

Infraestrutura ComputacionalIII

DNS - Domain Name System

E

DHCP - Dynamic Host Configuration Protocol

Luis C.E. Bona (bona@inf.ufpr.br)

Slides parcialmente baseados no livro:

Computer Networking: A Top Down Approach. Jim Kurose, Keith Ross



Visão geral

❖ Neste módulo

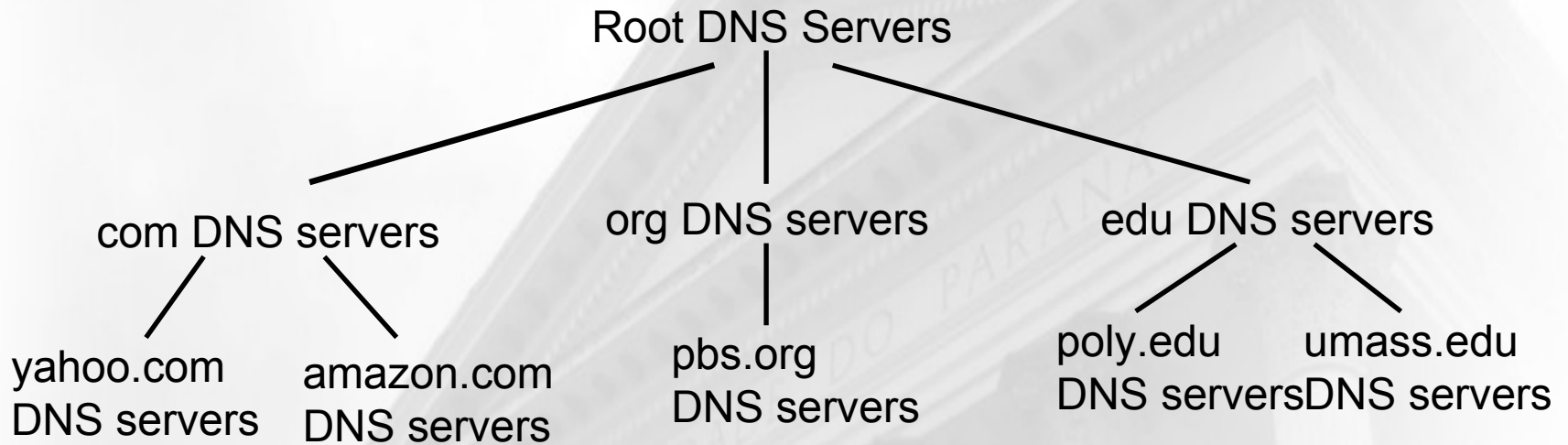
- Conceitos principais de redes de computadores
- Configuração dos principais serviços de rede em servidores Linux
- Ferramentas e aplicações de rede
- Conceitos de Nuvens Computacionais e virtualização



DNS

- ❖ Necessidade de ter nomes para os “hosts”
- ❖ Na Internet inicial: */etc/hosts*
- ❖ Domain Name System
 - É uma aplicação, parte da filosofia da Internet
 - Banco de dados distribuídos

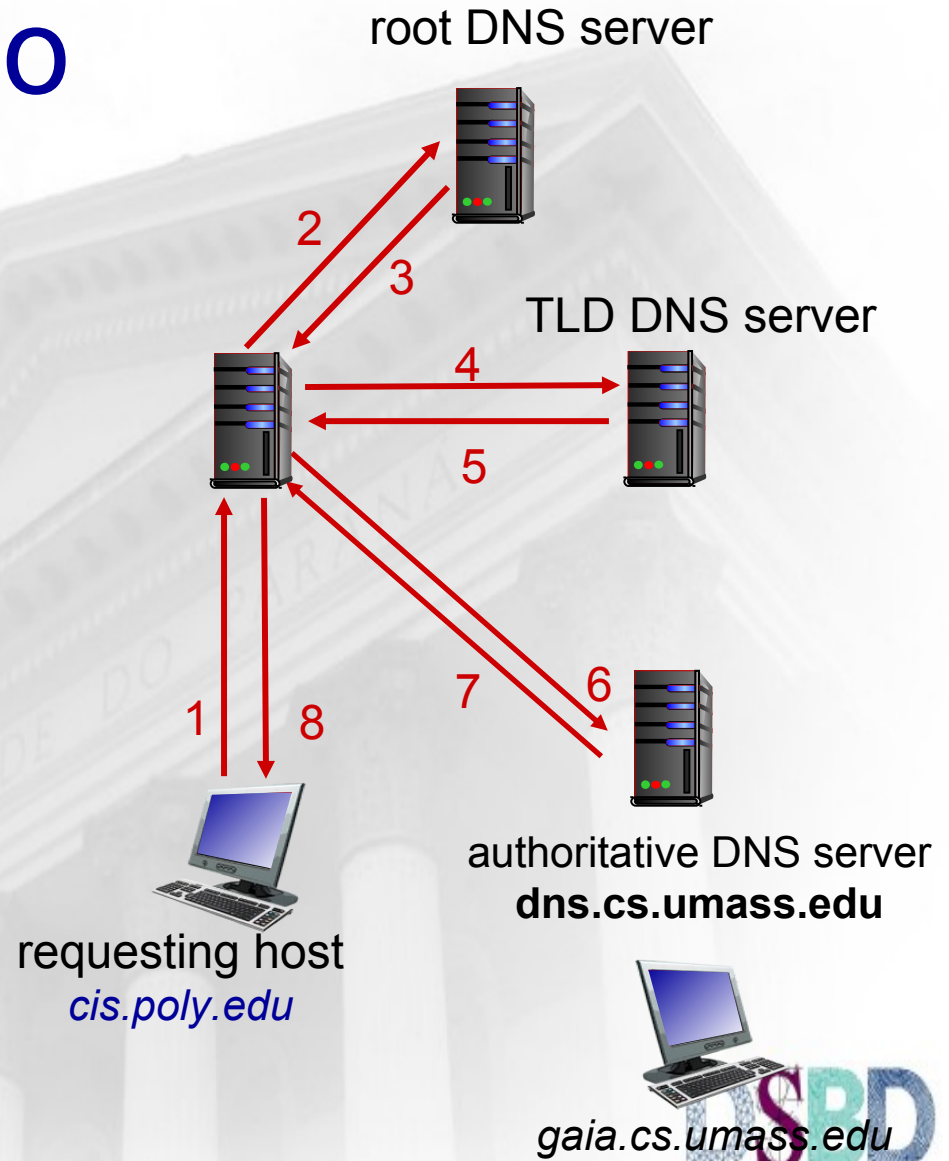
DNS, organização



DNS

- ❖ Top-level domain (TLD) servers
 - Responsáveis pelos domínios de países, e outros como com, org, net, edu
- ❖ Authoritative DNS
 - Organizações que possuem seus próprios servidores DNS
- ❖ DNS local
 - Normalmente o chamado “default name server”

DNS, exemplo



Dig

- ❖ O dig é um comando que permite fazer consultas diretamente a um servidor DNS
- ❖ O exemplo mais simples possível
 - `dig algum.end.qq`
- ❖ Mas pode ser muito “ruidoso”, tente:
 - `Dig +noall +answer`

Dig

- ❖ Principais tipos de registro: A, NS, MX
- ❖ O MX é o servidor de e-mail. Como descobrir qual máquina é responsável pelos e-mails de um determinado domínio?
 - `dig mx algumdominio`
- ❖ O NS é o servidor de nomes “autoridade” de um domínio. Por exemplo:
 - `dig ns dsdb.c3sl.ufpr.br`

Dig

- ❖ Descubra qual host é responsável pelos e-mails: @usp.br e @inf.ufpr.br
- ❖ Quem é o servidor de dns responsável pelos domínios:
 - ufpr.br, inf.ufpr.br e fisica.ufpr.br
 - E por estatistica.ufpr.br? O que você notou de diferente?

Dig

- ❖ Escalando a “árvore” do DNS
 - Um cliente interno do DNS recebe uma resposta recursiva do servidor, ou seja, o servidor dá a resposta “final”
 - Mas o servidor normalmente se comporta de forma diferente para clientes externos
- ❖ Você pode acessar outros servidores com o dig: “@server”
- ❖ Por exemplo o a.dns.br:
 - `dig @a.dns.br NS ufpr.br`
 - `dig @a.dns.br NS inf.ufpr.br`

Dig

- ❖ Por exemplo o a.dns.br:
 - `dig @a.dns.br NS ufpr.br`
 - `dig @a.dns.br NS inf.ufpr.br`
- ❖ Por que as repostas são quase idênticas? O servidor de domínio para ufpr.br e inf.ufpr.br não são diferentes?
- ❖ Observe ainda:
 - `dig @a.dns.br www.ufpr.br`

Dig

- ❖ Mas qual é a raiz do DNS? E com quem nós conseguimos esses endereços?
- ❖ Experimente a opção `+trace`
- ❖ **Bonus** : você consegue fazer uma sequencia de comandos usando `dig` para reproduzir o resultado do `+trace`?

DNS: cache

- ❖ Como garantir escalabilidade?
 - Diminuindo o número de requisições
- ❖ Introduzir um tempo de vida para cada resposta (TTL)
- ❖ Quanto deve ser esse tempo?

DNS Dinâmico

- ❖ Existem serviços gratuitos e pagos na rede que oferecem serviços de DNS dinâmico
- ❖ Este serviço permite que você registre um nome e oferecem uma interface para atualização do IP
- ❖ A dificuldade é o fato do IP ser dinâmico
 - Ainda existem outras quanto ao Firewall, IPv4 ou IPv6. Mas isso veremos depois...

Cache + DNS dinâmico

❖ Execute:

- `dig mau.duia.eu +noall +answer; sleep 10; dig mau.duia.eu +noall +answer`
- Execute novamente....

❖ O que você notou comparando esses resultados? E comparando com esse abaixo?

- `dig a.dns.br +noall +answer`

DNS Dinâmico

- ❖ Existem vários serviços gratuitos (ainda que limitados)
- ❖ Um experimento simples:
 - Crie uma conta no noip.com.escolha um domínio qualquer
 - Aponte o registro A para o IP da dsdb: 200.236.31.192
 - Comprove o funcionamento
- ❖ Nesse caso é simples porque nosso IP não muda; Como atualizar automaticamente?



DNS Dinâmico

- ❖ Muitas vezes os provedores de serviço de DNS dinâmico oferecem um *daemon*. Ele monitora as trocas de IP e reconfigura os endereços
- ❖ A parte principal é uma API:
 - <https://www.noip.com/pt-BR/integrate/request>
- ❖ Teste mudar para outro IP usando a API em um BROWSER
- ❖ Agora teste com `wget`
 - Isso pode ajudar:
 - <https://askubuntu.com/questions/29079/how-do-i-provide-a-username-and-password-to-wget>

Juntando tudo

- ❖ Agora o `usuario@dsdb` tem acesso a outras contas na mesma máquina `user{1-50}`
- ❖ Gere uma nova chave para o seu `userX` e instale a chave pública no próprio `userX`
 - Quer dizer agora o `ssh` de `userX@dsdb` para `userx@dsdb` não deve pedir senha

Juntando tudo

- ❖ Faça um forward da porta 80 da swamp para um porta local da maquina dsdb (*sugestão 80XX – onde xx é o número do userX. Por exemplo user1 usará a porta 8001*)
 - Quer dizer que você vai fazer um ssh de userX@dsdb para userX@dsdb para só para ter o tunelamento
 - Se tudo deu certo alguém responde ao "telnet localhost 80XX"
 - Mas de seu terminal funciona? Quer dizer dizer "telnet dsdb 80XX" é respondido?
 - Tente novamente com: -L *:
- ❖ Agora, se Deus ajuda os que estudam sábado o host dsdb atende externamente na porta 80XX

Juntando tudo

- ❖ Estamos quase lá!
- ❖ Certifique-se que o IP do no-ip está apontando para a dsdb.c3sl.ufpr.br
- ❖ Se tudo deu certo, abra um browser e digite a seguinte URL:
 - <http://seu.dominio.dinamico:80XX/>
 - Wow! O servidor da swamp agora aparece como fosse "meu"



E o cliente DNS?

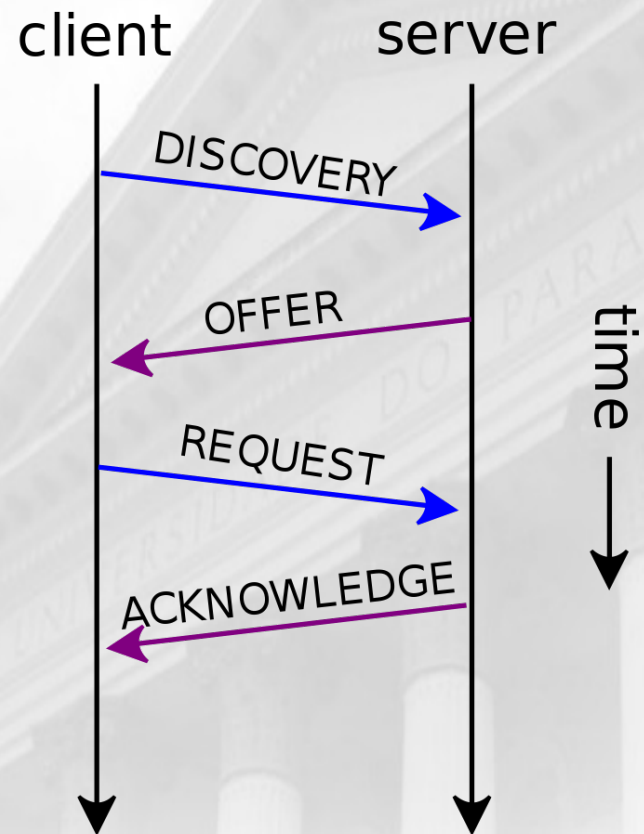
- ❖ As linguagens oferecem suporte ao protocolo DNS através de bibliotecas
- ❖ Essas bibliotecas interagem com o SO para saber (entre outras coisas) qual o servidor DNS que deve ser utilizado
- ❖ No Linux a informação básica está no:
`/etc/resolv.conf`
- ❖ As distribuições locais apontam a resolução de nomes para um servidor de cache local
- ❖ O seu servidor de nomes costuma ser indicado pelo provedor usando DHCP



DHCP

- ❖ Dynamic Host Control Protocol (DHCP)
 - Conectar um host na Internet requer a configuração de vários parâmetros: gateway, endereço e máscaras de rede, servidor de dns, etc...
 - Boa parte dos hosts obtém esses endereços automaticamente da rede através do DHCP

DHCP



DHCP

- ❖ Você pode observar os logs de interação do seu host com o DHCP
- ❖ Procure por `dhclient` no arquivo `/var/log/daemon.log`
- ❖ Normalmente os servidores de DHCP oferecem configurações de duas maneiras:
 - Estática, considerando o endereço físico da interface de rede (no exemplo acima `00:16:c8:ec:2f:fc`)
 - Dinâmica, oferecendo endereços de forma independente do endereço físico (ISP, são um exemplo)