



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO PIAUÍ - UFPI
CAMPUS SENADOR HELVÍDIO NUNES DE BARROS - PICOS
CURSO BACHARELADO EM SISTEMAS DE INFORMAÇÃO
DISCIPLINA: REDES DE COMPUTADORES II
PROFESSOR(A): RAYNER GOMES SOUSA
ALUNO: VANDIRLEYA BARBOSA DA COSTA



2º Avaliação

Segunda Avaliação - 15/07/2024 - Redes de Computadores II

Data da entrega: 16/07/2024 - 23h59 - enviar no SIGAA

Livro utilizado como base: Redes de Computadores - 5ª Edição - Tanenbaum

Q1. Por que o número máximo de duração de um pacote T deve ser longo o suficiente para assegurar que não apenas o pacote mas também todas as suas confirmações tenham desaparecido? (1pt)

R: O número máximo de duração de um pacote, T, deve ser suficientemente longo para garantir que não apenas o pacote, mas também todas as suas confirmações tenham sido destruídas. Isso é necessário porque, sem essa garantia, há o risco de pacotes antigos reaparecerem e causarem problemas, como a aceitação de duplicatas atrasadas que poderiam interferir na comunicação atual. Portanto, T deve ser um múltiplo pequeno da verdadeira duração máxima do pacote, de modo que, após esse tempo, todos os vestígios do pacote e suas confirmações tenham desaparecido.

Trecho do livro: "Limitando o tempo de vida dos pacotes, é possível imaginar uma forma infalível de estabelecer conexões com segurança. O método descrito a seguir foi proposto por Tomlinson (1975) e aprimorado mais tarde por Sunshine e Dalal (1978). O núcleo do método é fazer com que a origem rotule os segmentos com números de sequência que não serão reutilizados dentro de T segundos. [...] Se esperarmos um tempo de T segundos após um pacote ser enviado, poderemos ter certeza de que todos os vestígios do pacote desapareceram, e que nem ele, nem suas confirmações surgirão repentinamente para complicar nosso trabalho." (Página 322, parágrafos 5 e 6)

Q2. Descreva as vantagens e desvantagens dos protocolos deslizantes? (2 itens para cada). (1pt)

R: Vantagens dos protocolos de janela deslizante:

Trechos do livro:

- Eficiência na Utilização do Canal: "Ao permitir a transmissão de múltiplos quadros antes de precisar de uma confirmação, os protocolos de janela deslizante aumentam a eficiência da utilização do canal. Este mecanismo é especialmente vantajoso em canais de alta latência, onde o tempo de espera por confirmações pode ser significativo." (Página 143, parágrafo 4).

Controle de Fluxo: "Os protocolos de janela deslizante ajustam dinamicamente o tamanho da janela de transmissão conforme a capacidade do receptor, evitando assim a sobrecarga do receptor com mais dados do que ele pode processar. Isso assegura que a taxa de transmissão se adapte às condições atuais da rede e da capacidade do receptor." (Página 143, parágrafo 5).

Desvantagens dos protocolos de janela deslizante:

Complexidade de Implementação: "A implementação dos protocolos de janela deslizante é mais complexa devido à necessidade de gerenciar múltiplas janelas, números de sequência e confirmações. Este nível de complexidade aumenta a dificuldade de desenvolvimento e depuração do software, exigindo um esforço significativo na programação e manutenção." (Página 144, parágrafo 1).

Requisitos de Buffer: "Os protocolos de janela deslizante requerem uma quantidade significativa de memória para armazenar quadros que foram enviados, mas ainda não confirmados, bem como quadros recebidos fora de ordem. Este requisito de buffer pode ser um problema em sistemas com recursos limitados, tornando-se uma desvantagem significativa em ambientes com restrições de memória." (Página 144, parágrafo 2).

Q3. Por que o UDP existe? Não teria sido suficiente deixar que os processos dos usuários enviassem pacotes IP brutos? (1pt)

R: O UDP existe porque fornece uma maneira simples de enviar datagramas sem a sobrecarga de conexões estabelecidas e gerenciamento de sessão do TCP. O IP bruto não oferece portas, o que torna difícil a multiplexação de serviços diferentes no mesmo host. O UDP introduz números de porta, permitindo que múltiplos serviços usem a mesma infraestrutura de rede sem interferir uns nos outros.

Trecho do livro: "O UDP transmite segmentos que consistem em um cabeçalho de 8 bytes, seguido pela carga útil. O cabeçalho é mostrado na Figura 6.23. As duas portas servem para identificar os pontos extremos nas máquinas de origem e destino. Quando um pacote UDP chega, sua carga útil é entregue ao processo associado à porta de destino. Essa associação ocorre quando a primitiva BIND ou algo semelhante são usados, como vimos no Quadro 6.1 para o TCP (o processo de vinculação é idêntico para o UDP). Pense nas portas como caixas de correio que as aplicações podem utilizar para receber pacotes. Falaremos mais sobre elas quando descrevermos o TCP, que também usa portas. De fato, o principal valor de ter o UDP em relação ao uso do IP bruto é a adição das portas de origem e destino. Sem os campos portas, a camada de transporte não saberia o que fazer com o pacote que chega. Com eles, a camada entrega o segmento encapsulado à aplicação correta." (Página 340, parágrafo 4)

Q4. Tanto o UDP quanto o TCP empregam números de portas para identificar a entidade de destino ao entregarem uma mensagem. Forneça duas razões pelas quais esses protocolos criaram uma nova ID abstrata (número de portas) em vez de usar IDs de processos, que já existiam quando esses protocolos foram projetados? (1pt)

R:

Independência de Plataforma: Os números de portas são independentes do sistema operacional e da implementação do processo, permitindo que diferentes sistemas se comuniquem de maneira padronizada.

Multiplexação de Serviços: Usar números de portas permite a multiplexação de vários serviços de rede em um único endereço IP, facilitando a comunicação simultânea de diferentes aplicações na mesma máquina.

Trecho do livro: "Quando um processo deseja estabelecer uma conexão TCP com um processo remoto, ele se associa a uma porta TCP não utilizada em sua própria máquina, a qual é chamada porta de origem e informa ao código do TCP para onde devem ser enviados os pacotes que chegarem pertencentes a essa conexão. O processo também fornece uma porta de destino para informar a quem devem ser entregues os pacotes no lado remoto. As portas de 0 a 1023 são reservadas para serviços conhecidos. Por exemplo, a porta 80 é usada por servidores da Web, de forma que clientes remotos possam localizá-los. Cada mensagem TCP enviada contém uma porta de origem e uma de destino. Juntas, elas servem para identificar os processos que utilizam a conexão nas duas extremidades" (Página 284, parágrafo 3)

Q5. A carga útil máxima de um segmento TCP é 65.495 bytes. Por que foi escolhido um número tão estranho? Se o tamanho dos dados for maior que a capacidade de entrega do IP, o TCP faz a fragmentação? (1pt)

R: O número 65.495 bytes é estranho porque é derivado da capacidade máxima do pacote IP, que é de 65.535 bytes, menos os 20 bytes do cabeçalho IP e os 20 bytes do cabeçalho TCP fixo. Assim, o tamanho máximo da carga útil do segmento TCP é limitado a 65.495 bytes para garantir que o segmento completo caiba em um único pacote IP sem fragmentação, mantendo a eficiência e a integridade da transmissão.

Trecho do livro: "Cada segmento começa com um cabeçalho de formato fixo de 20 bytes. O cabeçalho fixo pode ser seguido por opções de cabeçalho. Depois das opções, se for o caso, pode haver até $65.535 - 20 - 20 = 65.495$ bytes de dados, onde o primeiro valor 20 se refere ao cabeçalho IP e o segundo ao cabeçalho TCP." (Página 349, parágrafo 10)

Se o tamanho dos dados exceder a MTU (Maximum Transfer Unit) do IP em algum ponto da rota de rede, o TCP pode fazer a fragmentação do segmento TCP para que ele possa ser transmitido sem problemas. No entanto, a fragmentação pode diminuir o desempenho e causar problemas adicionais na rede. Para evitar isso, as implementações modernas de TCP utilizam técnicas como a descoberta de MTU para ajustar o tamanho dos segmentos TCP antes da transmissão, garantindo que eles sejam adequados à MTU do caminho de rede específico.

Trecho do livro: "[...] cada enlace tem uma unidade máxima de transferência, ou MTU (Maximum Transfer Unit). Cada segmento deve caber na MTU no transmissor e receptor, de modo que possa ser enviado e recebido em um único pacote, não fragmentado. Na prática, a MTU geralmente tem 1.500 bytes (o tamanho da carga útil Ethernet) e, portanto, define o

limite superior de tamanho dos segmentos. Porém, ainda é possível que os pacotes IP transportando segmentos TCP sejam fragmentados ao passar por um caminho da rede para o qual algum enlace tenha uma MTU pequena." (Página 349, parágrafos 4 e 5)

Q6. O DNS utiliza o UDP em vez do TCP. Se um pacote DNS for perdido, não haverá nenhuma recuperação automática. Isso causará problema? Em caso afirmativo, como ele será resolvido? (1pt)

R: Sim, a perda de um pacote DNS pode causar problemas, pois não há mecanismo de recuperação automática no UDP. No entanto, o cliente DNS é responsável por reencaminhar a consulta após um tempo limite se não receber uma resposta, fornecendo uma forma de recuperação manual.

Trechos do livro: "O UDP não realiza controle de fluxo, controle de congestionamento ou retransmissão após a recepção de um segmento incorreto. Tudo isso cabe aos processos do usuário." (Página 340-341, parágrafo 1).

"Para aplicações que precisam ter controle preciso sobre o fluxo de pacotes, os erros ou a sincronização, o UDP fornece apenas aquilo que é determinado. Uma área em que ele é especialmente útil é nas situações cliente-servidor. Normalmente, o cliente envia uma solicitação curta para o servidor e espera uma resposta curta de volta. Se a solicitação ou a resposta se perderem, o cliente pode entrar em timeout e tentar novamente." (Página 341, parágrafo 2)

Q7. Qual a taxa de bits para transmissão de quadros em cores não compactadas de 1200 x 800 pixels com 16bits/pixel a 50 quadros por segundo? (1pt)

R: A taxa de bits pode ser calculada multiplicando a resolução (1200 x 800 pixels), a profundidade de cor (16 bits/pixel) e a taxa de quadros (50 quadros por segundo). Isso resulta em:

$1200 * 800 * 16 * 50 = 768.000.000$ bits por segundo ou 768 Mbps.

Trecho do livro: "Já deve estar claro, pela nossa discussão sobre vídeo digital, que a compressão é fundamental para o envio de vídeo pela Internet. Até mesmo o vídeo com quadros de qualidade-padrão de 640 por 480 pixels, 24 bits de informações de cor por pixel e 30 quadros/s exige mais de 200 Mbps." (Página 444, parágrafo 3)

Q8. Associe corretamente cada campo de cabeçalho ao respectivo protocolo: DNS (Domain Name System) ou HTTP (Hypertext Transfer Protocol). Também, explique sua finalidade e possíveis valores. (3 pontos)

Campo 1: Host HTTP

Campo 2: User-Agent HTTP

Campo 3: Query ID DNS

Campo 4: Status Code HTTP

Campo 5: Answer RRs DNS

DNS Campos (números):

3 - Query ID:

Finalidade: Identificador único para uma consulta DNS.

Possíveis Valores: Números inteiros gerados aleatoriamente para identificar uma consulta específica.

Trecho do livro: “Quando um resolvedor repassa um nome de domínio ao DNS, o que ele obtém são os registros de recursos associados àquele nome. Portanto, a principal função do DNS é mapear nomes de domínios em registros de recursos.” (página 387, parágrafo 3)

5 - Answer RRs:

Finalidade: Indica o número de registros de recurso na resposta DNS.

Possíveis Valores: tupla RRs(Nome_domínio, Tempo_de_vida, Classe, Tipo, Valor) (página 387, parágrafo 4).

Trecho do livro: “Um registro de recurso é uma tupla de cinco campos. Apesar de serem codificados em binário para proporcionar maior eficiência, na maioria das exposições os registros de recursos são mostrados como texto ASCII, uma linha para cada registro de recurso.” (página 387, parágrafo 4)

HTTP Campos (números):

1 - Host:

Finalidade: Host é o cabeçalho que identifica o servidor.

Possíveis Valores: O valor é geralmente o nome do domínio ou o endereço IP, e pode incluir um número de porta.

Trecho do livro: Host é o cabeçalho que identifica o servidor. Ele é retirado do URL. Esse cabeçalho é obrigatório, e é usado porque alguns endereços IP podem ter múltiplos nomes no DNS, e o servidor precisa ter algum meio de identificar o host a quem deve entregar a solicitação. (Página 433, parágrafo 5)

2 - User Agent

Finalidade: Permite ao cliente informar ao servidor sobre sua implementação de navegador.

Possíveis Valores: Os valores variam amplamente dependendo do navegador e dispositivo usados para fazer a solicitação, por exemplo, Mozilla/5.0 e Chrome/5.0.375.125.

Trecho do livro: O cabeçalho User-Agent permite ao cliente informar ao servidor sobre sua implementação de navegador (por exemplo, Mozilla/5.0 e Chrome/5.0.375.125). Essa informação é útil para permitir que os servidores ajustem suas respostas, pois diferentes navegadores podem ter capacidades e comportamentos variados. (Página 433, parágrafo 2)

A técnica que cada vez é mais utilizada é usar os mesmos protocolos da Web para dispositivos móveis e de desktop, fazendo com que os sites entreguem conteúdo adaptado para o dispositivo móvel quando o usuário está em tal dispositivo. Os servidores Web são

capazes de detectar se devem retornar versões das páginas Web para desktop ou dispositivo móvel examinando os cabeçalhos de solicitação. Um cabeçalho User-Agent é útil especialmente nesse aspecto, pois identifica o software navegador. Assim, quando um servidor Web recebe uma solicitação, ele pode examinar os cabeçalhos e retornar uma página com imagens pequenas, menos texto e navegação mais simples para um iPhone e uma página com todos os recursos para um usuário em um notebook.

(Página 436, 7.3.5, parágrafo 5)

4 - Status Code:

Finalidade: Informar o código de status da resposta HTTP, indicando o resultado do processamento da requisição.

Possíveis Valores:

| Código | Significado | Exemplos |
|--------|------------------|--|
| 1xx | Informação | 100 = servidor concorda em tratar da solicitação do cliente |
| 2xx | Sucesso | 200 = solicitação com sucesso; 204 = nenhum conteúdo presente |
| 3xx | Redirecionamento | 301 = página movida; 304 = página em cache ainda válida |
| 4xx | Erro do cliente | 403 = página proibida; 404 = página não localizada |
| 5xx | Erro do servidor | 500 = erro interno do servidor; 503 = tente novamente mais tarde |

Tabela 7.13 | Os grupos de respostas de código de status.

(Página 433, tabela 7.13 1 Os grupos de respostas de código de status.)

Trecho do livro: “Toda solicitação obtém uma resposta que consiste em uma linha de status e, possivelmente, informações adicionais (por exemplo, uma página Web ou parte dela). A linha de status contém um código de status de três dígitos informando se a solicitação foi atendida e, se não foi, por que não. O primeiro dígito é usado para dividir as respostas em cinco grupos importantes, como mostra a Tabela 7.13. Os códigos 1 xx são raramente usados na prática. Os códigos 2xx significam que a solicitação foi tratada com sucesso, e que o conteúdo (se houver) está sendo retornado. Os códigos 3xx informam ao cliente que ele deve procurar em outro lugar, usando um URL diferente ou seu próprio cache (conforme descreveremos mais adiante). Os códigos 4xx significam que a solicitação falhou devido a um erro do cliente, como uma solicitação inválida ou uma página inexistente. Por fim, os erros 5xx significam que o próprio servidor tem um problema interno, seja ele causado por um erro em seu código ou por uma sobrecarga temporária.” (Página 432, parágrafo 11)