2023

Introdução às redes

Antonio Fernando Traina

Módulo 9 – Camada de transporte

Índice

9 - Camada de transporte

9. Introdução

9.0. O que vou aprender neste módulo?

Título do módulo: Camada de transporte

Objetivo do módulo: Comparar as operações de protocolos de camada de transporte no suporte da comunicação de ponta a ponta.

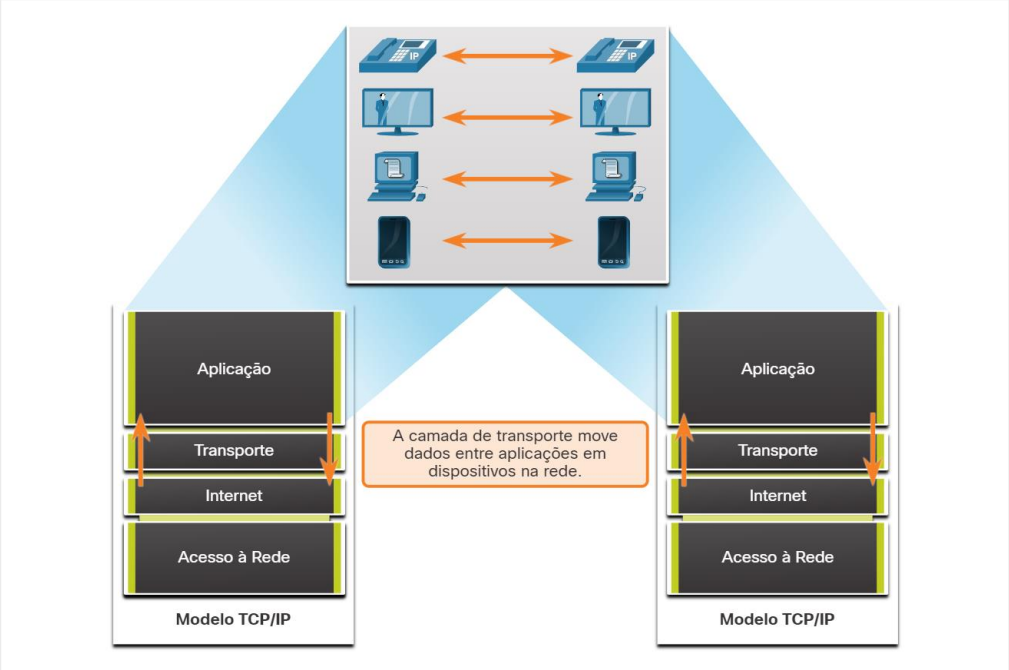
Título do Tópico	Objetivo do Tópico
Transporte de dados	Explicar a finalidade da camada de transporte no gerenciamento do transporte de dados em comunicação de ponta a ponta.
Visão geral do TCP	Explicar as características do TCP.
Visão Geral do UDP	Explicar as características da UDP.
Números de porta	Explicar como TCP e UDP usam números de porta.
Processo de comunicação TCP	Explicar como o estabelecimento e encerramento da sessão TCP processa Facilitar uma comunicação fiável.
Confiabilidade e controle de fluxo	Explicar como as unidades de dados do protocolo TCP são transmitidas e reconhecidas para garantia de entrega.
Comunicação UDP	Comparar as operações dos protocolos da camada de transporte no suporte comunicação de ponta a ponta.

9.1 Transporte de Dados

9.1.1 Propósito da Camada de Transporte

Os programas da camada de aplicação geram dados que devem ser trocados entre os hosts de origem e de destino. A camada de transporte é responsável pela comunicação lógica entre aplicativos executados em hosts diferentes. Isso pode incluir serviços como o estabelecimento de uma sessão temporária entre dois hosts e a transmissão confiável de informações para um aplicativo.

Como mostra a figura, a camada de transporte é o link entre a camada de aplicação e as camadas inferiores que são responsáveis pela transmissão pela rede.



A camada de transporte não tem conhecimento do tipo de host de destino, o tipo de mídia pela qual os dados devem percorrer, o caminho percorrido pelos dados, o congestionamento em um link ou o tamanho da rede.

A camada de transporte inclui dois protocolos:

- Protocolo TCP
- Protocolo UDP (User Datagram Protocol)

9.1.2 Responsabilidades da Camada de Transporte

A camada de transporte tem muitas responsabilidades.

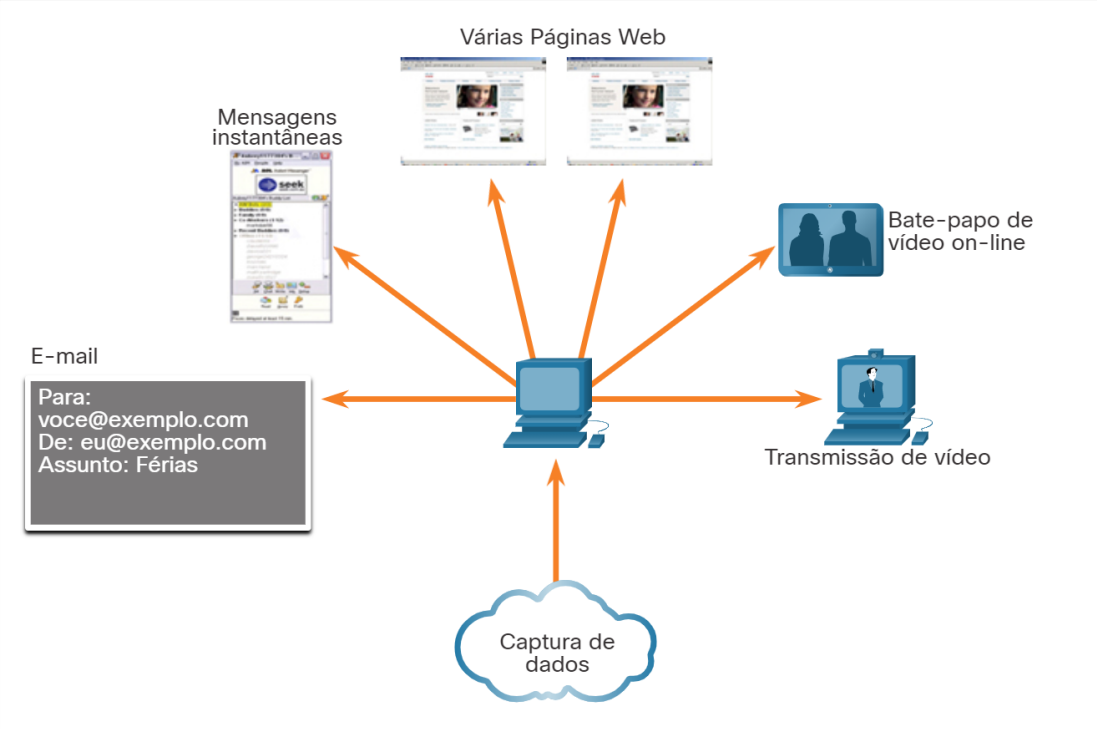
Rastreamento de Conversações Individuais

Na camada de transporte, cada conjunto de dados que flui entre um aplicativo de origem e um aplicativo de destino é conhecido como conversa e é rastreado separadamente. É responsabilidade da camada de transporte manter e monitorar essas várias conversações.

Como ilustrado na figura, um host pode ter vários aplicativos que estão se comunicando pela rede simultaneamente.

A maioria das redes tem uma limitação da quantidade de dados que pode ser incluída em um único pacote. Portanto, os dados devem ser divididos em partes gerenciáveis.

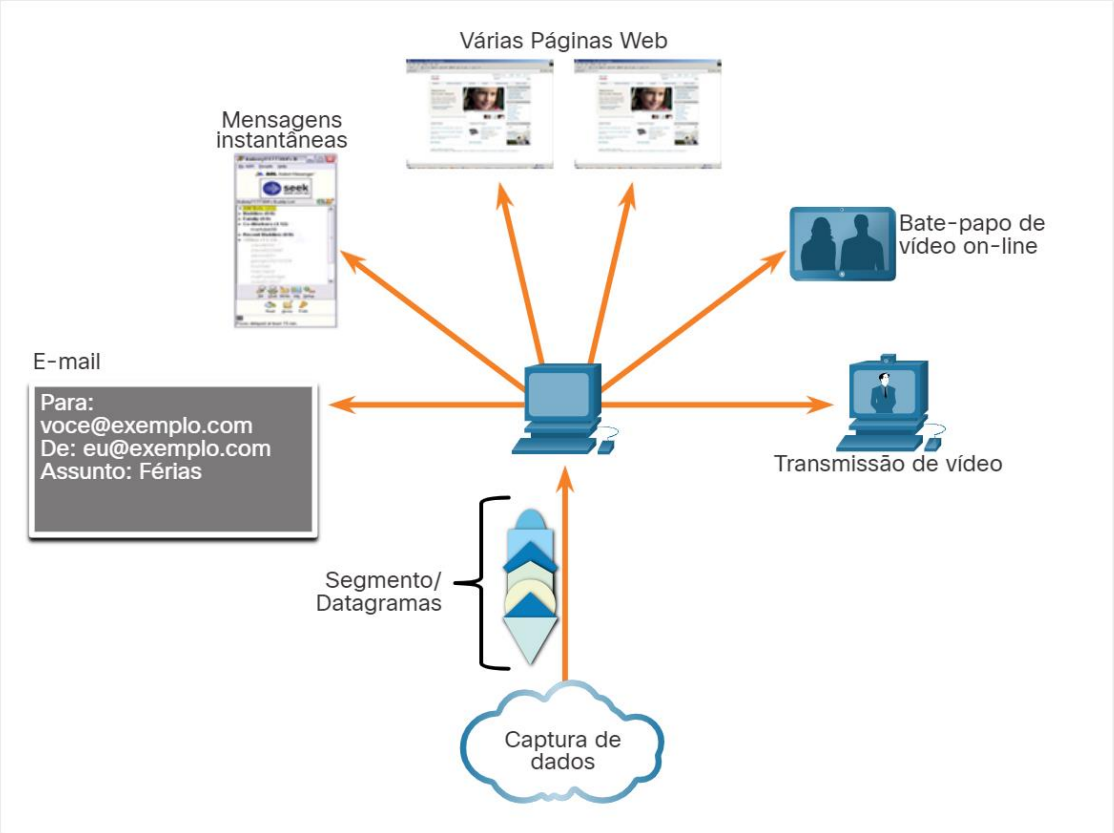
O PC executa simultaneamente vários aplicativos de rede, incluindo um cliente de e-mail, cliente de mensagens instantâneas, páginas da Web do navegador da Web, streaming de vídeo e um cliente de videoconferência.



Segmentação de Dados e Remontagem de Segmentos

É responsabilidade da camada de transporte dividir os dados do aplicativo em blocos de tamanho adequado. Dependendo do protocolo de camada de transporte usado, os blocos de camada de transporte são chamados de segmentos ou datagramas. A figura ilustra a camada de transporte usando blocos diferentes para cada conversa.

A camada de transporte divide os dados em blocos menores (ou seja, segmentos ou datagramas) que são mais fáceis de gerenciar e transportar.

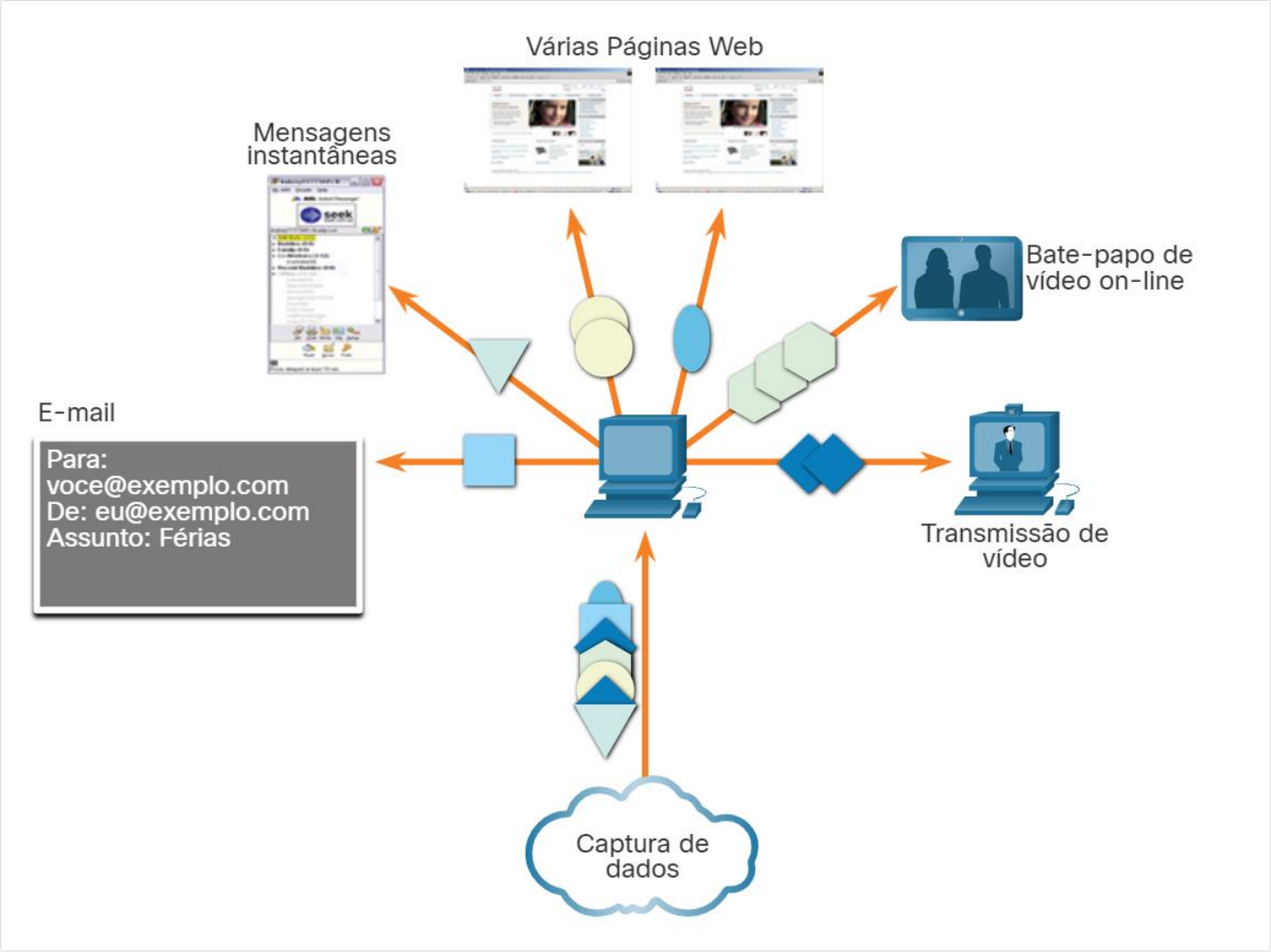


Adicionar Informações de Cabeçalho

O protocolo da camada de transporte também adiciona informações de cabeçalho contendo dados binários organizados em vários campos a cada bloco de dados. São os valores nesses campos que permitem que os vários protocolos da camada de transporte realizem diferentes funções no gerenciamento da comunicação de dados.

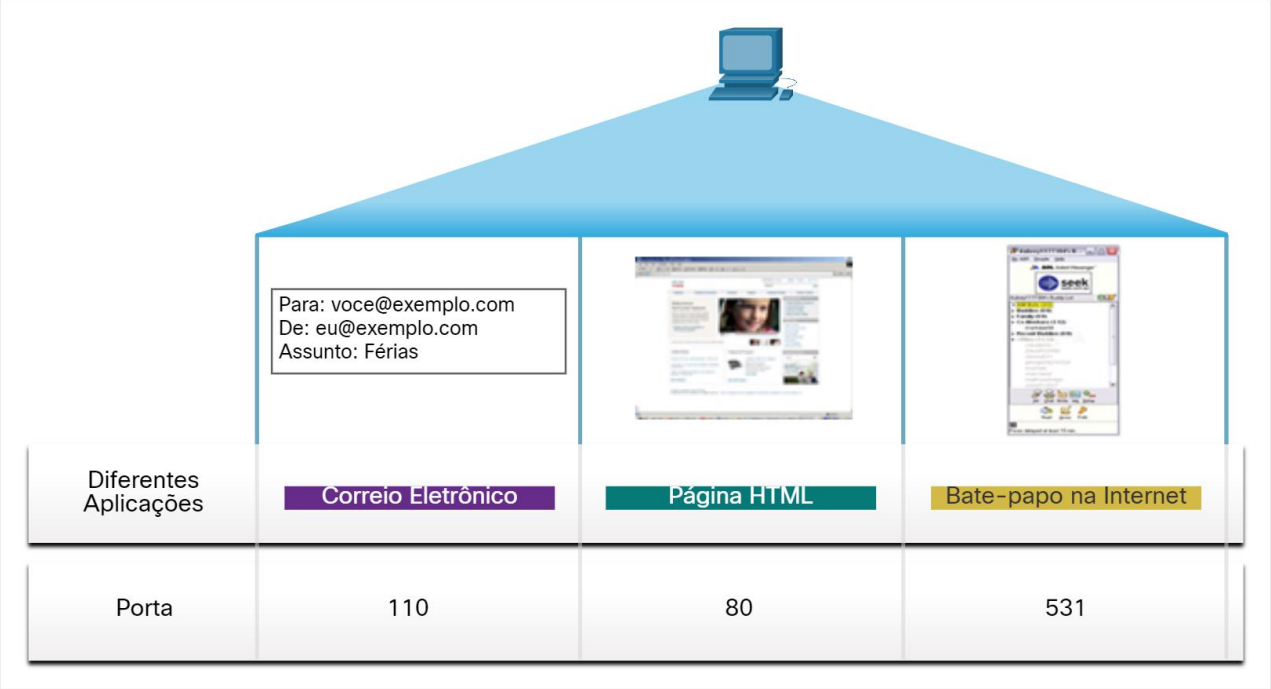
Por exemplo, as informações de cabeçalho são usadas pelo host de recebimento para remontar os blocos de dados em um fluxo de dados completo para o programa de camada de aplicativo de recebimento.

A camada de transporte garante que, mesmo com vários aplicativos em execução em um dispositivo, todos os aplicativos recebam os dados corretos.



Identificação das Aplicações

A camada de transporte deve separar e gerenciar várias comunicações com as diferentes necessidades de requisitos de transporte. Para passar fluxos de dados para os aplicativos adequados, a camada de transporte identifica o aplicativo de destino usando um identificador chamado número da porta. Conforme ilustrado na figura, cada processo de software que precisa acessar a rede recebe um número de porta exclusivo para esse host.

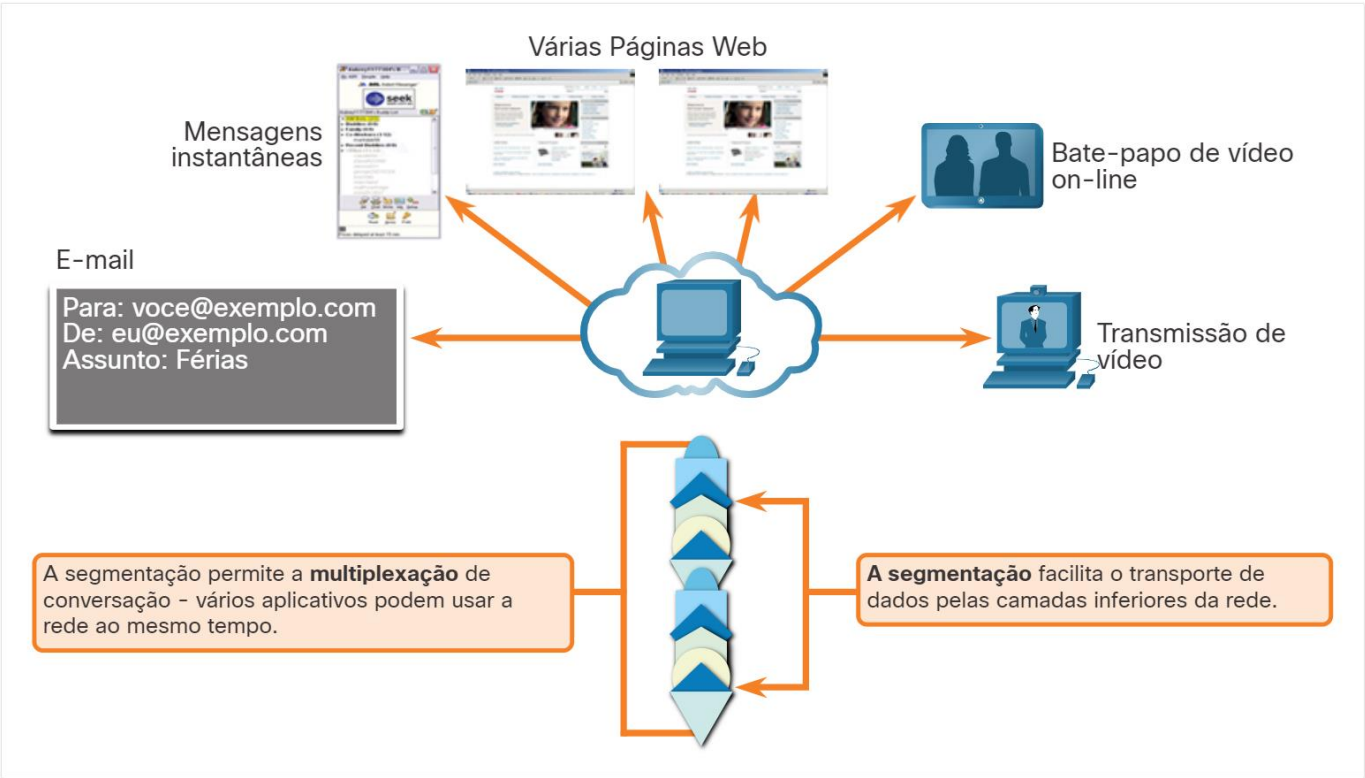


Multiplexação das Conversas

O envio de alguns tipos de dados (por exemplo, um vídeo de streaming) através de uma rede, como um fluxo de comunicação completo, pode consumir toda a largura de banda disponível. Isso impediria que outras conversas de comunicação ocorressem ao mesmo tempo. Isso também dificultaria a recuperação de erro e retransmissão dos dados danificados.

Como mostrado na figura, a camada de transporte usa segmentação e multiplexação para permitir que diferentes conversas de comunicação sejam intercaladas na mesma rede.

A verificação de erros pode ser realizada nos dados do segmento, para determinar se o segmento foi alterado durante a transmissão.

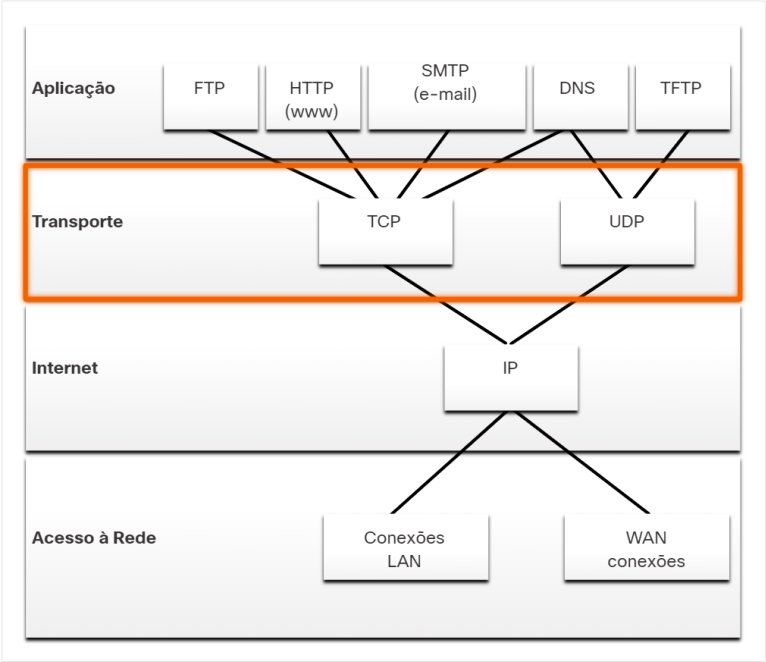


9.1.3 Protocolos da Camada de Transporte

O IP está preocupado apenas com a estrutura, endereçamento e roteamento de pacotes. O IP não especifica como a entrega ou o transporte de pacotes ocorrem.

Os protocolos de camada de transporte especificam como transferir mensagens entre hosts e são responsáveis pelo gerenciamento dos requisitos de confiabilidade de uma conversa. A camada de transporte inclui os protocolos TCP e UDP.

Diferentes aplicações têm diferentes necessidades de confiabilidade de transporte. Portanto, o TCP/IP fornece dois protocolos de camada de transporte, conforme mostrado na figura.



mostra como protocolos de camada de aplicativo como FTP, HTTP, SMTP usam TCP na camada de transporte e DNS e TFTP usam UDP. Como todos eles usam IP na camada da Internet, independentemente de se conectarem a uma LAN ou a uma WAN na camada de acesso à rede

9.1.4 Protocolo TCP

O IP se preocupa apenas com a estrutura, o endereçamento e o roteamento de pacotes, do remetente original ao destino final. A IP não é responsável por garantir a entrega ou determinar se uma conexão entre o remetente e o destinatário precisa ser estabelecida.

O TCP é considerado um protocolo de camada de transporte confiável, completo, que garante que todos os dados cheguem ao destino. O TCP inclui campos que garantem a entrega dos dados do aplicativo. Esses campos exigem processamento adicional pelos hosts de envio e recebimento.

Nota: O TCP divide os dados em segmentos.

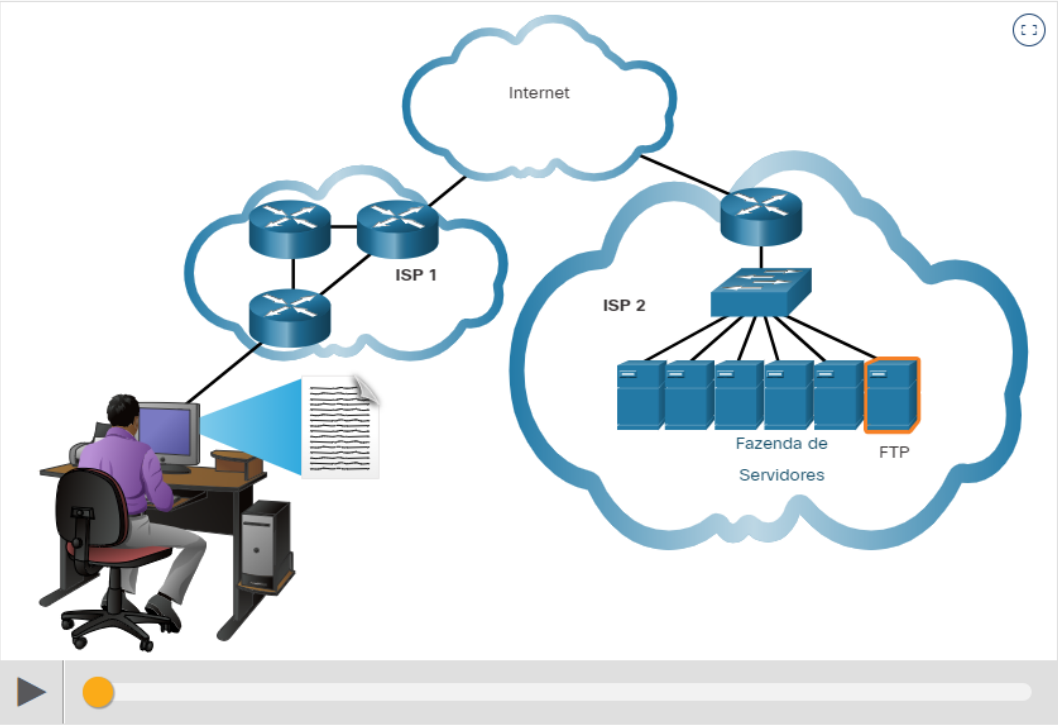
O transporte TCP é análogo a enviar pacotes que são rastreados da origem ao destino. Se um pedido pelo correio estiver dividido em vários pacotes, um cliente poderá verificar on-line a sequência de recebimento do pedido.

O TCP fornece confiabilidade e controle de fluxo usando estas operações básicas:

- Número e rastreamento de segmentos de dados transmitidos para um host específico a partir de um aplicativo específico;
- Confirmar dados recebidos;
- Retransmitir todos os dados não confirmados após um determinado período de tempo
- Dados de sequência que podem chegar em ordem errada
- Enviar dados a uma taxa eficiente que seja aceitável pelo receptor.

Para manter o estado de uma conversa e rastrear as informações, o TCP deve primeiro estabelecer uma conexão entre o remetente e o receptor. É por isso que o TCP é conhecido como um protocolo orientado a conexão.

Clique em Reproduzir na figura para ver como segmentos TCP e as confirmações são transmitidos do remetente ao destinatário.



mostra uma conexão com um servidor FTP iniciado com um handshake de 3 vias TCP e os segmentos de dados contabilizados usando números de sequência e confirmações

Um arquivo é enviado para um servidor usando o protocolo de aplicação FTP (File Transfer Protocol). O TCP rastreia a conversa e divide os dados a serem enviados em 6 segmentos.

Os três primeiros dos seis segmentos são encaminhados para o servidor.

O servidor de arquivos confirma os 3 primeiros segmentos recebidos.

O cliente encaminha os próximos três segmentos.

Nenhum segmento é recebido, nenhuma confirmação é enviada.

O cliente reenvia os 3 últimos segmentos.

Os 3 últimos segmentos são recebidos e confirmados.

9.1.5 Protocolo UDP (User Datagram Protocol)

O UDP é um protocolo de camada de transporte mais simples do que o TCP. Ele não fornece confiabilidade e controle de fluxo, o que significa que requer menos campos de cabeçalho. Como o remetente e os processos UDP receptor não precisam gerenciar confiabilidade e controle de fluxo, isso significa que datagramas UDP podem ser processados mais rápido do que segmentos TCP. O UDP fornece as funções básicas para fornecer datagramas entre os aplicativos apropriados, com muito pouca sobrecarga e verificação de dados.

Nota: O UDP divide os dados em datagramas que também são chamados de segmentos.

UDP é um protocolo sem conexão. Como o UDP não fornece confiabilidade ou controle de fluxo, ele não requer uma conexão estabelecida. Como o UDP não controla informações enviadas ou recebidas entre o cliente e o servidor, o UDP também é conhecido como um protocolo sem estado.

UDP também é conhecido como um protocolo de entrega de melhor esforço porque não há confirmação de que os dados são recebidos no destino. Com o UDP, não há processo de camada de transporte que informe ao remetente se a entrega foi bem-sucedida.

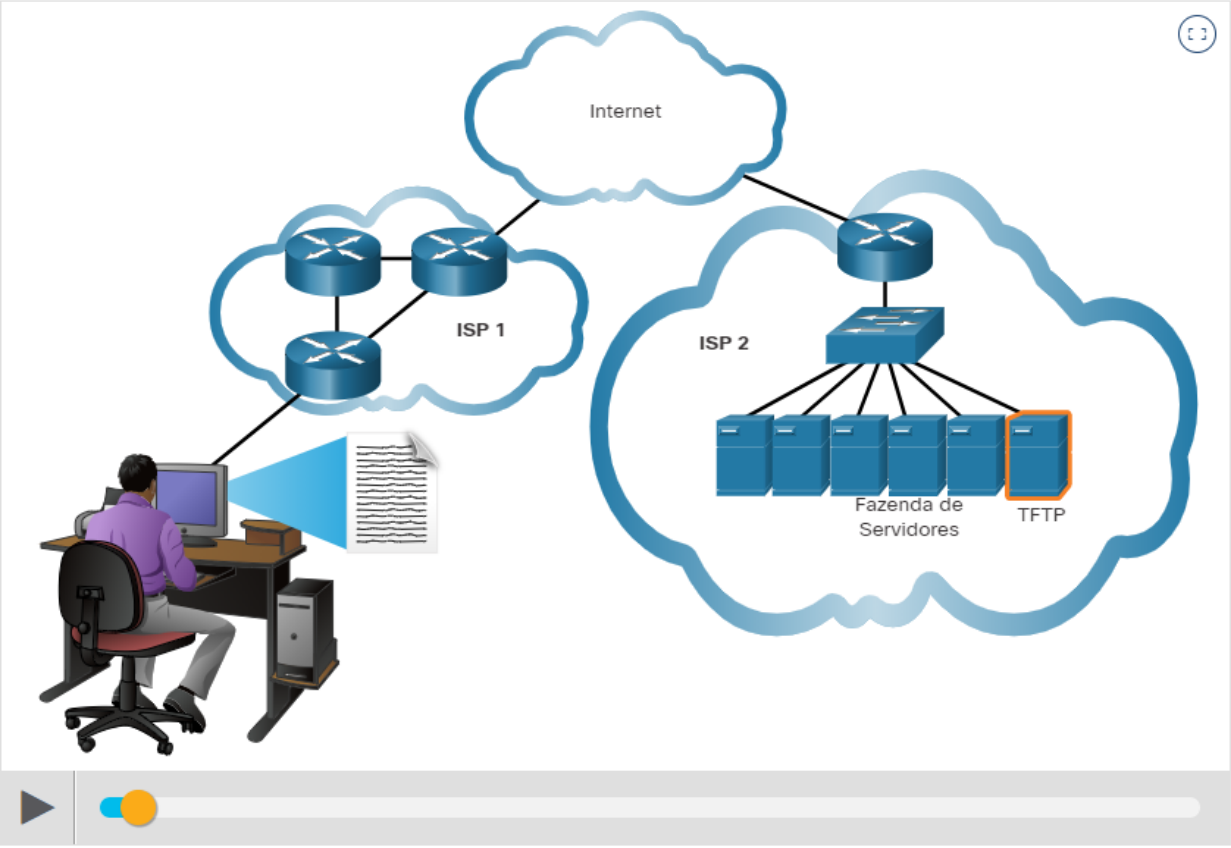
O UDP é como colocar uma carta regular, não registrada, no correio. O remetente da carta não tem conhecimento se o destinatário está disponível para receber a carta. Nem a agência de correio é responsável por rastrear a carta ou informar ao remetente se ela não chegar ao destino final.

Clique em Reproduzir na figura para ver uma animação dos datagramas UDP sendo transmitidos do remetente para o receptor.

mostra uma conexão com um servidor TFTP usando datagramas UDP que são enviados sem números de sequência ou confirmaçõesSS

Um arquivo é enviado a um servidor usando o protocolo de aplicação TFTP. O UDP divide os dados em datagramas e os envia usando a entrega do melhor esforço.

O servidor de arquivos recebe todas os 6 segmentos, nenhuma confirmação é enviada.



9.1.6 O protocolo de Camada de Transporte Certo para a Aplicação Certa

Alguns aplicativos podem tolerar a perda de dados durante a transmissão pela rede, mas atrasos na transmissão são inaceitáveis. Para esses aplicativos, o UDP é a melhor escolha, pois requer menos sobrecarga da rede. O UDP é preferível para aplicativos como Voz sobre IP (VoIP). Agradecimentos e retransmissão atrasariam a entrega e tornariam a conversa por voz inaceitável.

O UDP também é usado por aplicativos de solicitação e resposta onde os dados são mínimos, e a retransmissão pode ser feita rapidamente. Por exemplo, o serviço de nome de domínio (DNS) usa UDP para esse tipo de transação. O cliente solicita endereços IPv4 e IPv6 para um nome de domínio conhecido de um servidor DNS. Se o cliente não receber uma resposta em um período predeterminado de tempo, ele simplesmente envia a solicitação novamente.

Por exemplo, se um ou dois segmentos de uma transmissão de vídeo ao vivo não conseguir chegar, isso criará apenas uma interrupção momentânea na transmissão. Isso pode aparecer como uma distorção na imagem ou no som, mas pode não ser notado pelo usuário. Se o dispositivo de destino considerasse os dados perdidos, a transmissão poderia

atrasar, enquanto aguardasse as retransmissões, causando, portanto, grandes perdas de áudio e vídeo. Nesse caso, é melhor fornecer a melhor experiência de mídia com os segmentos recebidos e descartar a confiabilidade.

Para outras aplicações, é importante que todos os dados cheguem e que possam ser processados em sua sequência adequada. Para esses tipos de aplicativos, o TCP é usado como o protocolo de transporte. Por exemplo, aplicações como bancos de dados, navegadores e clientes de e-mail exigem que todos os dados enviados cheguem ao destino em seu estado original. Quaisquer dados ausentes podem corromper uma comunicação, tornando-a incompleta ou ilegível. Por exemplo, é importante ao acessar informações bancárias pela web certificar-se de que todas as informações são enviadas e recebidas corretamente.

Os desenvolvedores de aplicações devem escolher que tipo de protocolo de transporte é apropriado com base nas necessidades de suas Saplicações. O vídeo pode ser enviado através de TCP ou UDP. Os aplicativos que transmitem áudio e vídeo armazenados normalmente usam TCP. O aplicativo usa TCP para executar buffer, sondagem de largura de banda e controle de congestionamento, a fim de controlar melhor a experiência do usuário.

Vídeo e voz em tempo real geralmente usam UDP, mas também podem usar TCP, ou UDP e TCP. Um aplicativo de videoconferência pode usar UDP por padrão, mas como muitos firewalls bloqueiam UDP, o aplicativo também pode ser enviado por TCP.

Os aplicativos que transmitem áudio e vídeo armazenados usam TCP. Por exemplo, se sua rede, repentinamente, não comportar a largura de banda necessária para a transmissão de um filme sob demanda, a aplicação interrompe a reprodução. Durante essa interrupção, você deverá ver uma mensagem de “buffering...” , enquanto o TCP age para restabelecer a transmissão. Quando todos os segmentos estão em ordem e um nível mínimo de largura de banda é restaurado, a sessão TCP é retomada e o filme retoma a reprodução.

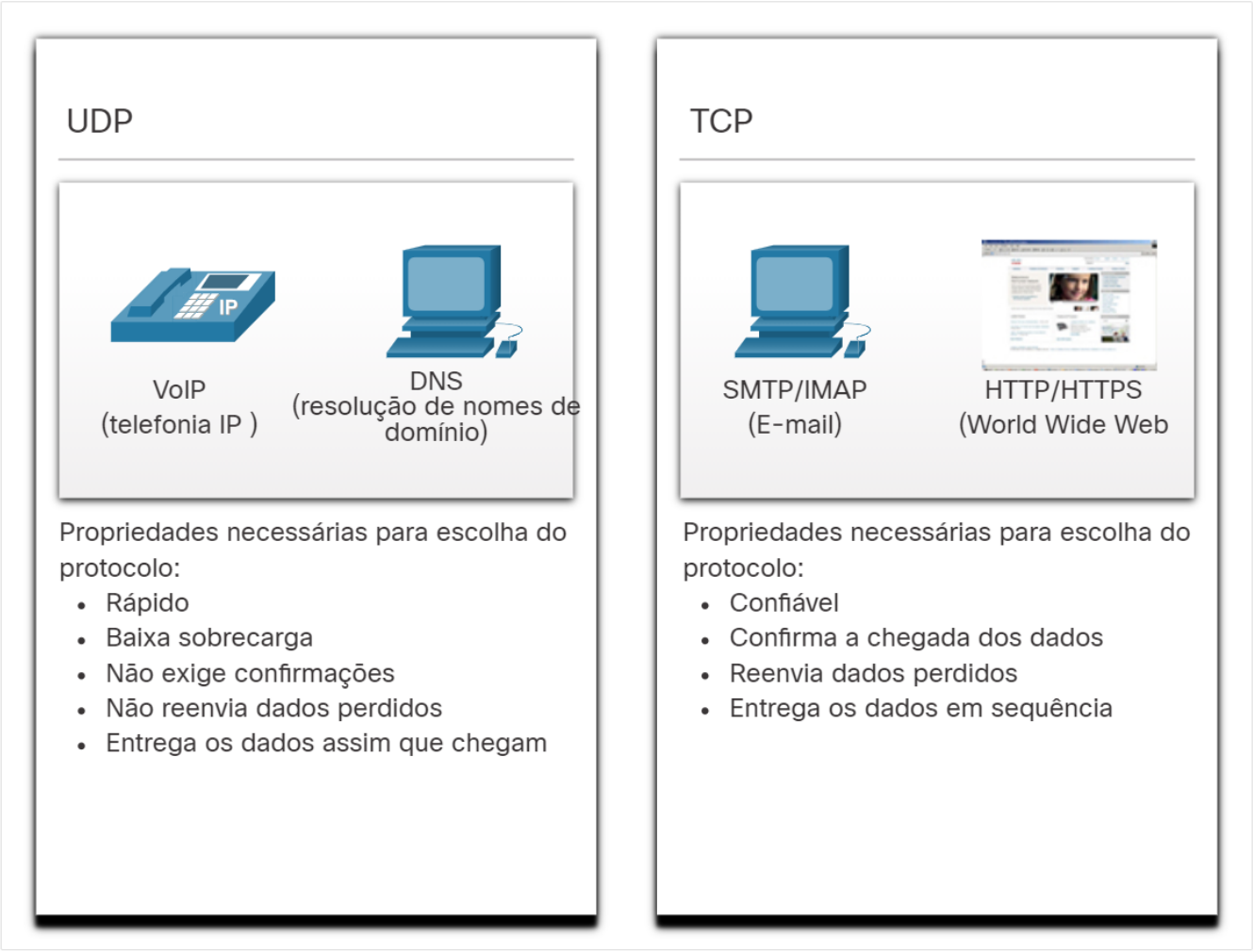
A figura resume as diferenças entre UDP e TCP.

lista as diferenças entre UDP: rápida, baixa sobrecarga, sem confirmações, sem reenvio e TCP: confiável, reconhece dados, reenvia dados perdidos e entrega dados com números de sequência

- Rápido
- Baixa sobrecarga
- Não exige confirmações
- Não reenvia dados perdidos
- Entrega os dados assim que chegam

Propriedades necessárias para escolha do protocolo:

- Confiável
- Confirma a chegada dos dados
- Reenvia dados perdidos
- Entrega os dados em sequência



9.1.7 Verifique o seu entendimento - Transporte de dados

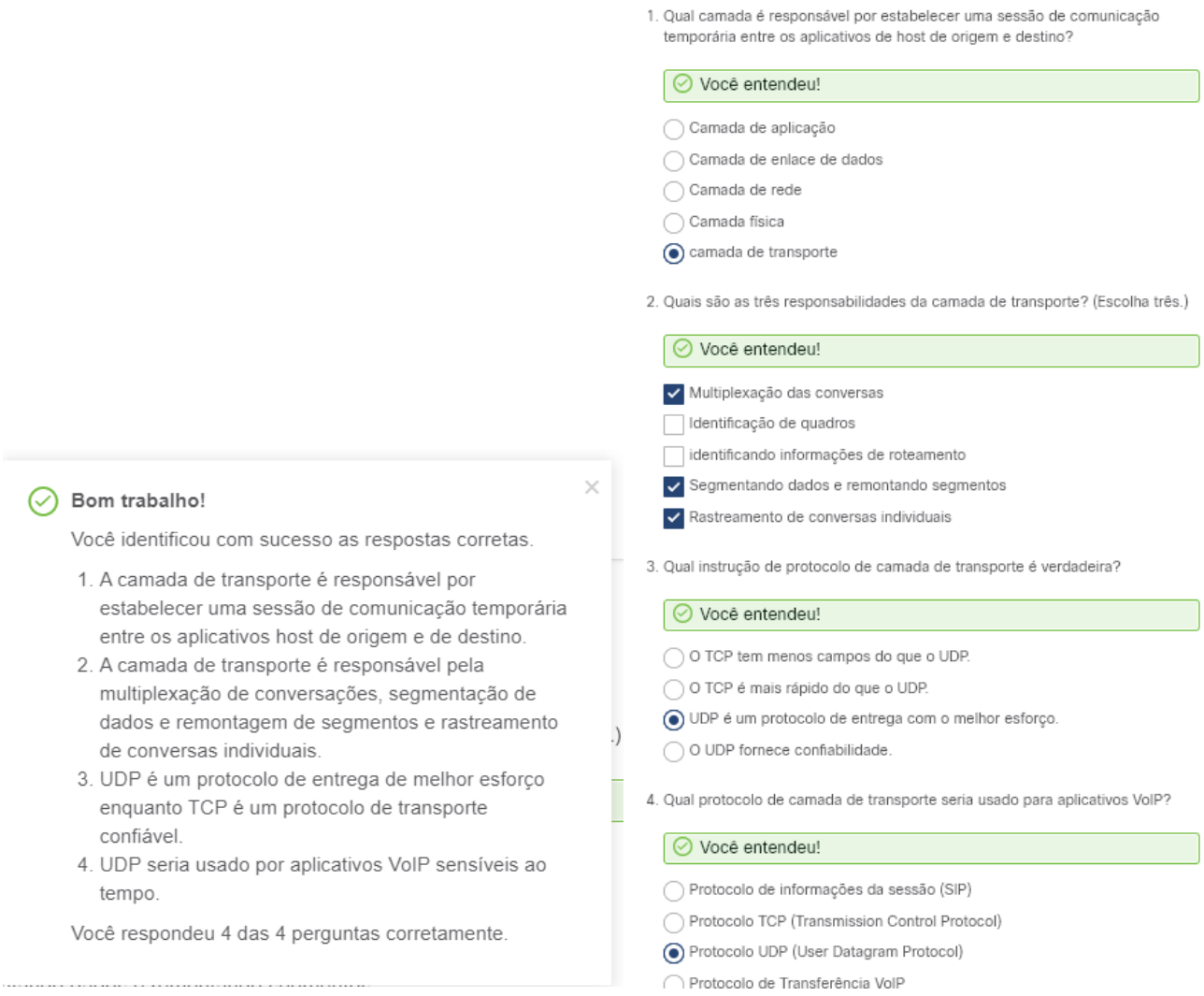
Verifique sua compreensão da camada de transporte escolhendo a MELHOR resposta para as seguintes perguntas.

1. Qual camada é responsável por estabelecer uma sessão de comunicação temporária entre os aplicativos de host de origem e destino?
   - ○ Camada de aplicação
   - ○ Camada de enlace de dados
   - ○ Camada de rede
   - ○ Camada física
   - ○ camada de transporte

2. Quais são as três responsabilidades da camada de transporte? (Escolha três.)
   - ☐ Multiplexação das conversas
   - ☐ Identificação de quadros
   - ☐ identificando informações de roteamento
   - ☐ Segmentando dados e remontando segmentos
   - ☐ Rastreamento de conversas individuais

3. Qual instrução de protocolo de camada de transporte é verdadeira?
   - ○ O TCP tem menos campos do que o UDP.
   - ○ O TCP é mais rápido do que o UDP.
   - ○ UDP é um protocolo de entrega com o melhor esforço.
   - ○ O UDP fornece confiabilidade.

4. Qual protocolo de camada de transporte seria usado para aplicativos VoIP?
   - Protocolo de informações da sessão (SIP)
   - Protocolo TCP
   - Protocolo UDP (User Datagram Protocol)
   - Protocolo de Transferência VoIP



9.2 Visão geral do TCP

9.2.1 Recursos TCP

No tópico anterior, você aprendeu que TCP e UDP são os dois protocolos de camada de transporte. Este tópico fornece mais detalhes sobre o que o TCP faz e quando é uma boa idéia usá-lo em vez de UDP.

Para entender as diferenças entre TCP e UDP, é importante entender como cada protocolo implementa recursos específicos de confiabilidade e como cada protocolo rastreia conversas.

Além de suportar as funções básicas de segmentação e remontagem de dados, o TCP também fornece os seguintes serviços:

- **Estabelece uma sessão** - O TCP é um protocolo orientado à conexão que negocia e estabelece uma conexão (ou sessão) permanente entre os dispositivos de origem e de destino antes de encaminhar qualquer tráfego. Com o estabelecimento da sessão, os dispositivos negociam o volume de tráfego esperado que pode ser encaminhado em determinado momento e os dados de comunicação entre os dois podem ser gerenciados atentamente.

- **Garante a entrega confiável** - Por várias razões, é possível que um segmento seja corrompido ou perdido completamente, pois é transmitido pela rede. O TCP garante que cada segmento enviado pela fonte chegue ao destino.
- **Fornece entrega no mesmo pedido** - Como as redes podem fornecer várias rotas que podem ter taxas de transmissão diferentes, os dados podem chegar na ordem errada. Ao numerar e sequenciar os segmentos, o TCP garante que os segmentos sejam remontados na ordem correta.
- **Suporta controle de fluxo** - os hosts de rede têm recursos limitados (ou seja, memória e poder de processamento). Quando percebe que esses recursos estão sobrecarregados, o TCP pode requisitar que a aplicação emissora reduza a taxa de fluxo de dados. Para isso, o TCP regula o volume de dados transmitido pelo dispositivo origem. O controle de fluxo pode impedir a necessidade de retransmissão dos dados quando os recursos do host receptor estão sobrecarregados.

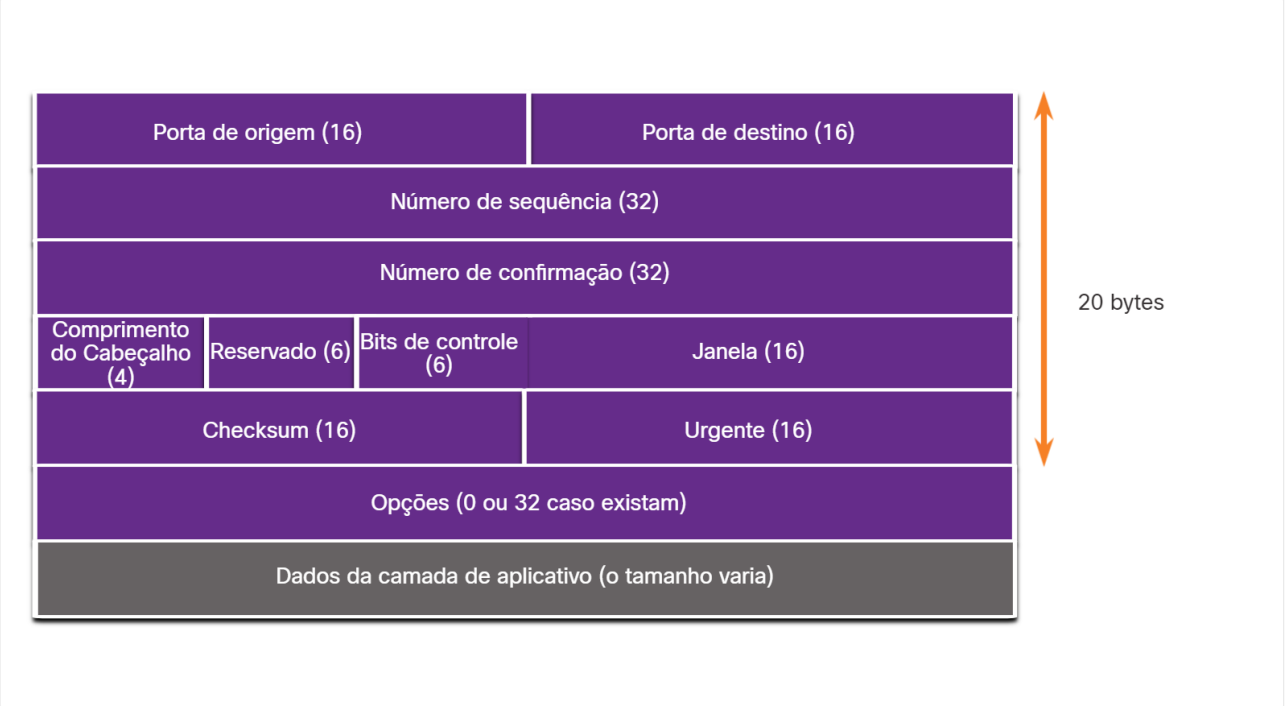
Para obter mais informações sobre o TCP, procure o RFC 793 na Internet.

9.2.2 Cabeçalho TCP

TCP é um protocolo stateful, o que significa que ele controla o estado da sessão de comunicação. Para manter o controle do estado de uma sessão, o TCP registra quais informações ele enviou e quais informações foram confirmadas. A sessão com estado começa com o estabelecimento da sessão e termina com o encerramento da sessão.

Um segmento TCP adiciona 20 bytes (ou seja, 160 bits) de sobrecarga ao encapsular os dados da camada de aplicativo. A figura mostra os campos em um cabeçalho TCP.

mostra os campos no cabeçalho TCP 20 bytesPorta de origem (16)Porta de destino (16)Número de confirmação (32)Comprimento do Cabeçalho (4)Janela (16)Checksum (16)Urgente (16)Opções (0 ou 32 caso existam)Dados da camada de aplicativo (o tamanho varia)Número de sequência (32)Reservado (6)Bits de controle (6)



9.2.3 Campos de cabeçalho TCP

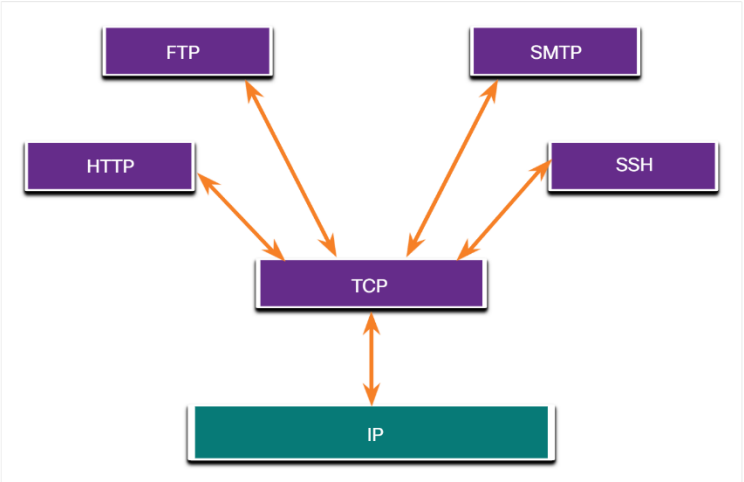
A tabela identifica e descreve os dez campos em um cabeçalho TCP.

Descrição do campo do cabeçalho TCP Porta de origem Um campo de 16 bits usado para identificar o aplicativo de origem por número de porta.Destination Porta 16 bits campo usado para identificar o aplicativo de destino pelo número da porta.Número de seqüência A 32 bits campo usado para fins de remontagem de dados.Número de confirmação Um campo de 32 bits usado para indicar que os dados foram recebidos e o próximo byte esperado de o Source.Header Comprimento Um campo de 4 bits conhecido como 'offset' de datas' que indica o comprimento do cabeçalho do segmento TCP.Reservado um campo de 6 bits que está reservado para uso futuro.Bits de controle Um campo de 6 bits usado que inclui códigos de bits ou sinalizadores, que indicam a finalidade e a função do segmento TCP.Window tamanho Um campo de 16 bits usado para indicar o número de bytes que podem ser aceito ao mesmo tempo.Checksum Um campo de 16 bits usado para verificação de erros do cabeçalho de segmento e data.Urgente Um campo de 16 bits usado para indicar se o dados contidos são urgentes.

Campo de cabeçalho TCP	Descrição
Porta de origem	Um campo de 16 bits usado para identificar o aplicativo de origem por número de porta.
Porta de destino	Um campo de 16 bits usado para identificar o aplicativo de destino por porta número.
Número sequencial	Um campo de 32 bits usado para fins de remontagem de dados.
Número de Confirmação	Um campo de 32 bits usado para indicar que os dados foram recebidos e o próximo byte esperado da fonte.
Comprimento do cabeçalho	Um campo de 4 bits conhecido como 'offset' de datas' que indica o comprimento do cabeçalho do segmento TCP.
Reservado	Um campo de 6 bits que é reservado para uso futuro.
Bits de controle	Um campo de 6 bits que inclui códigos de bits, ou sinalizadores, que indicam a finalidade e função do segmento TCP.
Tamanho da janela	Um campo de 16 bits usado para indicar o número de bytes que podem ser aceitos de uma só vez.
Checksum	Um campo de 16 bits usado para verificação de erros do cabeçalho e dos dados do segmento.
Urgente	Um campo de 16 bits usado para indicar se os dados contidos são urgentes.

9.2.4 Aplicações que usam TCP

O TCP é um bom exemplo de como as diferentes camadas do conjunto de protocolos TCP / IP têm funções específicas. O TCP lida com todas as tarefas associadas à divisão do fluxo de dados em segmentos, fornecendo confiabilidade, controlando o fluxo de dados e reordenando segmentos. O TCP libera a aplicação da obrigação de gerenciar todas essas tarefas. Aplicações como as mostradas na figura, podem simplesmente enviar o fluxo de dados à camada de transporte e usar os serviços TCP.

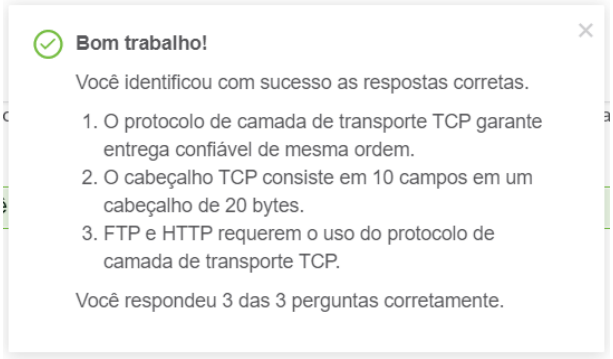


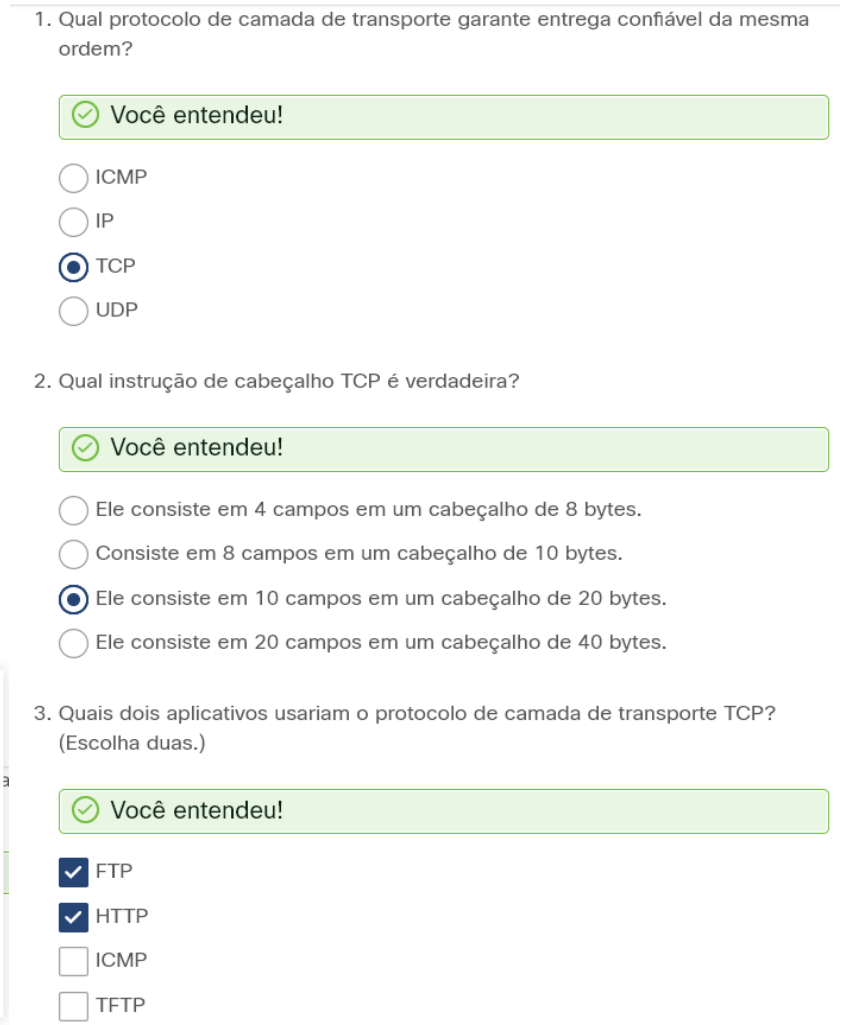
mostra as setas apontando ambas as direções de HTTP, FTP, SMTP e SSH para TCP e, em seguida, de TCP para IP

9.2.5 Verifique sua compreensão - Visão geral do TCP

Verifique sua compreensão do TCP, escolhendo a melhor resposta para as seguintes perguntas.

1. Qual protocolo de camada de transporte garante entrega confiável da mesma ordem?
   - ICMP
   - IP
   - TCP
   - UDP
2. Qual instrução de cabeçalho TCP é verdadeira?
   - Ele consiste em 4 campos em um cabeçalho de 8 bytes.
   - Consiste em 8 campos em um cabeçalho de 10 bytes.
   - Ele consiste em 10 campos em um cabeçalho de 20 bytes.
   - Ele consiste em 20 campos em um cabeçalho de 40 bytes.
3. Quais dois aplicativos usariam o protocolo de camada de transporte TCP? (Escolha duas.)
   - ☐ FTP
   - ☐ HTTP
   - ☐ ICMP
   - ☐ TFTP
   - ☐ VoIP





9.3 Visão Geral do UDP

9.3.1 Recursos UDP

Este tópico abordará o UDP, o que ele faz e quando é uma boa idéia usá-lo em vez de TCP. UDP é um protocolo de transporte de melhor esforço. O UDP é um protocolo de transporte leve que oferece a mesma segmentação de dados e remontagem que o TCP, mas sem a confiabilidade e o controle de fluxo do TCP.

O UDP é um protocolo simples, normalmente descrito nos termos do que ele não faz em comparação ao TCP.

Os recursos UDP incluem o seguinte:

- Os dados são reagrupados na ordem em que são recebidos.
- Quaisquer segmentos perdidos não são reenviados.
- Não há estabelecimento de sessão.
- O envio não é informado sobre a disponibilidade do recurso.

Para obter mais informações sobre o UDP, pesquise na Internet o RFC.

9.3.2 Cabeçalho UDP

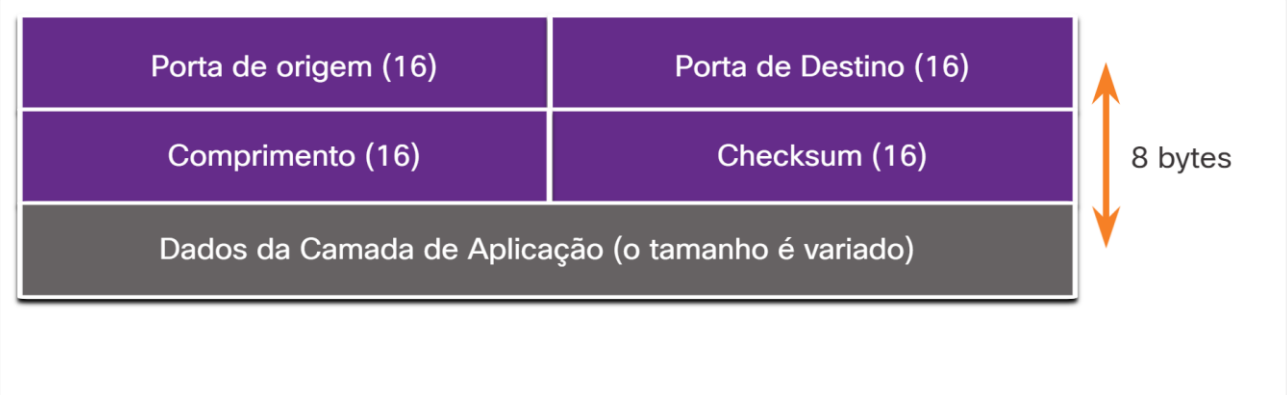
UDP é um protocolo sem estado, o que significa que nem o cliente nem o servidor rastreiam o estado da sessão de comunicação. Se a confiabilidade for necessária ao usar o UDP como protocolo de transporte, ela deve ser tratada pela aplicação.

Um dos requisitos mais importantes para transmitir vídeo ao vivo e voz sobre a rede é que os dados continuem fluindo rapidamente. Vídeo ao vivo e aplicações de voz podem tolerar alguma perda de dados com efeito mínimo ou sem visibilidade e são perfeitos para o UDP.

Os blocos de comunicação no UDP são chamados de datagramas ou segmentos. Esses datagramas são enviados como o melhor esforço pelo protocolo da camada de transporte.

O cabeçalho UDP é muito mais simples do que o cabeçalho TCP porque só tem quatro campos e requer 8 bytes (ou seja, 64 bits). A figura mostra os campos em um cabeçalho UDP.

O diagrama de datagrama UDP mostra 4 campos de cabeçalho: porta de origem, porta de destino, comprimento e soma de verificação, bem como os dados da camada de aplicação não cabeçalho



9.3.3

9.3.3 Campos de Cabeçalho UDP

A tabela identifica e descreve os quatro campos em um cabeçalho UDP.

Descrição do campo do cabeçalho UDP Porta de origem Um campo de 16 bits usado para identificar o aplicativo de origem por número de porta.Destination Porta 16 bits campo usado para identificar o aplicativo de destino pelo número da porta.Comprimento Um campo de 16 bits que indica o comprimento do cabeçalho de datagrama UDP. Checksum Um campo de 16 bits usado para verificação de erros do cabeçalho e dos dados do datagrama.

Campo de Cabeçalho UDP	Descrição
Porta de origem	Um campo de 16 bits usado para identificar o aplicativo de origem por número de porta.
Porta de destino	Um campo de 16 bits usado para identificar o aplicativo de destino por porta número.
Tamanho	Um campo de 16 bits que indica o comprimento do cabeçalho do datagrama UDP.
Checksum	Um campo de 16 bits usado para verificação de erros do cabeçalho e dos dados do datagrama.

9.3.4 Aplicações que usam UDP

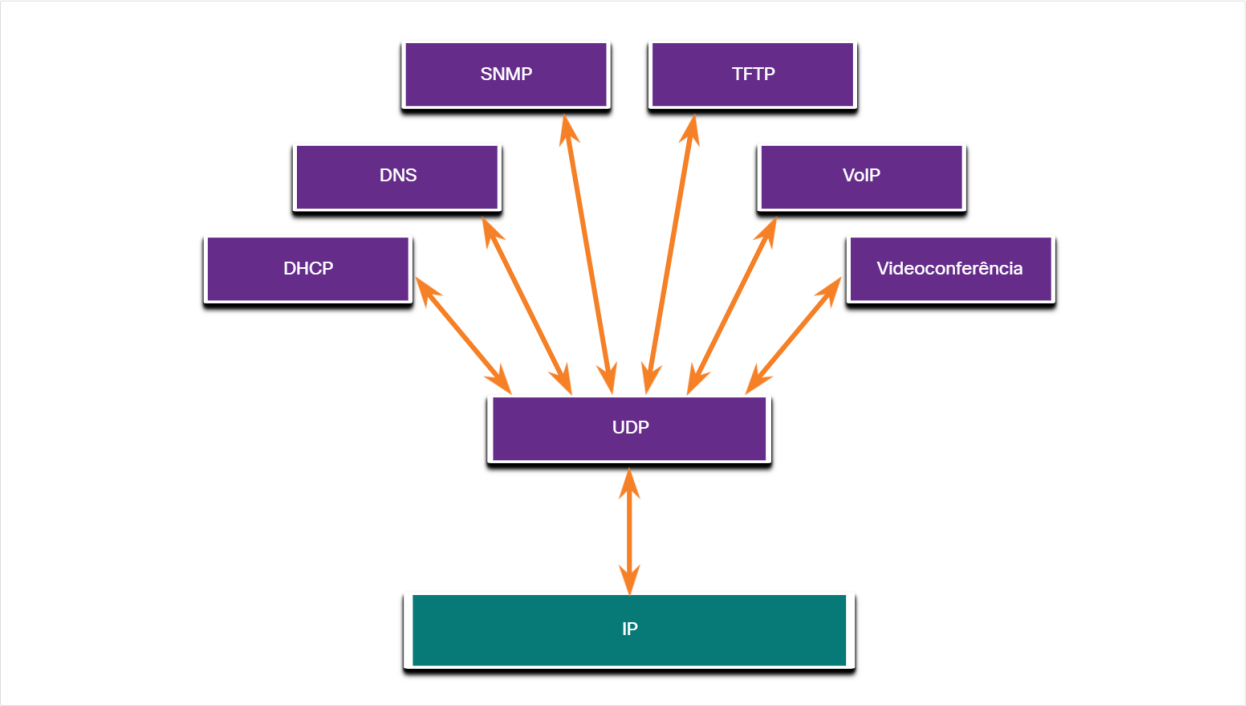
Há três tipos de aplicações que são mais adequadas para o UDP:

- **Aplicativos de vídeo e multimídia ao vivo** - Esses aplicativos podem tolerar a perda de dados, mas requerem pouco ou nenhum atraso. Os exemplos incluem VoIP e transmissão de vídeo ao vivo.
- **Solicitações simples e aplicativos de resposta** - aplicativos com transações simples em que um host envia uma solicitação e pode ou não receber uma resposta. Os exemplos incluem DNS e DHCP.

- **Aplicativos que lidam com a confiabilidade** - Comunicações unidirecionais em que o controle de fluxo, a detecção de erros, as confirmações e a recuperação de erros não são necessários ou podem ser gerenciados pelo aplicativo. Os exemplos incluem SNMP e TFTP.

A figura identifica aplicativos que exigem UDP.

mostra as setas apontando ambas as direções de DHCP, DNS, SNMP, TFTP, VoIP e IPTV para UDP e, em seguida, de UDP para IP



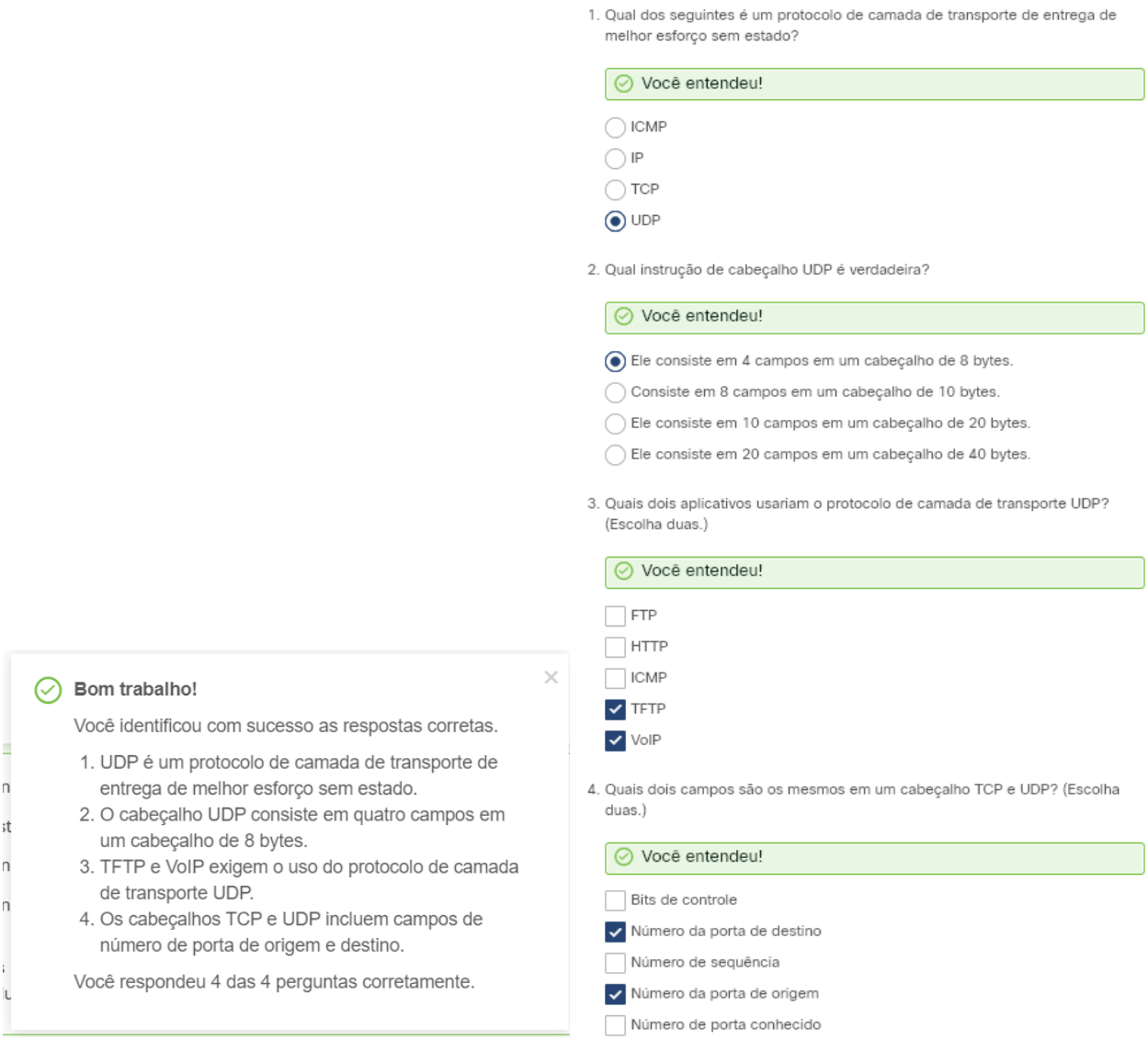
Embora por padrão DNS e SNMP usem UDP, ambos podem usar TCP. O DNS usará o TCP se a solicitação ou resposta de DNS for maior que 512 bytes, como quando uma resposta de DNS inclui muitas resoluções de nome. Da mesma forma, em algumas situações o administrador de redes pode querer configurar o SNMP para usar o TCP.

9.3.5 Verifique sua compreensão - Visão geral do UDP

Verifique sua compreensão do UDP escolhendo a melhor resposta para as seguintes perguntas.

1. Qual dos seguintes é um protocolo de camada de transporte de entrega de melhor esforço sem estado?
   - ICMP
   - IP
   - TCP
   - UDP
2. Qual instrução de cabeçalho UDP é verdadeira?
   - Ele consiste em 4 campos em um cabeçalho de 8 bytes.
   - Consiste em 8 campos em um cabeçalho de 10 bytes.
   - Ele consiste em 10 campos em um cabeçalho de 20 bytes.
   - Ele consiste em 20 campos em um cabeçalho de 40 bytes.
3. Quais dois aplicativos usariam o protocolo de camada de transporte UDP? (Escolha duas.)
   - ☐ FTP
   - ☐ HTTP
   - ☐ ICMP
   - ☐ TFTP
   - ☐ VoIP
4. Quais dois campos são os mesmos em um cabeçalho TCP e UDP? (Escolha duas.)

- ☐ Bits de controle
- ☐ Número da porta de destino
- ☐ Número de sequência
- ☐ Número da porta de origem
- ☐ Número de porta conhecido



9.4 Números de porta

9.4.1 Várias comunicações separadas

Como você aprendeu, existem algumas situações em que o TCP é o protocolo certo para o trabalho e outras situações em que o UDP deve ser usado. Independentemente do tipo de dados que estão sendo transportados, tanto o TCP quanto o UDP usam números de porta.

Os protocolos de camada de transporte TCP e UDP usam números de porta para gerenciar várias conversas simultâneas. Conforme mostrado na figura, os campos de cabeçalho TCP e UDP identificam um número de porta do aplicativo de origem e destino.

mostra a porta de origem e os campos de cabeçalho da porta de destino que são 2 bytes cada Porta de origem (16)Porta de destino (16)

O número da porta de origem está associado ao aplicativo de origem no host local, enquanto o número da porta de destino está associado ao aplicativo de destino no host remoto.

Por exemplo, suponha que um host está iniciando uma solicitação de página da Web a partir de um servidor Web. Quando o host inicia a solicitação de página da Web, o número da porta de origem é gerado dinamicamente pelo host para identificar exclusivamente a conversa. Cada solicitação gerada por um host usará um número de porta de origem criado dinamicamente diferente. Este processo permite que várias conversações ocorram simultaneamente.

Na solicitação, o número da porta de destino é o que identifica o tipo de serviço que está sendo solicitado do servidor Web de destino. Por exemplo, quando um cliente especifica a porta 80 na porta de destino, o servidor que receber a mensagem sabe que os serviços Web são solicitados.

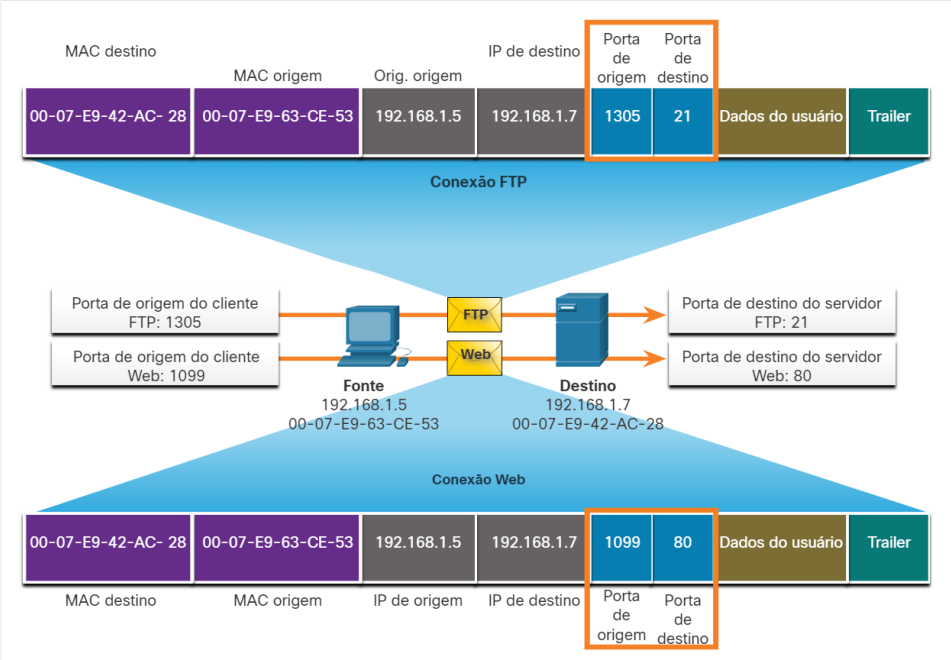
Um servidor pode oferecer mais de um serviço simultaneamente, como serviços web na porta 80, enquanto oferece o estabelecimento de conexão FTP (File Transfer Protocol) na porta 21.

9.4.2 Pares de Sockets

As portas origem e destino são colocadas no segmento. Os segmentos são encapsulados em um pacote IP. O pacote IP contém o endereço IP de origem e destino. A combinação do endereço IP de origem e o número de porta de origem, ou do endereço IP de destino e o número de porta de destino é conhecida como um socket.

No exemplo na figura, o PC está solicitando simultaneamente serviços FTP e Web do servidor de destino.

mostra um PC fazendo uma solicitação da Web e uma solicitação FTP para um servidor. As solicitações têm números de porta de origem e de destino que identificam o PC host e o serviço de aplicativo solicitado, respectivamente



No exemplo, a solicitação FTP gerada pelo PC inclui os endereços MAC da Camada 2 e os endereços IP da Camada 3. A solicitação também identifica o número da porta de origem 1305 (ou seja, gerado dinamicamente pelo host) e a porta de destino, identificando os serviços de FTP na porta 21. O host também solicitou uma página da Web

do servidor usando os mesmos endereços de Camada 2 e Camada 3. No entanto, ele está usando o número da porta de origem 1099 (ou seja, gerado dinamicamente pelo host) e a porta de destino identificando o serviço Web na porta 80.

O socket é usado para identificar o servidor e o serviço que está sendo solicitado pelo cliente. Um socket do cliente pode ser assim, com 1099 representando o número da porta de origem: 192.168.1.5:1099

O soquete em um servidor da web pode ser 192.168.1.7:80

Juntos, esses dois soquetes se combinam para formar um *par de soquetes*: 192.168.1.5:1099, 192.168.1.7:80

Os sockets permitem que vários processos em execução em um cliente se diferenciem uns dos outros, e várias conexões com um processo no servidor sejam diferentes umas das outras.

Este número de porta age como um endereço de retorno para a aplicação que faz a solicitação. A camada de transporte rastreia essa porta e a aplicação que iniciou a solicitação, de modo que quando uma resposta é retornada, ela pode ser encaminhada para a aplicação correta.

9.4.3 Grupos de Números de Porta

A Internet Assigned Numbers Authority (IANA) é a organização de padrões responsável por atribuir vários padrões de endereçamento, incluindo os números de porta de 16 bits. Os 16 bits usados para identificar os números de porta de origem e destino fornecem um intervalo de portas de 0 a 65535.

A IANA dividiu a gama de números nos três grupos de portos seguintes.

bem conhecidas 0 a 1.023Estas portas são reservados para serviços e aplicativos comuns ou populares, como navegadores da web, clientes de e-mail e clientes de acesso remoto. Definido bem conhecido para aplicativos comuns de servidor permite que os clientes identifiquem facilmente o serviço associado necessário.Portas registradas1,024 a 49.151Estas portas números são atribuídos pela IANA a uma entidade solicitante para usar com processos ou aplicações. Esses processos são principalmente individuais aplicativos que um usuário optou por instalar, em vez de comuns que receberiam um número de porta bem conhecido. Por exemplo, Cisco registrou a porta 1812 para a autenticação do servidor RADIUS Portas dinâmicas e / ou dinâmicas 49.152 a 65.535.Essas portas também são conhecido como portas efêmeras. O sistema operacional do cliente geralmente atribui números de porta dinamicamente quando uma conexão a um serviço é iniciada. A porta dinâmica é então usado para identificar o aplicativo cliente durante a comunicação.

Grupo de Portas	Intervalo de números	Descrição
Portas bem conhecidas	**0 a 1.023**	• Estes números de porta são reservados para serviços comuns ou populares e aplicativos como navegadores da web, clientes de e-mail e acesso remoto clientes. • Portas bem conhecidas definidas para aplicativos comuns de servidor permite para identificar facilmente o serviço associado necessário.
Portas registradas	**1.024 a 49.151**	• Esses números de porta são atribuídos pela IANA a uma entidade solicitante para usar com processos ou aplicativos específicos. • Esses processos são principalmente aplicativos individuais que um usuário optou por instalar, em vez de aplicativos comuns que receber um número de porta bem conhecido. • Por exemplo, a Cisco registrou a porta 1812 para seu servidor RADIUS processo de autenticação.

bem conhecidas 0 a 1.023Estas portas são reservados para serviços e aplicativos comuns ou populares, como navegadores da web, clientes de e-mail e clientes de acesso remoto. Definido bem conhecido para aplicativos comuns de servidor permite que os clientes identifiquem facilmente o serviço associado necessário.Portas registradas1,024 a 49.151Estas portas números são atribuídos pela IANA a uma entidade solicitante para usar com processos ou aplicações. Esses processos são principalmente individuais aplicativos que um usuário optou por instalar, em vez de comuns que receberiam um número de porta bem conhecido. Por exemplo, Cisco registrou a porta 1812 para a autenticação do servidor RADIUS Portas dinâmicas e / ou dinâmicas 49.152 a 65.535.Essas portas também são conhecido como portas efêmeras. O sistema operacional do cliente geralmente atribui números de porta dinamicamente quando uma conexão a um serviço é iniciada. A porta dinâmica é então usado para identificar o aplicativo cliente durante a comunicação.

Grupo de Portas	Intervalo de números	Descrição
Particular e/ou **portas dinâmicas**	**49.152 a 65.535**	• Essas portas também são conhecidas como portas *efêmeras*. • O sistema operacional do cliente geralmente atribui números de porta dinamicamente quando uma conexão a um serviço é iniciada. • A porta dinâmica é então usada para identificar o aplicativo cliente durante a comunicação.

Observação: Alguns sistemas operacionais clientes podem usar números de porta registrados em vez de números de porta dinâmicos para atribuir portas de origem.

A tabela exibe alguns números de porta conhecidos comuns e seus aplicativos associados.

Números de Portas Bem Conhecidas

PortaNumberProtocolApplication20TCPFile Transfer Protocol (FTP) - Dados 21TCPFile Transfer Protocol (FTP) - Control 22TCPSecure Shell (SSH) 23TCPTelnet25TCPSimple Mail Transfer Protocol (SMTP) 53UDP, TCPDomain Serviço de nome (DNS) 67UDPDisco dinâmico de configuração de host (DHCP) - Protocolo de Configuração de Host Server68UDPDynamic - Arquivo Trivial 69UDPTrivial do Cliente Protocolo de transferência (TFTP) 80TCPhyperText Transfer Protocol (HTTP) 110TCPPost Protocolo do Office versão 3 (POP3) 93TCPInternet Message Access Protocol (IMAP) 161UDPSimple Network Management Protocol (SNMP) 443TCPyperText Protocolo de transferência seguro (HTTPS)

Número da Porta	Protocolo	Aplicação
20	TCP	Protocolo de transferência de arquivos (FTP) - Dados
21	TCP	Protocolo de transferência de arquivos (FTP) - Controle
22	TCP	Secure Shell (SSH)
23	TCP	Telnet
25	TCP	Protocolo SMTP
53	UDP, TCP	Protocolo DNS

PortaNumberProtocolApplication20TCPFile Transfer Protocol (FTP) - Dados 21TCPFile Transfer Protocol (FTP) - Control 22TCPSecure Shell (SSH) 23TCPTelnet25TCPSimple Mail Transfer Protocol (SMTP) 53UDP, TCPDomain Serviço de nome (DNS) 67UDPDisco dinâmico de configuração de host (DHCP) - Protocolo de Configuração de Host Server68UDPDynamic - Arquivo Trivial 69UDPTrivial do Cliente Protocolo de transferência (TFTP) 80TCPhyperText Transfer Protocol (HTTP) 110TCPPost Protocolo do Office versão 3 (POP3) 93TCPInternet Message Access Protocol (IMAP) 161UDPSimple Network Management Protocol (SNMP) 443TCPyperText Protocolo de transferência seguro (HTTPS)

Número da Porta	Protocolo	Aplicação
67	UDP	Protocolo de Configuração Dinâmica de Host (DHCP) - Servidor
68	UDP	Protocolo de configuração dinâmica de host - cliente
69	UDP	Protocolo de Transferência Trivial de Arquivo (TFTP)
80	TCP	Protocolo HTTP
110	TCP	Protocolo POP3 (Post Office Protocol - Protocolo dos Correios)
93	TCP	Protocolo IMAP
161	UDP	Protocolo de Gerenciamento Simples de Rede (SNMP)
443	TCP	HTTPS (Secure Hypertext Transfer Protocol - Protocolo de Transferência de Hipertexto Seguro)

Algumas aplicações podem usar tanto TCP quanto UDP. Por exemplo, o DNS usa o protocolo UDP quando os clientes enviam requisições a um servidor DNS. Contudo, a comunicação entre dois servidores DNS sempre usa TCP.

Pesquise no site da IANA o registro de portas para visualizar a lista completa de números de portas e aplicativos associados.

9.4.4 O Comando netstat

Conexões TCP desconhecidas podem ser uma ameaça de segurança maior. Elas podem indicar que algo ou alguém está conectado ao host local. Às vezes é necessário conhecer quais conexões TCP ativas estão abertas e sendo executadas em um host de rede. O netstat é um utilitário de rede importante que pode ser usado para verificar essas conexões. Como mostrado abaixo, digite o comando **netstat** para listar os protocolos em uso, o endereço local e os números de porta, o endereço externo e os números de porta e o estado da conexão.

```
C:\> netstat

Conexões Ativas

  Proto  Endereço Local         Endereço Estrangeiro       Estado
  TCP    192.168.1.124:3126     192.168.0.2:netbios-ssn    Estabelecido
  TCP    192.168.1.124:3158     207.138.126.152:http       Estabelecido
  TCP    192.168.1.124:3159     207.138.126.169:http       Estabelecido
```

```
  TCP    192.168.1.124:3160     207.138.126.169:http      Estabelecido
  TCP    192.168.1.124:3161     sc.msn.com:http           Estabelecido
  TCP    192.168.1.124:3166     www.cisco.com:http        Estabelecido
(output omitted)
C:\>
```

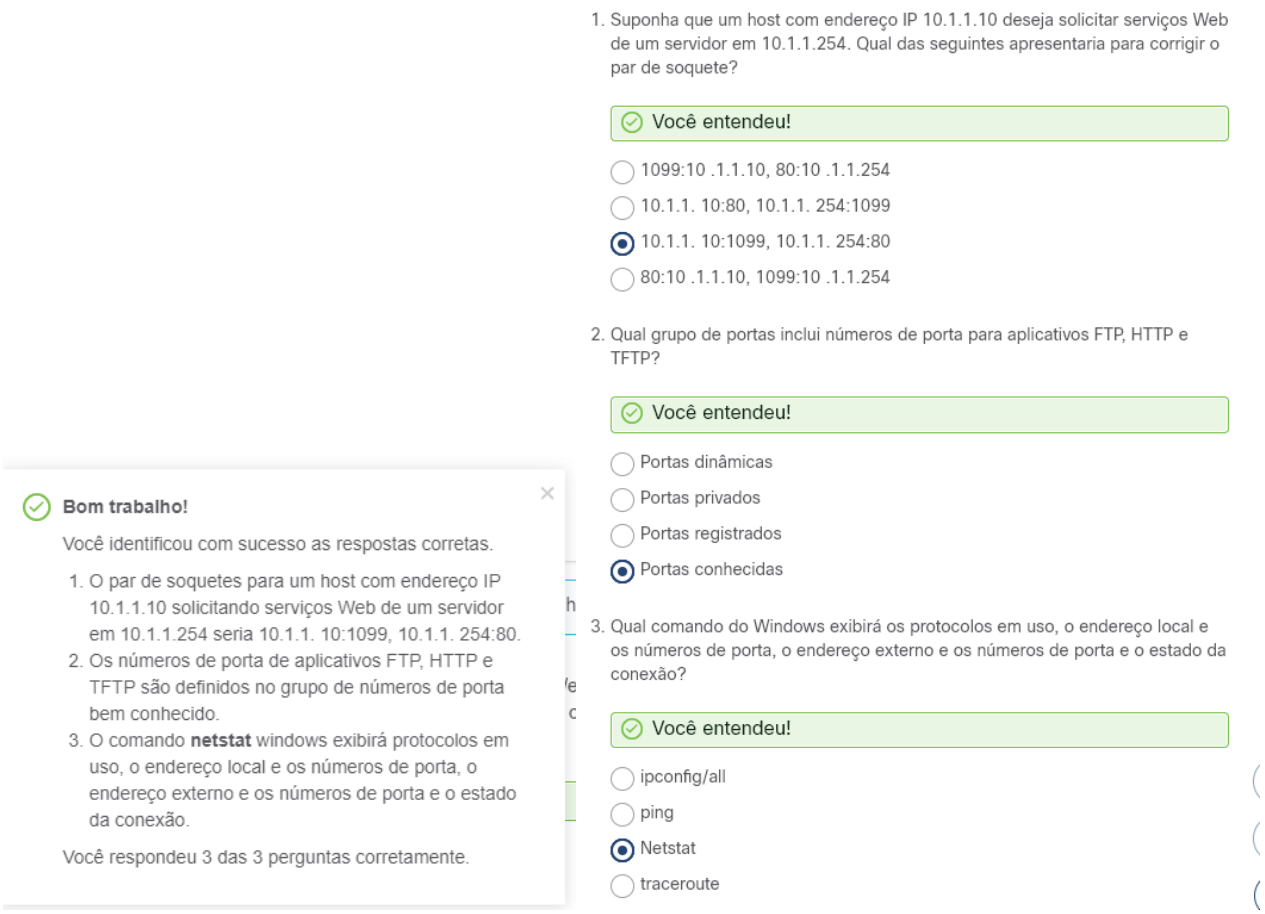
Por padrão, o comando **netstat** tentará resolver endereços IP para nomes de domínio e números de porta para aplicativos conhecidos. A **-n** opção pode ser usada para exibir endereços IP e números de porta em sua forma numérica.

9.4.5

Verifique sua compreensão - números de porta

Verifique sua compreensão dos números de porta, escolhendo a melhor resposta para as seguintes perguntas.

1. Suponha que um host com endereço IP 10.1.1.10 deseja solicitar serviços Web de um servidor em 10.1.1.254. Qual das seguintes apresentaria para corrigir o par de soquete?
   - 1099:10 .1.1.10, 80:10 .1.1.254
   - 10.1.1. 10:80, 10.1.1. 254:1099
   - 10.1.1. 10:1099, 10.1.1. 254:80
   - 80:10 .1.1.10, 1099:10 .1.1.254
2. Qual grupo de portas inclui números de porta para aplicativos FTP, HTTP e TFTP?
   - Portas dinâmicas
   - Portos privados
   - Portos registrados
   - Portas conhecidas
3. Qual comando do Windows exibirá os protocolos em uso, o endereço local e os números de porta, o endereço externo e os números de porta e o estado da conexão?
   - ipconfig/all
   - ping
   - Netstat
   - traceroute



9.5 Processo de Comunicação TCP

9.5.1Processos em Servidores TCP

Você já conhece os fundamentos do TCP. Compreender a função dos números de porta irá ajudá-lo a compreender os detalhes do processo de comunicação TCP. Neste tópico, você também aprenderá sobre os processos de handshake de três vias e terminação de sessão TCP.

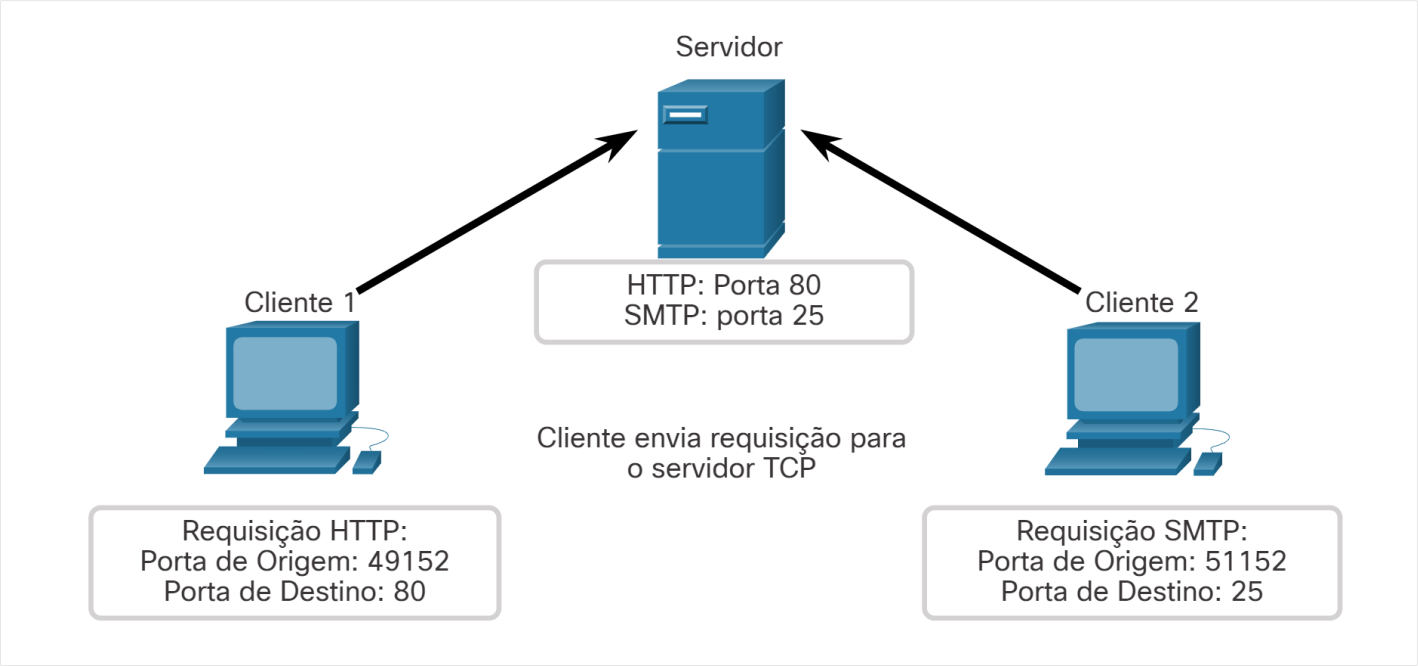
Cada processo de aplicativo em execução em um servidor está configurado para usar um número de porta. O número da porta é atribuído automaticamente ou configurado manualmente por um administrador do sistema.

Um servidor individual não pode ter dois serviços atribuídos ao mesmo número de porta dentro dos mesmos serviços de camada de transporte. Por exemplo, um host executando um aplicativo de servidor web e um aplicativo de transferência de arquivos não pode ter os dois configurados para usar a mesma porta, como a porta TCP 80.

Um aplicativo de servidor ativo atribuído a uma porta específica é considerado aberto, o que significa que a camada de transporte aceita e processa os segmentos endereçados a essa porta. Qualquer solicitação de cliente que chega endereçada ao soquete correto é aceita e os dados são transmitidos à aplicação do servidor. Pode haver muitas portas abertas ao mesmo tempo em um servidor, uma para cada aplicação de servidor ativa.

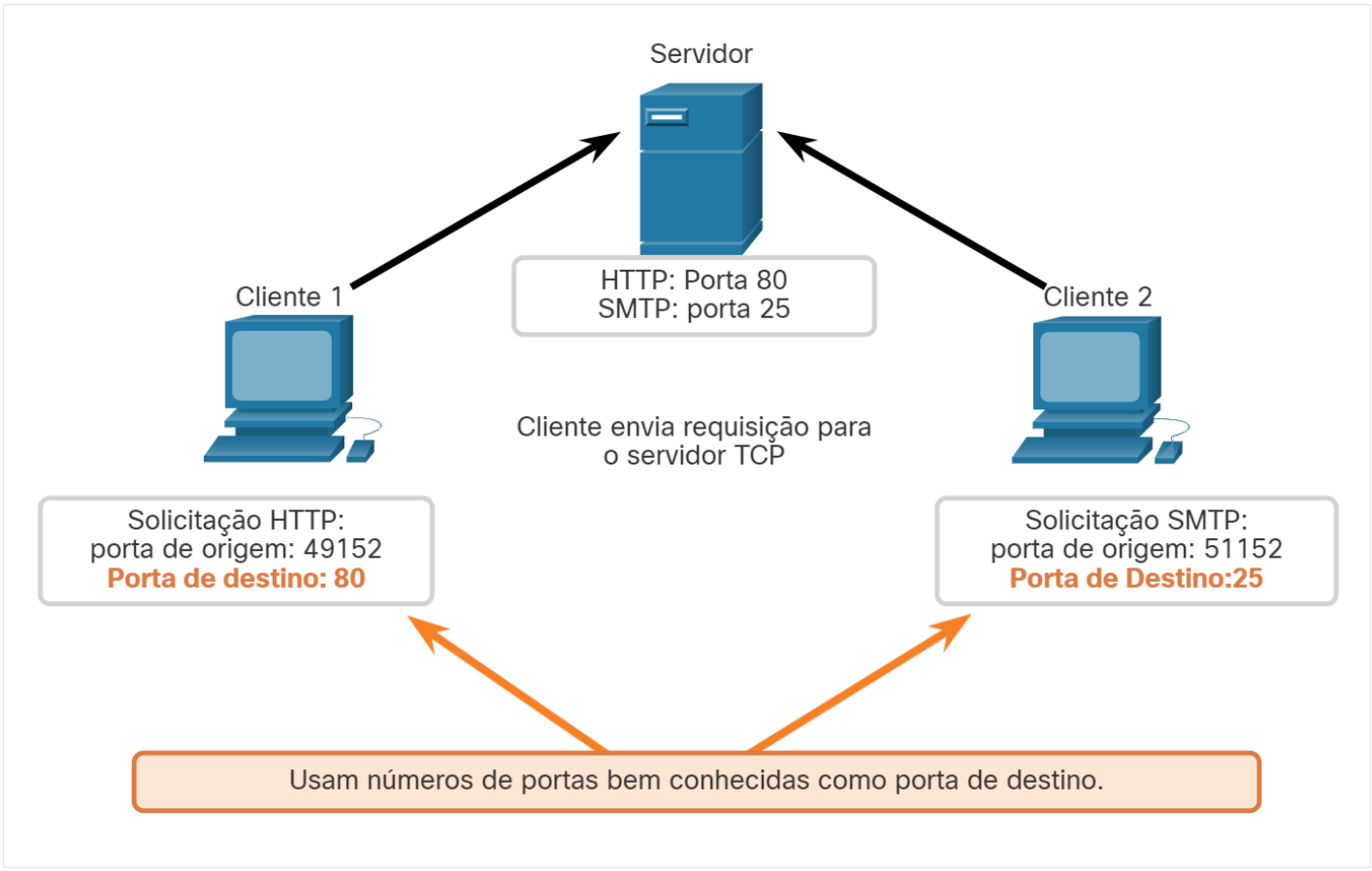
Clientes Enviando Requisições TCP

O Cliente 1 está a solicitar serviços Web e o Cliente 2 está a solicitar o serviço de correio electrónico do mesmo servidor.



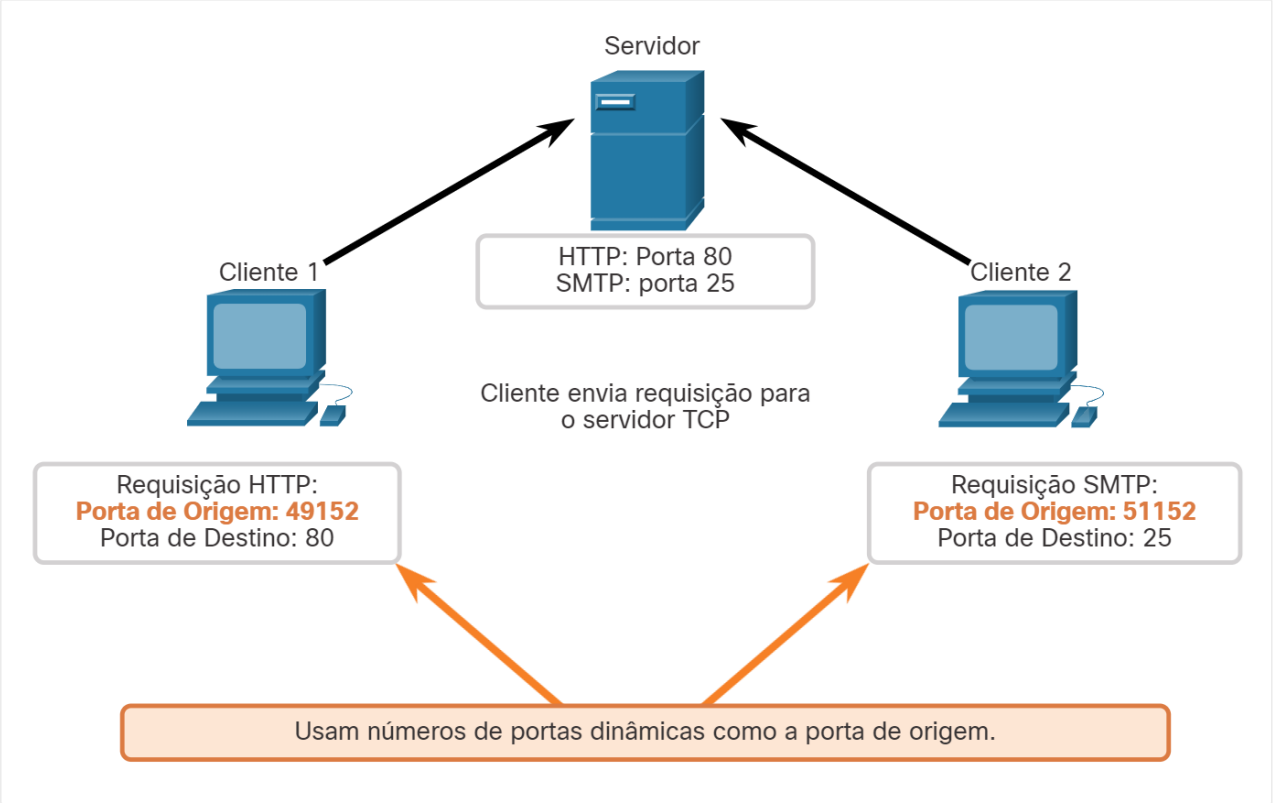
Portas de Destino das Requisições

O cliente 1 está solicitando serviços Web usando a porta 80 de destino bem conhecida (HTTP) e o cliente 2 está solicitando o serviço de email usando a porta 25 (SMTP) bem conhecida.



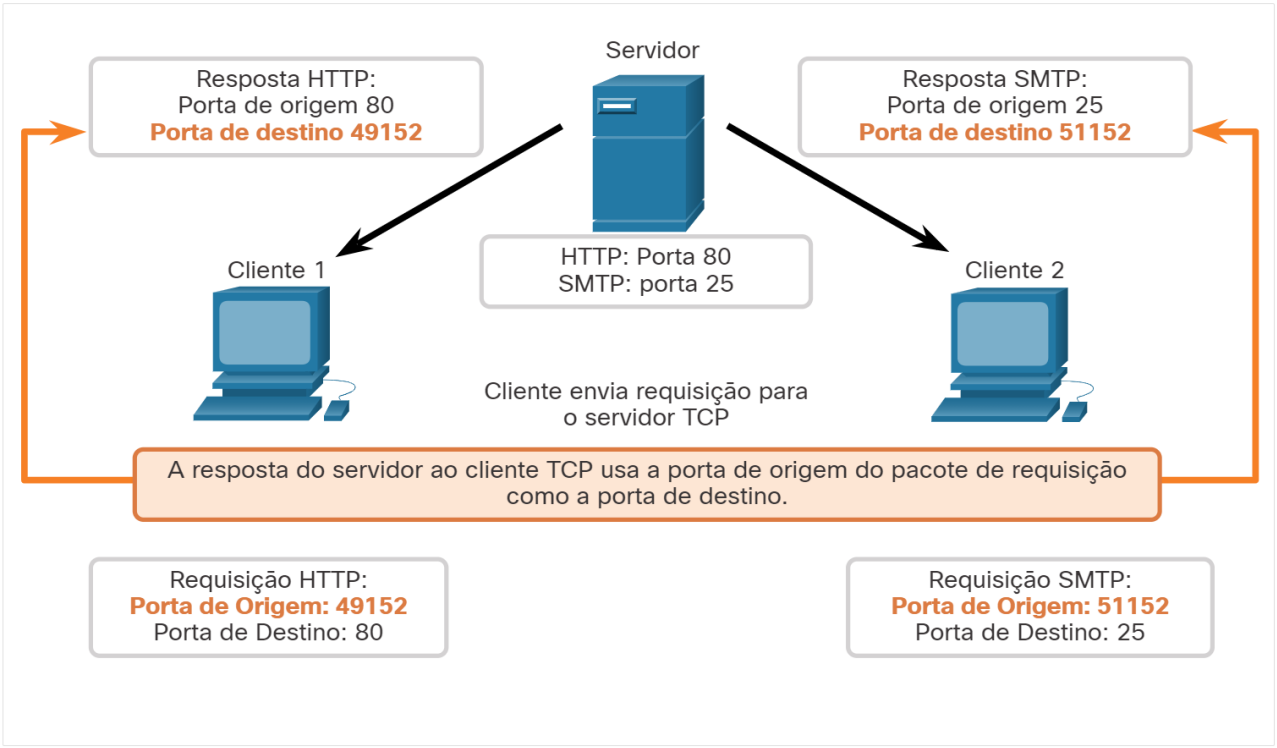
Portas de Origem das Requisições

As solicitações do cliente geram dinamicamente um número de porta de origem. Nesse caso, o cliente 1 está usando a porta de origem 49152 e o cliente 2 está usando a porta de origem 51152.



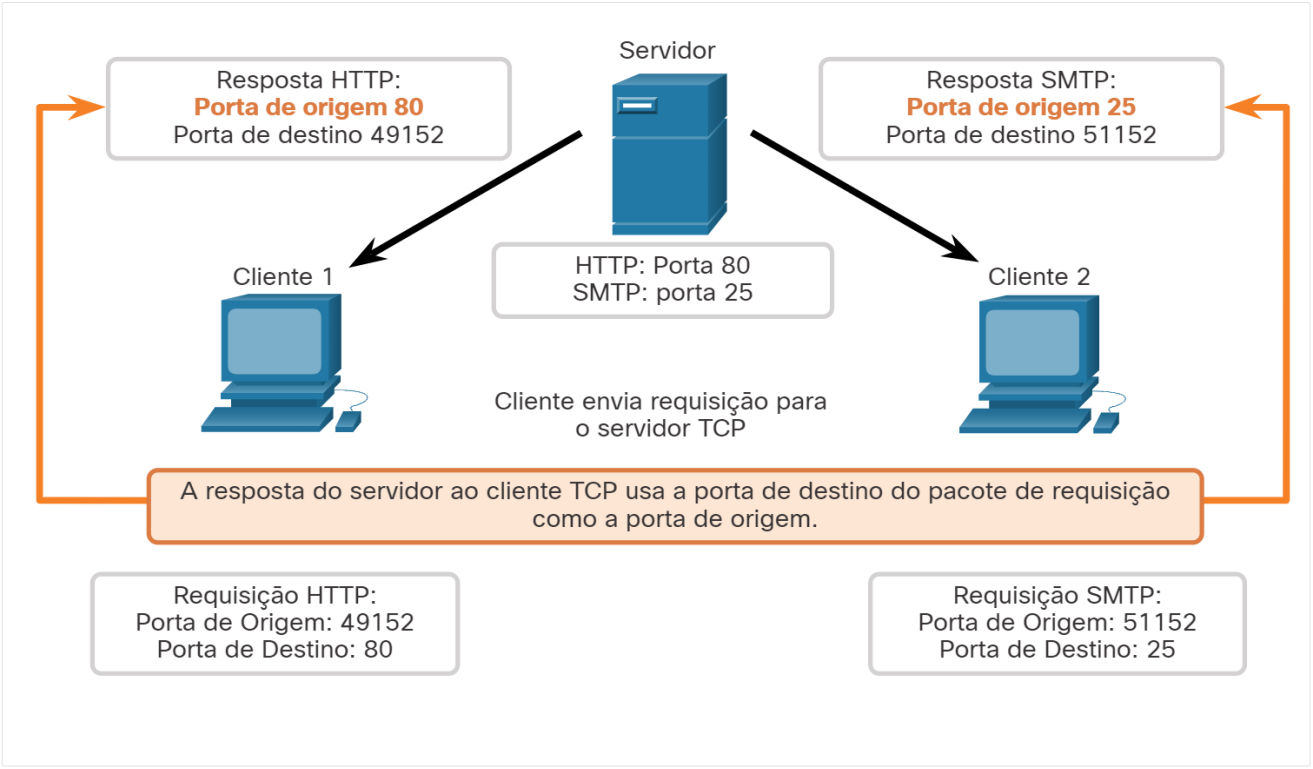
Portas de Destino das Respostas

Quando o servidor responde às solicitações do cliente, ele reverte as portas de destino e de origem da solicitação inicial. Observe que a resposta do servidor à solicitação da Web agora tem a porta de destino 49152 e a resposta de e-mail agora tem a porta de destino 51152.



Portas de Origem das Respostas

A porta de origem na resposta do servidor é a porta de destino original nas solicitações iniciais.

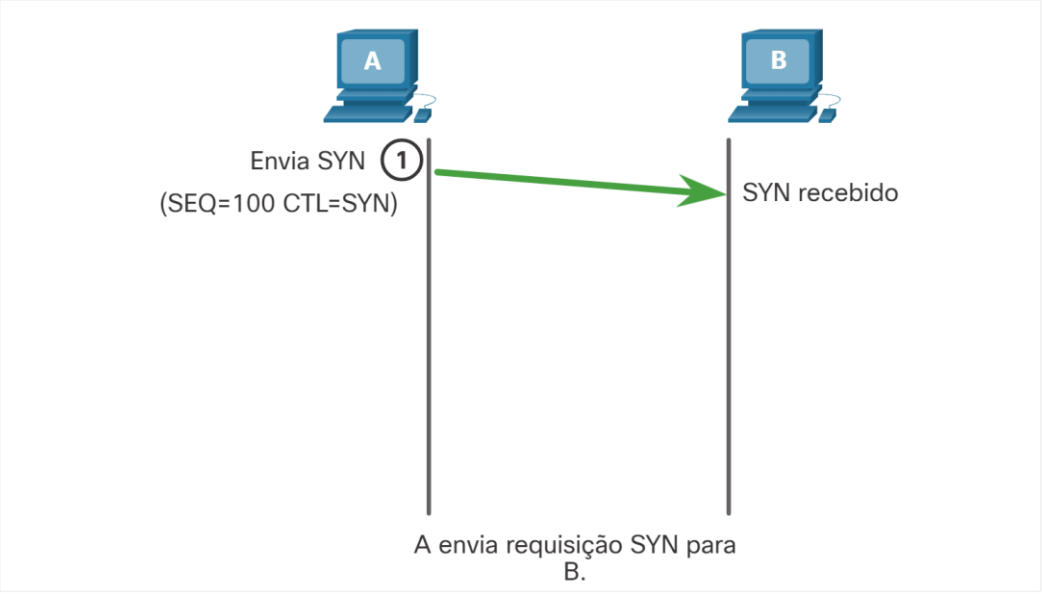


9.5.2 Estabelecimento de Conexão TCP

Em algumas culturas, quando duas pessoas se encontram, elas costumam se cumprimentar apertando as mãos. Ambas as partes entendem o ato de apertar as mãos como um sinal para uma saudação amigável. As conexões de rede são semelhantes. Nas conexões TCP, o cliente host estabelece a conexão com o servidor usando o processo de handshake de três vias.

Etapa 1. SYN

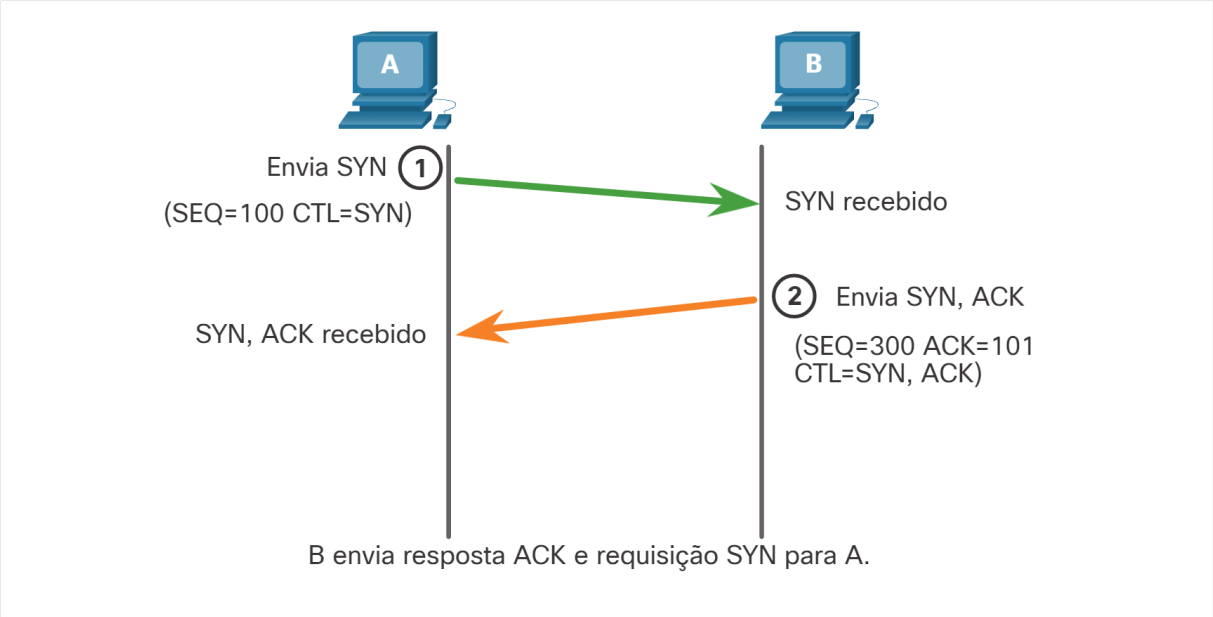
O cliente iniciador requisita uma sessão de comunicação cliente-servidor com o servidor.



O handshake de três vias valida se o host de destino está disponível para comunicação. Neste exemplo, o host A validou que o host B está disponível.

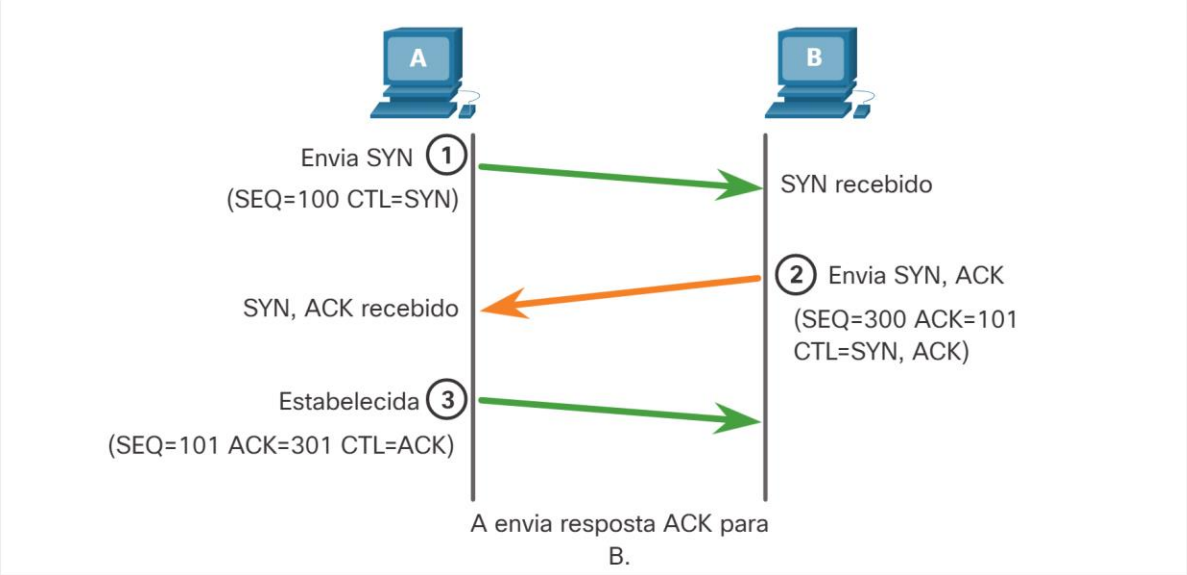
Etapa 2. ACK e SYN

O servidor confirma a sessão de comunicação cliente-servidor e requisita uma sessão de comunicação de servidor-cliente.



Etapa 3. ACK

O cliente iniciador confirma a sessão de comunicação de servidor-cliente.



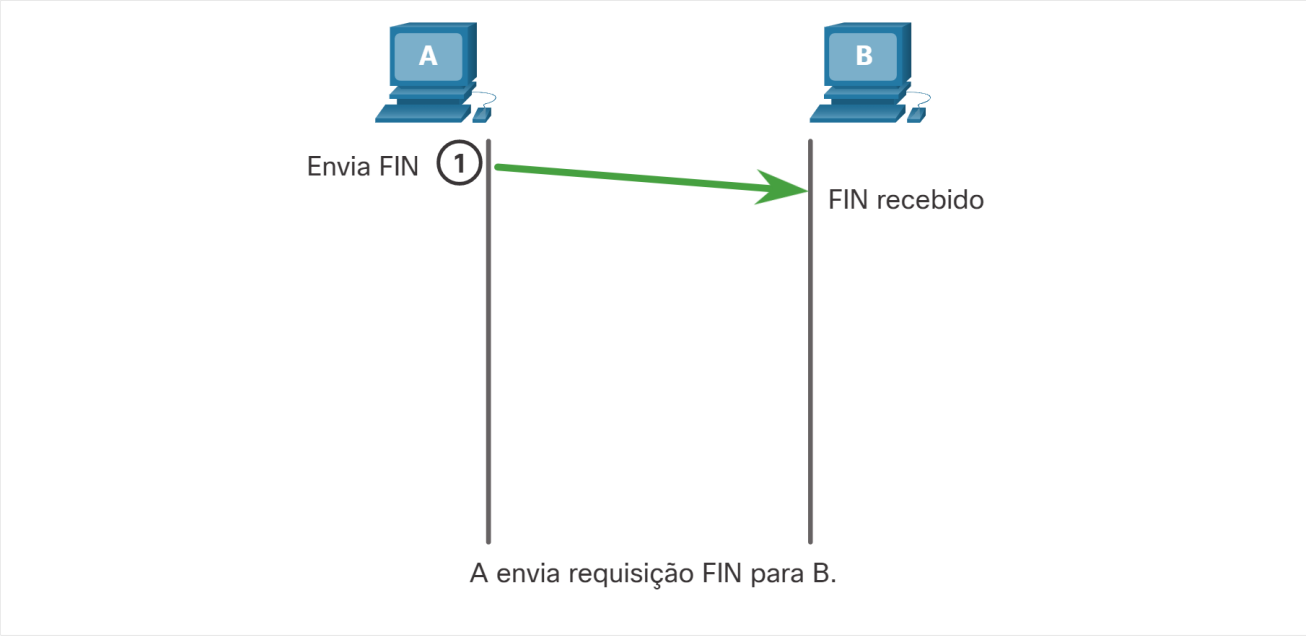
9.5.3 Encerramento da Sessão

Para fechar uma conexão, o flag de controle Finish (FIN) deve ser ligado no cabeçalho do segmento. Para terminar cada sessão TCP de uma via, um handshake duplo, consistindo de um segmento FIN e um segmento ACK (Acknowledgment) é usado. Portanto, para terminar uma conversação única permitida pelo TCP, quatro trocas são necessárias para finalizar ambas as sessões. O cliente ou o servidor podem iniciar o encerramento.

No exemplo, os termos cliente e servidor são usados como referência para simplificar, mas dois hosts que possuem uma sessão aberta podem iniciar o processo de finalização.

Etapa 1. FIN

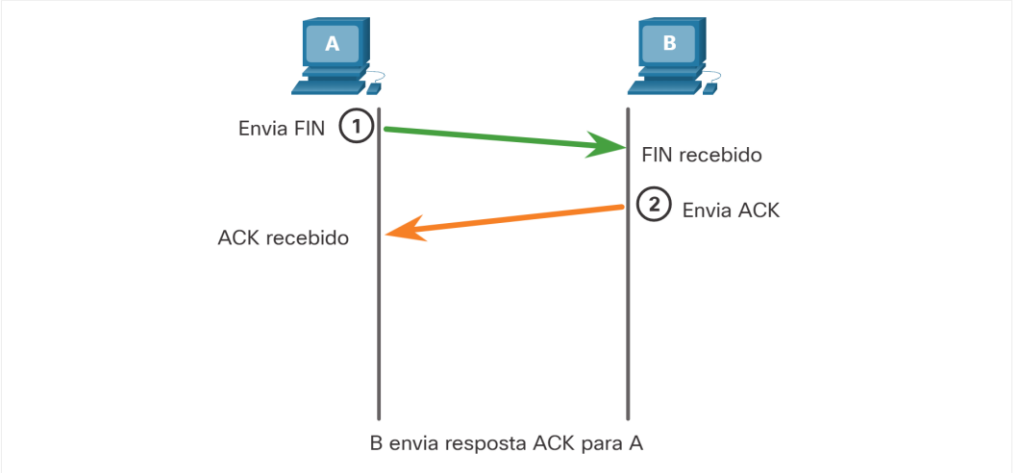
Quando o cliente não tem mais dados para enviar no fluxo, ele envia um segmento com um flag FIN ligado.



Etapa 2. ACK

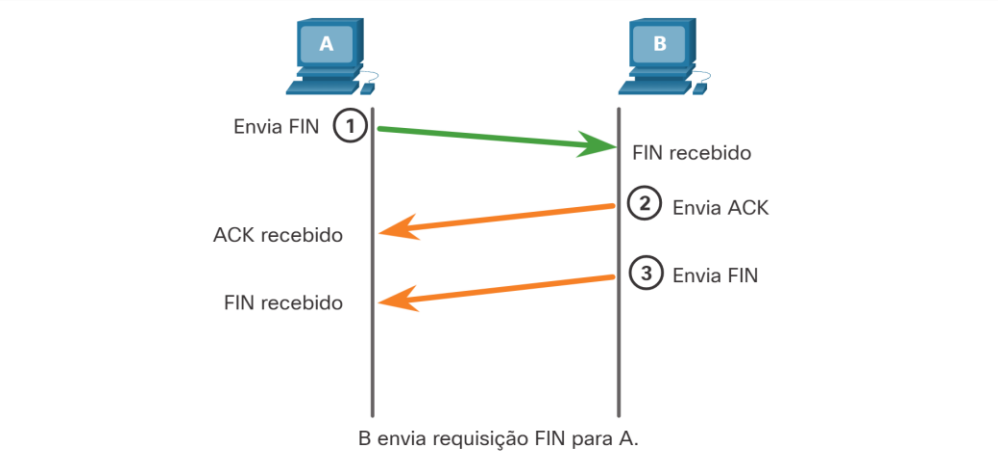
O servidor envia ACK para confirmar o recebimento de FIN para encerrar a sessão do cliente com o servidor.

Quando todos os segmentos tiverem sido reconhecidos, a sessão é encerrada.



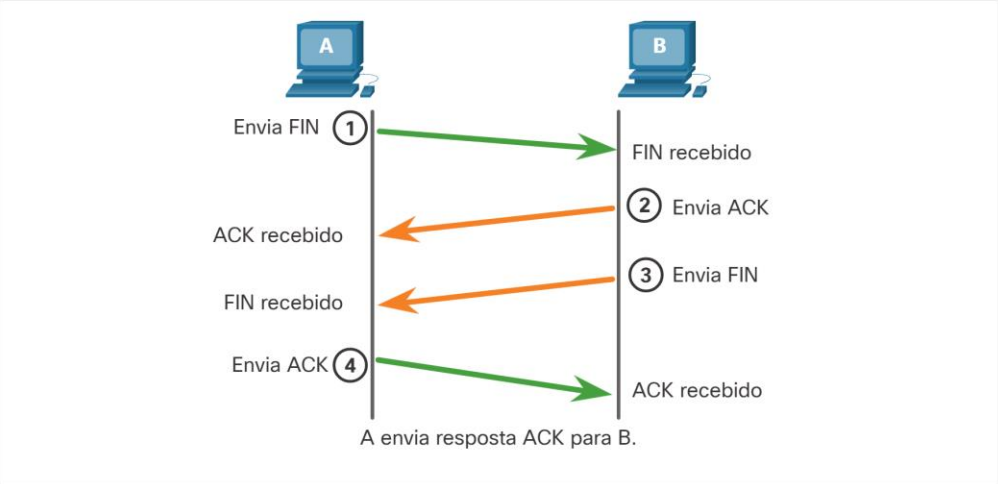
Etapa 3. FIN

O servidor envia um FIN ao cliente para encerrar a sessão do servidor-para-cliente.



Etapa 4. ACK

O cliente responde com um ACK para reconhecer o FIN do servidor.



9.5.4 Análise do Handshake Triplo do TCP

Os hosts mantêm o estado, rastreiam cada segmento de dados em uma sessão e trocam informações sobre quais dados são recebidos usando as informações no cabeçalho TCP. O TCP é um protocolo full-duplex, em que cada conexão representa duas sessões de comunicação unidirecional. Para estabelecer uma conexão, os hosts realizam um handshake triplo (three-way handshake). Conforme mostrado na figura, os bits de controle no cabeçalho TCP indicam o progresso e o status da conexão.

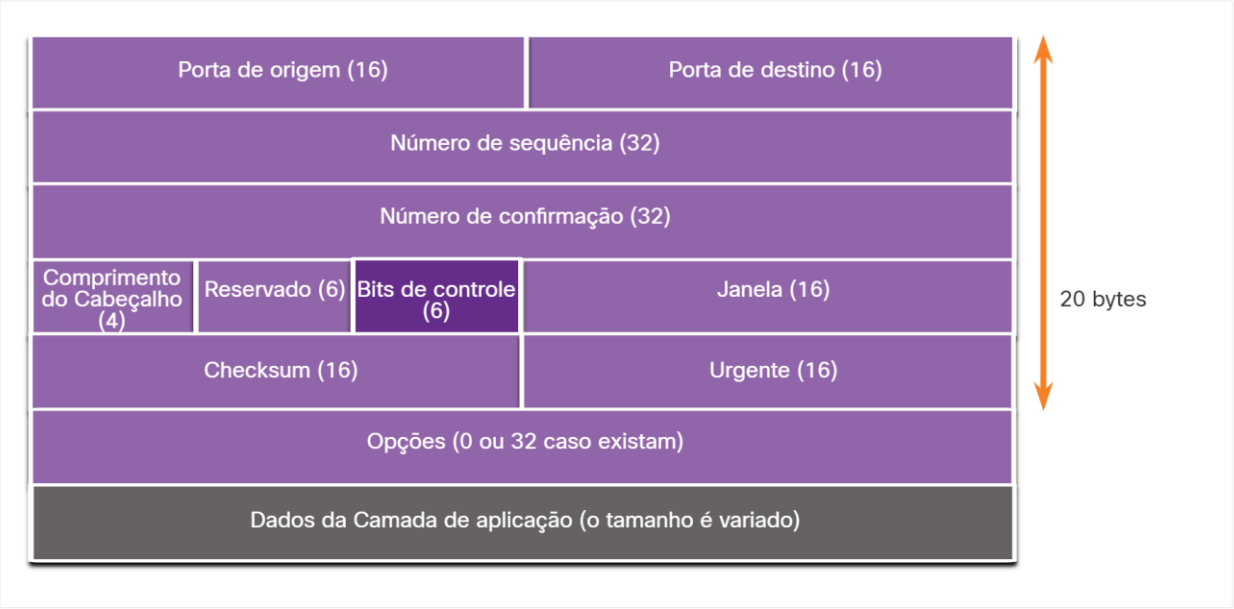
Estas são as funções do handshake de três vias:

- Estabelece que o dispositivo de destino está presente na rede.
- Ele verifica se o dispositivo de destino possui um serviço ativo e está aceitando solicitações no número da porta de destino que o cliente inicial pretende usar.
- Ele informa ao dispositivo de destino que o cliente de origem pretende estabelecer uma sessão de comunicação nesse número de porta.

Após a conclusão da comunicação, as sessões são fechadas e a conexão é encerrada. Os mecanismos de conexão e sessão ativam a função de confiabilidade do TCP.

mostra os campos de cabeçalho do segmento tcp com o campo de bits de controle de 6 bits destacado

Campo de bits de controle



Os seis bits no campo Bits de Controle do cabeçalho do segmento TCP são também conhecidos como flags. Um sinalizador é um pouco definido como ativado ou desativado.

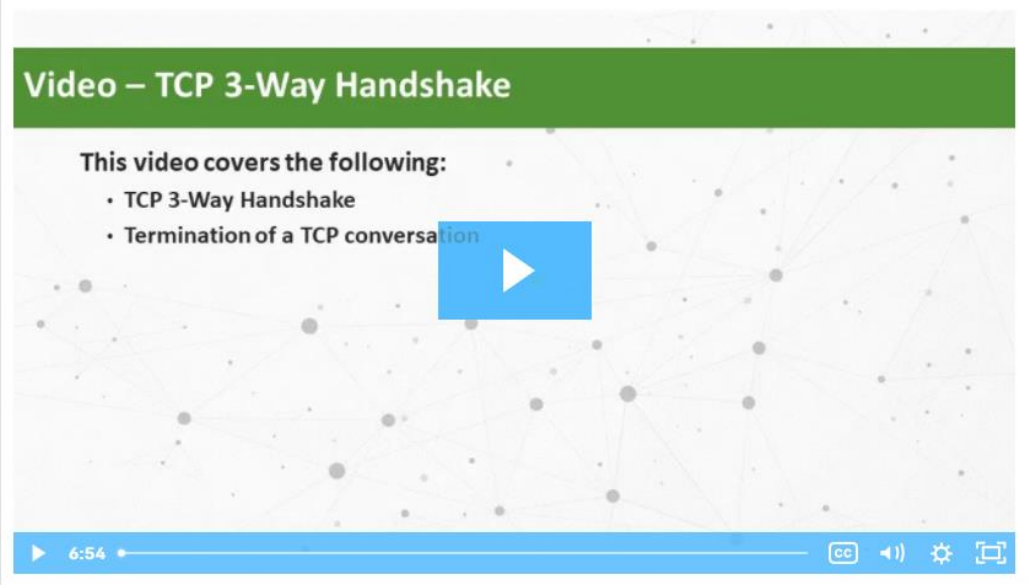
Os seis bits de controle sinalizadores são os seguintes:

- **URG** - Campo de ponteiro urgente significativo.
- **ACK** - Indicador de confirmação usado no estabelecimento de conexão e encerramento de sessão.
- **PSH** - Função Push.
- **RST** - Redefina a conexão quando ocorrer um erro ou tempo limite.
- **SYN** - Sincronizar números de sequência usados no estabelecimento de conexão.
- **FIN** - Não há mais dados do remetente e usados no encerramento da sessão.

Pesquise na Internet para saber mais sobre as bandeiras PSH e URG.

9.5.5 Vídeo - Handshake de 3 vias TCP

Clique em Reproduzir na figura para ver uma demonstração em vídeo do handshake de 3 vias TCP, usando o Wireshark.



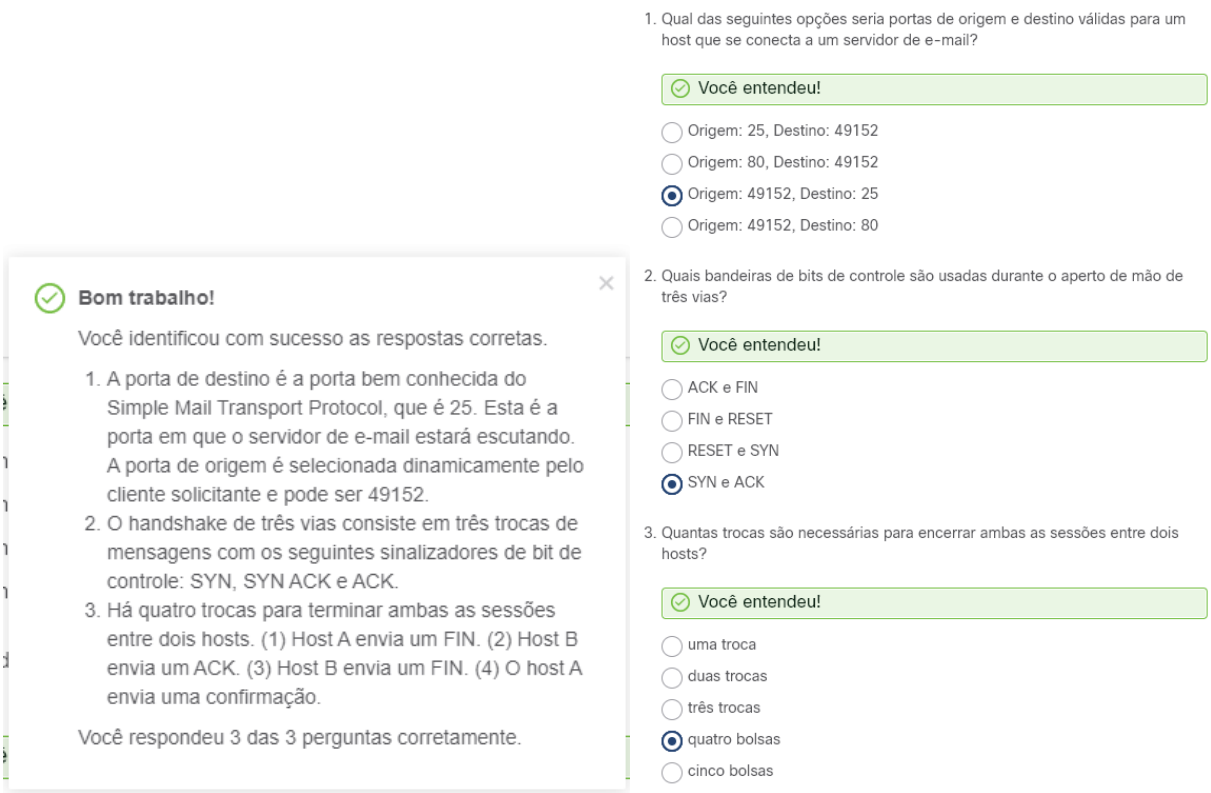
Duração: 6:54

9.5.6 Verifique o seu entendimento - Processo de comunicação TCP

Verifique sua compreensão do processo de comunicação TCP escolhendo a melhor resposta para as seguintes perguntas.

- Qual das seguintes opções seria portas de origem e destino válidas para um host que se conecta a um servidor de e-mail?
- Fonte: 25, Destino: 49152
- Fonte: 80, Destino: 49152
- Fonte: 49152, Destino: 25
- Fonte: 49152, Destino: 80

2. Quais bandeiras de bits de controle são usadas durante o aperto de mão de três vias?
   - ACK e FIN
   - FIN e RESET
   - RESET e SYN
   - SYN e ACK
3. Quantas trocas são necessárias para encerrar ambas as sessões entre dois hosts?
   - uma troca
   - duas trocas
   - três trocas
   - quatro bolsas
   - cinco bolsas



9.6 Confiabilidade e controle de fluxo

9.6.1 Confiabilidade do TCP - Entrega garantida e solicitada

A razão pela qual o TCP é o melhor protocolo para alguns aplicativos é porque, ao contrário do UDP, ele reenvia pacotes descartados e números de pacotes para indicar sua ordem correta antes da entrega. O TCP também pode ajudar a manter o fluxo de pacotes para que os dispositivos não fiquem sobrecarregados. Este tópico aborda esses recursos do TCP em detalhes.

Pode haver momentos em que os segmentos TCP não chegam ao seu destino. Outras vezes, os segmentos TCP podem chegar fora de ordem. Para que a mensagem original seja entendida pelo destinatário, todos os dados devem ser recebidos e os dados nesses segmentos devem ser remontados na ordem original. Os números de sequência são atribuídos no cabeçalho de cada pacote para alcançar esse objetivo. O número de sequência representa o primeiro byte de dados do segmento TCP.

Durante o estabelecimento de uma sessão, um número de sequência inicial (ISN) é definido. Este ISN representa o valor inicial dos bytes que são transmitidos ao aplicativo receptor. À medida que os dados são transmitidos durante a sessão, número de sequência é incrementado do número de bytes que foram transmitidos. Esse rastreamento dos bytes de dados permite que cada segmento seja identificado e confirmado de forma única. Segmentos perdidos podem então, ser identificados.

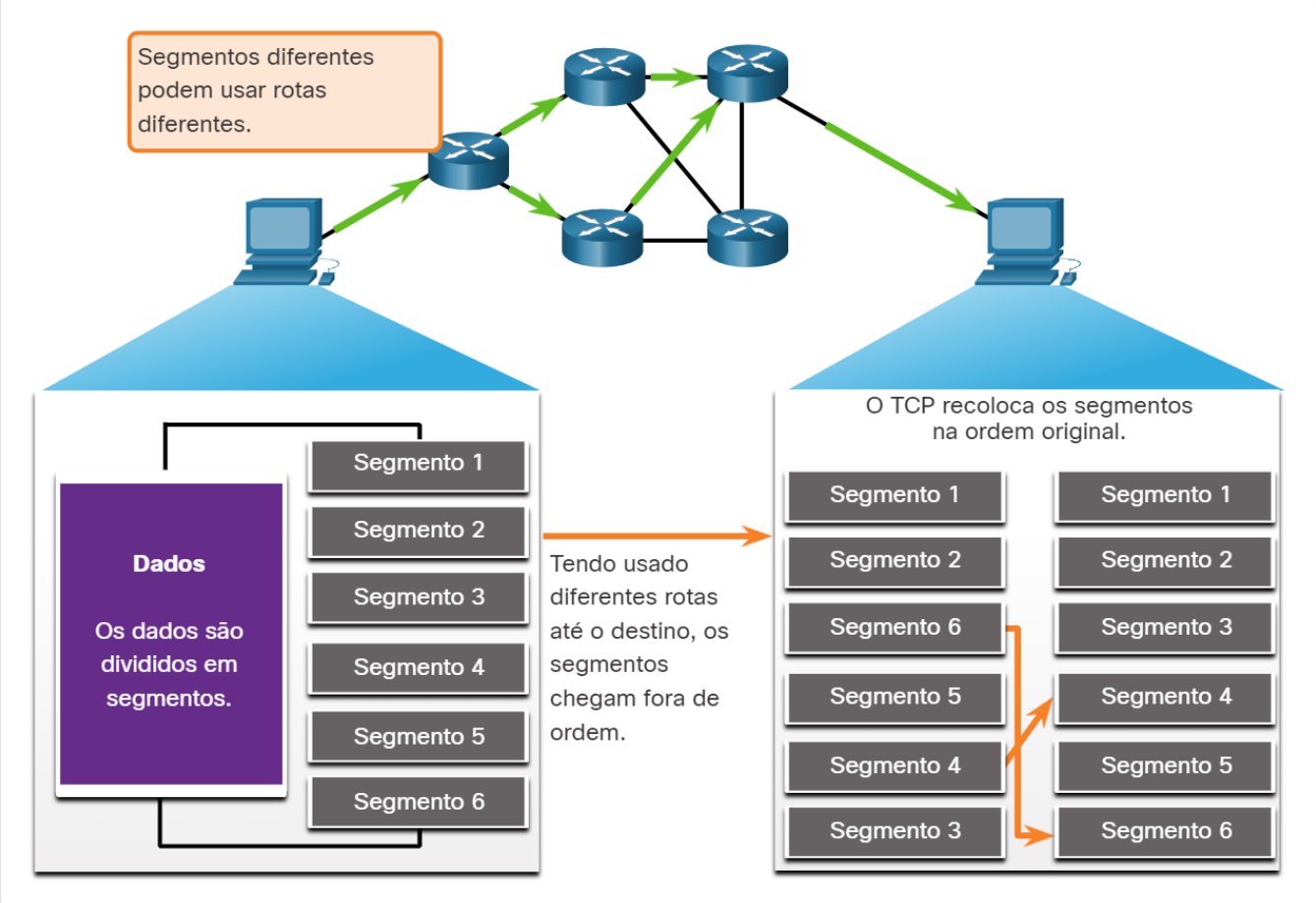
O ISN não começa em um, mas é efetivamente um número aleatório. Isso é para impedir determinados tipos de ataques maliciosos. Para simplificar os exemplos desse capítulo, usaremos um ISN de 1.

Os números de sequência do segmento indicam como remontar e reordenar os segmentos recebidos, como mostrado na figura.

mostra que, embora os segmentos possam tomar rotas diferentes e chegar fora de ordem no destino, o TCP tem a capacidade de reordenar os segmentos

Os Segmentos TCP São Reordenados no Destino

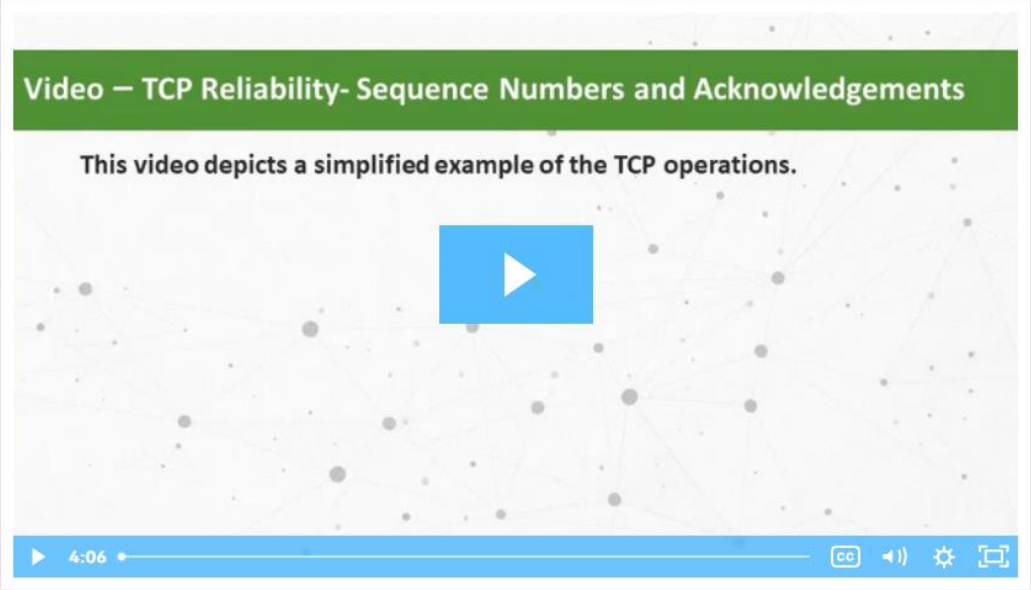
Os dados são divididos em segmentos.



O processo TCP receptor coloca os dados de um segmento em um buffer receptor. Os segmentos são então colocados na ordem de sequência correta e passados para a camada de aplicativo quando remontados. Qualquer segmento que chegue com números de sequência fora de ordem são retidos para processamento posterior. Por isso, quando os segmentos com os bytes que faltavam chegam, esses segmentos são processados.

9.6.2 Vídeo - Confiabilidade do TCP - Números de seqüência e Confirmações

Uma das funções do TCP é garantir que cada segmento chegue ao seu destino. Os serviços



Duração: 4:06

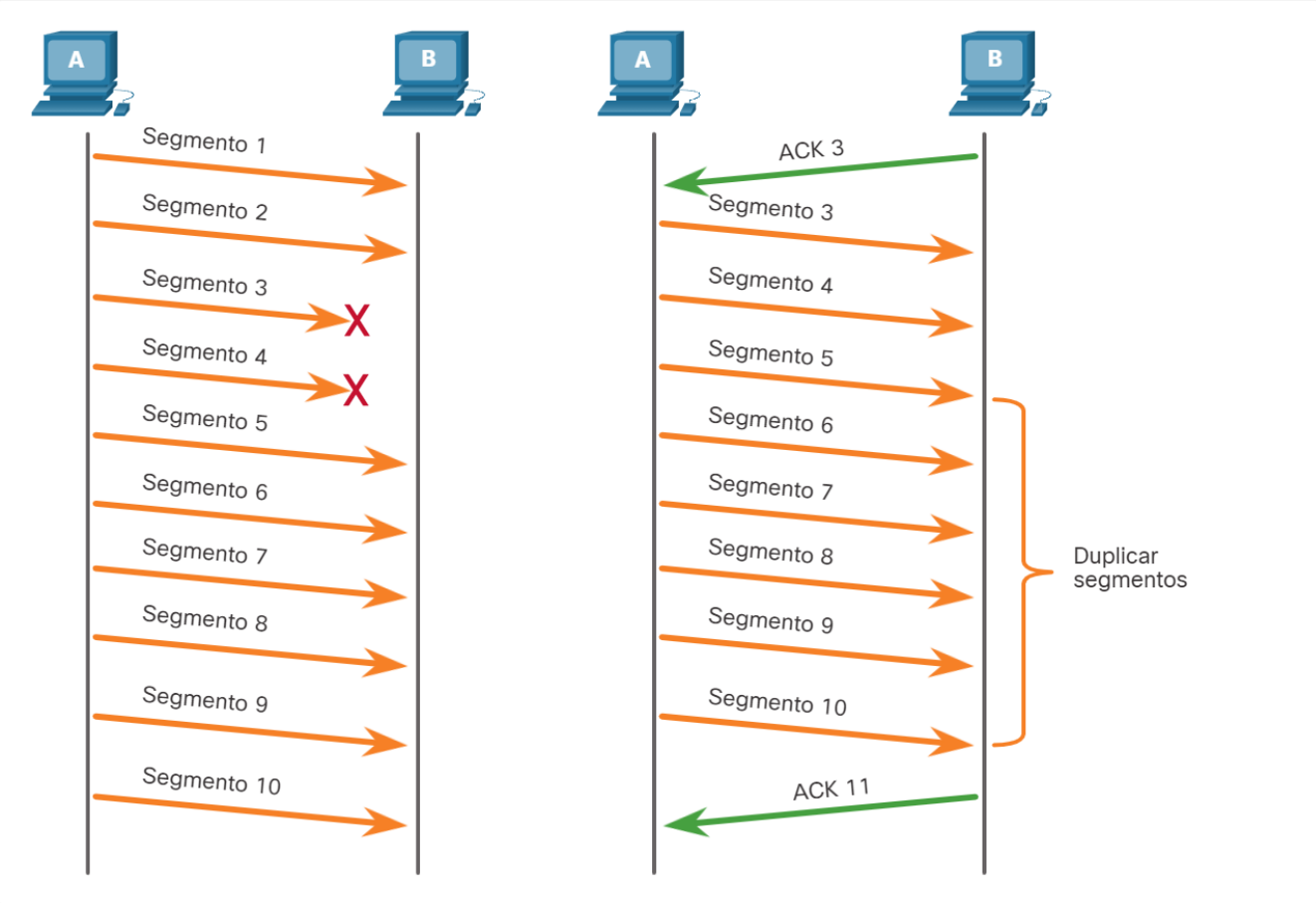
9.6.3 Confiabilidade do TCP - perda de dados e retransmissão

Não importa o quão bem projetada uma rede é, a perda de dados ocasionalmente ocorre. O TCP fornece métodos de gerenciamento dessas perdas de segmento. Entre esses métodos há um mecanismo que retransmite segmentos dos dados não confirmados.

O número de sequência (SEQ) e o número de confirmação (ACK) são usados juntamente para confirmar o recebimento dos bytes de dados contidos nos segmentos. O número SEQ identifica o primeiro byte de dados no segmento que está sendo transmitido. O TCP usa o número de confirmação (ACK) enviado de volta à origem para indicar o próximo byte que o destino espera receber. Isto é chamado de confirmação antecipatória.

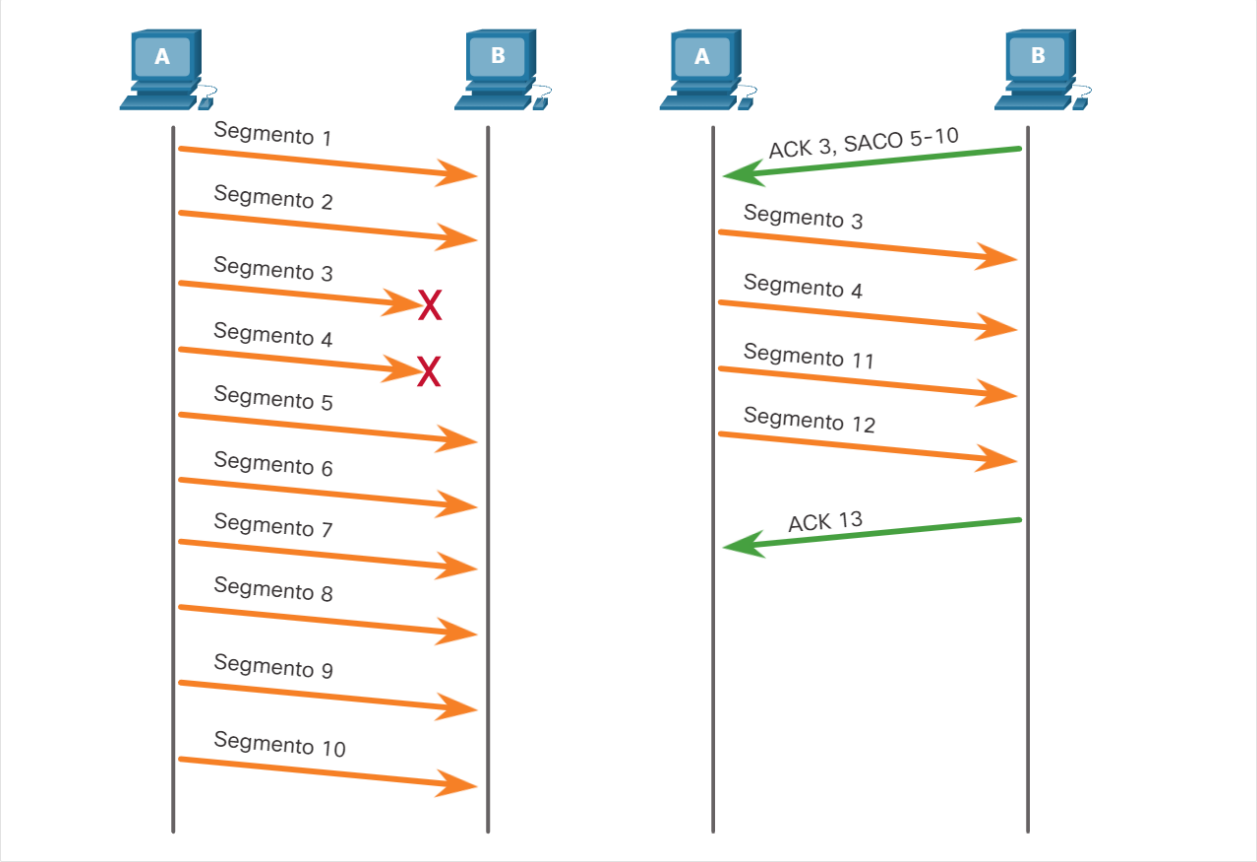
Antes de melhorias posteriores, o TCP só podia reconhecer o próximo byte esperado. Por exemplo, na figura, usando números de segmento para simplicidade, o host A envia os segmentos 1 a 10 para o host B. Se todos os segmentos chegarem, exceto os segmentos 3 e 4, o host B responderia com confirmação especificando que o próximo segmento esperado é o segmento 3. O Host A não tem idéia se outros segmentos chegaram ou não. O host A, portanto, reenviaria os segmentos 3 a 10. Se todos os segmentos reenviados chegarem com sucesso, os segmentos 5 a 10 seriam duplicados. Isso pode levar a atrasos, congestionamentos e ineficiências.

mostra PCA enviando 10 segmentos para PCB, mas os segmentos 3 e 4 não chegam. Assim, começando com o segmento 3, a PCA reende os segmentos 3 a 10, embora o PCB necessitasse apenas dos segmentos 3 e 4



Hoje em dia, os sistemas operacionais de host utilizam um recurso TCP opcional chamado reconhecimento seletivo (SACK), negociado durante o handshake de três vias. Se ambos os hosts suportarem SACK, o receptor pode reconhecer explicitamente quais segmentos (bytes) foram recebidos, incluindo quaisquer segmentos descontínuos. O host de envio, portanto, só precisa retransmitir os dados ausentes. Por exemplo, na próxima figura, novamente usando números de segmento para simplicidade, o host A envia segmentos 1 a 10 para o host B. Se todos os segmentos chegarem, exceto os segmentos 3 e 4, o host B pode reconhecer que recebeu segmentos 1 e 2 (ACK 3) e reconhecer seletivamente os segmentos 5 a 10 (SACK 5-10). O host A só precisaria reenviar os segmentos 3 e 4.

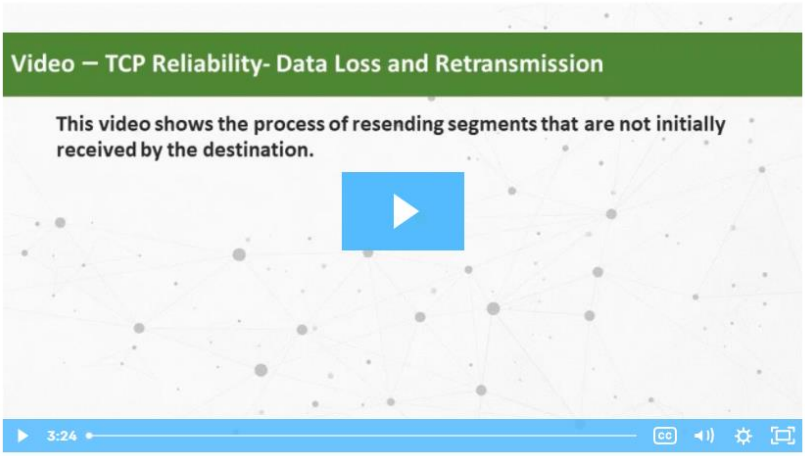
mostram PCA enviando 10 segmentos para PCB, mas os segmentos 3 e 4 não chegam. Desta vez PCB envia um ACK 3 e um SACK 5-10 deixando PCA saber para reenviar segmentos faltantes 3 e 4 e continuar com o segmento 11



Nota: O TCP normalmente envia ACKs para todos os outros pacotes, mas outros fatores além do escopo deste tópico podem alterar esse comportamento.

O TCP usa temporizadores para saber quanto tempo esperar antes de reenviar um segmento. Na figura, reproduza o vídeo e clique no link para baixar o arquivo PDF. O vídeo e o arquivo PDF examinam a perda de dados e a retransmissão TCP.

9.6.4 Vídeo - Confiabilidade TCP - Perda e retransmissão de dados



Duração: 3:24

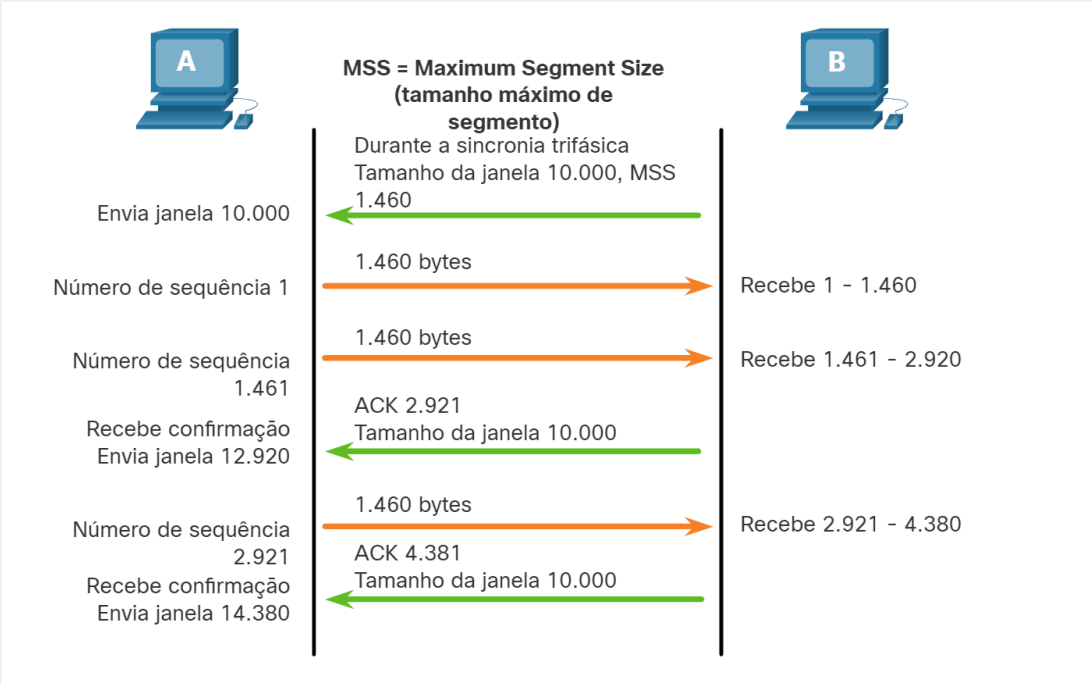
9.6.5 Controle de Fluxo TCP – Tamanho da Janela e Confirmações

O TCP também fornece mecanismos para controle de fluxo. Controle de fluxo é a quantidade de dados que o destino pode receber e processar de forma confiável. O controle de fluxo ajuda a manter a confiabilidade da transmissão TCP definindo a taxa de fluxo de dados entre a origem e o destino em uma determinada sessão. Para realizar isso, o cabeçalho TCP inclui um campo de 16 bits chamado de tamanho da janela.

A figura mostra um exemplo de tamanho da janela e confirmações.

mostra PCB enviando PCB um tamanho de janela negociado de 10.000 bytes e um tamanho máximo de segmento de 1.460 bytes. PCA começa a enviar segmentos começando com o número de sequência 1. Uma confirmação do PCB pode ser enviada sem esperar até que o tamanho da janela seja atingido e o tamanho da janela possa ser ajustado pela PCA criando uma janela deslizante

Exemplo de Tamanho da Janela TCP



O tamanho da janela determina o número de bytes que podem ser enviados antes de esperar uma confirmação. O número de reconhecimento é o número do próximo byte esperado.

O tamanho da janela é número de bytes que o dispositivo de destino de uma sessão TCP pode aceitar e processar de uma vez. Neste exemplo, o tamanho da janela inicial do PC B para a sessão TCP é de 10.000 bytes. No caso do primeiro byte ser número 1, o último byte que PC A pode enviar sem receber uma confirmação é o byte 10.000. Isso é conhecido como janela de envio do PC A. O tamanho da janela é incluído em todos os segmentos TCP, para que o destino possa modificar o tamanho da janela a qualquer momento, dependendo da disponibilidade do buffer.

O tamanho da janela inicial é determinado quando a sessão é estabelecida durante o handshake triplo. O dispositivo de origem deve limitar o número de bytes enviados ao dispositivo de destino com base no tamanho da janela do destino. Somente depois que o dispositivo de origem receber uma confirmação de que os bytes foram recebidos, ele poderá continuar a enviar mais dados para a sessão. Normalmente, o destino não esperará que todos os bytes que a sua janela comporta sejam recebidos para responder confirmando. À medida que os bytes forem recebidos e processados, o destino enviará confirmações para informar à origem que pode continuar a enviar bytes adicionais.

Por exemplo, é típico que o PC B não espere até que todos os 10.000 bytes tenham sido recebidos antes de enviar uma confirmação. Isso significa que o PC A pode ajustar sua janela de envio ao receber confirmações do PC B. Como mostrado na figura, quando o PC A recebe uma confirmação com o número de confirmação 2.921, que é o próximo byte esperado. A janela de envio do PC A irá incrementar 2.920 bytes. Isso altera a janela de envio de 10.000 bytes para 12.920. O PC A agora pode continuar enviando até outros 10.000 bytes para o PC B, desde que não envie mais do que sua nova janela de envio em 12.920.

Um destino que envia confirmações enquanto processa os bytes recebidos e o ajuste contínuo da janela de envio de origem é conhecido como janelas deslizantes. No exemplo anterior, a janela de envio do PC A incrementa ou desliza sobre outros 2.921 bytes de 10.000 para 12.920.

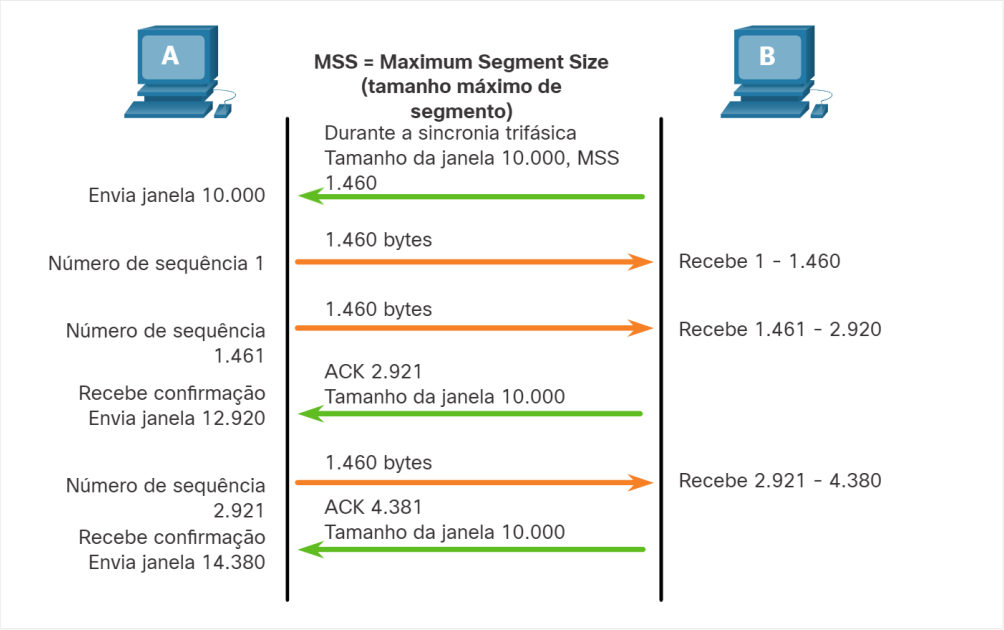
Se a disponibilidade do espaço de buffer do destino diminui, ele pode reduzir o tamanho da sua janela para informar à origem que reduza o número de bytes que ela deveria enviar sem receber uma confirmação.

Nota: Os dispositivos hoje usam o protocolo de janelas deslizantes. O receptor normalmente envia uma confirmação após cada dois segmentos que recebe. O número de segmentos recebidos antes de ser confirmado pode variar. A vantagem de janelas móveis é que permite que o emissor transmita continuamente segmentos, desde que o receptor esteja reconhecendo segmentos anteriores. Os detalhes das janelas móveis estão fora do escopo deste curso.

9.6.6 Controle de Fluxo TCP - Tamanho Máximo do Segmento (MSS)

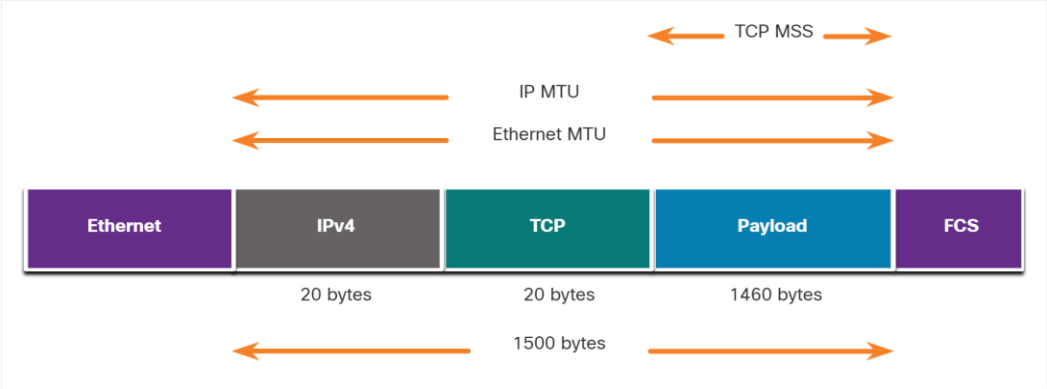
Na figura, a fonte está transmitindo 1.460 bytes de dados dentro de cada segmento TCP. Normalmente, este é o tamanho máximo do segmento (MSS) que o dispositivo de destino pode receber. O MSS faz parte do campo de opções no cabeçalho TCP que especifica a maior quantidade de dados, em bytes, que um dispositivo pode receber em um único segmento TCP. O tamanho do MSS não inclui o cabeçalho TCP. O MSS é normalmente incluído durante o handshake de três vias.

mostra o mesmo diagrama de antes, mas a ênfase está no MSS de tamanho máximo de segmento de 960



Um MSS comum é 1.460 bytes ao usar IPv4. Um host determina o valor do campo de MSS subtraindo os cabeçalhos de IP e de TCP da MTU (Maximum transmission unit, Unidade máxima de transmissão) da Ethernet. Em uma interface Ethernet, a MTU padrão é 1500 bytes. Subtraindo o cabeçalho IPv4 de 20 bytes e o cabeçalho TCP de 20 bytes, o tamanho padrão do MSS será 960 bytes, conforme mostrado na figura.

mostra um diagrama de um quadro Ethernet inteiro do qual o MTU é 1500 bytes, com 20 bytes sendo o cabeçalho IP, e 20 bytes sendo o cabeçalho TCP, isso deixa 960 bytes que é o tamanho máximo do segmento TCP MSS



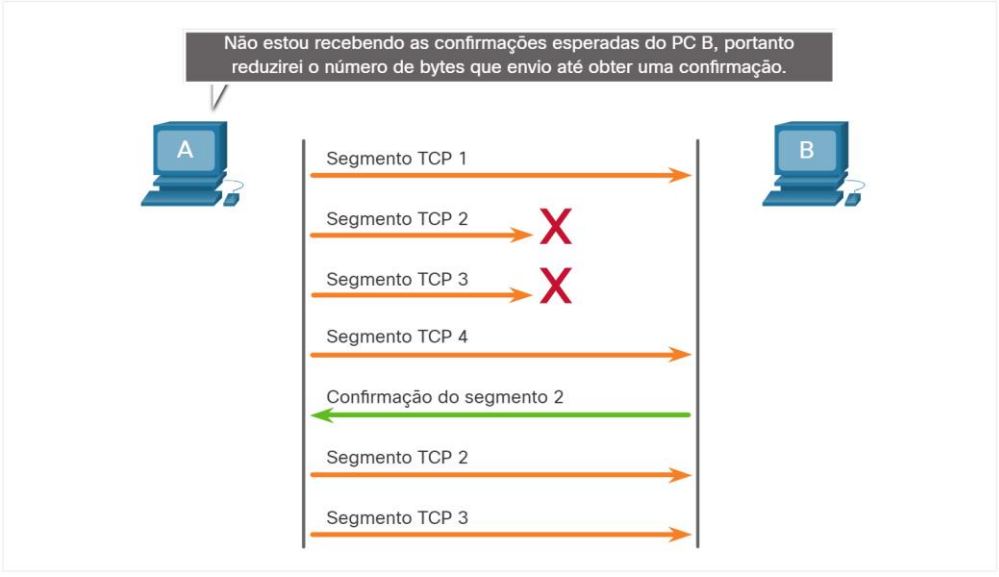
9.6.7 Controle de Fluxo TCP - Prevenção de Congestionamento

Quando ocorre um congestionamento em uma rede, isso resulta em pacotes sendo descartados pelo roteador sobrecarregado. Quando pacotes contendo segmentos TCP não atingem seu destino, eles são deixados sem serem reconhecidos. Ao determinar a taxa na qual os segmentos TCP são enviados, mas não confirmados, a origem pode pressupor um certo nível de congestionamento da rede.

Sempre que ocorrer um congestionamento, ocorrerá a retransmissão de segmentos TCP perdidos por parte da origem. Se a retransmissão não for devidamente controlada, a retransmissão adicional dos segmentos TCP pode agravar o congestionamento. Não só novos pacotes com segmentos TCP são introduzidos na rede, como também o efeito de feedback dos segmentos retransmitidos que foram perdidos aumentarão o congestionamento. Para evitar e controlar o congestionamento, o TCP emprega alguns mecanismos para lidar com o congestionamento, temporizadores e algoritmos.

Se a origem determina que os segmentos TCP não são confirmados ou não são confirmados em tempo hábil, isso pode reduzir o número de bytes enviados antes do recebimento de uma confirmação. Conforme ilustrado na figura, o PC A detecta que há congestionamento e, portanto, reduz o número de bytes que envia antes de receber uma confirmação do PC B.

mostra PCA enviando segmentos para PCB onde segmentos perdidos e retransmissão podem causar congestionamento. Controle de Congestionamento TCP



Os números de confirmação são para o próximo byte esperado e não para um segmento. Os números de segmento usados são simplificados para fins ilustrativos.

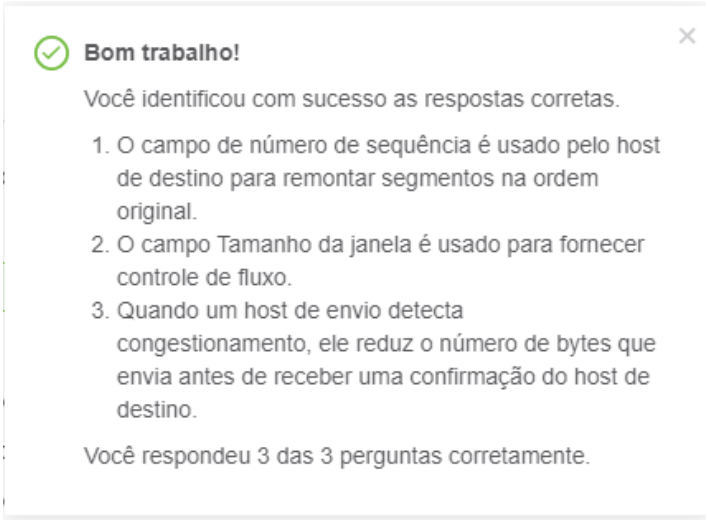
Observe que é a origem que está reduzindo o número de bytes não confirmados que envia e não o tamanho da janela determinado pelo destino.

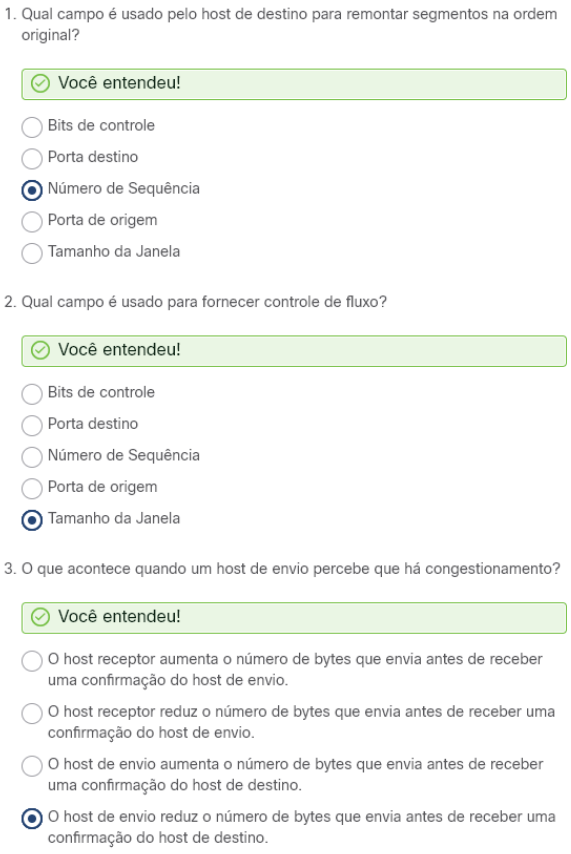
Nota: As explicações sobre os mecanismos, cronômetros e algoritmos reais de tratamento de congestionamento estão além do escopo deste curso.

9.6.8 Verifique sua compreensão - Confiabilidade e controle de fluxo

Verifique sua compreensão do processo de confiabilidade e controle de fluxo TCP, escolhendo a melhor resposta para as seguintes perguntas.

1. Qual campo é usado pelo host de destino para remontar segmentos na ordem original?
   - Bits de controle
   - Porta destino
   - Número de Sequência
   - Porta de origem
   - Tamanho da Janela
2. Qual campo é usado para fornecer controle de fluxo?
   - Bits de controle
   - Porta destino
   - Número de Sequência
   - Porta de origem
   - Tamanho da Janela
3. O que acontece quando um host de envio percebe que há congestionamento?
   - O host receptor aumenta o número de bytes que envia antes de receber uma confirmação do host de envio.
   - O host receptor reduz o número de bytes que envia antes de receber uma confirmação do host de envio.
   - O host de envio aumenta o número de bytes que envia antes de receber uma confirmação do host de destino.
   - O host de envio reduz o número de bytes que envia antes de receber uma confirmação do host de destino.

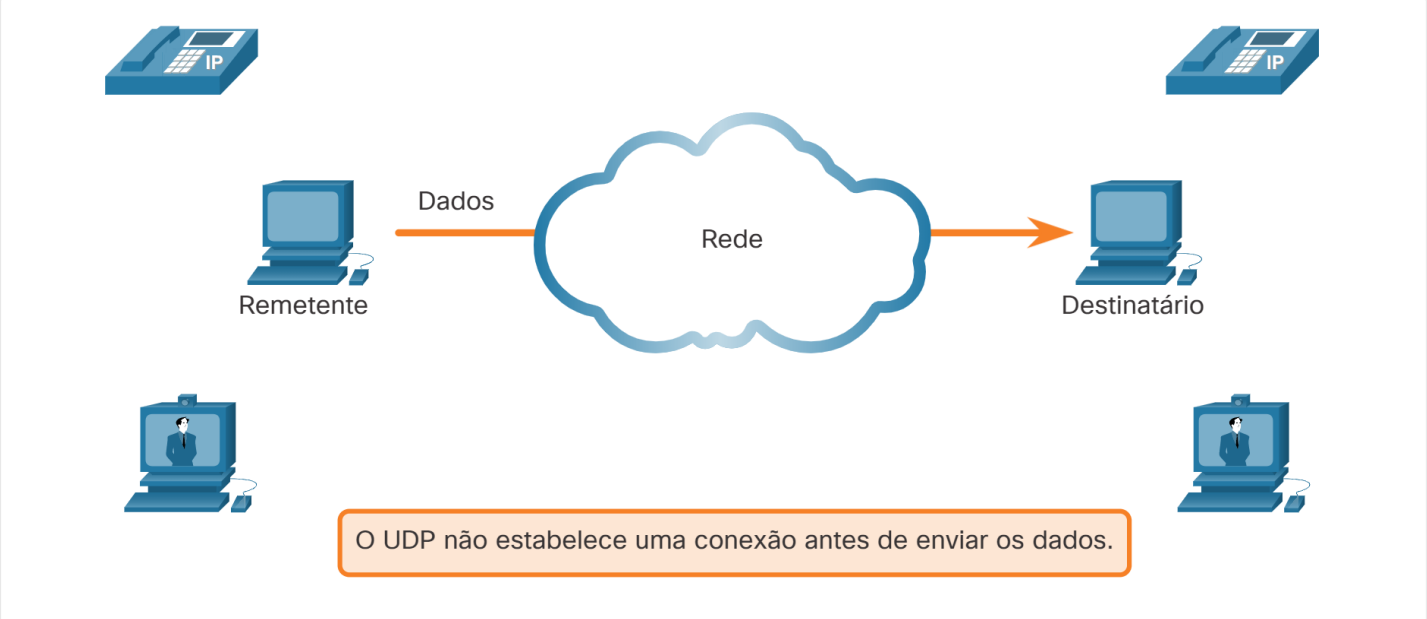




9.7 Comunicação UDP

9.7.1 Baixa Sobrecarga do UDP Versus Confiabilidade

Como explicado anteriormente, o UDP é perfeito para comunicações que precisam ser rápidas, como VoIP. Este tópico explica em detalhes por que o UDP é perfeito para alguns tipos de transmissões. Como mostrado na figura, o UDP não estabelece uma conexão. O UDP fornece transporte de dados de baixa sobrecarga, porque tem um cabeçalho de datagrama pequeno e nenhum tráfego de gerenciamento de rede.



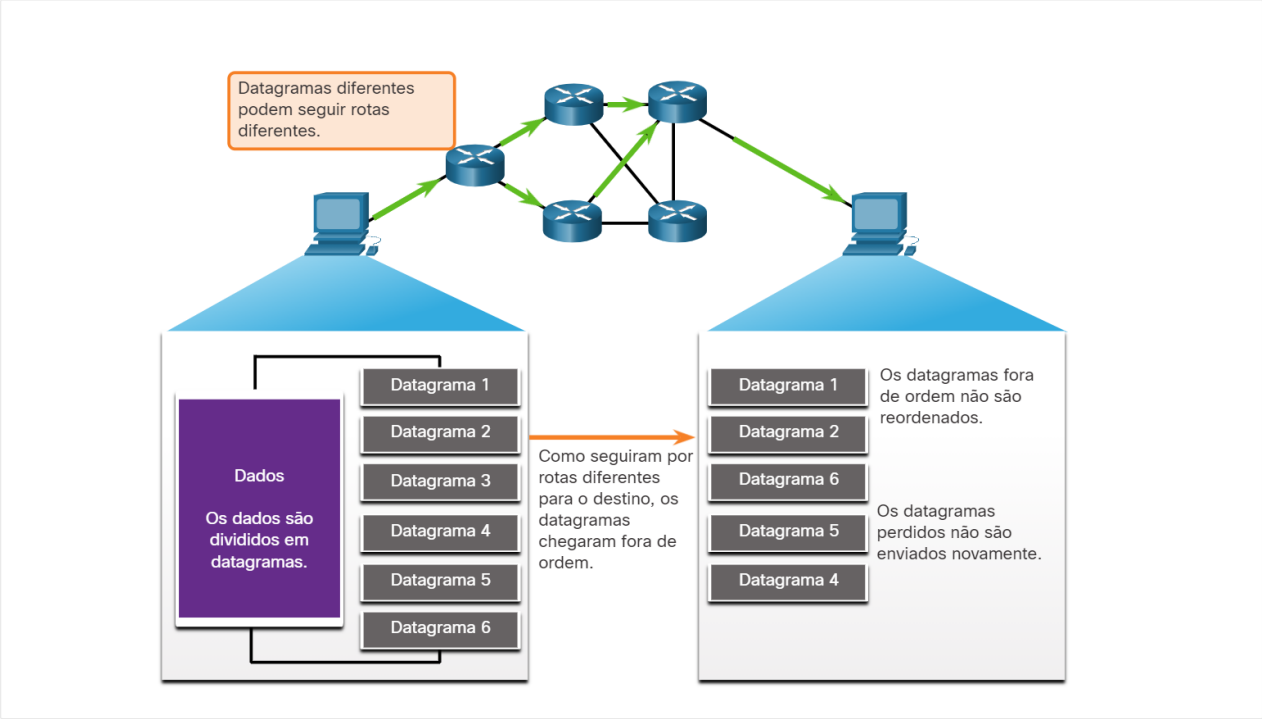
9.7.2 Remontagem do Datagrama UDP

Como ocorre com segmentos TCP, quando múltiplos datagramas UDP são enviados a um destino, eles geralmente tomam caminhos diferentes e chegam na ordem errada. O UDP não rastreia os números de sequência da forma que o TCP faz. O UDP não tem como reordernar os datagramas na sua ordem de transmissão, como mostrado na figura.

Portanto, o UDP simplesmente remonta os dados na ordem que eles foram recebidos e os encaminha para a aplicação. Se a sequência de dados for importante para a aplicação, a aplicação deverá identificar a sequência apropriada e determinar como os dados devem ser processados.

mostra datagramas UDP sendo enviados em ordem, mas chegando fora de ordem devido à possibilidade de diferentes rotas para chegar ao destino

UDP: Sem Conexão e Não Confiável

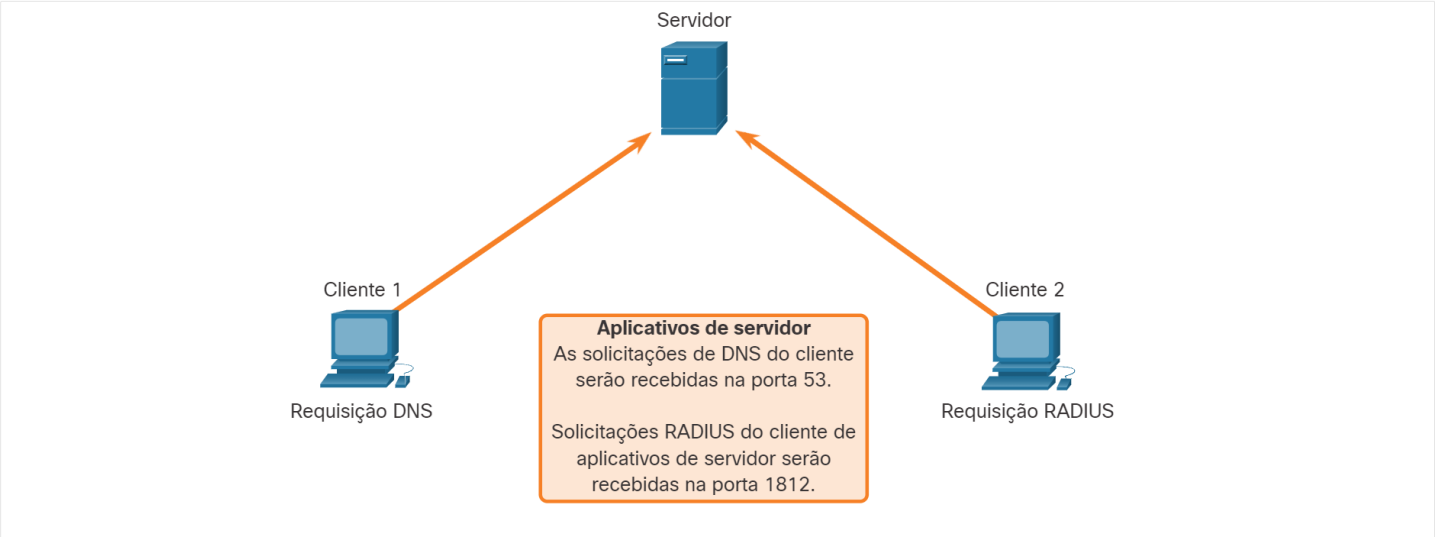


9.7.3 Processos em Servidores e Requisições UDP

Do mesmo modo que aplicações baseadas em TCP, as aplicações de servidor baseadas em UDP recebem números de portas bem conhecidas ou registradas, como mostrado na figura. Quando as aplicações ou processos estão sendo executados, eles aceitarão os dados correspondentes ao número de porta atribuído. Quando o UDP recebe um datagrama destinado a uma destas portas, ele encaminha os dados à aplicação apropriada com base em seu número de porta.

mostra que um aplicativo de servidor RADIUS usa UDP para escutar solicitações na porta 53

Servidor UDP Escutando Requisições As solicitações de DNS do cliente serão recebidas na porta 53. Solicitações RADIUS do cliente de aplicativos de servidor serão recebidas na porta 1812.Requisição DNSRequisição RADIUS



Note: O servidor RADIUS (Serviço de Usuário Discado por Autenticação Remota) mostrado na figura fornece serviços de autenticação, autorização e contabilidade para gerenciar o acesso do usuário. A operação do RADIUS está além do escopo deste curso.

9.7.4 Processos em Clientes UDP

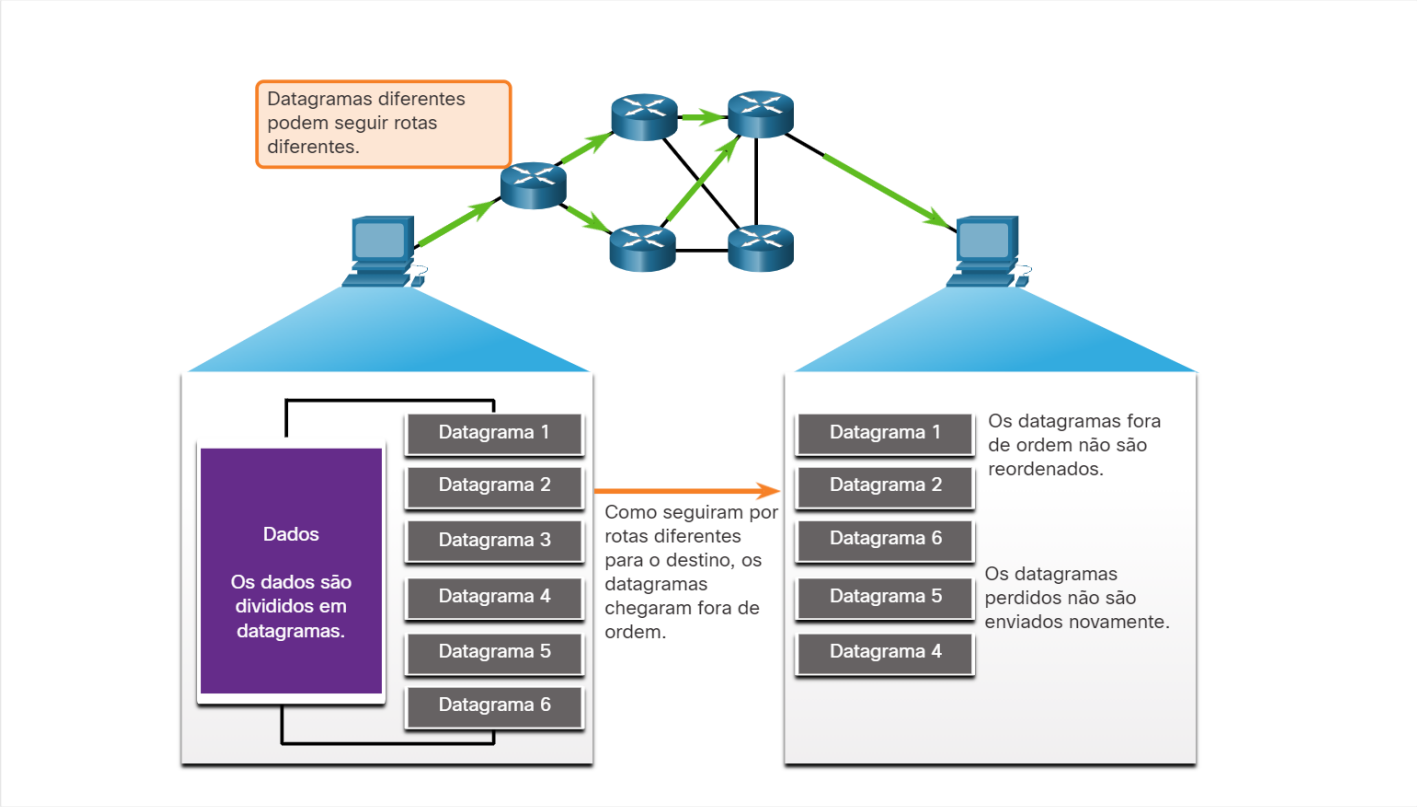
Assim como o TCP, a comunicação cliente servidor é iniciada por uma aplicação cliente que requisita dados de um processo em um servidor. O processo no cliente UDP seleciona dinamicamente um número de porta a partir de uma faixa de números de portas e a usa como a porta de origem para a conversa. A porta de destino será geralmente o número de porta muito conhecida ou registrada atribuído ao processo no servidor.

Depois que um cliente seleciona as portas de origem e de destino, o mesmo par de portas é usado no cabeçalho de todos os datagramas na transação. Para dados que retornam para o cliente vindos do servidor, os números da porta de origem e de destino no cabeçalho do datagrama são invertidos.

Clientes enviando solicitações UDP

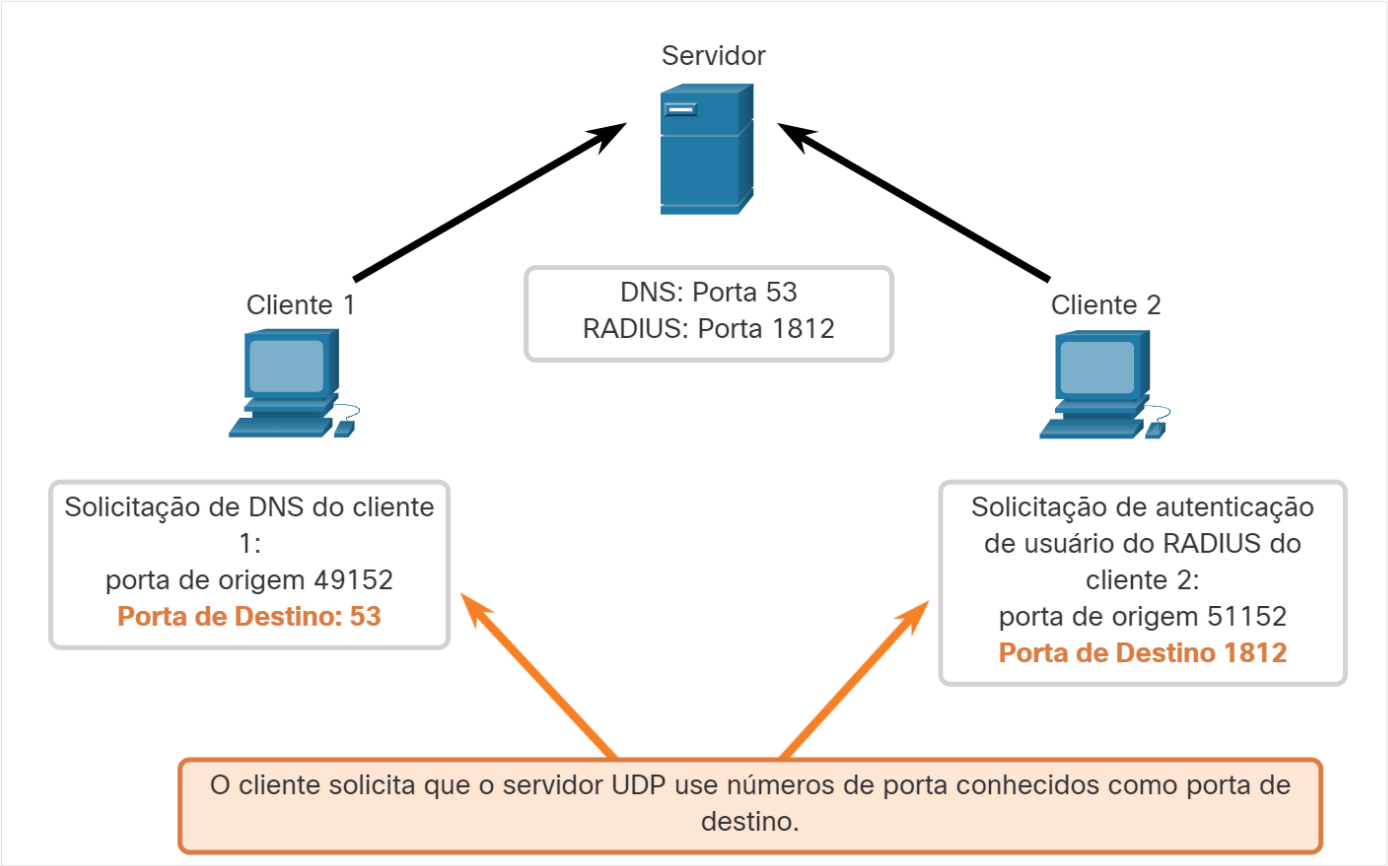
O Cliente 1 está enviando uma solicitação DNS usando a conhecida porta 53 enquanto o Cliente 2 está solicitando serviços de autenticação RADIUS usando a porta registrada 1812.

Dois clientes de PC diferentes precisam fazer uma solicitação para um servidor DNS



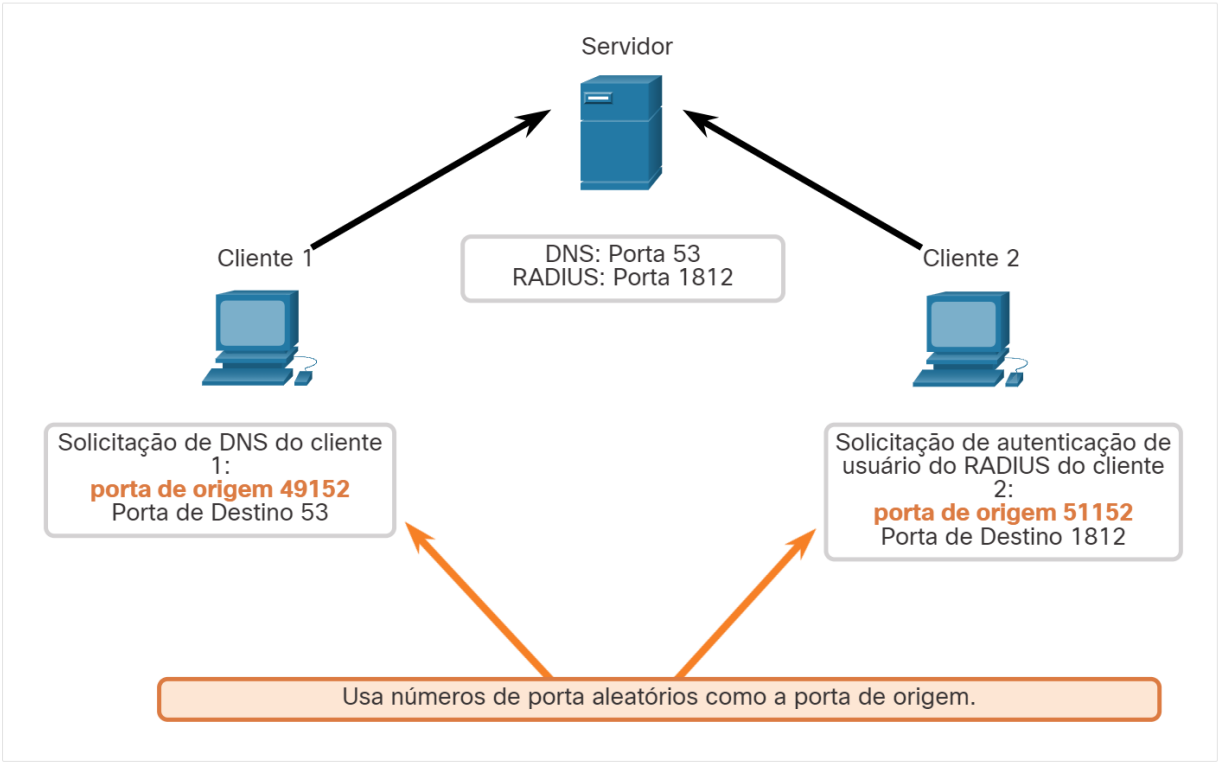
Portas de destino de solicitação UDP

As solicitações dos clientes geram dinamicamente números de porta de origem. Nesse caso, o cliente 1 está usando a porta de origem 49152 e o cliente 2 está usando a porta de origem 51152.



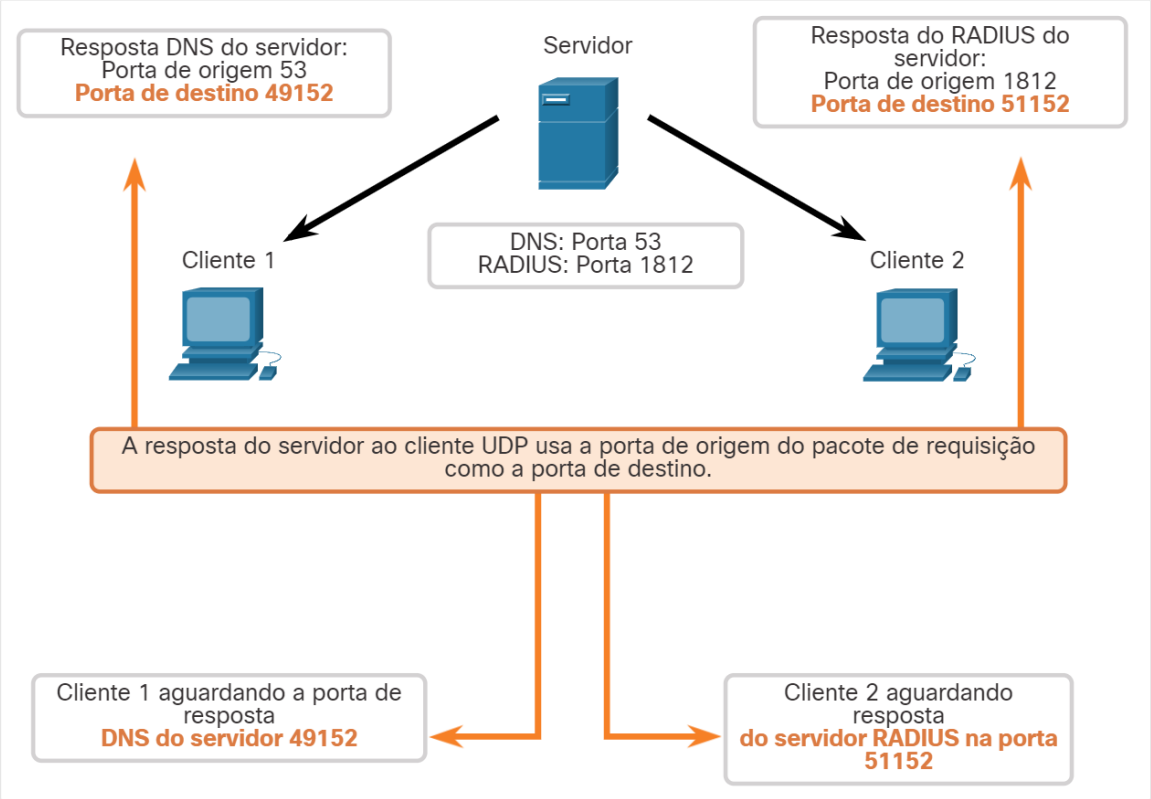
Portas de origem da solicitação UDP

Quando o servidor responde às solicitações do cliente, ele reverte as portas de destino e de origem da solicitação inicial.



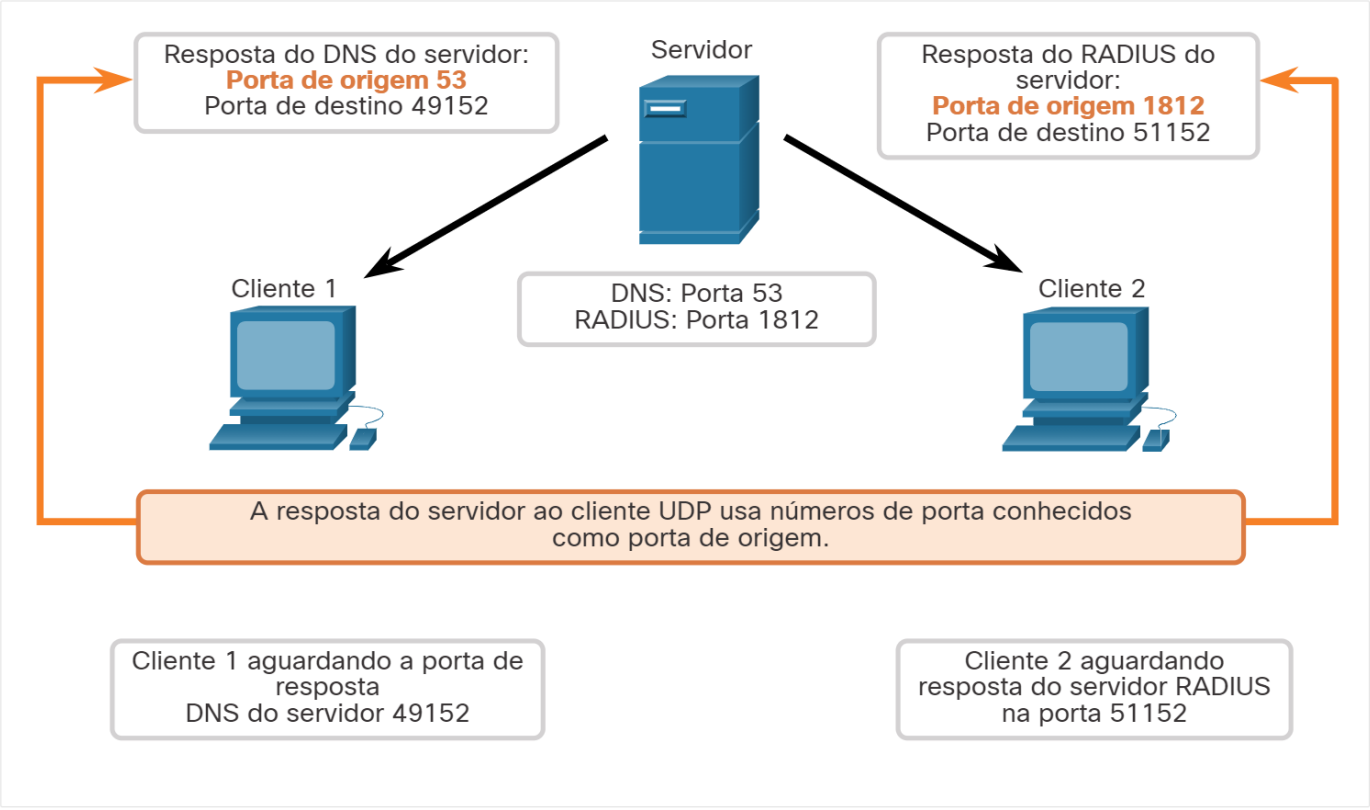
Destino de resposta UDP

Na resposta do servidor à solicitação DNS agora é a porta de destino 49152 e a resposta de autenticação RADIUS é agora a porta de destino 51152.



Portas de origem de resposta UDP

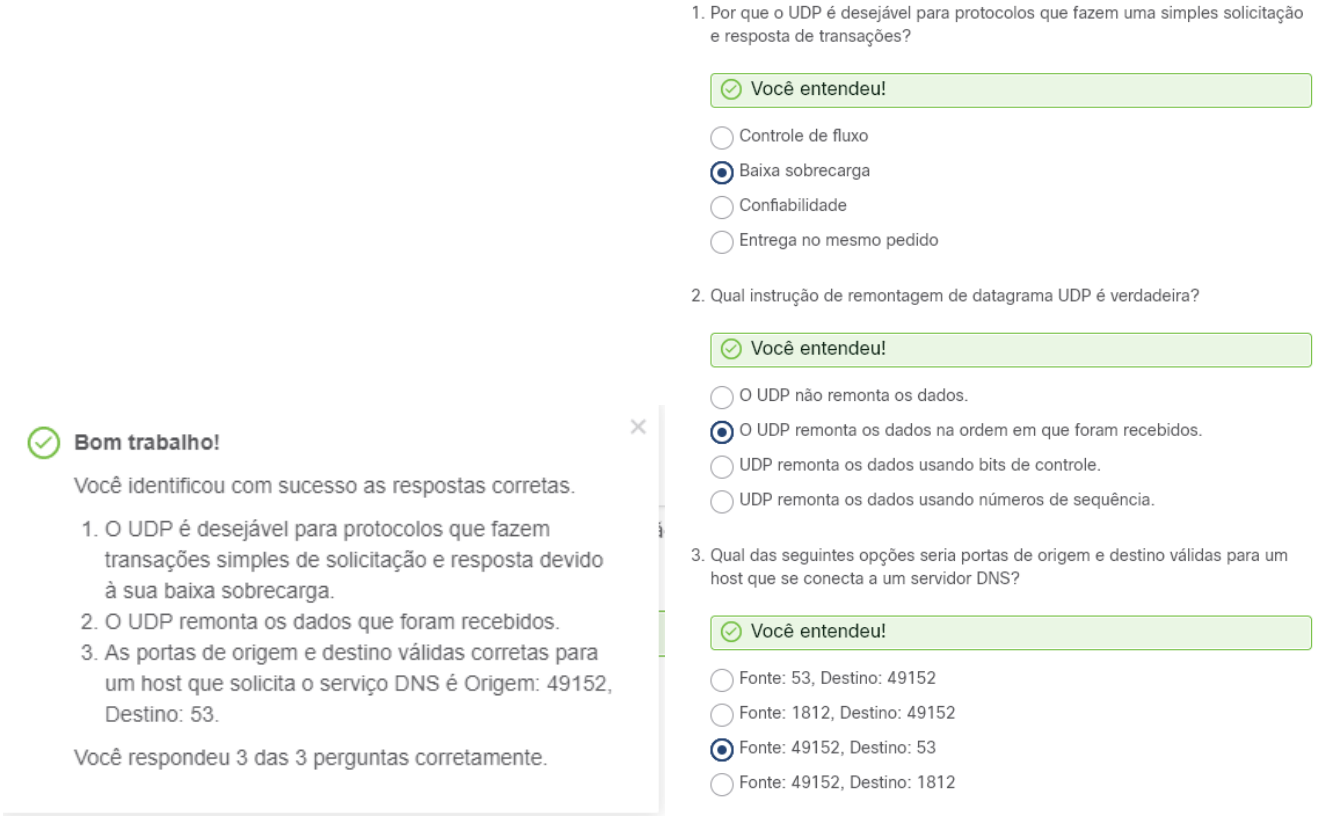
As portas de origem na resposta do servidor são as portas de destino originais nas solicitações iniciais.



9.7.5 Verifique o seu entendimento - Comunicação UDP

Verifique sua compreensão da comunicação UDP escolhendo a melhor resposta para as seguintes perguntas.

1. Por que o UDP é desejável para protocolos que fazem uma simples solicitação e resposta de transações?
   - Controle de fluxo
   - Baixa sobrecarga
   - Confiabilidade
   - Entrega no mesmo pedido
2. Qual instrução de remontagem de datagrama UDP é verdadeira?
   - O UDP não remonta os dados.
   - O UDP remonta os dados na ordem em que foram recebidos.
   - UDP remonta os dados usando bits de controle.
   - UDP remonta os dados usando números de sequência.
3. Qual das seguintes opções seria portas de origem e destino válidas para um host que se conecta a um servidor DNS?
   - Fonte: 53, Destino: 49152
   - Fonte: 1812, Destino: 49152
   - Fonte: 49152, Destino: 53
   - Fonte: 49152, Destino: 1812



9.8 Módulo Prática e Quiz

9.8.1 Packet Tracer: Comunicações TCP e UDP

Nesta atividade, você explorará a funcionalidade dos protocolos TCP e UDP, a multiplexação e a função dos números de porta para determinar qual aplicativo local solicitou os dados ou está enviando os dados.

Comunicações TCP e UDP

9.8.2 O que eu aprendi neste módulo?

Transporte de Dados

A camada de transporte é o link entre a camada de aplicativo e as camadas inferiores responsáveis pela transmissão da rede. A camada de transporte é responsável pela comunicação lógica entre aplicativos executados em hosts diferentes. A camada de transporte inclui TCP e UDP. Os protocolos de camada de transporte especificam como transferir mensagens entre hosts e é responsável por gerenciar os requisitos de confiabilidade de uma conversa. A camada de transporte é responsável por rastrear conversas (sessões), segmentar dados e remontar segmentos, adicionar informações de cabeçalho, identificar aplicativos e multiplexação de conversações. O TCP é stateful, confiável, reconhece dados, reenvia dados perdidos e entrega dados em ordem sequenciada. Utilize o TCP para correio electrónico e para a Web. O UDP é sem estado, rápido, tem baixa sobrecarga, não requer confirmações, não reenvia dados perdidos e fornece dados na ordem em que chegam. Use UDP para VoIP e DNS.

Visão geral do TCP

O TCP estabelece sessões, garante confiabilidade, fornece entrega de mesma ordem e oferece suporte ao controle de fluxo. Um segmento TCP adiciona 20 bytes de sobrecarga como informações de cabeçalho ao encapsular os dados da camada de aplicativo. Os campos do cabeçalho TCP são as portas de origem e destino, número de sequência, número de reconhecimento, comprimento do cabeçalho, reservado, bits de controle, tamanho da janela, soma de verificação e urgência. Os aplicativos que usam TCP são HTTP, FTP, SMTP e Telnet.

Visão Geral da UDP

O UDP reconstrói os dados na ordem em que são recebidos, os segmentos perdidos não são reenviados, nenhum estabelecimento de sessão e o UPD não informa o remetente da disponibilidade de recursos. Os campos do cabeçalho UDP são portas de origem e destino, comprimento e soma de verificação. Os aplicativos que usam UDP são DHCP, DNS, SNMP, TFTP, VoIP e videoconferência.

Números de porta

Os protocolos de camada de transporte TCP e UDP usam números de porta para gerenciar várias conversas simultâneas. É por isso que os campos de cabeçalho TCP e UDP identificam um número de porta de aplicativo de origem e destino. As portas origem e destino são colocadas no segmento. Os segmentos são encapsulados em um pacote IP. O pacote IP contém o endereço IP de origem e destino. A combinação do endereço IP de origem e o número de porta de origem, ou do endereço IP de destino e o número de porta de destino é conhecida como um socket. O socket é usado para identificar o servidor e o serviço que está sendo solicitado pelo cliente. Há uma gama de números de portas de 0 a 65535. Este intervalo é dividido em grupos: Portas bem conhecidas, Portas Registradas, Portas Privadas e/ou Dinâmicas. Existem alguns números de porta conhecidos que são reservados para aplicativos comuns, como FTP, SSH, DNS, HTTP e outros. Às vezes é necessário conhecer quais conexões TCP ativas estão abertas e sendo executadas em um host de rede. O netstat é um utilitário de rede importante que pode ser usado para verificar essas conexões.

Processo de comunicação TCP

Cada processo de aplicativo em execução em um servidor está configurado para usar um número de porta. O número da porta é atribuído automaticamente ou configurado manualmente por um administrador do sistema. Os processos do servidor TCP são os seguintes: clientes enviando solicitações TCP, solicitando portas de destino, solicitando portas de origem, respondendo a solicitações de porta de destino e porta de origem. Para terminar uma conversação única permitida pelo TCP, quatro trocas são necessárias para finalizar ambas as sessões. O cliente ou o servidor podem iniciar o encerramento. O handshake de três vias estabelece que o dispositivo de destino está presente na rede, verifica se o dispositivo de destino tem um serviço ativo e está aceitando solicitações no número da porta de destino que o cliente iniciante pretende usar e informa o dispositivo de destino que o cliente de origem pretende estabelecer uma sessão de comunicação sobre esse número de porta. Os seis sinais de bits de controle são: URG, ACK, PSH, RST, SYN e FIN.

Confiabilidade e controle de fluxo

Para que a mensagem original seja entendida pelo destinatário, todos os dados devem ser recebidos e os dados nesses segmentos devem ser remontados na ordem original. Os números de sequência são atribuídos no cabeçalho de

cada pacote. Não importa o quão bem projetada uma rede é, a perda de dados ocasionalmente ocorre. O TCP fornece maneiras de gerenciar perdas de segmento. Existe um mecanismo para retransmitir segmentos para dados não reconhecidos. Hoje em dia, os sistemas operacionais de host utilizam um recurso TCP opcional chamado reconhecimento seletivo (SACK), negociado durante o handshake de três vias. Se ambos os hosts suportarem SACK, o receptor pode reconhecer explicitamente quais segmentos (bytes) foram recebidos, incluindo quaisquer segmentos descontínuos. O host de envio, portanto, só precisa retransmitir os dados ausentes. O controle de fluxo ajuda a manter a confiabilidade da transmissão TCP, ajustando a taxa de fluxo de dados entre a origem e o destino. Para realizar isso, o cabeçalho TCP inclui um campo de 16 bits chamado de tamanho da janela. O processo de envio de confirmações pelo destino enquanto processa os bytes recebidos, e o ajuste contínuo da janela de envio da origem é conhecido como janelas deslizantes. Uma fonte pode estar transmitindo 1.460 bytes de dados dentro de cada segmento TCP. Este é o MSS típico que um dispositivo de destino pode receber. Para evitar e controlar o congestionamento, o TCP emprega vários mecanismos de manipulação de congestionamento. É a fonte que está reduzindo o número de bytes não reconhecidos enviados e não o tamanho da janela determinado pelo destino.

Comunicação UDP

O UDP é um protocolo simples que fornece as funções básicas da camada de transporte. Quando os datagramas UDP são enviados para um destino, eles geralmente seguem caminhos diferentes e chegam na ordem errada. O UDP não rastreia os números de sequência da mesma maneira que o TCP. O UDP não tem um meio de reordenar os datagramas em sua ordem de transmissão. O UDP simplesmente remonta os dados na ordem em que foram recebidos e os encaminha para o aplicativo. Se a sequência de dados for importante para a aplicação, a aplicação deverá identificar a sequência apropriada e determinar como os dados devem ser processados. Os aplicativos de servidor baseados em UDP recebem números de porta conhecidos ou registrados. Quando o UDP recebe um datagrama destinado a uma destas portas, ele encaminha os dados à aplicação apropriada com base em seu número de porta. O processo no cliente UDP seleciona dinamicamente um número de porta a partir de uma faixa de números de portas e a usa como a porta de origem para a conversa. A porta de destino será geralmente o número de porta muito conhecida ou registrada atribuído ao processo no servidor. Depois que um cliente seleciona as portas de origem e destino, o mesmo par de portas é usado no cabeçalho de todos os datagramas usados na transação. Para dados que retornam para o cliente vindos do servidor, os números da porta de origem e de destino no cabeçalho do datagrama são invertidos.

9.8.3 Questionário do Módulo - Camada de Transporte

1. Qual feição da camada de transporte é usada para estabelecer uma sessão orientada a conexão?
   - Indicador ACK UDP
   - Número da porta TCP
   - Número de sequência UDP
   - Handshake de 3 vias TCP
2. Qual é a gama completa de portas TCP e UDP bem conhecidas?
   - 256 - 1023
   - 0 a 255
   - 0 a 1023
   - 1.024 a 49.151
3. O que é um soquete?
   - A combinação da sequência de origem e de destino e dos números de confirmação
   - A combinação do endereço IP de origem e destino e endereço Ethernet de origem e destino
   - A combinação dos números de sequência de origem e de destino e dos números de porta
   - A combinação de um endereço IP de origem e número de porta ou um endereço IP de destino e número de porta
4. Como um servidor em rede gerencia solicitações de vários clientes para serviços diferentes?
   - O servidor envia todas as solicitações por meio de um gateway padrão.

- Cada solicitação é rastreada através do endereço físico do cliente.
- O servidor usa endereços IP para identificar serviços diferentes.
- Cada solicitação possui uma combinação de números de porta de origem e destino, provenientes de um endereço IP exclusivo.

5. O que acontece se parte de uma mensagem FTP não for entregue ao destino?
   - O host de origem FTP envia uma consulta para o host de destino.
   - Toda a mensagem FTP é reenviada.
   - A parte da mensagem FTP que foi perdida é reenviada.
   - A mensagem é perdida porque o FTP não usa um método de entrega confiável.
6. Que tipo de aplicações são mais adequadas para usar o protocolo UDP?
   - aplicações que são sensíveis a atrasos
   - aplicações que precisam de retransmissão de segmentos perdidos
   - aplicações que são sensíveis a perda de pacotes
   - aplicações que precisam de entrega confiável
7. Com o congestionamento da rede, a origem soube da perda de segmentos TCP que foram enviados ao destino. Qual é uma maneira do protocolo TCP lidar com isso?
   - A origem diminui o volume de dados que pode ser transmitido antes de receber uma confirmação do destino.
   - O destino envia menos mensagens de confirmação a fim de preservar a largura de banda.
   - O destino diminui o tamanho da janela.
   - A origem diminui o tamanho da janela para reduzir a taxa de transmissão do destino.
8. Quais duas operações são fornecidas pelo TCP, mas não pelo UDP? (Escolha duas.)
   - ☐ reconhecendo dados recebidos
   - ☐ identificando os aplicativos
   - ☐ reconstruindo dados na ordem recebida
   - ☐ Identificar de conversas individuais
   - ☐ retransmissão de quaisquer dados não reconhecidos
9. Qual é o propósito de usar um número de porta de origem em uma comunicação TCP?
   - Para notificar o dispositivo remoto de que a conversa acabou
   - Para manter o controle de várias conversas entre dispositivos
   - Para montar os segmentos que chegaram fora de ordem
   - Para pesquisar um segmento não recebido
10. Quais dois sinalizadores no cabeçalho TCP são usados em um handshake de três vias TCP para estabelecer conectividade entre dois dispositivos de rede? (Escolha duas.)
    - ☐ ACK
    - ☐ PSH
    - ☐ SYN
    - ☐ RST
    - ☐ URG
    - ☐ FIN
11. Qual mecanismo TCP é usado para melhorar o desempenho, permitindo que um dispositivo envie continuamente um fluxo contínuo de segmentos, desde que o dispositivo também esteja recebendo as confirmações necessárias?
    - pares de sockets
    - handshake duplo
    - Janelas deslizantes
    - handshake triplo
12. Qual ação é executada por um cliente ao estabelecer comunicação com um servidor através do uso de UDP na camada de transporte?
    - O cliente envia um segmento de sincronização para iniciar a sessão.
    - O cliente envia um ISN para o servidor para iniciar o handshake de 3 vias.

- O cliente seleciona aleatoriamente um número de porta de origem.
- O cliente define o tamanho da janela para a sessão.

13. Que dois serviços ou protocolos preferem usar o protocolo UDP para agilizar a transmissão e reduzir a sobrecarga? (Escolha duas)

- ☐ POP3
- ☐ DNS
- ☐ VoIP
- ☐ HTTP
- ☐ FTP

9. Qual número ou conjunto de números representa um soquete?

- 01-23-45-67-89-AB
- 192.168.1. 1:80
- 10.1.1.15
- 21

15. O que é uma responsabilidade dos protocolos de camada de transporte?

- traduzindo endereços IP privados em endereços IP públicos
- rastreamento de conversas individuais
- determinando o melhor caminho para encaminhar um pacote
- fornecendo acesso à rede

1. Qual função da camada de transporte é usada para estabelecer uma sessão orientada a conexão?

Tópico 14.5.0 - TCP usa o handshake de 3 vias. UDP não usa este recurso. O handshake de 3 vias garante que haja conectividade entre os dispositivos de origem e destino antes da transmissão ocorrer.

- ( ) Número da porta TCP
- ( ) Indicador ACK UDP
- ( ) Número de sequência UDP
- (•) Handshake de 3 vias TCP

2. Qual é a gama completa de portas TCP e UDP bem conhecidas?

Tópico 14.4.0 - Existem três intervalos de portas TCP e UDP. A gama bem conhecida de números de porta é de 0 a 1023.

- ( ) 0 a 255
- (•) 0 a 1023
- ( ) 1.024 a 49.151
- ( ) 256 - 1023

3. O que é um soquete?

Tópico 14.4.0 - Um soquete é uma combinação do endereço IP de origem e porta de origem ou o endereço IP de destino e o número da porta de destino.

- ( ) A combinação dos números de sequência de origem e de destino e dos números de porta
- (•) A combinação de um endereço IP de origem e número de porta ou um endereço IP de destino e número de porta
- ( ) A combinação da sequência de origem e de destino e dos números de confirmação
- ( ) A combinação do endereço IP de origem e destino e endereço Ethernet de origem e destino

4. Como um servidor em rede gerencia solicitações de vários clientes para serviços diferentes?

Tópico 14.4.0 - Cada serviço fornecido por um servidor, como transferências de e-mail ou arquivos, usa um número de porta específico. O número da porta de origem de uma solicitação de serviço identifica o cliente que está solicitando serviços. O número da porta de destino identifica o serviço específico. Os servidores não usam informações de endereço para fornecer serviços. Os roteadores e switches usam informações de endereçamento para mover o tráfego pela rede.

- ( ) O servidor usa endereços IP para identificar serviços diferentes.
- ( ) O servidor envia todas as solicitações por meio de um gateway padrão.
- ( ) Cada solicitação é rastreada através do endereço físico do cliente.
- (•) Cada solicitação possui uma combinação de números de porta de origem e destino, provenientes de um endereço IP exclusivo.

5. O que acontece se parte de uma mensagem FTP não for entregue ao destino?

Tópico 14.6.0 - Como o FTP usa o TCP como protocolo de camada de transporte, os números de sequência e confirmação identificam os segmentos ausentes, que serão reenviados para concluir a mensagem.

- ( ) A mensagem é perdida porque o FTP não usa um método de entrega confiável.
- ( ) O host de origem FTP envia uma consulta para o host de destino.
- ( ) Toda a mensagem FTP é reenviada.
- (•) A parte da mensagem FTP que foi perdida é reenviada.

6. Que tipo de aplicações são mais adequadas para usar o protocolo UDP?

Tópico 14.3.0 - O UDP não é um protocolo orientado a conexão e não fornece mecanismos de retransmissão, sequenciamento ou controle de fluxo. Ele oferece funções básicas da camada de transporte com uma sobrecarga muito inferior à do TCP. Uma sobrecarga inferior faz com que o protocolo UDP seja adequado para as aplicações sensíveis a atraso.

- ( ) aplicações que precisam de retransmissão de segmentos perdidos
- (•) aplicações que são sensíveis a atrasos
- ( ) aplicações que precisam de entrega confiável
- ( ) aplicações que são sensíveis a perda de pacotes

7. Com o congestionamento da rede, a origem soube da perda de segmentos TCP que foram enviados ao destino. Qual é uma maneira do protocolo TCP lidar com isso?

Tópico 14.6.0 - Se a fonte determinar que os segmentos TCP não estão sendo reconhecidos ou não são reconhecidos em tempo hábil, é possível reduzir o número de bytes que envia antes de receber um reconhecimento. Isso não envolve alterar a janela do cabeçalho do segmento. A origem não reduz a janela enviada no cabeçalho do segmento. A janela no cabeçalho do segmento é ajustada pelo host destino quando ele recebe dados mais rapidamente do que é capaz de processar, e não quando se depara com o congestionamento de rede.

- ( ) O destino envia menos mensagens de confirmação a fim de preservar a largura de banda.
- ( ) O destino diminui o tamanho da janela.
- (•) A origem diminui o volume de dados que pode ser transmitido antes de receber uma confirmação do destino.
- ( ) A origem diminui o tamanho da janela para reduzir a taxa de transmissão do destino.

8. Quais duas operações são fornecidas pelo TCP, mas não pelo UDP? (Escolha duas.)

Tópico 14.1.0 - Numeração e rastreamento de segmentos de dados, confirmação de dados recebidos e retransmissão de quaisquer dados não reconhecidos são operações de confiabilidade para garantir que todos os dados cheguem ao destino. O UDP não fornece confiabilidade. O TCP e o UDP identificam os aplicativos e acompanham conversas individuais. O UDP não numera segmentos de dados e reconstrói dados na ordem em que é recebido.

- [ ] reconstruindo dados na ordem recebida
- [x] reconhecendo dados recebidos
- [ ] identificando os aplicativos
- [ ] Identificar de conversas individuais
- [x] retransmissão de quaisquer dados não reconhecidos

9. Qual é o propósito de usar um número de porta de origem em uma comunicação TCP?

Tópico 14.4.0 - O número da porta de origem em um cabeçalho de segmento é usado para manter o controle de várias conversas entre dispositivos. Ele também é usado para manter uma entrada aberta para a resposta do servidor. As opções incorretas estão mais relacionadas ao controle de fluxo e entrega garantida.

- ( ) Para notificar o dispositivo remoto de que a conversa acabou
- (•) Para manter o controle de várias conversas entre dispositivos
- ( ) Para pesquisar um segmento não recebido
- ( ) Para montar os segmentos que chegaram fora de ordem

10. Quais dois sinalizadores no cabeçalho TCP são usados em um handshake de três vias TCP para estabelecer conectividade entre dois dispositivos de rede? (Escolha duas.)

Tópico 14.5.0 - TCP usa os sinalizadores SYN e ACK para estabelecer conectividade entre dois dispositivos de rede.

- [x] SYN
- [x] ACK
- [ ] PSH
- [ ] FIN
- [ ] URG
- [ ] RST

11. Qual mecanismo TCP é usado para melhorar o desempenho, permitindo que um dispositivo envie continuamente um fluxo contínuo de segmentos, desde que o dispositivo também esteja recebendo as confirmações necessárias?

Tópico 14.6.0 - O TCP usa janelas para tentar gerenciar a taxa de transmissão até o fluxo máximo que a rede e o dispositivo de destino podem suportar, minimizando perdas e retransmissões. O destino pode enviar uma requisição de redução da janela, quando sobrecarregado com dados. O processo de envio de confirmações pelo destino enquanto processa os bytes recebidos, e o ajuste contínuo da janela de envio da origem é conhecido como janelas deslizantes.

- (•) Janelas deslizantes
- ( ) handshake triplo
- ( ) pares de sockets
- ( ) handshake duplo

12. Qual ação é executada por um cliente ao estabelecer comunicação com um servidor através do uso de UDP na camada de transporte?

Tópico 14.7.0 - Como uma sessão não precisa ser estabelecida para UDP, o cliente seleciona uma porta de origem aleatória para iniciar uma conexão. O número de porta aleatória selecionado é inserido no campo de porta de origem do cabeçalho UDP.

- (•) O cliente seleciona aleatoriamente um número de porta de origem.
- ( ) O cliente define o tamanho da janela para a sessão.
- ( ) O cliente envia um segmento de sincronização para iniciar a sessão.
- ( ) O cliente envia um ISN para o servidor para iniciar o handshake de 3 vias.

13. Que dois serviços ou protocolos preferem usar o protocolo UDP para agilizar a transmissão e reduzir a sobrecarga? (Escolha duas)

Tópico 14.3.0 - BTanto o DNS quanto o VoIP usam UDP para fornecer serviços de baixa sobrecarga em uma implementação de rede.

- [x] VoIP
- [ ] FTP
- [ ] POP3
- [ ] HTTP
- [x] DNS

14. Qual número ou conjunto de números representa um soquete?

Tópico 14.4.0 - Um soquete é definido pela combinação de um endereço IP e um número de porta, e identifica exclusivamente uma comunicação específica.

- ( ) 01-23-45-67-89-AB
- ( ) 10.1.1.15
- (•) 192.168.1. 1:80
- ( ) 21

15. O que é uma responsabilidade dos protocolos de camada de transporte?

Tópico 14.1.0 - Existem três responsabilidades principais para protocolos de camada de transporte TCP e UDP:
- Acompanhamento de conversas individuais
- Segmentando dados e remontando segmentos
- Identificando os aplicativos

- fornecendo acesso à rede
- rastreamento de conversas individuais
- traduzindo endereços IP privados em endereços IP públicos
- determinando o melhor caminho para encaminhar um pacote