

1. **(0,5 pontos cada item)** Dois computadores, A e B, se comunicam a uma distância de 800Km através de um enlace de 10 Mbits/seg. A propagação do sinal no enlace é de 200.000 Km/seg.

a) Qual o tempo de transmissão de um pacote de 500 *bytes* pelo computador A?

**Resposta:**  $D_{trans} = L / R = 500 * 8 \text{ bits} / 10 \text{ M bits} / \text{seg} = 400 \mu\text{seg}$

b) Qual o tempo de transmissão de um pacote de 50 *bytes* pelo computador B?

**Resposta:**  $D_{trans} = L / R = 50 * 8 \text{ bits} / 10 \text{ M bits} / \text{seg} = 40 \mu\text{seg}$

c) Quanto tempo leva para o primeiro *bit* de um pacote ser propagado do computador A para o computador B?

**Resposta:**  $D_{prop} = D / S = 800\text{Km} / 200.000 \text{ Km} / \text{seg} = 4 \text{ mseg}$  (Independente de L)

d) Qual o número máximo de pacotes de 500 *bytes* que pode estar no enlace A-B?

**Resposta:**  $N = D_{prop} / D_{trans}(\text{item-a}) = 4\text{mseg} / 400 \mu\text{seg} = 10 \text{ pacotes}$

2. **(0,5 pontos cada item)** Suponha que você tenha acabado de criar a empresa "Network UFF" e que gostaria de registrar o domínio netuff.com.br na entidade registradora TLD ".com.br". Observações:

- Seu servidor DNS possui nome dns1.netuff.com.br e IP = 212.212.212.1;
- Seu servidor SMTP possui nome smtp.netuff.com.br e IP = 212.212.212.2;
- Seu servidor WWW possui nome www.netuff.com.br e IP = 212.212.212.3;

a) Liste quais os registros RR que devem inseridos no servidor TLD

**Resposta:**

Desprezando o campo TTL (*time to live*), temos:

(netuff.com.br, dns1.netuff.com.br, NS)

(dns1.netuff.com.br, 212.212.212.1, A)

b) Liste quais os registros RR que devem inseridos no seu servidor DNS

**Resposta:**

Desprezando o campo TTL (*time to live*), temos:

(netuff.com.br, smtp.netuff.com.br, MX)

(smtp.netuff.com.br, 212.212.212.2, A)

(www.netuff.com.br, 212.212.212.3, A)

c) Como as pessoas vão obter o endereço IP do seu *website*?

**Resposta:**

**Seja a consulta <http://www.netuff.com.br/>**

O navegador extrai o nome do hospedeiro (**www.netuff.com.br**) e repassa para o cliente da aplicação DNS. O cliente DNS envia a consulta por este nome para o servidor de DNS local, que caso o tenha em seu *cache*, já devolve o registro RR (**www.netuff.com.br, 212.212.212.3, A**). Caso contrário esta consulta é repassada pelo DNS local para um servidor de DNS TLD “com.br” (que normalmente já está na cache do DNS local) . O servidor TLD realiza em sua base de dados uma busca do tipo NS para o domínio **netuff.com.br**, e encontra o registro RR (**netuff.com.br,dns1.netuff.com.br, NS**) e subsequentemente busca pelo nome **dns1.netuff.com.br** e encontra o registro RR (**dns1.netuff.com.br, 212.212.212.1, A**). A consulta então é enviada para o servidor **dns1.netuff.com.br** e nele é buscado um registro do tipo “A”, que retorna o registro solicitado (**www.netuff.com.br, 212.212.212.3,A**). O DNS local recebe o endereço e, finalmente, este endereço é retornado para o cliente DNS que o repassa ao navegador (**212.212.212.3**).

d) Como um servidor de correio descobre o endereço IP do seu servidor de correio?

**Resposta:**

**Digamos, por exemplo, que o cliente de correio esteja enviando uma mensagem para bob@netuff.com.br:**

O servidor de correio do remetente extrai do endereço de email do destinatário **bob@netuff.com.br** o nome do domínio (**netuff.com.br**) e repassa para o lado cliente da aplicação DNS. O cliente DNS envia uma consulta do tipo “MX” do domínio (**netuff.com.br**) para servidor de DNS local e, que caso tenha em seu *cache* já retorna o registro do servidor SMTP. Caso contrário esta consulta é repassada pelo DNS local para um servidor de DNS TLD “com.br” (que normalmente já está na cache do DNS local) . O servidor TLD realiza em sua base de dados uma busca do tipo NS para o domínio **netuff.com.br**, e encontra o registro RR (**netuff.com.br, dns1.netuff.com.br, NS**) e subsequentemente busca pelo nome **dns1.netuff.com.br** e encontra o registro RR (**dns1.netuff.com.br, 212.212.212.1, A**). A consulta então é enviada para o servidor **dns1.netuff.com.br** e nele é buscado um registro do tipo “MX” do seu domínio (**netuff.com.br**) e encontra-se o registro (**netuff.com.br, smtp.netuff.com.br, MX**) e em seguida busca-se pelo nome **smtp.netuff.com.br** e encontra-se o registro RR (**smtp.netuff.com.br, 212.212.212.2, A**). O DNS local recebe estes registros e, finalmente, o endereço é retornado para o cliente DNS que o repassa ao servidor de correio (**212.212.212.2**).

3. (1,0 ponto) Comente sobre as vantagens de se utilizar a técnica de conexão de controle “fora da banda” no protocolo FTP.

**Resposta:**

No FTP, os comandos FTP são enviados e recebidos através de pacotes de controle por uma só conexão persistente na porta 21 que é responsável pela transmissão e recepção dos comandos, bem como pela manutenção dos estados em cada um dos lados da conexão (listas de arquivos, diretórios, diretório corrente, etc.). Para a transmissão de arquivos, uma conexão específica na porta 20 é criada para este fim

e encerrada ao seu término. Durante a transmissão do arquivo, a interatividade com o sistema de arquivo remoto pode continuar, sem interferência na tarefa de transferência de arquivos. Além de melhor interatividade, a tarefa de processamento de recepção dos pacotes de dados dos arquivos pode ser otimizada para este fim (lê da rede e grava em disco), enquanto a tarefa de recepção de pacotes de controle (comandos FTP) pode ser otimizada para este fim. Como pacotes de dados e pacotes de controle chegam em portas (conexões) diferentes, não há necessidade de um campo identificador do tipo de pacote (controle ou dado) no cabeçalho dos pacotes.

4. **(1,0 ponto cada item)** Explique para que servem os mecanismos relacionados abaixo nos protocolos que fazem transferência confiável de dados.

a. Números de sequência.

**Resposta:**

Os “números de sequência” são utilizados para que o receptor, nos protocolos que fazem a transferência confiável de dados, possa detectar se um pacote que acaba de ser entregue contém dados novos ou se é um pacote retransmitido. Se o protocolo faz transferência confiável de dados com paralelismo (*pipelined*), como é caso do protocolo Repetição Seletiva, os números de sequência servem também para que o receptor possa detectar pacotes perdidos ou recebidos fora de ordem. Nesse caso, o receptor deve preencher as lacunas no fluxo de *bytes* recebidos, antes da entrega dos dados para a aplicação.

b. Temporização.

**Resposta:**

Os temporizadores servem para tratar as perdas de pacotes pelo canal de transmissão. Se uma confirmação para um pacote enviado não é recebida, dentro do intervalo usado na temporização, então o pacote (ou seu ACK ou NAK) é considerado perdido e o pacote é retransmitido. Obviamente esta estratégia cria a possibilidade de pacotes em duplicata no receptor, mas este problema é resolvido através dos números de sequência.

c. *Checksum*

**Resposta:**

O *checksum* serve para verificar se durante a transmissão da origem ao destino *bits* do pacote não foram alterados, ou seja, faz uma verificação da integridade do pacote. Caso o pacote não passe pela verificação do *checksum* este não pode ser aceito no lado receptor da transmissão.

5. **(0,5 ponto cada item)** Os objetivos do controle de fluxo e do controle de congestionamento realizados pelo TCP, não são os mesmos.

a. Qual o objetivo do controle de fluxo?

**Resposta:**

O controle de fluxo visa impedir que um receptor fique sobrecarregado por estar recebendo pacotes a uma taxa superior à que este possa consumi-los.

b. De que forma é realizado o controle de fluxo?

**Resposta:**

Para implementar o controle de fluxo, o transmissor precisa conhecer o espaço disponível no *buffer* de recepção do lado receptor da comunicação. Este dado é fornecido através do campo “*Receive Window*”, que está presente no cabeçalho do TCP e que indica o espaço, em *bytes*, disponível no *buffer* de recepção. Como o TCP é *full duplex*, o campo “*Receive Window*” é preenchido, no lado receptor da conexão, todas as vezes que um segmento é enviado para o lado transmissor. Com o mecanismo de controle de fluxo, um problema poderia ser ocasionado quando a

janela de recepção disponível chegasse a zero. Nessa situação, o transmissor não enviaria dados para o receptor e caso o receptor não tivesse dados para enviar na conexão, o transmissor não teria como saber que o espaço na janela de recepção foi liberado. O TCP resolve este problema forçando o envio periódico de pacotes, por parte do transmissor, com apenas um *byte* de dados, quando a capacidade da janela de recepção chega a zero.

c. Qual o objetivo do controle de congestionamento?

**Resposta:**

O controle de congestionamento tem como alvo a infraestrutura de comunicação da Internet, que interliga os hospedeiros. Tem como objetivo evitar um colapso de comunicação no interior da rede IP evitando que os fluxos TCP enviem segmentos a uma taxa superior que a capacidade que a infraestrutura de comunicação tem, para transportar informação.

d. De que forma é realizado o controle de congestionamento?

**Resposta:**

O mecanismo de controle de congestionamento no TCP, que é baseado no “Aumento Aditivo, Diminuição Multiplicativa” (AIMD), mantém a conexão TCP em dois estados:

- **Início lento (*slow start* – SS):** A conexão TCP está neste estado quando a janela de congestionamento for inferior ao limiar (*threshold*). A cada confirmação (ACK) recebida em sequência o tamanho da janela de congestionamento é acrescido do tamanho de um MSS (*Maximum Segment Size*), o que resulta na duplicação da janela de congestionamento a cada RTT (*Round Trip Time*).
- **Prevenção de congestionamento (*congestion avoidance* - CA):** A conexão TCP está neste estado quando a janela de congestionamento (CongWin) for igual ou superior ao limiar (*threshold*). A cada confirmação (ACK) recebida em sequência o tamanho da janela de congestionamento é acrescido através da fórmula:

$$\text{CongWin} = \text{CongWin} + \text{MSS} \times \text{MSS}/\text{CongWin}$$

A aplicação desta fórmula resulta no acréscimo do tamanho de um MSS ao tamanho da janela de congestionamento a cada RTT (*Round Trip Time*). Obviamente, este crescimento na janela de congestionamento é efetuado de forma que não infrinja controle de fluxo.

Outros eventos, além da recepção de um ACK esperado e em sequência, são utilizados neste mecanismo:

- Quando um ACK é recebido em duplicata é incrementado o contador de ACKs em duplicata e quando o terceiro ACK em duplicata é recebido a janela de congestionamento é reduzida à metade e o limiar (*threshold*) também recebe este mesmo valor. Portanto, a conexão entra no estado de prevenção do congestionamento.
- Quando o temporizador expira (*timeout*) o limiar recebe o valor da metade do tamanho da janela de congestionamento e a janela de congestionamento é reduzida para seu tamanho mínimo, ou seja, de um MSS. Portanto, a conexão TCP entra na fase de início lento (*Slow Start*).