Fundamentos de Redes de Computadores

Cap. 1 - A Internet e como funcionam as aplicações distribuídas

Departamento de Informática da

FCT/UNL

"Felix, qui potuit rerum cognoscere causas."

(Feliz aquele que conhece a causa das coisas)

Virgílio, historiador romano

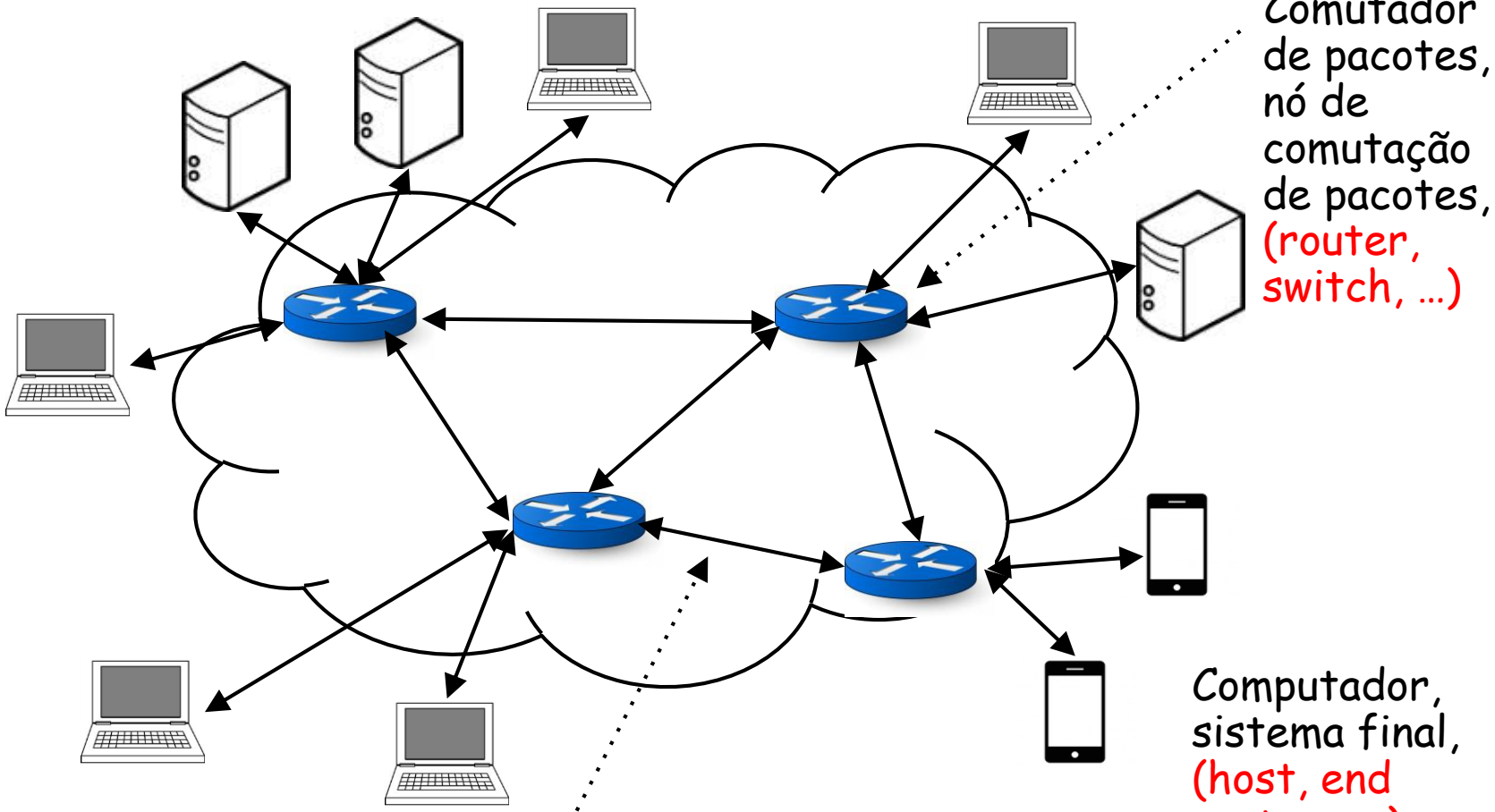
Objectivos

- A Internet é uma infraestrutura física que suporta o funcionamento de aplicações distribuídas
- Estas funcionam com base num conjunto de funcionalidades e conceitos relativamente pequeno
- Para a comunicação usam a troca de mensagens e os canais de comunicação, também conhecidos como as "abstrações do transporte"
- O objectivo desta lição é proporcionar uma primeira visão deste conjunto de conceitos

O que é uma rede de computadores

Uma rede de computadores permite que os programas executados pelos diferentes computadores que lhe estão ligados troquem mensagens com o objectivo de troca de informações e coordenação da sua execução.

Uma rede de computadores



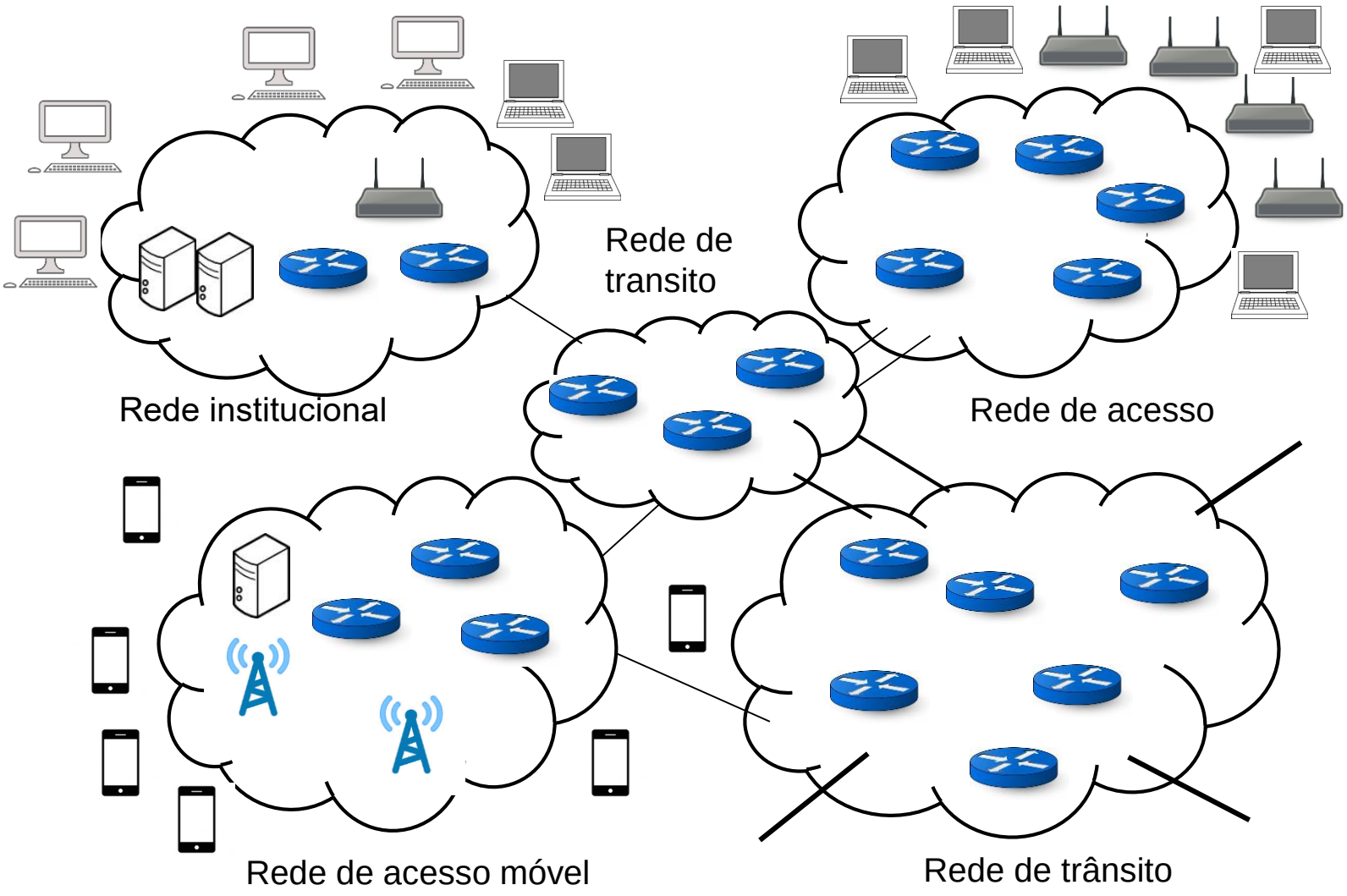
A Internet

A Internet é uma rede que interliga muitas redes que comunicam através da família de protocolos TCP/IP. Essas redes são muito variadas e incluem redes residenciais, institucionais, redes de provedores móveis, governamentais, ISPs locais, regionais e mundiais, redes de centros de dados, redes de fornecedores de conteúdos, etc.. Essas redes interligam biliões de dispositivos computarizados e usam uma grande variedade de canais de comunicação e comutadores de pacotes. A Internet suporta uma gigantesca massa de serviços de comunicação, coordenação e de acesso a informação.

A Internet é uma rede de redes

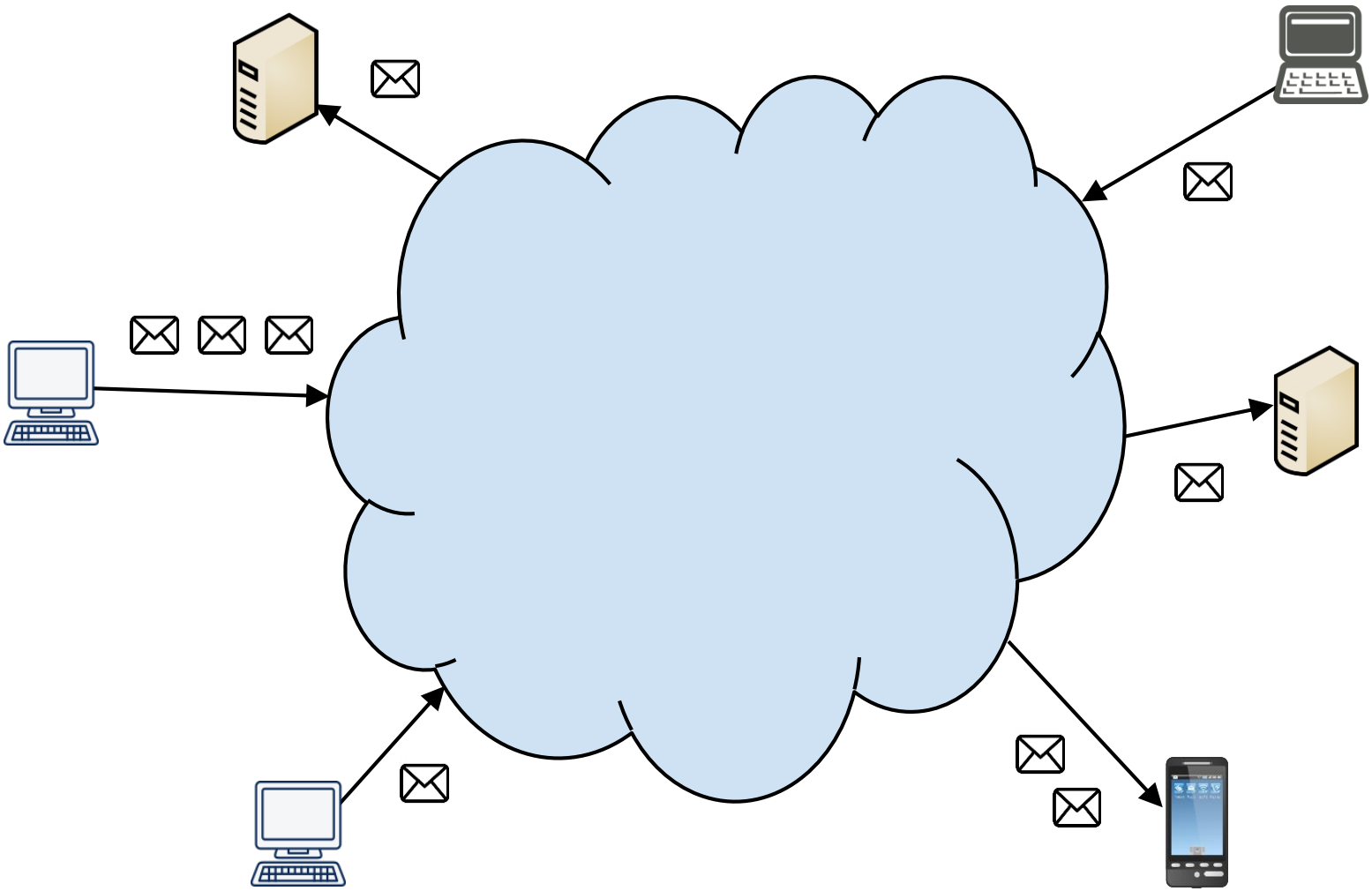
- A Internet
  - É uma rede de redes
  - Cada uma com as suas características e tecnologias específicas
  - Geridas independentemente
  - Com objectivos próprios
- O conjunto oferece um serviço comum
  - Transporte de pacotes de dados entre quaisquer dois computadores
  - Sistema de endereçamento comum, coordenado e uniforme

A Internet é uma rede de redes

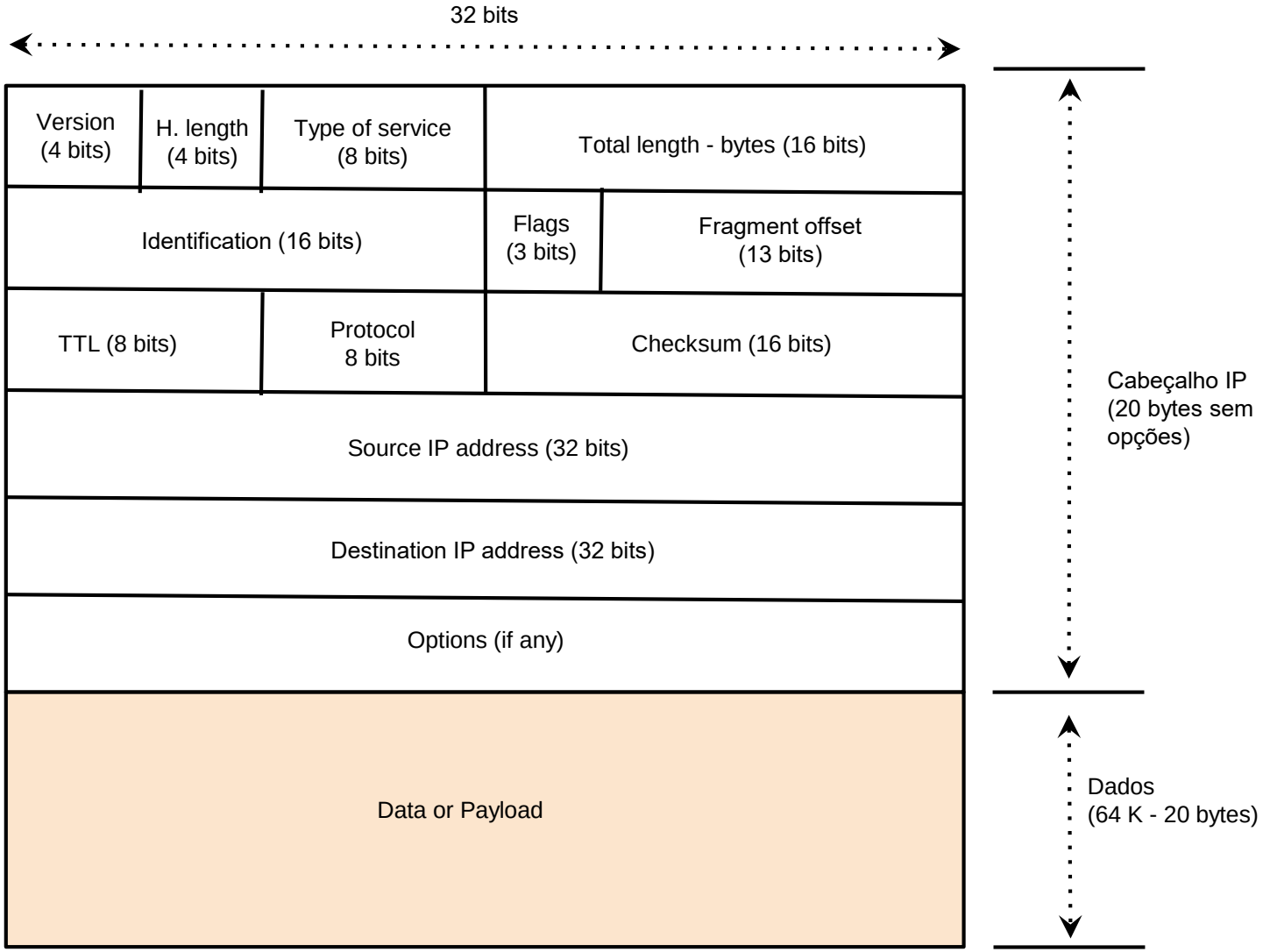


Redes de centros de dados

Internet Protocol - IP



Formato de um pacote IP

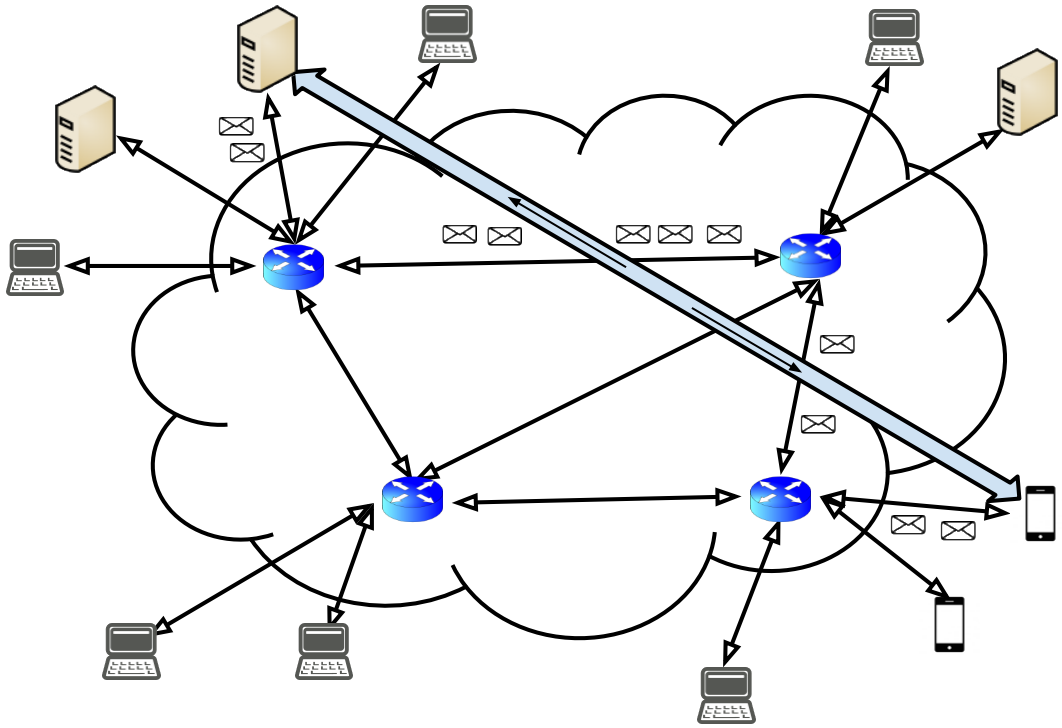


Nível de serviço do IP

- Caracterização do serviço IP
  - Os pacotes são trocados entre computadores (ou melhor entre interfaces de rede) e não entre processos do sistema de operação
  - Nem todos os dados e mensagens a transferir cabem num pacote
  - Os pacotes podem perder-se devido a erros e outros problemas
  - Podem chegar por uma ordem diferente da do envio
- Desenvolver aplicações com este nível de serviço é complexo
- Necessitamos de uma interface diferente
  - Diretamente acessível aos processos que executam no sistema de operação
  - Com outra qualidade de serviço
  - Uma nova abstração a disponibilizar pelo sistema de operação
  - Mas baseada no serviço IP

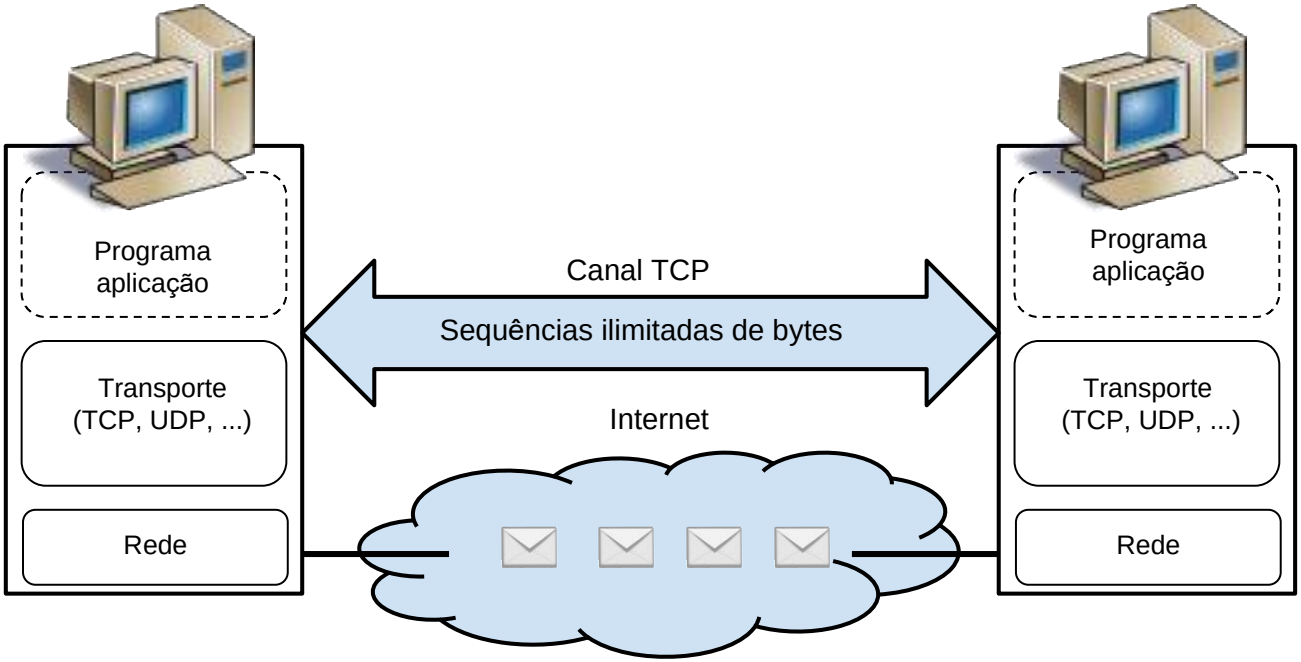
Interface de transporte de dados

- Os processos enviam e recebem mensagens através da interface de *Sockets*, uma API do sistema de operação
- Um *socket* dá acesso a canais virtuais
  - As mensagens enviadas de um lado do canal chegam por ordem ao outro lado



Exemplo: Canal TCP

- Um canal TCP é um canal lógico bidireccional e fiável que transmite sequências de bytes entre dois programas



API *Sockets* (Application Programming Interface)

- Fornece primitivas para criar *sockets*, enviar e receber mensagens processadas e formatadas pelas aplicações
- Exemplo: Java *Socket API*
  - Class Socket
  - Métodos: *socket(), send(), write(), receive(), read(), close(), ...*
- O sistema de operação disponibiliza assim um serviço de comunicação generalizada implementado sobre o serviço de rede IP

TCP — Transmission Control Protocol

- Serviço de comunicação entre dois computadores
  - Sequência ordenada e fiável de bytes
  - Transmite simultaneamente nos dois sentidos
  - Usa o serviço de rede (IP)
- Totalmente implementado pelos computadores, nos sistema de operação
  - Retransmissão dos pacotes de dados perdidos
  - Colocação dos dados por ordem e supressão dos duplicados
  - Controlo de fluxos para evitar "afogar" o receptor
  - Controlo de saturação para adaptar a velocidade de transmissão à capacidade da rede

Entrega dos dados e divisão do trabalho

- Rede
  - Transporta pacotes entre computadores identificados pelos seus endereços IP, responsabiliza-se por fazer chegar os pacotes às interfaces de rede dos mesmos. Se houverem problemas não apresenta desculpas! (e.g. "desculpe mas não funcionou")
- Sistema de operação
  - Envia e recebe os dados da rede mas implementa o nível de serviço adequado (ordem, fiabilidade, …)
  - Um *socket* é identificado pelo conjunto { endereços IP, protocolo, porta }
- Aplicação
  - Lê e escreve dados no *socket*
  - Interpreta os dados (e.g., sintetização de uma página WEB)
  - Executa um protocolo do nível aplicacional

O que é um protocolo ?

- Para que os computadores e o sistema possam funcionar necessitamos
  - Especificar o formato e os campos das mensagens que são trocadas (e.g. onde está o endereço de destinatário ?)
  - Indicar como as mesmas são interpretadas e tratadas (e.g. como detectamos que os dados estão a chegar na ordem correta?)
  - De que forma os interlocutores se põem de acordo (e.g. tenho a certeza que o receptor recebeu os dados?)
- Ao conjunto destas regras chama-se um protocolo, as mesmas têm de ser publicadas numa norma
- Os protocolos da Internet são normalizados pela IETF em documentos designados por RFCs (Request For Comments)

Protocolos

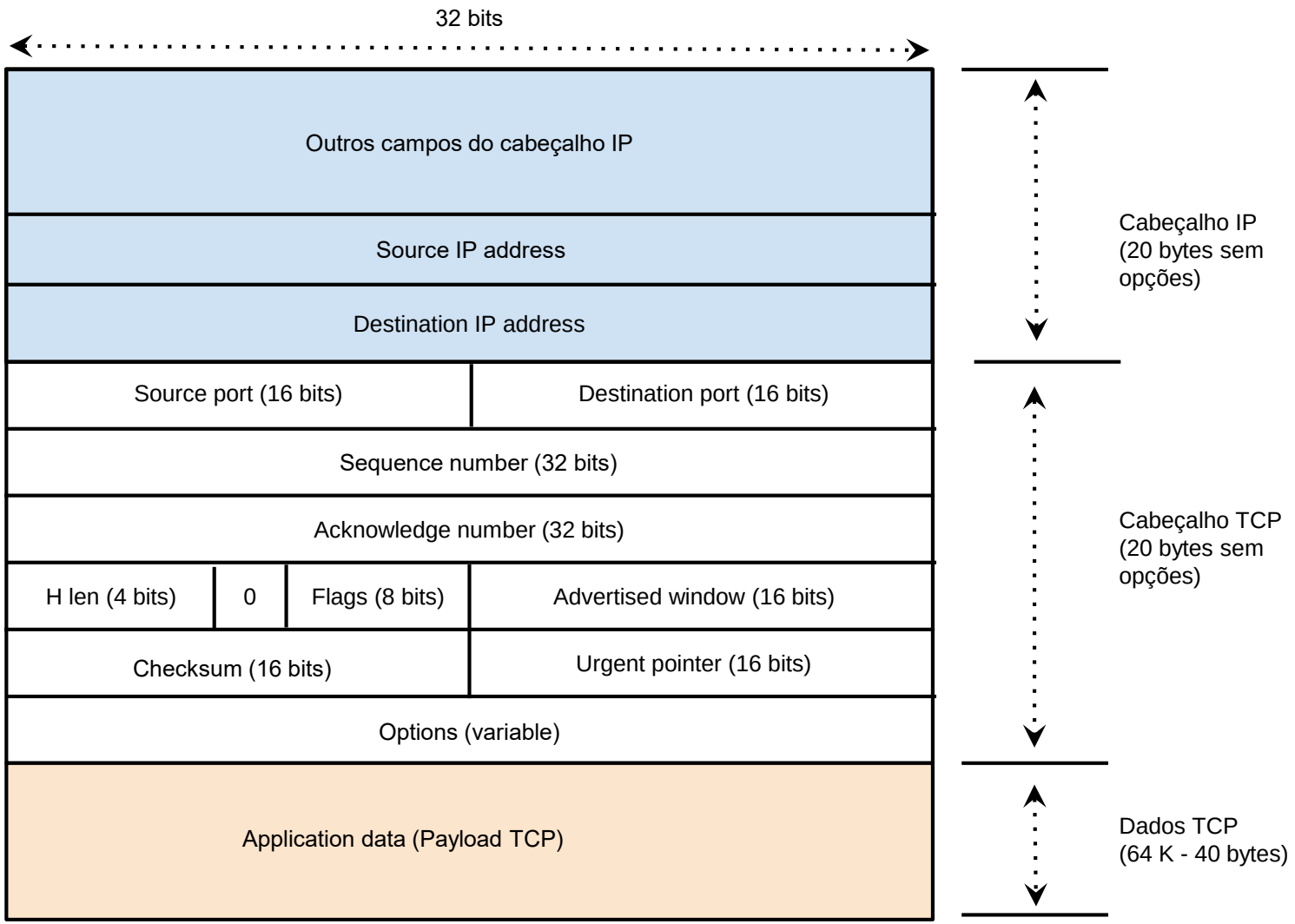
- Um protocolo é um conjunto de mensagens, regras sintácticas e regras semânticas que regulam a comunicação e coordenação de um conjunto de entidades para atingirem um objectivo comum.

- As mensagens usadas pelos protocolos são normalmente estruturadas em duas partes: o cabeçalho e os dados ou *payload*. Os cabeçalhos contém informação de controlo e os dados contém os dados transportados pela mensagem.

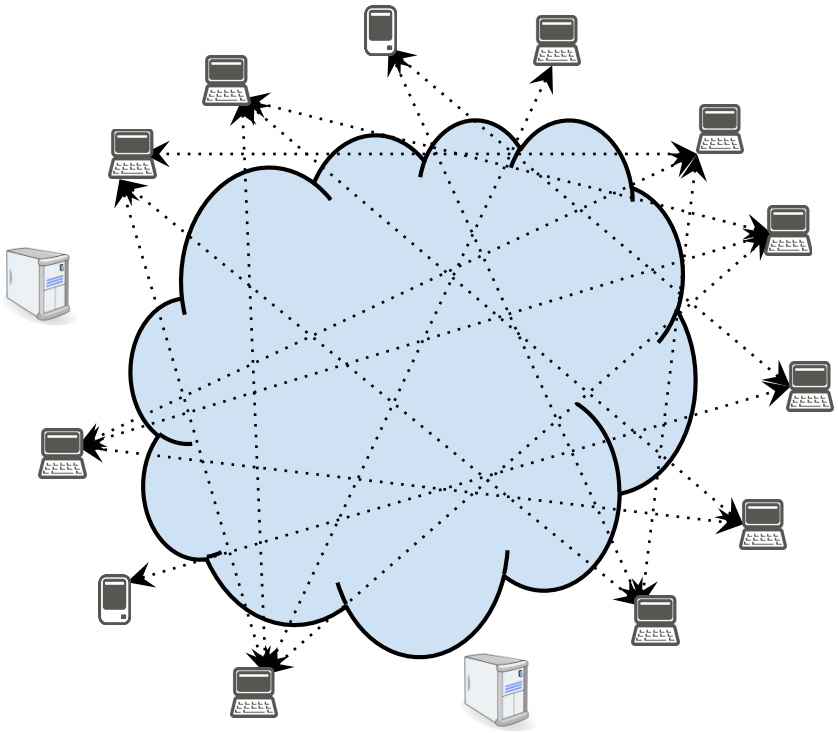
- Os protocolos podem ser abertos e publicamente estabelecidos como normas, ou propriedade de uma instituição que nesse caso os pode alterar a qualquer momento.

- Nas redes de computadores a maioria dos protocolos são normalizados por diferentes organismos especializados para esse efeito como o IETF, o IUT, …

Formato dos segmentos TCP

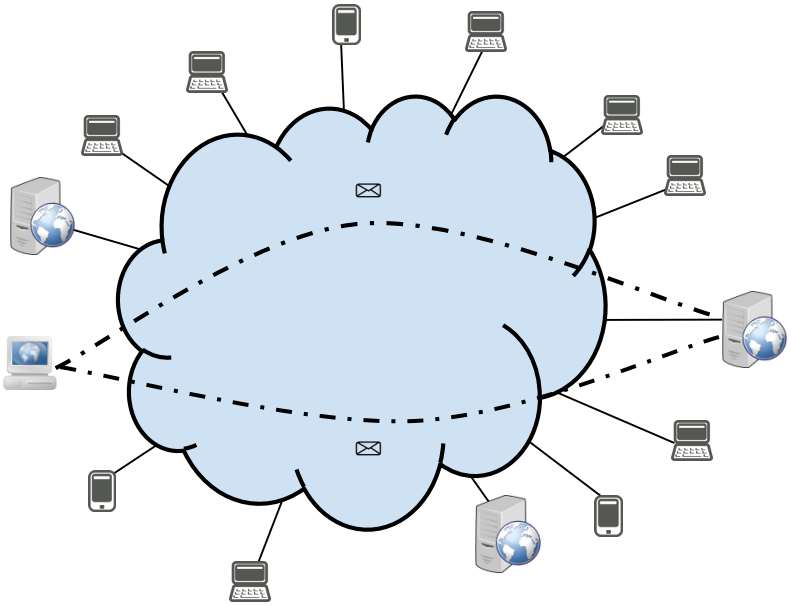


Aplicações distribuídas



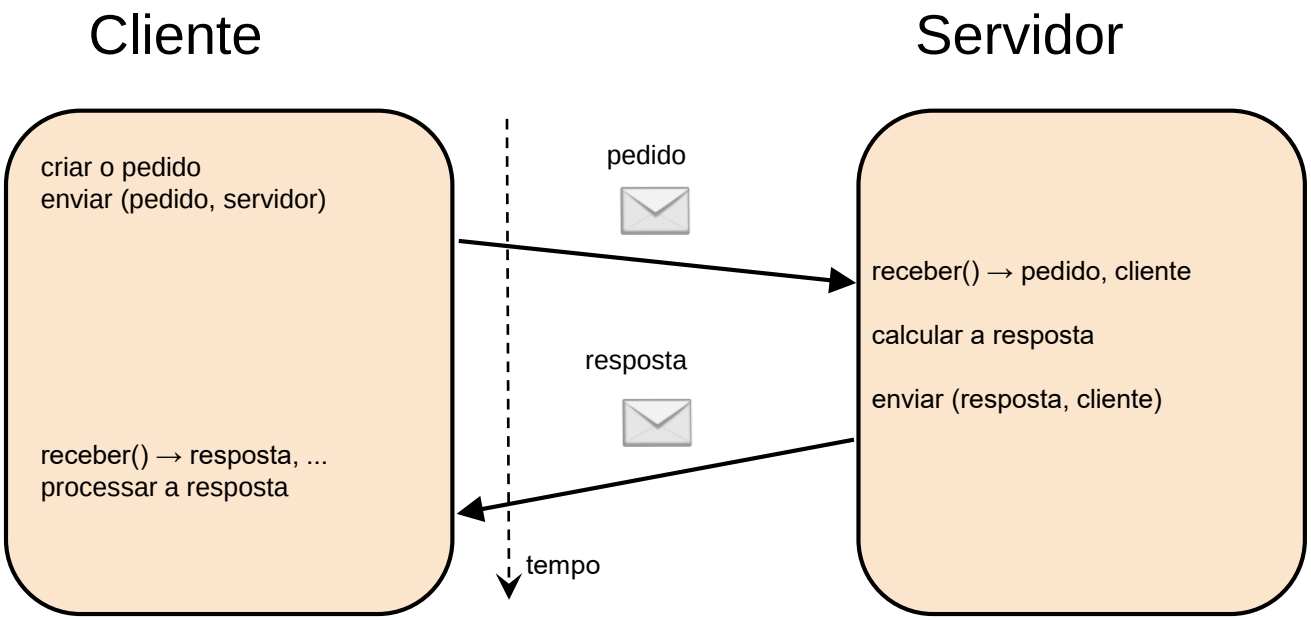
- Aplicação: processos distribuídos que comunicam através de mensagens
  - Executam nos computadores
  - Comunicam através de protocolos do nível aplicação
  - Usam os serviços do nível de transporte da rede
- Protocolos aplicacionais
  - Definem as mensagens trocadas pelas aplicações e a sua semântica
  - Não são senão uma parte da aplicação

Aplicações cliente / servidor



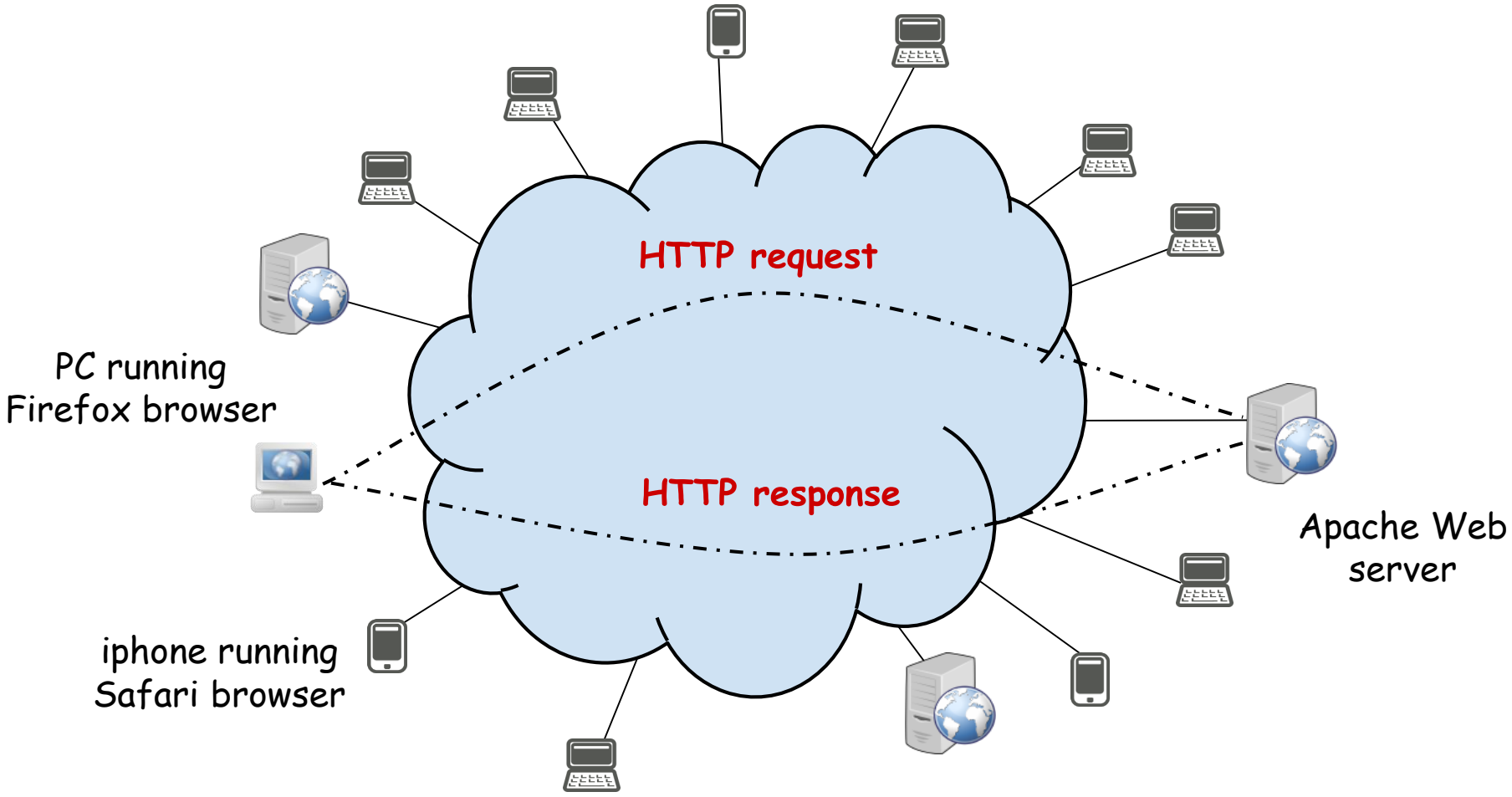
- O cliente *solicita serviços* ao servidor
- Este responde ao pedido
- Um servidor que serve um cliente de cada vez diz-se um *servidor iterativo (serializa os pedidos)*
- Um servidor que serve vários clientes em paralelo diz-se um *servidor concorrente*
- Ao protocolo base chama-se um protocolo *"pedido / resposta"* (*"request / reply"*)
- A sincronização característica é semelhante à da invocação de um procedimento ou método executados remotamente

O padrão cliente / servidor



- O cliente solicita serviços *solicita serviços* ao servidor e fica bloqueado à espera da resposta.. O servidor responde aos pedidos
- Ao padrão base chama-se padrão cliente / servidor
- A sincronização característica é semelhante à da invocação de um procedimento ou de métodos executados remotamente

Exemplo: HyperText Transfer Protocol



Mensagens HTTP

GET /index HTTP/1.0 <cr lf>
Host: www.wikipidea.org <cr lf>
User-Agent: Mozilla/4.03 <cr lf>
<cr lf>

Pedido

Resposta

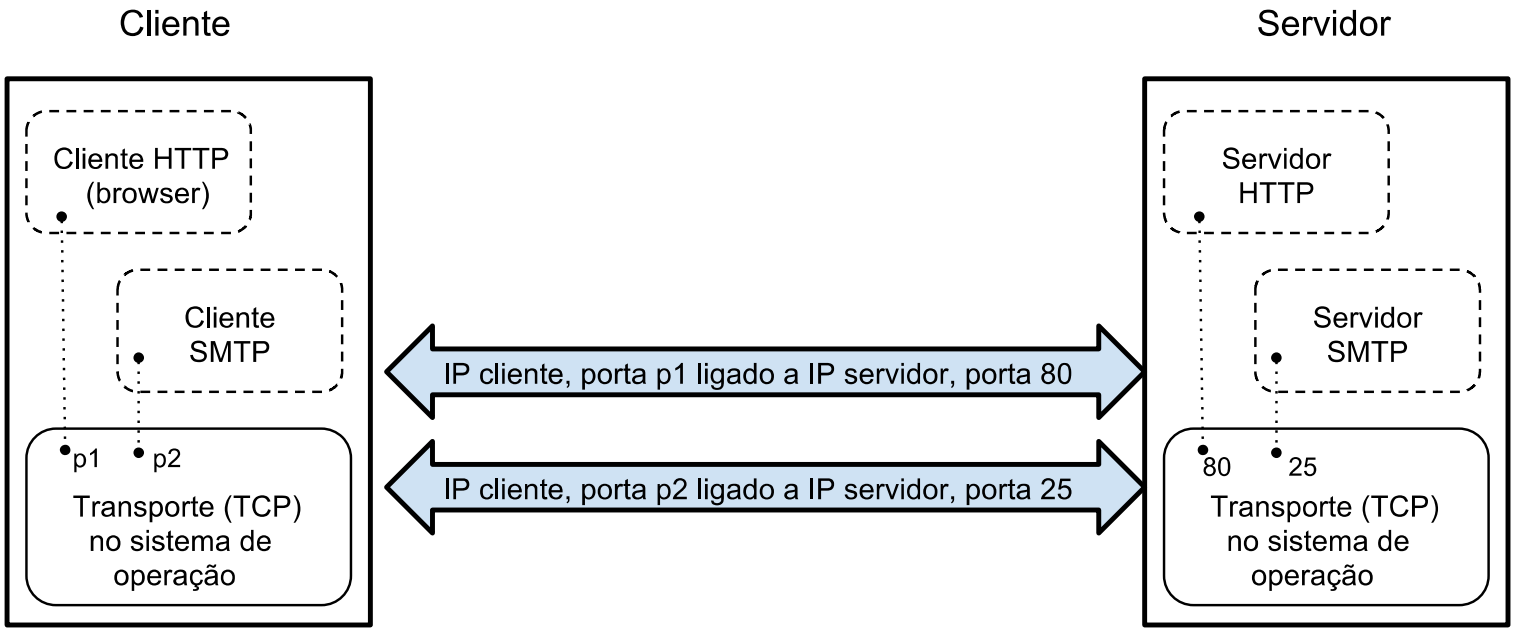
HTTP/1.1 200 OK <cr lf>
Date: Mon, 21 Feb 2015 17:51:37 GMT <cr lf>
Server: Apache/2.2.9 (Debian) PHP/5.2.6-1+lenny16 with … <cr lf>
Last-Modified: Wed, 17 Aug 2011 17:46:43 GMT <cr lf>
Accept-Ranges: bytes <cr lf>
Content-Length: 4768 <cr lf>
Vary: Accept-Encoding <cr lf>
Connection: close <cr lf>
Content-Type: text/html <cr lf>
<cr lf>
……

Tempo

O Papel das portas nos sockets

- É preciso separar cada *socket*
  - Conhecer os endereços dos computadores não chega pois é necessário separar os canais entre os mesmos dois computadores
  - Para esse efeito complementam-se os endereços com portas
- *As portas indicam serviços*
  - E.g., servidor HTTP na porta 80
  - E.g., servidor SMTP na porta 25

Os endereços diferenciam os computadores, as portas os serviços

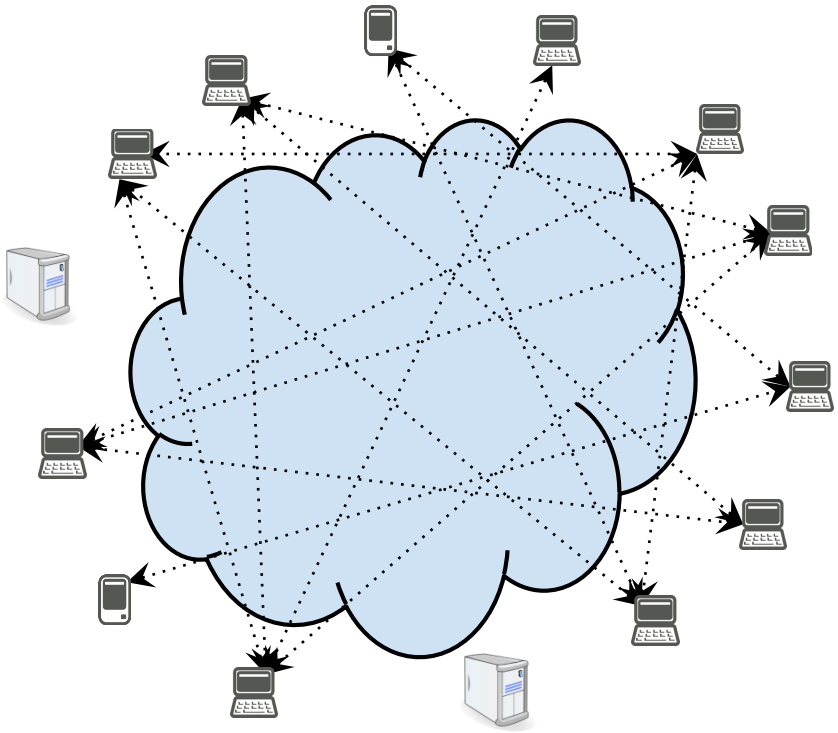


Servidores e clientes são diferentes

- Servidor – abertura passiva do *socket*
  - Está preparado para receber conexões
  - … mas não estabelece nenhuma
  - … até receber o pedido do canal

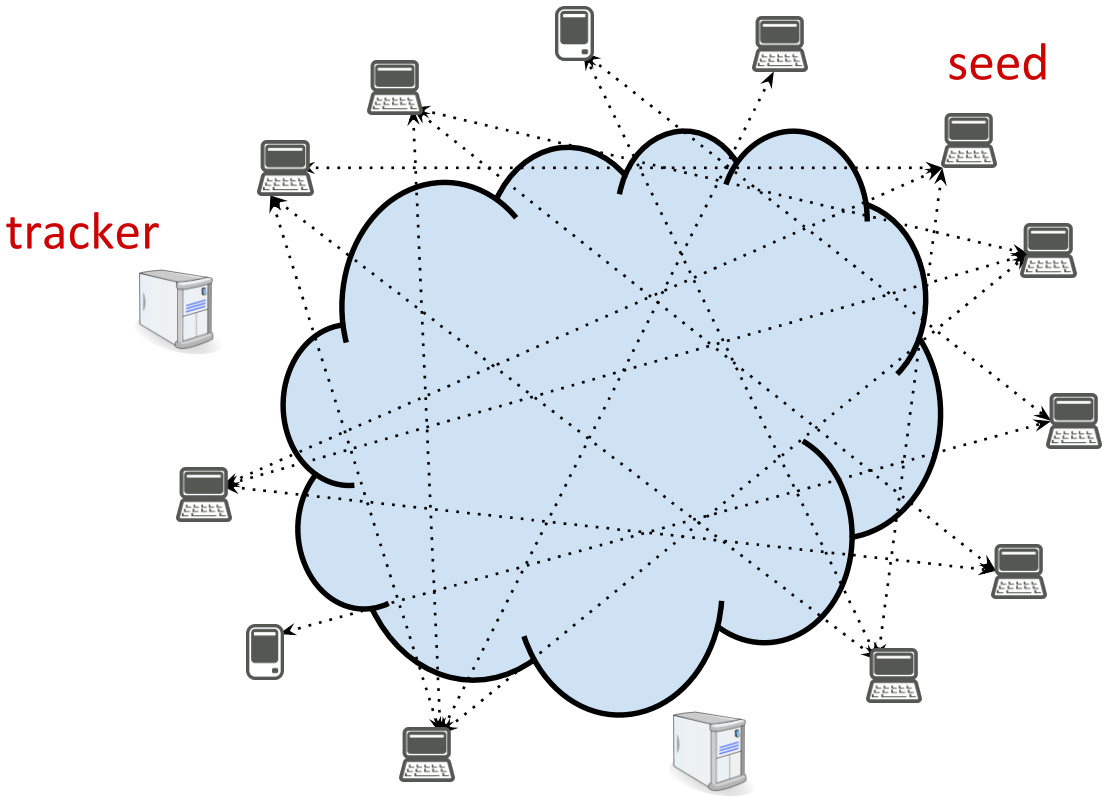
- Cria um *socket* diferente para cada cliente
  - Sempre que um cliente abre um *socket* para o servidor
  - … este cria um novo *socket* para aquele cliente

Aplicações P2P



- Não existe assimetria entre os computadores participantes
  - Todos os computadores participantes podem atuar como clientes ou servidores
  - Não existe uma hierarquia tão marcada
- Protocolos P2P (não cliente servidor)
  - Um exemplo bem conhecido tem por objetivo difundir ficheiros de forma cooperativa

Exemplo: BitTorrent

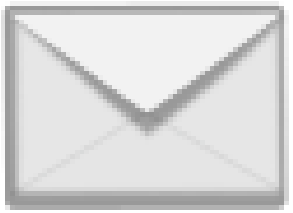


- Ficheiro partido em blocos (e.g. 256 Kbytes)
- *Peers* trocam blocos entre si até obterem todo o ficheiro
- *Tracker* – recenseia os *peers* no *enxame* – o grupo de *peers* que estão a fazer o *download* cooperativo do ficheiro
- *Seed* – é um *peer* com uma cópia de todos os blocos de ficheiro

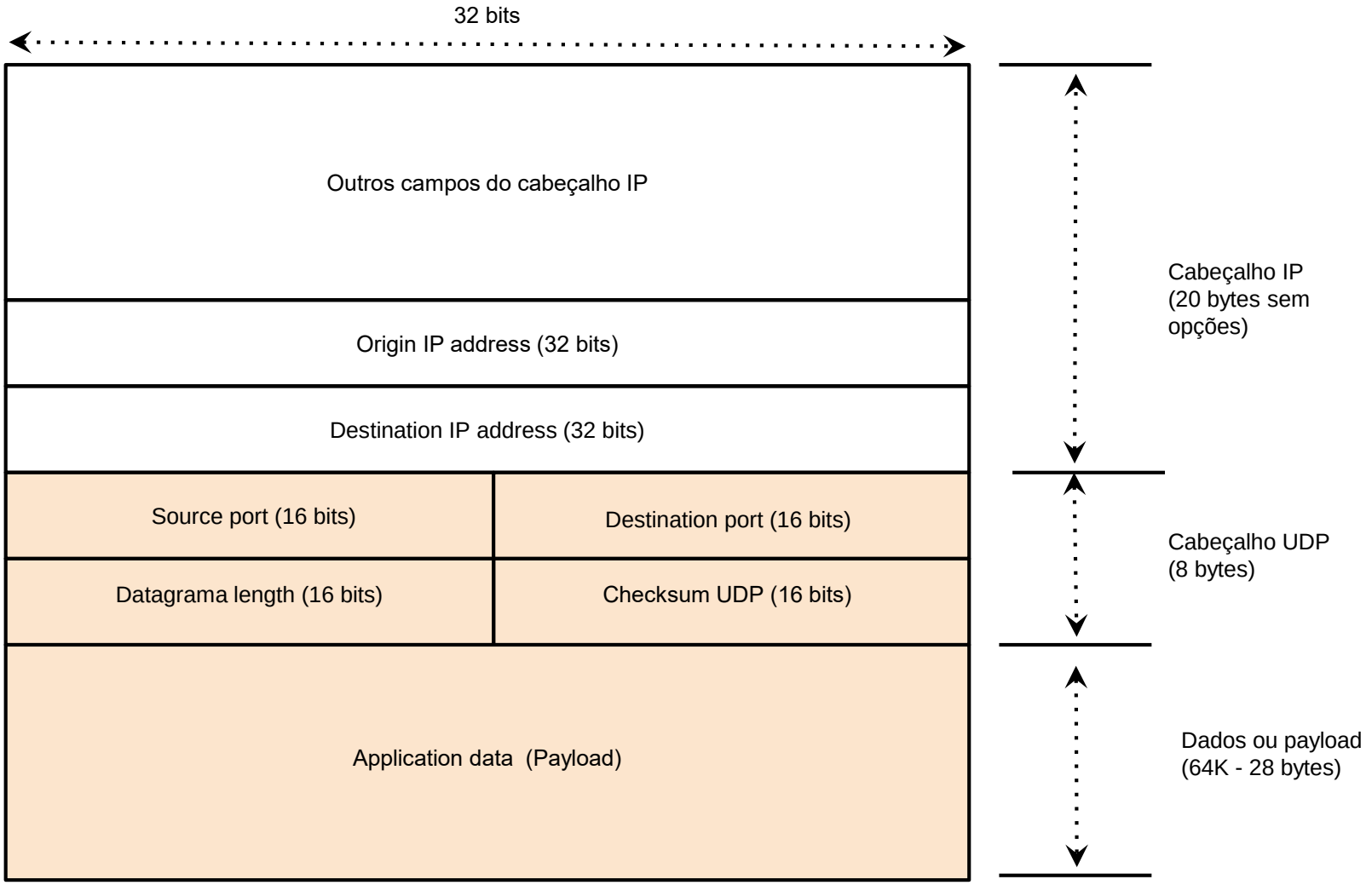
Sockets UDP

- Os canais TCP são adequados para a transferência de dados de forma fiável e suportam mensagens de dimensões arbitrárias
- Mas em certas situações não é necessário senão trocar pequenas mensagens, se recebermos uma resposta temos a certeza que o receptor recebeu a mensagem inicial
- Existe um transporte alternativo, designado UDP (*User Datagram Protocol*) mais adequado para estas situações
- Os *sockets* UDP ou *Datagram Sockets* designam-se por *sockets* sem ligação pois podem enviar mensagens (*datagramas*) para endereços e portas arbitrários de destino

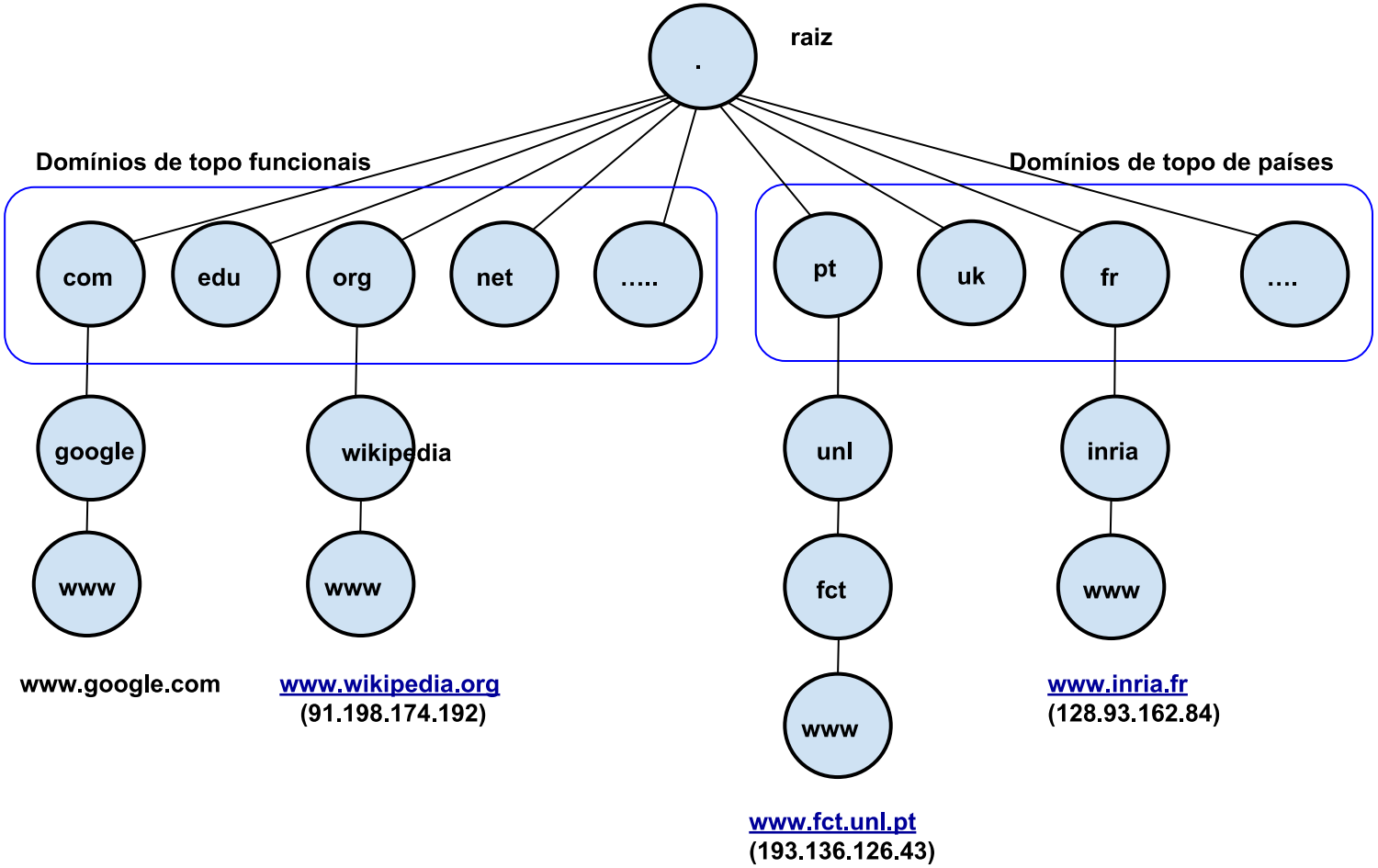
Um datagrama ou mensagem UDP



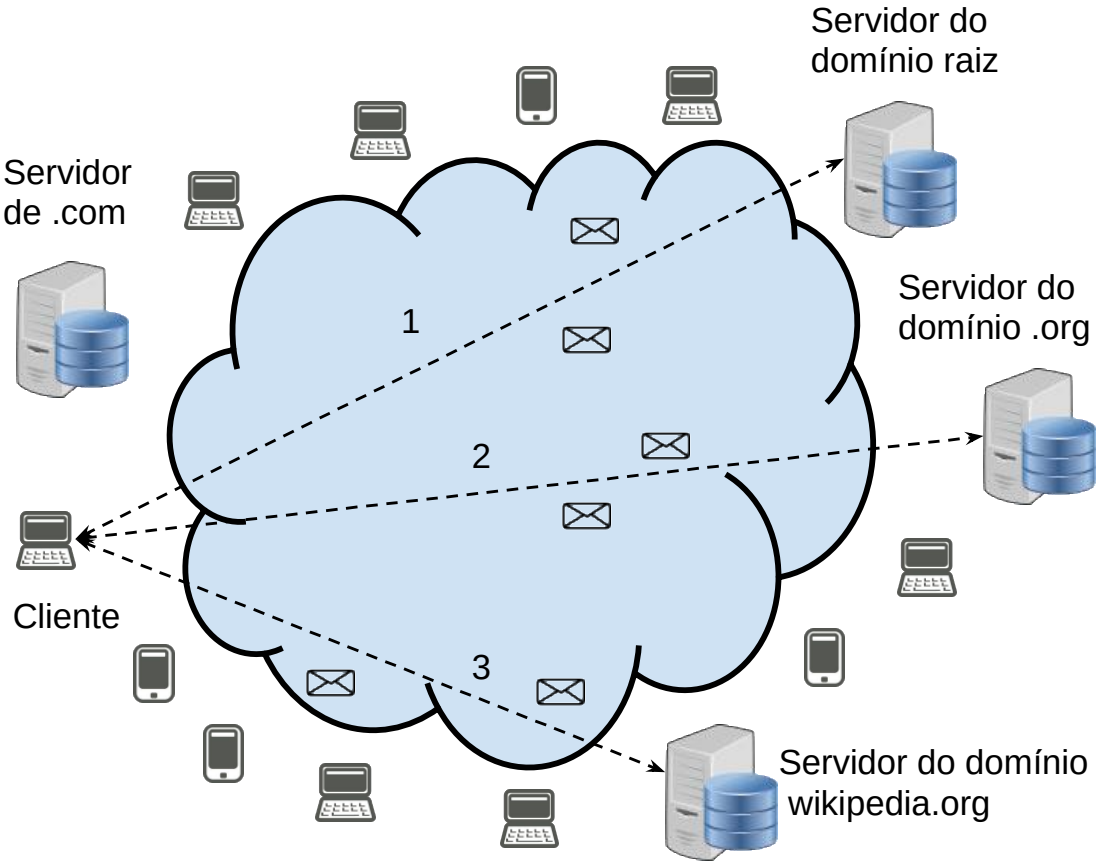
Formato dos *datagramas* UDP



Domain Name System

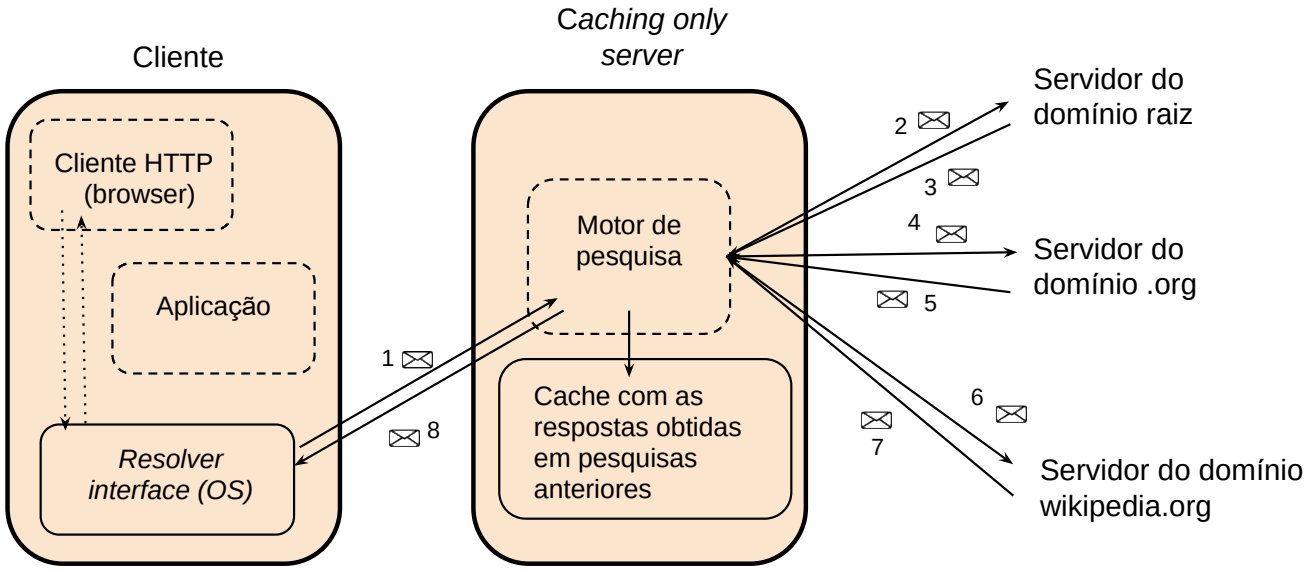


Resolução de um nome DNS



- O protocolo utiliza pedidos e respostas colocadas dentro de *datagramas* UDP
- Em caso de falha, os clientes são responsáveis pela re emissão dos pedidos e por contactar servidores alternativos

DNS Resolver e Local DNS Server



O Caching é baseado num *time-to-live* (TTL) definido pelo servidor responsável pelo nome. Permite evitar estar sempre a contactar os servidores remotos.

Transportes e suas características

- **TCP**
  - Orientado conexão (exige conexão prévia)
  - Transporte fiável de uma sequência de bytes
  - Tem controlo de fluxo e controlo de saturação
  - Não dá garantias de débito nem de latência
  - Adequado sempre que se requer fiabilidade nas transferências de quantidades significativas de dados
- **UDP**
  - Serviço para *datagramas* sem conexão (não exige conexão prévia)
  - Tem as mesmas garantias de banda, latência, controlo de fluxo ou saturação que o nível IP ("melhor esforço")
  - É adequado para interações curtas do tipo das do DNS ou quando não se requer fiabilidade na transferência dos dados mas o tempo de transferência dos dados é crítico

Conclusões

- A rede permite a existência de aplicações distribuídas
  - É a visão dos serviços fornecidos ou das aplicações acessíveis aos utilizadores
  - Baseados em aplicações programadas para executarem nos computadores
- As aplicações vêm a rede como um serviço de comunicação
  - Utilizam canais virtuais ao nível transporte, que abstraem a configuração e funcionamento interno da rede
  - As abstrações mais populares são os canais fiáveis (TCP) e a troca de mensagens (UDP)
- As aplicações distribuídas e a rede funcionam com protocolos
  - Dependentes uns dos outros (e.g. TCP e UDP implementados sobre o IP, HTTP sobre TCP, DNS sobre UDP)