FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Měření ztrátovosti a RTT

Dokumentace projektu do předmětu ISA

17.11.2017 Vích Štěpán xvichs00

Obsah

1 Úvod

Tato dokumentace důkladně popisuje projekt do předmětu ISA. K dokumentaci je navíc vytvořena manuálová stránka `testovac.1`, kterou lze zobrazit příkazem `make man`. V druhé části najdeme návod na použití aplikace, její překlad a spuštění. V další části nalezneme návrh a popis implementace programu. Dokumentace je vysázena v systému LaTeX.

2 Návod na použití

2.1 Překlad aplikace

Zdrojové soubory programu jsou přeložitelné příkazem make ve zdrojovém adresáři. Překladový skript je uložen v souboru Makefile, který navíc obsahuje příkaz pro smazání binárních souborů a jeden testovací příklad. Překlad probíhá pomocí standardu c++11 s využitím knihovny pthread.

2.2 Spuštění a parametry

Program se spouští pod právy superuživatele root. Proto je dobré spouštět program např. `sudo ./testovac www.root.cz -v`. Práva superuživatele jsou nutná kvůli využití RAW socketů. Program se spouští s volitelnými parametry. Každý s parametrů má volitelnou hodnotu, která bude platit pokud tento parametr nezadáme. Parametry se zpracují pomocí funkce getopt. Seznam volitelných parametrů:

- `-h` Vypíše se pouze nápověda a program se ukončí s hodnotou 0.
- `-l arg` Program poslouchá na portu specifikovaném argumentem. V tomto případě program čeká na příchod UDP paketu na daný port a odeslání paketu zpět na zdrojovou adresu.
- `-s arg` Argument určuje velikost odesílaných dat v paketu. Do této velikosti se nezapočítávají velikosti ICMP zpráv. Pokud posíláme UDP zprávu musí být argument větší než 0. Výchozí hodnotou je 56B.
- `-t arg` Argument určuje interval pro vyhodnocování ztrátovosti jednotlivých uzlů. Interval začíná od startu programu a ve výchozím nastavení je to 300 sekund. Jednotkou argumentu jsou sekundy.
- `-i arg` Argument určuje interval odesílání testovacích paketů. Výchozí hodnota je 100ms. Jednotkou argumentu jsou milisekundy.
- `-w arg` Maximální doba, po kterou čekáme na odpověd odeslaného paketu. Pokud do té doby paket nepřijde, označíme jej za ztracený. Jednotkou argumentu jsou sekundy. Výchozí hodnota je 2 sekundy.
- `-p arg` Specifikace portu na který odesíláme UDP pakety. Tento parametr je nutné kombinovat s parametrem -u jinak nemá žádný účinek. Výchozí hodnota je 0.
- `-r arg` Argument určuje hodnotu thresholdu. Pokud rtt paketu překročí hodnotu argumentu, tato skutečnost se uloží a při výpisu ztrátovosti se vypíše počet paketů, které překročili tuto ztrátovost.
- `-u` Parametr určuje odesílání paketů pomocí UDP protokolu.
- `-v` Verbose parametr. Při zadání se vypisují všechny příchozí pakety a informace o nich, např. odesílatel paketu, velikost paketu, rtt paketu.
- `host ...` Seznam uzlů, které se musejí monitorovat. Neuvádí se pokud je zadán parametr -l nebo -h. Adresa uzlu je IPv4/IPv6/hostname.

3 Návrh aplikace

Program je uložen ve zdrojovém souboru main.cpp. Kódy pro překlad zdrojového souboru se nachází v souboru Makefile. Program je napsán klasickým imperativním stylem, bez použití objektového programování, které C++ nabízí. Pomocné funkce jsou volané z hlavní funkce main. Byli vytvořeny tři pomocné struktury, jedna pro ukládání informací o odeslaných paketech (struktura `node`), druhá pro ICMPv6 hlavičku (struktura `icmpv6_hdr`) a třetí pro ukládání informací o již uvolněných paketech z paměti (struktura `stats`).

3.1 Návratové kódy

Program vrací 4 různé návratové kódy. Pokud vše proběhne v pořádku vrací program standardní hodnotu 0. Pokud jsou zadány neznámé příznaky programu, argumenty příznaků jsou v nestandardním formátu, nebo adresy jsou ve špatném formátu vrací program hodnotu 1. Obecně jakékoliv špatné spuštění programu vrací hodnotu 1. Pokud program spadne na alokaci paměti, vrací hodnotu 2. Pokud selže jakákoliv síťová funkce (bind, connect, send, recv, ...) vrací program hodnotu 3. Pokud nelze vytvořit podproces hlavního procesu vrací program hodnotu 4.

4 Popis implementace

V této části popisuji podrobně jednotlivé funkce a hlavní implemetace.

4.1 Funkce main

Funkce main nejprve zpracuje všechny zadané parametry pomocí funkce getopt. Poté se rozhodne zda spustí lokální server a bude poslouchat na lokálním portu na příjem paketů, nebo rozběhne odesílání a příjem paketů pro testování uzlů (záleží podle příznaku `-l`). Při testování uzlů, se pro každý uzel vytvoří podproces a tento podproces vytvoří vlákna obsluhující všechny potřeby aplikace. Toto řešení je elegantní, jelikož zpřehledňuje a snižuje velikost kódu programu. Každý podproces tak testuje jen jeden uzel a vypisuje o něm statistiky.

Před spuštěním vláken se provede zjišťování informací o adrese pomocí funkce `getaddrinfo` a převod této adresy do textového řetězce pomocí funkce `inet_ntop`. Tyto funkce zajistí kompatibilitu pro IPv4 i IPv6 adresy. Také dojde k vytvoření soketu typu `SOCK_RAW`, spojení soketu s adresou uzlu funkcí `connect` a předání tohoto soketu do funkcí `listening`, `ping` nebo `udpListening`, `udpPing` pokud je zadán parametr `-u`.

4.2 Funkce ping

Tato funkce běží ve vlákně, které vytvoří podproces zpracovávající jednotlivý uzel. Vstupem je soket na který funkce v pravidelných intervalech odesílá ICMPv4/v6 pakety pomocí funkce `send`. Po úspěšném odeslání se přidá nová položka do zřetězeného seznamu již odeslaných paketů. Mezi informace, které se do seznamu ukládají patří sekvenční číslo paketu a čas odeslání paketu. Při odeslání paketu se jako id icmp zprávy nastaví id podprocesu. To zajistí jedinečnost id v rámci programu a pomůže odfiltrovat ostatní icmp pakety. Funkce běží v nekonečné smyčce a na konci každé iterace se pomocí funkce `usleep` uspí na dobu předepsanou parametrem `i*1000`. Pokud se paket zasílá na IPv4 adresu odesílá se ICMPv4 zpráva, pokud na IPv6 adresu odesílá se ICMPv6 zpráva. Za obsahem ICMP zprávy se navíc posílají náhodná data, jejichž náhodnost (seed) se odvíjí od aktuálního času v milisekundách, který získáme funkcí `gettimeofday`. Počet náhodných dat je závislý na parametru `-s`. Jeho výchozí hodnota je 56B.

4.3 Funkce listening

Funkce běží v paralelním vlákně, který vytvoří proces zpracovávající jednotlivý uzel. Vstupem je soket na kterém funkce čeká na příjem ICMP zprávy. Čekání je neblokující (nastaveno u soketu pomocí funkce `fcntl`) a funkce běží v nekonečné smyčce. Příjem zprávy je uskutečněn pomocí funkce `recv`. Při přijetí se zkontroluje zda se jedná o ICMP zprávu, zkontroluje se její typ, kód a id zprávy, které se musí rovnat id podprocesu. Jakmile se zpráva přijme, ve zřetězeném seznamu se označí jako přijatá (pokud již není označená jako ztracená). Pokud je nastaven flag verbose, tak se každý příchozí paket vypíše na standardní výstup.

4.4 Funkce udpPing a udpListening

Fungují na podobném principu jako funkce `ping, listening` s tím rozdílem, že odesílá UDP pakety ze soket typu SOCK_DGRAM. Náhodná data se odesílají také na stejném principu jako funkce `ping, listening`.

4.5 Funkce statistics

Funkce běží v samostatném vlákně podprocesu. Probouzí se každou jednu hodinu a spočítá a vypíše podrobné statistiky o stavu uzlu. Mezi statistiky patří počet odeslaných a ztracených paketů, nejnižší, průměrné a nejvyšší RTT a střední odchylku všech RTT. Tyto výpočty provádí tak, že prochází zřetězený seznam paketů v cyklu a zaznamenává si hodnoty, které potřebuje pro výpočet daných veličin. Již uvolněné pakety své informace uložili do struktury `stats`, která se při výpočtech používá také. Jakmile se jednou statistika vypočte, vypočte se s ní i průměrné RTT, které se nastaví místo hodnoty timeout (ten nastavujeme pomocí parametru `-w`). Po výpočtu statistik dojde k uvolnění paměti zřetězeného seznamu.

4.6 Funkce lossability

Pracuje na stejném principu jako funkce `statistics`. Vypisuje a vypočítává pouze počet odeslaných a ztracených paketů. Pokud je nastaven příznak -r, vypisuje počet paketů jejichž rtt překročilo hodnotu zadanou argumentem (v ms).

4.7 Funkce requestTimeout

Funguje paralelně ve vlákně v nekonečné smyčce a zajišťuje kontrolu ztráty paketu. V nekonečném cyklu prochází zřetězený seznam a u všech odeslaných paketů kontroluje jestli ji nevypršel timeout. To poznají odečtením aktuálního času od času odeslání paketu. Pokud je tato hodnota vyšší než nastavený timeout, označí se paket za ztracený a již se nemodifikuje (ani kdyby na něj přišla odpověď).

4.8 Funkce udpServerStart

Tato funkce neběží v samostatném vlákně ani v podprocesu. Je volaná z hlavní funkce pokud je nastaven příznak `-l`. Funkce postupně vytvoří soket, připojí se na lokální port pomocí funkce `bind` a čeká v nekonečné smyčce na příchod paketů pomocí funkce `recvfrom`. U soketu je důležité nastavit volbu `SO_REUSEADDR`, aby lokální adresa prošla funkcí `bind`. Jakmile přijde udp paket, server tento paket odešle zpět. Funkce na standardní výstup nic nevypisuje.

5 Zdroje

Knihy, přednášky a internetové články o dané problematice, které pomáhali při tvorbě projektu.

- Přednášky, opora a skripta z předmětů ISA na VUT FIT v Brně.
- https://tools.ietf.org/html/rfc792
- https://en.wikipedia.org/wiki/Standard_deviation
- https://en.wikipedia.org/wiki/Internet_Control_Message_Protocol_version_6