

# Sistemas para Internet I 2014

Prof. Karina S. Machado

Email: [karinaecomp@gmail.com](mailto:karinaecomp@gmail.com)

Prédio do Centro de Ciências Computacionais – C3

2º. Andar – Laboratório LAMSA

# Alunos novos?

- Acessar o Moodle
- Estudar as aulas passadas
- Entrar em contato comigou ou com o Monitor Gabriel para dúvidas sobre o conteúdo já apresentado em aula

# Alunos novos?

- Para recuperação das aulas:
- De forma a recuperar as aulas perdidas pelos alunos novos (ingresso a partir de 14 de abril) por terem ocorrido antes de suas matrículas, esses alunos devem entregar a lista de exercícios 01 do 1º.bimestre toda resolvida A MÃO. A lista então deve ser digitalizada e enviada no ambiente Moodle até dia 19-05 as 23 horas.
- A não entrega dessa atividade implica em aplicação de faltas para os alunos novos nas datas de aula anteriores a sua matrícula.

# Data da prova

- **26-05-14**
- **Na sala de aula**
- **Individual e sem consulta**
- **70% da nota do bimestre  
(trabalho 30%)**

# Para alunos novos e que não entregaram trabalho 01

- Para os alunos que não entregaram o trabalho 01 do Vídeo.
- ATENÇÃO: SOMENTE PARA OS ALUNOS QUE NÃO ENTREGARAM O VÍDEO.

PARA OS ALUNOS MATRICULADOS TARDIAMENTE, O TRABALHO NÃO TERÁ DESCONTO NA NOTA.

PARA OS DEMAIS, HAVERÁ DESCONTO POR ENTREGA EM ATRASO (NÃO ENTREGOU O VÍDEO)

# Para alunos novos e que não entregaram trabalho 01

- Fazer um relatório de no mínimo 10 páginas sobre a historia da internet e suas principais características. Deverá conter:
  - Capa
  - Indice
  - Introdução
  - Historico
  - Caracteristicas
  - Conclusão
  - Referencias
- Entregar até domingo 25/05 via moodle em versão SOMENTE PDF.

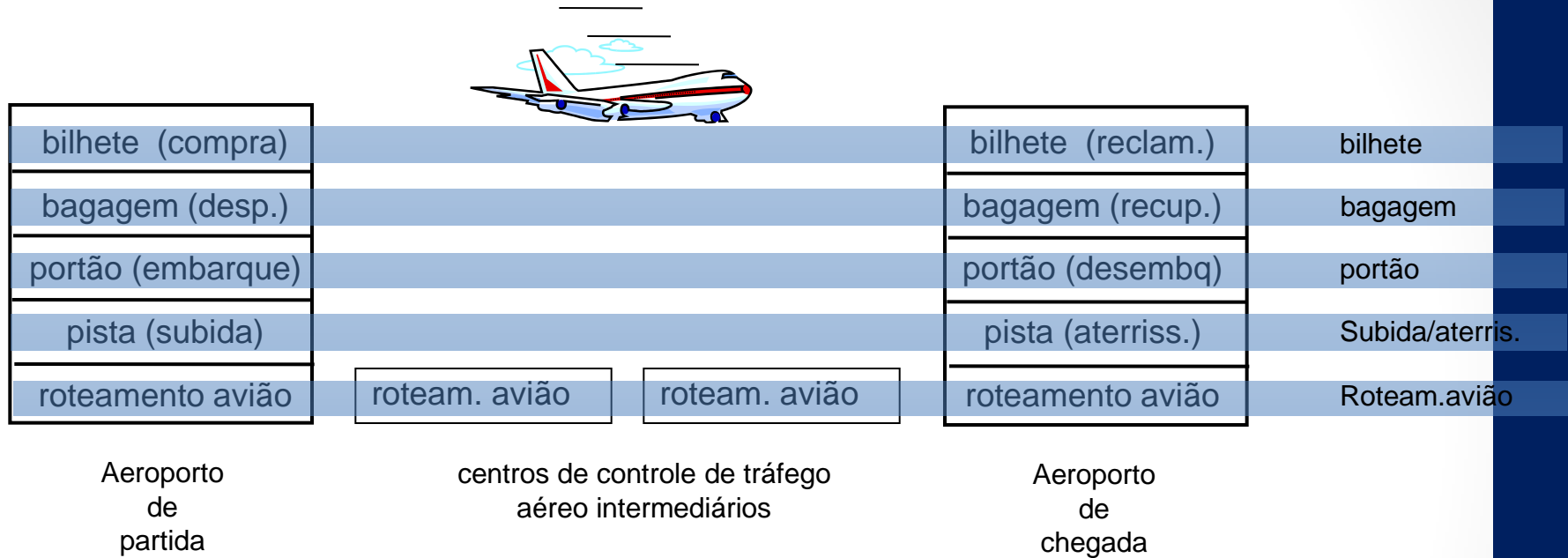
# Aula hoje:

- Revisão aulas anteriores
- Camada de Aplicação

Aula Passada...



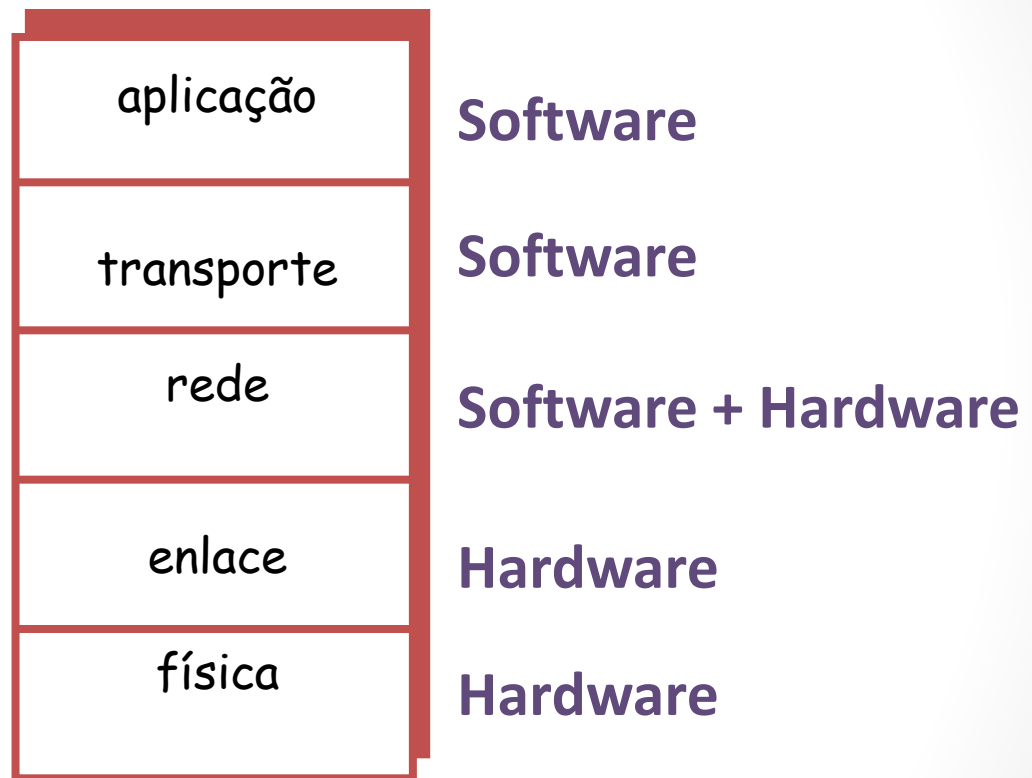
# Funcionalidade de uma empresa aérea em camadas



## Camadas:

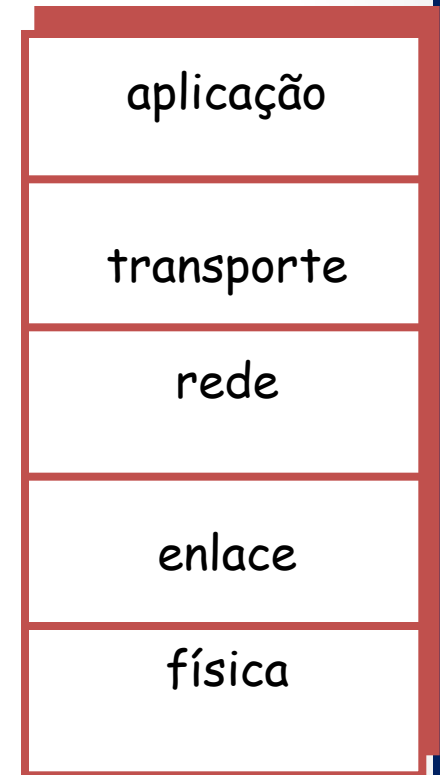
- cada camada implementa um serviço
- Utiliza serviços providos pela camada inferior

# Pilha de protocolos Internet

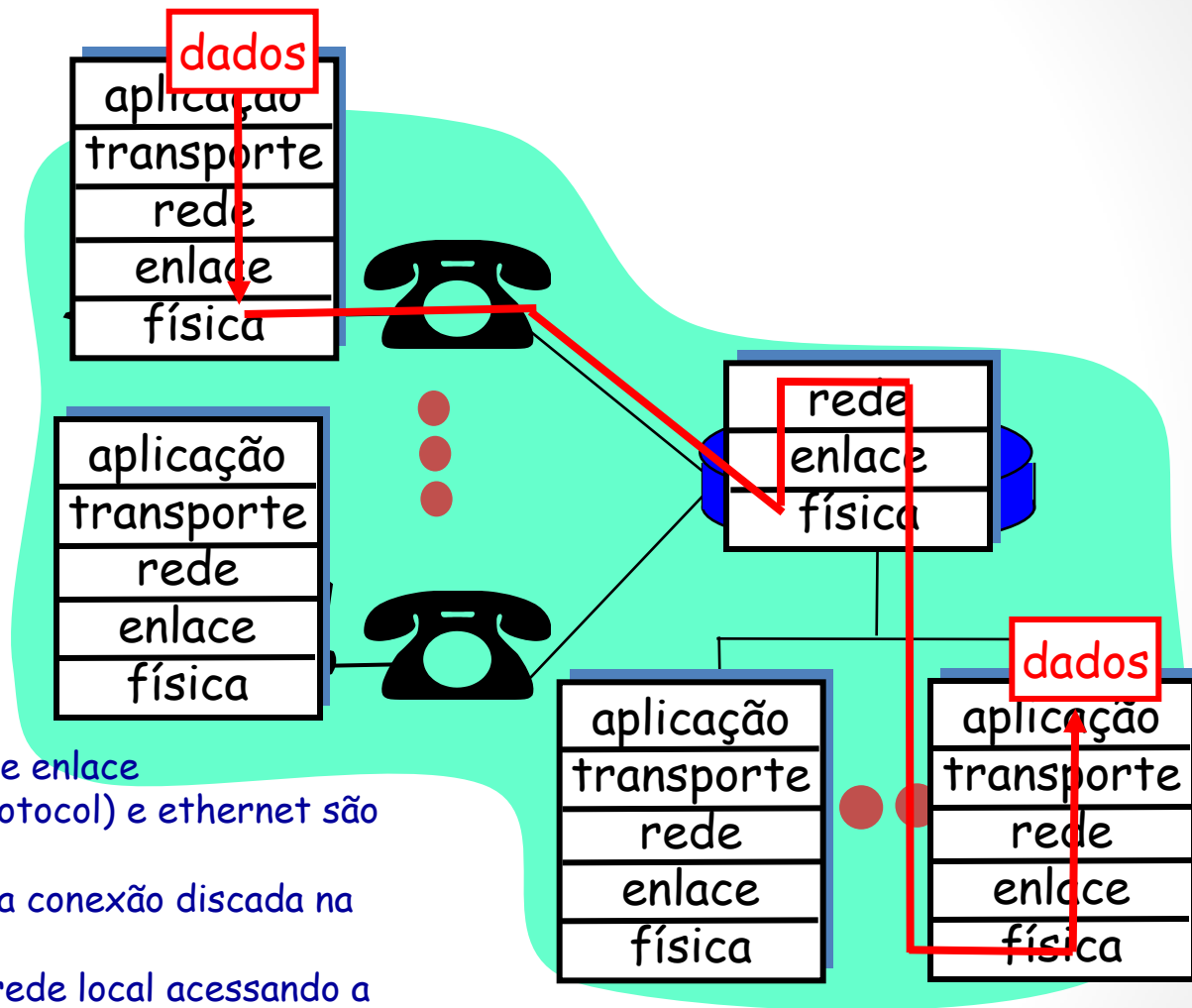


# RESUMO

- aplicação: dá suporte a aplicações de rede
  - FTP, SMTP, HTTP
  - Quase sempre software
- transporte: transferência de dados host-a-host
  - TCP, UDP
  - Quase sempre software
- rede: roteamento de datagramas da origem até o destino
  - IP, protocolos de roteamento
  - Misto de hardware e software
- enlace: transferência de dados entre elementos de rede vizinhos
  - PPP, Ethernet
  - Placa de interface de rede
- física: bits “no fio”

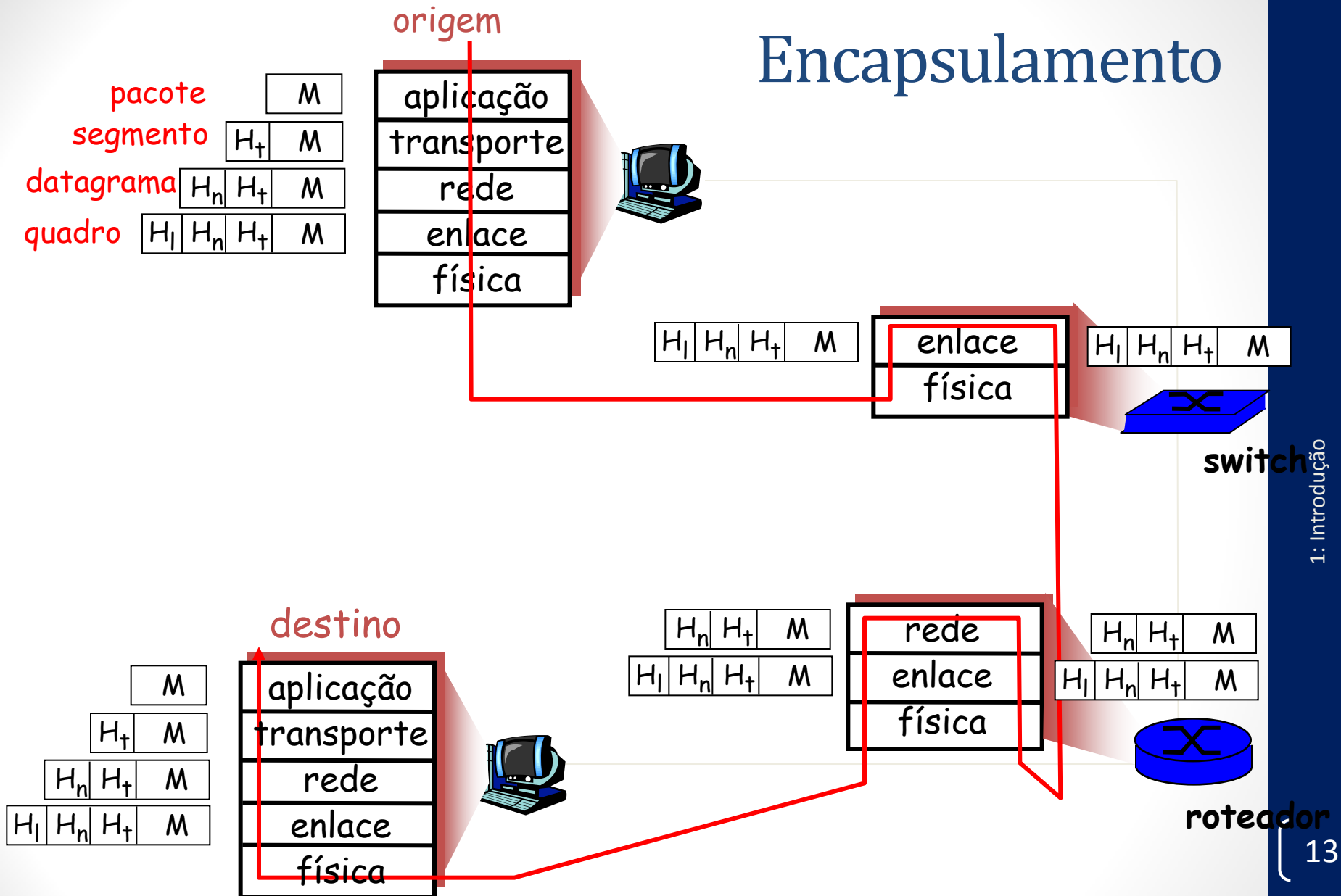


# Camadas: comunicação física



- existem vários protocolos de enlace
  - PPP (point-to-point protocol) e ethernet são os principais
    - PPP é usado numa conexão discada na internet
    - Ethernet numa rede local acessando a internet
- Físico
  - par trançado, fibra, coax, etc .

# Encapsulamento



Switch=comutadores=não reconhecem endereço IP, apenas endereço MAC da camada de enlace.

# Camada de Aplicação

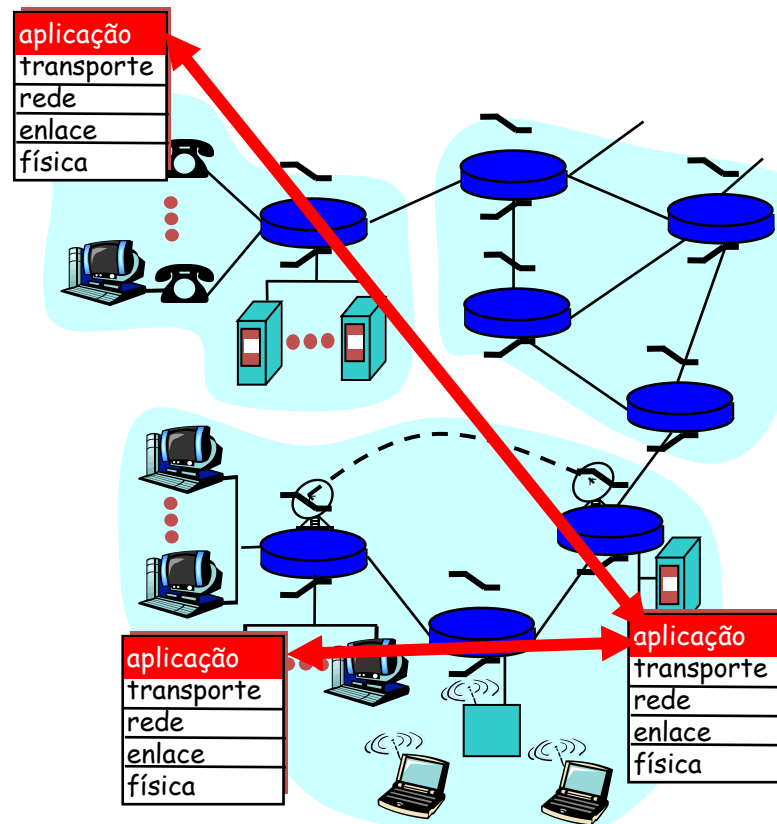
- Redes de computadores e a internet: uma abordagem top-down / James F. Kurose ; Keith W. Ross
  - Capítulo 02 – (TÓPICOS 2.1 E 2.2) Só os itens vistos em aula
- OU
- Redes de computadores / Andrew S. Tanenbaum
  - Capítulo 07 – (TÓPICOS 7.3) Só os itens vistos em aula

# CAMADA DE APLICAÇÃO

- E-mail
- Web
- Instant messaging
- Login remoto
- Compartilhamento de arquivos P2P
- Jogos de rede multi-usuários
- Vídeoclipes armazenados
- Voz sobre IP
- Vídeo conferência em tempo real
- Computação paralela em larga escala

# CAMADA DE APLICAÇÃO

- Arquiteturas de aplicações de rede

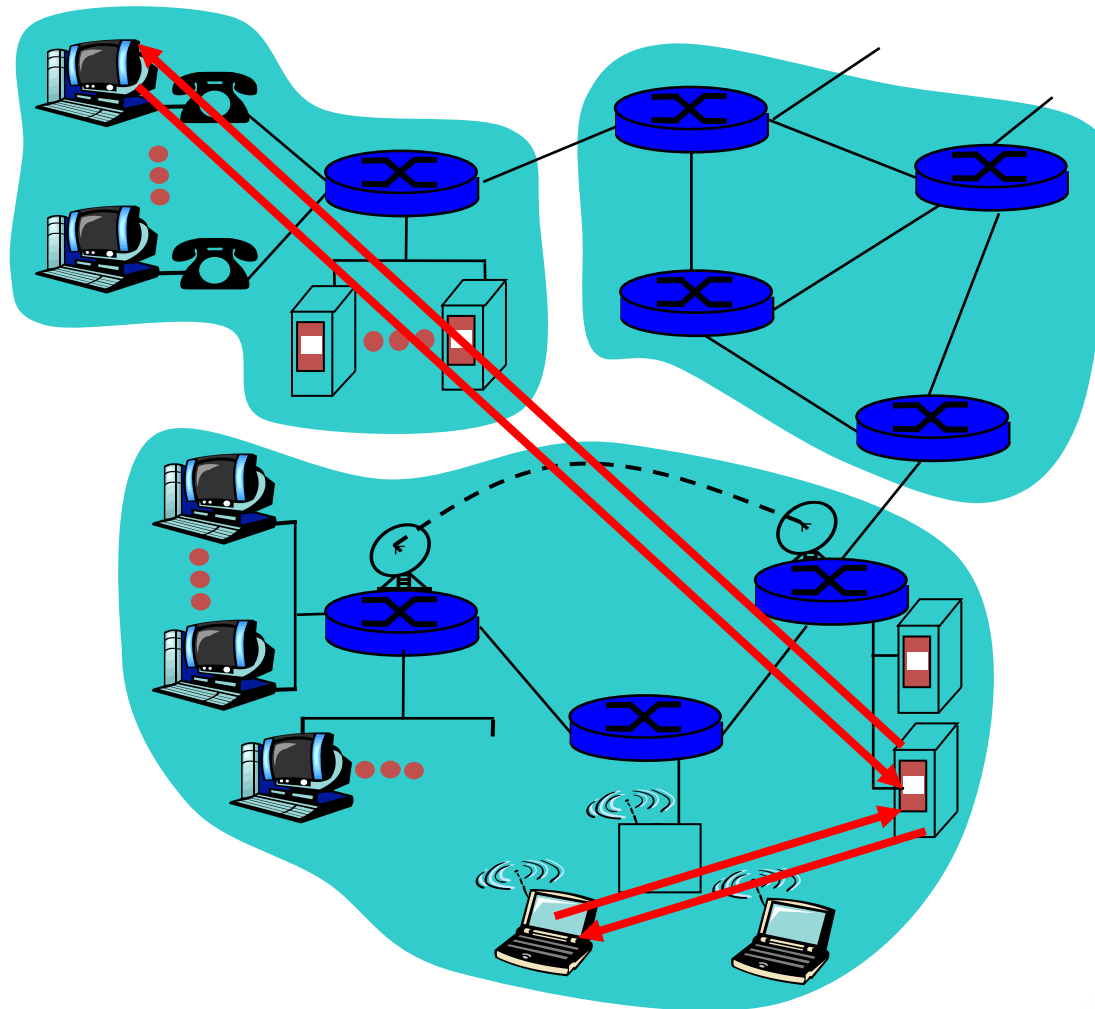




# Arquiteturas das aplicações

- Cliente-servidor
- Peer-to-peer (P2P)
- Híbrido de cliente-servidor e P2P

# Arquitetura cliente-servidor



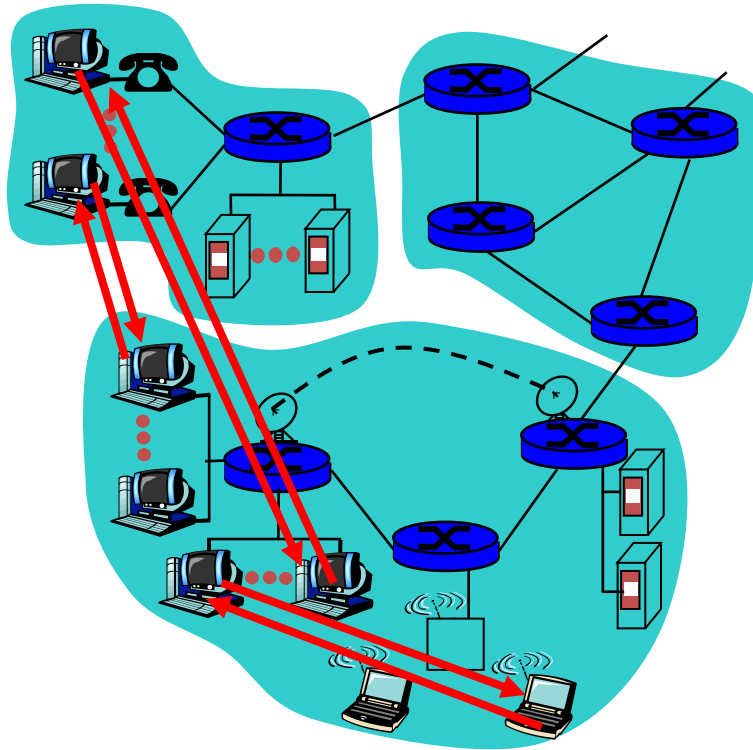
# Cliente-servidor

- Papel do Servidor
  - Aguardar conexões
  - Tem um IP e portas conhecidos pelos clientes
  - Fornece um serviço a vários clientes

Ex: Apache (Servidor WEB), Postfix (Servidor de Email)
- Papel do Cliente
  - Conecta-se ao servidor para requisitar o serviço
  - Não se comunicam diretamente uns com os outros

Ex.: Firefox, Internet Explorer, Outlook

# Arquitetura P2P pura



Alta escalabilidade  
Porém, difícil de gerenciar

# Modelo P2P

- “São sistemas distribuídos compostos de nós interconectados, aptos a se auto-organizar em topologias de rede, com o intuito de compartilhar recursos, como conteúdo, ciclos de CPU, largura de banda e armazenamento, com a capacidade de adaptação a faltas e acomodação a um número variável de nós, ao mesmo tempo que mantém a conectividade e o desempenho em níveis aceitáveis, sem a necessidade de suporte ou intermediação de um servidor centralizado.”
- (Adroutsellis-Theotokis & Spinellis, 2004)

# P2P

- Ponto a Ponto
- Não há um servidor central
- Os sistemas finais se comunicam diretamente
- Alta escalabilidade, difícil de gerenciar

# P2P Características

- Auto-organização: não há um coordenador do organização: não há um coordenador do grupo; toda a coordenação é distribuída
- Adaptabilidade: rede se ajusta ao ambiente, mesmo que ocorram falhas
- Escalabilidade: rede cresce em escala facilmente; não há ponto de estrangulamento
- Comunicação direta entre os pares: se opõe ao tradicional modelo cliente ao tradicional modelo cliente--servidor, já que servidor, já que cada nó pode fornecer ou obter recursos

# Modelo Híbrido

- BitTorrent
- Transferência de arquivo P2P
- Busca centralizada de arquivos:
  - Conteúdo de registro dos pares no servidor central
  - Consulta de pares no mesmo servidor central para localizar o conteúdo
- MSN
- Bate-papo entre dois usuários é por um servidor
  - Usuário registra seu IP com o servidor central quando fica online
  - Usuário contata o servidor central para encontrar endereços IP dos amigos
- Transferência de arquivos é P2P



# Requisitos do serviço de transporte de aplicações comuns: tabela

## **Perdas**

- sem perdas
- Tolerante

## **Banda**

- Elástica
- áudio: 5Kb-1Mb
- vídeo:10Kb-5Mb
- > alguns Kbps

## **Sensibilidade temporal**

- Não
- sim, 100's mseg
- sim e não

---

A Internet de hoje ainda não provê garantia de Banda e Sensibilidade Temporal

## Preencha a tabela:

<b>Aplicação</b>	<b>Perdas</b>	<b>Banda</b>	<b>Sensibilidade temporal</b>
transferência de arqs			
correio			
documentos WWW			
áudio/vídeo de tempo real			
videoconferência			
áudio/vídeo gravado			
jogos interativos			
Mensagem instantânea			

# Requisitos do serviço de transporte de aplicações comuns

<b>Aplicação</b>	<b>Perdas</b>	<b>Banda</b>	<b>Sensibilidade temporal</b>
transferência de arqs	sem perdas	elástica	não
correio	sem perdas	elástica	não
documentos WWW	sem perdas	elástica	não
áudio/vídeo de tempo real videoconferência	tolerante	áudio: 5Kb-1Mb vídeo:10Kb-5Mb	sim, 100's mseg
áudio/vídeo gravado	tolerante	como anterior	sim, alguns segs
jogos interativos	tolerante	> alguns Kbps	sim, 100's mseg
Mensagem instantânea	sem perdas	elástica	sim e não

A Internet de hoje ainda não provê garantia de Banda e Sensibilidade Temporal

# Aplicações Internet: seus protocolos e seus protocolos de transporte

<b>Aplicação</b>	<b>Protocolo da camada de apl</b>	<b>Protocolo de transporte usado</b>
correio eletrônico	SMTP [RFC 2821]	TCP
acesso terminal remoto	telnet [RFC 854]	TCP
Web	HTTP [RFC 2616]	TCP
transferência de arquivos	FTP [RFC 959]	TCP
streaming multimídia	HTTP(ex. YouTube), RTP	TCP ou UDP
telefonia Internet	SIP, RTP, ou Proprietário (Skype)	tipicamente UDP

# HTTP

- **página Web** consiste em **objetos**
- objeto pode ser arquivo HTML, imagem JPEG, applet Java, arquivo de áudio,...
- página Web consiste em **arquivo HTML básico** que inclui vários objetos referenciados
- cada objeto é endereçável por um **URL**
- **exemplo de URL:**

`www.someschool.edu/someDept/pic.gif`

nome do hospedeiro

nome do caminho

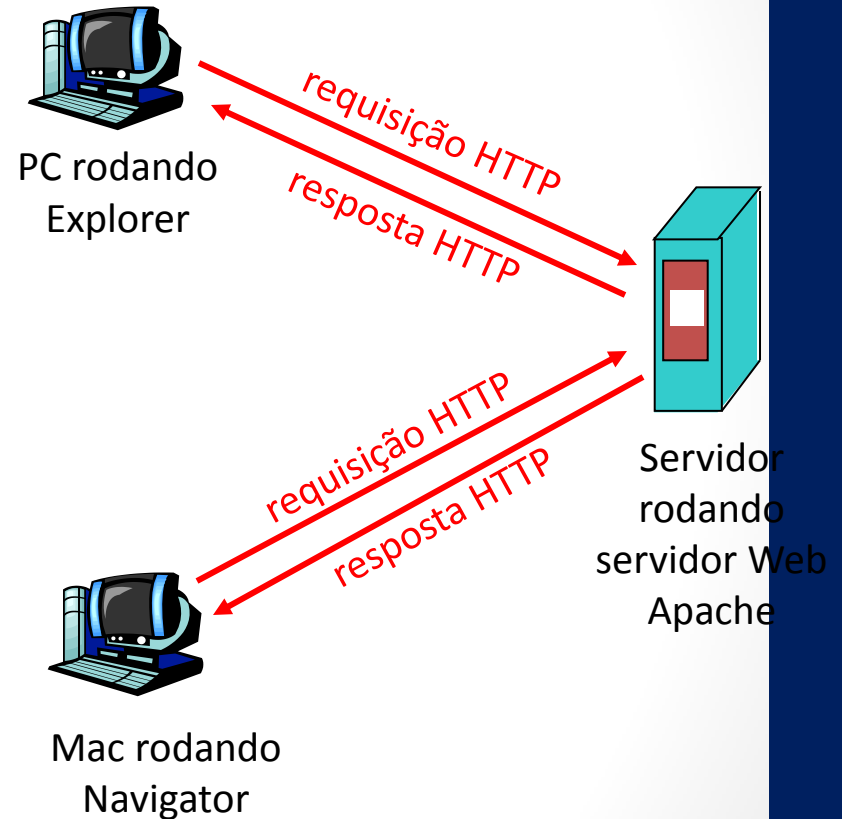
# Visão geral do HTTP

## HTTP: HyperText Transfer Protocol

- protocolo da camada de aplicação da Web
- modelo cliente/servidor
  - *cliente*: navegador que requisita, recebe, “exibe” objetos Web
  - *servidor*: servidor Web envia objetos em resposta a requisições

## usa TCP:

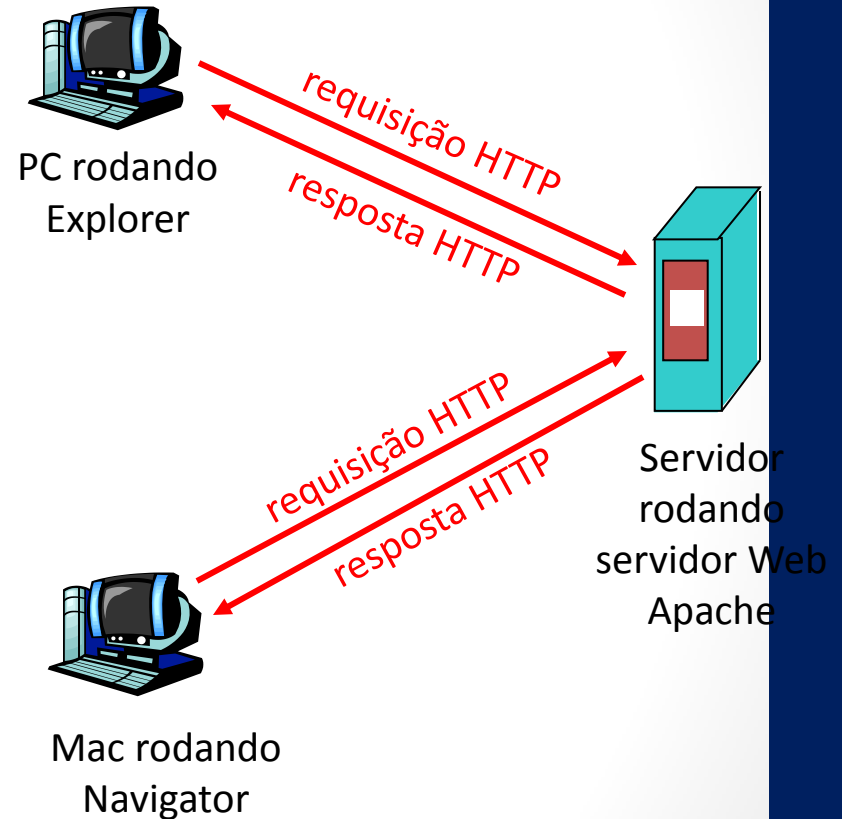
- cliente inicia conexão TCP (cria socket) com servidor, porta 80



# Visão geral do HTTP

## PASSOS

1. O usuário requisita página WEB
2. O browser envia ao servidor mensagens HTTP de requisição para os objetos da página
3. O servidor recebe as requisições e responde com mensagens HTTP que contém os objetos



# Conexões HTTP

## HTTP não persistente

- no máximo um objeto é enviado por uma conexão TCP.

## HTTP persistente

- múltiplos objetos podem ser enviados por uma única conexão TCP entre cliente e servidor.



## 2.2.2 Exemplo de HTTP não persistente

Supomos que usuário digita a URL  
www.algumaUniv.br/algumDepartamento/index.html

(contém texto,  
referências a 10  
imagens jpeg)

1a. Cliente http inicia conexão  
TCP a servidor http (processo)  
www.algumaUniv.br na Porta  
80 padrão para servidor http.

1b. servidor http no hospedeiro  
www.algumaUniv.br espera por  
conexão TCP na porta 80.  
"aceita" conexão, avisando ao  
cliente

2. cliente http envia *mensagem de  
pedido* de http (contendo URL  
incluindo /algumDepartamento  
/index.html) através do socket  
da conexão TCP

3. servidor http recebe mensagem  
de pedido, formula *mensagem  
de resposta* contendo objeto  
solicitado (algumDepartamento  
/index.html), envia mensagem via  
socket

tempo  
↓

# Exemplo de HTTP não persistente (cont.)

4. servidor http encerra conexão TCP .



5. cliente http recebe mensagem de resposta contendo arquivo html, mostra html. Analisando arquivo html, encontra 10 objetos jpeg referenciados

6. Passos 1 a 5 repetidos para cada um dos 10 objetos jpeg

tempo



# HTTP persistente

## problemas do HTTP não persistente:

- Demora na transmissão de cada página, principalmente se há muitos objetos
- navegadores geralmente abrem conexões TCP paralelas para buscar objetos referenciados

## HTTP persistente:

- servidor deixa a conexão aberta depois de enviar a resposta
- mensagens HTTP seguintes entre cliente/servidor enviadas pela conexão aberta
- cliente envia requisições assim que encontra um objeto referenciado

# Integração usuário-servidor

## Cookies

- Permitem que sites identifiquem e monitorem os seus usuários.
  - Restringir acessos
  - Apresentar conteúdo em função da identidade
- São textos que podem ser armazenados no disco rígido com dados do usuário.

# *Cookies*: manutenção do “estado” da conexão

## Tem 4 componentes:

1. linha de cabeçalho do *cookie* na mensagem de resposta HTTP
2. linha de cabeçalho do *cookie* na mensagem de pedido HTTP
3. Arquivo do *cookie* mantido no sistema final do usuário e gerenciado pelo browser do usuário
4. Banco de Dados (BD) de apoio no site Web

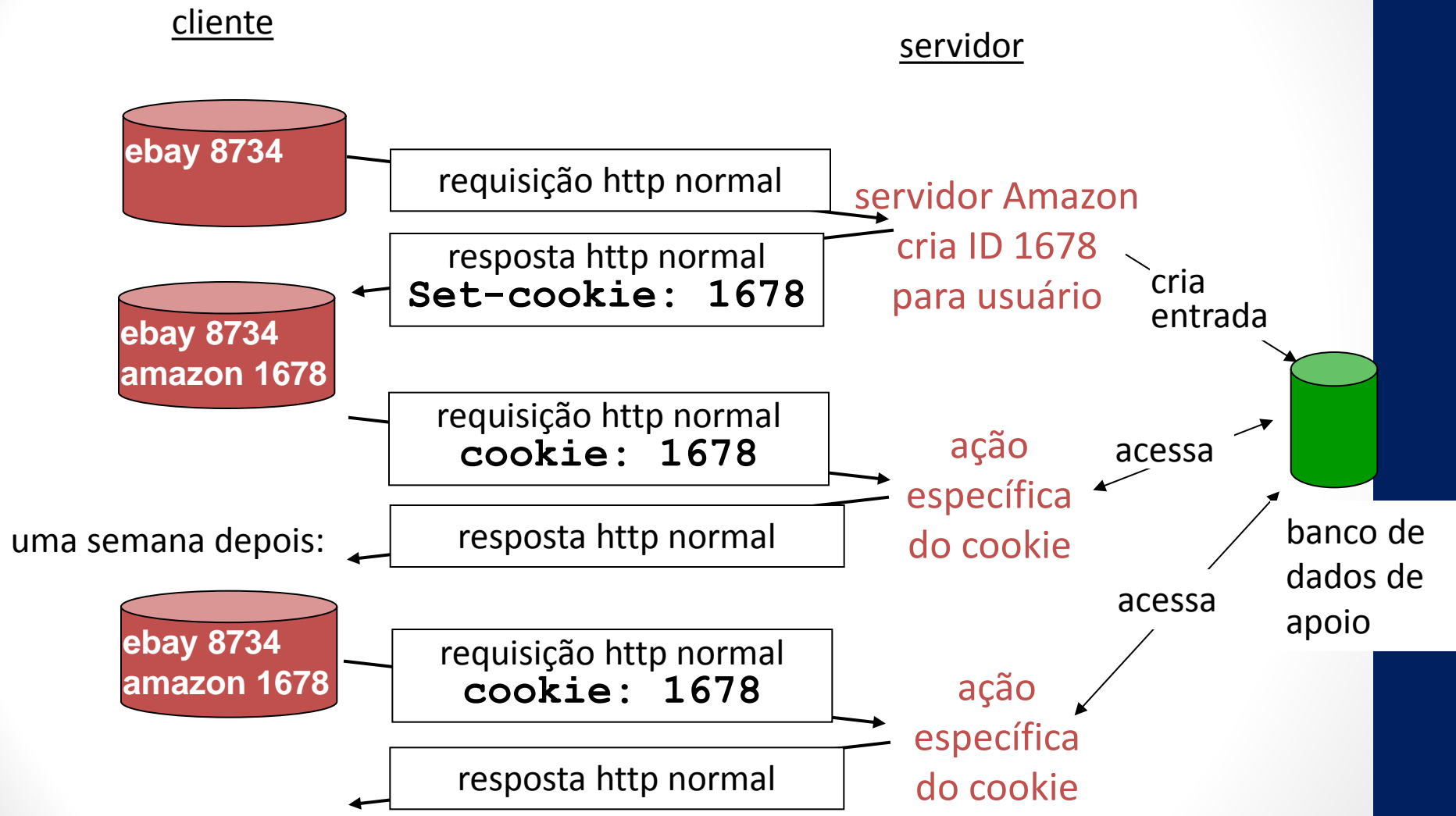
# *Cookies: manutenção do “estado” da conexão*

## Tem 4 componentes:

### Exemplo:

1. linha de endereço
  2. linha de cookies
  3. Arquivo de cookies  
gerenciado pelo navegador
  4. Banco de dados de cookies
- Suzana acessa a Internet sempre do mesmo PC
  - Ela visita um site específico de comércio eletrônico pela primeira vez
  - Quando os pedidos iniciais HTTP chegam no site, o site cria uma ID (ex. 1678) única e cria uma entrada para a ID no Banco de Dados de apoio

# Cookies: manutenção do “estado” (cont.)



# Cookies (continuação)

## O que os cookies podem fazer:

- **Autorização** após armazenamento do registro da pessoa
- **Registro** da lista de compras no Ecommerce
- **Sugestões** -recomendar produtos
- estado da sessão do usuário (*Web email*) – identificação do usuário
- Eles armazenam coisas que você acessou, sites que você viu -> problemas em relação a privacidade

## Cookies e privacidade:

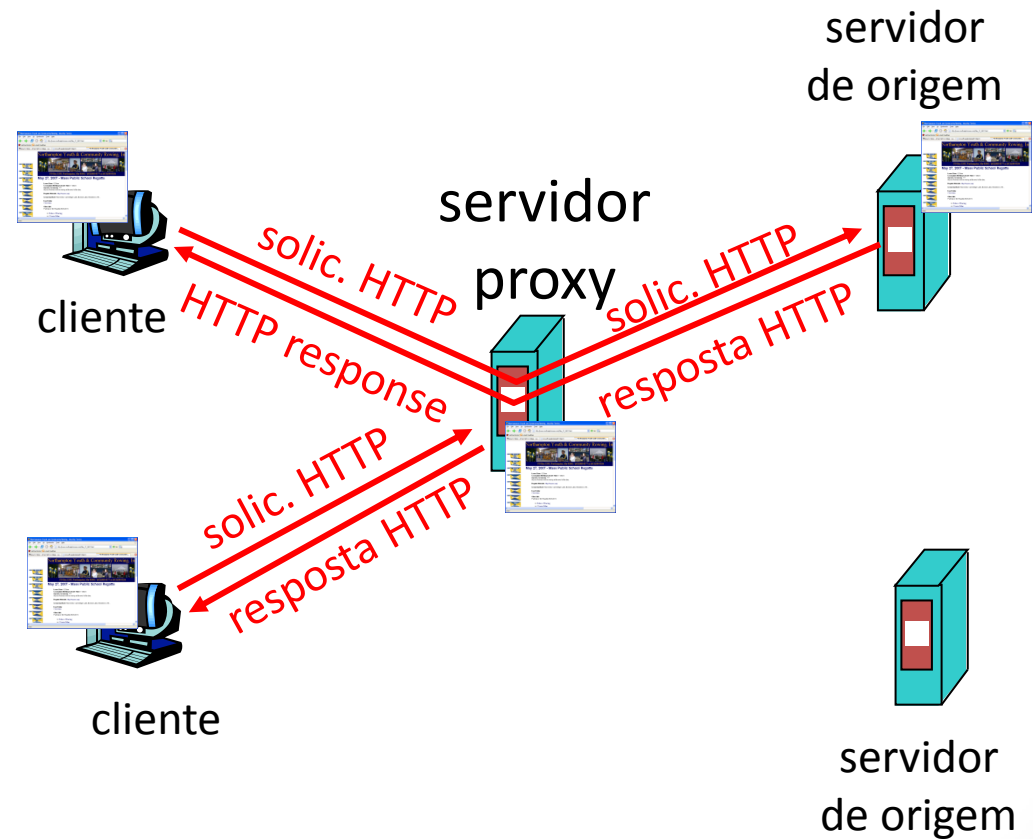
- ☐ cookies permitem que os sites aprendam muito sobre você
- ☐ você pode fornecer nome e e-mail para os sites
- ☐ mecanismos de busca usam redirecionamento e *cookies* para aprender ainda mais sobre você
- ☐ agências de propaganda obtêm perfil a partir dos sites visitados e oferecem produtos perturbando os usuários (ex. DoubleClick)



# Cache Web (servidor proxy)

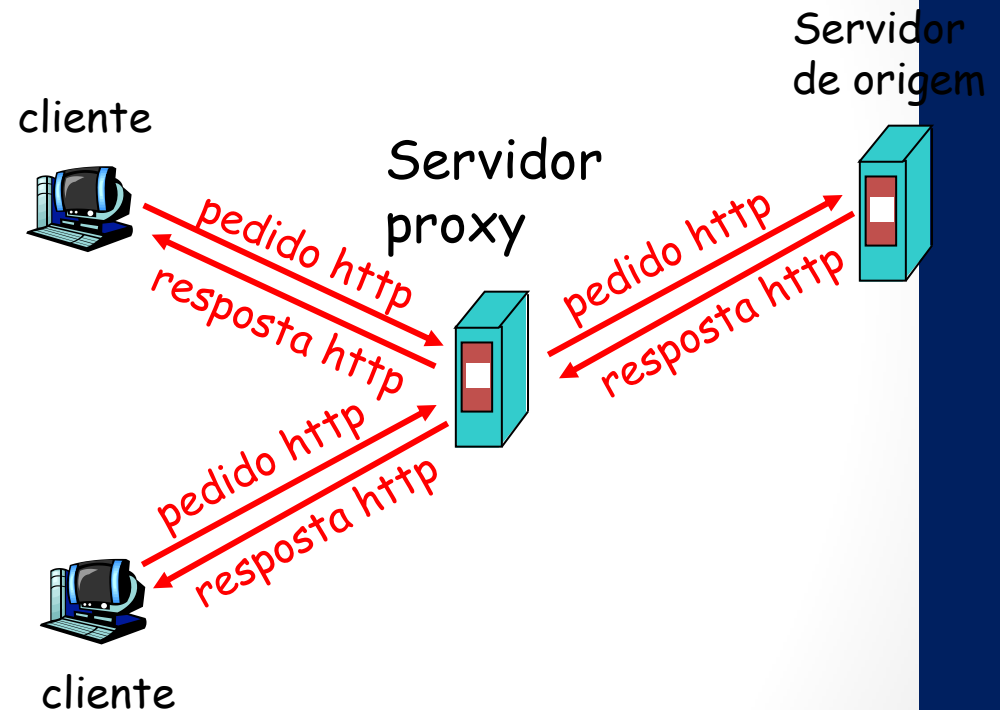
- É uma entidade da rede que atende requisições HTTP em nome de um servidor web de origem.
- Mantém em um disco próprio de armazenamento cópias dos objetos utilizados recentemente.
- Pode ser tanto servidor (quando atende requisições diretamente) quanto cliente (quando precisa buscar algo no servidor de origem)

## 2.2.5 Cache Web (servidor proxy)



## 2.2.5 Cache Web (servidor proxy)

1. O browser estabelece uma conexão TCP com a cache e envia um pedido HTTP
2. O proxy verifica se objeto está no cache ,
  1. Se o objeto está na cache, este o devolve imediatamente na resposta HTTP
  2. senão, solicita objeto do servidor de origem, depois devolve resposta HTTP ao cliente
  3. Quando recebe o objeto, o proxy guarda uma cópia em seu armazenamento local



# Mais sobre Caches Web

- Tipicamente o cache é instalado por um ISP (universidade, empresa, ISP residencial)

## Para que fazer cache Web?

- Redução do tempo de resposta para os pedidos do cliente (principalmente se houver uma conexão de alta velocidade entre o host e o proxy)
- Redução do tráfego no canal de acesso de uma instituição