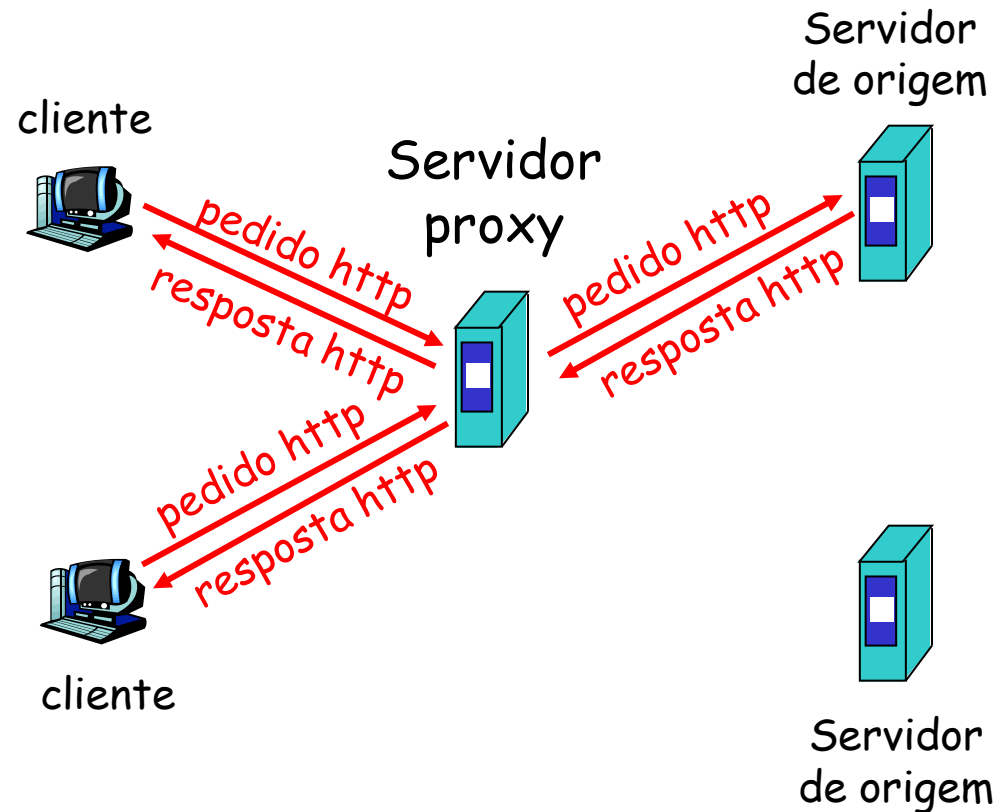


Aula 8

Cache Web (servidor proxy)

Meta: atender pedido do cliente sem envolver servidor de origem

- ❑ usuário configura browser: acessos Web via proxy
- ❑ cliente envia todos pedidos HTTP ao proxy
 - se objeto no cache do proxy, este o devolve imediatamente na resposta HTTP
 - senão, solicita objeto do servidor de origem, depois devolve resposta HTTP ao cliente



Mais sobre Caches Web

- ❑ Cache atua tanto como cliente quanto como servidor
 - Quando recebe requisições de um browser e lhe envia respostas, é um servidor.
 - Quando envia requisições para um servidor de origem e recebe respostas dele, é um cliente.
- ❑ Tipicamente o cache é instalado por um ISP (universidade, empresa, ISP residencial)

Mais sobre Caches Web

Para que fazer cache Web?

- ❑ Redução do tempo de resposta para os pedidos do cliente (se houver uma conexão de alta velocidade entre o cliente e o cache e se este tiver o objeto requisitado então poderá entregar rapidamente o objeto ao cliente)
- ❑ Redução do tráfego no canal de acesso de uma instituição à Internet (com isso a empresa não precisa ampliar sua largura de banda tão rapidamente o que diminui custos)
- ❑ Caches Web podem reduzir substancialmente o tráfego na Internet como um todo, melhorando assim, o desempenho para todas as aplicações.

Exemplo de cache (1)

A figura mostra duas redes

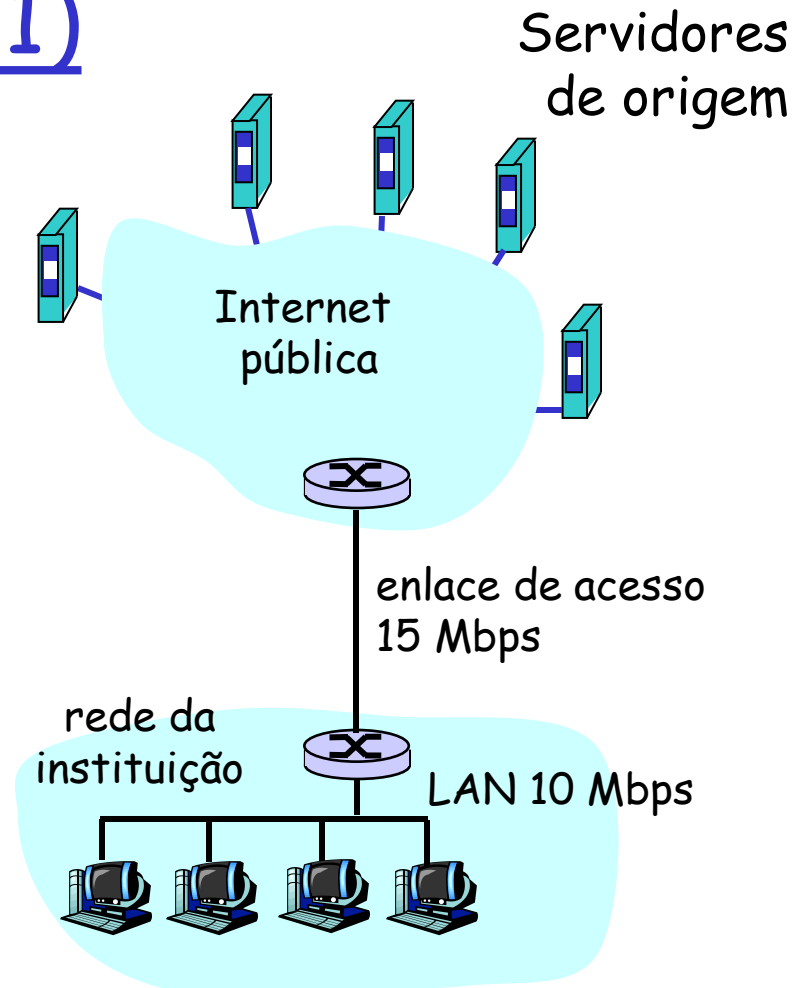
- Uma rede institucional
- Internet pública

Rede Institucional

- LAN de alta velocidade
- Um roteador da rede Institucional e um roteador da Internet estão ligados por um enlace de 15Mbps

Internet

- Os servidores de origem estão todos ligados à Internet (localizados pelo mundo todo)



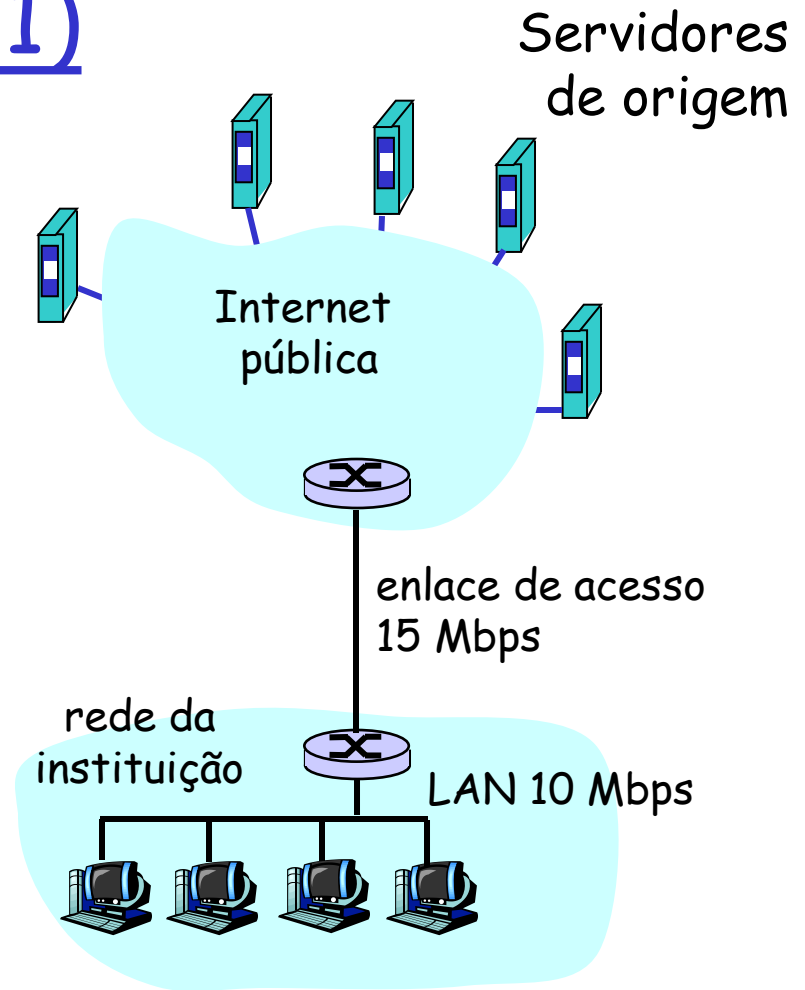
Exemplo de cache (1)

Hipóteses

- ❑ Tamanho médio de um objeto = 1 Mbits/s
- ❑ Taxa média de solicitações dos *browsers* de uma instituição para os servidores de origem = 15 requisições/seg
- ❑ Atraso do roteador institucional para qualquer servidor origem e de volta ao roteador = 2seg

Tempo de Resposta Total

- ❑ Tempo transcorrido entre a requisição de um objeto feita pelo browser e o recebimento dele = soma do atraso da LAN, do atraso entre os dois roteadores e do atraso da Internet.



Exemplo de cache (1)

Intensidade de tráfego na LAN

- $(15 \text{ requisições/segundo}) * (1 \text{ Mbits/requisição}) / (100 \text{ Mbps}) = 0,15$

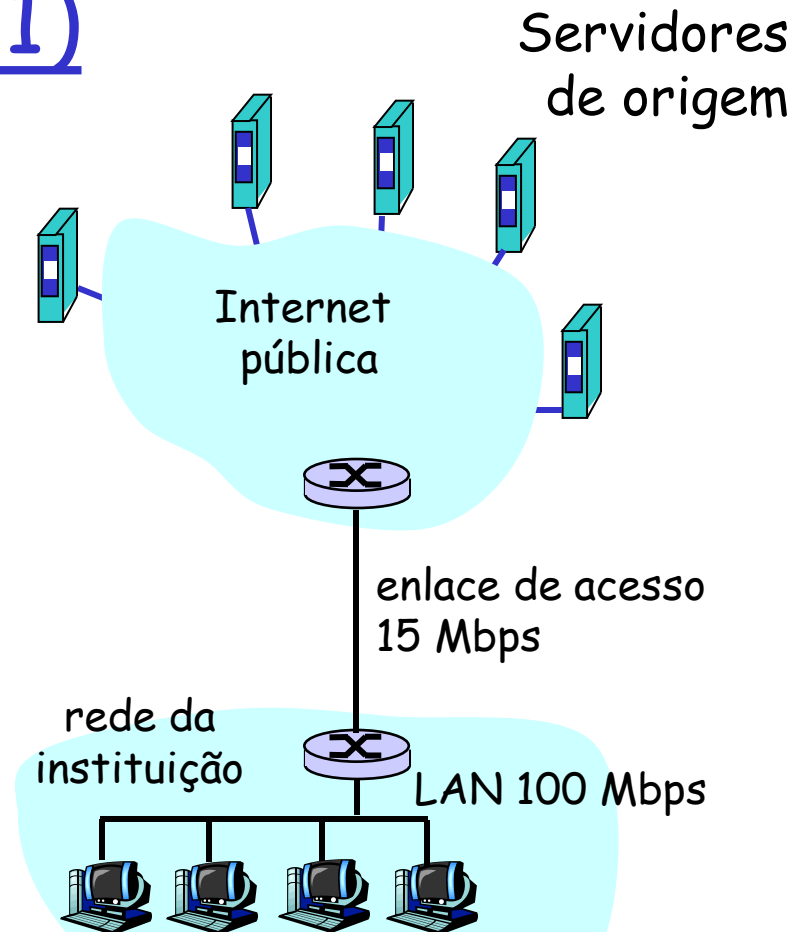
Intensidade de tráfego do roteador da Internet ao da Instituição

- $(15 \text{ requisições/segundo}) * (1 \text{ Mbits/requisição}) / (15 \text{ Mbps}) = 1$

Comparação

Uma intensidade de tráfego de 0,15 em uma LAN resulta em, no máximo, dezenas de milissegundos de atraso; consequentemente pode ser desprezado este atraso!

Contudo a medida que a intensidade de tráfego se aproxima de 1 (como é o segundo caso) o atraso em um enlace se torna muito grande e cresce sem limites.



Assim o tempo médio de resposta para atender requisições será da ordem de minutos, se não for maior, o que é inaceitável para os usuários!

Algo precisa ser feito!

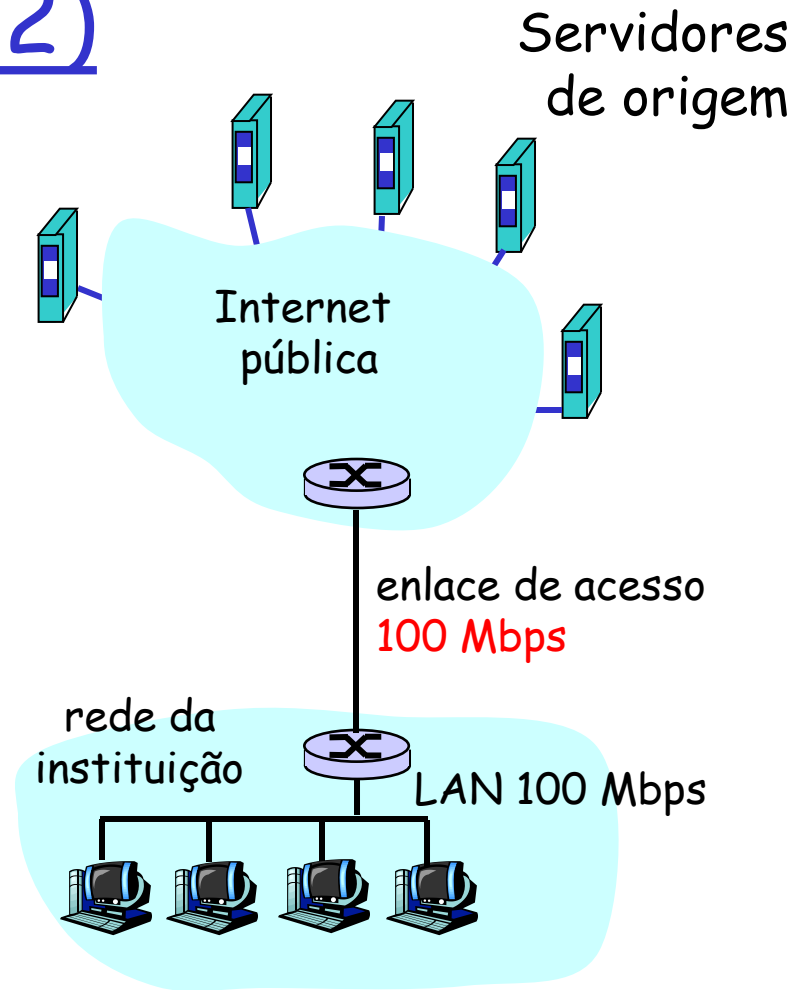
Exemplo de cache (2)

Solução em potencial

- Aumento da largura de banda do canal de acesso de 15Mbps para, por exemplo, 100 Mbps

Consequências

- O que pode ser muito dispendioso!!!!



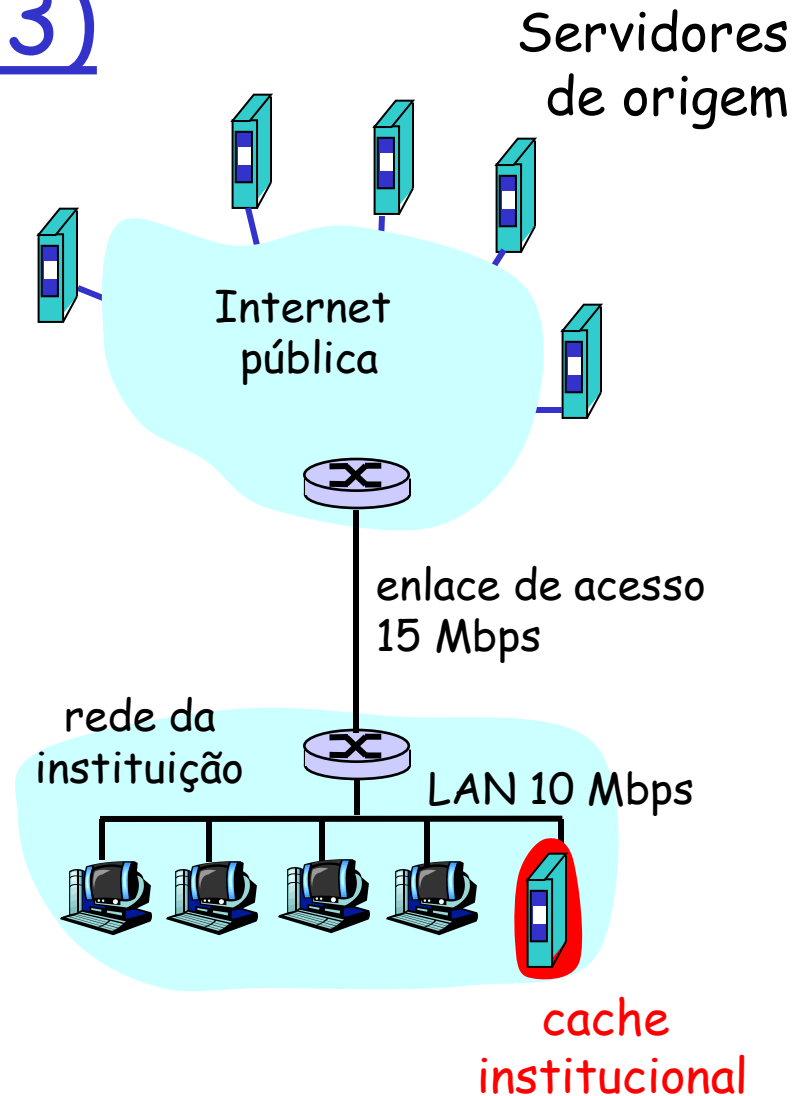
Exemplo de cache (3)

Instale uma cache

- Na prática, as requisições atendidas por um cache - variam de 0,2 a 0,7
- Suponha que a taxa seja de 0,4

Consequências

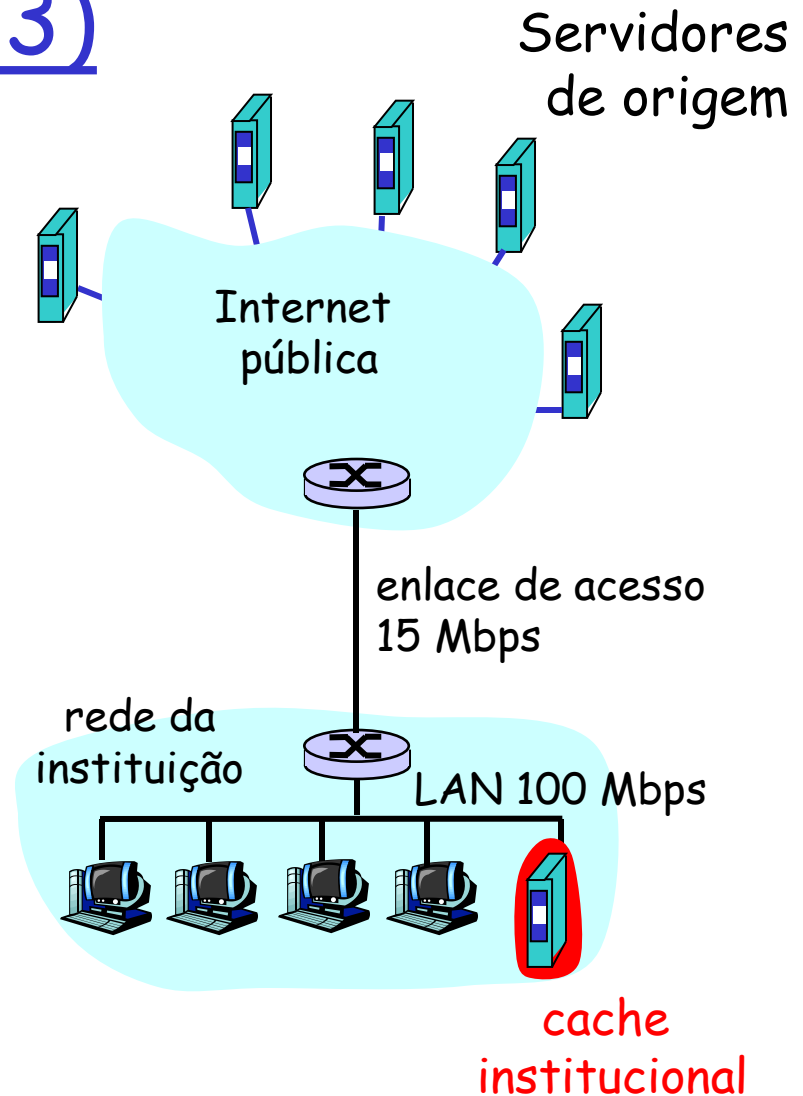
- Como os clientes e o cache estão conectados a mesma LAN de alta velocidade 40% dos pedidos serão atendidos quase que imediatamente (~ 10 milisegundos)
- 60% dos pedidos serão servidos pelos servidores de origem, resultando em atrasos desprezíveis (pois a intensidade do tráfego diminui de 1 para 0,6)
- Em geral, uma intensidade de tráfego menor do que 0,8 corresponde a um atraso pequeno comparado aos 2 segundos do atraso da Internet



Exemplo de cache (3)

Atraso médio é:

- ❑ $0,4 * (0,01 \text{ segundos}) + 0,6 * (2,01 \text{ segundos}) = \text{ligeiramente maior do que } 1,2 \text{ segundos}$
- ❑ Essa solução resulta em tempo de resposta até menor do que se conseguiria com a troca do enlace de acesso e não requer que a Instituição atualize seu enlace com a Internet
- ❑ Evidentemente, a Instituição terá que comprar e instalar um cache Web, mas esse custo é baixo - muitos caches usam softwares de domínio público que rodam em PCs baratos.



GET condicional

- Embora possa reduzir o tempo de resposta do ponto de vista do usuário, fazer cache introduz um novo problema - a cópia do objeto no cache pode estar desatualizada!
- O HTTP possui mecanismo que permite que um cache verifique se seus objetos estão atualizados (GET condicional)
- **Meta:** não enviar objeto se cliente já tem (no cache) versão atual

Uma mensagem de requisição HTTP é denominada uma mensagem GET condicional se:

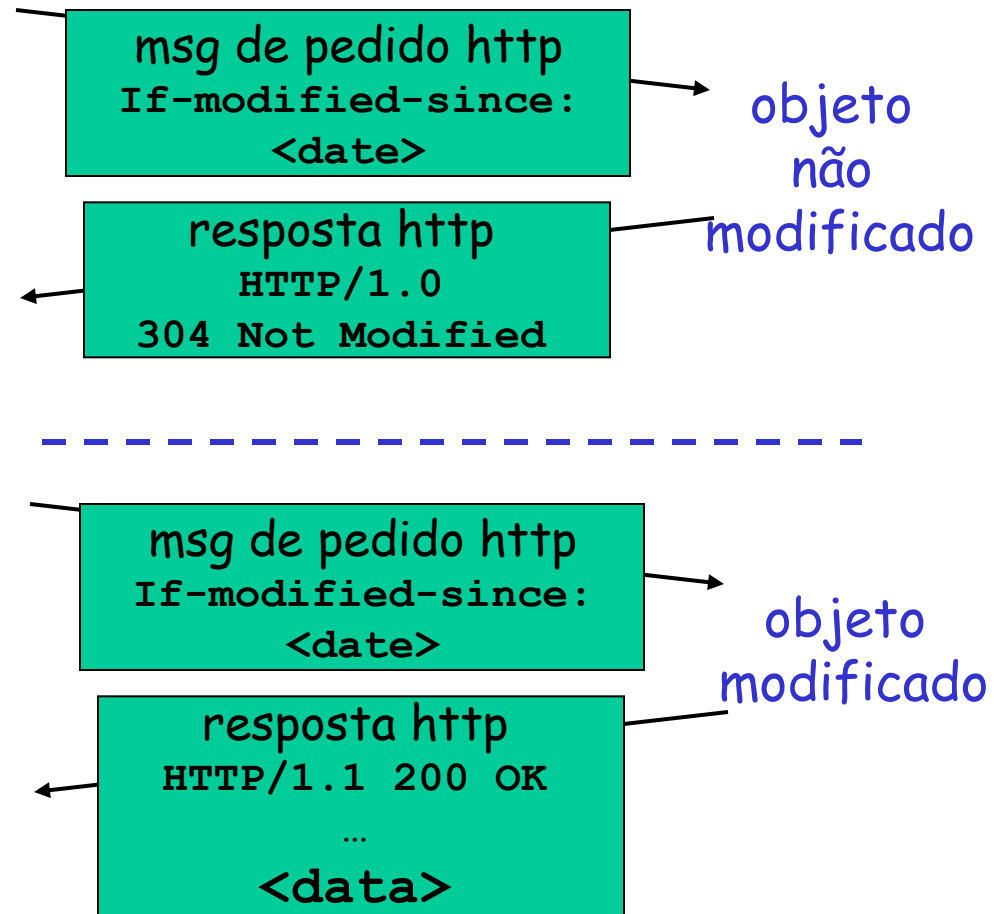
- (1) usar o método GET
- (2) Possuir uma linha de cabeçalho *If-modified-since*

GET condicional

- ❑ cache: especifica data da cópia no cache no pedido http
If-modified-since:
<date>
- ❑ servidor: resposta não contém objeto se cópia no cache é atual:
HTTP/1.0 304 Not Modified
- ❑ servidor: resposta contém objeto
HTTP/1.1 200 OK
- ❑ O cache: passa o objeto ao browser requisitante e guarda em sua memória cache local e também a data a última modificação.

cache

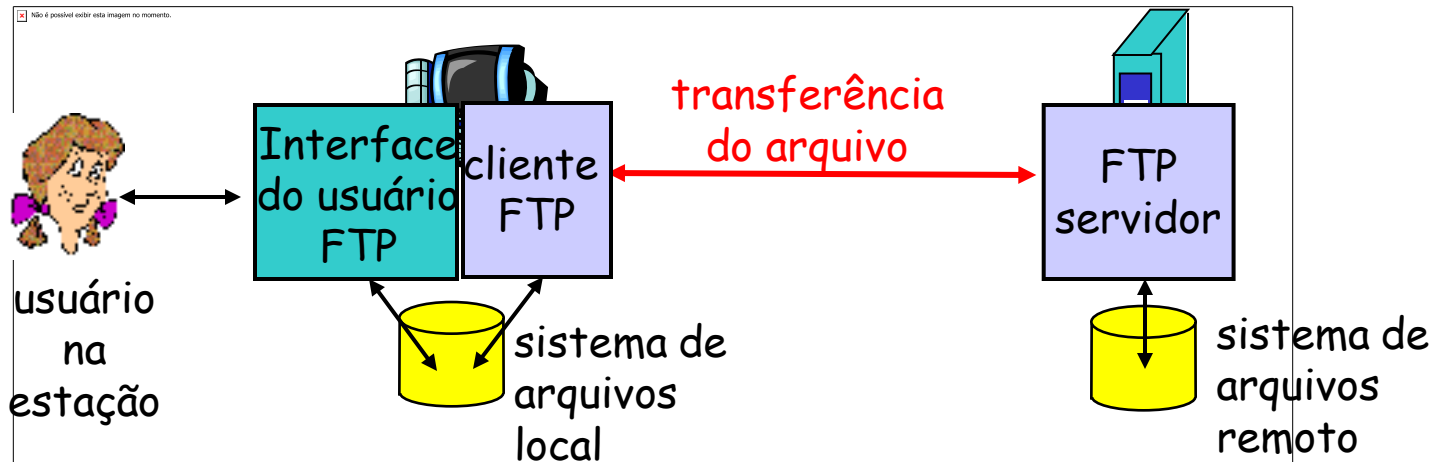
servidor



Capítulo 2: Roteiro

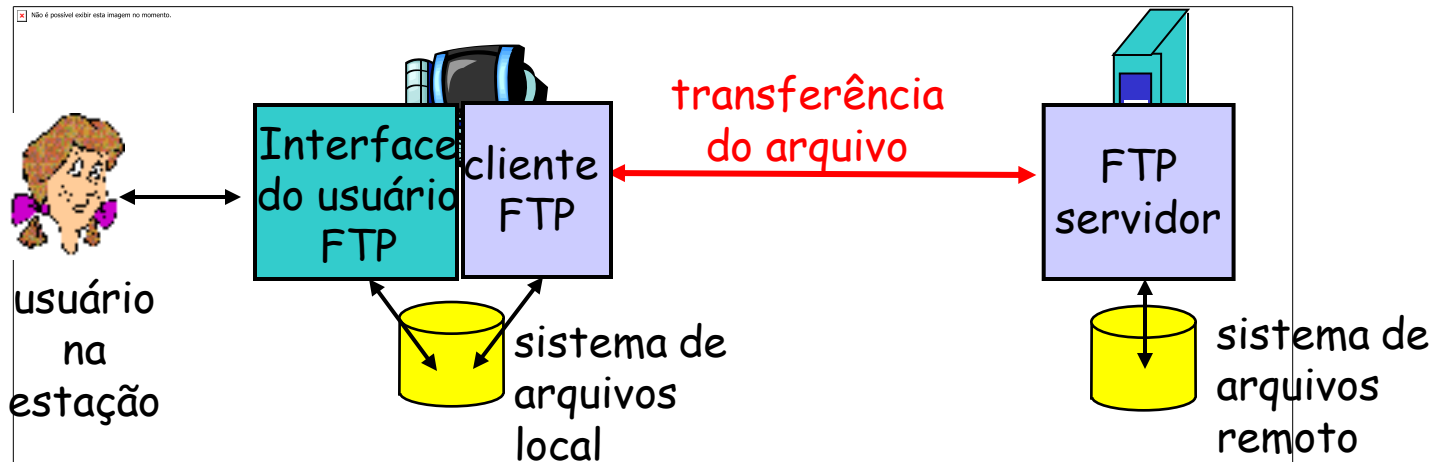
- ❑ 2.1 Princípios dos protocolos da camada de aplicação
- ❑ 2.2 Web e HTTP
- ❑ 2.3 FTP
- ❑ 2.4 Correio Eletrônico
 - SMTP, POP3, IMAP
- ❑ 2.5 DNS

FTP: o protocolo de transferência de arquivos



- ❑ transferir arquivo de/para hospedeiro remoto
- ❑ O usuário interage com o FTP por meio de um agente de usuário FTP
- ❑ modelo cliente/servidor
 - *cliente*: lado que inicia transferência (pode ser de ou para o sistema remoto)
 - *servidor*: hospedeiro remoto
- ❑ ftp: RFC 959
- ❑ servidor ftp: porta 21

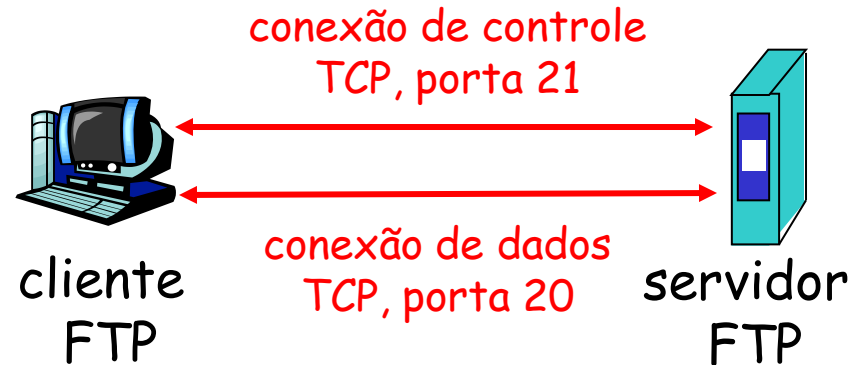
FTP: o protocolo de transferência de arquivos



- ❑ O usuário fornece o nome do hospedeiro remoto, o que faz com que o processo cliente FTP do hospedeiro local estabeleça uma conexão TCP com o processo servidor FTP do hospedeiro remoto.
- ❑ O usuário fornece sua identificação e senha que são enviadas pela conexão TCP como parte dos comandos FTP.
- ❑ Assim que autorizado pelo servidor, o usuário copia um ou + arquivos armazenados no sistema de arquivo local para o sistema de arquivo remoto (ou vice-versa).

HTTP versus FTP

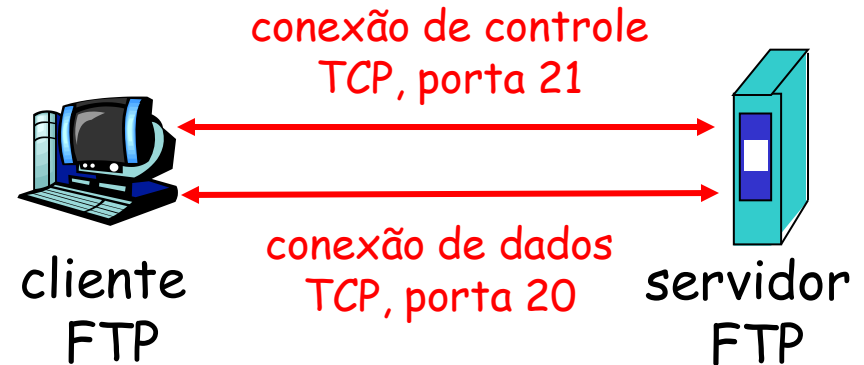
- ❑ Ambos são protocolos de transferência de arquivos;
- ❑ Ambos utilizam o TCP;
- ❑ **DIFERENÇAS:**
- ❑ FTP usa duas conexões TCP paralelas para transferir um arquivo: uma conexão de controle e outra de dados;
- ❑ A primeira é usada para enviar informações de controle entre os dois hospedeiros
- ❑ A segunda é usada para enviar um arquivo;
- ❑ Como o FTP usa uma conexão de controle separada dizemos que ele envia informações de controle **fora de banda**.



- ❑ O HTTP envia linhas de cabeçalho de requisição e de resposta pela mesma conexão TCP que carrega o próprio arquivo transferido;
- ❑ O HTTP envia suas informações de controle **na banda**.

FTP: conexões separadas p/ controle, dados

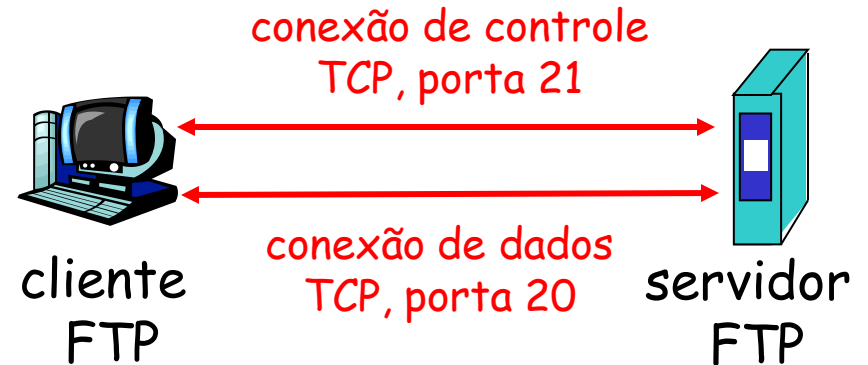
- ❑ cliente FTP contata servidor FTP na porta 21, especificando o TCP como protocolo de transporte
- ❑ O cliente obtém autorização através da conexão de controle
- ❑ O cliente consulta o diretório remoto enviando comandos através da conexão de controle
- ❑ Quando o servidor recebe um comando para a transferência de um arquivo, ele abre uma conexão de dados TCP para o cliente
- ❑ Após a transmissão de um arquivo o servidor fecha a conexão



- ❑ O servidor abre uma segunda conexão TCP para transferir outro arquivo
- ❑ Conexão de controle: permanece aberta durante toda a sessão do usuário, mas uma nova conexão de dados é criada para cada arquivo transferido dentro de uma sessão. (conexão de dados não persistente)

FTP: conexões separadas p/ controle, dados

- Durante uma sessão, o servidor FTP deve manter informações de **estado** sobre o usuário.
- O HTTP, por outro lado, é sem estado - não tem de monitorar o estado de nenhum usuário.



Exercícios

1. Por que a utilização de um cache Web na Lan é mais vantagem do que simplesmente aumentar a velocidade de acesso?
2. Quais as diferenças importantes entre o HTTP e o FTP?