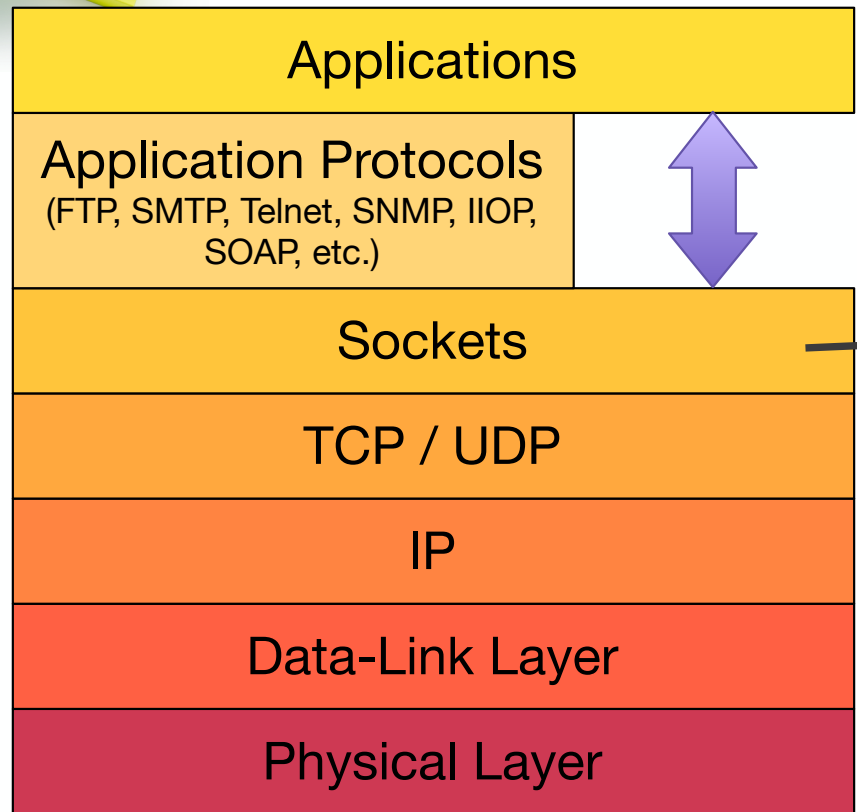




# Sockets



Conjunto de APIs mais difundido para programação sobre a arquitetura TCP/IP.



- ◉ Desenvolvido para o UNIX versão **Berkeley**
- ◉ **Método padrão** para conectar duas máquinas numa rede:
  - Padrão atual da Internet
- ◉ Oferece mecanismos para manipular o **fluxo bidirecional** de dados em uma conexão de rede virtual.

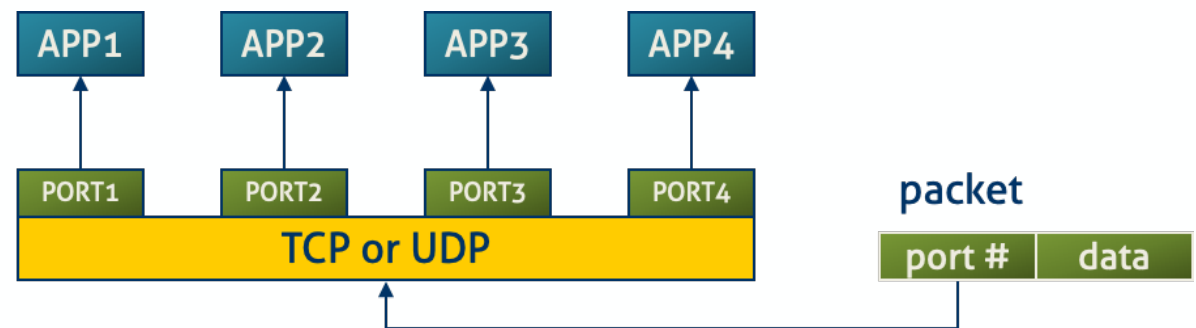


- Suportam serviços para:
  - protocolos orientados a conexão (**TCP**)
  - protocolos não-orientados a conexão (**UDP**)
- Usam o paradigma:
  - `open / read-write / close`
- para entrada e saída (típico de sistemas *UNIX-like*).



# Portas

- Tipicamente:
  - O número de processos com comunicação na rede em uma máquina excede o número de conexões físicas da máquina com a rede.
- **Portas:**
  - para enviar dados a aplicações (processos) específicas rodando no computador.





## Princípio de Sockets

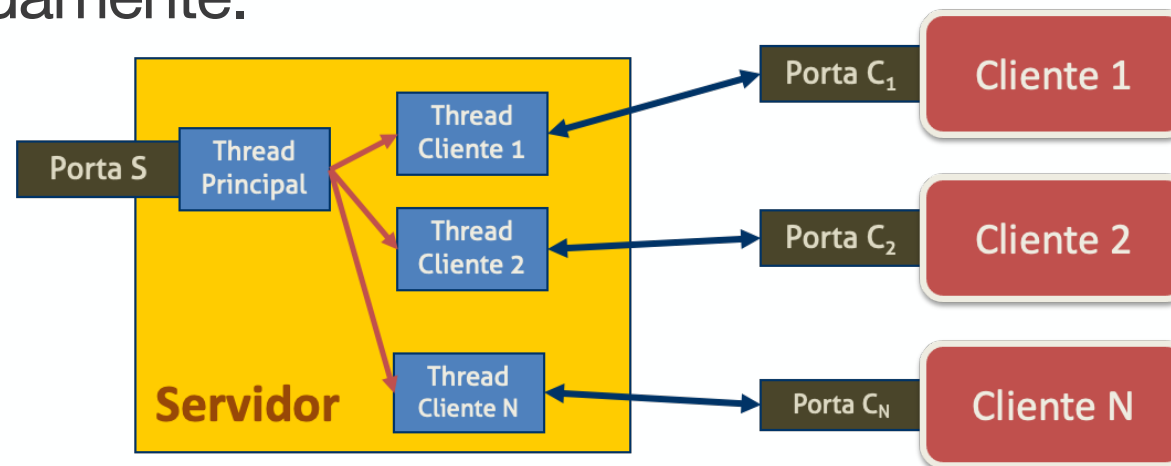
- Existe uma porta tanto do lado do cliente como do lado do servidor.





## Mais de um cliente

- O servidor pode utilizar a **mesma porta** para estabelecer várias conexões.
- Cada cliente deve ser atendido por um *thread* separadamente.





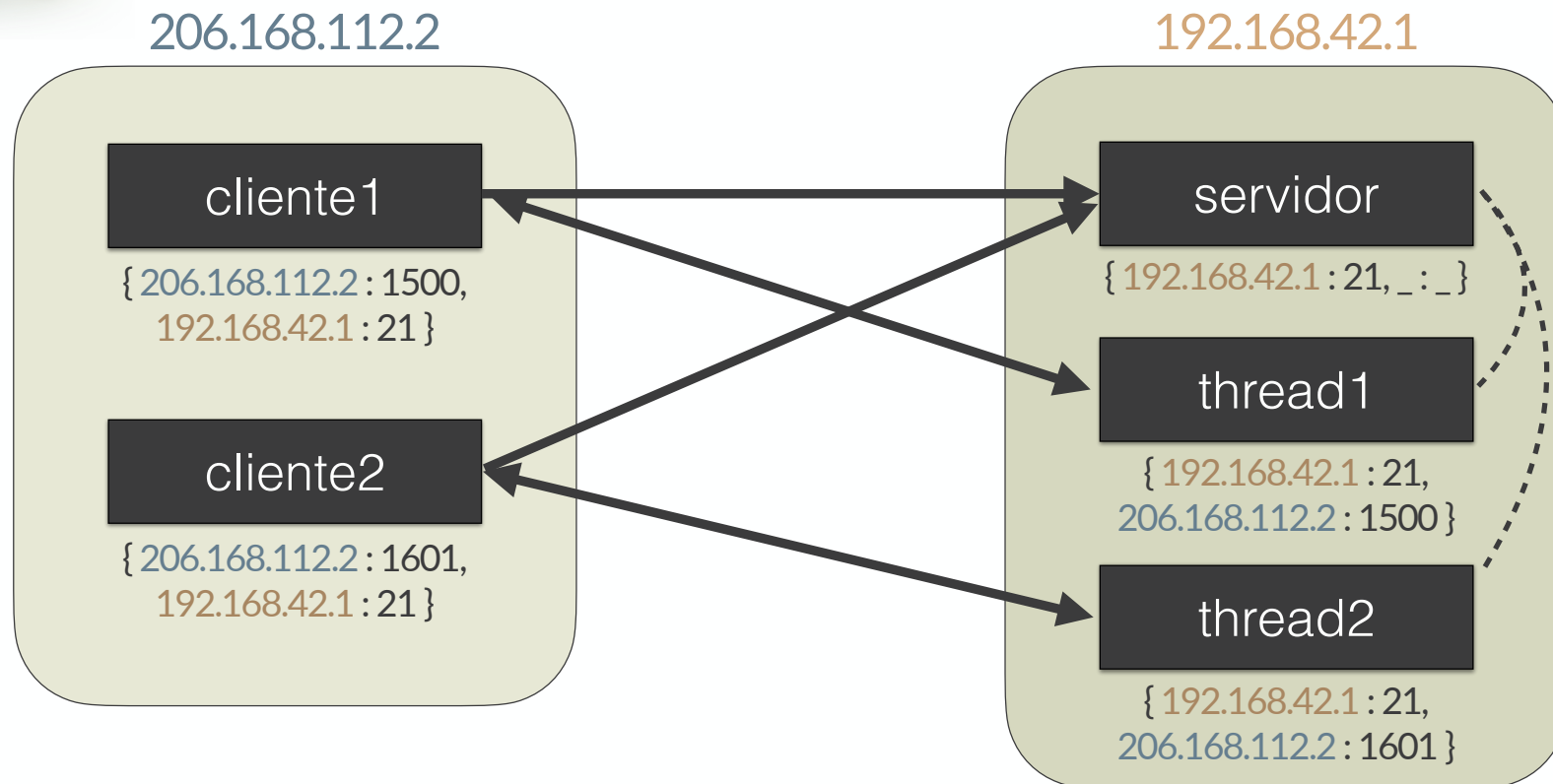
- **Par de sockets** em conexão TCP é formado por uma **quatro-tupla**:

{ IP Local:Porta local, IP Remoto:Porta Remota }

*Par de sockets identifica a conexão de forma única na rede.*



## Portas e Sockets

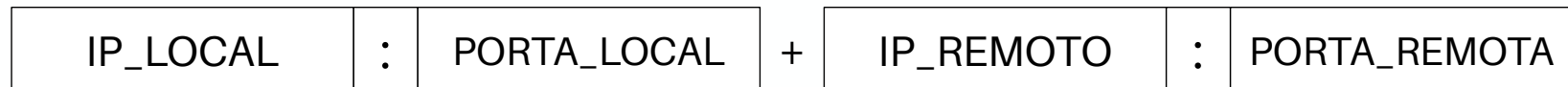






## Multiplexação TCP

- **Não somente** por número de **porta destino**.
- **Cada elemento do par de sockets** é examinado para determinar qual extremidade receberá os dados.





## Convenções de Portas

- ◉ Portas padrão de **servidor** têm números  $< 1024$
- ◉ Portas de **cliente** tem número  $\geq 1024$
- ◉ Até **64K** portas são possíveis em um servidor
- ◉ Portas servidoras são **padronizadas** por RFCs:
  - 22/TCP: SSH
  - 25/TCP: SMTP (e-mail)
  - 80/TCP: HTTP
  - 443/TCP: HTTPS
  - 161/UDP: SNMP (gerência de rede, ping...)