



T02- Camada de aplicação

 Subjects

RC

Aplicações de rede: princípios

Criação de uma aplicação de rede

- Um programa que corre em diferentes computadores e comunica pela rede
- Não necessário escrever software para os dispositivos da rede
- Dois tipos de arquitetura: Cliente-servidor; Peer-to-peer

Arquitetura cliente-servidor

- Servidor
 - Terminal sempre ligado
 - Endereço IP permanente
- Clientes
 - Não está sempre ligado
 - Endereço IP dinâmico
 - Não comunica diretamente entre si

Arquitetura P2P

- Não a um servidor sempre ligado
- Terminais comunicam diretamente entre si
- Sistema auto-escalável
 - Os peers servem-se, mas também servem
- Gestão complexa
 - Peers ligados de forma intermitente, IPs dinâmicos

Comunicação entre processos

- Processo: programa em execução num computador
 - No mesmo computador: → comunicação entre processos do sistema operativo
 - Computadores diferentes: troca de mensagens
- Arquitetura cliente-servidor:
 - Processo servidor: espera por contacto

- Processo cliente: inicia comunicação
 - Arquitetura P2P:
 - Peers: processo cliente + processo servidor
-

Socket

- Os processos enviam/recebem mensagens para/de um socket
-

Web e HTTP

Noções básicas

- Uma pagina web consiste num conjunto de objetos:
 - Ficheiros HTML , imagens , applets , ficheiros de audio e video
 - Uma pagina web consiste num ficheiro HTML-base que inclui várias referencias para outros objetos
-

HTTP: visão geral

- Segue modelo cliente/servidor
 - cliente : browser
 - servidor: servidor web
 - Utiliza TCP como protocol de transport
-

Métodos HTTP

HTTP/1.0

- GET
 - Permite obter um objeto
- POST
 - Permite enviar informação para o servidor
- HEAD
 - Pede resposta mas sem objeto

HTTP/1.1

- GET, POST, HEAD
 - OPTIONS
 - Permite saber que metodos podem ser usados
 - PUT
 - Permite enviar um ficheiros no corpo do pedido para ser colocada no local indicado no campo URL
 - DELETE
 - Remove ficheiros indicados no campo URL
-

Conceito #1: Ligações HTTP

- Ligações não persistentes
 - Cada ligação serve para transmitir um objeto e depois é fechada
 - Novo objeto = novo ligação

- Ligações persistentes
 - E possível pedir varios objetos utilizado sempre a mesma ligação

HTTP não persistente

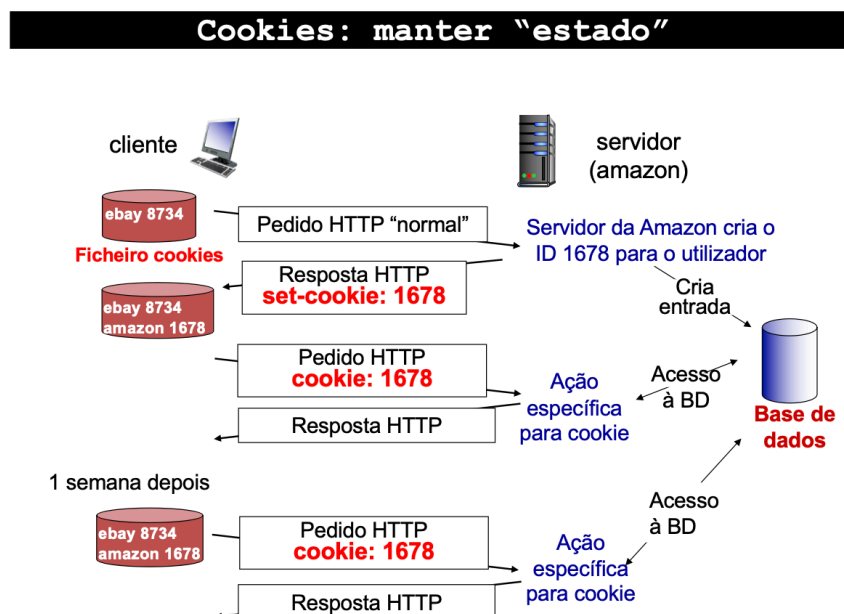
- RTT(Round Trip Time)
 - tempo entre enviar um pacote e receber a resposta
- Tempo total = (2RTT + tempo de transmissão) * num_objetos

HTTP persistente

- O servidor mantem a ligação aberta depois de enviar uma resposta
- Tempo total = RTT + (RTT + tempo de transmissão) * num_objetos

Conceito #2: HTTP é protocolo "stateless"

- HTTP e um protocol stateless
 - O servidor não guarda a informação
- Vantagem
 - E mais simples , escala melhor
- Desvantagem
 - O servidor não pode utilizar os pedidos passados do cliente
- O estado e mentido atraves de cookies→ o cliente guarda o estado



Cookies

- Os cookies são usados para : →
 - autorização
 - lista de compras
 - recomendações
-

Conceito #3: Web Caching

- Web caching →
 - Objetivo: satisfazer pedidos dos clientes sem utilizar o servidor original
 - Utiliza a configuração do browser para aceder a Web através de cache
 - O browser envia todos os pedidos HTTP para a cache →
 - Se esta em cache , a cache retorna o objeto
 - Se não , o cache pede o servidor original e encaminha para o cliente
 - Reduz o tempo de resposta e quantidade de trafego para o servidor original
-

GET condicional

- Objetivo: de não enviar o pedido se a cache tem a versão atualizada →
 - Remover atraso de trafico
 - Reduz utilização da ligação
 - A cache especifica a data do que esta em cache no pedido HTTP →
 - If-modified-since: <date>
 - A resposta de servidor não traz objeto se a copia em cache estiver atualizada →
 - HTTP/1.0 304 Not Modified
-

FTP

FTP: file transfer protocol

- E utilizado para enviar ficheiros para/de terminais remote
 - Modelo cliente/servidor →
 - cliente: inicia a transferencia
 - servidor: terminal remote
-

FTP: canais de controlo e dados separados

- Cliente FTP contacta servidor FTP no porto 21 usando o TCP →

- Ligação de controlo: autorização, pesquisa da diretoria remota e envio de comandos
 - Quando o servidor recebe um comando de transferência de ficheiros, a primeira ligação TCP → Ligação de dados: ligação para os ficheiros
 - Princípios importantes
 - A ligação de controlo é "out of band"
 - a ligação "in band" é usada para transmissão de dados
 - O servidor FTP mantém "estado": a diretoria atual, autenticação anterior
-

Correio eletrónico: POP3,IMAP,SMTP

Correio eletrónico

- Componentes:
 - Agentes do utilizador
 - Servidores de correio
 - SMTP: simple mail transfer protocol
 - Agente de utilizador: →
 - Cliente de correio eletrónico(outlook,etc)
 - Ler e escrever mensagens
 - Servidor de correio→
 - Guardam as mensagens destinadas ao utilizador
 - O protocolo SMTP é usado entre servidores de e-mail para enviar as mensagens→
 - Também : cliente → servidor
 - mas não : servidor → cliente
 - SMTP →
 - Modelo : cliente → servidor
 - Usa TCP , porto 25 , fiabilidade
 - Transferência direta do servidor emissor para o servidor destino (não passa por intermediários)
 - Interação do tipo comando-resposta →
 - Como HTTP e FTP
 - Comandos: texto ASCII
 - resposta: código (status code)e frase
 - Há 3 fases na transferência de um correio→
 - Handshaking → Transferência de mensagens → Fecho
-

SMTP: comparação com HTTP

- O SMTP usa ligações persistentes(Envia muitas mensagens para o mesmo destino → mesma ligação TCP)
 - O SMTP e um protocol de tipo push→ push a mensagens para o destino
 - o protocol de tipo pull e o http , o cliente pull o objeto que esta no servidor
 - SMTP: varios objetos são enviados atraves de uma unica mensagem
 - o http → cada objeto e encapsulado numa mensagem de resposta
 - SMTP e HTTP usam comandos e respostas ASCII (o SMTP obriga o corpo da mensagem ser tambem ASCII)
-

Protocolos de acesso ao correio

User1 →(SMTP)→Servidor do emissor →(SMTP)→Servidor do destinatario →(POP,IMAP) → User2

- **POP3** (Post Office Protocol Version 3)
 - Protocol simples , stateless
 - **IMAP**(Internet Mail Access Protocol) →
 - Protocol com mais funcionalidade: manipulação de mensagens no servidor
 - Stateful: necessário manter estado do utilizador entre sessões
-

DNS

DNS(Domain name system)→

- Os computadores também tem identificadores, eles usam os endereços IP (os nomes do sites são para os humanos perceber)
-

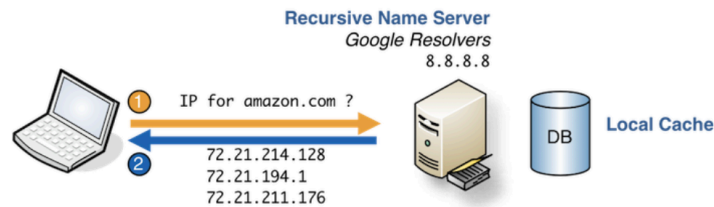
Como mapear entre um nome e um endereço IP?→

- Resposta: → Domain name system
 - Uma base de dados distribuida implementada como hierarquia de servidores de nomes
 - Protocolo de nivel de aplicação
 - Os terminais e os servidor comunicam para traduzir un endereço
 - Corre sobre UDP no porto 53
 - Servidor DNS; uma maquina UNIX que corre o software BIND
-

DNS: Serviços

- Serviços

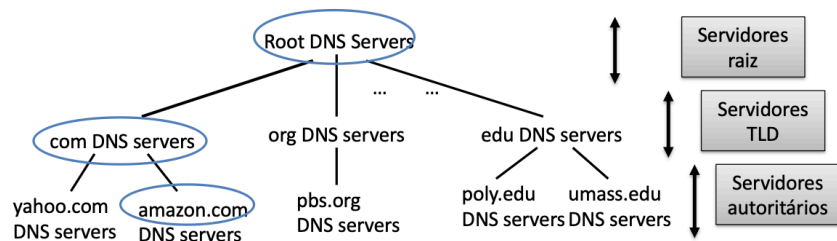
- Traduzir de endereços (traduz o nome de maquina para endereço IP e vice versa)
- Multiplos nomes (host aliasing)
 - nome proprios e alternativos
- Nomes de servidor de correio:
 - Quando temos servidores Web replicados podemos ter multiplos endereços IP a corresponder a um unico nome



Porque não centralizar o DNS?

- Tornase un SPOF(Single Point Of Failure)
- Todo o trafego vai ser concentrado nesse ponto
- Uma base de dados centralizada fica distante de maioria dos clientes
- Manutenção de uma base de dados tão grande. seria uma tarefa muito complexa

DNS: BD distribuida e hierárquica



Servidores TLD e servidores autorizados

- Servidores top-level domain(TLD) → Resposavel pelo dominios com (org , com edu, etc) e todos dominios top leevel de paises (pt,e,md)
- Servidores DNS autoritario(authoritative)

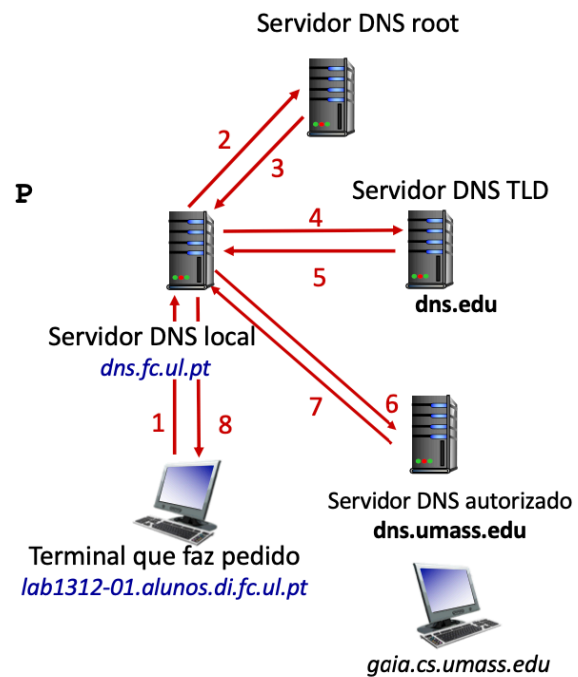
- Servidores DNS da organização responsável pelo domínio
- Gerido pela organização ou por um fornecedor de serviços

Servidor de DNS local

- Não pertence estritamente a hierarquia
- Cada ISP(empresa , residencial) tem pelo menos um servidor de nomes local
- Quando uma máquina faz um query ao DNS este é enviado ao servidor de nomes local
 - Intermediário que reencaminha os queries para a hierarquia de servidores
 - Tem um cache local com pares de endereço\nomes mais recentes

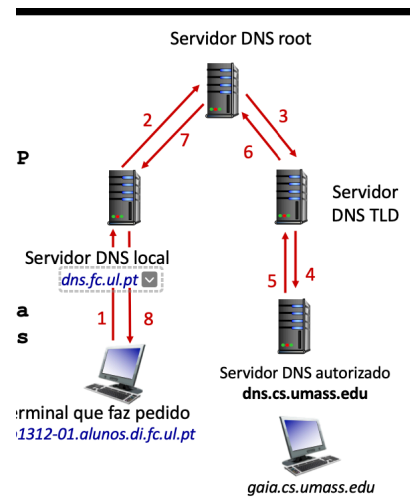
Resolução de nomes iterativa

- Query iterativo:
 - O servidor contactado responde com o nome do servidor a contactar
 - “Não sei mas sei quem sabe”



Resolução de nomes recursiva

- Query recursivo:
 - A responsabilidade da resolução de nomes passa para o servidor de nomes contactado



DNS: cache e atualização de registos:

- Sempre quando um servidor recebe um novo nome, ele guarda esse localmente num cache
 - As entradas no cache são removidas depois de um tempo determinado
 - A informação sobre os servidores TLD são guardados no cache dos servidores locais
- As entradas na cache podem estar desatualizadas
 - quando um endereço IP muda a internet pode demorar a perceber
- Mais um serviço "best effort"

	Requisitos	Protocolo de transport	Estado	Modelo	Ligação de control	Tipo de operação
HTTP	Fiabilidade	TCP	Stateless	Cliente-servidor	In-band	pull
FTP	Fiabilidade	TCP	Stateful	Cliente-servidor	out-of-band	pull
POP3	Fiabilidade	TPC	Stateless	Cliente-servidor	in-band	pull
IMAP	Fiabilidade	TPC	Stateful	Cliente-servidor	in-band	pull
SMTP	Fiabilidade	TCP	Stateful	Cliente-servidor	in-band	push
DNS	Fiabilidade	UDP	Stateless	Cliente-servidor	in-band	pull

Registos DNS

DNS General resource record format

Name	TTL	Class	Type	Rdata
------	-----	-------	------	-------

Type = A(address):

- **Name** e o hostname
- **Rdata** e o endereço IPv4

Type = NS (name server)

- **Name** e o domínio(ul.pt)
- **Rdata** e o hostname do servidor autorizado para o domínio

Type = CNAME(canonical name)

- **Name** e un nome alternativo para outro nome(www.ibm.com → servereast.buckup2.ibm.com)
- **Rdata** pode ser o nome oficial

Type=MX(mail server)

- **Rdata** contem o nome do servidor de mail associado a **Name**

Aplicações multimédia

Multimédia e Qualidade de Serviço:

- Aplicações multimedia : Audio e vídeo em rede
- Qualidade de Serviço(QoS): A rede oferece às aplicações o nível de desempenho que estas necessitam

Características fundamentais

- São tipicamente sensíveis ao atraso
 - Atraso end-to-end → Tempo total que os dados multimedia demoram a ser enviados do emissor ao recetor
- Atraso jitter → Variação no atraso dos pacotes do mesmo fluxo de dados
- São tolerantes a perdas

Classes de aplicações multimédia

- Difusão de conteúdo armazenado
 - modelo de cliente-servidor
 - Cada cliente recebe o seu fluxo de dados
- Difusão ao vivo
 - modelo cliente-servidor
 - O emissor envia os mesmos dados para varios clientes em simultaneo

- Interativas em tempo real
 - modelo P2P
 - cada utilizador interage com outro utilizador
-