始时间值为 60 秒。

在空闲状态下收到的任何其他事件都将被忽略。

（2）Connect 状态（连接状态）

在这个状态下，BGP 正在等待传输层协议连接的完成。

如果 TCP 连接成功，本地系统取消 Connect Retry 定时器，完成初始化，并向它的对等体发送 OPEN 报文，同时转换到 Open Sent 状态。如果 TCP 连接失败，本地系统复位 Connect Retry 定时器，并继续侦听远端的 BGP 对等体是否已主动和它建立连接，同时转换为 Active

状态。如果 Connect Retry 定时器超时，状态停留在连接阶段，本地系统复位 Connect Retry 定时器，并启动一个与其它 BGP 对等体的传输连接。

如果发生其他由系统或操作员启动的事件，本地系统释放所有与该连接相关的 BGP 资源，并回到 Idle 状态。

（3）Active 状态（激活状态）

在这个状态下，BGP 试图通过启动 TCP 连接获得一个对等体。

如果连接成功，本地系统取消 Connect Retry 定时器，完成初始化，并向对等体发送 OPEN 报文，同时转换为 Open Sent 状态，设置它的保持定时器为一个比较大的值，建议为 4 分钟。

如果 Connect Retry 定时器超时，本地系统复位 Connect Retry 定时器，BGP 回到连接状态。BGP 也监视有可能由远端对等体启动的连接。如果本地系统探测到远端对等体正试图与它建立 BGP 连接，而且远端对等体的 IP 地址不是它预期的那个，本地系统重启动连接重试定时器，拒绝这个尝试连接，继续监听由远端 BGP 对等体发出的连接，保持激活状态。

对任何其它的事件（由系统或者操作者发出的）的反应，本地系统释放所有的与这个连接相关的 BGP 资源，改变自己的状态到空闲。

通常，如果对等体的状态在连接和激活之间来回地转变，就表明故障发生了：TCP 传输连接没有连通。这可能是因为许多 TCP 的重传或对等体不稳定，不能到达其对等体的 IP 地址。

启动事件在激活状态下被忽略。

（4）Open Sent 状态

在这个状态下，BGP 正等待它的对等体发来的 OPEN 报文。

当收到报文后，对其进行正确性检查。如果检查出错，例如不适合的版本号或不能接受的 AS，本地系统就发送 notification 报文，并回到空闲状态。如果没有出错，BGP 开始发送 keepalive 报文，并启动 Keep Alive 计时器。在这个阶段，还要协商两者的保持时间（Hold Time）并取较小值。如果协商的保持时间是零，那么不用启动 Hold 定时器和 Keep Alive 定时器。另外，通过与它的对等体比较 AS 号码，BGP 会识别对等体属于同一个 AS 还是不同的 AS。最后，状态变换为 Open Confirm 状态。

如果发现 TCP 传输连接断开，就回到激活状态。如果有其他差错，例如 Keep Alive 定时器超时，BGP 就发送一个带有相应错误代码的 notification 报文给对等体，并回到空闲状态。

对由系统或操作员启动的停止事件的响应，状态将回到空闲状态。无论何时 BGP 从状态 Open Sent 变换为空闲状态，都将关闭传输层连接，并释放所有与该连接有关的资源。如果保持定时器超时，本地系统发送带有错误码为保持定时器超时的 notification 报文，改变状态为空闲状态。

启动事件在此状态下被忽略。

（5）Open Confirm 状态

在这个状态下，BGP 等待 keepalive 或 notification 报文。

如果收到 keepalive 报文，就进入已建立（Established）状态，且对等体的有关协商也完成了。此后，若系统收到 UPDATE 或 keepalive 报文，它就重新启动 Hold 定时器（假设协商的保持时间不是零）。

若收到 notification 报文，状态回到空闲状态。系统以 Keep Alive 定时器所设置的速率周

期地发送 Keep Alive 报文。如果 Keep Alive 定时器超时，本地系统发送 keepalive 报文，并重新启动 Keep Alive 定时器。如果保持定时器在收到 keepalive 消息之前超时，本地系统发送 notification 报文，改变自己的状态到空闲状态。

如果从下面的传输协议收到一个断开连接通知，本地系统改变自己的状态到空闲状态。对停止事件的反应（由系统或者操作者发出），本地系统发送带有错误码为终止的 notification 报文，且改变状态为空闲状态。作为对其他事件的响应，系统会发送带有 FSM 错误码的 notification 报文并回到空闲状态。

启动事件在本状态下被忽略。

（6）Established 状态（已建立）

这是对等体协商的最后阶段。在这个阶段，BGP 可以与它的对等体交换 UPDATE、notification、keepalive 报文。对正常的连接来说，主要是交换 UPDATE 数据报文和 keepalive 报文。

如果 Hold 定时器的值非零，本地系统在收到 UPDATE 或 keepalive 报文时，就重新启动 Hold 定时器。若系统收到任何 notification 报文，即发生一些差错，状态就回到空闲状态。

如果 keepalive 定时器超时，本地系统发送一条 keepalive 消息，并且重启动它的 keepalive 定时器。每次本地系统发送 keepalive 或者更新消息，就重启动它的 keepalive 定时器，除非协商保持时间值为 0。

收到 UPDATE 报文后，要检查报文的差错。如果发现差错，就向对等体发送 notification 报文并回到空闲状态。如果 Hold 定时器超时，或者从传输协议收到中断通知，或是收到停止事件，或者响应其他事件，系统也回到空闲状态。

对任何其他事件的反应，本地系统发送带有错误码限定状态机制的通知消息，并且改变状态到空闲状态。只要 BGP 的状态由确定到空闲，它关闭 BGP（和传输标准）连接，释放与连接相关的所有资源，删除从连接起源的所有路由。

启动事件在确定的状态下被忽略。

3．BGP 更新报文处理

本小节基于前面提到的关于路由信息库的三个概念模型以及状态机的转换，介绍不同情形下 BGP-4 协议对路由更新报文的处理，主要包括路由决策过程、更新报文的发送、选路规则等。

更新报文仅在 Established 状态下接收到才是合法的。当接收到一条更新报文，BGP 首先检查其每一个字段的有效性。如果可选的无传输属性是不可识别的，它将被完全忽略。如果可选的传输属性是不可识别的，部分在属性标记字节中的标志位（第三个高位序比特）设置为 1，属性为传播给其他 BGP 运行者保持。

如果可选属性是可识别的，且有一个有效值，那么，依赖于可选属性类型，它可能被局部处理、保持或者更新。如果更新报文的撤消路由字段非空，则之前它通告的目的地（表示为 IP 地址的头，包含在这个字段里的路由）将从 Adj-RIB-In 移走。这个 BGP 发言人将运行它的决策程序，通知以前通告的路由不再允许使用。

若更新消息包含了可行路由，它将被列在 Adj-RIB-In 属性中，并将采取下面的行为：

■　如果网络层可到达信息（NLRI）与当前储存在 Adj-RIB-In 中的一条路由相同，那么在 Adj-RIB-In 中新路由将会取代旧路由，这样就隐含地从服务器上撤回了旧路由。当旧路

由不再允许使用，BGP 运行者将运行决策进程。

- 如果新路由是一条重叠路由（包含在 Adj-RIB-In 中已经存在的路由中），BGP 运行者将运行决策进程，因为更详细的路由（More Specific）隐含地使特殊路由（Less Specific）的一部分变为不可用。
- 如果新路由与包含在 Adj-RIB-In 中的路由有相同的路径属性，并且比较早的路由更详细，则不需要进一步的行为。
- 如果新路由含有的网络层可到达信息（NLRI），不在 Adj-RIB-In 中任何路由中存在，那么新路由将放在 Adj-RIB-In 中，同时 BGP 运行它的决策进程。
- 如果新路由是一个比包含在 Adj-RIB-In 中某一路由更加特殊的重叠路由， BGP 将通过此路由运行设置目的地的决策进程。

（1）决策过程

在广播选择路由以前，BGP 根据本地策略信息库中所定义的各种策略对存储在 Adj-RIB-In 中的路由进行选择。决策过程的输出为一些将通告给所有对等体的路由。已选择的路由将被存储在本地的 Adj-RIB-Out 信息库。

选择过程通过定义一个选择函数实现，选择函数根据给定路由的属性作为选择依据，并且返回一个非负整型数，这个整型数表示路由的优先程度。决策进程对包含在 Adj-RIB-In 各条路由进行操作，主要有：

- 选择广播给 IBGP 邻居的路由。
- 选择广播给 EBGP 邻居的路由。
- 路由聚合和路由信息缩减。

决策进程发生在三个独立的阶段，每一个阶段都由不同的事件引发：

阶段 1：负责计算每一个从 EBGP 邻居接收到的路由的优先程度，并且可能会把那些到特定目的地的具有最高优先程度的路由通告给本地自治系统中其它的 BGP 运行者。

阶段 2：阶段 1 的工作完成以后激活阶段 2，它为每一个目的地选择所有可用路由中最佳的路由，并且把已选路由存放在相应的 LOC_RIB 中。

阶段 3：当 Loc_RIB 被修改时激活阶段 3，根据 RIB 中的策略，阶段 3 负责散布 Loc_RIB 中的各个路由到每一个 EBGP 邻居。路由聚合和信息缩减可以在这个阶段选择是否进行。

（2）阶段 1：计算路由优先级

本地 BGP 发言人接收到从 EBGP 对等体发出的 UPDATE 报文（这个 UPDATE 报文可能包含新路由、替代路由或者撤销路由），阶段 1 的进程就会被激活。阶段 1 决策功能是一个独立的进程，当它没有进一步的工作做的时候就自动结束。在操作包含在 Adj-RIB-In 里面的任何路由之前，进程函数将锁上 Adj-RIB-In，一旦操作进行完毕，解锁 Adj-RIB-In。

对接收到的每一个新路由或者替代路由，本地 BGP 运行者将决定该路由的优先级。如果路由是从本地自治系统中 BGP 运行者得到的，LOCAL_PREF 属性值将被作为优先程度。如果路由是从相邻自治系统的 BGP 运行者得到的，那么将在预配置策略信息基础上计算优先级，并且向 IBGP 广播这些路由时，将计算出来的优先级当作 LOCAL_PREF 属性值。其中，策略确切含义和相关的计算都是本地事件。

（3）阶段 2：路由选择

阶段 1 的工作完成以后激活阶段 2。

阶段 2 功能是独立的进程，既没有进一步的工作时完成。阶段 2 进程将考虑出现在 Adj-RIBs-In 中的所有路由，包括那些从位于相邻自治系统的 BGP 运行者和本自治系统中的 BGP 运行者接收到的。一旦阶段 3 决策进程被调用，阶段 2 决策进程从运行状态被阻塞。在操作包含在 Adj-RIB-In 里面的任何路由之前，进程函数将锁上 Adj-RIB-In，一旦操作进行完毕，解锁 Adj-RIB-In。如果 BGP 路由下一跳属性描述一个地址给在 Loc-RIB 中没有路由的本地 BGP 运行者，BGP 路由会被排除在阶段 2 决策功能之外。

BGP 执行选路规则的决策过程主要是基于各种属性值。当面对到同一目的地的多个路由时，BGP 为了保证数据量送到目的地，将选择最好的路由。主要是根据以下规则选择最好的路由：

- 选择本地优先级值最大的路由，如果下一跳不能到达的，这个路由将要忽略掉（所以要有一个到下一跳的 IGP 路由）。
- 如果路由的本地优先级相同，优选本地始发的路由。
- 如果本地优先级相同，优选具有最短 AS 路径的路由。
- 如果 AS 路径长度相同，优选具有较低起点类型的路由（其中 IGP 低于 EGP，EGP 低于 INCOMPLETE）。
- 如果起点类型相同，优选具有最低 MED 的路由。
- 如果路由具有相同的 MED，优选 AS 中最短内部路径到达目的地的路由（以最短路径到 BGP 的 NEXT_HOP）。
- 如果内部路径相同，BGP 路由器的 ID 号可以打破平衡，优选从 BGP 路由器来的具有最低路由器 ID 的路由。

（4）阶段 3：路由分散

阶段 2 的工作完成以后或者发生以下事件之一的时候激活阶段 3:

- Adj-RIB-In 到本地某一目的地的路由被修改。
- 学到新路由。
- 建立起一个新的 BGP 连接。

阶段 3 决策函数也是一个独立的进程，当它没有进一步的工作做的时候就自动结束。

为了更好地支持未来的 AS 间的组播能力，参加 AS 间组播路由的 BGP 发言人应该通告它从外部对等体收到的路由，同时，如果该路由在 Loc-RIB 内，还应该重新把该路由通告到接收路由的对等体。对没有参加 AS 间组播路由的 BGP 发言人，这个通告是可选的。如果做这样一个通告，NEXT_HOP 属性应该设置为对等体地址。

应用也可以优化这个通告，组合 AS_PATH 属性的信息不但包括自己的 AS 号而且包括通告路由的对等体的 AS 号（这个组合要求 ORIGIN 属性被设置为 Incomplete）。另外，在这种通告中，应用不需要传递可选的或者自决的路径属性。

一旦 Adj-RIBs-Out 和转发信息库（FIB）被更新，本地 BGP 发言人就应该开始运行外部更新进程。

（5）重叠路由

BGP 发言人可以传送具有重叠网络层可达信息（NLRI）的路由到别的 BGP 发言人。NLRI 重叠产生于非匹配的多个路由中一些目的地相似的路由中。由于 BGP 使用 IP 前缀对 NLRI 编码，重叠一般要展示子网关系。路由描述了更小范围的目的地（更长的前缀）称为更特别

路由，路由描述了更大范围的目的地（更短的前缀）成为更一般路由，反之同样。

这种优先关系把更一般的路由有效地分解为两部分：

- 仅仅使用更一般路由描述的一组目的地。
- 使用更一般和更特殊路由的重叠描述的一组目的地。

当重叠路由发生在同样的 Adj-RIB-In，更特殊的路由应该有优先权，顺序是更特殊到更一般。 重叠描述的目的地的集合表明一部分更一般路由是可用的，但是当前不可用。如果一个更特殊的路由后来撤销了，重叠描述的目的地的集合将可以使用更一般的路由到达。

如果 BGP 发言人接收了重叠路由，决策过程应该考虑重叠路由的语义。特别是，如果 BGP 发言人接收了同一个对端的更一般的路由同时拒绝了更特殊的路由，那末重叠表示的目的地可能不转发到路由的 AS-PATH 属性列出的ＡＳ那里。因此，BGP 发言人可以进行下面的选择：

1：只安装更一般和更特殊的路由

2：只安装更特殊的路由

3：只安装更一般路由的非重叠部分（这意味着解聚合）

4：聚合着两条路由同时安装聚合路由

5：安装更一般的路由

6：都不安装

如果一个 BGP 发言人选择 5，应该把 ATOMIC-AGGREGATE 属性加到路由中。承载 ATOMIC-AGGREGATE 属性的路由不能被解聚合。也就是说，路由的 NLRI 不能被更特殊化。向这个路由转发不保证 IP 包实际沿着路由的 AS-PATH 属性列出的 AS 路经转发。如果 BGP 发言人选择 1，必须不在通告更特殊路由时通告更一般的路由。

4．Update 发送

Update 发送进程负责广播 Update 报文到所有的对等体。例如，发布决策进程选择的路由到 IBGP 或者 EBGP 发言人。如果所有 BGP 发言人在同一个自治系统，则在这些 BGP 发言人之间路由信息的发布，被看作内部发布。

（1）内部更新

内部更新进程是发布路由信息到本地自治系统的 BGP 发言人。当 BGP 发言人从在本地自治系统别的 BGP 发言人收到 UPDATE 报文，接收 BGP 发言人不应该再分配 UPDATE 报文中的路由信息到本地自治系统内的别的 BGP 发言人。当 BGP 发言人接收了邻居自治系统的 BGP 发言人的一条新的路由，如果下面的情况之一发生，应该使用 UPDATE 报文广播路由到本地自治系统的所有的 BGP 发言人：

- 本地 BGP 发言人安排给新接收路由的优先程度高于本地发言人已经安排的接受自邻居自治系统的别的路由的优先级，或者
- 没有接收到邻居自治系统别的 BGP 发言人发送的路由
- 当 BGP 发言人收到了 UPDATE 报文，有非空的撤销路由域，应该从 Adj-RIB-In 里去掉所有在这个域中（IP 前缀表示）的目的地。同时做下面的附加行为：
- 如果相应的可用路由先前没有被广播，不需要做更多的行动。
- 如果相应的可用路由先前被广播，则：

① 如果被选择广播的新路由和不可用路由有相同的网络层可达信息，本地系统广播替

代路由。

② 如果不能广播替代路由，BGP 发言人应该在 UPDATE 报文的 WITHDRAWN ROUTES 域填入不可用路由的目的地（IP 前缀形式），同时把这个报文发送到先前广播了相应的可用路由的对等体。

所有的广播过的可用路由放入相应的 Adj-RIBs-Out 信息库中，所有不可用但是广播过的路由应该从 Adj-RIBs-Out 清除。

如果一个本地 BGP 发言人连接到邻居 AS 的几个 BGP 发言人，则有多个 Adj-RIBs-In 和这些对等体相关联。这些 Adj-RIB-In 可以包括到同一个目的地的多个相等优先级的路由，所有路由要广播到邻居自治系统。本地 BGP 发言人根据下面的法则选择其中一条路由：

■ 如果候选路由的 NEXT-HOP 和 MULTI-EXIT-DISC 属性不同，本地系统配置中考虑了 MULTI-EXIT-DISC 属性，选择有最低的 MULTI-EXIT-DISC 属性的路由。

■ 如果本地系统能够确定候选路由中 Next-Hop 属性描述的实体的路径的成本，选择成本较低的路由。

■ 在所有别的情况中，选择源 BGP 发言人具有较低 BGP 标识符的路由。

（2）外部更新

外部更新过程和邻居 AS 的 BGP 发言人路由信息的发布有关。作为阶段 3 选择过程的一部分，BGP 发言人更新它的 Adj-RIBs-Out 信息库和转发表。所有新安装的路由和所有新的但是没有替代路由的不可用路由都要通过 UPDATE 报文广播到邻居 AS 内的 BGP 发言人。

任何位于 LOC-RIB 的路由，如果是不可用的路由就应该被撤销。在自己的 AS 内，如果可达地址改变，应该发送 UPDATE 报文。

（3）控制路由流量开销

BGP 协议限制路由流量（也就是 UPDATE 报文），目的是减少广播 UPDATE 报文的带宽和增强 UPDATE 报文信息决策过程的处理能力。

① 路由广播的频率

参数 MinRouteAdvertisementInterval 确定 BGP 发言人广播到特定目的地的两个路由之间的最小时间。这个速率限制过程基于单个目的地，但是 MinRouteAdvertisementInterval 值是对每一个对等体设置的。

从单个 BGP 发言人广播可用路由到目的地集合的两个 UPDATE 报文必须至少按 MinRouteAdvertisementInterval 的值分开。无疑，只有精确保持目的地集合的计时器才能做到这一点。显然，这是不现实的开销。

由于需要在 AS 内部快速收敛，本过程不能用于从本 AS 中别的 BGP 发言人发来的路由。为了避免永久黑洞，本过程也不能用于明确撤销或者不可用路由（也就是，通过 IP 前缀表示的目的地在 UPDATE 报文中 WITHDRAWN ROUTES 域的路由）。

这个过程不限制路由选择的速率，只是限制路由广播的速率。当等待 MinRouteAdvertisementInterval 超时时，如果新的路由被选择多次，在 MinRouteAdvertisementInterval 的最后选择的路由将被广播。

② 路由产生速率

参数 MinASOriginationInterval 确定 BGP 发言人在自己的 AS 内，报告变化的 UPDATE 报文连续广播的最小间隔。

③ Jitter

为了减少给定 BGP 发言人的报文产生尖峰，jitter 应该同 MinASOriginationInterval、Keepalive、MinRouteAdvertisementInterval 计时器联系起来。无论 UPDATE 被送到哪个目的地，给定 BGP 发言人都应该应用同样的 jitter 数量。也就是说，jitter 不能被基于“每个对等体”使用。

（4）路由信息的有效组织

选择将要广播的路由信息，一个 BGP 发言人可以采用几个方法，组织信息为一个有效形式。信息简约是其中一种。信息简约可以在策略控制的程度上使用简约——在信息崩溃之后，在等价类上使用相同的策略。

决策进程使用下面可选的方法减少放在 Adj-RIB-Out 中的信息数量：

■ 网络层可达信息（NLRI）：目的 IP 地址可以被看作 IP 地址前缀。若地址结构和在 AS 管理者控制下的系统能够达成一致，有可能减少 UPDATE 报文中 NLRI 的尺寸。

■ AS_PATHs（AS 路径）：AS 路径信息可以表达为顺序 AS-SEQUENCE 和无序 AS-SET。AS-SET 可以用在路由聚合算法中，它通过把无论在聚合 AS-PATH 上出现几次的 AS 号只列出一次，减少了 AS-PATH 的尺寸。

一个 AS-SET 意味着 NLRI 列出的目的地能够通过由 AS-SET 中的部分 AS 组成的路径到达。AS-SET 提供有效的信息来避免路由环；然而使用它们也可能会剪除一些潜在的有用路径，原因是一些路径不再被单独通过 AS-SEQUENCE 方式列出。实际使用中这不是个问题，因为一旦 IP 报文到达 AS 组的边界，这个点上的 BGP 发言人更可能有更多的详细路径信息区分到目的地的路径。

5．错误处理

这里介绍在处理 BGP 协议的各种报文时，检测到差错是如何处理的。当检测到后面提到的出错情形时，系统就发送带有错误码、错误子码以及数据域的 notification 报文，并关闭对等体间的 BGP 连接（同时 TCP 连接中断，且释放所有相关资源）。如果没有规定相应的错误子码，用 0 代替。除非特别说明，一般 notification 报文的数据域为空。

（1）报文头错处理

处理报文头部时，一旦检测到任何错误就会发送错误码为头部错的 notification 报文，错误子码根据具体的情况填写：

■ 如果报文是 OPEN 报文，则报头的 Marker 域的值应该为全 1。根据 BGP 的 OPEN 报文中是否传送了认证信息以及具体的认证机制，所有其他类型的 BGP 报文 Marker 域由认证信息可选参数而定。

■ 如果报头的 Marker 域不是所要的，那么产生同步错误，错误子代码设置为 1(连接不同步错误)。

■ 如果报文的长度不符合协议规定，错误子代码为 2(错误报文长度)，数据域记录错误长度。具体包括：报文头的长度字段少于 19 或者大于 4096，或者 OPEN 报文的长度字段少于 OPEN 报文的最小长度，或者更新报文的长度字段少于更新报文的最小长度，或者 keepalive 报文的长度字段不等于 19，或者 notification 报文长度少于 notification 报文的最小长度。

■ 如果报头的类型是不可知的，错误子代码为 3(错误报文类型)，数据域记录错误类型。

（2）OPEN 报文错误处理

处理 OPEN 报文时，一旦检测到任何错误就会发送错误码为 OPEN 报文错的 notification 报文，错误子码根据具体的情况填写：

■　如果收到的 OPEN 报文中的版本号本地系统不支持，则错误子代码置为 1（不支持的版本号），数据域是一个 2 字节的无符号整数，记录本地系统所能支持的最高版本号（当然此版本号比远端对等体提出的小）。

■　如果 OPEN 报文标识的自治系统号本地系统不接受，则错误子代码为 2（错误的对等体 AS 号）。

■　如果 OPEN 报文的保持时间定时器的值不能接受，则错误子代码置为 6（不可接受的保持时间），一般拒绝 1～2 秒的保持时间值。

■　如果 OPEN 报文的 BGP 标识的构成是错误的（即 BGP 的标识域是一个无效的 IP 主机地址），则错误子代码置为 3（错误的 BGP 标识）。

■　如果 OPEN 报文中的可选参数是不可知的，错误代码置为 4（不支持的可选参数）。

■　如果 OPEN 报文带有认证信息（例如一个可选参数），那么认证过程失败时，错误子代码为 5（认证失败）。

（3）UPDATE 报文错误处理

处理 UPDATE 报文时，一旦检测到任何错误就会发送错误码为 UPDATE 报文错的 notification 报文，错误子码根据具体的情况填写，UPDATE 报文的错误检测主要是检测路径属性：

■　如果 UPDATE 报文不可行路由长度或整个属性长度太大，使得整个报文长度超过 4096，或者报文中一个属性多次出现，那么错误子代码为 1（畸形的属性列表）。

■　如果属性标志与属性类型代码不符合，错误代码置为 4（属性标志错误），数据域记录错误的属性三元组（类型、长度、数值）。

■　如果任何可知属性的属性长度不符合规定，错误子代码为 5（属性长度错误），数据域记录错误的属性三元组（类型、长度、数值）。

■　如果任何一个公认必遵的属性不存在，错误子代码为 3（公认属性丢失），数据域记录该丢失的公认属性的属性类型代码。

■　如果 UPDATE 报文中的公认必遵属性不可识，错误子代码为 2（不可被识别的公认必遵属性），数据域记录不可识的属性三元组（类型、长度、数值）。

■　如果 ORIGIN 属性含有未定义的值，错误子代码为 6（无效的 ORIGIN 属性），数据域记录未知的属性三元组（类型、长度、数值）。

■　如果 NEXT_HOP 属性域的构成是错误的（即标识了一个无效的 IP 地址），错误子代码设为 8（无效的下一跳属性），数据域记录错误的属性三元组（类型、长度、数值）。

■　如果发生了无效的 NEXT_HOP 属性错误，此类错误应记入日志，此时的路由信息可以忽略，且不用发送 notification 报文。

■　如果 AS_PATH 属性域或 NLRI 域构成错误，错误子代码分别为 11（异常的 AS_PATH）和 10（无效的网络域）。

■　如果在检测一个可知的可选属性值时发现差错，那么丢掉该属性，错误子代码为 9（可选属性错误），数据域记录该属性的三元组（类型、长度、数值）。

（4）notification 报文错误处理

对等体发送 notification 报文时，也可能发生错误，遗憾的是协议不能提供用“子”notification 报文报告该类错误。

但是一旦此类错误发生，任何不可识的错误代码和子代码都应当进行日志记录。具体方法和各种实现有关。

（5）FSM(有限状态机)错误处理

任何情况下检测到 FSM 错误，设定错误代码为 FSM 出错。

（6）HoldTimer（定时器）超时错误处理

若系统在规定的保持时间内,不能连续收到 keepalive 报文、UPDATE 报文或者 notification 报文，则 Hold 定时器超时，错误代码设为 Hold 定时器超时，并发送相应的 notification 报文，然后关闭 BGP 连接。

（7）停止

即使不发生上面提到的各种致命错误，BGP 发言人也可以在任何给定时间关闭 BGP 连接，这时需发送带有错误代码为停止的 notification 报文。

（8）连接冲突检测

当一对 BGP 发言人同时试图向对方主动建立 TCP 连接时，这对对等体之间便产生了两个并行的连接，这种情形我们称为连接冲突。显然，此时必须关闭其中一方主动发起的连接。BGP-4 协议就这种冲突提出了协商办法：比较冲突中的对等体的 BGP 标识值，保留由较大标识值的 BGP 发言人发起的连接。比较标识值时，按照 4 字节的无符号整数处理。

7.4.4 BGP 协议路由策略

作为一种域间路由协议，BGP 有非常丰富的路由策略，这些策略使 BGP 协议变得更加灵活。本小节介绍其中的几种主要路由策略，如 BGP 自治系统联邦、BGP 团体、反射体以及路由抖动的消除。

1．BGP 自治系统联邦

许多 Internet 服务提供者（Internet Service Provider, ISP）在其 AS 内设有大型的 IBGP 网络配置。减少 IBGP 连接的方法是把一个 AS 分成多个子 AS，然后把它们组成一个自治系统联盟。对外界而言，联盟看起来就像一个 AS。在联盟内部，每一个子 AS 内是全连接的，并且与同一联盟中的其他子 AS 也有连接。即使在不同子 AS 的对等体之间有 EBGP 会话，它们仍像 IBGP 对等体一样交换路由选择信息。具体地讲，是保存下一跳、MED 和本地优先信息。这些多个小的自治系统也可以组合在一起作为一个分配给自己的单一联邦使用。每一个新的自治系统具有一个 IBGP 网格，并且与联邦中其它的新自治系统相互连接。

因为它们属于同一个联邦，在新自治系统之间对等的 EBGP 可以交换它们的路由更新报文，就像它们使用 IBGP 进行连接一样。这意味着即使通常的 EBGP 连接不能传播该信息，下一跳、度量和本地优先权信息也可以在该联邦的 EBGP 之间进行传送。从各网络到联邦的连接则使用联邦 AS 号，而不管物理连接如何。

2．BGP 团体

BGP-4 的另一种扩展机制称为 BGP 团体，这种扩展机制能使 BGP 向它的直连邻居和远端邻居传递额外的信息，提出这种机制的初衷是为了减轻策略管理和 Internet 维护的复杂性。

BGP 通过控制路由信息的分布支持传输策略，这种控制机制在 RFC 1771 中有详细的描

述，并且在现实的网络中也已经得到了非常成功的应用。然而这种路由信息分布控制机制是基于 IP 地址前缀或者 AS-PATH 属性值的。为了简化路由信息分布的控制，IETF 提出了一个目的地址组的概念，从而使路由决策也能基于一个组的标识，这个概念大大简化了 BGP 关于控制路由信息分布的配置。

先引入团体这一概念：所谓团体是指一组在某些方面存在共同属性的目的地址。路由决策和其它 BGP 属性能够用于属于该团体的所有目的网络，每个自治系统管理者都会定义好一个目的地址属于哪个团体。缺省情况是，所有目的地都是属于大的 Internet 这个团体。

团体机制通过定义一个变长的可选过渡 COMMUNITIES 路径属性实现团体功能。这个属性由一个 4 字节的值的集合组成，每一个值定义一个团体，所有属于这个团体的路由都在属性列表中列出。COMMUNITIES 属性的类型号是 8。

团体被看作一个 32 位的值，值的范围是从 1 到 4294967200，或者是关键字 no-export 或 no-advertise 或 NO-EXPORT-SUBCONFED。对于管理者来说可能会做如下假设：从 0X00000000 到 0X0000FFFF 以及 0XFFFF0000 到 0XFFFFFFFF 为保留团体值。

剩余的团体值按一定规则编码：前两个字节为自治系统号，后两个字节的意义由自治系统相关含义定义（例如：自治系统 690 可能定义科研、教育和商业团体值，用来进行策略路由，AS 管理者就会用团体值 0X02B00000 到 OX02B2FFFF）。

有一些团体属性值具有全局意义，它们的操作应该在任何支持团体属性的 BGP 协议中实现，这些属性主要有：

■　NO-EXPORT（0XFFFFFF01）：收到的任何携带这个团体属性值的路由都不能广播到一个 BGP 联邦外面（一个独立的不属于任何联邦的自治系统本身被看作一个联邦）。

■　NO-ADVERTISE(0XFFFFFF02)：收到的任何携带这个团体属性值的路由都不能广播给其它邻居。

■　NO-EXPORT-SUBCONFED(0XFFFFFF03)：收到的任何携带这个团体属性值的路由都不能广播给外部对等体（包括同一个联邦内的其他自治系统内的对等体）。

BGP 发言人根据团体属性可以控制是否向其他邻居广播它收到的某些路由信息。如果收到的路由中没有这个属性，那么在广播给其它邻居时有可能再把这个属性加上。根据本地策略 ，BGP 发言人也有可能会对收到的路由中的团体属性值进行修改。

3．反射体

BGP-4 不允许 IBGP 对等体将那些其它 IBGP 对等体得知的已公布路由转发到第三方 IBGP 对等体。在一个单一自治系统中，所有 BGP 发言者必须全互连，并且任意外部路由信息必须被重新广播到此自治系统内的所有其他的 BGP 路由器。对于一个有 n 个 BGP 发言人的自治系统而言，需要维护 n*(n-1)/2 个独立的 IBGP 链接。因为当前 Internet 网络通常存在大量的 IBGP 发言者，每个发言者间相互交换大容量的路由信息，所以“全互连”的要求明显不现实，而且这对网络的可扩展性不是很好。路由反射器是另一种减少 IBGP 连接的方法。

路由反射器的内部对等体被分成两组：客户对等体和 AS 中的所有其它路由器（非客户对等体）。路由反射器反射两组之间的路由，该路由反射器及其客户对等体形成一个簇。非客户对等体必须是彼此全连接的，但客户对等体不必是全连接的。簇中的客户不同簇外的 IBGP 会话者通信。

当路由反射器收到路由信息时，它就完成以下的任务：

- 广播来自外部 BGP 会话者的路由到所有客户和非客户对等体。
- 广播来自非客户的路由到所有客户。
- 广播来自客户的路由到所有客户和非客户对等体。因此，客户对等体不必是全连接的。

提供已得知的路由给 AS 内部其它 IBGP 路由器的 IBGP 对等体被称为路由反射器(Route Reflector, RR)。作为一个路由反射器的 IBGP 路由器执行类似于一个 hub and spoke 配置的功能。使用 RR 特性的路由器与 AS 中所有的路由器都有一个对等连接，并且更新它们的路由表。非 RR 路由器只与 RR 路由器具有对等连接，如图 7-28 中所示。

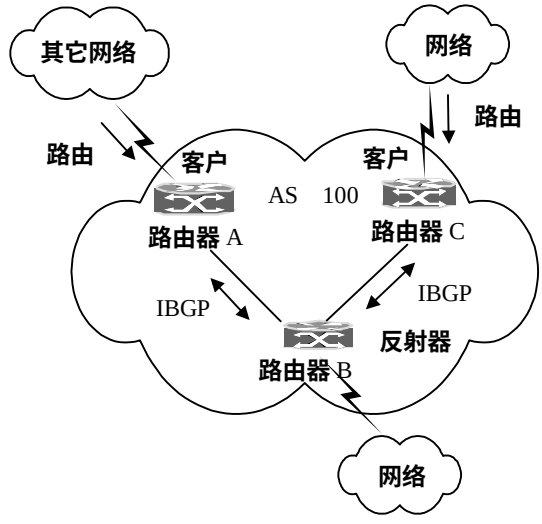


图 7-28 BGP 反射体拓扑结构

在这个简单的配置中，因为路由器 B 执行 RR 功能，因此不需要连接 AS 100 中的路由器 A、B 和 C 的完全 IBGP 网格。路由器 A 和 C 彼此之间并不建立对等关系。而是它们都只从路由器 B 获知路由。路由器 B 将从 A 接收的路由更新报文“反射”到 C 以及将更新报文从 C 反射到 A。作为 RR 的路由器将那些要从其上接收更新报文的邻居作为客户机。RR 及其客户机的组合称为一个簇（cluster)。

在图 7-28 中，路由器 A、B 和 C 形成一个簇，其中只有路由器 B 作为 AS 中的单一 RR。RR 的没有被定义为客户机的 IBGP 对等称为非客户机（non-client)。

在一个 AS 中可以有多个 RR，并且如果存在多个 RR，则就存在多个簇，一个 RR 与其它的 RR 进行通信，就像它们是 IBGP 对等体一样。多个 RR 可以在同一个簇中定义也可以到其它的簇中定义。

当一个 RR 从一个非客户机对等体接收一个路由，它将该路由反射给本簇中的所有客户机。如果 RR 接收到一个来自客户机对等体的路由，则该路由被反射到所有非客户机对等体和客户机对等体中。最后，如果 RR 所接收的路由是来自一个外部 BGP 对等体的，则 RR 将该更新报文送到所有客户机和非客户机对等体中。

由于路由被反射，已得知的内部 BGP 路由可能会引起一个路由信息环。用于路由反射的模式使用两种方法来避免一个路由信息环。

第一种方法是使用 originator-id 可选属性，该属性的长度是 4 个字节并且由 RR 创建。该属性值实际上是本地 AS 中路由源发站的路由器标识（route-id）。这使得路由源发站能够确定路由是否已经返回到自己，从而使得路由源发站能够忽略该路由更新报文。

第二种路由反射的方法是使用簇列表。簇列表是一个非传递的（non-transitive）BGP 可选属性，也是一个 cluster-id 序列，路由直通到该序列中。当 RR 将一个路由从一个客户机反射到一个非客户机对等体时，它会将自己的本地簇 ID 添加到簇列表的尾部。

使用簇列表并通过在路由公布信息的簇列表中查找自己的簇 ID，RR 能够确定环是否存在。如果 RR 在簇列表中找到自己的簇 ID，则该公布信息被忽略。如果簇列表是空的，则 RR 创建一个新的簇列表。cluster-id 变量值是由 RR 分配给该簇的 ID。该值最多可达 4 个字节。在使用冗余以避免一个簇的 RR 的单点失效时，cluster-id 是很重要的。通过在一个簇中定义多个 RR 提供冗余。簇中的每个 RR 必须使用相同的 cluster-id 以便利用 cluster-id 方法避免环。

如果簇的客户机是连通的，则不需要路由反射。缺省地，一个 RR 启用客户机到客户机的反射。重要的一点是，注意如果启用客户机到客户机的反射则不能使用对等组。一个路由反射器的任何客户从而就不能成为一个 BGP 对等组的成员。使客户机作为一个对等组的一部分将会导致无效路由被反射到那些不是对等组的一部分的客户机上。

当使用多个 RR 并且避免路由环时，必须查看两个特性。第一个是 set clause 命令。当该命令被指定用于出站路由映射时，它不会影响到那些被 RR 反射到 IBGP 对等方的路由。第二个特性是 RR 的自身下一跳（next-hop-self）特性。在一个 RR 路由器上使用自身下一跳只影响已得知的 EBGP 路由的下一跳，因为 RR 的下一跳没有发生变化。

4．路由抖动消除

网络稳定性是 Internet 的一个重要性能参数，是 QoS 的一个重要指标，也是用户和商家都比较关心的一个话题。在最近的十几年中，Internet 有了飞速的发展，从原来的实验性、学术性网络发展成为一个遍及全球的、广泛使用的公共通信系统的基础结构。过去，Internet 的失效影响的可能只是为数不多的科学研究人员和计算机专家。现在，Internet 一旦失效，将会给成千上万的用户的工作和学习造成影响，带来难以统计的损失。由于 Internet 的规模庞大、结构复杂，而且随时随地都在发生变化，Internet 有很多潜在的不稳定性因素。影响 Internet 稳定性的因素不仅包括高层的通信交换系统，而且包括低层的软硬件组件，特别是其中的报文转发技术和路由结构。

路由不稳定性（routing instability）的形式化定义是网络可达性和拓扑信息的快速变化。导致路由不稳定性的原因有很多，包括路由器设置错误、物理传输和数据链路问题以及软件 bug。所有的这些路由不稳定的原因会导致大量的路由更新发往 Internet 核心交换路由器。进而，这种不稳定能够沿着路由器传播到整个网络。

不稳定路由会严重影响网络性能，每当连接在网络中的某处发生故障或者不利于性能发挥时，就会使路由被反复重新计算，而后传播到整个网络。BGP 路由的这种反复快速广播和撤销的情景称为路由抖动。路由抖动对网络来说是致命的陷阱，它使网络设备特别是路由器处于连续的高负载工作状态，增加了网络的拥塞，大大浪费带宽。IETF 定义了一种机制以保护 BGP 不会产生不稳定路由，称为抖动路由消除机制。

为了限制路由的不稳定性，许多厂商在他们的路由器中实现了路由阻尼算法。这种算法

在一定时间内拒绝相信关于具有特定不稳定参数的路由更新。路由器将不处理受阻的路由更新直到参数重新设置或者设置的周期超时。大多数阻尼算法的实现技术还提供一定的机制来防止阻尼冲突。当然，路由阻尼算法并不是万能的，有其自身的和实际实现的缺陷，这里就不再详细讨论。

实际情况下，在可用和不可用之间摇摆不定的路由认为是一个抖动路由（flapping route）。当一个路由首次摇摆不定时，BGP 进程将会为该路由置一个历史状态（history）。一旦路由被置一个历史状态，该路由就被认为不是到达目的地的最佳路由。连续的路由抖动使 BGP 进程在该路由上设置一个损失值（penalty），损失值从 1000 开始并且是累积的。损失值存储在 BGP 中，在有问题的路由上作路由巡回直到出现损失。在一个半生存周期（half-life period）之后，损失值减半。这个过程会每 5 秒钟执行一次并且进行调整，以便尽量允许路由将它自己作为一个稳定路由再次建立。

经过一段时间，在路由上设置的累积损失可能会大于一个指定的抑制极限（suppress limit），抑制极限是权衡一个路由的损失的累积总量。一旦抑制极限被超过，路由的状态就会由历史（history）变为清除（damp）。

一个路由器经历最大时间时只能看成是被抑制的。该时间缺省为半生存周期的 4 倍。在清除状态，路由不再被公布到 BGP 邻居，一旦在重用极限（reuse-limit）水平之下损失值被删除，路由器将会把路由放回到 BGP 路由表中，并重新将该路由转发到邻居。确定在重用极限水平之下一个路由是否故障的进程每 10 秒钟执行一次。如果发现一个路由的损失在重用极限水平之下，则 BGP 进程就将该路由公布到所有其邻居中。这种清除路由的功能缺省为禁用。

第 8 章　路由表问题

Internet 按每年至少翻一番的速度迅猛增长，究竟到多大才能满足未来的需求仍然是一个值得探讨和研究的问题。更值得关心的问题是现在互联网的基础能否有足够因素而扩展以满足未来的需求。换句话说，互联网技术，或者构架，是否存在内在的限制的影响它的不断增长。本章之所以独立出来主要是因为路由表问题不仅仅是衡量高性能路由器性能的一个重要指标，而且也是影响 Internet 可伸缩性和稳定性的重要因素，同时路由表问题也是高性能路由器设计的一个难点。

可以把路由表问题分为几个方面：路由表容量、路由表的高效管理、路由表查找以及其如何影响路由器系统性能和 Internet 的可伸缩性与稳定性等方面。其中大容量路由表（特别是 BGP 路由表）的管理策略、查找算法是当前研究的重点和难点，这也是本章介绍的重点。

首先分析 Internet BGP 路由表的增长状况以及这种增长所带来的一些问题，随后介绍一些当前常用的路由查找算法，最后详细描述一种基于压缩树的路由查找，从而说明研究高效的管理路由表以及实现路由快速查找的难度何在。

本章 8.2 节对路由表的分析主要是参考了 Geoff Huston 的论文“Analyzing the Internet's BGP Routing Table”（The Internet Protocol Journal, Volume 4, Number 1, March 2001）以及参考文献所给出的结果。

8.1　概　　述

从整体上可以把 Internet 看作一个由很多自治系统组成的松散网络的集合，每一个网络都有自己的运行策略、收费规则、不同的服务和用户，它们都独立执行如何提供各种网络服务的策略。地址空间和路由系统把这些独立网络连接成一个连续的整体。维持网络内部路由的完整性通过内部网关协议完成，而域间的路由协议把网络连接成一个更大的路由域。

早在 1990 年，人们研究 Internet 的可扩展性时，就已经把两个关键的问题提了出来：路由问题和地址问题（RFC 1287）。随着越来越多的网络设备接入 Internet，要维护这些设备之间的可达性信息的路由表也在不断增长。人们对 32 位 Ipv4 地址空间局限性的研究确实获得了很多成果，包括制定了 Ipv6 的规范、使用地址转换（Network Address Translation，NAT）来允许在两个不同的网络地址域之间进行一定的透明交互等等。然而，整个路由系统的快速增长并没有减慢，因为路由空间受网络拓扑结构的复杂性、自治系统的个数以及一个路由表项覆盖的地址数目等综合因素影响。当一个网络向外部路由域广播一个路由更新信息时，这

个路由项（或多个）就会在整个 Internet 的所有外部路由域传播。

目前国际上很多机构和团体在对 Internet 的路由表变化的情况进行研究，这些研究主要是基于 BGP 路由表在近十几年来的变化。

8.2 Internet BGP 路由表的分析

为了衡量整个路由表的特性以及变化，必须选一个参考点（这个参考点是某一个外部路由域的一个组成部分），然后在这个参考点上考察 BGP 路由表的变化。

对路由表的分析是分散进行的。从 1988 年到 1992 年，Merit 就以间隔大概为一个月的频率对 Internet 的路由表进行了分析（RFC 1338），这项工作在 1994 年又在 Netherlands 的 Surfnet 由 Erik-Jan Bos 重新开始。从 1994 年初开始，Erik-Jan Bos 就以一小时的间隔时间反复观测 Internet BGP 路由表动态变化。Geoff Huston 在 1997 年借鉴了 Erik-Jan Bos 的观测方法，在澳大利亚的 AS 1221 边缘选择了一个参考点，开始对 Internet 的 BGP 路由表进行间隔一小时的观测。以上这些努力使我们能够对过去的 13 年里 Internet 路由表的动态变化以及发展趋势进行客观的分析，见图 8-1。

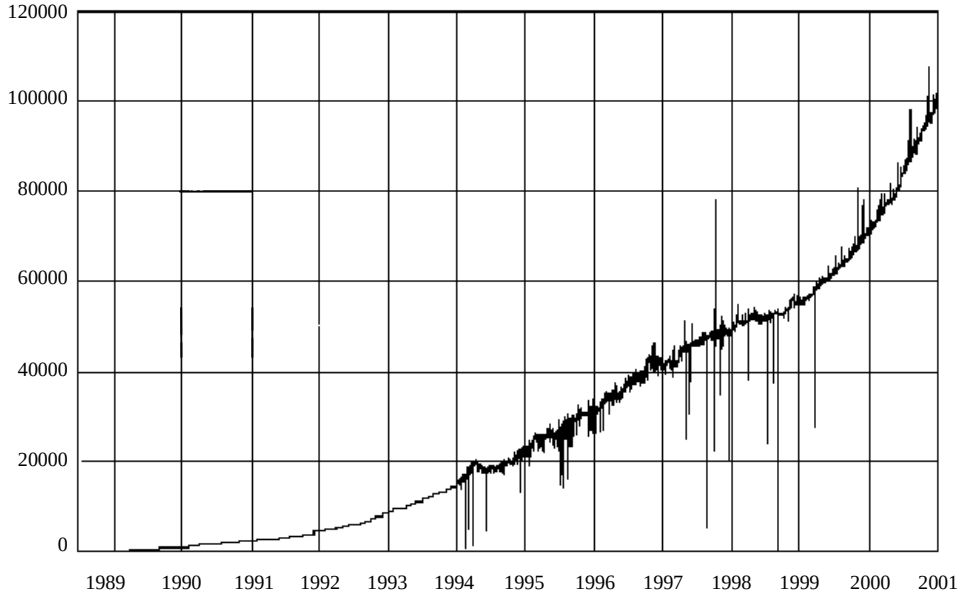


图 8-1 BGP 路由表的增长（1988～2001 年）

8.2.1 BGP 路由表的增长

从所采集的数据来看，在整体上可以把路由表的变化分为 4 个阶段，如图 8-2 所示。

1．没有使用 CIDR（无类域间路由）以前的增长

从 1988 至到 1994 年 4 月，路由表明显呈指数增长，见图 8-2。增长的主要原因是使用

了大量的 C 类地址（24 位地址前缀），毋庸置疑，这种增长速度在短短几年内就使非缺省路由器的 BGP 路由表达到饱和。从整体上来看，这种增长速度甚至超过了当时所用网络硬件和软件的性能增长速度。

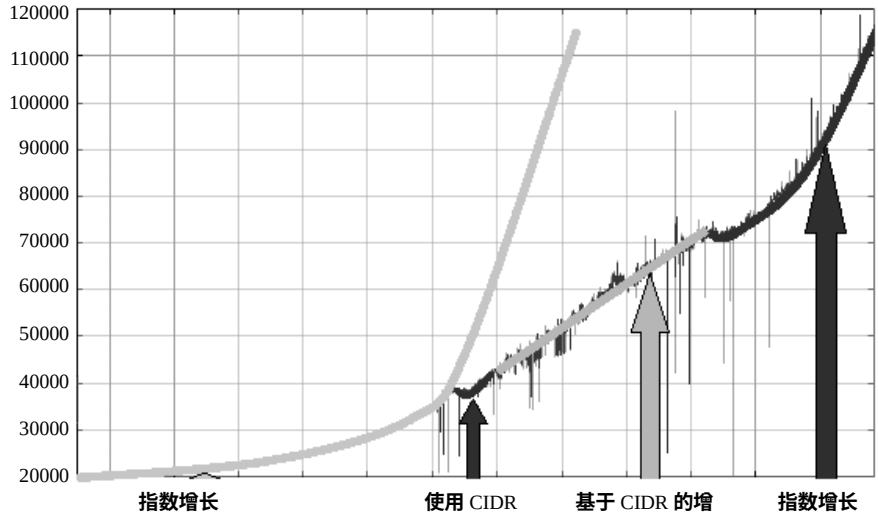


图 8-2　路由表变化的 4 个阶段

2．CIDR 的使用

作为对路由表快速增长的一种回应，网络界的工程师们开始使用一种只带有地址前缀和前缀长度的路由系统代替以前那种分配 A、B、C 类地址的路由系统。最值得关心的进展是在 1994 年到 1995 年间开始使用 CIDR 以及支持 CIDR 的 BGP-4 协议。这种努力的结果在路由表的变化上反映得非常明确。IETF 的 CIDR 工作组受到了很大鼓舞。

使用 CIDR 的一个目的就是提供路由聚合，网络服务商从 Internet 地址注册机构申请一个地址块，并把这个地址块向外部路由域广播。而服务商的客户又从该地址块中得到更小的地址（这里说的小地址是指前缀长度更长的地址，也就是更加明确的地址），而这些小地址并不直接在 Internet 上传播，而是经过服务商聚合以后才广播出去。在 1994 年间，聚合的使用使整个路由表的容量一直持续在 20000 条左右。

3．CIDR 后的增长

从 1994 年一直到 1998 年初的 4 年中，CIDR 对抑制 BGP 路由表的快速增长起到了非常重要和突出的作用。在这期间，当 Internet 的其他度量仍然成指数增长时，可喜的是 BGP 路由表的增长放慢了速度，呈线性增长，每年大概增加 10000 条，在 1997 到 1998 年甚至还低于线性增长。

这种增长背后的原因值得注意，在这期间所看到的 ISP 内部的聚合是与重新分配大量聚合后的地址块相陪伴的。更加细致的观测表明，使用聚合和 CIDR 使路由系统的稳定性增强了，每小时广播的路由总数开始下降，无论是占路由表的百分比还是绝对的数目。在一个网络边缘的不稳定性不会很快传播给路由核心区，路由的不稳定在最后一跳被聚合点吸收了。这同广泛使用的 BGP 关于消除路由抖动的策略一起在很大程度上减轻了短期内路由空间的抖动。

人们认识到 BGP 路由表的绝对大小是影响 Internet 可伸缩性的一个因素时，同时也明白了由于单个路由的撤销和广播所引起的连续更新路由表所带来的处理负载同样影响着 Internet 的稳定性。

总的说来，就是 BGP-4 协议的路由抖动消除策略和 CIDR 的使用在一定意义上消除了网络的动态不稳定性。有关 BGP-4 协议的路由抖动消除策略和 CIDR 的使用请读者参见同期出版的《Internet 路由结构分析》一书。

4．现在的增长

在 1998 年底，BGP 路由表的增长又开始变的激进起来，最近几年所观测的数据表明，路由表容量又有呈指数增长的趋势。这表明 CIDR 已经不能跟上 Internet 快速增长的步伐。如果按这种形势发展下去的话，路由表的增长速度又将会超过硬件提高的速度。

8.2.2 路由表引发的一系列问题

人们又开始研究这种快速增长所带来的一些问题，所做的主要工作包括统计当前路由系统里 AS 的个数、路由系统中单独 AS 路经的个数、路由表所覆盖地址空间的范围以及每一个路由项平均的覆盖范围。

BGP 路由表又重新呈指数增长的原因我们在这里暂且不做深入研究，比较明确的几个原因是 Internet 接入点数量的快速增长、大量小地址开始在骨干网上传播以及网络拓扑的复杂化等等。本小节主要是想通过分析 Internet 骨干网上路由表的变化趋势以及现在的状况，提出在设计核心路由器关于 BGP 路由表和 IP 路由表时所遇到的主要问题。

1．AS 号的大量消耗

任何多路（Multi-home）接入 Internet 并希望应用自己单独策略的网络都必须用申请一个 AS 号（这个 AS 号是全球唯一的）来匹配它所用策略广播的路由信息。总体上来说，一个网络只需一个自治系统号就够了，从而自治系统个数就与单独的路由策略的个数相匹配。也有一些例外：例如，一些大的网络服务提供商可能会在不同的区域应用不同的路由策略，因此就需要多个 AS 号，但这种情况很少。

在过去的 4 年里，AS 号的消耗也开始呈指数增长。自治系统号数目的增长与 BGP 路由表所覆盖地址空间的大小也有一定的关系。过去的几年里，BGP 路由表覆盖地址数都是以大概每年增长 7%的速度增加，而同时 AS 号的个数以 51%的速度增长，发生这种现象的一个原因就是大量单节点接入的（Single-home）连接转向了多路接入（Multi-home）。

如果按这种形势发展下去的话，16 位的自治系统号在 2005 年底就会被消耗完。现在 IETF 的 IDR（域间路由协议）工作组和一些大的网络设备提供商，如 Cisco、Juniper 等，正在努力修改现在的 BGP 协议以及制定新的标准，目的是在 BGP 协议中支持 32 位的自治系统号。伴随着协议的修改，同时业界也要制定一个可行的计划以使现行的系统逐步过渡到支持 32 位自治系统号的路由空间。

2．广播的平均前缀长度

使用 CIDR 聚合的最初目的是支持 BGP 路由广播的聚合。数据表明每一个 BGP 路由更新所覆盖的平均地址空间从 1999 年 11 月的 16000 条降到了 2000 年 10 月的 12100 条，这与 BGP 路由表中路由的平均前缀长度从 18.03 增加到 18.44 是相匹配的，因为路由前缀的长度是与其所覆盖的路由数目成反比。这些现象再一次表明在 Internet 上传输流量所使用的转发

表的条目数越来越多，而从大容量路由表中查找一条所需路由需要大量的时间，这里所说的时间是相对于转发数据所需的时间，因此设计更加高速的核心路由器路由表查找算法以及转发策略变得更加紧迫。

而另外一种潜在的趋势由路有更新所覆盖的地址空间还在不断降低。特别是随着 NAT 的广泛使用和已经将近利用完的 IPv4 路由空间，这种现象变得更为严峻。按这种趋势发展下去，BGP 路由表里路由的平均前缀长度将会以每 29 个月增加 1 位的速度增长。这就意味着当前所使用的基于最长前缀匹配的路由查找算法变得更加低效。

3．前缀长度的分发

除了分析平均的路由前缀的变化以外，对每一个大小的前缀在 BGP 路由表中所占用的比例以及它所覆盖的地址空间数目也是必要的。

在 6 年前，BGP 路由表中只有 20000 条到 30000 条路由项，而其路由项的前缀大部分为 19。2000 年 BGP 路由表达到了 100000 条路由，其中有 53000 条的前缀都是 24 位长的。也就是说 24 位长的路由项在 BGP 路由表中是增长速度最快的一种路由，当然，那些前缀长度更小的路由项的条数也在快速增长。

当前，传播一个 BGP 撤销路由的收敛时间大概为 2min，随着 BGP 网变得更为密集，这个时间可能还要增长，另外再伴随着路由动态改变变得更加频繁，使维护 BGP 路由系统的整体稳定变得更为困难。

8.2.3　地址耗尽

十几年前，人们开始考虑 Internet 可伸缩性的时候就已经意识到，IPv4 地址分配机制有限的地址空间将会是阻碍 Internet 持续快速发展的最大障碍。

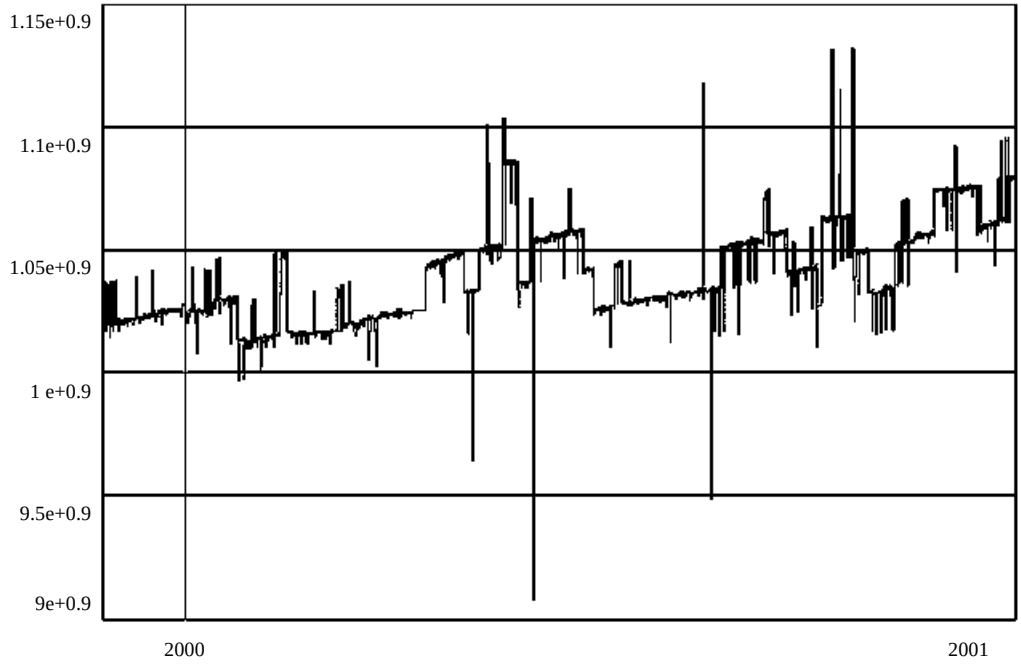


图 8-3　BGP 路由表覆盖的地址空间

现在，通过对 BGP 的路由表项所覆盖的整个地址空间的大小的观测与计算,再一次证明了这种现象。从 1999 年 11 月到 2000 年 12 月，Internet 的 BGP 路由表全部路由项所覆盖的地址空间从 10.2 亿条增长到了 10.6 亿条。同时还有许多前缀长度为 8 的路由在动态地广播，如果把这些短前缀长度的路由对路由表覆盖空间的影响算上的话，BGP 路由表覆盖的地址空间将会达到 10.9 亿，见图 8-3。

目前 IPv4 自治使用增长的速率为每年增加 7%，如果按这种增长速度发展下去，现在所剩余的地址空间最多能供人们使用 19 年。

8.2.4 其他一些畸形现象

由于一些连续的地址空间是公用的，它们不属于任何一个单独机构的管理范围，例如,各网络服务提供商之间的路由域就不被单独的服务商管理。这种地址空间经常会使路由表时不时呈现各种奇怪的现象。

在 1997 年底发生过一件值得注意的事情，一些大的前缀（长度很小）被分解成许多 24 位长的前缀集合，这些路由项自然地注入进那些各个服务商之间的路由域。BGP 路由表监视器里的图线突然发生大规模上升，BGP 路由表项从 50000 增加到了 78000 条，然后又突然降低。在当时，网络设备的硬件能力，包括处理能力和存储能力，还不能承受这种规模的路由更新。还有许多其他的奇怪现象，比如：路由表中突然有了许多 31 位长，甚至几百条 32 位长的路由。

虽然出现这些现象的原因可以归咎于各种各样的配置错误，但同时也表明人们还没有关心向路由表中增加路由时，使用什么样的过滤策略。在考虑到 BGP 路由表的分布性以及它在支持整个 Internet 得以存在所起到的主要作用时，这种现象可以看作 Internet 脆弱性的一个重要因素。现在广泛使用的一种对应方法是在 BGP 接收路由更新时加上一定的认证机制，从而维护路由表的完整性，在一定程度上消除路由的快速增长引起的一些畸形现象。

8.2.5 结论

与 BGP 路由表影响 Internet 的稳定性和动态变化特性一样，其增长也受很多因素影响，而大部分都不是技术上的因素。任何一个单独的团体或者个人都不能管理好这个公共的资源。人们期望的是，让那些国际性组织尽量采取一定的全局性的强制性措施和策略来改变目前这种状况，然而对 BGP 路由表这样分布性很强的公共资源，那样做有很大的难度。目前能做到的仅仅是分析出现这种状况的原因，然后从技术的角度来改变这种趋势。统计结果表明，BGP 路由表又重新呈指数曲线增长主要有以下几个原因。

1．多路接入小网络数目的增加

BGP 路由表近几年又重新快速增长的一个非常明确的原因就是大量多路接入小网络的出现。这些小网络同多个邻居和上层的网络服务提供商维持连接。同只与上层网络服务商维持一个邻居关系相比，在那些相对紧凑的网络拓扑中，使用多个连接能减少总的链路开销，同时这也是提高网络整体可用性的一个有效手段。建立多个接入连接基本上被看作是提高利用上层网络服务商提供服务的灵活性与弹性的一种有效策略。

但对整个网络来说这会带来一些负面影响，BGP 路由更新所携带的路由项数目增大了，而且还要无限制地相互学习。对于上层网络服务商来说，其客户与之维持多个 BGP 连接对他们本身来说没有任何不利，相反这能增加其服务的弹性和灵活性，而他们才不愿意自己付出

代价去维护 Internet 的灵活性，因为 Internet 是公用的。不知道他们想过没有 BGP 路由表增长的步伐能不能满足他们的客户所用的多个接入连接所带来的结果。

2．更加密集的互联网拓扑

随着网络服务费用的降低，Internet 的很多分支都创建了一个快速增长的网络服务市场，同时单点接入服务商的模式又逐渐被在 Internet 边缘接入扩展连接的模式所代替。商业目的的驱动使各种各样的网络连接方式都出现了，原来的分层网络体系结构已经变得不再那么明晰，这就使 Internet 的整体拓扑变得更加复杂，从而引入一大堆路由问题，也使路由更新变得更加频繁。

3．通过控制路由实现流量工程

另外一个驱动 BGP 路由表增长的因素是在多路接入网络中，在不同的链路上广播包含特定的小地址的路由更新，从而实现流量工程的目的。虽然在一个独立网络中使用 MPLS 作为一种基本的流控措施来实现流量工程，但在服务商之间以及服务商与其客户之间的网络环境中，使用这种方式只会使事情变得更加复杂。他们只能通过控制 BGP 路由更新的内容与广播路径达到流量工程的目的，这就导致 BGP 路由表的增长和路由域的稳定性变差。

8.3　高效的路由查找算法

路由查找是路由器应完成的一项最基本的功能。随着网络流量的增加，路由器在单位时间内路由的报文数量越来越多。研究和实现高效的路由查找算法也是设计和实现高性能路由器不可缺少的一个重要部分。

8.3.1　IP 地址组成变化及对路由查找的影响

在 Internet 上每一台主机和路由器都用 IP 地址进行标记和识别。IP 地址的组成包括两部分：网络号和主机号。这种组合方式对每一台主机或路由器而言是唯一的。路由器的选径即是用报文头中的目的 IP 地址查找由路由协议建立的路由表，得到报文的下一跳 IP 地址和输出端口。目的 IP 地址起索引作用。查找过程可以通过检查 IP 地址的类型查路由子表实现。可以想象，这个过程不复杂，而且也不费时间。

但是 IP 地址的组成方式在进入 90 年代之后发生了重要的变化，这一变化使路由查找问题一下子变得复杂起来。

进入 90 年代以后，网络规模不断发展，入网机器激增。这时出现了一个问题，引起了人们的关注：Internet 中的 B 类地址将要耗尽，而人们又不愿意申请数量多但主机数少的 C 类地址。为了解决这个问题，提出了新的地址分配策略：把一组连续的 C 类地址分配给申请者，而不是分配一个 B 类地址。这虽然解决了 B 类地址耗尽问题，但却使路由表成倍的增加。为了减小路由表空间，于是产生了无类域间路由 CIDR。路由表项可以通过汇集，使多个连续表项只占路由表中一项，通过可变长度的掩码标识。

这样一来，IP 地址的构成就成了如下形式：〈Prefix，Host〉。

IP 地址由前缀和主机地址组成。其中，前缀和主机地址长度也不再具有固定的位数。事物都是两方面的，路由表空间虽然得到有效的减少，但是却给路由表查找带来了麻烦，因为路由表中完全是一些无规则的前缀。要查找 IP 路由，必须要找到最长匹配的表项才行。举例

来说，若路由表中有前缀 P1=0101，P2=0101101，P3=010110101011，要查找的地址前 12 位为 010101101011，则最长匹配的前缀是 P1，而地址 010110101011 的最长匹配前缀是 P3，虽然 P2 也能匹配。

路由查找问题也被称为最长前缀匹配问题 LMP（Longest Matching Prefix）显然，象 hash 这样的匹配算法不能直接用于最长前缀匹配问题。

8.3.2 常用的路由表查找算法

后来，人们提出了一些新的算法解决 LMP 问题。例如对精确匹配算法的改进、基于 Trie 的方法、硬件解决方法、路由 cache 方法等。但是，这些实现方法要么访存次数太多，要么实现代价较大。用软件实现 IP 路由查找，关键是减少存储器访问次数，因为存储器相对于处理器的速度慢很多。

1．二分法搜索 hash 表

因为 hash 方法是一种高效的精确匹配算法，考虑把它应用到 LMP 问题中来。hash 方法分为线性搜索 hash 表算法和二分法搜索 hash 表算法。

（1）线性搜索 hash 表算法

如果把路由表组织成如下形式：

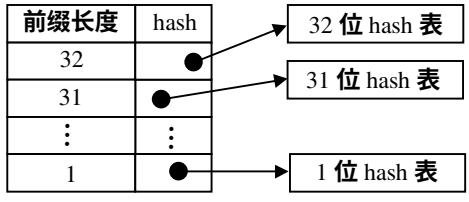


图 8-4 路由表组织

当得到一个要查找的 IP 地址时，把 IP 地址的 32 位前缀，31 位前缀等依次取出来去相应 hash 表中查找。若匹配成功，则算法结束，若失败则继续进行。这种算法称为线性搜索 hash 表算法。

LinearSearch 算法简单，但在最坏的情况下，需要 O(W)次 hash 表查找。

（2）二分法搜索 hash 表算法

如果对 hash 表的查找，不按前缀长度顺序进行，而是按二分法搜索前缀长度的 hash 表，那么可以使算法时间复杂度由 O(W)减少为 $O(\log_2 W)$。为了使查找顺利进行，在 hash 表中需要引入称为 marker 的伪前缀，用于指导查找往前缀长度大的方向进行。为了便于说明，下面以前缀 P1 =0，P2=00，P3=111 为例说明此算法。

假设要查找 111 的最长前缀匹配，如图 8-5 所示。

查找不是从前缀长度 3 开始，而是从前缀长度 2 开始，这样会查到一个为 11 的 marker，说明在更长的前缀 hash 表中有以 11 开头的前缀，所以下一步查找前缀长度为 3 的 hash 表，从而和 P3 匹配。

2．二分法查找 hash 表算法的改进

前缀长度	二分法	Hash 表	带 marker 的 hash 表
1		P1=0	P1=0
2	指向 1 和 2	P2=00	P2=00,M=11
3		P3=111	P3=111

图 8-5　查找 111 的最长前缀

从上述可以看出，对于 IPv4 来说，前缀长度最多有 32 种情况（实际上很少有比 8 小的前缀），那么二叉树的高度为 5（$\log_2 32$），查找在最坏的情况下要 5 次 hash 访问（每次 hash 访问至少需 1 次存储器访问）。如果能减少二叉树的高度，就能减少 hash 访问次数。基于此，提出如下的改进方法。

为了使二叉树的高度降低，可以采用压缩路由前缀的方法，具体做法是：把任一路由前缀都压缩到前缀长度为 4 的倍数上去。也就是说，把 8~11 位的前缀用 8 位的前缀表示，12~15 位的前缀用 12 位的前缀表示，等等。并且，给每一个压缩后的前缀增加一个信息域，用于记录被压缩掉的路由前缀。这个信息域是 2 字节的 bit 序列，如图 8-5 所示。

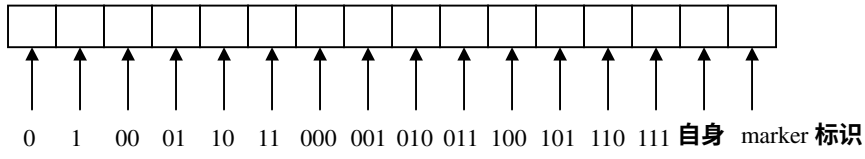


图 8-6　压缩域

当被压缩掉的 bit 位（至多压缩掉 3 位）存在时，可以把 bit 序列中相应的 bit 位置 1，否则置 0。最后一位为 marker 标示位，若被压缩后的前缀集合不是 marker，则此位置为 0，若此前缀是一个 marker（同时也可能包含纯前缀），则此位置为 1。

如给定以下前缀：

p1=01010001

p2=010100011

p3=0101000110

p4=01010001010

压缩为 8 位的前缀 p=01010001 对应的 bit 序列为 0100100010000010。

经过如上处理，hash 表的前缀长度由 32 种情况减少为 8 种情况。重新设计的二叉树如图 8-7 所示。

这样处理后，二叉树的高度由原来的 5 层减少为 3 层，hash 表的最长搜索路径从 5 步减少为 3 步。同时，对 marker 的存储要求也从 $O(\log_2 W)$减少为 $O(\log_2 \frac{W}{4})$，从而节约了存储空间。

由于对路由表进行了压缩，需要对 BinarySearch 算法进行一些改动。改动主要发生在为了

避免回溯，记下 LMP 的计算，因为此时不能只看预先计算好的 LMP，还要参考 bit 序列才行。

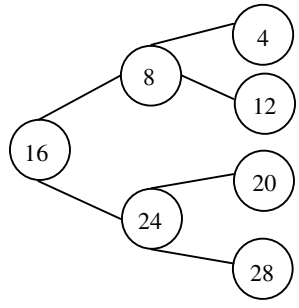


图 8-7　压缩二叉树

8.4　一种基于压缩树的路由查找算法

表示路由表的最常用的数据结构是树结构（严格地说是 **trie**，一个 trie 被组织成类似树的数据结构）。树的每个节点由零或多个子节点组成。每次关键字查询（路由查询中为目的 IP 地址）从树的根部开始，沿着树找到最长匹配。在每个节点处，trie 算法使用关键字一部分（目的 IP 地址的一个或几个 bit）决定具体哪个节点将被下次访问。与树不同的是，tire 的节点中没有存储完整的关键字（key）信息。压缩树的目的是寻求查找速度和存储器容量之间的折衷方案。

事实上这种树是由有效的网络前缀节点和连接有效网络前缀节点的无效前缀节点共同组成，因此称为前缀树。有效网络前缀的信息是从路由传播协议中获得。

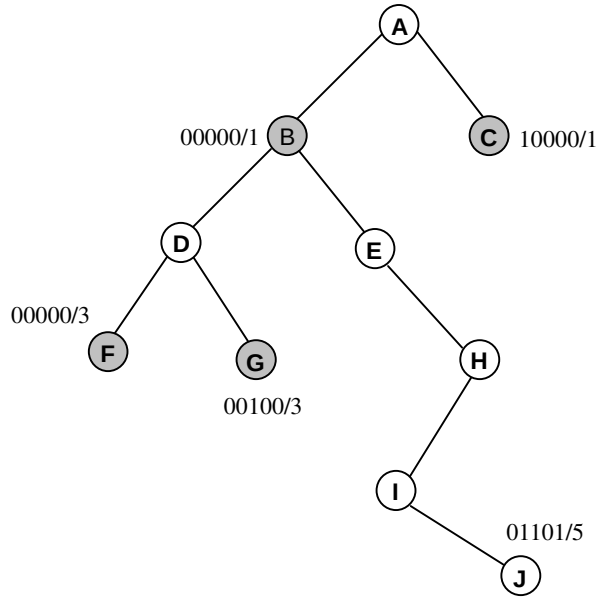


图 8-8　典型的前缀树

图 8-8 的二叉树中，有效网络前缀为 5 个，用灰色节点表示。利用这个前缀树我们可以有效地查到任何字符串的最长前缀匹配。例如：一个 5bit 的字符串 S1＝10001/5，因为 S1 的第一比特是 1，于是选择根节点 A 的右树枝（节点 C），因为节点 C 为叶节点，所以可以判断 S1 的最长前缀匹配是 C＝10000/1。再考虑另外一个字符串 S2＝00110/5，由于 S2 的第一比特是 0，于是选择节点 A 的左树枝（节点 B）。鉴于节点 B 是灰色的，我们知道 S2 匹配了节点 B 指示的前缀 00000/1。因为 S2 的第二比特仍然是 0，选择节点 B 的左树枝（节点 D）；因为 S2 的第三比特是 1，选择节点 D 的右树枝（节点 G）；因为节点 G 是叶节点，所以判断 S2 的最长匹配是 G＝00100/3，它比 00000/1 匹配得更长。为了在内存中记录树的数据结构，同时为了树的重建、查询的方便，对树中的节点进行如下分类和编码。

这种对每个节点进行四比特的编码方式中，按照从最高比特位（MSB）到最低比特位（LSB）的定义依次是：

Bit4：出口节点标志位。“0”非出口节点，“1”出口节点。

Bit3：有效网络前缀标志位。“0”节点对应无效网络前缀，“1”节点对应有效网络前缀

Bit2：具有左子节点标志位。“0”没有左子节点，“1”有左子节点。

Bit1：具有右子节点标志位。“0”没有右子节点，“1”有右子节点。

需要注意的是树的所有叶节点都是有效节点，都对应有效网络前缀；即无左右子节点，又不对应有效网络前缀的节点在树中不予以保存。如果某节点的左右两个子节点都是有效节点，则该节点标记为无效节点，因为无论下一比特是什么，都能找到的有效的匹配。

8.4.1　压缩树算法

1．16 位索引

如果将 32 比特的 IP 地址都用树型结构来存储，那么对于整个 IPv4 空间只需要一棵树就够了。但这样做的弊端是树的深度太大，最坏情况的查询可能需要 32 个循环，即意味着访问存储器 32 次，这种速度对拥有大容量路由表的路由器是不可容忍的。为了减少查询时间，必须减小树的深度。一种策略是将 IP 地址分为两部分：第一部分由 IP 地址的前 k 比特组成，作为一个 2k 个表项的线性表 TABLE-16 索引值（这里取 k=16）。第二部分包括剩余的(32－k)比特，用树型结构进行存储。该树的位置存放在 TABLE-16 中的页面指针域中。其原理示意如图 8-9 所示。

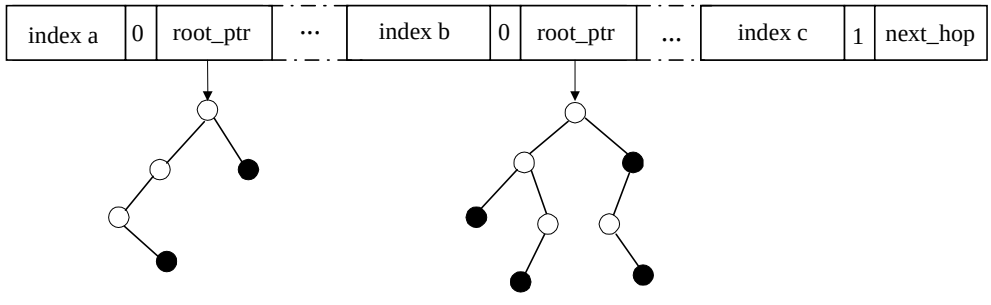


图 8-9　哈希索引的前缀树

TABLE-16 表项的每一项有 16 位。表项的格式如图 8-10 所示。

Done	下一跳指针/子树指针

图 8-10　表项

其中，若查询时 Done 为 1，表明网络前缀小于 16 位。查询结束，后 15 位存储的是下一跳指针信息，读出路由表的相应内容即可完成查找。若查询时 Done 为 0，表明有大于 16bit 的网络前缀，后 15 位存储的是子树指针，指向子树的根节点，因此需要二级查询。

可以看出 TABLE-16 表项的第二部分只有 15 位，最多只能支持 32K 子树，因此这种方法只准备支持 32k 个网络前缀。这种方法对部分长度小于 16 位的前缀进行了扩展。例如对于前缀 202.112/9，我们先将它扩展成 16 位，结果得到 2(16-9)个 16 位的前缀，这些前缀的下一站的值相同，都是 202.112/9 所对应的下一站值。整个算法进行二级查找，第一级查找匹配了 16 位，第二级查找在最坏情况下需要匹配 16 次。这样，两级查找的时间比原来几乎减少了一半。为了进一步减少查找时间，必须考虑增加每次访问存储器所获取的节点信息数，因此对深度为 16 的树进行压缩处理。

2．16 位压缩

压缩树时，考虑到树的递归型结构，可以将树分割成一棵棵的子树。例如将前面的树分割为如图 8-11 所示的子树。

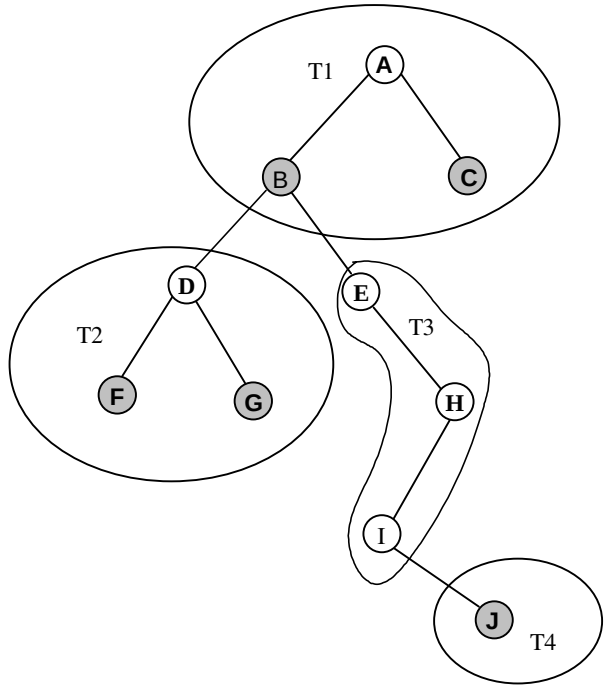


图 8-11　分割后的子树

图 8-11 中的树被分为 4 棵子树：T1{A,B,C}、T2{D,F,G}、T3{E,H,I}和 T4{J}，按广度优

先的顺序编号。图中有一类称为出口节点的节点。出口节点具有子节点并且和它的子节点不在同一棵子树中。本例中的出口节点是节点 B 和节点 I。

采用固定大小的连续内存块——页面（Page）的方式来存储每一棵子树。每页面能存储的节点数的上限是确定的，页面的数据结构示意如图 8-12 所示。

第一部分	第二部分	第三部分	第四部分
指针：指向下一跳	指针：指向下一页面	节点 1-节点 2-节点 3……节点 n	Rc1-Rc2- …… Rc N

图 8-12　页面的数据结构

页面的数据内容分为四个部分。第三部分存放该子树的拓扑结构，按照深度优先的顺序存放子树中各节点的编码。第四部分按照第三部分节点的顺序依次存放子树的每个对应节点的右子节点在第三部分中的位置，如果该节点没有右子节点或其右子节点不在本页面中则相应位置放入无效值。第一部分是子树中第一个有效网络前缀所对应的下一站地址的指针。第二部分是页面表的索引值，由该值索引到的页面是本页面中最左边的出口节点的第一个子节点所在的页面。

3．子树生成的描述与压缩算法

假设每个页面存放的节点数是 N。首先从树的根节点开始用广度优先的原则遍历该树，遍历到的前 N 个节点放在第一个页面中。然后以第一个出口节点的第一个子节点为根节点，重复上述的操作。各个页面的下一页面的指针分别指向该页面中第一个出口的第一个子节点（即最左边的子节点）所在的页面，下一站指针指向页面中第一个有效节点所对应的下一站指针。此算法表述如下：

[假设]：NH：下一站表
　　PG：页面表
　　T：要被压缩的树
第一步：
　　C：＝ T 的根节点；
　　i：＝ 0；
第二步：
　　如果 C 为空
　　　　则　{PG，NH}就是 T 的压缩结果，处理结束；
　　否则 进行第三步处理；
第三步：
　　n：＝ C 的第一个节点，把该节点从 C 中移走；
　　以广度优先的顺序遍历以 n 为根节点的子树；
　　　如果该子树有多于 N 的节点，则把前 N 个节点按照深度优先
　　　的顺序存放在 PG（i）的第三部分中；
　　　否则，该子树的全部节点按照深度优先的顺序存放在 PG（i）
　　　的第三部分中；
　　对于 PG（i）的第三部分的节点：

找出它们的右子节点（如果在 PG（i）中的话），把它们的位置编号按照相同的顺序填入 PG（i）的第四部分中；

如果 i=0　那么 PG（i）的下一页面指针值是 1，下一站指针值是 0；

如果 i>0　那么

{

PG（i）的下一页面指针值：＝PG（i）的下一页面指针值＋PG（i-1）中出口节点的子节点的数目；

PG（i）的下一站指针值：＝PG（i）的下一站指针值＋PG（i-1）中有效节点的个数

}

C：＝PG（i）中出口节点的子节点按照广度优先的顺序排列得到的列表

i：＝i+1；

第四步：

重复第二步。

4．查找的原理和搜索算法的描述

很明显，每个页面存储的是一棵子树，并且第一节点是这棵子树的根节点。为了查找的方便，在页面存储时我们采用深度优先的顺序：这样，我们搜索到一个节点并且需要继续搜索它的左子节点时：

■　如果该节点不是出口节点，那么在页面中紧接着它的那个节点就是它的左子节点。

■　如果它是出口节点，那么只要确定该页面中在它之前的出口节点的子节点数目，就可以确定它的左子节点所在的页面的索引；

反之，如果我们需要继续搜索某个节点的右子节点：

■　如果该节点不是出口节点，那么根据我们所存放的每个节点的右子节点的信息直接可以确定该右子节点的位置。

■　如果该节点是出口节点，那么只要确定该页面中在它之前的出口节点的子节点的数目，就可以确定它的右子节点所在的页面的索引。

搜索算法可以简要地描述如下：

[假设]　{PG,NH}：路由表在存储器中的表示：

PG（0）：　PG 表中的第一个页面；

T0　　：PG（0）所对应的子树；

在完成 T0 的搜索（即搜索到 T0 中的叶节点或者出口节点）时，可能出现的情况有：

■　到达一个叶子节点 L 而且没有找到匹配；

■　到达叶子节点 L 而且找到匹配；

■　到达出口节点 E 而且没有找到匹配；

■　到达出口节点 E 而且找到匹配。

如果搜索结果是前两种情况，那么表示搜索已经全部结束，对于找到匹配的，最后的一次匹配就是最长匹配。如果搜索情况是后两种，表示搜索没有完毕，需要进一步搜索。假设这时已经搜索到了 K 位，那么我们需要用 K+1 位对 E 的子节点 D 作进一步的搜索。如果 D

不存在则结束。如果 D 存在，假设 PG（0）的下一页面指针值是 P，经计算 D 是 PG（0）中出口节点中的第 M 个子节点，那么 D 所在的页面是 P＋M，然后我们在 PG（P+M）页面中继续搜索，直到遇到前两种情况为止。搜索结束时，如果得到的最长匹配发生在该页面中的第 X 个有效节点，并且这个页面的下一站指针是 N，那么下一站的地址是 NH（X+N）。

8.4.2　实现压缩树算法的改进

考察 16 位压缩算法可知，在极端情况下，每页储存一棵四级的树，这样查找两个页面就可以将树推进到第八层。尽管如此，在 IPv4 的 Internet 上大量存在的是 24 位的网络前缀，而从索引的根节点处（第 16 位）到第 24 位共有 9 层，因此会频繁地出现查询机构需要从内存中读取三个页面才能结束查找的情况。

至于这种状态导致页的利用率降低的推理是毫无根据的。因为这种设计思想确定了路由表的表项有且仅有 32K 个，即便是这样大量的 24 位的前缀都能恰好存在满满的各自页面里，也无法使路由项数增加。

为了减少上述情况下的查询时间，有两种可以考虑的方法：

- 改变每个页面中存放的节点数目；
- 将树的根节点往下推进一层，即从原来的 16 位推进到 17 位。

第一种方法很直观，只需改变一个参数，算法本身并不需要修改。但是这个参数的改变将导致页面大小的改变，不利于存储器的分配和管理。我们决定采用第二种方法，这种方法又有两种方式：17 位索引和 16 位的扩展。

1．17 位索引

这种方法和 16 位索引基本一致，所不同的是将页面表的索引项由原来的 64k 增加到 128k，扩大了一倍。采纳这种算法后，24 位处的问题得到完全的解决。

2．16 位索引扩展

[输入] T:待压缩的树、P:页面表、R:下一站表；

[输出] 压缩后的树：P 和 R。

第一步: 初始化

C:={ T 的根节点};

i:=0;

第二步: 检查结束条件

如果 C 是空集,则返回 R 和 P 作为 T 的表示，结束查找；

否则，继续第三步。

第三步：压缩子树

r：= C 的第一个节点，从 C 中移走第一个节点；

使用 Pi 存放压缩后的以 r 为根节点的子树，按照广度优先的顺序进行编号，

直到页面没有空闲空间或 r 根节点下的所有子节点都已经被包括；

将 Pi 中所有有效前缀以广度优先的顺序增加到 R 中；

C：= C＋Pi 中的出口节点的子节点，并且按广度优先的顺序排列；

if　i=0 , 则 NextPagesPtr(Pi) :=1 而且 NextHopPtr(Pi-1) := 0;

if i>0

{

NextPagePtr(Pi):=NextPagePtr(Pi-1)+P(i-1)**中出口节点的子节点的数目；**

NextHopPtr(Pi):=NextHopPtr(Pi-1)+P(i-1)**中有效节点的个数；**

}

i=i+1;

第四步：重复第二步。

第 9 章　高级交换技术

当今绝大部分的企业网都已变成实施 TCP/IP 协议的 Web 技术的内联网，用户的数据往往越过本地的网络在网际间传送，因而，路由器常常不堪重负。随着网络持续高速发展、新的网络技术不断涌现。这些技术对网络的扩充性、灵活性、透明性以及速率性能等提供了更高的支持，比较重要的两种交换技术是第三层交换和多协议标签交换技术（Multiprotocol Label Switching，MPLS）。

第三层交换，也称为 IP 交换技术、高速路由技术等，主要是基于第三层（L3）地址转发流量、执行交换功能、可能提供特别服务（例如认证）、也可能不具备路由处理功能的一种技术。MPLS 把灵活性与网络层丰富的功能结合起来，而且 MPLS 的简单性可以改善转发性能。由于多协议标签交换可以表示各种粒度，因此相同的转发功能可以支持各种路由能力，如基于目的地的路由、组播路由、层次路由以及灵活的路由控制，并且可以满足新的应用需求。本章介绍这两种技术实现快速路由和转发的机制。

9.1 概　　述

目前主要的第三层交换技术有两类：第一类是报文到报文（Packet-by-Packet）交换，每一个报文都要经历第三层处理（即至少是路由处理），并且数据流转发是基于第三层地址；第二类是流交换（Flow Switch），它不在第三层处理所有报文，只分析流中的第一个报文，以便完成路由处理，并基于第三层地址转发该报文。流中的后续报文使用一种或多种捷径技术进行处理，其设计目标是方便线速路由处理。采用的方法有 Ipsilon 的 IP 交换、Cisco 的标签交换、3Com 的 Fast IP、Cabletron 的 SecureFast 虚拟网络、ATM 论坛的 MPOA 等。

同传统路由器一样，第三层交换对报文的处理主要有：

- 基于第三层 IP 地址信息决定转发路径。
- 通过校验和检查第三层头的完整性。
- 检查（TTL）报文过期，并相应减少 TTL。
- 处理信息中的任何选项。
- 更新管理信息库（MIB）中的转发统计。
- 如果有要求，对数据进行保护。

理解第三层技术首先需要区分是否每一个报文都要经历第三层处理(至少是路由处理)，并且业务流转发是基于第三层地址的。

基本的网络互连设备有三种：网桥、路由器和网关。交换机是一种高级的网桥，路由器

就是路由器，而网关工作在第七层（应用层）。交换的基本功能就是转发信息流，将输入端口与输出端口对应起来。交换机的体系结构和实现方法（例如交换矩阵或 TDM 总线）决定了在什么时候进行对应以及如何对应。路由的基本功能是路由处理（路径确定、路由表的维护）和流量转发（地址解析、计数器维护、报文头重写）。

9.2 MPLS 交换

Internet 持续的增长要求 ISP 提供更高的带宽。然而 Internet 的增长并不是要求提供更高带宽的唯一驱动因素，对带宽的需求同时来自于多媒体应用。这种应用要求对单播和组播同时进行高效转发。Internet 的增长也要求改善 Internet 的可扩展性。为了支持高质量可扩展的路由系统，路由器必须维护大量路由信息并具有构造层次式路由的能力。

虽然在许多情况下，基于目的地的路由已经足够，但在有些情况下这种方法却有其不足。克服基于目的地路由的不足以及提供对流量更灵活的控制就变得越来越重要。

多协议标签交换技术（MPLS）可以用来解决这些问题。多协议标记交换（MPLS），由于其支持流量工程，而在新型公共网络中被作为一项重要的技术。流量工程是通过将大量的用户业务转移到经过服务提供商网络中特定节点的预先设定的路径来实现的。这些预先设定的路径被称作标记交换路径（LSP）。

9.2.1 MPLS 体系结构

MPLS 由两部分组成：转发组件和控制组件。转发组件利用报文中携带的标签信息及标签路由器（LSR）中维护的信息进行报文转发；控制组件由一组模块组成，这些模块负责在一组互连的标签交换路由器之间维护正确的标签转发信息，每个模块提供一个特定的控制功能。

1．转发组件

当标签交换路由器收到一个包含标签的报文时，LSR 使用标签作为索引在标签转发信息库（LIB）中进行查找。LIB 中的每一项包含输入标签和多个形如<输出标签，输出接口，输出链路信息>的子项。如果 LSR 找到一项，其输入标签与报文的标签相同，那么 LSR 就用输出标签替换报文上的标签，替换链路信息（如 MAC 地址），并把报文转发到输出接口。

从上面对转发组件的描述中，可以得出如下结论：

■ 转发决定基于固定长度的精确匹配算法。相对于网络层的最长前缀匹配，精确匹配要简单得多。这样就可以把转发算法用硬件实现，提高转发性能。

■ 转发决定与标签的粒度无关。比如，相同的转发算法既用于单播又使用于组播。单播只有一个<输出标签，输出接口，输出链路信息>子项，而组播包含多个这样的子项。这说明标签交换虽然使用相同的转发但却可以支持不同的路由功能。

■ 转发算法与网络协议无关。使用特定网络层协议的控制模块，可以用相同的转发机制支持各种网络层协议。这也是多协议标签交换名字的由来。

标签封装要求每一个报文携带标签信息。标签信息的封装方法主要有两种：

■ 作为第二层头的一部分（如帧中继和 ATM）

■　把标签封装在第二层头和第三层头之间。

2．控制组件

标签交换的实质是把标签与网络层路由绑定在一起。控制组件负责创建标签绑定并在 LSR 之间分发标签绑定信息。创建工作包括分配标签和绑定标签与路由两部分。在 LSR 之间分发标签绑定信息可以有两种选择：

■　对已有的控制协议进行扩展；

■　使用标签分配协议 LDP。

标签交换体系结构的一个重要特征是标签绑定信息的分发和维护与相关路由信息的分发和维护相一致。

为了在支持多种路由功能的同时提高可扩展性，MPLS 支持多种转发粒度。一个标签可以代表一组路由；也可以代表一个特定的应用流（如 RSVP 流）；还可以和多目广播树绑定在一起。更一般的，标签可以和等效转发类（FEC）绑定在一起。FEC 将在第三节中讨论。

控制组件由一组控制模块组成，每个模块支持一个特定的路由功能。为了支持新的路由功能，只需加入新的模块即可。下面介绍一些重要的控制模块。

（1）基于目的地的路由

这一节要描述标签交换如何支持基于目的地的路由。在普通的基于目的地的路由中，路由器根据存储在转发信息库（FIB）中的信息对报文头中的目的地址进行分析，决定其转发路径。FIB 中的信息是由路由协议进行维护的（如 OSPF、BGP 等）。标签交换路由器为了支持基于目的地的路由，也要使用路由协议创建 FIB。

有两种方法用于标签分配：下游标签分配和下游按需标签分配。LSR 分配标签，并与 FIB 中的地址前缀进行绑定。在下游分配中，报文携带的标签由下游 LSR 进行分配和绑定；下游按需分配是说当上游 LSR 发出请求时，下游 LSR 才进行标签分配和绑定。

① 下游标签分配

对 FIB 中的每一条路由，LSR 为其分配一个标签，并在 LIB 中创建一项。这一项的输入标签是刚分配的标签，并在相邻 LSR 之间广播这一标签绑定信息。广播这一信息可以通过改进已有的协议（如 OSPF）实现，也可以通过标签分发协议（LDP）来实现。一个 LSR 收到标签绑定信息，并且这一信息来源于路由的下一跳，那么，LSR 就把标签值放到对应 LIB 项的输出标签中。整个工作过程如图 9-1 所示。

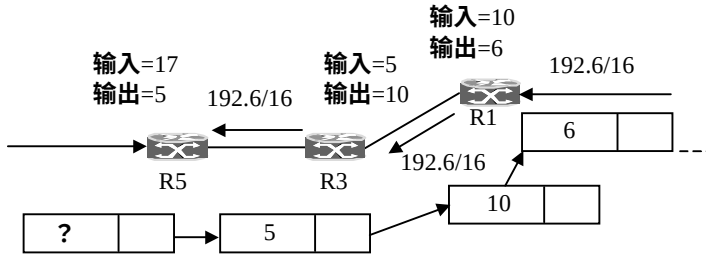


图 9-1　下游分配操作示例

当 LSR1 接收到<192.6/16,6>信息时（“6”表示标签，与图中长方框左侧内的数字对应），R1 就在 LIB 中创建一项对应于路由 192.6/16，并且把标签（6）作为输出标签。R1 也产生一

个标签（10）放到输入标签中，并把这一绑定信息广播给其它 LSR。如果 R3 把 R1 作为路由 192.6/16 的下一跳，R3 就在自己的 LIB 中创建一项，把（10）放到输出标签中，并产生标签（5），作为其输入标签，并把这一绑定信息广播给其它 LSR。使用相似的过程，其它所有的 LSR 在自己的 LIB 中创建对应于路由 192.6/16 的标签绑定。在这个例子中假定 R5 左边的路由器不是 LSR，那么 R5 收到的将是不带标签的 IP 报文。

当一个目的地是 192.6.10.5 的报文达到 R5 时，报文没有任何标签信息。R5 根据 LIB 中的信息进行转发决定。R5 会把标签（5）加到报文上，并把其转发给下一跳 R3，R3 收到该报文，取出报文中的标签，作为索引在 LIB 中进行查找，找到对应项，交换标签后，把报文转发给 R1。其它过程类似。

② 下游按需标签分配

在下游按需分配策略中，对于 FIB 中的每一条路由，LSR 确定路由的下一跳，并给下一跳发一个标签绑定请求（通过 LDP）。当下一跳收到这一请求时，它会分配一个标签，在其 LIB 中创建一项，并把创建的标签作为其输入标签，之后把这一绑定信息发给请求方。当请求方收到这一绑定信息，就在 LIB 中创建一项，并把收到的标签作为其输出标签。

这种方法主要应用于 ATM 交换机中。

值得一提的是，标签交换并没有完全消除网络层的转发。给没带标签的报文贴上标签要求网络层转发，这个功能由第一跳 LSR 完成。如图 9-1 中的 R5。

（2）层次路由

IP 路由体系结构由一组路由域组成。在一个域中，路由通过内部路由协议（如 OSPF）维护。跨域的路由由外部路由协议形成（如 BGP）。然而，域之内的所有路由器不仅要维护内部路由，还要维护外部路由。这样就会带来一些问题。首先大量信息需要更多的资源；其次路由信息增加会使路由收敛时间变长，因而也就降低了系统性能。

标签交换实现内部路由与外部路由的分离。使用标签交换，只有域边界的 LSR 才维护外部路由信息。这样就可以减少非边界 LSR 上的路由开销，减少了路由汇聚时间。为了支持这一功能，标签交换允许一个报文携带多个标签，形如一个栈。LSR 能交换栈顶的标签，弹出标签和压入标签。

一个处于路由域边界的 LSR 不仅要维护内部路由还要维护外部路由。内部路由提供到域内其它 LSR 的标签绑定信息。对每一条外部路由，边界 LSR 不仅要维护一个到那条路由的标签，还要维护一个到另一个边界 LSR 的标签，这条外部路由就是从那个 LSR 处学来的。

当一个报文在不同域的两个边界路由器之间进行交换时，报文的标签栈只包含一个标签（和外部路由相联系）。当报文在域之内转发时，标签栈就包含两个标签（第二标签由域的入口 LSR 放入并与到出口 LSR 的内部路由相对应）。栈顶的标签提供了入口 LSR 到出口 LSR 的 LSP（Label Switch Path）。栈顶由出口 LSR 弹出，之后进行正常转发。

这种情况下的控制组件和基于目的地的路由十分类似。不同之处在于，标签绑定信息不仅要在物理相邻的 LSR 之间分发，还要在一个域之内的边界 LSR 之间分发。

（3）QoS

为经过 LSR 的报文提供 QoS，需要两个机制：首先，需要把报文分成不同的类；其次，要保证对不同类提供适当的 QoS 特性（带宽、丢失率等）。

在报文第一次分类之后，标签交换就能提供简单的方法来标记报文属于某个特定的类。

最初的分类要使用配置信息、报文头中的网络层或更高层的信息。之后，和类对应的标签就应用于报文，无需再次进行报文分类，在 LSP 上的每个 LSR 就能高效地处理被标记的报文。

在 ISP 的网络中，标记交换能提供几类服务。在基于帧的媒质中，服务类别能编码到标签头的一个域中。在这种情况下，一个标签绑定由三元组构成<路由前缀、QoS 类别、标签>。这样的标签既用于转发决定也用于队列调度。为了提供更精细粒度的 QoS，可以把标签分配和 RSVP 一起使用。对 RSVP 进行简单的扩展，定义一个标签对象。这种对象能够携带 RSVP 预约信息。每一个 LSR 为 RSVP 会话分配一个标签，并把其传给上游 LSR。如此，标签与 RSVP 会话的结合就和标签与路由的下游分配过程非常类似。但是，此时的绑定是通过 RSVP 而不是 LDP 来完成。

当传输数据报文时，LSP 上的第一个 LSR 使用从下游邻居处接收的标签。这个标签能够用于在下一跳找到对应的预约信息，进行正确的转发和报文调度，并找到由下一跳提供的输出标签。

（4）灵活路由（显式路由）

基于目的地的路由一个最基本的特征是用于转发报文的信息只是报文中的目的地址。虽然这样使路由具有高度可扩展性，但也限制了报文所经过的实际路径。由此，也就限制了在多条链路上分布流量，使流量从负载重的链路转向负载轻的链路。

对于 QoS 路由，基于目的地的路由也是不够的，增加特定预约的路径数，可以改善预约成功的可能性。增加路径数，反过来，要求路由器具有发现多条路径的能力，而不仅仅依赖于目的地地址。为了支持与基于目的地的路由形成路径不同的转发路径，LSR 的控制组件允许把标签绑定到非目的地决定的路径上。

可以看出，MPLS 具有许多优点，可以支持路由层次、流量工程以及服务质量等。

9.2.2　转发粒度和等效前传类

传统的路由器定义了一些结构，用来存放转发信息，当报文到达时，路由器使用分类算法把报文映射到对应的转发表项，该项指出如何处理报文。这种分类算法可以看作把可能的报文空间分成等效前传类（FEC）。

LSP 上的每个路由器要确定 FEC 的下一跳。对于给定的 FEC，转发表中的对应项可能是动态创建的，也可以是由配置和配置与协议结合创建的。有许多方法定义 FEC，下面是一些例子：

- 使用目的地址前缀；
- 使用路径上的某个公共路由器；
- 使用源/目的网络地址；
- 使用源/目的地址、源/目的端口、传输层协议；
- 使用策略。

在同一个 LSR 中，各种 FEC 可以共存，而且，不同的 LSR 用于分类报文的过程可以是相同的。

MPLS 没有局限于某一特定协议，是一个多协议解决方案。MPLS 的转发组件很简单，可以达到很高的效率；控制组件则很灵活，可以支持各种路由功能。通过各种转发粒度使 MPLS 既有可扩展性又具有丰富的功能。控制功能与转发功能的相互独立又为新需求的加入

带来了方便。

9.3 第三层交换

在众多新的网络技术中，第三层交换技术起了至关重要的作用。第三层交换技术也称为 IP 交换技术，它将第二层交换机和第三层路由器两者的优势结合成为一个有机的整体，是一种利用第三层协议中的信息来加强第二层交换功能的机制，是新一代局域网的路由和交换技术。第三层交换，也称为 IP 交换技术、高速路由技术等，主要是基于第三层（L3）地址转发流量、执行交换功能、可能提供特别服务（例如认证）、也可能不具备路由处理功能的一种技术。

9.3.1 出现的原因

IP 交换技术最早出现于 1996 年，它的发展受当时一些因素的推动，一些与商业有关，一些与技术有关。从技术上讲，由于 Internet 及许多内部网处于一定的或近似的拥塞状态，存在一个很大的动力发展更快和更便宜的 IP 转发设备。这并不奇怪，因为事实上许多 IP 网络设计上就有拥塞。提供一个能百分之百地处理处于突发速率的业务流的网络将过于昂贵，因此它们以相对少的容量被建立起来，期望最多的端用户在最多的时间里，通过网络获得其最多的业务量。如果发生拥塞，TCP 主机将发送速率适配到网络容量允许的大小，拥塞消除，人们可以继续他们的商务。当然，如果主机不使用 TCP，并继续将分组注入一个已经阻塞的网络，而忽略或不理解隐含或显式的拥塞指示，拥塞可能不会消失。事实上问题可能更严重。

根据标准网络理论，性能瓶颈可以通过加快速度来减少或消除。提高数据分组的转发速率通常可以对网络性能起到一系列积极的作用。更快的传送速率意味着分组在队列中等待的时间会更少。队列将更快地清空，为瞬间的突发业务留出更多可用的队列容量。业务流的突发能得到更好的处理，不会由于队列容量不足而丢弃分组，造成重传或更坏的情况。整个网络的性能将得到改善。

现在事情并非真的这么简单。在任何网络总会有拥塞点，已经有建议提出，网络交换或路由结点在队列调度或管理中应起到更积极的作用。“拥塞忽略”主机将必须设法得到管制等等。但人们无疑将需要更多的带宽和更快的转发设备来处理业务量的增加并减少拥塞。

在吉比特路由器出现以前，使用 ATM 技术被许多人看作是理想的解决方案，以回避与当时路由器技术相关的性能瓶颈。一个标准的 ATM 交换机提供多吉比特转发速率，比当时的路由器快几个数量级。ATM 也提供极好的性能价格，且端口速率提高到 OC12 或更高，是十多年中工业研究和标准发展的一个主要焦点。无论如何，在 IP 路由的 Internet 环境中使用 ATM 技术对那些开始研究 IP 交换解决方案的人来说是最可取的。IP 交换出现的技术原因包括下列内容：

- 路由器是瓶颈，不能支持 OC3+转发速率的业务量。
- 路由表太大，查询时间太长。如果许多路由被映射到一个更小的标记组（如 VPI/VCI），且一个查询包括在硬件中索引 VPI/VCI 表，则会有所改善。
- 下一代 IP 网络和应用需要 ATM 所能提供的带宽管理、性能和 QoS。
- 需要一个简单的方法在一个面向连接的网络中支持无连接 IP 业务流。
- 需要一个简单的方法为需要直通路径的应用构造直通路径。

■ ATM 论坛的 UNI/PNNI 信令和路由过于繁琐。

■ ISP 核心网络已经或正计划开发 ATM，需要一个简单且可扩展的方法在 ATM 面向 VC 的网络上支持大数量的 IP 业务流。

■ 需要开发平台和网络以支持正在兴起且先进的 IP 功能，如 IP 综合业务和流量工程。

毫无疑问还有其他原因。但 IP 交换发展背后的主要技术推动力是这样一个愿望，即通过使用交换特别是 ATM 交换作为一些或者所有业务量的转发机制从而有效地提高性能。

IP 交换技术的出现也存在着商业上的原因，因为运营商和路由器及交换机厂商需要：

■ 确认在 ATM 上的投资。

■ 重定位 ATM 作为一个适合于 IP 的技术。

■ 采用提供 IP 路由和高容量 ATM 交换的综合平台，开发实际的及认识到的 Internet 和内部网的性能问题。

■ 提供新技术与已有的路由器机制竞争。

■ 提供新技术以迷惑市场并通过模糊其缺点来保护路由器机制。

很有可能，各种形式的 IP 交换将成为公共网和专用网的运营者因其增加的容量和特殊业务而开发的一个非常有用的技术。

灵活智能的路由引擎（FIRE）标志着第三代交换技术的来临。第三代交换技术并不仅是建立在第二代的进展上，而且为第三层路由、组播(Multicast)及用户可选的策略(Policy)等方面提供了线速性能，第二层与第三层的性能不再是不一致的了。FIRE 是 3Com 公司的第三代第三层交换机的核心部分，它是一个创新的集成化的网间互连体系结构，提供了广泛的第二层和第三层的功能，同时还可在多种网络接口类型上提供线速性能。

9.3.2　主要技术

根据 Strategic Networks 公司的创立者和董事长 Nick Lippis 在 1997 年初提出的观点，可以把第三层交换分为两种基本类型：一种在第三层对每一个信息报文进行处理，而另一种则不在第三层对每一个信息报文进行处理。我们将前者称为报文到报文第三层交换（Packet-by-Packet），而将后者称为第三层流交换（Flow Switch）。

仍然从开放系统互连（OSI）参考模型（图 9-2）说起。第三层报文到报文技术对每一个报文进行路由处理，并基于第三层地址转发所有报文。

层	名称
7	应用层
6	表示层
5	会话层
4	传输层
3	网络层
2	数据链路层
1	物理层

图 9-2　OSI 参考模型

图 9-3 显示了报文到报文的流活动。报文进入系统中 OSI 参考模型的第一层，即物理接口，然后在第二层接受检查（即送达目的 MAC 地址），如果不能交换则进到第三层。在第三层，报文要经过路径确定(即路由计算)、地址解析(即通过查表或其他机制确定对应第三层地址的第二层地址)以及某些特殊服务(如认证、获取记帐数据、变换成另一种第二层格式)。第三层处理完毕后，报文已被更新(报文头已重写、计数器已调整等)并准备传回到第二层。确定合适的输出端口后，报文通过第一层传送到物理介质上。

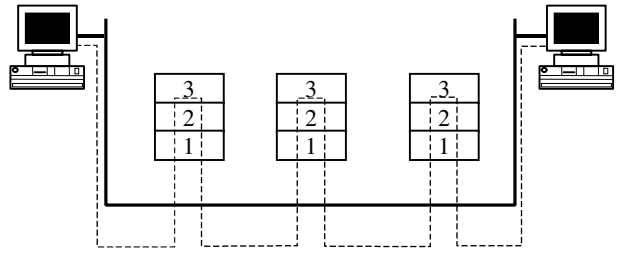


图 9-3　被路由的报文在每一个节点的第三层接受处理

在报文到报文第三层交换中，又可以作如下区分：

- 路由器中的交换：大部分功能由软件完成，稍近的可能会使用硬件缓存以提高报文转发的性能。
- 学习型网桥的交换：基于第三层地址转发信息流，但不进行路由处理。
- 路由交换机的交换：根据第三层地址转发信息流，并且进行路由处理。

传统路由器一直用于局域网间连接不同的 IP 子网络，最近也用于连接不同的虚拟局域网(VLAN)。学习型网桥这一名词用于描述一小类仅比第二层交换多一路由功能的产品，它被动地获取第三层映射的能力迅即成为其它类型产品的一个可选功能，因而不受特别的注意。

路由交换机强调组合路由和交换，一般只支持有限的路由协议，即支持选定技术所需要的那些协议。例如，一个只路由 IP 的交换机只需要支持 RIP 和 OSPF，就可以和同一网络中的其他路由器相配合。这又引出了设计的经济性以及扩充了使用硬件而非软件构造更多功能的能力。其设计技术包括：

- 削减处理的协议数，包括第三层(常常只对 IP)和第二层(例如：只对以太网)；
- 限制第三层支持的特殊服务；
- 尽可能将其做到硬件内(如 ASIC)而不采用软件实现。

完成交换、路由处理和第三层数据流转发的这一类产品称之为路由交换机。目的是降低路由处理代价，提高路由性能，以匹配第二层交换机。

流交换方法是为基于帧和基于信元的网络而设计的，比报文到报文方法具有更多的变化。流交换分析流中的第一个报文，以便完成路由处理并基于第三层地址转发该报文。流中的后续报文使用一种或多种捷径技术进行处理，其设计目标是方便线速路由处理。。一般将其分作两个主要类型：端系统启动的流交换和网络中心式流交换。端系统启动机制通常不需要改变网络的交换基础设施，而网络中心式的方法则必须由整个网络中的交换机安装和支持，才能达到主要的性能改善。

在流交换中，第一个报文被分析以确定其是否标识了一个“流”或者一组具有相同源地

址和目的地址的报文。例如，属于某一特定 TCP 会话的报文。如果第一个报文具有正确的特征，则该标识流中的后续报文将拥有同样的优先权，如访问权。流交换节省了检查每一个报文要花费的处理时间。此外，同一流中的后续报文被交换到基于第二层(甚至第三层，取决于特定的 FS 实现)的目的地址。第一个技巧是要识别第一个报文的哪一个特征标识一个流，这个流可使其余的报文走捷径，即第二层路径。第二个技巧是，一旦建立穿过网络的路径，就让流足够长以便利用捷径的优点。流交换方法又称为“直通路由”技术，因为后续报文无需路由选择而直接进行交换。

总之，引入第三层交换技术的目的就是只要在源地址和目的地址之间有一条更为直接的第二层通路，就没有必要经过路由器转发数据包。第三层交换使用第三层路由协议确定传送路径，此路径可以只用一次，也可以存储起来，供以后使用。之后数据包通过一条虚电路绕过路由器快速发送。表 9-1 给出了当前一些主要的第三层交换技术的主要技术特点。

表 9-1　　第三层交换技术的几种实现

厂　商	名　称	主要技术特点
Ipsilon 公司	Ipsilon IP 交换	IP 交换技术由 Ipsilon 公司首倡，即识别数据包流，尽量在第二层进行交换，以绕过路由器，改善网络性能。Ipsilon 改进了 ATM 交换机，删去了控制器中的软件，加上一个 IP 交换控制器，与 ATM 交换机通信。该技术适用于机构内部的局域网和校园网。
3Com	Fast IP	侧重数据策略管理、优先原则和服务质量。Fast IP 协议保证实时音频或视频数据流能得到所需的带宽。Fast IP 支持其它协议(如 IPX)，可以运行在除 ATM 外的其它交换环境中。客户机需要有设置优先等级的软件。
IBM	ARIS Aggregate Route 一 based IP Switching	与 Cisco 的标签交换技术相似，包上附上标记，借以穿越交换网。ARIS 一般用于 ATM 网，也可扩展到其它交换技术。边界设备是进入 ATM 交换环境的入口，含有第三层路由映射到第二层虚电路的路由表。允许 ATM 网同一端两台以上的计算机通过一条虚电路发送数据，从而减少网络流量。
ATM 论坛	MPOA：MultiProtocol Over ATM	ATM 论坛提出的一种规范。经源客户机请求，路由服务器执行路由计算后给出最佳传输路径。然后，建立一条交换虚电路，即可越过子网边界，不用再做路由选择。

目前 Cisco、3Com、北电网络、朗讯、Cabletron、Foundry 和 Extreme 等公司都有比较成熟的第三层交换产品和模块推出。

9.3.3　Ipsilon 公司的 IP switching

1996 年春天，Ipsilon 公司提出了 IP switching 技术。它的思想是将一个 IP 路由处理器捆绑在一个 ATM 交换机上，去除了交换机中所有的 ATM 论坛信令和路由协议，这样的一个结构我们就可以称之为 IP 交换机了。ATM 交换机由与其相连的 IP 路由处理器控制；IP 交换机作为一个整体运转，执行通常的 IP 路由协议，并进行传统的逐级跳方式的 IP 分组转发。当检测到一个大数据量、长持续时间的业务流时，IP 路由处理器将会与其邻接的上行节点协商，为该业务流分配一个新的虚通路和虚信道标识（VPI/VCI）来标记属于该业务流的信元，同

时更新 ATM 交换机中连接表对应的内容。一旦这个独立的处理过程在路由通路上的每一对 IP 交换机之间都得到执行，那么每一个 IP 交换机就可以很简单地把交换连接表中的上行和下行节点的表项入口正确地连接起来。这样，最初的逐级跳选路方式的业务流最终被转变成了一个 ATM 交换的业务流。

IP 交换抛开了复杂的 ATM 论坛的信令、路由和 MPOA 等协议，而让业务流能够动态经过一个 ATM 交换机构进行交换。实际上，IP 交换机就是一个运行通常路由协议的路由器。只是它采用了一个 ATM 交换器来作为分组转发的引擎。人们普遍相信，纯 ATM 信元的交换是进行大量数据转发的一种极其快速和有效的方法——它比当时市场上的任何一种路由器都要快得多。IP 交换也正是利用了这一点。

Ipsilon 开发并颁布了两个协议，即 IFMP 和 GSMP。所定义的 Ipsilon 流管理协议（Ipsilon Flow Management Protocol）使得两个相邻的交换机能够对同一个数据流的信元进行分类并重新标记 VPI/VCI 值，而所定义的通用交换机管理协议（General Switch Management Protocol）使得 IP 交换机的路由器实体能够控制 ATM 交换机内部的资源和连接表。在统一运行标准 IP 选路协议的相邻 IP 交换机网络上，IFMP 和 GSMP 提供了一种附加的功能，对数据流进行分类以判断是否可以交换而不是选路，进而利用 ATM 交换机所构成的第二层交换式通路进行转发。下面我们分别介绍这两个协议。

1．流量管理协议（IFMP）

Ipsilon 数据流管理协议（Ipsilon Flow Management Protocol －IFMP）作为 IP 交换机网络中分发数据流标记的协议，它在跨越外部数据链路的相邻的 ATM 交换机控制器（或者 IP 交换机入口和出口）之间工作。具体来说，下游 IP 交换机（接收端）利用它通知上游交换机（发送端）在某一段时间段内为某个数据流赋予某个 VPI/VCI 值。数据流用流标识符来标记。下游 IP 交换机必须在一定时间段内更新数据流的状态，否则数据流的状态将被删除。

IFMP 仅在相邻的 IP 交换机之间操作，与其他 IP 交换机的工作方式和功能无关。换句话说，IP 交换机从下游 IP 交换机接收到 IFMP 重定向消息后不会向其上游邻机发送类似的消息。IFMP 的基本目的是分发与某个数据流相关的新的 VPI/VCI 值，以便加速转发功能并有可能基于每个数据流进行交换，从而提高总吞吐量。

2．通用交换管理协议（GSMP）

通用交换机管理协议（General Switch Management Protocol，GSMP）是一个简单、通用的交换机管理和控制协议，用于对 IP 交换机内部交换机资源的管理。该控制协议是 Ipsilon 的 IP 交换解决方案的组成部分，而 GSMP 是两个解决方案特定的控制协议之一。IFMP 通过外部 ATM 数据链路运行在相邻的 IP 交换机控制器之间，而 GSMP 运行在 IP 交换机内部的 IP 交换机控制器和 ATM 交换机之间。GSMP 使 IP 交换机控制器能够建立和释放穿越该交换机的连接、在点到多点连接中加入和删除叶节点、管理交换机端口、请求配置信息。ATM 交换机也可以给使用 GSMP 的 IP 交换机控制器主动地提供事件信息。

IP 交换机的模块包括 IP 交换机控制器和 ATM 交换机，这两个模块交互 GSMP 协议消息以管理交换机资源。二者有主从关系，IP 交换机是主，ATM 交换机内的 GSMP 代理是从 GSMP 协议消息通过在 IP 交换机控制器和 ATM 交换机之间建立的标准控制 VC（VPI/VCI＝0/15）传递。RFC1483 定义的 LLC/SNAP 封装用于封装 GSMP 消息的头和数据段。

第 10 章　路由器QoS与安全

Internet 在过去几年所取得的巨大成就和未来所蕴涵的巨大发展潜力，几乎没有人怀疑。当人们在思考未来 Internet 的发展时，如何在 IP 网络上保证用户信息传输的质量就成为一个不容忽视的重要问题。为解决这一问题，QoS（Quality of Service，服务质量）技术便应运而生。更高的带宽及将音频、视频和数据集成在同一种网络媒介上的技术的出现，大大推动了对支持保证服务的多媒体应用的需要。QoS 就是保证服务质量的一种技术，在路由器上实现 QoS 是网络技术发展的一种必然趋势。

QoS 是指一个网络能够利用各种各样的基础技术向选定的网络通信提供更好的服务的能力。这些基础技术包括：帧中继（Frame Relay）、异步传输模式（Asynchronous Transfer Mode，简写为 ATM）、以太网和 802.1 网络、SONET 以及 IP-路由网络。

数据网络利用率的增长使得出现了将各种数据通信量（音频、视频、数据）放置于提供数据服务的底层基础之上的趋势。在试图合并音频、视频、数据于网络时，网络工程师面临的障碍是不同类型的通信需要不同级别的网络服务。

目前所用的路由器（包括国产的和国外的）几乎都是普通路由器和不具有加密功能的“安全路由器”。这些路由器有的只有路由功能而没有安全、加密功能；有的只增加了包过滤功能，系统在面对各种各样的攻击时非常脆弱。如何保证网络设备自身的安全以及存储在其上或通过其传输的敏感信息的安全，就成为必须解决的问题。保证路由器的安全，也可以说是在路由器上实现一些安全机制成为网络技术领域一个不可回避的重要问题。

对 QoS 和安全的策略以及它们的相关协议的完整讨论超出了本书的范围，本章只是简单介绍实现 QoS 和安全的关键机制。

10.1　路由器 QoS

考虑一个最基本的网络：由两台相互连接的计算机组成的网络。在理想情况下，这些计算机可以发送、接收任意数量的数据而无需担心延迟；在文件传输数量、声音呼叫、视频会话中所受的惟一限制，是系统自身的处理能力。它们有一个无限带宽的开放连接。

实际中的网络由有限带宽且有一定量延迟（可测量）的共享连接组成。能够处理所有连接点的完全输出的网络不经济，网络设计者力图在平均及峰值传输速率间平衡开销。网络设备，如路由器，采用多种缓冲排队机制以处理高于均值的暂时突发状况。

通常使用两种 QoS 来满足传输的要求：拥塞管理与拥塞避免。

拥塞管理功能的作用是一旦发生拥塞，可对拥塞进行控制。拥塞管理假定报文在队列中

等待，尝试依据报文服务级别顺序出队以减少队列的影响。网络组件处理接收通信量溢出的一个方法是：先使用排队算法给通信分类，然后，确定对通信按照优先次序排在输出链接上的方法。每一种排队算法都能解决某一个特殊的网络通信问题，并且对网络的性能有特殊的作用。

依据路由器如何对报文分类及报文出队顺序，拥塞管理或者排队功能包括：FIFO、优先权排队（Priority Queuing，简写为 PQ）、自定义排队（Custom Queuing，简写为 CQ）以及加权公本排队（Weighted Fair Queuing，简写为 WFQ）。

避免拥塞可尝试向通信量的源发送信号，降低发送速度以限制队列自身的大小。拥塞避免的目标是在拥塞发生前就做到防止拥塞。通常，在路由器上有两类通用拥塞避免方法：平台相关与平台无关。平台无关策略实现于平台的主处理器，而平台相关策略实现于指定平台的智能接口处理器。

10.1.1 拥塞管理

基于分配给数据包的优先权，通过确定数据包发送出某个接口的顺序，拥塞管理功能可以允许对拥塞进行控制。拥塞管理功能承担如下的任务：创建队列、基于数据包的分类把数据包分配给相应的队列以及调度队列中的数据包以便传输。拥塞管理 QoS 功能提供四种类型的排队协议，每一种都允许指定创建数目不等的队列。

依据路由器如何对报文分类及报文出队顺序，有四种类型的排队方式，它们构成了拥塞管理 QoS 功能：

■ FIFO（先入先出排队）方式：FIFO 不需要通信优先权以及分类的概念。使用 FIFO 时，数据包发送出接口的顺序依赖于数据包抵达这个接口的顺序。

■ WFQ（加权公平排队）方式：WFQ 提供了动态的、公平的排队方式，它基于权重来划分通信队列的带宽。WFQ 可以保证，所有的通信都能够根据它的权重而受到公平的对待。为了帮助理解 WFQ 是如何工作的，我们可以把 FTP（File TransferProtocol，文件传输协议）数据包链的队列视为集合的队列，同时，可以把离散交互式通信数据包的队列视为单个的队列。根据这些队列的权重，WFQ 可以保证，作为集合发送出去的所有 FTP 数据包的数量等于发送出的单个交互式通信数据包的数量。根据这种处理方式，WFQ 可以保证苛刻的应用得到比较令人满意的响应时间。这里所指的苛刻的应用的一个例子是交互式、基于事务处理的应用，这种程序不能够容忍性能降级。对于带宽为 E1（2.048Mbit/s）或者更低的串行接口来说，WFQ 是默认使用的。当没有配置其他排队策略的时候，所有其他接口都默认使用 FIFO 排队方式。

■ CQ（自定义排队）方式：在 CQ 排队方式下，对于每一种不同的通信种类来说，带宽是按照比例来分配的。CQ 排队方式允许指定从队列中抽取出来的字节或者数据包的总数。对于速度比较慢的接口来说，这种功能是非常有用的。

■ PQ（优先权排队）方式：在 PQ 排队方式下，属于某个通信优先权等级的数据包可以比所有优先权等级低的数据包先发送出去，以保证优先权级别高的数据包能够及时地发送出去。

所有这些拥塞管理技术，均应用于接口的输出保持队列与发送队列间。

1．先入先出排队方式

先入先出排队方式——即众所周知的先来先服务（First-come,First-served，简写为 FCFS）

排队方式——在最简单的格式中包含了如下两个内容：当网络发生拥塞的时候，存储数据包；当网络不再拥塞的时候，按照到达接口的顺序把存储的数据包发送出去。FIFO 体现了没有通信优先权或者等级的概念，并且因而不需要决定数据包的优先权。在这种排队方式下，只有一种队列，并且所有的数据包都得到平等的对待。数据包按照抵达某个接口的顺序，从这个接口发送出去。优先级较高的数据包不会比优先级低的数据包更早地发送出去。

FIFO 排队方式在用户数据通信上实现了数据包的无优先权策略。它不需要确定通信的优先权或者种类。当使用 FIFO 策略的时候，运行异常的数据源可能会耗费可用带宽，突发的数据源可能在时间敏感性的或者重要的通信中引发延时现象，并且，因为不重要的通信充满这个队列，重要的通信也许会被丢弃。

当网络上没有配置其他排队策略的时候，除了数据传输速率为 E1（2.048Mbit/s）或者更低的串行接口以外，所有其他的接口都默认使用 FIFO 策略。FIFO 是排队策略中最快的方法，对于延时很小以及拥塞情况很少发生的大型链接来说，是非常有效的。如果网络链接很少发生拥塞现象，FIFO 排队方式也许是惟一需要使用的排队方式。

2．加权公平排队方式 WFQ

WFQ 是一个自动的时序安排方法，它对所有的网络通信都提供了公平的带宽分配方案。WFQ 把优先权或者权重应用到确定的通信上以把通信划分成会话，并且决定每一个会话相对于其他的会话来说可拥有多少带宽。WFQ 是基于数据流的算法，它能够把交互式通信调度到队列的前端来减少响应时间，同时也能够在高带宽数据流中公平地分享剩余的带宽。换句话说，WFQ 为数据量少的通信（例如 Telnet 对话）提供比数据量大的通信（例如 FTP 对话）更高的优先权。WFQ 为并发的文件传输提供链接容量的平衡使用；也就是说，当多个文件传输任务同时进行的时候，为这些数据传输任务提供同等的带宽。

WFQ 克服了 FIFO 排队方式的严重局限性。当 FIFO 排队方式生效的时候，根据抵达接口的顺序将存储的数据包发送出去，而与带宽的耗费或者相关的延时无关。其结果是，文件传输和其他数据量较大的网络应用常常会生成相关数据的数据包系列。这些相关的数据包称为数据包链。数据包链是多组数据包，它们在网络上往往会作为整体进行传输。这些数据包链将可能耗费所有可用的带宽，从而剥夺其他通信的带宽。

WFQ 提供通信优先权管理功能，动态地把通信划分成组成一个会话的消息。WFQ 打破会话内的数据包链，以便保证带宽在单个的会话之间公平分享，保证数据量少的通信能够以一种及时的方式进行传输。

基于数据包报头的编址，WFQ 可以把通信划分成不同的数据流。数据包报头的编址包括如下一些特性：源和目的网络或者 MAC 地址、协议、源和目的端口以及一个会话中的网络接口数目、帧中继数据-链接标识符（data-linkconnection identifier，简写为 DLCI）的值以及服务类型（type of service，简写为 ToS）的值。存在两种类别的数据流：高-带宽会话及低-带宽会话。低-带宽通信拥有比高-带宽通信更高的优先权，并且根据所分配的权重，高-带宽通信按比例地分享传输服务。低-带宽通信数据流构成通信的绝大部分内容，并且获得优先服务，能够以及时的方式把整个负载传送出去。数据量大的通信数据流按照一定的比例分享剩余的数据容量。

在进行传输以前，WFQ 把各种会话的数据包放置在公平排队中。数据包从这些公平排队中被移走的顺序取决于每一个抵达数据包的最后一位所表示的传送虚拟时间。当达到拥塞消

息门限值以后，发送到高-带宽数据流的新消息将会被丢弃。但是，低-带宽数据流，包括控制-消息会话，仍然可以继续对数据进行排队。其结果是，公平排队有时所包含的消息的数目会超过门限值所指定的数目。

WFQ 能够管理双向传输的数据流（例如，在应用对之间传输的数据流）和单向传输的数据流（例如语音和视频数据）。WFQ 算法解决了往返延时发生变化的问题。如果多个大数据量的会话在同时进行数据传输，则它们的传输速率和交互抵达周期变得更加可以预测。WFQ 大大增强了某些算法，如 SNA 逻辑链接控制（Logical Link Control，简写为 LLC）和传输控制协议（Transmission Control Protocol，简写为 TCP）的拥塞控制和慢速启动功能。

在绝大多数所配置的运行速率为 E1（2.048Mbit/s）或者更低的串行接口上，WFQ 是作为默认的排队模式来使用的。在某些情况下，我们要求系统对任务繁忙和轻松的用户能够提供相同的响应时间，而不需要增加额外的带宽。WFQ 可以自动适应网络通信状况的变化。

3．自定义排队 CQ

自定义排队是一种与优先排队非常类似的平台无关拥塞管理机制。自定义排队为队列服务提供了公平性。不像优先排队可能会饿死非高优先级队列。自定义排队允许 16 个可设置队列，每个队列有特定的字节数目，是在系统移到另一队列前所应从队列中发送出的数据量。换句话说，16 个队列按时间轮转方式（round-robin fashion）得到服务，且在移到下一队列前，设定数目的字节被服务。CQ 所有的通信都保证某种等级的服务，因为可以给所有种类的通信分配带宽。通过决定队列所配置的数据包总数的容量，就能够决定这个队列的大小，因此可以控制对带宽的访问。

每次当某个队列接受服务的时候，CQ 允许指定从这个队列发送的字节总数，因此允许用户以最小的带宽或者等待时间需求在应用之间分享网络资源。也能够指定每一个队列所包含的数据包的最大数量。

工作原理是通过为每一个通信等级指定可以被服务的数据包或者字节总数，CQ 可以处理通信。CQ 以一种联合签名的方式在各个队列之间循环，在移动到下一个队列以前，为每一个队列传输所分配的带宽部分，从而向队列提供服务。如果某个队列是空的，则路由器将会从已经准备好发送数据包的相邻队列中发送数据包。当 CQ 在某个接口上生效的时候，系统将会为这个接口维护 17 个输出队列。用户可以指定队列 1 到 16。与每一个输出队列相关的是可配置字节总数。可配置字节总数指定在系统移动到下一个队列以前，系统应当从当前的队列中发送多少字节的数据。编号为 0 的队列是一个系统队列；在任何编号为 1 到 16 之间的队列得到处理以前，编号为 0 的队列将先被清空。系统把优先权级别高的数据包，例如保持活动数据包以及信令数据包，安排到这个队列。其他的通信不能够使用这个队列。

对于编号为 1 到 16 的队列来说，系统相继地在这些队列之间循环（以一种联合签名的方式），在每次循环中都从当前队列中取出配置好的字节总数，并且在移动到下一个队列以前把这些数据包发送出去。当处理某个特殊队列的时候，系统就会一直发送数据包，直到发送的字节总数已经超出这个队列的字节总数或者这个队列已经清空为止。一个特殊队列所使用的带宽只能够根据字节总数和队列长度来间接指定。

当线路负载很重的时候，CQ 能够保证没有任何一个应用或者指定的应用组可以得到超过总带宽的预定比例的带宽分配。与 PQ 相似的是，CQ 是静态配置的，因此不能够自动地适应不断变化的网络情况。在绝大多数的平台上，所有的协议都在快速交换通路里被分类。

4．优先权排队 PQ

PQ 是一种让某些报文绝对优先于其它报文的平台无关拥塞管理策略。PQ 允许定义在网络中如何区分数据包的先后次序。可以配置四种通信优先级。根据数据包的特性可以定义一系列的过滤器，使路由器把通信放置在这四种队列中；优先权级别最高的队列首先得到服务，直到这个队列为空为止，然后依次给优先权级别较低的队列提供服务。

PQ 保证严格的优先级，因为它保证某种类型的通信将会发送出去，并且可能会牺牲所有其他类型的数据包的利益。对于 PQ 来说，完全可以使得一个优先权级别较低的队列处于不利地位，并且，最坏的情况是，无论是在可用带宽的总数有限的情况下，还是在紧急通信的传输频率很高的情况下，都不允许先传输优先权级别低的队列的数据包。

PQ 在数据传输期间，给优先权级别高的队列提供比优先权级别低的队列绝对优惠的待遇；只要给予了最高的优先权级别，重要的通信总会比重要性低的通信提前得到服务。根据用户所定义的标准，对数据包进行分类，然后把它们放置在四种输出队列的其中一种队列中——高、中、一般以及低——基于所分配的优先权级别。未分配优先权级别的数据包将放置在一般队列中。当系统将把数据包发送到一个接口的时候，系统就按照优先权级别由高向低开始扫描这个接口上的优先权队列。先扫描高优先权级别的队列，然后扫描中优先权级别的队列，以此类推。然后，选中位于最高优先权级别队列最前面的数据包，并传输。每当要发送一个数据包的时候，就重复这个过程。一个队列的最大长度是由长度限制所定义的。当一个队列的长度超过长度限制的时候，所有额外的数据包都会被丢弃。

这个策略有一个缺陷：低优先级队列可能被饿死。如果进到某个低优先级队列中，可能等待很长一段时间，系统才会调用这个进程。如果有足够的进程不断进入比这个队列优先级高的队列中，则这个队列中的进程永远不能得到服务，直至被饿死！因此优先排队策略要小心使用。报文分类原则应有所选择以保证不会以很高的速度将报文放到高优先级队列中。

10.1.2　拥塞避免

本节我们讨论一种拥塞避免方法加权随机早检测。

当大量通信是自适应形式时，如 TCP，随机早检测是一种最有效的拥塞避免算法。TCP 将报文丢弃作为网络拥塞的隐含信号，当报文丢弃时，降低通信的发送速率（或回退）。当一个拥塞连接上大多数报文由不同发送者发出的 TCP 流组成时，FIFO 排队在队列满时尾丢弄，导致所有发放者同时回退。过了一段时间（t），TCP 发送者增大发送速率，当连接再次拥塞时，所有 TCP 发送者同时回退。这种摆动现象叫做全局同步。

随机早检测就是用来防止这种全局同步现象发生的一种方法。RED 的策略很简单：报文在队列满之前随机地被丢弃，而不是等队列真的满了才丢弃。如果报文丢弃时间是隔开的，则经过队列的 TCP 会话在不同的时间对报文丢弃做出反应，不会同步。

RED 定义了一个最小阈值、一个最大阈值以及一个平均队列大小。RED 基于如下公式计算平均队列大小（而不是瞬时队列大小）：

平均队列大小=[以前的平均队列大小*1/2 指数加权连续量+当前队列长度*1/2 指数加权连续量]

因为 RED 计算的是平均队列大小而不是瞬时队列大小，平均队列大小依赖于以前的平均队列大小。因此，RED 不避开突发通信。在 sally Floyd 及 Van Jacobon 的论文“ Random Early

Detection for Congestion Avoidence”（IEEE/ACM Transaction on Networking V.1,n.4,August 1993）中有关于 RED 的详细描述。

RED 按如下公式进行报文丢弃选择：

- 如果平均队列长度小于最小阈值：不丢弃报文。
- 如果队列长度大于最小阈值，但小于最大阈值，随着平均队列大小的增大，以更大的概率丢弃报文。
- 如果平均队列长度大于最大阈值，所有报文均丢弃。

深入一步，加权随机早检测（WRED）根据 IP 头部优先级对每个报文的被丢弃概率加权，可用于不同类的服务。

10.2 安　　全

由于 Internet 是无中心网，网络之间可以自由通信、自由交互、自由往来，网络安全问题更加突出。在人们追求服务质量的同时，不得不考虑另外一个问题：路由器转发我们的数据时是否是安全的？的确，网络安全目前已经日益成为业界关注的重点，就用户的网络而言，其面临的安全威胁主要源自于未经授权的非法网络访问、传输过程中可能出现的敏感信息泄漏与遗失、数据完整性破坏及计算机病毒的传播等几个方面。而解决这类问题的途径除了借助立法及强化内部管理等防范措施外，先进的安全技术也是网络安全的重要解决之道，如加密技术与基于路由器的安全防范技术等。

10.2.1 路由器的安全威胁

对路由器的安全威胁可以分为对路由器自身的威胁和对路由器功能的威胁两部分。对路由器自身的威胁，主要有：

- 硬件设备的干扰：设备的电磁干扰、电磁信息泄漏，以及在一些较恶劣的情况下的不稳定；
- 攻击操作系统：利用操作系统漏洞，如利用缓冲区溢出进行的攻击；
- 窃取路由器资源：对路由器访问资源的窃取和控制，如业务拒绝（DOS）；
- 攻击路由器服务：针对提供服务的安全漏洞所进行的攻击，如远程管理的漏洞、Telnet 明文传送认证信息等；
- 其它各种威胁路由器本身的攻击行为。

对功能的威胁主要是使路由器错误地转发或接受不希望转发或接受的包，使得这些包进入到系统内，对系统构成威胁。或者利用路由器没有对报文进行安全处理或处理强度低的情况，进行信息窃取。对路由器功能的威胁包括：

- 发布虚假的路由信息，使得数据被送到错误的地方；
- 通过监听、篡改信息、重放信息包使得信息包虚假；
- 窃听或破译没有加密的或加密强度低的信息包；
- 通过拒绝服务攻击使攻击者过多占用共享资源，导致服务超载或系统资源耗尽，使得路由器无法为其它用户提供合适的服务；

- IP 地址欺骗：主要包括利用虚假的地址，以虚假的回答进行响应。如：ARP 欺骗攻击、DNS 欺骗攻击、TCP 连接欺骗盲攻击；利用 IP 分片攻击，使非法数据流得以通过监控，从而穿过路由器进行攻击；
- ICMP 攻击是滥用 ICMP 包发布错误信息，破坏路由器正常的转发功能。

但是，目前所用的路由器（包括国产的和国外的）几乎都是普通路由器和不具有加密功能的“安全路由器”。这些路由器有的只有路由功能而没有安全、加密功能；有的只增加了包过滤功能，系统在面对各种各样的攻击时非常脆弱。如何保证网络设备自身的安全以及存储在其上或通过其传输的敏感信息的安全，就成为必须解决的问题。保证路由器的安全，也可以说是在路由器上实现一些安全机制成为网络技术领域一个不可回避的重要问题。

针对网络潜在的各种安全威胁，安全路由器在实现常规路由功能的基础上，在设计时强化数据传输加密以及各种协议处理的认证与监测等关键技术问题，增强信息保护与数据加密性能，能够有效检测及防范各类攻击事件的发生。已经有一些安全技术来增强路由器的安全，比如：访问列表、IPSec、一些加密算法、MPLS 的一些机制、隧道技术和 VPN 等等。但这些技术也只能说是增强路由器的安全特性，任何技术都保证不了路由器的绝对安全。

现在还没有一个完整的规范来验证和一致化路由器的安全特性，同时对究竟什么是安全路由器也没有严格的定义。当前人们通常认为安全路由器是集常规路由与网络安全防范功能于一身的网络安全设备，大部分安全路由器产品都是通过在现有常规路由平台之上加装安全加密卡，或相应的软件安全系统等各种形式的安全模块实现的。一般说来，具备 IPSec（IP Security）协议支持、能够有效利用 IPSec 保证数据传输机密性与完整性或能够借助其它途径强化本身安全性能的路由器都可以称之为安全路由器。

与常规路由器产品相比，安全路由器能够提供常规路由器所不具备的诸如 IPSec 协议支持、基于规则集的防火墙、基于 OSPF V2 协议的安全认证、BGP-4 协议认证（防止虚假路由信息的接收）、信息加密与分布式密钥管理等功能，能够对 IP 数据报进行智能加密，可提供安全 VPN 通道、抗源地址欺骗、抗源路由攻击、抗极小数据和抗重叠分片的分组过滤功能及实现基于硬件的信息加密等功能。另外，还能够隐藏内部网络拓扑结构、能够实现身份鉴别、数据签名和数据完整性验证、具有灵活的密钥配置、支持集中式密钥与分布式密钥管理。总的来说，安全路由器的作用主要体现在以下各方面：

- 能够按照需要向内部局域网或外部公用网转发授权包；
- 能够对每一条链路的访问权限进行控制；
- 能够协助对传输的数据进行加密、数据完整性、数据源认证的处理；
- 提供内外识别地址的转变；
- 管理方便，能够提供包括配置管理、性能管理、流量控制、安全管理等功能；
- 确保路由信息能够安全、准确地传递；
- 提供包括分组过滤、优先级、复用、加密、压缩等功能；
- 对正常数据包的转发、处理效率影响尽可能小。

10.2.2　提高路由器安全性的主要技术

安全路由器中的安全技术涉及到硬件、操作系统、网络的各个层次等几方面。从防范的时序上包括事前预防、事中告警、事后追踪等各项技术。事前预防主要是事先对硬件、操作

系统、各网络层次进行安全设置，使得攻击难以进行；事中告警则是在对进出的数据包进行安全分析的基础上，发现可疑点后，发出告警或进行动态策略调整,防止攻击的扩散；事后追踪则能够根据日志或其他的记录信息在事后进行分析，追踪攻击者的行迹，并协助提供攻击证据，意图通过法律手段对攻击者进行制裁。硬件的安全一方面指硬件设备本身要能够在不同温度、湿度条件下，防震、防电磁干扰、防雷、防电源波动以及在通信电缆的绝缘电阻、介电强度等方面有要求，其物理安全防护应符合 GB4943-90 和 GB9254-88；另一方面在硬件设计上考虑对安全的辅助，能够通过硬件动态地隔离开网络，使网络可以通过物理分离达到安全目的。

操作系统的安全很复杂，因为很多操作系统的不安全性来自于自身的漏洞。而这些漏洞可能就是由于操作系统实现时的疏忽而造成的。如：由于没有对输入的数据量进行长度控制造成的缓冲区溢出、由于用户密码管理文件的访问权限的门槛太低造成的密码文件泄漏。

在链路层，针对在本地终止的 PPP 连接，当作为访问路由器或要实现 VPDN（虚拟专用拨号网络）等业务时，就会有该层的安全要求。在 PPP 层要能提供实体身份认证、加密、压缩等。PPP 安全所涉及的实体身份认证协议主要有：PAP（口令认证协议）、CHAP（质询握手认证协议）。安全的链路层协议必须保证用户名、口令非明文传输，推荐采用 CHAP 机制实现。验证方式可以采用本地数据库或者 AAA 协议。PPP 提供的加密服务由 ECP（PPP 加密控制协议）来完成。通过 ECP 协议，PPP 在链路建立完成时，可以在通信的两点之间协商合适的算法，对数据进行加密（具体见 RFC1968）。由于 ECP 本身的协商不受保护，ECP 协议也没有密钥管理的描述，这使得 ECP 协议的安全性存在漏洞，需要进一步完善。

网络层的安全最能显示路由器的安全作用。它的作用是为 IP 层提供可操作的、高效的和基于密码技术的安全性。在这一层提供的安全服务包括了访问控制、无连接完整性、数据来源认证、防重放保护、加密和有限的业务流机密性。而且这些服务在 IP 层提供时，要为 IP 层或更高层协议统一提供保护。网络层安全协议目前主要支持 IPSec（网际协议安全体系结构）及密钥交换协议（见 RFC2401～RFC2412、RFC2451 等）。另外安全路由协议技术、NAT（网络地址转换）技术、IP 过滤器技术也可以体现在网络层中。

从目前的实现以及未来的发展趋势来看，IPSec 和 MPLS 的相关机制将会是将来提高路由器安全特性重点研究的技术。绝大多数安全路由产品都提供了对 IPSec 协议的支持，因为数据传输加密技术的目的就是实现网络传输流量的加密，保障网络内数据流量的安全，而 IPSec 协议正是 IETF Internet 工程任务小组为保证公用网数据传输机密性与安全性而制定的系列协议，它能够在 IP 层提供安全服务，为 IP 及上层协议、应用提供安全保护。

1．IPSec

IPSec 是一个协议套件，它提供了一种标准机制并定义了一套默认的、强制实施的算法，为 IP 及上层协议提供安全保证。假若想增加新的算法，其过程是非常直接的，不会对共通性造成破坏。为保障 IP 数据报的安全，IPSec 定义了一个特殊的方法，它规定了要保护什么通信、如何保护它以及通信数据发给何人，从而可保障主机之间、网络安全网关（如路由器或防火墙）之间或主机与安全网关之间的数据包的安全。由于受 IPSec 保护的数据报本身不过是另一种形式的 IP 包，所以完全可以嵌套提供安全服务，同时在主机之间提供像端到端这样的验证，并通过一个通道，将那些受 IPSec 保护的数据传送出去。

IPSec 协议包括 ESP（Encapsulating Security Payload）封装安全负载、AH（Authentication

Header）报头验证协议及 IKE（Internet Key Exchange）密钥管理协议、ISAKMP/Oakley 以及转码等。它们的关系如图 10-1 所示。

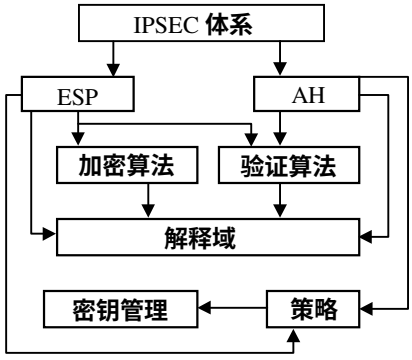


图 10-1　IPSec 各组件的关系

其中 AH 协议能够提供无连接的完整性、数据发起验证及重放保护，ESP 主要用于提供额外的加密保护，而 IKE 则主要提供安全加密算法与密钥协商，转码方式定义如何对数据进行转换，以确保它的安全。转码包括算法、密钥大小（以及它们如何演变）、转码程序以及算法专用的任何信息。对必要的信息来说，确保它们的惟一性显得尤其重要，只有这样才能使不同的实施方案相互间能够操作。

IPSec 的这些机制均独立于算法，协议的应用与具体加密算法的使用均可取决于用户及应用程序的安全性要求。因此，IPSec 是一个开放性的安全标准框架，可以在一个公共 IP 网络上确保数据通信的可靠性和完整性，能够保障数据安全穿越公用网而没有被侦听或窃改之虞，为实现通用安全策略所需的基于标准的解决方案提供了理想的应用框架。而且 IPSec 的部署极为简便，只需安全通道两端的路由器或主机支持 IPSec 协议即可，几乎无需对网络现有基础设施进行任何更动。IPSec 的优势主要表现为可以对所有 IP 级的通信进行加密和认证，可以为 IP 提供基于加密的互操作性强、高质量的通信安全，所支持的安全服务包括存取控制、无连接的完整性、数据发起方认证和加密。这正是 IPSec 协议能够确保包括远程登录、客户机、服务器、电子邮件、文件传输及 Web 访问在内的多种应用程序安全的主要依托。

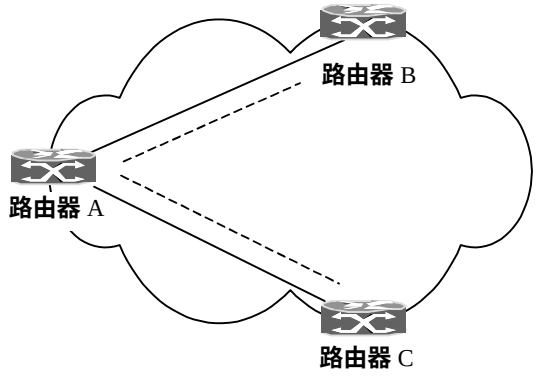


图 10-2　集成在路由器系统中实施

IPSec 可在主机、路由器或主机与路由器同时进行实施。在路由器和主机中实施 IPSec 各自强调不同的问题。在需要确保端到端的通信安全时，在主机实施显得尤其有用。然而，在需要确保网络一部分的通信安全时，在路由器中实施 IPSec 就显得非常必要，比如在 VPN 和内联网中。在主机和路由器实施 IPSec 的特点和方法列于表 10-1。

表 10-1　　IPSec 的实施方法

IPSec 实施方式	特　点	实 现 方 法
主机实施 IPSec	1．能够保障端到端的安全性。 2．能够实现所有的 IPSec 安全模式。 3．能够逐数据流提供安全保障。 4．在建立 IPSec 的过程中，能够维持用户身份验证的具体情形。	1．与操作系统集成，称之为“主机实施”。 2．作为一个中间件，在协议栈的网络层与数据链路层之间实现。
路由器实施 IPSec	1．能对网络间数据流提供完整的安全保护。 2．与 VPN 技术结合，能进行身份验证，并授权用户进入私用网络。	1．IPSec 集成在路由器软件中实现，成为路由器系统的一部分。如图 10-2 所示。 2．实现 IPSec 的路由器作为一个外接模块连接在需要保护的路由器的某个端口，这种方法类似于主机实施的中间件。如图 10-3 所示。

在路由器上实施 IPSec 会严重影响路由器的包转发能力，因为它要对各种数据报都进行一定的处理。而路由器，特别是骨干网上的路由器，通常以尽可能快的速度转发数据包为最终目标。因此在具体实现时，应该考虑到效率的问题。那些不要求安全保障的数据包应不会受 IPSec 的影响。它们仍应以正常速度转发。许多实施方案利用某些硬件设施来执行公共密钥运算、随机数生成、加密/解密以及散列计算。目前有许多特制的芯片组可帮助基本路由器硬件来执行安全运算。

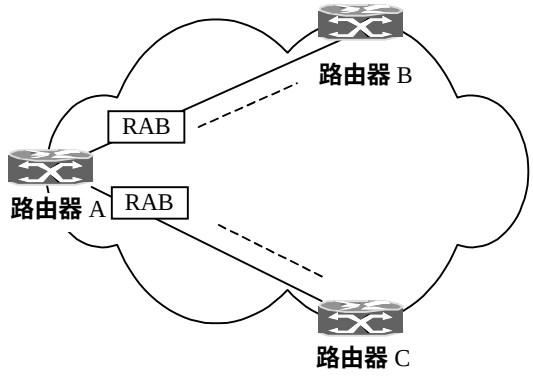


图 10-3　作为外接模块实施

在路由器上实施的另一个问题是 IPSec 的“上下文”问题。IPSec 运行时的“上下文”需要占用大量的存储空间，这对路由器来说也是一个很高的要求。因为路由器的首要任务是实

现报文的路由转发，它必须存储庞大的路由表，而且通常没有专门的硬盘空间来提供对虚拟内存的支持，假如维持的 IPSec“上下文”过多，将会影响路由器的存储能力和生产成本。

2．利用过滤和 NAT 增强安全性

当前几乎所有的路由器产品都提供访问列表，增强对访问的过滤功能，一般是基于 IP 地址和物理端口过滤。即使是基于 IP 地址过滤，过滤器也要能绑定到物理端口上，这样可以减轻如 IP 地址欺骗等类型的攻击。IP 过滤器技术用于实现对 IP 层及其以上的数据包进行访问控制。访问控制的依据是若干过滤规则。IP 过滤要满足：

- 动态性：能根据安全反馈信息自动进行安全策略的调整。
- 时效性；规则能够指定作用时间，以加强授权的可控性。
- 一致性检测机制：能够检测规则之间的冲突，降低因为配置错误造成的安全漏洞；

NAT 用于将一个地址、端口对映射成另一个地址、端口对，从而为终端主机提供透明路由的方法。它可以隐藏受保护网络的内部结构，通过设置映射策略对转发包进行控制，从而提高系统的安全性。NAT 映射关系包括静态地址转换、动态地址转换、地址及端口转换、负载分配。静态地址转换完成一组内部 IP 地址到外部 IP 地址的一对一静态转换；动态地址转换则可以使得外部 IP 地址动态地映射到内部的 IP 地址上；地址及端口转换将不同的 IP 地址及传输标识符（TCP、UDP 端口号，ICMP 询问标识符等）转换为一个或多个外部地址的不同传输标识符。这样的转化既可以是外部地址向内部地址的映射，也可以是内部地址向外部地址的映射。通过实现上述一组或多组的映射关系，可以加强映射关系的可控性。

附录 A　国际上主流高端路由器性能指标

Internet 骨干路由器所面临的系统上的挑战是非常复杂的。它们不仅要能适应现有的 OC-12 及 OC-48 速率骨干网和相关的 Intra-POP 结构，而且还要能够支持具有 OC-48 速率和吉比特以太网 Intra-POP 结构的 OC-192 速率骨干网络。衡量一种网络设备是否能够满足目前网络增长的需求，关键是要看它的性能指标与技术特点。这里我们给出目前国际上两家最具实力的网络设备供应商，Cisco 公司和 Juniper 公司，高端路由器产品的技术特点和相关性能指标。引用的数据主要来自两家公司的网站，如果与实际产品和技术文档有不一致的地方，以这两家公司公布的数据为准。

A.1　Cisco 12416 Internet 路由器

随着 Internet 业务的不断扩大和发展，服务供应商要求增加带宽的数量，提高带宽质量，以增加收入，降低成本。Cisco 12000 系列 Internet 路由器具有分布式体系结构，这种结构可以提供高性能、可扩展性和数据包的优先传输，降低成本，是服务供应商消除创收壁垒所需的必要工具。我们以 Cisco 12416 Internet 路由器为例，介绍 Cisco 12000 系列 Internet 路由器的技术特点和主要性能指标。

主要特点

Cisco 12416 Internet 路由器是一种 16 插槽机柜，属于 Cisco 12000 系列，基于已实现的 Cisco 12000 系列路由器的结构。该系列的总交换容量高达 320Gbit/s，每槽容量为 20Gbit/s（10Gbit/s 全双工）。Cisco 12416 Internet 路由器的分布式分组转发和纵横矩阵交换能够支持 10Gbit/s OC-192c/STM-64c 和四端口 OC-48c/STM-16c 接口，同时继续支持现有的所有 Cisco 12000 系列线路卡。创新的虚拟输出排队（VOQ）技术可以防止线端阻塞，增强型时钟调度程序可以保证所有线路卡能够平等地接入该结构。它广泛采用高性能专用集成电路（ASIC），支持以等待时间最短的线速转发实时业务，同时该结构可以复制硬件中的组播业务，提供更高性能。其主要技术特点归纳如下：

■　分布式体系结构：线路卡可以提供高性能、有保证的数据包优先传输，从而确保系统的高性能及稳定性。分布式数据包转发是在一个简单的纵横制交换矩阵（不具有存储功能）和单个线路卡转发引擎（带有数据包缓冲器）上进行，能够确保每个插槽的线速。该线路卡可以进行组播复制，支持广播和组播视频流业务，而且不影响线路卡的业务级别，能够支持

要求将等待时间和抖动频率减至最少的业务。Cisco 12416 Internet 路由器可以确保线速性能、可靠的数据包传输和 CoS，从而可以提供大量的优质带宽，满足下一代 IP 业务对核心网络支持的需求。

■　可扩展的平台容量：Cisco 12416 Internet 路由器具有 16 个 20Gbit/s 的插槽，可提供 320Gbit/s 的集合交换容量，使服务供应商能够将其网络扩展为真正的 10Gbit/s 网络。

■　线卡的灵活性及投资保护：Cisco 12416 Internet 路由器支持业内领先的 Cisco 12000 系列线路卡，其中包括新型 Cisco OC-192c/STM-64c 和四端口 OC-48c/STM-16c 接口。所有的插槽均支持传统卡，10 个插槽中有 8 个是拓宽型的（1.75in），可以支持 10GB 光纤线路卡，其余 2 个是标准宽度的（1.25in），可以支持冗余千兆位路由处理器（GRP）。多种接口、平台间的互换以及对传统卡的支持可以确保 Cisco 12416 Internet 路由器能够提供必要的接口技术，同时保护购买以前型号线路卡的投资。

■　新一代连网性能：基于 ASIC 的分组转发和增值特性，如 COS（class-of-service）以全接口速度支持线速性能。

■　运营商级冗余和一致性：Cisco 12416 Internet 路由器完全符合网络设备构建系统（NEBS）对运营商级网络设备的要求，以符合业界的安全规则及标准。该机柜包括冗余路由处理器、交换结构卡、时钟调度程序卡和电源。由于它支持同步光纤网路自动保护交换（SONET APS）和同步数字分级体系多业务交换路径（SDH MSP），从而可以实现网络级冗余。上述特性与 Cisco 12416 Internet 路由器分布式结构相结合，可以保证一个线路卡的故障不会中断其他线路卡的业务，使客户能够确保网络的可用性。

■　低网络运行成本：Cisco 12416 Internet 路由器的设计具有降低网络运行成本的特点，包括集成式电缆管理系统、灵活的电源配置选项，冗余报警面板，前端维护路径以及 Cisco GSR 管理器（一种能够有效地管理和配置系统资源的网元管理系统）。降低运行成本也就相应地提高创收能力。

■　业内识别软件：Cisco IOS 软件系列具有强韧的特性，且已被广泛使用，能够提供一系列诊断功能和 IP 路由协议，帮助网络工程师和设计者设计复杂的协议路由环境。

如果 SP 和 ISP 需要在高性能、高扩展性有保证的数据包优先提供以及高端口密度的基础设施中部署下一代 IP 业务，则 Cisco 12416 Internet 路由器可为其提供支持 10 千兆比特的平台。Cisco 12416 是业界已证实的 Cisco 12000 系列的重要组成部分，也是 Cisco IP+Optical 新一代网络战略的有机组成部分。

产品规格

Cisco 12416 Internet 路由器产品规格如表 A-1 所示。

表 A1　　Cisco 12416 Internet 路由器产品规格

指　标	描　述
兼容性	与目前所有 Cisco 12000 系列线路卡兼容。
软件兼容性	IOS 12.0 以上
协议	IPv4、MPLS、BGP-4、IS-IS、OSFP-2、EIGRP、RIP v2、IGMP、DVMRP、PIM DM/SM

续表

指　标	描　述	
组件	每个基系统包括： ■ 4 个 DC 电源或 3 个 AC 电源； ■ 1 个千兆比特路由处理器（GRP）； ■ 1 个用于线路卡及 GPR 的 16 插槽插卡箱；系统配备 15 个线路卡和 1 个单 GPR，或者 14 个线路卡和 2 个 GRP（1:1 冗余）； ■ 3 个交换结构卡； ■ 2 个时钟调度程序卡； ■ 2 个通风组件； ■ 电缆管理架； ■ 两个报警卡； ■ 不同国家的电源线； ■ 软件和业务实现工具； ■ Cisco IOS 运行系统； ■ 用于分布式数据包转发的 Cisco 快速转发（CEF）。	
卡，端口，槽	接口（每卡端口数）	接口密度（每系统端口总数）
	1 端口千兆以太网	15 个千兆以太网端口
	8 端口快速以太网	120 个快速以太网端口
	12 端口 DS3	180 个 DS3 端口
	6 端口 E3	90 个 E3 端口
	12 端口 E3	180 个 E3 端口
	4 端口 OC-3c/STM-1 POS	60 个 OC-3STM-1 POS 端口
	4 端口 OC-3/STM-1 ATM	60 个 OC-3/STM-1 ATM 端口
	16 端口 OC-3c/STM-1 POS	240 个 OC-3c/STM-1 POS 端口
	6 端口 ChT3(168 T1)POS	90-Ch T3(1512 T1)POS
	1 端口 OC-12c/STM-1 POS	15 个 OC-12c/STM-4 POS 端口
	4 端口 OC-12c/STM-4 POS	60 个 OC-12c/STM-4 POS 端口
	1 端口 OC-12/STM-1 POS	15 个 OC-12/STM-4 ATM 端口
	4 端口 OC-12c/STM-4 POS	60 个 OC-12/STM-4 ATM 端口
	1 端口	15 ChOC-12/STM-4
	Ch-12c/STM-4(4×OC-3/STM-1)POS	(60 OC-3/STM-1)POS

续表

指　标	描　述	
	1 端口 Ch-12c/STM-4(12×DS3)POS	15Ch OC-12/STM-4 (180 DS3)POS
	1 端口 OC-12c/STM-4 DPT	15 个 OC-12c/STM-4 DPT 环路
	1 端口 OC-48c/STM-16 POS	15 个 OC-8c/STM-16 POS 端口
	1 端口 OC-48c/STM-16 DPT	15 个 OC-48c/STM-16 DPT 端口
	4 端口 OC-48c/STM-16 POS	60 个 OC-48c/STM-16 POS 端口
	1 端口 OC-192c/STM-64c POS	15 个 OC-192c/STM-64c POS 端口
连接	POS、ATM、DPT、千兆比特和快速以太网	
存储器	128MB GRP，可升级到 256MB	
选项	■ GRP＝千兆比特路由处理器	
	■ GSR10-CSC＝时钟调度程序卡	
	■ GSR10-SFC＝交换结构卡	
	■ GSR-ALRM＝报警卡	
	■ GSR10-DISP＝显示卡	
	■ OSR10-BLOWSER＝通风组件	
	■ PWR-GSR10-DC＝DC 电源	
	■ PWR-GSR10-AC＝电源	
	■ GSR10-CHASSIS＝12416 带底板机柜	
性能	交换容量为 320Gbit/s	
环境条件	■ 温度：运行状态：32～122 ° F（0～50℃）、非运行状态：-4～149 ° F（-20 ° ～65℃）。	
	■ 湿度：运行状态：10%～85%，非冷凝；非运行状态：5%～95%，非冷凝。	
	■ 高度：运行状态：0～10000ft(0～3000m)；非运行状态：0～15000ft(0～4570m)。	
	■ 散热：最大（DC：2430W@8296 Btu/h；AC：2791W@9528 Btu/h)、11602 Btu/h。	
	■ 噪音：70dBa（最大)。	
	■ 震动：运行状态（半正弦)：21 in/sec(0.53m/s)；非运行状态（梯形脉冲)：20G,52 in/s (1.32m/s)。	
	■ 振动：运行状态：0.35Grms2 3～500Hz；非运行状态：1.0 Grms 3～500Hz。	
可靠性及可用性	系统冗余	
	■ 结构卡冗余 4:1；时钟调度程序卡冗余 1:1；	
	■ 电源冗余（DC 1:1，AC 负载平衡)；	
	■ 通风组件冗余；	
	■ 路由处理器冗余 1:1；	
	■ 报警卡冗余 1:1；	
	■ 通过线路卡实现双归；	

续表

指　标	描　述
	■ 支持 APS。
	平均故障间隔时间（MTBF）
	■ 时钟调度程序卡＝240078hr ；
	■ 交换结构卡=276062hr。
	网络管理命令行界面（CLI）
	■ Cisco GSR 管理器；
	■ 简单网络管理协议（SNMP）。
管理界面	GRP 支持 2 个串行端口（控制台和 AUX 连接）以及一个 10/100 端口。
审核要求	SR-3580 规定的 NEBS Criteria 3 级要求；
标准	安全性认证：
	■ CSA-22.2 No.950-UL 1950；
	■ EN60950/IEC60950；
	■ EN60825/IEC60825；
	■ ACA TS001；
	■ AS/NZS 3260。
	电磁标准（放射物）：
	■ FCC A 级；
	■ ICES-003 A 级；
	■ EN55022 B 级（到 1GHz）；
	■ VCCI B 级（到 1GHz）；
	■ AS/NZS 3548 B 级。
	抗扰性：
	■ EN300386（网络设备 EMC）；
	■ EN61000-3-2/IEC-1000-3-2（电源线谐波）；
	■ EN61000-4-3/IEC-1000-4-2（静电释放物[ESD]）；
	■ EN61000-4-3/IEC-1000-4-3（抗辐射）；
	■ EN61000-4-4/IEC-1000-4-4（EFT）；
	■ EN61000-4-5/IEC-1000-4-5（电涌）；
	■ EN61000-4-6/IEC-1000-4-6（低频率导电干扰）；
	■ EN61000-4-11/IEC-1000-4-11（电压弛度和垂度）EISI；
	■ ETS-300386；
	■ FCC class A compliance Notice(United States)；
	■ ICES-003 Class A Compliance Notice(Canada)；
	■ VCC1 Class B Compliance Notice(Japan)。

其中：

- G 是加速度值，1G 等于 32.17ft/sec(9.81msec)。
- Grms 是加速度的平方根值。
- 全冗余电源配置，电源输入模块 A1 和 B1 为系统负载区 1 提供冗余（上部浏览器模块和上部卡机箱），模块 A2 和 B2 为系统负载区 2 提供冗余（交换设备卡机箱、低部卡机箱和低部浏览器模块）。

A.2　Juniper 公司 M160 骨干路由器主要性能指标

Juniper 网络公司提供的 M 系列 Internet 骨干路由器是目前世界上最完备的路由系统之一，M 系列将 Internet 可伸缩性、Internet 控制及无可比拟的转发性能集于一身。M160 骨干路由器专为大型骨干核心网络设计，同时具有在将来转移至网络边缘所需的功能。对于那些既需要具有丰富功能的基础结构、同时又要具有可靠性能的大型网络，M160 是一个理想的平台。M160 能够提供每机箱 8 个 OC-192c/STM –64（每机架 16 个）或每机箱 32 个 OC-48c/STM -16（每机架 64 个）物理接口卡（PIC）的、真正提供线速性能的路由平台。M160 具有突破性的 ASIC 技术使光纤带宽能够真正应用于新的、不同的 IP 业务。

主要特点

M160 路由器具有超过 160Mpps 的路由查询速率，提供线速转发性能和高于 160Gbit/s 的吞吐能力。M160 真正能够提供 OC-192c/STM -64PIC 连接的、具有全双工 10Gbit/s 吞吐能力的路由器，其每机箱 32 块或每机架 64 块 OC-48c/STM-16 物理接口卡（PIC）的端口密度也是业界领先的。M160 绝无仅有的 ASIC 设计及已在实际网络中运行并被证实的路由软件将使其处于下一代 IP 技术的前沿。其主要技术特点归纳如下：

- 体系结构：高度集成的 ASIC 的转发提供 160Mpps 的查询速率；吞吐能力超过 160Gbit/s；对大型、复杂的转发表具有良好的可扩展性；对昂贵的线路进行充分利用；数据包的大小并不影响转发性能；坚固的系统稳定性；较少的部件以保证高可靠性。路由与转发性能完全分离，路由的抖动和网络的不稳定不会影响数据包的转发。快速收敛为对时延敏感的业务提供稳定、可靠的性能。单级缓存消除了头阻塞，有效使用可用接口带宽，实现对组播业务的最佳支持；通过只需对共享内存进行一次写读操作降低时延。
- 冗余设计：冗余交换及转发模块（SFM）、冗余路由引擎及综合控制系统（MCS）增加了系统可用性，确保一块 SFM 发生故障时可自动切换至冗余的 SFM，从而缩短修复间隔时间（MTTR）。
- 接口：业界领先的端口密度使 POP 机架空间利用率非常高效，不会因空间而限制将来的成长。规划细致的、可互换的接口可灵活地应用于多种环境之中，包括边缘的专线接入及类似于城域网 POP 的小型核心网。
- 环境：机箱最大功率为 2600 瓦，并具有冗余的、可进行负载的双直流电源，有效使用 POP 电力，经济地进行实施及运营。

M160 的主要技术指标

M160 路由器通过其先进的 Internet Processor II ASIC 及 OC-192c/STM -64PIC，提供了业界领先的性能，并具有数据包过滤、速率限制及采样等先进功能。M160 的主要技术指标如表 A-2 所示。

表 A-2　　JuniperM160 的主要技术指标

指　标	描　述
SFM	■ 一块用于提供 160Mpps 汇集分组查询速率的 Internet 处理器 II ASIC（每块 SFM 提供 40Mpps）； ■ 吞吐能力为 204.8Gbit/s（全双工 102.4Gbit/s）； ■ 两块用于协调存储及单级缓存的分布缓存管理器 ASIC； ■ 8MB 具有奇偶保护的 SSRAM； ■ 处理器子系统（一块 PowerPC603e 处理器，256KB 奇偶保护 2 级缓存，及 64MB 奇偶保护 DRAM）。
路由引擎	■ 具有内建 256KB 2 级缓存的 333MHz Pentium II； ■ SDRAM（3 排 168 针 DIMM，具有 768MB ECC SDRAM）； ■ 作为主引导设备的 80MB 集成闪存驱动器； ■ 作为第二引导设备的 6.4GB IDF 硬盘驱动器； ■ 作为第三引导设备的 110MB 闪存 PC 卡； ■ 用于带外管理的 10/100 Base-T 自动感应 RJ-45 以太网端口； ■ 用于控制及远程管理的两个 RS-232（DB9 接头）异步串行端口。
FPC	■ 12.8Gbit/s 吞吐容量； ■ 两块用于在 I/O 过滤器 ASIC 中分配及均衡分组的分组导向器 ASIC； ■ 4 块用于对分组进行线速解析、划分优先级，及排队的 I/O 管理器 ASIC。
MCS	■ 用于 SONET/SDH PIC 的 19.44MHz 3 层参考时钟； ■ 用于监测路由器组件状态的控制器。
直流电源需求	电源最大输出功率：2600 瓦； 输入电压： -48～-60V 输入电流：65A@-48VDC。
环境	温度：32°F 到 104°F/0°C到 40°C 最大海拔高度：10000ft/3048m 无性能影响； 相对湿度：5%～90%，非凝结； 地震：设计满足 Bellcore4 级抗震要求； 散热：9，400 BTU/小时。

续表

指　标	描　述
标准	安全 ■ CSA 22.2 NO.950； ■ *UL 1950； ■ EN 60950，信息技术设备安全性； ■ EN 60825-1，激光产品安全性-第一部分：设备分类； ■ 需求及用户指南； ■ EN 60825-2,激光产品安全性-第二部分：光纤通信系统安全性。 EMC ■ AS 3548 Class A（澳大利亚）； ■ BSMI Class A（台湾）； ■ EN 55022 Class A 辐射（欧洲）； ■ FCC Class A（美国）； ■ VCCI Class A（日本）。 抗干扰 ■ EN 61000-3-2 电源线谐波； ■ EN 61000-4-2 ESD； ■ EN 61000-4-3 辐射抗干扰性； ■ EN 61000-4-4 EFT； ■ EN 61000-4-5 浪涌 ； ■ EN 61000-4-6 低频公用抗干扰性； ■ EN 61000-4-11 电压瞬跌。 NEBS 设计以满足下述标准要求： ■ GR-63-CORE:ENBS，物理保护； ■ GR-1089-CORE：网络 EMC 和电气安全性通信设备； ■ SR-3580 NEBS 标准级别（符合第 3 级标准）。 ETSI ■ ETS-300386-2 交换设备。

附录B　参考的RFC标准列表

RFC由一系列草案组成，起始于1969年，内容和Internet相关。草案讨论了计算机网络通信的各个方面，重点在网络协议、过程、程序，以及一些会议注解、意见和风格方面的概念。如果要了解RFC系列的历史方面的情况，可以参考“30 Years of RFCs”。Internet协议族的文档部分（由Internet工程委员会IETF以及IETF下属的管理组IESG定义），也作为RFC文档出版。因此，RFC在Internet相关标准中有着重要的地位。

RFC编辑者的职责是由Internet中的大家提议形成的，所出版的语言也就和Internet一样。IETF和ISOC是代表了世界各地的国际性组织，英语是IETF的第一工作语言，也是IETF的正式出版语言。RFC 2026“The Internet Standards Process——Revision 3”允许RFC翻译成其他不同的语言。

路由器技术主要涉及到OSI参考模型的数据链路层、IP层和TCP层，这里我们给出所参考的主要RFC文档规范列表。列表按各种技术分类，包括ARP协议、PPP协议、IP协议、TCP协议、RIP协议、OSPF协议、BGP-4协议以及安全结构。

表B-1　　ARP和PPP协议相关的RFC

序号	RFC编号	RFC名称
1	RFC 826	An Ethernet Address Resolution Protocol
2	RFC 1662	PPP in HDLC-like Framing
3	RFC 1764	The PPP XNS IDP Control Protocol (XNSCP)
4	RFC 1763	The PPP Banyan Vines Control Protocol (BVCP)
5	RFC 1762	The PPP DECnet Phase IV Control Protocol (DNCP)
6	RFC 1968	The PPP Encryption Control Protocol (ECP)
7	RFC 1962	The PPP Compression Control Protocol (CCP)
8	RFC 1973	PPP in Frame Relay
9	RFC 1975	PPP Magnalink Variable Resource Compression
10	RFC 1977	PPP BSD Compression Protocol
11	RFC 1979	PPP Deflate Protocol
12	RFC 1967	PPP LZS-DCP Compression Protocol (LZS-DCP)

续表

序号	RFC 编号	RFC 名称
13	RFC 1974	PPP Stac LZS Compression Protocol
14	RFC 1976	PPP for Data Compression in Data Circuit-Terminating Equipment (DCE)
15	RFC 1963	PPP Serial Data Transport Protocol (SDTP)
16	RFC 1990	The PPP Multilink Protocol (MP)
17	RFC 1989	PPP Link Quality Monitoring
18	RFC 1978	PPP Predictor Compression Protocol
19	RFC 1993	PPP Gandalf FZA Compression Protocol
20	RFC 1994	PPP Challenge Handshake Authentication Protocol (CHAP)
21	RFC 2043	The PPP SNA Control Protocol (SNACP)
22	RFC 2097	The PPP NetBIOS Frames Control Protocol (NBFCP)
23	RFC 2118	Microsoft Point-To-Point Compression (MPPC) Protocol
24	RFC 2125	The PPP Bandwidth Allocation Protocol (BAP) The PPP Bandwidth Allocation Control Protocol (BACP)
25	RFC 2153	PPP Vendor Extensions
26	RFC 2284	PPP Extensible Authentication Protocol (EAP)
27	RFC 2363	PPP Over FUNI
28	RFC 2364	PPP over AAL5
29	RFC 2419	The PPP DES Encryption Protocol, Version 2 (DESE-bis)
30	RFC 2420	The PPP Triple-DES Encryption Protocol (3DESE)
31	RFC 2433	Microsoft PPP CHAP Extensions
32	RFC 2484	PPP LCP Internationalization Configuration Option
33	RFC 2509	IP Header Compression over PPP
34	RFC 2615	PPP over SONET/SDH
35	RFC 2661	Layer Two Tunneling Protocol 'L2TP'
36	RFC 2701	Multi-link Multi-node PPP
37	RFC 2716	PPP EAP TLS Authentication Protocol
38	RFC 2759	Microsoft PPP CHAP Extensions, Version 2
39	RFC 2823	PPP over Simple Data Link (SDL) using SONET/SDH with ATM-like framing
40	RFC 2878	PPP Bridging Control Protocol (BCP)
41	RFC 3078	Microsoft Point-To-Point Encryption (MPPE) Protocol
42	RFC 3153	PPP Multiplexed

表 B-2　　IP 协议以及 IPSec 相关的 RFC

序　号	RFC 编号	RFC 名称
1	RFC 1829	The ESP DES-CBC Transform
2	RFC 2104	HMAC: Keyed-Hashing for Message Authentication
3	RFC 2085	HMAC-MD5 IP Authentication with Replay Prevention
4	RFC 2401	Security Architecture for the Internet Protocol
5	RFC 2410	The NULL Encryption Algorithm and Its Use With IPsec
6	RFC 2411	IP Security Document Roadmap
7	RFC 2402	IP Authentication Header
8	RFC 2412	The OAKLEY Key Determination Protocol
9	RFC 2451	The ESP CBC-Mode Cipher Algorithms
10	RFC 2403	The Use of HMAC-MD5-96 within ESP and AH
11	RFC 2404	The Use of HMAC-SHA-1-96 within ESP and AH
12	RFC 2405	The ESP DES-CBC Cipher Algorithm With Explicit IV
13	RFC 2406	IP Encapsulating Security Payload (ESP)
14	RFC 2407	The Internet IP Security Domain of Interpretation for ISAKMP
15	RFC 2408	Internet Security Association and Key Management Protocol
16	RFC 2409	The Internet Key Exchange (IKE)
17	RFC 2857	The Use of HMAC-RIPEMD-160-96 within ESP and AH

表 B-3　　TCP 协议相关的 RFC

序　号	RFC 编号	RFC 名称
1	RFC 2861	TCP Congestion Window Validation
2	RFC 2883	An Extension to the Selective Acknowledgement (SACK) Option for TCP
3	RFC 2988	Computing TCP's Retransmission Timer
4	RFC 3042	Enhancing TCP's Loss Recovery Using Limited Transmit
5	RFC 3168	The Addition of Explicit Congestion Notification (ECN) to IP

表 B-4　　RIP 协议相关的 RFC

序号	RFC 编号	RFC 名称
1	RFC 1923	RIPv1 Applicability Statement for Historic Status
2	RFC 2081	RIPng Protocol Applicability Statement
3	RFC 2080	RIPng for IPv6
4	RFC 2091	Triggered Extensions to RIP to Support Demand Circuits
5	RFC 2092	Protocol Analysis for Triggered RIP

表 B-5　　OSPF 协议相关的 RFC

序号	RFC 编号	RFC 名称
1	RFC 1246	Experience with the OSPF Protocol
2	RFC 1245	OSPF Protocol Analysis
3	RFC 1587	The OSPF NSSA Option
4	RFC 1586	Guidelines for Running OSPF Over Frame Relay Networks
5	RFC 1765	OSPF Database Overflow
6	RFC 1793	Extending OSPF to Support Demand Circuits
7	RFC 1850	OSPF Version 2 Management Information Base
8	RFC 2096	IP Forwarding Table MIB
9	RFC 2328	OSPF Version 2
10	RFC 2329	OSPF Standardization Report
11	RFC 2370	The OSPF Opaque LSA Option
12	RFC 2740	OSPF for IPv6
13	RFC 2844	OSPF over ATM and Proxy PAR
14	RFC 3137	OSPF Stub Router Advertisement

表 B-6　　BGP 协议相关的 RFC

序号	RFC 编号	RFC 名称
1	RFC 1745	BGP4/IDRP for IP---OSPF Interaction
2	RFC 1771	A Border Gateway Protocol 4 (BGP-4)
3	RFC 1774	BGP-4 Protocol Analysis

续表

序号	RFC 编号	RFC 名称
4	RFC 1773	Experience with the BGP-4 protocol
5	RFC 1863	A BGP/IDRP Route Server alternative to a full mesh routing
6	RFC 1966	BGP Route Reflection An alternative to full mesh IBGP
7	RFC 1998	An Application of the BGP Community Attribute in Multi-home Routing
8	RFC 1997	BGP Communities Attribute
9	RFC 2270	Using a Dedicated AS for Sites Homed to a Single Provider
10	RFC 2385	Protection of BGP Sessions via the TCP MD5 Signature Option
11	RFC 2439	BGP Route Flap Damping
12	RFC 2519	A Framework for Inter-Domain Route Aggregation
13	RFC 2545	Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing
14	RFC 2796	BGP Route Reflection An alternative to full mesh IBGP
15	RFC 2842	Capabilities Advertisement with BGP-4
16	RFC 2858	Multiprotocol Extensions for BGP-4
17	RFC 2918	Route Refresh Capability for BGP-4
18	RFC 3065	Autonomous System Confederations for BGP

参 考 文 献

1．李津生，洪佩琳．下一代 Internet 的网络技术．北京：人民邮电出版社，2001

2．Vijay Bollapragada，Curtis Murphy，Russ White. Cisco IOS 精髓，邓劲生、白建军、齐宁 译．中国电力出版社，2001

3．（美）George C.Sackett . Cisco 路由器手册．前导工作室 译．机械工业出版社，2000

4．（美）Stephen Hutnik ,Michael Staterlee . CCIE 实验室操作指南．卢泽新，杨岳湘，周榕 等译．机械工业出版社，2000

5．（美）Jim Metzler, Lynn DeNoia. 第三层交换，卢泽新，周榕 等译．机械工业出版社，2000

6．（美）William R.Parkhurst. Cisco，潇湘工作室 译．路由器 OSPF 设计与实现．机械工业出版社，1999

7．（美）Thmoas M.Thomas II 等，长江网络工作室 译．组建可扩展的 Cisco 网络．机械工业出版社，1999

8．Douglas E.Comer, Douglas E.Comer, David L.Stevens，张娟，王海 译．用 TCP/IP 进行网络互连（第一卷：原理、协议与结构）．电子工业出版社，1998

9．Douglas E.Comer, Douglas E.Comer, David L.Stevens，张娟，王海 译．用 TCP/IP 进行网络互连（第二卷：设计、实现和内部构成）．电子工业出版社，1998

10．吴江，赵慧玲．下一代骨干网路由平台 MPLS．人民邮电出版社，2001

11．Gilbert Held．Ethernet Networks (3rd Edition)．人民邮电出版社, 1998. 4

12．R. Dube. A Comparison of Scaling Techniques for BGP. ACM Computer Communication Review, 29 (3), 1999

13．J. W. Stewart III . BGP4: Inter-Domain Routing in the Internet. Addision-Wesley. 1998

14．Stephen Kent，Charles Lynn and Karen Seo. Secure Border Gateway Protocol (S-BGP). IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATION, VOL. 18, NO.4, APRIL 2000

15．B Halabi. Internet Routing Architectures. Cisco-Press. 1997

16．C Partridge，P Carvey et al. A 50-Gb/s IP Router. IEEE/ACM Transactions on Networking, vol 6, Issue 3, Pg237-248, 1998

17．N McKeown et al. The Tiny Tera: A small high-bandwidth packet switch core. IEEE Micro, Jan-Feb 1997

18．P Newman. IP switching and gigabit routers. IEEE Communications, vol. 30, pp. 64-69, Feb. 1997

19．Keshav, S et al. Issues and Trends in Router Design. IEEE Communications Magazine, May 1998 pp 144 – 151

20．P Gupta, S Lin, N McKeown. Routing lookups in hardware at memory access speed. IEEE

INFOCOM (1998) 1240-1247

21. D S Alexander, M Shaw, S M Nettles and J Smith Active Bridging Proceedings of ACM SIGCOMM'97,Cannes, September 1997

22. S M Ballew. Managing IP Networks with Cisco Routers. O'Reilly 1997

23. A Brodnik, S Carlsson, M Degermark, S Pink. Small Forwarding Tables for Fast Route Lookups. Proceedingsof ACM SIGCOMM'97. Cannes, September 1997

24. JCR. Bennett and H Zhang. Hierarchical Packet Fair Queueing Algorithms. Proceedings of ACM SIGCOMM '96,Stanford, August 1996

25. JCR Bennett, D C Stephens and H Zhang. High Speed, Scalable, and Accurate Implementation of Fair Queueing Algorithms in ATM Networks. ICNP '97, 1997

26. A Demers, S Keshav and S Shenker. Design and Analysis of a Fair Queueing Algorithm. Proceedings of ACM SIGCOMM '89, Austin, September 1989

27. S Floyd and V Jacobson. Random Early Detection Gateways for Congestion Avoidance. IEEE/ACM Transactions on Networking, August 1993

28. P Goyal. Packet Scheduling Algorithms for Integrated Service Networks. PhD Thesis, UC Texas Austin, 1997

29. P Gupta, S Lin and N McKeown. Routing Lookups in Hardware at Memory Access Speeds. Proceedings of IEEE INFOCOM 98, March 1998

30. P Goyal, H Vin and H Chen. Start-time Fair Queueing: A Scheduling Algorithm for Integrated Services Packet Switching Networks. Proceedings of ACM SIGCOMM '96, August 1996

31. F Baker, ed. Requirements for IP Version 4 Routers. RFC 1812, June 1995

32. S Deering and R Hinden. Internet Protocol, Version 6 (Ipv6) Specification. Request for Comments (Proposed Standard) RFC 1883. Internet Engineering Task Force, January 1996

33. M Degermark, A Brodnik, S Carlsson and S Pink. Small Forwarding Tables for Fast Routing Lookups. ACM SIGCOMM'97, Palais des Festivals, Cannes, France, pp 3-14

34. W Doeringer, G Karjoth and M Nassehi. Routing on Longest- Matching Prefixes. IEEE/ACM Transactions on Networking, Vol 4, No 1, February 1996, pp 86-97

35. P Gupta, S Lin and N McKeown. Routing Lookups in Hardware at Memory Access Speeds. IEEE INFOCOM'98, San Francisco, April 1998, Session 10B-1

36. B Lampson, V Srinivasan and G Varghese. IP Lookups using Multiway and Multicolumn Search. IEEE INFOCOM'98, San Francisco, April 1998, Session 10B-2

37. A McAuley, P Francis. Fast Routing Lookup Using CAMs. Proc, IEEE INFOCOM 1993

38. T Pei and C Zukowaki. Putting Routing Tables in Sillicon. IEEE Network Magazine, January 1992

39. Y Rekhter, T Li. An Architecture for IP Address Allocation with CIDR. RFC 1518, September 1993

40. Tony Li. MPLS and the Evolving Internet Architecture. in IEEE Communications Magazine,Dec 1999 pp 38-41

41．Yakov Rekhter etc. Tag Switch Architecture Overview. in Proc of the IEEE,Dec 1997,pp.1973-1983
42．Arun Viswanathan etc .Evolution of Multiprotocol Label Switching. in IEEE Communications Magazine,May 1998 ,pp.165-173
43．Daniel o.Awduche .MPLS and Traffic Engineering in IP Networks. in IEEE Communications Magazine,Dec 1999
44．李珂，顾尚杰，诸鸿文．多协议标记交换中的关键技术．世界网络与多媒体，第四期。
45．胡琳．流量工程和 MPLS 技术．互联网技术。
46．徐荣，龚倩．多协议标签交换技术．通讯世界，2000.4。
47．赵慧玲，吴江．MPLS Qos 研究．世界电信，2000.5
48．Atkinson, R. Security Architecture for the Internet Protocol. RFC 1825, August 1995
49．Atkinson, R. IP Encapsulating Security Payload. RFC 1827, August 1995
50．Braden, R Clark, D Crocker, S and C Huitema. Report of IAB Workshop on Security in the Internet Architecture. RFC 1636, June 1994.
51．Atkinson, R. IP Authentication Header. RFC 1826, August 1995.
52．Steve Deering & Bob Hinden. Internet Protocol version 6(IPv6) Specification. RFC 1883, December 1995.
53．Steve Kent, Randall Atkinson. Security Architecture for the Internet Protocol. Internet Draft, May 1998.
54．Steve Kent, Randall Atkinson. IP Encapsulating Security Payload (ESP). Internet Draft, May 1998
55．徐恪等．宽带 IP 路由器的体系结构分析．软件学报，2000 11

参考网络资源

1．www.cisco.com 以及 www.cisco.com.cn，Cisco 网站的技术支持中心
2．www.juniper.com
3．www.pluris.com
4．www.agilent.com
5．www.nokia.com.cn
6．www.autosoft.com.cn
7．www.zte.com.cn
8．www.lucent.com
9．www.ietf.org 主要是一些 RFC 和草案