VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

DNS Lookup nástroj

IPK - Počítačové komunikace a sítě

9.4.2018 Matěj Mitaš (xmitas02)

Obsah

1 Zadání

Úkolem bylo vytvořit nástroj pro překlad doménových jmen. Projekt obsahuje spustitelný soubor *ipk-lookup* s následující konvencí volání:

./ipk-lookup -s server [-T timeout] [-t type] [-i] name

- h (help) - volitelný parametr, při jeho zadání se vypíše nápověda a program se ukončí.
- s (server) - povinný parametr, DNS server (IPv4 adresa), na který se budou odesílat dotazy.
- T (timeout) - volitelný parametr, timeout (v sekundách) pro dotaz, výchozí hodnota 5 sekund.
- t (type) - volitelný parametr, typ dotazovaného záznamu: A (výchozí), AAAA, NS, PTR, CNAME.
- name - překládané doménové jméno, v případě parametru -t PTR program na vstupu naopak očekává IPv4 nebo IPv6 adresy.

2 Teorie o DNS

Díky přednáškám jsem měl obecnou představu, jak je protokol DNS navržen. Nicméně pro praktickou implementaci bylo třeba prozkoumat více zdrojů. (viz sekce Reference).

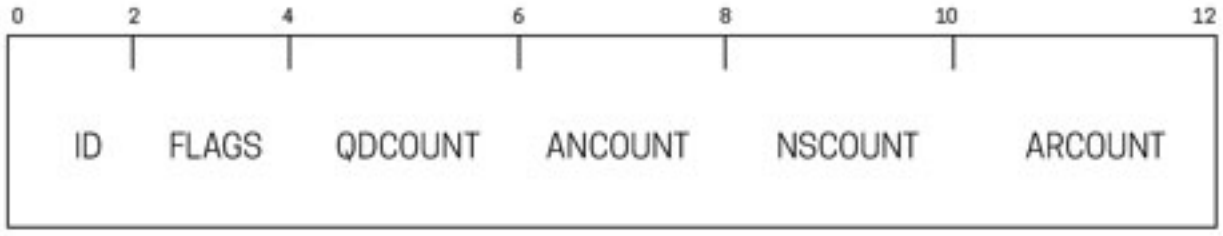
DNS Query obecně nabývá dvou hodnot, dotaz a odpověd. Pro svůj přenos používá protokol UDP, jelikož v množství požadavků by jakékoliv aplikace či obecněji programy pracující s touto technologií byly neúměrně zdržovány ať již tvořením, či dekodováním. Obecně je maximální délka $512B$. V případě překročení tohoto limitu se nastaví přepínač TC v hlavičce, ale o tom níže.

2.1 Obecná struktura DNS dotazu

Délka jednotlivých částí se může lišit, avšak jde hlavně o to, aby dotaz měl maximálně $512B$.

- **HEADER** Obsahuje identifikační a příznakovou část dat.
- **QUESTION** Sekce dotazů, zde je to nejdůležitější pro dotazy.
- **ANSWER** Sekce odpovědí, zde je to nejdůležitější pro odpovědi.
- **AUTHORITY** Informace o autoritativních serverech.
- **ADDITIONAL** Další informace.

2.2 DNS Header (hlavička)



pozn. uvedené hodnoty v grafu značí počet BAJT Ů určených pro jednotlivé části protokolu.

- **ID** - *2B* - Unikátní id, které posíláme v každém dotazu (zde použito *43951*) pro zpětnou kontrolu odpovědí. V případě nesouladu vypíšeme chybu.
- **Sekce příznaků** - *2B* - Příznaky, kterými ovládáme vlastní zprávu (viz níže).
- **QDCOUNT** - *2B* - Počet dotazů.
- **ANCOUNT** - *2B* - Počet odpovědí.
- **NSCOUNT** - *2B* - Počet jmenných serverů.
- **ARCOUNT** - *2B* - Počet doplňjících informací.

2.2.1 DNS Header Flags (příznaky)

pozn. uvedené hodnoty v grafu značí počet BITŮ určených pro jednotlivé části protokolu.

- **QR** - *1b* - 0: query (dotaz), 1: response (odpověď).
- **Opcode** - *4b* - 0: query (standartní dotaz), více než 1, jsou poté speciální typ, které jsou mimo rozsah tohoto projektu (IQUERY, STATUS, UPDATE, NOTIFY).
- **AA** - *1b* - 0: odpověď od neautoritativního serveru, 1: odpověď od autoritativního serveru.
- **TC** - *1b* - 0: bez oseknutí (překročíme $512B$), 1: oseknutí, použij TCP pro komunikaci.
- **RA** - *1b* - Dostupnost rekurze (pro rekurzivní a iterativní vyhledávání).
- **RD** - *1b* - Chceme rekurzi (pouze pro rekurzivní).
- **Z** - *3b* - Další informace (zpravidla prázndné)
- **Rcode** - *1b* - Návratové kódy operace.
  0 - V pořádku.
  1 - Chybný dotaz.
  2 - Server neumí odpovědět
  3 - Jméno neexistuje.
  4 - Nepodporovaný typ dotazu.
  5 - Refused od serveru.

Samotná hlavička je v mé implementaci reprezentována níže uvedenou strukturou, všimněme si položky *opt*, práce s ním bude vysvětlena v implementační sekci.

```
typedef struct ipk18_dns_Header {
        uint16_t id;
        uint16_t opt;
        uint16_t qdcount; // kolik vet ma dotaz
        uint16_t ancount; // kolik vet ma odpoved
        uint16_t nscount; // sekce pro odkazy na autoritativni servery
        uint16_t arcount; // doplnujici informace
} ipk18_dns_Header;
```

2.3 DNS Query (dotaz)

Je třeba vyplnit pole dotazu, na specifikace práce s *QNAME* narážím v části Implementace. V projektu použitelné tyto možnosti *QTYPE* a *QCLASS*.

- **QNAME** - *1B * počet znaků* - Doménové jméno, či IP adresa (viz Implementace)
- **QTYPE** - *2B* - Typ požadavku

  1 - A (IPv4 adresa).

  28 - AAAA (IPv6 adresa).

  2 - NS (jmenný server).

  5 - CNAME (CNAME záznam).

  12 - PTR (Reverzní překlad).
- **QCLASS** - *2B* - 1: IN, třída internet

2.4 DNS Response (odpověď)

Pole odpovědí (může být i více než jedna, vždy však v rámci $512B$), DNS používá komprimaci, proto tedy při dotazu na QNAME *www.matejmitas.com* nekopíruje do každé odpovědi celý string. Místo toho na jeho pozici vloží konstantu $0xc0$ a přímo za ní uloží adresu skoku (zde $0x0c$ což odpovídá indexu 12, tedy místa, kde je daný string uložený pro zmenšení datové náročnosti.

```
08 00 45 00   ...'.z.. ......E.
08 08 64 41   .PB...:. ......dA
81 80 00 01   C..5...< qP......
61 74 65 6a   .......w ww.matej
00 01 c0 0c   mitas.co m.......
69 52         ........ ....iR
```

Níže je struktura odpovědi (je třeba brát na vědomí, že se vždy vrací celý blok, u dotazu je to [hlavička + dotaz] a u odpovědi [hlavička + dotaz + odpověď].

- **Kopie** - *6B* - QNAME, QTYPE, QCLASS
- **TTL** - *2B* - Time-To-Live, doba platnosti
- **RDLENGTH** - *2B* - Délka nasledující datové části.
- **RDATA** - *1B * RDLENGTH* - Datová část

3 Implementace

Samotná aplikace se překládá a spouští z hlavního souboru *ipk-lookup.c*, který poté pracuje s dalšími připojenými soubory a společně tvoří páteř programu.

3.1 Obecný postup

První věc v programu je čtení argumentů, zde není nic zvláštního, je použita funkce *getopt*. Tyto parametry jsou uloženy (po kontrole validity) do struktury *ipk18_client_Args*, která je předána do hlavní programové části. Zde vytvoříme hlavní buffer programu (se velikosti $512B$, maximalní velikost pro DNS):

```
char buffer[DNS_SIZE] = {0, };
```

Tento buffer je postupne plněn daty, nejdříve vyplníme hlavičku, poté nastavíme příznaky a nakonec naplníme data (hostname či IP podle módu). Zde je třeba myslet na rozdílné bajtové pořadí u vícebajtových typů na serveru a na klientovi (funkce *htons* a *ntohs*).

Následně probíhá UDP komunikace. Nejdříve je třeba vytvořit socket, nastavit jeho timeout, zavolat funkce pro poslání našeho naplněného buffer (*sendto*) a poté přijmeme odpověď (*recvfrom*). Poté je komunikace ukončena a přechází se na čtení odpovědi.

Po příjetí zprávy přečteme příznaky hlavičky (*ipk18_dns_Header_read_opt*) a vyhodnotíme je (*ipk18_dns_Header_asses_opt*). V případě korektní odpovědi přečteme část s odpovědí, začneme **NAME** (použijeme metodu se skoky pro úsporu místa) a po kontrole **TYPE** a **CLASS** čteme data **RDATA** o délce **RDLENGHT**.

3.2 Zajímavé součásti

V této sekci bych rád vypíchl některé části implementace, kterým by čtenář měl věnovat zvláštní pozornost. Jelikož byla implementace provedena v C, je možné, že v některém vyšším jazyce (Python, Perl) by postup byl jednodušší, ale zde je možno vidět principy v praxi.

3.2.1 Práce s příznaky DNS hlavičky

Samotné příznaky jsou $1b$ až $4b$, což je ovšem s datovými typy jazyka C **nekompatibilní**, proto jsem se rozhodl všechny příznaky uložit do jednoho $16b$ čísla (typu *uint_16t*) Zápis probíhá ručně přes vložení hodnoty $0b0|0000|0|0|1|0|000|0000$, kde formát odpovídá (QR—Opcode—AA—TC—RD—RA—Z—Rcode), zde tedy nastavujeme pouze bit **RD - Recursion desired**, pro rekurzivní dotaz.

Ovšem pro čtení je třeba zvolit jiný přístup a to již práce s jednotlivými příznaky. Toto se děje ve funkci (*ipk18_dns_Header_read_opt*), kde probíhá maskování. Uvedu příklad pro příznak **AA**:

```
// opt je struktura pro priznaky
uint16_t mask_one_bit = 0x400;
opt->aa = (opts & mask_one_bit) >> 10;
```

Vymaskuji onen inkriminovaný bit (zde konkrétně na pozici 6) a posunu doprava o pozici vymaskovaných bitů.

3.2.2 Normalizace QNAME

QNAME se do DNS dotazu ukládá jako string, avšak má určité specifikum a to jest nahrazení teček počtem znaků za. Místo *www.matejmitas.com* vznikne *3www10matejmitas3com*. Důležité je pracovat s počtem znaků v binární (opravdu $0x03$, nikoliv $0x33$, což je znaková reprezentace v ASCII) formě.

Horší situace nastává u revezního překladu (IPv4 a IPv6 adresy). Nejen, že je třeba nahradit tečky, respektive dvojtečky, ale také otočit pořadí **oktetů** (8b) u IPv4 nebo **nibblů** (4b) u IPV6. Poslední krok je přidání přípony **.in-addr.arpa** u IPv4 nebo **.ip6.arpa**. Níže je demonstrace pro IPv4:

```
// vstupni tvar
211.31.55.23
// otocime oktety
23.55.31.211
// vymenime tecky za cisla, mezery
// pouze pro prehlednost
223 255 231 3211
// pridame koncovku
22325523132117in-addr4arpa
```

U IPv6 nám ještě komplikuje situaci shlukování nul (:: je znak pro shluk)

```
// vstupni tvar
2a02:2b88:1:4::34
// vystupni tvar
4.3.0.0.0.0.0.0.0.0.0.0.0.0.0.0.4.0.0.0.1.0.0.0.8.8.b.2.2.0.a.2.ip6.arpa
22325523132117in-addr4arpa
```

3.2.3 Realizace Timeout

Maximální udávané zpoždění se nastavuje parametrem **-t**, kde standartní hodnota je $5s$. Samotné zpozdění je realizováno přes funkci pro nastavení socketu *setsockopt*. Je volána s parametrem času, který je obsažen v:

```
struct timeval tm;
```

a obsahuje tyto položky:

```
// cteni casu z parametru
tm.tv_sec = args->timeout;
// vynulovani usec citace
tm.tv_usec = 0;
```

4 Funkčnost

Funkčnost byla otestována na mém osobním stroji (macOS), a také na serverech **merlin.fit.vutbr.cz** a **eva.fit.vutbr.cz**. Projekt byl napsán v souladu se standartem $C11$ a přeložen lokální verzí $GCC7.1$. Projekt je zatížen určitými nesrovnalostmí, jak je zmíněno v *Readme*.

Reference

BUSH, R. *Clarifications to the DNS Specification* [online]. 1997. [cit. 9. 4. 2018]. Dostupné z: `https://tools.ietf.org/html/rfc2181`.

FROM WIKIPEDIA, t. f. e. *Domain Name System* [online]. 2018. [cit. 9. 4. 2018]. Dostupné z: `https://en.wikipedia.org/wiki/Domain_Name_System`.

LIBOR DOSTALEK, A. K. *Velky pruuvodce protokoly TCP/IP a systemem DNS*. 5. vydani. Computer Press, 2012. ISBN 978-80-251-2236-5.

MOON, S. *DNS Query Code in C with Linux sockets* [online]. [cit. 9. 4. 2018]. Dostupné z: `https://www.binarytides.com/dns-query-code-in-c-with-linux-sockets/`.

RYSAVY, O. – RAB, J. *IPK - Sitova vrsta - 4. prednaska* [online]. 2018. [cit. 12. 3. 2018]. Dostupné z: `https://wis.fit.vutbr.cz/FIT/st/course-files-st.php?file=%2Fcourse%2FIPK-IT%2Flectures%2FIPK2017L-04-IPv4.pdf`.