本文第四章从实现的角度出发，对基于名字的 CDN 路由系统进行整体设计。首先，给出了路由系统的整体结构，对其中的主模块进行功能说明；然后，我们介绍了路由主进程和与其交互的系统消息，分析了涉及的路由模块；我们分别给出了 INRP 和 NBRP 的功能框图。为了便于理解，我们还给出了 INRP 和 NBRP 的主要报文处理流程图，以及 INRP 和 NBRP 交互的流程图。

最后给出了结论和展望，简要总结了本文的主要工作，指出了 CDN 的名字路由系统有待于进一步深入研究的几个方面。

关键词： 内容传送网络；基于名字路由；域名解析协议；基于名字的路由协议；邻居网关协议；宽带接入技术。

Research on Distributed Route Technologies for the CDN Networks
—— Research on Name-based Route Technologies

ABSTRACT

Despite the alluring promise of the Internet as the convergence channel for delivery of all types of media, it is obvious that there are many challenges that must be addressed to ensure consistent delivery to the end-user. New types of networks are being built to specifically address these needs – Content Delivery Network (CDN). The research subject of the dissertation is the key function layer of the CDN network - —Route layer. From the point of the theoretical analysis and the computer simulation, the Name-based route technology of the CDN route system is thoroughly studied.

Chapter 1 of the dissertation is an introduction. The background and system architecture of the CDN network are given. The significance of the investigation and the main topics of the dissertation are described. The skeleton of the paper is also introduced.

In Chapter 2, the main function units of the INRP protocol for the CDN routing system based on content name are summarized, then several units are deeply studied. First, the user access flow of INRP protocol is discussed. Second, the frame architecture of INRP is introduced. Third, packet process is the kernel process of INRP and the tag which differs from DNS protocol, the dissertation has a deeply discussion here. Forth, the table-driven route algorithm is discussed, the longest-suffix match mode is proposed. Fifth, lookup latency is one large portion of total INRP route time, several lookup algorithms are analyzed, a few performance simulation results are also given. One improved INRP route simulation algorithm is proposed and the results are given.

The Name-base Routing Protocol performs dynamic routing on content names. It is responsible for route table 's creation and updating. Chapter 3 covers NBRP protocol related problems. First, the properties of NBRP are introduced, the function of NBRP in route table 's creation and updating is also introduced. Second, the frame architecture and FSM of NBRP are discussed. Third, the dissertation shows route theory of NBRP, the NBRP route process based on route theory and protocol is detailed discussed. Forth, how to improve the NBRP performance is our research object, the route stability and security are analyzed.

In Chapter 4, we discuss the system design of name-based routing system. First, the system architecture is given. Second, the main routing task is discussed and we list the system messages. Third, the function diagrams of NBRP and INRP are also given. We also give the sequence diagram of the main protocol message.

Finally, the conclusion and epilogue are given. The main work of the dissertation is concluded. The fields of the CDN route system, which need to be studied deeply, are mentioned.

KEY WORDS: Content Delivery Network, Internet Name Resolution Protocol, Name-based Routing Protocol, Domain Name System, Border Gateway Protocol.

第一章 绪 论

§1.1 CDN 网络背景技术介绍

1.1.1 CDN 的出现

遍及全球的互联网正在使人类的信息传递发生革命性的变化。然而，网络信息传递性能的日益恶化却日渐成为互联网发展的一大瓶颈。

网站访问响应速度取决于许多因素，如网络的带宽瓶颈、传输途中的路由阻塞和延迟、网站服务器的处理能力等。多数情况下，网站响应速度和访问者与网站服务器之间的距离有密切的关系。尽管中国电信计划将骨干网络提速 8 倍，并且增加带宽，但是如果访问者和网站之间的距离较远，它们之间的通信一样需要经过重重的路由转发和处理，网络延误不可避免。

在一片改善互联网性能的呼声中，CDN（内容分发网络）概念应运而生，并在短短几年内迅速发展。有别于传统的互联网性能改善方案（如增加接入带宽，升级软硬件和建立多个镜象站点），CDN 提出“让内容离用户更近”的全新思路。

1.1.2 CDN 的概念介绍

CDN 全名为“内容分发网络”（Content Delivery Network），它通过实现用户对网站服务器的就近访问及网络流量智能分流，从技术上全面解决由于网络带宽小、用户访问量大、网点分布不均等对用户访问效果的影响，大大提高网络的响应速度。

CDN 的技术原理是在现有的互联网络中建立一个完善的中间层，将网站的内容发布到最接近用户的网络“边缘”，使用户能以最快的速度、从最接近用户的地方获得所需的信息。由于这种技术极大地缓解了互联网的拥塞情况，所以网站有能力提供更多类似视频节目、歌曲点播等数据流量巨大的内容服务，在线交易、网上银行等多种业务的可靠性也得到了有效保障。CDN 的网络层次图如下图 1.1 所示。

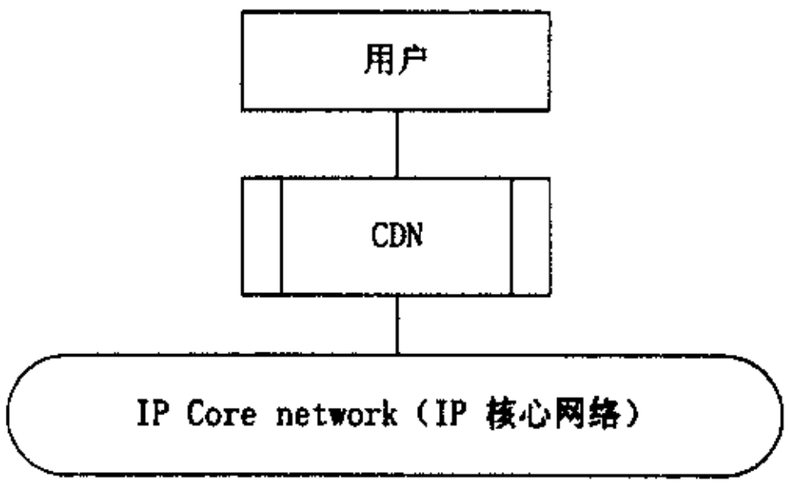


图 1.1 CDN 网络层次图

通常的内容分发方法包括配置内容服务器节点于遍布世界的主要网际网络接入点，和利用一个特殊路由机制重定向一个内容请求到最接近的内容服务器。例如，当 Web 用户点击一个可以内容分发的 URL，内容分发网络实时的根据网络流量和各节点的连接、负载状况

第一章 绪 论

§1.1 CDN 网络背景技术介绍

1.1.1 CDN 的出现

遍及全球的互联网正在使人类的信息传递发生革命性的变化。然而，网络信息传递性能的日益恶化却日渐成为互联网发展的一大瓶颈。

网站访问响应速度取决于许多因素，如网络的带宽瓶颈、传输途中的路由阻塞和延迟、网站服务器的处理能力等。多数情况下，网站响应速度和访问者与网站服务器之间的距离有密切的关系。尽管中国电信计划将骨干网络提速 8 倍，并且增加带宽，但是如果访问者和网站之间的距离较远，它们之间的通信一样需要经过重重的路由转发和处理，网络延误不可避免。

在一片改善互联网性能的呼声中，CDN（内容分发网络）概念应运而生，并在短短几年内迅速发展。有别于传统的互联网性能改善方案（如增加接入带宽，升级软硬件和建立多个镜象站点），CDN 提出“让内容离用户更近”的全新思路。

1.1.2 CDN 的概念介绍

CDN 全名为“内容分发网络”（Content Delivery Network），它通过实现用户对网站服务器的就近访问及网络流量智能分流，从技术上全面解决由于网络带宽小、用户访问量大、网点分布不均等对用户访问效果的影响，大大提高网络的响应速度。

CDN 的技术原理是在现有的互联网络中建立一个完善的中间层，将网站的内容发布到最接近用户的网络“边缘”，使用户能以最快的速度、从最接近用户的地方获得所需的信息。由于这种技术极大地缓解了互联网的拥塞情况，所以网站有能力提供更多类似视频节目、歌曲点播等数据流量巨大的内容服务，在线交易、网上银行等多种业务的可靠性也得到了有效保障。CDN 的网络层次图如下图 1.1 所示。

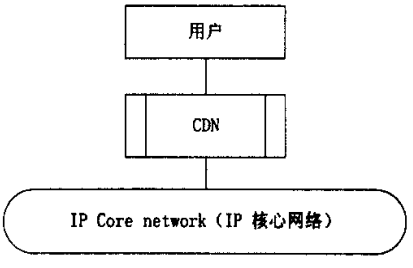


图 1.1 CDN 网络层次图

通常的内容分发方法包括配置内容服务器节点于遍布世界的主要网际网络接入点，和利用一个特殊路由机制重定向一个内容请求到最接近的内容服务器。例如，当 Web 用户点击一个可以内容分发的 URL，内容分发网络实时的根据网络流量和各节点的连接、负载状况

以及到用户的距离和响应时间等信息，将用户请求重定向到离用户最近的内容服务器，大大提高了网络的访问速度。

1.1.3 CDN 的系统架构

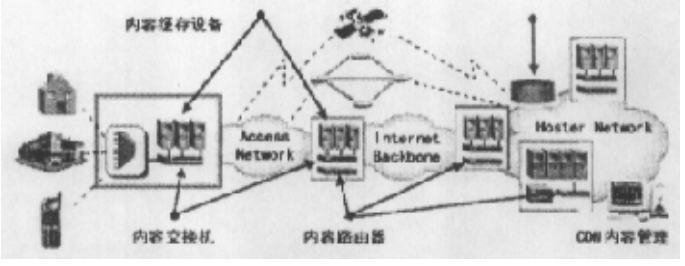


图 1.2 典型的 CDN 网络

典型的 CDN 网络架构如图 1.2 所示。CDN 网络由内容服务器（内容缓存设备）、内容交换机、内容路由器、内容分发系统和内容管理五部分组成。它们的具体作用如下表 1.1 所示。

内容服务器（内容缓存设备）	位于接入点或多点 IDC 中，可缓存静态的 Web 内容和流媒体内容
内容交互机	对内容缓存负载平衡及访问控制
内容路由器	负载为用户的请求选择最佳的访问站点
内容分发系统	内容缓存只可缓存静态内容。而对于一些要求对其所有分布站点的内容作镜像的用户,我们需要一个内容镜像系统,这就是内容分发系统
CDN 管理系统	由于 CDN 系统是提供给多个用户共享,所以需要一个中心管理系统作为 CDN 的集中管理

表 1.1 CDN 构成和功能

一套完整的 CDN 系统包括服务器负载均衡、动态内容路由、高速缓存机制、动态内容分发与复制、网络安全机制等多项技术。本文主要研究的是 CDN 的动态路由技术。

1.1.5 CDN 的发展

CDN 是下一代网络（NGN）的重要组成部分。目前，CDN 技术获得了蓬勃的发展。CDN 是研究的一个热点，国家“十五”863 基金已经对 CDN 立项，浙江大学通信与信息系统研究所已经申请到了该课题项目，对该课题展开研究。

目前，国内已经有了 CDN 的初步应用。从 2000 年 8 月取得信息产业部的"关于同意开展互联网内容快递业务实验的批复"（信电[2000] 146 号）后，CDN 运营商北京蓝讯公司与中国电信各省的电信公司以及国内主要的网络运营商合作，在各主要城市设立 ChinaCache 节点，并迅速在全国范围内提供了 ChinaCache CDN 加速服务，节点数量不断增加，截止到 2001 年 3 月 31 日，CDN 共完成了 13 个节点的安装工作。详细情况见下表 1.2。

以及到用户的距离和响应时间等信息，将用户请求重定向到离用户最近的内容服务器，大大提高了网络的访问速度。

1.1.3 CDN 的系统架构

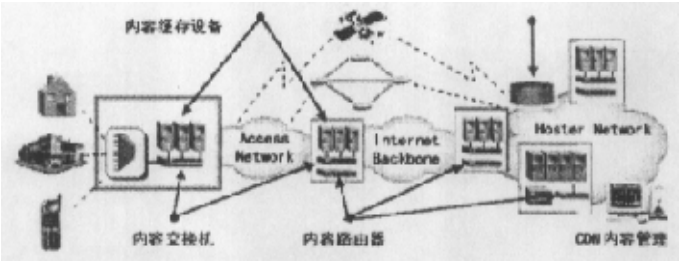


图 1.2 典型的 CDN 网络

典型的 CDN 网络架构如图 1.2 所示。CDN 网络由内容服务器（内容缓存设备）、内容交换机、内容路由器、内容分发系统和内容管理五部分组成。它们的具体作用如下表 1.1 所示。

内容服务器（内容缓存设备）	位于接入点或多点 IDC 中，可缓存静态的 Web 内容和流媒体内容
内容交互机	对内容缓存负载平衡及访问控制
内容路由器	负载为用户的请求选择最佳的访问站点
内容分发系统	内容缓存只可缓存静态内容。而对于一些要求对其所有分布站点的内容作镜像的用户,我们需要一个内容镜像系统,这就是内容分发系统
CDN 管理系统	由于 CDN 系统是提供给多个用户共享,所以需要一个中心管理系统作为 CDN 的集中管理

表 1.1 CDN 构成和功能

一套完整的 CDN 系统包括服务器负载均衡、动态内容路由、高速缓存机制、动态内容分发与复制、网络安全机制等多项技术。本文主要研究的是 CDN 的动态路由技术。

1.1.5 CDN 的发展

CDN 是下一代网络（NGN）的重要组成部分。目前，CDN 技术获得了蓬勃的发展。CDN 是研究的一个热点，国家“十五”863 基金已经对 CDN 立项，浙江大学通信与信息系统研究所已经申请到了该课题项目，对该课题展开研究。

目前，国内已经有了 CDN 的初步应用。从 2000 年 8 月取得信息产业部的"关于同意开展互联网内容快递业务实验的批复"（信电[2000] 146 号）后，CDN 运营商北京蓝讯公司与中国电信各省的电信公司以及国内主要的网络运营商合作，在各主要城市设立 ChinaCache 节点，并迅速在全国范围内提供了 ChinaCache CDN 加速服务，节点数量不断增加，截止到 2001 年 3 月 31 日，CDN 共完成了 13 个节点的安装工作。详细情况见下表 1.2。

以及到用户的距离和响应时间等信息，将用户请求重定向到离用户最近的内容服务器，大大提高了网络的访问速度。

1.1.3 CDN 的系统架构

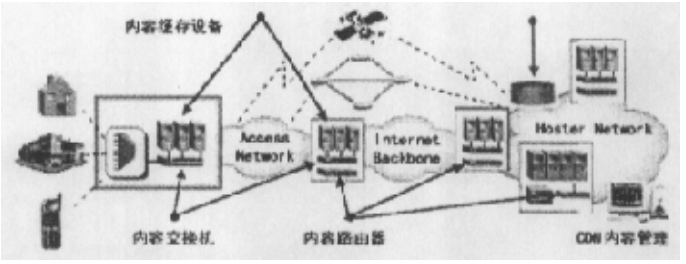


图 1.2 典型的 CDN 网络

典型的 CDN 网络架构如图 1.2 所示。CDN 网络由内容服务器（内容缓存设备）、内容交换机、内容路由器、内容分发系统和内容管理五部分组成。它们的具体作用如下表 1.1 所示。

内容服务器（内容缓存设备）	位于接入点或多点 IDC 中，可缓存静态的 Web 内容和流媒体内容
内容交互机	对内容缓存负载平衡及访问控制
内容路由器	负载为用户的请求选择最佳的访问站点
内容分发系统	内容缓存只可缓存静态内容。而对于一些要求对其所有分布站点的内容作镜像的用户,我们需要一个内容镜像系统,这就是内容分发系统
CDN 管理系统	由于 CDN 系统是提供给多个用户共享,所以需要一个中心管理系统作为 CDN 的集中管理

表 1.1 CDN 构成和功能

一套完整的 CDN 系统包括服务器负载均衡、动态内容路由、高速缓存机制、动态内容分发与复制、网络安全机制等多项技术。本文主要研究的是 CDN 的动态路由技术。

1.1.5 CDN 的发展

CDN 是下一代网络（NGN）的重要组成部分。目前，CDN 技术获得了蓬勃的发展。CDN 是研究的一个热点，国家“十五”863 基金已经对 CDN 立项，浙江大学通信与信息系统研究所已经申请到了该课题项目，对该课题展开研究。

目前，国内已经有了 CDN 的初步应用。从 2000 年 8 月取得信息产业部的"关于同意开展互联网内容快递业务实验的批复"（信电[2000] 146 号）后，CDN 运营商北京蓝讯公司与中国电信各省的电信公司以及国内主要的网络运营商合作，在各主要城市设立 ChinaCache 节点，并迅速在全国范围内提供了 ChinaCache CDN 加速服务，节点数量不断增加，截止到 2001 年 3 月 31 日，CDN 共完成了 13 个节点的安装工作。详细情况见下表 1.2。

	节点名称	ISP 网络	覆盖区域
1	广州	China NET	广州大区
2	上海	China NET	上海大区
3	天津	China NET	北京大区
4	西安	China NET	西安大区
5	南京	China NET	南京大区
6	杭州	China NET	上海大区
7	长春	China NET	沈阳大区
8	成都	China NET	成都大区
9	重庆	China NET	成都大区
10	北京	GBNET	吉通
11	北邮	CERNET	教育网
12	清华万博	CERNET	教育网
13	上海	Shanghai Online	上海大区

表 1.2 CDN 节点分布表

各城市中的 ChinaCache 节点直接与 ChinaNET 的省骨干路由器或相应的交换机相连，主要为本省和同一大区的 Internet 用户服务。从上表可以看出，目前 CDN 的节点分布在全国十个主要城市，基本覆盖了 163 骨干网 8 大区中 7 个。

1.1.5 与 CDN 相关的国际组织

CDN 已经提出就受到工业界和学术界的广泛重视，一些国际研究组织纷纷成立，正在制定 CDN 的相关标准。

Content Allieance 由 Cisco 等 62 家公司与 2000 年 8 月成立，正在对 CDN 网络互联技术进行标准化。

Content Bridge Alliance 由 Adero Inc 和 Inktomi Corp 等 10 家公司与 2000 年 8 月成立，该联盟目前正在通过建立一个小规模的多供应商的 CDN 服务网络，以决定那些技术需要进行标准化。

IETF CDN WorkGroup 该工作组的主要任务是制定 CDN 的相关规范。

斯坦福大学 DSG 研究组 DSG（Distributed systems Group）研究组的重要对象是 TRIAD（Translating Relaying Internet Architecture integrating Active Directories）技术。CDN 是其 TRIAD 的重要一员。目前，基于名字的路由（Name_Based Routing）是该研究组的研究重点。

§1.2 CDN 路由

1.2.1 CDN 路由的概念

CDN 路由又称为“内容路由”，指的是把用户的内容请求重定向，或者说，“路由”到离用户最近的内容服务器。

	节点名称	ISP 网络	覆盖区域
1	广外	China NET	广州大区
2	上海	China NET	上海大区
3	天津	China NET	北京大区
4	西安	China NET	西安大区
5	南京	China NET	南京大区
6	杭州	China NET	上海大区
7	长春	China NET	沈阳大区
8	成都	China NET	成都大区
9	重庆	China NET	成都大区
10	北京	GBNET	吉通
11	北邮	CERNET	教育网
12	清华万博	CERNET	教育网
13	上海	Shanghai Online	上海大区

表 1.2 CDN 节点分布表

各城市中的 ChinaCache 节点直接与 ChinaNET 的省骨干路由器或相应的交换机相连，主要为本省和同一大区的 Internet 用户服务。从上表可以看出，目前 CDN 的节点分布在全国十个主要城市，基本覆盖了 163 骨干网 8 大区中 7 个。

1.1.5 与 CDN 相关的国际组织

CDN 已经提出就受到工业界和学术界的广泛重视，一些国际研究组织纷纷成立，正在制定 CDN 的相关标准。

Content Allieance 由 Cisco 等 62 家公司与 2000 年 8 月成立，正在对 CDN 网络互联技术进行标准化。

Content Bridge Alliance 由 Adero Inc 和 Inktomi Corp 等 10 家公司与 2000 年 8 月成立，该联盟目前正在通过建立一个小规模的多供应商的 CDN 服务网络，以决定那些技术需要进行标准化。

IETF CDN WorkGroup 该工作组的主要任务是制定 CDN 的相关规范。

斯坦福大学 DSG 研究组 DSG（Distributed systems Group）研究组的重要对象是 TRIAD（Translating Relaying Internet Architecture integrating Active Directories）技术。CDN 是其 TRIAD 的重要一员。目前，基于名字的路由（Name_Based Routing）是该研究组的研究重点。

§1.2 CDN 路由

1.2.1 CDN 路由的概念

CDN 路由又称为“内容路由”，指的是把用户的内容请求重定向，或者说，“路由”到离用户最近的内容服务器。

	节点名称	ISP 网络	覆盖区域
1	广外	China NET	广州大区
2	上海	China NET	上海大区
3	天津	China NET	北京大区
4	西安	China NET	西安大区
5	南京	China NET	南京大区
6	杭州	China NET	上海大区
7	长春	China NET	沈阳大区
8	成都	China NET	成都大区
9	重庆	China NET	成都大区
10	北京	GBNET	吉通
11	北邮	CERNET	教育网
12	清华万博	CERNET	教育网
13	上海	Shanghai Online	上海大区

表 1.2 CDN 节点分布表

各城市中的 ChinaCache 节点直接与 ChinaNET 的省骨干路由器或相应的交换机相连，主要为本省和同一大区的 Internet 用户服务。从上表可以看出，目前 CDN 的节点分布在全国十个主要城市，基本覆盖了 163 骨干网 8 大区中 7 个。

1.1.5 与 CDN 相关的国际组织

CDN 已经提出就受到工业界和学术界的广泛重视，一些国际研究组织纷纷成立，正在制定 CDN 的相关标准。

Content Allieance 由 Cisco 等 62 家公司与 2000 年 8 月成立，正在对 CDN 网络互联技术进行标准化。

Content Bridge Alliance 由 Adero Inc 和 Inktomi Corp 等 10 家公司与 2000 年 8 月成立，该联盟目前正在通过建立一个小规模的多供应商的 CDN 服务网络，以决定那些技术需要进行标准化。

IETF CDN WorkGroup 该工作组的主要任务是制定 CDN 的相关规范。

斯坦福大学 DSG 研究组 DSG（Distributed systems Group）研究组的重要对象是 TRIAD（Translating Relaying Internet Architecture integrating Active Directories）技术。CDN 是其 TRIAD 的重要一员。目前，基于名字的路由（Name_Based Routing）是该研究组的研究重点。

§1.2 CDN 路由

1.2.1 CDN 路由的概念

CDN 路由又称为“内容路由”，指的是把用户的内容请求重定向，或者说，“路由”到离用户最近的内容服务器。

CDN 内容路由器负责内容路由，为用户的请求选择最佳的访问站点。

CDN 路由以现有的 IP 网络为基础，支撑点依然是现有 IP 路由机制。图 1.3 表明了 CDN 路由、IP 路由和用户内容请求的层次关系。我们把 CDN 路由所处的层次形象的称为“内容层”。内容层充分利用现有的 IP 核心路由网络，通过特定的内容路由策略，为用户的内容请求选择最佳的内容服务器，完成内容的重定向。

CDN 的路由技术的策略 CDN 路由技术可以分为两类：

◎ 基于 DNS 的路由

◎ 基于名字的路由

基于 DNS 的路由是中心服务器体系的路由方式，基于名字的路由采用分布式体系。

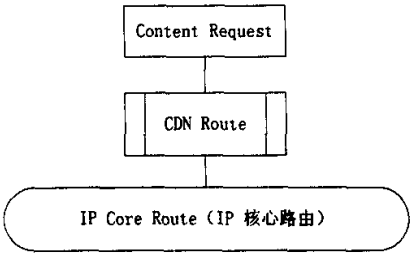


图 1.3 内容路由层次图

1.2.2 基于 DNS 的路由

❶ 网络架构

基于 DNS 的路由机制采用中心服务器体系的网络架构。图 1.5 是整个路由体系的网络架构图。DNS 内容路由主要由重定向 DNS 服务器和内容服务器节点所组成。其中，重定向 DNS 服务器是整个路由机制的核心。

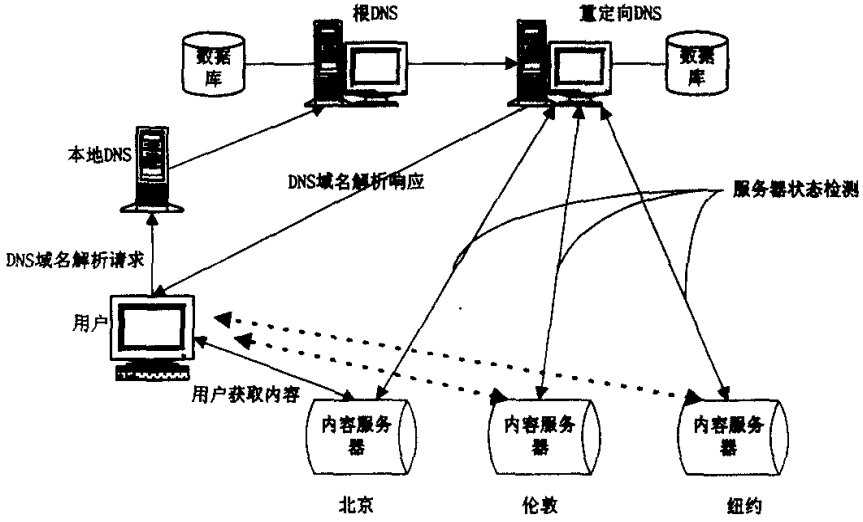


图 1.4 基于 DNS 重定向路由的网络架构

CDN 内容路由器负责内容路由，为用户的请求选择最佳的访问站点。

CDN 路由以现有的 IP 网络为基础，支撑点依然是现有 IP 路由机制。图 1.3 表明了 CDN 路由、IP 路由和用户内容请求的层次关系。我们把 CDN 路由所处的层次形象的称为“内容层”。内容层充分利用现有的 IP 核心路由网络，通过特定的内容路由策略，为用户的内容请求选择最佳的内容服务器，完成内容的重定向。

CDN 的路由技术的策略 CDN 路由技术可以分为两类：

◎ 基于 DNS 的路由

◎ 基于名字的路由

基于 DNS 的路由是中心服务器体系的路由方式，基于名字的路由采用分布式体系。

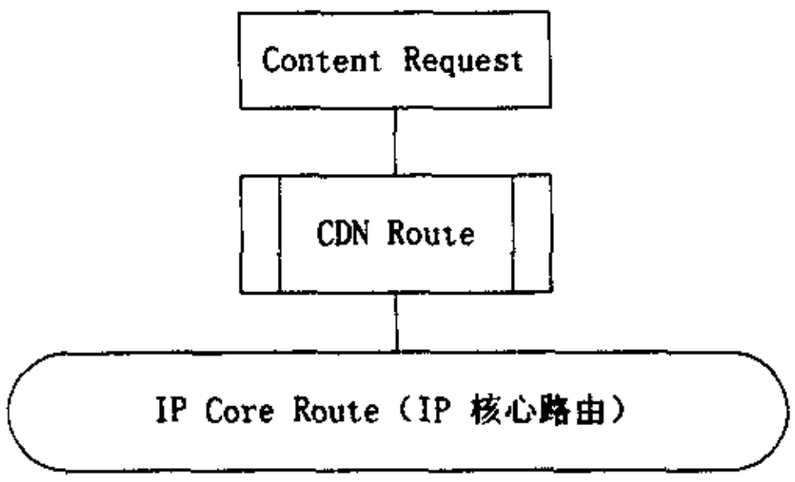


图 1.3 内容路由层次图

1.2.2 基于 DNS 的路由

◎ 网络架构

基于 DNS 的路由机制采用中心服务器体系的网络架构。图 1.5 是整个路由体系的网络架构图。DNS 内容路由主要由重定向 DNS 服务器和内容服务器节点所组成。其中，重定向 DNS 服务器是整个路由机制的核心。

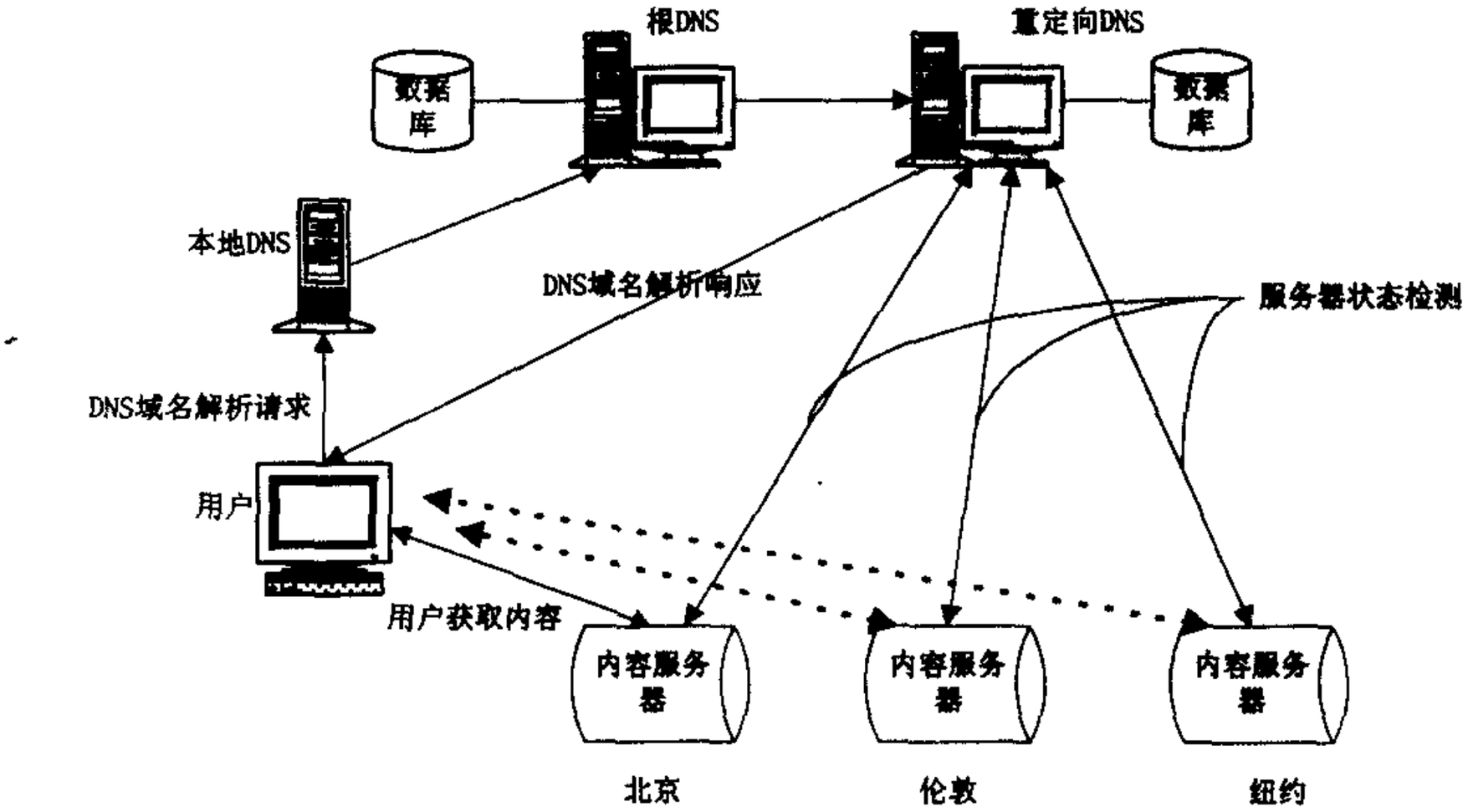


图 1.4 基于 DNS 重定向路由的网络架构

a. 重定向 DNS 服务器

重定向 DNS 是一个在多个地点之间进行网络交通优化、全球负载均衡、访问重定向的内容服务器。它与标准的 DNS 协议相兼容。可以根据网络反应速度，客户的地理分布以及服务器的状态来选择最佳的服务器来回应顾客的 DNS 请求。

b. 动态内容路由

内容路由是重定向服务器最基本也是最重要的功能。当用户访问加入 CDN 服务的网站时，域名解析请求将最终由重定向 DNS 负责处理。通过一组预先定义好的策略（如内容类型、地理区域、网络负载状况等），将当时最“接近”用户的节点地址提供给用户，使用户得到快速的服务。

c. 网络反应速度测量

重定向根据网络反应速度进行流量分配的动态负载均衡。它通过不断与内容服务器组进行通讯来获得网络反应速度。通告内容服务器的支持，可以测出在同一网络中的内容服务器与客户机之间的网络性能。根据网络反应速度等因素可以选择出对客户机回应最快的内容服务器

d. 地理位置就近访问

收到一个 DNS 域名解析请求时，重定向 DNS 将会把客户机的 IP 地址与它的数据库中地理位置数据库进行匹配，识别出客户机所处地区，并将客户机访问重定向到同一地区的内容服务器。就近访问将大大减少网络访问的延迟

e. 服务器的状态检测

重定向不停地监控所有服务器的工作负荷与健康状态。这使得重定向 DNS 能将客户的访问请求重定向到负荷最轻的服务器上，而且不会将客户的访问请求导向到不能正常工作的服务器上。从而提高的客户的网络访问响应速度，同时也提高了服务质量。

○ 工作原理

当用户访问已经加入 CDN 服务的网站时，首先通过 DNS 重定向确定最接近用户的 CDN 内容服务器节点，同时将用户的请求指向该节点。当用户的请求到达指定节点时，CDN 的内容服务器负责将用户请求的内容提供给用户。 下图 1.5 阐述了用户访问流程。

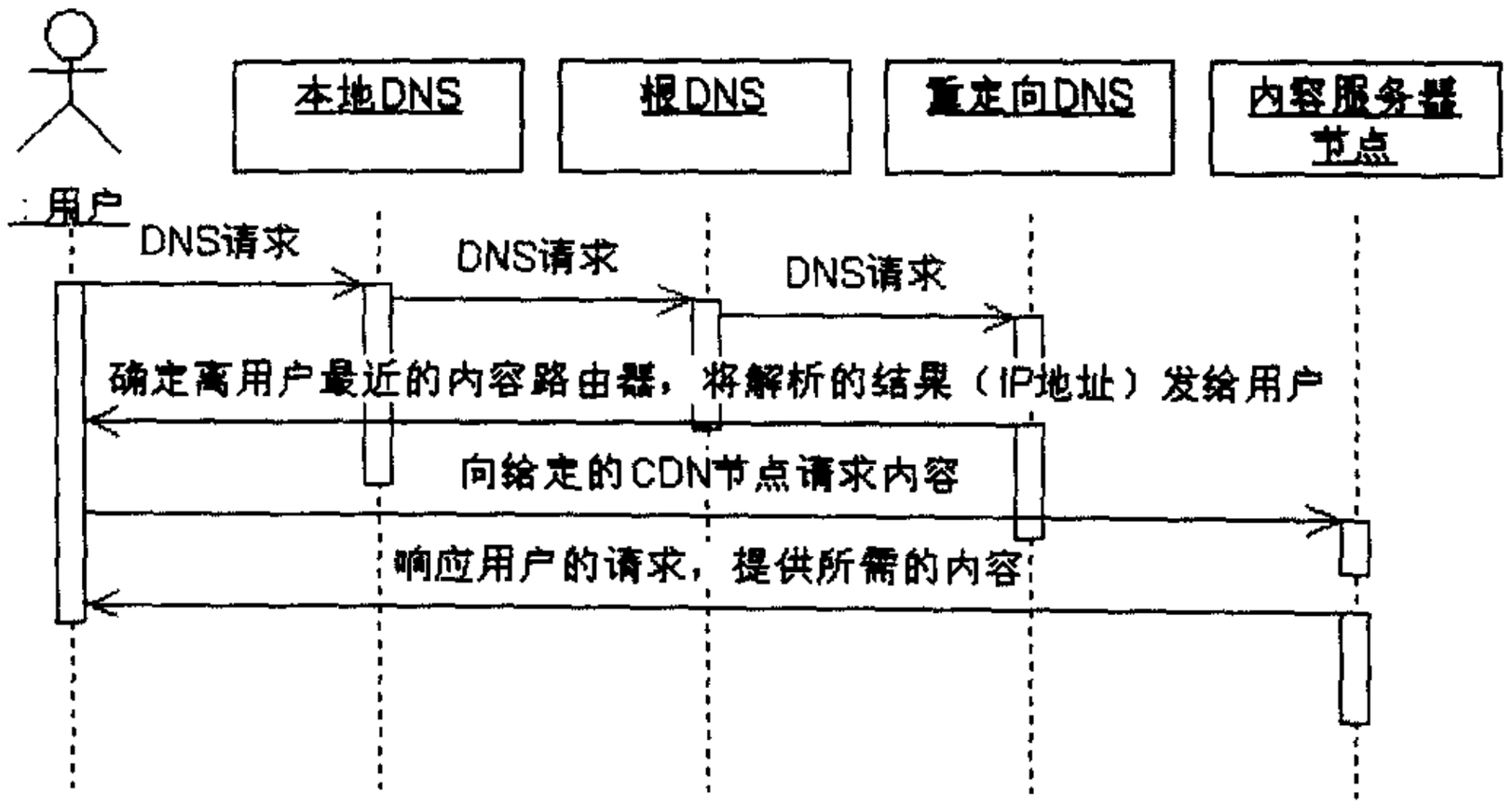


图 1.5　用户访问流程

由用户访问流程，我们推算出用户获取内容的 RTT(Round-trip Time)公式（式 1.1）：

获取内容的 RTT = DNS 重定向的 RTT ＋ 内容服务器提供内容的 RTT（式 1.1）；

其中

DNS 重定向的 RTT ＝ 用户 DNS 请求到达重定向 DNS 的时间耗费＋重定向 DNS 选择内容服务器的时间耗费＋DNS 响应返回用户的时间耗费；（式 1.2）

DNS 重定向的 RTT 又称为内容路由的 RTT。

❍ DNS 重定向的缺陷

这一种路由解决方案，由充分利用了现有的网络结构，结构简单，因而被人们认为是一种有效的技术，是现阶段 CDN 应用中主要采用的路由技术。但是，它的性能瓶颈非常突出。

a. DNS 重定向的 RTT 远大于内容服务器提供内容的 RTT。需要花较长的往返时间去定位离用户非常近的内容服务器。重定向的过程包括了访问远端根 DNS、重定向 DNS 服务器的过程（图 1.5）。相反，最后访问的内容服务器离用户却很近。显而易见，DNS 重定向耗费的 RTT 远大于用户访问内容服务器的 RTT（见图 1.5）。RTT 成了 CDN 性能提高的瓶颈之一。

b. 当我们提高网络带宽，缩短了用户请求到达根 DNS、重定向服务器的时间。DNS 海量数据的查询、重定向 DNS 复杂的定位选择过程的延迟，无形中造成了网络性能提高的又一瓶颈。

c. CDN 是分布式的，而重定向 DNS 技术却是中心服务器式的。由于对根 DNS 和重定向 DNS 的集中访问形成了网络拥塞点，而 CDN 的设计目标之一就是避免网络拥塞点，这就形成了矛盾。对服务器的集中访问无疑加重了服务器的负载，进一步导致了数据查询和处理延迟。另外，由于采用中心服务器方式，重定向 DNS 技术存在不可靠性。如果通往中心 DNS 的链路严重拥塞甚至断开，用户就不能访问与它近邻的内容服务器。

1.2.3 基于名字的 CDN 路由

重定向 DNS 的缺陷促使人们去开发新的内容路由技术。斯坦福大学提出了基于名字的

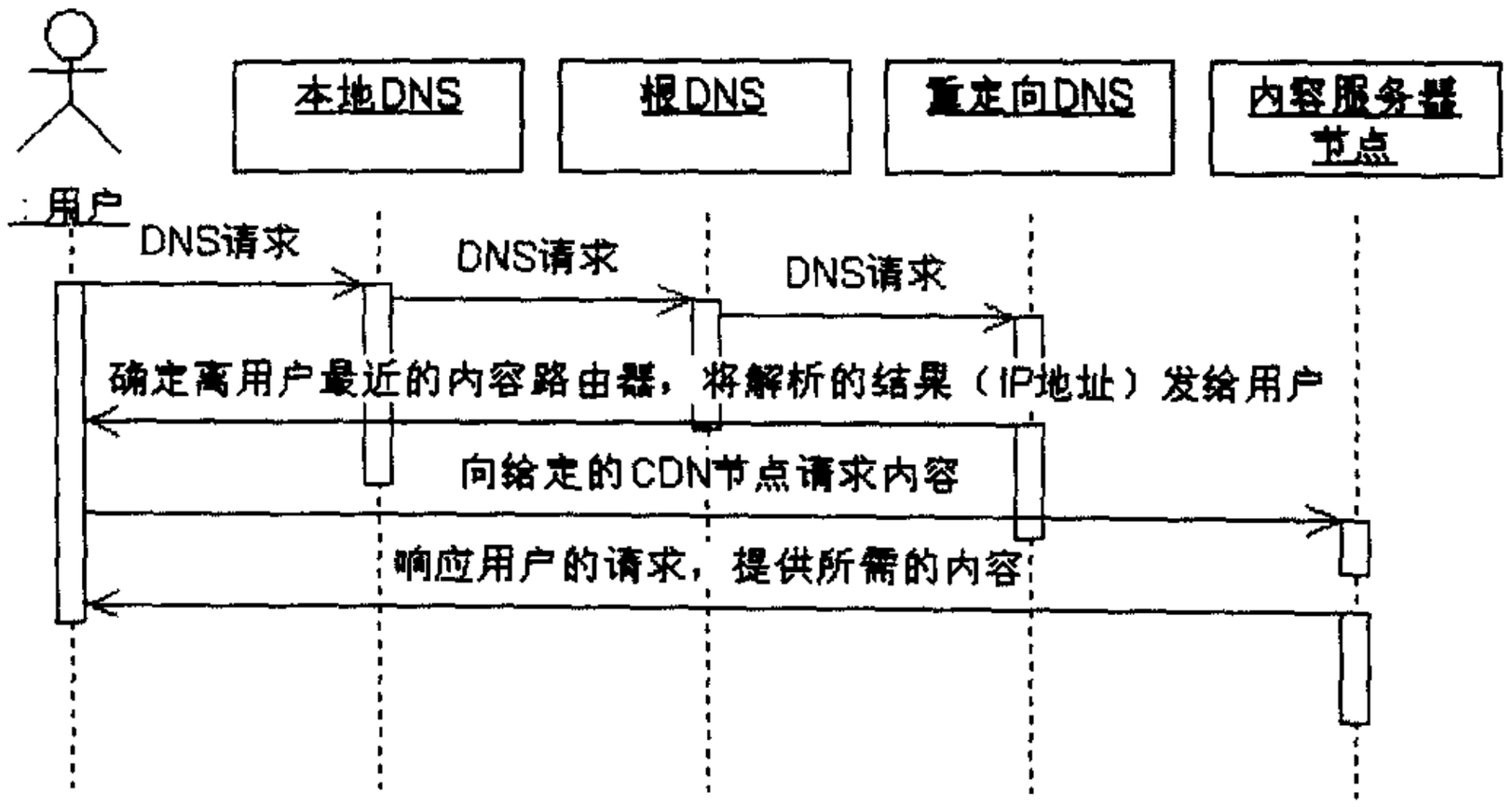


图 1.5　用户访问流程

由用户访问流程，我们推算出用户获取内容的 RTT(Round-trip Time)公式（式 1.1）：

获取内容的 RTT = DNS 重定向的 RTT ＋ 内容服务器提供内容的 RTT（式 1.1）；

其中

DNS 重定向的 RTT ＝ 用户 DNS 请求到达重定向 DNS 的时间耗费＋重定向 DNS 选择内容服务器的时间耗费＋DNS 响应返回用户的时间耗费；（式 1.2）

DNS 重定向的 RTT 又称为内容路由的 RTT。

❍ DNS 重定向的缺陷

这一种路由解决方案，由充分利用了现有的网络结构，结构简单，因而被人们认为是一种有效的技术，是现阶段 CDN 应用中主要采用的路由技术。但是，它的性能瓶颈非常突出。

a. DNS 重定向的 RTT 远大于内容服务器提供内容的 RTT。需要花较长的往返时间去定位离用户非常近的内容服务器。重定向的过程包括了访问远端根 DNS、重定向 DNS 服务器的过程（图 1.5）。相反，最后访问的内容服务器离用户却很近。显而易见，DNS 重定向耗费的 RTT 远大于用户访问内容服务器的 RTT（见图 1.5）。RTT 成了 CDN 性能提高的瓶颈之一。

b. 当我们提高网络带宽，缩短了用户请求到达根 DNS、重定向服务器的时间。DNS 海量数据的查询、重定向 DNS 复杂的定位选择过程的延迟，无形中造成了网络性能提高的又一瓶颈。

c. CDN 是分布式的，而重定向 DNS 技术却是中心服务器式的。由于对根 DNS 和重定向 DNS 的集中访问形成了网络拥塞点，而 CDN 的设计目标之一就是避免网络拥塞点，这就形成了矛盾。对服务器的集中访问无疑加重了服务器的负载，进一步导致了数据查询和处理延迟。另外，由于采用中心服务器方式，重定向 DNS 技术存在不可靠性。如果通往中心 DNS 的链路严重拥塞甚至断开，用户就不能访问与它近邻的内容服务器。

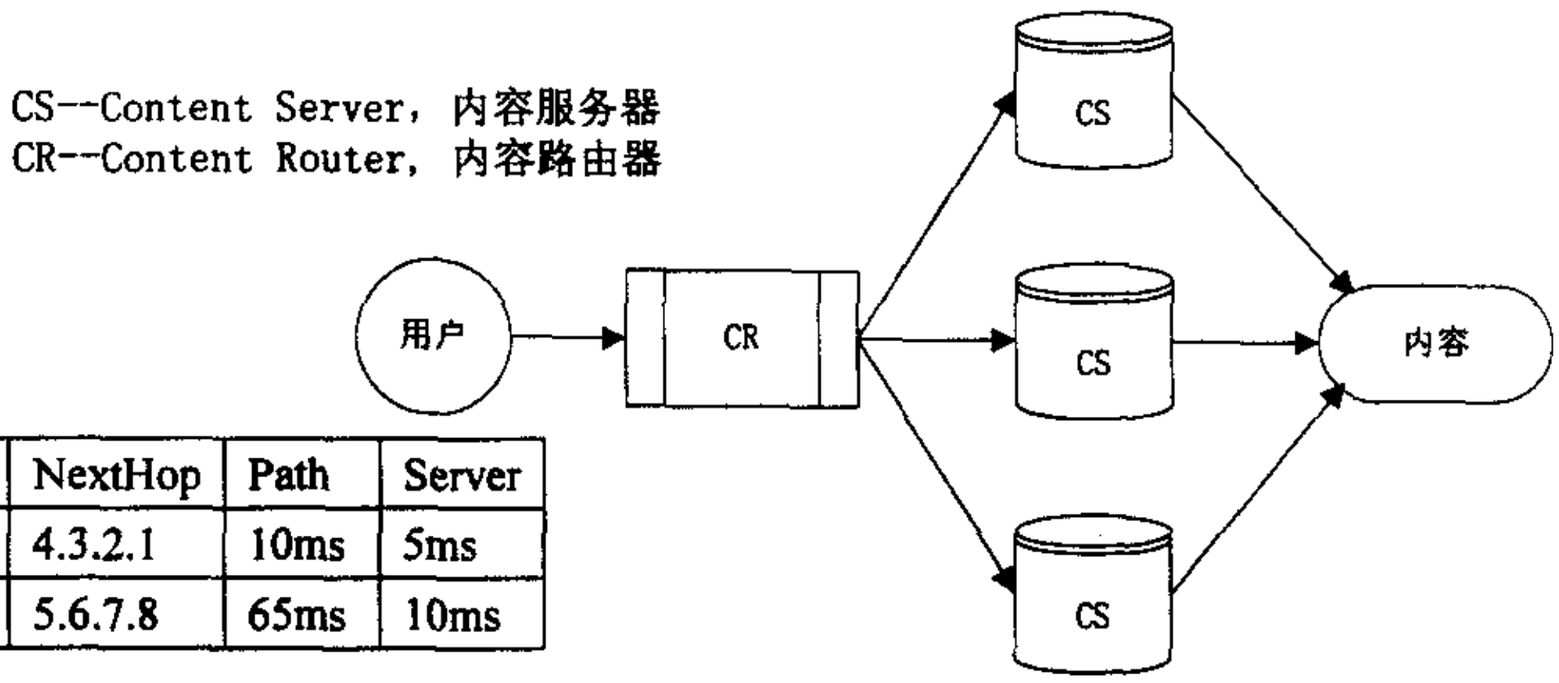
1.2.3 基于名字的 CDN 路由

重定向 DNS 的缺陷促使人们去开发新的内容路由技术。斯坦福大学提出了基于名字的

路由（Name_Based Routing）思想。它不同于传统的基于 IP 地址的路由思想，而是基于用户需求的内容的名字（通常是一个 URL）进行路由。基于名字的路由技术引起了人们的广泛注意，开始对它展开进一步的研究。无论在国内和国外，它都是一项崭新的技术。浙江大学的 CDN 实验室项目前期的一个重要工作之一就是基于名字的路由技术的研究。本文就是围绕名字路由而展开的。

※ 名字路由的思想

按照斯坦福的描述，在下一代网络里，用户想要连接的不是一个特定的服务器或者一个 IP 地址，而是以名字（例如 URL）标志的内容，如图 1.6。



Name	NextHop	Path	Server
Cnn.com	4.3.2.1	10ms	5ms
Cnn.com	5.6.7.8	65ms	10ms

表 1.3　CR 路由表

图 1.6　斯坦福基于名字路由思想

IP 地址在名字路由的模式下，已经不再标志一个网络终端。取而代之的是含义具体的象 URL 这样的内容名字。因特网上传播的路由信息除了 BGP 的 IP 地址前缀表示的路由信息，就是以名字后缀表示的内容路由信息。在这里，CS 已经不是一个单纯的负责内容分发的服务器，它被看成特殊的 CR。用户可以通过提供相同内容服务的不同 CS 获取同样的内容（图 1.6）。这让人容易联想到 IP 路由的多条路由路径的选择问题。

根据名字路由的思想，斯坦福公布了 CR 路由表设计（见表 1.3）。Name 一般代表 CS 的域名。NextHop 表示的是下一个对等体 CR（包括 CS）的 IP 地址。Path 和 Server 两项集中代表了从当前 CR 到达目的内容的时间耗费，见下面的公式（式 1.3）：

到达内容的时间耗费 ＝ 到达 CS 的时间耗费 ＋ CS 查询内容的时间耗费；（式 1.3）

其中 Path 项代表了到达 CS 的时间耗费，Server 项表示了 CS 查询内容的时间耗费。CR 根据最长后缀匹配原则，筛选出与用户请求的内容名一致的路由项，然后根据最小的时间耗费（Path + Server）选择下一个对等体 CR（或者 CS）。

※ 两个协议

支撑名字路由体系的是两个核心协议：INRP、NBRP。它们组成了内容路由层。

1. INRP(Internet Name Resolution Protocol)

INRP 与 DNS 协议兼容，INRP 的报文格式与 DNS 报文相同，但是有不同的含义。INRP 的作用：

a. 根据用户的内容名进行路由表的最长后缀匹配搜索，将用户的内容请求路由到最佳的 CS。内容路由的同时为用户下一次的流内容获取建立了 IP 路由链路。

b. 在响应超时等错误情况下，重新为用户请求选择另一个 CR 或 CS。

c.建立 CR 和用户、CR 和 CR（CS）的连接。

本文第二章将围绕 INRP 展开讨论。

2. NBRP(Name-Based Routing Protocol)

NBRP 可以看成是 CDN 网络中的“BGP”。它与 BGP 非常相似。与 BGP 一样，NBRP 的着眼点是选择最好的路由并控制路由的传播。最大的不同，BGP 基于 IP 地址，而 NBRP 基于名字。NBRP 的作用：

a. 传播内容路由信息，包括 CR、CS 的添加和删除等路由信息

b. 当网络拓扑发生变化时，进行路由表动态更新。

这一部分将在本文的第三章具体阐述。

※ 名字路由体系中的关键元素——CR 和 CS

基于名字路由的体系主要由 CS 和 CR 构成。下图 1.7 代表了未来的名字路由网络拓扑。

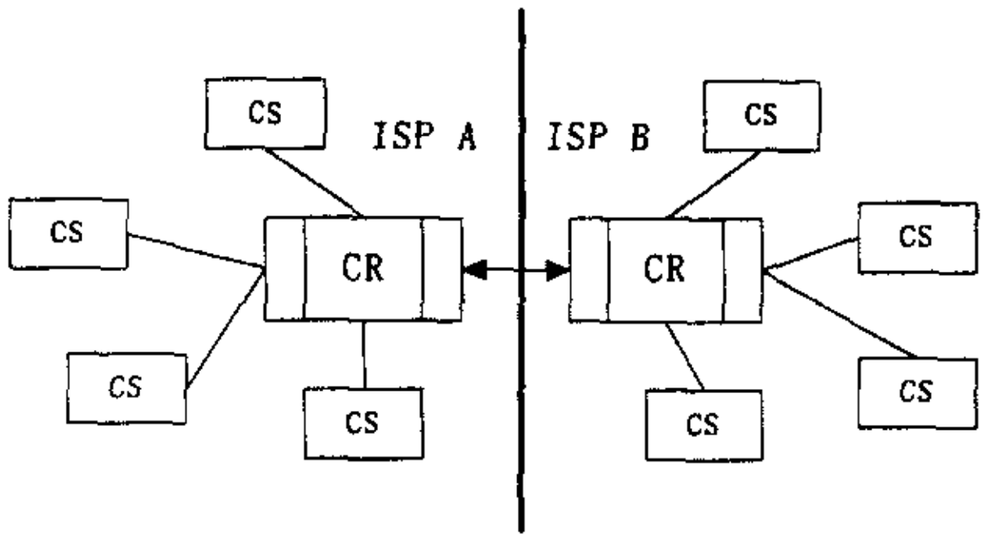


图 1.7　名字路由的网络拓扑

CR 的功能包括：

a. 参加名字路由体系的动态路由更新过程（NBRP）；

b.负载为用户的内容请求选择最佳的 CS，或者将内容请求前递到邻近的 CR。从这个意义上说，CR 可以被看成一个分布式的 DNS SERVER。

如图 1.7 描述，CS 已经不是一个单纯的负责内容分发的服务器，它的功能包括：

a. 提供 web server 等内容分发功能；

b. 响应用户内容请求（DNS/INRP），用户的内容请求最终通过 CR 到达 CS；

c. 通过 CR 通告 CS 的内容信息。

特殊的，如果 CR 包括了现有的 IP 路由功能，那么就可以用 CR 代替现在的 IP 路由器。我们用图例（图 1.8）形象的表示 CR 和 CS 的功能。

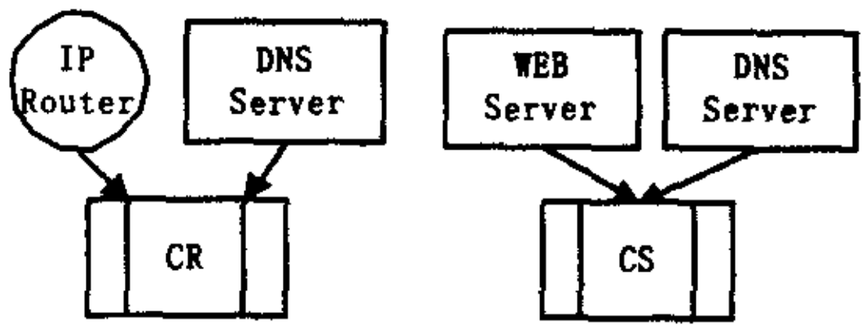


图1.8 CR和CS的功能图

※ 用户访问案例

我们用一个用户访问案例来阐述名字路由的工作原理和流程。案例分成三个部分。

1 CS 路由通告

浙江杭州存在一 CDN 节点 CS，域名“zju.edu.cn”。CS 以 NBRP 的形式向它邻近的 CR 通告。CR 通过 NBRP 协议，将 CS(zju.edu.cn)的内容可达信息通知 CR 的邻居对等体。如图 1.9 所示。

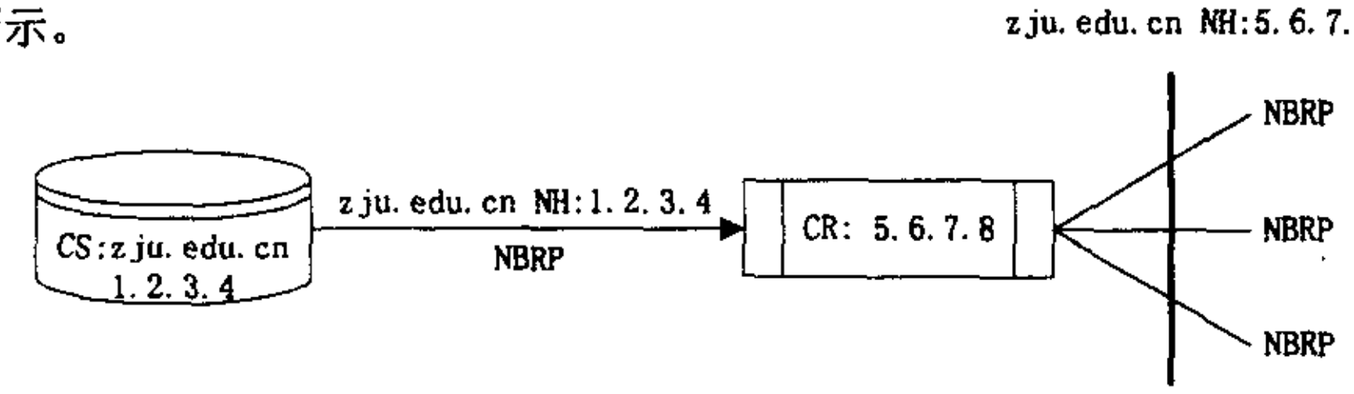


图 1.9 CS 路由通告

2 内容请求的路由

杭州一个用户敲入 URL:http://grs.zju.cn 时，该内容请求（DNS 请求）到达用户配置的 CR。INRP 通过路由表查询将该 URL 路由到离用户最近的节点 CS(zju.edu.cn, 1.2.3.4)。CS 响应内容请求。INRP 响应沿 CR 原路返回。

图 1.10 中 CR 与 CR 的虚线表示内容请求可能通过多个 CR 到达目的 CS。

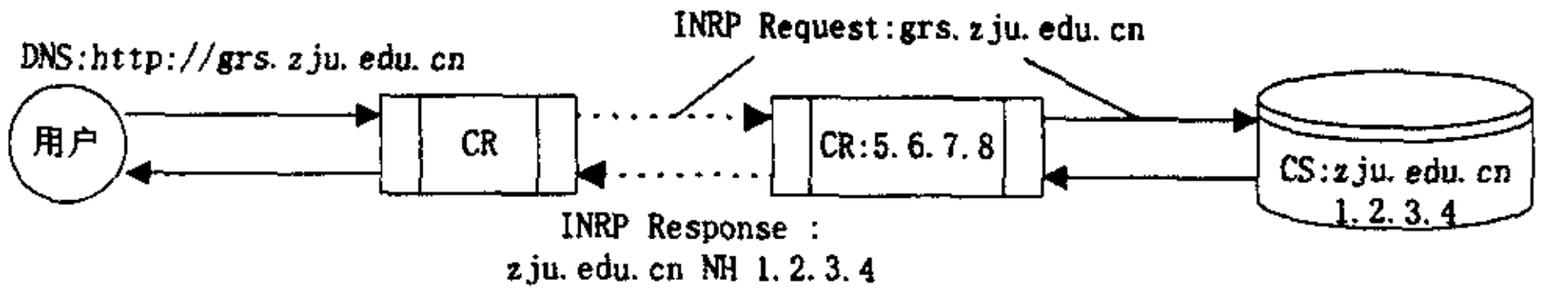


图 1.10　　内容路由

3 内容获取

用户根据 DNS（INRP）响应的 CS 的 IP 地址，向目的 CS 发起 HTTP 等请求，获取内容。这一部分与 DNS 路由的用户内容获取一样，充分利用了现有的 IP 资源。

4 RTT 的推导

由以上用户的访问流程，我们推导出用户获取内容的 RTT(Round-trip Time)公式（式 1.4）：

获取内容的 RTT = 内容路由的 RTT ＋ 内容服务器提供内容的 RTT（式 1.4）；

其中

内容路由的 RTT ＝到达 CS 的时间耗费 ＋ CS 查询内容的时间耗费＋CS 响应返回用户的时间耗费；（式 1.5）

将名字路由的 RTT 公式（式 1.4）与 DNS 重定向的 RTT(式 1.1)相比较，可以看出它们几乎是一致的。内容服务器提供内容的 RTT 不依赖于 CDN 路由方式，两种情况下都是相同的。最大的区别在于内容路由的 RTT 的不同（见式 1.2 和式 1.5）。由于基于名字路由采用了分布式的体系结构，内容路由的耗费是由用户附近的 CR 和 CS 所引起的，地理位置的优势和查询数据量的大大缩小决定了内容路由的 RTT 将原小于 DNS 重定向的 RTT。因此，名字路由获取内容的 RTT 将远远小于 DNS 路由的获取内容的 RTT。

基于名字路由的路由体系实际上是一个分布式的 DNS 体系，借鉴了 IP 路由的思想。它是根据 CDN 网络分布式的特点而设计的，避免了 DNS 路由的不可靠性。

§1.3 本文的主要贡献及篇章结构

本文的主要工作是围绕 CDN 的分布式路由——基于名字的路由具体展开的，主要的研

浙江杭州存在一 CDN 节点 CS，域名“zju.edu.cn”。CS 以 NBRP 的形式向它邻近的 CR 通告。CR 通过 NBRP 协议，将 CS(zju.edu.cn)的内容可达信息通知 CR 的邻居对等体。如图 1.9 所示。

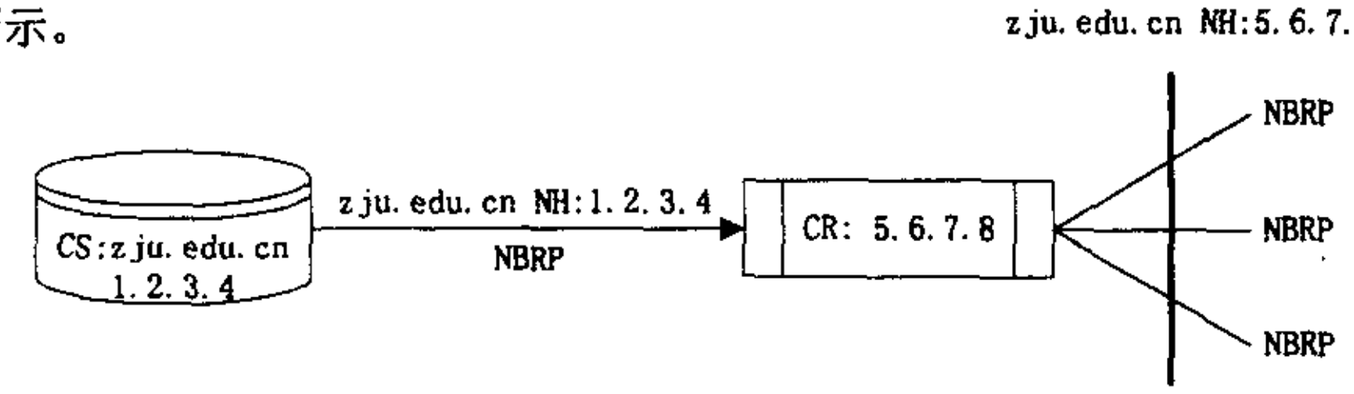


图 1.9 CS 路由通告

2 内容请求的路由

杭州一个用户敲入 URL:http://grs.zju.cn 时，该内容请求（DNS 请求）到达用户配置的 CR。INRP 通过路由表查询将该 URL 路由到离用户最近的节点 CS(zju.edu.cn, 1.2.3.4)。CS 响应内容请求。INRP 响应沿 CR 原路返回。

图 1.10 中 CR 与 CR 的虚线表示内容请求可能通过多个 CR 到达目的 CS。

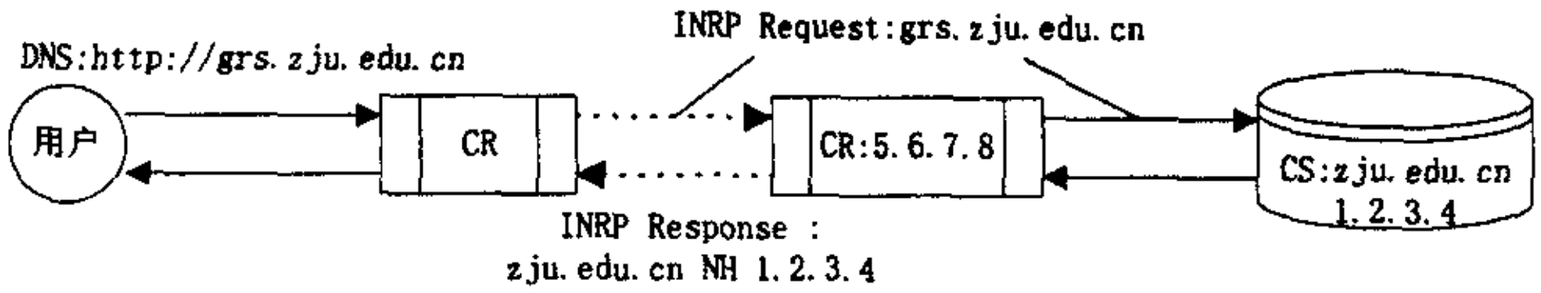


图 1.10　　内容路由

3 内容获取

用户根据 DNS（INRP）响应的 CS 的 IP 地址，向目的 CS 发起 HTTP 等请求，获取内容。这一部分与 DNS 路由的用户内容获取一样，充分利用了现有的 IP 资源。

4 RTT 的推导

由以上用户的访问流程，我们推导出用户获取内容的 RTT(Round-trip Time)公式（式 1.4）：

获取内容的 RTT = 内容路由的 RTT ＋ 内容服务器提供内容的 RTT（式 1.4）；

其中

内容路由的 RTT ＝到达 CS 的时间耗费 ＋ CS 查询内容的时间耗费＋CS 响应返回用户的时间耗费；（式 1.5）

将名字路由的 RTT 公式（式 1.4）与 DNS 重定向的 RTT(式 1.1)相比较，可以看出它们几乎是一致的。内容服务器提供内容的 RTT 不依赖于 CDN 路由方式，两种情况下都是相同的。最大的区别在于内容路由的 RTT 的不同（见式 1.2 和式 1.5）。由于基于名字路由采用了分布式的体系结构，内容路由的耗费是由用户附近的 CR 和 CS 所引起的，地理位置的优势和查询数据量的大大缩小决定了内容路由的 RTT 将原小于 DNS 重定向的 RTT。因此，名字路由获取内容的 RTT 将远远小于 DNS 路由的获取内容的 RTT。

基于名字路由的路由体系实际上是一个分布式的 DNS 体系，借鉴了 IP 路由的思想。它是根据 CDN 网络分布式的特点而设计的，避免了 DNS 路由的不可靠性。

§1.3 本文的主要贡献及篇章结构

本文的主要工作是围绕 CDN 的分布式路由——基于名字的路由具体展开的，主要的研

究对象是名字路由的发现、计算、传播、更新等方法。本文详细讨论了 INRP 和 NBRP 两个名字路由的核心协议，包括算法仿真、性能改进、系统设计等方面。

研究的意义：

从确立研究课题至今，本人认为本课题的研究意义在于以下几个方面：

※ CDN 是下一代网络（NGN）的重要组成部分，是当前网络通信领域研究和开发的一个重点。CDN 的目标提高网络的访问速度。一般的，充分利用现有的网络基础设施，对它进行升级以提供高速的宽带接入是比较合理的近期目标，因此选择 CDN 是合理的和符合中国的国情。

※ 国内外现阶段的 CDN 应用都是基于 DNS 的中心服务器方式。它有一些缺陷。本文研究的基于名字的路由可以解决基于 DNS 路由的固有缺陷。因此对基于名字的路由体系和功能模块的研究，有利于提高下一代网络的性能，并具有实际的应用价值。

※ 基于名字的路由解决了基于 DNS 的一些缺陷，它是新一代的 CDN 路由方式。

虽然学术界已经有人提出了基于名字路由的 CDN 路由，工业界也已经有公司致力于开发这一种新一代的路由系统，但是这种路由系统的很多方面都没有进行深入的分析和研究。即使是首先提出名字路由的斯坦福大学 DSG 研究组，对名字路由也是处于研究进展中。本着这个目的，本文对基于名字的 CDN 路由进行了比较深入的研究。

作者所作的主要工作：

近一年的时间里，作者专心致志的围绕 CDN 的路由技术，尤其是基于名字的路由技术进行学习和研究开发，主要完成的工作有：

※ 在广泛收集文献资料的基础上，利用掌握的基础理论知识和专业知识，认真学习并逐步分析、确立基于名字的路由各个功能模块和工作原理，利用各种渠道跟踪该领域研究的国内外动态。

※ 消化基于名字路由技术的相关文献，对基于名字路由的路由发现、计算、传播、更新等几个模块做了较深入的研究。这几个模块在现阶段并没有约定的方法和协议，因此对它们进行研究有利于提高系统性能。

※ 以理论分析和实际开发相结合的方法，对基于名字的路由技术，主要是在 INRP 和 NBRP 的两个主要模块进行深入研究，并取得一定程度的研究成果。

本文的主要贡献：

INRP 协议设计分析及模块设计：

1 讨论了 INRP 报文的结构，并与 DNS 报文做了比较

2 路由表结构的设计，分析与斯坦福大学的异同。

3　路由表搜索算法的最优设计，并进行性能比较
4　内容路由通道的建立，与基于 DNS 的路由比较
5　差错控制，主要是超时情况下路由的重新选择问题
6　基于 INRP 的路由仿真算法的设计以及对系统性能的仿真
6　协议的系统设计

NBRP 协议设计分析及模块设计：

1　分析 NBRP 报文的结构特点，并与 BGP 报文作了比较
2　NBRP 的状态机
3　NBRP 的路由原理和三个判决更新过程
4　路由环路和收敛的讨论
5　NBRP 安全扩展
6　NBRP 路由稳定性能讨论
5　协议的系统设计

本文的篇章结构：

第一章简单介绍了 CDN 网络出现的背景和它的发展前景，然后阐述了 CDN 的两种路由技术的研究概括。

第二章研究基于名字路由的 INRP 协议。首先介绍了基于名字路由的基本实现，对 INRP 进行了功能分析。讨论了报文的基本结构。针对 CDN 对路由表进行了设计，并对路由搜索算法进行了最优化设计。然后阐述了基于名字路由情况下内容路由通道的建立。提出了差错控制方法。最后提出了基于 INRP 的路由仿真算法，对系统性能进行仿真。

第三章研究 NBRP 协议。首先介绍了路由建立和更新原理，以及 NBRP 在其中的作用。给出了 NBRP 的基本报文结构、状态机。对 NBRP 的路由原理和三个判决更新过程进行了详细讨论。然后讨论了 NBRP 的安全性能和路由稳定性能。

第四章从实现的角度出发，讨论整个路由系统的设计。给出了系统架构以及各个主要模块。讨论了 INRP 和 NBRP 的功能框图，用一个流程图说明了它们的协同工作机制。同时还给出了两个路由协议的主要报文的流程图，

最后的结论和展望部分给出了全文的主要工作，还给出了有待进一步深入研究的方向。

第二章 基于名字的路由系统的路由机制

§ 2.1 基于名字路由系统的路由机制概括

名字路由分成两个部份。搜索名字路由表确定分组转发路径称为路由机制（Routing Mechanism）；确定如何创建刷新名字路由的一系列规则称为路由策略（Routing Policy）。本章将围绕路由机制展开。

用于名字路由机制的是 INRP（Internet Name Resolution Protocol）协议。INRP 具有与 DNS 相同的帧结构，但是相同的帧字段在 INRP 中具有不同的意义。INRP 的报文处理方法与 DNS 也截然不同。我们先来看一下 INRP 的用户访问流程。它可以分成四个部分。图 2.1 给出了正常情况用户访问的流程示意图。

1 用户把本身的 DNS 配置成包含名字路由模块的路由器 IP 地址。当客户机发起一个流媒体（例如 HTTP）的内容请求时，客户将首先联系名字路由器请求域名解析。这个过程就像客户机联系本地 DNS 服务器一样。用户的域名解析请求包含流媒体服务器的“名字”，典型的如 URL。

2 名字路由器维护着一张名字路由表。名字路由表包含了名字->下一跳(Name->NextHop）的映射关系，就像 IP 路由表包含了 IP 前缀同下一跳的映射关系。当名字路由表收到一个域名解析请求后，它就在名字路由表中寻找匹配的服务器名字，选择最佳的下一跳。名字路由器把请求转发到下一跳。这样，域名解析请求最终将到达最佳的内容服务器。

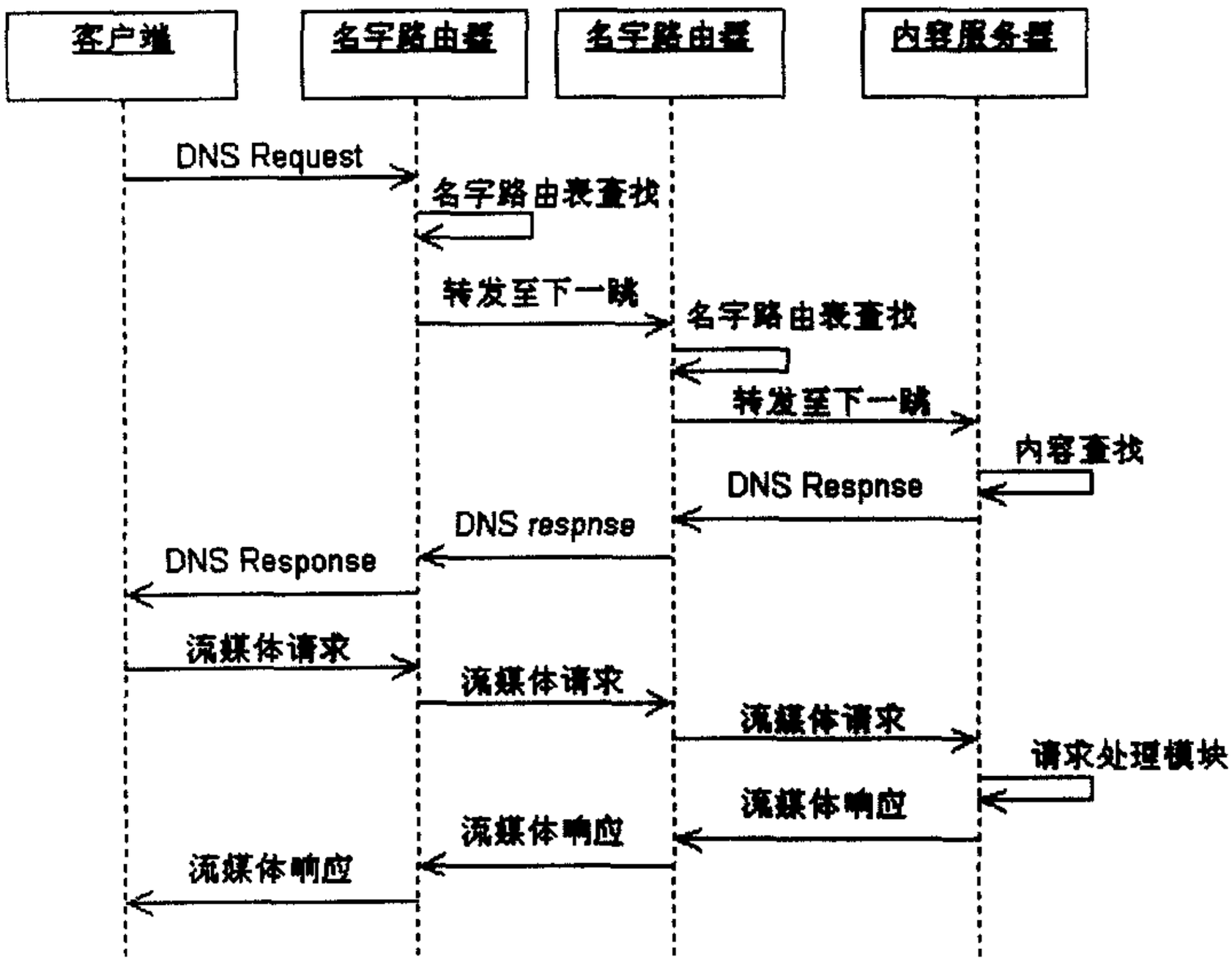


图 2.1 基于 INRP 的用户访问流程示意图

3 内容服务器确认流媒体的内容确实存在，将发出一个响应。响应报文和 DNS 的相应

第二章 基于名字的路由系统的路由机制

§ 2.1 基于名字路由系统的路由机制概括

名字路由分成两个部份。搜索名字路由表确定分组转发路径称为路由机制（Routing Mechanism）；确定如何创建刷新名字路由的一系列规则称为路由策略（Routing Policy）。本章将围绕路由机制展开。

用于名字路由机制的是 INRP（Internet Name Resolution Protocol）协议。INRP 具有与 DNS 相同的帧结构，但是相同的帧字段在 INRP 中具有不同的意义。INRP 的报文处理方法与 DNS 也截然不同。我们先来看一下 INRP 的用户访问流程。它可以分成四个部分。图 2.1 给出了正常情况用户访问的流程示意图。

1 用户把本身的 DNS 配置成包含名字路由模块的路由器 IP 地址。当客户机发起一个流媒体（例如 HTTP）的内容请求时，客户将首先联系名字路由器请求域名解析。这个过程就像客户机联系本地 DNS 服务器一样。用户的域名解析请求包含流媒体服务器的“名字”，典型的如 URL。

2 名字路由器维护着一张名字路由表。名字路由表包含了名字->下一跳(Name->NextHop）的映射关系，就像 IP 路由表包含了 IP 前缀同下一跳的映射关系。当名字路由表收到一个域名解析请求后，它就在名字路由表中寻找匹配的服务器名字，选择最佳的下一跳。名字路由器把请求转发到下一跳。这样，域名解析请求最终将到达最佳的内容服务器。

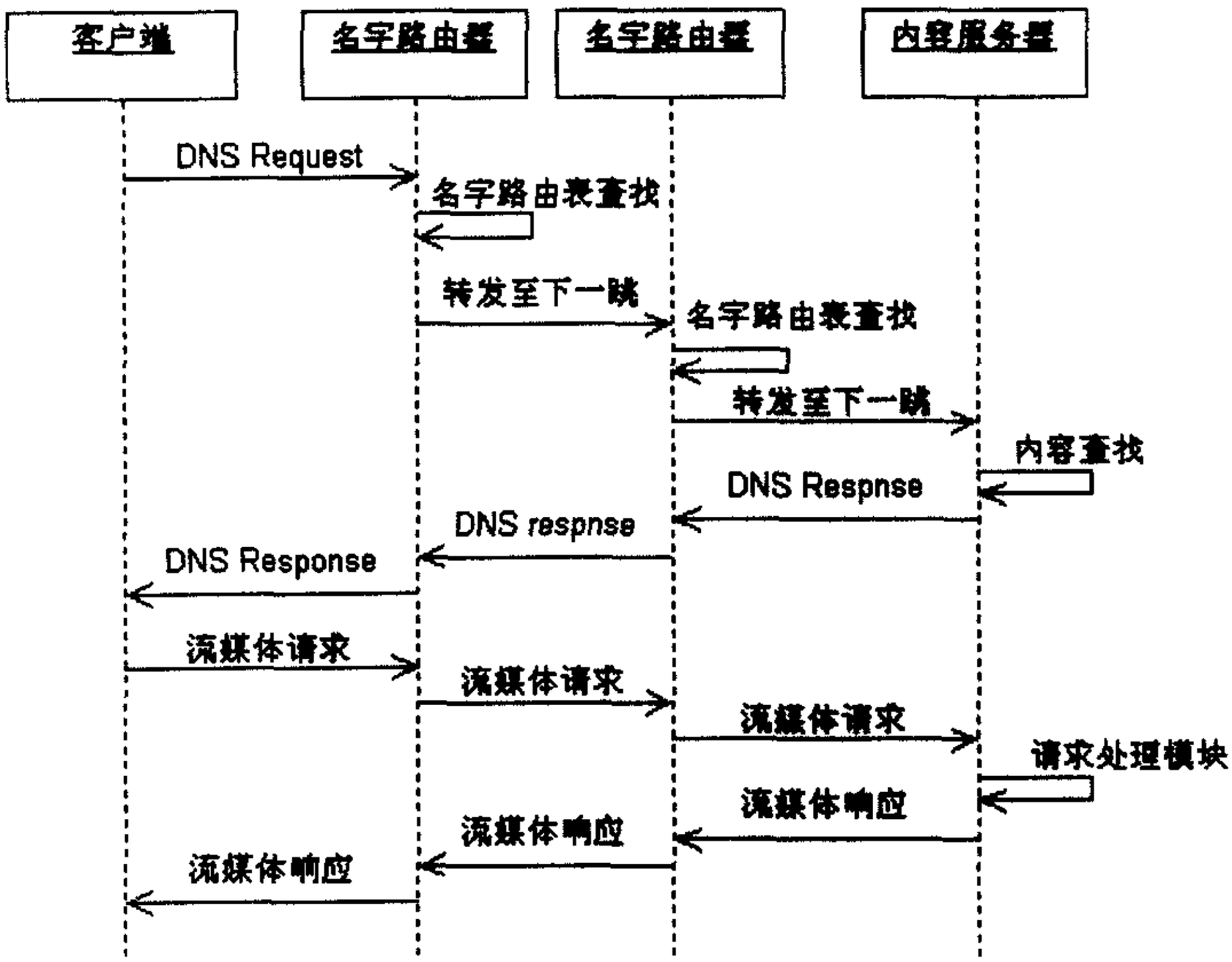


图 2.1 基于 INRP 的用户访问流程示意图

3 内容服务器确认流媒体的内容确实存在，将发出一个响应。响应报文和 DNS 的相应

报文一样，包含了内容服务器的 IP 地址。响应将按照域名解析请求报文的原路径返回给客户机。

4 客户机根据相应报文的内容服务器 IP 地址，向内容服务器发起流媒体的内容请求。内容服务器根据请求返回流媒体相应报文，报文中包含特定的内容。请求和响应根据 DNS 报文建立的路径穿梭。

我们需要指出的是，由于名字路由模块嵌入在传统 IP 核心路由器中，因此 INRP 内容路由的同时将为流媒体的内容请求建立 IP 路由链路。这是因为 INRP 内容路由基于 IP 路由，从客户机到目的内容服务器的沿途 IP 路由器的 Cach 将存在该条路由链路。这为内容获取节省了时间。

一个 INRP 协议可以看成由几个部分组成，每个部分完成一定的功能。基于名字路由的域名解析协议 INRP 主要由以下几个部分组成的。

※ 传入数据报文处理

当名字路由数据报文到达路由器接口时，网络接口软件就把它传给 IP 软件处理。如果数据报文的目的 IP 地址与主机的 IP 地址匹配，IP 软件就接受该数据报并将数据报文向上传递给名字路由报文处理模块做进一步处理。

※ 帧结构

帧结构定义了数据传输的格式和内容。一般的协议都有帧结构。INRP 的帧结构和 DNS 的帧结构相同，但是 DNS 的相同数据段在 INRP 中具有不同的意义，处理的方法也各不相同。名字路由的路由机制采用 DNS 报文的目的是为了和现行的 IP 架构兼容，最大限度的利用现有的 IP 资源。

※ 表驱动算法

通常的名字路由选路算法使用在每台机器上的一个名字路由选路表（Name routing table），该表存储有关可能目的内容服务器和怎样到达内容服务器的信息。名字路由器中的软件需要传送 INRP 报文时，就查询选路表来决定把数据报文发往何处。

※ 名字路由表查找算法

名字路由选路表的大小取决与互联网中内容服务器的数量。当内容服务器的数量很大时，选路表将变的很庞大。路由查找算法的效率将影响名字路由的往返时间。本章列举了几种选路查找算法，对它们进行了性能比较。

※ 报文处理进程

报文处理进程实现名字路由模块的路由机制，它是 INRP 协议区别与 DNS 协议的标志部分，充分的体现了名字路由的思想。本章对此进行了详细的分析和讨论。

※ 重定向机制

没有一个系统能在任何时候都工作正确。除了通信线路和处理区故障外，在目的机器临时或永久断连、寿命计数器超时、或者中间路由器被拥塞无法处理传入的通信业务时，路由器都无法投递数据报。本章主要侧重于寿命计数器超时后的重定向机制研究。

※ INRP 路由仿真算法

我们根据 INRP 的特点，对现有的路由仿真算法进行改进。考虑了路由器的负载、网络

状况、路由器的处理能力，对 INRP 路由系统进行了计算机仿真。

§ 2.2 INRP 协议帧结构

为了和现有的用户流媒体的访问流程匹配，最大限度的利用现有的 IP 资源，名字路由的路由机制 INRP 采用了与 DNS 相同的帧结构。

2.2.1 INRP 的报文格式

请求和相应都用一个 INRP（DNS）报文表示，图 2.2 显示了报文的大致格式。

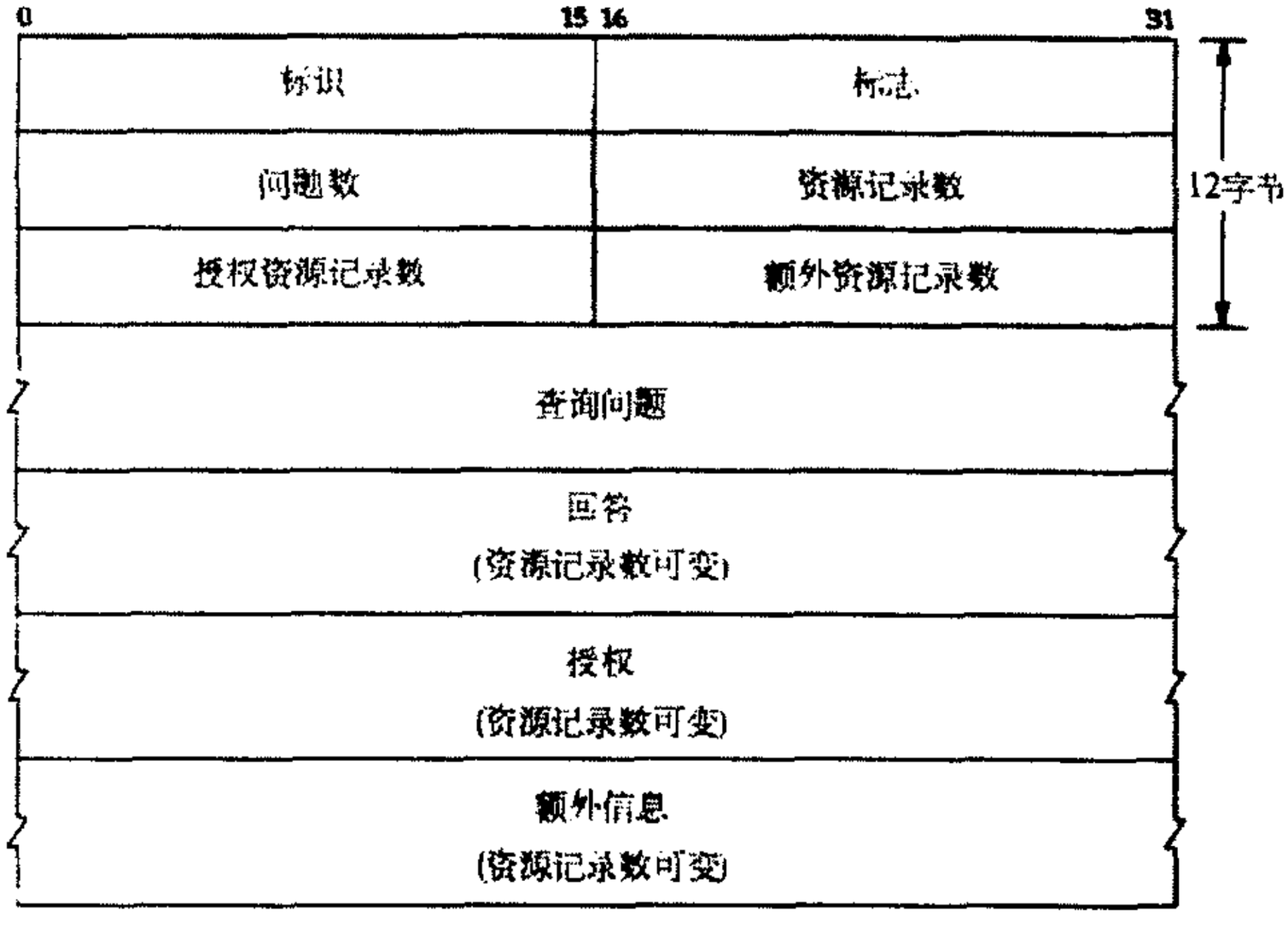


图 2.2 与 DNS 相同的 INRP 请求/响应报文

这个报文由 12 字节长的首部和 4 个长度可变的字段组成。

标识字段由客户程序设置并由服务器返回结果。客户程序通过它来确定响应与请求是否匹配。

16 bit 的标志字段被划分为若干子字段，如图 2.3 所示。

图 2.3 DNS 报文中的标志字段

我们从最左位开始依次介绍各子字段：

- QR 是 1 bit 字段：0 表示查询报文，1 表示响应报文。
- opcode 是一个 4 bit 字段：通常值为 0（标准查询），其他值为 1（反向查询）和 2（服务器状态请求）。
- AA 是 1 bit 标志，表示“授权回答(authoritative answer)”。该名字服务器是授权于该域的。
- TC 是 1 bit 字段，表示“可截断的(truncated)”。使用 UDP 时，它表示当应答的总长度超过 512 字节时，只返回前 512 个字节。
- RD 是 1 bit 字段表示“期望递归（ recursion desired）”。该比特能在一个查询中设

状况、路由器的处理能力，对 INRP 路由系统进行了计算机仿真。

§ 2.2 INRP 协议帧结构

为了和现有的用户流媒体的访问流程匹配，最大限度的利用现有的 IP 资源，名字路由的路由机制 INRP 采用了与 DNS 相同的帧结构。

2.2.1 INRP 的报文格式

请求和相应都用一个 INRP（DNS）报文表示，图 2.2 显示了报文的大致格式。

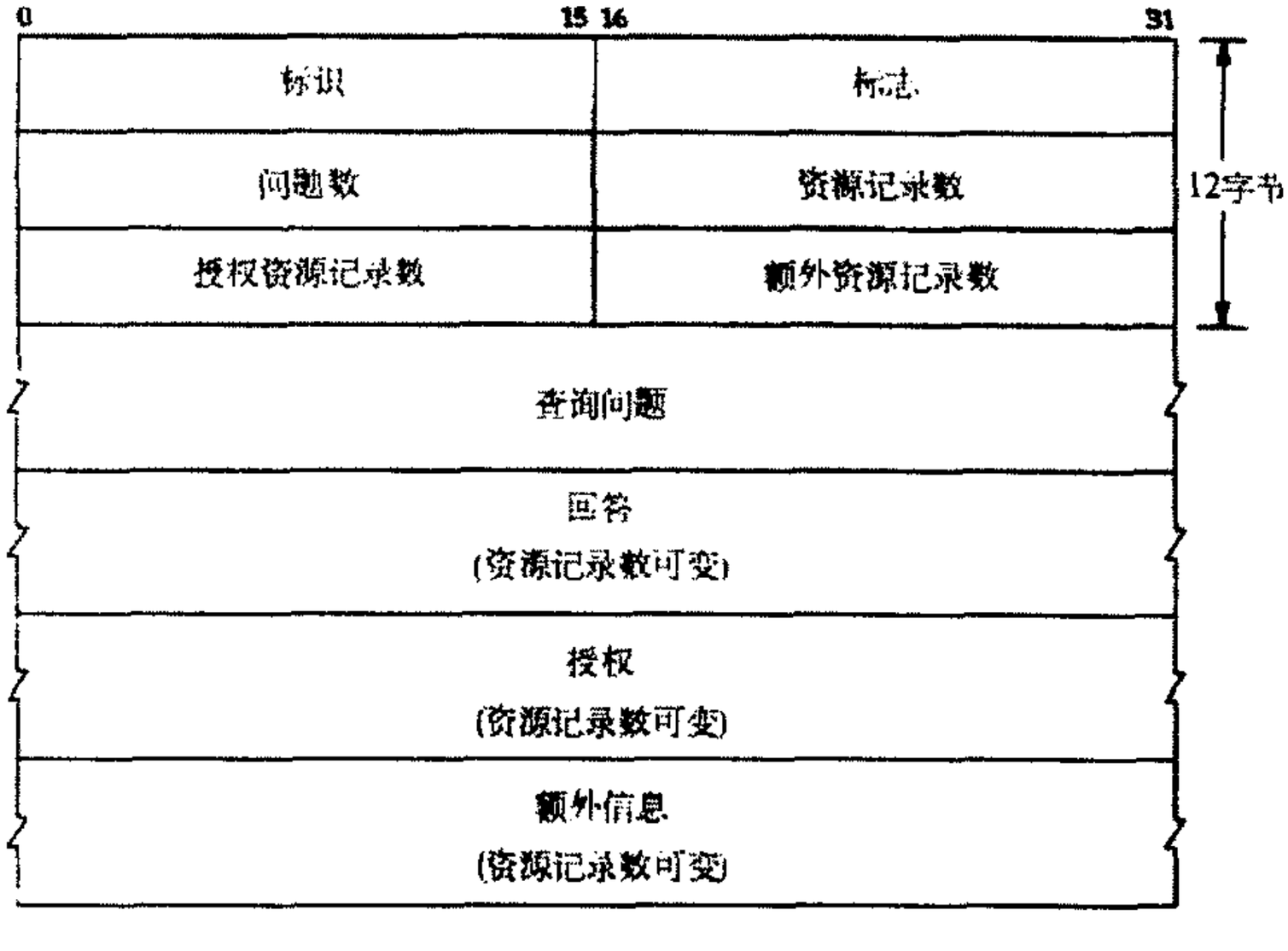


图 2.2 与 DNS 相同的 INRP 请求/响应报文

这个报文由 12 字节长的首部和 4 个长度可变的字段组成。

标识字段由客户程序设置并由服务器返回结果。客户程序通过它来确定响应与请求是否匹配。

16 bit 的标志字段被划分为若干子字段，如图 2.3 所示。

图 2.3 DNS 报文中的标志字段

我们从最左位开始依次介绍各子字段：

- QR 是 1 bit 字段：0 表示查询报文，1 表示响应报文。
- opcode 是一个 4 bit 字段：通常值为 0（标准查询），其他值为 1（反向查询）和 2（服务器状态请求）。
- AA 是 1 bit 标志，表示“授权回答(authoritative answer)”。该名字服务器是授权于该域的。
- TC 是 1 bit 字段，表示“可截断的(truncated)”。使用 UDP 时，它表示当应答的总长度超过 512 字节时，只返回前 512 个字节。
- RD 是 1 bit 字段表示“期望递归（ recursion desired）”。该比特能在一个查询中设

状况、路由器的处理能力，对 INRP 路由系统进行了计算机仿真。

§ 2.2 INRP 协议帧结构

为了和现有的用户流媒体的访问流程匹配，最大限度的利用现有的 IP 资源，名字路由的路由机制 INRP 采用了与 DNS 相同的帧结构。

2.2.1 INRP 的报文格式

请求和相应都用一个 INRP（DNS）报文表示，图 2.2 显示了报文的大致格式。

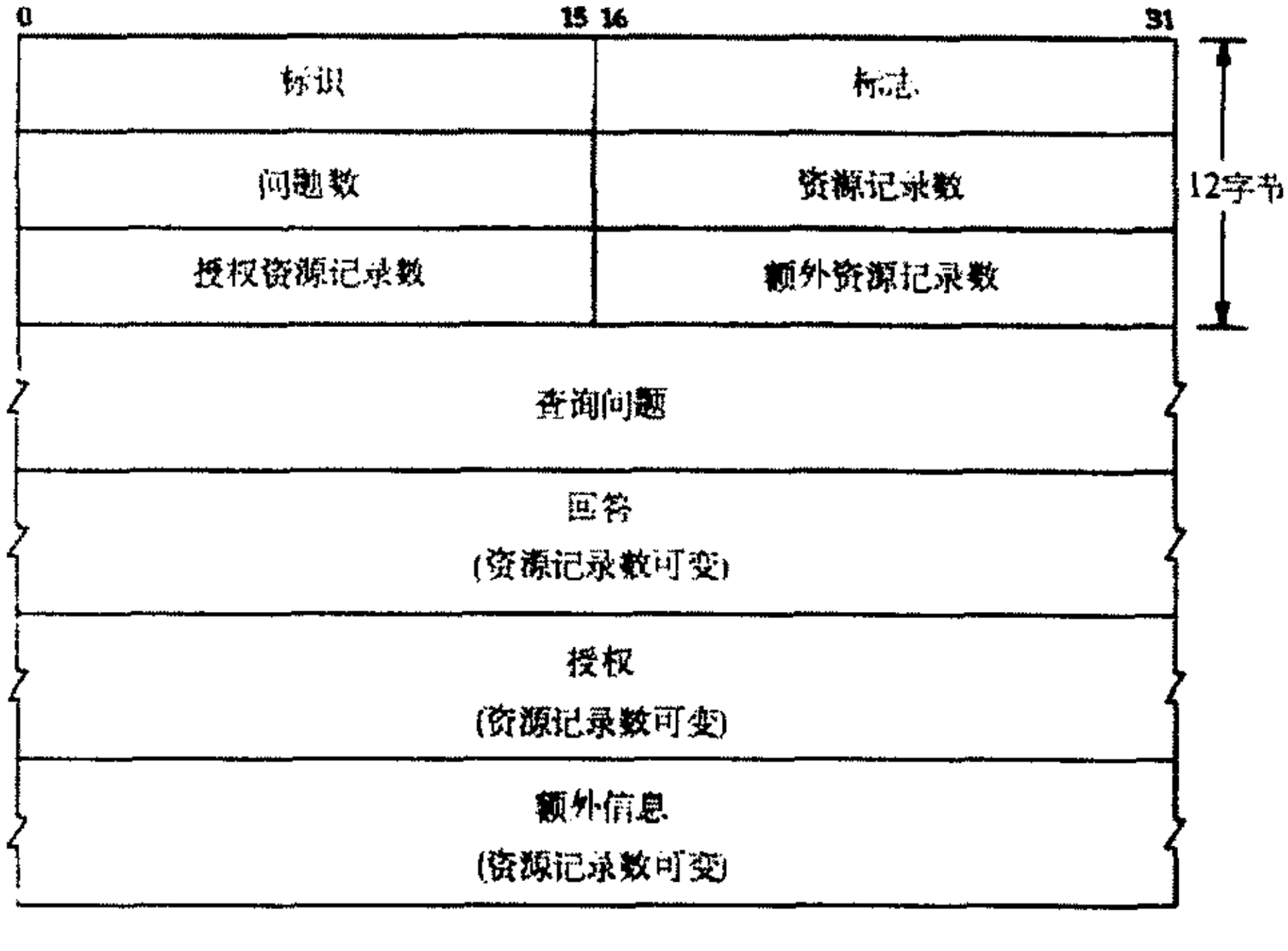


图 2.2 与 DNS 相同的 INRP 请求/响应报文

这个报文由 12 字节长的首部和 4 个长度可变的字段组成。

标识字段由客户程序设置并由服务器返回结果。客户程序通过它来确定响应与请求是否匹配。

16 bit 的标志字段被划分为若干子字段，如图 2.3 所示。

图 2.3 DNS 报文中的标志字段

我们从最左位开始依次介绍各子字段：

- QR 是 1 bit 字段：0 表示查询报文，1 表示响应报文。
- opcode 是一个 4 bit 字段：通常值为 0（标准查询），其他值为 1（反向查询）和 2（服务器状态请求）。
- AA 是 1 bit 标志，表示“授权回答(authoritative answer)”。该名字服务器是授权于该域的。
- TC 是 1 bit 字段，表示“可截断的(truncated)”。使用 UDP 时，它表示当应答的总长度超过 512 字节时，只返回前 512 个字节。
- RD 是 1 bit 字段表示“期望递归（ recursion desired）”。该比特能在一个查询中设

置，并在响应中返回。这个标志告诉名字服务器必须处理这个查询，也称为一个递归查询。如果该位为 0，且被请求的名字服务器没有一个授权回答，它就返回一个能解答该查询的其他名字服务器列表，这称为迭代查询。在现阶段的研究中，INRP 将不支持这个字段。

- RA 是 1 bit 字段，表示“可用递归”。如果名字服务器支持递归查询，则在响应中将该比特设置为 1。大多数名字服务器都提供递归查询。现阶段，INRP 不支持这个字段。
- 随后的3 bit字段必须为0。
- rcode 是一个4 bit的返回码字段。通常的值为0（没有差错）和3（名字差错）。名字差错只有从一个授权名字服务器上返回，它表示在查询中制定的域名不存在。

随后的4个16 bit 字段说明最后4个变长字段中包含的条目数。对于查询报文，问题（question）数通常是1，而其他3项则均为0。类似地，对于应答报文，回答数至少是1，剩下的两项可以是0或非0。

2.2.2 INRP 查询报文中的问题部分

问题部分中每个问题的格式如图 2.4 所示，通常只有一个问题。

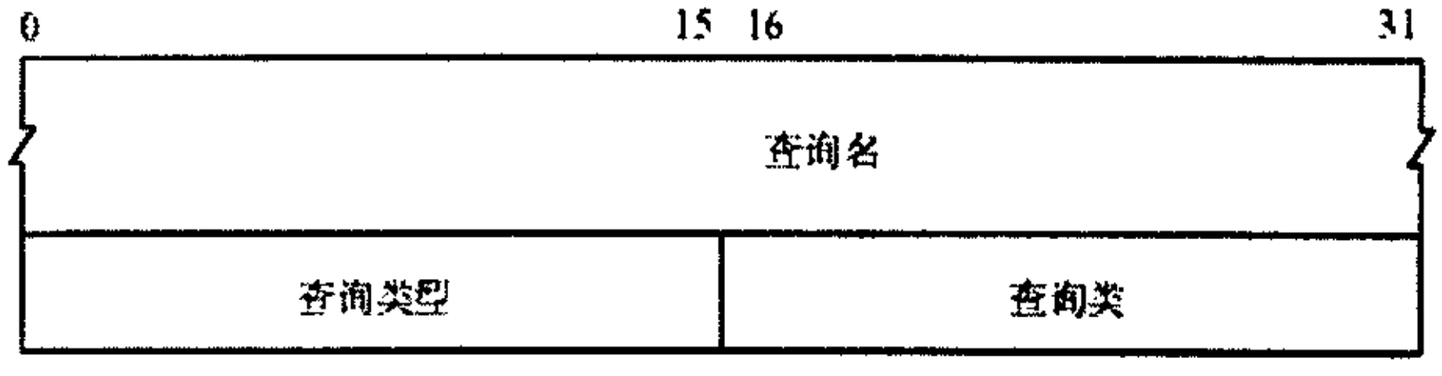


图 2.4 DNS 查询报文中的问题部分的格式

图 2.4 给出了 INRP（DNS）查询报文中问题部分的格式。查询名是要查找的名字，它是一个或多个标识符的序列。每个标识符以首字节的计数值来说明随后标识符的字节长度，每个名字以最后字节为 0 结束，长度为 0 的标识符是根标识符。

计数字节的值必须是 0 ~ 6 3 的数，因为标识符的最大长度仅为 6 3（在本节的后面我们将看到计数字节的最高两比特为 1，即值 1 9 2 ~ 2 5 5，将用于压缩格式）。不像我们已经看到的许多其他报文格式，该字段无需以整 32 bit 边界结束，即无需填充字节。

图2.5显示了如何存储域名gemini.tuc.noao.edu。

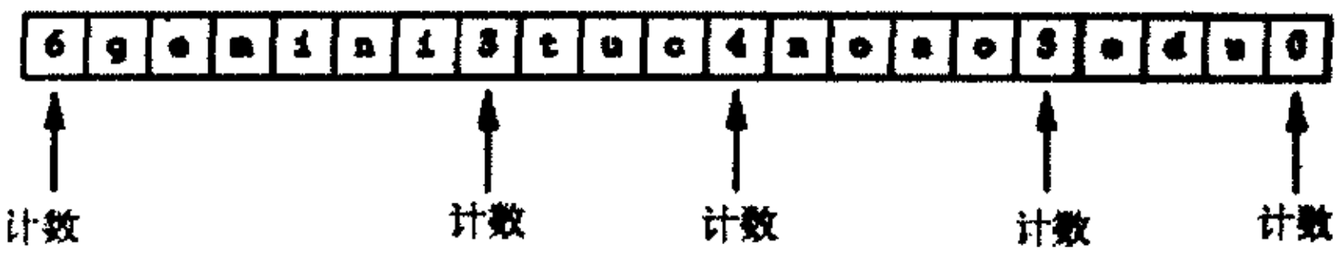


图 2.5 gemini.tuc.noao.edu 的表示

每个问题有一个查询类型，而每个响应（也称一个资源记录，我们下面将谈到）也有一个类型。大约有 2 0 个不同的类型值，其中的一些目前已经过时。图 2.6 显示了其中的一些值。

最常用的查询类型是A类型，表示期望获得查询名的I P地址。

查询类通常是1，指互联网地址（某些站点也支持其他非I P地址）。

置，并在响应中返回。这个标志告诉名字服务器必须处理这个查询，也称为一个递归查询。如果该位为 0，且被请求的名字服务器没有一个授权回答，它就返回一个能解答该查询的其他名字服务器列表，这称为迭代查询。在现阶段的研究中，INRP 将不支持这个字段。

- RA 是 1 bit 字段，表示“可用递归”。如果名字服务器支持递归查询，则在响应中将该比特设置为 1。大多数名字服务器都提供递归查询。现阶段，INRP 不支持这个字段。
- 随后的3 bit字段必须为0。
- rcode 是一个4 bit的返回码字段。通常的值为0（没有差错）和3（名字差错）。名字差错只有从一个授权名字服务器上返回，它表示在查询中制定的域名不存在。

随后的4个16 bit 字段说明最后4个变长字段中包含的条目数。对于查询报文，问题（question）数通常是1，而其他3项则均为0。类似地，对于应答报文，回答数至少是1，剩下的两项可以是0或非0。

2.2.2 INRP 查询报文中的问题部分

问题部分中每个问题的格式如图 2.4 所示，通常只有一个问题。

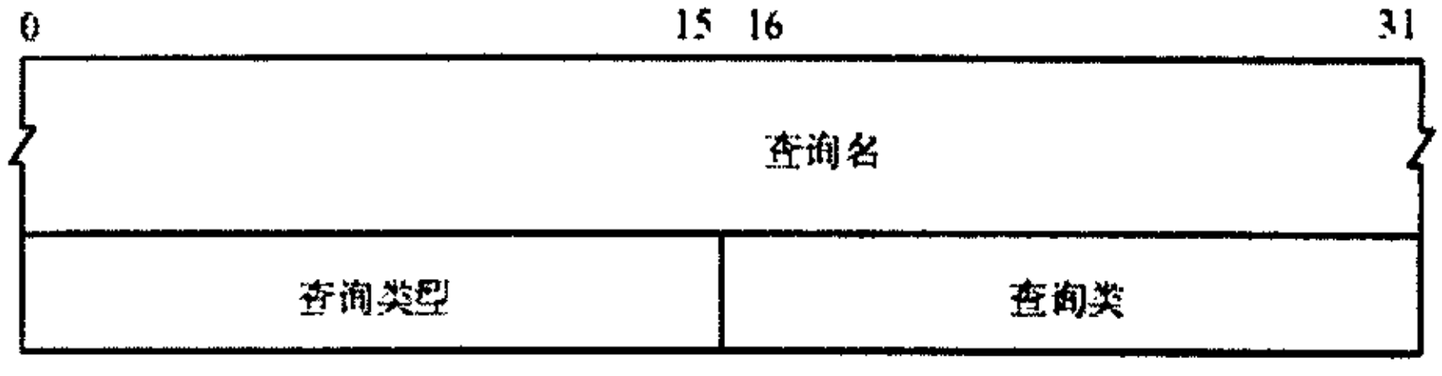


图 2.4 DNS 查询报文中的问题部分的格式

图 2.4 给出了 INRP（DNS）查询报文中问题部分的格式。查询名是要查找的名字，它是一个或多个标识符的序列。每个标识符以首字节的计数值来说明随后标识符的字节长度，每个名字以最后字节为 0 结束，长度为 0 的标识符是根标识符。

计数字节的值必须是 0 ~ 6 3 的数，因为标识符的最大长度仅为 6 3（在本节的后面我们将看到计数字节的最高两比特为 1，即值 1 9 2 ~ 2 5 5，将用于压缩格式）。不像我们已经看到的许多其他报文格式，该字段无需以整 32 bit 边界结束，即无需填充字节。

图2.5显示了如何存储域名gemini.tuc.noao.edu。

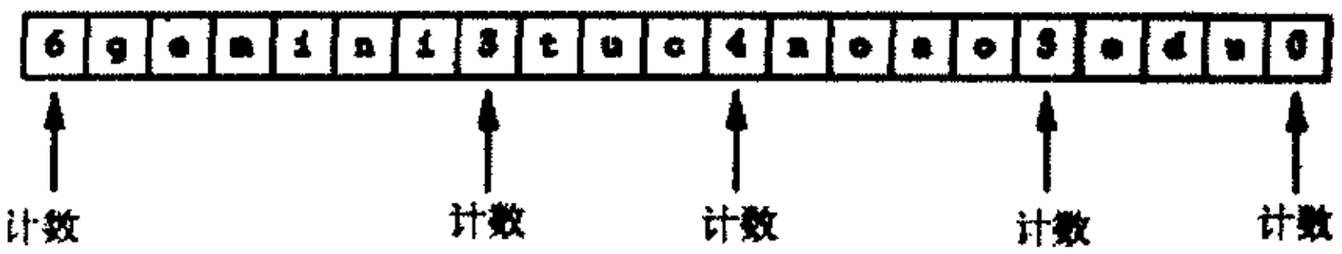


图 2.5 gemini.tuc.noao.edu 的表示

每个问题有一个查询类型，而每个响应（也称一个资源记录，我们下面将谈到）也有一个类型。大约有 2 0 个不同的类型值，其中的一些目前已经过时。图 2.6 显示了其中的一些值。

最常用的查询类型是A类型，表示期望获得查询名的I P地址。

查询类通常是1，指互联网地址（某些站点也支持其他非I P地址）。

名　字	数　值	描　述	类型?	查询类型
A	1	IP地址	•	•
NS	2	名字服务器	•	•
CNAME	5	规范名称	•	•
PTR	12	指针记录	•	•
HINFO	13	主机信息	•	•
MX	15	邮件交换记录	•	•
AXFR	252	对区域转换的请求		•
*或ANY	255	对所有记录的请求		•

图 2.6　INRP 问题和响应的类型值和查询的类型值

2.2.3　INRP 响应报文中的资源记录部分

DNS 报文中最后的三个字段，回答字段、授权字段和附加信息字段，均采用一种称为资源 RR(Resource Record)的相同格式。图 2.7 显示了资源记录的格式。

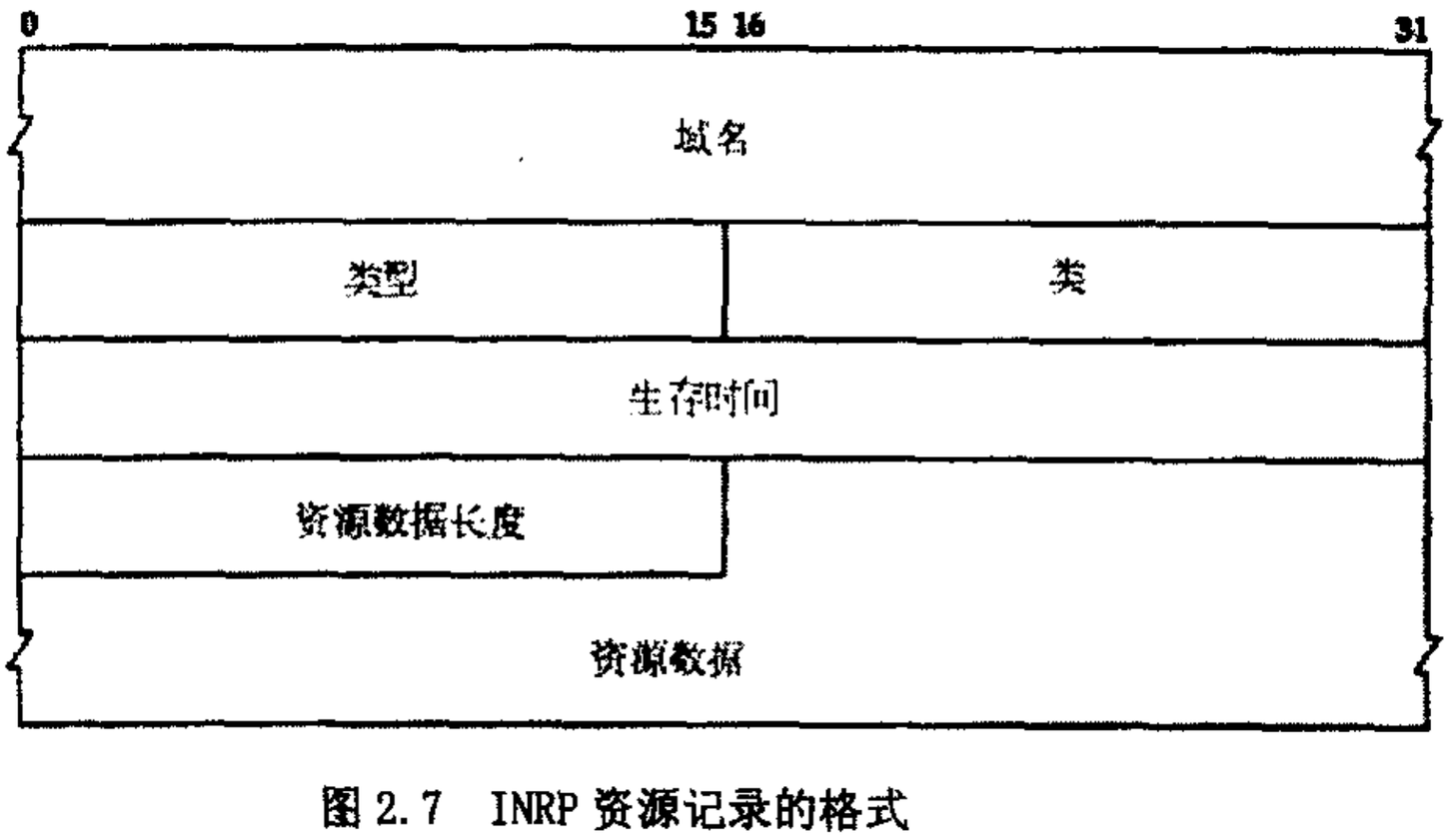


图 2.7　INRP 资源记录的格式

域名是记录中资源数据对应的名字。它的格式和前面介绍的查询名字段格式（图 2.5）相同。

类型说明 R R 的类型码。它的值和前面介绍的查询类型值是一样的。类通常为 1，指 Internet 数据。

生存时间字段是客户程序保留该资源记录的秒数。资源记录通常的生存时间值为 2 天。

资源数据长度说明资源数据的数量。该数据的格式依赖于类型字段的值。对于类型 1（A 记录），资源数据是 4 字节的 I P 地址。

§ 2.3　INRP 的报文处理进程

上节介绍了 INRP 的帧结构，它的报文与 DNS 的报文是完全一致的。本节将着重讨论报文的处理进程。

报文处理进程实现名字路由模块的路由机制，它由三个进程组成：

1 INRP(DNS)查询报文的接收进程

2 高速 DNS 缓存查找进程

3 事务处理进程

事务处理进程是 INRP 协议区别与 DNS 协议的标志部分。它充分的体现了名字路由的思想。

名　字	数　值	描　述	类型?	查询类型
A	1	IP地址	•	•
NS	2	名字服务器	•	•
CNAME	5	规范名称	•	•
PTR	12	指针记录	•	•
HINFO	13	主机信息	•	•
MX	15	邮件交换记录	•	•
AXFR	252	对区域转换的请求		•
*或ANY	255	对所有记录的请求		•

图 2.6　INRP 问题和响应的类型值和查询的类型值

2.2.3　INRP 响应报文中的资源记录部分

DNS 报文中最后的三个字段，回答字段、授权字段和附加信息字段，均采用一种称为资源 RR(Resource Record)的相同格式。图 2.7 显示了资源记录的格式。

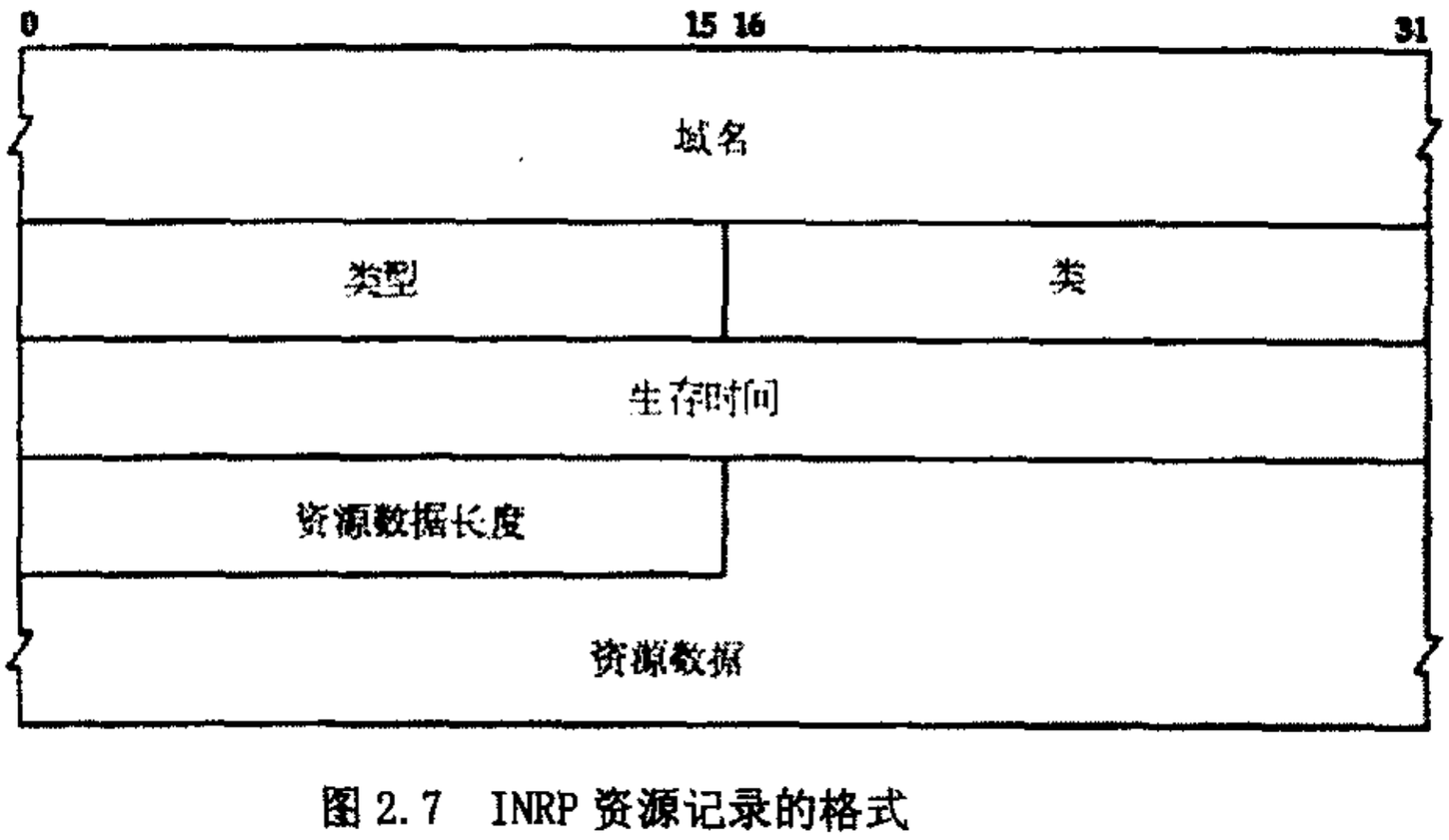


图 2.7　INRP 资源记录的格式

域名是记录中资源数据对应的名字。它的格式和前面介绍的查询名字段格式（图 2.5）相同。

类型说明 R R 的类型码。它的值和前面介绍的查询类型值是一样的。类通常为 1，指 Internet 数据。

生存时间字段是客户程序保留该资源记录的秒数。资源记录通常的生存时间值为 2 天。

资源数据长度说明资源数据的数量。该数据的格式依赖于类型字段的值。对于类型 1（A 记录），资源数据是 4 字节的 I P 地址。

§ 2.3　INRP 的报文处理进程

上节介绍了 INRP 的帧结构，它的报文与 DNS 的报文是完全一致的。本节将着重讨论报文的处理进程。

报文处理进程实现名字路由模块的路由机制，它由三个进程组成：

1 INRP(DNS)查询报文的接收进程

2 高速 DNS 缓存查找进程

3 事务处理进程

事务处理进程是 INRP 协议区别与 DNS 协议的标志部分。它充分的体现了名字路由的思想。

名　字	数　值	描　述	类型?	查询类型
A	1	IP地址	•	•
NS	2	名字服务器	•	•
CNAME	5	规范名称	•	•
PTR	12	指针记录	•	•
HINFO	13	主机信息	•	•
MX	15	邮件交换记录	•	•
AXFR	252	对区域转换的请求		•
*或ANY	255	对所有记录的请求		•

图 2.6　INRP 问题和响应的类型值和查询的类型值

2.2.3　INRP 响应报文中的资源记录部分

DNS 报文中最后的三个字段，回答字段、授权字段和附加信息字段，均采用一种称为资源 RR(Resource Record)的相同格式。图 2.7 显示了资源记录的格式。

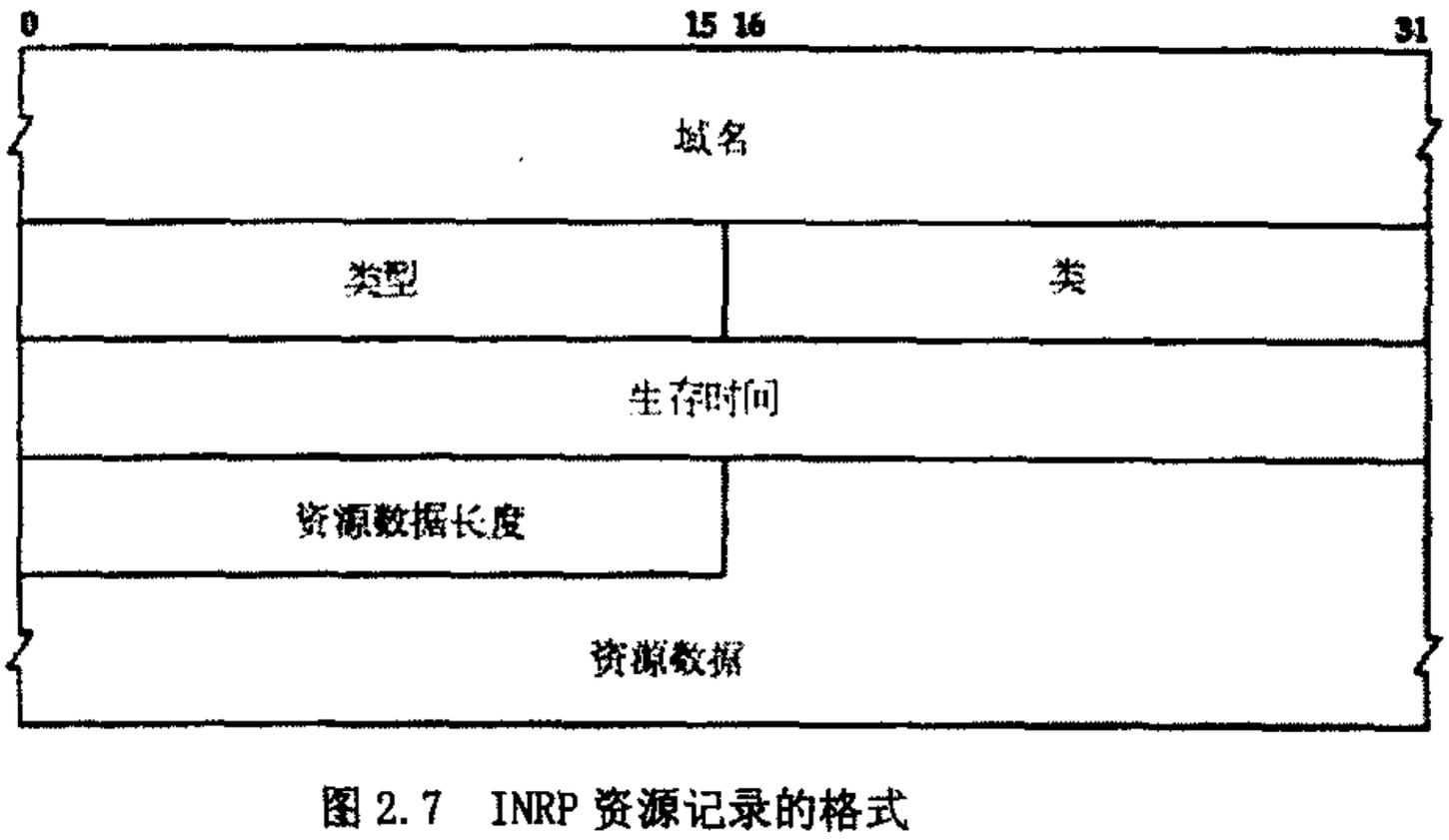


图 2.7　INRP 资源记录的格式

域名是记录中资源数据对应的名字。它的格式和前面介绍的查询名字段格式（图 2.5）相同。

类型说明 R R 的类型码。它的值和前面介绍的查询类型值是一样的。类通常为 1，指 Internet 数据。

生存时间字段是客户程序保留该资源记录的秒数。资源记录通常的生存时间值为 2 天。

资源数据长度说明资源数据的数量。该数据的格式依赖于类型字段的值。对于类型 1（A 记录），资源数据是 4 字节的 I P 地址。

§ 2.3　INRP 的报文处理进程

上节介绍了 INRP 的帧结构，它的报文与 DNS 的报文是完全一致的。本节将着重讨论报文的处理进程。

报文处理进程实现名字路由模块的路由机制，它由三个进程组成：

1 INRP(DNS)查询报文的接收进程

2 高速 DNS 缓存查找进程

3 事务处理进程

事务处理进程是 INRP 协议区别与 DNS 协议的标志部分。它充分的体现了名字路由的思想。

接受进程的任务就是从 TCP、UDP 接收 INRP（DNS）报文。对于一条新的查询报文，我们首先进行高速 DNS 缓存查找。如果直接命中高速 DNS 缓存，我们将向发出查询报文的机器发出响应报文。否则，我们要创建新的事务处理进程，负责该条报文的名字路由。一条新的查询报文对应一个新的事务处理进程。

事务处理进程按照表驱动路由算法，根据报文的域名进行名字路由表的搜索，确定最佳的下一跳。事务处理进程将查询报文转发到下一跳，并等待响应报文的到达。响应报文到达后，事务处理进程将响应报文转发回发出查询报文的机器。

一个完整的事务处理进程还包括重定向机制。当定时器超时而报文未到达，事务处理进程将启动重定向机制。它根据报文的查询域名及特定的策略重新选择一个下一跳，将查询报文转发并启动定时器。如果响应报文及时到达，事务处理进程将响应报文发向发出查询报文的机器。否则，事务处理进程将被终止。

图 2.8 给出了报文处理进程的模块和数据流图。

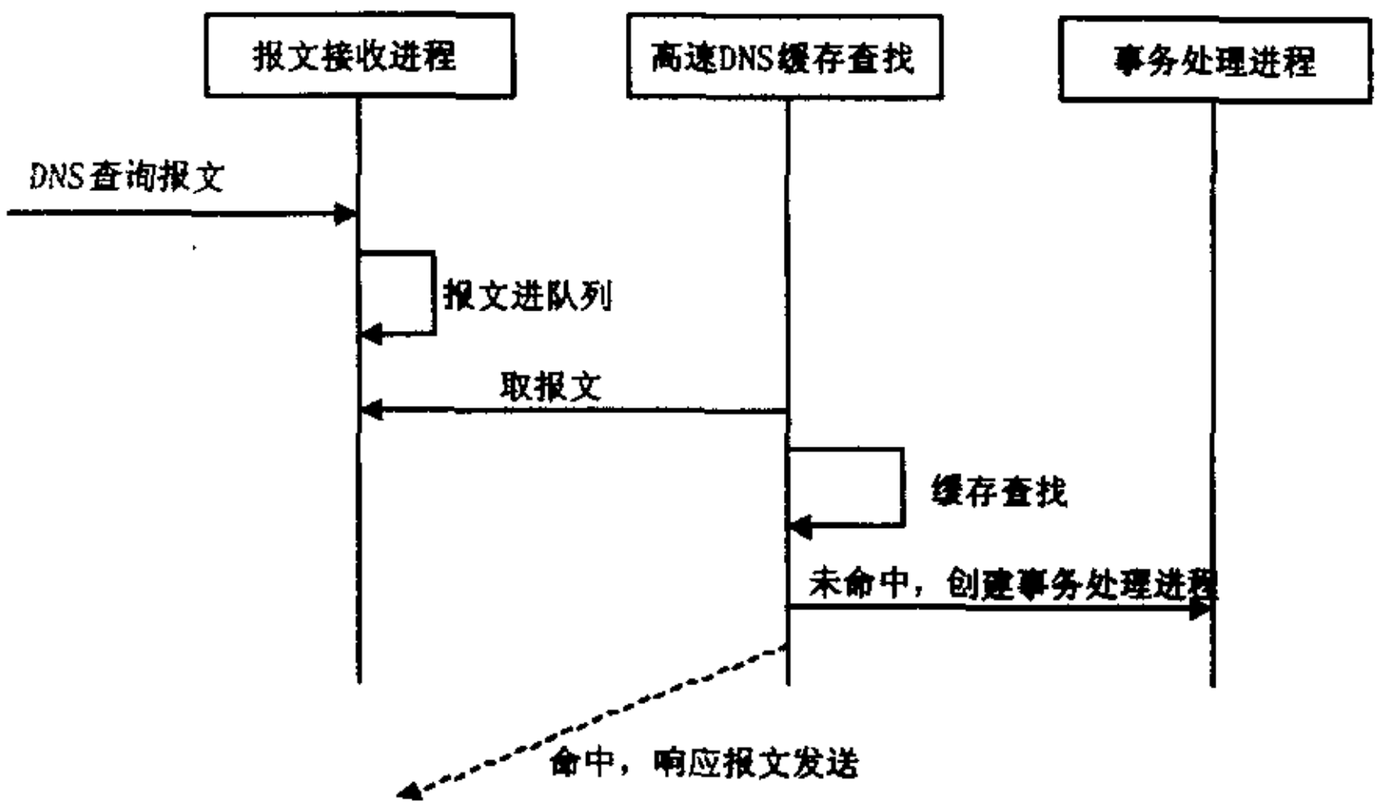


图 2.8 报文处理进程模块和数据流图

2.3.1 查询报文的接收进程

接收进程的任务围绕数据报的接收而展开。当 DNS 查询数据包到达主机时，网络接口软件就把它传给接受进程处理。接收进程分析报文。如果报文的各个数据字段格式正确，那么进程就把报文以及发送方的信息（IP 地址、端口号等）送入原始报文队列。DNS 缓存查找进程将从队列取报文信息。如果报文的数据字段有格式错误的话，主机将丢弃报文。

接收进程是名字路由的“引擎”。当接收进程不工作或者没有报文时，整个报文处理进程将处于“待机”状态；当接收进程“满荷” 甚至报文过多将进程拥塞时，整个报文处理进程也将充分利用主机资源处于“满荷载”状态，甚至可能耗尽主机资源。

原始报文队列的大小将成为制约接收进程工作效率的一个“瓶颈”。如果队列太小，进程将很容易因为报文的增多而“阻塞”，路由器将不能接收新的查询报文；如果队列太大，系统很有可能长期处于“超负荷”运行状态，甚至因为系统资源耗尽而崩溃。因此，队列的大小必须根据本地的网络状况以及系统的资源状况、处理能力而定。为了便于维护和管理，队列的大小必须是可以动态配置的。

对于名字路由器来说，名字路由模块应该是可配的。名字路由器如果去掉或者禁止名字路由模块，名字路由器就成为一个普通的 IP 路由器。前面提到，接收进程是名字路由的引擎。为了符合名字路由器的可配置的特性，接收进程也应该是可配置。接收进程由待机、运行、暂时挂起、参数设置、挂起五个状态组成。图 2.9 给出了接收进程的状态转移图。

接受进程的任务就是从 TCP、UDP 接收 INRP（DNS）报文。对于一条新的查询报文，我们首先进行高速 DNS 缓存查找。如果直接命中高速 DNS 缓存，我们将向发出查询报文的机器发出响应报文。否则，我们要创建新的事务处理进程，负责该条报文的名字路由。一条新的查询报文对应一个新的事务处理进程。

事务处理进程按照表驱动路由算法，根据报文的域名进行名字路由表的搜索，确定最佳的下一跳。事务处理进程将查询报文转发到下一跳，并等待响应报文的到达。响应报文到达后，事务处理进程将响应报文转发回发出查询报文的机器。

一个完整的事务处理进程还包括重定向机制。当定时器超时而报文未到达，事务处理进程将启动重定向机制。它根据报文的查询域名及特定的策略重新选择一个下一跳，将查询报文转发并启动定时器。如果响应报文及时到达，事务处理进程将响应报文发向发出查询报文的机器。否则，事务处理进程将被终止。

图 2.8 给出了报文处理进程的模块和数据流图。

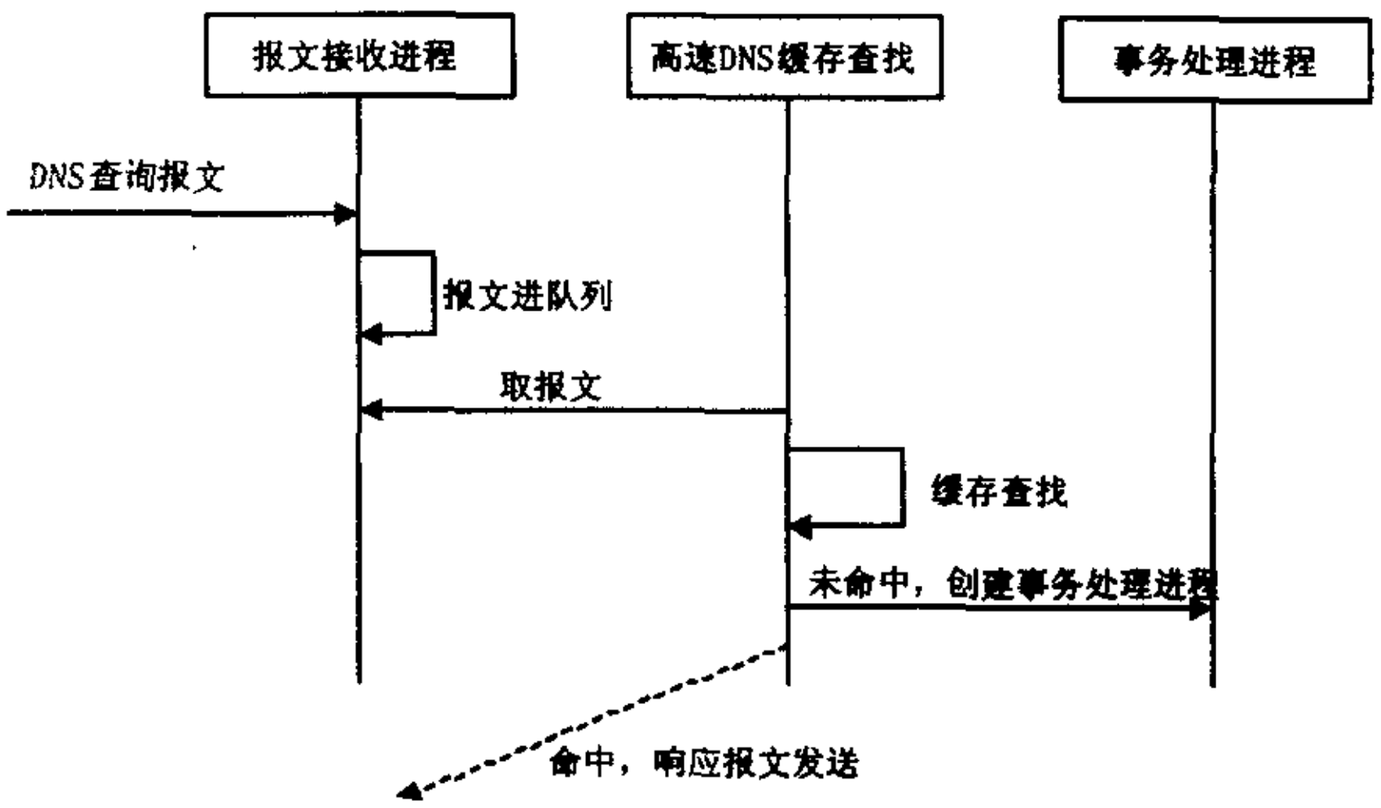


图 2.8 报文处理进程模块和数据流图

2.3.1 查询报文的接收进程

接收进程的任务围绕数据报的接收而展开。当 DNS 查询数据包到达主机时，网络接口软件就把它传给接受进程处理。接收进程分析报文。如果报文的各个数据字段格式正确，那么进程就把报文以及发送方的信息（IP 地址、端口号等）送入原始报文队列。DNS 缓存查找进程将从队列取报文信息。如果报文的数据字段有格式错误的话，主机将丢弃报文。

接收进程是名字路由的“引擎”。当接收进程不工作或者没有报文时，整个报文处理进程将处于“待机”状态；当接收进程“满荷” 甚至报文过多将进程拥塞时，整个报文处理进程也将充分利用主机资源处于“满荷载”状态，甚至可能耗尽主机资源。

原始报文队列的大小将成为制约接收进程工作效率的一个“瓶颈”。如果队列太小，进程将很容易因为报文的增多而“阻塞”，路由器将不能接收新的查询报文；如果队列太大，系统很有可能长期处于“超负荷”运行状态，甚至因为系统资源耗尽而崩溃。因此，队列的大小必须根据本地的网络状况以及系统的资源状况、处理能力而定。为了便于维护和管理，队列的大小必须是可以动态配置的。

对于名字路由器来说，名字路由模块应该是可配的。名字路由器如果去掉或者禁止名字路由模块，名字路由器就成为一个普通的 IP 路由器。前面提到，接收进程是名字路由的引擎。为了符合名字路由器的可配置的特性，接收进程也应该是可配置。接收进程由待机、运行、暂时挂起、参数设置、挂起五个状态组成。图 2.9 给出了接收进程的状态转移图。

待机状态： 接收进程的初始状态。在此期间，进程不接收报文，但是可以接收系统的配置信号，配置队列长度参数。当收到运行信号时，进入运行状态。前面说过，队列长度可以配置，待机状态同时接收队列长度配置信号并进入暂时挂起状态。需要注意的是，这个阶段的队列为空。当收到挂起信号后，进程进入挂起状态。

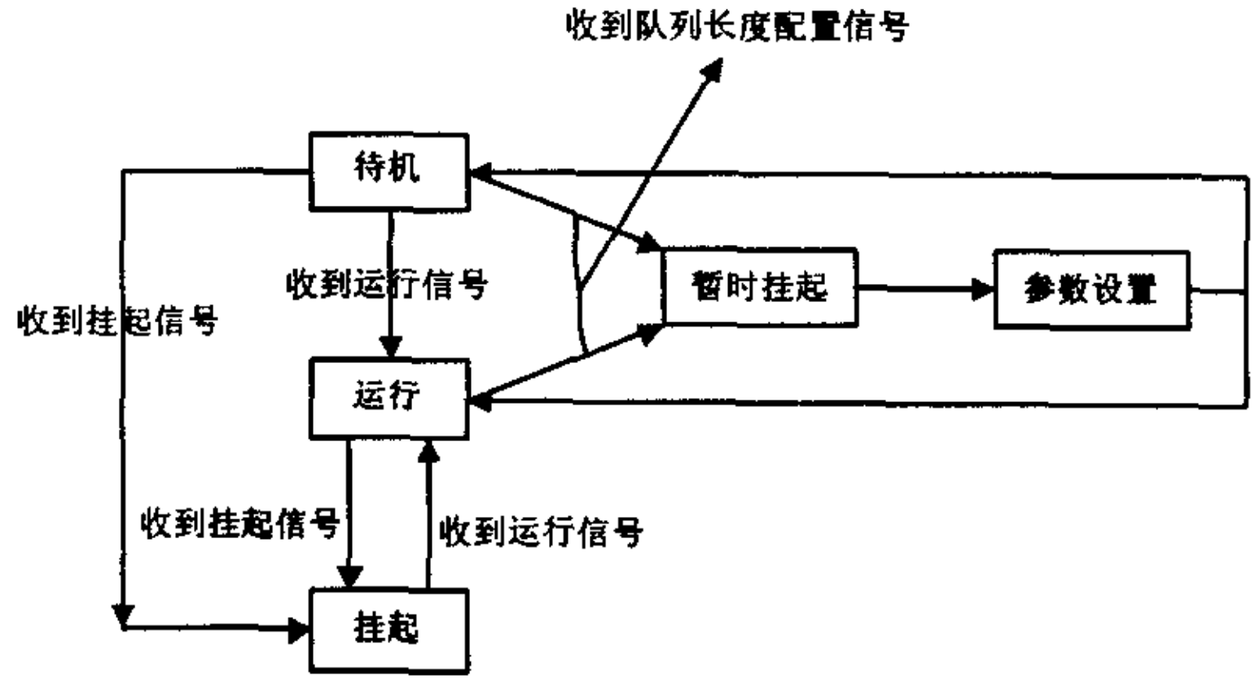


图 2.9 接收进程的状态转移图

暂时挂起状态： 进程进入本状态后，将关闭信号接收模块。进程将处于信号屏蔽状态，拒绝接收任何信号。当信号接收模块关闭后，进程自动进入参数设置状态。设置这个状态的原因是为了提高系统的健壮性。由于参数设置状态将动态改变队列的长度，涉及对队列的添加、删除等操作，如果在此期间进行状态转移或者继续接收报文，将破坏队列，导致报文丢失或者报文拥塞等情况发生。

参数设置状态： 进程将改变队列的长度。如果队列为空，进程将不进行任何动作；如果队列不为空，将依据要配置的队列长度参数值进行动作。

1. 现阶段队列长度大于参数值。进程依然保持现有队列，但退出参数设置状态回到运行状态后，进程将抛弃新到达的报文，直到队列长度小于设置的队列长度。进程开始恢复对报文的接收和处理。

2 现阶段队列长度小于等于参数值。这种情况比较简单，进程将设置队列的最大长度。当进程要退出参数设置状态时，将重新开启信号接收模块。

运行状态： 进程开始接收和处理报文。进程监听 DNS 众所周知的端口 53，接收 DNS 查询报文。如果有报文到达，进程检查报文的各个数据段的格式正确性。如果格式正确，进程准备将报文放入原始报文队列。进程必须检查当前队列是否已满。如果当前队列长度已经超过系统设置的队列长度，队列已经不能容纳新的报文，该条报文将被丢弃；否则，进程为报文申请内存空间，将该条报文放入队列，队列长度加 1。

2.3.2 高速 DNS 缓存查找进程

高速 DNS 缓存查找进程主要是为了提高系统的性能。我们不必要为每一条 DNS 查询报文创建一个新的事务处理进程，进行名字路由。我们可以使用缓存策略，把最近查询过的域名解析存放在高速的域名解析缓存表中。只有当在域名缓存表中查找失败的时候,我们才需要创建一个新的事务处理进程，进行真正完整的名字路由过程。

我们首先来看高速 DNS 缓存查找的整个进程。图 2.10 给出了进程的流程图。

待机状态： 接收进程的初始状态。在此期间，进程不接收报文，但是可以接收系统的配置信号，配置队列长度参数。当收到运行信号时，进入运行状态。前面说过，队列长度可以配置，待机状态同时接收队列长度配置信号并进入暂时挂起状态。需要注意的是，这个阶段的队列为空。当收到挂起信号后，进程进入挂起状态。

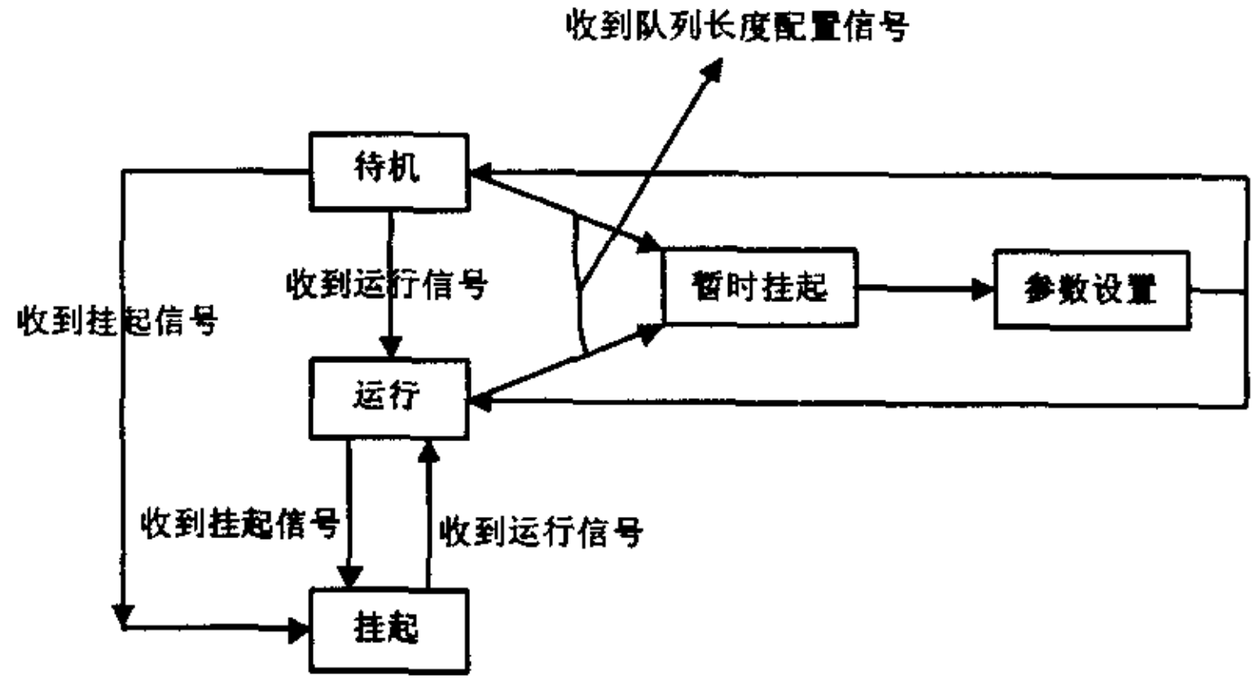


图 2.9 接收进程的状态转移图

暂时挂起状态： 进程进入本状态后，将关闭信号接收模块。进程将处于信号屏蔽状态，拒绝接收任何信号。当信号接收模块关闭后，进程自动进入参数设置状态。设置这个状态的原因是为了提高系统的健壮性。由于参数设置状态将动态改变队列的长度，涉及对队列的添加、删除等操作，如果在此期间进行状态转移或者继续接收报文，将破坏队列，导致报文丢失或者报文拥塞等情况发生。

参数设置状态： 进程将改变队列的长度。如果队列为空，进程将不进行任何动作；如果队列不为空，将依据要配置的队列长度参数值进行动作。

1. 现阶段队列长度大于参数值。进程依然保持现有队列，但退出参数设置状态回到运行状态后，进程将抛弃新到达的报文，直到队列长度小于设置的队列长度。进程开始恢复对报文的接收和处理。

2 现阶段队列长度小于等于参数值。这种情况比较简单，进程将设置队列的最大长度。当进程要退出参数设置状态时，将重新开启信号接收模块。

运行状态： 进程开始接收和处理报文。进程监听 DNS 众所周知的端口 53，接收 DNS 查询报文。如果有报文到达，进程检查报文的各个数据段的格式正确性。如果格式正确，进程准备将报文放入原始报文队列。进程必须检查当前队列是否已满。如果当前队列长度已经超过系统设置的队列长度，队列已经不能容纳新的报文，该条报文将被丢弃；否则，进程为报文申请内存空间，将该条报文放入队列，队列长度加 1。

2.3.2 高速 DNS 缓存查找进程

高速 DNS 缓存查找进程主要是为了提高系统的性能。我们不必要为每一条 DNS 查询报文创建一个新的事务处理进程，进行名字路由。我们可以使用缓存策略，把最近查询过的域名解析存放在高速的域名解析缓存表中。只有当在域名缓存表中查找失败的时候,我们才需要创建一个新的事务处理进程，进行真正完整的名字路由过程。

我们首先来看高速 DNS 缓存查找的整个进程。图 2.10 给出了进程的流程图。

1　缓存查找进程从原始报文队列提取报文。如果队列不为空，进程取出队列头的原始报文，队列长度减 1。

2　根据报文的解析域名，在高速的域名解析缓存表中查找对应的表项。如果在缓存表中名中对应的表项，则进入报文响应模块；如果缓存表中没有对应的表项，查找失败，则要为报文创建新的事务处理进程。

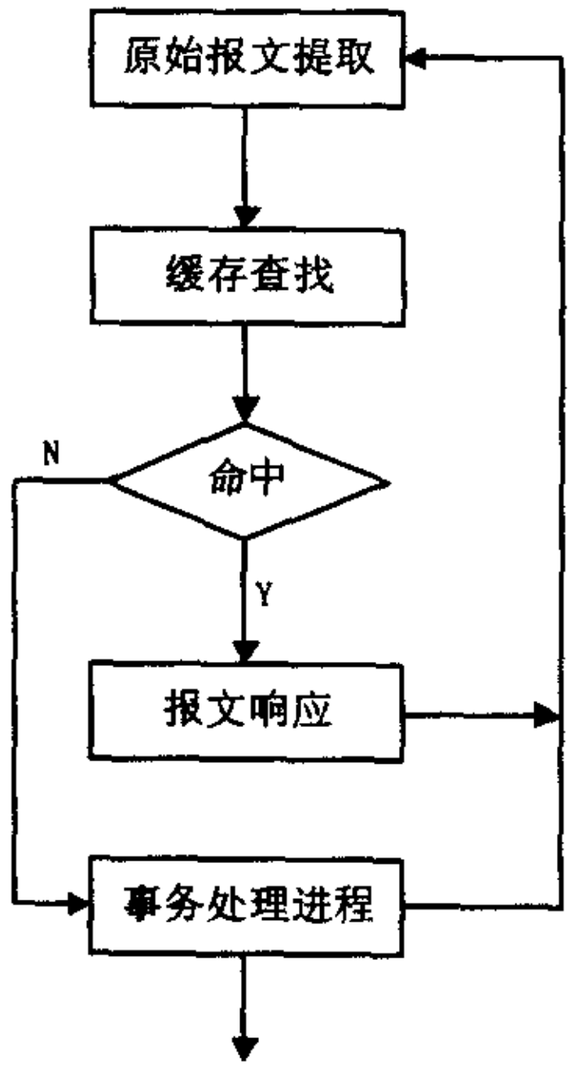


图 2.10　高速 DNS 缓存查找的流程图

3　由于报文命中高速缓存表，进程可以根据缓存表对应表项的目的 IP 地址，以及原始查询报文，创建响应报文。响应报文的具体格式可以如 2.2 节描述。一般的，响应报文只有一个 Answer Section，其中的 Address 填充为缓存表项的目的 IP 地址。进程可以从原始报文中提取发送报文方的 IP 地址和端口，向发送方发出响应报文。

4　当报文不存在对应的高速缓存表表项，进程必须为报文创建新的事务处理进程。进程将原始报文作为参数传给事务处理进程。事务处理进程将开始名字路由处理过程。

高速 DNS 缓存查找进程相对比较简单。由于高速缓存有限，因此表的大小受到系统资源的限制。

为了能够在查找性能上获得较大的提高,缓存的命中率就需要有一定的保证。假设缓存查找速度是查找速度的 20 倍,经过计算可以得出,为了能够使平均查找性能达到原来的 10 倍,缓存的命中率需要达到 95%左右。随着 Internet 的不断发展网络用户的不断增加以及业务数据量的多样化，数据的时间局部性变得越来越不明显，从而大大地降低了缓存的命中率。因此，缓存的大小应该能够随着网络用户或者网络数据业务量的增加而相应地呈线性增加。

○ 高速缓存表结构

Name	IP Addrsss
Zju.edu.cn	1.2.3.4
Cnn.com.cn	5.6.7.8

表 2.1　高速缓存域名解析表

表 2.1 给出了高速缓存域名解析表。域名解析表由域名和域名对应的 IP 解析地址。表中，Name 代表解析域名，查找算法通过报文解析域名与 Name 项的匹配程度，确定是否名

中该拦；IP Adderss 代表了与解析域名对应的 IP 地址。关于域名的匹配原则，将在 2.4 节描述。

○ 高速缓存表的查找算法

前面提到，为了提高查找的性能，我们在缓存的性能上做了一定的保证。缓存的大小随着网络用户或者网络数据业务量的增加而相应地呈线性增加。在现有条件下，如缓存大小一定，CPU 等资源一定，我们如果提高查找的性能呢？采用合理高效的表搜索算法是提高查找性能的一个有效途径。

2.4 节将围绕表查找算法进行详细的讨论。高速缓存表的查找算法将采用 HASH 算法。关于 HASH 算法可以 参见 2.4 节。

○ 高速缓存表的建立

上面简单讨论了缓存表的结构和缓存表的表查找算法，但是我们并没有说明系统是如何初始化缓存表以及如何更新缓存表的。这里将对此进行简单的讨论。

1 缓存表的初始化。一般来说，名字路由模块启动时，缓存表示空的。但是，有两种途径可以初始化缓存表：

a. ROM 读取。如果主机配有可存储的 ROM，缓存表将写到 ROM 中储存起来。当下一次名字路由模块启动时，将从 ROM 中读取上一次存储的缓存表。

b. 通过网管、或者 TFTP 等网络接口手动或者自动初始化缓存表。系统不定期的把当前的缓存表，通过网管或者 TFTP 等网络接口存储到网管服务器、TFTP 服务器。当名字路由模块启动时，将通过网络接口下载缓存列表，初始化主机缓存表。网管也可以手动添加表项，进行缓存表的初始化。

2 缓存表的更新。 缓存表的内容应该跟随网络的频繁交互而动态更新。更新的过程是和报文的事务处理过程紧密联系的。

当查询报文直接命中高速缓存，毫无疑问，不需要进行缓存表的更新。缓存的更新发生在查询报文高速缓存命中失败的情况下。由于命中失败，系统将为报文创建一个新的事务处理进程。事务处理进程进行接力跳方式的名字路由。进程在本地等待响应报文。当响应报文到达后，事务处理进程将进行一系列的操作。其中一个操作就是缓存表的更新。进程根据响应报文，形成一对域名与 IP 地址的绑定映射。这一对映射将被添加到缓存表中。我们在图 2.11 给出了算法的描述。

首先，根据域名对缓存表进行 HASH 查找。如果找到，说明系统出错，系统将向网管报错。如果没有找到，就进行下一步的操作。缓存表采用 FIFO 的机制。进程判断当前表长度是否以及超过表的额定长度。如果没有超过，那么将该对映射插入该 HASH 分支的队列尾部；否则将删除 HASH 表头的数据，并把新的映射插入对应 HASH 分支的队列尾部。【参见 2.4 节】

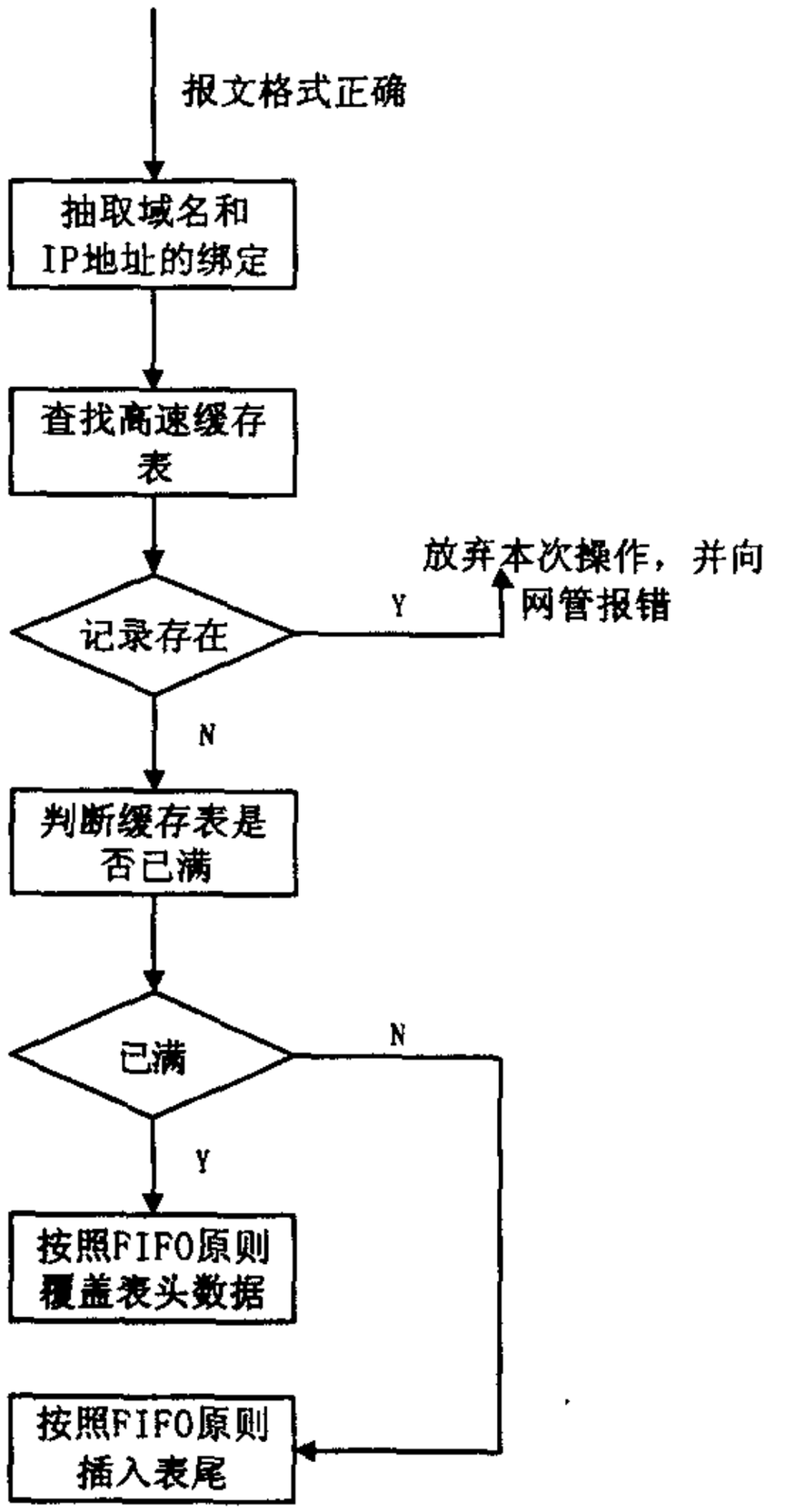


图 2.11　DNS 缓存表算法描述

2.3.3　事务处理进程

由于主机的高速 DNS 缓存没有查询报文的响应记录，报文处理进程将为报文创建新的事务处理进程。事务处理进程采用现行 IP 分布式路由的路由机制，进行报文的名字路由。通过一个或多个名字路由器的协同交互合作，完成整个名字路由的过程。

一个事务处理进程由一个查询报文和对应的响应报文（一个或者多个）组成。事务处理进程可以分为客户端（Client）和服务器端（Server）。客户端发出查询报文，服务器端发出响应报文。客户端和服务器端是按照逻辑功能对事务处理进程的划分。实际上，客户端和服务器端的功能都嵌入在事务处理进程模块中。考虑下图 2.12 的事务例图。在例图里，UAC(USER AGENT CLIENT)代表了发起请求报文的用户端，PROXY 执行了名字路由的事务处理进程，UAS(USER AGENT SERVER)代表了送出最终响应的服务器端。UAS 通常是内容服务器。如果查询报文在名字路由器的高速缓存命中，那么该名字路由器将被认为是一个 UAS。从例图中可以看到，执行事务处理进程的 PROXY 包括了客户端事务处理（CLIEN TRANSACTION）和服务器端事务(SERVER TRANSACTION)处理。

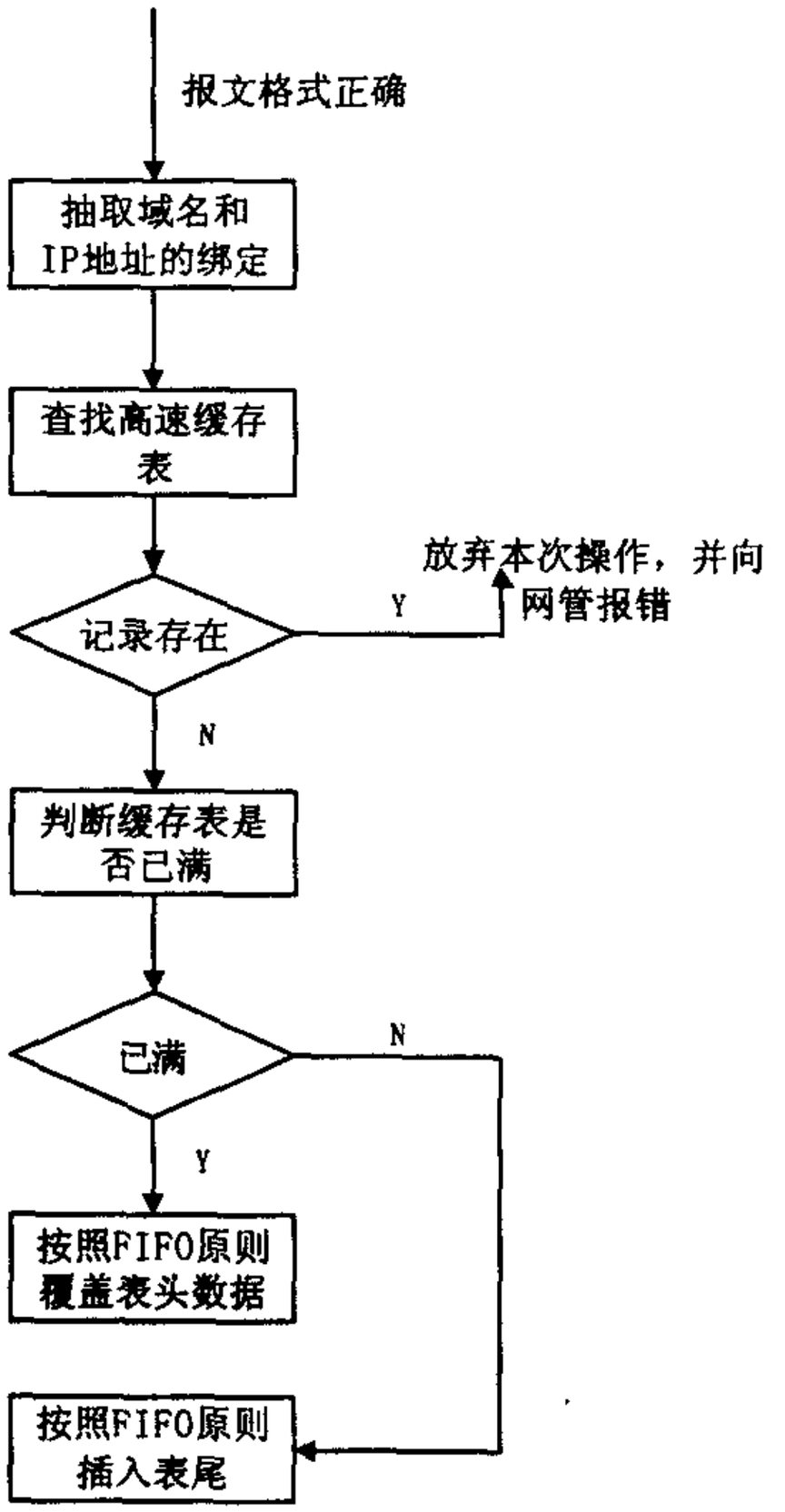


图 2.11　DNS 缓存表算法描述

2.3.3　事务处理进程

由于主机的高速 DNS 缓存没有查询报文的响应记录，报文处理进程将为报文创建新的事务处理进程。事务处理进程采用现行 IP 分布式路由的路由机制，进行报文的名字路由。通过一个或多个名字路由器的协同交互合作，完成整个名字路由的过程。

一个事务处理进程由一个查询报文和对应的响应报文（一个或者多个）组成。事务处理进程可以分为客户端（Client）和服务器端（Server）。客户端发出查询报文，服务器端发出响应报文。客户端和服务器端是按照逻辑功能对事务处理进程的划分。实际上，客户端和服务器端的功能都嵌入在事务处理进程模块中。考虑下图 2.12 的事务例图。在例图里，UAC(USER AGENT CLIENT)代表了发起请求报文的用户端，PROXY 执行了名字路由的事务处理进程，UAS(USER AGENT SERVER)代表了送出最终响应的服务器端。UAS 通常是内容服务器。如果查询报文在名字路由器的高速缓存命中，那么该名字路由器将被认为是一个 UAS。从例图中可以看到，执行事务处理进程的 PROXY 包括了客户端事务处理（CLIEN TRANSACTION）和服务器端事务(SERVER TRANSACTION)处理。

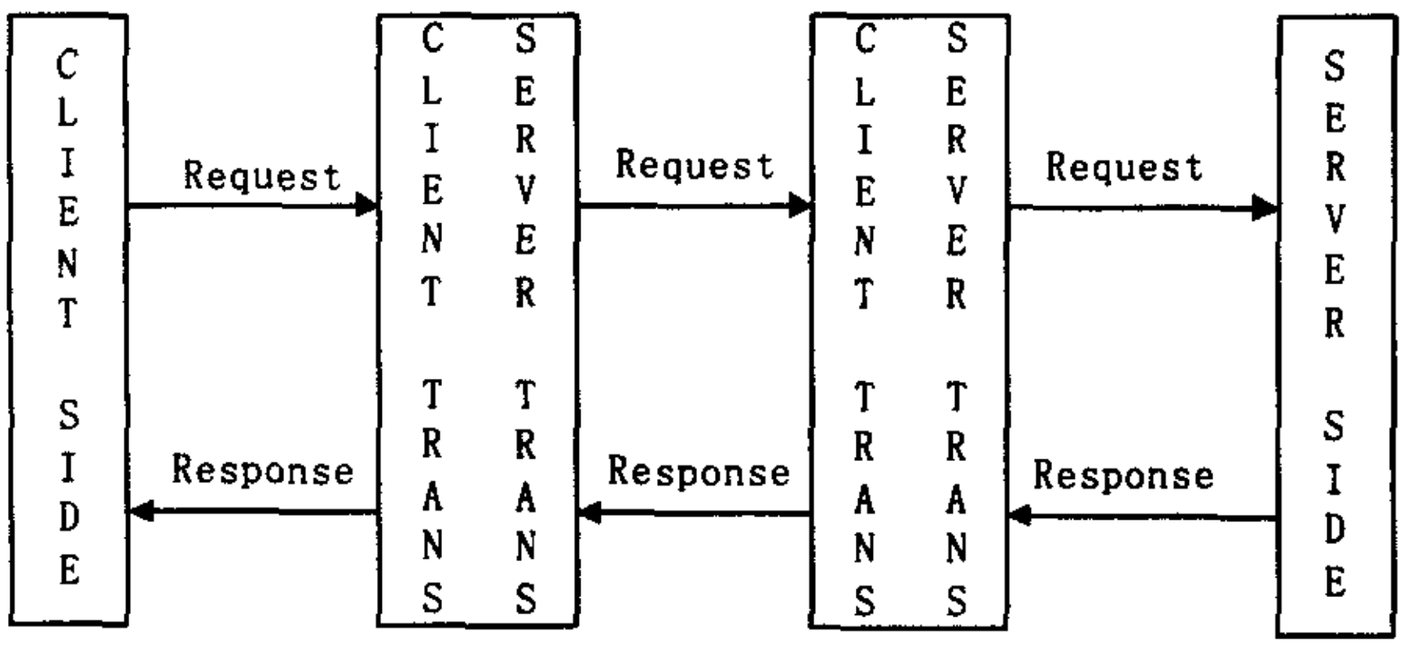


图 2.13　事务例图

客户端事务处理负责对查询报文（Request）的名字路由处理。它根据报文的查询域名进行名字路由表的表驱动路由。当它找到最佳的下一跳，就将报文转发到下一跳。

这个事务处理的过程类似与 IP 路由的分布式路由。与 IP 路由不同的是，名字路由是根据查询域名来进行表驱动路由的，而 IP 路由是以 IP 地址的网络前缀来进行表驱动路由的。从层次来说，IP 路由是名字路由的基础，名字路由依赖于 IP 路由。

DNS 与 INRP 的主要区别也体现在客户端事务处理过程。INRP 采用了分布式的“下一跳”路由机制。DNS 采用中心服务器的体制。在 DNS 中，如果高速缓存没有相应的记录，本地 DNS 会把查询报文发送给它的上一级 DNS 服务器，由上一级的 DNS 服务器处理该报文。

服务器端事务处理负责对响应报文的接收和处理。当它收到对应的响应报文后，首先会检查报文各个数据字段的正确性。通过检查后，报文将传递给响应的的客户端事务处理。客户端事务处理根据查询报文，抽取发送方的 IP 地址和端口号。进程将向发送方回送响应报文。事务处理进程将被结束。

虽然我们可以从逻辑功能上把事务处理进程划分成两个部分：客户端事务处理和服务器端事务处理。但是，在实际的事务处理进程中，它们是紧密联系，不可分割的。我们很难在下面的进程状态机划分它们。

※　进程状态机

图 2.14 给出了事务处理进程的状态机描述图。事务处理进程由六个状态组成。我们将分别给出详细的描述。

初始状态（INIT）指的是事务处理进程刚被创建时所处的状态。当查询报文没有命中 DNS 高速缓存时，报文处理进程将创建事务处理进程。事务处理进程一开始总是处于初始状态（INIT）。在这个状态期间，我们需要启动定时器 A。定时器 A 是整个事务处理进程的最长寿命时间。当 A 超时后，事务处理进程必须被终止。定时器 A 的时间片长度要和 DNS 客户端的生存时间保持一致，一般是 1800ms。

当报文处理进程将查询报文传递给事务处理进程后，事务处理进程将由初始状态（INIT）转入路由状态（Routing）。本状态负责为报文选择最佳的下一跳。这个过程基于名字路由的表驱动选路算法。我们将在 2.4 节描述名字路由的表驱动算法。如果选路时发生异常，或者寿命定时器 A 超时，将转入进程终止（Terminated）状态，终止该事务处理进程。

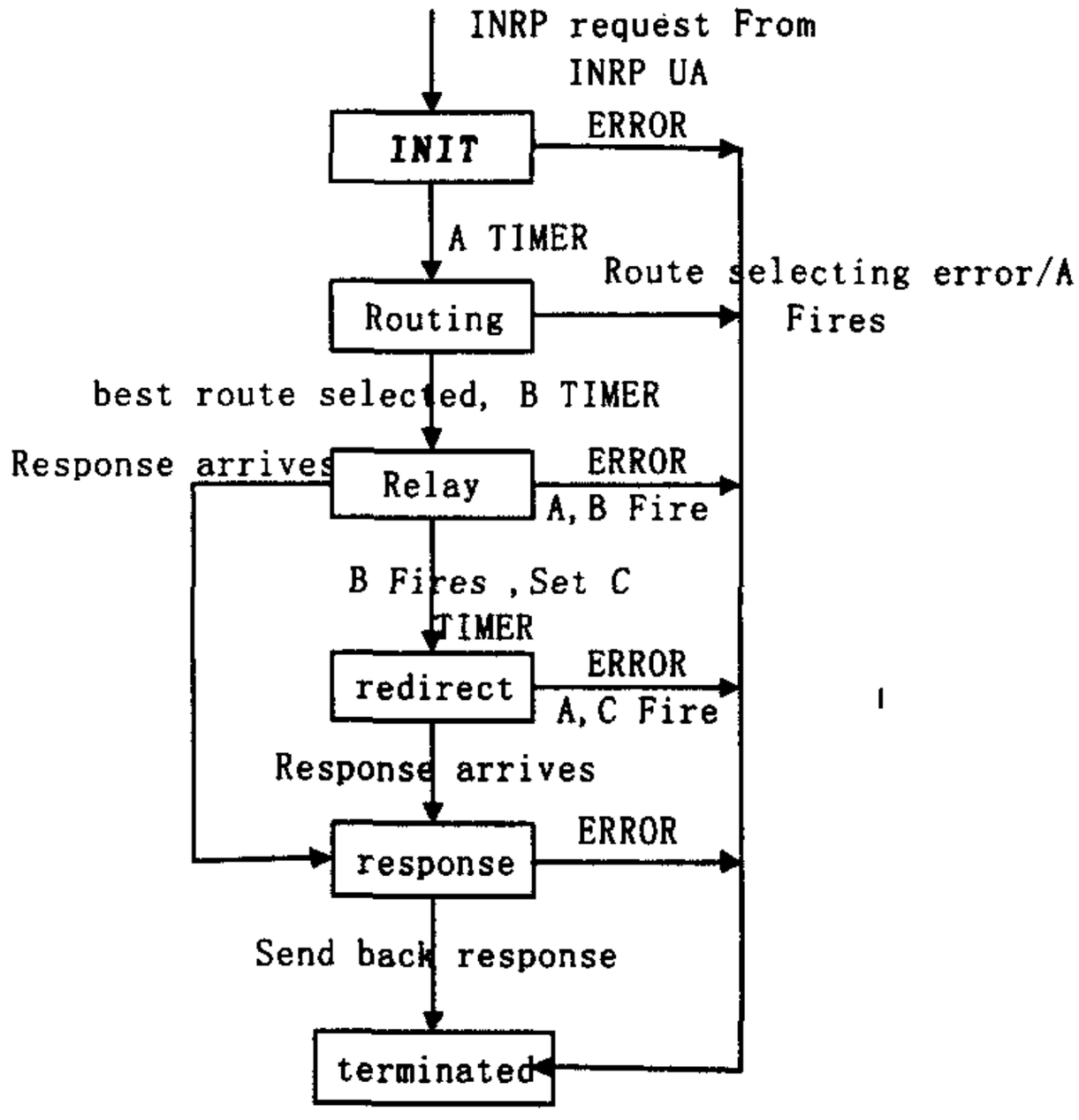


图 2.14　事务处理进程的状态机描述图

路由状态将为查询报文选择出最佳的下一跳。事务处理进程的状态也从由路由状态（Routing）进入报文转发状态（Relay）。报文转发状态负责将报文发往下一跳，主机同时等待响应报文。定时器 B 将被启动。设置定时器 B 的目的是为了保证网络的健壮。响应报文必须在寿命定时器 B 超时前到达；如果在定时器 B 的寿命内，主机没有收到响应报文，主机就可以认为下一跳不可达。必须选择新的下一跳。这种情况可以类比与 DNS 的重定向。如果主机在定时器 B 超时前收到报文，将转入报文响应状态（Response）。否则，进入重定向状态（Redirect）。定时器 B 的时间片长度为 1/3 A。

在定时器 B 超时的情况下，事务处理进程进入重定向状态。进程必须为报文重新选择一个下一跳。我们需要引入重定向机制来解决这个问题。这将在下面进行讨论。通过重定向机制，进程向新的下一跳转发查询报文，并启动定时器。定时器 C 的目的和 B 一样，都是为了确保网络的健壮。定时器 C 的时间片长度同样为 1/3A。如果进程在定时器 C 超时前收到报文，将转入报文响应状态（Response）。如果定时器 C 超时，进程将认为该域名不可解析，进程将直接进入终止状态（Terminated）。

报文响应状态（Response）主要负责响应报文的处理和回送。这个状态有三个动作：

1　首先要检查报文的各个数据字段的正确性。如果数据段格式错误，响应报文将被丢弃。

2　如果报文格式正确，将抽取域名和目的 IP 的绑定映射。该队映射将添加到 DNS 高速缓存里，DNS 高速缓存表将被更新。DNS 高速缓存表的建立更新算法在 2.3.2 节中已经进行了描述。图 2.12 给出了算法描述图。

3　响应报文将向发送查询报文的 IP 地址和端口的发送。

进程终止状态（Terminated）是进程的最终状态。进程在此期间，将释放所有的资源，包括内容资源、关闭定时器等。事务处理进程被终止。空闲的资源将为新的事务进程服务。

※ 重定向机制

重定向机制为了在路由故障导致寿命定时器超时引入的路由机制。路由保障包括目的机器临时或永久故障、中间路由器被拥塞无法处理传入的通信业务。在定时器 B 超时的情况下，事务处理进程进入重定向状态。进程必须为报文重新选择一个下一跳。

在 INRP 刷新本地名字路由表，将保留原先的到达目的内容的最佳路由。也就是说，本地路由表至少需要为同一目的内容网络保留两条路由表项。否则，重定向机制将采用默认路由方式。INRP 在 Routing 状态选用本地最佳路由，当定时器 B 超时，INRP 将选用次佳路由或者直接用默认路由。

$ 2.4 名字路由的表驱动算法

IP 路由基与 IP 路由表驱动算法，名字路由同样基于表驱动算法。不同的是，名字路由表基于报文的查询域名的内容前缀，IP 路由表基于 IP 地址的网络前缀。名字路由表借鉴了 IP 路由表下一跳的概念。

2.4.1 名字路由表的表结构

在基于名字的路由体系中，用户想要连接的不再是一个纯粹的 IP 终端或者是服务器，而是特定的内容。内容以内容名字表示，通常是我们常见的 URL。IP 地址在名字路由的模式下，已经不再标志一个网络终端。取而代之的是含义具体的象 URL 这样的内容名字。因特网上传播的路由信息除了 BGP 的 IP 地址前缀表示的路由信息，就是以名字后缀表示的内容路由信息。

由于名字路由借鉴了 IP 路由借鉴了接力跳的路由思想。因此，名字路由表和 IP 路由表有很大的相似性。表 2.2 给出了斯坦福公布的名字路由表：

Name	Next-hop	Path	Server
Cnn.com	4.3.2.1	10 ms	5ms
Zju.edu.cn	8.7.6.8	65ms	10ms

表 2.2 斯坦福公布的名字路由表

路由表的各个表项包含下列信息：

● 目的端内容服务器域名（Name）。一个内容服务器有一个内容域名，一个内容域名可以代表网络上相同域名的内容服务器。这些内容服务器在不同的地理位置为本地用户提供相同的内容供给服务。它们的区别在于各自的 IP 地址。

● “下一跳”路由器（Next_hop router）的 IP 地址。“下一跳”路由器位于直接相连的网络上，且通过它可以发送 DNS 数据包，它并不是最终的目的地。但它可以将传给它的数据报送到最终目的。

● 网络耗费时间（PATH）。它表示到达最终内容服务器的时间。网络时间耗费根据网络的拥塞度、路由器的处理能力而发生变化。度量尺度为毫秒单位。

● 内容服务器时间耗费（Server）。内容将根据到达的查询报文的域名进行本地内容的检索，确认用户要求服务的内容在本地服务器里。否则，将丢弃报文，拒绝服务。这一段处理的时间耗费将体现在 Server 项上。度量尺度为毫秒单位。

从表中可以看出，路由的耗费为网络耗费与内容服务器耗费之和。我们将根据最小的路由耗费选择最佳的下一跳。考虑到方便实现以及路由表的直观，我们将网络耗费与内容服务器的耗费合并，用距离（Metric）来表示。度量标准是到达目的地的距离。最小的距离（Metric）表示最佳的路由。因此，改进后的名字路由表如表 2.3 所示。

※ 重定向机制

重定向机制为了在路由故障导致寿命定时器超时引入的路由机制。路由保障包括目的机器临时或永久故障、中间路由器被拥塞无法处理传入的通信业务。在定时器 B 超时的情况下，事务处理进程进入重定向状态。进程必须为报文重新选择一个下一跳。

在 INRP 刷新本地名字路由表，将保留原先的到达目的内容的最佳路由。也就是说，本地路由表至少需要为同一目的内容网络保留两条路由表项。否则，重定向机制将采用默认路由方式。INRP 在 Routing 状态选用本地最佳路由，当定时器 B 超时，INRP 将选用次佳路由或者直接用默认路由。

$ 2.4 名字路由的表驱动算法

IP 路由基与 IP 路由表驱动算法，名字路由同样基于表驱动算法。不同的是，名字路由表基于报文的查询域名的内容前缀，IP 路由表基于 IP 地址的网络前缀。名字路由表借鉴了 IP 路由表下一跳的概念。

2.4.1 名字路由表的表结构

在基于名字的路由体系中，用户想要连接的不再是一个纯粹的 IP 终端或者是服务器，而是特定的内容。内容以内容名字表示，通常是我们常见的 URL。IP 地址在名字路由的模式下，已经不再标志一个网络终端。取而代之的是含义具体的象 URL 这样的内容名字。因特网上传播的路由信息除了 BGP 的 IP 地址前缀表示的路由信息，就是以名字后缀表示的内容路由信息。

由于名字路由借鉴了 IP 路由借鉴了接力跳的路由思想。因此，名字路由表和 IP 路由表有很大的相似性。表 2.2 给出了斯坦福公布的名字路由表：

Name	Next-hop	Path	Server
Cnn.com	4.3.2.1	10 ms	5ms
Zju.edu.cn	8.7.6.8	65ms	10ms

表 2.2 斯坦福公布的名字路由表

路由表的各个表项包含下列信息：

● 目的端内容服务器域名（Name）。一个内容服务器有一个内容域名，一个内容域名可以代表网络上相同域名的内容服务器。这些内容服务器在不同的地理位置为本地用户提供相同的内容供给服务。它们的区别在于各自的 IP 地址。

● “下一跳”路由器（Next_hop router）的 IP 地址。“下一跳”路由器位于直接相连的网络上，且通过它可以发送 DNS 数据包，它并不是最终的目的地。但它可以将传给它的数据报送到最终目的。

● 网络耗费时间（PATH）。它表示到达最终内容服务器的时间。网络时间耗费根据网络的拥塞度、路由器的处理能力而发生变化。度量尺度为毫秒单位。

● 内容服务器时间耗费（Server）。内容将根据到达的查询报文的域名进行本地内容的检索，确认用户要求服务的内容在本地服务器里。否则，将丢弃报文，拒绝服务。这一段处理的时间耗费将体现在 Server 项上。度量尺度为毫秒单位。

从表中可以看出，路由的耗费为网络耗费与内容服务器耗费之和。我们将根据最小的路由耗费选择最佳的下一跳。考虑到方便实现以及路由表的直观，我们将网络耗费与内容服务器的耗费合并，用距离（Metric）来表示。度量标准是到达目的地的距离。最小的距离（Metric）表示最佳的路由。因此，改进后的名字路由表如表 2.3 所示。

※ 重定向机制

重定向机制为了在路由故障导致寿命定时器超时引入的路由机制。路由保障包括目的机器临时或永久故障、中间路由器被拥塞无法处理传入的通信业务。在定时器 B 超时的情况下，事务处理进程进入重定向状态。进程必须为报文重新选择一个下一跳。

在 INRP 刷新本地名字路由表，将保留原先的到达目的内容的最佳路由。也就是说，本地路由表至少需要为同一目的内容网络保留两条路由表项。否则，重定向机制将采用默认路由方式。INRP 在 Routing 状态选用本地最佳路由，当定时器 B 超时，INRP 将选用次佳路由或者直接用默认路由。

$ 2.4 名字路由的表驱动算法

IP 路由基与 IP 路由表驱动算法，名字路由同样基于表驱动算法。不同的是，名字路由表基于报文的查询域名的内容前缀，IP 路由表基于 IP 地址的网络前缀。名字路由表借鉴了 IP 路由表下一跳的概念。

2.4.1 名字路由表的表结构

在基于名字的路由体系中，用户想要连接的不再是一个纯粹的 IP 终端或者是服务器，而是特定的内容。内容以内容名字表示，通常是我们常见的 URL。IP 地址在名字路由的模式下，已经不再标志一个网络终端。取而代之的是含义具体的象 URL 这样的内容名字。因特网上传播的路由信息除了 BGP 的 IP 地址前缀表示的路由信息，就是以名字后缀表示的内容路由信息。

由于名字路由借鉴了 IP 路由借鉴了接力跳的路由思想。因此，名字路由表和 IP 路由表有很大的相似性。表 2.2 给出了斯坦福公布的名字路由表：

Name	Next-hop	Path	Server
Cnn.com	4.3.2.1	10 ms	5ms
Zju.edu.cn	8.7.6.8	65ms	10ms

表 2.2 斯坦福公布的名字路由表

路由表的各个表项包含下列信息：

● 目的端内容服务器域名（Name）。一个内容服务器有一个内容域名，一个内容域名可以代表网络上相同域名的内容服务器。这些内容服务器在不同的地理位置为本地用户提供相同的内容供给服务。它们的区别在于各自的 IP 地址。

● “下一跳”路由器（Next_hop router）的 IP 地址。“下一跳”路由器位于直接相连的网络上，且通过它可以发送 DNS 数据包，它并不是最终的目的地。但它可以将传给它的数据报送到最终目的。

● 网络耗费时间（PATH）。它表示到达最终内容服务器的时间。网络时间耗费根据网络的拥塞度、路由器的处理能力而发生变化。度量尺度为毫秒单位。

● 内容服务器时间耗费（Server）。内容将根据到达的查询报文的域名进行本地内容的检索，确认用户要求服务的内容在本地服务器里。否则，将丢弃报文，拒绝服务。这一段处理的时间耗费将体现在 Server 项上。度量尺度为毫秒单位。

从表中可以看出，路由的耗费为网络耗费与内容服务器耗费之和。我们将根据最小的路由耗费选择最佳的下一跳。考虑到方便实现以及路由表的直观，我们将网络耗费与内容服务器的耗费合并，用距离（Metric）来表示。度量标准是到达目的地的距离。最小的距离（Metric）表示最佳的路由。因此，改进后的名字路由表如表 2.3 所示。

Name	Next Hop	Metric
Cnn.com	4.3.2.1	2
Zju.edu.cn	8.7.6.8	7

图 2.3　实际采用的名字路由表

名字路由表体现了名字路由的思想。名字路由器将根据域名的匹配程度选择最佳的下一跳。查询报文的域名将与路由表中的目的内容服务器的域名（Name）进行最长后缀匹配。而在 IP 路由中，路由机制是根据 IP 网络的前缀进行最长前缀匹配。

2.4.2　名字路由表的匹配方式

由于名字路由表的关键字是目的内容服务器的域名（Name）。IP 路由的最长前缀匹配方式将不适应名字路由方式。我们需要提出一种新的匹配方式。

名字路由表中的关键字是具有等级结构的域名。DNS的名字空间和Unix的文件系统相似，也具有层次结构。图2.15 显示了这种层次的组织形式。

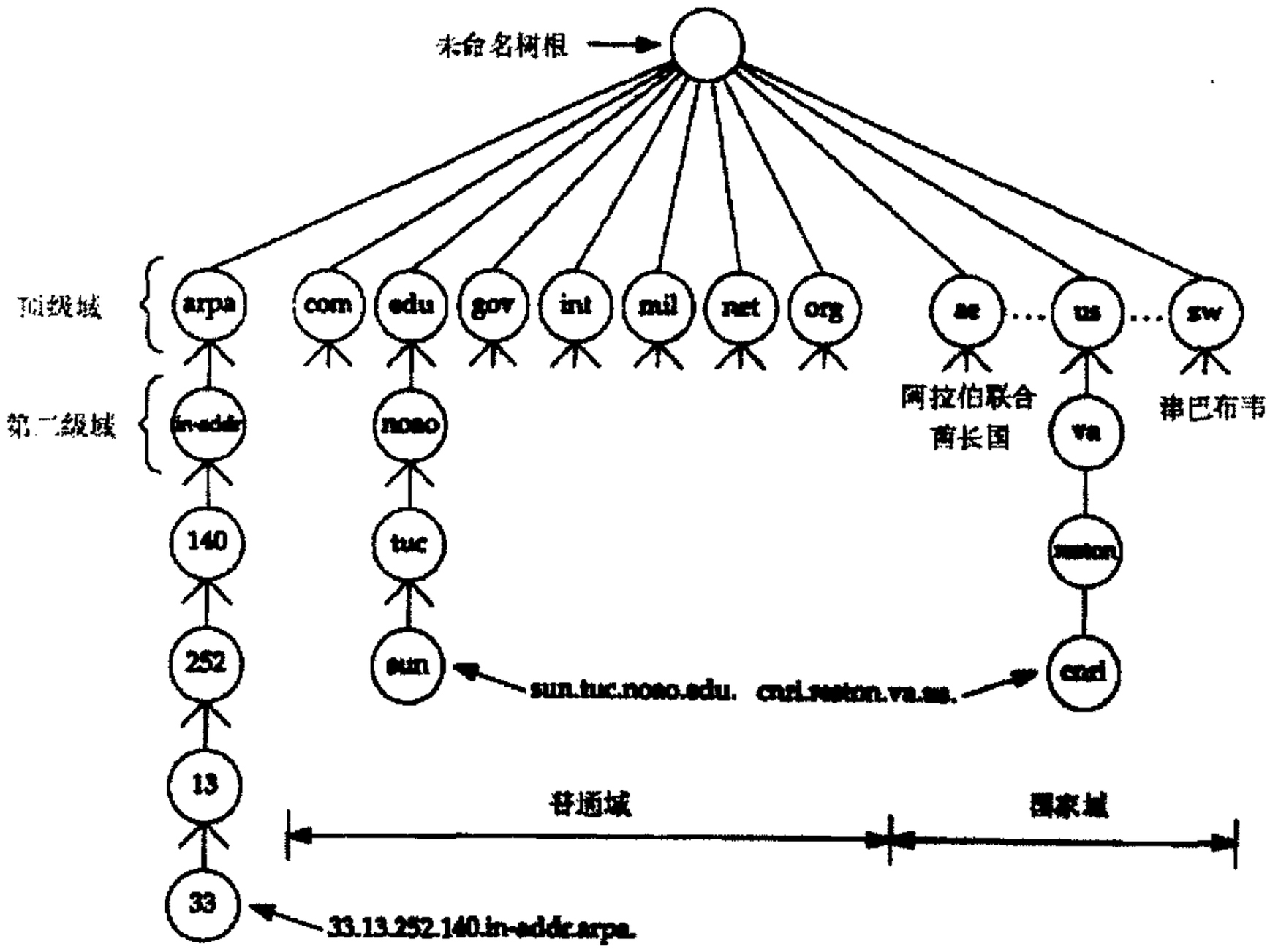


图2.15　DNS的层次组织

每个结点（图2.15中的圆圈）有一个至多63个字符长的标识。这颗树的树根是没有任何标识的特殊结点。命名标识中一律不区分大写和小写。命名树上任何一个结点的域名就是将从该结点到最高层的域名串连起来，中间使用一个点“.”分隔这些域名（注意这和Unix文件系统路径的形成不同，文件路径是由树根依次向下的形成的）。

最长后缀匹配方式适应了DNS采用从子域到父域的串连方式。用户提交的查询域名和名字路由表中的域名比较顺序是：顶级域名->二级域名->三级域名->……。

举例说明：

1）　比较查询域名grs. zju. edu. cn和路由表域名zju. edu. cn。由于顶级域名、二级域名、三级域名都匹配，grs. zju. edu. cn匹配与zju. edu. cn，匹配的结果是zju. edu. cn。

Name	Next Hop	Metric
Cnn.com	4.3.2.1	2
Zju.edu.cn	8.7.6.8	7

图 2.3　实际采用的名字路由表

名字路由表体现了名字路由的思想。名字路由器将根据域名的匹配程度选择最佳的下一跳。查询报文的域名将与路由表中的目的内容服务器的域名（Name）进行最长后缀匹配。而在 IP 路由中，路由机制是根据 IP 网络的前缀进行最长前缀匹配。

2.4.2　名字路由表的匹配方式

由于名字路由表的关键字是目的内容服务器的域名（Name）。IP 路由的最长前缀匹配方式将不适应名字路由方式。我们需要提出一种新的匹配方式。

名字路由表中的关键字是具有等级结构的域名。DNS的名字空间和Unix的文件系统相似，也具有层次结构。图2.15 显示了这种层次的组织形式。

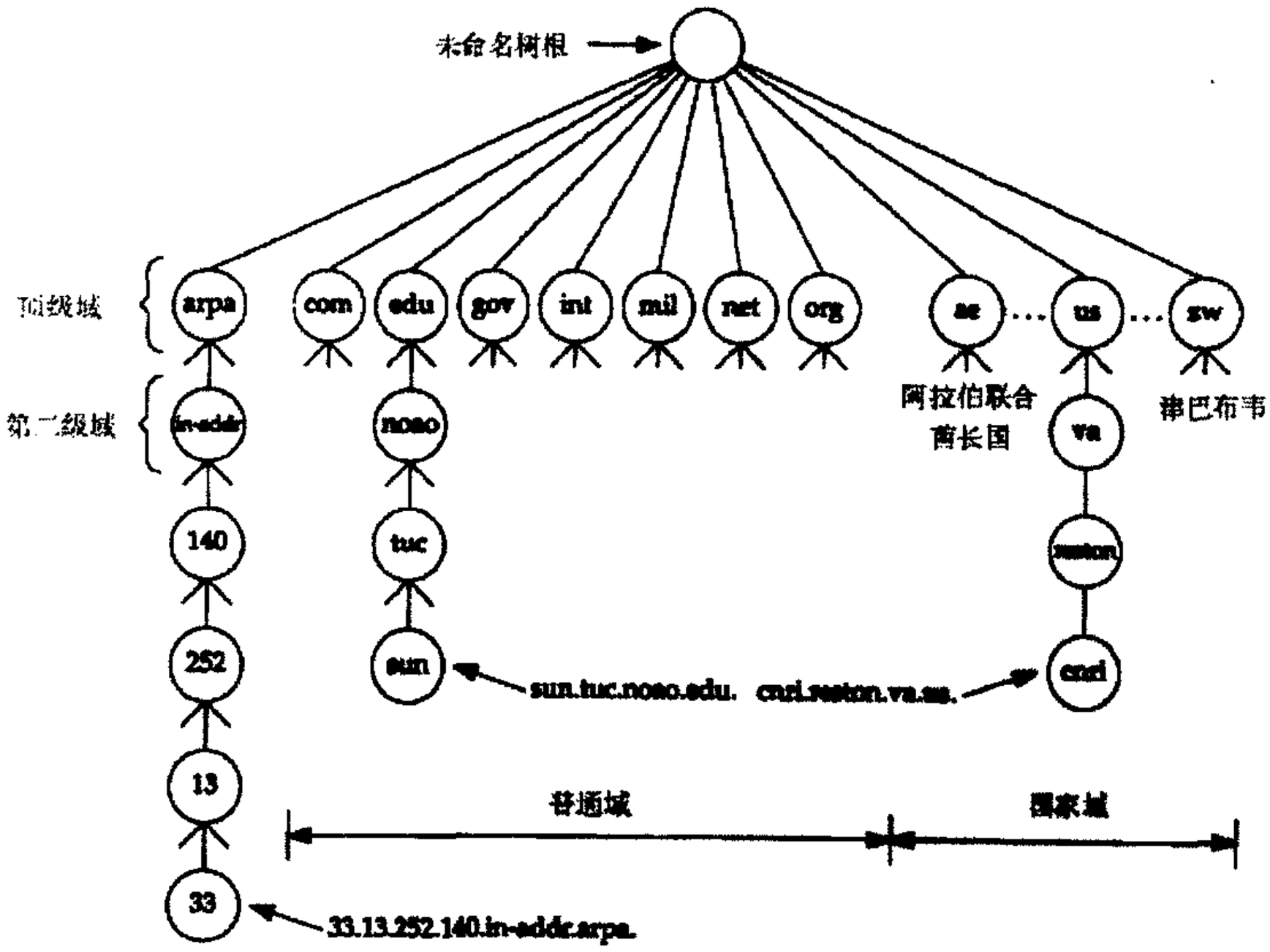


图2.15　DNS的层次组织

每个结点（图2.15中的圆圈）有一个至多63个字符长的标识。这颗树的树根是没有任何标识的特殊结点。命名标识中一律不区分大写和小写。命名树上任何一个结点的域名就是将从该结点到最高层的域名串连起来，中间使用一个点“.”分隔这些域名（注意这和Unix文件系统路径的形成不同，文件路径是由树根依次向下的形成的）。

最长后缀匹配方式适应了DNS采用从子域到父域的串连方式。用户提交的查询域名和名字路由表中的域名比较顺序是：顶级域名->二级域名->三级域名->……。

举例说明：

1）　比较查询域名grs. zju. edu. cn和路由表域名zju. edu. cn。由于顶级域名、二级域名、三级域名都匹配，grs. zju. edu. cn匹配与zju. edu. cn，匹配的结果是zju. edu. cn。

2）比较查询域名grs. sjtu. edu. cn和路由表域名zju. edu. cn。顶机域名和二级域名匹配，三级域名不匹配，匹配的结果是edu. cn，我们认为它们不匹配。

接下来我们讨论网络信息隐藏的问题。如果路由表存有所有可能的目的地域名信息，那么它不可能使路由表总是符合内容服务网络的当前情况。此外，因为可能的目的内容服务器的数量很大，所以机器没有足够的空间来储存信息。

因此我们采用信息隐藏的原理，使机器用最少的信息来进行路由选择。DNS域名分级结构帮我们达到了这个目标。例如，grs.zju.edu.cn和zdzsc.zju.edu.cn是从属于父域zju.edu.cn。路由表仅仅需要包含父域的信息而不需要包含它的各个子域的信息。

2.4.3 默认名字路由

默认路由是另一种用来隐藏信息、保持选路表很小的技术。它把多个表项统一到一个默认的情况中。它使选路进程首先在选路表中查找目的网络。如果表中没有路由，则选路进程就把数据报发送到默认名字路由器上。

当名字路由网络的本地内容服务器数目很小，并且与其他内容服务器网络只有一个连接时，默认的路由选择尤其有用。

2.4.4 名字路由的表驱动选路算法

名字路由是按照“接力跳”（hop-by-hop）完成的。从路由表可以可出，名字路由并不知道到达任何目的地的完整路径（除非目的内容服务器直接和发送主机相连），名字路由所能提供的仅是该 DNS 数据报将送达下一“跳”路由器的 IP 地址。它假定下一“跳”路由器离目的端要比发送主机来得近，而且下一“跳”路由器是和发送主机直接相连的。

名字路由执行下列操作：

1. 从数据报中抽取目的内容域名 S;
2. If S 与任何直接相连的内容服务器的域名匹配，
   then 把数据报投递到目的内容服务器上;
3. else if 路由表中包含了一个到 S 的指定路由，
   then 把数据报投递到表中指定的下一跳;
4. else if 路由表中包含了到 S 的父域的一个路由，
   then 把数据报投递到表中指定的下一跳;
5. else if 表中包含一个默认路由，
   then 把数据报发送动表中指定的默认路由上
6. else 宣布路由出错

§ 2.5 名字路由的表搜索算法

名字路由器的主要功能是按照 DNS 分组中的目的内容域名转发分组。查找路由表决定将分组发往哪个下一跳，这是转发分组过程中的重要一步。因此,快速的路由表查找算法是实现高速分组转发的关键。为了提高路由查找的性能，本节对多种路由查找算法进行了讨论，分析了路由查找问题和实现难点，并对它们进行了详细的分析和比较。最后确立了名字路由表的查找算法，并给出了具体实现。

2.5.1 路由查找算法的分类

最长后缀匹配查找的难点在于，在查找过程中不仅需要与内容后缀的比特值进行匹配查

2）比较查询域名grs. sjtu. edu. cn和路由表域名zju. edu. cn。顶机域名和二级域名匹配，三级域名不匹配，匹配的结果是edu. cn，我们认为它们不匹配。

接下来我们讨论网络信息隐藏的问题。如果路由表存有所有可能的目的地域名信息，那么它不可能使路由表总是符合内容服务网络的当前情况。此外，因为可能的目的内容服务器的数量很大，所以机器没有足够的空间来储存信息。

因此我们采用信息隐藏的原理，使机器用最少的信息来进行路由选择。DNS域名分级结构帮我们达到了这个目标。例如，grs.zju.edu.cn和zdzsc.zju.edu.cn是从属于父域zju.edu.cn。路由表仅仅需要包含父域的信息而不需要包含它的各个子域的信息。

2.4.3 默认名字路由

默认路由是另一种用来隐藏信息、保持选路表很小的技术。它把多个表项统一到一个默认的情况中。它使选路进程首先在选路表中查找目的网络。如果表中没有路由，则选路进程就把数据报发送到默认名字路由器上。

当名字路由网络的本地内容服务器数目很小，并且与其他内容服务器网络只有一个连接时，默认的路由选择尤其有用。

2.4.4 名字路由的表驱动选路算法

名字路由是按照“接力跳”（hop-by-hop）完成的。从路由表可以可出，名字路由并不知道到达任何目的地的完整路径（除非目的内容服务器直接和发送主机相连），名字路由所能提供的仅是该 DNS 数据报将送达下一“跳”路由器的 IP 地址。它假定下一“跳”路由器离目的端要比发送主机来得近，而且下一“跳”路由器是和发送主机直接相连的。

名字路由执行下列操作：

1. 从数据报中抽取目的内容域名 S;
2. If S 与任何直接相连的内容服务器的域名匹配，
   then 把数据报投递到目的内容服务器上;
3. else if 路由表中包含了一个到 S 的指定路由，
   then 把数据报投递到表中指定的下一跳;
4. else if 路由表中包含了到 S 的父域的一个路由，
   then 把数据报投递到表中指定的下一跳;
5. else if 表中包含一个默认路由，
   then 把数据报发送动表中指定的默认路由上
6. else 宣布路由出错

§ 2.5 名字路由的表搜索算法

名字路由器的主要功能是按照 DNS 分组中的目的内容域名转发分组。查找路由表决定将分组发往哪个下一跳，这是转发分组过程中的重要一步。因此,快速的路由表查找算法是实现高速分组转发的关键。为了提高路由查找的性能，本节对多种路由查找算法进行了讨论，分析了路由查找问题和实现难点，并对它们进行了详细的分析和比较。最后确立了名字路由表的查找算法，并给出了具体实现。

2.5.1 路由查找算法的分类

最长后缀匹配查找的难点在于，在查找过程中不仅需要与内容后缀的比特值进行匹配查

2）比较查询域名grs. sjtu. edu. cn和路由表域名zju. edu. cn。顶机域名和二级域名匹配，三级域名不匹配，匹配的结果是edu. cn，我们认为它们不匹配。

接下来我们讨论网络信息隐藏的问题。如果路由表存有所有可能的目的地域名信息，那么它不可能使路由表总是符合内容服务网络的当前情况。此外，因为可能的目的内容服务器的数量很大，所以机器没有足够的空间来储存信息。

因此我们采用信息隐藏的原理，使机器用最少的信息来进行路由选择。DNS域名分级结构帮我们达到了这个目标。例如，grs.zju.edu.cn和zdzsc.zju.edu.cn是从属于父域zju.edu.cn。路由表仅仅需要包含父域的信息而不需要包含它的各个子域的信息。

2.4.3 默认名字路由

默认路由是另一种用来隐藏信息、保持选路表很小的技术。它把多个表项统一到一个默认的情况中。它使选路进程首先在选路表中查找目的网络。如果表中没有路由，则选路进程就把数据报发送到默认名字路由器上。

当名字路由网络的本地内容服务器数目很小，并且与其他内容服务器网络只有一个连接时，默认的路由选择尤其有用。

2.4.4 名字路由的表驱动选路算法

名字路由是按照“接力跳”（hop-by-hop）完成的。从路由表可以可出，名字路由并不知道到达任何目的地的完整路径（除非目的内容服务器直接和发送主机相连），名字路由所能提供的仅是该 DNS 数据报将送达下一“跳”路由器的 IP 地址。它假定下一“跳”路由器离目的端要比发送主机来得近，而且下一“跳”路由器是和发送主机直接相连的。

名字路由执行下列操作：

1. 从数据报中抽取目的内容域名 S;
2. If S 与任何直接相连的内容服务器的域名匹配，
   then 把数据报投递到目的内容服务器上;
3. else if 路由表中包含了一个到 S 的指定路由，
   then 把数据报投递到表中指定的下一跳;
4. else if 路由表中包含了到 S 的父域的一个路由，
   then 把数据报投递到表中指定的下一跳;
5. else if 表中包含一个默认路由，
   then 把数据报发送动表中指定的默认路由上
6. else 宣布路由出错

§ 2.5 名字路由的表搜索算法

名字路由器的主要功能是按照 DNS 分组中的目的内容域名转发分组。查找路由表决定将分组发往哪个下一跳，这是转发分组过程中的重要一步。因此,快速的路由表查找算法是实现高速分组转发的关键。为了提高路由查找的性能，本节对多种路由查找算法进行了讨论，分析了路由查找问题和实现难点，并对它们进行了详细的分析和比较。最后确立了名字路由表的查找算法，并给出了具体实现。

2.5.1 路由查找算法的分类

最长后缀匹配查找的难点在于，在查找过程中不仅需要与内容后缀的比特值进行匹配查

找，而且还需要考虑内容后缀的长度，因此各种路由查找算法都可以归结为这两个方面的匹配查找过程。

1 基于名字后缀值的路由查找算法

基于名字后缀值路由查找算法的特点是：通过对整个名字后缀空间进行名字关键字穷举，来消除名字后缀长度对查找的影响。第 2.5.2 小节中介绍的线性匹配查找算法是基于名字后缀值查找中最为简单直观的地址后缀查找方法。

2 基于名字后缀长度的路由查找算法

基于名字后缀长度的路由查找算法的出发点是在后缀长度空间内进行查找。与基于名字后缀值查找算法类似，查找过程可以使用线性遍历法，二分法遍历法或者 HASH 查找法。第 2.5.3 节介绍的 Trie 树查找算法就是基于地址前缀长度的线性遍历法，因为对于查找的第 i 步来说,匹配的是长度为 i 的地址前缀。同理,第 2.5.3 节介绍的多分支 Trie 查找算法也是基于在地址前缀长度空间内的线性遍历法,但由于它采用大于 1 的步宽,所遍历的地址长度数目变少了,查找性能得到了提高。

2.5.2 线性查找算法

线性表结构是最简单的查找数据结构。因此，可以把所有的路由名字后缀，用线性链表的方式组织起来。每一次查找需要遍历线性表中的所有表项,在遍历过程中记录最长的路由后缀项，直到遍历完整个线性链表为止。对于 N 个路由后缀表项，查找过程的算法复杂度为 O(N)，存储空间复杂度为 O(N)，插入删除路由表项的算法复杂度为 O(1)（假设表项插入和删除均在线性表的末尾进行）。

2.5.3 字符树查找算法

※ 字符 Trie 树(binary Trie)

用字符 Trie 结构来表示名字后缀是一个常用的方法。Trie 采用一种基于树的数据结构，通过前缀中每一位比特的值来决定树的分支。图 2.16 就是用二进制 Trie 结构(树中每一个内部结点最多包含两个子结点)来表示的地址前缀表。

在 Trie 树中，处于第 L 层的结点代表了一类地址前 L 个字符均相同的地址空间，并且这前 L 个比特串就是由从根结点到这个结点路径上的 L 个字符特组成。例如，图 2.16 中处于第 2 层的最左边结点就代表了所有前 2 个比特为 aa 的字符串，而且字符串 aa 就是根结点到结点路径上的字符按照遍历顺序所构成的。

字符 Trie 树不足之处在于：查找过程需要大量的存储器访问操作。实验表明，使用字符 Trie 树来表示典型路由器中 47 113 表项数目的后缀表，最大搜索深度为 26,平均搜索深度也达到了 20。

※ 双链树 Trie 树(*multibit Trie*)

把一个关键码集合中出现的所有关键码用字符树的形式表示出来称作 Briandais 树，把它转换为对应的二叉树并用 llink-rlink（左儿子，右兄弟）法进行存储，则通常称作双链树。双链树中，结点的 rlink 指向原来的 Briandais 树中同一层的下一个兄弟结点；llink 指向原来的 Briandais 树的下一层中它的第一个子女结点。特殊的，对应于每个关键码的最后一个字符的结点的 llink 指向对应于此关键码的属性数据地址。为区分结点的 llink 到底是指向下一层的结点还是属性数据地址，在每个关键码的最后增加了一个特殊的字符，作为关键码的结束标记。在 c/c++语言中，这个结束符就是’\0’。双链树中，对应于特殊字符’\0’的结点的 llink 指向属性数据地址。

在双链树中检索给定关键码是沿着指针进行的。当待查关键码中的一个字符与双链树中

找，而且还需要考虑内容后缀的长度，因此各种路由查找算法都可以归结为这两个方面的匹配查找过程。

1 基于名字后缀值的路由查找算法

基于名字后缀值路由查找算法的特点是：通过对整个名字后缀空间进行名字关键字穷举，来消除名字后缀长度对查找的影响。第 2.5.2 小节中介绍的线性匹配查找算法是基于名字后缀值查找中最为简单直观的地址后缀查找方法。

2 基于名字后缀长度的路由查找算法

基于名字后缀长度的路由查找算法的出发点是在后缀长度空间内进行查找。与基于名字后缀值查找算法类似，查找过程可以使用线性遍历法，二分法遍历法或者 HASH 查找法。第 2.5.3 节介绍的 Trie 树查找算法就是基于地址前缀长度的线性遍历法，因为对于查找的第 i 步来说,匹配的是长度为 i 的地址前缀。同理,第 2.5.3 节介绍的多分支 Trie 查找算法也是基于在地址前缀长度空间内的线性遍历法,但由于它采用大于 1 的步宽,所遍历的地址长度数目变少了,查找性能得到了提高。

2.5.2 线性查找算法

线性表结构是最简单的查找数据结构。因此，可以把所有的路由名字后缀，用线性链表的方式组织起来。每一次查找需要遍历线性表中的所有表项,在遍历过程中记录最长的路由后缀项，直到遍历完整个线性链表为止。对于 N 个路由后缀表项，查找过程的算法复杂度为 O(N)，存储空间复杂度为 O(N)，插入删除路由表项的算法复杂度为 O(1)（假设表项插入和删除均在线性表的末尾进行）。

2.5.3 字符树查找算法

※ 字符 Trie 树(binary Trie)

用字符 Trie 结构来表示名字后缀是一个常用的方法。Trie 采用一种基于树的数据结构，通过前缀中每一位比特的值来决定树的分支。图 2.16 就是用二进制 Trie 结构(树中每一个内部结点最多包含两个子结点)来表示的地址前缀表。

在 Trie 树中，处于第 L 层的结点代表了一类地址前 L 个字符均相同的地址空间，并且这前 L 个比特串就是由从根结点到这个结点路径上的 L 个字符特组成。例如，图 2.16 中处于第 2 层的最左边结点就代表了所有前 2 个比特为 aa 的字符串，而且字符串 aa 就是根结点到结点路径上的字符按照遍历顺序所构成的。

字符 Trie 树不足之处在于：查找过程需要大量的存储器访问操作。实验表明，使用字符 Trie 树来表示典型路由器中 47 113 表项数目的后缀表，最大搜索深度为 26,平均搜索深度也达到了 20。

※ 双链树 Trie 树(*multibit Trie*)

把一个关键码集合中出现的所有关键码用字符树的形式表示出来称作 Briandais 树，把它转换为对应的二叉树并用 llink-rlink（左儿子，右兄弟）法进行存储，则通常称作双链树。双链树中，结点的 rlink 指向原来的 Briandais 树中同一层的下一个兄弟结点；llink 指向原来的 Briandais 树的下一层中它的第一个子女结点。特殊的，对应于每个关键码的最后一个字符的结点的 llink 指向对应于此关键码的属性数据地址。为区分结点的 llink 到底是指向下一层的结点还是属性数据地址，在每个关键码的最后增加了一个特殊的字符，作为关键码的结束标记。在 c/c++语言中，这个结束符就是’\0’。双链树中，对应于特殊字符’\0’的结点的 llink 指向属性数据地址。

在双链树中检索给定关键码是沿着指针进行的。当待查关键码中的一个字符与双链树中

找，而且还需要考虑内容后缀的长度，因此各种路由查找算法都可以归结为这两个方面的匹配查找过程。

1 基于名字后缀值的路由查找算法

基于名字后缀值路由查找算法的特点是：通过对整个名字后缀空间进行名字关键字穷举，来消除名字后缀长度对查找的影响。第 2.5.2 小节中介绍的线性匹配查找算法是基于名字后缀值查找中最为简单直观的地址后缀查找方法。

2 基于名字后缀长度的路由查找算法

基于名字后缀长度的路由查找算法的出发点是在后缀长度空间内进行查找。与基于名字后缀值查找算法类似，查找过程可以使用线性遍历法，二分法遍历法或者 HASH 查找法。第 2.5.3 节介绍的 Trie 树查找算法就是基于地址前缀长度的线性遍历法，因为对于查找的第 i 步来说,匹配的是长度为 i 的地址前缀。同理,第 2.5.3 节介绍的多分支 Trie 查找算法也是基于在地址前缀长度空间内的线性遍历法,但由于它采用大于 1 的步宽,所遍历的地址长度数目变少了,查找性能得到了提高。

2.5.2 线性查找算法

线性表结构是最简单的查找数据结构。因此，可以把所有的路由名字后缀，用线性链表的方式组织起来。每一次查找需要遍历线性表中的所有表项,在遍历过程中记录最长的路由后缀项，直到遍历完整个线性链表为止。对于 N 个路由后缀表项，查找过程的算法复杂度为 O(N)，存储空间复杂度为 O(N)，插入删除路由表项的算法复杂度为 O(1)（假设表项插入和删除均在线性表的末尾进行）。

2.5.3 字符树查找算法

※ 字符 Trie 树(binary Trie)

用字符 Trie 结构来表示名字后缀是一个常用的方法。Trie 采用一种基于树的数据结构，通过前缀中每一位比特的值来决定树的分支。图 2.16 就是用二进制 Trie 结构(树中每一个内部结点最多包含两个子结点)来表示的地址前缀表。

在 Trie 树中，处于第 L 层的结点代表了一类地址前 L 个字符均相同的地址空间，并且这前 L 个比特串就是由从根结点到这个结点路径上的 L 个字符特组成。例如，图 2.16 中处于第 2 层的最左边结点就代表了所有前 2 个比特为 aa 的字符串，而且字符串 aa 就是根结点到结点路径上的字符按照遍历顺序所构成的。

字符 Trie 树不足之处在于：查找过程需要大量的存储器访问操作。实验表明，使用字符 Trie 树来表示典型路由器中 47 113 表项数目的后缀表，最大搜索深度为 26,平均搜索深度也达到了 20。

※ 双链树 Trie 树(multibit Trie)

把一个关键码集合中出现的所有关键码用字符树的形式表示出来称作 Briandais 树，把它转换为对应的二叉树并用 llink-rlink（左儿子，右兄弟）法进行存储，则通常称作双链树。双链树中，结点的 rlink 指向原来的 Briandais 树中同一层的下一个兄弟结点；llink 指向原来的 Briandais 树的下一层中它的第一个子女结点。特殊的，对应于每个关键码的最后一个字符的结点的 llink 指向对应于此关键码的属性数据地址。为区分结点的 llink 到底是指向下一层的结点还是属性数据地址，在每个关键码的最后增加了一个特殊的字符，作为关键码的结束标记。在 c/c++语言中，这个结束符就是’\0’。双链树中，对应于特殊字符’\0’的结点的 llink 指向属性数据地址。

在双链树中检索给定关键码是沿着指针进行的。当待查关键码中的一个字符与双链树中

对应层的第一个结点比较不等时，则沿着 rlink 依次与同层的其他兄弟结点比较；当比较相等时则沿着 llink 进入双链树的下一层，并考查待查关键码中下一个位置的字符。

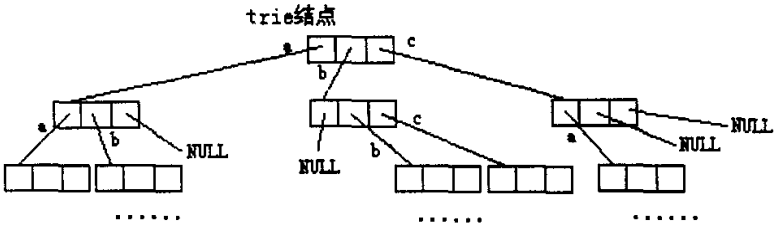


图 2.16 字符 Trie 树

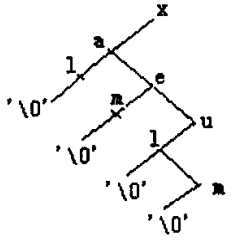


图 2.17 Briandais 树的双链树表示

※ 字符树性能简单比较

图 2.18 表示用 trie 结构和双链树实现的名字路由表的空间消耗情况。可以发现，这两种方法的空间需求都比较大。特别是 trie 结构的方法，其树结构本身的空间消耗就已经达到了双链树路由表总的空间消耗量。其总的空间消耗量平均是双链树方法的 123%。

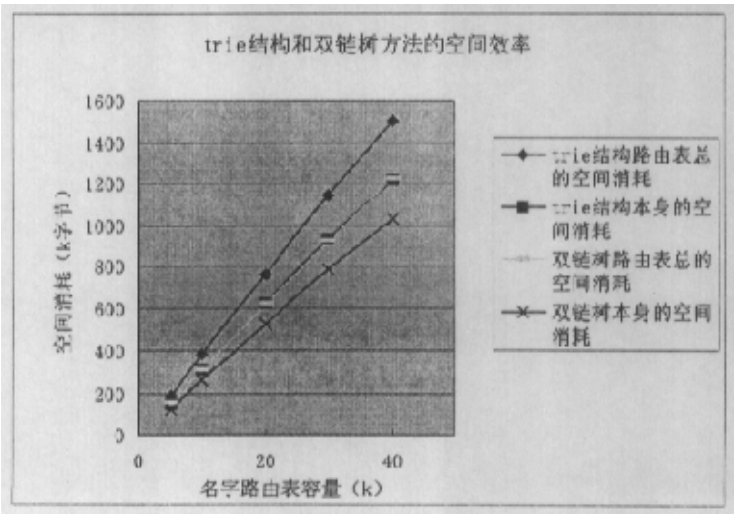


图 2.18 空间消耗图

2.5.4 HASH 查找算法

对应层的第一个结点比较不等时，则沿着 rlink 依次与同层的其他兄弟结点比较；当比较相等时则沿着 llink 进入双链树的下一层，并考查待查关键码中下一个位置的字符。

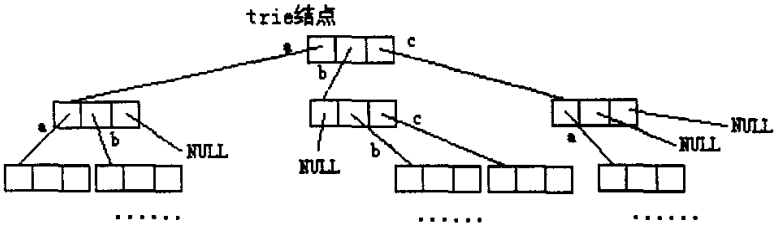


图 2.16 字符 Trie 树

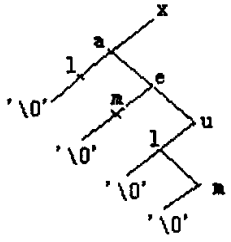


图 2.17 Briandais 树的双链树表示

※ 字符树性能简单比较

图 2.18 表示用 trie 结构和双链树实现的名字路由表的空间消耗情况。可以发现，这两种方法的空间需求都比较大。特别是 trie 结构的方法，其树结构本身的空间消耗就已经达到了双链树路由表总的空间消耗量。其总的空间消耗量平均是双链树方法的 123%。

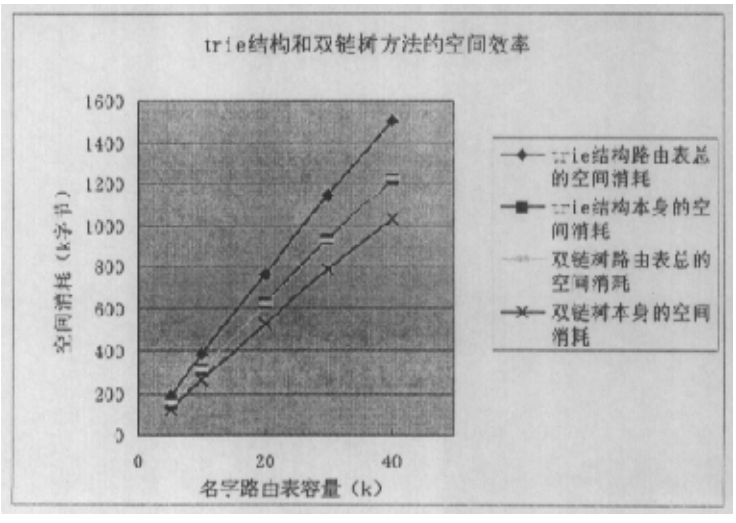


图 2.18 空间消耗图

2.5.4 HASH 查找算法

HASH，也译作散列。这是一种行之有效的存储方式和检索方法，获得了广泛的应用。本文将采用 HASH 作为路由表查找算法。它的基本思想是以关键码的值为自变量，通过一定的函数关系（称为 hash 函数或散列函数），计算出对应的函数值来，把这个值解释为结点的存储地址，将结点存入这样计算得到的存储单元中。检索时，再根据要检索的关键码用同样的函数计算地址，然后到相应的单元里去取要找的结点。所以又称关键码－地址转换法（KAT）。

当两（多）个不同的关键码，有相同的 hash 函数值时，称之为“碰撞”。碰撞的两（多）个关键码称之为同义词。通常，同一个结点中可以存放不止一个关键码，这样具有唯一 hash 值的一个结点，被称作一个 hash 桶。

当发生溢出时，需要为这个关键码寻找另一个桶。一般有两类方法：开地址法和链接法。开地址法是当溢出发生时，用某种方法在基本区域内形成一个探查序列，沿着这个探查序列一个桶一个桶的查找，直到找到一个还没满的桶为止。链接法是当溢出发生时就拉出一条链，建立一个链接方式的字表。理论和实践都已经证明，如果系统能够方便快速的分配存储空间，那链接法的效果要远优于开地址法。

Hash 方法有一个很重要的参数——“负载因子”α（也可称作“装填系数”）。

$$\alpha=\frac{\text{hash 表容纳的关键码数量 }(n)}{\text{hash 桶数量 }(B)}\quad;\text{（式 2.1）}$$

α 的大小对于碰撞的发生频率影响很大。直观上容易想象，散列表装的越满，则再要装入新的关键码时，发生溢出的机会就越大

※ 采用链接法的理论最优 HASH 函数的性能推导

设理论上最优的 hash 函数为 H，则 H 应该可以将关键码绝对均匀的分布于各个 hash 桶中。采用链接法时，一个 hash 桶也就是一个子链表。记某个桶中的记录数为 X，X 是一个随机变量。 当表足够大时，满足如下统计约束：

$$E(X)=\alpha,\ (\alpha=\frac{n}{B})\quad;\text{（式 2.2）}$$

对于 H 函数，可以使得 $X\in\{[\alpha],[\alpha]+1\}$，[]为向下取整符号

则

$$P1=P(X=[\alpha])=[\alpha]+1-\alpha,\ P2=P(X=[\alpha]+1)=\alpha-[\alpha];\quad\text{（式 2.3）}$$

因此，对于某一张表来说，假设其中每个关键码都以等可能概率被查询，则查询命中时的比较次数的数学期望为：

$$S=E(comps)=P1\cdot\frac{\sum_{i=1}^{[\alpha]}i}{[\alpha]}+P2\cdot\frac{\sum_{i=1}^{[\alpha]+1}i}{[\alpha]+1}=0.5+\frac{\alpha}{2};\quad\text{（式 2.4）}$$

查询一条不在表中的关键字的比较次数的期望值为：

$$U=E(X)=\alpha;\qquad\text{（式 2.5）}$$

溢出率为：

$$Flow = P1 \cdot \frac{[\alpha]-1}{\alpha} + P2 \cdot \frac{[\alpha]}{\alpha} = 1 - \frac{1}{\alpha}; \qquad \text{（式 2.6）}$$

X 的方差为：

$$D(X) = E(X^2) - [E(X)]^2 = \{P1 \cdot [\alpha]^2 + P2 \cdot ([\alpha]+1)^2\} - \alpha^2 \qquad \text{（式 2.7）}$$
$$= \alpha - (\alpha - [\alpha])^2 - [\alpha]$$

标准差为：

$$d(X) = \sqrt{D(X)} = \sqrt{\alpha - (\alpha - [\alpha])^2 - [\alpha]} \qquad \text{；（式 2.8）}$$

以上这些数值将是实际 hash 函数性能的理论极限。

※ 采用链接法的均匀 HASH 函数的性能估计

采用链接法的均匀 hash 函数的性能如下：（所谓均匀，是指对每个关键码都以等可能概率映射到某一个桶中）

查询一条不在表中的关键字的平均比较次数为：

$U_n \approx \alpha$； （式 2.9）

查询一条在表中的关键字的平均比较次数为：

$S_n \approx 1 + \alpha/2$； （式 2.10）

※ 针对字符串的常用 HASH 方法介绍

针对字符串，一般有几种常用的 hash 方法。

折叠法：

这种方法是将字符串 x 从左到右分成等长的几段，一般是每个字符为一段，然后将这些段累加起来，即得到 x 的 hash 地址。叠加时可以是移位叠加（即所有段的最低有效位对齐），也可以是折叠叠加（即相邻段的高位和低位对齐）。

平方取中法：

这种 hash 函数的计算过程是：对字符串的内码取平方，然后在平方数的中间部分取恰当的位数作为桶的地址。如果字符串较长，其内码长度超过计算机的字长，就需要编制特殊的程序来进行平方操作。

除留余数法：

这种方法是将字符串当作一个超长的整数，然后把这个数除以某个数 M，取余数作为该串的 hash 地址。这种方法要求选择一个大素数 M，在实用上，选择不含小于 20 的素因子的正整数作为 M 就可以了[1]。

数字分析法：

对关键码的各位进行分析，丢掉分布不均匀的位，留下分布均匀的位作为 hash 地址。不过这种方法需要事先知道关键码的分布情况。

基数转换法：

将关键码的若干位分段，然后把这若干段看作是在另一个基数制上的表示，接着把它转换成原来基数制的数，再用取余或数字分析法取其中几位作为地址。

图 2.19 给出了部分 HASH 方法的时间性能比较图。

Hash_8Add()函数采用移位折叠法，每个字符作为一段，将每个字符的 ASCII 码相加作为 hash 值。Hash_16Add()函数仍采用移位折叠法，但将两个字符作为一段。

Hash_ELF()函数是一个在 Unix 系统中使用的 hash 函数，它与用于 UNIX 系统 V 版本 4 中可执行文件与目标文件的“可执行链接格式”（Executable and Linking Format，即 ELF）一起使用。

Hash_XOR()，此函数采用将连续两个字符当作一个 16 位整数，求出这些整数的异或和作为 hash 值。

Hash_8w()，将每个字符的 ASCII 值乘上 8 的某个幂作为权重，然后求出带权和作为 hash 值。

Hash_7w()，与 hash_8w()基本类似，但权重是 7 的某个幂。

Hash_8Mod()，采用除留余数法。Hash_16Mod()，仍采用除留余数法，但为了加快计算速度，对每两个字符进行一次取余操作。

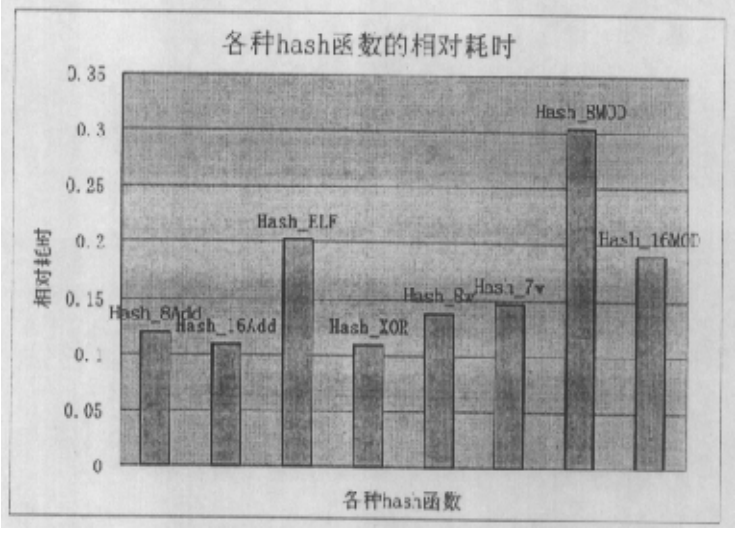


图 2.19 几种 HASH 函数的性能分析与比较

2.5.4 算法实现

本文的 HASH 查找算法的关键码采用数字分析法。关于关键码计算的 HASH 函数源码如下：

```
#define CM_HASH_MIX(a,b,c) \
{ \
  a -= b; a -= c; a ^= (c>>13); \
  b -= c; b -= a; b ^= (a<<8);  \
  c -= a; c -= b; c ^= (b>>13); \
  a -= b; a -= c; a ^= (c>>12); \
  b -= c; b -= a; b ^= (a<<16); \
  c -= a; c -= b; c ^= (b>>5);  \
  a -= b; a -= c; a ^= (c>>3);  \
  b -= c; b -= a; b ^= (a<<10); \
```

图 2.19 给出了部分 HASH 方法的时间性能比较图。

Hash_8Add()函数采用移位折叠法，每个字符作为一段，将每个字符的 ASCII 码相加作为 hash 值。Hash_16Add()函数仍采用移位折叠法，但将两个字符作为一段。

Hash_ELF()函数是一个在 Unix 系统中使用的 hash 函数，它与用于 UNIX 系统 V 版本 4 中可执行文件与目标文件的“可执行链接格式”（Executable and Linking Format，即 ELF）一起使用。

Hash_XOR()，此函数采用将连续两个字符当作一个 16 位整数，求出这些整数的异或和作为 hash 值。

Hash_8w()，将每个字符的 ASCII 值乘上 8 的某个幂作为权重，然后求出带权和作为 hash 值。

Hash_7w()，与 hash_8w()基本类似，但权重是 7 的某个幂。

Hash_8Mod()，采用除留余数法。Hash_16Mod()，仍采用除留余数法，但为了加快计算速度，对每两个字符进行一次取余操作。

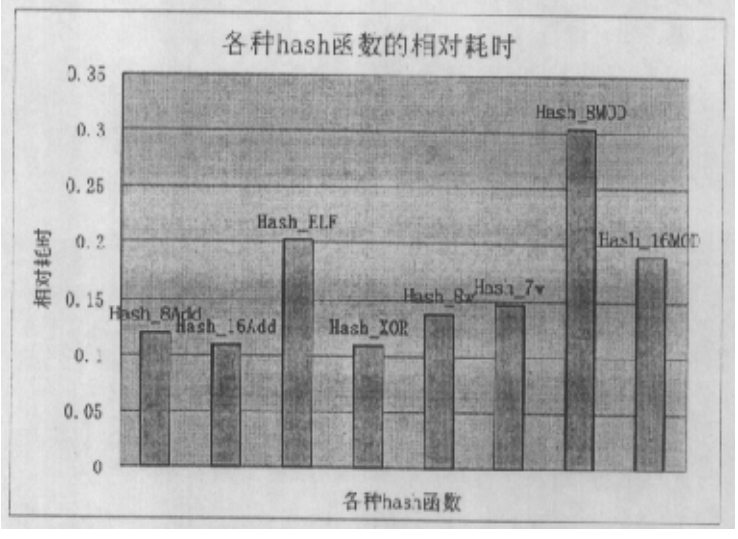


图 2.19 几种 HASH 函数的性能分析与比较

2.5.4 算法实现

本文的 HASH 查找算法的关键码采用数字分析法。关于关键码计算的 HASH 函数源码如下：

```
#define CM_HASH_MIX(a,b,c) \
{ \
  a -= b; a -= c; a ^= (c>>13); \
  b -= c; b -= a; b ^= (a<<8);  \
  c -= a; c -= b; c ^= (b>>13); \
  a -= b; a -= c; a ^= (c>>12); \
  b -= c; b -= a; b ^= (a<<16); \
  c -= a; c -= b; c ^= (b>>5);  \
  a -= b; a -= c; a ^= (c>>3);  \
  b -= c; b -= a; b ^= (a<<10); \
```

```
  c -= a; c -= b; c ^= (b>>15); \
}
HASH.C:
    U32                 a;                  /* hash variables */
    U32                 b;                  /* hash variables */
    U32                 c;                  /* hash variables */
    U32                 len;                /* length */
    /* Set up the internal state */
    len = keyLen;       /* key length */
    a = 0x9e3779b9;   /* a = b = the golden ratio; an arbitrary value */
    b = 0x9e3779b9;
    c = 0x12345678;   /* some value */
    /*---------------------------------------- handle most of the key */
    while (len >= 12)
    {
        a += (key[0] +((U32)key[1]<<8) +((U32)key[2]<<16) +((U32)key[3]<<24));
        b += (key[4] +((U32)key[5]<<8) +((U32)key[6]<<16) +((U32)key[7]<<24));
        c += (key[8] +((U32)key[9]<<8) +((U32)key[10]<<16)+((U32)key[11]<<24));
        CM_HASH_MIX(a, b, c);
        key += 12; len -= 12;
    }
    /*------------------------------------- handle the last 11 bytes */
    c += keyLen;
    switch(len)                     /* all the case statements fall through */
    {
    case 11: c+=((U32)key[10]<<24);
    case 10: c+=((U32)key[9]<<16);
    case 9 : c+=((U32)key[8]<<8);
        /* the first byte of c is reserved for the keyLen */
    case 8 : b+=((U32)key[7]<<24);
    case 7 : b+=((U32)key[6]<<16);
    case 6 : b+=((U32)key[5]<<8);
    case 5 : b+=key[4];
    case 4 : a+=((U32)key[3]<<24);
    case 3 : a+=((U32)key[2]<<16);
    case 2 : a+=((U32)key[1]<<8);
    case 1 : a+=key[0];
      /* case 0: nothing left to add */
    }
    CM_HASH_MIX(a,b,c);
    return C%M;   /* HASH table total number */
    /*-------------------------------------------- report the result */
```

- **HASH 查找算法流程**

图 2.20 给出了 HASH 算法的流程。首先对查询域名进行分析，判断它是否为顶级域名。

顶级域名表明 HASH 查找失败。如果不是顶级域名，就根据上面的 HASH 方法计算关键码。每一个关键码对应了 HASH 表的数组索引。由于“碰撞”，我们用链接法对每一个关键码建立一个链接方式的字表（见图 2.22）。接下来，需要把域名和链表里的每一个表项进行字符串的全匹配。如果找到匹配的表项，退出 HASH 查找，进行下一步操作。否则，将域名退化，如四级域名退化为三级域名，重新开始 HASH 查找。

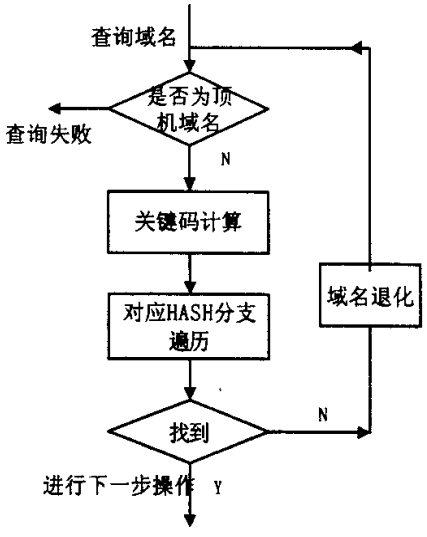


图 2.20 HASH 算法流程图

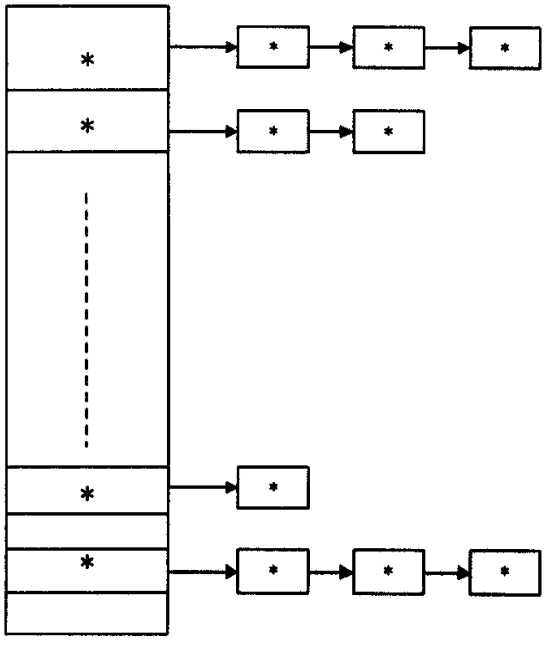


图 21 HASH 表结构

*：带*号的格子表示有数据，否则表示闲置。

● **HASH 表维护**

由于系统资源有限，HASH 表的表项总数受到限制。假设 HASH 表的索引数组的大小为 M，每一个索引对应的平均链接表长度为 N，则 HASH 表的大小为 M×N。我们必须记录 HASH 表的当前长度 ，判断 HASH 表长度是否已经越界。如果越界，系统将采取 FIFO 的方式更新 HASH 表。因此系统要有一张 FIFO 记录表，按照 FIFO 的原则记录索引数组的使用情况。

HASH 表的维护包括添加、删除、更新。更新操作非常简单，只需要根据域名关键字，找到对应的表项进行数据的更新。删除和添加操作需要设计 FIFO 记录表的操作。

添加操作：

1. 如果对应的数组索引链表长度为 0，并且当前表长度不越界，需要把数组索引添加到 FIFO 记录表的末尾。把数据插入链表表头。表长度加一。

2. 如果对应的数组索引链表长度不为 0，并且当前表长度不越界，把数据插入到链表的表头。表长度加一。

3. 如果当前表长度越界，按照 FIFO 记录表表头的数组索引，删除链表表头的数据，假如删除导致链表长度为空，移去 FIFO 表头的数据。然后按照情况 1 或 2，添加数据。表长度不变。

删除操作：删除对应数组索引链表表头的数据。假如删除导致链表长度为空，查找 FIFO 记录表，移去 FIFO 表中对应的索引数据。数组长度减 1。

§ 2.6 INRP 的路由仿真算法

本节根据前面几节的描述，对 INRP 名字路由算法进行路由仿真。仿真的流程如下图 2.22 所示：

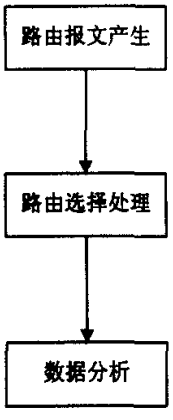


图 2.22 INRP 仿真流程

我们假定实际网络的路由报文的产生服从泊松分布。因此在仿真中，根据泊松算法，随机产生路由数据。仿真的目的在于取得下列指标：

- ✓ 路由器的吞吐量，即路由器单位时间（每秒）内所建立的连接数；
- ✓ INRP 的路由平均时延，即 INRP 的 RTT 时间；

INRP 的 RTT 由报文往返所经历的网络传输时延和途经路由器的处理时延、等待时延组成。用公式表示，

$$RTT = \sum_i 2*T1_i(\text{网络路径 i 传播时延}) + \sum_i (T2_i(\text{路由器 i 处理时延}) + T3_i$$

● HASH 表维护

由于系统资源有限，HASH 表的表项总数受到限制。假设 HASH 表的索引数组的大小为 M，每一个索引对应的平均链接表长度为 N，则 HASH 表的大小为 M×N。我们必须记录 HASH 表的当前长度 ，判断 HASH 表长度是否已经越界。如果越界，系统将采取 FIFO 的方式更新 HASH 表。因此系统要有一张 FIFO 记录表，按照 FIFO 的原则记录索引数组的使用情况。

HASH 表的维护包括添加、删除、更新。更新操作非常简单，只需要根据域名关键字，找到对应的表项进行数据的更新。删除和添加操作需要设计 FIFO 记录表的操作。

添加操作：

1. 如果对应的数组索引链表长度为 0，并且当前表长度不越界，需要把数组索引添加到 FIFO 记录表的末尾。把数据插入链表表头。表长度加一。

2. 如果对应的数组索引链表长度不为 0，并且当前表长度不越界，把数据插入到链表的表头。表长度加一。

3. 如果当前表长度越界，按照 FIFO 记录表表头的数组索引，删除链表表头的数据，假如删除导致链表长度为空，移去 FIFO 表头的数据。然后按照情况 1 或 2，添加数据。表长度不变。

删除操作：删除对应数组索引链表表头的数据。假如删除导致链表长度为空，查找 FIFO 记录表，移去 FIFO 表中对应的索引数据。数组长度减 1。

§ 2.6 INRP 的路由仿真算法

本节根据前面几节的描述，对 INRP 名字路由算法进行路由仿真。仿真的流程如下图 2.22 所示：

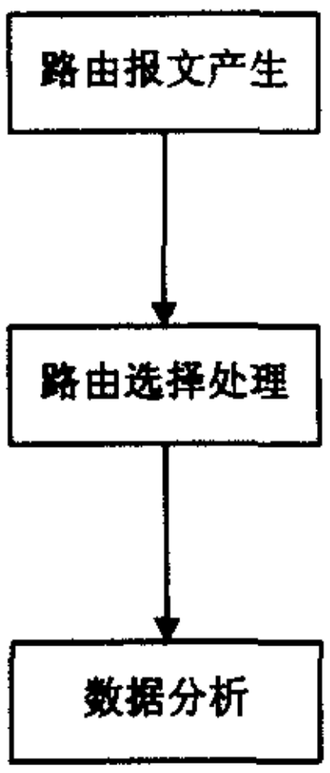


图 2.22 INRP 仿真流程

我们假定实际网络的路由报文的产生服从泊松分布。因此在仿真中，根据泊松算法，随机产生路由数据。仿真的目的在于取得下列指标：

- ✓ 路由器的吞吐量，即路由器单位时间（每秒）内所建立的连接数；
- ✓ INRP 的路由平均时延，即 INRP 的 RTT 时间；

INRP 的 RTT 由报文往返所经历的网络传输时延和途经路由器的处理时延、等待时延组成。用公式表示，

$$\text{RTT} = \sum_i 2*T1_i(\text{网络路径 i 传播时延}) + \sum_i (T2_i\ (\text{路由器 i 处理时延}) + T3_i$$

(在路由器 i 中的等待时延))；（式 2.11）

网络传播时延既要考虑到网络拓扑，又要考虑线路负荷。路由器的处理时延主要由名字路由表的查询时间耗费构成。路由器的等待时延其实代表了路由器的负荷程度。T3 和路由器处理时延等待列长度有关。假如路由等待队列长度为 n，那么等待时延 T3 可以由以下公式推出：

$$T3 = \quad (n - \frac{1}{2})\ T2;\ （式\ 2.12）$$

路由器的负荷越重，路由器的等待队列越长，报文的等待时间 T3 也就越长。反之，路由器的等待队列将缩短，T3 也就减小。

接下来我们讨论路由器的处理时延。在一定的条件下，路由器的处理时延是固定的。这里我们应用斯坦福的测试数据，该数据是在 PIII 600 MHZ，344MB 内存，Linux 内核 2.2.12 版本试验环境下测得的。名字路由路由器包含一张 500 万条内容记录的名字路由表。该路由表的数据是随机产生的二级域名，其中 80%是.com 的信息，10%是.org 的信息，10%是.net 的信息。每一个域名的长度在 3~17 之间随机分布。试验表明路由表的包处理的耗费约为 6ms。在接下来的仿真试验中，我们将假定路由器处理时延 T2 为 6ms。

T1 的计算实际上是基于流量的路由算法的应用。它既考虑了网络拓扑结构又兼顾载荷。假定分组平均长度是$1/\mu = 800$字节。有队列原理导出的各条线路的平均传播延迟：

$$T1 = \frac{1}{\mu C - \lambda};\ （式\ 2.13）$$

其中，$1/\mu$ 是以比特为单位的分组平均长度，C 是以 b/s 为单位的容量，λ 是分组/s 为单位的平均流量。

针对 T1 的路由算法我们给出了本次仿真所采用的网络拓扑，如图 2.23 所示。网络拓扑同时标注了网络的线路负荷。表一给出了通信量和路由选择矩阵。例如，从 B 到 D 的分组数量为 3 分组/s，使用路由 BFD。需要注意的是，这里已经运用了某种路由选择算法（例如 Dijkstra）来导出了矩阵中所示的路由。

根据表 2.4，我们可以推算出每条线路的总量λ_i。每一条线路的传播延迟$T1_i$可以根据式 2.12 计算出来。表 2.5 给出了在网络拓扑图 23 的情况下的详细数据。

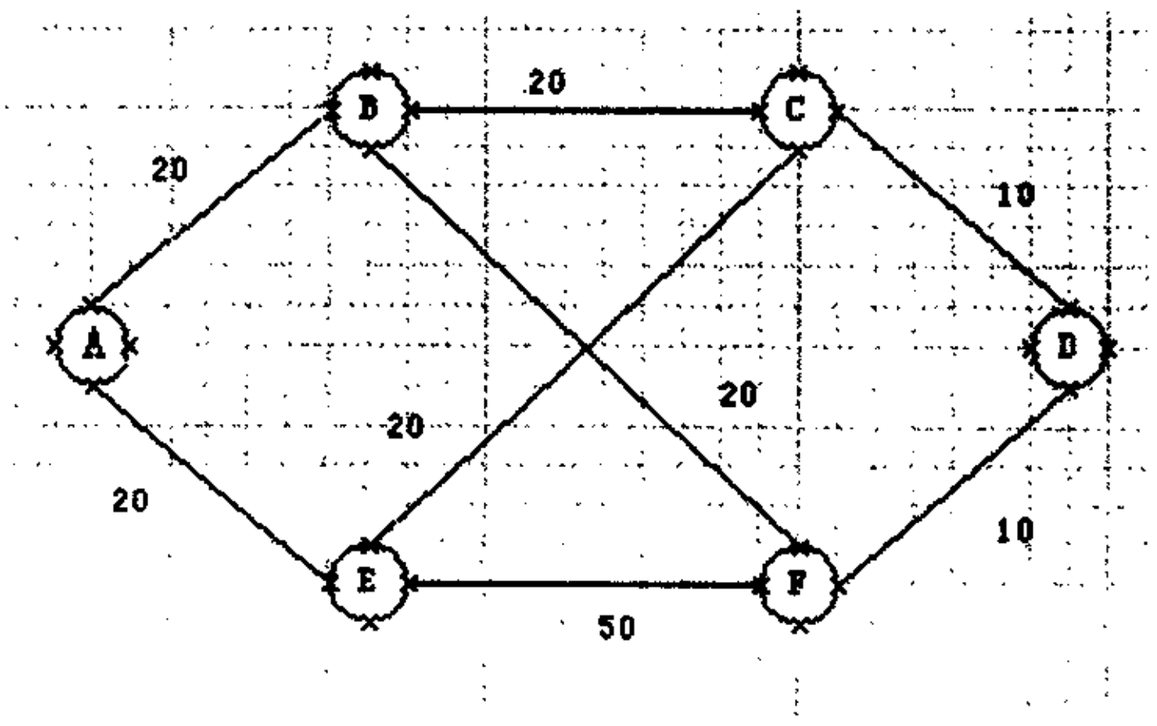


图 2.23 网络拓扑（用 kb/s 表示线路负荷）

	A	B	C	D	E	F
A		9: AB	4:ABC	1:ABFD	7:AE	4:AEF
B	9: BA		8:BC	3:BFD	2:BFE	4:BF
C	4:CBA	8:CB		3:CD	3:CE	3:CEF
D	1:DEBA	3:DFB	3:DC		3:DCF	4:DF
E	7:EA	2:EFB	3:EC	3:ECD		5:EF
F	4:EFA	4:FB	2:FEC	4:FD	5:FE	

表 2.4　通信量和路由选择矩阵（用分组/s 表示）

I	线路	λ_i（分组/s）	$C_i(kb/s)$	μC_i（分组/s）	$T1_i$（ms）	权值
1	AB	14	20	25	91	0.171
2	BC	12	20	25	77	0.145
3	CD	6	10	12.5	154	0.073
4	AE	11	20	25	71	0.134
5	EF	13	50	62.5	20	0.159
6	FD	8	10	12.5	222	0.098
7	BF	10	20	25	67	0.122
8	EC	8	20	25	59	0.098

表 2.5　采用了平均分组长度为 800 位的网络（图 2.22）分析。

因此，在以上条件下，INRP 路由的每一条报文的时延可以由式 2.11 计算。那么 INRP 平均时延

$$T=\frac{\sum_j \text{每一条报文的时延} T_j}{\text{总的报文数}}\ ;（式\ 2.14）$$

对于 Tj，构成变化的是等待时间 T3j。它和路由器的吞吐量以及路由器的负荷有着直接的关系。在处理能力一定的情况下，这种关系直接反映在路由等待队列的长度变化。

我们将用经典的 Dijkstra 算法进行路由选择。需要指出的是，在我们的网络弧段的标注中，包括了该条路径的通信负荷、流量、路由器的负荷程度等信息。尽管最后，标注的尺度是时间。而且，标注值是动态变化的。在网络处理条件一定的情况下，它将根据路由器的负

荷变化而变化。路由选择是以路由延迟为标准的。我们的目标是在一对路由器之间的路由，找出最短的路径。它的路由延迟将是最小的。

图 2.22 的网络拓扑将演化成下面的网络拓扑(见图 2.24)，它的标注发生了变化。图 2.24 也是本次性能仿真采用的实际模型。

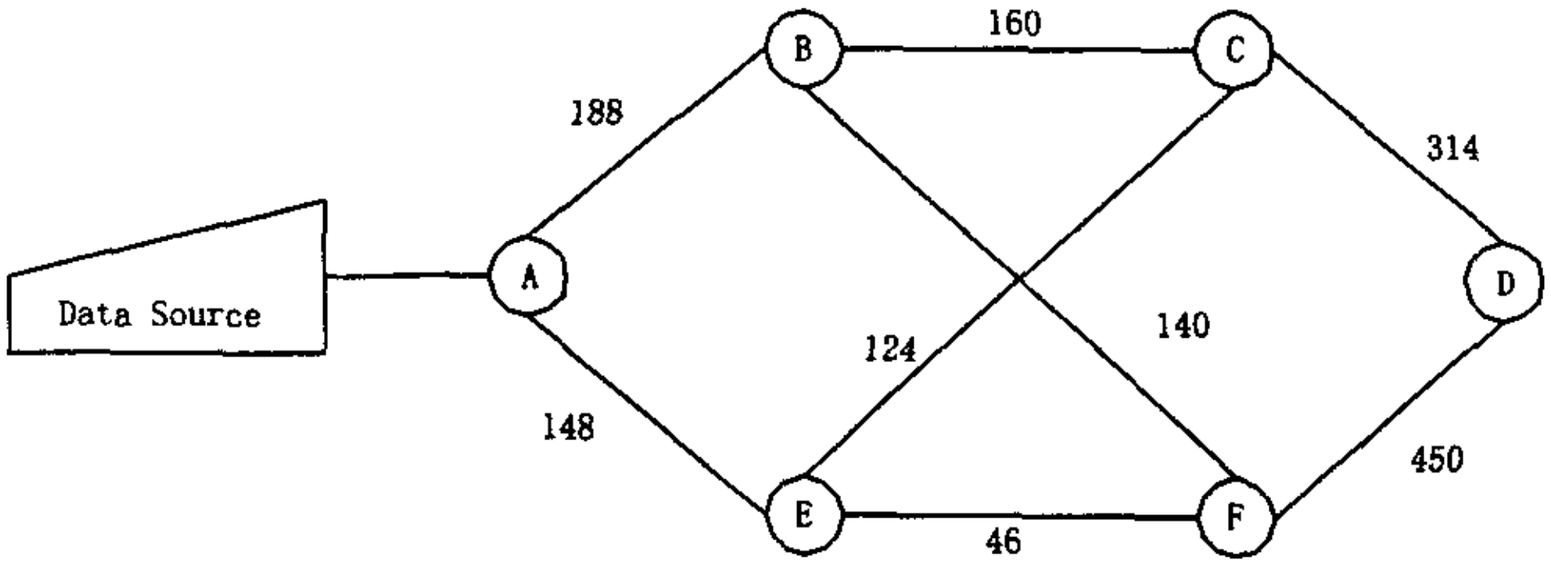


图 2.24　仿真网络拓扑

从图 2.24 的网络拓扑，我们可以看到，标注已经包括了网络的传播时延（来回）、路由器的处理时延。在计算最短路径的时候，我们还要加上途经路由器的负荷所造成的等待延迟。根据最小的路由延迟，我们选择最短的路由。同时，我们还需要对路由器的负荷状态进行更新。

图 2.24 中，Data Source 是本次仿真的路由源。我们假定路由数据产生时间服从泊松分布。 根据仿真结果数据，我们给出了路由平均时延图和路由器吞吐量图，分别如图 2.25 和图 2.26 所示。图中的 X 轴表示每秒产生的报文数。

仿真开始时，每秒产生的报文数不多，不足以对图 2.24 中的各个路由器造成负担，路由器的负载不大，处理能力有余。因此我们可以看到，路由器的吞吐量接近于每秒产生的报文数，而且随着报文产生数目的增长而迅速增加。在这个阶段，路由的平均时延很短，而且基本上保持不变。当每秒产生的报文数逐渐增加时，路由器的负载逐渐加重，路由平均时延也随之迅速上升，路由器的吞吐量也迅速上升，但是它的丢包率也随着上升，在每秒产生报文数的所占比例也随之下降。最后，当每秒产生的报文数目很多时，路由器基本上处于满荷状态，路由器的吞吐量和平均时延虽然随着每秒报文的增加而增加，当增幅很小，而且趋近于定值。

§ 2.7　本章小结

本章主要就基于名字路由技术的路由机制展开讨论。

本章首先概述了基于名字技术的路由机制，介绍了路由机制的主要流程，并简单的概括提出了路由机制的主要模块，即 INRP(DNS)查询报文的接收进程、高速 DNS 缓存查找进程、事务处理进程。

本章的随后几节都是围绕实现路由机制的 INRP 协议展开的。首先介绍了 INRP 的帧结构；然后分析了查询报文的接收进程，对其中的一些问题进行了讨论；然后，为了提高路由的性能，引入了 DNS 的高速缓存查找；事务处理进程是名字路由的主要模块，我们对它进

荷变化而变化。路由选择是以路由延迟为标准的。我们的目标是在一对路由器之间的路由，找出最短的路径。它的路由延迟将是最小的。

图 2.22 的网络拓扑将演化成下面的网络拓扑(见图 2.24)，它的标注发生了变化。图 2.24 也是本次性能仿真采用的实际模型。

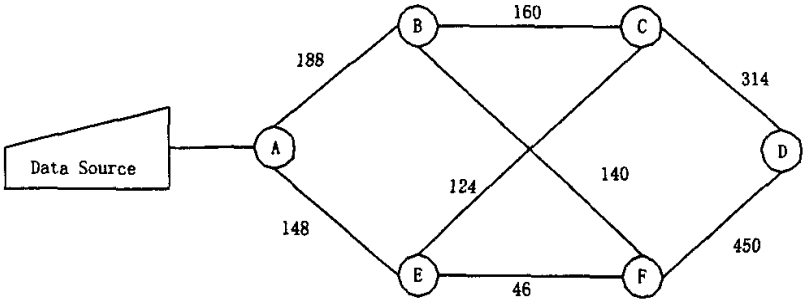


图 2.24 仿真网络拓扑

从图 2.24 的网络拓扑，我们可以看到，标注已经包括了网络的传播时延（来回）、路由器的处理时延。在计算最短路径的时候，我们还要加上途经路由器的负荷所造成的等待延迟。根据最小的路由延迟，我们选择最短的路由。同时，我们还需要对路由器的负荷状态进行更新。

图 2.24 中，Data Source 是本次仿真的路由源。我们假定路由数据产生时间服从泊松分布。 根据仿真结果数据，我们给出了路由平均时延图和路由器吞吐量图，分别如图 2.25 和图 2.26 所示。图中的 X 轴表示每秒产生的报文数。

仿真开始时，每秒产生的报文数不多，不足以对图 2.24 中的各个路由器造成负担，路由器的负载不大，处理能力有余。因此我们可以看到，路由器的吞吐量接近于每秒产生的报文数，而且随着报文产生数目的增长而迅速增加。在这个阶段，路由的平均时延很短，而且基本上保持不变。当每秒产生的报文数逐渐增加时，路由器的负载逐渐加重，路由平均时延也随之迅速上升，路由器的吞吐量也迅速上升，但是它的丢包率也随着上升，在每秒产生报文数的所占比例也随之下降。最后，当每秒产生的报文数目很多时，路由器基本上处于满荷状态，路由器的吞吐量和平均时延虽然随着每秒报文的增加而增加，当增幅很小，而且趋近于定值。

$2.7 本章小结

本章主要就基于名字路由技术的路由机制展开讨论。

本章首先概述了基于名字技术的路由机制，介绍了路由机制的主要流程，并简单的概括提出了路由机制的主要模块，即 INRP(DNS)查询报文的接收进程、高速 DNS 缓存查找进程、事务处理进程。

本章的随后几节都是围绕实现路由机制的 INRP 协议展开的。首先介绍了 INRP 的帧结构；然后分析了查询报文的接收进程，对其中的一些问题进行了讨论；然后，为了提高路由的性能，引入了 DNS 的高速缓存查找；事务处理进程是名字路由的主要模块，我们对它进

行了详细分析。

本章还对名字路由表结构进行了分析和改进，并对表搜索算法进行了分析比较。对本文采取的 HASH 算法进行了详细的讨论。

最后，我们对 INRP 的路由仿真算法进行了讨论，并给出了系统仿真结果。

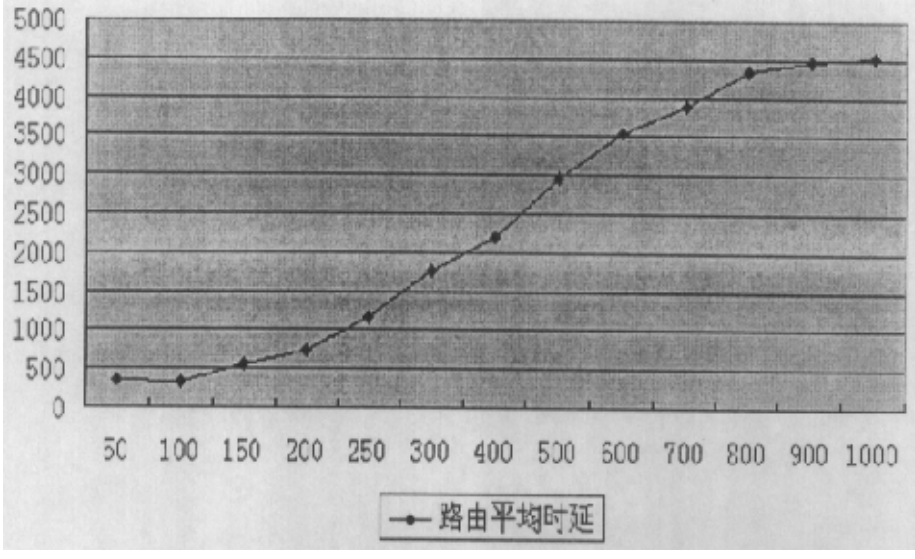


图 2.25 路由平均时延图

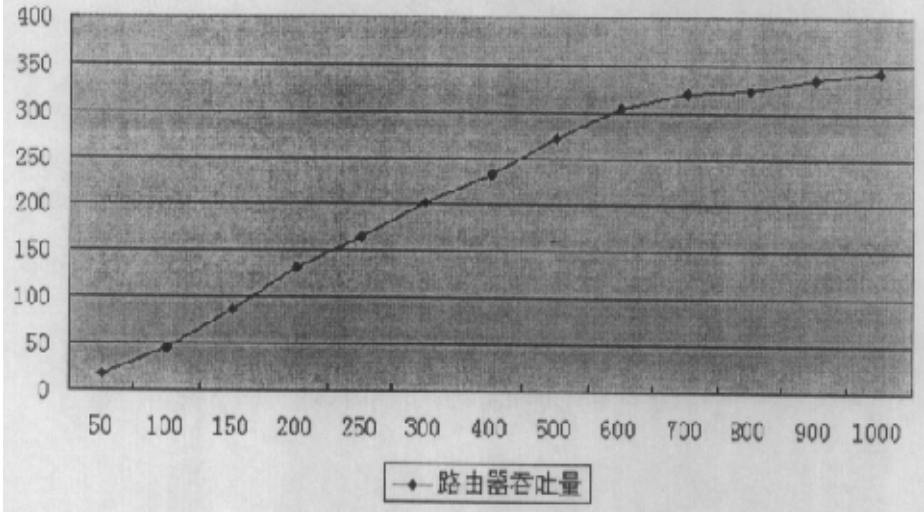


图 2.26 路由器吞吐量图

第三章　基于名字的路由系统的路由策略

§ 3.1　基于名字路由系统的路由策略概括

路由策略负责为路由系统创建和刷新名字路由表。在基于名字的路由系统中，实现路由策略的是 NBRP（Name-based router protocol）协议。本章将围绕路由策略与 NBRP 协议具体展开。

3.1.1　路由表的建立

在上一章中，我们讨论了关于名字路由器时内容网络互连的内容。每个名字路由器与两个以上的实际内容网络相连，负责在这些网络之间转发数据报。它从一个网络接口接收数据报，选路之后通过另一个网络接口转发出去。除了那些查询域名命中 DNS 高速缓存的数据报，主机把数据报按照查询域名对应的内容服务器的位置将其向前转发。数据报传输过程中经过若干个路由器，最后到达与目的站处于同一内容网络上的内容服务器。因此，名字路由系统组成了内容网络的基本体系结构，处理所有的内容网络数据报文通信量。

同时，我们研究了名字路由器向前转发数据报所用的名字路由选路算法，其中提到了名字路由表的表驱动算法。在名字路由表中的每个项目都指出了一个目的提供特定内容服务的内容域名和到达最近的内容服务器所经过的下一路由器的地址。

虽然我们已经了解名字路由系统向前转发名字路由数据报的基本原理。但是我们并不知道名字路由器如何获得选路表中的信息。这个问题可以分为两个方面：

1　名字路由表中该有什么数据？

2　名字路由器如何获得这些数据？

问题的答案以依赖与内容网络体系结构的复杂度、范围大小和管理策略。名字路由策略负责解决这个问题。

一般来说，路由的估计包括了初始化和更新两个问题。每个路由器在启动时都必须建立一个初始路由集合，还要在路由改变时去更新它（如网络接口发生故障）。初始化的策略取决与系统的配置。它包括：

1　ROM 读取。启动时从 ROM 或者二级存储器（硬盘）读取初始名字选路表的内容并将其驻留在内存中。

2　网络配置。启动时从 TFTP/FTP 等网络接口从网络服务器读取初始名字选路表的内容。

3　手动配置。初始化为空表，启动后使用显示的命令赋初值。

4　邻机获取。启动时推导出一个初始路由集合，推导的依据就是从相邻主机获得的路由表

初始路由表建立起来以后，路由器还要适应名字内容网络的变化情况。路在小型的、变化缓慢的内容网络中，管理者可以用手工方式来建立和更新名字路由表。而在大型的、迅速变化的网络环境下，人工更新的办法慢的不能接收。我们需要自动更新路由的方法。动态路由协议就是为了这个目的而设计。

3.1.2　动态距离矢量协议 NBRP

动态路由协议根据路由算法分为两类：

◎ 距离矢量路由协议，如 BGP，RIP 等

◎ 链路状态路由协议，如 OSPF，IS-IS 等。

第三章　基于名字的路由系统的路由策略

§ 3.1　基于名字路由系统的路由策略概括

路由策略负责为路由系统创建和刷新名字路由表。在基于名字的路由系统中，实现路由策略的是 NBRP（Name-based router protocol）协议。本章将围绕路由策略与 NBRP 协议具体展开。

3.1.1　路由表的建立

在上一章中，我们讨论了关于名字路由器时内容网络互连的内容。每个名字路由器与两个以上的实际内容网络相连，负责在这些网络之间转发数据报。它从一个网络接口接收数据报，选路之后通过另一个网络接口转发出去。除了那些查询域名命中 DNS 高速缓存的数据报，主机把数据报按照查询域名对应的内容服务器的位置将其向前转发。数据报传输过程中经过若干个路由器，最后到达与目的站处于同一内容网络上的内容服务器。因此，名字路由系统组成了内容网络的基本体系结构，处理所有的内容网络数据报文通信量。

同时，我们研究了名字路由器向前转发数据报所用的名字路由选路算法，其中提到了名字路由表的表驱动算法。在名字路由表中的每个项目都指出了一个目的提供特定内容服务的内容域名和到达最近的内容服务器所经过的下一路由器的地址。

虽然我们已经了解名字路由系统向前转发名字路由数据报的基本原理。但是我们并不知道名字路由器如何获得选路表中的信息。这个问题可以分为两个方面：

1　名字路由表中该有什么数据？

2　名字路由器如何获得这些数据？

问题的答案以依赖与内容网络体系结构的复杂度、范围大小和管理策略。名字路由策略负责解决这个问题。

一般来说，路由的估计包括了初始化和更新两个问题。每个路由器在启动时都必须建立一个初始路由集合，还要在路由改变时去更新它（如网络接口发生故障）。初始化的策略取决与系统的配置。它包括：

1　ROM 读取。启动时从 ROM 或者二级存储器（硬盘）读取初始名字选路表的内容并将其驻留在内存中。

2　网络配置。启动时从 TFTP/FTP 等网络接口从网络服务器读取初始名字选路表的内容。

3　手动配置。初始化为空表，启动后使用显示的命令赋初值。

4　邻机获取。启动时推导出一个初始路由集合，推导的依据就是从相邻主机获得的路由表

初始路由表建立起来以后，路由器还要适应名字内容网络的变化情况。路在小型的、变化缓慢的内容网络中，管理者可以用手工方式来建立和更新名字路由表。而在大型的、迅速变化的网络环境下，人工更新的办法慢的不能接收。我们需要自动更新路由的方法。动态路由协议就是为了这个目的而设计。

3.1.2　动态距离矢量协议 NBRP

动态路由协议根据路由算法分为两类：

◎ 距离矢量路由协议，如 BGP，RIP 等

◎ 链路状态路由协议，如 OSPF，IS-IS 等。

实现名字路由系统的路由策略的是 NBRP 协议。NBRP 在名字路由系统的扮演的角色类似与 IP 路由系统中的 BGP 协议。IP 路由系统中的 BGP 协议，不定期的在自治域系统之间，发布 IP 网络前缀地址，通告网络可达信息。与 BGP 类似，名字路由系统中的 NBRP 协议，将不定期的在名字路由器之间，发布对应特定内容服务的域名，并通告提供内容服务的内容服务器的可达性信息。

NBRP 协议属于动态距离矢量路由协议，同时丰富的属性使 NBRP 兼有链路状态路由的特点。图 3.1 给出了 NBRP 协议在 TCP/IP 的层次图。

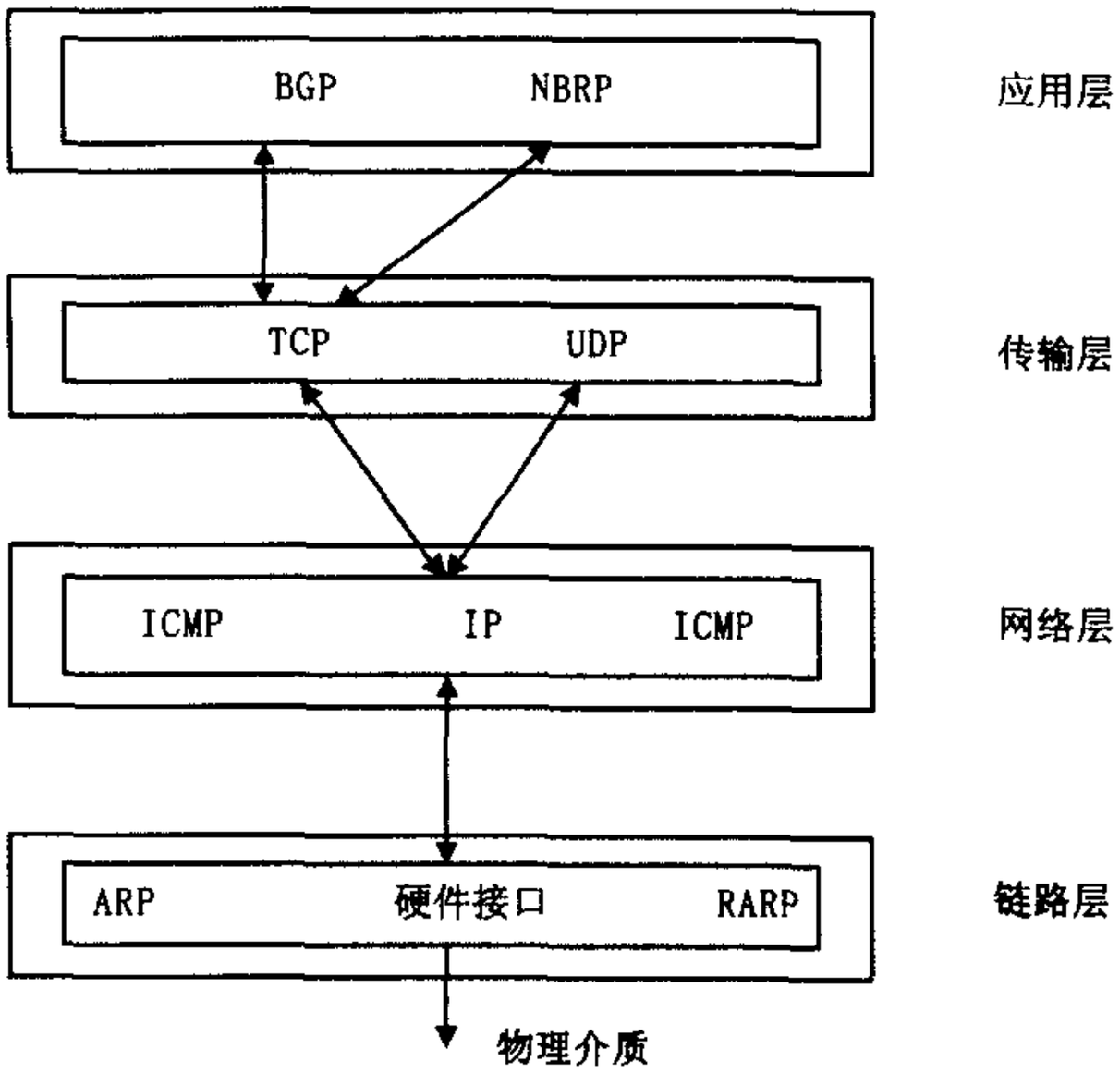


图 3.1　NBRP 协议与 TCP/IP 协议分层

序号	状态	含义
1	NR_PATH	记录了经过的路由器。值域由一系列的 NR 路径值组成。每个 NBR 路径由一个三元组<类型，长度，值>表示
2	NEXT-HOP	值域为四个字节，记录了下一跳的 IP 地址
3	LOCAL_PREF	值域为四个字节，用于区分到同一内容目的路由优先级
4	Aggregator	四个字节，表示形成名字聚合路由的最后一个路由器的 IP 地址。
5	Authentication Information	用于 NBRP 的报文解密机制，使 NBRP 通信安全

表 3.1 NBRP 属性表

NBRP 协议的特点：

◎ 使用面向连接的 TCP 作为其传输层协议，提高了协议的可靠性。

◎ 对网络拓扑没有限制，而且名字路由网络的拓扑本身就比较简单。

◎ 只有四种报文，结构简单。

◎ 路由更新时，NBRP 只发送增量路由，大大减少了 NBRP 传播路由所占用的带宽，适用于在名字路由网络上传播大量的名字路由信息。

◎ NBRP 名字路由携带了丰富的属性，由 NBRP 的路由策略来使用，供每个名字路由系统在入口和出口对路由进行过滤、选择和控制，使得 NBRP 简明、灵活、强大。表 3.1 给出了 NBRP 的属性。

NBRP 和 BGP 有很多相似的地方，比如都是动态矢量路由更新协议，目的都是控制路由更新报文的传播等。它们的区别在于：

1 NBRP 传播的是内容可达性，而 BGP 传播的是目的 IP 网络的可达性。

2 在 TCP/IP 中，尽管 NBRP 和 BGP 同处于应用层，但是 BGP 独立于 NBRP 协议，而 NBRP 依赖与 BGP。BGP 传播 IP 网络可达性，使得 IP 网络互通。名字路由系统依赖与 IP 路由系统，NBRP 的传播要在 IP 网络互通的基础上进行。

$ 3.2 NBRP 帧结构

本节描述 NBRP 的帧结构。NBRP 报文的传送是在可靠的传输协议上进行的。只有当整个报文收到以后，NBRP 才对报文进行处理。NBRP 的最大报文长度是 4096 字节，最小长度是 19 字节。

NBRP 共有四种报文类型：Open 报文、Update 报文、Notification 报文、Keepalive 报文。四种报文都有一个公共的消息头。

3.2.1 消息头：

NBRP 的每一个消息都有一个固定长度的消息头。图 3.2 给出了消息头的格式。

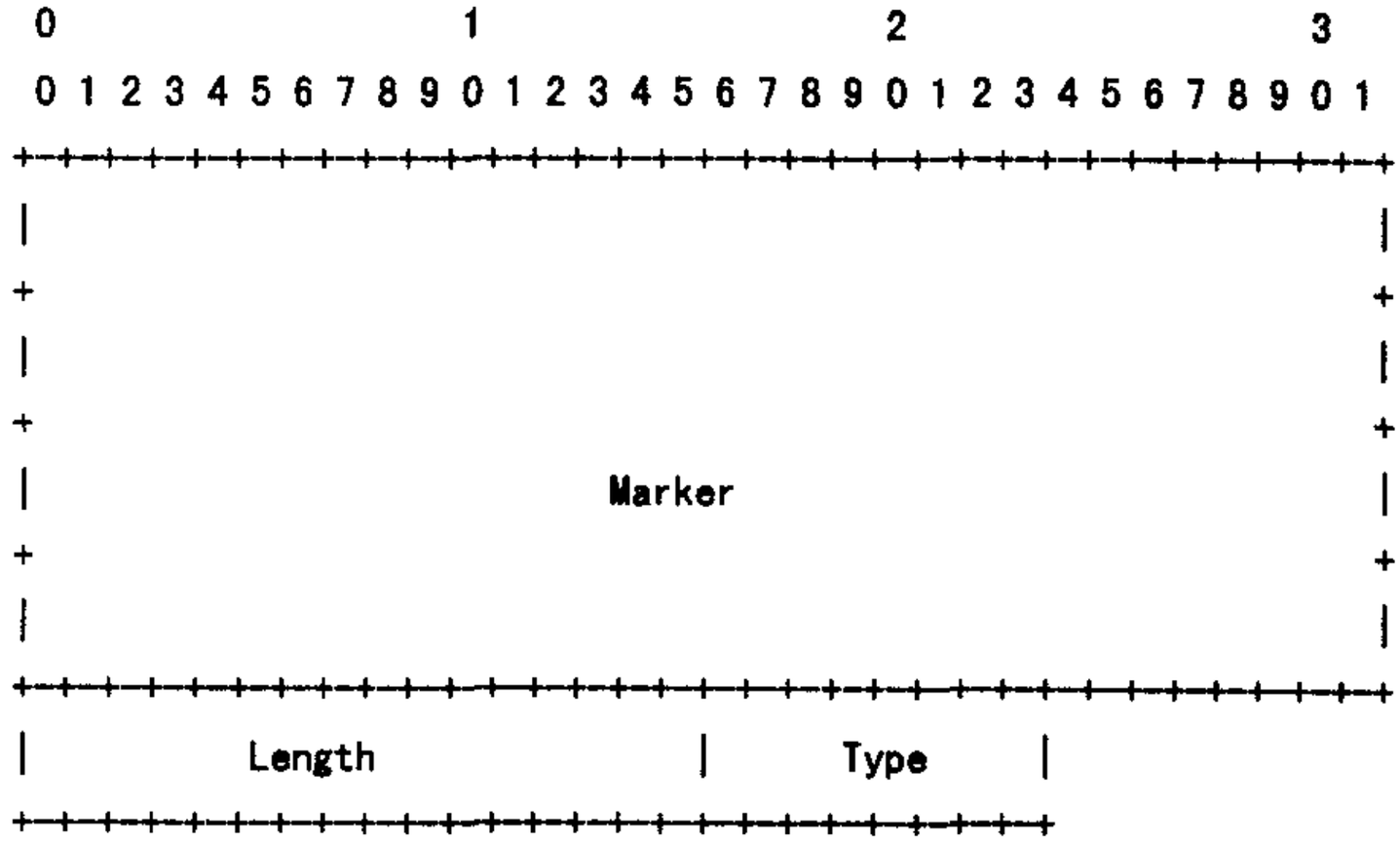


图 3.2 NBRP 的消息头

Marker 字段：

Marker 字段用来同步一对 NBRP 对等体的消息流，并对消息流加密。它占 16 个字节宽度。该字段的值取决于通信的 NBRP 对等体双方的协商结果。如果消息类型是 OPEN，或者 OPEN 消息没有携带任何加密信息，那么 Marker 字段的是各位全 1；如果 OPEN 消息制定加密机制，那么 Marker 的值取决于加密运算的结果。

Length 字段：

该字段指示了 NBRP 消息报文的长度（字节总数），包括消息头。该字段的值区间落于 19～4096 之间。

◎ 只有四种报文，结构简单。

◎ 路由更新时，NBRP 只发送增量路由，大大减少了 NBRP 传播路由所占用的带宽，适用于在名字路由网络上传播大量的名字路由信息。

◎ NBRP 名字路由携带了丰富的属性，由 NBRP 的路由策略来使用，供每个名字路由系统在入口和出口对路由进行过滤、选择和控制，使得 NBRP 简明、灵活、强大。表 3.1 给出了 NBRP 的属性。

NBRP 和 BGP 有很多相似的地方，比如都是动态矢量路由更新协议，目的都是控制路由更新报文的传播等。它们的区别在于：

1 NBRP 传播的是内容可达性，而 BGP 传播的是目的 IP 网络的可达性。

2 在 TCP/IP 中，尽管 NBRP 和 BGP 同处于应用层，但是 BGP 独立于 NBRP 协议，而 NBRP 依赖与 BGP。BGP 传播 IP 网络可达性，使得 IP 网络互通。名字路由系统依赖与 IP 路由系统，NBRP 的传播要在 IP 网络互通的基础上进行。

$ 3.2 NBRP 帧结构

本节描述 NBRP 的帧结构。NBRP 报文的传送是在可靠的传输协议上进行的。只有当整个报文收到以后，NBRP 才对报文进行处理。NBRP 的最大报文长度是 4096 字节，最小长度是 19 字节。

NBRP 共有四种报文类型：Open 报文、Update 报文、Notification 报文、Keepalive 报文。四种报文都有一个公共的消息头。

3.2.1 消息头：

NBRP 的每一个消息都有一个固定长度的消息头。图 3.2 给出了消息头的格式。

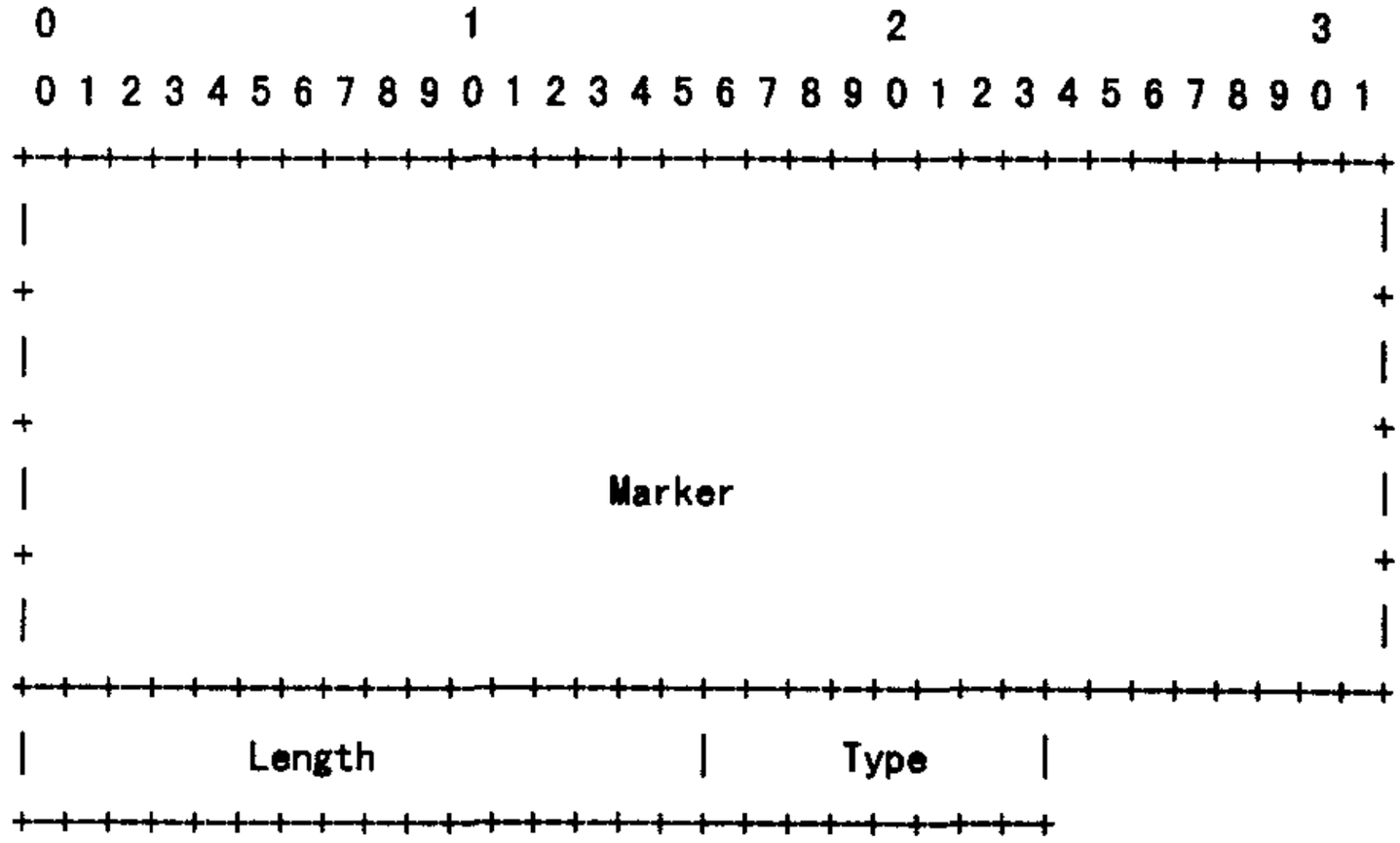


图 3.2 NBRP 的消息头

Marker 字段：

Marker 字段用来同步一对 NBRP 对等体的消息流，并对消息流加密。它占 16 个字节宽度。该字段的值取决于通信的 NBRP 对等体双方的协商结果。如果消息类型是 OPEN，或者 OPEN 消息没有携带任何加密信息，那么 Marker 字段的是各位全 1；如果 OPEN 消息制定加密机制，那么 Marker 的值取决于加密运算的结果。

Length 字段：

该字段指示了 NBRP 消息报文的长度（字节总数），包括消息头。该字段的值区间落于 19～4096 之间。

◎ 只有四种报文，结构简单。

◎ 路由更新时，NBRP 只发送增量路由，大大减少了 NBRP 传播路由所占用的带宽，适用于在名字路由网络上传播大量的名字路由信息。

◎ NBRP 名字路由携带了丰富的属性，由 NBRP 的路由策略来使用，供每个名字路由系统在入口和出口对路由进行过滤、选择和控制，使得 NBRP 简明、灵活、强大。表 3.1 给出了 NBRP 的属性。

NBRP 和 BGP 有很多相似的地方，比如都是动态矢量路由更新协议，目的都是控制路由更新报文的传播等。它们的区别在于：

1 NBRP 传播的是内容可达性，而 BGP 传播的是目的 IP 网络的可达性。

2 在 TCP/IP 中，尽管 NBRP 和 BGP 同处于应用层，但是 BGP 独立于 NBRP 协议，而 NBRP 依赖与 BGP。BGP 传播 IP 网络可达性，使得 IP 网络互通。名字路由系统依赖与 IP 路由系统，NBRP 的传播要在 IP 网络互通的基础上进行。

$ 3.2 NBRP 帧结构

本节描述 NBRP 的帧结构。NBRP 报文的传送是在可靠的传输协议上进行的。只有当整个报文收到以后，NBRP 才对报文进行处理。NBRP 的最大报文长度是 4096 字节，最小长度是 19 字节。

NBRP 共有四种报文类型：Open 报文、Update 报文、Notification 报文、Keepalive 报文。四种报文都有一个公共的消息头。

3.2.1 消息头：

NBRP 的每一个消息都有一个固定长度的消息头。图 3.2 给出了消息头的格式。

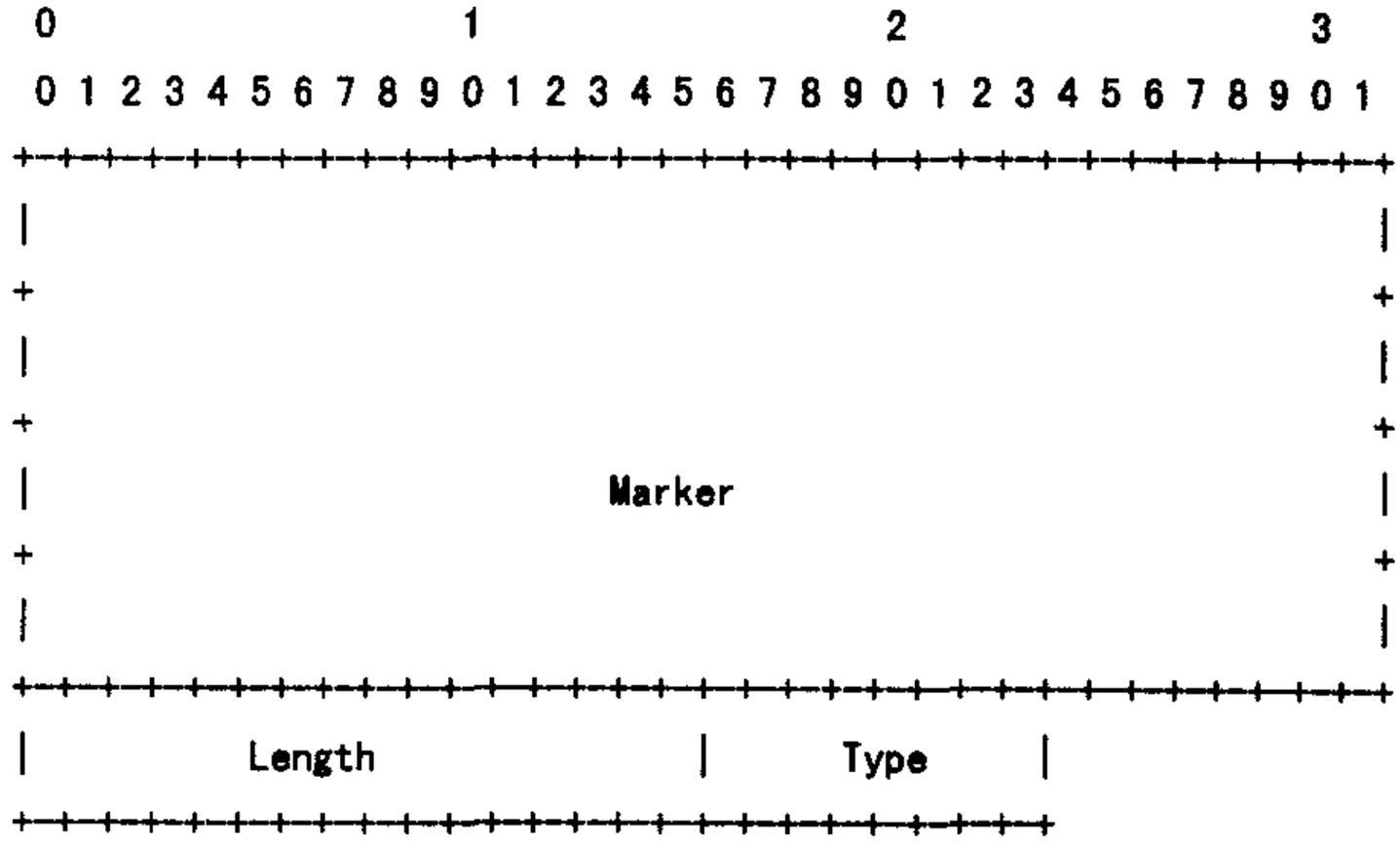


图 3.2 NBRP 的消息头

Marker 字段：

Marker 字段用来同步一对 NBRP 对等体的消息流，并对消息流加密。它占 16 个字节宽度。该字段的值取决于通信的 NBRP 对等体双方的协商结果。如果消息类型是 OPEN，或者 OPEN 消息没有携带任何加密信息，那么 Marker 字段的是各位全 1；如果 OPEN 消息制定加密机制，那么 Marker 的值取决于加密运算的结果。

Length 字段：

该字段指示了 NBRP 消息报文的长度（字节总数），包括消息头。该字段的值区间落于 19～4096 之间。

Type 字段段：

该字段的值标志报文类型。NBRP 支持四种报文类型：

1 — OPEN

2 — UPDATE

3 — NOTIFICATION

4 — KEEPLIVE

下面我们将各个类型报文进行简单介绍。

3.2.2 OPEN 报文格式

在一对 NBRP 的对等体建立 TCP 连接后，发送的第一个 NBRP 消息就是 OPEN 消息。图 3.3 给出了 OPEN 消息的格式。如果 Open 报文可以接收，需要发回一个 Keepalive 报文来确认 Open 报文。一旦确认了 Open 报文，即可以开始交换 Update、Keepalive 和 Notification 报文。

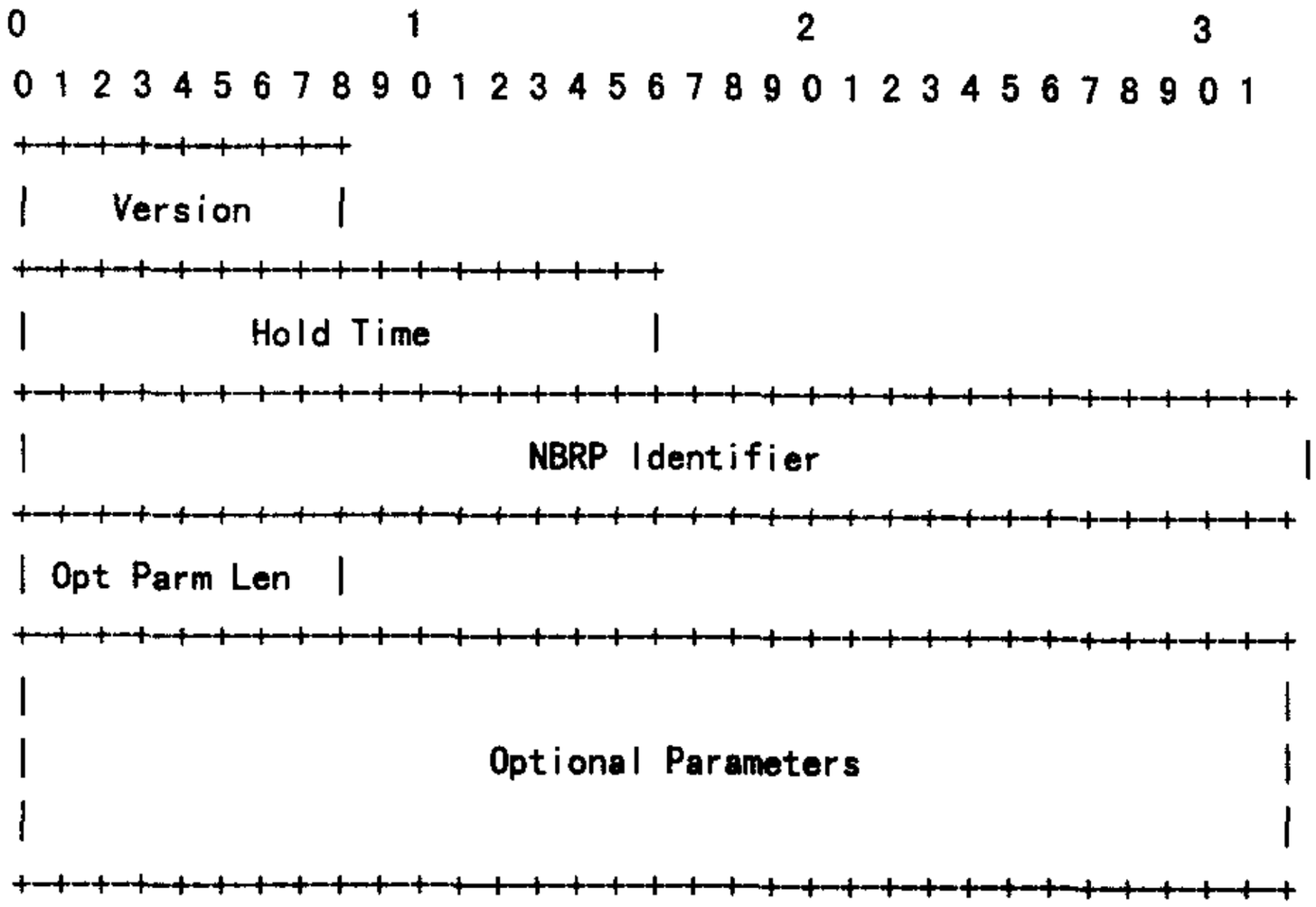


图 3.3 OPEN 消息格式

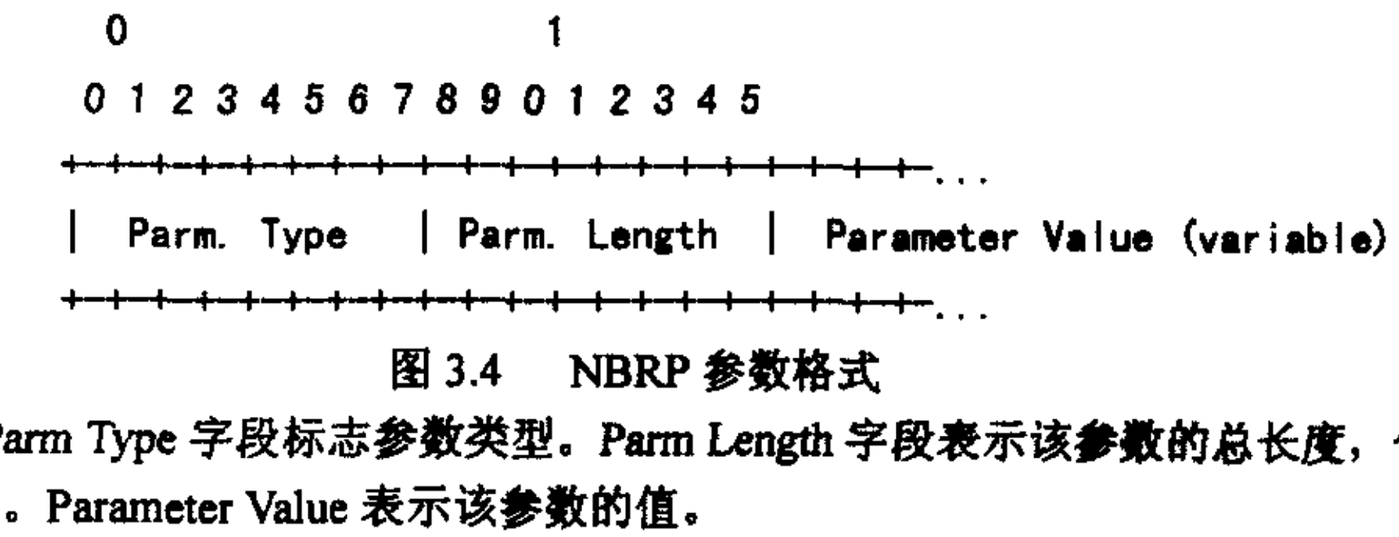


图 3.4 NBRP 参数格式

Parm Type 字段标志参数类型。Parm Length 字段表示该参数的总长度，包括参数的三个字段。Parameter Value 表示该参数的值。

Type 字段段：

该字段的值标志报文类型。NBRP 支持四种报文类型：

1 — OPEN

2 — UPDATE

3 — NOTIFICATION

4 — KEEPLIVE

下面我们将各个类型报文进行简单介绍。

3.2.2 OPEN 报文格式

在一对 NBRP 的对等体建立 TCP 连接后，发送的第一个 NBRP 消息就是 OPEN 消息。图 3.3 给出了 OPEN 消息的格式。如果 Open 报文可以接收，需要发回一个 Keepalive 报文来确认 Open 报文。一旦确认了 Open 报文，即可以开始交换 Update、Keepalive 和 Notification 报文。

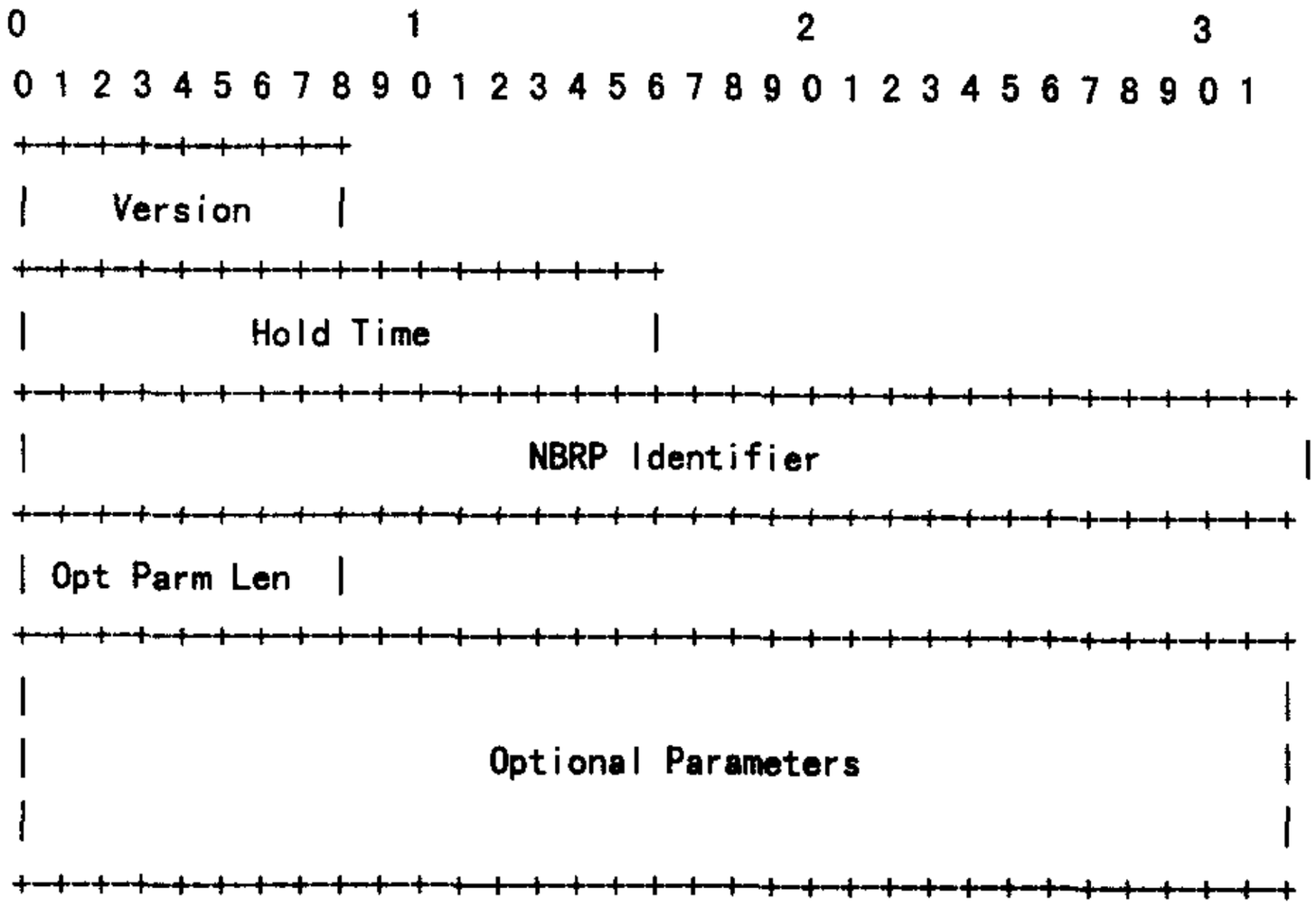


图 3.3 OPEN 消息格式

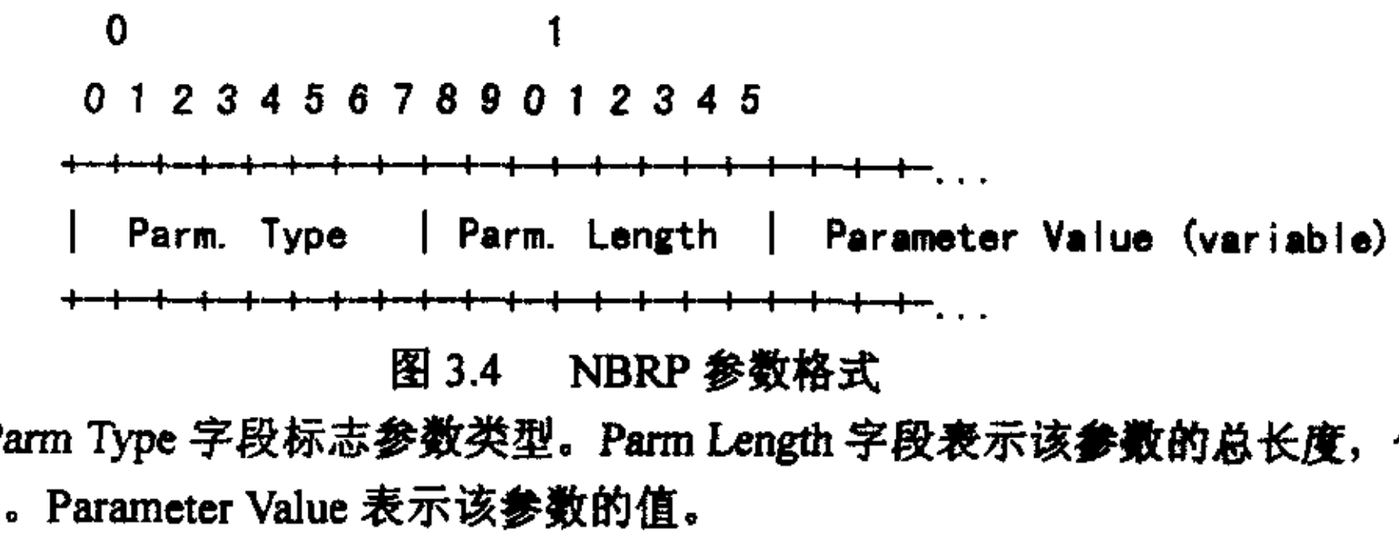


图 3.4 NBRP 参数格式

Parm Type 字段标志参数类型。Parm Length 字段表示该参数的总长度，包括参数的三个字段。Parameter Value 表示该参数的值。

字段	含义
版本（Version）	指示当前 NBRP 版本号，1 字节整数
保持时间(Hold Timer)	表示两个连续 NBRP 消息报文之间的最大时间间隙。主机将根据 Hold Time 值决定是否断掉 TCP 连接。该字段表示发送方建议的 Hold Timer 值，计量单位为秒。主机将根据发送方建议的 Hold Timer 值和本地策略确定该对连接的 Hold Timer 值。Hold Timer 的值要么为 0，要么大于等于 3 秒。
标志符（NBRP Identifier）	标志 NBRP 发送方，长度为四个字节。在 IPV4 的环境下，一般设为发送方本身的 IP 地址。
可选参数长度（Optional para len）	指示选项参数字段（Optional Parameters）的总长度
可选参数（Optional parameter）	包含了一个参数属性列表。每个参数是一个三元组，如图 3.5

表 3.2 Open 报文字段说明

3.2.3 Update 报文

NBRP 协议的核心是名字路由更新。路由更新包括 NBRP 用来组建无循环的内容网络结构所需的所有信息。更新报文包含的基本部分是：内容层可达性信息（CLRI）、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内容可行性路由，或者撤销多条内容不可行路由，也可以同时发布一条可行性内容路由和撤销多条内容不可行路由。Update 报文除了包含固定长度的报头外，还包含其他字段，如图 3.5 所示。

表 3.3 描述了 Update 报文各个字段。

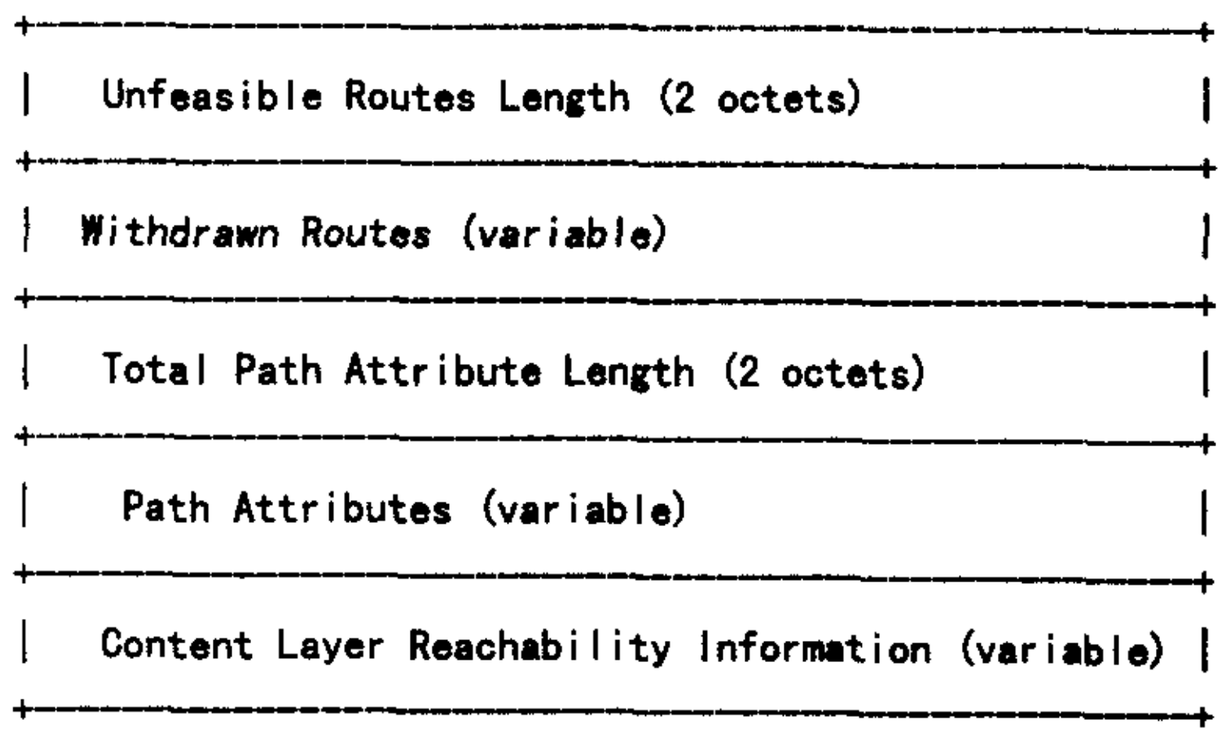


图 3.5 UPDATE 报文格式

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 字节为不可行名字路由长度、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 mi 名字路由。

一个 Update 报文可以列出从服务器撤回的多条路径。每一个这样的路由被它的一对内容标志（内容域名，内容服务器 IP 地址）识别，即在 NBRP 发言人广播的报文中的撤销路由域中确切地定义这些路由。然后，NBRP 发言人再与之前广播的路由相连。也可以只广播从

字段	含义
版本（Version）	指示当前 NBRP 版本号，1 字节整数
保持时间(Hold Timer)	表示两个连续 NBRP 消息报文之间的最大时间间隙。主机将根据 Hold Time 值决定是否断掉 TCP 连接。该字段表示发送方建议的 Hold Timer 值，计量单位为秒。主机将根据发送方建议的 Hold Timer 值和本地策略确定该对连接的 Hold Timer 值。Hold Timer 的值要么为 0，要么大于等于 3 秒。
标志符（NBRP Identifier）	标志 NBRP 发送方，长度为四个字节。在 IPV4 的环境下，一般设为发送方本身的 IP 地址。
可选参数长度（Optional para len）	指示选项参数字段（Optional Parameters）的总长度
可选参数（Optional parameter）	包含了一个参数属性列表。每个参数是一个三元组，如图 3.5

表 3.2 Open 报文字段说明

3.2.3 Update 报文

NBRP 协议的核心是名字路由更新。路由更新包括 NBRP 用来组建无循环的内容网络结构所需的所有信息。更新报文包含的基本部分是：内容层可达性信息（CLRI）、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内容可行性路由，或者撤销多条内容不可行路由，也可以同时发布一条可行性内容路由和撤销多条内容不可行路由。Update 报文除了包含固定长度的报头外，还包含其他字段，如图 3.5 所示。

表 3.3 描述了 Update 报文各个字段。

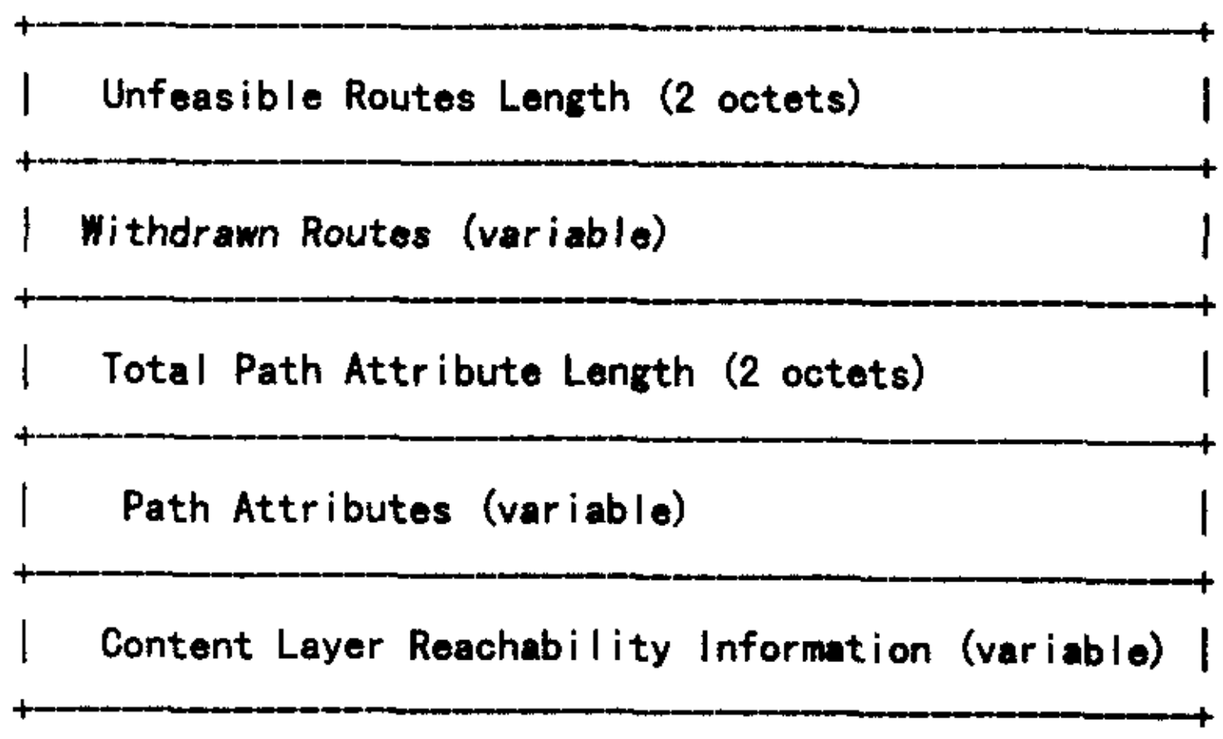


图 3.5 UPDATE 报文格式

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 字节为不可行名字路由长度、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 mi 名字路由。

一个 Update 报文可以列出从服务器撤回的多条路径。每一个这样的路由被它的一对内容标志（内容域名，内容服务器 IP 地址）识别，即在 NBRP 发言人广播的报文中的撤销路由域中确切地定义这些路由。然后，NBRP 发言人再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内容层可到达信息。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。

字段名	含义
撤销路由长度（Unfeasible Routes Length）	2 字节的无符号整数，表示以字节计数的整个撤销路由字段的长度
撤销路由 Withdrawn Routes	长度可变。表示内容域名。内容域名的格式遵循 DNS 的域名格式，它是一个或多个标识符的序列。每个标识符以首字节的计数值来说明随后标识符的字节长度。每个名字以最后字节为 0 结束，长度为 0 的标识符是根标识符。
总的路径属性长度（Total Path Attribute Length）	2 字节的无符号整数，标志路径属性字段的字节长度
路径属性 Path Attributes	此可变长度字段包含了与内容层可达性信息字段相关连的 NBRP 属性表。路径属性包括了一些如优先级等信息，用于过滤和路由选择过程
内容层可达信息 CLRI	长度可变，包含了通告的内容域名，它的表示方法同样遵循 DNS 的域名格式。

表 3.3 Update 报文各个字段

3.2.3 Keepalive 报文

NBRP 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(Hold Timer)不超时。Keepalive 周期性地在对等体间交换地报文，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 NBRP 报文头，后面没有数据。

Keepalive 报文以保证在保持时间（Hold Timer）内不溢出的速率发送。建议的速率是保持时间(Hold Timer)间隔的三分之一，即 30s。如果保持时间间隔为零，则不用发送周期性的 Keepalive 报文。消息的发送频率不应该超过每两秒一次。

3.2.4 Notification 报文

任何时候检测到错误，NBRP 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 NBRP 连接。除了定长的 NBRP 报文头，Notification 报文包含其他的几个字段，如图 3.6 所示。

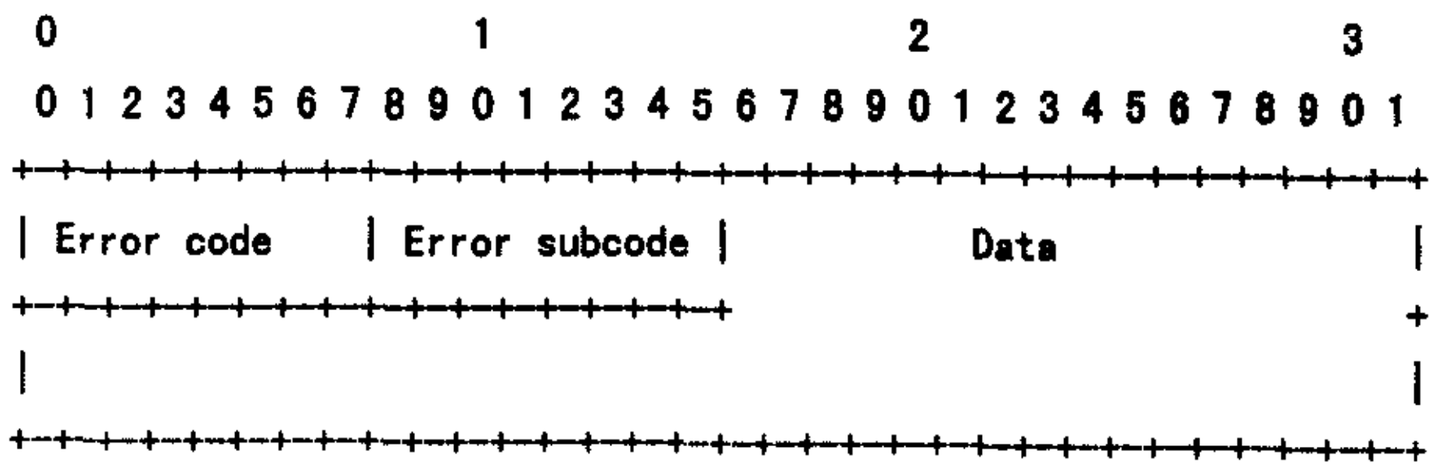


图 3.6 Notification 报文

服务器上撤回的路由，它不包括路径属性或者内容层可到达信息。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。

字段名	含义
撤销路由长度（Unfeasible Routes Length）	2 字节的无符号整数，表示以字节计数的整个撤销路由字段的长度
撤销路由 Withdrawn Routes	长度可变。表示内容域名。内容域名的格式遵循 DNS 的域名格式，它是一个或多个标识符的序列。每个标识符以首字节的计数值来说明随后标识符的字节长度。每个名字以最后字节为 0 结束，长度为 0 的标识符是根标识符。
总的路径属性长度（Total Path Attribute Length）	2 字节的无符号整数，标志路径属性字段的字节长度
路径属性 Path Attributes	此可变长度字段包含了与内容层可达性信息字段相关连的 NBRP 属性表。路径属性包括了一些如优先级等信息，用于过滤和路由选择过程
内容层可达信息 CLRI	长度可变，包含了通告的内容域名，它的表示方法同样遵循 DNS 的域名格式。

表 3.3 Update 报文各个字段

3.2.3 Keepalive 报文

NBRP 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(Hold Timer)不超时。Keepalive 周期性地在对等体间交换地报文，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 NBRP 报文头，后面没有数据。

Keepalive 报文以保证在保持时间（Hold Timer）内不溢出的速率发送。建议的速率是保持时间(Hold Timer)间隔的三分之一，即 30s。如果保持时间间隔为零，则不用发送周期性的 Keepalive 报文。消息的发送频率不应该超过每两秒一次。

3.2.4 Notification 报文

任何时候检测到错误，NBRP 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 NBRP 连接。除了定长的 NBRP 报文头，Notification 报文包含其他的几个字段，如图 3.6 所示。

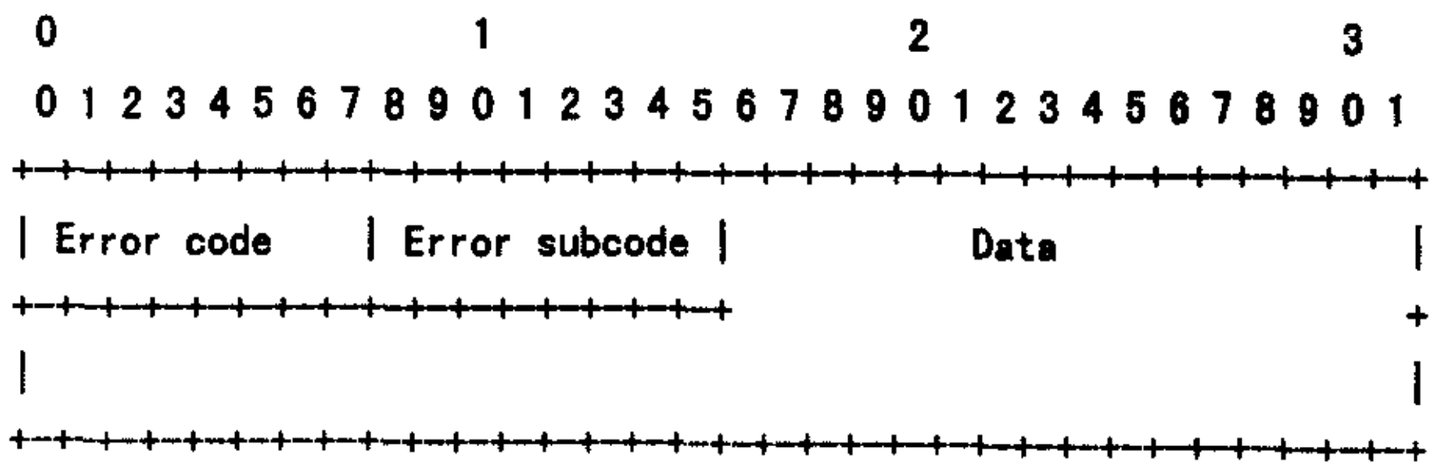


图 3.6 Notification 报文

服务器上撤回的路由，它不包括路径属性或者内容层可到达信息。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。

字段名	含义
撤销路由长度（Unfeasible Routes Length）	2 字节的无符号整数，表示以字节计数的整个撤销路由字段的长度
撤销路由 Withdrawn Routes	长度可变。表示内容域名。内容域名的格式遵循 DNS 的域名格式，它是一个或多个标识符的序列。每个标识符以首字节的计数值来说明随后标识符的字节长度。每个名字以最后字节为 0 结束，长度为 0 的标识符是根标识符。
总的路径属性长度（Total Path Attribute Length）	2 字节的无符号整数，标志路径属性字段的字节长度
路径属性 Path Attributes	此可变长度字段包含了与内容层可达性信息字段相关连的 NBRP 属性表。路径属性包括了一些如优先级等信息，用于过滤和路由选择过程
内容层可达信息 CLRI	长度可变，包含了通告的内容域名，它的表示方法同样遵循 DNS 的域名格式。

表 3.3 Update 报文各个字段

3.2.3 Keepalive 报文

NBRP 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(Hold Timer)不超时。Keepalive 周期性地在对等体间交换地报文，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 NBRP 报文头，后面没有数据。

Keepalive 报文以保证在保持时间（Hold Timer）内不溢出的速率发送。建议的速率是保持时间(Hold Timer)间隔的三分之一，即 30s。如果保持时间间隔为零，则不用发送周期性的 Keepalive 报文。消息的发送频率不应该超过每两秒一次。

3.2.4 Notification 报文

任何时候检测到错误，NBRP 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 NBRP 连接。除了定长的 NBRP 报文头，Notification 报文包含其他的几个字段，如图 3.6 所示。

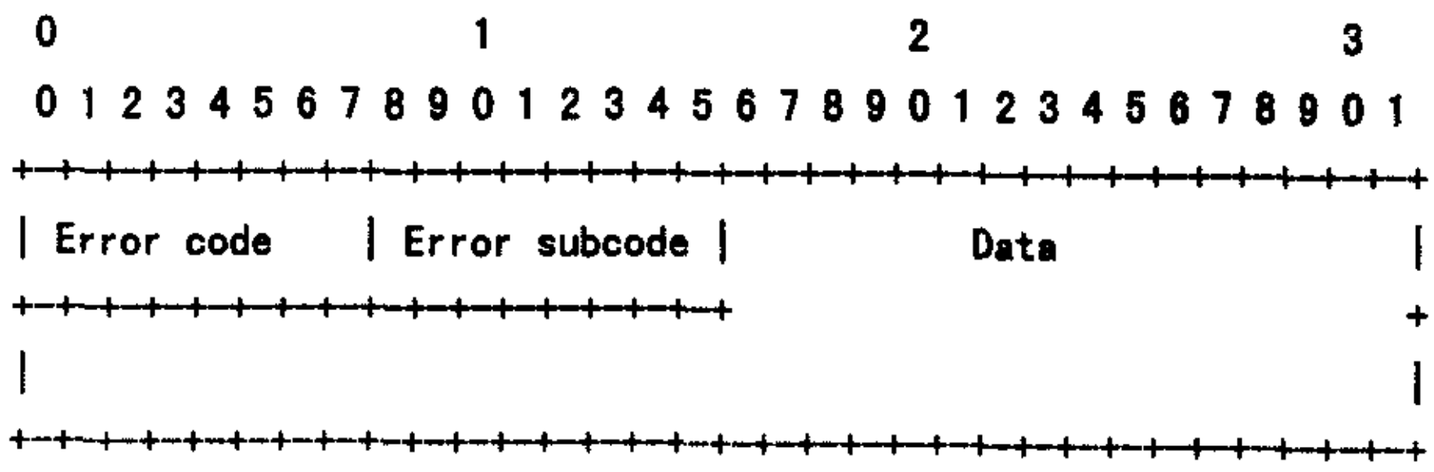


图 3.6 Notification 报文

错误码	符号名称
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

表 3.4 Notification 报文错误码

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	对端名字路由器错误
	3	非法的 NBRP 标志符
	4	不支持的选项参数
	5	认证失败
	6	不支持的保持时间长
3：Update 报文错误	1	畸形属性列表
	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 ORIGIN 属性
	7	NBR 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 NBR_PATH
其他	无	无

表 3.5 错误码信息表

图 3.6 中，错误码是 1 字节的无符号整数，标志 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（Error Code）和错误子码（Error subcode）而定，用来诊断发送 Notification 报文的原因。

错误子码（Error subcode）给出了更具体的错误信息，见表 3.5。每个错误码可以有一个或多个相关的错误子码。错误码定义列于表 3.4。

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。

§ 3.3 NBRP 协议状态机

NBRP 的通信过程，可以用一个确定有限状态机（FSM）来描述。本节介绍 NBRP 的状态机。我们将以有限状态机(FSM)的方式描述 NBRP 的操作，即通过 FSM 定义的状态，观察概括 NBRP 的操作。需要指出的是，NBRP 的 FSM 的各种状态及事件是针对每个 NBRP 对等体来说的。

NBRP 共有 6 种状态、13 种事件。图 3.7 给出了 NBRP 的协议状态机图。表 3.6、表 3.7 给出了 NBRP 的状态、事件列表。

序号	状态	含义
1	Idle	NBRP 最开始处于空闲状态
2	Connect	NBRP 正在等待传输层（TCP）协议连接的完成
3	Active	NBRP 正通过启动 TCP 连接获得一个对等体
4	OpenSent	正等待对等体发来 Open 报文。此状态下，NBRP 已经向该对等体发送过来的报文
5	OpenConfirm	NBRP 在收到 Open 报文后，等待对等体发送 Keepalive 或者 Notification 报文
6	Established	对等体协商的最后阶段。在这个阶段，NBRP 开始与对等体交互 Update 数据报。

表 3.6 NBRP 状态列表

序号	事件	含义
1	NBRP Start	NBRP 启动
2	NBRP Stop	NBRP 停止
3	NBRP Transport connection open	NBRP 传输连接打开
4	NBRP Transport connection closed	NBRP 传输连接关闭
5	NBRP Transport connection open failed	NBRP 传输连接打开失败
6	NBRP Transport fatal error	NBRP 传输重大错误
7	NBRP Connect Retry timer expired	Connect Retry 定时器超时
8	Hold Timer expired	Hold 定时器超时
9	KeepAlive timer expired	Keep alive 定时器超时
10	Receive OPEN message	收到 Open 报文

图 3.6 中，错误码是 1 字节的无符号整数，标志 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（Error Code）和错误子码（Error subcode）而定，用来诊断发送 Notification 报文的原因。

错误子码（Error subcode）给出了更具体的错误信息，见表 3.5。每个错误码可以有一个或多个相关的错误子码。错误码定义列于表 3.4。

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。

§ 3.3 NBRP 协议状态机

NBRP 的通信过程，可以用一个确定有限状态机（FSM）来描述。本节介绍 NBRP 的状态机。我们将以有限状态机(FSM)的方式描述 NBRP 的操作，即通过 FSM 定义的状态，观察概括 NBRP 的操作。需要指出的是，NBRP 的 FSM 的各种状态及事件是针对每个 NBRP 对等体来说的。

NBRP 共有 6 种状态、13 种事件。图 3.7 给出了 NBRP 的协议状态机图。表 3.6、表 3.7 给出了 NBRP 的状态、事件列表。

序号	状态	含义
1	Idle	NBRP 最开始处于空闲状态
2	Connect	NBRP 正在等待传输层（TCP）协议连接的完成
3	Active	NBRP 正通过启动 TCP 连接获得一个对等体
4	OpenSent	正等待对等体发来 Open 报文。此状态下，NBRP 已经向该对等体发送过来的报文
5	OpenConfirm	NBRP 在收到 Open 报文后，等待对等体发送 Keepalive 或者 Notification 报文
6	Established	对等体协商的最后阶段。在这个阶段，NBRP 开始与对等体交互 Update 数据报。

表 3.6 NBRP 状态列表

序号	事件	含义
1	NBRP Start	NBRP 启动
2	NBRP Stop	NBRP 停止
3	NBRP Transport connection open	NBRP 传输连接打开
4	NBRP Transport connection closed	NBRP 传输连接关闭
5	NBRP Transport connection open failed	NBRP 传输连接打开失败
6	NBRP Transport fatal error	NBRP 传输重大错误
7	NBRP Connect Retry timer expired	Connect Retry 定时器超时
8	Hold Timer expired	Hold 定时器超时
9	KeepAlive timer expired	Keep alive 定时器超时
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

表 3.7 NBRP 事件列表

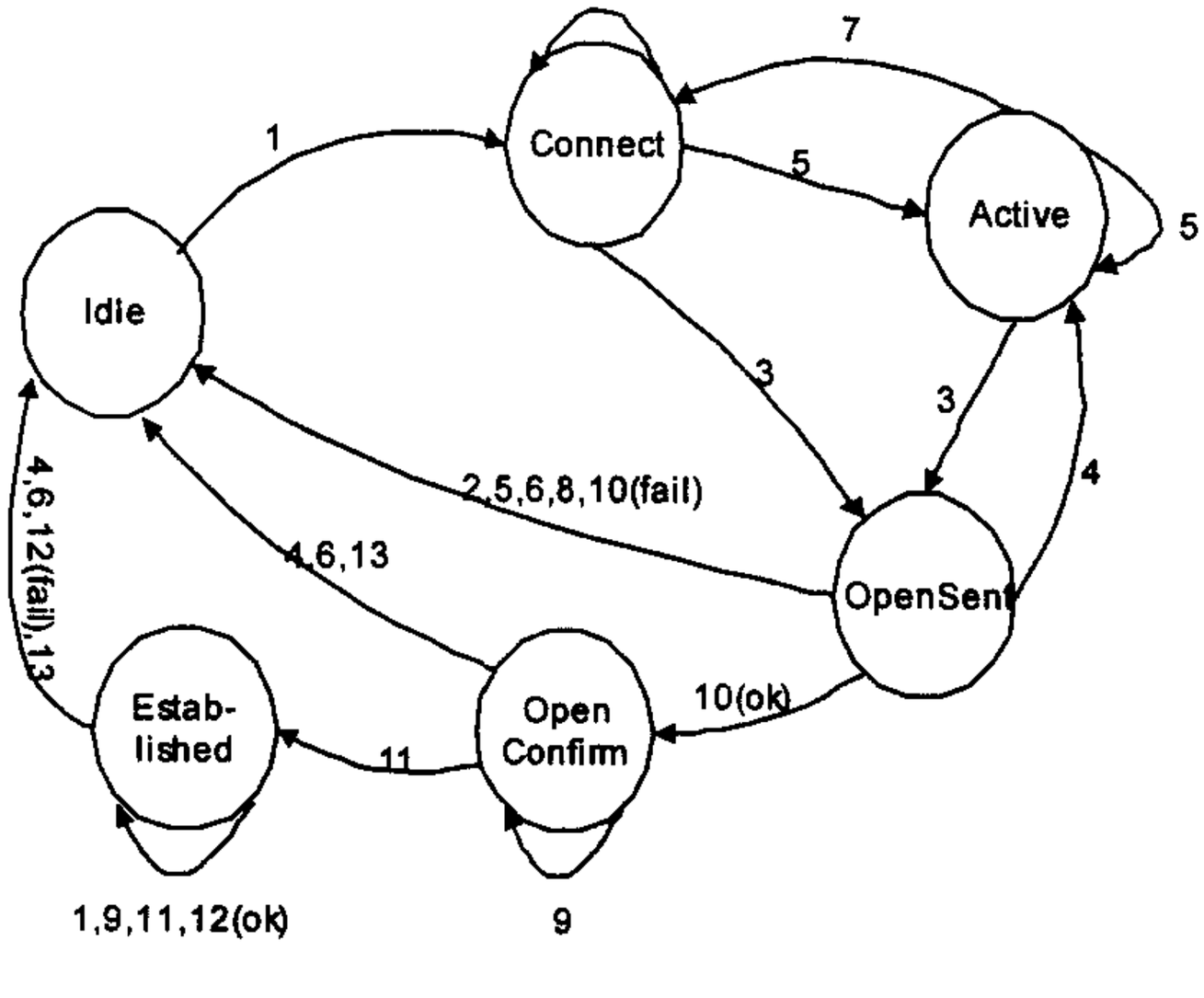


图 3.7 NBRP 的协议状态机

接下来，将对 NBRP 状态机的各种状态之间的转换给予详细描述。

3.3.1 Idle 状态（空闲状态）

最初 NBRP 处于空闲状态。在该状态下，NBRP 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 NBRP 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 NBRP 发言人发现有错误，就关闭连接并回到 Idle 状态。NBRP 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。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 NBRP 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。建议初始时间值为 60s。

在空闲状态下的收到的任何其他事件都将被忽略。

3.3.2 Connect 状态（连接状态）

在这个状态下，NBRP 正在等待传输层协议连接的完成。

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

表 3.7 NBRP 事件列表

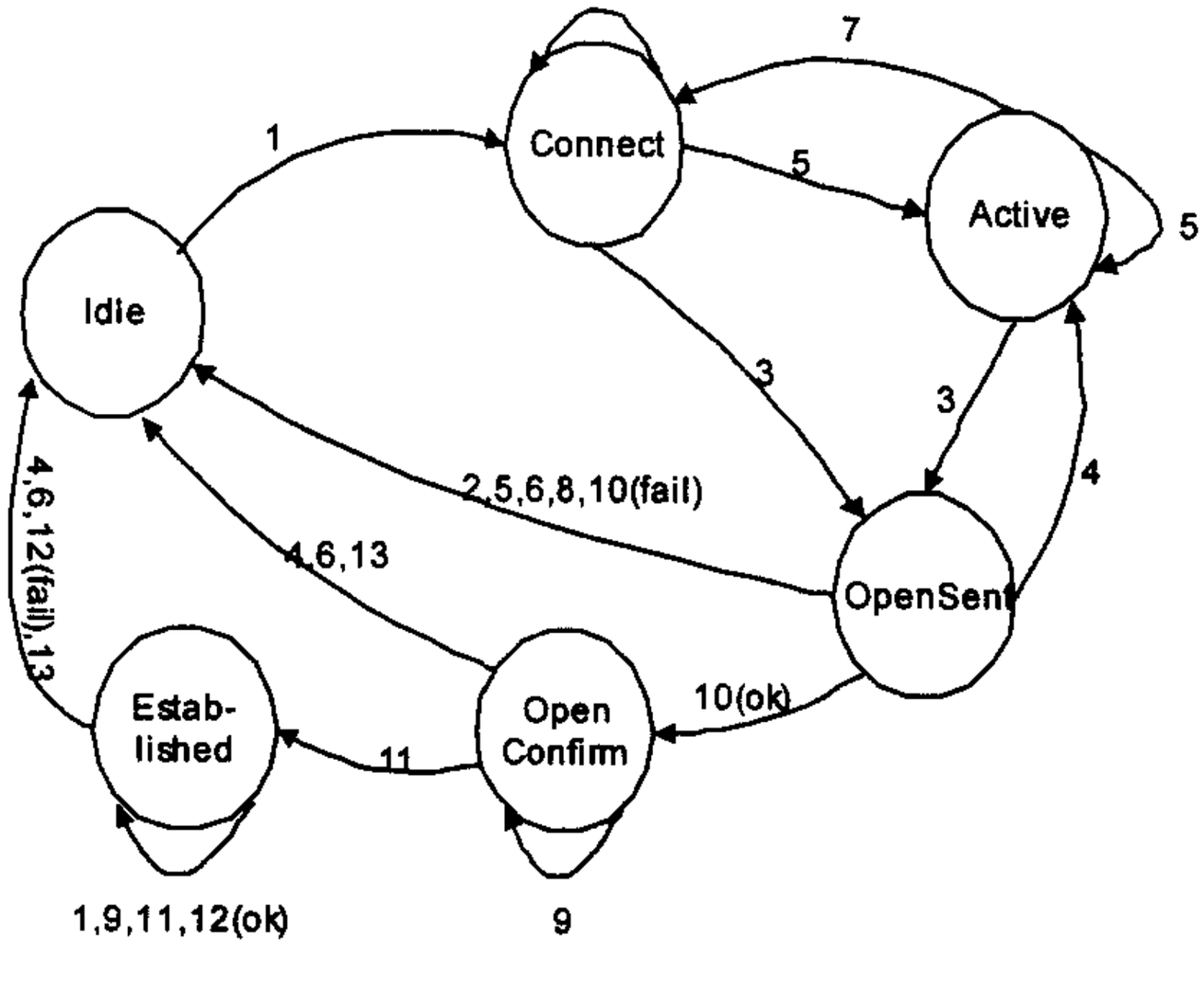


图 3.7 NBRP 的协议状态机

接下来，将对 NBRP 状态机的各种状态之间的转换给予详细描述。

3.3.1 Idle 状态（空闲状态）

最初 NBRP 处于空闲状态。在该状态下，NBRP 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 NBRP 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 NBRP 发言人发现有错误，就关闭连接并回到 Idle 状态。NBRP 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。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 NBRP 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。建议初始时间值为 60s。

在空闲状态下的收到的任何其他事件都将被忽略。

3.3.2 Connect 状态（连接状态）

在这个状态下，NBRP 正在等待传输层协议连接的完成。

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

表 3.7 NBRP 事件列表

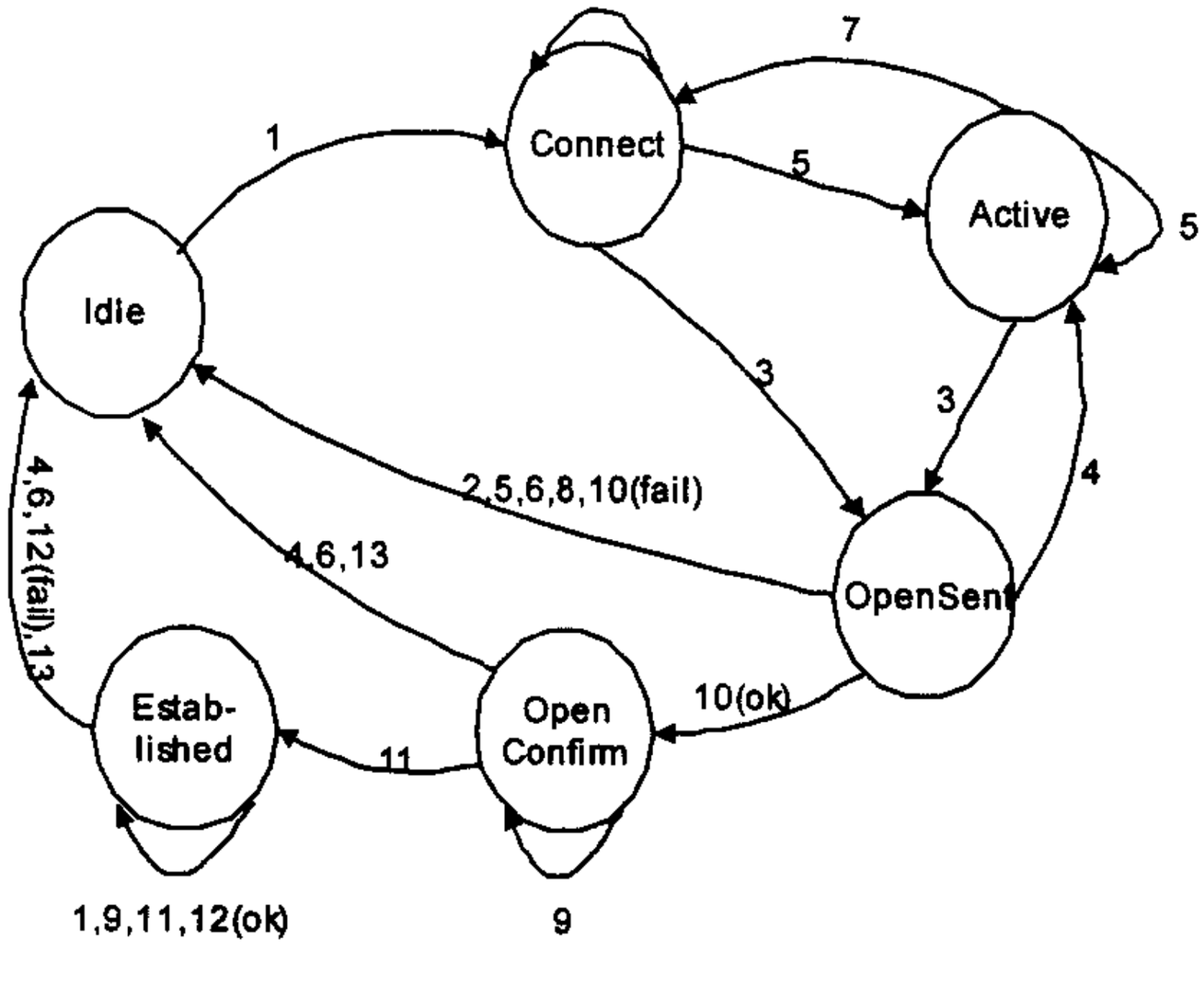


图 3.7 NBRP 的协议状态机

接下来，将对 NBRP 状态机的各种状态之间的转换给予详细描述。

3.3.1 Idle 状态（空闲状态）

最初 NBRP 处于空闲状态。在该状态下，NBRP 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 NBRP 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 NBRP 发言人发现有错误，就关闭连接并回到 Idle 状态。NBRP 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。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 NBRP 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。建议初始时间值为 60s。

在空闲状态下的收到的任何其他事件都将被忽略。

3.3.2 Connect 状态（连接状态）

在这个状态下，NBRP 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 NBRP 对等体是否已主动和它建立联系，同时转换到 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，并启动一个与其他 NBRP 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 NBRP 资源，并回到 Idle 状态。

3.3.3 Active 状态（激活状态）

在这个状态下，NBRP 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，NBRP 回到连接状态。NBRP 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 NBRP 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的转变，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。

启动事件在激活状态下被忽略。

3.3.4 Open Sent 状态

在这个状态下，NBRP 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名字路由器地址，本地系统就发送 Notification 报文，并回到空闲状态。如果没有出错，NBRP 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 Timer）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。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，NBRP 就发送一个带有响应错误代码的 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 NBRP 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。

3.3.5 Open Confirm 状态

在这个状态下，NBRP 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

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 NBRP 对等体是否已主动和它建立联系，同时转换到 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，并启动一个与其他 NBRP 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 NBRP 资源，并回到 Idle 状态。

3.3.3 Active 状态（激活状态）

在这个状态下，NBRP 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，NBRP 回到连接状态。NBRP 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 NBRP 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的转变，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。

启动事件在激活状态下被忽略。

3.3.4 Open Sent 状态

在这个状态下，NBRP 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名字路由器地址，本地系统就发送 Notification 报文，并回到空闲状态。如果没有出错，NBRP 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 Timer）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。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，NBRP 就发送一个带有响应错误代码的 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 NBRP 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。

3.3.5 Open Confirm 状态

在这个状态下，NBRP 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

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 NBRP 对等体是否已主动和它建立联系，同时转换到 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，并启动一个与其他 NBRP 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 NBRP 资源，并回到 Idle 状态。

3.3.3 Active 状态（激活状态）

在这个状态下，NBRP 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，NBRP 回到连接状态。NBRP 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 NBRP 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的转变，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。

启动事件在激活状态下被忽略。

3.3.4 Open Sent 状态

在这个状态下，NBRP 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名字路由器地址，本地系统就发送 Notification 报文，并回到空闲状态。如果没有出错，NBRP 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 Timer）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。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，NBRP 就发送一个带有响应错误代码的 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 NBRP 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。

3.3.5 Open Confirm 状态

在这个状态下，NBRP 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。

如果系统收到一个断开连接通知，本地系统改变自己的状态到空闲。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。

3.3.6 Established 状态（已建立）

这是对等体协商的最后阶段。在这个阶段，NBRP 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，并且重新启动它的 Keepalive 定时器。每次本地系统发送 Keepalive 或者更新消息时，也重新启动它的 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。只要 NBRP 的状态有确定到空闲，它关闭 NBRP（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中被忽略。

§ 3.4 NBRP 路由处理

NBRP 是一个用于名字路由系统之间的路由协议。与任何的路由协议一样，NBRP 维护路由表，传播路由的刷新信息并且根据路由度量做出路由选择。NBRP 的主要功能是与其他的 NBRP 系统交换内容层可达性信息。内容层可达性信息 CLRI 中包含了可达性信息所经过的名字路由器列表，从而构造了一个名字路由器连接图，以避免路由环路，同时也使得基于名字路由器级别的策略控制成为可能。从这一点上讲，NBRP 是一个综合了向量一距离算法和链路一状态算法的协议。

本节着重讨论 NBRP 路由处理过程。我们首先对 NBRP 路由的相关理论作了阐述。Update 报文的处理进程是整个 NBRP 路由处理的核心，在本文的第二小节中，我们结合 NBRP 的协议细节和路由理论，对它进行了详细的讨论。最后，我们给出了 NBRP 的性能仿真结果，并进行了简单的分析。

3.4.1 NBRP 路由的相关理论

※ NBRP 的路由定义

我们可以将一个互联的自治系统抽象成 S＝(N, G)，其中 N 是所有内容网络系统的集合，G ＝ (V, E)是一个 NBRP 会话连接图。V 是系统中所有 NBRP 发言人的集合，E 是所有 NBRP 会话的集合。抽象的名字路由系统的网络拓扑图如图 3.8 所示。

图 3.8 中的 v 代表名字路由器，s 代笔内容服务器群。每一个名字路由系统由名字路由器和与它直接相连的内容服务器群组成。内容服务器群也可以通过内容交换机与名字路由器相连。名字路由器 v 是每一个路由系统的发言人和管理者。每一个路由系统都可以根据本地

发送 Notification 报文，改变自己的状态到空闲状态。

如果系统收到一个断开连接通知，本地系统改变自己的状态到空闲。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。

3.3.6 Established 状态（已建立）

这是对等体协商的最后阶段。在这个阶段，NBRP 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，并且重新启动它的 Keepalive 定时器。每次本地系统发送 Keepalive 或者更新消息时，也重新启动它的 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。只要 NBRP 的状态有确定到空闲，它关闭 NBRP（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中被忽略。

§ 3.4 NBRP 路由处理

NBRP 是一个用于名字路由系统之间的路由协议。与任何的路由协议一样，NBRP 维护路由表，传播路由的刷新信息并且根据路由度量做出路由选择。NBRP 的主要功能是与其他的 NBRP 系统交换内容层可达性信息。内容层可达性信息 CLRI 中包含了可达性信息所经过的名字路由器列表，从而构造了一个名字路由器连接图，以避免路由环路，同时也使得基于名字路由器级别的策略控制成为可能。从这一点上讲，NBRP 是一个综合了向量一距离算法和链路一状态算法的协议。

本节着重讨论 NBRP 路由处理过程。我们首先对 NBRP 路由的相关理论作了阐述。Update 报文的处理进程是整个 NBRP 路由处理的核心，在本文的第二小节中，我们结合 NBRP 的协议细节和路由理论，对它进行了详细的讨论。最后，我们给出了 NBRP 的性能仿真结果，并进行了简单的分析。

3.4.1 NBRP 路由的相关理论

※ NBRP 的路由定义

我们可以将一个互联的自治系统抽象成 S＝(N, G)，其中 N 是所有内容网络系统的集合，G ＝ (V, E)是一个 NBRP 会话连接图。V 是系统中所有 NBRP 发言人的集合，E 是所有 NBRP 会话的集合。抽象的名字路由系统的网络拓扑图如图 3.8 所示。

图 3.8 中的 v 代表名字路由器，s 代笔内容服务器群。每一个名字路由系统由名字路由器和与它直接相连的内容服务器群组成。内容服务器群也可以通过内容交换机与名字路由器相连。名字路由器 v 是每一个路由系统的发言人和管理者。每一个路由系统都可以根据本地

发送 Notification 报文，改变自己的状态到空闲状态。

如果系统收到一个断开连接通知，本地系统改变自己的状态到空闲。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。

3.3.6 Established 状态（已建立）

这是对等体协商的最后阶段。在这个阶段，NBRP 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，并且重新启动它的 Keepalive 定时器。每次本地系统发送 Keepalive 或者更新消息时，也重新启动它的 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。只要 NBRP 的状态有确定到空闲，它关闭 NBRP（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中被忽略。

§ 3.4 NBRP 路由处理

NBRP 是一个用于名字路由系统之间的路由协议。与任何的路由协议一样，NBRP 维护路由表，传播路由的刷新信息并且根据路由度量做出路由选择。NBRP 的主要功能是与其他的 NBRP 系统交换内容层可达性信息。内容层可达性信息 CLRI 中包含了可达性信息所经过的名字路由器列表，从而构造了一个名字路由器连接图，以避免路由环路，同时也使得基于名字路由器级别的策略控制成为可能。从这一点上讲，NBRP 是一个综合了向量一距离算法和链路一状态算法的协议。

本节着重讨论 NBRP 路由处理过程。我们首先对 NBRP 路由的相关理论作了阐述。Update 报文的处理进程是整个 NBRP 路由处理的核心，在本文的第二小节中，我们结合 NBRP 的协议细节和路由理论，对它进行了详细的讨论。最后，我们给出了 NBRP 的性能仿真结果，并进行了简单的分析。

3.4.1 NBRP 路由的相关理论

※ NBRP 的路由定义

我们可以将一个互联的自治系统抽象成 S＝(N, G)，其中 N 是所有内容网络系统的集合，G ＝ (V, E)是一个 NBRP 会话连接图。V 是系统中所有 NBRP 发言人的集合，E 是所有 NBRP 会话的集合。抽象的名字路由系统的网络拓扑图如图 3.8 所示。

图 3.8 中的 v 代表名字路由器，s 代笔内容服务器群。每一个名字路由系统由名字路由器和与它直接相连的内容服务器群组成。内容服务器群也可以通过内容交换机与名字路由器相连。名字路由器 v 是每一个路由系统的发言人和管理者。每一个路由系统都可以根据本地

的系统运营状况，制定本地的路由策略。

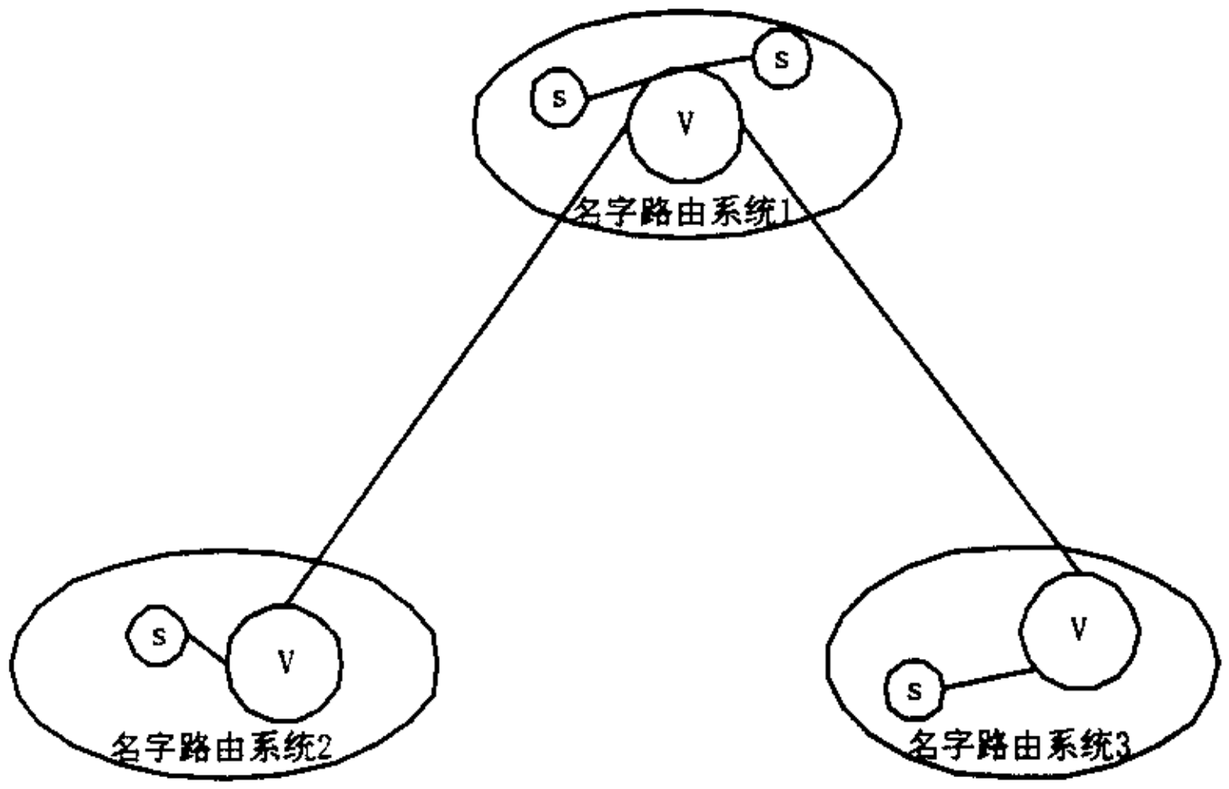


图 3.8 抽象的名字路由系统拓扑图

V 中的任意一点 v，通过 NBRP 的路由更新报文与它所有的 NBRP 发言人进行路由信息的交换。NBRP 的路由更新通告 r，包括下列重要属性：

（1） CLNI：内容网络可达性信息. 记为 r.clni；

（2） NEXT-HOP：下一跳的地址、记为 r. next_hop；

（3） NR_PATH：路由所经过的名字路由器，记为 r.nr_path；

（4） LOCAL_PREF：本地优先级，记为 r.loc_pref；

（5） ORIGIN:：改属性用来表示获得路由的方法，记为 r.origin。

※ NBRP 的路由策略

所谓的 NBRP 路由策略是由一个策略规则列表组成，其形式如下：

$$H_1 \Rightarrow A_1$$

$$H_2 \Rightarrow A_2$$

$$\vdots$$

$$H_n \Rightarrow A_n$$

每条规则的头部 H_i 是一个基于路由属性的布尔表达式，操作部分人可以是拒绝(Reject)、允许(Allow)或者是对路由某些属性的赋值。

NBRP 发言人所实施的路由策略又可以分为隐示路由策略和显示路由策略。所谓的隐示的路由策略是指 NBRP 自身必须进行的一些控制和处理。隐示的输入策略之一是避免路由环路的产生。设 $\partial\{l,v\}[R]$ 表示点 v 在边 l 上对接收到的路由集 R 实施的隐式输入路由策略，那么避免路由环路的输入策略可以表示为：如果 $\alpha(v) \in r.nr\_path$，那么 $\partial\{l,v\}[\{r\}]=\phi$，否则 $\partial\{l,v\}[\{r\}]=\{r\}$。

显式路由策略是指根据内容网络路由系统管理上的要求，人为配置的路由策略。典型的显式输入路由策略包括对输入的路由进行过滤，以及为待定路由设置 r. loc_pref 的值。设 $\beta\{l,v\}[R]$ 表示点 v 对于来自边 l 的路由集 R 实施的显式输入路由策略。最终，点 v 对于边 l 上接收到的路由集 R 实施的输入路由策略可表示为：$\lambda\{l,v\}[R]=\beta\{l,v\}[R']$，其中 $R'=(\partial\{l,v\}[R])$。

设 $\varphi\{l,v\}[R]$ 表示点 v 在边 l 对于即将发送的路由集 R 实施的隐式输出路由策略。隐式的输出路由策略包括将自身所在的名字路由器的 IP 地址，即 $\alpha(v)$ 插入到 r. nr-path 向量中，以及将 v 连接 l 的接口地址 a 赋值给 r. next_hop，r. next_hop= a 。设 $\xi\{l,v\}[R]$ 表示点 v 在边 l 对于即将发送的路由集 R 实施的显式输出路由策略。典型的显式输出路由策略包括对即将输出的路由进行过滤等工作。最终，点 v 在边 l 对于即将发送的路由集 R 先实施隐示输出路由策略，再实施显示输出路由策略。输出路由策略可表示为：$\eta(l,v)[R]=\xi\{l,v\}[R']$。其中 $R'=\varphi\{l,v\}[R]$。

※ 选择过程

NBRP 的路由选择过程是由 NBRP 更新通告触发的，以分布和异步的方式进行的。当 NBRP 发言人 v 在边 l 得到 u 发送的一个 BGP 更新通告 r_n 之后，v 将对 r_n. clni 中所包含的目的内容网络 $d(a_L/L)$ 实施 NBRP 的路由选择过程。a_L 表示内容网络的内容前缀。该选择过程是对所有能够到达目的内容网络 $d(a_L/L)$ 的路由集合 $R=\{r_1,r_2,r_3,\cdots,r_m\}$，按照选择函数 Select（R）进行最佳路由的选择。

选择过程可以描述为：

（1） 选取 r_i. loc_pref = max（r_1. loc_pref，r_2. loc_pref，…，r_n. loc_pref）；

（2） 否则，选取 length（r_i. nr_path）= min(length（r_1. nr_path），length（r_1. nr_path)，…，length（r_n. nr_path））；

（3） 否则，选取 r_i. next_hop = min(r_1. next_hop，r_2. next_hop，…，r_n. next_hop)；

从选择过程可以看到，规则 1 是基于路由策略的度量，规则 2 是基于最短路径的度量。规则 1 优于规则 2，打破了纯距离矢量路由协议中，使用单一的以最短距离为唯一度量的方法。我们在下一小节中还要对此进行讨论。

3.4.2 Update 报文的处理进程

NBRP 路由策略的核心是 Update 报文的处理进程。图 3.9 给出了报文处理进程的框图。

显式路由策略是指根据内容网络路由系统管理上的要求，人为配置的路由策略。典型的显式输入路由策略包括对输入的路由进行过滤，以及为待定路由设置 r. loc_pref 的值。设 $\beta\{l,v\}[R]$ 表示点 v 对于来自边 l 的路由集 R 实施的显式输入路由策略。最终，点 v 对于边 l 上接收到的路由集 R 实施的输入路由策略可表示为：$\lambda\{l,v\}[R]=\beta\{l,v\}[R']$，其中 $R'=(\partial\{l,v\}[R])$。

设 $\varphi\{l,v\}[R]$ 表示点 v 在边 l 对于即将发送的路由集 R 实施的隐式输出路由策略。隐式的输出路由策略包括将自身所在的名字路由器的 IP 地址，即 $\alpha(v)$ 插入到 r. nr-path 向量中，以及将 v 连接 l 的接口地址 a 赋值给 r. next_hop，r. next_hop= a 。设 $\xi\{l,v\}[R]$ 表示点 v 在边 l 对于即将发送的路由集 R 实施的显式输出路由策略。典型的显式输出路由策略包括对即将输出的路由进行过滤等工作。最终，点 v 在边 l 对于即将发送的路由集 R 先实施隐示输出路由策略，再实施显示输出路由策略。输出路由策略可表示为：$\eta(l,v)[R]=\xi\{l,v\}[R']$。其中 $R'=\varphi\{l,v\}[R]$。

※ 选择过程

NBRP 的路由选择过程是由 NBRP 更新通告触发的，以分布和异步的方式进行的。当 NBRP 发言人 v 在边 l 得到 u 发送的一个 BGP 更新通告 r_n 之后，v 将对 r_n. clni 中所包含的目的内容网络 $d(a_L/L)$ 实施 NBRP 的路由选择过程。a_L 表示内容网络的内容前缀。该选择过程是对所有能够到达目的内容网络 $d(a_L/L)$ 的路由集合 $R=\{r_1,r_2,r_3,\cdots,r_m\}$，按照选择函数 Select（R）进行最佳路由的选择。

选择过程可以描述为：

（1） 选取 r_i. loc_pref = max（r_1. loc_pref，r_2. loc_pref，…，r_n. loc_pref）；

（2） 否则，选取 length（r_i. nr_path）= min(length（r_1. nr_path），length（r_1. nr_path)，…，length（r_n. nr_path））；

（3） 否则，选取 r_i. next_hop = min(r_1. next_hop，r_2. next_hop，…，r_n. next_hop)；

从选择过程可以看到，规则 1 是基于路由策略的度量，规则 2 是基于最短路径的度量。规则 1 优于规则 2，打破了纯距离矢量路由协议中，使用单一的以最短距离为唯一度量的方法。我们在下一小节中还要对此进行讨论。

3.4.2 Update 报文的处理进程

NBRP 路由策略的核心是 Update 报文的处理进程。图 3.9 给出了报文处理进程的框图。

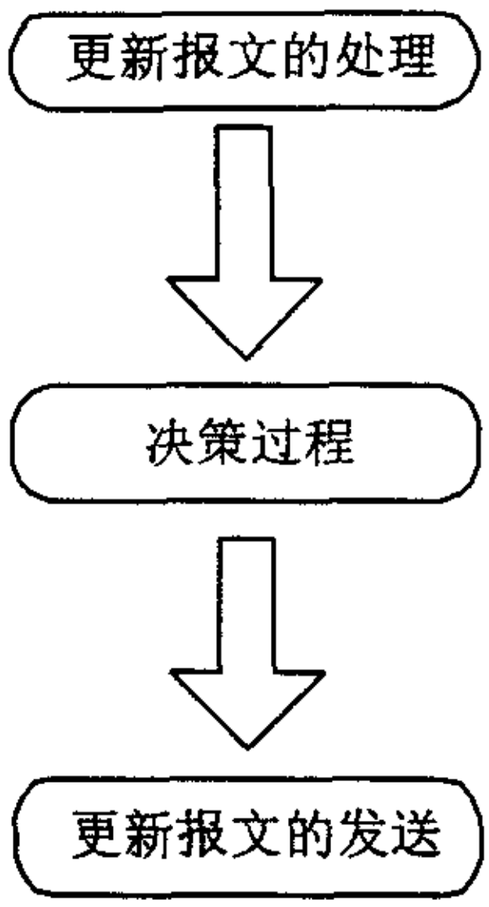


图 3.9　Update 报文的处理框图

从图 3.9 可以看出，它分成三个部分：更新报文的处理、决策过程、更新报文的发送。本节将围绕这三个部分进行详细的讨论。

在进行讨论之前，首先来介绍路由信息库的概念。

NBRP 的所有路由都储存在路由信息库里面。该路由信息库由三个部分组成:

1. Adj-RIB-IN: 该信息库记录的是本地 NBRP 从它的对等体获取的路由更新信息，该信息可以作为决策过程的输入。

2. LOCAL-RIB: 该信息库记录的是本地决策过程之后的路由信息，该信息库也是 INRP 所用的路由表。

3. Adj-RIB-OUT：记录的是本地 UPDATE 报文通告给特定的对等体的路由信息。

我们可以这么认为，Adj-RIB-IN 包含了从对等体节点广播到本地的名字路由选择信息；LOCAL-RIB 包含了本地决策过程后的信息；Adj-RIB-OUT 依靠本地 NBRP 组织更新路由信息，广播到特定的对等节点。

（一）更新报文的处理

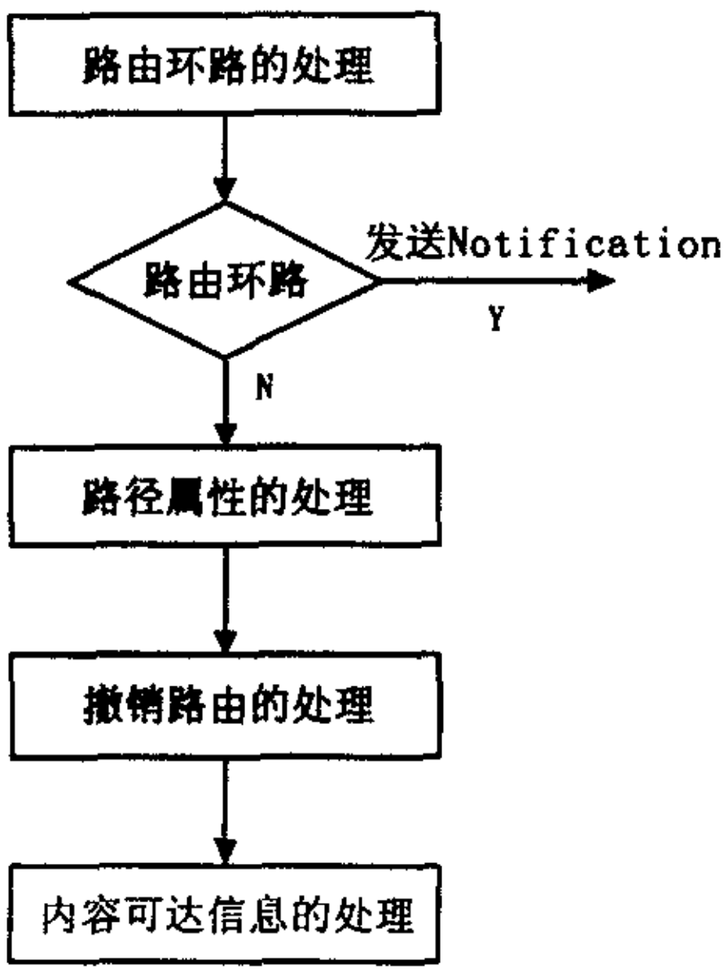


图 3.10 更新报文的处理

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，NBRP 首先检查其每一个字段的有效性。如果报文格式不正确，NBRP 将抛弃该条报文，并向发送报文的对等体发出 Notificaton 报文。

报文的处理如图 3.10 所示。它可以分成四个部分：路由环路的处理、路径属性的处理、撤销路由的处理、内容可达信息的处理。

◆ 路由环路的处理

路由环路是路由系统竭力避免的问题。路由环路将不可避免的造成路由风暴，造成网络拥塞。NBRP 基于距离矢量算法，但它参考了链路状态路由算法的优点。内容层可达性信息 CLRI 中包含了可达性信息所经过的名字路由器列表，从而构造了一个名字路由器连接图，避免了路由环路。

因此，报文处理的第一步就是判断是否形成路由环路。路由器主机遍历 CLRI 中的名字路由器列表，查看是否有本机的 IP 地址。如果路由环路发生，主机将抛弃该报文，并发出 NBR 路由环路 Notification 报文。

◆ 对路径属性的处理

如果可选的无传输属性（Optional Non_transitive）是不可识别的，而且传递标记未置位时，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传输标志位（第三个高位序比特）设置为 1，则该属性在报文处理过程中保持不变，为其对等体保持。

如果可选（Optional）属性是可识别的，且有一个有效值，该属性将被处理。依赖于可选属性类型，该属性可能被局部处理、保持或者更新。

◆ 撤销路由的处理（Withdrawn routes）

如果更新报文的撤销路由字段非空，我们需要对该撤销的名字路由进行处理。

一条撤销的名字路由信息特定的内容域名。它表示提供某种内容服务的内容服务器将不可用。内容域名标志该内容服务。对撤销路由的处理包括两个部分：

a. 根据撤销路由的内容域名进行信息库 Adj-RIP-In 的全匹配查找，对应的记录将从 Adj-RIB-In 移走。

b. 本地 NBRP 发言人将运行它的路由决策程序，通告它的对等体，通知该条名字路由信息失效。

◆ 内容层可达信息（CLNI）的处理

如果更新消息包含了 CLNI 可行名字路由信息，我们需要对 CLNI 信息进行一些处理。

首先，我们根据 CLNI 进行 Adj-RIB-In 信息库的最长后缀匹配查找。接下来，NBRP 采取的行为取决于查找的结果。

情况 1： 内容层可到达信息（CLRI）与当前储存在 Adj-RIB-In 中的一条路由相同。那么 NBRP 在 Adj-RIB-In 中新路由将会取代旧路由，这样就隐含地从名字路由路由器上撤回了旧路由。当旧路由不再允许使用，NBRP 主机将运行决策进程。

情况 2： 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）。这包括两种结果：

a. 新路由比旧路由的信息更明确。例如，新路由的 CLNI 是 grs.zju.edu.cn，旧名字路由是 zju.edu.cn。

b. 旧路由的信息比新路由更明确。例如，旧路由的 CLNI 是 grs.zju.edu.cn，新名字路由是 zju.edu.cn。

在结果 a 的条件下，NBRP 运行者将运行名字路由的决策进程。因为结果 a 将导致部分名字路由信息不可用。

在结果 b 的条件下，NBRP 根据本地路由策略配置决定是否需要进行进一步的行为。

情况 3： 如果新路由含有的内容层可到达信息（CLRI），不在 Adj-RiB-In 中的任何路由中存在，那么新路由将放在 Adj-RIB-In 中，同时 NBRP 运行它的决策进程。

（二） 本地决策过程

本小节可以看成是对 3.3.1 小节选择过程的细化。负责对存储在本地 RIB 中的路由进行选择。选择依赖于本地路由选择策略。决策过程的输出将作为通告信息，广播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。

◆ 选择函数

广播路由以前，NBRP 根据本地策略信息对存储在 Adj-RIB-In 中的路由进行选择。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。这个整型表示路由的优先程度。下面给出了一个常见的选择函数：

```
dop = 100 - numhops;                    （式 3.1）
```

其中 dop 代表优先级，numhops 表示该 CLNI 内容可达性信息所经过的名字路由器的总数。

◆ 决策进程的三个阶段

决策进程对包含在 Adj-RIB-In 中各条路由进行操作，包括两个部分：

a. 名字路由选择。路由选择的输出（名字路由信息）将广播给所有的对等体。

b. 名字路由聚合和名字路由信息缩减。

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。图 3.11 给出了决策过程的阶段转换图。

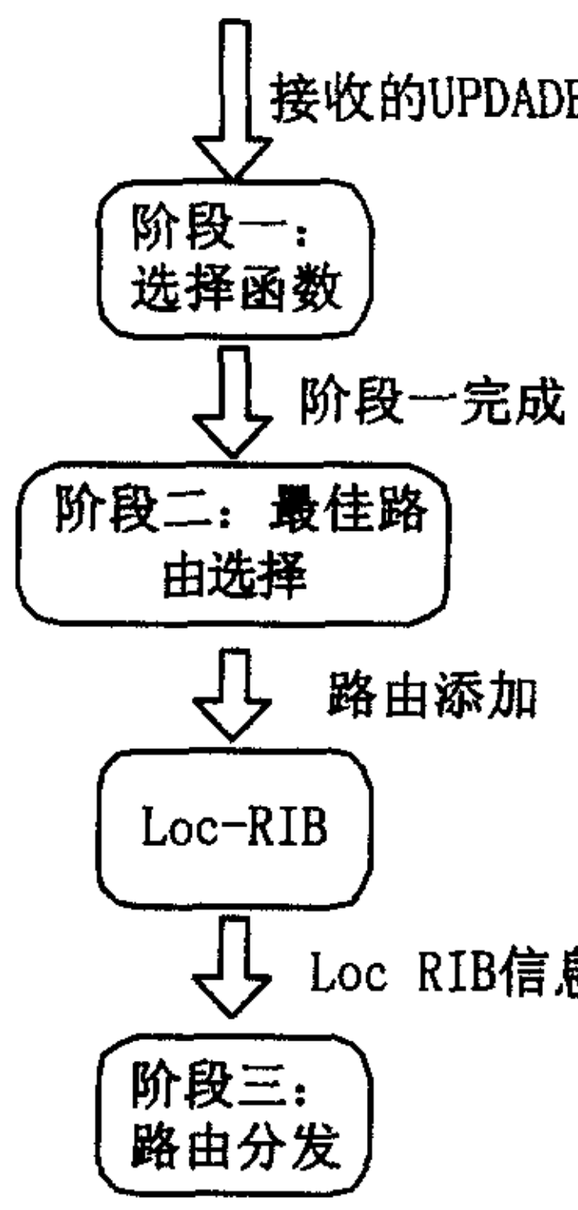


图 3.11 阶段转换图

阶段 1：负责计算每一个从名字路由对等体邻居接收到的名字路由的优先程度。路由优先程度的计算基于本地的选择函数。

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 L-RIB 被修改时激活阶段 3。根据本地策略，阶段 3 负责分发 Loc-RIB 中的各个名字路由到每一个名字路由对等体邻居。路由聚合和信息缩减可以在这个阶段选择是否进行。

下面将对三个阶段的处理进行详细讨论。

阶段 1：计算路由优先级

本地 NBRP 接收到从 NBRP 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。需要注意的是，在对 Adj-RIB-In 里面的任何路由进行操作之前，进程函数将锁上信息库 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表，本地 NBRP 运行者将决定该路由得优先级。如果路由的属性包含 LOCAL_PREF，LOCAL_PREF 属性值将被作为优先程度。否则，将在预配置策略信息基础上计算优先级，并且向本地对等体广播这些路由时，将计算出来的优先级当作 LOCAL_PREF 属性值。

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 的功能模块是独立的进程，当没有路由进行选择时自动结束。阶段 2 的进程将考虑出现在 Adj-RIB-In 中的所有路由。一旦阶段 3 决策进程被调用，阶段 2 决策进程的运行状态被阻塞。在对信息库 Adj-RIB-In 的任何路由进行操作之前，进程函数将锁上 Adj-RIB-In。一旦操作进行完毕，解锁 Adj-RIB-In。

NBRP 的决策过程基于 CLNI 的各种属性值。当面对同一内容目的（内容域名匹配）多个路由时，为了保证网络质量，将选择最佳的内容路由。

最佳路由选择的依据：

1. 选择本地优先级最大的路由
2. 如果本地优先级相同，选择本地始发的路由
3. 如果本地优先级相同，选择 CLNI 内容可达性信息所经过的名字路由器的总数（NR）最小的路由

阶段 3：路由分散

阶段 3 的决策过程也是一个独立的进程，当它没有进一步的工作需要完成的时候就自动结束。

阶段 3 的触发条件是 LOC-RIB 被修改。这个阶段的任务之一是将 LOC-RIB 当中的变化反映到信息库 Adj-RIB-OUT 中。LOC-RIB 中发生变化的路由将被安装到信息库 Adj-RIB-OUT 中。发送进程将根据 Adj-RIB-OUT 中的路由，向对等体广播更新报文。

阶段 3 的另一个任务是进行名字路由聚合和信息缩减。例如内容提供者 zju.edu.cn 包含了两个内容服务器，它们对应不同的内容服务。一台内容服务器提供研究生信息服务，它的标志域名 Grs.zju.edu.cn。而另一台内容服务器提供就业信息，它的标志域名是 Career.zju.edu.cn。如果这两台服务器独立的发布内容路由信息，在 LOC-RIB 中将同时存在两个路由表项。它们的域名分别是 grs.zju.edu.cn 和 career.zju.edu.cn。考虑内容路由的架构，如图 3.12 所示。

从图 3.12 看到，属于同一个内容域(zju.edu.cn)的内容服务器群（grs.zju.edu.cn 和 career.zju.edu.cn）通过本地交换机连接到本地内容路由器。对于内容路由器来说，内容路由 grs.zju.edu.cn 和 career.zju.edu.cn 的下一跳都是相同的。因此，我们将路由项

grs. zju. edu. cn 和 career. zju. edu. cn 合并为一项:zju. eu. cn。

这就是路由聚合的原理。路由聚合通过最长后缀名字匹配和下一跳的比较实现。如果几个路由表项

a. 通过最长后缀名字匹配，它们的下一级域名都相同。

b. 它们的下一跳都相同。

那么我们将把它们进行路由聚合。新的名字路由的域名将是它们的下一级域名，下一跳依然保持不变。

通过路由聚合，内容路由表的数目将大大缩小。这将缩小路由查询的时间。，这也将减小网络的拥塞，我们不需要给每一个内容服务器发布路由通告。我们只需要给每一个内容域发布名字路由通告。

需要注意的是，路由聚合是本地可选的策略。

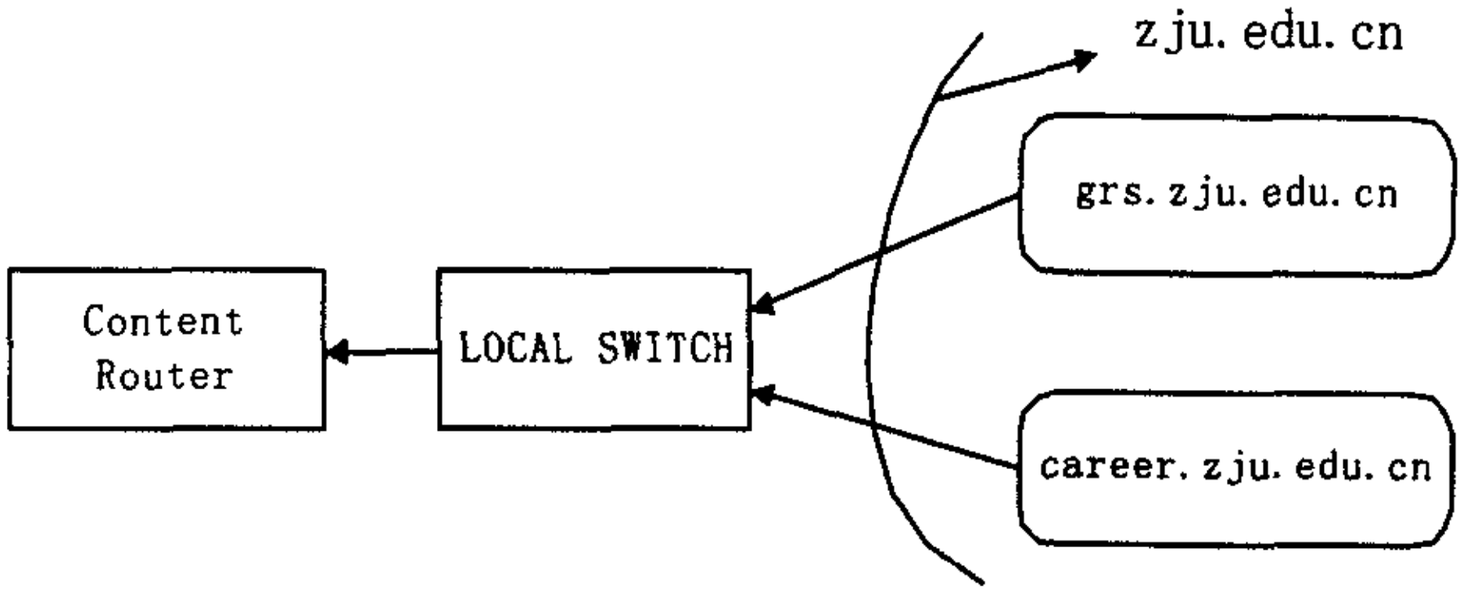


图 3.12　内容路由的建构图

（三） 更新报文的发送

Update 报文发送进程负责将 Update 报文发送到所有的对等体。

当阶段三的决策进程完成以后，更新报文的发送进程将被激活。该进程将把 Adj-RIB-OUT 中新安装的名字路由发送到所有的对等体。

发送进程需要对更新报文做一些修改。

1） 进程需要在 NR_PATH 中加入本地名字路由器的 IP 地址。

2） Next-hop 属性应该改为本地名字路由器的 IP 地址。

3) 如果该内容可达信息是本地路由聚合的结果，那么 Aggregator 将被设为本地的 IP 地址。

由于 NBRP 发现了路由更新，我们需要通知 INRP 协议更新本地名字路由表。

3.4.3 NBRP 的性能仿真（吞吐量）

图 3.13 给出了 NBRP 的性能仿真结果（吞吐量）。Update 报文的处理能力随着对等体数目的线性增加而逐步下降。在 1 个对等体的情况下，Update 报文的处理能力为 1100 个报文/s。随着对等体数目的增加，处理能力曲线下降，在 6 个对等体的情况下，处理能力降到了 380 个报文/s。

grs. zju. edu. cn 和 career. zju. edu. cn 合并为一项:zju. eu. cn。

这就是路由聚合的原理。路由聚合通过最长后缀名字匹配和下一跳的比较实现。如果几个路由表项

a. 通过最长后缀名字匹配，它们的下一级域名都相同。

b. 它们的下一跳都相同。

那么我们将把它们进行路由聚合。新的名字路由的域名将是它们的下一级域名，下一跳依然保持不变。

通过路由聚合，内容路由表的数目将大大缩小。这将缩小路由查询的时间。，这也将减小网络的拥塞，我们不需要给每一个内容服务器发布路由通告。我们只需要给每一个内容域发布名字路由通告。

需要注意的是，路由聚合是本地可选的策略。

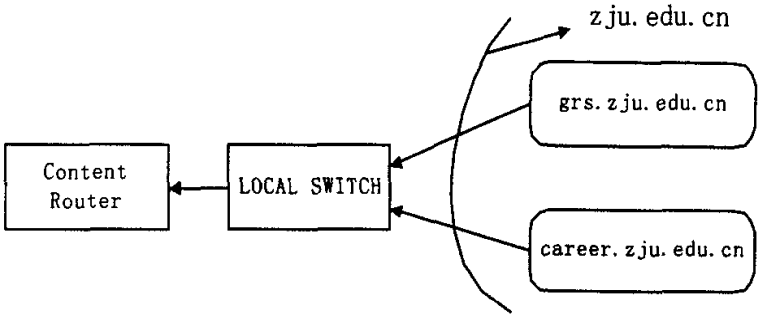


图 3.12 内容路由的建构图

（三） 更新报文的发送

Update 报文发送进程负责将 Update 报文发送到所有的对等体。

当阶段三的决策进程完成以后，更新报文的发送进程将被激活。该进程将把 Adj-RIB-OUT 中新安装的名字路由发送到所有的对等体。

发送进程需要对更新报文做一些修改。

1） 进程需要在 NR_PATH 中加入本地名字路由器的 IP 地址。

2） Next-hop 属性应该改为本地名字路由器的 IP 地址。

3) 如果该内容可达信息是本地路由聚合的结果，那么 Aggregator 将被设为本地的 IP 地址。

由于 NBRP 发现了路由更新，我们需要通知 INRP 协议更新本地名字路由表。

3.4.3 NBRP 的性能仿真（吞吐量）

图 3.13 给出了 NBRP 的性能仿真结果（吞吐量）。Update 报文的处理能力随着对等体数目的线性增加而逐步下降。在 1 个对等体的情况下，Update 报文的处理能力为 1100 个报文/s。随着对等体数目的增加，处理能力曲线下降，在 6 个对等体的情况下，处理能力降到了 380 个报文/s。

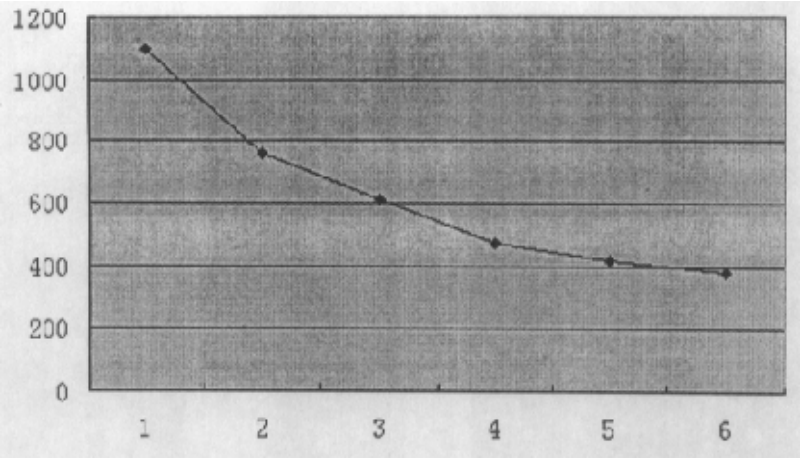


图 3.13 NBRP 吞吐量性能图

*：X 轴，NBRP 对等体个数；Y 轴，每秒处理的 update 报文数目

§ 3.5 NBRP 路由稳定性能的研究

近年来 Internet 爆炸式的发展和增长带来了很多复杂、严重而亟待解决的问题，诸如网络风暴、路出震荡等网络故障。这些问题将不可避免的出现在 CDN 网络中。我们试图从一切可能的途径寻找并实现对现行的以及将来的名字路由网络所出现的问题的解决方案。由于内容传送网络本身的庞大性和各个内容提供商业机构、组织对其所辖范围的自律性，使得要在实际的内容传送网络上进行名字路由稳定问题的分析和研究几乎是不可能的。而系统仿真是以系统理论、形式化理沦、随机过程与统计学理论为基础，以计算机和仿真系统软件为工具，对现实系统或未来系统进行动态的实验研究的理论和方法。将计算机仿真技术、实际网络上的可控信息的收集和统计学方法相结合的方式，是对 CDN 内容传送网络的稳定性和故障的研究的一个科学而且可行的方式。对仿真模型进行多次独立重复的运行后，将得到一系列仿真输出数据，根据仿真输出数据进行统计分析和推断，找到影响名字路由系统之间影响 NBRP 路由稳定因素。

3.5.1 关于 NBRP 的稳定性分析

※ NBRP 对 CDN 网络的影响

在初始的 NBRP 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。主要 NBRP 的核心路由器一般都有到达每个可达内容服务网络的一条路由，这样，每一个 NBRP 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内容前缀的最佳名字路由。随着内容传送网络的不断快速增长，这个数目也在不断增长。统计表明，现行的 IP 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。

NBRP 消耗的带宽和 CPU 周期不依赖于 UPDATE 条目数，而是 CDN 网络的稳定性。如果 CDN 网络是稳定的，则 NBRP 消耗的带宽和 CPU 周期，只是由于交换 NBRP 的 KeepAlive 报文引起的（KeepAlive 报文只在 NBRP 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 5bt 以 M，基本不消耗任何处理能力，这种开销可以忽略。如果 CDN 网络不稳定，则只有内容可达性

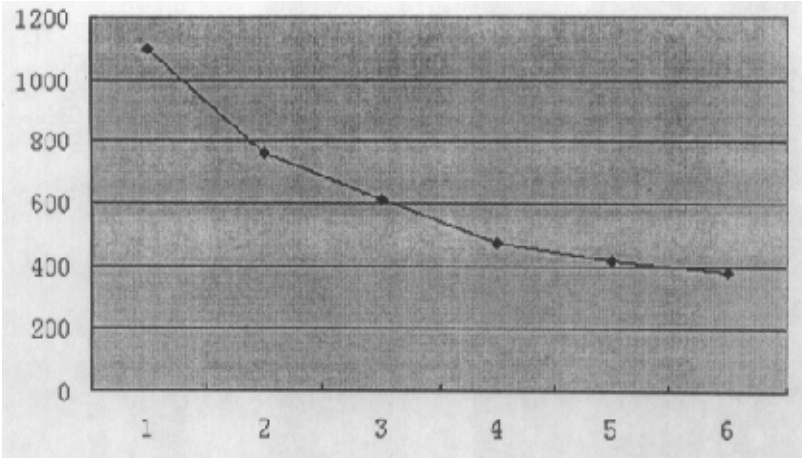


图 3.13 NBRP 吞吐量性能图

*：X 轴，NBRP 对等体个数；Y 轴，每秒处理的 update 报文数目

§ 3.5 NBRP 路由稳定性能的研究

近年来 Internet 轰炸式的发展和增长带来了很多复杂、严重而亟待解决的问题，诸如网络风暴、路出震荡等网络故障。这些问题将不可避免的出现在 CDN 网络中。我们试图从一切可能的途径寻找并实现对现行的以及将来的名字路由网络所出现的问题的解决方案。由于内容传送网络本身的庞大性和各个内容提供商业机构、组织对其所辖范围的自律性，使得要在实际的内容传送网络上进行名字路由稳定问题的分析和研究几乎是不可能的。而系统仿真是以系统理论、形式化理沦、随机过程与统计学理论为基础，以计算机和仿真系统软件为工具，对现实系统或未来系统进行动态的实验研究的理论和方法。将计算机仿真技术、实际网络上的可控信息的收集和统计学方法相结合的方式，是对 CDN 内容传送网络的稳定性和故障的研究的一个科学而且可行的方式。对仿真模型进行多次独立重复的运行后，将得到一系列仿真输出数据，根据仿真输出数据进行统计分析和推断，找到影响名字路由系统之间影响 NBRP 路由稳定因素。

3.5.1 关于 NBRP 的稳定性分析

※ NBRP 对 CDN 网络的影响

在初始的 NBRP 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。主要 NBRP 的核心路由器一般都有到达每个可达内容服务网络的一条路由，这样，每一个 NBRP 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内容前缀的最佳名字路由。随着内容传送网络的不断快速增长，这个数目也在不断增长。统计表明，现行的 IP 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。

NBRP 消耗的带宽和 CPU 周期不依赖于 UPDATE 条目数，而是 CDN 网络的稳定性。如果 CDN 网络是稳定的，则 NBRP 消耗的带宽和 CPU 周期，只是由于交换 NBRP 的 KeepAlive 报文引起的（KeepAlive 报文只在 NBRP 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 5bt 以 M，基本不消耗任何处理能力，这种开销可以忽略。如果 CDN 网络不稳定，则只有内容可达性

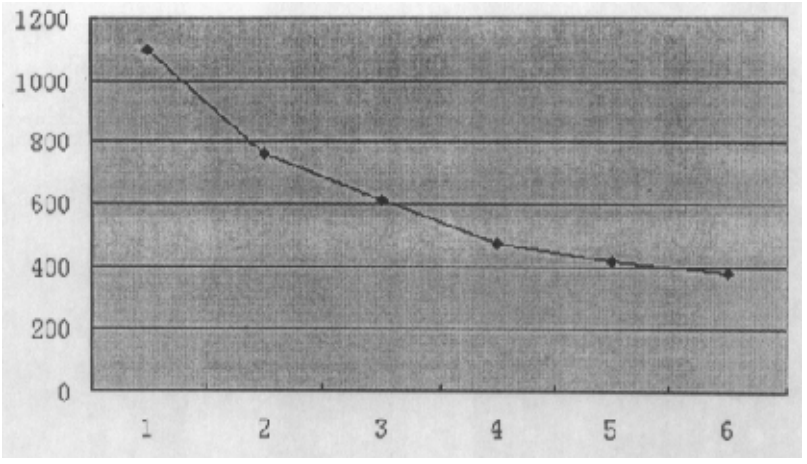


图 3.13 NBRP 吞吐量性能图

*：X 轴，NBRP 对等体个数；Y 轴，每秒处理的 update 报文数目

§ 3.5 NBRP 路由稳定性能的研究

近年来 Internet 轰炸式的发展和增长带来了很多复杂、严重而亟待解决的问题，诸如网络风暴、路出震荡等网络故障。这些问题将不可避免的出现在 CDN 网络中。我们试图从一切可能的途径寻找并实现对现行的以及将来的名字路由网络所出现的问题的解决方案。由于内容传送网络本身的庞大性和各个内容提供商业机构、组织对其所辖范围的自律性，使得要在实际的内容传送网络上进行名字路由稳定问题的分析和研究几乎是不可能的。而系统仿真是以系统理论、形式化理沦、随机过程与统计学理论为基础，以计算机和仿真系统软件为工具，对现实系统或未来系统进行动态的实验研究的理论和方法。将计算机仿真技术、实际网络上的可控信息的收集和统计学方法相结合的方式，是对 CDN 内容传送网络的稳定性和故障的研究的一个科学而且可行的方式。对仿真模型进行多次独立重复的运行后，将得到一系列仿真输出数据，根据仿真输出数据进行统计分析和推断，找到影响名字路由系统之间影响 NBRP 路由稳定因素。

3.5.1 关于 NBRP 的稳定性分析

※ NBRP 对 CDN 网络的影响

在初始的 NBRP 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。主要 NBRP 的核心路由器一般都有到达每个可达内容服务网络的一条路由，这样，每一个 NBRP 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内容前缀的最佳名字路由。随着内容传送网络的不断快速增长，这个数目也在不断增长。统计表明，现行的 IP 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。

NBRP 消耗的带宽和 CPU 周期不依赖于 UPDATE 条目数，而是 CDN 网络的稳定性。如果 CDN 网络是稳定的，则 NBRP 消耗的带宽和 CPU 周期，只是由于交换 NBRP 的 KeepAlive 报文引起的（KeepAlive 报文只在 NBRP 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 5bt 以 M，基本不消耗任何处理能力，这种开销可以忽略。如果 CDN 网络不稳定，则只有内容可达性

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内容网络的更新信息时，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 NR-PATH 属性。也就是说，改变的路由一般都有共同的 NR-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，NBRP 协议消耗的带宽和 CPU 周期仅仅依赖于 CDN 网络的稳定性，但丝毫不依赖于组成 CDN 的各个内容网络的数目。所以假设各个内容网络之间连接的总体稳定性可以被控制的话，NBRP 协议在带宽和路由器则周期上的开销就具有了非常好的可伸缩性。CDN 网络的快速增长使网络的稳定性成为一个非常重要的问题，但 NBRP 本身并没有为 CDN 带来任何不稳定因素。给网络带来不稳定因素的往往是本地的路由策略。

※ 路由策略引起的路由振荡

NBRP 是一种路径矢量协议，支持基于路由策略的路由，因而使得每个内容路由系统具有很高的独立性和自主性，同时也使得各个路由系统之间缺乏详细的了解。因此，在制定本地路由策略的时候，从局部的角度来看所制定的策略是十分合理的，但是，当这些独立制定的策略在协同工作的时候，则很有可能存在潜在的冲突，从而可能导致局部或全局的路由出现异常。路由策略冲突所表现出的主要症状之一，是持续性或持久性的路由振荡，即在多个名字路由系统之间不断反复交换相同的路由信息。持续性的振荡在某些条件下会收敛稳定，但是收敛的速度会在较大的范围内变化。

路由振荡的原因之一是路由策略冲突。下面给出相应定义。

定义1　若对一个目的地址d和相应的激发序列 $A(n)$。如果与 $A(n)$对应的节点序列 $v_n=(n=0,1,2,3,\cdots)$中的部分节点多次重复更新，即存在节点序列…，v_i，v_{i+1}，…，v_j，v_{j+1}，…，v_m，…，其中$v_i=v_j, v_{i+1}=v_{j+1}, \cdots, v_{j-1}=v_m$，并且$c_i=c_j, c_{i+1}=c_{j+1}, \cdots, c_{j-1}=c_m$，则称NBRP系统振荡，简称路由振荡。称 v_n 为振荡路径，称 v_i，v_{i+1} 为振荡环。如果振荡环无限循环出现，即 $\lim Count(v_n)=\infty$（$Count(v_n)$表示振荡环出现次数），则称NBRP路由系统持续振荡，简称持续振荡。

由定义，显然有下面结论。

定理1　对任意的NBRP系统，系统收敛的充分必要条件是不存在路由持续振荡。

证明：　1）　充分性。用反证法证明。设存在一个节点序列 $v_n(n=0,1,2,3\cdots)$，v_n 中的振荡环…，v_i，v_{i+1}，…，v_j，v_{j+1}，…，v_m，…(其中$v_i=v_j, v_{i+1}=v_{j+1}, \cdots, v_{j-1}=v_m$)无限循环出现。设振荡环$v_i$，$v_{i+1}$，…，$v_j$对应的赋值状态$s_i$，$s_{i+1}$，…，$s_j$（$s_i=<c_0^i, c_1^i, \cdots, c_h^i, c_{h+1}^i, \cdots, c_n^i>$）。由激发序列定义知，只有新的路由信息产生时，序列中的节点激发，所以一定存在一个j（h<j<k），使得$c_j^i \neq c_j^{i+1}$，进而有$s_i \neq s_{i+1}$，又由振荡环是无限循环出现的，所以不存在终止状态s，与NBRP系统收敛矛盾，充分性得证。

2） 必要性。由NBRP系统收敛定义，显然成立。

从网络性能角度，无论是振荡还是持续振荡都将直接影响网络性能。而振荡的原因之一是路由策略冲突。

3.5.2 仿真与分析

NBRP 是一个路由策略的路径向量路由协议，所有的运行 NBRP 的名字路由器都通过本地的策略选择性地接收和通告路由信息，而向外界屏蔽其自身所管理的内容服务器的网络拓扑结构和策略方案。前面的分析表明，正是由于这种屏蔽和自主策略的存在，一些原本具有收敛性的网络拓扑在一定的路由策略影响下，将引发一段时间甚至足永久性的路由振荡。路由振荡将给整个网络带来非常严重的损害。下面对 3 节点网络进行仿真实验研究表明，非永久性震荡的路由收敛时间与实际链路的延迟抖动有关，又与运行 NBRP 的路由器的本地策略配置有关。适当调整 NBRP 会话者对路由的通告时间，改变链路延迟将缩短收敛时间，并减少发包总数，增强路由稳定性。

（一） 问题抽样模型

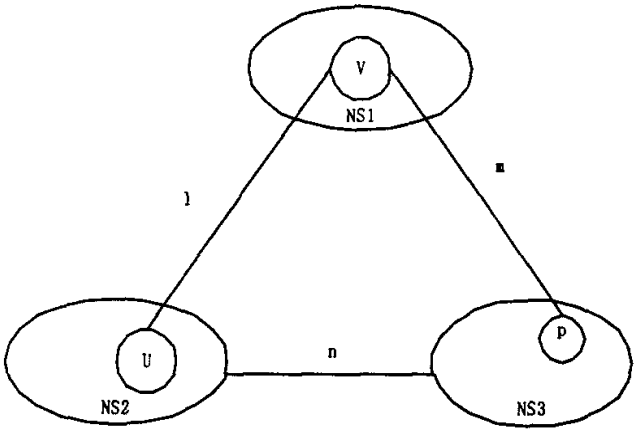


图 3.14 网络扑图

我们将具有三角形关系的名字路由系统之间的 NBRP 策略实施关系抽象成上面的 3 节点网络拓扑(图 3.14)。当一个名字路由系统 NS1 的 NBRP 会话者 v 向 NS2 的 NBRP 会话者 u 和 NS3 的 NBRP 会话者通告一条路由 d 时，将在该拓扑结构中引起路由振荡并波及到与名字路由器 u 和 p 相连接的其它网络。本文将进一步研究当不同名字路由系统中的 NBRP 会话者采取不同的时钟保持时间来处理路由时间 d，从而改变各个名字路由系统的 NBRP 会话者对路由 d 的处理时延，对路由稳定性和收敛过程的影响。还将研究在相同的策略下，不同的链路延迟对收敛过程的影响。

在本问题模型中，共有 3 个名字路由系统，其 NBRP 发言人分别为 u，v 和 p。每个 NBRP 发言人与其他两个发言人建立名字路由的 NBRP 会话：l，M 和 n。设建立 NBRP 会话 l，m 和 n 的内容网络前缀分别为：a_{Lm}/L_m，a_{Ln}/L_n，a_{L_l}/L_l。为每个 NBRP 的路由策略配置为：

发言人 u：1）$\beta(\mathrm{l},\mathrm{u})$配置为：接受来自 v 的所有路由更新且 r.loc_pref=0；2）$\beta(n,\ u)$

最小通告时间在一定范围内设置得尽量小并且二者之间保持适当的差距，将减少路由振荡的时间和同一路由更新信息的重复处理和传播，从而增强 NBRP 路由的稳定性。因此，可以通过对 NBRP 策略的改动来增强路由的稳定性。

$ 3.6 NBRP 的安全性能分析

NBRP 运行于可靠的传输协议之上，采用传输控制协议 TCP 作为其底层协议，这样便无须显式地进行报文的分片、重传、确认和排序。

然而，即使 TCP 是一个端到端的可靠传赖协议，它的安全性也不能得到很好的保证，任何恶意的攻击都可利用 NBRP 本身存在许多安全漏洞达到非法目的。例如，一个 NBRP 路由器上与 NBRP 相关的应用、配置信息或者路由数据库都有可能通过对路由器没有授权的访问、一个欺骗的分布通道而被修改或删除；已经授权的 NBRP 报文可能被捕获，经过一定的重组或压缩，再重新注入该链接；一个虚假的或者可能被攻击者利用的 NBRP 路由器会产生大量的实际上并不经过它的路由更新报文，它在这些报文加进一些非法代码然后再广播给其所有邻居。

NBRP 路由协议包括的安全信息很少，它无法对它传递的名字路由信息提供保护。在这种情况下，运行 NBRP 协议的名字路由器不得不相信它所接收到的所有的 UDDATE 报文，这相当于它必须信任内容网络上的所有名字路由器。这种情况显然是不能接受的。

虽然有一些安全措施，比如路由器级的链路加密、端到端的加密信息封装，有可能会防止一些上面提到的攻击，但这些措施在具体实施起来都比较团难，不太实际。必须在 NBRP 协议本身实施一些安全措施才能保证服报文传输的安全性。

本节的就 NBRP 协议的安全问题进行了探索研究。根据 NBRP 协议的安全目标，提出了相应的安全对策和具体措施。本节对 NBRP 的安全框架进行了研究。

3.6.1 NBRP 要实现的安全目标

目标是为 NBRP 提供认证机制、完整性控制、信息保密和访问控制。 具体来说，就是为了：

（1） 防止各类攻击者对路由信息进行伪装、修改和重传；

（2） 防止破坏性的链路和破坏性的发言者对 NBRP 路由信息的破坏；

（3） 不把任何路由信息泄漏给各类攻击者；

本文提出的对策是从NBRP报文设计角度出发，配以数字签名技术的使用。设法保证NBRP报文传送的完整性、机密性和确实性以及提供准入控制手段等，目的在于在解决NBRP系统安全上的缺陷以及旧报文重发的问题。同时避免了递归算法，提高了空间和时间效率。

3.6.2 NBRP 的安全扩展概述

增强NBRP系统安全性的基本思想是对NBRP报文增添鉴定信息并采取加密对策。通常在NBRP系统中有两种类型的通信：

1）相邻会话者之间的通信；

2）由路由选择动态决定的远程会话者之间的通信（是相邻会话者通信的逐步延伸）

相邻体会话者之间的通信由UPDATE报文组成，包括:发送方认为应该发送给接受方的、关于目的地址的路由更新信息。远程会话者之间的通信由UPDATE报文中描述整个路由的属性域组成,是NBRP报文的转发。

根据这两类不同的信息，我们提出了下面两类安全对策：

1. 通用 NBRP 报文保护：

(1) 增加报文序列号用于防止报文被攻击者重传或删除。

最小通告时间在一定范围内设置得尽量小并且二者之间保持适当的差距，将减少路由振荡的时间和同一路由更新信息的重复处理和传播，从而增强 NBRP 路由的稳定性。因此，可以通过对 NBRP 策略的改动来增强路由的稳定性。

$ 3.6 NBRP 的安全性能分析

NBRP 运行于可靠的传输协议之上，采用传输控制协议 TCP 作为其底层协议，这样便无须显式地进行报文的分片、重传、确认和排序。

然而，即使 TCP 是一个端到端的可靠传赖协议，它的安全性也不能得到很好的保证，任何恶意的攻击都可利用 NBRP 本身存在许多安全漏洞达到非法目的。例如，一个 NBRP 路由器上与 NBRP 相关的应用、配置信息或者路由数据库都有可能通过对路由器没有授权的访问、一个欺骗的分布通道而被修改或删除；已经授权的 NBRP 报文可能被捕获，经过一定的重组或压缩，再重新注入该链接；一个虚假的或者可能被攻击者利用的 NBRP 路由器会产生大量的实际上并不经过它的路由更新报文，它在这些报文加进一些非法代码然后再广播给其所有邻居。

NBRP 路由协议包括的安全信息很少，它无法对它传递的名字路由信息提供保护。在这种情况下，运行 NBRP 协议的名字路由器不得不相信它所接收到的所有的 UDDATE 报文，这相当于它必须信任内容网络上的所有名字路由器。这种情况显然是不能接受的。

虽然有一些安全措施，比如路由器级的链路加密、端到端的加密信息封装，有可能会防止一些上面提到的攻击，但这些措施在具体实施起来都比较团难，不太实际。必须在 NBRP 协议本身实施一些安全措施才能保证服报文传输的安全性。

本节的就 NBRP 协议的安全问题进行了探索研究。根据 NBRP 协议的安全目标，提出了相应的安全对策和具体措施。本节对 NBRP 的安全框架进行了研究。

3.6.1 NBRP 要实现的安全目标

目标是为 NBRP 提供认证机制、完整性控制、信息保密和访问控制。 具体来说，就是为了：

（1） 防止各类攻击者对路由信息进行伪装、修改和重传；

（2） 防止破坏性的链路和破坏性的发言者对 NBRP 路由信息的破坏；

（3） 不把任何路由信息泄漏给各类攻击者；

本文提出的对策是从NBRP报文设计角度出发，配以数字签名技术的使用。设法保证NBRP报文传送的完整性、机密性和确实性以及提供准入控制手段等，目的在于在解决NBRP系统安全上的缺陷以及旧报文重发的问题。同时避免了递归算法，提高了空间和时间效率。

3.6.2 NBRP 的安全扩展概述

增强NBRP系统安全性的基本思想是对NBRP报文增添鉴定信息并采取加密对策。通常在NBRP系统中有两种类型的通信：

1）相邻会话者之间的通信；

2）由路由选择动态决定的远程会话者之间的通信（是相邻会话者通信的逐步延伸）

相邻体会话者之间的通信由UPDATE报文组成，包括:发送方认为应该发送给接受方的、关于目的地址的路由更新信息。远程会话者之间的通信由UPDATE报文中描述整个路由的属性域组成,是NBRP报文的转发。

根据这两类不同的信息，我们提出了下面两类安全对策：

1. 通用 NBRP 报文保护：

(1) 增加报文序列号用于防止报文被攻击者重传或删除。

（2）在 NBRP 对等体建立连接时进行认证，通过认证后协商会话密钥。用此会话密钥加密所有的 NBRP 报文。这种方式可以保证路由信息的机密性；

2. NBRP 的 UPDATE 报文保护：

下面我们对这两类安全对策进行详细讨论。

3.6.3 通用 NBRP 报文保护

提供NBRP报文保护的目的是对相邻会话者之间通信的UPDATE报文提供机密性、确实性与完整性的保障。

3.6.3.1 报文序列号

在每条报文头里面增加一个序列号。在 NBRP 连接刚建立时被初始化为零，然后随每一个报文的发出递增。如果检测到某个序列号被跳过或者出现了重复，NBRP 发言者将在发送一个 NOTIFICATION 报文之后断开连接。序列号的范围必须足够大，保证它很难循环到 0。 即使循环到 0 也没有关系，这时原有连接将被断开，然后会重新建立一个新的连接，序列号会从 0 开始重新计数。

增加报文序列号可以防止报文被攻击者、重传或删除。

3.6.3.2 身份认证与加密

在 NBRP 连接建立的时候，NBRP 对等体之间将首先通过交换证书和密钥证书串进行身份认证，然后交换会话密钥，以后这条连接上的所有的 NBRP 信息都是由会话密钥加密的。这种加密提供了信息的保密性，同时保证了 KEEPALIVE 报文、NOTIFICATION 报文和 UPDATE 报文中某些域的完整性。当检测到发生了错误时，NBRP 发言者将发送一个 NOTIFICATON 报文之后断开连接。

我们首先来讨论密钥的生成和分发过程，然后介绍身份认证的具体过程。

（一）密钥生成与分发

在大规模的分布式内容路由系统中，每个名字路由器都可能与其他名字路由器所有通信，密钥的分配管理更是个极为复杂的问题。为了解决这一问题，提出了基于 RSA 的分布式密钥生成算法，使用该算法，可以保证所有生成的密钥在整个系统里唯一的，从而在不降低加密强度的基础上增强系统的扩展性。

1 分布式密钥生成

在一个典型的分布式路由系统中，相同管理层次上的所有路由器地位都是平等的，不存在任何特殊的实体。很自然的，考虑在系统中使用一种分布式的密钥生成方案，对应于某种密钥管理层次（可直接映射为路由管理层次），由分布的地位平等的密钥生成器来生成其管理区域的路由器的密钥，如图 3.15 所示。 图 3.15 中的自治域借用了 BGP 中的概念，可以把它和某个内容网络中的名字路由系统相对应，以简化密钥管理。子域可以对路由系统中的不同区域。

算法思想是：首先，给每一个密钥生成器分配一个全局唯一的密钥种子范围，这个范围包含了其负责区域的全局标识 $[Z', i']$，（网络区域 $Z=(z_1, z_2, \cdots, z_n)$）以及局部标识 i）；其次，保证其产生的密钥的某个固定部分落在此密钥种子范围内。进一步来说，假设第 g 个密钥生成器的密钥种子范围为 $[L_g, U_g]$，我们将产生如下的密钥，其低 m 位就

（2）在 NBRP 对等体建立连接时进行认证，通过认证后协商会话密钥。用此会话密钥加密所有的 NBRP 报文。这种方式可以保证路由信息的机密性；

2. NBRP 的 UPDATE 报文保护：

下面我们对这两类安全对策进行详细讨论。

3.6.3 通用 NBRP 报文保护

提供NBRP报文保护的目的是对相邻会话者之间通信的UPDATE报文提供机密性、确实性与完整性的保障。

3.6.3.1 报文序列号

在每条报文头里面增加一个序列号。在 NBRP 连接刚建立时被初始化为零，然后随每一个报文的发出递增。如果检测到某个序列号被跳过或者出现了重复，NBRP 发言者将在发送一个 NOTIFICATION 报文之后断开连接。序列号的范围必须足够大，保证它很难循环到 0。 即使循环到 0 也没有关系，这时原有连接将被断开，然后会重新建立一个新的连接，序列号会从 0 开始重新计数。

增加报文序列号可以防止报文被攻击者、重传或删除。

3.6.3.2 身份认证与加密

在 NBRP 连接建立的时候，NBRP 对等体之间将首先通过交换证书和密钥证书串进行身份认证，然后交换会话密钥，以后这条连接上的所有的 NBRP 信息都是由会话密钥加密的。这种加密提供了信息的保密性，同时保证了 KEEPALIVE 报文、NOTIFICATION 报文和 UPDATE 报文中某些域的完整性。当检测到发生了错误时，NBRP 发言者将发送一个 NOTIFICATON 报文之后断开连接。

我们首先来讨论密钥的生成和分发过程，然后介绍身份认证的具体过程。

（一）密钥生成与分发

在大规模的分布式内容路由系统中，每个名字路由器都可能与其他名字路由器所有通信，密钥的分配管理更是个极为复杂的问题。为了解决这一问题，提出了基于 RSA 的分布式密钥生成算法，使用该算法，可以保证所有生成的密钥在整个系统里唯一的，从而在不降低加密强度的基础上增强系统的扩展性。

1 分布式密钥生成

在一个典型的分布式路由系统中，相同管理层次上的所有路由器地位都是平等的，不存在任何特殊的实体。很自然的，考虑在系统中使用一种分布式的密钥生成方案，对应于某种密钥管理层次（可直接映射为路由管理层次），由分布的地位平等的密钥生成器来生成其管理区域的路由器的密钥，如图 3.15 所示。 图 3.15 中的自治域借用了 BGP 中的概念，可以把它和某个内容网络中的名字路由系统相对应，以简化密钥管理。子域可以对路由系统中的不同区域。

算法思想是：首先，给每一个密钥生成器分配一个全局唯一的密钥种子范围，这个范围包含了其负责区域的全局标识 $[Z', i']$，（网络区域 $Z=(z_1, z_2, \cdots, z_m)$）以及局部标识 i）；其次，保证其产生的密钥的某个固定部分落在此密钥种子范围内。进一步来说，假设第 g 个密钥生成器的密钥种子范围为 $[L_g, U_g]$，我们将产生如下的密钥，其低 m 位就

落在此区间内，用公式表示如下：

$$L_g \leq (n \bmod 2^m) \leq U_g$$

其中，n 是公钥，m 是公钥的固定部分位数（即低 m 位）。 在这里，采用了分布式的密钥生成方法，而且不需要一个能被所有人信任的特殊的路由器；一个被某路由器信任的密钥生成器，并不一定为其他路由器所信任。 这样就不再需要一个单独的受信实体。在消除了单一失效点的同时，也减少了产生系统瓶颈（尤其当使用密钥管理中心+ 密钥服务器生成和分发密钥时）的可能性。

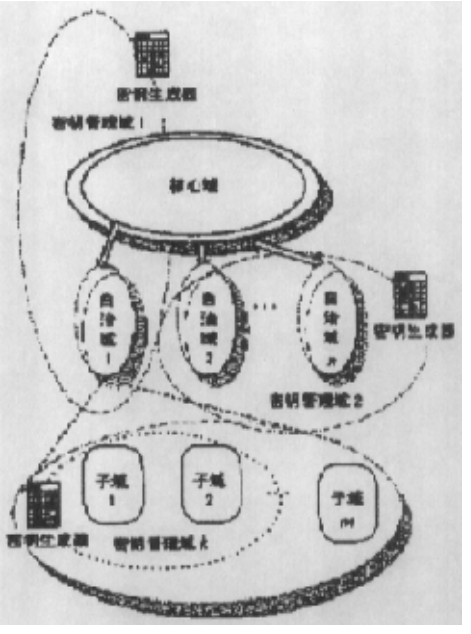


图 3.15　分布式密钥生成图

与此同时我们并没有降低任何安全性，因为在 RSA 加密体制中，加密密钥，是依赖于公钥的模数部分 n 的，而 n 是唯一的。这样一来，只有公钥的模数部分 n 需要被生成。

下面将给出一种基于 RSA 的生成算法：每当需要生成密钥时，可从密钥生成器 g 的密钥种子范围中随机选取一个种子数 $r \in [Lg, Ug]$，再将它作为公钥的模数部分（即 n）的低 m 位，换句话说，就是要求：$n = r(\bmod 2^m)$。为达到此目的，可用如下的算法生成：（设公钥的模数部分 n 约为 2L 位）

（1） 随机生成 L 位的大素数 p；

（2） 计算 $q_0 = \left(r * p^{\Psi(2^m)-1}\right)(\bmod 2^m) + 2^{L-1}$；

（3） 随机选取整数 i_0，计算 $q = q_0 + i * 2^m$，$i = i_0$，$i_0 + 1$，…；直至 q 为素数；若失败，则返回第 1 步；

（4） 由上节中计算 RSA 密钥的一般公式计算：$n = p * q$；选取 e、d 满足 $(e, \varphi(n)) = 1$；

$d*e=1\bmod\varphi(n)$

由算法可知，

$q=\left(r*p^{\Psi(2^m)-1}\right)(\bmod 2^m)+2^{L-1}+i*2^m$。所以，素数$q$的位数不低于L位，因而，$n$的位数约为$2L$位。此外，$q=\left(r*p^{\Psi(2^m)-1}\right)(\bmod 2^m)$。

所以，$n=p*q(\bmod 2^m)$；

所以，$n=p*r*p^{\varphi(2^m)-1}(\bmod 2^m)$；

而$p^{\varphi(2^m)}=1(\bmod 2^m)$；

故$n=r(\bmod 2^m)$，符合系统设计要求。

2 密钥的分发

密钥的分发在分布式路由系统中，每个参与路由的实体（Ri）都由信任实体配置一个证书（Ci），以证实此路由实体的基本信息，如路由的拥有者等。在密钥的分发中，Ri将证书（Ci）、本实体的公钥（Pi）以及本路由器的基本信息（Ii），用本路由器的私钥（Vi）加密签名，生成密钥证书串（Pi，Ci，Ii，$S(Vi,\ Pi,\ Ci,\ Ii)$）（S为签名函数），然后向外广播以进行密钥分发。

（二）身份的认证

身份认证的过程如下：（刚才已经讨论了密钥的分发过程，所以下面我们假定 NBRP 对等体互相知道对方的公钥）

（1） 请求建立连接的 NBRP 对等体（$B1$）向对方（$B2$）发送一条使用对方的公钥加密的包括自己的标识和一个随机数$R1$的报文。

（2） $B2$收到以后，使用自己的私钥解密，然后组织一条包括$R1$，会话密钥 K 和$B2$自己生成的随机数$R2$ 的报文并用$B1$ 的公钥加密，然后发送给$B1$。

（3） $B1$收到以后，看到该报文中有自己刚刚生成的随机数$R1$，因此它可以肯定对方是真的$B2$ 而不是冒充的，为了证明自己的身份，$B1$ 把$B2$生成的随机数$R2$ 用会话密钥 K 加密后发送给$B2$。

（4） $B2$收到这条报文后，发现$B1$送来了自己刚刚生成的随机数$R2$，因此$B2$ 也可以肯定对方是真正的$B1$。至此，身份认证过程完成。在这以后，所有的报文都可以使用会话密钥 K 加密传输了。

3.6.4 UPDATE域保护

用途是保护本地会话者和远程会话者之间的通信过程。为UPDATEE报文的属性域，尤其是在报文向前转发过程中不发生变化的属性=提供可靠性和完整性保证。

（1）增加UPDATE序列信息来防止重传攻击。

每条 UPDATE 报文都包括序列信息来防止重传攻击。这个序列信息是由产生路由的 NBRP 决策过程生成的，它可以采用序列号或者时间戳的形式。序列号或者时间戳用来防止攻击者

$d * e = 1 \bmod \varphi(n)$

由算法可知，

$q = \left(r * p^{\Psi(2^m)-1}\right)(\bmod 2^m) + 2^{L-1} + i * 2^m$。所以，素数 q 的位数不低于 L 位，因而，n 的位数约为 $2L$ 位。此外，$q = \left(r * p^{\Psi(2^m)-1}\right)(\bmod 2^m)$。

所以，$n = p * q(\bmod 2^m)$；

所以，$n = p * r * p^{\varphi(2^m)-1}(\bmod 2^m)$；

而 $p^{\varphi(2^m)} = 1(\bmod 2^m)$，

故 $n = r(\bmod 2^m)$，符合系统设计要求。

2 密钥的分发

密钥的分发在分布式路由系统中，每个参与路由的实体（*Ri*）都由信任实体配置一个证书（*Ci*），以证实此路由实体的基本信息，如路由的拥有者等。在密钥的分发中，*Ri* 将证书（*Ci*）、本实体的公钥（*Pi*）以及本路由器的基本信息（*Ii*），用本路由器的私钥（*Vi*）加密签名，生成密钥证书串（*Pi*，*Ci*，*Ii*，$S(Vi, Pi, Ci, Ii)$）（*S* 为签名函数），然后向外广播以进行密钥分发。

（二）身份的认证

身份认证的过程如下：（刚才已经讨论了密钥的分发过程，所以下面我们假定 NBRP 对等体互相知道对方的公钥）

（1） 请求建立连接的 NBRP 对等体（*B*1）向对方（*B*2）发送一条使用对方的公钥加密的包括自己的标识和一个随机数 *R*1 的报文。

（2） *B*2 收到以后，使用自己的私钥解密，然后组织一条包括 *R*1，会话密钥 K 和 *B*2 自己生成的随机数 *R*2 的报文并用 *B*1 的公钥加密，然后发送给 *B*1。

（3） *B*1 收到以后，看到该报文中有自己刚刚生成的随机数 *R*1，因此它可以肯定对方是真的 *B*2 而不是冒充的，为了证明自己的身份，*B*1 把 *B*2 生成的随机数 *R*2 用会话密钥 K 加密后发送给 *B*2。

（4） *B*2 收到这条报文后，发现 *B*1 送来了自己刚刚生成的随机数 *R*2，因此 *B*2 也可以肯定对方是真正的 *B*1。至此，身份认证过程完成。在这以后，所有的报文都可以使用会话密钥 K 加密传输了。

3.6.4 UPDATE 域保护

用途是保护本地会话者和远程会话者之间的通信过程。为UPDATEE报文的属性域，尤其是在报文向前转发过程中不发生变化的属性=提供可靠性和完整性保证。

（1）增加UPDATE序列信息来防止重传攻击。

每条 UPDATE 报文都包括序列信息来防止重传攻击。这个序列信息是由产生路由的 NBRP 决策过程生成的，它可以采用序列号或者时间戳的形式。序列号或者时间戳用来防止攻击者

重传 UPDATE 报文。它与 3.5.2.1.1 小节由相似性。

（2） 数字签名。

由发出 UPDATE 报文的 NBRP 发言者对 UPDATE 报文的所有的不变的域进行数字签名。为了保证 UPDATE 报文中不变信息的完整性和真实性，它将由初始的 NBRP 发言者进行数字签名。数字签名使用公钥机制，使用刚才讨论的密钥生成算法生成的密钥。UPDATE 的数字签名信息保存在 UPDATE 报文的 Marker 域中，它的计算包括以下的域：UPDADE 报文序列号、不可行路由长度（Unfeasible Route length）、撤销路由（Withawn Routes）、ORIGIN、AUTOMIC_AGGREGATE、AGGREGATOR、CLNI。

UPDATE 域保护措施的保护了 NBRP 转发报文信息。这类措施提供了认证机制和完整性控制。对于这类信息来说它的接收者是整个内容网络上的所有授权的 NBRP 发言者。

§ 3.7 本章小结

本章主要就基于名字路由技术的 CDN 路由系统的 NBRP 协议展开讨论。

本章首先概述了基于名字路由技术的 CDN 路由系统的 NBRP 协议，概括了 NBRP 协议在 CDN 路由系统中的作用和协议特征。简单介绍了路由表的建立原理以及 NBRP 协议在其中扮演的角色。

本章的随后几节围绕 CDN 路由系统的 NBRP 协议的几个组成部分展开的。首先介绍了 NBRP 协议的帧结构和 NBRP 协议的 FSM 有限状态机；接着详细分析了 NBRP 的路由处理进程，给出了 NBRP 路由的相关理论，并结合协议进行了详细分析；然后，从 NBRP 协议的性能出发，就 NBRP 路由的稳定性和安全性进行了详细的分析和探讨，提出了响应的解决方案；

本章还简单概括了 IP 路由系统的 BGP 协议和 CDN 路由系统的几个异同点。

重传 UPDATE 报文。它与 3.5.2.1.1 小节由相似性。

（2） 数字签名。

由发出 UPDATE 报文的 NBRP 发言者对 UPDATE 报文的所有的不变的域进行数字签名。为了保证 UPDATE 报文中不变信息的完整性和真实性，它将由初始的 NBRP 发言者进行数字签名。数字签名使用公钥机制，使用刚才讨论的密钥生成算法生成的密钥。UPDATE 的数字签名信息保存在 UPDATE 报文的 Marker 域中，它的计算包括以下的域：UPDADE 报文序列号、不可行路由长度（Unfeasible Route length）、撤销路由（Withawn Routes）、ORIGIN、AUTOMIC_AGGREGATE、AGGREGATOR、CLNI。

UPDATE 域保护措施的保护了 NBRP 转发报文信息。这类措施提供了认证机制和完整性控制。对于这类信息来说它的接收者是整个内容网络上的所有授权的 NBRP 发言者。

§ 3.7 本章小结

本章主要就基于名字路由技术的 CDN 路由系统的 NBRP 协议展开讨论。

本章首先概述了基于名字路由技术的 CDN 路由系统的 NBRP 协议，概括了 NBRP 协议在 CDN 路由系统中的作用和协议特征。简单介绍了路由表的建立原理以及 NBRP 协议在其中扮演的角色。

本章的随后几节围绕 CDN 路由系统的 NBRP 协议的几个组成部分展开的。首先介绍了 NBRP 协议的帧结构和 NBRP 协议的 FSM 有限状态机；接着详细分析了 NBRP 的路由处理进程，给出了 NBRP 路由的相关理论，并结合协议进行了详细分析；然后，从 NBRP 协议的性能出发，就 NBRP 路由的稳定性和安全性进行了详细的分析和探讨，提出了响应的解决方案；

本章还简单概括了 IP 路由系统的 BGP 协议和 CDN 路由系统的几个异同点。

第四章　基于名字的路由系统的系统设计

在前面几章中，我们主要基于名字的路由系统的 INRP 协议、NBRP 协议的几个主要内容作了详细的介绍，给出了一些仿真结果和结论。本章研究整个路由系统的系统设计等内容。

§4.1　系统的总体架构

图 4.1 给出了基于名字的路由系统的总体架构。我们把整个系统分为 System Service、Routing Protocl、System Manager 和 Service Provider 四个主模块。模块与模块之间通过数据交互和消息驱动的方式进行通信。

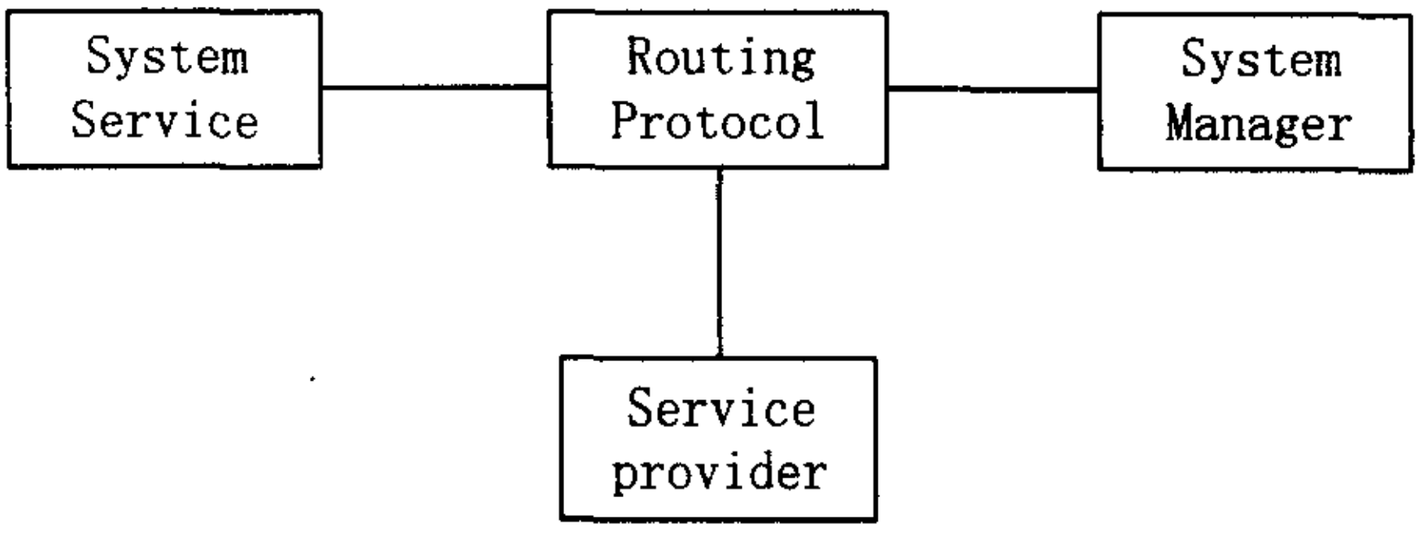


图 4.1　系统总体架构图

下面我们对各个模块进行简单说明。

System Service：　该模块主要提供堆栈管理、时钟管理、日期/时间管理、资源管理和初始化。堆栈管理包括堆栈队列的初始化、添加、删除和查找等操作。时钟管理包括定时器的初始化、取消和激活。日期/时间管理包括获取当前日期时间、系统时间等操作。资源管理包括内存资源的申请、释放和系统资源的监测等。初始化模块主要负责系统的启动和系统进程的注册管理工作。

System Mangger：　该模块提供路由系统的网管模块。包括配置管理、性能管理、计费管理等。它为系统的灵活管理和配置提供有力的平台。

Routing Protocol：　该模块运行 INRP、NBRP 协议。它包括 INRP 和 NBRP 的通信、协议的管理进程、路由信息的控制管理、协议消息的解析和编解码操作、协议 FSM 状态机的控制等。

Service Provider：　主要为系统提供必要的网络传输连接的建立、维持、监测和终止等功能。名字路由协议通过该接口模块收发协议报文消息。

§4.2　系统消息流和路由主进程

为了提高处理效率，减轻系统负担，整个系统设计为单进程结构，各个模块之间的交互采用函数调用和数据交换的方式。系统的核心模块是 Routing Protocol 模块。Routing Protocol 的主进程是 MgrTask。MgrTask 的任务是接收消息，同时根据消息的类型将消息分发到相应的模块。图 4.2 给出了 MgrTask 与各个模块之间的消息流。

第四章　基于名字的路由系统的系统设计

在前面几章中，我们主要基于名字的路由系统的 INRP 协议、NBRP 协议的几个主要内容作了详细的介绍，给出了一些仿真结果和结论。本章研究整个路由系统的系统设计等内容。

§4.1　系统的总体架构

图 4.1 给出了基于名字的路由系统的总体架构。我们把整个系统分为 System Service、Routing Protocl、System Manager 和 Service Provider 四个主模块。模块与模块之间通过数据交互和消息驱动的方式进行通信。

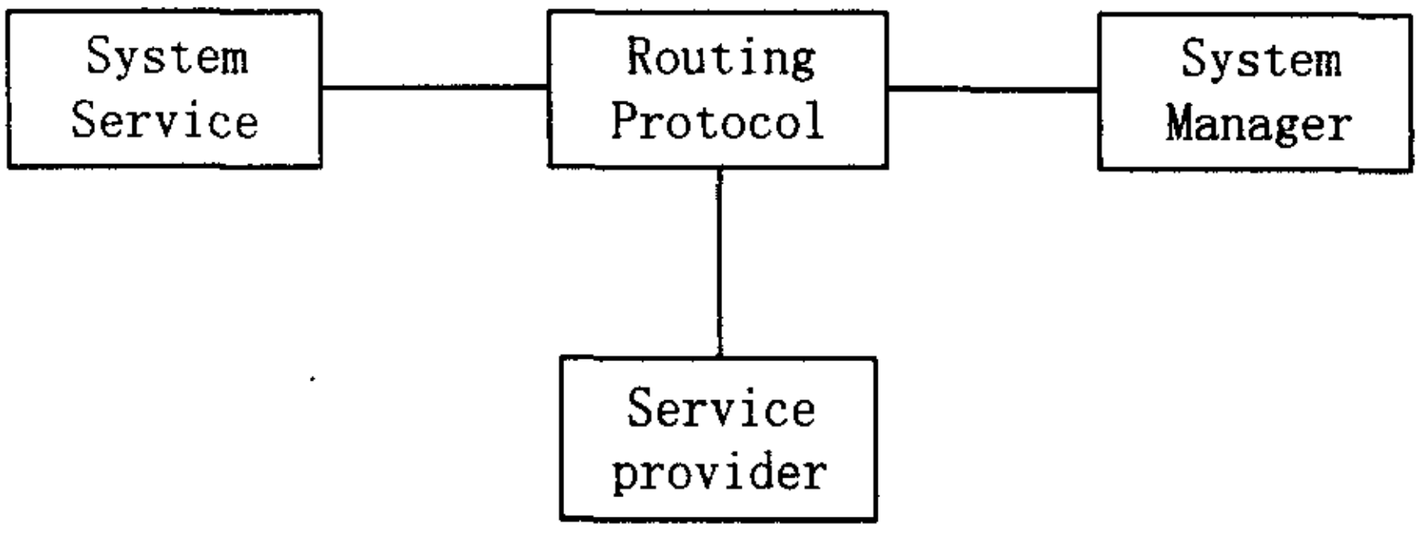


图 4.1　系统总体架构图

下面我们对各个模块进行简单说明。

System Service：　该模块主要提供堆栈管理、时钟管理、日期/时间管理、资源管理和初始化。堆栈管理包括堆栈队列的初始化、添加、删除和查找等操作。时钟管理包括定时器的初始化、取消和激活。日期/时间管理包括获取当前日期时间、系统时间等操作。资源管理包括内存资源的申请、释放和系统资源的监测等。初始化模块主要负责系统的启动和系统进程的注册管理工作。

System Mangger：　该模块提供路由系统的网管模块。包括配置管理、性能管理、计费管理等。它为系统的灵活管理和配置提供有力的平台。

Routing Protocol：　该模块运行 INRP、NBRP 协议。它包括 INRP 和 NBRP 的通信、协议的管理进程、路由信息的控制管理、协议消息的解析和编解码操作、协议 FSM 状态机的控制等。

Service Provider：　主要为系统提供必要的网络传输连接的建立、维持、监测和终止等功能。名字路由协议通过该接口模块收发协议报文消息。

§4.2　系统消息流和路由主进程

为了提高处理效率，减轻系统负担，整个系统设计为单进程结构，各个模块之间的交互采用函数调用和数据交换的方式。系统的核心模块是 Routing Protocol 模块。Routing Protocol 的主进程是 MgrTask。MgrTask 的任务是接收消息，同时根据消息的类型将消息分发到相应的模块。图 4.2 给出了 MgrTask 与各个模块之间的消息流。

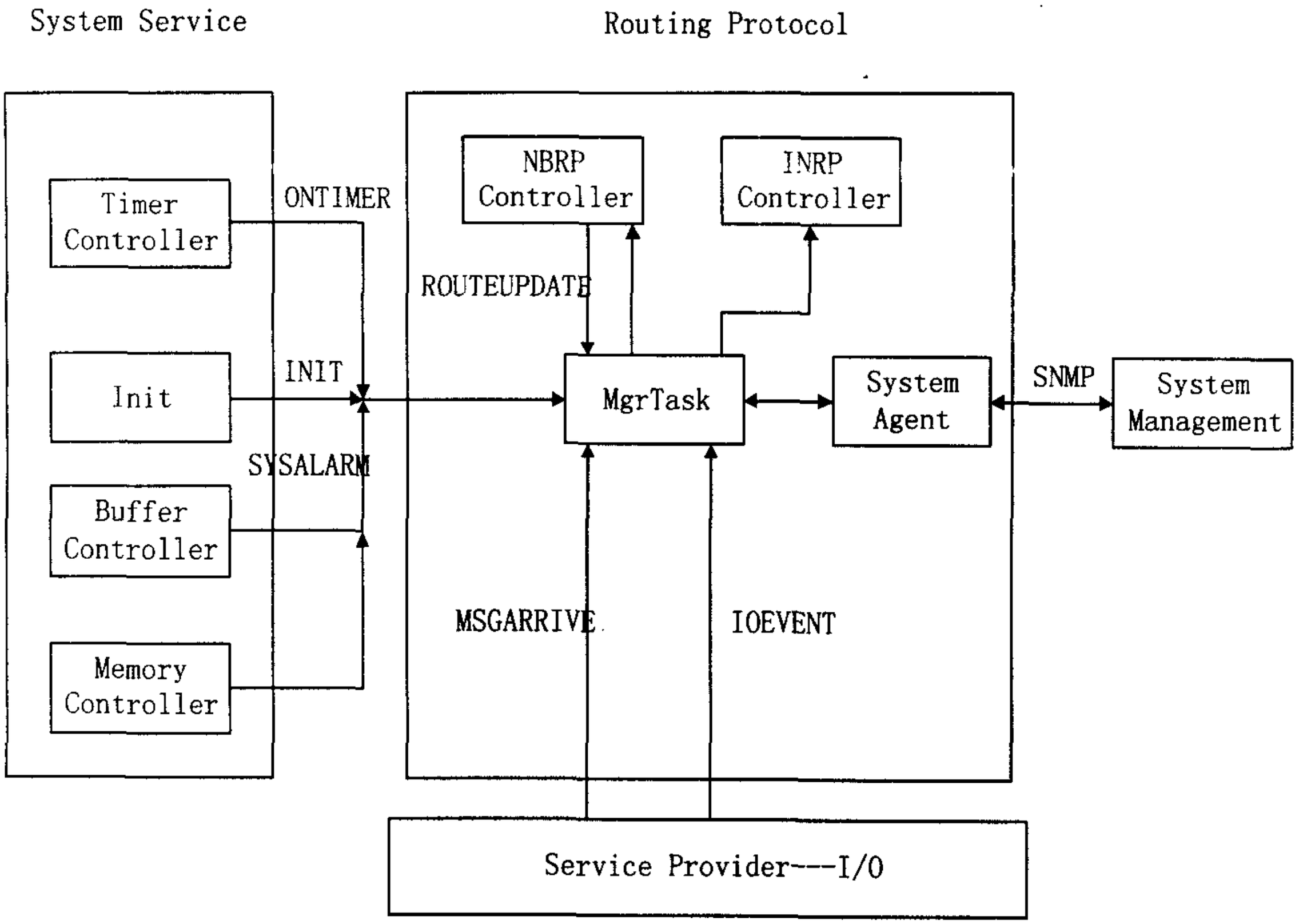


图 4.2　路由主进程与系统消息交互图

由图 4.2 可见，系统的消息可以分为：

1.　定时器消息。定时器超时的 ON_TIMER 消息。

2.　系统告警信息（SYSALARM）。主要是由 System Service 主模块的堆栈管理和资源管理发出，包括队列溢出，系统资源临界等消息。

3.　报文到达消息。负责网络连接的 Service Provider 主模块发现有新的报文（NBRP、DNS）到达时，将发出 MSGARRIVE 消息。

4.　系统初始化消息。System Service 发出的 INIT 消息，指示系统进行初始化工作。

5.　网管消息.System Manager 网管平台向路由系统发出的管理配置信息，一般是 SNMP 数据包的信息格式。

6.　路由更新信息。 当路由系统通过 NBRP 协议发现名字路由信息发生变化时，比如新的名字路由或者撤销旧的名字路由，运行 NBRP 协议的模块将发出 ROUTEUPDATE 消息，通知 INRP 协议更新本地名字路由表。

7.　I/O 状态消息。如果 Service Provier 发现 TCP、UDP 连接链路发生变化，如链路故障，它将发出 IOEVENT 消息。

图 4.2 同时给出了 System Service 的功能模块和 Routing Protocol 的一些主要模块。Sysem Service 分为 Timer Controller、Init、Memory Controller 和 Buffer Controller 四个模块。它们

分别为系统提供不同的系统服务。

Init： 启动整个路由系统。发送 Start 消息到 MgrTask 主进程；从 Config 数据库对用户配置文件或控制台输入进行分析，从中读取 NBRP 协议或 INRP 协议运行时所需的各项参数，同时进行语法和语义检查，把正确的参数写入 Config 数据库中；初始化全局数据结构(如事件队列、时钟链等)，并为某些结构分配空间；

Timer Controller：负责定时器时钟管理。为整个路由系统提供时钟链，包括定时器的初始化、激活、取消。但定时器超时等事件发生时，模块将向 MgrTask 主进程发送超时消息。MgrTask 根据消息，传递给 NBRP 和 INRP 协议控制块，进行状态状态机 FSM 的状态更新。

Buffer Contrller：负责堆栈缓冲等队列管理。存在的队列包括 FSM 状态机队列、MgrTask 消息队列、I/O 接收队列、I/O 发送队列和定时器队列等。该模块对这些队列实施统一的管理。包括队列的创建、添加、删除、更新等操作。

Memory Controller： 负责内存管理和系统资源的监控。当系统需要资源时，由该模块进行资源的申请和管理，包括资源的创建和释放等。该模块同时监控系统的资源耗费情况，当发现系统处于满荷状态时，将拒绝新的资源申请，并通过主进程 MgrTask 向网管报警。

Routing Protocol 的一些主要模块如图 4.2 所示。MgrTask 是路由主进程。NBRP Controller 和 INPR Controller 模块分别控制 NBRP 和 INRP 协议的运行。我们在接下来的 NBRP 功能框图和 INRP 的功能框图中对它们进行详细说明。还包括 System Agent 网管代理模块和 Config 本地数据库。下面是它们的一些简单说明。

MgrTask： 路由的主进程。主要任务是接收和分发消息到不同的功能模块，驱动事件处理模块。消息包括：定时器超时、I/O 事件（TCP 连接失败或成功）、系统初始化、网管命令、报文到达、路由更新和系统告警等。它是整个路由系统的核心模块。由于多进程机制不便于管理，易耗尽系统资源，为了提高处理效率，减轻系统负担，我们采用单进程结构，用 MgrTask 进程负责整个消息流的分发。

System Agnet：网管代理。接收从网管平台 System Manger 的消息，完成消息转换并传递给主进程 MgrTask，指示系统完成响应的动作。同时将系统的运行信息提供给 System Manger，供 System Manger 对系统运行状态进行分析，采取响应措施。System Agent 还包括对 Config 配置数据库进行配置。

Config：存放协议软件本地运行时所需的各项配置参数。这些参数可能来自本地配置文件，可以由用户通过控制台输入，也可以由 System Manger 通过网管代理 System Agent 配置。

NBRP Controller： 该逻辑模块负责控制和维护 NBRP 路由协议的运行。我们将在下面给出 NBRP 协议的功能框图，对该逻辑协议控制块进行细分。它可以分为 FSM Controller、NBRP Encoder、NBRP Decoder、RouteInfo Update 和 RIB 路由信息数据库。

INRP Controller： 该逻辑模块负责控制和维护 INRP 路由协议的运行。我们将在下面给出 INRP 协议的功能框图，对该逻辑协议控制块进行细分。它可以分为 Transaction Controller、DNSCach Check、DNS Encoder、DNS Decoder、RouteInfo Update 和 Routing Table 路由表。

§4.3 协议功能框图和流程例图

4.3.1 INRP 的功能框图和流程例图

分别为系统提供不同的系统服务。

Init： 启动整个路由系统。发送 Start 消息到 MgrTask 主进程；从 Config 数据库对用户配置文件或控制台输入进行分析，从中读取 NBRP 协议或 INRP 协议运行时所需的各项参数，同时进行语法和语义检查，把正确的参数写入 Config 数据库中；初始化全局数据结构(如事件队列、时钟链等)，并为某些结构分配空间；

Timer Controller：负责定时器时钟管理。为整个路由系统提供时钟链，包括定时器的初始化、激活、取消。但定时器超时等事件发生时，模块将向 MgrTask 主进程发送超时消息。MgrTask 根据消息，传递给 NBRP 和 INRP 协议控制块，进行状态状态机 FSM 的状态更新。

Buffer Contrller：负责堆栈缓冲等队列管理。存在的队列包括 FSM 状态机队列、MgrTask 消息队列、I/O 接收队列、I/O 发送队列和定时器队列等。该模块对这些队列实施统一的管理。包括队列的创建、添加、删除、更新等操作。

Memory Controller： 负责内存管理和系统资源的监控。当系统需要资源时，由该模块进行资源的申请和管理，包括资源的创建和释放等。该模块同时监控系统的资源耗费情况，当发现系统处于满荷状态时，将拒绝新的资源申请，并通过主进程 MgrTask 向网管报警。

Routing Protocol 的一些主要模块如图 4.2 所示。MgrTask 是路由主进程。NBRP Controller 和 INPR Controller 模块分别控制 NBRP 和 INRP 协议的运行。我们在接下来的 NBRP 功能框图和 INRP 的功能框图中对它们进行详细说明。还包括 System Agent 网管代理模块和 Config 本地数据库。下面是它们的一些简单说明。

MgrTask： 路由的主进程。主要任务是接收和分发消息到不同的功能模块，驱动事件处理模块。消息包括：定时器超时、I/O 事件（TCP 连接失败或成功）、系统初始化、网管命令、报文到达、路由更新和系统告警等。它是整个路由系统的核心模块。由于多进程机制不便于管理，易耗尽系统资源，为了提高处理效率，减轻系统负担，我们采用单进程结构，用 MgrTask 进程负责整个消息流的分发。

System Agnet：网管代理。接收从网管平台 System Manger 的消息，完成消息转换并传递给主进程 MgrTask，指示系统完成响应的动作。同时将系统的运行信息提供给 System Manger，供 System Manger 对系统运行状态进行分析，采取响应措施。System Agent 还包括对 Config 配置数据库进行配置。

Config：存放协议软件本地运行时所需的各项配置参数。这些参数可能来自本地配置文件，可以由用户通过控制台输入，也可以由 System Manger 通过网管代理 System Agent 配置。

NBRP Controller： 该逻辑模块负责控制和维护 NBRP 路由协议的运行。我们将在下面给出 NBRP 协议的功能框图，对该逻辑协议控制块进行细分。它可以分为 FSM Controller、NBRP Encoder、NBRP Decoder、RouteInfo Update 和 RIB 路由信息数据库。

INRP Controller： 该逻辑模块负责控制和维护 INRP 路由协议的运行。我们将在下面给出 INRP 协议的功能框图，对该逻辑协议控制块进行细分。它可以分为 Transaction Controller、DNSCach Check、DNS Encoder、DNS Decoder、RouteInfo Update 和 Routing Table 路由表。

§4.3 协议功能框图和流程例图

4.3.1 INRP 的功能框图和流程例图

分别为系统提供不同的系统服务。

Init： 启动整个路由系统。发送 Start 消息到 MgrTask 主进程；从 Config 数据库对用户配置文件或控制台输入进行分析，从中读取 NBRP 协议或 INRP 协议运行时所需的各项参数，同时进行语法和语义检查，把正确的参数写入 Config 数据库中；初始化全局数据结构(如事件队列、时钟链等)，并为某些结构分配空间；

Timer Controller：负责定时器时钟管理。为整个路由系统提供时钟链，包括定时器的初始化、激活、取消。但定时器超时等事件发生时，模块将向 MgrTask 主进程发送超时消息。MgrTask 根据消息，传递给 NBRP 和 INRP 协议控制块，进行状态状态机 FSM 的状态更新。

Buffer Contrller：负责堆栈缓冲等队列管理。存在的队列包括 FSM 状态机队列、MgrTask 消息队列、I/O 接收队列、I/O 发送队列和定时器队列等。该模块对这些队列实施统一的管理。包括队列的创建、添加、删除、更新等操作。

Memory Controller： 负责内存管理和系统资源的监控。当系统需要资源时，由该模块进行资源的申请和管理，包括资源的创建和释放等。该模块同时监控系统的资源耗费情况，当发现系统处于满荷状态时，将拒绝新的资源申请，并通过主进程 MgrTask 向网管报警。

Routing Protocol 的一些主要模块如图 4.2 所示。MgrTask 是路由主进程。NBRP Controller 和 INPR Controller 模块分别控制 NBRP 和 INRP 协议的运行。我们在接下来的 NBRP 功能框图和 INRP 的功能框图中对它们进行详细说明。还包括 System Agent 网管代理模块和 Config 本地数据库。下面是它们的一些简单说明。

MgrTask： 路由的主进程。主要任务是接收和分发消息到不同的功能模块，驱动事件处理模块。消息包括：定时器超时、I/O 事件（TCP 连接失败或成功）、系统初始化、网管命令、报文到达、路由更新和系统告警等。它是整个路由系统的核心模块。由于多进程机制不便于管理，易耗尽系统资源，为了提高处理效率，减轻系统负担，我们采用单进程结构，用 MgrTask 进程负责整个消息流的分发。

System Agnet：网管代理。接收从网管平台 System Manger 的消息，完成消息转换并传递给主进程 MgrTask，指示系统完成响应的动作。同时将系统的运行信息提供给 System Manger，供 System Manger 对系统运行状态进行分析，采取响应措施。System Agent 还包括对 Config 配置数据库进行配置。

Config：存放协议软件本地运行时所需的各项配置参数。这些参数可能来自本地配置文件，可以由用户通过控制台输入，也可以由 System Manger 通过网管代理 System Agent 配置。

NBRP Controller： 该逻辑模块负责控制和维护 NBRP 路由协议的运行。我们将在下面给出 NBRP 协议的功能框图，对该逻辑协议控制块进行细分。它可以分为 FSM Controller、NBRP Encoder、NBRP Decoder、RouteInfo Update 和 RIB 路由信息数据库。

INRP Controller： 该逻辑模块负责控制和维护 INRP 路由协议的运行。我们将在下面给出 INRP 协议的功能框图，对该逻辑协议控制块进行细分。它可以分为 Transaction Controller、DNSCach Check、DNS Encoder、DNS Decoder、RouteInfo Update 和 Routing Table 路由表。

§4.3 协议功能框图和流程例图

4.3.1 INRP 的功能框图和流程例图

图 4.3 给出了 INRP 的功能框图。

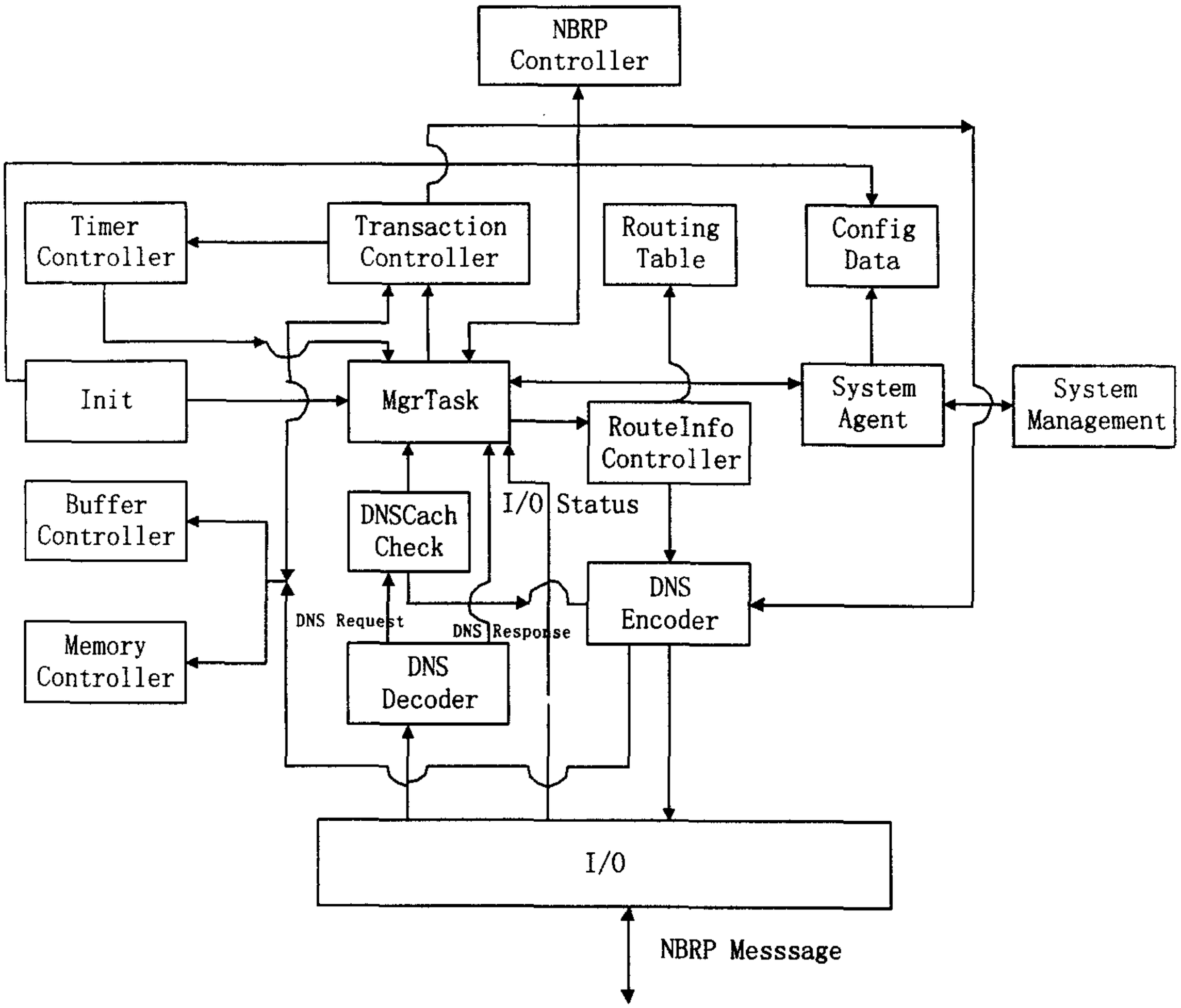


图 4.3 INRP 的功能框图

INRP 的协议控制块由以下几个模块组成：Transaction Controller、DNSCach Check、DNS Encoder、DNS Decoder、RouteInfo Controller 和 Routing Table 路由信息库。下面是这几个模块的说明。

Transaction Controller：管理维护 Transaction，接收 MgrTask 传递的报文消息。如果是 DNS Request，将向 Memory 申请资源，创建新的 Transaction，并由 Buffer Controller 插入到 Transaction 队列。Trasation 将负责路由查找、路由建立、报文转发等工作。如果是 DNS Response 报文，Transaction Controller 查找响应的 Transaction，产生状态变迁及相应动作，包括定时器的创建和释放，DNS cach 的更新等。本模块还通过 DNS Encoder 和 I/O 模块，发送 DNS 报文。

DNSCach Check： 缓存名字路由表的 INRP 域名对应记录。如果 DNS Request 命中本地 DNSCach ，将调用 I/O 和 DNS Encoder 模块向对应的路由器或用户发送 DNS Response。

DNS Decoder：对收到的 DNS 报文进行正确性检查和分析，向 System Service 申请报文消息的存放和管理，MgrTask 将被通告有报文消息到达。如果是 DNS resonse 报文，MgrTask 要通过调用模块 DNS Check 更新 DNS cach。

DNS Encoder：对 DNS 报文进行格式化，将报文放入发送队列，并调用 I / O 模块将其发出。

RouteInfo Conroller：接收 MgrTask 发来的消息，进行名字路由表的更新。同时负责对 DNS cach 的更新。路由更新请求可能来自网管用户需求，但主要来自与 NBRP 协议控制块的路由更新信息。I/O 链路故障导致某条网络连接不可用时，RouteInfo Controller 需要对路由表的路由信息更新。为了减小路由表的的规模，提高处理效率，这里还对 RoutintTable 中的路由进行合并和压缩。另外，本模块还将提供了对 Routing Table 中的数据结构进行操作的所有函数。

Routing Table：名字路由表存放所有 NBRP 协议软件所产生和 INRP 协议要利用的名字路由信息。

I / O：直接调用操作系统提供的 TCP、UDP 服务接口。完成 TCP、UDP 连接的建立和释放；接收相邻路由器送来的 DNS 报文，提交给 DNS Decoder 模块，接收 DNS Encoder 模块发来的格式化为字节流的 DNS 报文，发送给相应的名字路由器。此外，对这些操作中产生的相应事件，本模块还通过 MgrTask 发送 Transaction Controller。

图 4.4 和图 4.5 还给出了路由系统对 DNS Request 和 DNS Response 报文处理的流程例图。

◆ **DNS Request 的 INRP 流程：**

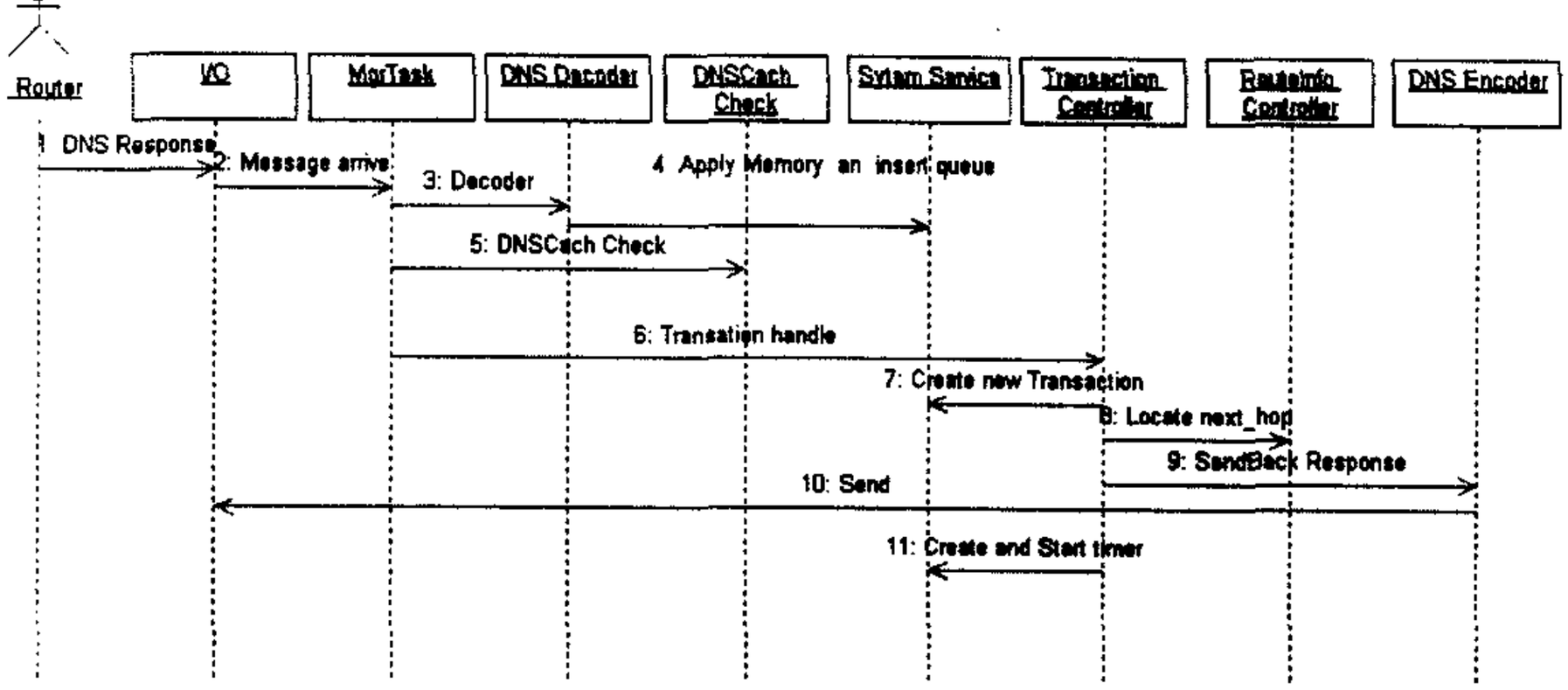


图 4.4　DNS Request 流程例图

◆ **DNS Resopnse 的 INRP 流程例图：**

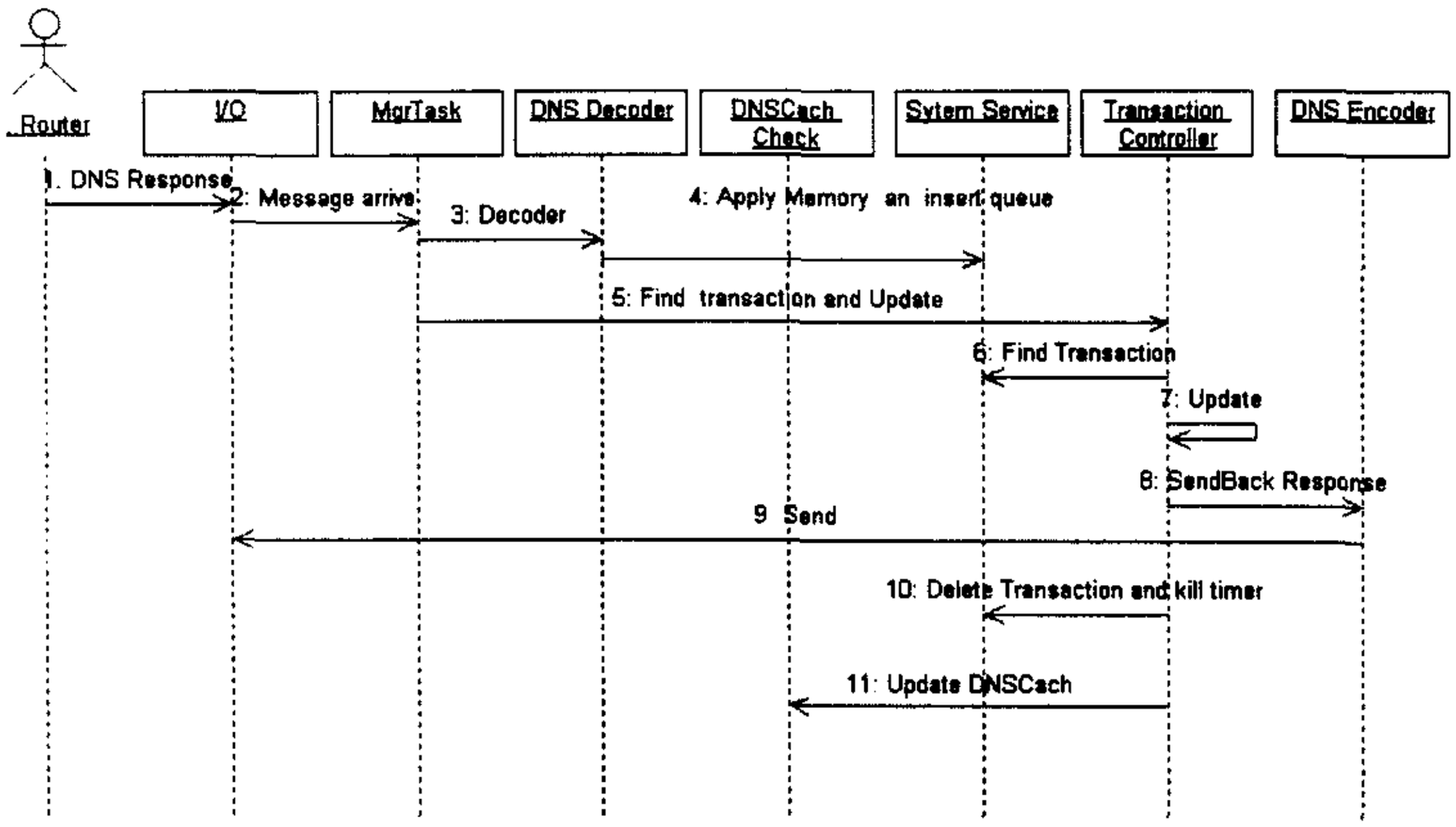


图 4.5 DNS Response 流程例图

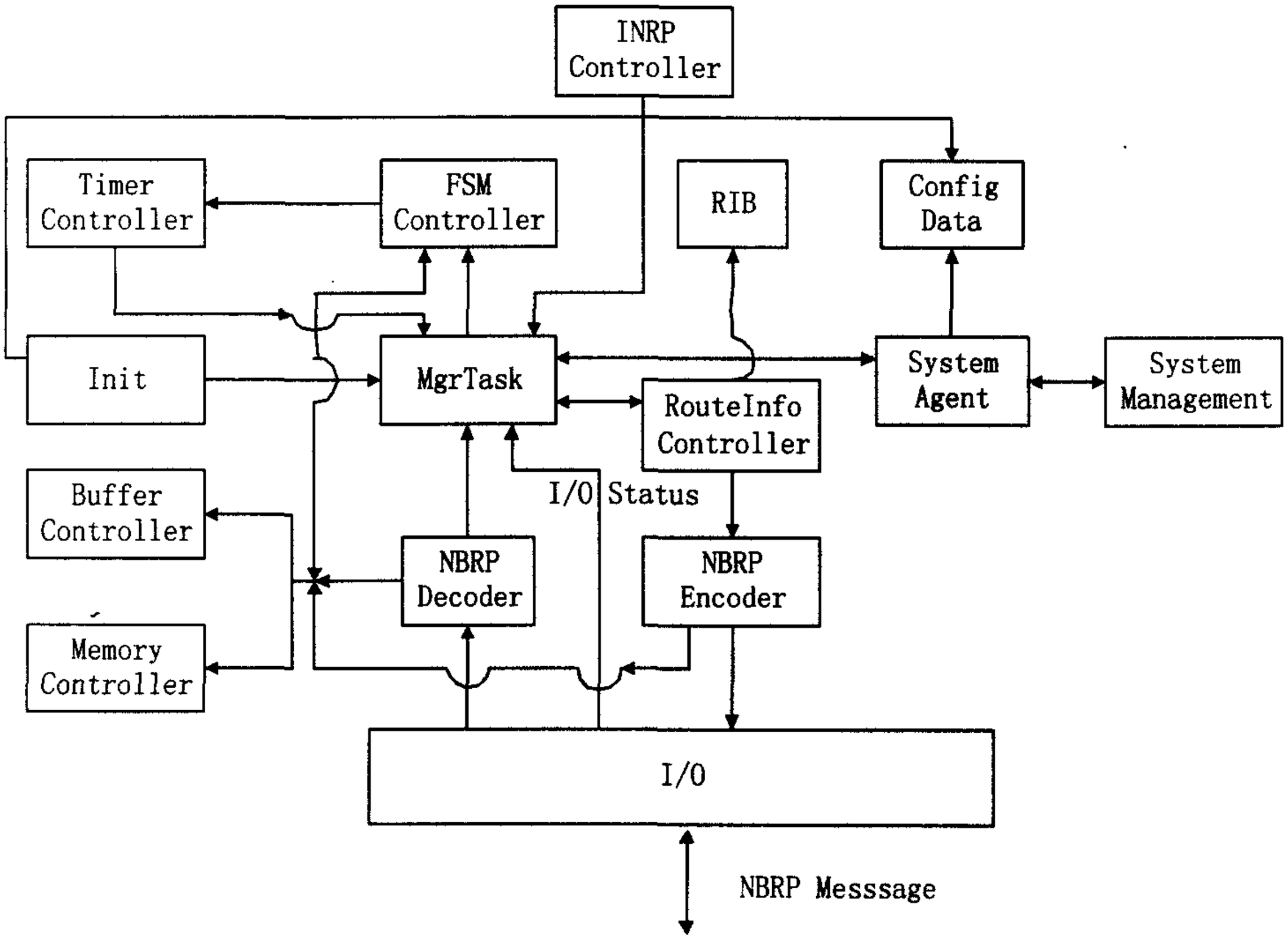


图 4.6 NBRP 的功能框图

4.3.2 NBRP 的功能框图和流程例图

图 4.6 给出了 NBRP 的功能框图。

NBRP 的协议控制块主要由以下几个模块组成：FSM Controller、Routing Controller、NBRP Decoder、NBRP Encoder 和 RIB 信息库。

FSM Controller：接收 MgrTask 送来的事件。如果是新的 Open 消息，将向 Memory 申请资源，创建新的 FSM 状态机，并由 Buffer Controller 插入到 FSM 状态机队列。事件作为有限状态机的输入，产生状态变迁及相应动作，包括定时器的创建和释放。

NBRP Decoder：对收到的 4 种 NBRP 报文进行正确性检查和分析，向 System Service 申请消息的存放和管理，MgrTask 将被通告有报文消息到达。如果是 UPDAT 正报文，MgrTask 要通过调用模块 Routing Information Processor 更新路由信息数据库 RIB，调用模块 Messager Encoder 向相邻的 NBRP 路由器发送协议报文，并通告 INRP 更新本地名字路由表。

NBRP Encoder：对 4 种 NBRP 报文进行格式化，并调用 I / O 模块将其发出。

RouteInfo Conroller：从 NBRP Decoder 模块中得到要撤销的和声明为有效的路由，进行路由选择，更新路由信息库 RIB。为了减小 RIB 的规模，提高处理效率，这里还对 RIB 中的路由进行合并和压缩。另外，本模块还将提供了对 RIB 中的数据结构进行操作的所有函数。本模块还将周期性地扫描路由器中的全局路由表，向相邻的 NBRP 路由器广播本地路由表的变化情况。同时，该模块还发出 ROUTEUPDATE 消息，通知 INRP 协议控制块更新本地名字路由表。

I / O：直接调用操作系统提供的 TCP 服务接口。完成 TCP 连接的建立和释放；接收相邻 NBRP 路由器送来的报文，由 MgrTask 提交给 NBRP Decoder 模块，接收 NBRP Encoder 模块发来的格式化为字节流的 NBRP 报文，发送给相应的 NBRP 路由器。此外，对这些操作中产生的相应事件，本模块还通过 MgrTask 发送给 FSM Controller。

RIB：存放所有协议软件所产生和要利用的路由信息的数据库。实际上分为 3 个相互独立的数据结构：Adj-RIBs-In 存放从其他 NBRP 路由器收到的路由信息，Loc—RIB 存放路由器中本地路由表的映射，Adj-RIBs-out 存放向相邻 NBRP 路由器广播过的路由信息.

◆ Open Message 的 NBRP 流程例图

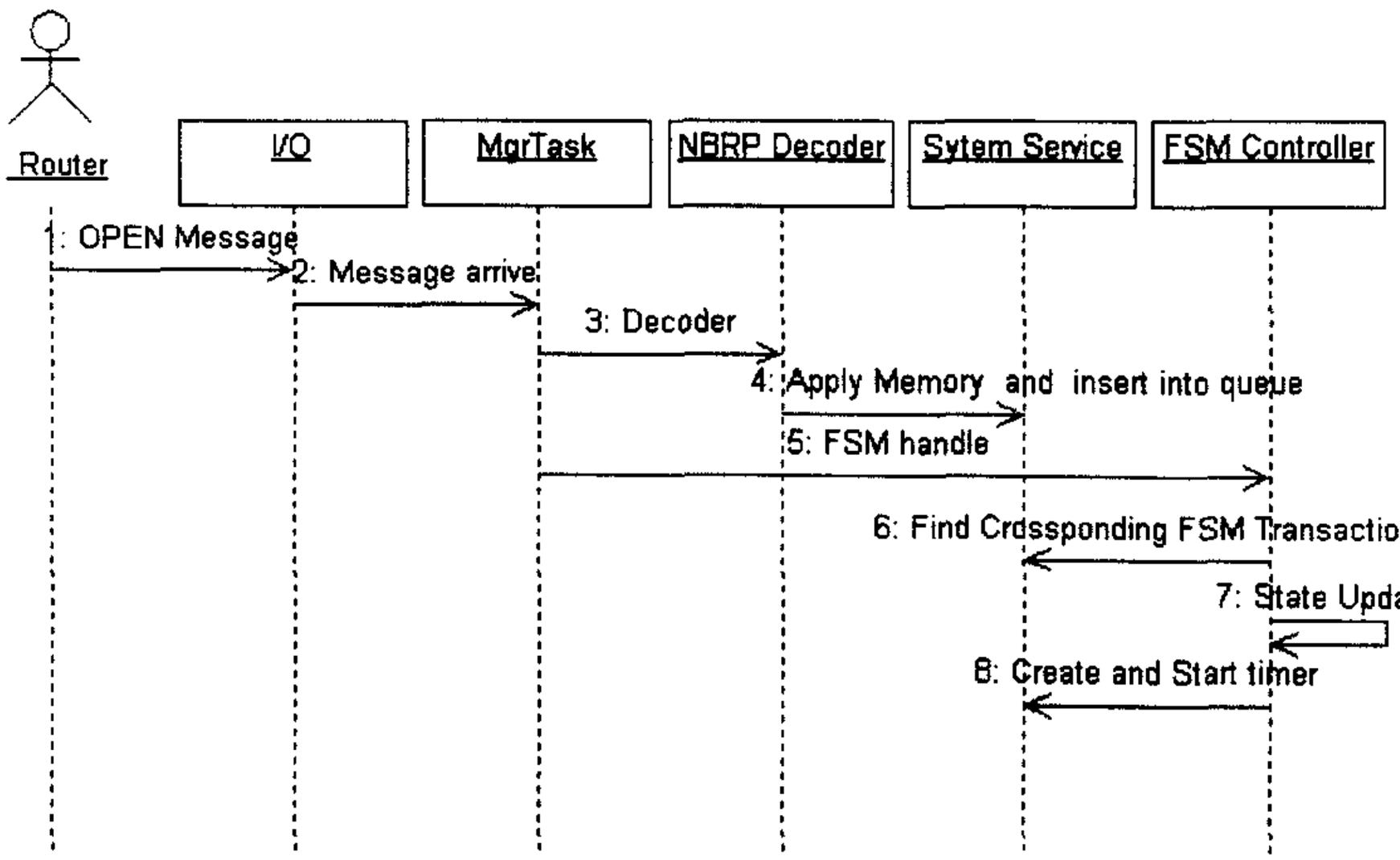


图 4.7　OPEN Message 流程例图

◆ Update Message 的 NBRP 流程例图

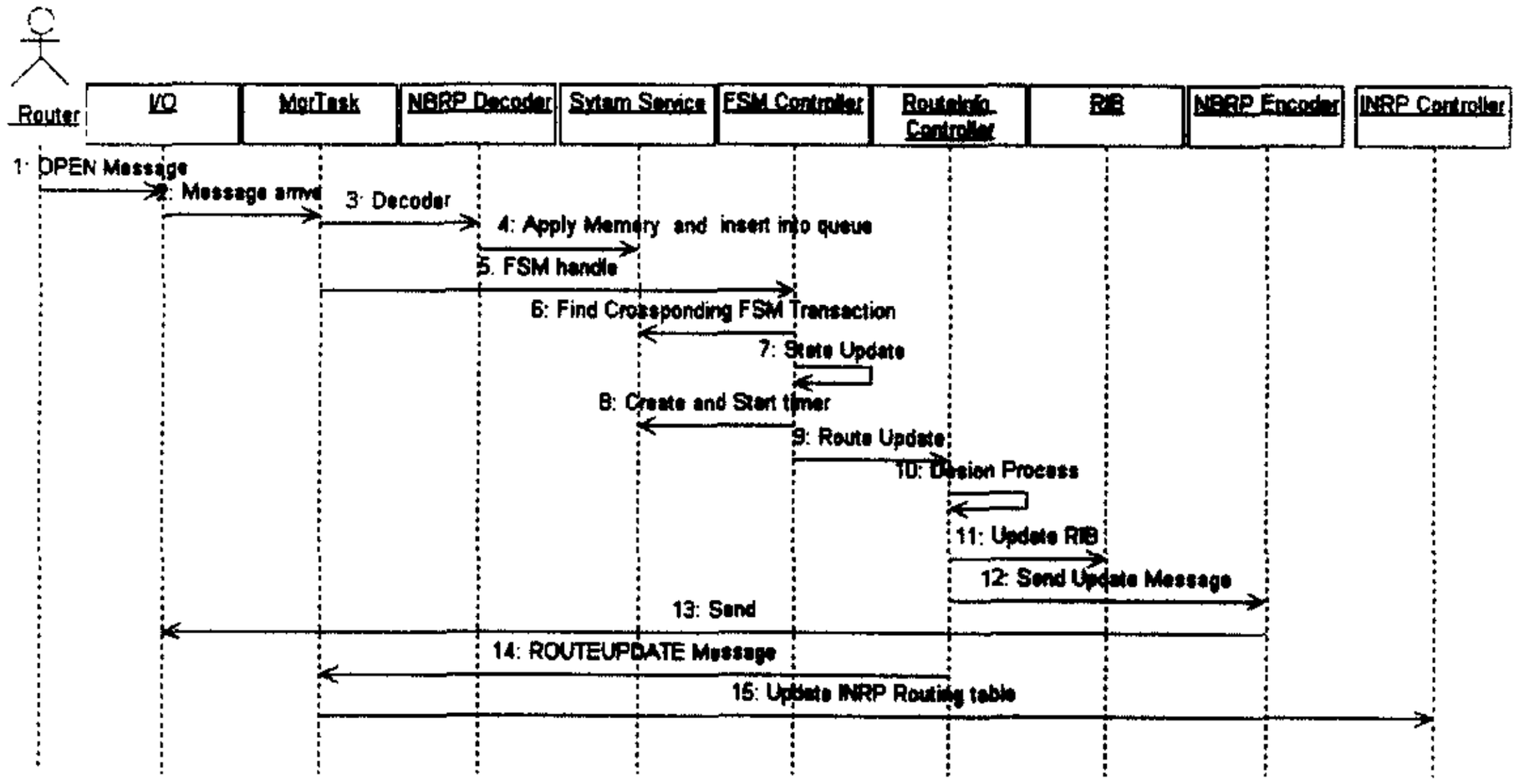


图 4.8 UPDATE Message 流程例图

4.3.3 INRP 和 NBRP 的通信

INRP 和 NBRP 的通信是单方面的。由于 INRP 和 NBRP 的分工不同，INRP 负责为内容请求建立路由，NBRP 负责创建和刷新名字路由表，因此 NBRP 将通过主进程 MgrTask 向 INRP 发出路由表更新命令。INRP 协议模块被动接收路由更新命令，进行名字路由表的刷新。图 4.9 给出了在 UPDATE 报文到达的情况下，INRP 和 NBRP 交互的流程例图。

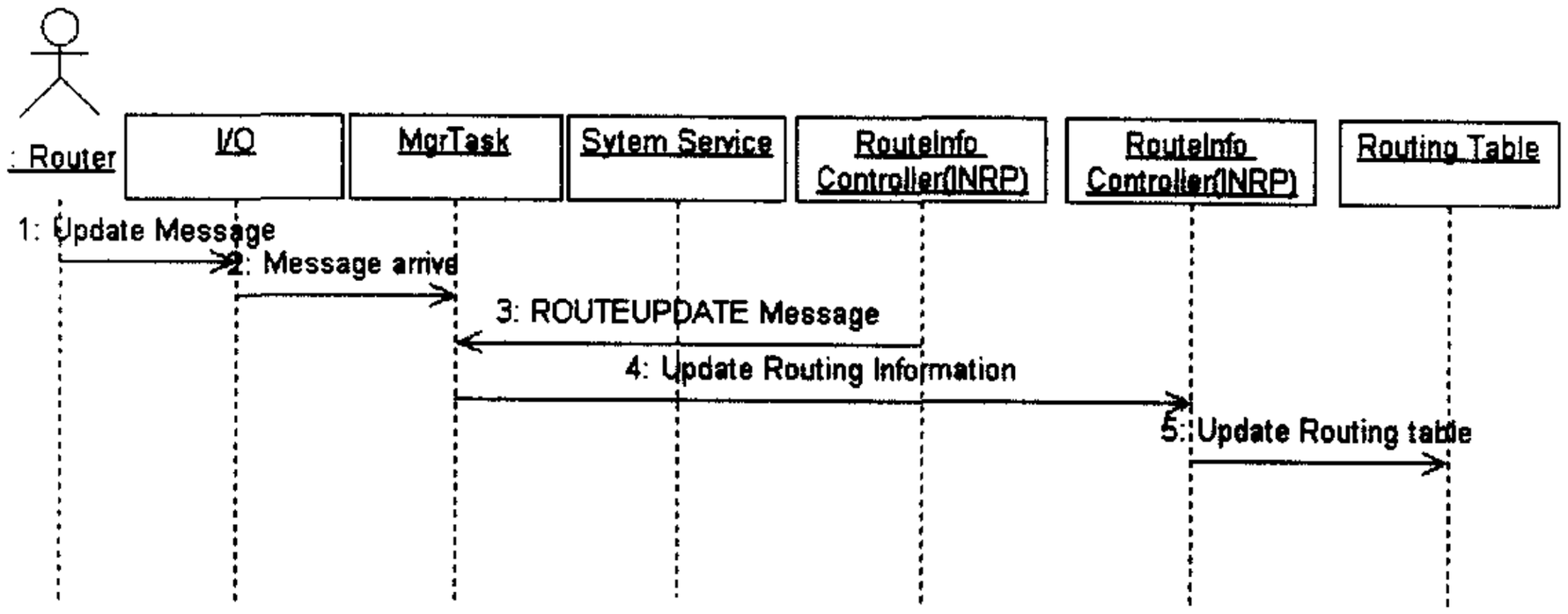


图 4.9 INRP 和 NBRP 的交互例图

§4.4 本章小结

本章讨论名字路由系统的系统设计。

首先，我们给出了路由系统的系统架构，对其中的主要模块进行了说明；为了避免多进程的弊端，采用了单进程机制。以路由主进程为中心，列出了系统消息以及各个主要模块并对它们具体说明。

最后，本章分别给出了 INRP 和 NBRP 的功能框图和主要报文流程例图。同时，还讨论了 INRP 和 NBRP 的通信机制，并给出了流程例图。

结论和展望

如何提高网络信息传递性能是当今通信界一个研究和开发的热点，CDN（内容传送网络）技术的出现就是为了改善互联网性能。它的目的是：让内容离用户更近。它通过实现用户对网站服务器的就近访问及网络流量智能分流，从技术上全面解决由于网络带宽小、用户访问量大、网点分布不均等对用户访问效果的影响，大大提高网络的响应速度。

CDN 网络是网络界的新事物，对它的研究进展日新月异。CDN 路由技术是一项崭新的技术，尤其是本文讨论的名字路由技术，作者曾经在追踪它的最新进展方面花了很大一部分精力，确立研究主题并进行深入研究。它们是 CDN 路由系统的为用户请求建立路由的 INRP 协议模块、创建和刷新路由表的 NBRP 协议模块。总结全文，作者认为本文在以下几个方面体现它的理论价值和实际意义：

1. 首先就基于名字路由技术的路由机制展开讨论。简单介绍了基于名字路由技术的路由机制的主要流程，提出了几个主要功能模块，对其中的几个模块进行了较深入的讨论。首先分析了查询报文的接收进程，针对实现过程中将涉及的一些问题进行了讨论；然后，为了提高路由的性能，引入了 DNS 的高速缓存查找；事务处理进程是名字路由的主要模块，我们对它进行了详细分析，提出了事务处理进程的 FSM 的有限状态机模型；然后我们介绍了名字路由技术的表驱动算法，介绍了名字路由的最长后缀匹配方式，对名字路由表结构进行了分析和改进；然后对表搜索算法进行了分析比较，对 HASH 算法进行了详细的讨论；然后概述了针对 INRP 的路由模型的路由仿真算法，并给出了 INRP 的性能仿真结果。

2. NBRP 协议解决了 CDN 路由系统的路由表建立和刷新问题。首先对基于名字路由技术的 CDN 路由系统的 NBRP 协议进行了简单介绍，概括了 NBRP 协议在 CDN 路由系统中的作用和协议特征；详细描述了 NBRP 协议的帧结构和 NBRP 协议的 FSM 有限状态机；接着详细分析了 NBRP 的路由处理进程，给出了 NBRP 路由的相关理论，并结合协议进行了详细分析，给出了 NBRP 路由决策过程的算法；由于 NBRP 本地策略带来的不稳定性，我们通过 3 节点问题模型的仿真分析，提出了增强 NBRP 稳定性的措施；然后我们对 NBRP 协议的安全性能进行了详细的分析和探讨。由于 NBRP 本身存在安全隐患，通过分析，分两个方面对 NBRP 的安全性能进行了扩展。

由于时间有限，与本课题有关的许多议题还没有涉及，许多关键技术还有待于深入展开和实际验证。我们将在后续的研究工作中对下列问题进行深入的研究：

1. 名字路由系统的开发，包括 INRP 和 NBRP 模块。虽然本文对这两个核心模块进行了很多方面的讨论，并给出了系统设计方案，但具体的开发还有待于下一步的工作。

2. INRP 的重定性问题。尽管本文提出了 INRP 重定向的解决方案，但是不够完善，还需要进一步的分析、研究。

3. 路由系统的网管问题。完善的管理手段和先进的管理技术是名字路由系统实用化的基本保障，因而 CDN 路由系统的网络管理问题必须提上议事日程。

4. 名字路由系统和现有网络的无缝连接问题。CDN 的目的是为了改善网络性能，它只有应用于实际骨干网络才能体现它的价值。

致　　谢

本文是在导师赵问道副教授的悉心指导下完成的。几年来，导师始终重视对作者各方面能力的培养，使作者的独立科研能力有较大的提高。导师渊博的知识，平易近人，严谨的治学态度给作者留下了深刻的印象。在此，作者谨向他表示最衷心的感谢。

陈惠芳副教授也给予作者很多帮助。在写论文的过程中，多次提供很多指导意见，作者向她表示最衷心的感谢。

同时，作者要感谢通信与信息研究所得各位老师对作者在浙江大学信电系求学期间给予的指导和帮助，感谢所有提供过帮助的老师和同学。

最后，衷心的感谢作者家人对作者的支持和理解，感谢徐丽对我的帮助和建议，感谢她用爱和信心伴我同行。

作者：　金世杰

2003.2

参 考 文 献

【1】 White Paper, The Ins and Outs of Content Delivery Networks, Stanford University, 2001

【2】 White Paper, Content Networking and Edge Services: Leveraging the Internet for Profit. Stanford University, 2001

【3】 M. Gritter, An Architecture for Content Routing Support in the Internet, USITS '01 presentation, March 26, 2001

【4】 M. Gritter, The TRIAD Content Layer: An Internet Architecture for Content Routing Support, Stanford networking seminar, November 2, 2000.

【5】 Border Gateway Protocol 4(BGP-4), RFC 1771, March 1995. Douglas E. Comer

【6】 www.cisco.com

【7】 www.agilent.com

【8】 www.itef.org

【9】 www-dsg.stanford.edu

【10】计算机网络（第三版），清华大学出版社，1998.11

【11】离散时间系统模拟，清华大学出版社，1988.3

【12】用 TCP / IP 进行网际互联一：原理. 协议与结构（第四版），电子工业出版社 ，2002-1-1

【13】UML with Rational Rose 从入门到精通，电子工业出版社，2001.10