

## **Análise Comparativa dos Protocolos de Transporte TCP e UDP em Redes de Computadores**

Jeferson Barros Feliciano  
Leonardo Leite Cruz

## RESUMO

Este artigo explora os principais protocolos da camada de transporte do modelo OSI: o Protocolo de Controle de Transmissão (TCP) e o Protocolo de Datagrama do Usuário (UDP). Ambos desempenham papéis fundamentais na comunicação entre sistemas finais em redes de computadores, mas possuem abordagens distintas. O TCP é um protocolo orientado à conexão, que garante a entrega confiável e ordenada de pacotes, sendo ideal para aplicações que requerem precisão, como navegação web e transferência de arquivos. Por outro lado, o UDP é um protocolo sem conexão, que oferece menor sobrecarga e é amplamente utilizado em aplicações que exigem baixa latência, como streaming de vídeo e jogos online. O estudo apresenta uma revisão teórica desses protocolos, suas vantagens, desvantagens e uma análise comparativa de sua eficiência em diferentes contextos de aplicação. Conclui-se que a escolha entre TCP e UDP depende das necessidades específicas de cada aplicação, seja pela confiabilidade ou pela velocidade de transmissão dos dados.

**Palavra-chave:** Camada de Transporte, TCP, UDP, Protocolos.

## INTRODUÇÃO

A camada de transporte faz parte do modelo de Interconexão de Sistemas Abertos (OSI) e é responsável por garantir a entrega de dados entre sistemas finais em uma rede. Dois dos protocolos mais utilizados nesta camada são o Protocolo de Controle de Transmissão (TCP) e o Protocolo de Datagrama do Usuário (UDP). Ambos têm funções distintas e desempenham papéis fundamentais em diferentes tipos de aplicações de rede.

O TCP é conhecido por ser um protocolo orientado à conexão, que garante a entrega confiável de pacotes. Ele é amplamente utilizado em aplicações como navegação na web, e-mails e transferências de arquivos. Já o UDP é um protocolo sem conexão, que oferece menor sobrecarga, sendo ideal para aplicações em tempo real como streaming de vídeo e jogos online.

Este estudo tem como objetivo contextualizar os protocolos e delinear suas principais vantagens e desvantagens, preparando o terreno para uma análise comparativa de eficiência e desempenho.

## REVISÃO DA LITERATURA

### O que é a camada de transporte?

A camada de transporte é a quarta camada do Modelo OSI (acrônimo do inglês *Open System Interconnection*) que é antecedida pelas: camada física, camada de enlace de dados e camada de rede. O modelo OSI é uma referência para redes de computadores e divide o processo em camadas distintas para tornar a visualização das redes mais compreensível.

Para Kurose (2013) o protocolo da camada de transporte fornece comunicação lógica entre processos de aplicação que rodam em hospedeiros diferentes, isso quer dizer que, uma mensagem é enviada de um ponto até o outro transmitindo um dado ou informação, independente da aplicação usada e do tipo, topologia ou configuração das redes físicas existentes entre elas.

A mensagem é fragmentada em pequenos pedaços de dados menores e adiciona-se um cabeçalho de camada de transporte a cada pedaço para criar o segmento de camada de transporte. Essa camada, então, passa o segmento para a

de rede no sistema final remetente, onde ele é encapsulado em um pacote de camada de rede enviado ao destinatário (KUROSE , 2013).

A OSI define o protocolo de transporte para operar em dois modos: Orientado à conexão e Não-Orientado à conexão, sendo assim a comunicação lógica nesse contexto significa que, do ponto de vista de uma aplicação, tudo se passa como se os hospedeiros que rodam os processos estivessem conectados diretamente, e dois dos principais protocolos desta camada são o TCP e UDP.

Os dois protocolos citados acima são regidos pelo RFC (sigla em inglês para *Request for Comments*, que em português significa "pedido de comentários"). Trata-se de documentos técnicos que descrevem padrões, protocolos, procedimentos e pesquisas relacionados à Internet.

Existe também algo muito importante para ser citado referente à camada de transporte que é a multiplexação e a demultiplexação e ambos são utilizados para os protocolos TCP e UDP. De acordo com Kurose e Ross (2013) o trabalho de reunir dá origem às partes de dados, encapsular cada parte de dados com informações de cabeçalho para criar segmentos, e passar esses segmentos para a camada de rede é denominada **multiplexação**. Já a tarefa de entregar os dados contidos nesse segmento da camada de transporte ao destino correto é denominada **demultiplexação**.

## TCP

O TCP (Transmission Control Protocol) é um protocolo de comunicação orientado à conexão, o que significa que ele estabelece uma conexão entre o emissor e o receptor antes de começar a transmissão de dados. Conforme definido na RFC 793, o TCP é projetado para ser um protocolo confiável de ponta a ponta, garantindo a entrega segura e ordenada de dados, projetado para encaixar em uma hierarquia em camadas de protocolos que oferecem suporte a uma variedade de redes e aplicações.

Ele trabalha, primeiramente, estabelecendo uma comunicação entre o remetente e destinatário, através de um processo denominado **apresentação de três vias** (*three-way handshake*), em que ambos os dispositivos trocam mensagens para estabelecer uma conexão segura e confiável. Este processo envolve o emissor envia um pacote com o sinalizador SYN (*Synchronize*) para iniciar a conexão, o receptor responde com um pacote SYN-ACK, sinalizando que está pronto para a

conexão e o emissor responde com um pacote ACK (*Acknowledgment*), confirmando a conexão estabelecida. Somente após esse procedimento a troca de dados pode começar.

Os dados são divididos em segmentos, que incluem tanto o payload quanto as informações de controle necessárias para a entrega confiável. Cada segmento contém campos como número de sequência, número de confirmação (ACK) e flags de controle, como SYN, ACK, FIN (finalizar), entre outras, que gerenciam o estabelecimento, manutenção e encerramento da conexão.

O TCP garante que os dados sejam entregues ao destino na mesma ordem em que foram enviados, utilizando números de sequência em seus segmentos. O receptor usa esses números para montar os dados na ordem correta e solicitar a retransmissão de pacotes perdidos ou corrompidos. Isso é feito por meio de um ACK que confirma o recebimento de cada segmento.

Além disso, o TCP monitora a rede para detectar e evitar congestionamento, usando um mecanismo de controle de fluxo para evitar que o emissor envie mais dados do que o receptor pode processar. Esse controle é feito com base na janela deslizante (*sliding window*), que ajusta dinamicamente o tamanho dos dados que podem ser transmitidos, garantindo que o receptor não seja sobrecarregado. Adicionalmente, é possível a transferência simultânea em ambas direções (emissor-receptor) durante toda a sessão, bem como a multiplexação de várias conexões simultâneas devido ao uso de portas de origem e destino.

Quando a comunicação chega ao fim, o TCP finaliza a conexão utilizando um processo chamado de *"four-way handshake"*, onde o emissor envia um pacote FIN, e o receptor responde com um ACK, e em seguida, também envia um FIN. O emissor então responde com um último ACK, garantindo que ambos os lados concordam com o encerramento.

É complementado pelo protocolo da Internet (IP), normalmente chamado de TCP/IP, sendo utilizado em diversas aplicações da Internet, como o HTTP/HTTPS (para navegação na web), FTP (transferência de arquivos), SMTP/POP3/IMAP (e-mails), entre outros.

Essas características tornam o TCP ideal para aplicações que requerem confiabilidade, controle de fluxo e entrega ordenada de dados, como transferência de arquivos, streaming de mídia e comunicação entre servidores. Por outro lado, devido à sua sobrecarga, ele pode não ser ideal para aplicações que exigem baixa

latência, como jogos online ou streaming em tempo real, onde o protocolo UDP é mais utilizado.

## **UDP**

O UDP (*User Datagram Protocol*) é um protocolo não-orientado à conexão, o que significa que ele não garante a entrega dos pacotes nem a ordem correta de chegada. Ele é descrito na RFC 768 e permite que a aplicação envie um datagrama encapsulado num pacote, que oferece à aplicação solicitante um serviço não confiável, mas com a possibilidade de grandes volumes de dados.

Este protocolo fornece um procedimento para que programas de aplicação enviem mensagens para outros programas com um mínimo de mecanismo de protocolo. O protocolo é orientado a transações e a entrega e proteção contra duplicação não são garantidos.

Basicamente o UDP funciona da seguinte forma: pega-se as mensagens do processo da aplicação, anexa os campos de número da porta de origem e de destino e então adiciona dois outros pequenos campos e é enviado para camada de rede, que é encapsulada e enviada para o receptor, sendo que não há garantias que ela chegará, pois não foi feita uma conexão antes para determinar se a conexão estava ativa (KUROSE e ROSS, 2013).

O protocolo até fornece uma verificação de erro, porém ele faz para recuperar-se de um erro, algumas implementações do UDP apenas descartam o segmento danificado, outras passam o segmento errado à aplicação acompanhado de um aviso.

## **METODOLOGIA**

A metodologia deste estudo foi baseada em uma pesquisa bibliográfica realizada em diversas plataformas online, com ênfase na análise de artigos acadêmicos, materiais educacionais e livros.

Foram consultados websites de reconhecida relevância no campo, além de vídeo-aulas, que forneceram uma compreensão mais prática sobre o tema. A seleção das fontes seguiu critérios de atualidade, relevância e credibilidade, priorizando conteúdos que contribuíram de maneira significativa para a construção do referencial teórico e análise crítica do objeto de estudo.

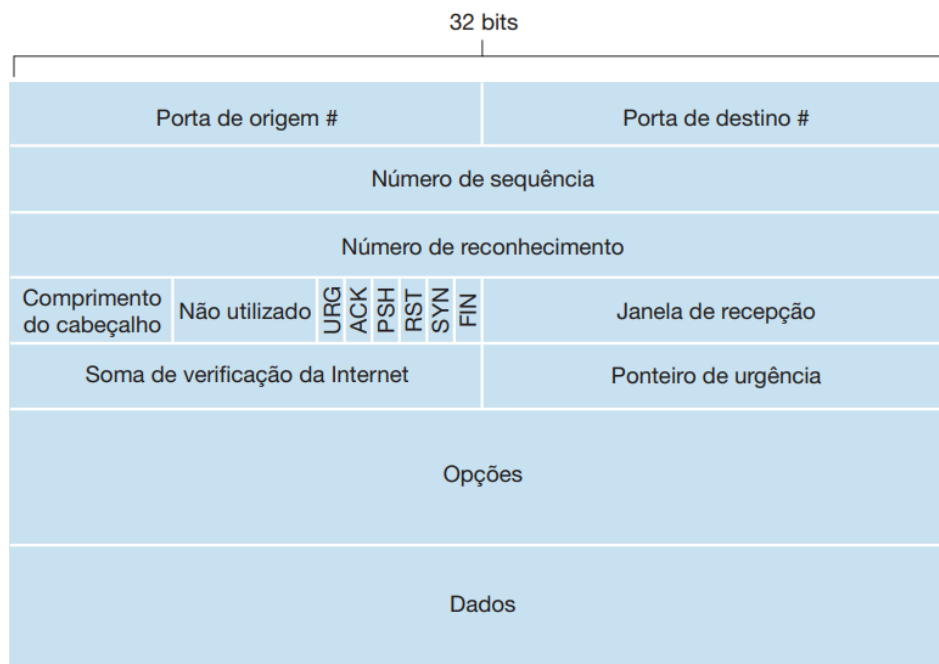
## DESENVOLVIMENTO

### Comparação entre as camadas de transportes TCP e UDP

Nesta seção, examinamos os princípios de funcionamento dos protocolos TCP e UDP, destacando suas principais diferenças. O TCP é caracterizado por seu controle de congestionamento e mecanismos de confiabilidade, que incluem a retransmissão de pacotes e o estabelecimento de conexão, enquanto o UDP adota uma abordagem mais simples e leve, priorizando a velocidade de transmissão sobre a confiabilidade.

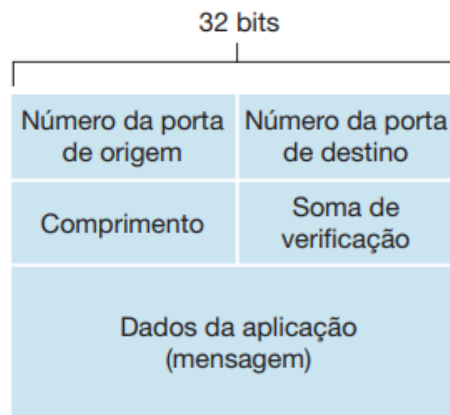
Ao comparar os formatos de cabeçalhos do TCP e do UDP, notamos que o cabeçalho do TCP é significativamente mais complexo. Essa complexidade resulta em uma maior segurança quanto à entrega dos dados, garantindo que as informações cheguem ao receptor sem erros. Em contraste, o UDP não possui os mesmos requisitos de segurança, uma vez que seu objetivo principal é a transmissão rápida de dados. Essa diferença é ilustrada nas Figuras 1 e 2.

**Figura 1: Formato de Cabeçalho TCP**



Fonte: KUROSE e ROSS, 2013

**Figura 2: Formato de Cabeçalho UDP**



Fonte: KUROSE e ROSS, 2013

O título TCP é estruturado em vários campos com funções específicas para garantir uma comunicação eficiente entre dispositivos. Entre eles, a porta de origem e a porta de destino que identificam os aplicativos de envio e recepção, respectivamente. O número de sequência permite a correta proteção da mensagem ao marcar a posição do primeiro byte de dados, enquanto o número de confirmação garante que o emissor saiba qual byte o receptor espera a seguir. Outros campos incluem o tamanho do cabeçalho, um campo reservado, flags de controle (como URG, ACK, PSH, RST, SYN e FIN), e a janela de recepção, que determina quantos bytes o receptor pode aceitar antes de enviar uma confirmação.

Além disso, o cabeçalho contém a soma de verificação que verifica a integridade dos dados, um ponteiro de urgência que prioriza certos dados, e um campo de opções que permite a inclusão de funcionalidades adicionais, como escalonamento de janela e carimbos de data/hora. Finalmente, o campo de dados armazena a informação transmitida, cujo tamanho varia de acordo com o conteúdo dos segmentos, completando assim a estrutura necessária para o protocolo TCP.

Já o cabeçalho UDP é mais simples em comparação ao TCP, contendo apenas os campos essenciais, como porta de origem e destino, seu comprimento, a soma de verificação e os dados para ser transmitido. Essa estrutura reduzida torna o UDP significativamente mais leve e rápido em comparação ao TCP. Essa leveza é uma das razões pelas quais o UDP é amplamente utilizado em aplicações de latência



crítica, como streaming de vídeo e jogos online. Assim como conclui Kurose E Ross (2013), o protocolo TCP assim como o do UDP, inclui números de porta de origem e de destino, que são usados para multiplexação e demultiplexação de dados de/para aplicações de camadas superiores.

**Figura 3: Aplicações populares da Internet e seus protocolos de transporte**

Aplicação	Protocolo da camada de aplicação	Protocolo de transporte subjacente
Correio eletrônico	SMTP	TCP
Acesso a terminal remoto	Telnet	TCP
Web	HTTP	TCP
Transferência de arquivo	FTP	TCP
Servidor de arquivo remoto	NFS	Tipicamente UDP
Recepção de multimídia	Tipicamente proprietário	UDP ou TCP
Telefonia por Internet	Tipicamente proprietário	UDP ou TCP
Gerenciamento de rede	SNMP	Tipicamente UDP
Protocolo de roteamento	RIP	Tipicamente UDP
Tradução de nome	DNS	Tipicamente UDP

Fonte: KUROSE e ROSS, 2013

Na **Figura 3**, são apresentadas as principais aplicações da Internet e seus protocolos de transporte associados. Por exemplo, o e-mail utiliza o protocolo TCP, pois é fundamental que cada informação seja entregue corretamente, garantindo maiores precisão nos dados. Por outro lado, em aplicações de multimídia, como streaming, o UDP é frequentemente preferido, pois a continuidade da transmissão é mais crítica do que a entrega individual de cada pacote. Isso demonstra que cada protocolo tem sua funcionalidade específica, adaptando-se às necessidades das aplicações.

### **Vantagens e desvantagens**

Pela diversidade de abordagens, os dois protocolos são essenciais para aplicação de diversos tipos, desde navegação na web até streaming de jogos transmitidos ao vivo, sempre observando qual melhor se integra com as necessidades da aplicação, que ou pode trazer mais desempenho ou até segurança na transmissão desses dados. Ao fazer uma comparação entre os protocolos TCP e UDP, foi feito um levantamento das suas principais vantagens e desvantagens.

Principais Vantagens de Usar o Protocolo TCP:

- Integridade de Dados
- Recuperação de Pacotes Perdidos
- Conexão Ponto a Ponto
- Confiabilidade
- Controle de Fluxo
- Controle de Congestionamento
- Comunicação Full Duplex
- Entrega ordenada de dados
- Multiplexação

Desvantagens de usar o protocolo TCP:

- Alta Latência
- Lentidão em Redes Congestionadas
- Complexidade de Implementação
- Suscetibilidade ao Sequestro de Conexão

Principais Vantagens de Usar o Protocolo UDP:

- Baixa Latência
- Menos Sobrecarga de Cabeçalho
- Simples e Eficiente
- Melhor controle no nível da aplicação sobre o que é enviado e quando
- Não Exige Estabelecimento de Conexão
- Não Mantém Estado de Conexão
- Pequena Sobrecarga de Cabeçalho de Pacote

Desvantagens de usar o protocolo UDP:

- Baixa Confiabilidade
- Sem Garantia de Entrega
- Falta de Controle de Erros
- Sem Controle de Congestionamento
- Não Sequencia Dados
- Dependente da Camada de Aplicação para Controle

## **CONCLUSÃO**

Podemos perceber que, a escolha entre TCP e UDP fica dependente das necessidade que determinada aplicação possui, caso precise de mais confiabilidade e segurança que as informações devem chegar até o receptor a escolha melhor é o TCP, agora se precisa que chegue às informações, mesmo que falte algo e isso não prejudique a aplicação no final o melhor a ser utilizado é o protocolo UDP, além do mais por destacar a velocidade que consegue transmitir os dados.

Sendo assim, o crucial é entender o funcionamento de cada protocolo, pois permite ao desenvolvedor escolher a melhor opção no momento, fazendo com que otimize-se mais tempo e recursos e trazendo uma melhor experiência aos usuários e uma melhor performance no geral. A coexistência desses protocolos na arquitetura da internet reflete a diversidade de aplicações e demandas que caracterizam o mundo digital contemporâneo.

Além disso, fica então como sugestão para pesquisas futuras explorar as inovações e aprimoramentos nas tecnologias de transmissão de dados que envolvem TCP e UDP.

## REFERÊNCIAS

ARAÚJO, W. S. **Estudo Comparativo da Utilização de Protocolos da Camada de Transporte e Aplicação em Ambientes de Mobile Cloud Computing**. Universidade Federal do Ceará, Fortaleza, 2022. Disponível em: [https://repositorio.ufc.br/bitstream/riufc/70974/1/2022\\_tcc\\_wsaraujo.pdf](https://repositorio.ufc.br/bitstream/riufc/70974/1/2022_tcc_wsaraujo.pdf). Acesso em: 18 set. 2024.

KRETCHOU, P., **Protocolos TCP e UDP**. YouTube, 1 de abr. de 2013. 18min10s. Disponível em: [https://www.youtube.com/watch?v=uRvjPlbJ\\_98](https://www.youtube.com/watch?v=uRvjPlbJ_98). Acesso em: 21 set. 2024.

KUROSE, J. F.; ROSS, K. W. **Redes de computadores e a internet: uma abordagem top-down**. 6. ed. São Paulo: Pearson, p. 135-189. 2013.

ORESTES, Y., **O que é UDP e TCP? Entenda quais as diferenças e como funciona cada Protocolo**, DevOps, ALURA, Setembro de 2023. Disponível em: <https://www.alura.com.br/artigos/quais-as-diferencas-entre-o-tcp-e-o-udp>. Acesso em: 18 de set. 2024.

POSTEL, J., **Transmission Control protocol**, RFC 793, USC/Information, Sciences Institute, Setembro de 1981. Disponível em: <https://datatracker.ietf.org/doc/html/rfc793>. Acesso em: 21 de set. 2024.

\_\_\_\_\_, **User Datagram Protocol**, RFC 768, USC/Information, Sciences Institute, Agosto de 1980. Disponível em: <https://datatracker.ietf.org/doc/html/rfc768>. Acesso em: 21 de set. 2024.