

# Camada de Aplicação

Redes Industriais

Prof. Rone Ilídio

# 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
    - 2.1.3 Serviços de transporte disponíveis para a aplicação
    - 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
    - 2.2.5 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

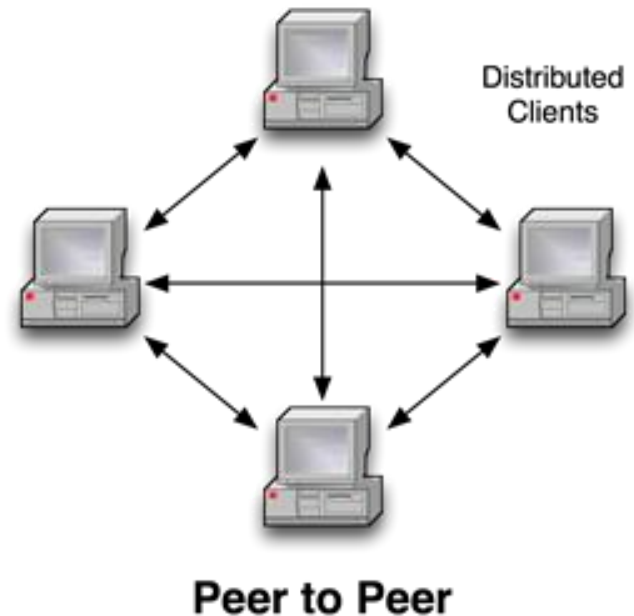
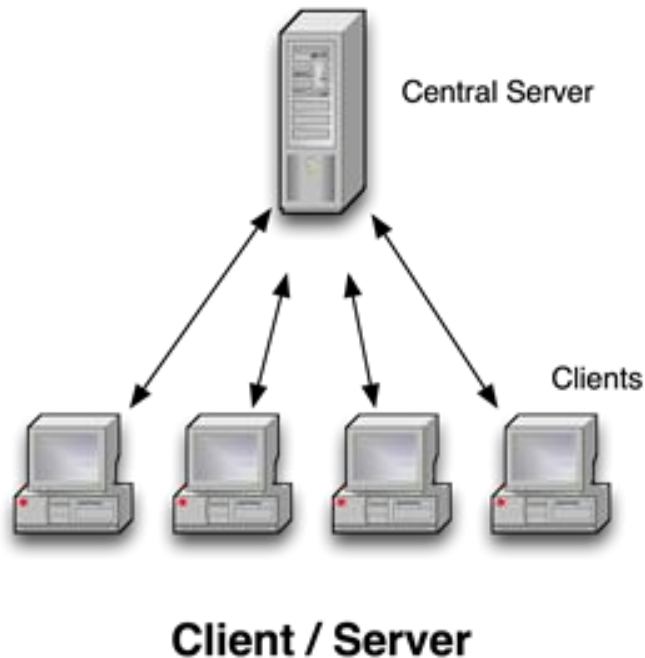
# Introdução

- Protocolos que fornecem os serviços utilizados pelos usuários
- Exemplos de protocolos da Camada de Aplicação:
  - HTTP: web
  - FTP: transferência de arquivos
  - SMTP: e-mail
  - DNS
  - IPTV

# 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

# Arquiteturas de aplicação de rede



# Serviços de transporte disponíveis para a aplicação

- 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

# Protocolos da Camada de Aplicação

- Definem como 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

# 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](http://www.ronepage.com/figuras/eu_sozinho_taua.JPG)
- Utiliza TCP (transporte confiável)
- Protocolo sem estado (cliente – servidor)
- Toda comunicação é definida pelo HTTP

# 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: fr
```
- 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 ClienteTCP {
    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: fr\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

# Formato da Mensagem HTTP

- Resposta

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=AHFZnwdNa7sGrhVCdezdwBj--  
wmR9A\_AccuGRhkwayHSu6cHTiapBf0cJylouSmf7w9zWh6qRGNH1IQHEws8-  
zbtY7GF62ElZRmDTtj8VpHcSRPKv4XJDhTrr1vqR5y; expires=Thu, 22-Nov-2018 01:02:49  
GMT; path=/; domain=.google.com; HttpOnly

Accept-Ranges: none

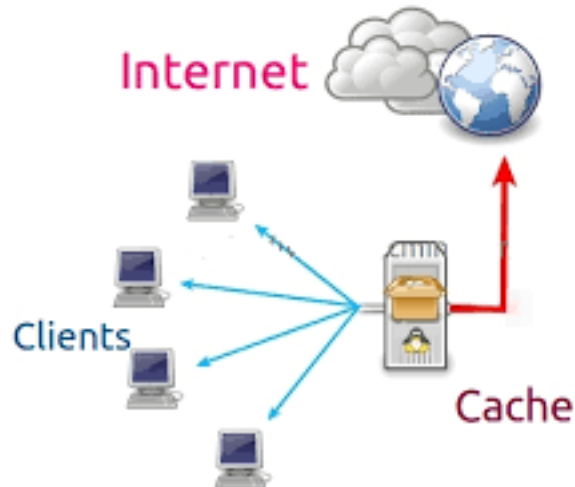
Vary: Accept-Encoding

Connection: close

...Dados da página ...

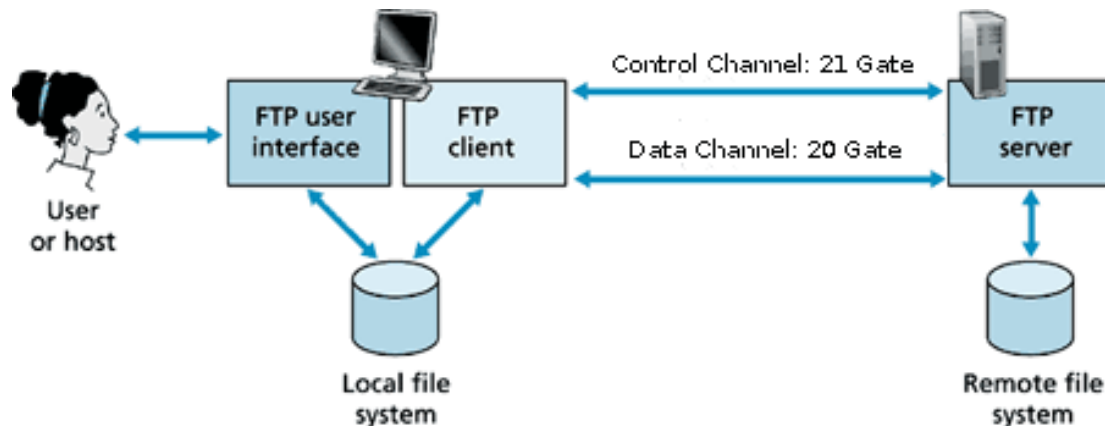
# Caches Web

- 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



# 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
- Autenticação com usuário e senha
- Por linha de comando ou por programa específico (BareFTP, Filezilla)
- Versão criptografada: SFTP

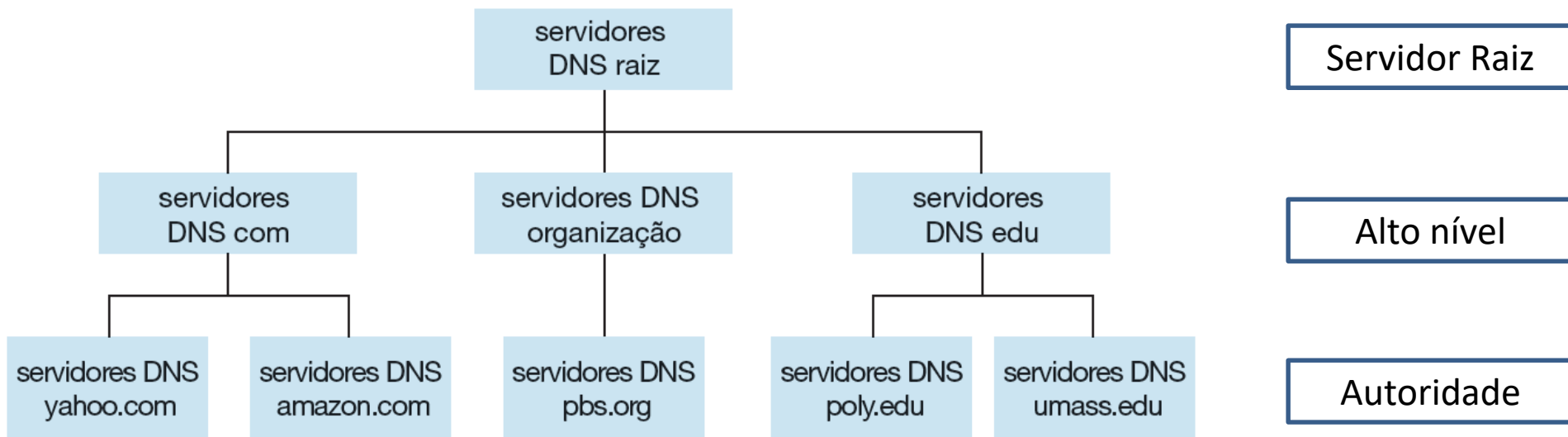


# DNS

- Domain Name System – sistema de nomes de domínio
- Protocolo da camada de aplicação e um banco de dados distribuído
- Tradução do nome do host ou URL para o IP
  - URL - Uniform Resource Locator
  - Nome canônico ou cname – nome real de um host
  - Apelido do hospedeiro – simplifica o cname
- Distribuição de cargas: existem sites com vários servidores espelhados

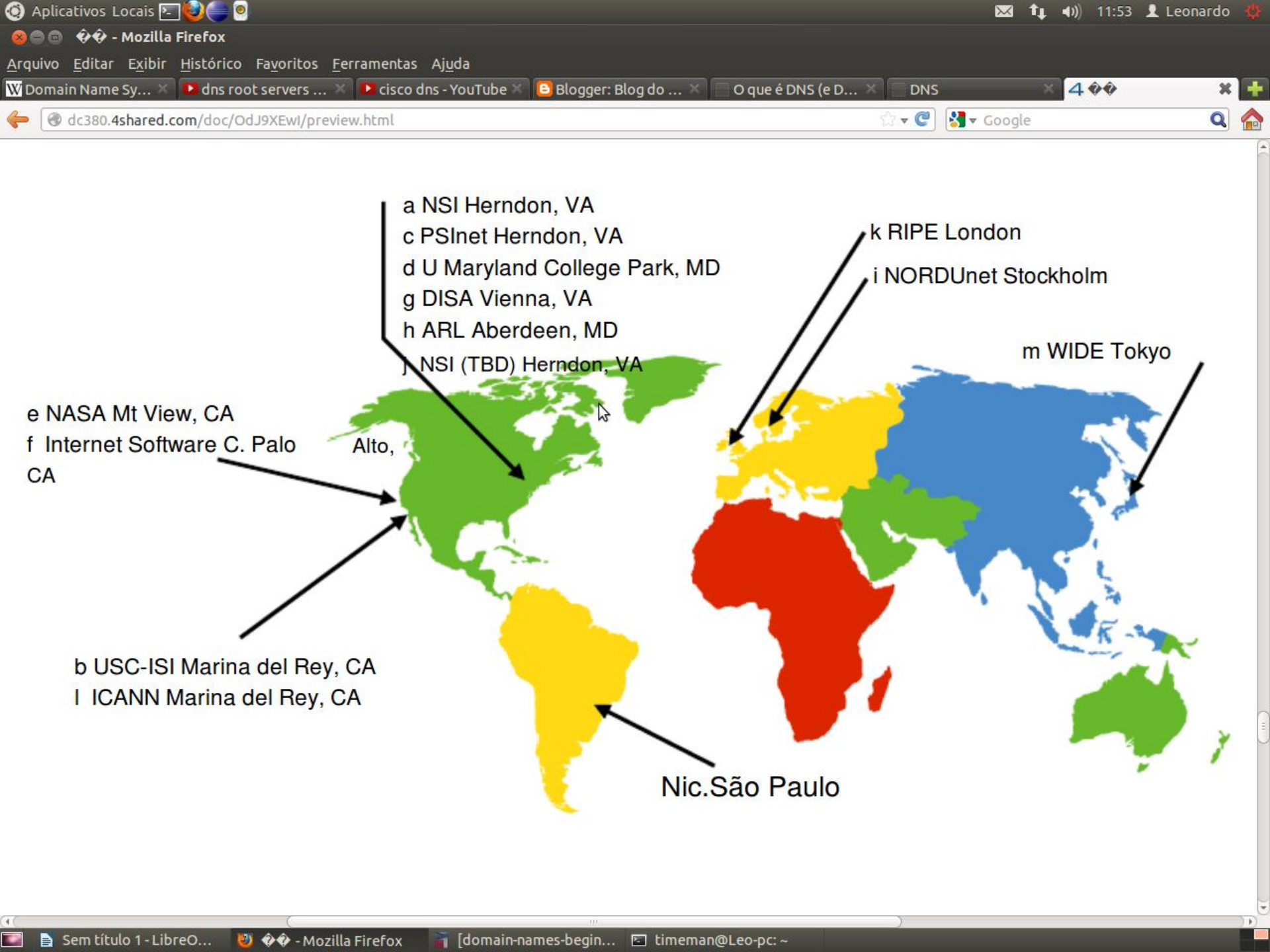
# DNS

- Funcionamento hierárquico
- Três classe de servidores de nome
  - Servidores de nomes raiz
  - Servidores de domínio de alto nível
  - Servidores com autoridade



# 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>



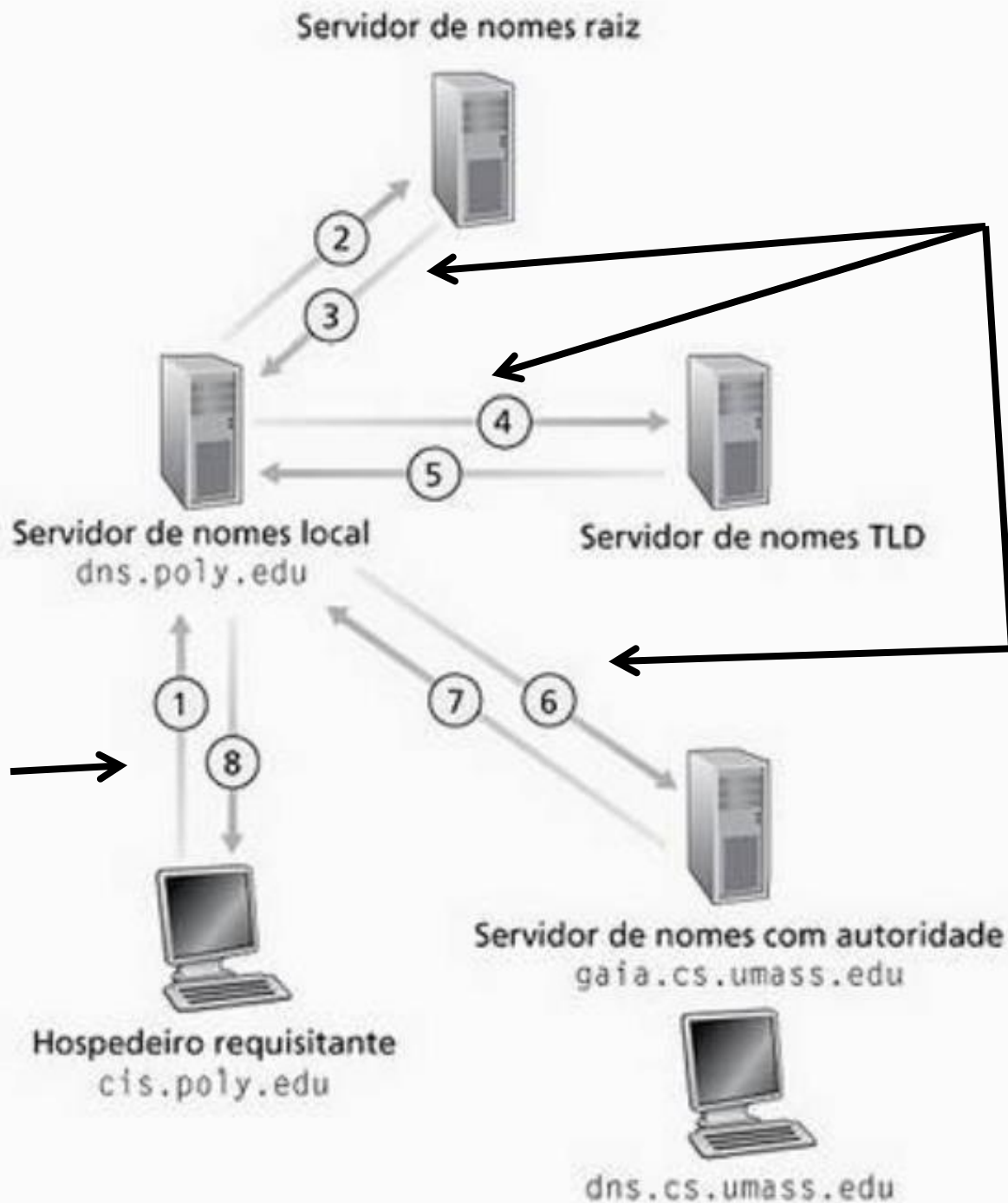
# Servidores DNS de Alto Nível

- 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

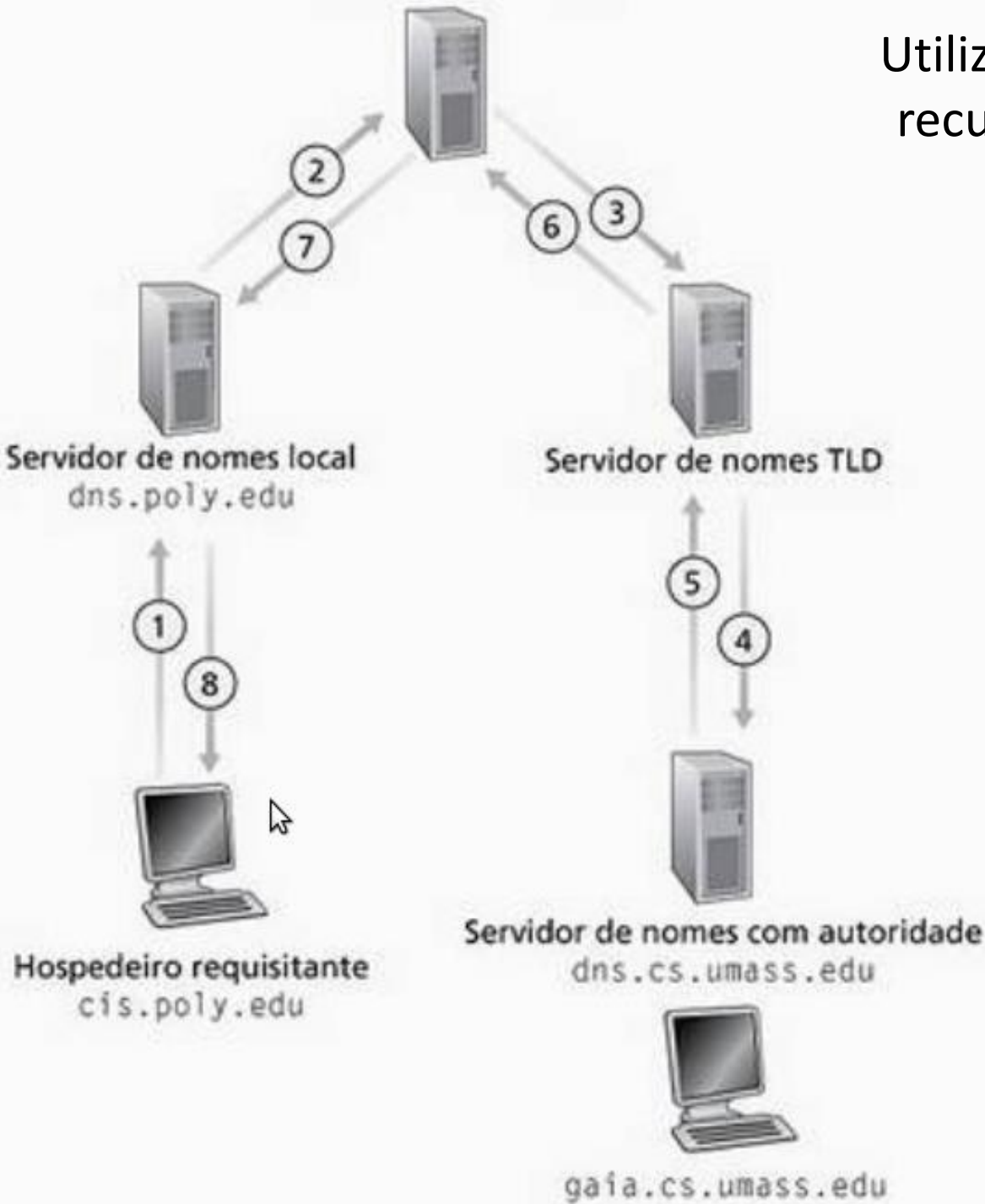
- Estão em servidores que podem ser acessados publicamente, com os servidores Web
- Responsáveis pelos hosts que estiverem abaixo dele na hierarquia

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