Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação
Disciplina: Redes de Computadores I
Gabarito da AP3 - 2° semestre de 2009.

1 **(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.

1.1 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)

1.2 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)

1.3 Como navegadores irão 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 subseqüentemente 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**).

1.4 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 subseqüentemente 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**).

2 **(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.

3 **(1,0 ponto)** Comente sobre o desempenho do protocolo HTTP em modo não persistente, persistente sem *pipelining* e persistente com *pipelining*.

Resposta:

O HTTP não persistente abre uma conexão TCP a cada requisição e fecha a conexão após o envio de cada resposta. O HTTP persistente mantém a conexão TCP aberta aguardando por novas requisições. Com isso, conexões TCP longas tendem a permitir uma melhor utilização da rede, através do uso de uma maior janela de congestionamento. O HTTP persistente com *pipelining* é capaz de transmitir os objetos requisitados em "paralelo". Com isso, o HTTP não persistente tende a ter o maior tempo de resposta, pois existe o custo de estabelecimento e liberação de cada conexão TCP. Em relação aos persistentes, a versão com *pipelining* tende a ser a mais eficiente já que não há o intervalo de inatividade entre as requisições dos objetos.

4 **(0,5 pontos cada item)** Em uma rede que emprega comutação por pacotes, considere o envio de pacotes de um *host* transmissor para um *host* receptor sobre uma dada rota com N roteadores.

4.1 Enumere os retardos componentes (*delays*) do retardo fim-a-fim de um pacote e explique o significado de cada retardo.

Resposta:

- **Retardo de processamento (d_{proc})**: este retardo inclui o tempo para examinar o cabeçalho do pacote e determinar para qual saída será encaminhado. Este retardo também inclui o tempo para a verificação do pacote quanto aos possíveis erros introduzidos durante a transmissão no enlace.
- **Retardo de transmissão (d_{trans})**: este retardo depende do tamanho do pacote (L) e da à taxa de transmissão (R) suportada, e seu valor é L / R.
- **Retardo de propagação (d_{prop})**: é o tempo de propagação de um bit de um extremo do enlace até o outro e depende da velocidade de propagação do sinal no meio utilizado (S) e da distância (D) e seu valor é D/S.
- **Retardo de fila (d_{fila})**: tempo que o pacote aguarda na fila de entrada para ser encaminhado à fila de saída do roteador e o tempo de espera na fila de saída até que este seja o próximo pacote a ser transmitido na interface de saída.

4.2 Qual a expressão do retardo fim-a-fim?

Resposta:

Para N roteadores atravessados da origem até destino, considerando retardos idênticos em cada um dos roteadores, a expressão seria a seguinte:

$$d_{total} = (N+1) * (d_{fila} + d_{proc} + d_{trans} + d_{prop}).$$

Já se considerarmos retardos distintos nos roteadores, a expressão seria a seguinte:

$$d_{total} = \sum_{i=1}^{N+1} \{d_{fila}(i) + d_{proc}(i) + d_{trans}(i) + d_{prop}(i)\}$$

5 **(0,5 pontos cada item)** Responda verdadeiro ou falso, explicando sua escolha:

5.1 Suponha que o hospedeiro A esteja enviando para o hospedeiro B um arquivo grande por meio de uma conexão TCP. O número de *bytes* não reconhecidos que o hospedeiro A envia não pode exceder o tamanho do *buffer* de recepção do hospedeiro B.

Resposta:
Verdadeiro. O controle de fluxo do TCP, através do campo "RcvWindow", indica para o lado transmissor a sua capacidade de recepção de dados. Portanto, o número de *bytes* não reconhecidos não excederá a capacidade de recepção.

5.2 Suponha que o hospedeiro A esteja enviando para o hospedeiro B um arquivo grande por meio de uma conexão TCP. Se o número de sequência para um segmento transmitido nessa conexão é m, então o número de sequência para o segmento subseqüente e m+1.

Resposta:
Falso. No caso do TCP, o número de sequência do segmento subsequente seria: **m + k**, onde **k** é a quantidade de *bytes* de dados contidos no segmento anterior.

6 (2,0 pontos) Imagine que no estabelecimento de uma conexão TCP, em vez do *handshake* de três vias, tenha sido empregado o *handshake* de duas vias. Em outras palavras, a terceira mensagem não é usada. É possível que ocorra algum problema transferência confiável de dados? Forneça um exemplo do problema ou mostre que não existe qualquer problema em se usar o *handshake* de duas vias.

Resposta:
Se o estabelecimento da conexão se der em duas vias é possível sim que que a transferência confiável de dados não ocorra. O ponto crucial do problema é a existência de duplicatas atrasadas na rede. Imagine a situação que cada pacote sofre temporização e é retransmitido duas ou três vezes. Alguns pacotes, tanto de controle como de dados, podem ficar retidos no núcleo da rede e emergir muito tempo depois. Lembremos que quando do estabelecimento da conexão cada lado da conexâo começa com número de sequência diferente e ambos devem concordar em relação ao número de sequência escolhido por cada lado. Agora vamos ver como o handshake de três vias resolve o problema de pacotes de controle duplicados. Imagine que emerge no hospedeiro 2 (o servidor) uma duplicata do segmento SYN de uma antiga conexão. Essa duplicata chega ao hospedeiro 2 sem o conhecimento do hospedeiro 1 (o cliente). O servidor responde ao cliente com um segmento SYNACK, para verificar se o cliente deseja realmente estabelecer uma nova conexão. O cliente percebe que o número de sequência que está sendo confirmado corresponde a um segmento SYN antigo que ficou retido na rede e rejeita o estabelecimento da conexão. Sem a terceira via do estabelecimento da conexão, isto é o ACK do SYNACK, o servidor poderia receber segmentos de dados duplicados que ficaram retidos na rede, e emergiram após a duplicata do segmento de controle SYN recebida no hospediro 2, como mencionado anteriormente.

7 (0,5 pontos cada item) Os objetivos do controle de fluxo e do controle de congestionamento realizados pelo TCP, não são os mesmos. Responda:

7.1 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.

7.2 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.

7.3 Qual o objetivo do controle de congestionamento?

Resposta:

O controle de congestionamento tem como alvo a infra-estrutura 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 infra-estrutura de comunicação tem, para transportar informação.

7.4 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:

- **Inicio 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:

  CongWin = CongWin + MSS x MSS/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 inicio lento (*Slow Start*).