

# Activity Week 3

# Table of Contents

|   |    |
|---|----|
| Tipos de Entrega de Mensagens .....                   | 2  |
| Protocolos da Camada de Transporte: TCP vs. UDP ..... | 7  |
| O Modelo OSI e a Arquitetura de Protocolos .....      | 12 |
| ARPANET: O Berço da Internet .....                    | 17 |
| Conclusão .....                                       | 20 |
| Referências .....                                     | 21 |

# Tipos de Entrega de Mensagens

Em redes de computadores, a transmissão de pacotes de uma origem para um ou mais destinos é categorizada em três modos fundamentais de entrega: Unicast, Broadcast e Multicast. Cada modo possui características técnicas, mecanismos de endereçamento e casos de uso distintos que determinam sua eficiência e aplicabilidade.

## Unicast

O modo Unicast representa a forma mais fundamental de comunicação em uma rede: uma transmissão de um ponto a outro.

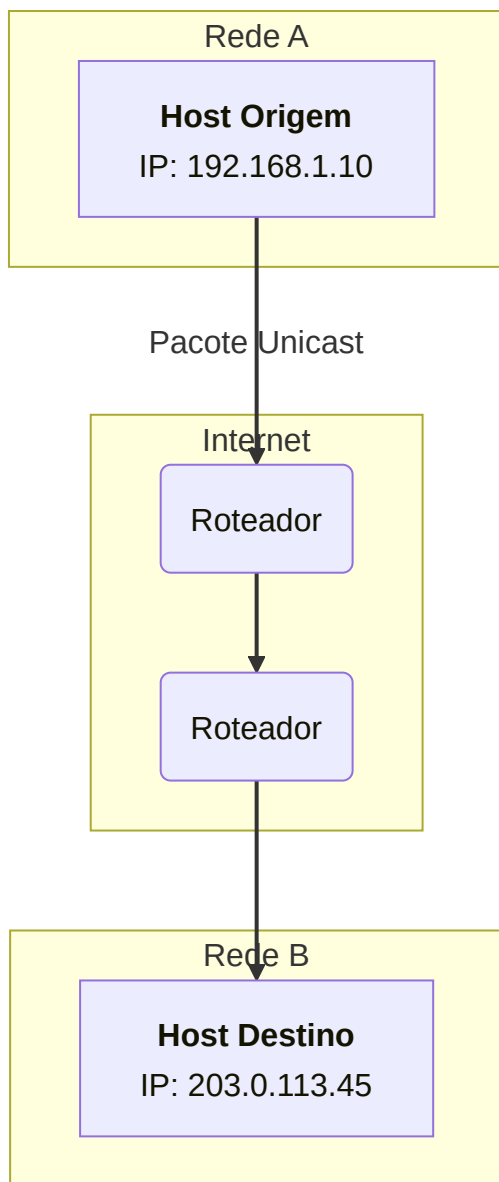
### Detalhes Técnicos

Em uma transmissão Unicast, um único pacote é gerado por um host de origem e endereçado a um único host de destino. Cada host na rede possui um endereço único (como um endereço IP e um endereço MAC), e os pacotes Unicast contêm o endereço específico do host de destino em seu cabeçalho.

Os dispositivos intermediários, como switches e roteadores, utilizam esse endereço de destino para encaminhar o pacote através da rede pela rota mais eficiente até que ele alcance seu destino final. Esta é a base para a grande maioria do tráfego na Internet, incluindo navegação em sites (HTTP), envio de e-mails (SMTP) e transferência de arquivos (FTP).

### Diagrama de Comunicação Unicast

O diagrama abaixo ilustra um host de origem enviando um pacote para um único host de destino.



## Broadcast

O modo Broadcast é um mecanismo de comunicação de um para todos, onde um único pacote de origem é entregue a todos os hosts dentro de um mesmo domínio de rede.

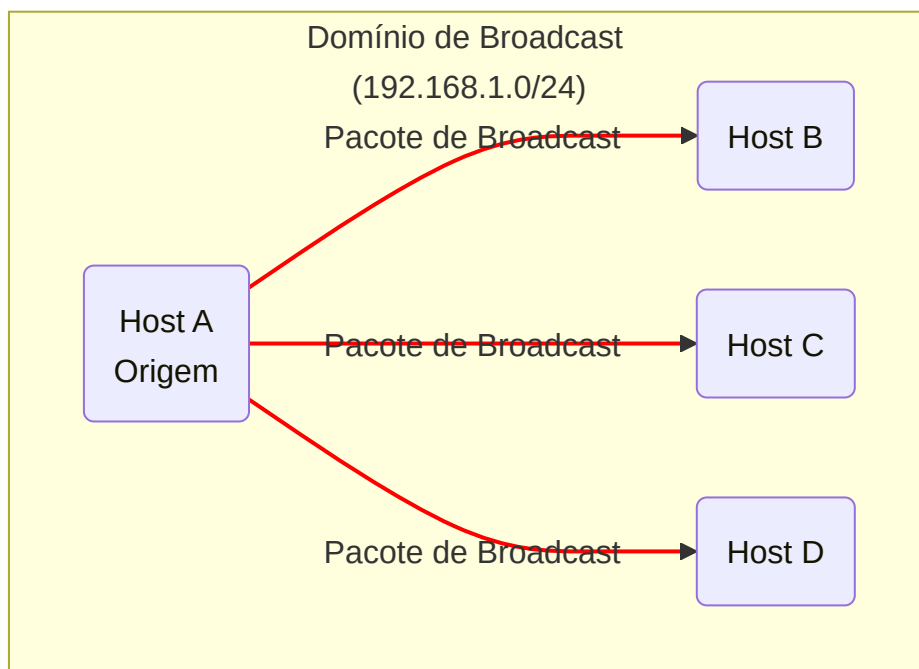
### Detalhes Técnicos

Uma transmissão Broadcast utiliza um endereço especial de destino. Na Camada 2 (Enlace), o endereço MAC de destino é `FF:FF:FF:FF:FF:FF`. Na Camada 3 (Rede), para IPv4, utiliza-se um endereço de broadcast, que geralmente é o endereço mais alto de uma sub-rede (por exemplo, `192.168.1.255` em uma rede `192.168.1.0/24`).

Quando um switch recebe um quadro (frame) com o MAC de broadcast, ele o replica para todas as suas portas, exceto a porta de origem. Todos os hosts no domínio de broadcast recebem e processam o pacote, o que pode gerar uma carga de processamento significativa em todos os dispositivos da rede. Por essa razão, os roteadores, por padrão, não encaminham pacotes de broadcast, confinando-os à rede local (domínio de broadcast).

## Diagrama de Comunicação Broadcast

Este diagrama mostra um host enviando um pacote de broadcast que é recebido por todos os outros hosts na mesma rede local.



## Casos de Uso

- **ARP (Address Resolution Protocol):** Um host envia uma requisição ARP em broadcast para descobrir o endereço MAC correspondente a um endereço IP na rede local.
- **DHCP (Dynamic Host Configuration Protocol):** Um cliente que acabou de entrar na rede envia uma mensagem DHCPDISCOVER em broadcast para encontrar um servidor DHCP.

## Multicast

O modo Multicast é um modelo de comunicação de um para muitos, que combina a eficiência do broadcast com o direcionamento do unicast.

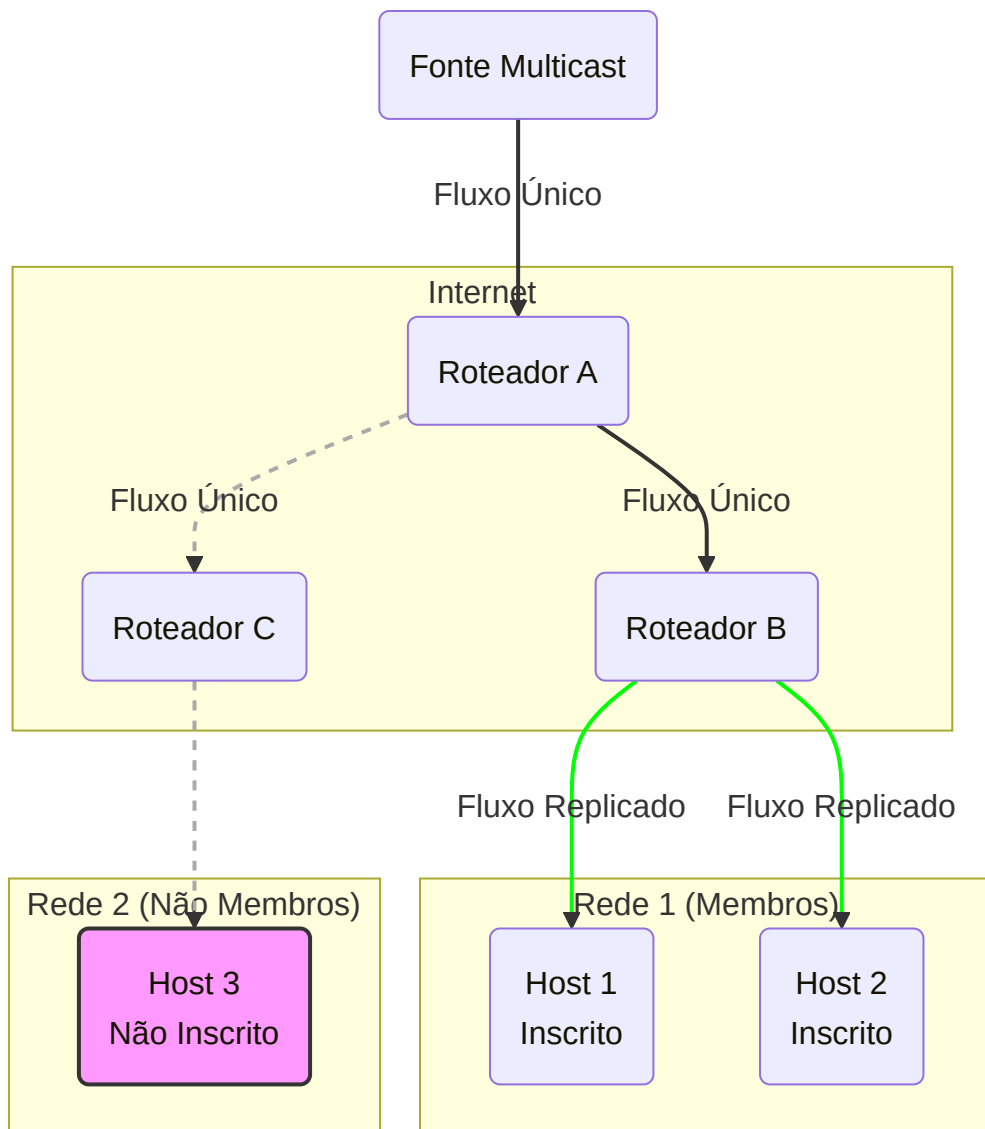
## Detalhes Técnicos

O Multicast permite que uma fonte envie um único pacote que é replicado e entregue apenas a um grupo específico de hosts "interessados" em recebê-lo. Isso otimiza drasticamente o uso da largura de banda em comparação com o envio de múltiplos pacotes unicast.

- **Endereçamento Multicast:** Utiliza-se um bloco especial de endereços. Em IPv4, são os endereços de Classe D, no intervalo de 224.0.0.0 a 239.255.255.255. Cada endereço nesse intervalo representa um "grupo multicast".
- **Gerenciamento de Grupos (IGMP):** Os hosts que desejam receber o tráfego de um grupo multicast precisam se inscrever nele. Eles fazem isso usando o **IGMP (Internet Group Management Protocol)**. O host envia uma mensagem IGMP para seu roteador local, informando seu interesse. O roteador, por sua vez, passa a encaminhar o tráfego daquele grupo para a interface de rede onde o host está conectado. Quando um host não quer mais receber os pacotes, ele envia uma mensagem de "saída" (leave).
- **Roteamento Multicast:** Roteadores utilizam protocolos de roteamento especiais, como o **PIM (Protocol-Independent Multicast)**, para construir árvores de distribuição que garantem que os pacotes multicast sejam encaminhados apenas pelas rotas que levam a membros do grupo.

## Diagrama de Comunicação Multicast

O diagrama ilustra uma fonte enviando um único fluxo de dados. Os roteadores replicam o fluxo apenas para as redes que contêm hosts inscritos no grupo multicast.



## Casos de Uso

- **IPTV:** Transmissão de canais de televisão pela Internet para múltiplos assinantes.
- **Streaming de Eventos ao Vivo:** Distribuição de vídeo e áudio para um grande público.
- **Jogos Online Massivos:** Sincronização do estado do jogo entre múltiplos jogadores.
- **Mercado Financeiro:** Distribuição de cotações de ações em tempo real para terminais de operadores.

# Protocolos da Camada de Transporte: TCP vs. UDP

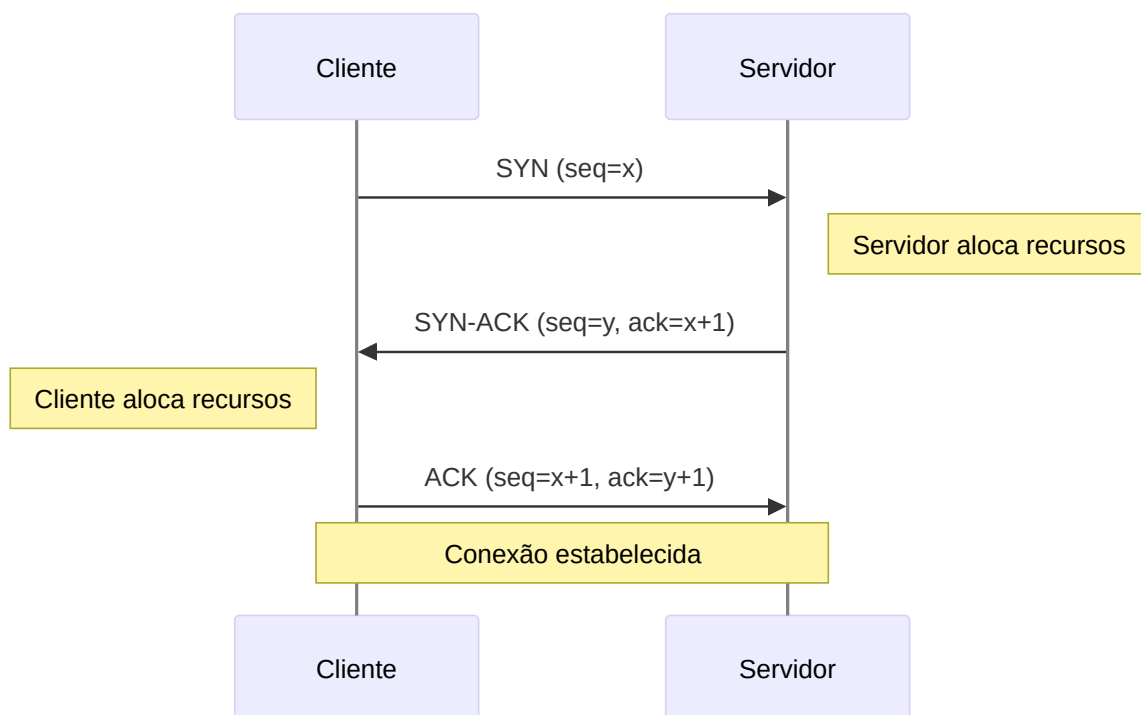
A Camada de Transporte (Camada 4 do Modelo OSI) é responsável por fornecer a comunicação lógica ponta a ponta (host-a-host) para as aplicações. Ela utiliza os serviços da Camada de Rede (IP) para mover pacotes, mas adiciona funcionalidades cruciais como a multiplexação de portas e, dependendo do protocolo, confiabilidade e controle de fluxo. Os dois protocolos predominantes nesta camada são o TCP e o UDP.

## TCP (Transmission Control Protocol)

O TCP é o protocolo que garante a confiabilidade na entrega de dados. Ele foi projetado para assegurar que cada byte enviado pela aplicação de origem chegue intacto e na ordem correta à aplicação de destino, mesmo em redes não confiáveis.

### Características e Mecanismos

- **Orientado à Conexão:** Antes de qualquer troca de dados, o TCP estabelece uma conexão virtual entre os dois hosts através de um processo conhecido como **Three-Way Handshake**.

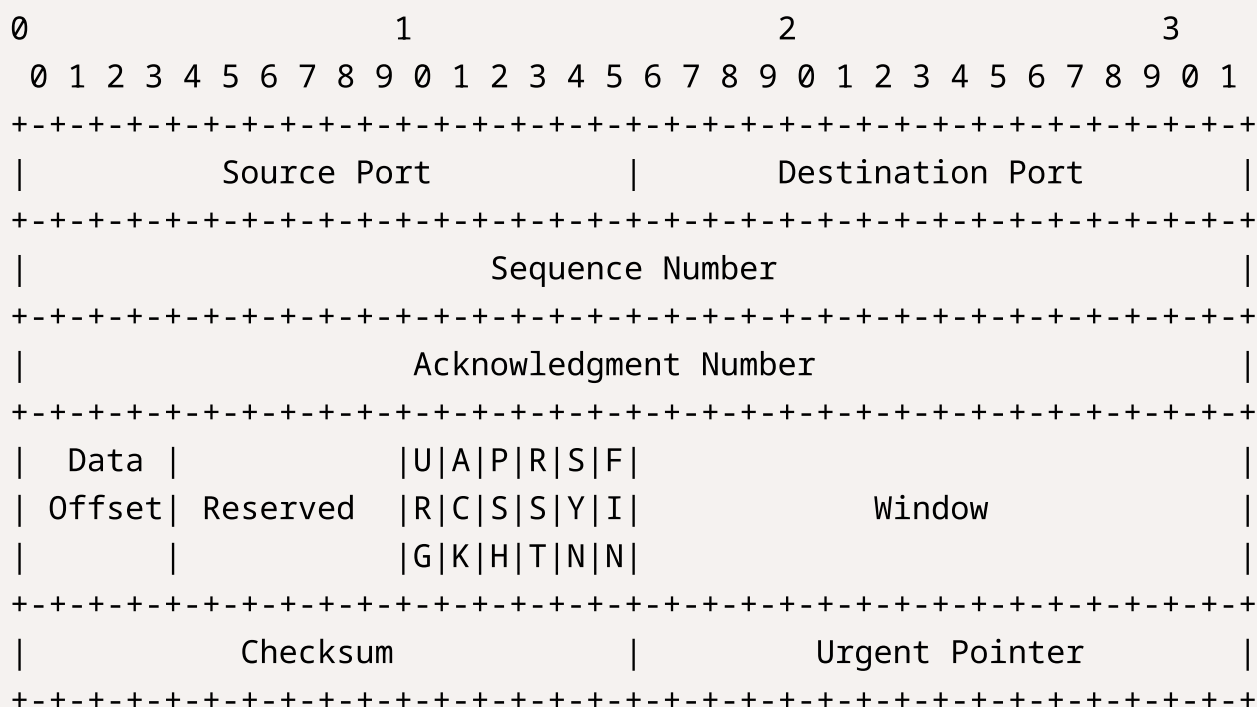


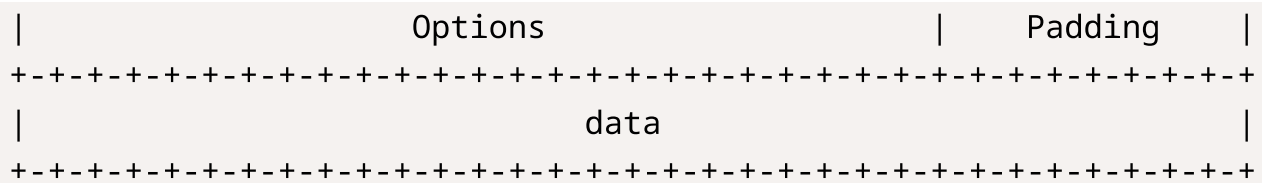


- **Entrega Confiável e Ordenada:** O TCP numera os segmentos de dados com um número de sequência (Sequence Number). O receptor utiliza esses números para reordenar os segmentos que chegam fora de ordem e para identificar segmentos perdidos. A confirmação de recebimento (Acknowledgement - ACK) informa ao remetente quais dados foram recebidos com sucesso.
- **Controle de Fluxo (Flow Control):** O receptor utiliza um mecanismo de "janela deslizante" (Sliding Window) para controlar a quantidade de dados que o remetente pode enviar. O tamanho da janela (anunciado no campo *Window Size* do cabeçalho TCP) informa ao remetente o espaço disponível no buffer do receptor, evitando que o remetente sobrecarregue o destinatário.
- **Controle de Congestionamento (Congestion Control):** Para evitar o congestionamento da rede (e não apenas do host receptor), o TCP utiliza algoritmos como *Slow Start* e *Congestion Avoidance*. Ele monitora a perda de pacotes e os tempos de ida e volta (RTT) para inferir o nível de congestionamento da rede e ajusta sua taxa de transmissão dinamicamente.

## Estrutura do Cabeçalho TCP

O cabeçalho TCP é complexo, refletindo suas múltiplas funcionalidades. Seu tamanho mínimo é de 20 bytes.





## UDP (User Datagram Protocol)

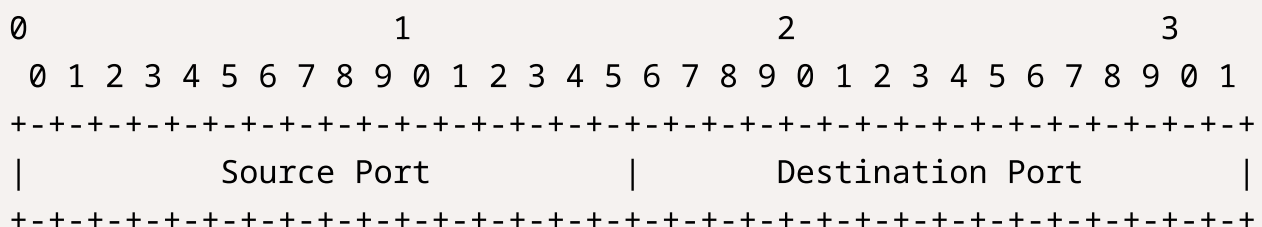
O UDP é a antítese do TCP. Ele oferece um serviço de entrega de datagramas simples, não orientado à conexão e não confiável. Sua principal vantagem é a velocidade e o baixo overhead.

### Características e Mecanismos

- **Não Orientado à Conexão:** Não há estabelecimento de conexão. Um datagrama UDP é simplesmente enviado ao destino assim que a aplicação o entrega à camada de transporte. É um modelo "dispare e esqueça" (*fire and forget*).
- **Entrega Não Confiável:** O UDP não garante a entrega, a ordem ou a unicidade dos datagramas. Se um datagrama se perde, ele não é retransmitido. Se chegam fora de ordem, a aplicação de destino é quem deve lidar com isso.
- **Sem Controle de Fluxo ou Congestionamento:** O UDP envia os dados na velocidade que a aplicação os produz, sem se preocupar com a capacidade do receptor ou da rede. Isso pode levar à perda de pacotes se a taxa de envio for muito alta.
- **Checksum:** O UDP possui um campo de checksum para verificação de erros no cabeçalho e nos dados. Se o checksum falhar, o datagrama geralmente é descartado silenciosamente pelo sistema operacional do receptor.

### Estrutura do Cabeçalho UDP

O cabeçalho UDP é extremamente simples, com apenas 8 bytes, o que contribui para seu baixo overhead.





## Comparativo e Casos de Uso

| Característica    | TCP (Transmission Control Protocol)  | UDP (User Datagram Protocol)                |
|-------------------|--------------------------------------|---|
| Conexão           | Orientado à conexão (Handshake)      | Não orientado à conexão                     |
| Confiabilidade    | Alta (confirmações e retransmissões) | Baixa (entrega "best-effort")               |
| Ordenação         | Garante a ordem dos segmentos        | Não garante a ordem dos datagramas          |
| Controle de Fluxo | Sim (Sliding Window)                 | Não   |
| Controle de Cong. | Sim (Slow Start, etc.)               | Não   |
| Velocidade        | Mais lento                           | Muito rápido                                |
| Overhead          | Alto (cabeçalho de 20+ bytes)        | Baixo (cabeçalho de 8 bytes)                |
| Casos de Uso      | HTTP, FTP, E-mail (SMTP), SSH        | DNS, VoIP, Streaming de vídeo, Jogos Online |

## Justificativa de Escolha

- **Download de Arquivos (FTP, HTTP):** A escolha é **TCP**. A integridade do arquivo é absoluta. Cada bit deve ser transferido corretamente e na ordem exata. A sobrecarga do TCP é um preço pequeno a pagar pela garantia de que o arquivo não será corrompido.
- **Jogo Online / Streaming de Vídeo:** A escolha é **UDP**. Nestas aplicações, a latência é o fator crítico. É preferível perder um quadro de vídeo ou uma atualização de posição de um jogador (causando um pequeno "glitch" visual) do que esperar pela retransmissão de um pacote antigo, o que causaria pausas e "lag" (atraso) que arruinariam a experiência em tempo real.

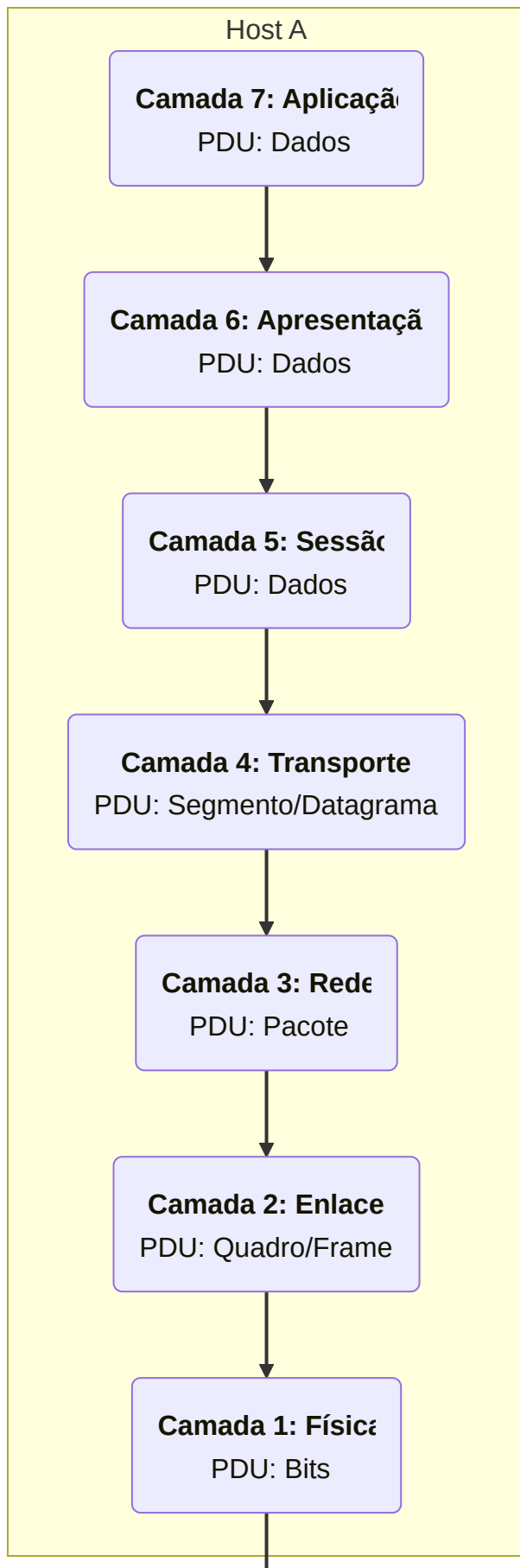
# O Modelo OSI e a Arquitetura de Protocolos

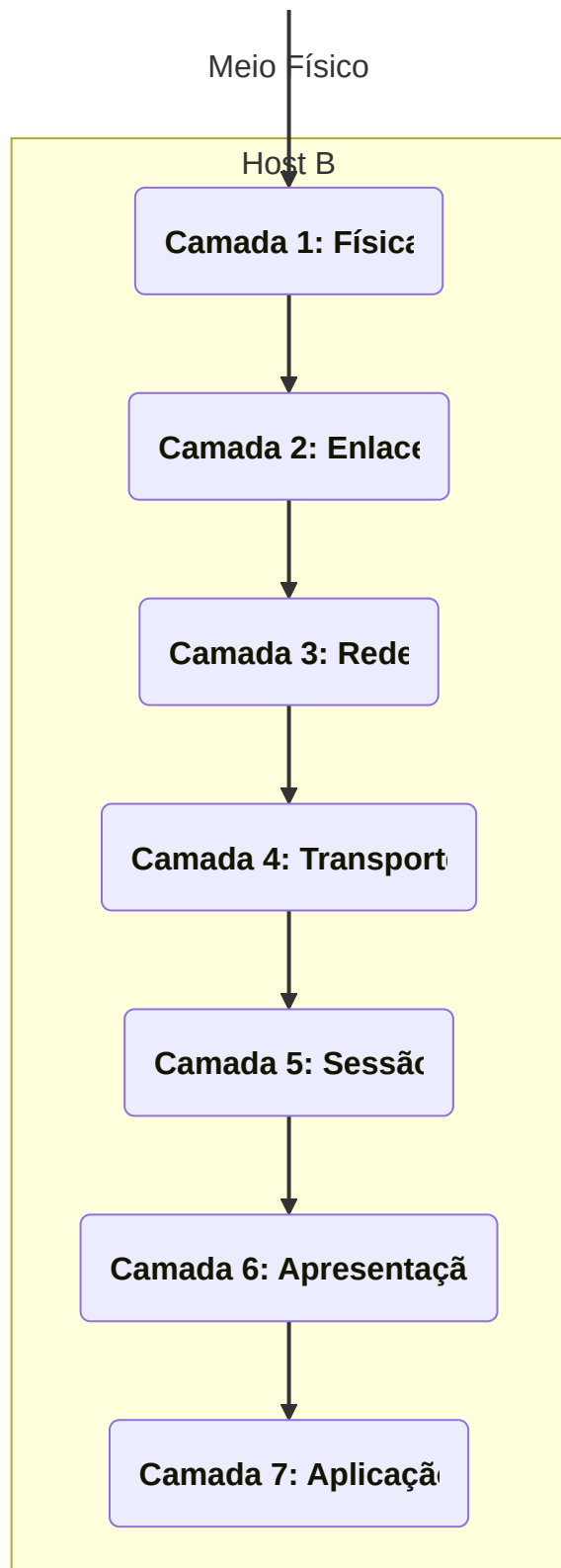
O Modelo OSI (Open Systems Interconnection), desenvolvido pela ISO, é um modelo de referência conceitual que divide as complexas funções de uma rede de computadores em sete camadas de abstração. Ele serve como um guia para o desenvolvimento de padrões e para o ensino de redes, permitindo que especialistas foquem em uma área específica e garantindo a interoperabilidade entre diferentes tecnologias.

Embora o modelo TCP/IP seja o mais implementado na prática, o Modelo OSI continua sendo a principal ferramenta pedagógica para descrever a arquitetura de rede.

## As Sete Camadas do Modelo OSI

O processo de comunicação é dividido em camadas, onde cada uma adiciona um cabeçalho (encapsulamento) aos dados recebidos da camada superior, antes de passá-los para a camada inferior.





1. **Camada 7: Aplicação:** A camada mais próxima do usuário. Fornece a interface para que as aplicações (navegadores, clientes de e-mail) acessem os serviços de rede.

2. **Camada 6: Apresentação:** Garante que os dados sejam "apresentáveis" ao sistema receptor. Lida com a formatação, criptografia e compressão dos dados.
3. **Camada 5: Sessão:** Estabelece, gerencia e encerra as sessões (conexões) entre aplicações. Controla o diálogo e a sincronização.
4. **Camada 4: Transporte:** Responsável pela comunicação ponta a ponta (host-a-host). Oferece serviços de segmentação, controle de fluxo e correção de erros.
5. **Camada 3: Rede:** Lida com o endereçamento lógico (endereços IP) e o roteamento de pacotes através de múltiplas redes para determinar o melhor caminho até o destino.
6. **Camada 2: Enlace de Dados (Link):** Responsável pela transferência de quadros (frames) entre nós adjacentes em uma mesma rede. Lida com o endereçamento físico (endereços MAC) e a detecção de erros no nível do enlace.
7. **Camada 1: Física:** Define as especificações elétricas, mecânicas e funcionais do meio físico de transmissão (cabos, fibras ópticas, sinais de rádio). Transmite os dados como um fluxo de bits.

## Mapeamento de Protocolos por Camada

A seguir, a justificativa técnica para a localização de cada protocolo dentro do modelo.

### Camada 7 (Aplicação)

- **HTTP (Hypertext Transfer Protocol):** Define o formato das requisições e respostas trocadas entre um cliente web (navegador) e um servidor web. Por prover diretamente o serviço de navegação na web ao usuário, é um protocolo de aplicação.
- **FTP (File Transfer Protocol):** Provê o serviço de transferência de arquivos entre hosts. Ele gerencia uma conexão de controle e múltiplas conexões de dados para realizar essa tarefa, sendo uma aplicação de rede clássica.
- **DNS (Domain Name System):** Funciona como uma aplicação distribuída que provê o serviço de tradução de nomes de domínio (ex: `www.google.com`) para endereços IP. Outras aplicações, como o navegador, consultam o DNS para poderem iniciar uma conexão, tornando o DNS um protocolo de aplicação que serve a outras aplicações.

### Camada 4 (Transporte)



- **TCP (Transmission Control Protocol) e UDP (User Datagram Protocol):** Ambos operam aqui, pois sua função é prover um serviço de comunicação para as aplicações da camada superior. Eles não se importam com o conteúdo dos dados, mas sim em como transportá-los de um processo em um host para um processo em outro host. O TCP oferece um transporte confiável e orientado à conexão, enquanto o UDP oferece um transporte rápido e não confiável. Eles gerenciam portas, segmentação e (no caso do TCP) a confiabilidade da entrega.

## Camada 3 (Rede)

- **IP (Internet Protocol):** É o principal protocolo desta camada. Sua única função é o endereçamento lógico e o roteamento de pacotes. Ele insere um cabeçalho com o endereço IP de origem e destino e se esforça para entregar o pacote ao destino final, sem garantias. Ele é a base para a interconexão de redes (a "internet").
- **ICMP (Internet Control Message Protocol):** Atua como um protocolo de suporte para o IP. Ele não transporta dados de aplicação, mas sim mensagens de controle e erro sobre a própria camada de rede. Exemplos incluem o ping (ICMP Echo Request/Reply) e mensagens de "Destination Unreachable" enviadas por roteadores. Por lidar exclusivamente com o estado e problemas da camada de rede, ele pertence a esta camada.

## Interface entre Camadas 2 e 3

- **ARP (Address Resolution Protocol):** Este é um caso especial. A função do ARP é mapear um endereço de Camada 3 (endereço IP) para um endereço de Camada 2 (endereço MAC) dentro de uma mesma rede local. Ele é necessário porque, para entregar um pacote em uma rede local (Ethernet, Wi-Fi), o endereço físico (MAC) do destino é obrigatório. Como ele opera com informações de ambas as camadas e serve como uma "ponte" entre elas, ele não se encaixa perfeitamente em uma única camada, sendo frequentemente descrito como um protocolo de Camada 2.5.

# ARPANET: O Berço da Internet

A ARPANET (Advanced Research Projects Agency Network) foi uma rede de computadores pioneira que serviu como o campo de testes para as tecnologias que hoje formam a espinha dorsal da Internet. Sua criação não foi um mero exercício acadêmico, mas uma resposta a uma necessidade estratégica no auge da Guerra Fria.

## Contexto Histórico e a Necessidade de uma Rede Robusta

Nos anos 60, o Departamento de Defesa dos EUA (DoD) estava preocupado com a vulnerabilidade de seus sistemas de comunicação, que eram centralizados e baseados em comutação de circuitos (como a rede de telefonia). Em um cenário de ataque nuclear, a destruição de um único ponto central poderia incapacitar toda a comunicação.

A ARPA (hoje DARPA) foi encarregada de financiar um projeto para criar uma rede de comando e controle descentralizada, robusta e resiliente, que pudesse sobreviver a danos parciais.

## A Inovação Central: Comutação de Pacotes

A solução para essa necessidade foi a **comutação de pacotes** (*packet switching*), um conceito revolucionário para a época, desenvolvido com base nos trabalhos teóricos de Leonard Kleinrock, Paul Baran e Donald Davies.

Em vez de dedicar um circuito físico entre dois pontos, a comutação de pacotes funciona da seguinte forma:

1. **Fragmentação:** A mensagem original é quebrada em pequenos blocos de tamanho definido, chamados **pacotes**.
2. **Endereçamento:** Cada pacote recebe um cabeçalho contendo informações vitais, como o endereço IP do remetente e do destinatário, e um número de sequência.
3. **Roteamento Independente:** Os pacotes são enviados para a rede individualmente. Cada um pode seguir um caminho diferente até o destino, sendo roteado de nó em nó de forma independente.
4. **Reagrupamento:** Ao chegar ao destino, os pacotes são reagrupados na ordem correta (usando o número de sequência) para reconstruir a mensagem original.

Essa abordagem trouxe duas vantagens cruciais:

- **Resiliência:** Se um nó ou caminho da rede falhasse, os pacotes poderiam ser dinamicamente redirecionados por rotas alternativas, mantendo a comunicação ativa.
- **Eficiência:** Os canais de comunicação eram compartilhados. Pacotes de múltiplas conversas podiam trafegar pela mesma linha, maximizando o uso da largura de banda.

## O Nascimento da Rede e a Primeira Mensagem

A rede física começou a ser construída em 1969. Os primeiros "roteadores", chamados **IMPs (Interface Message Processors)** — minicomputadores robustos da Honeywell — foram instalados em quatro locais:

1. UCLA (Universidade da Califórnia, Los Angeles)
2. Stanford Research Institute (SRI)
3. UC Santa Barbara (UCSB)
4. Universidade de Utah

Em **29 de outubro de 1969**, ocorreu a primeira transmissão. O programador Charley Kline, da UCLA, tentou fazer um login remoto em um computador no SRI. A intenção era digitar **LOGIN**, mas o sistema travou após as duas primeiras letras. Assim, a primeira mensagem transmitida pela ARPANET foi, acidentalmente, " **LO** ".

## A Evolução para TCP/IP

O protocolo inicial da ARPANET era o NCP (Network Control Program). No entanto, o NCP foi projetado para uma única rede e não previa a interconexão de múltiplas redes heterogêneas (uma "rede de redes").

Para resolver esse problema, Vint Cerf e Bob Kahn propuseram, em 1974, uma nova arquitetura que dividia as responsabilidades:

- **IP (Internet Protocol):** Um protocolo simples na Camada 3, responsável apenas pelo endereçamento e roteamento de pacotes entre redes distintas.

- **TCP (Transmission Control Protocol):** Um protocolo robusto na Camada 4, responsável por garantir a entrega confiável dos dados, controlando o fluxo, a ordem e os erros.

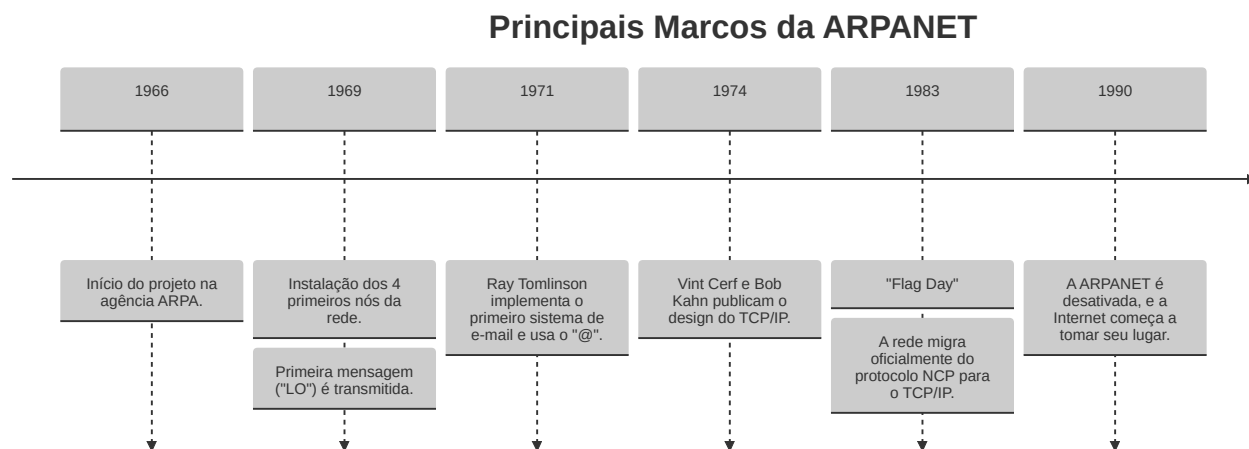
Essa nova suíte de protocolos, o TCP/IP, tornou-se o padrão. A transição oficial de toda a rede do NCP para o TCP/IP ocorreu em **1º de janeiro de 1983**, um evento conhecido como "Flag Day", considerado por muitos o verdadeiro nascimento da Internet moderna.

## Legado e a "Killer App"

Embora o login remoto (Telnet) e a transferência de arquivos (FTP) fossem aplicações importantes, o **e-mail**, desenvolvido por Ray Tomlinson em 1971, rapidamente se tornou a "killer app" da ARPANET. Ele demonstrou o imenso potencial da rede para a comunicação humana e colaboração científica.

A ARPANET foi formalmente desativada em 1990, pois sua função foi absorvida por redes mais novas como a NSFNET, que por sua vez deu origem à Internet comercial que conhecemos hoje. O legado da ARPANET é a prova de que a comutação de pacotes e o TCP/IP eram a base sólida necessária para construir uma rede global.

## Linha do Tempo



# Conclusão

Neste trabalho, realizei uma investigação sobre os conceitos fundamentais que estruturam as redes de computadores, com o objetivo de solidificar meu entendimento sobre o tema para as atividades do laboratório.

Iniciei o estudo analisando os três **tipos de entrega de mensagens**: Unicast, Broadcast e Multicast. A compreensão de suas diferenças foi crucial para entender como a eficiência e o direcionamento da comunicação são gerenciados em diferentes cenários de rede.

A análise comparativa entre os protocolos **TCP e UDP** me permitiu discernir as situações ideais para cada um. Concluí que a confiabilidade do TCP é indispensável para tarefas como o download de arquivos, enquanto a agilidade do UDP é vital para aplicações sensíveis à latência, como jogos online.

Posteriormente, classifiquei um conjunto de protocolos essenciais (como FTP, DNS, IP, entre outros) de acordo com suas camadas no **Modelo OSI**. Essa tarefa me ajudou a visualizar a arquitetura de rede e a função específica que cada protocolo desempenha no processo de comunicação.

Finalmente, a pesquisa sobre a **ARPANET** me proporcionou uma perspectiva histórica sobre a origem da Internet. Ficou claro que inovações como a comutação de pacotes e o desenvolvimento do TCP/IP não foram apenas marcos tecnológicos, mas a verdadeira fundação sobre a qual a conectividade global foi construída.

A conclusão deste estudo representa um passo importante na minha formação, fornecendo a base teórica necessária para abordar os desafios práticos propostos pelo laboratório.

# Referências

- [https://www.uomustansiriyah.edu.iq/media/lectures/5/5\\_2017\\_02\\_12!05\\_29\\_03\\_PM.pdf](https://www.uomustansiriyah.edu.iq/media/lectures/5/5_2017_02_12!05_29_03_PM.pdf)  
([https://www.uomustansiriyah.edu.iq/media/lectures/5/5\\_2017\\_02\\_12!05\\_29\\_03\\_PM.pdf](https://www.uomustansiriyah.edu.iq/media/lectures/5/5_2017_02_12!05_29_03_PM.pdf))
- <https://www.metered.ca/blog/tcp-vs-udp-protocol/>  
(<https://www.metered.ca/blog/tcp-vs-udp-protocol/>)
- <https://web.stanford.edu/class/cs244a/handouts/H6%20Transport%202008.pdf>  
(<https://web.stanford.edu/class/cs244a/handouts/H6%20Transport%202008.pdf>)
- <https://www.ime.usp.br/~is/abc/abc/node20.html>  
(<https://www.ime.usp.br/~is/abc/abc/node20.html>)
- <https://olhardigital.com.br/2019/10/24/internet-e-redes-sociais/mae-da-internet-conheca-a-historia-da-arpanet/>  
(<https://olhardigital.com.br/2019/10/24/internet-e-redes-sociais/mae-da-internet-conheca-a-historia-da-arpanet/>)
- <https://aws.amazon.com/pt/what-is/osi-model/> (<https://aws.amazon.com/pt/what-is/osi-model/>)
- <https://www.freecodecamp.org/portuguese/news/o-que-e-o-modelo-osi-redes-de-computadores-para-iniciantes/>  
(<https://www.freecodecamp.org/portuguese/news/o-que-e-o-modelo-osi-redes-de-computadores-para-iniciantes/>)
- <https://www.geeksforgeeks.org/computer-networks/open-systems-interconnection-model-osi/>  
(<https://www.geeksforgeeks.org/computer-networks/open-systems-interconnection-model-osi/>)
- <https://www.geeksforgeeks.org/computer-networks/differences-between-tcp-and-udp/>  
(<https://www.geeksforgeeks.org/computer-networks/differences-between-tcp-and-udp/>)