

IF678 – Infraestrutura de Comunicação (Turma EC)
3ª Avaliação Escrita (22/09/2022)
Prof. José Augusto Suruagy Monteiro

Esta avaliação contém 4 páginas e 9 questões.
À exceção das questões dissertativas (nesta primeira página), responda às demais nas próprias folhas de questões.

- 1) (1 ponto) **Multiplexação e demultiplexação.** Explique como funciona a multiplexação e a demultiplexação, nos seguintes casos:
 - a) (0,4 ponto) no UDP – No UDP para a identificação do processo destino basta o endereço IP e a porta destino que é único para cada socket.
 - b) (0,6 ponto) no TCP – No TCP todos os sockets associados a um mesmo serviço, têm o mesmo IP e mesmo número de porta (80, por exemplo para os servidores Web). Desse modo, para fazer a demultiplexação é preciso contar também com o endereço IP e o número de porta da origem.

- 2) (1 ponto) **Mecanismos para a transferência confiável.** Explique o que são e para que servem os seguintes mecanismos para a transferência confiável de dados:
 - a) (0,2 ponto) Soma de verificação (checksum) – somam-se os dados que estão sendo transmitidos e anexa-se ao final o complemento desta soma. A ideia é que erros introduzidos na transferência entre a origem e o destino gerem uma soma distinta, identificando-se assim o erro. O método não é infalível e há casos em que erros não são detectados.
 - b) (0,2 ponto) Reconhecimento (ACK) – mensagem enviada no sentido contrário ao do fluxo informando o que foi recebido (ou o próximo pacote ou byte esperado). Serve justamente para confirmar o recebimento sinalizando a não necessidade de retransmissão do dado enviado anteriormente.
 - c) (0,2 ponto) Retransmissão – consiste no reenvio de dados enviados anteriormente, mas que foram perdidos ou chegaram corrompidos. É essencial para a recuperação de erros.
 - d) (0,2 ponto) Número de sequência – indica o número de sequência do pacote (ou do primeiro byte transmitido). Serve para reconstruir a sequência original, em caso de reordenamento, e identificação de duplicatas.
 - e) (0,2 ponto) Temporizador – contador regressivo configurado para o tempo esperado para o recebimento do reconhecimento. Caso estoure, sem a recepção do reconhecimento, é realizada a retransmissão do primeiro pacote pendente ou todos a partir do pendente.

- 3) (1,5 ponto) **Protocolos com paralelismo (pipeline):** Em que consiste, quais as vantagens e as desvantagens de cada um dos tipos abaixo de protocolos com paralelismo:
 - (a) (0,75 ponto) Volte-a-N (GBN – Go-Back N) – O transmissor assume que o receptor aceita os pacotes apenas na sequência correta, descartando os pacotes recebidos fora de ordem ou após um pacote com erro, isso apesar do receptor ter a opção de salvar os pacotes recebidos fora de ordem. Utiliza um único temporizador e ACKs cumulativos (reconhecendo todos os pacotes anteriores ao indicado). VANTAGEM: simplicidade (temporizador único e retransmissão de todos os pacotes a partir do perdido). DESVANTAGEM: desperdício de banda com a retransmissão de pacotes que potencialmente já foram recebidos corretamente.

(b) (0,75 ponto) Retransmissão Seletiva (SR – *Selective Retransmission*) – no esquema clássico, os pacotes são recebidos mesmo fora de ordem e cada pacote possui o seu próprio temporizador e usa ACK individual (não cumulativo). Em caso de estouro do temporizador associado a um determinado pacote, apenas o pacote correspondente é retransmitido. VANTAGEM: potencialmente retransmite apenas os pacotes perdidos (ou corrompidos). DESVANTAGENS: complexidade (número de temporizadores); como o ACK é não cumulativo, a perda de um ACK não é compensada com a chegada de um próximo ACK (para outro pacote), desse modo há a possibilidade de retransmissão desnecessária de pacotes.

4) (0,5 ponto) **Controle de fluxo do TCP:** Explique para que serve e como funciona o controle de fluxo no TCP. – O controle de fluxo tem a finalidade de impedir que um transmissor mais rápido inunde um receptor mais lento, fazendo com que estoure o seu buffer de recepção. No TCP, ao enviar um ACK informando o número de sequência do próximo byte esperado, o receptor informa ainda no campo “rwind” (janela de recepção) do cabeçalho do segmento, quantos bytes a partir do esperado estão livres no buffer de recepção.

5) (1 ponto) **Abordagens para o Controle de Congestionamento:**

a) (0,5 ponto) Explique com suas palavras o que é e como funciona a abordagem fim-a-fim para o controle de congestionamento. – Na abordagem fim-a-fim o congestionamento é inferido pelas extremidades a partir da monitoração das perdas e/ou dos atrasos, sem nenhuma notificação explícita por parte do núcleo da rede. Obviamente o controle em si consiste na redução da janela de transmissão, reduzindo a quantidade de pacotes em trânsito.

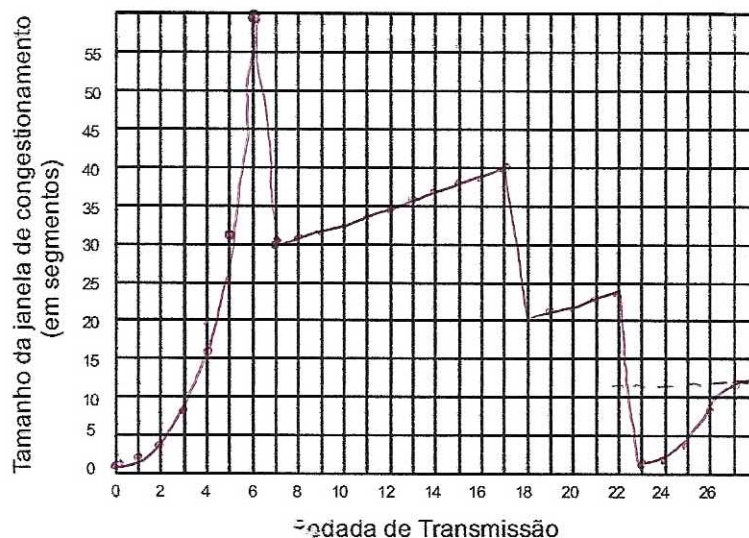
b) (0,5 ponto) Explique com suas palavras o que é e como funciona a abordagem assistida pela rede para o controle de congestionamento. – Nessa abordagem, o núcleo da rede informa o estado de seus buffers, indicando a formação de congestionamento, permitindo que o transmissor reduza o seu tráfego antes que ocorram perdas, por exemplo.

6) (1 pontos) **Controle de Congestionamento TCP baseado em atraso.** Apresente com suas palavras como poderia ser implementado controle de congestionamento baseado em atrasos para o TCP. – O TCP já monitora constantemente o RTT dos pacotes transmitidos de modo a ajustar apropriadamente o seu temporizador. Neste esquema pode-se registrar o menor RTT observado como sendo o obtido em uma situação de rede não congestionada. Desse modo, se o RTT observado estiver próximo a esse mínimo, o transmissor pode aumentar (em geral incrementalmente) a sua janela de congestionamento. Por outro lado, quando o RTT observado estiver muito acima do RTT mínimo, é uma indicação de congestionamento em formação, e o transmissor deveria reduzir a sua janela de congestionamento.

- 7) (1 ponto) **Controle de congestionamento:** na figura abaixo queremos representar a evolução do tamanho da janela de congestionamento do TCP como uma função das rodadas de transmissão. Assuma um limiar (*threshold*) inicial de 60 segmentos.

Desenhe abaixo a evolução da janela, até a 28ª rodada, assumindo:

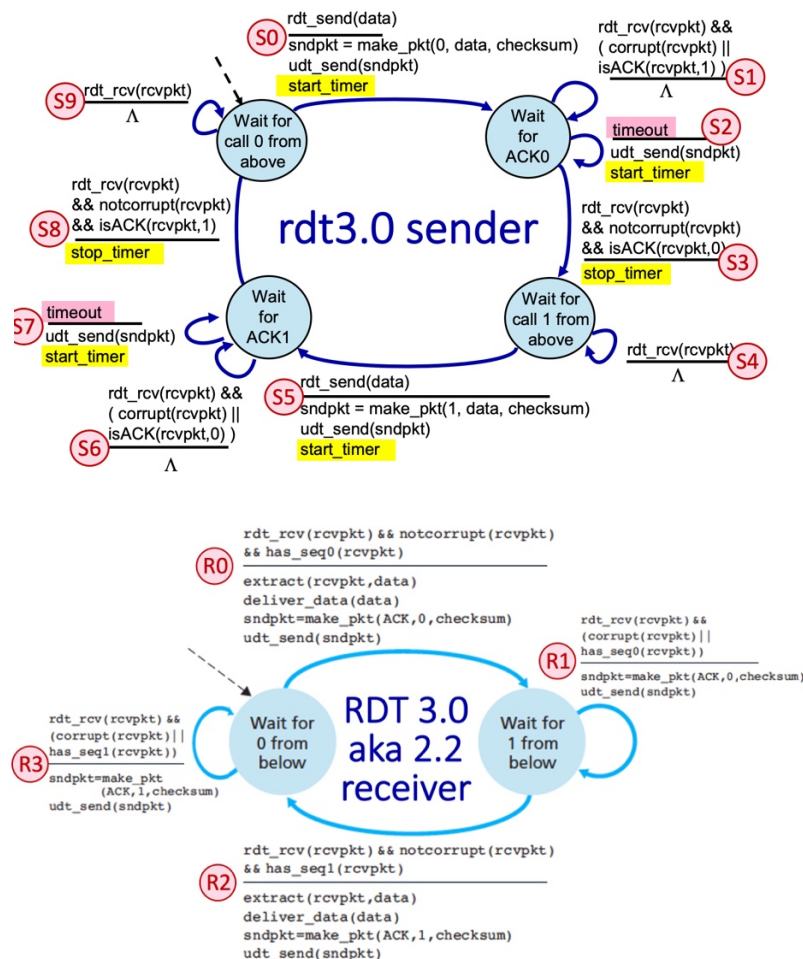
- Uso do TCP Reno (sem considerar os detalhes da recuperação rápida);
- Iniciamos com a partida lenta na rodada 0;
- Identificação de perda por 3 ACKs duplicados na 6ª rodada;
- Identificação de perda por 3 ACKs duplicados na 17ª rodada.
- Identificação de perda por estouro do temporizador na 22ª rodada.



- 8) (2 ponto) Assuma que utilizemos o TCP em conjunto com o HTTP. Inicialmente o cliente solicita o estabelecimento de uma conexão TCP. Uma vez estabelecida a conexão, o cliente envia para o servidor uma solicitação (GET) de recuperação de dados de 300 bytes, sendo que o arquivo HTML solicitado possui um comprimento de 2000 bytes e contém referências a duas imagens de baixa resolução, a primeira de 2500 bytes e a segunda de 1800 bytes. Considere ainda que as mensagens GET de solicitação das imagens sejam ambas de 420 bytes e que o tamanho máximo do segmento (MSS) seja de 1460 bytes.

No diagrama de tempo abaixo, complete os valores dos campos em aberto para cada um dos segmentos, onde *seq* é o valor do campo do número de sequência, *ack* é o valor do campo de reconhecimento (*acknowledgment*) e *len* é o comprimento do segmento. Indique claramente quando os dados são entregues à camada superior nos dois sistemas finais, assim como os instantes em que o temporizador é ligado (LT), religado (RT) e desligado (DT).

- 9) (1 ponto) **O Protocolo RDT 3.0.** Considere o transmissor e o receptor rdt3.0 apresentados abaixo, com as transições rotuladas de S0 a S9 na FSM do transmissor e de R0 a R3 na FSM do receptor. Complete a sequência de transições que fazem o transmissor e o receptor, em ordem global, para entregar duas mensagens do transmissor para o receptor, assumindo que o ACK da primeira mensagem transmitida se perca, assim como seja corrompido o primeiro pacote transmitido da segunda mensagem, e não ocorra mais nenhum outro erro.



Resposta: a sequência de transições, com comentários, é:

S0 – Transmissor envia 1a mensagem (pacote 0)

R0 – Receptor recebe a 1a mensagem, encaminha os dados para cima, envia ACK

S2 – Como o ACK da 1ª mensagem se perde, o temporizador estoura e a 1ª mensagem é retransmitida

R1 – Receptor identifica a mensagem como duplicata e reenvia o ACK

S3 – Desta vez o ACK é recebido corretamente, desliga o temporizador

S5 – Transmissor envia a 2ª mensagem (pacote 1)

R1 – Mensagem chega corrompida, é reenviado o ACK do pacote 0

S6 – Transmissor ignora a recepção do ACK do pacote 0

S7 – Estoura o temporizador, a 2ª mensagem é retransmitida

R2 – Receptor recebe a 2ª mensagem, encaminha os dados para cima, envia ACK

S8 – Transmissor recebe o ACK, desliga o temporizador.