SERVIÇOS DE REDES DE COMPUTADORES

UNIDADE 4 - PROTOCOLO DHCP E SERVIDOR PROXY: QUAIS SÃO AS SUAS FUNÇÕES?

Aline Izida

Introdução

O termo redes de computadores se tornou um termo antiquado, embora já esteja enraizado e utilizamos para nos referir a uma disciplina e a diversos contextos nela inseridos. Não apenas computadores, como também diversos dispositivos fazem parte de uma rede informatizada, digamos assim. Desde computadores pessoais, computadores do tipo servidores, *tablets*, *smartphones*, até roteadores.

No início, nosso meio de acessar à rede mundial de computadores era o computador, no entanto, rapidamente, surgiram os outros dispositivos. Primeiro surgiu o *notebook*, um computador portátil que permitiu que os usuários acessassem diversas redes aonde quer que fosse necessário, em casa, no trabalho, na universidade ou na cafeteria. Hoje, os *smartphones* tomaram uma grande proporção e são responsáveis pelo aumento em potencial da utilização da internet.

As tecnologias de redes tiveram que ser aprimoradas ao passo em que a internet cresceu. Uma dessas tecnologias foi o servidor DHCP, permitindo que qualquer dispositivo se conecte à rede de forma automática, de modo que você, como usuário, não perceba qualquer interrupção na sua navegação na internet, por exemplo.

Já imaginou como seria se cada vez que você mudasse de rede ou sub-rede (o que, às vezes, você nem percebe que mudou), tivesse que solicitar a um profissional de redes do local para configurar seu dispositivo com todas as configurações necessárias para acessar à internet? Seria inviável, não é mesmo? Ainda mais em locais onde diversos usuários entram e saem e precisam acessar à internet.

Outra questão importante que vamos abordar nesta unidade é sobre o servidor *proxy*. Você já reparou que as páginas *web* geralmente carregam de forma rápida? O *proxy* tem uma função de cache que faz com que arquivos da página *web* já acessada sejam salvos localmente, o que evita a busca dos arquivos no servidor *web* remoto, tornando, assim, seu acesso mais prático e ágil.

Ao acessar um computador na sua universidade ou no trabalho, você sabia que o administrador de rede pode configurá-la para que seu usuário não tenha acesso a determinados *sites* ou funções? Isso também é possível graças às configurações do servidor *proxy*, objetivando a segurança da rede interna de uma organização.

Vamos aprender mais detalhes sobre esses assuntos nesta unidade. Ótimos estudos!

4.1 Serviço DHCP: função

De acordo com Comer (2016), o antecessor do DHCP foi o *Bootstrap Protocol* (BOOTP). Antes do computador poder usar o BOOTP para obter um endereço, o administrador de rede deveria configurar o servidor BOOTP para conhecer o endereço IP do computador. Com o passar do tempo, tivemos a necessidade de acessar pontos de *Wi-Fi* em estabelecimentos, por exemplo, e isso não era possível com BOOTP, surgindo, então, a necessidade de criar um servidor DHCP com a capacidade dinâmica de distribuição de endereços IP.

O *Dynamic Host Configuration Protocol* (DHCP), ou Protocolo de Configuração Dinâmica de Host ou de Endereços de Rede, permite que um computador arbitrário participe de uma nova rede e obtenha um endereço IP de forma automática sem a necessidade de configuração manual do servidor. Por vezes, esse conceito pode ser chamado de *plug and play networking*, isto é, a rede onde o computador se conecta e sai rodando diretamente (COMER, 2016; SCHMITT, 2013).

VOCÊ SABIA?

O DHCPv6 foi criado para possibilitar que os endereços IPv6 sejam administrados de forma central, contudo, o IPv6 foi projetado para usar a configuração automática, de modo a possibilitar que dois nós IPv6 isolados se comunicassem em uma rede não administrada e sem quaisquer servidores. Portanto, em vez de usar o DHCP, foi concebido um nó IPv6 capaz de gerar seu próprio endereço IP único.

No geral, o DHCP trata de configurações necessárias para que um dispositivo esteja conectado em uma rede TCP-IP. Isso significa que um dispositivo habilitado para utilizar a internet ou uma rede interna precisa estar configurado com pelo menos quatro parâmetros de rede: endereço IP, máscara de rede, endereço do primeiro roteador da rede, o *gateway* padrão (*default*), e endereço do servidor de nomes (DNS).
Conforme apontam Kurose e Ross (2013), um administrador de rede pode configurar o DHCP para que determinado *host* obtenha o mesmo endereço IP sempre que se conectar. Outra possibilidade é que o *host* obtenha um endereço IP temporário diferente sempre que se conectar. Além disso, como já mencionado, o DHCP possibilita que o *host* descubra informações adicionais.

VOCÊ QUER LER?

O protocolo DHCP está definido nos documentos RFC 2131 (DROMS, 1997) (https://tools.ietf.org/pdf/rfc2131.pdf.) e 2132 (ALEXANDER; DROMS, 1997) (https://tools.ietf.org/pdf/rfc2132.pdf.). Nesses documentos, é possível entender o protocolo em detalhes, desde seu conceito até seu funcionamento. RFC significa *Request for Comments* ou Pedido para Comentários. São documentos técnicos desenvolvidos e mantidos pela instituição *Internet Enginnering Task Force* (IETF), que especifica os padrões implementados e utilizados na internet.

O DHCP é ideal para situações em que há muito trânsito de usuários, pois os endereços são usados por um tempo limitado. A maioria dos servidores é configurada para utilizar um conjunto de endereços dinâmicos que são atribuídos a *hosts* arbitrários. Não podemos obter um endereço em nosso *host* e mantê-lo para sempre, por isso cada atribuição de endereço é limitada a um tempo fixo. Essa alocação é chamada de *lease* ou locação. Esse tempo de *lease* é especificado pelo administrador de rede quando estabelece o conjunto de endereços.

VAMOS PRATICAR?

A rede de uma universidade é configurada com algumas sub-redes. Um estudante pode conectar seu *notebook* ou seu *smartphone* na biblioteca, no centro de convivência, no restaurante, em diversos blocos das respectivas faculdades etc. Para cada local, ele pode se conectar a uma nova sub-rede, o que significa que precisará de um novo endereço de IP em cada uma delas.

Pesquise e analise como o administrador de rede dessa universidade deve configurar essa rede e com qual protocolo, para que os dispositivos sejam identificados com seus endereços IP em cada local que estiverem.

De acordo com Comer (2016), utilizar a locação possibilita que o servidor DHCP recupere endereços que já foram atribuídos, se isso for necessário. Além disso, quando a concessão do endereço expira, o *host* que estava usando o endereço pode abandoná-lo ou pode renegociar com o DHCP a extensão do tempo de locação. Enquanto o DHCP executa suas atividades, os usuários não percebem nada, como, por exemplo, que uma locação foi estendida.

VOCÊ O CONHECE?

Não é uma personalidade, mas sim uma empresa. A Xerox PARC é uma importante empresa no ramo da informática. Fundada na década de 1970 por Alan Kay e Jack Goldman, foi responsável pela criação do conceito cliente-servidor, muito utilizado em serviços *web*, tais como o DHCP. Você pode acessar o *site* da empresa pelo *link*: https://www.parc.com/. Com sede no Vale do Silício, até hoje a PARC trabalha com grandes especialistas em pesquisas científicas, buscando inovações na área computacional.

Assim, embora o DHCP atue atribuindo IP de modo temporário, esse tempo pode ser estendido caso o *host* permaneça na rede mais tempo do que o definido pela configuração do DHCP. Em geral, esse tempo é automaticamente estendido. Entretanto, a configuração pode ser realizada de modo que as extensões sejam negadas e o endereço não possa mais ser usado pelo *host*, quando isso for necessário.
A figura a seguir mostra um esquema simples do funcionamento do DHCP.

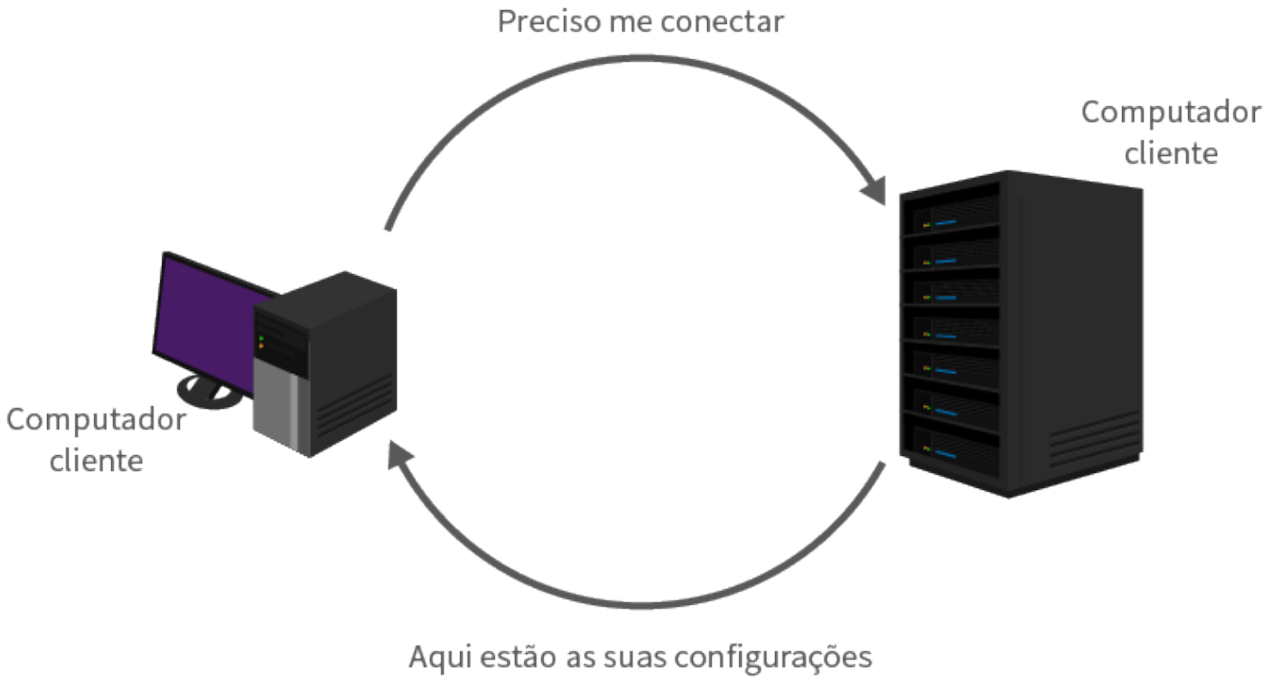

Figura 1 - Esquema simplificado do funcionamento do protocolo DHCP.
Fonte: SCHMITT, 2013, p. 111.

Quando um dispositivo é ligado, ele precisa receber os dados de configuração e enviar uma mensagem na rede solicitando a algum servidor de DHCP as informações necessárias para se conectar à rede. Um servidor DHCP responde enviando ao dispositivo os parâmetros de rede. Assim, na prática, quando você conecta seu *smartphone* ou *notebook* em um aeroporto, um hotel ou mesmo em modo móvel em qualquer lugar, há uma troca de mensagens DHCP para que seu dispositivo possa usar a rede.

4.1.1 Funcionamento

O DHCP é um protocolo que segue o paradigma cliente-servidor, em que o cliente é um *host* que chega à rede para obter informações sobre a configuração da rede. Pode ser que cada sub-rede tenha um servidor DHCP, mas também pode ser que esse servidor não exista e, então, o agente DHCP, que geralmente é um roteador, sabe o endereço de um servidor DHCP para essa sub-rede. A figura a seguir apresenta um cenário cliente-servidor DHCP.

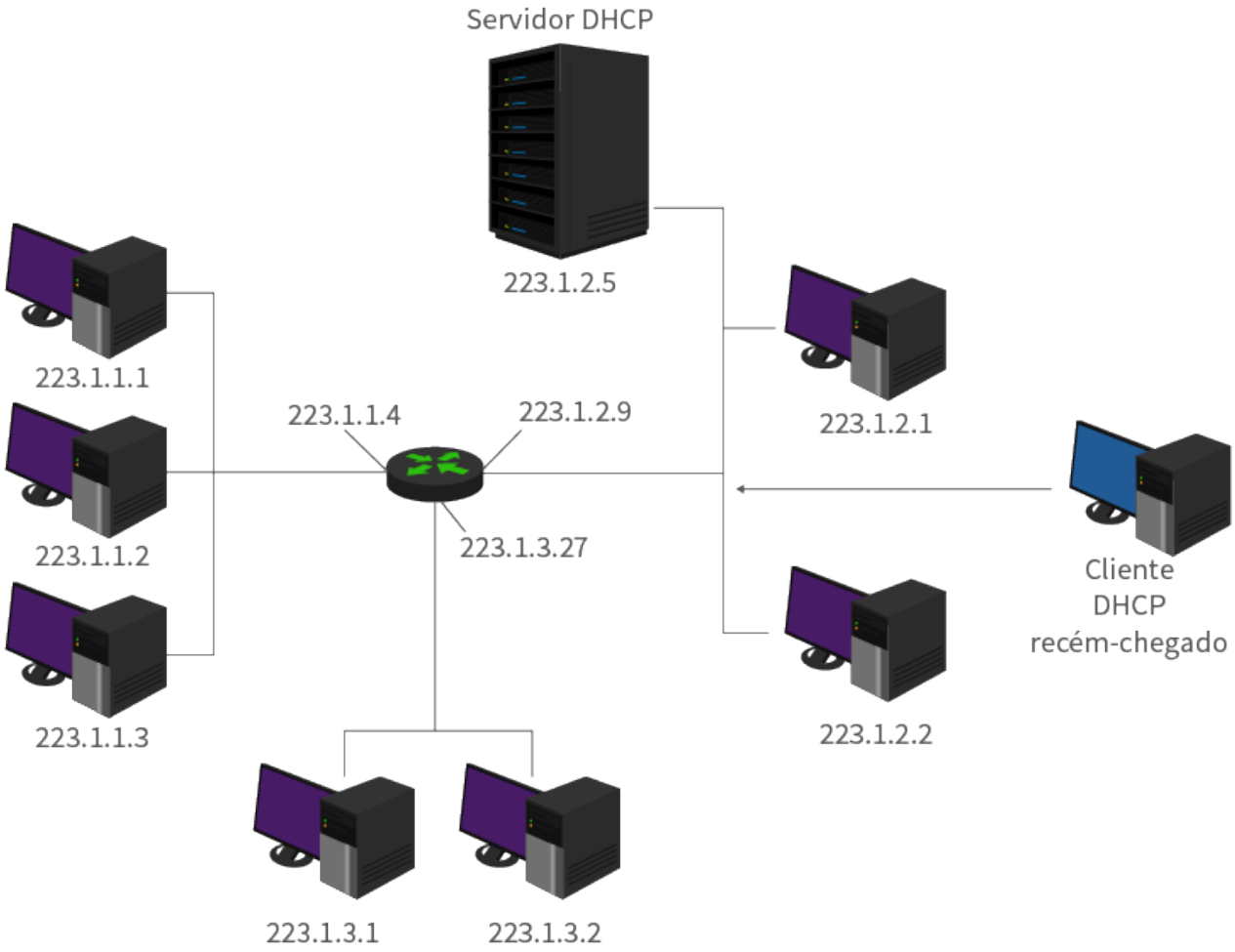

Figura 2 - Cenário cliente-servidor DHCP.
Fonte: KUROSE; ROSS, 2013, p. 256.

Veja que na figura um servidor DHCP está conectado à rede 223.1.2/24, servindo o roteador de agente de repasse para clientes que chegam conectados às sub-redes 223.1.1/24 e 223.1.3/24. Na descrição do cenário a seguir, vamos considerar que um servidor DHCP está disponível na sub-rede.

VOCÊ QUER VER?

Vamos relembrar como funciona o endereçamento IP com CIDR para representar as máscaras? Veja na videoaula (2017) a seguir o que significa a representação CIDR para entender por que se utiliza endereços com /24: https://www.youtube.com/watch?v=igUcU0SFDeI. Aproveite e veja outros exemplos de uso do CIDR, acessando a parte 1 (2013): https://www.youtube.com/watch?v=Xr6gLTENiBU e a parte 2 (2013): https://www.youtube.com/watch?v=0MhAZfu7YGk.

Quando o *host* chega à sub-rede, o DHCP funciona em um processo de quatro etapas, como mostra a figura a seguir, sobre a interação entre cliente e servidor DHCP, onde a palavra "Internet" significa o endereço do *host* na internet e indica que o endereço está sendo alocado para um cliente que recém chegou.

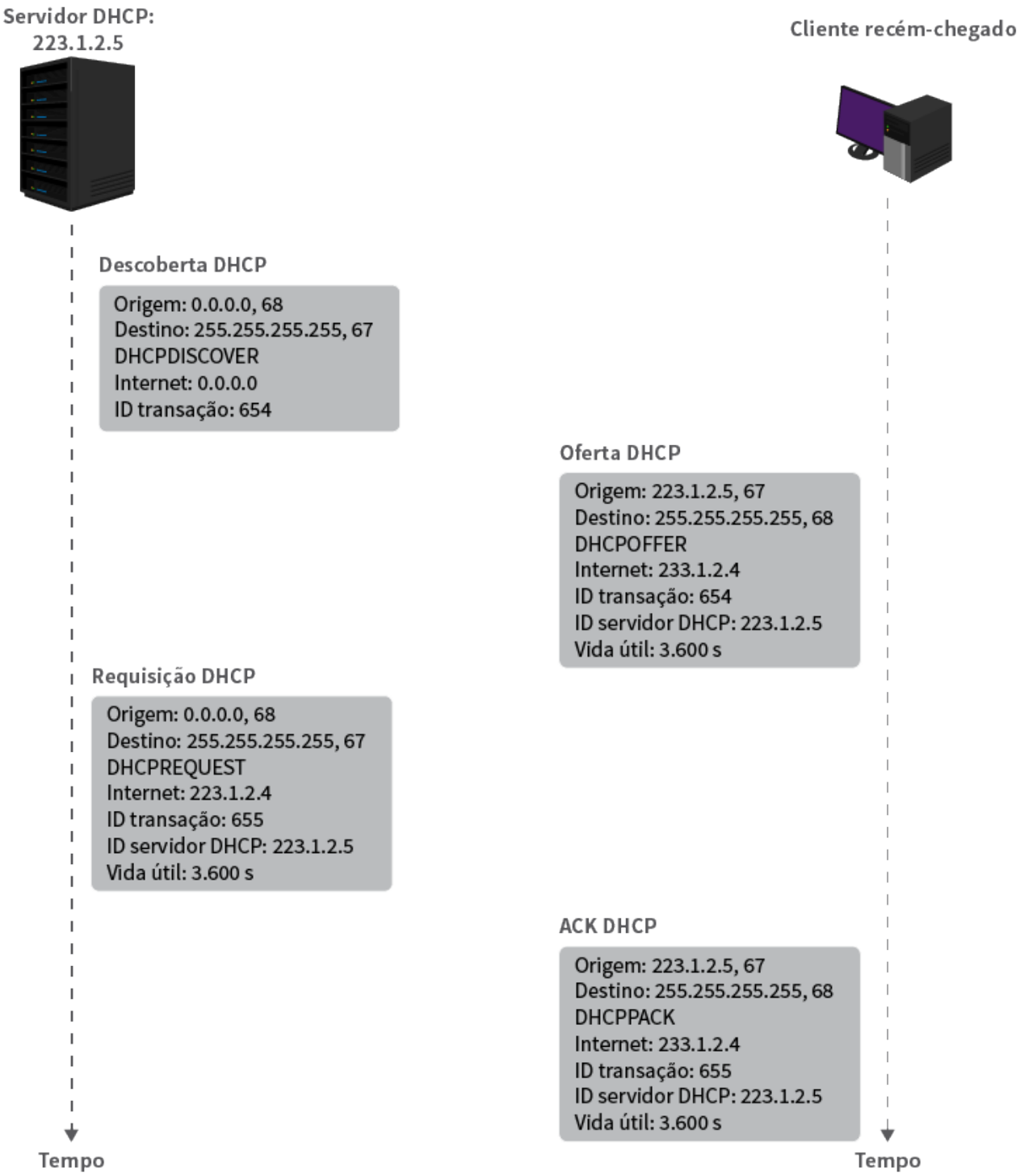

Figura 3 - Cenário cliente-servidor DHCP.
Fonte: KUROSE; ROSS, 2013, p. 256.

Desse modo, o processo de funcionamento do DHCP consiste nas seguintes quatro etapas, segundo Kurose e Ross (2013). Clique na interação a seguir para conhecê-las.

descoberta do servidor DHCP: a primeira tarefa do *host* que chega à rede ou sub-rede é encontrar um DHCP com quem possa interagir. Oficialmente, é transmitida uma mensagem de descoberta DHCP que o cliente envia dentro de um pacote UDP para a porta 67 e que é envolvido em um datagrama IP. Como o *host* não sabe o endereço IP da rede em questão e nem o endereço de um servidor DHCP para essa rede, o cliente DHCP envia um datagrama IP que contém a sua mensagem de descoberta DHCP com o endereço IP de destino de *broadcast* (ou

difusão) 255.255.255.255 e um endereço IP destinatário 0.0.0.0. Assim, o cliente DHCP transmite o datagrama IP por *broadcast* para a camada de enlace que, então, transmite esse quadro para todos os nós conectados à sub-rede;

oferta(s) dos servidores DHCP: um servidor recebe a mensagem de descoberta DHCP do cliente e a responde com uma mensagem de oferta DHCP, que é transmitida por *broadcast* para todos os nós presentes na sub-rede através do endereço de IP de transmissão 255.255.255.255. O cliente pode escolher dentre muitas ofertas, pois diversos servidores DHCP podem estar disponíveis. Cada mensagem de oferta do servidor contém o ID da transação da mensagem de descoberta recebida, o endereço IP proposto para o cliente, a máscara de rede e o tempo de concessão do endereço IP (tempo de *lease).* É comum o servidor definir o tempo de concessão para várias horas ou dias;

solicitação DHCP: o cliente que recém chegou à rede deve escolher entre uma ou mais ofertas do servidor e responder à oferta selecionada com uma mensagem de solicitação DHCP, repetindo os parâmetros de configuração;

DHCP ACK: o servidor responde, então, à mensagem de requisição DHCP com uma mensagem DHCP ACK confirmando os parâmetros requisitados.

Quando o cliente recebe o DHCP ACK, a interação é finalizada e ele poderá usar o endereço IP alocado pelo DHCP durante o tempo de concessão. Lembre-se de que se o tempo de concessão terminar, a configuração DHCP vai renovar automaticamente ou negar a concessão.

Desse modo, quando o dispositivo envia mensagem para descoberta do servidor, embora a mensagem seja para todos os dispositivos na rede, apenas um servidor DHCP melhor habilitado responde.

Mensagens

Vamos especificar as mensagens que são trocadas pelo protocolo DHCP para realizar uma conexão na rede, conforme Schmitt (2013):

- DHCPDISCOVER: *broadcast* enviado pelo cliente para encontrar o servidor;
- DHCPOFFER: mensagem enviada pelo servidor oferecendo seus serviços para o cliente;
- DHCPREQUEST: mensagem enviada pelo cliente para o servidor, solicitando, confirmando ou renovando um empréstimo dos parâmetros;
- DHCPACK: mensagem do servidor com as configurações DHCP;
- DHCPRELEASE: mensagem do cliente liberando os parâmetros emprestados;
- DHCPINFORM: mensagem do cliente requisitando informações de configuração de rede.

A seguir, é descrito o formato dessas mensagens. Acompanhe!

4.1.2 Formato de mensagem DHCP

Segundo Comer (2016), por ter sido criada como uma extensão do BOOTP, a versão IPv4 do DHCP definiu uma versão um pouco diferente do cabeçalho BOOTP, como mostra a figura abaixo.

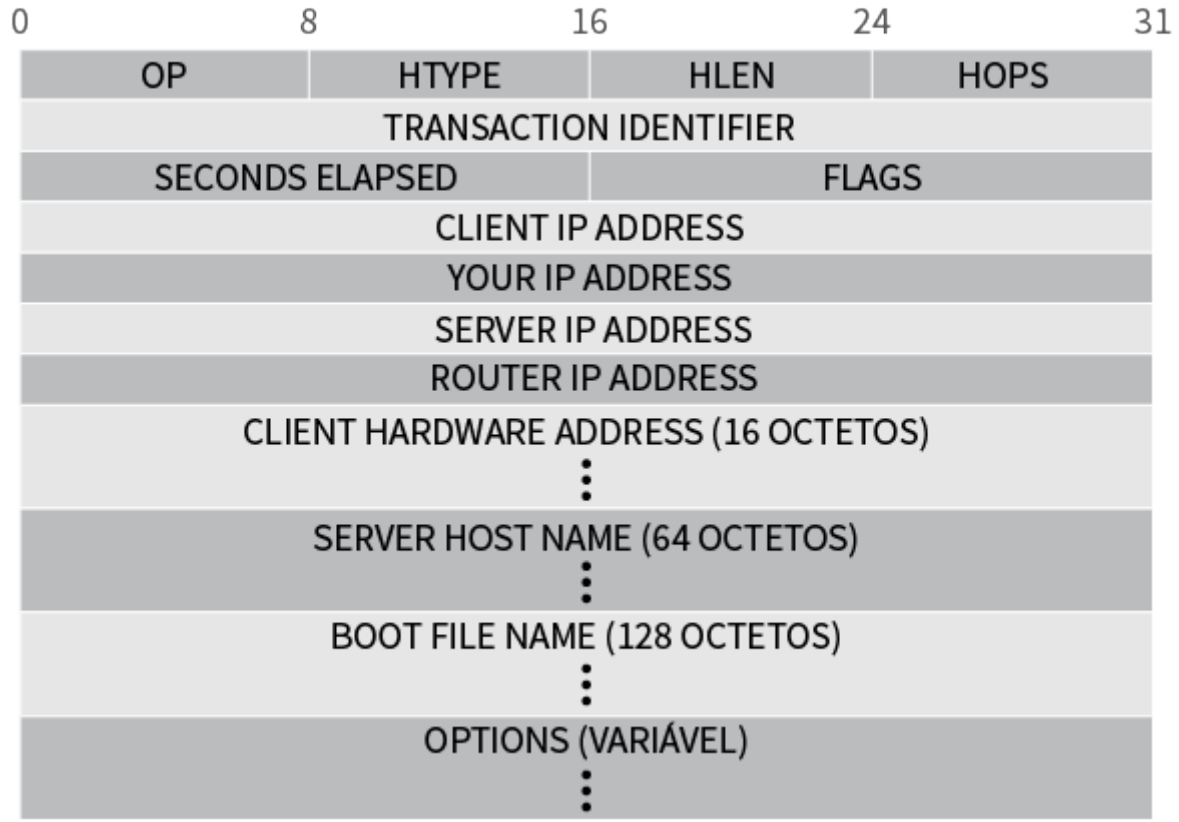

Figura 4 - O formato do cabeçalho DHCP com endereços IPv4.
Fonte: COMER, 2016, p. 353.

Para o autor, cada campo em uma mensagem DHCP tem um tamanho fixo, com exceção do campo OPTIONS. Os sete primeiros campos contêm informações para processar a mensagem. O campo OP diz se a mensagem é uma requisição no sentido cliente-servidor ou uma resposta no sentido servidor-cliente. O campo OPTIONS especifica o tipo entre as várias mensagens existentes, como, por exemplo, se o cliente possui mensagens para descobrir servidores ou para solicitar um endereço, e se o servidor possui mensagens para reconhecer ou negar um pedido. Os campos HTYPE e HLEN ditam o tipo de rede e o comprimento do seu endereço de *hardware*. Já um cliente utiliza o campo FLAGS para especificar se ele pode receber mensagens *broadcast* ou respostas dirigidas. O campo HOPS vai especificar quantos servidores encaminharam a requisição, enquanto o campo TRANSACTION IDENTIFIER fornece um valor que um cliente pode usar para determinar se uma resposta corresponde ao seu pedido. Já o campo SECONDS ELAPSED vai especificar quantos segundos se passaram desde que uma máquina iniciou. Por fim, caso a máquina conheça o seu endereço IP (usando outro mecanismo que não o DHCP), ela preenche o campo CLIENT IP ADDRESS na requisição (COMER, 2016).
Os campos utilizados para resposta às máquinas requisitantes são apresentados ao final. Quando se concede uma extensão de tempo, o servidor DHCP retorna um endereço IPv4 no campo YOUR ADDRESS. O DHCP também utiliza o campo OPTIONS para retornar a máscara de endereço e o endereço do roteador ou *gateway* padrão (COMER, 2016).

4.1.3 Acesso indireto ao servidor DHCP por meio de relay

Aprendemos que a requisição DHCP para encontrar um servidor é transmitida em *broadcast*. Embora isso aconteça, o DHCP não exige que cada rede individual tenha um servidor. O que pode acontecer é existir um agente de *relay* que encaminha requisições e respostas entre um cliente e o servidor, caso este esteja em uma

rede diferente. Assim, ao menos um agente de *relay* precisa estar presente em cada rede, sendo configurado com o endereço do servidor DHCP. Desse modo, quando o servidor responde, o agente fica encarregado de encaminhar a resposta para o cliente (COMER, 2016).

Os administradores de rede preferem gerenciar diversos agentes de *relay*, pois, primeiramente, em uma rede com um servidor DHCP e vários agentes, a administração é centralizada em um único dispositivo. Com isso, o administrador de rede não necessita interagir com diversos dispositivos para alterar a política de extensão de tempo dos endereços IP ou determinar o *status* atual do servidor. Em segundo lugar, diversos roteadores comerciais contêm um modo de fornecer serviço de relay DHCP em todas as redes as quais o roteador está conectado. No mais, a ativação de um agente de *relay* em um roteador é de fácil manuseio, através da configuração da permissão de encaminhamento e especificação do endereço do servidor DHCP, o que dificilmente é mudado (COMER, 2016).

VAMOS PRATICAR?

Em universidades, a quantidade de computadores, *smartphones* ou *tablets* é grande, o que exige claramente uma configuração dinâmica para que o trabalho da equipe de TI seja agilizado e todos os dispositivos consigam navegar na rede sem interrupções em cada local que se desloquem. À medida que os *hosts* entram e saem da sub-rede, o servidor DHCP precisa atualizar sua lista de endereços IP disponíveis, assim, automaticamente poderão ser atribuídos os IPs para os dispositivos cada vez que mudarem de local, isto é, de sub-rede.

Vamos supor que você é um administrador de redes e precisa configurar um servidor DHCP de acordo com as especificidades da empresa em que trabalha. Pesquise como configurar um servidor DHCP, de acordo com o sistema operacional utilizado no servidor da empresa, que pode ser Windows, Linux ou MAC.

Apesar de citarmos quatro parâmetros fundamentais para que um dispositivo utilize a rede, sendo endereço IP, máscara, *gateway* padrão e endereço do servidor DNS, existem também outros elementos que podem ser distribuídos através do DHCP, dentre eles estão o nome do servidor de *logs*, servidor NTP e o servidor *proxy*, sendo este último nosso assunto do próximo tópico.

4.2 Servidor PROXY: conceito e aplicação

A palavra *proxy* pode ser traduzida como procuração ou procurador. Sendo assim, dizemos que uma aplicação cliente utiliza um *proxy* que passa uma procuração para que este servidor realize uma tarefa intermediária de conexão com o servidor de aplicação, além de realizar suas tarefas e repasse o resultado para o cliente. O *proxy* opera na camada de aplicação do modelo TCP/IP, isto é, realiza suas tarefas sobre os protocolos de aplicação. Desse modo, o *proxy* é uma aplicação, um *software* instalado em um sistema operacional, assim como um servidor HTTP, DNS etc. (SCHMITT, 2013).

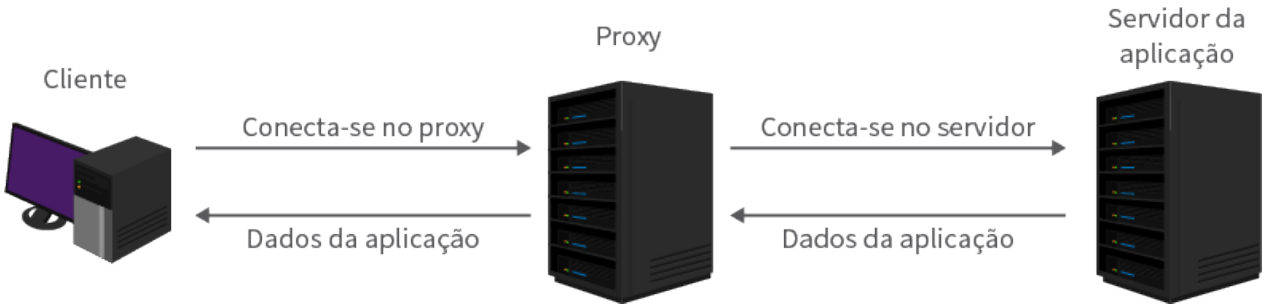

Figura 5 - Funcionamento de um servidor proxy.
Fonte: SCHMITT, 2013, p. 146.

A figura acima mostra a interação entre cliente, servidor *proxy* e servidor de aplicação. Para que essa interação e comunicação funcione, o *proxy* atua como intermediador.

CASO

Quando você recebe um *e-mail*, ele pode passar por um servidor *proxy* para que algumas informações sejam filtradas, como, por exemplo, os *spams*. Muitas vezes os *spams* vão direto para um diretório (pasta) específica no seu *e-mail* e não chega à sua caixa de entrada. Isso acontece porque houve uma filtragem através de um servidor *proxy*. Nesse caso, um *proxy* de *e-mail*, que funciona como anti-*spam*, está entre o servidor SMTP (recebendo o *e-mail* da internet) e sua caixa de *e-mail*. Nesse caso, quando o servidor SMTP receber um *e-mail*, ele repassa para o *proxy* SMTP e depois devolve para o servidor SMTP o *e-mail* classificado e filtrado para, então, chegar ao destino, a sua caixa de entrada. Se o servidor *proxy* classificar como *spam*, o *e-mail* vai direto para o diretório adequado e poupa seu tempo de excluir a mensagem, pois os *e-mails* na pasta de *spam* são excluídos automaticamente após alguns dias. No entanto, pode acontecer de *e-mails* desejados sejam filtrados como *spam*, pois você pode estar esperando um *e-mail* de um destinatário não comum. Nesse caso, você deve sempre verificar sua caixa de *spams*.

Vamos considerar que, para realizar uma classificação de *e-mails*, o *proxy* SMTP seja capaz de interpretar os cabeçalhos SMTP. Segundo Schmitt (2013), isso significa que o *proxy* deve saber qual IP do servidor que está enviando o *e-mail*, qual o domínio desse servidor, quem é o remetente, qual o assunto do *e-mail* e quais são os dados do corpo do *e-mail*. Sendo assim, o *proxy* é específico da camada de aplicação, não sendo capacitado a intermediar uma comunicação HTTP.

Um *proxy* utilizado para mensagens instantâneas, ou conversas simétricas, tal como através dos aplicativos WhatsApp e Messenger, pode salvar todas as mensagens trocadas entre os usuários. Atualmente, essas mensagens podem ser protegidas por criptografia enquanto são transferidas, no entanto, por existir a capacidade de acessá-las para análises posteriores. Assim, se o *proxy* conseguir interpretar os cabeçalhos e analisar os dados do nível de aplicação, qualquer aplicação poderá ter um.

VOCÊ SABIA?

É muito comum utilizar servidores *proxy* juntamente com filtros de pacotes na implementação de estruturas de *firewall*. Dessa maneira, o *proxy* é usado como elemento de filtragem de conteúdo de aplicação para implementação da política de segurança de uso na internet.

O servidor *proxy*, também conhecido como cache ou *proxy*-cache, é muito conhecido por trabalhar sobre o acesso às páginas *web* como um armazenador temporário de arquivos. O *proxy* atua também como protetor de rede privada aos acessos indevidos às informações contidas nos servidores e nos dispositivos. Com o *proxy*, é possível administrar a autenticação de usuários, registrar os acessos à rede, além de extrair relatórios gerenciais sobre a utilização da rede de internet pelos usuários. Já o *firewall*, nesse contexto, possibilita controlar o acesso externo à rede interna.

Com a configuração correta do servidor *proxy* e do *firewall*, podemos acessar páginas *web* mais rapidamente, economizando banda do *link* da organização, por exemplo. Isso porque a informação é lida por meio do servidor e então salva de forma local no diretório de cache. Assim, quando há outro acesso a essas informações, estas serão obtidas localmente. Além disso, outra vantagem é que ao promover a segurança, o servidor *proxy* permite bloquear tentativas de acesso à rede privada por quem não é autenticado, e essas tentativas são registradas.

Podemos dizer que o *proxy* é um servidor que repassa as requisições da aplicação cliente para um servidor. Sendo assim, um servidor *proxy* pode mudar a solicitação do cliente ou a resposta do servidor.

Portanto, o *proxy* é usado para justificar a necessidade de diminuição do tráfego de rede através da criação de caches temporárias de objetos acessados por vários dispositivos na rede. Além disso, é responsável por filtrar dados para aplicação de políticas de segurança, análise de dados, adaptação ou transformação de dados e realiza tarefas de manipulação dos dados entre o cliente e o servidor de uma aplicação.

VAMOS PRATICAR?

A rede de uma universidade é uma rede local que se conecta à rede externa por um servidor intermediário. Deve haver uma configuração no dispositivo do aluno ou alguém autorizado para que seja possível acessar à internet, pois apenas alunos, professores e funcionários podem fazer isso, para evitar que outras pessoas acessem à rede de internet na universidade e provoquem a saturação do tráfego na rede. Essa configuração é feita através do servidor *proxy*.

Como um administrador de rede, pesquise, identifique e analise como configurar uma autenticação de usuário, mantendo, além do controle de tráfego, também a segurança de acesso à internet na universidade e a garantia da identificação de quem acessa a rede.

Também temos o conceito de *proxy* transparente, que se refere a uma técnica desenvolvida para obrigar os usuários de determinada rede a utilizarem o *proxy*. Assim, é possível impor regras para utilização da rede, de modo que seja possível resgatar dados estatísticos. O termo transparência é utilizado porque dizemos que o

proxy intercepta o tráfego na *web*, fazendo com que a conexão dos usuários esteja sempre condicionada a determinadas regras de utilização da rede. O RFC 3040 explica detalhadamente como funciona o *proxy* transparente.

Uma outra aplicação do *proxy* é a configuração do *proxy* reverso, que se trata de um servidor "burro", que é conceituado dessa forma porque apenas recebe as solicitações e as delega ou faz tarefas simples, como devolver uma página *web* já pré-processada. Assim, ele é dito "burro" porque não tem conhecimento para responder uma solicitação do cliente de forma completa. No entanto, isso não significa que ele não seja útil.

Imagine que existe um servidor *proxy* reverso e diversos outros servidores *web* atrás deste servidor *proxy* reverso. O servidor *proxy* reverso repassa as solicitações para os servidores *web* adequados a respondê-las. Isso é útil para dividir as tarefas entre servidores. Outra utilidade é como um dispositivo de segurança como proteção para servidores HTTP em uma intranet de uma organização, por exemplo, protegendo, assim, os servidores da intranet contra invasores advindos da internet. Essa segurança é fornecida pelo fato de o servidor *proxy* fornecer um único ponto de acesso a todos os servidores HTTP da intranet.

O tipo mais comum de *proxy* é o HTTP, discutido no tópico a seguir.

4.2.1 Proxy HTTP

O tipo mais comum de *proxy* é o HTTP ou também conhecido como *proxy web*, em que o cliente se conecta ao *proxy* e repassa a URL que contém o objeto que deseja obter do mesmo modo que faria um servidor *web*. Assim, o *proxy* se conecta ao servidor *web* e faz a requisição desse objeto, repassando a requisição do navegador cliente. Depois que o objeto é recebido, o *proxy* repassa esse objeto para o navegador (SCHMITT, 2013).

O *proxy* HTTP fornece um cache de objetos para todos os dispositivos da rede, com o objetivo de minimizar o tráfego. Além disso, outro objetivo é fornecer um mecanismo de segurança na implementação da política de segurança para acesso às páginas *web*.

Caso o objeto requisitado esteja no cache, o navegador precisa ter certeza de que o objeto está atualizado, isto é, verificar se todos os arquivos no *site* requisitado estão exatamente como nos arquivos armazenados em cache. Isso é possível porque o sistema operacional grava um conjunto de dados que estão vinculados a esse arquivo, tais como data e hora de gravação ou modificação do arquivo. Tanto o arquivo gravado no servidor *web* como os armazenados em cache tem informações de hora e data de gravação.

No entanto, é preciso observar que a data e hora de gravação do arquivo no cache do navegador será sempre posterior à data e hora do arquivo no servidor *web*, afinal, o arquivo foi gerado e gravado no servidor antes de estar disponível no navegador. Por isso, para saber, de fato, se um arquivo é atual, isto é, igual ao arquivo que está no servidor *web,* o navegador deve requisitar ao servidor *web* se a data de gravação do arquivo é posterior à data do arquivo local, para, então, saber se o arquivo foi modificado após o acesso e gravação no cache local. Assim, se o arquivo no cache foi gravado ou modificado em data posterior ao mesmo arquivo no servidor, o navegador saberá que o arquivo em cache é atualizado. Caso contrário, o navegador deve obter o arquivo atualizado.

Essa verificação de atualização de um arquivo é feita utilizando uma área de variáveis das requisições HTTP, segundo Schmitt (2013). Um campo chamado *if-modified-since* (traduzido como "se modificado após"), é enviado ao servidor, conforme mostra a figura a seguir.

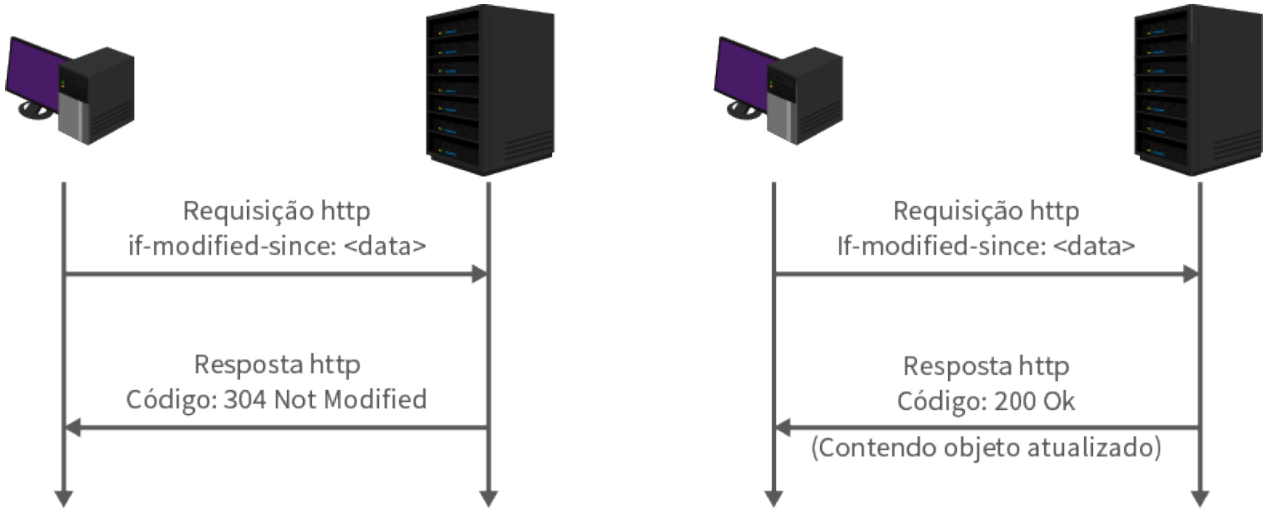

Figura 6 - Requisição de arquivos com verificação de atualização.
Fonte: SCHMITT, 2013, p. 149.

De acordo com a figura acima, podemos verificar duas possíveis respostas a uma requisição com o campo *if-modified-since.* Clique nos itens para conhecê-las.

Resposta com o campo not-modified que indica que o arquivo não foi alterado.
Resposta que contém o arquivo, que indica que ele foi alterado.

Para agilizar o processo, a pergunta *if-modified-since* é ignorada pelo servidor *web* caso o arquivo tenha sido modificado. O arquivo é então enviado da mesma forma que uma requisição comum. Esse mecanismo de cache faz com que o seu dispositivo economize tráfego de rede, mas quando falamos de servidor *proxy*, este atende a todos os dispositivos da rede. Um *proxy* funciona na medida em que os navegadores encaminham suas requisições HTTP para o *proxy*. Quando o *proxy* recebe essas solicitações, ele verifica em seu cache se possui o arquivo e, caso não possui, vai procurar no servidor *web* indicado na solicitação e, então, resgata o objeto pela primeira vez ou atualizado. Isso significa que o servidor *proxy* manterá uma cópia do arquivo solicitado em seu cache. Do mesmo modo, se o arquivo requisitado estiver no cache do *proxy*, deverá ser feita a verificação se este é atualizado, através do campo *if-modified-since*. Ao acessar uma página *web*, o navegador faz com que o *proxy* mantenha uma cópia dos arquivos da página em cache. Caso outro dispositivo na rede requisite a mesma página *web*, o *proxy* poderá realizar a análise e entregar os arquivos ao navegador.
Desse modo, em relação ao cache, podemos concluir que o servidor *proxy* se comporta da mesma maneira que um navegador para armazenar um cache local. Entretanto, o servidor *proxy* atende à rede inteira. O cache utiliza algoritmos de expiração para remover arquivos de acordo com o tempo que foram acessados. Exemplos são os algoritmos *Least Recently Used* (LRU) e *Least Frequently Used* (LFU). O LRU extrai os arquivos que estão há mais tempo sem ser usados e o LFU extrai os documentos que são usados com menos frequência (KUROSE; ROSS, 2013).

Síntese

Nesta unidade, você aprendeu que um servidor DHCP é importante para que seu dispositivo possa acessar uma rede de forma automática, o que se tornou uma necessidade com as redes *Wi-Fi*. Esse trabalho manual seria inviável, pois seria necessário que um profissional configurasse cada dispositivo toda vez que ele mudasse de rede, muitas vezes em um mesmo ambiente. Além do endereço IP, o servidor DHCP provê a máscara de rede, o

endereço do *gateway* padrão (*default*) e endereço do servidor de nomes (DNS). Você também constatou que o servidor *proxy* tem a função de agir como intermediário entre clientes e servidores de aplicações de rede. Ele pode armazenar temporariamente arquivos das páginas *web* para que o acesso a eles seja gravado localmente, evitando o grande tráfego de arquivos na internet. Ele também pode administrar a autenticação de usuário e proteger sua rede privada, dentre outras funções.

Nesta unidade, você teve a oportunidade de:

- compreender o servidor DHCP como o responsável por conectar seu dispositivo a uma nova rede (ou sub-rede) e obter um endereço IP de forma automática, sem interromper sua navegação;
- analisar como as requisições da aplicação cliente passam pelo *proxy* antes de chegarem ao servidor e as respostas passam pelo *proxy* antes de chegar ao cliente, constatando quais as funções do servidor *proxy* nesse contexto.

Bibliografia

ALEXANDER, S.; DROMS, R. **RFC 2132** – DHCP Options and BOOTS Vendor Extensions. Silicon Graphics, Inc; Bucknell University, 1997. Disponível em: https://tools.ietf.org/pdf/rfc2132.pdf. Acesso em: 13 ago. 2019.

BARRETT, D. **Redes de computadores**. Rio de Janeiro: LTC. 2010.

COMER, D. E. **Redes de computadores e internet**. 6. ed. Porto Alegre: Bookman, 2016.

DROMS, R. **RFC 2131** – Dynamic Host Configuration Protocol. Bucknell University, 1997. Disponível em: https://tools.ietf.org/pdf/rfc2131.pdf. Acesso em: 13 ago. 2019.

EXEMPLO de uso de endereços CIDR – Parte 1. 2013. 1 vídeo (14 min 47 s). Publicado no canal Othon Batista. Disponível em: https://www.youtube.com/watch?v=Xr6gLTENiBU. Acesso em: 13 ago. 2019.

EXEMPLO de uso de endereços CIDR – Parte 2. 2013. 1 vídeo (10 min 08 s). Publicado no canal Othon Batista. Disponível em: https://www.youtube.com/watch?v=0MhAZfu7YGk. Acesso em: 13 ago. 2019.

KUROSE, J. F.; ROSS, K. W. **Redes de computadores e a internet:** uma abordagem top-down 6. ed. São Paulo: Pearson Education do Brasil, 2013.

O QUE é a conotação CIDR? /24? /22? – LPIC-2. 2017. 1 vídeo (4 min 49 s). Publicado no canal Linux4Fasters. Disponível em: https://www.youtube.com/watch?v=igUcU0SFDeI. Acesso em: 13 ago. 2019.

SCHMITT, M. A. R. **Rede de computadores:** nível de aplicação e instalação de serviços. Porto Alegre: Bookman, 2013.