

Lista - Camada de Transporte
Redes de Computadores
Instituto de Ciência e Tecnologia - ICT
Universidade Federal de São Paulo - UNIFESP
2o semestre de 2021

Observações:

- Lista de exercícios para fixação de conteúdo.
- Livro texto onde se encontram as questões:
 - KUROSE, James F.; Ross, Keith W.. Redes de computadores e a internet: uma abordagem top- down. **5.ed.** São Paulo: Addison-Wesley, 2010.
- A resolução da lista é INDIVIDUAL.
- Para cada questão, crie uma caixa de texto e a utilize para responder.
- As respostas são dissertativas. Pondere completude e objetividade nas respostas.

Nome:

Thiago Henrique Leite da Silva - 139920

Exercícios de Fixação (pág. 215):

3, 4, 6, 7, 8, 9, 10, 11, 14, 15, 17

Problemas (pag. 216)

5, 6, 7, 14, 21, 33, 39

Questões Dissertativas (pag. 224)

2, 4

Pág. 215

Ex. 03

No caminho contrário, as portas também se invertem, logo, a porta fonte é a “y” e a porta de destino é a “x”.

Ex. 04

O desenvolvedor pode optar por rodar sua aplicação sobre UDP por conta do ganho de velocidade e eficiência, porém o mesmo não poderia estar preocupado com perdas de pacotes ou com o reenvio dos mesmos em caso de falha, já que o UDP não retorna uma mensagem a origem garantindo que os pacotes chegaram. Um exemplo de uma aplicação em que isso seria viável são as relacionadas a telefonia ou comunicação por vídeo chamada.

Ex. 06

É possível, porém não trivial. Diferentemente do TCP, para garantir esta confiabilidade no UDP, seria necessário a implementação própria de ferramentas para confirmar o sucesso na transmissão dos dados para o protocolo da camada de aplicação pelos

desenvolvedores da mesma. Ou seja, é possível, porém bastante trabalhoso, já que precisamos pensar em formas de tornar as transmissões confiáveis e seguras sem perder as principais vantagens do protocolo.

Ex. 07

No caso especificado na questão, de fato os dois segmentos serão encaminhados ao mesmo socket, e o processo C conseguirá saber que eles vieram de dois computadores diferentes através de seus respectivos IP's, que serão diferentes.

Ex. 08

Não, as conexões de sockets ocorrem separadamente e são identificadas através do IP e porta de origem, e IP e porta de destino. Neste exemplo, quando o computador C receber um datagrama IP, ele definirá, de acordo com a identificação citada, para qual socket encaminhar o trabalho do segmento TCP. Logo, os computadores A e B passarão por sockets distintos. Os dois sockets terão a porta 80 como destino, porém com diferentes IP de origem.

Ex. 09

Para diferenciar os novos pacotes dos pacotes que estão sendo retransmitidos.

Ex. 10

Necessitamos do temporizador para que o pacote mais antigo que ainda não foi reconhecido, sendo assim, caso o temporizador exceda o tempo limite, ele irá retransmitir todos os pacotes que não foram reconhecidos. Assim conseguimos diminuir o impacto das perdas, já que conseguimos retransmitir estes pacotes.

Ex. 11

Sim, ainda seria necessário. O que acontece é que se o emissor conhece o atraso de viagem de ida e volta entre ele e o receptor, ele sabe quanto tempo leva exatamente para saber se um pacote foi perdido ou não, portanto, o temporizador serviria para sabermos se o retorno levou mais tempo que o previsto, e por ser um tempo constante, teríamos a certeza de que o pacote se perdeu; além disso, poderíamos já retransmiti-lo nesses casos.

Ex. 14

- a) F
- b) F
- c) V
- d) F
- e) V
- f) F
- g) F

Ex. 15

- a) A quantidade de dados do primeiro segmento é de 20 bytes.
- b) O fato do primeiro segmento ter sido perdido, não faz com que o hospedeiro B deixe de esperá-lo, por conta disso, o número de reconhecimento, neste caso, é 90, já que o primeiro segmento tem número de sequência 90.

Ex. 17

A velocidade de transmissão que o TCP irá tentar disponibilizar é metade para cada uma das conexões, que nesse caso seria a metade da velocidade R bps, porém não necessariamente essa divisão será exatamente igualitária para ambos os lados.

Pág. 216

Ex. 05

Absolutamente certo não, por conta do método de verificação que admite falhas do UDP. O método consiste em invertermos os bits dos pacotes recebidos e calcular a soma dos mesmos, sendo assim, caso o conjunto de bits X e o conjunto de bits Y, dado X diferente de Y, possuam a mesma soma dos bits invertidos, o erro passará despercebido pela validação.

Ex. 06

Um exemplo de um problema que pode ocorrer para levar a operação entre remetente e destinatário em um estado de travamento, é quando temos um pacote corrompido no envio do ACK do receptor para o emissor, pois dessa forma, quando o ACK chega no emissor, o emissor verá que ele está corrompido e irá enviar um novo pacote, porém o receptor que não está esperando pelo mesmo pacote, ficará respondendo com um NAK, e teremos esse estado de deadlock, nesse looping infinito.

Ex. 07

Pois no rdt3.0 eles não serão necessários para o reconhecimento da duplicata. Nessa abordagem, o remetente irá aguardar um certo período de tempo pelo ACK em trânsito, logo serão necessários os temporizadores. Quando ocorre o timeout, ocorre a retransmissão se nenhum ACK foi recebido neste meio tempo. Isso poderá causar a duplicata, caso o ACK apenas esteja atrasado e não tenha sido perdido, porém o ACK duplicado será apenas desprezado pelo remetente.

Ex. 14

Primeiramente vamos calcular o tempo de transmissão de um pacote de acordo com a página 164 do livro, que será $1500 \text{ bytes} / 10^9 \text{ bits} = 12000 \text{ bits} / 10^9 \text{ bits} = 0,000012 = 12 \text{ microssegundos}$.

Logo, $t = L/R = 12 \text{ microssegundos}$.

A utilização do remetente é dada por $(L/R) / (RTT + L/R)$.

Para que ela ultrapasse os 95%, o resultado da conta deve ser superior a 0,95.

Sendo assim, $0,000012 / \text{RTT} + 0,000012 > 0,95$

RTT deve ser próximo de zero e a vazão efetiva acima de 1425 bytes por segundo.

Ex. 33

Na minha opinião, acredito que os projetistas do TCP, através de testes e estudo de caso, notaram que esperar por 3 ACK's duplicados geraria o melhor desempenho em relação a retransmissão de dados. Essa informação foi de fato comprovado por um cientista da computação norte-americano, que chegou ao resultado de que com 3 ACK's, serão deixados de retransmitir segmentos em uma escala de 2 e meio para 1.

Ex. 39

Para tornar o protocolo mais seguro, já que apenas tratar os casos de duplicação do intervalo de esgotamento de temporização não resolveria o problema do congestionamento, logo, é necessário esse controle de congestionamento com as janelas, que serve para limitar a taxa de envio dos pacotes para o remetente TCP. Em resumo, esses dois mecanismos se completam.

Pág. 224

Ex. 02

Para tornar a internet justa, devemos implementar mecanismos que tornem isso possível, de modo que haja uma igualdade de recursos, tempo e processamento para todas as conexões paralelas da aplicação cliente-servidor.

Ex. 04

O grande problema se todas as aplicações adotassem as múltiplas conexões é que o TCP é um protocolo justo, ou seja, ele divide a largura de banda do enlace entre todas as conexões de forma o mais igual possível. Portanto, é trivial pensarmos que quanto maior for o número de conexões, menor será a banda de cada conexão, o que faz com que esta ideia deixe de ser interessante.