



# ELEMENTOS DA CAMADA DE TRANSPORTE

## A camada de transporte na Internet

### Mas para que serve a camada de transporte?

Executadas na camada de aplicação, as aplicações precisam de um modelo de rede no qual haja a entrega de uma mensagem (ou um fluxo de dados) tanto em um ponto de rede quanto em sua aplicação par no hospedeiro destino.

O objetivo da camada de transporte é, independentemente das redes físicas em uso, promover a confiabilidade na transferência de dados entre os hospedeiros origem e destino.

Como veremos no decorrer do nosso estudo, essa camada deve oferecer um **serviço de transferência confiável**, embora caiba à aplicação decidir sobre o seu uso.

## Serviço de Transporte

### Aspectos Fundamentais da Camada de Transporte

Em uma arquitetura de camadas, podemos afirmar que o objetivo geral de uma camada é oferecer serviços àquela imediatamente superior. No caso da camada de transporte, sua pretensão é oferecê-los à de aplicação.

### Atenção!

Lembre-se de que, neste estudo, estamos considerando a arquitetura TCP/IP, na qual não existem as camadas de sessão e de apresentação.

Como um dos principais objetivos da camada de transporte é ofertar um serviço confiável e eficiente a seus usuários, ela precisa oferecer, no mínimo, um serviço orientado à conexão e outro **sem conexão**.

Para atingir esse objetivo, a camada de transporte utiliza os serviços oferecidos pela de rede. No **serviço de transporte** orientado à **conexão** (serviço confiável), existem **três fases**:



Por meio de um controle apurado da conexão, esse serviço de transporte consegue verificar quais pacotes chegaram com erro ao destino e até mesmo aqueles que não foram enviados, sendo capaz de retransmiti-los até que os dados estejam corretos.

Já no **serviço de transporte sem conexão**, não existe nenhum controle sobre os pacotes enviados. Se um deles se perder ou chegar ao destino com erro, nada será feito para obter a sua recuperação.

Se a rede oferece um serviço com que garanta uma entrega sem erros, por que uma aplicação optaria por um serviço sem essa garantia?

A resposta é simples: por questões de desempenho.

Pelo fato de ser preciso cuidar de cada pacote no serviço orientado à conexão, verificando-os e retransmitindo-os em caso de necessidade, esse controle gera um overhead. Como nada disso é feito no serviço sem conexão, os pacotes são entregues no destino de forma mais simples e rápida.

Aplicações como transferência de arquivos e e-mail exigem que seus dados cheguem ao destino livres de erros. Dessa forma, elas utilizam um serviço orientado à conexão.

Ainda assim, em certas aplicações, o mais importante é a **chegada a tempo** de uma **informação**, mesmo que ela **contenha erros** ou que a **mensagem anterior tenha se perdido**.



No serviço de telefonia em rede, por exemplo, o atraso na transmissão tem um efeito pior que um pequeno ruído causado pela eventual perda de pacote.

## Endereçamento dos processos

### Endereçamento (camada de transporte)

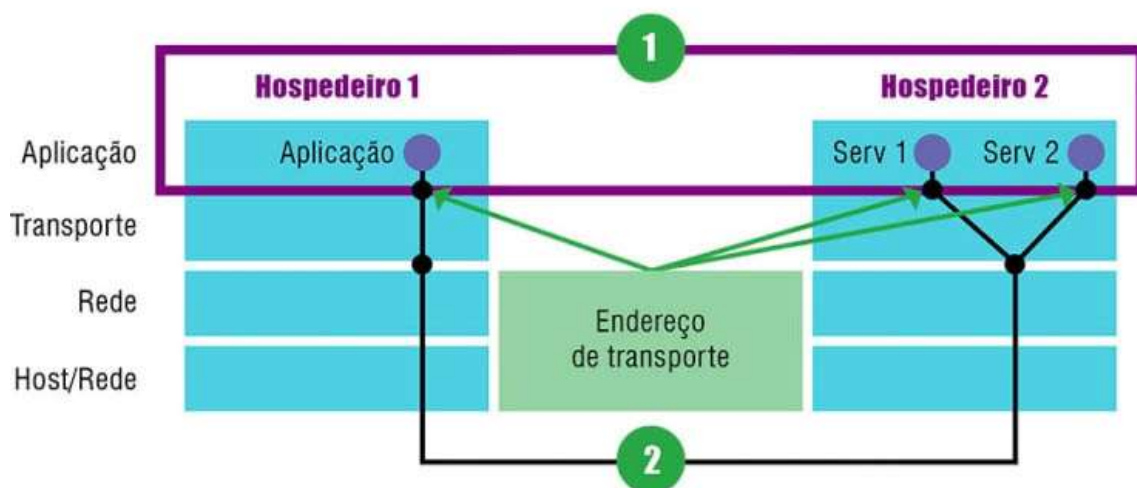
Quando seu programa solicita algo a um servidor, o sistema envia uma mensagem para ser entregue à aplicação que executa em um hospedeiro remoto. Mas podem existir várias aplicações nele.

### Como identificamos uma aplicação específica?

Surge neste momento o **endereçamento** no **nível de transporte**. Sua função é **identificar** em **qual aplicação** determinada mensagem **deve ser entregue**. Afinal, toda mensagem do protocolo de transporte carrega o endereço da aplicação.

Verificaremos agora a importância do endereçamento no nível de transporte. Afinal, é necessário indicar em qual aplicação os dados devem ser entregues por meio de seu endereço (de transporte). Assim, o hospedeiro destino consegue saber o destino deles.

Estudaremos mais adiante TCP e UDP, dois protocolos da camada de transporte da arquitetura TCP/IP. Neles, o endereço de transporte é conhecido como **porta**. Vejamos o esquema a seguir:



A partir da numeração presente na imagem anterior, continuaremos com o exemplo:

1. Como a aplicação do **hospedeiro 1** sabe em que endereço de transporte se encontra o **servidor no 2**? Uma possibilidade é que:
  - Ele esteja associado ao endereço há anos;
  - Aos poucos, todos os usuários da rede tenham se acostumado com isso.
2. Neste modelo, os serviços possuem **endereços estáveis** que podem ser impressos e distribuídos aos novos usuários quando eles se associam à rede.

## Atenção!

Um esquema alternativo é utilizar um processo especial denominado **servidor de nomes** (*name server*) ou, às vezes, **servidor de diretórios** (*directory server*). Para localizar o endereço de transporte correspondente a determinado nome de serviço, uma aplicação estabelece uma conexão com o servidor de nomes. Em seguida, envia uma mensagem especificando o nome do serviço, enquanto o servidor de nomes retorna o endereço.

## Multiplexação e demultiplexação

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

No hospedeiro destino, a camada de transporte recebe segmentos de dados da

camada de rede, tendo a responsabilidade de entregá-los ao processo de aplicação correto.



Um processo pode ter um ou mais endereços de transporte (conhecidos como portas na arquitetura TCP/IP) pelos quais dados passam da rede para o processo – e vice-versa.

Desse modo, a camada de transporte do hospedeiro destino os entrega diretamente a uma porta.

## Como o hospedeiro destino direciona à porta correta um segmento que chega?

Para essa finalidade, cada segmento da camada de transporte tem um conjunto de campos de endereçamento no cabeçalho. No receptor, a camada de transporte examina esses campos para identificar a porta receptora e direcionar o segmento a ela. A tarefa de entregar os dados contidos em um segmento para a porta correta é denominada **demultiplexação**.

Já a **multiplexação** consiste no trabalho de, no hospedeiro origem:

1. Reunir porções de dados provenientes de diferentes portas;

2. Encapsular cada porção de dados com as informações de cabeçalho (as quais, mais tarde, serão usadas na demultiplexação) para criar segmentos;
3. Passar os segmentos para a camada de rede.

Vejamos um exemplo, vamos pensar no computador que Eduardo utiliza em suas atividades.

1. Navegando na web, ele acessa seu e-mail e faz o download de arquivos usando um programa específico para isso.

De fato, o objetivo da multiplexação é **possibilitar** uma **melhor utilização do meio de comunicação** ao permitir que ele seja compartilhado pelos diversos programas utilizados.



2. Eduardo utiliza a Internet, cujo protocolo de transporte é o TCP.



3. Todos os programas operados por ele (browser web, cliente de e-mail e programa de transferência de arquivos) utilizam o TCP, que fará a transferência da informação até o destino.



A **multiplexação**, portanto, permite que vários programas possam utilizar o TCP ao mesmo tempo, fazendo, assim, com que Eduardo possa ter tantos programas quanto queira ao acessar a rede.



## Como o TCP sabe quem é quem?

Para fazer uso dele, um processo deve se registrar em uma porta (endereço de transporte) do protocolo TCP. Servidores possuem portas conhecidas, mas programas clientes se registram nas aleatórias.

Vamos supor que os programas de Eduardo se registraram nas seguintes **portas**:

### **Browser web**

11278.

### **Cliente de e-mail**

25786.

### **Transferência de arquivos**

3709.

Dessa maneira, o TCP pode identificar cada uma. Quando o browser envia uma solicitação a um servidor web, o TCP coloca na informação enviada o número de porta 11278. O servidor, portanto, já sabe que deve responder-lhe enviando a resposta para essa porta.

Observemos, por fim, a multiplexação e a demultiplexação na prática:

## **Multiplexação**

Ao receber mensagens das aplicações para envio, o protocolo de transporte as identifica por seus respectivos números de porta, permitindo, assim, que várias aplicações possam utilizá-los ao mesmo tempo.

## **Demultiplexação**

Quando recebe as mensagens do hospedeiro remoto para entregá-las em cada aplicação, o protocolo de transporte verifica o número da porta destino que a mensagem carrega e a entrega para o processo registrado nela.