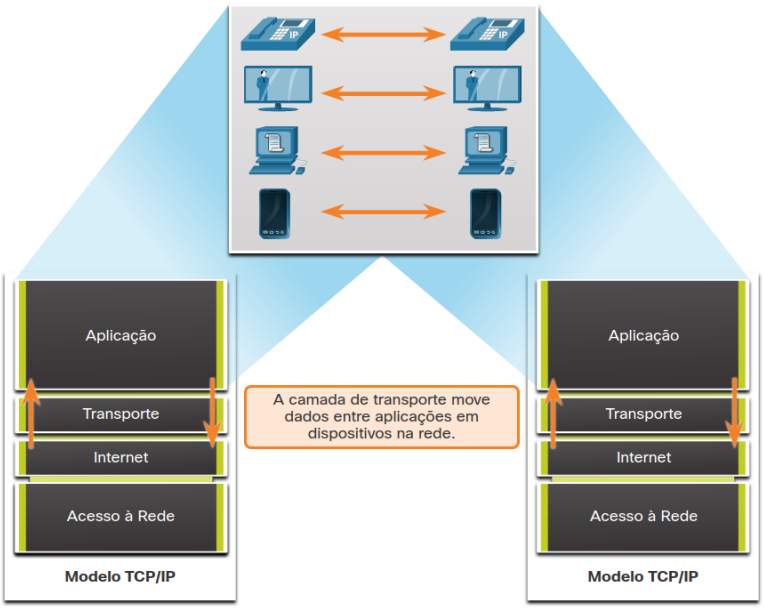
Camada de Transporte

↳ Responsável pela comunicação lógica entre os aplicativos executados em hosts diferentes



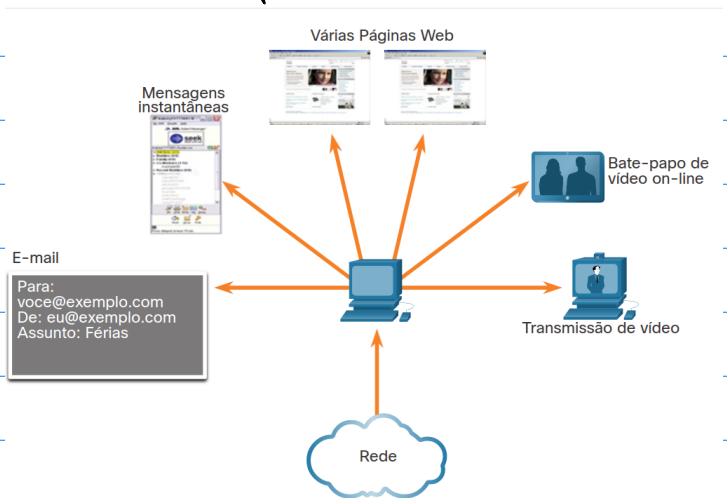
- Protocolos da Camada de rede
- TCP (Transmission control protocol)
- UDP (user datagram protocol)

Responsabilidades da Camada de transporte

- Rastreamento de conversações individuais

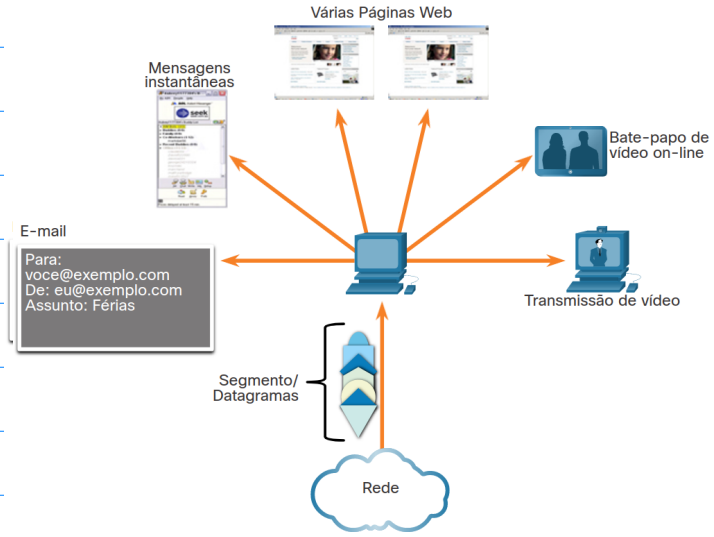
↳ Cada conjunto de dados que flui entre o app origem e destino é conhecido como conversa e é rastreado separadamente.

↳ Camada de Transporte deve manter e monitorar essas conversações.



- Segmentação de dados e remontagem de segmentos

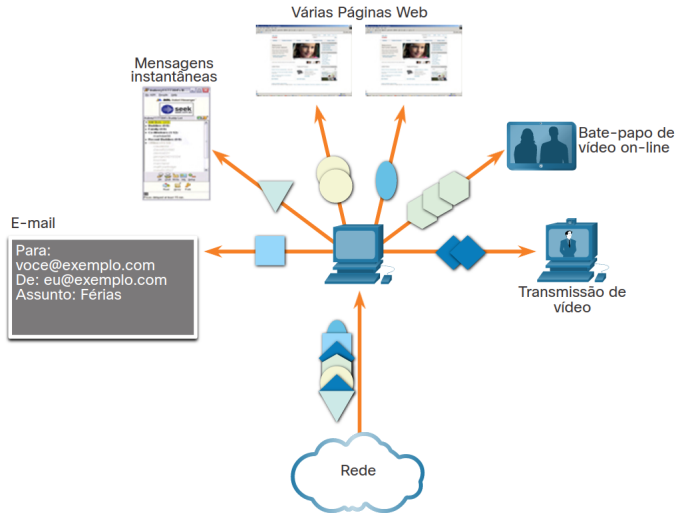
↳ Divide os dados do app em blocos de tamanho adequado (segmentos ou datagrama) que são mais fáceis de gerenciar e transportar.



- Adiciona informações de cabeçalho

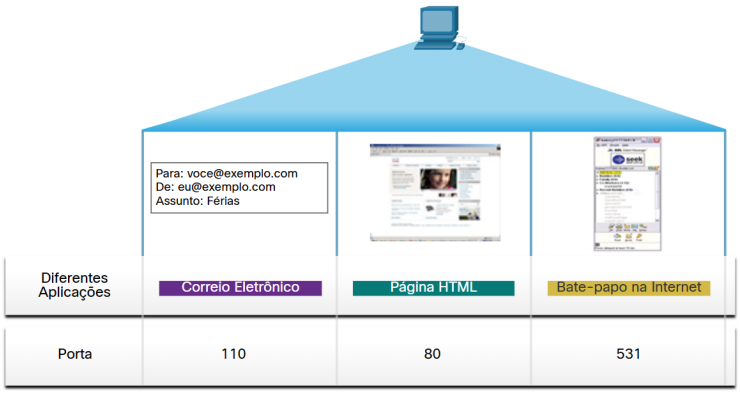
↳ Add informações de cabeçalho contendo dados binários organizados em vários campos a cada bloco de dados.

↳ Garante que todos os apps recebam os dados corretos.



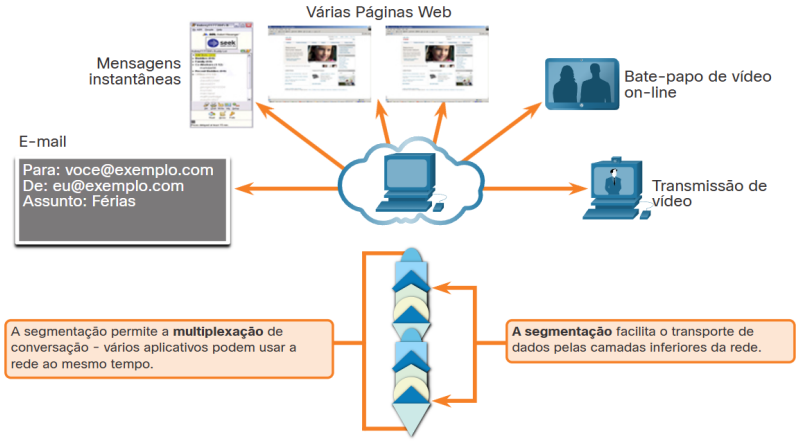
Identificação das aplicações

↳ Para passar fluxos de dados para as apps certas, a camada de transporte identifica o app de destino usando um identificador chamado número de porta



Multiplexação das conversas

- A multiplexação / segmentação é usada para permitir que diferentes conversas de comunicação sejam intercaladas na mesma rede.



Protocolos da Camada de Transporte

- TCP (Transmision Control Protocol)
  - Fornece confiabilidade e controle de fluxo.
  - Num. e rastreamento de segmentos.
  - Confirmação de dados recebidos.
  - Retransmissão de dados NÃO confirmados.
  - Dados podem chegar na ordem errada.
  - Envia dados de uma forma eficiente.

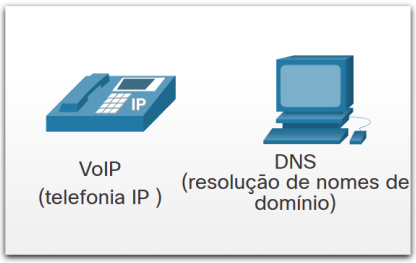
* Primeiro ele estabelece uma conexão entre o remetente e o receptor
Orientado a conexão.

- UDP (User Datagram Protocol)
  - Não gerenciam confiabilidade e controle de fluxo
- O UDP divide os dados em datagramas (segmentos)
- É sem conexão, confirmação de entrega
- Protocolo de entrega de menor esforço

Resumo

1. A camada de transporte é responsável por estabelecer uma sessão de comunicação temporária entre os aplicativos host de origem e de destino.
2. A camada de transporte é responsável pela multiplexação de conversações, segmentação de dados e remontagem de segmentos e rastreamento de conversas individuais.
3. UDP é um protocolo de entrega de melhor esforço enquanto TCP é um protocolo de transporte confiável.
4. UDP seria usado por aplicativos VoIP sensíveis ao tempo.

UDP



VoIP
(telefonia IP)

DNS
(resolução de nomes de domínio)

Propriedades necessárias para escolha do protocolo:

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

TCP

SMTP/IMAP
(E-mail)

HTTP/HTTPS
(World Wide Web

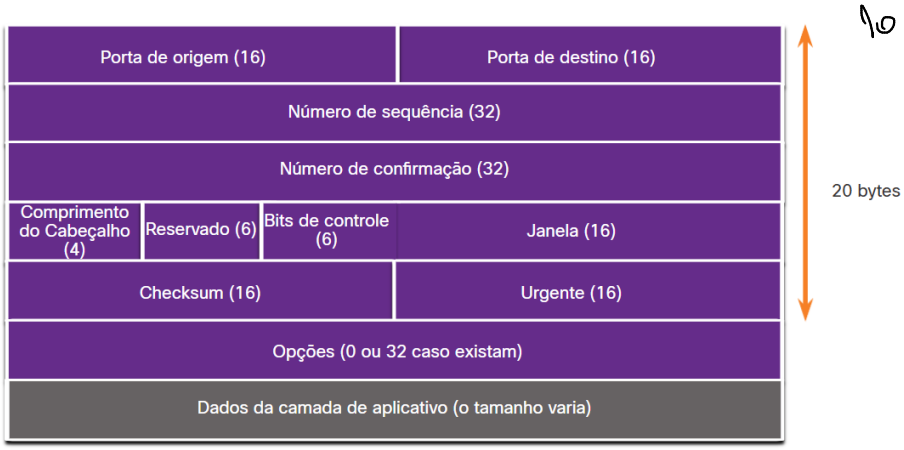
Propriedades necessárias para escolha do protocolo:

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

TCP

- Estabelece uma sessão.
- Garante entrega confiável.
- Fornece entrega no mesmo pedido (Garante que os segmentos sejam remontados da forma correta)
- Suporte controle de fluxo.

Cabeçalho TCP



No campo

1. UDP é um protocolo de camada de transporte de entrega de melhor esforço sem estado.
2. O cabeçalho UDP consiste em quatro campos em um cabeçalho de 8 bytes.
3. TFTP e VoIP exigem o uso do protocolo de camada de transporte UDP.
4. Os cabeçalhos TCP e UDP incluem campos de número de porta de origem e destino.

Aplicações que usam TCP

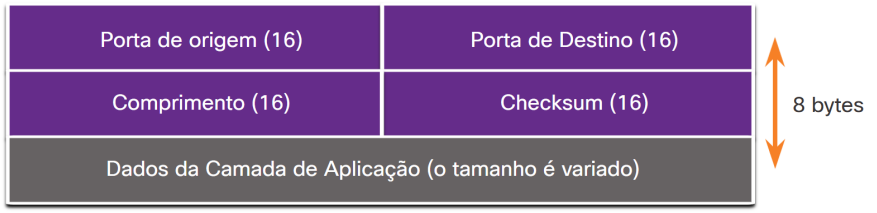
- HTTP → SMTP
- FTP → SSH

UDP

- Dados Reagrupados na ordem que são recebidos.
- Segmentos perdidos não são reenviados.
- Não há estabelecimento de sessão.
- O envio não é informado sobre a disponibilidade do recurso.

Cabeçalho UDP

4 campos



Aplicações que usam UDP

- Aplicativos de vídeo e multimídia ao vivo
- Solicitações simples e app de reporte
- Aplicativos que lidam com a confiabilidade

→ DHCP, DNS, SNMP, TFTP, VoIP, videoconferência

Números de porta

↳ Porta origem está associado ao app origem no host local, já a porta destino está associado ao app de destino no host remoto.

Pares de sockets

- O socket é usado para identificar o servidor e o serviço que está sendo solicitado
  ↳ Eles 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 dos outras.
  ↳ O Num de porta age como um endereço de retorno para a aplicação que faz a solicitação, de modo que quando uma resposta é retornada, ela vai para a aplicação correta.

Grupos de Números de porta

- Divisão da IANA

Grupo de Portas	Intervalo de números	Descrição
Portas Comuns	**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.
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.

Número de portas Comuns

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
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 de E-mail)
143	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)

* Alguns aplicações podem usar tanto TCP quanto UDP.

Ex: DNS → UDP (quando os clientes enviam requisições a um servidor DNS) → TCP (comunicação entre 2 servidores DNS)

Comando Netstat

Utilizado para verificar quais conexões TCP ativas estão sendo executados em um host de rede.

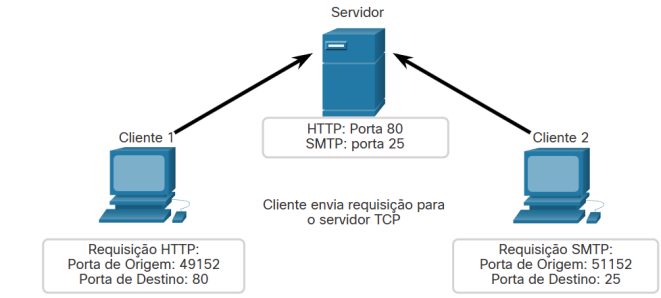
```
C:\> netstat

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    192.168.1.124:3126     192.168.0.2:netbios-ssn   ESTABLISHED
  TCP    192.168.1.124:3158     207.138.126.152:http      ESTABLISHED
  TCP    192.168.1.124:3159     207.138.126.169:http      ESTABLISHED
  TCP    192.168.1.124:3160     207.138.126.169:http      ESTABLISHED
  TCP    192.168.1.124:3161     sc.msn.com:http           ESTABLISHED
  TCP    192.168.1.124:3166     www.cisco.com:http        ESTABLISHED
(output omitted)
C:\>
```

1. O par de soquetes para um host com endereço IP 10.1.1.10 solicitando serviços Web de um servidor em 10.1.1.254 seria 10.1.1. 10:1099, 10.1.1. 254:80.
2. Os números de porta de aplicativos FTP, HTTP e TFTP são definidos no grupo de números de porta bem conhecido.
3. O comando **netstat** windows exibirá protocolos em uso, o endereço local e os números de porta, o endereço externo e os números de porta e o estado da conexão.

Processo de comunicação TCP

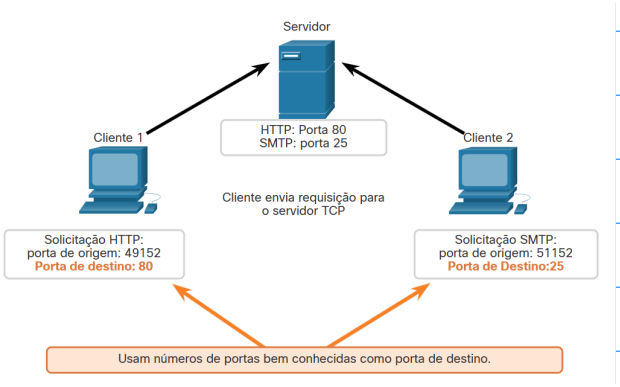
Cada App em execução em um servidor usa um número de porta, onde não é possível atribuir o mesmo número de porta a dois serviços no mesmo servidor

- Clientes Enviando Requisições TCP

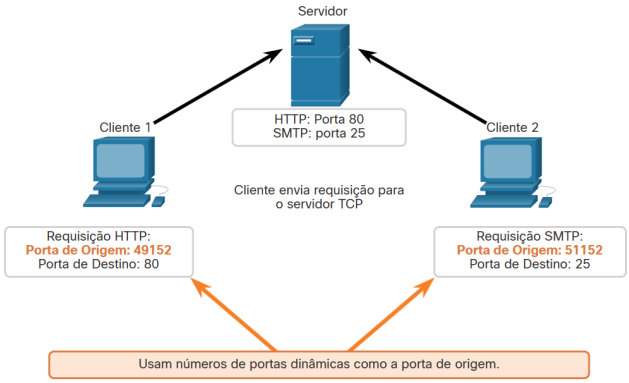


serviços Web Correio Eletrônico

- Porta de destino das Requisições

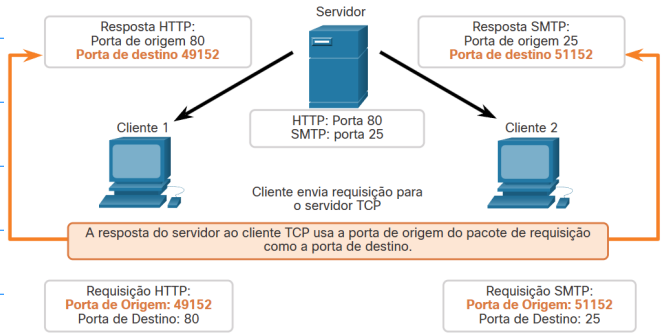


- Portas de origem das Requisições



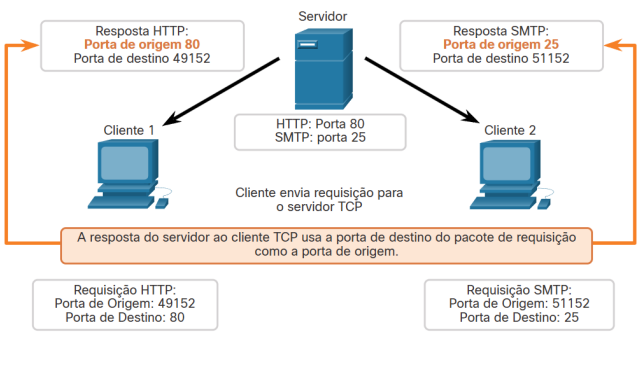
- Portas de destino das Respostas

→ Quando o servidor responde às solicitações do cliente.



- Portas de origem das Respostas

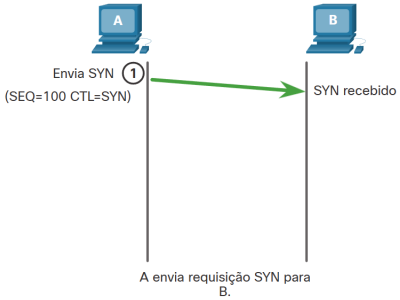
- Porta de destino original nas solicitações iniciais.



Estabelecimento de conexão TCP

- Etapa 1 SYN

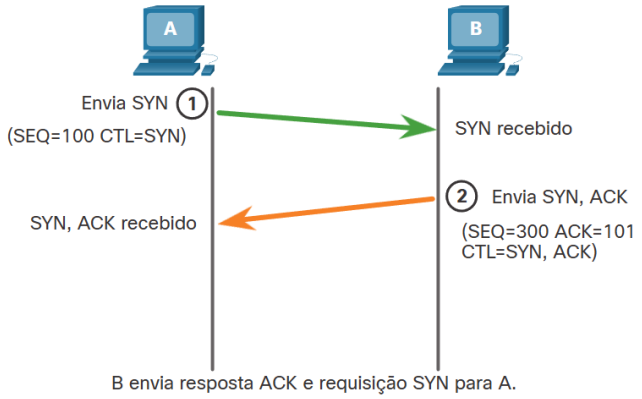
O cliente iniciador requisita uma sessão de comunicação cliente-servidor com o servidor.



- O handshake de 3 vias valida se o host destino está disponível.

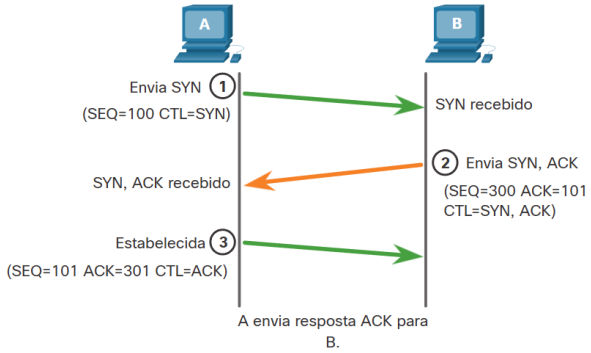
• Etapa 2 ACK e SYN

- 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.



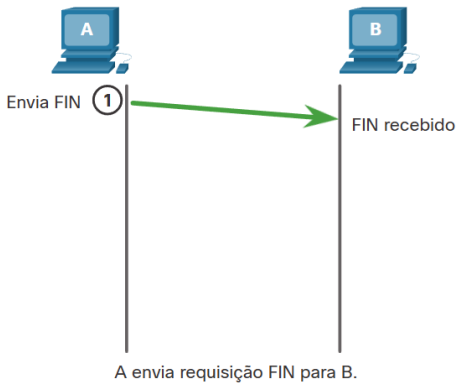
Encerramento da sessão

→ Para terminar uma conversação, 4 fluxos são necessários.

→ O Cliente ou o servidor podem iniciar o encerramento.

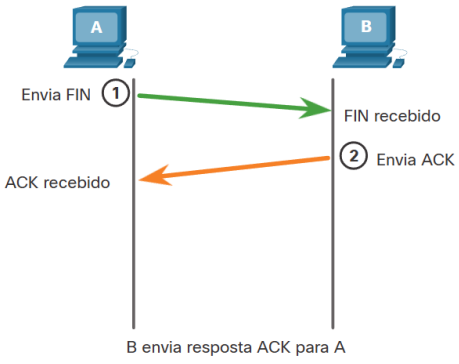
Etapa 1 - FIN

- Quando não se tem mais dados para enviar, é enviado um segmento com um flag FIN ligado.



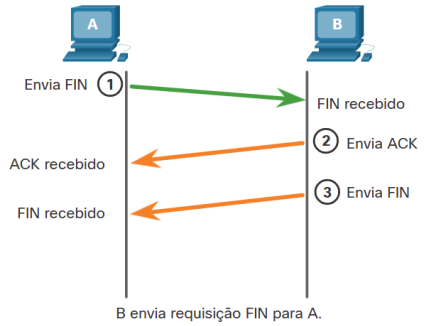
Etapa 2 - ACK

- O servidor envia um ACK para confirmar o recebimento de FIN para encerrar a sessão



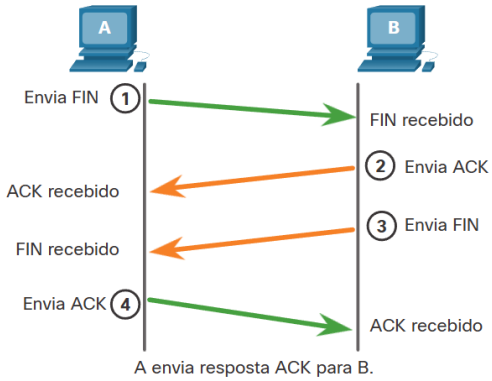
- Etapa 3 · FIN

- O servidor envia um FIN ao cliente para encerrar a sessão (servidor - cliente)



- Etapa 4 · ACK

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



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

Os seis bits de controle sinalizadores são os seguintes:

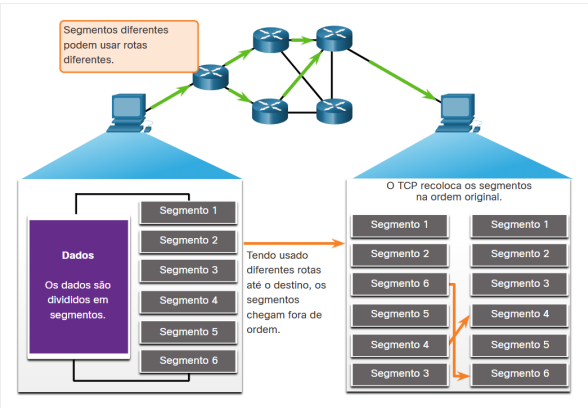
- **URG** - Campo de ponteiro urgente significativo.
- **ACK** - Indicador de confirmação usado no estabelecimento de conexão e encerramento de sessão.
- **PSH** - Função Push.
- **RST** - Redefina a conexão quando ocorrer um erro ou tempo limite.
- **SYN** - Sincronizar números de sequência usados no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

1. A porta de destino é a porta bem conhecida do Simple Mail Transport Protocol, que é 25. Esta é a porta em que o servidor de e-mail estará escutando. A porta de origem é selecionada dinamicamente pelo cliente solicitante e pode ser 49152.
2. O handshake de três vias consiste em três trocas de mensagens com os seguintes sinalizadores de bit de controle: SYN, SYN ACK e ACK.
3. Há quatro trocas para terminar ambas as sessões entre dois hosts. (1) Host A envia um FIN. (2) Host B envia um ACK. (3) Host B envia um FIN. (4) O host A envia uma confirmação.

Confiabilidade e controle de fluxo

* durante o estabelecimento de sessão, um número de sequência inicial (ISN) é definido
↳ valor inicial dos bytes que são transmitidos.

Os Segmentos TCP São Reordenados no Destino



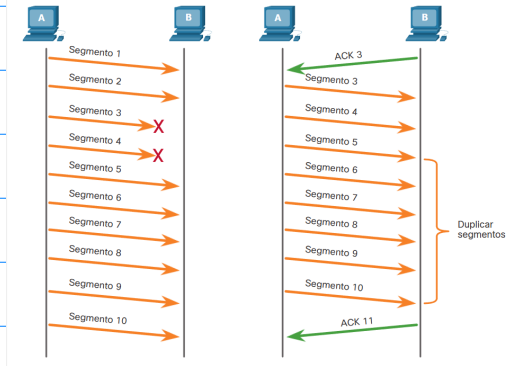
→ Perda de dados

o O número de sequência (SEQ) e o número de confirmação (ACK) são usados para confirmar o recebimento.

SEQ → 1º byte de dados.

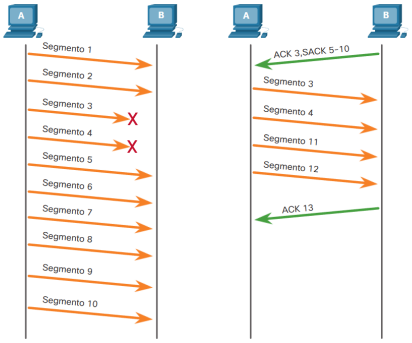
ACK → Indica o próximo byte que se espera receber.

Funcionamento Antigamente



vs

Funcionamento Atual



o A confirmação só ocorre após o envio de todos
- Atrasos.
- Duplicação de segmentos.
- Receptor não sabe quais do range não chegaram

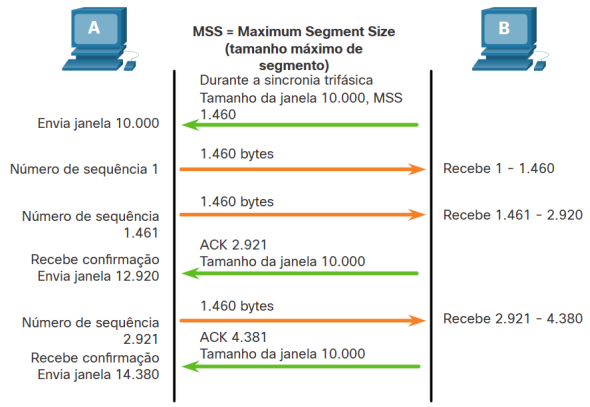
SACK

Caso não chegue algum é enviado um ACK solicitando apenas os que faltam.

➪ Tamanho da janela

- O Tcp também faz o controle de fluxo, para realizar isto, o cabeçalho TCP inclui um campo de 16 bits chamado janela.

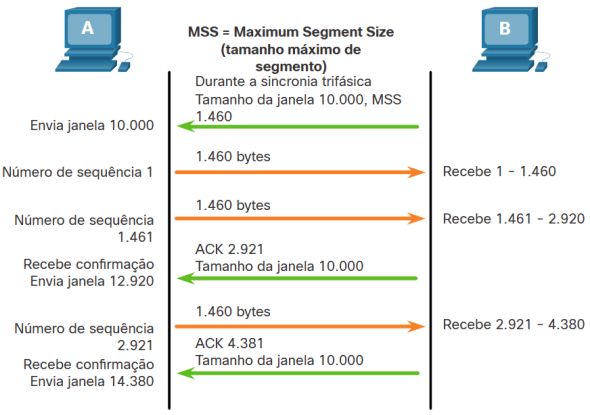
Exemplo:



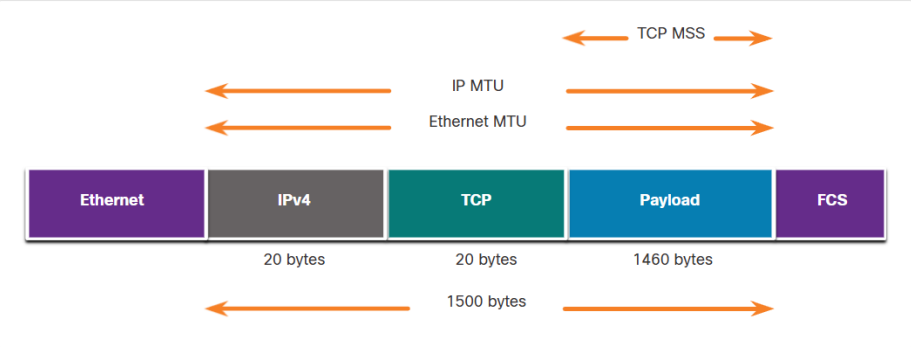
- O tamanho da janela determina o número de bits que podem ser enviados antes de exigir uma confirmação.

- Tamanho máximo do segmento (MSS)

- geralmente 1460 bytes
- Normalmente é incluído durante o handshake de 3 vias



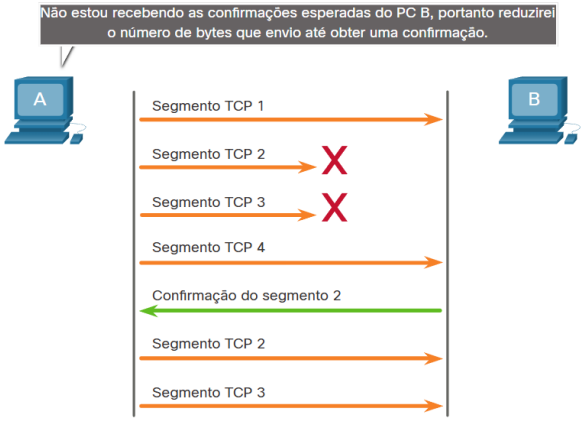
Um MSS comum é 1.460 bytes ao usar IPv4. Um host determina o valor do campo de MSS subtraindo os cabeçalhos de IP e de TCP da MTU (Maximum transmission unit, Unidade máxima de transmissão) da Ethernet. Em uma interface Ethernet, a MTU padrão é 1500 bytes. Subtraindo o cabeçalho IPv4 de 20 bytes e o cabeçalho TCP de 20 bytes, o tamanho padrão do MSS será 1460 bytes, conforme mostrado na figura.



o Prevenção de congestionamento

- Para evitar congestionamentos, o TCP emprega alguns algoritmos e temporizadores

- Como a origem percebe que os segmentos não estão recebendo confirmação, ele reduz o número de bits enviado antes de receber uma confirmação.

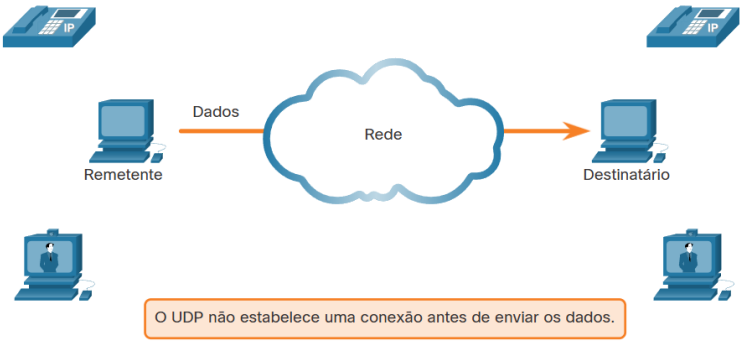


1. O campo de número de sequência é usado pelo host de destino para remontar segmentos na ordem original.
2. O campo Tamanho da janela é usado para fornecer controle de fluxo.
3. Quando um host de envio detecta congestionamento, ele reduz o número de bytes que envia antes de receber uma confirmação do host de destino.

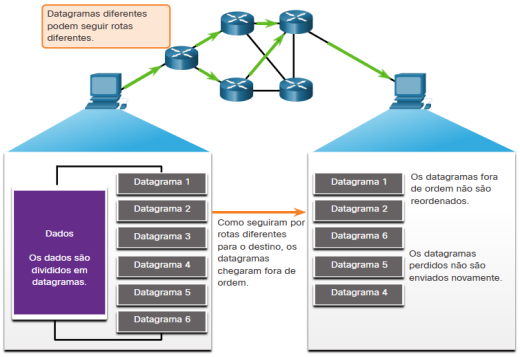
(*) Redução do número de bytes não confirmados que envia não o tamanho da janela

Comunicação UDP

- Rapida
- Não confiavel

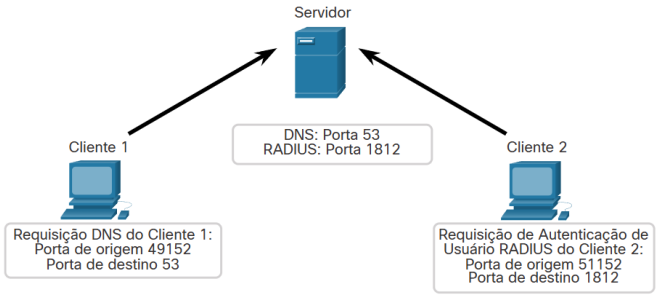


- O UDP Não restaura os números de sequencia, logo ele remonta os dados simplesmente na ordem que chegam

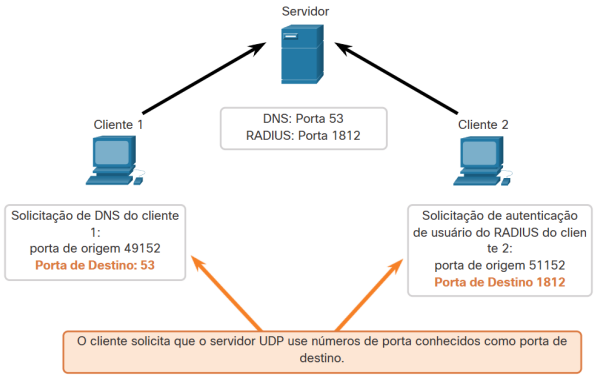


- Comunicação cliente - servidor
- Cliente solicita dados do servidor
- Cliente UDP seleciona um número de porta dinamicamente (porta origem)
- Porta destino é do servidor
- São usadas as portas em todos os datagramas na transação

- Cliente enviando solicitação UDP



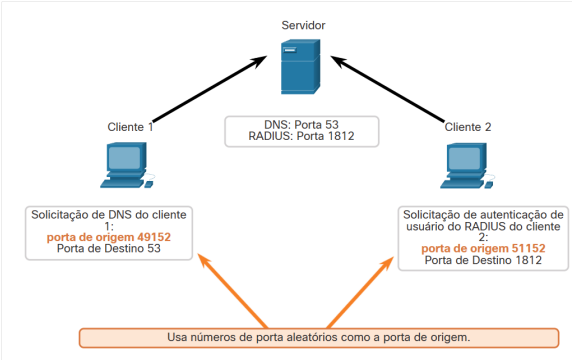
- Porta de destino de solicitação UDP



Portas de origem

Portas de origem da solicitação UDP

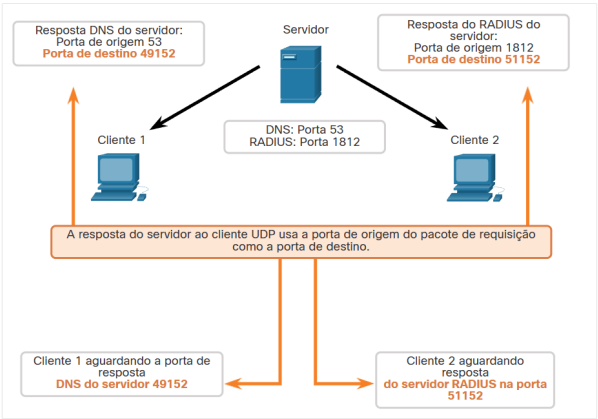
Quando o servidor responde às solicitações do cliente, ele reverte as portas de destino e de origem da solicitação inicial.



Destino de Resposta

Destino de resposta UDP

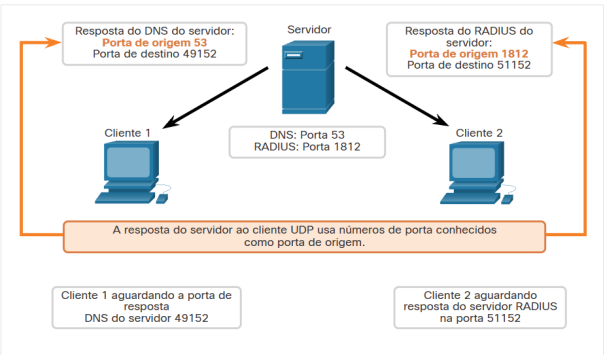
Na resposta do servidor à solicitação DNS agora é a porta de destino 49152 e a resposta de autenticação RADIUS é agora a porta de destino 51152.



Portas origem de resposta

Portas de origem de resposta UDP

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



1. O UDP é desejável para protocolos que fazem transações simples de solicitação e resposta devido à sua baixa sobrecarga.
2. O UDP remonta os dados que foram recebidos.
3. As portas de origem e destino válidas corretas para um host que solicita o serviço DNS é Origem: 49152, Destino: 53.