Programação para redes

Live de Python # 229

Roteiro

1. O mínimo necessário sobre redes

Uma "revisão introdutória"

2. Scapy

Conhecendo a biblioteca

3. Camadas e Protocolos

Botando os scapy para funcionar

4. Sinffing de pacotes

Analisando trafego

picpay.me/dunossauro

apoia.se/livedepython

pix.dunossauro@gmail.com

Ajude o projeto <3

Ademar Peixoto, Adilson Herculano, Adriana Cavalcanti, Adriano Ferraz, Alexandre Harano, Alexandre Lima, Alexandre Takahashi, Alexandre Villares, Alex Lima, Allan Almeida, Alynne Ferreira, Alysson Oliveira, Ana Carneiro, Andre Azevedo, André Rafael, Aquiles Coutinho, Arnaldo Turque, Aurelio Costa, Bruno Batista, Bruno Divino, Bruno Freitas, Bruno Guizi, Bruno Lopes, Bruno Ramos, Caio Felix, Caio Nascimento, Carina Pereira, Christiano Morais, Clara Battesini, Dandara Sousa, Daniel Freitas, Daniel Haas, Daniel Santos, Danilo Segura, David Couto, David Kwast, Delton Porfiro, Denis Quirino, Diego Farias, Diego Guimarães, Dilenon Delfino, Dino Aguilar, Diogo Paschoal, Douglas Bastos, Douglas Zickuhr, Eduardo Tolmasquim, Emerson Rafael, Eneas Teles, Erick Ritir, Érico Andrei, Eugenio Mazzini, Euripedes Borges, Everton Silva, Fabiano Tomita, Fabio Barros, Fábio Barros, Fabio Castro, Fábio Thomaz, Fabricio, Fabricio Araujo, Felipe Rodrigues, Fernanda Prado, Fernando Florëncio, Firehouse, Flávio Meira, Flavkaze, Gabriel Barbosa, Gabriel Mizuno, Gabriel Nascimento, Gabriel Simonetto, Geandreson Costa, Guilherme Felitti, Guilherme Gall, Guilherme Ostrock, Guilherme Piccioni, Guilherme Silva, Gustavo Suto, Harold Gautschi, Heitor Fernandes, Helvio Rezende, Henrique Junqueira, Hugo Cosme, Igor Taconi, Ismael Ventura, Italo Silva, Izac Silva, Jairo Jesus, Jairo Lenfers, Janael Pinheiro, João Paulo, Joelson Sartori, Johnny Tardin, Jônatas Silva, José Barbosa, José Gomes, Joseíto Júnior, Jose Mazolini, José Pedro, Juan Gutierrez, Juliana Machado, Julio Franco, Júlio Gazeta, Júlio Pereira, Julio Silva, Kaio Peixoto, Kaneson Alves, Leandro Miranda, Lengo, Leonardo Mello, Leonardo Nazareth, Leon Solon, L. Perciliano, Luã, Luancomputacao Roger, Lucas Adorno, Lucas Carderelli, Lucas Mello, Lucas Mendes, Lucas Nascimento, Lucas Schneider, Lucas Simon, Lucas Valino, Luciano Filho, Luciano Ratamero, Luciano Silva, Luciano Teixeira, Luiz Junior, Luiz Lima, Luiz Paula, Maicon Pantoja, Maiquel Leonel, Marcelino Pinheiro, Márcio Martignoni, Marcio Moises, Marco Mello, Marcos Gomes, Marco Yamada, Maria Clara, Maria Gabriela, Marina Passos, Mateus Lisboa, Matheus Cortezi, Matheus Oliveira, Matheus Silva, Matheus Vian, Mauricio Fagundes, Mauricio Nunes, Mírian Batista, Mlevi Lsantos, Murilo Andrade, Murilocunha, Murilo Viana, Nando Sangenetto, Natan Cervinski, Nathan Branco, Nicolas Teodosio, Osvaldo Neto, Otávio Carneiro, Patricia Minamizawa, Patrick Felipe, Paulo D., Paulo Tadei, Pedro Henrique, Pedro Pereira, Pedro Silva, Peterson Santos, Priscila Santos, Rafael Lopes, Rafael Romão, Ramayana Menezes, Regis Santos, Regis Tomkiel, Rene Bastos, Ricardo Silva, Ricarte Jr, Riverfount, Rjribeiro, Robson, Robson Maciel, Rodrigo Alves, Rodrigo Cardoso, Rodrigo Freire, Rodrigo Messias, Rodrigo Quiles, Rodrigo Ribeiro, Rodrigo Vaccari, Rodrigo Vieira, Rogério Lima, Rogério Nogueira, Rogério Sousa, Ronaldo Silva, Ronaldo Silveira, Rudiney Pereira, Rui Jr, Samanta Cicilia, Sebastião Tolentino, Stash, Talita Rossari, Tay Turnner, Thaynara Pinto, Thi, Thiago Araujo, Thiago Borges, Thiago Curvelo, Thiago Moraes, Thiago Souza, Tiago Minuzzi, Tiago Souza, Tony Dias, Tony Santos, Tyrone Damasceno, Uadson Emile, Valcilon Silva, Valdir Tegon, Vcwild, Vinicius Stein, Vitor Luz, Vladimir Lemos, Walter Reis, Wesley Mendes, Willian Lopes, Wilson Duarte, Wilson Neto, Wilson Rocha, Xico Silvério, Yury Barros

Obrigado você

Uma "revisão introdutória"

Redes

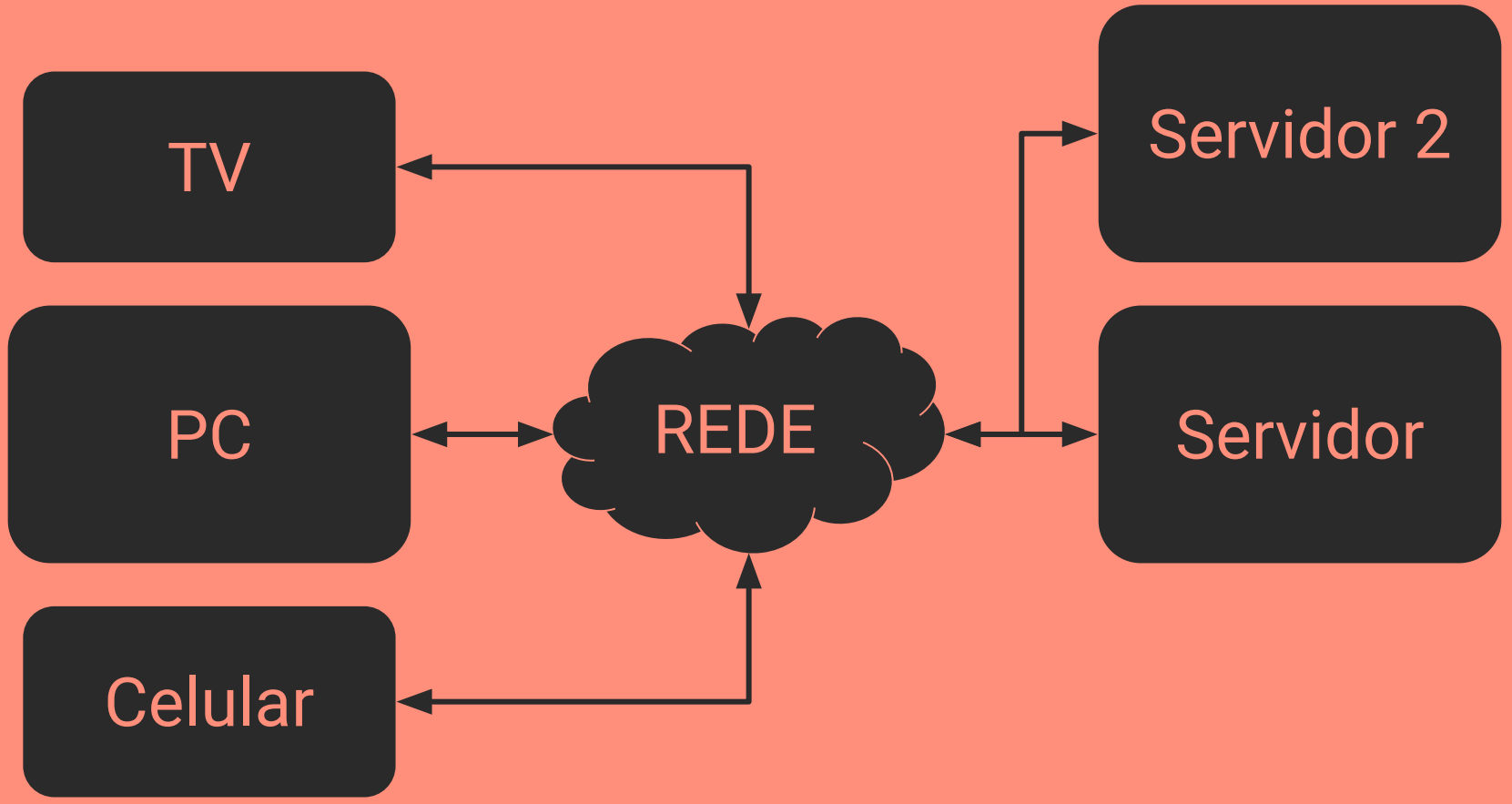
O básico necessário

Rede de computadores é um conjunto de dois ou mais dispositivos interligados por um sistema de comunicação digital que seguem algumas regras e especificações para compartilhar recursos entre si.

O básico necessário

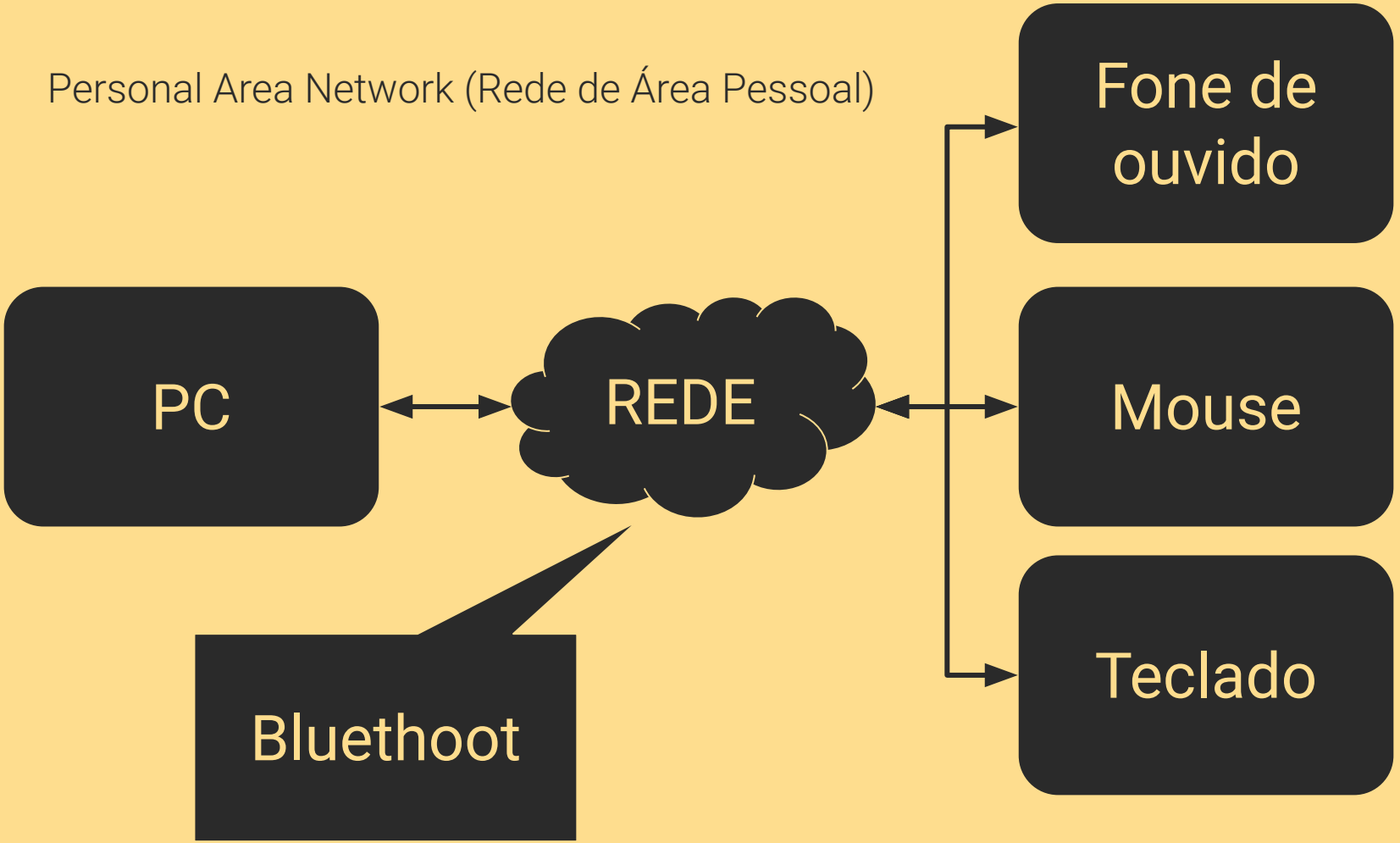
Rede de computadores é um conjunto de dois ou mais dispositivos interligados por um sistema de comunicação digital que seguem algumas regras e especificações para compartilhar recursos entre si.

O básico necessário



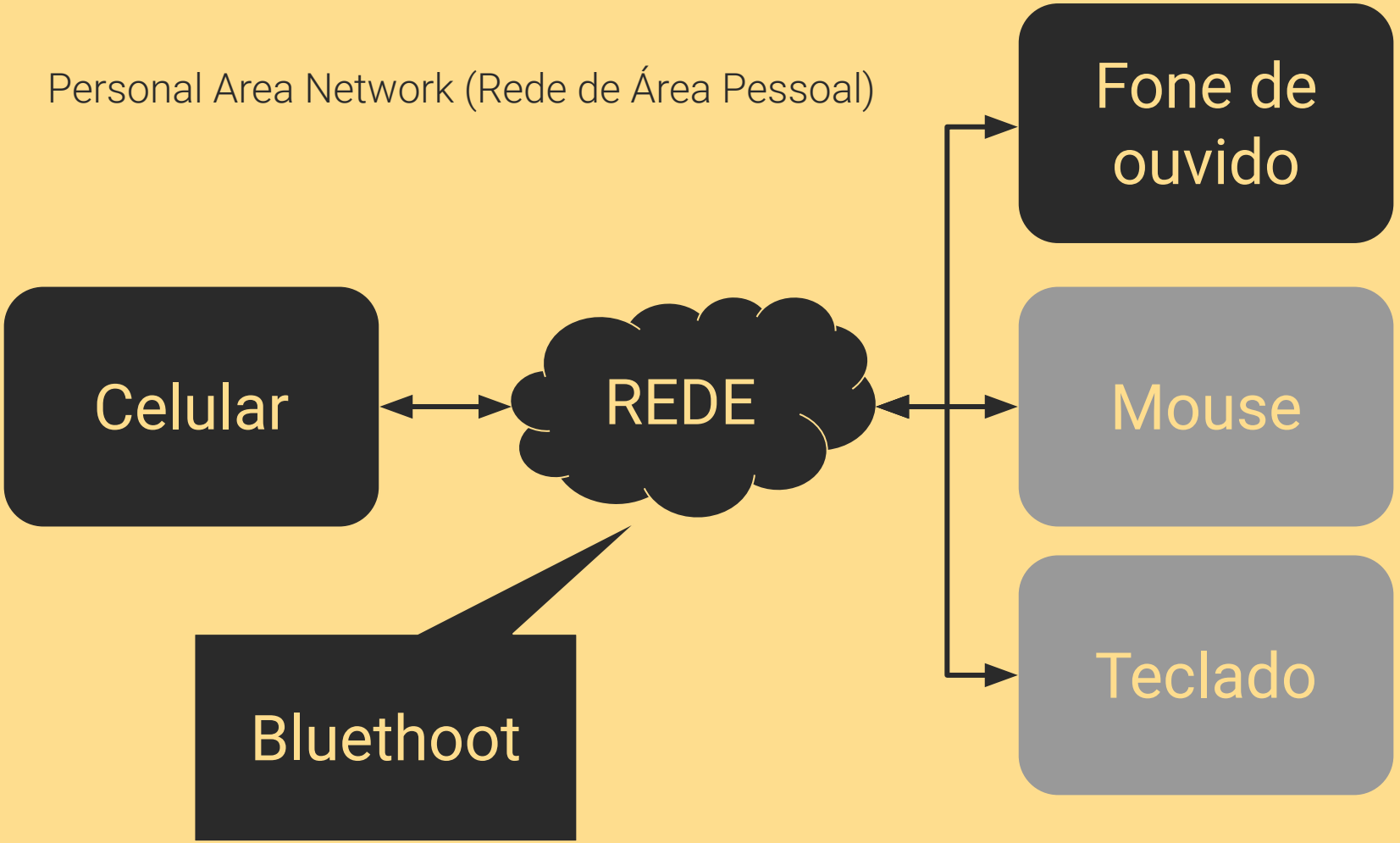
Tipos de redes — PAN

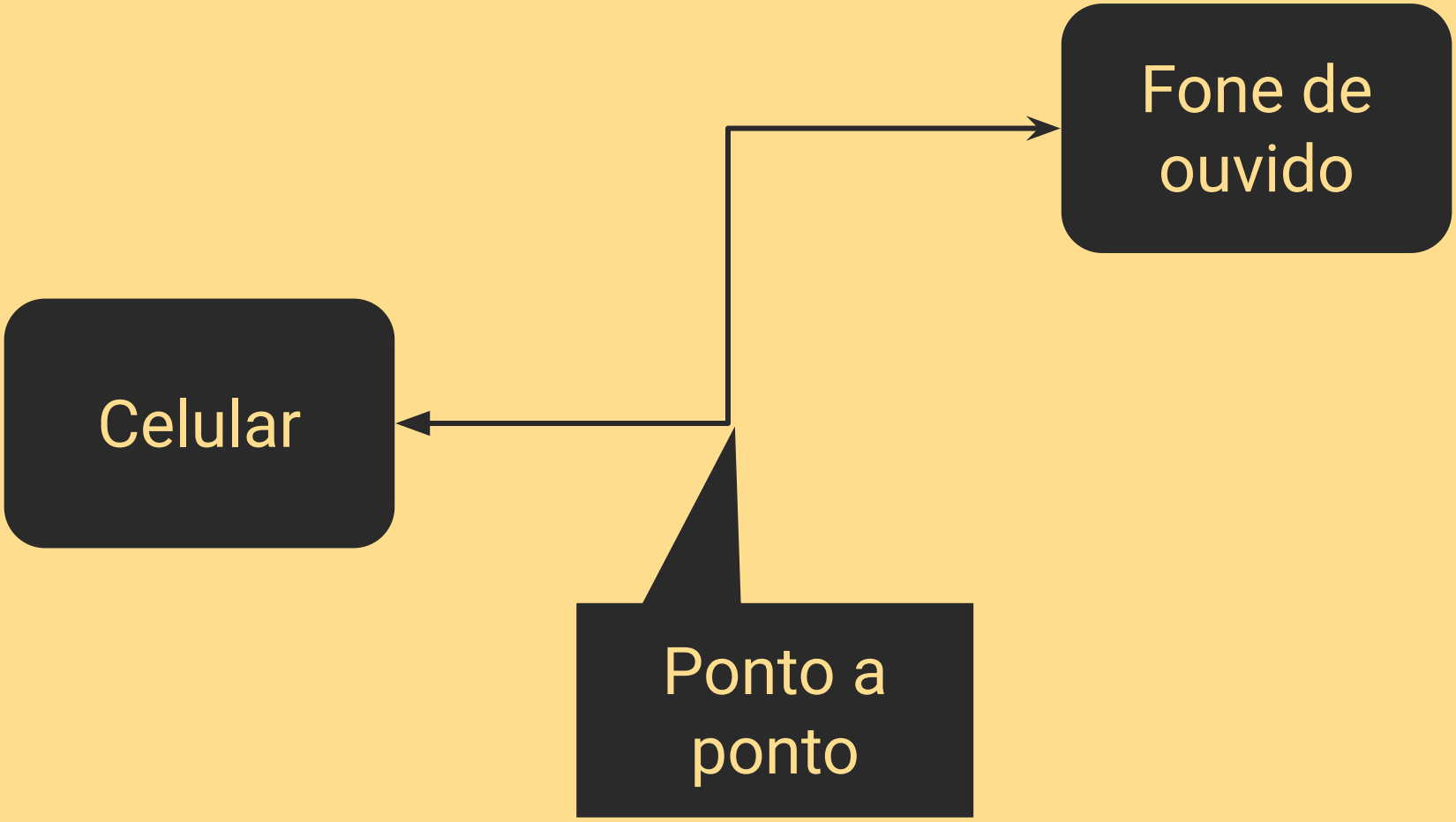
Personal Area Network (Rede de Área Pessoal)



Tipos de redes — PAN

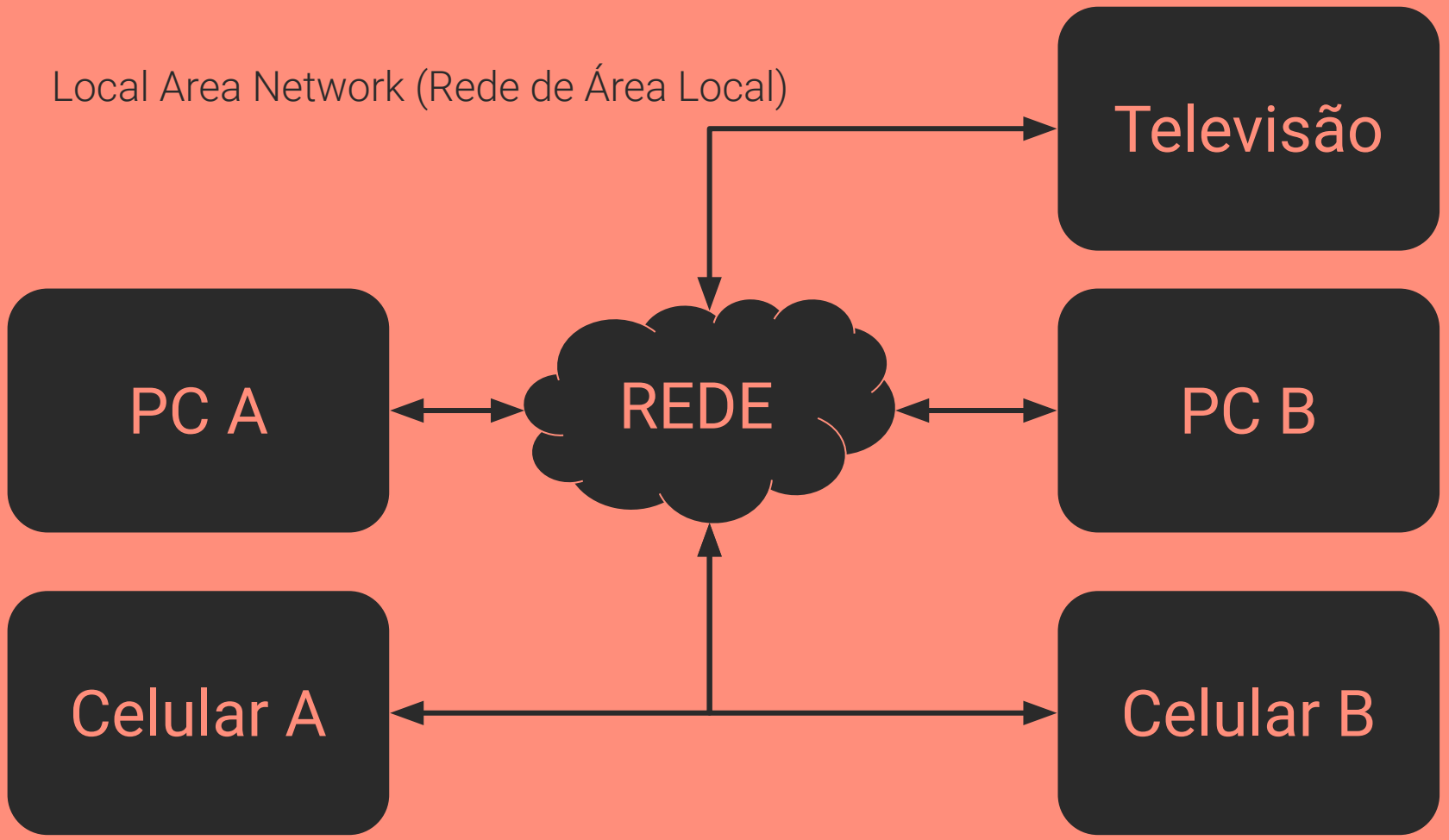
Personal Area Network (Rede de Área Pessoal)




Fone de ouvido
Celular
Ponto a ponto

Tipos de rede — LAN

Local Area Network (Rede de Área Local)



Switch

Televisão

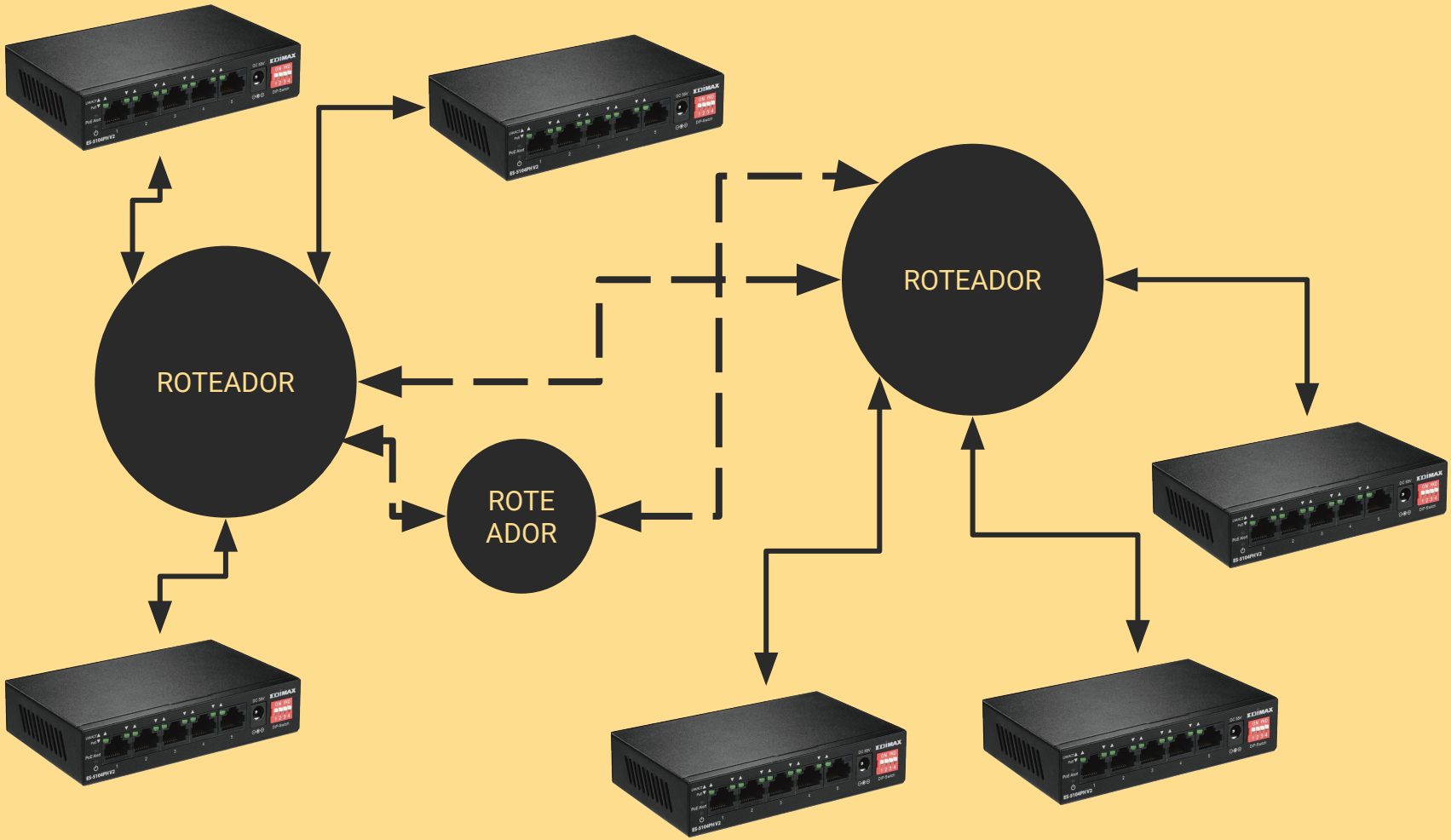
PC A

PC B

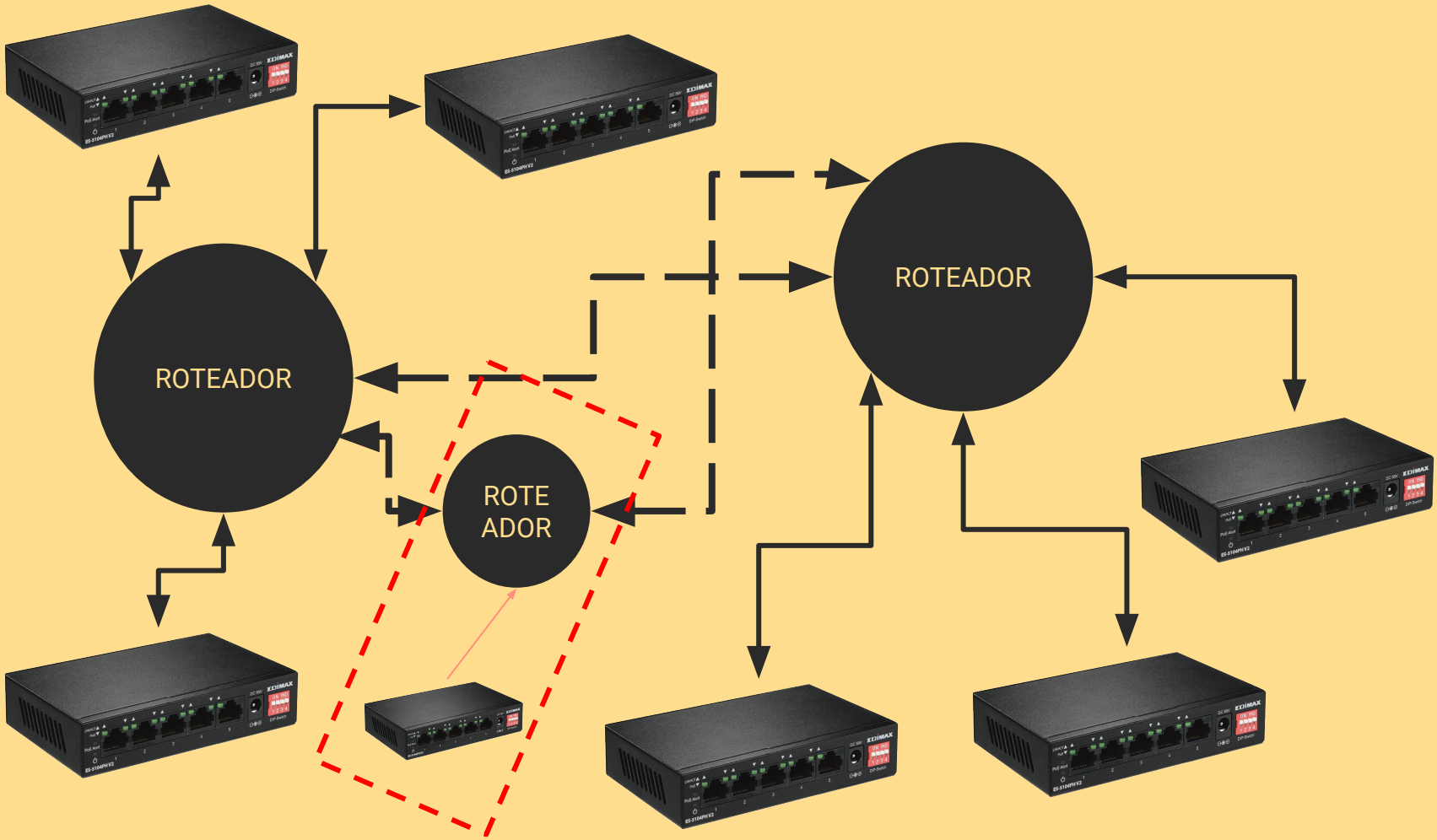
Celular A

Celular B

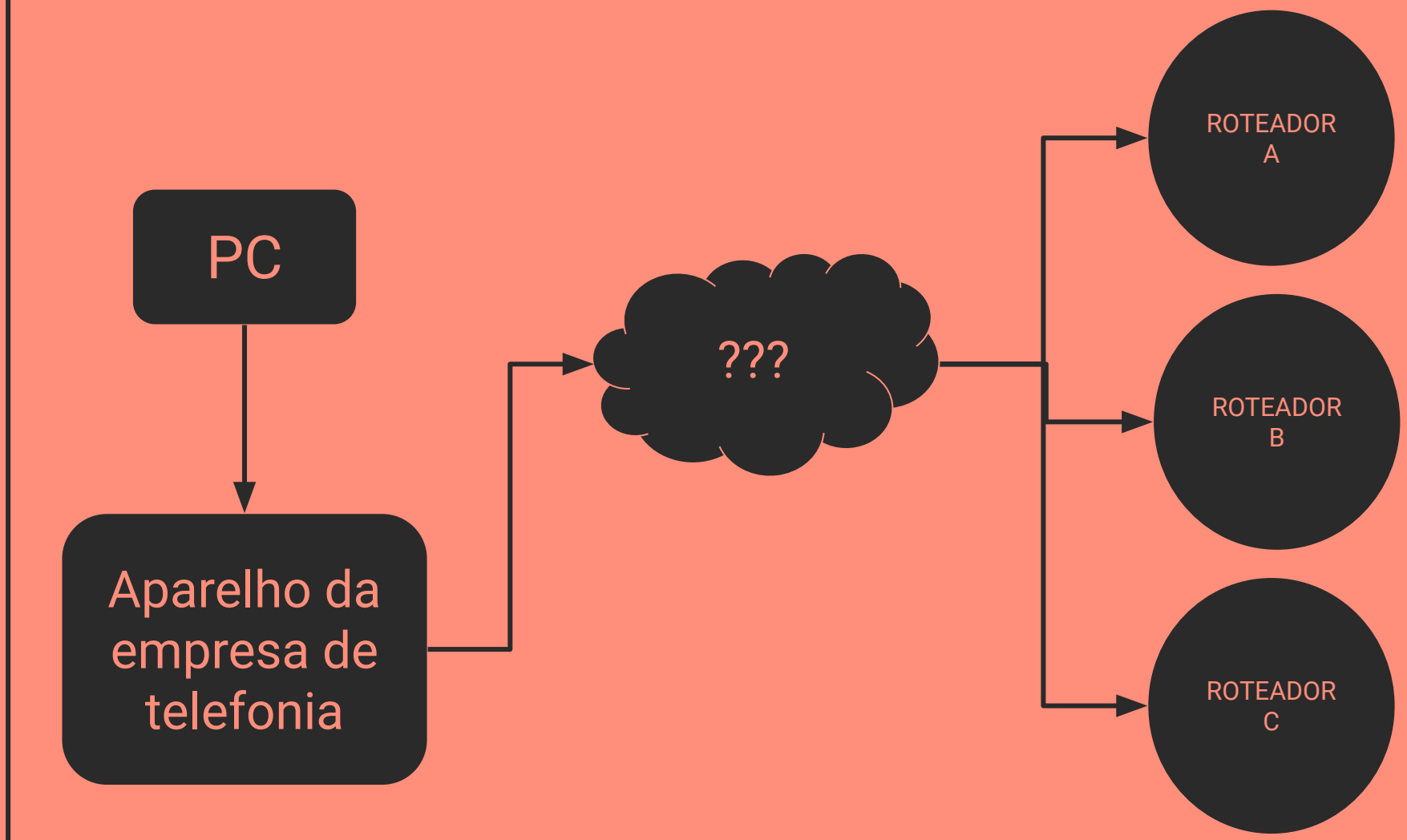
Tipos de rede — WAN (wide)



Tipos de rede — WAN (wide)



Nosso fluxo até a internet



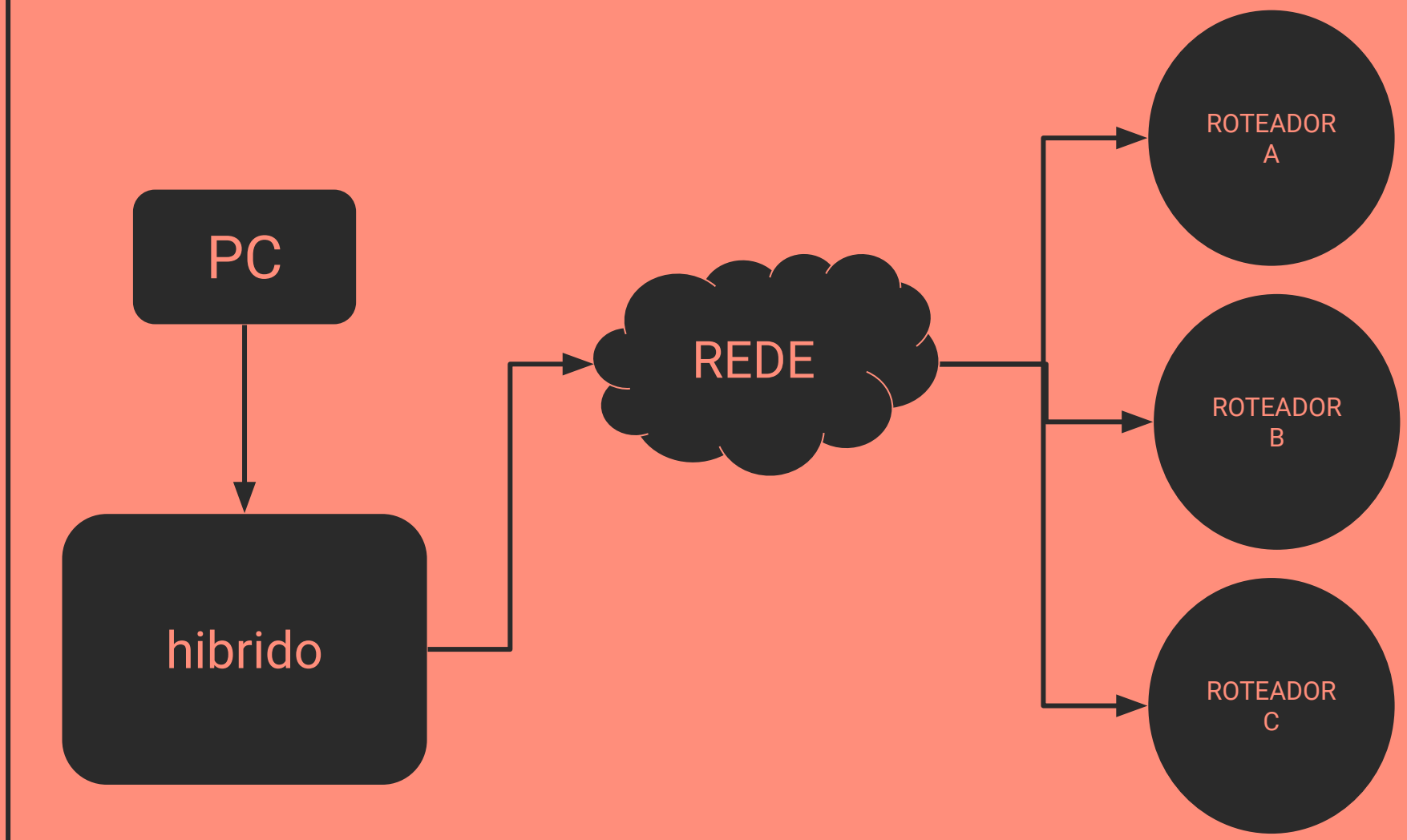
O monstro que as empresas nos "emprestam"

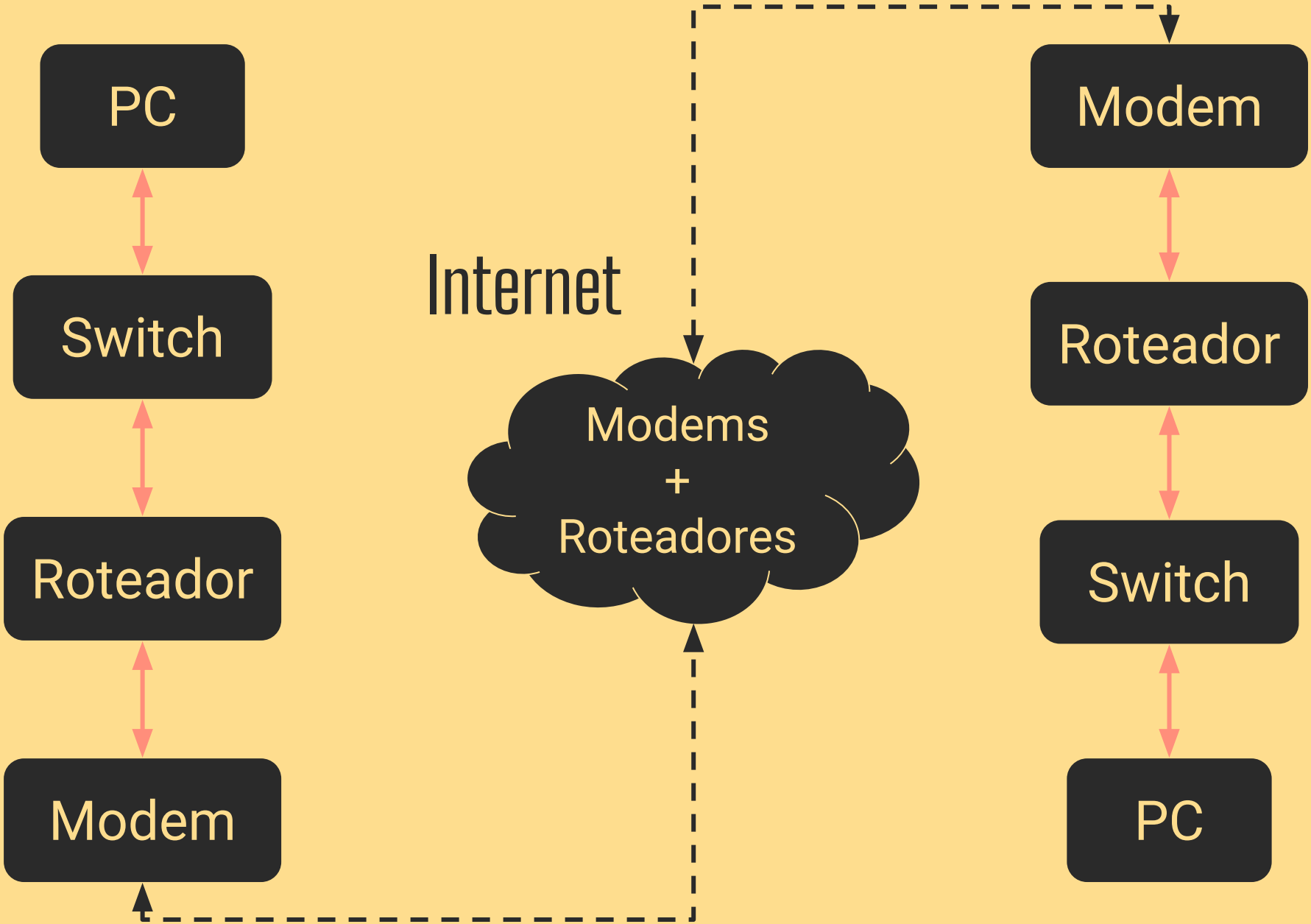
O r

Um aparelho hibrido muito maluco:

- Switch (Permite rede local)
- Roteador (Permite interação com outros)
- Modem (Converte sinais digitais para analógicos)

Nosso fluxo até a internet




PC
Switch
Roteador
Modem
Internet
Modems
+
Roteadores
Modem
Roteador
Switch
PC

Isso é a parte física!

O básico necessário

Rede de computadores é um conjunto de dois ou mais dispositivos interligados por um sistema de comunicação digital **que seguem algumas regras e especificações para compartilhar recursos entre si**.

A parte lógica das redes é dividida em camadas

Para que a comunicação entre os dispositivos aconteça foram criados diversos protocolos que padronizam a comunicação. Os protocolos são divididos em camadas.

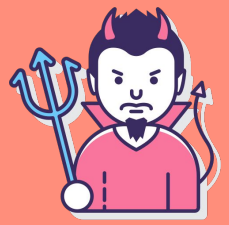
A parte lógica das redes é dividida em camadas

Para que a comunicação entre os dispositivos aconteça foram criados diversos protocolos que padronizam a comunicação. Os protocolos são divididos em camadas.

A parte lógica das redes é dividida em camadas

Para que a comunicação entre os dispositivos aconteça foram criados diversos protocolos que padronizam a comunicação. Os protocolos são divididos em camadas.

Filósofo-Tradutor-Secretária (Tanenbaum)



Odeio humanos

L: Esperanto

Mi malamas homojn

Bola santa

L: Esperanto

Mi malamas homojn

I hate humans

L: Esperanto

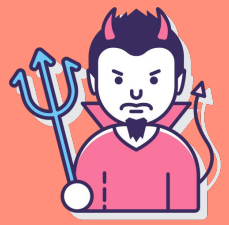
Mi malamas homojn

Bola santa

L: Esperanto

Mi malamas homojn

Filósofo-Tradutor-Secretária (Tanenbaum)



Odeio humanos

Mensagem

I hate humans

L: Esperanto

Mi malamas homojn

Tradução

L: Esperanto

Mi malamas homojn

Bola santa

L: Esperanto

Mi malamas homojn

Destino

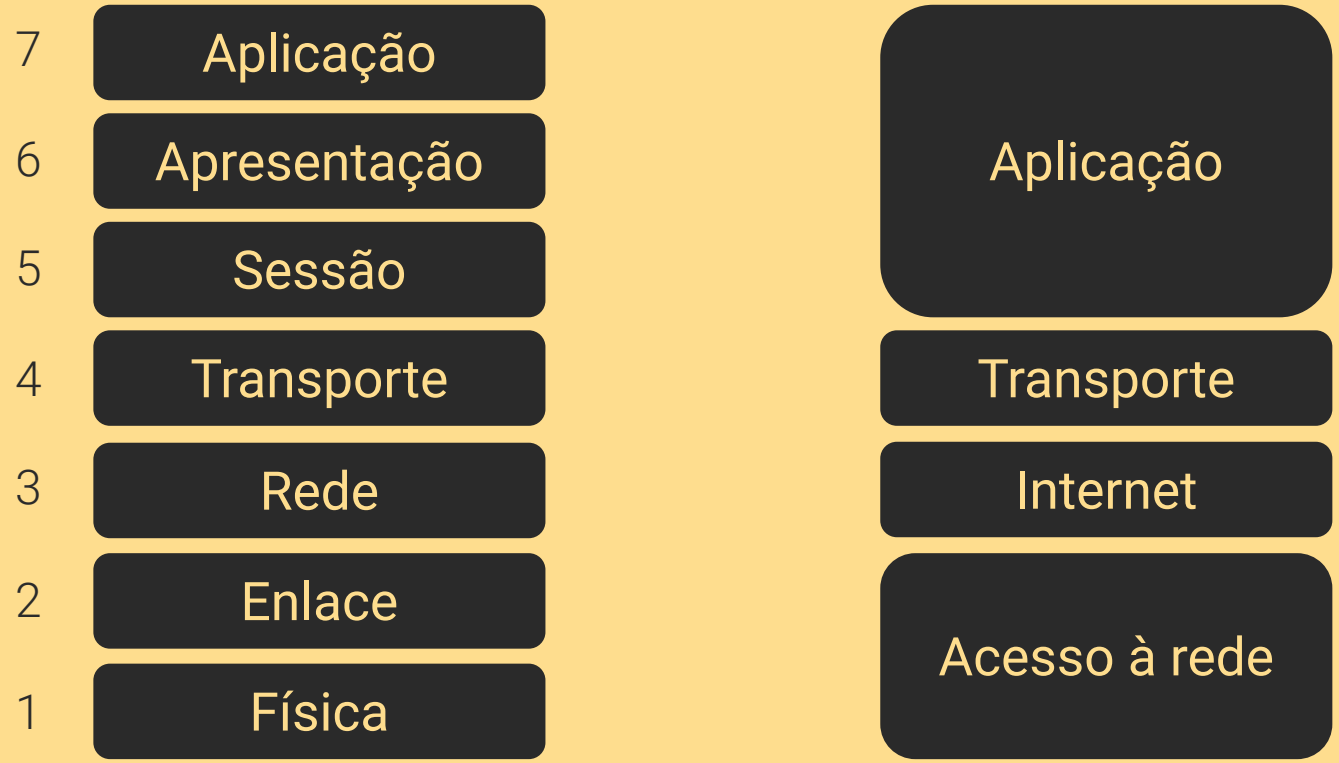
Bola santa

L: Esperanto

Mi malamas homojn

Modelos de referência

Existem dois modelos de referência quando falamos de camadas de rede

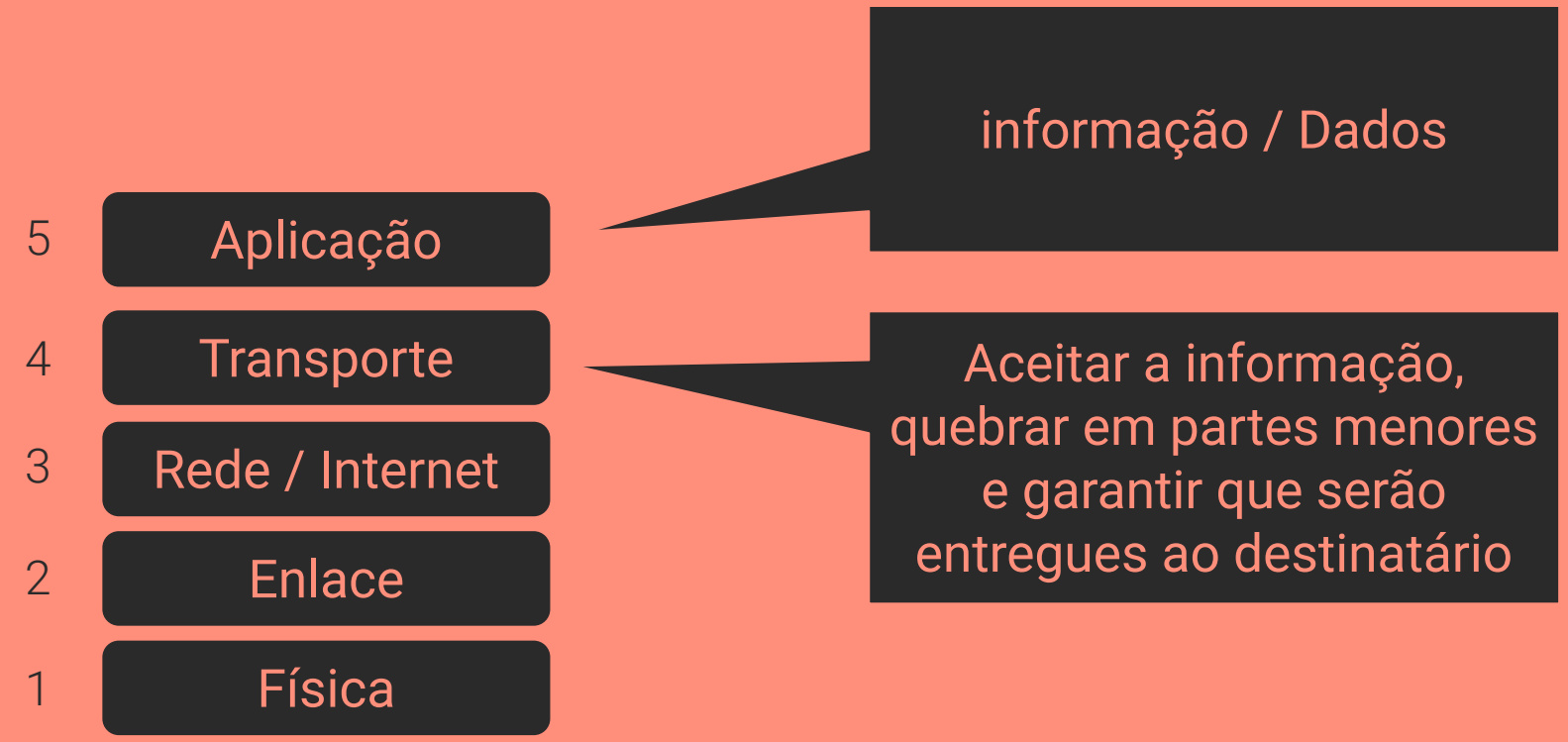


Um modelo mais realista

Existem dois modelos de referência quando falamos de camadas de rede

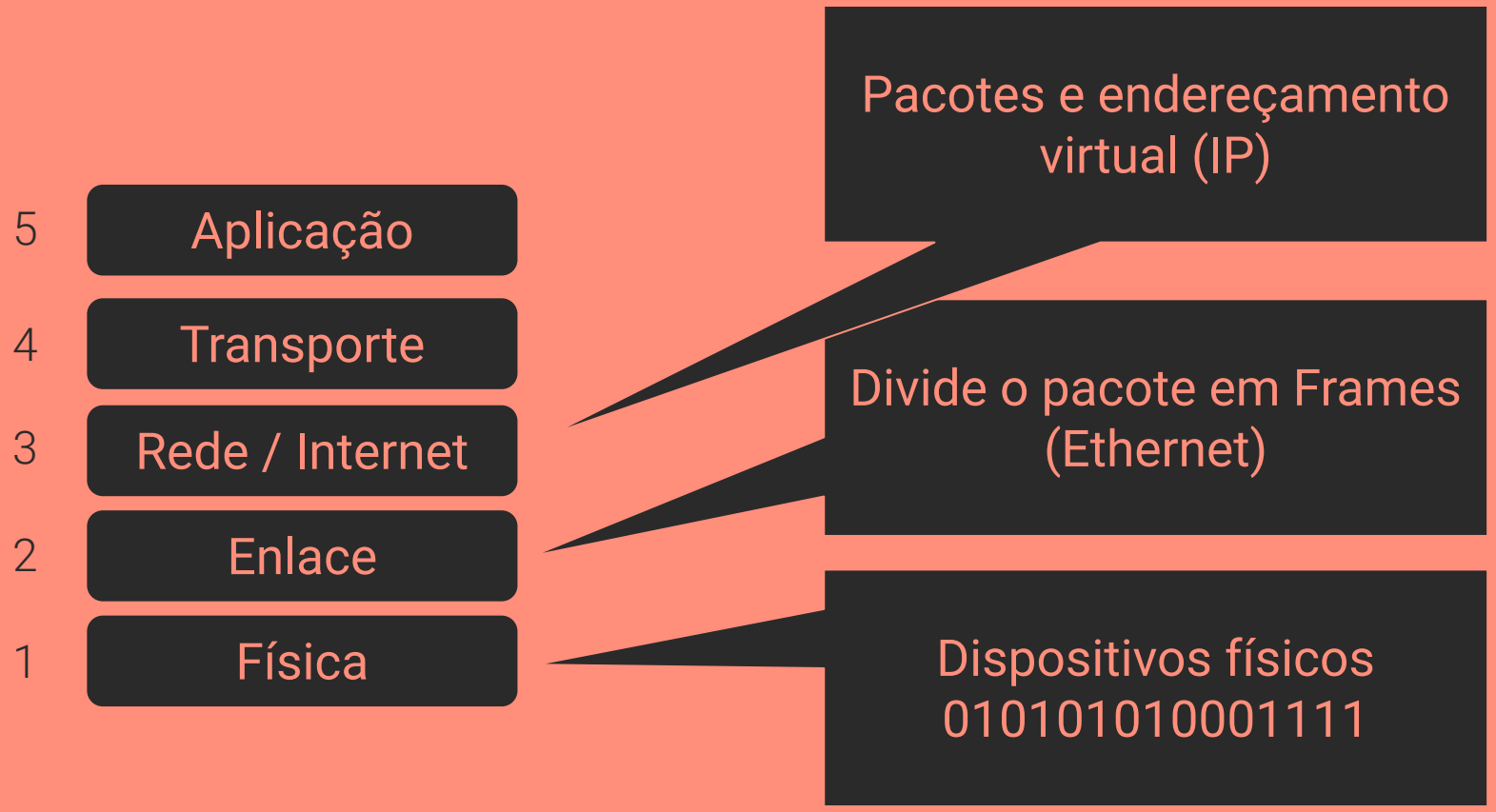
Um modelo mais realista

Existem dois modelos de referência quando falamos de camadas de rede



Um modelo mais realista

Existem dois modelos de referência quando falamos de camadas de rede



Um aló antes dos protocolos

Scapy

Scapy

Scapy é uma biblioteca python para criação (válidos ou não), manipulação e farejamento de pacotes de rede. Com ele você consegue trabalhar em todas as camadas.

- Licença: software livre GPL2
- Primeira versão: teve seu primeiro commit no git em 2003 (é mais antiga, mas não consegui informações antes de 2003) +20 anos
- Atual: versão 2.4.5
- Suporta: Linux, OSX, BSDs e Windows. Funciona com as versões 2 e 3 do Python

Scapy por Scapy

Scapy é um poderoso programa interativo de manipulação de pacotes. É capaz de forjar ou decodificar pacotes de um grande número de protocolos, enviá-los no fio, capturá-los, combinar solicitações e respostas e muito mais. Ele pode lidar facilmente tarefas mais clássicas, como varredura, tracerouting, sondagem, testes de unidade, ataques ou descoberta de rede (pode substituir hping, 85% de nmap, arpspoof, arp-sk, arping, tcpdump, tshark, p0f, etc.). Ele também funciona muito bem em muitos outras tarefas específicas que a maioria das outras ferramentas não consegue lidar, como enviar mensagens inválidas quadros, injetando seus próprios quadros 802.11, combinando técnicas (salto de VLAN+ARP envenenamento de cache, decodificação VOIP em canal criptografado WEP, ...), etc.

https://scapy.net/

Curiosidade

O nome sCAPy é oriundo do libCAP.

Biblioteca base para o TCPdump e que criou BPF (veremos isso mais tarde??)

pip install scapy

Instalação

pip install scapy[basic]

Caso queira o shell interativo

Dependências

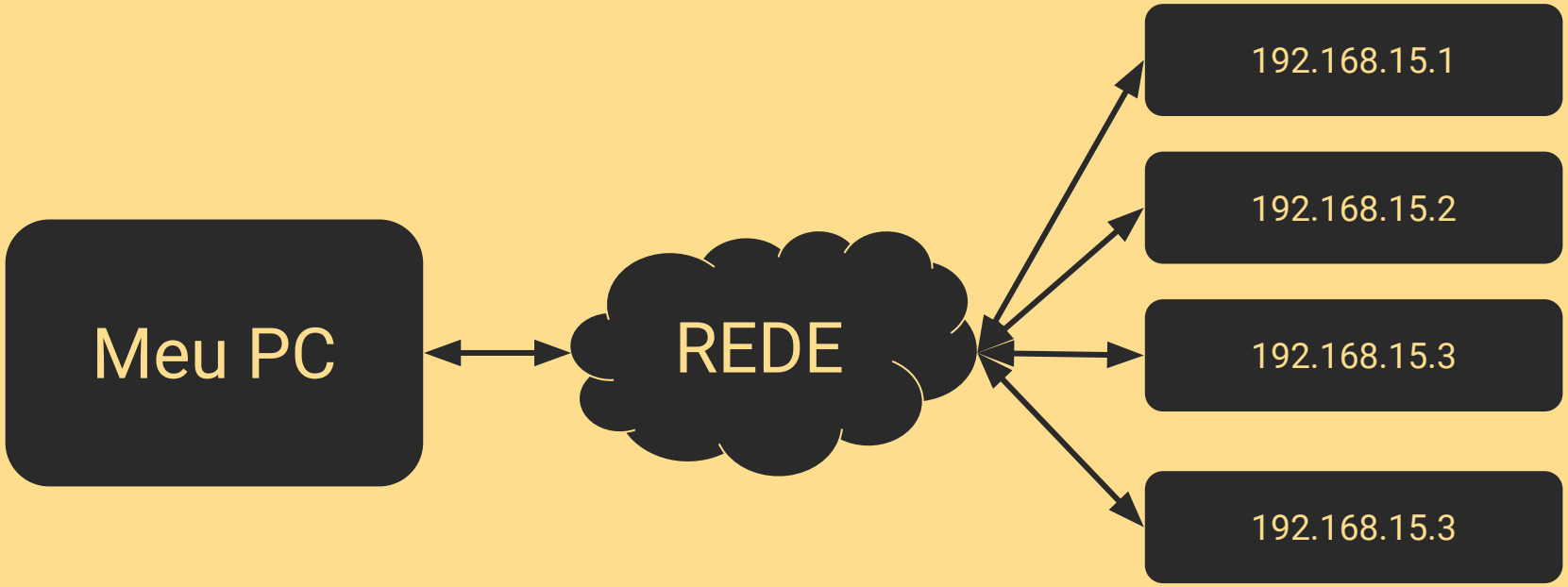
Para o funcionamento pleno do scapy precisamos instalar o libcap no sistema:

- Pacote no **linux**: tcpdump
- Pacote no **OSX**: libcap
- Pacotes nos **BSDs**: libpcap tcpdump
- Pacote no **Windows**: Winpcap

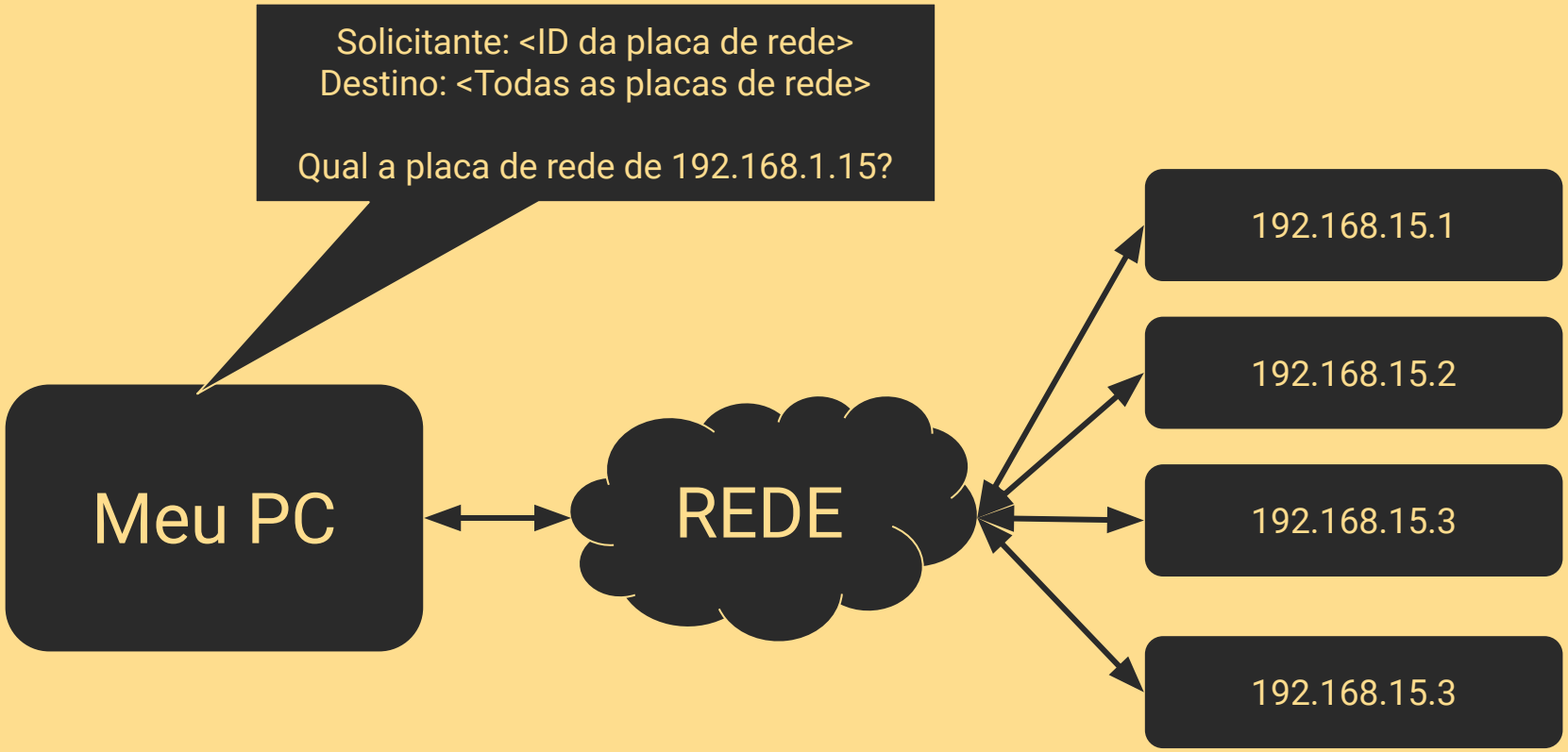
Um exemplo básico (ARP)

Quero falar com o endereço IP 192.168.15.1.

Quem é você na rede?



Um exemplo básico (ARP)



O código

```python
from scapy.layers.l2 import Ether, ARP
from scapy.sendrecv import srp1

BROADCAST_MAC = 'ff:ff:ff:ff:ff:ff'
SERVER_IP = '192.168.15.1'

pacote_ethernet = Ether(dst=BROADCAST_MAC)
pacote_arp = ARP(pdst=SERVER_IP)
pacote_final = pacote_ethernet / pacote_arp

resposta = srp1(pacote_final, timeout=2)
resposta.show()
```

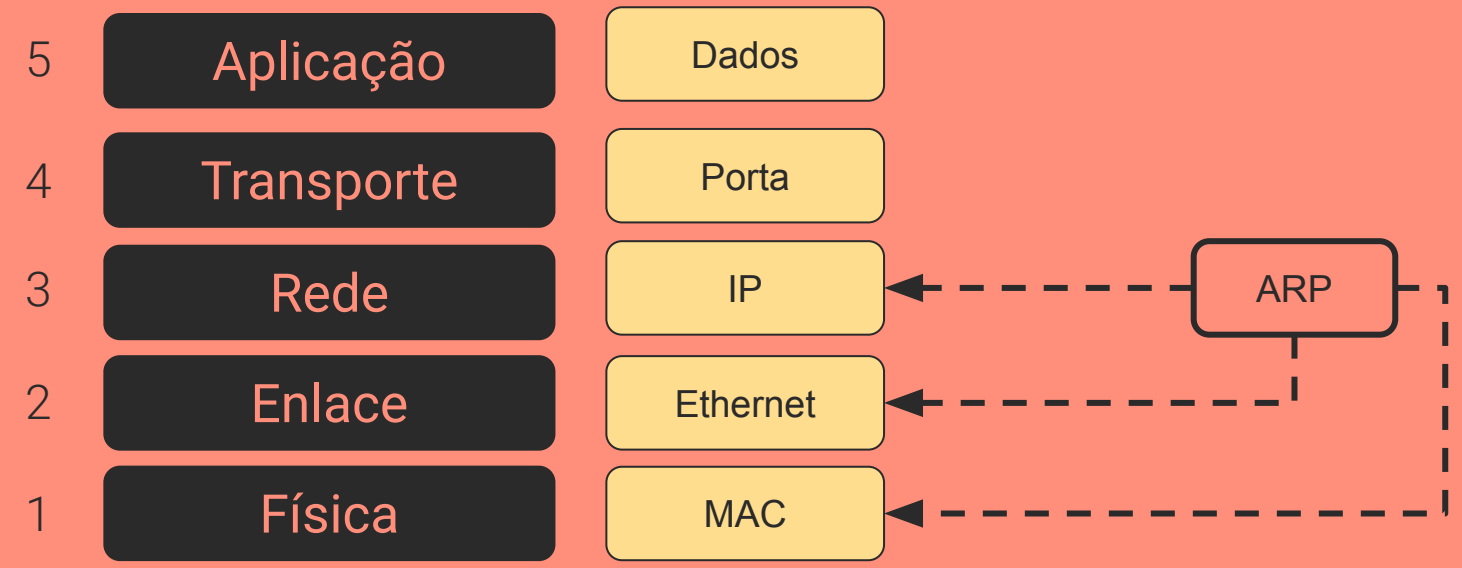
Camadas e procolos

Botando o scapy pra funcionar

Camadas e protocolos

De maneira simplista, protocolos tem um único objetivo.

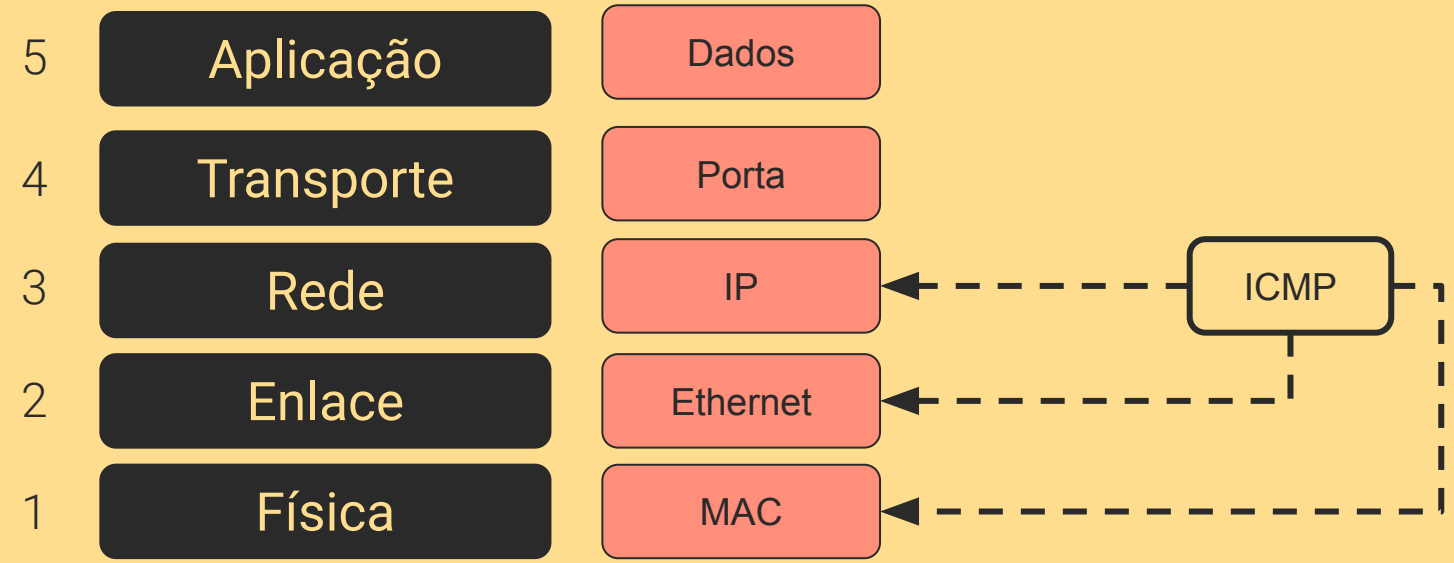
ARP (Address Resolution Protocol): Descobrir quem tem um determinado IP



Camadas e protocolos

ICMP (Internet Control Message): Você que tem esse IP, está de pé?

(O famoso PING)



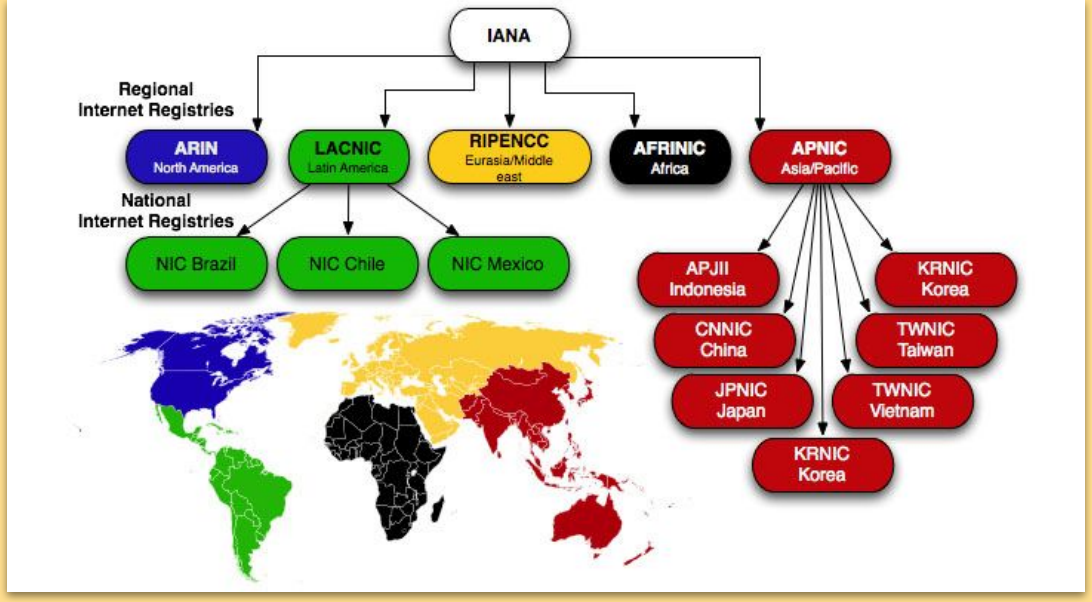
O código

```python
from scapy.layers.l2 import Ether
from scapy.layers.inet import IP, ICMP
from scapy.sendrecv import srp1

def ping(origem_ip, origem_mac, destino_ip, destino_mac):
    pacote = Ether(src=origem_mac, dst=destino_mac)
    pacote /= IP(src=origem_ip, dst=destino_ip)
    pacote /= ICMP()

    return srp1(pacote, timeout=2)
```

DNS (Domain Name Server)

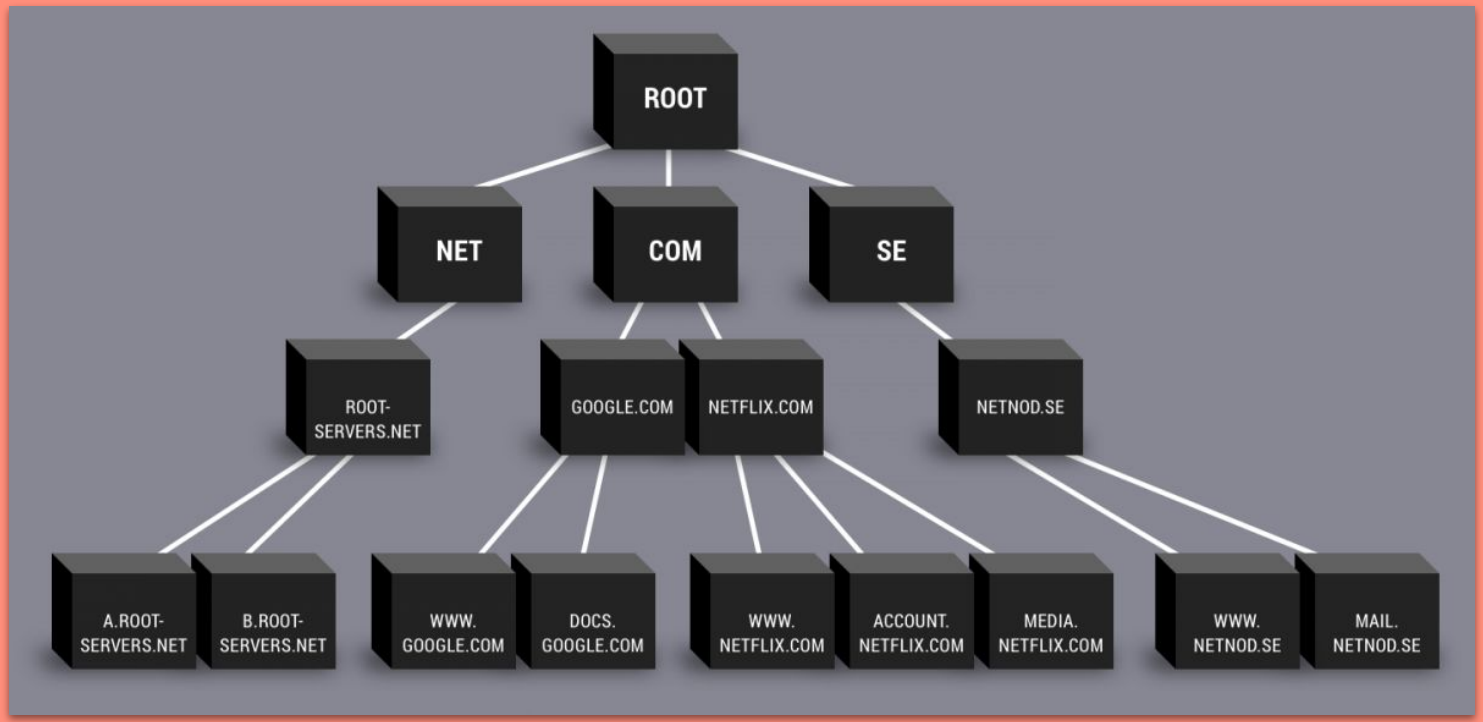
Mas se as máquinas falam por identificadores IP, como consigo acessar coisas por nome. Como "www.duckduckgo.com"?

Existe um serviço na rede chamado DNS, é um servidor que resolve nomes. Como assim resolver? Ele lista os nomes referentes a cada IP.



DNS

Internet Assigned Numbers Authority



Os servidores são interligados

Então posso mandar uma pergunta destina ao meu servidor de dns e ele perguntará a outros, até encontrar o IP requerido.

```python
from scapy.layers.inet import UDP, IP
from scapy.layers.dns import DNS, DNSQR  # Question Record
from scapy.sendrecv import sr1


def host_ip(url='ddg.gg'):
    packet = IP(dst='192.168.15.1')
    packet /= UDP(dport=53)
    packet /= DNS(qd=DNSQR(qname=url))

    return sr1(packet)
```

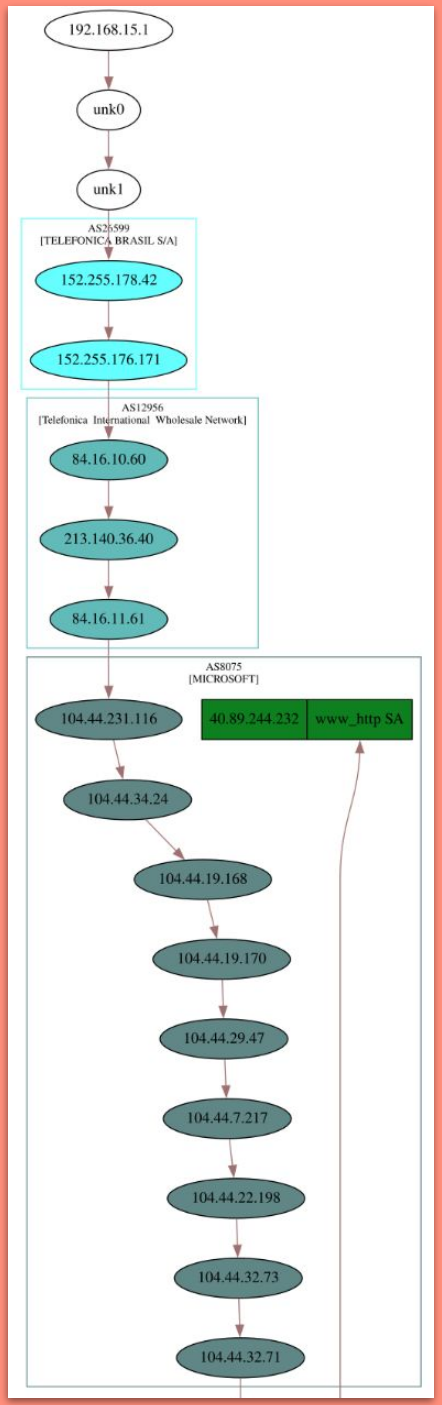
Mas como essa requisição corre na rede?

Por exemplo, uma máquina vai até outra, que vai até outra. Existem roteadores interligando todas as redes? Como saber o caminho percorrido?

```
1   from scapy.layers.inet import traceroute
2
3   caminho = traceroute('<URL>', verbose=False)
```

Pulos entre roteadores

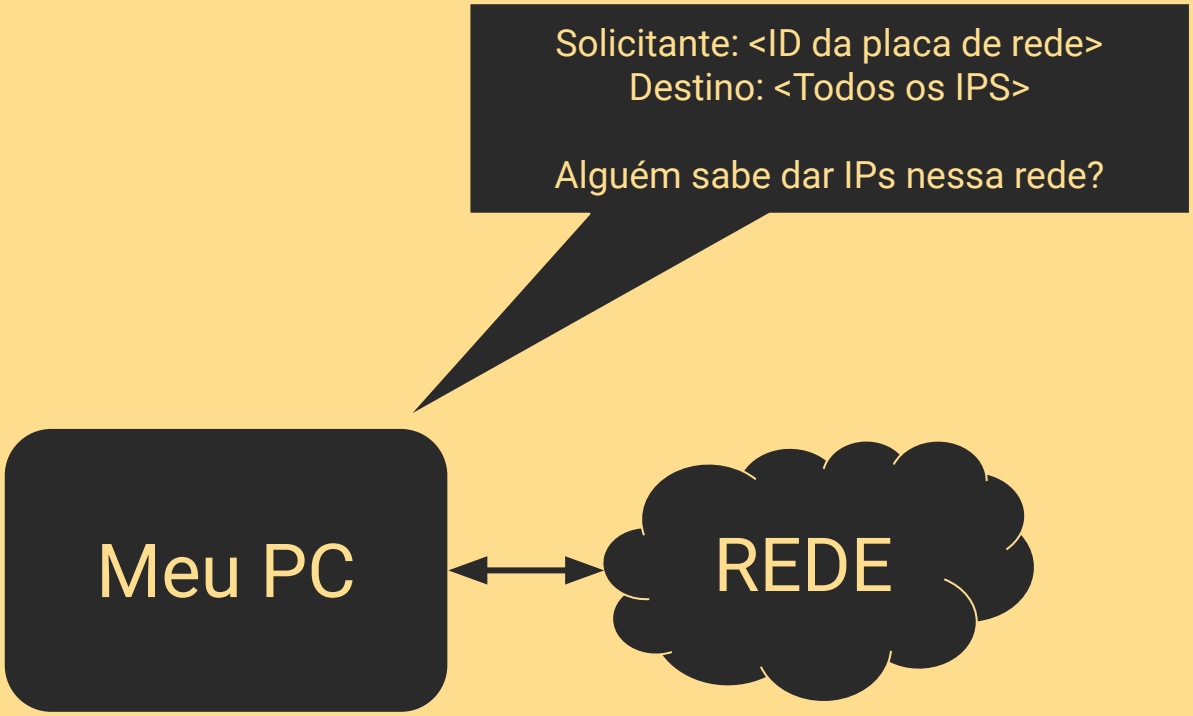
```
1  from scapy.layers.inet import traceroute
2  resp, _ = traceroute('ddg.gg', verbose=False)
3  resp.show()
4  '''
5     40.89.244.232:tcp80
6  1  192.168.15.1    11
7  4  152.255.178.42  11
8  5  152.255.176.171 11
9  6  84.16.10.60     11
10 7  213.140.36.40   11
11 8  84.16.11.61     11
12 9  104.44.231.116  11
13 10 104.44.34.24    11
14 11 104.44.19.168   11
15 12 104.44.19.170   11
16 13 104.44.29.47    11
17 14 104.44.7.217    11
18 15 104.44.22.198   11
19 16 104.44.32.73    11
20 17 104.44.32.71    11
21 28 40.89.244.232   SA
22 29 40.89.244.232   SA
23 30 40.89.244.232   SA
24 '''
25 resp.graph() # gerou a imagem ao lado
```



Mas espera!

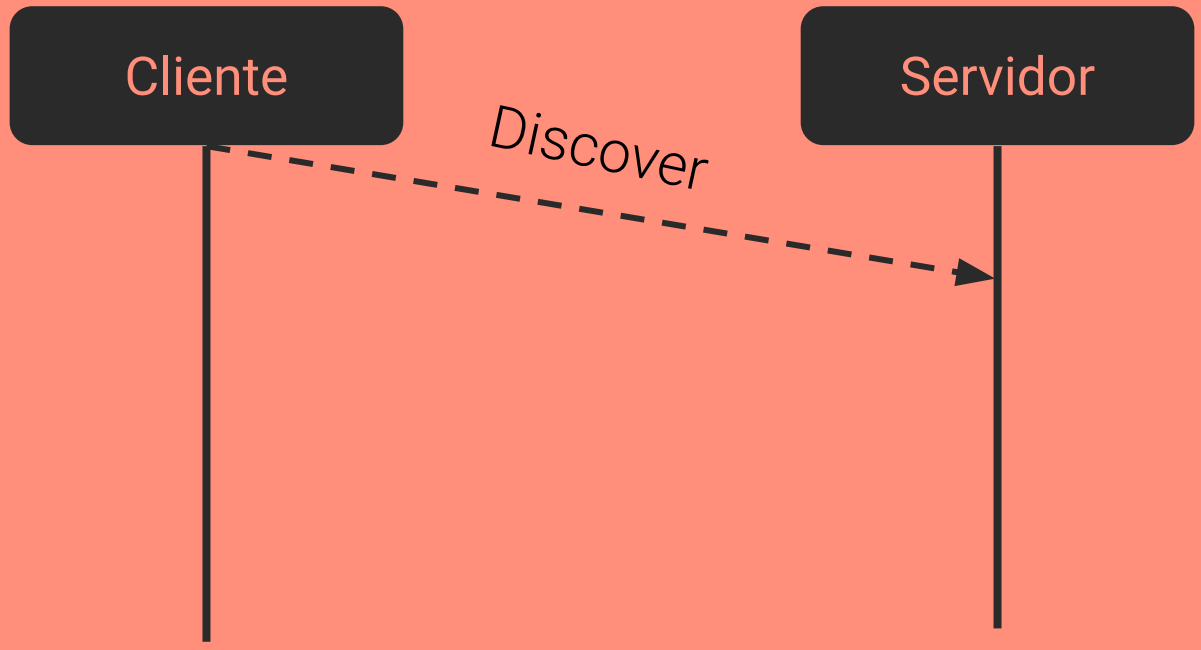
Como obtenho um endereço IP?

Entro na rede e ele simplesmente surge?

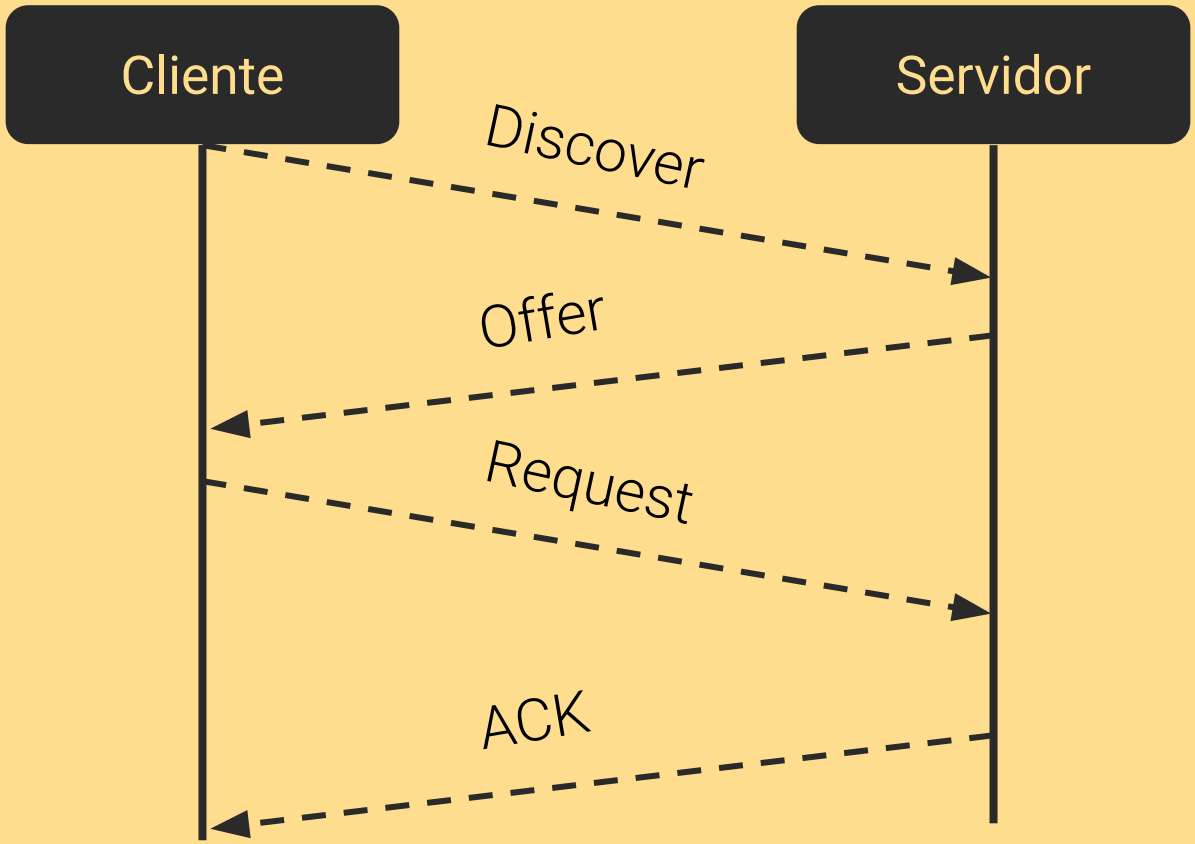


DHCP (Dynamic Host Configuration)

Em praticamente todas as redes existe um servidor de DHCP. Quando o computador se conecta a rede, ele pergunta a todos os nós (broadcast) perguntando se alguém pode prover um endereço de IP válido nessa rede.

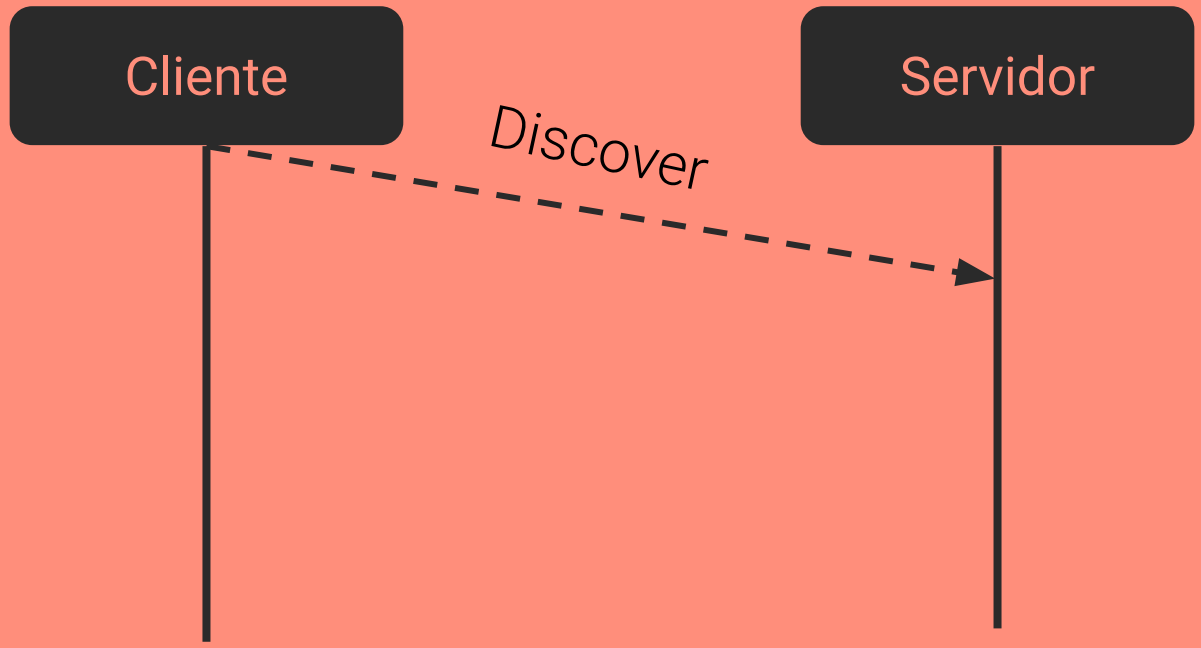


DHCP (Dynamic Host Configuration)



DHCP (Dynamic Host Configuration)

Vamos nos concentrar no começo primeiro!



Discover

```python
def dhcp_discovery():
    """Decobindo quem é o servidor de DHCP."""
    pacote = Ether(src=conf.iface.mac, dst=BROADCAST_MAC)
    pacote /= IP(src='0.0.0.0', dst=BRADCAST_IP)
    pacote /= UDP(sport=68, dport=67)
    pacote /= BOOTP(chaddr=get_if_raw_hwaddr(conf.iface)[1], xid=RandInt())
    pacote /= DHCP(
        options=[
            ('message-type', 'discover'),
            'end'
        ]
    )

    sendp(pacote, iface=conf.iface)
```

Sniffing

Captura de pacotes

Sniffing

É uma forma de monitoramento (farejamento) de pacotes. Com eles podemos ver todos os pacotes que trafegam nas nossas interfaces de rede. Pacote por pacote. Tudo que passa na nossa placa fica registrado.

Existem programas incríveis para fazer esse tipo de coisa, como Wireshark.

Scapy Sniffer

O scapy tem dois sniffer disponíveis. Um síncrono. Para visualização e eventos. E um async para pregarmos os resultados em lotes.

```
1   from scapy.senrecv import sniff
2
3   sniff(filter='<filtro>', prn=função_de_callback)
```

Async Sniff

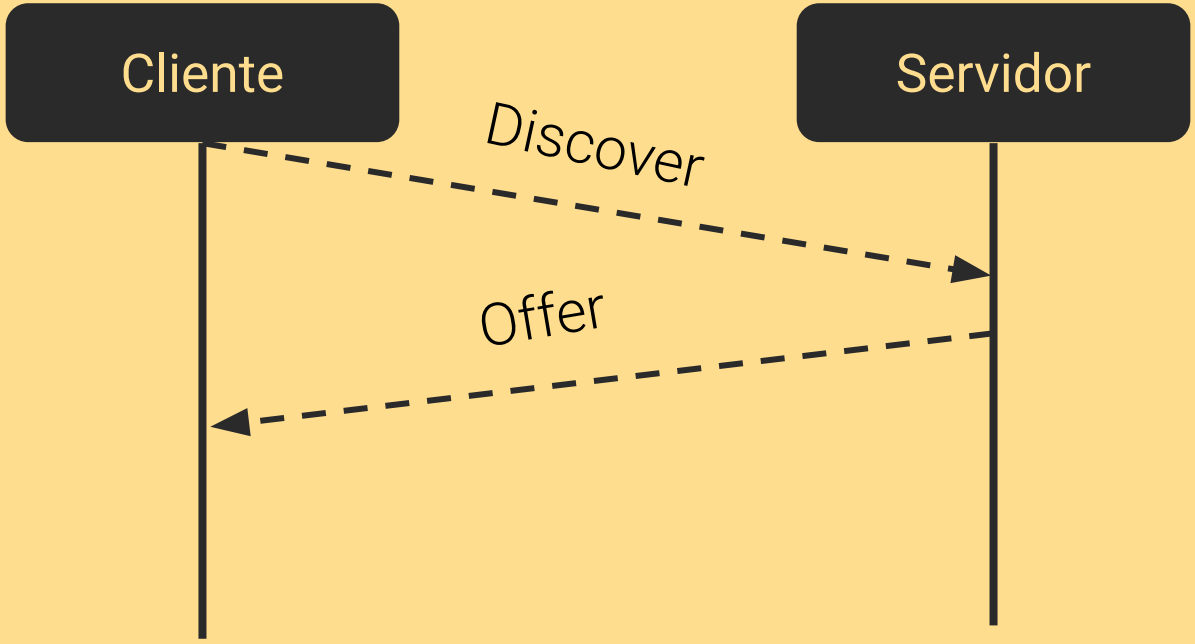
```python
from scapy.senrecv import AsyncSniffer

t = AsyncSniffer()
t.start()
time.sleep(1)
print(t.results)
t.stop()
```

De volta ao DHCP

Porque sim

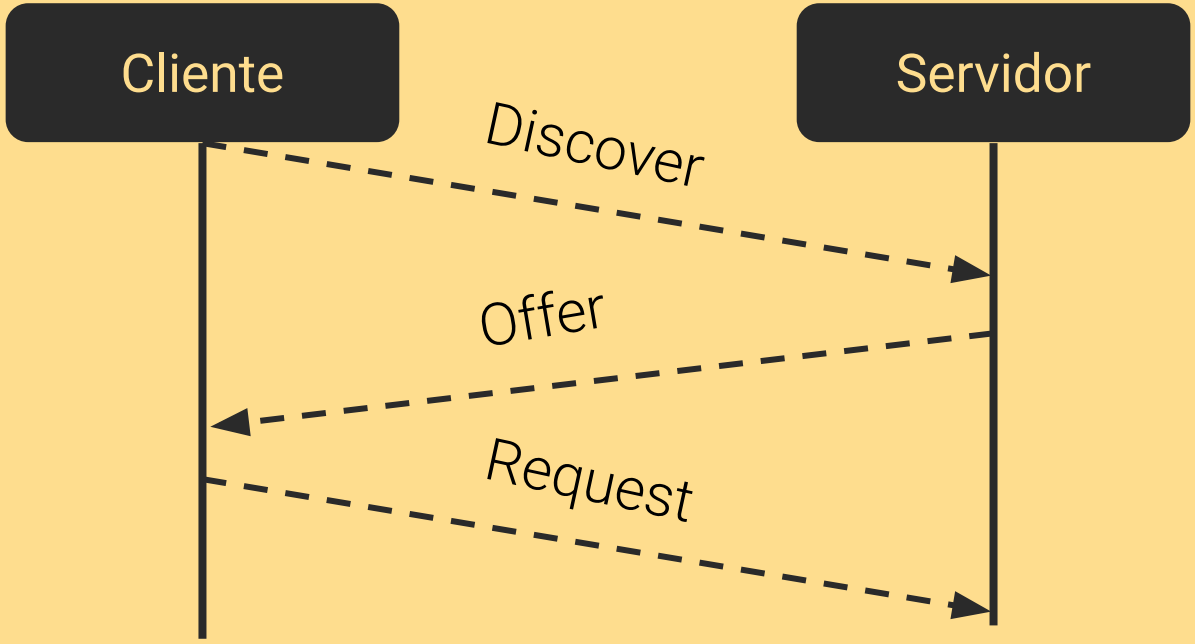
DHCP (Dynamic Host Configuration)

Em praticamente todas as redes existe um servidor de DHCP. Quando o computador se conecta a rede, ele pergunta a todos os nós (broadcast) perguntando se alguém pode prover um endereço de IP válido nessa rede.



DHCP (Dynamic Host Configuration)

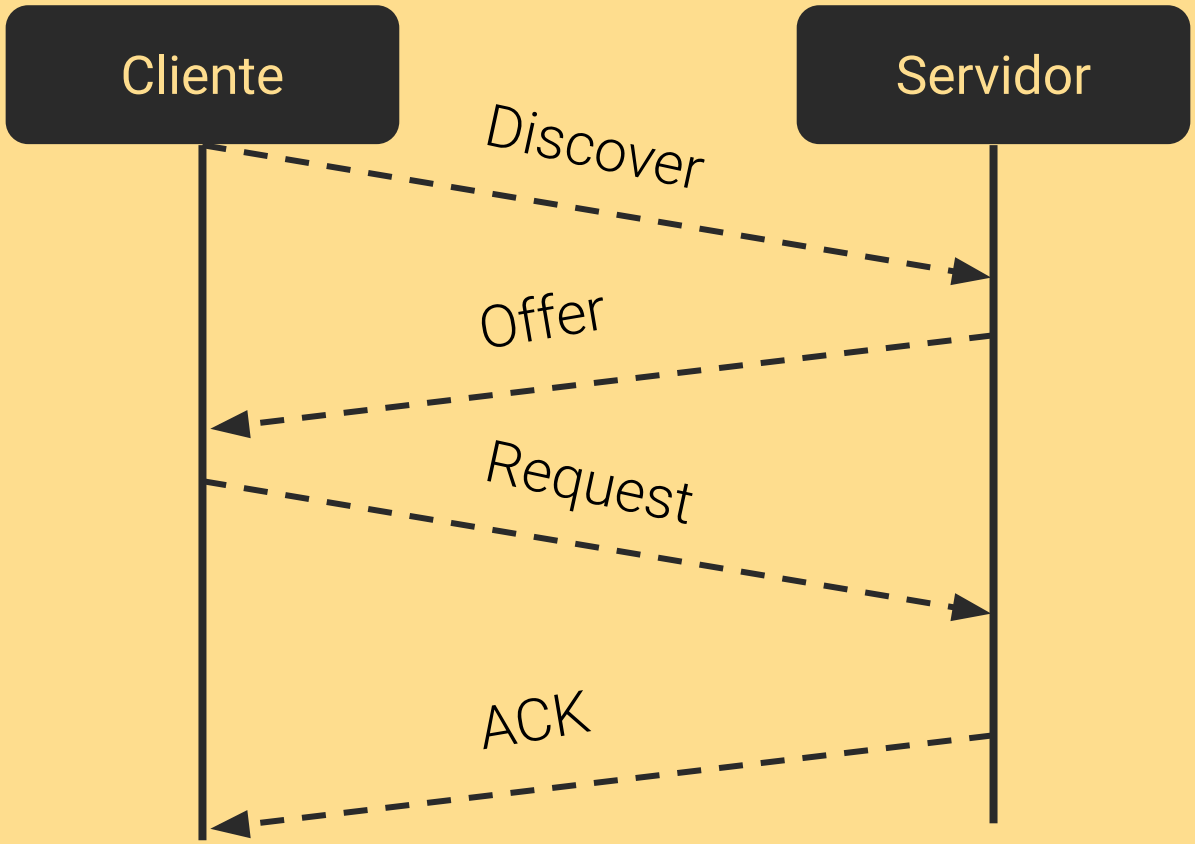
Em praticamente todas as redes existe um servidor de DHCP. Quando o computador se conecta a rede, ele pergunta a todos os nós (broadcast) perguntando se alguém pode prover um endereço de IP válido nessa rede.



Request

```python
def dhcp_request(server_id, requested_ip):
    pacote = Ether(src=conf.iface.mac, dst=BROADCAST_MAC)
    pacote /= IP(src='0.0.0.0', dst=BRADCAST_IP)
    pacote /= UDP(sport=68, dport=67)
    pacote /= BOOTP(
        chaddr=get_if_raw_hwaddr(conf.iface)[1],
        xid=RandInt()
    )
    pacote /= DHCP(
        options=[
            ('message-type', 'request'),
            ('server_id', server_id),
            ('requested_addr', requested_ip),
            'end'
        ]
    )
    sendp(pacote)
```

DHCP (Dynamic Host Configuration)



picpay.me/dunossauro

apoia.se/livedepython

pix.dunossauro@gmail.com

Ajude o projeto <3