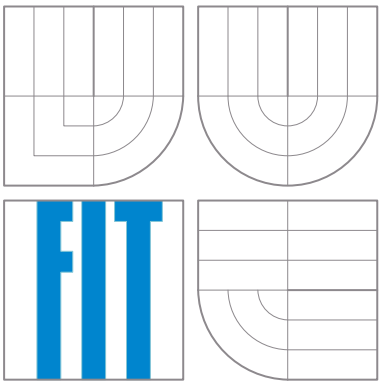
Vysoké učení technické v Brně
Fakulta informačních technologií



Dokumentace k projektu pro předmět ISA

Traceroute

19. listopadu 2012

Autor: Kateřina Zaklová, xzaklo00@stud.fit.vutbr.cz
Fakulta Informačních Technologií
Vysoké Učení Technické v Brně

Obsah

1 Úvod

Tato zpráva vznikla jako dokumentace k projektu do předmětu Síťové aplikace a správa sítí. Dokument se ve zkratce zabývá představením a vysvětlením pojmu traceroute, a dále popisuje návrh, implementaci a použití výsledné aplikace. Tato aplikace může být využívána pro zobrazení průchodu packetu sítí.

Tato dokumentace byla vytvořena v LaTeXu.

2 Důležité pojmy

Pro návrh a implementaci programu traceroute je důležité mít alespoň částečné znalosti, co se týče funkčnosti aplikace, použitých datagramů aj. V této části dokumentu je popsán základní princip aplikace traceroute, datagramy, které aplikace odesílá a typy zpráv, kterými na tento přenos odpovídají jednotlivé uzly.

2.1 Traceroute

Program traceroute se používá především pro analýzu počítačových sítí. Aplikace vypisuje směrovače (uzly) na cestě datagramů odesílaných ze zdroje k danému cíli, kde jednotlivé uzly snižují TTL (time to live) v hlavičce datagramu.

Program po každém úspěšném odeslání packetu zvýší TTL, přičemž první odeslaný packet má vždy hodnotu 1. Packet prochází jednotlivými uzly, každý uzel sníží TTL packetu o 1 a pošle packet dál. Ve chvíli, kdy TTL klesne na hodnotu 0 a packet není v cílovém uzlu, směrovač vyšle ke zdroji chybovou ICMP zprávu a packet zahodí. Aplikace následně z těchto ICMP zpráv sestaví tabulku cesty packetu. Odesílané datagramy mohou být ICMP, TCP nebo UDP, pro naše zadání úkolu je to pouze ICMP.

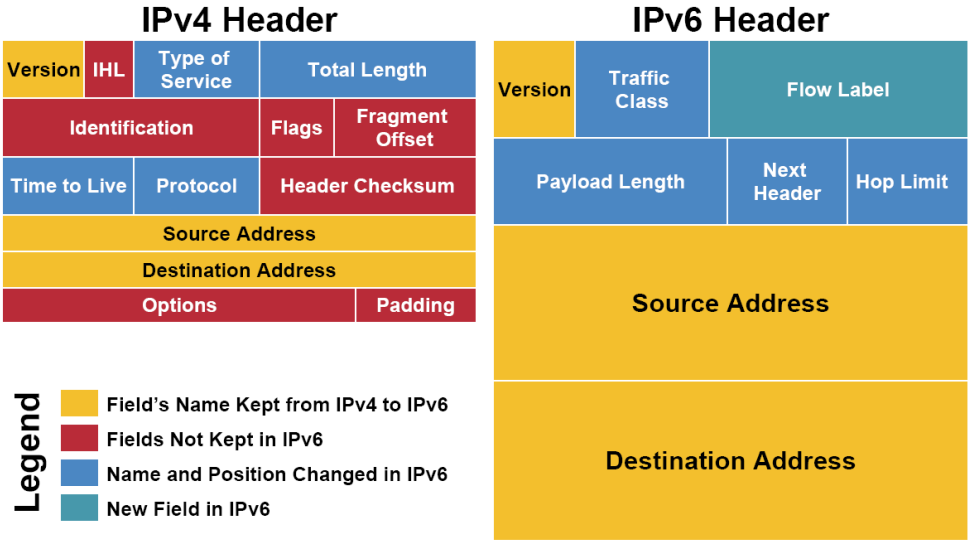
2.2 IP Datagramy

Datagramy IPv4 a IPv6 se využívají k doručování dat mezi jednotlivými počítači. Rozdíly mezi strukturou těchto datagramů jsou znázorněny na obrázku 1. Jejich hlavičky obsahují množství položek, pro implementaci traceroute však potřebujeme jen některé z nich.

Pro IPv4:

- *source address* (zdrojová IP adresa),
- *destination address* (cílová IP adresa),
- *version* (IPv4, IPv6),
- *ihl* (délka hlavičky),
- *tot_len* (celková délka packetu),
- *frag_off* (posunutí fragmentu),
- *protocol* (protokol vyšší vrstvy pro komunikaci, v tomto případě ICMP) a
- *ttl* (doba života datagramu).

U IPv6 pro zjednodušení postačí pracovat pouze s *ttl*.

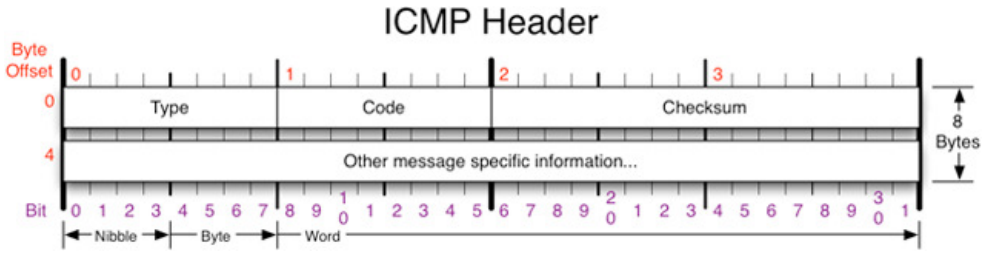


Obrázek 1: Hlavička IPv4 a IPv6 *Zdroj:* `http://www.tcpip6.com/`

2.3 ICMP Datagramy

Protokol ICMP se používá v síti především pro odesílání chybových zpráv (například oznámení, že služba není dostupná, router není dosažitelný aj.). ICMP se obvykle nevyužívá přímo, výjimkou je program ping nebo naše aplikace. Ta posílá ICMP zprávy typu *Echo Request* a očekává odpověď pro zjištění dosažitelnosti cílového PC a času, za který packety dojdou do cíle a zpět.

ICMP zprávy se konstruují z IP datagramu, který je zapouzdřuje. Existují dvě verze ICMP protokolu, pro IPv4 je to ICMPv4, IPv6 používá obdobný protokol ICMPv6. Hlavička ICMP datagramu je zobrazena na obrázku 2.



Obrázek 2: Hlavička ICMP *Zdroj:* `http://www.insecure.in/packet_header_analysis.asp`

2.4 Typy ICMP zpráv

Existuje řada ICMP zpráv, zde jsou však znázorněny jen některé z nich – ty, které potřebujeme pro náš program. Jednotlivé zprávy se u protokolů ICMPv4 a ICMPv6 liší typem a kódem, tyto rozdíly znázorňuje tabulka 1.

Zpráva	ICMPv4		ICMPv6		Popis
	Typ	Kód	Typ	Kód	
Echo request	8	0	128	0	požadavek na odpověď
Echo reply	0	0	129	0	odpověď na požadavek
Time exceeded	11	-	3	-	vypršel časový limit
Network unreachable	3	0	1	0	nedostupná cílová síť
Host unreachable	3	1	1	3	nedostupný cílový stroj
Protocol unreachable	3	2	4	1	není možné použít vybraný protokol
Communication administratively prohibited	3	13	1	1	zakázaná komunikace

Tabulka 1: Typy ICMP zpráv[2]

3 Návrh programu

Aplikace *trace* musí pracovat s hlavičkami datagramů, proto využívá *RAW_SOCKET*y. Pomocí nich posílá a přijímá packety složené z IP a ICMP datagramu. Z těchto datagramů pak získává potřebné informace – zdrojovou a cílovou IP adresu, typ poslané/přijaté zprávy. Zároveň využívá položku IP datagramu *ttl*, pomocí které získává informace o bodech, kterými packet na své cestě prochází.

Aplikace posílá packety po jednom, a po každém úspěšném odeslání navýší v IP hlavičce packetu položku *ttl* o hodnotu 1. První odeslaný packet má *ttl* 1. Packety prochází na cestě k cíli řadou směrovačů, kde každý z těchto směrovačů sníží *ttl* o 1 a odešle packet dál. V okamžiku, kdy na směrovač dojde packet s hodnotou *ttl* 0, směrovač packet zahodí a odešle chybovou zprávu typu *Time exceeded* zpět na zdroj. V případě, kdy packet dorazí až na cílový směrovač, zdroj dostane odpověď typu *Echo reply* a program úspěšně končí. Může však dojít k situaci, kdy cíl není dostupný, používaný protokol je zakázán nebo není povolena komunikace. Pak program končí s chybou a příslušným výpisem pro chybový stav. Aplikace preferuje IPv6 traceroute, pokud zdrojový stroj nemá IPv6 konektivitu, pak se spustí traceroute pro IPv4.

4 Implementace

Program je implementován v jazyce C/C++. Není navržen objektově, avšak využívá některých objektů ze standardních knihoven C++. Aplikace je vytvořena pro OS Linux, byla vyvíjena a testována na Ubuntu 10.04, Ubuntu 12.04 a Xubuntu 11.10. Program nebyl testován na operačních systémech Unix.

4.1 IPv4 traceroute

Běh programu spouští výsledek funkce `getaddrinfo()`. Pokud tato funkce zjistí požadavek na IPv4 adresu, která nemá IPv6 alternativu, dojde k volání funkce `makeSocket()`, která vytvoří přijímací a odesílací sockety. Pokud jsou sockety vytvořeny úspešně, dojde k volání funkce `trace4()`, která zajišťuje chod celého IPv4 traceroutu. Nejdříve se nastaví potřebné údaje pro IP a ICMP datagramy, poté se v cyklu v rozmezí zadaných parametrů (implicitně nastaveno

na 1-30) zvyšuje *ttl* a posíláme zprávu pomocí funkce `sendto()` a následně prostřednictvím funkce `select()` kontrolujeme, zda nám došla odpověď. V případě, že odpověď nepřijde ve vymezeném časovém úseku, dochází k timeoutu a výpisu `*`. V případě, že odpověď došla, funkcí `recvfrom()` ji přečteme, zkontrolujeme podle *ID* a *sekvenčního čísla* v ICMP hlavičce, zda se jedná o správný packet, a následným voláním funkce `printIPv4()` tiskneme průběžně uzly na cestě k cíli. Po dosažení cíle, maximálního *ttl* nebo chybového stavu dojde voláním funkce `cleanTheMess()` k uvolnění alokované paměti a program končí.

4.2 IPv6 traceroute

Program funguje obdobně jako IPv4 traceroute, avšak používá oddělené funkce. IPv6 traceroute spouští rovněž výsledek funkce `getaddrinfo()`. Pokud zadaná cílová adresa podporuje IPv6, dojde přednostně k volání funkcí pro IPv6. Nejdříve funkce `makeSocket6()` vytvoří potřebné sockety pro odesílání a přijímání zpráv, následně dojde k volání `trace6()`. Tato funkce nastaví pouze ICMP hlavičku, u IP hlavičky nastavuje jen *ttl* prostřednictvím funkce `setsockopt()`, nastavovat další údaje zde není bezprostředně nutné. Další chod IPv6 traceroutu je velice podobný jako u verze IPv4, posíláme zprávy, kontrolujeme, zda nám došla odpověď, odpovědi čteme, kontrolujeme jejich správnost, a tiskneme směrovače v cestě pomocí funkce `printIPv6()`. Funkce pro uvolnění alokovaných zdrojů je společná pro obě verze.

4.3 Checksum

Pro odeslání zprávy je nutné nastavit kontrolní součet v ICMP hlavičce. Tento výpočet zajišťuje funkce `chksm()`, kterou jsem kompletně převzala z internetového zdroje [1].

4.4 Použité knihovny

Pro běh programu jsou využívány funkce, struktury a konstanty z následujících knihoven:

```
#include <stdlib.h>
#include <netdb.h>
#include <netinet/ip6.h>
#include <netinet/icmp6.h>
#include <netinet/ip_icmp.h>
#include <sys/time.h>
#include <arpa/inet.h>
#include <iostream>
#include <iomanip>
```

5 Použití aplikace

Program může být spuštěn pouze s právy roota, bez těchto práv nedojde k vytvoření *RAW_SOCKET*u, a program předčasně skončí s chybou.

Parametry spuštění programu jsou následující:
-h, ***-help***, ***--help*** Program vypíše na standardní výstup nápovědu, poté končí.
[-f first_ttl] [-m max_ttl] host Spouští běh programu.

Argument ***-f*** specifikuje počáteční *ttl* programu, 1-255, je nepovinný, implicitně nastavený na 1.
Argument ***-m*** specifikuje maximální*ttl* programu, 1-255, je nepovinný, implicitně nastavený na 30.
Argument ***host*** je IPv4, IPv6 nebo DNS jméno cíle. Pokud jsou použity argumenty ***-f***, ***-m***, musí být uvedeny před tímto argumentem.

6 Závěr

Program posílá pomocí *RAW_SOCKET*ů zprávy k cílové adrese a vypisuje směrovače na cestě packetu k tomuto cíli. Může sloužit jako prototyp aplikace pro analýzu sítě. Program je překládán překladačem g++, pro překlad slouží soubor *Makefile*. Aplikace byla úspěšně otestována v prostředí operačního systému Linux, konkrétně na distribucích Ubuntu 10.04, Ubuntu 12.04 a Xubuntu 11.10.

A Metriky kódu

Počet souborů: 1 soubor

Počet řádků zdrojového textu: 777 řádků

Velikost statických dat: 844B

Velikost spustitelného souboru: 30748B

Reference

[1] Stuart HASLAM. Network connectivity for embedded systems, 2003. [Online] `http://www.bozon.net/projects/ncfes/code/syntax-highlighted/icmp.c.html`.

[2] J. POSTEL. Internet control message protocol, 1981. [Online] `http://tools.ietf.org/html/rfc792`.