

Arthur de Sá Braz de Matos

1) A porta na camada de transporte identifica processos específicos em um dispositivo, permitindo que vários serviços usem a rede simultaneamente, enquanto o endereço de rede (como o IP) na camada de rede identifica de forma única o dispositivo em uma rede, permitindo o roteamento dos dados até o destino correto. Assim, o IP localiza o dispositivo e a porta localiza o serviço dentro do dispositivo.

2) A camada de transporte dá sentido à pilha de protocolos porque é responsável por assegurar que os dados transmitidos entre aplicações em diferentes dispositivos cheguem corretamente, na ordem certa e, quando necessário, com confiabilidade, criando uma ponte lógica entre os processos de origem e destino, independentemente das particularidades da rede subjacente.

3) Uma aplicação do tipo one shot realiza uma única troca de mensagem entre cliente e servidor, como uma requisição simples com resposta imediata. O protocolo TCP pode ser utilizado, mas não é ideal, pois seu processo de estabelecimento e encerramento de conexão adiciona sobrecarga desnecessária. Já o UDP é mais adequado, pois é leve, rápido e sem conexão, ideal para comunicações breves como nas aplicações one shot.

4) Em uma aplicação cliente-servidor com TCP, o servidor executa as primitivas `socket()` para criar o soquete, `bind()` para associá-lo a uma porta, `listen()` para escutar conexões e `accept()` para aceitar conexões dos clientes. Já o cliente usa `socket()` para criar o soquete e `connect()` para se conectar ao servidor. Após a conexão estabelecida, ambos usam `send()` e `recv()` para enviar e receber dados, encerrando a comunicação com `close()`.

5) A transferência confiável de dados no TCP é garantida através do uso de controle de fluxo, controle de congestionamento e verificação de erros. Quando um cliente envia dados, ele os divide em segmentos numerados sequencialmente. Cada segmento contém um número de sequência (Sequence Number), que indica a posição do primeiro byte de dados no segmento. O destinatário, ao receber os dados, envia de volta um número de reconhecimento (Acknowledgement Number), que é o próximo número de sequência esperado, indicando que todos os bytes até aquele ponto foram recebidos corretamente.

O campo ACK é usado para sinalizar que o número de sequência foi corretamente recebido e que a próxima parte dos dados pode ser esperada. Se um segmento não for reconhecido dentro de um tempo específico, o remetente retransmitirá o segmento. O processo de confirmação com ACKs contínuos assegura que os dados sejam entregues de maneira confiável, sem perdas ou duplicações. O TCP também lida com a retransmissão de pacotes com base na ausência de ACKs.

6) O gerenciamento de conexões do TCP envolve o processo de estabelecimento e término de conexões. No estabelecimento, é realizado o 3-way handshake: primeiro, o cliente envia um segmento com o sinal SYN, indicando que deseja iniciar uma conexão e inclui um número de sequência. O servidor responde com um segmento SYN-ACK, sinalizando que está pronto para a conexão e confirmando o recebimento do pedido do cliente. Por fim, o cliente envia um ACK, confirmando a resposta do servidor e completando o estabelecimento da conexão. No término da conexão, é utilizado o 4-way handshake: o cliente envia um segmento FIN, solicitando o fechamento da conexão, e o servidor responde com um ACK, confirmando o recebimento da solicitação. O servidor, então, envia seu próprio FIN, indicando que também terminou sua comunicação, e o cliente responde com um ACK, finalizando o processo de encerramento da conexão. Esses sinais SYN, ACK e FIN garantem que a comunicação seja estabelecida e finalizada de forma ordenada e confiável.

7) O controle de fluxo do TCP é responsável por evitar que o remetente envie dados mais rapidamente do que o receptor consegue processar. Ele é implementado através da negociação do tamanho da janela, que limita a quantidade de dados que podem ser enviados antes de receber um reconhecimento (ACK). O campo Window Size no cabeçalho do segmento TCP especifica o número de bytes que o receptor está pronto para receber, ou seja, o espaço disponível no buffer de recepção. Quando o remetente envia dados, ele observa o valor da janela para garantir que não ultrapasse a capacidade do receptor.

8) Controle de fluxo é utilizado para evitar que o remetente envie dados mais rapidamente do que o receptor consegue processar, garantindo que o buffer de recepção do destinatário não transborde. Isso é feito através do campo Window Size no cabeçalho TCP, que indica a quantidade de dados que o receptor está pronto para aceitar.

Já o controle de congestionamento visa evitar que a rede se sobrecarregue com muitos pacotes, o que poderia causar perdas e atrasos. O TCP usa um algoritmo adaptativo que ajusta a taxa de envio com base no estado da rede.

9) O controle de congestionamento do TCP tem como objetivo evitar que a rede se sobrecarregue e para garantir uma transmissão eficiente de dados. Ele adapta a taxa de envio de pacotes com base nas condições da rede, evitando congestionamentos e perdas de pacotes. Ele começa com uma taxa de transmissão baixa (chamada slow start), aumentando progressivamente até que a rede mostre sinais de congestionamento, como a perda de pacotes. Quando isso ocorre, o TCP reduz a taxa de envio e recomeça o processo de aumento gradual, buscando encontrar um equilíbrio ideal entre a utilização da largura de banda e o não congestionamento da rede.

10) A figura mostra a variação do tamanho da janela de congestionamento (congestion window) de um protocolo TCP durante uma conexão, ilustrando os comportamentos de crescimento exponencial

(slow start), crescimento linear (congestion avoidance) e resposta a congestionamento. Inicialmente, a janela cresce exponencialmente até atingir o valor de limiar (threshold), quando então passa a crescer linearmente. Ao ocorrer um timeout por perda de pacote, o tamanho da janela é drasticamente reduzido para 1 KB e o limiar é ajustado para metade da janela anterior. A partir daí, o crescimento volta a ser exponencial até o novo limiar, retomando o comportamento linear. Esse mecanismo busca equilibrar desempenho e controle de congestionamento na rede.