

Tecnologia em Análise e Desenvolvimento de Sistemas - TADS

Redes de Computadores

Prof. Luciano Vargas Gonçalves

E-mail: luciano.goncalves@riogrande.ifrs.edu.br



INSTITUTO FEDERAL
Rio Grande do Sul
Campus Rio Grande





Aula 8 – Camada de Transporte

Camada de Transporte

Nos modelos RM-OSI e TCP/IP as camadas de Transporte são equivalentes;



Camada de Transporte

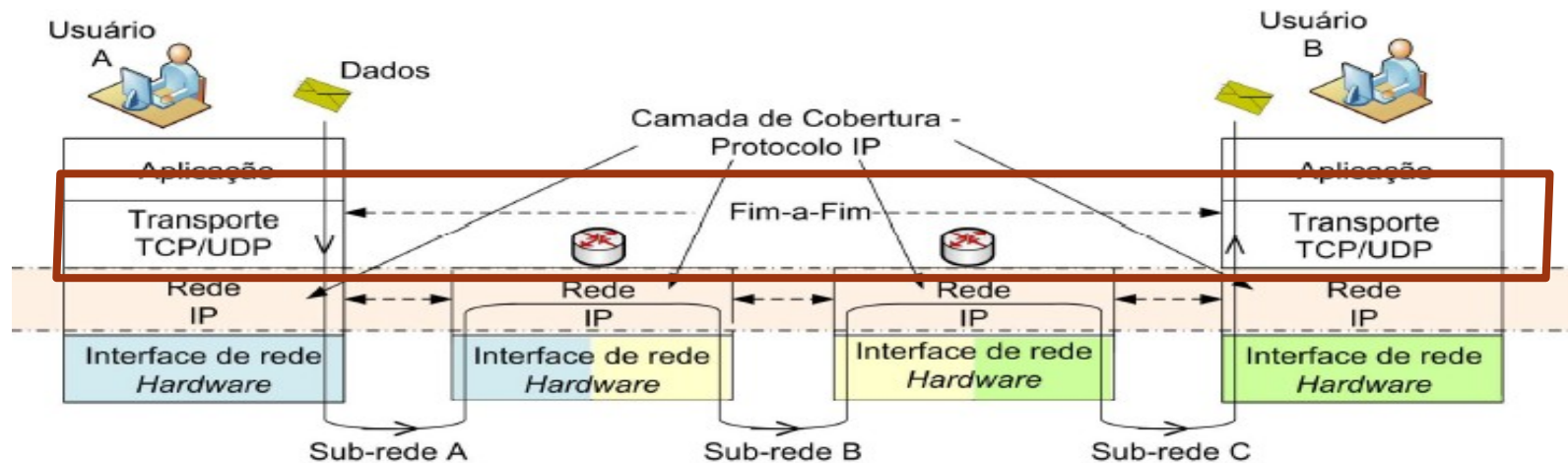
Objetivo da camada

- O Principal objetivo da camada de transporte é fornecer um serviço, confiável, econômico a seus usuários que, em geral, são processos presentes na camada de Aplicação;

Camada de Transporte

- **Canal lógico**

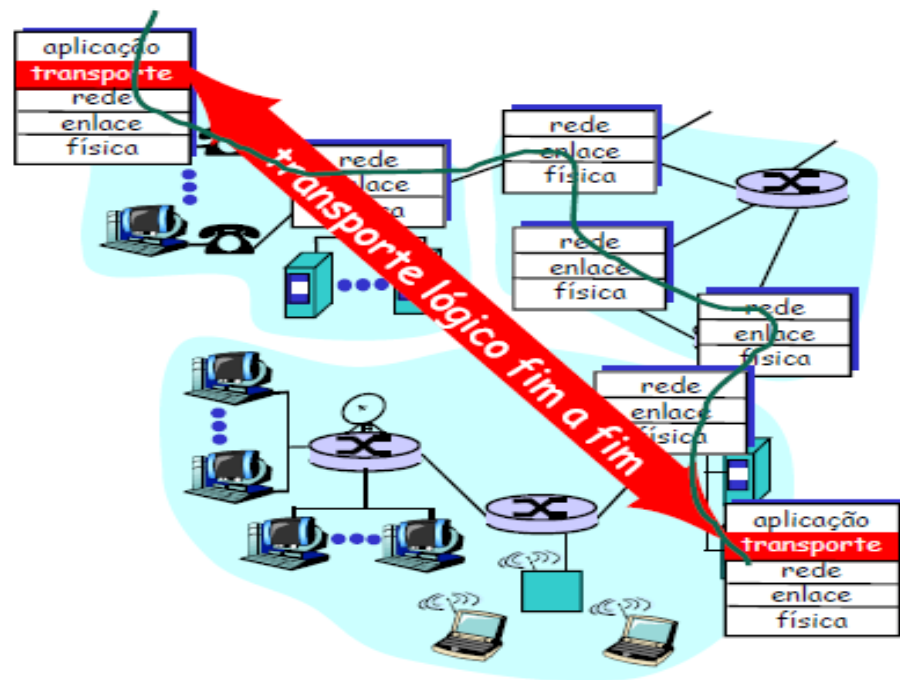
- Provê um **canal lógico de comunicação** entre processos em diferentes sistemas finais. Para a camada de aplicação, os sistemas finais estão diretamente conectados (FIM-a-FIM)



Camada de Transporte

Provê um canal lógico (Direto)

- Não existe fisicamente
- Apenas via software



Camada de Transporte

- **Camada transporte**
 - Usa os recursos da camada de rede, para:
 - Promover um *canal lógico seguro*,
 - Com controle de fluxo;
 - Com controle de erros;
 - Unidade: **Segmento**
 - Camada de Transporte **é implementada apenas nos host**
 - **Emissor e Receptor**

Camada de Transporte

- **Funções da camada transporte:**

- Controle de Perdas pacotes;
 - Controle de Repetições (pacotes descartados)
 - Ordenamento dos Dados (chegada pacotes)
 - Integridade dos Dados (testa o conjunto pacotes)
 - Segurança nas Comunicações
 - Qualidade de Serviço (QoS)
 - Controle de Fluxo (Receptor lento)
 - Segmentação
- **Trata problemas** ocorridos no envio das informações pela rede física (camadas 1,2,3);

Camada de Transporte

- **Camada de transporte trabalha com 2 tipos serviços:**
 - **Orientado a Conexão;**
 - **Serviço completo**
 - **Não Orientado a Conexão;**
 - **Serviço Básico**

Camada de Transporte

- **Orientado a Conexão:**

- Processos são **obrigados a estabelecerem uma conexão** antes da troca de informação e também a encerrarem uma conexão ao finalizar a troca informação.
- *“Semelhante ao cumprimento (Bom dia, Até mais ver)”*

Conexão é a troca de pacotes de sincronização, antes da troca de informações.

Processo Three-Way HandShack

Camada de Transporte

- **Conexões:**

- Baseadas no envio de segmentos de controle:
 - SYN (Synchronous) -> Inicia uma conexão;
 - ACK (Acknowledgement) Confirmação dos dados enviados;

Transmissão dos Dados.....

- FIN (Finnal) -> Finaliza uma conexão;



Prontos para transmissão
– Transmissão de Dados

Camada de Transporte

- **Orientado a Conexão:**

- Processos são obrigados a estabelecerem uma conexão antes da troca de informação e também a encerrarem uma conexão ao finalizar a troca informação.
- *Todos os pacotes devem ser confirmados pelo destinatário (Pacote - ACK)*

- **Analogia**

- *Carta Registrada, confirmação do recebimento, rastreamento dos pacotes*



Camada de Transporte

- **Não Orientado a Conexão**

- Não existe a necessidade de conexão para início de transmissão.
 - Dados são enviados diretamente
 - *Pacotes **não** são confirmados pelo destinatário;*

- Analogia:

- **Carta Comum**



Camada de Transporte - TCP

- Protocolos de Comunicação (TCP):
 - **TCP - Transmission Control Protocol**
 - Orientado a conexão
 - É necessário estabelecer a **conexão** para posterior transmissão;
 - Entrega **confiável e ordenada**
 - Pacotes enviados pelo Emissor, **disparam confirmações** no Receptor!!!
 - O Emissor aguarda confirmações para envio de novos pacotes.

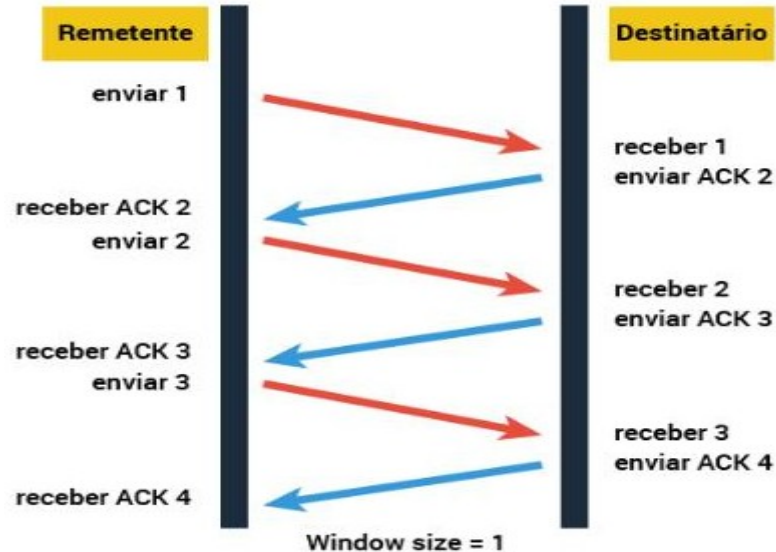
Camada de Transporte - TCP

- **Protocolo TCP - Características**

- É ponto-a-ponto (UM para UM)
 - Um transmissor e um receptor
- Transmissão full duplex
- Fluxo de dados bidirecional na mesma conexão
- Controle de fluxo
 - Receptor não será afogado pelo transmissor
- Controle de congestionamento
 - Evitar a saturação dos enlaces da rede

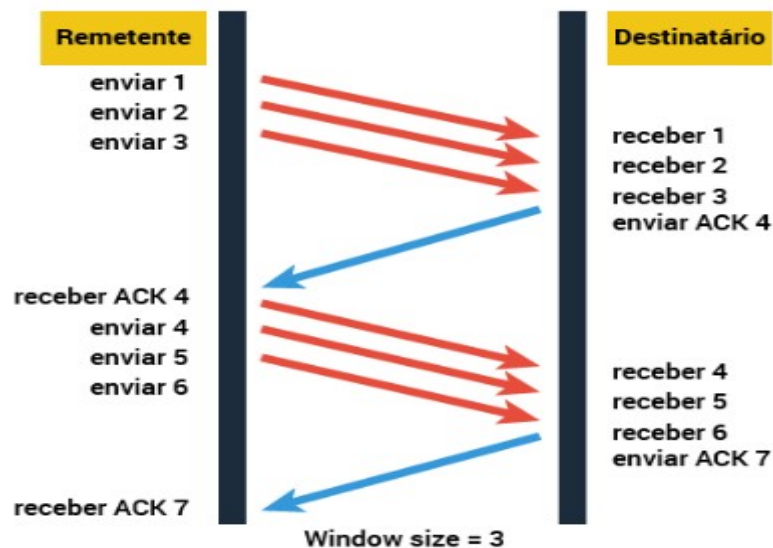
Camada de Transporte - TCP

- Protocolo de Comunicação (TCP):
 - Envio de informações;
 - Confirmações dos pacotes (ACK)



Camada de Transporte - TCP

- Protocolos de Comunicação (TCP):
 - Confirmações dos pacotes (ACK), com janela deslissante igual a 3 (3 envios 1 confirmação)



Camada de Transporte - TCP

- **Protocolo TCP - Características**

- É ponto-a-ponto
 - Um transmissor e Um receptor
- Transmissão full duplex
 - Fluxo de dados bidirecional na mesma conexão
- Controle de fluxo
 - Receptor não será afogado pelo transmissor
- Controle de congestionamento
 - Evitar a saturação dos enlaces da rede

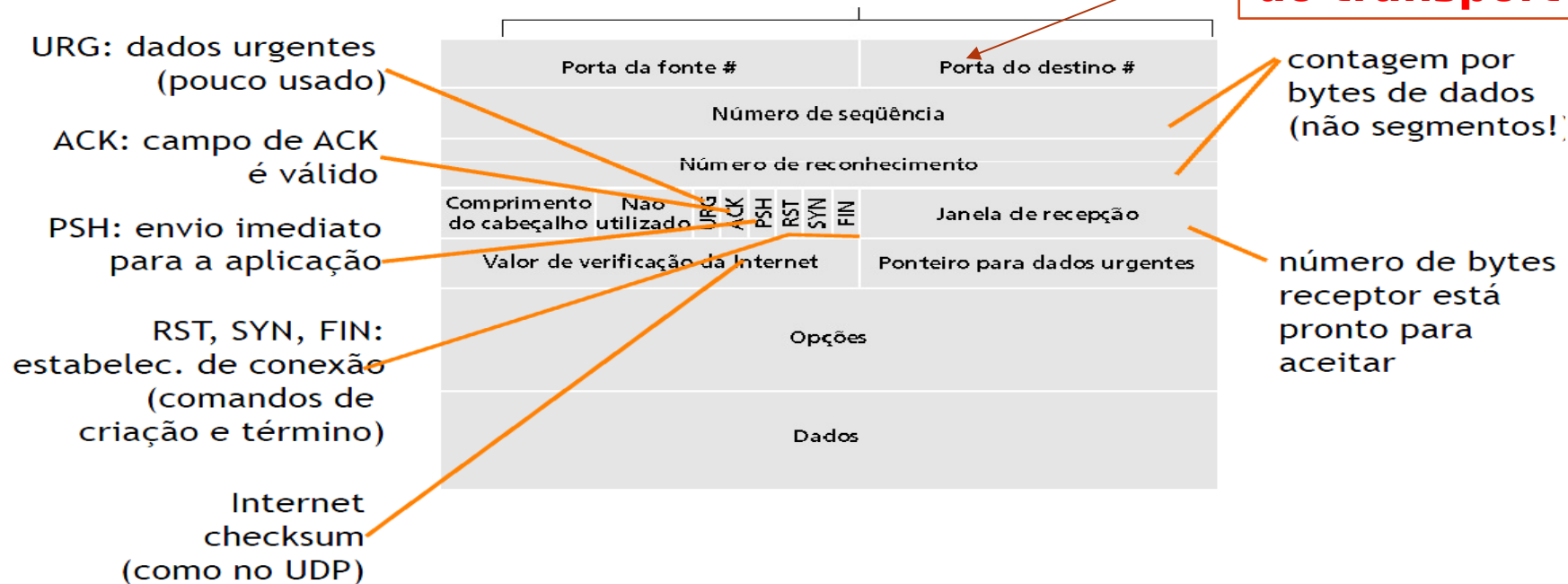
Camada de Transporte - TCP

- **Protocolo TCP - Características**

- Provê um serviço confiável (TCP) sobre o serviço não confiável (IP)
- Serviços implementados sobre TCP:
 - Transmissão de Páginas Http
 - Transmissão de EMAIL (POP3, IMAP, SMTP)
 - Download de Arquivos;
 - Etc.....

Camada de Transporte - TCP

- Cabeçalho Datagrama TCP



Camada de Transporte - UDP

- Protocolo de Transporte (UDP):
 - **UDP - User DataGram Protocol**
 - **Não Orientado a Conexão;**
 - Não necessita criar a conexão;
 - Entrega não-confiável e não-ordenada
 - Não gera confirmações pelo receptor!!!
 - Rápido e Barato.

Camada de Transporte - UDP

- **UDP - User DataGram Protocol (RFC 768)**
 - Protocolo de transporte mínimo
 - Serviço de melhor esforço.
 - Segmentos UDP podem ser perdidos
 - Entregues à aplicação fora de ordem
 - Sem conexão
 - Não há “confirmação” entre remetente e o receptor
 - Tratamento independente de cada segmento UDP

Camada de Transporte - UDP

- **Por quê é necessário?**

- Elimina o estabelecimento de conexão
 - Menor latência (tempo entrega)
- É simples
- Não mantém “estado” da conexão no Emissor e no Receptor
- Cabeçalho de segmento reduzido
- Não há controle de congestionamento
- UDP pode transmitir tão rápido quanto desejado (e possível)

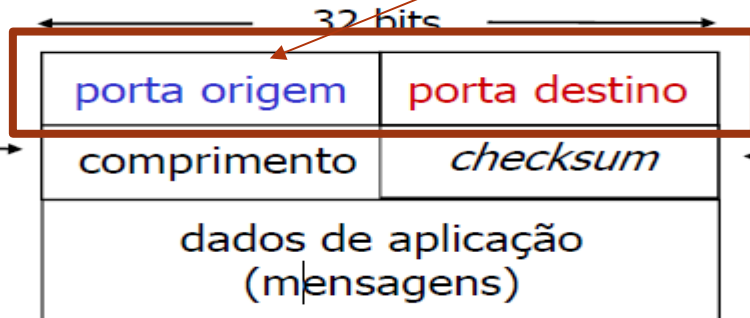
Camada de Transporte - UDP

- **Utilizado para aplicações multimídias**
 - Tolerantes a perdas
 - Sensíveis à taxa de transmissão (tempo)
- Transferência confiável com UDP?
 - É necessário acrescentar confiabilidade na camada de aplicação caso necessário.

Camada de Transporte

- Cabeçalho do DATAGRAMA UDP é composto de:
 - **Porta** de *Origem e Destino*
 - Comprimento e Checksum

Comprimento em bytes do segmento UDP, incluindo cabeçalho



Endereçamento Do transporte

soma de verificação

Endereçamento - Transporte

- **Conceito de Porta de Comunicação**

- Um Prédio tem um endereço físico(rua, bairro, cep);
- Todo apartamento tem o mesmo endereço físico;
- O que diferencia um apartamento de outro?



Endereçamento - Transporte

- **Conceito de Porta de Comunicação**

- Prédio tem um endereço físico (rua, bairro, cep);
 - Todo apartamento tem o mesmo endereço físico;
- O que diferencia um apartamento de outro?



Número do Apartamento



Endereçamento Secundário (Interno)

Endereçamento - Transporte

- Conceito de Porta de Comunicação
 - Um computador tem **endereço físico (IP)**;
 - O que diferencia uma aplicação (programa) de outro?



Aplicativos Rede

Endereçamento - Transporte

- Conceito de Porta de Comunicação
 - **Endereçamento da camada de transporte (16bits)**



Endereço de Porta 16bits
65536 endereços para aplicações



Aplicativos Rede

Endereçamento - Transporte

- **Conceito de Porta de Comunicação**

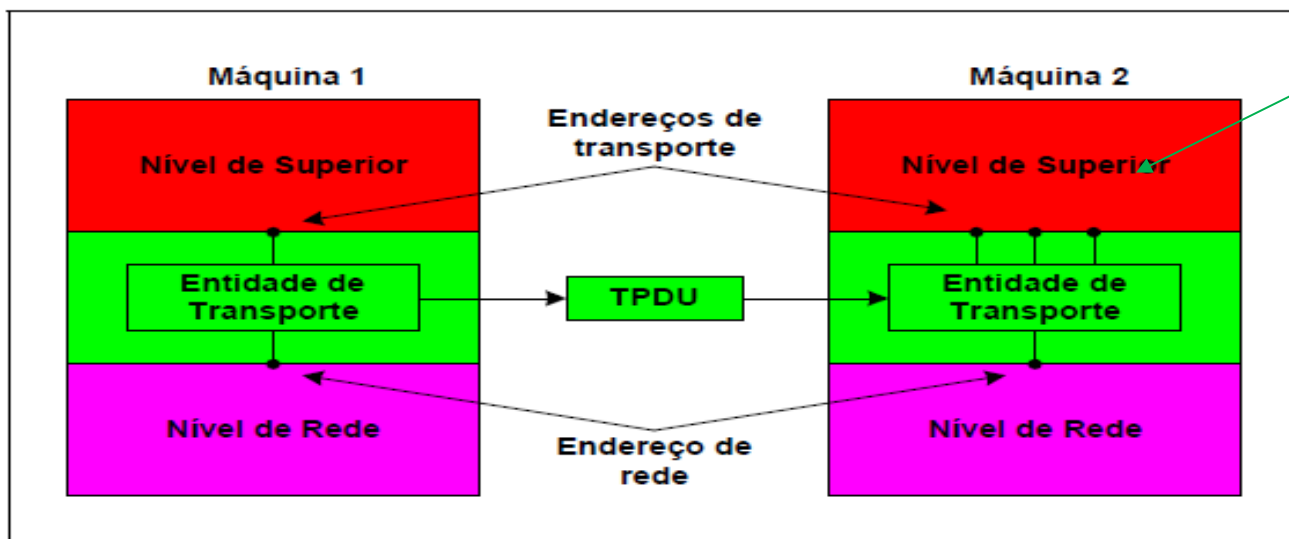
- Atrás de cada porta existe um serviço de rede (Camada de aplicação) responsável por receber ou enviar os pacotes:
 - **Servidor:** Http(80), Email(25), Telnet(23), SSH(21)
 - Portas até 1024 são privadas, uso de sistemas padronizados
 - **Clientes:** Http(??), Email(??), Telnet(??), SSH(??), etc.
 - Usam portas acima de 1024 - Portas públicas



Endereço de Porta 16bits
65536 endereços para aplicações

Camada de Transporte

- A entidade de transporte tem fluxo único com a camada de Rede;
- Fluxo é compartilhado pelas várias aplicações;

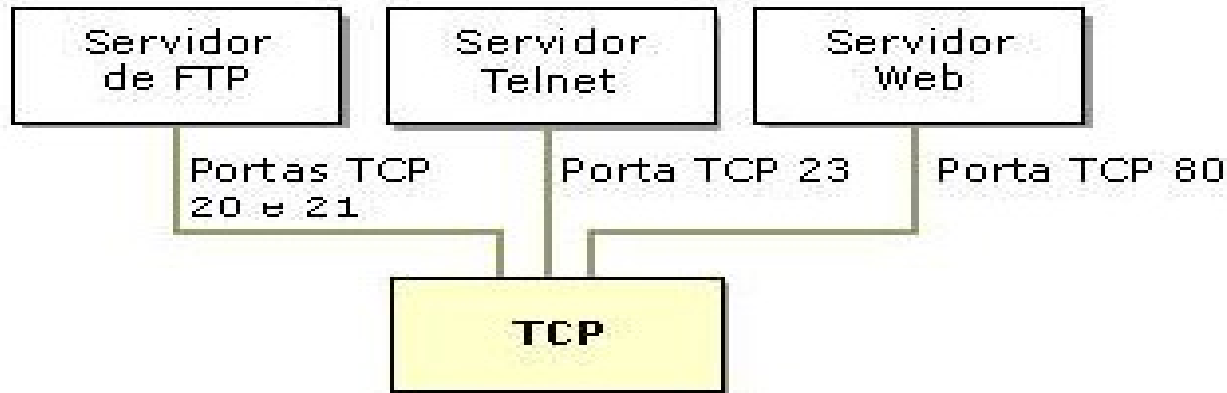


Portas
Comunicações,
Várias aplicações

Figura 1. Entidade de transporte

Camada de Transporte

- Atrás das Portas UDP e TCP têm os serviços de aplicação



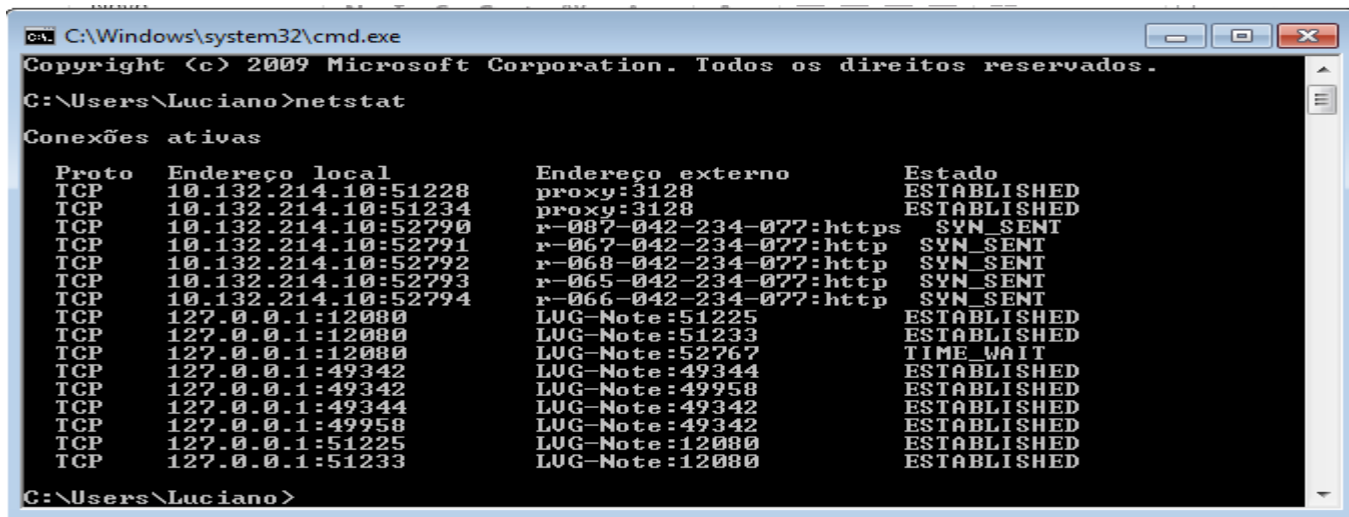
Camada de Transporte

- **Existem 65536 portas de comunicação(16bits):**
 - 1 a 1024 são portas privadas de uso restrito; Outras são liberadas para uso qualquer; Definidas por Padrão pela IANA.

Porta	Tipo Serviço	Serviço	descrição
21	TCP/UDP	FTP	Transferência de arquivos
22	TCP/UDP	SSH	Terminal Remoto Seguro
23	TCP/UDP	Telnet	Terminal Remoto
25	TCP/UDP	SMTP	Envio de Email
80	TCP/UDP	HTTP	Páginas WEB
110	TCP/UDP	POP	Recebimento de email
443	TCP/UDP	Https	Páginas Web criptografadas

Camada de Transporte

- Como examinar as portas da máquina?
 - O comando “netstat” retorna todas as portas em uso em uma máquina;



```
C:\Windows\system32\cmd.exe
Copyright (c) 2009 Microsoft Corporation. Todos os direitos reservados.
G:\Users\Luciano>netstat

Conexões ativas

Proto  Endereço local      Endereço externo    Estado
TCP    10.132.214.10:51228   proxy:3128          ESTABLISHED
TCP    10.132.214.10:51234   proxy:3128          ESTABLISHED
TCP    10.132.214.10:52790   r-087-042-234-077:https SYN_SENT
TCP    10.132.214.10:52791   r-067-042-234-077:http  SYN_SENT
TCP    10.132.214.10:52792   r-068-042-234-077:http  SYN_SENT
TCP    10.132.214.10:52793   r-065-042-234-077:http  SYN_SENT
TCP    10.132.214.10:52794   r-066-042-234-077:http  SYN_SENT
TCP    127.0.0.1:12080      LUG-Note:51225       ESTABLISHED
TCP    127.0.0.1:12080      LUG-Note:51233       ESTABLISHED
TCP    127.0.0.1:12080      LUG-Note:52767       TIME_WAIT
TCP    127.0.0.1:49342      LUG-Note:49344       ESTABLISHED
TCP    127.0.0.1:49342      LUG-Note:49958       ESTABLISHED
TCP    127.0.0.1:49344      LUG-Note:49342       ESTABLISHED
TCP    127.0.0.1:49958      LUG-Note:49342       ESTABLISHED
TCP    127.0.0.1:51225      LUG-Note:12080       ESTABLISHED
TCP    127.0.0.1:51233      LUG-Note:12080       ESTABLISHED

G:\Users\Luciano>
```

Camada de Transporte

- 1) Abra um terminal e digite: **netstat | grep tcp**
- 2) Após abra um Browser e abra várias conexões com sites
- 3) Após repita o comando do item 1
- 4) Várias novas conexões aparecerão

```
luciano@luciano-pc:~$ netstat | grep tcp
tcp        1      0 77 luciano-pc.local:59937 ce-in-f95.1e100.n:https ÚLTIMO_ACK
tcp        0      0 luciano-pc.local:35130 gru06s25-in-f3.1e:https ESTABELECIDA
tcp        0      0 luciano-pc.local:47960 gru06s26-in-f14.1:https ESTABELECIDA
tcp        0      0 luciano-pc.local:33710 gru06s25-in-f5.1e:https ESTABELECIDA
tcp        0      0 luciano-pc.local:47270 74.121.140.94:http    ESTABELECIDA
tcp        0      0 luciano-pc.local:40988 64.233.190.189:https  ESTABELECIDA
tcp        0      0 luciano-pc.local:47985 gru06s26-in-f14.1:https ESTABELECIDA
tcp        0      0 luciano-pc.local:47985 gru06s26-in-f14.1:https ESTABELECIDA
tcp6       1      0 ip6-localhost:41109 ip6-localhost:ipp     ESPERANDO_FECHAR
```

conexões

Portas

Camada de Transporte

- Informações do NETSTATUS
 - **Proto** – Protocolo que pode ser TCP, UDP, TCPv6, ou UDPv6
 - **Local Address** – Endereço local (seu PC)
 - **Foreign Address** – Endereço remoto
 - **State** – Exibe o estado da conexão de rede que podem ser
 - CLOSE_WAIT, CLOSED, ESTABLISHED,
 - FIN_WAIT_1, FIN_WAIT_2, LAST_ACK, LISTEN, SYN_RECEIVED, SYN_SEND, e TIME_WAIT
-

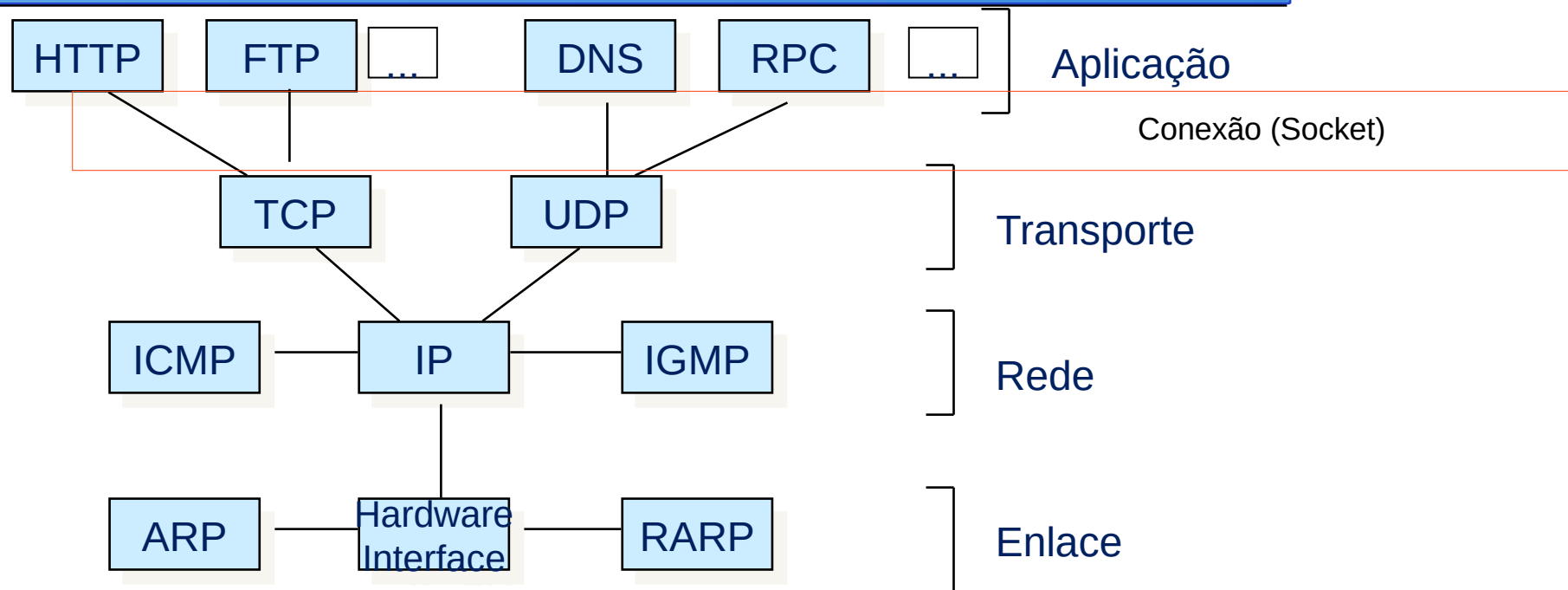
Camada de Transporte

- Alguns Parâmetros do NETSTATUS
 - -a : exibe todas as conexões e as portas TCP e UDP.
 - -e : exibe estatísticas básicas de conexões
 - -n : exibe os números das portas ao invés do nome.
 - -o : exibe o PID (Process ID) que
 - -p protocolo: Mostra as conexões por protocolo.
 - -s : exibe estatísticas por protocolo.
 - -r : exibe a tabela de roteamento interna.

Características do TCP e UDP

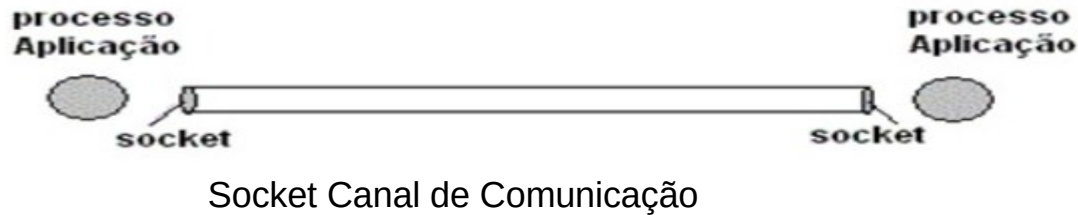
UDP	TCP
Serviço sem conexão; nenhuma sessão é estabelecida entre os hosts.	Serviço orientado por conexão; uma sessão é estabelecida entre os hosts.
UDP não garante ou confirma a entrega ou sequência dos dados.	TCP garante a entrega através do uso de confirmações e entrega sequenciada dos dados.
Os programas que usam UDP são responsáveis por oferecer a confiabilidade necessária ao transporte de dados.	Os programas que usam TCP têm garantia de transporte confiável de dados.
UDP é rápido, necessita de baixa sobrecarga e pode oferecer suporte à comunicação ponto a ponto e ponto a vários pontos.	TCP é mais lento, necessita de maior sobrecarga e pode oferecer suporte apenas à comunicação ponto a ponto.

Características do TCP e UDP

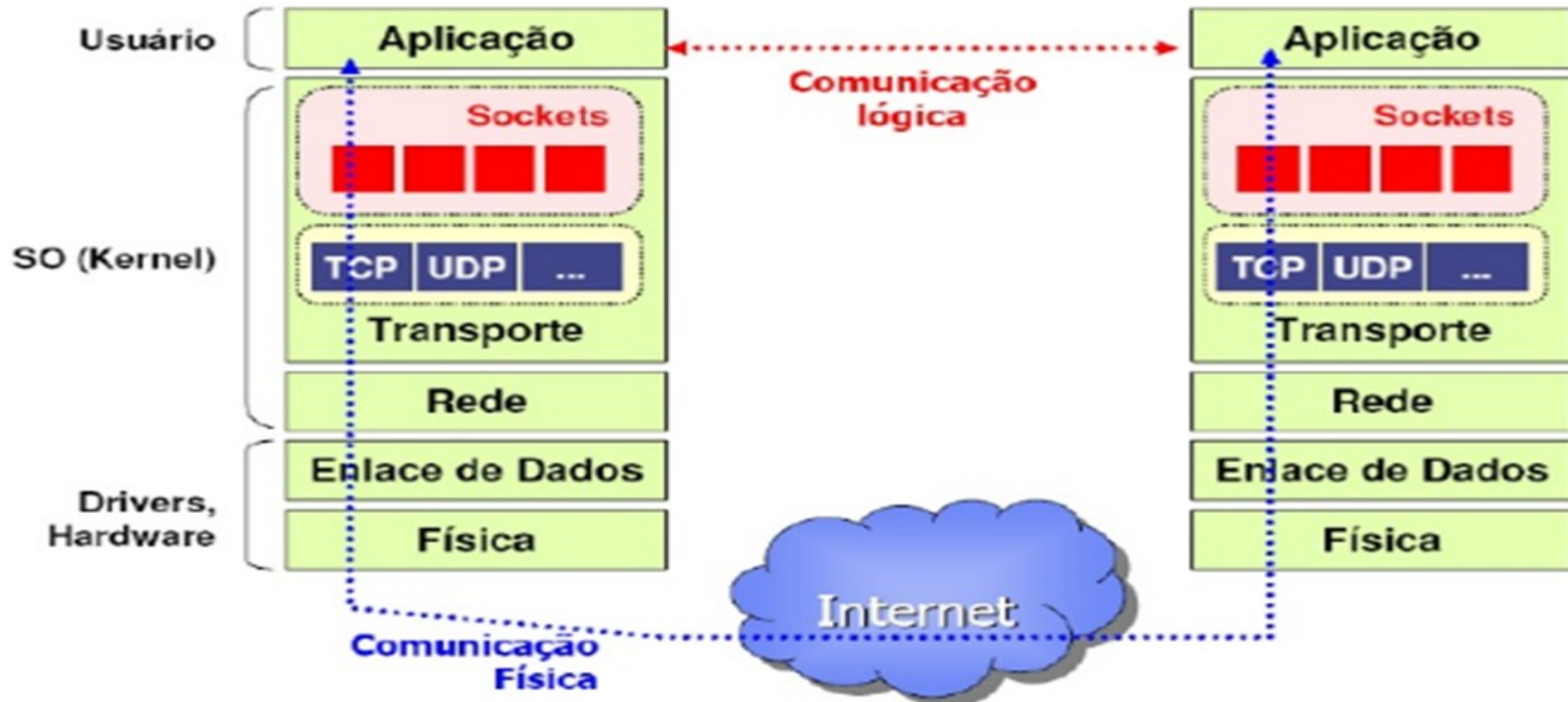


Conexão por Socket

- Interface padrão para comunicação entre processos em redes TCP/IP
- Nasceu com o Unix de Berkeley
- Implementada hoje em vários **S.O**
- Programar com sockets pode ser visto como desenvolver um protocolo de aplicação.



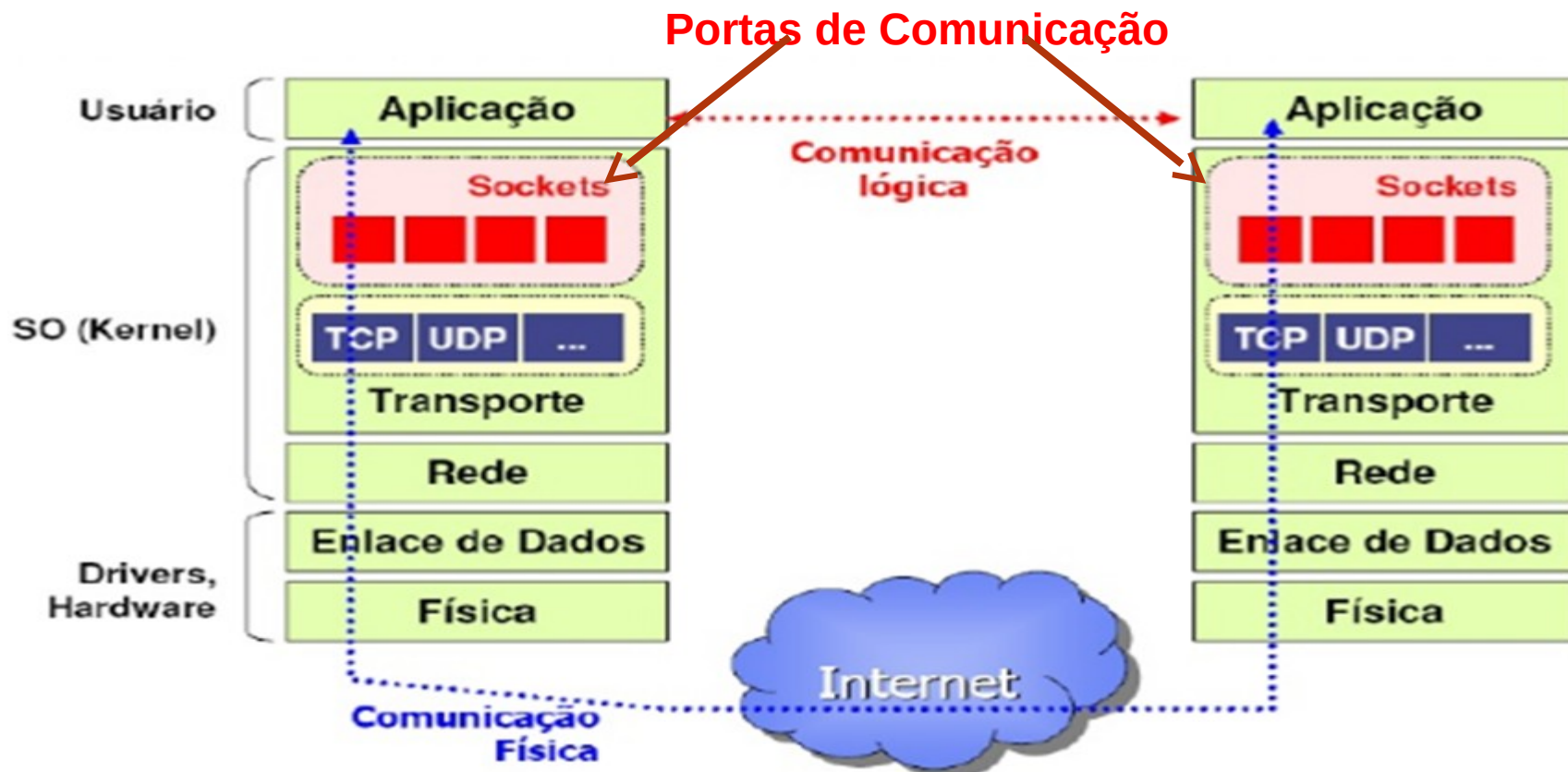
Conexão por Socket



Conexão por Socket

- Interliga Software da Camada de Aplicação com os protocolos da camada de Transporte.
- Faz uso das portas de comunicação da camada de Transporte.
- Conecta cada aplicação a um porta de comunicação.

Conexão por Socket

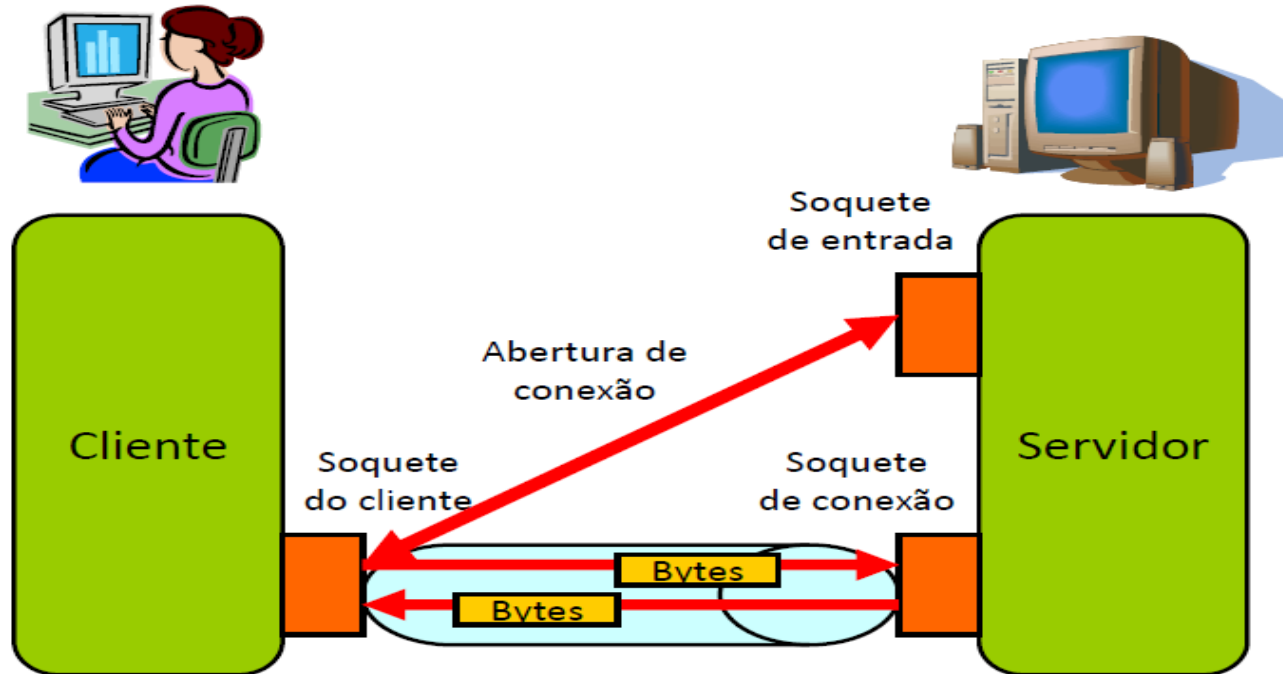


API Socket – Comandos básicos

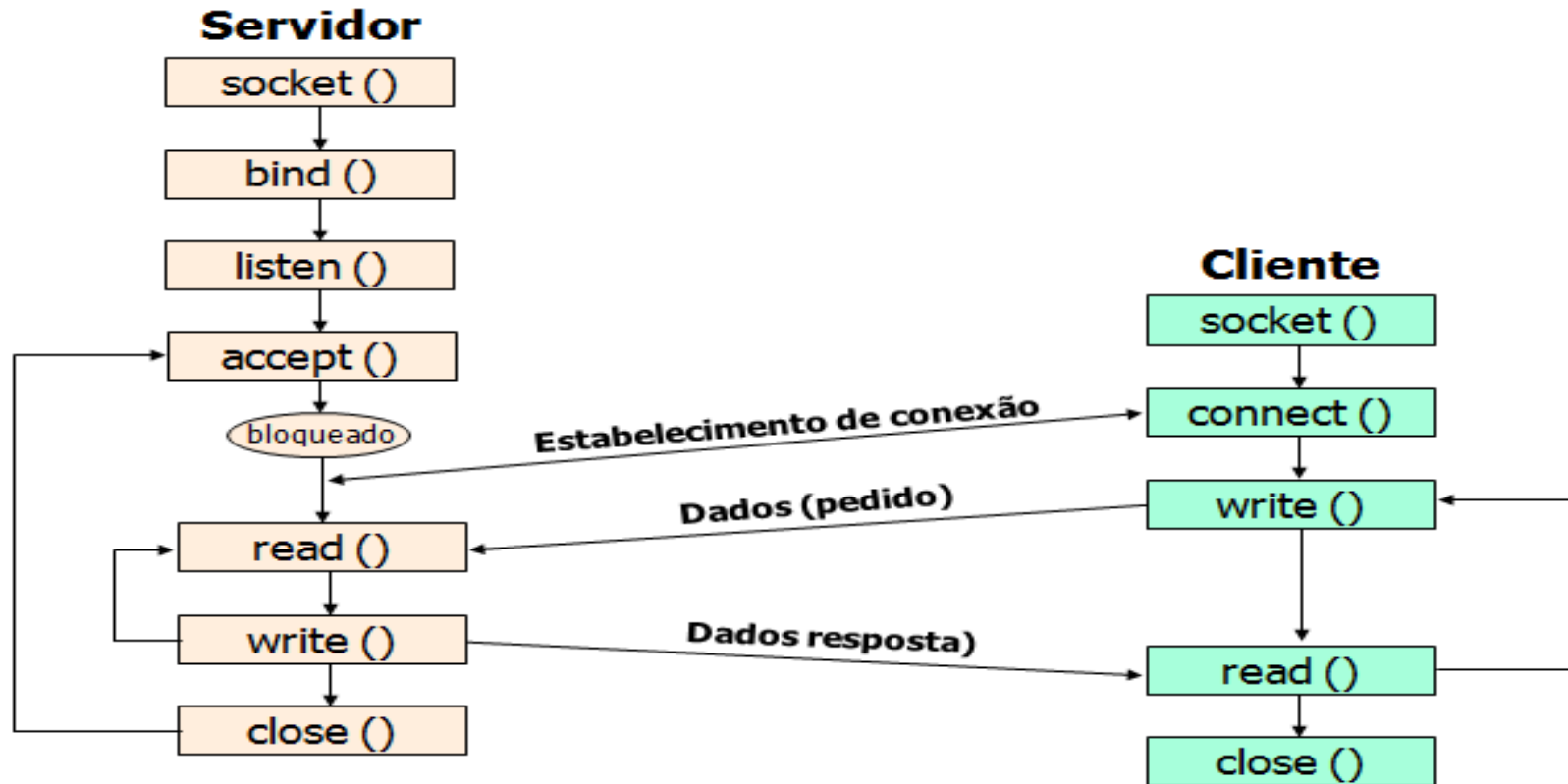
socket	Cria um novo descritor para comunicação
connect	Iniciar conexão com servidor
write	Escreve dados em uma conexão
read	Lê dados de uma conexão
close	Fecha a conexão
bind	Atribui um endereço IP e uma porta a um socket
listen	Coloca o socket em modo passivo, para “escutar” portas
accept	Bloqueia o servidor até chegada de requisição de conexão
recvfrom	Recebe um datagrama e guarda o endereço do emissor
sendto	Envia um datagrama especificando o endereço

Socket TCP

Servidor TCP, um Socket para tratar abertura de conexão
Outro Socket para troca de dados

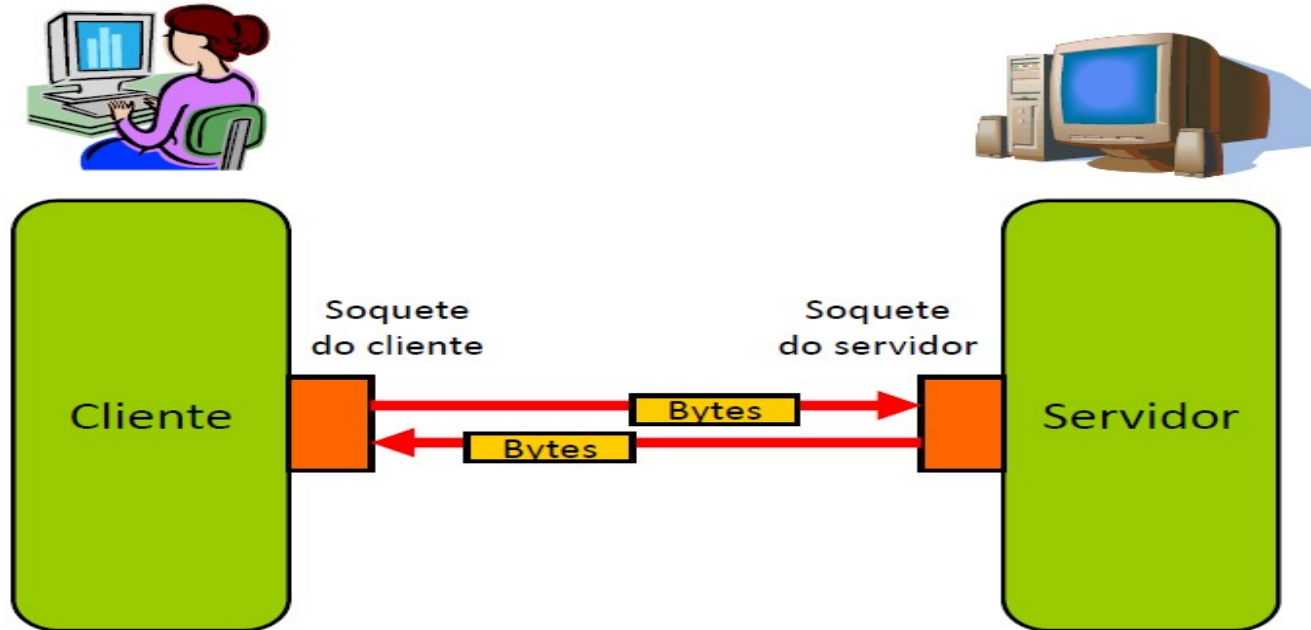


Socket TCP – Fluxograma

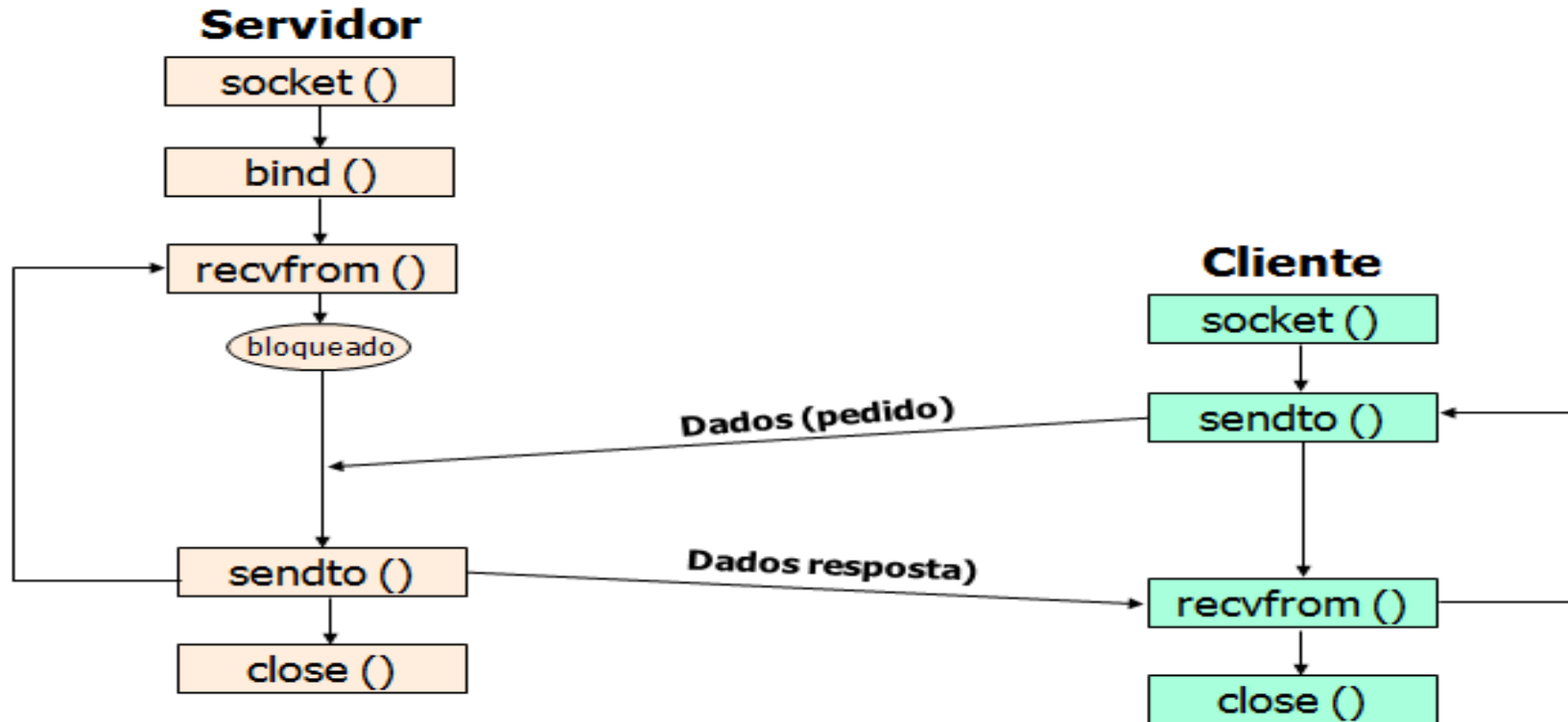


Socket UDP

Servidor UDP – um Socket para troca de dados



Socket UDP



JAVA - SOCKET

- Em Java a interligação possui suporte no pacote *java.net*.
 - HTTP
 - FTP
 - *Soquetes básicos,*
 - *Orientados à conexão* usando o *protocolo TCP*
 - *Orientados à datagramas,* usando o *protocolo UDP*

Servidor TCP

```
import java.io.*;
import java.net.*;

public class SimpleJavaServer {
    public static void main(String[] args) {
        try {
            ServerSocket s = new ServerSocket(9999);
            String str;
            while (true) {
                Socket c = s.accept();
                InputStream i = c.getInputStream();
                OutputStream o = c.getOutputStream();
                do {
                    byte[] line = new byte[100];
                    i.read(line);
                    o.write(line);
                    str = new String(line);
                } while ( !str.trim().equals("bye") );
                c.close();
            }
        } catch (Exception err){
            System.err.println(err);
        }
    }
}
```

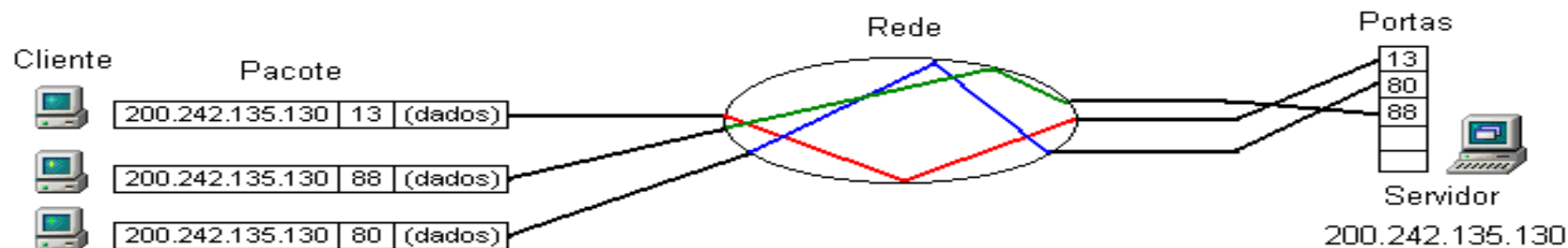
C

```
import java.net.*;
import java.io.*;
public class SimpleJavaClient {
    public static void main(String[] args) {
        try {
            Socket s = new Socket("127.0.0.1", 9999);
            InputStream i = s.getInputStream();
            OutputStream o = s.getOutputStream();
            String str;
            do {
                byte[] line = new byte[100];
                System.in.read(line);
                o.write(line);
                i.read(line);
                str = new String(line);
                System.out.println(str.trim());
            } while ( !str.trim().equals("bye") );
            s.close();
        }
        catch (Exception err) {
            System.err.println(err);
        }
    }
}
```

Cliente TCP

Camada de Transporte

Interação Cliente / Servidor Várias conexões



Socket – Leitura próxima Aula

- Exemplo de Socket C
 - <http://equipe.nce.ufrj.br/thome/grad/so/Trabs/Olds/Tutorial%20de%20Sockets.pdf>
 - <http://www-usr.inf.ufsm.br/~giovani/sockets/sockets.txt>
 - <http://www-usr.inf.ufsm.br/~giovani/sockets/sockets1.pdf>
 -

Dúvidas??

