

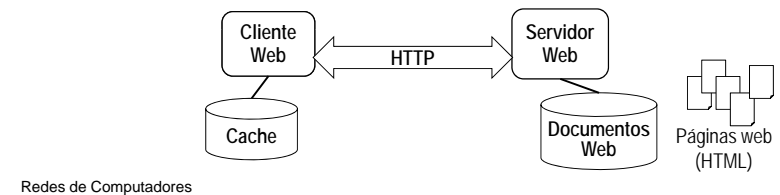
Redes de Computadores

HyperText Transfer Protocol

Aula 27

Introdução

- Serviço world wide web
 - Aplicação cliente-servidor
 - Publicação de documentos (servidor) e a recuperação e visualização de documentos (cliente)
 - Evoluiu para acesso a documentos multimídia
- Protocolo específico para interação cliente-servidor:
 - HTTP (*Hyper Text Transfer Protocol*)
- O cliente se encarrega de interpretar o documento HTML (página) e exibir para o usuário



Página web

- Documento web
 - Descrito em HTML (Hyper Text Markup Language)
- Composta por uma série de objetos
 - Arquivo base HTML, imagens, applets, cliques de vídeo, etc
 - e.x.: página com texto e cinco imagens possui seis objetos
- Acessível através de um URL (*Uniforme Resource Locator*)
 - Nome de um hospedeiro e porta
 - Caminho para um recurso no sistema de arquivos do hospedeiro
 - Protocolo (esquema) para buscar o recurso

Protocolo://nome_máquina:porta/recurso ⇨ <http://www.inf.ufrgs.br/~asc/redes>

↳ http, https, ftp, mailto, news, telnet, ldap, file

Tipos de documentos web

- Páginas estáticas
 - Documentos de conteúdo fixo criados e armazenados no servidor
 - Cliente obtém uma cópia do documento
 - Elaboradas com auxílio de linguagens (HTML, XML, XSL,)
- Páginas dinâmicas
 - Criado, por demanda, pelo servidor através de um aplicativo ou script
 - O conteúdo do documento varia de uma requisição a outra
 - Elaboradas com auxílio de linguagens (Java, JSP, ASP, CGI...)
- Páginas ativas
 - Possuem um programa que executa do lado cliente
 - Emprega linguagens de scripts como javascript ou applets java

Cliente Web

- Para o usuário, a web é conjunto de páginas
- Páginas são visualizadas por programas específicos:
 - Browsers (Netscape, Mozilla Firefox, MS Internet Explorer, Chrome, etc)
- *Browser*:
 - Possui três componentes: controlador, protocolos clientes e Intepretadores (HTML, java, javascritp...)
 - Permite carga e execução de páginas web de diferentes tipos

Servidor web

- Servidor faz o armazenamento dos objetos
 - Ex: Apache, Microsoft Internet Information Server, Netscape Enterprise server
- Interação baseada em dois elementos:
 - URL: Uniform Resource Locator
 - HTTP: HyperText Transfer Protocol
- Cada site *web* possui um *daemon* (servidor *web*)
- Espera requisições de clientes (conexões TCP)
 - Cliente envia uma requisição ao servidor
 - Servidor atende requisição e envia resposta
 - Conexão é desfeita*

Hyper Text Transfer Protocol (HTTP)

- Protocolo de aplicação da web
- Define forma como clientes solicitam páginas web aos servidores e como esses as transferem para os clientes
 - Determina apenas como os dados são transferidos: não indica, nem impõem tipos de dados e determina nenhum tipo de processamento
 - Browser é que determina como os dados são processados
- Emprega conexões TCP na porta 80 (*default*)
- Protocolo *stateless* (sem estado) e orientado a mensagens
 - Servidor não mantém nenhuma informação sobre o cliente
- Possuem três versões: HTTP 0.9, HTTP 1.0 e HTTP 1.1

Conexões não-persistentes

- Usada nas versões 0.9, 1.0 e 1.1
- Uma conexão para cada objeto
 - Cada conexão transporta exatamente uma requisição/resposta
 - Cliente abre a conexão, envia a requisição, o servidor envia a resposta e fecha a conexão (ex: uma página com 5 figuras envolve seis conexões, uma para página e uma para cada figura)
 - Conexões podem ser feitas em paralelo para páginas com mais de um objeto
 - Número máximo de conexões é uma configuração do cliente
- Desvantagens:
 - Tempo para estabelecimento
 - Consumo de recursos do sistema (memória e descritores)
 - Cada conexão sofre o processo de partida lenta (*slow start*) do TCP

Conexões persistentes

- Disponível na versão HTTP/1.1
- Uma conexão pode atender várias requisições/respostas
 - A conexão é fechada pelo servidor em resposta a uma requisição de encerramento feita pelo cliente ou por *time-out*
 - Configurado no servidor
- Duas versões:
 - Sem paralelismo: cliente envia requisição / após ter recebido respostas da requisição *i-1*
 - Com paralelismo: envio de várias requisições sem esperar pela resposta
- Questão: quando encerrar?
 - Por solicitação, time-out ou excesso de conexões abertas

Métodos HTTP

- HTTP foi criado para ser genérico
 - Tratamento de objetos e definições de métodos para manipulá-los
- Métodos suportados na versão 1.1 (visão do cliente)
 - ☆ GET: usado para solicitar a recuperação de um objeto
 - ☆ POST: similar ao GET, mas envia informações obtidas por formulários
 - PUT: solicita gravação de um objeto no servidor
 - PATCH: envia lista de modificações a serem feitas em um objeto
 - HEAD: solicita leitura apenas do cabeçalho da página
 - MOVE, COPY, DELETE, LINK e UNLIK: similares aos comandos de movimentação, cópia, remoção, criação e remoção de aliases em um sistema de arquivos
 - OPTION: consulta propriedades do servidor ou de um objeto

Formatos da mensagem HTTP

- Dois formatos básicos:
 - Requisição e resposta
 - Mensagem em ASC II (exceto o corpo da mensagem em alguns casos)
- Genericamente, ambas possuem três seções:
 - Linha de requisição ou de status
 - Linhas de cabeçalho
 - Corpo de entidade

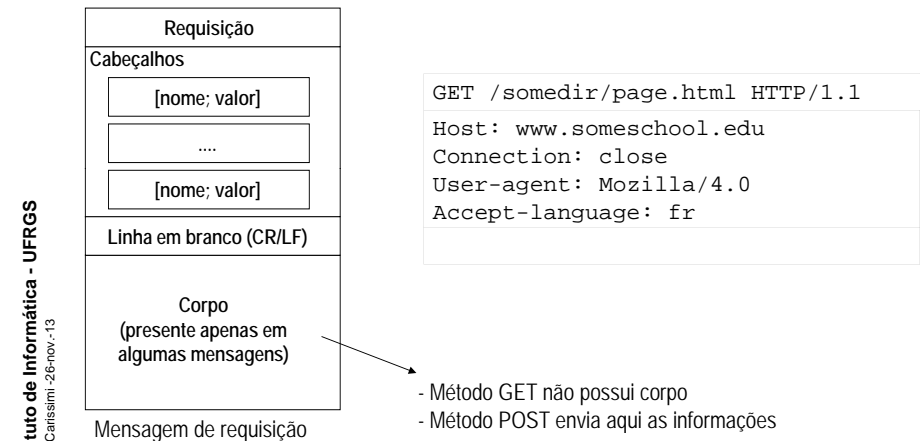
Mensagem de requisição

- Linha de requisição possui três campos:
 - Tipo da requisição (métodos)
 - GET, HEAD, POST, PUT, PATCH, COPY, MOVE, DELETE, LINK, UPLINK, OPTION
 - Uniform Resource Locator (URL): [protocolo] : // [máquina] : [porta] / [recurso]
 - Versão HTTP
- Linha(s) de cabeçalho
 - Conjunto de pares {nome de campo; valor}
- Corpo da entidade
 - Para métodos como POST que envia parâmetros para um formulário

Cabeçalhos da requisição

- Pares nome e valor
- Indicam ao servidor como realizar a requisição
- Exemplos:
 - *Host*: nome do servidor onde o objeto se encontra
 - *Connection*: se servidor deve encerrar ou não a conexão após enviar objeto
 - *User-agent*: tipo do cliente que está fazendo a requisição
 - *Accept-language*: idioma de preferência do usuário

Mensagem de requisição: formato e exemplo



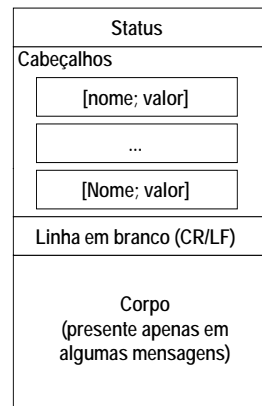
Mensagem de resposta

- Linha de status define:
 - Versão HTTP
 - Código de status
 - Frase explicativa do código de status
- Linhas de cabeçalho
 - Conjunto de pares {nome, valor}
- Corpo da entidade
 - Objeto solicitado

Códigos de status e cabeçalho de resposta

- **Códigos mais comuns**
 - 200: requisição processada com sucesso
 - 301: objeto solicitado foi movido
 - 400: erro genérico no processamento
 - 404: objeto solicitado não existe
 - 505: versão requisitada não é suportada
- **Cabeçalhos mais comuns**
 - *Connection*: informa ao cliente se o servidor encerrará ou não a conexão
 - *Date*: hora e data do envio do objeto
 - *Last-modified*: data e hora de criação ou modificação do objeto
 - *Server*: tipo do servidor:
 - *Content-length*: tamanho da mensagem em bytes
 - *Content-type*: tipo do objeto que está sendo enviado

Mensagens de resposta: formato e exemplo



```
HTTP/1.1 200 OK
Connection: close
Date: Thu, 06 Aug 1998 12:00:15 GMT
Server: Apache/1.3.0 (Unix)
Last-Modified: Mon, 22 Jun 1998 09:23:24 GMT
Content-Length: 6821
Content-Type: text/html

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

Mensagem de resposta

17

Multipurpose Internet Mail Extensions (MIME)

- Nem toda página (documento) é um HTML
 - A mensagem HTTP de resposta informa o tipo (usa padrão MIME)
- Necessário interpretar o conteúdo de acordo com o tipo
 - O navegador possui um interpretador
 - Necessário chamar um visualizador externo
- Estratégias para visualizadores externo
 - *Plug-in* : extensão do navegador (plugin para PDF, Flash, Quicktime...)
 - Execução de programa específico (e.g. power-point)
 - e.g.: *application/vnd.ms-powerpoint*

Redes de Computadores

18

Exemplo de sessão http

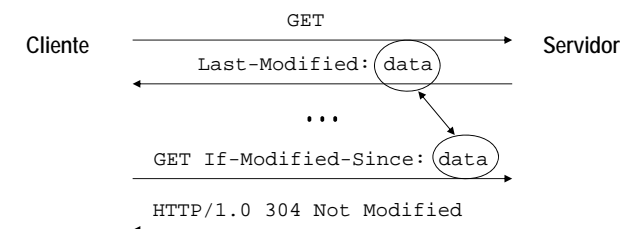
- Acesso a página
 - <http://www.inf.ufrgs.br/~asc/redes/http.html>
- Sessão wireshark
 - Arquivo http 1.1 exemplo



19

GET condicional

- Funcionamento baseado na cache cliente
- Cópia na cache pode estar desatualizada
 - Emprega cabeçalho *If-modified-since*

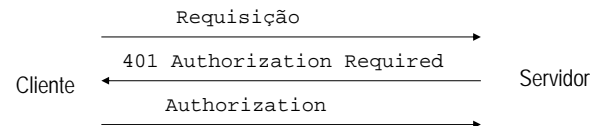


Redes de Computadores

20

Interação usuário-servidor: autenticação

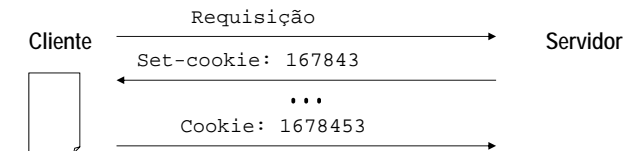
- Mecanismo para identificar usuários
 - Exige a definição de um nome de usuário e uma senha
 - Codificação de status e cabeçalhos específicos para tal



21

Interação usuário-servidor: *cookies*

- Armazenam informações dos usuários
 - Criado e armazenado no servidor ao receber uma requisição *http*
 - É um registro associando usuário ao *cookie*
 - *Cookie* é enviado na resposta e armazenado em um diretório mantido pelo *browser* do usuário
- Usos de *cookies*
 - Identificar usuários sem necessitar nome e senha
 - Lembrar preferências de um usuário (carrinhos de compras, publicidade, etc)
- Problema: usuários nômades (armazenamento é local)



22

Alguns detalhes sobre *cookies*

- Definido na RFC 2109
- Um *string* de até 4 KB que o servidor cria e associa a um navegador
- Possui até 5 campos:
 - Domínio: site de onde o *cookie* foi gerado
 - Caminho: diretório no servidor onde o *cookie* é válido
 - Conteúdo: valor
 - Data expiração: *cookies* persistentes ou não (hora GMT)
 - Segurança: enviar *cookie* apenas se conexão for via SSL (sim ou não)
- Existem *cookies* que rastreiam usuários e sites (*spyware*)

23

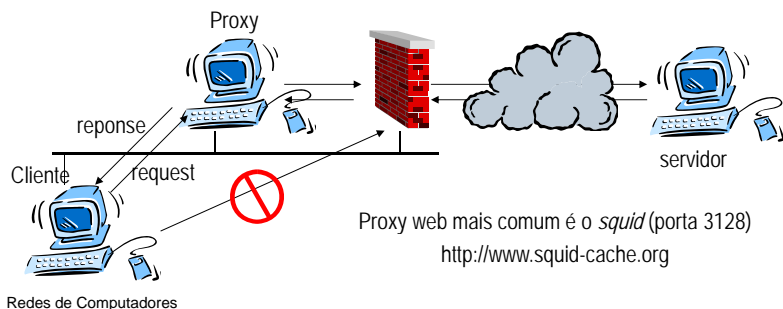
Cache no cliente web

- Navegador web mantém uma cópia no disco das páginas (recursos) usadas recentemente
- Quando desconsiderar?
 - Estratégia 1:
 - Combinação Last-Modified + Date
 - Expires + Date
 - Heurísticas
 - Estratégia 2:
 - Baseada em if-Modified + Etag + If-none-Match
- Diretiva no-cache para páginas dinâmicas e páginas com autenticação

24

Proxy Web

- Melhorar o desempenho realizando *caching* das páginas acessadas
 - Reduz carga no servidor, diminui tráfego na rede, diminui latência
- Contribui para a segurança
 - Máquina cliente não é visível do exterior
 - Possibilidade de monitorar sites, palavras chaves, etc



25

Hyper Text Transfer Protocol over SSL (HTTPS)

- O HTTP não oferece nenhum tipo de segurança
 - As mensagens HTTP são em ASCII
 - Não garantias quanto a sua autenticidade, integridade e confidencialidade
- Protocolo de aplicação da web que utiliza criptografia
 - Garante confidencialidade (pela cifragem de dados)
 - Arquivo https exemplo
 - Garante a identidade das partes envolvidas em uma transferência (pelo uso de chaves públicas e privadas)
- Emprega a porta 443/TCP invés da 80

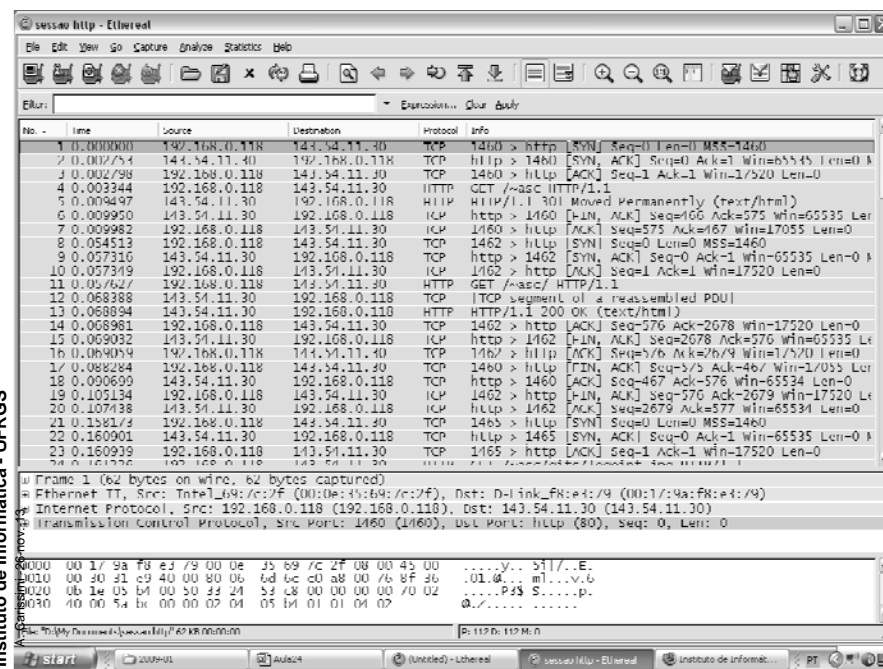
https://nome_máquina/recurso

Redes de Computadores

26

Leituras complementares

- Tanenbaum, A. *Redes de Computadores* (4ª edição), Campus 2003.
 - Capítulo 7, seção 7.3.4
- Carissimi, A.; Rochol, J.; Granville, L.Z.; *Redes de Computadores*. Série Livros Didáticos. Bookman 2009.
 - Capítulo 7, seção 7.3



Redes de Computadores

28

Parque de servidores

- Forma de oferecer melhor desempenho para acesso a servidores web sob alta carga de acesso
 - Balanceamento de carga
 - Distribuição geográfica
- Duas estratégias:
 - Rodízio IP via DNS
 - Uso de front-end para direccionar requisições (middlebox)