



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

**Curso de Tecnologia em Sistemas de Computação**

**Disciplina: Redes de Computadores I**

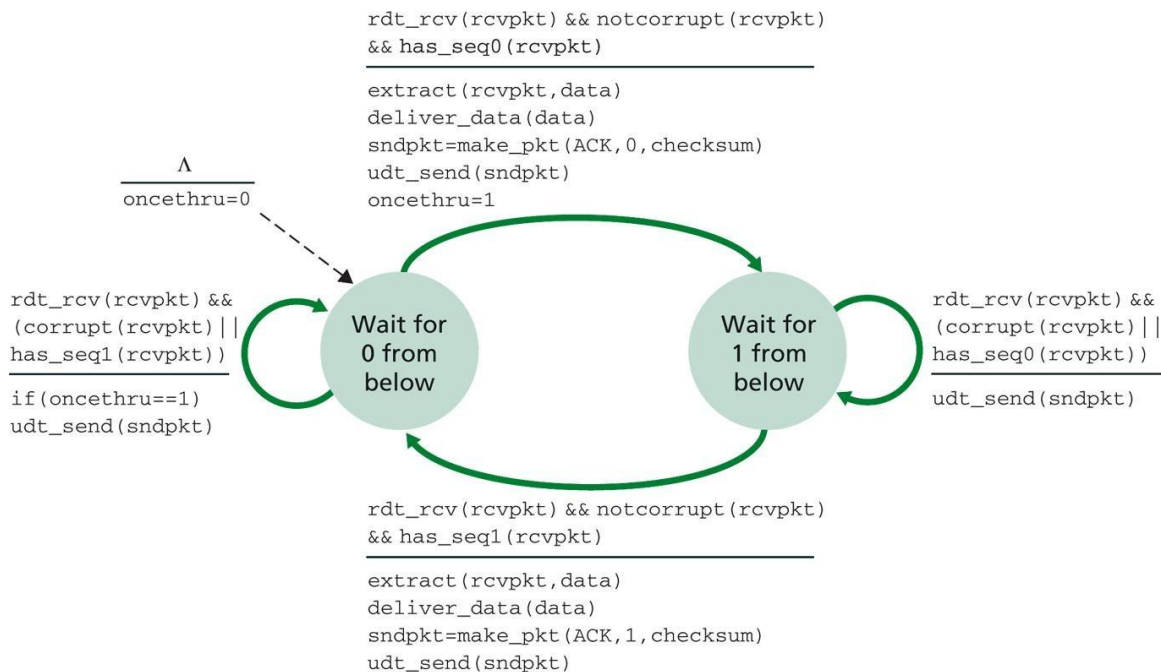
**Gabarito da AD2 - 2º semestre de 2011.**

- 
1. (1,0 ponto) No protocolo rdt3.0 do nosso livro texto, os pacotes de confirmação que fluem do destinatário para transmissor não têm números de sequência (embora tenham um campo de confirmação isto é para o ACK, que contém o número de sequência do pacote que estão confirmando). Por que esses pacotes de confirmação não requerem números de sequência?

**Resposta:** Números de sequência são necessários apenas para que o remetente possa ser informado pelo receptor que houve duplicidade no envio de um pacote. No caso de ACKs, o remetente não precisa desta informação, uma vez que no rdt3.0, o remetente, ao receber um ACK original, realiza sua transição para o próximo estado, ignorando assim ACKs que por ventura cheguem de forma duplicada.

2. (1,5 pontos) Elabore a máquina de estado finito (*finite-state machine* – FSM) para o lado destinatário do rdt3.0.

**Resposta:** Apenas o lado remetente do rdt3.0 difere-se em relação ao rdt2.2 pela adição de *timeouts*, o que pode ocasionar pacotes duplicados no destinatário. Porém, como o destinatário do rdt2.2 já trata o recebimento de pacotes duplicados, o destinatário do rdt2.2 é o mesmo do rdt3.0 .



3. (1,5 pontos) Considere o protocolo rdt3.0 seja usado na comunicação entre dois sistemas finais A e B. Considere ainda que o tempo de ida e volta (round trip time - RTT) entre A e B, é de 30 milissegundos. Suponha que A e B estejam conectados por um canal que tem taxa de transmissão igual a 1 gigabit por segundo. Considere finalmente que cada pacote enviado de A para B tem tamanho 1.500 bytes. Qual deve ser o tamanho da janela para que a utilização do canal seja maior que 95%?

**Resposta:**

**Tamanho do pacote = 1500 Bytes = 1500 \* 8 bits**

**Taxa de transmissão = 1Gbps =  $10^9$  bits/s**

**Aplicando  $L/R = 1500 \cdot 8 / 10^9 = 0,000012 = 12$  microssegundos ou 0,012 milissegundos**

**Utilização do canal  $\geq 95\%$**

**Tamanho da janela (n) = ?**

**$U = 0,95$**

**$RTT = 0,15 + 0,15 = 0,30$  milissegundos (tempo de ida e volta)**

**Aplicando  $U = (L/R) / (L/R + RTT)$**

**$0,95 = (0,012 \cdot n) / (30 + 0,012)$**

**$n = (0,95 \cdot 30,012) / 0,012$**

**$n = 2375,95$**

**Portanto n é aproximadamente 2376 pacotes. Ou seja, para essa utilização do canal, a janela é de aproximadamente 2376 pacotes.**

4. (1,0 ponto cada item) Uma aplicação pode escolher o UDP como protocolo de transporte pois ele oferece um controle por parte da aplicação melhor (do que usando o TCP) de quais dados são enviados em um segmento e quando isso ocorre. Responda:

1. Por que uma aplicação possui mais controle de quais dados são enviados em um segmento?

**Resposta:** para o envio utilizando TCP a aplicação escreve seus dados em um *buffer* gerenciado por este protocolo. O TCP apanha bytes desse *buffer* sem a necessidade de colocar uma única mensagem da aplicação em um segmento (o TCP pode colocar em um segmento, mais ou menos de uma mensagem da aplicação). Diferentemente, o UDP encapsula e envia em um segmento assim que a aplicação lhe fornece dados, isto é, uma mensagem. Desta forma, fica claro que o UDP permite maior controle por parte da aplicação do que é enviado em um segmento.

2. Por que uma aplicação possui mais controle de quando o segmento é enviado?

**Resposta:** Devido aos controles de congestionamento e de fluxo realizado pelo TCP, podem haver atrasos significativos desde o momento em que a aplicação escreve seus dados no seu *buffer* de envio até o TCP entregá-los para a camada de rede. Já no UDP não há controle de congestionamento nem de fluxo, o que permite a aplicação ter o controle de quando o segmento é enviado.

5. (2,0 pontos cada item) No protocolo para transferência confiável de dados Repetição Seletiva, chamamos de  $MaxSeq = 2^n - 1$  (sendo n é o número de bits do campo que transporta o número de sequência). Apesar dessa condição ser desejável para tornar a utilização dos bits do *header* mais eficiente, não provamos que ela é necessária.

1. Por exemplo, se  $n = 3$ , e  $MaxSeq = 4$  (isto é, o protocolo utiliza os números de sequência: 0, 1, 2, 3, 4, 0, 1, 2... ) o protocolo funciona corretamente? Explique sua resposta.

**Resposta:**

**Considerando os números de sequência (0, 1, 2, 3, 0, 1, 2, 3, 0, 1, 2, 3, ...) e com  $MaxSeq = 4$**

**Vamos inicialmente considerar o tamanho de janela igual a 3. Nesse caso, vamos supor que os pacotes de 0 a 2 são transmitidos, recebidos e reconhecidos pelo destinatário. A janela do receptor, passa então a aguardar os pacotes com os números de sequência (3, 0, 1). Caso, as**

confirmações sejam perdidas, o remetente irá temporizar e retransmitir o pacote com número de sequência 0, e este será aceito pelo receptor como um pacote contendo novos dados (e não uma duplicata). O destinatário não tem como saber se ocorreu uma retransmissão ou se trata-se da recepção de um pacote contendo novos dados. Portanto, o protocolo não funciona.

Vamos agora considerar o tamanho de janela igual a 2. Nesse caso, vamos supor que os pacotes de 0 a 1 são transmitidos, recebidos e reconhecidos pelo destinatário. A janela do receptor, passa então a aguardar os pacotes com os números de sequência (2, 3). Caso, as confirmações sejam perdidas, o remetente irá temporizar e retransmitir o pacote com número de sequência 0, e este não será aceito pelo receptor como um pacote contendo novos dados, já que na sua janela não está sendo esperado pacote com número de sequência 0. Portanto, o protocolo funciona, se as janelas do transmissor e do receptor forem dimensionadas corretamente.

2. Considerando os valores do item 1., qual deve ser o tamanho máximo das janelas do transmissor e do receptor? Explique sua resposta.

**Resposta:** É necessário garantir que números de sequência contidos em uma janela e na janela subsequente não tenham intercessão, para garantir que o problema relatado inicialmente no item 1, não ocorra. Ou seja, o tamanho máximo da janela, para os números de sequência (0, 1, 2, 3, 0, 1, 2, 3, 0, 1, 2, 3, ...) deve ser:.

$$(\text{MaxSeq} + 1)/2 = \lfloor (4+1)/2 \rfloor = 2 \text{ (isto é, piso de } (4+1)/2)$$