

UNIVERSIDADE FEDERAL DE MINAS GERAIS
DEPARTAMENTO DE ENGENHARIA ELETRÔNICA

UF *mg* G



UNIVERSIDADE FEDERAL
DE MINAS GERAIS

ELT091– TURMA TEE

ESTUDOS DIRIGIDOS REDES TCP/IP
ESTUDO DIRIGIDO EM GRUPO

Guilherme Astolfo Rigacci
Augusto Ribeiro
Matheus Miranda

3 de junho de 2025

Estudos dirigidos redes TCP/IP

Estudo dirigido em grupo

Estudo dirigido 4

Autores:

Guilherme Astolfo Rigacci

Augusto Ribeiro

Matheus Miranda

Prof. Luciano de Errico

3 de junho de 2025

Sumário

1	Questões	1
1.1	Questão 1	1
1.2	Questão 2	3
1.3	Questão 3	6

1 Questões

1.1 Questão 1

Leia o texto sobre ATM da Seção 3.1.2 (p. 112-115, incluindo a nota “Onde estão eles agora?” sobre ATM) e a Seção 4.3 do livro-texto e responda:

- a. Explique resumidamente os princípios básicos da tecnologia ATM. Responda: por que ATM não se tornou a tecnologia dominante em redes?
- b. O conceito básico do MPLS é o de “roteamento baseado em rótulos”. Explique como isso funciona e quais são as vantagens.
- c. Em que consiste o “roteamento explícito”, que vantagens ele traz e como o MPLS pode implementá-lo?
- d. Explique como o MPLS pode ser usado para implementar uma VPN de camada 3.

Resposta:

a. **Princípios básicos da tecnologia ATM:**

A tecnologia ATM (Asynchronous Transfer Mode) foi projetada para integrar diferentes tipos de tráfego (voz, vídeo, dados) em uma rede unificada, com baixa latência e alta previsibilidade. Seu funcionamento baseia-se no uso de comutação de pacotes fixos de 53 bytes (células), o que facilita o tratamento em hardware e garante desempenho constante. ATM usa comutação orientada à conexão, sendo necessário estabelecer um caminho virtual antes da transmissão de dados.

Por que ATM não se tornou dominante:

ATM foi superada por tecnologias baseadas em IP devido à sua complexidade, alto custo de implementação e necessidade de redes especializadas. Além disso, o avanço de redes Ethernet de alta velocidade e o crescimento da Internet, baseada em protocolos IP, tornaram o modelo de transporte unificado do ATM menos atrativo. A flexibilidade, simplicidade e ampla adoção do IP acabaram prevalecendo.

b. Roteamento baseado em rótulos no MPLS:

O MPLS (Multiprotocol Label Switching) funciona atribuindo um rótulo (label) a cada pacote no início do seu caminho na rede. Em vez de analisar o cabeçalho IP em cada salto, os roteadores (neste caso chamados de LSRs – Label Switch Routers) usam esse rótulo para encaminhar o pacote de forma eficiente, comutando com base em tabelas de rótulos pré-estabelecidas.

Vantagens:

- Redução da carga de processamento nos roteadores.
- Encaminhamento mais rápido.
- Suporte eficiente a QoS (Qualidade de Serviço).
- Capacidade de engenharia de tráfego (traffic engineering).
- Integração com múltiplos protocolos de rede.

c. Roteamento explícito e sua implementação via MPLS:

Roteamento explícito é a técnica onde o caminho completo ou parcial que o pacote deve seguir é definido previamente, em vez de ser calculado dinamicamente por protocolos de roteamento tradicionais.

Vantagens:

- Melhor controle sobre o tráfego na rede.
- Otimização da utilização de links.
- Redução de congestionamento e melhoria de desempenho.

O MPLS permite o roteamento explícito ao permitir que os rótulos sejam atribuídos com base em caminhos específicos configurados manualmente ou por algoritmos de engenharia de tráfego (por exemplo, usando RSVP-TE ou CR-LDP).

d. Implementação de VPN de camada 3 com MPLS:

O MPLS pode ser usado para construir VPNs de camada 3 (IP VPNs) através da utilização de rótulos empilhados. Cada pacote IP do cliente é encapsulado com dois rótulos MPLS: um identifica o túnel (path) até o roteador de borda de saída (PE), e o outro identifica a VPN específica (via tabelas de

encaminhamento VRF – Virtual Routing and Forwarding).

Isso permite isolar o tráfego entre diferentes clientes, manter tabelas de roteamento separadas e suportar múltiplos serviços sobre uma mesma infraestrutura compartilhada, de maneira segura e escalável.

1.2 Questão 2

Calcule a vazão para cada um dos casos abaixo:

- Stop and Wait, $RTT = 8$ ms, $BW = 1$ Mbps, tamanho do pacote = 1000 bytes;
- Go Back N, $W = 2$ pacotes, $RTT = 8$ ms, $BW = 1$ Mbps, tamanho do pacote = 1000 bytes;
- Stop and Wait, $RTT = 500$ ms, $BW = 1$ Mbps, tamanho do pacote = 1000 bytes;
- Go Back N, $W = 2$ pacotes, $RTT = 500$ ms, $BW = 1$ Mbps, tamanho do pacote = 1000 bytes;
- para o cenário em (d), qual seria o tamanho mínimo de janela para que se conseguisse atingir 100% de utilização?

Resposta:

Primeiro, vamos definir algumas variáveis e calcular valores comuns que serão utilizados em todos os itens:

- Tamanho do pacote (L): 1000 bytes. Como 1 byte = 8 bits, então $L = 1000 \times 8$ bits = 8000 bits.
- Largura de banda (BW ou R): 1 Mbps. Como 1 Mbps = 1×10^6 bits por segundo (bps), então $R = 1 \times 10^6$ bps.
- Tempo de transmissão do pacote (T_{trans}): Este é o tempo necessário para colocar todos os bits de um pacote no link. É calculado como $T_{trans} = \frac{L}{R}$.
$$T_{trans} = \frac{8000 \text{ bits}}{1 \times 10^6 \text{ bps}} = 0.008 \text{ segundos} = 8 \text{ milissegundos (ms)}.$$

A vazão (Throughput, Th) é a taxa efetiva de transferência de dados. Ela pode ser calculada como o produto da Utilização do canal (U) pela Largura de Banda (R):
 $Th = U \times R$.

a. **Stop and Wait, RTT = 8 ms, BW = 1 Mbps, tamanho do pacote = 1000 bytes**

- Round-Trip Time (RTT) = 8 ms = 0.008 s.
- Tempo de transmissão (T_{trans}) = 8 ms = 0.008 s (calculado acima).
- No protocolo Stop and Wait, o transmissor envia um pacote e espera por um reconhecimento (ACK) antes de enviar o próximo. O tempo total para um ciclo completo (enviar pacote e receber ACK) é $RTT + T_{trans}$. Durante este ciclo, o transmissor está efetivamente transmitindo dados apenas durante T_{trans} .
- A utilização (U_{SW}) é a fração do tempo em que o transmissor está ocupado: $U_{SW} = \frac{T_{trans}}{RTT + T_{trans}}$
- Substituindo os valores: $U_{SW} = \frac{0.008 \text{ s}}{0.008 \text{ s} + 0.008 \text{ s}} = \frac{0.008 \text{ s}}{0.016 \text{ s}} = 0.5$
- A vazão (Th_{SW}) é: $Th_{SW} = U_{SW} \times R = 0.5 \times 1 \text{ Mbps} = 0.5 \text{ Mbps}$ (ou 500 kbps).

b. **Go Back N, W = 2 pacotes, RTT = 8 ms, BW = 1 Mbps, tamanho do pacote = 1000 bytes**

- Janela de envio (W) = 2 pacotes.
- RTT = 8 ms = 0.008 s.
- $T_{trans} = 8 \text{ ms} = 0.008 \text{ s}$.
- No Go Back N, o transmissor pode enviar até W pacotes sem esperar por ACKs individuais. O canal pode ser mantido ocupado se a quantidade de dados enviados dentro da janela ($W \times L$) for suficiente para preencher o "pipe" de dados durante o tempo $RTT + T_{trans}$.
- A utilização (U_{GBN}) é dada por (considerando um cenário ideal sem perdas): $U_{GBN} = \min \left(1, \frac{W \times T_{trans}}{RTT + T_{trans}} \right)$
- Primeiro, calculamos $W \times T_{trans}$: $W \times T_{trans} = 2 \times 0.008 \text{ s} = 0.016 \text{ s}$.
- E $RTT + T_{trans}$: $RTT + T_{trans} = 0.008 \text{ s} + 0.008 \text{ s} = 0.016 \text{ s}$.
- Como $W \times T_{trans} = RTT + T_{trans}$ (ou seja, $0.016 \text{ s} = 0.016 \text{ s}$), o transmissor pode manter o canal 100% ocupado. $U_{GBN} = \frac{0.016 \text{ s}}{0.016 \text{ s}} = 1$
- A vazão (Th_{GBN}) é: $Th_{GBN} = U_{GBN} \times R = 1 \times 1 \text{ Mbps} = 1 \text{ Mbps}$.

c. Stop and Wait, RTT = 500 ms, BW = 1 Mbps, tamanho do pacote = 1000 bytes

- $RTT = 500 \text{ ms} = 0.5 \text{ s}$.
- $T_{trans} = 8 \text{ ms} = 0.008 \text{ s}$.
- Usando a mesma fórmula para U_{SW} do item (a): $U_{SW} = \frac{T_{trans}}{RTT + T_{trans}}$
- Substituindo os valores: $U_{SW} = \frac{0.008 \text{ s}}{0.5 \text{ s} + 0.008 \text{ s}} = \frac{0.008 \text{ s}}{0.508 \text{ s}} \approx 0.01574803$
- A vazão (Th_{SW}) é: $Th_{SW} = U_{SW} \times R \approx 0.01574803 \times 1 \text{ Mbps} \approx 0.015748 \text{ Mbps}$ (ou aproximadamente 15.75 kbps).

d. Go Back N, W = 2 pacotes, RTT = 500 ms, BW = 1 Mbps, tamanho do pacote = 1000 bytes

- Janela de envio (W) = 2 pacotes.
- $RTT = 500 \text{ ms} = 0.5 \text{ s}$.
- $T_{trans} = 8 \text{ ms} = 0.008 \text{ s}$.
- Usando a mesma fórmula para U_{GBN} do item (b): $U_{GBN} = \min \left(1, \frac{W \times T_{trans}}{RTT + T_{trans}} \right)$
- Calculando $W \times T_{trans}$: $W \times T_{trans} = 2 \times 0.008 \text{ s} = 0.016 \text{ s}$.
- Calculando $RTT + T_{trans}$: $RTT + T_{trans} = 0.5 \text{ s} + 0.008 \text{ s} = 0.508 \text{ s}$.
- Como $W \times T_{trans} < RTT + T_{trans}$ (ou seja, $0.016 \text{ s} < 0.508 \text{ s}$), a janela não é grande o suficiente para manter o pipe cheio. $U_{GBN} = \frac{0.016 \text{ s}}{0.508 \text{ s}} \approx 0.03149606$
- A vazão (Th_{GBN}) é: $Th_{GBN} = U_{GBN} \times R \approx 0.03149606 \times 1 \text{ Mbps} \approx 0.031496 \text{ Mbps}$ (ou aproximadamente 31.50 kbps).

e. Para o cenário em (d), qual seria o tamanho mínimo de janela para que se conseguisse atingir 100% de utilização?

- Temos $RTT = 500 \text{ ms} = 0.5 \text{ s}$.
- E $T_{trans} = 8 \text{ ms} = 0.008 \text{ s}$.
- Para atingir 100% de utilização ($U = 1$), o transmissor deve ser capaz de enviar pacotes continuamente, preenchendo o "pipe" de comunicação. O tempo total do ciclo para receber um ACK para o primeiro pacote de uma janela (e assim poder deslizar a janela) é $RTT + T_{trans}$.

- Durante este tempo $RTT + T_{trans}$, o número de pacotes que poderiam ter sido transmitidos se o canal estivesse continuamente ocupado é $\frac{RTT+T_{trans}}{T_{trans}}$.
- Para garantir que o pipe esteja sempre cheio, a janela W deve ser grande o suficiente para cobrir todos esses pacotes.
- $W_{min} = \frac{RTT+T_{trans}}{T_{trans}}$
- Substituindo os valores: $W_{min} = \frac{0.5\text{ s}+0.008\text{ s}}{0.008\text{ s}} = \frac{0.508\text{ s}}{0.008\text{ s}} = 63.5$
- Como o tamanho da janela (W) deve ser um número inteiro de pacotes, e precisamos garantir que o pipe esteja sempre cheio (ou seja, que o transmissor possa enviar pelo menos 63.5 pacotes antes de ter que parar para esperar um ACK), devemos arredondar para cima.
- Portanto, o tamanho mínimo da janela W_{min} é $\lceil 63.5 \rceil = 64$ pacotes.
- *Verificação alternativa usando Produto Largura de Banda-Atraso (BDP):* O BDP é a quantidade de dados que podem estar "em trânsito" no link: $BDP = R \times RTT = 1 \times 10^6 \text{ bps} \times 0.5 \text{ s} = 500000 \text{ bits}$. O número de pacotes que "cabem" no pipe é $N_{pipe} = \frac{BDP}{L} = \frac{500000 \text{ bits}}{8000 \text{ bits/pacote}} = 62.5$ pacotes. Para manter o pipe cheio continuamente, a janela de envio W deve ser grande o suficiente para cobrir esses N_{pipe} pacotes mais o pacote que está sendo atualmente transmitido. Uma forma comum de expressar isso é $W \geq \frac{R \times RTT}{L} + 1$ (onde o +1 representa o pacote que está sendo transmitido enquanto os ACKs dos pacotes no pipe ainda não chegaram). $W \geq 62.5 + 1 = 63.5$. Arredondando para cima, $W_{min} = 64$ pacotes. A fórmula $W_{min} = \lceil \frac{RTT+T_{trans}}{T_{trans}} \rceil$ é mais precisa pois representa diretamente o número de "slots" de transmissão de pacote que ocorrem em um ciclo $RTT + T_{trans}$.

1.3 Questão 3

Leia o artigo "End-to-End Arguments in System Design" e responda: Muitos opositores do argumento fim a fim afirmam, entre outras coisas, que este apenas diz que "redes devem ser o mais simples e o mais estúpidas possíveis". Você concorda? Defensores deste argumento indicam o sucesso da Internet como prova de que o argumento fim a fim é válido. Você concorda? Justifique suas respostas.

Resposta:

A ideia central, conforme o artigo original, é que funções específicas de uma aplicação devem ser implementadas nas extremidades (nos sistemas finais ou aplicações) e não nos nós intermediários da rede, a menos que haja razões de desempenho muito fortes para fazê-lo.

A lógica é que os pontos finais (as aplicações) são os únicos que têm o conhecimento completo dos requisitos da função. Tentar implementar essa funcionalidade em um nível inferior (na rede) muitas vezes resulta em uma implementação incompleta, redundante ou que não atende plenamente às necessidades da aplicação. A rede, nesse contexto, deve focar em sua tarefa primária: mover dados de um ponto a outro da forma mais eficiente e geral possível.

Desta forma, a expressão do sucesso do argumento ser a Internet é válida, pois a arquitetura da Internet foi projetada com base nesse princípio. A rede é simples e flexível, permitindo que novas aplicações sejam desenvolvidas sem a necessidade de modificações na infraestrutura da rede. Isso possibilitou uma inovação rápida e contínua, resultando em uma vasta gama de serviços e aplicações que utilizam a Internet.

Logo, a implementação de um protocolo fim a fim, se provou como uma abordagem eficaz para garantir que as aplicações possam evoluir e se adaptar às necessidades dos usuários sem depender de mudanças na infraestrutura da rede.