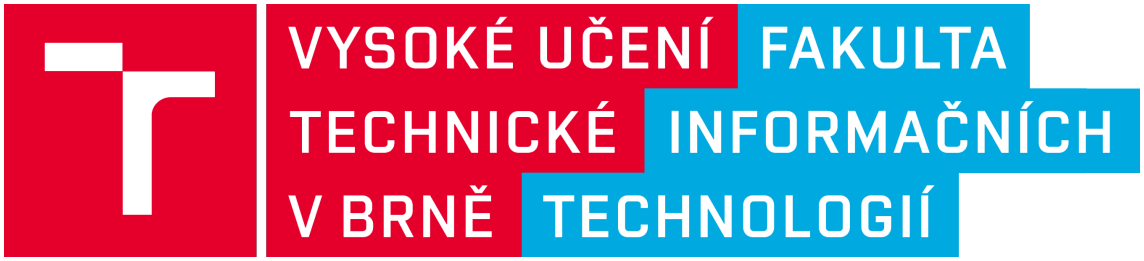


ISA - TECHNICKÁ DOKUMENTÁCIA K PROJEKTU DNS RESOLVER

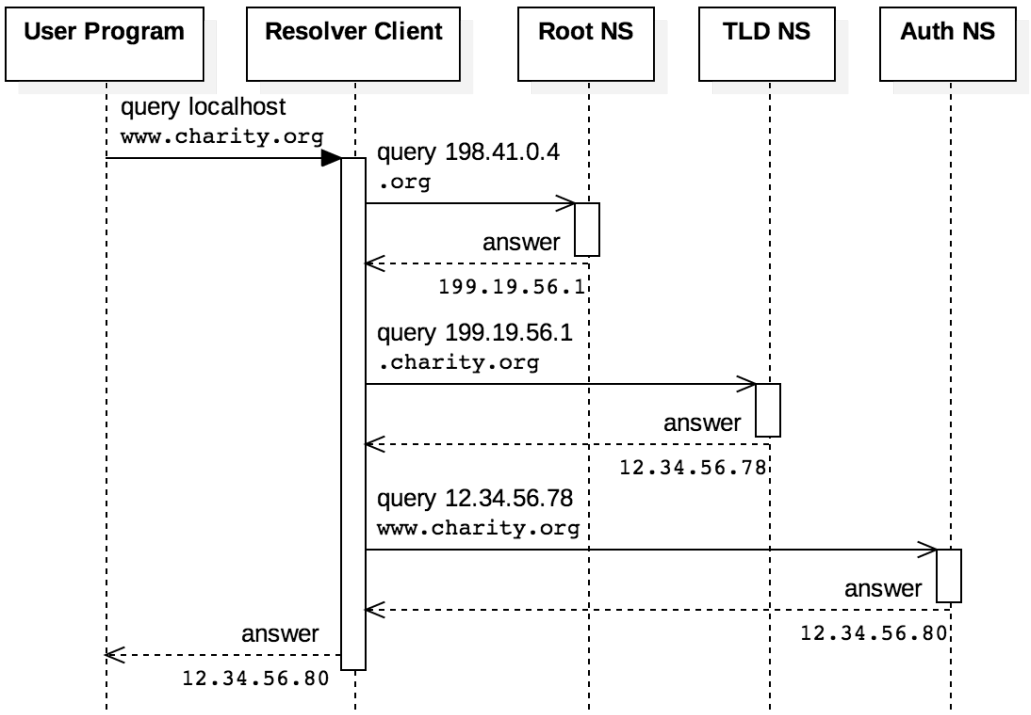
2023/2024

Patrik Gáfrik (xgafri00)

19. novembra 2023

1 Úvod

Základným predpokladom úspešnej komunikácie v rámci internetu je identifikácia zariadenia s ktorým chceme komunikovať. Pre identifikáciu koncového zariadenia sa využívajú IP adresy, ktoré sú pre človeka ťažko zapamätateľné. Pokiaľ chceme komunikovať napríklad s webovým serverom FIT VUT, nechceme si pamätať jeho IP adresu, chceme sa s ním spojiť cez názov domény *www.fit.vutbr.cz*. Preto by sme potrebovali zoznam doménových mien namapovaných na príslušné IP adresy. Systém DNS funguje ako distribuovaná databáza rôznych typov záznamov, ktoré obsahujú informácie o doménach vrátane IP adresy, ktorá k nim patrí a je základom pre správne fungovanie internetu. Po zadaní názvu domény v prehliadači prebehne odoslanie dotazu na nakonfigurovaný DNS resolver, ktorý sa pokúsi požadovaný záznam nájsť v pamäti cache, pokiaľ ho nenájde prebehne tzv. lookup - resolver odošle dotaz vybranému root serveru, ktorého IP adresu pozná, root server zašle odpoveď v podobe záznamu, ktorý obsahuje IP adresu TLD serveru. Resolver potom pošle dotaz na TLD server, ktorý odpovie IP adresou autoritatívneho serveru pre požadovanú doménu. Autoritatívny server obsahuje všetky informácie o doméne a typicky vráti resolveru záznam obsahujúci IP adresu patriacu požadovanej doméne.



Obr. 1: Sekvenčný diagram DNS rezolúcie

[1] Cieľom tohoto projektu je navrhnúť a implementovať vlastný DNS resolver, ktorý pošle paket s dotazom na DNS server, spracuje paket s odpoveďou, ktorú vo vhodnej forme vypíše na štandardný výstup. Pri návrhu je potrebné vyriešiť niekoľko problémov. Najprv potrebujeme vyriešiť nadviazanie komunikácie s DNS serverom, ktorého doménové meno/IP adresu zadá užívateľ pri spustení programu. Potom musíme vytvoriť DNS paket vo formáte definovanom RFC 1035 a poslať ho na DNS server. Nakoniec treba implementovať analýzu odpovedi od DNS serveru.

[1] Obr. 1 ref. Kirkpatrick [2023]

2 Nadviazanie komunikácie s DNS serverom

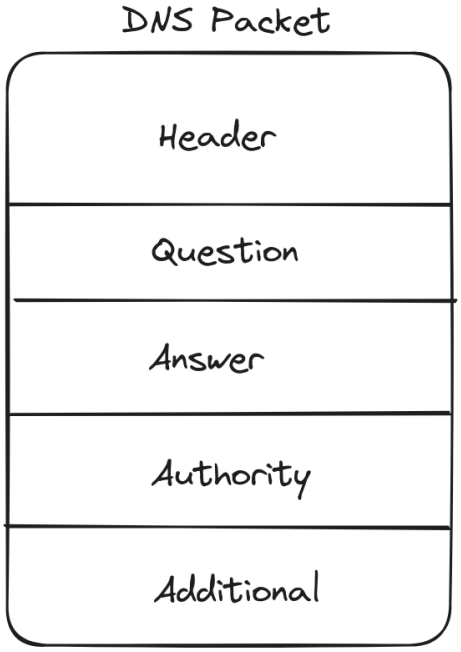
Najskôr potrebujeme previesť doménové meno/IP adresu DNS serveru a číslo portu na zoznam štruktúr addrinfo, ktoré sú potrebné pre vytvorenie socketu a spojenie so serverom. Získanie zoznamu štruktúr addrinfo pre požadovanú doménu/IP adresu zabezpečuje funkcia getaddrinfo(). Nasleduje cyklus, ktorý pre každú získanú štruktúru vytvorí UDP socket pomocou funkcie socket() a tento socket sa pošle ako parameter funkcie connect(), ktorá nastaví defaultnú adresu, s ktorou bude prebiehať UDP komunikácia. Pokiaľ funkcia connect() vráti návratovú hodnotu 0, tak nadviazanie komunikácie prebehlo úspešne a preruší sa cyklus. Pokiaľ connect() vráti -1 (neúspech), cyklus pokračuje a opakuje proces pre nasledujúcu štruktúru v zozname. Pokiaľ cyklus prejde celý zoznam bez úspešného volania connect(), nadviazanie komunikácie nebolo úspešné a prebehne výpis chybovej hlášky a ukončenie programu s hodnotou 1. Kerrisk [2023]

3 Štruktúra DNS paketu

Štruktúra DNS paketu (message) je definovaná štandardom RFC 1035 a obsahuje niekoľko sekcií.

Sekcie DNS paketu:

- **Header** obsahuje základné informácie o pakete (ID, flags, počet záznamov v každej sekcií paketu)
- **Question** sekcia obsahujúca názov, typ a triedu dotazovaného záznamu
- **Answer** sekcia obsahujúca odpoveď od serveru vo forme záznamov
- **Authority** sekcia obsahujúca záznamy o autoritatívnych serveroch
- **Additional** sekcia obsahujúca doplňujúce záznamy



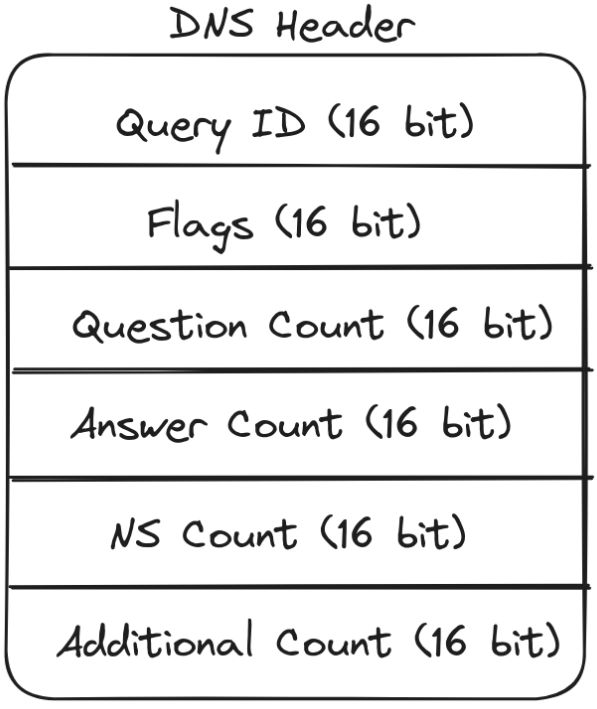
Obr. 2: Štruktúra DNS paketu

4 Konštrukcia DNS paketu

Konštrukcia a odoslanie DNS paketu je implementované vo funkcií **constructDNSPacket**. Ako prvé je potrebné vytvoriť **Header**. Štruktúra DNS headeru je definovaná štandardom RFC 1035 a obsahuje niekoľko sekcií. V implementácií programu je DNS header reprezentovaný pomocou **struct DNS_header**.

Sekcie DNS headeru:

- **Query ID** je náhodne vygenerovaná 16-bitová hodnota, ktorá jednoznačne identifikuje DNS paket. Generovanie je implementované s využitím funkcie srand().
- **Flags** je 16-bitová hodnota kde každý bit alebo skupina bitov má určitý význam. Bit ktorý nás bude zaujímať je bit RD (recursion desired) a ten nastavujem na hodnotu 1 v prípade, že program bol spustený s prepínačom -r. Inak je hodnota flags nastavená na 0.
- **Question Count** je 16-bitová hodnota udávajúca počet dotazov. Pri každom spustení programu sa predpokladá jeden dotaz a preto je táto hodnota nastavená na 1.
- **Answer Count** je 16-bitová hodnota udávajúca počet záznamov v Answer sekcií paketu. Pri vytvorení headeru pre dotaz ju explicitne nenastavujem, v headeri pre odpoveď odpovedá počtu záznamov.
- **NS Count** je 16-bitová hodnota udávajúca počet záznamov v Authority sekcií paketu. Pri vytvorení headeru pre dotaz ju explicitne nenastavujem, v headeri pre odpoveď odpovedá počtu záznamov.
- **Additional Count** je 16-bitová hodnota udávajúca počet záznamov v Additional sekcií paketu. Pri vytvorení headeru pre dotaz ju explicitne nenastavujem, v headeri pre odpoveď odpovedá počtu záznamov.

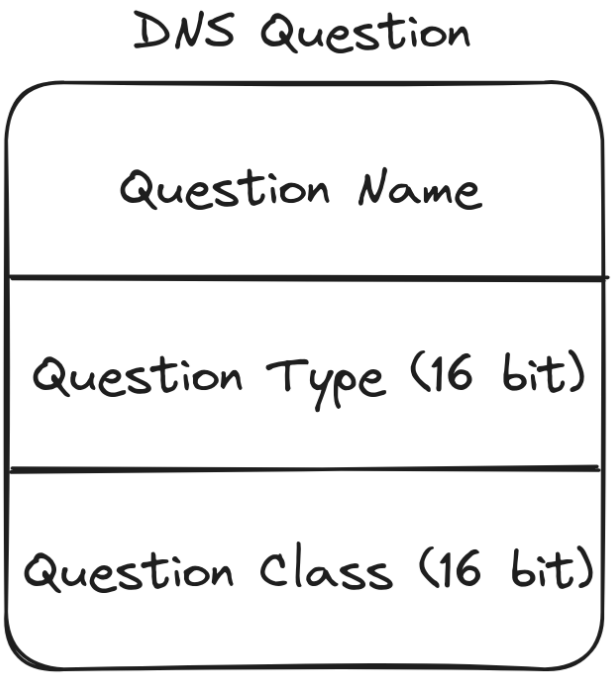


Obr. 3: Štruktúra DNS headeru

Ďalej je potrebné vytvoriť **Question** sekciu paketu. Štruktúra Question sekcie je definovaná v RFC 1035 a obsahuje tri sekcie.V implementácií programu je Question sekcia reprezentovaná pomocou **struct DNS_question**

Štruktúra Question sekcie:

- **Question Name** je variable length hodnota a predstavuje dotazované doménové meno alebo IP adresu (v prípade dotazu na reverzný záznam). Táto hodnota musí byť vo formáte, kde na začiatku každého labelu domény musí byť číslo vyjadrujúce dĺžku labelu v bytoch napr. *www.fit.vutbr.cz* na musí previesť na *3www3fit5vutbr2cz.*
- **Question Type** je 16-bitová hodnota a predstavuje typ dotazovaného záznamu. Pri spustení programu sa implicitne predpokladá dotaz na A záznam a preto sa hodnota nastaví na 1. V prípade spustenia programu s prepínačom -6, ktorý predstavuje dotaz na AAAA záznam sa hodnota nastaví na 28.
- **Question Class** je 16-bitová hodnota a predstavuje triedu dotazovaného záznamu. Predpokladá sa záznam triedy IN a preto sa vždy nastaví na hodnotu 1.



Obr. 4: Štruktúra Question sekcie

Po vytvorení DNS Headeru a Question sekcie je potrebné tieto dve štruktúry spojiť do jedného paketu a tento paket poslať na server. Štruktúra paketu je pole bytov. Ukazateľ na toto pole je predaný ako parameter funkcie **send()** ktorá zabezpečí poslanie paketu.

5 Spracovanie DNS odpovedi

Získanie a spracovanie DNS odpovedi je implementované vo funkcií **getDNSAnswer**. Najskôr je potrebné vytvoriť pole o veľkosti 512 bytov (maximálna dĺžka DNS paketu s odpoveďou). Potom nasleduje volanie funkcie **recvfrom()**, ktorá paket s odpoveďou skopíruje do vytvoreného pola. Toto pole obsahuje paket, ktorý sme získali zo strany serveru a teraz už môžeme analyzovať jednotlivé sekcie paketu a vypísať ich na štandardný výstup.

5.1 Analýza Headeru

Ako prvé je implementovaná analýza headeru v pakete s odpoveďou. Na začiatku máme ukazateľ na struct DNS_header, ktorý je inicializovaný na začiatok paketu s odpoveďou - zaberá vždy prvých 12 bytov paketu. V headeri nás budú zaujímať bity AA, RD, TC a skupina bitov RCODE v sekcií Flags. V prípade získania validnej odpovedi bude RCODE obsahovať hodnotu 0. V prípade že táto hodnota nie je 0, program vypíše chybovú hlášku a ukončí sa s hodnotou 1. V prípade, že odpoveď bola získaná od autoritatívneho serveru, bit AA bude nastavený na hodnotu 1 v opačnom prípade na hodnotu 0. Rovnako to bude v prípade bitov RD a TC, ktoré sú nastavené na hodnotu 1 v prípade, že odpoveď bola získaná rekurzívne, alebo bola skrátená. Na základe týchto hodnôť program vypíše hlášku na štandardný výstup. Ďalej nás bude zaujímať počet záznamov v jednotlivých sekciách paketu. Tieto hodnoty sa tiež objavia na výstupe.

5.2 Analýza Question Section

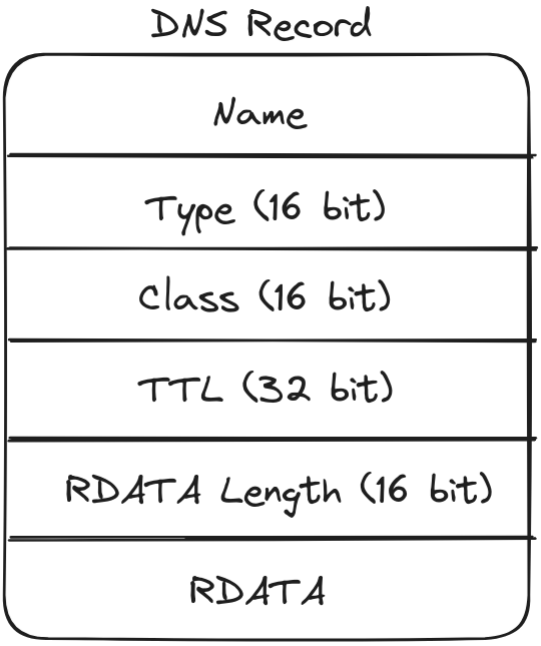
Question Section obsahuje dotazovaný záznam a je identická s tou v pakete s dotazom. Program skontroluje typ a triedu záznamu a spolu s dotazovaným menom ich vypíše na štandardný výstup.

5.3 Analýza Answer Section

Pri analýze Answer Section je dôležité poznať štruktúru DNS záznamov, ktoré sú definované v RFC 1035. Každý DNS záznam obsahuje niekoľko sekcií.

Sekcie DNS záznamu:

- **Name** je variable length hodnota a reprezentuje názov záznamu
- **Type** je 16-bitová hodnota a reprezentuje typ záznamu
- **Class** je 16-bitová hodnota a reprezentuje triedu záznamu
- **TTL** je 32-bitová hodnota a reprezentuje hodnotu Time To Live čo je čas v sekundách počas ktorého je záznam uložený v pamäti cache resolveru
- **RDATA Length** je 16-bitová hodnota a reprezentuje dĺžku hodnoty RDATA v bytoch
- **RDATA** je variable length hodnota a reprezentuje dáta záznamu



Obr. 5: Štruktúra DNS záznamu

V implementácií programu je DNS záznam reprezentovaný pomocou **struct DNS_record**. Pre zachovanie jednoduchosti implementácie som sa rozhodol do tejto štruktúry nezahrnúť variable length sekcie DNS záznamu, takže neobsahuje Name a RDATA a má vždy pevnú dĺžku 10 bytov (od Type po RDATA Length). Na začiatku analýzy Answer Section je dôležité nastaviť tri ukazatele: ukazateľ **name_start** ktorý ukazuje na prvý byte sekcie Name, ukazateľ **records**, ktorý ukazuje na začiatok záznamu tj. na Type a zaberá 10 bytov, a ukazateľ **rdata**, ktorý ukazuje na prvý byte RDATA sekcie.

Analýza prebieha vo funkcií **parseAnswerSection** ako cyklus, ktorý sa vykoná x krát, kde x je počet záznamov v Answer Section (hodnota získaná z headeru). Vo vnútri cyklu sa na základe typu záznamu vyberie rutina, ktorá sa vykoná. Pre účely tohoto projektu predpokladáme tri možné typy záznamov v Answer Section: **A, CNAME, AAAA**. Postup analýzy jednotlivých záznamov spočíva v tom, že sa vypíše názov záznamu pomocou funkcie **printName()** (popis funkcie v podkapitole 5.6), typ a trieda záznamu a RDATA. Na konci analýzy sa nastavia ukazatele na nasledujúci záznam.

5.4 Analýza Authority Section

Analýza Authority Section je implementovaná vo funkcií **parseAuthoritySection** a funguje na rovnakom princípe ako analýza Answer Section s rozdielom, že predpokladáme len jeden typ záznamov - **NS** záznam.

5.5 Analýza Additional Section

Analýza Additional Section je implementovaná vo funkcií **parseAdditionalSection** a funguje na rovnakom princípe ako analýza predchádzajúcich sekcií, ale predpokladajú sa dva typy záznamov: **A** a **AAAA**.

5.6 Riešenie kompresie v DNS záznamoch

Aby sa predišlo vzniku redundancie, sekcie záznamov Name a RDATA neobsahujú vždy kompletný reťazec dát. Pokiaľ sa už niektoré dáta vyskytli v predchádzajúcich záznamoch využije sa kompresia - dvojica bytov reprezentujúca ukazateľ na konkrétny byte v pakete, kde začína reťazec s týmito dátami. Tieto dva byty majú vždy prvé dva bity nastavené na 1. Zvyšných 14 bitov reprezentuje hodnotu, ktorá udáva offset od začiatku paketu (index na ktorom sa nachádzajú požadované dáta). Takto je možné rozlíšiť kompresiu od dát v štandardnom formáte. Riešenie kompresie je implementované v rámci funkcie **printName**. Táto funkcia dostáva na vstup ukazateľ na prvý byte sekcie Name alebo RDATA a ukazateľ na prvý byte paketu. Následne prejde byte po byte celú sekciu a tieto byty vypíše na štandardný výstup. Pokiaľ narazí na byte v ktorom sú prvé dva bity nastavené na 1, vypočíta offset zo zvyšných 14 bitov a rekurzívne sa zavolá s predaným ukazateľom na miesto v pakete kde začínajú compressed dáta.

6 Testovanie aplikácie

Aplikácia bola otestovaná Python skriptom **tester.py**. Test je založený na porovnaní výstupu programu dig s výstupom aplikácie. Najskôr sa definujú regulárne výrazy pre adresy IPv4 a IPv6. Potom sa spustí program dig a program dns s rovnakou adresou DNS serveru a dotazovanou adresou a výstupy oboch programov sa vypíšu na štandardný vystup. Pomocou regulárnych výrazov sa z oboch výstupov vyfiltrujú adresy IPv4 a IPv6 a uložia sa do dátových štruktúr tuple. Potom sa tieto štruktúry porovnajú a pokiaľ sa zhodujú test bol úspešný a vypíše sa hláška, že boli nájdené zhodné IP adresy.

Literatúra

Computerphile. How dns works, 2020. URL https://www.youtube.com/watch?v=uOfonONtIuk&ab_channel=Computerphile.

Michael Kerrisk. *getaddrinfo - resolve domain name or IP address to IPv4 or IPv6 address*. Linux Manual Pages, 2023. URL https://man7.org/linux/man-pages/man3/getaddrinfo.3.html.

Michael S. Kirkpatrick. Udp socket programming: Dns, 2023. URL https://w3.cs.jmu.edu/kirkpams/OpenCSF/Books/csf/html/UDPSockets.html.

P. Mockapetris. Domain names - implementation and specification. Technical Report 1035, Internet Engineering Task Force, 1987. URL https://www.ietf.org/rfc/rfc1035.txt.