



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

MĚŘENÍ ZTRÁTOVOSTI A RTT

MEASUREMENT OF LOSS AND RTT

SÍŤOVÉ APLIKACE A SPRÁVA SÍTÍ

NETWORK APPLICATIONS AND NETWORK ADMINISTRATION

AUTOR PRÁCE

AUTHOR

MARTIN VAŠKO

BRNO 2017

Obsah

1	Úvod	2
2	Návrh a spracovanie	3
2.1	Round trip time	3
2.2	IPv4 IPv6	3
2.3	ICMP a UDP	3
2.3.1	Zloženie ICMP správy	3
2.4	Paralelne spracovanie uzlov	4
3	Implementácia	5
3.1	Parametre	5
3.2	Schránky a vlákna	5
3.2.1	Vlákna	5
3.2.2	Schránky	5
3.2.3	Časovače	5
3.3	Zaujímavé časti implementácie	6
3.3.1	Implementácia prijímača datagramov UDP	6
3.3.2	Algoritmus pre prijímanie paketov od uzlov	7
3.3.3	Nastavenia pripojenia	7
3.3.4	Zoznam použitých knižníc	7
4	Obmedzenia riešenia a práca s programom	9
4.1	Obmedzenia	9
4.2	Príklad používania	9
4.3	Ukončenie programu	10
5	Záver	11
	Literatura	12

Kapitola 1

Úvod

Projekt sa zameriava na monitorovanie sieťových uzlov paralelne. V prípade straty paketov alebo prekročení hodnoty RTT (Round trip time) nad zvolenú hodnotu, vypíše informácie o strate na užívateľský výstup. Každú hodinu sa vypisuje štatistika, a každých **-t**[9] minút štatistika o počte stratených paketov (pri kombinácii s **-r** sa vypisuje aj koľko paketov prekročilo hodnotu RTT).

Kapitola 2

Návrh a spracovanie

2.1 Round trip time

Round trip time alebo obojsmerné oneskorenie je doba, ktorá uplynie od vyslania signálu(paketu) z jednej stanice na druhú až po návrat späť na prvú stanicu. Tento round trip time môžeme zisťovať pomocou voľne dostupného nástroja s názvom **ping**. Pre zisťovanie týchto časov je potrebné zaistiť časovú známku pre daný dátový segment v pakete. Pre jednoduchosť som zvolil časovú známku o veľkosti 16 bajtov a jedná sa konkrétne o štruktúru *struct timeval*[1].

2.2 IPv4 IPv6

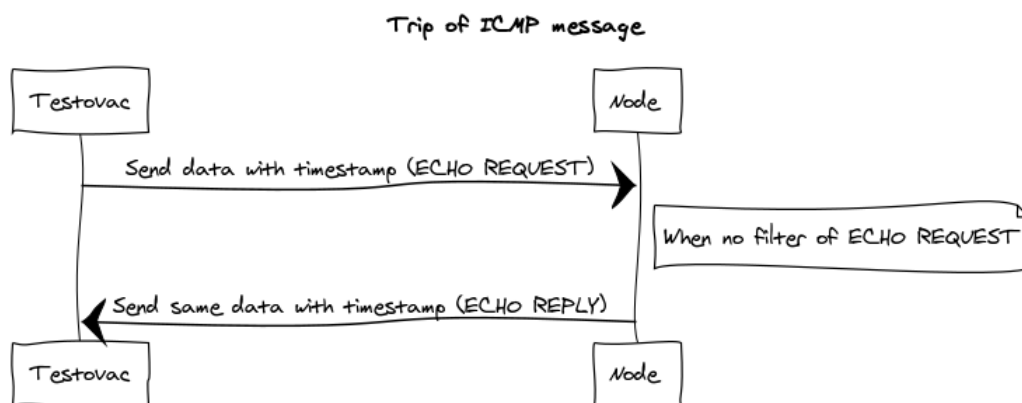
Táto aplikácia podporuje posielanie packetov pre uzol s IPv4 adresou alebo aj IPv6, podľa typu adresy uzla. Táto diverzita prináša v prípade ICMP správy IPv4 hlavičku ako odpoveď narozdiel od IPv6 ICMP správy. Preto v projekte museli byť zohľadnené dve verzie spracovávania packetov, IPv4 spracovanie a IPv6 spracovanie.

2.3 ICMP a UDP

ICMP správa typu ECHO_REQUEST[4] je kontrolná správa, ktorá nám zisťuje o uzloch či sú aktívne, v prípade úspechu nám odpovie uzol ICMP správou ECHO_REPLY v iných prípadoch nám posiela ICMP správy iných typov, pre ktoré nie je potrebné ďalšie spracovanie vrámci merania stratovosti alebo zisťovania RTT hodnoty.

2.3.1 Zloženie ICMP správy

ICMP správa sa skladá z typu (ECHO_REQUEST) kódu 0, čísla vlákna [2] a sekvenčného čísla. Za touto hlavičkou nasleduje časová známka, ktorá nám hovorí o čase, v ktorom bola správa vygenerovaná a zasiela sa na daný uzol. Zvyšok dát je náhodne generovaných.



2.4 Paralelne spracovanie uzlov

Paralelizmus je zaistený pomocou vlákien(ang. threads). Týmto spôsobom je implementovaná časť, ktorá spracováva uzol, odosielanie ICMP paketov, časť, v ktorej testovač pôsobí ako “server” a štatistická časť.

Kapitola 3

Implementácia

3.1 Parametre

Testovač obsahuje triedu *Parse_param*, ktorá obsahuje flagy všetkých možných prepínačov. Ako prvé spracuje všetky tieto prepínače (ak za nimi musí nasledovať hodnota spracuje sa aj táto hodnota). Zvyšné parametre vstupu by mali byť uzly na spracovanie a odosielanie dát. Zapišu sa do vektora nespracovaných parametrov po prvom prechode. Potom sa tento vektor vyprázdni a zisťuje sa či daný uzol je uzlom IPv4, IPv6. Ak ani jedna z týchto možností nenastala jedná sa o chybu. Program sa ukončí s **návratovou hodnotou 1**.

3.2 Schránky a vlákna

Hodnoty spracovaných prepínačov prenesieme ako informácie do novej triedy s názvom *Socket_thread*. Maximálny počet vlákien v programe je 16384, pre uzly je možné použiť buď 16380 ak máme komunikáciu ICMP pre komunikáciu UDP 16379, kvôli osobitnému vláknu pre prijímanie a odosielanie UDP datagramov.

3.2.1 Vlákna

Pre používanie štandardného výstupu a viac vlákňového programovania museli byť zavedené taktiež semaforey aby sa predišlo prepisovaniu výstupu rôznych vlákien. Štatistické vlákno jedine prebieha bez toho aby sa čakalo na jeho ukončenie rámci posielania/prijímania dát z uzlu¹.

3.2.2 Schránky

Schránky sú dvojého typu IPv4 a IPv6 podľa typu uzlu. **Pozor** pre účely použitia ICMP správ (bez prepínača **-u a -p**) je potrebné použiť práva privilegovaného používateľa (príkaz **sudo** na linuxovej distribúcii). Dôvodom je použitie schránok RAW[odkaz raw socket].

3.2.3 Časovače

V mnohých prípadoch je potrebné používať funkciu **usleep**² aby sa zbytočne nemárnilo procesorovým časom a mohli tak fungovať iné vlákna kým vlákno čaká napr. interval -i. Hodinová štatistika funguje nad presným časovačom C++[3]. Možnosť použiť **usleep** nebolo

¹Oddelené spracovanie vlákna http://man7.org/linux/man-pages/man3/pthread_detach.3.html

možné, lebo by sa čakalo kým proces dospí a až potom by sa ukončil program. Tento chrono časovač sa počíta po každom prijatom package alebo po odozve timeout.

3.3 Zaujímavé časti implementácie

Schránka je súborový popisovač, kde sa zapisujú a odosielajú dáta. Každý blok prijímania dat je realizovaný pomocou linuxovej funkcie select, ktorá má za úlohu čakať na súborový popisovač, kým nie je možné z neho čítať. UDP datagram musí mať pri špecifikácii veľkosti aspoň 16 bajtovú veľkosť, pre ICMP správu je to 24 bajtov. Pri zadaní menších jednotiek je táto hodnota zadaná na najmenšiu možnú napr. pri zadaní 8 bajtov a UDP datagramu program si nastaví hodnotu veľkosti na 16 bajtov.

3.3.1 Implementácia prijímača datagramov UDP

Prijímač správ, ktorý sa špecifikuje parametrom `-l` bol implementovaný cez schránku typu `AF_INET6` kde bola nastavená pokročila možnosť prijímať cez túto schránku IPv4 aj IPv6 komunikáciu súčasne pomocou `IPV6_V6ONLY`. Možnosť prijať správu a zistiť jej veľkosť bez toho aby sme nenačítavali zbytočné prázdne znaky bolo docielené vo funkcii `recvfrom`. Položku flags bolo potrebné nastaviť na `MSG_TRUNC` pre zistenie veľkosti dát a `MSG_PEEK` pre zachovanie dát na sieťovej karte. Po tomto prijatí sa vyhradilo potrebné miesto v pamäti pre prijatie správy a preposlanie naspäť prijemcovi.

²<http://man7.org/linux/man-pages/man3/usleep.3.html>

3.3.2 Algoritmus pre prijímanie paketov od uzlov

Protokol IPv4 a IPv6 bol v kóde rozdelený do separátnych častí tak ako ICMP a UDP ale princíp ostal rovnaký.

Data: socket, adresa, retazec a hodnota intervalu -i a -w^[9]

Result: Prijatý a spracovaný paket

Inicializácia dát;

Nastavenie štruktúry timeval a čakanie na príchod dát do schránky;

while *Paket je určený pre mňa alebo nezachytený signál SIGINT* **do**

if *Prijaté dáta bez prekročenia času* **then**

 Načítaj dáta;

 Z paketu vyber čas, ktorý sa nachádza na začiatku dátovej sekcie;

 Vypočítaj hodnotu RTT z času odoslania a aktuálneho času;

 Pripočítaj paket do celkového počtu;

if *Bola zadaná hodnota -r a skontroluj či nebol prekročený čas* **then**

 Zvýš počet prekročených paketov;

if *verbose mód* **then**

 Vypíš informácie o prekročení;

end

end

end

if *Prijaté dáta s prekročením času* **then**

 Zvýš počet nezachytených paketov;

else

 chyba funkcie select(), vypíš chybu a ukonči program;

end

end

Algorithm 1: Spracovávanie prijatých paketov

3.3.3 Nastavenia pripojenia

Trieda Socket_thread obsahuje metódu **set_connections**, ktorá sa stará o prevedenie všetkých uzlov na samostatné vlákno podľa typu adresy. Inicializuje mutexy pre správu výstupu a inkrementovanie globálnej premennej a zachytávanie signálu SIGINT. Použitie prepínaču -l vedie k vytvoreniu vlákna, ktoré obsluhuje schránku pripojenú na konkrétny port.

3.3.4 Zoznam použitých knižníc

```
#include <fcntl.h>
#include <algorithm> pre výpočet štatistík
#include <cmath>
#include <sys/socket.h>
#include <numeric>
#include <sys/types.h>
#include <sys/syscall.h>
#include <sys/time.h>
#include <arpa/inet.h>
#include <netinet/in.h>
```



```
#include <netinet/ip.h>
#include <netinet/icmp6.h>
#include <chrono> pre časovače
#include <netdb.h>
#include <unistd.h>
#include <pthread.h> pre vlákna
#include <string>
#include <vector> vektory parametrov a hodnôt round trip time
#include <csignal> Signál handler
#include <sys/stat.h>
#include <sys/wait.h>
#include <iostream>
#include <ctime> gettimeofday
#include <sched.h>
#include <iomanip>
#include <cstdio>
#include <netinet/ip_icmp.h>
```

Kapitola 4

Obmedzenia riešenia a práca s programom

4.1 Obmedzenia

Pri testovaní na vlastnom rozhraní sieťovej karty(loopback) s nízkou hodnotou `i[9]` (do 50) je detekované pomerne veľké množstvo paketov, ktoré prekročili hodnotu 2-krát RTT, kvôli veľkej odozve množstva odpovedí nestíha sieťová karta spracovávať požiadavky dostatočne rýchlo aby sa neprekročila daná hodnota.

Priatie ICMP správy nie je úplne dokonalé. Riešenie predpokladá, že je možné prijatú správu dostať celú, ak náhodou dojde k nedokonalému priatiu paket sa nezapočíta do straty paketu a čaká sa na ďalší paket. Dôsledkom je potom odchýlka v počte paketov, ktoré sa poslali a či ich bolo možné spracovať.

4.2 Príklad používania

Všeobecné použitie je možné dohľadať aj v manuálových stránkach.

```
testovac [-h] [-u] [-t <interval>] [-i <interval>] [-p <port>] [-l <port>] [-s <size>]
[-r <value>] <uzol1> <uzol2> <uzol3> ... -h – zobrazí možnosti použitia testovača.
-u – pre testovanie se použije UDP protokol.
-s – veľkosť dát pre odeslanie, predvolená hodnota je 56B.
-t <interval> – interval v sekundách, za ktorý je strátovosť vyhodnocovaná, predvolená
hodnota je 300s.
-i <interval> – interval v milisekundách ako často zasielať testovacie správy, predvolené
nastavenie 100 ms.
-w <timeout> – doba ako dlho sa čaká na odpoveď, iba pri neobdržaní odpovede,
predvolená hodnota je 2s inak 2 x hodnota vypočítaného RTT.
-p <port> – špecifikácia UDP portu.
-l <port> – špecifikácia načúvaného UDP portu.
-r <value> – špecifikácia RTT hodnoty, ak RTT prekračuje danú hodnotu, reportuje
sa.
-v - verbose mód, program vypisuje na stdout prijaté pakety, tj. chová sa ako príkaz
ping.
```

<uzol> – IPv4/IPv6/hostname adresa uzlu

Posielanie ICMP správ na monitorovanie uzlu

testovac merlin.fit.vutbr.cz

Tento príklad zahŕňa všeobecné nastavenia parametrov. Vyhodnocovanie strátovosti správ je raz za 300 sekúnd, správy sa generujú každých 100 milisekúnd a na prvú odpoveď čakáme dobu 2 sekundy.

Posielanie UDP datagramov na monitorovanie uzlu

testovac localhost -u -p 12345 -l 12345 -i 20 -t 60 -w 4

Spustenie testovača v tomto režime zahájí komunikáciu UDP nad portom 12345 a zároveň sa pustí vlákno s nastaveným socketom na odposluch na port 12345 (ďalší parameter za -l). Vyhodnocovanie strátovosti datagramov je raz za minútu (60 sekúnd). Generovanie datagramov je každých 20 milisekúnd a na prvý datagram čakáme dobu 4 sekundy.

Meranie round trip time

testovac www.dsl.sk -r 7

Testovač v tomto režime zahájí komunikáciu ICMP. V tomto prípade ostávajú všetky hodnoty generovania a vyhodnocovania v preddefinovanom stave ale chceme merať, koľko správ presiahlo dobu round trip time o viac ako 7 milisekúnd. Štatistika môže vyzeráť nasledovne (testované zo siete kolejnet).

2017-11-20 13:51:17.82 www.dsl.sk: 0.003% packet loss, 9 packet lost

2017-11-20 13:51:17.82 www.dsl.sk: 0.013% (39) packets exceeded RTT threshold 7ms.

4.3 Ukončenie programu

Program beží v nekonečnom cykle a je ho možné prerušiť signálom SIGINT (klávesová skratka na linuxe Ctrl+C). Tento signál vyvolá ukončenie všetkých nekonečných cyklov, zmazanie všetkých dynamicky pridelených tried, uzavretie schránok. Úspešne ukončenie programu skončí návratovou hodnotou 0. Neúspešné ukončenie programu môže nastať z viacerých rôznych dôvodov. Ukončenie v prípade neinicializovanej schránky hodnotou 2. Ukončenie s prepínačom -l pri nenaviazaní schránky hodnotou 3.

Kapitola 5

Záver

Projekt je funkcionalitou podobný nástroju fping. Má zopár výhod oproti bežným nástrojom napr. je možné nastaviť testovač aj do módu na prijímanie paketov ako server, takže je možné robiť nad ním klasickú UDP prevádzku. Nevýhodou je zasielanie UDP odpovede s rovnakým obsahom, čo na bežnej internetovej prevádzke je neobvyklý jav (každá aplikácia má svoj štandard správ). Keďže dáta sú náhodné a UDP datagram obsahuje 16 bajtovú informáciu o čase, nedokážeme dostať odpoveď od uzlov iba ak na nich beží podobný prijímač ako sa dá špecifikovať prepínačom -l. Existuje varianta spracovania komunikácie, ktorá nemá žiadne obmedzenia ale pre experimentálne účely bolo rozhodnuté o určitej odchýlke zasielania a spracovávaní paketov.

Literatura

- [1] *Linux Programmer's Manual*. [Online; navštíveno 2.11.2017].
URL <http://man7.org/linux/man-pages/man2/gettimeofday.2.html>
- [2] *Linux Programmer's Manual*. [Online; navštíveno 7.11.2017].
URL <http://man7.org/linux/man-pages/man2/gettid.2.html>
- [3] *Time library*. [Online; navštíveno 10.11.2017].
URL <http://www.cplusplus.com/reference/chrono/>
- [4] Postel: *Internet control message protocol*. [Online; navštíveno 28.10.2017].
URL <https://tools.ietf.org/html/rfc792>