

Lista I – Infraestrutura de comunicação 2024.2

1. O que é um protocolo? Exemplifique com protocolos das camadas de aplicação, transporte e redes.

Resposta

Um protocolo é um conjunto de regras e convenções que permitem que sistemas de computadores ou dispositivos se comuniquem entre si de forma eficiente e segura. Ele define a estrutura, formato e sequências das mensagens trocadas entre os sistemas e como essas mensagens são processadas, garantindo que a comunicação ocorra de maneira organizada e sem erros.

A **Camada de Aplicação** é a camada mais próxima do usuário e é responsável pela interação com o software do usuário, além de definir os protocolos usados para comunicação entre as aplicações. Ela fornece serviços diretamente para os programas e os usuários finais.

Exemplos de protocolos na camada de aplicação:

- **HTTP (Hypertext Transfer Protocol):** Usado para comunicação entre servidores web e navegadores
- **FTP (File Transfer Protocol):** Usado para a transferência de arquivos entre um cliente e um servidor.
- **SMTP (Simple Mail Transfer Protocol):** Usado para envio de e-mails entre servidores de correio eletrônico.
- **DNS (Domain Name System):** Usado para a resolução de nomes de domínio em endereços IP.

Camada de Transporte

A **Camada de Transporte** é responsável por garantir a entrega correta dos dados entre os sistemas. Ela controla a segmentação, o fluxo e a correção de erros, além de garantir que a comunicação seja realizada de forma confiável e eficiente.

Exemplos de protocolos na camada de transporte:

- **TCP (Transmission Control Protocol):** Protocolo confiável e orientado a conexão. Ele garante que os dados sejam entregues corretamente e na ordem certa. Se algum pacote de dados for perdido ou corrompido, o TCP solicita a retransmissão do pacote.
- **UDP (User Datagram Protocol):** Protocolo não confiável e sem conexão. Ele transmite os dados sem garantir que eles cheguem ao destino corretamente ou na ordem correta, mas é mais rápido que o TCP, sendo usado em situações onde a velocidade é mais importante que a confiabilidade, como em transmissões de vídeo em tempo real.

Camada de Rede

A **Camada de Rede** é responsável por encaminhar os pacotes de dados entre diferentes redes e garantir que eles cheguem ao destino correto. Ela lida com o endereçamento lógico e o roteamento dos pacotes.

Exemplos de protocolos na camada de rede:

- **IP (Internet Protocol):** Protocolo fundamental para a comunicação em redes baseadas na Internet. Ele é responsável por endereçar e encaminhar pacotes de dados entre dispositivos, garantindo que cheguem ao destino correto. O IP usa endereços IP (como 192.168.1.1) para identificar dispositivos na rede.
- **ICMP (Internet Control Message Protocol):** Usado para relatar erros e informações de controle na rede. O exemplo mais conhecido de ICMP é o comando "ping", que verifica a conectividade entre dois dispositivos em uma rede.

2. Como ocorre o acesso à Internet por DSL? Quais as principais desvantagens em relação ao FTTH?

Resposta:

O **DSL (Digital Subscriber Line)** é uma tecnologia que utiliza a linha telefônica de cobre tradicional para fornecer acesso à Internet de alta velocidade.

Tipos de DSL:

- **ADSL (Asymmetric DSL):** Mais comum para conexões residenciais, onde a velocidade de download (recebimento de dados) é maior que a velocidade de upload (envio de dados).
- **VDSL (Very High-Speed DSL):** Uma versão mais rápida, com maiores velocidades de upload e download, usada em áreas mais próximas ao provedor de Internet.

Principais Desvantagens do DSL em Relação ao FTTH

FTTH (Fiber to the Home) é uma tecnologia de acesso à Internet onde a fibra ótica chega diretamente até a residência ou local de trabalho do usuário, oferecendo velocidades significativamente superiores.

DLS perde em:

- Velocidade mais baixa
- Limitações de capacidade
- Estabilidade inferior
- Custo e disponibilidade

3. Apresenta os conceitos de comutação de pacotes e de circuitos. Faça uma comparação entre eles.

Comutação de Pacotes (Packet Switching):

A **comutação de pacotes** é um método de transmissão de dados onde **as informações são divididas em pequenos blocos chamados pacotes**. Cada pacote contém uma parte da mensagem original, além de informações de controle (como endereços de origem e destino) e dados necessários para a reconstrução correta da mensagem no destino.

Comutação de Circuitos (Circuit Switching):

A **comutação de circuitos** é um método em que **é estabelecida uma conexão dedicada e exclusiva entre o remetente e o receptor antes de iniciar a comunicação**. A comunicação ocorre de forma contínua e não é interrompida até que a transmissão seja concluída. A conexão permanece ativa durante toda a duração da comunicação, independentemente da quantidade de dados sendo transmitida.

5. Descreva o funcionamento do protocolo HTTP, especificando tipos de mensagens e sua respectiva camada.

O **HTTP** é um protocolo de comunicação baseado no modelo **requisição-resposta**, funcionando na **camada de aplicação**. Ele permite que um cliente (como um navegador) envie uma requisição ao servidor, que por sua vez, processa a requisição e envia uma resposta com o recurso solicitado ou informações sobre o erro. O HTTP utiliza a **camada de transporte**, geralmente com o **TCP**, para garantir a entrega confiável dos dados entre o cliente e o servidor. As mensagens HTTP incluem informações como cabeçalhos e, opcionalmente, um corpo com o conteúdo da resposta.

6. Apresente a arquitetura do correio eletrônico atual, quais os protocolos envolvidos e onde eles são utilizados.

A arquitetura do correio eletrônico atual é baseada em um modelo cliente-servidor, no qual os clientes enviam e recebem mensagens de servidores de correio eletrônico. Essa arquitetura utiliza diversos protocolos para garantir a entrega, o envio e o armazenamento de e-mails.

Resumo dos Protocolos:

- **SMTP:** Envio de e-mails (entre servidores e para o servidor de envio).
- **POP3:** Recuperação de e-mails do servidor (cliente para servidor).
- **IMAP:** Acesso e organização de e-mails no servidor (cliente para servidor).
- **MIME:** Suporte para anexos e multimídia em e-mails.

7. Diferencie uma consulta DNS iterativa de uma consulta DNS recursiva. Apresente vantagens e desvantagens.

Consulta DNS Iterativa:

- O **cliente DNS** consulta o servidor DNS, que, se não souber a resposta, encaminha para outro servidor.
- **Vantagens:** Menor carga nos servidores DNS.
- **Desvantagens:** Maior latência, o cliente precisa fazer várias consultas.

Consulta DNS Recursiva:

- O **servidor DNS** busca a resposta completa em nome do cliente, fazendo as consultas necessárias.
- **Vantagens:** Menor latência, mais simples para o cliente.
- **Desvantagens:** Maior carga no servidor DNS.

8. Considere as arquiteturas cliente-servidor e P2P. Qual delas é potencialmente mais escalável em condições iguais de infraestrutura? Justifique.

Arquitetura Cliente-Servidor:

- **Descrição:** Nesta arquitetura, os **clientes** solicitam serviços ou recursos dos **servidores**. O servidor centraliza a gestão dos dados e processamentos, enquanto os clientes interagem com ele.

Arquitetura P2P (Peer-to-Peer):

- **Descrição:** Na arquitetura P2P, todos os **nós (peers)** podem atuar tanto como **clientes quanto servidores**, compartilhando recursos diretamente entre si, sem a necessidade de um servidor central.

A **arquitetura P2P** é mais escalável que a **cliente-servidor**. Na P2P, cada nó contribui com recursos, permitindo uma **escala horizontal** eficiente, onde o aumento de nós melhora a capacidade da rede. Já na arquitetura cliente-servidor, a escalabilidade é limitada pela capacidade do servidor, o que exige mais recursos e servidores conforme o número de clientes cresce. Assim, a P2P é mais **flexível e resiliente** em termos de escalabilidade.

9. Como ocorre um ataque de DDoS através do DNS?

Um **ataque DDoS através do DNS** ocorre quando o atacante envia **consultas DNS falsificadas** (spoofing) para servidores DNS.

10. Qual o papel da camada de transporte? Considerando o protocolo UDP, que serviços são prestados por essa camada?

A **camada de transporte** garante a comunicação entre processos e oferece serviços como controle de fluxo e erros. O **UDP**, especificamente, oferece um serviço simples, sem conexão e sem garantias de entrega, adequado para aplicações que priorizam a **velocidade** sobre a **confiabilidade**.

11. Descreva os passos necessários para o estabelecimento de conexão TCP

- **Passo 1:** O cliente envia um SYN para o servidor, solicitando a conexão.
- **Passo 2:** O servidor responde com um SYN-ACK, confirmando a solicitação e estabelecendo sua parte da comunicação.
- **Passo 3:** O cliente envia um ACK final para confirmar a resposta do servidor, completando o processo de estabelecimento da conexão.

Após esses três passos, a conexão TCP está **estabelecida** e pronta para a troca de dados.

12. O RTT estimado pelo TCP é uma média móvel exponencialmente ponderada (MMEP). Apresente a fórmula para a estimativa do RTT e justifique a definição de MMEP.

O **RTT (Round-Trip Time)** é o tempo total que um pacote leva para viajar do cliente ao servidor e voltar

Fórmula para a Estimativa do RTT no TCP

A fórmula para calcular o RTT estimado (denotado como SRTT, Smoothed RTT) utilizando a MMEP é:

$$SRTT_{\text{novo}} = (1 - \alpha) \times SRTT_{\text{anterior}} + \alpha \times RTT_{\text{medido}}$$

Onde:

- **SRTT:** RTT estimado (Smoothed RTT).
- **RTT medido:** RTT medido recentemente (tempo de ida e volta para um pacote específico).
- α : Fator de suavização, normalmente configurado como 0,125 (12,5%).

Justificativa/Definição: é utilizada para estimar o RTT porque ela prioriza os valores mais recentes, permitindo que a estimativa se ajuste rapidamente a mudanças nas condições da rede

13. Apresente o conceito de controle de fluxo e de controle de congestionamento do TCP(especificando o objetivo de cada um deles).

Controle de Fluxo no TCP

O **controle de fluxo** é um mecanismo que **regula a quantidade de dados** que o remetente pode enviar antes de receber uma confirmação (ACK) do destinatário. O objetivo principal é **evitar que o receptor seja sobrecarregado** com dados mais rápidos do que ele pode processar, garantindo que o receptor tenha recursos suficientes (como buffer) para armazenar os pacotes recebidos.

Controle de Congestionamento no TCP

O **controle de congestionamento** é um mecanismo que **previne que a rede seja sobrecarregada** devido ao envio excessivo de dados, o que pode levar a perdas de pacotes e degradação da qualidade da comunicação.

Diferença entre controle de fluxo e congestionamento:

- **Controle de Fluxo** lida com a capacidade do **receptor**.
- **Controle de Congestionamento** lida com a **capacidade da rede**.

14. Como funciona a prevenção de congestionamento do controle de congestionamento do TCP? Em que situações ela é aplicada?

O controle de congestionamento no TCP busca evitar que a rede seja sobrecarregada com tráfego excessivo, o que pode causar perda de pacotes, aumento de latência e degradação da qualidade da comunicação. O TCP usa diferentes algoritmos de controle par

A prevenção de congestionamento no TCP é aplicada nas seguintes situações:

Perda de Pacotes

Aumento no Round-Trip Time (RTT)

Tráfego Excessivo na Rede

15. Diferencie as versões Tahoe e Reno do TCP.

TCP Tahoe e TCP Reno são versões do TCP que diferem principalmente na forma como lidam com a perda de pacotes:

- **TCP Tahoe:** Quando há perda de pacote, o TCP retorna ao Slow Start, reiniciando a janela de congestionamento.
- **TCP Reno:** Ao detectar perda de pacote, utiliza Fast Retransmit e Fast Recovery, permitindo que a janela de congestionamento não seja completamente reiniciada, acelerando a recuperação.

Em resumo, TCP Reno é mais eficiente que TCP Tahoe porque mantém a transmissão ativa e evita reiniciar o processo de Slow Start após a perda de pacotes.