FIT VUT v Brně

Počítačové komunikace a sítě

2017/2018

IPK – Projekt číslo 2

DNS Lookup nástroj

David Endrych (xendry02)

Brno, 7. 4. 2018

Obsah

Uvedení do problematiky

DNS je systém doménových jmen realizovaný DNS servery a DNS protokolem. Úkolem DNS je převod doménových jmen a IP adres sítě. IP adresy jsou buď IPv4 záznam typu A nebo IPv6, záznam typu AAAA. Doménové jméno může být dlouhé až 255 znaků. Jednotlivé subdomény jsou odděleny tečkou a můžou být dlouhé až 63 znaků. Servery jsou organizovány hierarchicky. Na autorizovaných serverech jsou trvale uloženy záznamy k určité doméně/zóně. Autonomních serverů je většinou více na danou doménu/zónu. Minimálně dva primární a sekundární. Na nich jsou uloženy stejné záznamy, proto se informace na nich musí udržovat synchronizovaně.

Protokol je postaven na jednoduchém principu. Na server se pošle dotaz a server na něj odpoví. Protokol pracuje s transportním protokolem UDP ale také i s TCP. Pokud je dotaz odeslán pomocí UDP, ale odpověď přesahuje maximální délku 512 B, tak se odpověď pošle zkráceně a pokud odesílatel dotazu chce celou odpověď tak musí dotaz poslat znovu, ale tentokrát pomocí transportního protokolu TCP. Servery naslouchají u obou transportních protokolů na port 53.

DNS protokol pracuje s více typy dotazů, nejčastěji používané jsou A, AAA, CNAME, NS, PTR. Typ A je požadavek na získání IPv4, AAAA slouží k získání IPv6 adresy, CNAME k získání doménovému jménu aliasu, NS k získání autoritativních name serverů, PTR k získání doménového jména z IPv4 nebo IPv6 adresy.

Formát packetu

Formát packetu je stejný jak pro dotaz, tak i pro odpověď. Rozdíl je v tom, které části se vyplňují a jakými hodnotami. Protokol má pět částí záhlaví, dotazy, odpovědi, autoritativní nameservery, doplňující informace.

Záhlaví

Záhlaví se vyplňuje jak v dotazu, tak i v odpovědi. Je dlouhé dvanáct bajtů. První dva bajty je identifikátor zprávy. Další dva bajty jsou řídící.

První bit určuje, jestli se jedná o dotaz nebo odpověď. Další čtyři bity určují typ otázky. Následující bit určuje, jestli je odpověď autoritativní. Další bit je nastaven na hodnotu jedna, pokud byla odpověď zkrácena na 512 B v případě komunikace pomocí transportního protokolu UDP. Další dva bity nastavují rekurzivní překlad. První bit se nastavuje při dotazu určuje, jestli klient vyžaduje rekurzivní překlad, druhý se nastavuje při odpovědi a určuje, jestli server vůbec umožňuje rekurzivní překlad. Další tři bity jsou rezervovány pro budoucí použití. Poslední čtyři bity určují návratový kód.

Dále následují čtyři dvojbajtová pole. První pole určuje počet dotazů, následuje počet odpovědí, počet autoritativních name serverů v sekci a poslední pole určuje počet doplňujících informací.

Dotaz

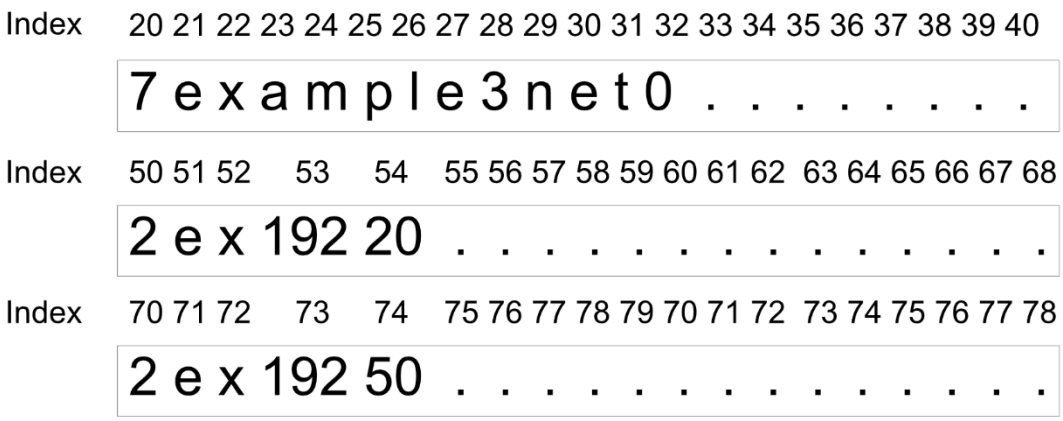
Při dotazu se vyplňuje většinou pouze sekce záhlaví, kde se uvede počet dotazů na jedna a poté se vyplní sekce odpovědi. Další sekce se nechají prázdné. Sekce dotaz se skládá za tří částí doménového jména, typu dotazu a třídy dotazu. Doménové jméno se nezapisuje tečkovou notací, ale vždy se před část, kterou by normálně oddělovala tečka přidá bajt, který svou hodnotou určuje kolik znaků se bude za ním vyskytovat, než by se narazilo na tečku, na konci se nachází bajt s hodnotou nula což znamená, že doménové jméno už končí příklad jméno www.fit.vutbr.cz. By se uložilo do packetu jako 3www3fit5vutbr2cz0. Typ dotazu je číselná hodnota např. typ AAA má hodnotu 28. Třída dotazu je také číselná hodnota v projektu je to vždy hodnota 1 což značí třídu internet.

Odpověď

Při odpovědi je v packetu uloženo záhlaví, dotaz, a navíc se vyplňují i sekce odpovědi, autoritativní name servery a doplňující informace. Každé sekce nemusí být vyplněná nebo může mít více uložených položek například více odpovědí. Počet záznamu v těchto sekcích se pozná podle informací uložených v záhlaví. V těchto sekcích mají záznamy společnou strukturu, pokud se jedná o běžné typy, což je případ tohohle projektu jsou to typy A, AAAA apod., ale např. typ SOA má jinou strukturu záznamu. V případě běžných typů mají formát, že jako první je zde uvedeno doménové jméno, poté se na dva bajty uloží typ věty, dále třída věty, taky se zde jedná o dva bajty, doba platnosti, která je uložena ve čtyřech bajtech, délka datové části uložena ve dvou bajtech a následuje datová část ve které je buď uloženo doménové jméno nebo IP adresa.

Způsob komprese

Doménové jméno nebo jeho část se může opakovat, proto je povolena komprese, která slouží k redukci velikosti packetu. Tahle komprese se realizuje pomocí ukazatele na předchozí výskyt v packetu. Ukazatel je velký 16 bitů. Pozná se podle toho, že první dva bity jsou nastaveny na hodnotu jedna. Následujících 14 bitů určuje index. Čísluje se od začátku packetu. Komprese může mít více zanoření. Například jako první je v packetu uloženo doménové jméno example.net, poté se objeví v packetu doménové jméno ex.example.net, protože chceme šetřit jméno, tak ex zapíšeme do packetu a poté následuje ukazatel na example.net, někde se ale poté může objevit doménové jméno ex.ex.example.net tudíž zapíšeme ex do packetu a poté použijeme ukazatel na ex.example.net ve kterém se ale také nachází ukazatel na example.net viz obrázek.



Dotazování

Rekurzivní

Pokud nastavíme dotaz, že vyžadujeme rekurzivní překlad. Tak server, na který dotaz pošleme, pokud nezná odpověď spustí algoritmus, který má za úkol vyhledat požadovanou odpověď. Hledání začne u kořenových DNS serverů a postupuje do nižších úrovní až k požadované odpovědi a klientovy pošle konečný výsledek. Ne každý server umožňuje rekurzivní dotazovaní. Pokud server dotazování neumožňuje odešle klientovy pouze odpověď, kterou zná sám. Tím nám říká, kde máme odpověď hledat.

Iterativní

V případě iterativního dotazovaní děláme část, kterou za nás dělal server u rekurzivního dotazování sami. Začneme dotazem na kořenový DNS server. Poté postupně postupujeme do nižších úrovní až narazíme na požadovanou odpověď.

Návrh aplikace

Aplikace jako první zpracuje zadané argumenty, která zadal uživatel. Poté je potřeba podle zadaných argumentů připravit DNS záhlaví a dotaz. V případě rekurzivního dotazovaní se nastaví v řídících bitech příznak vyžádání rekurze a jméno dotazu a typ dotazu se nastaví podle zadaných argumentů. V případě iterativního dotazování se nastaví název dotazu na root tj. tečka a typ dotazu se nastaví na NS. Dále se vytvoří socket přes který se bude komunikovat, packet který se bude posílat na server, nastaví se pro něj timeout, adresa DNS serveru. Následuje poslání packetu a následné přijetí odpovědi ze serveru. Zde může nastat nastavený timeout pokud by server neodpovídal. Poté se přechází na zpracování odpovědi, způsob zpracovávání se liší podle toho, jestli se jedná o iterativní způsob nebo rekurzivní.

V případě rekurzivního způsobu se zpracovává pouze hlavička a pokud nenastala chyba tak se dále zpracovává pouze sekce odpovědi. Všechny odpovědi se vypíší na terminál v požadovaném tvaru. Pokud se nenajde odpověď požadovaného typu tak dojde k ukončení s návratovým kódem 1.

V případě iterativního způsobu potřebujeme jako první získat IP adresu nameserveru, který jsme získali po zpracování odpovědi. Jako název dotazu se zvolí název nameserveru a zvolí se typ A. Pokud se nepodaří získat adresu, tak se aplikace ukončí s návratovým kódem 1. Poté se už můžeme postupně ptát dotaz, který zadal uživatel přeš argumenty, dokud nenajdeme odpověď nebo nenastane nějaká chyba. Adresy, na které se dotazuje se postupně ukládají do zásobníků. Při zpracování odpovědi je několik možností, které se můžou stát.

První varianta je, že se najde odpověď, která byla požadovaná. Tím pádem můžeme aplikaci ukončit s návratovou hodnotou 0. Další možnost je že nastane chyba na serveru a tím pádem se aplikace ukončí s návratovou hodnotou 1. Další možnost je, že se najde další nameserver a jeho IP adresa na kterém se má pokračovat v hledání. Poslední varianta je, že se najde nameserver, ale nenajde s k němu IP adresa, v tomhle případě se vyhodí ze zásobníků poslední adresa a začne se IP adresa hledal na následujícím serveru z vrcholu zásobníku. Poté když se podaří najít IP adresu tak se pokračuje v hledání dotazu z argumentu na získané adrese name serveru.

Popis zajímavých částí

Řešení timeoutu

Timeout řeším pomocí setsockopt, kterému se předává struktura timeval, které nastavuji délku v sekundách. Poté když se volá funkce recvfrom tak kontroluji návratovou hodnotu funkce. Když dojde k timeoutu ukončí se aplikace s návratovou hodnotou 1.

Parsování adres

Na převody, výpisy adres jsem se rozhodl, že nepoužiji knihovní funkce, ale vytvořil jsme si vlastní funkce jako je například parse_address_aaaa nebo parse_adress_ptr.

Návod použití

./ipk-lookup [-h]

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

- h - vypíše nápovědu
- s - povinný parametr, DNS server (IPv4 adresa), na který se budou odesílat dotazy
- T - timeout v sekundách
- t - typ dotazovaného záznamu: A (výchozí), AAAA, NS, PTR, CNAME.
- i - vynucení iterativního způsobu
- name - překládané doménové jméno, v případě parametru -t PTR program na vstupu očekává IPv4 nebo IPv6 adresu.

Program vrací hodnotu 0 v případě úspěchu, hodnotu 1 v případě neúspěchu a hodnotu 2 pro chybně zadané parametry při spuštění.

Bibliografie

[0] P. Mockapetris, RFC 1034: Domain names - concepts and facilities, https://tools.ietf.org/html/rfc1034.

[1] P. Mockapetris, RFC 1035: Domain names - implementation and specification, https://tools.ietf.org/html/rfc1035.

[2] S. Thomson, et al., RFC 3596: DNS Extensions to Support IP Version 6, https://tools.ietf.org/html/rfc3596.

[3] KABELOVÁ, Alena a Libor DOSTÁLEK. Velký průvodce protokoly TCP/IP a systémem DNS. Computer Press. ISBN 8025138860, 9788025138861.