SERVIÇOS DE REDES DE COMPUTADORES

UNIDADE 2 - SERVIDORES NTP E DNS E PROTOCOLOS HTTP/HTTPS: O QUE SIGNIFICAM EM UMA REDE DE COMPUTADORES?

Aline Izida

Introdução

Você já parou para pensar como o seu computador ou *smartphone* sabe definir a hora correta? De onde eles buscam essa informação o tempo todo? Nesta unidade, você entenderá como isso é feito em servidores através do protocolo NTP. Assim, você saberá que é um método fundamental para o funcionamento da rede mundial de computadores e constatará que essa tarefa não é tão simples quanto pode parecer. Lembrando que esse sistema de sincronização de tempo deve funcionar 24 horas por dia e 7 dias por semana. Os benefícios da utilização do NTP são tanto para os usuários como para os administradores de rede.

Você também já analisou a relação entre o nome do *site* que você digita no seu navegador *web* e os endereços de IP? Afinal, você já aprendeu que os endereços de IP identificam unicamente cada dispositivo conectado à rede, no entanto, você não tem contato direto com ele, não é mesmo? Isso porque existe o servidor DNS, responsável por possibilitar que o nome digitado seja traduzido no endereço IP correspondente. Seria muito complicado decorar tantos endereços IP, concorda?

Ademais, você já ouviu falar que se deve pagar a hospedagem de um *site*, senão ele não fica disponível para acesso? Vamos abordar nesta unidade como o protocolo HTTP funciona e como ele interage com os servidores HTTP, isto é, os servidores *web* que hospedam páginas *web*. Desse modo, ao final desta unidade, você vai aprender como funciona o HTTP e qual a diferença para o HTTPs e entender qual a função do servidor *web* nesse contexto.

Aproveite o conteúdo e ótimos estudos!

1.1 Servidores NTP: importância do controle central de hora

O *Network Time Protocol* (NTP) ou Protocolo de Tempo para Redes é o protocolo padrão na internet utilizado para sincronizar relógios em dispositivos de redes, tais como servidores, computadores, roteadores, dispositivos conectados à rede. O NTP utiliza o *Coordinated Universal Time* (UTC), o qual define o fuso horário de referência para os demais. No modelo TCP/IP, o NTP atua na camada de aplicação e foi idealizado por David L. Mills, sendo até hoje mantido por ele e demais especialistas. Assim, os servidores NTP garantem a sincronização da hora correta dos relógios dos dispositivos em rede (NTP.BR, 2019).

VOCÊ QUER LER?

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 documento RFC 1305 (MILLS, 1992) define o NTP versão 3, de 1992, e você pode acessá-lo em: https://tools.ietf.org/pdf/rfc1305.pdf. Já o RFC 5905 (MILLS et al., 2010) define a versão 4, de 2010, e está disponível em:

<https://tools.ietf.org/pdf/rfc5905.pdf>

Na prática, uma vez que o administrador da rede configura os servidores com o protocolo NTP, todos dispositivos conectados na rede sincronizam a hora automaticamente. Desse modo, o administrador da rede não precisará configurar cada dispositivo de forma manual para acertar a hora. Na teoria, pode parecer simples o fato de consultar o servidor e ajustar o relógio local de acordo com um intervalo de tempo, contudo, isso envolve muitos fatores, pois existem diversos servidores que fornecem o tempo e isso precisa ser administrado conforme a exatidão de horários, segurança, dentre outros fatores.

Vamos supor que você precisa entregar um trabalho às 23:59 no *site* do seu curso a distância. Caso o servidor do qual seu dispositivo é cliente ou servidor do *site* do seu curso não estiver implementando o protocolo NTP, provavelmente haverá conflitos de horários e um minuto pode fazer diferença. Por exemplo, enquanto o relógio da máquina que hospeda o *site* do seu curso marca 00:00, o relógio do seu dispositivo pode marcar 23:59. Se você deixou para o último minuto, pode perder o prazo da entrega do seu trabalho.

Qual seria a solução para que esse tipo de problema não aconteça? Claro que pode não ser possível que você ajuste o servidor da sua universidade, mas quem sabe pode mandar um *e-mail* para o setor de TI para saber se eles configuraram o servidor de acordo com as horas do projeto NTP. Além disso, você pode configurar no seu computador seguindo os tutoriais, conforme seu sistema operacional em utilização:

Windows <https://ntp.br/guia-win-comum.php>

Linux <https://ntp.br/guia-linux-comum.php>

MAC <https://ntp.br/guia-mac.php>

Se você configurar seu dispositivo e deixar a sincronia de horário de acordo com o projeto NTP, e aumenta a possibilidade de garantia de que todos os dispositivos estejam sincronizados com a mesma hora.

VOCÊ QUER VER?

No site da NTP.br é disponibilizado, além de diversas informações sobre o NTP, um vídeo (MOREIRAS, 2010) sobre a importância e o funcionamento do sincronismo de tempo na internet. Você vai entender como e os motivos pelos quais o NTP foi idealizado. Acesse em: http://www.zappiens.br/portal/VisualizarVideo.do?_InstanceIdentifier=0&_EntityIdentifier=cgiFRWd-V3ZGGRqao9GOvyXEQXLYfpqFgvmOU-1_2tcChk.&idRepositorio=0.

Dessa forma, o NTP é importante à medida que possibilita o correto funcionamento de sistemas e redes; apoia processos de detecção e tratamento de incidentes de segurança; é essencial para a documentação e preservação de evidências uteis para realizar investigações de crimes envolvendo informática. Vale ressaltar que o NTP precisa de 4 a 7 referencias de tempo para ser confiável (NÚCLEO DE INFORMAÇÃO E COORDENAÇÃO DO PONTO BR, 2010).

1.1.1 Arquitetura

Uma topologia hierárquica forma os servidores NTP, que é dividida em camadas ou os chamados estratos, que são numerados de zero a 16. No entanto, o estrato 0 (*stratum* 0) representa a referência primária de tempo, portanto, não faz parte da rede de servidores NTP, sendo um receptor do Sistema de Posicionamento Global (GPS) ou um relógio atômico. O estrato 16 também apresenta uma particularidade, ele representa um servidor que está inoperante (NTP.BR, 2019).

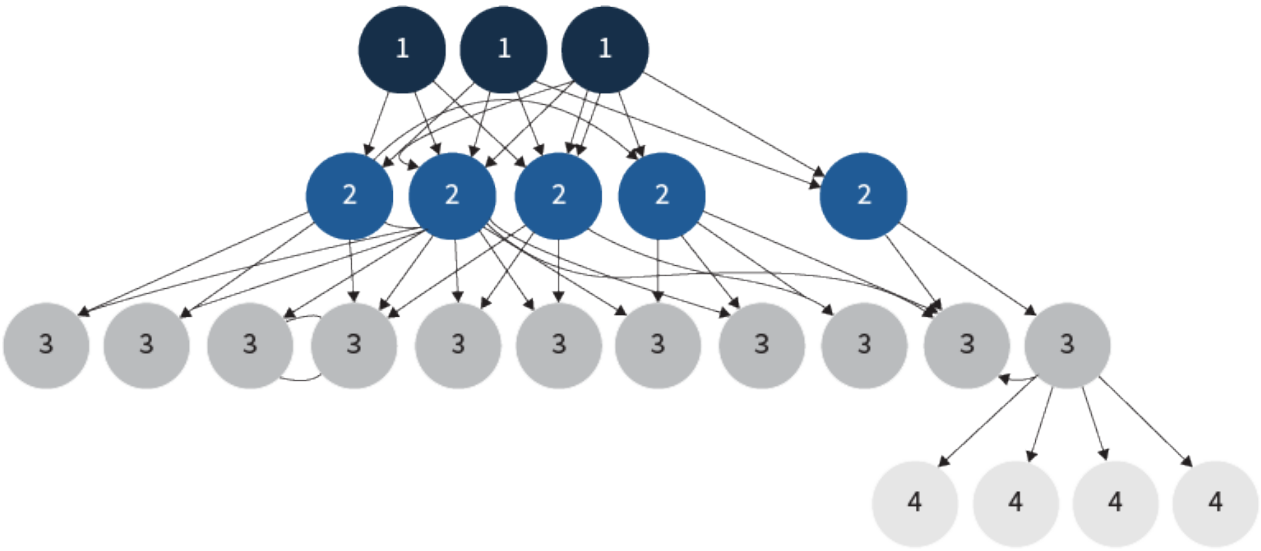

Figura 1 - Arquitetura do NTP: topologia hierárquica em camadas (estratos).
Fonte: NTP.BR, 2019, on-line.

Funciona como uma cascata, ao passo que o estrato 0 fornece o tempo correto para o estrato 1, enquanto o estrato 1 fornece o tempo correto para o estrato 2 e assim por diante. Considerando a topologia, como mostra a figura apresentada, como uma árvore, quanto mais o servidor estiver perto da raiz (estrato 0), mais exato é o tempo. Dessa maneira, o NTP é servidor, porque fornece o tempo, e é cliente, pois consulta o tempo. Cada servidor pertence a um estrato, logo, o NTP calcula qual servidor é o melhor para consultar o tempo a partir de qual estrato o servidor pertence, contudo, isso não influencia consideravelmente na diferença de hora, pois outros fatores são considerados, tais como a carga em que os servidores estão submetidos, o atraso da rede. Além disso, se acontecer de o servidor perder a conexão com as fontes de tempo ou qualquer fator que modifique a topologia da rede, o estrato utilizado pode variar (NTP.BR, 2019).
As relações entre os dispositivos NTP são chamadas de associações, podendo ser do tipo permanentes, priorizáveis ou efêmeras, conforme NTP.BR (2019). Clique nos cards para saber mais.

Permanentes

Criadas por uma configuração ou comando e mantidas sempre.

Priorizáveis

Criadas por um comando ou uma configuração. Se houver um servidor melhor ou depois de um tempo, pode ser desfeita essa configuração.

Efêmeras ou transitória

Criadas por solicitação de outro dispositivo NTP, podendo ser desfeitas quando acontece um erro ou após um tempo.

Existem três associações ou modos de operação, sendo cliente-servidor, de modo simétrico e modo *broadcast* ou *multicast*, que são selecionados de acordo com o escopo do serviço, fluxo de valores de tempo e meios de configuração. Todos suportam tanto o IPv4 quanto o IPv6.

O modo cliente-servidor funciona com o cliente solicitando informações sobre o tempo para um servidor, ao passo que envia um pacote, então o servidor responde. Esse processo também é chamado de *pull*, pois o cliente obtém os valores de tempo a partir do servidor. O modo simétrico consiste em dispositivos NTP que são configurados como pares, de modo que um funciona como *backup* para o outro. Cada par trabalha com uma ou mais fontes de relógios de referência, se um dos pares perder esses relógios de referência, ou cessar a operação, os outros pares serão reconfigurados automaticamente para que os valores de tempo fluam dos pares sobreviventes para todos os outros nessa rede (MILLS, 2006). O modo simétrico pode ser ativo ou passivo, conforme NTP.BR (2019). Clique nas abas para conhecê-los.

Ativo

O dispositivo A configura o dispositivo B como seu par (criando dessa forma uma associação permanente). Por sua vez, o dispositivo B também configura o dispositivo A como seu par (também cria uma associação permanente).

Passivo

O dispositivo A configura o dispositivo B como seu par (modo simétrico ativo). Mas o dispositivo B não tem o dispositivo A em sua lista de servidores ou pares. Ainda assim, ao receber um pacote de A, o dispositivo B cria uma associação transitória, de forma a poder fornecer ou receber o tempo de A.

O modo simétrico tem uma particularidade negativa por ser suscetível a ataques de segurança. Uma vez configurado como modo simétrico ativo, um dispositivo intruso pode passar informações de tempo falsas para outro dispositivo, por isso é necessário utilizar técnicas de criptografia. Já o modo de operação *broadcast* ou *multicast* é útil quando se necessita de uma configuração que envolva um ou poucos servidores e muitos dispositivos clientes. Por um curto tempo, cliente e servidor trocam pacotes para conhecer o atraso envolvido e depois o cliente passa a receber apenas pacotes em modo *broadcast* ou *multicast* do servidor (MILLS, 2006).

VOCÊ QUER LER?

Para entender o funcionamento do protocolo NTP, é necessário estudar complexos algoritmos, que escolhem dentre as referências que o NTP está consultando, quais estão certas ou erradas e como é possível calcular o horário local através dessas referências. Você pode estudar esses algoritmos no site NTP.BR (2019), disponível em:

<https://ntp.br/ntp.php>

Cada associação (modo de operação) deve ser utilizada conforme situações específicas. Se confiabilidade e precisão são mais importantes, opera-se no modo cliente-servidor ou simétrico. Caso contrário, podem ser usados os modos *broadcast* ou *multicast*. Se a intenção é implementar redundância, o modo simétrico é o mais indicado. E servidores NTP que provem sincronização apenas a clientes NTP ou que não provem sincronização a outros servidores locais, devem ser configurados no modo cliente-servidor (MILLS, 2006).

1.1.2 Segurança

É preciso que o NTP atenda às quatro propriedades de segurança da informação: integridade, disponibilidade, autenticidade e confidencialidade. A integridade e a disponibilidade são garantidas na medida em que os algoritmos NTP proporcionam isso juntamente com configuração do sistema correta e número suficiente de fontes de tempo com referências primarias independentes.
Já a confidencialidade não é considerada necessária, pois o tempo é uma informação pública, por isso não precisamos escondê-lo e trabalhar com informação de tempo cifrada não é viável por demandar tempo do servidor e do cliente, enquanto o sistema precisa ser muito exato. E quem garante a autenticidade da informação são os algoritmos de criptografia, com métodos de chave simétrica ou chave pública.

Existe um outro modo de manter a hora legal brasileira atualizada de acordo com o projeto NTP. É uma configuração mais eficaz do que a padrão do Windows, que possui, por padrão, uma implementação simplificada do NTP que não mantém a hora legal brasileira atualizada constantemente. Recomenda-se, então, utilizar um programa especializado nisso. Esse *software* utilizado na maioria dos servidores NTP foi desenvolvido por David Mills, criador da primeira RFC desse protocolo. No *site* NTP.br você encontra o passo a passo de configurações avançadas do cliente NTP (NTP.BR, 2019e) no Windows (https://ntp.br/guia-win-avancado.php), MAC (NTP.BR, 2019b) (https://ntp.br/guia-mac-avancado.php), Linux (NTP.BR, 2019c) (https://ntp.br/guia-linux-avancado.php) e em Roteadores (NTP.BR, 2019d) (https://ntp.br /guia-rot.php). Experimente ao menos instalar esse programa específico em seu sistema operacional e compare a hora com a do seu *smartphone*, por exemplo.

A seguir, você verá o funcionamento do servidor DNS que tem ligação direta com os endereços IP para simplificação de acesso a páginas *web*.

1.2 Servidor DNS: funcionamento

O *Domain Name System* (DNS) ou simplesmente Sistema de Nomes de Domínio foi criado para facilitar o acesso aos endereços IP na internet, afinal, lembrar de uma sequência de números IP é nada prático. É o DNS que nos auxilia na transformação do que digitamos em um navegador *web* em um endereço de rede adequado.

CASO

O DNS é utilizado pelas aplicações para traduzir o nome atribuído para um endereço IP, tal como www.google.com.br.

Um modo de descobrir o endereço IP de um nome no sistema operacional Windows, é abrir o prompt de comando e digitar "tracert google.com", sem as aspas, que será mostrada a rota realizada para chegar ao IP do google, que será o último a ser apresentado, isto é, o endereço final. Ou você pode simplesmente digitar o comando "ping" seguido do nome do *site*, por exemplo: "ping google.com". No sistema operacional Linux, esse último método também é válido.

Quem projetou o DNS, em 1984, foi Paul Mockpetris. O DNS é um banco de dados distribuído, isto é, está em diversos servidores diferentes, que guardam tabelas de banco de dados com os nomes e os respectivos endereços IP. Dessa forma, o DNS consiste em servidores e tradutores de nomes. Os tradutores estão localizados na camada de Aplicação do *software* de rede de cada computador habilitado com TCP/IP (BARRETT, 2010).
Veja o que é apresentado na tela do navegador *web* quando o DNS consulta os servidores e não encontra o IP referente ao nome requisitado pelo usuário.

Figura 2 - Mensagem de erro apresentada ao tentar acessar um nome que não existe.
Fonte: Elaborada pela autora, 2019.

Segundo Kurose e Ross (2013), o DNS costuma ser empregado por outras entidades da camada de aplicação, tais como HTTP, SMTP e FTP, para traduzir nomes de *hosts* fornecidos por usuários para endereços IP. Por exemplo, quando um navegador, ou seja, um cliente HTTP, executado no *host* de algum usuário, requisita o URL www.faculdadexyz.com.br. Primeiramente, o endereço IP será obtido e depois o *host* poderá enviar uma mensagem de requisição HTTP ao servidor www.faculdadexyz.com.br. Conforme Kurose e Ross (2013, p. 96), isso ocorre do seguinte modo:

1. A própria máquina do usuário executa o lado cliente da aplicação DNS.
2. O navegador extrai o nome de hospedeiro, www.someschool.edu, do URL e passa o nome para o lado cliente da aplicação DNS.
3. O cliente DNS envia uma consulta contendo o nome do hospedeiro para um servidor DNS.
4. O cliente DNS por fim recebe uma resposta, que inclui o endereço IP correspondente ao nome de hospedeiro.
5. Tão logo o navegador receba o endereço do DNS, pode abrir uma conexão TCP com o processo servidor HTTP localizado na porta 80 naquele endereço IP.

Vemos, por esse exemplo, que o DNS adiciona mais um atraso às aplicações de internet que o usam. Felizmente, o endereço IP procurado quase sempre está no *cache* de um servidor DNS "próximo", o que ajuda a reduzir o tráfego DNS na rede, bem como o atraso médio do DNS.

VOCÊ SABIA?

Um domínio não é o suficiente para que o seu *site* fique disponível na internet. É necessária uma hospedagem em algum servidor. Assim, o domínio é o endereço do seu *site*, a hospedagem é onde o seu *site* está guardado e o *site* em si são os arquivos que formam o seu *site*. Assim, analogamente, como seu endereço de casa, a sua casa e os móveis, respectivamente.

O primeiro serviço é o de diretório que traduz os nomes para endereços IP, sendo a tarefa principal do DNS da internet. Kurose e Ross (2013) descrevem outros três serviços oferecidos pelo DNS. Clique nos itens para saber sobre eles.

- **Apelidos (aliasing) de hosts**

  Quando um nome é muito grande, é possível atribuir um apelido, isto é, um nome menor, mais fácil de lembrar. O nome original é chamado de nome canônico. Assim, o DNS fornece a tradução para o endereço IP e a obtenção do nome canônico.

- **Apelidos de servidor de correio**

  A mesma ideia dos apelidos de hosts, porém para endereços de e-mail.

- **Distribuição de carga**

  Sites muito movimentados são replicados em diversos servidores, cada um rodando em um sistema final e com um endereço IP diferente. O DNS realiza a distribuição de carga entre esses servidores replicados. Na prática, quando clientes consultam um nome mapeado para um conjunto de endereços, o DNS responde com um conjunto de endereços IP, mas faz um rodízio da ordem deles dentro de cada resposta.

A seguir veremos, de forma breve, como funciona o serviço de tradução do DNS e em seguida como é organizada a estrutura de hierarquia dele.

1.2.1 Funcionamento: visão geral

Considerando o serviço de tradução de nome de *hosts* para endereço IP, vamos exemplificar o seu funcionamento. Uma aplicação é executada no *smartphone* de um usuário, sendo esta uma página *web*. Sendo assim, essa aplicação precisa traduzir um nome de *host* para um endereço IP. Primeiramente, a aplicação irá chamar o cliente do DNS, dizendo qual o nome do *host* que necessita da tradução. O DNS do *smartphone* do usuário envia uma mensagem de consulta para a rede (através de datagramas UDP na porta 53). Depois de um provável atraso de milissegundos ou segundos, o DNS no *smartphone* do usuário recebe uma mensagem de resposta DNS fornecendo o mapeamento desejado, ou seja, é transferido para a aplicação que está requisitando (KUROSE; ROSS, 2013).
Do ponto de vista da aplicação, parece simples, mas, na verdade, considere uma caixa-preta que executa o serviço de forma complexa, considerando que diversos servidores DNS distribuídos ao redor do mundo são consultados, considerando também o funcionamento do protocolo da camada de aplicação que especifica a comunicação entre os servidores DNS e os *hosts* que fazem a consulta.

O DNS é um nome atribuído para identificar um *site*, para facilitar o acesso a ele, pois a rede de computadores conhece o dispositivo ou *site* pelo seu número de IP. Do mesmo modo, o nome de uma pessoa pode ser usado para identificar uma pessoa de uma forma mais simples, do que seria pelos números de RG ou CPF. Essa é única semelhança conceitual entre o DNS e os números de RG e CPF de uma pessoa. As pessoas podem ser registradas com nomes e sobrenomes iguais, já o DNS deve ser único. O RG deve ser único por estado e o CPF em todo Brasil, enquanto o endereço IP pode ser diferente para o mesmo *site*, como no caso em que a carga para acesso a um *site* muito movimentado é dividida entre servidores diferentes, assim, cada servidor atribui um número de IP diferente para um mesmo *site*.

Vamos praticar encontrar um endereço IP de um nome no sistema operacional Windows? Para isso, basta você digitar o comando "tracert brasil.gov.br", sem as aspas, no prompt de comando. Repare na rota de enlaces da rede até chegar ao endereço IP do nome requisitado, apresentado como endereço IP final. Nos sistemas operacionais Linux e Windows, você pode digitar o comando "ping" seguido do nome do *site*, por exemplo: "ping brasil.gov.br".

Portanto, o DNS depende de um protocolo de transporte fim a fim subjacente para transferir mensagens DNS entre sistemas finais que se comunicam e trabalha entre sistemas finais que se comunicam utilizando o sistema cliente-servidor.

1.2.2 Hierarquia

Na rede mundial de computadores, os servidores DNS são interligados de forma lógica em uma estrutura de hierarquia. Toda rede deve ter um servidor DNS que lê um nome de domínio (www) e descobre o seu endereço IP correspondente. Se um determinado servidor DNS não possuir o endereço IP correspondente, então vai procurar e consultar em outros servidores DNS espalhados pela rede.
A hierarquia existe para tratar a questão da escala, uma vez que o DNS utiliza vários servidores, por isso os mapeamentos são distribuídos por eles, uma vez que um servidor sozinho não tem acesso a todo mapeamento

de todos os *hosts* da internet. Desse modo, existe o que chamamos de servidores raiz, os de domínio de alto nível (TLD) e os autoritativos conforme mostra a figura abaixo.

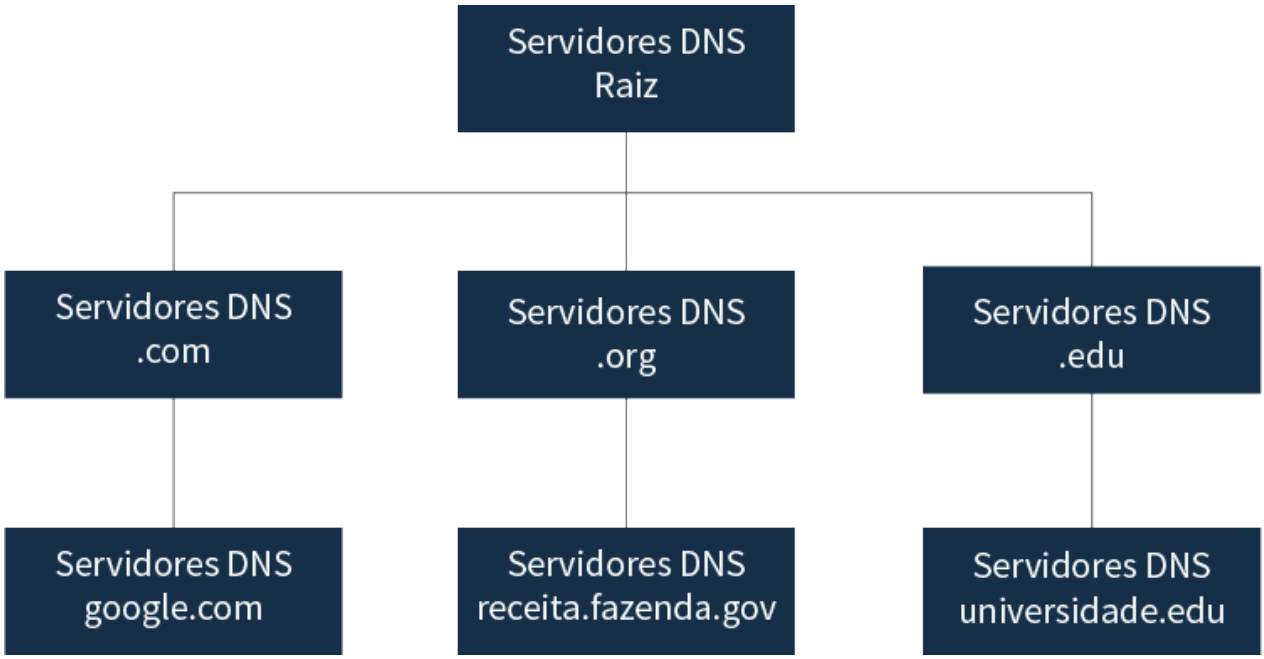

Figura 3 - Exemplo de parte da hierarquia de servidores DNS, composto por um servidor raiz, servidores de TLD e servidores autoritativos.
Fonte: Elaborada pela autora, adaptado de KUROSE; ROSS, 2013, p. 99.

Kurose e Ross (2013) descrevem as três classes de servidores DNS. Clique nas abas e conheça-as.

- **Servidores DNS raiz**

  Existem 13 servidores DNS raiz, sendo que a maioria está localizado na América do Norte. Cada um desses 13 servidores é um conglomerado de servidores replicados por motivos de segurança e confiabilidade.

- **Servidores DNS de TLD**

  São responsáveis por domínios de alto nível, tais como .com, .org, .net, .edu, .gov., além de domínios de países, como o .br, .uk, .pt etc.

- **Servidores DNS autoritativos**

  Uma organização que possua hosts que possam ser acessados publicamente na internet deve fornecer registros DNS que também devem ser acessíveis publicamente. Esses registros devem mapear os nomes desses hosts para endereços IP. O servidor DNS autoritativo dessa organização vai abrigar esses registros. No entanto, a organização pode executar seu próprio servidor DNS autoritativo para abrigar esses registros, ou pode pagar para armazená-los em um servidor DNS autoritativo de algum provedor de serviço, o que torna o serviço mais seguro caso a organização for menor. Muitas organizações de grande porte conseguem manter seus próprios servidores DNS primário e secundário (backup) autoritativos.

Para entender como essas três classes conversam entre si, imagine que um cliente DNS deseja determinar o endereço IP para o nome de um *host* www.google.com. Como uma primeira aproximação, irão ocorrer os

seguintes eventos: primeiro, o cliente irá contatar um dos servidores TLD, que retornará o endereço IP de um servidor autoritativo para google.com. Em seguida, o cliente irá contatar um dos servidores autoritativos para google.com, que vai retornar o endereço IP para o nome do *host* www.google.com.
Outro tipo de DNS, além das três classes citadas, é o servidor DNS local. Ele não pertence à hierarquia, contudo, é importante para a arquitetura DNS. Cada provedor de serviço de internet, tal como de uma universidade, de uma empresa ou de uma residência, tem um servidor DNS local (também chamado de DNS *default*). No momento que um *host* se conecta a um provedor, esse provedor fornece os endereços IP de um ou mais de seus servidores DNS locais, normalmente por DHCP (KUROSE; ROSS, 2013).
No próximo tópico, você verá os conceitos dos protocolos HTTP e HTTPS, os quais interagem com o serviço DNS ao passo que uma aplicação do usuário requisite uma página *web* HTTP/HTTPS.

1.3 Protocolos HTTP/HTTPS: conceitos

O protocolo *HyperText Transfer Protocol* (HTTP), isto é, Protocolo de Transferência de Hipertexto, atua na camada de aplicação do modelo TCP-IP. Ele é utilizado para comunicação entre navegadores e servidores *web*. Quando um navegador recebe uma página, é porque ele se comunicou com um servidor através de um HTTP. Não confunda com HTML, que é uma forma de codificação das páginas *web* para que possam ser apresentadas pelos navegadores (SCHMITT, 2013).
Na prática, quando um http:// aparece em um URL, significa que um usuário está sendo conectado a um servidor *web* para que seja possível transferir arquivos do servidor para um navegador, com o objetivo de visualizar uma página *web*. Contudo, isso apenas especifica o que o navegador e o servidor *web* dizem um para o outro, mas não diz como eles se comunicam, afinal, quem faz isso é o protocolo TCP.
Nesse contexto, Kurose e Ross (2013, p. 72) lembram da importância de definir o que é uma página *web*:

> Uma **página Web** (também denominada documento) e constituída de objetos. Um **objeto**é apenas um arquivo – tal como um arquivo HTML, uma imagem JPEG, um applet Java, ou um clipe de vídeo – que se pode acessar com um único URL. A maioria das páginas Web são constituídas de um **arquivo-base HTML**e diversos objetos referenciados.

Assim, cada URL contém: o nome do *host* (*hostname*) do servidor que abriga o objeto e o nome do caminho do objeto, isto é, dois componentes. Por exemplo, https://mail.google.com/mail/u/0/#label/Faculdade. Esse URL é composto pelo *hostname* do servidor, que é mail.google.com e o nome do caminho: mail/u/0/#label/Faculdade. Veja que o *hostname* se refere ao *e-mail* do google (Gmail) e o caminho é referente a uma pasta dentro do *e-mail* (configurada como marcador) chamada Faculdade. Se você estivesse vendo uma imagem, certamente o final da URL seria o nome da imagem seguida de sua extensão.

VOCÊ O CONHECE?

O cientista da computação Tim Berners-Lee é considerado o pai da World Wide Web (WWW). Em 1989, ele teve a ideia e montou uma proposta do que chamavam de sistema de rede de gerenciamento e troca de informações. Em maio de 1990, tornou o projeto real, implementando a primeira comunicação entre um cliente HTTP e um servidor na internet.

Os servidores HTTP também são conhecidos como servidores *web*. Eles são responsáveis por permitir a publicação das páginas *web*. Qualquer *site* disponível para acesso deve estar configurado em um servidor HTTP. Normalmente, para que você possa hospedar um *site* e torná-lo visível na internet por uma página *web*, você precisa pagar um servidor *web*, que é mantido por alguma empresa, provavelmente com custos em manter uma máquina especialmente para servir como servidor *web*. Fica claro, dessa forma, que se trata de uma aplicação cliente-servidor, em que o cliente faz uma requisição e o servidor devolve uma resposta.

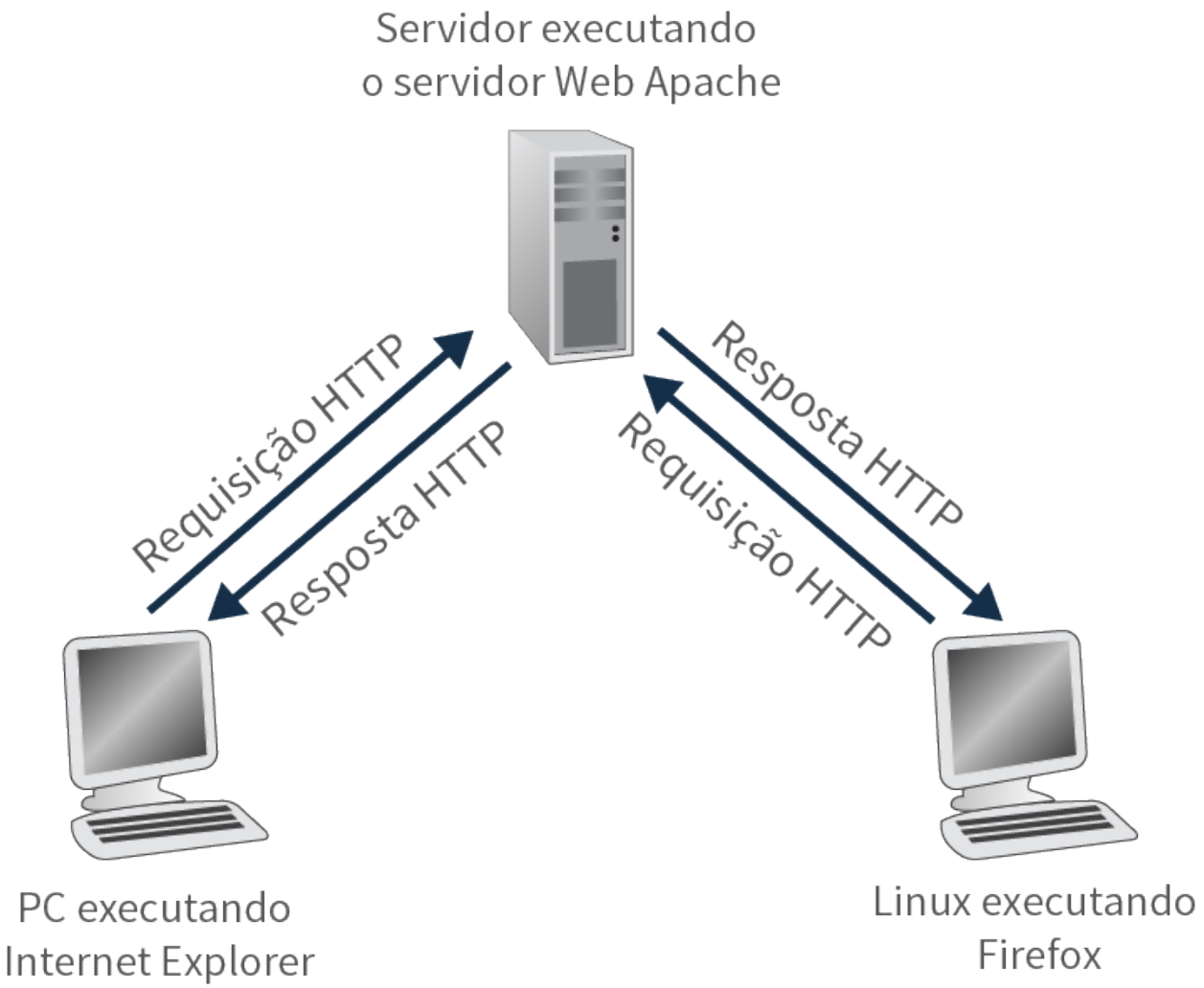

Figura 4 - Comportamento de requisição-resposta do HTTP.
Fonte: KUROSE; ROSS, 2013, p. 77.

Desse modo, é o HTTP que define como os clientes requisitam páginas *web* aos servidores e como eles as transferem aos clientes, como mostra a figura acima.

1.3.1 Funcionamento

Quando uma página é requisitada, ou seja, quando o usuário clica na URL ou a insere no navegador, o navegador envia ao servidor as mensagens de requisição HTTP para os objetos da página *web*. Então, o servidor recebe as requisições e responde com as mensagens de resposta HTTP que contêm os objetos requisitados.

VOCÊ SABIA?

O Apache é um dos servidores compatíveis com HTTP mais utilizado do mundo, livre e de código aberto, mantido pela *Apache Software Fundation*. Sua instalação é simples e pode ser utilizado em diversos sistemas operacionais, sendo, desse modo, multiplataforma.

Em se tratando de internet, o HTTP utiliza do TCP para transportar os pacotes na camada subjacente, já que o interesse é trabalhar com entregas confiáveis. Primeiramente, o cliente HTTP começa uma conexão TCP com o servidor, em seguida os processos do navegador e do servidor acessam o TCP através de suas interfaces de *socket* , que é a porta entre o processo cliente e a conexão TCP. Enquanto do lado do servidor, a interface de *socket* é a porta entre o processo servidor e a conexão. Em seguida, o cliente envia mensagens de requisição HTTP para sua interface *socket* e recebe mensagens de resposta HTTP de sua interface *socket*. Do mesmo modo, o servidor HTTP também recebe mensagens de requisição de sua interface *socket* e envia mensagens de resposta para sua interface *socket*. Assim, quando o cliente envia uma mensagem para sua interface *socket,* a mensagem passa a ser trabalhada pelo TCP (KUROSE; ROSS, 2013).

É importante ressaltar que o servidor tem memória fraca, ou melhor dizendo, não tem memória, logo, depois que envia ao cliente os arquivos solicitados naquela requisição e depois requisita novamente o mesmo objeto, o servidor manda novamente o objeto sem lembrar que já mandou essa mesma resposta há poucos segundos. O servidor HTTP tem a característica de ser um protocolo sem estado, pois não armazena informação alguma sobre clientes. Além disso, o servidor *web* está em constante funcionamento, por isso tem um IP fixo, atendendo às requisições de inúmeros navegadores diferentes.

Métodos HTTP e Códigos de erro

Os métodos HTTP definem o que o servidor precisa fazer com o URL fornecido pelo usuário, isto é, quando é feita uma requisição do cliente para o servidor. Clique nos itens e veja os métodos, de acordo com SCHMITT (2013).

GET	Faz a solicitação de um arquivo ou script através do protocolo HTTP. Sendo o método mais comum, é conhecido por todos servidores.
HEAD	Faz o mesmo que o GET, no entanto, sem retorno de recurso. Útil para receber metainformações sem que seja necessário obter todo o conteúdo.
POST	Utilizado para enviar dados que precisam ser processados, como, por exemplo, dados de um formulário HTML.
PUT	Utilizado para atualizar determinado recurso.
DELETE	Utilizado para excluir determinado recurso.
TRACE	Ecoa a requisição por todo o caminho até chegar ao recurso de destino. É útil para realizar teste de loopback e para debug.
OPTIONS	Utilizado pelos clientes para descobrir quais são as opções de requisição permitidas para determinado recurso em um determinado servidor.

CONNECT	Utilizado para iniciar uma comunicação bidirecional com o recurso desejado. É útil para abrir um túnel.
LINK	Conecta dois recursos existentes.
UNLINK	Encerra uma conexão existente entre dois recursos.

O mais comum é que o servidor HTTP implemente ao menos os métodos GET e HEAD, de modo que um cliente estabeleça uma conexão com o servidor e, então, envie uma requisição. Essa requisição contém a URL, a versão do protocolo, uma mensagem MIME (codifica dados em formato ASCII para dados compatíveis para serem transferidos pela internet) que contém os modificadores da requisição, as informações sobre o cliente e o conteúdo no corpo da mensagem. Em seguida, o servidor responde a requisição devolvendo uma linha de *status*, a versão de protocolo e um código indicando operação bem-sucedido ou indicando erro, as informações do servidor, as metainformações da entidade e o conteúdo no corpo da mensagem. Uma vez terminado o envio da resposta do servidor para o cliente, a conexão se encerra.
Esses códigos de resposta são padrões. Eles são conhecidos como *status* e são muitos, descritos a seguir, conforme Barrett (2010, p. 283):

1xx: indica uma mensagem apenas informativa
2xx: indica sucesso de algum tipo
- **200:** OK
- **201:** Created (Criada)
- **202:** Accepted (Aceita)
- **204:** No Content (Sem Conteúdo)

3xx: Redireciona o cliente para outro URL
- **301:** Moved Permanently (Mudou Permanentemente)
- **302:** Moved Temporarily (Mudou Temporariamente)
- **304:** Not Modified (Não Modificada)

4xx: Indica um erro na parte do cliente
- **400:** Bad Request (Requisição Malformada)
- **401:** Unauthorized (Não Autorizado)
- **403:** Forbidden (Proibido)
- **404:** Not Found (Não Encontrado)

5xx: Indica um erro na parte do servidor
- **500:** Internal Server Error (Erro Interno do Servidor)
- **501:** Not Implemented (Não Implementado)
- **502:** Bad Gateway (Gateway Errado)
- **503:** Service Unavailable (Serviço Indisponível)

Os códigos citados são alguns definidos pelo protocolo HTTP, contudo, cada servidor pode definir códigos próprios. As especificações do HTTP, tais como são os formatos das mensagens de requisição e de resposta, assim como todas as outras especificações de funcionamento, são definidas na RFC 1945 e na RFC 2616.

1.3.2 HTTPs: HTTP sobre SSL/TLS

De forma simples, a sigla HTTPs se refere à versão segura do HTTP, significando Protocolo de Transferência de Hipertexto Seguro. Com a internet cada vez mais alvo de ataques de segurança, foi desenvolvido um método que assegurasse transações e navegação de forma segura principalmente em páginas *web* de bancos e qualquer outra que solicite dados pessoais de cartões de crédito, por exemplo.

Sabemos que existem algumas técnicas de criptografia utilizadas para prover segurança em aplicações específicas, como as que exigem autenticação de ponto final, de usuário, de integridade de dados, de sigilo. Para isso existe o *Secure Sockets Layer* (SSL), um sistema de criptografia que utiliza uma chave pública e uma chave privada que apenas o destinatário conhece. Portanto, se em vez de http:// aparece https:// na URL da página *web*, significa que os dados serão criptografados para prover a segurança.

De acordo com Kurose e Ross (2013), o SSL protege, na verdade, as conexões TCP, então, ele pode ser empregado por qualquer aplicação que execute o TCP, no entanto, muitas vezes é usado para oferecer segurança em transações que ocorrem pelo HTTP.

Em tempo, é importante ressaltar que o SSL se demonstrou inseguro em 2014, o que fez com que surgisse o TLS (*Transport Layer Security*), que funciona de forma semelhante ao SSL e com a mesma função. A questão é que a criptografia do SSL foi quebrada e a tecnologia teve que ser atualizada. No entanto, muitos ainda utilizam o termo SSL por estar mais enraizado quando se fala de segurança.

As tecnologias são desenvolvidas de forma muito rápida e é complexo trabalhar com conceitos aperfeiçoados de outros antes mesmo do "antigo" conceito ser difundido em larga escala, como foi o caso do SSL.

Síntese

Você pôde conhecer o NTP e entender como controlar a hora mundial em redes de computadores é algo importante e que, se não for implementado pelos servidores, pode trazer alguns problemas, inclusive de segurança. Além disso, você viu que o conceito de servidores se aplica também ao DNS e ao HTTP, já que ambos implementam suas funcionalidades nos servidores. Enquanto o DNS é um banco de dados distribuído em vários servidores para traduzir nomes em endereços de IP, o HTTP é um protocolo que torna possível a comunicação entre navegadores e servidores *web* com a finalidade de exibir as páginas *web*.

Nesta unidade, você teve a oportunidade de:

- compreender como o NTP é importante para controlar a hora mundial e como isso pode implicar nas tarefas realizadas por dispositivos conectados à rede mundial de computadores;
- entender que o DNS é utilizado pelas aplicações para traduzir nomes de domínio de *hosts* para endereços IP para facilitar a tarefa dos usuários dos aplicativos.
- analisar como o protocolo HTTP e os servidores *web* colaboram para que a páginas *web* sejam exibidas na internet.

Bibliografia

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

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.

MILLS, D. L. **Computer network time syncronization:** the network time protocol. CRC Press. 2006.

MILLS, D. L. **RFC 1305**: Network Time Protocol (Version 3) Specification, Implementation and Analysis. University of Delaware, 1992. Disponível em: https://tools.ietf.org/pdf/rfc1305.pdf. Acesso em: 22 jul.2019.

MILLS, D. L. *et al*. **RFC 5905**: Network Time Protocol Version 4: Protocol and Algorithms Specification. JHU/APL, 2010. Disponível em: https://tools.ietf.org/pdf/rfc5905.pdf. Acesso em: 22 jul. 2019.

MOREIRAS, A. M. **Importância e funcionamento do sincronismo de tempo na Internet e do NTP (Tutorial NTP 2010)**. 2010. 1 vídeo (79 min 31 s). Disponível em: http://www.zappiens.br/portal/VisualizarVideo.do?_InstanceIdentifier=0&_EntityIdentifier=cgiFRWd-V3ZGGRqao9GOvyXEQXLYfpqFgvmOU-1_2tcChk.&idRepositorio=0. Acesso em: 7 jul. 2019.

NTP.BR. **O NTP**. 2019. Disponível em: https://ntp.br/ntp.php. Acesso em: 24 jun. 2019.

NTP.BR. **Guia Windows**. 2019a. Disponível em: https://ntp.br/guia-win-avancado.php. Acesso em: 7 jul. 2019.

NTP.BR. **Guia MAC**. 2019b. Disponível em: https://ntp.br/guia-mac-avancado.php. Acesso em: 7 jul. 2019.

NTP.BR. **Guia Linux/BSD**. 2019c.Disponível em: https://ntp.br/guia79-linux-avancado.php. Acesso em: 7 jul. 2019.

NTP.BR. **Guia Roteadores**. 2019d.Disponível em: https://ntp.br/guia-rot.php. Acesso em: 7 jul. 2019.

NTP.BR. **Guia Windows**. 2019e.Disponível em: https://ntp.br/guia-win-comum.php. Acesso em: 7 jul. 2019.

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

NÚCLEO DE INFORMAÇÃO E COORDENAÇÃO DO PONTO BR **Tutorial NTP**: Importância e Funcionamento do Sincronismo de Tempo na Internet e do NTP. 2010. Disponível em: http://www.ceptro.br/pub/CEPTRO/MenuCEPTROEventoTutorialNTP/tutorial-ntp-2.pdf. Acesso em: 24 jun. 2019.