



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

Dokumentacia

Počítačová komunikácia a siete – 2.projekt
Bandwidth measurement

9. dubna 2018

Ján Jakub Kubík

Obsah

1	Popis problematiky merania Banwidthu	2
2	Moja implementácia	3
2.1	popis činnosti a obmedzenia	3
3	Demonštrácia činnosti a testovanie	4
3.1	Konkrétne príklady	4

1 Popis problematiky merania Bandwidthu

Bandwidth je definovaný ako dostupná šírka prenosového pásma. Bandwidth nie je jedna hodnota ktorá je počas prenosu dát rovnaká. Bandwidth je počas prenosu ovplyvňovaný mnohými faktormi. Bandwidth sa určuje ako šírka najužšieho pásma pri prenose dát z jednej stanice na inú. Pri komunikácii cez sieť sa neprenáša z jednej stanice na inú len 1 prúd dát, ale prenáša sa veľké množstvo dátových prúdov z jednej stanice na mnoho iných. Potenciálne každé zariadenie ktoré je v sieti, cez ktorú prebieha komunikácia medzi povodnými dvoma stanicami je schopné ovplyvniť bandwidth komunikačného kanálu tých dvoch staníc.

Komunikácia prostredníctvom TCP si sama reguluje maximálnu možnú rýchlosť ale UDP nie. Z toho vyplýva, že bandwidth je dôležitý pre výpočet optimálnej rýchlosti prenosu dát pri UDP komunikácii. Pretože Ak máme prenosovú rýchlosť vysokú, tak dostupná šírka prenosového pásma sa nám vyčerpá a UDP pakety sa začnú zahadzovať vo väčšom množstve v závislosti na dostupnej šírke prenosového pásma. Ak máme rýchlosť prenosu nízku, tak sa šírka prenosového pásma nezmenšuje, ale prenos nedosahuje maximálnu možnú rýchlosť bez stratovosti paketov a to môže mať zlý vplyv na aplikáciu čo príjma a spracováva UDP pakety napr. u streamovania videa by to mohlo spôsobiť sekanie videa.

Z tohoto dôvodu je veľmi dôležité zistiť aktuálnu šírku dostupného prenosového pásma a následne posielanie paketov takou rýchlosťou aby sa dostupný bandwidth nevyčerpával a aby aj rýchlosť bola čo najvyššia pre lepší užívateľský zážitok.

2 Moja implementácia

2.1 popis činnosti a obmedzenia

Moje riešenie meria prenosovú rýchlosť na server (upload).

Pre meter (klienta) mám vytvorné 2 triedy. Jednu pre spracovanie argumentov, druhú pre implementáciu klientovej časti programu. Pre reflector (server) mám taktiež 2 triedy s rovnakým významom ako má meter pre svoje 2 triedy.

V maine v module ipk-mtrip.cpp vytvorím meter alebo reflector na základe prvého prepínača programu a volám príslušnú funkciu, patriacu buď metru alebo reflektoru ktorá spracúva hlavnú časť celého programu pre meter alebo reflector.

Ďalej vytvorím na jednom porte 2 sockety pre UDP a TCP komunikáciu. UDP pre zasielanie dát pre meranie. TCP pre spoľahlivý prenos požadovaných dát pre výpočet, a príznakov pre správnu činnosť programu.

Pakety posielam z metra po streamoch reflektoru (inšpiroval som sa programom patload ktorý je v literatúre pre tento projekt)(1). Jeden stream obsahuje 100 paketov dĺžky n bytov ktoré si zvolil používateľ programu. V programe som obmedzil maximálnu veľkosť UDP dátovej časti paketu na 65507 bytov. Po každom pakete v každom streame mám nastavenú pauzu. Defaultne ju mám nastavenú na 10 milisekúnd. No túto dĺžku pauzy mením po každom streame a to buď zväčšujem o 60 nanosekúnd alebo znižujem o 50 nanosekúnd na základe odpovede od reflectora.

Reflector si počíta prijaté UDP pakety. Ak dostane od metra TCP paket tak zistí, či má meter zmenšiť alebo zväčšiť pauzy. A to na základe povolenej straty paketov. V mojom programe to je 5 percent. Následne odošle metru 5 bytovu TCP správu obsahujúcu na prvom byte prepínač či má meter zmenšiť intervaly pre pauzu ak je strata menej ako 5 percent alebo zväčšiť intervaly ak je viac. Zvyšné 3 byty obsahujú počet prijatých UDP paketov, ktoré následne meter využije na výpočet rýchlosti prenosu.

Meter počíta priemerné RTT na základe časového rozdielu medzi odoslaním TCP paketu z metra pre ukončenie aktuálneho streamu a odpovede od reflectora. Ako štandardnú odchýlku rátam štandardnú odchýlku rýchlosti.

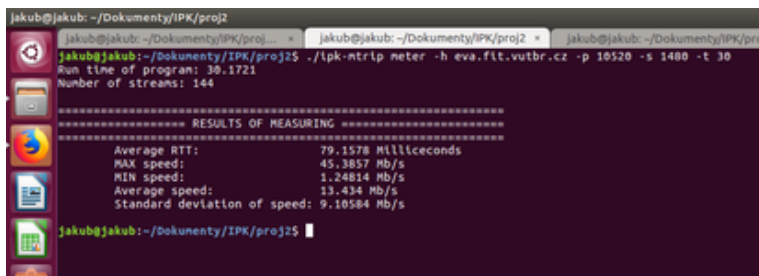
Meter ukončí svoju činnosť ak dĺžka behu programu pre aktuálne ukončený stream presiahne požadovanú dĺžku behu programu. Následne pošle meter reflektoru TCP správu s príznakom, že sa má ukončiť, dorátá všetky požadované hodnoty, vypíše ich a ukončí sa. Reflector sa po prijatí TCP správy s príznakom na ukončenie taktiež ukončí. Reflector je schopný obslúžiť len jedného metra (klienta). Pre ďalšiu obsluhu musí byť znova spustený.

3 Demonštrácia činnosti a testovanie

Program som testoval na internátovej wifi a na školských serveroch ku ktorým mám prístup. Testoval som meter aj reflector spustené na localhoste mojho PC a aj rozne kombiácie meter u mňa na PC reflector na serveri, naopak atd. Testovanie bolo celkom neobvyklé, pretože pre tento program neboli žiadne referenčné výstupy a ani nemohli byť, keďže meranie Banwidthu a všetkého ostatného spojeného s tým nadobúda v každej sieti iných hodnôt a aj v jednej sieti pre jedno meranie sa banwith môže v priebehu merania niekoľkokrát zmeniť.

3.1 Konkrétne príklady

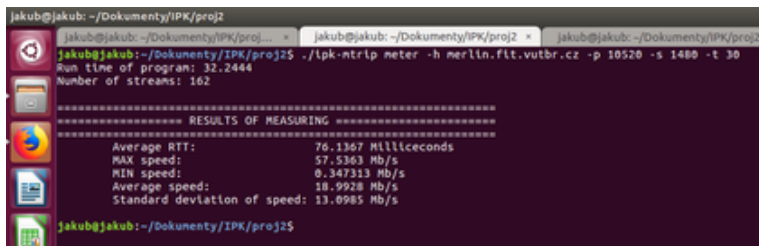
Reflektor umiestneny na eve a meter umiestneny na mojom PC



```
jakub@jakub: ~/Dokumenty/IPK/proj2
jakub@jakub:~/Dokumenty/IPK/proj2$ ./lpm-ntrip meter -h eva.ftt.vutbr.cz -p 10520 -s 1480 -t 30
Run time of program: 30.1721
Number of streams: 144

===== RESULTS OF MEASURING =====
Average RTT: 79.1578 milliseconds
MAX speed: 45.2857 Mb/s
MIN speed: 1.24814 Mb/s
Average speed: 13.434 Mb/s
Standard deviation of speed: 9.10584 Mb/s
jakub@jakub:~/Dokumenty/IPK/proj2$
```

Reflektor umiestneny na merlinovi a meter umiestneny na mojom PC



```
jakub@jakub: ~/Dokumenty/IPK/proj2
jakub@jakub:~/Dokumenty/IPK/proj2$ ./lpm-ntrip meter -h merlin.ftt.vutbr.cz -p 10520 -s 1480 -t 30
Run time of program: 32.2444
Number of streams: 162

===== RESULTS OF MEASURING =====
Average RTT: 76.1367 milliseconds
MAX speed: 57.5363 Mb/s
MIN speed: 0.347313 Mb/s
Average speed: 18.9928 Mb/s
Standard deviation of speed: 13.0985 Mb/s
jakub@jakub:~/Dokumenty/IPK/proj2$
```

Reference

- [1] M. Jain and C. Dovrolis, *Pathload: a measurement tool for end-to-end available bandwidth*, in Passive and Active Measurement Workshop, 2002.
- [2] Ryšavy, O. (2018). *Sítové aplikace* [PowerPoint prezentácia].
<https://wis.fit.vutbr.cz/FIT/st/course-files-st.php?file=%2Fcourse%2FIPK-IT%2Flectures%2FIPK2017L-02-APLIKACE.pdf&cid=11963>
- [3] Informácie na rozlíšenie prijatého TCP alebo UDP paketu
<https://stackoverflow.com/questions/27679809/how-to-use-both-tcp-and-udp-in-one-application-in-c>
- [4] Trieda pre časovanie
<https://gist.github.com/mcleary/b0bf4fa88830ff7c882d>