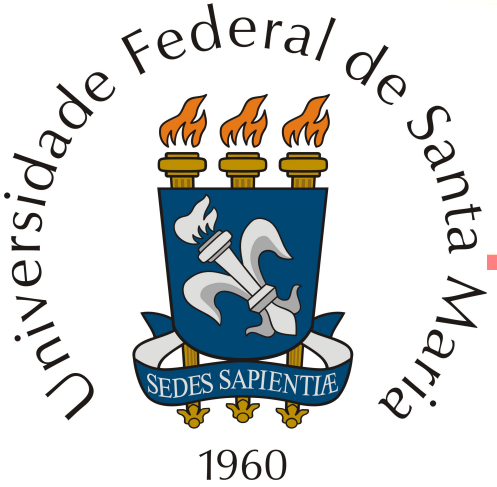


Universidade Federal de Santa Maria

Departamento de Eletrônica e Computação

Programação com sockets

Redes de comunicação de dados

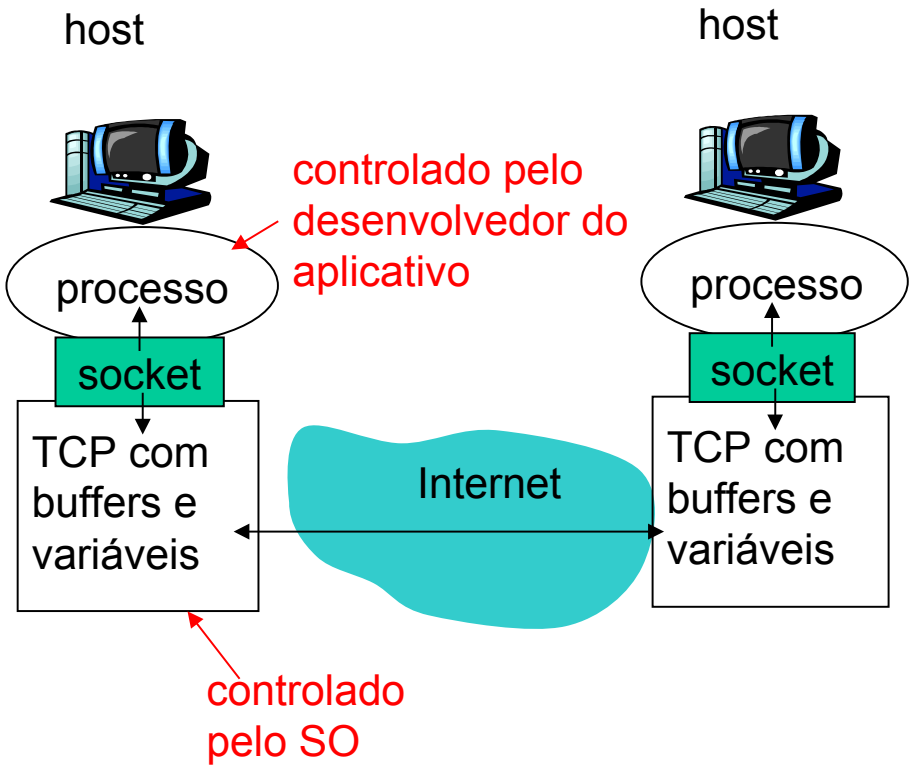
Carlos Henrique Barriquello
barriquello@gmail.com

Soquetes (sockets)

processo envia/recebe mensagens de/para seu *socket.*

"socket" é um ponto de conexão com a rede (como uma tomada)

Na prática, utiliza-se uma API (interface de programação) para a programação com *sockets*



Programação com sockets

Objetivo: progrmar uma aplicação cliente/servidor que se comunica usando sockets

Socket API

introduzido no BSD4.1 UNIX, 1981

modelo cliente/servidor

Dois tipos de serviço de transporte via API de sockets:

- **datagrama** (não confiável) usando UDP

- **fluxo de bytes**, confiável usando TCP

socket

Uma interface *host-local*, *criada pello aplicativo controlada pelo SO* pela qual o processo do aplicativo pode enviar e receber mensagens de/para outro processo de aplicação

Interação Cliente/Servidor

Cliente

1. Cria um socket TCP
2. Comunica
3. Fecha a conexão

Servidor

1. Cria um socket TCP
2. Repetidamente:
   a. Aceita nova conexão
   b. Comunica
   c. Fecha a conexão

Programação com sockets em Java

1 - Abrir a conexão (cliente):

```
import java.io.* ;    // streams
import java.net.* ;   // sockets
Socket clientSocket = new Socket
   (“www.javasoft.com”, 80);
```

Programação com sockets em Java

2 - Pegando os fluxos (*streams*) de entrada e saída:

```
DataInputStream inbound = new
 DataInputStream
    ( clientSocket.getInputStream( ) );

DataOutputStream outbound = new
  DataOutputStream
    ( clientSocket.getOutputStream( ) );
```

Programação com sockets em Java

3 - Utilizando os fluxos de entrada e saída:

```
outbound.writeInt( 3 );
outbound.writeUTF( "Hello" );

int k =    inbound.readInt( );
String s = inbound.readUTF() ;
```

Programação com sockets em Java

4 – Fechando os streams de entrada e saída:

```
inbound.close () ;
outbound.close () ;
```

5 – Fechando o socket:

```
clientSocket.close() ;
```

Programação com sockets em Java

1 - Criar o socket do servidor:

```
ServerSocket serverSocket =
   new ServerSocket (80, 5);
```

2 - Aguardar conexões de clientes:

```
Socket clientSocket =
   serverSocket.accept ();
```

Programação com sockets em Java

3 - Criar streams de entrada e saída do cliente:

```
DataInputStream inbound = new
  DataInputStream

     ( clientSocket.getInputStream( ) ) ;


DataOutputStream outbound = new
  DataOutputStream
     ( clientSocket.getOutputStream( ) ) ;
```

Programação com sockets em Java

4 - Comunicando com o cliente:

```
int k =    inbound.readInt( );
String s = inbound.readUTF() ;

outbound.writeInt( 3 );
outbound.writeUTF( "Hello" );
```

Programação com sockets em Java

5 - Fechando fluxos e socket cliente:

```
inbound.close () ;
outbound.close () ;
clientSocket.close() ;
```

6 - Fechando o socket servidor:

```
serverSocket.close() ;
```