Lista 1 - Camada de Aplicação

Redes de Computadores

Instituto de Ciência e Tecnologia - ICT

Universidade Federal de São Paulo - UNIFESP

2o semestre de 2021

Observações:

- Os exercícios são de fixação.
- As respostas são dissertativas.
- Não copie respostas.
- Pondere completude e objetividade nas respostas.
- A resolução da lista é individual.
- Utilize o espaço nas caixas de texto para responder.

Nome:

Thiago Henrique Leite da Silva - RA: 139920

1) Explique as diferenças entre arquiteturas cliente-servidor e P2P.

Na arquitetura cliente-servidor, há um servidor central que irá prover serviços aos clientes, um exemplo disso é quando acessamos um site no navegador web. Nela os escopos são muito bem definidos, o servidor sempre irá prover serviço, e o cliente sempre irá requisitá-lo. Em contrapartida, na arquitetura P2P (par a par) nós não teremos este servidor central em sua estrutura, cada elemento da rede pode prover ou requisitar um serviço. Sendo assim, quando o número de clientes na rede cliente-servidor aumenta, o serviço fornecido vai sofrer uma certa degradação por conta da alta demanda, o contrário do que ocorre na rede P2P, pois quanto mais elementos na rede, maior a chance de um provedor do serviço desejado.

2) Para ocorrer comunicação entre processos remotos, quais informações são importantes entre origem e destino?

Uma informação essencial para a comunicação entre origem e destino ocorrer com sucesso é a origem conhecer o endereço IP de destino, ou seja, do servidor, pois o servidor é uma entidade passiva e sem seu endereço IP não conseguimos nos comunicar. Como podemos ter vários servidores respondendo sobre o mesmo IP, será necessário também conhecer a porta de destino para diferenciar os servidores.

3) Explique o que são e para que servem os *sockets*.

O socket é um descritor especial semelhante à porta que serve para permitir a comunicação entre dois processos. Ele é uma camada de abstração nesta comunicação, uma interface que vai receber a mensagem que o processo do cliente vai escrever para o servidor, essa mensagem é temporariamente bufferizada no buffer de envio do TCP (ou UDP), o protocolo vai encapsular bytes desse segmento até formar quadros e esses quadros dão múltiplos saltos até atingir o nó de destino. No destinatário, esses quadros vão ser recebidos demultiplexados, vai atingir o buffer de recebimento que está orientado ao socket, e o payload vai ser colocado no buffer de recebimento. Quando o processo do servidor chama a primitiva para ler sobre o socket, ocorre a troca de contexto e o SO pega os bytes que estão armazenados no buffer de recebimento do TCP e coloca no buffer de recebimento da aplicação servidor.

4) Explique em quais tipos de aplicações recomenda-se transporte TCP ou UDP.

O transporte TCP é recomendado para aplicações em que não podemos ter perda de pacotes, necessitamos de confiabilidade dos dados e que sejam enviados na ordem correta ou até mesmo necessitamos que em caso de falha o pacote seja reenviado. Alguns exemplos de aplicação onde esse protocolo é recomendado são: transferência de arquivos, web, acesso remoto e e-mail.
Já o protocolo UDP, apesar de não ser confiável, é recomendado para aplicações em que a velocidade de transmissão é prioridade em relação a segurança dessas informações. Um exemplo de uso para o protocolo UDP é a telefonia da internet.

5) Explique como ocorrem as requisições e respostas HTTP.

As requisições são as mensagens que o cliente envia para iniciar uma ação por parte do servidor, ela contém três elementos principais: o método HTTP (Get, Put, Post, Delete, etc), o destino da requisição (dado normalmente pela URL) e a versão HTTP. Já as respostas contém as seguintes informações: a versão do HTTP, o código de status que indicará se a requisição teve sucesso ou esbarrou em algum erro, uma breve descrição textual do código de status obtido, o tipo de conteúdo da resposta (imagens, html, etc), e o próprio conteúdo. As respostas vão variar de acordo com o método HTTP utilizado na requisição.

6) Explique as diferenças entre conexão HTTP persistentes e não-persistentes.

Na conexão HTTP persistente, nós teremos, no máximo, um objeto enviado por uma conexão TCP. Já na conexão HTTP não persistente, nós já podemos ter múltiplos objetos enviados por uma única conexão TCP entre o cliente e o servidor. Na não persistente, há sobrecarga do SO para cada conexão TCP e os navegadores geralmente abrem múltiplas conexões TCP para buscar os

objetos referenciado de forma paralela; já na persistente o servidor deixa a conexão em aberto depois de enviar a resposta, além disso, as mensagens HTTP posteriores entre o cliente e servidor serão enviadas pela conexão aberta.

7) Explique o que são, para que servem e como funcionam os *cookies*.

Cookies são identificadores de sessão, eles identificam unicamente a sessão de um usuário. Eles possuem quatro componentes: 1) linha de cabeçalho de cookie da mensagem de resposta HTTP, 2) linha de cabeçalho de cookie na máquina de do usuário, controlado pelo navegador do usuário, 4) banco de dados de apoio do website. Eles são necessários por conta do protocolo HTTP não possuir estados, ou seja, ele não guarda os dados de requisições antigas, sendo assim, apenas com o protocolo não conseguiríamos manter nenhuma informação salva nos sites que acessamos. Como forma de resolver este impasse, foram desenvolvidos os cookies, agora portanto, quando o usuário acessa um website na internet pela primeira vez, é gerado para ele um ID exclusivo de sua sessão, um cookie, além de uma entrada no banco de dados de apoio para o ID no lado do servidor. Pelo fato de armazenar algumas informações do cliente, temos uma questão bem comentada sobre privacidade, já que eles permitem que o servidor descubra muitas informações sobre o cliente.

8) Explique como funcionam caches web.

O objetivo principal dos caches web é atender a requisição do cliente sem a necessidade de envolver o servidor de origem. Para isso, é utilizado um servidor proxy, que é um nó intermediário mais próximo do browser. Quando o usuário acessa pela primeira vez o site, este objeto ao invés de vir direto para o browser, ele passa pelo proxy, o proxy baixa esse objeto e devolve para o usuário. Caso outro usuário requisite o mesmo objeto, não é necessário um novo download, pois o objeto estará salvo localmente. Como o proxy está mais próximo do usuário, na mesma rede local, não temos taxa de gargalo, irá depender muito mais da tecnologia, com isso conseguimos melhorar a experiência do usuário pela rapidez da resposta, além de pouparmos banda do link de acesso, diminuindo a carga de trabalho de acesso. Em resumo, o cache irá atuar como cliente e servidor para reduzir o tempo de resposta para a requisição do cliente e reduzir o tráfego no enlace de acesso.

9) Explique como funciona o protocolo FTP.

A ideia do protocolo FTP é realizar atividades de transferência de arquivos. Ele transfere arquivos para um host remoto e utiliza a arquitetura cliente servidor. O cliente precisa conectar ao servidor a partir de uma porta, que por padrão é a 21; já do lado do cliente, o SO aloca uma porta temporária. Este protocolo usa o TCP na camada de transporte e possui dois canais de comunicação: uma conexão fora da banda e uma conexão para transmissão de dados. Ele também permite o acesso

ao diretório remoto através dos comandos enviados sobre determinada conexão de controle. É importante salientar que após a transferência de um arquivo, o servidor fecha a conexão de dados, mas mantém aberta a conexão de controle.

10) Explique o funcionamento e a interação dos protocolos envolvidos no envio e recuperação de mensagens de email.

A transmissão de e-mail entre servidores de e-mail ocorre via protocolo SMTP e a recuperação de e-mails que um usuário faz a partir de um servidor de e-mail ocorre através de dois principais protocolos, o POP3 ou IMAP.

O SMTP (Simple Mail Transfer Protocol) é executado nos servidores, ele usa TCP para transferir as mensagens com confiabilidade do cliente para o servidor. A porta padrão no SMTP é a 25 e a transferência entre as partes é direta (servidor de envio ao servidor de recepção). Na transferência, nós temos três etapas principais: o handshaking, a transferência de mensagens e o fechamento.

O POP3 possui um funcionamento mais simples, possui uma fase de autorização e uma fase de transação. O usuário não pode reler um e-mail se mudar o cliente no POP e ele também não possui estado entre as sessões.

Por fim, o IMAP mantém todas as mensagens em um local, o servidor, além de permitir que o usuário organize as mensagens em pastas. Diferentemente do POP, o IMAP mantém o estado do usuário entre as sessões, como nome das pastas e mapeamento entre ID 's de mensagem e nome de pasta.

11) Explique o que é e para que serve o protocolo DNS.

DNS é o protocolo de nomes de domínio, que é uma base de dados distribuída implementada em hierarquia de muitos servidores de nomes. Os hosts na internet e roteadores são identificados pelo endereço IP, porém os usuários não buscam um serviço geralmente pelo IP, e sim pelo “nome” do site, sendo assim, precisamos mapear esses nomes de site e IPs, a forma encontrada para se fazer isso é do DNS. Agora, hosts, roteadores e servidores DNS se comunicam para resolver nomes, ou seja, traduzir IP/nome. Os servidores DNS não são centralizados pois teríamos um único ponto de falha, o volume de tráfego seria gigantesco, o banco de dados seria centralizado e distante, além da manutenção ser outro problema, ou seja, não seria escalável.

12) Explique como é organizada a distribuição de nomes na hierarquia DNS.

A hierarquia DNS é organizada em três camadas:

DNS Root: 13 sistemas espalhados pelo mundo.

DNS TLD: Responsáveis pelos domínios de mais alto nível, como .com, .org, .edu.

DNS Authoritative: Provêem resolução de nomes dos servidores da organização: yahoo.com, amazon.com, unifesp.br, etc.

13) Explique quais são as diferenças entre consultas DNS iterativas e recursivas.

A principal diferença entres as consultas é que na consulta iterativa, o servidor de nomes não busca a resposta completa, e sim retorna uma referência a outros servidores DNS que podem ter a resposta. Já na consulta recursiva, o servidor DNS que recebeu sua consulta irá fazer a busca da resposta, e durante este processo, o servidor pode consultar outros servidores DNS da internet em seu nome, por isso a ideia da recursividade.

14) Explique como funcionam as aplicações de bittorrent, DHT e Skype.

Nas aplicações de BitTorrent, o arquivo é fragmentado em blocos de 256 Kbytes e são distribuídos na rede P2P torrent. Primeiro, o usuário baixa um arquivo .torrent na web; depois, com a aplicação o cliente abre o arquivo, se registra no tracker e baixa a lista de pares contendo o IP e porta do par; o terceiro passo é o cliente iniciar as conexões TCP simultâneas com os pares para baixar os blocos que compõem o arquivo. Ao terminar o download de um bloco, outros pares irão conectar ao cliente para baixá-lo; ao final, quando um par tem um arquivo inteiro, ele pode sair ou permanecer no torrent.

O DHT (Distributed Hash Table) é uma base de dados P2P distribuída. O banco de dados armazena tuplas do tipo (chave, valor), e os pares consultam esse banco com a chave, o banco por sua vez retorna os valores de acordo com a chave informada. Esses pares também podem inserir mais tuplas no banco de dados. Essas tabelas são úteis para localizarmos informações em uma grande coleção de dados, já que todas as chaves estão em um formato consistente e o conjunto de chaves pode ser dividido de forma a permitir uma rápida identificação onde o valor será encontrado.

O Skype já entra no modelo híbrido, onde temos um servidor de login skype, que tem algumas similaridades com o tracker. Os usuários dão o start nas aplicações skype e esse nó precisa definir o status online e passar para o servidor qual em qual porta ela está escutando. O servidor terá o registro de todos os nós na rede. A rede P2P serve para os pares de usuários se comunicarem. O protocolo próprio da camada de aplicação é deduzido por engenharia reversa.