第6章 DNS：名字服务器

作者：Rima S. Regas

本章内容包括：

- 域名系统概述
- 授权局
- DNS分布数据库
- 域和区
- Internet顶级域
- 选择一个域名服务器
- 名字服务解析过程
- 高速缓存
- 反向解析(Pointer)查询
- DNS安全
- 资源记录

在以前，网络上的用户需要维护一个 HOSTS配置文件，这个文件包括当此工作站和网络上的其他系统通信时所需要的一切信息。问题是显然的，因为每台机器的 HOSTS文件需要手工单独更新，几乎没有自动配置，因此使得 HOSTS文件更新是既枯燥又费时的工作。

HOSTS文件包括名字和IP地址的对应信息。当一台计算机需要定位网络上的另一台计算机时，就会查看本地HOSTS文件，如果在HOSTS文件中没有关于此计算机的表项，说明其不存在。域名系统 (DNS)改变了这一切。 DNS允许系统管理员使用一个服务器作为 DNS主机。当网络上的计算机要定位另一台服务器时，它就会查看 HOSTS文件。

今天，仍在使用HOSTS文件，这仅仅是为防止局域网上的一台机器使用DNS查找一台本地机器，因为通过HOSTS文件查找本地会更快。简单的讲，计算机的网络软件在使用缺省 DNS服务器之前，首先查看本地 HOSTS文件来指导定位，如果在 HOSTS文件中存在一个匹配，客户端软件就会直接和远端的主机通信，这样一来就极大地缩短了通过 DNS发现IP地址的时间。

6.1 域名系统概述

使用名字的原因是名字比数字更容易记忆。虽然大多数人有记忆电话号码、地址、数量以及其他与生命相关特征的惊人能力，但是存在如此众多的 IP地址，记住它们绝非易事，因此产生了命名。举例来说， C|Net的域名是www.cnet.com，这个域名没有竖线和大写字母，与站点的名字比较，域名的含义相当不明显，但是，域名和站点的名字还是相当好地吻合了。C|Net是世界上最大的且最经常被访问的站点之一。

在出现DNS之前，人们曾想让一个系统负责把名字翻译成号码。当翻译或解析一个 Web站点的域名(如www.cnet.com)并且找到了域名对应的 IP号(204.162.80.181)时， IP号就是实际的地址。这样Internet内容就可以传送到你的 Web浏览器上。这个过程需要一个称为 DNS或域

名服务器(系统)的网络系统。当前，这些服务器都连接到一个叫做网络解决方案 (NSI)的公司，这个公司位于弗吉尼亚。NSI，也就是Internet网络信息中心(InterNIC)，负责对域名进行所有权管理和分发。

其中的一部分工作是提供基本的根域名服务器，其他的 DNS服务器会查找此服务器得到所需要的信息。其结果是，全世界有许多共享信息的 DNS服务器，这样，它们所在地的用户就能找到它们的站点和其他的 Internet资源。这意味着如果用户的 DNS服务器不能解析一个域名到IP地址，服务器就会和另外的 DNS服务器联系。假如那个 DNS不能找到域名，它还将继续搜索直到超时。在这种情况下，就会返回一个错误，如果客户端容许的话，还会显示一条错误消息。在Web站点不能被发现的情况下，浏览器就会显示一条错误信息 —— 不能定位服务器或存在DNS错误。

DNS就如其组织结构分层一样，从顶级 DNS根服务器向下延伸，并把名字和 IP地址传播到遍布世界的各个服务器上。一个能清楚说明 DNS如何使用分层结构的例子是 DNS服务器不在本地存储全部的名字和IP地址的映射，一旦DNS服务器在自身的数据库中没有找到 IP地址，它会请求上一级DNS服务器看是否能找到这一 IP地址，这个过程会继续下去直到超时或找到答案。下一节会更详细地讨论这一结构，因为它还有另一个更重要的特性。

DNS结构化组织

在1982年出版的RFC819文档中，Zaw-Sing SU和Jon Postel提出了DNS概念和DNS计划。下面摘出文档中的一段话：

“域名集构成层次。使用图论表示，这种层次可以模型化为有向图。有向图由一组点和一组弧组成，弧由不同的有序点对标识 [1]，图中每个点表示一个域，有序点对 (B, A)，即B到A的弧，说明B是A的一个子域，B是一个简单的名字，且在A内惟一”。

这句话不是很好理解，我们会以一种更简单的方式解释这个概念。其中的原理很简单，用户有一个顶级域，如COM或EDU。COM和EDU域是最高的域级，不会有比它们更高的域，但是有一个例外。实际上，在 COM后有一个句号(COM.)，那个句号是比COM更高的域，但是这个句号没有使用意义，因为它仅仅有表示作用且通常被忽略。顶级域 (如COM、EDU和ORG)称为通用名，因为它们包含层次在其下面的域和子域，它们非常像树根。从顶级移至中间级，中间域名的例子包括coke.com、whitehouse.gov以及disney.com，这些域名可以被注册，但是这样做通常仅仅是为了覆盖相关域名范围。在一个子域内注册一个完全的域是很常见的，SU和Postel称这些域为端域。端域的例子包括www.coke.com、www.whitehouse.gov以及www.disney.com。当然，这些都是InterNIC注册系统中有代表性的域，美国之外的系统使用不同的方式来指示位置和国家，如www.bbc.co.uic，是指BBC的Web站点是一个商业站点 (CO和COM相似)位于英国(UK)。

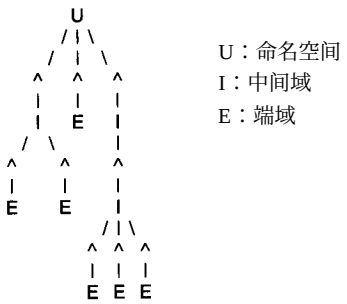

图6-1 原始域层次示意图

图6-1显示了Su和Postel提出的最初域层次图。

子域可以是任何东西，这样的例子是www. netscape. com、home.netscape.com及www1. netscape. com，所有这

些可以位于Internet上非常不同的地方，并且经常包含不同的信息。当然如果 Web管理员愿意，这些都可以指向一个站点或者一个页面。 Netscape就是这样做的，可以通过所有的URL列表访问Netscape。

注意 作为另一个例子，Webopedia选择注册webopedia.com作为子域，这是因为许多人对古典希腊字母不熟悉。

6.2 授权局

DNS的组织方式允许服务器成为根名字服务器的下属机构控制一个域。一个很好的例子是为个人或公司驻留 Web站点的本地ISP。当X公司向InterNIC注册了域名(www. companyx. com)时，就宣布它的ISP的主控和辅助DNS服务器作为其DNS服务器。 InterNIC会把相应信息放进.com根服务器上，让其传播。

注意 DNS服务器周期性地和其他DNS服务器上的各种数据库同步，并检查其他服务器上的新表项。这个过程通常称为传播。域名注册过程不是瞬时完成的，但是一个新域名大约会在3～4天内完成传播，能在世界各地获得相关信息。

为了使这种系统能工作，需要一个方法来决定网络上的一台机器位于何处。这就需要引入系统层次概念，对机器按功能进行分类。假如要查找的机器位于教育机构，它的顶级域应是.EDU。假如一个站点是商业机构它就应位于 .COM顶级域中。这个概念形成了 DNS层次中的根，但不是根名服务器。

6.3 DNS分布数据库

DNS分布数据库结构分布相当广泛，功能非常强大。前面已述及为了更好地控制域或子域内的流量，可以把权力分配给处于更佳位置的其他服务器。这种信息传播到其他服务器的方式，通过完全的分布计划来完成。

每个域指定一个所有者，定义为域授权开始 (SOA)的一部分， SOA将在后面作详细介绍。顶级域如COM，为一个域名，如 disney.com授权给一个特定 DNS服务器作为其基本 DNS服务器。这样减轻了顶级域控制器为每一个 Internet上的DNS请求作处理的负担。一旦一个 SOA与一个域控制器联系起来，它就能把子域控制授权给其他的 DNS服务器，等等。这就是授权从最顶级分布到最低层的工作方式。

6.4 域和区

域和区经常成对出现，但是二者有些微妙的区别。一个区在名字世界中是一个基本域，它授权给另一个DNS服务器以便管理。 Disney.com是一个区，但www.disney.com实际上是那个区内的一个子域。管理的责任授权给一个基本的 DNS服务器， Disney站点的基本DNS服务器是huey.disney.com，其IP地址是204.128.192.10，因为指定huey为Disney的基本DNS，那么它也是Disney.com的基本区。这个服务器的指定所有者是 root @ huey.disney.com，在SOA中为root.huey.disney.com。

注意 要使用正确格式的SOA，因为SOA会在各个阶段通过去掉所有邮件地址中的@符号改变管理员的邮件地址，有的公司喜欢在电子邮件地址的 @前加一分隔符，这样就

会出现问题，因为如果一个管理员的电子邮件地址是 host.admin @ example.com，在SOA中将成为host.admin.example.com。任何要解析那个地址的系统将假设电子邮件地址为host @ admin.example.com，这样只能收到一个DNS错误(认为那不是有效地址)。

考虑这样一种情形， thing_1.com作为用户的域， thing_1.com的DNS服务器称为 dns.thing_1.com，它作为thing_1.com的授权DNS。thing_1. com的一个子域可以是www.thing_1.com，也可以是home.thing_1.com。这个子域的基本DNS服务器是dns.thing_1.com。如果用户想注册另一个域名thing_2.com， 也可以授权给dns.thing_1.com，这两个域名将位于双区内，由thing_1和thing_2组成。任何数目的子域可以加入到任何一个区内，只要子域名在它们的区内惟一。

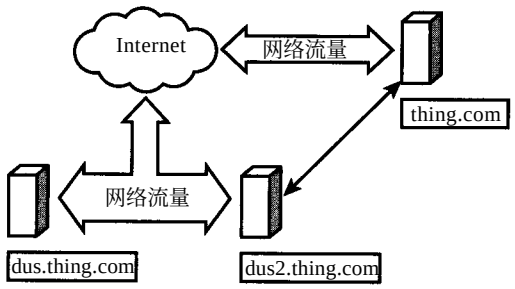

图6-2 thing.com区和DNS基本示图

图6-2显示了一种更复杂的情形，图中建立了一个 thing.com的区，授权给主控 DNS：dns.thing.com和辅助DNS：dns2.thing.com，这为用户提供了基准域。

现在，加入一个子域，典型情况下加入 www.thing.com。为了流量均衡目的，用户可以加入ww1.thing.com和ww2.thing.com。流量可以在内部处理，使新的流量调度到最少被使用的服务器上。现在用户的区如图 6-3所示。

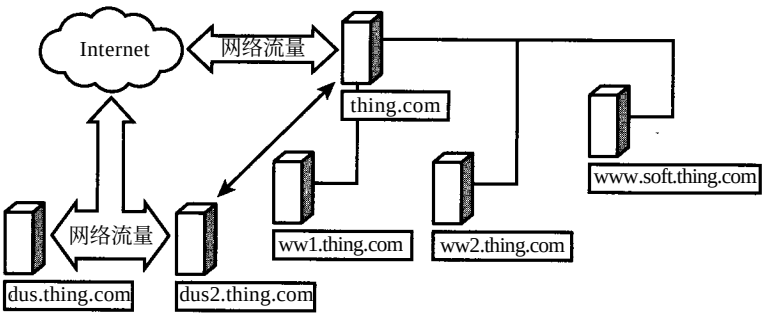

图6-3 三个新服务器作为区主服务器和两个DNS服务器的下属

注意所有的DNS流量仍将穿过 thing.com。虽然看起来次级服务器直接访问 DNS服务器，但它们是通过thing.com的网关IP来完成访问的。

6.5 Internet顶级域

有大量的顶级域(TLD)，最熟悉的是COM、EDU、GOV、MIL、NET和ORG。然而还有许许多多的TLD。每个国家有自己双字母的TLD，UK是英国的TLD，NZ是新西兰的TLD，JP是日本的TLD等等。一个使用TLD的例子是www.bbc.co.uk。

注意 有许多美国之外的外国地址域使用CO表示商业，CO类似于InterNIC中的COM。

警告 没有什么能阻止人们在其他国家注册域，即使服务器与URL指示的国家距离更远一些。

6.6 选择一个域名服务器

在今天，选择一个域名服务器有些不必要。 Windows NT/2000具有一个内嵌的适合于各种平台的应用程序。苹果有一个称为 MacDNS的自由软件。UNIX和相关操作系统一直提供一个可靠的BIND软件。还有许多功能强大的第三方 DNS服务器可用。

寻找服务器的最好地方是 Dava中心(www.davecentral.com)和Tucows(www.tucows.com，正准备选择一个本部服务器作为分支机构)。它们会考察每一个软件，并给用户提供建议：哪个软件流行，哪个软件不流行。流行的更好一些，因为它说明人们愿意使用它，不是所有的人都能忍受界面不友好的应用程序。

6.7 名字服务解析过程

当一个客户端程序，如浏览器，产生一个 URL请求交给本地 DNS服务器时，这个服务器会尽可能把名字分析成号码。假如解析成功，数据会被向前发送，开始下一段旅程， DNS服务器处理下一个请求。假如服务器不能定位地址，它有两个选择，这要依赖于服务器的配置方式，这两个选择是递归 (Recursive)和叠代(Iterative)。

6.7.1 递归查询

递归查询是最典型的。如果到达服务器的查询在 Cache中找到了一个记录，就不再需要更进一步的查询。假如没有找到，服务器会请求另一个服务器，这个过程就像爬梯子一样。这里要引入存活时间(Time to Live, TTL)机制。如果查询费了很多时间也没找到记录，查询过程会停止，发出请求的DNS服务器会返回一个“ address not found ”错误。

6.7.2 叠代查询

叠代查询被强制在局部范围内进行，这其中是有原因的。最主要的原因是另一个 DNS服务器不可得到。递归特性可能在用户访问的服务器上被禁止了，但是这不大可能。服务器会定位Cache中最好的匹配。如果没发现，就说明不在那里， DNS会返回一个错误。

6.8 高速缓存

由于DNS整天都在忙碌，它会挑选资源存储起来。这些资源记录 (Resource Record, RR)包括被查询URL的信息。TTL机制在这里也起作用，服务器会高速缓存尽可能多的并且有效的信息。

6.9 反向解析（Pointer）查询

典型的查询是前向的，尽力解析一个 URL的IP地址。反向解析恰恰相反，尽力找到 IP对应的URL。有一些程序可以为用户做到这一点，但是为 Windows开发的一个最好程序是 Luc Nieijens的CyberKit。这是个免费网络软件，可以从 www.ping.be/cyberkit/上下载。

6.10 DNS安全

用户可以升级到动态 DNS服务器来保证安全，因为它们的记录可以在没有管理员干预的

情况下自动更新。

6.11 资源记录

所有的DNS记录有相似的格式。虽然在 DNS文件中可以找到有许多快捷的表示方式，但是这些例子使用最简单的术语以消除二义性。

DNS记录中的第一个域总是 IP地址或主机名。如果此域为空的话，前一个记录的名字或地址就适用于此记录。注意所有的名字或地址以点结尾 (.)，这表明名字或地址是绝对的而不是相对的。绝对地址，也称为全称域名，是相对于根的，然而相对地址相对于缺省域 (可能不是根)，这个域可以跟可选的生存时间 (TTL)值，它指出此域中的信息有效的时间长度。

第二个域指出地址类型。在今天的 DNS数据库中，串“ IN ”指明是Internet地址。这个域的存在有其历史原因，为的是和旧的系统兼容。

第三个域是字符串，指出资源记录的类型。这个域后跟一个可选的与 RR相关的参数。

1. 授权开始（SOA）

记录存储DNS系统的名字及其负责人姓名。下面是一个 RR顶部SOA的例子：

```
；Start of Authority (SOA) record
dns.com. IN SOA dns1.dns.com   owner.dns.com (
                    0000001; serial # (counter)
                    10800; refresh (3小时)
                     3600; retry(1小时)
                    604800 ; expire(1星期)
                     86400); TTL (1天)
```

这里要注意的一些事情是所有者以 owner.dns.com列出，应读作owner @ dns.com，同时后面跟一个左括号，右括号结束于 TTL值之后。这对于方便读记录很重要，用下面的方式显示仍然有效：

```
Dns.com. IN SOA dns1.dns.com owner.dns.com.
[ic:ccc] (0000001 10800 3600 604800 86400)
```

分号(；)在这里是解释符，分号后的语句被忽略。最后要注意的是所有的数字以秒记数，各项内容在后序节中讨论。

2. 序列号

序列号标明DNS数据库的可用版本。当数据库被更新时，这个数必须加 1，这样辅助服务器知道该用哪一个。本例中的序列号是一个简单的记数器，也可以使用一个数据或其他的记数系统来满足用户的喜好和需要。

3. 刷新

刷新告诉每一个辅助名字服务器检查和更新信息的频率 (10 800秒是3小时)。

4. 重试

如果辅助名字服务器不能和主控名字服务器联系，它就会每隔“ retry(重试) ”秒重新试图联接(3600秒是1小时)。

5. 超时

如果辅助名字服务器经过“ expire(超时) ”秒后未能和主控名字服务器相联系，那么辅助名字服务器就停止应答任何关于此域的请求。这样做的理由是：在某些时候，一些数据太旧

了，可能会带来损害，不应答会比应答坏的信息要好一些 (604 800是一星期)。

6. TTL

TTL(生存时间)与数据库请求的应答一起返回，告诉请求者 (或其他服务器)信息可缓存(cache)的最长时间(86 400秒是一天)。这个值是文件中所有记录的缺省值，这个值可被一个RR提供的值所覆盖。

7. 地址资源记录

A记录中包含和域中第一个主机名相关的IP地址。

8. 名字服务（NS）资源记录

NS记录包括名字服务器的地址：在这个例子中，有两个名字服务器具有关于 foo.com DNS的信息。

9. 规则名字记录（CNAME）

CNAME记录包含一个主机别名，这个主机别名与记录第一个域中的主机名相联系。虽然可以给一个系统指定一个 A记录和多个CNAME记录，但是要小心。一些邮件程序解析 MX主机名时，使用CNAME记录而不是A记录，这样一来会发生一些不可预测的事情。

10. 指针（PTR）记录

指针是容许一个域指向另一个域的记录。在网络上使用多个 URL建立一个新名时，通常使用这种记录类型。前面提及的Netscape是个很好的例子，它的三个URL实际上仅指向一个。

11. 授权域

授权域把一个域的管理责任从一个服务器移向另一个子服务器。 InterNIC对.com和.有管理控制权，但是把所有在它们下面注册的域授权给各自的控制域。实际上，大多数站点是驻留的，这意味着服务器空间卖给了个人，站点的所有者对硬件不需要维护，它的工作是提供驻留服务，对驻留在站点上域的控制权经常授权给 DNS服务器管理员，让其维护主机上的 DNS服务器。有一个免费的DNS服务器可以为用户提供空间，其站点是http://soa.granitecanyon.com。

6.12 小结

在这一章读者了解了Internet中域名系统的层次结构和控制。这个系统使名字成为 Internet中资源寻址的基本形式成为可能。读者也学到了许多 DNS确定域名IP地址的方法。需要重点记住的是，和其他系统一样，这个系统也会崩溃。如果读者对这些内容还是不清楚或对它的结构感到迷惑，就让ISP来处理，这毕竟是访问Web站点或在其他协议下定位资源的方式。