



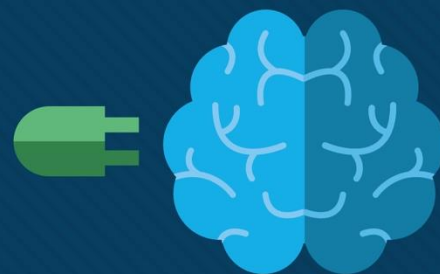
Módulo 9: A Camada de Transporte



CyberOps Associate v1.0

Prof. Clemilson Oliveira

clemilson.oliveira@edu.sc.senai.br



Objetivos do módulo

Título do Módulo: A Camada de Transporte

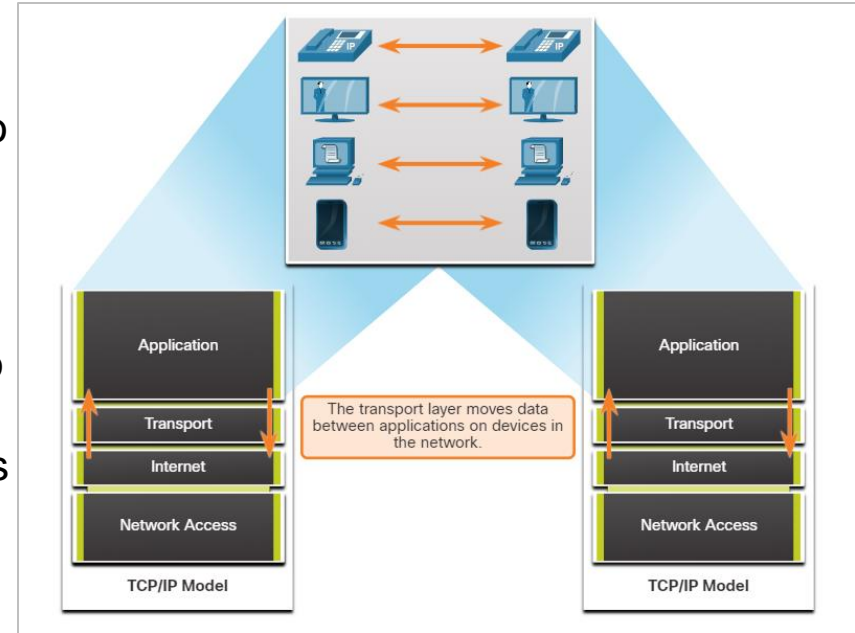
Objetivo do Módulo: Explique como os protocolos da camada de transporte oferecem suporte à funcionalidade de rede.

Título do Tópico	Objetivo do Tópico
Características da Camada de Transporte	Explicar como os protocolos da camada de transporte oferecem suporte à comunicação de rede.
Estabelecimento da Sessão da Camada de Transporte	Explicar como a camada de transporte estabelece sessões de comunicação.
Confiabilidade da Camada de Transporte	Explicar como a camada de transporte estabelece comunicações confiáveis.

9.1 Características da camada de transporte

Papel da Camada de Transporte

- A camada de transporte é responsável pela comunicação lógica entre aplicativos executados em hosts diferentes.
- 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 viajar, o caminho percorrido pelos dados, o congestionamento em um link ou o tamanho da rede.
- A camada de transporte inclui dois protocolos, Transmission Control Protocol (TCP) e User Datagram Protocol (UDP).

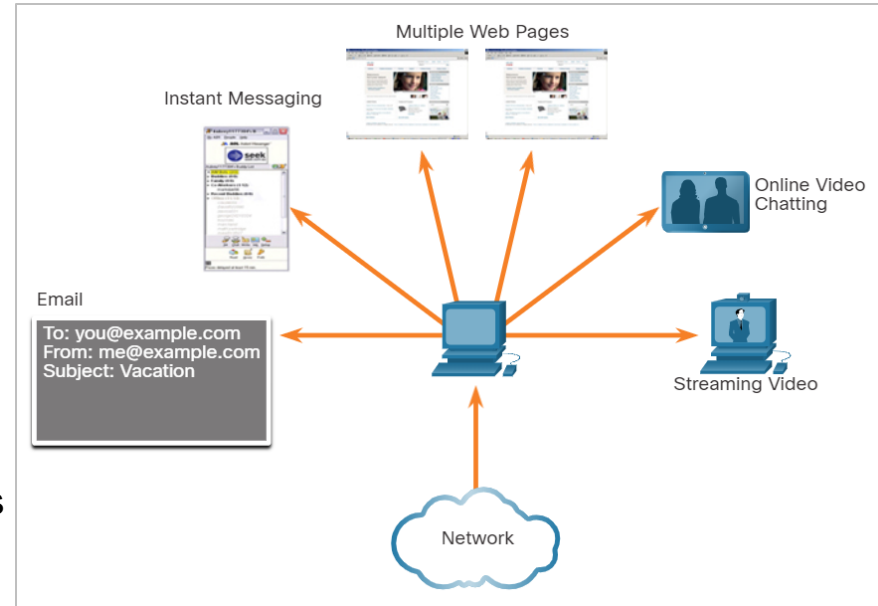


Responsabilidades da Camada de Transporte

A camada de transporte tem muitas responsabilidades.

Rastreamento de Conversações Individuais

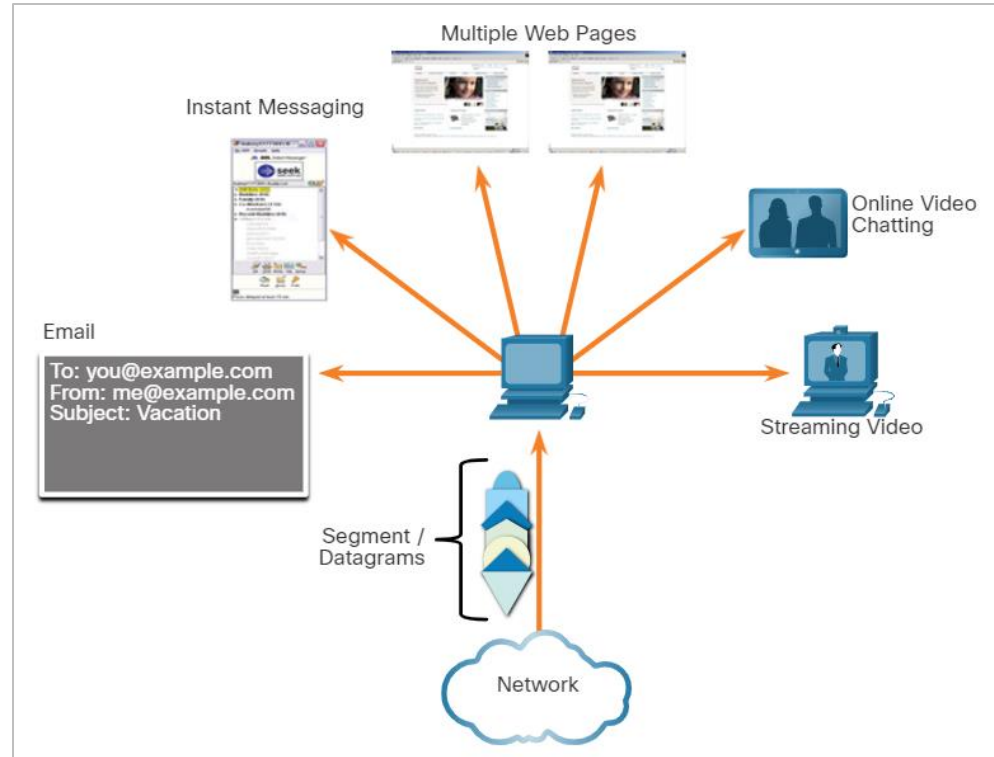
- Cada conjunto de dados fluindo entre um aplicativo de origem e um aplicativo de destino é conhecido como uma conversa e é rastreado separadamente.
- É responsabilidade da camada de transporte manter e monitorar essas várias conversações.
- Conforme mostrado na figura, um host pode ter vários aplicativos que se comunicam pela rede simultaneamente.
- A maioria das redes tem uma limitação da quantidade de dados que pode ser incluída em um único pacote. Os dados devem ser divididos em partes gerenciáveis.



Responsabilidades da Camada de Transporte (cont.)

Segmentação de Dados e Remontagem de Segmentos

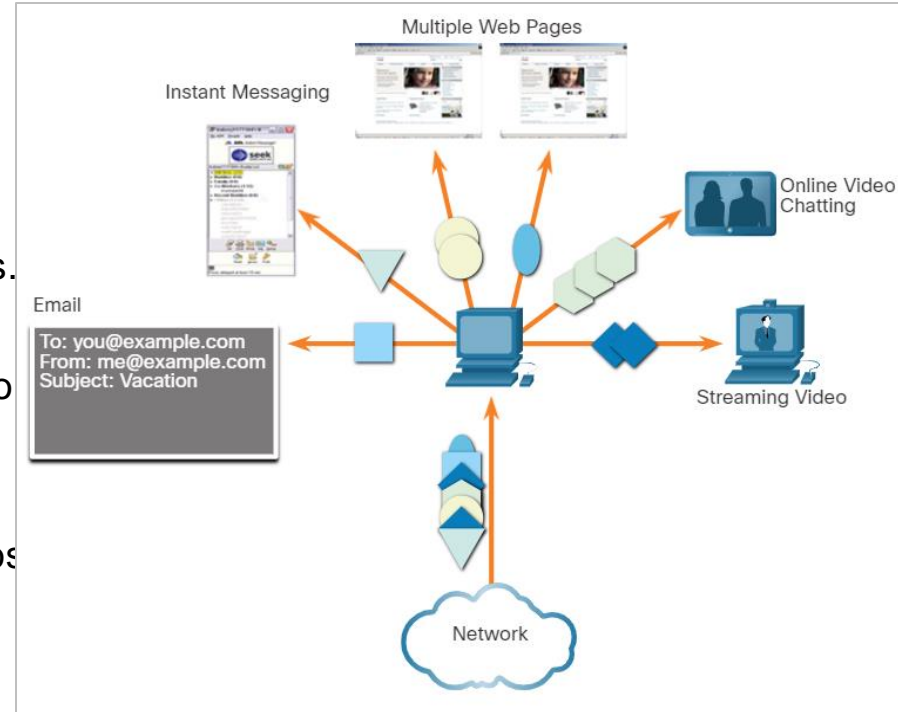
- É responsabilidade da camada de transporte dividir os dados do aplicativo em blocos de tamanho adequado.
- Dependendo do protocolo de camada de transporte usado, os blocos de camada de transporte são chamados de segmentos ou datagramas.
- A figura mostra a camada de transporte usando blocos diferentes para cada conversa.
- A camada de transporte divide os dados em blocos menores (segmentos ou datagramas) que são mais fáceis de gerenciar e transportar.



Responsabilidades da Camada de Transporte (cont.)

Adicionar Informações de Cabeçalho

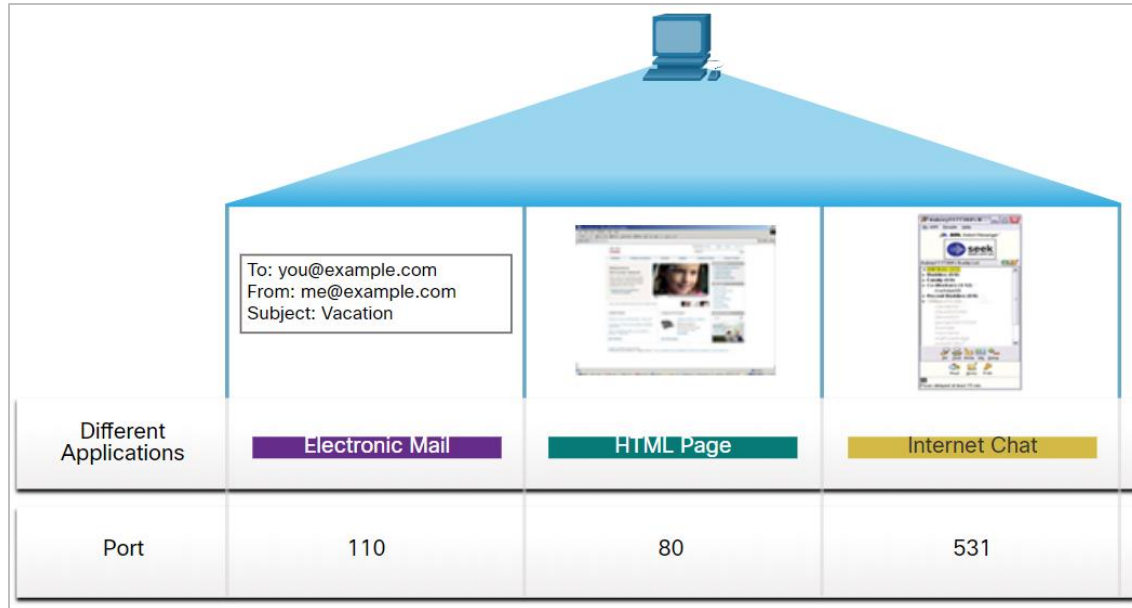
- O protocolo da camada de transporte também adiciona informações de cabeçalho contendo dados binários organizados em vários campos a cada bloco de dados.
- Os valores nesses campos permitem que vários protocolos da camada de transporte executem funções diferentes no gerenciamento da comunicação de dados.
- As informações do cabeçalho são usadas pelo host receptor para remontar os blocos de dados em um fluxo de dados completo para o programa da camada de aplicativo receptor.
- A camada de transporte garante que, mesmo com vários aplicativos em execução em um dispositivo, todos os aplicativos recebam os dados corretos.



Responsabilidades da Camada de Transporte (cont.)

Identificação das Aplicações

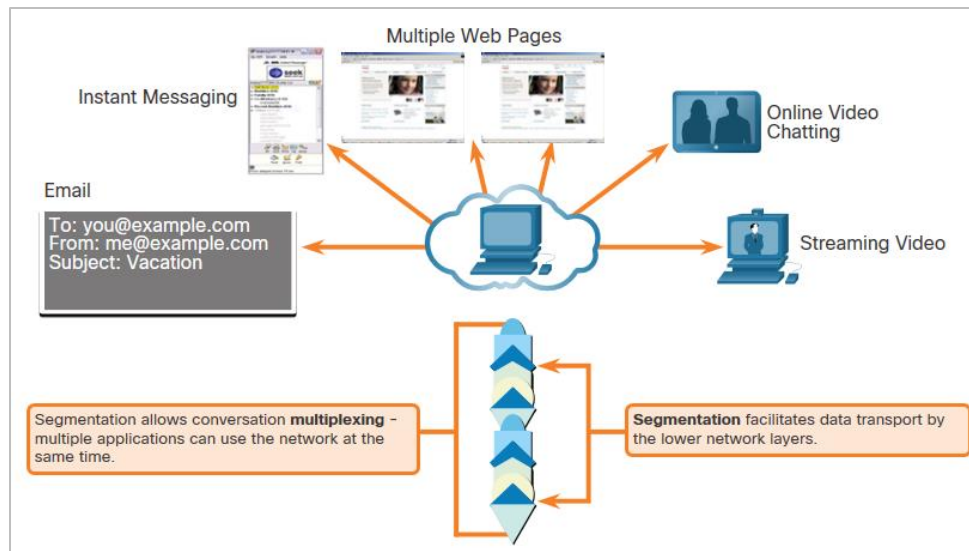
- A camada de transporte deve separar e gerenciar várias comunicações com as diferentes necessidades de requisitos de transporte.
- Para passar fluxos de dados para os aplicativos adequados, a camada de transporte identifica o aplicativo de destino usando um identificador chamado número da porta.
- Conforme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



Responsabilidades da Camada de Transporte (cont.)

Multiplexação das Conversas

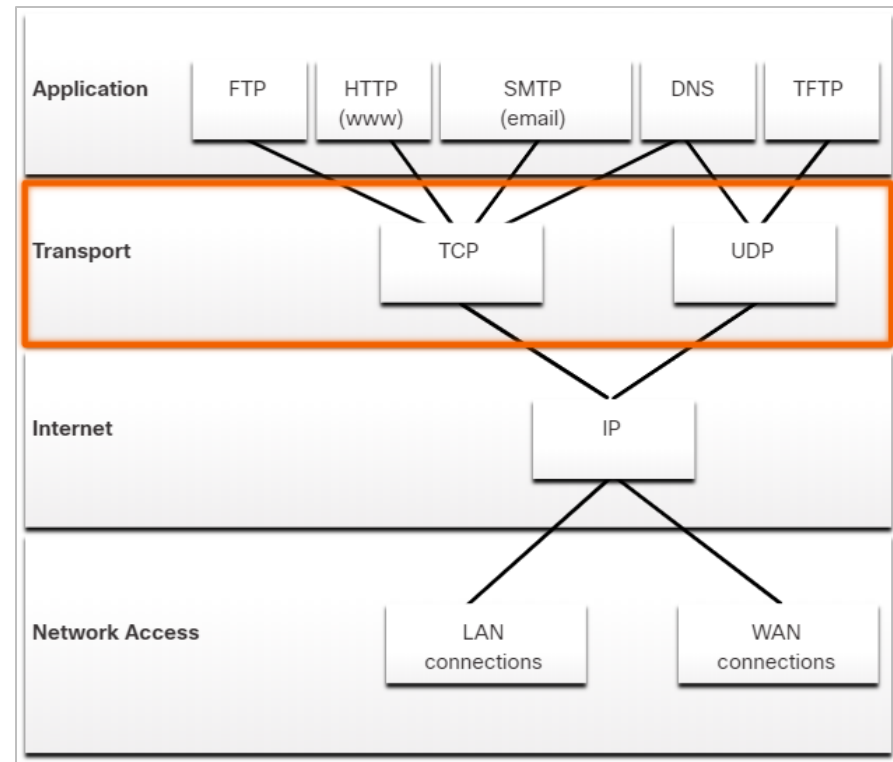
- O envio de alguns tipos de dados por uma rede, como um fluxo de comunicação completo, pode consumir toda a largura de banda disponível.
- Isso evita que outras conversas de comunicação ocorram ao mesmo tempo e também dificultam a recuperação de erros e a retransmissão de dados danificados.
- Como mostrado 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.



A Camada de Transporte

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 da camada de transporte (TCP e UDP) especificam como transferir mensagens entre hosts e são responsáveis por gerenciar os requisitos de confiabilidade de uma conversação.
- 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.



Protocolo de Controle de Transmissão (TCP)

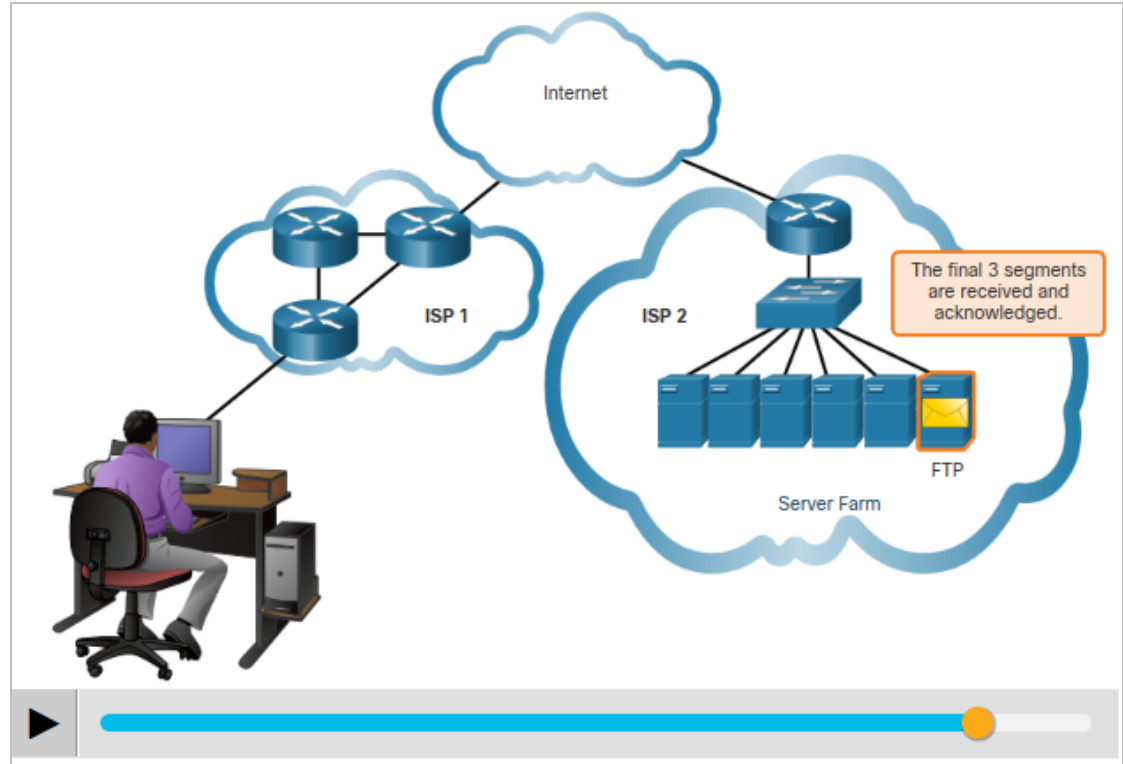
- O TCP é considerado um protocolo de camada de transporte confiável, completo, que garante que todos os dados cheguem ao destino.
- O TCP inclui campos que garantem a entrega dos dados do aplicativo. Esses campos exigem processamento adicional pelos hosts de envio e recebimento.
- O transporte TCP é análogo a enviar pacotes que são rastreados da origem ao destino.
- 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 quaisquer dados não reconhecidos após um certo período de tempo
 - Dados de sequência que podem chegar em ordem errada
 - Enviar dados a uma taxa eficiente que seja aceitável pelo receptor



Nota: TCP divide os dados em segmentos.

o protocolo TCP (Protocolo de Controle de Transmissão de Camada de Transporte) (Cont.)

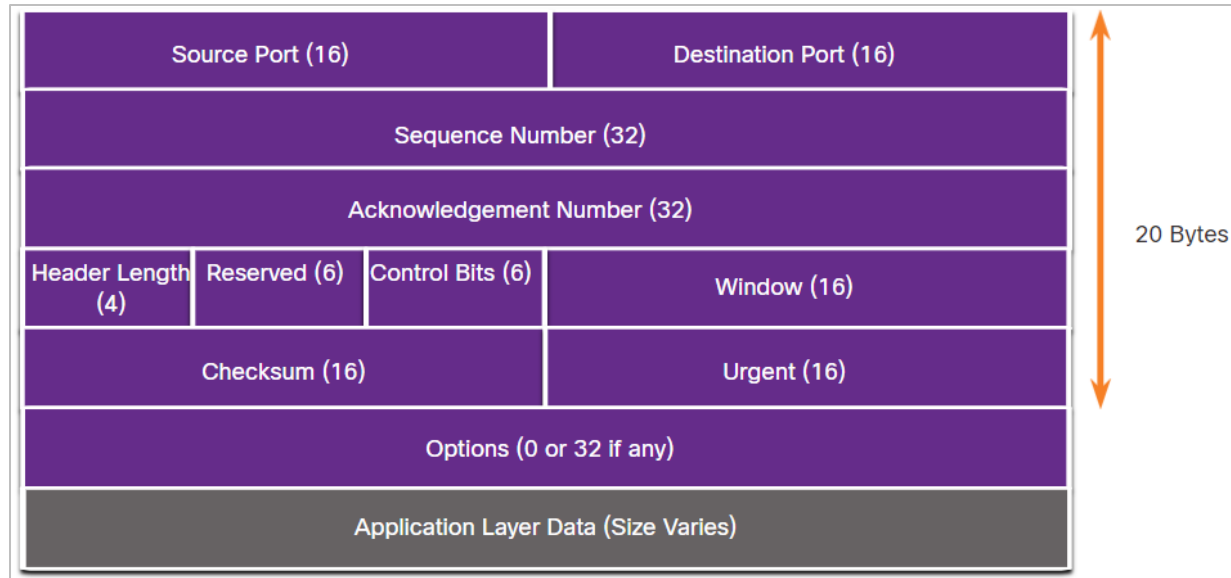
Para manter o estado de uma conversa e rastrear as informações, o TCP deve primeiro estabelecer uma conexão entre o remetente e o receptor. É por isso que o TCP é conhecido como um protocolo orientado a conexão.



A Camada de Transporte

Cabeçalho TCP

- O TCP é um protocolo com monitoração de estado, pois 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 (160 bits) de sobrecarga ao encapsular os dados da camada de aplicativo. A figura mostra os campos em um cabeçalho TCP.



Os campos de cabeçalho TCP da camada de transporte

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

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 pelo número da porta.
Número de Sequência	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 origem.
Tamanho do cabeçalho	Um campo de 4 bits conhecido como “offset de dados” que indica o comprimento do cabeçalho de 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 a função do segmento TCP.
Tamanho da janela	Um campo de 16 bits usado para indicar o número de bytes que podem ser aceitos ao mesmo tempo.
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.

Protocolo de Datagrama do Usuário (UDP)

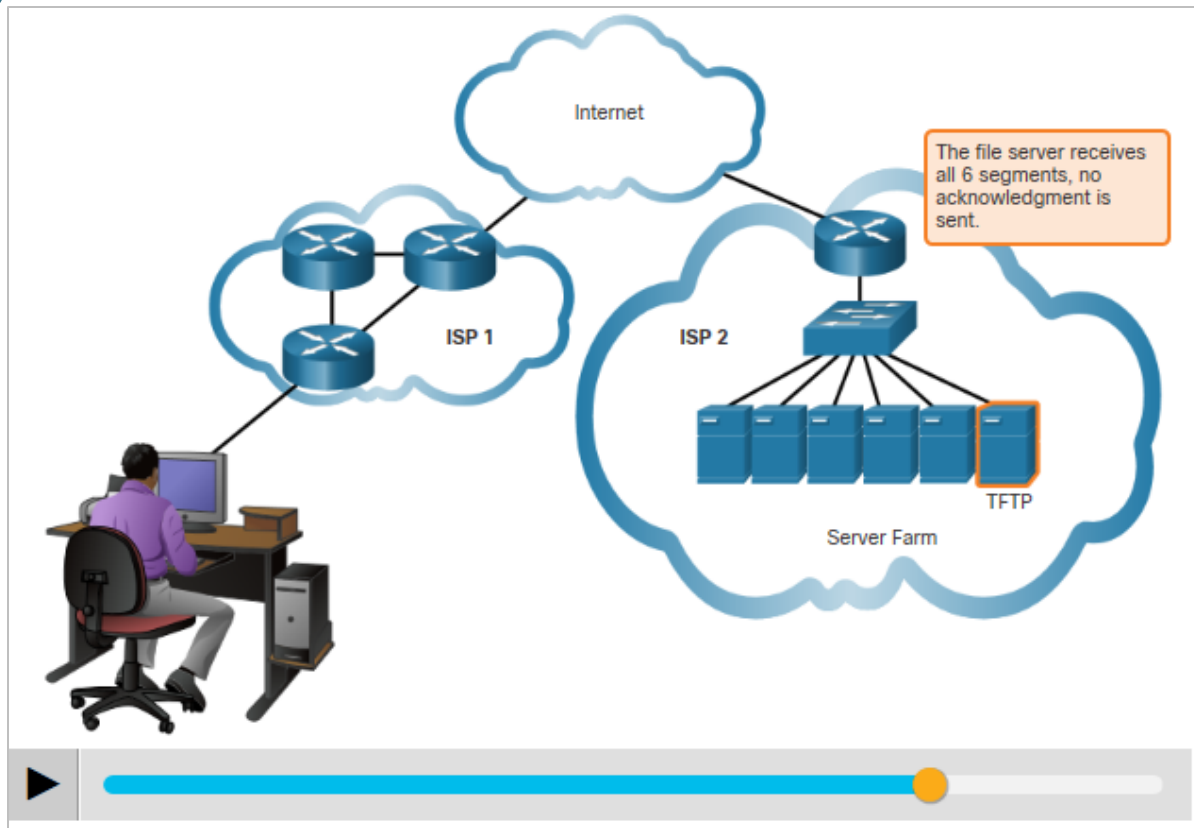
- O UDP é um protocolo de camada de transporte mais simples do que o TCP.
- Não fornece confiabilidade e controle de fluxo, o que significa que requer menos campos de cabeçalho.
- Os processos UDP do emissor e do receptor não precisam gerenciar a confiabilidade e o controle de fluxo, isso significa que os datagramas UDP podem ser processados mais rapidamente do que os segmentos TCP.
- O UDP fornece as funções básicas para fornecer datagramas entre os aplicativos apropriados, com muito pouca sobrecarga e verificação de dados.
- UDP é um protocolo sem conexão. Como o UDP não fornece confiabilidade ou controle de fluxo, ele não requer uma conexão estabelecida.
- O UDP também é conhecido como protocolo sem estado. Porque o UDP não rastreia as informações enviadas ou recebidas entre o cliente e o servidor.



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

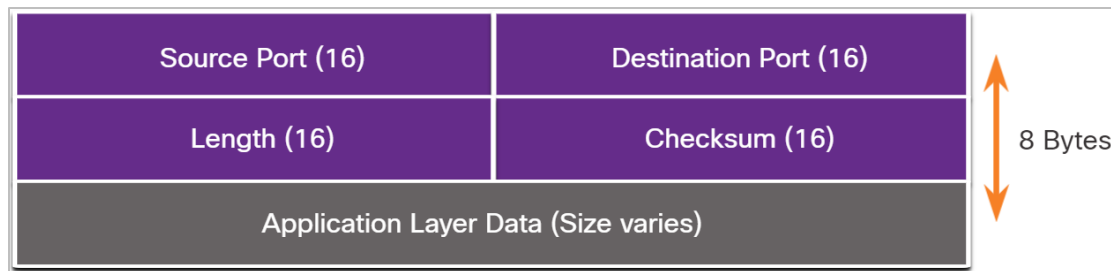
O protocolo UDP (Protocolo de Datagrama de Usuário da Camada de Transporte) (Cont.)

- UDP também é conhecido como um protocolo de entrega de melhor esforço porque não há confirmação de que os dados são recebidos no destino.
- 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.



Cabeçalho UDP

- O UDP é um protocolo sem estado, o que significa que nem o cliente, nem o servidor, rastreia 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.
- Os requisitos para o fornecimento de vídeo e voz ao vivo pela rede é que os dados continuem a fluir rapidamente. Os aplicativos de vídeo e voz ao vivo podem tolerar alguma perda de dados e são perfeitamente adequados para 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 tem apenas quatro campos e requer 8 bytes (64 bits). A figura mostra os campos em um cabeçalho UDP.



Campos de Cabeçalho UDP

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

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 pelo número da porta.
Número de Sequência	Um campo de 32 bits usado para fins de remontagem de dados.
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.

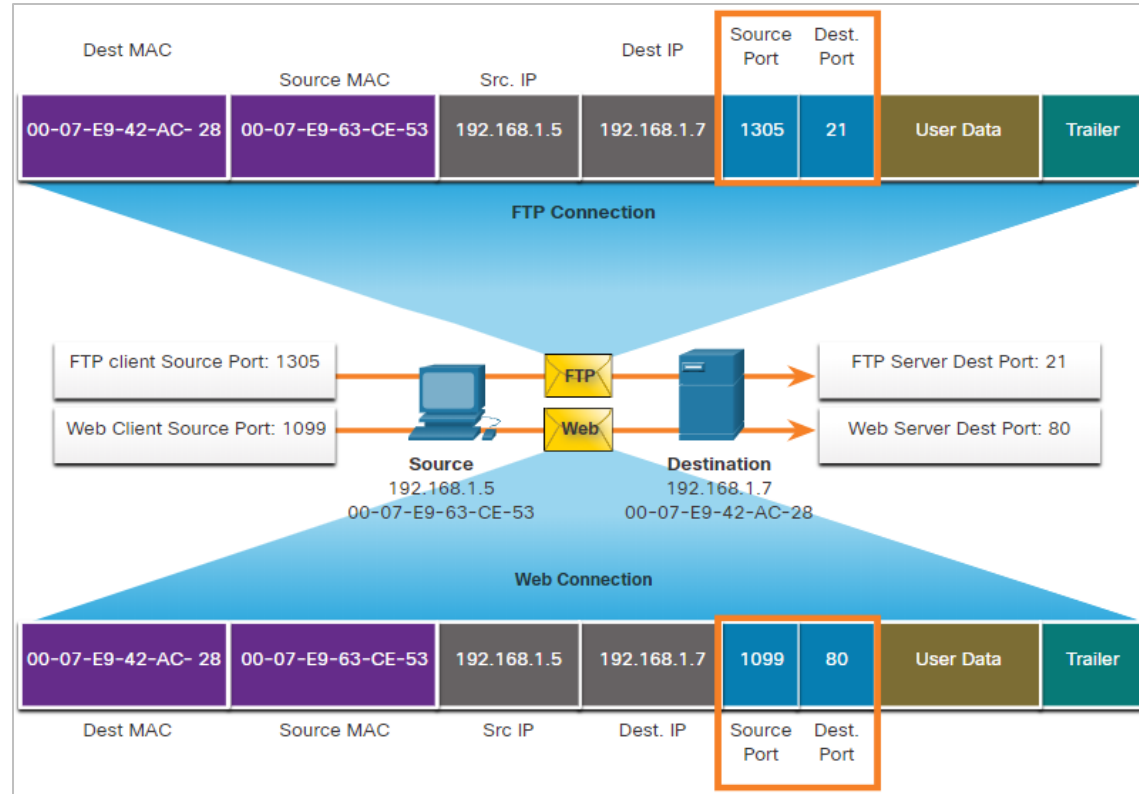
Pares de Soquete

- 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 número da porta de origem, ou endereço IP de destino e número da porta de destino, é conhecida como soquete.
- 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.

A Camada de Transporte

Pares de Soquete (Cont.)

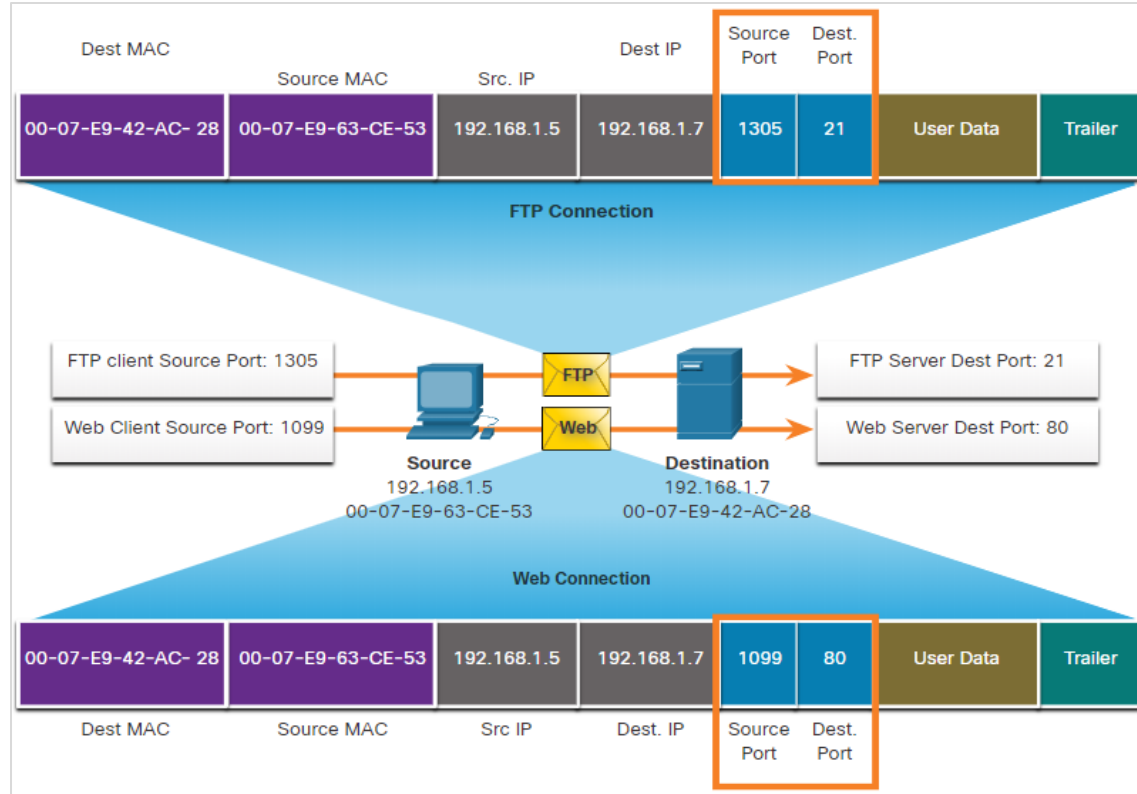
- Na figura, o PC está solicitando simultaneamente serviços de FTP e da Web do servidor de destino.
- A solicitação de FTP gerada pelo PC inclui os endereços MAC da Camada 2 e os endereços IP da Camada 3. A solicitação também identifica o número da porta de origem 1305 e a porta de destino, identificando 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.



A Camada de Transporte

Pares de Soquete (Cont.)

- Ele está usando o número da porta de origem 1099 e a porta de destino que identifica o serviço da web na porta 80.
- O socket é usado para identificar o servidor e o serviço que está sendo requisitado pelo cliente.
- Um soquete de cliente com 1099 representando o número da porta de origem pode ser 192.168.1.5:1099. O soquete em um servidor 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



9.2 Estabelecimento das sessões da camada de transporte

Processos do servidor TCP

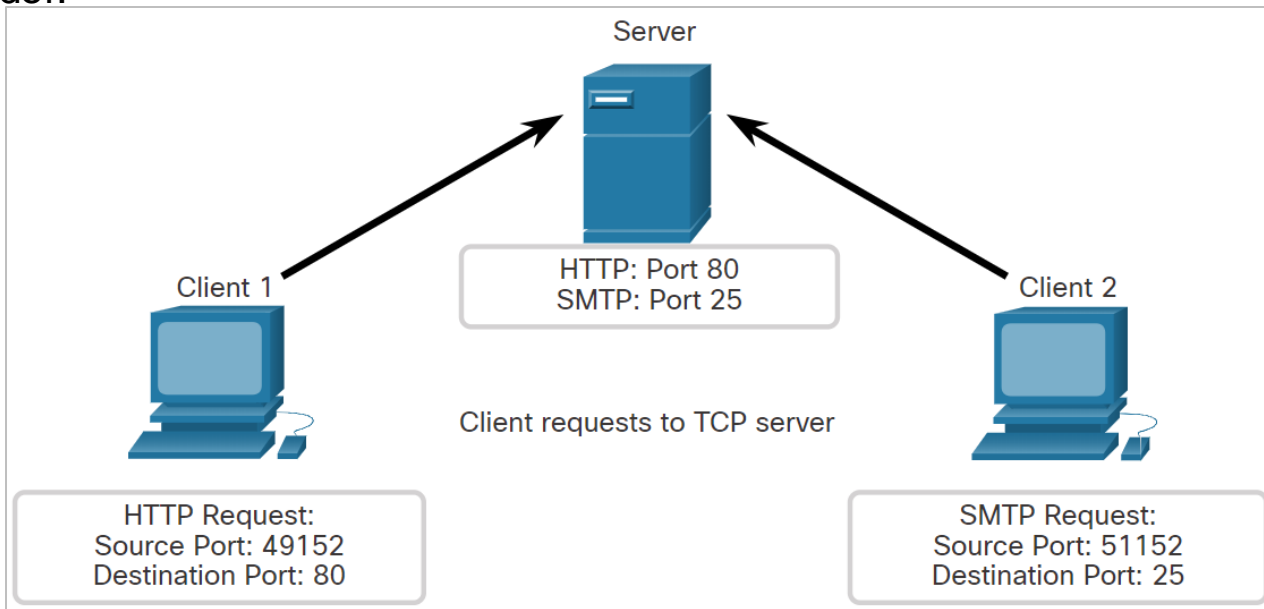
- Cada processo de aplicativo em execução em um servidor está configurado para usar um número de porta. O número da porta é atribuído automaticamente ou configurado manualmente por um administrador do sistema.
- Um servidor individual não pode ter dois serviços atribuídos ao mesmo número de porta dentro dos mesmos serviços de camada de transporte.
- Um host que executa um aplicativo de servidor da web e um aplicativo de transferência de arquivos não pode ser configurado 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.

Estabelecimento da Sessão da Camada de Transporte

Processos do servidor TCP (Cont.)

Clientes Enviando Requisições TCP

O cliente 1 está solicitando serviços da web e o cliente 2 está solicitando serviço de e-mail do mesmo servidor.

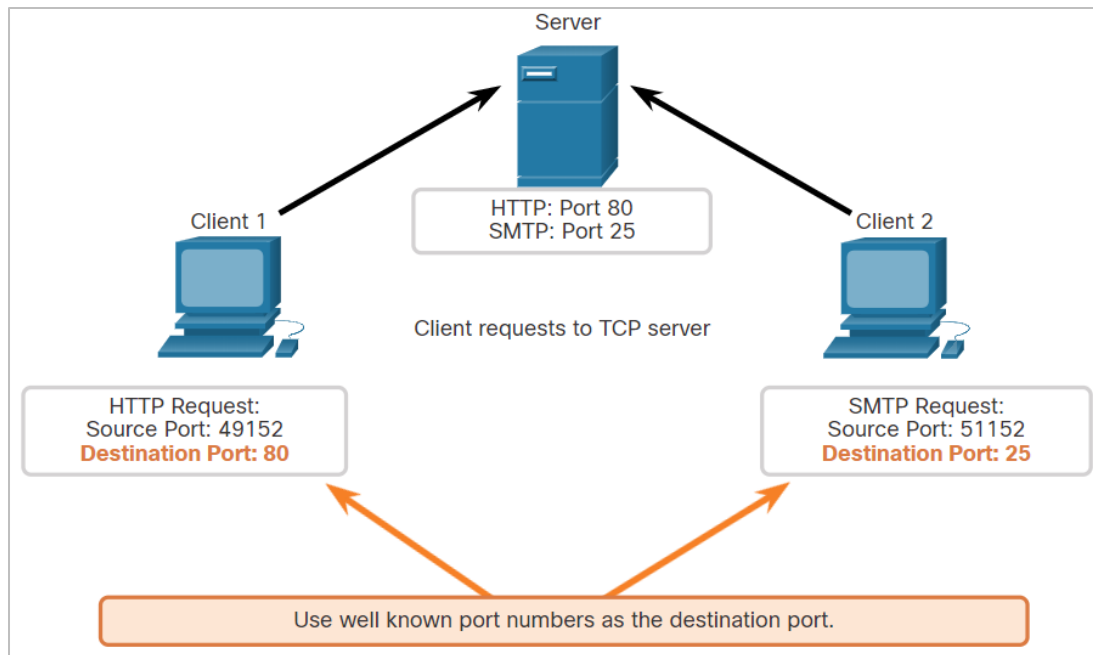


Estabelecimento da Sessão da Camada de Transporte

Processos do servidor TCP (Cont.)

Portas de Destino das Requisições

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

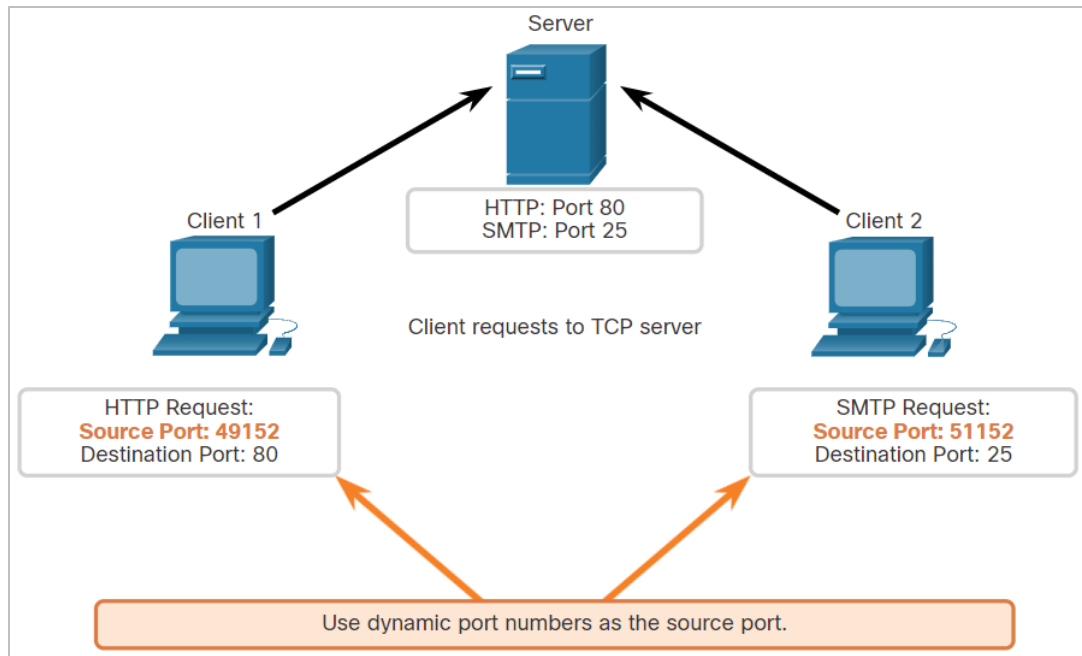


Estabelecimento da Sessão da Camada de Transporte

Processos do servidor TCP (Cont.)

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.

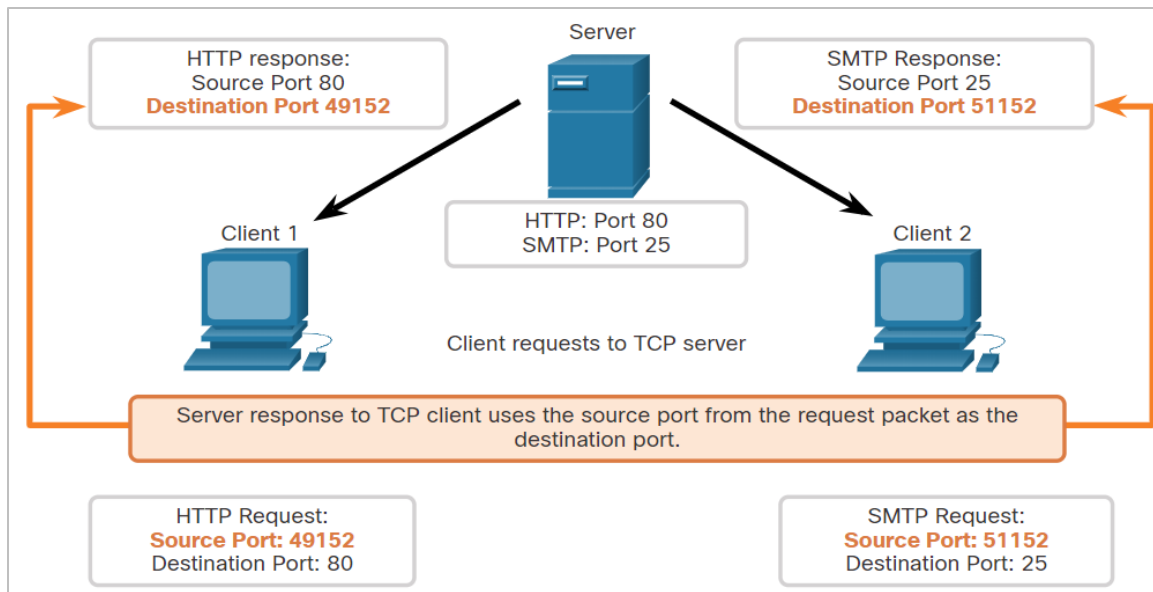


Estabelecimento da Sessão da Camada de Transporte

Processos do servidor TCP (Cont.)

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.

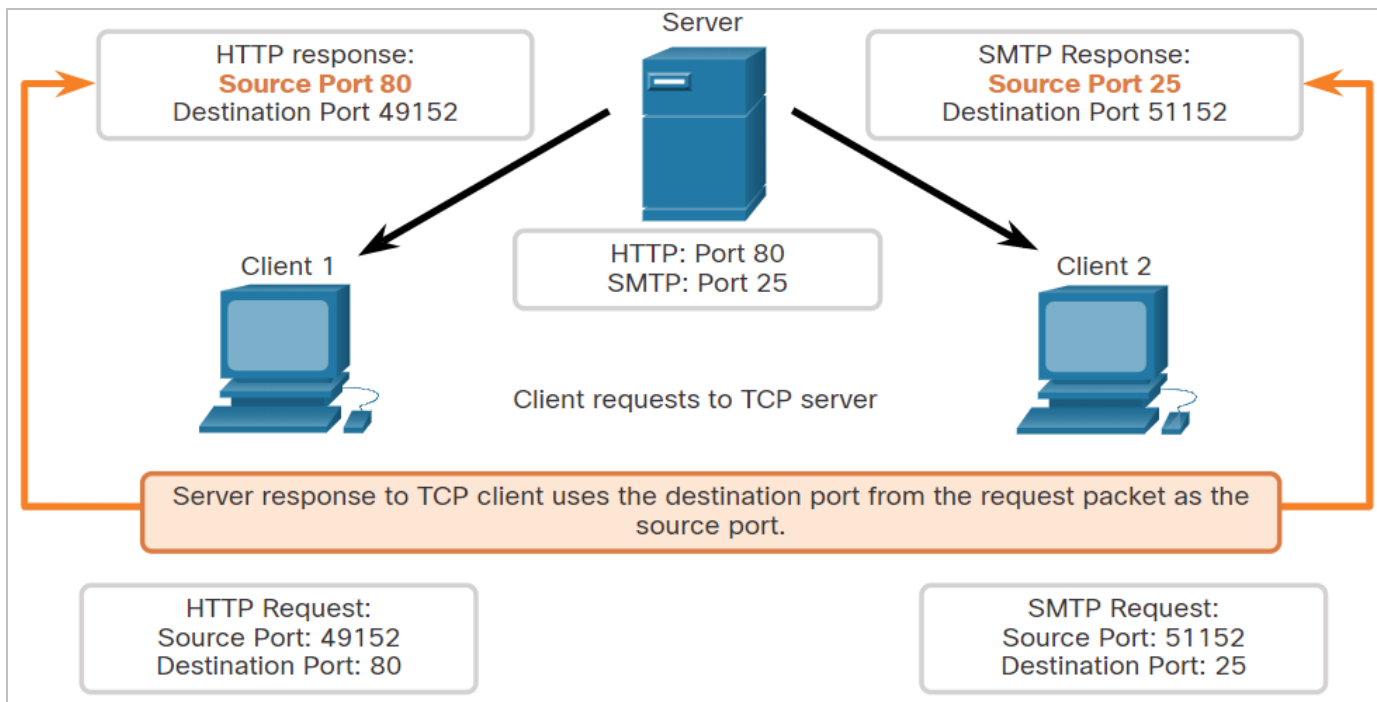


Estabelecimento da Sessão da Camada de Transporte

Processos do servidor TCP (Cont.)

Portas de Origem das Respostas

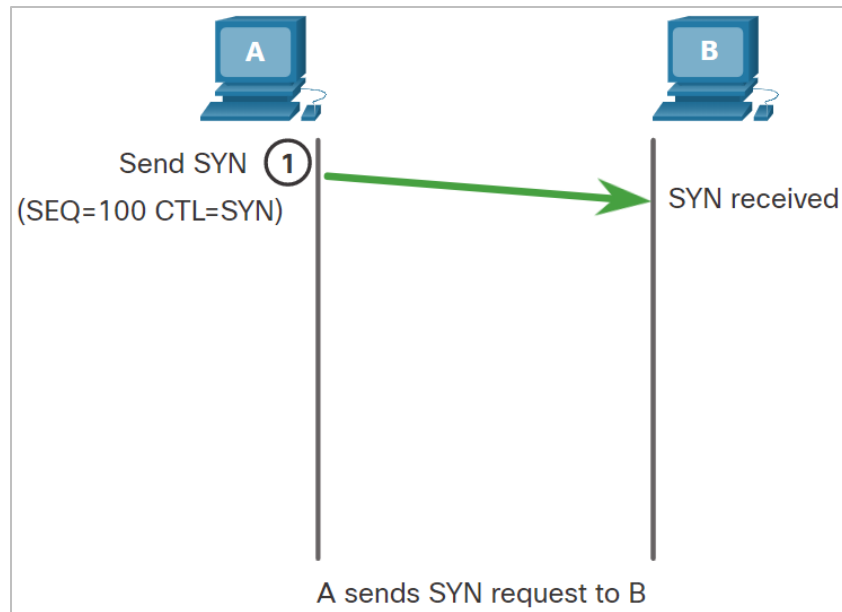
A porta de origem na resposta do servidor é a porta de destino original nas solicitações iniciais.



Estabelecimento da Sessão da Camada de Transporte

Estabelecimento de conexão TCP

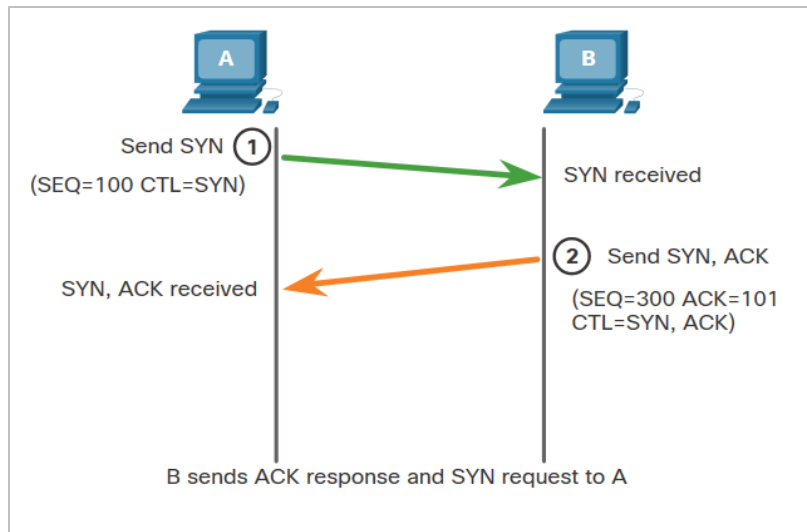
- Nas conexões TCP, o cliente host estabelece a conexão com o servidor usando o processo de handshake de três vias.
- O handshake de três vias valida se o host de destino está disponível para comunicação.
- As etapas de estabelecimento da conexão TCP são:
 - **Etapas 1. SYN:** O cliente inicial solicita uma sessão de comunicação cliente-servidor com o servidor.



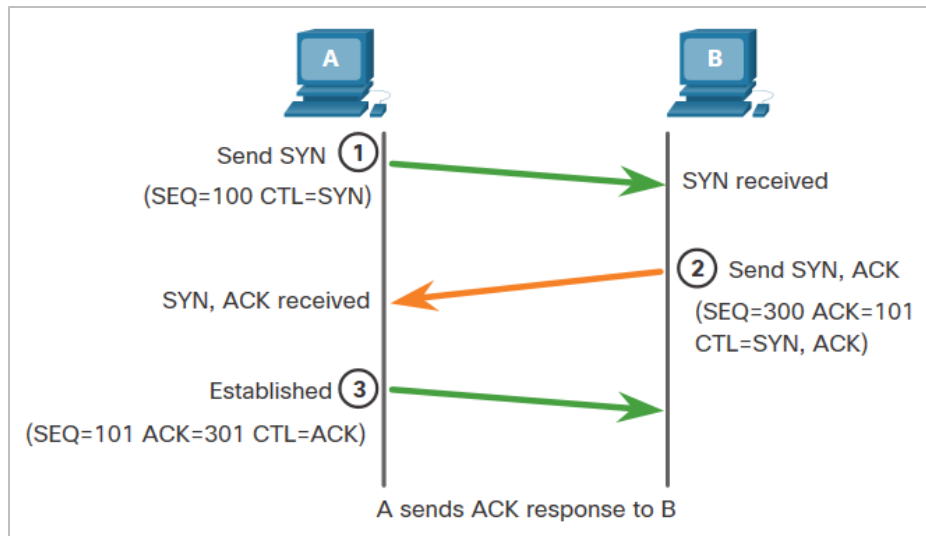
Estabelecimento da Sessão da Camada de Transporte

Estabelecimento de conexão TCP (Cont.)

Etapa 2. ACK e SYN: O servidor reconhece a sessão de comunicação cliente-servidor e solicita uma sessão de comunicação servidor-cliente.



Etapa 3. ACK: O cliente inicial reconhece a sessão de comunicação entre o servidor e o cliente.



Término da Sessão

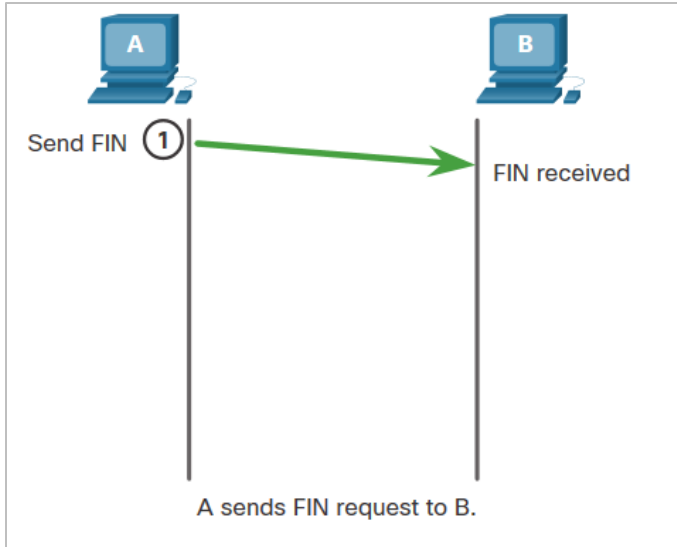
- Para fechar uma conexão, o flag de controle Finish (FIN) deve ser ligado no cabeçalho do segmento.
- Para terminar cada sessão TCP de uma via, um handshake duplo, consistindo de um segmento FIN e um segmento ACK (Acknowledgment) é usado.
- Portanto, para terminar uma conversa única permitida pelo TCP, quatro trocas são necessárias para finalizar ambas as sessões. O cliente ou o servidor podem iniciar o encerramento.
- Os termos cliente e servidor são usados como referência para simplificar, mas quaisquer dois hosts que tenham uma sessão aberta podem iniciar o processo de encerramento.
- Quando todos os segmentos tiverem sido confirmados, a conexão é encerrada.

Estabelecimento da Sessão da Camada de Transporte

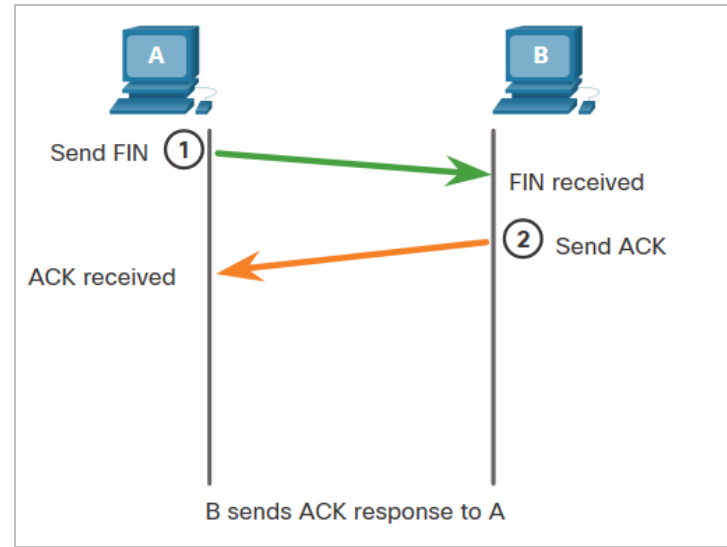
Término da Sessão (Cont.)

As etapas de encerramento da sessão são:

Etapa 1. FIN: Quando o cliente não tem mais dados para enviar no fluxo, ele envia um segmento com o sinalizador FIN definido.



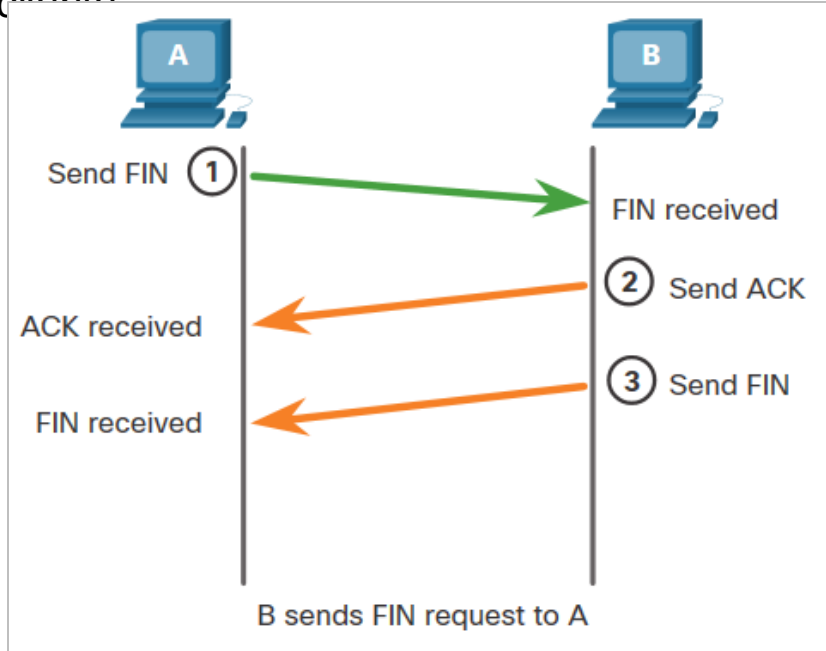
Etapa 2. ACK: O servidor envia um ACK para confirmar o recebimento do FIN para encerrar a sessão do cliente para o servidor.



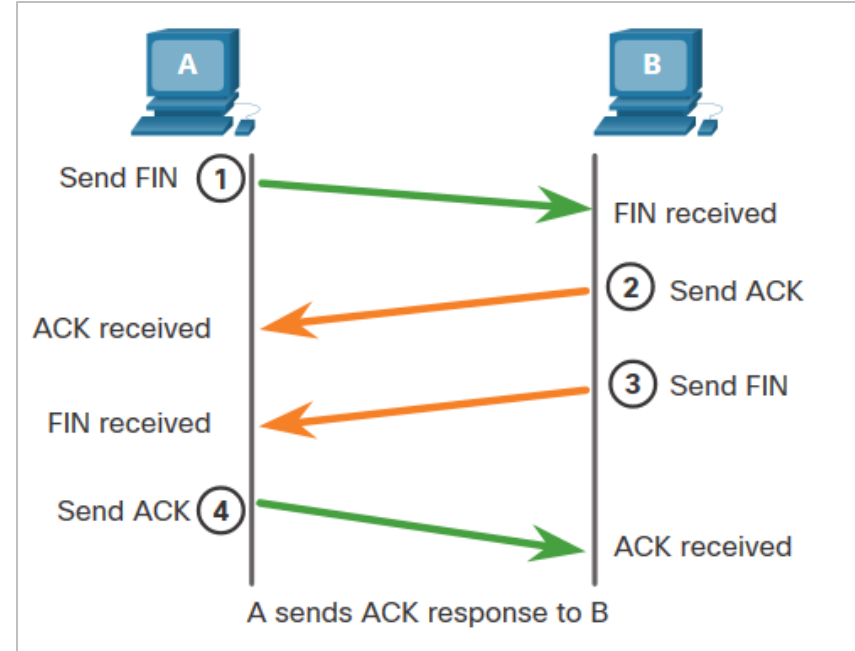
Estabelecimento da Sessão da Camada de Transporte

Término da Sessão (Cont.)

Etapa 3. FIN: O servidor envia um FIN ao cliente para encerrar a sessão servidor para cliente.



Etapa 4. ACK: O cliente responde com um ACK para reconhecer o FIN do servidor.



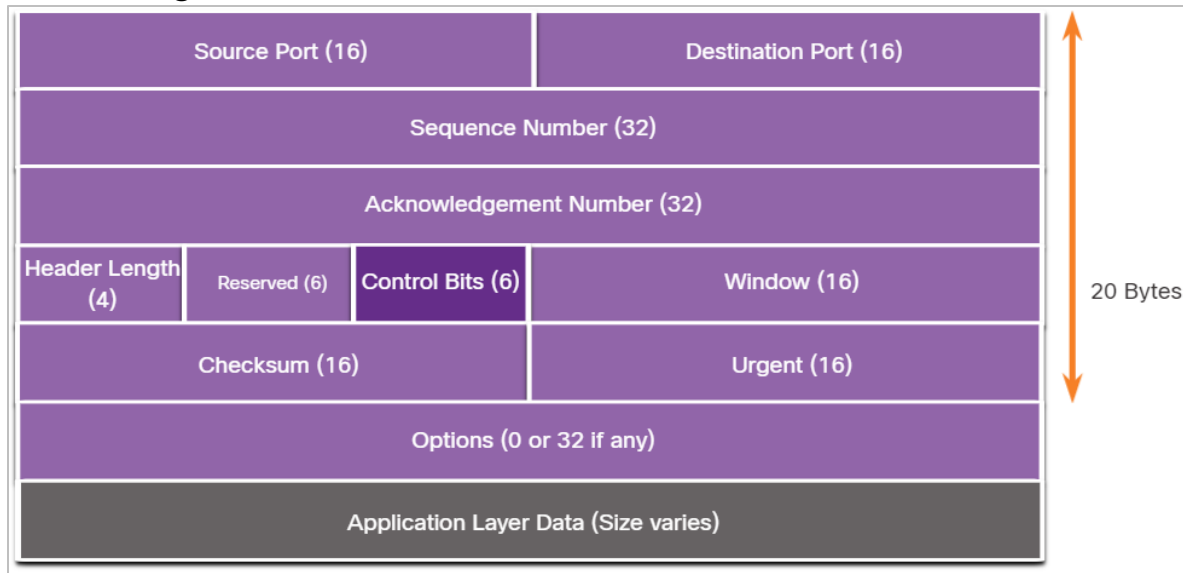
Análise de handshake TCP de três vias

- Os hosts mantêm o estado, rastreiam cada segmento de dados em uma sessão e trocam informações sobre os dados 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.
- As funções do handshake de três vias são:
 - 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.

Análise de Handshake TCP de Três Vias (Cont.)

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:

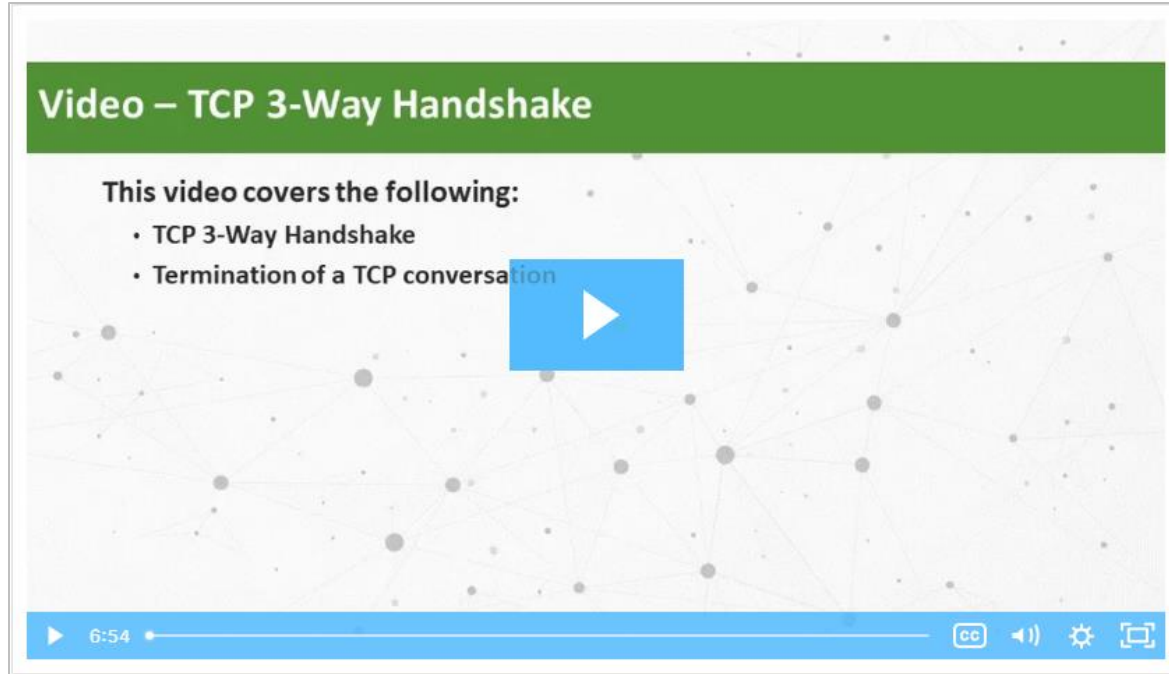
- **URG** - Campo indicador de urgência
- **ACK** - Indicador de confirmação usado no estabelecimento de conexão e encerramento de sessão
- **PSH** - Função Push
- **RST** - Redefina a conexão quando ocorrer um erro ou tempo limite
- **SYN** - Sincronizar números de sequência usados no estabelecimento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



Estabelecimento da Sessão da Camada de Transporte

Video – Handshake TCP de 3 Vias

Assista ao vídeo para saber mais sobre o handshake TCP de 3 vias.



Lab – Usando Wireshark para Observar o Handshake TCP de 3 Vias

Neste laboratório, você completará os seguintes objetivos:

- **Parte 1:** Prepare os hosts para capturar o tráfego
- **Parte 2:** Analise os pacotes usando o Wireshark
- **Parte 3:** Veja os pacotes usando tcpdump

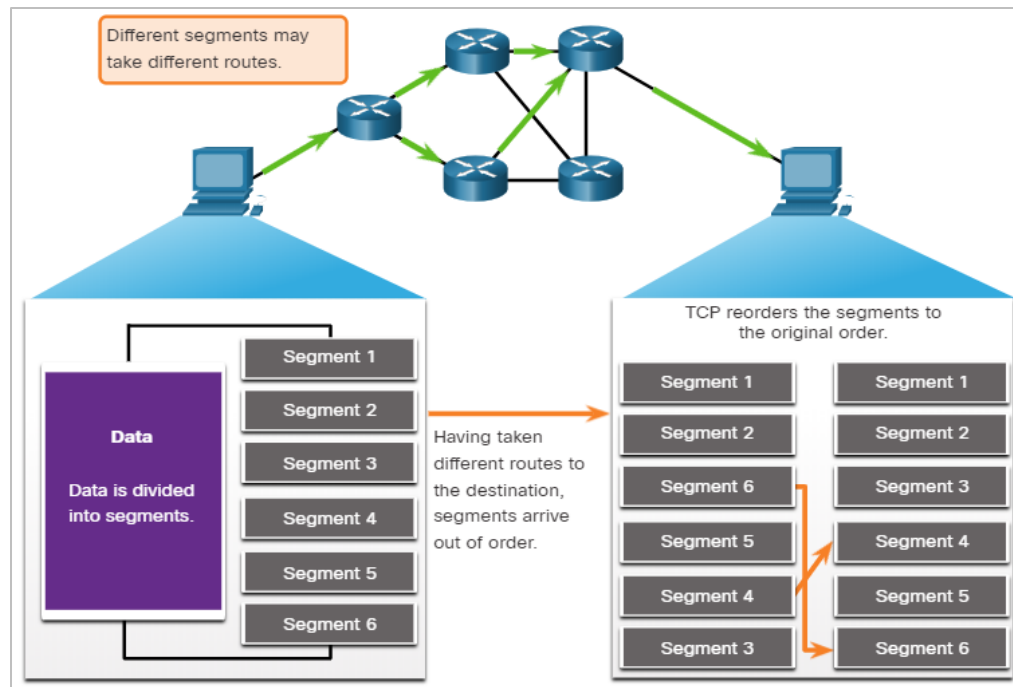
9.3 Confiabilidade da camada de transporte

Confiabilidade TCP - Entrega Garantida e Solicitada

- Pode haver momentos em que os segmentos TCP não chegam ao destino ou chegam fora de ordem.
- Para que a mensagem original seja compreendida pelo destinatário, todos os dados devem ser recebidos e os dados nesses segmentos devem ser remontados no pedido original.
- Os números de sequência são atribuídos no cabeçalho de cada pacote para atingir esse objetivo. O número de sequência representa o primeiro byte de dados do segmento TCP.
- Durante a configuração da sessão, um número de sequência inicial (ISN) é definido, o que 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 é efetivamente um número aleatório que impede certos tipos de ataques maliciosos.

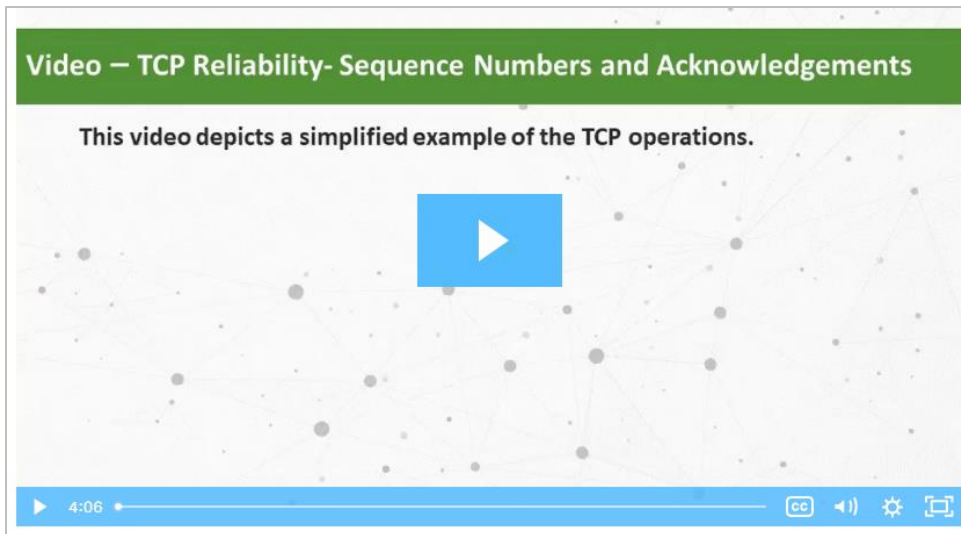
Confiabilidade TCP - Entrega Garantida e Solicitada (Cont.)

- Os números de sequência do segmento indicam como remontar e reordenar os segmentos recebidos, como mostrado na figura.
- 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.
- Então, quando os segmentos com os bytes ausentes chegam, esses segmentos são processados em ordem.



Video - Confiabilidade TCP – Números de Sequência e Reconhecimentos

- Uma das funções do TCP é garantir que cada segmento chegue ao seu destino. Os serviços TCP no host de destino reconhecem os dados que foram recebidos pelo aplicativo de origem.
- Clique em Reproduzir na figura para assistir a uma aula sobre números de sequência TCP e confirmações.

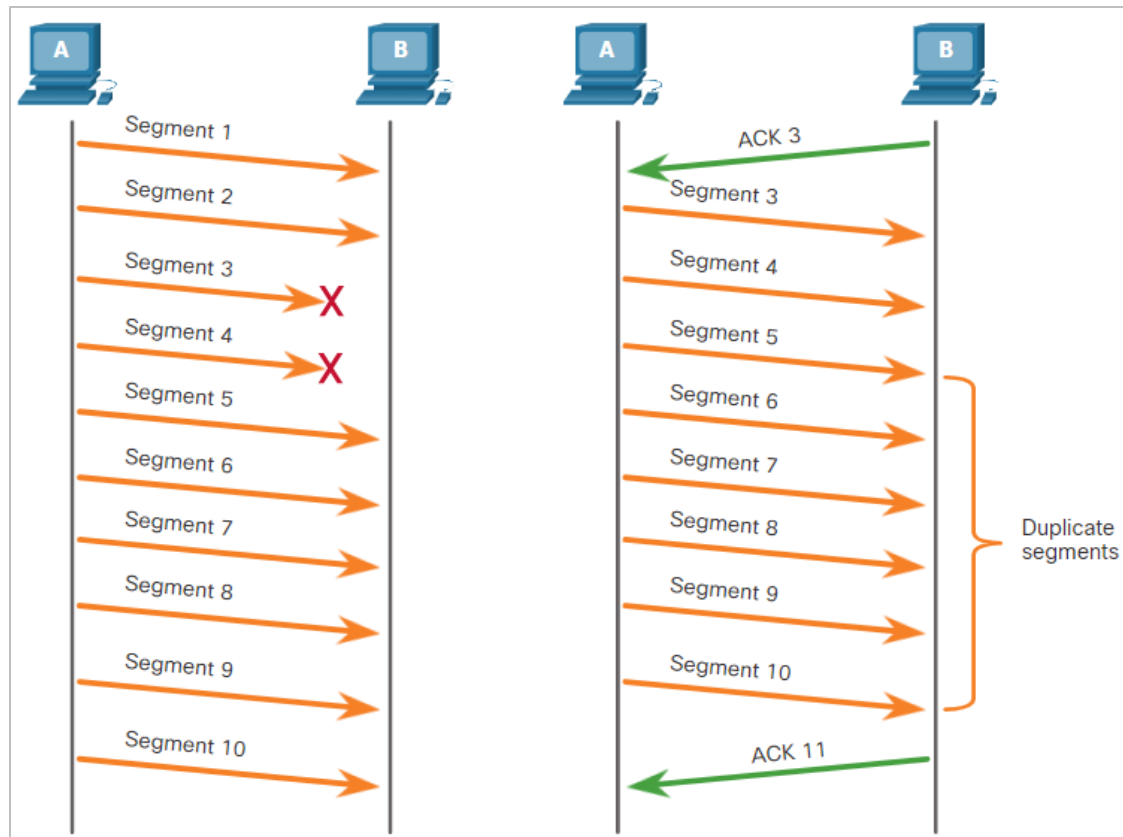


Confiabilidade TCP - Perda de Dados e Retransmissão

- O TCP fornece métodos de gerenciamento de perdas de segmento, retransmitindo os segmentos para 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.

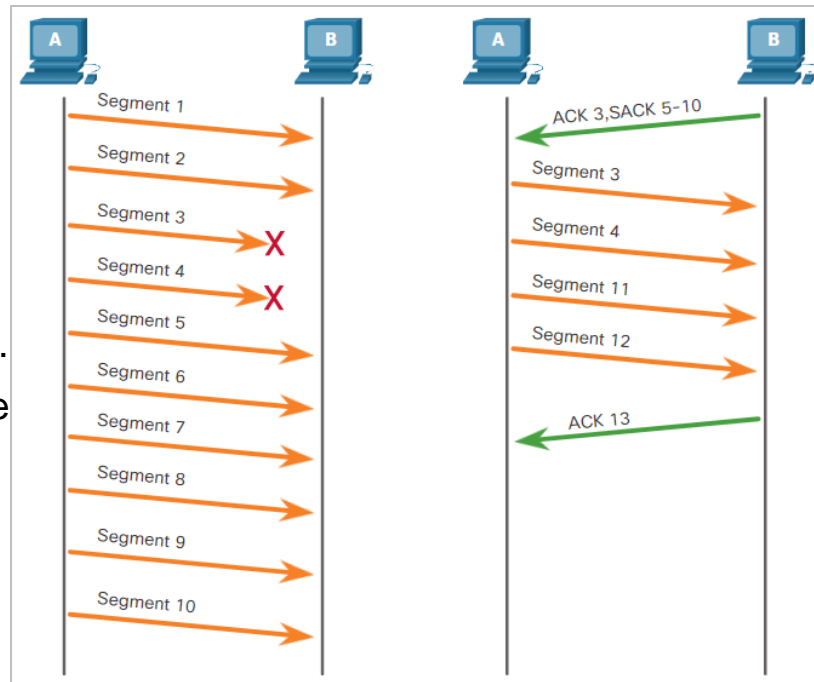
Confiabilidade TCP - Perda de Dados e Retransmissão (Cont.)

- Na figura, 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 responderá com confirmação, especificando que o próximo segmento esperado é o segmento 3.
- O Host A não tem ideia se algum outro segmento chegou ou não. Ele 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.



Confiabilidade TCP - Perda de Dados e Retransmissão (Cont.)

- Os sistemas operacionais host empregam um recurso TCP opcional denominado confirmação seletiva (SACK), negociado durante o handshake triplo.
- Se ambos os hosts suportam SACK, o receptor pode reconhecer quais segmentos (bytes) foram recebidos, incluindo quaisquer segmentos descontínuos.
- O host de envio precisaria apenas retransmitir os dados ausentes.
- Na figura, 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 pode reconhecer que recebeu os segmentos 1 e 2 (ACK 3) e, seletivamente, reconhecer os segmentos 5 a 10 (SACK 5-10). O host A só precisaria reenviar os segmentos 3 e 4.



Video - Confiabilidade TCP - Perda de Dados e Retransmissão

Clique em Reproduzir na figura para assistir a uma aula sobre retransmissão TCP.

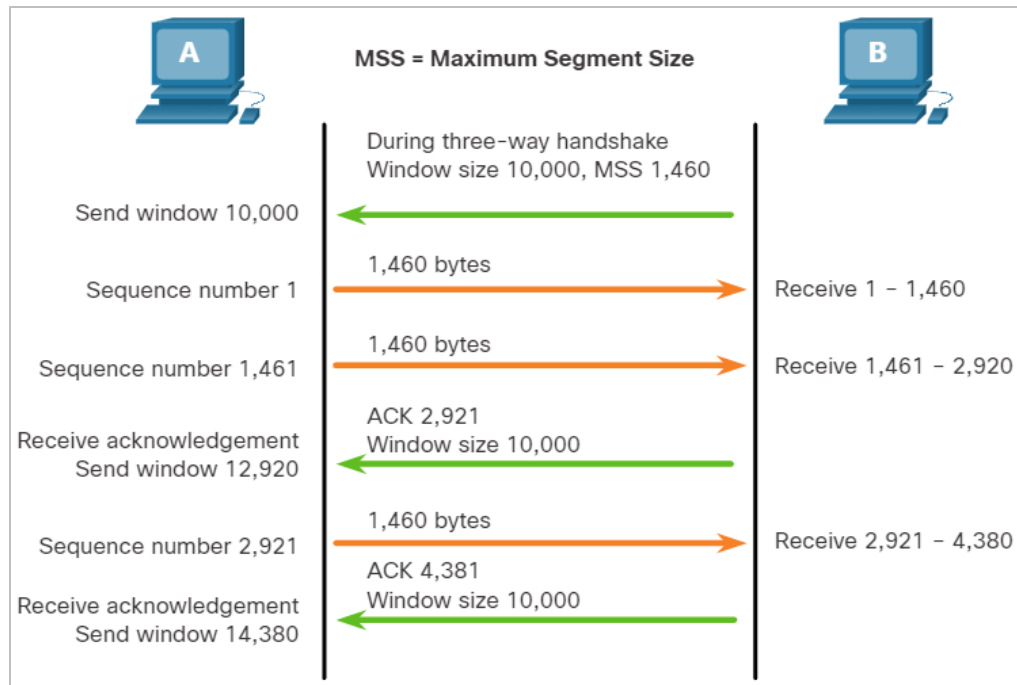


Controle de Fluxo TCP - Tamanho da Janela e Reconhecimentos

- 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.
- O tamanho da janela que 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.

Controle de Fluxo TCP - Tamanho da Janela e Reconhecimentos (Cont.)

- A figura mostra um exemplo de tamanho da janela e confirmações.
- O tamanho da janela é incluído em cada segmento TCP de maneira que o destino possa alterar o tamanho da janela a qualquer momento, dependendo da disponibilidade de 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 a fonte recebe uma confirmação, ela pode continuar enviando mais dados para a sessão.



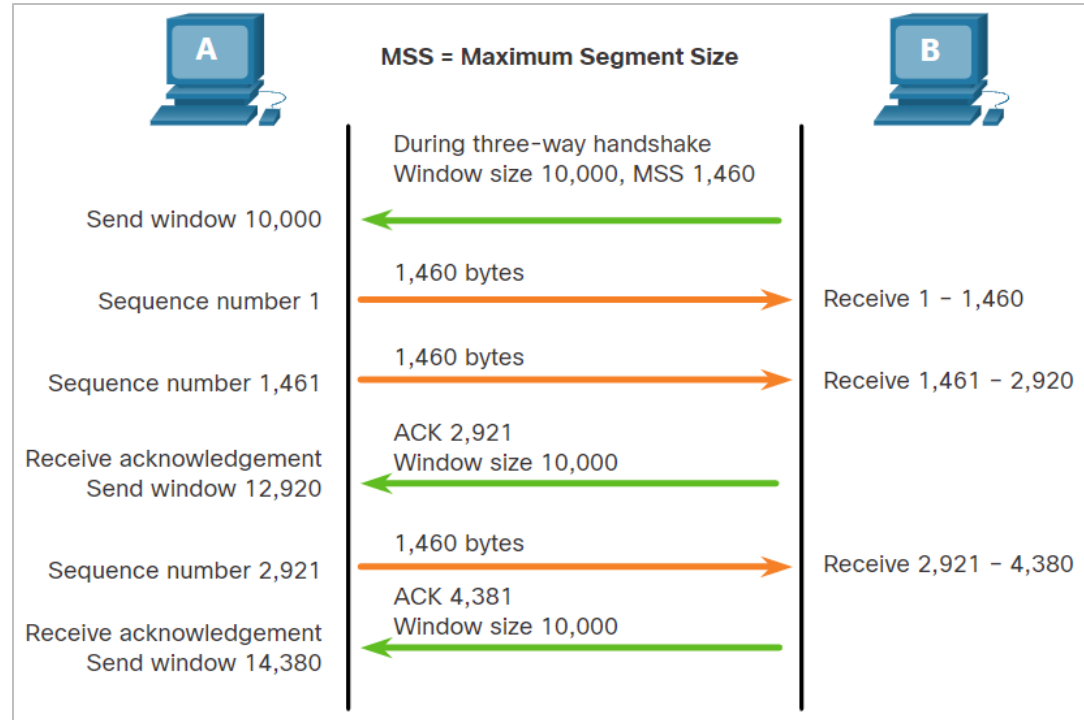
Controle de Fluxo TCP - Tamanho da Janela e Reconhecimentos (Cont.)

- O destino não esperará que todos os bytes do tamanho de sua janela sejam recebidos antes de responder com uma confirmação.
- À medida que os bytes forem recebidos e processados, o destino enviará confirmações para informar à origem que pode continuar a enviar bytes adicionais.
- 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.
- 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 envia uma confirmação a cada dois segmentos que recebe. A vantagem de janelas móveis é que permite que o emissor transmita continuamente segmentos, desde que o receptor esteja reconhecendo segmentos anteriores.

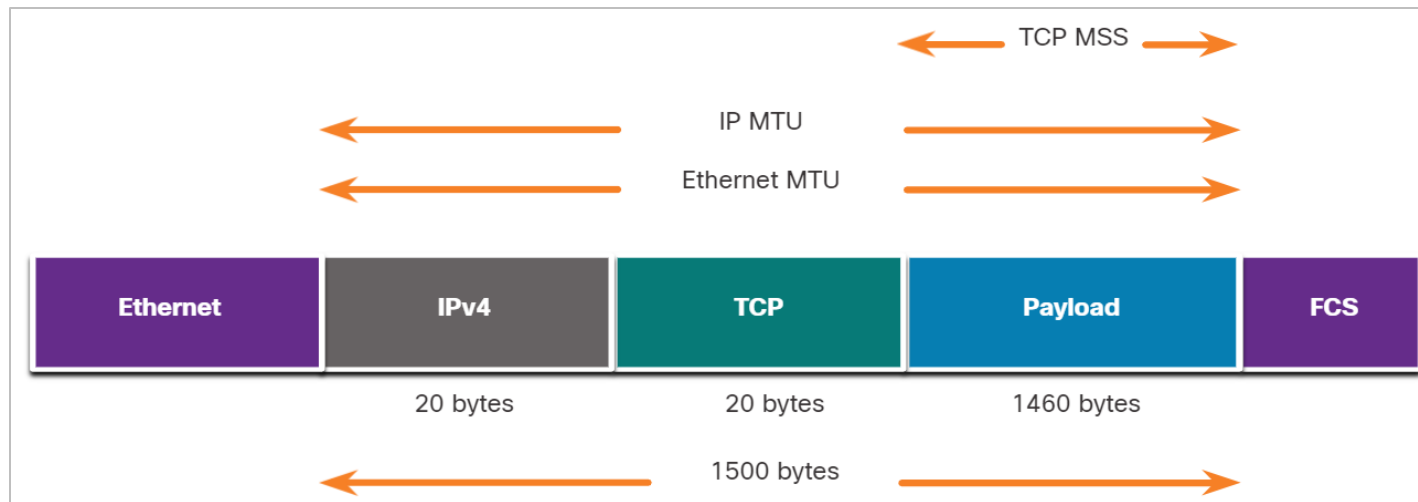
Controle de Fluxo TCP - Tamanho Máximo do Segmento (MSS)

- Na figura, a fonte está transmitindo 1.460 bytes de dados dentro de cada segmento TCP. 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 incluído durante o handshake triplo.



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

- 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 1460 bytes, conforme mostrado na figura.

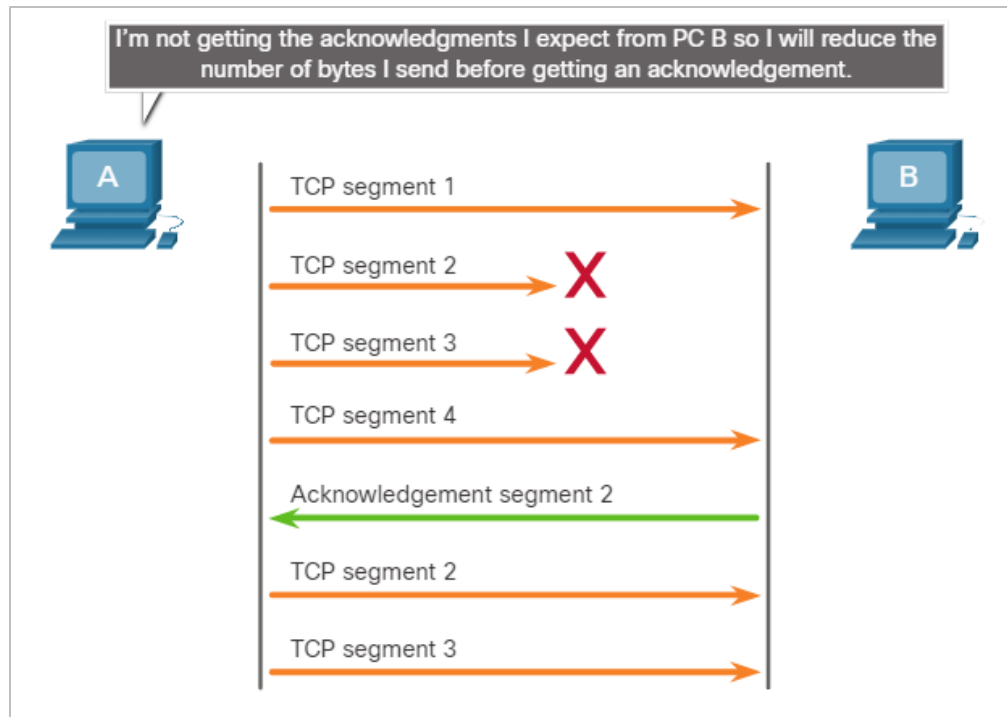


Controle de Fluxo TCP - Prevenção de Congestionamento

- 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.

Controle de Fluxo TCP - Prevenção de Congestionamento (Contd.)

- 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 mostrado 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.
- 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.



Lab – Explorando o Nmap

- A varredura de portas geralmente faz parte de um ataque de reconhecimento.
- Há uma variedade de métodos de varredura de portas que podem ser usados.
- Vamos explorar como usar o utilitário Nmap. Nmap é um utilitário de rede poderoso usado para descoberta de rede e auditoria de segurança.

9.4 O resumo da camada de transporte

O Que Aprendi Neste Módulo?

- A camada de transporte é o elo entre a camada de aplicação e as camadas inferiores do modelo OSI que são responsáveis pela transmissão da rede.
- 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 de segmento, identificar aplicativos e multiplexar conversas.
- O TCP é estável e confiável. Ele reconhece os dados, reenvia os dados perdidos e entrega os dados em ordem sequencial. TCP é usado para e-mail e web.
- UDP não tem estado e é rápido. Ele tem baixa sobrecarga, não requer confirmações, não reenvia dados perdidos e processa os dados na ordem em que eles chegam. UDP é usado para VoIP e DNS.

O Que Aprendi Neste Módulo? (Continuação)

- 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.
- O handshake triplo estabelece que o dispositivo de destino está presente na rede. Ele verifica se o dispositivo de destino tem um serviço ativo que está aceitando solicitações no número da porta de destino que o cliente inicial pretende usar.
- Os seis sinalizadores de bits de controle são: URG, ACK, PSH, RST, SYN e FIN e são usados para identificar a função das mensagens TCP enviadas.
- 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 sistemas operacionais host de hoje geralmente empregam um recurso TCP opcional denominado confirmação seletiva (SACK), que é negociado durante o handshake triplo.
- 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.

