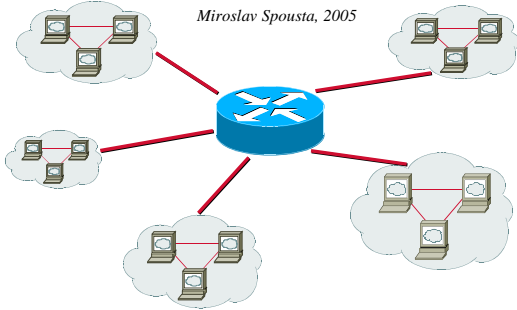


## Počítačová síť II

### 15. DNS

Miroslav Spousta, 2005



1

## Domain Name System

- DNS = doménový jmenný systém
- IP používá číselné adresy (32 nebo 128b)
  - těžko zapamatovatelné
  - tyto adresy slouží pro funkce IP, TCP, UDP
    - vlastní celého Internetu, až na lidi
- příklad ze snadno zapamatovatelných jmen na IP adresy
  - a obrácen
- hierarchická databáze
  - příklady obměsní
- mechanismus pro převod adres
  - jmenové servery
- pravidla pro vytváření jmen

2

## Historie

- původní řešení: centralizované
  - v zárodkovém ARPANETu
  - existovala centrální autorita, která přidělovala jména (a jejich přiřazení k IP adresám)
  - aktualizace informací probíhala pomocí distribuce tzv. hosts souboru
- s nárůstem velikosti souboru přestala tato možnost vyhovovat
  - problém s velikostí a rychlostí aktualizací
  - bylo potřeba databázi distribuovat (rozložení zátěže i pravomocí)
- poátkem 80. let začal vznikat DNS
- dodnes existuje soubor /etc/hosts
  - obsahující statické přiřazení jmen k adresám
  - ale spravuje se pouze lokálně (pro jeden uzel)
  - hodí se pro malé/ad-hoc sítě

3

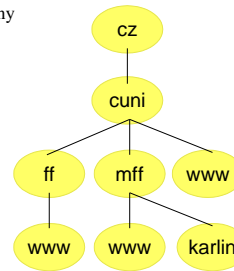
## Vlastnosti DNS

- DNS je jedna z nejdůležitějších služeb Internetu
  - bez ní Internet fakticky nefunguje, pro lidi je však nepoužitelný
  - například e-maily nebudou fungovat v běhu
- robustnost
  - odolnost proti výpadku uzlu (DNS server) – replikace
  - distribuované místo – odolnost proti výpadku sítě / spojení sítě, rozložení zátěže
- distribuce pravomocí
  - aby adresy nemusely přidělovat centrální autorita
  - mnoho subjektů, které si chtějí spravovat jména samy (instituce, firmy, ...)
- snadno zapamatovatelné, logické názvy
  - jednoduchá hierarchická struktura, jména jsou jednoznačná
  - rozdělení prostoru na několik částí – úrovně

4

## Hierarchie

- jmenný prostor je rozdělený na části – domény
- vytváří stromovou strukturu
  - kořen je počátek prostoru
  - cesta od kořene k listu udává plně jméno uzlu
  - neboli fqdn – fully qualified domain name
- doména je jedna úroveň stromu
  - je jednoznačně určena cestou ke kořeni (.)
  - jsou v ní definovány další domény nebo jména – nižší úrovně
  - úrovně se sloučí od 1 (ale říká se jí nejvyšší)
  - domény se zapisují za sebou, jednotlivé úrovně jsou odděleny tečkou
  - zapisuje se zprava doleva (vpravo je nejvyšší doména – úroveň 1)



5

## Doménová jména

- mohou obsahovat velká a malá písmena anglické abecedy, číslice a pomlčku
  - pomlčka nesmí stát na začátku ani na konci jména
- max. velikost na jedné úrovni je 63 znaků
- velká a malá písmena se nerozlišují (Sun.COM == sun.com)
- počet úrovně (poddomén) není explicitně omezen
  - ale fqdn musí mít maximálně 255 znaků
- čemu odpovídá jedna úroveň (doména)?
  - organizační členění? prostorovému rozdělení?
  - není explicitně určeno, záleží na uživateli
- velikost domény souvisí s její udržovatelností
  - v rámci domény musí být jména jednoznačná
  - doména je většinou spravovaná centrálně, z jednoho místa

6

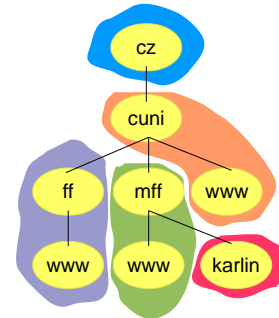
## Domény

- TLD (top level domain – domény nejvyšší (první) úroveň)
  - jsou centrálně řízené a spravované
- například .cz, .sk, .uk, .com, .org, .net, .us
- jejich správa (přidělování domén nižší úroveň) je delegováno
- zájemci o nižší (druhou) úroveň se obracují na národní registry
  - CZ: NIC.CZ, poskytovatelé připojení
- v rámci své domény si subjekty přidělují poddomény libovolně
  - malé organizace mají většinou všechna jména přímo ve svých doménách
  - větší domény mohou dělit (cuni.cz => ff.cuni.cz, natur.cuni.cz, mff.cuni.cz, ...)

7

## Zóny

- zóna: doména s (některými) poddoménami, která je jednotně spravovaná
  - tedy jedním subjektem (např. ISP, firmou, ...)
- z podstromu mohou být „vykousnuty“ některé podstromy
  - které spravuje někdo jiný, tedy pravomoc (autorita) jsou delegovány na jiný subjekt
- zóna je zpravidla uložena v jednom souboru (tzv. zónový soubor)
  - v něm jsou položky vztahující se k dané zóně – resource records (RR)



8

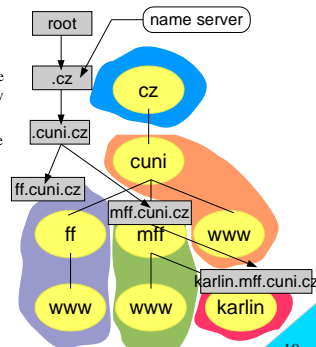
## Name servery

- pro každou doménu je potřeba uchovávat různé informace
  - například na IP adresy (uzel se jménem pročte má adresu 1.2.3.4)
  - který uzel přijímá poštu pro danou doménu
  - které name servery jsou dostupné pro doménu/poddoménu
  - ...
- informace jsou distribuovány po celém Internetu
  - jsou soustředěny na *name serverech*
  - které mají informace o určité doméně (zóně)
  - a odpovídají na dotazy klientů
- každá doména má svůj name server
  - ale může ho sdílet s jinou doménou (jeden name server pro více domén)

9

## Name servery

- name servery tvoří stromovou hierarchii
  - kořenem stromu je tzv. kořenový (root) name server, který uchovává odkazy na všechny name servery pro TLD
  - ve skutečnosti nese toto name server více (a.root-servers.net – m.root-servers.net)
  - odkazy na nižší zóny se šíří pomocí tzv. *glue records*
- name server uchovává informace pro celou zónu
- pro jednu doménu existují aspoň dva name servery
  - ochrana proti výpadkům uzlů/sítí



10

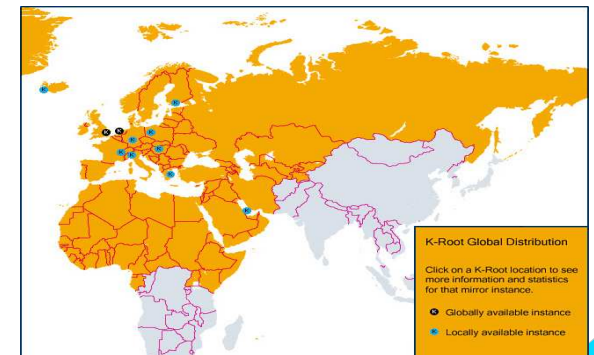
## Root name servers

### Map of the Root Servers



11

## Root name servers



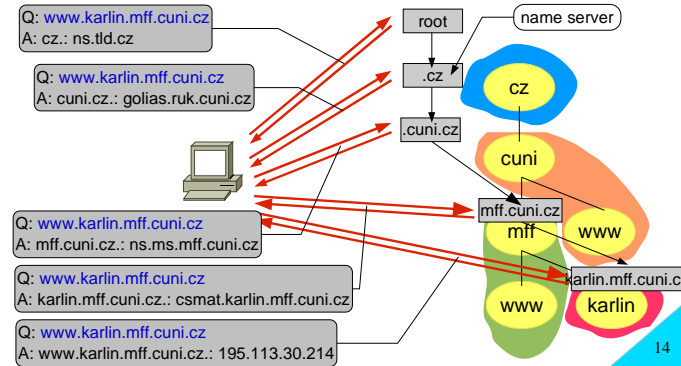
12

## Primární a sekundární NS

- primární NS pro doménu
  - pro každou doménu existuje právě jeden
  - je na něm uložen zónový soubor
  - slouží jako primární zdroj dat
- sekundární NS
  - obsahuje obraz dat pro doménu primárního NS
  - pravidelně si aktualizuje tento obraz dat (pomocí zone transfer)
  - odpovídá na dotazy stejně jako primární NS
  - může být od primárního NS dostatečně vzdálen (v jiné síti)
  - pro každou doménu by měl existovat alespoň jeden
- primární i sekundární NS jsou autoritativní pro danou doménu
  - neboli vždy vrací platné informace (aspoň by měly)

13

## DNS dotaz



14

## DNS dotaz: www.cuni.cz

```

; <<> DiG 9.2.1 <<> +trace www.cuni.cz
;; global options: printcmd
.      272471 IN      NS      A.ROOT-SERVERS.NET.
.      272471 IN      NS      B.ROOT-SERVERS.NET.
.      272471 IN      NS      C.ROOT-SERVERS.NET.
.
.      272471 IN      NS      I.ROOT-SERVERS.NET.
.      272471 IN      NS      J.ROOT-SERVERS.NET.
.      272471 IN      NS      K.ROOT-SERVERS.NET.
.      272471 IN      NS      L.ROOT-SERVERS.NET.
.      272471 IN      NS      M.ROOT-SERVERS.NET.
;; Received 324 bytes from 195.113.0.2#53 (195.113.0.2) in 36 ms
cz.    172800 IN      NS      NS2.NIC.FR.
cz.    172800 IN      NS      NS.RIPE.NET.
cz.    172800 IN      NS      SUNIC.SUNET.SE.
cz.    172800 IN      NS      NS-EXT.VIX.COM.
cz.    172800 IN      NS      NS.TLD.cz.
cz.    172800 IN      NS      NSS.TLD.cz.
;; Received 269 bytes from 198.41.0.4#53 (A.ROOT-SERVERS.NET) in 122 ms
cuni.cz. 18000 IN      NS      goliass.ruk.cuni.cz.
cuni.cz. 18000 IN      NS      ns.ces.net.
;; Received 94 bytes from 192.36.125.2#53 (SUNIC.SUNET.SE) in 37 ms
www.cuni.cz. 86400 IN      CNAME   tarantula.ruk.cuni.cz.
tarantula.ruk.cuni.cz. 86400 IN      A       195.113.0.123
ruk.cuni.cz. 86400 IN      NS      cuce.ruk.cuni.cz.
ruk.cuni.cz. 86400 IN      NS      goliass.ruk.cuni.cz.
ruk.cuni.cz. 86400 IN      NS      ruzenka.prf.cuni.cz.
;; Received 187 bytes from 195.113.0.2#53 (goliass.ruk.cuni.cz) in 0 ms
    
```

15

## DNS resolver

- knihovny, které zajišťují kontakt s DNS serverem
  - DNS: klient – server architektura
  - klient přijímá od aplikací dotazy a pokládá je NS
  - vyhodnocuje a vrací odpovědi aplikaci
  - obvykle forma sdílené knihovny
- name server
  - odpovídá na dotazy ohledně vlastních domén (vlastní zóny)
  - a eš dotazy od klienta, hledá a vrací odpověď
  - musí obsahovat také resolver
  - dotazy si ukládá v cache (non-authoritative answer)

16

## Optimalizace DNS

- redundanci name servery
  - primární a sekundární
  - rozkládání zátěže
- cachování dotazů
  - nameserver, který vyšetří dotaz pro klienta si ho uloží do své cache
  - každá položka má dobu, po kterou může být stačena
  - při dalším dotazu vrátí záznam z cache (největší optimalizace)
    - tzv. neautoritativní odpověď
- caching only name server
  - NS, který pouze vyšetří dotazy
  - není autoritativní pro žádnou zónu
- forwarding name server
  - ani ne vyšetří dotazy, jen přeposílá na jiný NS

17

## Resource records

- obsah DNS databáze: položky – RR (resource records)
- každá položka RR se skládá z
  - jméno (identifikace položky)
  - TTL (jak dlouho může klient držet záznam v cache)
  - class (třída) – pro Internet IN
  - typ (A, NS, MX, ...)
  - RDATA – data, interpretace závisí na typu
- položky jsou většinou uloženy v jednoduchých textových databázích
  - zónovém souboru
  - jedna položka na jednom řádku
  - pochází z BIND

18

## Typy RR

- SOA: Start of Authority – informace o zón
  - na začátku každého zónového souboru
  - e-mail na osobu zodpovědnou za danou zónu
  - časové údaje pro synchronizaci mezi primárním a sekundárním NS
- A: záznam na IP adresu
- CNAME: alias pro jiné jméno
- NS: Name Server pro danou doménu (FQDN)
- MX: Mail eXchange
  - mail server pro doménu
  - može jich být více, liší se prioritou, čím nižší číslo, tím vyšší priorit
- PTR: ukazatel jinam do jiné části DNS stromu
  - pro reverzní záznam

19

## Zóna

```
$TTL 3D
@          IN      SOA     ns.linux.bogus. hostmaster.linux.bogus. (
199802151 ; serial, today's date + serial #
8H        ; refresh, seconds
2H        ; retry, seconds
4W        ; expire, seconds
1D        ; negative TTL
);
;
; NS
; NS ns                ; Inet Address of name server
MX 10 mail.linux.bogus. ; Primary Mail Exchanger
MX 20 mail.friend.bogus. ; Secondary MX
;
localhost A 127.0.0.1
ns         A 192.168.196.2
mail      A 192.168.196.4
```

20

## DNS protokol

- běží nad UDP (TCP), port 53
  - na dotazy na jméno preferován UDP
  - na transfery zón se používá TCP
- stejný formát paketu pro dotaz i odpověď
- Header: hlavička požadavku
  - jestli se jedná o dotaz i odpověď, kolik položek je v které části
- Question: dotaz (třída, typ, jméno)
- Answer: odpověď (RR odpovědi, může jich být více)
- Authority: NS, odkud záznam(y) pochází
- Additional: další informace, které souvisí s dotazem
  - například záznam NS/MX na IP adresu

Header
Question
Answer
Authority
Additional

21

## DNS dotaz

```
; ->HEADER<- opcode: QUERY, status: NOERROR, id: 3018
; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 3, ADDITIONAL: 3
; QUESTION SECTION:
idnes.cz.             IN      A
; ANSWER SECTION:
idnes.cz.             2940   IN      A       194.228.207.200
; AUTHORITY SECTION:
idnes.cz.             2940   IN      NS      ns2.mafra.cz.
idnes.cz.             2940   IN      NS      ns.mafra.cz.
idnes.cz.             2940   IN      NS      ns2.tel.cz.
; ADDITIONAL SECTION:
ns.mafra.cz.          1579   IN      A       194.228.207.7
ns2.tel.cz.           1712   IN      A       194.228.2.1
ns2.mafra.cz.         1579   IN      A       62.209.231.7
```

22

## nslookup

- příkazy pro testování DNS
- nslookup** může fungovat také v interaktivním módu
- hodí se pro jednoduché testování DNS
- umí automaticky převést i obráceným směrem: IP -> jméno
- provozuje BIND (Berkeley Internet Name Domain)
- najdete je všude (Linux, UNIX, Windows, ...)
- pokročilejší nástroj: **dig**

```
bug:~# nslookup
> idnes.cz
Server:             62.24.64.2
Address:            62.24.64.2#53

Non-authoritative answer:
Name:               idnes.cz
Address:            194.228.207.200
> quit
Server:             62.24.64.2
Address:            62.24.64.2#53

** server can't find quit: NXDOMAIN
>
```

23

## Reverzní záznamy

- chceme obrácenou službu: z IP adresy získat doménové jméno
  - například web/mail server chce do logu zapsat, odkud přišel požadavek
- funguje pomocí záznamu PTR
- speciální doména: in-addr.arpa
  - v ní je po jednotlivých bytech uveden celý IPv4 rozsah
  - například pro adresu 195.113.31.125 je to 125.31.113.195.in-addr.arpa
  - bajty jsou uvedeny v obráceném pořadí (!)
  - pro toto jméno existuje záznam pomocí PTR na jméno: artax.karlin.mff.cuni.cz

```
1.33.168.192.in-addr.arpa 1D IN PTR test1.apus.cz.
2.33.168.192.in-addr.arpa 1D IN PTR test2.apus.cz.
3.33.168.192.in-addr.arpa 1D IN PTR test3.apus.cz.
```

24

## CIDR a reverzní záznamy

- výše uvedený postup se příliš nehodí pro CIDR
  - len ní na podsít nemusí být po byte, je problém s autoritou nad danou zónou
- řešení: místo PTR ukazatel se uvede odkaz (glue records) na NS nižší úrovn

```
1.33.168.192.in-addr.arpa. IN NS ns
2.33.168.192.in-addr.arpa. IN NS ns
3.33.168.192.in-addr.arpa. IN NS ns
ns.33.168.192.in-addr.arpa IN A 192.168.33.1
```

```
@ 1D IN SOA ns hostmaster.in-addr.arpa. (0 3600 120 3600 3600)
@ 1D IN NS ns
ns 1D IN A 192.168.33.1
1.33.168.192.in-addr.arpa 1D IN PTR apus.example.
2.33.168.192.in-addr.arpa 1D IN PTR prunella.example.
3.33.168.192.in-addr.arpa 1D IN PTR otus.example.
```

25

## Diakritika v DNS

- v původním návrhu DNS se s národními znaky nepočítalo
  - pouze anglická abeceda, čísla a „-“
- byla snaha povolit národní znaky
  - např. košíky.cz
- řešení: IDN (Internationalized Domain Names)
  - RFC 3490
  - v doménových jménech je možné používat podmnožinu UNICODE
  - nadstavba nad DNS, překládá se na straně klienta (neboli v DNS se nic nemění)
- jméno se nejprve přeloží do speciální formy (ne-ASCII znaky se zakódují podobně jako UTF-7)
- toto zakódované jméno se použije pro DNS dotaz
  - již je v čistém ASCII, jen pro člověka nečitelné

26

## IDN Příklad

- nejprve se jméno (v UNICODE) převeď na kanonický tvar
  - sjednocení variant zápisu (velká a malá písmena)
- poté se jméno převeď na tzv. punycode
  - algoritmus převodu je reverzibilní (je možné převést punycode zpět na UNICODE)
  - přidá se prefix xn--, za ním znaky bez diakritiky, zakódované znaky nakonec
  - např. košíky.cz => xn--koky-wpa6qow.cz
- takto převezené jméno (vyhovuje původním požadavkům DNS) se pošle jako dotaz
- problémy:
  - je potřeba registrovat domény speciálního tvaru
  - ne všude je možné zadat jméno v UNICODE => pak je potřeba používat přeloženou formu
  - pishing

27