

Exercícios

Exercício (1)

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

↳ A camada de rede estabelece conexão entre endereços / Pd rede
(por meio de roteamento)

↳ A camada de transporte estabelece conexão
entre serviços / aplicação, identificadas por seus respectivos portos.

Exercício (2)

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

↳ Pois é ela que define o funcionamento das demais camadas, estabelecendo uma conexão entre diferentes máquinas e garantindo a confiabilidade durante a transmissão (fazendo controle de fluxo, retransmissão de pacote, etc.). Além disso, é ela que organiza o fluxo de informações na rede.

Exercício (3)

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

Aplicação One-shot é aquela que realiza uma troca de mensagens rápida e breve, sem necessidade de manter a conexão persistente. Neste caso, o UDP é mais adequado para esta apl., já que ele é + simples, não precisa de conexão e é rápido. O TCP ainda pode ser usado, mas é muito complexo e persistente para o que é solicitado.

Exercício (4)

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

Servidor:

- ↳ criamos o socket (socket.socket) → server.bind
- ↳ vinculamos o socket a um endereço IP
- ↳ server.listen() → conectar e escutar por conexões
- ↳ server.accept() → aceita conexões
- ↳ client.socket.recv() e client.socket.send() p/ enviar e receber os dados.

Cliente:

- ↳ criamos socket (socket.socket)
- ↳ client.connect → conecta-se ao servidor
- ↳ client.send() e client.recv() p/ enviar e receber os dados.

Exercício (5)

- Descreva a transferência confiável de dados do TCP e mostre como os campos *Sequence* e *Acknowledgement Numbers* e ACK atuam nesse caso

↳ o TCP garante uma entrega confiável de dados pois sempre confirma a mensagem enviada, por meio de números que indicam cada byte transmitido (Sequence Number) e números de confirmação que indicam o próximo byte esperado pelo receptor (ACK).

Exercício (6)

- Descreva o gerenciamento de conexões do TCP e mostre como os campos SYN, ACK e FIN funcionam no estabelecimento e término de conexões TCP

(nº de sequência e confirmação) → Emissor inicia enviando um $SYN=1$ e $ACK=0$ (o receptor indica deseja entrar em sincronização)

(ACK e nº de sequência) → Receptor responde c/ $SYN=L$ e $ACK=L$, aloando buffers e especificando o byte inicial de sequência

próx. byte que deve receber, esperar receber todos os anteriores) → Emissor recebe o SYNACK e envia outro ACK

FIN é usada quando um deseja finalizar a comunicação do seu lado (mas o outro ainda pode enviar)

Exercício (7)

- Descreva o controle de fluxo do TCP e mostre como o campo Window Size funciona nesse caso

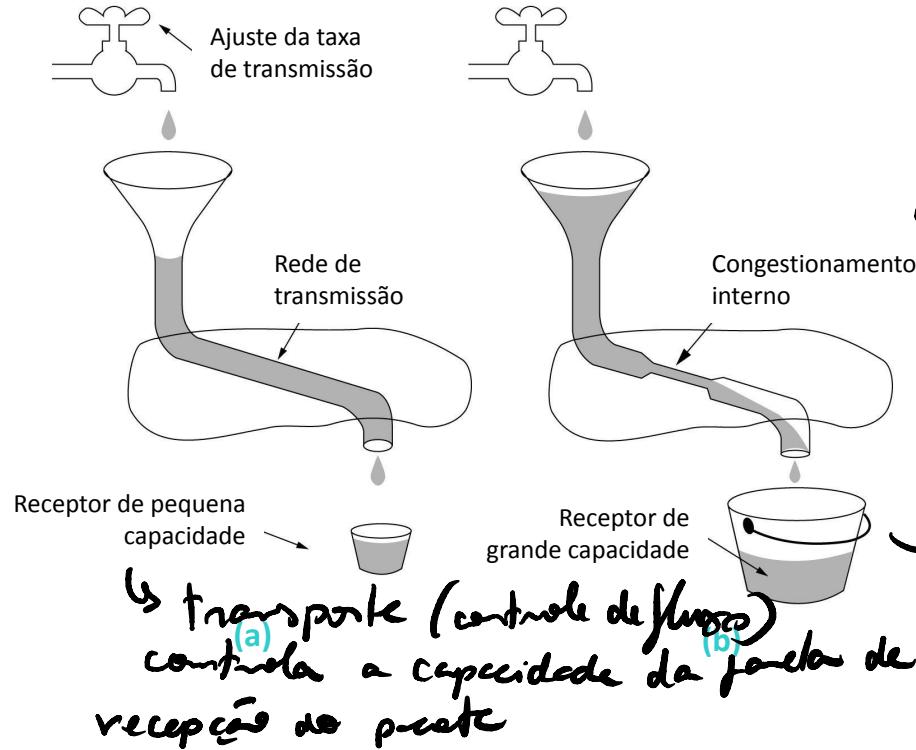
↳ O tamanho da janela é a qtd de dados que um dispositivo pode receber antes que envie uma resposta de confirmação ACK.

Assim, ele informa ao envar o tam. de janela disponível no buffer de recebimento, e o envar ajusta a qtd de dados enviada.

Uma janela pequena pode gerar confirmações frequentes

Exercício (8)

- Explique a figura abaixo no contexto dos controles de fluxo e de congestionamento do TCP



o TCP limita o envio ao menor dos dois (garante que não há sobrelocação de receptores nem da rede)

rede (controle de congestionamento) controla a capacidade da rede no geral p/ não haver gatilhos

Exercício (9)

- Descreva o controle de congestionamento do TCP e mostre como funciona o algoritmo de iniciação lenta

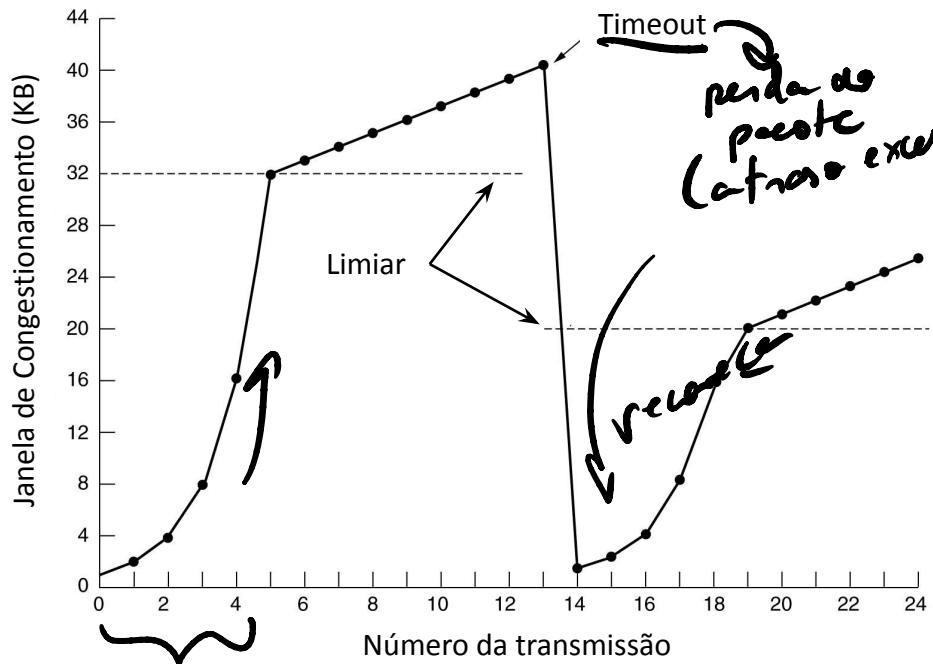
→ o controle de congestionamento usa a iniciação lenta p/ melhorar o fluxo de dados da rede (aumentá-lo) e controlar a taxa de transmissão.

→ a janela de congestionamento se inicia com um segmento

→ p/ cada ACK, ela é dobrada até atingir o limite

Exercício (10)

- A figura abaixo mostra o tamanho da janela de congestionamento de um nó à medida que ele efetua transmissões em uma mesma conexão, explique a figura e os valores da janela de congestionamento



• Da 4 → tamanho da janela cresce a cada transmissão (ACK recebido)
• Isto acontece até chegar ao limite, em que o crescimento para mais lento (+ lento), o que evita que o envio de dados corra rápido demais e sobrecarregue a rede

Exercício (11)

- Abra o Wireshark e faça a captura de pacotes. Em seguida, abra o www.google.com e faça algumas pesquisas. Volte para o Wireshark, utilize um dos dois filtros abaixo e explique os campos SEQ e ACK:

- ($\text{ipv6.addr} == \text{seu-ipv6} \mid\mid \text{ipv6.dst} == \text{seu-ipv6}$) $\&\&$ ($\text{tcp.dstport} == 80 \mid\mid \text{tcp.srcport} == 80$)
- ($\text{ip.addr} == \text{seu-ipv4} \mid\mid \text{ip.dst} == \text{seu-ipv4}$) $\&\&$ ($\text{tcp.dstport} == 80 \mid\mid \text{tcp.srcport} == 80$)

*mº do Lº byte de
dados no segmento
TCP atual*

*próximo nº de
sequência que o
receptor espera
receber.*