

## T02 – A/B/C Redes de Dados I

### 03 – Camada de Aplicação (DNS - HTTP)

Prof. Edson J. C. Gimenez  
*soned@inatel.br*

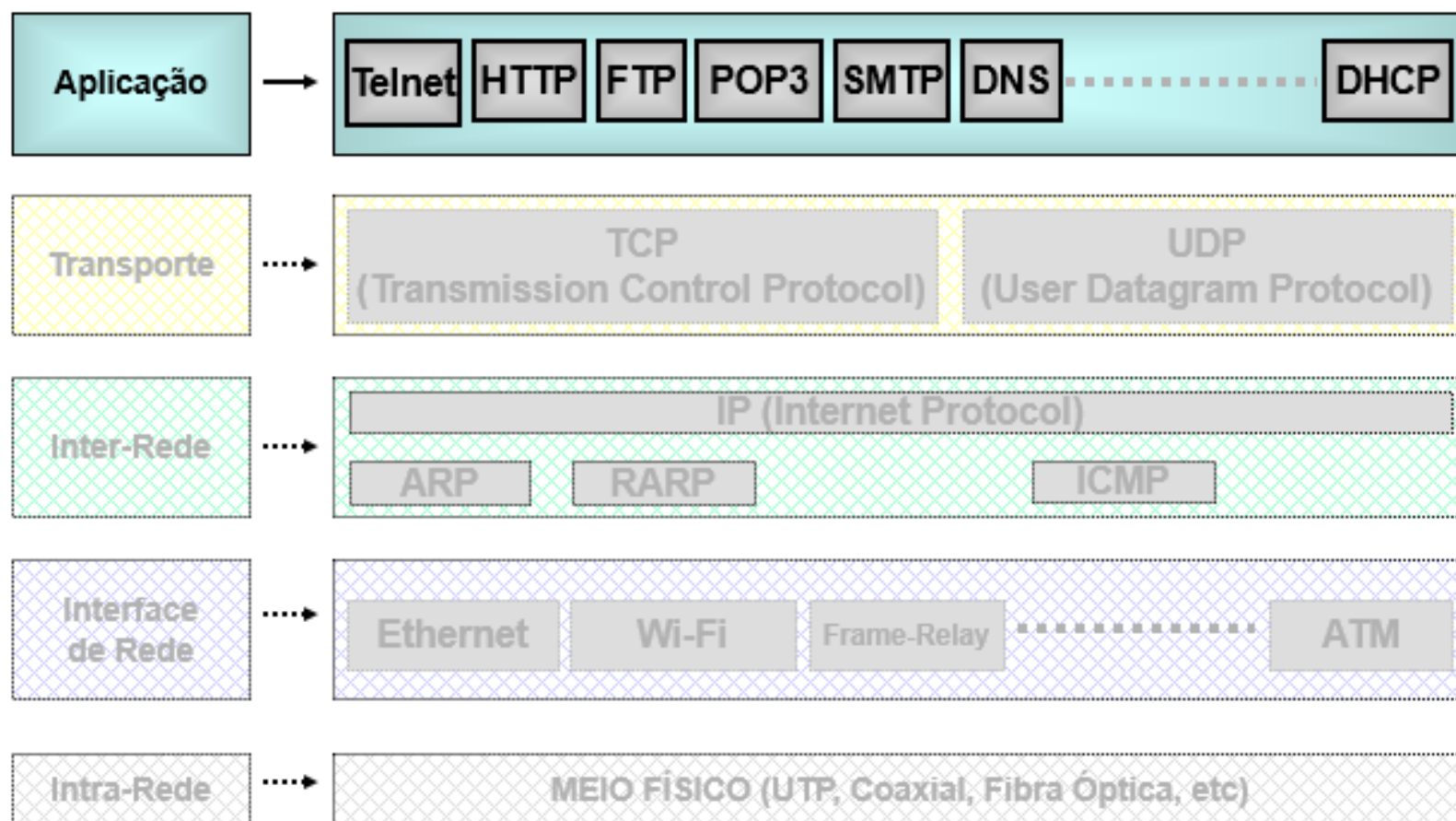
## Referências:

- ✓ Kurose & Ross. Redes de Computadores e a Internet: uma abordagem top-down. Capítulo 2.
- ✓ Tanenbaum & Wetherall. Redes de Computadores; capítulo 7.
- ✓ Farrel. A Internet e seu Protocolos: uma Análise Comparativa; capítulo 12.
- ✓ Forouzan & Mosharraf. Redes de Computadores: uma abordagem top-down; capítulo 2.
- ✓ Comer. Interligação de Redes com TCP/IP; volume 1; capítulos 23 e 27.

- ✓ A camada de aplicação da Internet contém os aplicativos que são disponibilizados para os usuários, e que utilizam os protocolos subjacentes (TCP ou UDP, IP e camada 2) para trocar informações.
- ✓ Assim, um aplicativo interage com pelo menos um dos protocolos da camada de transporte (TCP ou UDP) para enviar e/ou receber dados.
- ✓ Cada aplicativo escolhe o serviço de transporte necessário, UDP ou TCP, de acordo com as suas necessidades.

- ✓ Definem como os processos de uma aplicação, em sistemas finais diferentes, trocam mensagens entre si, estabelecendo:
  - Os tipos de mensagens trocadas.
  - A sintaxe dessas mensagens.
  - Regras para envio e resposta dessas mensagens.

✓ Exemplos:



## DNS (Domain Name System)

```
PS C:\Users\edsonjcg> ipconfig /flushdns
```

Configuração de IP do Windows

Liberação do Cache do DNS Resolver bem-sucedida.

```
PS C:\Users\edsonjcg> ipconfig /displaydns
```

```
ns1.inatel.br
```

```
-----
Nome do Registro. . . . . : ns1.inatel.br
Tipo de Registro. . . . . : 1
Tempo de Vida . . . . . : 7613
Comprimento dos Dados . . . . . : 4
Seção. . . . . : Resposta
Registro (Host). . . . . : 131.221.240.5
```

```
outlook.office365.com
```

```
-----
Nome do Registro. . . . . : CPQ-efz.ms-acdc.office.com
Tipo de Registro. . . . . : 1
Tempo de Vida . . . . . : 4349
Comprimento dos Dados . . . . . : 4
Seção. . . . . : Resposta
Registro (Host). . . . . : 52.97.19.146
```

### DNS (Domain Name System).

- ✓ Identificar o endereço IP corresponde a um nome de domínio específico.

```
PS C:\Users\edsonjcg> nslookup
Servidor Padrão:  DC2-SRV.local.inatel.br
Address:  10.100.0.12

> inatel.br
Servidor:  DC2-SRV.local.inatel.br
Address:  10.100.0.12

Nome:      inatel.br
Address:   119.8.151.60

> uol.com.br
Servidor:  DC2-SRV.local.inatel.br
Address:  10.100.0.12

Não é resposta autoritativa:
Nome:      uol.com.br
Addresses: 2804:49c:3102:401:ffff:ffff:ffff:36
           2804:49c:3101:401:ffff:ffff:ffff:45
           200.147.3.157
```

- ✓ Definido nas RFCs 1034 e 1035 (nov/1987), além de diversas atualizações.
- ✓ Faz uso da porta UDP 53, na camada de transporte.

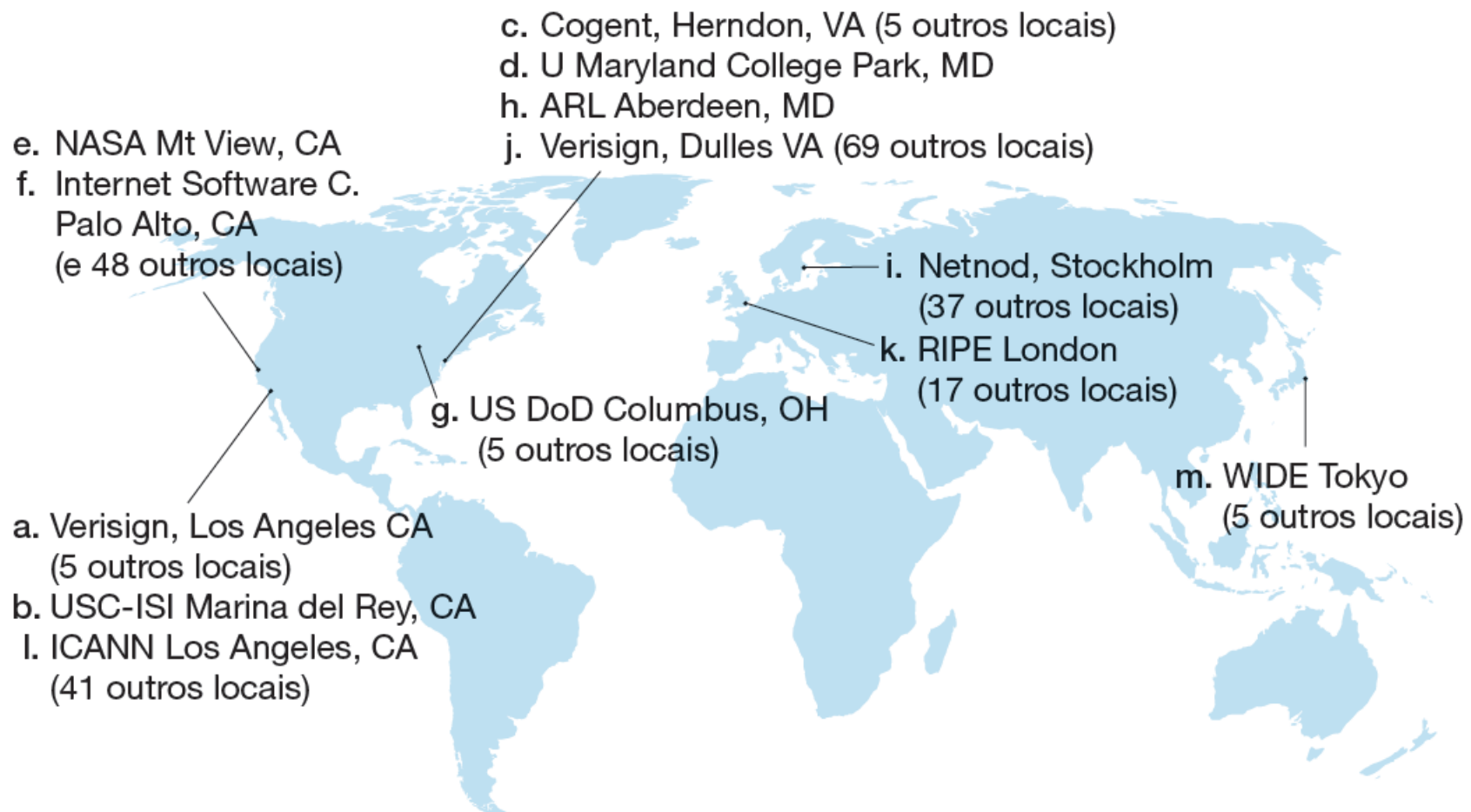
DNS (Domain Name System).

- ✓ Há duas maneiras de identificar um hospedeiro: nome e endereço.
- ✓ Exemplos: inatel.br → 119.8.151.60  
uol.com.br → 200.147.3.157
- ✓ Para conciliar isso, é necessário um serviço de diretório que **traduza nomes de hospedeiro para endereços IP**.
- ✓ Esta é a tarefa principal do DNS.
- ✓ Assim, o serviço de diretório (DNS) pode ser visto como:
  - 1) Um banco de dados distribuído, executado em uma hierarquia de servidores: raiz, TLD (auto nível) e autoritativos.
  - 2) Um protocolo de camada de aplicação que permite que hospedeiros consultem esse banco de dados distribuído.



### Servidores DNS raiz.

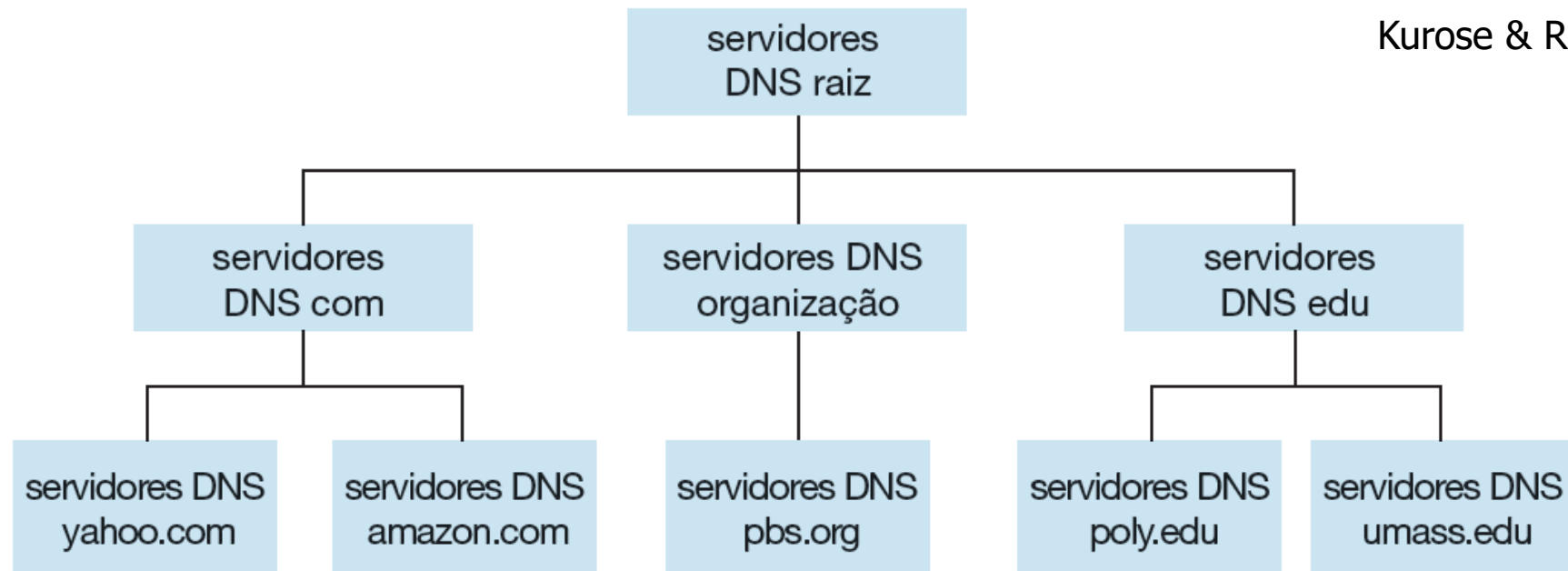
- ✓ Na Internet há 13 servidores DNS raiz (denominados de A à M), a maior parte deles localizados na América do Norte.
  - Na realidade, cada um é um conglomerado de servidores replicados, para fins de segurança e confiabilidade.





- ✓ Servidores DNS de Domínio de Alto Nível (TLD).
  - São responsáveis por domínios de alto nível como *com*, *org*, *net*, *edu*, *gov*, e por todos os domínios de alto nível de países (*uk*, *fr*, *br*, etc.)
- ✓ Servidores DNS autoritativos.
  - São responsáveis por abrigar os registros DNS acessíveis publicamente de uma organização

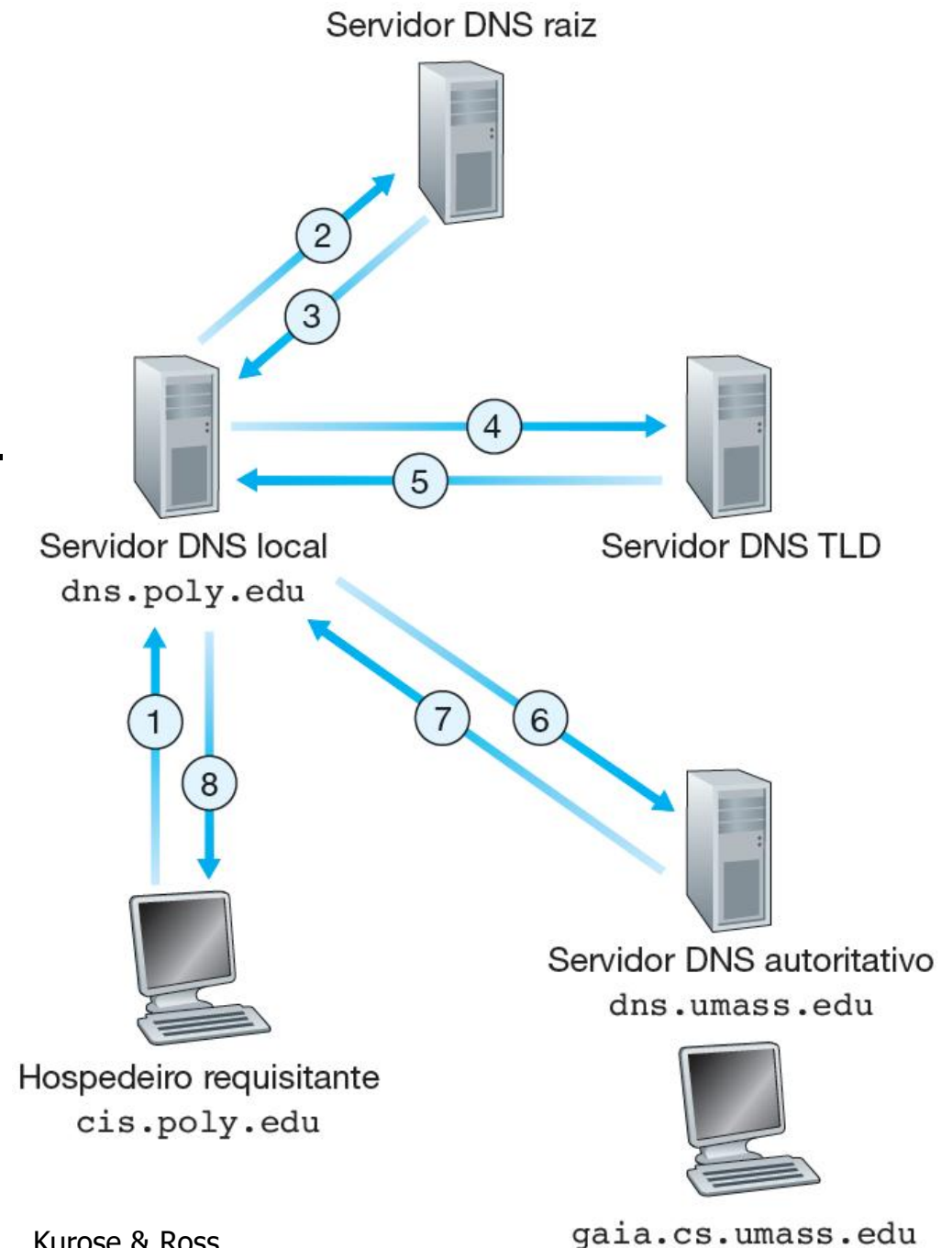
Exemplo hierarquia de servidores DNS:



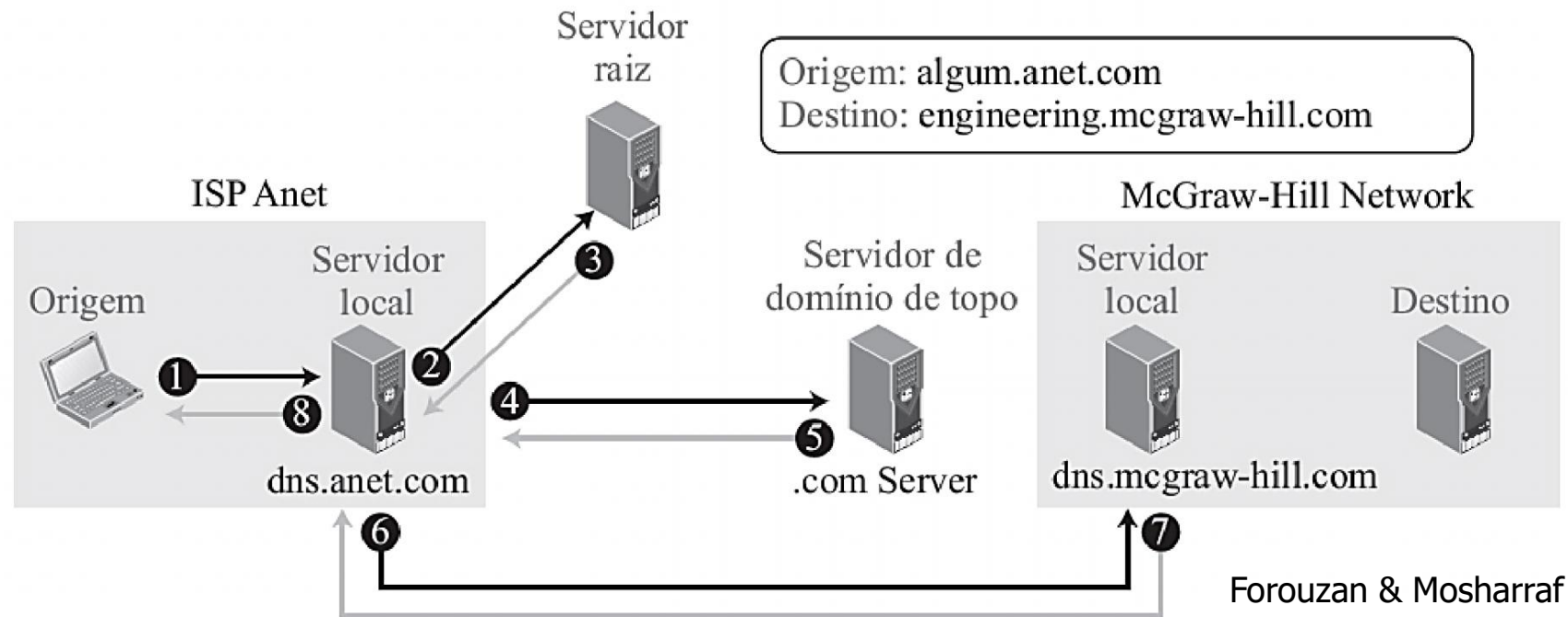
Obs.: Há ainda os servidores DNS locais, que não pertencem propriamente à hierarquia de servidores DNS

**Operação básica:** cis.poly.edu deseja obter o end. IP de gaia.cs.umass.edu:

- (1) cis.poly.edu consulta seu servidor local (dns.poly.edu).
- (2) dns.poly.edu consulta o servidor raiz, que (3) retorna uma lista de servidores TLD responsáveis por .edu.
- (4) dns.poly.edu consulta um servidor TLD responsável por .edu, que (5) retorna o endereço IP de um servidor autoritativo do sufixo umass.edu.
- (6) dns.poly.edu consulta o servidor dns.umass.edu, que (7) responde com o endereço IP de gaia.cs.umass.edu.
- (8) dns.poly.edu envia ao cis.poly.edu o endereço IP solicitado, e armazena uma cópia para futuras solicitações.



### ✓ Exemplo de resolução DNS:



### Protocolo de Transferência de Hipertexto.

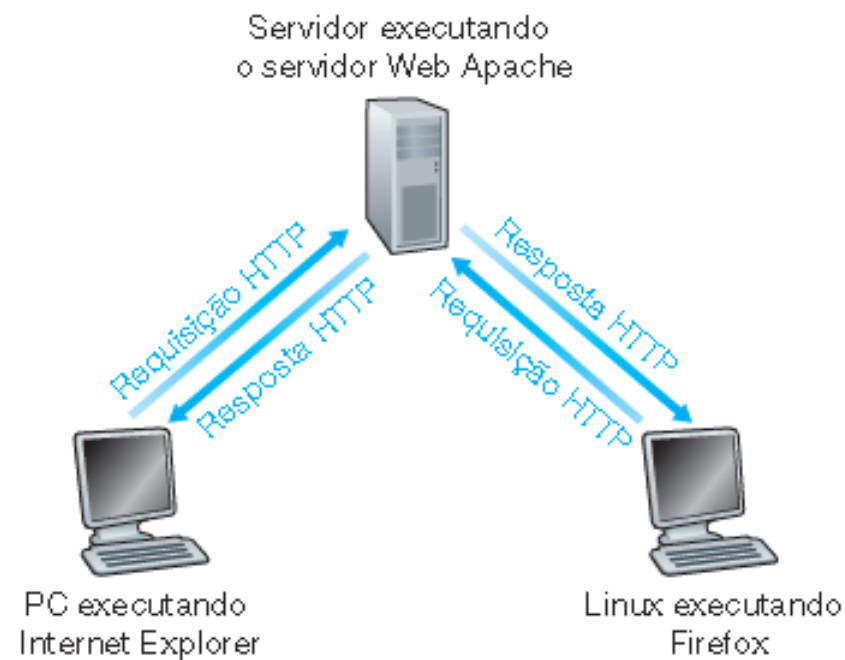
- HTTP/1.0 (RFC 1945, May/1996)
- HTTP/1.1 (RFC 2616, June/1999).
- HTTP/2 (RFC 7540, May/2015)
- \*\* HTTP/3 (RFC 9114, June/2022) → QUIC (Quick UDP Internet Connections)

### ✓ Protocolo padrão para comunicação Web

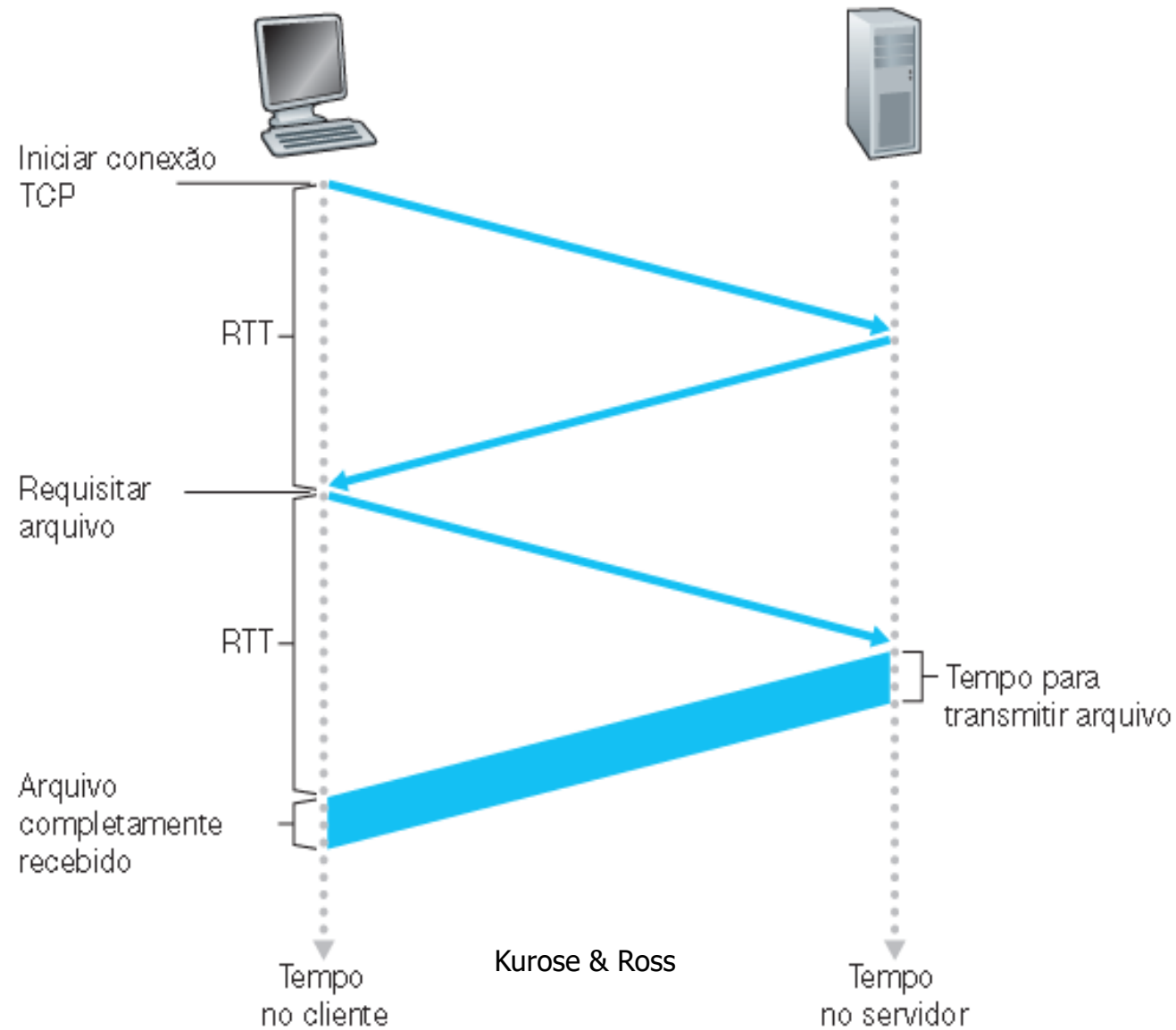
### ✓ Define como as páginas Web são requisitadas pelos clientes e enviadas pelos servidores.

- Em geral, as páginas Web são constituídas de um arquivo-base HTML e diversos objetos referenciados.
- Cada objeto é um arquivo, tal como arquivo html, imagem jpeg, arquivo de vídeo, etc., que se pode acessar com um único URL (Uniform Resource Locator).
- Cada URL tem dois componentes básicos: o **nome de hospedeiro** (hostname) do servidor que abriga o objeto e o **nome do caminho** do objeto.
  - <http://www.someschool.edu/somedepartment/picture.gif>
  - <http://cisco.com/index.html>

- ✓ Utiliza o TCP como seu protocolo de transporte.
- ✓ Trabalha no modo cliente / servidor.
  - Cliente envia requisições, servidor responde às requisições



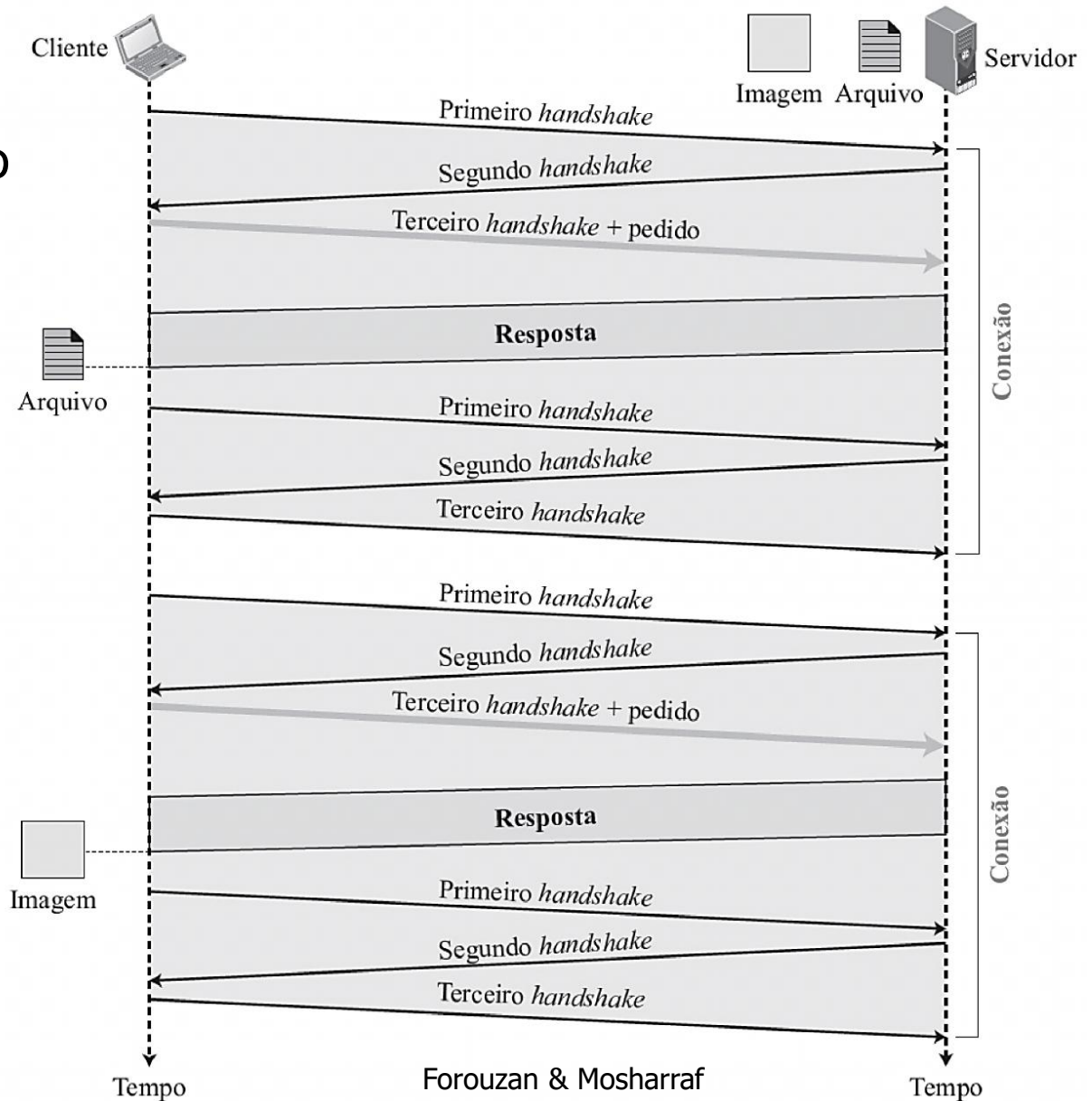
Kurose & Ross





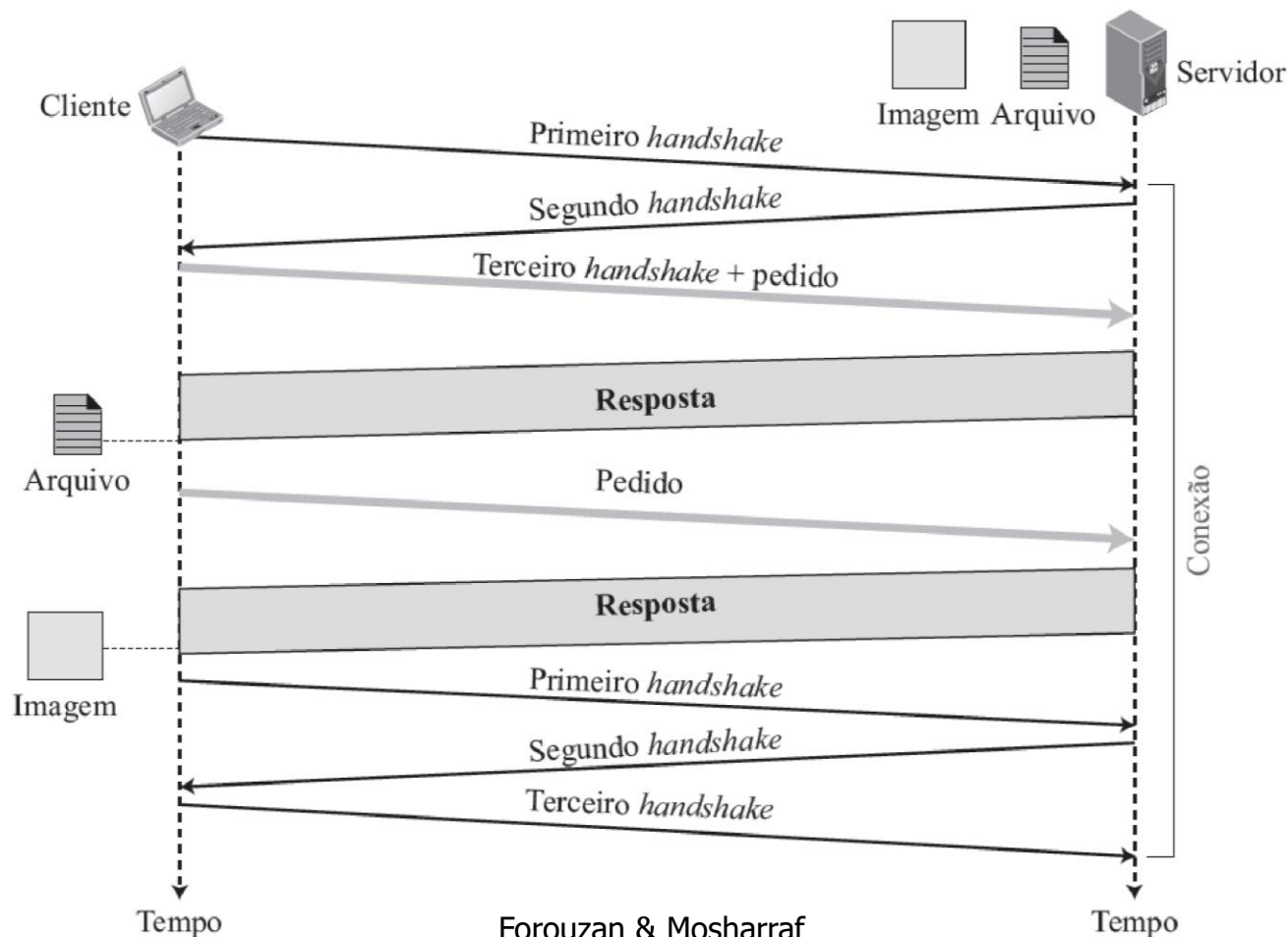
### Requisição/Resposta HTTP – conexões não persistentes:

- Cada par de requisição/resposta deve ser enviado por uma conexão TCP distinta
- No máximo um objeto é enviado em cada conexão TCP.
- HTTP/1.0



### Requisição/Resposta HTTP - Conexões persistentes:

- Todas as requisições e suas respostas são enviadas por uma mesma conexão TCP.
- Múltiplos objetos podem ser enviados em uma única conexão.
- HTTP/1.1 e HTTP/2





### Exemplos de mensagens HTTP:

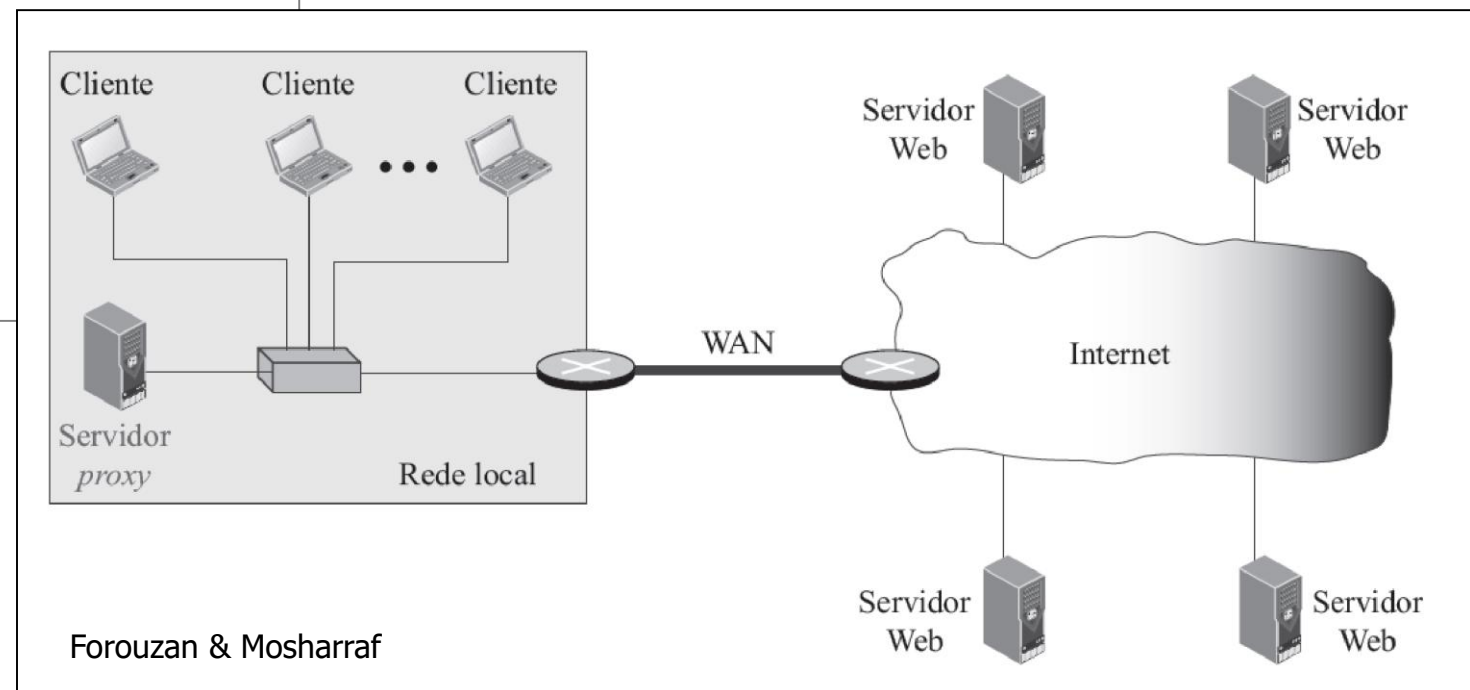
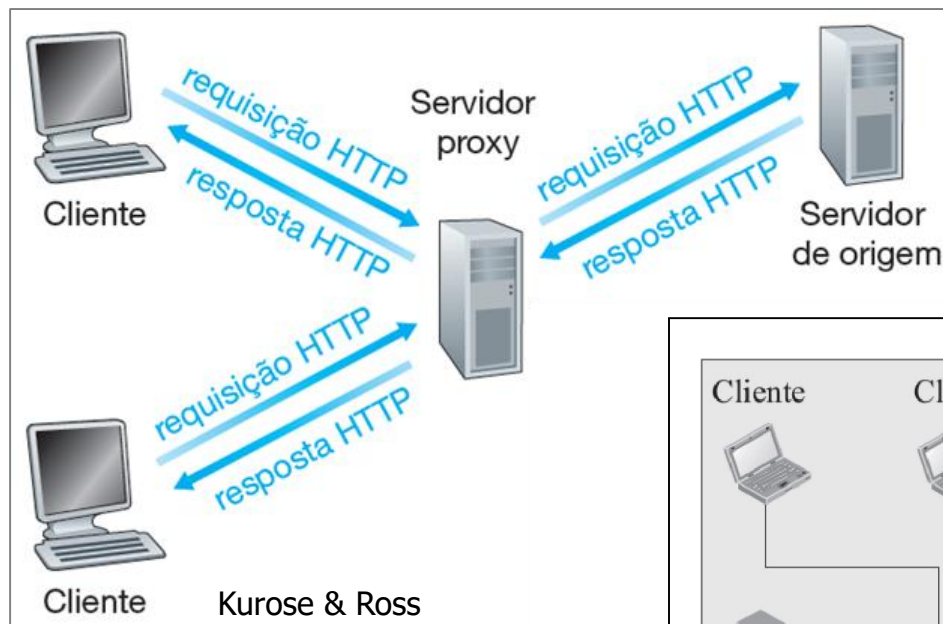
#### ✓ Mensagens de requisição:

- GET – cliente solicita um documento.
- HEAD – cliente solicita informações sobre um documento.
- POST – cliente envia alguma informação ao servidor.
- PUT – cliente envia um documento ao servidor.
- DELETE – remove uma página Web.
- Etc.

#### ✓ Mensagens de resposta:

- 200 OK – solicitação realizada.
- 301 Moved Permanently – os dados solicitados foram removidos.
- 400 Bad Request – solicitação mal formulada ou impossível de validar.
- 404 Not Found – servidor não encontrou nada no endereço indicado.
- 501 Not Implemented – servidor não suporta o serviço solicitado.
- 505 HTTP Version Not Supported – versão do protocolo HTTP requisitada não é suportada pelo servidor.

- ✓ Um **cache Web** (servidor proxy) é uma entidade da rede que atende requisições HTTP em nome de um servidor Web de origem, mantendo cópias de objetos recentemente requisitados.
  - Mensagem “conditional GET” – mecanismo que permite que um *cache* verifique se seus objetos estão atualizados.



Exemplo: Supondo que o navegador requisita o objeto <http://www.someschool.edu/campus.gif> ao cache Web.

1. O navegador estabelece uma conexão TCP com o cache Web e envia a ele uma requisição HTTP para o objeto.
2. O cache Web verifica se tem uma cópia do objeto armazenada localmente.
  - a) **Se tiver o objeto:** envia o objeto ao navegador do cliente, dentro de uma mensagem de resposta HTTP.
  - b) **Se não tiver o objeto:** o cache Web abre uma conexão TCP com o servidor de origem [www.someschool.edu](http://www.someschool.edu); então, envia uma requisição HTTP do objeto para a conexão TCP. O servidor de origem então envia o objeto solicitado ao cache Web, dentro de uma resposta HTTP. Recebendo o objeto, o cache Web guarda uma cópia em seu armazenamento local e envia outra cópia, dentro de uma mensagem de resposta HTTP, ao navegador do cliente, pela conexão TCP existente entre o navegador do cliente e o cache Web.

- ✓ Embora o HTTP seja flexível, não é um protocolo seguro.
  - As mensagens requisitadas enviam informações ao servidor em modo texto simples.
  - As respostas do servidor, normalmente páginas HTML, também não são criptografadas.
- ✓ O HTTPS possibilita conversações seguras na Internet, fazendo uso de autenticação e criptografia para proteger os dados trocados entre o cliente e o servidor.
  - O HTTPS usa o mesmo processo de requisição do cliente e resposta do servidor do HTTP, porém o fluxo de dados é criptografado antes de ser transportado através da rede.
  - TLS (Transport Layer Security), sucessor do SSL (Secure Sockets Layers), atua junto ao TCP garantindo a integridade dos dados através da criptografia e da autenticação.
- ✓ Faz uso da porta 443 do TCP.

- ✓ Atividade 02 – Protocolos de Aplicação: DNS e HTTP