

# Ch.02 – Camada de Aplicação

## 2.1 – Princípios de Aplicações de Rede

### 2.1.1 – Arquitetura de Aplicações de Rede

### 2.1.2 – Comunicação entre Processos

### 2.1.3 – Serviços de Transporte Disponíveis

### 2.1.4 – Serviços de Transporte na Internet

### 2.1.5 – Protocolos da Camada de Aplicação

### 2.1.6 – Aplicações de Rede Usuais

## 2.2 – World Wide Web e o HTTP

### 2.2.1 – Descrição Geral do HTTP

### 2.2.2 – Conexões Persistentes e Não Persistentes

### 2.2.3 – Formato de Mensagens HTTP .. ..

## ... Ch.02 – Camada de Aplicação

2.2 – World Wide Web e o HTTP .. continuação ..

2.2.4 – Interação Usuário/Servidor (Cookies)

2.2.5 – Cache na World Wide Web

2.2.6 – GET Condicional

2.3 – File Transpor Protocol (FTP)

2.3.1 – Camadas e Respostas FTP

2.4 – Correio Eletrônico – eMail (continuação)

2.4.1 - Sample Mail Transport Protocol (SMTP)

2.4.2 – Comparação com HTTP

2.4.3 – Formatos da Mensagem de eMail

2.4.4 – Protocolo de Acesso ao eMail

## ... Ch.02 – Camada de Aplicação

### 2.5 – Domain Name System (DNS)

#### 2.5.1 – Serviços do DNS

#### 2.5.2 – Visão Geral do DNS

#### 2.5.3 – Registros e Mensagens do DNS

### 2.6 – Aplicações P2P

#### 2.6.1 – Distribuição de Arquivos P2P

#### 2.6.2 – Distributed Hash Tables (DHTs)

### 2.7 – Programação de Sockets

#### 2.7.1 – Programação de Sockets UDP

#### 2.7.2 – Programação de Sockets TCP

# Referências Bibliográficas

- James F. Kurose; Keith W. Ross – Redes de Computadores e a Internet: Uma Abordagem Top-Down – Pearson São Paulo; 6ª Ed.; 2013; ISBN-13 : 978-8581436777
- ... Lectures dos autores James F. Kurose; Keith W. Ross  
<http://www.pearsonhighherd.com/kurose-ross>
- WireShark Labs – V7.0 – CSCI 363 Computer Networks  
[https://gaia.cs.umass.edu/kurose\\_ross/wireshark.htm](https://gaia.cs.umass.edu/kurose_ross/wireshark.htm)

# Referências Bibliográficas

- James F. Kurose; Keith W. Ross – Redes de Computadores e a Internet: Uma Abordagem Top-Down – Pearson São Paulo; 6ª Ed.; 2013; ISBN-13 : 978-8581436777
- ... Lectures dos autores James F. Kurose; Keith W. Ross  
<http://www.pearsonhighherd.com/kurose-ross>
- WireShark Labs – V7.0 – CSCI 363 Computer Networks  
[https://gaia.cs.umass.edu/kurose\\_ross/wireshark.htm](https://gaia.cs.umass.edu/kurose_ross/wireshark.htm)

## 2 – Camada de Aplicação

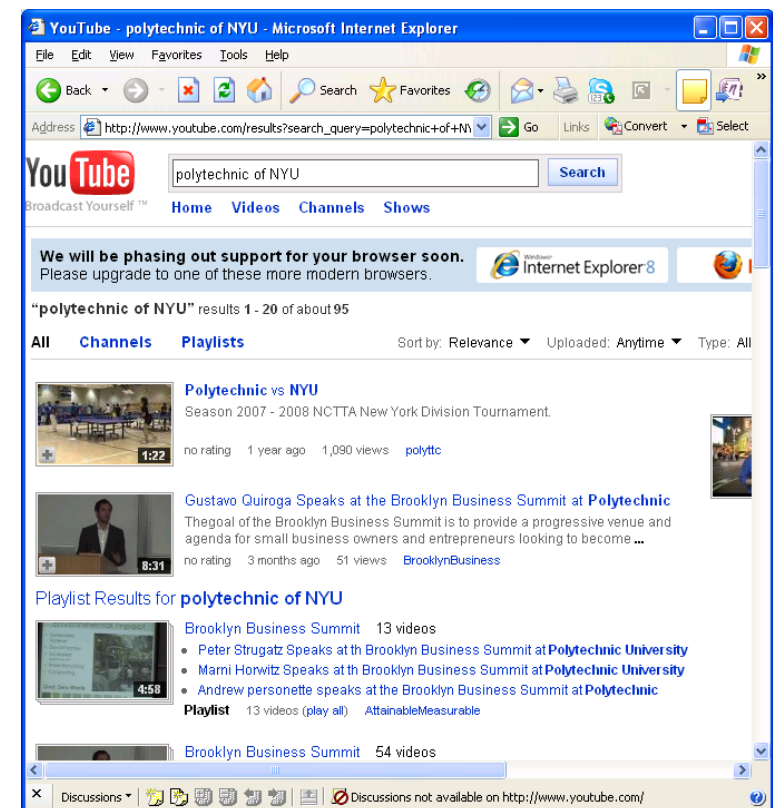
# 2. Camada de Aplicação

- “**objetivos**” .. descrever os aspectos conceituais e de implementação de protocolos de aplicação de redes de computadores.
- .. paradigma cliente/servidor e paradigma peer-to-peer.
- .. modelos de serviço de transporte » camada de aplicação.
  
- “**perscepção de protocolos**” .. examinando-os alguns deles:
- HyperText Transport Protocol (HTTP).
- File Transport Protocol (FTP).
- Sample Mail Transport Protocol (SMTP) e POP3/IMAP.
- Post Office Protocol (POP3) / Internet Mail Access Protocol (IMAP).
- Domain Name Service (DNS).

## 2 – Camada de Aplicação

# 2. Camada de Aplicação

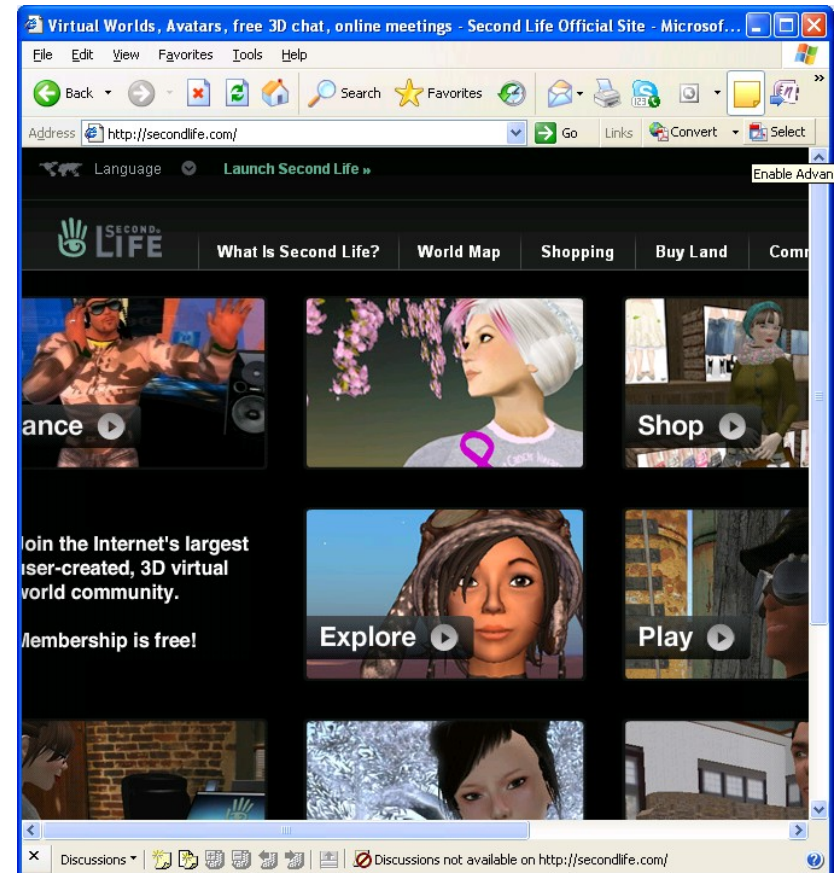
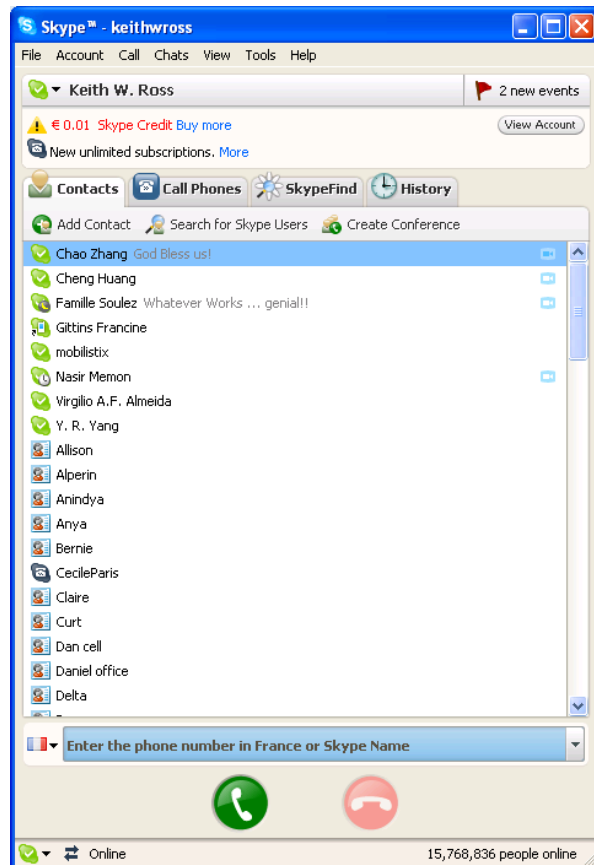
- Aplicações Clássicas (populares nos anos 70 e 80) .. correio eletrônico, acesso remoto, transferência de arquivo e grupos de discussão.
- Aplicações que fizeram Sucesso (década de 90) .. World Wide Web, abrangendo a navegação na Web, busca e o comércio eletrônico.



## 2 – Camada de Aplicação

# 2. Camada de Aplicação

- Década de 2000 .. explosão de aplicações de voz e vídeo, p.ex., Voz sobre IP (VoIP), videoconferência sobre IP, como Skype; vídeo sob-demanda, como YouTube; e vídeo sob-demanda (Netflix).





## 2 – Camada de Aplicação / 2.1 Princípios de Aplicações de Rede

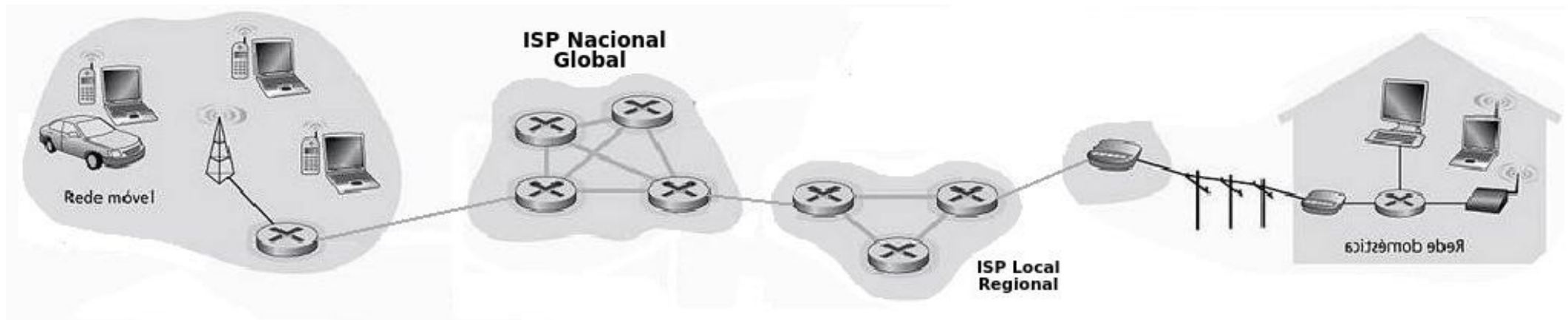
### 2.1 Princípios de Aplicações de Rede

- Algumas Aplicações de Rede:
- Correio Eletrônico .. Servidor de eMail.
- Conteúdo de Páginas Web .. World Wide Web.
- Troca de Mensagens .. mensagem instantânea.
- Acesso Remoto .. login remoto.
- Compartilhamento de Arquivos .. P2P.
- Jogos Multiusuários .. plataforma de jogos em rede.
- Vídeo sob Demanda .. clipes de vídeo em fluxo contínuo.
- Mix de Conteúdos .. redes sociais.
- Ligação de Telefone sobre IP .. (VoIP) Voice over IP.
- Conferência Web .. vídeoconferência em tempo real.
- Computação de Alto Desempenho .. computação em grade.

## 2 – Camada de Aplicação / 2.1 Princípios de Aplicações de Rede

### 2.1.1 – Arquitetura de Aplicação de Rede

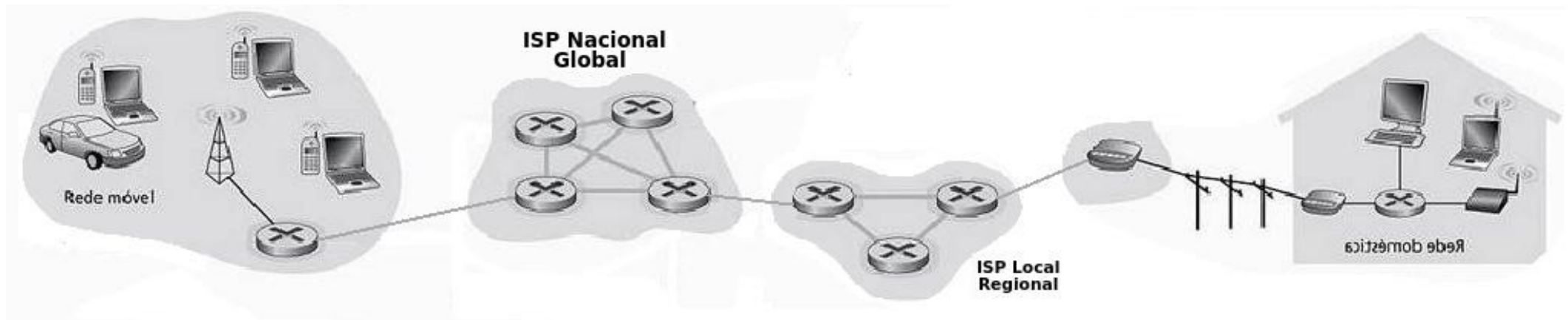
- “**arquitetura da aplicação**” .. determina como a aplicação é organizada nos vários sistemas finais ou “hosts” / “end systems”
- .. executam em (diferentes) “end systems” e se comunicam pela rede, p.ex., navegador Web se comunica com servidor Web.



## 2 – Camada de Aplicação / 2.1 Princípios de Aplicações de Rede

### ... 2.1.1 – Arquitetura de Aplicação de Rede

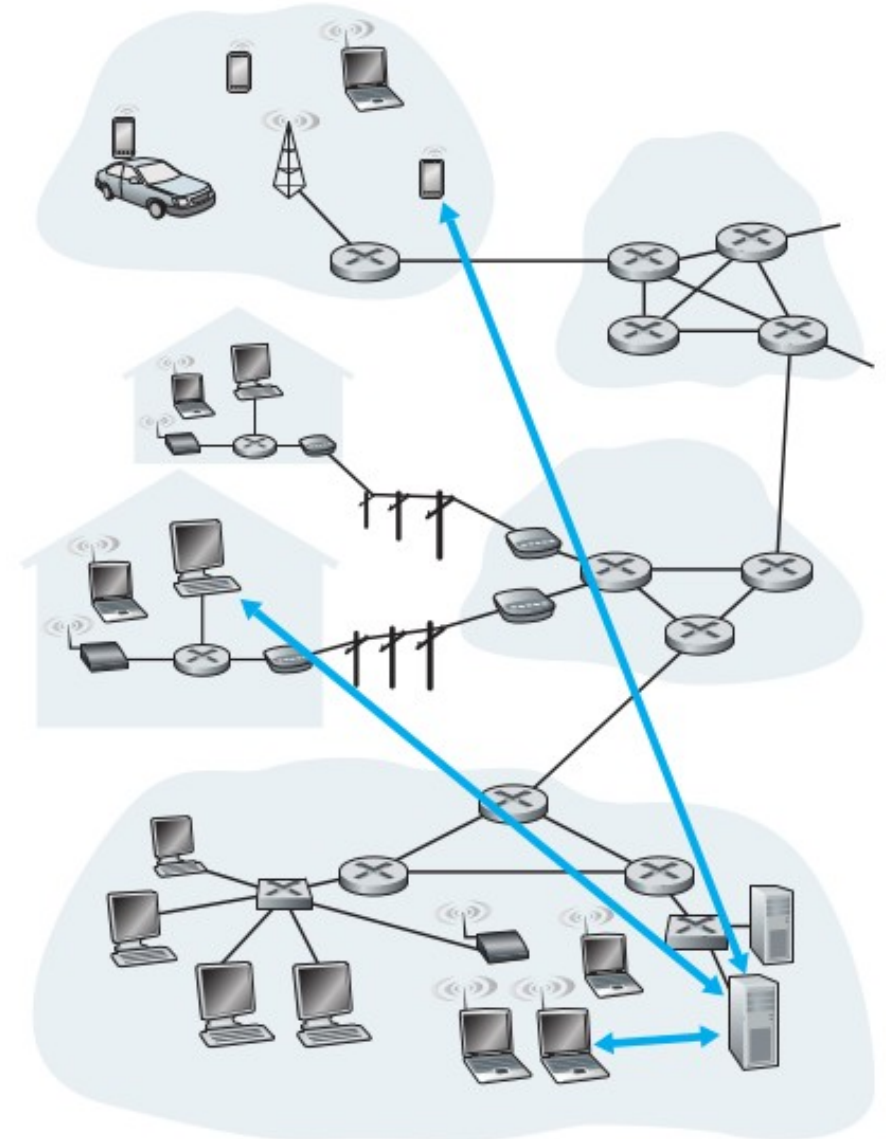
- .. dispositivos do núcleo da rede não executam aplicações do usuário, por outro lado, as aplicações nos “hosts” ou “end systems” permitem rápido desenvolvimento de novas aplicações bem como seu uso.
- “**observação**” .. “não há necessidade de se escrever software de aplicação para dispositivos do núcleo da rede, ou seja, roteadores”.



## 2 – Camada de Aplicação / 2.1 Princípios de Aplicações de Rede

### ... 2.1.1 – Arquitetura de Aplicação de Rede

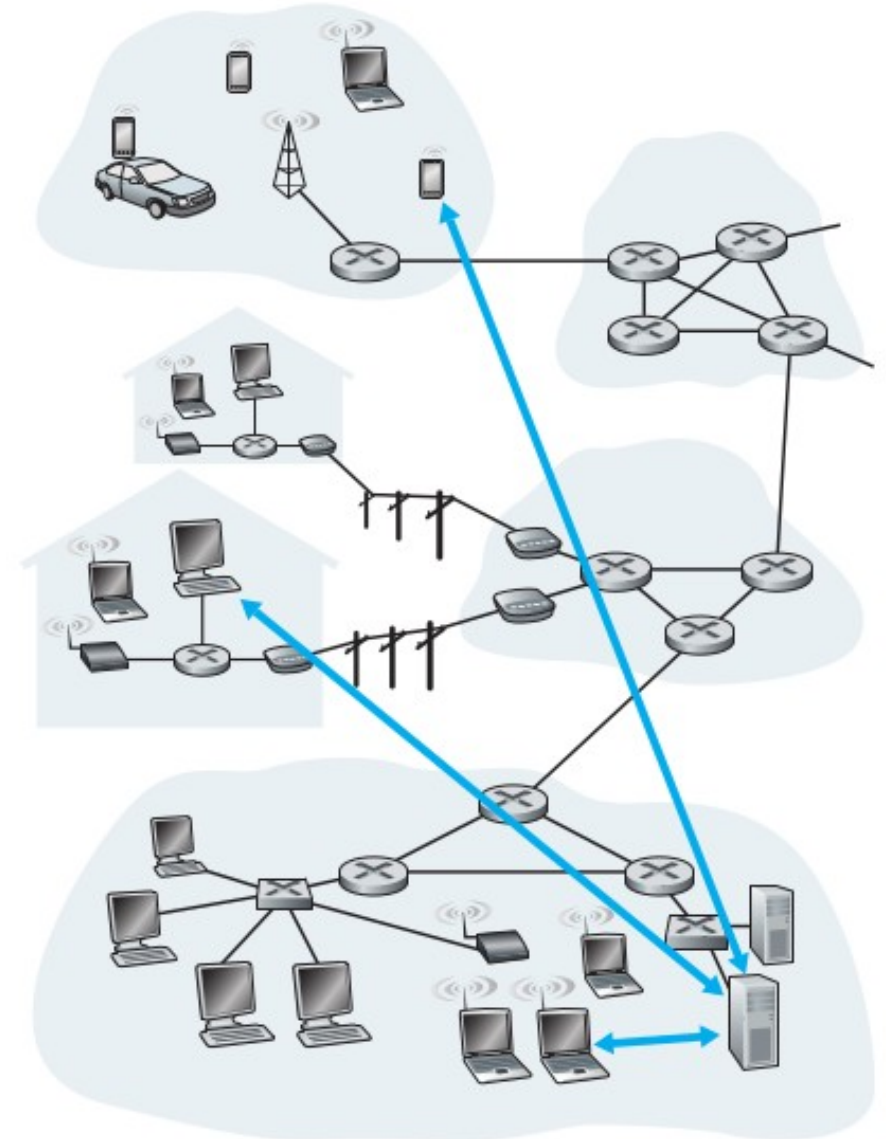
- Arquitetura Cliente/Servidor .. há um “host” executando 100% do tempo denominado servidor, que atende a requisições de muitos outros “hosts”, denominados clientes.
- “**características**” do Servidor ..
- .. “host” sempre ligado;
- .. endereço IP permanente;
- .. “server farms” para expansão.



## 2 – Camada de Aplicação / 2.1 Princípios de Aplicações de Rede

### ... 2.1.1 – Arquitetura de Aplicação de Rede

- Arquitetura Cliente/Servidor .. há um “host” executando 100% do tempo denominado servidor, que atende a requisições de muitos outros “hosts”, denominados clientes.
- “**características**” do Cliente ..
- .. comunicam-se com o servidor;
- .. conexões intermitentes;
- .. podem ter endereços IP dinâmicos;
- .. não se comunicam entre si (direto).



## 2 – Camada de Aplicação / 2.1 Princípios de Aplicações de Rede

### ... 2.1.1 – Arquitetura de Aplicação de Rede

- Centros de Dados do Google .. custo estimado de US\$ 600 Mi
- .. gastos de US\$ 2.4 Bi em 2007 em novos centros de dados.
- .. cada centro de dados consome de 50 a 100 MegaWatts.

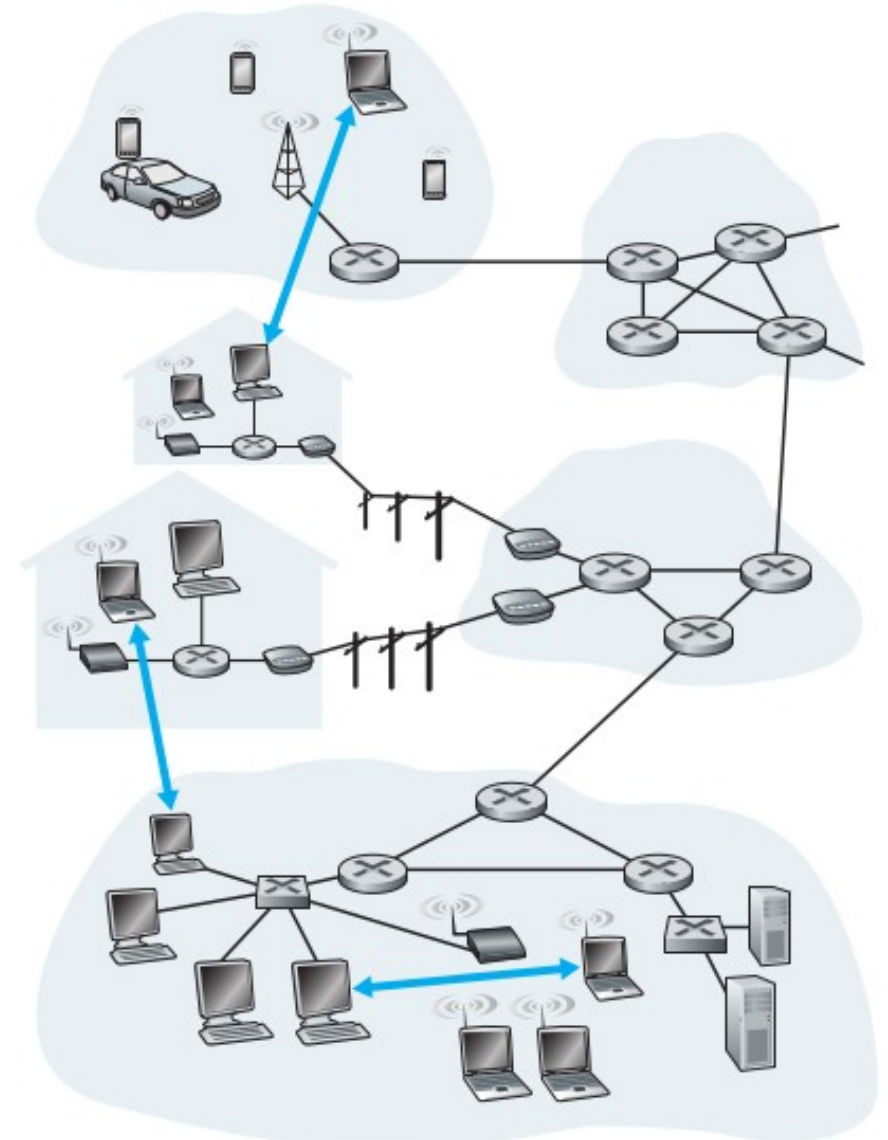




## 2 – Camada de Aplicação / 2.1 Princípios de Aplicações de Rede

### ... 2.1.1 – Arquitetura de Aplicação de Rede

- Arquitetura P2P .. aplicação utiliza a comunicação direta entre duplas de hospedeiros conectados alternadamente, denominados pares.
- .. “nenhum” servidor sempre ligado.
- .. “hosts” ou sistemas finais arbitrários se comunicam diretamente.
- .. pares se conectam de forma intermitente e mudam endereços IP.
- .. característica mais forte do P2P é sua autoescalabilidade.



## 2 – Camada de Aplicação / 2.1 Princípios de Aplicações de Rede

### ... 2.1.1 – Arquitetura de Aplicação de Rede

- **“desafios para novas aplicações”** .. P2P
- ISPs oferecem MAIOR Largura de Banda .. ISPs residenciais, p.ex., DSLs e ISPs a Cabo, oferecem largura de banda “assimétrica”, ou seja, baixa largura de banda para “upload”.
- **“segurança”** .. em razão de sua natureza altamente distribuída, há a necessidade de contemplar aspectos de segurança.
- **“recursos dos usuários”** .. usuários participam efetivamente no oferecimento de largura de banda, armazenamento e recursos da computação para que possam tirar maior proveito das aplicações.



## 2 – Camada de Aplicação / 2.1 Princípios de Aplicações de Rede

### ... 2.1.1 – Arquitetura de Aplicação de Rede

- Aplicação Híbrida - Skype:
- .. aplicação P2P .. Voice-over-IP P2P
- .. servidor centralizado .. encontra endereço da parte remota.
- .. conexão cliente-cliente .. direta (não através de servidor).
  
- Aplicação Híbrida de Mensagem Instantânea:
- .. bate-papo entre dois usuários é ponto-a-ponto.
- .. serviço centralizado p/ detecção/localização da presença do cliente.
- .. usuário registra seu endereço IP com servidor quando está on-line.
- .. usuário contacta servidor central para descobrir parceiros ou IPs.

## 2 – Camada de Aplicação / 2.1 Princípios de Aplicações de Rede

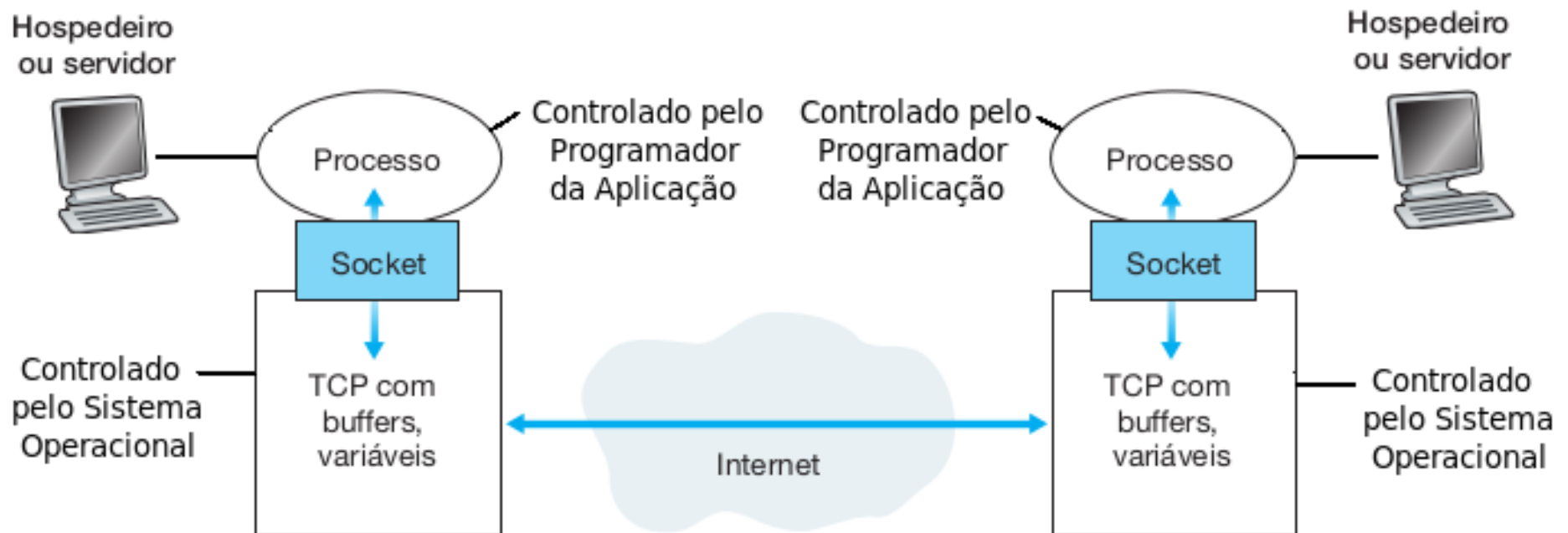
### 2.1.2 – Comunicação entre Processos

- “**contexto**” ... quando processos executam no mesmo “host”, os mesmos usam a comunicação interprocessos, cujas regras são determinadas pelo sistema operacional do “sistema final”.
- “**mundaça**” .. processos em dois sistemas finais distintos se comunicam trocando mensagens por meio da rede de computadores.
- ... ou seja, processos se comunicam usando a camada de aplicação da pilha de comunicação com 05 camadas da Arquitetura TCP/IP.
- “**processo cliente**” .. processo que inicia a comunicação.
- “**processo servidor**” .. processo que espera para ser contactado.
- Aplicações P2P .. processos são clientes e servidores simultaneamente e, não há a divisão de funções como no Cliente/Servidor.

## 2 – Camada de Aplicação / 2.1 Princípios de Aplicações de Rede

### ... 2.1.2 – Comunicação entre Processos

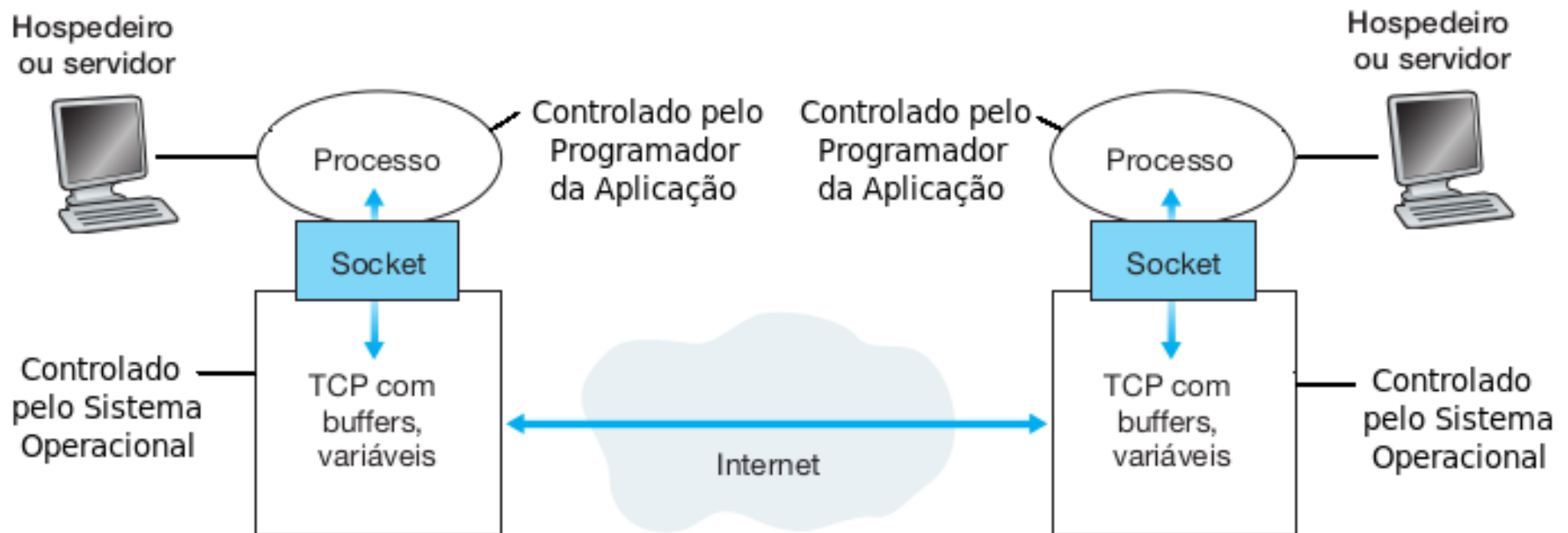
- .. processo envia mensagens para a rede e recebe mensagens da rede através de uma interface de software denominada “**socket**”.
- “**analogia**” .. socket assemelha-se a uma “porta”
- .. processo envia ou empurra mensagens pela porta origem (socket);
- .. processo receptor recebe mensagens na porta destino (socket).



## 2 – Camada de Aplicação / 2.1 Princípios de Aplicações de Rede

### ... 2.1.2 – Comunicação entre Processos

- Application Programming Interface (API) .. interface de programação entre aplicação e a rede pela qual as aplicações são criadas.
  - 1) .. escolha do protocolo de transporte, p.ex., TCP ou UDP.
  - 2) .. capacidade de contemplar requisitos funcionais adicionais com o intuito de atender os requisitos exigidos pela aplicação.



## 2 – Camada de Aplicação / 2.1 Princípios de Aplicações de Rede

### ... 2.1.2 – Comunicação entre Processos

- “**contexto**” .. para que um processo em um “host” envie pacotes a um processo em um outro “host”, o receptor precisa ter um endereço.
  - 1) necessidade de obter o endereço do “host” no qual executa o “receptor” (dispositivo “host” tem endereço IP exclusivo de 32 bits);
  - 2) processo “receptor” deve ser “**identificado**” de modo que possa receber mensagens e, não somente, o processo “remetente”.
- e.g., use o comando “**ipconfig**” (SO Windows) no “**command prompt**” para obter seu endereço IP no seu “desktop” ou “notebook”.

## 2 – Camada de Aplicação / 2.1 Princípios de Aplicações de Rede

### ... 2.1.2 – Comunicação entre Processos

- “**dúvida**” .. é suficiente apenas o “endereço IP” do “host” em que o processo “receptor” é executado para identificar o “processo” ?
- .. não, pois muitos processos podem se encontrar no mesmo “host”.
- Identificador do “Host” inclui endereço IP do “host”.
- Nro. de Porta identifica o “processo receptor” no “host”.
- e.g., Nro. de Porta 80 para o Servidor HTTP
- e.g., Nro. de Porta 25 para Servidor de eMail

## 2 – Camada de Aplicação / 2.1 Princípios de Aplicações de Rede

### 2.1.3 – Serviço de Transporte para Aplicações

- “**contexto**” .. protocolo da camada de transporte tem a responsabilidade de levar as msgs. pela rede até o socket do destinatário.
- “**tipos de mensagens**” trocadas .. “request” e “replay”.
- “**sintaxe da mensagem**” .. campos nas mensagens; tamanhos dos campos campos para endereços; campos para detecção de erros; etc.
- “**semântica da mensagem**” .. significado dos vários campos que compõem as diferentes mensagens trocadas entre os pares.
- “**regras procedimentais**” .. quando e como processos enviam e respondem as msgs. que governam a troca de informações entre os elementos pares (hardware, software ou combinação).
- “**protocolo**” .. conjunto de regras e procedimentos que governam a troca de informações entre elementos pares.

## 2 – Camada de Aplicação / 2.1 Princípios de Aplicações de Rede

### ... 2.1.3 – Serviço de Transporte para Aplicações

- **“domínio público”** .. protocolos cuja especificação estão disponíveis, p.ex., RFCs ou “Reference For Comments”
- .. provê interoperabilidade .. p.ex., HTTP, SMTP, BitTorrent
- **“protocolos proprietários”** .. especificação fechada e não disponível, p.ex., Skype, PPStream
- **“serviços oferecidos pela camada de transporte”** .. quais são os serviços que protocolos da camada de transporte podem oferecer ?
- ... podemos classificar os possíveis serviços segundo 04 dimensões, ou seja, “transferência confiável de dados”; “vazão”; “temporização na entrega de mensagens” e “segurança”.



## 2 – Camada de Aplicação / 2.1 Princípios de Aplicações de Rede

### ... 2.1.3 – Serviço de Transporte para Aplicações

- ... podemos classificar os possíveis serviços segundo 04 dimensões: transferência confiável de dados, vazão, temporização e segurança.
- “**transferência de dados**” .. algumas aplicações (p.ex., áudio) podem tolerar alguma perda, enquanto que outras aplicações. (p.ex., transferência de arquivos) exigem transf. de dados 100% confiável.
- “**temporização**” .. algumas aplicações (p.ex., telefonia na Internet jogos interativos) exigem pouco atraso para serem “eficazes”.
- “**vazão**” .. algumas aplicações (p.ex., multimídia) exigem um mínimo de vazão para serem “eficazes”, enquanto outras aplicações (“aplicações elásticas”) utilizam qualquer vazão que receberem.
- “**segurança**” .. criptografia, integridade de dados, etc.

## 2 – Camada de Aplicação / 2.1 Princípios de Aplicações de Rede

### 2.1.4 – Serviço de Transporte do TCP/IP

- Rede Internet ou Arquitetura TCP/IP disponibiliza 02 protocolos de transporte para aplicações, o UDP e o TCP.
- ... cada um deles oferece um conjunto diferente de serviços para as aplicações solicitantes, no entanto, cada aplicação contempla os seus próprios requisitos permitindo que se faça a escolha apropriada.

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 – 1 Mbit/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

## 2 – Camada de Aplicação / 2.1 Princípios de Aplicações de Rede

### ... 2.1.4 – Serviço de Transporte do TCP/IP

- **“Protocolo TCP”** .. inclui um serviço orientado a conexão e um serviço confiável de transferência de dados (garantia de entrega) voltado exclusivamente para as aplicações que o utilizam:
- **“serviço orientado a conexão”** .. cliente e o servidor trocam informações de controle de camada de transporte antes que as mensagens de camada de aplicação comecem a fluir;
- **“serviço orientado a conexão”** .. conexão é full-duplex, visto que os dois processos podem trocar mensagens ao mesmo tempo.
- **“serviço confiável de transporte”** .. processos comunicantes contam com a entrega de todos os dados enviados sem erro e em ordem.
- **“controle de congestionamento”** .. serviço voltado ao bem-estar geral da Internet e não ao benefício direto dos processos comunicantes.

## 2 – Camada de Aplicação / 2.1 Princípios de Aplicações de Rede

### ... 2.1.4 – Serviço de Transporte do TCP/IP

- “**Protocolo UDP**” .. protocolo de transporte simplificado, leve, com um modelo de serviço minimalista (serviço não orientado para conexão).
- .. não estabelece conexão e nem oferece confiabilidade.
- .. não contempla controle de fluxo.
- .. não contempla controle de congestionamento.
- .. não contempla garantia de vazão ou segurança.
  
- “**serviço oferecido**” .. transferência de dados não confiável entre processo remetente e processo receptor.
  
- “**dúvida**” .. por que se incomodar ? Por que existe um UDP ?
- “**resposta**” .. há aplicações em que a entrega de dados no menor tempo é mais crítico que a eventual perda de alguns poucos dados.

## 2 – Camada de Aplicação / 2.1 Princípios de Aplicações de Rede

### ... 2.1.4 – Serviço de Transporte do TCP/IP

- **“aplicação de protocolos de transporte”** .. protocolos de transporte usados por algumas aplicações populares da Internet.
- p.ex., correio eletrônico, acesso remoto, Web e transferência de arquivos usam um serviço confiável de transferência de dados, garantindo que os dados cheguem a seu destino.

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

## 2 – Camada de Aplicação / 2.1 Princípios de Aplicações de Rede

### 2.1.5 – Protocolos de Camada de Aplicação

- **“protocolo de camada de aplicação”** .. define como se dá a troca de mensagens de aplicação em sistemas finais distintos.
- **“definição”** .. conjunto de regras e procedimentos que governam a troca de informação entre elementos pares, ou seja, ...
- **“tipos de mensagens”** trocadas .. “request” e “replay”.
- **“sintaxe da mensagem”** .. campos nas mensagens; tamanhos dos campos campos para endereços; campos para detecção de erros; etc.
- **“semântica da mensagem”** .. significado da informação nos campos.
- **“regras procedimentais”** .. quando e como processos enviam e respondem as msgs. que governam a troca de informações entre os pares.

## 2 – Camada de Aplicação / 2.1 Princípios de Aplicações de Rede

### 2.1.6 – Aplicações abordadas no Livro

- **“aplicações discutidas”** .. “ênfase” em um pequeno nro. de aplicações populares e muito utilizadas pelo público em geral.
- **“HyperText Transport Protocol - HTTP”** .. World Wide Web ou Web
- **“TELNET”** .. Transferência de Arquivos
- **“Simple Mail Transport Protocol - SMTP”** .. Correio Eletrônico
- **“Domain Name System - DNS”** .. Serviço de Diretório
- **“Point to Point - P2P”** .. Aplicações P2P

## 2 – Camada de Aplicação

### 2.2 - World Wide Web e o HTTP

- World Wide Web [Berners-Lee, 1994] ... aplicação que chamou a atenção do público e transformou a Internet na única Rede de Dados.
- .. proporcionou uma transformação drástica na maneira como usuários interagem dentro e fora de seus ambientes de trabalho.
- “**benefícios / vantagens**” no uso da World Wide Web ..
- “**conexões**” .. links e sites de busca que ajudam a navegar pelo oceano de sites com todo tipo de informação e serviço.
- “**dispositivos gráficos**” .. imagens, gráficos e vídeos facilitam o aprendizado e estimulam os sentidos dos usuários.
- “**funcionalidades**” .. formulários, applets Java e muitos outros recursos possibilitam a interação com páginas e sites.
- “**plataformas**” .. ambiente para aplicações de sucesso que surgiram após 2003, incluindo YouTube, Gmail e Facebook



## 2 – Camada de Aplicação / 2.2 - World Wide Web e o HTTP

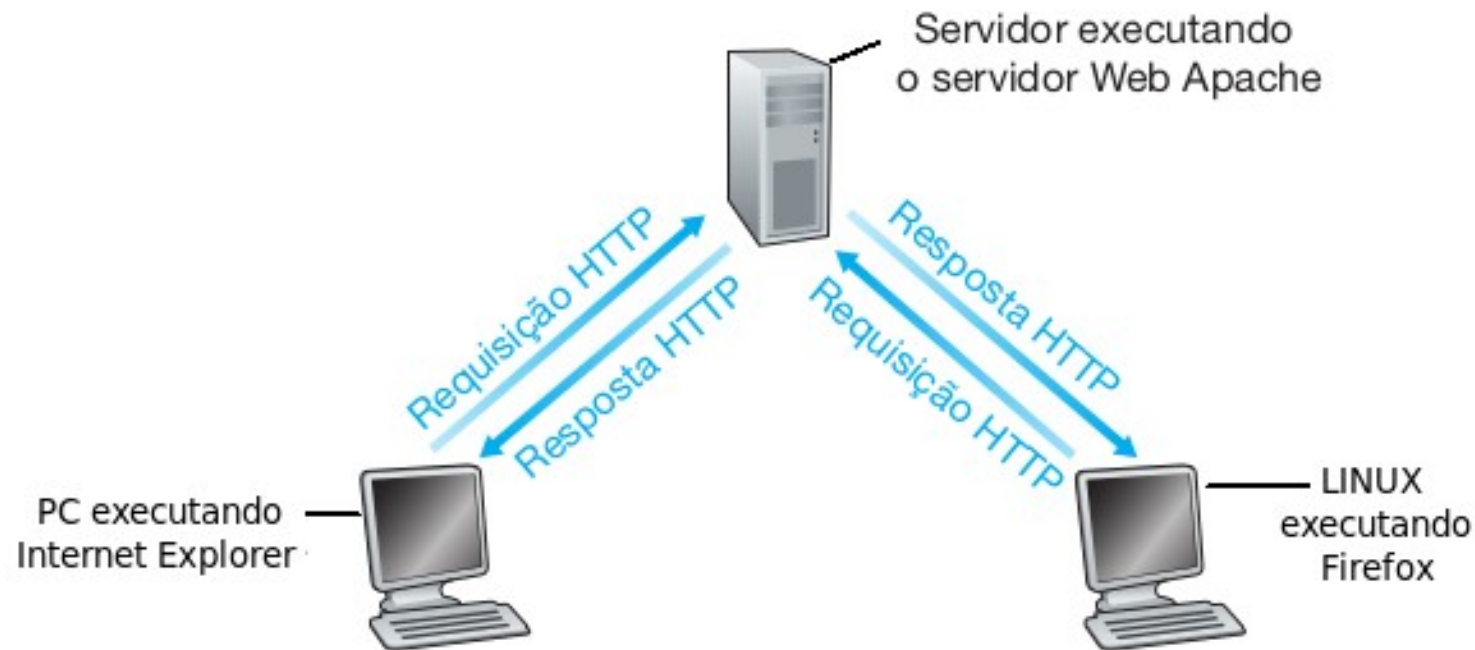
### 2.2.1 – Descrição Geral do HTTP

- HyperText Transport Protocol .. [RFC 1945] .. [RFC 2616] .. protocolo executado entre cliente e servidor (sistemas finais) e que conversam entre si por meio da troca de mensagens HTTP.
- Página Web (documento) .. coletânea de objetos, onde cada objeto é apenas um arquivo, tal como um arquivo HTML, uma imagem JPEG, um applet Java, ou um clipe de vídeo.
- .. ou seja, maioria das páginas Web é constituída de um arquivo base HTML e diversos objetos referenciados na forma de URLs.
- Universal Resource Locator (URL) .. abriga o nome de hospedeiro (“hostname” do servidor) que abriga o objeto e o caminho do objeto.
- e.g., **`http://www.someSchool.edu/someDepartment/picture.gif`**

## 2 – Camada de Aplicação / 2.2 - World Wide Web e o HTTP

### ... 2.2.1 – Descrição Geral do HTTP

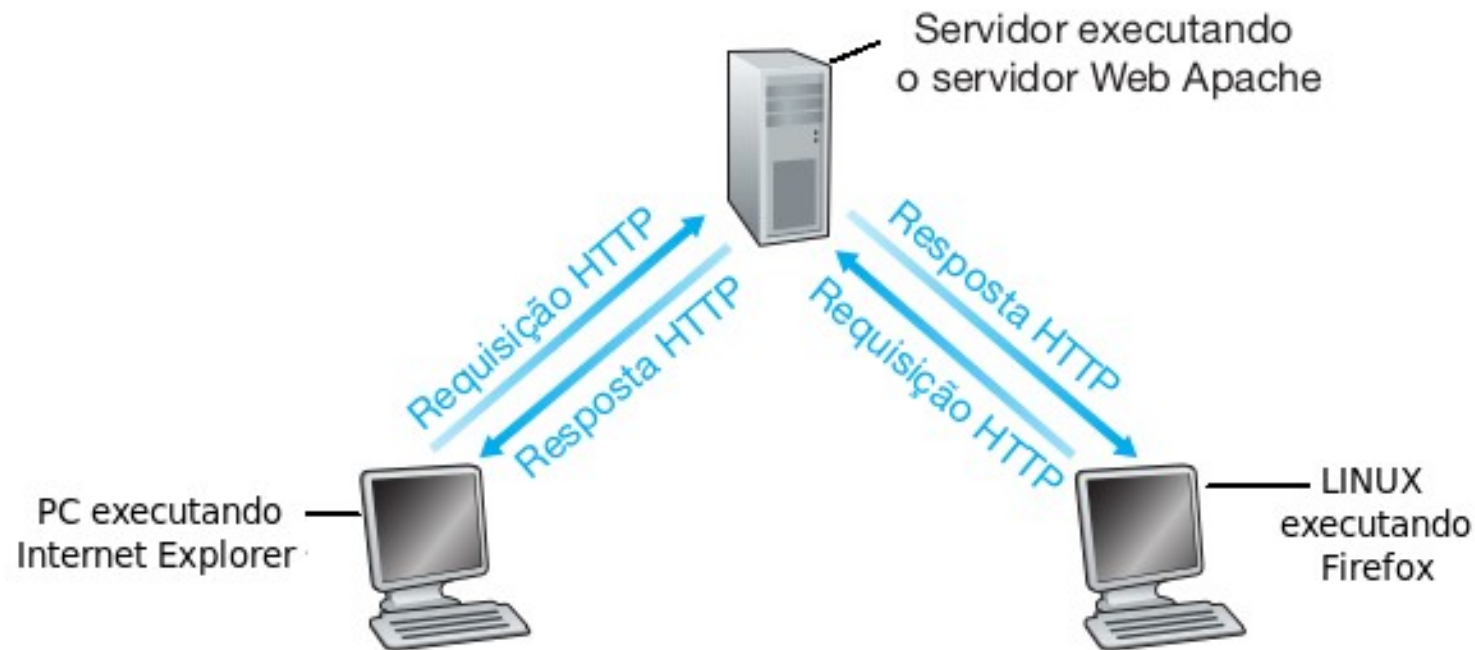
- HTTP define como os clientes requisitam páginas aos servidores e como os servidores as transferem aos clientes.
- e.g., quando um usuário requisita uma página Web (p.ex., seleciona o link) e, na sequência o navegador envia ao servidor mensagem de requisição HTTP para os objetos da página (uma por vez).



## 2 – Camada de Aplicação / 2.2 - World Wide Web e o HTTP

### ... 2.2.1 – Descrição Geral do HTTP

- e.g., quando um usuário requisita uma página Web (p.ex., seleciona o link) e, na sequência o navegador envia ao servidor mensagem de requisição HTTP para os objetos da página (uma por vez).
- cliente .. navegador requisita, recebe, “exibe” objetos Web.
- servidor .. servidor Web envia objetos em resposta as requisições.



## 2 – Camada de Aplicação / 2.2 - World Wide Web e o HTTP

### ... 2.2.1 – Descrição Geral do HTTP

- “**não mantém estado**” .. HTTP é “sem estado” .. servidor não guarda informações sobre requisições passadas pelo cliente.
- ... protocolos que mantêm “estado” são complexos, pois a história passada (estado) deve ser mantida.
- “**alguns problemas**” .. se servidor/cliente falharem, suas visões do “estado” podem ser incoerentes, exigindo que sejam reconciliadas.

## 2 – Camada de Aplicação / 2.2 - World Wide Web e o HTTP

### ... 2.2.1 – Descrição Geral do HTTP

- HTTP .. usa o Protocolo TCP como seu protocolo de transporte subjacente (em vez de rodar em cima do Protocolo UDP).
- “**cliente inicia conexão**” .. TCP com Servidor Web na Porta 80, na sequência servidor aceita conexões TCP de Clientes e, somente depois de se estabelecer as conexões, mensagens são trocadas.
- “**mensagens HTTP**” .. (camada de aplicação) são trocadas entre Navegador (Cliente HTTP) e Servidor Web (Servidor HTTP).
- ... não havendo mais troca de dados ou a pedido de uma das partes (Navegador ou Servidor HTTP) a conexão TCP é fechada.

## 2 – Camada de Aplicação / 2.2 World Wide Web e o HTTP

### 2.2.2 – Conexão Persistente e Não-Persistente

- .. muitas aplicações, cliente e servidor se comunicam por um período prolongado de tempo, o que pode justificar a manutenção da conexão.
- “**dúvida**” .. enviar cada par de requisição e de resposta por conexão TCP distinta ou todas as requisições e suas respostas devem ser enviadas por uma mesma conexão TCP ?
- “**conexões não persistentes**” .. cada par de requisição e de resposta é enviado por conexões TCP distintas, ou seja, após a troca de informações (requisição / resposta) a conexão é encerrada.
- “**conexão persistente**” .. todas as requisições e suas respostas devem ser enviadas por uma mesma conexão TCP.

## 2 – Camada de Aplicação / 2.2 World Wide Web e o HTTP

### ... 2.2.2 – Conexão Persistente e Não-Persistente

- e.g., Seja a URL para o arquivo-base HTTP e que referencia 10 outros objetos .. `http://www.someSchool.edu/someDepartment/home.index`
- 1) Cliente HTTP inicia uma conexão TCP para o Servidor no endereço “`www.someSchool.edu`” e na Porta 80 (default no HTTP) .. lembre-se que há sockets associados ao Cliente e ao Servidor.
  - 2) Cliente HTTP envia uma mensagem de requisição HTTP ao Servidor por meio de seu socket, que inclui o caminho do dado requisitado .. “`/someDepartment/home.index`”.
  - 3) Servidor HTTP recebe a msg. de requisição por meio de seu socket, extrai o objeto “`/someDepartment/home.index`”, encapsula o mesmo em uma msg. de resposta HTTP e a envia ao cliente pelo socket.

## 2 – Camada de Aplicação / 2.2 World Wide Web e o HTTP

### ... 2.2.2 – Conexão Persistente e Não-Persistente

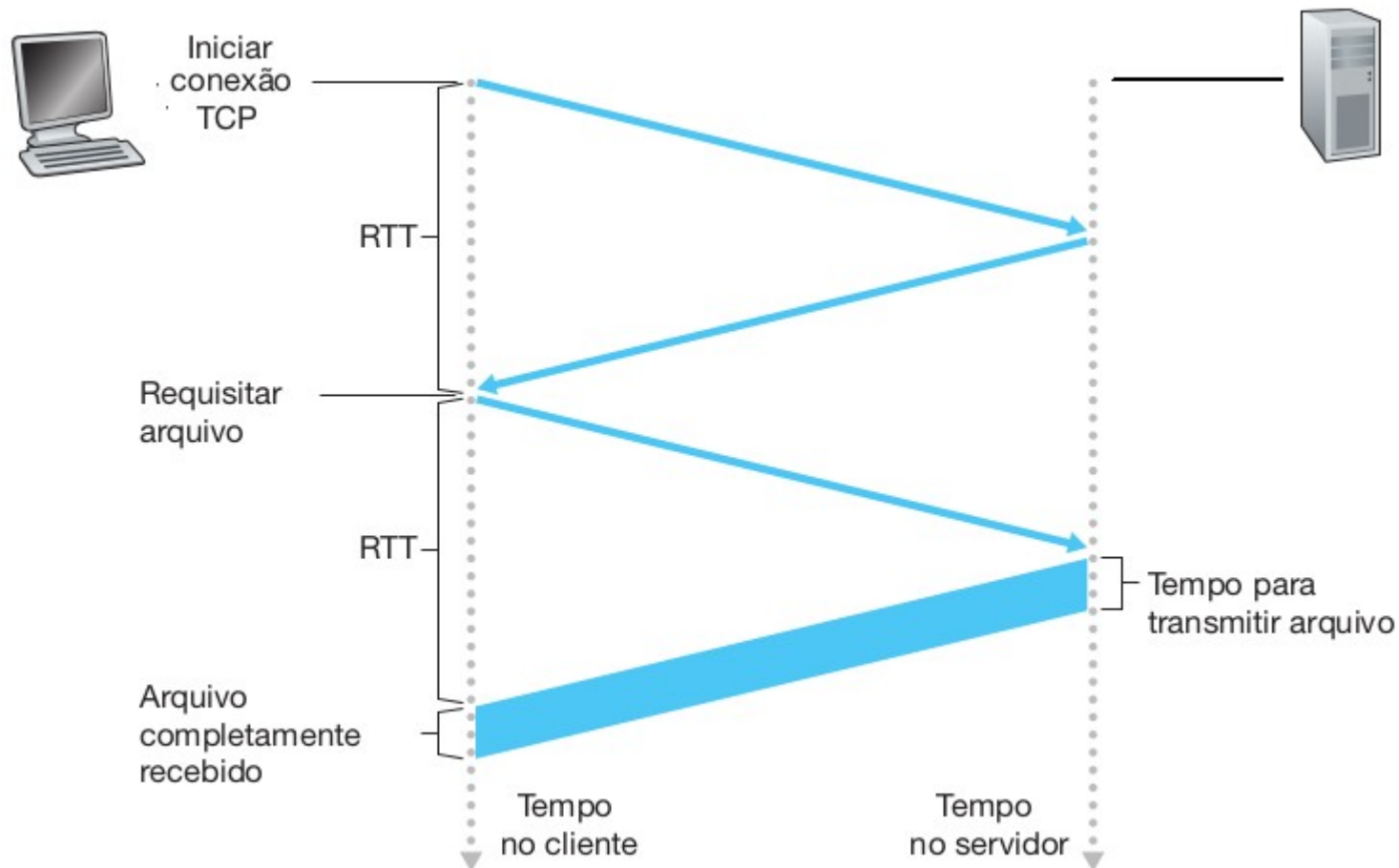
- Seja a URL para o arquivo-base HTTP ..  
`http://www.someSchool.edu/someDepartment/home.index`
- 4) Cliente HTTP recebe a mensagem de resposta e a conexão TCP é encerrada. .. mensagem indica que o objeto encapsulado é um arquivo HTML que ao ser analisado referencia aos 10 objetos JPEG.
  - 5) Servidor HTTP ordena ao TCP que encerre a conexão TCP, embora, o encerramento só ocorra quando tiver certeza de que o cliente recebeu a mensagem de resposta intacta.
  - 6) Etapas de 1) a 4) são repetidas para cada um dos objetos JPEG referenciados no arquivo “/someDepartment/home.index”.



## 2 – Camada de Aplicação / 2.2 World Wide Web e o HTTP

### ... 2.2.2 – Conexão Persistente e Não-Persistente

- Tempo de Resposta do HTTP Não-Persistente .. tempo para uma pequena mensagem trafegar do cliente ao servidor e retornar.



## 2 – Camada de Aplicação / 2.2 World Wide Web e o HTTP

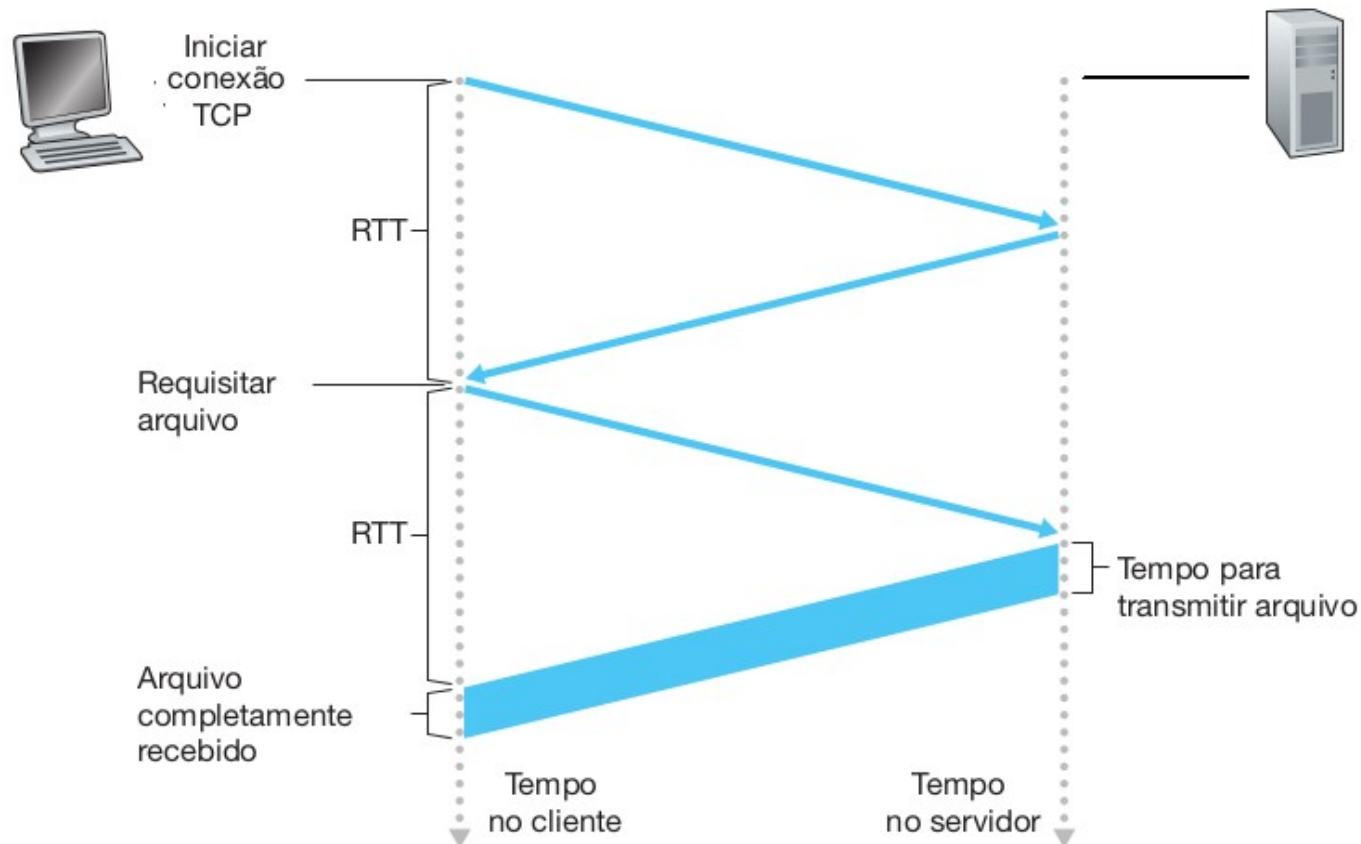
### ... 2.2.2 – Conexão Persistente e Não-Persistente

- Tempo de Resposta do HTTP Não-Persistente .. tempo para uma pequena mensagem trafegar do cliente ao servidor e retornar.
- 01 RTT para iniciar a conexão TCP (estabelecimento da conexão);
- 01 RTT para a resposta do Cliente ao estabelecimento da conexão que também carrega a Requisição HTTP (objeto sendo solicitado) + resposta contento o objeto da requisição HTTP;
- Tempo de Transmissão do objeto solicitado na Requisição HTTP.
- “**tempo total**” .. é igual  $2 * \text{RTT} + \text{Tempo de Transmissão do Objeto}$ .

## 2 – Camada de Aplicação / 2.2 World Wide Web e o HTTP

### ... 2.2.2 – Conexão Persistente e Não-Persistente

- Tempo de Resposta do HTTP Não-Persistente .. tempo para uma pequena mensagem trafegar do cliente ao servidor e retornar.
- “**tempo total**” .. é igual  $2 * \text{RTT} + \text{Tempo de Transmissão do Objeto}$ .

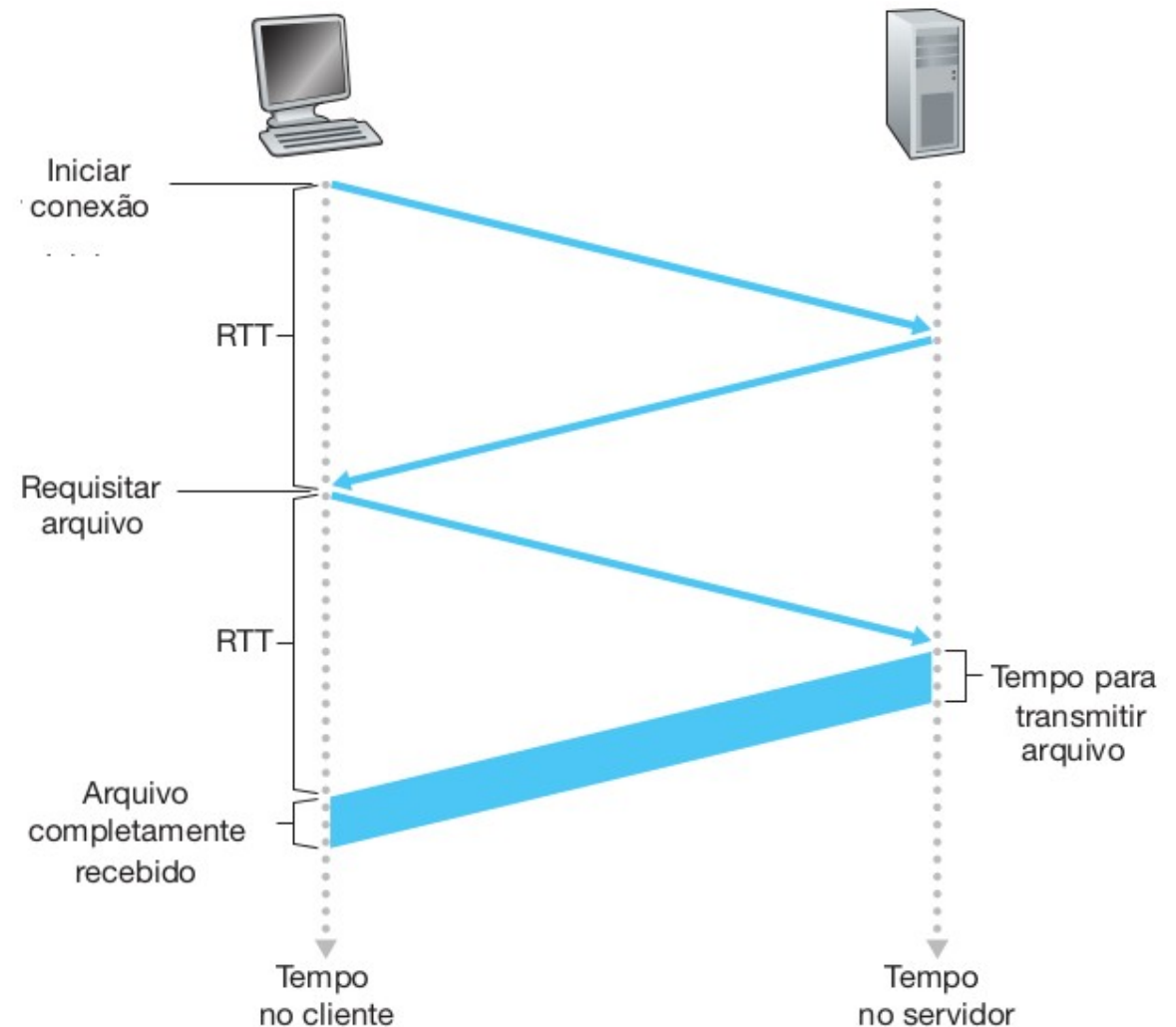


## 2 – Camada de Aplicação / 2.2 World Wide Web e o HTTP

### ... 2.2.2 – Conexão Persistente e Não-Persistente

- “**problemas**” .. HTTP Não-Persistente:

- 1) requer 02 RTTs por objeto (por requisição);
- 2) overhead do SO para *cada* conexão TCP;
- 3) navegador podem abrir várias conexões TCP em paralelo para buscar objetos referenciados, o que, replica todo o processo de estabelecimento de conexões para cada objeto.



## 2 – Camada de Aplicação / 2.2 World Wide Web e o HTTP

### ... 2.2.2 – Conexão Persistente e Não-Persistente

- “**vantagens**” .. HTTP Persistente:

- 1) servidor manter a conexão aberta depois de enviar a resposta (permitindo que novos objetos sejam transmitidos);
- 2) mensagens “requisição”/“resposta” são trocadas entre cliente/servidor (conexão mantida facilita todo o processo de transferência de dados);
- 3) cliente solicita novas requisições para objetos assim que encontra um objeto referenciado ao receber a resposta na requisição anterior;
- 4) economia de 01 RTT (no mínimo) para cada um dos objetos referenciados, sem contar a economia no encerramento de conexões.

## 2 – Camada de Aplicação / 2.2 - World Wide Web e o HTTP

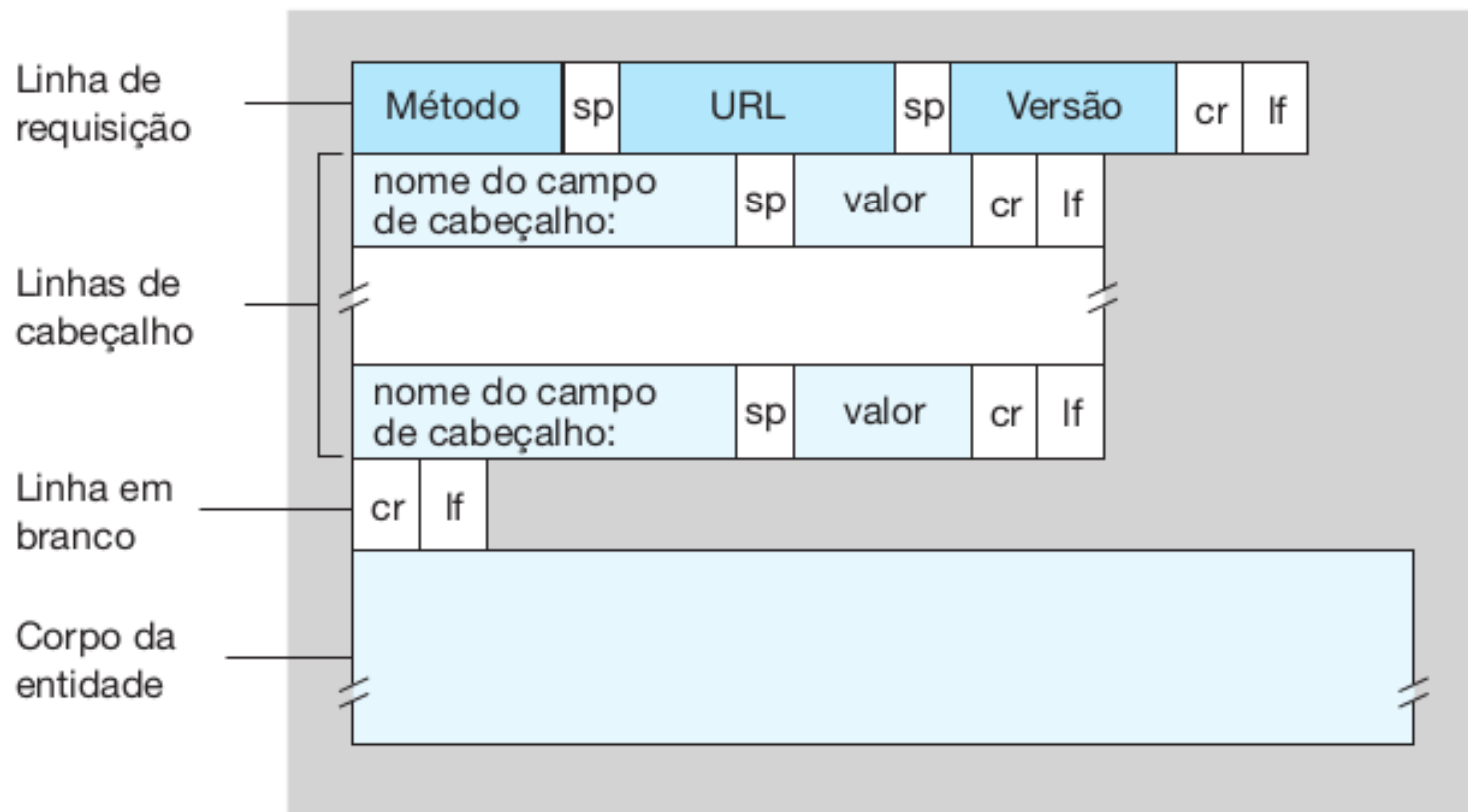
### 2.2.3 – Formato da Mensagem HTTP

- [RFC 1945; RFC 2616] incluem as definições dos formatos das mensagens HTTP, ou seja, 02 tipos .. requisição e resposta.
- Mensagem Típica de Requisição HTTP:  
GET /somedir/page.html HTTP/1.1  
Host: www.someschool.edu  
Connection: close  
User-agent: Mozilla/5.0  
Accept-language: fr
- Linha de Requisição .. GET .. POST .. HEAD .. PUT .. DELETE
- Linhas de Cabeçalho
- **“carriage-return” e “line-feed”**

## 2 – Camada de Aplicação / 2.2 - World Wide Web e o HTTP

### ... 2.2.3 – Formato da Mensagem HTTP

- “**corpo de entidade**” .. fica vazio com o método GET, mas é utilizado com o método POST (contempla os informações e dados).
- Cliente HTTP geralmente usa o método POST quando o usuário preenche um formulário .. p.ex., palavras de busca a um site buscador.



## 2 – Camada de Aplicação / 2.2 - World Wide Web e o HTTP

### ... 2.2.3 – Formato da Mensagem HTTP

- [RFC 1945; RFC 2616] incluem as definições dos formatos das mensagens HTTP, ou seja, 02 tipos .. requisição e resposta.
- “**mensagem típica**” 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

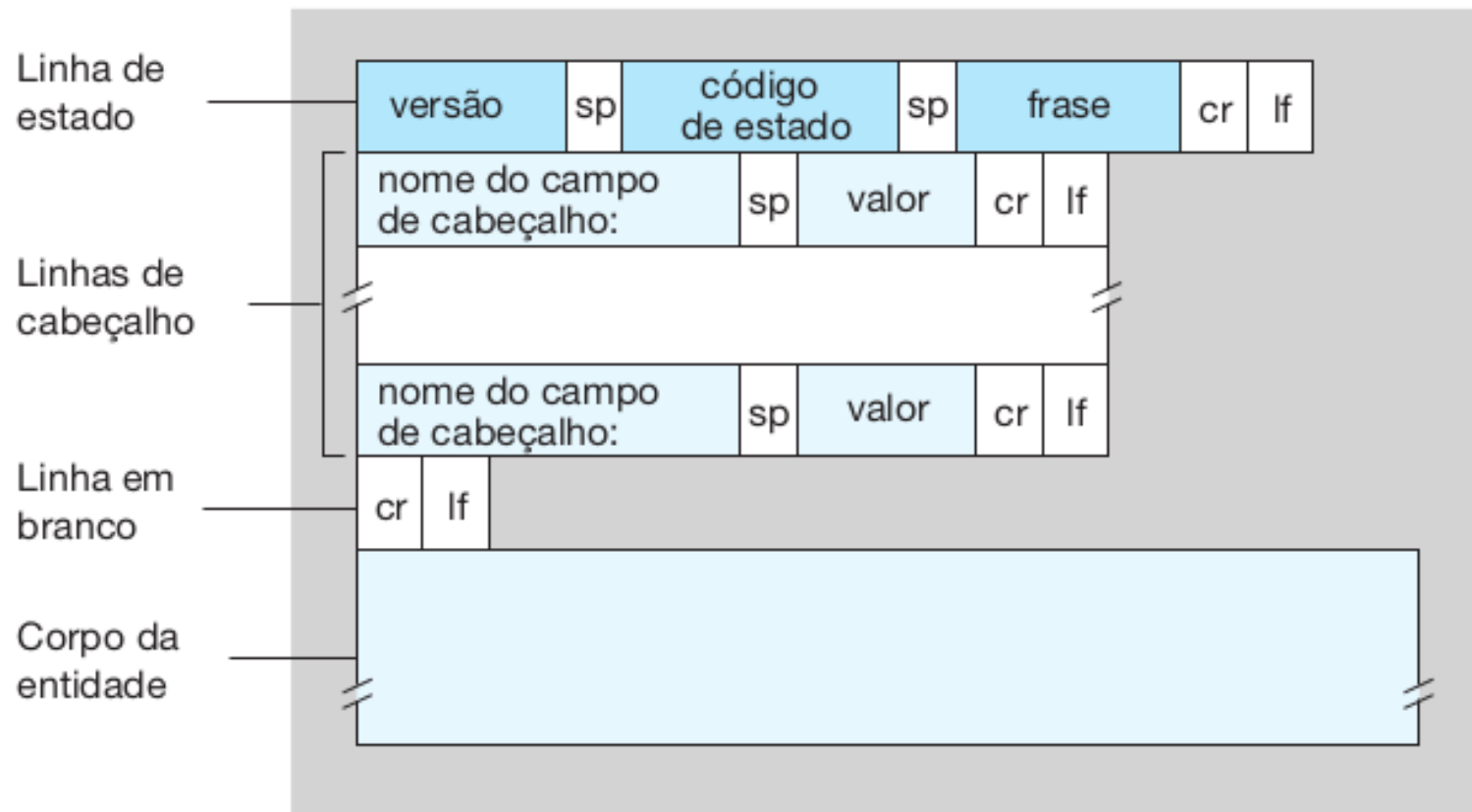
(dados dados dados dados dados ...)



## 2 – Camada de Aplicação / 2.2 - World Wide Web e o HTTP

### ... 2.2.3 – Formato da Mensagem HTTP

- “**mensagem de resposta**” .. contempla um código da resposta.
- servidor usa “**connection: close**” para informar ao cliente que fechará a conexão TCP após enviar a mensagem.



## 2 – Camada de Aplicação / 2.2 - World Wide Web e o HTTP

### ... 2.2.3 – Formato da Mensagem HTTP

- Códigos de Estado e frases comuns associadas aos códigos:
- **200 OK** : requisição bem-sucedida.
- **301 Moved Permanently** .. objeto requisitado foi removido em definitivo e nova URL é especificada no cabeçalho “location” da resposta, assim, software do cliente recuperará automaticamente o novo URL.
- **400 Bad Request** .. código genérico de erro que indica que a requisição não pôde ser entendida pelo servidor.
- **404 Not Found** .. documento requisitado não existe no servidor.
- **505 HTTP Version Not Supported** .. versão do protocolo HTTP requisitada não é suportada pelo servidor.

## 2 – Camada de Aplicação / 2.2 World Wide Web e o HTTP

### 2.2.4 – Cookies na Interação Cliente/Servidor

- Cookies [RFC 6265] .. permitem que sites monitorem seus usuários e, hoje, maioria dos sites comerciais utiliza cookies.
- **“componentes dos cookies”** .. são 04 componentes ..
- 1) linha de cabeçalho de cookie na mensagem de resposta HTTP.
- 2) linha de cabeçalho de cookie na mensagem de requisição HTTP.
- 3) arquivo de cookie mantido no “host” do usuário e gerenciado pelo navegador do usuário (pode ser eliminado limpando a cache).
- 4) banco de dados mantido e gerenciado no site (objeto de requisição).

## 2 – Camada de Aplicação / 2.2 World Wide Web e o HTTP

### ... 2.2.4 – Cookies na Interação Cliente/Servidor

- e.g., Susan que acessa com frequência a Web através do Internet Explorer, acaba de acessar o “amazon.com” pela primeira vez, mas, no passado recente, já havia visitado o “ebay.com”.
- ... quando a requisição chega ao servidor da Amazon, ele cria um nro. de identificação exclusivo e uma entrada no seu banco de dados, que é indexado pelo número de identificação.
- ... então, o servidor da “amazon.com” responde ao navegador de Susan, incluindo na resposta HTTP um cabeçalho “set-cookie: 1678” que contém o nro. de identificação.

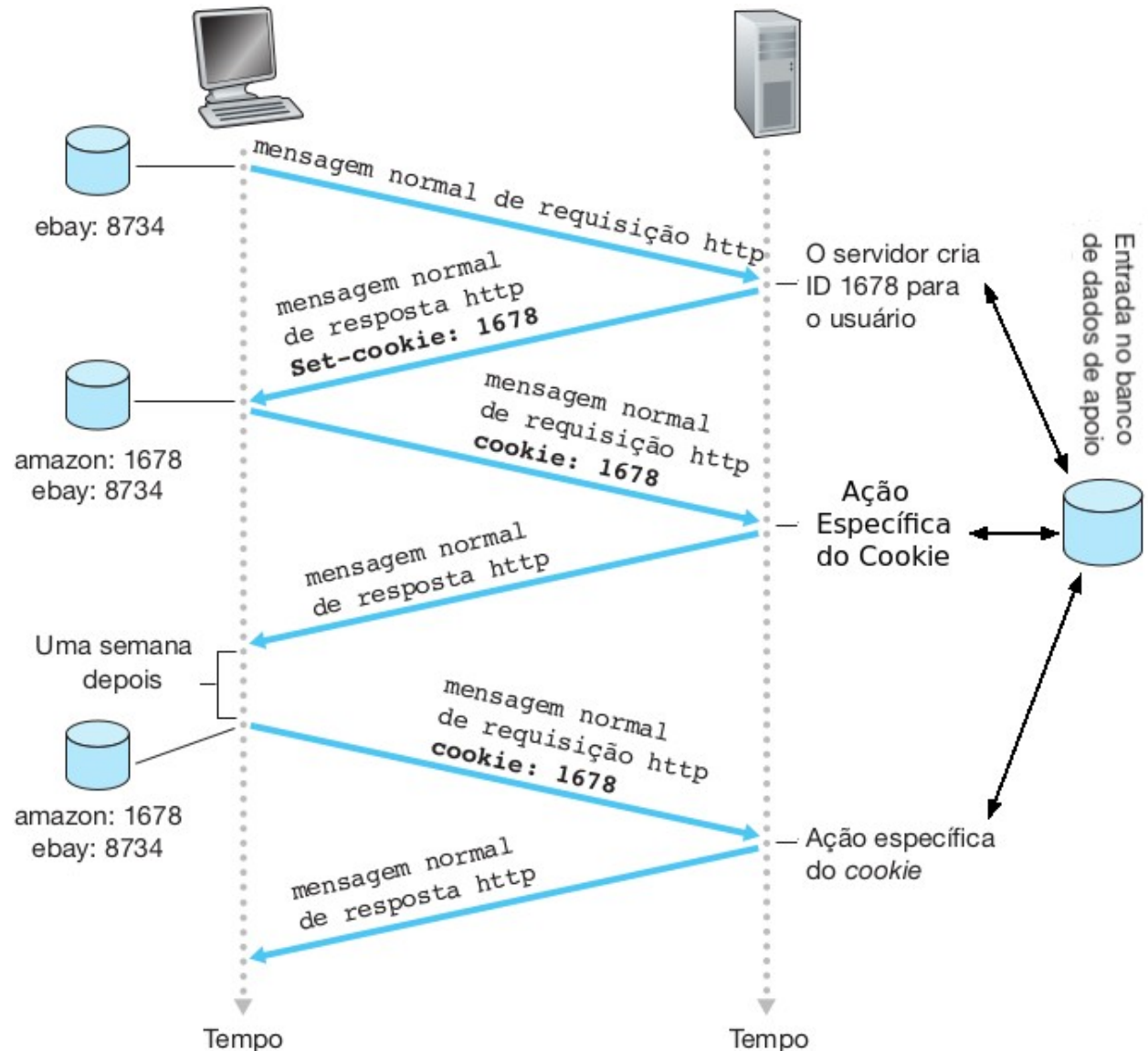
**Set-cookie: 1678**

## 2 – Camada de Aplicação / 2.2 World Wide Web e o HTTP

### ... 2.2.4 – Cookies na Interação Cliente/Servidor

- e.g., Susan acaba de acessar o site “amazon.com” pela primeira vez.

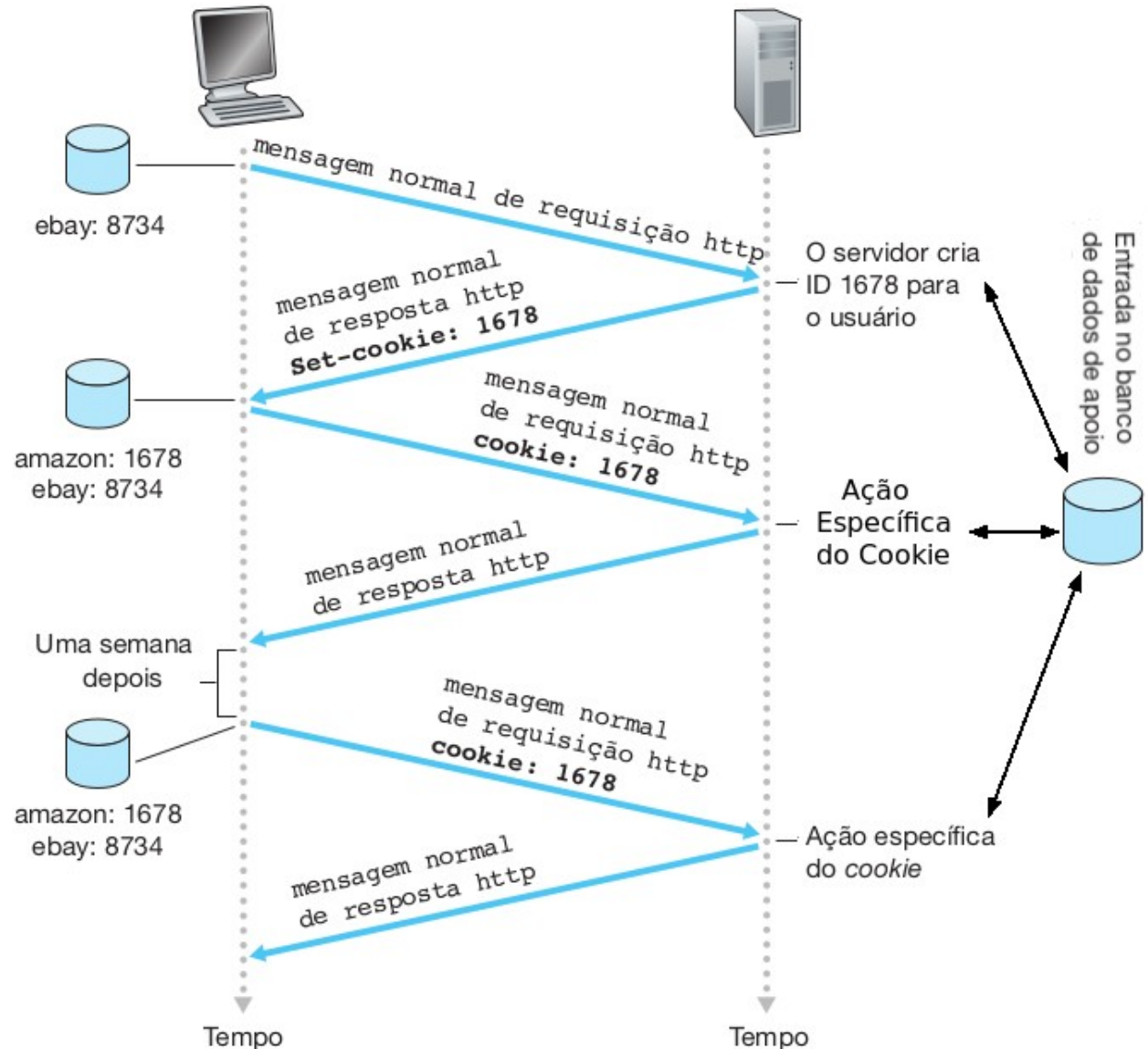
Set-cookie: 1678



## 2 – Camada de Aplicação / 2.2 World Wide Web e o HTTP

### ... 2.2.4 – Cookies na Interação Cliente/Servidor

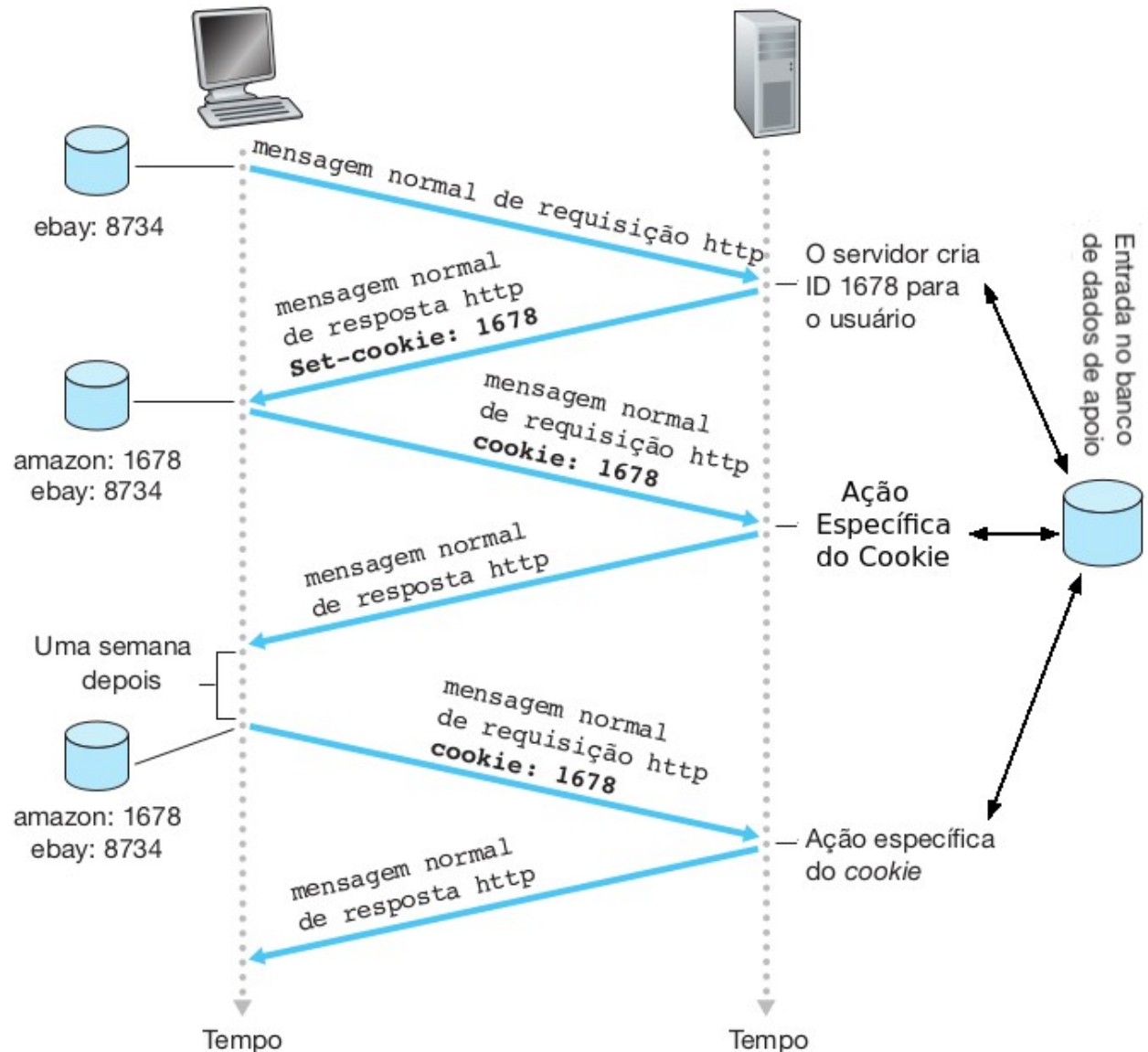
- Ao receber a resposta HTTP, o navegador de Susan identifica o cabeçalho “set-cookie:”  
...
- ... e anexa esta linha ao arquivo de “cookies” que gerencia.



## 2 – Camada de Aplicação / 2.2 World Wide Web e o HTTP

### ... 2.2.4 – Cookies na Interação Cliente/Servidor

- Susan já visitou o “ebay.com” e, por isso, contempla “in loco” entrada de “cookie” para “ebay.com”.
- .. toda requisição de página enquanto navega pelo site da “amazon”, gera consulta no seu arquivo de “cookies”.

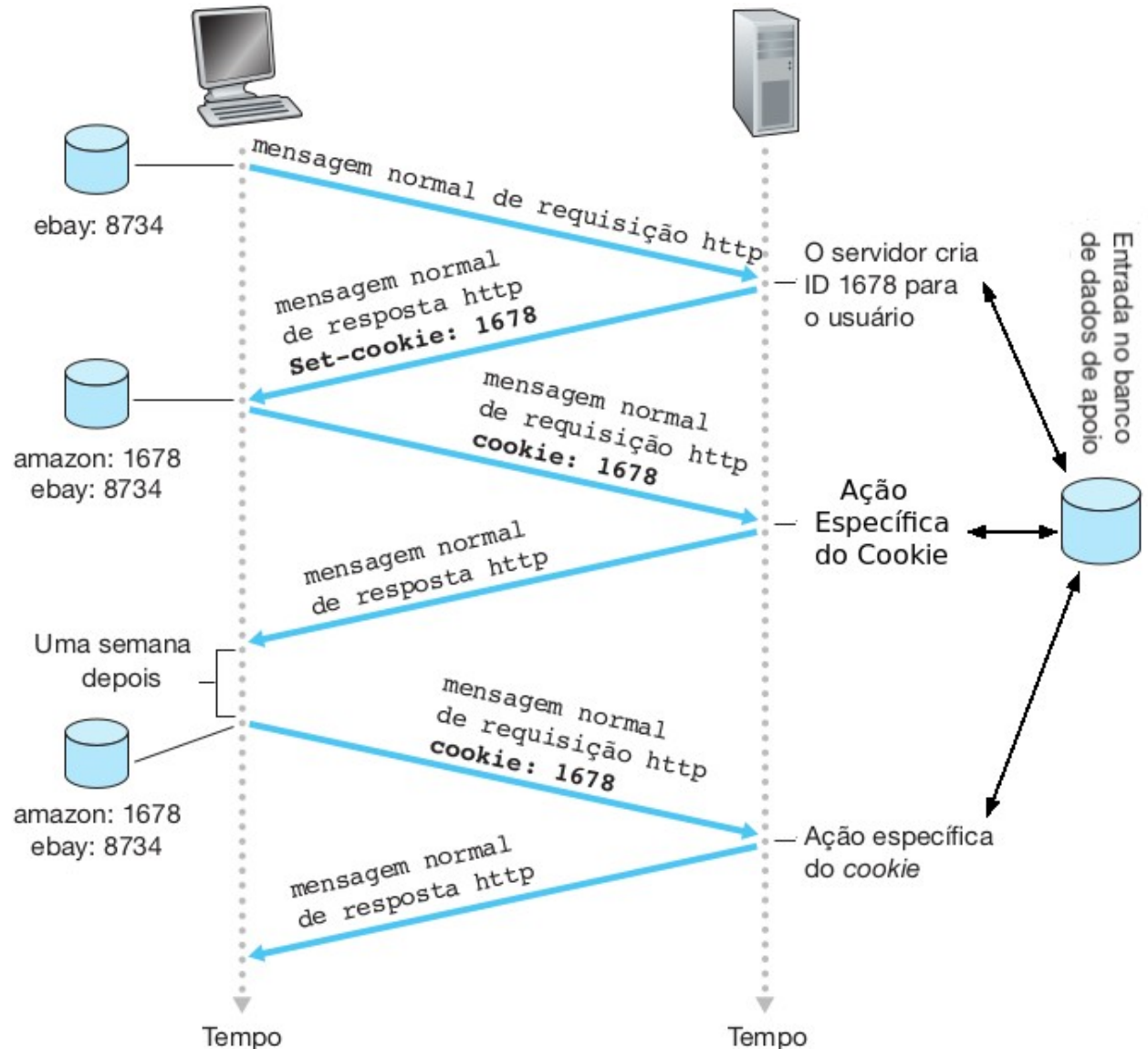




## 2 – Camada de Aplicação / 2.2 World Wide Web e o HTTP

### ... 2.2.4 – Cookies na Interação Cliente/Servidor

- “amazon.com” pode monitorar a atividade de Susan em seu site, embora não saiba que o nome dela é Susan.





## 2 – Camada de Aplicação / 2.2 World Wide Web e o HTTP

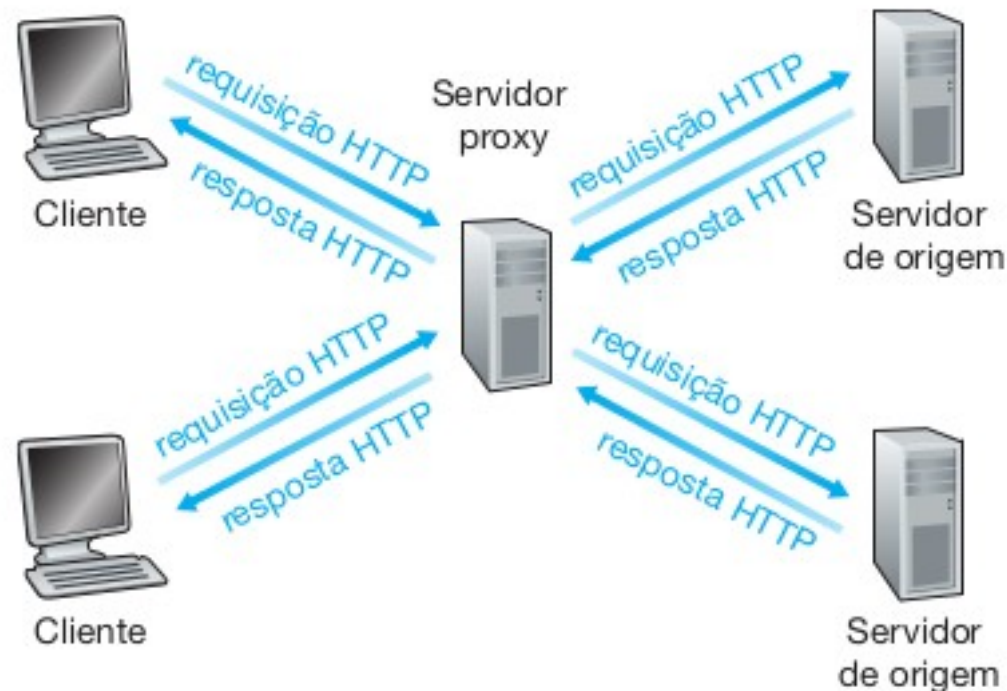
### ... 2.2.4 – Cookies na Interação Cliente/Servidor

- Cookies podem contemplar “autorização”, “carrinhos de compras”, “recomendações” e “estado da sessão” do usuário (e-mail da Web).
- “**problemas**” com Cookies .. é a Privacidade dos Dados ..
- cookies permitem que os sites descubram muito sobre o usuário, ou seja, nas situações em que o usuário fornece, p.ex., nome e e-mail.
- Como manter “Estado” ?
- “**extremidades do protocolo**” (cliente e servidor) .. mantêm estado no emissor e no receptor para múltiplas transações.
- “**cookies**” .. mensagens HTTP transportam estado, mas o estado é mantido pelo navegador no lado cliente e pelo “site” no lado servidor.

## 2 – Camada de Aplicação / 2.2 World Wide Web e o HTTP

### 2.2.5 – Cache World Wide Web

- “**objetivo**” .. satisfazer a requisição do cliente sem envolver necessariamente o servidor de origem (origem dos dados).
- “**cache web**” ou “**servidor proxy**” .. entidade da rede que atende requisições HTTP em nome de um servidor Web de origem.



## 2 – Camada de Aplicação / 2.2 World Wide Web e o HTTP

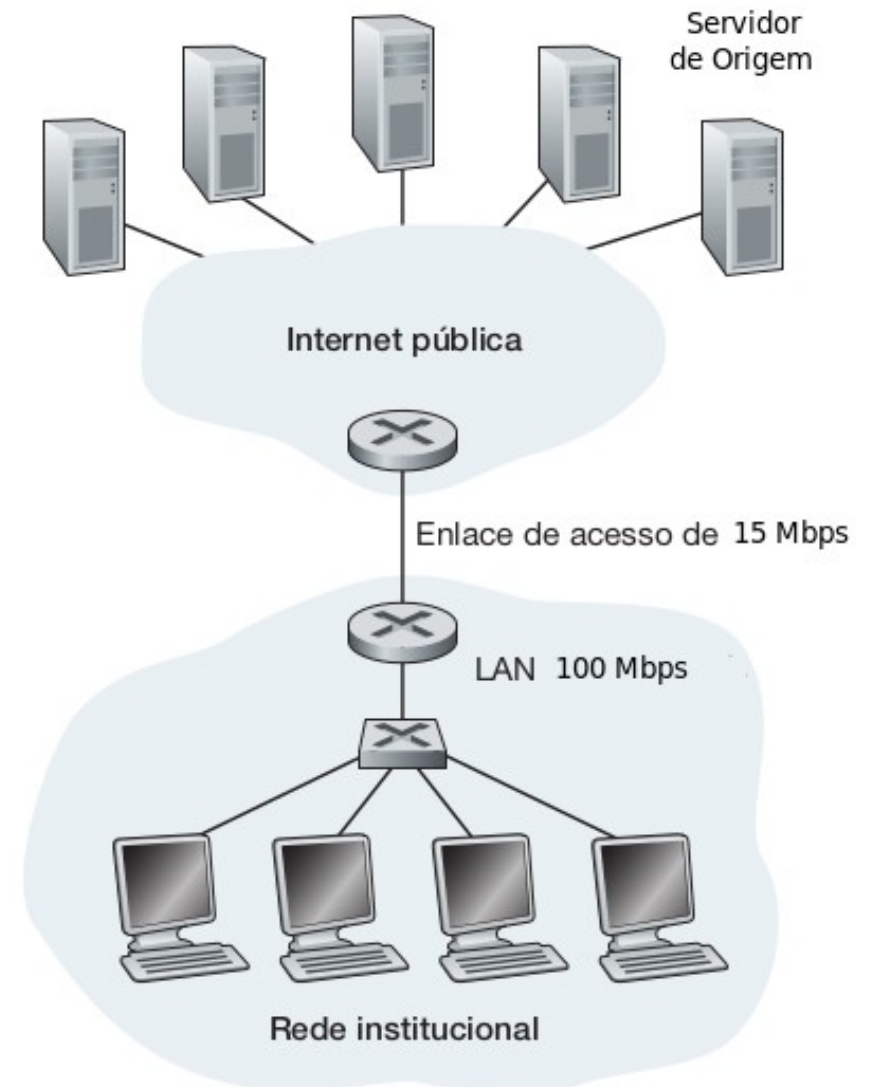
### ... 2.2.5 – Cache World Wide Web

- “**configuração**” .. usuário configura o navegador para acessos via cache, assim, o navegador envia todas as requisições HTTP ao cache.
- .. se objeto estiver na cache .. cache retorna objeto.
- .. se objeto não estiver na cache .. cache requisita objeto do servidor de origem, aguarda o retorno e, na sequência, retorna o objeto.
- “**cache web**” .. atua como cliente e servidor e, normalmente, é instalado por ISP, p.ex., universidade, empresa ou residencial.
- “**vantagens**” da Cache Web:
  - .. reduz tempo de resposta à requisição do cliente.
  - .. reduz tráfego no enlace de acesso de uma instituição.

## 2 – Camada de Aplicação / 2.2 World Wide Web e o HTTP

### ... 2.2.5 – Cache World Wide Web

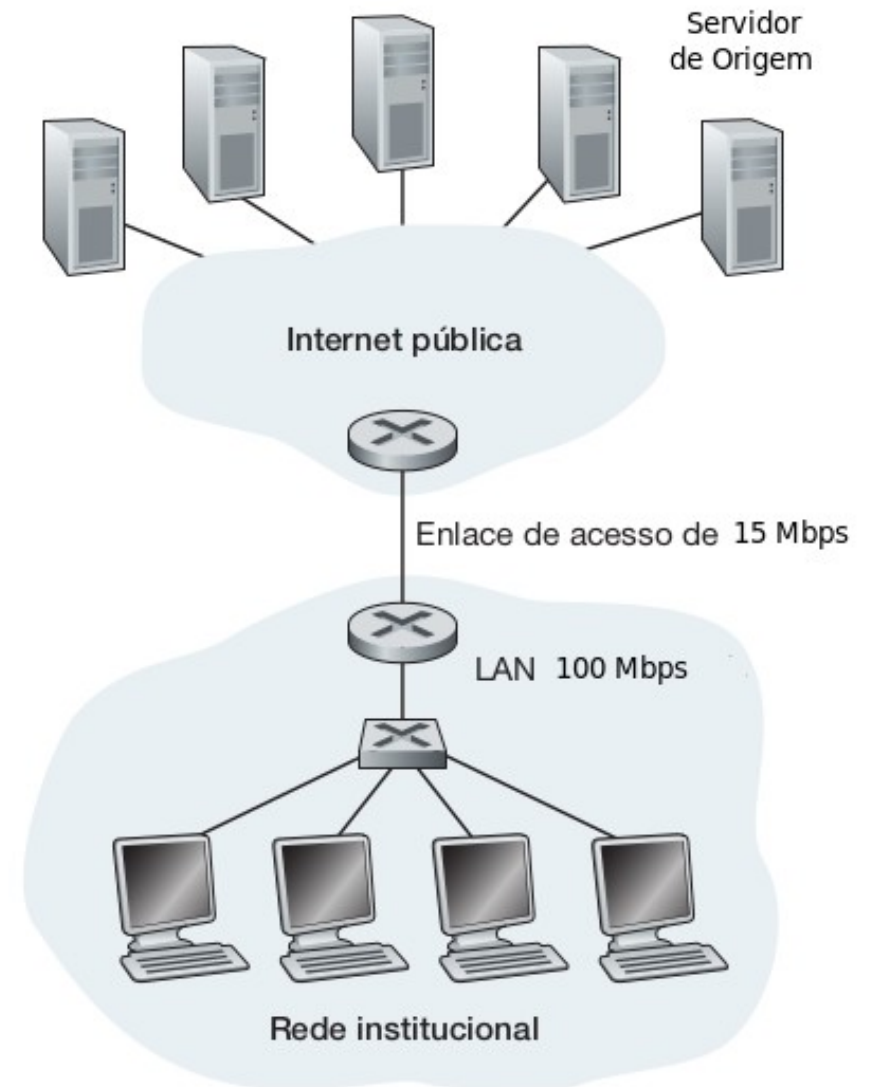
- e.g., considere uma rede institucional de alta velocidade, p.ex., 100 Mbps e o roteador institucional ligado a um roteador da Internet por um enlace de 15 Mbps (exemplos no livro estão com valores diferentes).
- ... considere objetos de busca sejam de 1.000.000 bits (valores sugeridos tem por objetivo facilitar os cálculos).
- ... considere a taxa de requisição dos navegadores da instituição aos servidores de origem = 15 req. por segundo e atraso de ida e volta p/ servidores de origem = 2 segundos.



## 2 – Camada de Aplicação / 2.2 World Wide Web e o HTTP

### ... 2.2.5 – Cache World Wide Web

- $(15 \text{ requisições por seg}) * (1.000.000 \text{ bits por requisição}) = 15 \text{ Mbps}$
- $15 \text{ Mbps} / 100 \text{ Mbps (LAN)} = 15\% \text{ de utilização da LAN.}$
- $(15 \text{ requisições por seg}) * (1.000.000 \text{ bits por requisição}) = 15 \text{ Mbps}$
- $15 \text{ Mbps} / 15 \text{ Mbps (Enlace do Roteador)} = 100\% \text{ de utilização enlace.}$



## 2 – Camada de Aplicação / 2.2 World Wide Web e o HTTP

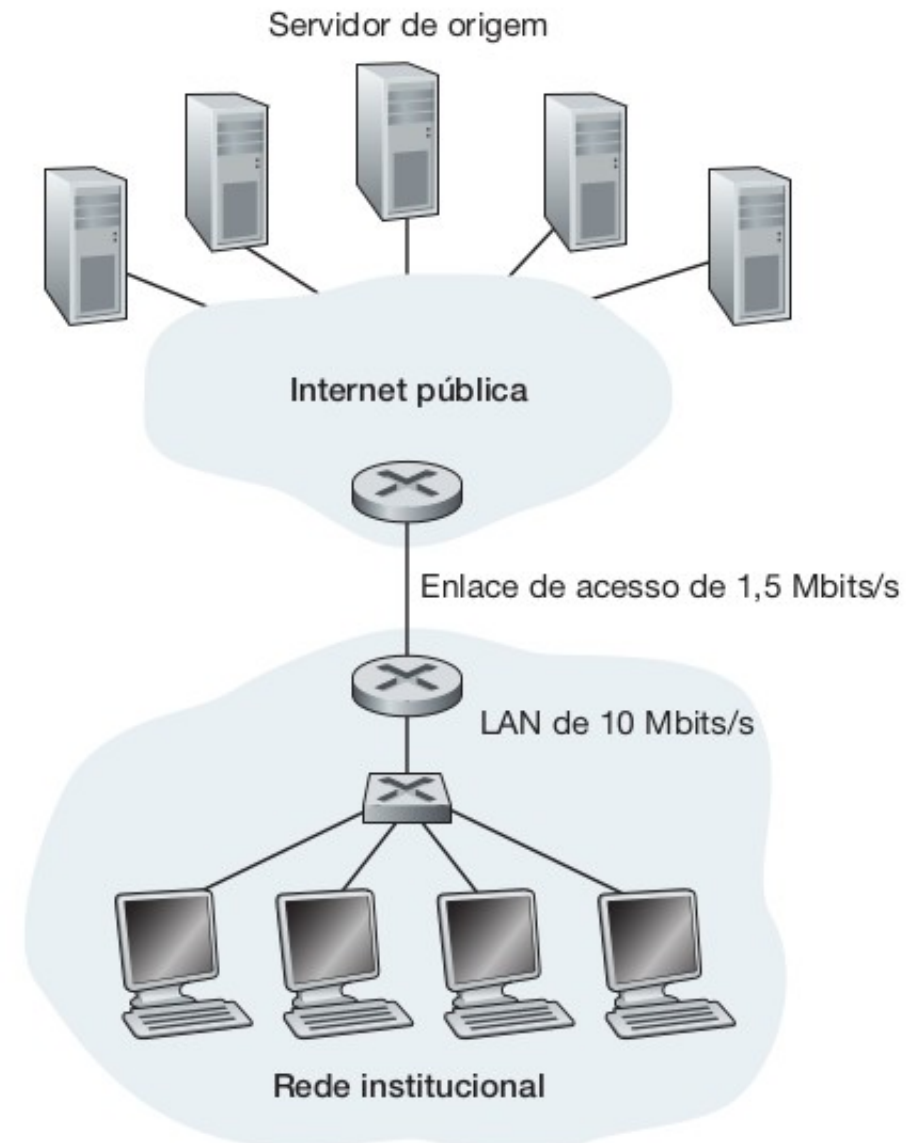
### ... 2.2.5 – Cache World Wide Web

- e.g., considere uma rede institucional de alta velocidade, p.ex., 100 Mbps e o roteador institucional ligado a um roteador da Internet por um enlace de 15 Mbps (exemplos no livro estão com valores diferentes).
- Intensidade de Tráfego .. taxa de utilização na LAN = 15%
- Utilização no Enlace de Acesso = 100%
- Atraso Total = Atraso da Internet + Atraso do Acesso + Atraso da LAN, ou seja, é igual a .. 2 s + x minutos + y milissegundos

## 2 – Camada de Aplicação / 2.2 World Wide Web e o HTTP

### ... 2.2.5 – Cache World Wide Web

- Obs.: ... no enunciado anterior, os valores citados são 15 Mbps (enlace) e 100 Mbps na LAN.
- Já a Figura (Livro Texto) informa 1,5 Mbps de capacidade do Enlace entre Roteadores e 10 Mbps na LAN.



## 2 – Camada de Aplicação / 2.2 World Wide Web e o HTTP

### ... 2.2.5 – Cache World Wide Web

- “**solução**” .. aumentar largura de banda do enlace, p.ex., 100 Mbps.
- .. alguns resultados .. Utilização na LAN = 15%
- Utilização no Enlace de Acesso = 15%
- Atraso Total = Atraso da Internet + Atraso do Acesso + Atraso da LAN, ou seja, é igual a .. 2 s + x minutos + y milisegundos
- “**conclusão**” .. atualização um TANTO quanto DISPENDIOSA !!



## 2 – Camada de Aplicação / 2.2 World Wide Web e o HTTP

### ... 2.2.5 – Cache World Wide Web

- “**algumas considerações**” .. quanto a Cache Web:
- taxa de resposta local = fração de requisições atendidas por um “cache” (normalmente entre 0,2 a 0,7) – supor que seja 0,4.
- suposição que 40% (0,40) das requisições serão atendidas quase de imediato pelo cache e, portanto, pela LAN de alta velocidade.
- considere que requisições serão atendidas quase de imediato pelo cache, digamos, em 10 ms ou 0,010 segundos.
- suposição de que 60% das requisições serão atendidos pelos servidores de origem e, portanto, necessitam do enlace.
- .. tendo por base as considerações, o atraso médio =  $0,4 * (0,01 \text{ segundos}) + 0,6 * (2,01 \text{ segundos}) = 1,21 \text{ segundos}$
- “**conclusão**” .. tempo de resposta é menor do que a 2ª proposta, além de não requerer que a instituição atualize seu enlace.

## 2 – Camada de Aplicação / 2.2 World Wide Web e o HTTP

### 2.2.6 – GET Condicional

- “**Cache Web**” .. reduz o tempo de resposta do ponto de vista do usuário, mas introduz um novo problema, ou seja, cópia de um objeto existente no “cache” pode estar desatualizada.
- “**solução**” .. HTTP dispõe de um mecanismo que permite que um “cache” verifique se seus objetos estão atualizados – “GET Condicional”
- GET é Condicional (requisição HTTP) se ..
  - (1) usar o método GET.
  - (2) possuir uma linha de cabeçalho “**if-modified-since**”.

## 2 – Camada de Aplicação / 2.2 World Wide Web e o HTTP

### ... 2.2.6 – GET Condicional

- e.g., seja a solicitação do “cache proxy” ao servidor em nome de um navegador requisitante (cliente usando o navegador).

GET /fruit/kiwi.gif HTTP/1.1

Host: www.exotiquecuisine.com

- Servidor Web envia ao cache uma mensagem de resposta 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

(dados dados dados dados dados ...)

## 2 – Camada de Aplicação / 2.2 World Wide Web e o HTTP

### ... 2.2.6 – GET Condicional

- Navegador realiza uma verificação de atualização emitindo um GET Condicional, ou seja, o cache envia ao servidor:

GET /fruit/kiwi.gif HTTP/1.1

Host: www.exotiquecuisine.com

If-modified-since: Wed, 7 Sep 2011 09:23:24

- GET Condicional com “**if-modified-since**” (linhas do cabeçalho) ...
- ... informa que o servidor deve enviar o objeto “/fruit/kiwi.gif” somente se ele tiver sido modificado desde a data especificada.

## 2 – Camada de Aplicação / 2.2 World Wide Web e o HTTP

### ... 2.2.6 – GET Condicional

- Resposta ao GET Condicional .. servidor envia uma mensagem de resposta, mas não inclui o objeto requisitado.

HTTP/1.1 304 Not Modified

Date: Sat, 15 Oct 2011 15:39:29

Server: Apache/1.3.0 (Unix)

... (corpo de mensagem vazio)

- .. se o fizesse, desperdiçaria largura de banda e aumentaria o tempo de resposta do ponto de vista do usuário.

## 2 – Camada de Aplicação / 2.2 World Wide Web e o HTTP

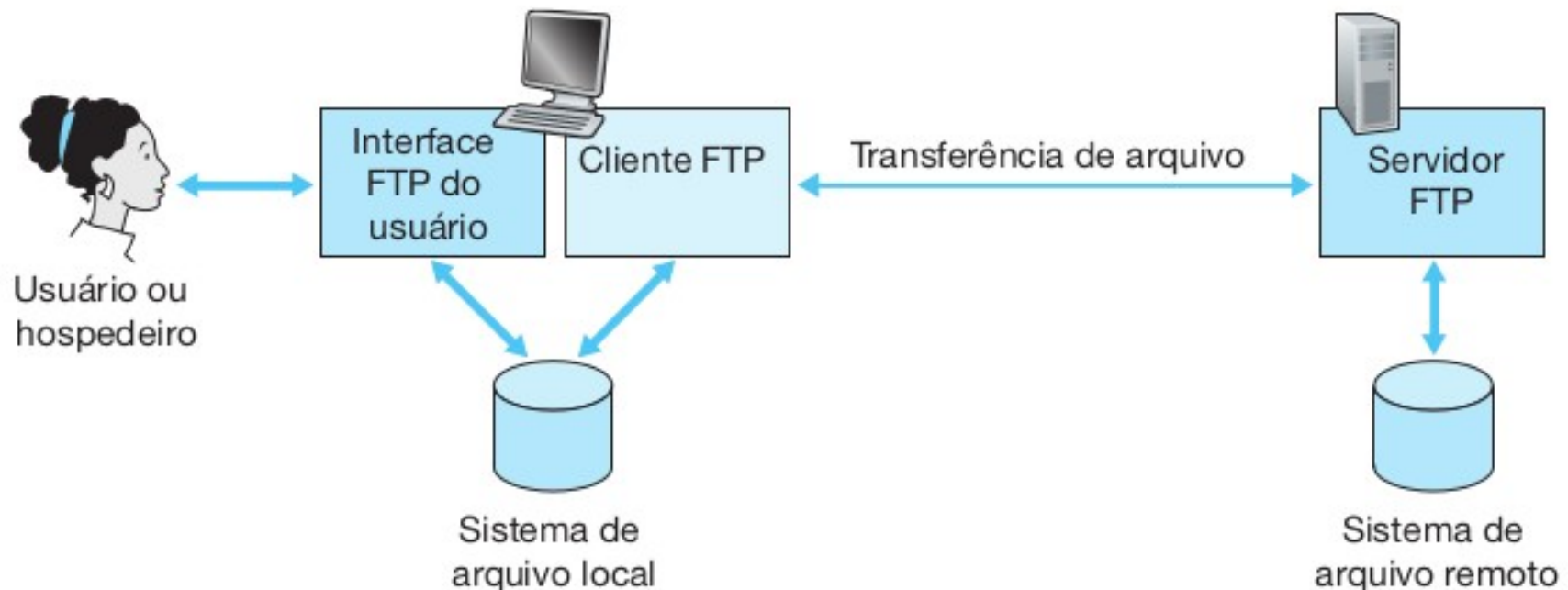
### ... 2.2.6 – GET Condicional

- “**resumo**” .. finalizamos nossa discussão sobre HTTP, o primeiro protocolo da Internet (um Protocolo da Camada de Aplicação);
- ... apresentamos o formato das mensagens HTTP e as ações tomadas pelo cliente e servidor quando essas msgs. são enviadas e recebidas.
- ... estudamos a infraestrutura da aplicação, incluindo caches, cookies e banco de dados de apoio, associados de algum modo ao HTTP.

## 2 – Camada de Aplicação / 2.3 – File Transport Protocol

### 2.3 - File Transport Protocol

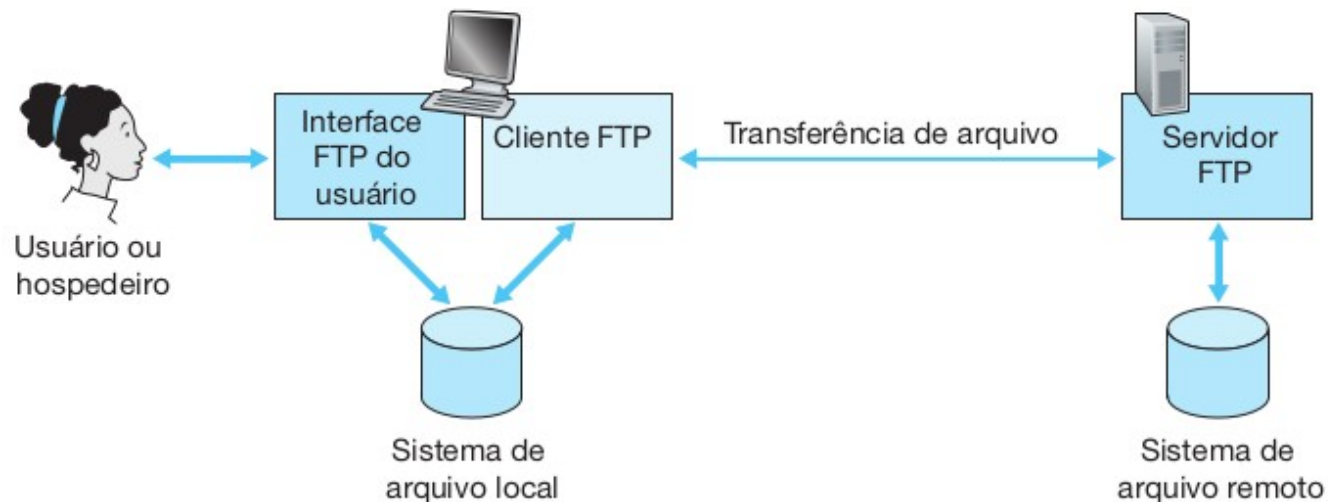
- Sessão FTP [RFC 959] .. usuário acessa uma conta remota para transferir arquivos entre o “host” local e o “host” remoto.
- .. após confirmada a autenticação, recebe a autorização para transferir arquivos do “host” local para o “host” remoto e vice-versa.



## 2 – Camada de Aplicação / 2.3 – File Transport Protocol

### ... 2.3 - File Transport Protocol

- Agente de Usuário .. forma pela qual o FTP interage com o Usuário.
  - 1) .. usuário fornece nome do “host” remoto, permitindo que o Cliente FTP estabeleça uma conexão com o “host” remoto.
  - 2) .. usuário fornece sua identificação e senha, que são enviadas pela conexão TCP como parte dos comandos FTP.
  - 3) .. uma vez autenticado e autorizado pelo Servidor FTP, o usuário copia 01 ou mais arquivos armazenados no sistema de arquivo local para o sistema de arquivo remoto (ou vice-versa).

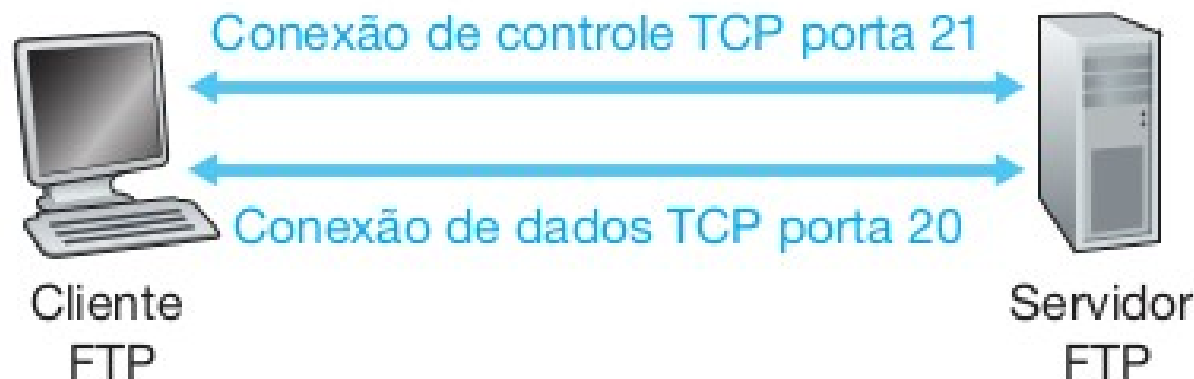




## 2 – Camada de Aplicação / 2.3 – File Transport Protocol

### ... 2.3 - File Transport Protocol

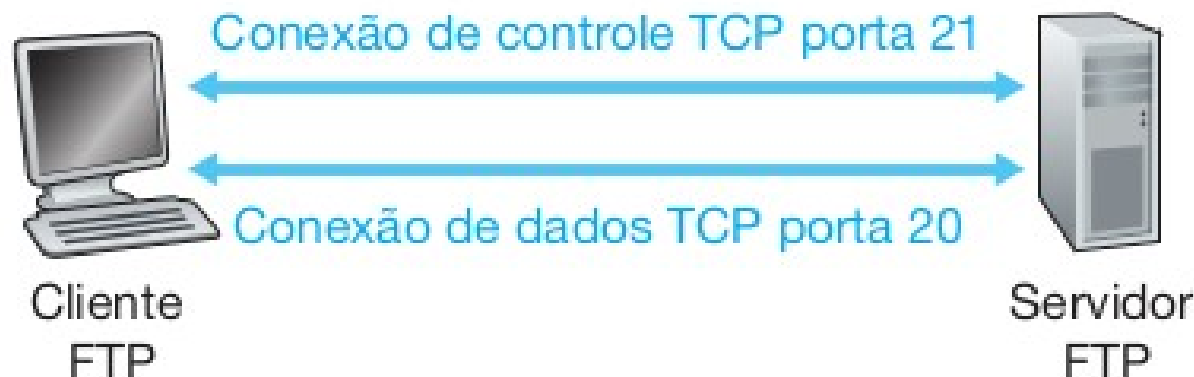
- HTTP e FTP são protocolos de transferência de arquivos e têm características em comum, p.ex., utilizam o TCP.
- ... mas tem diferenças, p.ex., FTP usa duas conexões TCP paralelas para transferir um arquivo - 01 conexão de controle e 01 de dados e, por isso, envia informações de controle “**fora da banda**”.
- “**observação**” .. HTTP envia linhas de cabeçalho de requisição e de resposta pela mesma conexão TCP que carrega o arquivo transferido, ou seja, HTTP envia informações de controle na **banda**.



## 2 – Camada de Aplicação / 2.3 – File Transport Protocol

### ... 2.3 - File Transport Protocol

- “**conexão de controle**” .. informações de controle entre os 02 “hosts” como identificação de usuário, senha, comandos para troca de diretório e comandos para “envio” (put) e “recepção” (get) de arquivos.
- ... conexão de controle separada = controle “**fora da banda**”.
- “**conexão de dados**” .. usada para enviar o arquivo de dados.



## 2 – Camada de Aplicação / 2.3 - File Transport Protocol

### 2.3.1 – “Requests” e “Replays” do FTP

- **“conexão de controle”** .. permite o envio de “requisições” do cliente para o servidor e os “replays” do servidor para o cliente em ASCII.
- .. ou seja, mensagens podem ser lidas pelos usuários.
- **“formato das mensagens”** .. comandos são enviados sucessivamente, assim, são utilizados os caracteres “carriage return” e “line feed” para encerramento de cada comando.
- ... cada comando é constituído de 04 caracteres ASCII maiúsculos, alguns contemplam argumentos opcionais.
- p.ex., USER iacopo
- p.ex., PASS faina

## 2 – Camada de Aplicação / 2.3 - File Transport Protocol

### ... 2.3.1 – “Requests” e “Replays” do FTP

- Comandos mais COMUNS no FTP:
- USER “username” .. identificação do usuário ao servidor.
- PASS “password” .. senha do usuário ao servidor.
- LIST .. solicitação de envio de uma lista com todos os arquivos existentes no atual diretório remoto.
- .. lista de arquivos é enviada por meio de uma conexão de dados não persistente e não pela conexão TCP de controle.
- RETR “filename” .. requisição para recuperação de um arquivo do diretório atual do “host” remoto, na sequência, conexão é aberta e o arquivo requisitado é enviado ao requisitante.
- STOR “filename” .. solicitação para inserção (armazenar) de um arquivo no diretório atual do “host” remoto.

## 2 – Camada de Aplicação / 2.3 - File Transport Protocol

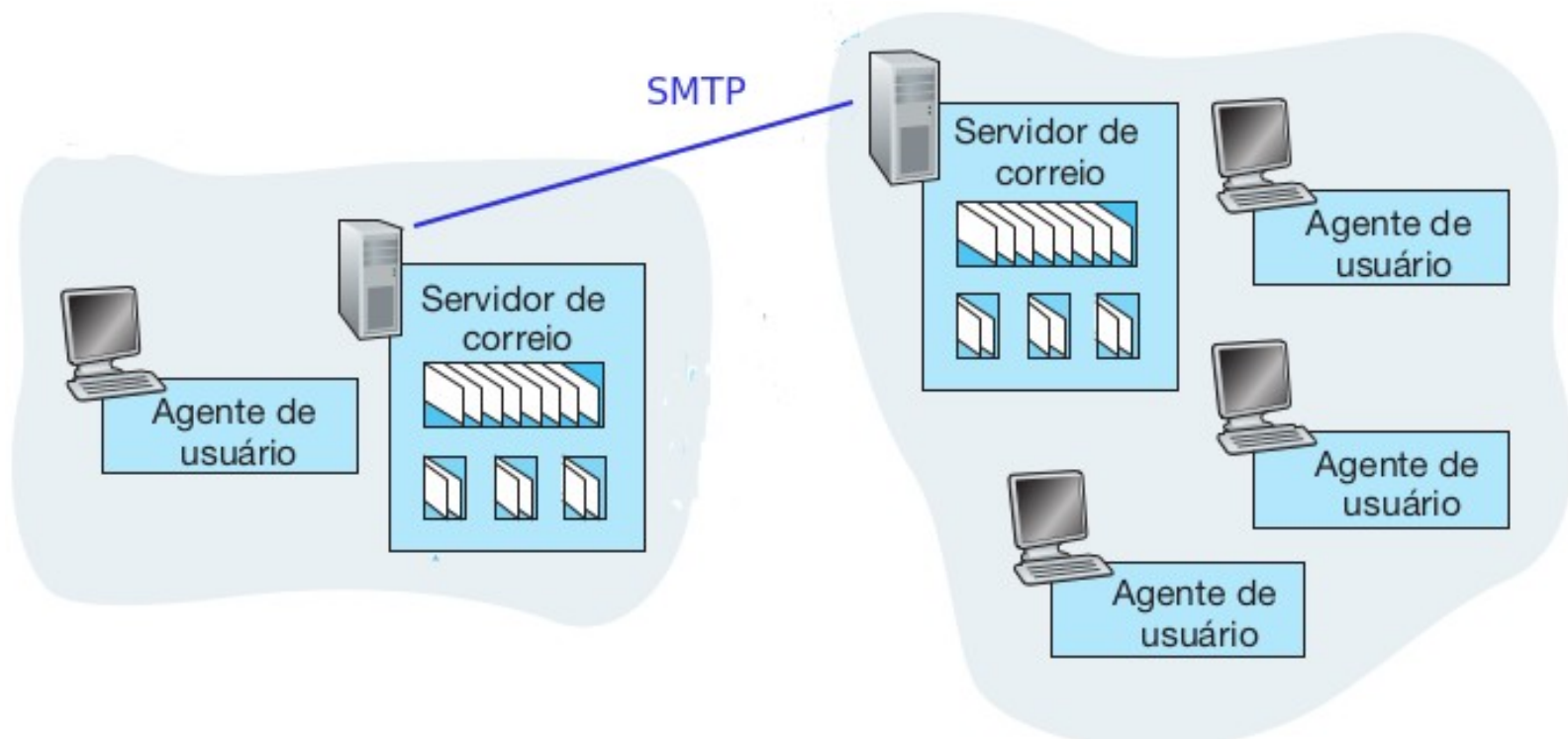
### ... 2.3.1 – “Requests” e “Replays” do FTP

- **“replays”** .. cada comando é seguido de uma resposta composta de 03 dígitos com uma mensagem opcional (servidor >> cliente).
- **“respostas típicas” + “possíveis mensagens”** ..
  - 331 .. nome de usuário OK, senha requisitada.
  - 125 .. conexão de dados já aberta; iniciando transferência.
  - 425 .. não é possível abrir a conexão de dados.
  - 452 .. erro ao escrever o arquivo.

## 2 – Camada de Aplicação / 2.4 – Correio Eletrônico (eMail)

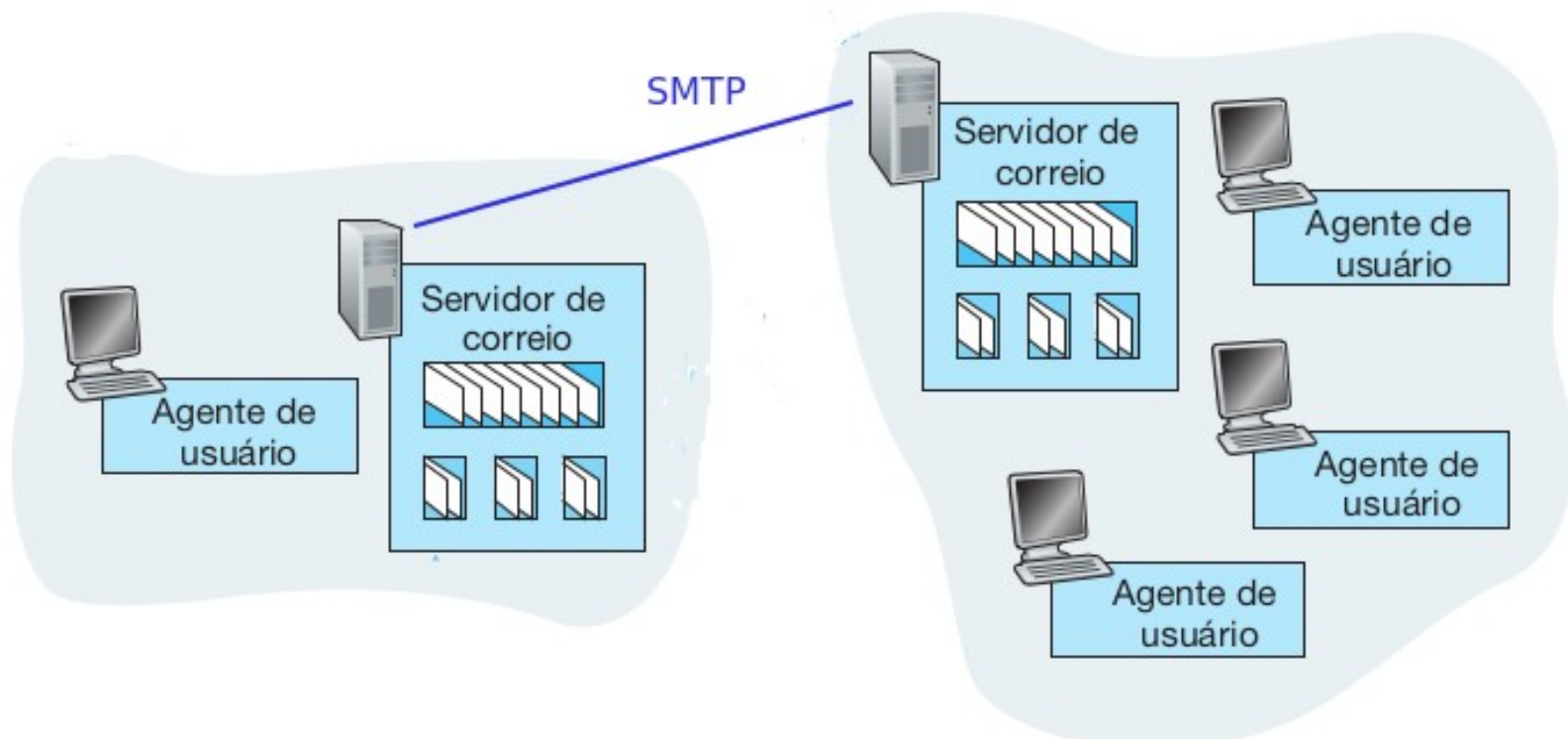
### 2.4 – Correio Eletrônico

- “**correio eletrônico**” .. meio de comunicação assíncrono no qual os usuários enviam e recebem mensagens quando da sua conveniência.
- “**sistema de eMail**” contempla 03 componentes principais: i) agentes de usuário; ii) servidores de correio; e iii) Protocolo SMTP.



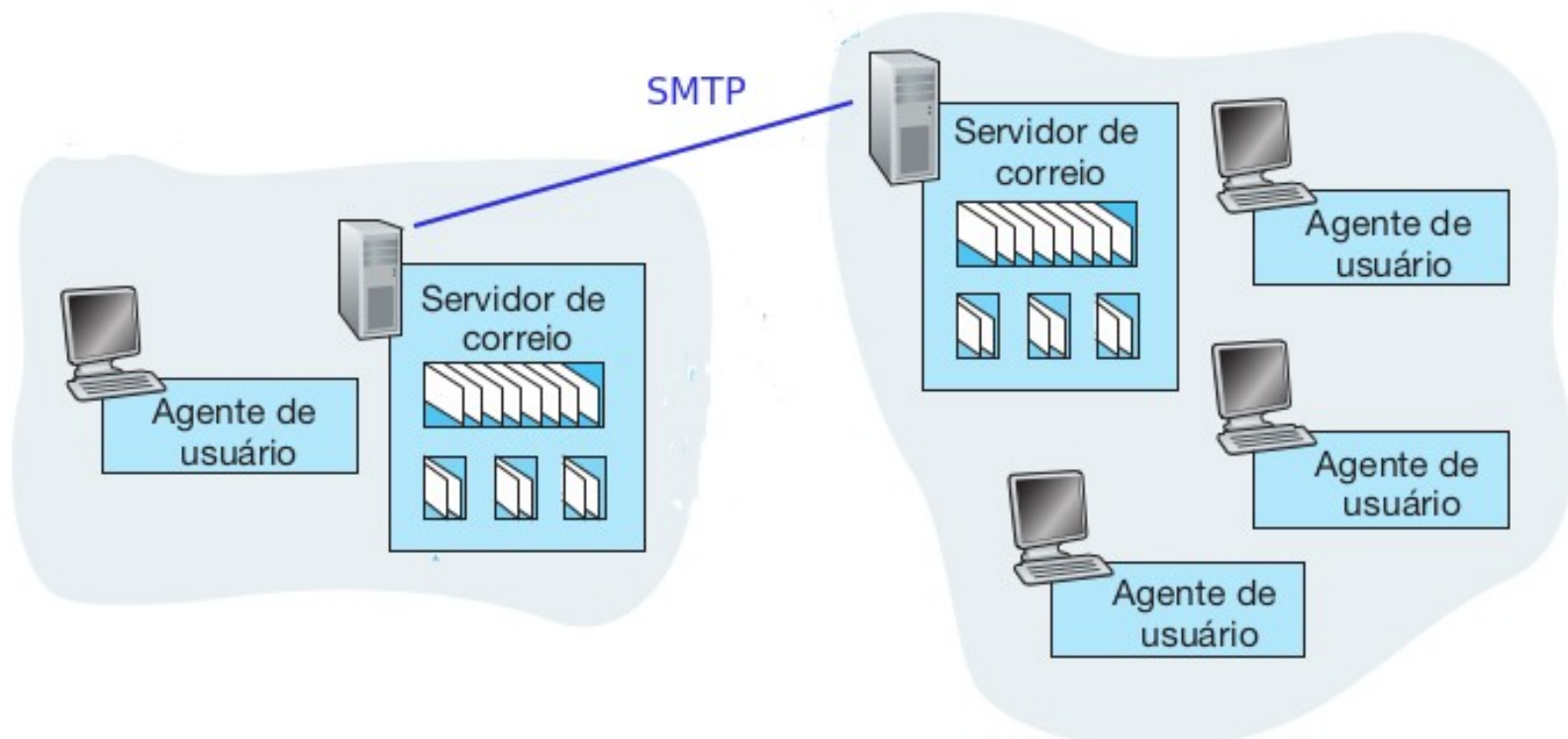
## 2 – Camada de Aplicação / 2.4 – Correio Eletrônico (eMail) ... 2.4 – Correio Eletrônico

- “**servidores de eMail**” formam o NÚCLEO da infraestrutura do eMail.
- ... se um servidor de eMail não puder entregar a correspondência de um remetente para um dado usuário, o servidor irá manter a msg. em uma fila de mensagens e tentará transferí-la mais tarde.



## 2 – Camada de Aplicação / 2.4 – Correio Eletrônico (eMail) ... 2.4 – Correio Eletrônico

- “**agentes de usuários**” .. ou clientes de eMail permitem que usuários leiam, respondam, encaminhem, salvem e componham mensagens.
- p.ex., Microsoft Outlook, Apple Mail, ThunderBird, etc.
- “**usuário**” .. cada usuário dispõe de uma caixa postal localizada em um desses servidores que administra e guarda as mensagens do mesmo.





## 2 – Camada de Aplicação / 2.4 – Correio Eletrônico (eMail)

### ... 2.4 – Correio Eletrônico

- SMTP [RFC 5321] .. principal protocolo de camada de aplicação do correio eletrônico da Internet e utiliza transf. confiável de dados.
- SMTP é Cliente/Servidor .. lado cliente (servidor de email do remetente) e um lado servidor (servidor de email do destinatário).
- ... ambos funcionam em todos os servidores de correio.
- “**perscepção**” .. quando um servidor de email envia mensagens de emails para outros, age como um cliente SMTP.
- ... quando o servidor de email recebe mensagens de emails de outros domínios, age como um servidor SMTP.

## 2 – Camada de Aplicação / 2.4 – Correio Eletrônico (eMail)

### 2.4.1 – Sample Mail Transport Protocol

- SMTP [RFC 5321] .. transfere mensagens de servidores de eMail remetentes para servidores de eMail destinatários.
- SMTP restringe o corpo (e não apenas o cabeçalho) de todas as mensagens de correio ao simples formato ASCII de 7 bits.
- ... restrição é um tanto incômoda, exigindo que os dados binários de multimídia sejam codificados em ASCII antes de serem enviados.
- “**solução**” .. estender o formato de protocolo de transferência de correio simples (SMTP) para inserir diferentes tipos de conteúdo.

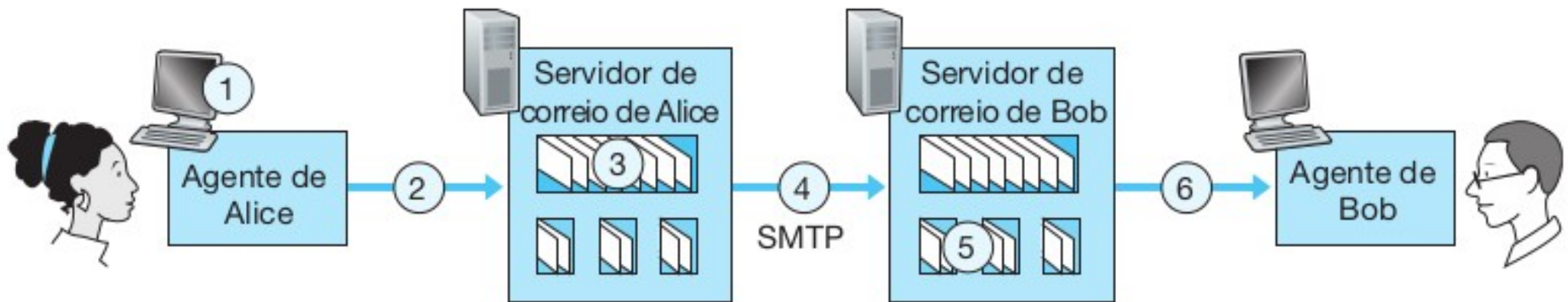
## 2 – Camada de Aplicação / 2.4 – Correio Eletrônico (eMail)

### ... 2.4.1 – Sample Mail Transport Protocol

- MIME (Multipurpose Internet Mail Extensions) .. protocolo definido para permitir a inclusão de dados não-ASCII via e-mail.
- ... atua convertendo os dados não-ASCII inseridos pelo usuário, para dados ASCII, de forma que possam ser transmitidos pelo SMTP.
- ... no destinatário é necessário que a msg. em ASCII seja decodificada para o sistema binário depois do transporte pelo SMTP.
- ... como muitas das mensagens possuem uma associação bem próxima entre os padrões SMTP e MIME, é comum que tais mensagens sejam denominadas de mensagens SMTP/MIME.

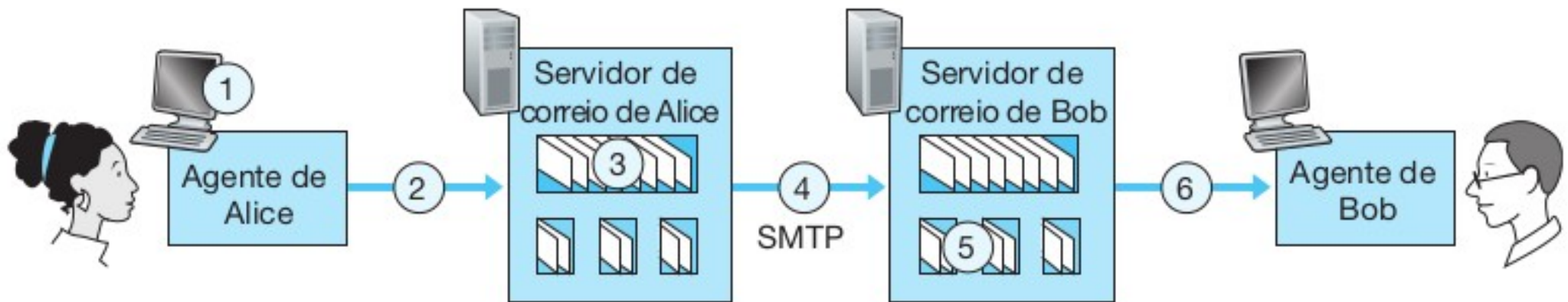
## 2 – Camada de Aplicação / 2.4 – Correio Eletrônico (eMail) ... 2.4.1 – Simple Mail Transport Protocol

- e.g., Operação Básica do SMTP .. seja um cenário em que Alice envia uma mensagem ASCII para Bob.
- 1) Alice chama seu agente de usuário de e-mail, fornece o endereço de Bob (p.ex., bob@some school.edu ), compõe uma mensagem e instrui o agente de usuário para enviá-la.
- 2) .. agente de usuário de Alice envia a mensagem para seu servidor de correio, onde é inserida em uma fila de mensagens.



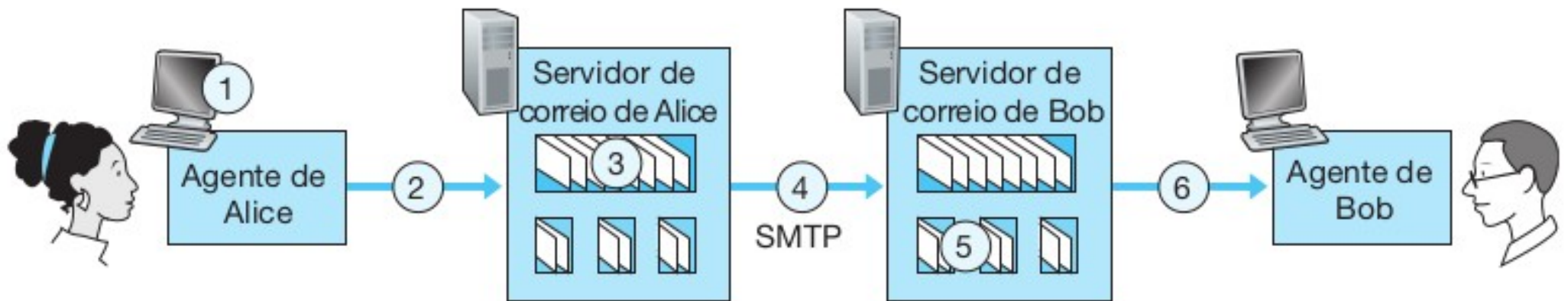
## 2 – Camada de Aplicação / 2.4 – Correio Eletrônico (eMail) ... 2.4.1 – Simple Mail Transport Protocol

- e.g., Operação Básica do SMTP .. seja um cenário em que Alice envia uma mensagem ASCII para Bob.
- 3) Cliente do SMTP (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 procedimentos iniciais de apresentação ou “handshaking”, o cliente SMTP envia a mensagem de Alice pela conexão TCP.



## 2 – Camada de Aplicação / 2.4 – Correio Eletrônico (eMail) ... 2.4.1 – Simple Mail Transport Protocol

- e.g., Operação Básica do SMTP .. seja um cenário em que Alice envia uma mensagem ASCII para Bob.
- 5) .. no servidor de correio de Bob, o lado servidor do SMTP recebe a mensagem e a coloca na caixa postal de Bob.
- 6) Bob chama seu agente de usuário para ler a mensagem quando for mais conveniente para ele.



## 2 – Camada de Aplicação / 2.4 – Correio Eletrônico (eMail)

### ... 2.4.1 – Simple Mail Transport Protocol

- e.g., considere a troca de mensagens ASCII que compõem uma mensagem de e-mail entre um Cliente (C) cujo “hostname” é “crepes.fr” e um Servidor (S) SMTP cujo “hostname” é “hamburger.edu”.
- 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
- ... ..

## 2 – Camada de Aplicação / 2.4 – Correio Eletrônico (eMail)

### ... 2.4.1 – Simple Mail Transport Protocol

- e.g., considere a troca de mensagens ASCII que compõem uma mensagem de e-mail entre um Cliente (C) cujo “hostname” é “crepes.fr” e um Servidor (S) SMTP cujo “hostname” é “hamburger.edu”.
- ... ..
- 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



## 2 – Camada de Aplicação / 2.4 – Correio Eletrônico (eMail)

### ... 2.4.1 – Simple Mail Transport Protocol

- Obs.: ... cliente também envia uma linha que contém um único “.” “**ponto final**”, que indica o final da mensagem para o servidor.
- ... jargão ASCII, cada mensagem termina com “**CRLF . CRLF**”, cujo significado é “**carriage return**” e “**line feed**” respectivamente.
- “**exercício**” .. utilize o Telnet para executar um diálogo direto com um servidor SMTP e, neste sentido, digite “telnet servername 25”
- ... este procedimento estabelece uma conexão TCP entre o “host” local e o servidor de correio e, deverá obter como resposta o código 220.
- .. na sequência, digite os comandos “HELO”, “MAIL FROM”, “RCPT TO”, “DATA”, CRLF, mas CRLF e “QUIT” nos momentos apropriados.

## 2 – Camada de Aplicação / 2.4 – Correio Eletrônico (eMail)

### 2.4.2 – Comparação com HTTP

- “**diferença**” .. finalidade de uso do HTTP e SMTP.
- SMTP / HTTP .. ambos os protocolos são usados para transferir arquivos de um “host” para outro .. “**então o que muda**” ??
- HTTP transfere arquivos (objetos) de um servidor para um cliente Web.
- SMTP transfere arquivos (emails) de um servidor de correio para outro.
- Obs.: HTTP Persistente e SMTP usam conexões persistentes.

## 2 – Camada de Aplicação / 2.4 – Correio Eletrônico (eMail)

### ... 2.4.2 – Comparação com HTTP

- “**diferença**” .. “push protocol” ou “pull protocol”
- HTTP .. protocolo de recuperação de informações (“pull protocol”), ou seja, alguém carrega informações em um servidor Web e os usuários utilizam o HTTP para recuperá-las quando quiserem.
- ... conexão TCP é ativada pelo “host” que quer receber o arquivo.
- SMTP .. protocolo de envio de informações ou “push protocol”, ou seja, o servidor de email remetente envia o arquivo para o serv. destinatário.
- ... conexão TCP é ativada pela máquina que quer enviar o arquivo.

## 2 – Camada de Aplicação / 2.4 – Correio Eletrônico (eMail)

### ... 2.4.2 – Comparação com HTTP

- “**diferença**” .. codificação da mensagem quanto ao conteúdo.
- SMTP .. exige que cada mensagem, inclusive o corpo da mensagem, esteja no formato ASCII de 7 bits (8 bits - 8º bit = zero).
- ... se contiver caracteres que não estejam nesse formato (p.ex., caracteres acentuados em francês) ou dados binários (p.ex., imagem), terá de ser codificada em ASCII de 7 bits.
- HTTP ... não impõem esta restrição.

## 2 – Camada de Aplicação / 2.4 – Correio Eletrônico (eMail)

### ... 2.4.2 – Comparação com HTTP

- “**diferença**” .. modo como um documento que contém texto e imagem (juntamente com outros tipos possíveis de mídia) é manipulado.
- HTTP .. encapsula cada objeto em sua própria mensagem HTTP.
- SMTP .. todos os objetos de mensagem em uma única mensagem.

## 2 – Camada de Aplicação / 2.4 – Correio Eletrônico (eMail)

### 2.4.3 – Formato de Mensagens de Correio

- RFC 5322 ... especifica o formato exato das linhas de cabeçalho das mensagens, bem como suas interpretações semânticas.
- .. palavras obrigatórias .. p.ex., “From: xx”, “To: xx”, “Subject: zzzz”.
- Cabeçalho de Mensagem típico .. após o cabeçalho da msg., vem uma linha em branco e, em seguida, o corpo da mensagem (em ASCII).

From: alice@crepes.fr

To: bob@hamburger.edu

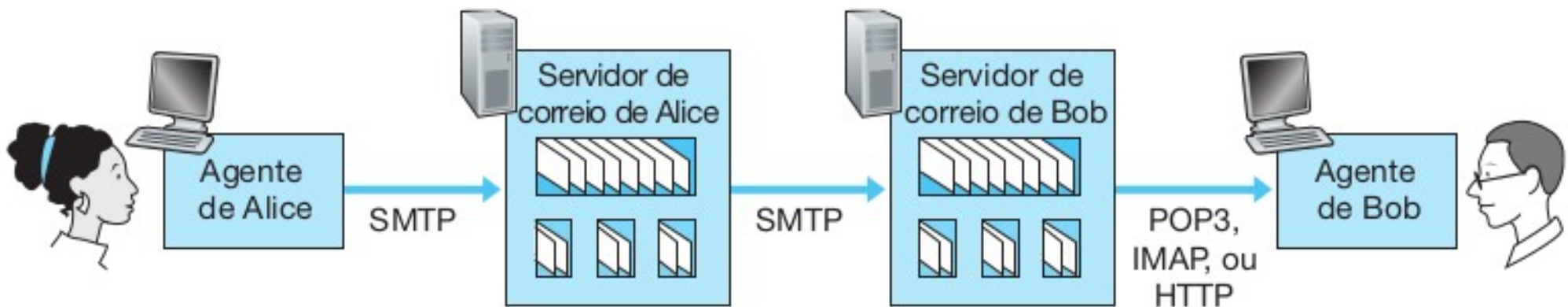
Subject: Searching for the meaning of life.

- e.g. .. utilize o Telnet “telnet servername 25” para enviar a um servidor de correio (p.ex., o seu servidor) uma mensagem que contenha algumas linhas de cabeçalho, inclusive “Subject: “.

## 2 – Camada de Aplicação / 2.4 – Correio Eletrônico (eMail)

### 2.4.4 – Protocolos de Acesso ao Correio

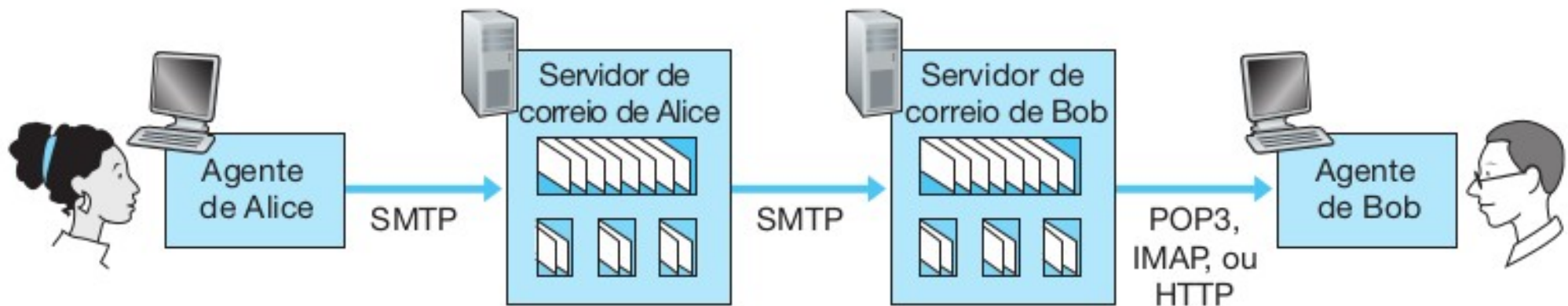
- **“acesso aos e-mails”** .. usuário típico lê eMails com um cliente que funciona em seu sistema final, p.ex., notebook ou smartphone.
- .. agente de usuário do remetente usa o protocolo SMTP para enviar um e-mail ao servidor de e-mail do seu domínio de eMail.
- .. Cliente SMTP no Servidor eMail do Remetente encaminha a msg. para o Servidor SMTP do Servidor de eMail do Destinatário.



## 2 – Camada de Aplicação / 2.4 – Correio Eletrônico (eMail)

### ... 2.4.4 – Protocolos de Acesso ao Correio

- e.g., .. de que forma um destinatário como Bob, que executa um agente de usuário em seu desktop local, obtém suas mensagens que estão em um servidor de correio dentro do seu ISP?
- Obs.: .. note que o agente de usuário de Bob não pode usar SMTP para obter as mensagens porque essa operação é de recuperação (pull protocol), e o SMTP é um protocolo de envio (push protocol).

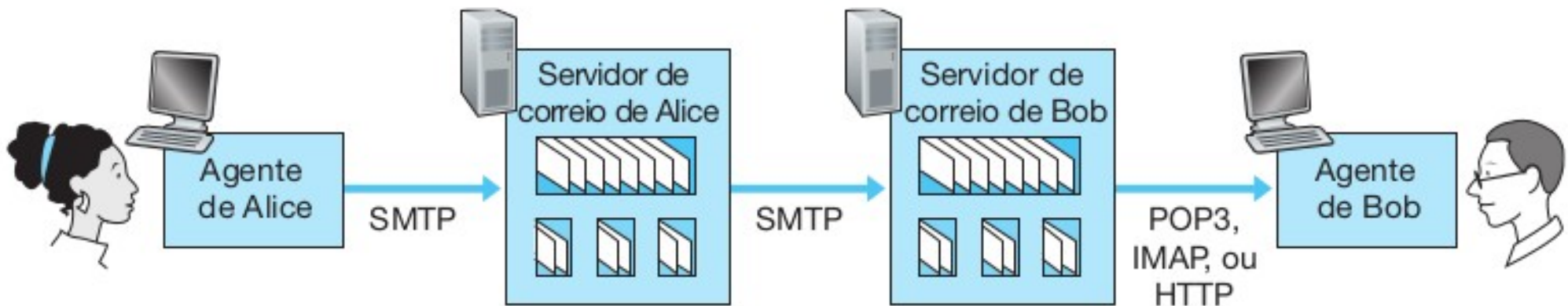




## 2 – Camada de Aplicação / 2.4 – Correio Eletrônico (eMail)

### ... 2.4.4 – Protocolos de Acesso ao Correio

- “**solução**” .. uso de um protocolo especial de acesso ao correio que transfere mensagens do servidor de correio de Bob para seu desktop.
- ... há vários protocolos populares de acesso a correio:
- POP3 (Post Office Protocol v3);
- IMAP (Internet Mail Access Protocol);
- HTTP (HyperText Transport Protocol).



## 2 – Camada de Aplicação / 2.4 – Correio Eletrônico (eMail)

### ... 2.4.4 – Protocolos de Acesso ao Correio

- POP3 [RFC 1939] .. protocolo de acesso de correio de extrema simplicidade, curto e bem fácil de ler, mas limitado.
- Agente de Usuário do POP3 (Usuário do Desktop - Cliente) abre uma conexão TCP com o Servidor de eMail (Servidor) na porta 110.
- Protocolo POP3 com 03 fases: autorização, transação e atualização.
- “**autorização**” .. agente de usuário envia um nome de usuário e uma senha (às claras) para autenticar o usuário.
- “**transação**” .. possibilita a recuperação de mensagens bem como a seleção de mensagens para serem removidas.
- “**atualização**” .. dá início após emissão do comando “quit” que encerra a sessão POP3, permitindo, p.ex., que se apague as msgs. marcadas.

## 2 – Camada de Aplicação / 2.4 – Correio Eletrônico (eMail)

### ... 2.4.4 – Protocolos de Acesso ao Correio

- **“troca de mensagens pelo protocolo”** .. agente de usuário emite comandos e o servidor emite uma resposta para cada um dos comandos (02 respostas possíveis):
- +OK .. as vezes seguida de dados do servidor para o cliente e usada pelo servidor para indicar que o comando foi executado corret/te.
- -ERR .. algo errado na execução do comando anterior.

## 2 – Camada de Aplicação / 2.4 – Correio Eletrônico (eMail)

### ... 2.4.4 – Protocolos de Acesso ao Correio

- **“autorização”** .. 02 comandos: user <username> e pass <password> .
- e.g., sugere-se que se realize uma sessão telnet “telnet mailserver 110” com um servidor POP3 e emita os 02 comandos.
- ... suponha que mailServer seja o nome de seu servidor de correio.  
telnet mailServer 110  
+OK POP3 server ready  
user bob  
+OK  
pass hungry  
+OK user successfully logged on
- **“observação”** .. ao escrever um comando errado, o POP3 responde com uma mensagem –ERR.

## 2 – Camada de Aplicação / 2.4 – Correio Eletrônico (eMail)

### ... 2.4.4 – Protocolos de Acesso ao Correio

- **“transação”** .. agente de usuário que utiliza POP3 pode ser configurado (pelo usuário) para “ler-e-apagar” ou para “ler-e-guardar”.
- ... sequência de comandos emitida por um agente de usuário POP3 depende do modo em que o agente de usuário estiver operando.
- **“problema no modo ler-e-apagar”** .. destinatário (Bob) pode ser nômade e querer acessar seu correio de muitas máquinas, p.ex., do desktop de seu escritório, do desktop de sua casa e de seu notebook.
- ... este modo remete as mensagens de correio entre as 03 máquinas; em particular, se ele ler primeiro uma mensagem no desktop de seu escritório, não poderá lê-la de novo mais tarde em seu notebook..

## 2 – Camada de Aplicação / 2.4 – Correio Eletrônico (eMail)

### ... 2.4.4 – Protocolos de Acesso ao Correio

- **“transação”** .. agente de usuário que utiliza POP3 pode ser configurado (pelo usuário) para “ler-e-apagar” ou para “ler-e-guardar”.
- ... sequência de comandos emitida por um agente de usuário POP3 depende do modo em que o agente de usuário estiver operando.
- **“modo ler-e-guardar”** .. agente de usuário deixa as mensagens no servidor de correio após descarregá-las.
- ... nesse caso, Bob pode reler mensagens em máquinas diferentes; pode acessar uma mensagem em seu local de trabalho e, uma semana depois, acessá-la novamente em casa.

## 2 – Camada de Aplicação / 2.4 – Correio Eletrônico (eMail)

### ... 2.4.4 – Protocolos de Acesso ao Correio

- e.g., suponha que o usuário tenha duas mensagens em sua caixa postal. .. legenda C: (cliente) é o agente de usuário e S: (servidor), o servidor de correio e a transação será mais ou menos assim:

```
C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: (blah blah ...
S: .....
S: .....blah)
... ..
```

## 2 – Camada de Aplicação / 2.4 – Correio Eletrônico (eMail)

### ... 2.4.4 – Protocolos de Acesso ao Correio

- ... sequência de comandos emitida por um agente de usuário POP3 depende do modo em que o agente de usuário estiver operando.

... ..

C: dele 1

C: retr 2

S: (blah blah ...

S: .....

S: .....blah)

S: .

C: dele 2

C: quit

S: +OK POP3 server signing off



## 2 – Camada de Aplicação / 2.4 – Correio Eletrônico (eMail)

### ... 2.4.4 – Protocolos de Acesso ao Correio

- Note que após a fase de “autorização” o agente de usuário empregou alguns comandos: “list”, “retr”, “dele” e “quit”, cuja sintaxe RFC 1939.
- ... após processar o comando de saída (quit), o servidor POP3 entra na fase de atualização e remove as mensagens 1 e 2 da caixa postal.
- **“informação de estado”** .. durante uma sessão, o servidor mantém alguma informação de estado; p.ex., msgs. marcadas para apagar.
- ... mas mantém informação de estado entre sessões POP3.
- Essa falta de informação simplifica a execução de um Servidor POP3.

## 2 – Camada de Aplicação / 2.4 – Correio Eletrônico (eMail)

### ... 2.4.4 – Protocolos de Acesso ao Correio

- POP3 .. usuário pode fazer o “download” de mensagens “in-loco”, criar pastas de correspondência e até armazená-las nas pastas.
- .. na sequência pode transferir mensagens para as pastas criadas, movê-las entre as pastas, removê-las ou procurar por alguma msg.
- “**problemas**” .. hierarquia de pastas está no “desktop” e não no servidor de e-mails, dificultando ou exigindo a estrutura em cada dispositivo a partir do qual o usuário acessa seus e-mails.
- Protocolo POP3 não provê nenhum meio para um usuário criar pastas remotas e designar mensagens a pastas.

## 2 – Camada de Aplicação / 2.4 – Correio Eletrônico (eMail)

### ... 2.4.4 – Protocolos de Acesso ao Correio

- IMAP [RFC 3501] .. protocolo de acesso a emails, porém com mais recursos e mais complexo associa cada mensagem a uma pasta.
- .. quando uma mensagem chega a um servidor pela 1ª vez, a mesma é associada com a pasta INBOX do destinatário, que, então, pode transferí-la para uma nova pasta, lê-la, apagá-la e assim por diante.
- IMAP (Internet Message Access Protocol) provê comandos para que os usuários criem pastas e transfiram mensagens entre as pastas.
- Servidor IMAP mantém informação de estado de usuário entre sessões IMAP, p.ex., os nomes das pastas e quais mensagens estão associadas a elas .. diferentemente do POP3.

## 2 – Camada de Aplicação / 2.4 – Correio Eletrônico (eMail)

### ... 2.4.4 – Protocolos de Acesso ao Correio

- “**característica importante**” .. contempla comandos que permitem ao agente de usuário obter componentes de mensagens.
- p.ex., agente de usuário pode obter apenas o cabeçalho ou somente uma das partes de uma mensagem MIME (Multipurpose Internet Mail Extensions), posto que a msg. contempla múltiplas partes.
- “**contexto**” .. em uma conexão com pequena largura de banda, o usuário pode decidir não baixar todas as mensagens de sua caixa postal, evitando, p.ex., mensagens longas que possam conter (vídeo).

## 2 – Camada de Aplicação / 2.4 – Correio Eletrônico (eMail)

### ... 2.4.4 – Protocolos de Acesso ao Correio

- **“eMails pelo Navegador HTTP”** .. uso cada vez mais frequente do navegador para acesso, leitura, remoção e envio de eMails.
- p.ex., Hotmail lançou o e-mail com acesso pela Web em meados da década de 1990, mas atualmente, acesso também é fornecido por outros como Google, Yahoo, Microsoft, etc.
- **“Serv. de eMail pelo Navegador”** .. agente de usuário = navegador e o usuário se comunica com sua caixa postal remota via HTTP.

## 2 – Camada de Aplicação / 2.4 – Correio Eletrônico (eMail)

### ... 2.4.4 – Protocolos de Acesso ao Correio

- **“leitura de email”** .. quando um usuário acessa uma mensagem em sua caixa postal, a msg. é enviada do servidor de email para o navegador do usuário usando o Protocolo HTTP e não POP3 ou IMAP.
- **“envio de email”** .. mensagem é enviada para o Servidor de eMail do solicitante por HTTP e não por SMTP.
- Obs.: Servidor de eMail do Usuário continua a enviar mensagens para outros Servidores de eMail usando Protocolo SMTP.

## 2 – Camada de Aplicação / 2.5 – Domain Name System

### 2.5 – Domain Name System (DNS)

- **“analogia”** .. assim como seres humanos podem ser identificados de muitas maneiras, o mesmo acontece com “hosts” na Rede Internet.
- ... identificador é o nome do “host” ou “hostname” e, pode ser utilizado para identificá-lo univocamente na Rede Internet.
- ... nomes de “hosts” como “www.google.com”, “www.yahoo.com”, “gaia.cs.umass.edu” e “cis.poly.edu” são fáceis de lembrar;
- ... no entanto, fornecem pouca ou nenhuma informação sobre a localização do “host” na Internet, ou seja, localização física.
- ... nomes de “hosts” podem conter caracteres alfanuméricos de comprimento variável, o que é difícil de se processar em roteadores.
- **“solução”** .. “hosts” devem ser identificados pelo seus endereços IPs.

## 2 – Camada de Aplicação / 2.5 – Domain Name System

### ... 2.5 – Domain Name System (DNS)

- “**endereço IPv4**” .. identificador com 32 bits que reflete uma estrutura hierárquica normalmente representada na notação decimal (IPv4).
- p.ex., notação decimal para cada conjunto de 08 bits separado por “.”, ou seja, decimal de 0 a 255, p.ex., “121.7.106.83”; “200.131.225.49”.
- “**hierárquico**” ... ao examiná-lo da esquerda para a direita, obtém-se gradativamente mais informações específicas sobre em qual rede o “host” está localizado na Rede Internet (Redes das Redes).



## 2 – Camada de Aplicação / 2.5 – Domain Name System

### 2.5.1 - Serviços fornecidos pelo DNS

- Domain Name System [RFC 1034 e 1035] contempla:
- ... banco de dados distribuído em servidores de domínio, servidores estes organizados hierarquicamente em forma de árvore.
- ... protocolo de camada de aplicação que permite que “hosts” consultem o banco de dados distribuído.
- Servidores DNS muitas vezes executam o BIND (Berkeley Internet Name Domain) em máquinas UNIX usando o UDP na Porta 53.
- “**aplicação**” .. utilizado por outros protocolos como HTTP, SMTP e FTP para traduzir nomes de “hosts” em URLs para endereços IP.

## 2 – Camada de Aplicação / 2.5 – Domain Name System

### ... 2.5.1 - Serviços fornecidos pelo DNS

- e.g., seja a execução da URL “www.someschool.edu/index.html” no navegador (Cliente HTTP) da máquina de algum usuário.
  - 1) .. máquina do usuário executa o lado Cliente da Aplicação DNS, ou seja, solicitação parte do navegador utilizado pelo usuário.
  - 2) .. navegador extrai o nome de “host” - “www.someschool.edu” da URL e passa o nome para o lado Cliente da Aplicação DNS.
  - 3) .. Cliente DNS envia uma consulta para um Servidor DNS contendo o nome do hospedeiro - “www.someschool.edu”... continuação ..

## 2 – Camada de Aplicação / 2.5 – Domain Name System

### ... 2.5.1 - Serviços fornecidos pelo DNS

- e.g., execução no navegador (Cliente HTTP) da máquina de algum usuário URL “www.someschool.edu/index.html”.

.. continuação ..

4) Cliente DNS (navegador) por fim recebe uma resposta, que inclui o endereço IP correspondente ao nome de “host”.

5) Navegador recebe o endereço IP do DNS e, então, pode abrir uma conexão TCP com o Processo Servidor HTTP na Porta 80 do “host” IP.

- “**observação**” .. DNS adiciona mais um atraso - às vezes substancial às aplicações de Internet que o utilizam (p.ex., HTTP, SMTP e FTP).

## 2 – Camada de Aplicação / 2.5 – Domain Name System

### ... 2.5.1 - Serviços fornecidos pelo DNS

- Serviços providos pelo DNS (Domain Name System):
- **“host aliasing”** .. “host” com nome longo pode ter um ou mais apelidos, p.ex., “relay1.west-coast.enterprise.com” pode ter dois apelidos, como “enterprise.com” e “www.enterprise.com”.
- ... nesse caso, o nome de “host” **“relay1.west-coast.enterprise.com”** é denominado **nome canônico** ou **“canonical name”**.
- Obs.: .. ao ser chamado por uma aplicação, o DNS pode fornecer o **“canonical name”**, bem como o seu endereço IP do “host”.

## 2 – Camada de Aplicação / 2.5 – Domain Name System

### ... 2.5.1 - Serviços fornecidos pelo DNS

- Serviços providos pelo DNS (Domain Name System):
- “**mail alias**” .. ao ser chamado por uma aplicação de e-Mail para obter o nome canônico a partir de um apelido fornecido, o DNS deve fornecer o endereço IP do “host” servidor de domínio de e-Mail.
- ... trata-se do Registro MX que permite ao servidor de e-Mail e o servidor Web de uma empresa terem nomes (apelidos).
- ... p.ex., Servidor Web e o Servidor de e-Mail de uma empresa podem em conjunto serem identificados por “enterprise.com”.

## 2 – Camada de Aplicação / 2.5 – Domain Name System

### ... 2.5.1 - Serviços fornecidos pelo DNS

- Serviços providos pelo DNS (Domain Name System):
- **“balanceamento de carga”** .. possibilita a distribuição de carga entre servidores replicados, tais como os servidores Web replicados.
- p.ex., no caso de servidores Web replicados, um conjunto de endereços IP fica associado a um único “nome canônico” no DNS.
- DNS [RFC 1034; RFC 1035] .. mencionamos, por ora, os aspectos fundamentais de sua operação, no entanto, o DNS é atualizado em diversos outras RFCs e não somente nas RFCs 1034 e 1035.

## 2 – Camada de Aplicação / 2.5 – Domain Name System

### 2.5.2 - Visão Geral do DNS

- “**como traduzir um “hostname” para IP” ??**
- p.ex, suponha que um navegador Web ou um leitor de e-Mail necessite traduzir um nome de “host” para um endereço IP.
- ... aplicação chama o lado Cliente do DNS, especificando o nome de “host” que precisa ser traduzido.
- ... em muitas máquinas UNIX, “**gethostbyname(..)**” é a chamada de função que uma aplicação invoca para realizar a tradução.
- ... do ponto de vista da aplicação (máquina do cliente), o DNS é uma caixa-preta que provê um **serviço de tradução simples e direto**.

## 2 – Camada de Aplicação / 2.5 – Domain Name System

### ... 2.5.2 - Visão Geral do DNS

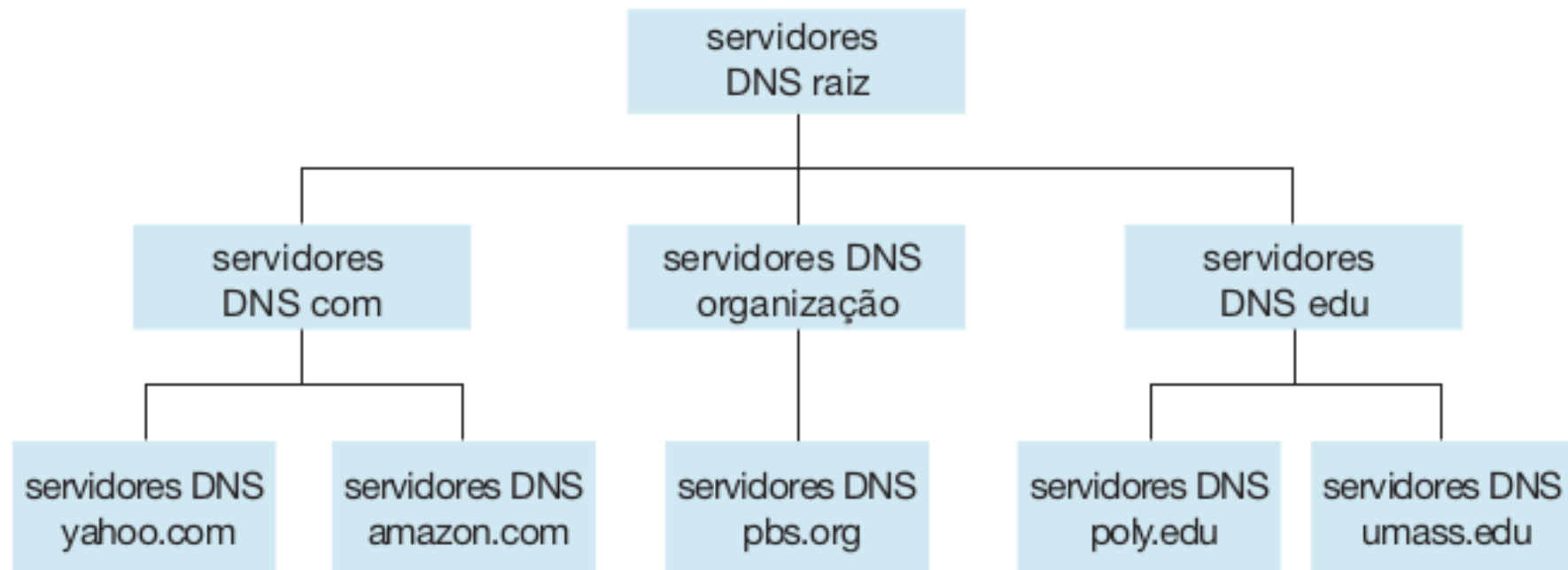
- **“proposta de solução”** .. ter um único servidor DNS contendo todos os mapeamentos de “hostnames” e seus respectivos IPs.
- **“ponto de falha”** - se o servidor quebrar, a Internet inteira quebra!
- **“volume de tráfego”** .. único servidor DNS tem de manipular todas as requisições HTTP e de e-Mail geradas por 1000s de 1000s de “hosts”.
- **“banco de dados distante”** .. um único servidor DNS está próximo de 100s de clientes e distante de 1000s de 1000s de outros clientes.
- **“manutenção”** .. único servidor DNS deve manter registros de todos os “hosts” da Internet, algo impensável ou inimaginável.



## 2 – Camada de Aplicação / 2.5 – Domain Name System

### ... 2.5.2 - Visão Geral do DNS

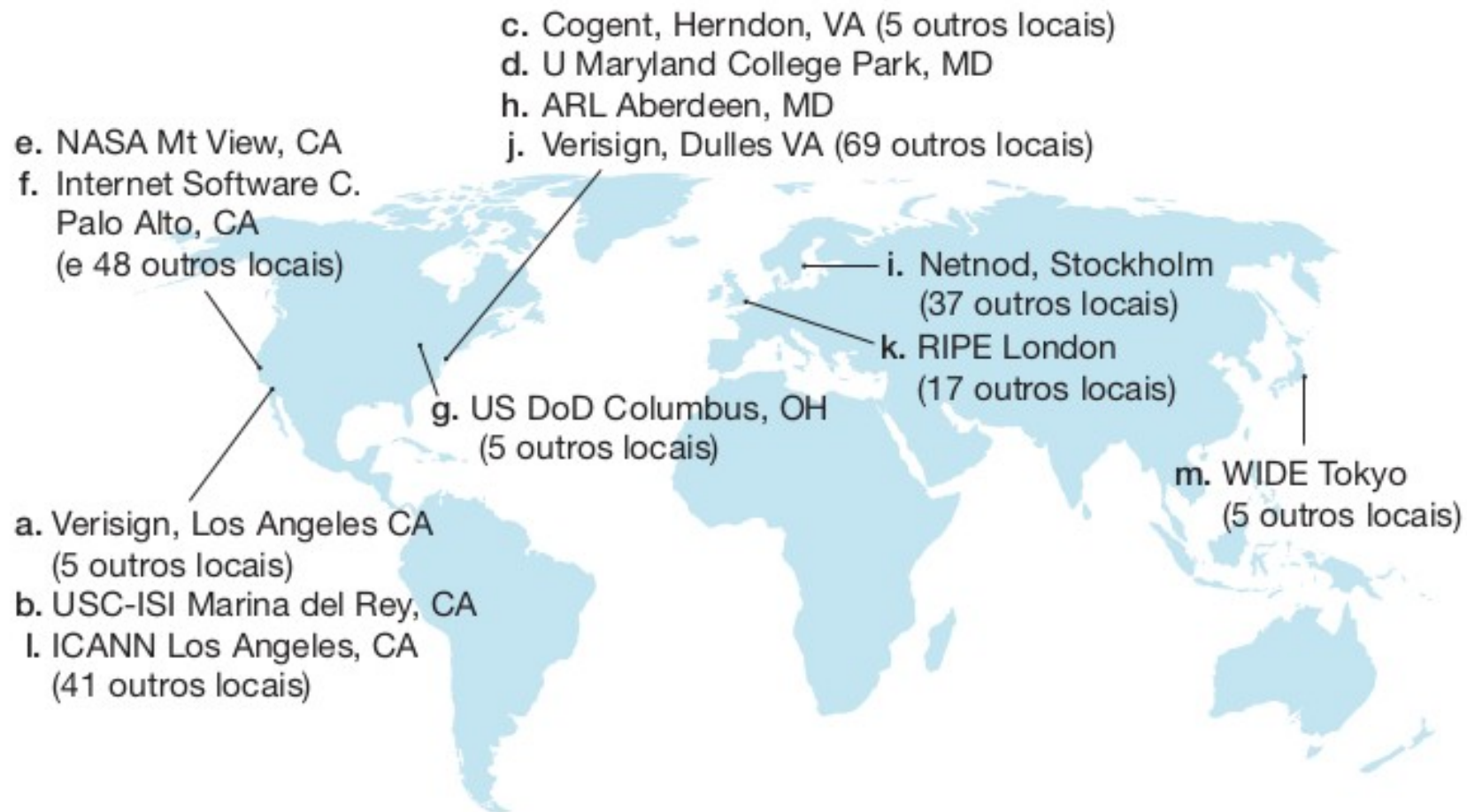
- “**solução**” .. DNS compõe-se de um grande nro. de servidores, organizados de maneira hierárquica e distribuídos por todo o mundo.
- “**servidores raiz**” .. há 13 servidores DNS raiz (denominados A a M) e a maior parte deles está localizada na América do Norte.



## 2 – Camada de Aplicação / 2.5 – Domain Name System

### ... 2.5.2 - Visão Geral do DNS

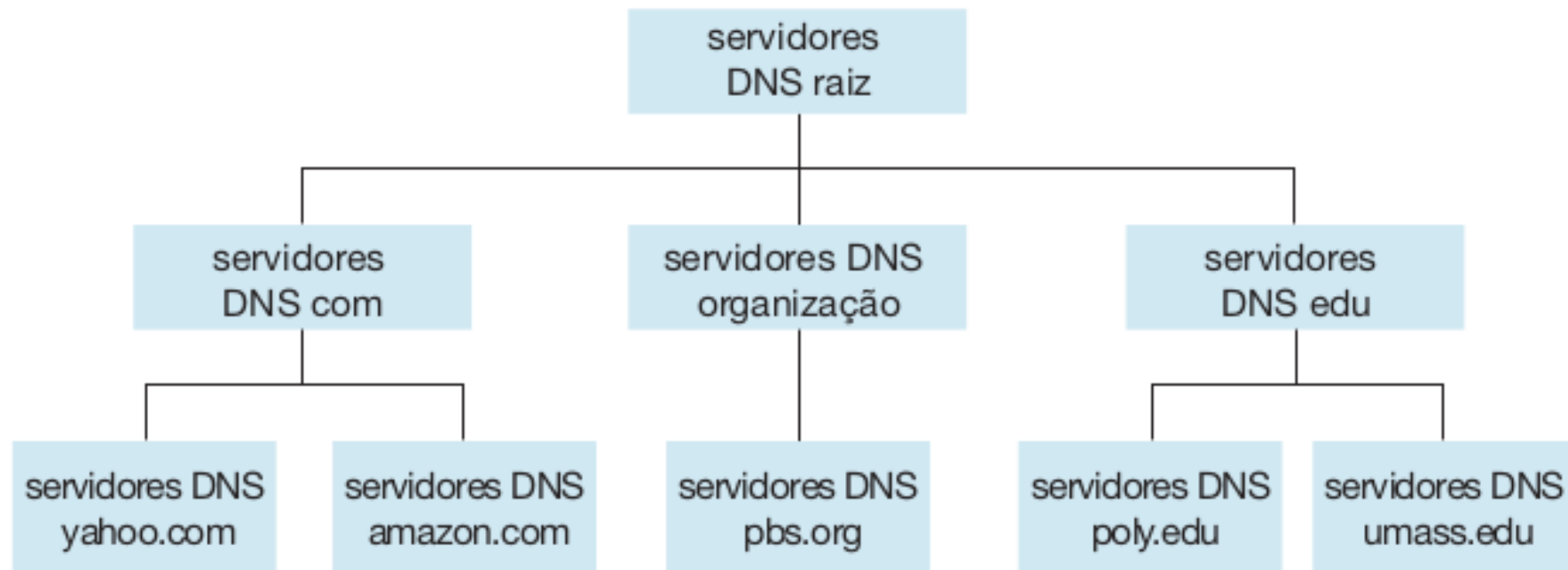
- “**servidores raiz**” .. há 13 servidores DNS raiz (denominados A a M) e a maior parte deles está localizada na América do Norte.



## 2 – Camada de Aplicação / 2.5 – Domain Name System

### ... 2.5.2 - Visão Geral do DNS

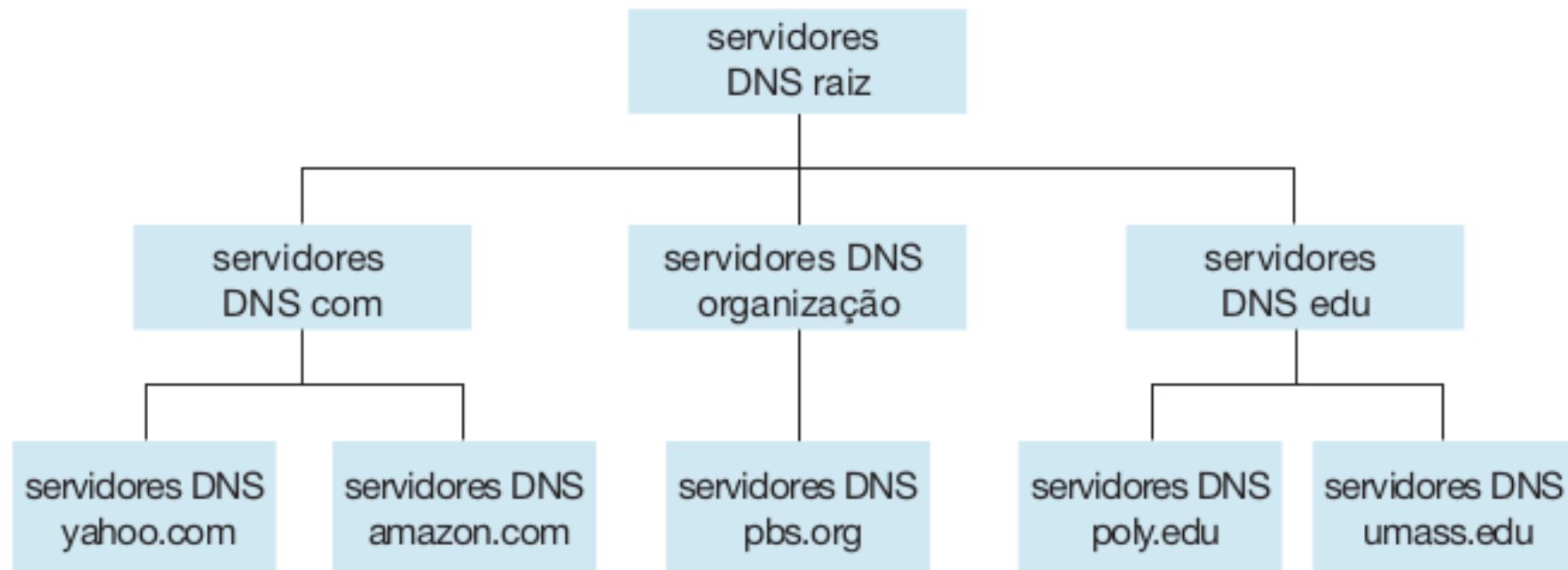
- “**servidores de alto nível**” .. servidores responsáveis por domínios de alto nível como **com**, **org**, **net**, **edu** e **gov**, e por todos os domínios de alto nível de países, tais como **uk**, **fr**, **ca** e **jp**.
- “**servidores de alto nível**” .. “**Top Level Domain**”



## 2 – Camada de Aplicação / 2.5 – Domain Name System

### ... 2.5.2 - Visão Geral do DNS

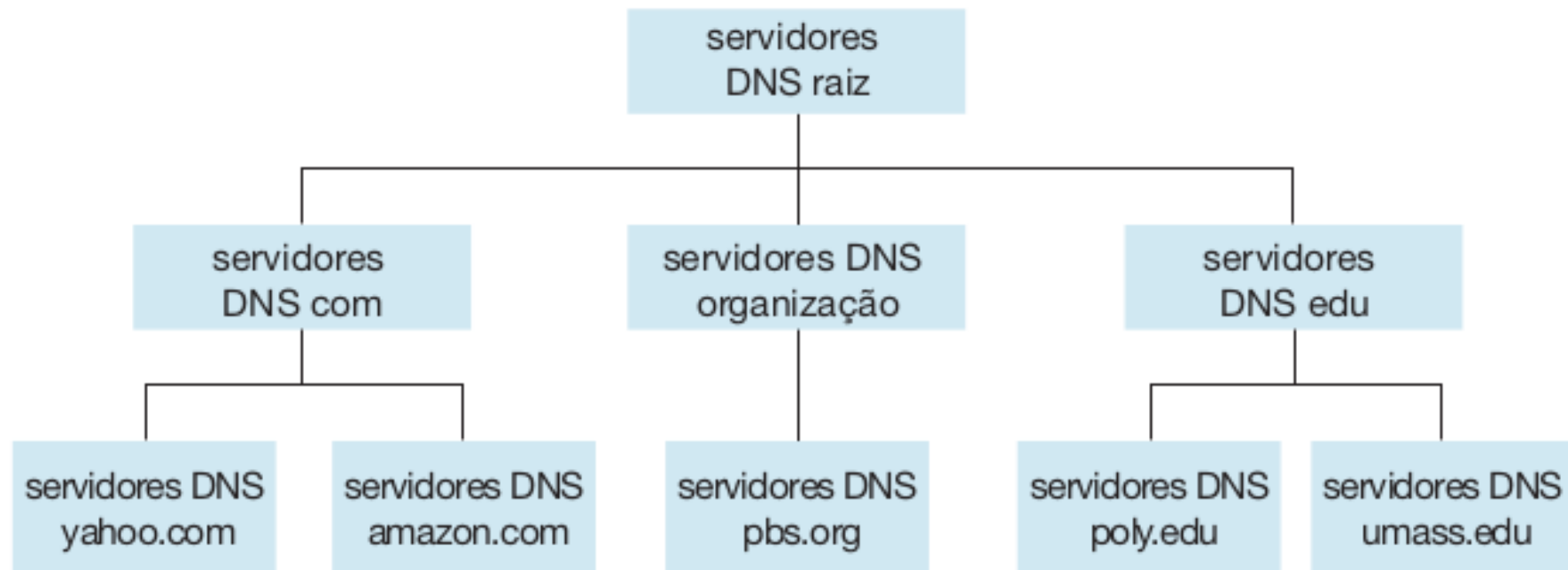
- “Verisign Global Registry Services” mantém os servidores “Top Level Domain” para o domínio de alto nível “**com**”, enquanto “EDUCAUSE” mantém os servidores TLD para o domínio de alto nível “**edu**”.
- “**Internet Assigned Number Authority**” (IANA) .. lista de todos os “domínios de alto nível” ou “Top Level Domain” (TLD).



## 2 – Camada de Aplicação / 2.5 – Domain Name System

### ... 2.5.2 - Visão Geral do DNS

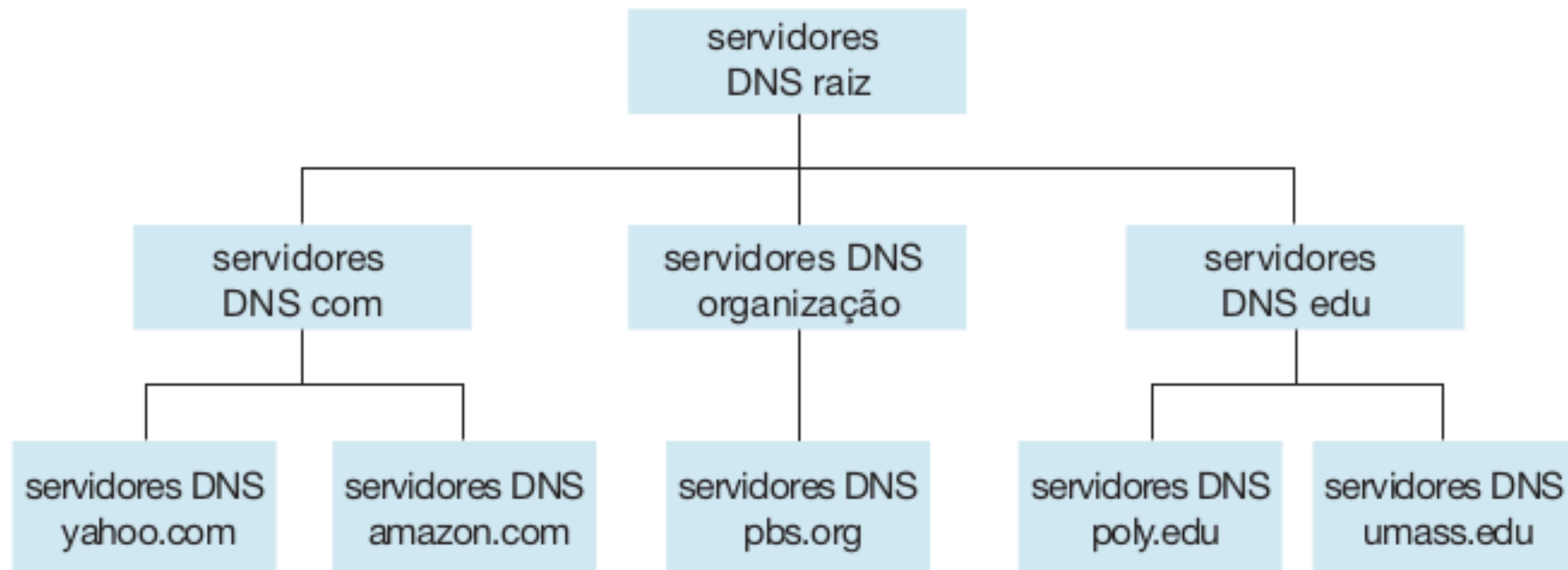
- “**servidores autoritativos**” .. servidores de uma organização que contém os registros DNS dos seus “hosts” e endereços IP.
- ... toda organização que tiver “hosts” que possam ser acessados publicamente na Internet deve acomodar um “**authoritative server**”.



## 2 – Camada de Aplicação / 2.5 – Domain Name System

### ... 2.5.2 - Visão Geral do DNS

- “**servidores autoritativos**” .. uma organização pode preferir executar seu próprio servidor DNS autoritativo para abrigar esses registros.
- ... como alternativa, a organização pode pagar para armazená-los em um servidor DNS autoritativo de algum provedor de serviço.



## 2 – Camada de Aplicação / 2.5 – Domain Name System

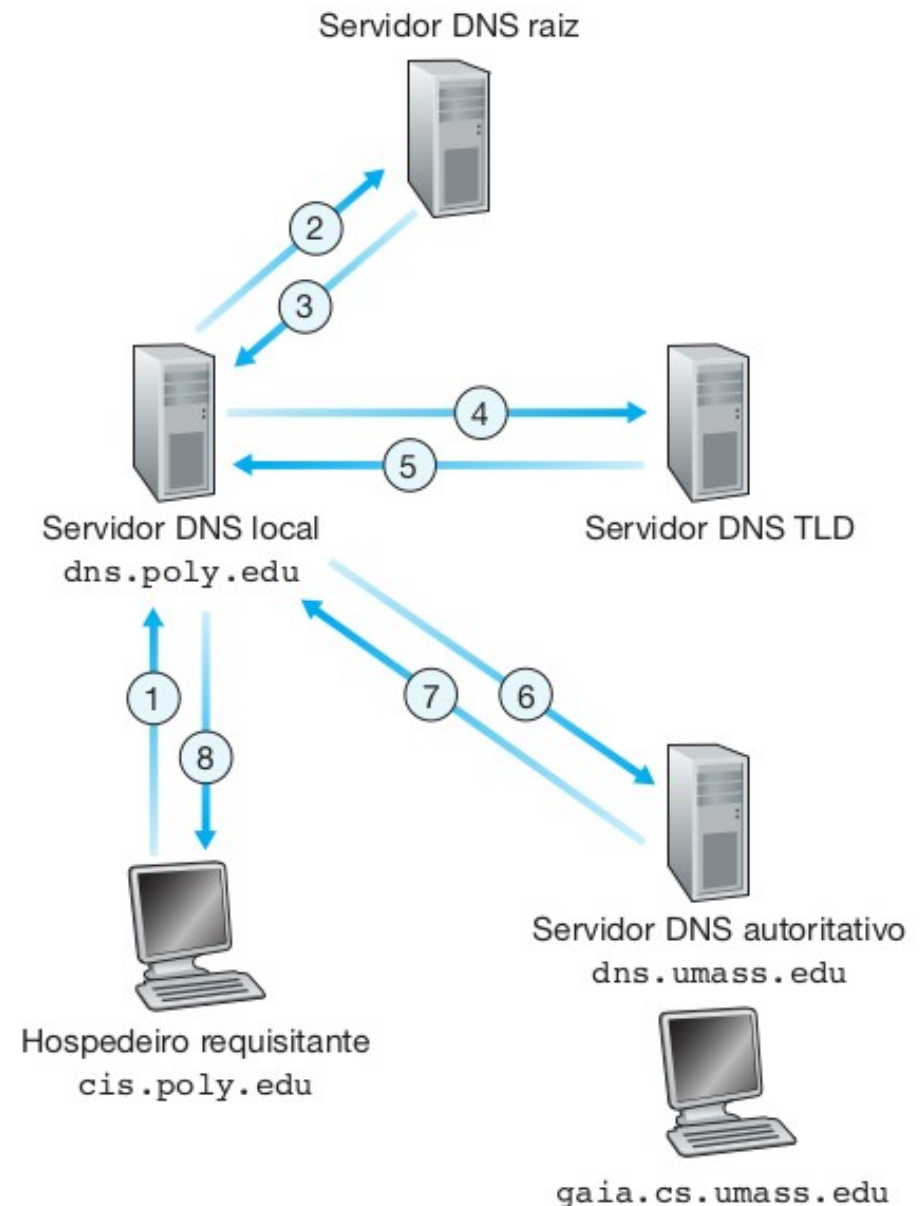
### ... 2.5.2 - Visão Geral do DNS

- Servidores DNS Raiz, TLD e Autoritativo pertencem à hierarquia de servidores DNS como apresentado anteriormente.
- “**Servidor DNS Local**” .. servidor DNS que não pertence, estritamente, à hierarquia de servidores, mas mesmo assim, é central para no DNS.
- Cada ISP como o de uma universidade, de um departamento acadêmico, de uma empresa ou de uma residência — tem um **Servidor DNS Local** (também denominado servidor **DNS default**).

## 2 – Camada de Aplicação / 2.5 – Domain Name System

### ... 2.5.2 - Visão Geral do DNS

- e.g., Suponha que o “host” “**cis.poly.edu**” precise do endereço IP de “**gaia.cs.umass.edu**” para fazer um consulta.
- ... considere que o Servidor DNS Local seja “**dns.poly.edu**”.
- ... considere que o Servidor DNS Autoritativo para “**gaia.cs.umass.edu**” seja “**dns.umass.edu**”.
- “**pergunta**” .. “como resolver o nome” ?

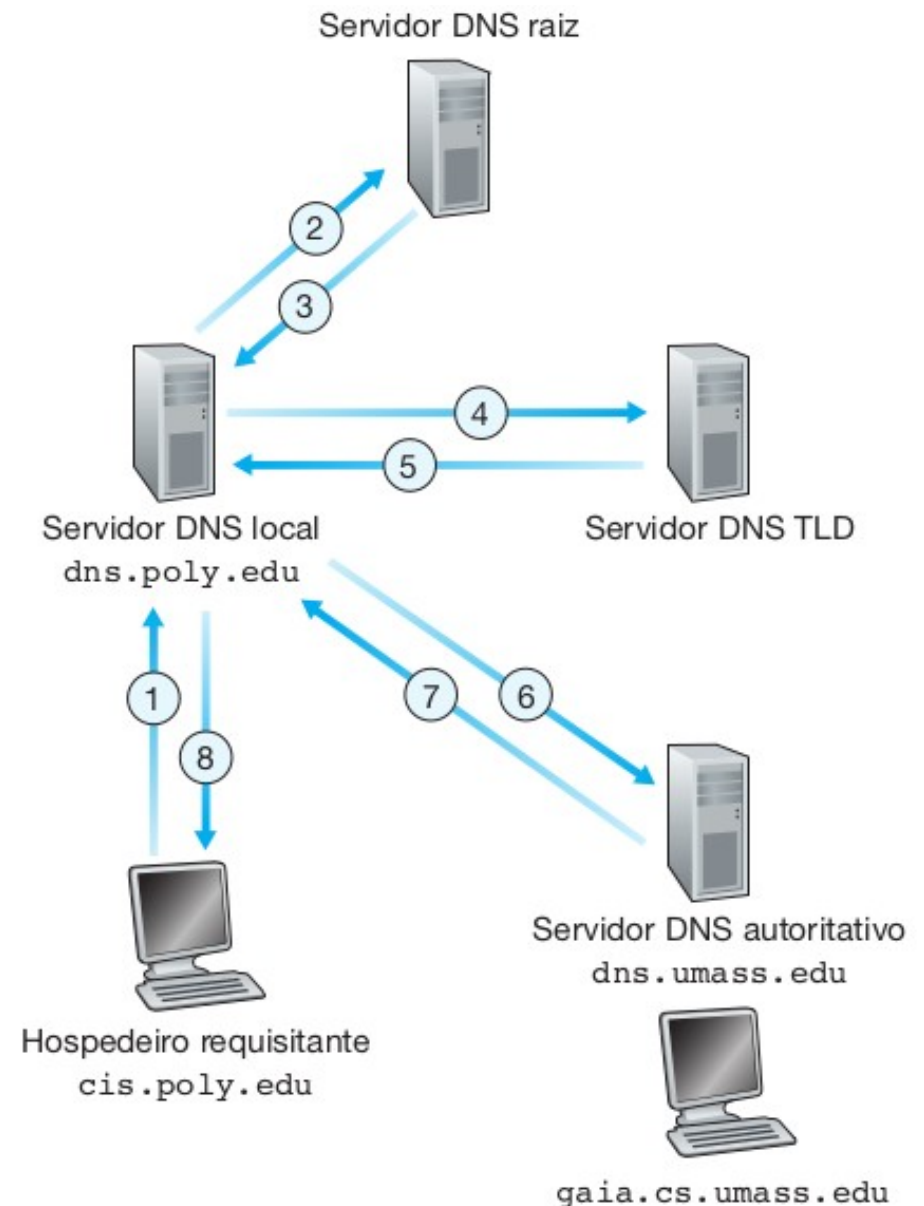




## 2 – Camada de Aplicação / 2.5 – Domain Name System

### ... 2.5.2 - Visão Geral do DNS

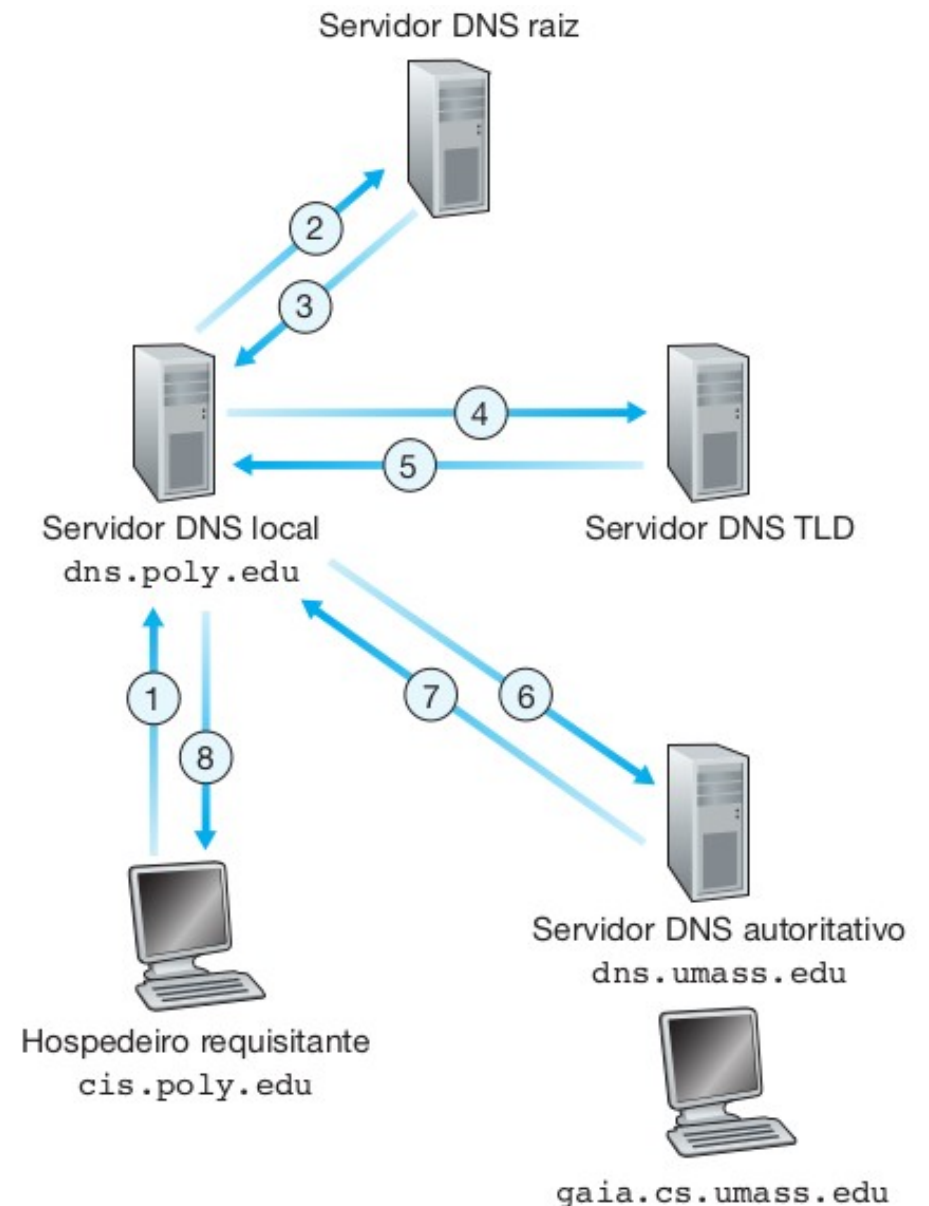
- 1) “host” “**cis.poly.edu**” envia uma mensagem de consulta DNS a seu servidor DNS local “**dns.poly.edu**” que contém o nome de “host” a ser traduzido para endereço IP, isto é, “**gaia.cs.umass.edu**”.
- 2) Servidor DNS local transmite a consulta a um servidor DNS raiz;



## 2 – Camada de Aplicação / 2.5 – Domain Name System

### ... 2.5.2 - Visão Geral do DNS

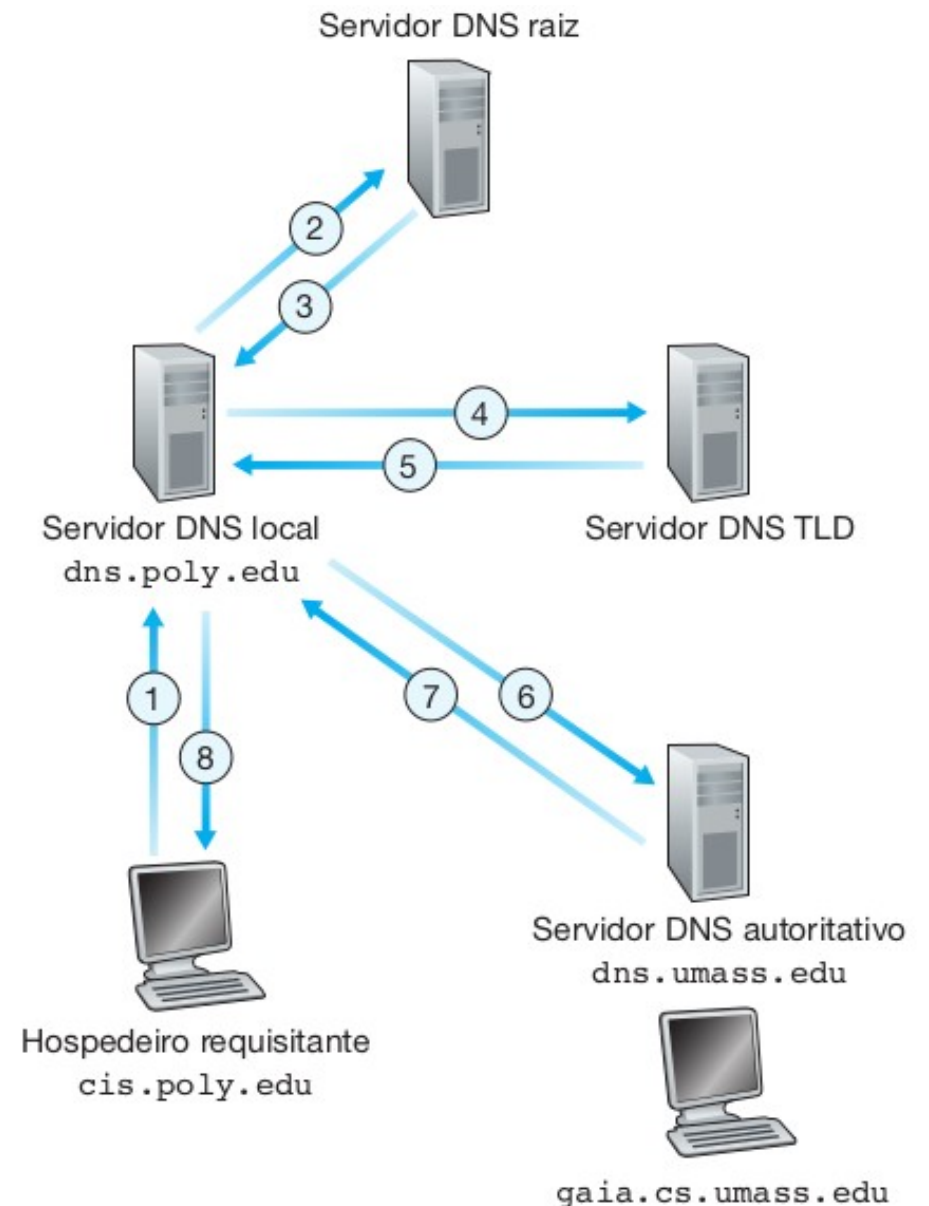
- 3) Servidor Raiz percebe o sufixo “edu” e retorna ao servidor DNS local “dns.poly.edu” uma lista de endereços IP contendo servidores TLD responsáveis por “edu”.
- 4) Servidor DNS Local “dns.poly.edu” retransmite a mensagem de consulta a um desses servidores TLD.



## 2 – Camada de Aplicação / 2.5 – Domain Name System

### ... 2.5.2 - Visão Geral do DNS

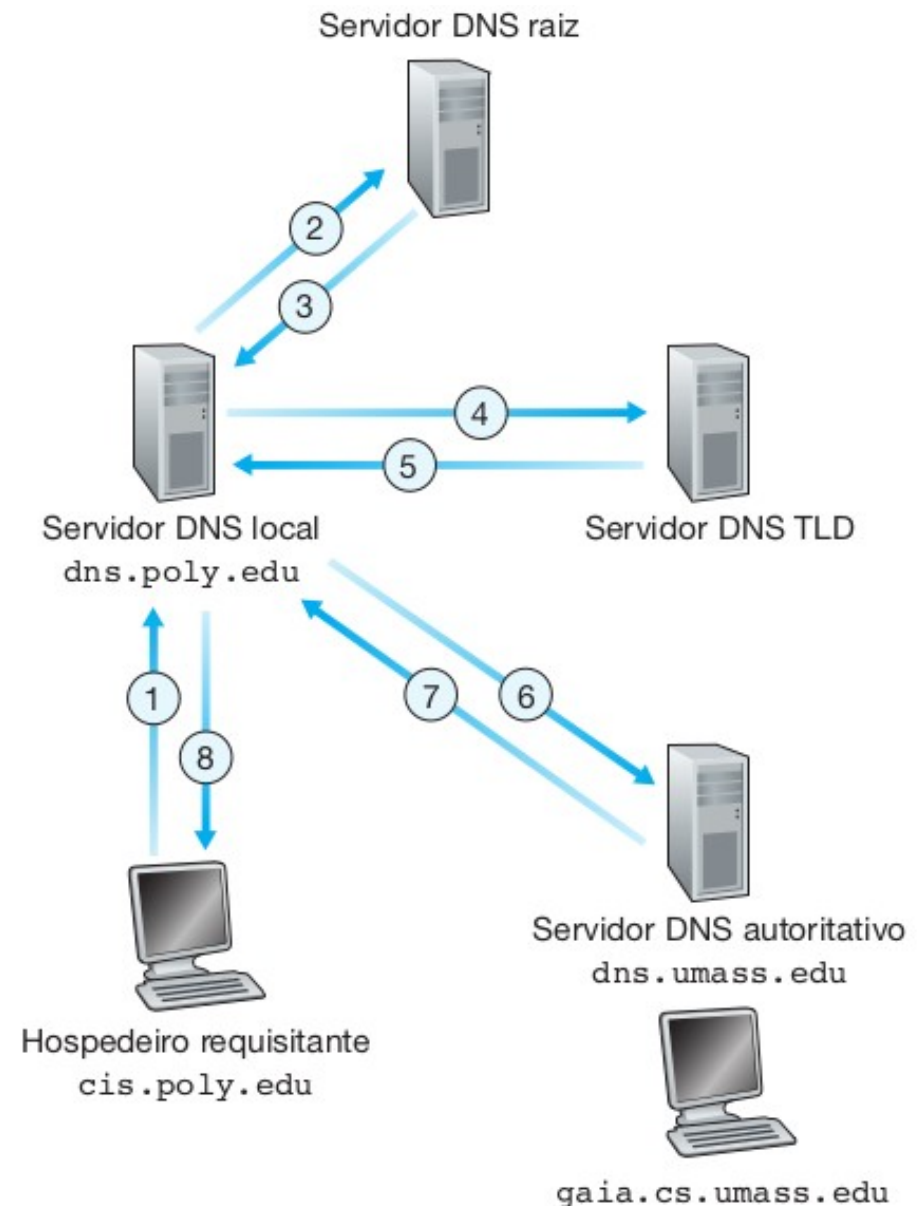
- 5) Servidor TLD percebe o sufixo “umass.edu” e responde com o endereço IP do servidor DNS Autorizado “dns.umass.edu”, ou seja, Servidor Autoritativo para a University of Massachusetts.
- 6) Servidor DNS Local “dns.poly.edu” reenvia a mensagem de consulta diretamente a “dns.umass.edu”.



## 2 – Camada de Aplicação / 2.5 – Domain Name System

### ... 2.5.2 - Visão Geral do DNS

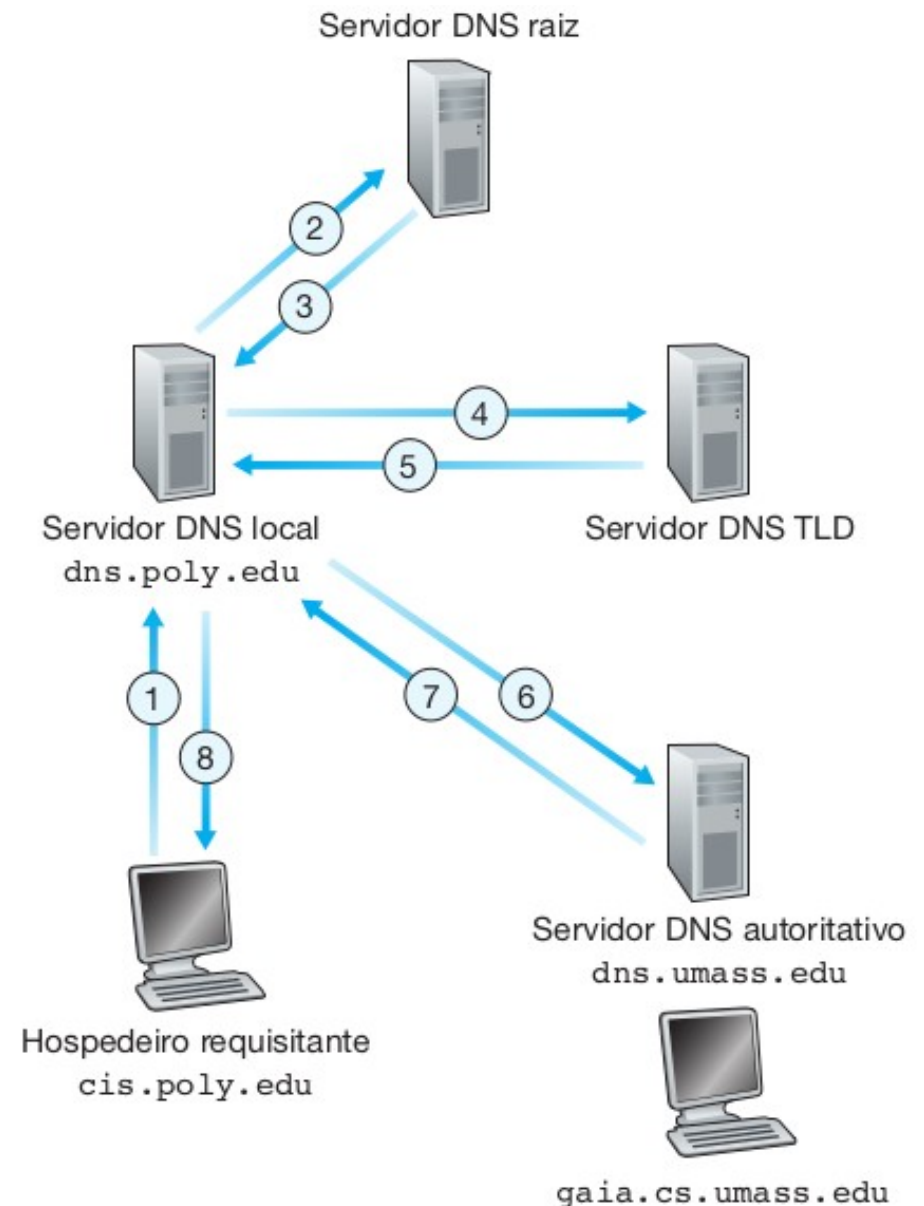
- 7) Servidor DNS Autoritativo “**dns.umass.edu**” responde com o endereço IP de “**gaia.cs.umass.edu**”.
- 8) Servidor DNS Local “**dns.poly.edu**” responde ao “host” “**cis.poly.edu**” o endereço IP de “**gaia.cs.umass.edu**”.



## 2 – Camada de Aplicação / 2.5 – Domain Name System

### ... 2.5.2 - Visão Geral do DNS

- e.g., Suponha que o “host” “**cis.poly.edu**” deseje o endereço IP de “**gaia.cs.umass.edu**”.
- ... considere que o Servidor DNS Local seja “**dns.poly.edu**”.
- ... considere o Servidor DNS Autoritativo “**dns.umass.edu**” para “**gaia.cs.umass.edu**”.
- **solução** .. consultas recursivas e consultas iterativas entre os servidores local e remotos.



## 2 – Camada de Aplicação / 2.5 – Domain Name System

### ... 2.5.2 - Visão Geral do DNS

- **“solução”** .. consultas recursivas e consultas iterativas entre os servidores locais e remotos.
- ... consulta enviada do “host” “cis.poly.edu” para Servidor DNS Local “dns.poly.edu” é recursiva, visto que pede a “dns.poly.edu” que obtenha o mapeamento de nome para IP em nome de “cis.poly.edu”.
- ... 03 consultas subsequentes são iterativas, visto que todas as respostas são retornadas diretamente a “dns.poly.edu”.
- **“observação”** .. qualquer consulta DNS pode ser Iterativa ou Recursiva, mas na prática a consulta de um “host” requisitante ao Servidor DNS Local é Recursiva e todas as outras consultas são Iterativas.

## 2 – Camada de Aplicação / 2.5 – Domain Name System

### ... 2.5.2 - Visão Geral do DNS

- “**Cache DNS**” .. DNS explora extensivamente o cache para melhorar o desempenho quanto ao atraso e reduzir o nro. de mensagens.
- “**princípio**” .. servidor DNS ao receber uma resposta DNS, pode fazer cache das informações da resposta em sua memória local.
- e.g., suponha que o “host” “**cis.poly.edu**” deseje o endereço IP de “**gaia.cs.umass.edu**”, que o servidor DNS Local seja “**dns.poly.edu**” e o “**dns.umass.edu**” seja o autoritativo para “**gaia.cs.umass.edu**”.
- ... toda vez que o servidor DNS local “**dns.poly.edu**” recebe uma resposta de algum servidor DNS, pode fazer **cache** de qualquer informação contida na resposta.

## 2 – Camada de Aplicação / 2.5 – Domain Name System

### 2.5.3 – Registros e Mensagens DNS

- “**Registros de Recursos**” ou RRs .. informações que o Servidores DNS armazenam e que fornecem mapeamentos de nomes para IPs.
- “**registro de recurso**” .. tupla de 04 elementos que contém os seguintes campos: **Name, Value, Type, TTL**
- TTL ou “Time To Live” .. tempo de vida útil do registro de recurso e determina quando um recurso deve ser removido de um cache.
- Para a 1ª análise, vamos ignorar o TTL e entender o significado de 03 elementos da tupla, ou seja, **Name, Value e Type**:



## 2 – Camada de Aplicação / 2.5 – Domain Name System

### ... 2.5.3 – Registros e Mensagens DNS

- Seja o significado de 02 elementos da tupla, ou seja, **Name** e **Value** são dependentes do elemento **Type** ..
- Se **Type=A**, então **Name** é um nome de “host” e **Value** é o endereço IP para o nome de “host”, ou seja, mapeamento padrão entre nomes de “hosts” e endereços IP.
- Se **Type=NS**, então **Name** é um domínio (como “foo.com” ) e **Value** é o nome de um Servidor DNS Autoritativo que sabe como obter os endereços IP para “hosts” do domínio.
- ... esse registro é usado para encaminhar consultas DNS ao longo da cadeia de consultas (como vimos no exemplo anterior).

## 2 – Camada de Aplicação / 2.5 – Domain Name System

### ... 2.5.3 – Registros e Mensagens DNS

- Para a 1ª análise, vamos ignorar o TTL e entender o significado de 02 elementos da tupla, ou seja, **Name** e **Value** dependentes do **Type**:
- Se **Type=CNAME**, então **Value** é um nome canônico de “host” para o apelido de hospedeiro contido em **Name**.
- ... esse registro pode fornecer aos hospedeiros consultantes o nome canônico correspondente a um apelido de “host”.
- Se **Type=MX**, então **Value** é o nome canônico de um servidor de e-Mail cujo apelido de “host” está contido em **Name**.
- .. p.ex., Registro MX - “foo.com”, “mail.bar.foo.com”, MX.

## 2 – Camada de Aplicação / 2.5 – Domain Name System

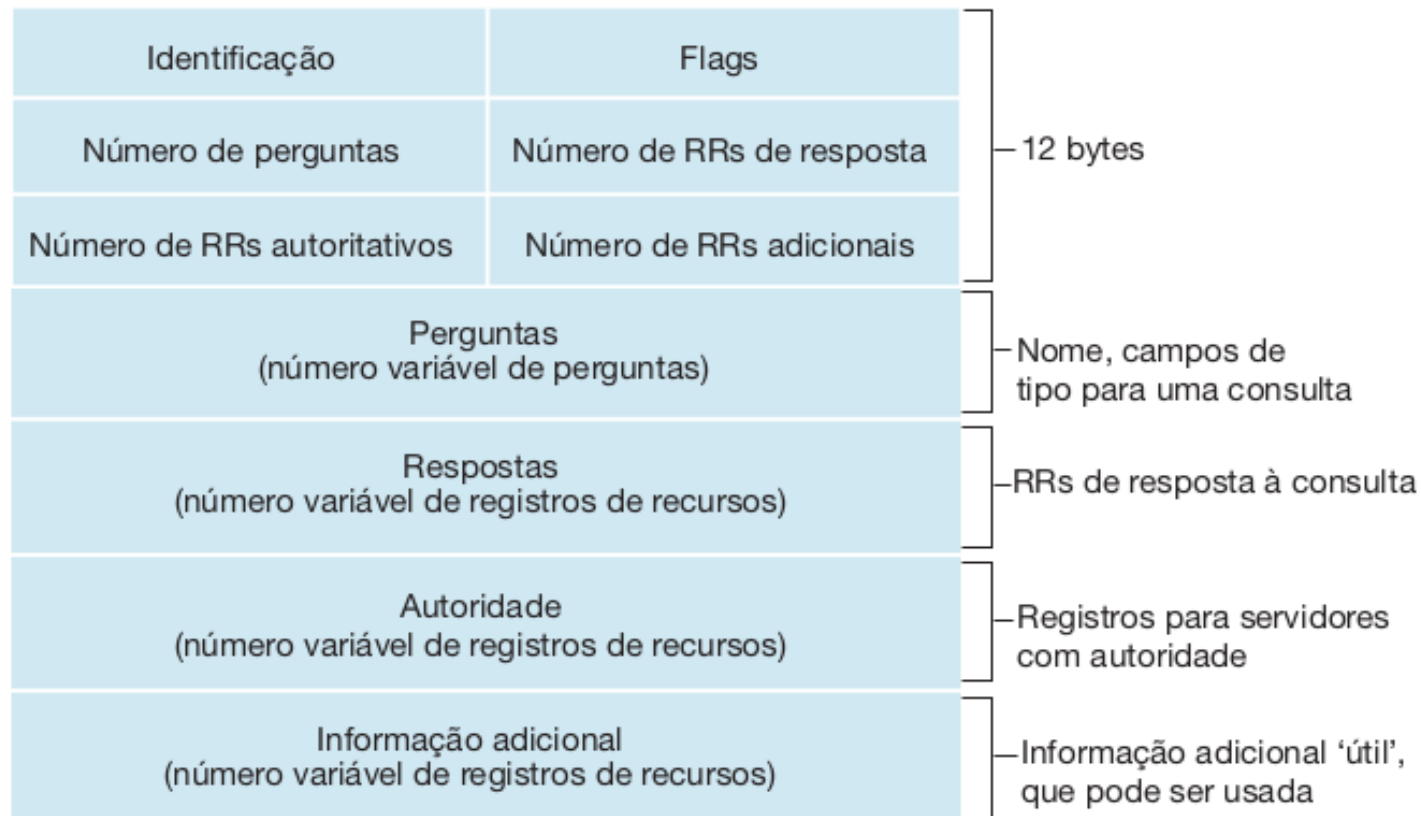
### ... 2.5.3 – Registros e Mensagens DNS

- Se um servidor DNS tiver Autoridade para determinado nome de “host”, então conterá um registro Type A para o nome de “host”.
- ... mesmo que não tenha autoridade, o servidor DNS pode conter um registro Type A em sua cache.
- Se um servidor não tiver Autoridade para um nome de “host”, conterá um registro Type NS para o domínio que inclui o nome de Serv. DNS.
- ... nome e um registro Type A que fornece o endereço IP do servidor DNS no campo Value do registro NS.

## 2 – Camada de Aplicação / 2.5 – Domain Name System

### ... 2.5.3 – Registros e Mensagens DNS

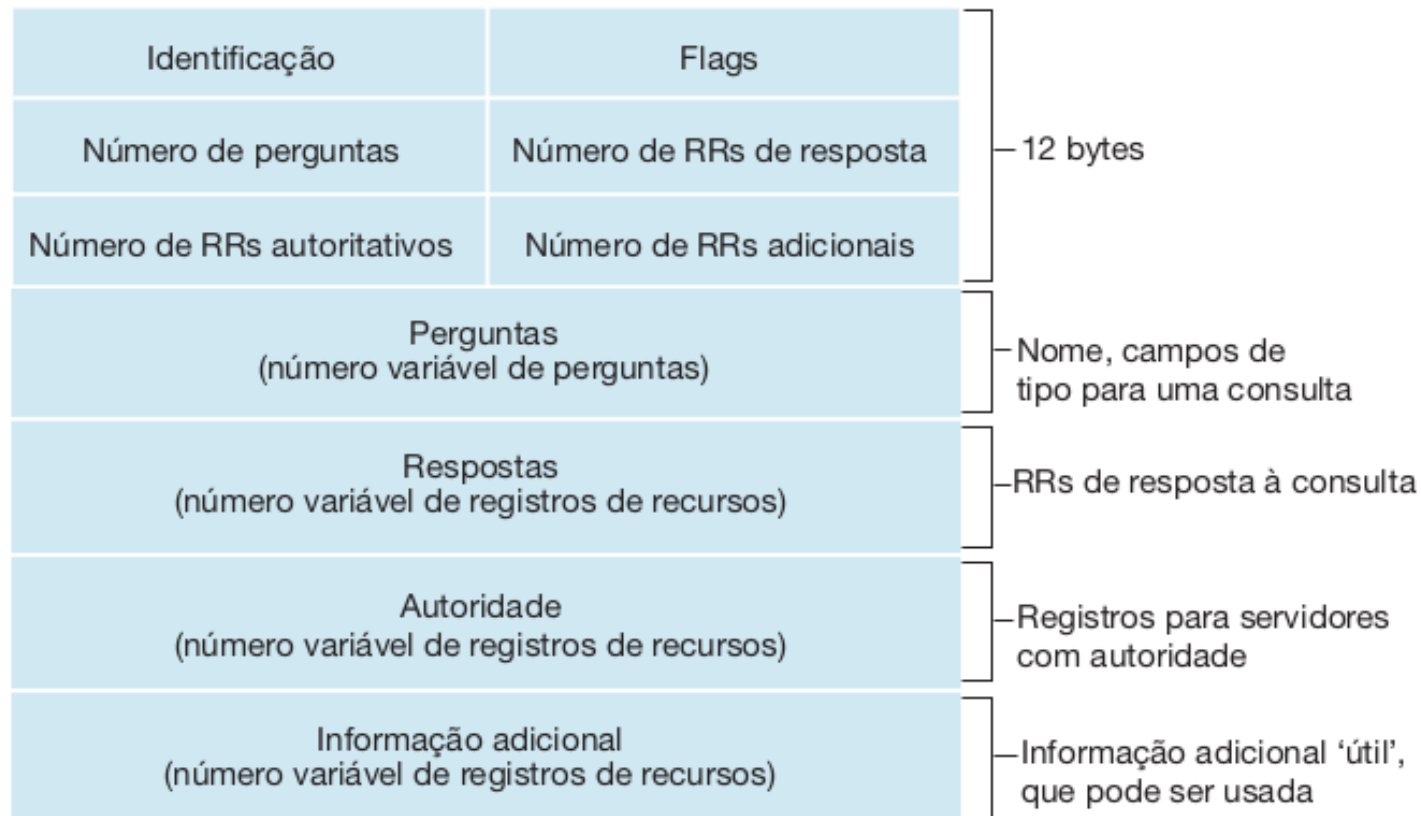
- “Mensagem DNS” .. “request” e “reply” tem o mesmo formato, com cada linha do cabeçalho contemplando 4 bytes ou 32 bits.
- “**identificador**” com 16 bits .. permite a identificação e/ou combinação de respostas recebidas para com as consultas enviadas.



## 2 – Camada de Aplicação / 2.5 – Domain Name System

### ... 2.5.3 – Registros e Mensagens DNS

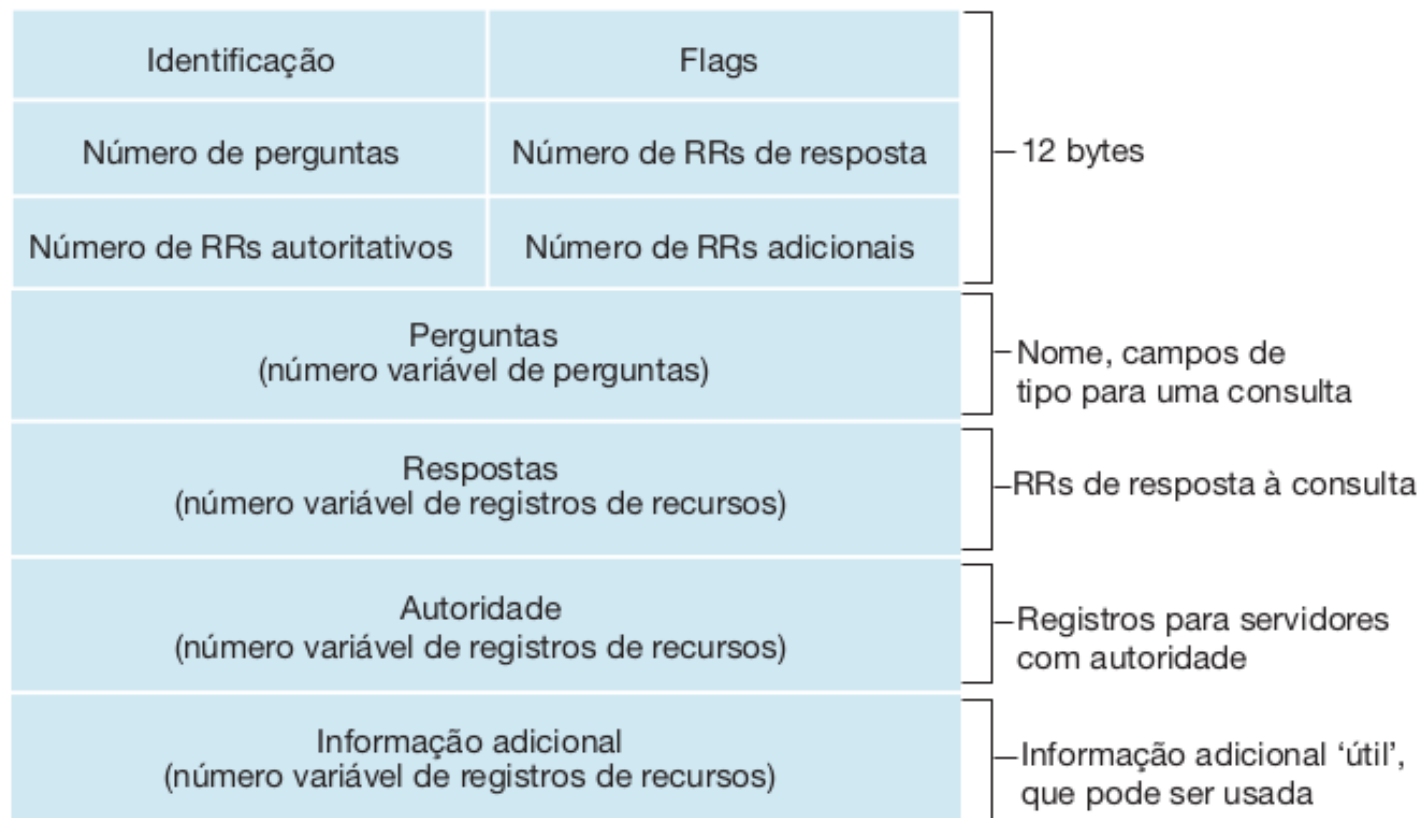
- “**flag**” com 16 bits .. contempla bits para marcação de servidor autoritativo, consulta com recursão, etc.
- “**04 campos tamanho**” com 16 bits .. indicam o nro. de ocorrências dos 04 tipos de seção de dados que se seguem ao cabeçalho.



## 2 – Camada de Aplicação / 2.5 – Domain Name System

### ... 2.5.3 – Registros e Mensagens DNS

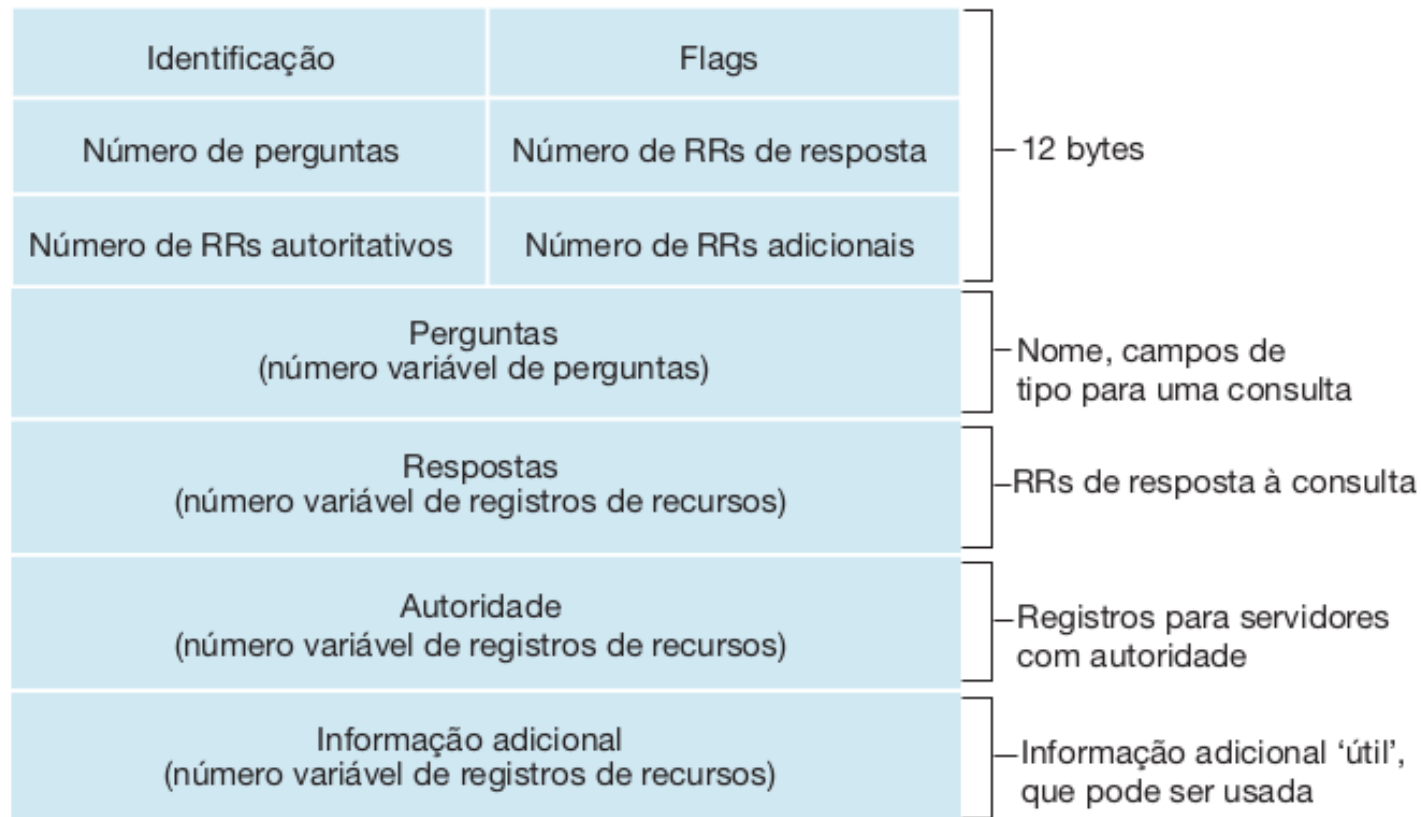
- “**pergunta**” .. contém o nome que está sendo consultado e a pergunta que está sendo feita para o nome em questão.
- p.ex., endereço de “host” associado a um nome (Type A) ou o servidor de e-Mail para um nome (Type MX)



## 2 – Camada de Aplicação / 2.5 – Domain Name System

### ... 2.5.3 – Registros e Mensagens DNS

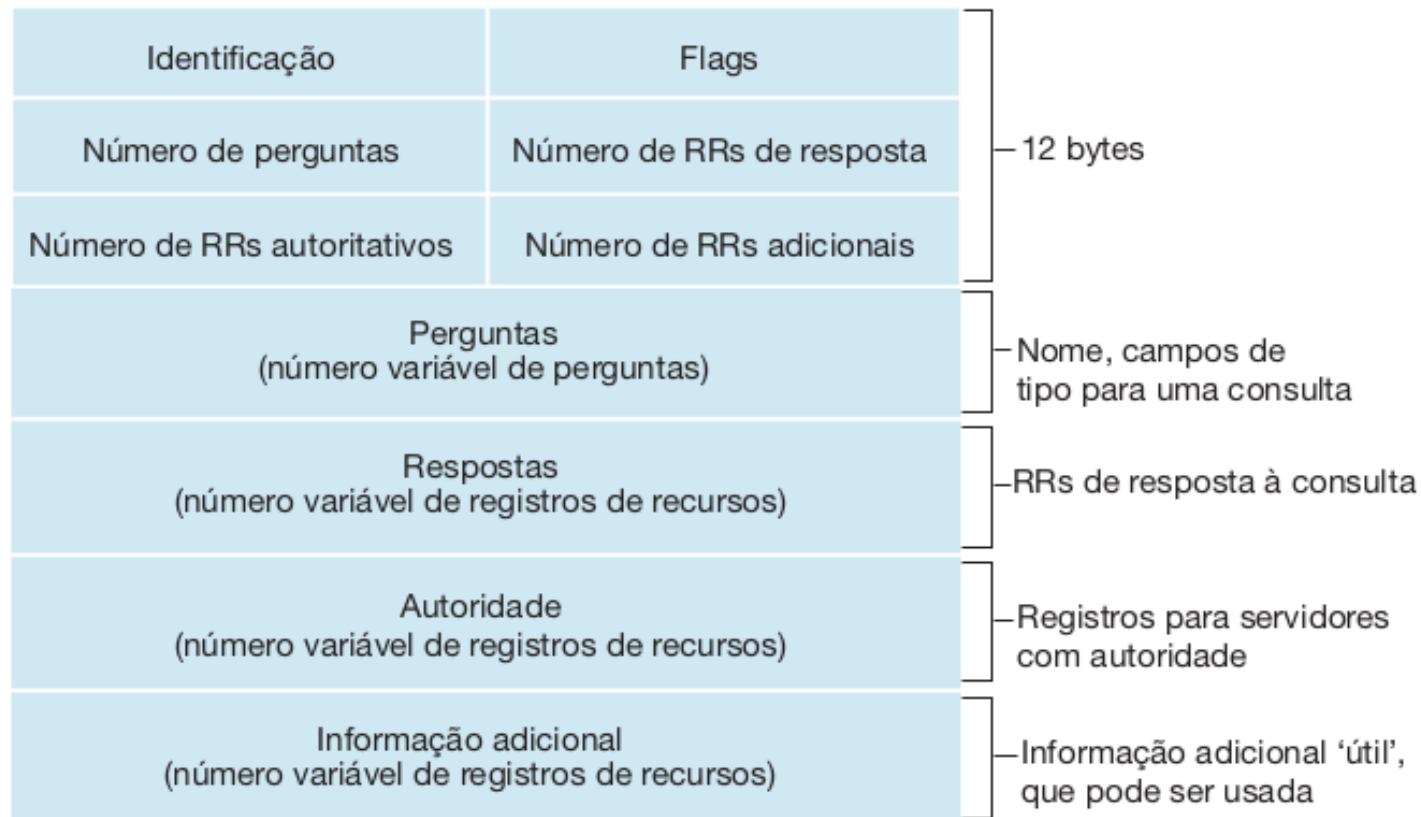
- “**resposta**” .. contém os registros de recursos (respostas) para o nome que foi consultado (pergunta) originalmente.
- p.ex., lembre-se de que em cada registro de recurso há o Type (por exemplo, A, NS, CSNAME e MX), o Value e o TTL.



## 2 – Camada de Aplicação / 2.5 – Domain Name System

### ... 2.5.3 – Registros e Mensagens DNS

- “**autoridade**” .. contém registros de outros servidores autoritativos.
- “**informação adicional**” .. contém outros registros úteis, p.ex., campo resposta em uma resposta a uma consulta MX conterá um registro de recurso que informa o nome canônico de um servidor de e-Mail.





## 2 – Camada de Aplicação / 2.5 – Domain Name System

### ... 2.5.3 – Registros e Mensagens DNS

- “**Consulta DNS**” .. isso pode ser feito facilmente com o programa “**nslookup**” disponível na maioria das plataformas Windows e UNIX.
- e.g., No Sistema Operacional Windows, abra o Prompt de Comando e chame o programa “**nslookup**” apenas digitando “nslookup”.
- ... depois de chamar o programa, você pode enviar uma consulta DNS a qualquer servidor DNS (Raiz, TLD ou Autoritativo).
- ... após receber a mensagem de resposta do servidor DNS, o **nslookup** apresenta os registros incluídos na resposta (formato legível).

## 2 – Camada de Aplicação / 2.5 – Domain Name System

### ... 2.5.3 – Registros e Mensagens DNS

- **“Inserindo Registro no DNS”** .. 1º passo é registrar o nome de domínio p.ex., “networkutopia.com” em uma entidade registradora.
- **“entidade registradora”** .. organização comercial que verifica se o nome de domínio é exclusivo, registra-o no banco de dados do DNS e cobra uma taxa por seus serviços.
- **Internet Corporation for Assigned Names and Numbers (ICANN)** .. responsável pelo credenciamento de entidades registradoras.
- ... lista completa dessas entidades - “<http://www.internic.net>”.

## 2 – Camada de Aplicação / 2.5 – Domain Name System

### ... 2.5.3 – Registros e Mensagens DNS

- e.g., seja o registro do nome de domínio “networkutopia.com”.
- .. esta solicitação exige que seja informado os nomes e endereços IP dos seus Servidores DNS com Autoridade, Primários e Secundários.
- ... suponha que os nomes e IPs sejam: “dns1.networkutopia.com” 212.212.212.1 e “dns2.networkutopia.com” 212.212.212.2.
- “Entidade Registradora” .. está encarregada de providenciar a inserção dos registros Type NS e Type A nos servidores TLD do domínio
- ... com cada um dos 02 servidores de nomes autorizados:
- (networkutopia.com, dns1.networkutopia.com, NS)
- (dns1.networkutopia.com, 212.212.212.1, A)

## 2 – Camada de Aplicação / 2.5 – Domain Name System

### ... 2.5.3 – Registros e Mensagens DNS

- e.g., seja o registro do nome de domínio “networkutopia.com”.
- .. esta solicitação exige que seja informado os nomes e endereços IP dos seus Servidores DNS com Autoridade, Primários e Secundários.
- ... providenciar a inserção nos Servidores DNS com Autoridade o RR Type A para o Servidor Web “www.networkutopia.com”.
- ... providenciar a inserção nos Servidores DNS com Autoridade o RR Type MX para o Servidor de e-Mail “mail.networkutopia.com”.
- Recentemente, acrescentou-se ao Protocolo DNS a opção UPDATE, que permite a inserção ou remoção dinâmica no banco de dados por meio de mensagens DNS [RFC 2136] e [RFC 3007].

## 2 – Camada de Aplicação / 2.6 – Aplicações P2P

### 2.6 – Aplicações P2P

- HTTP ou Web, e-Mail e DNS ... empregam Arquitetura Cliente/Servidor com dependência significativa na disponibilidade 24/7, ou seja, disponibilidade de servidores 24 horas por 07 dias na semana.
- “**comparação**” .. na Arquitetura P2P, a dependência é mínima (se houver) para que os servidores permaneçam sempre ligados.
- “**peer protocols**” ... pares não são de propriedade de um provedor de serviços, mas sim de desktops e notebooks controlados por usuários, logo, não estão disponíveis de forma homogênea e ininterrupta.

## 2 – Camada de Aplicação / 2.6 – Aplicações P2P

### ... 2.6 – Aplicações P2P

- “**disponibilidade**” .. probabilidade de um sistema se encontrar disponível e em operação para ser usado.
- ... para calcular a disponibilidade, é preciso comparar as horas em que o ativo esteve disponível com as horas de trabalho planejadas.
- “**confiabilidade**” .. capacidade de um item desempenhar uma função sob condições específicas, durante um dado intervalo de tempo.
- MTBF (Mean Time Between Failure) .. é um bom indicador da confiabilidade de um ativo durante o seu ciclo de vida e, normalmente, quanto maior a confiabilidade, maior a disponibilidade do sistema.

## 2 – Camada de Aplicação / 2.6 – Aplicações P2P

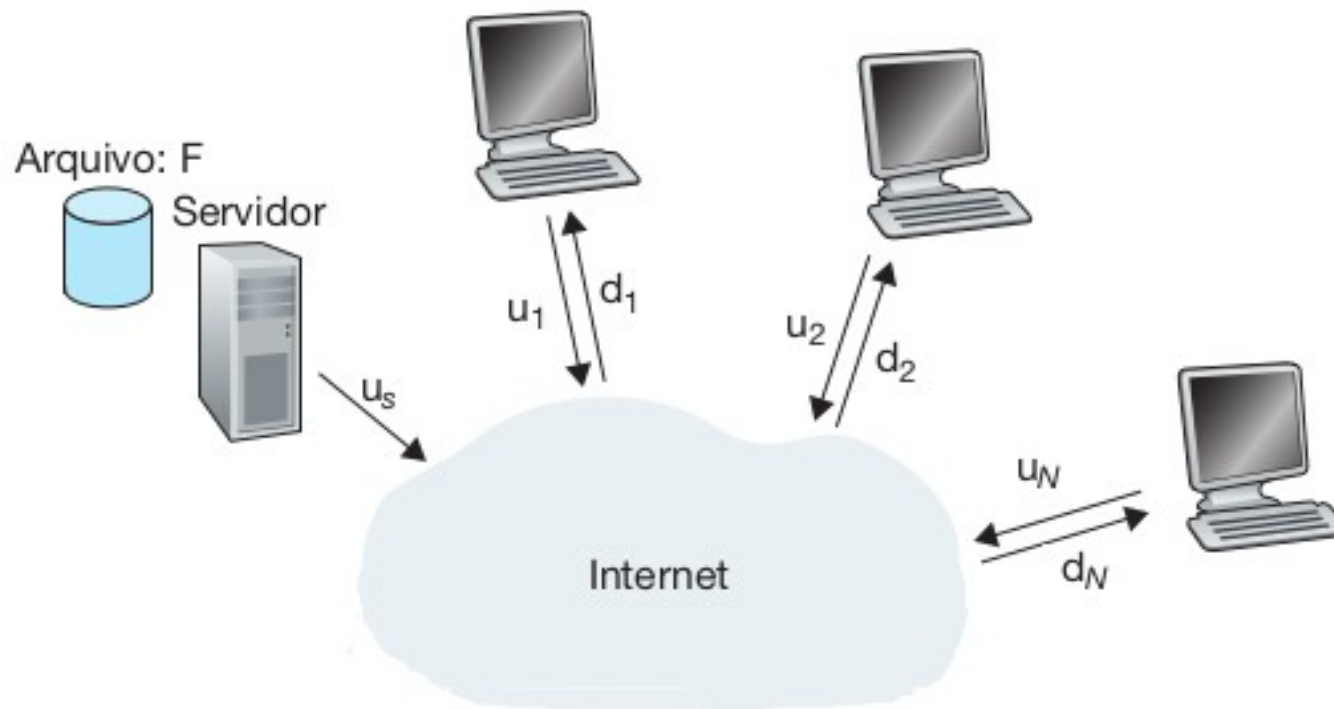
### ... 2.6 – Aplicações P2P

- “**objetivo**” - estudar Aplicações P2P que expõem com clareza a autoescalabilidade da Arquitetura P2P.
- “**distribuição de arquivos**” .. estudar aplicações para distribuição de arquivos, onde a partir de uma única fonte, distribui-se para um grande nro. de pares. p.ex., Sistema BitTorrent.
- “**banco de dados distribuídos**” ... estudar um banco de dados distribuído em uma grande comunidade de pares, p.ex., explorando o conceito de Distributed Hash Table (DHT).

## 2 – Camada de Aplicação / 2.6 – Aplicações P2P

### 2.6.1 – Distribuição de Arquivos P2P

- e.g., Seja a Distribuição de um Arquivo GRANDE a partir de um único servidor a um grande nro. de “hosts” (chamados pares ou “peers”).
- **“distribuição de arquivo cliente-servidor”** .. servidor deve enviar uma cópia para cada par colocando um enorme fardo sobre o servidor e consumindo grande quantidade de largura de banda do mesmo.





## 2 – Camada de Aplicação / 2.6 – Aplicações P2P

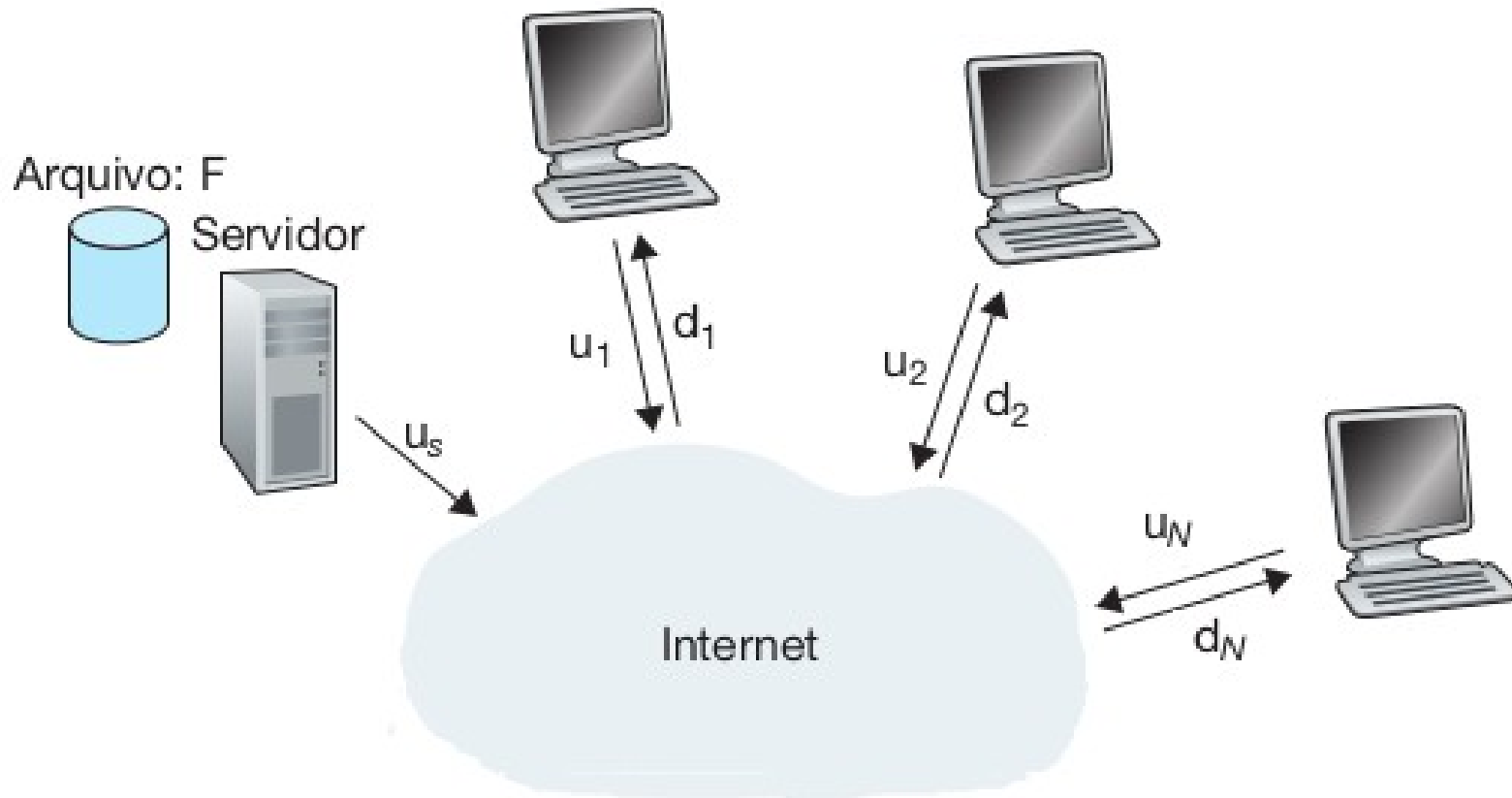
### ... 2.6.1 – Distribuição de Arquivos P2P

- e.g., Seja a Distribuição de um Arquivo GRANDE a partir de um único servidor a um grande nro. de “hosts” (chamados pares ou “peers”).
- **“distribuição de arquivo peer-to-peer”** .. cada par pode redistribuir qualquer parte do arquivo recebido para outros pares, auxiliando, assim, o servidor no processo de distribuição.
- e.g., Sistema BitTorrent foi desenvolvido por Bram Cohen, mas já é possível usar diferentes clientes independentes de BitTorrent.
- ... o fato dos diferentes clientes manterem a conformidade com o Protocolo BitTorrent é o que garante a interoperabilidade.

## 2 – Camada de Aplicação / 2.6 – Aplicações P2P

### ... 2.6.1 – Distribuição de Arquivos P2P

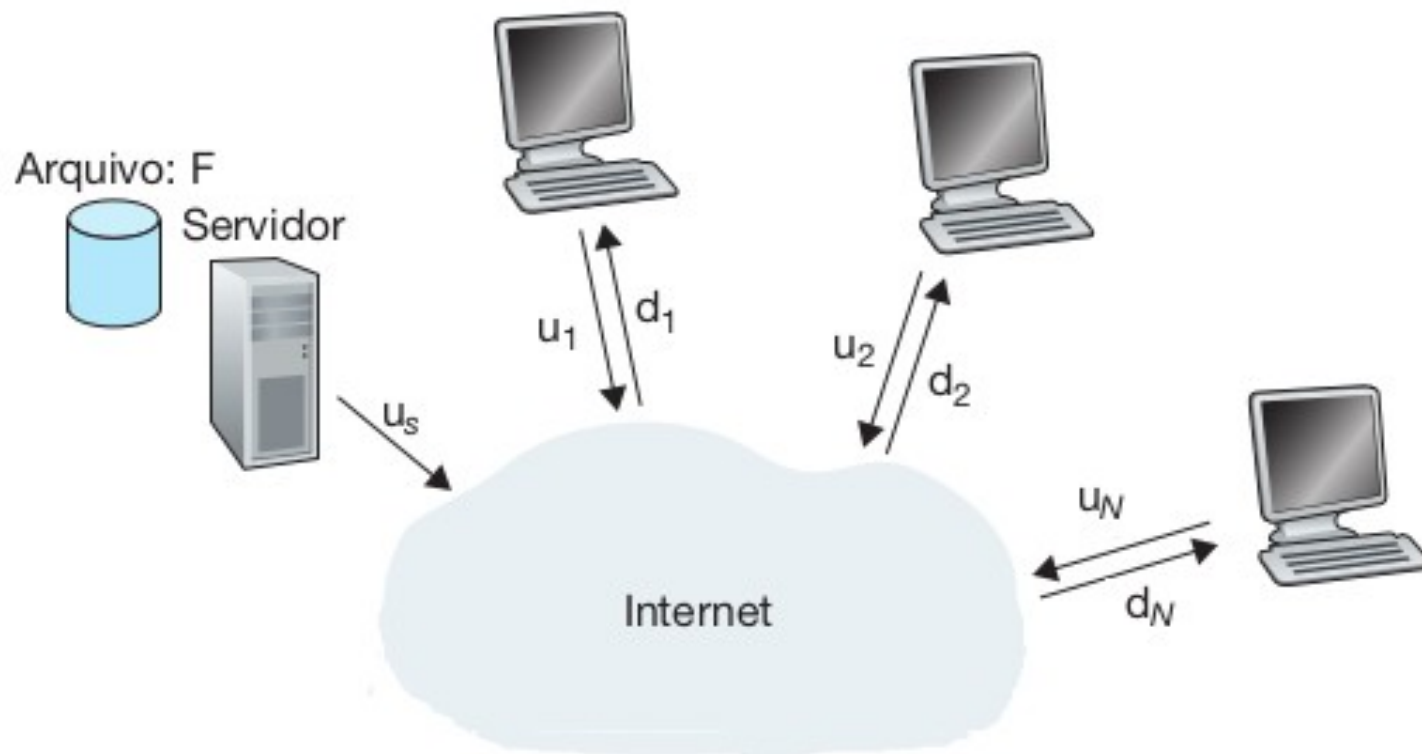
- “**comparar arquiteturas quanto a escalabilidade**” .. considere um modelo quantitativo para a distribuição de arquivo a um conjunto fixo de pares para ambos os tipos de arquitetura.



## 2 – Camada de Aplicação / 2.6 – Aplicações P2P

### ... 2.6.1 – Distribuição de Arquivos P2P

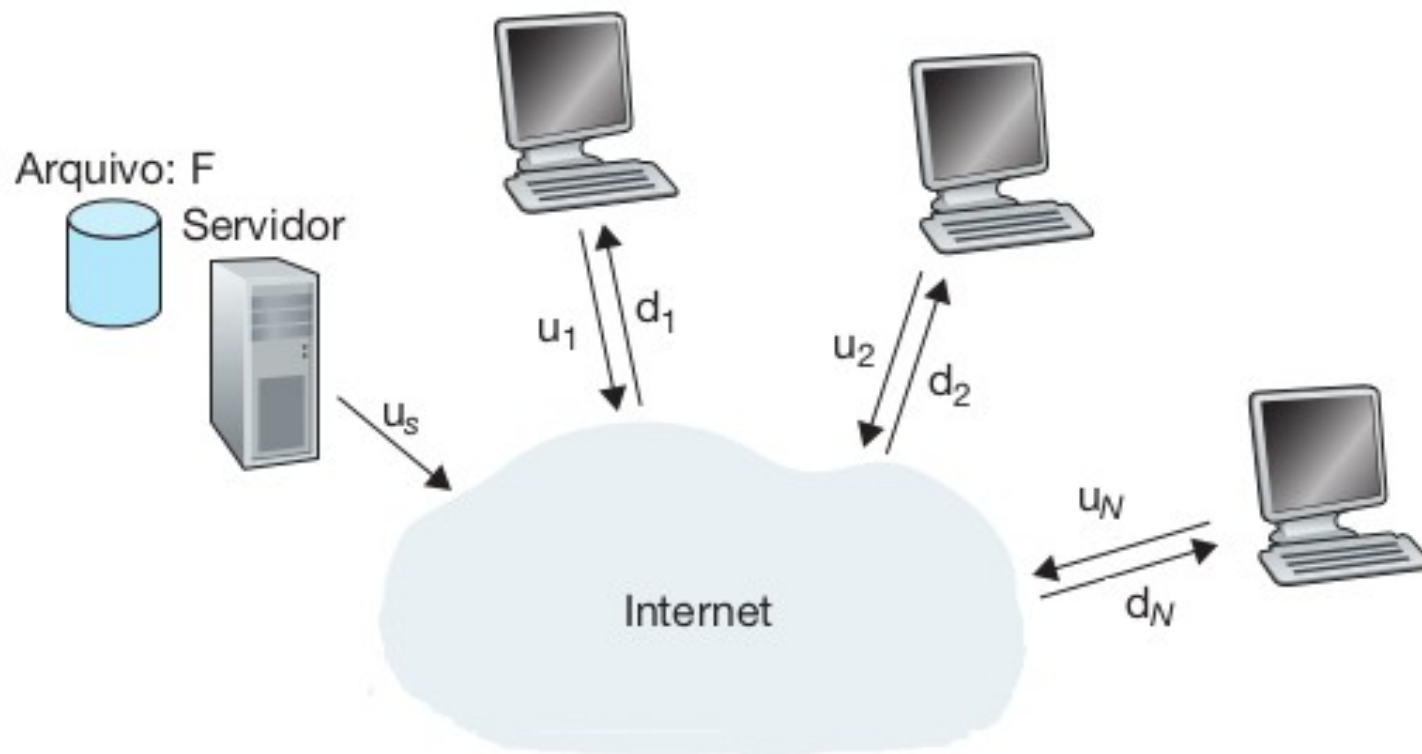
- “**modelo quantitativo**” ... **taxa de upload** do enlace de acesso do servidor é  $U_s$ , **upload** do enlace de acesso do par “ $i$ ” é  $U_i$  e **taxa de download** do enlace de acesso do par “ $i$ ” é  $D_i$ .



## 2 – Camada de Aplicação / 2.6 – Aplicações P2P

### ... 2.6.1 – Distribuição de Arquivos P2P

- ... onde **F** é o tamanho do arquivo a ser distribuído (em bits) e **N** representa o nro. de pares que recebem o arquivo.
- “**tempo de distribuição**” .. tempo necessário para que todos os pares recebam uma cópia do arquivo cujo tamanho é **F** bits.



## 2 – Camada de Aplicação / 2.6 – Aplicações P2P

### ... 2.6.1 – Distribuição de Arquivos P2P

- “**suposição**” .. núcleo da rede dispõe de largura de banda abundante, ou seja, todos os gargalos estão no acesso da rede.
- “**suposição**” .. servidor e clientes não participam de nenhuma outra rede, assim toda a largura de banda de “upload” e “download” é totalmente dedicada para a distribuição do arquivo.
- “**análise para cliente/servidor**” ... servidor deve transmitir  **$N * F$  bits** e, considerando que a taxa de “upload” é  **$Us$** , então o tempo para distribuição do arquivo deve ser de pelo menos  **$= N * F / Us$  seg.**
- Obs.: ... na arquitetura cliente/servidor nenhum dos pares auxilia na distribuição do arquivo, logo, todo o trabalho é do servidor.

## 2 – Camada de Aplicação / 2.6 – Aplicações P2P

### ... 2.6.1 – Distribuição de Arquivos P2P

- “**análise para cliente/servidor**” ... servidor deve transmitir  **$N * F$  bits** e, considerando que a taxa de “upload” é  **$U_s$** , então o tempo para distribuição do arquivo deve ser de pelo menos  **$= N * F / U_s$  seg.**
- ... seja  **$D_{min}$**  a taxa de download do par com menor taxa de download, ou seja,  **$D_{min} = \min\{ D_1, D_2, ..., D_n \}$** .
- ... par com a menor taxa de download irá obter o arquivo em  **$F / D_{min}$** , assim, o tempo de distribuição mínimo é de pelo menos  **$F / D_{min}$** .
- **$D_{cs}$  “greater than”  $\max\{ N * F / U_s ; F / D_{min} \}$**
- ... há um limite inferior para o tempo mínimo de distribuição para a Arquitura Cliente/Servidor – indicado por  **$D_{cs}$**  .

## 2 – Camada de Aplicação / 2.6 – Aplicações P2P

### ... 2.6.1 – Distribuição de Arquivos P2P

- $D_{cs}$  “greater than”  $\max\{ N * F / U_s ; F / D_{min} \}$
- ... há um limite inferior para o tempo mínimo de distribuição para a Arquitura Cliente/Servidor – indicado por  $D_{cs}$  .
- ... para um nro grande de servidores ou valor grande de  $N$ , o tempo de distribuição do arquivo será dado por  $N * F / U_s$  uma vez que a 2ª parcela “ $F / D_{min}$ ” é bem menor do que a 1ª parcela.
- “**conclusão**” ... isto significa que se o nro de pares crescer 1000, então o tempo para distribuir o arquivo para todos crescerá 1000 vezes.

## 2 – Camada de Aplicação / 2.6 – Aplicações P2P

### ... 2.6.1 – Distribuição de Arquivos P2P

- “**análise para peer-to-peer**” ... cada par pode ajudar no download de partes do arquivo para os demais pares tendo por base as parcelas do arquivo que cada par dispõe para download.
- .. primeiramente, servidor deve enviar cada bit do arquivo ao menos uma vez para o enlace de acesso, assim o tempo de distribuição mínimo é ao menos  **$F / U_s$  seg.**
- ... par com a menor taxa de “download” não pode obter todos os bits do arquivo em menos de  **$F / D_{\min}$  segundos** assim, tempo de distribuição mínimo é de pelo menos  **$F / D_{\min}$  segundos.**
- ... capacidade total de “upload” do sistema é a soma das taxas de “upload” do servidor e dos pares ..  **$U_s + U_1 + U_2 + \dots + U_n$**  logo, o tempo mínimo de distribuição é  **$N * F / (U_s + U_1 + U_2 + \dots + U_n)$ .**



## 2 – Camada de Aplicação / 2.6 – Aplicações P2P

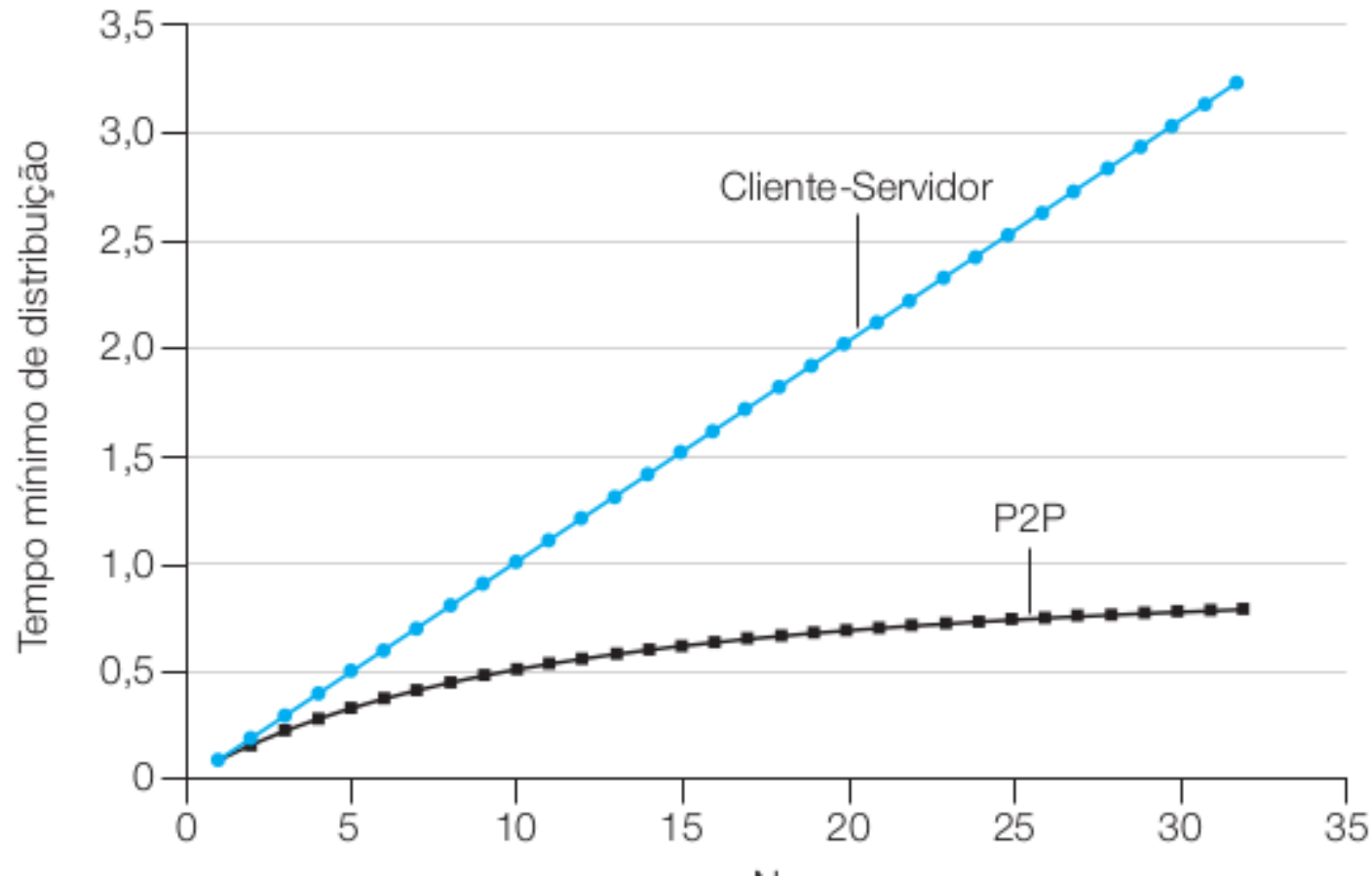
### ... 2.6.1 – Distribuição de Arquivos P2P

- **“análise para peer-to-peer”** ... cada par pode ajudar no “download” de partes do arquivo para os demais pares tendo por base as parcelas do arquivo que cada par dispõe para “download”.
- Juntando as três observações, obtem-se o mínimo para o tempo de distribuição, indicado por  $D_{P2P}$  (peer-to-peer)
- $D_{P2P}$  **“greater than”**
- $\max \{ F / U_s ; F / D_{min}; N * F / (U_s + U_1 + U_2 + \dots + U_n) \}$
- Obs.: ... equação fornece o limite inferior para o tempo mínimo de distribuição na arquitetura peer-to-peer se considerarmos que cada par redistribuirá um bit assim que o receber.

## 2 – Camada de Aplicação / 2.6 – Aplicações P2P

### ... 2.6.1 – Distribuição de Arquivos P2P

- “análise para peer-to-peer” ..  $F/U_s = 1$  hora;  $U_s = 10u$  e  $D_{\min} \geq U_s$   
assim um par pode transmitir o arquivo completo em 60 min, sendo que a taxa de transmissão do servidor é 10 \* “upload” do par.



## 2 – Camada de Aplicação / 2.6 – Aplicações P2P

### ... 2.6.1 – Distribuição de Arquivos P2P

- “**BitTorrent**” ... protocolo peer-to-peer popular para distribuição de arquivos, no qual, o “**torrent**” representa a coleção de todos os pares que participam da distribuição de um determinado arquivo.
- ... pares fazem o “download” de blocos do arquivo entre si iguais em tamanho, com um tamanho típico de bloco igual a 256 KBytes.
- ... qualquer par pode sair do torrent a qualquer momento com apenas um subconjunto de blocos, e depois voltar.

## 2 – Camada de Aplicação / 2.6 – Aplicações P2P

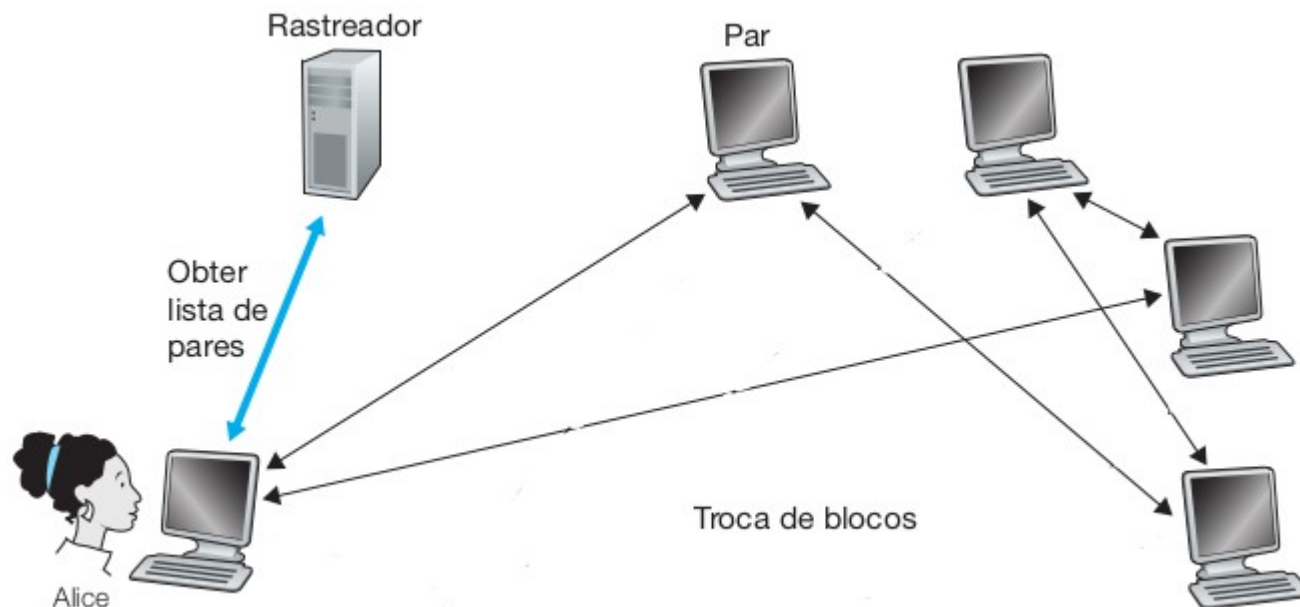
### ... 2.6.1 – Distribuição de Arquivos P2P

- “**idéia**” .. apresentar os mecanismos mais importantes do protocolo bittorrent sem entrar no mérito dos detalhes do mesmo:
- ... cada torrent tem um nó de infraestrutura chamado **rastreador** que mantém um registro dos pares que participam de cada torrent.
- ... um determinado torrent pode ter menos de 10 ou mais de 1000 pares participando a qualquer momento.
- ... quando um par chega em um torrent, ele se registra com o rastreador e periodicamente informa ao rastreador que ainda está lá.

## 2 – Camada de Aplicação / 2.6 – Aplicações P2P

### ... 2.6.1 – Distribuição de Arquivos P2P

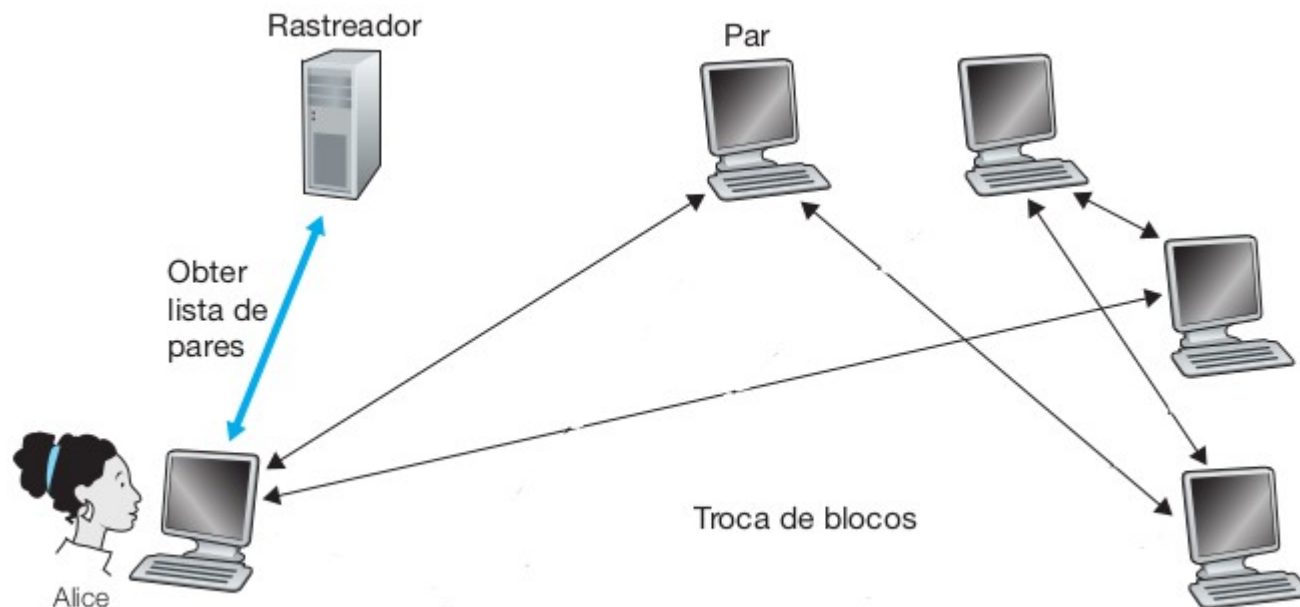
- Quando um novo par (Alice) chega, o rastreador seleciona aleatoriamente um subconjunto de pares (p.ex., 50 pares) do conjunto de pares participantes e envia os seus endereços IP.
- ... com a lista de pares, Alice tenta estabelecer conexões TCP simultâneas com todos (aqui denominados de pares vizinhos).



## 2 – Camada de Aplicação / 2.6 – Aplicações P2P

### ... 2.6.1 – Distribuição de Arquivos P2P

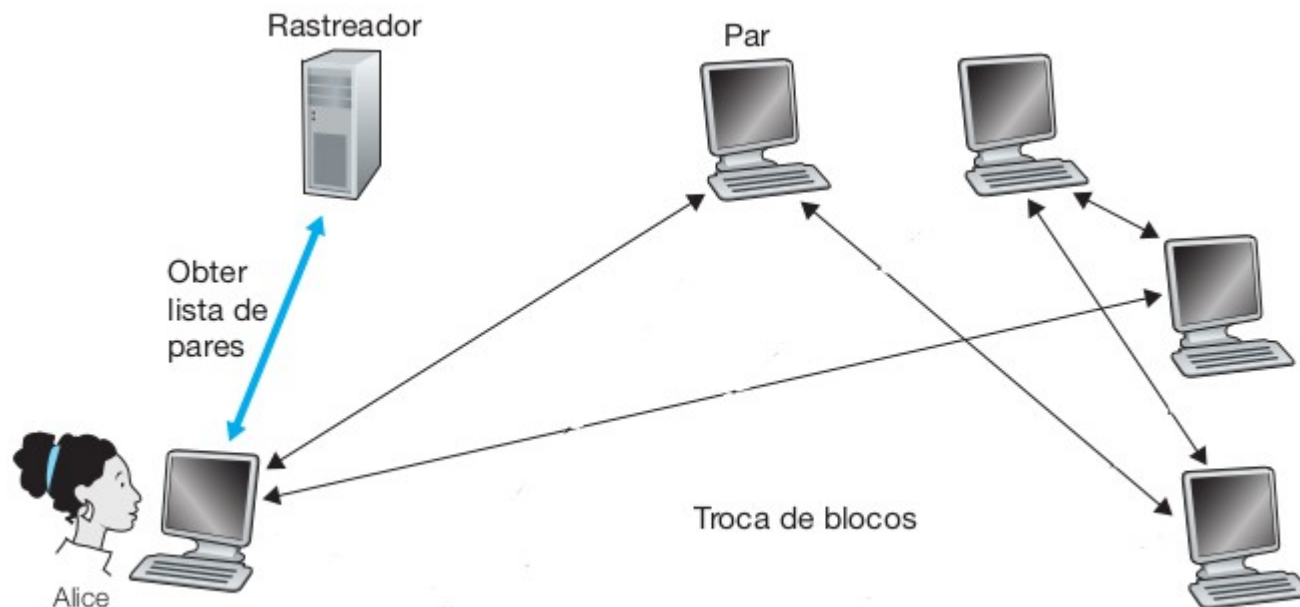
- Com o tempo, alguns desses pares podem sair e outros (fora dos 50 iniciais) podem tentar estabelecer conexões TCP com Alice.
- ... portanto, os pares vizinhos de um par podem flutuar com o tempo.



## 2 – Camada de Aplicação / 2.6 – Aplicações P2P

### ... 2.6.1 – Distribuição de Arquivos P2P

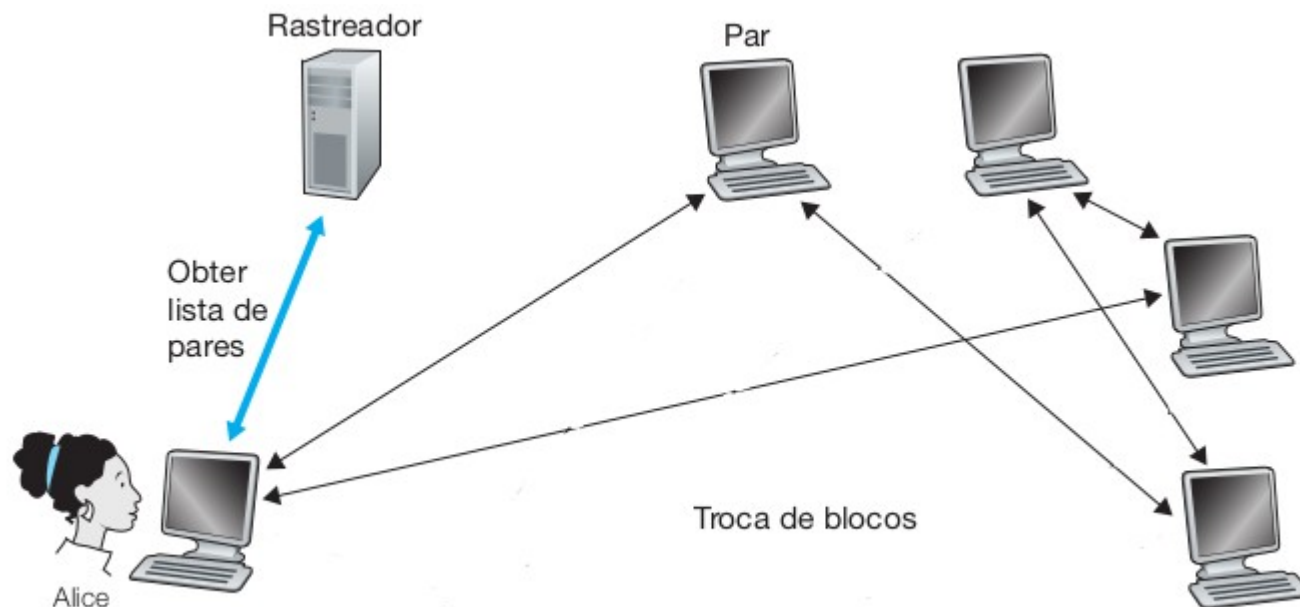
- .. em qualquer momento, cada par tem um subconjunto de blocos do arquivo, com pares diferentes com sub-conjuntos diferentes.
- .. caso Alice tenha **L** vizinhos diferentes, ela obterá **L** listas de blocos, o que possibilita que Alice lance solicitações (novamente, nas conexões TCP) de blocos que ela não tem.



## 2 – Camada de Aplicação / 2.6 – Aplicações P2P

### ... 2.6.1 – Distribuição de Arquivos P2P

- **“rarest first”** .. ao decidir quais blocos solicitar, Alice usa a técnica do bloco mais raro primeiro, ou seja, dentre os blocos que ela não tem, quais são os mais raros dentre os seus vizinhos;
- ... que é determinada contabilizando os blocos com o menor número de cópias repetidas na lista de vizinhos diferentes.

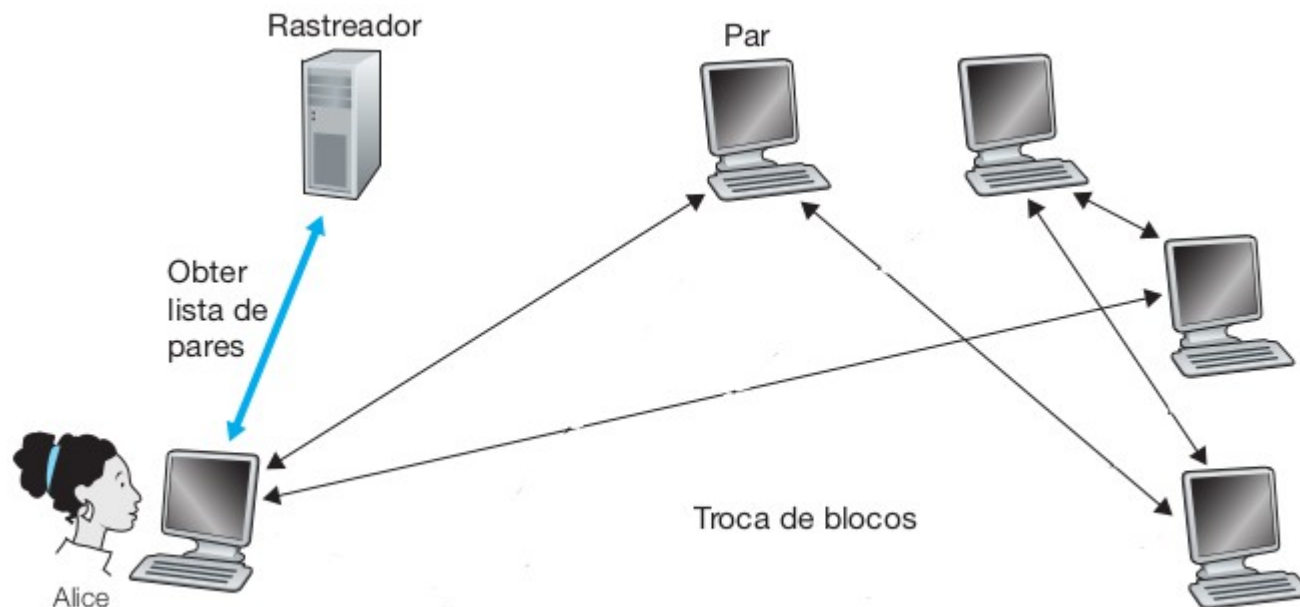




## 2 – Camada de Aplicação / 2.6 – Aplicações P2P

### ... 2.6.1 – Distribuição de Arquivos P2P

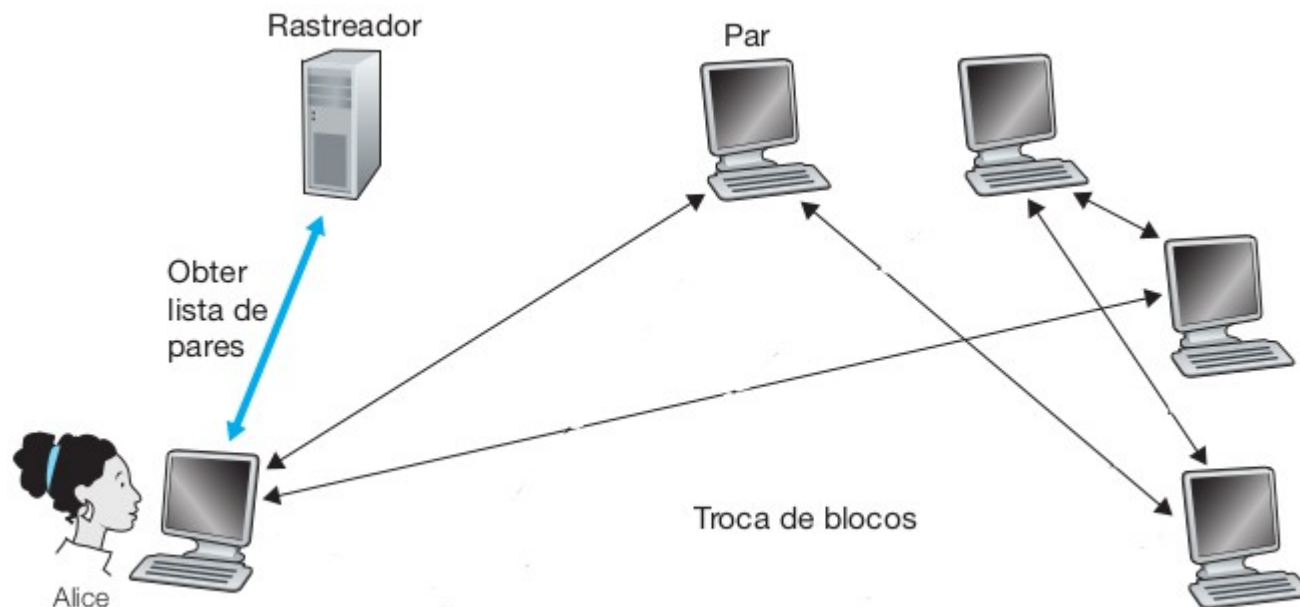
- “**rarest first**” .. ao decidir quais blocos solicitar, Alice usa a técnica do bloco mais raro primeiro, ou seja, dentre os blocos que ela não tem, quais são os mais raros dentre seus vizinhos.
- ... dessa forma, os blocos mais raros são redistribuídos mais depressa, procurando equalizar os números de cópias de cada bloco no torrent.



## 2 – Camada de Aplicação / 2.6 – Aplicações P2P

### ... 2.6.1 – Distribuição de Arquivos P2P

- “**quais pedidos atender**” .. BitTorrent usa um algoritmo de troca inteligente onde a ideia básica é Alice dar prioridade aos vizinhos que estejam fornecendo seus dados com a maior taxa.



## 2 – Camada de Aplicação / 2.6 – Aplicações P2P

### ... 2.6.1 – Distribuição de Arquivos P2P

- BitTorrent ... muito bem-sucedido, com milhões de pares simultâneos compartilhando arquivos ativamente em 100s de 1000s de torrents.
- Variantes do Protocolo BitTorrent [Guo, 2005; Piatek, 2007].
- Além disso, muitas das aplicações de transmissão em tempo real P2P, como PPLive e PPStream, foram inspiradas pelo BitTorrent [Hei, 2007].
- “PPLive” .. programa utilizado para a transmissão de canais de TV na China, que cumpre requisitos de tempo real que as aplicações de transmissão de vídeo ao vivo exigem.
- .. entre as soluções empregadas para atender aos requisitos de atraso do vídeo ao vivo, estão a implementação de 02 (dois) buffers para armazenar os pedaços de vídeo recebidos.

## 2 – Camada de Aplicação / 2.6 – Aplicações P2P

### 2.6.2 – Distributed Hash Table (DHT)

- “**objetivo**” .. conceber um banco de dados distribuído que contemple pares do tipo “chave” e “valor” e que possibilite a comunicação dos pares segundo a arquitetura peer-to-peer.
- e.g., se o banco de dados armazenar os nomes de conteúdos e seus endereços IPs, ao consultarmos um nome de conteúdo, o banco de dados retorna o endereço IP que armazena aquele conteúdo.
- “**constatação**” .. no sistema peer-to-peer cada par mantém uma parte do total de pares (chave, valor), permitindo que qualquer par consulte o banco de dados distribuído com uma chave em particular.

## 2 – Camada de Aplicação / 2.6 – Aplicações P2P

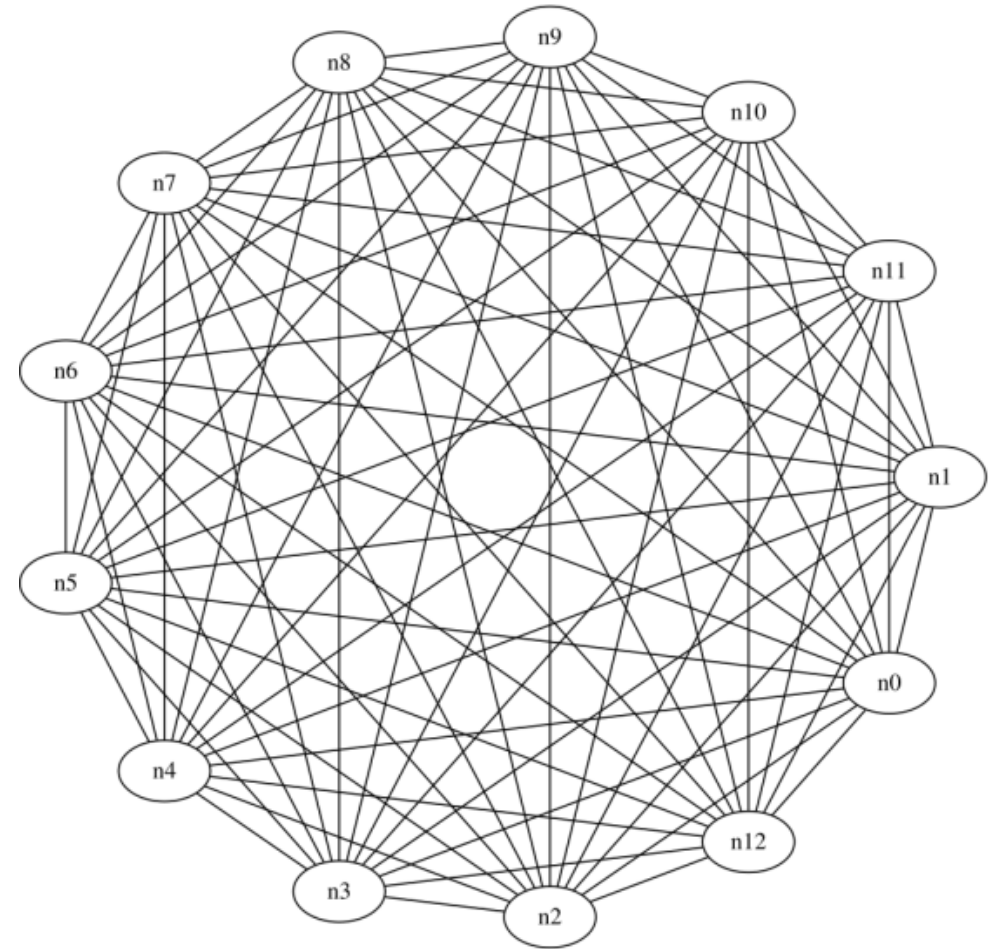
### ... 2.6.2 – Distributed Hash Table (DHT)

- “**constatação**” .. no sistema peer-to-peer cada par mantém uma parte do total de pares (chave, valor), permitindo que qualquer par consulte o banco de dados distribuído com uma chave em particular.
- ... banco de dados distribuído, então, localiza os pares que possuem os pares (chave, valor) correspondentes e retorna os pares (chave, valor) ao remetente ou solicitante da consulta.
- .. qualquer peer poderá inserir pares (chave, valor) no banco de dados, que como já percebido é uma tabela “hash” distribuída (DHT).

## 2 – Camada de Aplicação / 2.6 – Aplicações P2P

### ... 2.6.2 – Distributed Hash Table (DHT)

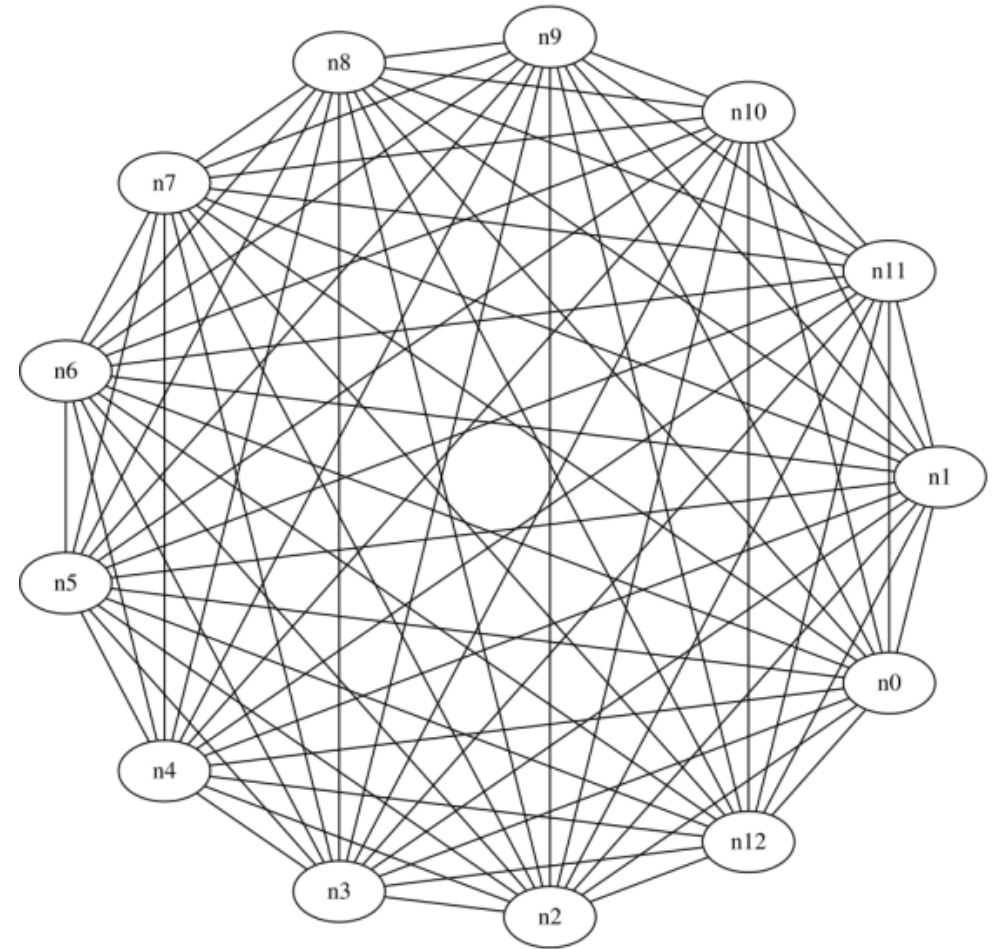
- “**organização dos pares**” .. espalhar ao acaso as duplas (chave, valor) por todos os pares e fazer com que cada um mantenha uma lista dos IPs de todos os pares.
- .. par solicitante da consulta envia sua consulta a todos os pares, e aqueles contendo as duplas (chave, valor) que combinam com a chave podem responder com suas duplas correspondentes.



## 2 – Camada de Aplicação / 2.6 – Aplicações P2P

### ... 2.6.2 – Distributed Hash Table (DHT)

- .. par solicitante da consulta envia sua consulta a todos os pares, e aqueles contendo as duplas (chave, valor) que combinam com a chave podem responder com suas duplas correspondentes.
- “**não escalável**” .. exige que cada par não apenas conheça todos os outros pares (talvez 100s de 1000s), mas consulta deve ser feita a todos os pares da malha de pares.



## 2 – Camada de Aplicação / 2.6 – Aplicações P2P

### ... 2.6.2 – Distributed Hash Table (DHT)

- “**alternativa**” .. designar identificador para cada par onde cada identificador é um número inteiro na faixa  $[0, 2^{(n-1)}]$  de algum “n” fixo.
- .. cada identificador pode ser expresso por “n” bits.
- .. cada chave é um número inteiro na mesma faixa  $[0, 2^{(n-1)}]$ , o que pode exigir algum mapeamento do objeto de busca (p.ex., nros. de seguridade social e nomes de conteúdo) para nros. inteiros.
- .. para criar nros inteiros a partir do objeto de busca, necessário usar uma função hash que mapeie cada chave (p.ex., nro de seguridade social) em um número inteiro na faixa  $[0, 2^{(n-1)}]$ .



## 2 – Camada de Aplicação / 2.6 – Aplicações P2P

### ... 2.6.2 – Distributed Hash Table (DHT)

- **“função de hash”** .. função de muitos-para-um para a qual 02 entradas diferentes podem ter a mesma saída (mesmo nro. inteiro), mas a probabilidade de terem a mesma saída é extremamente pequena.
- ... função de hash deve ser pública para todos os pares e, adicionalmente, quando se refere a “chave”, a referência é para o “hash” da chave original (objeto de busca que foi mapeado).
- p.ex., caso a chave original seja “Led Zeppelin IV”, a chave usada no DHT será o nro. inteiro que corresponda ao hash de “Led Zeppelin IV”.

## 2 – Camada de Aplicação / 2.6 – Aplicações P2P

### ... 2.6.2 – Distributed Hash Table (DHT)

- “**problema**” .. armazenar as duplas (chave, valor) na DHT pode ser resolvida definindo-se uma regra para designar chaves aos pares.
- “**premissa**” .. cada par tem um identificador inteiro e cada chave contempla um nro. inteiro na mesma faixa.
- “**solução**” .. designar cada dupla (chave, valor) ao par cujo identificador está mais próximo da chave objeto da busca.

## 2 – Camada de Aplicação / 2.6 – Aplicações P2P

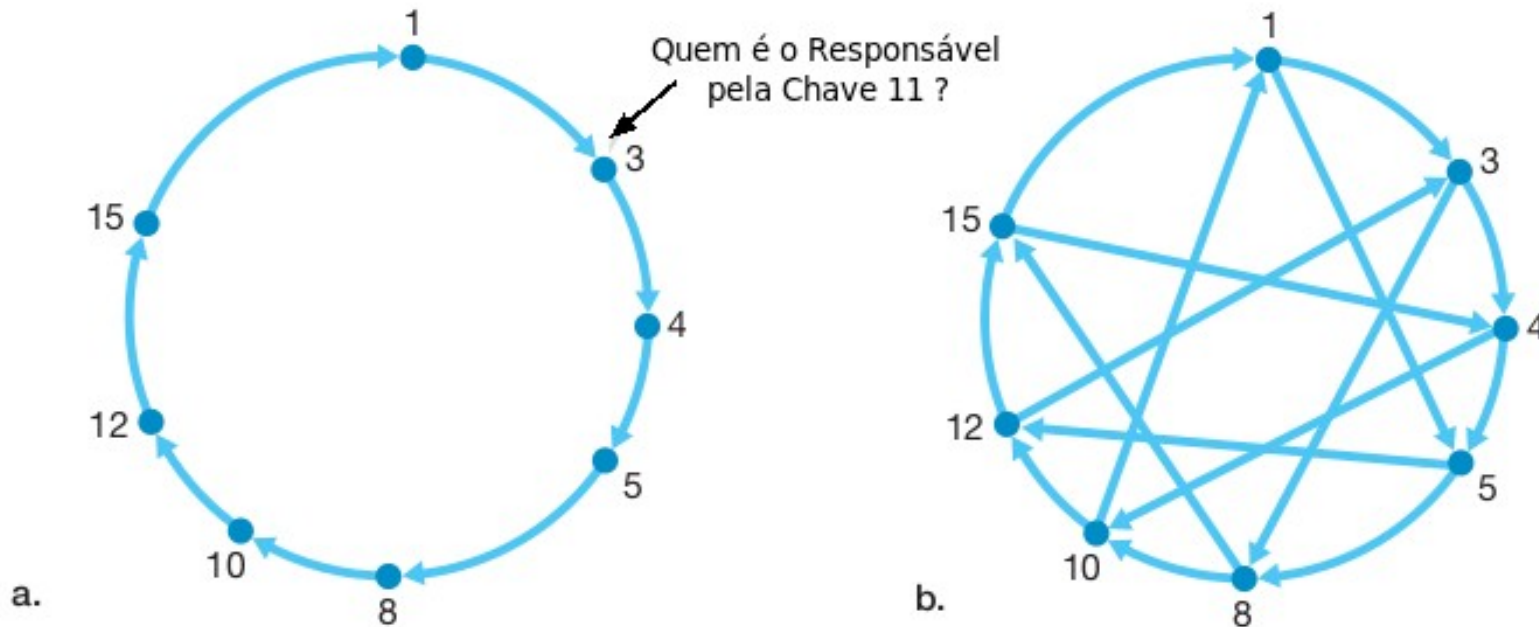
### ... 2.6.2 – Distributed Hash Table (DHT)

- “**solução**” .. designar cada dupla (chave, valor) ao par cujo identificador está mais próximo da chave objeto da busca.
- e.g., Considere que  $n = 4$  bits, portanto, todos os identificadores de par e chave estão na faixa de  $[0, 15]$ .
- “**suposição**” .. tem-se 08 pares no sistema cujos identificadores são 01, 03, 04, 05, 08, 10, 12 e 15.
- “**objetivo**” .. armazenar a dupla, p.ex.,  $(\text{chave, valor}) = (11, \text{Johnny Wu})$  em um dos 08 pares de nós que podem armazenar duplas.

## 2 – Camada de Aplicação / 2.6 – Aplicações P2P

### ... 2.6.2 – Distributed Hash Table (DHT)

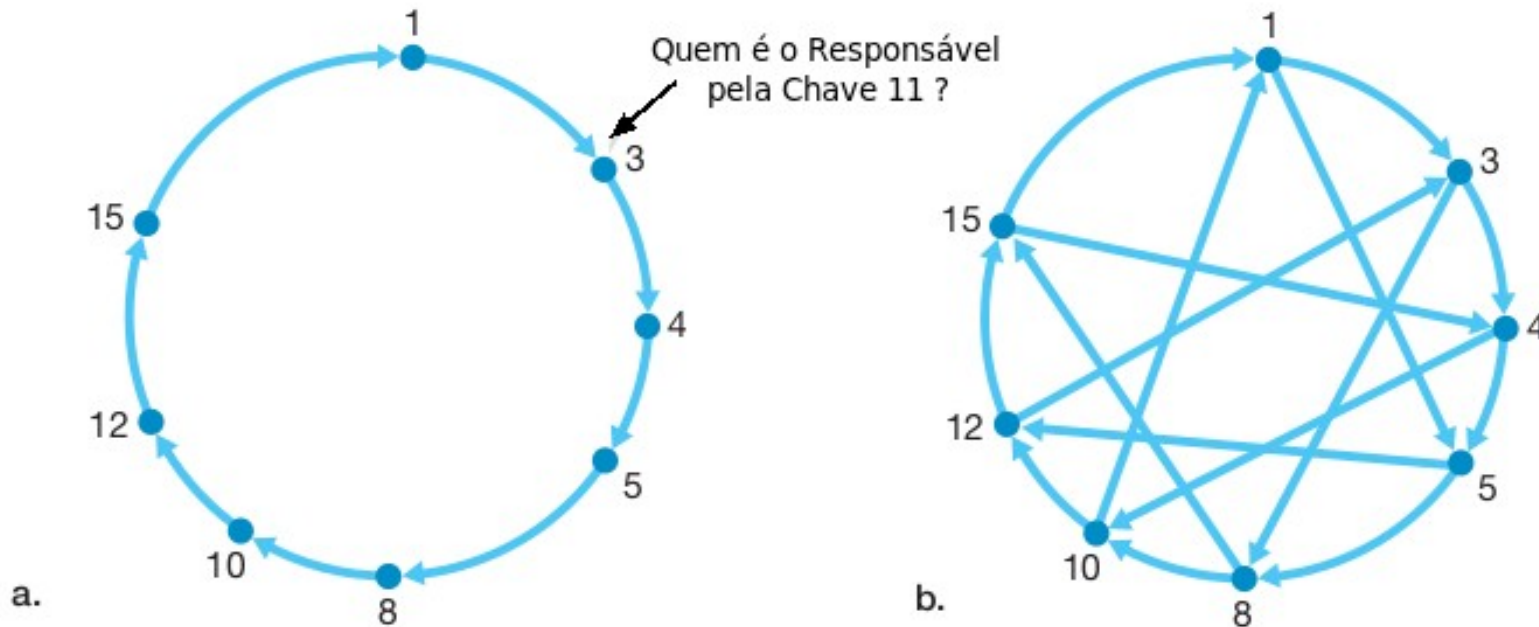
- e.g., Considere que  $n = 4$  bits, portanto, todos os identificadores de par e chave estarão na faixa de  $[0, 15]$ . Deseja-se armazenar a dupla chave-valor (11, Johnny Wu) em um dos 08 pares.
- “**suposição**” .. tem-se 08 pares no sistema cujos identificadores são 01, 03, 04, 05, 08, 10, 12 e 15.



## 2 – Camada de Aplicação / 2.6 – Aplicações P2P

### ... 2.6.2 – Distributed Hash Table (DHT)

- “**problema**” .. qual peer deve-se armazenar a dupla (11, Johnny Wu) ??
- “**convenção do mais próximo**” .. como o par 12 é sucessor imediato da chave 11, pode-se armazenar dupla (11, Johnny Wu) no par 12.



## 2 – Camada de Aplicação / 2.6 – Aplicações P2P

### ... 2.6.2 – Distributed Hash Table (DHT)

- **“problemas da convenção do mais próximo”** .. alguns casos:
- ... para concluir nossa definição de mais próximo, caso a chave seja idêntica a um dos identificadores do par, armazena-se a dupla (chave, valor) no par correspondente.
- ... caso seja maior do que todos os identificadores de par, usaremos uma convenção módulo-2 de “n”, que armazena a dupla (chave, valor) no par com o menor identificador.

## 2 – Camada de Aplicação / 2.6 – Aplicações P2P

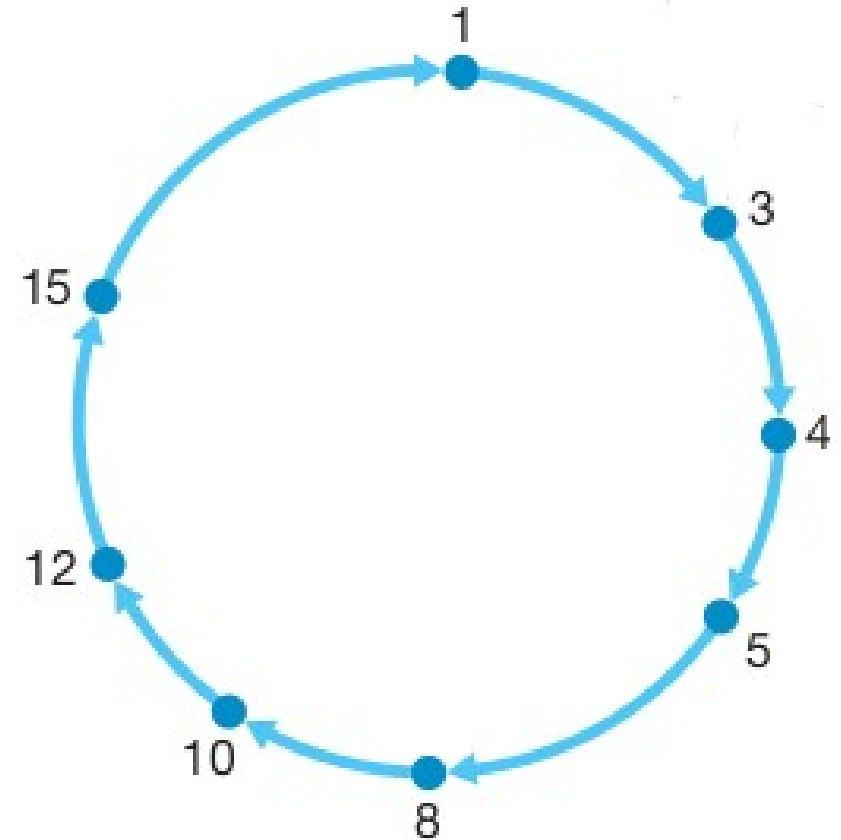
### ... 2.6.2 – Distributed Hash Table (DHT)

- “**dúvida**” .. que convenção usar para inserir (chave, valor) na DHT ?
- e.g., Seja a inserção de uma dupla (chave-valor) no DHT.
- .. primeiro determina-se o par cujo identificador é o mais próximo da chave e, na sequência, envia uma msg. ao par, instruindo-o a armazenar a dupla (chave, valor).
- “**dúvida**” .. como determinar o par mais próximo da chave ?
- “**solução**” .. se for possível rastrear todos os pares no sistema (IDs de par e IPs correspondentes), pode-se determinar localmente o par mais próximo, mas há um porém nesta abordagem !!
- ... cada par rastreia todos os outros pares, algo impraticável para um sistema de grande escala com 100s de 1000s de pares.

## 2 – Camada de Aplicação / 2.6 – Aplicações P2P

### ... 2.6.2 – Distributed Hash Table (DHT)

- DHT Circular .. considera-se a organização dos pares em círculo, onde cada par rastreia seu sucessor e predecessor imediatos (módulo  $2^n$  ).
- e.g., “n” = 4 bits e todos os identificadores de par e chave estão na faixa de [0, 1, 2, 3, ..., 15].
- ... cada par está ciente apenas de seu sucessor e predecessor imediatos, p.ex, par 5 conhece o Identificador e IP do par 8 e do par 4.

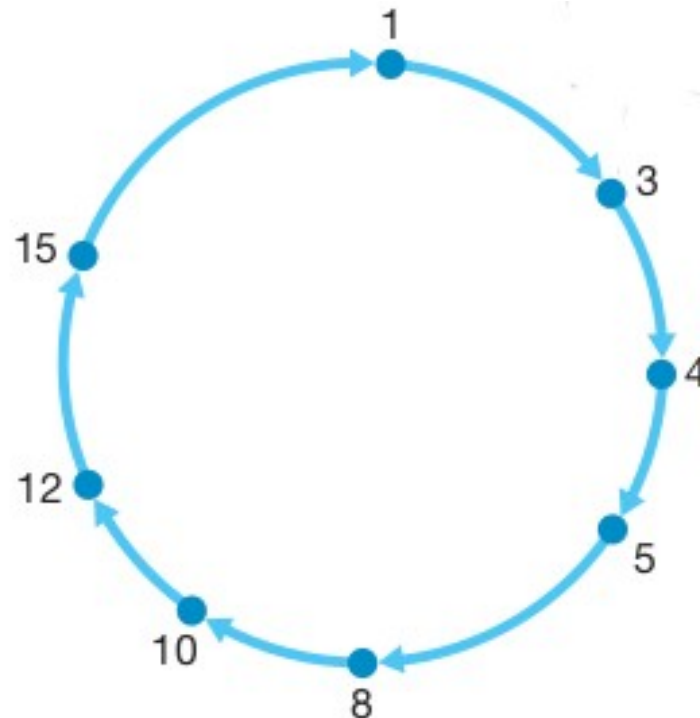




## 2 – Camada de Aplicação / 2.6 – Aplicações P2P

### ... 2.6.2 – Distributed Hash Table (DHT)

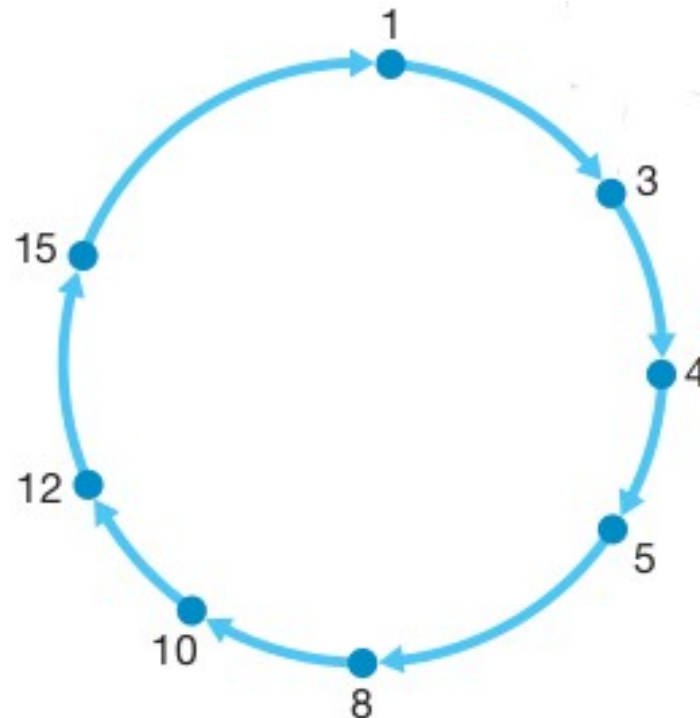
- “**rede de sobreposição**” .. pares formam uma rede lógica abstrata que reside acima da topologia física da rede de computadores e que consiste de enlaces físicos, roteadores e “hosts”.
- e.g., considere a rede de sobreposição circular e suponha que o par 3 deseje determinar qual par no DHT é responsável pela chave 11.



## 2 – Camada de Aplicação / 2.6 – Aplicações P2P

### ... 2.6.2 – Distributed Hash Table (DHT)

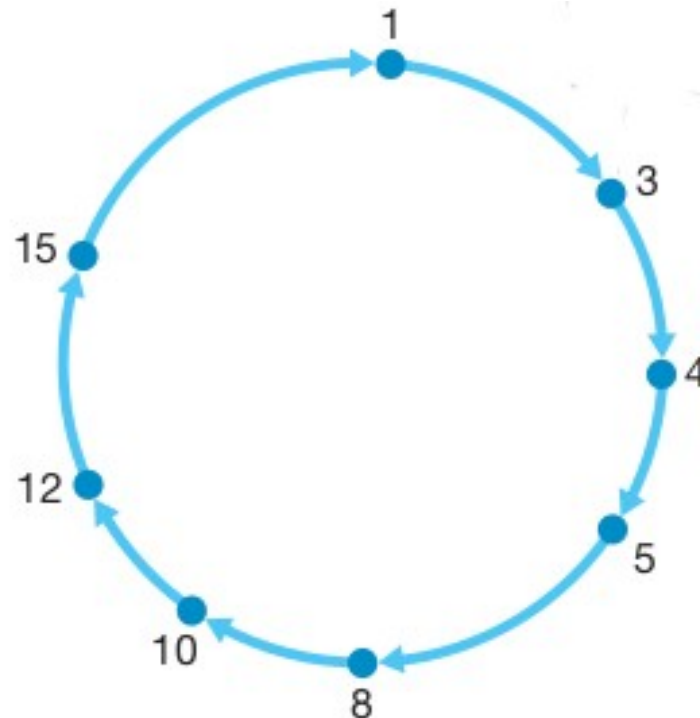
- e.g., considere a rede de sobreposição circular e suponha que o par 3 deseje determinar qual par no DHT é responsável pela chave 11.
- ... usando a rede de sobreposição circular, o par de origem (#3) cria uma mensagem com a pergunta “Quem é responsável pela chave 11?” e a envia no sentido horário ao redor do círculo.



## 2 – Camada de Aplicação / 2.6 – Aplicações P2P

### ... 2.6.2 – Distributed Hash Table (DHT)

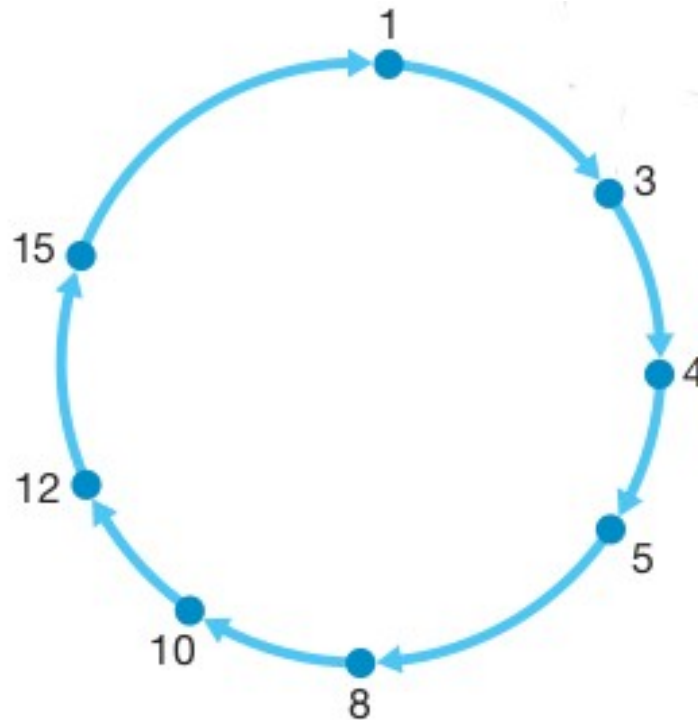
- e.g., considere a rede de sobreposição circular e suponha que o par 3 deseje determinar qual par no DHT é responsável pela chave 11.
- ... sempre que um par recebe essa mensagem, como conhece o identificador de seu sucessor e predecessor, pode determinar se é responsável pela chave em questão.



## 2 – Camada de Aplicação / 2.6 – Aplicações P2P

### ... 2.6.2 – Distributed Hash Table (DHT)

- e.g., considere a rede de sobreposição circular e suponha que o par 3 deseje determinar qual par no DHT é responsável pela chave 11.
- ... caso um par não seja responsável pela chave, ele apenas envia a mensagem a seu sucessor, até encontrar pelo par responsável pela (chave, valor), que pode na sequência responder a origem.



## 2 – Camada de Aplicação / 2.6 – Aplicações P2P

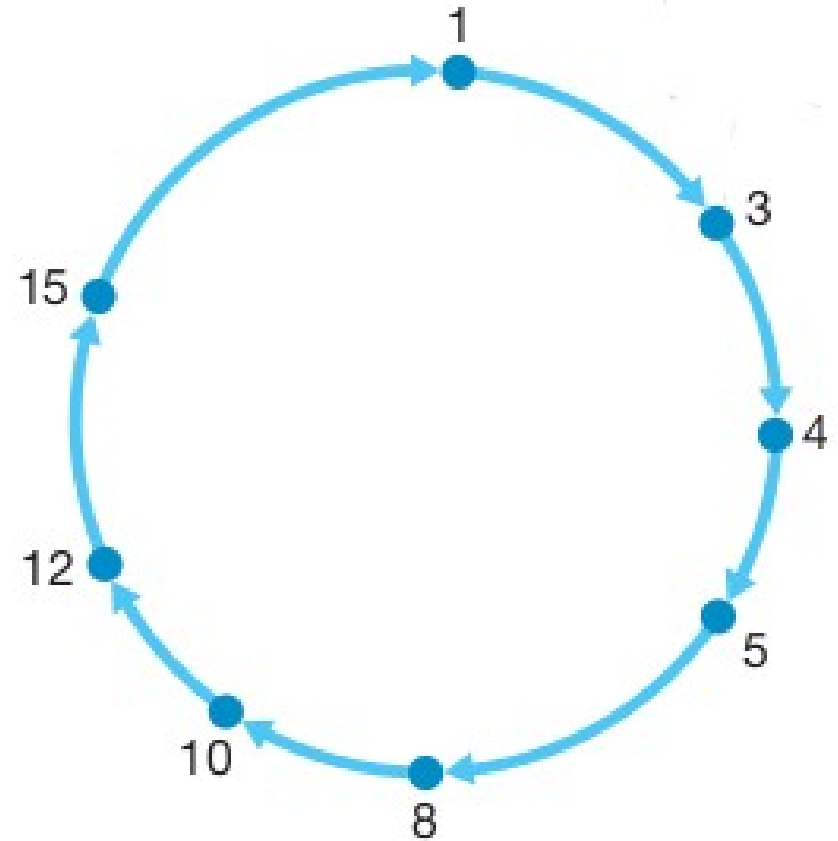
### ... 2.6.2 – Distributed Hash Table (DHT)

- e.g., considere a rede de sobreposição circular e suponha que o par 3 deseje determinar qual par no DHT é responsável pela chave 11.
- DHT Circular oferece uma solução elegante para reduzir a quantidade de informação sobreposta que cada par deve gerenciar, p.ex., percepção do sucessor e predecessor imediatos.

## 2 – Camada de Aplicação / 2.6 – Aplicações P2P

### ... 2.6.2 – Distributed Hash Table (DHT)

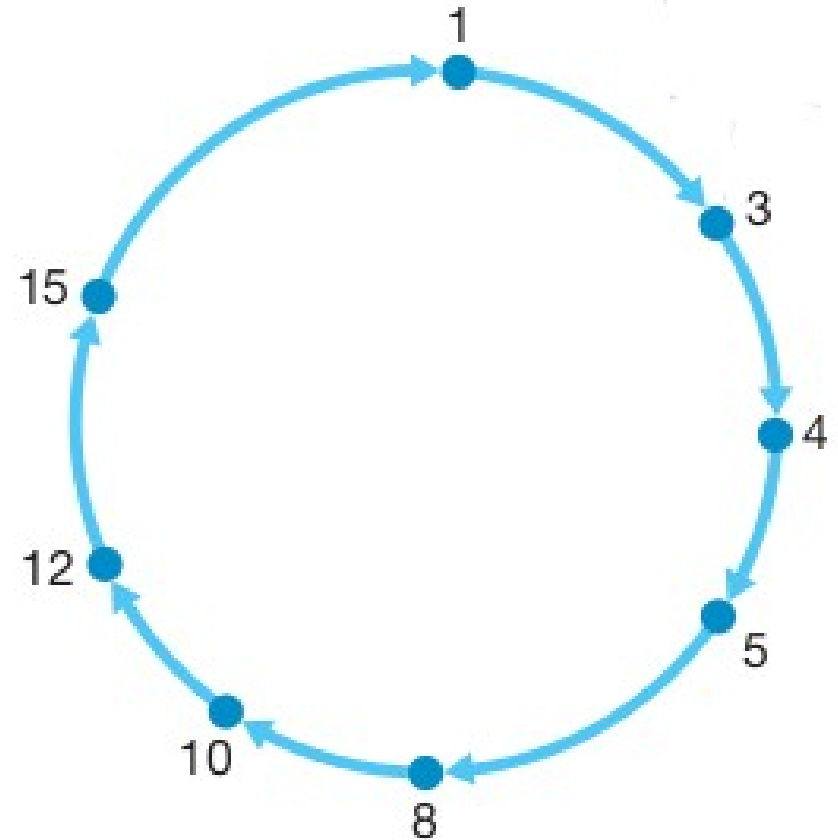
- “Peer Churn” .. um par pode vir ou ir sem aviso, logo, é necessário se preocupar em manter a DHT na presença da rotatividade dos pares que compõem a DHT.
- e.g., “n” = 4 bits e todos os identificadores de par e chave estão na faixa de [0, 15], onde cada peer conhece o sucessor e predecessor.
- .. na presença de “peer churn” o problema pode ser amenizado exigindo-se que cada par conheça ou rastreie 02 sucessores.



## 2 – Camada de Aplicação / 2.6 – Aplicações P2P

### ... 2.6.2 – Distributed Hash Table (DHT)

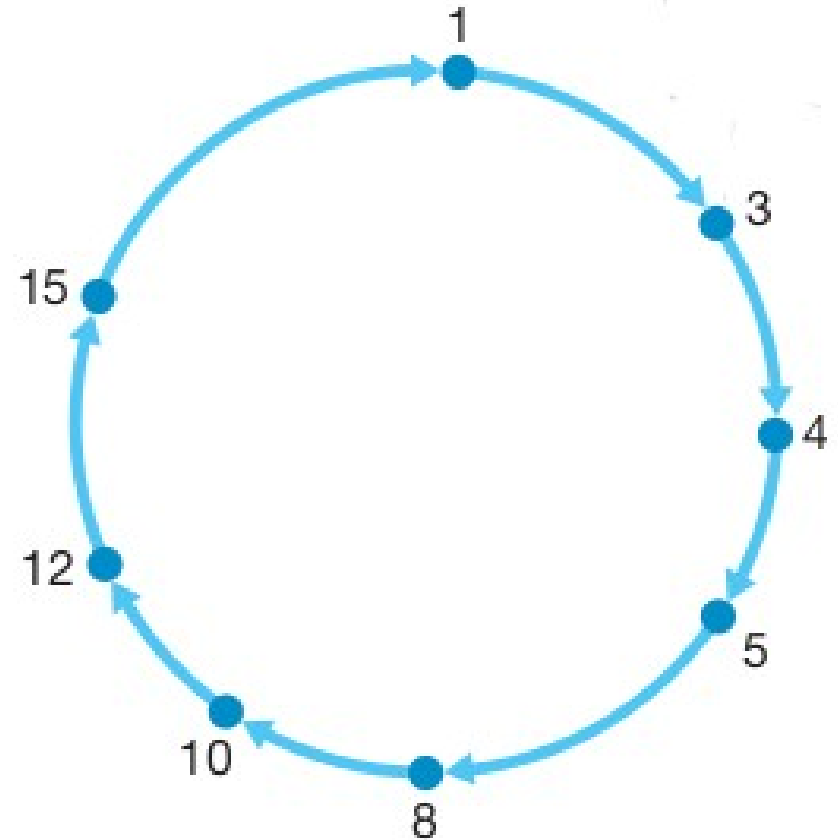
- e.g., “n” = 4 bits e todos os identificadores de par e chave estão na faixa de [0, 15], onde cada peer conhece o sucessor e predecessor.
- “**pergunta**” .. como inserir um par cujo identificador é 13 ??
- “**premissa**” .. par 13 conhece a existência do par 01 !!
- .. par 13 envia ao par 1 uma msg. perguntando “Quem é o predecessor e quem é o sucessor do par 13 ?”



## 2 – Camada de Aplicação / 2.6 – Aplicações P2P

### ... 2.6.2 – Distributed Hash Table (DHT)

- ... msg. é encaminhada pelo DHT até alcançar o par 12, que percebe que será o predecessor do par 13.
- .. na sequência o par 12 envia as informações de sucessor e predecessor ao par 13.
- .. par 13 se inseri na DHT, tomando o par 15 como seu sucessor e notificando ao par 12 que deve mudar seu sucessor imediato para 13.





## 2 – Camada de Aplicação / 2.7 – Sockets UDP e TCP

### 2.7 – Programação de Sockets UDP e TCP

- “**objetivo**” .. avaliar como são escritos os programas de aplicação, ou seja, par de programas, seja cliente/servidor ou peer-to-peer.
- Há dois tipos de Aplicações de Rede.
  - 1) ... protocolo de comunicação normalmente denominado “**aberto**” e especificado por RFC ou algum outro documento padrão.
  - 2) ... protocolo **proprietário**, ou seja, protocolo de aplicação que não foi publicado abertamente em uma RFC ou em outro lugar.
- “**protocolos abertos**” .. devem obedecer às regras ditadas pela RFC para manter a conformidade com a especificação e, desta forma, garantir a interoperabilidade com outras implementações.

## 2 – Camada de Aplicação / 2.7 – Sockets UDP e TCP

### ... 2.7 – Programação de Sockets UDP e TCP

- e.g., seja um programa cliente com uma implementação do lado cliente do protocolo FTP [RFC 959] e o programa servidor com uma implementação do servidor do protocolo FTP [RFC 959].
- e.g., muitas aplicações de rede que envolvem comunicação entre cliente e servidor são concebidas por programadores diferentes.
- ... neste contexto, o navegador Firefox (cliente) que se comunica com um servidor Web Apache (servidor), ou um cliente BitTorrent (cliente) que se comunica com um rastreador BitTorrent (servidor).
- .. para garantir a interoperabilidade dessas implementações é necessário garantir a conformidade das implementações com as especificações dos protocolos por meio das RFCs.

## 2 – Camada de Aplicação / 2.7 – Sockets UDP e TCP

### ... 2.7 – Programação de Sockets UDP e TCP

- “**objetivo**” .. examinar questões fundamentais do desenvolvimento de uma aplicação cliente-servidor **examinando-se** o código de uma aplicação “simples” segundo a arquitetura cliente-servidor.
- “**fase de desenvolvimento**” .. uma das primeiras decisões é a escolha do serviço a ser utilizado da camada de transporte.
- TCP .. orientado a conexão e provê um canal confiável de cadeia de bytes, pelo qual fluem dados entre dois sistemas finais.
- UDP .. não orientado a conexão e provê envio de segmentos de dados de forma independente do sistema final e sem garantia de entrega.

## 2 – Camada de Aplicação / 2.7 – Sockets UDP e TCP

### ... 2.7 – Programação de Sockets UDP e TCP

- “**objetivo**” .. examinar questões fundamentais do desenvolvimento de uma aplicação cliente-servidor **examinando-se** o código de uma aplicação “simples” segundo a arquitetura cliente-servidor.
- ... apresenta-se a programação de sockets UDP e TCP por meio de aplicações UDP e TCP em “**python**”.
- ... opção por “python” se deu pela exposição clara dos principais conceitos contemplados nos sockets (combinação “port” e IP).

## 2 – Camada de Aplicação / 2.7 – Sockets TCP e UDP

### 2.7.1 – Programação de Sockets com UDP

- **“troca de mensagens”** .. processos que rodam em máquinas diferentes comunicam entre si enviando mensagens por sockets.
  - **“objetivo”** .. ver os detalhes da interação entre 02 processos que se comunicam através de sockets UDP.
- 
- 1) .. antes de encaminhar o pacote propriamente dito, incluir o endereço de destino para que seja possível inserir o pacote pelo socket.
  - 2) .. depois que o pacote passa pelo socket do emissor, a rede usa o endereço de destino para rotear o pacote até o socket do receptor.
  - 3) .. ao chegar no socket receptor, o processo receptor recolhe o pacote no socket e inspeciona o conteúdo do pacote e toma alguma ação.

## 2 – Camada de Aplicação / 2.7 – Sockets TCP e UDP

### ... 2.7.1 – Programação de Sockets com UDP

- “**pergunta**” .. o que há no endereço de destino acrescentado ao pacote ?
- .. ao incluir o endereço IP do destino no pacote, os roteadores na rede podem rotear o pacote até o “host” de destino, mas não ao processo !!
- “**pergunta**” .. como o hospedeiro pode executar tantos processos de aplicação de rede, cada um com um ou mais sockets, sem que mensagens de um processo sejam recebidas por outros processos ?
- .. é necessário identificar o socket no “host” de destino, ou seja, identificar qual processo em um dado “host” !
- ... opção por “**python**” se deu pela exposição clara dos principais conceitos contemplados nos sockets (combinação port e IP).

## 2 – Camada de Aplicação / 2.7 – Sockets TCP e UDP

### ... 2.7.1 – Programação de Sockets com UDP

- “**resumo**” .. processo emissor inclui no pacote um endereço de destino que consiste no endereço IP do “host” destino e o nro da porta destino.
- .. adicionalmente, o endereço de origem do emissor consiste de um endereço IP do “host” de origem e o nro de porta de origem.
- “**aplicação exemplo**” .. funcionalidades da aplicação para demonstrar a programação de socket com UDP e TCP:
  - 1) cliente lê caracteres do teclado e envia para o servidor;
  - 2) servidor recebe os dados e os converte em maiúsculas;
  - 3) servidor envia os dados modificados ao cliente;
  - 4) cliente recebe os dados modificados e apresenta na tela.

## 2 – Camada de Aplicação / 2.7 – Sockets TCP e UDP

### ... 2.7.1 – Programação de Sockets com UDP

- “**figura**” .. destaca as operações de socket realizadas pelo cliente e servidor, que se comunicam por meio de um serviço de transporte.
- “Cliente UDP” .. envia uma msg. simples ao servidor que utiliza o “port” 12000 para o número de porta do servidor de aplicação.
- ... para que o servidor receba a msg. do cliente, é necessário que esteja executando como um processo antes que o cliente envie a msg.

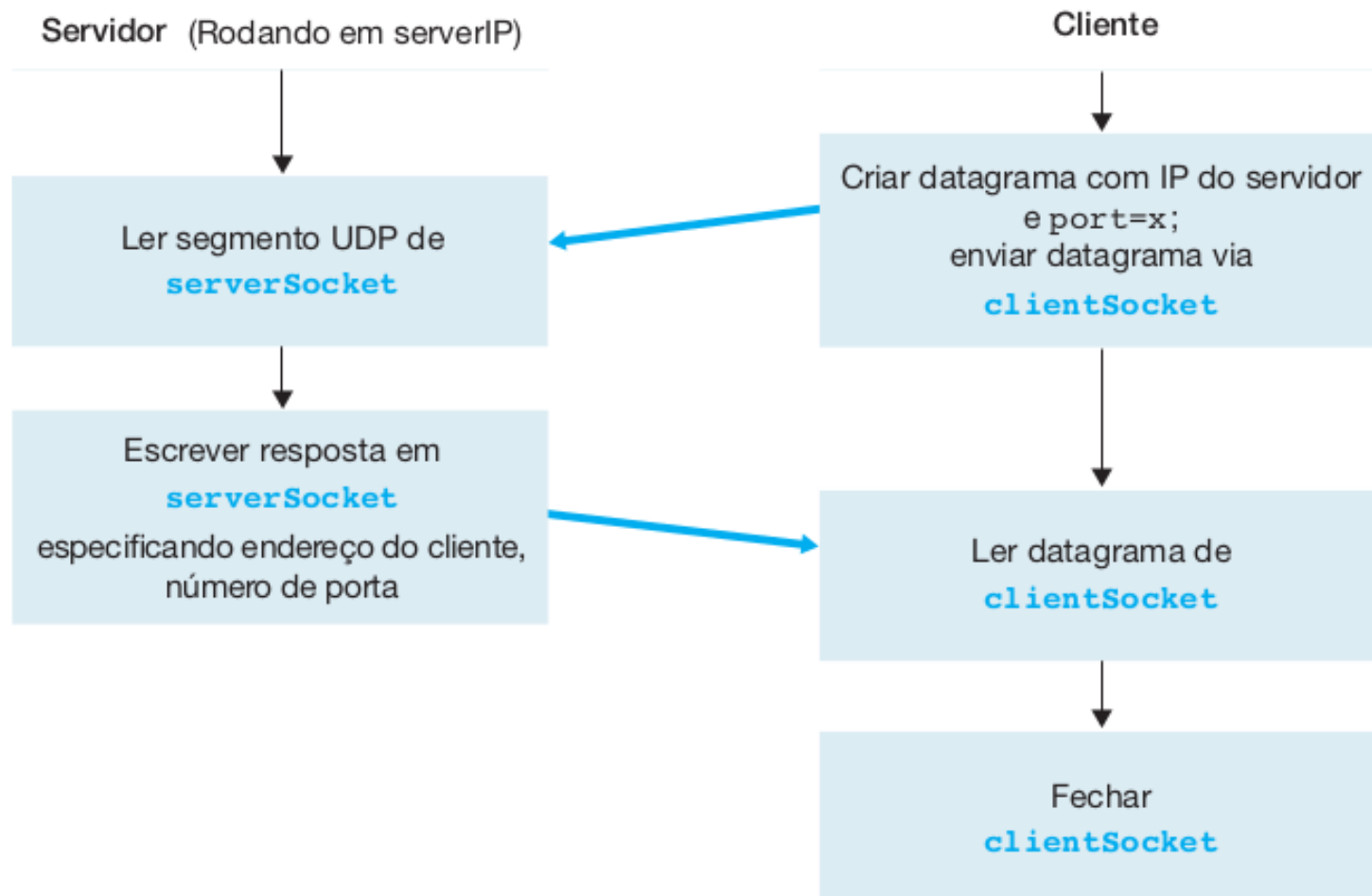




## 2 – Camada de Aplicação / 2.7 – Sockets TCP e UDP

### ... 2.7.1 – Programação de Sockets com UDP

- “**figura**” .. destaca as operações de socket realizadas pelo cliente e servidor, que se comunicam por meio de um serviço de transporte.



## 2 – Camada de Aplicação / 2.7 – Sockets TCP e UDP

### ... 2.7.1 – Programação de Sockets com UDP

- UDPClient.py .. programa Cliente em “python”.

```
01. from socket import *
02. serverName = 'hostname'
03. serverPort = 12000
04. clientSocket = socket(socket.AF_INET, socket.SOCK_DGRAM)
05. message = raw_input('Input lowercase sentence:')
06. clientSocket.sendto(message,(serverName, serverPort))
07. modifiedMessage, serverAddress = clientSocket.recvfrom(2048)
08. print modifiedMessage
09. clientSocket.close()
```

## 2 – Camada de Aplicação / 2.7 – Sockets TCP e UDP

### ... 2.7.1 – Programação de Sockets com UDP

- UDPClient.py .. programa Cliente em “python”
- ... módulo socket forma a base de todas as comunicações de rede em “python” incluindo a linha da qual se cria um socket no programa.

**from socket import \***

- ... define a cadeia serverName como “hostname”, ou seja, IP do servidor (p.ex., “128.138.32.126”) ou nome de “host” (p.ex., “cis.poly.edu”)

**serverName = 'hostname'**

**serverPort = 12000**

- ... socket cliente onde o parâmetro indica a família do endereço (p.ex., AF\_INET indica que a rede subjacente está usando IPv4);

**clientSocket = socket(socket.AF\_INET, socket.SOCK\_DGRAM)**

## 2 – Camada de Aplicação / 2.7 – Sockets TCP e UDP

### ... 2.7.1 – Programação de Sockets com UDP

- ... “raw\_input()” é uma função interna da linguagem “python” que quando executada faz com que o cliente receba o texto “Input data:”;  
**message = raw\_input('Input lowercase sentence:')**
- ... de posse do socket e da mensagem, envia-se a mensagem pelo socket ao “host” de destino pelo método “sendto()”;  
**clientSocket.sendto(message,(serverName, serverPort))**
- ... ao receber um pacote no socket cliente, os dados são colocados nas variáveis “modifiedMessage” e “serverAddress”, enquanto o método “recvfrom” usa o tamanho do buffer como 2048 bytes.  
**modifiedMessage, serverAddress = clientSocket.recvfrom(2048)**

## 2 – Camada de Aplicação / 2.7 – Sockets TCP e UDP

### ... 2.7.1 – Programação de Sockets com UDP

- ... esta linha imprime “modifiedMessage”;  
**print modifiedMessage**
- ... fecha o socket e o processo é concluído.  
**clientSocket.close()**
- Obs.: ... variável “serverAddress” contém tanto o endereço IP quanto o nro. de porta do servidor e embora o cliente já conheça o IP do servidor desde o início, o “python” oferece o endereço do servidor, apesar disso.

## 2 – Camada de Aplicação / 2.7 – Sockets TCP e UDP

### ... 2.7.1 – Programação de Sockets com UDP

- UDPServer.py .. programa Server em “python”.

```
01. from socket import *
02. serverPort = 12000
03. serverSocket = socket(AF_INET, SOCK_DGRAM)
04. serverSocket.bind(('', serverPort))
05. print "The server is ready to receive"
06. while 1:
07.     message, clientAddress = serverSocket.recvfrom(2048)
08.     modifiedMessage = message.upper()
09.     serverSocket.sendto(modifiedMessage, clientAddress)
```

## 2 – Camada de Aplicação / 2.7 – Sockets TCP e UDP

### ... 2.7.1 – Programação de Sockets com UDP

- UDPServer.py .. programa Server em “python”.
- ... linha vinculada ao nro. de porta 12000 e ao endereço IP do servidor, ambos combinados no socket do servidor (UDPServer).

**serverSocket.bind(("", serverPort))**

- ... laço “while” permite que UDPServer receba e processe pacotes dos clientes indefinidamente, ou seja, enquanto houver pacotes a receber e o servidor continuar em execução como um processo.  
**while 1:**
- ... quando um pacote chega, os dados são colocados na variável “message” e o IP de origem é colocado na variável “clientAddress”.  
**message, clientAddress = serverSocket.recvfrom(2048)**

## 2 – Camada de Aplicação / 2.7 – Sockets TCP e UDP

### ... 2.7.1 – Programação de Sockets com UDP

- ... recebe a linha de caracteres e usa o método “upper()” para convertê-las em letras maiúsculas, ou seja, é o núcleo da aplicação.  
**modifiedMessage = message.upper()**
- ... anexa o endereço IP e número de porta do cliente à mensagem em letras maiúsculas, enviando o pacote ao socket do servidor.  
**serverSocket.sendto(modifiedMessage, clientAddress)**
- ... após enviar o pacote, o servidor permanece no laço “while” esperando até que outro pacote UDP chegue de qualquer cliente.
- “**desenvolver a própria aplicação**” .. modifique pequenos trechos dos programas cliente e servidor, p.ex., para permitir que o cliente continue a enviar caracteres minúsculos para conversão em maiúsculas.



## 2 – Camada de Aplicação / 2.7 – Sockets TCP e UDP

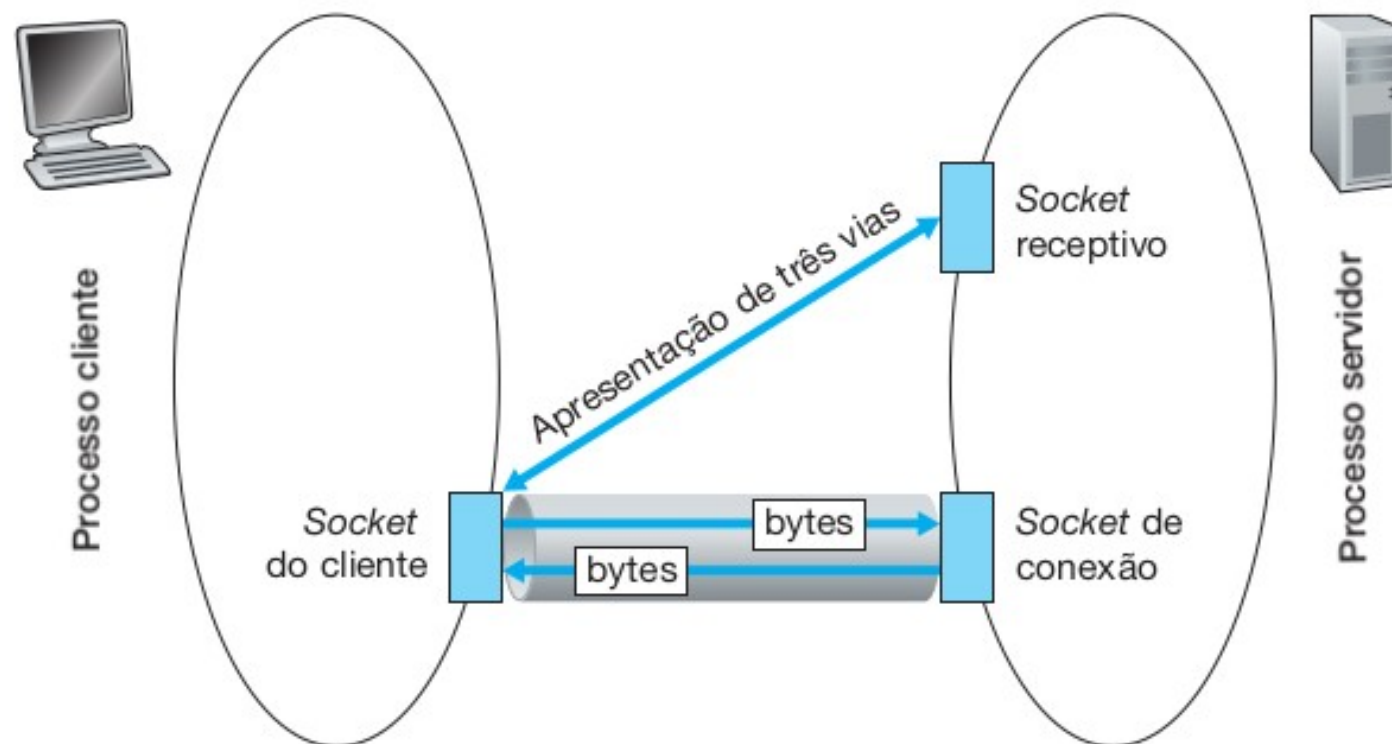
### 2.7.2 – Programação de Sockets com TCP

- Como o TCP é um protocolo orientado a conexão, é necessário que se estabeleça uma conexão entre cliente e servidor para que ambos possam enviar e receber dados um do outro.
- ... uma ponta dessa conexão está ligada ao socket cliente e a outra está ligada a um socket servidor.
- “**conexão**” .. associação do socket cliente (IP e “port”) ao socket do servidor (IP e “port”) realizada de comum acordo entre ambos.
- ... uma vez estabelecida, para qualquer um dos lados que deseje enviar dados, basta colocá-los na conexão através do socket.
- ... bem diferente do UDP, o servidor precisa anexar um endereço de destino ao pacote, antes de deixá-lo no socket.

## 2 – Camada de Aplicação / 2.7 – Sockets TCP e UDP

### ... 2.7.2 – Programação de Sockets com TCP

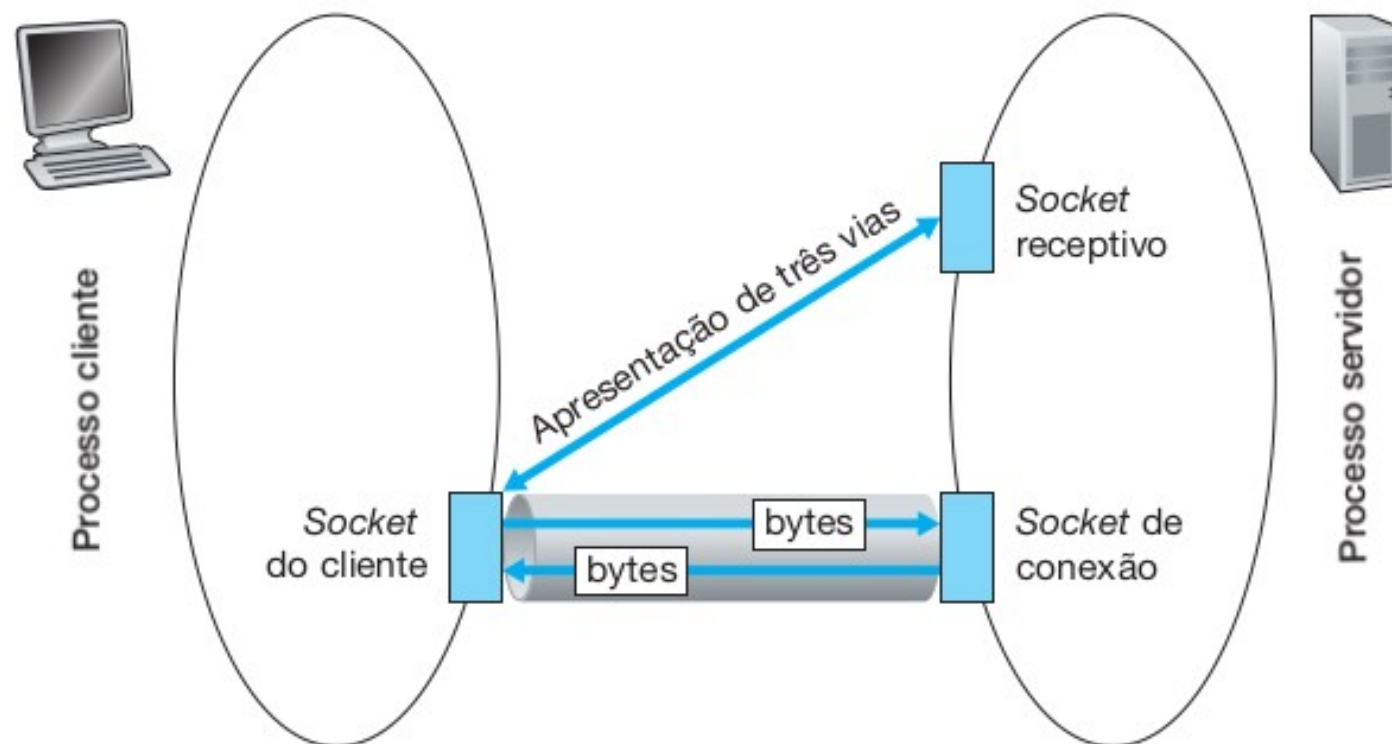
- “**ponto de vista da aplicação**” .. socket do cliente e do servidor estão conectados diretamente, como se houvesse uma tubulação entre eles.
- ... cliente pode enviar bytes para seu socket de modo arbitrário, pois o TCP garante que o processo servidor receberá.



## 2 – Camada de Aplicação / 2.7 – Sockets TCP e UDP

### ... 2.7.2 – Programação de Sockets com TCP

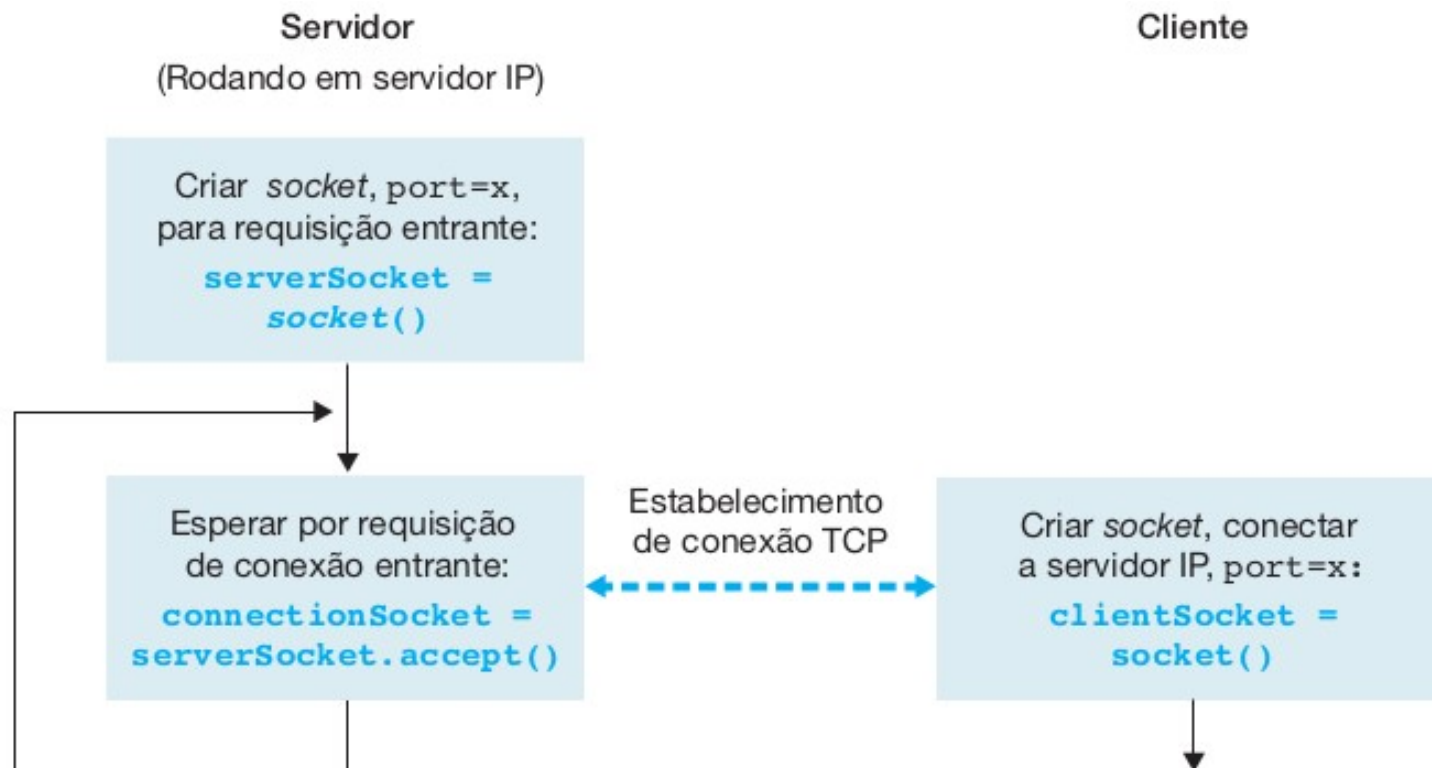
- ... adicionalmente, o processo cliente não somente envia bytes em seu socket, mas também os recebe no mesmo socket.
- ... da mesma forma, o processo servidor não só recebe bytes de seu socket de conexão, mas também envia dados por ele.



## 2 – Camada de Aplicação / 2.7 – Sockets TCP e UDP

### ... 2.7.2 – Programação de Sockets com TCP

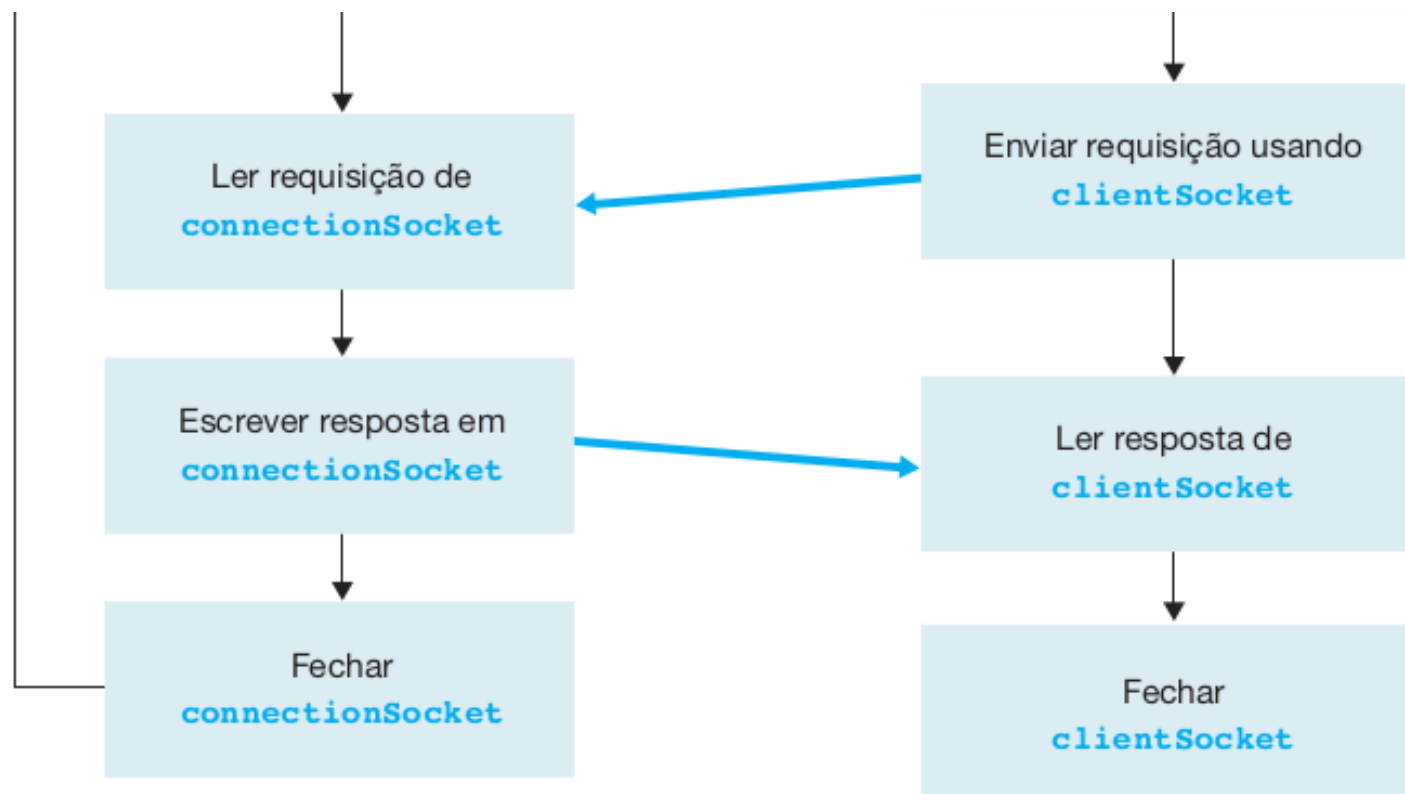
- “**aplicação cliente-servidor**” .. processo cliente envia uma linha de dados ao servidor, este converte a linha para letras maiúsculas e a remete de volta ao processo cliente.
- ... neste caso, garante-se a entrega confiável de dados.



## 2 – Camada de Aplicação / 2.7 – Sockets TCP e UDP

### ... 2.7.2 – Programação de Sockets com TCP

- “TCP” .. provê um serviço confiável de entrega de dados entre os processos cliente e servidor, ou seja, garante-se que uma vez enviados, os dados serão entregues, seja no cliente, seja no servidor.



## 2 – Camada de Aplicação / 2.7 – Sockets TCP e UDP

### ... 2.7.2 – Programação de Sockets com TCP

- TCPClient.py .. programa Client em “python”.

```
01. from socket import *
02. serverName = 'servername'
03. serverPort = 12000
04. clientSocket = socket(AF_INET, SOCK_STREAM)
05. clientSocket.connect((serverName,serverPort))
06. sentence = raw_input('Input lowercase sentence:')
07. clientSocket.send(sentence)
08. modifiedSentence = clientSocket.recv(1024)
09. print 'From Server:', modifiedSentence
10. clientSocket.close()
```

## 2 – Camada de Aplicação / 2.7 – Sockets TCP e UDP

### ... 2.7.2 – Programação de Sockets com TCP

- ... “clientSocket” .. criação do socket cliente, cujo primeiro parâmetro indica que a rede subjacente está usando IPv4;
- ... segundo parâmetro indica que o socket é do tipo SOCK\_STREAM , ou seja, é um socket TCP (em vez de um UDP);

**clientSocket = socket(AF\_INET, SOCK\_STREAM)**

- ... antes do cliente enviar dados ao servidor (e vice-versa) usando um socket TCP, é necessário estabelecer uma conexão TCP entre eles;
- ... ao ser executado, estabelece-se uma conexão que envolve a troca de mensagens em três etapas (C > S; S > C e C > S);

**clientSocket.connect((serverName,serverPort))**

## 2 – Camada de Aplicação / 2.7 – Sockets TCP e UDP

### ... 2.7.2 – Programação de Sockets com TCP

- ... obtém uma sentença do usuário até que o mesmo termine a linha digitando a tecla “enter”  
**sentence = raw\_input('Input lowercase sentence:')**
- ... envia a cadeia de caracteres pelo socket do cliente na conexão sem adicionar ao pacote o IP e “port”, como foi feito no socket UDP;
- **clientSocket.send(sentence)**



## 2 – Camada de Aplicação / 2.7 – Sockets TCP e UDP

### ... 2.7.2 – Programação de Sockets com TCP

- ... quando os caracteres chegam do servidor, eles são colocados na cadeia “modifiedSentence” e continuam a ser acumulados em até que a linha termine com um caractere de tecla “enter”;

**modifiedSentence = clientSocket.recv(2048)**

- ... fecha o socket e, portanto, fecha a conexão TCP entre cliente e servidor, ou seja, há também troca de mensagens entre cliente e servidor para formalizar o fechamento da conexão;

**clientSocket.close()**

## 2 – Camada de Aplicação / 2.7 – Sockets TCP e UDP

### ... 2.7.2 – Programação de Sockets com TCP

- TCPServer.py .. programa Server em “python”.

```
01. from socket import *
02. serverPort = 12000
03. serverSocket = socket(AF_INET,SOCK_STREAM)
04. serverSocket.bind(('',serverPort))
05. serverSocket.listen(1)
06. print 'The server is ready to receive'
07. while 1:
08.     connectionSocket, addr = serverSocket.accept()
09.     sentence = connectionSocket.recv(1024)
10.     capitalizedSentence = sentence.upper()
11.     connectionSocket.send(capitalizedSentence)
12.     connectionSocket.close()
```

## 2 – Camada de Aplicação / 2.7 – Sockets TCP e UDP

### ... 2.7.2 – Programação de Sockets com TCP

- ... assim como em TCPClient, o servidor cria um socket TCP;  
**serverSocket=socket(AF\_INET,SOCK\_STREAM)**
- ... de modo semelhante a UDPServer, associa-se o número de porta do servidor “serverPort” ao socket;
- ... “serverSocket” é o socket de entrada tão logo a conexão seja estabelecida e, na sequência, basta aguardar o cliente;  
**serverSocket.bind(("",serverPort))**
- ... faz com que o servidor escute as requisições de conexão do cliente, sendo que o parâmetro especifica o nro. máximo de conexões em fila;  
**serverSocket.listen(1)**

## 2 – Camada de Aplicação / 2.7 – Sockets TCP e UDP

### ... 2.7.2 – Programação de Sockets com TCP

- ... quando o cliente bate nesta porta, o programa chama o método “accept()” para “serverSocket”, que cria um novo socket no servidor, chamado “connectionSocket”, dedicado a esse cliente específico;  
**connectionSocket, addr = serverSocket.accept()**
- ... cliente e servidor completam a apresentação, criando uma conexão TCP entre o clientSocket do cliente e o “connectionSocket” do servidor;
- ... após estabelecer a conexão TCP, cliente e servidor podem enviar bytes um para o outro por ela;
- ... com TCP, todos os bytes enviados de um lado têm garantias não apenas de que chegarão ao outro lado, mas também na ordem.

## 2 – Camada de Aplicação / 2.7 – Sockets TCP e UDP

### ... 2.7.2 – Programação de Sockets com TCP

- ... depois de enviar a sentença modificada ao cliente, o socket da conexão é fechado, mas como “serverSocket” permanece aberto, outro cliente agora pode bater à porta e enviar uma sentença ao servidor.
- **connectionSocket.close()**
- “**sugestão**” .. executar os dois programas em dois “hosts” separados, bem como modificá-los para realizar objetivos ligeiramente diferentes.
- ... adicionalmente, compare o par de programas UDP com o par de programas TCP e identifique as diferenças !!