

A PROGRAMOZÁS ALAPJAI 2.

HÁZI FELADAT DOKUMENTÁCIÓ

Adatcsomagok statisztikája

KÉSZÍTETTE: LOVÁSZI MÁRTON, IKQOC2
lovaszi.marton@edu.bme.hu

2025. 05. 05.

TARTALOMJEGYZÉK

Felhasználói dokumentáció	3
Osztályok statikus leírása.....	4
IP_PACKET	4
Felelőssége.....	4
Attribútumok	4
Metódusok.....	4
PRIORITY_IP_PACKET	5
Felelőssége.....	5
Ősosztályok.....	5
Attribútumok	5
Metódusok.....	5
IP_STATISTIC	5
Felelőssége.....	5
Attribútumok	5
Metódusok.....	6
UML osztálydiagramm	7
Összegzés	8
Mit sikerült és mit nem sikerült megvalósítani a specifikációból?	8
Mit tanultál a megvalósítás során?	8
Továbbfejlesztési lehetőségek.....	8
Képernyőképek a futó alkalmazásról.....	9

Felhasználói dokumentáció

Az alkalmazás használatához kell egy log fájl, amely tartalmazza a router adatait. Ebből fogja a program kiolvasni a statisztikákat. Ha nincsen log fájl, akkor a program nem tud működni. A log fájlnak abban a mappában kell lennie, amelyben a program van. A futtatható fájlt elindítva egy konzol ablak jelenik meg amely arra kér, hogy adjuk meg a router log fájljának a nevét, ezzel lehetővé teszi, hogyha van több log fájl a mappában akkor ki tudjuk választani, hogy melyiket szeretnénk használni. A fájlnev megadása után 6 darab opciót kínál fel a program. Az 1-4-ig opciók arra vannak, hogy egyenként külön-külön tudjuk lefuttatni a specifikációban szereplő 4db statisztikai funkciót. Az 5. opció az összes futtatása, ilyenkor a program végigmegy az összes funkción és nem kell egyenként lefuttatni. A 6. opció a programból való kilépés. A választott lehetőség megadását az opció számának a konzolra írásával tudjuk megtenni. Több opciónál kell megadni utólagosan további információkat pl: egy tetszőleges feladó-címzett IP cím pár megadása szükséges az első opciónál mivel e között a két IP cím között fog átlagos sávszélességet számolni. Ezeket szintén a konzolon lehet megadni. Ha befejeztük a statisztikák futtatását és kilépünk a programból a 6. opció választásával akkor az összes statisztika elmentődik egy AnalyzerLog.txt fájlba. Fontos, hogy elmentsük egy másik fájlba a tartalmát mivel ez a fájl minden futtatás elején törlődik és csak az adott futtatás adatait tárolja el. A statisztikákat nem csak az AnalyzerLog.txt fájlba menti el a program, hanem az adott funkció futtatása után kiírja a program a konzolra is az eredményt.

A router log fájljának követnie kell formai mintákat:

```
102.45.187.23 87.162.34.10 00:00:00 00:00:03 421 2
```

Az adatok szóközzel vannak elválasztva. A példa sorban az első két elem a feladó és a címzett IP címe. A harmadik és a negyedik elem a beérkezés és a tovább küldés időpontja. Az ötödik elem a csomag mérete Mb-ben. A hatodik elem pedig a csomag prioritása. Ha a prioritás 0 akkor nem prioritásos csomag.

A programban ha például az 1-es opciót választjuk akkor az választott szám beírása után be kell írunk a két IP címet. Majd a program pedig szöveges válasz formájában kiírja a statisztikát

példa az alábbi bemenet-kimenet párra:

Bemenet:

- 1)Átlagos sávszélesség számítása egy adott forrás-cél parra
- 2)Adott prioritású csomagok átlagos érkezési időközének számítása
- 3)Az összes csomagból adódó sávszélesség számítása
- 4)Az átvitt prioritásos csomagok adatmennyiségének aránya az összes adathoz képest
- 5) Az összes funkció futtatása
- 6) kilépés

A választott opció száma: 1

Adjon meg egy kereséshez használandó forrás-cél ip cím párt

A forrás ip cím: 102.45.187.23

A cél ip cím: 87.162.34.10

Kimenet:

Az átlagos sávszélesség az adott forrás-cél párra: 145.167Mbps

Ez a szöveges válasz pedig megtalálható lesz az AnalyzerLog.txt fájlban is a következő alkalomig ameddig újra nem indítjuk a programot.

Osztályok statikus leírása

IP_PACKET

Felelőssége

Ennek az osztálynak az a dolga, hogy reprezentáljon egy IP csomagot a beolvasott router log fájl egy sora mentén. Ez az osztály tárolni fogja a csomagok összes információját.

Attribútumok

Privát

- SENDER_IP string: Ez a változó a csomag küldőjének IP címét rögzíti
- RECEIVER_IP string: Ez a változó a csomag címzettjének IP címét rögzíti
- TIME_OF_RECEIVING int: Ez változó a csomag fogadásának egy időpontját tárolja el int formában mivel ezzel könnyebb lesz később számolni.
- TIME_OF_SENDING int: Ez a változó a csomag tovább küldésének időpontját rögzíti szintén int formátumban.
- P_SIZE int: az adott csomag méretét rögzíti Mb-ben.

Metódusok

Publikus

- IP_PACKET() void : Konstruktor az IP csomaghoz
- ~IP_PACKET() virtual void : Destruktor az IP csomaghoz

Getter függvények amelyekkel hozzá lehet férni a privát tagváltozók értékéhez:

- getSenderIp(): string
- getReceiverIp(): string
- getTimeOfSending(): int
- getPsize(): int
- PrintInfo() virtual void : Kiírja a csomag adatait.
- GetSecondsInRouter() virtual int : Megmutatja hogy hány másodpercet töltött a routerben, azaz mennyi idő telt el a beérkezés és küldés között. Ez azért szükséges, hogy a statisztikai számolásoknál ne kelljen kódot ismételni.

PRIORITY_IP_PACKET

Felelőssége

Ez az osztály prioritási értékkel rendelkező IP csomagokat valósít meg.

Ősosztályok

Az IP_PACKET osztály leszármazottja ez az osztály mivel minden tulajdonsága megegyezik egy általános IP csomaggal, csak van egy plusz prioritási értéke.

Attribútumok

Privát

- PRIORITY_RATING int: A csomag prioritási besorolását tárolja

Metódusok

Publikus

- PRIORITY_IP_PACKET() void: Az osztály konstruktora
- ~PRIORITY_IP_PACKET() void: Az osztály destruktora
- getPriorityRating() int : A prioritási érték getter függvénye
- Printinfo() void: Kírja a csomag adatait
- GetSecondsInRouter() int: Megmutatja hogy hány másodpercet töltött a routerben, azaz mennyi idő telt el a beérkezés és küldés között. Ez azért szükséges, hogy a statisztikai számolásoknál ne kelljen kódot ismételni.

IP_STATISTIC

Felelőssége

Ez az osztály a statisztikai függvényeket valósítja meg. Fő célja annak hogy osztály formában vannak, hogy jobban átláthatóbb legyen a kód, és el legyenek szeparálva a futtatáshoz szükséges UI függvények és számolásokat csináló statisztikai függvények.

Attribútumok

privát

- Data std::vector<IP_PACKET*> : Ebben a vektorban a az eltárolt csomagok szerepelnek, heterogén kollekciót valósít meg mert PRIORITY_IP_PACKET elemek is vannak benne. Ezeken az elemeken lesznek lefuttatva a statisztikai függvények.
- AvarageBandwith_SENDER_RECEIVER float: A kiszámolt átlag sávszélesség az adott IP cím pár között van benne eltárolva
- AvarageArival_PRIORITY float: A kiszámolt átlag érkezési idő az adott prioritású csomagok között van benne eltárolva
- Bandwith_ALL float : Az összes adatból származó sávszélességet tárolja

- Ratio_PRIO_NON_PRIO float: A prioritásos csomagok adatmennyiségének aránya az összes adathoz képest van benne tárolva

Metódusok

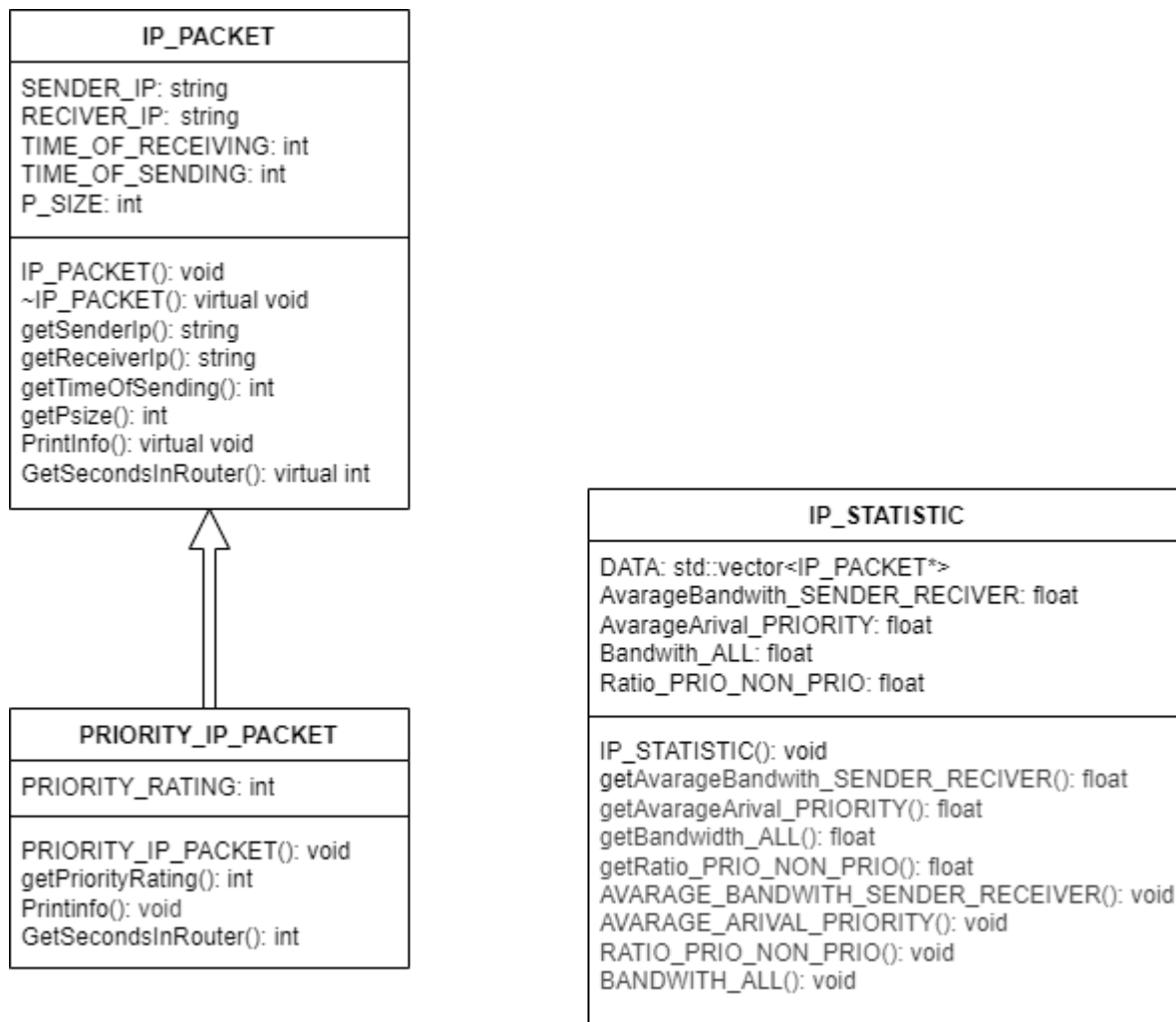
Publikus

- IP_STATISTIC() void : Az osztály konstruktora

Getter függvények, amelyekkel hozzá lehet férni a privát tagváltozók értékéhez:

- getAvarageBandwith_SENDER_RECEIVER() float:
- getAvarageArival_PRIORITY(): float
- getBandwidth_ALL(): float
- getRatio_PRIO_NON_PRIO(): float
- AVERAGE_BANDWIDTH_SENDER_RECIVER() void: Megállapítja az átlag sávszélességet egy adott küldő-cél IP cím párra
- AVERAGE_ARRIVAL_PRIORITY() void: Megállapítja az átlag érkezési időközét egy adott prioritású csomag csoportnak
- RATIO_PRIO_NON_PRIO() void: Megállapítja a prioritásos csomagok adatmennyiségének arányát az összes adathoz képest
- BANDWIDTH_ALL() void: Megállapítja az összes adatból származó sávszélességet

UML osztálydiagramm



Összegzés

Mit sikerült és mit nem sikerült megvalósítani a specifikációból?

A specifikációból mindent sikerült megvalósítanom amit elterveztem.

Mit tanultál a megvalósítás során?

Amit a legnagyobb tanulságnak éltem meg, hogy mennyire fontos az, hogy legyen egy átlátható terv mielőtt nekikezdek a feladatnak. Habár a házi feladat leírás nem kérte specifikusan, de készítettem magamnak folyamatábrákat a függvények működéséről és az algoritmusok kivitelezéséről melyek után már csak szinte le kellett fordítanom c++ nyelvre a jegyzeteimet és sokkal kevesebbet kellett debuggolnom mert jobban átláttam a program működését. Persze voltak utána dolgok, amiket menet közben kellett megoldanom viszont azok már nem voltak akkora kihívások. A gyengeségem, amit még gyakorolnom kell az a jobban értelmezhető és konzisztens változónév adás

Továbbfejlesztési lehetőségek

Szerintem ezt a programot könnyen tovább lehetne fejleszteni mivel statisztikai adatokat számol és abból bőven van még, amit hozzá lehetne adni.

Képernyőképek a futó alkalmazásról

```
IP packet Analyzer

Adja meg az elemzeshez hasznalt fajl nevet.
log.txt

Az elemzas elkezdeshez valasszon az alabbi opciok kozul!(1-6):
1)Atlagos savszelesseg szamitasa egy adott forras-cel parra
2)Adott prioritasu csomagok atlagos erkezesi idokozenek szamitasa
3)Az osszes csomagbol adodo savszelesseg szamitasa
4)Az atvitt prioritasos csomagok adatmennyisegenek aranya az osszes adathoz kepest
5) Az osszes funkcio futtatasa
6) kilepes
A valasztott opcio szama: |
```

```
A valasztott opcio szama: 5

Adjon meg egy kereseshez hasznalando forras-cel ip cim part
A forras ip cim: 102.45.187.23

A cel ip cim: 87.162.34.10
Adjon meg egy prioritasi erteket(1-3): 2

-----
Az atlagos savszelesseg az adott forras-cel parra: 145.167Mbps
Az adott prioritasu csomagok atlagos erkezesi idokoze:32.25 masodperc
Az osszes adatbol adodo savszelesseg: 6811.33Mbps
Az osszes adat 71.5817% szazleka a prioritasos csomagok adata
-----

IP packet Analyzer

Az elemzas elkezdeshez valasszon az alabbi opciok kozul!(1-6):
1)Atlagos savszelesseg szamitasa egy adott forras-cel parra
2)Adott prioritasu csomagok atlagos erkezesi idokozenek szamitasa
3)Az osszes csomagbol adodo savszelesseg szamitasa
4)Az atvitt prioritasos csomagok adatmennyisegenek aranya az osszes adathoz kepest
5) Az osszes funkcio futtatasa
6) kilepes
A valasztott opcio szama: |
```