VYSOKÉ UČENIE TECHNICKÉ V BRNE

Fakulta informačných technológií

Počítačové komunikácie a siete
2017/2018

Projekt č.2

DNS Lookup nástroj

Martin Nizner (xnizne00) 08.04.2018

Obsah

1.Úvod do problematiky

1.1 Systém DNS

Základnou úlohou **systému DNS** je mapovanie (prevod) doménových adries (napr. pcmatousek.fit.vutbr.cz) na IP adresy (147.229.12.101). Doménovej adrese sa často hovorí tiež doménové meno (domain name).
Už pri zavedení IP adries v 70. rokoch 20. storočia sa začali používať menné ekvivalenty, ktoré sa vždy prekladali na IP adresu. Práve **systém DNS** obsahuje celosvetovou databázu IP adries a ich zrozumiteľných ekvivalentov (doménových mien). Takisto definuje, ako budeme pristupovať k týmto dátam. Pretože ide o veľmi rozsiahlu databázu, je distribuovaná na viac počítačov, kde bežia špeciálni servery, ktorým sa hovorí nameservery. IP adresu zisťujeme z doménového mena požiadavkom na server DNS. Proces vyhľadávania v systému DNS nazývame rezolúciou.
Systém DNS využívajú všetky aplikačné protokoly (např. HTTP, SMTP, FTP) pre preklad doménových adries na IP adresy. To je vždy prvá aktivita, ktorú aplikácia urobí potom, čo ju požiadame o pripojenie na vzdialenú službu a zadáme adresu v podobe doménového mena. Služba si najprv požiada o preklad na IP adresu. Až potom môže dôjsť k naviazaniu spojenia.

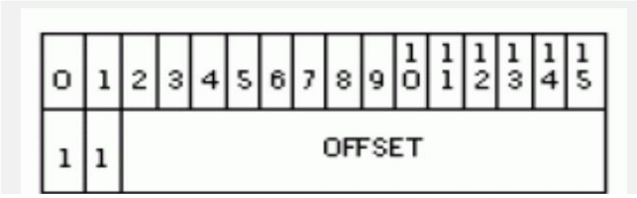
1.2 Formát záznamu DNS

Všetky typy **záznamov DNS** majú rovnaký obecný formát definovaný štandardom RFC 1035 [28]. Formát záznamov obsahuje položky NAME, TYPE, CLASS, TTL, RDLENGTH a RDATA. Každý záznam DNS obsahuje všetky uvedené položky. Položka RDATA sa líši podľa typu záznamu, kde pre daný typ (napr. A či MX) obsahuje odpovedajúce informácie. Položka NAME obsahuje meno uzlu v stromu DNS, kde je daný záznam uložený. Položka TYPE určuje typ záznamu (napr. CNAME), CLASS definuje triedu záznamu, TTL maximálnu dobu platnosti záznamu. Pokiaľ je hodnota TTL nastavená na 0, čo je bežné napr. pre záznamy SOA, záznam nesmie byť uložený v pamäti cache. Polia RDLENGTH (určuje dĺžku RDATA) a RDATA obsahujú hodnotu záznamu, na ktorú sa obvykle pýtame. Pri DNS požiadavku obsahuje formát záznamu DNS iba položky NAME, TYPE a CLASS.

Resource Records Format	Example
Name (variable length)	www.fit.vutbr.cz
Type (16 bits)	CNAME
Class (16 bits)	IN (0x0001)
TTL (32 bits)	4106 (1 h 8 min 26 s)
RDLENGTH (16 bits)	9
RDATA (variable length)	tereza.fit.vutbr.cz

1.3 Spôsob kompresie správ

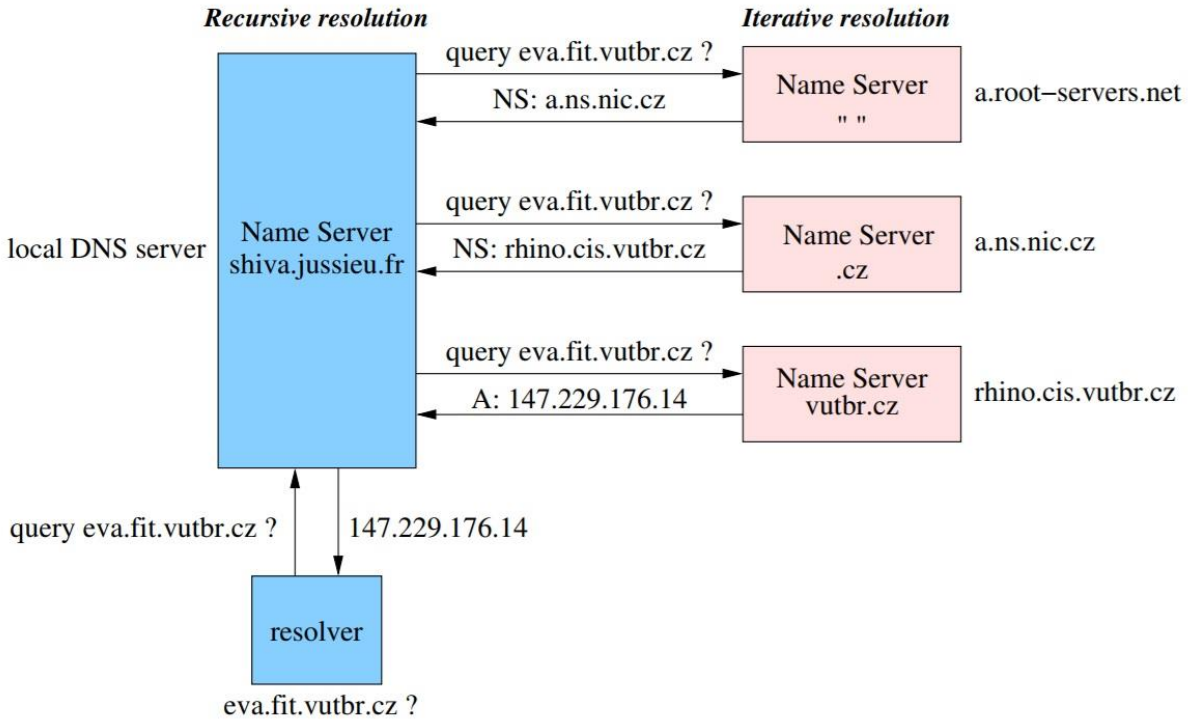
Formát DNS odpovede môže niekedy obsahovať to isté doménové meno niekoľkokrát. Opakovanie tohto doménového mena je zbytočná strata bitov v danej odpovedi. Technika kompresie správ spočíva v redukcií počtu použitých bitov tak, že nahradí opakujúce sa doménové meno pomocou ukazateľa na dané doménové meno. Ukazateľ je zapísaný na 16 bitov a má nasledovný formát:



Pointer má prvé dva bity nastavené na 1, aby sa odlíšil od dátových polí.

1.4 Rekurzívny a iteratívny spôsob požiadavku

Rekurzívny dotaz je podobný rekurzii, ktorú poznáme napr. z programovania. Resolver zašle dotaz na určitý údaj v stromu DNS konkrétnemu serveru DNS. Server DNS musí odpovedať na dotaz buď požadovanými dátami alebo chybovou hláškou, keď napr. nepozná odpoveď. Pokiaľ server nie je autoritatívny pre hľadané dáta, musí sa spýtať ďalších serverov a nájsť autoritatívny odpoveď. Môže poslať rekurzívny dotaz na niektorý z autoritatívnych serverov a čakať na odpoveď. Alebo môže poslať iteratívny dotaz a získať odkaz na iný server, ktorý pozná odpoveď. Či server podporuje rekurzívne či iteratívne dotazovanie záleží na jeho konfigurácií. Server DNS, ktorý obdržal rekurzívny dotaz, preposiela vždy rovnaký dotaz na ďalšie servery. Nikdy sa nepýta napr. na záznamy NS pro hľadanú doménu. Väčšina programov (napr. nslookup, dig) posiela rekurzívne dotazy.



Iteratívny dotaz šetrí prácu na strane serveru DNS. Pri tomto dotazovaní vráti server resolveru najlepšiu odpoveď, ktorú môže dať. Viac sa nedotazuje. Dotazovaný server DNS sa podíva do svojej lokálnej databáze. Pokiaľ nenájde odpoveď, vráti adresy serverov, ktoré sú najbližšie hľadanej adrese.

2. Popis vlastného riešenia

2.1 Spracovanie vstupných parametrov

Argumenty sa spracúvajú v hlavnej funkcií **main()** v cykle pomocou funkcie **getopt(),** ktorá postupne overí a načíta všetky vstupné parametre. Ak niektorý s parametrov chýba, prípadne je niektorý s parametrov zadaný chybne alebo dvakrát, tak dochádzka k chybe a ukončeniu činnosti programu.

2.2 Komunikácia so serverom DNS

Komunikácia so serverom DNS prebieha pomocou konečného automatu, ktorý je riešený pomocou cyklu **while** a prepínača **switch**, na základe ktorého sa v každej iterácií určí v ktorom stave sa automat nachádza a dôjde k vykonaniu požadovanej akcie pre daný stav (odoslanie správy serveru DNS, výpis prijatej správy...).
Komunikácia začína stavom **0**, v ktorom prebieha vypĺňanie pomocného bufferu potrebnými dátami. V pomocnom bufferi sa najprv vyplní úvodných 12bitov, ktoré majú vždy rovnakú hodnotu. Následne dochádza k úprave zadaného vstupného mena na formát DNS správy a vloženiu takto upraveného mena do pomocného bufferu. Nakoniec sa do pomocného bufferu zapíše typ (A, AAAA...) a trieda záznamu (IN). Takto vyplnený buffer sa následne sa odošle serveru DNS pomocou funkcie **sendto.**
Komunikácia pokračuje stavom **1**, v ktorom dochádza k prijímaniu odpovede od serveru DNS. Správa sa najprv prijme pomocou funkcie **recvfrom**. Následne dochádza ku kontrole a výpisu dat s pomocného bufferu pre prijaté správy. Úvodné bity pomocného bufferu sú ignorované a kontrola začína až od bitu, ktorý obsahuje typ prijatej odpovede. Po zistení tohto typu dochádza k výpisu prijatých dát na štandardný výstup. Spôsob, akým sa dané dáta budú vypisovať záleží na type prijatých dát. Po výpise dát na štandardný výstup dochádza k ukončeniu činnosti programu (pokiaľ nebolo vyžiadané iteratívne dotazovanie).
Ak bolo zadané iteratívne dotazovanie, tak komunikácia začína stavom **2**, ktorý slúži k úprave zadaného vstupného mena. Tento stav využíva ďalší pomocný buffer určený pre iteratívne dotazovanie. V každej novej iterácií tohto stavu sa do pomocného bufferu pre iteratívne dotazovanie pripojí jedna subdoména zadaného vstupného mena. Následne sa pokračuje stavom **0** rovnako ako pri rekurzívnom dotazovaní s tým rozdielom, že namiesto zadaného vstupného mena je použitá iba časť tohto vstupného mena, ktorá sa momentálne nachádza v pomocnom bufferi pre iteratívne dotazovanie.

2.3 Popis a návod na použitie programu

Program sa spúšťa s jedným povinným a troma nepovinnými parametrami, prípadne jedným nepovinným parametrom (**-h** slúži ako nápoveda)

Parametre:

-h <help> voliteľný parameter, pri jeho zadaní sa vypíše nápoveda a program sa ukončí.

-s <server> povinný parameter, DNS server (IPv4 adresa), na ktorý sa budú odosielať dotazy

-T <timeout> voliteľný parameter, timeout (v sekundách) pre dotaz, pevná hodnota 5 sekund.

-t <type> voliteľný parameter, typ dotazovaného záznamu: A (pevná), AAAA, NS, PTR, CNAME.

-i voliteľný parameter, vynútenie iteratívneho spôsobu rezolúcie

-f značí, že bude vrátená informácia o domácom adresári užívateľa pre daný login a **-l** značí, že bude vrátený zoznam všetkých užívateľov (ak bol zadaný login tak sa použije ako prefix pre výber užívateľov)

name povinný vstupný argument, prekladané doménové meno, v prípade parametru -t PTR program na vstupu naopak očakáva IPv4 nebo IPv6 adresu

2.4 Obmedzenia programu

Program pri niektorých typoch iteratívneho dotazu nezobrazí po ceste k výsledku všetky NS a A záznamy, ktoré sa na danej ceste nachádzajú (napr. po vypísaní záznamu 147.in-addr.arpa. IN NS r.arin.net. program vypisuje ako ďalší záznam r.arin.net. IN A 199.180.180.6 a ignoruje záznamy, ktoré sa nachádzajú medzi týmito dvoma záznamami.

3. Záver

Program bol riadne otestovaný na serveri merlin.fit.vutbr.cz. Program bol implementovaný v jazyku C. Behom implementácie bol pre lepšiu analýzu a vyhodnocovanie správ použitý program **Wireshark**.

4. Bibloigrafia

1. **P.Matoušek, Síťové aplikace a jejich architektura, 2014, ISBN 978-80-214-3766-1**
2. **Wikipedie: Otevřená encyklopedie. Aktualizované 4.10.2017 v 9:25. Dostupné na https://cs.wikipedia.org/wiki/Domain_Name_System**