



Projektová dokumentace

DNS resolver

Síťové aplikace a správa sítí

Obsah

| | | |
|----------|-----------------------------|----------|
| 1 | Úvod | 1 |
| 2 | Teoretický základ | 1 |
| 3 | Návrh a implementace | 2 |
| 4 | Překládový systém | 4 |
| 5 | Návod na použití | 4 |
| 6 | Přílohy | 5 |
| 7 | Reference | 5 |

1 Úvod

Cílem projektu je vytvořit aplikaci DNS resolver v jazyce C/C++. Program naslouchá na vybraném portu a filtruje DNS dotazy. DNS resolver podporuje dotazy typu A zaslané pomocí UDP protokolu transportní vrstvy. Pro ostatní dotazy je zaslán DNS paket informující o problému. Resolver načte z dodaného souboru domény určené k vyfiltrování. Pokud je v dotazu požadován překlad doménového jména, či jeho subdomény, který je uveden ve filtru, pak je tento dotaz zahozen. V opačném případě je dotaz dále přeposlán na specifikovaný DNS server a zpráva z něj je vrácena původnímu tazateli.

2 Teoretický základ

DNS je protokol aplikační vrstvy modelu TCP/IP. Jedním z využití DNS je překlad doménových jmen na IP adresy, který je pro tento projekt stěžejní. Při DNS komunikaci je na transportní vrstvě využívána komunikace pomocí UDP datagramů.

2.1 DNS rezoluce

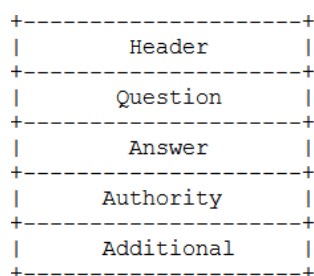
Rezoluce je proces hledání odpovědi v systému DNS. DNS prohledává strom, v kterém jsou uloženy záznamy obsahující doménové jméno a k němu odpovídající adresu. Rezoluce může být buď rekurzivní, nebo iterativní.

V případě rekurzivního dotazu je tazateli ze serveru vrácena odpověď a dohledávání uloženého záznamu je plně na DNS serveru. Ten, pokud nemá uložen požadovaný záznam, se musí na tento záznam sám doptat jiných DNS serverů.

V případě iterativního dotazu vrací DNS server buď odpověď na dotaz – přeložené doménové jméno, nebo adresu jiného DNS serveru, který odpověď zná, nebo jí je „blíže“.

2.2 DNS paket

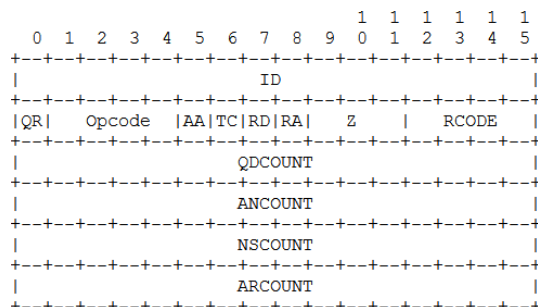
DNS paket se skládá z 5 částí uvedených v obrázku 1. V rámci projektu jsou důležité první 3 části – hlavička, dotaz a odpověď.



Obrázek 1: Struktura DNS paketu

2.2.1 Hlavička

Hlavička obsahuje informace o paketu – jeho identifikační číslo, příznaky, počty informací v dalších částech nebo návratové a operační kódy. Struktura hlavičky DNS paketu je popsána na obrázku 2.



Obrázek 2: Struktura hlavičky DNS paketu

Každá položka hlavičky udává určité informace o celém paketu:

- ID: identifikátor paketu.
- QR: příznak, který je nastaven do hodnoty 1 v případě, že paket představuje odpověď, jinak obsahuje hodnotu 0.
- OpCode: operační kód specifikující typ paketu, v rámci projektu není podstatný.
- AA: příznak nastavený do hodnoty 1, pokud odpověď pochází z autoritativního DNS serveru
- TC: Detekce zkrácené odpovědi, v rámci projektu není jeho význam důležitý.
- RD: bit je nastaven do hodnoty 1, pokud klient vyžaduje použití rekurzivní rezoluce.
- RA: bit nastavuje DNS server, pokud vyhoví požadavku klienta na rekurzivní rezoluci.
- Z: bity rezervované pro budoucí využití, jsou vždy nastaveny na hodnotu 0.
- RCode: kód specifikující chyby. Bližší popis a jejich využití v projektu je uveden v tabulce 2.

Dále hlavička DNS paketu obsahuje počty dotazů a počty v dalších sekcích paketu. DNS resolver kontroluje počet dotazů uvedený v QDCOUNT části DNS hlavičky.

2.2.2 Dotaz

Tělo DNS zprávy se skládá z dotazů, ke kterým je uveden typ a třída. DNS resolver podporuje pouze 1 dotaz v těle zprávy, pro jiný počet zpráv je zaslán chybový DNS paket s kódem 4 – požadovaná služba není implementována.

3 Návrh a implementace

Aplikace je rozdělena do jednotlivých souborů obsahující implementaci tříd, které implementují výsledný systém. Zdrojové soubory se nachází v adresáři src.

Po spuštění programu jsou zpracovány argumenty předané z příkazové řádky. Poté je spuštěn server, který vytváří filtr a čeká na DNS pakety na uvedeném portu.

3.1 Parsování vstupních argumentů

Parsování vstupních argumentů předané programu z příkazového řádku zajišťuje třída `argparse`. Při vytváření instance třídy jsou konstruktoru předány veškeré parametry z příkazové řádky. Po zpracování je možné se pomocí k tomu určených metod dotazovat na potřebné údaje – IP adresu dns serveru, jméno souboru obsahující filtr a případně číslo portu.

V případě, že je programu předáno doménové jméno dns serveru, pak je jeho IP adresa dohledána pomocí funkce `gethostbyname`.

Neplatné vstupní argumenty jsou ignorovány. V případě, že je programu zadáno neplatné číslo portu, pak program využije výchozí hodnotu 53. Program končí v chybovém stavu, pokud nezíská informaci o IP adrese dns serveru a název souboru obsahující filtr.

Aplikace může být přepnuta do módu, v němž je uživatel informován o jednotlivých krocích programu zprávami směřovanými na standardní výstup. Mód může být aktivován využitím přepínače `-v`. Výpis informací je implementován třídou `verbose`.

3.2 Zpracování souboru filtru

Zpracování vstupního souboru obsahující filtr je implementováno třídou `filter`. Při vytváření objektu třídy je konstruktoru předán název souboru obsahující filtr. Konstruktor ukládá jednotlivé řádky obsahující doménová jména k vyfiltrování do vektoru textových řetězců.

Při zpracovávání souboru jsou vynechány prázdné řádky a řádkové komentáře začínající symbolem `#`. Před zpracováváním každého řádku je prvně provedeno odebrání bílých znaků na začátku a na konci daného řádku.

V případě, že soubor neexistuje, je program ukončen a na standardní chybový výstup je uvedena informace o nastalé chybě.

3.3 Serverová část

Zajištění síťové komunikace s klientem zasílající DNS dotazy je implementováno pomocí třídy `server`. Konstruktoru objektu třídy `server` jsou předány hodnoty čísla portu, na němž má server čekat na zprávy, ip adresa dns serveru, na něž mají být směřovány dotazy, které nebudou vyfiltrovány, a jméno souboru obsahující domény určené k vyfiltrování.

Po extrakci dat z filtru je vytvořen soket serveru, který je napojen na daný port. V případě problémů s vytvářením soketu a jeho připojením na port je program ukončen s příslušným chybovým kódem a na standardní chybový výstup je vypsána hláška vysvětlující důvod ukončení programu.

Po vytvoření instance třídy `server` je možné spustit pomocí metody `run_server` samotný server, který čeká na dns dotazy, které buď vyfiltruje a zašle paket s příslušným návratovým kódem, nebo paket přeposílá dál.

V případě chyby na straně serveru je klientovi zaslán DNS paket informující o selhání serveru. Pokud server detekuje při příjmu, zpracovávání, nebo odesílání paketu problém, pak ukončuje zpracovávání daného paketu a čeká na příchod dalšího. Pokud je server spuštěn pomocí metody `run_server`, pak je jej možné ukončit pouze vysláním příslušného signálu, např. `SIGINT`.

3.4 Klientská část

Komunikaci s určeným DNS serverem je implementována pomocí třídy `client`. Při instanciaci je vytvořen soket, který je připojen na port 53, jenž je primárně určen pro DNS komunikaci.

Pomocí soketu je možné zaslat s využitím funkce `send_data` DNS paket dále DNS serveru, který tento přeloží a zašle zpět. Přijatý paket je poté vrácen tazateli, tedy objektu třídy `server`.

Metoda zasílající data vrací DNS odpověď a délku vráceného paketu. Soket je možné zavřít pomocí metody `close_socket`. Při zpracovávání jednoho DNS paketu je postupně otevřen soket, zaslány data a soket poté uzavřen. Při zpracovávání dalšího paketu je tento proces opakován.

3.5 Módy aplikace

Aplikace může pracovat ve dvou různých režimech. Ve výchozím nastavení pracuje server v tichém režimu – na standardní výstup nejsou vypisovány žádné informace, případné chybové hlášky jsou vypisovány na standardní chybový výstup. Výpisy jsou realizovány pomocí k tomu určených třídních metod třídy `verbose`.

Hlášení průběhu překladu jsou uvozena znakem `*` a je možné je tak odlišit od hlášek směřovaných na standardní chybový výstup.

3.6 Chybové stavy

Pokud se program dostane do chybového stavu, pak tento stav řeší třídní metody třídy `err_handler`. Třída informuje uživatele o problému. Pokud nelze dále pokračovat a chybový stav ošetřit, pak je program ukončen s chybovým návratovým kódem. Chybové kódy jsou uvedeny v tabulce 1.

Chybová hlášení jsou vypisována na standardní chybový výpis. Pro jednodušší odlišení od výpisu na standardní výstup jsou tato hlášení uvozena znaky `***`.

3.7 Pomocné funkce

V rámci projektu jsou využívány další pomocné metody implementované třídou `func`. Pomocí regulárních výrazů jsou validovány IP adresy předané jako argument z příkazové řádky. Je tak možné detekovat chybu dříve, než je tato adresa využita pro vytvoření socketu připojující se na DNS server.

Další funkce umožňují zkontrolovat, zda uvedený textový řetězec obsahuje pouze čísla – je možná kontrola čísla portu. Při zpracovávání souboru obsahující filtr jsou pak využívány metody odebírající bílé znaky ze začátku a konce textového řetězce.

4 Překladačový systém

Překladačový systém je vytvořen pomocí utility `make`. Zdrojové soubory je možné přeložit pomocí připojeného souboru `Makefile`. Pro překlad je využíván kompilátor `gcc`. Po překladu je v kořenovém adresáři projektu umístěn spustitelný soubor `dns`.

Pomocí příkazu `make test` je možné spustit testy, které byly využity při testování aplikace.

5 Návod na použití

Po stažení zdrojových souborů proved te překlad pomocí utility `make` s využitím přiloženého souboru `Makefile`. Po přeložení je možné spustit program `dns`.

Program je možné spustit s následujícími argumenty:

- `-s <server>`: Pomocí parametru `s` je možné specifikovat DNS server, na který budou dále zasílány DNS dotazy, které nebudou vyfiltrovány a budou vyhovovat všem požadavkům. Server je možné určit pomocí jeho doménového jména, případně pomocí IP adresy.
- `-f <soubor>`: Pomocí tohoto parametru je možné vybrat soubor obsahující domény určené k vyfiltrování.
- `-p <port>`: Pomocí parametru je možné určit číslo portu, na kterém bude program očekávat příchozí DNS pakety. Parametr není povinný.
- `-v`: Program je přepnut do módu, v kterém jsou vypisovány informace o stavu programu. Parametr není povinný.

V případě, že bude uveden jiný parametr, program jej bude ignorovat, případně bude vypsána nápověda.

6 Přílohy

| Chybový kód | Popis chyby |
|-------------|---|
| 0 | Program byl ukončen bez detekování chyby |
| 1 | Problém při zpracovávání vstupních argumentů |
| 2 | Formát IP adresy je neplatný, nebo nebyla získána IP adresa |
| 3 | Nedefinovaná hodnota portu, program využije výchozí port 53 |
| 4 | Program nenalezl uvedený soubor s filtrem |
| 5 | Špatná konstrukce souboru s filtrem |
| 6 | Chyba při otvírání soketu |
| 7 | Chyba při připojení soketu na port |
| 8 | Chyba příchozích dat |
| 9 | Chyba zaslání dat |
| 10 | Chyba zaslání dat – byla zaslána pouze část dat |

Tabulka 1: Návrátové kódy programu

| Návrátový kód | Význam kódu |
|---------------|--|
| 0 | Paket byl zpracován bez chyb |
| 1 | Chyba formátu, v rámci projektu jej může nastavit pouze DNS server provádějící překlad |
| 2 | Chyba na straně serveru, v rámci projektu nastaven v případě, že DNS resolver detekuje interní problém |
| 3 | Chyba doménového jména, v projektu jej může nastavit pouze DNS server provádějící překlad |
| 4 | Požadovaná služba není implementována, v rámci projektu nastaven, pokud je přijat jiný typ než A |
| 5 | Požadovaná služba byla zamítnuta, v rámci projektu nastaven, pokus je daná doména vyfiltrována |

Tabulka 2: Návrátové kódy DNS paketu

7 Reference

1. [Fundamentals of Computer Networking, Project 1: Simple DNS Client](#)
2. [RFC 1035: domain names – implementation and specification](#)