

## PRIMEIRO TESTE

**1) Considere um pacote de tamanho  $L$  o qual é transmitido pelo sistema final A. O pacote viaja sobre um enlace até um roteador e deste sobre um segundo enlace até alcançar o sistema final B. Sejam  $d_i$ ,  $s_i$  e  $R_i$ , o comprimento, a velocidade de propagação e a taxa de transmissão do enlace  $i$ , para  $i = 1$  e  $2$ , respectivamente. O roteador atrasa o pacote de  $d_{proc}$ . Assumindo que não existe atraso na fila do roteador, qual o atraso fim-a-fim para o pacote em termos de  $d_i$ ,  $s_i$  e  $R_i$  e  $L$ ?**

Deve-se calcular o atraso de A até o roteador, e do roteador até B. Como o  $i$  para os primeiros  $d$ ,  $s$  e  $R = 1$ , o atraso é, simplesmente,  $1 + 1 + 1 = 3$ . No segundo caso, o  $i=2$ , então é  $2+2+2 = 6$ . O atraso fim-a-fim, portanto, é  $3 + 6 = 9$ .

**2) O que é um protocolo de comunicação e quais os três elementos básicos que fazem parte da definição de um tal protocolo?**

Um protocolo define o formato e a ordem das mensagens trocadas entre duas ou mais entidades comunicantes, bem como as ações realizadas na transmissão ou no recebimento de uma mensagem ou outro evento. Os três elementos básicos são: a requisição (request), a resposta (reply), e o arquivo.

**3) Explique a diferença entre as redes de circuitos virtuais e as redes de datagrama.**

Redes de datagrama: as rotas podem mudar durante uma sessão. Além disso, é o endereço de destino que determina o próximo passo.

Redes de circuitos virtuais: cada pacote leva um número (Virtual Circuit ID), é este que determina o próximo salto. O caminho é fixo, escolhido no instante de estabelecimento da conexão.

**5) Suponha que dentro do seu browser você clica em um link para obter uma página da web. Suponha que o endereço IP para a URL associada está armazenado localmente de forma que não é necessário um acesso à tabela DNS. Denote por RTT o tempo de ida e volta entre o hospedeiro local e o servidor que contém a página Web. Assuma que a página Web consista de um arquivo HTML base e três pequenas imagens. Assuma que os tempos de transmissão para todos os objetos são desprezíveis em comparação com o RTT. Quanto tempo leva (em número de RTTs) desde quando o usuário clica no link até que o cliente receba a página Web completa em cada um dos casos abaixo? Justifique sua resposta.**

**a) HTTP não-persistente, sem paralelismo.**

Há 4 conexões, e são 2 RTT por conexão TCP, logo, leva-se  $4 \times 2 = 8$  RTTs.

**b) HTTP não-persistente, com até 5 conexões paralelas.**

Leva-se 4RTTs, um para cada arquivo solicitado.

**c) HTTP persistente com paralelismo.**

Leva-se apenas um RTT, pois não se preocupa com o RTT de retorno.

**10) Diga se é verdadeiro ou falso e justifique:**

**Suponha que um usuário requisite uma página Web que consiste de um texto e duas imagens. Para essa página, o cliente enviará uma mensagem de requisição e receberá três mensagens de resposta.**

Falso, ele receberá apenas a mensagem de resposta da página solicitada.

**Com conexões não persistentes entre o browser e o servidor de origem, é possível que um único segmento TCP transporte duas mensagens distintas de requisição HTTP.**

Falso. Nas conexões não-persistentes, cada conexão TCP é encerrada após o servidor enviar o objeto originalmente solicitado.

**O cabeçalho Date: na mensagem de resposta HTTP indica a última vez que o objeto da resposta foi modificado.**

Falso. "Date:" se refere a quando a resposta HTTP foi criada e enviada pelo servidor.

**Duas páginas Web distintas podem ser enviadas pela mesma conexão persistente.**

Verdadeiro, desde que ambas páginas pertençam ao mesmo servidor.

## SEGUNDO TESTE

**1) Qual a principal serviço provido pela camada de transporte? Como ele é denominado? Qual a função dos números de porta de origem e de destino?**

Entrega de dados processo a processo e verificação de erros. São denominados: multiplexação e demultiplexação. É por meio destes números de portas que a camada de transporte sabe a que processo corretamente entregar seus pacotes.

**2) Além das aplicações multimídias, cite três tipos de aplicação (ou protocolos) que utilizam o UDP para transporte de segmentos. Explique por que eles utilizam UDP.**

DNS: o UDP é usado pela velocidade requerida. Caso o pacote se perca e ele não ache o domínio procurado, ele simplesmente busca em outro lugar.

Atualização de tabelas de roteamento RIP: eventuais perdas de pacotes não fazem diferença, uma vez que as atualizações ocorrem periodicamente. Assim, usa-se UDP pelo ganho de velocidade.

SNMP: o UDP é usado neste caso porque este protocolo deve funcionar mesmo em uma rede sobrecarregada, o que o TCP atrapalharia devido às suas políticas.

**3) O protocolo de transferência confiável RDT 3.0 consegue tratar tanto erros de bits nos pacotes (como o RDT 2.2 fazia) quanto perda de pacotes no meio de transmissão. O que este protocolo teve que acrescentar como nova característica para ser capaz de tratar esta perda de pacotes? Ele possui bom desempenho de utilização do meio? Por quê?**

Acrescentou o temporizador de contagem regressiva. Não possui bom desempenho, porque não tem paralelismo, uma vez que é do tipo pare e espere.

**4) No protocolo GBN, por que o destinatário não precisa armazenar pacotes que chegam fora de ordem? Por que o protocolo SR tem que os armazenar?**

Porque, neste caso, o GBN descarta o pacote e reenvia um ACK para o mais recente que foi recebido na ordem correta.

No SR, pacotes recebidos fora de ordem ficam no *buffer* até que todos os faltantes tenham sido recebidos, ou seja, até que um conjunto de pacotes possa ser entregue, em ordem, à camada superior.

**5) Cite quatro das principais características do TCP. Por que um mesmo segmento TCP carrega um número de sequência e um de ACK? O que eles significam? Como são obtidos?**

Orientado a conexão; conexão ponto-a-ponto; apresentação em três vias; *full-duplex*. É por conta de ser *full-duplex* que há o número de sequência e um de ACK.

O número de sequência é o número do primeiro *byte* do segmento. O número ACK é o número de sequência do próximo *byte* pelo qual se está aguardando.

O número de segmentos depende da quantidade de *bytes* sendo enviados.

**6) No TCP, o que acontece se o ACK de um determinado segmento se perde?**

Se receber outros ACKs antes do *timeout*, nada ocorre, pois o segmento foi recebido. Se os ACKs forem recebidos depois do *timeout*, é necessário reenviar.

### **7) Como funciona o controle de fluxo no TCP?**

Ele compatibiliza a taxa com a qual o remetente está enviado junto à taxa com a qual a aplicação receptora está lendo. Para isto, o remetente mantém uma variável denominada janela de recepção, usada para se ter noção do espaço de *buffer* livre disponível no destinatário.

### **8) Por que o TCP reage de forma diferente quando acontece uma perda, dependendo da forma de detecção da perda (temporização ou 3 acks duplicados)?**

Com 3 ACKs duplicados (indicam que a rede é capaz de entregar alguns segmentos), corta-se a janela de congestionamento ao meio, e ela volta a crescer linearmente. Com o esgotamento do temporizador (mais grave), a janela de congestionamento volta a 1 MSS, cresce exponencialmente até  $janelaCongestionamento/2$ , e daí em diante cresce linearmente.

### **9) Diz-se que o TCP é um protocolo justo porque divide equitativamente a banda entre conexões, se têm o mesmo MSS e RTT. Por que isso acontece?**

Pois as vazões aproximam-se sempre à interseção do igual compartilhamento da largura de banda com a da igual utilização total da banda. Assim, divide-se as bandas das conexões igualmente, de forma que, se ocorrer perda, por exemplo, ambas caem pela metade.

### **10) Qual a limitação do TCP em redes com alta largura de banda? Por quê?**

A limitação é o tamanho médio da janela de congestionamento, que deveria ser muito grande em número de segmentos. Como o TCP de hoje, para alcançar números assim, o protocolo deve tolerar uma probabilidade de perdas de segmentos irreal: cerca de  $2 \cdot 10^{-10}$  perdas.

## **PROVA DO ANO PASSADO**

**4) Suponha que durante a fase inicial do controle de congestionamento do protocolo TCP (slow start), o tamanho da janela de congestionamento seja igual a 1MSS (tamanho máximo de segmento), que o MSS seja igual a 500 bytes e que o RTT seja igual a 200ms. Nesse contexto, qual será a taxa de transmissão inicial em kbits/s? Justifique.**

Dá 20kbits/s, pois  $500 \text{ bytes} = 500 * 8 \text{ bits} = 4000$ ;  $200\text{ms} = 0,2\text{s}$ . Assim,  $4000/0.2 = 20\text{kbits/s}$ .

**5) Quais as características ou requisitos típicos de aplicações que se comportam melhor quando construídas sobre protocolos de transporte não-confiáveis? Explique por que se comportam melhor sobre estes protocolos.**

São aplicações tolerantes à perda de pacotes, e que necessitam de agilidade no transporte. Estes protocolos costumam ser mais simples que os confiáveis: não há estabelecimento de conexão; esta mesma conexão não tem estados; o excesso de cabeçalho do pacote é menor; há um controle melhor, por parte da aplicação, sobre quais dados serão transmitidos e quando. Perde-se, claro, na confiança, mas, uma vez que as aplicações sejam tolerantes à perda, o não-confiável se justifica.

**6) O que significa dizer que o TCP trabalha com ACKs cumulativos?**

Significa dizer que o TCP reconhece como recebidos os *bytes* até o primeiro *byte* que estiver faltando na cadeia dos que foram transmitidos.

**7) Verdadeiro ou falso, justifique.**

**Suponha que o *host A* envie um segmento com número de sequência 38 e 4 bytes de dados sobre uma conexão TCP para o *host B*. Neste mesmo segmento, o número do ACK é necessariamente 42.**

Verdadeiro, pois  $38 + 4 = 42$ .

**No protocolo Selective Repeat (SR), é possível que o transmissor receba um ACK para um pacote que esteja fora da sua janela corrente.**

Verdadeiro, pode receber pacotes cujo número de sequência esteja abaixo da atual base da janela, por exemplo.

**8) Por que, na arquitetura Cliente-Servidor, o tempo de distribuição de arquivos cresce linearmente, enquanto na P2P não? Explique o comportamento do tempo em cada situação.**

O C-S é linear porque apenas o servidor realiza a distribuição para todos os clientes. Destarte, maior o número de clientes, maior o tempo gasto, proporcionalmente.

No P2P, quando um cliente baixa as partes de seu arquivo, ele começa a distribuí-las entre os outros clientes os quais, por sua vez, também distribuem entre si e assim sucessivamente. Com isto, o tempo é menor pois conforme o tempo passa, aumenta o número de fontes de distribuição disponíveis.

**9) Descreva uma maneira na qual o cache é usado no DNS. Descreva duas maneiras nas quais pode-se usar cache para acesso na Web.**

Em uma cadeia de consultas, quando um servidor DNS recebe uma resposta do tipo DNS, faz-se o cache das informações da resposta, guardando-o em sua memória local. O cache pode ser utilizado para retornar imediatamente um endereço IP, sem ter de consultar outros DNS. Também pode fazer caches de endereços IP de servidores TLD.

**10) Por que se diz que o FTP envia informações de controle fora da banda?**

Porque ele usa uma conexão de controle separada, em outra porta.

**11) Descreva a função dos três principais componentes de um correio eletrônico: o agente de usuário, o servidor de correio e o protocolo SMTP. Caso o HTTP seja utilizado, qual sua função neste contexto? No SMTP, quem é o cliente e quem é o servidor?**

~~A gente de usuário: quando nós nos vestimos de usuário.~~

Agentes de usuário: permitem ler, responder (~~e apagar o recado do orkut~~), encaminhar e compôr mensagens.

Servidores de correio: são o núcleo da infraestrutura do *e-mail*. Neles está a caixa postal. Também é responsável por autenticar os utilizadores. Cuida, igualmente, de falhas por parte de outros servidores de correio, como quando a mensagem não é recebida.

SMTP: usa o transporte TCP para transferir mensagens de servidores de *e-mail* remetentes para destinatários.

Se usar o HTTP, ele substitui o SMTP como protocolo de envio, bem como substitui outros protocolos que façam acesso (IMAP, POP3 etc.)

No SMTP, o remetente é quem ativa o TCP, funcionando como cliente. Assim, é o destinatário o servidor. Isto ocorre porque o SMTP, ao contrário do HTTP, é um *push protocol*.

## **COISAS QUE MÁRCIO DISSE SER IMPORTANTE**

### **0) A camada de aplicação**

Executam em hospedeiros no espaço de usuário. Trocam mensagens para implementar a aplicação. Um exemplo é a Web, os serviços de correios, de transferência de arquivos. Seus protocolos definem as mensagens trocadas pelas aplicações e as ações a serem tomadas. Usam serviços providos pelas camadas inferiores, de transporte.

### 1) Definição de protocolo de transporte.

Um protocolo da camada de transporte fornece comunicação lógica entre processos de aplicação que rodam em hospedeiros diferentes, ou seja, tudo se passa entre os hospedeiros como se eles estivessem conectados diretamente. No remetente, a camada de transporte converte as mensagens que recebe de um processo de aplicação remetente em pacotes. No destinatário, a camada de transporte processa o segmento recebido, disponibilizando os dados para a camada de aplicação.

### 2) Um pouco de DNS

Fornece apelidos para *hosts* com nomes complicados; apelido para serviços de *e-mail*; promove a distribuição de carga.

O DNS é distribuído para que, se um ponto falhar, não falte o serviço; também para que não haja um tráfego exacerbado em um único ponto; e, por fim, para ficar mais próximo dos *hosts*, uma tentativa de diminuir o atraso.

O DNS é hierárquico; ele busca sempre num servidor de nível mais baixo, subindo na hierarquia até que encontre o domínio desejado. Quanto mais alto, maior o tempo gasto no alcance.

Quando um servidor aprende um mapeamento, ele o coloca no cache local. Futuras consultas costumam, então, ser feitas a partir deste cache.

### 3) Arquitetura Cliente-Servidor e P2P, características e diferenças.

P2P: Servidores não estão sempre *on*. Os sistemas finais se comunicam diretamente. Os pares estão conectados de forma intermitente e mudam seus IPs. Neste caso, o servidor faz upload de, pelo menos, uma cópia.

CS: Server sempre *on*, IP fixo. Para  $N$  clientes querendo um arquivo, deve enviar  $N$  cópias dele.

Em ambos os casos, para enviar uma só cópia, a fórmula do tempo gasto é  $F/u_s$ , sendo  $u$  a taxa de upload do servidor;  $F$  é o tamanho do arquivo, em *bits*. No CS, como são  $N$  cópias, o resultado deve ser multiplicado por  $N$ . O tempo mínimo necessário para que todos os clientes recebam é  $F/d_{\min}$ , sendo esta variável  $d$  o valor da menor taxa de *download* disponível entre os clientes.

Assim, no cliente-servidor, o tempo de distribuição é o valor máximo entre  $F/u_s$  e  $F/d_{\min}$ .

No P2P, é o máximo entre os mesmos valores de antes, e também o valor  $NF/(u_s + \text{Somatório}(u_i))$ .  $U_i$  é a taxa de upload de cada cliente.

#### 4) O BitTorrent (~~ha!, toma essa, uTorrent~~)

O arquivo é dividido em chunks de 256kb. Os pares recebem e enviam esses chunks. Um tracker mantém o registro dos pares participando do torrent, ou seja, da troca de chunks. Os pares fazem download e upload de chunks entre si. Um par pode sempre mudar de pares, e pode entrar e sair a qualquer momento. Geralmente, há um subconjunto de até 50 pares para realização do torrent.

Os chunks mais raros costumam ser pegos e transferidos primeiro. Os chunks costumam ser enviados ao chamado top 4. O top 4 é composto pelos 4 pares que, antes, tenham enviado chunks mais rapidamente para o enviador de chunks atual. Este top 4 é reavaliado a cada 10s. A cada 30s, um novo par é aleatoriamente escolhido e começa-se a lhe enviar chunks.

#### 5) Diferença entre Go-Back-N (GBN) e SR (Selective Repeat):

GBN: o receptor não envia ACK se existir um *gap*. O transmissor tem *timer* para o pacote mais antigo não reconhecido. Quando o timer expira, retransmite todos os pacotes a partir deste.

SR: o receptor envia ACK individual para cada pacote. O transmissor mantém um *timer* para cada pacote não reconhecido. Quando o timer expira, transmite somente o pacote não reconhecido.

#### 6) Controle de fluxo

Os hospedeiros de cada lado da conexão TCP reservam um buffer de recepção para a conexão. A aplicação não necessariamente lê estes dados no momento em que são recebidos. Assim, se a conexão continuar enviando muitos dados e a aplicação na outra ponta não os ler a tempo, o buffer pode estourar.

Assim, o TCP compatibiliza a taxa à qual o emissor está enviando com a taxa a qual o receptor está lendo.

O emissor tem uma janela de recepção, usada para ter noção do espaço livre em buffer do receptor. Há uma variável `ultimoByteEnviado`, e outra `ultimoByteReconhecido`. O receptor deve sempre garantir que `ultimoByteEnviado - ultimoByteReconhecido` seja  $\leq$  `janelaRecepcao`.



## 7) Controle de congestionamento

Muitas fontes enviando dados a uma taxa alta, acima da capacidade de rede de tratá-los. Sintomas envolvem perda de pacotes e grandes atrasos. O controle desse congestionamento pode ser por AIMD ou por slow start.

Quando a janelaCongestionamento está abaixo do limite (threshold), o transmissor entra em fase de *slow-start*, e faz a janela crescer exponencialmente.

Quando a janelaCongestionamento está acima do limite (threshold), o transmissor entra em fase de evitar-congestionamento, e faz a janela crescer linearmente.

Se houver 3 ACKs duplicados, o limite (threshold) é ajustado para janelaCongestionamento/2 e ela é ajustada para threshold.

Quando esgota a temporização, o threshold é ajustado para janelaCongestionamento/2 e ela é ajustada para 1MSS.

Para calcular a vazão de uma conexão =  $w * mss / rtt$ , sendo  $w$  = tamanho em *bytes* da janela.

Vazão em uma conexão de alta largura de banda =  $(1.22 * mss) / (rtt * \text{raiz}(L))$ , sendo  $L$  a taxa de perda.

**THX CAMILA, RENAN E ENDREW, ESTE RESUMO TAMBÉM É DE VOCÊS**