FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Počítačové komunikácie a siete
Projekt č. 2
Varianta 3: DNS Lookup nástroj

2018 Patrik Holop

Obsah

1 Úvod

Tento dokument reprezentuje dokumentáciu k projektu č. 2 z predmetu Počítačové komunikácie a siete (IPK). Vytvorené programy v rámci projektu sú implementované v jazyku C++[1]. Riešená bola varianta č. 3 - DNS Lookup nástroj.

1.1 Cieľ projektu

Cieľom projektu bolo vytvoriť funkčnú klientsku aplikáciu, ktorá bude schopná posielať, prijímať a spracovávať dotazy a odpovede protokolu **DNS**.
Výsledný tvar spustenia programu potom vyzerá nasledovne

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

2 Popis systému DNS

DNS (The Domain Name System) je systém doménových mien, ktorý predovšetkým určuje spôsob, akým sú internetové doménové mená používané človekom (napr. *www.google.com*) vyhľadávané a prekladané na IP adresy používané počítačom (napr. 192.168.0.1). Jedná sa o decentralizovaný systém, ktorý distribuuje mapovanie DNS v rámci internetu autoritatívnou hierarchiou. Je realizovaný servermi a protokolom rovnakého mena.

DNS klient pošle dotaz (**DNS query**) serveru, ktorý má za úlohu spracovať požiadavku a odoslať odpoveď (**DNS response**). Keď server obdrží dotaz z inej domény na meno v rámci vlastnej domény, odošle autoritatívnu odpoveď. Problém nastane, keď server obdrží dotaz z vlastnej domény na meno v inej doméne. V takom prípade (ak nie je dané meno inak uložené, napr. v cache) musí server dotaz preposlať inému serveru. V takom prípade sa jedná o server najvyššej domény (top-level domain), ktorý v prípade, že odpoveď taktiež nepozná, preposiela dotaz autoritatívnemu serveru pre špecifickú doménu a tento proces pokračuje pokým sa nenájde odpoveď, prípadne sa zistí neexistencia takého mena.

2.1 Formát správy

Protokol DNS dokáže pracovať s viacerými typmi dotazov, typy implementované v tomto projekte nájdete v nasledujúcej tabuľke.

Typy DNS záznamov:

Typ	id	Popis
`A`	1	Záznam IPv4 adresy
`NS`	2	Deleguje DNS zónu pre použitie autoritatívnych serverov
`CNAME`	5	Alias mena
`PTR`	12	Obrátené DNS vyhľadávanie
`AAAA`	28	Záznam IPv6 adresy

Formát správy DNS:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
ID správy															
QR	OPCODE				AA	TC	RD	RA	res1	res2	res3	RCODE			
Počet dotazov															
Počet odpovedí															
Počet položiek v autoritatívnej časti															
Počet položiek v dodatočnej časti															
Dáta															

[1] Norma C++11, viz. `https://isocpp.org/wiki/faq/cpp11`

ID správy sa používa pri identifikácii správy a správnom priradení odpovedi. V tomto projekte je reprezentované ako ID procesu klienta. **QR** bit odlišuje dotaz (0) od odpovedi (1). **OPCODE** reprezentuje typ operácie, v tomto projekte má vždy hodnotu `standart query`. **AA** určuje autoritatívnu odpoveď, používa sa pri odpovedi. Nastavený **TC** bit určuje parciálnu správu (obsah bol príliš dlhý a poslaný vo viacerých správach). **RD** značí vyžiadanie rekurzívneho dotazovania. **RA** naopak značí, či dns server rekurzívne dotazy podporuje. **RCODE** určuje validitu správy, v prípade hodnoty 0 značí správny formát, prípadne odpoveď serveru.

V prípade dotazu sa vložia dotazy (questions) do DNS správy (v tomto projekte z pravidla 1) a Počet dotazov sa nastaví na 1, všetky ostatné sekcie majú dĺžku 0. V prípade odpovede sa spracovávajú všetky sekcie.

Formát dotazu (question):

Typ	Počet bitov	Popis
`QNAME`	Premenlivé	Dotazované meno
`QTYPE`	16	Typ dotazu (A, AAAA...)
`QCLASS`	16	Trieda (v projekte vždy IN - internet)

Formát odpovede (response):

Typ	Počet bitov	Popis
`NAME`	Premenlivé	Vrátené meno
`TYPE`	16	Typ odpovede (A, AAAA...)
`CLASS`	16	Trieda dotazu
`TTL`	32	Time to live v sekundách
`LENGTH`	16	Dĺžka dát (počet bytov)
`DATA`	LENGTH * 8	Samotné dáta

2.2 Spôsob kompresie dát

Aby boli správy DNS čo najkratšie, duplikované mená prípadne časti mien nahrádza DNS odkazmi. V rámci DNS existujú 2 typy označení, a to Označenie dát (**Data label**) a Označenie odkazu (**Compression label**). Označenie dát má formát `Ndata`, kde N je počet bytov doménového mena, napr. `3www`, označenie odkazu je 16 bitové číslo značiacie offset od začiatku DNS správy, kde je uložený zvyšok dát.

Aby boli jednotlivé označenia jednoznačne rozoznateľné, prvý byte označenia s hodnotou 0-63 značí označenie dát, väčšie číslo značí odkaz.

2.3 Spôsob dotazovania

V rámci dns existujú 2 spôsoby dotazovania. Prvý spôsob dotazovania je **iteratívne**. V tomto prípade sa klient dotazuje len príslušného DNS serveru na meno v rámci domény, prvý dotaz smeruje na koreňový DNS server v rámci adresy. V prípade, že server toto meno nepozná, pošle namiesto priamej odpovede zoznam autoritatívnych serverov, na ktoré sa má klient v prípade, že bude v hľadaní pokračovať, odkazovať. Tieto servery sa dajú nájsť v autorizačnej časti popísanej vo formáte správy DNS. V dodatočnej časti sa potom môžu vyskytnúť aj záznamy typu A/AAAA, z ktorých vie klient priamo získať IP adresu autoritatívneho serveru. Ak sa tak nestane, musí sa na adresu opäť dotazovať a ďalší dotaz posiela na server danej adresy. Tento postup pokračuje až pokým sa nenájde odpoveď, prípadne hľadanie skončí neúspešne.

Rekurzívny spôsob dotazovania automatizuje vyššie popísaný proces a jednotlivé pomocné dotazy nemusí vykonávať klient, ale príslušný server. V takom prípade klient dostáva priamu odpoveď. Server nemusí rekurzívne dotazovanie podporovať.

3 Programová realizácia

Klient bol naprogramovaný v jazyku C++ objektovo orientovaným prístupom. Implementácia komunikácie vychádza z dema UDP komunikácie v archíve Demo_C.zip[2].

3.1 Kontrola parametrov

Kontrola parametrov prebieha pomocou funkcie `getopt`. Nie je povolené opakované zadanie parametru, čo zabezpečuje premenná typu bool (flag) pre každý parameter, ktorá značí, či bol už parameter zadaný. Rovnakým spôsobom sa rieši nepovolená kombinácia parametrov -h s inými. Či bola zadaná hodnota pri určitom parametre sa zisťuje pomocou premennej `optarg` vychádzajúca z funkcie `getopt`. Zadanie loginu prípadne príliš mnoho parametrov sa vypočíta vzťahom `argc-optind`, kde argc je celkový počet zadaných parametrov a optind je počet nespracovaných parametrov po spracovaní funkciou getopt. Pri parametri -T sa kontroluje validita prevodu na číslo funkciou `strtol`.

3.2 Hlavička DNS správy

Hlavička DNS správy je reprezentovaná triedou **BIT_DNS_HEADER** v súbore `ipk-loookup.cpp`. Celá hlavička je reprezentovaná **usporiadanou množinou 96 bitov** implementovanej pomocou `std::bitset` jazyka C++. Pri práci s jednotlivými flagmi/oblasťami v rámci DNS hlavičky sa správny offset zisťuje z preddefinovanej mapy `std::map<string, int>` (mapovanie názvu napr. "rd"na offset 72) a hodnoty sa nastavia pomocou bitovej operácie XOR.

3.3 Obsah DNS správy

Trieda **BIT_DNS_QUERY** reprezentuje celú DNS správu a rozširuje triedu BIT_DNS_HEADER (DNS hlavičku) o samotné dáta správy reprezentované typom `string`. V konštruktore sa pri tvorbe DNS správy zakódujú všetky vstupné dáta pomocou bitových množín do vhodného formátu. V prípade, že sa jedná o iteratívne dotazovanie nastaví potrebný bit v hlavičke.

3.4 Klient

Klient je reprezentovaný triedou **Client** v súbore `ipk-lookup.cpp` a zabezpečuje samotnú UDP komunikáciu so serverom pomocou vyššie spomenutých tried. Taktiež má na starosti vracať aj vypisovať dáta vo vhodnom formáte.

Atribúty				Hlavné funkcie		
typ	**meno**	**popis**		**typ**	**meno**	**popis**
int	port	port (53)		int	send_query	pošle dotaz a uloží odpoveď
string	server_name	adresa/dns serveru		int	get_name	extrahuje celé meno zo správy
int	timeout	timeout[s]		vector	process_response	extrahuje všetky dáta zo správy
string	message	DNS správa		vector	establish_connection	zabezpečí celú komunikáciu
int	client_socket	klientsky socket		int	load_answers	načíta odpovede zo správy
sockaddr_in	server_adress	adresa servera		int	print_ipv6	vypíše ipv6 adresu

3.4.1 Timeout

V rámci projektu bolo nutné riešiť timeout zadaný užívateľom, prípadne základnou hodnotou 5 sekúnd. UDP socket umožnuje nastaviť timeout v rámci komunikácie pomocou štruktúry `timeval` a funkcie `setsockopt`. Takto nakonfiguraný timeout spôsobí, že funkcia `recvfrom` po uplynutí danej časovej dĺžky skončí v návratovým kódom ≤ 0 rovnako ako v prípade zlyhania doručenia správy.

[2]Dostupné z `https://wis.fit.vutbr.cz/FIT/st/course-files-st.php?file=%2Fcourse%2FIPK-IT%2Fother%2FDemo_C.zip&cid=11963`

3.5 Spôsob dotazovania

V prípade, že klient pomocou parametru -i iteratívny spôsob nevyžiada, automaticky na nastavuje bit "rd"v hlavičke DNS správy. Iteratívne dotazovanie je v projekte implementované vo funkcii `iterative_search`. Najprv sa pošle dotaz sa východzí server, v prípade že sa meno nenájde sa získava IP adresa nového autorizovaného servera z autorizovanej a dodatočnej časti odpovede. Ak sa tam nenájde, skúsi sa poslať dotaz typu `A` na referečný server. Funkcia sa rekuzívne zavolá pre nový nájdený server. Iteratívny spôsob dotazovania podporuje cachovanie nájdených hodnôt. Slúži na to globálna mapa `std::map<string, string>` mapujúca mená na adresy. Keďže cachovanie nie je implementačne úplne dokončené, može sa dotaz na adresu tej istej úrovne v rámci DNS hierarchie vyskytnúť opakovane.

3.6 Reverzné vyhľadávanie

V prípade dotazov typu `PTR` sa musí vstupná adresa správne zakódovať počas vytvárania DNS správy. V prípade **IPv4** adresy je poradie vstupných osmíc bitov prehodené a pridaný postfix `.in-addr.arpa.`. V prípade **IPv6** adresy sa musia získať všetky hexadecimálne čísla tvoriace adresu a obrátiť sa ich poradie, za každým číslom nasleduje znak '.'. Postfix pre tento typ adresy je `.ip6.arpa.`

4 Demonštrácia funkčnosti

Funkčnosť celej komunikácie budem demonštrovať na serveri Merlin. Ako demonštračné parametre boli použité príklady zo zadania.

```
xholop01@merlin: ~/ipk2$ ./ipk-lookup -s 8.8.8.8 www.fit.vutbr.cz
www.fit.vutbr.cz. IN A 147.229.9.23
xholop01@merlin: ~/ipk2$ ./ipk-lookup -s 8.8.8.8 www4.fit.vutbr.cz.
www4.fit.vutbr.cz. IN CNAME tereza.fit.vutbr.cz.
tereza.fit.vutbr.cz. IN A 147.229.9.22
xholop01@merlin: ~/ipk2$ ./ipk-lookup -s 8.8.8.8 -t CNAME www4.fit.vutbr.cz.
www4.fit.vutbr.cz. IN CNAME tereza.fit.vutbr.cz.
xholop01@merlin: ~/ipk2$ ./ipk-lookup -s 8.8.8.8 -t AAAA www4.fit.vutbr.cz.
www4.fit.vutbr.cz. IN CNAME tereza.fit.vutbr.cz.
xholop01@merlin: ~/ipk2$ echo "$?"
1
xholop01@merlin: ~/ipk2$ _
```

Obrázek 1: Ukážka základných dotazov

```
xholop01@merlin: ~/ipk2$ ./ipk-lookup -s 8.8.8.8 -t PTR 2001:67c:1220:8b0::93e5:b013
3.1.0.b.5.e.3.9.0.0.0.0.0.0.0.0.b.8.0.0.2.2.1.c.7.6.0.1.0.0.2.ip6.arpa. IN PTR merlin6.fit.vutbr.cz.
xholop01@merlin: ~/ipk2$ ./ipk-lookup -s 8.8.8.8 -t PTR 147.229.13.238
238.13.229.147.in-addr.arpa. IN PTR pckucerajan.fit.vutbr.cz.
xholop01@merlin: ~/ipk2$ _
```

Obrázek 2: Ukážka reverzného vyhľadávania

```
xholop01@merlin: ~/ipk2$ ./ipk-lookup -s 8.8.8.8 -t AAAA -i www.fit.vutbr.cz
. IN NS a.root-servers.net.
a.root-servers.net. IN A 198.41.0.4
cz. IN NS a.ns.nic.cz.
a.ns.nic.cz IN A 194.0.12.1
vutbr.cz. IN NS pipit.cis.vutbr.cz.
pipit.cis.vutbr.cz IN A 77.93.219.110
fit.vutbr.cz. IN NS gate.feec.vutbr.cz.
gate.feec.vutbr.cz IN A 147.229.71.10
www.fit.vutbr.cz IN AAAA 2001:67c:1220:809::93e5:917
xholop01@merlin: ~/ipk2$
xholop01@merlin: ~/ipk2$ ./ipk-lookup -s 8.8.8.8 -t PTR -i 147.229.13.238
. IN NS m.root-servers.net.
m.root-servers.net. IN A 202.12.27.33
in-addr.arpa. IN NS b.in-addr-servers.arpa.
b.in-addr-servers.arpa IN A 199.253.183.183
147.in-addr.arpa. IN NS r.arin.net.
net. IN NS h.gtld-servers.net.
h.gtld-servers.net IN A 192.54.112.30
arin.net. IN NS ns1.arin.net.
ns1.arin.net IN A 199.212.0.108
r.arin.net. IN A 199.180.180.63
229.147.in-addr.arpa. IN NS rhino.cis.vutbr.cz.
cz. IN NS c.ns.nic.cz.
c.ns.nic.cz IN A 194.0.14.1
vutbr.cz. IN NS pipit.cis.vutbr.cz.
pipit.cis.vutbr.cz IN A 77.93.219.110
rhino.cis.vutbr.cz. IN A 147.229.3.10
238.13.229.147.in-addr.arpa. IN PTR pckucerajan.fit.vutbr.cz
```

Obrázek 3: Ukážka iteratívneho dotazovania

5 Použité zdroje

BOTH D., *Introduction to the Domain Name System (DNS)* [online] 2017 [cit. 2018]. Dostupné z: https://opensource.com/article/17/4/introduction-domain-name-system-dns

MOCKAPETRIS P., *RFC 1034: Domain names - concepts and facilities* [online]. Dostupné z: https://tools.ietf.org/html/rfc1034

THOMPSON S. et al., *RFC 3596: DNS Extensions to Support IP Version 6* [online]. Dostupné z: https://tools.ietf.org/html/rfc3596

MOCKAPETRIS P., *RFC 1035: Domain names - implementation and specification* [online]. Dostupné z: https://tools.ietf.org/html/rfc1035

RYŠAVÝ O., RÁB J., *IPK - BSD schránky - 3. přednáška*, 2017 [online]. Dostupné z: https://wis.fit.vutbr.cz/FIT/st/course-files-st.php?file=%2Fcourse%2FIPK-IT%2Flectures...