**路由器部分**

[1. 路由器概述 2](#_Toc464303723)

[1.1 路由器的功能 2](#_Toc464303724)

[1.2 路由器的任务 2](#_Toc464303725)

[2. 路由器的内部结构 3](#_Toc464303726)

[3. 路由器的工作原理 4](#_Toc464303727)

[3.1 路由协议的基本概念 5](#_Toc464303728)

[3.2 路由结构 8](#_Toc464303729)

[3.3 RIP 协议设计 11](#_Toc464303730)

[3.4 OSPF协议设计 23](#_Toc464303731)

[3.5 BGP协议设计 30](#_Toc464303732)

[4. 路由器配置 41](#_Toc464303733)

[4.1 路由器基础 41](#_Toc464303734)

[4.2 路由器配置 44](#_Toc464303735)

[4.3 cisco ios使用方法 52](#_Toc464303736)

[4.4 路由器常用配置 65](#_Toc464303737)

[4.5 路由器配置实验 72](#_Toc464303738)

1. **路由器概述**

### 1.1 路由器的功能

路由器是在网络层实现互联的设备。路由器实现网络层上数据包的存储转发，它具有路径选择功能，可依据网络当前的拓扑结构，选择“最佳”路径，把接收的数据包转发出去，从而实现网络负载平衡，减少网络拥塞路由器工作在网络层，用于连接不同的局域网和广域网，故称为“LAN网间互联设备”。一个路由器可以连接两个局域网、一个局域网和一个广域网，或两个广域网。

路由器的具体功能如下：

* 路由功能（寻径功能）——寻找并记录到达目的网段的最佳路径，体现在路由器上则包括路由表的建立、维护和查找
* 交换功能——路由器的交换功能与以太网交换机执行的交换功能不同，路由器的交换功能是指在网络之间转发分组数据的过程，涉及到从接收接口收到数据帧，解封装，对数据包做相应处理，根据目的网络查找路由表，决定转发接口，做新的数据链路层封装等过程
* 隔离广播、指定访问规则——路由器阻止广播的通过，并且可以设置访问控制列表(ACL)对流量进行控制
* 异种网络互连——支持不同的数据链路层协议，可以连接异种网络
* 子网间的速率匹配——路由器有多个接口，不同接口具有不同的速率，路由器需要利用缓存及流控协议进行速率适配

### 1.2 路由器的任务

路由器的主要任务是把通信引导到目的地网络，然后到达特定的节点站地址。后一个功能是通过网络地址分解完成的。例如，把网络地址部分的分配指定成网络、子网和区域的一组节点，其余的用来指明子网中的特别站。分层寻址允许路由器对有很多个节站的网络存储寻址信息。在广域网范围内的路由器按其转发报文的性能可以分为两种类型，即中间节点路由器和边界路由器。尽管在不断改进的各种路由协议中，对这两类路由器所使用的名称可能有很大的差别，但所发挥的作用却是一样的。中间节点路由器在网络中传输时，提供报文的存储和转发。同时根据当前的路由表所保持的路由信息情况，选择最好的路径传送报文。由多个互连的LAN组成的公司或企业网络一侧和外界广域网相连接的路由器，就是这个企业网络的连界路由器。它从外部广域网收集向本企业网络寻址的信息，转发到企业网络中有关的网络段；另一方面集中企业网络中各个LAN段向外部广域网发送的报文，对相关的报文确定最好的传输路径。

1. 路由器的内部结构

路由器结构从功能上可以分成两个部分：分组转发部分和路由选择部分。分组转发主要由三个部分组成：输入端口，输出端口，交换结构。路由选择部分也可以称作控制部分，其核心是路由选择处理机。

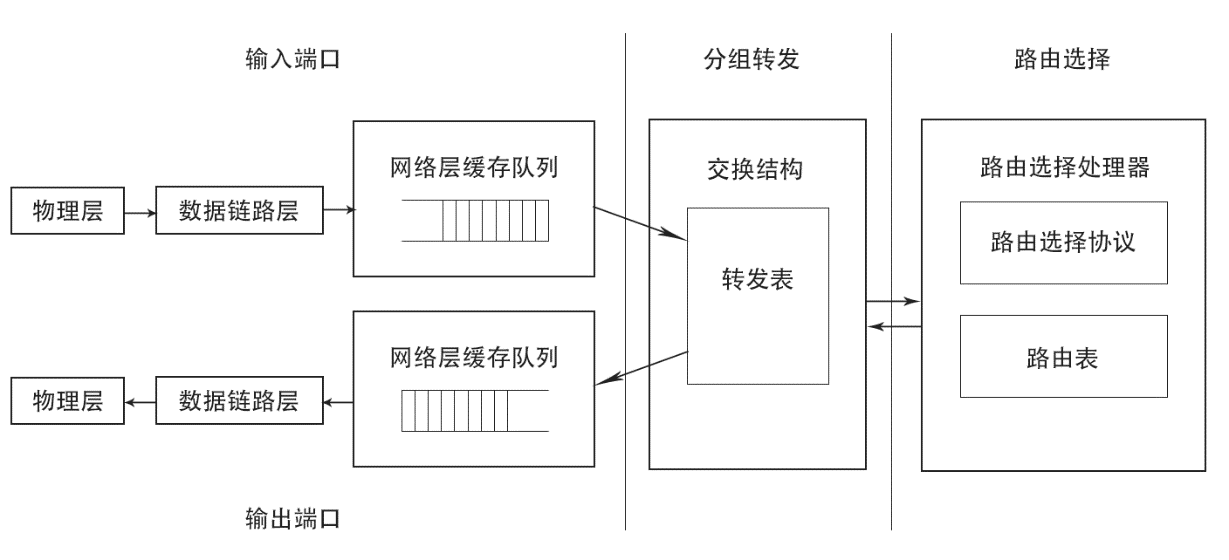


图1.1 路由器的内部结构

* 输入端口：输入端口是物理链路和输入包的进口处。端口通常由线卡提供，一块线卡一般支持4、8或16个端口，一个输入端口具有许多功能。第一个功能是进行数据链路层的封装和解封装。第二个功能是在转发表中查找输入包目的地址从而决定目的端口（称为路由查找），路由查找可以使用一般的硬件来实现，或者通过在每块线卡上嵌入一个微处理器来完成。第三，为了提供QoS（服务质量），端口要对收到的包分成几个预定义的服务级别。第四，端口可能需要运行诸如SLIP（串行线网际协议）和PPP（点对点协议）这样的数据链路级协议或者诸如PPTP（点对点隧道协议）这样的网络级协议。一旦路由查找完成，必须用交换开关将包送到其输出端口。如果路由器是输入端加队列的，则有几个输入端共享同一个交换开关。这样输入端口的最后一项功能是参加对公共资源（如交换开关）的仲裁协议。
* 交换开关：交换开关可以使用多种不同的技术来实现。迄今为止使用最多的交换开关技术是总线、交叉开关和共享存贮器。最简单的开关使用一条总线来连接所有输入和输出端口，总线开关的缺点是其交换容量受限于总线的容量以及为共享总线仲裁所带来的额外开销。交叉开关通过开关提供多条数据通路，具有N×N个交叉点的交叉开关可以被认为具有2N条总线。如果一个交叉是闭合，输入总线上的数据在输出总线上可用，否则不可用。交叉点的闭合与打开由调度器来控制，因此，调度器限制了交换开关的速度。在共享存贮器路由器中，进来的包被存贮在共享存贮器中，所交换的仅是包的指针，这提高了交换容量，但是，开关的速度受限于存贮器的存取速度。尽管存贮器容量每18个月能够翻一番，但存贮器的存取时间每年仅降低5%，这是共享存贮器交换开关的一个固有限制。
* 输出端口：输出端口在包被发送到输出链路之前对包存贮，可以实现复杂的调度算法以支持优先级等要求。与输入端口一样，输出端口同样要能支持数据链路层的封装和解封装，以及许多较高级协议。
* 路由处理器：路由处理器计算转发表实现路由协议，并运行对路由器进行配置和管理的软件。同时，它还处理那些目的地址不在线卡转发表中的包。

## 路由器的工作原理

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的下一跳送往何处，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，并在新的网络加入和删除一个网络时，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地情况。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。

### 3.1 路由协议的基本概念

#### 3.1.1 网络发现和路由表维护

动态路由协议两个重要进程是最初的发现远程网络和在路由表中维护这些网络的列表。

**1. 动态路由协议的目的**

路由协议由一组处理进程、算法和消息组成，用于交换路由信息，并将其选的最佳路径添加到路由表中。路由协议的用途如下：

* 发现远程网络
* 维护最新路由信息
* 选择通往目的网络的最佳路径
* 当前路径无法使用时找出新的最佳路径

数据结构 － 某些路由协议使用路由表和/或数据库来完成路由过程。此类信息保存在内存中。  
算法 － 算法是指用于完成某个任务的一定数量的步骤。路由协议使用算法来维护路由信息并确定最佳路径。  
路由协议消息 － 路由协议使用各种消息找出邻近的路由器，交换路由信息，并通过其它一些任务来获取和维护准确的网络信息。

**2. 动态路由协议的运行**

所有路由协议都有着相同的用途 － 获取远程网络的信息，并在网络拓扑结构发生变化时快速作出调整。所用的方式由该协议所使用的算法及其运行特点决定。动态路由协议的运行过程由路由协议类型及协议本身所决定。一般来说，动态路由协议的运行过程如下： 1. 路由器通过其接口发送和接收路由消息。 2. 路由器与使用同一路由协议的其它路由器共享路由消息和路由信息。 3. 路由器通过交换路由信息来了解远程网络。 4. 如果路由器检测到网络拓扑结构的变化，路由协议可以将这一变化告知其它路由器。

#### 3.1.2 动态路由协议的收敛

路由协议的重要特征之一，就是当拓扑发生变化时如何能快速收敛。  
收敛是指所有网络路由器的路由表达到一致的过程。当所有路由器都获取到完整而准确的网络信息时，网络即完成收敛。收敛时间是指路由器共享网络信息、计算最佳路径并更新路由表所花费的时间。网络在完成收敛后才可以正常运行，因此，大部分网络都需要在很短的时间内完成收敛。  
收敛过程既具有协作性，又具独立性。路由器之间既需要共享路由信息，各个路由器也必须独立计算拓扑结构变化对各自路由过程所产生的影响。由于路由器独立更新网络信息以与拓扑结构保持一致，所以，也可以说路由器通过收敛来达成一致。  
收敛的有关属性包括路由信息的传播速度以及最佳路径的计算方法。可以根据收敛速度来评估路由协议。收敛速度越快，路由协议的性能就越好。通常，RIP和IGRP收敛较慢，而EIGRP、OSPF和IS-IS收敛较快。

#### 3.1.3 管理距离

在路由进程决定使用哪条路由来转发数据包之前，它必须先决定哪条路由放到路由表中。路由器经常会学到多于一条的对于目标网络的路由来源。路由进程将要决定使用哪个路由来源。管理距离就用作此目的。

**1. 多个路由来源**

路由器通过静态路由和动态路由协议来了解与其直接相连的领近网络以及远程网络的信息。实际上，路由器可能会从多个来源获悉通往同一网络的路由。例如，为某一网络/子网掩码配置静态路由后，动态路由协议（如RIP）又动态了解到该网络/子网掩码。路由器需要选择在路由表中添加哪条路由。  
在同一个网络中可以部署多个动态路由协议，不过这种情况很少见。在某些情况下，有必要使用多个路由协议（如RIP和OSPF）来路由同一网络地址。由于不同路由协议使用不同的度量——RIP使用跳数，而OSPF使用带宽——因此，不能通过比较度量值来确定最佳路径。  
那么，当路由器从多个路由来源获取到同一网络的路由信息时，将如何确定在路由表中添加哪条路由？路由器根据路由来源的管理距离作出判断。

**2. 管理距离的用途**

管理距离（AD）定义路由器来源的优先级别。对于每个路由器（包括特定路由协议、静态路由或是直接相连的网络），使用管理距离值从高到低的优先顺序来排定优先级。如果从多个不同的路由来源获取到同一目的的网络路由信息，路由器将会使用AD功能来选择最佳路径。  
管理距离是从0到255的整数值。值越低表示路由来源的优先级别越高。管理距离值为0表示优先级别最高。只有直接相连的网络的管理距离为0，而且这个值不能更改。值为255表示路由器不信任该路由来源，并且不会将其添加到路由表中。

#### 3.1.4 动态路由协议的分类

根据是否在一个自治域内部使用，动态路由协议分为内部网关协议（IGP）和外部网关协议（EGP）。这里的自治域指一个具有统一管理机构、统一路由策略的网络。自治域内部采用的路由选择协议称为内部网关协议，常用的有RIP、OSPF；外部网关协议主要用于多个自治域之间的路由选择，常用的是BGP和BGP-4。

**1. 距离矢量和链路状态路由协议**

内部网关协议（IGP）可以划分为两类：

* 距离矢量路由协议
* 链路状态路由协议

距离矢量是指以距离和方向构成的矢量来通告路由信息。距离按跳数等度量来定义，方向则是下一跳的路由器或送出接口。距离矢量协议通常使用贝尔曼-福特算法来确定最佳路径。

与距离矢量路由协议的运行过程不同，配置链路状态路由协议的路由器可以获取所有其他路由器的信息来创建网络的“完整视图”，即拓扑结构。我们继续拿路标来作类比，使用链路状态路由协议就好比拥有一张完整的网络拓扑图。从源到目的网络的路途中并不需要路标，因为所有链路状态路由器都使用相同的“网络地图”。链路状态路由器使用链路状态信息来创建拓扑图，并在拓扑结构中选择到达所有目的网络的最佳路径。

**2. 有类和无类路由协议**

所有的路由协议可以被分为以下两种之一：

* 有类路由协议
* 无类路由协议

有类路由协议在路由信息更新过程中不发送子网掩码信息。最早出现的路由协议（如RIP）都属于有类路由协议。那时，网络地址是按类来分配的：A类、B类或C类。路由协议的路由信息更新中不需要包括子网掩码，因为子网掩码可以根据网络地址的第一组二进制八位数来确定。

有类路由协议包括RIPv1和IGRP。 在无类路由协议的路由更新中，同时应包括网络地址和子网掩码。如今网络已不再按照类来分配地址，子网掩码也就无法根据网络地址的第一组二进制八位数来确定。如今的大部分网络都需要使用无类路由协议，因为无类路由协议支持VLSM，非连续网络以及其他一些功能。  
无类路由协议包括RIPv2、EIGRP、OSPF、IS-IS和BGP等。

### 3.2 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3.1所示。

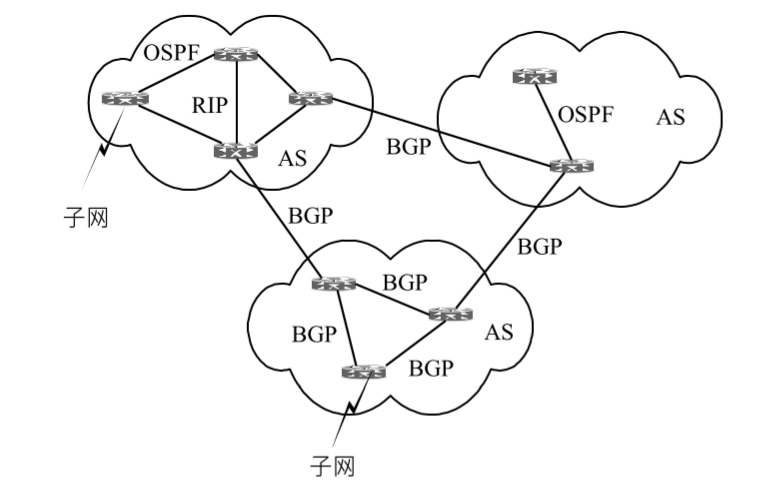
[](https://raw.githubusercontent.com/focus7eleven/TeamOfNetwork/master/Homework_1/src/router_gdr/route_structure.PNG)

图3.1 路由结构示意图

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AS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。

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。

### 3.3 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。

#### 3.3.1 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3.2所示的路由器和网络。

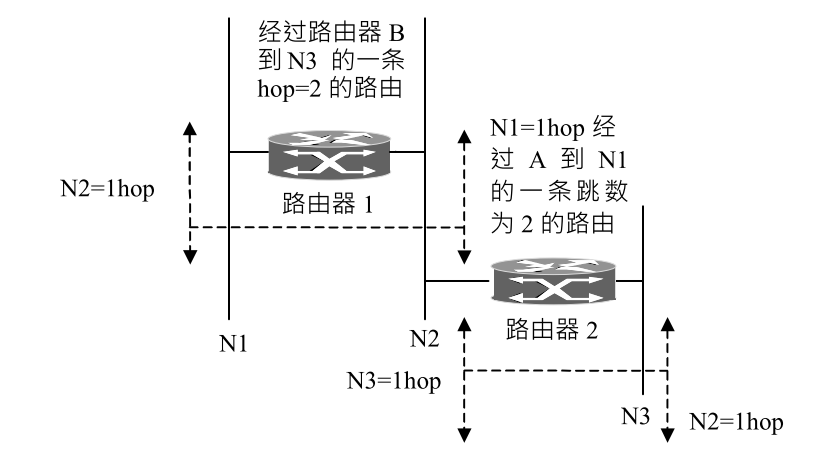
[](https://raw.githubusercontent.com/focus7eleven/TeamOfNetwork/master/Homework_1/src/router_gdr/%E5%BA%A6%E9%87%8F%E5%92%8C%E7%AE%A1%E7%90%86%E8%B7%9D%E7%A6%BB.PNG)

图3.2 度量和管理距离

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。

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。
* 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3.3所示。

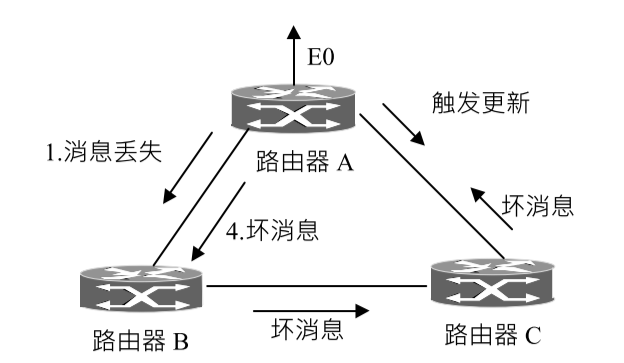
[](https://raw.githubusercontent.com/focus7eleven/TeamOfNetwork/master/Homework_1/src/router_gdr/%E8%AE%B0%E6%95%B0%E5%88%B0%E6%97%A0%E7%A9%B7%E5%A4%A7.PNG)

图3.3 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3.4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 学来的路由信息。

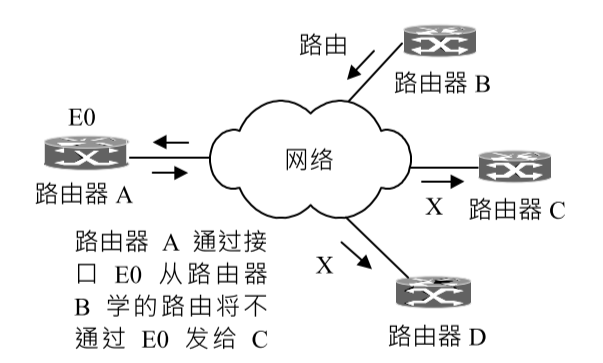
[](https://raw.githubusercontent.com/focus7eleven/TeamOfNetwork/master/Homework_1/src/router_gdr/%E6%B0%B4%E5%B9%B3%E5%88%86%E5%89%B2.PNG)

图3.4 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3.5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。

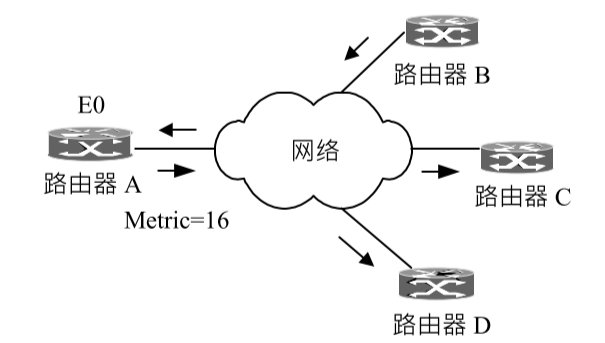
[](https://raw.githubusercontent.com/focus7eleven/TeamOfNetwork/master/Homework_1/src/router_gdr/%E6%AF%92%E6%80%A7%E9%80%86%E8%BD%AC.PNG)

图3.5 毒性逆转

（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 记数到无穷大 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。

#### 3.3.2 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3.6所示。

[](https://raw.githubusercontent.com/focus7eleven/TeamOfNetwork/master/Homework_1/src/router_gdr/RIP%E6%8A%A5%E6%96%87%E6%A0%BC%E5%BC%8F.PNG)

图3.6 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 的报文格式如下图所示，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。

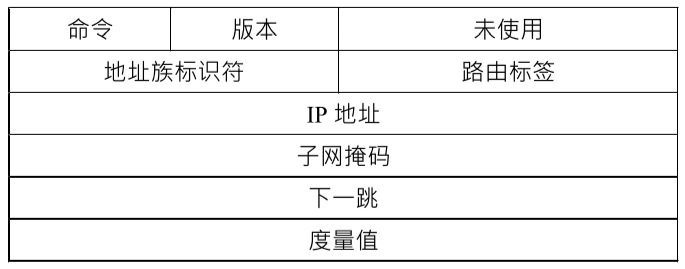
[](https://raw.githubusercontent.com/focus7eleven/TeamOfNetwork/master/Homework_1/src/router_gdr/RIP%20v2%E6%8A%A5%E6%96%87%E6%A0%BC%E5%BC%8F.PNG)

图3.7 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3.8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。

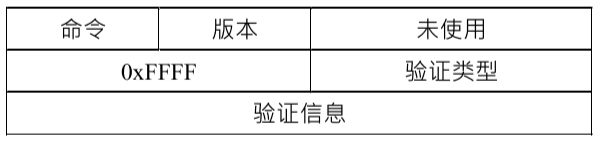
[](https://raw.githubusercontent.com/focus7eleven/TeamOfNetwork/master/Homework_1/src/router_gdr/%E5%B8%A6%E6%9C%89%E8%AE%A4%E8%AF%81%E4%BF%A1%E6%81%AF%E7%9A%84RIP%E6%8A%A5%E6%96%87%E6%A0%BC%E5%BC%8F.PNG)

图3.8 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。

### 3.4 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。

#### 3.4.1 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3.9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。

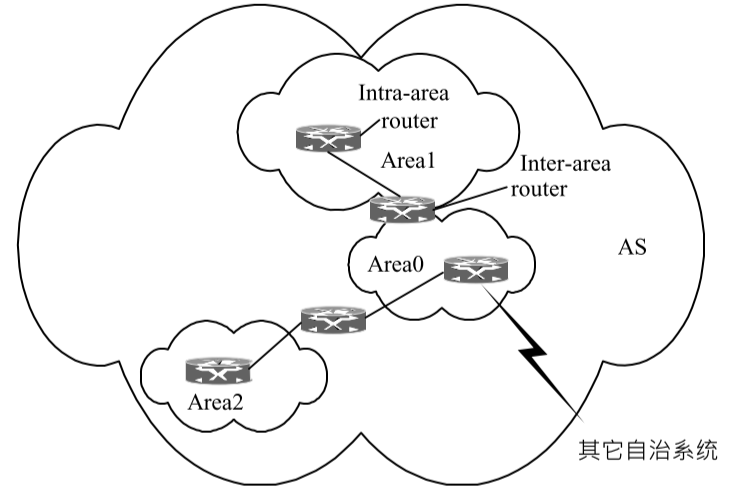
[](https://raw.githubusercontent.com/focus7eleven/TeamOfNetwork/master/Homework_1/src/router_gdr/OSPF%E5%8C%BA%E5%9F%9F.PNG)

图3.9 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。

#### 3.4.2 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3.10所示的简单网络采用表3.1的链路状态数据库表示各种数据。

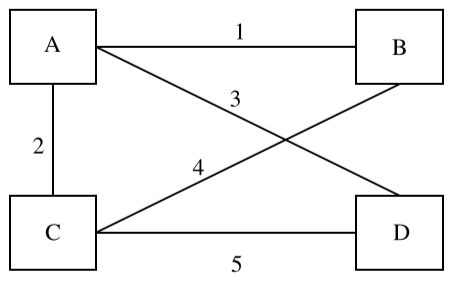
[](https://raw.githubusercontent.com/focus7eleven/TeamOfNetwork/master/Homework_1/src/router_gdr/%E7%AE%80%E5%8D%95%E7%BD%91%E7%BB%9C%E6%8B%93%E6%89%91%E5%9B%BE.PNG)

图3.10 简单网络拓扑图

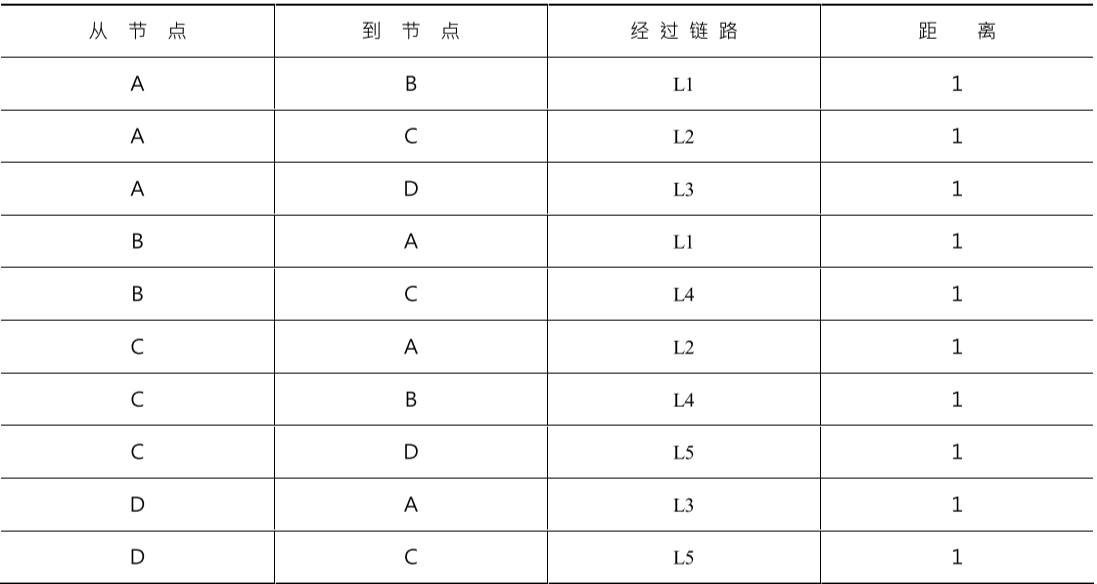
[](https://raw.githubusercontent.com/focus7eleven/TeamOfNetwork/master/Homework_1/src/router_gdr/%E9%93%BE%E8%B7%AF%E7%8A%B6%E6%80%81%E6%95%B0%E6%8D%AE%E5%BA%93%E8%A1%A8.PNG)

表3.1 链路状态数据库表

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TransitNetworks，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）。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。

#### 3.4.3 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。

### 3.5 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。

#### 3.5.1 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。  
（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3.11所示为一个路由聚合的示例。

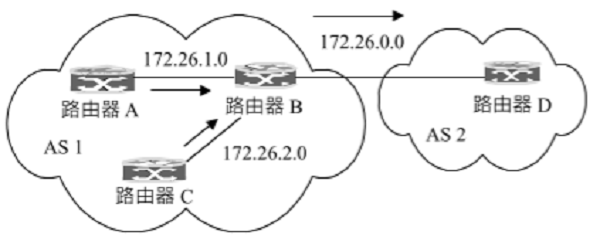
[](https://raw.githubusercontent.com/focus7eleven/TeamOfNetwork/master/Homework_1/src/router_gdr/%E8%81%9A%E5%90%88%E8%B7%AF%E7%94%B1.PNG)

图3.11 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。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。

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。

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。

#### 3.5.2 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。

1. 路由器配置

本论文主要涉及路由器的配置相关内容，主要包括路由器基础、路由器配置、cisco ios使用方法、路由器常用配置和路由器配置实验五个方面。

路由器基础中主要从路由器的功能和cisco路由器系统组成两方面做了介绍，描述了其包括的内容以及主要关注点。。

路由器配置主要从配置途径、ios的启动与系统配置对话和路由器状态以及配置模式三方面进行了解析，给出了详细的配置步骤。

cisco ios使用方法模块，首先介绍了使用帮助的用法，然后介绍了命令行的注释和默认配置，如何显示路由器状态和查看相邻的网络设备，重点介绍了ios及配置文件的备份，同时还介绍了配置控制、改变工作模式命令、口令管理以及路由器测试命令相关内容。

在路由器常用配置模块，主要介绍了ip协议的配置、ip路由配置、路由协议配置、广域网协议配置和NAT配置的相关原理，并对相关配置命令格式进行了解析。

在路由器配置实验中，通过具体的实验拓扑图为指导，描述了各种协议的具体配置过程。

本文从各个方面对路由器的配置进行了描述和解析，对于理解路由器的配置各方面的原理以及具体如何配置有很好的参考价值。

4.1 路由器基础

4.1.1路由器的功能

路由器是在网络层实现互联的设备。路由器实现网络层上数据包的存储转发，它具有路径选择功能，可依据网络当前的拓扑结构，选择“最佳”路径，把接收的数据包转发出去，从而实现网络负载平衡，减少网络拥塞路由器工作在网络层，用于连接不同的局域网和广域网，故称为“LAN网间互联设备”。一个路由器可以连接两个局域网、一个局域网和一个广域网，或两个广域网。

路由器的具体功能如下：

* 路由功能（寻径功能）——寻找并记录到达目的网段的最佳路径，体现在路由器上则包括路由表的建立、维护和查找；
* 交换功能——路由器的交换功能与以太网交换机执行的交换功能不同，路由器的交换功能是指在网络之间转发分组数据的过程，涉及到从接收接口收到数据帧，解封装，对数据包做相应处理，根据目的网络查找路由表，决定转发接口，做新的数据链路层封装等过程。
* 隔离广播、指定访问规则——路由器阻止广播的通过，并且可以设置访问控制列表(ACL)对流量进行控制。
* 异种网络互连——支持不同的数据链路层协议，可以连接异种网络。
* 子网间的速率匹配——路由器有多个接口，不同接口具有不同的速率，路由器需要利用缓存及流控协议进行速率适配。

路由器的主要任务是把通信引导到目的地网络，然后到达特定的节点站地址。后一个功能是通过网络地址分解完成的。例如，把网络地址部分的分配指定成网络、子网和区域的一组节点，其余的用来指明子网中的特别站。分层寻址允许路由器对有很多个节站的网络存储寻址信息。在广域网范围内的路由器按其转发报文的性能可以分为两种类型，即中间节点路由器和边界路由器。尽管在不断改进的各种路由协议中，对这两类路由器所使用的名称可能有很大的差别，但所发挥的作用却是一样的。中间节点路由器在网络中传输时，提供报文的存储和转发。同时根据当前的路由表所保持的路由信息情况，选择最好的路径传送报文。由多个互连的LAN组成的公司或企业网络一侧和外界广域网相连接的路由器，就是这个企业网络的连界路由器。它从外部广域网收集向本企业网络寻址的信息，转发到企业网络中有关的网络段；另一方面集中企业网络中各个LAN段向外部广域网发送的报文，对相关的报文确定最好的传输路径。

4.1.2 cisco路由器系统组成

1. CPU

与计算机一样，路由器也包含了一个中央处理器（CPU）。不同系列和型号的路由器，其中的CPU也不尽相同。Cisco路由器一般采用Motorola 68030和Orion/R4600两种处理器。

路由器的CPU负责路由器的配置管理和数据包的转发工作，如维护路由器所需的各种表格以及路由运算等。路由器对数据包的处理速度很大程度上取决于CPU的类型和性能。

1. 存储器

ROM:存储开机诊断程序，用于引导操作系统，类似于计算机的BIOS

RAM:路由器的主存储器，存放Running-config，路由器，ARP表，类似于计算机的内存。

FLASH:路由器的快闪存储器，用于存放路由器的IOS，类似于计算机硬盘。

NVRAM:非易失存储器，用于放置启动配置文件Startup-Config文件

1. 接口

所有路由器都有接口（Interface），每个接口都有自己的名字和编号。一个接口的全名称由它的类型标志与数字编号构成，编号自0开始。

对于接口固定的路由器（如Cisco 2500系列）或采用模块化接口的路由器（如Cisco 4700系列），在接口的全名称中，只采用一个数字，并根据它们在路由器的物理顺序进行编号，例如Ethernet0表示第1个以太网接口，Serial1表示第2个串口。

对于支持“在线插拔和删除”或具有动态更改物理接口配置的路由器，其接口全名称中至少包含两个数字，中间用斜杠“/”分割。其中，第1个数字代表插槽编号，第2个数字代表接口卡内的端口编号。如Cisco 3600路由器中，serial3/0代表位于3号插槽上的第1个串口。

对于支持“万用接口处理器（VIP）”的路由器，其接口编号形式为“插槽/端口适配器/端口号”，如Cisco 7500系列路由器中，Ethernet4/0/1是指4号插槽上第1个端口适配器的第2个以太网接口。

**i**控制台端口

　　几乎所有路由器都在路由器背后安装了一个控制台端口。控制台端口提供了一个EIA/TIA—232(以前叫作RS—232)异步串行接口、使我们能与路由器通信。至于同控制台端口建立哪种形式的物理连接，则取决于路由器的型号。有些路由器采用一个DB25母连接(DB25F)，有些则用RJ45连接器。通常，较小的路由器采用RJ45控制台连接器，而较大路由器采用DB25控制台连接器。

**ii**辅助端口

大多数Cisco路由器都配备了一个“辅助端口”(Auxiliary Port)。它和控制台端口类似，提供了一个EIA／TIA—232异步串行连接，使我们能与路由器通信。辅助端口通常用来连接Modem，以实现对路由器的远程管理。远程通信链路通常并不用来传输平时的路由数据包，它的主要的作用是在网络路径或回路失效后访问一个路由器。

1. IOS

IOS为CISCO的专有操作系统，功能有连接多种网络，用于不同协议的路由和转换，实现流量控制、QoS服务质量控制、网络安全服务，网络拨号及VPN等。

有两种类型的IOS配置：

1)运行配置：有时也称作“活动配置”，驻留于RAM，包含了目前在路由器中“活动”的IOS配置命令。配置IOS时，就相当于更改路由器的运行配置。

2)启动配置：启动配置驻留在NVRAM中，包含了希望在路由器启动时执行的配置命令。有时也把启动配置称作“备份配置”。这是由于修改并认可了运行配置后，通常应将运行配置复制到NVRAM里，将作出的改动“备份”下来，以便路由器下次启动时调用。启动完成后，启动配置中的命令就变成了“运行配置”。

两者均以ASCII文本格式显示。所以，我们能够很方便地阅读与操作。一个路由器只能从这两种类型中选择一种。

4.2 路由器配置

4.2.1路由器配置途径

1、 控制台

将PC机的串口直接通过Rollover线与路由器控制台端口Console相连，在PC计算机上运行终端仿真软件，与路由器进行通信，完成路由器的配置。也可将PC与路由器辅助端口AUX直接相连，进行路由器的配置。

2、 虚拟终端(Telnet)

　　如果路由器已有一些基本配置，至少有一个端口有效(如Ethernet口)，就可通过运行Telnet程序的计算机作为路由器的虚拟终端与路由器建立通信，完成路由器的配置。

3、 网络管理工作站

　　路由器可通过运行网络管理软件的工作站配置，如Cisco的CiscoWorks、HP的OpenView等。

4、 CISCO　ConfigMaker

　　ConfigMaker是一个由CISCO开发的免费的路由器配置工具。ConfigMaker采用图形化的方式对路由器进行配置，然后将所做的配置通过网络下载到路由器上。ConfigMaker要求路由器运行在IOS　11.2以上版本，可用Show Version命令查看路由器的版本信息。

5、 TFTP(Trivial File Transfer Protocol)服务器

TFTP是一个TCP/IP简单文件传输协议，可将配置文件从路由器传送到TFTP服务器上，也可将配置文件从TFTP服务器传送到路由器上。TFTP不需要用户名和口令，使用非常简单。

**注意**：路由器的第一次设置必须通过第一种方式进行；这时终端的硬件设置为波特率：9600，数据位：8，停止位：1，无校验。

4.2.2 ios的启动和系统配置对话

1. **ios的启动**

ios的正确启动包含以下几个步骤：

1. 自ROM执行上电、检测CPU、内存、接口电路；
2. 引导操作系统（由配置寄存器的引导域决定：FLASH或网络下载，Boot system命令确定）
3. 自ROM执行引导，将操作系统下载到主存；
4. 操作系统下载到低地址内存，下载后操作系统确定路由器的工作，硬件和软件部分，分别在屏幕上显示其结果。
5. NVRAM中存储的配置文件装载到主内存，并检测通行词，配置命令启动路由进程，提供接口地址、设置介质特性。如果NVRAM中没有有效的配置文件，则进入setup会话模式。
6. 进入系统配置会话，提示配置信息，如每个接口的配置信息

(2)系统配置对话

cisco路由器的启动和系统配置对话过程如下：

1) 用CISCO随机带CONSOLE线，一端连在CISCO路由器的CONSOLE口,一端连在计算机的COM口。

2) 打开电脑，启动超级终端.为您的连接取个名字,比如CISCO\_SETUP,下一步选定连接时用COM1,下一步选定第秒位数9600,数据位8,奇偶校验无，停止位1,数据流控制无.最后选确定。

3) 打开路由器电源，这时超级终端将出现以下画面：

　System Bootstrap, Version 11.1(20)AA2, EARLY DEPLOYMENT RELEASE SOFTWARE (fc1) Copyright (c) 1999 by cisco Systems, Inc.C3600 processor with 32768 Kbytes of main memory Main memory is configured to 64 bit mode with parity disabled program load complete, entry point: 0x80008000, size: 0x4ed478 Self decompressing the image : [OK] Restricted Rights Legend Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subparagraph 　(c) of the Commercial Computer Software - Restricted 　　Rights clause at FAR sec. 52.227-19 and subparagraph 　　(c) (1) (ii) of the Rights in Technical Data and Computer 　　Software clause at DFARS sec. 252.227-7013. 　　cisco Systems, Inc. 　　170 West Tasman Drive 　　San Jose, California 95134-1706 　　Cisco Internetwork Operating System Software 　　IOS (tm) 3600 Software (C3640-I-M), Version 12.1(2)T, RELEASE SOFTWARE (fc1) 　　Copyright (c) 1986-2000 by cisco Systems, Inc. 　　Compiled Tue 16-May-00 12:26 by ccai 　　Image text-base: 0x600088F0, data-base: 0x60924000 　　cisco 3640 (R4700) processor (revision 0x00) with 24576K/8192K bytes of memory. 　　Processor board ID 25125768 　　R4700 CPU at 100Mhz, Implementation 33, Rev 1.0 　　Bridging software. 　　X.25 software, Version 3.0.0. 　　2 FastEthernet/IEEE 802.3 interface(s) 　　1 Serial network interface(s) 　　DRAM configuration is 64 bits wide with parity disabled. 　　125K bytes of non-volatile configuration memory. 　　8192K bytes of processor board System flash (Read/Write)

--- System Configuration Dialog ---

Would you like to enter the initial configuration dialog? [yes/no]: **y**

<!--您是否进入初始化配置对话，选Y-->

At any point you may enter a question mark '?' for help.

Use ctrl-c to abort configuration dialog at any prompt.

Default settings are in square brackets '[]'.Basic management setup configures only enough connectivity

for management of the system, extended setup will ask you

to configure each interface on the system

Would you like to enter basic management setup? [yes/no]: **n**

<!--您是否进入基本配置安装，选N-->

First, would you like to see the current interface summary? [yes]: **y**

<!--首先，您是否看一下当前端口状态-->

Any interface listed with OK? value "NO" does not have a valid configuration

Interface IP-Address OK? Method Status Protocol

FastEthernet0/0unassigned NO unset up down

Serial0/0 unassigned NO unset down down

FastEthernet0/1unassigned NO unset up down

Configuring global parameters:

Enter host name [Router]:**RouterA**

<!--输入路由器的名字-->

The enable secret is a password used to protect access to

privileged EXEC and configuration modes. This password, after

entered, becomes encrypted in the configuration.

Enter enable secret: **aaa**

<!--输入密文-->

The enable password is used when you do not specify an

enable secret password, with some older software versions, and

some boot images.

Enter enable password: **bbb**

<!--输入密码(不能和密文相同)-->

The virtual terminal password is used to protect

access to the router over a network interface.

Enter virtual terminal password: **ccc**

<!--输入虚拟终端的密码(以备远程登录)-->

Configure SNMP Network Management? [yes]: **n**

<!--配置简单网管吗?选N-->

Configure IP? [yes]: **y**

<!--配置IP吗?选Y-->

Configure IGRP routing? [yes]: **n**

<!--配置IGRP路由选择协议吗?选N-->

Configure RIP routing? [no]: 　**n**

<!--配置RIP路由选择协议吗?选N-->

Configure bridging? [no]: **n**

<!--配置桥接吗?选N-->

Async lines accept incoming modems calls. If you will have

users dialing in via modems, configure these lines.

Configure Async lines? [yes]: **n**

<!--配置异步线路吗?选N-->

Configuring interface parameters:

Do you want to configure FastEthernet0/0 interface? [yes]: **y**

<!--您是否想配置fastethernet0/0接口?选Y-->

Use the 100 Base-TX (RJ-45) connector? [yes]: **y**

<!--用RJ45的连接器吗?选Y-->

Operate in full-duplex mode? [no]: **y**

<!--选用全双工模式?选Y-->

Configure IP on this interface? [yes]: **y**

<!--在这个接口上配置IP吗?选Y-->

IP address for this interface: 192.168.0.1

<!--配置该接口的IP地址(在此地址为192.168.0.1-->

Subnet mask for this interface [255.255.255.0] :

<!--配置该接口的子网掩码.(默认的是255.255.255.0,可以手工输入修改)-->

Class C network is 192.168.0.0, 24 subnet bits; mask is /24

Do you want to configure Serial0/0 interface? [yes]: **y**

<!--您想配置serial0/0接口吗?选Y-->

Some supported encapsulations are

ppp/hdlc/frame-relay/lapb/x25/atm-DXi/smds

Choose encapsulation type [hdlc]:

选择封装方式(默认的封装方式是HDLC,您可根据与您的路由器相连选用的封装类型来决定用什么样的封装类型

No serial cable seen.

Choose mode from (dce/dte) [dte]:

(因为没有连串口线所以会让您选择设备类型)

Configure IP on this interface? [yes]: **y**

(在接口上配置IP)

Configure IP unnumbered on this interface? [no]:

IP address for this interface: 172.16.0.5

配置该接口的IP地址(在此地址为172.16.0.5)

Subnet mask for this interface [255.255.0.0] : 255.255.255.252

配置该接口的子网掩码.(默认的是255.255.0.0,可以手工输入修改为255.255.255.252)

Class B network is 172.16.0.0, 30 subnet bits; mask is /30

(以下配置同上)

Do you want to configure FastEthernet0/1 interface? [yes]:

Use the 100 Base-TX (RJ-45) connector? [yes]:

Operate in full-duplex mode? [no]: y

Configure IP on this interface? [yes]: y

IP address for this interface: 172.16.0.9

Subnet mask for this interface [255.255.0.0] : 255.255.255.252 Class B network is 172.16.0.0, 30 subnet bits; mask is /30

The following configuration command script was created:

(把您的配置显示出来)

hostname aaa

enable secret 5 $ul/V$ezbZFgvzGHD.YPSieC0Ew/

enable password RouterA

line VTY 0 4

password ccc

no snmp-server

!

ip routing

no bridge 1

!

interface FastEthernet0/0

media-type 100BaseX

full-duplex

ip address 192.168.0.1 255.255.255.0

!

interface Serial0/0

encapsulation hdlc

ip address 172.16.0.5 255.255.255.252

!

interface FastEthernet0/1

media-type 100BaseX

full-duplex

ip address 172.16.0.9 255.255.255.252

dialer-list 1 protocol ip permit

dialer-list 1 protocol ipx permit

!

end

以下提示您是否保存这次设置

[0] Go to the IOS command prompt without saving this config.

[1] Return back to the setup without saving this config.

[2] Save this configuration to nvram and exit.

Enter your selection [2]: 2

选择2保存设置并存入NVRAM中

Building configuration...

[OK] Use the enabled mode 'configure' command to modify this configuration.

Press RETURN to get started

4.2.3路由器状态以及配置模式

路由器的配置模式是通过控制台连接路由器进入的模式，该模式下路由器有以下几个状态。

(1)用户命令状态

前置符类似“Router>”，此时路由器处于用户命令状态，这时用户可以看路由器的连接状态，访问其它网络和主机，但不能看到和更改路由器的设置内容。

(2)特权命令状态

前置符类似“Router#”，用户命令状态下输入“enable”即可进入，此时路由器处于特权命令状态，这时不但可以执行所有的用户命令，还可以看到和更改路由器的设置内容。

(3)全局设置状态

前置符类似“Router(config)#”，特权命令状态下输入“configure terminal”即可进入，此时路由器处于全局设置状态，这时可以进行路由器端口以外的一些设置，如：路由协议，nat等。

(4)局部设置状态

从全局设置状态进入，对某个功能的详细设置，这时可以设置路由器某个局部的参数。

(5)RXBOOT状态

前置符为“>”，在开机后60秒内按ctrl-break可进入此状态，这时路由器不能完成正常的功能，只能进行软件升级和手工引导。

(6)设置对话状态

这是一台新路由器开机时自动进入的状态，在特权命令状态使用SETUP命令也可进入此状态。这时可通过对话方式对路由器进行设置。

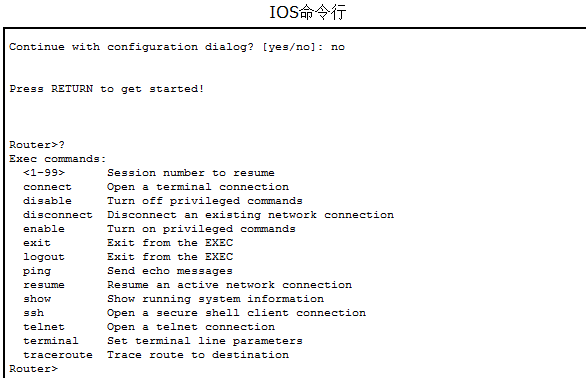
4.3 cisco ios使用方法

Cisco IOS（Intenetwork Operating System）是运行于Cisco路由器和交换机上的操作系统，其能提供多种网络服务，进而支持多种网络应用。IOS是被用来传送网络服务并启动网络应用的。能用来加载网络协议和功能、在设备间连接高速流量、在控制访问中添加安全性防止未授权的网络使用、为简化网络的增长和冗余备份，提供可缩放性、为连接到网络中的资源，提供网络的可靠性。

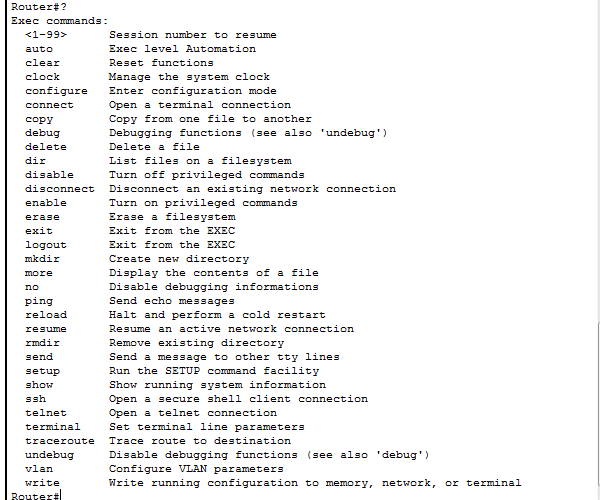
4.3.1使用帮助

在电脑上装好Packet Tracer之后，即可以使用IOS命令了。在Packet Tracer中拖动路由器图标到主串口，双击路由器即可打开配置窗口。第三个Tab页就是使用IOS命令的命令行工具。在这里输入命令即可得到执行。

在用户模式下输入？即可得到命令帮助。如下图：



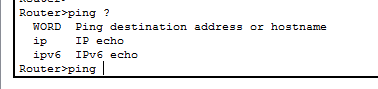
在特权模式下输入？同样可以得到帮助命令，帮助命令稍多。



关于用户模式和特权模式请见6、改变工作模式命令。

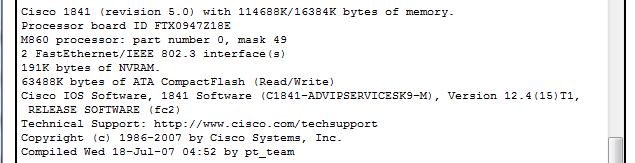
4.3.2命令行的注释和默认设置

1. 注释：很多命令都带有参数，用户输入命令的时候，命令行工具会执行输入命令，会检查命令的正确与否。在1、使用帮助中我们看到了使用？可以查看命令，同样可以看到输入命令的解释和参数的情况。比如使用常用的ping命令。



可以看到起参数后面是一个ip地址。

1. 默认设置：路由器启动的时候，IOS命令行会自动输出默认设置。我们可以关闭路由器然后开启，就可以看到路由器的默认配置。



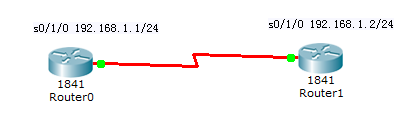
图中包含了路由器的型号、内存使用情况、处理器板ID、NVRAM的大小等一系列路由器的默认设置信息。

4.3.3显示路由器状态和查看相邻的网络设备

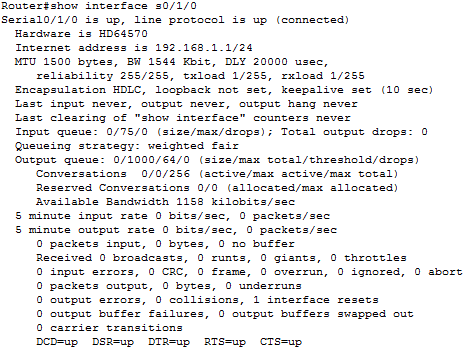
路由器的状态影响着网络的工作情况，所以经常需要了解路由器的状态，比如端口是开启还是关闭、设备是否连接通等问题。这些命令包括：

* show running-config显示当前运行在RAM中的路由器配置。
* show startup-config显示存储在NVRAM中的路由器配置。这是在路由器接通电源的时候使用的配置，除非进行了其他特殊的配置。
* show flash 显示闪存中的格式和内容，包括闪存中的IOS映像的文件名称。
* Show buffers 显示路由器上的缓冲区的统计信息。
* show mem显示路由器内存的统计信息，包括自由库统计信息。
* Show processec cpu显示路由器中正在运行的活动过程或程序的统计信息。
* show protocols 显示路由器中配置的所有协议的信息，以及在每个接口上配置的关于网络层地址的信息。
* Show stacks 显示关于过程对内存堆栈的使用以及中断例程的信息，还包括上次系统重新启动的原因。
* show version显示系统硬件的配置信息、软件版本、配置文件的名称和来源，以及启动映像。
* show ip route显示路由器的路由表
* Show interfaces后面还跟有参数，显示某接口的具体情况

上面我们用于查看路由器组件和过程状态的命令总称为SHOW命令。下面我们用一个简易的拓扑来检查路由器状态。



使用show interface为例来查看路由器接口状态：



从上图中可以看出，第一行显示Router0路由器的Serial0/1/0端口已经开启，第二行显示硬件版本是HD64570，第三行显示网络地址是192.168.1.1/24，第五行显示封装协议是HDLC，回环地址没有设置。

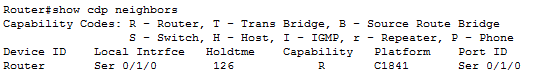
除了关注自己的配置，路由器还需要查看相邻的网络设备的各种状态。连接端口的情况可以通过上述show interfaces命令查看。另外，需要通过使用  查看邻居路由 命令：

show cdp interface

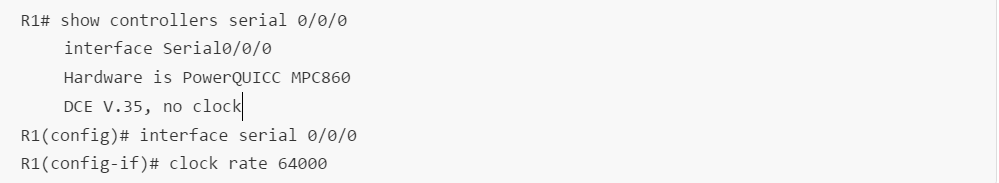
show cdp neighbors [detail]

show cdp entry routerA

通过这三个命令可以查看邻居的路由。cdp是Cisco Discovery protocol，思科发现协议。下图是用上图的网络拓扑图中的Router0进行查看相邻的网络设备：

 如图所示，在Router0的控制台上输入“show cdp neighbors”命令后，控制台输出与Router0相邻的网络设备Router1，其中的Local Intrfce表示路由器上的接口，Capability包括了路由器、交换机、主机等类型，Port ID表示连接在远程路由器上的接口。

默认cisco串行接口为DTE设备，但也可以将他们配置为DCE设备。DTE电缆的连接器为插头型，而DCE电缆的连接器为插孔型，可以通过“show controllers”命令确定路由器连接的电缆是DCE还是DTE，DCE设备需要通过“clock rate”命令来设置时钟。



4.3.4 ios及配置文件的备份

为了方便管理，我们可以 把IOS系统及设备配置文件备份在本地计算机上，当出故障时，我们可以把IOS系统或配置文件恢复到Cisco设备上。

路由器可以将配置信息复制到TFTP服务器上，或从TFTP服务器上复制配置信息。这使网络管理员可以将配置信息保存在服务器上，以跟踪配置、监视修改或者灾难恢复。如果配置大于32000字节，则需要将配置保存在TFTP服务器上，因为32000字节是NVRAM可以保存的最大配置文件。当用TFTP将配置文件传送到路由器时，可以将其放置在闪存、NVRAM或RAM内存中。当将配置放置在闪存中时，仍然需要将其放置在NVRAM或RAM，目的是路由器可以使用它。COPY TETP命令可以通过控制台或VTY会话来执行。

复制配置文件到TFTP服务器或从TFTP服务器复制配置文件的命令如下：

* Copy tftp running-config直接从TFTP服务器复制配置文件到RAM，而配置路由器。
* Copy tftp startup-config用来自TFTP服务器的文件覆盖存储在NVRAM中的配置文件。
* Copy running-config tftp在TFTP服务器上对RAM中的路由器正在运行的配置进行复制。
* Copy startup-config tftp将NVRAM中的配置复制到TFP服务器上。
* Copy flash tftp把flash存储器内的ios文件备份到TFTP server上。
* Earase 命令是删除配置文件，慎用。

路由器的配置参数较多，可根据实际需要增减。同时在1使用帮助中提过？帮助命令，可以查看copy命令的帮助和解释以及参数的情况，根据实际情况，做相应的备份。

4.3.5配置控制

使用Cisco ios配置路由器的命令和过程是复杂的，有时候需要保存此次的配置文件以便下次使用，或者载入上次的同样配置文件继续做模拟实验，需要对配置进行控制。

在使用有些软件时可以通过一些界面化操作来载入和导出配置文件。如packet tracer的界面化操作：



如果不是模拟软件，则需要使用配置导入命令来导入配置文件。在3.4ios及配置文件的备份中我们已经了解到了配置文件如何备份。接下来描述配置文件恢复和载入到当前的路由器中。

1)确保tftp服务器处于开启状态，以及要导入的配置文件正确无误，适合路由器本身的各项参数;

2)清除路由器原来的startup-config文件,并reload路由器;

配置路由器Fastenternet口的临时IP地址，保证和tftp服务器正常通信;

开启路由器并进入全局模式，将tftp服务器上的配置文件复制到路由器上;

执行的命令copy tftp startup-config

3)接着输入tftp服务器的IP地址，例如：172.16.20.223

4)输入要导入的文件名，要和服务器上的文件名保持一致;

5)将startup-config文件复制到running-config中，copy startup-config

running-config

6)在路由器上show running，查看路由信息是否与导入的配置信息保持一致;

为了更方便的进行各种操作，更好的在IOS中管理配置文件，需要了解以下命令：

* dir 这条指令用来显示文件夹下的文件列表，输入dir ? 可以查看可选参数
* cd 改变路径，改变当前所在的路径
* copy 这个命令用来将 IOS 或一个配置文件拷贝到某处。可以用这个命令将路由器配置文件拷贝到 TFTP服务器上，或者拷贝到路由器里的某个文件夹中作为备份。还可以用 copy 命令将新的IOS 文件从TFTP服务器拷贝到路由器里，实现路由器升级
* delete 和 rm 两个命令都很简单 delete 用来删除文件， rm 用来删除文件夹
* show flash 用来显示flash中的文件。show flash 命令和 dir flash 命令类似，但是前者比后者显示出的信息更丰富一些，即多出了flash内存大小和类型信息
* erase 和 format 需要知道应该format flash中的文件系统，而erase nvram里的文件系统。其余文件则根据其类型既可以erase又可以format。erase 命令大多数时候都是用在清除路由器配置，恢复出场配置的情况。具体的命令就是 erase startup-configuration
* more 这个命令可以显示文本/配置文件的内容。比如你想查看一个备份的配置文件，就可以使用more 命令来查看该文件的内容
* verify 这个个命令用来核查或者计算一个文件的MD5校验和
* mkdir 和DOS环境一样，可以在路由器中使用 mkdir 命令创建文件夹。一般用这个命令来创建备份文件夹，用来存储配置文件或者ISO文件的备份。
* fsck FAT 文件系统检测主要是用来检测flash文件系统的完整性。如果你感觉ISO文件有损坏，可以通过这个命令对文件系统进行检查。

4.3.6改变工作模式命令

路由器配置工作模式分为用户(USER)模式、特权(privileged)模式、全局模式、端口配置模式和协议配置模式。

USER模式的特性：用户模式仅允许基本的监测命令，在这种模式下不能改变路由器的配置，router>的命令提示符表示用户正处在USER模式下。

privileged模式的特性：特权模式可以使用所有的配置命令，在用户模式下访问特权模式一般都需要一个密码，router#的命令提示符表示用户正处在特权模式下。

全局模式：全局模式是用于用户配置协议及端口的属性，router(config)#的命令提示符表示用户正处在全局模式下。

端口配置模式：端口配置模式是用于配置端口属性，router(config-if)#的命令提示符表示用户正处在端口配置模式下。

协议配置模式：协议配置模式是用于配置路由协议，router(config-router)#的命令表示提示符表示用户正处在路由协议配置模式下。

模式切换如下：

从用户模式router>切换到特权模式router#，使用如下命令:router>enable(第一次启动路由器时不需要密码)，这时，路由器的命令提示符变为router#。

从特权模式router#切换到用户模式router>，使用如下命令:router#exit(第一次启动路由器时不需要密码)，这时，路由器的命令提示符变为router>。

从特权模式router#切换到全局模式router(config)#，使用如下命令:router#conf t，这时，路由器的命令提示符变为router(config)#。

从全局模式router(config)#切换到特权模式router#，使用如下命:router(config)#

exit，这时，路由器的命令提示符变为router#。

从全局模式router(config)#切换到端口配置模式router(config-if)#，使用如下命令:router#int f0/0(f0/0可以是其它端口，比如s0/0)，这时，路由器的命令提示符变为router(config-if)#。

从端口配置模式router(config-if)#切换到全局模式router(config)#，使用如下命令:router(config-if)#exit，这时，路由器的命令提示符变为router(config)#。

从全局模式router(config)#切换到协议配置模式router(config-router)#，使用如下命令: router(config)#router rip(rip可以是其它路由协议)，这时，路由器的命令提示符变为router(config-router)#。

从协议配置模式router(config-router)#切换到全局模式router(config)#，使用如下命令: router(config-router)#exit，这时，路由器的命令提示符变为router(config)#。

4.3.7口令管理

在CISCO路由器产品中，我们在最初进行配置的时候通常需要使用限制一般用户的访问。这对于路由器是非常重要的，在默认的情况下，我们的路由器是一个开放的系统，访问控制选项都是关闭的，任一用户都可以登陆到设备从而进行更进一步的攻击，所以需要我们的网络管理员去配置密码来限制非授权用户通过直接的连接、CONSOLE终端和从拨号MODEM线路访问设备。

关于Cisco IOS 的登录密码以及权限分配设置

enable password pwd 初级密码，用于验证从用户模式到特权模式的验证

enable secret pwd MD5加密密码，同样用于从用户模式到特权模式

enabl password level pwd 指定密码作用于哪个级别

enable secret level xx MD5加密密码，同样用于指定密码作用于哪个级别

配置进入特权模式的密码和密匙：这两个密码是用来限制非授权用户进入特权模式。因为特权密码是未加密的，所以我们一般都推荐用户使用特权密匙，且特权密码仅在特权密匙未使用的情况下才会有效。

router(config)#enable password cisco

命令解释：开启特权密码保护

router(config)#enable secret cisco

命令解释：开启特权密匙保护。

配置控制端口的用户密码

router(config)#line console 0

命令解释：进入控制线路配置模式。

router(config-line)#login

命令解释：开启登陆密码保护

router(config-line)#password cisco

命令解释:设置密码为cisco，这里的密码区分大小写。

配置辅助端口（AUX）的用户密码

router(config)#line aux 0

命令解释:进入辅助端口配置模式。

router(config-line)#login

命令解释：开启登陆密码保护

router(config-line)#password cisco

命令解释:设置密码为cisco，这里的密码区分大小写。

配置VTY(telnet)登陆访问密码

router(config)#line vty 0 4

命令解释:进入VTY配置模式。

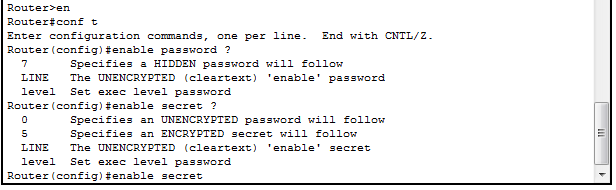
router(config-line)#login

命令解释：开启登陆密码保护

router(config-line)#password cisco

命令解释:设置密码为cisco，这里的密码区分大小写。

通过在一个路由器上查看password命令的详细解释：

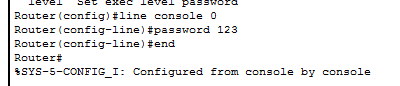


注意这里：0 5 7 的 含义不同。0 为不加密，5 为MD5 。7 为Cisco自有加密算法（不牢靠，容易被破解）。

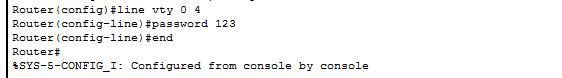
service password-encryption 是使用7的加密方法加密存储在本地的所有密码。Cisco官方不推荐使用这个方法。

下面是简单密码验证：

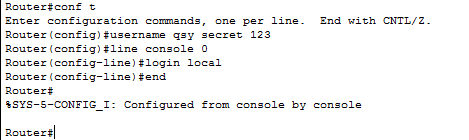
console 验证配置：



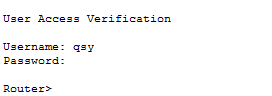
Vty 验证：



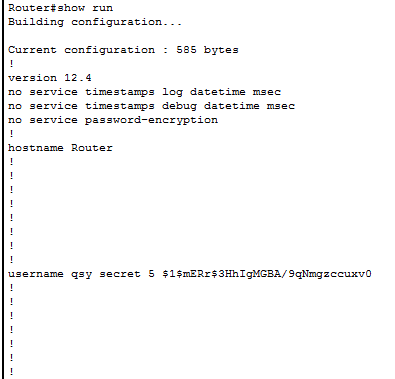
带用户名的密码验证：

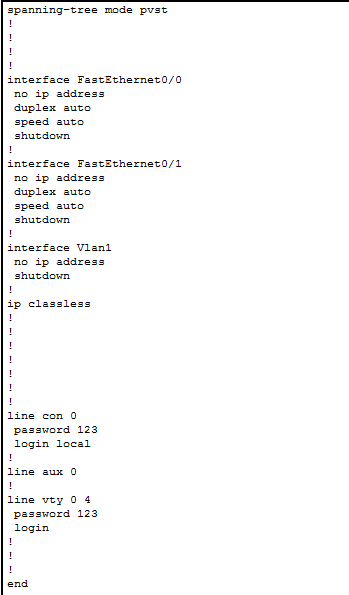


当重新要进入终端时，需要登录：



可以使用show run命令查看配置的情况：





分级的权限验证：如果用户只想让Level 7 的用户在特权模式下：clear counters 和reload

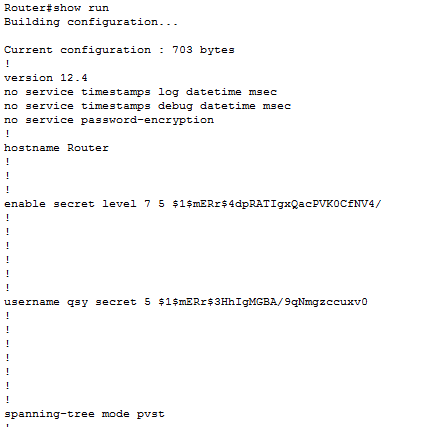
配置如下：

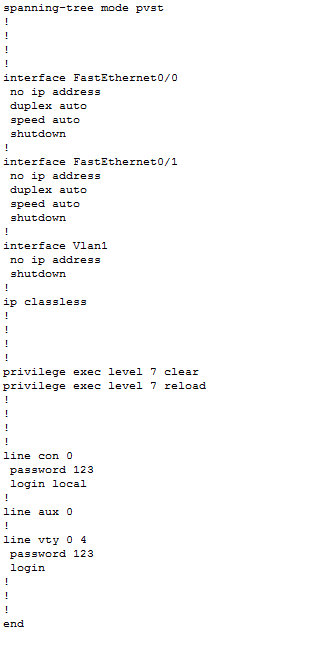
privilege exec level 7 clear counters

privilege exec level 7 reload

enable secret level 7 xxxx

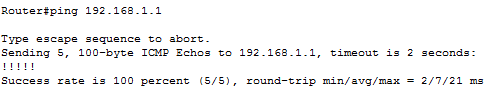
使用show run可以看到配置后的状况：





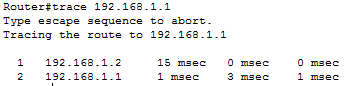
4.3.8路由器测试命令

Cisco IOS软件包括几个命令，它可以用于测试IP网络中的基本连接情况。Ping是一个工具，用于仅仅测试网络层的连接情况。它向目的地发送一系列的ICMP回送数据包，并跟踪目的地发送回的ICMP应答信号。我们可以在用户模式下使用ping的默认特性（5个100字节数据包，2秒钟暂停），但是如果处于特权模式，可以使用其他几个选项。这就是所谓的扩展ping。扩展ping的某些可以使用的其他选项包括：大小不同的数据包，增加暂停时间，一次发送多于5个数据包，在IP报头设置“不分段”位，甚至在其他协议中使用ping，例如IPX和Apple Talk。下图是ping目的主机的IP地址：



Ping测试命令

5个感叹号说明路由器成功地收到了响应数据包。如果不是感叹号，而是点(句号)，则说明连接中断，这或者是因为ICMP回送要求从来就没有达到过目的地，或者是因为响应在网络上的某个地方被损坏了或者路由错误。测试网络层连接情况的另一个命令是TRACEROUTE命令。TRACEROUTE提供这样的信息，即通信量正在互连网上采取哪一条路径，这样的信息是按跳提供的，以及每个跳有多长。这里是一个TRACEROUTE输出的例子：



Traceroute测试命令

4.4 路由器常用配置

4.4.1 IP协议配置

4.4.1.1配置以太网接口

在配置以太网接口时，我们需要为以太网接口配置IP地址及子网掩码来进行IP数据包的处理。默认情况下，以太网接口是管理性关闭的，所以在配置完成ip地址后，我们还需要激活接口。

我们以一个简易的实例来说明以太网接口的IP协议配置。在这个实例中，我们需要为以太网接口配置192.168.0.1的IP地址并且激活接口。

相关配置命令如下：

1、在特权模式下进入全局配置模式。

router#configure terminal

2、进入第一个以太网接口。

router(config)#interface f0/0

3、这时，命令提示符变为router(config-if)#，在提示符后输入以下命令为接口配置私有IP地址192.168.0.1，使用默认子网掩码255.255.255.0。

router(config-if)#ip address 192.168.0.1 255.255.255.0

4、在默认情况下，cisco路由器的接口是在关闭状态下的，我们需要键入“no shutdown”命令来激活接口。

router(config-if)#no shutdown

4.4.1.2配置串行接口

我们可以通过虚拟终端来配置一个串行接口，配置串行接口需要以下步骤：

1、在全局模式下键入命令“interface serial 0”进入到串行接口配置模式下。

2、每一个连接的串行接口都必须有一个IP地址和子网掩码来转发IP数据包，我们可以在接口配置模式下键入“ip address <IP address> <netmask>”的命令来配置串行接口的IP地址。

3、如果串行接口连接的是一个DCE设备，我们还需要为串行接口配置一个时钟频率，如果是DTE设备则不需要。默认情况下，cisco路由器是一个DTE设备，但是我们可以通过使用命令来将其配置成DCE设备。我们可以在串行接口配置模式下键入“clock rate”的命令来配置时钟频率，可利用的时钟频率有“1200、2400、9600、19200, 38400、56000、64000、72000、125000、148000、500000、800000、1000000、1300000、2000000或者4000000。

4、在默认情况下，cisco路由器的接口是在关闭状态下的，我们需要键入“no shutdown”命令来激活接口，如果因为管理的要求，需要关闭一个接口，可以在相应的接口模式下键入“shutdown”就可以管理性关闭这个接口了。

4.4.2 IP路由配置

通过配置静态路由，用户可以人为地指定对某一网络访问时所要经过的路径,在网络结构比较简单，且一般到达某一网络所经过的路径唯一的情况下采用静态路由。

相关配置命令如下：

建立静态路由：ip route prefix mask {address | interface} [distance] [tag tag] [permanent]

Prefix :所要到达的目的网络

mask :子网掩码

address :下一个跳的IP地址，即相邻路由器的端口地址。

interface :本地网络接口

distance :管理距离（可选）

tag tag :tag值（可选）

permanent :指定此路由即使该端口关掉也不被移掉。

以下在Router1上设置了访问192.1.0.64/26这个网下一跳地址为192.200.10.6，即当有目的地址属于192.1.0.64/26的网络范围的数据报，应将其路由到地址为192.200.10.6的相邻路由器。在Router3上设置了访问192.1.0.128/26及192.200.10.4/30这二个网下一跳地址为192.1.0.65。由于在Router1上端口Serial 0地址为192.200.10.5，192.200.10.4/30这个网属于直连的网，已经存在访问192.200.10.4/30的路径，所以不需要在Router1上添加静态路由。

Router1:

ip route 192.1.0.64 255.255.255.192 192.200.10.6

Router3:

ip route 192.1.0.128 255.255.255.192 192.1.0.65

ip route 192.200.10.4 255.255.255.252 192.1.0.65

同时由于路由器Router3除了与路由器Router2相连外，不再与其他路由器相连，所以也可以为它赋予一条默认路由以代替以上的二条静态路由，

ip route 0.0.0.0 0.0.0.0 192.1.0.65

即只要没有在路由表里找到去特定目的地址的路径,则数据均被路由到地址为192.1.0.65的相邻路由器。

4.4.3路由协议配置

路由选择用于划分拥塞的网络，增加网络上允许的节点数目、过滤器数目和管理通信量。路由选择协议用于保证网络上的路由可以使用。度用于确定从起源路由器到目的网络之间的距离或代价。度是一个路由变量值，由路由选择协议计算。路由选择信息表由路由选择协议维护，以帮助选择到达某个目的网络的最佳路径。

距离向量协议是比较古老的路由选择算法协议，它在最少跳的基础上选择路由。跳是在达到目的网络之前，必须经过的路由器的数量。距离相邻协议路由器向它们的相邻的路由器发送整个路由选择信息表。这个信息被拷贝到那个邻居的表中，并用新的跳进行重新计算，然后转发给其他的邻居。这意味着距离向量路由选择表是建立在第2手信息的基础之上的。距离向量协议由于其路由选择表计算的简单性和路由选择算法的简单性，而降低了CPU的开销。距离向量路由选择协议很容易被路由选择循环骚扰，但是通常通过水平分割、破坏逆转，或者暂停间隔以对付这种问题。

链路状态协议用于大型网络。它们在路由选择中使用代价度，并且在链路状态数据库中保存路由选择信息。当一个链路状态路由器位于网络上时，它向它的邻居发送呼叫数据包。邻居用它所了解的和它相连的链路及相关的代价信息来响应。起源路由器在来自邻居的信息的基础上建立它自己的链路状态数据库。一个链路状态路由器将定期向它的邻居发送链路状态通告，包括那个路由器的链路和相关代价。每个邻居复制数据包，并且将LSA转发到下一个邻居，这个过程称为泛洪。因为路由器并不在泛洪LSA之前再次计算路由选择数据库，减少了收敛时间。

4.4.3.1 RIP配置

RIP(Routing information Protocol)是应用较早、使用较普遍的内部网关协议(Interior Gateway Protocol,简称IGP)，适用于小型同类网络，是典型的距离向量(distance-vector)协议。在TCP/IP协议中实际上有两个版本的RIP。版本1是原始的，版本2是更新的版本。版本2由于其增强的功能而得到广泛的使用。

RIP通过广播UDP报文来交换路由信息，每30秒发送一次路由信息更新。RIP提供跳跃计数(hop count)作为尺度来衡量路由距离，跳跃计数是一个包到达目标所必须经过的路由器的数目。如果到相同目标有二个不等速或不同带宽的路由器，但跳跃计数相同，则RIP认为两个路由是等距离的。RIP最多支持的跳数为15，即在源和目的网间所要经过的最多路由器的数目为15，跳数16表示不可达。

RIP相关配置命令如下：

* 指定使用RIP协议：router rip
* 指定RIP版本 ：version {1|2}
* 指定与该路由器相连的网络 ：network network
* 配置好以后使用“show ip route”或者“show ip protocols”命令进行查看。

4.4.3.2 IGRP配置

IGRP (Interior Gateway Routing Protocol)是一种动态距离向量路由协议，它由Cisco公司八十年代中期设计。使用组合用户配置尺度，包括延迟、带宽、可靠性和负载。 缺省情况下，IGRP每90秒发送一次路由更新广播，在3个更新周期内(即270秒)，没有从路由中的第一个路由器接收到更新，则宣布路由不可访问。在7个更新周期即630秒后，Cisco IOS 软件从路由表中清除路由。

IGRP相关配置命令如下：

* 指定使用IGRP协议 ：router igrp autonomous-system1
* 指定与该路由器相连的网络 ：network network
* 指定与该路由器相邻的节点地址 ：neighbor ip-address
* 同样的，配置好以后使用“show ip route”或者“show ip protocols”命令进行查看。

4.4.3.3 OSPF配置

OSPF(Open Shortest Path First)是一个内部网关协议(Interior Gateway Protocol,简称IGP)，用于在单一自治系统(autonomous system,AS)内决策路由。与RIP相对，OSPF是链路状态路有协议，而RIP是距离向量路由协议。

链路是路由器接口的另一种说法，因此OSPF也称为接口状态路由协议。OSPF通过路由器之间通告网络接口的状态来建立链路状态数据库，生成最短路径树，每个OSPF路由器使用这些最短路径构造路由表。

OSPF配置命令如下：

* 指定使用OSPF协议 ：router ospf process-id1
* 指定与该路由器相连的网络 ：network address wildcard-mask area area-id2
* 指定与该路由器相邻的节点地址 ：neighbor ip-address
* 配置OSPF代价：ip ospf cost{代价}
* 修改链路状态通告再传输之间的秒数：ip ospf retransmit- interval {秒数}
* 规定传输链路状态通告时间：ip ospf transmit- delay {秒数}。
* 设置优先类：ip ospf priority {编号}
* 规定呼叫数据包之间的秒数：ip ospf呼叫- interval {秒数}
* 确定某个路由器的呼叫数据包丢失的秒数：ip ospf dead interval {秒数}
* OSPF简单密码鉴别：ip ospf authentication - key {密匙}
* OSPF MD5鉴别：ip ospf mesagedigest - key {密匙标识符} MD5 {密匙}。

4.4.3.4 BGP配置

BGP（Border Gateway Protocol）是运行于 TCP 上的一种自治系统的路由协议。 BGP 是唯一一个用来处理像因特网大小的网络的协议，也是唯一能够妥善处理好不相关路由域间的多路连接的协议。 BGP 构建在 EGP 的经验之上。 BGP 系统的主要功能是和其他的 BGP 系统交换网络可达信息。网络可达信息包括列出的自治系统（AS）的信息。这些信息有效地构造了 AS 互联的拓朴图并由此清除了路由环路，同时在 AS 级别上可实施策略决策。

BGP相关配置命令如下：

* 指定使用IGRP协议 ：router igrp autonomous-system1
* 指定与该路由器相连的网络 ：network network
* 指定与该路由器相邻的节点地址 ：neighbor ip-address remote-as number2
* 查看配置：show ip bgp
* 清除BGP协议：clear ip bgp

4.4.4广域网协议配置

4.4.4.1 HDLC配置

HDLC（High-Level Data Link Control）是点到点串行线路上（同步电路）的帧封装格式，其帧格式与以太网帧格式有很大的差别，HDLC帧没有源MAC地址和目的MAC地址。HDLC是CISCO路由器使用的缺省协议，一台新路由器在未指定封装协议时默认使用HDLC封装。

配置方法如下：

* 在特权模式下输入config t命令进入配置模式。
* 配置连接的WAN接口。如：interface serial 1,进入config-if模式。
* HDLC协议封装。输入encapsulation HDLC，按回车。
* 设置带宽。输入bandwidth 传输速率[kilobits/second]，如56K，则输入bandwidth 56。

4.4.4.2 PPP配置

PPP（Point to Point Protocol）和HDLC一样，PPP也是串行线路上（同步电路或者异步电路）的一种帧封装格式，但是PPP可以提供对多种网络层协议的支持。PPP支持认证、多线路捆绑、回拨、压缩等功能。PPP经过4个过程在一个点到点的链路上建立通信连接：

* 链路的建立和配置协调：通信的发起方发送LCP帧来配置和检测数据链路
* 链路质量检测：在链路已经建立、协调之后进行，这一阶段是可选的
* 网络层协议配置协调：通信的发起方发送NCP帧以选择并配置网络层协议

CHAP（Challenge Handshake Authentication Protocol）和PAP（Password Authentication Protocol）通常被用于在PPP封装的串行线路上提供安全性认证。使用CHAP和PAP认证,每个路由器通过名字来识别，可以防止未经授权的访问。

配置方法如下：

* 特权模式下输入config t命令，进入配置模式。
* 配置WAN接口。输入具体接口号，如interface serial 0，进入config –if模式。
* 封装PPP协议，输入encapsulation PPP。
* 设置带宽，输入bandwidth 带宽。
* Ctrl+Z，结束配置。

4.4.5 NAT配置

NAT（Network Address Translation，网络地址转换），当在专用网内部的一些主机本来已经分配到了本地IP地址（即仅在本专用网内使用的专用地址），但现在又想和因特网上的主机通信（并不需要加密）时，可使用NAT方法。

这种方法需要在专用网连接到因特网的路由器上安装NAT软件。装有NAT软件的路由器叫做NAT路由器，它至少有一个有效的外部全球IP地址。这样，所有使用本地地址的主机在和外界通信时，都要在NAT路由器上将其本地地址转换成全球IP地址，才能和因特网连接。另外，这种通过使用少量的公有IP 地址代表较多的私有IP 地址的方式，将有助于减缓可用的IP地址空间的枯竭。

NAT相关配置命令如下：

* 接口配置命令：ip nat {inside|outside}
* 全局配置命令：ip nat inside source static local-ip global-ip
* 显示当前存在的NAT转换信息：show ip nat translations
* 查看NAT的统计信息：show ip nat statistics
* 显示当前存在的NAT转换的详细信息：show ip nat translations verbose
* 显示出每个被转换的数据包：debug ip nat
* 删除NAT映射表中的所有内容： Clear ip nat translations
* ip nat inside source list access-list-number pool pool-name [overload]：使用该命令定义访问控制列表与NAT内部全局地址池之间的映射
* ip nat outside source list access-list-number pool pool-name [overload]：使用该命令定义访问控制列表与NAT外部局部地址池之间的映射
* ip nat inside destination list access-list-number pool pool-name：使用该命令定义访问控制列表与终端NAT地址池之间的映射

4.5 路由器配置实验

4.5.1静态路由协议的配置

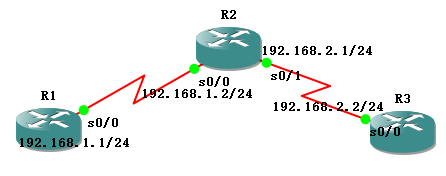


图5.1 静态路由实验拓扑图

R1的配置：

|  |
| --- |
| R1#conf ter  Enter configuration commands, one per line. End with CNTL/Z.  R1(config)#int s0/0  R1(config-if)#ip address 192.168.1.1 255.255.255.0  R1(config-if)#clock rate 56000  R1(config-if)#no shut  R1(config)#ip route 192.168.2.0 255.255.255.0 192.168.1.2 //静态路由配置 |

R2的配置：

|  |
| --- |
| R2(config)#int s0/0  R2(config-if)#ip add 192.168.1.2 255.255.255.0  R2(config-if)#no shut  R2(config-if)#int s0/1  R2(config-if)#ip add 192.168.2.1 255.255.255.0  R2(config-if)#no shut |

R3的配置：

|  |
| --- |
| R3(config)#int s0/0  R3(config-if)#ip address 192.168.2.2 255.255.255.0  R3(config-if)#clock rate 56000  R3(config-if)#no shut  R3(config)#ip route 192.168.1.0 255.255.255.0 192.168.2.1 //静态路由配置 |

此时网络连通，在R3执行ping 192.168.1.1指令能够成功Ping通。

4.5.2路由协议的配置实验

4.5.2.1 RIP配置实验

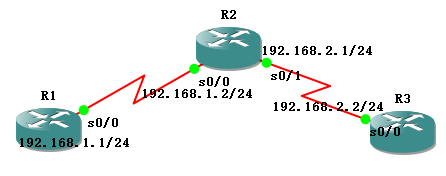


图5.2 RIP实验拓扑图

各端口的ip地址配置如上，此时要连通网络还需要配置RIP动态路由协议。

R1：

|  |
| --- |
| R1(config)#router rip  R1(config-router)#network 192.168.1.0 |

R2：

|  |
| --- |
| R2(config)#router rip  R2(config-router)#network 192.168.1.0  R2(config-router)#network 192.168.2.0 |

R3：

|  |
| --- |
| R3(config)#router rip  R3(config-router)#network 192.168.2.0 |

此时在R1应该能够ping通R3. (执行命令ping 192.168.2.2)

4.5.2.2 IGRP配置实验

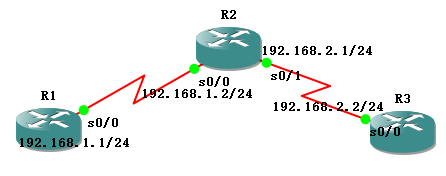


图5.3 IGRP配置实验拓扑图

端口的ip地址按如图5.3所示配置（具体命令参见5.1.1）。

R1:

|  |
| --- |
| R1(config)#router eigrp 55  R1(config-router)#network 192.168.1.0 |

R2:

|  |
| --- |
| R2(config)#router eigrp 55  R2(config-router)#network 192.168.1.0  R2(config-router)#network 192.168.2.0  R2(config-router)#exit  R2(config)#end |

R3:

|  |
| --- |
| R3(config)#router eigrp 55  R3(config-router)#network 192.168.2.0 |

此时在R1应该能够ping通R3. (执行命令ping 192.168.2.2)

4.5.2.3 OSPF配置实验

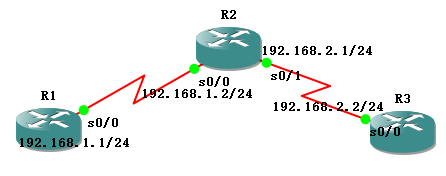


图5.4 OSPF实验拓扑图

各端口的ip地址配置如图5.4所示。（具体的ip地址配置命令参见5.1.1）

如图5.4是三个路由器的简单拓扑图，我们在进行OSPF实验的时候可以把这三个路由器都放在同一个区域内，就可以做单区域的OSPF配置实验。

R1：

|  |
| --- |
| R1#configure terminal  Enter configuration commands, one per line. End with CNTL/Z.  R1(config)#router ospf 1  R1(config-router)#network 192.168.1.0 0.0.0.255 area 0  R1(config-router)#exit  R1(config)#end |

R2：

|  |
| --- |
| R2#configure terminal  Enter configuration commands, one per line. End with CNTL/Z.  R2(config)#router ospf 1  R2(config-router)#network 192.168.1.0 0.0.0.255 area 0  R2(config-router)#network 192.168.2.0 0.0.0.255 area 0  R2(config-router)#exit  R2(config)#end |

R3：

|  |
| --- |
| R3#configure terminal  Enter configuration commands, one per line. End with CNTL/Z.  R3(config)#router ospf 1  R3(config-router)#network 192.168.2.0 0.0.0.255 area 0  R3(config-router)#exit  R3(config)#end |

之后，我们再在R3上加入一条外部路由，命令如下：

|  |
| --- |
| R3#configure terminal  Enter configuration commands, one per line. End with CNTL/Z.  R3(config)#ip route 199.9.9.0 255.255.255.0 null0  R3 (config)#router ospf 1  R3 (config-router)#redistribute static  R3(config)#end |

加入外部路由以后可以使用“show ip route”查看路由表。在OSPF实验中，我们还可以进行区域间路由归纳与外部路由归纳，命令如下：

在R1上进行回环地址的配置，可以配置三个回环地址，再进行区域间路由归纳。

|  |
| --- |
| R1 (config)#int lo0  R1 (config-if)#ip address 172.16.1.1 255.255.255.0  R1 (config-if)#exit  R1 (config)#int lo1  R1 (config-if)#ip address 172.16.2.1 255.255.255.0  R1 (config-if)#exit  R1 (config)#exit  R1 (config)#int lo2  R1 (config-if)#ip address 172.16.3.1 255.255.255.0  R1 (config-if)#end  R1 (config)#router ospf 1  R1 (config-router)#area 1 range 172.16.0.0 255.255.252.0  R1 (config-router)#exit |

按照上面命令所配置，R1的三个回环地址就可以归纳为172.16.0.0/24了。而对于外部路由归纳，我们可以按照下面命令配置：

|  |
| --- |
| R3 (config)#router ospf 1  R3 (config-router)#summary-address 199.9.0.0 255.255.0.0  R3 (config-router)#end |

4.5.2.4 BGP配置实验

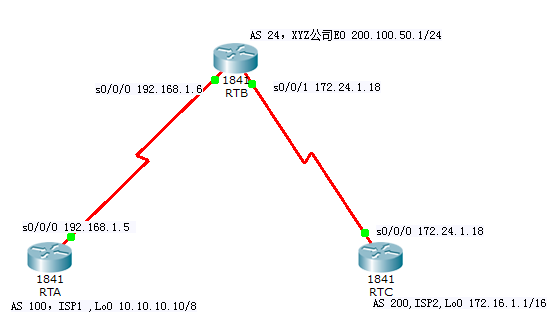


图5.7 BGP配置实验拓扑图

如图所示RTA和RTC是连接外部Internet的ISP路由器，RTB是XYZ公司的边路网关路由器，首先为RTA和RTC这两台ISP路由器配置环回接口IP地址。这些环回接口将用于模拟通过ISP可以达到的真实网络。

|  |
| --- |
| RTA (config)#int lo0  RTA (config-if)#ip address 10.10.10.10 255.0.0.0  RTA (config-if)#exit  RTB(config)#int lo0  RTB (config-if)#ip address 172.16.1.1 255.255.0.0  RTB (config-if)#exit |

配置ISP路由器：

|  |
| --- |
| RTA (config)#router bgp 100  RTA (config-router)#neighbor 192.168.1.6 remote-as 24  RTA (config-router)#network 10.0.0.0  RTC (config)#router bgp 24  RTC (config-router)#neighbor 172.24.1.17 remote-as 24  RTC (config-router)#network 172.16.0.0 |

配置路由器RTB与两家服务供应商运行BGP协议：

|  |
| --- |
| RTB (config)#router bgp 24  RTB (config-router)#neighbor 192.168.1.5 remote-as 100  RTB (config-router)#network 172.24.1.18 remote-as 200  RTB (config-router)#network 200.100.50.0 |

4.5.3广域网协议(PPP和HDLC) 配置实验

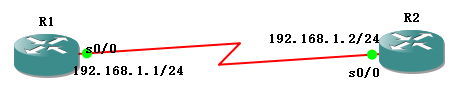


图5.5 PPP,HDLC实验的配置实验

按图5.5所示配置好各端口的ip地址。（具体的ip地址配置命令参见5.1.1）

此时思科路由器使用默认的hdlc协议，可以用show int s0/0查看。

|  |
| --- |
| R1#show int s0/0  Serial0/0 is up, line protocol is up  Hardware is GT96K Serial  Internet address is 192.168.1.1/24  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,  reliability 255/255, txload 1/255, rxload 1/255  Encapsulation HDLC, loopback not set  Keepalive set (10 sec)  CRC checking enabled  Last input 00:00:02, output 00:00:04, output hang never  Last clearing of "show interface" counters never  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0  Queueing strategy: weighted fair  Output queue: 0/1000/64/0 (size/max total/threshold/drops)  Conversations 0/1/256 (active/max active/max total)  Reserved Conversations 0/0 (allocated/max allocated)  Available Bandwidth 1158 kilobits/sec  5 minute input rate 0 bits/sec, 0 packets/sec  5 minute output rate 0 bits/sec, 0 packets/sec  1878 packets input, 121397 bytes, 0 no buffer  Received 901 broadcasts, 0 runts, 0 giants, 0 throttles  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort  1715 packets output, 111742 bytes, 0 underruns  0 output errors, 0 collisions, 14 interface resets  0 output buffer failures, 0 output buffers swapped out  0 carrier transitions  DCD=up DSR=up DTR=up RTS=up CTS=up |

若要改为PPP协议，可以用encapsulation ppp指令进行修改。（注意，此处链路两边需要同步修改）

|  |
| --- |
| R1#conf ter  Enter configuration commands, one per line. End with CNTL/Z.  R1(config)#int s0/0  R1(config-if)#encapsulation ppp  R2# conf ter  Enter configuration commands, one per line. End with CNTL/Z.  R2(config)#int s0/0  R2(config-if)#encapsulation ppp |

此时，链路联通，且链路层协议改为PPP协议，可用show int s0/0进行查看确认。

|  |
| --- |
| R1#show int s0/0  Serial0/0 is up, line protocol is up  Hardware is GT96K Serial  Internet address is 192.168.1.1/24  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,  reliability 255/255, txload 1/255, rxload 1/255  Encapsulation PPP, LCP Open  Open: IPCP, CDPCP, loopback not set  Keepalive set (10 sec)  CRC checking enabled  Last input 00:00:01, output 00:00:01, output hang never  Last clearing of "show interface" counters 00:06:40  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0  Queueing strategy: weighted fair  Output queue: 0/1000/64/0 (size/max total/threshold/drops)  Conversations 0/2/256 (active/max active/max total)  Reserved Conversations 0/0 (allocated/max allocated)  Available Bandwidth 1158 kilobits/sec  5 minute input rate 0 bits/sec, 0 packets/sec  5 minute output rate 0 bits/sec, 0 packets/sec  167 packets input, 9284 bytes, 0 no buffer  Received 0 broadcasts, 0 runts, 0 giants, 0 throttles  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort  173 packets output, 8938 bytes, 0 underruns  0 output errors, 0 collisions, 5 interface resets  0 output buffer failures, 0 output buffers swapped out  0 carrier transitions  DCD=up DSR=up DTR=up RTS=up CTS=up |

如要改回HDLC协议，可输入类似的命令修改：

|  |
| --- |
| R1#conf ter  Enter configuration commands, one per line. End with CNTL/Z.  R1(config)#int s0/0  R1(config-if)#encapsulation hdlc  R2# conf ter  Enter configuration commands, one per line. End with CNTL/Z.  R2(config)#int s0/0  R2(config-if)#encapsulation hdlc |

4.5.4 NAT实验配置

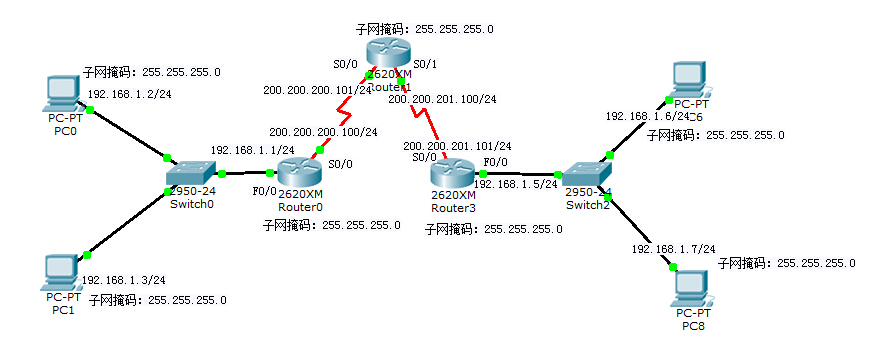


图5.6NAT实验拓扑图

在Router0及Router3处配置Nat，替换192.168.1.0处的地址。各处地址配置如图。

使用RIP动态路由，配置如下：

Router0:

|  |
| --- |
| router rip  network 200.200.200.0 |

Router1:

|  |
| --- |
| router rip  network 200.200.200.0  network 200.200.201.0 |

Router3:

|  |
| --- |
| router rip  network 200.200.201.0 |

4.5.4.1静态NAT配置

在Router0处配置NAT

|  |
| --- |
| Router>enable  Router#conf t  Enter configuration commands, one per line. End with CNTL/Z.  Router(config)#ip nat inside source static 192.168.1.2 200.200.200.120  Router(config)#ip nat inside source static 192.168.1.3 200.200.200.121  Router(config)#int s0/0  Router(config-if)#ip nat outside  Router(config-if)#int f0/0  Router(config-if)#ip nat inside  Router(config-if)#end |

输入show ip nat translations命令查看目前的NAT表

|  |
| --- |
| Router#show ip nat translations  Pro Inside global Inside local Outside local Outside global  --- 200.200.200.120 192.168.1.2 --- ---  --- 200.200.200.121 192.168.1.3 --- --- |

上图可知，内部网络地址已经成功转为可用的外部网络地址。但是外部网络的目的地址还没有，是因为此时还没有ping任何一个外部网络，故NAT转发表还未建立。

使用pc0 ping 路由器1(Route1)

|  |
| --- |
| Router#show ip nat translations  Pro Inside global Inside local Outside local Outside global  icmp 200.200.200.120:10192.168.1.2:10 200.200.201.100:10 200.200.201.100:10  icmp 200.200.200.120:11192.168.1.2:11 200.200.201.100:11 200.200.201.100:11  icmp 200.200.200.120:12192.168.1.2:12 200.200.201.100:12 200.200.201.100:12  icmp 200.200.200.120:9 192.168.1.2:9 200.200.201.100:9 200.200.201.100:9  --- 200.200.200.120 192.168.1.2 --- ---  --- 200.200.200.121 192.168.1.3 --- --- |

由上图可知，在pc0上192.168.1.2已经成功建立了NAT转发表，pc1上192.168.1.3因为没有ping任何外部网络，所以它的转发表为空。

在Router3处配置NAT

类似于Router0的配置。

4.5.4.2动态NAT配置

Router0：

|  |
| --- |
| Router0(config)#int f0/0  Router0(config-if)#ip address 192.168.1.1 255.255.255.0  Router0(config-if)#no shut  Router0(config-if)#ip nat inside  Router0(config-if)#int s0/0  Router0(config-if)#ip address 200.200.200.100 255.255.255.0  Router0(config-if)#clock 56000  Router0(config-if)#no shut  Router0(config-if)#ip nat outside  Router0(config-if)#exit  Router0(config)#access-list 1 permit 192.168.1.0 0.0.0.255 (取所有192.168.1.0段的值)  Router0(config)#ip nat inside source list 1 int s0/0 overload |

此时Router0处的Nat配置已完成，从PC0(PC1)能ping通Router1和Router3.

Router3执行类似配置：

|  |
| --- |
| Router3(config)#int f0/0  Router3(config-if)#ip address 192.168.1.5 255.255.255.0  Router3(config-if)#no shut  Router3(config-if)#ip nat inside  Router3(config-if)#int s0/0  Router3(config-if)#ip address 200.200.201.101 255.255.255.0  Router3(config-if)#clock 56000  Router3(config-if)#no shut  Router3(config-if)#ip nat outside  Router3(config-if)#exit  Router3(config)#access-list 1 permit 192.168.1.0 0.0.0.255 (取所有192.168.1.0段的值)  Router3(config)#ip nat inside source list 1 int s0/0 overload |

此时Router3处的Nat配置已完成，从PC6(PC8)能ping通Router0和Router1.