Eduardo Kalil de Santana

Integração de DNS e DHCP utilizando atualização dinâmica

Monografia apresentada ao Departamento de Ciência de Computação da Universidade Federal de Lavras, como parte das exigências do curso de Pós-graduação *Lato Sensu* em Administração de Redes Linux, para obtenção do título de Especialista em Administração em Redes Linux.

Orientador
Prof. Samuel Pereira Dias

LAVRAS
MINAS GERAIS – MG
2006

Eduardo Kalil de Santana

Integração de DNS e DHCP utilizando atualização dinâmica

Monografia apresentada ao Departamento de Ciência de Computação da Universidade Federal de Lavras, como parte das exigências do curso de Pós-graduação *Lato Sensu* em Administração de Redes Linux, para obtenção do título de Especialista em Administração em Redes Linux.

APROVADA em 29 de Setembro de 2006

_____

Prof. Dsc. Heitor Augustus Xavier Costa

_____

Prof. Msc. Denilson Vedoveto Martins

_____

Prof. Samuel Pereira Dias
(Orientador)

LAVRAS
MINAS GERAIS – MG
2006

DEDICATÓRIA

Aos meus pais, Sebastião e Rosane que sempre me passaram força e incentivo e me mostraram que os sonhos podem ser realizados.

AGRADECIMENTOS

Agradeço a Deus por ter me dado força e confiança. Agradeço em especial ao professor Samuel que apesar de toda a dificuldade não desistiu do projeto, aos profissionais que trabalham comigo que com sua experiência de vida e profissional me ajudaram no possível, aos amigos e, mais uma vez, a minha família.

RESUMO

Este trabalho trata da integração do DNS e DHCP utilizando a atualização dinâmica, em *software* livre. A iniciativa surgiu graças à necessidade do Governo Federal migrar do *software* proprietário para o *software* livre. O texto mostra conceitos sobre DNS, DHCP, atualização dinâmica e que soluções livres podem substituir soluções proprietárias. É mostrado como configurar a atualização dinâmica e como configurar clientes Linux e Windows. Por fim é apresentado um projeto que implementa a atualização dinâmica no MCT – Ministério da Ciência e Tecnologia, os resultados obtidos no projeto.

Sumário

Lista de Figuras

1. Introdução

Atualmente, várias empresas e órgãos governamentais estão migrando seus sistemas proprietários para sistemas baseados em *software* livre. Existem duas formas de realizar uma migração, a primeira é transparente ao usuário e a segunda é não-transparente ao usuário.

Os serviços transparentes são aqueles que os usuários não tomam conhecimento do que está acontecendo na rede, ou seja, não alteram as atividades do seu dia-a-dia, por exemplo a migração do servidor de correio eletrônico e DNS (*Domain Name System* – Sistema de Nomes de Domínios). Esses serviços geralmente são migrados quando não há pico de usuários e sua substituição raramente é percebida por eles durante a leitura de suas mensagens ou navegação. Os serviços não-transparentes são os que podem alterar a forma como o usuário realiza suas funções cotidianas, por exemplo os aplicativos manipulados diretamente pelo o usuário, como sistemas operacionais, suítes de escritório, navegador *Web*, clientes de *e-mail* e outros programas.

O objetivo deste trabalho é apresentar a integração de dois serviços que são importantes no uso redes de computadores, o DHCP (*Dynamic Host Configuration Protocol* – Protocolo de Configuração Dinâmica de *Hosts*) e o DNS, através de atualização dinâmica. O DNS e DHCP são serviços importantes para o administrador gerenciar sua rede com agilidade e flexibilidade. O serviço de DHCP permite a atribuição automática de endereços IP (*Internet Protocol*) a

estações durante sua inicialização, além de outras informações necessárias para configuração do acesso à rede local. O serviço de DNS fornece mecanismos para resolução de nomes mnemônicos em endereços IP da rede ou externos.

A atualização dinâmica facilita o gerenciamento da rede, mantendo sua escalabilidade e a consistência dos dados sobre cada estação conectada. Com as bases de dados continuamente atualizadas, a localização de estações também torna-se mais rápida e confiável, oferecendo conforto aos usuários. Considerando o cenário crescente nas empresas, em que a mobilidade vem tornando-se comum e necessária, a atualização dinâmica permite que usuários possam mover-se entre departamentos da empresa, usando endereçamento diferentes, sem ter seu nome de *host* alterado.

Além do objetivo principal do trabalho, que é mostrar a configuração da atualização dinâmica com DNS e DHCP, o trabalho visa também mostrar que a cada dia que passa soluções livres estão se equivalendo ou, em certos aspectos, sendo superiores às soluções proprietárias. Por fim, este trabalho também objetiva suprir falta de documentação sobre o assunto, que se encontra escassa e contendo poucas informações.

A solução apresentada será empregada no projeto de migração de *software* proprietário para *software* livre na rede do Ministério da Ciência e Tecnologia – MCT. No referido projeto, estão os serviços de DNS, DHCP e atualização dinâmica, que junto com outros serviços são executados no sistema operacional MS-Windows 2003. Tais serviços são oferecidos por produtos

proprietários, em sua maioria compostos por serviços nativos do próprio sistema operacional. A solução apresentada neste texto foi empregada em um ambiente de teste e atendeu às necessidades e requisitos para entrar em produção[1]. No ambiente de teste foi, usado como servidor o sistema operacional FreeBSD, onde os serviços foram instalados e configurados, e quatro máquinas como clientes, Linux (Debian), Linux (Fedora), Windows XP e Windows 2000.

A integração do DNS com DHCP pode ser descrita, sucintamente, como a atribuição de configurações aos clientes pelo DHCP (IP, máscara de rede, servidor DNS, entre outras configurações) de forma automática. Após a atribuição destas configurações, inicia-se o processo de atualização, onde o DHCP, de posse do nome da máquina cliente, solicita a atualização da zona do DNS, incluindo ou atualizando os registros necessários.

No Capítulo 2, são mostrados conceitos básicos sobre DNS e DHCP não sendo abordados, entretanto, detalhes sobre a configuração dos serviços. A atualização dinâmica é analisada com mais detalhe no Capítulo 3. No Capítulo 4, encontram-se os recursos de segurança que são oferecidos pelo BIND. No Capítulo, 5 é apresentada a configuração da atualização dinâmica, mostrando os parâmetros que serão necessários para o seu funcionamento. O Capítulo 6, mostra como configurar os clientes Windows e Linux. No capítulo 7, será mostrado o resultados obtidos no projeto. O Capítulo 8 e dedicado a conclusão do projeto.

1 Significa que o(s) serviço(s) foram aprovados nos testes e serão aplicados na rede principal.

2. DHCP e DNS

2.1. DHCP

O DHCP (*Dynamic Host Configuration Protocol*) é um acrônimo para Protocolo de Configuração Dinâmica de *Hosts.* A principal função do DHCP é configurar os *hosts* de forma dinâmica, ou seja, de forma automática sem intervenção do administrador. De acordo com [1], o DHCP consiste em dois componentes básicos:

- a função de entregar determinadas configurações fornecidas pelo servidor aos clientes;
- um mecanismo que entrega os endereços aos clientes.

Como descrito em [2], quando um *host* cliente está configurando para receber configurações pelo DHCP, ele pode receber algumas configurações e se conectar a uma rede local. Através do DCHP, são designados a ele um endereço IP, o endereço do servidor DNS e diversas outras configurações que, dependendo da necessidade da rede estarão disponíveis. Cabe a ressalva que nem todos os serviços poderão ser configurados pelo DHCP.

O DHCP também faz a renumeração quando os endereços de rede ou a máscara de rede são modificados. Também faz a reconfiguração dos clientes, caso o DNS ou WINS tenham seus endereços de rede alterados.

Na Figura 1, é mostrado o funcionamento de uma rede que faz o uso de um servidor DHCP. Assim que o sistema e inicializado ou mesmo determinado

pelo próprio usuário, o cliente manda uma mensagem, que é enviada para toda a rede (*broadcast*), perguntando quem é o servidor de DHCP na rede. Caso haja algum servidor ativo, ele responde e envia as configurações definidas. Assim, o cliente obtém um endereço IP e conhece quem é o *gateway* da rede, servidor DNS e outras configurações.

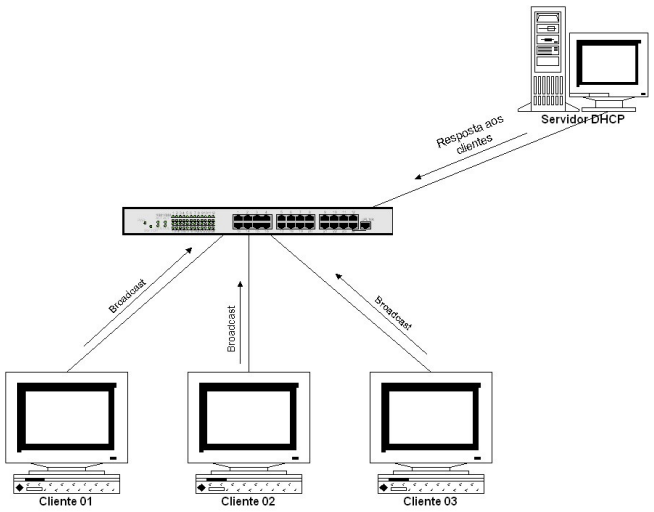

Figura 1: Comunicação do DHCP

2.2. DNS

Antes de conhecer o DNS (*Domain Name System*), é preciso conhecer um pouco da ARPAnet. Como descrito em [2], em meados dos anos 60 surgiu a ARPAnet, que foi desenvolvida pelo departamento de defesa dos Estados Unidos. Inicialmente surgiu, a ARPA (*Advanced Research Projects Agency*),

que tinha a função de conectar organizações de pesquisas e com sua grande utilização, surgiu a ARPAnet, que nessa época se popularizou por dois motivos. O primeiro foi a criação do TCP/IP (*Transmission Control Protocol/Internet Protocol*) e a segunda foi a conectividade da ARPAnet com o protocolo TCP/IP, ou seja, todos os computadores, conectados à ARPAnet, passariam a adotar o TCP/IP como padrão.

Outro ponto importante foi o desenvolvimento e o progresso do UNIX, da Universidade de Berkley, conhecido como BSD, que foi pioneiro no uso do TCP/IP como uma camada de rede. O BSD estava disponível para outras universidades a um custo muito baixo e, com isso, o número de computadores ligados à ARPAnet cresceu.

Com o grande crescimento das redes, havia a necessidade que serviços fossem criados para facilitar o gerenciamento dessas redes e, com esse grande crescimento, foi desenvolvido o DNS. De acordo com [3], o DNS tem a função de traduzir nomes de *host* em endereços IP, usando o protocolo UDP (*User Datagram Protocol*), na porta 53.

De acordo com [4], o sistema do DNS funciona como uma base de dados distribuída e essa estrutura permite que a base seja compartilhada, estando disponível para a rede interna ou externa e sendo acessado pelo mecanismo cliente/servidor. Hoje, existem vários servidores espalhados pelo mundo que compartilham esses mapeamentos, provendo a resolução de nomes.

De acordo com [5], o sistema da base de resolução de nomes é dividido

em 03 níveis: o primeiro nível é chamado de '*root servers*', o segundo nível é composto por TLDs (*Top Level Domains*) e o terceiro nível é formado pelos domínios concedidos às organizações, como exemplificado na Figura 2.

Com citado, existem vários servidores espalhados pelo mundo, esses servidores são conhecidos como *root servers* e são representados como ".", ou seja, são a raiz da base de dados do DNS. Eles possuem informações sobre os "*Top Level Domains* (TLDs)", que de acordo com [4], são divididos em sete domínios: *com*, *edu*, *gov*, *mil*, *net*, *org* e *int*. Um servidor raiz encaminha uma consulta para o TLD apropriado, que por sua vez a repassa para o servidor DNS responsável pelo domínio requisitado.

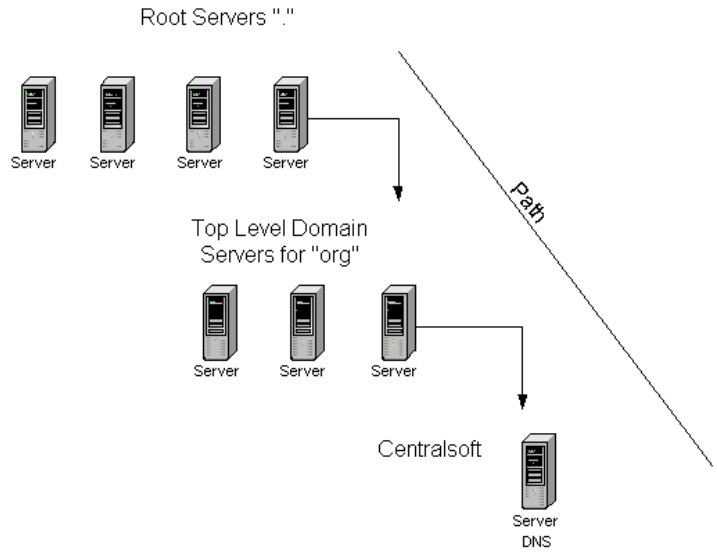

Figura 2: Estrutura de pesquisa DNS

A Figura 2 mostra com mais detalhes o caminho seguido quando uma

pesquisa é requisitada. Por exemplo, em uma rede local, um cliente faz a requisição ao domínio 'linux.org' para o servidor de DNS local. Se a informação não estiver em *cache* de uma consulta anterior, o servidor DNS local faz a requisição ao *root server*, que devolve o endereço do TLD responsável pelo '.org' para que o DNS local repita a consulta. O TLD, ao ser consultado, também redireciona o DNS local ao servidor de domínio do nível abaixo que é responsável pelo domínio ‘linux.org’, para que a consulta seja realizada. Por fim, o servidor de DNS responsável pelo domínio ‘linux.org’ é capaz de responder ao cliente qualquer endereço que esteja em sua zona de autoridade. Essas consultas podem ser armazenadas em *cache* pelo DNS local por um período de tempo, para evitar um novo acesso externo.

3. Atualização Dinâmica

Uma das principais responsabilidades do administrador é garantir que a rede esteja organizada e funcional. Quando se fala em organização, é preciso definir algumas características, por exemplo, as máquinas devem ter seu endereço IP (*Internet Protocol*) e seu nome de *host* definidos, endereços e nomes devem ser únicos e o cabeamento, estruturado. Quanto à funcionalidade, deve-se garantir que os serviços disponíveis na rede estejam funcionando corretamente e disponíveis sempre que forem requisitados.

Como pode-se observar, existem duas formas para identificar um *host* na rede: através de seu endereço IP (numérico) e por meio de um nome (alfanumérico). Para os usuários, é mais fácil lidar com nomes que a seqüência numérica dos endereços IP. É necessário, portanto, que os nomes estejam definidos e associados a um endereço IP e que cada *host* tenha como saber quais são seus dados. Existem dois serviços que trabalham em específicos com eles, o DNS e o DHCP.

Em uma rede com vários *hosts,* seria inviável para que o usuário soubesse os endereços, sendo necessário um serviço de DNS configurado e que saiba exatamente o que responder quando for feita a solicitação pelo nome. Para que o DNS saiba responder a essa requisição e preciso que a sua base de dados, composta pelos arquivos de zona, esteja preenchida com informações sobre os *hosts*, é possível preencher a base de duas maneiras: manualmente e de forma

automática.

A forma manual não é viável por ser trabalhosa, especialmente para redes de médio e grande porte. Manter a unicidade dos endereços e a atualização da base seria uma tarefa dispendiosa e que consumiria tempo. A forma automatizada, tema deste trabalho, libera o administrador da tarefa de atualização, mas para isso, é preciso integrar os serviços de DNS e DHCP.

De acordo com [6], o termo atualização dinâmica é usado quando se refere a modicar, deletar, atualizar ou adicionar registros em um arquivo de zona. Tal procedimento pode ser feito de forma individual para cada zona e, ainda assim, manter um nível de segurança com uso das opções TSIG (*Transaction Signatures*) e DNESEC *(Domain Name System Security Extensions)*, junto com as regras `allow-update` e `update-police` (detalhadas no Capítulo 4), no arquivo `named.conf`[2].

A atualização dinâmica passou a ser uma característica importante do DNS, incluída a partir da versão 8 do BIND (*Berkeley Internet Name Domain*). É um recurso relativamente recente e o autor considera que, futuramente, o serviço de atualização dinâmica será bem mais utilizado.

O formato da mensagem de atualização é definido em [7] e pode ser visto na Figura 3. Essa mensagem é responsável por descrever detalhes do sistema de domínios e protocolos.

---

2 Na maioria das distribuições Linux, encontrado em `/etc/named.conf`.

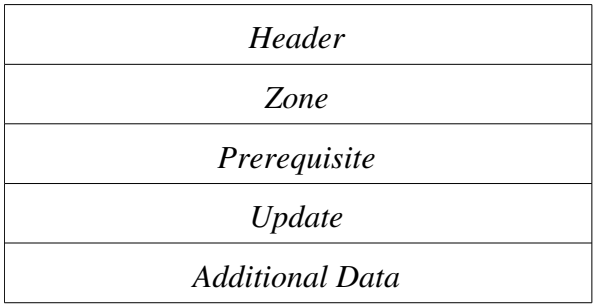

Figura 3: Formato da Mensagem de Atualização Dinâmica

O campo "*Header*" especifica que a mensagem será para atualização e descreve os tamanhos dos outros campos. O campo "*Zone*" especifica qual será a zona atualizada. O campo "*Prerequisite*" especifica o início das variantes, ou seja, conteúdo das zonas que são requisitos para a atualização. O campo "*Update*" contém a modificação que será aplicada e o campo "*Additional Data*" é reservado para conter dados que possam ser necessários para completar a atualização.

Com a atualização dinâmica em funcionamento, o registro adicionado no arquivo de zona é parecido com os dados exibidos na Figura 4, o *host* de nome TYGEL foi adicionado com o endereço (campo `A "Adress Record"`) 10.6.4.134, o campo `TXT` (*The documentation entries*) registra as entradas de texto em formato livre, onde são colocadas informações pertinentes ao sistema ou ao usuário.

```
$TTL 21600  ; 6 hours
TYGEL A   10.6.4.134
      TXT "31637023f03893120ed1b1c26d8ce1ef72"
```

Figura 4: Entrada adicionada dinamicamente à zona de DNS

4. Segurança

Como visto na Seção 2.2, o DNS é responsável por traduzir nomes de domínios para seus respectivos endereços IP (*Internet Protocol*). Como todo serviço de rede, é suscetível a falhas de segurança, portanto é de suma importância ter um certo nível de conhecimento nas opções de segurança oferecidas pelo sistema. Por exemplo, o TSIG *(Transaction Signatures),* que, de acordo com [8], está presente desde a versão 8.2 do BIND, funciona como um complemento no nível de segurança do BIND. Ele permite que as transações sejam realizadas de forma segura através de chaves criptográficas.

O DNSSEC *(Domain Name System Security Extensions),* como descrito em [8], é uma extensão de segurança para o BIND, que foi implementado a partir de sua versão 9. Este conjunto de regras favorece a adição segura de novos registros, pelo uso de chaves, combinando a funcionalidade do TSIG com o DNSSEC e possível deixar o BIND mais seguro.

Para gerar uma chave TSIG, é necessário usar um comando que faz parte do DNSSEC, o `dnssec-keygen`. Esse comando é apresentado no Capítulo 5, juntamente com o processo de criação da chave e aplicação das regras `update-police` ou `allow-update`.

A diretiva `update-police` é um conjunto de regras, com as quais é possível negar ou permitir operações para nomes a serem atualizados no arquivo de zona. Por outro lado, a `allow-update` permite que a atualização ocorra

sem restrições. Em ambos os casos, é possível gerar uma chave TSIG (*Transaction signature*), ou seja, com a criação das chaves, será possível autenticar e criptografar os dados. É importante ressaltar que as diretivas são mutuamente exclusivas, ou seja, apenas uma pode estar presente na configuração.

A integração entre DNS e DHCP não é possível usando a diretiva `update-police`, embora a atualização dinâmica em si seja permitida. É necessário usar `allow-update` para integrar com o serviço de DHCP e realizar a atualização dinâmica.

5. Configuração dos serviços

Este trabalho não visa mostrar como e a configuração básica do DNS e do DHCP. O foco principal é mostrar como implementar o uso da atualização dinâmica entre o DNS e o DHCP.

O *software* usado para os serviços do DNS e do DHCP foram o BIND *(Berkeley Internet Name Domain)* e o DHCP3. Os testes com BIND foram feitos a partir da versão 9, sem a aplicação de *patch* para corrigir falhas ou aumentar suas funcionalidades.

Para configurar o serviço de atualização dinâmica, inicialmente será preciso gerar o par de chaves secretas. Essas chaves serão usadas para que DHCP consiga se comunicar com o DNS.

O comando usado para gerar as chaves é o `dnssec-keygen`. Com ele, é possível definir o tamanho das chaves de 1 a 512 *bits* de comprimento. As chaves devem ser criadas no mesmo diretório em que estão os arquivos de configuração do BIND e ter permissão de leitura e escrita apenas para o proprietário do arquivo (representação octal "600"). A sintaxe do comando é mostrado na Figura 5 e suas opções são:

- `-a`: Seleciona o algoritmo de criptografia, RSAMD5 ou RSASHA1, DSA, DH, ou HMAC-MD5. É obrigatorio usar HMAC-MD5 para TSIG, de acordo com o manual do comando;
- `-b`: Especifica o tamanho da chave em *bits*, dependendo da criptografia

escolhida;

- `-n`: Especifica o tipo de chave que será usada (*host*, *zone*, *entity*, etc.).

```
dnssec-keygen -a <algoritimo> -b <tamanho da chave> -n
HOST <nome da chave>
```

Figura 5: Sintaxe do `dnssec-keygen`

Como mostrado na Figura 6, o comando cria o par de chaves com o nome `dhcp_dns`. Na primeira linha, encontram-se o comando e suas opções e, na segunda linha, observa-se a saída de sua execução. Após a criação da chave, é possível observar dois arquivos novos no diretório, um chamado `Kdhcp_dns.+157+56426.key` e o outro, denominado `Kdhcp_dns.+157+56426.private`.

```
# dnssec-keygen -a HMAC-MD5 -b 128 -n HOST dhcp_dns
Kdhcp_dns.+157+56426
```

Figura 6: Exemplo de criação de chaves

O arquivo com a extensão *.key* contém a chave-pública que será usada na integração. O arquivo *.private* contém o campo com o algoritmo que foi gerado pelo criptografia simétrica HMAC-MD5.

5.1. DNS

Para configurar o DNS, é preciso adicionar algumas linhas no arquivo de configuração do BIND (named.conf). A Figura 7 apresenta as linhas que devem ser adicionadas ao named.conf: o nome da chave, o algoritmo que foi usado e também a chave secreta que foi gerada com o comando mostrado na Figura 6 (encontra-se no arquivo com extensão *private*).

```
key dhcp_dns {
      algorithm hmac-md5;
      secret "h6y5nGdLzCHEL4jZUK4tgA==";

}
```

Figura 7: Configuração da chave no arquivo `named.conf`

A Figura 8 apresenta um exemplo de configuração da zona do domínio. A configuração varia de acordo com a necessidade de cada administrador, mas pode-se ressaltar a indicação de qual chave será usada nas atualizações da zona.

```
zone “teste.org.br” IN {
     type master;
     notify no;
     file “teste.org.br.dns”;
     allow-update { key dhcp_dns; };
};
zone “1.6.10.in-addr.arpa” IN {
     type master;
     notify no;
     file “0.1.10.rev”;
     allow-update { key dhcp_dns; };
};
```

Figura 8: Configuração das zonas de domínio

5.2. DHCP

Para configurar o DHCP, é necessário adicionar algumas linhas ao arquivo `dhcpd.conf`[3]. O procedimento é parecido com o que foi feito no arquivo `named.conf` e algumas linhas são as mesmas que foram usadas no aquivo de configuração do BIND.

Na Figura 9, apresenta-se a linha que deve ser adicionada no início da configuração do DHCP para que a atualização dinâmica seja ativada. Após essa linha, são adicionadas as mesmas linhas mostradas na Figura 10. A Figura 10 apresenta as linhas de configuração adicionais que descrevem as zonas que serão usadas na atualização dinâmica. O objetivo é identificar qual a zona será atualizada, em que servidor se encontra e qual chave será usada na atualização. Após essa configuração é necessário reiniciar os serviços de DNS e DHCP e

3 Na maioria das distribuições Linux, encontrado em `/etc/dhcpd/dhcpd.conf`.

configurar os clientes.

```
# Esta linha ativa o DNS dinâmico
ddns-update-style interim;
```

Figura 9: Ativação da atualização dinâmica no DHCPD

```
zone teste.org.br. {
     primary 10.6.1.1;
     key dhcp_dns;
}
zone 1.6.10.in-addr.arpa. {
     primary 10.6.1.1;
     key dhcp_dns;
}
```

Figura 10: Identificação das zonas no DHCP

6. Configuração do cliente

Para configurar um cliente Linux ou Windows, é necessário que o adaptador de rede esteja habilitado para o modo DHCP e não no modo STATIC. Para configurar um cliente Linux, e necessário editar o arquivo `/etc/dhclient-eth0.conf`. Caso o arquivo não exista, é necessário criá-lo acrescentando a linha mostrada na Figura 11.

```
send host-name “nome da máquina”;
```

Figura 11: Configuração do cliente Linux

Em seguida, o arquivo com as configurações de rede deve ser editado. Neste trabalho, usando uma distribuição compatível com a *Red Hat*, o arquivo é `/etc/sysconfig/network-scripts/ifcfg-eth0`. As configurações são apresentadas na Figura 12. Após configurar, é preciso reiniciar o serviço de rede do sistema.

```
DEVICE=eth0
ONBOOT=yes
BOOTPROTO=dhcp
USERCTL=no
PEERDNS=yes
TYPE=Ethernet
DHCP_HOSTNAME=<nome da maquina>
```

Figura 12: Configuração da placa de rede no Linux

Para máquinas que usam o Windows, é preciso verificar na configuração do sistema se a placa de rede está configurada para usar DHCP. O procedimento é válido para o Windows 2000 e XP. Para iniciar a configuração é preciso seguir

os seguintes passos: clique no *menu* “Iniciar” “Configurações”, “Painel de Controle” e “Conexão de Rede”.

Após selecionar essa opção, será exibida uma janela como a mostrada na Figura 13. Selecione a conexão e clique com o botão direito do *mouse* e selecione a opção “Propriedades”. Em seguida, será mostrada uma tela semelhante à mostrada na Figura 14.

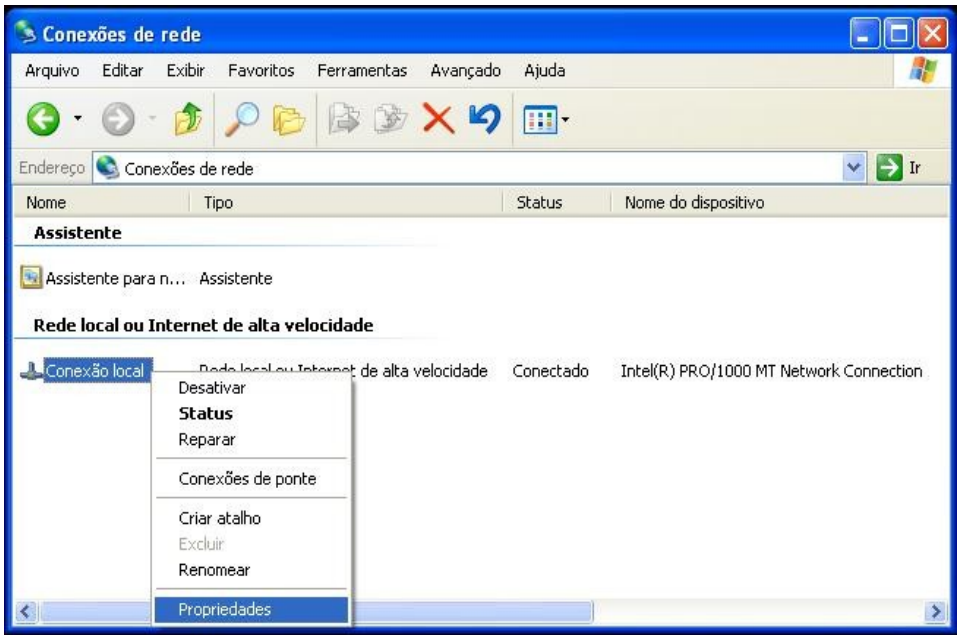

Figura 13: Configuração de rede no Windows

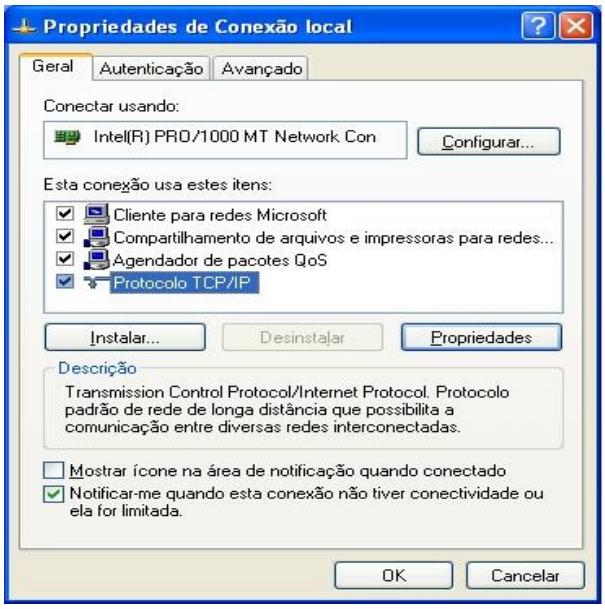

Figura 14: Propriedades da conexão de rede

Na janela da Figura 14, selecione o opção “Protocolo TCP/IP” e clique em “Propriedades”. Será exibida uma janela semelhante à Figura 15, onde deve-se selecionar as opções “Obter um endereço IP automaticamente” e “Obter o endereço dos servidores de DNS automaticamente”.

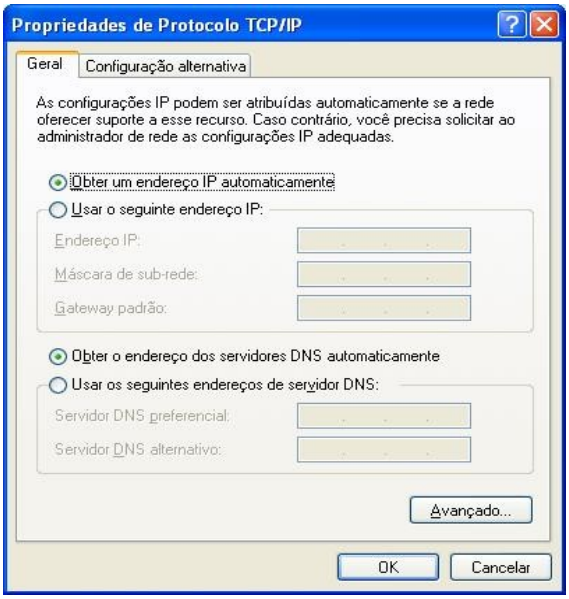

Figura 15: Propriedades do TCP/IP

7. Resultados e Discussão

Com o crescimento das redes de computadores, serviços como o DNS e DHCP mostram que a administração de redes pode deixar ser uma atividade complexa. Esses serviços existem para facilitar a administração e atualização da rede.

O trabalho foi realizado no sistema operacional FreeBSD, para a configuração dos serviços foi utilizada apenas uma máquina para atuar como servidor. Os serviços de DNS e DHCP foram instalados e configurados na sua configuração básica para ter os serviços funcionando, para que a integração pudesse ser feita.

Após configurar a integração do DNS e o do DHCP, foram adicionadas algumas entradas para teste através de clientes Linux e Windows. Foi observado que o sistema cria um arquivo com a extensão ".jnl" no mesmo diretório que o arquivo de domínio do DNS foi configurado. Esse arquivo possui informações sobre as entradas que foram feitas e tem a função de recuperar as últimas entradas que foram adicionadas em caso de desligamento incorreto ou repentino do sistema (um registro das transações para recuperação das informações, parecido com o usado nos *filesystems* que possuem suporte ao *journaling)*. Esse arquivo é binário e é lido apenas pelo serviço de DNS.

Foi observado que se, futuramente, o administrador necessitar deletar ou fazer alguma alteração nos registros existentes na base, o procedimento terá que

ser feito de forma manual. Em outras palavras, o administrador terá que editar o arquivo de zona manualmente para remover entradas que não são mais necessárias, pois não existe ainda um serviço de exclusão automática da base de dados. Essa característica evita que usuários mal intencionadas possam excluir registros, tornando a base incompleta e inconsistente.

8. Conclusão

Para uma rede que funciona completamente com *software* proprietário, soluções alternativas em *software* livre teriam que ser apresentadas. O *software* livre usado neste trabalho para DNS e DHCP é amplamente empregado por muitos administradores de rede. A atualização dinâmica existe nas duas formas de licenciamento de *software*, embora haja uma pequena quantidade de documentação disponível sobre o assunto, especialmente empregando soluções livres. Espera-se que este trabalho preencha parte dessa lacuna.

A intenção de conseguir integrar o DNS com DHCP foi alcançada, pois foi possível fazer com que os serviços trabalhassem em conjunto no ambiente de teste. Com essa integração, foi possível dar um maior dinamismo à rede, permitindo uma maior mobilidade dos usuários. Além disso, a meta de tê-la em *software* livre foi aproximada com a migração desses serviços.

Como trabalho futuro, pode-se identificar a necessidade de uma ferramenta para gerenciamento do serviço de atualização dinâmica. Essa ferramenta permitiria a adição, para modificação e para remoção de registros do arquivo de domínio do DNS em vez de realizar uma edição manual. Além disso, poderia automatizar o processo de configuração dos serviços e criação das chaves.

Uma questão que foi levantada e que pode servir de referência para continuidade do trabalho também, seria como aumentar a segurança da

atualização dinâmica, como por exemplo, somente máquinas da rede local poderiam fazer essa atualização, com impedir que outra máquina de fora, como um *notebook* de um visitante, faça essa atualização.

9. Referências Bibliográficas

[1] R. Droms, **Dynamic Host Configuration Protocol (DHCP)**, RFC 2131, March 1997. Disponível na web em: (http://www.ietf.org/rfc/rfc2131.txt?number=2131). Último acesso e em: 16/08/2006.

[2] Ball, Bill; Pitts, David et al. **Dominando Rede Hat Linux 7**, 2002

[3] HUNT, Craig. Linux Servidores de Rede, 2004

[4] Liu, Cricket and Albitz; Paul**. DNS and BIND, Fifth Edition**, 2006

[5] Adelstein, Tom; Traditional DNS Howto. Disponível na Web em: (**http://www.howtoforge.com/traditional_dns_howto**)**.** Último acesso em 18/06/2006.

[6] Vixie, P., Thomson, S., Rekhter, Y. and J. Bound, **Dynamic Updates in the Domain Name System (DNS UPDATE)**, RFC 2136, April 1997. Disponível na web em: (http://www.ietf.org/rfc/rfc2136.txt?number=2136). Último acesso e em: 16/08/2006.

[7] P. Mockapetris, **DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION**, RFC 1035, November 1987. Disponível na web em: (http://www.ietf.org/rfc/rfc1035.txt?number=1035). Último acesso e em: 16/08/2006.

[8] BAUER, Michel D. Building Secure Servers with Linux, 2002

[9] Foster, Matt. ***Dynamic DNS with BIND9 and DHCP***. Disponível na web

em: (http://www.mattfoster.clara.co.uk/ddns.htm). Último acesso e em: 16/08/2006.

Apêndice A

Exemplo de configuração do *dhcpd.conf*

```
# dhcpd.conf

ddns-update-style interim;

key dhcp_dns {
     algorithm hmac-md5;
     secret "h6y5nGdLzCHEL4jZUK4tgA==";
};

zone teste.org.br. {
     primary 10.6.1.1;
     key dhcp_dns;
}
zone 1.6.10.in-addr.arpa. {
     primary 10.6.1.1;
     key dhcp_dns;
}

option domain-name "teste.org.br";
option domain-name-servers 10.6.1.1;
option netbios-name-servers 10.6.1.42;

default-lease-time 21600;
max-lease-time 43200;


subnet 10.6.0.0 netmask 255.255.0.0 {
  range dynamic-bootp 10.6.4.1 10.6.4.254;
  option broadcast-address 10.6.255.255;
  option routers 10.6.1.1;
}
```

Exemplo de configuração do *named.conf*

```
# named.conf

options {
        directory       "/etc/namedb";
        version         "[No version!]";
        query-source address * port 53;
        pid-file        "/var/run/named/pid";
        dump-file       "/var/dump/named_dump.db";
        statistics-file "/var/stats/named.stats";
        listen-on       { 127.0.0.1; 10.6.1.40;  };
        notify yes;
};

key dhcp_dns {
     algorithm hmac-md5;
     secret "h6y5nGdLzCHEL4jZUK4tgA= =";


        zone "." {
                type hint;
                file "named.root";
        };

        zone "0.0.127.IN-ADDR.ARPA" {
                type master;
                file "master/localhost.rev";
        };

        zone "1.6.10.in-addr.arpa" {
                type master;
                file "master/1.6.10.in-addr.arpa";
        };

    zone "teste.org.br" IN {
                    type master;

                    notify no;
```

```
                allow-update { key dhcp_dns; };
                file "teste.org.br.dns";
        };
        zone "1.6.10.in-addr.arpa" IN {
                type master;
                notify no;
                allow-update { key dhcp_dns; };
                file "0.1.10.rev";
        };
};
```