DNS PROTOCOL CLIENT SUITABLE FOR LECTURING

Tomáš Sousedík

Master Degree Programme (2), FEEC BUT

E-mail: xsouse01@stud.feec.vutbr.cz

Supervised by: Jan Jeřábek

E-mail: jerabekj@feec.vutbr.cz

Abstract: This paper describes DNS protocol and messages it uses to obtain desired information. The protocol is described from its creation and occasions that caused the creation of the protocol. There are also mentioned multicast versions of DNS. Later on is described developer application the way it sends DNS queries and process domain name to required format.

Keywords: DNS, DNS properties, DNS client, description of DNS

1. ÚVOD

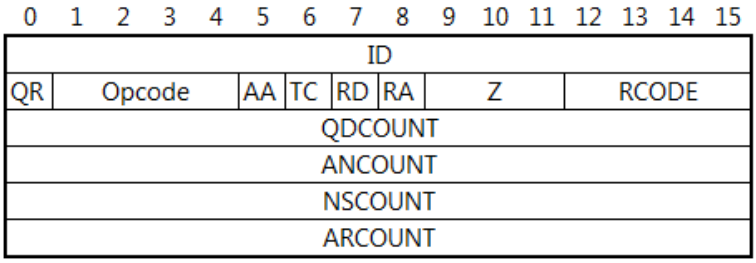
Tento článek popisuje návrh vlastního klienta protokolu DNS, který by byl vhodný pro prezentační účely. Protokol DNS je velmi důležitý stavební kámen internetu, neboť bez něj bychom si byli nuceni pamatovat všechny IP adresy, místo pro člověka jednodušších, doménových jmen. Na začátku této práce je popsáno, co vedlo ke vzniku tohoto protokolu. Popsány jsou i nejdůležitější zprávy, za jejichž pomoci probíhá výměna záznamů a hledaných informací.

2. POPIS SYSTÉMU DNS

Hlavní cíl DNS protokolu [1] je vytvoření konzistentního jmenného prostoru, se snadným přístupovým systémem. S postupným rozrůstáním sítě se zvětšoval i počet internetových stránek a dat, ke kterým se dá přistupovat. Myšlenka DNS je velice jednoduchá. Jedná se o nahrazení 32 bitových IP adres jednoduchým doménovým jménem. Tato doménová jména se ukládají na serverech, ke kterým se přistupuje pomocí programů – resolverů. Dříve všechna jména byla spravována jednou institucí, a to Network Information Center (NIC) v jediném souboru (HOSTS.TXT), který byl přenášen pomocí FTP (file transfer protocol) všem ostatním hostům v síti. Přenos tohoto souboru při vzniku nové verze byl náročný na prostředky sítě. S růstem internetu a jednotlivých sítí vznikl požadavek, aby si organizace spravovaly vlastní jména a adresy. Tyto organizace však při každé změně musely počkat, až NIC změní obsah souboru HOSTS.TXT, aby se změny v rámci organizace projevily i na Internetu. Postupem času také rostl zájem strukturovat vlastní jmenný prostor. Velikost databáze a četnost aktualizací naznačuje, že jednotlivé databáze musí byt spravovány distribuovaně na různých serverech. Aby se zamezilo přílišnému opakování stejných dotazů, tak si dotazující se stanice ukládá odpovědi serveru do paměti pro pozdější použití.

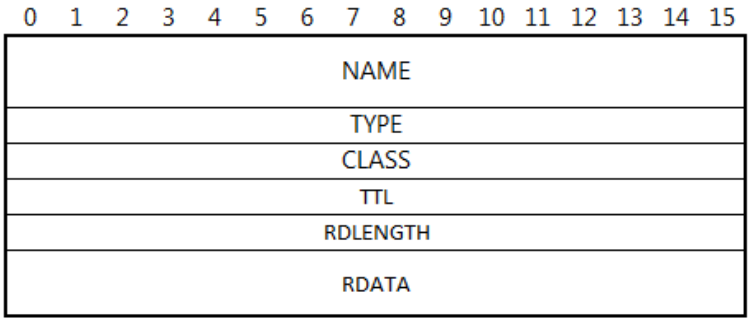
2.1. ZPRÁVY V DNS

DNS je aplikační protokol. Z hlediska ISO/OSI modelu leží na 7. vrstvě. K jeho přenosu se používá transportní protokol UDP i TCP, v obou případech se využívá portu 53. Při komunikaci s DNS serverem se vyměňují různé zprávy, které mají sjednocený formát. Některé části jsou vždy přítomny, jiné se ve zprávě vyskytují jen v případě potřeby.



Obrázek 1: Záhlaví protokolu DNS

Záhlaví, které je znázorněno na Obrázek 1, je ve zprávě vždy. Specifikuje, zda jsou přítomné další části a v jakém počtu, tedy například kolik odpovědí má server na náš dotaz. Také je zde určeno, jestli je zpráva dotazem nebo odpovědí. Například zpráva dotaz obsahuje pole nutná pro popis dotazu zasílaného na jmenný server. Jedná se o pole jako druh dotazu (QTYPE), třída dotazu (QCLASS) a samotné jméno (QNAME).



Obrázek 2: Struktura v dalších částech DNS zprávy

Poslední částí jsou RR (resource records) záznamy. Tyto mají všechny stejnou strukturu, ať již se jedná o obyčejnou odpověď, autoritativní odpověď nebo dodatečné informace. Struktura je znázorněna na Obrázek 2.

Přijatá data se ukládají do paměti počítače, aby bylo zamezeno častému opakování stále stejných dotazů. Z tohoto důvodu jsou záznamy vybaveny hodnotou TTL, která určuje, jak dlouho mohou být uloženy ve vyrovnávací paměti před tím, než se musí vyslat nový dotaz. Je to také kvůli tomu, kdyby došlo ke změně IP adresy, tak se bude klient pokoušet připojit na již neplatnou adresu. Po vypršení tohoto časovače se klient zeptá znovu a připojí se na správnou adresu. Většinou však, když má dojít k nějaké takové změně se časovače nastaví na velmi malou hodnotu nebo na 0, což znamená, že se údaj nesmí ukládat vůbec. [2]

Zajímavostí DNS systému je multicastová varianta [3], která má snahu využít i zpráv nevyslaných vlastním přičiněním. V případě, že stanice ve stejné síti vyšle DNS query, tak následnou odpověď zachytí i jiné stanice na síti a informace uložené ve zprávě uloží do své paměti pro případné použití.

3. NÁVRH KLIENTA SYSTÉMU DNS

Pro tento projekt jsem si vybral programovací jazyk C#. Je to objektově orientovaný programovací jazyk vyvinutý firmou Microsoft pro vývoj programů na platformu .NET. Jazyk je velmi podobný C++ a má i obdobnou syntaxi.

Mnou vyvinutý program DNSresolver vysílá zprávy DNS query na zvolený server a následně zpracovává DNS response zprávy zaslané serverem. Po zpracování a dekódování pak získané informace zobrazuje na obrazovku. Vstupem programu je hledaná adresa, způsob přenosu (IPv4/IPv6) a hledaná informace (A nebo AAAA apod.), téměř vše, co nastavení DNS protokolu umožňuje. Je potřeba ošetřit, jestli zadaná IP adresa je opravdu IP adresa serveru, ale to zjistíme až podle toho,

jestli odpovídá na daném portu nebo ne. Z toho důvodu je možné nastavit počet pokusů o spojení a také dobu, po jejímž uplynutí je pokus považovaný za neúspěšný. Po úspěšném spojení se serverem může teprve dojít k odeslání dat. Po přijetí dat program zpracuje přijaté informace. Pokud dojde zpráva poškozena nebo dojde neúplná, pokusí se program o nové vyslání dat. Nepovede-li se ani tak zprávu úspěšně zpracovat je zpracovávání přerušeno. Kvůli různé délce doménových jmen, mají tato speciální tvar, který můžete vidět v ukázce na Obrázek 3.

0	1	2	3	4	5	6	7	8	9	10	11	12	13
3	119	119	119	5	118	117	116	98	114	2	99	122	0
3	w	w	w	5	v	u	t	b	r	2	c	z	0

Obrázek 3: Tvar doménového jména přenášený v DNS zprávě

Z toho je patrné, že doménové jméno předchází byte, který určuje počet následujících bytů a je následován tímto počtem bytů a tak dále až k ukončujícímu oktetu nul. Jelikož se doménové jména ve zprávě opakují a to při odpovědi, používá se pointerů ke zkrácení zprávy. Dojde-li při přenosu dat k nějaké chybě, tak se program pokusí opětovně zaslat dotaz. Po úspěšném zpracování dat, jsou data zobrazena na obrazovku. Úspěšný výsledek můžete vidět na Obrázek 4.

```
Waiting for connection...
Id: 0
Flags: 32896
Qdcount: 1
Ancount: 2
Nscount: 0
Arcount: 0
Qname: www.vutbr.cz
Qtype: A
Qclass: IN
Answer to : www.vutbr.cz
Type: CNAME
Class: IN
Time to live: 00:00:31
Data Length: 13
Data: piranha.ro.vutbr.cz
Answer to : piranha.ro.vutbr.cz
Type: A
Class: IN
Time to live: 00:55:31
Data Length: 4
Data: 147.229.2.90
```

Obrázek 4: Ukázka výstupu programu

4. ZÁVĚR

V práci byly popsány způsoby překladu doménových jmen. Je vidět, že základem je protokol DNS, který tvoří jeden ze základních kamenů Internetu. Na tomto protokolu jsou založena další vylepšení, zejména multicastové varianty DNS, tedy mDNS a LLMNR. V praktické části jsem provedl předběžný návrh DNS klienta. Zprovozněny jsou všechny základní funkce. Jsou v něm ošetřeny základní možné chyby, kdy je například zadána špatná adresa serveru nebo když server neodpovídá. V navazující práci mám v plánu přidat grafické rozhraní, které umožní přehledně zvolit požadované hodnoty a srozumitelně prezentovat přijaté informace.

REFERENCE

[1] MOCKAPETRIS, P.V. RFC 1034. *Domain names - concepts and facilities*. 1987. Dostupné z: http://www.rfc-editor.org/rfc/rfc1034.txt

[2] MOCKAPETRIS, P.V. RFC 1035. *Domain names - implementation and specification*. 1987. Dostupné z: http://www.rfc-editor.org/rfc/rfc1035.txt

[3] CHESHIRE, S. a M. KROCHMAL. APPLE. RFC 6762. *Multicast DNS*. Únor 2013. Dostupné z: http://www.rfc-editor.org/rfc/rfc6762.txt