



DTECH

Camada de Aplicação

Prof. Rone Ilídio da Silva

DTECH-CAP-UFSJ

Baseado no capítulo II do livro Redes de Computadores e a Internet - Kurose

Itens do Livro

Redes de Computadores e a Internet, Kurose 5ª edição

2 Camada de Aplicação

2.1 Princípios de aplicações de rede

2.1.1 Arquiteturas de aplicação de rede

2.1.2 Comunicação entre processos

2.1.3 Serviços de transporte disponíveis para a aplicação

2.1.4 Serviços de transporte providos pela Internet

2.1.5 Protocolos da camada de aplicação

2.2 A Web e o HTTP

2.2.1 Descrição geral do HTTP

2.2.3 Formato da mensagem HTTP

2.2.4 Interação usuário-servidor: cookies

2.2.5 Caches Web

2.3 Transferência de arquivo: FTP

2.5 DSN: o serviço de diretório da Internet

2.5.1 Serviços fornecidos pelo DNS

2.5.2 Visão geral do modo de funcionamento do DNS



Universidade Federal
de São João del-Rei

Introdução

- A Camada de Aplicação possui os protocolos que fornecem os serviços utilizados pelos usuários
- Exemplos:
 - **HTTP**: páginas
 - **FTP**: transferência de arquivos
 - **DNS**: conversão url – endereço IP
 - SMTP: e-mail
 - IPTV: conteúdo televisivo (fluxo contínuo)



Introdução

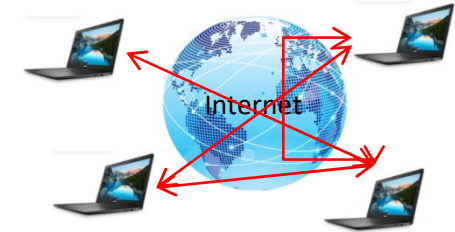
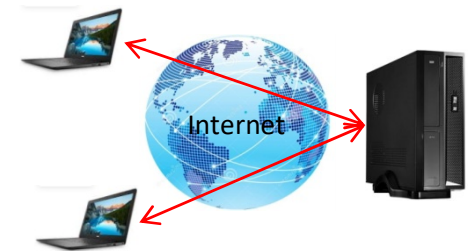
- Protocolos da Camada de Aplicação: definem como os processos de uma aplicação, que funcionam em sistemas finais diferentes, passam mensagens entre si.
- Um protocolo deve definir:
 - Tipos de mensagens
 - Sintaxe (como cada campo é posicionado)
 - Semântica (o que é cada campo)
 - Regras de envio e recepção

Introdução



Arquiteturas de aplicação de rede

- Cliente-servidor
 - Um servidor atende a vários clientes
 - Infraestrutura cara com o aumento do número de clientes
 - Exemplo: Web, e-mail, Telnet e FTP
- Peer to peer (P2P)
 - Os pares de computadores se comunicam diretamente
 - Mais escalável
 - Infraestrutura inexistente
 - Problemas com segurança
 - Problemas com tráfego baixo de upload
 - Ex: BitTorrent e eMule



Obs: BitTorrent é híbrido, pois possui um Tracker como servidor central com dados dos pares

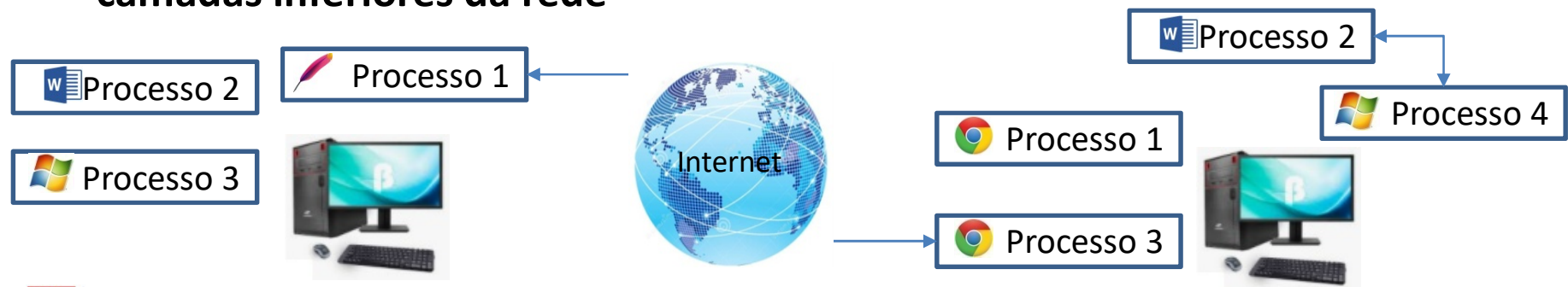
Comunicação entre Processos



Universidade Federal
de São João del-Rei

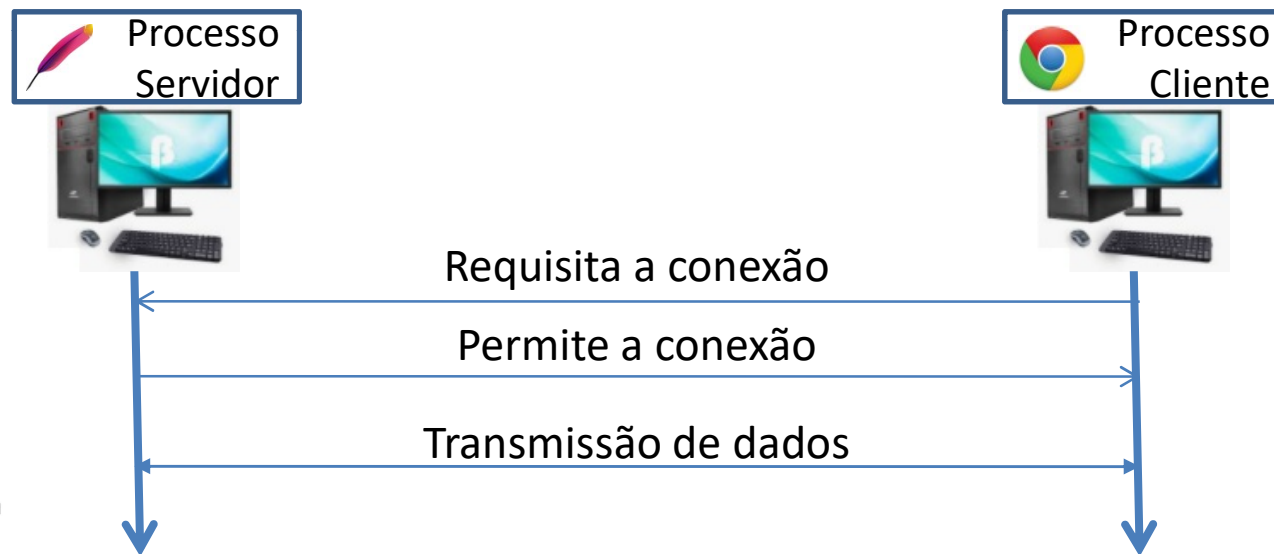
Comunicação entre Processos

- Podemos entender um processo como um programa executado em uma máquina
- Um processo se comunica com outro por um socket
- Socket cliente e Socket Servidor
- **Socket pode ser definido como a interface entre a aplicação e as camadas inferiores da rede**



Comunicação entre Processos

- Cliente Socket – Servidor Socket
 - Cliente inicia a conexão
 - Servidor aguarda o pedido de conexão do cliente

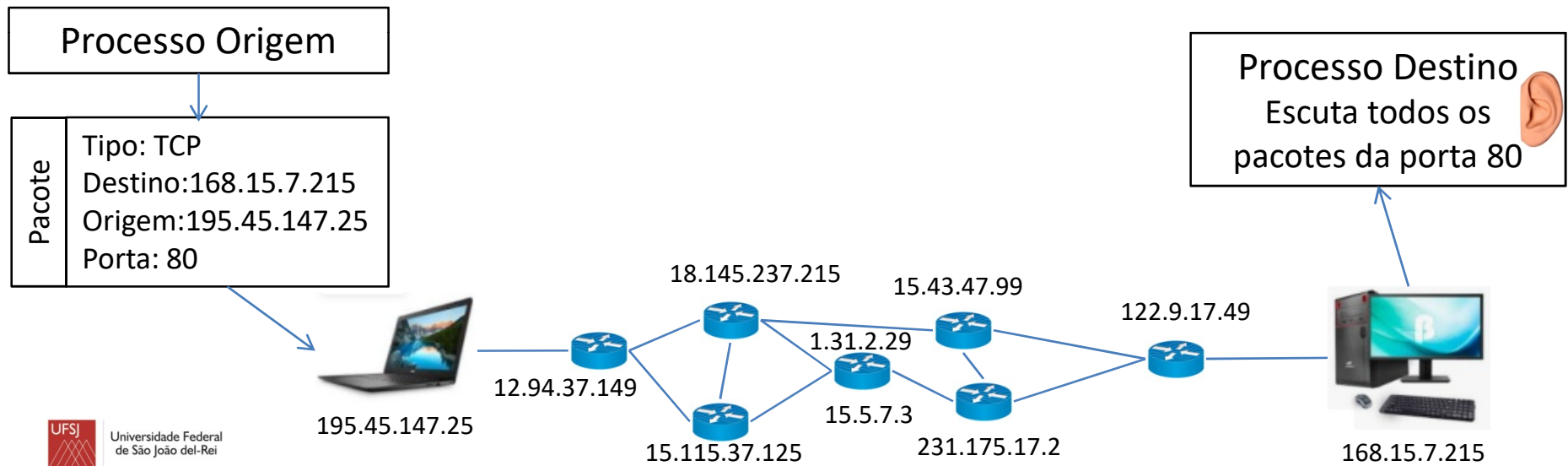


Comunicação entre Processos



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



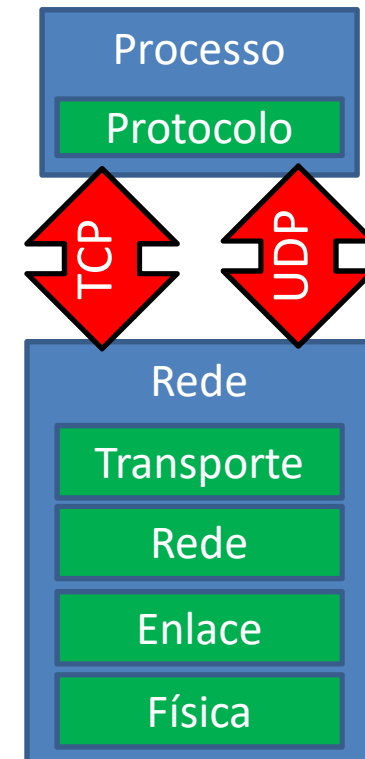
Serviços de transporte disponíveis para aplicações



Universidade Federal
de São João del-Rei

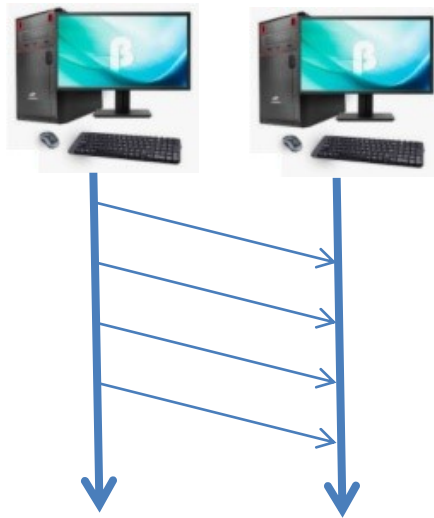
Serviços de transporte disponíveis para aplicações

- Serviço TCP:
 - Confiável
 - Orientado a conexão
 - Retransmite pacotes perdidos
- Serviço UDP:
 - Melhor esforço
 - Não orientado a conexão



Serviços de transporte disponíveis para aplicações

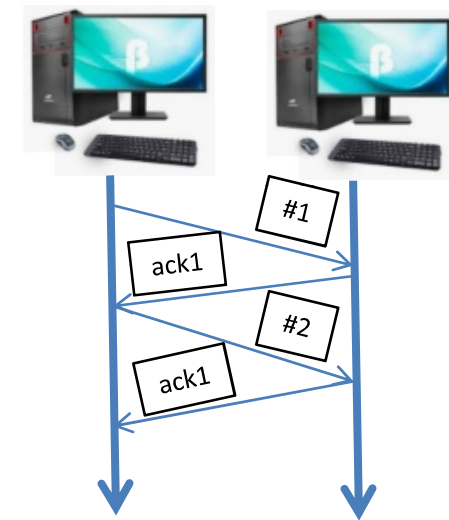
Transporte UDP



Fluxo contínuo

Simplificação do funcionamento dos protocolos

Transporte TCP



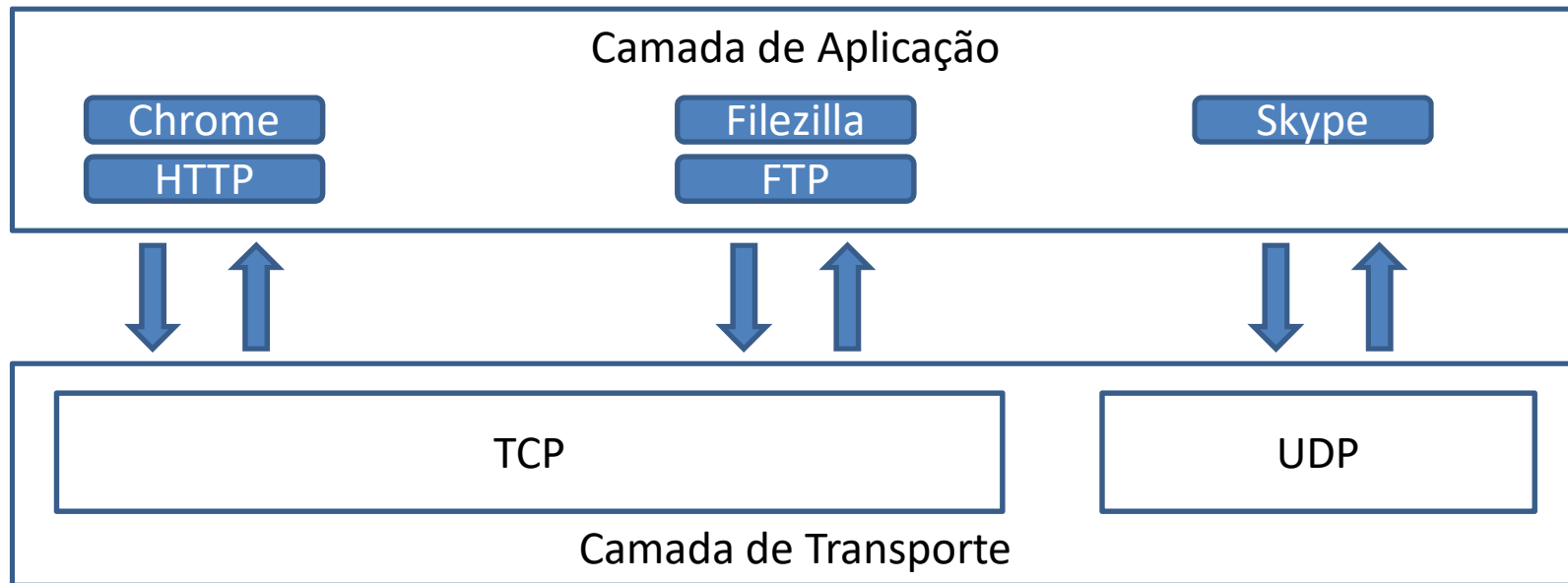
Requer confirmação

Serviços de transporte disponíveis para aplicações

- Como saber qual protocolo da camada de transporte utilizar?
- Analisar os serviços
 - Transferência confiável de dados
 - Vazão (garantia de vazão mínima disponível)
 - Temporização (exemplo: jogos)
 - Segurança
- Normalmente:
 - TCP: transmissão de dados
 - UDP: vídeo e áudio



Comunicação entre Processos



A Web e o HTTP



Universidade Federal
de São João del-Rei

A Web e o HTTP

- HTTP – Hypertext Transfer Protocol
- WEB
 - Aplicação composta de dois programas: cliente (navegador, como o Chrome, IE, Firefox) e servidor (servidor Web, como Apache, Tomcache e Jboss)
 - Transmissão de objetos: arquivo HTML, figura, vídeo, applet (programa em java), etc.
- Cada objeto possui uma url, ex:
http://www.ronepage.com/figuras/eu_sozinho_taua.JPG
- Utiliza TCP (transporte confiável)
- Protocolo sem estado (cliente – servidor)



A Web e o HTTP

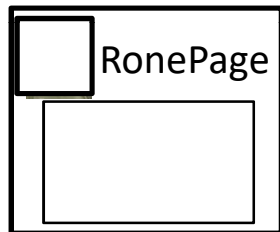
- Requisição/Resposta



Servidor



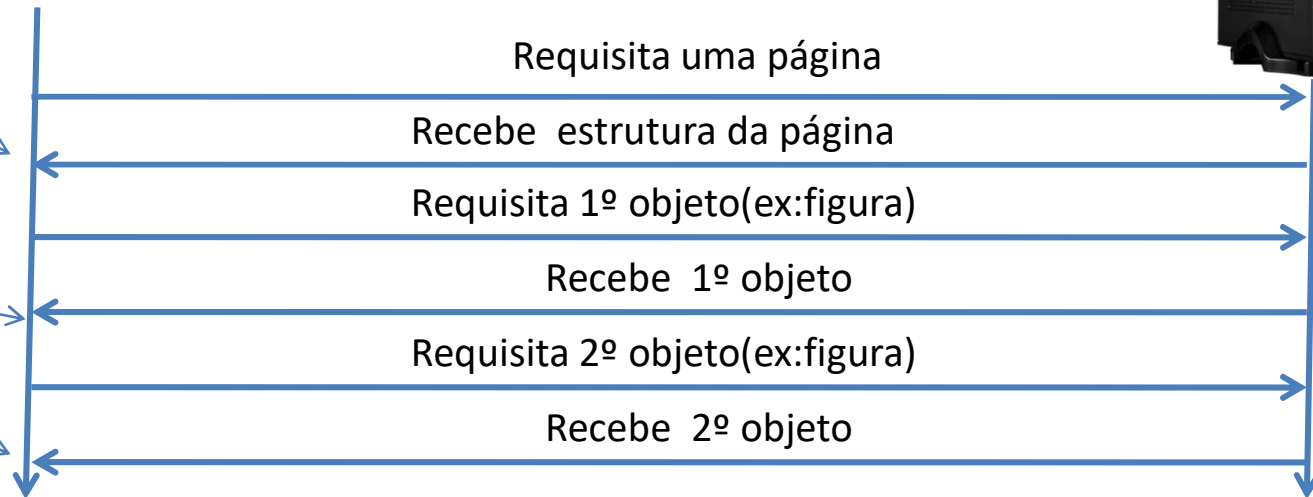
Cliente



Importante:
Aviso 1
Aviso 2



Universidade Federal
de São João del-Rei



Formato da Mensagem HTTP

- Dois tipos: requisição e resposta
- Exemplo de mensagem de requisição

```
GET / HTTP/1.1
Host: www.google.com:80
Connection: close
User-agent: Mozilla/4.0
Accept-language: pt-br
```
- Ao final de cada linha existe um \n, inclusive da última
- Veja socket que envia essa mensagem no próximo slide



```

import java.io.*;
import java.net.*;
import javax.swing.JOptionPane;
public class Google {
    public static void main(String args[]) {
        Socket socket = null; // Socket cliente
        PrintWriter out = null; // Escreve no socket
        BufferedReader in = null; // Le do socket
        try {
            socket = new Socket("www.google.com", 80);
            out = new PrintWriter(socket.getOutputStream(), true);
            in = new BufferedReader(new InputStreamReader(socket.getInputStream()));
            String entradaservidor = "";
            String entradausuario = "";
            entradausuario = "GET / HTTP/1.1\nHost: www.google.com:80\nConnection: close\nUser-agent: Mozilla/4.0\nAccept-
                language: pt-br\n";
            System.out.println("Requisição:\n" + entradausuario + "\n\n");
            out.println(entradausuario);
            while((entradaservidor = in.readLine()) != null) System.out.println(entradaservidor);
            out.close(); in.close(); socket.close();
        } catch (IOException e) {
            System.err.println("Erro na criação dos objetos");
            System.exit(1);
        }
    }
}

```



Formato da Mensagem HTTP

- 1ª linha:
 - GET: requisição
 - /: local onde se encontra, no caso a pasta raiz
 - HTTP/1.1: versão do protocolo
- 2ª linha → Host: url
- 3ª linha → User-agent: quem está pedindo (navegador)
- 4ª linha → Connection: close (conexões não persistentes)
- 5ª linha → Accept-language: linguagem esperada



HTTP/1.1 200 OK

Date: Wed, 23 May 2018 01:02:49 GMT

Expires: -1

Cache-Control: private, max-age=0

Content-Type: text/html; charset=ISO-8859-1

P3P: CP="This is not a P3P policy! See g.co/p3phelp for more info."

Server: gws

X-XSS-Protection: 1; mode=block

X-Frame-Options: SAMEORIGIN

Set-Cookie: 1P_JAR=2018-05-23-01; expires=Fri, 22-Jun-2018 01:02:49 GMT; path=/; domain=.google.com

Set-Cookie: NID=130=AHFZnwdNa7sGrhVCdezdweBj--

wmR9A_AccuGRhkwayHSu6cHTiapBf0cJylouSmf7w9zWh6qRGNH1lQHEws8-

zbtY7GF62EIZRmDTtj8VpHcSRPKv4XJDhTrr1vqR5y; expires=Thu, 22-Nov-2018 01:02:49 GMT; path=/;

domain=.google.com; HttpOnly

Accept-Ranges: none

Vary: Accept-Encoding

Connection: close

...Dados da página ...

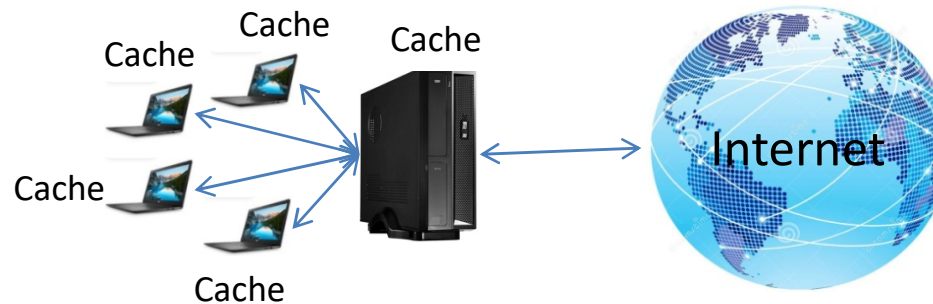
Formato da Resposta HTTP



Universidade Federal
de São João del-Rei

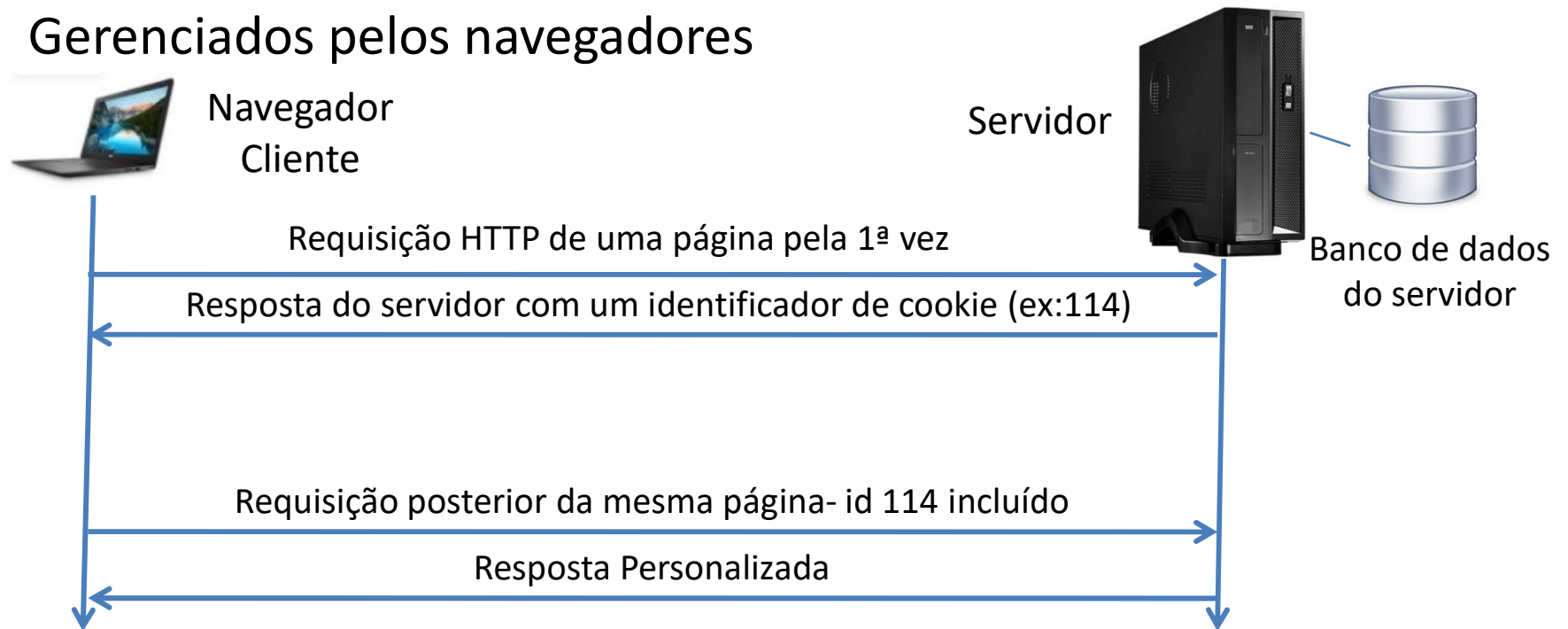
Caches Web

- Web cache (também chamado de servidor proxy) armazena localmente objetos HTTP
- Pode ser local ou em um computador que faz a interface entre as redes interna e externa
- Quando o usuário requisita um objeto, se ele estiver no cache o web cache responde, sem utilizar a rede externa



Interação usuário-servidor:cookies

- Cookies: informações de identificação de usuários de sites
- Gerenciados pelos navegadores



Transferência FTP



Universidade Federal
de São João del-Rei

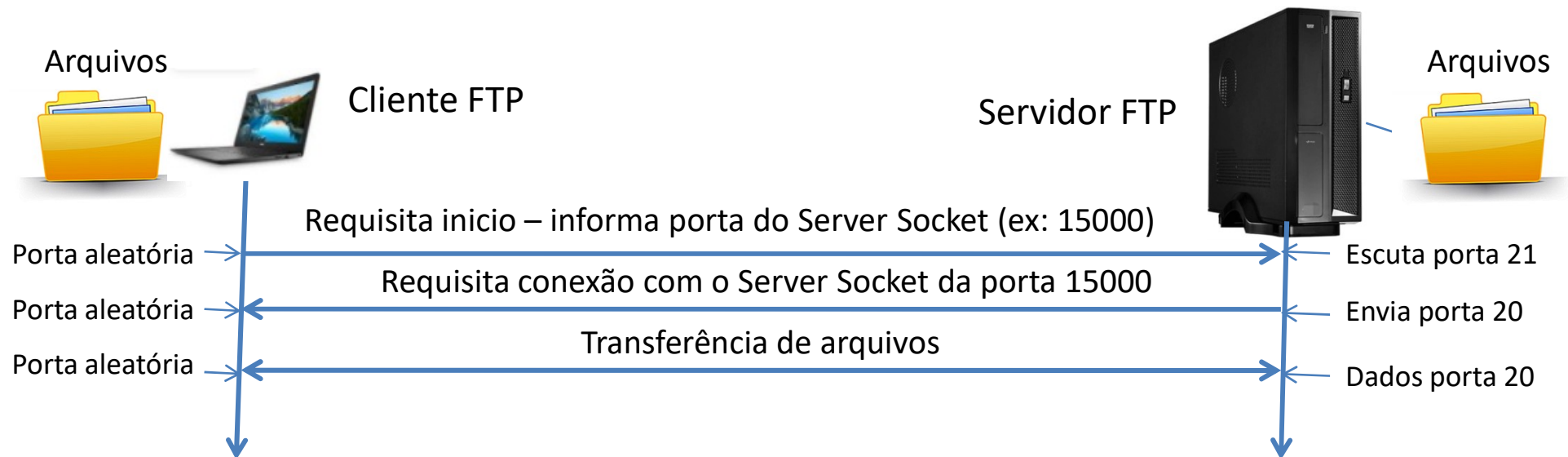
Transferência FTP



- FTP – File Transfer Protocol
- Abre duas conexões TCP:
 - Porta 21: canal de controle, se mantém ativo durante toda a transmissão
 - Porta 20: canal de dados, criado para a transferência de arquivos
- Por linha de comando ou por programa específico (BareFTP, Filezilla)
- SFTP: com criptografia

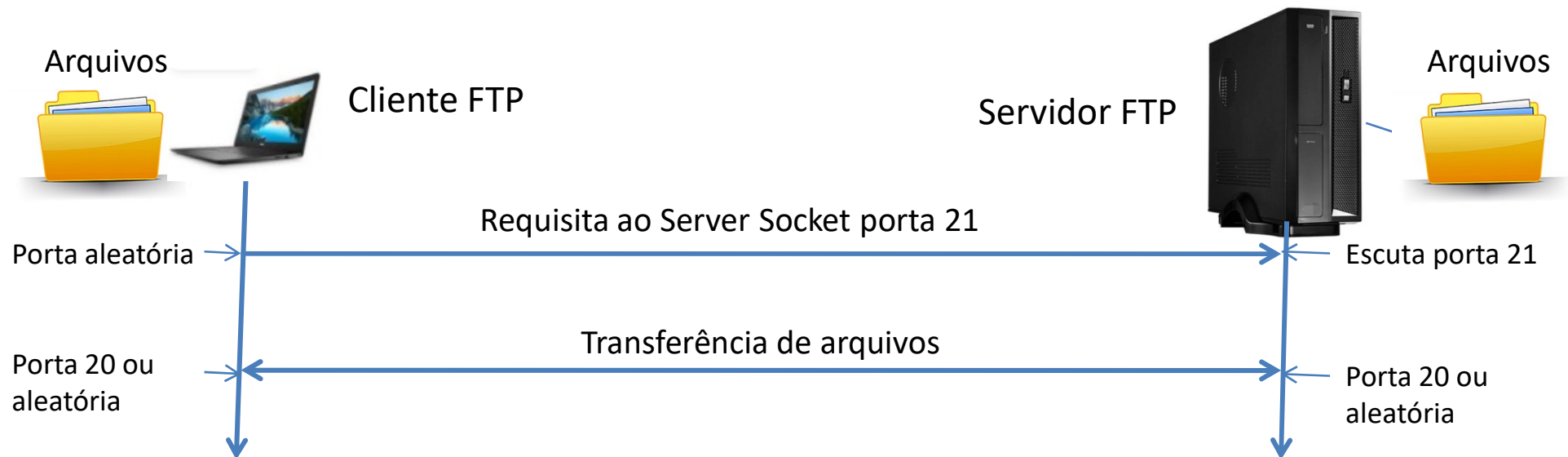
FTP Ativo

- O servidor socket fica no cliente
- Problema: cliente protegido por Firewall ou com NAT



FTP Passivo

- FTP passivo → Mais utilizado
- Servidor socket no Servidor FTP



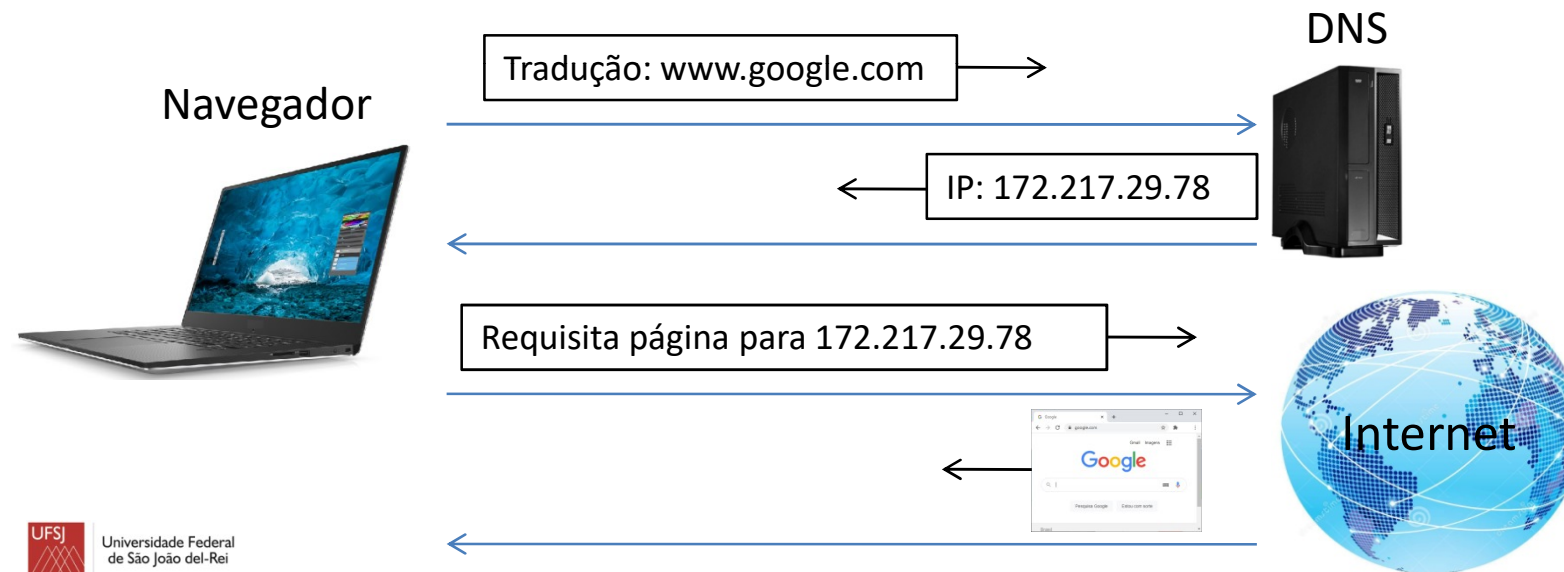
DNS



Universidade Federal
de São João del-Rei

DNS

- Domain Name System – protocolo + banco de dados distribuído
- Principal serviço: tradução URL <--> IP



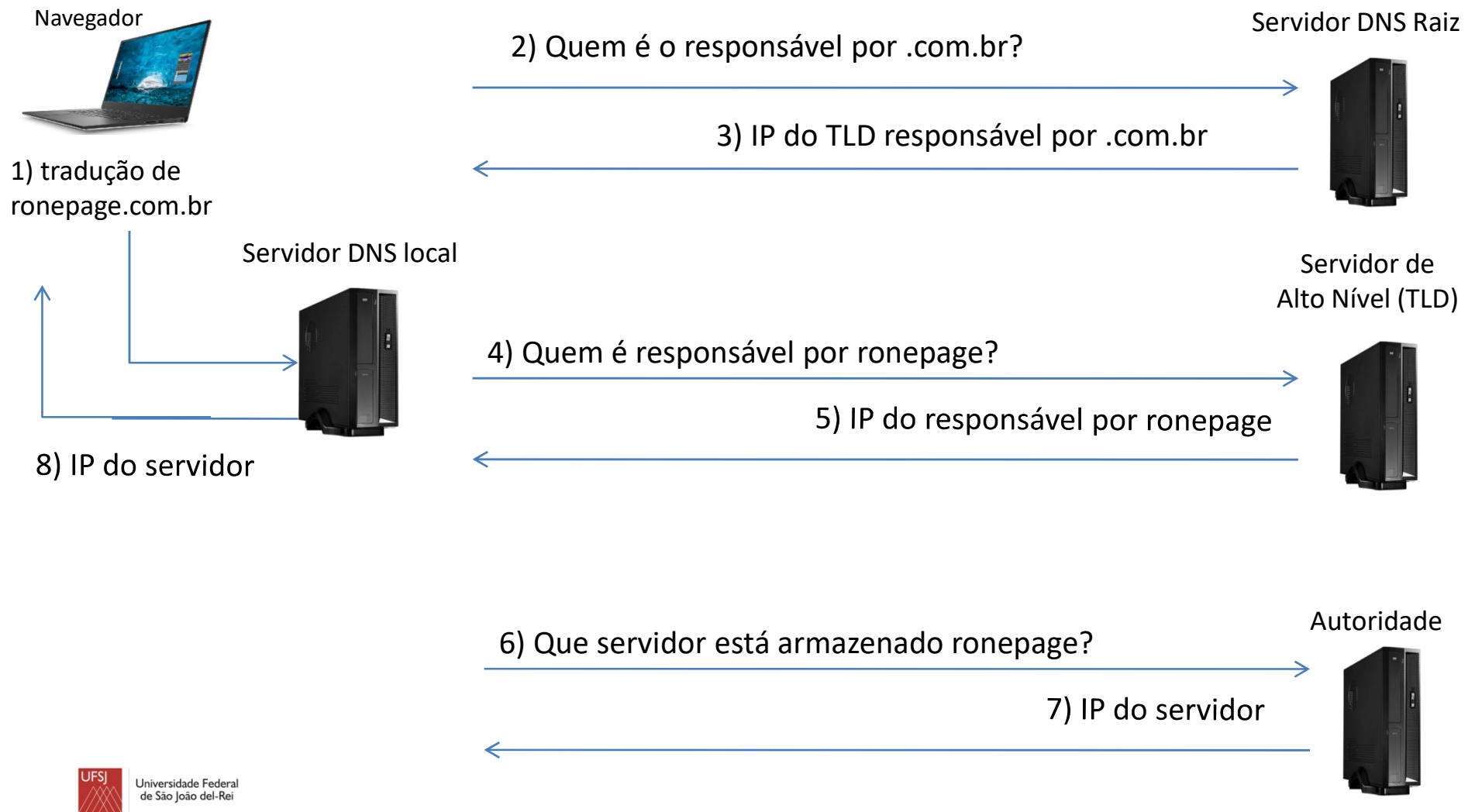
DNS

- Outros serviços:
 - Tradução de apelido para nome real (nome canônico ou cname), tanto de servidores web como servidores de e-mail
 - Ex: de *hotmail.com* para *relay1.west-coast.hotmail.com*
 - Distribuição de carga
 - Ex: o servidor *google.com* possui vários endereços IP de servidores espelhados

DNS

- Funcionamento hierárquico
- Três classe de servidores de nome
 - Servidores de nomes raiz
 - Servidores de domínio de alto nível (TLD – Top Level Domain)
 - Servidores com autoridade
- Servidores locais (configurados em cada máquina ou rede local)

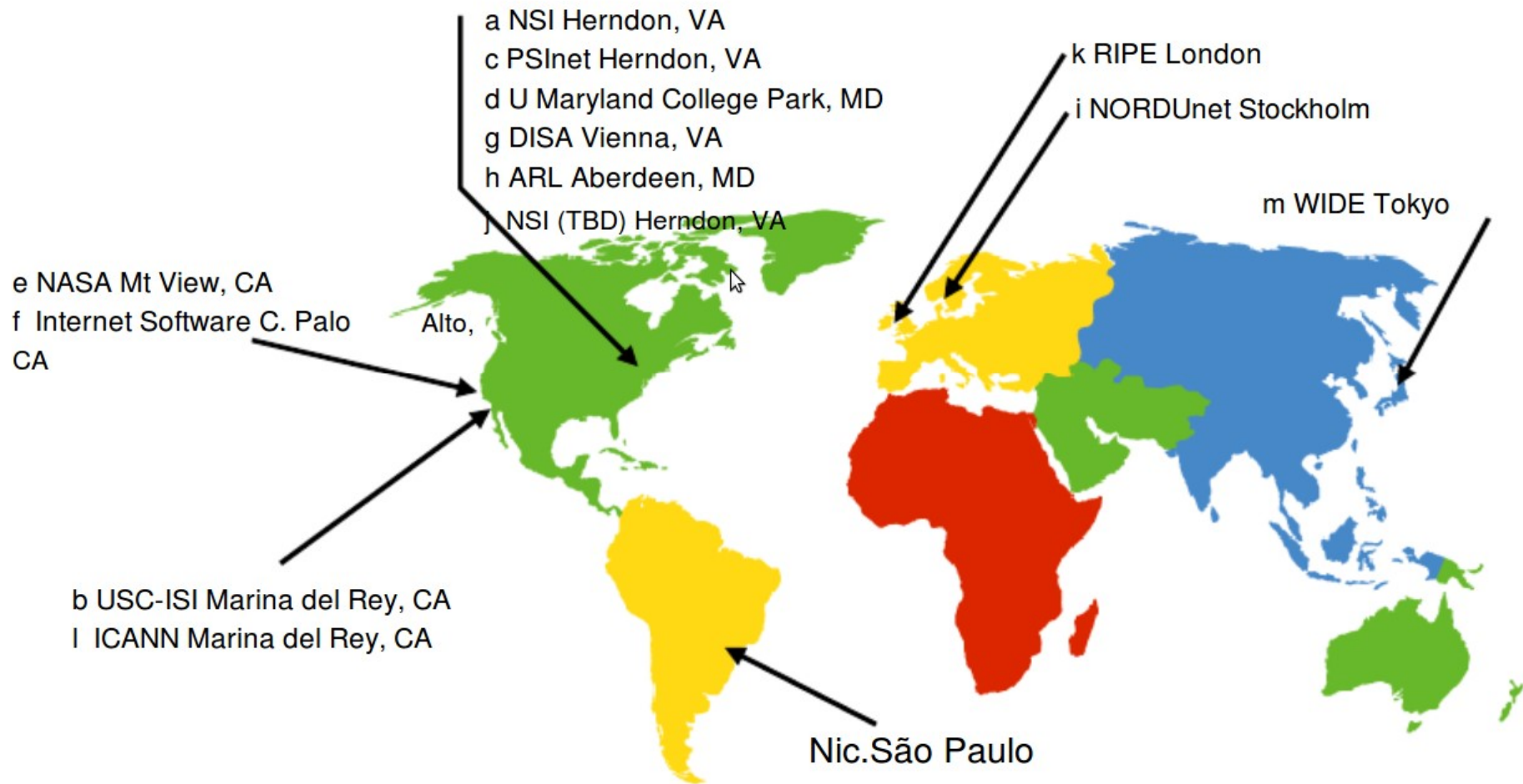




Servidores DNS Raiz

- Primeiros a serem acessados
- Direcionam a requisição para o DNS de alto nível específico
- São 13, identificados por letras de A a M
 - A maioria nos EUA
 - Lista de servidores raiz pode ser encontrada em <https://pt.wikipedia.org/wiki/Servidor-raiz>
- Obs: existem réplicas espalhadas pelo mundo





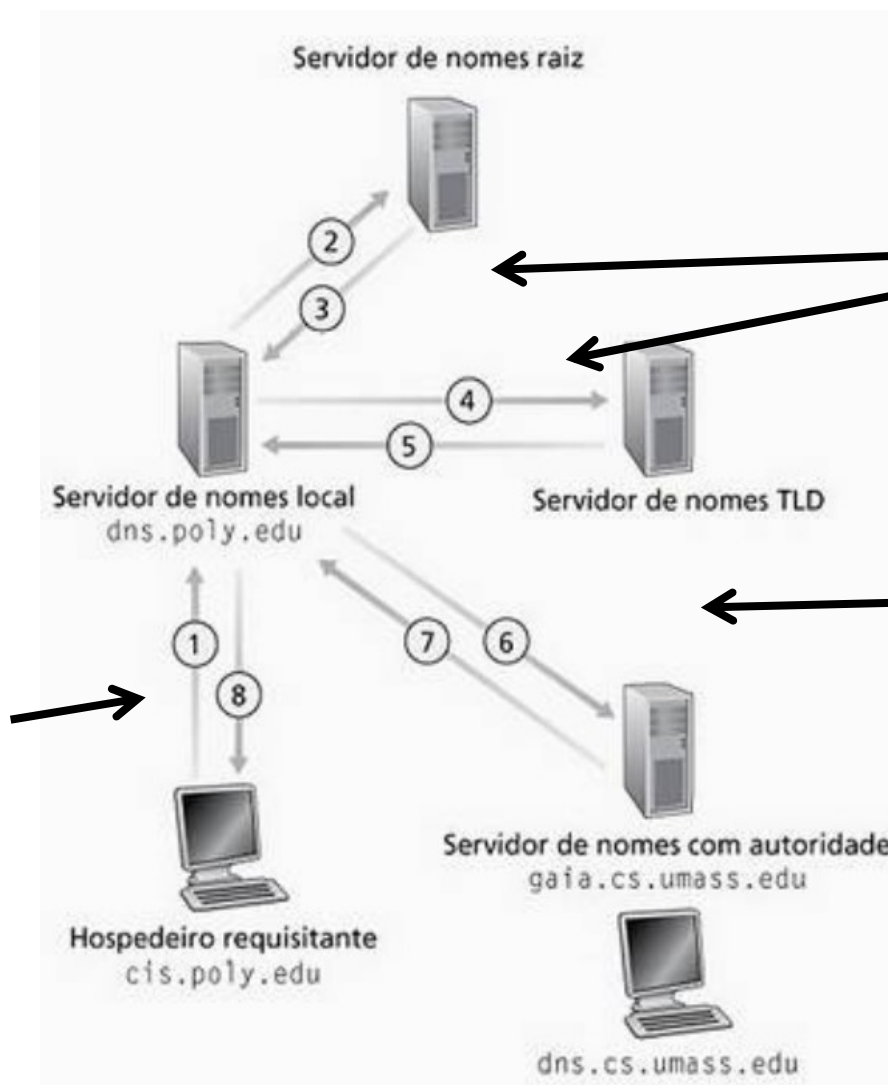
Servidores DNS de Alto Nível

- Responsáveis pelos domínios de alto nível, como com, org, edu e gov, e por todos os domínios de países, como br, uk, jp e ca.
- Também chamados TLD – Top Level Domain

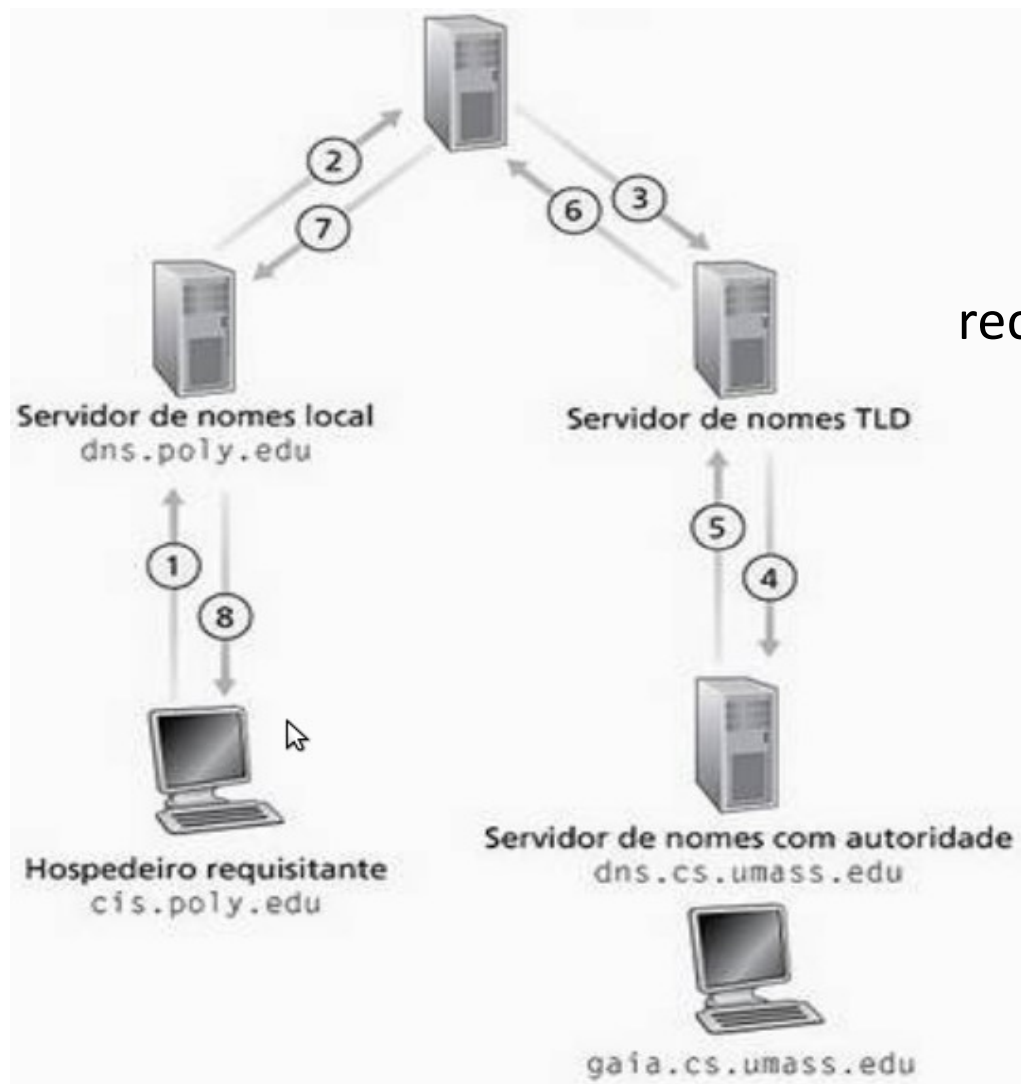
Servidores DNS de Autoridade

- Estão em servidores que podem ser acessados publicamente, junto com os servidores Web
- Responsáveis pelos hosts que estiverem abaixo dele na hierarquia na empresa de hospedagem

Consulta
Recursiva



Consulta
Iterativas



Utilização somente de consultas recursivas --> não utilizado na prática

Cache DNS

- Instalado em qualquer servidor
- Armazena os valores IP/URL por um período de tempo (dias)
- Primeiro acessa o cache para depois acessar os demais
- Cache de IPs de servidores de alto nível

