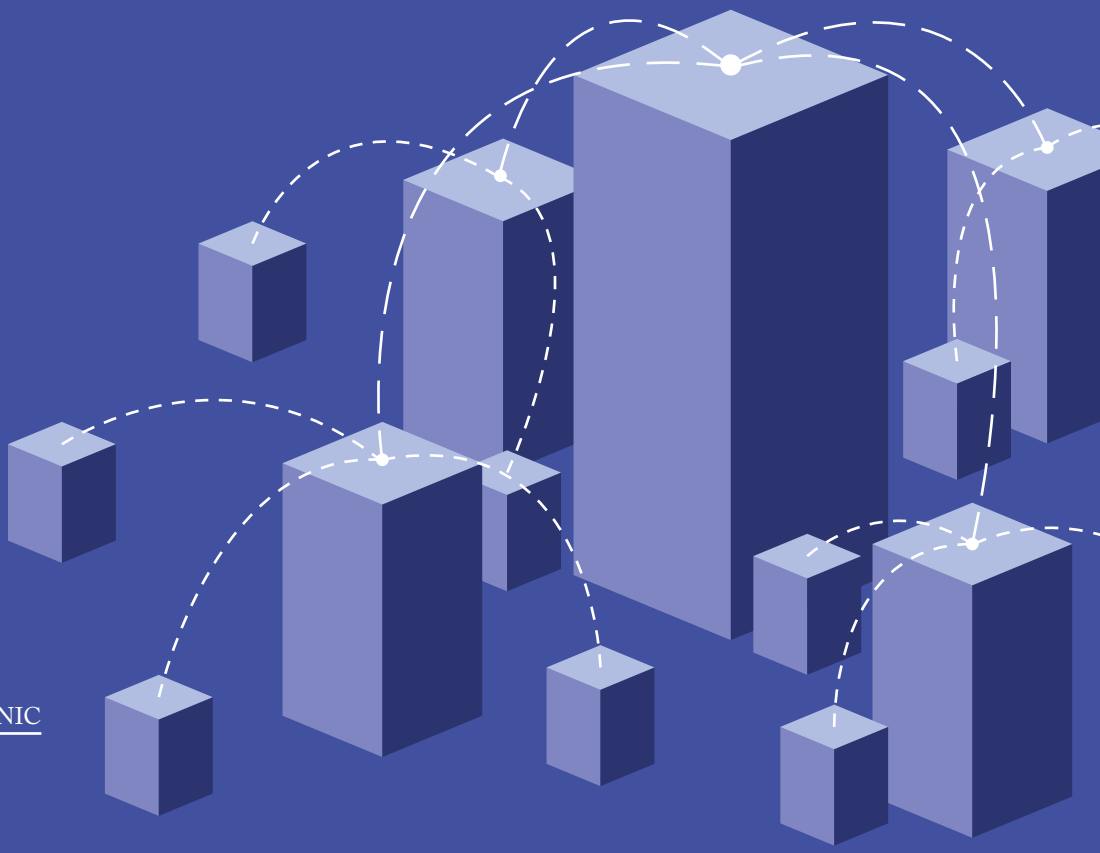
Pavel Satrapa, Ondřej Filip

Domain Name System

Principy fungování DNS a praktické otázky spojené s jeho používáním



Edice CZ.NIC

Domain Name System
Principy fungování DNS a praktické otázky spojené s jeho fungováním

Pavel Satrapa
Ondřej Filip

Vydavatel:
CZ.NIC, z. s. p. o.
Milešovská 1136/5, 130 00 Praha 3
Edice CZ.NIC
www.nic.cz

1. vydání, Praha 2023
Kniha vyšla jako 30. publikace v Edici CZ.NIC.
ISBN 978-80-88168-74-4

ISBN 978-80-88168-74-4

— Pavel Satrapa, Ondřej Filip

Domain Name System

Principy fungování DNS a praktické otázky spojené s jeho fungováním

— Edice CZ.NIC

Předmluva vydavatele

Milí čtenáři,

v Edici CZ.NIC vyšla za patnáct let její existence úctyhodná řada knih pokrývající širokou škálu témat. Knížka *Domain Name System*, kterou právě držíte v ruce, se však jako vůbec první věnuje dennímu chlebu sdružení CZ.NIC, totiž Systému doménových jmen (DNS). Jejím hlavním autorem je Pavel Satrapa, který si za třicet let působení na poli technické literatury vydobyl velké renomé především díky svému charakteristickému stylu, jímž dokáže podat i složitá témata zábavnou a čtivou formou. Druhý autor, Ondřej Filip, pak přispěl svými bohatými praktickými zkušenostmi z řízení a správy systému DNS, jak na národní, tak i mezinárodní úrovni.

DNS jakožto globální distribuovaná databáze byla a je vedle transportního protokolu TCP hlavním pilířem a předpokladem fenomenálního úspěchu Internetu. Jak TCP, tak i DNS se dokázaly během pěti dekád, byť s jistými úpravami a rozšířeními, skvěle vyrovnat jak s raketovým nárůstem počtu uživatelů a přenosových rychlostí, tak i s poměrně zásadními změnami v architektuře Internetu – přechodem od transparentní end-to-end sítě k všudypřítomnému překladu adres (NAT), dodatečným zabezpečením veškeré komunikace, jakož i zavedením nové třídy adres IPv6.

Postupným přidáváním nových funkcí protokolu i typů záznamů v databázi, navíc za pochodu, se z DNS vcelku logicky stal solidní moloch. Pokud se s ním člověk chce seznámit aspoň do té hloubky, aby si v diskusích s experty nepřipadal trapně, musí přečíst bratru tucet dokumentů RFC, ale navíc si osvojit i některé vědomosti a žargon, které se šíří spíše neformálními kanály. Tato kniha je v tomto ohledu velmi vítaným pomocníkem, neboť tuto sumu informací nejen integruje z různých zdrojů, ale navíc přístupnou formou vysvětluje a doplňuje ukázkami a příklady. Velice se mi líbí i skvělé ilustrace a diagramy, které text doprovázejí, jakož i celkové typografické zpracování knihy.

Speciální pochvalnou zmínku zasluhuje část II (*DNS pro starší a pokročilé*), která se věnuje několika opravdu komplikovaným součástem DNS. Nevím o jiném textu, který by tak podrobně a přitom srozumitelně popisoval teorii i praxi bezpečnostních rozšíření DNSSEC (*DNS Security Extensions*). Kapitola 10 pak pěkně vysvětluje složitý algoritmus, jímž se beze změny protokolu DNS realizují doménová jména v neanglických abecedách (*IDN, Internationalized Domain Names*). Jsou zde také vysvětleny důvody a potíže, kvůli nimž prozatím nebyla jména s nabodeníčky v doméně *.cz* zavedena.

Velmi zajímavá je i 7. kapitola, v níž jsou popsány principy, aktéři a historie správy kořenové domény DNS i naší „národní" domény *.cz*. Jako pamětník a aktivní uživatel předchozí československé domény jsem s nostalgií zavzpomínal na ony pionýrské doby, kdy mnozí opravdoví muži a ženy instalovali DNS server *BIND* na dnes již neexistující operační systémy a zřízení nových domén druhé úrovně domlouvali „po sousedsku" s kolegy z VŠCHT.

Knihu mohu doporučit ke studiu i zástupcům netechnických profesí – právníkům, ekonomům, marketérům, ba i politikům – byť z ní třeba přečtou jen pár úvodních kapitol. I z nich se totiž mohou dozvědět nejen jak DNS funguje, ale také proč není dobré tento protokol ohýbat a zneužívat k věcem, pro něž nebyl určen.

Přeji vám příjemné čtení!

Ladislav Lhotka
České Budějovice, srpen 2023

Předmluva

Předmluva

Domain Name System je fascinující hned z několika pohledů. Jedná se o klíčovou službu Internetu, kterou všichni denně využíváme, ale přitom „není vidět“. Automaticky používáme doménová jména k identifikaci webů nebo v poštovních adresách a nezabýváme se tím, jak budou převedena na číselné adresy, jež potřebuje IP pro navázání komunikace.

Druhým fascinujícím aspektem DNS je, jak složitý a košatý systém vyrostl na původně jednoduchých základech. Stále platí RFC 1034 a RFC 1035, která v roce 1987 definovala protokol a datové struktury systému. Od té doby vzniklo (a zaniklo) mnoho typů záznamů, rozhojnily se přenosové protokoly a DNS začalo být využíváno i tam, kde se s ním původně vůbec nepočítalo. V tomhle připomíná samotný Internet, jehož základní principy jsou také velmi jednoduché a nadstavba nad nimi monumentální.

DNS je silně distribuované a o data v něm obsažená se starají miliony správců. Jsme lidé, děláme chyby. Občas nás pozlobí technika, tu a tam se někdo pokouší i cíleně škodit. Navzdory tomu celý tenhle Babylon dlouhodobě velmi dobře funguje. Výpadek DNS je velmi citlivá věc, protože znepřístupní Internet uživatelům i strojům. V globálním měřítku k němu nedošlo nikdy a jakékoli rozsáhlejší výpadky jsou velmi vzácné.

Systém je zkrátka velmi robustní a díky tomu platí, co jsme napsali v prvním odstavci – automaticky používáme doménová jména a nezabýváme se tím, jak je DNS převede na IP adresy. Protože to prostě funguje.

V knize popisujeme, proč a jak DNS funguje. Je rozdělena do čtyř částí: První se věnuje základům. Vysvětlujeme v ní základní pojmy a pravidla komunikace, používané transportní protokoly či konfiguraci klienta a zabýváme se i organizačními záležitostmi, tedy jak je celý systém spravován a jak získat doménu. Ve druhé části se zabýváme pokročilejšími prvky DNS, jako je jeho ochrana pomocí DNSSEC nebo používání národních znaků v doménových jménech. Dostaneme se i k jeho využívání jinými službami či k bezpečnosti protokolu.

Třetí část je poněkud encyklopedická. Probíráme zde různé typy záznamů, do nichž jsou v DNS ukládána data. Je jich přes osmdesát, takže jsme je uspořádali do několika tematických skupin. Čtvrtá část je věnována DNS serverům. Nemůže chybět dominantní BIND, s nímž je historie DNS do velké míry spjata. Kromě něj jsme se pokusili představit několik dalších, které jsou ve větší míře používány. V přílohách pak najdete vybraná RFC a rejstřík pro snazší orientaci v textu.

Děkujeme všem, kteří přispěli ke vzniku tohoto textu. V první řadě si zaslouží poděkování naše rodiny, které statečně snášely naše sepisování a trpělivě nám poskytovaly potřebné zázemí. Za obsahové připomínky děkujeme zejména Ondřeji Caletkovi, Vladimíru Čunátovi, Tomáši Hálovi a Danielu Salzmanovi, jejich návrhy pomohly text vylepšit.

Pavel Satrapa, Ondřej Filip
Liberec, Praha, září 2023

Obsah

Předmluva vydavatele **7**
Předmluva **11**
Obsah **15**

1 Jemný úvod do DNS **25**
1.1 Historie DNS 25
1.2 K čemu slouží 27

I Základní principy **31**

2 Jak to funguje **33**
2.1 Doménový strom 33
2.2 DNS záznamy 37
2.3 Domény a servery 38
2.4 Resolver čili řešič 41
2.5 Život jednoho dotazu 43
2.6 Rekurzivní a nerekurzivní chování serverů 47
2.7 Domény, zóny a zónové soubory 48
2.8 Reverzní dotazy aneb hledá se jméno pro adresu 53

3 Vnitřní život DNS **59**
3.1 Doménová jména 59
3.2 Formát DNS zpráv 61
3.3 Komunikace mezi klientem a serverem 67
3.3.1 Klient 68
3.3.2 Server 70
3.4 Zónové přenosy 72
3.5 Žolíkové záznamy 76
3.6 Vyrovnávací paměť 81

4 DNS z pohledu klienta **85**
4.1 Konfigurace 86
4.1.1 Microsoft Windows 11 86
4.1.2 Microsoft Windows 10, 8 a 7 89
4.1.3 Linux a systémy odvozené z Unixu 93
4.1.4 macOS 96
4.2 Programy pro dotazování DNS 98
4.2.1 host 98
4.2.2 nslookup 102
4.2.3 dig 105
4.3 Resolver a relativní jména 110
4.4 DNS přímo z aplikace 112

5 Transportní vrstva pro DNS **115**
5.1 UDP 115
5.2 TCP 117
5.3 Stavové DNS 119
5.4 Šifrované DNS 121
5.5 DNS po TLS (DoT) 123
5.6 DNS po DTLS (DoD) 124
5.7 DNS po HTTPS (DoH) 125
5.8 DNS po QUIC (DoQ) 128

6 DNS v praxi **131**
6.1 Scénáře použití 131
6.1.1 Pouze klienti 131
6.1.2 Klienti a autoritativní server 133
6.1.3 Neveřejný autoritativní server 135
6.1.4 DNS hosting 136
6.1.5 Otevřené rekurzivní servery 137
6.2 Návrh nasazení DNS 139
6.3 Podpůrné programy 140
6.3.1 Programy doprovázející server 141
6.3.2 Zonemaster 143
6.3.3 dnswalk 145
6.3.4 Squish DNS Checker 147
6.3.5 MX Toolbox 149
6.3.6 DNSDiag 151
6.3.7 DNS Benchmark 153
6.3.8 Programy pro správu DNS dat 155

7 Správa domén a vlastně i celého Internetu **157**
7.1 Jak to celé začalo 157
7.2 Éra ICANN 158
7.3 Rozšiřování domén nejvyšší úrovně 161
7.4 Správa kořenové zóny 164
7.5 Klíče od Internetu 165
7.6 Správa domén – registr, registrátor, držitel 168
7.7 Registrace domény 171
7.8 Vlastní registrace 174
7.9 Informace o doménách a držitelích 176
7.10 Historie domény cz a sdružení CZ.NIC 182

II DNS pro starší a pokročilé **189**

8 Principy DNSSEC **191**
8.1 Digitální podpisy 192
8.2 Autentizační řetězec 194
8.3 Když řetězec nenavazuje 198
8.4 Ověřená neexistence 199
8.5 Příklad 202
8.6 Chování serverů a klientů 208
8.7 DNSSEC a žolíkové záznamy 212

9 DNSSEC prakticky **217**
9.1 Ověřování odpovědí 218
9.2 Klíče 222
9.3 Podpis vlastní zóny 235
9.4 Dokumentace 243

10 Národní znaky v doménách, čili IDN **245**
10.1 Jak funguje IDN 246
10.1.1 Přípustné znaky a jmenovky 251
10.1.2 Punycode 252
10.1.3 IDNA2003 – Nameprep, ToASCII a ToUnicode 254
10.2 Problémy a otázky 256
10.2.1 Bezpečnost 256
10.2.2 Přístupnost 258
10.2.3 Politika a strategie 259
10.3 IDN ve světě a u nás 261

11 Dynamické DNS **263**
11.1 Jak funguje 263
11.2 Chování serveru 267
11.3 Zase ta bezpečnost 269
11.4 Praktický příklad 271

12 ENUM **279**
12.1 Jak funguje 280
12.2 Nasazení 284
12.3 Infrastrukturní ENUM 284
12.4 Situace v České republice 286

13 Na pokraji DNS **287**
13.1 Skupinové DNS (mDNS) 287
13.2 Link-Local Multicast Name Resolution (LLMNR) 291
13.3 Výběrové adresování aneb anycast 294

13.4 DNS a rozkládání zátěže 296
13.5 Aktivní server aneb DNS push 299
13.6 Katalogové zóny 301

14 DNS v cizích službách 305
14.1 SPF – Sender Policy Framework 306
14.2 Sender ID 315
14.3 DKIM – DomainKeys Identified Mail 316
14.4 DNSBL a seznamy hodných, zlých a ošklivých 328
14.5 Automatická konfigurace poštovních klientů 330
14.6 DNS a objevování služeb (DNS-SD) 332
14.7 Certifikáty, klíče, PKI a další bezpečnostní harampádí 336
14.8 DNS a TLS čili DANE 339
14.9 Osobní certifikáty a klíče pro elektronickou poštu 343

15 Bezpečnost protokolu DNS 345
15.1 Bellovinův útok 345
15.2 Kaminského útok aneb otrávení vyrovnávací paměti 346
15.3 Fragmentační útoky 350
15.4 Útok náhodnými dotazy 352
15.5 Útok nekonečnou rekurzí 353
15.6 Zesilované útoky 354
15.7 Lhaní pod kontrolou aneb RPZ 359
15.8 Ochrana soukromí a minimalizace dotazů 363

III Typy záznamů 367

16 Obecně o zdrojových záznamech 369

17 Základní typy 371
17.1 A – Address (1) 371
17.2 AAAA – IPv6 Address (28) 371
17.3 CNAME – Canonical Name (5) 371
17.4 MX – Mail Exchange (15) 373
17.5 NS – Name Server (2) 374
17.6 PTR – Pointer (12) 375
17.7 SOA – Start of Authority (6) 376
17.8 TXT – Text (16) 378

18 Servisní typy 379
18.1 AXFR – Full Zone Transfer (252) 379
18.2 CSYNC – Child-to-Parent Synchronization (62) 379
18.3 DHCID – DHCP Identifier (49) 381

18.4 DNAME – Delegation Name (39) 381
18.5 IXFR – Incremental Zone Transfer (251) 383
18.6 OPT – Option (41) 384

19 Typy pro DNSSEC a bezpečnostní mechanismy 391
19.1 CAA – Certificate Authority Authorization (257) 391
19.2 CDNSKEY – Child DNS Key (60) 393
19.3 CDS – Child Delegation Signer (59) 394
19.4 CERT – Certificate (37) 395
19.5 DNSKEY – DNS Key (48) 397
19.6 DS – Delegation Signer (43) 398
19.7 IPSECKEY – IPsec Key (45) 401
19.8 KEY (25) 403
19.9 NSEC – Next Secure (47) 404
19.10 NSEC3 – Next Secure version 3 (50) 405
19.11 NSEC3PARAM – NSSEC3 Parameters (51) 408
19.12 OPENPGPKEY (61) 408
19.13 RRSIG – RR Signature (46) 409
19.14 SIG – Signature (24) 412
19.15 SMIMEA (53) 412
19.16 SSHFP – SSH Key Fingerprint (44) 413
19.17 TKEY – Transaction Key (249) 415
19.18 TLSA (52) 417
19.19 TSIG – Transaction Signature (250) 419
19.20 ZONEMD – Zone Message Digest (63) 422

20 Aplikační a ostatní typy 423
20.1 AMTRELAY (260) 423
20.2 EUI48 (108) 424
20.3 EUI64 (109) 425
20.4 HIP – Host Identity Protocol (55) 425
20.5 L32 – Locator32 (105) 425
20.6 L64 – Locator64 (106) 426
20.7 LP – Locator Pointer (107) 426
20.8 NAPTR – Naming Authority Pointer (35) 427
20.9 NID – Node Identifier (104) 431
20.10 NULL (10) 431
20.11 SRV – Service (33) 432
20.12 URI – Uniform Resource Identifier (256) 433

21 Typy odmítnuté, zastaralé a nepoužívané 435
21.1 A6 – IPv6 Address (38) 435
21.2 AFSDB – AFS Data Base (18) 435
21.3 APL – Address Prefix List (42) 436

21.4 ATMA – ATM Address (34) 436
21.5 DLV – DNSSEC Lookaside Validation (32769) 436
21.6 EID – Endpoint Identifier (31) 436
21.7 GID (102) 437
21.8 GPOS – Geographical Position (27) 437
21.9 HINFO – Host Information (13) 437
21.10 ISDN (20) 437
21.11 KX – Key Exchanger (36) 437
21.12 LOC – Location (29) 437
21.13 MAILA – Mail Agent (254) 438
21.14 MAILB – Mailbox Related (253) 438
21.15 MB – Mailbox (7) 438
21.16 MD – Mail Destination (3) 438
21.17 MF – Mail Forwarder (4) 438
21.18 MG – Mail Group (8) 438
21.19 MINFO – Mail Information (14) 438
21.20 MR – Mail Rename (9) 439
21.21 NIMLOC – Nimrod Locator (32) 439
21.22 NINFO – Zone Information (56) 439
21.23 NSAP – NSAP Address (22) 439
21.24 NSAP-PTR – NSAP Pointer (23) 439
21.25 NXT – Next Domain (30) 439
21.26 PX – X.400 Mail Mapping (26) 439
21.27 RKEY – Resource Key (57) 440
21.28 RP – Responsible Person (17) 440
21.29 RT – Route Through (21) 440
21.30 SINK – Kitchen Sink (40) 440
21.31 SPF – Sender Policy Framework (99) 440
21.32 TA – DNSSEC Trust Authorities (32768) 441
21.33 UID (101) 441
21.34 UINFO (100) 441
21.35 UNSPEC (103) 441
21.36 WKS – Well Known Service (11) 441
21.37 X25 – X.25 Address (19) 441

IV DNS servery 443

22 Obecně ke konfiguraci serveru 445
22.1 Typ serveru a obsluhovaná komunita 445
22.2 Zónové soubory 446
22.2.1 localhost 446
22.2.2 Reverzní zóny pro localhost 447
22.2.3 Kořenové servery 447

22.3 Ošetření nesmyslných reverzních dotazů 451

23 BIND 455
23.1 Základní konfigurace 455
23.1.1 Autoritativní server 456
23.1.2 Rekurzivní server 464
23.1.3 Smíšený server 467
23.2 Doprovodné programy 469
23.2.1 Kontrola konfigurace a dat 469
23.2.2 rndc – pan řídící 470
23.3 Konfigurace pro náročné 472
23.3.1 Automatické podepisování DNSSEC 472
23.3.2 Pohledy 474
23.4 Bundy, dříve BIND 10 477

24 NSD 479
24.1 Základní konfigurace 479
24.2 Doprovodné programy 484
24.2.1 nsd-checkconf 484
24.2.2 nsd-control 485
24.3 Konfigurace pro náročné 486
24.3.1 DNSSEC 486
24.3.2 Dynamické zóny 487
24.3.3 Omezování frekvence dotazů 488

25 Unbound 491
25.1 Základní konfigurace 491
25.2 Doprovodné programy 495
25.2.1 unbound-checkconf 495
25.2.2 unbound-control 496
25.2.3 unbound-host 497
25.3 Konfigurace pro náročné 499
25.3.1 Řízení vyrovnávací paměti a optimalizace 499
25.3.2 Speciální zóny a předávání dotazů 500

26 Knot DNS 505
26.1 Základní konfigurace 505
26.2 Doprovodné programy 513
26.2.1 knotc 513
26.2.2 kdig a khost 515
26.3 Pokročilé nastavení 515
26.3.1 DNSSEC 515
26.3.2 Moduly 517
26.3.3 Dynamické aktualizace 520

27 Knot Resolver **523**
27.1 Základní konfigurace 523
27.2 Spuštění a komunikace s programem 527

28 PowerDNS **529**
28.1 Základní konfigurace 530
28.2 Doprovodné programy 536
28.2.1 WWW server 536
28.2.2 pdns_control 537
28.2.3 zone2sql 538
28.3 Konfigurace pro náročné 538
28.3.1 DNSSEC 539
28.3.2 Dynamické aktualizace 543

29 PowerDNS Recursor **545**
29.1 Základní konfigurace 545
29.2 Doprovodné programy 547
29.2.1 rec_control 547
29.3 Konfigurace pro náročné 549
29.3.1 Speciální zóny 549
29.3.2 Skriptování 549

30 OpenDNSSEC **551**
30.1 Koncepce programu 551
30.2 Instalace a konfigurace 554
30.3 Provoz 570
30.4 Bezpečnostní modul 573

V Přílohy **575**

A Přehled RFC **577**
A.1 Jádro protokolu 577
A.2 Transport 577
A.3 Speciální prvky a typy záznamů 577
A.4 DNSSEC 578
A.5 IDN 578
A.6 ENUM 578
A.7 IPv6 579
A.8 Různé 579
A.9 Související technologie 579

Literatura **581**

Rejstřík **585**

Jemný úvod do DNS

1 Jemný úvod do DNS

Domain Name System (DNS) patří mezi nejstarší a nejdůležitější služby sítě Internet. Kdybyste se však běžného uživatele zeptali, zda ji používá, pravděpodobně by vůbec netušil, o čem mluvíte. DNS sice během komunikace v síti využíváme každou chvíli, jedná se však o službu poměrně neviditelnou, která má ryze servisní charakter. Vnímáme jako samozřejmost, že můžeme cíle v síti pojmenovávat ve stylu *www.nic.cz* a příliš neuvažujeme o tom, že v pozadí této přirozené vlastnosti stojí rozsáhlý mechanismus, na jehož chodu se podílejí statisíce správců z celého světa.

1.1 Historie DNS

Počítačové sítě používají pro identifikaci počítačů číselné adresy. Platí to pro současný Internet a jeho Internet Protocol verze 4, platilo to pro jeho předchůdce i všechny konkurenční síťové architektury a bude to platit i pro nastupující Internet Protocol verze 6. Pro uživatele jsou však číselné adresy nepohodlné a obtížně zapamatovatelné, stejně jako třeba telefonní čísla. Znáte třeba 142.251.37.100? Nebo 2a00:1450:4014:80f::2004? Určitě ano, ovšem nejspíš jej oslovujete jako *www.google.com*. Je zcela pochopitelné, že snahy zavést pro číselné adresy snadněji zapamatovatelná jména mají kořeny již v šerém síťovém dávnověku.

Už když se Internet ještě jmenoval ARPAnet a byla do něj zapojena jen hrstka počítačů, objevily se první snahy zavést „adresář" počítačů a používat jména místo číselných adres. Tehdejší přístup se podobal tomu, co dnes známe z kontaktů v mobilních telefonech – každý počítač měl svůj vlastní seznam známých jmen, nezávislý na ostatních. Tato schopnost ostatně zůstala zachována dodnes. V operačních systémech odvozených od Unixu najdete soubor */etc/hosts* obsahující jednoduchou převodní tabulku. Slouží jako doplněk DNS a obvykle obsahuje pouze údaje o vlastním počítači, něco jako:

```
127.0.0.1   localhost
127.0.1.1   pokus.nic.cz      pokus
# IPv6
::1         localhost ip6-localhost ip6-loopback
fe00::0     ip6-localnet
ff00::0     ip6-mcastprefix
ff02::1     ip6-allnodes
ff02::2     ip6-allrouters
ff02::3     ip6-allhosts
```

Jak vidíte, převod jména *localhost* na IPv4 adresu 127.0.0.1 či IPv6 adresu ::1 se provádí lokálně prostřednictvím tohoto souboru. V operačních systémech firmy Microsoft najdete soubor *hosts*

také. Počínaje Windows NT a Windows XP se nachází ve složce *system32\drivers\etc*, ve starších Windows 95 a 98 byl umístěn přímo v systémové složce. Cestu k němu lze nastavit v registrech.

Izolovaná lokální správa převodu jmen na adresy má zjevně sklon k nekonzistenci. Tentýž cíl se na různých místech může vyskytovat pod odlišnými jmény, ledacos bude chybět, uživatel se zkrátka na cizím počítači nemůže na nic spolehnout. Dokud bylo počítačů v síti málo, snažili se tomu její správci čelit vydáváním centrálního adresáře. Na pevně daném místě udržovali veřejně dostupnou aktuální verzi souboru *hosts*, zahrnující jména a adresy všech připojených počítačů. Správci jednotlivých systémů si jej stahovali a instalovali. Aktuálnost poskytovaných informací závisela na píli správce centrálního úložiště a správce místního stroje.

S rostoucím počtem připojených počítačů tento systém přestával stačit. Bylo potřeba vymyslet lepší řešení, nejlépe takové, které by adresy vyhledávalo on-line v okamžiku použití a bylo tak vždy aktuální. Počáteční varianty zůstávaly u centralizovaného přístupu. Nadále existoval jeden společný adresář počítačů, jen jeho kompletní distribuci nahradilo cílené dotazování na konkrétní jména. Na tomto principu je postaven *Hostname Server* definovaný v RFC 811 a později RFC 953. Poskytuje aktuální informace odpovídající stavu centrální databáze, ovšem s rostoucím počtem připojených strojů rychle narazí na meze své škálovatelnosti. Představte si, že by se tímto způsobem někdo pokusil pojmenovat současný Internet s několika miliardami připojených zařízení.

DNS vzniklo jako přímý konkurent Hostname Serveru a řešilo tento problém. Klíčové požadavky a předpoklady, jež stály u jeho kolébky, byly:

- vytvořit konzistentní systém jmen identifikujících jednotlivé zdroje,
- poskytovat aktuální informace,
- umožnit distribuovanou správu dat, aby informace zadával pokud možno místní správce,
- lze očekávat, že většina dat se bude měnit jen velmi pomalu (ale systém musí být schopen zvládat rychle se měnící údaje),
- systém by měl být efektivní a nezatěžovat příliš komunikační infrastrukturu,
- dostupnost údajů je důležitější než stoprocentní konzistence.

Na základě těchto východisek vznikl silně distribuovaný a redundantní Domain Name System. Skládá se ze tří základních částí:

- *Data*, tedy stromově uspořádaný jmenný prostor a v něm se nacházející záznamy s informacemi o vlastnostech jednotlivých prvků.
- *Servery*, které mají uloženy části datového stromu – informace o jednotlivých doménách a vztazích mezi nimi – a odpovídají na dotazy.
- *Klienti (resolvery)* představují rozhraní systému vůči uživatelským aplikacím. Pokládají dotazy serverům a zpracovávají jejich odpovědi.

Požadavek na efektivitu vedl k zavedení vyrovnávacích pamětí (cache). Ty se mohou vyskytovat jak u klientů, tak u serverů a umožňují opětovně využívat nedávno získané informace, které však již nemusí odpovídat aktuálnímu stavu. Odpověď proto obsahuje příznak, zda pochází od autoritativního zdroje či z vyrovnávací paměti.

Vysokou dostupnost řeší DNS redundancí – každá doména musí být poskytována alespoň dvěma nezávislými servery, což opět může ohrozit konzistenci dat (na jednom již byla provedena změna, na druhém zatím ne). Systém se k problému staví pragmaticky. Bere riziko drobných a časově omezených nekonzistencí, jako jsou zastaralé informace ve vyrovnávací paměti či odlišnost verzí poskytovaných pro tutéž doménu různými servery, za přijatelnou cenu, již platíme za dostupnost a efektivitu.

Autorem DNS je Paul Mockapetris, který jeho základní specifikaci publikoval koncem roku 1983 v RFC 882 *Domain Names – Concepts and Facilities* a RFC 883 *Domain Names – Implementation and Specification.* Vytvořil také první implementaci navrženého mechanismu a brzy po ní následovaly další. V roce 1984 se objevila první implementace pro operační systém Unix, kterou o rok později významně změnil a přejmenoval Kevin Dunlap. Pro svou verzi zvolil jméno *BIND – Berkeley Internet Name Domain* a výsledný program se stal legendou. Zcela ovládl pole, prosadil se snad do všech operačních systémů odvozených od Unixu jako jejich standardní součást, dočkal se portace i do jiných systémů, stal se de facto standardem a dodnes jej používá drtivá většina DNS serverů. Od roku 1988 jej udržoval a rozvíjel Paul Vixie, s jehož jménem je BIND nejčastěji spojován.

Praktické zkušenosti s provozem DNS se promítly do druhé generace jeho dokumentů. Koncem roku 1987 vyšla dvojice RFC 1034 a RFC 1035, která nahradila původní specifikace RFC 882 a RFC 883. Tyto dva základní dokumenty zůstávají v platnosti dodnes. Protokolům, jež tvoří jádro DNS, je tedy již přes 30 let. Existuje k nim pochopitelně řada pozdějších doplňků a rozšíření, z nichž mimořádný význam mají zejména *Internationalized Domain Names (IDN)* umožňující používat ve jménech znaky národních abeced a *Domain Name System Security Extensions (DNSSEC)*, díky němuž lze ověřit pravost obsahu DNS a chránit se proti podvrhům. Neustále se také rozšiřuje sortiment informací, které DNS poskytuje.

1.2 K čemu slouží

Základním úkolem DNS a příčinou jeho vzniku byl převod symbolických jmen na IP adresy, aby si je uživatelé nemuseli pamatovat. To ovšem není zdaleka jediná služba, kterou nabízí. Ostatně, když už se podařilo vyvinout dobře fungující distribuovanou databázi, byla by škoda ji nevyužít i pro další účely. Jako první se samozřejmě nabízí převod opačným směrem aneb zjištění, pod jakým jménem je v DNS zaregistrována určitá IP adresa. Tato informace se hodí pro protokoly

o činnosti různých serverů, abychom mohli třeba analyzovat, odkud přichází nejvíce klientů dané služby, nebo pro zobrazení výsledků příkazu *traceroute*. Výpis ve tvaru:

```
 1  router-h (147.230.16.3)
 2  147.230.250.49 (147.230.250.49)
 3  195.113.235.101 (195.113.235.101)
 4  cesnet.rt1.pra.cz.geant2.net (62.40.124.29)
 5  so-6-3-0.rt1.fra.de.geant2.net (62.40.112.38)
 6  abilene-wash-gw.rt1.fra.de.geant2.net (62.40.125.18)
 7  so-0-0-0.0.rtr.atla.net.internet2.edu (64.57.28.6)
 8  so-3-2-0.0.rtr.hous.net.internet2.edu (64.57.28.43)
 9  so-3-0-0.0.rtr.losa.net.internet2.edu (64.57.28.44)
10  hpr-lax-hpr--i2-newnet.cenic.net (137.164.26.132)
11  ucla--lax-hpr1-ge.cenic.net (137.164.27.6)
12  border-1--core-1-10ge.backbone.ucla.net (169.232.4.100)
13  core-1--anderson-2-ge.backbone.ucla.net (169.232.8.31)
```

dává slušnou představu, kterými sítěmi a lokalitami pakety procházely. Rozhodně výrazně lepší než samotné adresy. Ostatně, poznali byste bez DNS, že 64.57.28.6 leží v Atlantě, 64.57.28.43 v Houstonu a 64.57.28.44 v Los Angeles?

DNS využívají všechny síťové služby. Některé jen k tomu, aby mohly používat jména zúčastněných strojů a nikoli jejich adresy, jiné si z DNS berou klíčové informace pro svou činnost. Nejvýznamnějším představitelem této skupiny je nepochybně elektronická pošta. DNS totiž obsahuje údaje o tom, které servery přijímají poštu pro danou doménu. Řekněme, že pošlete dopis na adresu *kontakt@nic.cz*. Váš poštovní server ji podle zavináče rozdělí na uživatelské jméno a doménu a následně kontaktuje DNS, aby zjistil, které poštovní servery zajišťují příjem pošty pro doménu *nic.cz*. Dozví se, že nejvhodnější je *mail.nic.cz* a případně může použít i *mx.nic.cz*. DNS mu dodá jejich jména, adresy i priority. V uvedeném pořadí se je pokusí kontaktovat a předat jim váš dopis.

Mimochodem, právě díky tomu je elektronická pošta velmi rychlá a efektivní. Stejnou informaci o přijímajících serverech dostane tazatel z celého Internetu – ať se server odesílající dopis nachází v Praze, Paříži či v Buenos Aires, pokaždé se dozví, že má nejprve zkusit kontaktovat *mail.nic.cz* a doručí díky tomu dopis rovnou skoro až domů. Cesta elektronického dopisu proto obvykle zahrnuje jen několik málo skoků.

Tím však spolupráce elektronické pošty a DNS nemusí zdaleka končit. V rámci boje proti všudypřítomnému spamu se objevila některá řešení, která využívají DNS k posouzení míry důvěryhodnosti přicházejícího dopisu. Do této skupiny patří například Sender Policy Framework (SPF)

či DomainKeys Identified Mail (DKIM). DNS totiž kromě jednoúčelových záznamů s pevně definovaným obsahem, jako jsou například IP adresy, nabízí i obecnější prostředky. Značné popularitě se těší zejména obecný textový záznam, díky němuž lze do DNS uložit ledacos. Mechanismy jako SPF či DKIM obvykle stanoví určité pevně dané jméno záznamu a strukturu textové informace v něm obsažené. Příslušný záznam si pak vyzvednou a podle jeho obsahu dopis posoudí.

Příklady táhnou a DNS jako fungující distribuovaná databáze monumentálních rozměrů nachází další a další uplatnění. Po bok elektronické pošty se časem zařadily i další aplikace, které pilně pracují s DNS a hledají v něm informace pro svou potřebu. Patří mezi ně i IP telefonie, která si pod označením ENUM zavedla svůj vlastní typ záznamů i organizaci části doménového stromu. Využívá jej k překladu mezi telefonními čísly používanými v klasickém telefonním světě a IP adresami, protokoly a porty, s nimiž pracuje telefonie v počítačové síti.

Podstatnou nevýhodou DNS je, že své informace poskytuje bez záruk. Pro technicky zdatného a motivovaného útočníka není velkým problémem je podvrhnout a docílit toho, že po zadání jména přistanete na úplně jiném serveru, než jste zamýšleli. Proto vznikl bezpečnostní mechanismus DNSSEC, který tento problém pomocí digitálních podpisů odstraňuje a činí DNS atraktivním pro další aplikace. Mimo jiné se v něm objeví důvěryhodné kryptografické klíče, které lze využít pro šifrování či ověřování podpisů.

Bohužel se DNSSEC do praxe prosazuje jen ztěžka a pomalu, což omezuje některé další možnosti využití DNS. My jsme na tom dobře. Doména *cz* byla podepsána mezi prvními a patří dlouhodobě v této oblasti ke světové špičce. V polovině roku 2023 byla podepsána více než polovina jejích poddomén, jinde je ale situace o poznání horší. Pokud se DNSSEC dostatečně rozšíří, DNS se docela dobře může stát kýženou infrastrukturou veřejných klíčů (PKI), kterou si bezpečnostní inženýři vysnili už před dlouhými lety, která je však dosud reálná asi jako jednorožec.

Část I

Základní principy

První část knihy popisuje základní konstrukce DNS světa a principy, na nichž je postaven. Seznámíte se s uspořádáním domén, úlohami jednotlivých programů, vzájemnými vztahy mezi nimi a postupu při hledání odpovědi na dotaz.

Další kapitola zevrubně popisuje data – formáty zpráv, pravidla pro komunikaci mezi klienty a servery, či využití vyrovnávacích pamětí. Klientům v nejběžnějších operačních systémech se pak věnuje samostatná kapitola.

Jelikož se nám postupem času zkomplikovala otázka vzájemné komunikace mezi klientem a serverem, vyhradili jsme celou kapitolu popisu transportní vrstvy a jejích různých protokolů využívaných pro DNS.

Závěr první části se zabývá praktickými tématy. Dozvíte se, jak vypadá správa jednotlivých domén, které organizace se na ní podílejí a v jakých úlohách i jak prakticky postupovat, pokud si hodláte pořídit vlastní doménu. Závěrečná kapitola poskytuje praktická doporučení pro návrh a provoz DNS a doporučení užitečných programů, které vám tuto úlohu usnadní.

2 Jak to funguje

V této kapitole se podíváme trochu podrobněji, co se skrývá pod kapotou DNS – jak vypadá doménový strom, jaké informace obsahuje, jakou roli mají servery při vyřizování dotazů a jak všechny součásti vzájemně spolupracují. Stará programátorská poučka říká, že návrh by měl vždy začínat datovými strukturami. Proto se nejprve podívejme na data v DNS – doménový strom a zdrojové záznamy. Pak budeme pokračovat zbývajícími částmi systému – servery a resolvery.

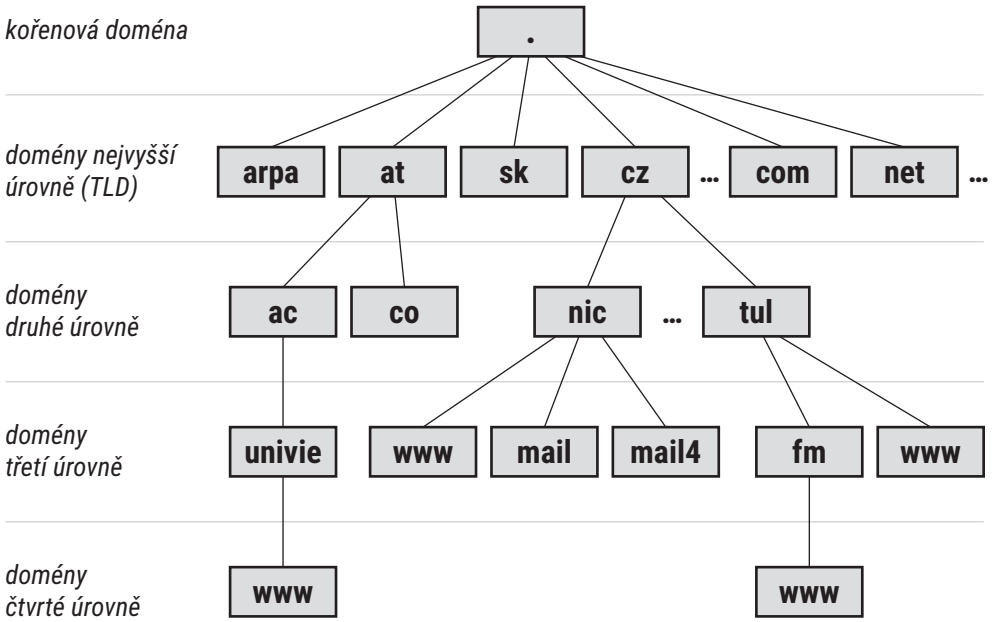
2.1 Doménový strom

Z hlediska obsahu je klíčovou komponentou DNS doménový strom. V této datové struktuře jsou hierarchicky uspořádány veškeré informace, které obsahuje. Všichni víte, že doménová jména se zapisují v podobě *www.fm.tul.cz* – jednotlivé domény jsou v zápisu oddělovány tečkami a uvádějí se v pořadí od konkrétních k těm nejobecnějším. Jméno však zároveň funguje – směrem zezadu dopředu – jako popis průchodu doménovým stromem. V tomto konkrétním případě říká, že z kořene doménové stromu je třeba jít do domény *cz* (označující Českou republiku), z ní pokračovat do její poddomény *tul* (Technická univerzita v Liberci), dále pokračovat do poddomény *fm* (Fakulta mechatroniky) a v ní navštívit poddoménu *www*. Doménové jméno tedy jednoznačně identifikuje konkrétní uzel v doménovém stromě a zároveň popisuje cestu od kořene k němu. Zejména tato druhá vlastnost je klíčová, protože je využívána při hledání odpovědi.

Zastavme se ještě chvíli u vlastního doménového stromu. Jako každý správný strom obsahuje jediný kořen – tak zvanou *kořenovou doménu (root domain)*. Jelikož je jediná, nemá smysl ji ve jméně uvádět, a proto ji v zápisu nenajdete. V případě nutnosti se zapisuje jako samotná tečka (tímto způsobem je také znázorněna na našich obrázcích).

Některé nástroje či konfigurační soubory rozlišují mezi absolutně a relativně zadaným doménovým jménem. *Absolutně zadané jméno* končí tečkou – například *www.nic.cz.* – a znamená to, že je úplné, protože obsahuje všechny údaje až po kořenovou doménu. Používá se pro ně také pojem plné či plně specifikované doménové jméno, v originále *Fully Qualified Domain Name (FQDN)*.

Pokud na konci není tečka, chápe se jako *jméno relativní* a příslušný program si k němu doplní, či alespoň může doplnit, nějakou koncovku. Zkuste si třeba do WWW prohlížeče zadat jako cílovou adresu jen samotné *www* a uvidíte, že si k němu iniciativně něco doplní a přistanete na konkrétním webu. Relativní jména jsou běžná v datových souborech DNS, kde se vždy vztahují k aktuální doméně. Jinak bychom se upsali. Problematika je složitější, protože nelze chtít po uživatelích, aby jména důsledně ukončovali tečkou. Operační systémy a aplikace proto odhadují,



Obrázek 2.1: Doménový strom

zda jméno bez tečky na konci skutečně je či není relativní. Jejich chování se budeme věnovat podrobněji v části 4.3 na straně 110.

Hlavní úlohou kořenové domény je držet celý strom pohromadě a poskytnout výchozí bod pro řešení dotazů. Bezprostředně pod ní se nacházejí tak zvané *domény nejvyšší úrovně (top-level domains, TLD)*. Původně vznikly s cílem stručně charakterizovat držitele domény a zároveň snížit pravděpodobnost konfliktů. Když DNS přišlo na svět, mělo na nejvyšší úrovni pět domén: *com* pro komerční firmy, *edu* pro vzdělávací instituce, *gov* pro vládu, *mil* pro vojáky a *org* pro organizace nezařaditelné do žádné z ostatních kategorií. Záhy k nim přibyly domény jednotlivých států a postupem času i další. Podle určení a klíčových vlastností je lze rozdělit do čtyř základních kategorií (Carl von Linné by měl radost):

- *Obecné domény (generic top-level domains, gTLD)* navazují na původní členění. Dělí se do dvou odrůd: Nesponzorované obecné domény, jako například *com*, *net*, *org* nebo *info*, patří do přímé kompetence ICANN. Sponzorované obecné domény (sTLD) jsou založeny a spravovány určitou organizací, která určuje podmínky pro registraci v nich. Bývají vymezeny geograficky nebo tematicky, například v sTLD *aero* mohou získat poddoménu jen subjekty působící v oblasti letectví. Jako další příklady sponzorovaných obecných domén lze jmenovat *asia*, *jobs*, *museum* či *xxx*.

- *Státní domény (country-code top-level domains, ccTLD)* jsou domény přidělené jednotlivým státům. Jejich zkratky odpovídají (až na několik výjimek) dvoupísmenným zkratkám příslušných států podle standardu ISO 3166-1. Jistě nejvýznamnější výjimkou je doména *eu* pro Evropskou unii. Za zmínku stojí též britská doména *uk*, přestože Velká Británie má ve zmiňovaném standardu přidělenu zkratku *gb*. Důvody výjimek jsou zpravidla historické.
- *Státní domény v národních abecedách (internationalized ccTLD, IDN ccTLD)* odpovídají, podobně jako předchozí kategorie, jednotlivým státům. Jejich názvy jsou ovšem zapsány v národních abecedách, které nevycházejí z latinky (čínská, arabská a další písma). Problematiku ne-ASCII jmen probereme podrobněji v kapitole 10 na straně 245.
- *Infrastrukturní doména (infrastructure top-level domain)* je pouze jediná. Jedná se o doménu *arpa*, která slouží pro některé interní mechanismy DNS, zejména pro reverzní převody IP adres na jména a pro ENUM. Má poměrně zajímavou historii – původně byla určena pro agenturu ARPA (Advanced Research Projects Agency), jež stála u kolébky Internetu. Později doména změnila svůj význam a odpovídajícím způsobem byl upraven i význam zkratky, v současnosti znamená Address and Routing Parameter Area.

Po bouřlivém začátku, kdy kolem poloviny 80. let vznikla valná většina domén nejvyšší úrovně, nastalo dvacetileté období klidu. Kořenová doména byla tou dobou velmi konzervativní, obsahovala kolem 300 poddomén a narůstala průměrným tempem jedné až dvou nových TLD ročně. Roku 2005 pak byl zahájen proces vedoucí ke vzniku celé řady nových obecných domén. Podrobněji se k němu vrátíme v části 7.3 na straně 161.

Kromě výše uvedených čtyř existuje ještě jedna skupina doménových jmen nejvyšší úrovně. Mezi předchozí ovšem nepatří, protože tyto domény neexistují a ani existovat nemohou. RFC 2606 rezervovalo několik jmen nejvyšší úrovně pro speciální účely:

- *test* pro testování DNS programů,
- *example* pro příklady v dokumentaci,
- *invalid* pro vytváření neplatných jmen a
- *localhost* pro odkaz sama na sebe.

Kromě nich byly vyhrazeny ještě tři domény druhé úrovně (*example.com*, *example.net* a *example.org*), opět pro potřeby příkladů v dokumentaci. Tyto domény nejsou součástí doménového stromu a mohou být volně využívány k daným účelům.

Jelikož se později objevilo ještě několik jmen rezervovaných pro speciální účely, RFC 6761 *Special-Use Domain Names* je shrnulo a zavedlo pro ně registr, který obhospodařuje IANA. Aktuální přehled doménových jmen se speciálním významem najdete na adrese:

🔗 *http://www.iana.org/assignments/special-use-domain-names/special-use-domain-names.xhtml*

Poddoménami TLD jsou tak zvané *domény druhé úrovně*. K jejich přidělování existují dva základní přístupy: Většinou jsou přidělovány již konkrétním organizacím či jednotlivcům. Tímto způsobem je spravována například naše doména *cz*, proto třeba Technická univerzita v Liberci disponuje doménou *tul.cz*, Seznam doménou *seznam.cz* a podobně.

V některých doménách se snaží na druhé úrovni rozlišit charakter vlastníka, podobně jako původní nejvyšší domény *com*, *edu* a spol. Domény organizací a dalších vlastníků se pak nacházejí až na třetí úrovni. Nejznámějšími příklady tohoto uspořádání jsou Rakousko (*at*), Velká Británie (*uk*) či Austrálie (*au*), ale je jich mnohem víc. Seznam dostupných koncovek doménových jmen obsahuje *Public Suffix List*, který najdete na adrese:

🔗 *https://publicsuffix.org/list/*

Například vídeňská univerzita je držitelem domény *univie.ac.at*, kde *ac* na druhé úrovni signalizuje akademickou instituci. Tento způsob organizace domén je výrazně menšinový a spíše se od něj ustupuje – třeba v Rakousku je dnes již možná také registrace domén druhé úrovně. Podrobnější informace o správě domén se dočtete v kapitole 7 na straně 157.

Hloubka doménového stromu je omezena na 127 úrovní. Tento limit je však ryze teoretický, v praxi se jen zřídka setkáte s hloubkou větší než pět. Výjimkou jsou různé speciální mechanismy – ENUM pro telefonní čísla a zejména reverzní dotazy pro IPv6, které zasahují až do 35. úrovně. To je hodně extrémní případ, přesto se dostal jen do čtvrtiny maximální přípustné hloubky.

Když už jsme u extrémů, zmiňme se o existenci alternativních DNS stromů. Naše kniha je věnována „originálnímu a jedině pravému DNS“. Kromě něj ovšem existují i jeho vedlejší odnože, které používají stejné protokoly, technologie i programy, jen jejich doménový strom je odlišný. Má svůj vlastní kořen, ze kterého obvykle vedou odkazy na všechny TLD oficiálního DNS, ovšem kromě nich ještě některé navíc. Důvody pro jejich existenci mohou být ideové (nesouhlas s některými pravidly a omezeními, jimž podléhá standardní kořenová doména), komerční, nebo se může jednat o čistě interní záležitost určité organizace.

Jejich společnou vlastností je, že vyžadují speciální konfiguraci na straně strojů, které DNS prohledávají, a jejich dosah bývá velmi omezený. Drtivá většina Internetu o nich nemá tušení. Hlavním problémem jejich existence je samozřejmě nekonzistence informací. Kromě toho vznikají nepříjemné konflikty, pokud některý z alternativních kořenů vytvoří svou specifickou TLD a později je stejnojmenná doména založena v oficiálním DNS, což se přihodilo například doméně *biz*. Internetová komunita nepohlíží na alternativní doménové stromy s žádným nadšením, což se odráží i v RFC 2826 *IAB Technical Comment on the Unique DNS Root*. Jeho obsah by se dal stručně shrnout heslem „jeden Internet – jedno DNS“. I nám se existence alternativních doménových stromů jeví jako krajně problematická a nebudeme se jim více věnovat.

2.2 DNS záznamy

Doménový strom obsahuje pro jednotlivá jména pestrou směs různorodých údajů. V předchozí kapitole jsme uvedli několik příkladů informací, které lze do DNS ukládat. Ve skutečnosti je jich ale ještě mnohem více. Vše se ukládá do tak zvaných *zdrojových záznamů (resource records, RR)*, které představují základní jednotku informace, s níž DNS pracuje. Zdrojový záznam obsahuje pět položek:

- *Jméno (name)*, ke kterému se vztahuje.
- *Životnost (time to live, TTL)* udává počet sekund, po které lze záznam uložit ve vyrovnávací paměti.
- *Třída (class)* určuje rodinu protokolů, pro něž je záznam určen. DNS lze teoreticky používat i pro jiné síťové architektury. V praxi se to samozřejmě neděje, takže třída vždy obsahuje hodnotu 1 reprezentující Internet (je zapisována jako IN).
- *Typ (type)* určuje, jakého typu jsou informace nesené tímto záznamem. Přehled několika málo nejčastěji používaných typů najdete v tabulce 2.1. Všechny dostupné typy, včetně těch zavedených pozdějšími rozšířeními, shrnuje část III na straně 369.
- *Data (data)* nesou vlastní datový obsah záznamu. Jejich struktura a interpretace závisí na typu – například záznam typu A obsahuje ve svých datech IPv4 adresu, zatímco záznam typu MX nese prioritu a doménové jméno serveru pro příjem elektronické pošty.

Typ	*Význam*
A	IPv4 adresa
AAAA	IPv6 adresa
PTR	jméno k příslušné IP adrese
CNAME	přezdívka pro existující jméno
MX	poštovní server pro danou doménu
NS	DNS server pro danou doménu
SOA	základní informace o doméně

Tabulka 2.1: Nejběžnější typy zdrojových záznamů

Pro jedno jméno doménový strom často obsahuje několik záznamů. Jedná se o naprosto libovolnou směs, která může obsahovat kombinaci několika různých typů i opakovaných výskytů stejného typu. Například pro doménové jméno *nic.cz* existují následující záznamy:

- Začíná zde působnost nového správce (NIC.CZ), obsahuje proto jeden záznam typu SOA se základními informacemi.
- Obhospodařují ji tři autoritativní servery, obsahuje proto tři záznamy typu NS.

- Příjem pošty pro ni zajišťují dva různé poštovní servery, jimž odpovídají dva záznamy MX.
- Má přiřazenu i IP adresu (po jednom záznamu typů A a AAAA), aby se návštěvníci *www.nic.cz* nemuseli namáhat s úvodním *www*.
- Obsahuje dva záznamy typu TXT.
- Doména je chráněna DNSSEC, což zahrnuje 12 záznamů tří různých typů (RRSIG, DNSKEY a NSEC3PARAM).

Jak vidíte, pro doménové jméno *nic.cz* najdeme ve stromě celkem devět typů záznamů, z nichž některé se vyskytují opakovaně. Dohromady napočítáme 21 různých záznamů. Naproti tomu celá řada koncových počítačů má v současnosti jediný záznam typu A, který oznamuje jejich IPv4 adresu. Obecně platí, že domény ve vyšších patrech hierarchie obvykle mívají větší počet záznamů než běžné počítače nacházející se v listech doménového stromu.

Různorodost uložených informací se nutně promítá do složení DNS dotazu. V něm je nutno uvést, co konkrétně tazatele zajímá – tedy jaký typ záznamu a pro jaké doménové jméno požaduje[1]. Iniciátorem DNS transakce bývá aplikace a od ní také pochází rozhodnutí, jaké typy záznamů a pro které jméno má DNS najít. Jestliže ve svém prohlížeči zatoužíte po *www.nic.cz*, vyvoláte tak dotaz na záznam typu A nebo AAAA pro jméno *www.nic.cz*. Když pošlete elektronický dopis na adresu *info@nic.cz*, váš poštovní server bude poptávat záznamy MX pro jméno *nic.cz*.

Analogicky jsou i odpovědi tvořeny sadami zdrojových záznamů. Obsahují poptávanou informaci (pokud existuje) – konkrétně všechny záznamy požadovaného typu pro dané doménové jméno. Odesílající server často k odpovědi přibalí i další informace, které by tazatel nejspíš vzápětí poptával. Zvyšuje se tím efektivita celého systému.

Vyjděme z posledního příkladu, kdy tazatel poptával záznamy typu MX pro doménu *nic.cz*. Existují celkem dva: poštu přijímají *mail.nic.cz* a *mx.nic.cz*. To jsou však stále jména, pro vlastní odeslání dopisu bude třeba je převést na adresy. Jelikož odesílající server zná situaci v doméně *nic.cz*, rovnou ke své odpovědi přibalí i doplňkové informace o odpovídajících IP adresách. Konkrétně pošle záznamy typu A a AAAA pro oba uvedené servery.

2.3 Domény a servery

Klíčovou úlohu v celém systému hrají servery. Obsahují jednotlivé části oné obrovské distribuované databáze, zasílají si dotazy a odpovědi a spolupracují při distribuci informací. Díky jejich vzájemné provázanosti a spolupráci celý systém vůbec může fungovat. DNS servery mají ke

1: V dotazu se uvádí také požadovaná třída záznamu, což je ovšem ryzí formalita. Veškeré současné DNS se točí kolem třídy IN. Proto budeme v textu až na výjimky třídu ignorovat.

konkrétním doménám různý vztah, ze kterého pak vyplývá jejich úloha při řešení dotazů. Pro určitou konkrétní doménu může mít server jednu z následujících rolí:

- *Autoritativní server* obsahuje data dané domény a zodpovídá dotazy na ně. Obsah autoritativních serverů tvoří onu distribuovanou databázi, kterou DNS představuje. O odpovědích autoritativního serveru ohledně dané domény se nepochybuje, jsou považovány za správné, protože pocházejí přímo od zdroje. Na základě vzájemných vztahů lze rozlišit dvě varianty autoritativních serverů: primární a sekundární.
- *Primární server (master)* je místo, kde vznikají informace o doméně. Pokud je třeba udělat v jejích datech změny, musí být provedeny na primárním serveru. Z podstaty věci je jasné, že každá doména má obvykle právě jeden primární server.
- *Sekundární server (slave)* je automatickou kopií primárního. V pravidelných intervalech se na něj obrací s dotazem, zda v doméně nedošlo ke změně, a pokud ano, stáhne si aktuální verzi dat. Současné implementace zpravidla umožňují, aby primární server aktivně upozornil sekundární, že se cosi změnilo a je čas přihlásit se o aktuální data. Díky tomu je synchronizace údajů podstatně rychlejší.
  Úloha sekundárního serveru je jasná – dokáže zastoupit primární v případě výpadku a slouží i pro rozkládání zátěže. Je také autoritativní, jeho odpovědi mají v DNS stejnou váhu jako odpovědi primárního serveru. Každá doména musí mít dva autoritativní servery, typicky jeden primární a jeden sekundární, který by měl být z hlediska síťové topologie co nejnezávislejší na primárním.
- *Rekurzivní server* má dvojí tvář. Vůči svým klientům, což bývají typicky stroje z místní sítě, vystupuje jako server. Přijímá jejich dotazy, řeší je a posílá odpovědi. Během řešení se dotazuje autoritativních serverů, vůči kterým vystupuje jako klient. Informace, které při řešení získá, si ukládá do vyrovnávací paměti a dokud nevyprší jejich životnost, využívá je pro své další odpovědi. Hlavním smyslem rekurzivních serverů je poskytnout společnou vyrovnávací paměť pro místní klienty. Aby se například adresy často používaných webů nehledaly znovu a znovu. Zopakuje-li se dotaz na jméno, které má rekurzivní server v paměti, rovnou pošle odpověď. Zvyšuje tak efektivitu celého systému, zmenšuje počet dotazů a odpovědí přenášených po síti a zrychluje odezvy. Jeho odpovědi samozřejmě nejsou autoritativní. Proto se pro něj používá také pojem *pomocný server (caching only)*.

Ještě jednou zdůrazněme, že výše zmiňované role serverů se vždy vztahují ke konkrétní doméně. Jeden a tentýž server může být primární pro tři domény, sekundární pro dalších pět a pomocný pro všechny ostatní. Rozhoduje o tom pouze a jedině jeho konfigurace.

Z pohledu DNS komunikace není žádný rozdíl mezi primárním a sekundárním serverem. Dokonce ani není viditelný. Pro doménu se ukládají (v záznamech typu NS) všechny autoritativní servery, primární a sekundární se nijak nerozlišují. Odpověď v sobě nese příznak (označovaný *AA* – Authoritative Answer), zda je autoritativní nebo nikoli. Jeho hodnota závisí na tom, zda

odpověď odeslal přímo některý z autoritativních serverů, nebo zda ji odeslal server pomocný ze své vyrovnávací paměti.

Rozdělení na primární a sekundární servery je čistě organizační záležitost, která se promítá do jejich konfigurace. Týká se toho, jak data vznikají, nikoli jejich věrohodnosti. Z pohledu DNS jsou odpovědi všech autoritativních serverů pro doménu stejně důvěryhodné. Některé domény koncept primárních a sekundárních serverů vůbec nepoužívají. Data mají například uložena v databázi, ze které si je berou jejich autoritativní servery. Synchronizaci jejich obsahu zajišťují databázové prostředky.

Mimochodem, jedny z nejošklivějších chyb vznikají, pokud se rozejde obsah primárního serveru se sekundárními. K této situaci může dojít, pokud správce domény upravuje její obsah ručně a zapomene zvětšit sériové číslo, které slouží jako indikátor změny. Sekundární servery si pak nechají původní obsah domény a jejich odpovědi se budou lišit od primárního. Jednotliví klienti se budou obracet na různé autoritativní servery (v závislosti na softwarové implementaci klienta a dobách odezvy jednotlivých serverů) a budou dostávat nekonzistentní odpovědi. Rozložení příjemců špatných odpovědí bude v podstatě náhodné, což řešení problému rozhodně neulehčí.

Případné problematické stavy ještě zhoršují pomocné servery. Na jedné straně výrazně zkracují doby odezvy DNS a snižují zátěž autoritativních serverů, na straně druhé však do něj zavádějí určitou setrvačnost. Dojde-li ke změně, vyrovnávací paměti pomocných serverů nadále obsahují původní údaje a dokud nevyprší jejich platnost, budou je poskytovat svým tazatelům. Tyto servery jsou rozesety po celém světě, jsou zcela mimo působnost správce domény a dotyčné záznamy si uložily v různém čase. Postupně budou nahrazovány aktuální verzí, ale tento proces může trvat několik dnů. Správce domény jej může ovlivňovat jen nepřímo – vhodným nastavením životnosti záznamů. K problematice se vrátíme později. Zde pouze konstatujme, že DNS neposkytuje za každých okolností stoprocentně konzistentní a aktuální informace. Je to cena, kterou platíme za jeho škálovatelnost a efektivitu.

Obecně platí, že čím významnější je doména, tím větší péče bývá věnována jejím autoritativním serverům. Má jich zpravidla větší počet a jsou vhodně rozmístěny v různých částech Internetu. Počet autoritativních serverů poslouží jako orientační měřítko významu domény. Posuďte sami: doména *tul.cz* má dva, *nic.cz* tři, doména *cz* čtyři, *com* hned třináct a stejný počet má i kořenová doména. Ve skutečnosti je jich podstatně více, protože za jedním jménem v záznamu typu NS se může skrývat mnoho fyzických serverů. Více se dočtete v části 13.3 na straně 294.

Vyšší počet autoritativních serverů má dva významy: větší robustnost, protože lépe dokáže překonat i větší výpadky sítě, a také rozkládání zátěže. Na autoritativní servery velkých domén se valí obrovské množství dotazů, jejich obsluhování více stroji rozloženými v různých částech sítě je holou nezbytností.

Zcela zásadní roli mají autoritativní servery kořenové domény, tak zvané *kořenové servery (root servers)*. U nich začíná řešení všech dotazů, které pak postupuje doménovým stromem po jednotlivých patrech od kořene až k cíli. Hnedle se mu budeme věnovat. Případná nedostupnost všech kořenových serverů by učinila DNS nefunkčním. Proto se za jejich třinácti adresami ve skutečnosti skrývají stovky serverů[2]. Používají se tak zvané výběrové adresy (anycast), kdy se stejná adresa přidělí několika strojům a směrování se postará o doručení paketu nejbližšímu z jejích majitelů. Více se o tomto způsobu adresování dočtete v části 13.3 na straně 294. Podrobnosti o kořenových serverech včetně mapky jejich rozložení najdete na adrese:

www.root-servers.org

Budiž autorům a správcům DNS připsáno ke cti, že za celou jeho historii nedošlo k vyřazení všech kořenových serverů, přestože se vyskytlo několik cílených útoků na ně.

Zmíněných třináct adres kořenových serverů je *velmi* konzervativních, protože je musí znát miliony dalších serverů v celém Internetu. Mění se proto zhruba rychlostí pohybu kontinentů. Jejich nejvýznamnější změnou v posledních letech byla postupná implementace IPv6 a s ní přidávané IPv6 adresy.

2.4 Resolver čili řešič

Máme tedy data, máme i servery a zbývá nám už jen klient, který se bude systému ptát. V oficiální terminologii se pro něj používá termín *resolver* (čili řešič). Jeho hlavní úlohou je zprostředkovávat místním aplikacím styk s DNS – na základě jejich požadavků sestavuje DNS dotazy, zasílá je serverům a zpracovává příchozí odpovědi. Obvykle také mívá vlastní vyrovnávací paměť, aby zbytečně neobtěžoval stále stejnými dotazy.

Resolver bývá součástí operačního systému. Z programátorského hlediska má zpravidla podobu knihovny funkcí, které tvoří jeho rozhraní (API) vůči aplikacím. Touto cestou mu jednotlivé programy zadávají úkoly a přebírají si výsledky. Uživatel se na něj přímo obrátit nemůže, proto bývají součástí systému jednoduché aplikace, jejichž prostřednictvím lze komunikovat s resolverem poměrně přímo a zjišťovat tak aktuální informace v DNS. Podrobněji se jim budeme věnovat v kapitole 4 na straně 85.

Podle schopností existují dva základní druhy resolverů:

2: Konkrétně v polovině roku 2023 jich bylo zhruba 1500.

- *Plnohodnotný (též rekurzivní) resolver* je schopen samostatné existence. Zná adresy kořenových serverů a pokud nemá ve vyrovnávací paměti nic, co by pomohlo s vyřešením aktuálního dotazu, obrátí se na některý z nich. Podrobněji celý proces popíšeme vzápětí. Tento typ resolverů bývá základem výše zmíněných rekurzivních DNS serverů.
- *Jednoduchý resolver* (v originále *pahýlový*, *stub resolver*) představuje minimalistický přístup. Potřebuje ke své činnosti adresu místního serveru (případně několika serverů), na který se má obracet se svými dotazy. Dostane-li od aplikace požadavek, převede jej do podoby DNS dotazu a pošle danému serveru. Ten dotaz vyřeší a pošle zpět výslednou odpověď. Jednoduchý resolver nezná adresy kořenových serverů a není schopen samostatné existence. Sám dotazy neřeší, jen je předává místnímu serveru. Součástí běžných operačních systémů bývají právě takové resolvery.

Jednoduchý resolver se snadněji implementuje. Sortiment jeho funkcí je omezený, a kód programu tudíž jednodušší. Vyžaduje však konfiguraci – potřebuje se dozvědět adresu lokálního serveru, který bude řešit jeho dotazy. Její nastavení musí zajistit místní správce, a to buď přímo odpovídajícím konfiguračním zásahem na koncovém stroji, nebo (častěji) prostřednictvím konfiguračního protokolu DHCP. Otázkám konfigurace DNS klienta se budeme opět věnovat v kapitole 4 na straně 85.

Nabízí se otázka, proč operační systémy neobsahují plnohodnotné resolvery. Jejich kód i nároky by sice byly o něco větší, rozdíl by však nebyl nijak zásadní a pro současný hardware v podstatě zanedbatelný. Zato bychom získali jednoduchost použití – plnohodnotný resolver je soběstačný a téměř bezúdržbový. Potřebuje jen adresy kořenových serverů, které se prakticky nemění, a případnou změnu by šlo řešit obvyklou cestou systémových aktualizací.

Tím bychom se ale připravili o hlavní výhodu, společnou vyrovnávací paměť. Jednoduché resolvery strojů v lokální síti posílají své dotazy místnímu rekurzivnímu serveru, který je řeší a výsledky si ukládá do vyrovnávací paměti. Jelikož se všechny lokální stroje obracejí na stejný server, mohou využívat záznamy, které zjistil při řešení dřívějších dotazů jejich sousedů. To podstatným způsobem zvyšuje efektivitu vyrovnávací paměti a snižuje objem síťového provozu odcházejícího mimo lokální síť.

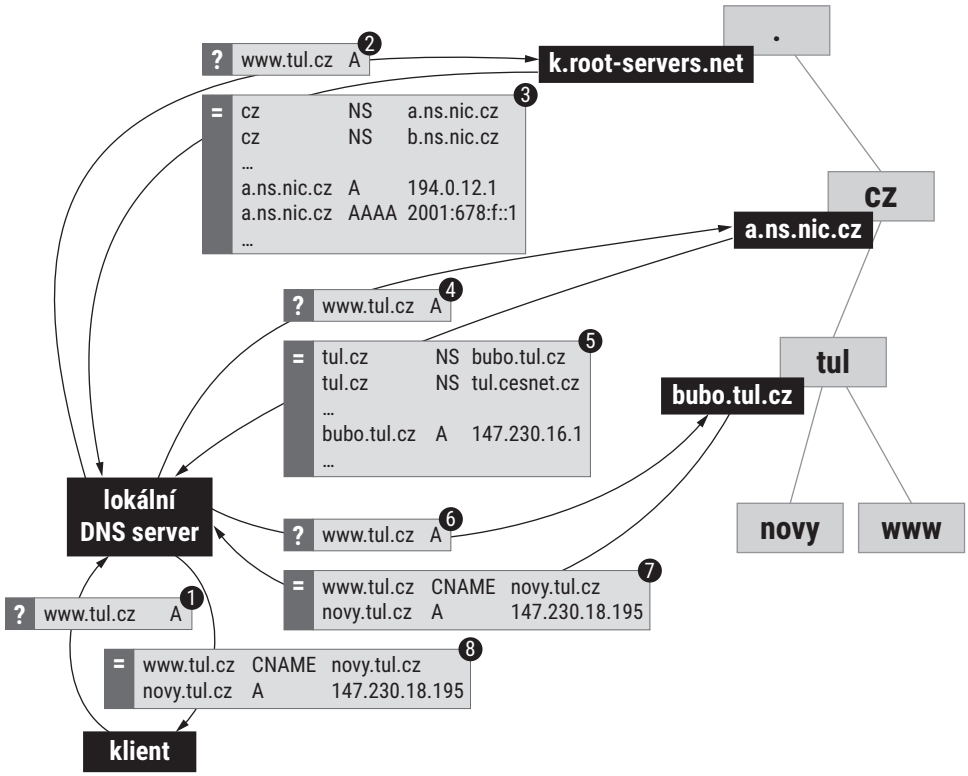
Kromě toho může rekurzivní server něco přidat k datům z veřejného DNS stromu. Jestliže se například v místní síti používají neveřejné adresy, neměly by se objevit ve veřejném DNS. Ovšem interně pro ně chcete používat jména. Často se tato situace řeší tak, že se rekurzivní server nastaví jako autoritativní pro speciální doménu, která obsahuje záznamy s neveřejnými adresami. Tato doména není v doménovém stromě, takže nedostává dotazy zvenčí. Ovšem místní stroje mu posílají všechny své dotazy, takže jim může zasílat odpovědi z této nestandardní domény.

Zanedlouho se k vyrovnávací paměti vrátíme, ale nejprve se podívejme na reálný příklad.

2.5 Život jednoho dotazu

Máme už pohromadě základní stavební kameny DNS – doménový strom, zdrojové záznamy, servery a klienty. Je načase podívat se, jak jednotlivá kolečka zapadají do sebe a jak celý systém funguje. Projděme si kroky, které proběhnou mezi položením dotazu a příchodem odpovědi.

Řekněme, že sháníte informace o Technické univerzitě v Liberci. Do webového prohlížeče proto zadáte *www.tul.cz* a DNS klient ve vašem počítači dostane za úkol najít k tomuto jménu odpovídající IP adresu. Ve své konfiguraci má zapsáno, kde začít – IP adresu lokálního DNS serveru obsluhujícího místní síť.



Obrázek 2.2: Hledání adresy pro *www.tul.cz*

Celá DNS transakce začne tím, že klient na základě požadavku vašeho prohlížeče vytvoří DNS dotaz požadující záznam typu A pro jméno *www.tul.cz* a zašle jej lokálnímu serveru. Zatím necháme stranou vyrovnávací paměti a popíšeme postup řešení dotazu bez nich. Je znázorněn na obrázku 2.2, jednotlivé kroky jsou opatřeny pořadovými čísly – toto byl krok 1.

Oslovený server se chopí řešení dotazu. Odpověď samozřejmě nezná, ale zaúkoluje svůj plnohodnotný resolver, jenž má k dispozici adresy kořenových serverů, u nichž začíná řešení všech dotazů. Lokální server některý z nich vybere, řekněme *k.root-servers.net*, a pošle mu stejný dotaz: hledám záznam typu A pro *www.tul.cz* (krok 2).

Hierarchické uspořádání domén svádí k myšlence, že dotaz bude postupně „probublávat" hierarchií vzhůru. Tak to ale není, cestu nahoru absolvuje jedním skokem. Je to rychlejší a snadněji udržitelné – stačí, aby každý místní server znal třináct adres kořenových serverů, které jsou pro všechny společné a stabilní. Mívá je ve své konfiguraci a při startu si zjišťuje jejich aktuální složení pomocí tak zvaných přípravných dotazů (priming queries, podrobněji se problematice věnuje RFC 8109 *Initializing a DNS Resolver with Priming Queries*).

Hierarchické uspořádání se začne uplatňovat až při sestupu doménovým stromem dolů. Oslovený kořenový server zná situaci v kořenové doméně. Ví, že doména nejvyšší úrovně *cz* skutečně existuje a kdo jsou její autoritativní servery. Na rozdíl od lokálního serveru však nemá čas dotaz řešit – dostává jich příliš mnoho. Proto rovnou pošle tazateli odpověď, která však místo požadovaného záznamu A pro *www.tul.cz* napovídá, kde se ptát dál (krok 3). V tomto konkrétním případě bude obsahovat informace o autoritativních serverech pro doménu *cz*:

```
cz.                 NS      d.ns.nic.cz.
cz.                 NS      a.ns.nic.cz.
cz.                 NS      b.ns.nic.cz.
cz.                 NS      c.ns.nic.cz.

d.ns.nic.cz.        AAAA    2001:678:1::1
c.ns.nic.cz.        AAAA    2001:678:11::1
b.ns.nic.cz.        AAAA    2001:678:10::1
a.ns.nic.cz.        AAAA    2001:678:f::1
d.ns.nic.cz.        A       193.29.206.1
c.ns.nic.cz.        A       194.0.14.1
b.ns.nic.cz.        A       194.0.13.1
a.ns.nic.cz.        A       194.0.12.1
```

Lokální server nyní může sestoupit o jedno patro hierarchie níže. Dozvěděl se jména a adresy autoritativních serverů pro doménu *cz*, takže stejný dotaz (hledám záznam typu A pro *www.tul.cz*) položí jednomu z nich, například *a.ns.nic.cz* (krok 4). Opakuje se podobná situace jako v kořeni, jen jsme o něco blíže řešení. Oslovený server opět nezná odpověď a nemá kapacitu dotaz řešit, ale zná doménu *cz*. Proto ví, že *tul.cz* skutečně existuje a jaké jsou její autoritativní servery. Tuto informaci pošle tazateli jako odpověď (krok 5):

```
tul.cz.             NS      tul.cesnet.cz.
tul.cz.             NS      bubo.tul.cz.

bubo.tul.cz.        A       147.230.16.1
bubo.tul.cz.        AAAA    2001:718:1c01:16::aa
```

Lokální server se zase o krok přiblížil vyřešení dotazu. Vybere si jeden z autoritativních serverů pro doménu *tl.cz*, třeba *bubo.tul.cz*, a zopakuje mu svůj dotaz (krok 6). Jelikož je poptávané jméno jen o jedno patro níže, je jisté, že tentokrát již dostane definitivní odpověď (krok 7). V daném případě:

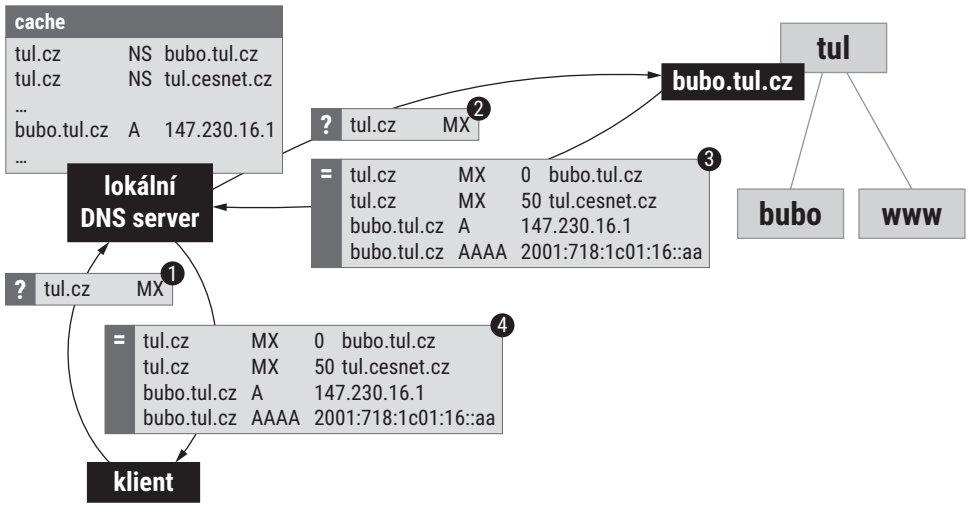
```
www.tul.cz.         CNAME   novy.tul.cz.
novy.tul.cz.        A       147.230.18.195
```

Říká se v ní, že jméno *www.tul.cz* je ve skutečnosti přezdívkou pro *novy.tul.cz*, jehož adresa je v odpovědi také obsažena. Předá ji klientovi (krok 8). Zdůrazněme, že postup řešení bude víceméně stejný odkudkoli z celého Internetu. Lišit se bude jen výběr oslovených serverů, vždy však lokální server nejprve osloví jeden z kořenových serverů, pak některý z autoritativních serverů pro doménu *cz* a do třetice vybere jeden z autoritativních serverů domény *tul.cz*. Obecně platí, že dotaz se vždy vyhoupne do kořene doménového stromu a pak postupně sestupuje po jednotlivých patrech blíže a blíže danému cíli.

Možná vám trochu vrtá hlavou, jak může kořenový server poslat záznam typu A třeba pro *a.ns.nic.cz*. Kořenová doména obsahuje pro *cz* záznamy typu NS, v nichž jsou uvedena doménová *jména* jejích autoritativních serverů. Vyvstává klasický problém slepice versus vejce: abychom se mohli zeptat autoritativního serveru *a.ns.nic.cz* na informace ohledně domény *cz*, potřebujeme jeho adresu. Ta je ovšem uložena kdesi v doméně *cz*.

Tento problém řeší tak zvané *slepující záznamy (glue records)*. Jedná se o adresní záznamy (typy A a AAAA) umístěné ve vyšších patrech doménové hierarchie, než by jim náleželo. Jestliže jméno autoritativního serveru pro určitou poddoménu samo leží v této poddoméně, je standardním postupem nezjistitelné. Proto se do nadřízené domény společně se záznamem typu NS uvádějícím jméno serveru umístí i záznamy obsahující IP adresy tohoto serveru. Kořenová doména tedy obsahuje slepující záznamy typu A a AAAA pro *a.ns.nic.cz* a doména *cz* podobně obsahuje slepující záznam A pro *bubo.tul.cz*. Pokud server odesílá příslušný záznam NS, automaticky do své zprávy přibalí i odpovídající slepující záznamy, pokud je má k dispozici.

V příkladu jsme zatím zcela ignorovali existenci *vyrovnávacích pamětí (cache)*. Nastal čas věnovat jim trochu pozornosti. Lokální server si samozřejmě uloží získanou odpověď pro *www.tul.cz* a pokud se v době jeho životnosti bude po této informaci shánět některý z místních klientů, odpoví mu rovnou. Celá DNS komunikace pak bude mít jen dva kroky: klient se zeptá místního serveru a ten mu rovnou pošle odpověď s příznakem, že není autoritativní.



Obrázek 2.3: Hledání poštovních serverů pro *tul.cz* s využitím cache

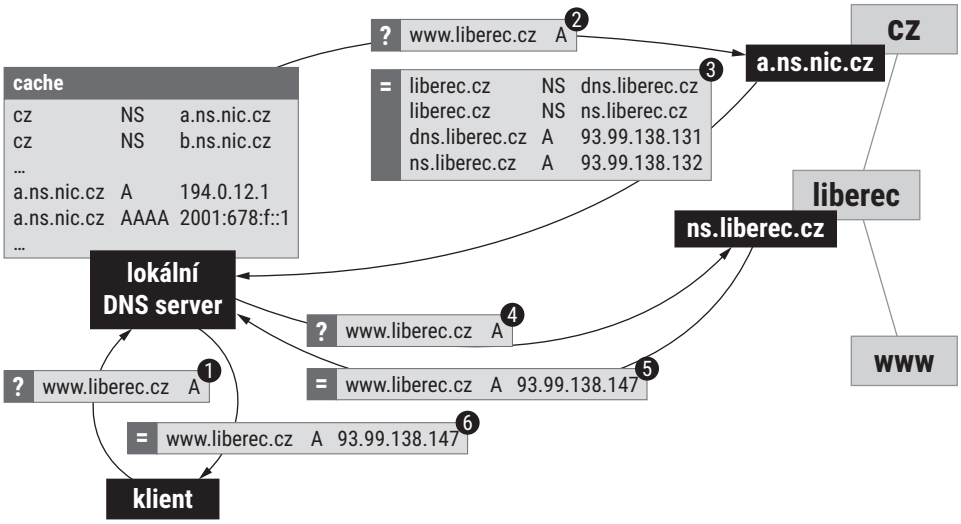
Server si však neukládá jen výsledné řešení, ale veškeré informace, které během jeho hledání získal. V tomto případě autoritativní servery pro domény *cz* a *tul.cz*. Když po chvíli brouzdání po stránkách pošlete elektronickou poštou na adresu *info@tul.cz* nějaký dotaz, bude váš poštovní server poptávat záznamy typu MX pro doménu *tul.cz*. Situaci ilustruje obrázek 2.3. Jeho resolver se obrátí na lokální DNS server, který má ve vyrovnávací paměti platné údaje o autoritativních serverech pro tuto doménu. Díky tomu se zeptá rovnou některého z nich a obratem se dozví:

```
tul.cz.              MX    0 bubo.tul.cz.
tul.cz.              MX    50 tul.cesnet.cz.

bubo.tul.cz.         A     147.230.16.1
bubo.tul.cz.         AAAA  2001:718:1c01:16::aa
```

Dorazí-li zanedlouho lokálnímu DNS serveru dotaz na *www.liberec.cz* (chystáte se studovat v Liberci a zajímá vás, co je ve městě k vidění) využije uložené informace o autoritativních serverech pro doménu *cz* a zeptá se některého z nich, aniž by začínal u kořenového serveru. Jak by to zhruba vypadalo, vidíte na obrázku 2.4.

Při řešení dotazů a uplatňování vyrovnávacích pamětí hraje klíčovou roli skutečnost, že doménové jméno je zároveň jednoznačným popisem cesty doménovým stromem, jež vede k získání



Obrázek 2.4: Hledání adresy *www.liberec.cz* s využitím cache

požadované informace. Díky tomu každý z oslovených serverů okamžitě ví, kudy se má pokračovat k vyřešení. Zároveň si dokáže najít ve vyrovnávací paměti nejkonkrétnější relevantní informaci a tu využít k omezení síťové komunikace a zrychlení odezvy.

2.6 Rekurzivní a nerekurzivní chování serverů

V příkladu z předchozí části jsme popsali dva výrazně odlišné způsoby chování DNS serverů. Zatímco lokální server se dotazu chopil, vyřešil jej a poslal výslednou odpověď, kořenový server se dotazem nezabýval a jako odpověď poskytl obratem informace o serverech, které mají blíže k řešení. Pro tyto dva duhy chování existují i oficiální názvy:

- *Rekurzivní řešení dotazu* je takové, kdy se server dotazu ujme, vyřídí veškerou potřebnou korespondenci až po získání odpovědi, kterou pak odešle tazateli. Rekurzivní chování server samozřejmě zatěžuje. Musí si uchovávat datové struktury pro rozpracované dotazy, přeposílat dotazy a podle reakcí na ně se rozhodovat, jak pokračovat dál. Hlavní a základní výhodou rekurzivního řešení dotazů je, že si server plní vyrovnávací paměť. Procházejí jím dílčí i finální odpovědi, které si ukládá a může je využít při řešení dalších dotazů. Toto chování je typické pro lokální servery, které fungují jako rekurzivní resolvery obsluhující jednoduché resolvery místních strojů.

- *Nerekurzivní řešení dotazu* znamená, že server jednoduše odkáže tazatele jinam. Obdrží dotaz, nahlédne do svých dat, odešle jako odpověď informace o autoritativních serverech bližších k řešení (nebo chybový kód, pokud jste se zeptali na doménu, k níž nemá co říci) a pustí dotaz z hlavy. Nemusí si nic uchovávat, zátěž serveru je minimální, což je přesně ten důvod, proč se nerekurzivně chovají autoritativní servery v horních patrech doménové hierarchie – tedy kořenové a servery domén nejvyšší úrovně.

Obecné doporučení zní, že v každé síti připojené k Internetu by měl být místní rekurzivní server, na nějž se obracejí zdejší klienti. Jeho hlavní úlohou je poskytnout společnou vyrovnávací paměť a omezit tak objem DNS komunikace směřující do Internetu. Pokud hledáte hojně používaný server, třeba *www.seznam.cz*, skoro jistě budete obslouženi z vyrovnávací paměti, protože adresu před vámi požadoval někdo jiný. Čím více klientů se schází na jednom lokálním serveru, tím vyšší je pravděpodobnost opakování dotazů a efektivita využití uložených informací. Proto není dobrý nápad stavět rekurzivní servery na každém rohu. Ideální počet je jeden pro celou místní síť. Tím ovšem vzniká slabé místo, protože při jeho výpadku přestane místním strojům fungovat DNS. Pokud je síť větší, je rozumné postavit do ní dva rekurzivní DNS servery – jeden hlavní a jeden záložní pro případ výpadku prvního. Více rozhodně ne.

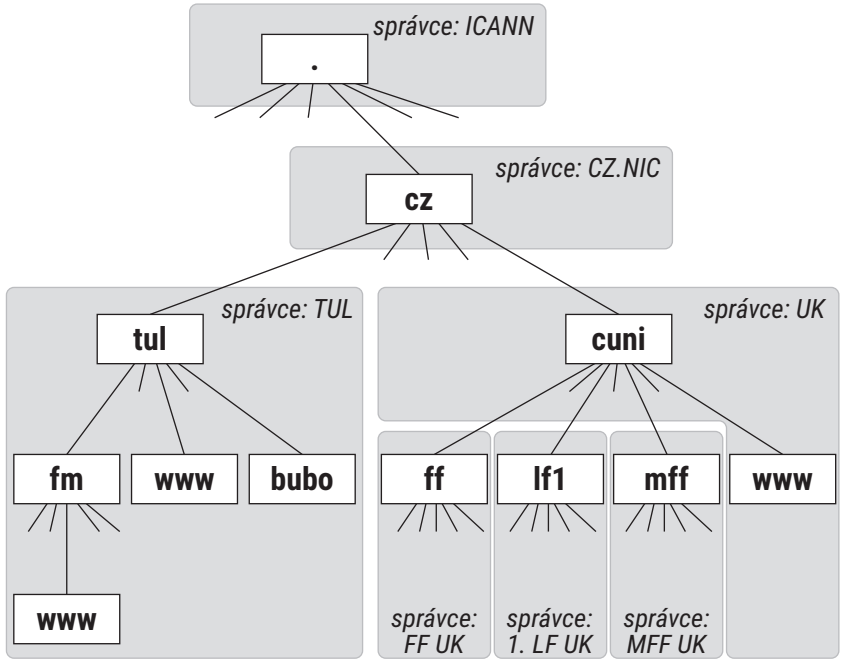
Na druhé straně je však třeba upozornit, že rekurzivnímu serveru se dají podstrčit falešné informace a on následně bude své klienty mystifikovat. Je proto velmi důležité, aby byl správně konfigurován a chránil se před podobnými pokusy. Naprostou samozřejmostí by mělo být, že rekurzivní služby poskytuje jen strojům z místní sítě, nikoli veřejně do Internetu. Otázkami bezpečnosti DNS se budeme zabývat v kapitole 15 na straně 345.

Pokud má koncová síť přidělenu vlastní doménu, otevírá se zajímavá otázka vztahu autoritativních serverů této domény a pomocných (rekurzivních) serverů koncové sítě. Lze k ní přistoupit dvěma způsoby: puristicky nebo úsporně. Vhodnější puristická varianta používá zcela oddělené servery, zatímco úsporná sází na jeden společný server pro obě role. Rozebereme to podrobněji v části 6.1.2 na straně 133.

2.7 Domény, zóny a zónové soubory

V textu jsme opakovaně zdůraznili distribuovaný charakter DNS databáze. Jednotlivé části doménového stromu jsou spravovány různými subjekty a informace poskytovány různými servery. Souvislá část doménového stromu, jejíž správa byla delegována jednomu subjektu, je označována jako *zóna (zone)*. Jednou zónou je například kořenová doména, jejíž správu vykonává ICANN. Pod ní se nachází například zóna odpovídající doméně *cz* spravovaná sdružením CZ.NIC.

Hranice mezi zónami prochází vždy hranou doménového stromu, jež spojuje nadřazenou doménu spadající do jedné zóny s podřazenou, která je již v jiné zóně. Pro tyto hranice se používá pojem *zónový řez (zone cut)* a označují vlastně hranice zodpovědnosti jednotlivých subjektů. Zónový řez znamená „od tohoto místa spravuje připojený podstrom někdo jiný". Dotyčný správce může skutečně spravovat celý podstrom, nebo jeho části odřízne a svěří do správy dalším subjektům. To už je čistě otázkou jeho rozhodnutí.



Obrázek 2.5: Rozdělení domén do zón

Podívejme se na příklad části doménového stromu na obrázku 2.5, který popsaný princip ilustruje. Správce kořenové zóny odřízl její poddoménu *cz* a svěřil ji CZ.NIC. Ten pochopitelně neobhospodařuje celý příslušný podstrom, ale opět svěřuje jednotlivé domény druhé úrovně subjektům, které o ně projeví zájem. Takže například doménu *tul.cz* získala do správy Technická univerzita v Liberci a vznikla tak další zóna. Celou doménu zde obhospodařuje centrální správa sítě, proto pod *tul.cz* nenajdete žádné další řezy a zóna zahrnuje celý podstrom. Jiný přístup zvolila podstatně větší Univerzita Karlova, která delegovala fakultní domény do správy příslušných fakult. Pod zónou *cuni.cz* spravovanou centrálním Ústavem výpočetní techniky UK tak vznikly zóny *mff.cuni.cz*, *lf1.cuni.cz*, *ff.cuni.cz* a další, jejichž správu si zajišťují přímo odpovídající fakulty.

Existence zón se promítá i do obsahu DNS. V doméně bezprostředně nad zónovým řezem musí být informace o tom, že podřízená doména existuje, avšak má už jiné autoritativní servery. To zajišťují záznamy typu NS. Například doména *cz* proto obsahuje dvojici záznamů NS pro doménu *tul.cz* se jmény jejích autoritativních serverů *bubo.tul.cz* a *tul.cesnet.cz*. Jména obou serverů leží v zónách podřízených zóně *cz*, proto musí obsahovat také slepující záznamy s jejich adresami. Právě tyto záznamy pošle autoritativní server domény *cz* tazateli, který se bude zajímat o nějaké informace z domény *tul.cz*.

Každá zóna ve svém kořeni obsahuje záznam typu SOA (Start of Authority), ze kterého zjistíte jméno jejího primárního serveru, e-mail správce a časové konstanty, které ovlivňují zacházení s jejími daty. Podrobně je popsán v části 17.7 na straně 376.

Data zóny bývají na serveru uložena v podobě tak zvaného *zónového souboru*, jehož podobu definuje RFC 1035 pod názvem *master file*. Je to poněkud nezvyklé, protože formát lokálních dat by měl být spíše věcí implementace a některé se skutečně od standardní podoby odchylují[3]. Nicméně většina serverů tuto syntaxi dodržuje a používá se také ve všech dokumentech o DNS. Budeme se jí držet i v naší knize.

Jedná se o textový soubor, v němž je každý záznam zapsán na jeden řádek. Pokud jsou data dlouhá (například v záznamu typu SOA), lze je pro lepší čitelnost uzavřít do závorek a rozdělit do několika řádků. Uvnitř závorek má konec řádku stejný význam jako mezera. Záznam má pět položek vzájemně oddělených libovolnou kombinací mezer a/nebo tabulátorů:

doménovéJméno životnost třída typ data

Doménové jméno určuje jméno, k němuž se záznam vztahuje. Bývá také označováno jako vlastník záznamu. Pokud je zakončeno tečkou, chápe se jako absolutní a úplné. V opačném případě se za ně doplní aktuální doména. Takže například jméno *www* v zónovém souboru pro *nic.cz* znamená *www.nic.cz*. Jméno může na začátku záznamu zcela chybět, v takovém případě se zdědí z předchozího. Jestliže jako jméno použijete samotný znak „@“, označuje aktuální doménu. Tedy doménu, v jejímž zónovém souboru se vyskytuje nebo která byla určena direktivou $ORIGIN, jíž se zanedlouho budeme věnovat.

Životnost určuje počet sekund životnosti záznamu ve vyrovnávací paměti a *třída* jeho třídu. Jejich pořadí může být prohozeno. Údaje také mohou chybět a server za ně dosadí implicitní hodnoty. Poslední dvě položky jsou jako jediné povinné. Určují *typ* záznamu a v závislosti na něm pak *data*, která obsahuje.

3: Existují i servery, jež si lokálně data ukládají do databází, tedy principiálně odlišně od oficiální podoby.

Hodnoty jednotlivých položek tvoří celá čísla, doménová jména, IP adresy a řetězce znaků. Jejich zápisy odpovídají běžným konvencím. V případě celých čísel zdůrazněme, že jsou zapisována v desítkové soustavě jako souvislé sekvence číslic. Nelze do nich vkládat oddělovače řádů (mezery, čárky a podobně).

Výše jsme uvedli, že zóna *cz* musí obsahovat záznamy NS pro autoritativní servery *tul.cz* a také odpovídající slepující záznamy. Příslušný úsek zónového souboru by vypadal takto:

```
tul             NS    bubo.tul
                NS    tul.cesnet
bubo.tul        A     147.230.16.1
                AAAA  2001:718:1c01:16::aa
tul.cesnet      A     195.113.233.250
```

Všimněte si, že všechna jména jsou relativní (nekončí tečkou), takže se za ně doplní doména, v jejímž souboru se vyskytují. Například *tul* proto znamená *tul.cz*.

Řetězce znaků, jsou-li souvislé, lze psát přímo tak jak jsou. Pokud však má řetězec obsahovat mezery, je třeba obklopit jej uvozovkami (`"`), aby nebyly pochopeny jako oddělovače položek. Řetězec v uvozovkách může obsahovat i konce řádků a být tedy rozdělen do několika řádků za sebou. Pokud má obsahovat uvozovky, vložte před ně zpětné lomítko (`\"`). Tento přístup je obecný – chcete-li potlačit speciální význam libovolného znaku, zapište před něj zpětné lomítko. O vložení zpětného lomítka do řetězce se tudíž postará `\\`. Zcela libovolný znak lze vložit konstrukcí `\`*DDD*, kde *DDD* jsou číslice udávající kód znaku v osmičkové soustavě. Také tyto znaky jsou interpretovány jako pasivní část textu a je potlačen jejich případný speciální význam. Alternativním zápisem pro vložení uvozovek by tedy bylo `\042`, protože znak `"` má ASCII kód 34, v osmičkové soustavě 42.

Ke zvýšení čitelnosti lze do zónového souboru vkládat prázdné řádky a komentáře. Ty začínají středníkem, všechny následující znaky až po konec řádku jsou interpretovány jako komentář

$ORIGIN *jméno*	změní aktuální doménu – počínaje následujícím řádkem bude za relativní doménová jména doplňováno uvedené *jméno*
$INCLUDE *soubor*	vloží obsah daného *souboru* jako by jeho text byl zapsán v místě výskytu direktivy
$TTL *čas*	nastavuje implicitní dobu životnosti záznamů v zóně (definována v RFC 2308)

Tabulka 2.2: Direktivy zónových souborů

a serverem ignorovány. Kromě běžných záznamů se v zónovém souboru mohou objevit také tak zvané *direktivy*, které představují příkazy ovlivňující způsob interpretace souboru. Různé servery si přidávají vlastní, ale standardně jsou definovány jen tři, jež shrnuje tabulka 2.2.

Zónový soubor pro velmi zjednodušenou zónu *tul.cz*, která obsahuje jen web, server pro příjem pošty a jednu poddoménu s vlastním webem, by mohl vypadat například takto:

```
$TTL 48h
$ORIGIN tul.cz.
@       SOA  bubo.tul.cz. pavel\.satrapa.tul.cz. (
               2023071001 ;sériové číslo
               86400      ;aktualizovat po 1 dni
               7200       ;opakovat po 2 hod
               3600000    ;zrušit po 1000 hod
               172800  )  ;minimum 2 dny

        NS     bubo
        NS     tul.cesnet.cz.
        MX     0  bubo
        MX     50 tul.cesnet.cz.

bubo    A      147.230.16.1
        AAAA   2001:718:1c01:16::aa
        MX     0  bubo
web     A      147.230.16.27
        AAAA   2001:718:1c01:16:216:3eff:fe1a:d23f
www     CNAME web

$ORIGIN fm.tul.cz.
; poddoména fm.tul.cz
; zavádíme sirius.fm.tul.cz a www.fm.tul.cz

sirius A      147.230.72.241
www    CNAME sirius
```

V souboru používáme relativní jména, abychom jej zkrátili a učinili o něco přehlednějším. U jmen ležících mimo *tul.cz* (zde například *tul.cesnet.cz*) je třeba nezapomenout na konci tečku, aby byla správně interpretována jako absolutní. Bez závěrečné tečky by *tul.cesnet.cz* bylo interpretováno jako *tul.cesnet.cz.tul.cz*.

Zde zónový soubor obsahuje kompletní data pro celou zónu. To odpovídá logice věci, nicméně není to nezbytné a pro velké zóny to nebude ani příliš praktické – se soubory obludné velikosti se po všech stránkách špatně pracuje. Bylo by klidně možné pro *fm.tul.cz* vytvořit samostatnou zónu s vlastním souborem:

```
$TTL 48h
$ORIGIN fm.tul.cz.
@       SOA  bubo.tul.cz. pavel\.satrapa.tul.cz. (
               2023071001 ;sériové číslo
               86400      ;aktualizovat po 1 dni
               7200       ;opakovat po 2 hod
               3600000    ;zrušit po 1000 hod
               172800  )  ;minimum 2 dny

        NS     bubo
        NS     tul.cesnet.cz.

sirius  A      147.230.72.241
www     CNAME  sirius
```

Pokud jsou obsluhovány stejnými servery, není třeba ani do *tul.cz* zavádět delegaci záznamem NS. Tazatel hledající *www.fm.tul.cz* se propracuje k autoritativním serverům pro *tul.cz*, jež mají oba zónové soubory a rovnou mu odpovědí. Z původního zónového souboru by tedy stačilo vymazat závěrečné řádky (počínaje druhým $ORIGIN). Koncepčnější ale je přidat do rodičovské domény záznamy s delegací, stejně jako by podřízenou doménu spravoval jiný subjekt. Závěrečné řádky původního zónového souboru je tedy lépe nahradit dvojicí:

```
fm      NS     bubo
        NS     tul.cesnet.cz.
```

Pokud spravujete zónu s poddoménami, je jen na vašem rozhodnutí, zda ji celou uložíte do jednoho souboru, nebo zda ji rozdělíte po jednotlivých doménách – obě varianty mají své přednosti.

2.8 Reverzní dotazy aneb hledá se jméno pro adresu

Reverzní dotaz vychází z IP adresy (verze 4 nebo 6) a snaží se zjistit, jaké doménové jméno jí náleží. Zlidšťují se tak protokoly o činnosti serverů, výstupy testovacích programů typu *traceroute*, záznamy o cestě elektronického dopisu a podobně. Aby se dalo DNS použít k řešení tohoto typu problémů, musí se z adresy vytvořit určité speciální doménové jméno, které se následně stane předmětem dotazu. Nepůjde to ale udělat přímočaře, protože doménové jméno má obecné části

vzadu a směrem dopředu se zpřesňuje, zatímco v případě adresy je směr přesně opačný. Začíná obecnou adresou sítě, za ní následuje podsíť a až na konci se nachází adresa konkrétního stroje v podsíti.

DNS je postaveno na principu distribuované správy, aby informace mohli upravovat místní správci. Proto bylo třeba vymyslet způsob převodu IP adresy na dotazované jméno, který by umožnil například doménu odpovídající síťovému prefixu 147.230.0.0/16 svěřit do správy organizaci, jíž byla přidělena dotyčná síť.

Řešením je prosté otočení adresy. Konkrétně z IPv4 adresy se vytvoří poptávané doménové jméno tak, že se vyjde z jejího standardního zápisu, v něm se obrátí pořadí jednotlivých bajtů a na konec se přidá přípona *in-addr.arpa*. Pokud bychom například hledali, pod jakým jménem je registrována adresa 147.230.16.27, poptávali bychom doménové jméno:

27.16.230.147.in-addr.arpa

Síti 147.230.0.0/16 odpovídá v tomto pojetí doména *230.147.in-addr.arpa*, kterou lze snadno svěřit do správy držiteli adresy. Distribuovaná správa je možná a informace nadále mohou být do DNS vkládány přímo tam, kde vznikají. Pro reverzní dotazy byl vytvořen speciální typ zdrojových záznamů nazvaný PTR (pointer). Jeho hodnotou je doménové jméno odpovídající příslušné adrese. Například dotaz na záznam typu PTR pro výše uvedené doménové jméno vás oblaží odpovědí, že náleží stroji *web.tul.cz*.

TUL, jejíž doménu *tul.cz* jsme používali pro příklady v předchozí části, disponuje právě sítí 147.230.0.0/16. Zónový soubor pro její reverzní doménu *230.147.in-addr.arpa*, který by odpovídal našemu příkladu výše, by vypadal následovně:

```
$TTL 48h
$ORIGIN 230.147.in-addr.arpa.
@       SOA   bubo.tul.cz. pavel\.satrapa.tul.cz. (
              2023071001 ;sériové číslo
              86400      ;aktualizovat po 1 dni
              7200       ;opakovat po 2 hod
              3600000    ;zrušit po 1000 hod
              172800  )  ;minimum 2 dny

        NS    bubo.tul.cz.
        NS    tul.cesnet.cz.

1.16    PTR   bubo.tul.cz.
27.16   PTR   web.tul.cz.
241.72  PTR   sirius.fm.tul.cz.
```

Jsme v reverzní doméně, relativní jména jsou proto vztažena k ní – například *1.16* znamená *1.16.230.147.in-addr.arpa*. Veškerá jména z domény *tul.cz* musí být zapsána jako absolutní. Považujeme za rozumné držet celou zónu pohromadě v jednom souboru, nedelegovat její části do dalších zón.

Popsaný systém byl navržen v době, kdy IPv4 adresy patřily do tříd A, B nebo C a přidělované prefixy končily vždy na hranici bajtů. Když bylo v polovině 90. let nasazeno beztřídní adresování (CIDR, Classless Inter-Domain Routing), vznikl problém, protože najednou mohla koncová síť dostat prefix libovolné délky. Poskytovatel Internetu, který má při správě adres roli lokálního internetového registru (LIR), například dostal pro své zákazníky k dispozici prefix 10.1.2.0/24. Vzhledem k velikosti koncových sítí a aktuálním pravidlům se jej rozhodl rozdělit na čtyři části různé velikosti a ty přidělit jednotlivým zákazníkům:

- prefix 10.1.2.0/25 dostal zákazník xyz1,
- prefix 10.1.2.128/26 dostal zákazník xyz2,
- prefix 10.1.2.192/27 dostal zákazník xyz3 a
- prefix 10.1.2.224/27 dostal zákazník xyz4.

Při zachování původní koncepce reverzních zón bylo jedinou možností, aby jejich společnou zónu *2.1.10.in-addr.arpa* spravoval poskytovatel a na základě požadavků jednotlivých zákazníků do ní vkládal záznamy pro jejich stroje. Takové uspořádání je ale dost nepraktické a popírá princip, aby si reverzní zónu spravoval sám držitel příslušného prefixu. Bylo potřeba vymyslet způsob, jak reverzní zónu delegovat kdesi uprostřed bajtu.

S řešením přišlo RFC 2317 *Classless IN-ADDR.ARPA delegation*. Do reverzního jména navrhlo vložit další doménu, která popisuje rozdělení bajtu a umožňuje delegovat jednotlivé poddomény různým subjektům. Toto vložení nelze udělat obecně, protože adresní prostor je členěn podle potřeby, jeho jednotlivé části se liší a zdaleka ne každá potřebuje delegaci uvnitř bajtu. Držitel či správce prefixu, na jehož konci došlo k rozdělení, musí tuto skutečnost zanést do odpovídající reverzní zóny. V bajtu, ve kterém došlo k rozdělení, zavede poddomény pro jeho jednotlivé části a hodnoty do nich rozdělí. Používají se k tomu záznamy typu CNAME (přezdívka, viz 17.3 na straně 371).

Konkrétně v našem příkladu by poskytovatel v doméně *2.1.10.in-addr.arpa* zavedl poddomény *0/25*, *128/26*, *192/27* a *224/27* a delegoval je příslušným zákazníkům. Kromě toho by do domény *2.1.10.in-addr.arpa* vložil záznamy CNAME, jimiž by převedl původní reverzní jména na jména v těchto poddoménách. Všechny adresy přidělené prvnímu zákazníkovi by byly převedeny do domény *0/25.2.1.10.in-addr.arpa*, takže například jméno *15.2.1.10.in-addr.arpa* by se stalo přezdívkou pro *15.0/25.2.1.10.in-addr.arpa*. Pro adresy druhého zákazníka by přezdívky vedly do domény *128/26.2.1.10.in-addr.arpa* – například *137.2.1.10.in-addr.arpa* by se díky CNAME změnila na *137.128/26.2.1.10.in-addr.arpa* a tak dále.

Výhodou je, že dělící zónu lze připravit předem a celou, bez ohledu na to, zda adresy byly přiděleny nebo nikoli. Zóna bude obsahovat záznamy CNAME pro všech 256 možných hodnot a až při dotazu autoritativních serverů jednotlivých poddomén spravovaných zákazníky se zjistí, zda pro danou adresu existuje PTR záznam či nikoli. Okamžitě po přidělení prefixů by tedy poskytovatel mohl vytvořit zónový soubor pro doménu *2.1.10.in-addr.arpa* s následujícím obsahem:

```
$TTL 48h
$ORIGIN 2.1.10.in-addr.arpa.
@       SOA  ...

;poddomény spravované zákazníky
0/25    NS  ns1.xyz1.cz
        NS  ns2.xyz1.cz
128/26  NS  ns1.xyz2.cz
        NS  ns2.xyz2.cz
192/27  NS  ns1.xyz3.cz
        NS  ns2.xyz3.cz
224/27  NS  ns1.xyz4.cz
        NS  ns2.xyz4.cz

;pro všechny adresy v rozsahu prvního zákazníka
0       CNAME  0.0/25
1       CNAME  1.0/25
2       CNAME  2.0/25
...
127     CNAME  127.0/25

;pro všechny adresy v rozsahu druhého zákazníka
128     CNAME  128.128/26
129     CNAME  129.128/26
130     CNAME  130.128/26
...
191     CNAME  191.128/26

;pro všechny adresy v rozsahu třetího zákazníka
192     CNAME  192.192/27
193     CNAME  193.192/27
...
223     CNAME  223.192/27
```

```
;pro všechny adresy v rozsahu čtvrtého zákazníka
224     CNAME  224.224/27
225     CNAME  225.224/27
...
255     CNAME  255.224/27
```

Poddomény pak budou obsahovat běžné záznamy typu PTR pro existující stroje. Reverzní zóna *0/25.2.1.10.in-addr.arpa* spravovaná zákazníkem xyz1 by mohla obsahovat například:

```
$TTL 48h
$ORIGIN 0/25.2.1.10.in-addr.arpa.
@       SOA  ...
        NS   ns1.xyz1.cz.
        NS   ns2.xyz1.cz.

1       PTR  www.xyz1.cz.
2       PTR  ns1.xyz1.cz.
3       PTR  ns2.xyz1.cz.
4       PTR  podpora.xyz1.cz.
35      PTR  pc1.xyz1.cz.
36      PTR  pc2.xyz1.cz.
127     PTR  router.xyz1.cz.
```

Pro IPv6 adresy se používá analogický postup, jen je vzhledem k zápisu adres o něco složitější. Opět vychází ze zápisu adresy, který se rozvine na úplný tvar – tedy doplní se všechny vynechané nuly. Pořadí jednotlivých šestnáctkových číslic v adrese se obrátí a každá z nich se stane samostatnou doménou. Za ně se pak přidá konstantní přípona *ip6.arpa*. Vezměme jako příklad adresu 2001:db8:89ab:1:2a0:ecff:fe12:345. Doplníme chybějící nuly:

2001:0db8:89ab:0001:02a0:ecff:fe12:0345

otočíme pořadí šestnáctkových číslic, uděláme z nich poddomény a připojíme *ip6.arpa*. Výsledný dotaz tedy bude poptávat doménové jméno:

5.4.3.0.2.1.e.f.f.f.c.e.0.a.2.0.1.0.0.0.b.a.9.8.8.b.d.0.1.0.0.2.ip6.arpa

Tento přístup opět umožňuje delegovat poddomény v souladu s přidělenými částmi adresního prostoru, navíc to vzhledem k důsledné agregaci adres půjde dělat hierarchicky. Vlastní informace o příslušném doménovém jméně je uložena v záznamu typu PTR, stejně jako v případě IPv4.

Nikde není zaručeno, že dopředná a reverzní informace budou konzistentní. Tedy že odpověď získaná reverzním dotazem je skutečným doménovým jménem, pod nímž je zaregistrována daná adresa. Nesrovnalosti vznikají chybami, v horším případě úmyslně s cílem podvádět. Nikdo totiž nemůže zabránit správci reverzní domény *230.147.in-addr.arpa*, aby do ní zanesl, že některá ze zdejších adres přísluší třeba *www.google.cz*. Pokud na tom záleží (například při omezování přístupu ke službě podle klientovy domény), je záhodno si získané jméno ověřit – nechat si pro ně najít adresy a podívat se, zda je původní adresa mezi nimi. Pokud ne, nejsou data v reverzní doméně v pořádku.

Řekněme, že nejmenovaný server má povolen přístup pouze pro stroje z domény *tul.cz*. Dorazí-li mu paket z IPv4 adresy 147.230.33.44, získá reverzním dotazem jeho jméno, řekněme *pc33.tul.cz*. Jelikož se této informaci nedá věřit, poptá vzápětí záznamy typu A pro *pc33.tul.cz*. Ty už do DNS vkládá správce domény *tul.cz* a má je pod kontrolou. Pokud se v odpovědi dozví třeba:

```
pc33    A    147.230.11.12
        A    147.230.33.44
```

může nalezenému jménu důvěřovat, protože zkoumaná adresa se nachází v jeho záznamech A.

Daleko častější chybou než úmyslné falšování je chybějící reverzní záznam. Nemálo správců se jejich údržbou nezatěžuje a pokus o zjištění jména k takové adrese skončí neúspěchem.