Estácio

Protocolos de Redes

Prof.: Boanerges Teixeira

Protocolos de Transporte

Divisão em Camadas

Imagina que você quer enviar um presente para um amigo que mora em outra cidade. Se você simplesmente jogasse o presente num caminhão sem organizar nada, poderia dar ruim no caminho: talvez o presente se perca, se quebre ou nem chegue ao destino. Para evitar isso, usamos um sistema organizado de **camadas** no transporte, mais ou menos assim:

1. **Empacotamento** (Aplicação) → Você coloca o presente numa caixa e escreve um bilhete.

2. **Endereçamento** (Transporte) → Você escreve o nome e o endereço do amigo no pacote.

3. **Roteamento** (Rede) → A transportadora escolhe a melhor rota para entregar.

4. **Entrega Física** (Enlace/Física) → O caminhão, moto ou avião leva o pacote até o destino.

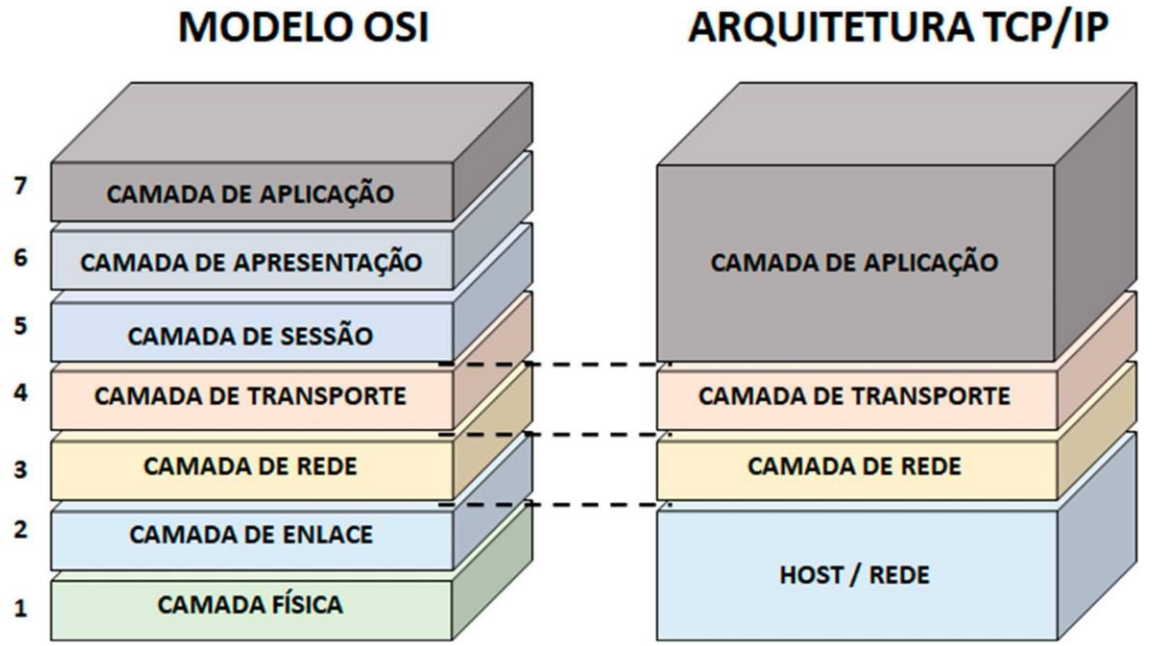
Agora, trazendo isso para redes de computadores:

- Os **protocolos de transporte** (como TCP e UDP) garantem que os dados cheguem corretamente ao destinatário, como uma transportadora confiável.
- Dividir em camadas permite que cada parte do processo funcione sem precisar saber detalhes das outras. Assim, o programador só precisa se preocupar em enviar dados (como um e-mail ou um vídeo) e os protocolos se viram para entregar. Dá para pensar nas camadas como uma linha de produção bem organizada: cada setor faz sua parte e, no final, o produto chega ao cliente direitinho.

Protocolos de Transporte

Divisão em Camadas

Os protocolos de transporte são divididos em camadas para **modularizar a comunicação** e facilitar a interoperabilidade entre diferentes sistemas e redes. Essa abordagem segue o **modelo OSI (Open Systems Interconnection)** e o **modelo TCP/IP**, permitindo que cada camada tenha funções específicas, isolando responsabilidades e melhorando a eficiência da rede.



Protocolos de Transporte

Divisão em Camadas

Principais razões para dividir em camadas:

1.Abstração e Independência: Cada camada funciona de forma independente, ou seja, um desenvolvedor pode criar um novo protocolo de transporte sem modificar os protocolos de rede ou de aplicação.

2.Facilidade de Manutenção e Evolução: Mudanças em uma camada não afetam diretamente as outras, tornando o sistema mais modular e flexível.

3.Interoperabilidade: Permite que dispositivos e sistemas operacionais diferentes se comuniquem, desde que sigam os mesmos padrões.

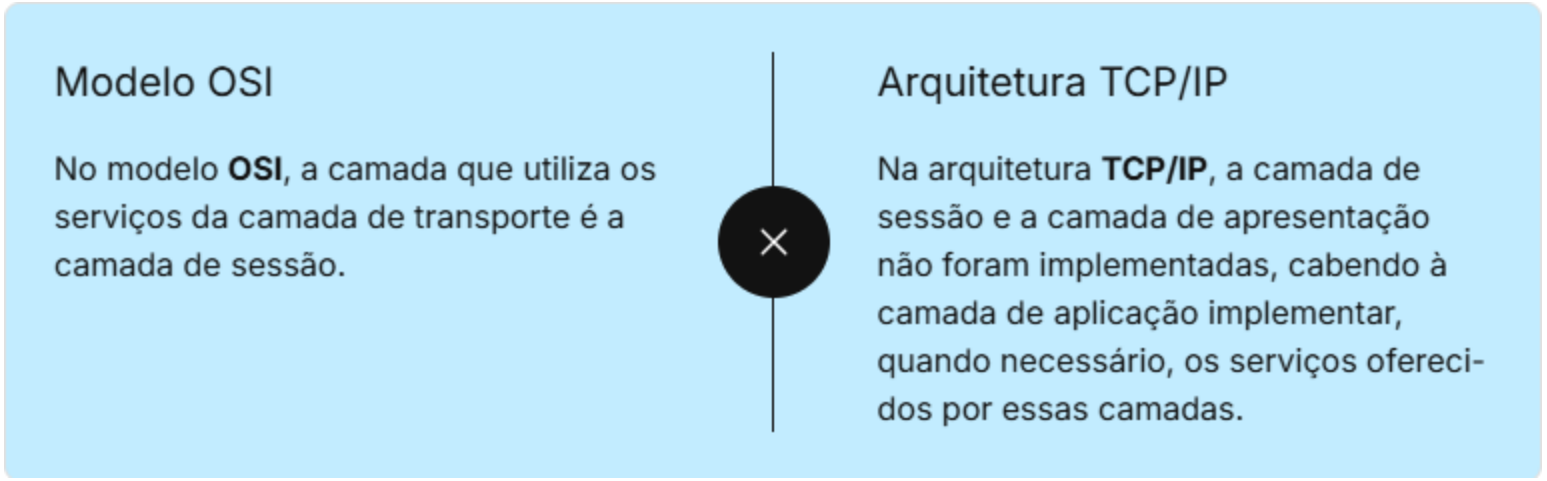
4.Reutilização de Funcionalidades: Camadas superiores podem usar serviços das camadas inferiores sem precisar implementar tudo do zero.

Protocolos de Transporte

Divisão em Camadas

Interação com Outras Camadas

1.Aplicação (HTTP, FTP, etc.) → Envia dados brutos para a camada de transporte.
2.Transporte (TCP/UDP) → Divide os dados em segmentos e adiciona informações de controle.
3.Rede (IP) → Encapsula os segmentos em pacotes e define roteamento.
4.Enlace (Ethernet, Wi-Fi) → Transmite os pacotes fisicamente pelo meio de comunicação.



Protocolos de Transporte

Divisão em Camadas

Multiplexação e demultiplexação

A **multiplexação** e a **demultiplexação** fornecem um serviço de entrega processo a processo para aplicações que são executadas nos hospedeiros.

Hospedeiro de destino

No **hospedeiro de destino**, a camada de transporte recebe segmentos de dados da camada de rede e tem a responsabilidade de entregar esses dados ao processo de aplicação correto.

Hospedeiro remoto

Cada camada do modelo de rede denomina os dados trocados com o **hospedeiro remoto** de uma forma diferente das demais camadas.

Protocolos de Transporte

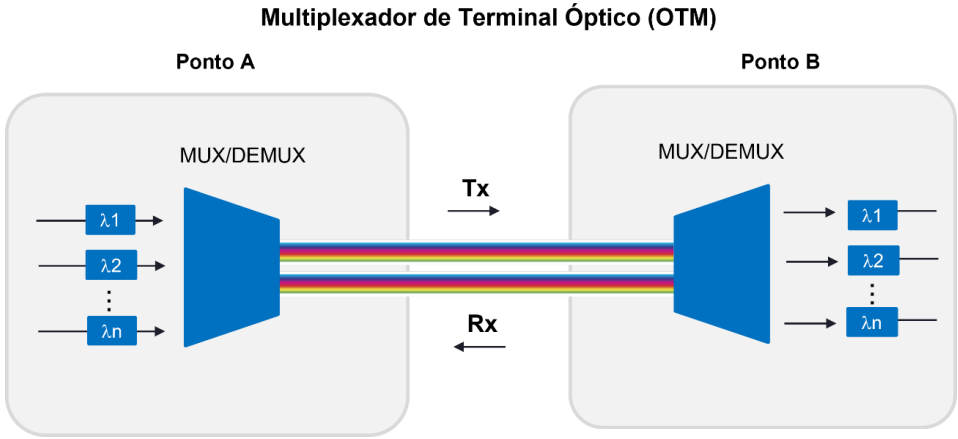
Divisão em Camadas

A tarefa de entregar os dados contidos em um segmento à porta correta é denominada **demultiplexação**.

A tarefa de reunir, no hospedeiro de origem, porções de dados provenientes de diferentes portas, encapsular cada porção de dados com informações de cabeçalho (que, mais tarde, serão usadas na demultiplexação) para criar segmentos, e passar esses segmentos para a camada de rede é denominada **multiplexação**.

Exemplo

Eduardo está navegando na web, acessando seu e-mail e fazendo download de arquivos utilizando um programa específico para isso. Todos os programas utilizados por Eduardo (navegador web, cliente de e-mail e programa de transferência de arquivos) utilizam o TCP, que fará a transferência da informação até o destino. Então, a multiplexação está permitindo que vários programas possam utilizar o TCP ao mesmo tempo, fazendo, assim, com que Eduardo possa ter diversos programas acessando a rede.



Protocolos de Transporte

Divisão em Camadas

O ato de colocar o segmento da camada de transporte dentro de um pacote da camada de rede é denominado encapsulamento.

Em condições normais o pacote da camada de rede também é encapsulado em um **quadro** da camada de enlace de dados, sendo o quadro o nome dado ao conjunto de dados + cabeçalho intercambiado entre entidades da camada de enlace de dados.

Protocolos de Transporte

Antes de Começar...

** O **checksum** é um mecanismo de verificação de integridade usado para detectar erros em dados transmitidos ou armazenados. Ele funciona gerando um valor numérico (hash) a partir dos dados originais e comparando esse valor no destino para garantir que os dados não foram corrompidos.*

Como funciona o checksum?

1.Cálculo no emissor → Antes de enviar os dados, o remetente realiza uma operação matemática sobre o conteúdo (como soma binária, XOR, ou algoritmos mais complexos) e gera um valor de checksum.

2.Transmissão → O dado é enviado junto com o checksum calculado.

3.Verificação no receptor → O destinatário recalcula o checksum a partir dos dados recebidos.

4.Comparação → Se o checksum calculado no destino for igual ao enviado, os dados provavelmente estão intactos. Se forem diferentes, houve erro na transmissão.

No **UDP**, o checksum é **opcional** e protege apenas os cabeçalhos e dados contra corrupção.

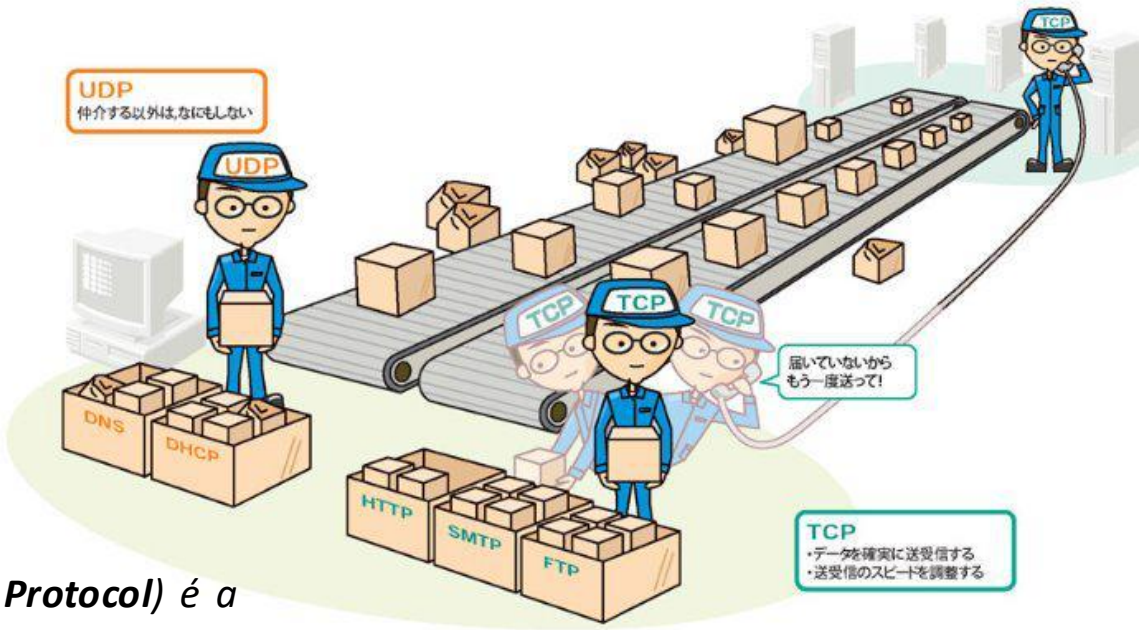
No **TCP**, o checksum é **obrigatório** e verifica a integridade da transmissão fim a fim.

Protocolos de Transporte

UDP (User Datagram Protocol)

2. UDP (User Datagram Protocol)

- **Não confiável**: Não garante ordem ou entrega de pacotes.
- **Menos overhead**: Menos cabeçalhos, transmissão mais rápida.
- **Sem conexão**: Os pacotes são enviados sem estabelecer uma conexão formal.
- **Uso**: Streaming de vídeo, VoIP, DNS, jogos online.



* *Um **datagrama** no contexto do UDP (**User Datagram Protocol**) é a unidade básica de transmissão de dados. Ele é um pacote **independente e autossuficiente**, contendo todas as informações necessárias para ser entregue ao destino sem depender de conexões pré-estabelecidas ou verificações de entrega.*

Protocolos de Transporte

UDP (User Datagram Protocol)

Estrutura de um Datagrama UDP

Cada datagrama contém:

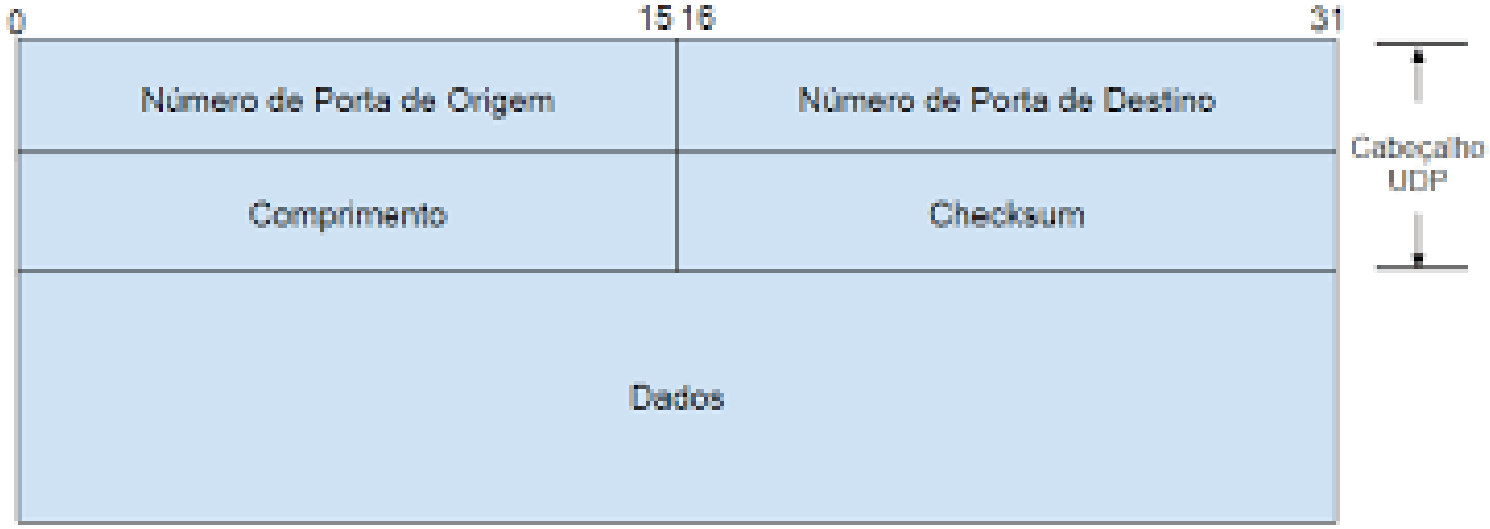
1.Porta de origem (16 bits) → Indica o processo de envio.

2.Porta de destino (16 bits) → Indica o processo receptor.

3.Comprimento (16 bits) → Tamanho total do datagrama (cabeçalho + dados).

4.Checksum (16 bits) → Verificação de integridade opcional.

5.Dados → A carga útil transmitida.

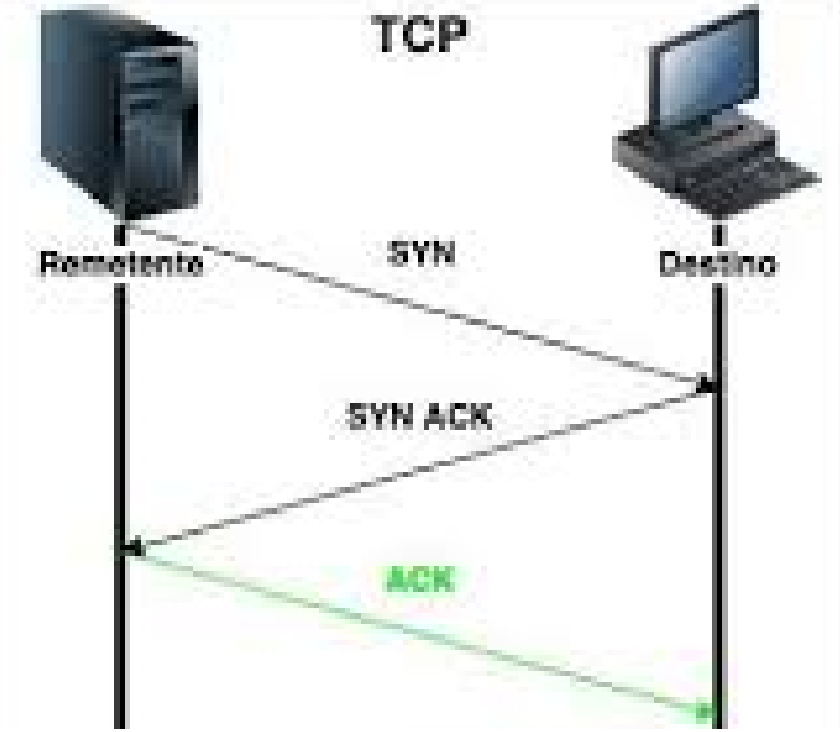


Protocolos de Transporte

TCP (Transmission Control Protocol)

TCP (Transmission Control Protocol)

•**Confiável**: Garante que os dados cheguem completos e na ordem correta.
•**Controle de fluxo**: Ajusta a taxa de envio de acordo com a capacidade do receptor.
•**Controle de congestionamento**: Evita sobrecarga na rede.
•**Orientado a conexão**: Estabelece uma conexão antes da transmissão (handshake de três vias).
•**Uso**: HTTP, HTTPS, FTP, SMTP, SSH.



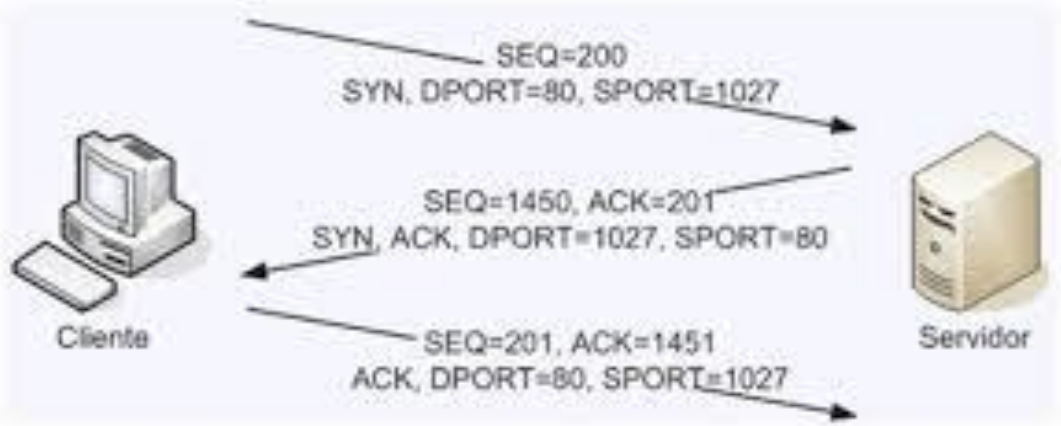
Protocolos de Transporte

TCP (Transmission Control Protocol)

Definição de Flag

No contexto de redes e protocolos, uma **flag** é um campo dentro do cabeçalho de um pacote que sinaliza uma instrução ou estado específico. Flags são **bits individuais** que podem estar ativados (**1**) ou desativados (**0**), funcionando como indicadores de controle para a comunicação entre dispositivos.

O protocolo **TCP (Transmission Control Protocol)** usa diferentes flags para controlar a comunicação entre dois dispositivos na rede. Três das mais importantes são **SYN, ACK e FIN**, que ajudam a estabelecer, manter e encerrar conexões.



Cliente: Servidor, estás aí?? (SYN)
Servidor: Sim estou... (ACK) e tu, está aí? (SYN)
Cliente: Sim, estou... (ACK)

Protocolos de Transporte

TCP (Transmission Control Protocol)

1. SYN (Synchronize)

- **Função**: Inicia uma conexão TCP.
- **Descrição**: Quando um dispositivo quer se conectar a outro, ele envia um pacote **SYN** para indicar que deseja estabelecer uma comunicação.
- **Uso**: Primeira etapa do **"Three-Way Handshake"**.

Exemplo: O cliente envia um **SYN** para o servidor (indicando intenção de conexão).

2. ACK (Acknowledgment)

- **Função**: Confirma o recebimento de pacotes.
- **Descrição**: O **ACK** é usado para dizer "Recebi seu pacote com sucesso". No TCP, cada pacote enviado precisa ser reconhecido pelo destinatário.
- Durante a transmissão de dados: Cada pacote recebido corretamente é confirmado com um **ACK**.
- **Uso**: Acontece em **várias fases** da comunicação TCP.

Exemplo: Durante o handshake: O servidor responde com **SYN + ACK**, confirmando o recebimento do SYN e também enviando sua própria solicitação.

3. FIN (Finish)

- **Função**: Finaliza uma conexão TCP.
- **Descrição**: Quando um dispositivo deseja encerrar a comunicação, ele envia um pacote **FIN** para informar que não tem mais dados para transmitir.
- **Uso**: Fase de encerramento da conexão TCP.

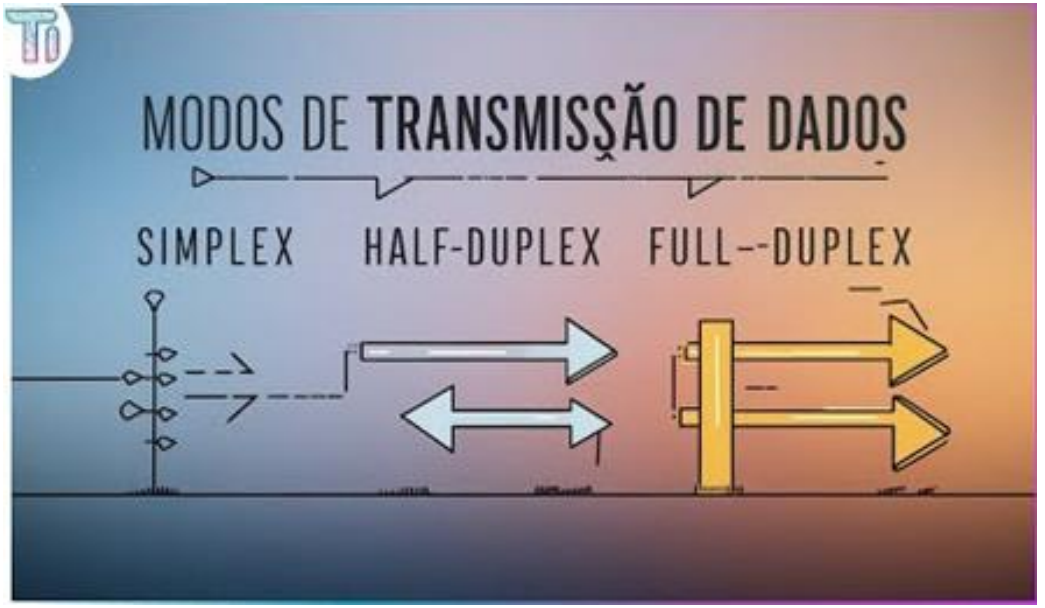
Exemplo: O cliente ou servidor envia um **FIN** para encerrar a conexão -> O outro lado responde com um **ACK** e, se necessário, também envia um **FIN**.
-> Por fim, o primeiro dispositivo responde com um último **ACK**, encerrando a sessão.

Protocolos de Transporte

TCP - Controle de fluxo

Os hospedeiros que fazem parte de uma conexão precisam reservar *buffers* tanto para a transmissão quanto para a recepção de dados. Quando os dados chegam, a entidade TCP os recebe, os coloca em um *buffer* e avisa ao processo da aplicação receptora que dados estão disponíveis para serem retirados de lá. Porém, o processo não necessita retirar esses dados do *buffer* imediatamente. O momento em que esses dados serão lidos do *buffer* dependerá da forma como aplicação foi programada.

Deve-se perceber que, como o TCP é um protocolo0 *full-duplex*, a transmissão de dados pode ocorrer nos dois sentidos da conexão simultaneamente. Portanto, existem duas janelas de transmissão, uma em cada sentido, e o tamanho de uma janela pode ser diferente da outra.



Protocolos de Transporte

TCP - Congestionamento

Em uma rede de computadores pode ocorrer de vários hospedeiros realizarem transferências de dados tentando utilizar a vazão máxima que seus enlaces permitem. Quando isso ocorre, o tráfego no núcleo da rede pode se elevar a níveis que comprometem a capacidade de transferência da rede, promovendo longas filas nos roteadores e podendo, inclusive, levar à perda de pacotes. Essa situação é conhecida como congestionamento e leva a uma séria degradação do desempenho da rede como um todo.

O objetivo do controle de congestionamento, no entanto, não pode ser somente evitar que o congestionamento ocorra. É preciso evitar o congestionamento e, ao mesmo tempo, **fazer com que a rede ofereça um bom desempenho, utilizando ao máximo a largura de banda disponível.**

Protocolos de Transporte

TCP - Congestionamento

Em uma rede de computadores pode ocorrer de vários hospedeiros realizarem transferências de dados tentando utilizar a vazão máxima que seus enlaces permitem. Quando isso ocorre, o tráfego no núcleo da rede pode se elevar a níveis que comprometem a capacidade de transferência da rede, promovendo longas filas nos roteadores e podendo, inclusive, levar à perda de pacotes. Essa situação é conhecida como congestionamento e leva a uma séria degradação do desempenho da rede como um todo.

Regulando a taxa de transmissão

Quando a rede percebe que está entrando em congestionamento (ou já se encontra nesse estado), a única solução é fazer com que os hospedeiros transmissores diminuam suas taxas de transferência cessar o congestionamento.

Protocolos de Transporte

TCP - Congestionamento

Basicamente existem duas formas de controle de congestionamento:

Controle de congestionamento fim a fim

A camada de rede não oferece suporte à camada de transporte para o controle de congestionamento, cabendo à camada de transporte reconhecer e tratar o congestionamento.

Controle de congestionamento assistido pela rede

Os roteadores (camada de rede) informam aos transmissores sobre a ocorrência de congestionamentos. Essa informação pode variar desde simples avisos da ocorrência de congestionamentos até o fornecimento de informações completas, como a taxa máxima de transmissão que pode ser utilizada.

Protocolos de Transporte

TCP - Congestionamento

O aviso ao transmissor sobre a ocorrência do congestionamento pode se dar de duas formas distintas:

Diretamente ao transmissor (retroalimentação direta)

Nesse tipo de aviso, o roteador deve enviar ao transmissor um **pacote de congestionamento (*choke packet*)** com as informações sobre o congestionamento.

Por intermédio do destinatário

Nesse tipo de aviso, os pacotes que passam por um congestionamento são marcados pelos roteadores. Ao chegar ao hospedeiro de destino, essa marcação é percebida e o hospedeiro transmissor é, então, notificado. Esse tipo de notificação requer que a mensagem chegue primeiro ao destino para que esses hospedeiros notifiquem o hospedeiro transmissor, levando mais tempo do que a retroalimentação direta.

Protocolos de Transporte

TCP - Congestionamento

O aviso ao transmissor sobre a ocorrência do congestionamento pode se dar de duas formas distintas:

Diretamente ao transmissor (retroalimentação direta)

Nesse tipo de aviso, o roteador deve enviar ao transmissor um **pacote de congestionamento (*choke packet*)** com as informações sobre o congestionamento.

Por intermédio do destinatário

Nesse tipo de aviso, os pacotes que passam por um congestionamento são marcados pelos roteadores. Ao chegar ao hospedeiro de destino, essa marcação é percebida e o hospedeiro transmissor é, então, notificado. Esse tipo de notificação requer que a mensagem chegue primeiro ao destino para que esses hospedeiros notifiquem o hospedeiro transmissor, levando mais tempo do que a retroalimentação direta.

Protocolos de Transporte

Ilustração