VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Síťové aplikace a správa sítí

DNS Resolver

17. listopadu 2019

Tomáš Zálešák

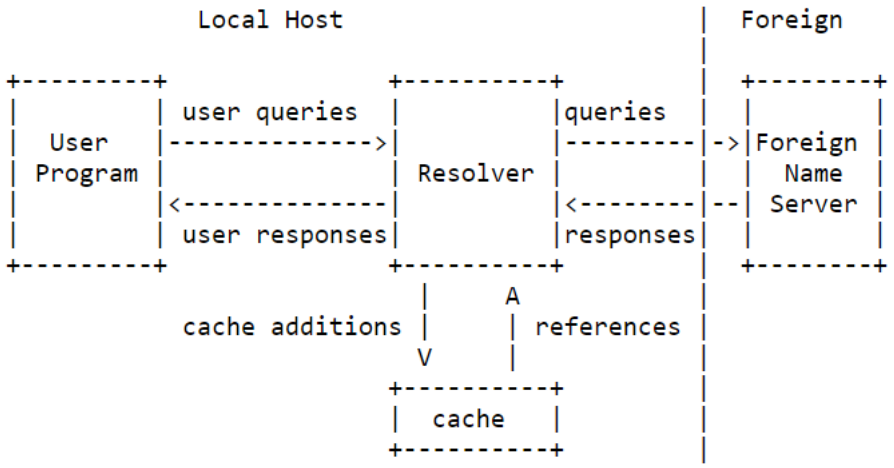
Obsah

1 Uvedení do problematiky

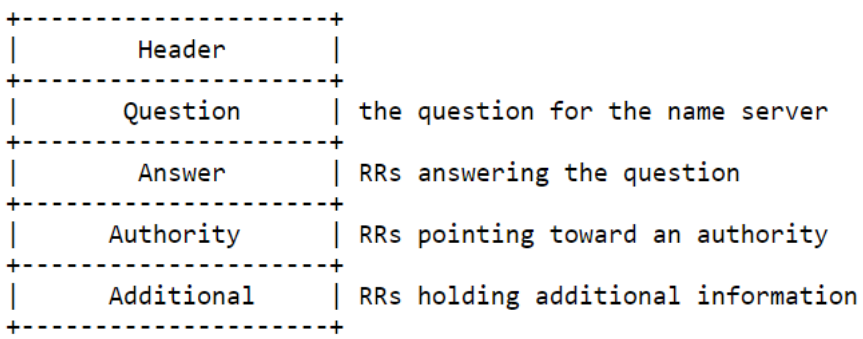
Domain Name System je systém doménových jmen, jenž umožňuje překlad domén na IP adresy. Díky němu si nemusíme pamatovat IP adresy webových stránek a počítač si sám zjistí, na jakou IP adresu má např. zaslat HTTP požadavek.

1.1 DNS dotaz

Počítač musí poslat DNS dotaz na DNS server, ten dotaz vyhodnotí a vrátí mu odpověď. Na unixových a unix-like systémech jsou DNS servery uvedeny v souboru /etc-resolv.conf. Dotazy jsou malé množství dat poslané pomocí protokolu UDP.



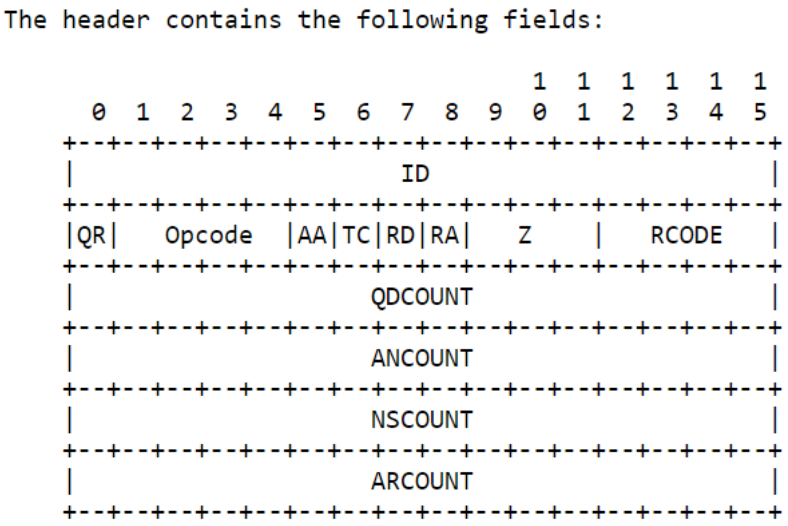
Celé dotazy/odpovědi vypadají následovně:



Skládájí se z 12 bajtů pro nastavení a samotného dotazu.
Nastavení dotazu - DNS HEADER:

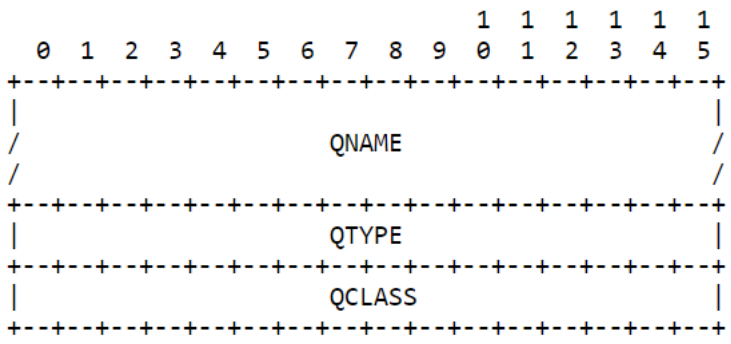
- 2 bajty identifikace transakce - použil jsem ID klientského procesu
- 2 bajty pro nastavení dotazu: QR (dotaz či odpověď), OPCODE (druh dotazu), AA (autoritativní odpověď), TC (zkráceno), RD (požadování rekurze), RA (dostupnost rekurze), Z (nepoužívá se), RCODE (kód odpovědi)
- 2 bajty specifikující počet dotazů, v našem případě 1
- 2 bajty specifikující počet odpovědí

- 2 bajty specifikující počet autoritativních záznamů
- 2 bajty specifikující počet dalších záznamů



Dotaz:

- název dotazu QNAME
- 2 bajty pro specifikaci typu dotazu, A/AAA/PTR/atd.
- 2 bajty pro specifikaci třídy dotazu, používáme třídu Internet

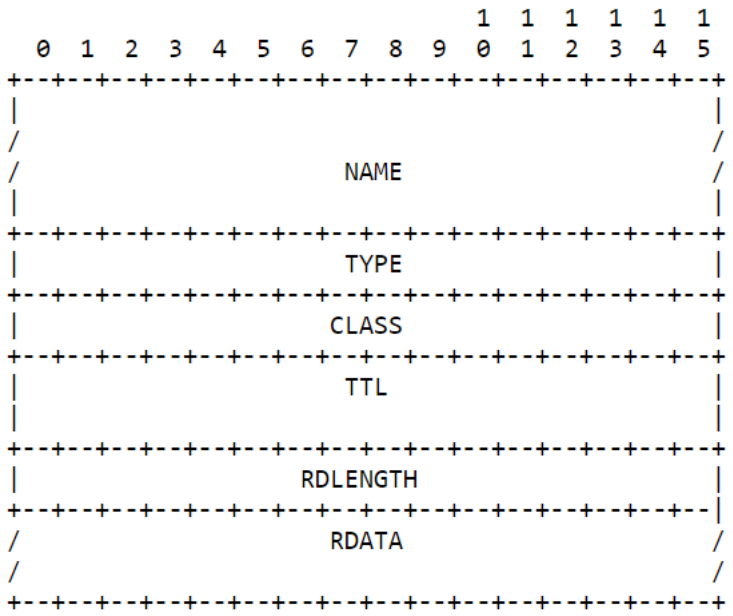


1.2 DNS odpověď

Odpověď nejprve obsahuje daný dotaz a za ním následuje obsah odpovědi. Následující odpovědi mají stejnou strukturu, mají proměnlivou délku a následují hned za sebou.

- název odpovědi NAME, proměnlivá délka, může být použita komprese
- 2 bajty pro specifikaci typu dotazu, A/AAA/PTR/CNAME/atd.
- 2 bajty pro specifikaci třídy dotazu, používáme třídu Internet
- 4 bajty obsahující TTL
- 2 bajty udávající délku výsledného jména v bajtech

- výsledné jméno



2 Návrh aplikace

Úkolem je napsat program dns, který pošle dotaz na DNS server a vypíše v čitelné podobě odpověď. Je uvažována pouze komunikace pomocí UDP.

2.1 Návod na použití

`dns [-r] [-x] [-6] -s server [-p port] adresa`

- -r: požadována rekurze (Recursion Desired = 1), jinak bez rekurze
- -x: reverzní dotaz místo přímého
- -6: dotaz typu AAAA místo výchozího A
- -s: IP adresa nebo doménové jméno serveru, kam se má zaslat dotaz
- -p port: číslo portu, na který se má poslat dotaz, výchozí 53
- adresa: dotazovaná adresa
- -h nápověda

Program podporuje záznamy typu A, AAAA, PTR, CNAME.

2.2 Implementace

Nejprve se program vypořádá s argumenty s pomocí funkce `getopt()`. Pak naplní strukturu DNS HEADER, která obsahuje nastavení dotazu. Pokud je požadována rekurze, tak se nastaví položka `dns_header.rd_=_1;`. Volbu IPv4 dotazu nastavíme v `query_type = htons(1); // A type`, IPv6 `query_type = htons(28); // AAAA type`. Pokud požadujeme reverzní dotaz, tak se daná položka

nastaví `query_type = htons(12); // PTR type`. Jsou použity funkce pro převod čísla do síťové podoby a nazpět: `htons, ntohs` atd.

Program vyplní jméno dotazu:

- v případě nerekurzivního dotazu převede např. www.google.com na 3www6google3com pomocí funkce `change_hostname_to_dns_query_name`
- v případě rekurzivního dotazu IPv4: např. 8.8.4.4 na 4.4.8.8.in-addr.arpa, IPv6: 2001:db8::567:89ab na b.a.9.8.7.6.5.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa a následně taky použije funkci `change_hostname_to_dns_query_name` pro převod teček na dané čísla

Vytvoří se socket data se pošlou pomocí funkce `sendto()`. Nastaví se time out pro příjem datagramu, aby se nečekalo do nekonečna:
`setsockopt(socket_fd, SOL_SOCKET, SO_RCVTIMEO, (char *) &timeout, sizeof timeout);`

Pro příjem datagramu vytvoříme buffer o velikosti 65536 bajtů, což je maximální velikost datagramu UDP. Přijmeme datagram pomocí funkce `recvfrom()`.

Přiřadíme ukazateli typu `struct DNS_HEADER` adresu odpovědi, což nám umožní přehledný přístup k položkám odpovědi.

Nejprve zkontroluji návratový kód RCODE v odpovědi.

- 0: OK
- 1: Format error
- 2: Server failure
- 3: Name Error
- 4: Not Implemented
- 5: Refused

Vypíšu info o daném dotazu: Authoritative Ano/Ne - hodnota AA z DNS Headeru, Recursive Ano/Ne - obě hodnoty RD a RA z DNS Headeru musí být 1, Truncated Ano/Ne - hodnota TC z DNS Headeru.

Vytvořím si pomocný ukazatel `reader`, který budu postupně zvětšovat při procházení odpovědi. Ukazatel ukazuje na poslední nepřečtený bajt v odpovědi.

Všechny DNS odpovědi mají podobnou strukturu, takže projdu postupně všechny typy odpovědí: answer, authority, a additional sekce.

Nejprve je zde jméno. Jedná se o posloupnost znaků ukončených nulovou hodnotou nebo o pointer na místo v paměti, kde se jméno vyskytuje. Jméno přečtu pomocí funkce `read_raw_name` a vrátí mi počet bajtů, které mám přeskočit pro další čtení. Následně vypíšu Type, Class, TTL, RDLENGTH, a RDATA, které obsahují samotnou odpověď. Může se jednat o doménové jméno v případě PTR dotazu nebo o IPv4/IPv6 adresu v případě A/AAAA dotazu.

Stejně to je provedeno pro následující typy odpovědí Authority a Additional.

3 Literatura

Podrobnosti jsou dostupné na následujících odkazech, ze kterých aplikace vychází.

`https://tools.ietf.org/html/rfc1035`
`https://tools.ietf.org/html/rfc3596`