



Departamento de Informática
Faculdade de Ciências e Tecnologia
UNIVERSIDADE NOVA DE LISBOA

Licenciatura em Engenharia Informática — Exame de Redes de Computadores
Ano lectivo: 2008-2009 — Chamada de Época Normal — (29 de Junho de 2009)
Exame sem consulta, com 10 questões e com a duração de 2 horas e 30 minutos

Pode-se responder a lápis mas não se podem separar as folhas umas das outras
A interpretação do enunciado faz parte da avaliação
Não se podem usar calculadoras nem telemóveis durante a resolução — só papel de rascunho

Aluno nº _____ Nome: _____

1) Um pacote de dados chega a um nó de comutação de pacotes R (um router por exemplo) é processado e transmitido por um canal C que o leva ao nó N. Indique que aspectos influenciam o tempo que passa desde que o pacote chegou a R até entregue ao nó N via o canal C. Indique a influência que cada aspecto que indicou tem.

Aspecto 1: tempo de processamento necessário para examinar a cabecsa do pacote, determinar para onde dirigir o pacote. e ainda corrigir possíveis erros.

Aspecto 2: tempo de fila, sendo que este depende da ordem de chegada do pacote ao router antes do pacote em questão. Este atraso deve-se ao facto do router funcionar no modo FIFO.

Aspecto 3: tempo de transmissão que é o tempo que o router leva a passar os bits para o canal

Aspecto 4: tempo de propagação que é o tempo que o pacote leva a atravessar os canais. tal tempo depende da dimensão do canal.

2) Considere um canal físico com 10.000 Km de comprimento, velocidade de propagação de 200.000 Km / s e com capacidade total de 1 Mbps (velocidade de transmissão). Este canal foi subdividido em sub canais fixos através de uma técnica de "multiplexagem" temporal (TDM - Time Division Multiplexing) usando 10 "time slots" de igual dimensão. Um destes sub canais, associado a um dos "time slots", está a ser usado para fornecer um canal que interliga directamente dois computadores A e B. Qual a taxa de utilização desse sub canal que liga os computadores A e B, por um protocolo do tipo "stop & wait", que utiliza mensagens com 10.000 bits de comprimento, admitindo que não há erros, desprezando o tempo de transmissão dos ACKs e que o mecanismo dos "time slots" não introduz nenhum overhead significativo. A resposta só está certa com contas indicadas e certas.

Contas a realizar para deduzir qual a opção certa:

$$T_1 = \frac{10^4}{10^5} = 100 \text{ ms} \quad T_2 = \frac{10^4}{2 \times 10^5} = 50 \text{ ms}$$

11250/10

$$\text{Taxa Ut.} = \frac{T_1}{T_1 + T_2 + T_P} = \frac{100}{100 + 50 + 50} = 0,5 = 50\%$$

pac ACK

Taxa de utilização (%) = 1 5 10 15 17 20 22 25 30 33 45 (50) 80 100% outro valor diferente destes

3) Um cliente HTTP acede a uma página HTML num servidor. Depois de obter essa página, o cliente deduz que a mesma tem 3 imagens e que as mesmas devem ser obtidas igualmente, a partir desse mesmo servidor, para mostrar o conteúdo total ao utilizador. O tempo de trânsito ida e volta (RTT) entre o cliente e o servidor é de 100 mili segundos. O cliente não tem nenhuma conexão aberta para o servidor antes de aceder à página mas já conhece o seu endereço IP. O tempo necessário para transmitir os pacotes com os comandos, a página ou as imagens são negligenciáveis.

a) Qual o tempo necessário para obter a página e as imagens usando o protocolo HTTP 1.0? A resposta só está certa com contas indicadas e certas.

Contas necessárias:

$$T = 1 \text{ ped. pag.} + 3 \text{ ped. img.} = 4 \text{ ped. HTTP} \xrightarrow{(1.0)} \begin{array}{|c|} \hline \text{HS} \\ \hline \text{ped} \\ \hline \text{ISP} \\ \hline \end{array} \xrightarrow{2 \text{ RTT}} \begin{array}{|c|} \hline \text{B} \\ \hline \text{S} \\ \hline \end{array}$$

$$= 4 \times (2 \text{ RTT}) = 8 \times 100 = 800 \text{ ms}$$

Tempo total em ms = 100 200 300 350 400 450 500 600 700 (800) 900 950 1000 1100 1200 1300 1400 1500 1600 1700 1800 1900 2000 nenhum destes valores

b) Qual o menor tempo necessário para obter a página e as imagens usando o protocolo HTTP 1.1 sem pipelining? A resposta só está certa com contas indicadas e certas.

Contas necessárias:

$$T = 1 \text{ ped. pag.} + 3 \text{ ped. img.} = 4 \text{ ped. HTTP/1.1 ao mesmo servidor.} =$$

$$= 1 \text{ HS} + 4 \text{ ped./resp.} = 5 \times \text{RTT} = 5 \times 100 = 500 \text{ ms}$$

Tempo total em ms = 100 200 300 350 400 450 500 600 700 800 900 950 1000 1100 1200 1300 1400 1500 1600 1700 1800 1900 2000 nenhum destes valores

4) Porque é que se diz que no protocolo FTP a informação de controlo é enviada out-of-band?

O protocolo FTP usa duas conexões TCP, em simultâneo (portas diferentes) uma denominada controlo e outra data. A conexão data funciona para o envio normal de dados e a conexão de controlo serve para envio de informação de autenticação e comandos de controlo.

out-of-band $\Rightarrow$ info. de controlo circula fora da banda de dados.

5) Um computador A tem uma ligação TCP para um computador B tal que o canal *bottleneck* atravessado pelos pacotes tem a velocidade de transmissão de 100 Mbps. O RTT médio entre A e B é de 100 ms (mili segundos). O protocolo TCP acabou de transitar da fase *slow start* para a fase *congestion avoidance*. O valor da janela do emissor é tal que o mesmo só usa 50% da capacidade disponível entre A e B.

a) Qual o valor aproximado em bits da dimensão da janela do emissor no momento referido no enunciado?

$$\text{Capacidade} = \frac{\text{Wind. Size}}{\text{RTT}} \Rightarrow W = \text{cap.} \times \text{RTT}$$

$$\text{capacidade} = 50\% \text{ de } 100 \text{ Mbps} = 50 \text{ Mbps}$$

$$\text{RTT} = 100 \text{ ms}$$

$$W = 50 \times 0,1 = 5 \text{ Mb} = 5 \times 10^6$$

- b) Quanto tempo leva a janela do emissor a aumentar na fase de *congestion avoidance* até ao valor que permitiria utilizar toda a capacidade de transmissão disponível entre A e B admitindo que não se perdem pacotes nem se inverte a ordem dos mesmos e que o MSS (*Maximum Segment Size*) é de 10.000 bits ?

Na fase *cong. avoidance*, por cada RTT a janela aumenta 1 MSS
 Cap. max janela = $100 \text{ Mbps} \times 0,15 = 10 \text{ Mbps}$ Aumenta janela em 5 Mbps
 $1 \text{ RTT} \rightarrow 10.000 \text{ bits}$
 $2 \rightarrow 5.000.000 \text{ bits}$ $x = 500 \text{ RTT} = 50 \text{ s}$

- c) Que conclusões tira destes resultados ?

- 6) O desenho apresenta um diagrama de um pacote IPv4. Assinale os campos que são modificados quando o pacote atravessa (sem fragmentação) um router. Escolha uma e uma só das opções. Uma resposta errada desconta 30% da cotação da pergunta.

0	4	8	16	19	24	31
VERS	COMP.	TIPO SERVIÇO	COMPRIMENTO TOTAL			
IDENTIFICAÇÃO			FLAGS	OFFSET		
TTL	PROTOCOLO		CHECKSUM			
Endereço IP origem						
Endereço IP destino						
OPÇÕES					PADDING	
DADOS						
.....						

- Os campos TTL, Tipo de Serviço e Opções
- Os campos TTL e OFFSET
- Os campos TTL, CHECKSUM e FLAGS
- Os campos TTL e CHECKSUM
- Os campos TTL e Endereço Origem
- Os campos TTL, CHECKSUM e Endereço Origem
- Os campos Endereço Origem e Identificação
- Os campos Identificação e Padding
- Os campos CHECKSUM e Endereço Origem

offset não muda por não há fragmentação

→ TTL muda à chegada, logo o checksum também
 ↳ Time-To-Live

7) Complete o pseudo código abaixo, necessário para o tratamento realizado por um *router* de uma das entradas (*knownNet*, *knownCost*) contida no vector de anúncios que recebeu via a interface *i*, de um vizinho à distância *cost*, sabendo que o *router* está a usar um algoritmo de encaminhamento por vector de distancias ou Bellman-Ford.

knownNet designa o prefixo de rede a que a entrada no vector de anúncios se refere. *knownCost* designa o custo para atingir esse prefixo de rede segundo o emissor do anúncio. O anúncio foi recebido pela interface *i*. *cost* designa o custo para atingir o emissor do anúncio via a interface *i*, isto é, o custo do canal que liga o *router* ao emissor do anúncio. O *router* tem uma tabela de encaminhamento *routingTab* que regista para cada prefixo de rede que este conhece o custo e a interface a usar para chegar à mesma. *routingTab.putNetwork (network n, int c, interface i)* regista na tabela de encaminhamento que o prefixo de rede *n* está acessível via a interface *i* à distância *c* e esmaga qualquer informação anterior associada a *n*. *routingTab.getInterface (network n)* devolve a interface a usar para chegar à rede *n* ou UNKNOWN se a mesma não é conhecida. *routingTab.getCost (network n)* devolve o custo para chegar à rede *n* se esta é conhecida. O método *putNetwork* assegura que o valor que denota custo infinito nunca é ultrapassado.

```
void processNetworkReachabilityAnnouncement ( network knownNet, int knownCost, interface i, int cost ) {

    if (routingTab.getInterface ( knownNet ) == UNKNOWN ) {

        return; };

    if (routingTab.getInterface ( knownNet ) == i) {

        return; };

    if (routingTab.getInterface ( knownNet ) != UNKNOWN && routingTab.getCost (knownNet) > knownCost+cost ) {

        return; };

}
```

8) Um *router* tem uma interface Ethernet com o endereço IP 192.168.0.1/24 e recebe um pacote com endereço de destino 192.168.0.230. A tabela de ARP do *router* tem o seguinte conteúdo:

192.168.0.1	23:45:A0:4F:67:CD
192.168.0.4	23:45:AB:2F:67:AD
192.168.0.10	23:45:AB:2F:60:CD
192.168.0.67	23:45:CD:4A:67:2D

quais os endereços origem e destino do(s) frame(s) Ethernet que o *router* deve enviar para encaminhar o pacote até ao seu destino final.

Tipo do conteúdo do frame 1: arp who has 192.168.0.230

Endereço origem: 23:45:A0:4F:67:CD

Endereço destino: FF:FF:FF:FF:FF:FF

Tipo do conteúdo do frame 2: arp reply

Endereço origem: 23:45:AB:2F:67:AD

Endereço destino: 23:45:A0:4F:67:CD

Tipo do conteúdo do frame 3: send IP

Endereço origem: 23:45:A0:4F:67:CD

Endereço destino: arp-reply

9) Considere uma rede Ethernet comutada (Switched Ethernet) com vários *hosts* e *switches*, organizada em árvore. Como sabe o algoritmo de encaminhamento é o de inundação (*flooding*) com filtragem baseada em aprendizagem pelo caminho inverso. Responda de forma justificada às seguintes questões.

a) Nesta rede é necessário existir um mecanismo de detecção de duplicados ?

Não, uma vez que os caminhos são únicos devido à organização em árvore.

b) Admitindo que o destino de um *frame* é desconhecido de todos os *switches*, o encaminhamento é otimizado ?

Não, porque é necessário usar encaminhamento por inundação (muito pouco otimizado)

c) Admitindo que o destino de um *frame* é conhecido de todos os *switches*, o encaminhamento é realizado pelo melhor caminho ?

Sim, uma vez que, devido à organização em árvore, apenas existe um caminho para o destino.

10) O Bob pretende enviar duas mensagens confidenciais M_1 e M_2 à Alice, separadas no tempo. Para tal, o Bob conhece a chave secreta simétrica da Alice, K_A , mas não quer usar essa chave constantemente para diminuir as hipóteses de um atacante a apanhar. O Bob pretende que M_2 permaneça em segredo até que M_1 tenha sido lida pela Alice. No final, o Bob quer, ainda, ter uma prova que a Alice leu as duas mensagens. Descreva um protocolo que realize o pretendido de forma segura usando a nomenclatura introduzida nas aulas teóricas.

Bob $\xrightarrow{\{N_A, H_1, K_S\} K_A}$ Alice

Bob $\xrightarrow{\{N_B, M_2\} K_2}$ Alice

Alice $\xrightarrow{\{N_A, N_B\} K_S}$ Bob

Explicações e justificação do conteúdo das mensagens trocadas entre a Bob e a Alice:

Bob envia mensagem codificada com chave simétrica da Alice contendo número de série único que identifica a sessão, mensagem M_1 e chave proposta para a sessão.

Bob envia mensagem codificada com chave de sessão contendo novo número de série e mensagem M_2 . (se Alice recebeu a msg, então consegue abrir esta)

Alice confirma que leu mensagens, enviando mensagem codificada

com chave de sessão contendo os números série das msgs.