

Redes de Computadores

Profa. Ms. Valéria Pinheiro



UNIVERSIDADE
FEDERAL DO CEARÁ
CAMPUS DE RUSSAS

Aula 08 – Programação com sockets

Introdução

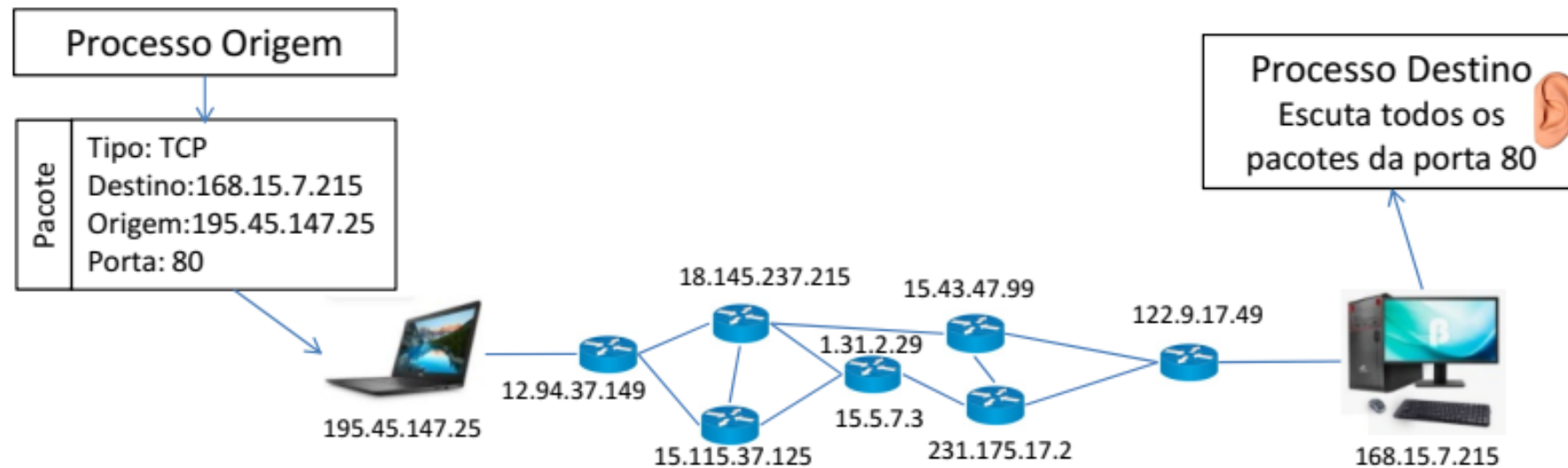
- Objetivo: aprender como construir aplicações cliente/servidor que se comunicam com sockets
- Socket: porta entre o processo da aplicação e protocolo fim-a-fim

Comunicação entre Processos



Comunicação entre Processos

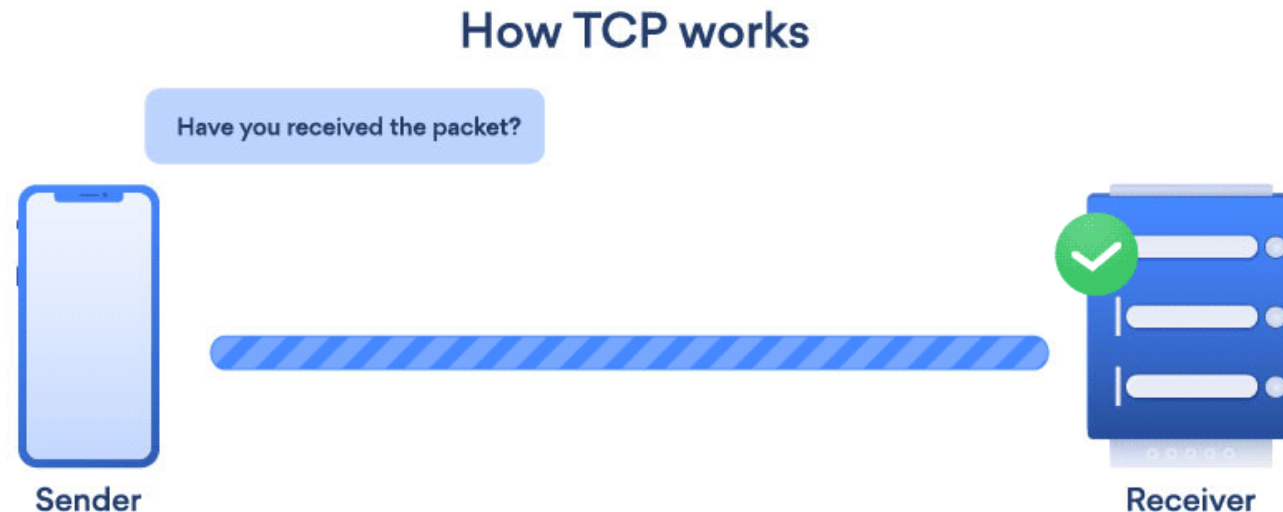
- **Endereço IP:** identificador de dispositivo conectado à Internet
- **Porta:** número associado a um processo que aguarda por pacotes
Portas reservadas de 0 a 1023 para serviços exclusivos



Serviços de transporte disponíveis para aplicações

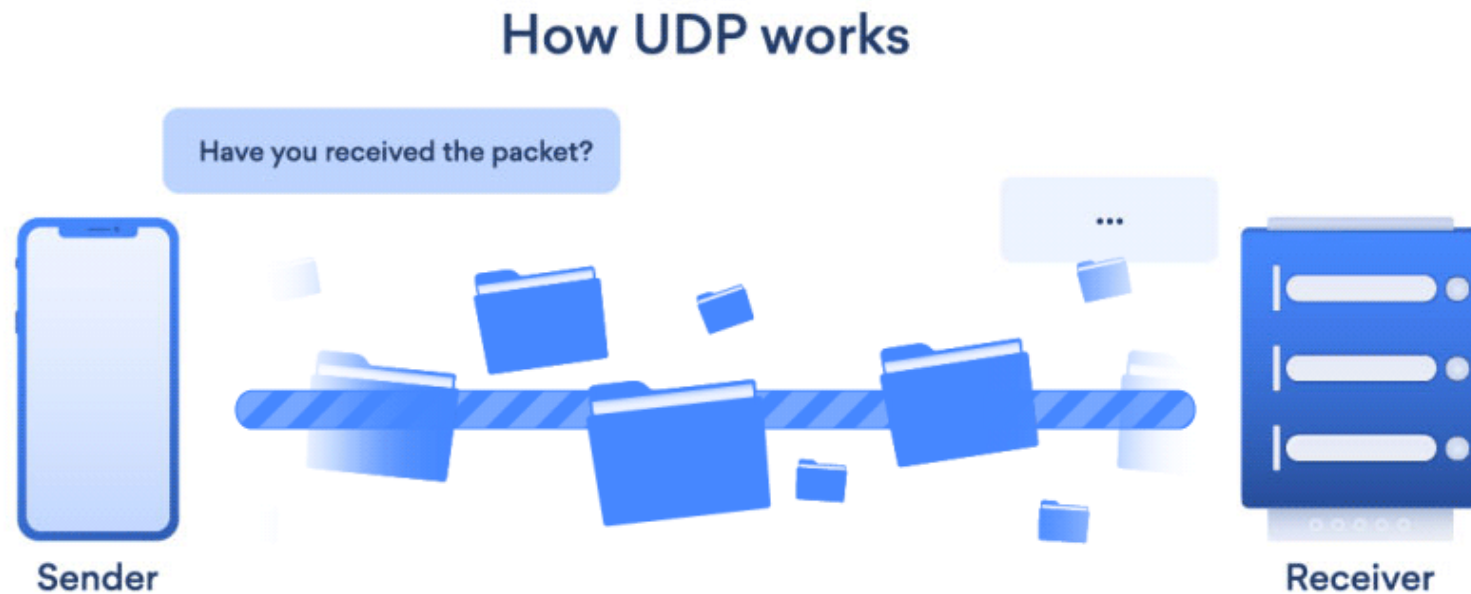
- **Serviço TCP:**

- Confiável
- Orientado a conexão
- Retransmite pacotes perdidos



Serviços de transporte disponíveis para aplicações

- **Serviço UDP:**
 - Melhor esforço
 - Não orientado a conexão



Módulo sockets - Python

O python utiliza sockets através da classe *socket*, que deve ser importada em todos os programas que necessitem a utilizar obedecendo o formato:

```
import socket
```

Programas que utilizam sockets trabalham sempre em formato **cliente-servidor**, e para isto é sempre necessário criar o programa servidor, que escutará conexões em uma porta, e o programa cliente, que fará a conexão a algum *socket* disponível em alguma porta.

Módulo sockets - Python

Principais funções:

- `socket ()` - *para definir o protocolo a ser utilizado.*
- `bind ()` - *iniciando a escuta de conexões no lado do servidor.*
- `listen()` - *controle de conexões simultâneas que o servidor suportará*
- `accept()` - *aceita novas conexões*
- `connect()` - *para criar o objeto de conexão que irá conectar em um servidor*
- `send()` - *para enviar dados*
- `recv()` - *para receber dados controlando o tamanho do buffer.*
- `close()` - *fecha o socket*

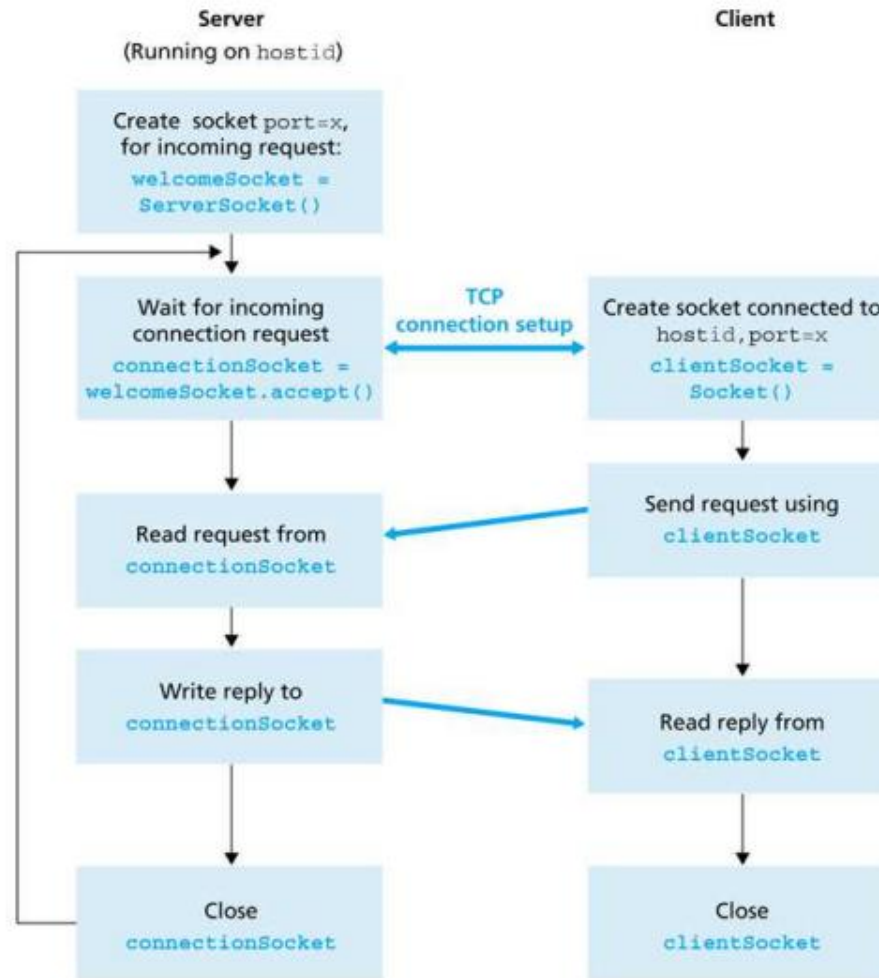
<https://docs.python.org/3/library/socket.html>

Módulo sockets - Python

Criamos um objeto do tipo socket usando o método ***socket.socket()***

- Família de Endereços
 - AF_INET (endereço IPv4)
 - AF_INET6 (endereço IPv6)
- Tipo de Socket
 - SOCK_STREAM (para socket TCP)
 - SOCK_DGRAM (para socket UDP)

Programação Socket com TCP



Servidor TCP - Python

```
import socket

HOST = 'localhost'
PORT = 5000
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind((HOST, PORT))
s.listen()
print ('Aguardando conexão de um cliente')
conn, ender = s.accept()
print ('Conectado em ', ender)
while True:
    data = conn.recv(1024)
    if not data:
        print ('Fechando a conexão')
        conn.close()
        break
    conn.sendall(data)
```

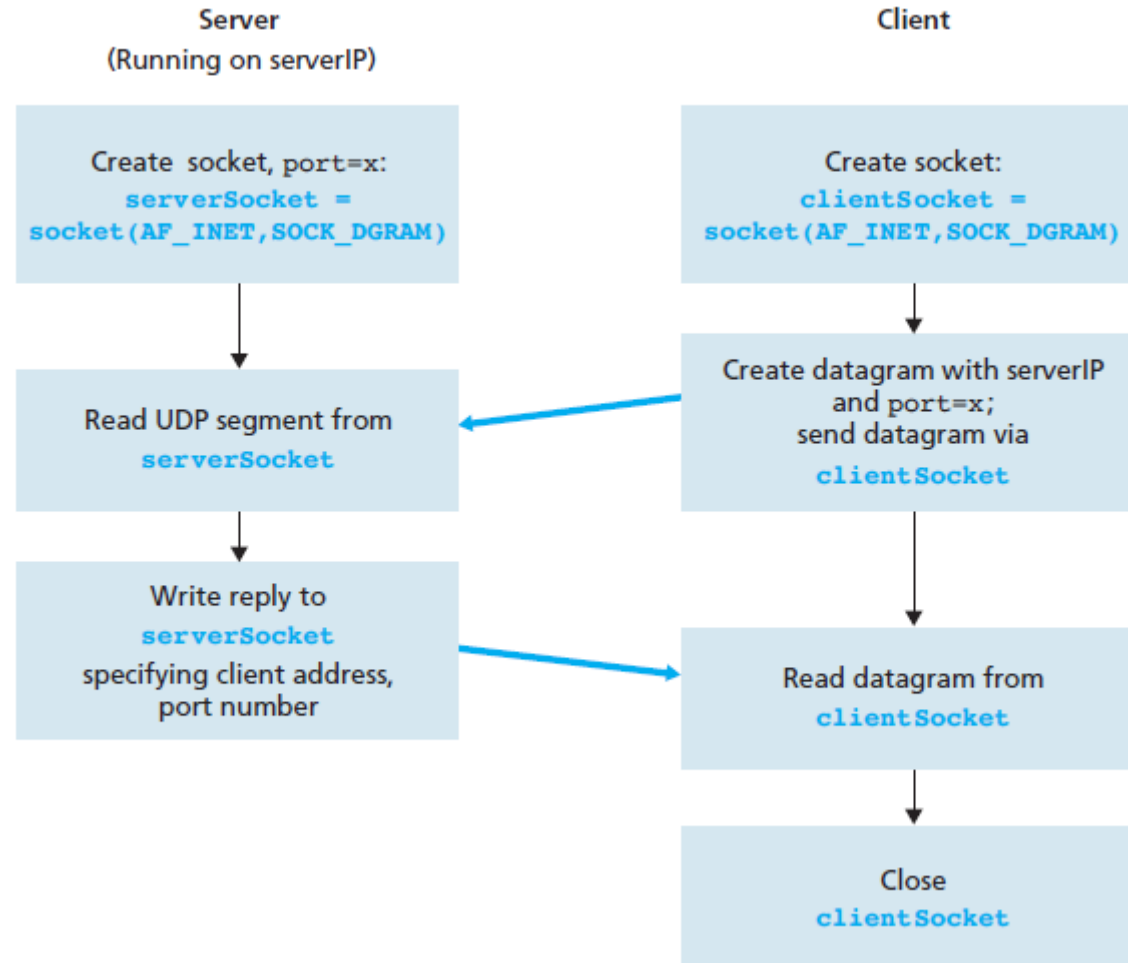
Cliente TCP - Python

```
import socket

HOST = '127.0.0.1'
PORT = 5000

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((HOST, PORT))
s.sendall(str.encode('Bom dia'))
data = s.recv(1024)
print('Mensagem ecoada:', data.decode())
```

Programação Socket com UDP



Referências

KUROSE, James F.; ROSS, Keith W. Redes de computadores e a internet: uma abordagem top-down. 5. ed. São Paulo: Pearson Addison Wesley, 2010. 614 p. ISBN 9788588639973.

Material disponibilizado de:

Prof. Rone Ilídio da Silva

Prof. Rodrigo Ronner T. da Silva