Vysoké učení technické v Brně
Fakulta informačních technologií

Počítačové komunikace a sítě - 2. projekt

Bandwidth Measurement

9. dubna 2018 Vojtěch Čurda (xcurda02)

Obsah

1 Algoritmy pro měření přenosové rychlosti

Přenosová rychlost (angl. *bandwidth*) značí jak velký objem informace se přenese za jednotku času. Jako jednotka se používá bit za sekundu. V rámci zadání tohoto projektu měříme dostupnou přenosovou rychlost sítě (angl. *available bandwidth*), nikoliv maximální možnou přenosovou rychlost, ktrou může spoj mezi dvěmi body v síti vyvinout (angl. *capacity*).

Jedním z principů měření dostupné přenosové rychlosti je tzv. SLoPS[**1**] (*Self-loading Periodic Streams*). Ten funguje na principu, kdy odesílatel posílá proud stejně velkých paketů v určité míře příjemci. Na základě jednosměrného zpoždění paketů se poté postupně upravuje míra odesílání, tak aby byla co nejblíže reálné dostupné rychlosti přenosu.

2 Implementace

2.1 Úvod

Ačkoliv existují ověřené varianty, jak lze měřit přenosovou rychlost mezi dvěma body v síti, rozhodl jsem se implementovat vlastní, jednodušší, ale také asi méně přesnější algoritmus.

2.2 Princip funkčnosti algoritmu

Prvek typu Meter v cyklu odesílá pakety dané velikosti reflektoru a ve stejném cyklu zároveň pakety přijímá. Do každého paketu je vkládána informace o času odeslání daného paketu. Prvek typu Reflektor po přijetí paketu neposílá zpět celý paket, ale pouze dříve vloženou informaci o času odeslání paketu. Podle počtu navrácených paketů počítá Meter dostupnou přenosovou rychlost. Algoritmus tedy měří dostupnou přenosovou rychlost na cestě od Meteru k Reflektoru.

2.3 Bližší pohled na implementaci algoritmu

2.3.1 Přenos paketů

Posílání paketů je realizováno funkcemi `sendto()` a `recvfrom()`, kdy na straně Meteru je přijímání nastaveno na neblokující, resp. blokující po nejmenší možný interval ($1\mu s$), ke zpřesnění výsledků. Tento blokující inteval se vkládá do nastavení socketu funkcí `setsockopt()`, resp. flagem `SO_RCVTIMEO`. Časový záznam získaný knihovní funkcí `gettimeofday()` je do paketu přibalován z důvodů přesného změření RTT.

2.3.2 Rozdělení měření do jednotlivých běhů

Měření začíná po třech sekundách posílání paketů z důvodu velmi nepřesných výsledků během tohoto prvotního intervalu. Čas zadaný uživatelem není do tohoto intervalu započítán.

Poté probíhá samotné měření, kdy se každou uplynulou sekundu nashromážděné informace jako počet poslaných paketů, celkový RTT a počet přijatých bytů přepočítají na průměrnou rychlost přenosu a průměrný RTT z tohoto běhu. Tyto záznamy se uchovávají až do konce samotného měření, kdy se z nich určí celkové statistiky všech běhů, které se vypisují na konci intervalu, po který měření probíhá, nebo při předčasným ukončením programu.

2.4 Ukázka činnosti programu

Následující obrázek demonstruje činnost programu při zadaných následujících parametrech a při spuštěném programu Reflektoru na školním serveru Merlin.

Host:	merlin.fit.vutbr.cz
Port:	10121
Velikost sondy:	4000
Čas měření:	15

```
Reflector: merlin.fit.vutbr.cz:10121    Probe size: 4000    Time: 15
------------------------------------------------------------------------------
   1: RTT:  23.422 ms   bandwidth: 38.048 Mb/s
   2: RTT:  25.527 ms   bandwidth: 39.328 Mb/s
   3: RTT:  16.876 ms   bandwidth: 46.048 Mb/s
   4: RTT:  17.002 ms   bandwidth: 42.432 Mb/s
   5: RTT:  25.115 ms   bandwidth: 31.552 Mb/s
   6: RTT:  20.543 ms   bandwidth: 49.344 Mb/s
   7: RTT:  17.375 ms   bandwidth: 35.328 Mb/s
   8: RTT:  21.293 ms   bandwidth: 36.416 Mb/s
   9: RTT:  20.069 ms   bandwidth: 45.664 Mb/s
  10: RTT:  19.217 ms   bandwidth: 45.28 Mb/s
  11: RTT:  33.486 ms   bandwidth: 33.44 Mb/s
  12: RTT:  27.219 ms   bandwidth: 50.592 Mb/s
  13: RTT:  20.891 ms   bandwidth: 40.096 Mb/s
  14: RTT:  20.085 ms   bandwidth: 49.824 Mb/s
  15: RTT:  21.883 ms   bandwidth: 46.048 Mb/s
----------------------- Bandwidth measurement statistics ----------------------
Avg bandwidth : 41.9627 Mb/s, Avg RTT: 21.8169 ms
Min bandwidth: 31.552 Mb/s,  Max bandwidth: 50.592 Mb/s,  stdev: 5.94759 Mb/s
```

Výsledek měření po omezení rychlosti uploadu na 1Mb/s na lokálním stroji a při stejných spouštěcích parametrech:

```
Reflector: merlin.fit.vutbr.cz:10121    Probe size: 4000    Time: 15
------------------------------------------------------------------------------
   1: RTT: 1076.64 ms   bandwidth: 0.992 Mb/s
   2: RTT:  1084.1 ms   bandwidth: 1.056 Mb/s
   3: RTT: 1076.53 ms   bandwidth: 0.992 Mb/s
   4: RTT: 1067.18 ms   bandwidth: 1.024 Mb/s
   5: RTT: 1073.04 ms   bandwidth: 0.992 Mb/s
   6: RTT: 1076.14 ms   bandwidth: 0.992 Mb/s
   7: RTT: 1089.51 ms   bandwidth: 1.024 Mb/s
   8: RTT:  1095.6 ms   bandwidth: 0.992 Mb/s
   9: RTT: 1063.05 ms   bandwidth: 1.056 Mb/s
  10: RTT: 1085.99 ms   bandwidth: 0.992 Mb/s
  11: RTT: 1080.94 ms   bandwidth: 1.024 Mb/s
  12: RTT: 1068.23 ms   bandwidth: 0.992 Mb/s
  13: RTT: 1052.47 ms   bandwidth: 1.024 Mb/s
  14: RTT:  1081.7 ms   bandwidth: 0.992 Mb/s
  15: RTT: 1107.04 ms   bandwidth: 1.024 Mb/s
----------------------- Bandwidth measurement statistics ----------------------
Avg bandwidth : 1.0112 Mb/s, Avg RTT: 1078.51 ms
Min bandwidth: 0.992 Mb/s,  Max bandwidth: 1.056 Mb/s,  stdev: 0.0227778 Mb/s
```

2.5 Nevýhody algoritmu

Algoritmus není optimální jelikož neměří přesně za každých okolností. K získání reálné dostupné přenosové rychlosti totiž potřebuje posílat dostatečně velké sondovací pakety. Pokud jsou příliš malé, nedokáže vyvinout dostatečně velkou přenosovou rychlost. Na druhou stranu pokud se používají větší sondovací pakety, než je potřeba, síť se postupně zahlcuje a díky tomu RTT jednotlivých paketů postupně roste.

Odkazy

[1] C. Dovrolis, R. Prasad, M. Murray, and k. claffy, *"Bandwidth estimation: metrics, measurement techniques, and tools"*, IEEE Network, vol. 17, no. 6, pp. 27–35, Apr 2003