

UNIVERSIDADE CATÓLICA DE PETRÓPOLIS
CENTRO DE ENGENHARIA E COMPUTAÇÃO
TECNÓLOGO EM REDES DE COMPUTADORES

PRINCÍPIO DE COMUNICAÇÃO DE DADOS

PARTE 2 - AULA 01 – Sockets – Revisão de TCP/IP

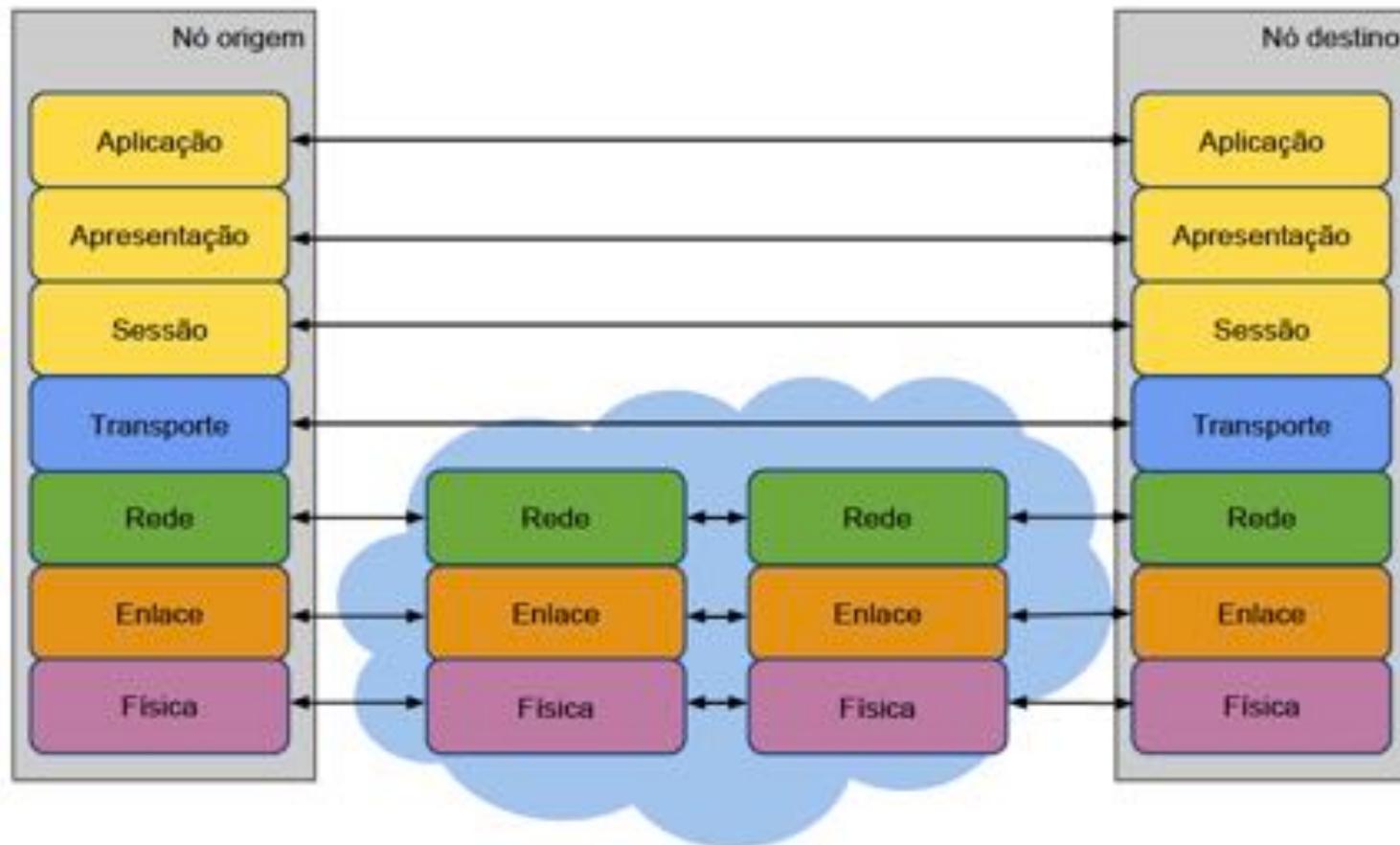
[1]

SOCKETS | Revisão

[1.1]

SOCKETS | Revisão

RM/OSI - Modelo de Referencia



[1.2]

SOCKETS | Revisão

TCP/IP – Modelo de Fato



Modelo de Referência OSI



Modelo de Referência TCP/IP



Pilha de Protocolos da Internet

[1.3]

SOCKETS | Revisão

Endereçamento IPv4

An IPv4 address (dotted-decimal notation)

172 . 16 . 254 . 1

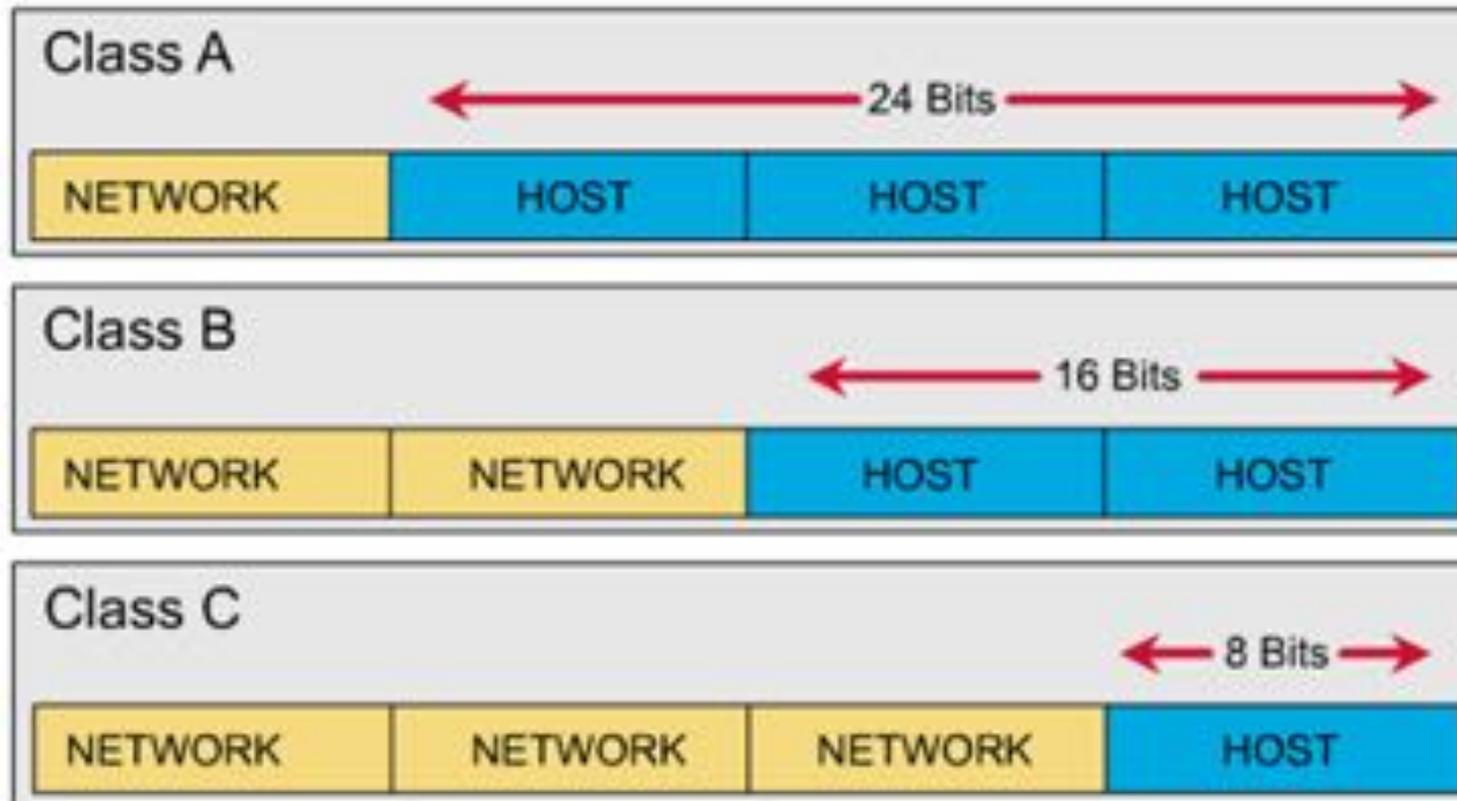


10101100 . 00010000 . 11111110 . 00000001

1 byte=8 bits

32 bits (4 x 8), or 4 bytes

Table 1





SOKECTS | Revisão de TCP/IPv4

Endereçamento IPv4

https://en.wikipedia.org/wiki/Reserved_IP_addresses

Address block	Address range	Number of addresses	Scope	Description
0.0.0.0/8	0.0.0.0–0.255.255.255	16 777 216	Software	Current network ^[25] (only valid as source address).
10.0.0.0/8	10.0.0.0–10.255.255.255	16 777 216	Private network	Used for local communications within a private network . ^[4]
100.64.0.0/10	100.64.0.0–100.127.255.255	4 194 304	Private network	Shared address space ^[25] for communications between a service provider and its subscribers when using a carrier-grade NAT .
127.0.0.0/8	127.0.0.0–127.255.255.255	16 777 216	Host	Used for loopback addresses to the local host. ^[2]
169.254.0.0/16	169.254.0.0–169.254.255.255	65 536	Subnet	Used for link-local addresses ^[26] between two hosts on a single link when no IP address is otherwise specified, such as would have normally been retrieved from a DHCP server.
172.16.0.0/12	172.16.0.0–172.31.255.255	1 048 576	Private network	Used for local communications within a private network. ^[4]
192.0.0.0/24	192.0.0.0–192.0.0.255	256	Private network	IETF Protocol Assignments. ^[3]
192.0.2.0/24	192.0.2.0–192.0.2.255	256	Documentation	Assigned as TEST-NET-1, documentation and examples. ^[7]
192.88.99.0/24	192.88.99.0–192.88.99.255	256	Internet	Reserved. ^[8] Formerly used for IPv6 to IPv4 relay ^[8] (included IPv6 address block 2002::/16).
192.168.0.0/16	192.168.0.0–192.168.255.255	65 536	Private network	Used for local communications within a private network. ^[4]
198.18.0.0/15	198.18.0.0–198.19.255.255	131 072	Private network	Used for benchmark testing of inter-network communications between two separate subnets. ^[10]
198.51.100.0/24	198.51.100.0–198.51.100.255	256	Documentation	Assigned as TEST-NET-2, documentation and examples. ^[7]
203.0.113.0/24	203.0.113.0–203.0.113.255	256	Documentation	Assigned as TEST-NET-3, documentation and examples. ^[7]
224.0.0.0/4	224.0.0.0–239.255.255.255	268 435 456	Internet	In use for IP multicast . ^[11] (Former Class D network).
240.0.0.0/4	240.0.0.0–255.255.255.254	268 435 455	Internet	Reserved for future use. ^[12] (Former Class E network).
255.255.255.255/32	255.255.255.255	1	Subnet	Reserved for the "limited broadcast" destination address. ^{[3][13]}

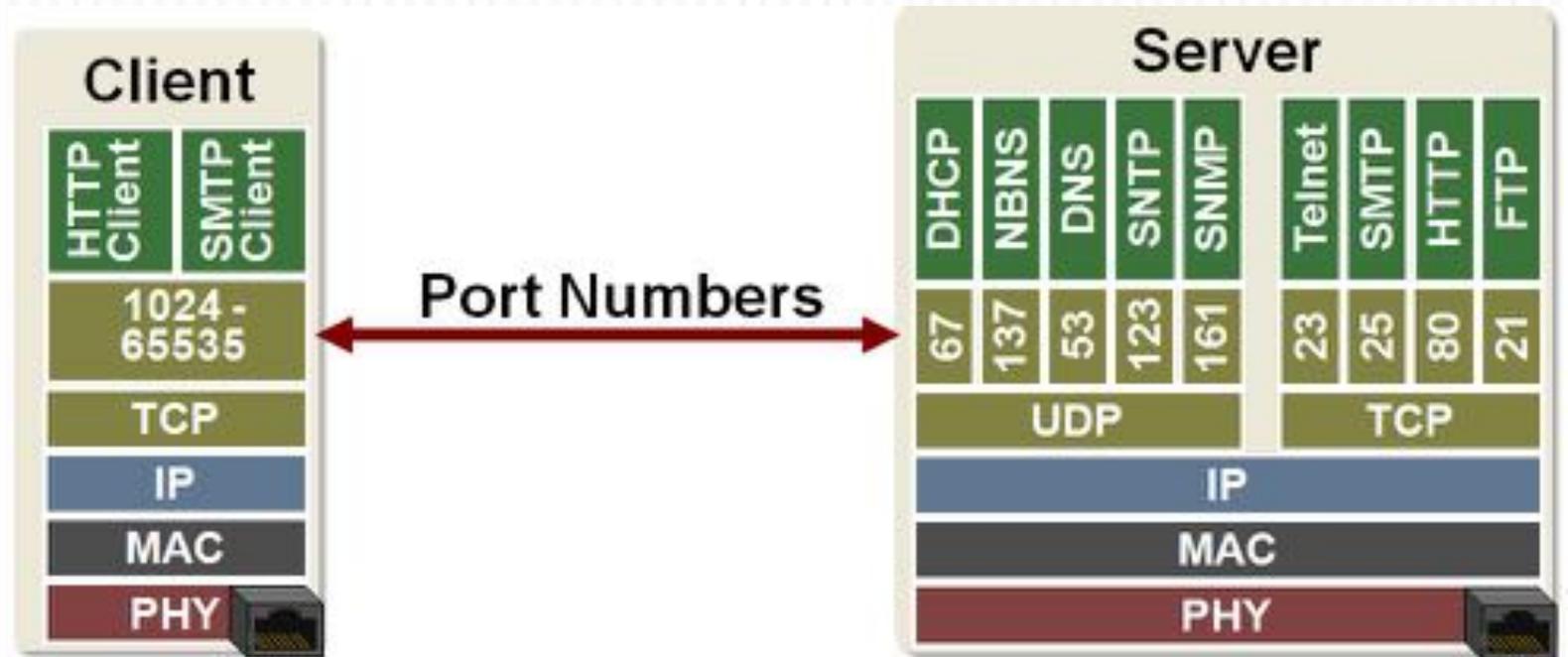
Nome	Faixa de endereços IP	Número de IPs	<i>classful</i> Descrição	Maior bloco CIDR	Referência
8-bit block	10.0.0.0 – 10.255.255.255	16,777,216	Uma classe A	10.0.0.0/8	RFC 1597 ↗ (obsoleto), RFC 1918 ↗
12-bit block	172.16.0.0 – 172.31.255.255	1,048,576	16 classes B	172.16.0.0/12	
16-bit block	192.168.0.0 – 192.168.255.255	65,536	256 classes C	192.168.0.0/16	
16-bit block	169.254.0.0 – 169.254.255.255	65,536	Uma classe B	169.254.0.0/16	RFC 3330 ↗ , RFC 3927 ↗

https://pt.wikipedia.org/wiki/Rede_privada

[1.4]

SOCKETS | Revisão

Portas



https://pt.wikipedia.org/wiki/Lista_de_portas_dos_protocolos_TCP_e_UDP

- Portas são endereços providos pela camada de transporte
- Podem estar associadas aos protocolos TCP e UDP
- Elas determinam para onde os fluxos devem ser enviados.
- Elas permitem que um hosts com um único endereço IP hospede vários serviços
- Cada porta identifica um serviços distinto

https://pt.wikipedia.org/wiki/Lista_de_portas_dos_protocolos_TCP_e_UDP

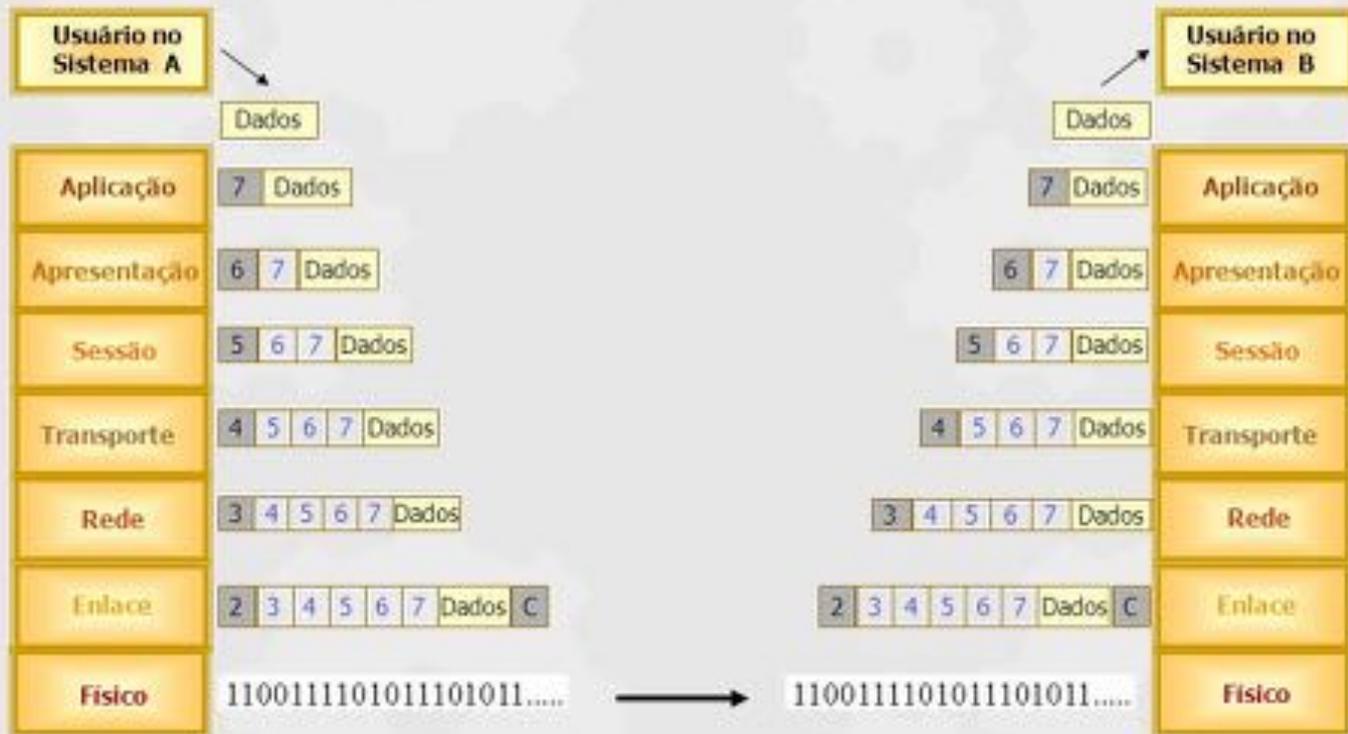
- Em cada host há 65535 portas disponíveis.
- O uso das portas é regulado pelo Internet Corporation for Assigning Names and Numbers (**ICANN**).
- Elas são divididas em três categorias:
 - **0 à 1023** → **well known ports**, associada a serviços e protocolos conhecidos.
 - **1024 à 49151** → **registered ports**, associadas a serviços registrados no ICANN.
 - **49152 à 65 535** → **dynamic (private, high)**, são utilizadas quando uma seção é estabelecida e liberadas quando uma seção é finalizada; Podem ser utilizadas por aplicações e serviços que estão em teste.

[1.5]

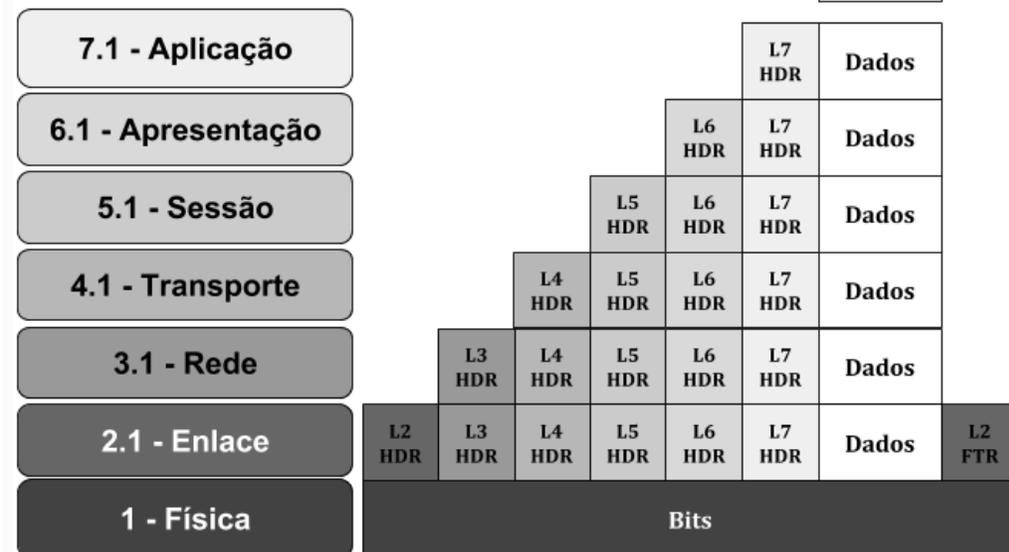
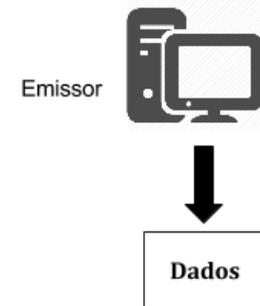
SOCKETS | Revisão

Encapsulamento de dados

Modelo OSI - Processo de encapsulamento

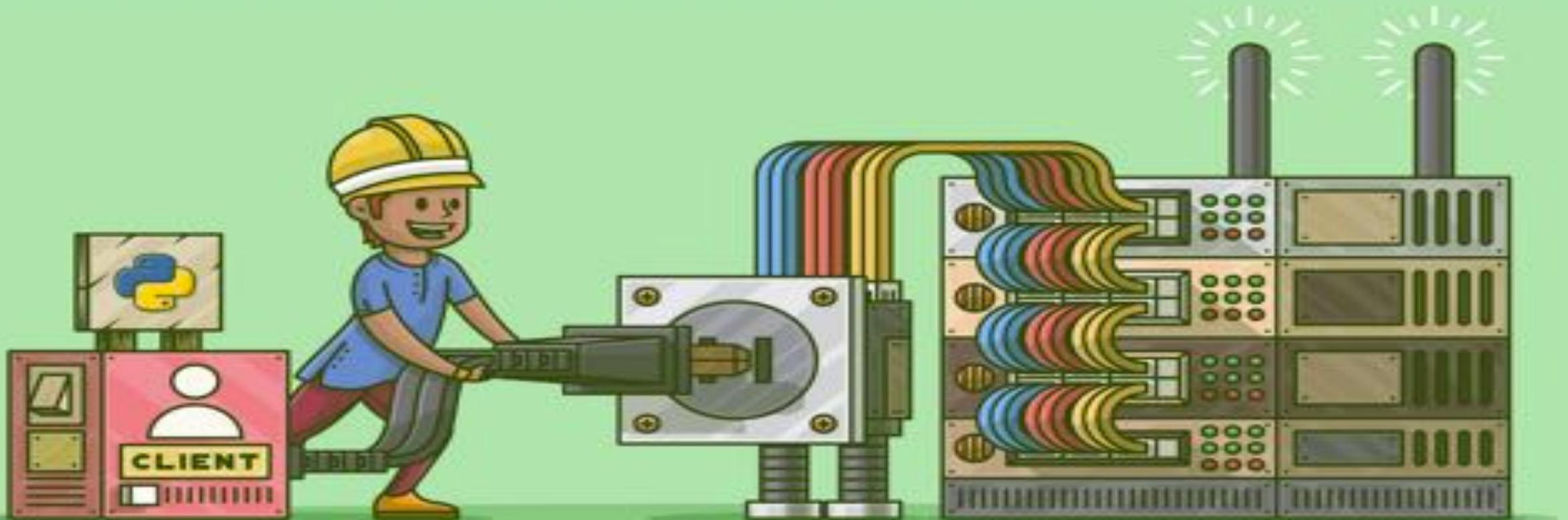


HDR = Header ou Cabeçalho
 L = Layer ou Camada
 FTR = Footer ou Rodapé



[2]

SOCKETS | Conceitos



[2.1]

SOCKETS | Conceitos

Conceitos Básicos

Sockets são os
“endpoints” de uma
comunicação bilateral

- São os “endpoints” de uma comunicação bilateral;
- Permitem a comunicação entre processos (IPC)
 - ▣ Entre processos da mesma máquina
 - ▣ Entre processos em máquina distintas
- Eles permitem a troca de dados e o acesso a recursos remotos

- Implementa o processo de IPC de mais baixo nível.
- Permitindo a interconexão entre vários elementos (softwares)
- Conceito semelhantes ao sistema telefônico
- Permitem a conexão com um ou múltiplos elementos

- **Domínio** é o “espaço” no qual o endereço é especificado.
 - ▣ **INTERNET - AF_INET** - os endereços consistem do end. de rede da máquina e da identificação do no. da porta, o que permite a comunicação entre processos de sistemas diferentes
 - ▣ **Unix: AF_UNIX** - os processos se comunicam referenciando um pathname, dentro do espaço de nomes do sistema de arquivos

[2.2]

SOCKETS | Conceitos

AF_INET | Internet Sockets

- ❑ Os Sockets criados pelos programas devem ser nomeados de forma que possam ser identificados.
- ❑ Esses nomes geralmente são traduzidos em endereços
- ❑ E um endereço é especificado por um domínio

- Aplicações podem utilizar os sockets para utilizar as funcionalidades da pilha TCP/IP
- Neste domínio (AF_INET) os sockets possuem dois níveis de endereços:
 - Local → Endereço da porta (0 à 65535)
 - Remoto → End. IPv4 ou End. IPv6
- Geralmente utilizado para implementar uma arquitetura Cliente X Servidor

- Cliente X Servidor
 - O Servidor fica escutando o socket aguardando por uma solicitação do cliente
 - O Cliente precisa conhecer o endereço do servidor e a porta na qual o servidor aguarda pela conexão.
 - Ao receber uma solicitação o Servidor cria um novo socket e associa a conexão que está sendo estabelecida.

- Cliente X Servidor
 - O Servidor deve utilizar uma porta cujo valor seja conhecido, nesta ele deve aguardar pelas conexões;
 - Ao aceitar uma nova conexão o servidor cria um outro socket e o associa a uma nova porta, alta, geralmente acima de 1024
 - Volta a aguardar por conexões na porta original

- Protocolos Suportados
 - Camada de Transporte
 - TCP → stream_sockets
 - UDP → datagram_sockets
 - Camada de Rede
 - raw_sockets

[2.2.1]

SOCKETS | Conceitos | AF_INET

RAW Sockets

Fluxos bidirecionais sem
garantia de entrega
(não confiáveis)

raw_socket

- Acesso a interface de protocolos de rede.
- A aplicação pode acessar diretamente protocolos de comunicação de baixo nível.
- Permite a construção de novos protocolos sobre os protocolos de baixo nível já existentes
- Normalmente orientados a datagrama

[2.2.2]

SOCKETS | Conceitos | AF_INET

User Datagram Protocol (**UDP**) Sockets

Fluxos bidirecionais sem
garantia de entrega
(não confiáveis)

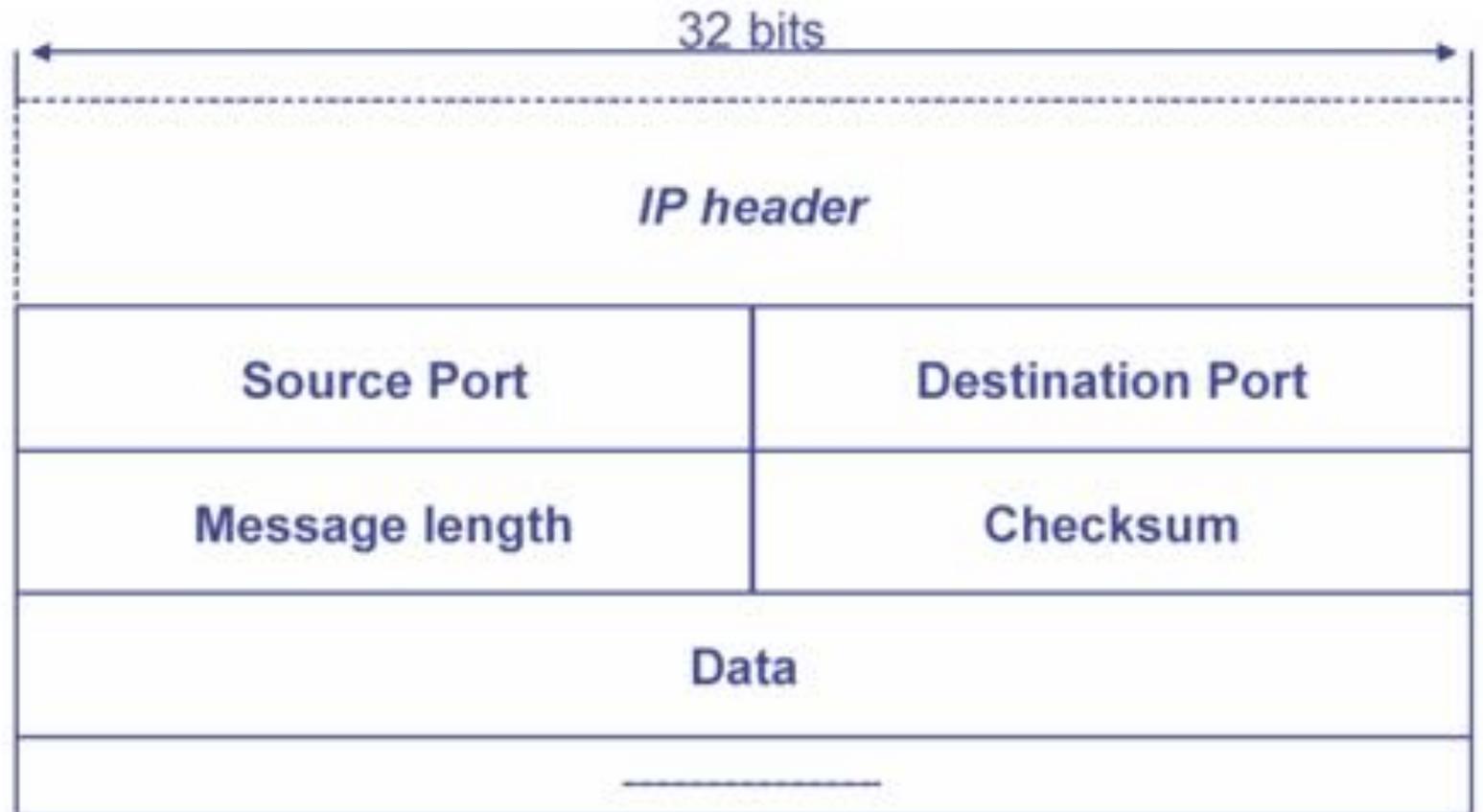
datagram_sockets (UDP):

- Fluxo de dados bidirecional
- Não oferece um serviço confiável → Protocolo não orientado a conexão
- Mensagens duplicadas, perdidas, e em ordem diferente podem ocorrer

datagram_sockets (UDP):

- Sem conexão
- Mensagens de tamanho fixo (datagramas) são transmitidas individualmente para destinações especificadas.
- Sem garantia de entrega.

datagram_sockets (UDP):



[2.2.3]

SOCKETS | Conceitos | AF_INET

Transmission Control Protocol (**TCP**) Sockets

Fluxos bidirecionais com
garantia de entrega
(confiáveis)

stream_sockets (TCP):

- Sequenciamento e fluxo bidirecional.
- Transmissão sobre um base confiável e com capacidade de transmissão de dados expressos.
- No domínio UNIX, o SOCKET_STREAM trabalha igual a um pipe.
- No domínio INTERNET este tipo de socket é implementado sobre TCP/IP.

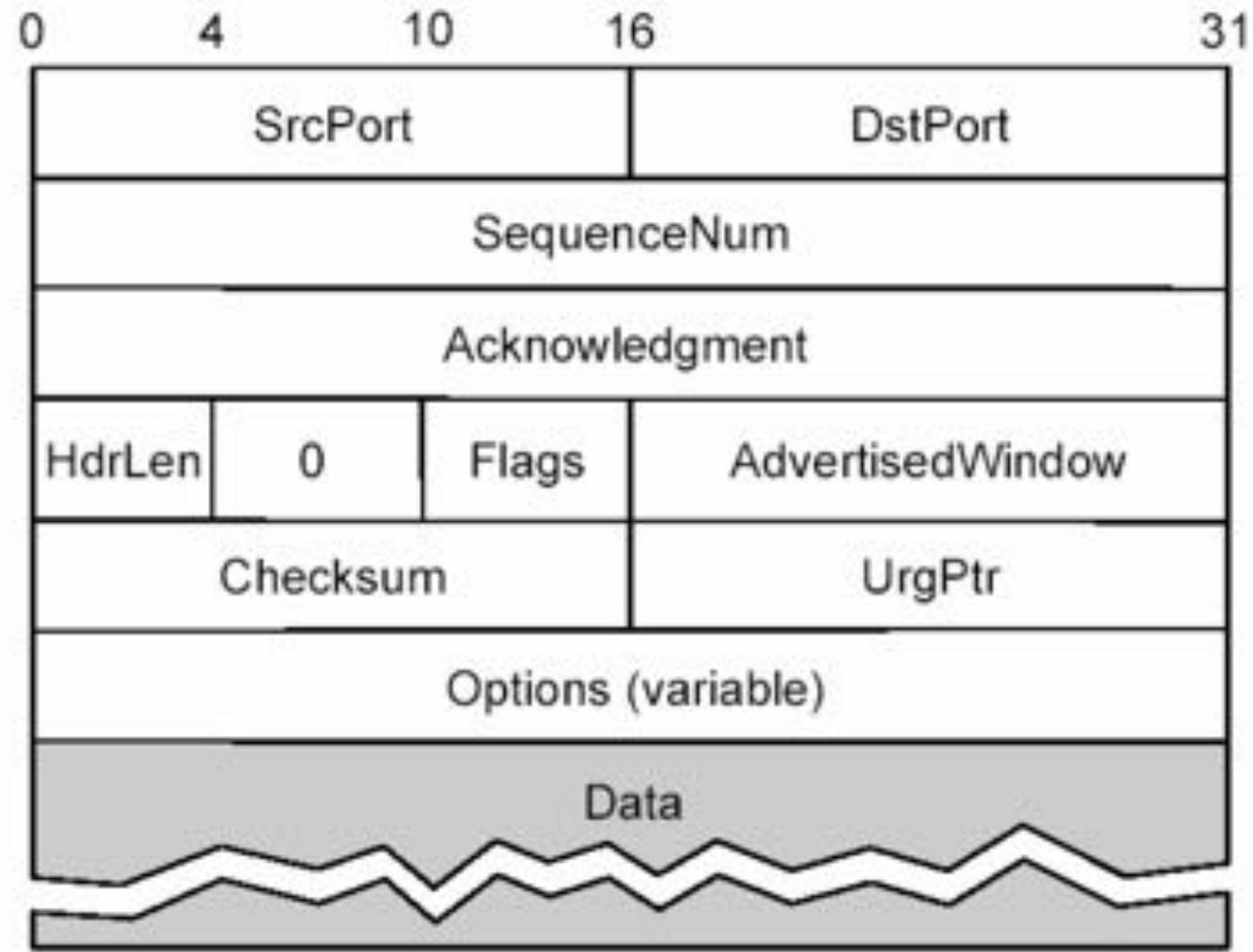
stream_sockets (TCP):

- Baseado no TCP (Transmission Control Protocol)
- Orientado a conexão
- Fases:
 - Fase de conexão
 - Fase de transmissão
 - Fase de desconexão

stream_sockets (TCP):

- Orientado a conexão
- Conexão lógica é explicitamente estabelecida entre dois processos que se comunicam.
- A camada de transporte se encarrega de transferir a sequência de forma correta e confiável. Garantia de entrega.

stream_sockets (TCP):



[2.3]

SOCKETS | Conceitos

Modelo Cliente - Servidor

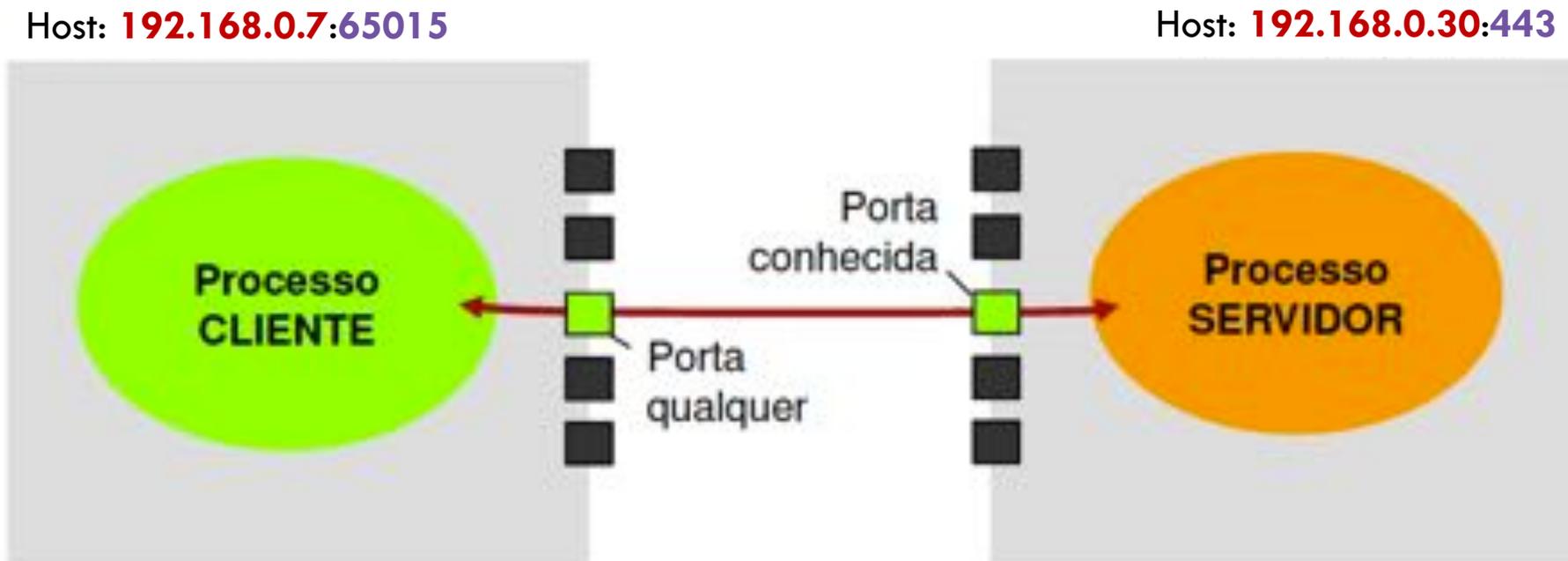
Servidor é o elemento passivo que aguarda pelas conexões.

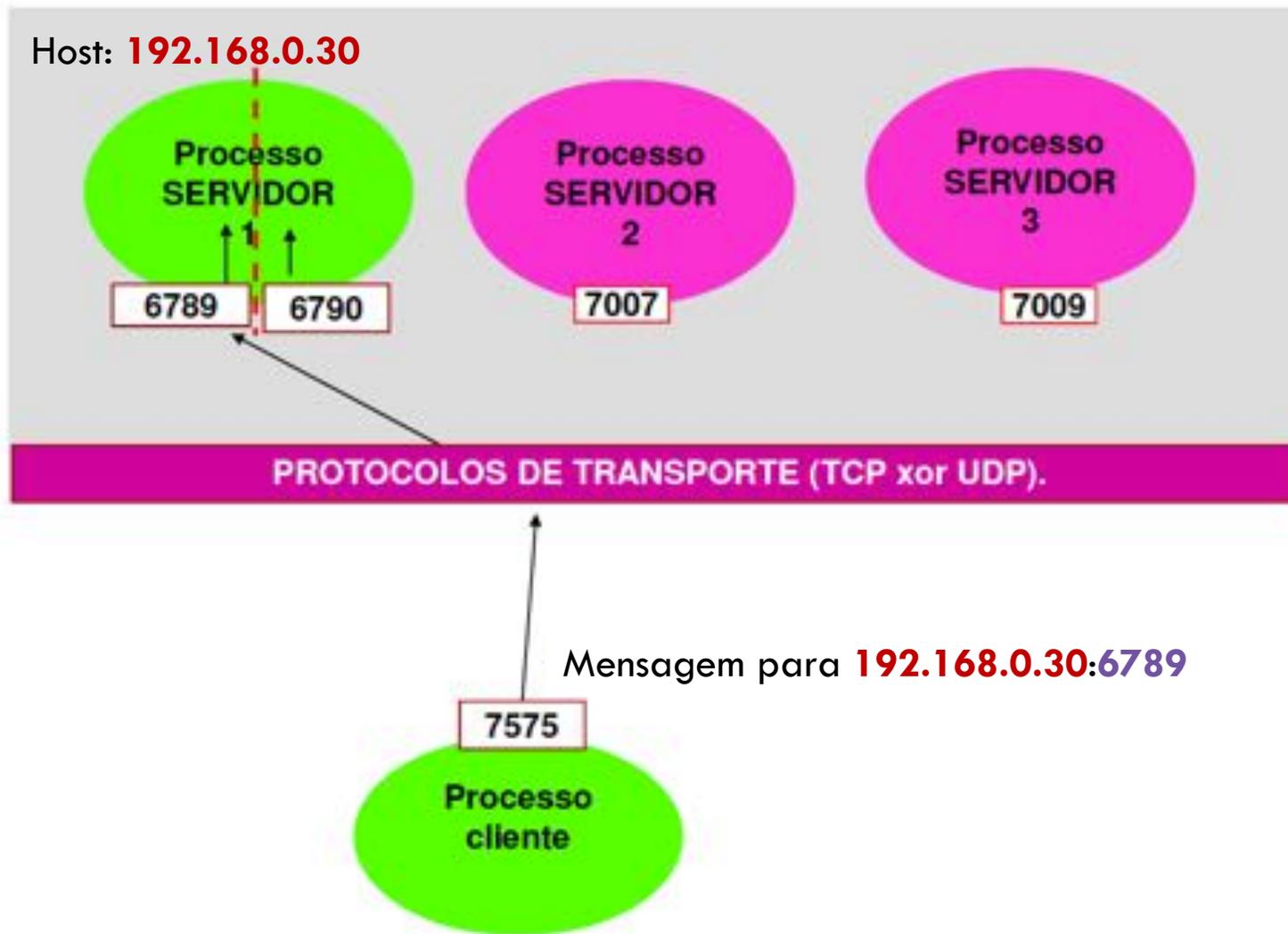
□ Pilha de Protocolos



- Comunicação baseada em Sockets

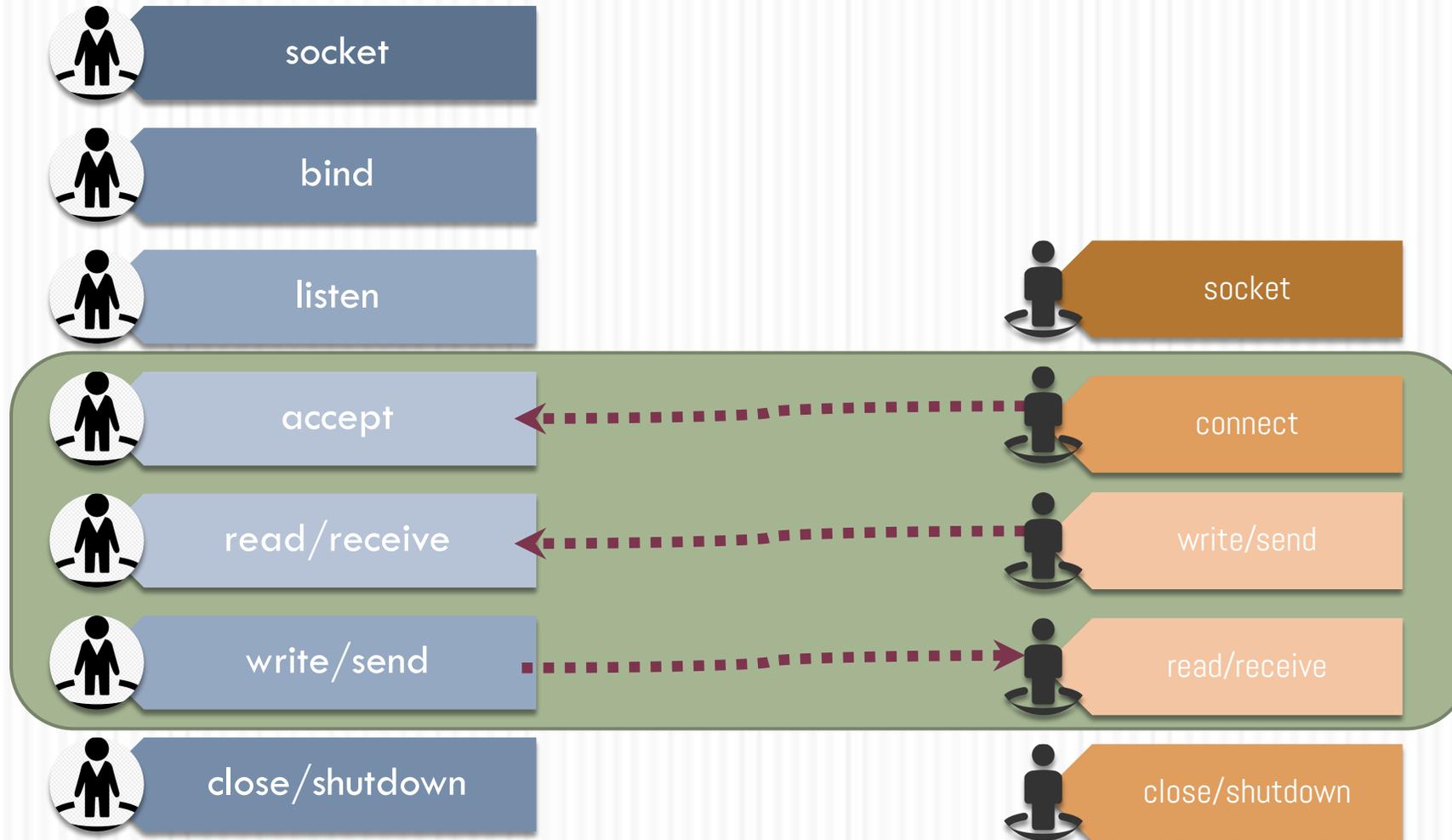
Sockets = (End. IPv4, Porta)





SOCKETS | Conceitos

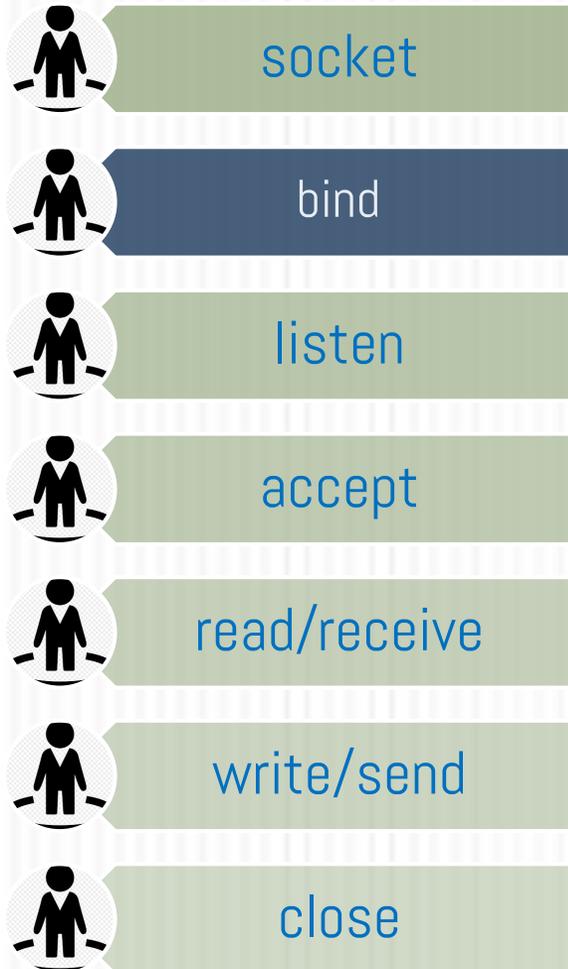
Modelo
Cliente - Servidor





socket () :

- Cria um novo descritor
- Define o tipo da conexão (Domínio e Protocolo)
- Será utilizado como referencia pelas demais funções



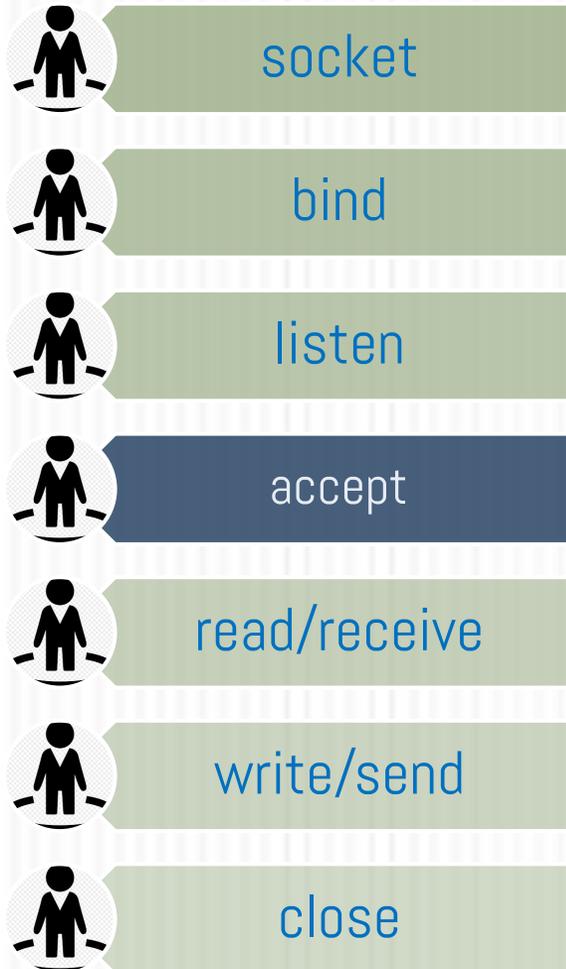
bind () :

- Função exclusiva do servidor
- Associa o socket a determinando endereço



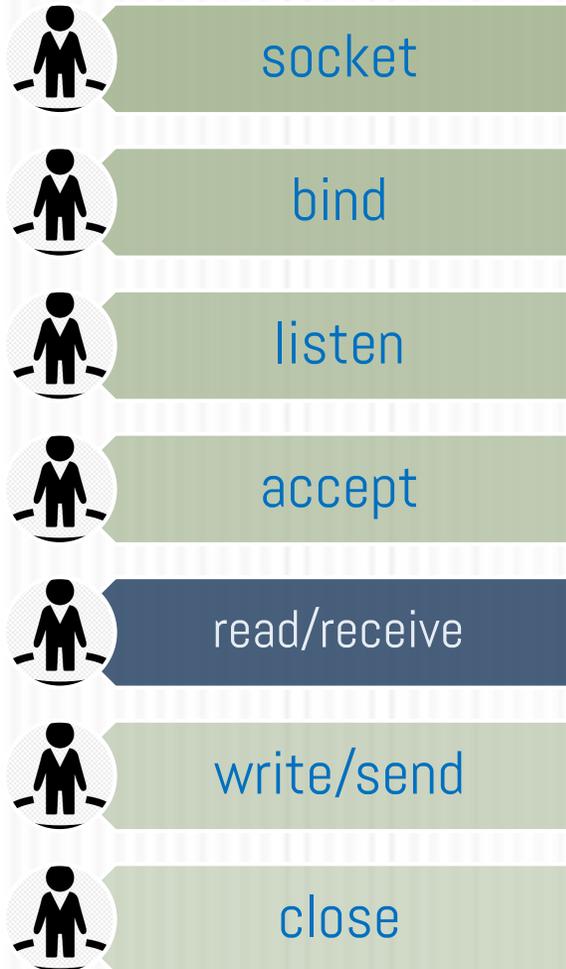
listen () :

- coloca o socket um modo de espera
- socket passa a escutar/aguardar conexões



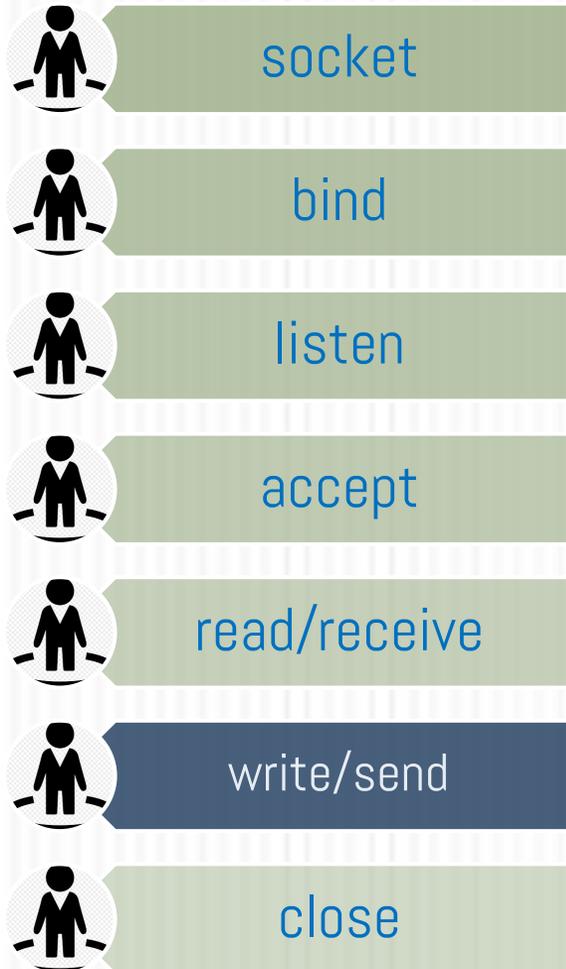
accept () :

- recebe uma das conexões que estavam na fila
- cria um novo socket para tratar a nova conexão
- deixa o socket original aguardando por novas conexões



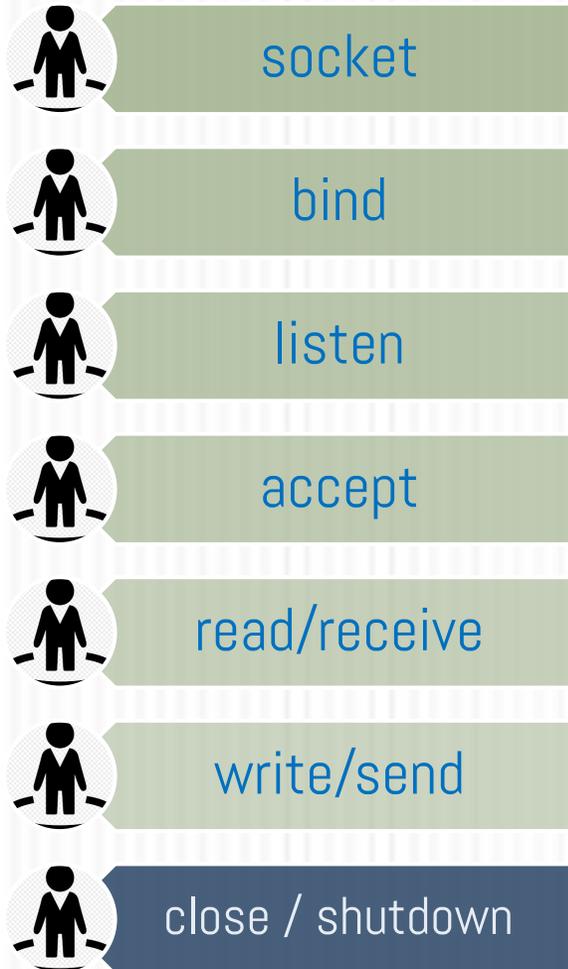
read () - receive ():

- Lê o conteúdo do buffer associado ao socket;



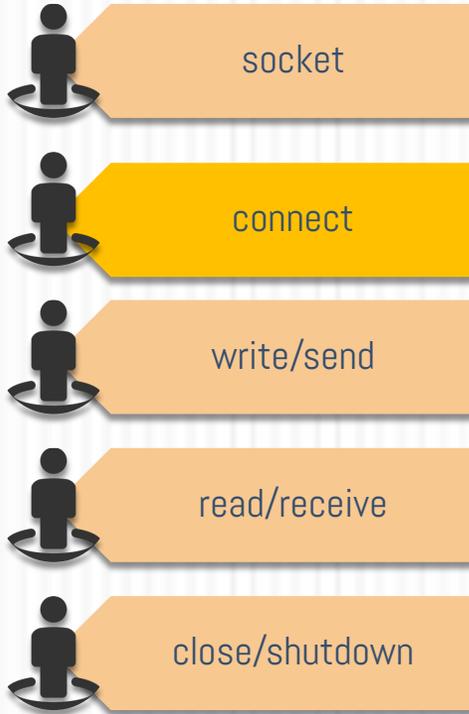
write () – send () :

- Escreve dados em um buffer associado ao Socket



close () – shutdown () :

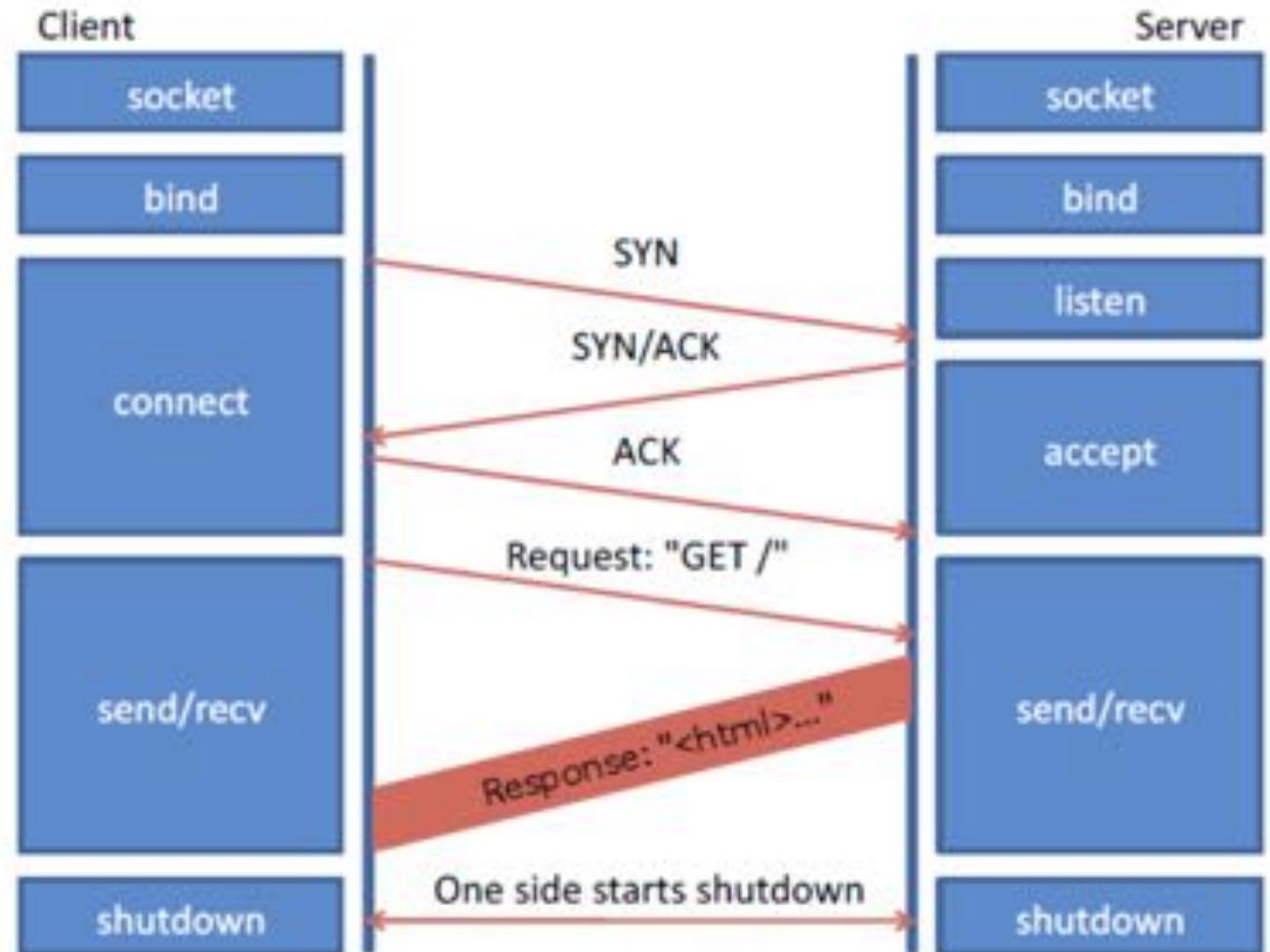
- o fecha o sockets ()



connect () :

- estabelece uma nova conexão com um servidor;

□ Interação cliente-servidor



UNIVERSIDADE CATÓLICA DE PETRÓPOLIS
CENTRO DE ENGENHARIA E COMPUTAÇÃO
TECNÓLOGO EM REDES DE COMPUTADORES

PRINCÍPIO DE COMUNICAÇÃO DE DADOS

PARTE 2 - AULA 01 – Sockets – Revisão de TCP/IP