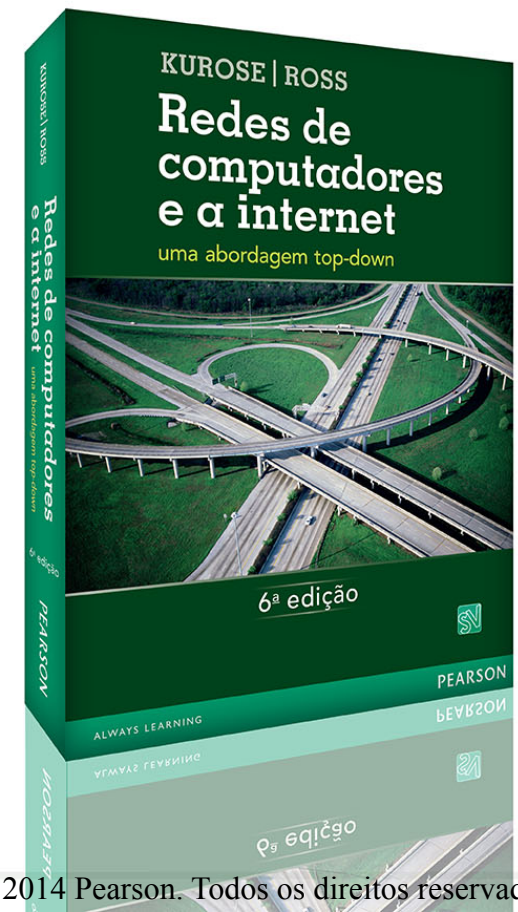


A camada de aplicação



Princípios de aplicações de rede

KUROSE | ROSS

Redes de computadores e a internet

uma abordagem top-down

6ª edição

- O núcleo do desenvolvimento de aplicação de rede é escrever programas que rodem em sistemas finais diferentes e se comuniquem entre si.
- Ao desenvolver sua nova aplicação, você precisará escrever um software que rode em vários sistemas finais.
- Você não precisará escrever programas que executem nos elementos do núcleo de rede, como roteadores e comutadores.

Arquiteturas de aplicação de rede

KUROSE | ROSS

Redes de computadores e a internet

uma abordagem top-down

6ª edição

- A **arquitetura de rede** é fixa e provê um conjunto específico de serviços.
- A **arquitetura da aplicação** é projetada pelo programador e determina como a aplicação é organizada nos vários sistemas finais.

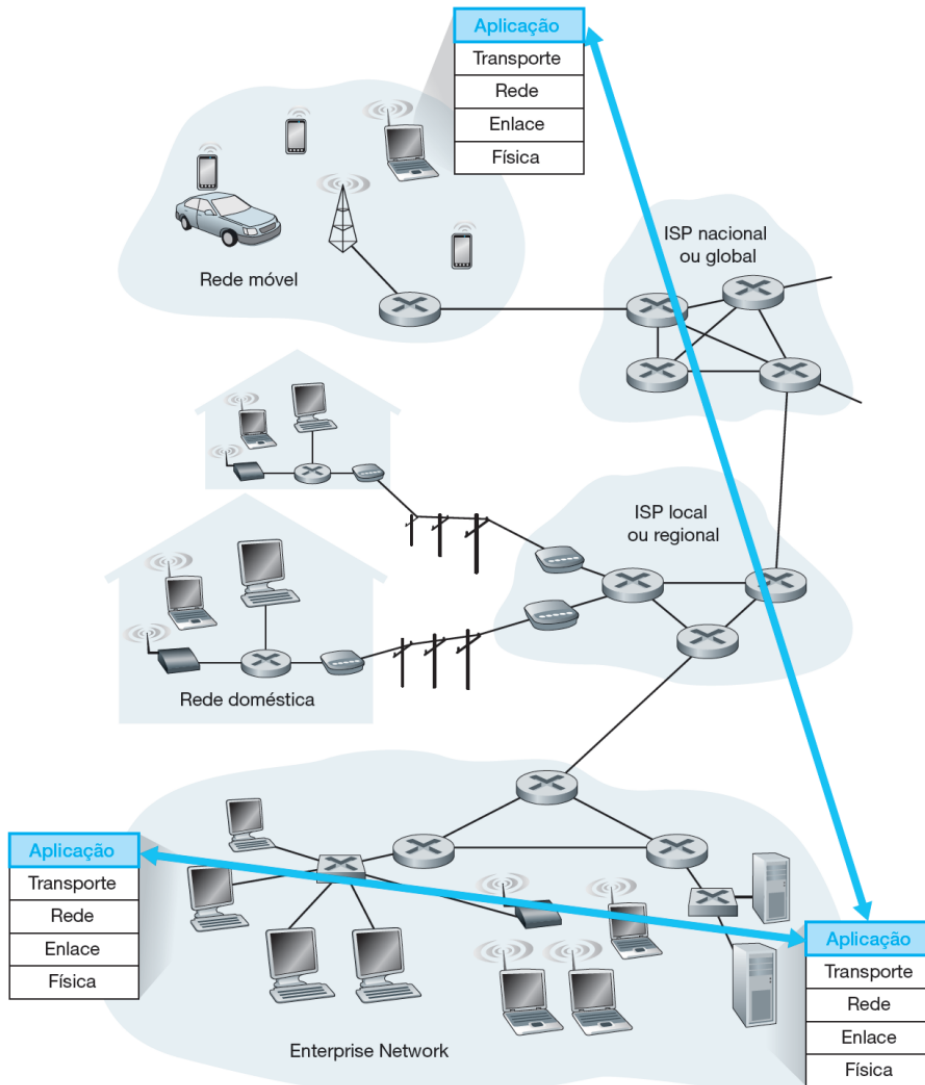
Arquiteturas de aplicação de rede

KUROSE | ROSS

Redes de computadores e a internet

uma abordagem top-down

6ª edição



- A comunicação de uma aplicação de rede ocorre entre sistemas finais na camada de aplicação.

Arquiteturas de aplicação de rede

KUROSE | ROSS

Redes de computadores e a internet

uma abordagem top-down

6ª edição

- Na **arquitetura cliente-servidor** há um hospedeiro sempre em funcionamento, denominado *servidor*, que atende a requisições de muitos outros hospedeiros, denominados *clientes*.
- *Clientes* podem estar em funcionamento às vezes ou sempre.
- *Clientes* não se comunicam diretamente uns com os outros.
- *Servidor* tem um endereço fixo e bem conhecido.

Arquiteturas de aplicação de rede

KUROSE | ROSS

Redes de computadores e a internet

uma abordagem top-down

6ª edição

- Na **arquitetura P2P** não há um servidor sempre funcionando no centro da aplicação.
- Pares arbitrários de hospedeiros, denominados *peers*, comunicam-se diretamente entre si.
- Como os pares se comunicam sem passar por nenhum servidor especial, a arquitetura é denominada par-a-par (*peer-to-peer* – P2P).
- Nenhuma das máquinas participantes precisa estar sempre em funcionamento.
- Um hospedeiro participante pode mudar seu endereço IP cada vez que for ligado.

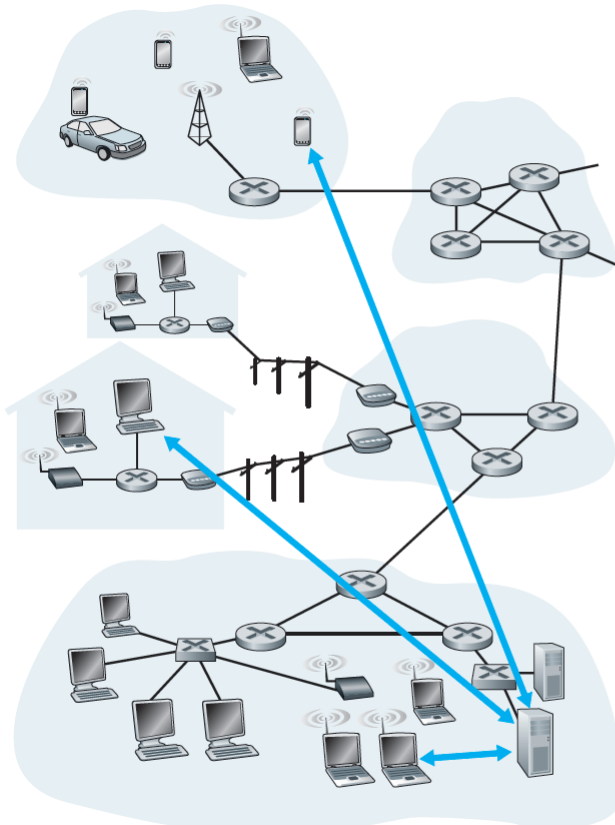
Arquiteturas de aplicação de rede

KUROSE | ROSS

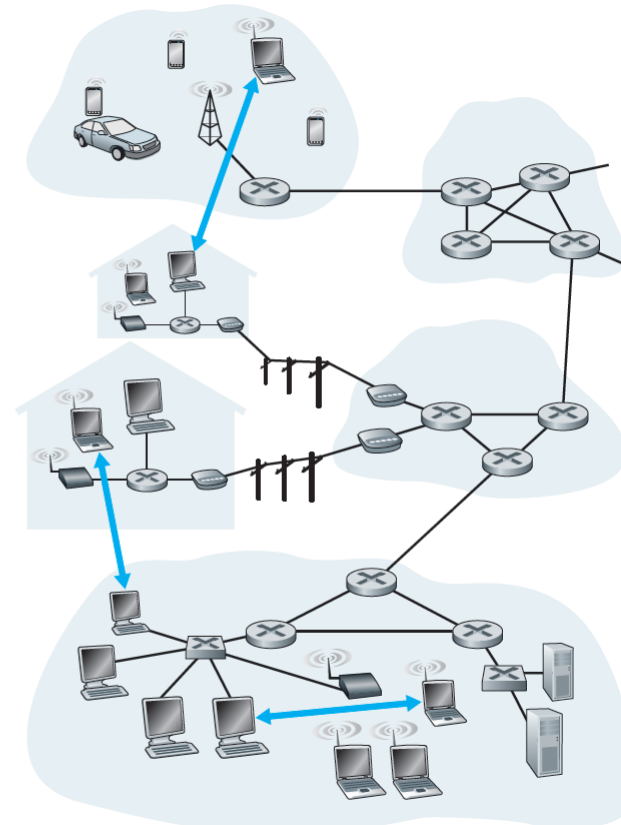
Redes de computadores e a internet

uma abordagem top-down

6ª edição



a. Arquitetura cliente-servidor



b. Arquitetura P2P

Comunicação entre processos

KUROSE | ROSS

Redes de computadores e a internet

uma abordagem top-down

6ª edição

- Uma aplicação de rede consiste em pares de processos que enviam mensagens uns para os outros por meio de uma rede.
- Um processo envia mensagens para a rede e recebe mensagens dela através de uma interface de software denominada *socket*.
- Para identificar o processo receptor, duas informações devem ser especificadas:
 1. Endereço IP do hospedeiro.
 2. Número de porta que especifica o processo receptor no hospedeiro de destino.

Comunicação entre processos

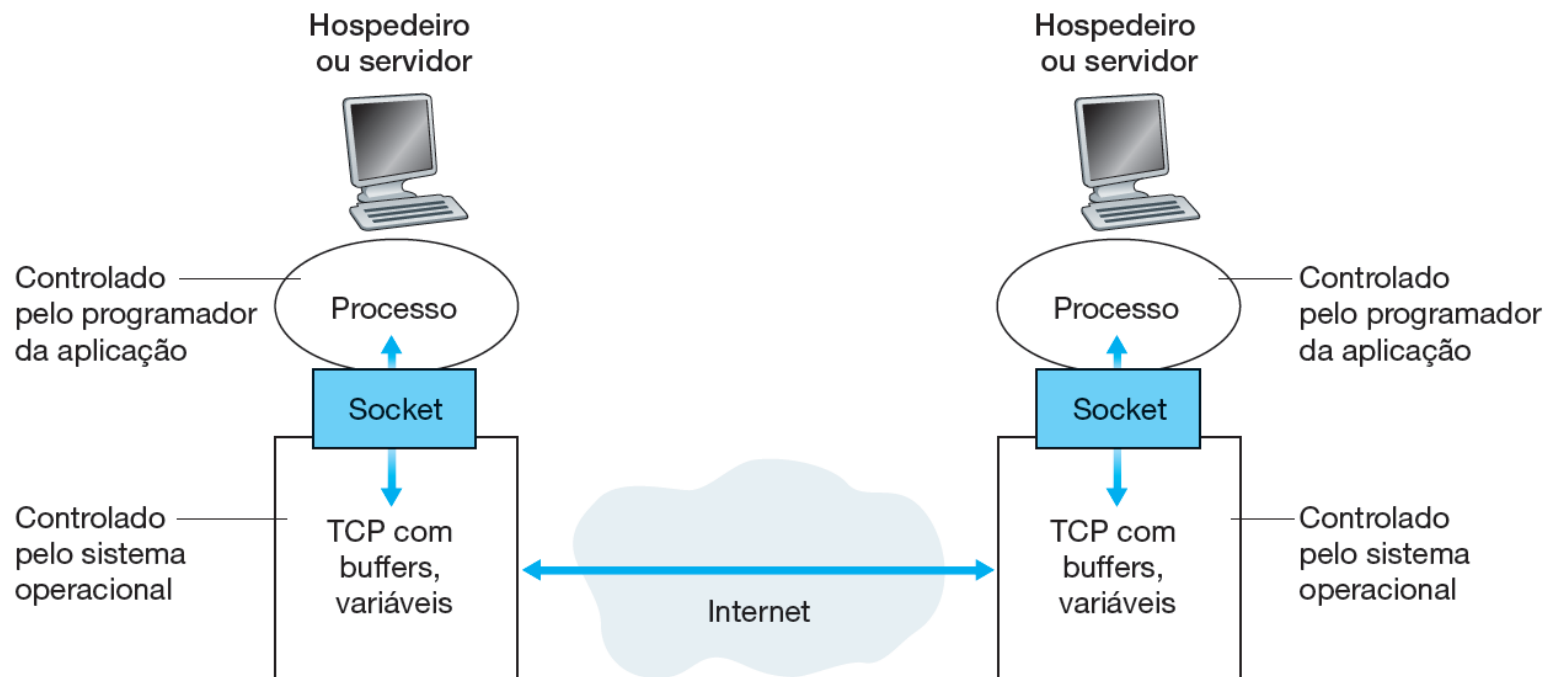
KUROSE | ROSS

Redes de computadores e a internet

uma abordagem top-down

6ª edição

- Processos de aplicação, *sockets* e protocolo de transporte subjacente.



Serviços de transporte disponíveis para aplicações

KUROSE | ROSS

Redes de computadores e a internet

uma abordagem top-down

6ª edição

- Uma aplicação deve escolher um dos protocolos de camada de transporte da Internet.
- A escolha depende dos serviços que a aplicação necessita.
- **Transferência confiável de dados**
 - Aplicação tolera perda de dados ou não?
- **Vazão**
 - Aplicação tem de transmitir dados a uma certa velocidade para ser efetiva?
- **Temporização**
 - Aplicação possui limitação estrita no atraso fim-a-fim para a entrega dos dados?

Serviços de transporte providos pela Internet

KUROSE | ROSS

Redes de computadores e a internet

uma abordagem top-down

6ª edição

- A Internet disponibiliza dois protocolos de transporte para aplicações, o UDP e o TCP.
- Requisitos de algumas aplicações de rede:

Aplicação	Perda de dados	Vazão	Sensibilidade ao tempo
Transferência / download de arquivo	Sem perda	Elástica	Não
E-mail	Sem perda	Elástica	Não
Documentos Web	Sem perda	Elástica (alguns kbits/s)	Não
Telefonia via Internet/ videoconferência	Tolerante à perda	Áudio: alguns kbits/s – 1Mbit/s Vídeo: 10 kbits/s – 5 Mbits/s	Sim: décimos de segundo
Áudio/vídeo armazenado	Tolerante à perda	Igual acima	Sim: alguns segundos
Jogos interativos	Tolerante à perda	Poucos kbits/s – 10 kbits/s	Sim: décimos de segundo
Mensagem instantânea	Sem perda	Elástico	Sim e não

Serviços de transporte providos pela Internet

KUROSE | ROSS

Redes de computadores e a internet

uma abordagem top-down

6ª edição

- Aplicações populares da Internet, seus protocolos de camada de aplicação e seus protocolos de transporte subjacentes:

Aplicação	Protocolo de camada de aplicação	Protocolo de transporte subjacente
Correio eletrônico	SMTP [RFC 5321]	TCP
Acesso a terminal remoto	Telnet [RFC 854]	TCP
Web	HTTP [RFC 2616]	TCP
Transferência de arquivos	FTP [RFC 959]	TCP
Multimídia em fluxo contínuo	HTTP (por exemplo, YouTube)	TCP
Telefonia por Internet	SIP [RFC 3261], RTP [RFC 3550] ou proprietária (por exemplo, Skype)	UDP ou TCP

Protocolos de camada de aplicação

KUROSE | ROSS

Redes de computadores e a internet

uma abordagem top-down

6ª edição

Um protocolo de camada de aplicação define:

- Os tipos de mensagens trocadas.
- A sintaxe dos vários tipos de mensagens, tais como os campos da mensagem e como os campos são delineados.
- A semântica dos campos, isto é, o significado da informação nos campos.
- Regras para determinar quando e como um processo envia mensagens e responde a mensagens.

Protocolos de camada de aplicação

KUROSE | ROSS

Redes de computadores e a internet

uma abordagem top-down

6ª edição

Distinção entre aplicações de rede e protocolos de camada de aplicação:

- Um protocolo de camada de aplicação é apenas um pedaço de uma aplicação de rede.
- A Web é uma aplicação cliente-servidor que permite aos usuários obter documentos de servidores Web.
- A aplicação Web consiste em muitos componentes, como um padrão para formatar documentos (HTML), browsers Web (Chrome, Firefox), servidores Web (Apache, IIS) e um protocolo de camada de aplicação (HTTP).

World Wide Web

- Até a década de 1990, a Internet era usada primordialmente por pesquisadores, acadêmicos e estudantes universitários.
 - Acessar e-mails, enviar e receber notícias, transferir arquivos, etc.
- No início da década de 1990 surge a World Wide Web.
 - Aplicação da Internet que chamou a atenção do público em geral.
 - Alçou a Internet de apenas mais uma entre muitas redes de dados para, essencialmente, a única rede de dados.

World Wide Web

- A Web funciona por demanda.
 - Usuários recebem o que querem, quando querem.
- Qualquer indivíduo pode fazer com que informações fiquem disponíveis na Web.

World Wide Web

- Uma página Web (documento) é constituída de objetos.
- Um objeto é um arquivo que se pode acessar com um único URL (arquivo HTML, imagem JPEG, etc.).
- A maioria das páginas Web é constituída de um arquivo-base HTML e diversos objetos referenciados.
- O arquivo-base HTML referencia os outros objetos na página com os URLs dos objetos.

World Wide Web

- Cada URL tem dois componentes:
 1. Nome do hospedeiro servidor que abriga o objeto.
 2. Nome do caminho do objeto.

- Exemplo:

`http://www.ufmg.br/ciencia_da_computacao/imagem.jpeg`

- **`www.ufmg.br`** é o nome do hospedeiro.
- **`/ciencia_da_computacao/imagem.jpeg`** é o nome do caminho

HTTP

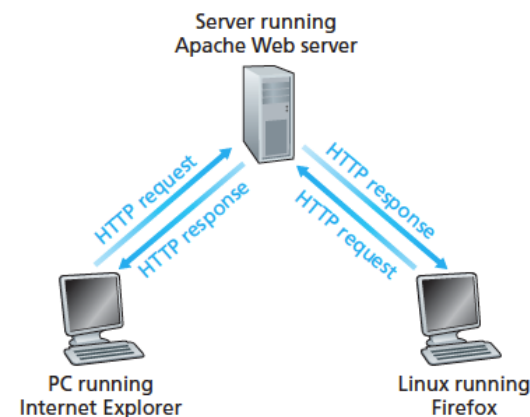
KUROSE | ROSS

Redes de computadores e a internet

uma abordagem top-down

6ª edição

- O HTTP (HyperText Transfer Protocol) é o protocolo da camada de aplicação da Web [RFC 1945] e [RFC 2616].
- O HTTP é executado em dois programas:
 1. Programa cliente.
 2. Programa servidor.
- O HTTP define como clientes Web requisitam páginas Web aos servidores e como eles as transferem a clientes.
- O HTTP usa o TCP como protocolo de transporte subjacente.



Conexões persistentes e não persistentes

KUROSE | ROSS

Redes de computadores e a internet

uma abordagem top-down

6ª edição

- O HTTP pode usar conexões persistentes e não persistentes:
- **Conexões não persistentes:** cada par de requisição/resposta deve ser enviado por uma conexão TCP distinta.
- **Conexões persistentes:** todas as requisições e suas respostas devem ser enviadas por uma mesma conexão TCP.
 - **Sem paralelismo:** cliente emite nova requisição somente quando a resposta anterior for recebida.
 - **Com paralelismo:** cliente pode fazer uma nova requisição antes de receber uma resposta a uma requisição anterior.

Formato da mensagem HTTP

KUROSE | ROSS

Redes de computadores e a internet

uma abordagem top-down

6ª edição

- O HTTP define o formato das mensagens trocadas entre um cliente HTTP e um servidor HTTP.
- Há dois tipos de mensagens HTTP:
 - Mensagens de **requisição**.
 - Mensagens de **resposta**.

Formato da mensagem HTTP

KUROSE | ROSS

**Redes de
computadores
e a internet**

uma abordagem top-down

6ª edição

Exemplo de uma mensagem de requisição HTTP:

GET /somedir/page.html HTTP/1.1

Host: www.someschool.edu

Connection: close

User-agent: Mozilla/5.0

Accept-language: fr

Formato da mensagem HTTP

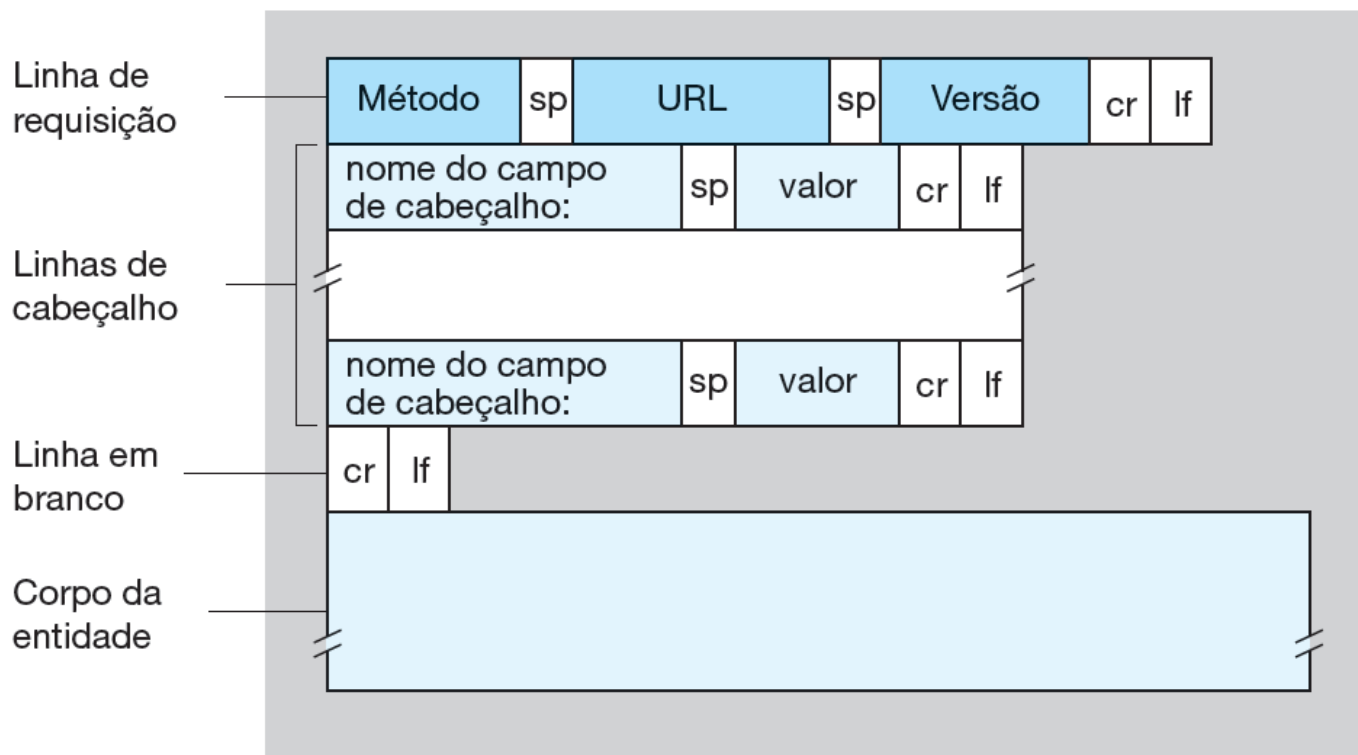
KUROSE | ROSS

Redes de computadores e a internet

uma abordagem top-down

6ª edição

- Formato geral de uma mensagem de requisição HTTP



Formato da mensagem HTTP

KUROSE | ROSS

Redes de computadores e a internet

uma abordagem top-down

6ª edição

- Alguns métodos HTTP:
 - **GET**: retorna o objeto requisitado.
 - **HEAD**: retorna apenas informações sobre o objeto.
 - **POST**: envia informação do navegador para o servidor (formulários).
 - **PUT**: envia objetos para o servidor.
 - **DELETE**: deleta objetos armazenados no servidor.

Formato da mensagem HTTP

KUROSE | ROSS

**Redes de
computadores
e a internet**

uma abordagem top-down

6ª edição

Exemplo de uma mensagem de resposta HTTP

HTTP/1.1 200 OK

Connection: close

Date: Tue, 09 Aug 2011 15:44:04 GMT

Server: Apache/2.2.3 (CentOS)

Last-Modified: Tue, 09 Aug 2011 15:11:03 GMT

Content-Length: 6821

Content-Type: text/html

(dados dados dados dados dados ...)

Formato da mensagem HTTP

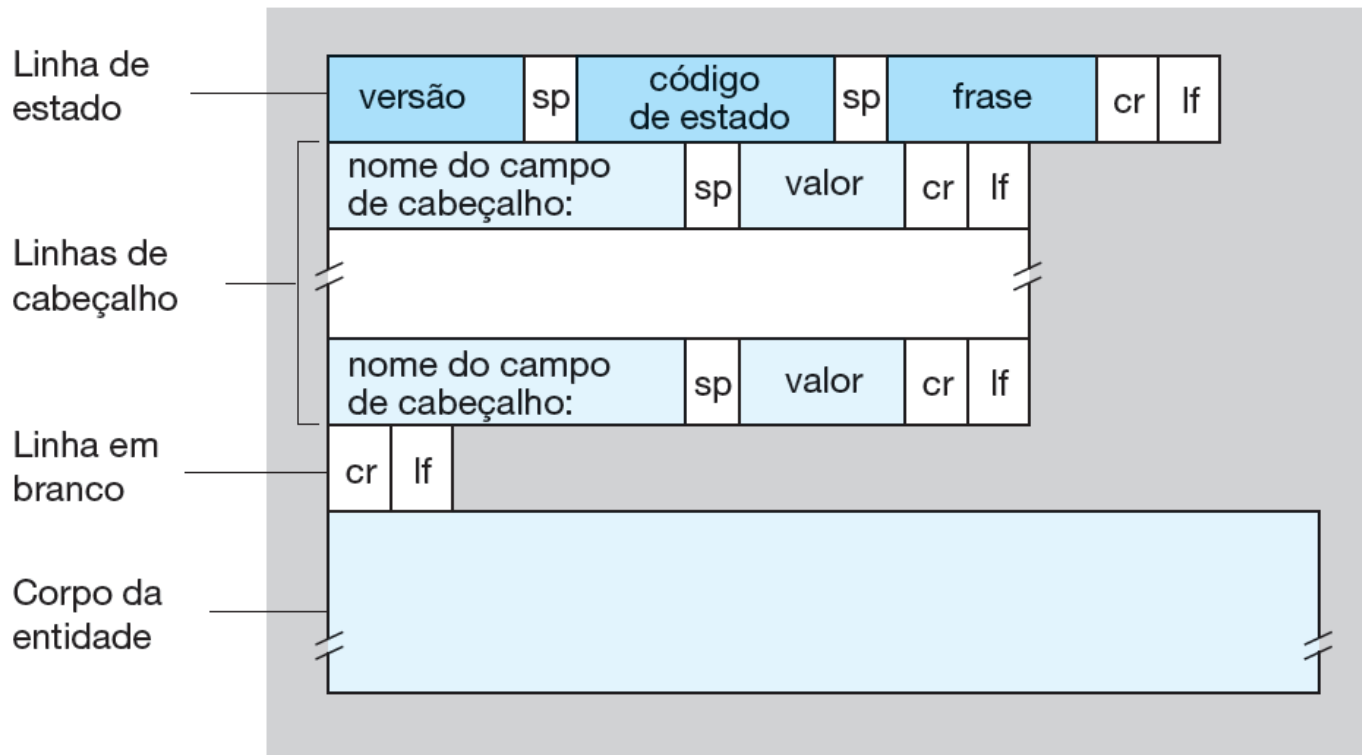
KUROSE | ROSS

Redes de computadores e a internet

uma abordagem top-down

6ª edição

- Formato geral de uma mensagem de resposta HTTP



Interação usuário-servidor: *cookies*

KUROSE | ROSS

**Redes de
computadores
e a internet**

uma abordagem top-down

6ª edição

- O HTTP é um protocolo **sem estado**.
 - Servidor não mantém informações sobre requisições anteriores realizadas por um cliente.
- Simplifica o projeto do servidor.
 - Permite o desenvolvimento de servidores Web de alto desempenho que podem manipular milhares de conexões TCP simultâneas.
- No entanto, é interessante que sites Web sejam capazes de identificar usuários.
 - Restrição de acesso de usuários.
 - Apresentação de conteúdo em função da identidade do usuário.

Interação usuário-servidor: *cookies*

KUROSE | ROSS

**Redes de
computadores
e a internet**

uma abordagem top-down

6ª edição

Cookies, definidos no [RFC 6265], permitem que sites monitorem seus usuários.

A tecnologia dos cookies tem quatro componentes:

1. uma linha de cabeçalho de cookie na mensagem de resposta HTTP;
2. uma linha de cabeçalho de cookie na mensagem de requisição HTTP;
3. um arquivo de cookie mantido no sistema final do usuário e gerenciado pelo navegador do usuário;
4. um banco de dados de apoio no site.

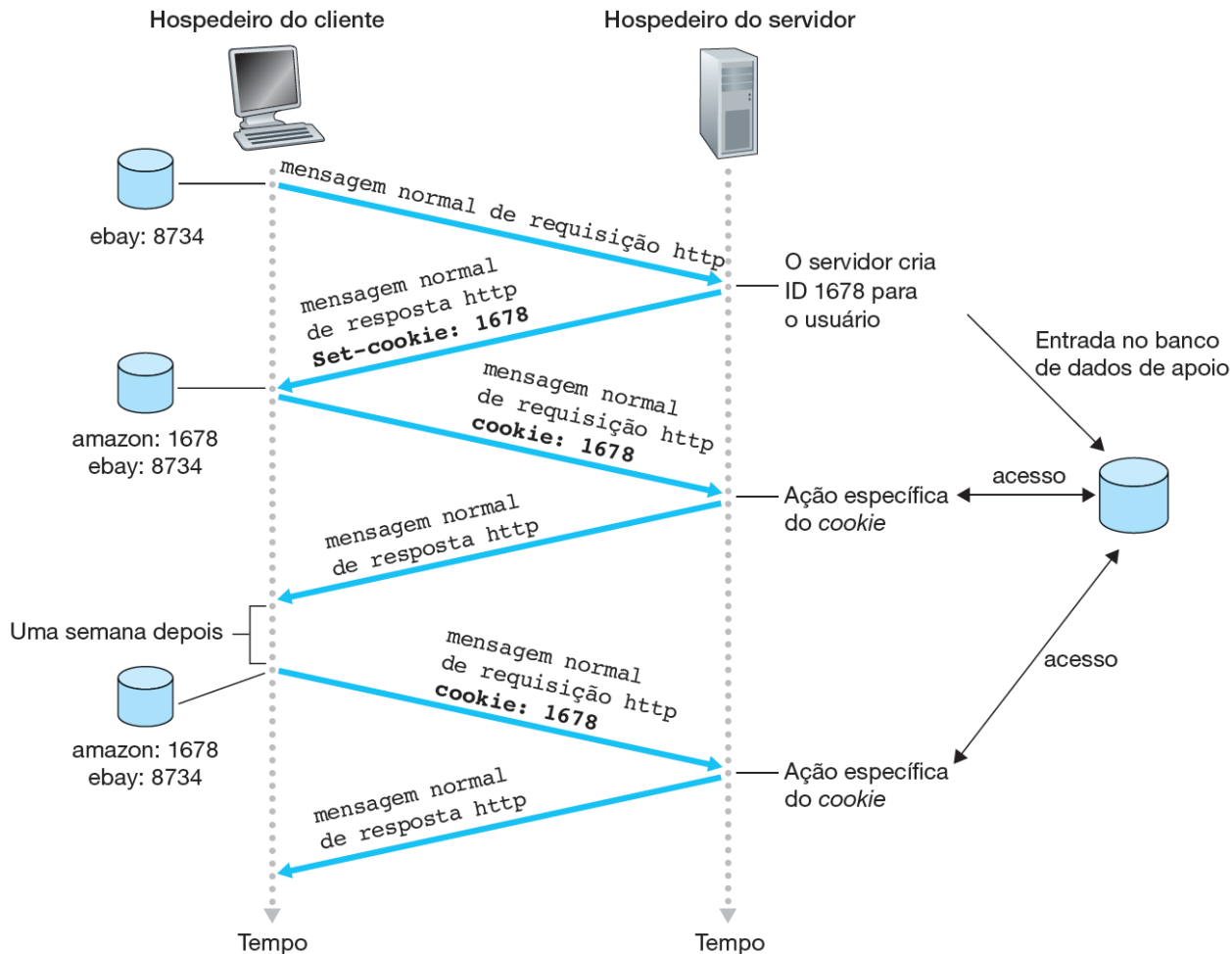
Interação usuário-servidor: *cookies*

KUROSE | ROSS

Redes de computadores e a internet

uma abordagem top-down

6ª edição



- Mantendo o estado do usuário com *cookies*.

Redes de computadores e a internet

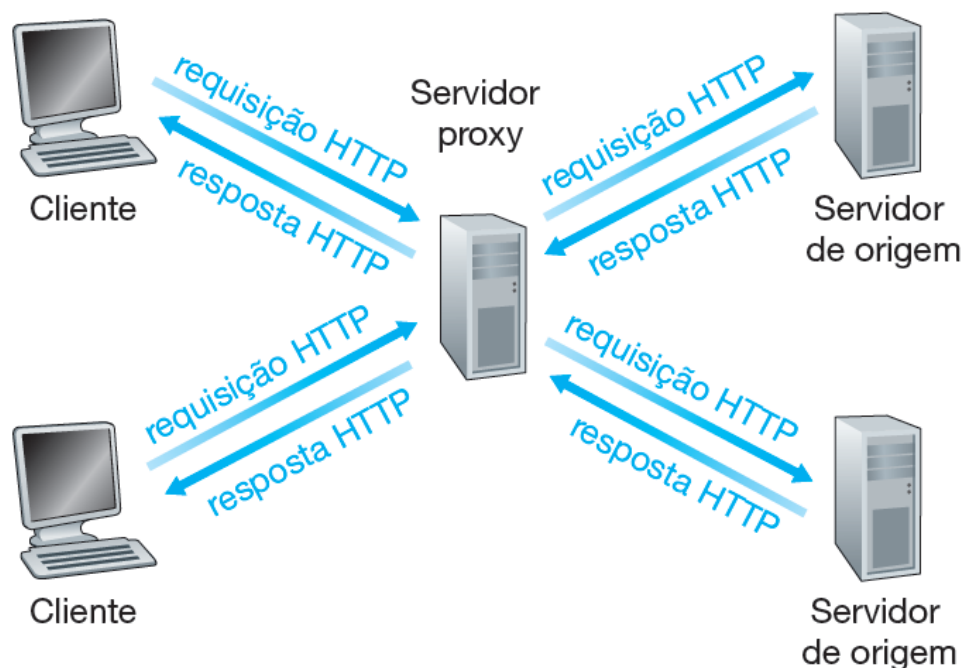
uma abordagem top-down

6ª edição

Caches Web

- Um **cache Web** — também denominado **servidor proxy** — é uma entidade da rede que atende requisições HTTP em nome de um servidor Web de origem.

Clientes requisitando objetos por meio de um *cache Web*:



GET condicional

- GET condicional – mecanismo que permite que um *cache* verifique se seus objetos armazenados localmente estão atualizados.
1. Inicialmente, em nome do browser, o cache envia uma requisição para o servidor Web.

```
GET /fruit/kiwi.gif HTTP/1.1  
Host: www.exotiquecuisine.com
```

2. O servidor Web responde com o objeto requisitado.

```
HTTP/1.1 200 OK  
Date: Sat, 8 Oct 2011 15:39:29  
Server: Apache/1.3.0 (Unix)  
Last-Modified: Wed, 7 Sep 2011 09:23:24  
Content-Type: image/gif
```

```
(data data data data data ...)
```

GET condicional

3. O cache encaminha o objeto para o browser e também armazena o objeto localmente. O cache também armazena a data em que o objeto foi modificado pela última vez.
4. Uma semana depois, um outro browser requisita o mesmo objeto pelo cache e o objeto ainda está armazenado no cache. Como o objeto pode ter sido modificado no servidor Web durante a última semana, o cache realiza uma verificação utilizando um GET condicional.

```
GET /fruit/kiwi.gif HTTP/1.1  
Host: www.exotiquecuisine.com  
If-modified-since: Wed, 7 Sep 2011 09:23:24
```


GET condicional

5. Assumindo que o objeto não foi modificado no servidor Web, ele responde ao cache informando que o objeto não foi modificado.

```
HTTP/1.1 304 Not Modified  
Date: Sat, 15 Oct 2011 15:39:29  
Server: Apache/1.3.0 (Unix)
```

(empty entity body)

6. Essa mensagem diz ao cache que ele pode encaminhar para o browser o objeto que ele possui armazenado localmente para o browser.