DNS - Domain Name System

Luiz Arthur

Este material tem como referência os autores:

TANENBAUM, Andrew S. **Redes de Computadores**. 4ª ed. Campus, 2003.

COMER, Douglas. **Interligação em rede com TCP/IP: Princípios, protocolos e arquitetura** vol. 1. Rio de Janeiro: Campus, 1998.

TORRES, Gabriel. **Redes de computadores: curso completo**. Rio de Janeiro: Axcel Books, 2001.

HUNT, Craig. **Linux Servidores de rede**. Editora Ciência Moderna. Rio de Janeiro. 2004.

DNS - Domain Name System

Luiz Arthur

Toda rede baseada no modelo TCP/IP utiliza endereços IP's para identificar redes e hosts, tais endereços são representados por 32 bits e em decimal é algo parecido com: `200.123.229.233`. Tais endereços IP's permitem o endereçamento e o roteamento de pacotes através de uma ou mais redes de computadores, tal como a Internet.

Porém, **é um pouco complicado lembrar de um endereço IP** quando queremos acessar algum host em uma dada rede, o que piora rapidamente se tivermos que acessar vários hosts, cada um com seu IP.

Imagine o seguinte cenário: Se para acessarmos o site de uma dada empresa tivéssemos que digitar em nosso browser (Firefox, Opera, Internet Explorer, etc) o endereço **`http://9.255.171.235`** e para acessarmos os e-mails desta mesma empresa, que são armazenados em um servidor diferente do site, tivéssemos que acessar o servidor `200.135.5.145`, ou seja, para mandarmos um e-mail para o funcionários joão deveríamos digitar em nosso email **`joao@200.135.5.145`**. Este cenário seria assustador já que diariamente podemos trabalhar com dezenas ou centenas de hosts na Internet.

Desta forma, teria de haver uma “lista telefônica” para a Internet, mas imprimir e gerenciar tal lista seria quase que impossível o que torna rapidamente inviável está prática, além do que deixaria a Internet menos dinâmica, já que na Internet atual as empresas, por exemplo, podem mudar de endereço IP constantemente.

Ainda levando em consideração o cenário descrito anteriormente imagine que a dada empresa mudasse o servidor de email para o endereço IP `111.234.132.10`, todos os interessados deveriam ser notificados de tal mudança para continuar enviando e-mails para tal empresa, tal como acontece com os números de telefone.

Tentando amenizar este cenário **foram introduzidos nomes em ASCII para desacoplar os momes das máquinas dos endereços IP dessas máquinas**. Desse modo, o endereço do e-mail do joão pode ser algo como `joao@empresa.com.br`. Todavia, a **rede TCP/IP em si só reconhece endereços IP**, sendo desta forma, necessário algum tipo de mecanismo para converter as strings ASCII em endereços IP. Tal mecanismo é chamado de **servidor DNS, no qual são enviados nomes e devolvidos endereços IP's**.

Assim, caso a empresa mude o endereço IP de seu servidor de e-mail esta mudança não será vista a principio por quem utiliza o domínio `empresa.com.br`, pois é só dizer ao servidor de nomes DNS que o nome `empresa.com.br` agora responde pelo IP `111.222.121.212` (IP do novo servidor) ao invés do endereço IP antigo (`200.135.5.145`), isto facilita muito a vida de quem utiliza as redes de computadores, tanto é que a maioria das pessoas desconhecem que a Internet é movida por endereços IP's, pois estas pessoas em sua grande maioria só conhecem nomes e domínios (`www.google.com.br`, `www.yahoo.com.br`, `www.kernel.org`, `www.linux.org`, `www.microsoft.com`), o que ajuda a Internet a se tornar mais amigável.

DNS - Domain Name System

Luiz Arthur

Nos primórdios da ARPANET, havia simplesmente um arquivo, **`hosts.txt`**, que listava todos os nomes de hosts e seus respectivos endereços IP, ou seja, **este arquivo fazia o papel de um servidor** de nomes e tinha por exemplo, a referencia de que o nome `www.empresa.com.br` era equivalente ao IP `111.222.121.212`. Toda noite, esse arquivo era copiado por todos os hosts de forma que tais nomes e seus IP's fossem compartilhado com todos os hosts da rede. Para uma rede de algumas centenas de máquinas, essa estratégia funcionava razoavelmente bem. No entanto, quando milhares de máquinas foram conectadas à rede, essa **estratégia** se tornou rapidamente **inviável**.

Para resolver tal problema, foi criado o **DNS (Domain Name System – Sitema de Nomes de Domínios**) a essência do DNS é a criação de um **esquema hierárquico de atribuição de nomes** baseado no domínio e de um sistema de **banco de dados distribuídos** para implementar esse esquema de nomenclatura. O servidor DNS é usado principalmente para mapear nomes de hosts e destinos de mensagens de correio eletrônico em endereços IP.

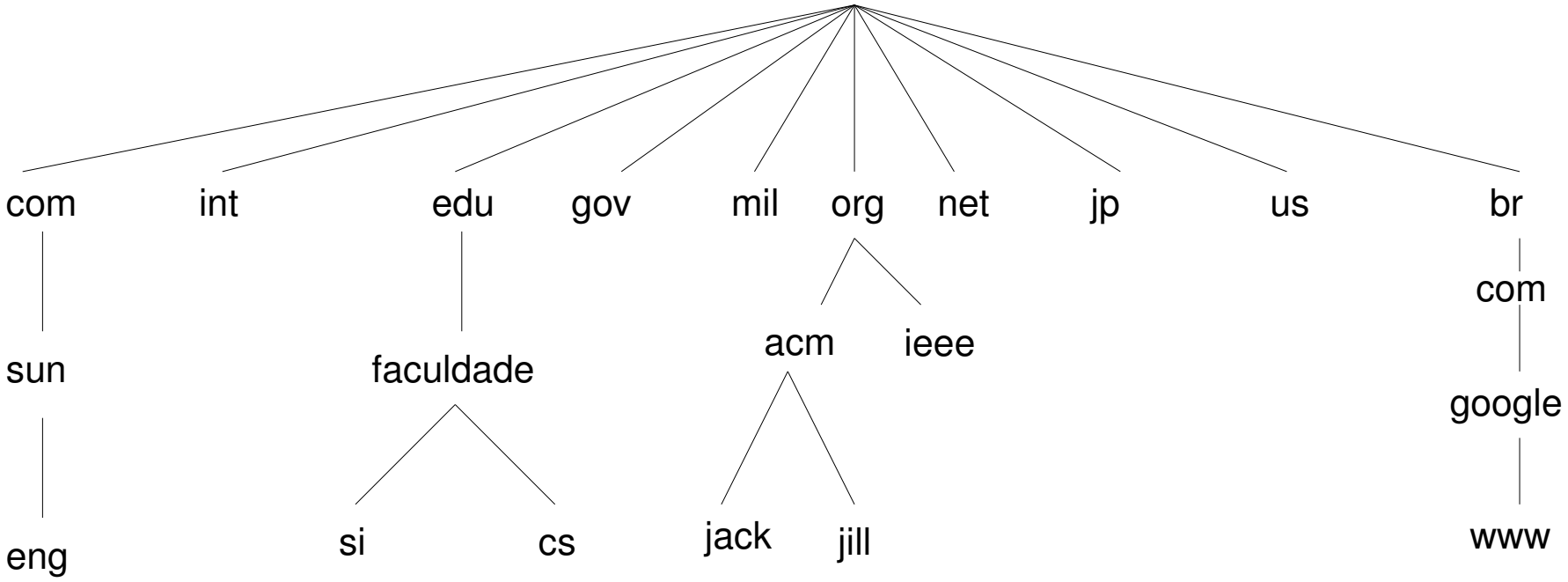
Em resumo, **o DNS é utilizado para mapear um nome em um endereço IP**, sendo que um programa aplicativo chama um procedimento de biblioteca denominado **resolvedor** e repassa a ele o nome a ser consultado como um parâmetro. O resolvedor envia um pacote **UDP** a um servidor DNS local, que procura o nome e **retorna o endereço IP ao resolvedor**. Em seguida, o resolvedor retorna o endereço IP ao programa aplicativo que fez a chamada. Munido do endereço IP, o programa pode então estabelecer uma conexão com o destino.

DNS - Domain Name System

Luiz Arthur

Gerenciar um grande conjunto de nomes que está constantemente mudando não é fácil. Em um sistema posta, tal como os **correios**, o gerenciamento de nomes é feito através do uso de letras que especificam, o pais, o estado, a cidade e a rua do destinatário. Através do uso desse tipo de endereçamento hierárquico, não há confusão entre o João da Silva que mora na Rua XI de Novembro, São Paulo, e o João da Silva que mora na Rua XI de Novembro, Parana. O DNS funciona da mesma forma.

Conceitualmente, a **Internet é dividida em mais de 20 domínios** de nível superior, onde cada domínio cobre muitos hosts. Cada domínio é particionado em subdomínios, que também são particionados e assim por diante.



Existem dois tipos de **domínios de nível superior**: genéricos e de países. **Os domínios genéricos** originais eram `com` (comercial), `edu` (instituições educacionais), `gov` (instituições governamentais), `int` (certas organizações internacionais), `mil` (órgãos militares), `net` (provedores de rede) e `org` (organizações sem fins lucrativos). **Os domínios de países** incluem uma entrada para cada país, confirme a definição da ISO 3166 (Argentina-AR, Brasil-BR, Itália-IT, Japão-JP, Estados Unidos-US). Com o passar do tempo novos domínios podem ser acrescentados conforme surge a necessidade, isto já ocorreu e continuara acontecendo.

Em geral, **é fácil obter um domínio de segundo nível**, como `empresa.com`. Isso exige apenas um registro do domínio de nível superior correspondente (nesse caso, `com`) para verificar se o nome desejado está disponível e não é marca registrada de outra pessoa. Se não houver nenhum problema, o solicitante pagará uma pequena **taxa anual** e conseguirá o nome.

Cada domínio tem seu nome definido pelo caminho ascendente entre ele e a raiz (sem nome). Esses componentes são **separados por pontos**. Dessa forma o departamento de vendas de nosso domínio empresa poderia ser `vendas.empresa.com`, em vez de um nome no estilo UNIX `/com/empresa/vendas` ou no bem conhecido Windows `c:\com\empresa\vendas`. Observe que essa nomenclatura hierárquica significa que `vendas.empresa.com` não entra em conflito com um possível udo de vendas em `vendas.comercio.com`, que provavelmente é o departamento de vendas de outra instituição.

Os **nomes de domínio não fazem distinção entre letras maiúsculas e minúsculas**. Portanto edu, Edu e EDU têm o mesmo significado. Os nomes de componentes podem ter até 63 caracteres, e os nomes de caminhos completos não podem exceder 255 caracteres.

Em princípio, **os domínios podem ser inseridos na árvore de duas formas distintas**. Por exemplo, `cs.yale.edu` poderia ser igualmente listado sob o domínio de país us como `cs.yale.ct.us`.

Contudo, **na prática, quase todas as organizações dos Estados Unidos estão sob um domínio genérico** e, praticamente, todas fora dos Estados Unidos estão sob o domínio de seu país. Não existe regra contra o registro sob dois domínios de nível superior, mas poucas organizações além das multinacionais o fazem (por exemplo, `sony.com` e `sony.nl`).

Cada domínio controla como serão alocados todos os domínios que estão abaixo dele. Por exemplo, o Japão tem os domínios `ac.jp` e `co.jp` que espelham `edu` e `com`. A Holanda não faz essa distinção e coloca todas as organizações diretamente sob `nl`.

Para que um novo domínio seja criado, é necessária a permissão de domínio no qual ele será incluído. Por exemplo, se o grupo de Sistemas de Informação tiver começado em um domínio chamado `faculdade.edu` e quiser ser conhecido como `si.informatica.faculdade.edu`, ele precisará da permissão de quem gerencia `informatica.faculdade.edu`.

Desta forma, os conflitos de nomes são evitadas e cada domínio pode controlar seus subdomínios. **Uma vez que um novo domínio tenha sido criado e registrado, ele poderá criar subdomínios**, sem que seja necessária a permissão de alguém que esteja em um nível mais alto na árvore.

A atribuição de nomes leva em consideração as fronteiras organizacionais, e não as redes físicas. Por exemplo, mesmo que os departamentos de ciência da computação e de engenharia elétrica estejam localizados no mesmo prédio e compartilhem a mesma LAN, ele poderão ter domínios distintos. Da mesma forma, mesmo que o departamento de ciência da computação esteja dividido em dois prédios, normalmente todos os hosts instalados em ambos pertencerão ao mesmo domínio.

Registros de recursos

Todo domínio, independente de ser um único host ou um domínio de nível superior, pode ter um conjunto de registros de recursos associados a ele. Para um único host, o registro de recurso mais comum é apenas seu endereço IP, mas também existem muitos outros tipos de registros de recursos.

Quando um resolvedor repassa um nome de domínio ao DNS, o que ele obtém os registros de recursos associados ao nome em questão. Portanto, a principal função do DNS é mapear nomes de domínio em registros de recursos.

Um registro de recurso é um tupla de cinco campos. Apesar de serem codificados em binário para proporcionar maior eficiência, na maioria das exposições, os registros de recurso são mostrador como texto ASCII, uma linha para cada registro de recurso. O formato que utilizaremos é:

```
Domain_name      Time_to_live Class  Type   Value
```

`Domain_name`: informa o **domínio** ao qual esse registro se aplica. Normalmente, existem muitos registros para cada domínio, e cada cópia do banco de dados armazena informações sobre vários domínios. Assim, esse campo é a chave de pesquisa primária utilizada para atender às consultas. A ordem dos registros no banco de dados não é significativa.

`Time_to_live`: fornece uma **indicação da estabilidade do registro**. As informações muito estáveis são definidas como um número alta como 86.400 (o número de segundos em 1 dia). As informações muito voláteis recebem um número baixo como 60 (1 minuto).

O terceiro campo de cada registro de recurso é **`Class`**. No caso de informações relacionadas à Internet, ele é sempre **`IN`**. Para informações não relacionadas à Internet, podem ser empregados outros códigos; porém, esses outros códigos raramente são encontrados na prática.

O **campo `Type`** informa qual é o **tipo do registro**. Os tipos mais importantes estão são: `SOA`, `A`, `MX`, `NS`, `CNAME`, `PTR`, `HINFO` e `TXT`.

Um registro `SOA` (Inicio de Autoridade) **fornece o nome da principal fonte de informações sobre a zona do servidor de nomes**, o endereço de correio eletrônico do administrador, um número de série exclusivo e diversos flags e timeouts.

O tipo de registro mais importante é o registro **`A`** (Address). Ele **contém um endereço IP de algum host**. Todos os hosts da Internet devem ter pelo menos um endereço IP, de forma que outras máquinas possam se comunicar com ele. Alguns hosts têm duas ou mais conexões de rede; nesse caso, eles terão um registro de recurso do tipo `A` por conexão de rede. O DNS pode ser configurado para circular por esses endereços, retornando o primeiro registro na primeira solicitação, o segundo registro na segunda solicitação e assim por diante, isto é útil para prover alta disponibilidade ou balanceamento de carga.

O próximo tipo de registro é o registro **`MX`**. **Ele especifica o nome do host preparado para aceitar mensagens de correio eletrônico** para o domínio especificado. O registro `MX` é utilizado porque nem toda máquina está preparada para aceitar correio eletrônico. Se alguém quiser enviar correio eletrônico para `bill@microsoft.com`, o host transmissor precisará encontrar um servidor de correio em `microsoft.com` que esteja disposto a aceitar correio eletrônico. O registro `MX` pode fornecer essa informação.

Os registros **`NS` especificam servidores de nomes**. Por exemplo, todos os bancos de dados DNS têm um registro `NS` para cada um dos domínios de nível superior; assim, as mensagens de correio eletrônico podem ser enviadas para partes distantes da árvore de atribuição de nomes.

Os registro **`CNAME` permite a criação de nomes alternativos**. Por exemplo, uma pessoa familiarizada com a atribuição de nomes na Internet em geral que deseja enviar uma mensagem para alguém cujo nome de login seja `paul` no departamento de ciência da computação do MIT poderá imaginar que `paul@cs.mit.edu` seja o endereço correto. Na realidade, esse endereço não servirá, pois o domínio do departamento de Sistemas da Informação do MIT é `lcs.mit.edu`. No entanto, o MIT poderia criar uma entrada CNAME para orientar pessoas e programas na direção correta, oferecendo um serviço para quem não sabe disso. Uma entrada como esta poderia executar essa função.

```
cs.mit.edu   86400  IN CNAME  lcs.mit.edu
```

A exemplo do `CNAME`, `PTR` **indica outro nome**. No entanto, ao contrário de `CNAME` que, na verdade, é apenas uma definição de macro, `PTR` é um tipo de dados comum do DNS cuja interpretação depende do contexto no qual se encontra. **Na prática, essa entrada é quase sempre usada para associar um nome a um endereço IP**, a fim de permitir pesquisas de endereços IP e retornar o nome da máquina correspondente. Essas pesquisas são chamadas **pesquisas inversas**, ou reversas.

Os registros **HINFO** permitem que as pessoas descubram a que **tipo de máquina e sistema operacional** um domínio corresponde. Por fim, os registros TXT permitem que os domínios se identifiquem de forma arbitrária. Esses dois tipos de registros são usados para conveniência do usuário. Nenhum deles são obrigatório.

Por fim, chegamos ao campo Value. Esse campo pode ser um número, um nome de domínio ou um string ASCII.

```
; Authoritative data for cs.vu.nl
cs.vu.nl.    86400   IN  SOA star boss (952771,7200,7200,2419200,86400)
cs.vu.nl.    86400   IN  TXT “Divisão Wiskunde de Informática.”
cs.vu.nl.    86400   IN  TXT “Universidade Vrije Amsterdam.”
cs.vu.nl.    86400   IN  MX  1   zephyr.os.vu.nl.
cs.vu.nl.    86400   IN  MX  2   top.cs.vu.nl.

flits.cs.vu.nl   86400   IN  HINFO   Sun Unix
flits.cs.vu.nl   86400   IN  A       130.37.16.112
flits.cs.vu.nl   86400   IN  A       192.31.231.165
flits.cs.vu.nl   86400   IN  MX      1 flits.cs.vu.nl.
flits.cs.vu.nl   86400   IN  MX      2 zephyr.os.vu.nl.
www.cs.vu.nl     86400   IN  CNAME   star.os.nu.nl.
fto.cs.vu.nl     86400   IN  CNAME   zephyr.os.vu.nl.

Rowboat                  IN  A       130.37.56.201
                         IN  MX      1 rowboat
                         IN  MX      2 zephyr
                         IN  HINFO   Sun Unix
```

O que **não é mostrado** no exemplo anterior e que não está nesse arquivo são **os endereços IP a serem usados na pesquisa de domínios de nível superior**. Esses endereços são necessários para pesquisar hosts distantes, mas como não fazem parte do domínio `cs.vu.nl`, eles não estão nesse arquivo. **Esses endereços são fornecidos pelos servidores raiz**, cujos endereços IP estão presentes em um arquivo de configuração do sistema e são carregados para o cache do DNS quando o servidor DNS é inicializado.

Existem cerca de uma dezena de servidores raiz espalhados pelo mundo, e que cada um deles conhece os endereços IP de todos os servidores de domínio de nível superior. Desse modo, se uma máquina conhecer o endereço IP de pelo menos um servidor raiz, ela poderá pesquisas qualquer nome DNS.

Servidores de nomes

Pelo menos na teoria, **um único servidor de nomes poderia conter o banco de dados DNS inteiro e responder a todas as consultas referentes ao banco**. **Na prática**, esse servidor ficaria tão sobrecarregado que seria inútil. Além disso, caso **esse servidor viesse a ficar fora do ar, toda a Internet seria atingida**.

Para evitar os problemas associados à presença de uma única fonte de informações, o espaço de nomes do **DNS é dividido em zonas não superpostas**.

Cada zona contém uma parte da árvore e também servidores de nomes que armazenam informações referentes a essa zona. **Normalmente, uma zona terá um servidor de nomes principal, que obtém suas informações a partir de um arquivo contido em sua unidade de disco e um ou mais servidores de nomes secundários, que obtêm suas informações a partir do servidor de nomes principal**.

A localização das fronteiras de uma zona fica a cargo de seu administrador. Essa decisão é tomada principalmente com base no número de servidores de nomes desejados.

Quando um **resolvedor** tem uma **consulta** sobre um nome de domínio, ele a envia a um dos **servidores de nomes locais**. Se o domínio que estiver sendo procurado estiver sob jurisdição do servidor de nomes, serão retornados os registros de recursos oficiais. Um registro oficial é aquele fornecido pela autoridade que gerencia o registro e, portanto, sempre está correto. Os registros mantidos em cache, ao contrário dos registros oficiais, podem estar desatualizados.

No entanto, **se o domínio for remoto e não houver informações sobre o domínio solicitado disponíveis no local, o servidor de nomes enviará uma mensagem de consulta para o servidor de nomes de nível superior** fazendo perguntas sobre o domínio solicitado. Este método é denominado **consulta recursiva**, pois cada servidor que não tiver as informações solicitadas poderá encontrá-las em algum lugar e informar o que encontrou.

Também é possível uma **forma alternativa**. Nessa estratégia, quando não pode ser satisfeita no local, a **consulta falha**, mas **é retornado o nome do próximo servidor** a ser consultado ao longo da linha. Alguns servidores não implementam consultas recursivas e sempre retornam o nome do próximo servidor a ser consultado.

Também vale a pena destacar que, quando um cliente DNS deixar de obter uma resposta antes de seu timer expirar, em geral ele tentará acessar outro servidor na próxima vez.

Embora o DNS não seja extremamente importante para o funcionamento correto da Internet, já que tudo que ele faz na realizada é mapear nomes simbólicos de máquinas em seus endereços IP. Mas uma coisa é certa com a possibilidade do uso de nomes ao invés de endereços IP's, proporcionada pelo o DNS, **facilita muito a vida dos Internautas**.

DNS – BIND

A maioria dos sistemas Linux, o DNS é implementado com o software Berkeley Internet Name Domain (BIND).

O DNS do BIND é um sistema cliente/servidor. **O cliente é chamado de resolvedor**, e forma as consultas e as envia ao servidor de nomes. Todo computador em sua rede executa um resolvedor. Muitos sistemas só executam o resolvedor.

Configurando o resolvedor

O resolvedor no Linux é configurado por **dois tipos** de arquivos. **Um tipo diz ao resolvedor que serviços de nomes usar e em que ordem usá-los**. O outro arquivo de configuração, `/etc/resolv.conf`, configura o resolvedor para usar interação com o servidor DNS. Toda vez que um processo que usa o resolvedor começa, ele lê o arquivo `/etc/resolv.conf`, e guarda em cache a configuração, durante a vida do processo. Caso não exista este arquivo o sistema local é utilizado. Mas o uso do arquivo `/etc/resolv.conf` é altamente recomendável, digamos que é praticamente obrigatório.

Configuração do resolvedor

O arquivo `/etc/resolv.conf` é um arquivo texto que pode conter os seguintes comandos:

• `nameserver` *`endereço_IP`* – **Define o endereço IP de um servidor de nome** que o resolvedor deve usar. Até três comandos **`nameserver`** podem ser incluídos na configuração, sendo que os servidores são consultados na ordem que aparecem no arquivo. Normalmente o primeiro servidor é o que responde as consultas, os outros só entrarão em ação caso o primeiro servidor esteja indisponível. Se não existir o `nameserver` a consulta será feita localmente.

• search *searchlist* – **Define uma lista de domínios que são usadas para ampliar um hostname antes de ser enviado ao servidor de nome** searchlist contém até seis nomes de domínio, separados por um espaço em branco, sendo que a pesquisa é feita na ordem. Ao contrário do comando domain, que cria uma lista de pesquisa default que contém apenas o domínio local, o comando search cria uma lista de pesquisa explícita que contém todos os domínios especificados em *searchlist*.

• options *option* – **Modifica o comportamento padrão do resolvedor**. Há várias opções disponíveis:

- Degub - liga a depuração e emite saídas que podem serem utilizadas, por exemplo, para resolver problemas;
- ndots:*n* - Define o número de pontos que indicam quando o resolvedor deve juntar valores da lista de pesquisa para o hostname antes de enviar a consulta ao servidor de nomes. Por padrão o resolvedor não modificará um hostname se contiver apenas um ponto. Como resultado o hostname www será estendido com um valor na lista de pesquisa antes de ser enviado ao servidor de nomes, mas o hostname www.empresa não. Mas usando a opção ndots para modificar este comportamento, por exemplo, ndots:2, tal hostname [www.empresa] será submetido ao search. O único momento que ndots é requerido é se algum componente de seu domínio pudesse ser confundido com um domínio do nível de topo.);

- `timeout:`*n* - Ajusta o intervalo de tempo de consulta inicial para o resolvedor, o padrão é cinco segundos. Mude este valor apenas se o seu servidor demorar mais do que cinco segundos para responder sua consulta);
- `attempts:`*n* - Define o número de vezes nas quais o resolvedor tentará novamente uma consulta, o padrão é 2;
- `rotate` - Liga a possibilidade de compartilhamento de carga entre os servidores de nome, executando uma rotatividade entre eles, esta opção fará um rodizio entre os servidores de nome, o que não é feito por padrão;
- `inet6` - Faz o resolvedor consultar por endereços IPv6; Existem outras opções mais essas são as mais populares.

Um exemplo de arquivo `resolv.conf`:

```
# cat /etc/resolv.conf
search si.faculdade.edu   faculdade.edu
nameserver 192.168.0.1
nameserver 200.200.200.200
```

O resolvedor Lightweitgh

O BIND 9 introduziu uma nova biblioteca resolver lightweight (de peso leve). A nova biblioteca pode ser ligada em qualquer aplicação, mas **foi projetada para aplicações que precisam usar endereços IPv6**. Isto foi necessário por que o **IPv6 aumentou muito a complexidade do resolvedor**. Por isto, o resolvedor lightweight divide o resolvedor em uma **biblioteca usada pelas aplicações** e um **daemon de resolvedor** separado que controla o tamanho do processo do resolvedor. As rotinas de biblioteca enviam consultas a porta 921 UDP no host local usando o protocolo do resolvedor lightweight. O daemon do resolvedor pega a consulta, e a soluciona usando protocolos de DNS padrão.

O daemon do resolvedor é `lwresd`. É essencialmente um **servidor cache** de nomes que recursivamente soluciona consultas para a biblioteca do resolvedor lightweight. O `lwresd` não requer o mesmo nível de configuração que um servidor de cache, porque alguns valores defaults, como a lista de servidores raiz, são compilados em `lwresd`. Ao contrário, `lwresd` usa os mesmos comandos de configuração que o resolvedor stub, e lê sua configuração de `resolv.conf`. Porém, interpreta os comandos `nameserver` de uma maneira ligeiramente diferente. Se comandos `nameserver` estiverem definidos em `resolv.conf`, `lwresd` trata os servidores listados lá como expedidores e tenta remeter todas as consultas a estes servidores para resolução. Na maioria dos casos, o comando `lwrest` é executado sem qualquer opção, mas elas existem. E lembre-se, **`lwresd` não é necessário para a maioria dos sistemas**.

fim