

Exercícios Unidade 03

Rafael Amauri Diniz Augusto - 651047

Q.1 - Faça um paralelo entre a porta para a camada de transporte e o endereço de rede para a camada de rede.

É na camada de transporte que a porta de origem e a porta de destino são conectadas ao datagrama, da mesma forma que é na camada de rede que o IP de origem e o IP de destino são conectados ao pacote

Q.2 - Por que a camada de transporte dá sentido à pilha de protocolos?

Porque a camada de rede só é responsável pelo endereço IP de origem e destino, e não é responsável pelas portas.

Q.3 - O que significa uma aplicação do tipo one shot? Os protocolos TCP pode ser utilizado na camada de transporte para esse tipo de aplicação? E o UDP? Justifique suas respostas.

Aplicações one-shot são aplicações que não precisam de receber nenhum tipo de confirmação de retorno, apenas de enviar o pacote sem qualquer tipo de confirmação. Conexões lossy como streaming de vídeo e chamadas por voip são exemplos de aplicações one-shot. O TCP não pode ser usado para esse tipo de aplicação, pois o TCP obrigatoriamente requer uma confirmação de conexão e recebimento de pacotes. O UDP pode ser usado para esse tipo de aplicação, pois ele não requer qualquer tipo de confirmação de recebimento de dados.

Q.4 - Explique como executamos as primitivas de soquetes para o TCP em uma aplicação do tipo cliente-servidor

Servidor:

SOCKET: Especifica o formato do endereço, tipo de serviço (e.g., fluxo de bytes confiável) e protocolo.

BIND: Atribui um endereço de rede ao soquete. \footnote{Observação: a primitiva socket não atribui diretamente um endereço porque alguns processos utilizam endereços anteriormente definidos}

LISTEN: Aloca espaço para a fila de chamadas recebidas.

ACCEPT: Bloqueia o responsável pela chamada até que o servidor receba uma tentativa de conexão. Quando o servidor recebe uma conexão, ele pode desviá-la para outro processo (thread) e voltar à espera por uma nova conexão.

Cliente:

SOCKET: Como no servidor, esta primitiva especifica o formato do endereço, tipo de serviço (e.g., fluxo de bytes confiável) e protocolo. \footnote{A primitiva BIND não é necessária no cliente, pois seu endereço é irrelevante para o servidor.}

CONNECT: Tenta estabelecer uma conexão com o servidor.

SEND / RECEIVE: As partes devem ter uma política para definir de quem é a vez de enviar dados.

CLOSE: (Desconexão simétrica) cada parte faz sua desconexão para informar que não enviará mais dados.

Q.5 - Como os campos Sequence e Acknowledgement numbers atuam na transferência confiável de dados do TCP

O sequence number indica o número do primeiro byte no segmento de dados e o acknowledgment number especifica o número do próximo byte esperado. Com esses 2 campos, sabemos quem é o primeiro byte e qual é o byte seguinte, dessa forma sabendo a ordem exata deles e assim tendo uma transferência confiável.

Q.6 - Explique como os campos SYN, ACK e FIN funcionam no estabelecimento e no término de conexões TCP

Campo SYN: Estabelece conexões

Campo ACK: Indica se o campo acknowledgment number é válido

Campo FIN: Finaliza uma conexão

Q.7 - Descreva o controle de fluxo no TCP

O controle de fluxo tem como objetivo evitar que o emissor esgote a capacidade de recebimento do receptor. No estabelecimento da conexão, é informado um valor para o windowSize, que indica quantos bytes podem ser enviados.

Q.8 - Descreva o controle de congestionamento no TCP

A fim de o emissor não esgotar a capacidade de transporte da rede, é estabelecida uma janela de congestionamento. No início da conexão, a janela de congestionamento começa vazia e o emissor vai mandando uma quantidade crescente de pacotes até a rede avisar que não consegue receber uma quantidade maior daquele emissor. Depois disso, o emissor envia só aquela quantidade fixa de pacotes por tempo. Caso a rede detecte novos emissores e veja um risco de congestionamento, ela pede para os emissores limparem a sua janela de congestionamento e achem a sua nova capacidade de envio para cada emissor.