

Universidade Federal do Rio Grande do Sul
Instituto de Informática
Curso de Bacharelado em Ciência da
Computação

Um estudo sobre o DNS na
Internet

por
Gustavo Silveira Barlem

Profa. Dra. Maria Janilce B. Almeida -
orientadora
Porto Alegre, julho de 1997



Resumo

Este trabalho tem como intuito buscar um esclarecimento mais profundo sobre um assunto pouco estudado na área da redes. O Domain Name System, também conhecido como DNS ou Sistema de Nomes de Domínio é responsável por toda a conversão de nomes para endereços realizada em toda a rede mundial. Tem também um grande e valioso uso em redes corporativas com muitas máquinas, facilitando a administração e gerenciamento dos nomes.

O DNS é uma especificação aberta podendo ser usada também para outros protocolos que não o TCP/IP, possibilitando o uso em redes heterogêneas. Devido a escassez de materiais e documentos sobre este tema, a sua complexidade de uso, e havendo poucos administradores de redes que conheçam o sistema de domínio com a devida abrangência, este trabalho foi realizado.

[Animação ilustrando o funcionamento do DNS](#)

Sumário

Lista de Figuras

Lista de Tabelas

Resumo

1. [Introdução](#)

2. [Noções Básicas](#)

- 2.1 Uma Breve História da Internet
- 2.2 Sobre A Internet e as internets
- 2.3 A História do DNS
- 2.4 Uma visão geral do DNS
- 2.5 A História do BIND

3. Funcionamento

3.1 O Espaço de Nomes de Domínio

3.1.1 Nomes de Domínio

3.1.2 Domínios

3.1.3 Registros de Recursos

3.2 O Espaço de Nomes de Domínio da Internet

3.2.1 Domínios *Top-level* (primeiro nível)

•

3.3 Aprofundando-se mais

3.3.1 Lendo Nomes de Domínio

3.4 A Distribuição de Domínios

3.5 Servidores de Nomes

3.5.1 Delegando Domínios

3.5.2 Tipos de Servidores de Nomes

3.5.3 Arquivos de Dados

3.6 Os resolvers

3.7 A Resolução

3.7.1 Servidores de Nomes da Raiz

3.7.2 Recursão

3.7.3 Interação

3.7.4 Mapeando Endereços para Nomes

3.7.5 Buscas invertidas

3.8 A Cache

3.8.1 O Time to Live

4. Parâmetros

4.1 Classes

4.2 Tipos de Registros de Recursos

4.3 Formato das Mensagens

4.4 Códigos de Operação

4.5 Códigos de Resposta

5. Conclusão

Apêndice 1 - Sites Úteis sobre DNS na Internet

Apêndice 2 - RFCs relacionados ao DNS
Apêndice 3 - Domínios Top-Level

Bibliografia

Lista de Figuras

- Figura 2.1 Comparação entre a base de dados do DNS e um sistema de arquivos UNIX
- Figura 2.2 Forma de leitura dos nomes no DNS e em um sistema de arquivos UNIX
- Figura 2.3 Administração remota de sub-domínios e sistemas de arquivos
- Figura 2.4 Um "alias" do DNS apontando para um nome canônico
- Figura 2.5 Resolvendo-se o problema da colisão de nomes
- Figura 3.1 A estrutura do espaço de nomes do DNS
- Figura 3.2 Garantia de nomes únicos no DNS e em sistemas de arquivos
- Figura 3.3 O domínio "ufrgs.br"
- Figura 3.4 O diretório "/usr"
- Figura 3.5 Um nodo em mais de um domínio
- Figura 3.6 Um nodo do interior da árvore com dados estruturais e de host
- Figura 3.7 O domínio "ufrgs.br" é delegado à Universidade Federal do RS
- Figura 3.8 O domínio "ca"
- Figura 3.9 A zona "ca"
- Figura 3.10 Resolução do nome "beethoven.telesc.gov.br" na Internet
- Figura 3.11 O processo de resolução de nomes
- Figura 3.12 O domínio in-addr.arpa
- Figura 3.13 Hierarquia de nomes e endereços IP
- Figura 3.14 Resolução do nome caracol.inf.ufrgs.br
- Figura 4.1 Formato da mensagem

Lista de Tabelas

- Tabela 4.1 Classes do DNS
- Tabela 4.2 Tipos mais comuns de registros de recursos
- Tabela 4.3 Tipos de dados e perguntas da classe Internet
- Tabela 4.4 Códigos de Operação
- Tabela 4.5 Códigos de resposta

1. Introdução

No começo, os usuários de computadores tinham que programar o sistema com o uso de chaves de comutação para definir os endereços das máquinas e inserir informações. Com o passar dos tempos, a linguagem assembler foi desenvolvida de modo que os usuários pudessem usar alguns símbolos e instruções simples para programar as máquinas. Hoje em dia, as linguagens de programação de alto nível aceitam instruções em inglês, que são o fáceis de entender. Com as linguagens mais novas, os usuários de computador não têm que se preocupar mais em saber endereços de memória específicos para variáveis - podem referir-se a eles simplesmente através dos nomes das variáveis.

Essa analogia também é verdadeira na interligação em rede. Em vez de ter que endereçar um host específico através de seu endereço numérico, é mais fácil usar um nome simbólico. Esse método melhora a produtividade porque os usuários cometem menos erros. Isso também ajuda na identificação do local e do tipo de recurso que uma determinada máquina fornece.

Qual designador de máquina é mais fácil de entender: *Servidor de Banco de Dados de Clientes* ou *199.246.41.8*? Com base no nome, você pode dizer facilmente que a máquina pertence ao departamento de Atendimento ao Cliente e que é um servidor de banco de dados. Mas ao olhar para o número IP, seu palpite é pode ser qualquer um! Outra vantagem em acessar um recurso através do nome é que seus usuários não terão que saber se você moveu aquele serviço fisicamente de uma máquina para outra.

Por exemplo, ao usar nomes, seus usuários irão se conectar ao Servidor de Banco de Dados para acessar os arquivos de banco de dados. Se algum dia você decidir mover os arquivos de banco de dados de um host com o endereço IP 199.246.41.15 para outro host com o endereço IP 199.246.41.17, terá simplesmente que alterar o mapeamento do nome para o endereço IP. Seus usuários não irão nem notar. Contudo, se você usar endereços IP, terá que informar a todos os usuários sobre a alteração - um possível pesadelo no gerenciamento.

É fácil atribuir um nome a qualquer máquina em sua instalação. Contudo, como você poderá garantir que ele será único? Em geral, se você não estiver conectado com o mundo exterior, como a Internet, e em um local relativamente pequeno, terá boas chances de controlar a atribuição de nomes e cada nome que usar será único. Contudo, logo que você se unir a uma rede nacional, ou mesmo regional, as máquinas sob controles administrativos diferentes sofrerão conflitos em relação a nomes. Mesmo se você não se unir a qualquer rede externa, como poderá evitar que nomes duplicados venham a ser usados em sua empresa enquanto sua rede IP crescer?

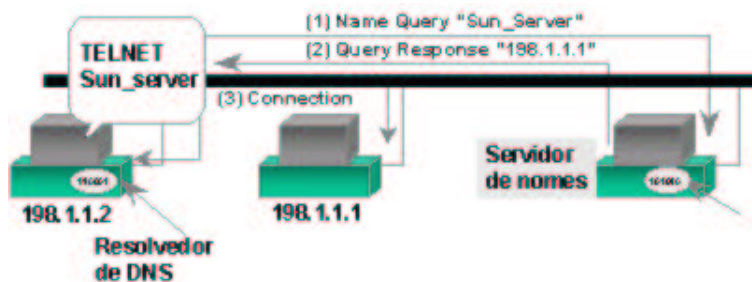
Há duas maneiras para implementar uma tabela de pesquisa de nomes para endereços IP. Você pode ter uma tabela local para cada estação de trabalho ou uma centralizada para todos os usuários. A solução da tabela local é ideal para um pequeno número de usuários ou para quando você quer que cada usuário seja capaz de usar sua própria

convenção de atribuição de nomes. Tal implementação é conhecida como tabela de hosts local. Contudo, você não tem controle direto sobre o conteúdo do arquivo porque ele reside em todas as máquinas. Quando uma alteração for necessária, será preciso atualizar todas as máquinas. A vantagem desse método é que se um usuário desordenar sua tabela de hosts, ele não afetará nenhuma outra tabela.

A solução ideal é usar algum tipo de sistema de atribuição de nomes centralizado que traduza nomes para endereços IP. Se uma administração centralizada for usada juntamente com um banco de dados, não haverá chances de duplicação. O maior inconveniente desse esquema é que se você cometer um erro, ele afetará todos os usuários, ao contrário da implementação da tabela de hosts local.

A Internet tomou exatamente essa abordagem para gerenciar sua grande quantidade de nomes. Inicialmente, o "namespace" (ou espaço de nomes) da Internet era *contínuo*. Cada nome consistia em uma seqüência de caracteres, sem outra estrutura. Embora isso tenha simplificado a atribuição de nomes, a Internet logo achou que o namespace contínuo poderia facilmente lidar com grande quantidade de nomes que existem hoje. Até 1990, havia mais de 137.000 nomes de hosts registrados na Internet [ARN 94].

Enquanto o número de hosts registrados crescia, a carga de trabalho envolvida na manutenção da atualização da quantidade de tráfego para um único local acumulava. Para resolver esses problemas, um esquema de atribuição de nome hierárquico, denominado DNS (Domain Name System), foi desenvolvido. Para ajudar a reduzir o gargalo do tráfego e adicionar redundância a todo o sistema, foi decidido que o namespace seria dividido em domínios diferentes, com cada um tendo máquinas de mapeamento de nomes para o endereço IP.



O DNS é usado principalmente para a conversão de nomes para endereços. Não deve ser confundido com NIS (Network Information Services). O NIS faz mais do que converter nomes para endereços, ele também fornece informações ao usuário, como a user id (uid - identificação do usuário) e a group id (gid - identificação do grupo).



2. Noções Básicas

É importante saber um pouco da história da ARPANET para entender o Sistema de Nomes de Domínio (ou Domain Name System). O DNS foi desenvolvido para resolver problemas em particular na ARPANET, e a Internet - uma descendente da ARPANET - que continua sendo sua principal usuária.

2.1 Uma Breve História da Internet

No final da década de 1960, a Agência de Projetos de Pesquisas Avançadas do Departamento de Defesa dos Estados Unidos da América - ARPA (Department of Defense's Advanced Research Projects Agency) mais tarde chamada de DARPA - começou a consolidar uma rede experimental de computadores de longa distância, chamada de ARPANET, que espalhou-se pelos Estados Unidos. O objetivo original da ARPANET era permitir aos fornecedores do governo compartilhar caros e também escassos recursos computacionais. Desde o início, entretanto, usuários da ARPANET também usavam a rede para colaboração. Essa colaboração abrangia desde compartilhamento de arquivos e programas e troca de mensagens correio eletrônico (e-mail) até desenvolvimento conjunto e pesquisas usando computadores remotos compartilhados.

O conjunto de protocolos TCP/IP (Transmission Control Protocol / Internet Protocol) foi desenvolvido no início da década de 1980, e rapidamente tornou-se o protocolo padrão de rede na ARPANET. A inclusão do conjunto de protocolos sobre o popular sistema operacional BSD UNIX de Berkeley na Universidade da Califórnia foi instrumento de democratização entre as redes. O BSD UNIX era gratuito para universidades. Isto significava que conectar-se a rede ficou repentinamente disponível a um baixo custo para muito mais organizações do que já estavam conectadas a ARPANET. Muitos dos computadores que estavam sendo conectados a ARPANET, estavam também conectados a redes locais também, e muito brevemente os outros computadores das redes locais estavam se comunicando via a ARPANET através deles. A rede cresceu de um punhado de computadores para uma rede de dezenas de milhares de computadores. A ARPANET original tornou-se o backbone (espinha dorsal) de uma confederação de redes locais e regionais baseados em TCP/IP, chamada de Internet.

Em 1988, entretanto, o DARPA decidiu que o experimento estava terminado. O Departamento de Defesa começou a dismantlar a ARPANET. Uma outra rede, fundada pela Fundação Nacional de Ciência (National Science Foundation) e chamada de NSFNET, substituiu a ARPANET como backbone da Internet.

Mesmo mais recentemente, no primeiro semestre de 1995, a Internet sofreu uma transição do uso da NSFNET como backbone para usar múltiplos backbones comerciais, correndo sobre linhas de longa distância como MCI e Sprint, e antigas redes comerciais como PSINet e Altnet.

Hoje, a Internet conecta milhões de computadores em todo o mundo. De fato, uma significativa proporção dos computadores tirando-se os PC's do mundo estão conectados a Internet. Os novos backbones comerciais pode carregar, com folga, um volume de 45 megabits por segundo, quase mil vezes mais a largura de banda da ARPANET original. Dezenas de milhões de pessoas usam a Internet para comunicação e colaboração diariamente.

2.2 Sobre A Internet e as internets

Uma explicação sobre "A Internet" e as "internets" em geral, torna-se necessária. A diferença impressa entre as duas é bastante sutil: a primeira letra maiúscula. A distinção entre seus significados, entretanto, é significativa. A Internet, com o "I" maiúsculo, refere-se a rede que começou sua vida como a ARPANET e continua hoje como, grosseiramente falando, a confederação de todas as redes TCP/IP interligadas, direta ou indiretamente, a backbones comerciais norte-americanos. XXXX?, xxx - backbones TCP/IP comerciais, redes TCP/IP regionais, redes TCP/IP de corporações e do governo dos Estados Unidos, e redes TCP/IP em outros países - interconectados por circuitos digitais de alta velocidade.

A internet com inicial minúscula, por outro lado, é simplesmente qualquer rede feita por múltiplas redes menores usando o mesmo protocolo de comunicação. Uma internet não precisa obrigatoriamente estar conectada a Internet, nem necessita usar o TCP/IP como protocolo de comunicação. Existem internets isoladas de corporações, e existem internets baseadas em XNS da Xerox e internets baseadas em DECnet.

O novo termo "intranet" é somente mais um termo de marketing para uma internet baseada em TCP/IP, usado para enfatizar o uso de tecnologias desenvolvidas e introduzidas na Internet, dentro de uma rede corporativa interna de uma empresa.

2.3 A História do DNS

Durante a década de 1970, a ARPANET era uma comunidade pequena, amigável de alguns poucos hosts. Um simples arquivo, HOSTS.TXT, continha todas as informações necessárias sobre estes computadores: um mapeamento nome-para-endereço para cada host conectado à ARPANET. A conhecida e familiar tabela de hosts do UNIX, /etc/hosts, era simplesmente copiado do HOSTS.TXT (tirando-se as informações/campos que o UNIX não usava).

O arquivo era mantido pelo Centro de Informações da Rede do SRI (conhecido como NIC - Network Information Center), e distribuído a partir de uma única máquina, SRI-NIC. Os administradores da ARPANET repetidamente enviavam e-mail's com suas mudanças para o NIC, e periodicamente buscavam HOSTS.TXT, via ftp de SRI-NIC. Todas as mudanças eram compiladas em um novo HOSTS.TXT, uma ou duas vezes por semana. Como a ARPANET cresceu, todavia, este método tornou-se impraticável. O tamanho do HOSTS.TXT cresceu em proporção ao crescimento do número de computadores na ARPANET. Além disso, o tráfego gerado pelo processo de atualização cresceu cada vez mais: cada host adicional significava mais do que uma linha a mais em HOSTS.TXT, mas também mais uma atualização em potencial.

Então, quando a ARPANET mudou para os protocolos TCP/IP, a população da rede explodiu. E neste momento surgiram uma série de problemas com HOSTS.TXT:

- Tráfego e sobrecarga
 - O uso da máquina SRI-NIC, em termos de tráfego de rede e uso de CPU estavam tornando-se insuportáveis.
- Colisões de nomes
 - Não podem existir mais de uma máquina com o mesmo nome em HOSTS.TXT. Entretanto, enquanto o NIC conseguia garantir a atribuição de endereços de forma única, ele não tinha autoridade sobre os nomes dos hosts. Não havia nada para prevenir alguém de adicionar mais um computador com um mesmo nome já existente e estragar todo o esquema. Se alguém incluísse mais uma máquina com o mesmo nome de um grande servidor de mail's, por exemplo, poderia romper o serviço de mail de boa parte da ARPANET.
- Consistência
 - Manter a consistência do arquivo através de uma rede em expansão tornava-se cada vez mais difícil. Enquanto um novo HOSTS.TXT alcançava as mais longínquas praias da crescida ARPANET, uma máquina havia trocado de endereço, ou um novo host surgido, mudanças das quais os usuários precisavam.

O problema essencial era que o mecanismo do HOSTS.TXT não expandia-se bem. Ironicamente, o sucesso da ARPANET como experimento conduziu ao fracasso e obsolescência do HOSTS.TXT.

Os controladores da ARPANET iniciaram uma procura por um sucessor para o HOSTS.TXT. Seu objetivo era criar um sistema que resolvesse os problemas inerentes a um sistema de tabela de hosts unificado. O novo sistema deveria permitir administração local dos dados, mas permitindo acesso global. A descentralização da administração deveria eliminar o gargalo de máquinas únicas e aliviar o problema do tráfego. Conseqüentemente, um gerenciamento local iria facilitar a tarefa de manter os dados sempre atualizados. E deveria usar um contexto de nomes hierárquicos para os nomes das máquinas. Isto deveria garantir nomes únicos.

Foi Paul Mockapetris, do Instituto de Ciência da Informação da USC - Estados Unidos, o responsável por desenhar a arquitetura do novo sistema. Em 1984, ele lançou as RFCs 882 e 883, que descreviam o "Domain Name System" ou DNS. Estas RFCs foram atualizadas, mais tarde, pelas RFCs 1034 e 1035, as atuais especificações vigentes. As RFCs 1034 e 1035 agora foram acrescidas pelas RFCs 1535, 1536 e 1537, que descrevem potenciais problemas de segurança do DNS, problemas de implementação, e truques de administração, respectivamente.

2.4 Uma visão geral do DNS

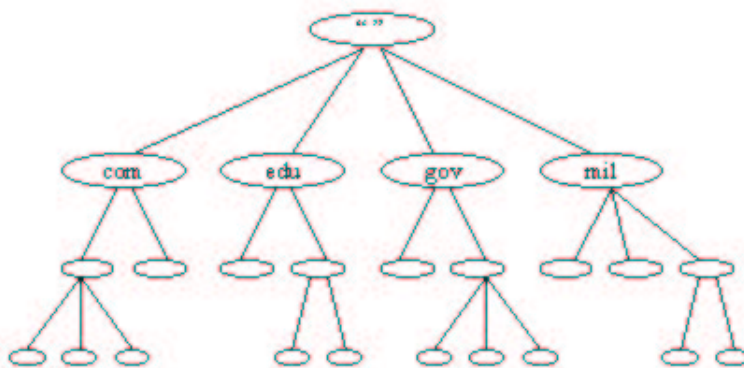
O Domain Name System é uma base de dados distribuída. Isto permite o controle local de segmentos da base completa. Todavia dados em cada segmento são disponíveis através de toda a rede por intermédio de um modelo cliente-servidor. Robustez e performance adequada são conseguidas através de um esquema de replicação e cache.

Programas chamados de servidores de nomes constituem a metade servidora do modelo cliente-servidor do DNS. Servidores de Nomes (ou name servers), contém informações sobre alguns segmentos da base de dados, e a disponibilizam para os clientes, chamados de resolvedores (ou resolvers). Os resolvers são, muitas vezes, somente rotinas de bibliotecas de funções programação, que buscam informações em servidores de nomes através da rede.

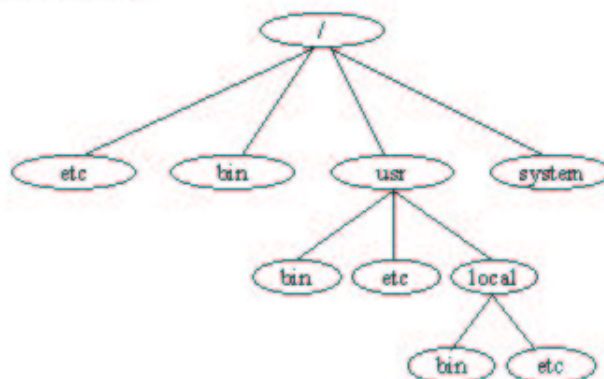
A estrutura da base de dados do DNS, mostrada na figura abaixo, é muito parecida com a do sistema de arquivos do UNIX (ou do DOS, se preferir). A base como um todo (e o sistema de arquivos) é mostrada como uma árvore invertida, com a raiz (ou root) no topo. No UNIX, a raiz é denotada por uma barra ("/"). No DNS, o nome da raiz é um nome nulo (''), mas é escrito como um ponto ("."), para simplificar.

Cada nodo da árvore representa um parte de toda o banco de dados - um diretório no sistema de arquivos UNIX, ou um domínio no Domain Name System. Cada domínio ou diretório pode ser sempre dividido em outras partes, chamadas de sub-domínios no DNS, assim como os sub-diretórios do sistema de arquivos. Sub-domínios, como sub-diretórios, são desenhados como filhos dos seus nodos pais.

Base de dados DNS

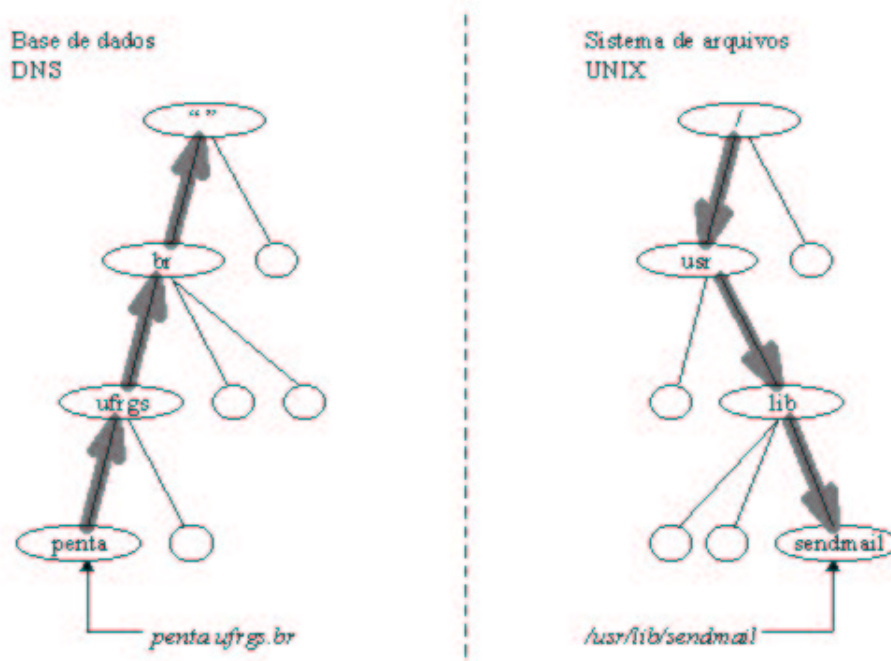


Sistema de arquivos UNIX



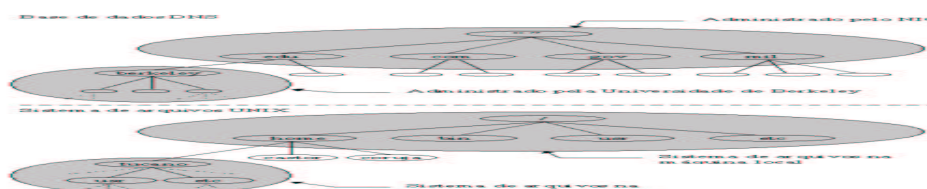
Cada domínio tem um nome, assim como cada diretório. Esse nome, é o que o identifica em relação ao seu domínio pai. Isso é equivalente ao caminho relativo de

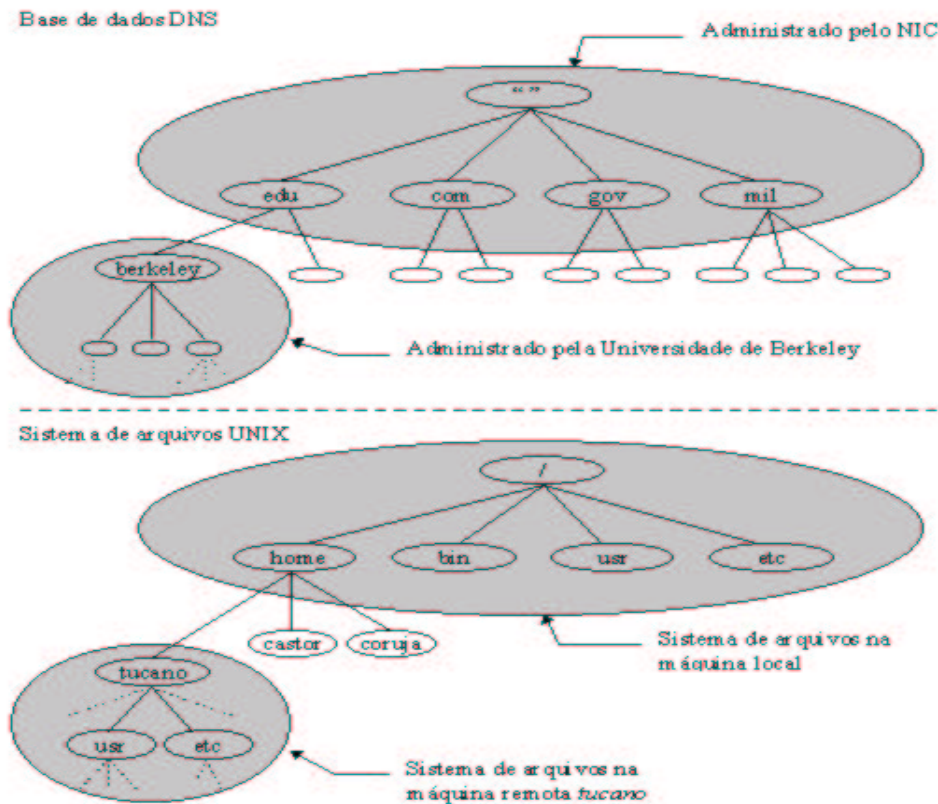
um diretório. E assim como um caminho absoluto no sistema de arquivos, um nome de domínio completo (ou domain name) é o que identifica a sua posição na árvore. No DNS um nome de domínio completo é a seqüência de todos os nomes desde o domínio raiz, separados por ".". No sistema de arquivos do UNIX, um caminho absoluto é a lista de todos os nomes relativos lidos desde a raiz até a folha (na direção inversa a do DNS, como mostra a figura a seguir), usando uma barra para separar os nomes.



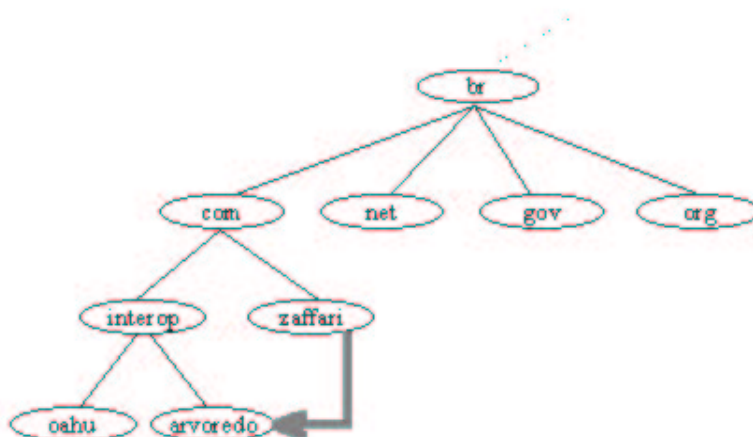
No DNS, cada domínio pode ser administrado por uma organização diferente. Cada organização pode então quebrar o seu domínio em vários sub-domínios e distribuir a responsabilidade para esses sub-domínios para outras organizações. Por exemplo, o NIC controla o domínio "edu" (educational), mas atribui a autoridade sobre o domínio "berkeley.edu" para a Universidade de Berkeley (figura 2.3). Isto também é semelhante a montar um sistema de arquivos com NFS (Network File System): certos diretórios podem ser, na verdade, diretórios de outros hosts, montados remotamente via NFS. O administrador do host "tucano", por exemplo (figura 2.3, parte inferior), é responsável pelo sistema de arquivos que aparece na máquina local como "/home/tucano".

Em um sistema de arquivos, diretórios contém arquivos e também podem conter sub-diretórios. Da mesma forma, domínios podem conter tanto máquinas como outros domínios: seus sub-domínios. Nomes de domínio são usados como índices para a base de dados DNS. Você pode pensar em um dado como associado a um nome de domínio.





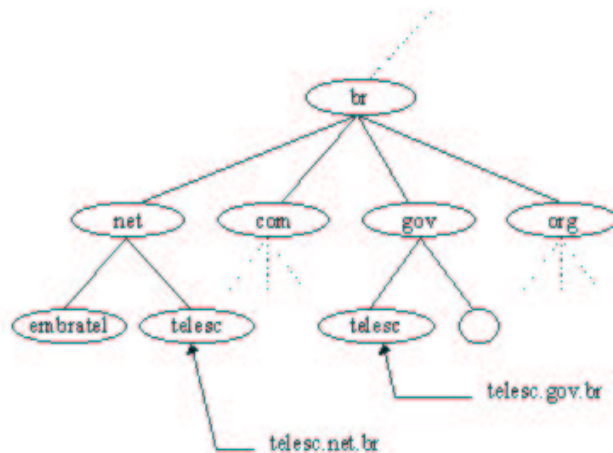
Cada máquina em uma rede tem um domínio, o qual aponta informação sobre esta máquina (veja figura 2.4). Esta informação pode incluir endereços IP, informações sobre roteamento de mail, etc. Os computadores podem também ter um ou mais "alias" (apelidos) de nome de domínio, que são simplesmente ponteiros de um nome de domínio para outro (o nome de domínio oficial, ou "canonical"). Na figura, "zaffari.com.br" é um "alias" para o nome oficial "arvoredo.interop.com.br".



Por que toda essa complicada estrutura? A razão é resolver os problemas que HOSTS.TXT tinha. Por exemplo, fazendo os nomes hierárquicos eliminam-se as armadilhas das colisões de nomes.

Os domínios têm nomes de domínio únicos, portanto as organizações são livres para

escolher dentro de seus domínios. Qualquer nome que eles escolham, ele não conflitará com outros nomes de domínios, uma vez que ele possui seu domínio único no final do nome. Podem existir dois sub-domínios com nomes iguais (como mostrado na figura 2.5), já que eles têm diferentes domínios pais.



2.5 A História do BIND

A primeira implementação do DNS chamava-se JEEVES, escrita pelo próprio Paul Mockapetris. Uma implementação posterior era o BIND, escrita para o sistema operacional BSD UNIX 4.3 por Kevin Dunlap. O BIND atualmente é mantido por Paul Vixie sob o patrocínio do ISC (Internet Software Consortium).

BIND, que é a sigla para Berkeley Internet Name Domain, é uma implementação bem concentrada do padrão DNS. É de longe a implementação de DNS mais usada e popular que existe, e quase se confunde com a definição.

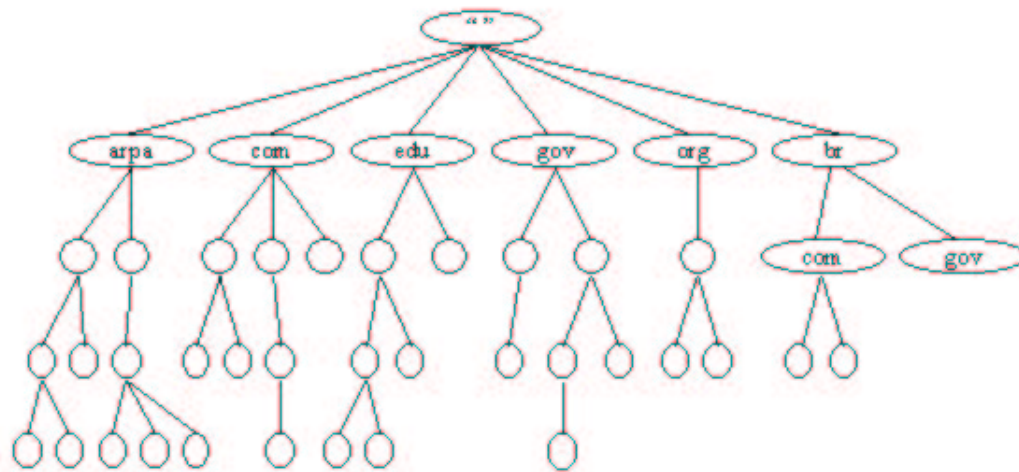


3 Funcionamento

No capítulo anterior já foram explicados alguns aspectos importantes do DNS, incluindo a arquitetura cliente-servidor, e a estrutura da base de dados DNS. Entretanto, ainda não foram dados muitos detalhes.

Neste capítulo serão esclarecidos os mecanismos que fazem o DNS funcionar. Também serão introduzidos alguns termos técnicos sobre DNS, que facilitam certas explicações.

3.1 O Espaço de Nomes de Domínio



Cada unidade de dado na base distribuída do DNS é indexada pelo nome. Esses nomes são essencialmente apenas caminhos em uma enorme árvore invertida, chamada de espaço de nome de domínio (em inglês, "Domain Name Space"). Como já foi mencionado no capítulo anterior, a estrutura hierárquica da árvore, mostrada na figura 3.1, é muito parecida com a estrutura do sistema de arquivos UNIX. A árvore tem uma raiz simples no topo. No UNIX, ela é chamada de diretório raiz (ou root), e representada pela barra ("/"). Em DNS, ela é simplesmente chamada de "a raiz", ou "o domínio raiz". Como um sistema de arquivos, a árvore de DNS pode ter quantos caminhos se necessitar em cada interseção, chamada nodo. A profundidade da árvore é limitada em 127 níveis, que é um limite muito difícil de se alcançar.

3.1.1 Nomes de Domínio

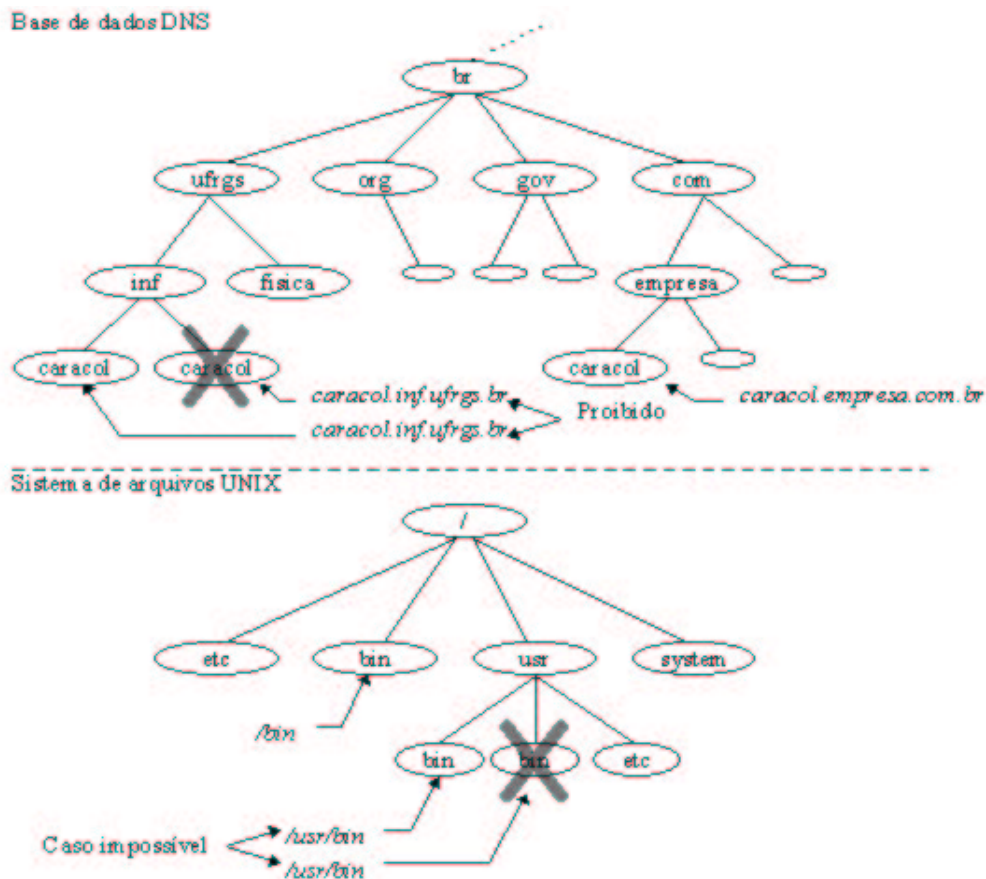
Cada nodo na árvore é chamado por um nome simples (sem pontos). Este nome pode ter até 63 caracteres de comprimento. O domínio raiz tem um nome nulo (sem comprimento, com tamanho zero), o qual é reservado. O nome de domínio completo de qualquer nodo na árvore é a seqüência dos nomes simples em todo o seu caminho até a raiz. Nomes de domínios são sempre lidos a partir do nodo em direção à raiz (árvore acima), e com pontos simples separando os nomes no caminho.

Se o domínio raiz realmente aparece no nome de domínio de um nodo, o nome parece

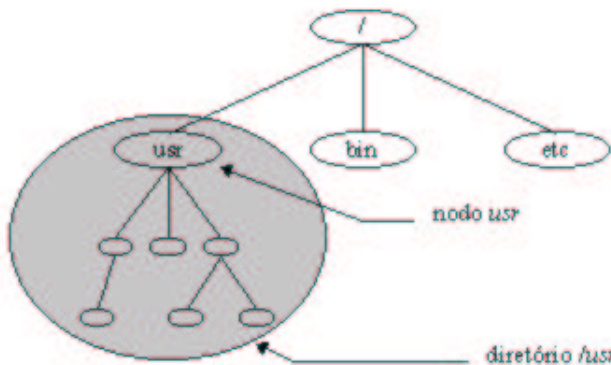
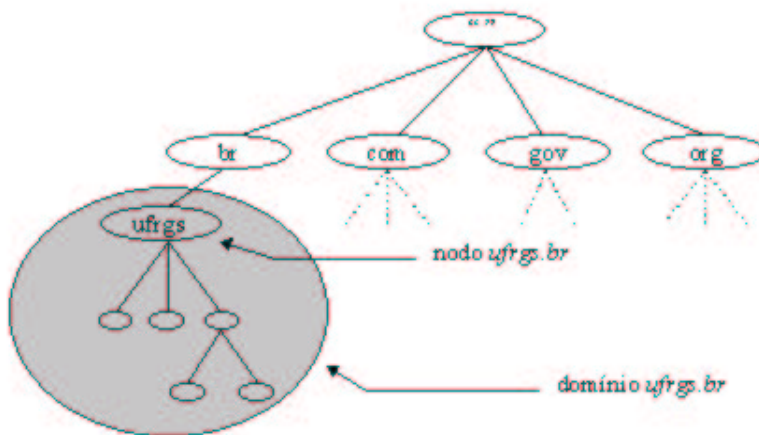
como se estivesse terminado em um ponto (na verdade termina com um ponto, o separador e um nome nulo). Quando um domínio raiz aparece ele é escrito com um ponto simples, por maior conveniência. Conseqüentemente alguns programas interpretam um ponto final em um nome de domínio para indicar que o nome de domínio é absoluto. O nome de domínio absoluto é escrito relativamente a raiz. Isto especifica a localização de um nodos sem ambigüidade na hierarquia. Um nome de domínio absoluto é também referenciado como um nome de domínio totalmente qualificado, as vezes abreviado como FQDN, do inglês Fully Qualified Domain Name. Nome sem ponto no fim são as vezes interpretados como nome de domínios relativos, assim como nome de diretórios sem um barra na frente são interpretados como diretórios ou caminhos relativos.

O DNS precisa que nodos irmãos, ou seja, nodos filhos do mesmo pai, tenham nomes únicos entre eles. Esta restrição garante que um nome de domínio seja identificado unicamente com um único nodo na árvore. A restrição, realmente, não é uma limitação desde que nomes precisam ser únicos somente entre os irmãos ou filhos de um mesmo pai, e não entre todos os nodos da árvore. A mesma restrição aplica-se ao sistema de arquivos UNIX, você não pode ter dois nodos irmãos com um mesmo nome. Assim como não podem existir dois diretórios `/usr/bin`, também não podem existir dois nodos `caracol.inf.ufrgs.br` no espaço de nomes (figura 3.2). Entretanto, podem existir um nodo `caracol.inf.ufrgs.br` e outro `caracol.empresa.com.br`, e da mesma forma, existir um diretório `/bin` e outro `/usr/bin`.

3.1.2 Domínios

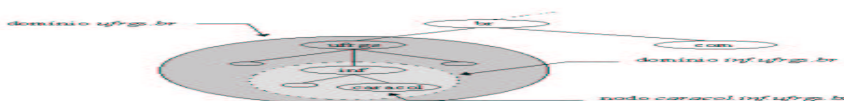


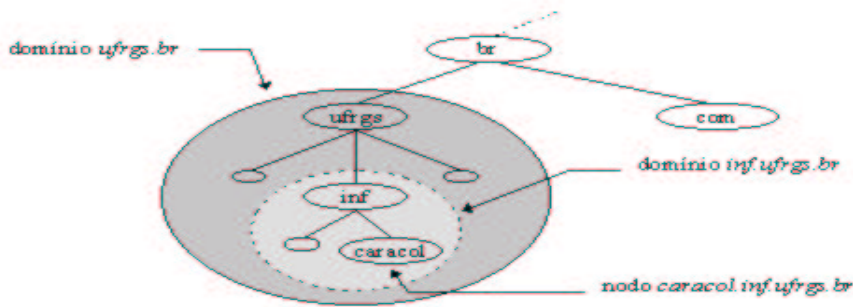
Um domínio é simplesmente uma sub-árvore de um espaço de nome de domínio. O nome de um domínio é o mesmo nome de domínio do nodo raiz da sub-árvore. Isto somente significa que o nome de um domínio é o nome do nodo do topo do domínio. Entã o, por exemplo, o topo do domínio "ufrgs.br" é o nodo chamado "ufrgs.br", como mostrado na figura 3.3.



Da mesma forma, em um sistema de arquivos, no topo do diretório "/usr" você encontra um nodo chamado "/usr", como mostra a figura 3.4.

Cada nome de domínio na sub-árvore é considerado uma parte de um domínio. Partindo da idéia de que um nome de domínio pode estar em muitas sub-árvores, então conclui-se que um nome de domínio pode estar, também, em muitos domínios. Por exemplo, o nome de domínio *caracol.inf.ufrgs.br* é parte do domínio *inf.ufrgs.br*, e também parte do domínio *ufrgs.br*, como mostra a figura 3.5.



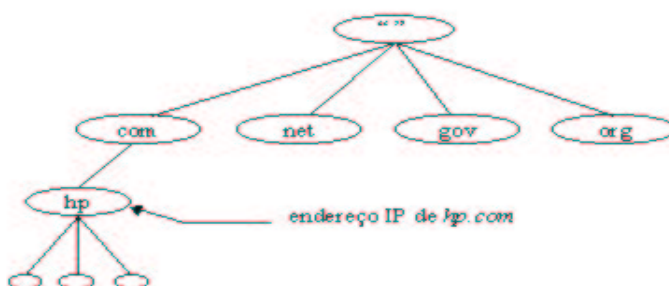


Então, resumindo-se, um domínio é somente uma sub-árvore de um espaço de nomes de domínio. Entretanto, se um domínio é formado somente de nomes de domínio e outros domínios, onde estão todos os hosts? Bem, domínios são grupos de máquinas.

As máquinas estão lá, mas elas são domínios também. E lembre-se, nomes de domínio são somente índices dentro da base de dados do DNS. Os "hosts" no DNS são os nomes de domínio que apontam para informações sobre computadores individuais. E um domínio contém todas as máquinas as quais seus nomes de domínio estão dentro deste domínio. Os computadores são agrupados de forma lógica, frequentemente por regiões ou por organizações, e não necessariamente por rede, ou endereço ou tipo de hardware. Você pode ter dez máquinas diferentes, cada uma delas em uma rede diferente, e talvez cada uma delas em um país diferente, todos no mesmo domínio.

Os domínios que são as folhas da árvore geralmente representam máquinas individuais. Seus nomes de domínios podem apontar para endereços de rede, informações de hardware e informações de roteamento de mail. Nomes de domínio no interior da árvore podem ser um computador e podem apontar para informações estruturais sobre os filhos do domínio, ou seja, seus sub-domínios. Nomes de domínio interiores não estão restritos a um ou ao outro caso. Podem representar tanto o domínio que eles representam ou uma máquina particular na rede. Por exemplo, "hp.com" é o nome do domínio da companhia Hewlett-Packard e também o nome de domínio do computador que distribui o mail entre a HP e a Internet.

O tipo de informação (estrutural ou sobre uma máquina) devolvido em uma pergunta ao servidor dns, depende do contexto do qual o nome de domínio está sendo usado. Em uma procura pelos filhos de um nó pode retornar dados estruturais, ao passo que se fizer um telnet para o nome de domínio, a informação retornada é sobre o host ao qual se deseja conectar (por exemplo, o endereço IP de "hp.com", mostrado na figura 3.6).



O espaço de nomes de domínio da Internet existente, entretanto, tem alguma estrutura imposta para ele mesmo. Especialmente nos domínios de níveis superiores, os nomes de domínio seguem certas tradições (e não o regras, uma vez que eles podem e têm quebrado). São essas tradições que ajudam os nomes de domínios não parecerem tão caóticos. Entender elas é um grande passo se você está tentando decifrar um nome de domínio.

3.2.1 Domínios *Top-level* (primeiro nível)

Os domínios de primeiro nível originais dividiram o espaço de nomes de domínio da Internet por tipos de organizações. Como inicialmente a Internet estava restrita aos Estados Unidos (seus fundadores), esta divisão era praticada para as organizações norte-americanas. Aqui estão os sete principais domínios *top-level*:

- **com** - Organizações comerciais, como Hewlett-Packard (*hp.com*), Sun Microsystems (*sun.com*), e IBM (*ibm.com*)
- **edu** - Organizações educacionais, como as universidades norte-americanas de Berkeley (*berkeley.edu*) e Purdue (*purdue.edu*)
- **gov** - Organizações governamentais, como a NASA (*nasa.gov*) e a Fundação Nacional de Ciência dos Estados Unidos (*nsf.gov*)
- **mil** - Organizações militares, como o exército (*army.mil*) e a marinha (*navy.mil*) dos Estados Unidos
- **net** - Organizações da grande rede, que de alguma forma administram ou colaboram com a Internet, como a NSFNET (*nfs.net*)
- **org** - Organizações não-governamentais e sem fins lucrativos, como a Fundação da Fronteira da Eletrônica (*eff.org*)
- **int** - Organizações internacionais, como a OTAN (*nato.int*)

Existe ainda um domínio de primeiro nível chamado *arpa*, que originalmente era usado durante a transição das tabelas de hosts (o arquivo HOSTS.TXT) para o DNS, na ARPANET. Todos as máquinas tinham os seus nomes originalmente sob o domínio *arpa*, onde eram facilmente encontrados. Depois, eles foram movidos para vários sub-domínios dos domínios organizacionais de primeiro nível.

Pode-se notar um certo preconceito nacionalista nos exemplos acima: todos citados são organizações primariamente norte-americanas. E isto é fácil de entender, quando se recorda que a Internet começou com a ARPANET, um projeto de pesquisa fundado pelos Estados Unidos. Ninguém poderia prever o sucesso da ARPANET, ou que ela poderia tornar-se um dia, uma rede internacional como atualmente é a Internet.

Para acomodar a internacionalização da Internet, os mentores do seu espaço de nomes entraram num acordo fazendo concessões. Em vez de insistir que todos os domínios de primeiro nível descrevessem afiliações organizacionais, eles decidiram permitir também, designações geográficas. Os novos domínios *top-level* ficaram reservados (mas não necessariamente criados) para corresponder individualmente a cada país. Seus nomes de domínio seguiram um padrão internacional já existente, a norma ISO 3166. Esta norma estabelecia abreviaturas oficiais, de duas letras, para cada país no mundo. Esta lista está incluída no Apêndice A deste trabalho.

3.3 Aprofundando-se mais

Dentro desses domínios de primeiro nível, as tradições e algumas extensões suas que são seguidas variam. Alguns dos domínios de primeiro nível da ISO 3166 seguem o esquema original organizacional de perto. Por exemplo, o domínio do Brasil, *br*, tem sub-domínios *com.br*, para domínios comerciais, *gov.br* para organizações governamentais, etc. A única exceção no Brasil fica para as universidades, primeiras entidades brasileiras a entrar na Internet, que estão diretamente sob o domínio *br*, por exemplo *ufrgs.br*. Outro exemplo; o domínio da Austrália, *au*, segue o padrão original, com sub-domínios como *edu.au* e *com.au*.

Alguns outros domínios *top-level* da ISO 3166 seguem o mesmo caminho do Reino Unido, *uk*, e têm sub-domínios como *co.uk* para corporações, e *ac.uk* para a comunidade acadêmica. Entretanto, na maioria dos casos, mesmo esses domínios geográficos estão subdivididos de maneira organizacional.

Mas isso não é verdade para o domínio *us* dos Estados Unidos. O domínio *us* tem cinquenta sub-domínios que correspondem aos cinquenta estados norte-americanos. Cada um recebe o seu nome de domínio de acordo com o padrão de abreviatura de duas letras para o estado (a mesma abreviatura padronizada pelo serviço postal dos Estados Unidos). Dentro do domínio de cada estado, a organização ainda é geográfica: a maioria dos sub-domínios correspondem a cidades individualmente. Abaixo das cidades, então, os sub-domínios correspondem aos computadores.

3.3.1 Lendo Nomes de Domínio

Agora que já se sabe o que representam a maioria dos domínios de primeiro nível e como seus espaços de nomes são estruturados, será muito mais fácil entender os nomes de domínio. Por exemplo:

```
caracol.inf.ufrgs.br
```

Para entender-se esse nome de domínio deve-se lembrar que *ufrgs.br* é o domínio da Universidade Federal do Rio Grande do Sul (UFRGS), universidade localizada no Brasil (domínio *br*). Caso não se recorde disso, pode-se tentar adivinhar que o domínio pertence a uma organização brasileira, e das primeiras a ligar-se na Internet (provavelmente uma universidade), pois não tem nenhum outro domínio que especifique o tipo de organização. O sub-domínio *inf* indica o departamento de informática, e finalmente, *caracol* é o nome do host ligado à rede.

arvoredos.interop.com.br

Bem, neste exemplo podemos verificar que este nome de domínio pertence a uma organização comercial, uma empresa brasileira chamada InterOp. Descobrimos isto simplesmente vendo-se os seus domínios pais, *com.br*, que sugerem ser uma entidade comercial (sub-domínio *com*) brasileira (domínio *br*). E finalmente *arvoredos* é um host dentro do domínio da empresa acima nominada.

beethoven.telesc.gov.br

Neste caso acima, pode-se deduzir que provavelmente este nome de domínio seja de uma empresa do governo brasileiro (*gov.br*), a TELESC, empresa de telecomunicações do estado de Santa Catarina. E mais uma vez o nome mais abaixo deste domínio especifica um computador.

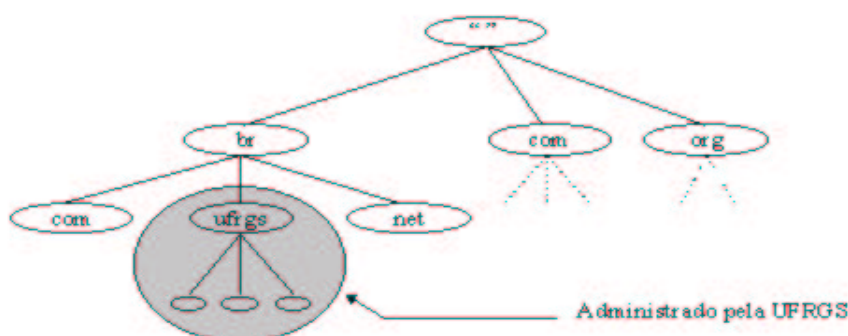
fernwood.mpk.ca.us

Aqui precisa-se entender o domínio *us*. O sub-domínio *ca.us* pertence ao estado da Califórnia, Estados Unidos, mas *mpk* dificilmente poderia-se descobrir que é o domínio de Menlo Park, sem conhecer-se a geografia da Baía de São Francisco.

3.4 A Distribuição de Domínios

É sempre bom lembrar que um dos principais objetivos no planejamento do DNS foi a administração descentralizada. E isto é alcançado delegando-se. Delegar-se domínios assemelha-se bastante com delegar-se tarefas. Um gerente pode dividir um grande projeto em tarefas menores e delegar a responsabilidade de cada uma das tarefas para diferentes empregados.

Desta forma, uma organização administrando um domínio pode dividi-lo em sub-domínios. Cada um desses sub-domínios pode ser delegado, inclusive, a outras organizações. Isto significa que a organização para a qual é delegada um domínio fica responsável por manter toda a informação naquele sub-domínio. Ela pode livremente mudar os dados, e pode até dividir seu sub-domínio em outros e delegá-los a outras organizações. O domínio pai contém somente ponteiros para as fontes de informações do sub-domínio, para que possa encaminhar as perguntas a elas. Por exemplo, o domínio *ufrgs.br* é delegado ao pessoal da UFRGS que gerencia a rede da universidade (figura 3.7).





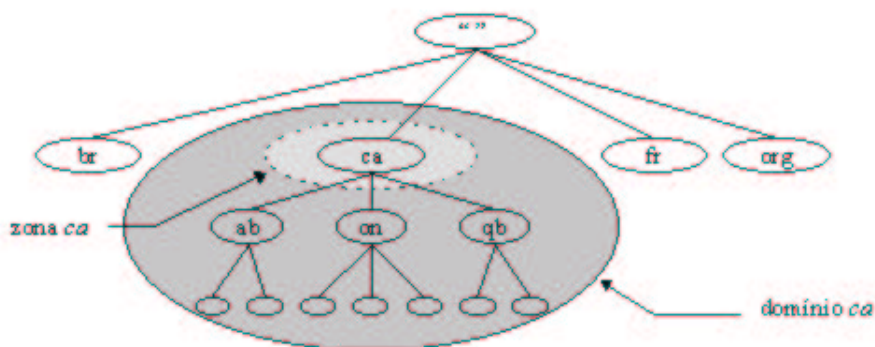
Nem todas as organizações delegam seus domínios inteiros, assim como nem todos gerentes delegam todo o seu trabalho. Em um domínio podem haver vários sub-domínios, e também, conter máquinas que não estão em nenhum sub-domínio. Em um exemplo fictício (retirado de um livro), a Acme Corporation, que possui uma divisão em Rockaway e seus escritórios centrais em Kalamazoo, pode ter os sub-domínios *rockaway.acme.com* e *kalamazoo.acme.com*. Entretanto, alguns poucos hosts nos escritórios de vendas da Acme, espalhados pelos Estados Unidos, ficariam melhor diretamente abaixo de *acme.com*.

Desta forma, é importante que entenda-se o conceito do termo delegar, que se refere a atribuir a responsabilidade de um sub-domínio a uma outra organização.

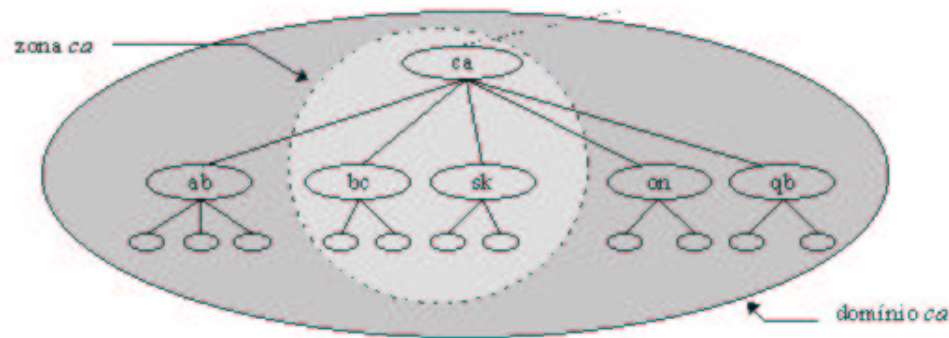
3.5 Servidores de Nomes

Os programas que armazenam as informações sobre o espaço de nomes de domínio são chamados de *servidores de nomes* (em inglês, *name server*). Servidores de nomes geralmente têm informações completas sobre alguma parte do espaço de nomes de domínio, chamada de *zona*. Servidores de nomes podem ter autoridade por mais de uma zona, também.

A diferença entre uma zona e um domínio é tênue. Uma zona contém os nomes de domínio e os dados que um domínio contém, exceto pelos nomes de domínio e as informações que são delegados para outras zonas. Por exemplo, o domínio de primeiro nível *ca* (de Canadá) pode ter os sub-domínios *ab.ca*, *on.ca*, e *qb.ca*, para as províncias de Alberta, Ontario e Quebec, respectivamente. A autoridade para os domínios *ab.ca*, *on.ca* e *qb.ca* pode ser delegada para organizações em cada uma das províncias. O domínio *ca* contém todos os dados em *ca* mais todos os dados em *ab.ca*, *on.ca* e *qb.ca*. Mas a zona *ca* somente contém os dados em *ca* (figura 3.8).



Entretanto, se um sub-domínio não é delegado para outrem, a zona do domínio pai contém os nomes de domínio e dados do sub-domínio. Então, os sub-domínios *bc.ca* e *sk.ca* (British Columbia e Saskatchewan) do domínio *ca* podem existir, mas não serão delegados a ninguém. Neste caso, a zona *ca* contém dados de mais de um nível, contendo *bc.ca* e *sk.ca*, mas não outros sub-domínios de *ca*, como mostra a figura 3.9.



Então, por que servidores de nomes carregam zonas em vez de domínios? Porque um domínio pode conter mais informações que um servidor de nomes pode precisar: ele poderia conter dados delegados a outros servidores de nomes, o que é desnecessário.

No caso de um domínio recém criado, ele provavelmente não terá nenhum sub-domínio. Desta forma, como não há nada delegado a ninguém, o domínio e a sua zona conterão os mesmos dados.

3.5.1 Delegando Domínios

O ato de delegar, resumindo-se, é a atribuição da responsabilidade por alguma parte do seu domínio para outra organização. O que realmente acontece, então, é a atribuição da autoridade sobre seus sub-domínios para diferentes servidores de nomes.

Os seus dados, em vez de conterem informações sobre os sub-domínios que foram delegados, incluem ponteiros para os servidores de nomes que têm autoridade sobre os seus sub-domínios. Desta forma, se um dos seus servidores de nomes é perguntado sobre dados de algum de seus sub-domínios delegados, ele pode responder com uma lista dos servidores de nomes com os quais se pode conseguir maiores informações.

3.5.2 Tipos de Servidores de Nomes

As especificações do DNS definem dois tipos de servidores de nomes: *primary master* (controlador primário) e *secondary master* (controlador secundário). Um servidor de nomes primário pega os dados para as zonas que tem autoridade a partir de arquivos na máquina em que está rodando. Um servidor de nomes secundário pega os dados da sua zona de outro servidor de nomes que tenha autoridade para a zona. Quando um servidor secundário começa a rodar, ele contacta o servidor de nomes do qual ele deve pegar as informações e, se necessário, busca os dados da zona. Isto é chamado de *zone transfer* (transferência na zona).

O DNS provê estes dois tipos de servidores de nomes para facilitar a administração. Uma vez criados os dados para uma zona, e configurado um servidor de nomes primário, não é mais necessário perder-se tempo copiando-se os dados de máquina para máquina a fim de criarem-se mais servidores de nomes para a zona. Simplesmente configuram-se os servidores de nomes secundários para carregarem os dados a partir do servidor primário. Uma vez configurados, os secundários irão periodicamente contactar o primário para verificarem se estão atualizados, e se

necessário, buscarem novamente os dados.

Isto é importante porque é uma grande ideia ter-se mais de um servidor de nomes para qualquer zona. As vantagens são a redundância dos dados, a distribuição da carga da rede, e para ter-se certeza que todas as máquinas tem um servidor de nomes por perto. Usar servidores de nomes secundários torna a administração mais prática.

Chamar um servidor de nomes de servidor primário ou secundário é, entretanto, um pouco complicado. Um servidor de nomes pode ter autoridade sobre mais de uma zona. Da mesma forma, um servidor de nomes pode ser um controlador primário para uma zona, e controlador secundário para outra. Todavia, a maioria dos servidores de nomes são, ou primários para a maioria das zonas que carregam, ou secundários para a maioria de suas zonas. Então, chamando-se um servidor de nomes em particular de primário ou secundário, significa que ele é controlador primário ou secundário para a maioria das zonas que ele carrega.

3.5.3 Arquivos de Dados

Os arquivos de onde os servidores de nomes primários carregam os dados de suas zonas são chamados, simplesmente, de arquivos de dados ou arquivos das zonas. Geralmente referem-se a eles como *db files*, abreviatura em inglês para arquivos da base de dados. Os servidores de nomes secundários algumas vezes também carregam os dados da zona a partir arquivos locais. Frequentemente os servidores secundários são configurados para armazenar cópias de segurança, em arquivos, dos dados que eles buscaram em um servidor de nomes primário. Se a execução de um servidor secundário é terminada e reiniciada, ele irá, primeiramente, ler as suas cópias de segurança, e então conferir se elas estão atualizadas. Isto elimina a necessidade de transferir todos os dados da zona no caso de eles não terem mudado, e provê um fonte de dados no caso do servidor primário estar parado.

Os arquivos de dados contém registros de recursos que descrevem a zona. Os registros de recursos descrevem todas as máquinas na zona, e marcam qualquer sub-domínios que estejam delegados. O BIND também permite diretivas especiais para incluir os conteúdos de outros arquivos em arquivo de dados, muitos parecidos com a declaração `#include` da programação em linguagem C.

3.6 Os resolvers

Os *resolvers* (ou resolvidores) são os clientes que acessam os servidores de nomes. Os programas rodando em um computador que necessita informações do espaço de nomes de domínio usam o *resolver*. O *resolver* trata de:

- perguntar ao servidor de nomes,
- interpretar respostas (que podem ser registros de recursos ou erros), e
- retornar a informação pedida ao programa que a requisitou.

No BIND, o *resolver* é somente um conjunto de funções de uma biblioteca que ficam compiladas dentro de programas como o *telnet* e o *ftp*. Eles não são nem processos

separados. Eles são o bastante espertos e suficientes para enviar a pergunta, aguardar a resposta, e reenviá-la caso não seja respondida. Mas é só isso. A maior parte do trabalho de achar uma resposta para uma pesquisa fica com o servidor de nomes. As especificações do DNS chamam esse tipo usado no BIND de *stub resolver* (resolvedor curto, ou burro).

Outras implementações do DNS têm *resolvers* mais inteligentes, que podem fazer tarefas mais sofisticadas, como construir uma *cache* das informações já buscadas dos servidores de nomes. Contudo, estas implementações não são comuns como o *stub resolver* do BIND.

3.7 A Resolução

Os servidores de nomes são profundos conhecedores da arte de buscar informações do espaço de nomes de domínio. Eles devem ser, dada a limitada inteligência de alguns resolvedores. Eles podem, não somente, responder dados sobre zonas que eles têm autoridade, mas também podem pesquisar o espaço de nomes de domínio para achar dados que eles não têm autoridade. Este processo é chamado de resolução de nomes (em inglês, *name resolution* ou simplesmente *resolution*).

Por causa da estrutura do espaço de nomes ser uma árvore invertida, um servidor de nomes somente precisa de uma peça de informação para encontrar o caminho até qualquer ponto da árvore: os nomes e endereços dos servidores de nomes da raiz. Um servidor de nomes pode enviar uma pergunta para um servidor da raiz, pedindo dados sobre qualquer nome no espaço de nomes de domínio, e o servidor raiz irá indicar a quem perguntou sobre o caminho que deve começar a procurar.

3.7.1 Servidores de Nomes da Raiz

Os servidores de nomes da raiz sabem onde estão os servidores com autoridade sobre todos os domínios de primeiro nível (*top-level*). Recebida uma pergunta sobre qualquer nome de domínio, os servidores de nomes da raiz podem pelo menos responder os nomes e endereços dos servidores com autoridade sobre o domínio de primeiro nível, ao qual o nome de domínio pertence. E estes servidores de primeiro nível podem prover a lista dos servidores de nomes com autoridade sobre o domínio de segundo nível, no qual o nome de domínio está dentro. Cada servidor de nomes pesquisado responde com a informação sobre como chegar-se mais "perto" da resposta procurada, ou provê a própria resposta.

Os servidores de nomes da raiz são um elo chave na resolução de nomes. Desta forma, sendo tão importantes, o DNS provê mecanismos (como cache, que serão discutidos mais adiante) para ajudar a descarregar os servidores de nomes da raiz. Porém, na ausência de outra informação, a resolução deve começar nos servidores da raiz. Isto torna os servidores de nomes da raiz cruciais para a operação do DNS; se todos os servidores de nomes raiz da Internet tornarem-se inalcançáveis por um período longo, toda resolução de nomes na Internet irá falhar. Por este motivo, a Internet tem nove servidores de nomes da raiz (quando este trabalho foi terminado), espalhados por diferentes partes da rede. Alguns estão na MILNET, a porção militar

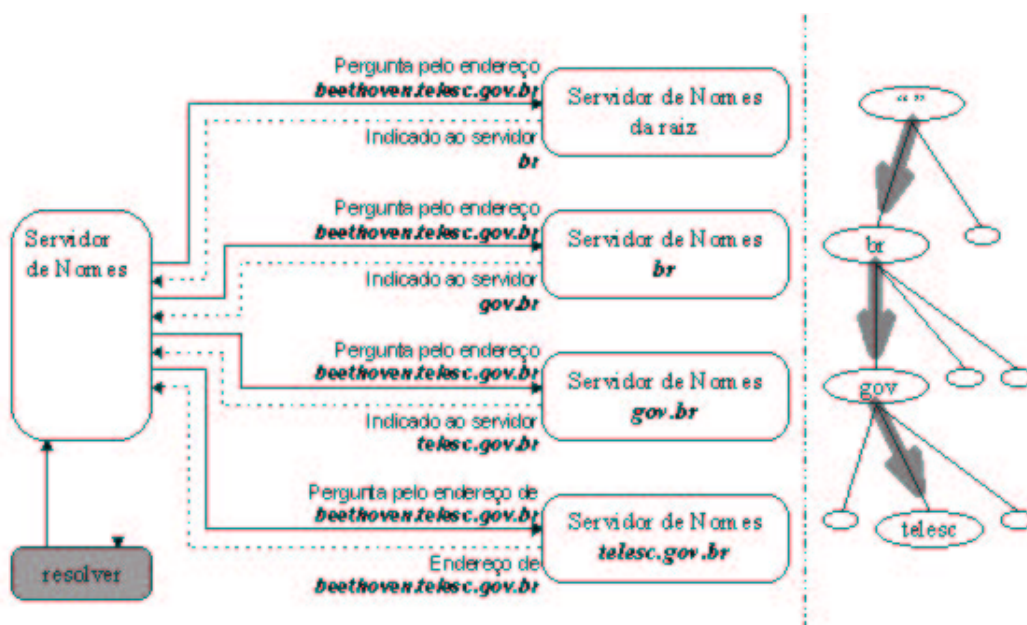
dos Estados Unidos da Internet; outro na SPAN, internet da NASA; um está na Europa; e um é mantido por um provedor de acesso comercial à Internet.

Sendo alvo de tantas pesquisas deixam os servidores raiz muito ocupados; mesmo com muitos servidores, o tráfego para cada servidor de nomes da raiz é muito alto. Um estudo feito em 1992 pela U.S.C. dos Estados Unidos concluiu que o seu servidor da raiz recebia aproximadamente 20.000 pacotes com perguntas em uma hora, ou praticamente 6 perguntas por segundo. Estatísticas mais recentes feitas pelo InterNIC mostram que o seu servidor de nomes da raiz, *ns.internet.net*, recebe 255.600 perguntas por hora, ou quase 71 perguntas por segundo.

Apesar da alta carga dos servidores da raiz, a resolução de nomes na Internet funciona muito bem. A figura 3.10 mostra o processo de resolução para o endereço de uma máquina real em um domínio real, incluindo como o processo atravessa a árvore do espaço de nomes de domínio.

O servidor de nomes local pergunta a um servidor da raiz pelo endereço de *beethoven.telesc.gov.br* e recebe indicação para os servidores de nomes de *br*. Ele faz a mesma pergunta a um servidor de nomes de *br*, e é indicado para os servidores de nomes de *gov.br*. Os servidores de nomes de *gov.br* referenciam o servidor de nomes do domínio *telesc.gov.br*. Finalmente, o servidor de nomes local pergunta ao servidor de *telesc.gov.br* pelo endereço, e então, recebe a resposta desejada.

3.7 2 Recursão



Pode-se notar uma grande diferença na quantidade de trabalho feito pelos servidores de nomes, neste último exemplo. Quatro dos servidores simplesmente retornaram a melhor resposta que eles tinham (a maioria indicava outros servidores de nomes) para as perguntas recebidas. Eles não precisaram enviar suas próprias perguntas para achar os dados requisitados. Contudo um servidor de nomes, aquele pesquisado pelo *resolver*, teve que seguir sucessivas indicações até receber a resposta final.

E por que ele não poderia simplesmente indicar ao resolvidor outro servidor? Porque um *stub resolver* não tem a inteligência necessária para seguir um referência. E como um servidor de nomes sabe que não deve responder com uma indicação? Basta o *resolver* enviar uma pergunta recursiva.

As perguntas podem ser de dois tipos: *recursivas* e *iterativas* (ou *não-recursivas*). Perguntas recursivas localizam a maior parte do trabalho de resolução em só um servidor de nomes. *Recursão*, ou *resolução recursiva*, é somente um nome para o processo de resolução usado por um servidor de nomes quando ele recebe perguntas recursivas.

Interação ou *resolução interativa*, de outro lado, refere-se ao processo de resolução usado por um servidor de nomes quando recebe perguntas interativas.

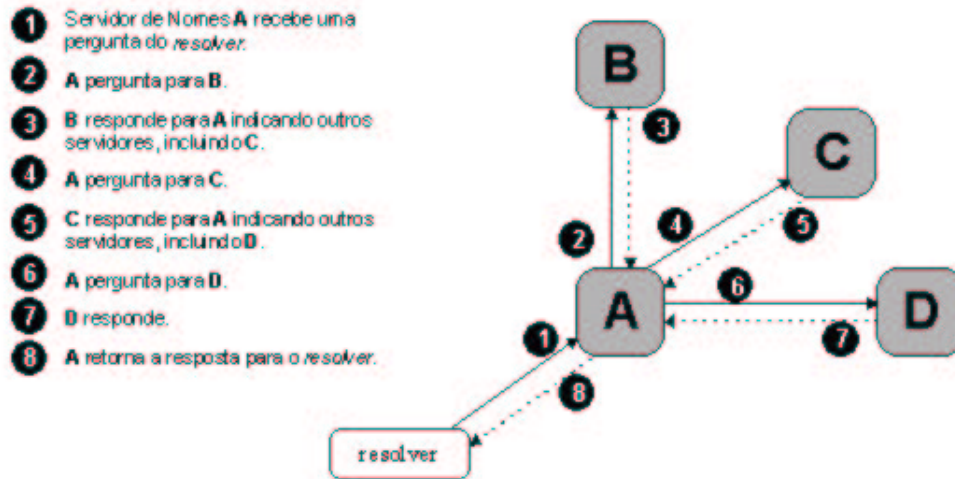
Na recursão, um *resolver* envia uma pergunta a um servidor de nomes por informações sobre um nome de domínio em particular. O servidor de nomes perguntado é então forçado a responder com o dado requisitado, ou com um erro especificando que o dado do tipo pedido não existe ou que o nome de domínio especificado não existe. O servidor de nomes não pode somente indicar a quem perguntou para um servidor de nomes diferente, porque a pergunta era recursiva.

Se o servidor de nomes pesquisado não tiver autoridade sobre os dados requisitados, ele terá que perguntar a outro servidor para encontrar a resposta. Ele pode enviar perguntas recursivas aos outros servidores de nomes, desta forma obrigando-os a encontrar uma resposta e devolve-la (passando o problema adiante). Ou então enviar perguntas interativas, e possivelmente ser indicado para outros servidores de nomes que estejam "mais perto" do nome de domínio que se está procurando. As implementações atuais são geralmente mais "educadas" e fazem o último caso, seguindo as indicações até que uma resposta seja encontrada.

3.7.3 Interação

A resolução interativa, de outra forma, não exige muito trabalho da parte do servidor de nomes perguntado. Na resolução interativa, o servidor de nomes simplesmente fornece a melhor resposta que *ele já sabe* a quem perguntou-lhe. Não é necessário que ele saia perguntando mais. O servidor perguntado consulta seus dados locais (incluindo a sua cache, que será explicado mais adiante), procurando pela informação requisitada. Se ele não encontra o dado ali, ele faz a melhor tentativa para dar a quem perguntou uma informação que auxilie a continuar no processo de resolução. Geralmente essas são os nomes e endereços dos servidores de nomes "mais perto" do dado que se está procurando.

O que isto importa para o processo de resolução pode-se observar por completo na figura 3.11.



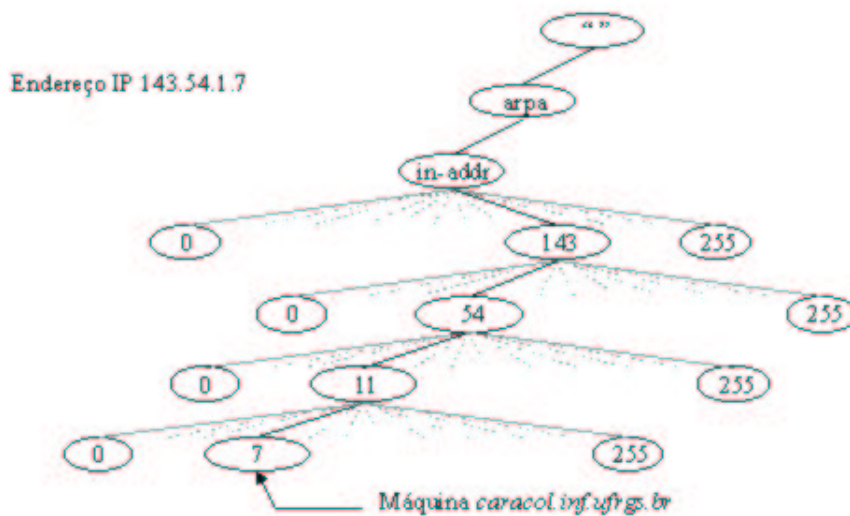
Um resolver pergunta a um servidor de nomes local, que então pergunta a vários outros servidores de nomes no intuito de responder ao resolver. Cada servidor que o servidor local pergunta referencia a outro servidor de nomes com autoridade sobre uma zona cada vez mais abaixo no espaço de nomes e mais perto da resposta final. Finalmente o servidor de nomes local pergunta ao servidor com autoridade sobre o nome de domínio perguntado, que devolve uma resposta.

3.7.4 Mapeando Endereços para Nomes

Uma parte importante da funcionalidade do processo de resolução de nomes ainda está faltando: como endereços são mapeados de volta para nomes. O mapeamento endereço para nome é usado para produzir uma saída mais fácil para os humanos lerem e interpretarem (em arquivos de log, por exemplo). Também é usado em algumas conferências de autorizações. Máquinas UNIX mapeiam endereços para nomes de máquinas com o intuito de comparar com as suas entradas dos arquivos *.rhosts* e *hosts.equiv*, por exemplo. Quando se usa tabelas de hosts (*host tables*), o mapeamento endereço para nome é trivial. Executa-se uma busca seqüencial pelo endereço, procurando-se em todo o arquivo de hosts. A procura retorna o nome oficial da máquina listado. No DNS, entretanto, o mapeamento endereço para nome não é tão simples. Os dados, incluindo endereços, no sistema de nomes de domínio são indexados por nome. Encontrar um endereço dado um nome de domínio é relativamente fácil. Todavia, encontrar um nome de domínio que corresponda a um dado endereço iria requerer uma busca exaustiva em todos os nomes de domínio da árvore.

Na verdade, existe uma solução melhor que é ao mesmo tempo simples e efetiva. Considerando-se que é fácil encontrar dados quando têm-se o nome, que serve de índice, por que não criar uma parte do espaço de nomes de domínio que use endereços como nomes? No espaço de nomes de domínio da Internet, esta porção do espaço de nomes é o domínio *in-addr.arpa*.

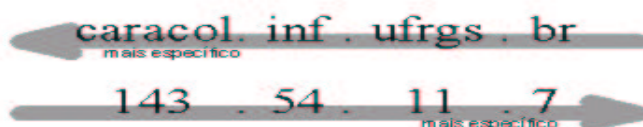
Os nodos no domínio *in-addr.arpa* são nomeados através dos números na representação *dotted-octet* (octetos-pontuados) de endereços IP. (A representação *dotted-octet* é o método mais comum de expressar os endereços IP de 32 bits como números de 0 a 255, separados por pontos.) O domínio *in-addr.arpa*, por exemplo, pode ter até 256 sub-domínios, cada um correspondendo a um possível valor no primeiro octeto de um endereço IP. Cada um destes sub-domínios pode ter mais 256 sub-domínios para si, correspondendo aos possíveis valores do segundo byte. Finalmente, no quarto nível abaixo, existirão os registros de recursos associados ao octeto final dando o nome de domínio completo da máquina ou rede daquele endereço IP. Isto torna *in-addr.arpa* um domínio incrivelmente grande, mostrado em parte na figura 3.12, e suficiente para cada endereço IP na Internet.

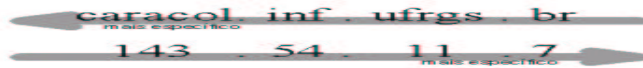


Pode-se notar que quando se lê como um nome de domínio, o endereço IP aparece invertido, ou seja, de trás para frente, de forma que o nome é lido da folha até a raiz da árvore. Por exemplo, se *caracol.inf.ufrgs.br* tem o endereço IP *143.54.11.7*, o sub-domínio *in-addr.arpa* correspondente é *7.11.54.143.in-addr.arpa*, que mapeia de volta para o nome de domínio *caracol.inf.ufrgs.br*.

Os endereços IP poderiam ser representados da maneira oposta no espaço de nomes, com o primeiro octeto do endereço IP mais abaixo do domínio *in-addr.arpa*. Desta forma, o endereço IP seria lido corretamente no nome de domínio.

Entretanto, endereços IP são hierárquicos, assim como nomes de domínio. Números de rede são distribuídos assim como nomes de domínio, e os administradores podem dividir os seus endereços de rede e delegar as sub-redes. A diferença é que endereços IP ficam mais específicos da esquerda para a direita, enquanto nomes de domínio ficam menos específicos da esquerda para a direita. A figura 3.13 exemplifica melhor isto.





Fazendo-se os primeiros bytes no endereço IP aparecerem mais acima na árvore possibilita aos administradores a habilidade de delegar a autoridade para os sub-domínios *in-addr.arpa* juntamente com os seus endereços de rede. Por exemplo, o domínio *54.143.in-addr.arpa*, que contém as informações de mapeamento reverso para todos os hosts que possuem endereços IP que comecem com 143.54, pode ser delegado para os administradores da rede 143.54.0.0. Isto seria impossível se os octetos ficassem na ordem inversa. Se os endereços IP fossem representados da outra forma, o domínio *54.143.in-addr.arpa* consistiria de todas as máquinas que tivessem um endereço IP terminado com 54.143 (não um domínio prático de delegar-se).

3.7.5 Buscas invertidas

O espaço de nomes *in-addr.arpa*, entretanto, somente pode ser usado para o mapeamento endereço IP a nome de domínio. A procura por um nome de domínio que é indexado por um tipo de dado arbitrário (algo além de um endereço), no espaço de nomes de domínio requeriria um outro espaço de nomes especializado como o *in-addr.arpa*, ou uma busca exaustiva.

Esta procura exaustiva é possível, até certo grau. Uma busca invertida (*inverse query*, em inglês) é uma procura por um nome de domínio que é indexado por um determinado dado. Ela é processada somente pelo servidor de nomes que recebeu a pergunta. Este servidor de nomes procura em todos os seus dados locais pelo item procurado e retorna o nome de domínio correspondente, se possível. Caso ele não encontre o dado, ele desiste. Nenhuma tentativa é feita para repassar a busca para outro servidor de nomes.

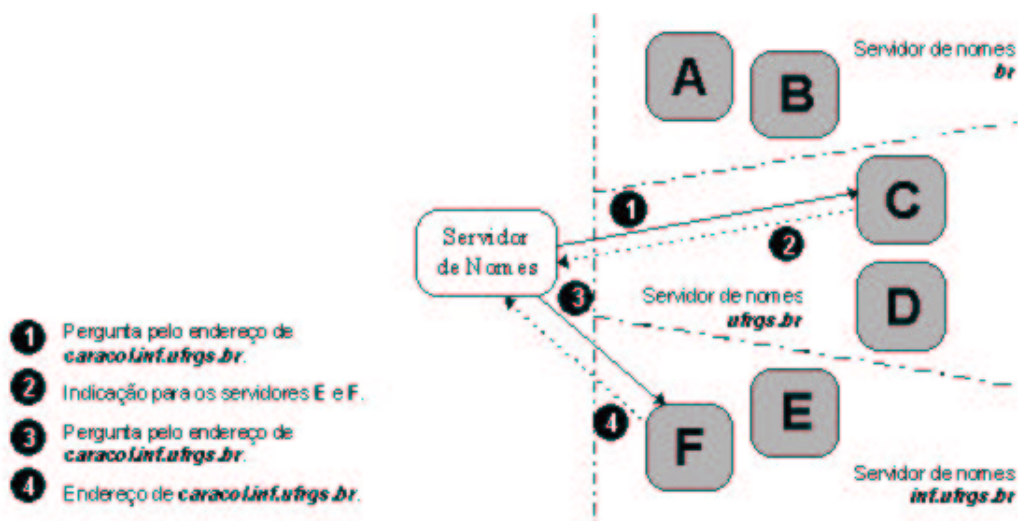
Uma vez que qualquer servidor de nomes somente possui dados de parte de todo o espaço de nomes de domínio, uma busca invertida nunca tem garantia de retornar uma resposta. Por exemplo, se um servidor de nomes recebe uma busca invertida por um endereço IP que ele não sabe nada a respeito, ele não pode retornar uma resposta, mas também não sabe se o endereço IP existe ou não, uma vez que somente possui parte da base de dados do DNS. E ainda tem mais, a implementação de busca invertidas é opcional dependendo da especificação de DNS. O BIND 4.9.4 ainda possui o código que implementa buscas invertidas, embora esteja comentado.

3.8 A Cache

Todo o processo de resolução parece horrivelmente confuso e enfadonho para quem está acostumado a simples buscas através da tabela de hosts. Realmente, o uso das *hosts tables* é usualmente um processo muito rápido. E uma das características que aumenta consideravelmente a velocidade é manter-se uma *cache* (memória temporária).

Um servidor de nomes processando uma pesquisa recursiva poderá ter que enviar algumas outras perguntas para encontrar uma resposta. Contudo, ele descobre bastante

informação sobre o espaço de nomes de domínio enquanto procura. Cada vez que ele recebe uma outra lista de servidores de nomes, ele aprende que aqueles servidores têm autoridade sobre determinada zona, e aprende os endereços desses servidores. E, ao final do processo de resolução, quando ele finalmente encontra o dado procurado originalmente, pode armazenar o dado para uso futuro. Com a versão 4.9.3 do BIND, os servidores de nomes também implementam *negative caching*: se um servidor de nomes com autoridade responde a uma pergunta com uma resposta, afirmando que o tipo de informação procurada não existe, para o nome de domínio especificado, o servidor de nomes local irá armazenar temporariamente na *cache* esta informação. Os servidores de nomes guardam na *cache* todos esses dados para ajudar a agilizar pesquisas sucessivas. Assim, a próxima vez que um *resolver* perguntar ao servidor de nomes por um dado, sobre um nome de domínio que o servidor já saiba algo, o processo é encurtado um pouco. O servidor de nomes poderia ter a resposta na *cache*, positiva ou negativamente. Neste caso ele simplesmente retorna a resposta para o resolvidor. Mesmo que ele não tenha a resposta na *cache*, ele poderia já ter "aprendido" as identidades dos servidores de nomes com autoridade sobre a zona na qual está o nome de domínio, e estar pronto para perguntá-los diretamente.



Por exemplo, caso um servidor de nomes (*ns1*) já tenha procurado pelo endereço de *penta.ufrgs.br*. No processo de procura, ele armazenaria na *cache* os nomes e endereços dos servidores de nomes de *penta.ufrgs.br* e *ufrgs.br* (mais o endereço IP de *penta.ufrgs.br*). Agora, se um resolver perguntasse ao servidor *ns1* pelo endereço de *caracol.inf.ufrgs.br*, o *ns1* iria pular a pergunta aos servidores de nomes da raiz. Reconhecendo que *ufrgs.br* é o antecessor mais perto de *caracol.inf.ufrgs.br* que ele conhece, o servidor de nomes *ns1* iria começar a pesquisar no servidor de nomes de *ufrgs.br*, como mostra a figura 3.14. De outra forma, se o servidor de nomes *ns1* tivesse encontrado que não existia endereço para *caracol.inf.ufrgs.br*, a próxima vez que ele fosse perguntado pelo endereço, ele poderia simplesmente responder apropriadamente com a informação da sua *cache*.

Além de aumentar a velocidade da resolução de nomes, o uso de cache evita mais buscas aos servidores de nomes da raiz. Isto significa que os servidores não são mais tão dependentes da raiz, e os servidores da raiz não irão sofrer muito com as perguntas de outros servidores.

3.8.1 O Time to Live

Os servidores de nomes não podem guardar os dados na *cache* para sempre, é claro, e por isso é também chamada de memória temporária. Se os servidores a guardassem para sempre, as mudanças dos dados nos servidores de dados com autoridade não iriam nunca alcançar o resto da rede. Os servidores de nomes remotos iriam continuar somente a usar os dados da *cache*. Por este motivo, o administrador da zona que contém os dados decide um *time to live* (tempo de vida), ou TTL, para os dados. O *time to live* é a quantidade de tempo que qualquer servidor de nomes está permitido a armazenar na *cache* os dados. Depois de expirado o tempo de vida, o servidor de nomes deve descartar os dados da *cache* e buscar novos dados dos servidores de nomes com autoridade sobre a zona. Isto também se aplica aos dados armazenados negativamente na *cache*; o servidor de nomes deve desconsiderar uma resposta negativa após um período, também, para o caso de dados novos terem sido adicionados nos servidores de nomes com autoridade sobre a determinada zona. Todavia, o tempo de vida para dados negativos da *cache* não pode ser selecionado pelo administrador do domínio; é fixo em dez minutos.

Os servidores de nomes não podem guardar os dados na *cache* para sempre, é claro, e por isso é também chamada de memória temporária. Se os servidores a guardassem para sempre, as mudanças dos dados nos servidores de dados com autoridade não iriam nunca alcançar o resto da rede. Os servidores de nomes remotos iriam continuar somente a usar os dados da *cache*. Por este motivo, o administrador da zona que contém os dados decide um *time to live* (tempo de vida), ou TTL, para os dados. O *time to live* é a quantidade de tempo que qualquer servidor de nomes está permitido a armazenar na *cache* os dados. Depois de expirado o tempo de vida, o servidor de nomes deve descartar os dados da *cache* e buscar novos dados dos servidores de nomes com autoridade sobre a zona. Isto também se aplica aos dados armazenados negativamente na *cache*; o servidor de nomes deve desconsiderar uma resposta negativa após um período, também, para o caso de dados novos terem sido adicionados nos servidores de nomes com autoridade sobre a determinada zona. Todavia, o tempo de vida para dados negativos da *cache* não pode ser selecionado pelo administrador do domínio; é fixo em dez minutos.

Decidir um *time to live* para os seus dados é essencialmente decidir uma relação entre performance e consistência. Um TTL pequeno irá ajudar a assegurar que os dados sobre o seu domínio fiquem consistentes através da rede, por que os servidores de nomes remotos irão descartar os seus dados mais rapidamente, e serão forçados a perguntar aos seus servidores de nomes com autoridade mais frequentemente por novos dados. De outro modo, isto irá aumentar a carga de trabalho nos seus servidores e, conseqüentemente aumentar o tempo médio de resolução por informações no seu domínio.

Um TTL grande irá encurtar o tempo médio que leva para resolver-se informações no seu domínio, uma vez que poderá ser armazenada por um longo tempo. Em contrapartida, as informações sobre o seu domínio ficarão inconsistentes por um período maior caso sejam feitas alterações nos dados dos seus servidores de nomes.

Deste modo, o melhor é encontrar uma relação ótima entre esses dois itens, baseando-se na carga média dos seus servidores e na frequência das alterações no seu domínio, resultando em um *time to live* ideal.



4. Parâmetros do DNS

O DNS inclui vários tipos de parâmetros. Estes são documentados, na sua maior parte, nas especificações RFC1034 e RFC1035. Os parâmetros adicionais para as diversas classes são definidos em outras RFCs conforme indicado nas tabelas.

Para cada nome armazenado no banco de dados do DNS é atribuído um tipo de objeto para identificá-lo. Por exemplo, o nome é para um host, caixa de correio ou usuário? Por exemplo, como você determina se *inf* em *inf.ufrgs.br* é o nome de um host ou um sub-domínio? Quando um cliente pede para o DNS converter um nome (isto é, para executar o mapeamento de nome para endereço), ele deve especificar o tipo de resposta desejado. Por exemplo, um aplicativo telnet remoto deve especificar que deseja o endereço IP de uma máquina, e não o endereço do servidor de correio eletrônico do domínio. Em outras palavras, a pergunta deve ser bastante específica.

Um servidor de nomes pode trabalhar em dois modos: recursiva ou interativamente, com já foi mencionado no capítulo anterior, dependendo da solicitação do cliente. Quando um cliente conversor de nomes consulta um servidor de nomes, a mensagem contém as informações seguintes:

- O nome a ser convertido.
- A classe do nome (o grupo de protocolo a ser usado).
- O tipo de resposta desejado (o endereço IP, caixa de correio, etc. associado a esse nome).
- Um "código de operação" que especifica se o servidor de nome deve traduzir o nome completamente.

1. Classes

O DNS foi criado para ser independente de protocolo. Por isso, o campo classe na consulta identifica o grupo de protocolo do registro de interesse. É possível que haja vários registros no banco de dados DNS que tenham os mesmos dados no campo "nome" mas para protocolos diferentes. Para os usuários do TCP/IP, o código de classe é IN de Internet.

Decimal	Nome	Referências
0	Reservado	Internet Assigned Numbers Authority, Dezembro 1994
1	Internet (IN)	RFC1035
2	Não associado	Internet Assigned Numbers Authority, Dezembro 1994

3	Chaos (CH)	RFC1035
4	Hesiod (HS)	RFC1035
5-253	Não o associado	Internet Assigned Numbers Authority, Dezembro 1994
254	Nenhum	Paul Vixie, Junho 1997
255	Qualquer [somente QCLASS]	RFC1035
256-65534	Não o associado	Internet Assigned Numbers Authority, Dezembro 1994
65535	Reservado	Internet Assigned Numbers Authority, Dezembro 1994

Tabela 4.1 Classes do DNS

4.2 Tipos de Registros de Recursos

Você leu anteriormente que o DNS pode ser usado para traduzir um nome de host para um endereço IP e também pesquisar um endereço de servidor de correio eletrônico para um domínio. Essa informação é diferenciada pelos tipos de objetos. Na tabela 4.1, estão listados os tipos de RR (registros de recursos) mais comuns usados no DNS.

A	Endereço IP do host
CNAME	Nome de domínio canônico para um alias (nome alternativo)
HINFO	Informações de CPU e sistema operacional para um host
MX	Nome do servidor de correio eletrônico para o domínio
NS	Servidor de nome (que tem autoridade) para o domínio
PTR	Ponteiro para o nome do domínio (como uma ligação simbólica nos sistemas de arquivo)
SOA	Start of Authority - indica a autoridade sobre os dados do domínio

TXT	Informação textual
RP	Pessoa responsável

Tabela 4.2 Tipos mais comuns de registros de recursos

A maioria dos registros que você encontra em um banco de dados DNS é do tipo A, o que significa que eles consistem no nome de um host e seus endereços IP. O próximo tipo comum de registro é provavelmente MX. Ele contém nomes de hosts que atuam como recurso para troca de correspondência eletrônica (gateway) para um sub-domínio dado. Os registros MX habilitam você a enviar uma correspondência eletrônica para alguém em determinado sub-domínio sem precisar conhecer o nome do gateway de correio eletrônico. Por exemplo, você precisa endereçar a correspondência eletrônica para *fulano@inf.ufrgs.br* sem saber que o nome do servidor de correio eletrônico é *caracol*. Senão, você teria que endereçar sua correspondência eletrônica para *fulano@caracol.inf.ufrgs.br*.

Abaixo segue uma tabela com todos os tipos de dados e perguntas (TYPES e QTYPES) definidos para a classe Internet (IN).

Tipo	Valor	Significado	Referências
A	1	um endereço de host	RFC1035
NS	2	um servidor de nomes <i>authoritative</i>	RFC1035
MD	3	um destino de mail (Obsoleto à MX)	RFC1035
MF	4	um mail forwarder (Obsoleto à MX)	RFC1035
CNAME	5	o nome canônico para um alias	RFC1035
SOA	6	define o início de uma zona com autoridade	RFC1035
MB	7	um nome de domínio de uma caixa de correio (EXPERIMENTAL)	RFC1035
MG	8	um membro de um grupo de mail (EXPERIMENTAL)	RFC1035
MR	9	um nome de domínio	RFC1035

		renomeado de mail (EXPERIMENTAL)	
NULL	10	um RR nulo (EXPERIMENTAL)	RFC1035
WKS	11	descrição de um serviço bastante conhecido (Well Known Service)	RFC1035
PTR	12	um ponteiro para um nome de domínio	RFC1035
HINFO	13	informações sobre um host	RFC1035
MINFO	14	informações sobre um mailbox ou uma lista de mail	RFC1035
MX	15	um servidor de mail (mail exchange)	RFC1035
TXT	16	texto simples	RFC1035
RP	17	para Pessoa Responsável	RFC1183
AFSDB	18	para a localização de base de dados AFS	RFC1183
X25	19	para endereço X.25 PSDN	RFC1183
ISDN	20	para endereço ISDN	RFC1183
RT	21	para indicação de rota	RFC1183
NSAP	22	para endereço NSAP (estilo de registro A para NSAP)	RFC1706
NSAP-PTR	23	idem (estilo de PTR para NSAP)	RFC1706
SIG	24	para assinatura de segurança	RFC2065
KEY	25	para chave de segurança	RFC2065
PX	26	informação de mapeamento de mail X.400	RFC1664
GPOS	27	posição geográfica	RFC1712

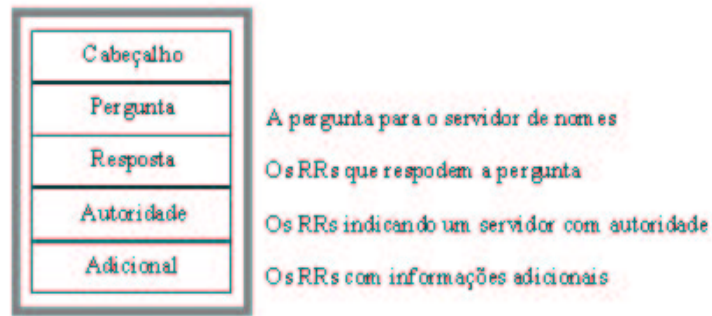
AAAA	28	endereço IP6	Susan Thomson, Agosto 1995
LOC	29	informação sobre localização	Paul Vixie, Junho 1997
NXT	30	próximo domínio	RFC2065
EID	31	identificador de fim	Michael Patton, Junho 1995
NIMLOC	32	localizador Nimrod	Michael Patton, Junho 1995
SRV	33	servidor selecionado	RFC2052
ATMA	34	endereço ATM	George Dobrowski, Julho 1996
NAPTR	35	ponteiro para autoridade de um nome	Ron Daniel, Junho 1997
TSIG	36	assinatura de transação	Paul Vixie, Junho 1997
IXFR	251	transferência incremental	RFC1995
AXFR	252	transferência de toda uma zona	RFC1035
MAILB	253	RRs relacionados a mailbox (MB, MG ou MR)	RFC1035
MAILA	254	RRs para agentes de mail (Obsoleto - veja MX)	RFC1035
*	255	Um pedido por todos os registros	RFC1035

Tabela 4.3 Tipos de dados e perguntas da classe Internet

3.

Formato das Mensagens

Toda a comunicação envolvendo o protocolo do DNS é carregado por um só formato de pacote chamado mensagem. O formato em um primeiro nível da mensagem é dividido em 5 seções (algumas delas ficam vazias em alguns casos), mostrado na figura 4.1.



cabeçalho está sempre presente. O cabeçalho inclui campos que especificam quais das seções restantes estão presentes na mensagem, e também indica se é uma pergunta ou uma resposta, uma pergunta padrão ou algum outro código de operação, etc.

Os nomes das seções depois do cabeçalho são derivados dos seus usos nas perguntas padrão. A seção pergunta contém campos que descrevem uma pergunta a um servidor de nomes. Esses campos são: QTYPE (tipo), QCLASS (classe) e QNAME (nome de domínio). As últimas três seções possuem o mesmo formato: uma lista de registros de recursos (RRs) concatenados, podendo ser vazia. A seção de resposta contém RRs que respondem a pergunta. A seção autoridade contém RRs que apontam para servidores de nomes com autoridade sobre o nome de domínio perguntado. A seção adicional contém RRs que se relacionam com a pergunta, mas não são respostas diretas para a pergunta.

5. Códigos de Operação

Quando um servidor de nomes receber uma consulta, ele verificará se o nome está dentro do sub-domínio para o qual tem autoridade. Nesse caso, ele pesquisará o nome no banco de dados e retornará a informação solicitada, se houver.

Se o servidor de nome não puder converter o nome completamente, ele verificará qual "código de operação" o cliente especificou. Se o cliente tiver solicitado uma conversão recursiva (pesquisa completa), o servidor de nome irá contactar outro servidor de nomes para ver se poderá convertê-lo; se esse servidor contactado não puder converter o nome, ele irá indicar o próximo servidor e assim por diante até que tenha sucesso ou até que o tempo limite de nome não encontrado seja decorrido.

Código	Nome	Significado	Referências
0	Query	Pergunta recursiva	RFC1035
1	IQuery	pergunta não-recursiva	RFC1035

2	Status	Estado	RFC1035
3	Reservado		Internet Assigned Numbers Authority, Dezembro 1994
4	Notify	Notificação	RFC1996
5	Update	Atualização	Paul Vixie, Junho 1997

Tabela 4.4 Códigos de Operação

Se o *resolver* pedir conversão interativa (pesquisa não recursiva), o servidor de nome irá gerar um erro se não puder converter o nome. Como parte da mensagem de resposta, o servidor de nome indicará outro servidor no qual o cliente deverá fazer a próxima tentativa.

6. Códigos de Resposta

Sempre que um cliente envia uma pergunta para um servidor de nomes, este responde seguindo um código para as respostas. Abaixo seguem os códigos de resposta e seus respectivos significados.

Código	Nome	Significado	Referências
0	NoError	Sem erro	RFC1035
1	FormErr	Erro de formato	RFC1035
2	ServFail	Falha no servidor	RFC1035
3	NXDomain	Domínio não existente	RFC1035
4	NotImp	Não implementado	RFC1035
5	Refused	Pergunta recusada	RFC1035
6	YXDomain	Nome existe, mas não deveria	RFC2136
7	YXRRSet	RR afirma existir, mas não deveria	RFC2136
8	NXRRSet	RR afirma que deveria existir, mas não existe	RFC2136

9	NotAuth	Servidor sem autoridade sobre a zona	RFC2136
10	NotZone	Nome não está contido na zona	RFC2136

Tabela 4.5 Códigos de resposta



5. Conclusão

Este trabalho visa esclarecer algumas questões na área de informática, principalmente no que diz respeito a utilização de uma gama enorme de informações, através das facilidades que hoje em dia nos é oferecido pela Internet.

Antes de mais nada, procuramos definir bem, neste trabalho, a sistemática do DNS, a resolução de nomes usados na grande rede global, seu funcionamento, características principais, seu uso e importância.

Com a realização deste trabalho, muitos conceitos estudados no decorrer do curso ficaram mais consolidados, evidenciando uma perfeita integração entre as diversas áreas no campo da Ciência da Computação.



Apêndice 1 - Páginas Úteis sobre DNS na Internet

- <http://www.dns.net/dnsrd>

DNS Resources Directory - Um site bastante completo, com muitas informações relevantes, e links para outras páginas sobre DNS.

- <http://rs.internic.net/help/domain/dns.html>

DNS Background Material - Excelente resumo informativo contendo um material inicial sobre DNS, sendo bastante conciso e completo. Mantido pelo InterNIC.

- <http://www.isc.org/isc/bind.html>

Site of Bind - O site oficial do BIND atualmente mantido pelo ISC. Possui as últimas versões do software e documentação em diversos formatos.

- <http://eeunix.ee.usm.maine.edu/guides/dns/dns.html>

DNS Guide - Um guia sobre DNS feito por Glenn Stevens.

- <http://web.syr.edu/~jmwobus/comfaqs/faq-dns>

DNS FAQ - Um documento com as perguntas mais frequentes sobre DNS.

- <http://www.verinet.com/dns>

Setting up DNS - Um guia bastante interessante para a configuração do BIND.

- <http://www.cg.org.br/docsoficiais/minuta.html>

Regras para obtenção de nomes de domínio no Brasil - Um documento do Comitê Gestor da Internet no Brasil, ditando as regras de nomes de domínio no Brasil.

Apêndice 2 - RFCs relacionados ao DNS

Muitos desses documentos são distribuídos como parte da distribuição do BIND.

RFC 819

The Domain Naming Convention for Internet User Applications de Z. Su e J.Postel

Documenta as idéias estruturais originais do DNS. Agosto 1982

RFC 920

Domain Requirements de J.Postel e J.Reynolds

Documento Administrativo sobre domínios. Outubro 1984

RFC 974

Mail Routing and the Domain System de Craig Partridge

Descreve o processamento dos registros MX (mail exchange). Janeiro 1986

RFC 1032

Domain Administrator's Guide de M.Stahl

Explica a função do administrador de domínio. Novembro 1987

RFC 1033 - atualizado pela RFC 1912

Domain Administrators Operations Guide de M. Lottor

Guia explicativo sobre a administração de DNS. Novembro 1987

RFC 1034 - atualizado pela RFC 1101

Domain Names - Concepts and Facilities de P. Mockapetris

Guia de referência, cobre um pouco de tudo. Novembro 1987

RFC 1035 - atualizado pela RFC 1706

Domain Names - Implementation and Specification de P. Mockapetris

Explica a mecânica do DNS (protocolo, tipos de dados, etc.) Novembro 1987

RFC 1101 - atualiza a RFC 1034

DNS Encoding of Network Names and Other Types de P. Mockapetris

Como adicionar nomes de rede e máscaras ao DNS. Abril 1989

RFC 1122

Requirements for Internet Hosts - Communication Layers editado por R.

Braden

Não é relacionado diretamente com o DNS, mas a seção 4 discute alguns tópicos de UDP e TCP que têm efeitos importantes e de baixo nível no DNS.
Outubro 1989

RFC 1123

Requirements for Internet Hosts - Application and Support editado por R. Braden

Inclui o capítulo 6, sobre DNS. Outubro 1989

RFC 1178

Choosing a Name for Your Computer de D. Libes

Bons conselhos para se ter em mente quando se escolhe nomes para computadores. Agosto 1990

RFC 1183

New DNS RR Definitions de C. Everhart, L. Mamakos e R. Ullmann e editado por P. Mockapetris

Novos registros de recursos (RR - Resource Records), não muito usados.
Outubro 1990

RFC 1348 - atualiza a RFC 1035, substituída por RFC 1706.

Julho 1992

RFC 1464

Using the Domain Name System To Store Arbitrary String Attributes de R. Rosenbaum

Usando registros TXT para armazenar texto no DNS. Maio 1993

RFC 1480

The US Domain de A. Cooper e J. Postel

Orientações e procedimentos relacionados ao domínio de primeiro nível US.
Junho 1993

RFC 1535

A Security Problem and Proposed Correction With Widely Deployed DNS Software de E.Gavron

Ressalta as possibilidades de subversão com as listas de procura padrão do resolver. Outubro 1993

RFC 1536

Common DNS Implementation Errors and Suggested Fixes de A. Kumar, J.Postel, C. Neuman, P. Danzig e S. Miller

Para desenvolvedores. O que e como consertar erros. Outubro 1993

RFC 1537 - substituída pela RFC 1912.

Outubro 1993

RFC 1591

Domain Name System Structure and Delegation de J.Postel

Detalhes administrativos sobre o espaço de nomes DNS. Março 1994

RFC 1611

DNS Server MIB Extensions de R. Austein e J. Saperia

Uma interface SNMP para o lado do servidor DNS. Maio 1994

RFC 1612

DNS Resolver MIB Extensions de R. Austein e J. Saperia

Uma interface SNMP para o lado cliente DNS. Maio 1994

RFC 1637- substitui a RFC 1348, substituída pela RFC 1706.

Junho 1994

RFC 1664

Using the Internet DNS to Distribute RFC 1327 Mail Address Mapping Tables de C. Allochio, A.Bonito, B.Cole, S. Giordano e R.Hagens

Mapeando informações para conversão de endereços entre X.400 e SMTP no DNS. Agosto 1994

RFC 1706 - substitui RFC 1348 e RFC 1637

DNS NSAP Resource Records de B. Manning e R. Colella

Como adicionar NSAPs do estilo OSI no DNS usando registros PTR.
Outubro 1994

RFC 1712 - substituída pela RFC 1876

DNS Encoding of Geographical Location de C. Farrell, M. Schulze, S. Pleitner e D. Baldoni

Muito criticado por Paul Vixie, mas publicado mesmo assim. Novembro 1994

RFC 1713

Tools for DNS debugging de A. Romao

Visã o geral sobre algumas ferramentas DNS. Novembro 1994

RFC 1794

DNS Support for Load Balancing de T. Brisco

Suporte do DNS para balancear carga de vá rios tipos. Abril 1995

RFC 1876 - substitui a RFC 1712

A means for Expressing Location Information in the Domain Name System de C. Davis, P. Vixie, T. Goodwin e I. Dickinson

Registros de DNS para localizaçã o geográ fica. Janeiro 1996

RFC 1884

IP Version 6 Addressing Architecture editado por R. Hinden e S. Deering

Tudo sobre endereç os Ipv6. Dezembro 1995

RFC 1886

DNS Extensions to support IP version 6 de S. Thomson e C. Huitema

Extensõ es do DNS para compatibilidade com IPv6, incluindo o novo tipo de registro AAA e o novo domínio IP6.INT. Dezembro 1995

RFC 1912 - atualiza RFC 1537

Common DNS Operational and Configuration Errors de D. Barr

Erros e práticas comuns na operação de servidores e formatos de dados.
Fevereiro 1996

RFC 1956

Registration in the MIL Domain de D. Engebretson e R. Plzak

Descreve a orientação para registro do domínio do Departamento de Defesa norte-americano. Junho 1996

RFC 1982

Serial Number Arithmetic de R. Elz e R. Bush

Define como os números seriais são comparados para determinar se uma zona foi atualizada. Agosto 1996

RFC 1995

Incremental Zone Transfer in DNS de M. Ohta

Um mecanismo para uso com notificação (NOTIFY) que permite a transferência de somente a parte da zona que sofreu alterações. Agosto 1996

RFC 1996

Notify: a mechanism for prompt notification of authority zone changes de P. Vixie

Descreve o novo código de operação/controle NOTIFY para avisar servidores escravos que dados no servidor principal foram alterados. Agosto 1996

RFC 2010

Operational Criteria for Root Name Servers de B. Manning e P. Vixie

Requisitos para servidores de nomes da raiz. Outubro 1996

RFC 2052

A DNS RR for specifying the location of services (DNS SRV) de A. Gulbraandsen e P. Vixie

Registros MX generalizados para outros serviços além de mail. Outubro 1996

RFC 2053

The AM (Armenia) Domain de E. Der-Danieliantz

Procedimentos para registro no domínio AM TDL. Outubro 1996

RFC 2065

Domain Name System Security Extensions de D. Eastlake, 3rd e C. Kaufman

Assinaturas digitais para integridade dos dados e autenticação no DNS.
Janeiro 1997

RFC 2136

Dynamic Updates in the Domain Name System (DNS UPDATE) de P.Vixie (editor), S. Thomson, Y.Rekhter e J.Bound

Adição e exclusão atômicas a nível de registros das informações do DNS.
Abril 1997

RFC 2137

Secure Domain Name System Dynamic Update de D. Eastlake 3rd

Segurança para atualizações dinâmicas. Abril 1997

Apêndice 3 - Domínios *Top-Level*

Esta tabela lista todos os códigos de duas letras para países e os domínios de primeiro nível que não são os países. Nem todos os países estão registrados no espaço de nomes da Internet, mas estes não são muitos.

Domínio	País ou Organização
AD	Andorra
AE	Emirados Árabes Unidos
AF	Afeganistão
AG	Antigua e Barbuda
AI	Anguilla
AL	Albânia

AM	Armê nia
AN	Antilhas Holandesas
AO	Angola
AQ	Antá rtica
AR	Argentina
ARPA	Internet do DARPA (Estados Unidos)
AS	Samoa Oriental
AT	Áustria
AU	Austrá lia
AW	Aruba
AZ	Azerbaijã o
BA	Bósnia e Herzegovínia
BB	Barbados
BD	Bangladesh
BE	Bé lgica
BF	Burkina Faso
BG	Bulgá ria
BH	Bahrein
BI	Burundi
BJ	Benim
BM	Bermudas
BN	Brunei Darussalam
BO	Bolívia
BR	Brasil
BS	Bahamas
BT	Butã

BV	Ilha Bouvet
BW	Botsuana
BY	Belarus
BZ	Belize
CA	Canadá
CC	Ilhas Cocos (Keeling)
CF	República Central Africana
CG	Congo
CH	Suíça
CI	Cote d' Ivoire
CK	Ilhas Cook
CL	Chile
CM	República dos Camarões
CN	China
CO	Colômbia
CR	Costa Rica
CU	Cuba
CV	Cabo Verde
CX	Ilha Christmas
CY	Chipre
CZ	República Tcheca
DE	Alemanha
DJ	Djibuti
DK	Dinamarca
DM	Dominica
DO	República Dominicana

DZ	Argé lia
EC	Equador
EDU	educaç ã o (Estados Unidos)
EE	Estônia
EG	Egito
EH	Saara Ocidental
ER	Eritrea
ES	Espanha
ET	Etió pia
FI	Finlâ ndia
FJ	Fiji
FK	Ilhas Malvinas (Falkland)
FM	Microné sia
FO	Ilhas Fé roe
FR	Franç a
FX	Franç a, Metropolitana
GA	Gabã o
GB	Reino Unido da Grã -Bretanha (na prá tica, é usado "UK")
GOV	governo (Estados Unidos)
GD	Granada
GE	Geórgia
GF	Guiana Francesa
GH	Gana
GI	Gibraltar
GL	Groenlâ ndia
GM	Gâ mbia

GN	Guiné
GP	Guadalupe
GQ	Guiné Equatorial
GR	Grécia
GS	Ilhas South Georgia e South Sandwich
GT	Guatemala
GU	Guam
GW	Guiné -Bissau
GY	Guiana
HK	Hong Kong
HM	Ilhas Heard e McDonald
HN	Honduras
HR	Croácia
HT	Haiti
HU	Hungria
ID	Indonésia
IE	Irlanda
IL	Israel
IN	Índia
INT	entidades internacionais
IO	Território Britânico no Oceano Índico
IQ	Iraque
IR	Irã
IS	Islândia
IT	Itália
JM	Jamaica

JO	Jordânia
JP	Japão
KE	Quênia
KG	Quirguízia
KH	Cambodja
KI	Kiribati
KM	Comores
KN	São Cristóvão e Nevis
KP	Coreia do Norte
KR	Coreia do Sul
KW	Kuwait
KY	Ilhas Caimãs
KZ	Cazaquistão
LA	Laos
LB	Líbano
LC	Santa Lúcia
LI	Liechtenstein
LK	Sri Lanka (Ceilão)
LR	Libéria
LS	Lesoto (Basutolândia)
LT	Lituânia
LU	Luxemburgo
LV	Latvia
LY	Líbia
MA	Marrocos
MC	Mônaco

MD	Moldá via
MG	Madagascar
MH	Ilhas Marshall
MIL	militar (Estados Unidos)
MK	Macedônia
ML	Mali
MM	Myanmar
MN	Mongó lia
MO	Macau
MP	Ilhas Marianas
MQ	Martinica
MR	Mauritâ nia
MS	Montserrat
MT	Malta
MU	Maurício
MV	Maldivas
MW	Malawi (Niassalâ ndia)
MX	Mé xico
MY	Malá sia
MZ	Moçambique
NA	Namíbia
NATO	OTAN (Organizaçã o do Tratado do Atlâ ntico Norte)
NC	Nova Caledônia
NE	Níger
NF	Ilha Norfolk
NG	Nigé ria

NI	Nicarágua
NL	Holanda
NO	Noruega
NP	Nepal
NR	Nauru
NU	Niue
NZ	Nova Zelândia
OM	Omã
ORG	organizações (Estados Unidos)
PA	Panamá
PE	Peru
PF	Polinésia Francesa
PG	Papua - Nova Guiné
PH	Filipinas
PK	Paquistão
PL	Polônia
PM	Saint Pierre e Miquelon
PN	Pitcairn
PR	Porto Rico
PT	Portugal
PW	Palau
PY	Paraguai
QA	Qatar
RE	Reunião
RO	Romênia
RU	Rússia

RW	Ruanda
SA	Arábia Saudita
SB	Ilhas Salomão
SC	Seychelles
SD	Sudão
SE	Suécia
SG	Cingapura
SH	Santa Helena
SI	Slovênia
SJ	Svalbard (Spitsbergen)
SK	Slováquia
SL	Serra Leoa
SM	San Marino
SN	Senegal
SO	Somália
SR	Suriname (Guiana Holandesa)
ST	São Tomé e Príncipe
SV	El Salvador
SY	Síria
SZ	Suazilândia (Ngwane)
TC	Ilhas Turke e Caicos
TD	Chade
TF	Territórios Franceses do Sul
TG	Togo
TH	Tailândia (Sião)
TJ	Tadjiquistão

TK	Ilhas Tokelau
TM	Turcomenistã o
TN	Tunísia
TO	Tonga
TP	East Timor
TR	Turquia
TT	Trinidad e Tobago
TV	Tuvalu (Ilhas Ellice)
TW	Taiwan (Formosa)
TZ	Tanzâ nia (Tanganica e Zanzibar)
UA	Ucrâ nia
UG	Uganda
UK	Reino Unido (Grã -Bretanha)
UM	Ilhas Midway
US	Estados Unidos
UY	Uruguai
UZ	Usbequistã o
VA	Cidade do Vaticano
VC	Sã o Vicente e Grenadines
VE	Venezuela
VG	Ilhas Virgens (Britâ nicas)
VI	Ilhas Virgens (E.U.A)
VN	Vietnã
VU	Vanuatu
WF	Wallis e Futura
WS	Samoa

YE	Iê men
YT	Mayotte
YU	Iugoslá via
ZA	África do Sul
ZM	Zâ mbia (Rodé sia do Norte)
ZR	Zaire



Bibliografia

[ALB 97]

ALBITZ, Paul; LIU, Cricket. DNS and BIND, segunda edição. Sebastopol: O' Reilly & Associates, Inc., Jan. 1997.

[ARN 94]

ARNETT, Matthew Flint; DULANEY, Emmett; HARPER, Eric. Inside TCP/IP. Indianapolis: New Riders Publishing, 1994.

[HUN 92]

HUNT, Craig. TCP/IP Networking Administration. Sebastopol: O' Reilly & Associates, Inc., 1992.

[COR 91]

CORNER, Douglas E. Internetworking with TCP/IP, volume 1, segunda edição. Englewood Cliffs: Prentice Hall, 1991.

[RFC1033]

LOTTOR, M. K. Domain Administrators Operations Guide. Menlo Park: SRI International, DDN Network Information Center; Nov. 1987; RFC 1033. 22 p.

[RFC1032]

STAHL, M. K. Domain Administrators Guide. Menlo Park: SRI International, DDN Network Information Center; Nov. 1987; RFC 1032. 14p.

[RFC1034]

MOCKAPETRIS, Paul. Domain Names - Concepts and Facilities. Marina del Rey: University of Southern California, Information Sciences Inst.; Nov. 1987; RFC 1034. 55p.

[RFC1035]

MOCKAPETRIS, Paul. Domain Names - Implementation and Specification. Marina del Rey:

University of Southern California, Information Sciences Inst.; Nov. 1987; RFC 1035. 55p.

[NIC50007]

GARCIA-LUNA, Jose; STAHL, Mary K.; WARD, Carol A. Internet Protocol Handbook: The Domain Name System (DNS) Handbook, volume quatro. Menlo Park: SRI International, Network Information Systems Center; Ago. 1989; NIC50007. 219p.

[RFC974]

PARTRIDGE, Craig. Mail Routing and the Domain System. Cambridge: CSNET CIC BBN Labs., Inc.; Jan. 1986; RFC 974. 7p.

[RFC920]

POSTEL, J.; REYNOLDS, J. Domain Requirements. Marina del Rey: University of Southern California, Information Sciences Inst.; Out. 1984; RFC 920; 14p.

[RFC1101]

MOCKAPETRIS, Paul. DNS Encoding of Network Names and Other Types. Marina del Rey: University of Southern California, Information Sciences Inst.; Abr. 1989; RFC 1101; 14p.

[DEL 86]

McNALLY, Rand. Atlas Delta Universal. Rio de Janeiro: Editora Delta, 1986.

