MAC 448 – PRC
BCC – 1º Semestre de 2009
Exercícios – Camada de aplicação - HTTP e FTP

1. Qual a diferença entre Aplicação e protocolo de Aplicação?

Aplicação é o programa que atende ao usuário. No caso da WEB a aplicação é o browser (lado cliente) e o servidor de páginas (lado servidor). No caso do email a aplicação é o email reader (lado cliente) e o email server (lado servidor).
O protocolo de aplicação é um conjunto de regras e formatos de mensagens usadas para se implementar a Aplicação. O protocolo HTTP é usado para que o browser "fale" com o servidor de páginas. O protocolo SMTP é usado para o programa do usuário fale com servidor de email

2. Quais as principais aplicações da Internet e os seus respectivos protocolos de aplicação?

WWW – protocolo HTTP
email – protocolos SMTP, POP3, IMAP
transferência de arquivos – FTP
servidor de nomes - DNS

3. Dê o nome de alguns agente-usuário (lado cliente) de mercado para as seguintes aplicações:

Web
Mozilla Firefox, Internet Explorer, Netscape Navigator, ...
Email
Outlook, Eudora, ...
Áudio e vídeo armazenado
Real Player, windows media player, ...

4. Como na internet um processo em uma máquina identifica um outro processo de outra máquina com o qual deseja se comunicar?

Através de 2 elementos: o endereço IP que identifica a máquina e o número de porta que identifica o particular processo dentro desta máquina.

5. O que é arquitetura cliente-servidor?

São os 2 lados de uma aplicação. Em geral o cliente é o lado do usuário e é quem inicia a operação. O servidor atende o usuário, fornecendo as respostas solicitadas.
É a arquitetura mais utilizada nas aplicações da internet.

6. Para as aplicações abaixo, diga se usam o protocolo de transporte TCP ou UDP e por quê?

email – TCP – não tolera perda de dados mas tolera atrasos
web – TCP - não tolera perda de dados (na verdade tolera pouco) mas tolera atrasos
transferência de arquivos – TCP - não tolera perda de dados mas tolera atrasos
audio/video armazenado - UDP - tolera perda de dados mas não tolera atrasos

7. Porque o HTTP é dito ser um protocolo "sem estado", ou "sem contexto"?

A cada requisição de página, a conexão é estabelecida, a página enviada e a conexão fechada. O protocolo não prevê que sejam guardadas informações durante toda a interação entre o browser e o servidor.

8. Qual a diferença entre HTTP persistente com paralelismo (pipelining) e HTTP persistente sem paralelismo? Qual o padrão na versão HTTP/1.1?

persistente com paralelismo: o browser estabelece conexão, recebe a página inicial (HTML) e solicita todos os elementos que constituem esta página em paralelo, isto é, não espera receber um para solicitar o próximo. Ao final fecha a conexão.
persistente sem paralelismo – idem, só que os elementos que constituem a página são solicitados em sequência, isto é, recebe um e só depois disso solicita o próximo.

HTTP/1.1 – persistente com paralelismo.

9. Como o browser solicita ao servidor para não usar paralelismo?

Através do parâmetro: Connection:close.

10. Na mensagem de request do HTTP, qual o significado dos 3 comandos existentes:

GET – solicita uma página
POST – solicita uma página enviando dados digitados pelo usuário
HEAD – na resposta, o conteúdo da página solicitada não é enviado (usado para depurar sistemas)

11. Na mensagem de request do HTTP, como o browser informa o servidor qual o seu tipo e qual a versão do HTTP que ele suporta?

O tipo é através ao parâmetro User-agent: <tipo do browser>.
A versão é na própria linha de mensagem do GET (ou POST ou HEAD: GET página HTTP/1.x

12. Como é feito o GET condicional para que seja usado o cashing do próprio browser?

O browser tem em seu cash a página recebida em uma determinada `<data><hora>`
O GET é enviado com o parâmetro `if-modified-since <data><hora>.`
Se a página no servidor foi modificada após a data especificada, a mesma é enviada, senão é enviado o status `not-modified` e faz com que o browser mostre a página que está em seu cash.

13. Como é feita a identificação do usuário no protocolo HTTP?

Cookies
O servidor responde com parâmetro `Set-cookie: nnnnnnn`, onde `nnnnnnn` é a identificação atribuida a este usuário. O cliente armazena esse número.
Em todas as requisições seguintes a este site (inclusive nos dias subsequentes) o cliente inclui o parâmetro `cookie:nnnnnn` em cada página solicitada.
Assim o servidor fica sabendo quem está consultando

É uma identificação fraca.

14. Como um servidor Cache numa rede local agiliza o tempo de resposta no acesso a páginas dos clientes desta rede local?

Toda requisição de página é feita ao servidor cache. O servidor de cache solicita a página ao servidor original com o parâmetro If-modified-since:<data>. O servidor original devolve a nova página se houve modificação ou o status Not-modified senão houve atualização desde aquela <data><hora>. O servidor de cache tem todas as páginas recentemente consultadas armazenadas.

Devolve ao requisitante a página modificada ou a página armazenada.
É claro que se a página está no cash e não foi modificada o tráfego no link de internet será menor. Desta forma todos os usuários da rede local vão usufruir melhor do link de internet e portanto o tempo de resposta vai melhorar.

15. Além de reduzir o tempo de resposta quais as outras vantagens de se usar servidores cache?

A – reduz o tráfego da rede interna de/para a Internet. Se o tráfego for muito alto, a solução seria aumen tar o link com a internet, o que é muito caro.
B – provê uma estrutura de distribuição de conteúdo. Com o passar do tempo, todas as páginas mais solicitadas, ficaram perto dos usuários que as solicitam com freqüência. Isto é, haverá uma distribuição efetiva do conteúdo.

16. O que é uma rede de caching cooperativo?

Existe uma hierarquia de servidores cachê. Quando o primeiro não tem a página requisitada, solicita ao segundo. Se o segundo não tem, solicita ao terceiro e assim por diante. Até chegar ao último da hierarquia que se também não tem, solicita ao servidor original.

17. O que é uma rede CDN – Content Delivery Network?

Uma rede de caches para uma determinada aplicação. A idéia é que o cache que atende o usuário seja sempre o que estiver mais perto deste.

18. Supondo uma rede local de 25 Mbps, cujo router de saida para a internet possui uma link de 2Mbps. Supondo que na rede local haja 50 requisições à internet por segundo, que o tamanho médio da página seja de 100Kbits. Qual a intensidade do tráfego na rede local? E no link da internet?

Na rede local: 50 * 100K / 25M = 5M/25M = 0,2 (muito razoável)

No link da Internet: 50 * 100K / 2.5M = 5M / 2M = 2.5 (Não vai funcionar)

19. Supondo que nesta rede local seja colocado um servidor de cache que atende em média 60% das requisições. Qual a intensidade do tráfego na rede local? E no link da internet?

Na rede local: não muda nada

Na Internet: só 40% vai para o link da Internet: 0.4 * 50 * 100K / 2M = 1 (seria melhor < 1)

20. Porque o protocolo FTP não é considerado “stateless”

Guarda o user, diretório origem e diretório destino correntes.

21. O protocolo FTP usa 2 portas lógicas. Qual o uso de cada uma delas?

Porta 20 – para transferência de dados (arquivos) e porta 21 – para comandos de controle

22. Quais os principais comandos do FTP?

USER username – identificação do usuário
PASS password – senha do usuário
LIST – lista os arquivos do diretório remoto corrente
RETR arquivo – obtém arquivo do host remoto

STOR arquivo – envia arquivo para o host remoto

23. As conexões nas portas lógicas do FTP são ou não persistentes?

Porta 20 – uma conexão é criada para cada arquivo (não persistente).
Porta 21 – A conexão permanece até o comando quit (persistente).