Redes de Computadores

Camada de Aplicação

Agenda

Web e HTTP
Correio Eletrônico

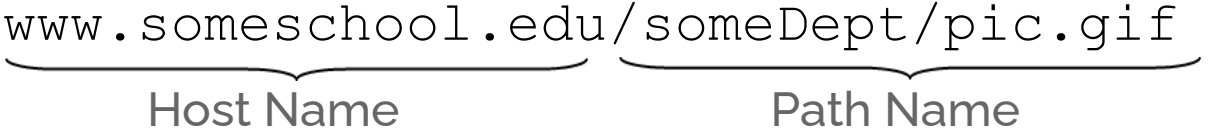
1

Web e HTTP

Definições, Conexões Persistentes e Não-Persistentes, Formato de Mensagem, Cookies, Caches

Visão Geral do HTTP

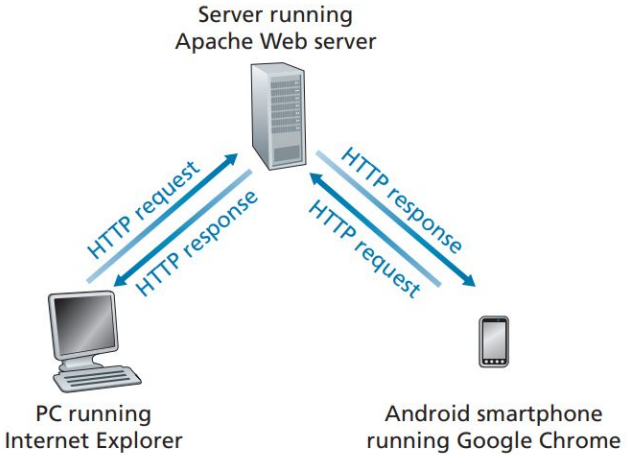
- Uma página da Web consiste em vários arquivos HTML incluindo objetos referenciados, cada um deles acessado via URL, por exemplo



- Acessível com protocolo HTTP

Visão Geral do HTTP

- Hypertext Transfer Protocol
- Arquitetura Cliente-Servidor
  - Cliente: um browser faz requisições e recebe respostas, para exibir objetos Web
  - Servidor: responde requisições enviando objetos Web



Visão Geral do HTTP

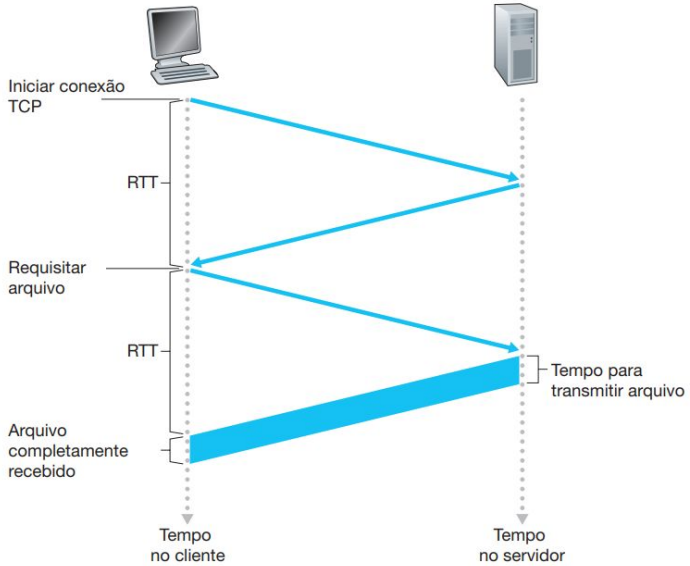
- Utiliza Transporte TCP
  - Cliente inicia conexão TCP (cria socket) para o servidor na porta 80
  - Servidor aceita uma conexão TCP do cliente
  - Mensagens HTTP (mensagens do protocolo de camada de aplicação) são trocadas entre o browser (cliente HTTP) e o servidor Web (servidor HTTP)
  - A conexão TCP é fechada

Visão Geral do HTTP

- HTTP é um protocolo **stateless**
  - Por default, o servidor não mantém informação sobre os pedidos passados pelos clientes
  - Protocolos que mantêm informações de “estado” são complexos!
    - Histórico do passado (estado) deve ser mantido
    - Se o servidor/cliente quebra, suas visões de “estado” podem ser inconsistentes, devendo ser reconciliadas

Tempo de Resposta HTTP

- Round-Trip Time — RTT: tempo para enviar um pequeno pacote entre o cliente e o servidor e retornar
- Tempo de resposta: Um RTT para iniciar a conexão TCP
- Um RTT para requisição HTTP e 1os bytes da resposta HTTP para retorno
- Tempo de transmissão de arquivo



Total = 2 RTT+ tempo de transmissão

Conexões do HTTP

- Conexão Não-Persistente
  - Um objeto é enviado por vez sobre uma conexão TCP
  - Download de múltiplos objetos requer múltiplas conexões
- Conexão Persistente
  - Múltiplos objetos podem ser enviados sobre uma única conexão TCP entre o cliente e o servidor

HTTP Não Persistente

- Requer 2 RTTs por objeto
- Sistema Operacional deve manipular e alocar recursos do hospedeiro para cada conexão TCP
    - Os browsers frequentemente abrem conexões TCP paralelas para buscar objetos referenciados

HTTP Persistente

- Servidor deixa a conexão aberta após enviar uma resposta
- Mensagens HTTP subsequentes entre o mesmo cliente/servidor são enviadas pela conexão

HTTP Persistente

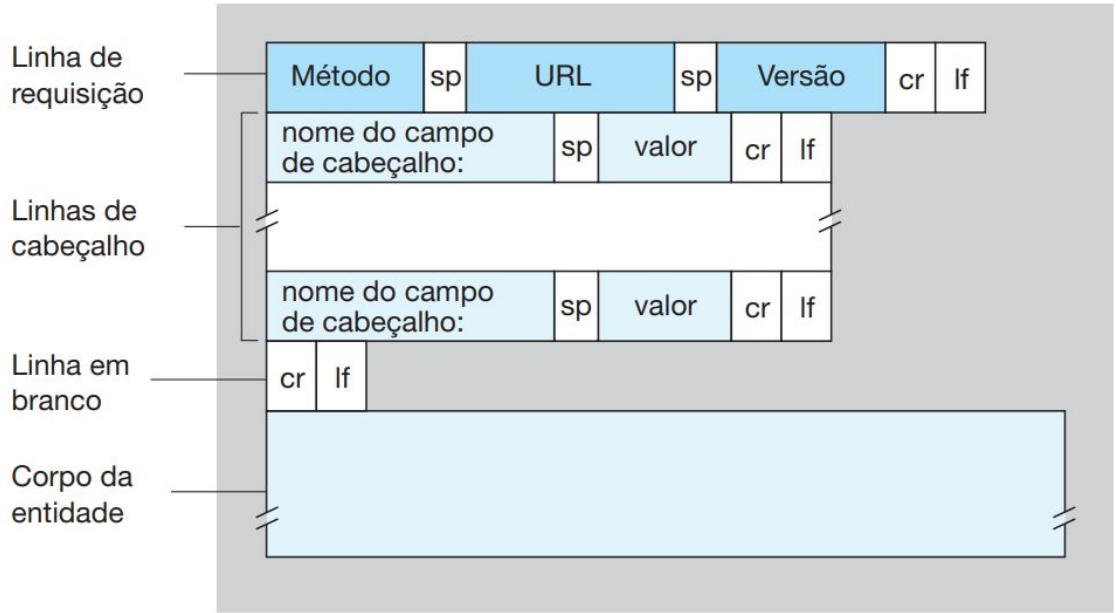
- Persistente sem pipelining
  - O cliente emite novas requisições apenas quando a resposta anterior for recebida
  - Um RTT para cada objeto referenciado
- Persistente com pipelining
  - Padrão no HTTP/1.1
  - O cliente envia requisições assim que encontra objeto referenciado
  - Tão curto quanto um RTT para todos os objetos referenciados

Formato de Mensagem

- Dois tipos de mensagens HTTP: request, response
  - HTTP **request message**: ASCII (formato legível para humanos)
- Linha de Requisição
- Linhas de Cabeçalho
- Carriage Return (linha em branco, fim da mensagem)

```
GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
Connection: close
User-agent: Mozilla/5.0
Accept-language: fr
```

Formato de Mensagem



Formato de Mensagem

- Método GET
  - A entrada é enviada no campo de URL da linha de requisição
- Método POST
  - Página Web freqüentemente inclui entrada de formulário
  - A entrada é enviada para o servidor no corpo da entidade
- Método HEAD
  - Solicita cabeçalhos (somente) que seriam retornados se a URL especificada fosse solicitada com um método HTTP GET

Formato de Mensagem

- Método PUT
  - Carrega novo objeto para o servidor, substituindo o arquivo existente na URL especificada pelo conteúdo no corpo da entidade
- Método DELETE
  - Apaga o arquivo especificado no campo de URL
- Método TRACE
  - Loop-back em nível de aplicação (request é enviada no corpo da resposta)

Formato de Mensagem

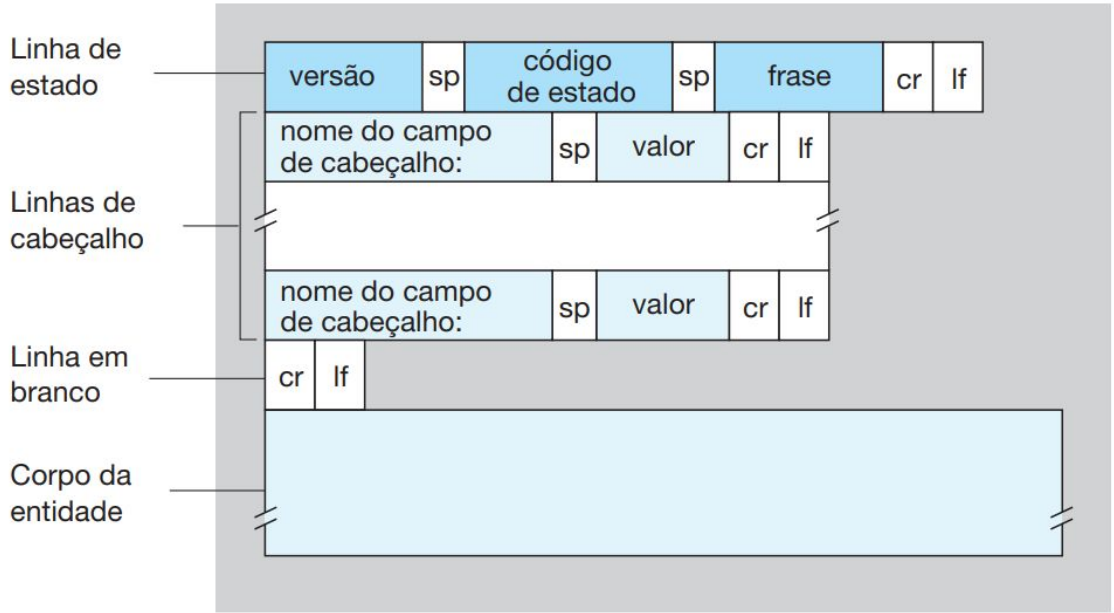
Método	Descrição
GET	Lê uma página Web
HEAD	Lê um cabeçalho de página Web
POST	Acrescenta algo a uma página Web
PUT	Armazena uma página Web
DELETE	Remove a página Web
TRACE	Ecoa a solicitação recebida
CONNECT	Conecta através de um proxy
OPTIONS	Consulta opções para uma página

Formato de Mensagem

- HTTP response message
- Linha de Status
- Linhas de Cabeçalho
- Carriage Return (linha em branco, fim da mensagem)

```
HTTP/1.1 200 OK
Connection: close
Date: Tue, 18 Aug 2015 15:44:04 GMT
Server: Apache/2.2.3 (CentOS)
Last-Modified: Tue, 18 Aug 2015 15:11:03 GMT
Content-Length: 6821
Content-Type: text/html
(data data data data data ...)
```

Formato de Mensagem



Formato de Mensagem

- 200 OK: requisição bem-sucedida e a informação é entregue com a resposta
- 301 Moved Permanently: objeto requisitado foi removido em definitivo; novo URL é especificado no cabeçalho Location da mensagem de resposta. O software do cliente recuperará automaticamente o novo URL

Formato de Mensagem

- 400 Bad Request: código genérico de erro que indica que a requisição não pôde ser entendida pelo servidor
- 404 Not Found: o documento requisitado não existe no servidor
- 505 HTTP Version Not Supported: a versão do protocolo HTTP requisitada não é suportada pelo servidor

- HTTP Status Codes (link)

Formato de Mensagem

Código	Significado	Exemplos
1xx	Informação	100 = servidor concorda em tratar da solicitação do cliente
2xx	Sucesso	200 = solicitação com sucesso; 204 = nenhum conteúdo presente
3xx	Redirecionamento	301 = página movida; 304 = página em cache ainda válida
4xx	Erro do cliente	403 = página proibida; 404 = página não localizada
5xx	Erro do servidor	500 = erro interno do servidor; 503 = tente novamente mais tarde

Formato de Mensagem

Cabeçalho	Tipo	Conteúdo
User-Agent	Solicitação	Informações sobre o navegador e sua plataforma
Accept	Solicitação	O tipo de páginas que o cliente pode manipular
Accept-Charset	Solicitação	Os conjuntos de caracteres aceitáveis para o cliente
Accept-Encoding	Solicitação	As codificações de páginas que o cliente pode manipular
Accept-Language	Solicitação	Os idiomas com os quais o cliente pode lidar
If-Modified-Since	Solicitação	Data e hora para verificar atualização
If-None-Match	Solicitação	Tags enviadas anteriormente para verificar atualização
Host	Solicitação	O nome DNS do servidor
Authorization	Solicitação	Uma lista das credenciais do cliente
Referrer	Solicitação	O URL anterior do qual a solicitação veio
Cookie	Solicitação	Cookie previamente definido, enviado de volta ao servidor
Set-Cookie	Resposta	Cookie para o cliente armazenar
Server	Resposta	Informações sobre o servidor
Content-Encoding	Resposta	Como o conteúdo está codificado (p. ex., *gzip*)
Content-Language	Resposta	O idioma usado na página
Content-Length	Resposta	O tamanho da página em bytes

Formato de Mensagem

Cabeçalho	Tipo	Conteúdo
Content-Type	Resposta	O tipo MIME da página
Content-Range	Resposta	Identifica uma parte do conteúdo da página
Last-Modified	Resposta	Data e hora da última modificação na página
Expires	Resposta	Data e hora de quando a página deixa de ser válida
Location	Resposta	Informa para onde o cliente deve enviar sua solicitação
Accept-Ranges	Resposta	Indica que o servidor aceitará solicitações de intervalos de bytes
Date	Ambos	Data e hora em que a mensagem foi enviada
Range	Ambos	Identifica uma parte de uma página
Cache-Control	Ambos	Diretivas para o modo de tratar caches
ETag	Ambos	Tag para o conteúdo da página
Upgrade	Ambos	O protocolo para o qual o transmissor deseja passar

Uso de Cookies

- A maioria dos grandes sites Web utilizam cookies
- Quatro componentes:
  - Linha de cabeçalho do cookie na mensagem HTTP response
  - Linha de cabeçalho de cookie na mensagem HTTP request
  - Arquivo de cookie mantido no hospedeiro do usuário e manipulado pelo browser do usuário
  - Banco de dados backend no site Web

Uso de Cookies

- O que os cookies podem trazer:
  - Autorização
  - Cartões de compra
  - Recomendações
  - Estado de sessão do usuário (Web e-mail)

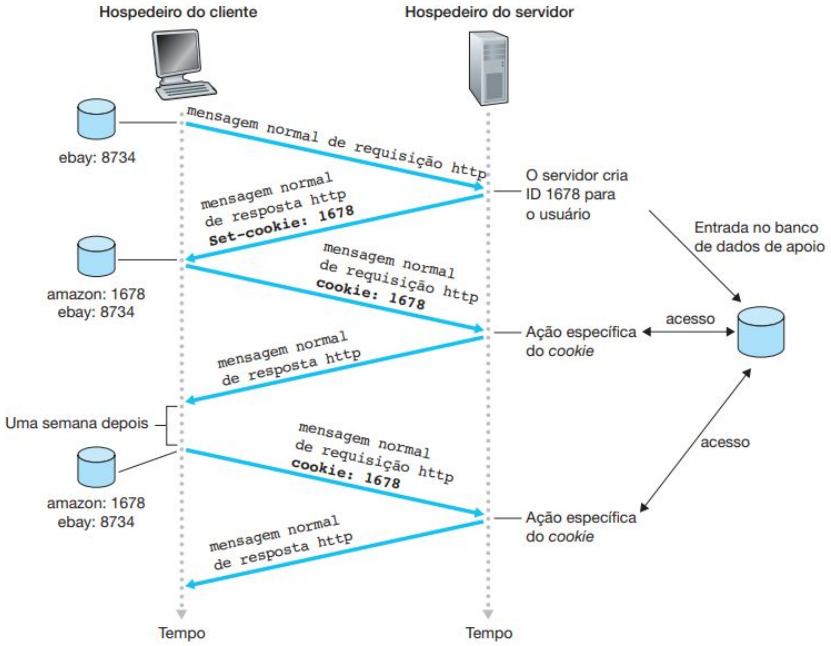
Uso de Cookies

- Cookies permitem que sites saibam muito sobre você
- Mecanismos de busca usam redirecionamento e cookies para saberem mais sobre você
- Empresas de propaganda obtêm informações por meio dos sites

Uso de Cookies

- Exemplo: Joe acessa a Internet sempre do mesmo PC
- Ela visita um site específico de comércio eletrônico pela primeira vez
- Quando a requisição HTTP inicial chega ao site, este cria um identificador (ID) único e uma entrada no banco de dados backend para este ID

Uso de Cookies



Uso de Cookies

- Cookies permitem que sites saibam muito sobre você
- Mecanismos de busca usam redirecionamento e cookies para saberem mais sobre você
- Empresas de propaganda obtêm informações por meio dos sites

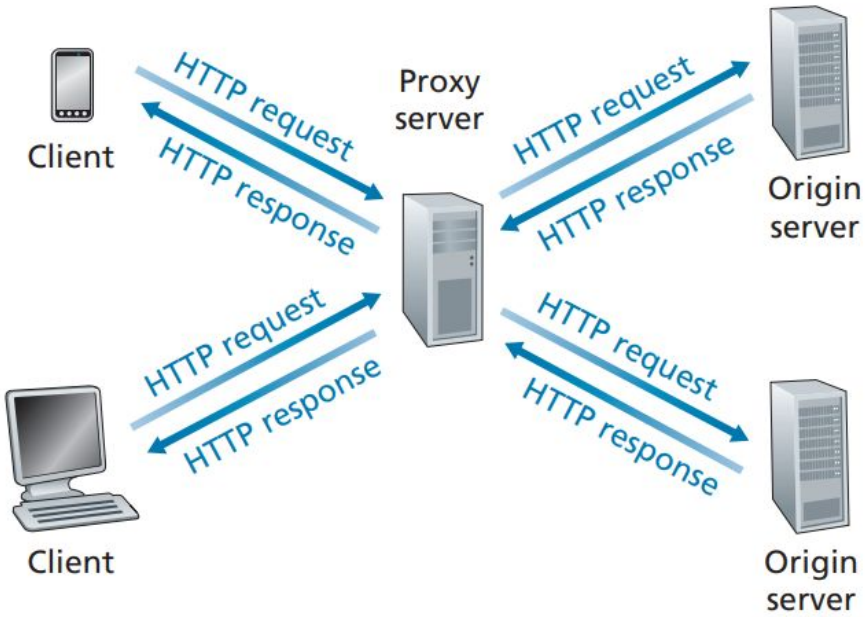
Uso de Cache

- Objetivo: atender o cliente sem envolver o servidor Web originador da informação
  - Uso de Proxy Servers
- A cache atua tanto no servidor como no cliente
- Tipicamente, a cache é instalada pelo ISP (acadêmico, empresarial ou residencial)

Uso de Cache

- Usuário configura o browser: o acesso Web é feito por meio de um procurador (proxy)
- Cliente envia todos os pedidos HTTP para o proxy
- Se o objeto existe na cache do proxy, ele retorna o objeto
- Caso contrário, o procurador solicita o(s) objeto(s) do servidor original, e então envia o(s) objeto(s) ao cliente

Uso de Cache


Client
HTTP request
HTTP response
Proxy server
HTTP request
HTTP response
Origin server
Client
HTTP request
HTTP response
HTTP request
HTTP response
Origin server

Uso de Cache

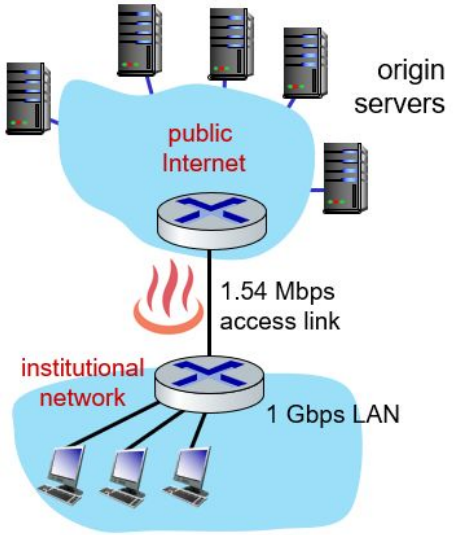
- Por que Web caching?
  - Reduz o tempo de resposta para a requisição do cliente
  - Reduz o tráfego em um enlace de acesso de uma instituição
    - Links externos podem ser caros e facilmente congestionáveis
  - A densidade de caches na Internet habilita os “fracos” provedores de conteúdo a efetivamente entregarem o conteúdo (mas fazendo P2P file sharing)

Uso de Cache

- Exemplo:
  - Taxa de acesso: 1.54 Mbps
  - RTT do roteador institucional para o servidor: 2s
  - Tamanho do objeto: 100 Kb
  - Média do acesso do browser para o servidor de origem: 15 req/s
    - Média de dados para os browsers: 1.5 Mbps

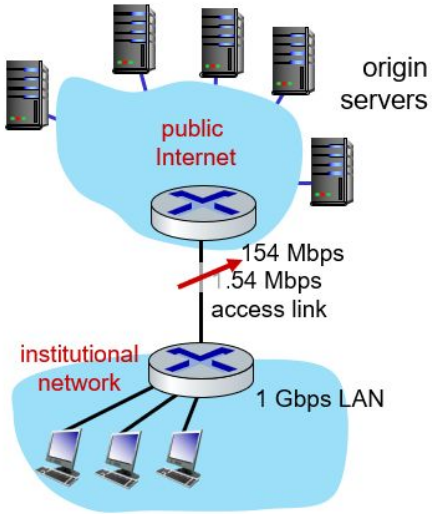
Uso de Cache

- Exemplo:
  - Performance: Utilização do acesso: 97%
  - Utilização da LAN: 0,15 %
  - Delay = Internet + Link de Acesso + LAN = 2 s + **minutos** + n s



Uso de Cache

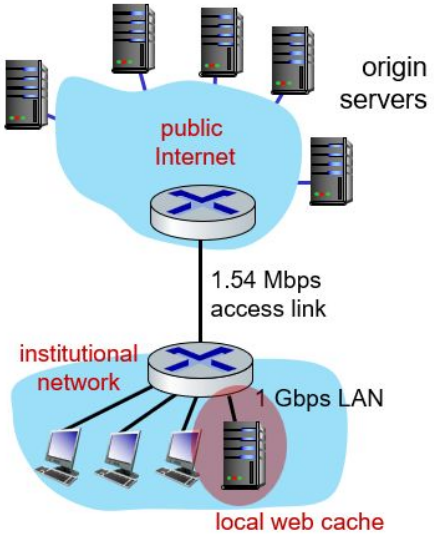
- Solução 1: Comprar um acesso mais rápido:
  - 1.54 => 154 Mbps
  - Performance: Utilização do acesso: 0,97%
  - Utilização da LAN: 0,15 %
  - Delay = Internet + Link de Acesso + LAN = 2 s + ms + n s



Uso de Cache

- Solução 2: Instalar um Web Caching
  - Solução mais barata
  - Considerando a permanência do mesmo link e um cache de 40%
    - 40% das requisições estarão em cache
    - 60% virão do servidor de origem
      - Taxa de acesso do browser

        0.6 * 1.50 Mbps = 0.9 Mbps
      - Utilização do link = 0.9/1.54 = 58 %

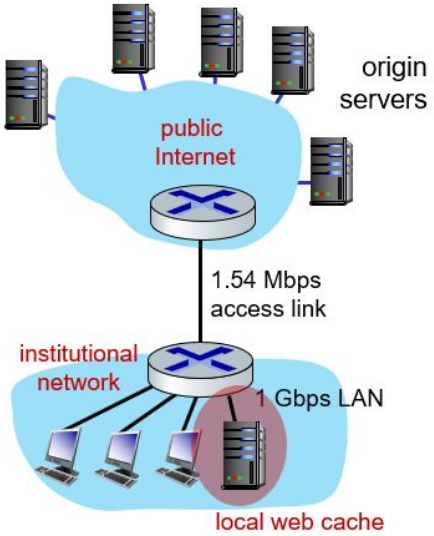


Uso de Cache

- Solução 2: Instalar um Web Caching
  - Média de Delay:
    - 0.6 * (delay do servidor de origem)

      \+ 0.4 * (delay quando em cache)

      = 0.6 (2.01) + 0.4 * (~ms) = ~1.2 s



GET Condicional

- Objetivo: não enviar objetos se já existe uma versão em cache atualizada
  - Não há delay de transmissão (ou uso de recursos de rede)
- Cliente: especifica a data da cópia em cache na requisição HTTP
  - If-modified-since: <date>
- Servidor: Resposta não contém o objeto se a cópia em cache é atualizada
  - HTTP/1.0 304 Not Modified

HTTP/2

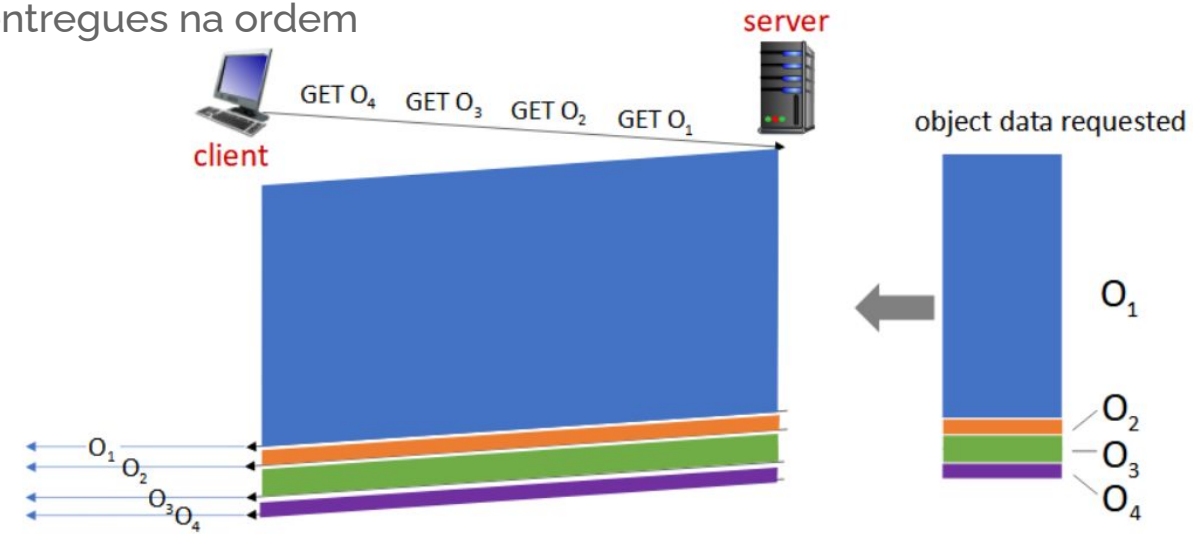
- Objetivo: reduzir o atraso em solicitações HTTP de vários objetos
- HTTP/1.1: introduziu vários GETs em pipeline em uma única conexão TCP
  - O servidor responde em ordem (FCFS: agendamento por ordem de chegada) a solicitações GET
  - Com FCFS, o objeto pequeno pode ter que esperar pela transmissão (bloqueio de cabeçalho de linha (HOL)) atrás de objeto(s) grande(s)
  - Prevê recuperação de perda (retransmissão de segmentos TCP perdidos) paralisa a transmissão de objetos

HTTP/2

- Objetivo: reduzir o atraso em solicitações HTTP de vários objetos
- HTTP/2: Maior flexibilidade no servidor no envio de objetos para o cliente
  - Métodos, códigos de status, a maioria dos campos de cabeçalho inalterados em relação ao HTTP/1.1
  - Ordem de transmissão de objetos solicitados com base na prioridade de objeto especificada pelo cliente (não necessariamente FCFS)
  - Envia objetos não solicitados para o cliente
  - Divide objetos em quadros, agendar quadros para mitigar o bloqueio de HOL

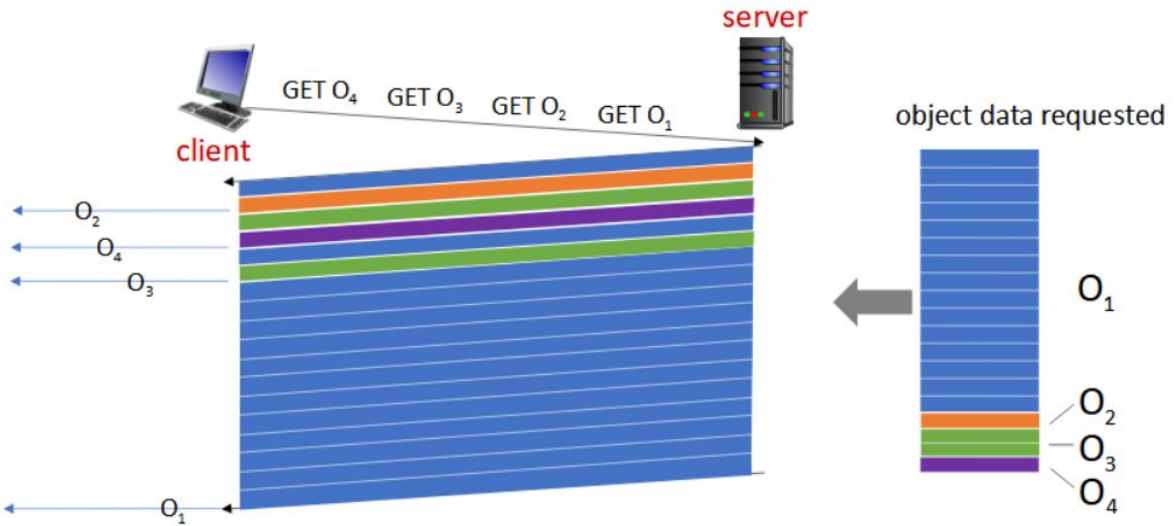
HTTP/2

- HTTP/1.1: O cliente requisita 1 objeto grande e 3 pequenos (Os objetos são entregues na ordem



HTTP/2

- HTTP/2: Os objetos são divididos em frames, transmitidos de forma intercalada



HTTP/2 para HTTP/3

- HTTP/2 sobre uma única conexão TCP significa a recuperação da perda de pacotes ainda paralisa todas as transmissões de objetos
  - Como no HTTP 1.1, os navegadores têm incentivo para abrir várias conexões TCP paralelas para reduzir a paralisação, aumentar a taxa de transferência geral
- Sem segurança sobre a conexão TCP
- HTTP/3: adiciona segurança, por erro de objeto e controle de congestionamento (mais pipelining) sobre UDP
  - Veremos mais sobre HTTP/3 na camada de transporte

2

Correio Eletrônico

Visão Geral, SMTP, Comparação com HTTP, Formatos de Mensagem, Protocolos de Acesso

Visão Geral

- Três componentes principais
  - Agentes de usuário
  - Servidores de correio
  - Simple mail transfer protocol (SMTP)

Visão Geral

- Agente de usuário
  - “leitor de correio”
  - Composição, edição, leitura de mensagens de correio
  - Ex.: Eudora, Outlook, elm, Netscape Messenger, Thunderbird, iPhone Mail Client
  - Mensagens de entrada e de saída são armazenadas no servidor

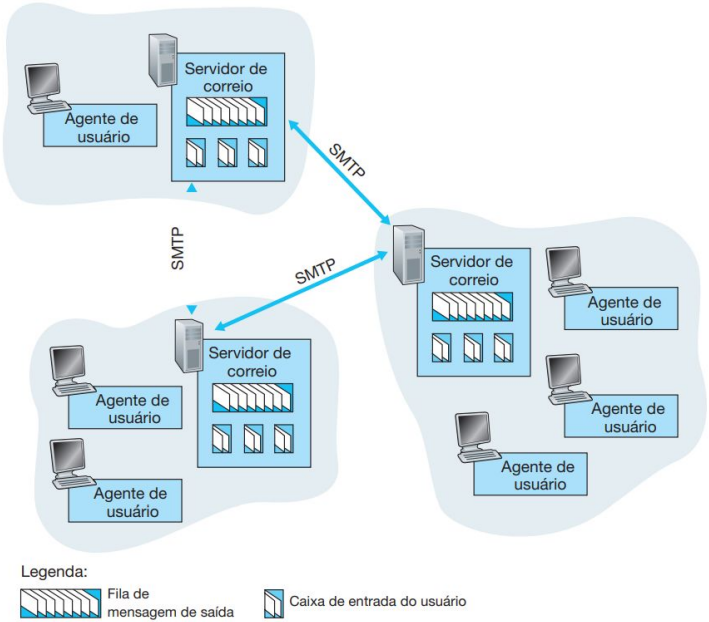
Visão Geral

- Servidores de correio
  - Caixa postal contém mensagens que chegaram (ainda não lidas) para o usuário
  - Fila de mensagens contém as mensagens de correio a serem enviadas

Visão Geral

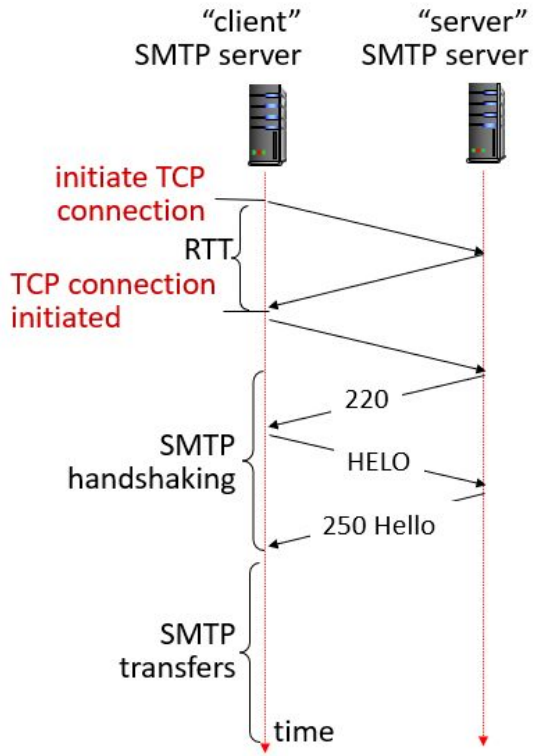
- Protocolo SMTP
  - Permite aos servidores de correio trocarem mensagens entre si
  - Cliente: servidor de correio que envia
  - “Servidor”: servidor de correio que recebe

Visão Geral


Servidor de correio
Agente de usuário
SMTP
SMTP
SMTP
Servidor de correio
Agente de usuário
Agente de usuário
Agente de usuário
Servidor de correio
Agente de usuário
Agente de usuário
Legenda:
Fila de mensagem de saída
Caixa de entrada do usuário

Protocolo SMTP

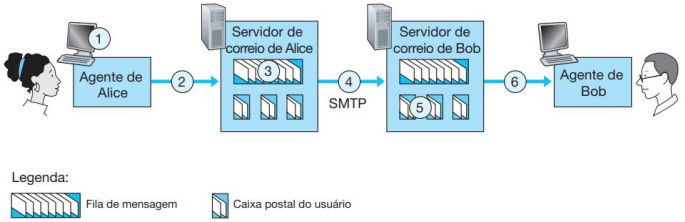
- Usa TCP para transferência confiável de mensagens de correio do cliente ao servidor, porta 25
- Transferência direta: servidor que envia para o servidor que recebe
- Três fases de transferência
  - Handshaking (apresentação)
  - Transferência de mensagens
  - Fechamento



Protocolo SMTP

- Interação comando/resposta
  - Comandos: texto ASCII
  - Resposta: código de status e frase
- Mensagens devem ser formatadas em código ASCII de 7 bits

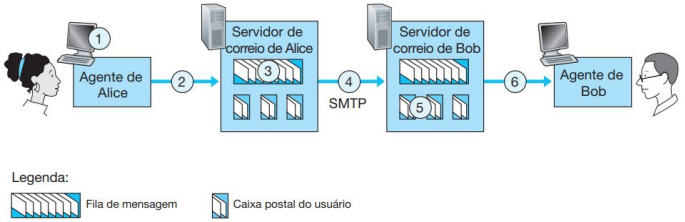
Protocolo SMTP



- Exemplo:

1. Alice chama seu agente de usuário para e-mail, fornece o endereço de Bob (por exemplo, bob@someschool.edu), compõe uma mensagem e instrui o agente de usuário a enviá-la.
2. O agente de usuário de Alice envia a mensagem para seu servidor de correio, onde ela é colocada em uma fila de mensagens.

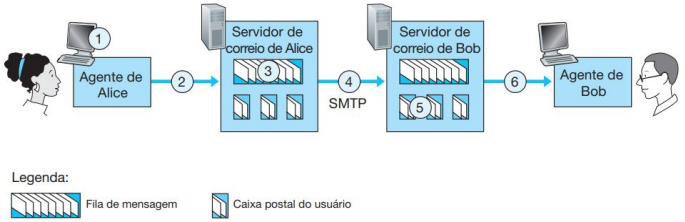
Protocolo SMTP



- Exemplo:

3. O lado cliente do SMTP, que funciona no servidor de correio de Alice, vê a mensagem na fila e abre uma conexão TCP para um servidor SMTP, que funciona no servidor de correio de Bob.
4. Após alguns procedimentos iniciais de apresentação (handshaking), o cliente SMTP envia a mensagem de Alice pela conexão TCP.

Protocolo SMTP



- Exemplo:

5. No servidor de correio de Bob, o lado servidor do SMTP recebe a mensagem e a coloca na caixa postal dele.
6. Bob chama seu agente de usuário para ler a mensagem quando for mais conveniente para ele.

Protocolo SMTP

- Exemplo:

```
S: 220 hamburger.edu
C: HELO crepes.fr
S: 250 Hello crepes.fr, pleased to meet you
C: MAIL FROM: <alice@crepes.fr>
S: 250 alice@crepes.fr ... Sender ok
C: RCPT TO: <bob@hamburger.edu>
S: 250 bob@hamburger.edu ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: Do you like ketchup?
C: How about pickles?
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 hamburger.edu closing connection
```

Protocolo SMTP

- SMTP usa conexões persistentes
- SMTP exige que as mensagens (cabeçalho e corpo) estejam em ASCII de 7 bits
- Servidor SMTP usa CRLF.CRLF para indicar o final da mensagem

Protocolo SMTP

- Comparação com HTTP
  - HTTP: pull; E-mail: push
  - Ambos usam comandos e respostas em ASCII, interação comando/resposta e códigos de status
  - HTTP: cada objeto encapsulado na sua própria mensagem de resposta
  - SMTP: múltiplos objetos são enviados numa mensagem multiparte

Protocolos de Acesso

- SMTP: entrega e armazena no servidor do destino
- Protocolo de acesso: recupera mensagens do servidor
  - POP: Post Office Protocol [RFC 1939]
    - Autorização (agente <-->servidor) e download
  - IMAP: Internet Mail Access Protocol [RFC 1730]
    - Maiores recursos (mais complexo)
    - Manipulação de mensagens armazenadas no servidor
  - HTTP: Hotmail , Yahoo! Mail, Gmail, etc.

Protocolo POP3

- O exemplo a seguir usa o modo *download-and-delete*
  - Bob não pode reler o e-mail se ele trocar o cliente
  - *download-and-keep*: cópias das mensagens em clientes diferentes
  - POP3 é stateless através das sessões

Protocolo POP3

- Fase de autorização
  - Comandos do cliente
    - user: declara nome do usuário
    - pass: password
  - respostas do servidor
    - +OK
    - -ERR

```
telnet mailServer 110
+OK POP3 server ready
user bob
+OK
pass hungry
+OK user successfully logged on
```

Protocolo POP3

- Fase de transação
  - list: lista mensagens e tamanhos
  - retr: recupera mensagem pelo número
  - dele: apaga
  - quit

```
C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: (blah blah ...
S: ..................
S: ..........blah)
S: .
C: dele 1
C: retr 2
S: (blah blah ...
S: ..................
S: ..........blah)
S: .
C: dele 2
C: quit
S: +OK POP3 server signing off
```

Protocolo IMAP

- Mantém todas as mensagens em um lugar: o servidor
- Permite que o usuário organize as mensagens em pastas
- Mantém o estado do usuário através das sessões:
- Nomes das pastas e mapeamentos entre os IDs da mensagem e o nome da pasta

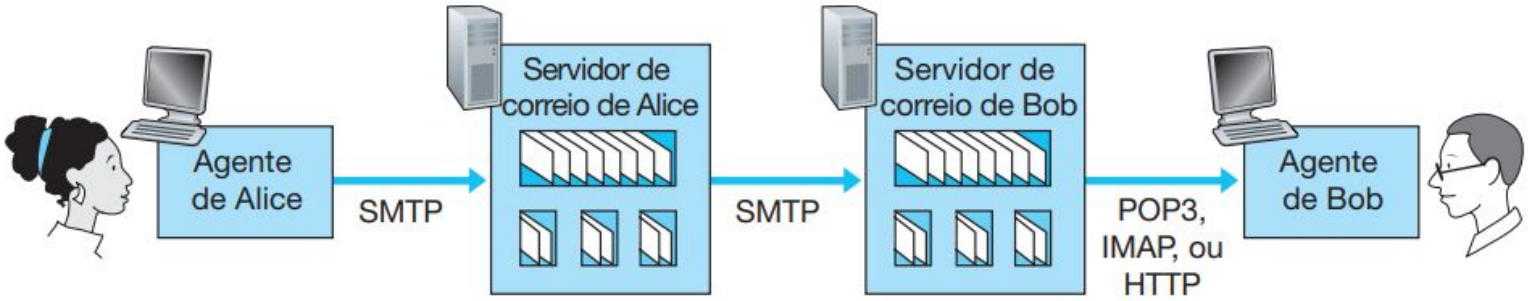
Comandos IMAP

Comando	Descrição
CAPABILITY	Lista capacidades do servidor
STARTTLS	Inicia o transporte seguro (TLS; ver Capítulo 8)
LOGIN	Login no servidor
AUTHENTICATE	Login com outro método
SELECT	Seleciona uma pasta
EXAMINE	Seleciona uma pasta apenas de leitura
CREATE	Cria uma pasta
DELETE	Exclui uma pasta
RENAME	Renomeia uma pasta
SUBSCRIBE	Acrescenta pasta no conjunto ativo
UNSUBSCRIBE	Remove pasta do conjunto ativo
LIST	Lista as pastas disponíveis
LSUB	Lista as pastas ativas
STATUS	Captura o status de uma pasta

Comandos IMAP

Comando	Descrição
STATUS	Captura o status de uma pasta
APPEND	Acrescenta uma mensagem a uma pasta
CHECK	Captura um ponto de verificação de uma pasta
FETCH	Captura mensagens de uma pasta
SEARCH	Localiza mensagens em uma pasta
STORE	Altera flags de mensagem
COPY	Faz uma cópia de uma mensagem em uma pasta
EXPUNGE	Remove mensagens marcadas para exclusão
UID	Emite comandos usando identificadores exclusivos
NOOP	Não faz nada
CLOSE	Remove mensagens marcadas e fecha pasta
LOGOUT	Efetua o logout e fecha a conexão

Protocolos de Acesso



Obrigado!

Dúvidas?