

SERVIÇOS INTERNET

UNIDADE 3 – CAMADA DE TRANSPORTE

Autora: *Fernanda Rosa da Silva*

Revisor: *Rafael de Jesus Rehm*

INICIAR

Introdução

Caro(a) estudante,

A camada de transporte (modelo TCP/IP) é a camada responsável pela transferência de dados entre os equipamentos da rede, independentemente das aplicações utilizadas, topologia ou configuração da rede física existente entre elas e os protocolos utilizados pelas outras camadas. A camada de transporte reúne o protocolo de transmissão ponta a ponta entre as máquinas, garantindo que as mensagens sejam transmitidas sempre que necessário. Por isso, vamos analisar os protocolos mais

utilizados na camada de transporte, a confiabilidade que ela oferece e de que forma os processos que envolvem essa camada são conduzidos e conectados com os outros processos da rede, garantindo que os pacotes sejam enviados e recebidos corretamente. Também iremos analisar conceitos importantes sobre a funcionalidade dos elementos que compõe os protocolos da camada de transporte: como os cabeçalhos e os recursos utilizados para controlar o processo de transmissão de dados.

Bons estudos!

3.1 Elementos intermediários em redes de computadores

Antes de entender de que forma a camada de transporte opera em uma rede de computadores, é importante saber que existem dispositivos intermediários que operam na infraestrutura e são responsáveis por direcionar o caminho dos dados, sem gerar ou alterar o conteúdo dos dados, operando entre as camadas de rede e transporte.

Os dispositivos intermediários que conectam as redes são chamados roteadores. O papel do roteador é selecionar o caminho e direcionar os pacotes aos seus destinos.

A Figura 1 ilustra dois computadores que operam em redes diferentes, cada um deles faz a transmissão de dados entre as camadas. Todas as informações passam pelo respectivo roteador e as mesmas camadas do modelo OSI, que exercem suas funções, até que o destino receba o pacote e seja capaz de captar a mensagem.

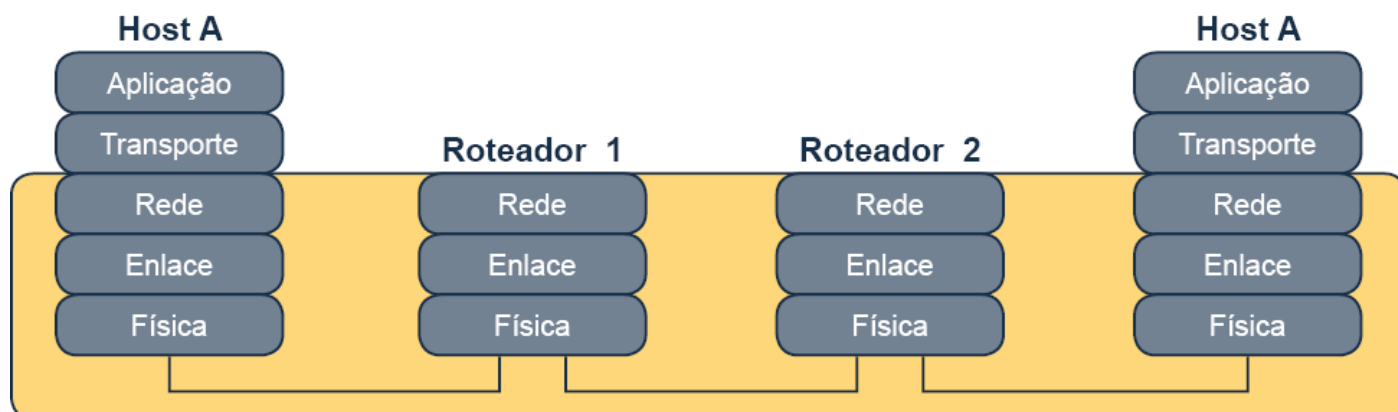


Figura 1 – Comunicação na camada de rede.

Fonte: Elaborada pelo autor, 2021.

Segundo o Teleco (2020), a principal função da camada de transporte é fornecer comunicação ponto a ponto, ou seja, conectar a aplicação do dispositivo de origem à aplicação no dispositivo de destino, independentemente do número de dispositivos intermediários utilizados entre as pontas. Como resultado, a camada de transporte deve aceitar dados de diferentes aplicativos, encapsulá-los (incluindo informações que serão usadas para controlar e identificar o aplicativo de origem e o aplicativo de destino) e, em seguida, enviar os dados encapsulados para a camada inferior.

Os equipamentos intermediários, assim como outros componentes e os protocolos que envolvem a funcionalidade das arquiteturas de rede são responsáveis por cada uma das funções estabelecidas na camada de transporte, por isso, na próxima seção, vamos analisar o papel da camada de transporte em uma rede de computadores.

Os equipamentos intermediários, por sua vez, representam a conexão entre a camada de transporte e a camada de rede. Dessa forma, a responsabilidade dos protocolos UDP e TCP, de acordo com Kurose e Ross (2014, p. 261), também inclui a ampliação do serviço de entrega do protocolo IP entre os sistemas, que caracteriza as funções de multiplexação/demultiplexação da camada de transporte.

Teste seus conhecimentos

Atividade não pontuada.

3.2 O papel da camada de transporte em redes de computadores

Assim como a comunicação verbal, forma como os humanos se comunicam entre si, os computadores também utilizam uma linguagem própria e se comunicam transmitindo dados através do uso de protocolos específicos.

No modelo OSI e na arquitetura TCP/IP, cada camada é responsável por um processo específico, pois os dados serão recebidos até chegarem à outra extremidade. De forma simples, podemos pensar nos protocolos de comunicação como uma forma de interação, que representa a forma como a comunicação ocorre no mundo real, quando duas pessoas se comunicam, por exemplo. A Figura 2 ilustra o processo citado.

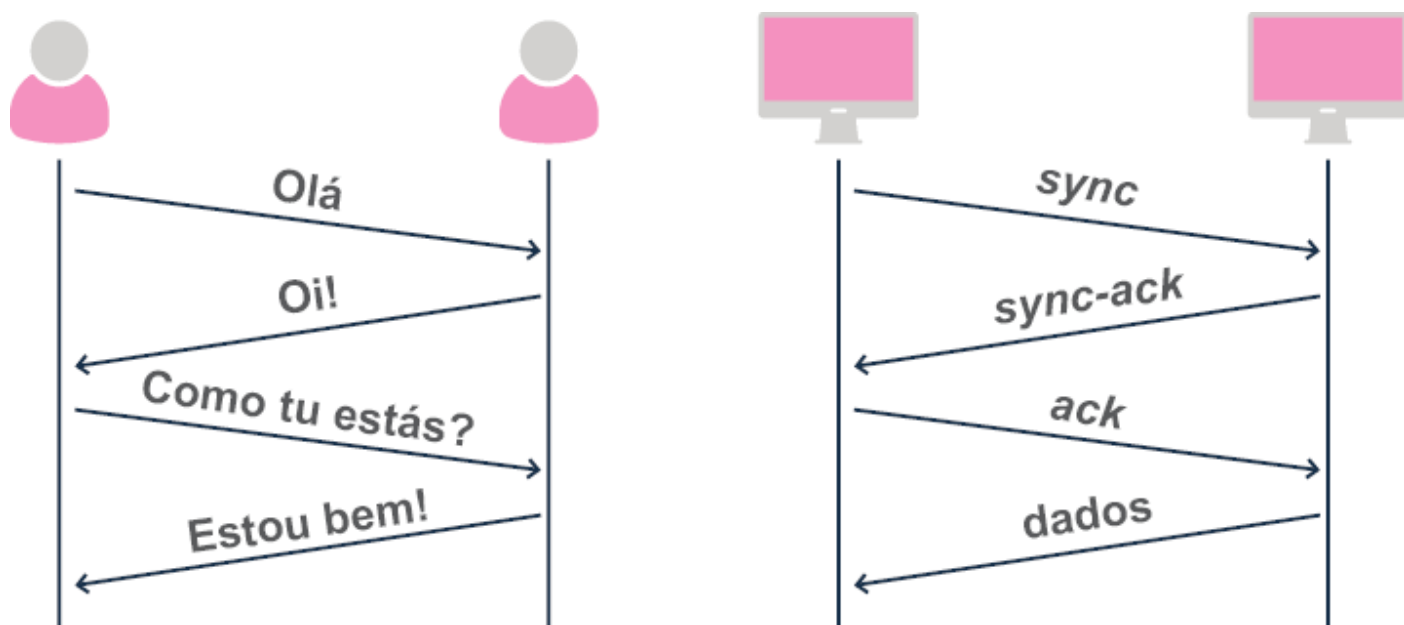


Figura 2 – Comparação da comunicação verbal e da camada de transporte.

Fonte: SCHMITT, 2013, p. 178. (Adaptada).

A abreviação SYN representa o envio de uma mensagem *synchronize* através de um pacote TCP. Esse tipo de mensagem é enviado para outro computador quando o primeiro deles precisa solicitar que uma conexão seja estabelecida.

Quando a mensagem SYN for recebida pela segunda máquina, um SYN/ACK é enviado em resposta, confirmando que a solicitação foi recebida. Na sequência, o computador de destino envia uma mensagem ACK e a conexão é estabelecida entre eles.

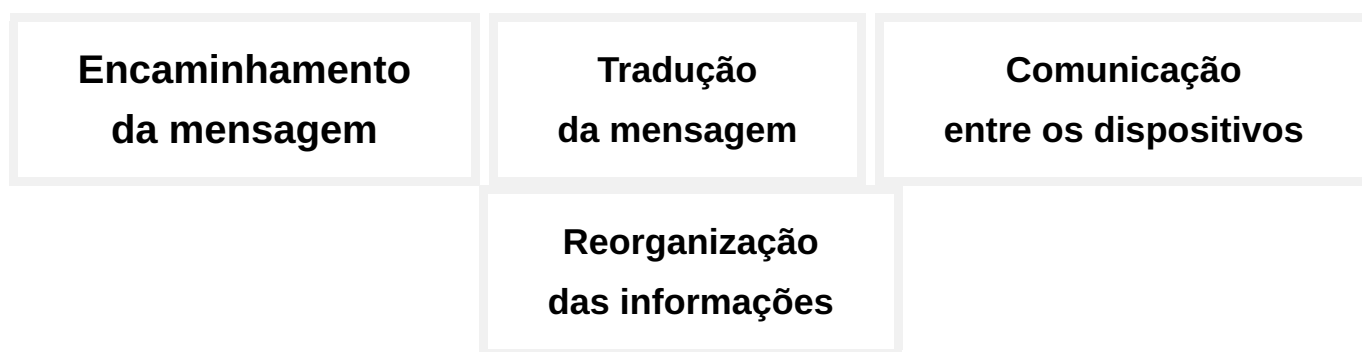
A camada de transporte representa a quarta camada no modelo OSI, que pode ser descrita como uma camada que troca todo o fluxo de mensagens entre o endereço de origem e o endereço de destino. Além disso, ela controla e encaminha mensagens utilizando a segmentação de pacotes.

Além da transmissão de dados, outro papel importante da camada de transporte é realizar a intermediação entre suas camadas inferiores, que tratam do meio físico (camada física, enlace e rede) e superiores, que tratam da aplicação oferecida ao usuário (camadas de sessão, apresentação e aplicação).

Diversos aplicativos de comunicação permitem a interação dos usuários a partir da camada de aplicação, como o *hangouts* , mas, para isso, a camada de transporte exerce uma função muito importante .

Transmissão de mensagens na rede

» Clique nas abas para saber mais sobre o assunto



A camada de transporte encaminha o pacote de dados para a camada de rede, que inclui o endereço de destino ao pacote de dados, e então passa-o para a camada de enlace, que adiciona informações de controle de erro.

A última camada que recebe a mensagem é a camada física, que realiza sua conversão. O pacote nesse momento é transmitido na forma de pulsos (eletricidade, luz ou ondas) e usa meios físicos (cabos, conectores e dispositivos intermediários) para transmitir informações para a mesma camada do dispositivo de destino.

Além do mais, a camada de transporte pode ser baseada em dois tipos de conexões, ambas com suas vantagens e desvantagens. Essas conexões são estabelecidas por meio de protocolos que têm como principal objetivo estabelecer um canal de comunicação lógico entre o aplicativo em execução no cliente e o aplicativo em

execução no servidor.

Nesse sentido, para que o papel da camada de transporte seja compreendido, considere o exemplo indicado no caso a seguir.

CASO



Durante os acessos que um site recebe, o servidor que o hospeda deve ser capaz de identificar os computadores clientes, quando as solicitações são recebidas, para respondê-las. A fim de determinar para quem as informações devem ser enviadas, a camada de transporte utiliza uma porta (ponto de conexão para um sistema operacional, associada a um endereço IP para permitir a comunicação). Cada computador possui no máximo 65.535 portas lógicas por protocolo e, para cada conexão existente, o sistema define uma porta e fecha a conexão, por meio de um canal lógico que permite a troca de dados.

Depois de processar a solicitação do programa, o protocolo da camada de aplicação se comunicará com a camada de transporte e seus protocolos. A camada de rede, por sua vez, é responsável por organizar os dados recebidos da camada de aplicação, controlando os erros e realizando o controle de fluxo de ponta a ponta.

A camada de transporte recebe os dados enviados pela camada superior, divide-os em segmentos ou datagramas e os envia para a camada imediatamente inferior. Isso pode se dar de forma confiável ou com maior agilidade, de acordo com o cenário de rede e as necessidades que a aplicação deve atender.

3.2.1 Multiplexação e demultiplexação

Diversos processos fazem parte da forma como a camada de transporte funciona em uma arquitetura de rede. Entre eles estão os processos de multiplexação e demultiplexação.

A multiplexação define o processo de coleta de dados de múltiplos sockets e envelopamento dos dados, e contém um cabeçalho que será utilizado no processo de demultiplexação; já a demultiplexação representa a entrega dos segmentos recebidos ao socket correto. A Figura 3 ilustra como isso ocorre.

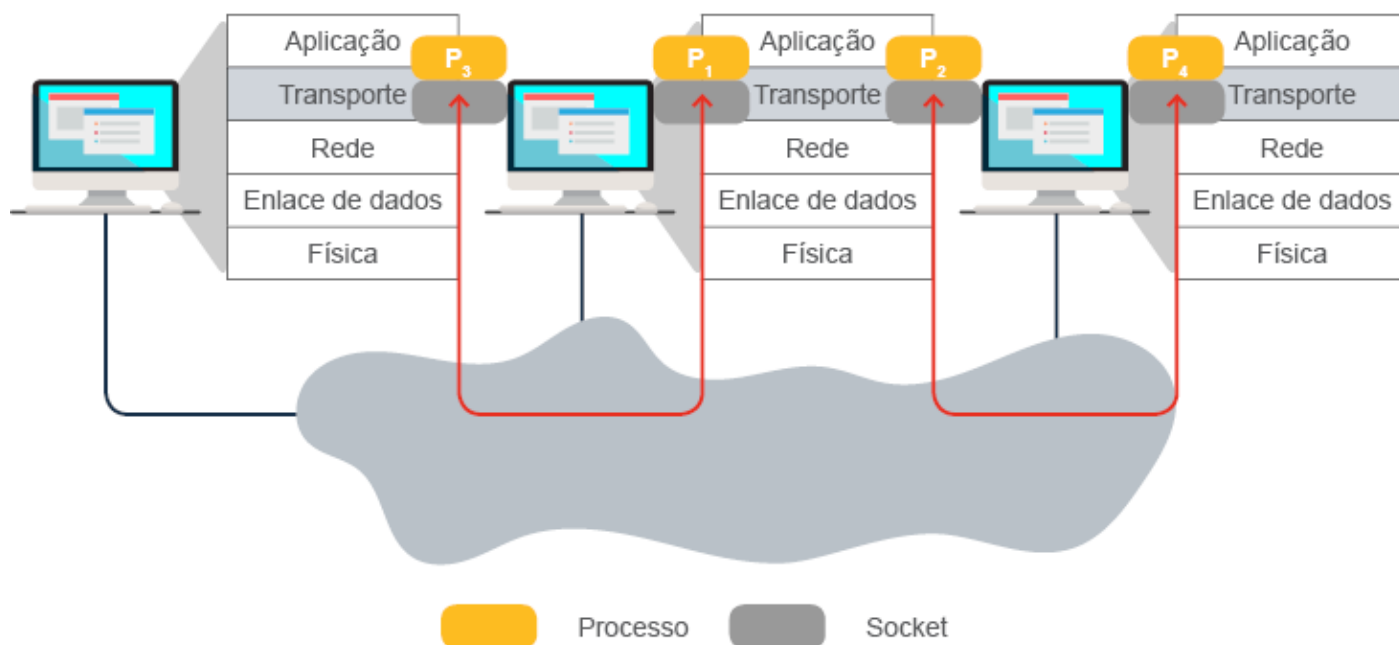


Figura 3 – Multiplexação e de multiplicação na camada de transporte.

Fonte: KUROSE; ROSS, 2014, p. 140. (Adaptada).

Enquanto a multiplexação ocorre no dispositivo transmissor, a demultiplexação ocorre no dispositivo receptor. Assim, o processo e o modo como ocorrem pode ser resumido, de acordo com Kurose e Ross (2014), conforme os cards a seguir.

Processo de multiplexação e demultiplexação

» Clique nas setas ou arraste para visualizar as imagens

Permite que a rede suporte comunicações de diferentes usuários, que podem ser combinadas (multiplexados) na mesma rede, ao mesmo tempo ou ocorrem de forma intercalada.

TRANSMISSÃO DE DADOS

Fornece os meios para que seja possível enviar e receber dados ao

executar diversas aplicações na rede.

O cabeçalho é adicionado em cada segmento para identificá-lo. A tarefa de passar os dados contidos em uma parte da camada de transporte para o socket correto define a multiplexação.

COLETA DE INFORMAÇÕES

Ainda no processo de multiplexação, é realizada a coleta de segmentos de dados de diferentes *sockets* no host de origem, encapsulando cada segmento de dados com informações de cabeçalho para criar segmentos e, em seguida, passar esses segmentos para a camada de rede.

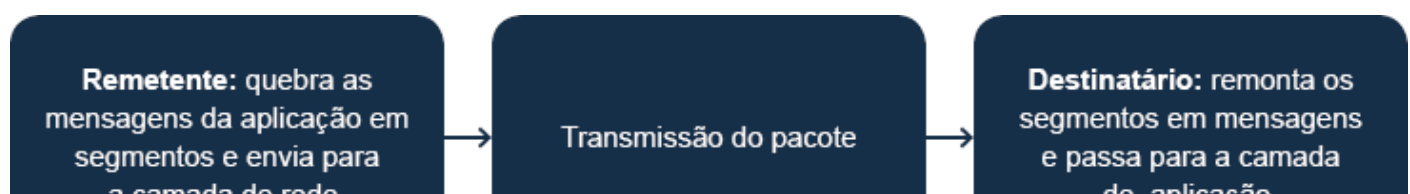
DEMULTIPLEXAÇÃO

O mesmo processo ocorrido na multiplexação, é realizado de maneira inversa sempre que a demultiplexação é executada.

3.3 Protocolos da camada de transporte

Os protocolos são, em qualquer camada, os responsáveis por fornecerem comunicação lógica entre os processos de aplicação em diferentes hospedeiros. Especificamente, na camada de transporte, eles são executados nos sistemas finais para garantir a troca de dados, tornando transparente o fato de redes que oferecem serviços e podem ser ou não orientadas à conexão. Além disso, a preocupação central do protocolo nessa camada é prover benefícios, como segurança ou desempenho durante a transmissão de dados. Sua funcionalidade se dá conforme ilustrado no Diagrama 1.

Diagrama 1 – Processo de multiplexação e demultiplexação



a camada de rede.

de aplicação.

Há mais de um protocolo de transporte disponível para as aplicações, conforme veremos nesta seção. Antes disso, vamos discutir sobre como o protocolo funciona para comunicar e transmitir dados em uma aplicação ou serviço de rede. É importante considerar que a camada de transporte de origem deve se comunicar somente com a camada de transporte de destino; porém, ela entrega os dados para a camada imediatamente superior, garantindo que o fluxo de transmissão de dados seja padronizado de acordo com a estrutura proposta pelo modelo OSI e aplicada à arquitetura TCP/IP.

CASO

No Natal, 12 crianças enviam, de locais diferentes, cartas para o Papai Noel. Elas solicitam os presentes que desejam, e as cartas são levadas ao correio pelo assistente de marketing de uma empresa. Ele enviará as cartas para o responsável pela ação. Assim, cada um dos elementos que envolve a comunicação nesse caso, podem ser descritos do seguinte modo:



- Processos = crianças;
- Mensagens da aplicação = cartas nos envelopes (que serão lidas para que o destinatário compreenda o pedido);
- Protocolo de transporte = assistente de marketing;
- Protocolo da camada de rede = serviço postal.

Quando esse processo ocorre, de algum modo é necessário garantirmos que a carta chegue ao seu destino. Para tal, analisaremos o conceito de confiabilidade na camada de transporte.

3.3.1 Confiabilidade de dados

Além de ser utilizada para iniciar e encerrar sessões, a camada de transporte é responsável por garantir a confiabilidade de dados. Porém, é necessário salientar que nem sempre isso é uma regra para a transmissão de mensagens. A confiabilidade é um fator bastante importante quando se considera a transmissão de informações mais críticas na rede, que precisam estar protegidas fim a fim, garantindo que não existam perdas ou qualquer interceptação das informações enquanto as mensagens são encaminhadas.

De forma geral, quando uma aplicação é implementada, existem prioridades que devem ser levadas em consideração para definir o nível de confiabilidade necessário. Em alguns casos, o desempenho é o fator mais importante. Dessa forma, a confiabilidade acaba ficando em segundo plano, muitas vezes sem configurar uma preocupação ou condição crítica que influencie diretamente em sua confiabilidade.

Tanto no modelo OSI, como na arquitetura TCP/IP, dois protocolos são considerados na camada de transporte, o UDP e o TCP. Vamos analisar suas características abaixo.

3.3.2 TCP e UDP: Comparação entre os protocolos

O TCP (Transmission Control Protocol – protocolo de controle de transmissão) é um protocolo de comunicação orientado à conexão da camada de transporte, cuja principal responsabilidade é verificar as informações enviadas pela rede, garantindo que elas sejam entregues na ordem correta e que não haja erros no processo de transmissão.

O protocolo IP complementa às funções principais do protocolo TCP. Ao usar esse conjunto de protocolos, aplicativos como SSH, FTP e HTTP podem ser executados corretamente na rede.

As principais características do TCP são a sua transferência de dados segura, com robustez e controle de fluxo. Além disso, o protocolo é *full duplex*, ou seja, é capaz de

enviar informações em ambas as direções e ao mesmo tempo quando dois dispositivos se comunicam na rede.

Aplicações que transferem informações a partir do uso do TCP, por sua vez, garantem que o caminho completo do pacote seja analisado. Nesse caso, quando a operação apresenta alguma falha, o problema pode ser resolvido, geralmente, com a retransmissão do pacote. A Figura 4 ilustra, de forma simplificada, o funcionamento do TCP iniciando pela segmentação dos dados na rede de origem.

Importante lembrar que quando os dados são segmentados, nada impede que eles sigam rotas diferentes, garantindo que o processo ocorra de forma mais rápida, ao invés de cada um dos segmentos aguardarem a finalização do envio do pacote anterior. O que ocorre é que os pacotes acabam chegando fora de ordem. Nesse caso, TCP é capaz de reorganizá-lo, reestruturando-o à ordem original.

Os segmentos TCP são reordenados no destino

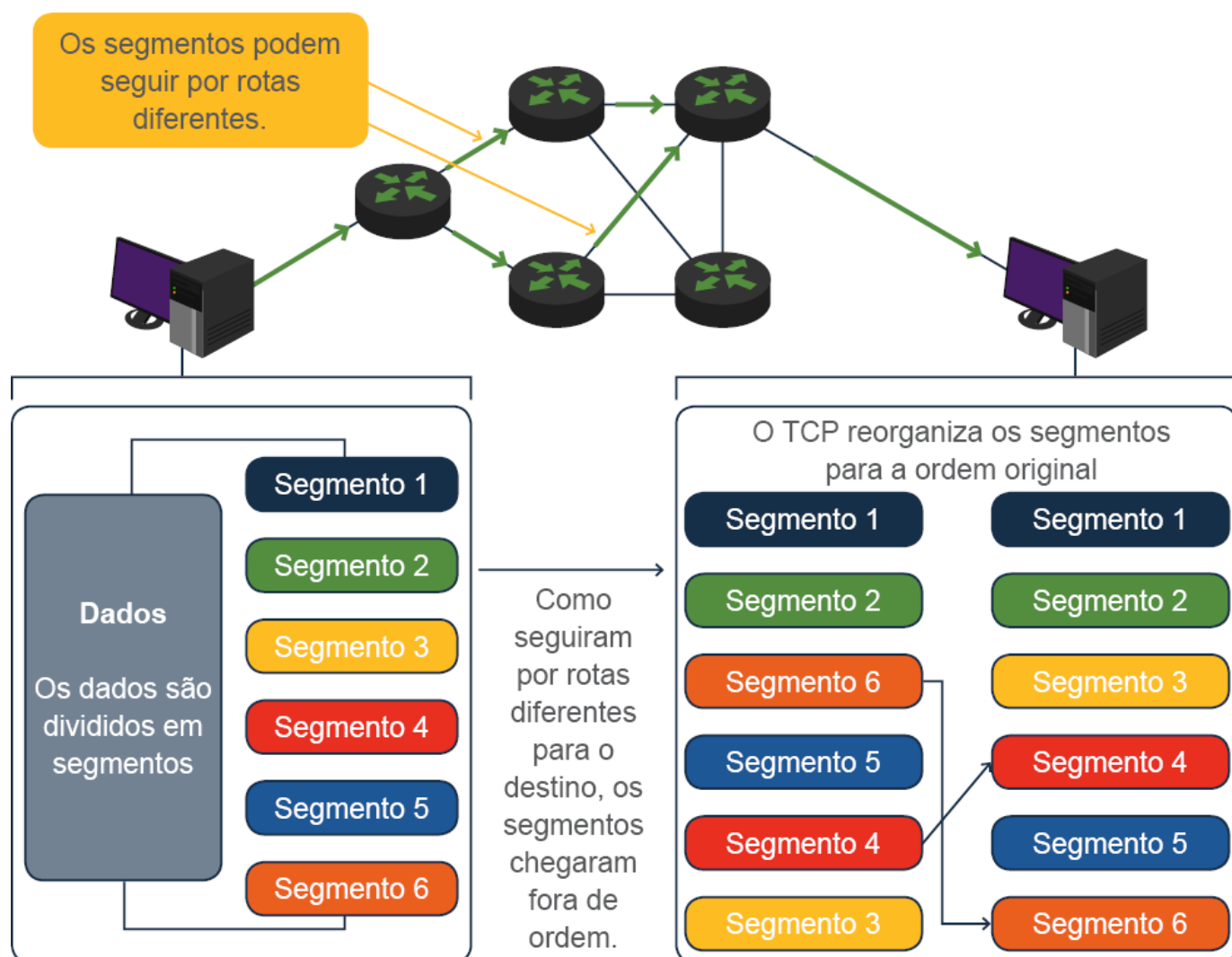


Figura 4 – Segmentação e ordenação na camada de transporte.

Fonte: KUROSE; ROSS, 2014, p. 3. (Adaptado).

Durante esse processo, o envio de cada segmento é confirmado, assim como o agrupamento dos segmentos, o que garante a ordem correta dos pacotes e sua compreensão. A entrega de dados, dessa forma, sempre é confiável e elimina segmentos duplicados, recuperando dados corrompidos. Segundo Forouzan (2010), o protocolo TCP é um protocolo *host-to-host*, isto é, que pode transmitir um pacote de um dispositivo a outro.

A principal característica que define esse protocolo é o fato de ele ser orientado à conexão (solicitação/resposta), em que, para cada conjunto de dados encaminhado do servidor para o cliente, uma resposta/notificação é emitida, alertando ao dispositivo de origem que o destino recebeu o pacote destinado a ele.

Uma das características do protocolo TCP é a forma como ele segmenta as informações, pois ele transmite o pacote de dados fracionado em pacotes menores, o que facilita a transmissão das informações. Desse modo, o pacote de dados é reorganizado e reconstruído no destino, em ordem correta, sem gerar conflitos durante todo o processo de transmissão.

Com o TCP, quando o pacote de dados chega ao destino, o aparelho analisa se a informação está correta e encaminha uma notificação ao remetente para indicar que o processo pode continuar. Do mesmo modo, ele irá notificar o remetente caso seja necessário reenviar o pacote, verificando o pacote novamente até que as informações estejam completas.

O objetivo principal desse protocolo, por sua vez, é controlar a confiabilidade da transmissão, evitar que pacotes de dados não cumpram a ordem de envio e verificar se há erros em cada pacote de dados enviado. Além disso, o protocolo TCP executa suas funções fornecendo serviços confiáveis, usando bits no cabeçalho para controlar o fluxo e receber mensagens. Observe a Figura 5 para saber como funciona o cabeçalho definido para esse protocolo.



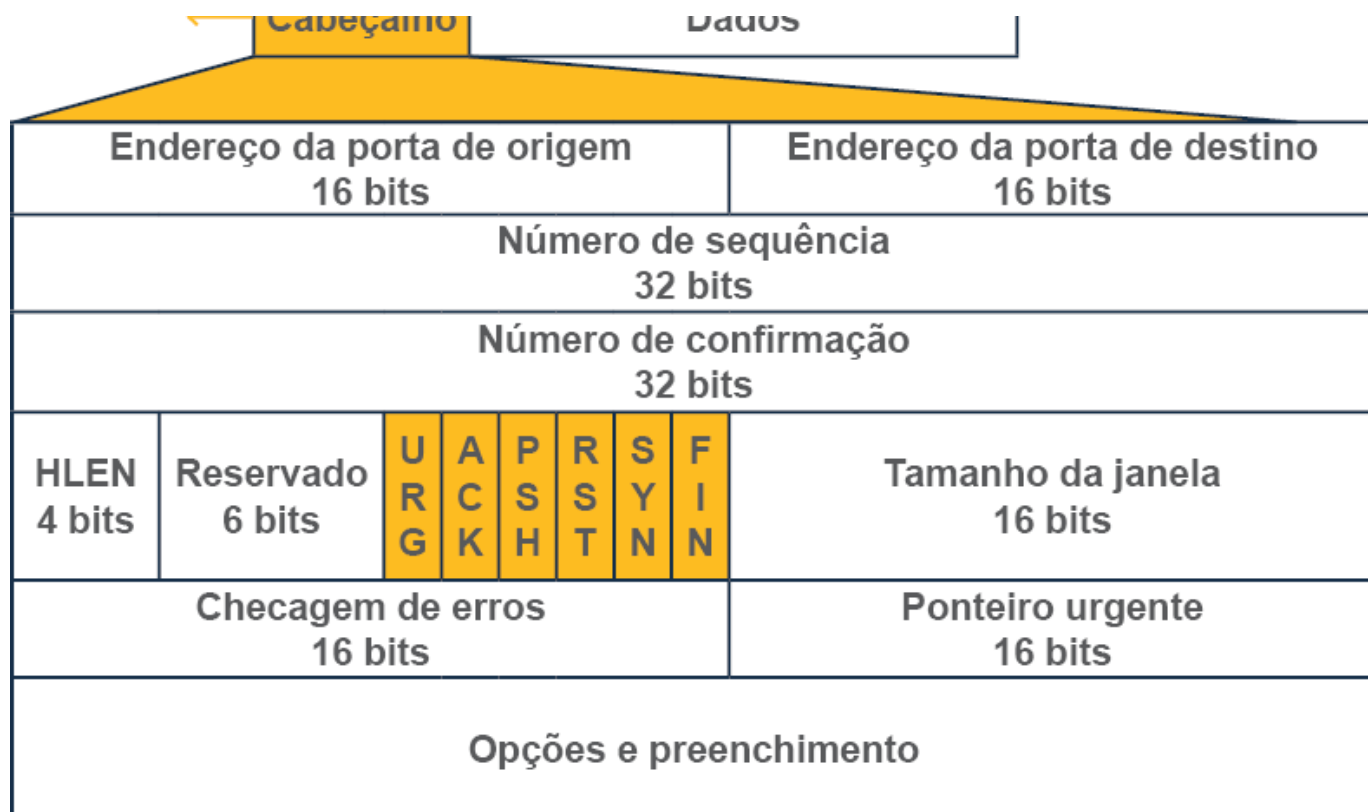


Figura 5 – Cabeçalho TCP.

Fonte: FOROUZAN, 2010, p. 721. (Adaptada).

A grande parte dos sites desenvolvidos na web usa o protocolo TCP para garantir que os usuários recebam todas as informações e possam visualizar o conteúdo de forma integrada, sem perder nenhuma informação ou alterar o formato do layout. Portanto, se houver um problema com o monitor utilizado para visualização no cliente, a retransmissão ou atualização da página deve ser capaz de restaurar a página e resolver o problema.

CASO



Quando o destino percebe que não recebeu o pacote corretamente ou que alguma parte da informação está faltando ou corrompida, ele encaminha a notificação ao dispositivo de origem, e o dispositivo de origem continuará o processo, reenviando o pacote de dados ausente antes de reinstalar as informações. O cabeçalho ainda pode ser útil para identificar qual parte do arquivo está faltando.

Controle de fluxo

O TCP provê um serviço de controle de fluxo às suas aplicações, para eliminar a possibilidade de o remetente estourar o buffer do receptor. O controle de fluxo é um serviço de correspondência de velocidade. O TCP fornece serviços de controle de fluxo, permitindo que o remetente mantenha uma variável chamada janela de recebimento.

De acordo com Kurose e Ross (2014, p. 184), as conexões TCP reservam um buffer para receber a conexão, colocando os dados nesse buffer para que a sequência possa ser reestabelecida. Essa característica define o serviço de controle de fluxo. Além disso, eles apontam que o buffer é definido por duas variáveis:

- **LastByteRead:** número do último byte na cadeia de dados lido do buffer pelo processo de aplicação em B;
- **LastByteRcvd:** número do último byte na cadeia de dados que chegou da rede e foi colocado no buffer de recepção de B.

Ao considerar quais cenários são ideias para o uso de cada um dos protocolos citados, é correto dizer que o protocolo TCP deve ser explorado para cenários em que se é necessário garantir a integridade dos dados ou a ordem absoluta da informação transmitida.

CASO



Por exemplo, ao fazer o download de um arquivo, na maioria dos casos, podemos modificá-lo enquanto o conteúdo é baixado. Esse processo é realizado para resolver problemas que ocorram na transferência do pacote, retomando o processo quando necessário, para substituir ou adicionar automaticamente a parte transferida incorretamente do arquivo. Isso significa que o processo não é tolerável a falha, por isso o TCP é utilizado para garantir que se possa corrigir os problemas identificados.

Todos os processos seguidos pelo TCP, para manter a integridade dos dados, pode acabar prejudicando o desempenho das aplicações. Nesse caso, vamos analisar o protocolo UDP, que também funciona na camada de transporte.

O protocolo UDP (*User Datagram Protocol* – protocolo de controle de transmissão) não é orientado à conexão, portanto, sua entrega não é confiável. Sua estrutura não usa confirmação para garantir que o datagrama tenha chegado ao seu destino, nem classifica as mensagens recebidas. Sendo assim, datagramas UDP podem ser perdidos, duplicados ou inutilizáveis a qualquer momento durante a transmissão, o que não é, necessariamente, uma preocupação para todas as aplicações.

Além disso, no processo de serviço prestado pelo protocolo UDP, a quantidade de informações trocadas é menor que a do protocolo TCP, portanto, a velocidade é maior. Kurose e Ross (2014, p. 38) para ilustrar como isso funciona, cita o transporte rodoviário e a forma como os veículos seguem o caminho desejado até o destino, sem se preocupar com qual rota é a mais confiável, e sim a mais rápida.

CASO



Se uma fábrica precisar transportar uma carga grande até um destino específico e geograficamente distante, será necessário dividir a carga em vários caminhões para que a entrega completa do produto seja concluída, considerando a capacidade de cada veículo. Ao chegarem ao destino, os produtos são descarregados e separados de acordo com a carga. Para pacotes de dados na rede, segue-se o mesmo processo. Após transmiti-los ao destino de forma segmentada, eles devem ser remontados para que as informações sejam transmitidas corretamente. De acordo com Forouzan (2010), entretanto, isso pode não ser tão efetivo com o uso do protocolo UDP, uma vez que ele não prioriza a confiabilidade na rede.

Embora o protocolo UDP não seja tão confiável quanto o TCP, é um protocolo mais rápido. O protocolo apenas envia informações, sem se preocupar se elas são recebidas ou não, eliminando assim o bit de controle, tornando seu cabeçalho mais simples (além de não haver uma descrição detalhada como é a utilizada pelo protocolo TCP).

Ao usar UDP, portanto, deve ser suficiente que o remetente envie o pacote ao destino desejado. Nenhuma confirmação é enviada para garantir da origem para o destino, com a intenção de verificar se o pacote realmente foi entregue. Aplicações que enviam dados em tempo real, não enfrentam problemas com o pacote pela falta de qualquer bit, por isso o UDP acaba sendo ideal para esses casos. Consulte o cabeçalho do protocolo UDP padrão na Figura 6.

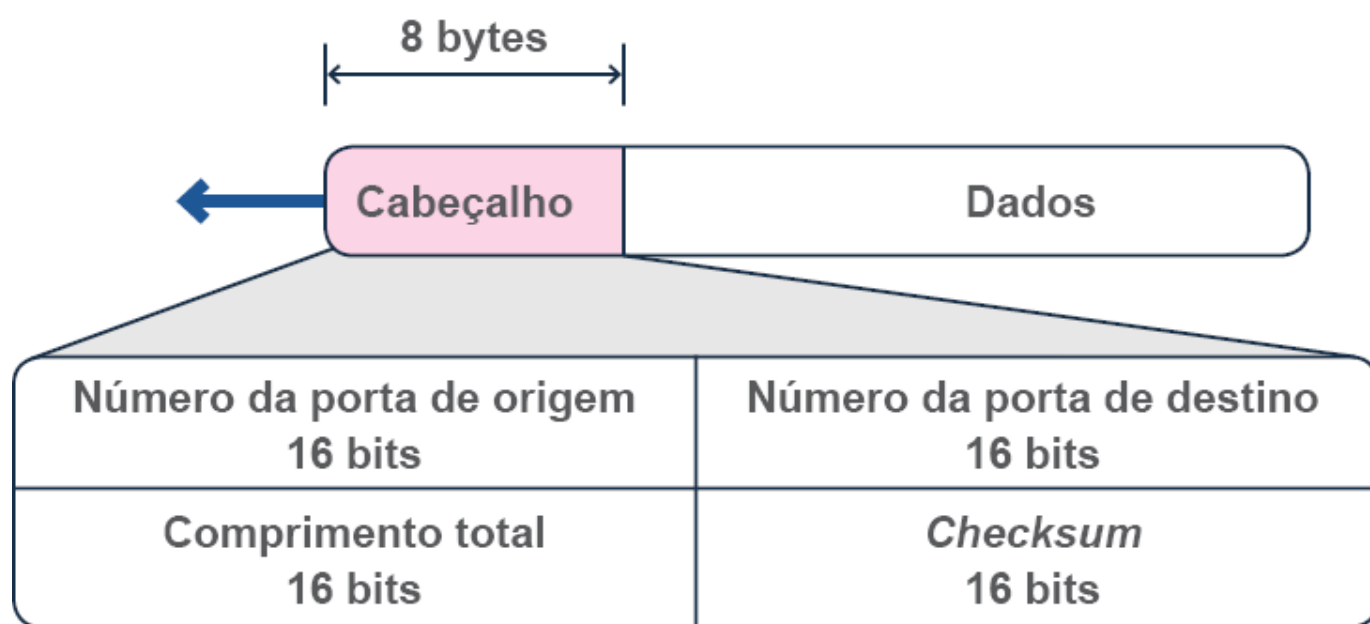


Figura 6 – Cabeçalho UDP.

Fonte: FOROUZAN, 2010, p. 711. (Adaptada).

Em termos de funcionalidade, o protocolo UDP também divide as informações em pacotes menores, mas ao reconstruir o pacote não investe tempo para analisar a integridade dos dados, sem se preocupar ou tomar qualquer medida para corrigir as perdas ocorridas.

Enquanto isso, o TCP é orientado à conexão, e utiliza um campo denominado *acknowledgment* (o termo pode ser traduzido como confirmação) para validar a

entrega dos dados.

Ao comparar os protocolos dessa camada, é óbvio que o protocolo TCP é mais robusto do que o protocolo UDP e usa mais recursos de computação através do método de transmissão de rede para atingir seus objetivos. Essa visualização pode ser verificada com clareza simplesmente avaliando os cabeçalhos de cada protocolo que compõem a camada de transporte.

CASO



Um exemplo de aplicação que utiliza o protocolo UDP para transmissão são as aplicações baseadas em *streaming* em tempo real, em que perdas de pacotes não são relevantes e os pacotes são enviados em taxa constante, independentemente de congestionamentos que possam acontecer na rede ou em sua capacidade da aplicação.

Considerando que o protocolo TCP utiliza mecanismos específicos de controle e segurança, enquanto o UDP não utiliza mecanismos para esse fim, pode-se dizer que o UDP torna o processo mais ágil e flexível, embora as etapas de segurança sejam ignoradas. Quando é necessário avaliar o método mais adequado, pode-se concluir que o TCP pode garantir a integridade dos dados ou a ordem absoluta da transmissão das informações, enquanto o UDP não pode atender a esse objetivo.

A tolerância a falhas, nesse sentido, é uma desvantagem para quem utiliza o protocolo UDP, por isso, embora o UDP seja mais leve, não é adequado para situações em que problemas relacionados à informação não podem ser tolerados. É cada vez mais usado em ambientes e aplicativos desconhecidos, como jogos online e *streamings* de vídeo e voz.

Ao analisar e processar as informações para enviar e monitorar corretamente o pacote, em comparação com o protocolo UDP, o TCP leva mais tempo para processar os pacotes de dados.

Segundo Almeida (2017), ao utilizar o protocolo TCP, o processo inclui o envio de pacotes de dados, a destruição dos dados a serem enviados e o envio dos dados na ordem correta, bem como contempla a utilização de um identificador único para confirmar o recebimento. No entanto, durante esse tempo, os pacotes de programas podem se acumular e causar perdas significativas de desempenho em jogos *multiplayer*, por exemplo, em que os jogadores buscam agilidade; já para outras aplicações, em que perder dados não representa um problema crítico, torna-se mais eficiente o uso do protocolo UDP.

Em aplicações em tempo real, por exemplo, como em chamadas de vídeo, o descarte de pacotes muitas vezes nem é percebido. Dessa forma, considerando que a prioridade é receber os dados o mais rápido possível, um protocolo rápido é o mais adequado.

Ao usar o TCP, um número de sequência é usado para permitir que os segmentos sejam agrupados na ordem original ao chegarem ao seu destino, tornando o meio mais confiável. Sendo assim, considerando as principais diferenças entre os protocolos, vamos analisar a funcionalidade de cada um deles.

3.3.3 Outros protocolos

Além dos protocolos apresentados anteriormente, existem outros protocolos que atuam na camada de transporte. Apesar de não serem mais utilizados, vamos analisar rapidamente suas características principais. Por meio da descrição de cada um deles é possível também acompanhar a evolução dos protocolos utilizados nesta camada em relação à forma que os dados eram transferidos anteriormente e como são tratados com o uso dos protocolos UDP e TCP atualmente.

IPX

O protocolo IPX (*Internet Protocol Exchange*) é um protocolo de transmissão de pacotes de dados na Internet desenvolvido pela Novell, semelhante ao protocolo padrão de transferência de dados da Internet TCP/IP, o protocolo IPX é utilizado para

transferência de pacotes. O protocolo IPX (*Internet Protocol Exchange*) é um tipo de protocolo que não precisa de uma conexão constante para que os dados sejam transmitidos. O IPX é usado não só para entregar pacotes de dados entre computadores na Internet, mas também para fornecer o endereçamento de rede, bem como o roteamento da internet para computadores, arquivos e servidores de impressão.

NetWare

NetWare é um protocolo de transferência de arquivos utilizado em servidores, desenvolvido pela Novell para possibilitar o compartilhamento de arquivos e impressoras de maneira confiável e fácil de gerenciar nos PCs.

Funções adicionais da camada de transporte

Além de todos os protocolos utilizados e as funções estabelecidas à camada de transporte a partir da funcionalidade de cada um deles, por último, a camada de transporte também é capaz de definir portas para que a camada de aplicação possa garantir a interação dos usuários e a utilização dos serviços que precisam se comunicar na rede. Os números de portas são utilizados tanto pelo protocolo TCP como pelo UDP para diferenciar cada um dos aplicativos e isso reflete diretamente na forma como o usuário recebe as informações.

A camada de transporte define a porta 110, por exemplo, para o protocolo POP3, 80 para o HTTP e 531 para o IM (mensagem instantânea) utilizado para troca de mensagens em aplicativos de bate-papo na Internet.

Esse detalhamento pode ser mais bem compreendido por meio do Quadro 1, que mostra algumas aplicações comumente utilizadas na Internet e o protocolo de transporte utilizado por cada uma delas.

Quadro 1 – Aplicações e protocolos da camada de transporte

APLICAÇÃO	PROTOCOLO DA	PROTOCOLO DE
	CAMADA DE APLICAÇÃO	TRANSPORTE UTILIZADO

Correio eletrônico	SMTP	TCP
Acesso ao terminal remoto	Telnet	TCP
Web	HTTP	TCP
Transferência de arquivo	FTP	TCP
Servidor de arquivo remoto	NFS	Tipicamente UDP
Recepção de multimídia	Tipicamente proprietário	UDP ou TCP
Telefonia por internet	Tipicamente proprietário	UDP ou TCP
Gerenciamento de rede	SNMP	Tipicamente UDP
Protocolo de roteamento	RIP	Tipicamente UDP

Tradução de nome**DNS****Tipicamente UDP**

Fonte: KUROSE; ROSS, 2014, p. 47. (Adaptado).

Teste seus conhecimentos

Atividade não pontuada.

3.3.4 Handshake na camada de transporte

Em detalhe, as operações realizadas pela camada de transporte segundo Moraes (2010), em relação ao que já vimos sobre a comunicação dos dispositivos na Figura 3, podem ser descritas conforme a linha do tempo a seguir.

Processos de *handshake* na camada de transporte

» Clique nas setas ou arraste para visualizar as imagens

Dois computadores ou sistemas iniciam a comunicação e identificam seu endereço IP e o caminho pelo qual o pacote será enviado (camada de aplicação).

2

Posteriormente, a camada de transporte realiza a troca de informações entre os dispositivos sem se preocupar com o roteamento (implementado posteriormente na camada de rede).

3

Na sequência, a camada de transporte realiza o controle do fluxo, ou seja, se o receptor não for capaz de receber a mensagem ou ocorrer

retransmitidos, com o objetivo de entregar a mensagem de forma íntegra.

Nesse ponto, podemos definir o *handshake* (aperto de mão) como o processo pelo qual duas ou mais máquinas afirmam que reconheceram umas às outras e estão prontas para iniciar a comunicação, dando início a transmissão de dados. O *handshake* é utilizado em protocolos de comunicação, tais como: FTP, TCP, HTTP, SMB e SMTP.

Dentro desse conceito, existe o termo **handshake triplo** (*three-way-handshake*), esse processo é necessário porque os números de sequência não são vinculados a um relógio global na rede e os protocolos TCP podem ter mecanismos diferentes para captar o ISN (*Initial Sequence Number*). Esse processo pode ser mais bem detalhado a seguir.

Comunicação *handshake* na camada de transporte

» Clique nas abas para saber mais sobre o assunto



A camada de transporte é responsável por estabelecer uma sessão de comunicação temporária entre os dois aplicativos e fornecer dados entre eles. Desempenha um papel básico na transmissão de pacotes de dados, independente da aplicação que está sendo utilizada, da topologia ou das configurações que envolvam o meio e os dispositivos conectados entre si através da rede, oferecendo as funções necessárias ao nível de aplicação da arquitetura de rede.



VAMOS PRATICAR?

Vamos utilizar o *wireshark* para descobrir se uma aplicação utiliza o protocolo TCP ou UDP para transportar as informações. A ferramenta pode ser baixada no link: < <https://www.wireshark.org/#download> >.

- Escolha uma aplicação da sua escolha que já utiliza ou que queira analisar o protocolo da camada de transporte utilizado na sua estrutura;
- Prepare *wireshark* para capturar pacotes de dados usando filtros TCP e UDP e rode a aplicação;
- Execute a aplicação e analise os pacotes que serão capturados durante sua utilização. Você pode brincar com os filtros para encontrar facilmente os pacotes que têm relação com a aplicação que foi escolhida;
- Escreva um pequeno resumo sobre o protocolo utilizado e informações relevantes que você filtrou no laboratório, explicando porque na sua opinião, esse protocolo é utilizado.

Síntese

Nesta unidade, abordamos os tipos de protocolo e funções que definem as funcionalidades. Além disso, analisamos suas respectivas vantagens e desvantagens, analisando de que forma podem ser úteis para priorizar a segurança e confiabilidade dos dados, assim como o desempenho. Podemos concluir que o comportamento de cada protocolo na camada de transporte e as funções vinculadas relacionadas à camada de transporte, assim como a diferença entre eles e os cenários ideais para aplicação de cada um de acordo com as prioridades de cada serviço e a forma como ele precisa ser entregue para a camada de aplicação à sua estrutura são definidas pela forma como está conectado às camadas de rede e de aplicação. Abordamos conceitos importantes que definem a prioridade dos aplicativos e a forma como as

mensagens são transmitidas pela rede e pelos dispositivos intermediários até chegarem ao seu destino.

SAIBA MAIS



Título : Redes de Computadores. Autor : James F. Kurose e Keith W.

Ross Ano : 2005 **Comentário :** Esse livro aborda diversos conceitos de

Redes de computadores, apresentando conceitos e o capítulo 3 aborda conceitos importantes sobre a camada de rede incluindo os serviços, princípios de transferência confiável e gerenciamento de conexão.

Onde encontrar : Biblioteca Virtual da Ânima.



Título : A internet e seu impacto nos processos de recuperação da informação

Autor : Cenidalva Miranda de Sousa Teixeira e Ulrich Schiel

Ano : 1997

Comentário : Esse conteúdo retrata a importância da Internet para atender as necessidades das aplicações e na transmissão de informações, descrevendo os fundamentos da recuperação da informação e considerações importantes sobre a Internet.

Onde encontrar: https://www.scielo.br/scielo.php?script=sci_arttext&

pid=S0100-19651997000100009

Referências bibliográficas

ALMEIDA, T. C. Desenvolvimento de jogos em rede: Protocolo UDP. **Fábrica de Jogos**, 26 maio 2017. Disponível em: < <https://www.fabricadejogos.net/posts/desenvolvimento-de-jogos-em-rede-protocolo-udp/> >. 2017. Acesso em: 04 nov. 2020.

FOROUZAN, B. A. **Comunicação de dados e redes de computadores**. 4. ed. Porto Alegre: Bookman, 2010.

KUROSE, J. F.; ROSS, K. W. **Redes de computadores e a internet** : uma abordagem top-down. 6. ed. São Paulo: Pearson Addison Wesley, 2014.

MORAES, A. F. **Redes de computadores** : fundamentos. 7. ed. São Paulo: Saraiva, 2010.

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

TELECO. **Qualidade de serviços (QoS)** : conceitos de rede. [s. l.], [s. d.]. Disponível em: < https://www.teleco.com.br/tutoriais/tutorialqosotm/pagina_2.asp >. Acesso em 04 dez. 2020.
