INTRODUÇÃO A REDES DE COMPUTADORES

CAPÍTULO 3 - COMO FUNCIONAM AS REDES DE COMPUTADORES E ARQUITETURA OSI E TCP/IP?

Josiane Boeira Kirinus Fernandes

Introdução

Sabemos que as redes de computadores e seus serviços têm papel fundamental na vida das pessoas e de organizações e quando pensamos na quantidade de dispositivos e computadores interligados em rede, temos que levar em conta que cada um deles tem sua identificação. Para compreender como isso funciona, vamos começar partindo do Serviço de nomes de domínios e das aplicações P2P.

As camadas de protocolos e seus serviços são princípios fundamentais da arquitetura de redes e, neste capítulo, vamos estudar duas camadas: a de transporte e a de rede. A camada de transporte fica posicionada entre as camadas de aplicação e a de rede é um componente central da arquitetura em camadas. É por meio dela que os serviços de comunicação ocorrem diretamente nos processos de aplicações, que são executados em hospedeiros diferentes.

Vamos entender os princípios fundamentais da camada de transporte e os princípios de sua implementação em protocolos da internet mais comuns e utilizados, tais como UDP e o TCP. Você conhece os principais fundamentos de funcionamento desses protocolos?

Também vamos entender mais detalhes sobre como ocorre a implementação do serviço de comunicação hospedeiro a hospedeiro e o que está por baixo, fundamentando esse serviço de comunicação.

Assim, vamos observar o relacionamento entre as camadas de transporte e de redes e como se dá a comunicação, de forma confiável, entre sistemas ou hospedeiros diferentes. Você conhece os principais aspectos que fundamentam cada uma dessas camadas? Você sabe como se dá a comunicação de pacotes em cada uma dessas camadas? Sabe quais são as características e principais diferenças entre elas? Quais são seus principais recursos?

A partir de agora, vamos conduzir o conteúdo, com o objetivo de responder todos esses questionamentos.

Vamos em frente! Bons estudos!

3.1 DNS e P2P

Na camada de aplicação que se encontram todas as aplicações. As camadas inferiores a ela têm a função de proporcionar um serviço de transporte confiável, entretanto elas não realizam qualquer tarefa para os usuários. Mesmo na camada de aplicação é necessária a utilização de protocolos que darão suporte e permitirão que as aplicações funcionem. Um desses protocolos é o DNS, que é responsável pela nomenclatura na internet. Quanto aos tópicos destinados às aplicações P2P, será possível a constatação de que nesse tipo de arquitetura, as aplicações não exigem dependência de servidores com infraestrutura sempre em funcionamento e a comunicação entre pares intermitentemente de hospedeiros conectados, podem realizar a comunicação entre si.

3.1.1 O sistema de nomes de domínios

Há vários tipos de identificação para pessoas, como o nome na certidão de nascimento, o número no RG, o número no CPF e até da carteira de habilitação. Dependendo da situação, se usa um ou outro. Da mesma forma que as pessoas podem ser identificadas de maneiras diferentes, isso ocorre com os hospedeiros na internet.

Nomes de hospedeiros podem ser formados por caracteres alfanuméricos de tamanho variável e ficaria mais difícil de serem compreendidos e processados por roteadores. Em função disso, hospedeiros também são identificados pelos endereços IP.

VOCÊ SABIA?

Um endereço IP é formado por quatro conjuntos de 4 *bytes*, parecido com este: 192.165.106.20. Cada conjunto de *bytes*, separado por ponto, pode ser expresso de 0 a 255. Endereços IP possuem uma hierarquia, ou seja, ele é interpretado da esquerda para direita, dando uma noção de informações específicas sobre o lugar onde o hospedeiro está localizado na internet (revelando qual rede, das muitas que fazem parte da internet).

Segundo Kurose e Ross (2014, p. 96), “o DNS é (1) um banco de dados distribuído implementado em uma hierarquia de servidores de nome (servidores DNS), e (2) um protocolo da camada de aplicação que possibilita que hospedeiros consultem o banco de dados distribuído”.

Frequentemente, o DNS é usado por outras entidades da camada de aplicação (HTTP, SMTP e FTP) para efetuar a tradução nomes hospedeiros fornecidos por usuários para endereços IP. Para facilitar a compreensão, imagine o que acontece quando um navegador executado em um computador de um usuário, solicita alguma URL. Para que o computador do usuário consiga enviar uma mensagem de solicitação HTTP ao servidor *web*, esse computador primeiramente necessita a obtenção do endereço IP dessa URL. E como é realizado?

Segundo Kurose e Ross (2014), esse processo é realizado seguindo alguns passos, clique nos itens a seguir.

- 1. O próprio computador do usuário executa o lado cliente da aplicaçã DNS.

- 2. O navegador retira o nome de hospedeiro, www.exemplourl.edu.br, d URL e transfere o nome para o lado cliente da aplicação DNS.

- 3. O computador cliente manda uma consulta com o nome do hospedei para um servidor DNS.

- 4. O computador cliente DNS finalmente recebe um retorno, incluindo endereço IP, que corresponde ao nome de hospedeiro.

- 5. Assim que o navegador recebe o endereço do DNS, já pode abrir um conexão TCP com o processo servidor HTTP localizado naquele endereç IP.

Pode acontecer de o DNS adicionar mais um atraso às aplicações de Internet que o utilizam. Entretanto, o endereço IP desejado normalmente fica no cache de um servidor DNS “perto”, o que facilita a redução do tráfego DNS na rede e também o atraso médio do DNS.

VOCÊ QUER LER?

O livro "Introdução ao DNS: Aprenda a instalar e configurar uma infraestrutura de DNS na prática" (FURTADO, 2016), apresenta conceitos fundamentais do DNS, serviço que possibilita que o usuário tenha acesso a recursos disponibilizados na rede fazendo uso de um nome de um computador, em vez do seu endereço IP. Traz, também, aspectos relevantes do IPv4 e IPv6.

O DNS fornece alguns outros serviços de importância, além de traduzir nomes hospedeiros para endereços IP. Listamos na interação a seguir. Clique para ver.

Atribuição de apelidos de hospedeiro: pode acontecer de um hospedeiro ter um nome complicado. Kurose e Ross (2014) destacam o seguinte exemplo: *relayl.west-coast.enterprise.com*, que pode ter alguns apelidos, como *enterprise.com* e *www.enterprise.com*. O nome de hospedeiro *relayl.west-coast.interprise.com* é chamado de nome canônico. Os apelidos, quando existem, são mais fáceis de lembrar do que o nome canônico. O DNS pode ser chamado por alguma aplicação, para a obtenção do nome canônico correspondente a um apelido fornecido, bem como, para obtenção do endereço IP do hospedeiro.

Atribuição de apelidos de servidor de correio: endereços de *e-mail* podem não serem fáceis de lembrar. Kurose e Ross (2014) citam um exemplo, se Bob tem uma conta no Hotmail, seu endereço pode ser simplesmente *bob@hotmail.com.* Entretanto, o nome de hospedeiro do servidor do Hotmail é mais complicado e difícil de lembrar do que apenas *hotmail.com,* como o nome canônico ser algo semelhante com *relayl.west-coast.hotmail.com*. O DNS pode ser requisitado por uma aplicação de correio eletrônico para a obtenção do nome canônico com base no apelido provido e também o endereço IP do hospedeiro.

Distribuição de carga: o DNS também pode ser utilizado para a realização de distribuição de carga entre servidores replicados, como os servidores *web* replicados. Os *sites* que recebem muitas visitas, ou seja, mais movimentados são replicados em muitos servidores, onde cada servidor roda em um sistema final diferente e tem um endereço IP diferenciado. Então, quando os servidores *web* são replicados, um conjunto de endereços IP é associado a um único nome canônico e contido no banco de dados do DNS.

O próximo tópico tem como objetivo fornecer uma visão geral do funcionamento do DNS. Vamos entender o serviço que disponibiliza a tradução do nome e hospedeiro para endereço IP.

3.1.2 Uma visão geral do funcionamento do DNS

Imagine a situação de uma determinada aplicação, como um navegador *web*, que roda na máquina de um usuário e necessita da tradução do nome de hospedeiro para endereço IP. Essa aplicação precisa solicitar o lado cliente do DNS, evidenciando o nome de hospedeiro que necessita da tradução. E então, o DNS do hospedeiro do usuário assume o controle, transmitindo uma mensagem e consultando para dentro da rede. As mensagens de consulta e de retorno do DNS são transmitidas dentro de datagramas UDP para a porta 53. Após a ocorrência de atraso de milissegundos a segundos, o DNS no hospedeiro do usuário recebe uma mensagem de retorno do DNS disponibilizando o solicitado, que, então, é encaminhado para a aplicação que tem interesse. Assim sendo,

do prisma dessa aplicação, que fica no computador do cliente, o DNS efetua o serviço de tradução de uma forma simplificada e direta. É interessante salientar que esse serviço que implementa a tradução é complexo e necessita de um número elevado de servidores de nomes distribuídos pelo mundo, tal qual um protocolo de camada de aplicação que determina como deve acontecer a comunicação entre os servidores de nomes e os hospedeiros clientes.
Uma forma simplificada de funcionamento do DSN seria constituída de um servidor de nomes com todos os mapeamentos de forma centralizada, onde os clientes simplesmente direcionariam a totalidade de consultas a esse servidor único de nomes, que retornaria resposta de forma direta aos clientes que estão realizado as consultas. Apesar dessa forma simplificada ser atrativa, hoje ela não é a ideal para Internet e sua imensidão de hospedeiros que só cresce. Esse arranjo simplificado geraria alguns tipos de problemas. Listamos alguns na interação a seguir. Clique para ver.

Ser único ponto de falha: se esse servidor parar de funcionar, toda a internet também parará.

Grande volume de tráfego: esse servidor de nomes único teria de realizar a manipulação de todas as consultas DNS, ou seja, todas as solicitações HTTP e *e-mail*s geradas por milhões de hospedeiros.

Centralização do banco de dados distante: se existisse um único servidor de nomes, ele nunca conseguiria estar “perto” de todos os clientes que realizam requisições. Para facilitar a compreensão, se esse único servidor de nomes ficasse localizado na cidade de São Paulo, as requisições de Japão e Nova Zelândia teriam de “viajar” para o outro lado do mundo e teriam de enfrentar vias congestionadas e lentas, o que resultaria em atrasos relevantes.

Manutenção: esse único servidor de nomes deveria manter registrados todos os hospedeiros da internet, o banco de dados seria imenso e necessitaria de atualizações frequentes para conseguir atender a todos os novos hospedeiros.

Então, para que esses problemas não ocorram, o DNS utiliza um vasto número de servidores, que são disponibilizados de forma organizada, hierárquica e distribuída pelo mundo afora. Impossível ter um servidor de nomes com todos os mapeamentos para todos os hospedeiros que compõem a internet. No entanto, os mapeamentos são distribuídos pelos servidores de nomes. Existem três classes de servidores de nomes: servidores de nomes raiz, servidores DNS de domínio de alto nível e servidores DNS com autoridade, que são organizados em forma de hierarquia (figura abaixo).

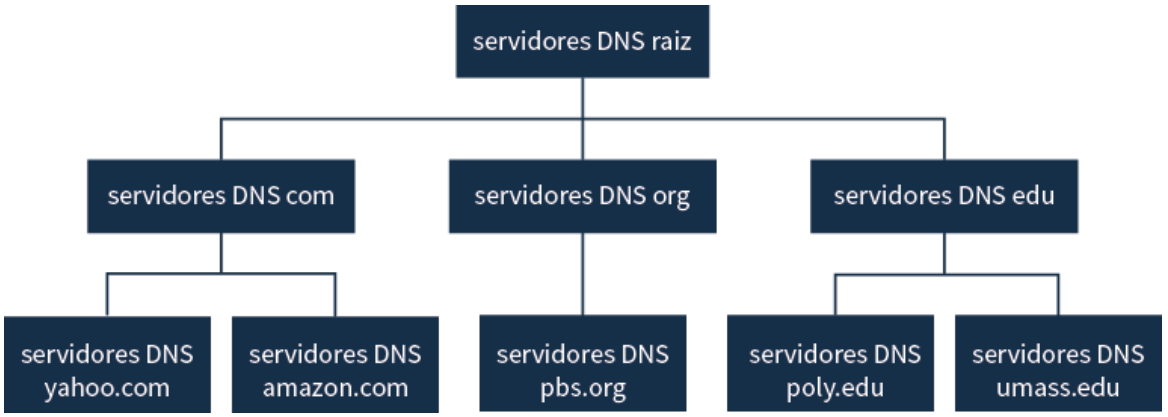


Figura 1 - Demonstração da estrutura hierárquica de servidores DNSs, em suas três classes: servidores de nomes raiz, servidores DNS de domínio de alto nível (TDL – *Top-level domain*) e servidores DNS com autoridade.
Fonte: KUROSE; ROSS, 2014, p. 99.

Como se dá o funcionamento nessa hierarquia de três classes de servidores? Para facilitar a compreensão, imagine a situação: um cliente DNS necessita estabelecer o endereço IP para o nome de hospedeiro *www.exemploURL.com,* como uma primeira opção. Poderia transcorrer assim: primeiro, o cliente estabeleceria contato com um dos servidores raiz, que retornaria endereços IP dos servidores TDL, para o domínio de alto nível *.com.* O próximo passo seria o cliente contatar um desses servidores TDL, que retornará o endereço IP de um servidor com autoridade para *exemploURL.com* e, por fim, o cliente estabeleceria contato com um dos servidores com autoridade para amazona.com, que retornaria o endereço IP para o nome de hospedeiro *www.exemploURL.com.* Conseguiu compreender o processo?
Vamos entender um pouco mais sobre essas três classes de servidores DNS clicando nos cards a seguir.

Servidores de nomes raiz

De acordo com Kurose e Ross (2014), na internet existem13 servidores de nomes raiz, que são denominados de A a M e a grande maior fica na América do Norte (figura a seguir). Cabe ressaltar que, apesar da menção de cada um dos 13 servidores de nomes raiz como sendo um, na verdade cada um é um conglomerado de servidores replicados, com o objetivo de garantir mais confiabilidade e segurança.

Em sua grande maioria, universidades e empresas e organizações de grande porte realizam a implementação e manutenção de seus próprios servidores DNS primário e secundário, para *backup* de segurança, com autoridade.

Figura 2 - Disposição dos servidores DNS raiz em 2012, com seu nome, organização e a referida localização.
Fonte: KUROSE; ROSS, 2014, p. 99.

É importante destacar que os servidores raiz, TDL e com autoridade estão relacionados à hierarquia de servidores DNS.

3.1.3 Aplicações P2P

Aplicações como a WWW, *e-mail* e DNS fazem uso da arquitetura cliente-servidor e dependem significativamente de servidores que permanecem sempre ligados e disponíveis. Na arquitetura P2P, se houver essa dependência e no caso a ocorrência é mínima de servidor sempre disponível e ligado. Ao contrário, existem duplas de hospedeiros conectados de forma descontinuada, que são denominados pares e efetuam a comunicação entre si. Esses pares são de propriedade dos próprios usuários.
Considere a situação na qual é necessária a distribuição de um arquivo grande por meio de um único servidor para um extenso número de hospedeiros, os denominados pares. Esse arquivo pode ser, por exemplo, uma versão nova de algum *software* ou, até mesmo, um arquivo de música MP3. Na arquitetura cliente-servidor, o servidor enviaria uma cópia do arquivo para cada um dos pares, o que resulta no consumo elevado de banda do servidor. Essa distribuição de arquivos P2P funciona da seguinte maneira: cada par pode realizar a redistribuição de qualquer parte do arquivo que recebeu de outros pares, auxiliando dessa forma, o servidor nesse processo de distribuição.
O protocolo mais utilizado na distribuição de arquivos P2P é o BitTorrent, desenvolvido por Bram Cohen.

VOCÊ O CONHECE?

Bram Cohen é um cientista e cofundador da BitTorrent e desenvolvedor do protocolo de distribuição de arquivos P2P. Antes de criar o BitTorrent, Bram trabalhou em uma empresa chamada MojoNation, que possibilitava que usuários dividissem arquivos confidenciais em partes codificadas e distribuíssem essas partes para outros usuários que tivessem o MojoNation. Foi essa concepção que Bram fundamentou para desenvolvimento do BitTorrent.

O BitTorrent é um protocolo P2P bastante utilizado e popular para a realização de distribuição de arquivos. No linguajar dele, o grupo de todos os pares que participam da distribuição de um arquivo específico é denominada torrente. Esses pares em um *torrent* realizam o *download* de partes de mesmo tamanho do arquivo entre si, no caso esse tamanho fica em torno de 256*kBytes*. No momento que um par participa de um *torrent*, ele não tem nenhuma parte. Entretanto, com o passar do tempo, ele vai acumulando mais partes. No momento que está fazendo *download* de partes, e ele também pode fazer *uploads* de partes para outros pares.

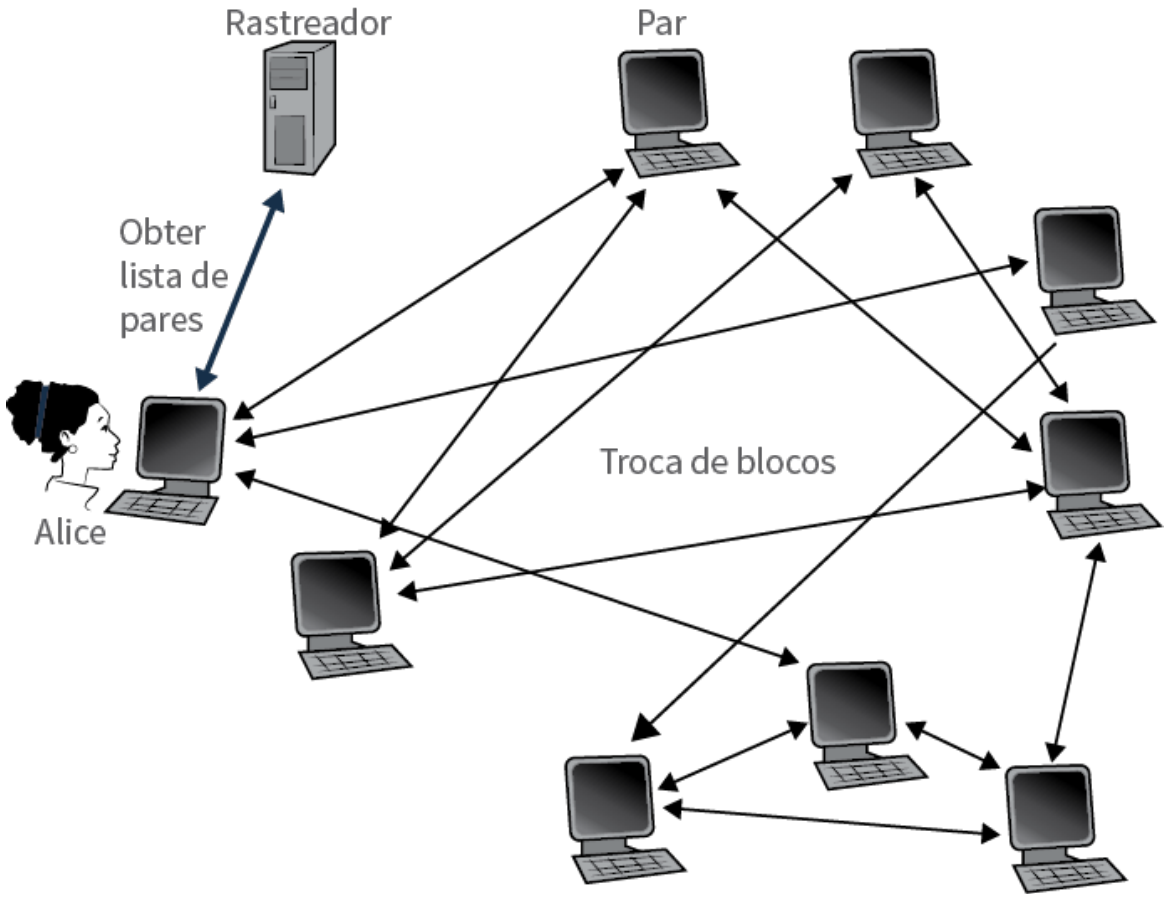


Figura 3 - Demonstração de como ocorre a distribuição de arquivos com a utilização do BitTorrent.
Fonte: KUROSE; ROSS, 2014, p. 110.

No momento em que o par conseguir adquirir todo o arquivo desejado, ele pode então, sair do *torrent* ou então, ele permanece no *torrent* e continua oferecendo *upload* de partes a outros pares. É importante ressaltar que a qualquer momento o par pode sair do *torrent* com somente um subconjunto de partes e depois retornar.

3.2 Camada de transporte

Sabe-se que a camada de transporte tem a responsabilidade de comunicação entre os processos finais de uma mensagem inteira. Por processo, entende-se programa aplicativo sendo executado em um *host*. Apesar de a camada de rede fazer o gerenciamento da entrega de pacotes individuais do local de origem até o local de destino, ela não presume que existe qualquer tipo de relacionamento entre os pacotes. Essa camada de rede trata cada pacote de maneira independente, como se cada um desses pacotes fosse ligado a uma mensagem diferente.
Já a camada de transporte consegue garantir integridade e a sequência de entrega dos pacotes de uma mensagem completa, cuidado e fazendo o controle de erros durante a

transmissão e, também, controla o fluxo de dados. A seguir, serão apresentados os princípios da camada de transporte, bem como o protocolo UDP.

3.2.1 Introdução à camada de transporte

A camada de transporte fica localizada entre as camadas de aplicação e a camada de rede. É considerada peça chave na arquitetura de redes em camadas. A camada de transporte tem como objetivo o fornecimento dos serviços de comunicação direta aos processos de aplicações executados em hospedeiros diferentes.

VOCÊ QUER VER?

O filme *A rede* (BRANCATO; FERRIS, 1995) mostra o mundo das redes de computadores e como acessá-las para a obtenção de vantagens. Neste filme, a atriz Sandra Bullock vive a história de uma programadora que tem sua identidade apagada, quando acessa, por engano, um *software* de uma organização criminosa.

Sabemos que um protocolo da camada de transporte visa o fornecimento de comunicação lógica entre processos de aplicação, que são executados em hospedeiros diferentes. Mas o que se entende por comunicação lógica? No prisma de uma aplicação, tudo se passa como se os hospedeiros que executam os processos fossem conectados de forma direta e o que acontece, na verdade, é que esses hospedeiros podem estar localizados em extremos opostos no mundo, conectados por diversos roteadores e vários tipos de enlaces.

Esses processos de aplicação fazem uso da comunicação lógica fornecida pela camada de transporte para o envio de mensagens entre si, sem se preocupar com detalhamento da infraestrutura física usada no transporte dessas mensagens. Na figura a seguir, é possível ter uma noção de como funciona a comunicação lógica.

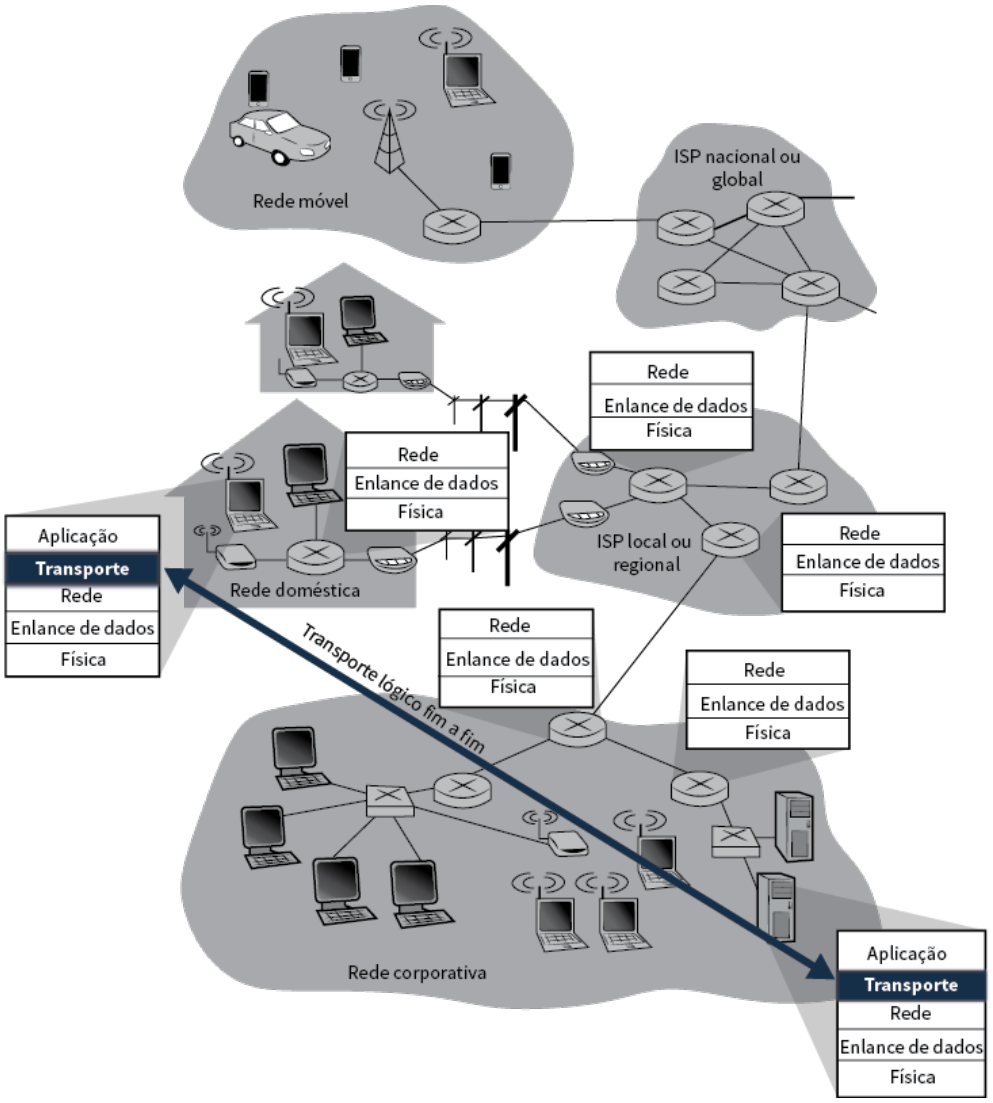


Figura 4 - Demonstração de como a camada de transporte faz o fornecimento da comunicação lógica e não física entre processos de aplicações.
Fonte: KUROSE; ROSS, 2014, p. 141.

Na figura acima, é possível visualizar que os protocolos de camada de transporte são implementados nos sistemas finais, porém não em roteadores de rede. Na origem, a camada de transporte faz a conversão das mensagens que recebe de um processo de aplicação em pacotes de camada de transporte, chamado segmentos de camada de transporte. Como isso é realizado? As mensagens da aplicação são fragmentadas em partes menores e, então, é adicionado um cabeçalho de camada de transporte para cada uma dessas partes para criar o segmento de transporte. Segundo Kurose e Ross (2014), a camada de transporte passa o segmento para a camada de rede no sistema final remetente, onde ele é encapsulado em um pacote de camada de rede (um datagrama) e transmitido ao destinatário.

VOCÊ QUER LER?

O livro “Protocolos de Comunicação” (BRITO; BRITO, 2013) traz fundamentos sobre protocolos de comunicação, abordando desde os aspectos básicos de redes de computadores até os protocolos e suas características.

Observe que os roteadores de rede agem unicamente nos capôs de camada de rede datagrama, ou seja, não são examinados os campos do segmento de camada de transporte encapsulado com o datagrama. No outro lado, no destinatário, a camada de rede faz a extração do datagrama do segmento de camada de transporte e fornece-o novamente para a camada de transporte.

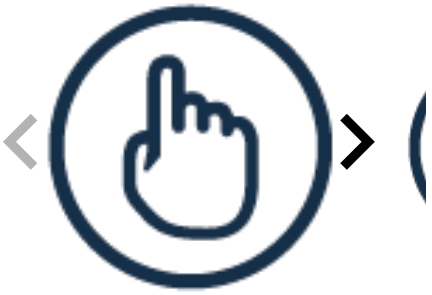
A camada de transporte faz uso de protocolos que podem ser divididos em categorias importantes: orientado a conexão e não orientado a conexão. Um protocolo de camada de transporte que é não orientado à conexão, trata cada segmento independentemente como pacote e realiza a entrega à camada de transporte no equipamento do destinatário. Já um protocolo de camada de transporte orientado à conexão tem a responsabilidade de estabelecer uma conexão de forma virtual com a camada de transporte do equipamento de destino, antes de começar a transferir os pacotes de dados. No momento em que todos os dados tiverem sido transferidos, essa conexão encerra-se. O próximo tópico abordará os protocolos mais utilizados, UDP, não orientado à conexão e o TCP, orientado à conexão.

3.2.2 UDP

O UDP (*User Datagram Protocol*) é um protocolo de transporte do tipo sem conexão e não confiável. O que isso quer dizer? Ele não adiciona controle algum aos serviços de entrega IP, com exceção do fato de realizar a implementação da comunicação entre os processos, no lugar da comunicação entre *hosts*. Da mesma forma, a verificação de erros é implementada de maneira muito limitante.

Mas se o UDP é simples e não confiável, porque um processo o usaria? Porque, apesar das limitações, existem algumas vantagens. O UDP é um protocolo muito simples e com um mínimo de *overhead*. Quando um processo desejar enviar uma mensagem pequena e não ter a preocupação com a confiabilidade, o UDP é uma boa opção. O envio de mensagens pequenas usando o UDP exige menor interação entre o remetente e o

receptor do que quando se usa por exemplo, o TCP.
O protocolo TCP não é sempre preferido ao UDP, apesar de fornecer serviço confiável de transferência de dados. Muitas aplicações se adaptam melhor ao UDP em função dessas razões, apresentadas na interação a seguir. Clique para ler.



Oferecimento de melhor controle no nível da aplicação sobre quais dados são transmitidos e em que momento: usando o UDP, assim que um processo de aplicação transmita dados ao UDP, o protocolo empacotará esses dados dentro de um segmento UDP e os transmitirá de forma imediata à camada de rede. Já o TCP apresenta um recurso de controle de congestionamento que impõe limite ao remetente TCP da camada de transporte, quando um ou vários enlaces entre hospedeiros da fonte e do destinatário, ficam com excesso de congestionamento. O TCP também continuará o reenvio de um segmento, até que o hospedeiro destinatário reconheça o recebimento deste segmento. Isso ocorre de forma independente do tempo que levará a entrega confiável.

A figura a seguir apresenta alguns exemplos de portas conhecidas, usadas pelo UDP. Alguns dos números de portas podem ser usados tanto pelo UDP, quanto pelo TCP.

Porta	Protocolo	Descrição
7	Echo	Ecoa um datagrama recebido de volta para o emissor
9	Discard	Descarta qualquer datagrama recebido
11	Users	Usuários ativos
13	Daytime	Retorna data e hora
17	Quote	Retorna um comentário do dia
19	Chargen	Retorna uma string de caracteres
53	Nameserver	Domain Name Services
67	BOOTPs	Servidor bootstrap
68	BOOTPc	Cliente bootstrap´
69	TFTP	Trivial File Transfer Protocol
111	RPC	Remote Procedure Call
123	NTP	Network Time Protocol
161	SNMP	Simple Network Management Protocol
162	SNMP	Simple Network Management Protocol (trap)

Figura 5 - Demonstração das portas conhecidas e utilizadas pelo UDP
Fonte: FOROUZAN, 2008, p. 710.

Os pacotes UDP, chamados datagramas de usuário, contém um cabeçalho cujo tamanho é fixo, 8 *bytes*. A figura a seguir apresenta o formato de um datagrama de usuário.

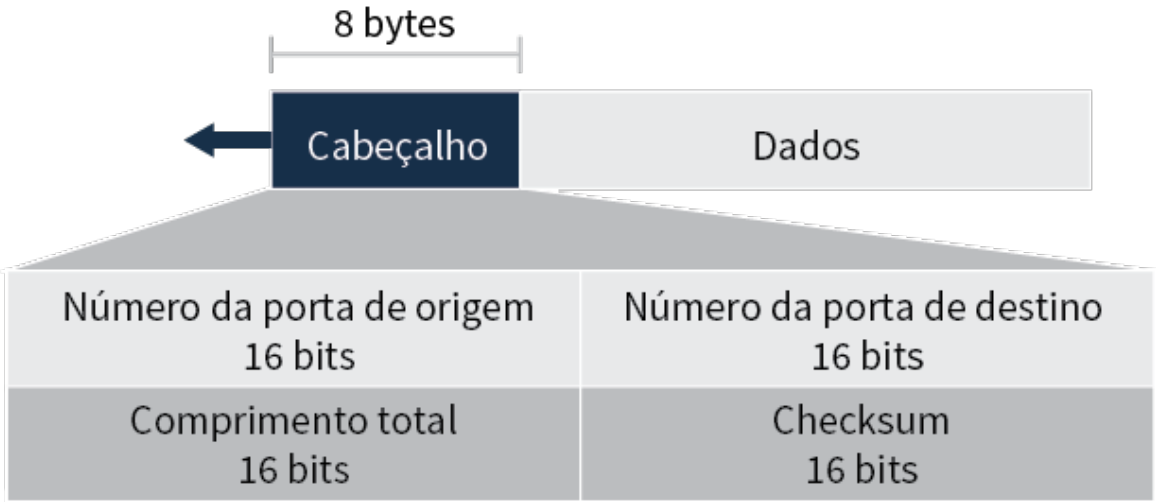


Figura 6 - Demonstração do formato de um datagrama de usuário UDP.
Fonte: FOROUZAN, 2008, p. 711.

Segundo Forouzan (2008) o formato de um datagrama de usuário possui os campos listados a seguir. Clique em cada um deles para ler a descrição.

Porta de origem

É nesse campo que é especificado o número da porta usada pelo processo em execução no *host* de origem. Com comprimento de 16 *bits*, ou seja, o número da porta pode ter uma variação entre 0 e 65535. Se o *host* de origem for um cliente, por exemplo, enviando uma solicitação, o número da porta, na maioria dos casos, é um número de porta breve solicitado pelo processo cliente e escolhido aleatoriamente pelo programa UDP que está sendo executado no *host* de origem. Agora se o *host* for um servidor, por exemplo enviando uma resposta, o número da porta, frequentemente, é um número de porta conhecido.

Porta de destino

Esse campo tem a responsabilidade de especificar o número de porta usado pelo processo que está sendo executado no *host* de destino. Esse campo também tem tamanho 16 *bits* de comprimento. Caso o *host* de destino for um servidor, ou seja, um cliente enviando uma solicitação, o número da porta, frequentemente é um número de porta conhecido. Agora se o *host* de destino for um cliente, ou seja, um servidor enviando um retorno, o número da porta, frequentemente é um

	número de porta breve. Assim sendo, o servidor realiza a cópia do número de porta breve que recebeu no pacote de solicitação.
Comprimento	O campo de 16 *bits* faz a definição do comprimento total de um datagrama UDP, englobando cabeçalho mais dados. Esses 16 *bits* podem realizar a definição de um comprimento total de 0 a 65.535 *bytes*. No entanto, o comprimento total deverá ser menor, porque um datagrama UDP deverá ser repassado em datagrama IP de comprimento igual a 65.535 *bytes*.
Checksum	Responsável pelo campo de 16 *bits*, é utilizado para a detecção de erros na transmissão de datagrama UDP, ou seja, cabeçalho mais dados.

É de conhecimento de todos que os serviços e aplicações utilizadas na internet fazem uso de protocolos. As aplicações mais populares como o correio eletrônico, *logging* remoto, a WWW e a transferência de arquivos usam o TCP, pois todas necessitam do serviço confiável de transferência de dados que o TCP proporciona. Entretanto, muitas aplicações também importantes são executadas sobre o UDP, como o roteamento com protocolo RIP (*Routing Information Protocol*). Alguns outros serviços são executados tanto com o UDP, quanto com o TCP, que é o caso de aplicações multimídia, como videoconferência, telefone pela internet, entre outros. A seguir, será apresentado um com mais detalhado o protocolo TCP.

3.3 TCP

O protocolo da camada de transporte orientado à conexão é o TCP (*Transmission Control Protocol*) e tem como objetivo proporcionar a transferência confiável de dados. Para que o TCP possa permitir o fornecimento de transferência confiável de dados, ele necessita contar com alguns conceitos como detecção de erros, retransmissão, temporizadores e campos para números de sequenciação e de reconhecimento.
O TCP é orientado à conexão justamente porque antes que um processo de aplicação possa ter seu início do envio de dados a outro, esses dois processos devem, antes de tudo, realizar uma “apresentação”, ou seja, devem fazer o envio de alguns segmentos

preliminares, um ao outro, para o estabelecimento de parâmetros da transferência de dados propriamente dita.

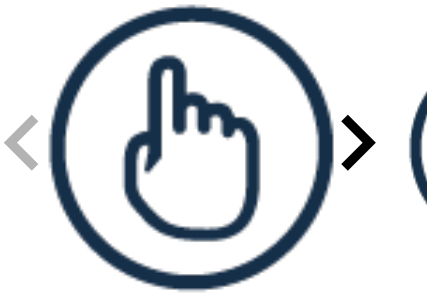
3.3.1 Formato do segmento TCP

A estrutura do segmento TCP é composta por campos de cabeçalho e um campo de dados. O campo referente a dados contém uma quantidade de dados de aplicação, o MSS impõe um limite de tamanho máximo do campo de dados de um segmento. No momento que o TCP precisa enviar um arquivo grande, como, por exemplo, uma imagem de uma página *web*, ele frequentemente "quebra" o segmento em pedaços de tamanho MSS, com exceção do último pedaço, que na maioria das vezes é de tamanho menor do que o MSS.

Quando são enviadas aplicações interativas, frequentemente realizam a transmissão de quantidades de dados menores do que o MSS. Para facilitar a compreensão, tem-se o exemplo das aplicações de *login* remoto com o Telnet, que o campo de dados do segmento TCP é, diversas vezes, de somente 1 *byte*. Em função do cabeçalho TCP ter caracteristicamente 20 *bytes* mais do que o cabeçalho do UDP, o comprimento dos segmentos enviados por Telnet pode ser de somente 12 *bytes*.

Da mesma forma que ocorre com o UDP, o cabeçalho do segmento TCP engloba números de porta de fonte e de destino, que são utilizados para processos como multiplexação e de multiplexação de dados de/para aplicações de camadas que estão acima, e conforme no UDP, está incluso um campo de soma de verificação.

Segundo Kurose e Ross (2014) um cabeçalho de segmento TCP contém os seguintes campos detalhados a seguir. Clique para ler.



Campo de número de sequência de 32 *bits* e o campo de número de reconhecimento de 32 *bits* são usados pelos TCPs, tanto na origem, quanto no destino na implementação de um serviço confiável de transferência de dados.

Como podemos constatar, o TCP é orientado para conexão, pois antes que um processo de aplicação consiga iniciar a transmissão de dados a outro, os dois processos necessitam estabelecer primeiro uma "apresentação", ou seja, devem realizar o envio de alguns segmentos preliminares um ao outro, para o estabelecimento dos parâmetros da transferência de dados propriamente dita. O próximo tópico explorará mais detalhadamente esses processos de estabelecimento e finalização de conexão do TCP.

3.3.2 Estabelecimento e finalização de conexão

Agora, vamos examinar como uma conexão TCP é estabelecida e encerrada. É interessante compreender que o estabelecimento da conexão TCP tem um impacto relevante nos atrasos percebidos, como ao realizar navegação pela *web*. E como a conexão é estabelecida? Imagine um processo que é executado no hospedeiro cliente, que necessita começar uma conexão com outro processo em um outro hospedeiro, agora servidor. O processo de aplicação do hospedeiro cliente deve informar ao TCP cliente que deseja estabelecer uma conexão com um processo no servidor. O TCP no cliente faz o estabelecimento de uma conexão TCP com o TCP no servidor seguindo alguns passos. Clique na interação a seguir para conhecê-los.

Passo 1	A parte cliente do TCP envia um segmento específico TCP ao lado servidor do TCP. Este segmento específico não inclui nenhum dado de camada de aplicação, porém um dos *bits* de *flag* no seu cabeçalho, o *bit* denominado SYN, é configurado para 1. Em função disso, este segmento é chamado um segmento SYN. Além do mais, o cliente decide de forma aleatória um número de sequência inicial e põe esse número no campo de número de sequência do segmento TCP SYN inicial. Então, este segmento é encapsulado em um datagrama IP e enviado ao servidor.
Passo 2	Assim que o datagrama IP que contém o segmento TCP SYN chegar ao hospedeiro servidor, partindo do princípio que ele chegue, o servidor retira o segmento TCP SYN do datagrama, faz alocação de *buffers* e de variáveis TCP à conexão e envia um segmento de aceitação de conexão ao TCP cliente. Este segmento de aceitação de conexão também não possui nenhum dado de camada de aplicação. Entretanto, possui três informações relevantes no cabeçalho do segmento, são elas: o *bit* SYN está com valor 1; o campo de reconhecimento do cabeçalho do segmento TCP está configurado para client_ins+1 e, por fim, o servidor decide seu próprio número de sequência inicial server_ins e coloca esse valor no campo de número de sequência do cabeçalho do segmento TCP. Em outras palavras, o segmento entende que recebeu o pacote SYN para começar uma conexão. A denominação para o segmento de concessão da conexão é segmento SYNACK.
Passo 3	No momento que recebe o segmento SYNACK, o cliente também pode reservar *buffers* e variáveis para a conexão. O hospedeiro cliente então faz o envio ao

servidor de mais um segmento. Este último segmento faz o reconhecimento do segmento de confirmação da conexão do servidor. O *bit* SYN é configurado para 0, visto que aconteceu o estabelecimento da conexão. A terceira etapa da apresentação de três vias pode realizar a condução dos dados cliente-servidor na carga útil do segmento.

Concluídos os três passos, os hospedeiros, tanto cliente, quanto servidor, podem realizar o envio de segmentos contendo dados, um ao outro.
Quanto ao encerramento da conexão, qualquer um dos dois processos que participam de uma conexão TCP está apto a encerrá-la. Quando encerrada a conexão, os recursos, ou seja, *buffers* e variáveis, nos devidos hospedeiros são liberados.

3.3.3 Controle de fluxo e congestionamento

Já é de conhecimento que o TCP oferece um serviço de transferência confiável entre dois processos que são executados em hospedeiros diferenciados. Um recurso bastante importante que o TCP disponibiliza é o seu mecanismo de controle de congestionamento.
O TCP faz uso do controle de congestionamento fim a fim, ao invés de realizar o controle de congestionamento assistido pela rede, já que a camada IP não faz o fornecimento aos sistemas finais realimentação relacionada ao congestionamento da rede.
A forma usada pelo TCP é de impor a cada remetente um limite de taxa à qual podem enviar tráfego para a sua conexão como se fosse uma função do congestionamento de rede percebido. Caso um remetente TCP ter a percepção de que existe pouco congestionamento no trajeto entre si e o destinatário, aumentará sua taxa de envio. Caso perceba que existe congestionamento ao longo do trajeto, poderá reduzir sua taxa de envio.

3.4 Camada de rede

A comunicação na camada de rede é computador-computador, ou seja, um computador localizado em qualquer lugar do planeta necessita se comunicar com outro computador em outra parte do planeta. Como, normalmente, essa comunicação ocorre por meio da Internet, o pacote enviado pelo computador remetente pode percorrer várias LANs (*Local Area Network*) ou WANs (*World Area Network*), antes de chegar até o computador de destino. Este tópico tem por objetivo apresentar como ocorre essa comunicação.

3.4.1 Princípios da camada de rede

Agora, já entendemos que a camada de transporte fornece diversas maneiras de comunicação processo a processo, de acordo com o serviço de comunicação, hospedeiro a hospedeiro, da camada de rede. Uma questão importante de ressaltar é que a camada de transporte realiza esse serviço de comunicação, sem o conhecimento de como a

camada de rede implementa esse serviço. Neste tópico, será possível entender como a camada de rede realiza o serviço de comunicação hospedeiro a hospedeiro. Será possível perceber que existe uma parte da camada de rede em cada um dos hospedeiros e roteadores, na rede, diferentemente da camada de transporte.

A figura a seguir apresenta uma rede, na qual existem dois hospedeiros denominados H1 e H2 e muitos roteadores no caminho entre o hospedeiro H1 e o Hospedeiro H2. Imagine a seguinte situação: H1 deseja enviar informações a H2. Qual a função da camada de rede nos hospedeiros H1 e H2 e roteadores? A camada de rede em H1 agarrará segmentos da camada de transporte em H1 e encapsulará cada segmento em um datagrama, ou seja, um pacote de camada de rede e só então começará a trajetória dos datagramas até chegarem ao seu destino, ou seja, transmitirá os datagramas para o roteador que faz vizinhança, no caso o R1.

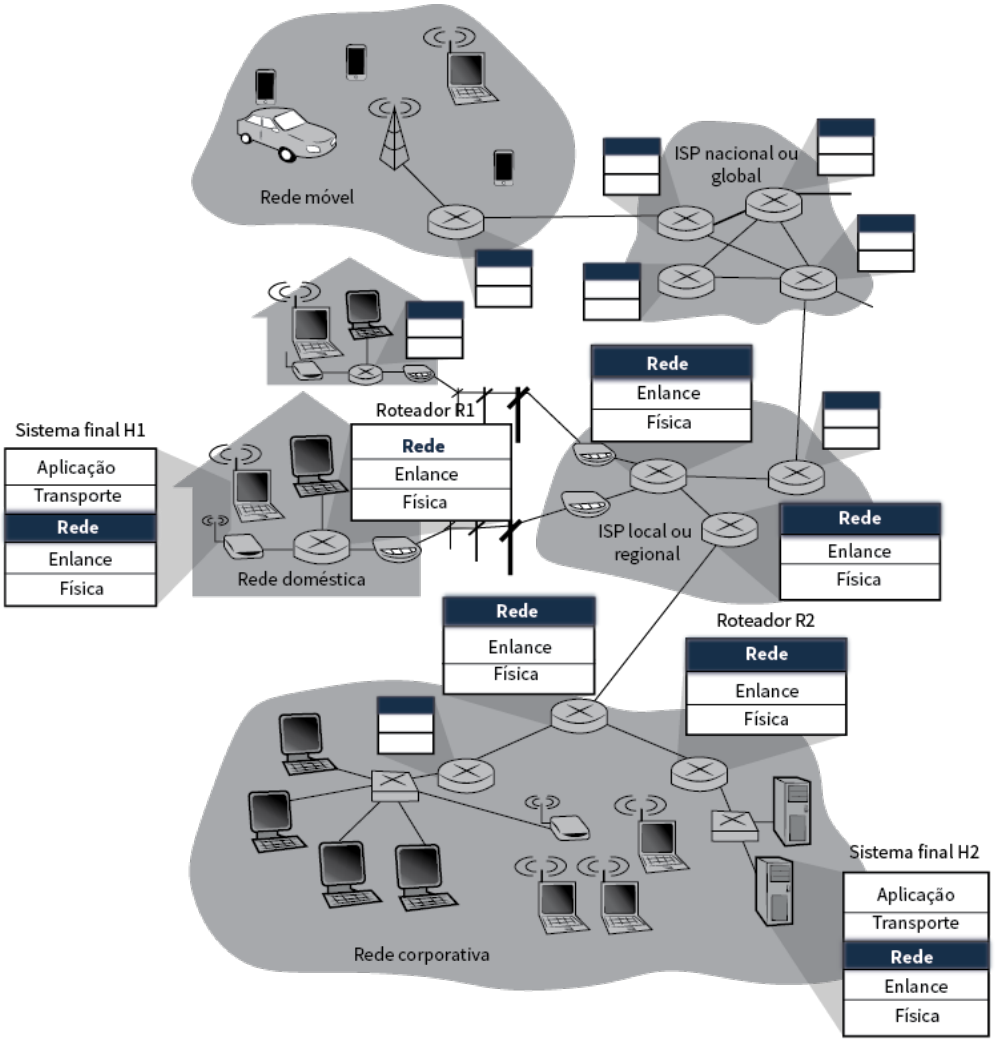


Figura 7 - Demonstração de como acontece a comunicação entre dois hospedeiros diferentes, com inúmeros enlaces no caminho.
Fonte: KUROSE; ROSS, 2014, p. 240.

No hospedeiro destinatário, o H2, a camada de rede faz o recebimento dos datagramas do roteador vizinho, no caso o R2. E qual é a função principal e fundamental dos roteadores? É o repasse de datagramas de enlaces de entrada para os enlaces de saída.
Parece que o papel da camada de rede é simplificado, transporte de pacotes de um hospedeiro emissor para um hospedeiro receptor. E, para tanto, são necessárias duas funcionalidades da camada de rede: repasse, que ocorre quando um pacote faz sua chegada no enlace de entrada de um roteador, que fará a condução até o enlace de saída mais adequado, e o roteamento, no qual a camada de rede faz a determinação da rota ou trajeto, necessários para que os pacotes fluam de um emissor a um destinatário.

VOCÊ SABIA?

Existem algoritmos que realizam o cálculo desses trajetos e são denominados algoritmos de roteamento. São os responsáveis por determinar o caminho que os pacotes fluiriam de um hospedeiro a outro hospedeiro.

O conceito repasse está relacionado a ação executada por um roteador para a transferência de um pacote de um enlace de entrada para o enlace de saída adequado. E o roteamento está relacionado ao procedimento de modo geral da rede que estabelece o trajeto fim a fim para que os pacotes sejam percorridos da origem até o destino.

3.4.2 Formato do datagrama IPv4

O endereço IPv4 tem 32 *bits* de comprimento e estabelece de forma exclusiva e universal como se dará a conexão de um dispositivo, por exemplo, um computador, ou um roteador, à internet.
Os endereços IPv4 são únicos, pois cada endereço atribui definição de uma, e somente uma, conexão com a internet.
Nunca acontecerá de existir dois dispositivos na internet com o mesmo endereço e no mesmo momento. Agora, se um equipamento que trabalha na camada de rede tiver muitas conexões com a internet, ele necessitará ter muitos endereços. Os endereços IPv4 são universalmente aceitos por qualquer *host* que deseja se conectar à internet.
Existem duas notações predominantes para indicar um endereço IPv4: notação binária e notação decimal pontuada. Na notação binária, o endereço IPv4 é apresentado como 32 *bits* de comprimento. Cada parte é composta de 8 *bits*, mais conhecido como *byte*. Por

exemplo: 01110101 10010101 00011101 00000010. Já a Notação Decimal Pontuada é apresentada para possibilitar que o endereço IPv4 seja menor e mais fácil de ser lido e compreendido. Os endereços internet, frequentemente são escritos na forma decimal, com um ponto decimal que faz a separação dos *bytes*. Por exemplo, 118.150.30.

3.4.3 Endereçamento com classe e sem classe

O endereçamento com classes é uma forma que está ficando ultrapassada. Neste tipo de endereçamento, o espaço de endereços recebe uma divisão de cinco classes: A, B, C, D e. Cada uma dessas classes ocupa alguma parte do espaço de endereços. É possível encontrar a classe de um endereço no momento que for fornecido o endereço, de duas maneiras distintas, na notação binária ou na notação decimal pontuada. Vemos isso na figura a seguir.

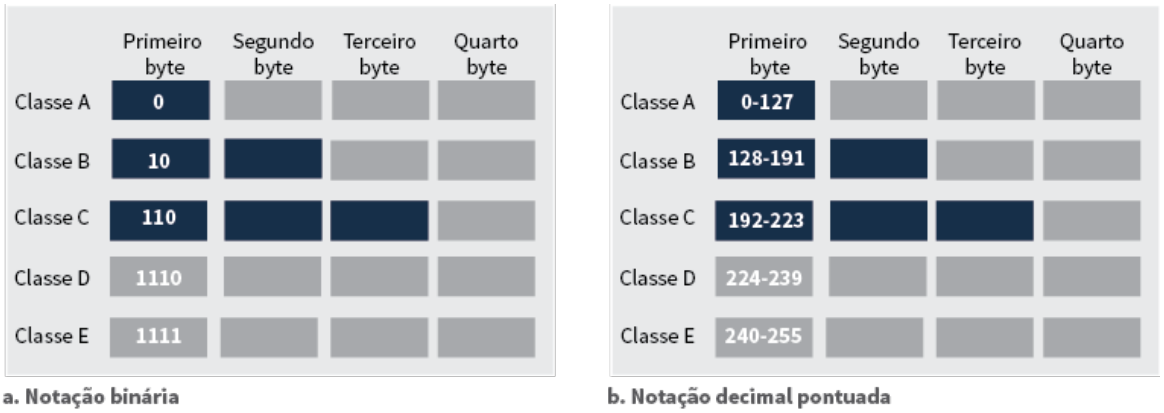


Figura 8 - O endereçamento com classe pode ser dividido em duas maneiras diferentes: anotação binária e a notação decimal pontuada.
Fonte: FOROUZAN, 2008, p. 552.

Quando um endereço for fornecido em notação binária, alguns poucos *bits*, logo no princípio já se consegue a informação da classe do endereço em questão. Se o endereço for fornecido em notação decimal pontuada, o primeiro *byte* faz a definição da classe.
Já o endereçamento sem Classes foi desenvolvido e implementado com o objetivo de superar o esgotamento de endereços e disponibilizar o acesso à internet a um número maior de organizações. Neste tipo de endereçamento, as classes não existem, entretanto, os endereços são contemplados em blocos.

CASO

O objetivo desse caso é mostrar que existe o que chamam de convenções que são utilizadas para decidir os nomes que serão usados no mecanismo e processo de tradução de nomes do DNS. Fica convencionado que, por exemplo, todas as universidades americanas estão sob o domínio .edu. Já as universidades inglesas estão sob o subdomínio .ac (acadêmico) do domínio .uk, ou seja, United Kingdom – Reino Unido. Essas convenções, muitas vezes, são definidas sem que qualquer um tome uma decisão explícita. Por exemplo, um *site* esconde o *host* exato, usado sua central de *e-mail* por trás do registro MX. Uma alternativa teria sido adotar a convenção de enviar o *e-mail* para usuário@.cs.princeton.edu, da mesma forma que se espera encontrar o diretório FTP público em ftp.cs.princeton.edu e também seu servidor *web* em www.cs.princeton.edu. Esse último é tão predominante que muitos usuários nem sequer de dão conta que essa é apenas uma convenção. Existem também convenções no nível local, onde uma organização nomeia suas máquinas de acordo com algum conjunto de regras que ache coerente. Os nomes de *hosts* vênus, saturno e marte estão entre os mais utilizados e populares na Internet. Fica fácil identificar que se trata de uma convenção de nomeação comum. Entretanto, algumas convenções de *hosts* não são tão comuns e são criativas, como o *site* que denominou suas máquinas de “ligado”, “desligado”, “falhou” e “reiniciando” entre outros, levando a declarações um pouco confusas como “reiniciando falhou” e “ligado” está desligado”. Outras opções são os usos de números inteiros que são usados para dar nomes às máquinas (PETERSON, 2013).

No método de endereçamento sem classes, no momento que uma entidade, seja ela pequena ou grande, necessitar de conexão com a internet, é disponibilizado um bloco, ou seja, um intervalo de endereços. O tamanho desse bloco pode variar de acordo com a natureza e o tamanho da entidade.

3.4.4 Endereços públicos e privados

De uma maneira simplificada e de fácil compreensão, um domínio com endereços privados está relacionado a uma rede, na qual seus endereços somente possuem

significado para equipamentos e dispositivos que fazem parte desta rede. Esses endereços não são oficialmente designados por instituições devidamente autorizadas da internet. Já os endereços públicos são de unicidade global. Porém, se os endereços privados são compreendidos somente dentro de uma determinada rede, como é que o endereçamento é administrado no momento que pacotes são recebidos ou enviados para a internet global, onde os endereços são únicos e exclusivos? Então, existe uma tabela chamada NAT (*Network Address Translation*) que realiza a tradução de endereços de rede. Segundo Tanenbaum e Wetherall (2011), a NAT realiza a atribuição a cada empresa um único endereço IP para tráfego na internet. Na empresa, todo computador tem um endereço IP exclusivo usado no roteamento interno. Entretanto, no momento em que um pacote sai dessa empresa, é necessária uma conversão do endereço IP interno, para endereço público.

Com isso, finalizamos o conteúdo que também abrangeu aspectos fundamentais do DNS e as aplicações baseadas na arquitetura P2P. Vimos, também, conceitos de introdução à camada de transporte e a utilização dos protocolos que rodam nela, o protocolo orientado à conexão, TCP e o protocolo não orientado para conexão, o UDP. Já quanto à camada de rede, estudamos o processo de implementação do serviço de comunicação hospedeiro a hospedeiro.

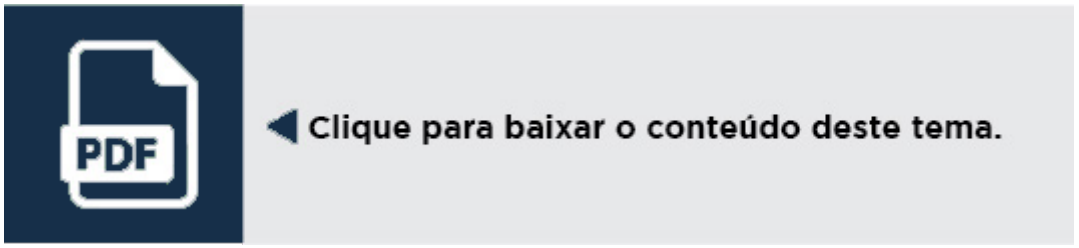
Síntese

Chegamos ao final deste capítulo. Estudamos o sistema de Domínios de nomes e domínios, que fornece serviços de importância, como a tradução de nomes hospedeiros para endereços IP, bem como aplicações P2P. Também foi possível conhecer de forma um pouco mais detalhada a camada de transporte, que tem a grande responsabilidade pela entrega de uma mensagem entre processos finais. Para que a compreensão dessa camada de transporte ocorresse de forma mais facilitada, foram abordados protocolos utilizados e de extrema importância, os protocolos UDP e TCP.

Outro ponto de destaque foi a explanação da camada de rede, que é onde a comunicação se dá *host-host*, ou seja, de computador a computador. Definiu-se conceitos fundamentais da camada de rede, tais como o endereçamento IPv4, endereçamento com classe e sem classe e endereços públicos e privados.

Neste capítulo, você teve a oportunidade de:

- introduzir termos, terminologias e conceitos básicos sobre telecomunicações e redes;
- fornecer ao aluno uma visão geral sobre a arquitetura OSI e TCP/IP;
- identificar e compreender todas as camadas e as suas funções no modelo de referência OSI e na arquitetura TCP/IP.



Bibliografia

BRANCATO, J.; FERRIS, M. **A rede**. Direção: Irwin Winkler. Produção: Irwin Winkler, Rob Cowan. Cor. (115min). Estados Unidos: 1995.

BRITO, F. T.; BRITO, F. T. **Protocolos de Comunicação**. Curitiba: LT, 2013.

CARUSO, A. A. C.; STEFFEN, F. D. **Segurança em Informática e de Informações**. São Paulo: Senac, 1999.

CASTRO, E.; HYSLOP, B. **HTML5 e CSS3**. Rio de Janeiro: Alta Books, 2013.

FEY, A. F.; GAUER, R. R. **Introdução às redes de computadores**: modelos OSI e TCP/IP. 3. ed. Caxias do Sul: ITIT, 2015.

FOROUZAN, B. A. **Comunicação de dados e Redes de computadores.** 4. ed. São Paulo: McGraw-Hill, 2008.

FURTADO, C. M. **Introdução ao DNS**: Aprenda a instalar e configurar uma infraestrutura de DNS na prática. São Paulo: Novatec, 2016.

KUROSE, J. F.; ROSS, K. W**. Redes de Computadores e a Internet - Uma**

Abordagem *Top-Down*. 6. ed. São Paulo: Pearson, 2014.
LAUDON, K. C.; LAUDON, J. P. **Sistemas de Informações Gerenciais**. São Paulo: Pearson, 2009.
MOLINARI, W. **Desconstruindo a *web***: as tecnologias por trás de uma requisição. São Paulo: Casa do Código, 2016.
MORAES, A. F. **Redes de Computadores**. São Paulo: Érica, 2014.
PETERSON, L. L.; DAVIE, B. S. **Redes de computadores: uma abordagem de sistemas**. 5. ed. Rio de janeiro: Elsevier, 2013.
PUREWAL, S. **Aprendendo a desenvolver aplicações *web*.** Rio de Janeiro: Alta Books, 2014.
SOARES, L. F. G.; LEMOS, G.; COLCHER, S. **Redes de computadores**: das LANs, MANs e WANs às redes ATM. 23ª reimpressão. Rio de Janeiro: Elsevier, 1995.
TANENBAUM, A. **Redes de computadores**. 4. ed. 17ª reimpressão. Rio de janeiro: Elsevier, 2003.
TANENBAUM, A. S. WETHERALL, D. **Redes de Computadores**. 5. ed. São Paulo: Pearson, 2011.