Os protocolo TCP e UDP

TCP (Transfer Control Protocol)

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. Se um pedido pelo correio estiver dividido em vários pacotes, um cliente poderá verificar on-line a sequência de recebimento do pedido.

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

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.

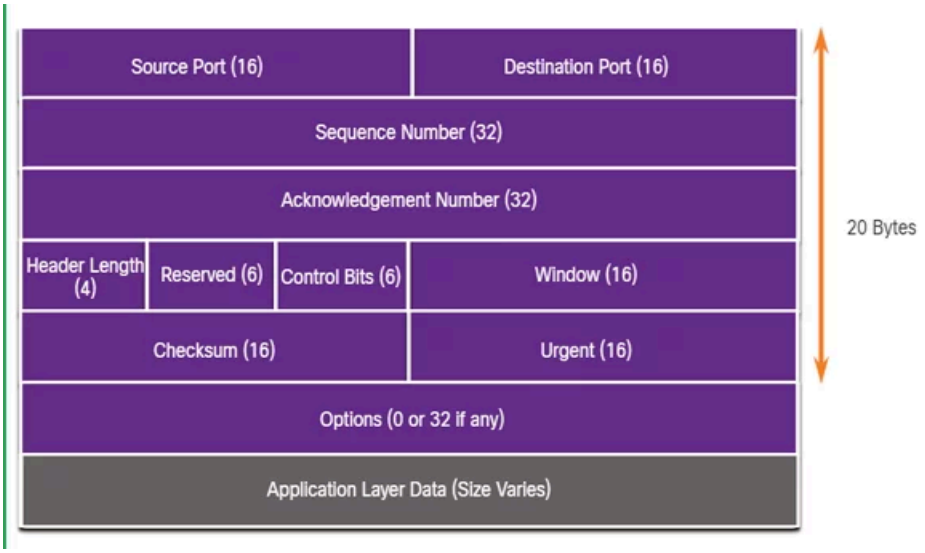
Suporta controle de fluxo

Os hosts de rede têm recursos limitados. 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.

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 de sobrecarga ao encapsular os dados da camada de aplicativo. A tabela abaixo identifica e descreve os dez campos de 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 por número de porta
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. O próximo byte esperado da fonte
Comprimento do cabeçalho	Um campo de 4 bit 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



UDP (User Datagram Protocol)

O UDP é um protocolo de camada de transporte que não fornece 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.

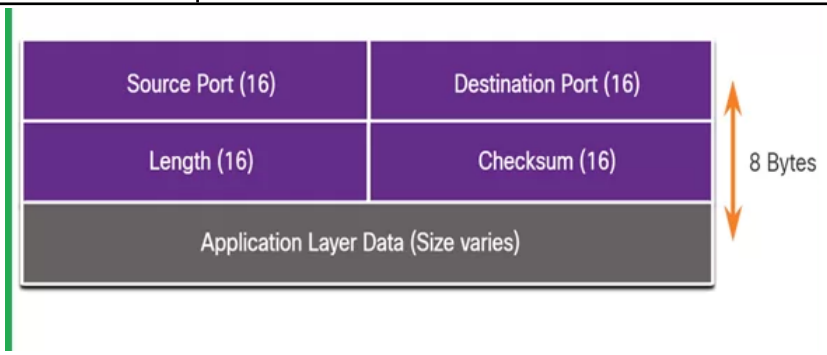
O UDP é um protocolo de transporte leve que oferece a mesma segmentação de dados e remontagem que o TCP, mas sem a confiabilidade e o controle de fluxo do TCP.

Os recursos UDP incluem o seguinte:

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

UDP é um protocolo sem estado, o que significa que nem o cliente nem o servidor rastreiam o estado da sessão de comunicação. 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. A tabela abaixo descreve e identifica 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 por número de porta
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



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

- **Aplicativos de vídeo e multimídia:** Esses aplicativos podem tolerar a perda de dados, mas requerem pouco ou nenhum atraso.
- **Solicitações simples e aplicativos de resposta:** Aplicativos com transações simples em que um host envia uma solicitação e pode ou não receber uma resposta.
- **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 protocolos de camada de transporte TCP e UDP usam números de porta para gerenciar várias conversas simultâneas. 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.

Sockets e portas de conexão de rede

As portas de 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.

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

Grupo de portas	Intervalo de números	Descrição
Portas comuns	0 a 1.023	Estes números de porta são reservados para serviços populares e aplicativos como navegadores da web, clientes de e-mail e acesso remoto de clientes Portas bem conhecidas definidas para aplicativos comuns de servidor permite para identificar facilmente o serviço associado
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 específico 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 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.

Alguns sistemas operacionais clientes podem usar números de porta registrados em vez de números de porta dinâmicos para atribuir portas de origem. tabela exibe alguns números de porta conhecidos comuns e seus aplicativos associados.

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 e TCP	Protocolo DNS
67	UDP	Protocolo de configuração Dinâmica de Hosts (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 - 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)

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 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.

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.
- **Etapa 2 - ACK e SYN:** O cliente iniciador requisita uma sessão de comunicação cliente-servidor com o servidor.
- **Etapa 3 - ACK:** O cliente iniciador confirma a sessão de comunicação de servidor-cliente.

Para fechar uma conexão, a 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.

- **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.
- **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.

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. As funções do handshake de três vias são:

- Estabelecer que o dispositivo de destino está presente na rede
- Verificar 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
- Informa ao dispositivo de destino que o cliente de origem pretende estabelecer uma sessão de comunicação nesse número de porta.

Os seis bits no campo Bits de Controle do cabeçalho do segmento TCP são também conhecidos como flags. Uma flag (sinalizador/bandeira) é um bit que é definido como ligado (1) ou desligado (0).

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

Durante o estabelecimento de uma sessão TCP, 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. 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.

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.

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 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 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.

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.

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.

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.

Quando ocorre um congestionamento em uma rede, isso resulta em pacotes sendo descartados pelo roteador sobrecarregado. 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.

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.

O servidor RADIUS (***Serviço de Usuário Discado por Autenticação Remota***) fornece serviços de autenticação, autorização e auditoria para gerenciar o acesso do usuário.

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.