

Technická univerzita v Košiciach
Fakulta elektrotechniky a informatiky
Katedra počítačov a informatiky

**Implementácia algoritmov vzorkovania pre potrebu
neintruzívnych meraní**

Vedúci diplomovej práce:
Ing. František Jakab

Diplomant:
Martin Hronský

Konzultant diplomovej práce:
Ing. František Jakab

Košice 2005

Čestné prehlásenie

Prehlasujem, že som diplomovú prácu vypracoval samostatne s využitím uvedenej odbornej literatúry.

V Košiciach dňa 15.4.2005

.....
vlastnoručný podpis

Pod'akovanie

Ďakujem Ing. Františkovi Jakobovi, celému tímu projektu Basicmeter a členom Laboratória počítačových sietí za cenné rady, pripomienky a odbornú pomoc pri tvorbe diplomovej práce.

Názov práce : Implementácia algoritmov vzorkovania pre potrebu
neintruzívnych meraní

Katedra : Katedra počítačov a informatiky, TU FEI Košice

Autor : Martin Hronský

Vedúci DP : Ing. František Jakab

Konzultant DP : Ing. František Jakab

Dátum : 15.4.2005

Kľúčové slová : Vzorkovanie, filtrovanie, systematické vzorkovanie,
náhodné vzorkovanie, počítačová sieť, internet, protokol,
dátový tok, paket, identifikátor, IP, PSAMP, pasívne
meranie, QoS parametre

Anotácia : Prezentovaná práca sa zaoberá problematikou
vzorkovania paketov v neintruívnych meraniach so
zameraním na podporu vznikajúceho štandardu PSAMP.
Riešenie pozostáva z návrhu a implementácie časti
odchyťavajúcej sieťovej prevádzky, realizujúcu filtrovanie
a vzorkovanie v základnom nástroji pre meranie QoS
parametrov .

Thesis title : Implementation of sampling algorithms for nonintrusive measurement

Department : Department of Computers and Informatics, TU FEI Košice

Author : Martin Hronský

Supervisor : Ing. František Jakab

Tutor : Ing. František Jakab

Date : 15.4.2005

Keywords : Sampling, Filtering, systematic sampling, random sampling, computer network, internet, protocol, flow, packet, identifier, IP, PSAMP, passive measurement, QoS parameter

Annotation : Presented thesis deals with problematic of sampling algorithms in passive measurement of QoS parameters in computer networks. In particular, we focus on the new emerging PSAMP standard. The design and implementation of filtering and sampling of basicmeter was introduced.

Obsah

1. Úvod.....	1
1.1 Formulácia	1
1.2 Motivácia úlohy	1
2. Analýza problematiky neintruzívnych meraní pre potreby vyhodnocovania kvality služieb.....	4
2.1 Pasívne metódy s jedným meracím miestom.....	4
2.2 Pasívne metódy s dvoma meracími bodmi	5
2.3 Umiestnenie meracieho zariadenia.....	7
2.4 Synchronizácia hodín	8
2.5 Odchytávanie paketov	9
2.6 Generovanie časových známk.....	9
2.7 Filtrovanie.....	10
2.8 Vzorkovanie.....	10
2.9 Generovanie identifikátora paketu	11
Výber polí hlavičky IP paketu.....	11
2.10 Prenos nameraných údajov	14
3. Architektúra nástroja pre neintruzívne merania („BasicMeter“).....	16
3.1 Špecifikácia požiadaviek	16
3.2 Koncepcia meracieho nástroja	17
3.3 Návrh meracieho nástroja	19
3.4 Architektúra meracieho nástroja	21
4. Koncepcia vzorkovania prevádzky v sieťových infraštruktúrach	23
4.1 Terminológia.....	23
4.2 Techniky výberu paketov	29
4.2.1 Filtrovanie.....	29
4.2.2 Vzorkovanie.....	31
4.2.3 Šablóny.....	33
4.3 Popis architektúry vzorkovania PSAMP	33
4.3.1 Proces selekcie.....	34
4.3.2 Proces exportu a oznamovací proces	34
4.3.3 Formát záznamu.....	36
4.4 Požiadavky na vzorkovanie	37
4.5 Popis parametrov opisujúcich vzorkovanie	39
4.6 Bezpečnostné riziko vzorkovania.....	41
5. Vzorkovacie algoritmy, ich vlastnosti a komparácia	42

5.1	Systematické vzorkovanie	42
5.2	Náhodné vzorkovanie	44
5.2.1	n-z-N vzorkovanie	45
5.2.2	Pravdepodobnostné vzorkovanie	45
5.2.3	Uniformné pravdepodobnostné vzorkovanie	45
5.2.4	Neuniformné pravdepodobnostné vzorkovanie	46
5.2.5	Neuniformné vzorkovanie závislé na stave toku	46
5.3	Štatistické modely vhodné pre popis a optimalizáciu vzorkovacích metód.....	47
5.3.1	Počiatkové podmienky	47
5.3.2	Vzorkovanie n-z-N.....	48
5.3.3	Pravdepodobnostné vzorkovanie	49
5.3.4	Systematické vzorkovanie	50
5.3.5	Porovnanie efektivity vzorkovania n-z-N a pravdepodobnostného vzorkovania.....	51
5.4	Porovnanie vzorkovacích algoritmov.....	51
6.	Implementácia vybraných vzorkovacích algoritmov pre potreby nástroja „Basicmeter“	57
6.1	Algoritmus spracovania paketu.....	57
6.2	Knižnica Libpcap	58
6.3	Požiadavky kladené na jednotlivé časti.....	59
6.4	Návrh a analýza časti capture.....	60
6.5	Návrh a analýza časti sample.....	61
	Analýza vstupných dát	61
	Analýza výstupných dát	61
6.6	Experimentálne overenie funkčnosti nástroja a výsledkov vzorkovania..	61
6.6.1	Inštalácia.....	61
6.6.2	Experimentálne meranie v laboratórnom segmente.....	62
6.6.3	Meranie využitia šírky pásma.....	62
7.	Zhodnotenie riešenia	71
8.	Zoznam použitej literatúry	72
9.	Zoznam príloh.....	75
10.	Zoznam obrázkov a tabuliek.....	76

1. Úvod

1.1 Formulácia

Cieľom diplomovej práce je analyzovať problematiku vzorkovacích algoritmov pri neintruzívnych meraniach parametrov kvality služieb (Quality of Service, QoS) v počítačových sieťach založených na protokole IP a implementovať podporu pre vzorkovanie do meracieho nástroja „Basic Meter“.

Hlavnou úlohou vzorkovania je zvoliť reprezentatívnu vzorku paketov ktorá bude popisovať charakteristiku pôvodnej nevzorkovanej prevádzky.

V rámci práce budú podrobne analyzovaný existujúci štandard[14] a algoritmy pre podporu vzorkovacích metód merania QoS parametrov v počítačových sieťach. V analýze bude pozornosť venovaná pasívnym meraniam, ktoré sú na rozdiel od aktívnych meraní založené na existencii sieťovej prevádzky v sieti.

Algoritmy budú po fáze analýzy navrhnuté a následne implementované v module jednoduchého nástroja pre realizáciu pasívnych meraní s podporou vznikajúceho štandardu vzorkovania paketov Paket Sampling [14] a nadväzujúceho exportovacieho protokolu Internet Protocol Flow Information Export [8] a následne budú realizované experimentálne merania v laboratórnych podmienkach.

1.2 Motivácia úlohy

Existujú dve hlavné príčiny nárastu požiadavok pre meranie prevádzkových parametrov v počítačových sieťach. Prvou je to, že rýchlosti v počítačových sieťach sa zvyšujú zároveň s potrebou túto prevádzku merať, čím vzrastá veľkosť nameraných údajov. Druhou príčinou je, že nárast prevádzky je sprevádzaný aj kvalitatívnym zvýšením požiadaviek na prevádzkové parametre QoS. Zariadenia ako smerovače, ktoré vykonávajú tieto merania, potrebujú sofistikované služby na monitorovanie prevádzky, čo zahŕňa zachytávanie hlavičky paketov a eventuálne aj časti samotného tela paketu, filtrovanie, následné vzorkovanie a klasifikáciu týchto paketov na jednotlivé toky. Všetky tieto faktory môžu viesť k obrovskému množstvu nameraných údajov, ktoré

budú vyžadovať vysoké požiadavky na výpočtovú kapacitu samotného merania, ukladania, transportu údajov a ich spracovania.

Nepretržité zachytávanie kompletnej sieťovej prevádzky plnou linkovou rýchlosťou (100Mbit/s až 10Gbit/s) môže byť vykonávané špecializovanými hardvérovými meračmi. Avšak cena týchto zariadení spolu s infraštruktúrou podporujúcou meranie je značne vysoká. Na miesto použitia drahého špecializovaného hardvéru sa v mieste meracieho bodu aplikuje forma redukcie meraných údajov, či už pomocou filtrovania, vzorkovania alebo ich kombinácie.

Diplomová práca sa zaoberá analýzou, popisom a špecifikáciou existujúcich vzorkovacích metód pre umožnenie vykonávania neintruzívnych meraní prevádzkových parametrov vo vysoko rýchlostných počítačových sieťach.

Práca je štruktúrovaná nasledujúcim spôsobom. Prvá kapitola podrobne formuluje úlohu a cieľ práce. Druhá kapitola sa venuje analýze problematiky neintruzívnych meraní pre potreby vyhodnocovania parametrov kvality služieb. Táto kapitola sa podrobne venuje pasívnym metódam meraní, zaoberá sa analýzou problémov vznikajúcich pri vykonávaní pasívnych meraní a navrhuje možné riešenia. Tretia kapitola sa podrobne venuje architektúre nástroja pre neintruzívne merania. Obsahom štvrtej kapitoly je koncepcia vzorkovania prevádzky v sieťových infraštruktúrach v rámci novovznikajúceho štandardu PSAMP a jeho nadväznosti na IPFIX. Piata kapitola sa podrobne venuje vlastnostiam vzorkovacích algoritmov a ich vhodnosti pre implementáciu na základe použitia metód matematickej štatistiky. Posledná kapitola sa zaoberá popisom koncepcie, návrhu, implementácie vzorkovacích algoritmov a filtrovania v nástroji a experimentov, ktoré majú za úlohu overiť funkčnosť nástroja.

Pri návrhu, plánovaní a aj zabezpečovaní komunikačných služieb sa okrem základnej požiadavky spoľahlivej konektivity čoraz väčší dôraz kladie aj na zabezpečenie určitej, požadovanej úrovne kvality služieb. Úroveň kvality služieb je definovaná hodnotami prevádzkových parametrov kvality služieb, preto je potrebné vypracovať a navrhnúť metódy pre zabezpečenie kvality služieb. Pre sledovanie a monitorovanie prevádzkových parametrov počítačovej siete existuje viacero dôvodov.

Mnoho aplikácií, najmä aplikácií zaoberajúcich sa prenosom obrazových a zvukových informácií, je citlivých na dodržanie určitej úrovne parametrov prevádzky — týka sa to najmä oneskorenia, jednosmerného aj spätného. Pojmy týkajúce sa kvality služieb sa stávajú predmetom zmluvných vzťahov medzi poskytovateľmi služieb a zákazníkmi. Zákazníkov zaujíma, či hodnoty prevádzkových parametrov, zmluvne dohodnuté v Service Level Agreement (SLA), sú aj skutočne dodržiavané.

Výstupy z meraní prevádzkových parametrov počítačových sietí sa často používajú pre ďalšie optimalizácie topológie sietí, ich profylaktiku a plánovanie ďalších rozšírení.

2. Analýza problematiky neintruzívnych meraní pre potreby vyhodnocovania kvality služieb

Meracie metódy môžu byť začlenené do nasledujúcich kategórií: pasívne alebo neintruzívne, semi-aktívne a aktívne. Pasívne merania parametrov kvality služieb sú založené na princípe merania existujúcej prevádzky v sieti. Ich výhodou je to, že negenerujú ďalšiu prevádzku a ani ju nijako neovplyvňujú. Na proti tomu semi-aktívne merania využívajú existujúcu prevádzku v sieti, avšak jednotlivé pakety modifikujú pridaním napr. časovej značky k paketu. Aktívne metódy generujú vlastnú prevádzku na testovanie parametrov QoS.

2.1 Pasívne metódy s jedným meracím miestom

Merania založené na objeme prenesených dát

Pasívne metódy s jedným meracím bodom sa používajú najviac pre merania objemových charakteristík prevádzky dát. Táto charakteristika je popisovaná parametrami šírka pásma a rýchlosť prenosu paketov.

Spôsob výpočtu týchto charakteristík je nasledovný:

$$\text{Rate} = \text{Count} / \text{Time}$$

kde Rate je požadovaný výsledný parameter, Count je počet prenesených jednotiek za čas Time. Count môže byť buď počet bitov alebo počet paketov.

Merania založené na časových charakteristikách

Pomocou merania v jednom bode môžeme jednoducho merať celkové spiatocné oneskorenie(RTT). Podstatou metódy je fakt, že poznáme pakety na ktoré cieľové zariadenia odpovedajú ihneď (napr. TCP-SYN/SYN-ACK, DNS žiadosť/odpoveď, ICMP echo request/ICMP echo relpy). Pri odoslaní paketu zo žiadosťou si poznamenáme jeho identifikátory a čas jeho zachytenia a pri odpovedi na tento paket rozoznaného ako odpoveď na žiadosť si poznačíme jeho čas zachytenia. Následne ak odčítame tieto časy tak dostaneme čas uzavretej slučky (RTT).

Pretože v uzavretej slučke slučka môže byť asymetrická [22] a jednotlivé cesty majú obyčajne rôzne charakteristiky, nemôže byť RTT jednoducho použitá pre výpočet oneskorenia jedným smerom s veľkou presnosťou. Napriek tomu kvôli kontrole SLA môže RTT poskytnúť prvú, veľmi skreslenú informáciu o oneskorení prevádzky v sieti. Napr. ak RTT zostáva pod požadovaným oneskorením v jednej ceste potom je zabezpečené, že oneskorenie v tejto jednej ceste je v dohodnutých hraniciach.

2.2 Pasívne metódy s dvoma meracími bodmi

Pomocou využitia dvoch meracích bodov môže byť meraná variácia oneskorenia, jednosmerné oneskorenie a strata paketov. Podstatou týchto meraní je porovnávanie časových známk tých istých paketov. Jednou z kľúčových metód pasívneho merania oneskorenia je umiestnenie meracích bodov a metóda rozpoznania paketov. Pasívne metódy vôbec neovplyvňujú prevádzku. Z toho vyplýva, že nie je generovaná žiadna testovacia premávka a pakety nie sú oneskorené a ani modifikované. Rozpoznávanie paketov musí byť realizované na základe údajov, ktoré už paket obsahuje. Podrobnejšie sa generovaniu paket ID venuje kapitola 2.9.

Všeobecná architektúra merania oneskorenia

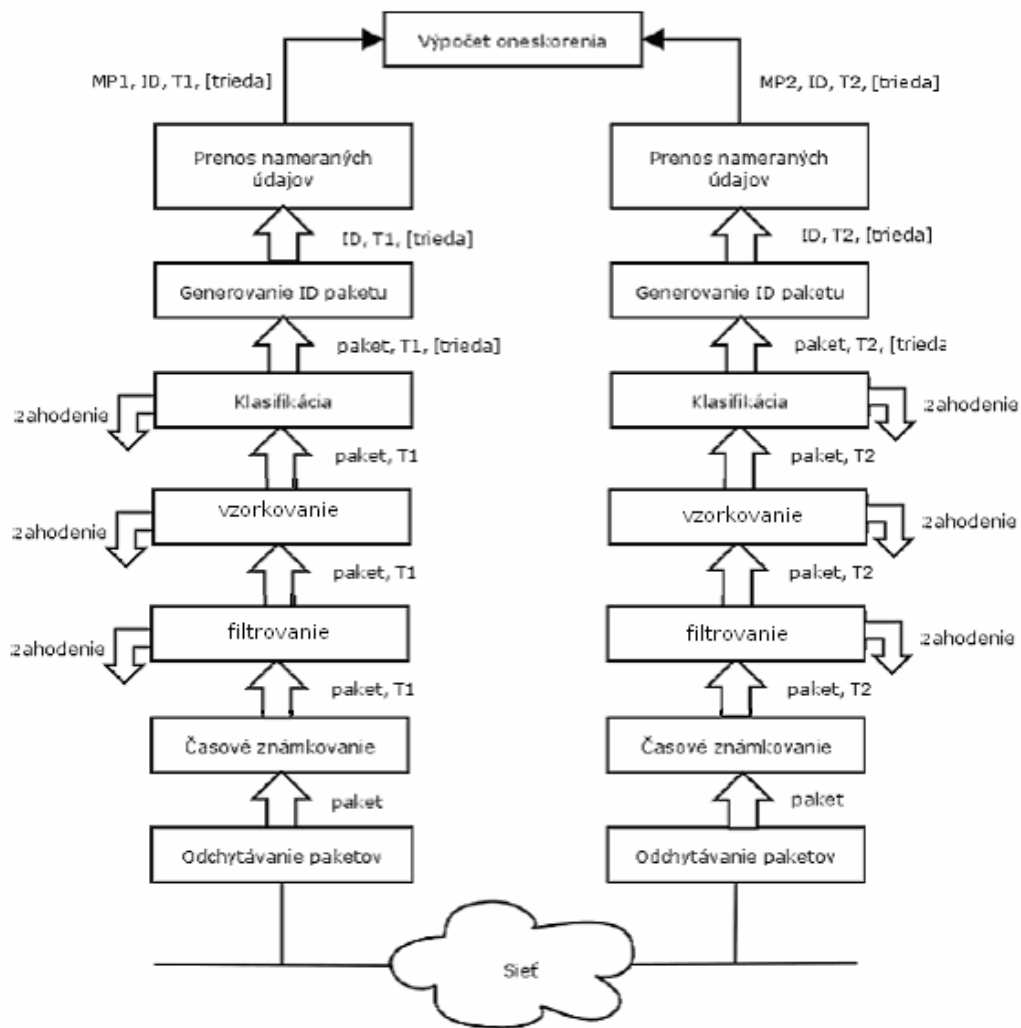
Spôsob realizácie pasívneho merania časových charakteristík je založený na generovaní časových značiek a identifikátora (ID) paketu pre každý paket prechádzajúci cez dve meracie miesta a odoslanie informácie o časovej značke a identifikátore do miesta kde je vypočítané oneskorenie alebo iný parameter. Identifikátor paketu je nutný kvôli prideleniu časových známk z rôznych meracích bodov k prisluchajúcemu paketu. Pre každý žiadaný meraný paket je potrebné vygenerovať paket ID a časovú známku, a odoslať tieto údaje do zberného miesta, kde sa urobí výpočet parametrov oneskorenia na základe výsledkov z rôznych bodov merania. Veľkosť nameraných výsledných údajov, ktoré vznikajú za jednotku času závisí od množstva meraných paketov n za sekundu, počtu bitov $l(t)$ použitých pre reprezentáciu časovej známky a počtu bitov $l(id)$ použitých pre reprezentáciu ID paketu. Vyššie spomínaný zberný bod

nemusí byť nutne fyzicky rôzny od bodu merania. Môže byť umiestnený v jednom meracom bode, najmä za účelom redukcie réžie prenášania nameraných dát, samozrejme ak na zberný bod máme v meracom bode voľné výpočtové prostriedky.

Zodpovedajúce procesy sú odchytyvanie paketov, filtrovanie, vzorkovanie, časové známkovanie, generovanie ID paketu, klasifikácia a prenos nameraných údajov. Požiadavky pre tieto bloky sú analyzované v nasledujúcich častiach. Popis architektúry merania časových charakteristík je na obr.2.1. Ďalším dôležitým problémom, ktorým je potrebné sa zaoberať je problém súkromia, keď odchytyvame prevádzku klientov, ošetrovanie potrebných úprav ak sú pakety stratené alebo duplikované. Ďalším problémom, ktorý je nutné riešiť v prípade použitia viac ako jedného meracieho miesta je synchronizácia hodín medzi všetkými meracími bodmi. Očakávané jednosmerné oneskorenie IP paketu medzi dvoma meracími bodmi MP1 a MP2 je rozdiel času príchodu paketu do meracích bodov:

$$\Delta T = T_1 - T_2$$

Súvislosť medzi dvoma časovými známami rovnakého IP paketu je daná prostredníctvom ID paketu a parametru ΔT . Ak paket s daným ID je odchytený v jednom bode ale nie je detekovaný v druhom bode do uplynutia času ΔT je paket považovaný za stratený. Typy paketov, ktoré sú uvažované pri meraní závisia od aplikovaného filtra. Kvôli tomu by mal byť popis filtrov alebo konkretizácia typu paketu na ktorom sa realizovalo meranie oneskorenia poskytnutá ako časť výsledku.



Obr. 2.1 Všeobecná architektúra merania oneskorenia

2.3 Umiestnenie meracieho zariadenia

Voľba polohy meracích nástrojov je kritická, pretože poloha kde je merač udáva koncové body merania. Ak je cieľom merania je merať oneskorenie medzi zdrojovým a cieľovým hostiteľom, meracie zariadenie musí byť umiestnené čo najbližšie k zdroju, respektíve cieľu. Presne povedané nikdy takýmto spôsobom nebude možné presne odmerať oneskorenie medzi dvoma koncami, ale v praxi je možné získať veľmi presný odhad ak je dodržané vhodné umiestnenie merača. Ak medzi meračom a zdrojovým resp. cieľovým hostiteľom sú umiestnené len pasívne sieťové prvky(napr. horizontálna

kabeláž) tak môžeme namerané hodnoty považovať za veľmi presné. Výpočtom rýchlosti šírenia elektrického signálu v kabeláži je možné na základe vzdialenosti merača od hostiteľa spresniť meranie. Aby bolo možné umiestniť merač, je potrebná znalosť fyzickej cesty, ktorou budú prechádzať pakety alebo dátové toky, ktoré sa budú analyzovať. Následne je potrebné umiestniť merače na tejto ceste.

2.4 Synchronizácia hodín

Z dôvodu merania časových parametrov, je nutné, aby merania boli založené na meracích systémoch so synchronizovaných časom. Aj keď koordinácia meraní by mohla byť založená na systéme TCP 3-way handshake cez komunikačný kanál, aspoň pri meraní jednosmerného oneskorenia je potrebné lokálne synchronizovať obidve strany.

Bežne používanými riešeniami pre časovú synchronizáciu sú:

- Network Time Protocol (NTP),
- Global Positioning System (GPS),
- Rádiové hodinové signály (napr. DCF77),

pričom každé z riešení má svoje výhody a nedostatky.

Network Time Protocol (NTP)

Presnosť synchronizácie pomocou NTP [15] závisí na výkonnosti koncového prenosu po sieti, vrátane NTP klientov a serverov. Dosiahnuteľná presnosť je 50 mikrosekúnd, ak NTP server je synchronizovaný priamo so stratum 1 serverom.

Global Positioning System (GPS)

Global Positioning System (GPS) je pozičný a navigačný systém pozostávajúci z vesmírneho segmentu, riadiaceho segmentu a segmentu používateľov.

Samotný GPS prijímač dosahuje presnosť s maximálnou odchýlkou 100 nanosekúnd. Hlavným nedostatkom GPS je, že GPS prijímače vyžadujú priamu viditeľnosť na štyri alebo viac satelitov. To môže spôsobovať problémy, pretože meracie zariadenia sú často umiestnené blízko hlavných uzlov infraštruktúry, ktoré sú

umiestnené v miestnostiach serverov, ktoré sú bez okien alebo sú v suterénoch. Okrem toho je GPS ešte stále drahšie v porovnaní s ostatnými riešeniami.

Rádiový hodinový signál

Ďalšou možnosťou je použitie rádiového signálu pre synchronizáciu času. Napr. rádiový signál DCF77 je vysielaný na dlhých rádiových vlnách s frekvenciou 77,5 kHz. Celková presnosť je limitovaná na 1 ms alebo 20 mikrosekúnd za dobrých podmienok v závislosti od spôsobu odčítania času.

2.5 Odchyťavanie paketov

Pre uskutočnenie merania potrebujeme odchytiť isté množstvo údajov z paketu. Pri generovaní ID paketu je potrebné zohľadniť pravdepodobnosť kolízie, ktorá závisí od generujúcej funkcie a počtu bytov použitých ako vstup. Podľa [5] je postačujúce odchyťavať prvých 40 bytov od začiatku IP hlavičky. Výkon zariadenia odchyťavajúceho prevádzku je závislý na:

- počte prerušení generovaných sieťovým rozhraním,
- počte prepnutí kontextu,
- množstve dát prenesených do používateľskej oblasti ,
- zaťažení stroja (spôsobeným inými procesmi, napr. generovaním ID).

2.6 Generovanie časových známk

Časová známka paketu (timestamp) môže byť reprezentovaná ako absolútna hodnota času. Potom veľkosť údajov potrebných na reprezentáciu je úmerný požadovanej presnosti. Pre zníženie počtu bitov potrebných na reprezentáciu časovej známky je použiť relatívne hodnoty času. V takom prípade je potrebné určiť maximálny čas za ktorý paket môže prejsť od jedného meracieho bodu k druhému meraciemu bodu. Časová známka má byť jednoznačná iba v rámci tohto limitu. Presnosť môže byť ovplyvnená, ak čas použitý procesom generujúcim časové známky nie je rovnaký pre všetky pakety.

2.7 Filtrovanie

Filtrácia paketov je potrebná ak budeme merať len vybrané pakety. Filtrácia paketov je výhodná ak chceme zmenšiť množstvo výsledných dát merania a čas potrebný na spracovanie ďalšími procesmi (napr. generovanie ID). Filtrácia môže vyberať pakety so špecifickými charakteristikami. To môžu byť napríklad všetky pakety patriace do špecifického toku. V niektorých prípadoch je podstatné uchovávať informáciu do ktorého toku patrí meraný paket. Napríklad ak súčasne meriame QoS pre rôzne typy premávky je nutné uchovať informáciu o type filtra spolu s ID paketu a časovou známku.

V niektorých prípadoch, paket ID už obsahuje dodatočnú informáciu, pretože bolo vypočítané bijektívnou komprimačnou funkciou na jednotlivých položkách paketu. V ostatných prípadoch musí byť požadovaná informácia prenesená do ďalšej časti aplikácie aj s paket ID.

Najjednoduchším spôsobom filtrovania je lineárne vyhľadávanie. Lineárne vyhľadávanie využíva na uchovanie množiny pravidiel spojkový zoznam. Pravidlá sú uložené v poradí znižujúcej sa priority. Paket sa sekvenčne porovnáva s každým pravidlom pokiaľ sa nenájde pravidlo ktoré vyhovuje všetkým relevantným poliam. Tento algoritmus je jednoduchý, pamäťovo efektívny, no jeho škálovateľnosť je slabá. Závislosť času výpočtu klasifikácie paketu je lineárna s množstvom pravidiel.

2.8 Vzorkovanie

Vzorkovanie je zamerané na výber reprezentatívnej vzorky paketov. Táto podmnožina je použitá na získanie informácií o celej množine pozorovaných paketov, bez nutnosti ich všetky spracovať. Výber môže byť závislý od pozície paketu a/alebo obsahu paketu a/alebo pseudonáhodných rozhodnutí.

Koncepcia vzorkovania je uvedená v kapitole č. 4 „Koncepcia vzorkovania prevádzky v sieťových infraštruktúrach“.

2.9 Generovanie identifikátora paketu

Pasívne merania sú založené na metódach, ktoré neznačkujú a nemodifikujú prevádzku v sieti. Rozpoznávanie paketov je teda realizované generovaním identifikátora paketu, ktorý je založený na existujúcich položkách paketu. ID paketu musí byť v jednotlivých meracích bodoch rovnaké a teda je nutné ho z invariantných alebo predpovedateľných polí. Polia ktoré sú rôzne medzi paketmi (napr. kontrolný súčet IP hlavičky) sú vhodnejšie ako polia, ktoré sú konštantné alebo sa v nich strieda malý počet hodnôt. ID by nemalo byť príliš veľké kvôli množstvu generovaných dát čo je v rozpore s požiadavkou na nízky počet kolízií (jedinečnosť ID). V neposlednom rade funkcia generujúca ID by mala byť rýchla.

Výber polí hlavičky IP paketu

Generovanie paketového ID by malo byť založené na poliach, ktoré:

- už existujú v pakete (nie je potrebná žiadna modifikácia paketu),
- sú invariantné ale predpovedateľné počas prenosu (aspoň na ceste zo vstupného miesta do výstupného bodu s druhým bodom merania),
- majú vysokú variabilnosť medzi rôznymi paketmi.

Nemenné polia hlavičky IP paketu:

- Verzia
- Dĺžka hlavičky
- Celková dĺžka
- ID datagramu
- Protokol
- Zdrojová adresa
- Cieľová adresa

Premenlivé polia hlavičky IP paketu:

- Typ služby (TOS): V špecifikácii IP nie je určené, ktoré pole je zmenené
- Príznamy: Toto je možné meniť, pretože smerovače môžu okamžite nastaviť príznak „nefragmentovať“;
- Fragment Offset: Toto pole je možné zmeniť, ak je vykonaná refragmentácia;
- Time to Live (TTL): TTL sa dekrementuje v každom smerovači;
- Kontrolná suma hlavičky: Toto pole sa zmení vždy vtedy, keď sa menia ostatné polia v hlavičke.

Podľa vyššie uvedeného je vhodné generovať ID pri IPv4 z týchto hlavičiek:

- Celková dĺžka
- ID datagramu
- Protokol
- Zdrojová adresa
- Cieľová adresa
- Užitočné data (z pohľadu 3.vrstvy ISO/OSI modelu)

Pri IPv6 je vhodné generovať ID z týchto hlavičiek:

- Dĺžka užitočného množstva dát v pakete (payload)
- Protokol
- Zdrojová adresa
- Cieľová adresa
- Užitočné data (z pohľadu 3.vrstvy ISO/OSI modelu)

Funkcia generujúca ID

Požiadavka pre malý počet kolízií (jedinečnosť ID) je v rozpore s požiadavkou na malé hodnoty ID paketu, pretože čím viacej bitov je použitých na reprezentáciu ID paketu, tým nižšia je pravdepodobnosť kolízie. Pravdepodobnosť kolízie v zaznamenanej premávke závisí od:

- distribúcie postupnosti bitov na vstupe funkcie generujúcej paket ID
- veľkosti ID paketu $l(id)$
- funkcie generujúcej paket ID
- použitej implementácii IP protokolu v operačnom systéme (ak uvažujeme datagram ID).

Cieľom je dosiahnuť akceptovateľne nízku pravdepodobnosť kolízie s ID paketu, ktoré neprekračuje dostupnú kapacitu pre prenos nameraných údajov. Tak ako pri časovej známke, ID paketu musí byť jedinečné iba v danom časovom intervale $[0, t_{max}]$. Toto limituje počet možných kombinácií na množstvo paketov n_{max} , ktoré môžu byť merané v tomto časovom intervale

Možné spôsoby generovania ID paketu z uvažovaných polí:

- nezmenené polia
- jednosmerné hašovacie funkcie (MD5, SHA-1, atď.)
- kontrolné sumy (CRC)
- kompresné funkcie.

Pri funkcii ktorá mapuje veľké množstvo postupnosti bitov do menšej postupnosti bitov, sa vždy môže vyskytnúť kolízia. Použitie vybraných častí hlavičky ako identifikátora je spôsob ako je možné dosiahnuť minimálnu pravdepodobnosť kolízie. Navyše tento spôsob nevyžaduje ďalšie spracovanie. Výsledná veľkosť ID je suma veľkostí vybraných polí, čo zvyšuje množstvo výstupných dát a rýchlosť ich generovania, v krajnom prípade až na rýchlosť samotného toku.

MD5 [16] funkcia je kryptografická hašovacia funkcia určená na generovanie 128 bitového odtlačku správy ľubovoľnej dĺžky. Používa sa na autentifikáciu správ, no vzhľadom na nájdenie kolízie kompresnej funkcie (1996) sa už jej použitie neodporúča na zabezpečenie kriticky dôležitých údajov. Je veľmi rýchla – 60 Mb/s (Pentium 90MHz) [18]. Aj keď kolízia kompresnej funkcie už bola dokázaná, spôsob nájdenia kolízie hash-u ešte nebol nájdený. Vzhľadom na túto skutočnosť a jej rýchlosť ju možno odporúčať ako funkciu vhodnú na generovanie ID.

SHA-1 [17] hašovacia funkcia je určená na generovanie 160 bitového odtlačku správy ľubovoľnej dĺžky. V súčasnosti sa považuje za bezpečnú a je najviac preferovanou metódou na podpisovanie elektronických správ. Jej dĺžka 160bitov neumožňuje nájsť v najbližších 20 rokoch kolíziu. Je o dosť pomalšia ako MD5 – 21 Mb/s (Pentium 90MHz) [18].

Kryptografické funkcie sú odolné voči kolíziám a majú veľmi dobrú distribúciu hašovacích hodnôt.

CRC bolo vyvinuté pre detekciu chýb a ich opravu. Nebolo však navrhované aby bolo odolné voči kolíziám. Frekvencia kolízií závisí na veľkosti CRC a generačnom polynóme.

Kompresnými funkciami použitými na vybrané polia hlavičky vieme dosiahnuť nízku pravdepodobnosť kolízie. Stupeň kompresie vieme ovplyvniť množstvom predpokladov o meranej premávke (napr. verzia IP je 4 alebo 6, maximálna hodnota TTL). Výhodou týchto funkcií je, že sú bijektívne. To znamená, že vieme získať ďalšie informácie o meraní priamo z ID.

2.10 Prenos nameraných údajov

Pre výpočet časových parametrov QoS je potrebné analyzovať údaje z viacerých paketov. Ak je na meranie použitých viacero meracích bodov, je potrebné preniesť namerané a vygenerované údaje do zberného bodu. Zberným bodom môže byť aj jeden z meracích bodov najmä kvôli redukcii množstva prenášaných dát.

Existuje viacero spôsobov prenosu nameraných údajov:

- prenos údajov v paketoch

- prenos údajov v okruhu — namerané a vygenerované údaje sú prenášané tou istou cestou ako prevádzka, z ktorej boli získané
- prenos údajov mimo okruhu — namerané a vygenerované údaje sú prenášané inou cestou ako prevádzka, z ktorej boli získané

3. Architektúra nástroja pre neintruzívne merania („BasicMeter“)

Praktickým cieľom diplomovej práce je implementovať jednoduchý merací nástroj, ktorého cieľom bude odchyťvanie paketov, generovanie paket ID, filtering a sampling na najnižšej úrovni nástroja „basicmeter“. Prvotná verzia nástroju „basicmeter“ bola naprogramovaná v jazyku C, no vzhľadom na problémy s odladovaním zdrojového kódu a značnú nestabilitu sa rozhodlo že celý nástroj bude naprogramovaný od začiatku objektovo s prihliadnutím na dekompozíciu na jednoduché objekty s jasne definovaným interfacom a vysvetľujúcim komentárom ku každej relevantnej metóde a atribútu. Ako implementačný jazyk sme zvolili C++, vzhľadom na jeho širokú podporu a rýchlosť v operačnom systéme GNU/Linux. Cieľom projektu je vytvoriť voľne dostupnú a použiteľnú platformu pre neintruzívne merania prevádzkových parametrov v počítačových sieťach. Úloha sa primárne nezameriava na vytvorenie komplexného nástroja na meranie veľkého množstva parametrov, cieľom je skôr vyvinúť z používateľského hľadiska jednoduchý, a zároveň jednoducho modifikovateľný nástroj na základoch ktorého by už bolo možné navrhovať komplexnejšie meracie platformy.

3.1 Špecifikácia požiadaviek

Hlavné požiadavky na implementáciu meracieho nástroja boli špecifikované v nasledujúcich bodoch:

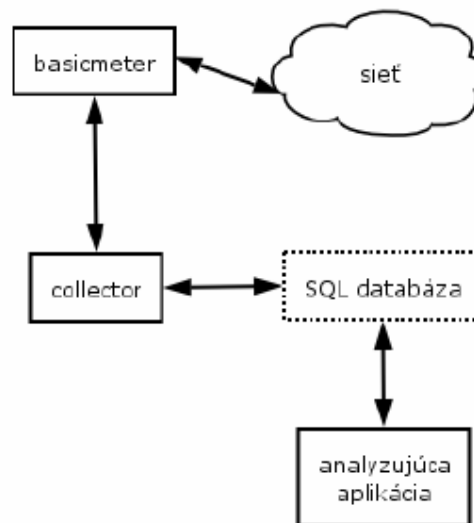
- vytvorenie jednoduchého nástroja pre ovládanie z konfiguračného súboru
- navrhnuť modulárny dizajn, ktorý by zaručoval rozšíriteľnosť jednoduchým spôsobom
- možnosť jednoduchého vzdialeného ovládania meracieho nástroja
- podpora protokolu NetFlow verzie 9

- podpora novo vznikajúceho štandardu PSAMP
- podpora šablón, podpora verifikovateľnosti zadaných šablón
- možnosť konfigurácie jednoduchým textovým konfiguračným súborom
- implementácia čo najviac portabilným spôsobom
- implementácia by mala byť východiskovým bodom pre podporu meraní podľa štandardu IPFIX
- navrhnuť podporu pre implementáciu vzorkovacích funkcií

V ďalšom popise bude navrhnutý a implementovaný nástroj označovaný názvom „basicmeter“.

3.2 Konceptia meracieho nástroja

Koncepciou meracieho nástroja je podpora štandardov PSAMP[14] a IPFIX[6] v čo najlepšej možnej miere. Navrhovaný a implementovaný merací nástroj je súčasťou tejto koncepcie znázornenej na obrázku 3.1.



Obr. 3.1: Architektúra meracej platformy

Merací nástroj je súčasťou projektu vývoja kompletnej meracej platformy pre pasívne merania kvality služieb v počítačových sieťach. Časti tejto kompletnej meracej platformy sú naznačené v tabuľke 3.1.

Časť architektúry	Popis
basicmeter (merací proces)	popisovaná aplikácia slúžiaca ako merací proces, zachytávanie paketov a vytváranie dát pre zhromažďovací proces
collector (zhromažďovací proces)	zhromažďovací proces, spracováva exportované pakety z exportovacieho procesu, nie je súčasťou popisovaného riešenia
analyzujúca aplikácia	aplikácia, ktorá pristupuje k exportovaným dátam a vykonáva analýzu (grafickú, štatistickú) na požiadanie používateľa

Tab. 3.1: Popis jednotlivých častí architektúry

SQL databáza nie je súčasťou špecifikácie IPFIX a ani protokolu NetFlow[3] — v schéme je to naznačené bodkovaním. Všeobecne sa na uloženie exportovaných paketov predpokladá akýkoľvek dátový sklad (súbor, databáza pracujúca so súborovým systémom, vyhradená partícia disku pre uloženie naformátovaných dát) Kvôli jednoduchému použitiu a dobrým možnostiam ďalšieho spracovania uložených dát bola ako dátový sklad zvolená práve SQL databáza.

3.3 Návrh meracieho nástroja

Aplikácia je koncipovaná ako samostatný spustiteľný program z príkazového riadku. Ako hlavný implementačný jazyk bol zvolený jazyk C++, vzhľadom na svoju rýchlosť, nezávislosť na platforme a širokú podporu na UNIX-ových operačných systémoch. S ohľadom na maximálnu možnú portabilitu programu nebudú využívané neštandardné funkcie jadra operačného systému. Pre vývoj bola zvolená platforma operačného systému GNU/Linux kvôli jeho dostupnosti, stabilite, flexibilita a možnostiam vývoja. Bloková schéma aplikácie je na obrázku 3.2. Aplikácia je logicky rozdelená do troch častí:

- časť odchyťovania, filtrovania, vzorkovania a sledovania paketov
- klasifikačná časť
- exportovacia časť (exportovací proces)

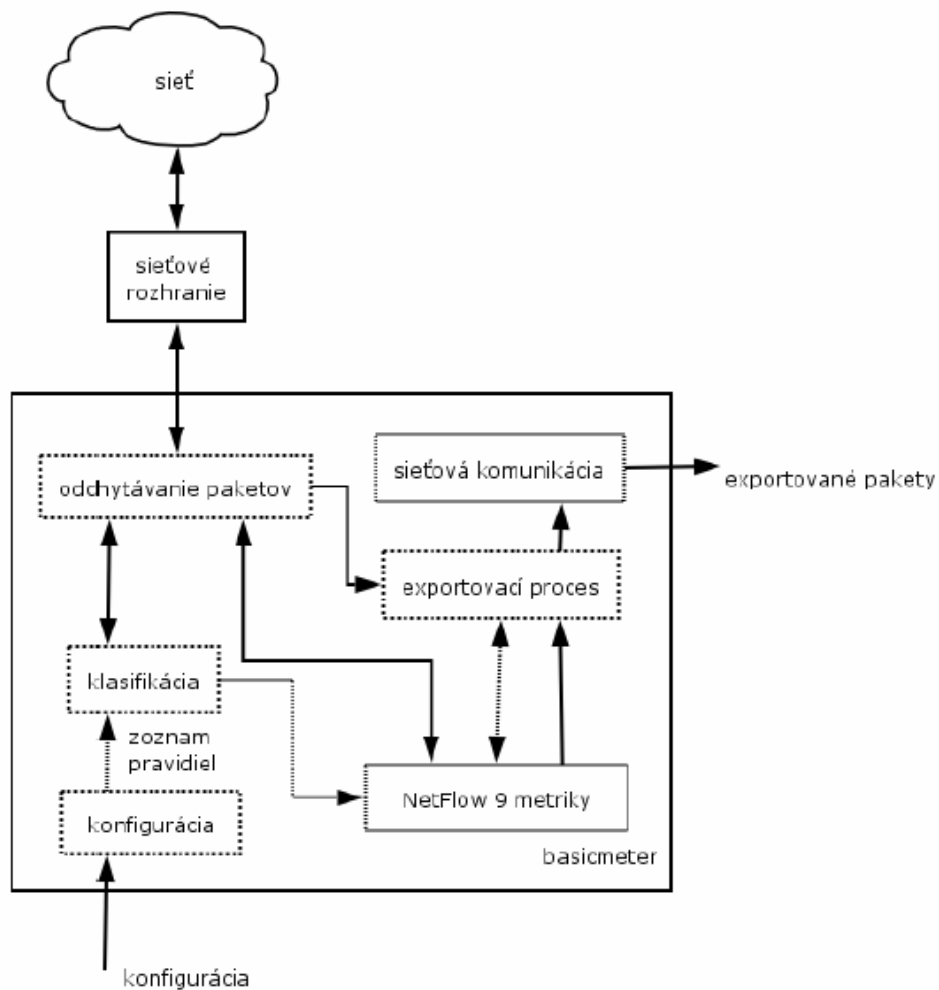
Tieto logické časti spolu komunikujú pomocou štandardných prostriedkov poskytovaných operačným systémom.

Časť sledovania a odchyťovania paketov používa k vykonávaniu týchto úloh knižnicu libpcap [7]. Knižnica libpcap bola zvolená kvôli svojej podpore na rôznych unixových operačných. Knižnica libpcap spolupracuje s jednoduchým paketovým filtrom[2] obsiahnutým v jadrách väčšiny unixových operačných systémov a tým umožňuje významne zjednodušiť proces filtrovania paketov.

Časť vykonávajúca vzorkovanie bola navrhnutá s prihliadnutím na novo vznikajúci štandard PSAMP[14]. Jej implementácia podporuje jednoduché pridanie resp. modifikáciu konkrétnych vzorkovacích metód. Parametre vzorkovania sú načítavané z konfiguračného súboru.

Exportovacia časť (exportovací proces) je samostatná časť programu, ktorá spracúvava dáta získané odchyťavacou časťou sledovaním procesu. Exportovacia časť bola navrhnutá tak, aby podporovala protokol NetFlow[3] verzie 9 aj s možnosťami vytvárania šablón. Šablóny sú uložené v textových súboroch ako XML (Extensible

Markup Language) [1] dokumenty. Pre validáciu týchto dokumentov je použitý zjednodušený jazyk pre popis štruktúry XML dokumentov — jazyk pre popis schémy XML dokumentov s názvom RelaxNG [4]. Pre všetky činnosti súvisiace so spracovaním XML dokumentov bude použitá knižnica libxml2 [10]. Prenos exportovaných paketov bude prebiehať prostredníctvom UDP.



Obr. 3.2: Architektúra meracieho nástroja

3.4 Architektúra meracieho nástroja

Architektúra nástroja je navrhovaná tak, aby vyhovela všetkým špecifikovaným požiadavkám. Nástroj je rozdelený do troch hlavných častí, do časti odchyťavajúcej pakety, do časti klasifikátora a do exportovacej časti (exportovací proces). Ďalšou časťou podieľajúcou sa na činnosti programu je časť spracovávajúca konfiguráciu programu z textového konfiguračného súboru. Časť konfigurácia má za úlohu načítať a spracovať konfiguračné parametre z textového konfiguračného súboru. Pre tieto činnosti je použitá knižnica libconfuse [11] — knižnica určená na spracovanie textových konfiguračných súborov. Odchyťavacia a sledovacia časť (capture) má na starosti sledovanie a odchyťavanie paketov na základe parametrov spracovaných konfiguračnou časťou a odovzdaných odchyťavacej časti v procese inicializácie. Táto časť využíva funkcie knižnice libpcap [7], ktorá realizuje samotné odchytenie paketov a ich prenos z priestoru jadra (kernel space) do priestoru používateľa (userspace). Na túto operáciu je potrebné najvyššie práva, teda práva administrátora systému - roota, aplikáciu je potrebné spúšťať s efektívnym používateľským identifikátorom (user identifier, UID) 0 — v unixových operačných systémoch označuje používateľa s najvyššími právami v subsystéme pridelovania práv. Samotný výber paketov na sledovanie a odchytenie je realizovaný pomocou vysoko úrovňového abstraktného popisného jazyka podobného tomu, ktorý je použitý v nástroji na sledovanie sieťovej prevádzky s názvom tcpdump [9]. Odchyťavajú sa všetky pakety patriace aspoň do jedného toku. Samotné odchyťavanie paketov je realizované pomocou berkeleyského paketového filtra (Berkeley Packet Filter, BPF) prítomného v jadrách väčšiny unixových operačných systémov. Filter je implementovaný tak, že je možné vyberať pakety na základe ďalších polí v hlavičke okrem štandardných (zdrojová a cieľová adresa, zdrojové a cieľové porty), teda je možné vyhovieť špecifikácii IPFIX[6]. Použitie filtra znižuje počet prepnutí kontextu — prenosov medzi priestorom jadra a používateľským priestorom — pretože sa prenášajú len pakety so žiadanou informáciou a týmto prispieva k zvýšeniu výkonnosti meracieho nástroja. Po filtrovaní sa zapája vzorkovanie, ktoré určuje či bude paket vybraný pre ďalšie spracovanie alebo nie.

Klasifikátor má za úlohu nájsť dátový tok, do ktorého patrí práve spracovávaný paket a odovzdať informáciu o identifikátore toku do časti vytvárania tokov (NetFlow 9 metriky). Komponent NetFlow 9[24] metriky slúži na napĺňanie dátových tokov

získaných z pozorovaných informácií na základe spracovania položiek v jednotlivých šablónach. Toky sú vytvárané a spravované v časti exportovací proces, kde sú vytvárané, spravované a odosielané toky do časti sieťovej komunikácie. V časti exportovacieho procesu sú načítavané a verifikované (validované) šablóny pre správnosti zadania. Šablóny sú definované v textových súboroch ako dokumenty XML (Extensible Markup Language) [1]. Pre verifikáciu šablón sa používa jazyk pre popis štruktúry XML s názvom RelaxNG. Časť sieťovej komunikácie je použitá ako abstraktná vrstva medzi exportovacou časťou a transportnými protokolmi. V súčasnosti program používa ako transportný protokol UDP (User Datagram Protocol).

4. Koncepcia vzorkovania prevádzky v sieťových infraštruktúrach

Neintruzívne merania poskytujú elegantný spôsob priameho zisťovania parametrov QoS sieťovej prevádzky zákazníka. Sú založené na existujúcej prevádzke v sieti a poskytujú objektívne výsledky bez generovania nadbytočnej prevádzky v sieti. A predsa aj pri pasívnych meraniach prevádzkové náklady na meranie môžu narastať kvôli zvyšujúcim sa rýchlostiam v počítačovej sieti a kvôli zvyšujúcim sa nárokom na požiadavky merania (napr. overovanie SLA, detekcia DoS útoku, účtovanie prevádzky).

Vzorkovanie umožňuje zníženie nákladov na meranie skúmaním len časti zachytenej prevádzky. Nahrádza pôvodnú metriku metriku s istým odhadom, takže redukcia nákladov prináša určitú degradáciu presnosti. Typ a rozsah tejto nepresnosti závisí len na type a parametroch použitého vzorkovania a charakteristike meranej prevádzky.

V nasledujúcich kapitolách uvediem terminológiu používanú pri popise vzorkovacích stratégií, charakterizujem jednotlivé typy selekcie paketov. Zvyšok kapitoly bude zameraný na popis architektúry PSAMP[14], jednotlivých požiadavok opisujúcich vzorkovanie vrátane opisu bezpečnostného rizika vzorkovania.

4.1 Terminológia

Pozorovací bod

Pozorovací bod je miesto v sieti kde sú IP pakety sledované. Príkladom je napríklad sonda pripojená k zdieľanému médiu ako napr. LAN sieti založenej na Ethernete, port smerovača, skupina rozhraní (fyzických, alebo logických) smerovača alebo vložený merací systém s rozhraním. Jeden pozorovací bod môže byť množina viacerých iných pozorovacích bodov[23].

Pozorovaný tok paketov

Pozorovaný tok paketov je množina všetkých paketov pozorovaných v pozorovacom bode[23].

Tok paketov

tok paketov je podmnožina pozorovaného toku paketov ktorý prúdi z meracieho bodu. Príkladom môže byť výstup procesu selekcie[23].

Obsah paketu

Obsah paketu zahŕňa štruktúru hlavičiek (spojová, sieťová a transportná vrstva) a užitočné dáta[23].

Proces selekcie

Proces selekcie berie ako vstup pozorovaný tok paketov a volí jeho podmnožinu ako výstup[14].

Stav selekcie

Proces selekcie môže udržiavať stavové informácie používané procesom selekcie alebo aj oznamovacím procesom. V danom čase proces selekcie môže závisieť na pozorovaných paketoch v danom čase, pred daným časom a iných premenných napríklad:

- Poradie paketu pri vstupe do selektora
- Časová známka pozorovania paketu v pozorovacom bode
- Iterácie používané generátorom náhodných čísel
- Kontrolný súčet počítaný počas procesu selekcie
- Indikátor selekcie paketu

Proces selekcie môže meniť časti stavu selekcie ako výsledok spracovania paketu. Stav selekcie paketu je stav po selekcii paketu[23].

Selektor

Selektor definuje akciu procesu selekcie. Vstupom je jeden paket a výstupom príznak či je daný paket zvolený ako časť výstupného toku paketov[14].

Selektor môže použiť nasledovné informácie na rozhodnutie o selekcii:

- Obsah paketu
- Informáciu zo spracovania paketu v pozorovacom bode
- Akýkoľvek stav selekcie ktorý udržiava proces selekcie

Zložený selektor

Zložený selektor je usporiadaná kompozícia selektorov, ktorých výstup prvého selektora je napojený ako vstup do nasledujúceho selektora[23].

Primitívny selektor

Primitívny selektor je selektor ktorý nie je Zložený selektor[23].

Oznamovací proces

Oznamovací proces vytvára oznamovací tok na paketoch zvolených procesom selekcie ako prípravu na export[14]. Ako vstup do oznamovacieho procesu môže slúžiť informácia dostupná pre proces selekcie napr.:

- Obsah zvoleného paketu
- Informáciu zo spracovania paketu v pozorovacom bode
- Akýkoľvek stav selekcie ktorý udržiava zdroj, čiže proces selekcie na základe modifikácie stavu selekcie počas procesu selekcie.

Protokol paketu

Protokol paketu je konfigurovateľná sada informácií slúžiaca ako vstup do oznamovacieho procesu vrátane obsahu paketu, informácií získaných pri spracúvaní paketu (napr. výstupné rozhranie) a prideleného stavu selekcie[14].

Interpretácia protokolu

Interpretácia protokolu zahŕňa pomocné informácie týkajúce sa jedného alebo viacerých paketov. Používa sa na interpretáciu jednotlivých protokolov paketov. Ako príklad môžem uviesť konfiguračné parametre procesu selekcie a oznamovacieho procesu[23].

Oznamovací tok

Oznamovací tok je výstup oznamovacieho procesu obsahujúci dve významné informácie: protokol paketu a interpretáciu protokolu[14].

Merací proces

Merací proces je kompozícia procesu selekcie ktorý berie Pozorovaný tok paketov ako vstup nasledujúci oznamujúcim procesom[14].

Exportovací proces

Exportovací proces posielá vo forme paketov exportu výstup jedného alebo viacerých meracích procesov jednému alebo viacerým kolektorom[23].

Paket exportu

Paket exportu je kombináciou interpretácie protokolu a/alebo jedného alebo viacerých protokolov paketu ktoré sú dané dokopy exportovacím procesom do formy paketu exportu pre export do kolektoru[14].

PSAMP zariadenie

PSAMP zariadenie je zariadenie zhromažďujúce najmenej pozorovací bod, merací proces a exportovací proces. Typické PSAMP zariadenie je router[23].

Kolektor

Kolektor dostáva ako vstup oznamovací tok exportovaný jedným alebo viacerými exportujúcimi procesmi. V niektorých prípadoch môže hositeľ slúžiť ako merací proces, exportér a kolektor zároveň[23].

Metódy selekcie**Filtrovanie**

Filter je selektor ktorý volí pakety deterministicky na základe obsahu paketu, jeho spracovania alebo funkcií ktoré vzniknú počas procesu selekcie. Príkladom môže byť match filtering a selekcia na základe hash-u[14].

Vzorkovanie

Selektor ktorý nie je filtrom je nazývaný vzorkovacím selektorom. Vzorkovanie používa na selekciu paketu aspoň jednu inú vlastnosť ako jeho obsah. Príkladom môže byť poradie paketu, čas príchodu paketu, udaná pravdepodobnosť[14].

Obsahovo nezávisle vzorkovanie

Obsahovo nezávisle vzorkovanie je operácia, ktorá nepoužíva obsah paketu alebo funkcie odvodené z obsahu paketu ako základ pre výber.

Príkladom je systematické vzorkovanie alebo náhodné vzorkovanie[14].

Obsahovo závisle vzorkovanie

Obsahovo závisle vzorkovanie je operácia, ktorá používa obsah paketu ako jedno z kritérií

výberu paketu. Ako príklad možno uviesť pseudonáhodnú selekciu na základe obsahu paketu a generátora náhodných čísel. Je nutné si všimnúť, že to nie je filter, pretože výber nie je deterministický[14].

Hash doména

Podmnožina z obsahu paketu a informácie zo spracovania zobrazená ako N-bitový reťazec pre nejaké kladné celé číslo N[14].

Rozsah hash-u

Množina M-bitových pre nejaké kladné celé číslo M[14].

Hashovacia funkcia

Deterministická mapa z hash domény do hash rozsahu[14].

Zvolený rozsah hash-u

Podmnožina rozsahu hash-u. paket je zvolený ak výsledok hashovacej funkcie z hashovacej domény je zo zvoleného rozsahu hash-u[14].

Selekcia založená na hash-i

Filtrovanie špecifikované hashovacou doménou, hashovacou funkciou, rozsahom hashu a zvoleným rozsah hash-u[14].

Približná selekcia

Vyššie menované selektory môžu byť aproximované operáciami rovnakej alebo inej kategórie pre účel implementácie. Napríklad uniformné pseudonáhodné vzorkovanie môže byť aproximované selekciou založenou na hash-i použitím vhodne zvolenej hashovacej funkcie a domény[14].

Populácia

Populácia je počet všetkých paketov v toku paketov, ktorý meriame alebo podmnožina tohto toku.

Za populáciu môžeme považovať za množinu z ktorej robíme selekciu. Príkladom sú všetky pakety z pozorovaného toku paketov ktoré sú odchytené za istý časový interval[14].

Veľkosť populácie

Počet paketov v populácii[14].

Veľkosť vzorky

Počet paketov vybraných výberovou operáciou z populácie[14].

Konfigurovat'el'ná časť selekcie

Pomer počtu zvolených paketov selektorom z celkového počtu vstupnej populácie a veľkosti populácie zvolený ako konfiguračný parameter selektora[14].

Dosiahnutá časť selekcie

Aktuálny pomer počtu paketov zvolených selektorom zo vstupnej populácie ku veľkosti populácie[14].

4.2 Techniky výberu paketov

Pod výberom paketu rozumieme operáciu ktorá na základe vstupných parametrov (napr. zdrojová IP adresa, cieľový port, časť tela paketu, náhodná pravdepodobnosť) rozhodne o selekcii paketu[14].

4.2.1 Filtrovanie

Filtrovanie umožňuje deterministický vyberá podmnožiny paketov so spoločnými vlastnosťami. Je použité keď máme záujem iba o podmnožinu celej populácie paketov. Vlastnosti môžu byť získavané priami z obsahu paketu, alebo závisieť od spracovania paketu smerovačom. Filtrovanie je deterministická operácia. Závisí od obsahu paketu alebo spracovania, nikdy nezávisí od polohy paketu v toku, alebo od pseudonáhodných rozhodnutí.

Bežnou technikou na selekciu paketu je použitie hashovacej funkcie na niektorých bitoch hlavičky paketu a/alebo obsahu a na voľbu či výsledok hashovacej funkcie je zo zvoleného rozsahu hashu. Aj keď hashovanie je deterministická operácia nad obsahom paketu, je to filtrovacía technika podľa definície. Aj napriek tomu sa hashovacie funkcie používajú na emuláciu náhodného vzorkovania. V závislosti od zvolenej hashovacej funkcie a hashovacieho rozsahu, táto technika môže úspešne emulovať náhodnú selekciu paketov s danou pravdepodobnosťou p . Jej výhodou je konzistentná selekcia paketu počas svojej cesty cez pozorovacie body.

Identifikujeme nasledujúce tri filtrovacie techniky. Prvé dva (Mask/match filtrovanie a Hash filtrovanie) sú bezstavové, a preto môžu svoje rozhodnutia založiť len na obsahu paketu. Tretí (filtrovanie podľa stavu smerovača) vyžaduje aj prístup k informácii o stave smerovača a je preto viac komplexnejšia[14].

Mask/match filtrovanie

Tento typ filtrovania vyberá pakety nasledujúcim spôsobom : Je vybraný istý počet pozícií bitov v závislosti od cieľa filtrovania. Potom je logickým AND aplikovaná maska na prichádzajúci paket, aby boli zachované iba vybrané bity. Výsledok tejto bitovej operácie je potom porovnaný s preddefinovanou hodnotou (napr. špecifická zdrojová IP adresa, TCP port), množinou hodnôt, alebo intervalom. Paket je následne vybraný podľa výsledku tohoto porovnania[14].

Hash filtrovanie

Hashovacia funkcia h mapuje obsah paketu c , alebo nejakú jeho časť, na hashovací interval R . Paket je vybraný ako $h(c)$ je element intervalu S , ktorý je podintervalom R a nazýva sa výberový hash interval. Preto je hash-výber v skutočnosti špeciálny typ filtrovania. Výber založený na hash filtrovaní má väčšinou dva typy použitia : Ponúka cestu ako aproximovať náhodné vzorkovanie použitím tela paketu na generovanie pseudonáhodných čísel, alebo konzistentne vybrať podmnožinu paketov zdieľajúcu istú spoločnú vlastnosť[14].

Filtrovanie podľa stavu smerovača

Táto trieda filtrov vyberá paket na základe podmienok a stavu smerovača. Nasledujúci zoznam ukazuje príklady takýchto podmienok. Tieto môžu byť kombinované operátormi AND, NOT, alebo OR[14].

- Vstupné rozhranie je rovné preddefinovanému rozhraniu.
- Výstupné rozhranie je rovné preddefinovanému rozhraniu.
- Paket porušil ACL (Access control list) na smerovači.
- Pri pakete zlyhalo Reverse path forwarding (RPF).

- Nebola nájdená cesta pre paket (route not found).
- Autonómny systém pôvodu sa rovná preddefinovanej hodnote, alebo leží v určitom intervale.
- Autonómny systém destilácie sa rovná preddefinovanej hodnote, alebo leží v určitom intervale.

4.2.2 Vzorkovanie

Vzorkovanie je zamerané na výber reprezentatívnej vzorky paketov. Táto podmnožina je použitá na získanie informácií o celej množine pozorovaných paketov, bez nutnosti ich všetky spracovať. Výber môže byť závislý od pozície paketu a/alebo obsahu paketu a/alebo pseudonáhodných rozhodnutí[14].

Implementácia vzorkovacích techník je zameraná na získanie informácií o špecifickej charakteristike celej populácie pri menších nákladoch, ako pri skúmaní celej populácie. Preto je dôležité naplánovať vhodnú vzorkovaciu stratégiu na zistenie žiadanej informácie s určitou presnosťou.

Je dôležité poznamenať znalosť typu metriky, ktorá bude použitá. Môže ísť o jednoduchý počet paketov alebo o zložité charakteristiky jednotlivých tokov (napr. priemerná veľkosť paketu, doba trvania toku).

Presnosť získaných výstupov vzorkovania je tak tiež veľmi dôležitá. Jednotlivé toky dát majú z pohľadu poskytovateľa služieb rôznu cenu, napr. Voice over IP alebo odosielanie emailov. Požiadavky na presnosť sú kriticky dôležité pri poskytovaní garantovaných služieb zákazníkom požadujúcich garantovanú kvalitu služieb. Tieto požiadavky sú uvedené v SLA (dohoda o poskytovaní úrovne služieb)

Vzorkovacie metódy môžu byť charakterizované vzorkovacím algoritmom, typom spúšťača a vzorkovacím intervalom. Vzorkovací algoritmus popisuje proces výberu vzoriek. Tieto charakteristiky musia byť dostupné všetkým aplikáciám

merajúcim dáta. Iba s touto znalosťou je možné správne interpretovať namerané výsledky. Bližšie sa problematikou vzorkovania zaoberá kapitola 5.

Algoritmus vzorkovania/filterovania	Determinizmus voľby	Závislosť na obsahu
Systematický podľa počtu	✓	✗
Systematický podľa času	✓	✗
Náhodný – výber n z N	✗	✗
Náhodný , uniformná pravdepodobnosť	✗	✗
Náhodný, neuniformná pravdepodobnosť	✗	✓*
Náhodný, neuniformná pravdepodobnosť, závislosť od stavu flow-u (IPFIX)	✗	✓*
Filtrovanie mask/match	✓	✓
Filtrovanie podľa hash funkcie	✓	✓
Filtrovanie podľa stavu routera	✓	✓*

Tab: 4.1: Tabuľka popisujúca charakteristiky vzorkovania a filtrovania

✓* je označená možnosť ak metóda môže ale nemusí používať pre výpočet obsah paketu

4.2.3 Šablóny

Definícia toku dát je pri štandarde NetFlow verzie 9[24] založená na šablónach. NetFlow je flexibilná a rozšíriteľná metóda firmy Cisco Systems umožňujúca záznam dát používaných na meranie výkonnosti siete. Šablóny predstavujú rozšíriteľný návrh pre paket exportu. Táto vlastnosť umožňuje budúce rozšírenia bez potreby zmeny základných vlastností formátu záznamu tokov.

Šablóna je usporiadaná n-tica (napr. <type,length>, TLV), použitá na určenie štruktúry určitej informácie, ktorá komunikuje s IPFIX zariadeniami smerom na kolektor. Každá šablóna je jedinečne identifikovateľná nejakým spôsobom (napr. použitím Template ID).

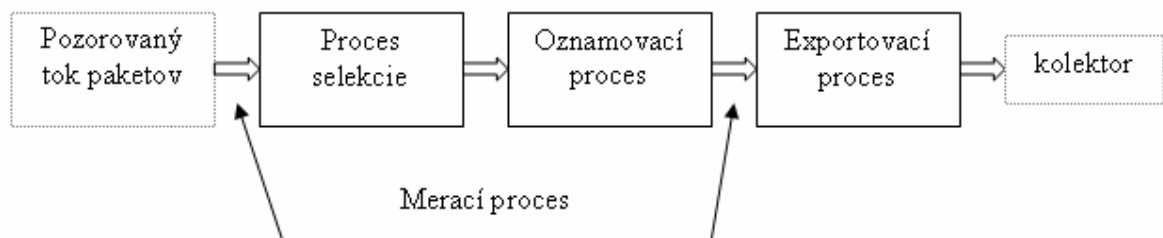
Použitie šablón má niekoľko výhod:

- NetFlow je odolný voči zmenám nových alebo vyvíjajúcich sa protokolov, pretože je možné jednoducho dodať podporu pre tieto protokoly,
- nové vlastnosti môžu byť k NetFlow[24] protokolu pridané jednoducho bez znefunkčnenia existujúcich implementácií,
- aplikácie pre zhromažďovanie alebo analýzu dát môžu byť jednoduchým spôsobom (zmenou šablóny) obohatené o tieto nové vlastnosti.
- Exportný paket sa v protokole NetFlow verzia 9 skladá z dátových položiek, položiek definujúcich šablóny a z položiek nastavujúcich parametre pre zhromažďovací proces.

4.3 Popis architektúry vzorkovania PSAMP

Architektúra vzorkovania PSAMP (packet sampling) sa skladá z viacerých navzájom prepojených procesov. Ku každému z týchto procesom môžeme priradiť jednoznačné vstupy a výstupy. Každý proces si načítava konfiguráciu z vopred pripraveného konfiguračného súboru.

Tok paketov je odchyťovaný v pozorovacom bode. Proces selekcie skúma každý paket či má byť vybraný. Oznamovací proces zostavuje správu o každom pakete s využitím obsahu paketu a iných informácií týkajúcich sa paketu ako napr. časová známka, výstupné rozhranie. Exportovací proces posielá správy kolektoru spolu s informáciou potrebnou na ich interpretáciu. Kompozícia procesu selekcie a oznamovacieho procesu je známa ako merací proces. Obrázok 4.1 popisuje architektúru PSAMP zariadenia[23].



Obr. 4.1: Architektúra PSAMP zariadenia

4.3.1 Proces selekcie

Úlohou procesu selekcie paketov je voľba vzorky z celkovej populácie. Selekcia môže byť založená na obsahu paketu a/alebo na redukcii počtu paketov v meracom bode[12]. Selekcia paketov je založená na kombinácii procesu filtrovania a vzorkovania. Podrobnejšie sa filtrovaniu venuje kapitola 4.1 a procesu vzorkovania kapitola 5.

4.3.2 Proces exportu a oznamovací proces

Po prebehnutí procesu selekcie je potrebné odovzdať oznamovaciemu procesu hlavičku paketu a časť payloadu. Oznamovací proces zvolí konkrétne položky už vzorkovaného paketu, pridá ďalšie informácie (napr. hodnota hash-u, časová známka) zo stavu selekcie a stavu smerovača. Stav smerovača môže závisieť od obsahu paketu (IP prefix alebo cieľový AS asociovaný s cieľovou IP adresou v hlavičke paketu, vstupné a výstupné rozhranie paketu). Správa taktiež môže obsahovať ďalšie položky vypočítané ako funkcia zvoleného paketu a stavu smerovača. Na celkové zjednodušenie

celkového dizajnu môže tento výsledok funkcie byť priradený k paketu až pri procese exportu a nie ihneď pri procese selekcie. Avšak je nutné zaručiť že táto pridaná informácia bude sa týkať stavu routra v čase zachytenia paketu a nie v čase jeho exportu.

Zariadenie generujúce záznamy merania je nakonfigurované tak, že posiela údaje jednému alebo viacerým procesom kolektora. Exportovanie týchto záznamov ma podstatný vplyv na analýzu údajov [12]:

Transport

Existujú 2 základné spôsoby transportu: spoľahlivý a nespoľahlivý. V nespoľahlivom režime je obsahom paketu kompletná informácia o meraní. Táto informácia je zapuzdrená to UDP paketu a zaslaná kolektoru na preddefinovanú IP adresu z konfiguračného súboru. Odosielajúce zariadenie si neuchováva stav o tomto pakete (iný ako číslo poradia paketu exportu). V spoľahlivom režime zariadenie exportuje pakety pomocou TCP spojenia kolektoru, ktorý musí byť schopný podporovať TCP[12].

Rýchlosť exportu

Zariadenie prevádzajúce export by malo obsahovať konfigurovateľný limit počtu odoslaných záznamov pre kolektor za jednotku času. Ináč je možné, že meracie zariadenie neúnosne preťaží merací systém a kolektor. Tento problém sa ešte môže zhoršiť použitím spoľahlivého transportného módu, kde zariadenie sa snaží preposlať stratené pakety. Počas merania meracie zariadenie môže generovať nové záznamy rýchlejšie ako je jeho limit. V takomto prípade zariadenie by malo radšej zahodiť nadbytočné záznamy ako sa pokúsiť ich vyslať kolektoru. Zariadenie by si malo pamätať informáciu (číslo sekvencie alebo paketu) aby upozornilo kolektor na chýbajúce záznamy[12].

Maximálne oneskorenie exportovaných záznamov

Exportovacie zariadenie môže si radiť do fronty záznamy merania a neskôr ich poslať ako jeden paket kolektoru. Tu by mal byť definovaný istý časový interval počas ktorého sa musia exportovať záznamy. Má to dve príčiny: Prvou je fakt, že kolektor

musí dostávať ako tak aktuálne informácie o vzorkovaných paketoch. Druhou príčinou je to, že meracie zariadenie môže priradiť relatívnu časovú známku vzorkovaného paketu.

Lokálny export

Záznamy o tokoch môžu byť exportované kolektoru umiestnenému na rovnakom hostiteľovi kde sídli merací proces. Výhodou je možnosť navrhnutia špecifického API, ktoré nemusí používať protokol TCP alebo UDP.

4.3.3 Formát záznamu

Export záznamu vyžaduje spojenie jedného alebo viacerých záznamov a poslanie paketu s týmito záznamami kolektoru. Táto správa môže obsahovať viacero typov informácií ako[12]:

Informácia o pakete

Záznam pre každý vzorkovaný paket môže obsahovať informáciu vrátane rozličných položiek (IP adresa, číslo portu) a taktiež pomocné informácie (napr. časová známka, vstupné a výstupné rozhrania)

Konfiguračné parametre

Tok záznamov by mal obsahovať aj informáciu o konfigurácii merania toku údajov (napr. typ vzorkovania, parametre vzorkovania, filter). Toto zabezpečí že namerané záznamy sa dajú interpretovať bez dodatočnej informácie a kolektor môže ich priamo spracovať. Zmeny v konfigurácii musia byť okamžite hlásené v toku záznamov kolektoru.

Nazhromaždené informácie

Záznam by mal obsahovať dostatočnú informáciu pre kolektor aby kolektor bol schopný detekovať a ošetriť porušené alebo zničené pakety exportu. Záznam by mal obsahovať počet bytov a paketov ktoré vyhovujú filtru, alebo ktoré prešli procesmi filtrovania a vzorkovania.

Na úsporu miesta na ukladanie nameraných záznamov a šírky pásma, zariadenie môže komprimovať informáciu v pakete pri exporte. Kompresia by mala byť efektívna pretože vzorkované pakety obsahujú väčšinou veľmi podobné informácie (napr. číslo portu) [12] .

Architektúra PSAMP ako komponent architektúry IPFIX

Merací proces architektúry IPFIX obsahuje zachytávanie hlavičiek paketov ako prvý krok. Táto funkcia môže byť poskytnutá implementáciou PSAMP architektúry dvoma odlišnými spôsobmi [25].

Merací proces IPFIX-u môže slúžiť ako proces kolektora v architektúre PSAMP. Potom informácia o pakete vzorkovaného pomocou PSAMP komponentu môže byť poslaná z PSAMP exportovacieho procesu do IPFIX meracieho procesu použitím PSAMP protokolu. Alternatívne, bez použitia štandardizovaného protokolu alebo API, proces selekcie a proces vzorkovania architektúry PSAMP môžu poskytovať priamo informáciu o pakete meraciemu procesu IPFIX.

V oboch prípadoch, PSAMP komponent bude prevádzať zachytávanie hlavičky paketu, priradenie časovej známky a vzorkovací proces pre IPFIX merací proces [19].

4.4 Požiadavky na vzorkovanie

Požiadavky pre proces selekcie

Jednoduchosť: proces selekcie musí byť dostatočne jednoduchý aby sa dal implementovať všade pri maximálnej linkovej rýchlosti [23].

Použitelnosť: množina selektorov musí podporovať široký rozsah existujúcich a vznikajúcich aplikácií a protokolov merania. Toto vyžaduje dosiahnuť kompromis medzi aplikáciami a operačnými úlohami jednotlivých aplikácií a komplexnou množinou možností[23].

Rozšíriteľnosť: protokol musí byť schopný akceptovať nové selekčné procesy ktoré ešte nie sú definované[23].

Flexibilita: protokol musí podporovať voľbu paketov rozličných sieťových protokolov a iných zapuzdrení vrátane IPv4, IPv6 a MPLS[23].

Robustnosť selekcie: proces selekcie paketu musí byť odolný voči úmyselným útokom zvonka. Príkladom môže byť vyhnutie sa selekcie paketu alebo preťaženie meracieho systému[23].

Paralelné meracie procesy: protokol musí podporovať paralelné operácie viacerých nezávislých meracích procesov na jednom hostiteľovi[23].

Kauzalita: voľba selekcie každého paketu môže závisieť nanajvýš veľmi slabo na príchode budúcich paketov[23].

Kryptované pakety: procesy selekcie ktoré interpretujú polia paketu musia umožňovať ignorovanie zakryptovaných paketov ak sú tieto zvolené[23].

Konfiguračné požiadavky

Jednoduchosť: Jednoduchosť konfigurácie vzorkovania a parametrov exportu pre automatizovanú konfiguráciu na diaľku ako odozvu na správy kolektoru[23].

Bezpečnosť: možnosť konfigurácie pomocou protokolov podporujúcich ochranu voči neautorizovanému prístupu. Odpočúvanie konfigurácie meracieho procesu môže

poskytnúť útočníkovi znalosti ktoré by mu pomohli vyhnúť sa meraniu alebo preťažiť merací proces[23].

4.5 Popis parametrov opisujúcich vzorkovanie

Parametre popisujúce vzorkovanie sú elementy potrebné na jednoznačné definovanie najznámejších techník vzorkovania[14].

Parametre vzorkovania:

SELECTOR_ID

SELECTOR_TYPE

SELECTOR_PARAMETERS

ASSOCIATIONS

SELECTOR_ID je jedinečný identifikátor vzorkovača paketov, vypočítaný ako kombinácia ASSOCIATIONS a identifikátora lokálneho zariadenia.

SELECTOR_TYPE je typ použitého vzorkovacieho algoritmu.

Možné hodnoty: systematické vzorkovanie založené na počte, systematické vzorkovanie založené na čase, náhodné vzorkovanie typu n-z-N, náhodné vzorkovanie s uniformnou pravdepodobnosťou, náhodné vzorkovanie s neuniformnou pravdepodobnosťou alebo náhodné neuniformné vzorkovanie, závislé od stavu toku.

SELECTOR_PARAMETERS sú parametre definujúce vstup pre proces smplovania.

Dĺžka intervalu pri systematickom vzorkovaní znamená že všetky pakety počas zvoleného intervalu budú zvolené. Parameter medzery udáva miesto v čase alebo počte paketov ktoré nebudú vzorkované.

systematické vzorkovanie založené na počte

dĺžka intervalu odchyťavania (v paketoch), dĺžka medzery (v paketoch)

systematické vzorkovanie založené na čase

dĺžka intervalu odchyťavania (v μs), dĺžka medzery (v μs)

náhodné vzorkovanie typu n-z-N

veľkosť populácie N, veľkosť vzorky n

náhodné vzorkovanie s uniformnou pravdepodobnosťou

pravdepodobnosť vzorkovania p

náhodné vzorkovanie s neuniformnou pravdepodobnosťou

funkcia kalkulujúca pravdepodobnosť vzorkovania p

náhodné neuniformné vzorkovanie závislé od stavu toku

informácia o stave toku

ASSOCIATIONS: Parameter opisuje bod pozorovania a voliteľne proces IPFIX s ktorým je proces selekcie s asociovaný. AK nie je definovaný žiaden IPFIX proces tak selektor je pridelený každému IPFIX procesu na pozorovacom bode. STREAM_ID opisuje pôvod toku paketov ako vstup pre proces selekcie. Môže to byť buď pozorovací bod alebo identifikátor iného selektora.

Možné hodnoty: <STREAM_ID, ID meracieho procesu IPFIX, ID Exportovacieho procesu IPFIX, identifikátory iných asociovaných procesov>.

4.6 Bezpečnostné riziko vzorkovania

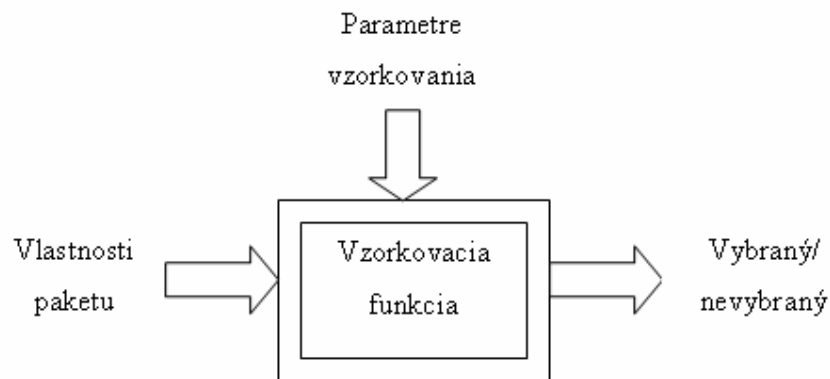
Zlomyselní užívatelia alebo útočníci sa môžu snažiť o ukrytie paketov z dohľadu poskytovateľov služieb alebo sieťových operátorov. Napr. ak paket selektor je použitý na účtovanie prevádzky alebo detekciu prieniku, užívateľ môže chcieť dosiahnuť zníženie počtu zvolených paketov.

Ak je použité deterministické vzorkovanie alebo ak proces selekcie berie ako vstup obsah paketu, užívateľ môže sa snažiť posielat' pakety v takom poradí a počte na základe ktorého budú menej vyberané. Aj keď funkcia selekcie je známa užívateľovi, merací nástroj môžeme ovplyvniť malou zmenou v konfigurácii nástroja tak aby volil navonok neodhadnuteľné poradia selekcie. Táto zmena procesu selekcie musí byť braná do úvahy pri voľbe vhodného procesu selekcie.

Ďalšie bezpečnostné riziko vzniká pri poškodení parametrov vzorkovania alebo poškodení komunikácie medzi parametrami vzorkovania a vyhodnocovacou aplikáciou[14].

5. Vzorkovacie algoritmy, ich vlastnosti a komparácia

Rozhodnutie o tom, či paket bude zvolený alebo nebude je výsledkom funkcie vzorkovania, ktorej vstupnými parametrami sú parametre vzorkovania a eventuálne časť paketu. Parametre vzorkovania ostávajú rovnaké pre celý proces vzorkovania a sú definované administrátorom pri spustení vzorkovania. Špeciálnym parametrom môže byť obsah paketu (napr. slúžiaci ako pseudo generátor náhodných čísel). Ktorá vlastnosť paketu bude braná ako vstup do vzorkovacej funkcie závisí od typu vzorkovacieho algoritmu a jeho parametrov. Na obrázku 5.1 je naznačená schéma vzorkovacej funkcie a jej vstupy a výstupy.



Obr. 5.1: Schéma vzorkovania

Obsahom tejto kapitoly bude opis jednotlivých vzorkovacích metód, analýza ich presnosti pomocou metód matematickej štatistiky a nakoniec voľba vzorkovacích algoritmov pre účel implementácie v nástroji „basicmeter“

5.1 Systematické vzorkovanie

Systematické vzorkovanie popisuje proces získavania štartovacích bodov a trvanie intervalu vzorkovania na základe deterministickej funkcie. Toto môže byť napríklad zvolenie každého k-teho elementu toku ale taktiež selekcia všetkých paketov zachytených v danom časovom intervale začínajúceho v preddefinovaných časových okamihoch. Aj keď proces selekcie nekoreluje s periodickou funkciou (časy medzi vzorkovacími intervalmi sú rôzne) nazývame toto vzorkovanie systematickým pokiaľ je selekcia deterministická.

Použitie systematického vzorkovania prináša riziko neobjektívnych výsledkov. Ak systematickosť v procese selekcie koreluje so systematickosťou sieťovej prevádzky sledovanej aplikácie, potom tu je vysoká pravdepodobnosť že výsledky budú neobjektívne. Systematickosť v sledovanej aplikácii resp. toku dát nemusí byť zväčša dopredu známa.

Uvažujme rovnako veľké medzery medzi bodmi vzorkovania, ktoré sa spúšťajú periodicky, či už na základe času alebo poradia paketov.. Potom všetky pakety spadajúce do vzorkovacieho intervalu sú zvolené[14].

Systematické vzorkovanie založené na počte

V systematickom vzorkovaní založenom na počte sú štartovacie a ukončovacie body pre vzorkovací interval dané pozíciou paketu v pozorovanom toku paketov v pozorovacom bode. Príkladom tohto vzorkovania môže byť selekcia každého n-tého paketu. Pre tento typ vzorkovania je potrebná implementácia počítadla paketov v meracom bode. Keďže pri pasívnych meraniach negenerujeme prevádzku, čas potrebný na zachytenie n špecifických paketov je vopred neznámi. Z toho vyplýva že čas potrebný na meranie, ktorého veľkosť intervalu počas ktorého sa nebude merať je značne veľká, môže byť veľmi dlhý (aj nekonečný) ak cieľom vzorkovania je minimálna veľkosť vzorky. Tento problém sa dostáva do popredia najmä pri pomalej prevádzke v toku[14].

Príkladom systematického vzorkovania založeného na počte môže byť nasledujúce schéma:

←dĺžka intervalu=8→←dĺžka medzery=6→←dĺžka intervalu=8→

Pozícia paketu : 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21

Dĺžka intervalu udáva počet vzorkovaných paketov za jeden interval

Dĺžka medzery udáva počet nevzorkovaných paketov

Potom výslednou vzorkou bude postupnosť paketov s poradím: 1, 2, 3, 4, 5, 6, 7,8, 15,16,17,18,19,20,....

Systematické vzorkovanie založené na čase

V systematickom vzorkovaní založenom na čase sú štartovacie a ukončovacie body pre vzorkovací interval dané časom. Všetky pakety z pozorovaného toku paketov v pozorovacom bode sú zvolené ak dôjdu v intervale definovaným štartovacím a ukončovacím bodom(napr. čas príchodu paketu je väčší ako štartovací čas a menší ako ukončujúci čas). Príkladom môže byť vzorkovania paketov prichádzajúcich každých 20 sekúnd. Ak ukončovací bod je taktiež daný pozíciou v čase, potom dĺžka intervalu vzorkovania je daná ich rozdielom. Pri tejto metóde môže nastať situácia, keď počas intervalu vzorkovania nepríde ani jeden požadovaný paket alebo naopak prídu všetky pakety a počas intervalu čakania nepríde ani jeden[14].

Obidve schémy vzorkovania sú nezávislé na obsahu paketu, čiže nie sú filtrom.

5.2 Náhodné vzorkovanie

Náhodné vzorkovanie si volí štartovacie a ukončovacie body vzorkovacieho intervalu na základe náhodného procesu. Voľba jednotlivých elementov je formou nezávislých experimentov. Na základe náhodnosti vieme dosiahnuť objektívne výsledky. V porovnaní so systematickým samplingom vyžaduje náhodné vzorkovanie generátor náhodných čísel[14]. Rozlišujeme dve metódy náhodného vzorkovania:

5.2.1 n-z-N vzorkovanie

V n-z-N vzorkovaní je náhodne vybraných n elementov z celkovej populácie ktorá pozostáva z N elementov. Príkladom môže byť generovanie n rozdielnych náhodných čísel z intervalu $[1, N]$ a selekcia všetkých paketov ktorých pozícia je rovná jednému z náhodných čísel. Pre tento typ vzorkovanie je veľkosť vzorky n fixná. Vstupný parameter n je možné zadať ako absolútne číslo n , alebo ako časť z celkovej populácie n/N [14].

5.2.2 Pravdepodobnostné vzorkovanie

V pravdepodobnostnom vzorkovaní rozhodnutie či vybrať alebo nevybrať element je generované na základe preddefinovanej pravdepodobnosti selekcie. Príkladom môže byť hod mincou pre každý paket a zvolenie všetkých paketov pre ktoré minca dopadne na stranu s označením hlava. Pre túto metódu vzorkovania je veľkosť vzorky variabilná a nedeterministická. Pravdepodobnosť vybrania paketu nemusí byť nutne rovnaká pre všetky pakety. Na základe variability pravdepodobnosti rozlišujeme medzi uniformným pravdepodobnostným vzorkovaním (pravdepodobnosť vybrania paketu je rovnaká pre všetky pakety) a neuniformným pravdepodobnostným vzorkovaním (pravdepodobnosť selekcie paketu je rozličná pre rôzne pakety) [14].

5.2.3 Uniformné pravdepodobnostné vzorkovanie

Pre uniformné pravdepodobnostné vzorkovanie sú pakety vyberané nezávisle s pravdepodobnosťou p . Toto vzorkovanie môže byť založené na počte, a je niekedy označované aj ako geometrické náhodné vzorkovanie, keďže po sebe idúce vybrané pakety sú nezávislé náhodné premenné s geometrickým rozložením a priemerom $1/p$. Časovo založená analógia, exponenciálne náhodné vzorkovanie, má čas medzi spúšťačmi exponenciálne rozložený.

Obidva, geometrické a exponenciálne náhodné vzorkovanie, sú príklady aditívneho náhodného vzorkovania, definované ako vzorkovanie ktorého interval alebo počet paketov odoberania vzoriek je nezávislý pokus identický s distribúciou náhodnej premennej[14].

5.2.4 Neuniformné pravdepodobnostné vzorkovanie

Toto je variant pravdepodobnostného vzorkovania v ktorej vzorkovacie pravdepodobnosti môžu závisieť od vstupu výberového procesu. Príkladom môže byť použitie hash-u z časti paketu a pravdepodobnostnej funkcie na výber paketu. To môže byť použité na váženie vzorkovacích pravdepodobností napr. na zväčšenie šance vyvzorkovania málo sa vyskytujúcich paketov, ktoré sú dôležité. Objektívny odhad týkajúci sa kvantitatívnej štatistiky je zrekonštruovaný renormalizáciou vzorkovacích hodnôt[14].

5.2.5 Neuniformné vzorkovanie závislé na stave toku

Ďalším typom vzorkovania, ktoré môže byť klasifikované ako neuniformné pravdepodobnostné, je úzko späté s konceptom IPFIX tokov a je používané spolu s funkciou monitorujúcou toky. Pakety sú vyberané v závislosti na stave výberu. Stav výberu je odvodnený od stavu toku a/alebo od stavov ostatných tokov práve monitorovaných. Príkladom algoritmu môžu byť nasledujúce metódy:

Ak je paket zaúčtovaný v zázname toku ktorý už existuje v IPFIX procese zaznamenania toku, tak je zvolený.

Ak paket nemá záznam v žiadnom z existujúcich záznamov toku je zvolený s pravdepodobnosťou p . Ak je zvolený tak sa vytvorí nový tok.

Tento typ vzorkovania je závislý na obsahu paketu, pretože na identifikáciu toku, ku ktorému paket patrí je potrebné analyzovať časť tela paketu. Ak je tento paket vybraný, je predaný ako vstup IPFIX monitorovacej funkcii (tzv. lokálny export). Výber

paketov v závislosti od stavu pamäte tokov (flow cache) je užitočné v prípade nedostatku pamäťových zdrojov[14].

5.3 Štatistické modely vhodné pre popis a optimalizáciu vzorkovacích metód

Výsledkom štatistického popisu každého algoritmu vzorkovania je určenie intervalu presnosti pomocou relatívnej odchýlky. Po splnení počiatočných podmienok pre veľkosť vzorky a celkovej populácie môže štatistický model vzorkovacích algoritmov slúžiť na objasnenie a spresnenie výsledku nameraných hodnôt merania. Taktiež vopred známa veľkosť odchýlky nameraných hodnôt od skutočnej hodnoty môže pomôcť pri určení správnej stratégie vzorkovania, teda k určení vhodného vzorkovacieho algoritmu a jeho vstupných parametrov.

5.3.1 Počiatočné podmienky

Počiatočné podmienky je nutné splniť, aby analýza vzorkovacích metód zo štatistického hľadiska dávala zmysluplné výsledky.

Veľkosť vzorky (n) je dostatočne veľká ak spĺňa nasledujúcu podmienku:

$$n > \frac{9}{p(1-p)}$$

kde p je pravdepodobnosť výberu požadovaného paketu z celkovej populácie.

Podiel veľkosti vzorky a celkovej populácie je nevyhovujúci ak spĺňa nasledujúcu podmienku:

$$\frac{n}{N} \leq 0.05$$

kde N je veľkosť celkovej populácie[21].

Veľkosť celkovej populácie je dostatočná ak spĺňa podmienku[20]:

$$N - 1 \approx N$$

	Populácia	Vzorka	Odhad
Počet paketov	N	n_R	Nie je potrebný (existencia počítadla)
Počet paketov porušujúcich SLA	M	m	\hat{M}
Podiel paketov porušujúcich SLA	$P = \frac{M}{N}$	$p = \frac{m}{n_R}$	\hat{P}

Tab. 5.1: Terminológia

5.3.2 Vzorkovanie n-z-N

Pri vzorkovaní n-z-N vyberáme presne n paketov z celkovej populácie N nameraných v meracom bode v istom meranom intervale. Môžeme približne určiť počet paketov narušiteľov v meranom intervale z počtu paketov narušiteľov SLA vo vzorke:

$$\hat{M} = \frac{N}{n} \cdot m = \frac{N}{n} \cdot \sum_{i=0}^n x_i$$

kde

$$x_i \sim \text{Be}(P)$$

Náhodná premenná X_i udáva prispôsobenie sa vzorkovaným paketom SLA (Service level agreement).

Pričom $X_i = 0$ ak parameter paketu QoS vyhovuje a

$X_i = 1$ ak parameter paketu QoS nevyhovuje.

X_i môže byť modelované Bernouliho distribučnou funkciou náhodnej premennej s pravdepodobnosťou $P=M/N$.

Približný podiel paketov porušujúcich SLA v meracom intervale je:

$$\hat{P} = \frac{\hat{M}}{N} = \frac{1}{n_R} \cdot \sum_{i=0}^{n_R} X_i$$

kde n_R je skutočná veľkosť vzorky.

Počet paketov narušiteľov m vo vzorke môže byť modelovaný ako počet zásahov v nezávislom experimente s n pokusmi. Pretože nemôžeme zvoliť paket ešte raz, musíme aplikovať selekciu bez náhrady, čiže m môže byť braná ako náhodná premenná s hypergeometrickou distribúciou. Podiel paketov narušiteľov P môžeme odhadnúť v meracom intervale pomocou podielu vyvzorkovaných paketov narušiteľov vo vzorke. Výraz n je veľkosť vzorky. Štandardná chyba je:

$$\sigma_{\hat{P}} = \sqrt{\frac{P \cdot (1-P)}{n}} \cdot \sqrt{1 - \frac{n}{N}}$$

Odhad presnosti závisí na veľkosti vzorky a skutočnom podiele paketov narušiteľov. Ak je veľkosť vzorky malá ($< 5\%$) tak môžeme zanedbať úpravu konečnej veľkosti vzorky. Skutočný podiel narušiteľov je neznámy a môže byť aproximovaný zo vzorky[20].

5.3.3 Pravdepodobnostné vzorkovanie

Pri pravdepodobnostnom vzorkovaní volíme paket na základe zadanej pravdepodobnosti bez ohľadu veľkosti populácie a počtu už zvolených paketov. Na základe toho je veľkosť vzorky rozdielna pri každom pokuse, v drvivej väčšine nebude rovná cieľovej veľkosti vzorky n . S veľkosťou celkovej populácie rastie

pravdepodobnosť sa priblížiť k cieľovej veľkosti vzorky n . V tomto priblížení sa bude veľkosť vzorky zanedbávať.

Počet paketov vyhovujúcich SLA je aproximovaný modelovaním procesu selekcie Bernouliho distribučnou funkciou náhodnej premennej ω_i s pravdepodobnosťou $f=n/N$. ω_i bude rovné 1 ak bude paket zvolený a rovné 0 ak paket nebude zvolený.

Na základe toho bude približný počet paketov narušujúcich SLA:

$$\hat{M} = \frac{N}{n} \cdot m = \frac{N}{n} \cdot \sum_{i=0}^M \omega_i$$

kde

$$\omega_i \sim \text{Be}\left(\frac{n}{N}\right)$$

Približný podiel paketov porušujúcich SLA v meracom intervale je:

$$\hat{P} = \frac{\hat{M}}{N} = \frac{1}{n_T} \cdot \sum_{i=0}^M \omega_i$$

kde n_T je cieľová veľkosť vzorky.

Štandardná chyba potom je:

$$\sigma_{\hat{P}} = \sqrt{\frac{P}{n}} \cdot \sqrt{1 - \frac{n}{N}}$$

5.3.4 Systematické vzorkovanie

Ak by všetky oneskorenia paketov v meracom intervale boli nezávislé, bolo by možné aplikovať rovnaký matematický model ako pri vzorkovaní n -z- N . Avšak ak existuje korelácia, systematická selekcia môže interferovať s periodicitou poradia paketov. V takomto prípade nedostaneme selekciou reprezentatívnu vzorku, ale môžeme dostať pakety s nejakou špeciálnou vlastnosťou (napr. s veľkým oneskorením) a tým pádom neobjektívne výsledky. Presnosť selekcie silne závisí na type prevádzky,

kvôli tomu je nemožné navrhnuť model popisujúci presnosť ako pri pravdepodobnostnom vzorkovaní.

5.3.5 Porovnanie efektivity vzorkovania n-z-N a pravdepodobnostného vzorkovania

Štandardná chyba vzorkovania n-z-N je rovná štandardnej chybe pravdepodobnostného vzorkovania vynásobeného výrazom $\sqrt{(1-P)}$. Keď $0 \leq \sqrt{(1-P)} \leq 1$ tak vzorkovanie n-z-N produkuje menšiu chybu ako pravdepodobnostné vzorkovanie a tým pádom je vhodnejšie na implementáciu.

5.4 Porovnanie vzorkovacích algoritmov

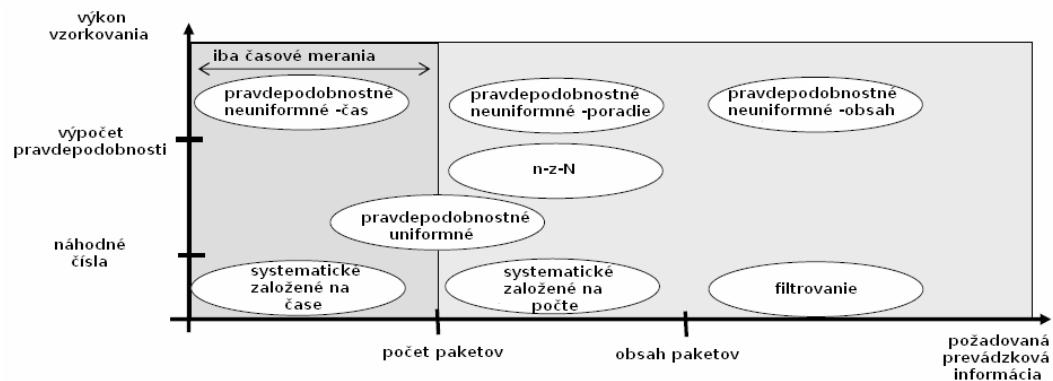
V nasledujúcej tabuľke 5.2 sú zhrnuté vstupné parametre a funkcie potrebné v meracom bode na implementáciu konkrétneho vzorkovacieho algoritmu.

vzorkovanie	vstupné parametre	funkcie	vhodné na implementáciu
Systematické založené na počte	pozícia paketu	počítadlo paketov	áno
Systematické založené na čase	čas príchodu paketu	časovač	áno
n-z-N	pozícia paketu	počítadlo paketov, generátor náhodných čísel	áno (s obmedzenou veľkosťou vzorky)
Uniformné pravdepodobnostné	pravdepodobnosť vzorkovania	generátor náhodných čísel	áno

Neuniformné pravdepodobnostné	pozícia paketu	výberová funkcia, výpočet pravdepodobnosti	nie
Neuniformné, závislé od stavu toku	stav toku, časti paketu	výberová funkcia, výpočet pravdepodobnosti	nie

Tab. 5.2: Vzorkovacie algoritmy a ich požiadavky na implementáciu

Na nasledujúcom obrázku 5.2 je opísaná závislosť výkonu vzorkovania od jednotlivých vstupných parametrov.



Obr. 5.2: Presnosť vzorkovacích algoritmov

Ako algoritmy vhodné na implementáciu boli zvolené:

Systematické založené na počte

Systematické založené na čase

Vzorkovanie n-z-N

Uniformné pravdepodobnostné

Neuniformné pravdepodobnostné založené na počte

Vstupné parametre pre jednotlivé algoritmy budú načítané z konfiguračného súboru.

Systematické vzorkovanie založené na počte a na čase bolo zvolené pre možnosť jednoduchšej implementácie a jeho celkovú nenáročnosť na systémové zdroje.

Vzorkovanie n-z-N je na implementáciu vhodné, avšak pri jeho použití sme obmedzený veľkosťou testovaného súboru paketov. Na použitie v nepretržitej prevádzke sa však nehodí kvôli obmedzenej veľkosti celkovej populácie N.

Vzorkovanie s náhodnou uniformnou pravdepodobnosťou bolo zvolené ako jeden z možných kandidátov na optimálnu metódu z hľadiska výpočtovej náročnosti a náhodnosti vybranej vzorky. Pod výpočtovou náročnosťou rozumieme počet cyklov procesora potrebných na vykonanie daného algoritmu.

Vzorkovanie s náhodnou neuniformnou pravdepodobnosťou je vhodné na implementáciu, avšak jeho náročnosť na výpočtové prostriedky bude oproti vzorkovanu s náhodnou uniformnou pravdepodobnosťou väčšia.

Vzorkovanie s náhodnou neuniformnou pravdepodobnosťou závislé od stavu toku je v súčasnej fáze vývoja meracieho nástroja „basicmeter“ nerealizovateľné, nakoľko návrh meracieho nástroja s jeho využitím nerátal.

Treba podotknúť že vzorkovacie algoritmy boli posudzované z hľadiska rýchlosti na použitie vo vysoko rýchlostných sieťach typu 1Gbit Ethernet a 10Gbit Ethernet. Pri takýchto rýchlostiach a pri priemernej veľkosti paketu 1KB je bežná prenosová rýchlosť 1,250 milióna rámcov za sekundu, čo kladie veľmi vysoké nároky na výpočtový proces. Cieľom meracieho nástroja „basicmeter“ je vytvorenie jednoduchšej meracej platformy ktorá nebude vyžadovať drahý a náročný hardvér, nakoľko meranie prevádzkových parametrov je len doplnková funkcia.

Na použitie v meracom nástroji „Basicmeter“ bol z hľadiska efektivity výberu reprezentatívnej vzorky zvolený algoritmus neuniformného pravdepodobnostného vzorkovania založeného na počte paketov pre použitie v kontinuálnych dlhodobých meraniach (napr. 1 týždeň).

Pre merania kde postačuje fixná veľkosť vzorky bolo ako najvhodnejšie vzorkovanie zvolené vzorkovanie n-z-N.

Experimentálne meranie časovej náročnosti jednotlivých algoritmov

Pre účel otestovania časovej náročnosti jednotlivých vzorkovacích algoritmov bol navrhnutý program, ktorý tvorí zjednodušený modul odchyťovania, filtrovania a vzorkovania paketov v testovacom nástroji „basicmeter“. Jeho úlohou bude otestovať časovú náročnosť implementovaných algoritmov za rovnakých podmienok na jednotnej populácii paketov.

Vstupom pre meranie sú nasledujúce parametre:

- Súbor s 1 000 000 paketmi získaný pomocou programu TCPDUMP
- Filter nastavený na akceptovanie všetkých paketov.

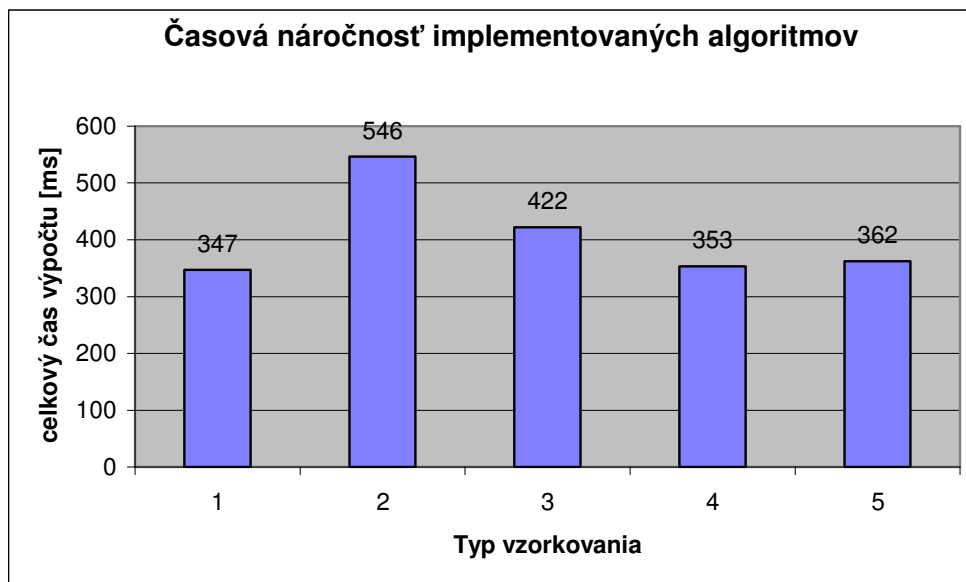
Súbor bol pred meraním načítaný do RAM disku, čo je technológia ktorá vyhradí miesto v operačnej pamäti a používa ho ako klasický suborový systém (napr. Ext3, FAT32). Výhodou je podstatné zvýšenie rýchlosti čítania zo súboru.

Merania prebehli na stroji s konfiguráciou AMD Duron 750MHz, 512MB SDRAM a 40GB pevným diskom.

Jednotlivé vstupné parametre pre vzorkovacie algoritmy boli zvolené tak, aby cieľová vzorka paketov tvorila 10 % celkovej populácie, čo vyhovuje podmienke definovanej v časti 5.3.1, sú v tabuľke 5.3. V tabuľke 5.3 sú uvedené aj výsledné namerané hodnoty doby vykonávania algoritmov v užívateľskom priestore (user-space) a v priestore jadra (kernel-space). Obrázok 5.3 zobrazuje celkový čas potrebný na vykonanie vzorkovania pre dané typy algoritmov.

typ alg.	vzorkovanie	1. param.	2. param.	čas výpočtu v userspace [ms]	čas výpočtu v kernel-space [ms]	celkový čas výpočtu [ms]	čas výpočtu v porovnaní s 2.algoritmom
1	systematické vzorkovanie založené na počte paketov	9	1	140	207	347	64%
2	systematické vzorkovanie založené na čase príchodu paketov	9	1	165	381	546	100%
3	vzorkovanie n_z_N	100000	1000000	214	208	422	77%
4	uniformné náhodné pravdepodobnostné vzorkovanie	10	-	149	204	353	65%
5	neuniformné náhodné pravdepodobnostné vzorkovanie založené na počte paketov	20	2	161	201	362	66%

Tab. 5.3: Časová náročnosť jednotlivých algoritmov vzorkovania



Obr. 5.3: Celková časová náročnosť algoritmov vzorkovania

Z nameraných výsledkov vyplýva, že algoritmus systematického vzorkovania založeného na čase príchodu paketov má najväčšiu časovú náročnosť, ďalej nasleduje

algoritmus vzorkovania n -z- N , a po ňom takmer s rovnakou časovou náročnosťou algoritmy uniformného náhodného pravdepodobnostného vzorkovania, neuniformného náhodného pravdepodobnostného vzorkovania založeného na počte paketov a systematického vzorkovania založeného na počte paketov.

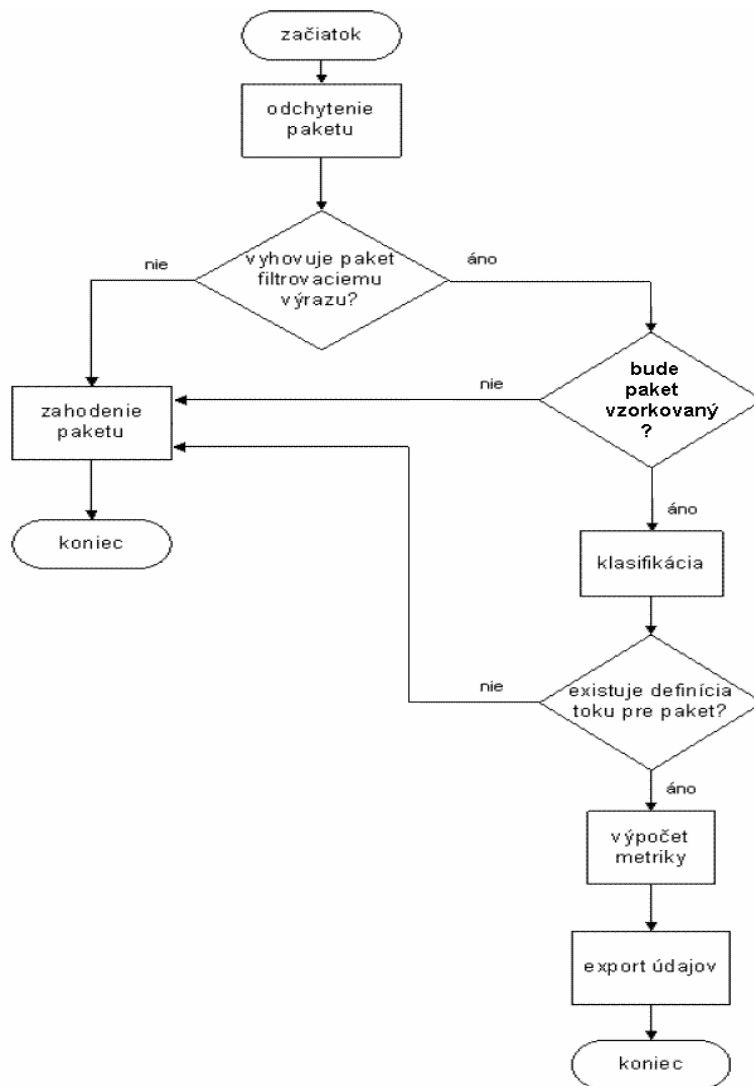
Po zvážení vyššie uvedených faktov a celkovej efektivity selekcie paketu možno konštatovať, že pre použitie v meracom nástroji je najvhodnejší algoritmus neuniformného náhodného pravdepodobnostného vzorkovania založeného na počte paketov. Daný algoritmus má nízku časovú náročnosť a jeho efektívnosť vzorkovania je s pomedzi testovaných algoritmov najväčšia.

6. Implementácia vybraných vzorkovacích algoritmov pre potreby nástroja „Basicmeter“

Pre implementáciu bol zvolený programovací jazyk C++, najmä z dôvodu prenositeľnosti medzi jednotlivými typmi unixového operačného systému. Pri implementácii boli použité štandardné funkcie operačného systému a voľne šíriteľné knižnice dostupné vo forme zdrojových kódov.

6.1 Algoritmus spracovania paketu

Algoritmom spracovania paketu je konceptuálne popísaná činnosť meracieho nástroja. V tomto algoritme nie je popísaná inicializačná časť, kde prebieha inicializácia jednotlivých modulov architektúry meracieho nástroja, teda nastavenie príslušného rozhrania, načítanie šablón a nadviazanie spojenia na zhromažďovací proces (collector). Samotný algoritmus spracovania paketu v schematickom vyjadrení značiek vývojového diagramu je znázornený na obrázku 6.1. Algoritmus spracovania paketu znázorňuje cestu paketu od jeho odchytenia, filtrovania, vzorkovania až po prípadné zaradenie do toku alebo odmietnutie. Paket je po odchytení skontrolovaný voči nadefinovaným pravidlám pre odchyťovanie (filtrovanie prevádzky pri odchyťovaní paketov). V prípade, že vyhoví, je vzorkovaný preddefinovaným algoritmom. Po vzorkovaní je paket klasifikovaný. Klasifikácia spočíva v porovnaní položiek paketu s filtrovacími výrazmi pre jednotlivé toky. V kladnom prípade sú informácie z paketu extrahované do položiek v toku, tak ako sú zadané načítanou a inicializovanou šablónou. Po naplnení zásobníka tokov sú záznamy exportované pomocou protokolu NetFlow verzie 9 na proces kolektora. Tento proces sa opakuje dovtedy, pokiaľ nie je činnosť meracieho nástroja ukončená používateľom alebo počet odchytených paketov nedosiahne veľkosť N pri vzorkovaní n -z- N .



Obr. 6.1: Algoritmus spracovania paketu

6.2 Knižnica Libpcap

Ako dôsledok zvyšovania požiadaviek na tvorbu používateľských nástrojov pre odchytyvanie sieťovej prevádzky, vznikla knižnica libpcap [13] ako prenositeľné, systémovo nezávislé rozhranie pre odchytyvanie paketov na používateľskej úrovni. Táto knižnica pre odchytyvanie paketov poskytuje vysokoúrovňové rozhranie pre programy

odchytávajúce pakety. Všetky pakety v sieti, dokonca aj tie, ktoré sú určené pre iné počítače sú prístupné pomocou tohto mechanizmu po prepnutí rozhrania do promiskuitného režimu. K paketom je možné pristupovať dvomi spôsobmi: čítať ich priamo zo sieťového rozhrania, alebo ich čítať zo súboru získaného pomocou programu TCPDUMP[9]. Zápis paketov do súboru je realizovaný funkciami libpcap knižnice. Nie je nutné odchytať celé veľkosti paketov, ale len postačujúcu vopred definovanú veľkosť. Libpcap knižnica umožňuje špecifikovať filtrovací výraz, ktorý sa použije na odchytenie špecifikovaných paketov. Filtrovanie je realizované pomocou Berkley Packet Filter (BPF) [2]. BPF je často implementovaný v jadre operačného systému. V takom prípade knižnica použije implementáciu v jadre, pretože filtrovanie je rýchlejšie, prenáša sa menej dát z priestoru jadra do priestoru používateľa a uskutoční sa menej prepnutí kontextu, čím sa zvyšuje výkon procesu odchyťovania a znižuje sa zaťaženie počítača. BPF je založené na abstraktnom modeli orientovaného acyklického CFG (Control Flow Graph) grafu, ktorý je použitý na vytvorenie abstraktného stroja s registrovým pseudo-jazykom. Program v tomto jazyku sa používa na zistenie, či je paket akceptovaný filtrom. Knižnica libpcap obsahuje kompilátor a optimalizátor, ktorý prekladá vysokoúrovňový popisný jazyk používaný na používateľský príjemný zápis filtrovacích výrazov do programu BPF. Tento preklad je voči používateľovi transparentný.

6.3 Požiadavky kladené na jednotlivé časti

capture

možnosť definovať zdroj pre odchyťovanie dát (súbor, sieťové rozhranie)

možnosť definovať pravidlá filtrovania pomocou syntaxe tcpdumpu (libpcap)

umožniť jednoduché napojenie triedy sample

načítavanie pravidiel zo vstupného konfiguračného súboru

sample

možnosť definovať vzorkovaciu metódu a jej parametre
načítavanie parametrov zo vstupného konfiguračného súboru

6.4 Návrh a analýza časti capture

Časť capture má za úlohu realizovať samotné odchytenie paketu a de facto jeho vstup do basicmetra. Samotné odchytenie paketu je možné realizovať viacerými spôsobmi, dokonca na viacerých úrovniach (userspace alebo kernelpspace), ukazuje sa však, že pre efektívnu činnosť časti capture prichádza do úvahy prakticky jedine kernelpspace, kde narážame na nutnosť modifikácie jadra a s tým súvisiacich problémov s portabilitou na jednotlivé architektúry, so zložitejším vývojom (do istej miery obmedzené pamäťové možnosti a prísnejšie pravidlá vývoja v kernelpspace) ako aj s celkovou problematickou stabilitou takéhoto riešenia. Preto možnosti odchytenia paketu vyžadujúce modifikáciu jadra vopred zamietame, pre úplnosť ich však vymenujeme.

Existujú nasledujúce spôsoby a postupy:

- knižnica libpcap
- iptables
- traffic controlling
- návrh vlastného spôsobu

Analýza vstupných dát

Definícia vstupného filtra: pri časti capture je obzvlášť dôležitá efektívnosť odchyťovania paketov, preto je aj celý návrh zameraný najmä na rýchlosť riešenia. Jedným z významných parametrov ovplyvňujúcich rýchlosť je množstvo dát na vstupe. Množstvo dát je možné obmedziť vhodným filtrom. Štandardným, preddefinovaným filtrom je filter na obmedzenie dátového toku na IP prevádzku.

Analýza výstupných dát

- timestamp

- paket
- paket_id

6.5 Návrh a analýza časti sample

V časti sample bude realizované samotné vzorkovanie, čiže rozhodnutie o zvolení paketu.

Analýza vstupných dát

Vstupom pre proces vzorkovania sú vzorkovacie parametre. Keďže v nástroji basicmeter nevyužívame časť paketu ako súčasť generátora náhodných čísel, nie je potrebné aby vzorkovací algoritmus vedel o konkrétnom pakete.

Analýza výstupných dát

Výstupom vzorkovacieho procesu bude príznak toho, či daný paket bude vzorkovaný a teda poslaný na ďalšie spracovanie alebo nebude vzorkovaný a tým pádom zahodený.

6.6 Experimentálne overenie funkčnosti nástroja a výsledkov vzorkovania

Experimentálnym overovaním funkčnosti sa rozumie inštalácia merača do infraštruktúry laboratórneho segmentu a jeho používanie a získavanie dát pre ďalšie vyhodnocovanie.

6.6.1 Inštalácia

Pre testovacie účely bol merací nástroj nainštalovaný na počítač sisa2.cnl.tuke.sk umiestnený v Laboratóriu počítačových sietí na Katedre počítačov a informatiky

Fakulty elektrotechniky a informatiky Technickej univerzity v Košiciach. Počítač bol do infraštruktúry laboratórneho segmentu pripojený cez zariadenie nazývané ethernet tap. Ethernet tap je aktívne zariadenie pracujúce na prvej vrstve ISO/OSI modelu. Toto zariadenie je možné transparentne pripojiť na linku typu full duplex Ethernet. Všetky prechádzajúce signály z tejto linky sú zachytávané a preposielané na odbočku.

V laboratóriu počítačových sietí bol inštalovaný PC s operačným systémom Linux (procesor Intel Celeron 400 MHz, sieťové rozhranie 3Com 3c905C-TX, Linux kernel 2.6.7, distribúcia Linuxu Debian GNU/Linux. V laboratóriu sa nachádza niekoľko lokálnych počítačových sietí na báze Ethernetu prepojených switchom Cisco Catalyst 4006. Pripojenie do verejnej siete je realizované pripojením do lokálnej siete Katedry počítačov a informatiky, ktorá je súčasťou univerzitnej siete TUNET.

6.6.2 Experimentálne meranie v laboratórnom segmente

Experimentálne meranie, odlaďovanie a testovanie funkčnosti prebiehalo v zapojení basicmeter, kolektor a vyhodnocovacia aplikácia. Vzhľadom na to, že nástroj bol navrhovaný ako súčasť komplexnej meracej platformy s ohľadom na PSAMP a IPFIX špecifikáciu, samotný nástroj neposkytuje žiadne možnosti analýzy prevádzky. Preto ako zhromažďovací proces bol použitý programový balík Jxcoll. Zhromažďované dáta boli posielané vyhodnocovacej aplikácii, ktorá v reálnom čase poskytovala výsledky o aktuálnej prevádzke na sieti, vyhodnocovacia aplikácia však nebola schopná interpretovať vzorkovanie, takže namerané výsledky nebudú zodpovedať zrekonštruovanému toku, ale len toku vzorkovaných paketov. Maximálne zaťaženie procesora na ktorom bežal basicmeter dosahovalo 4%.

6.6.3 Meranie využitia šírky pásma

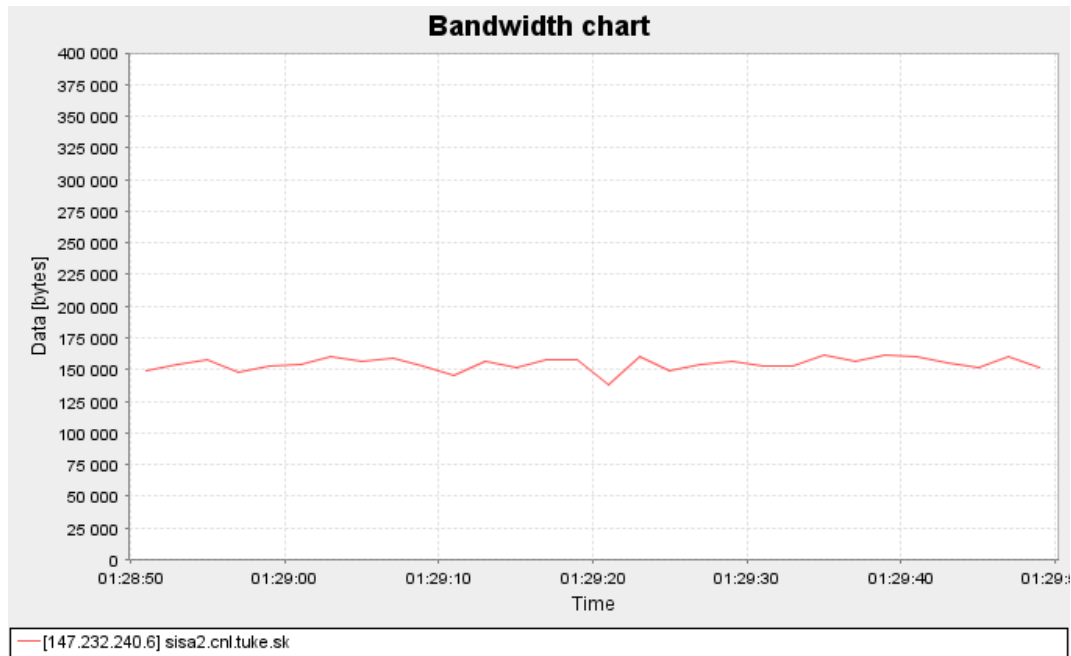
Pri meraní využitia šírky pásma sa pozornosť sústredila na využitie šírky pásma službou ftp počas doby 60 sekúnd(obr. 6.12). Testovacia prevádzka bola obmedzená na maximálny tok 1024kB za sekundu. Následne bolo na túto testovaciu prevádzku aplikované vzorkovanie pomocou implementovaných algoritmov vzorkovania.

Jednotlivé parametre vzorkovania sú uvedené v tabuľke 6.1. Parametre boli volené tak, aby zodpovedali požadovanej minimálnej veľkosti vzorky podľa kapitoly 5.3.1. a aby cieľová vzorka tvorila 20 % z celkovej populácie.

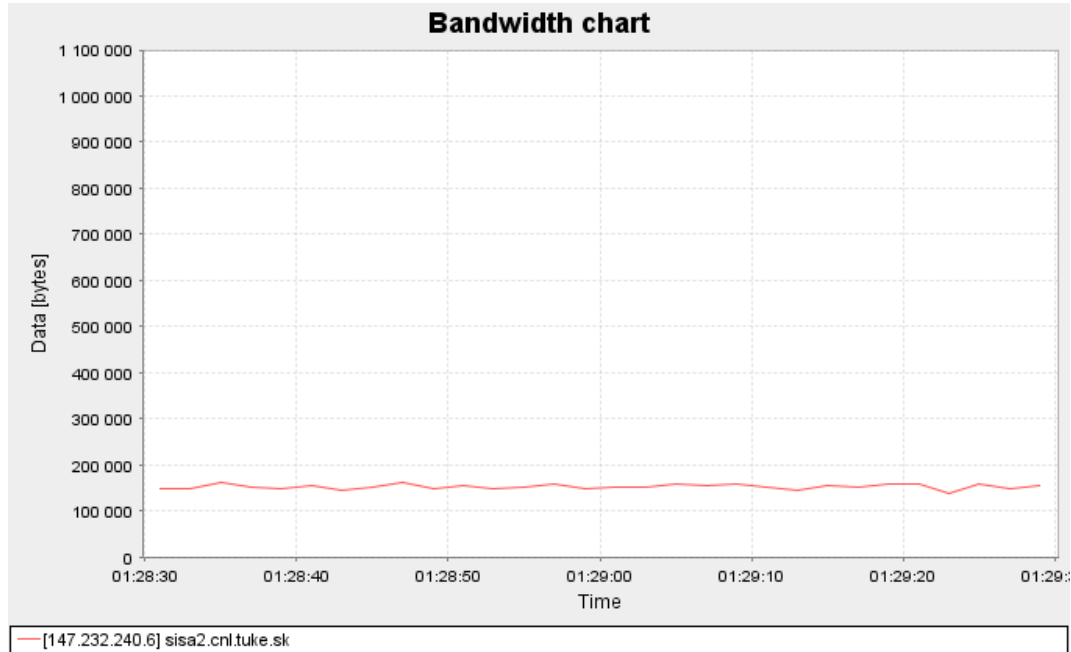
Vzorkovací algoritmus	Sampling parameter <i>hodnota</i>	Sampling parameter2 <i>hodnota</i>	Obrázok s maximom 400KB/s	Obrázok s maximom 1100KB/s
systematické vzorkovanie založené na počte paketov	Dĺžka intervalu počas ktorého sa nevzorkuje v paketoch 4	Dĺžka intervalu vzorkovania v paketoch 1	6.2	6.3
systematické vzorkovanie založené na čase príchodu paketov	Dĺžka intervalu vzorkovania v sekundách 4	Dĺžka intervalu vzorkovania v sekundách 1	6.4	6.5
vzorkovanie n_z_N	Veľkosť vzorky n 200 000	Veľkosť celkovej populácie N 1 000 000	6.6	6.7
uniformné náhodné pravdepodobnostné vzorkovanie	Pravdepodobnosť selekcie paketu v % 20	Nezáleží -	6.8	6.9
neuniformné náhodné pravdepodobnostné vzorkovanie založené na počte	Pravdepodobnosť selekcie paketu v % 40	Poradie vzorkovaného paketu 2	6.10	6.11

Tab. č. 6.1: Popis konfiguračných direktív vzorkovania

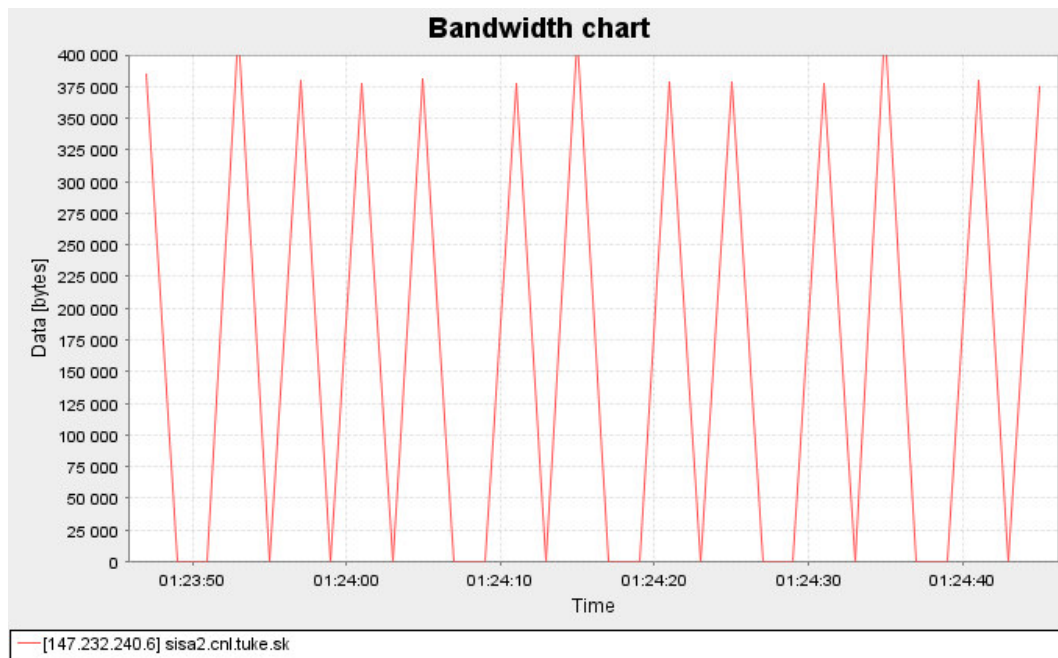
Výsledky merania sú naznačené v grafoch očíslovaných podľa tabuľky 6.1.



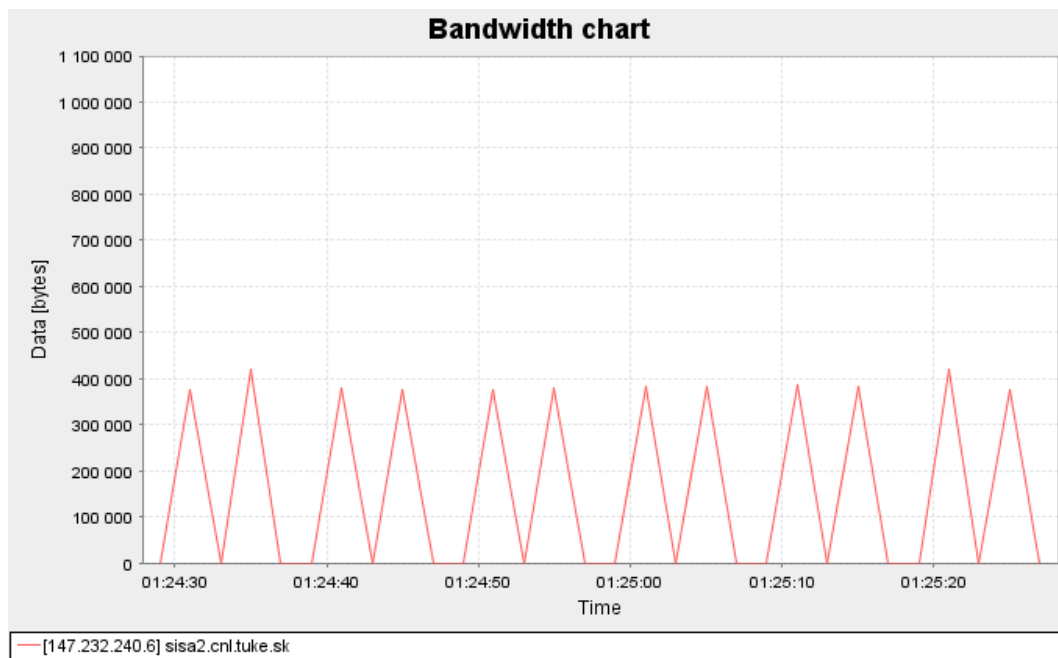
Obr. 6.2: Systematické vzorkovanie založené na počte s maximom v 400KB/s



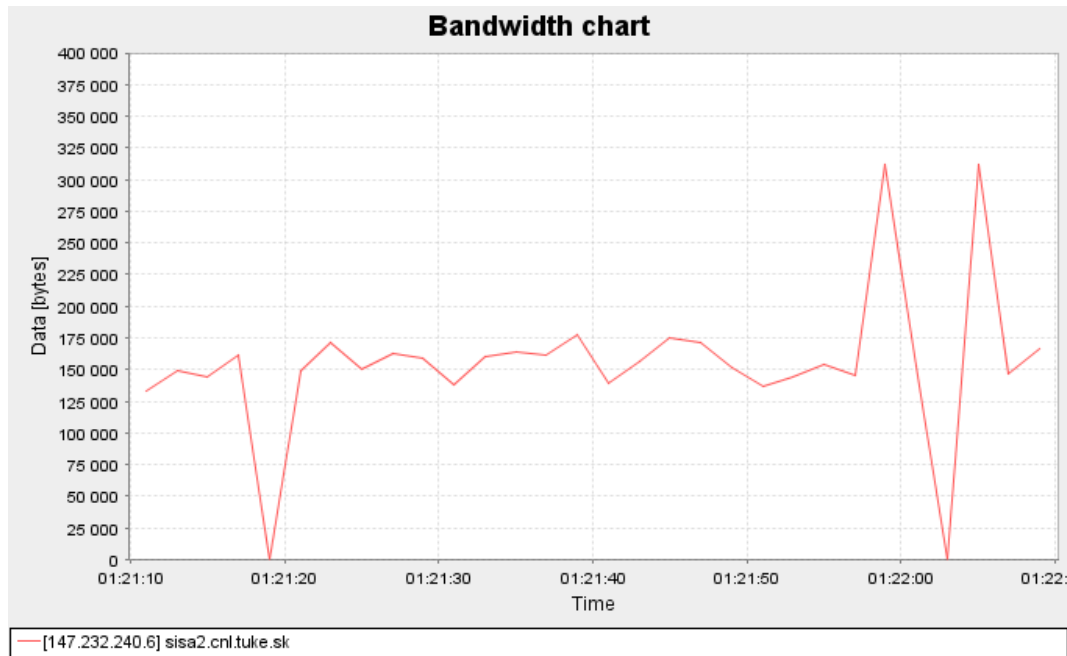
Obr. 6.3: Systematické vzorkovanie založené na počte s maximom v 1100KB/s



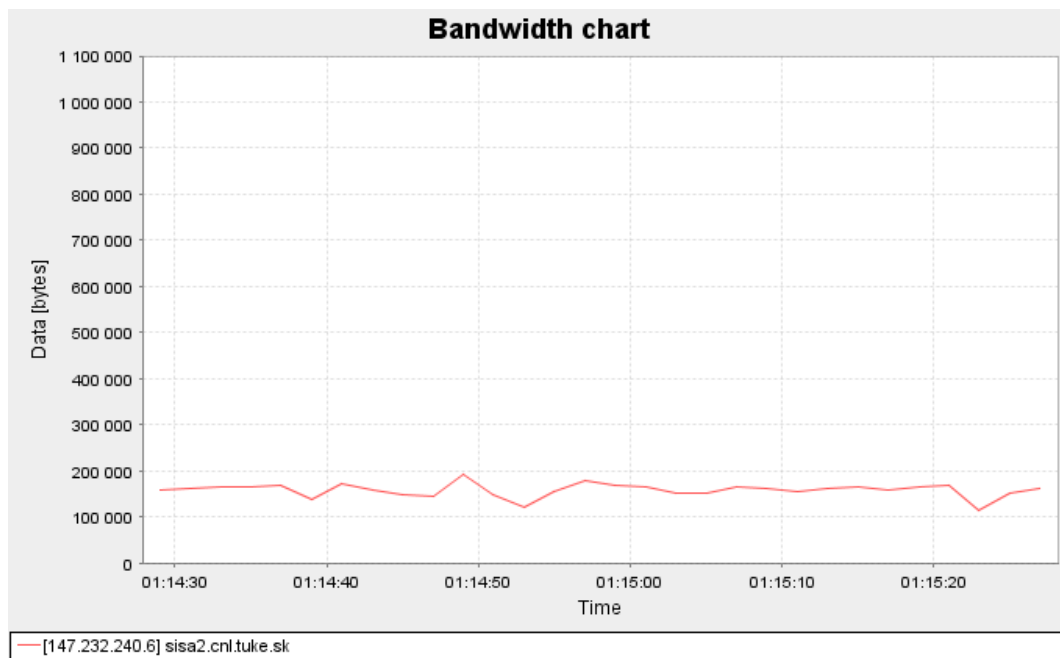
Obr. 6.4: Systematické vzorkovanie založené na čase s maximom v 400KB/s



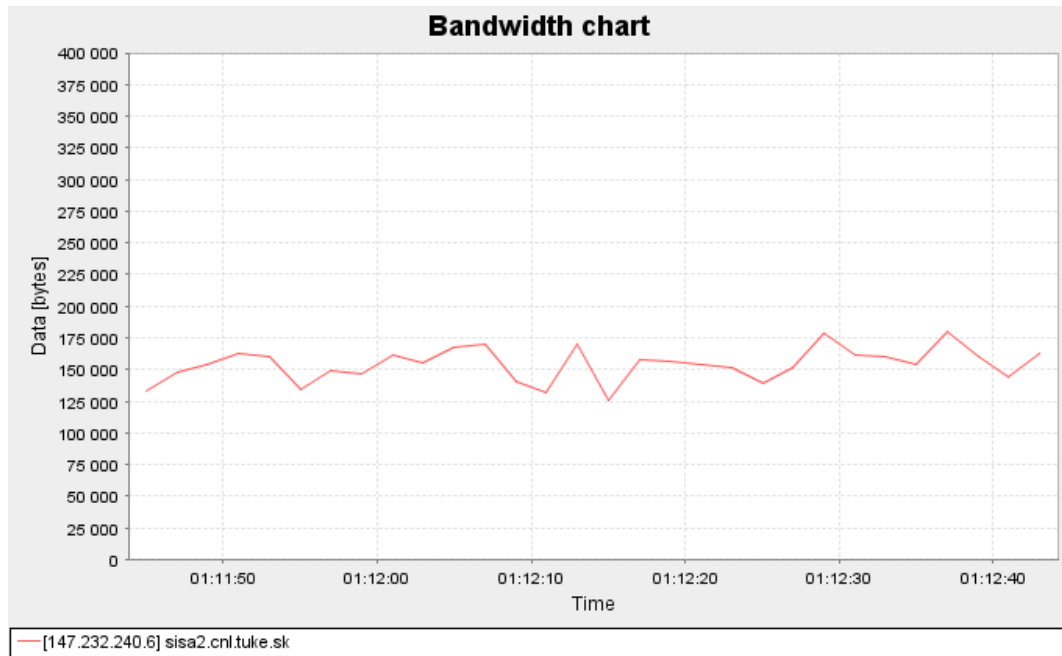
Obr. 6.5: Systematické vzorkovanie založené na čase s maximom v 1100KB/s



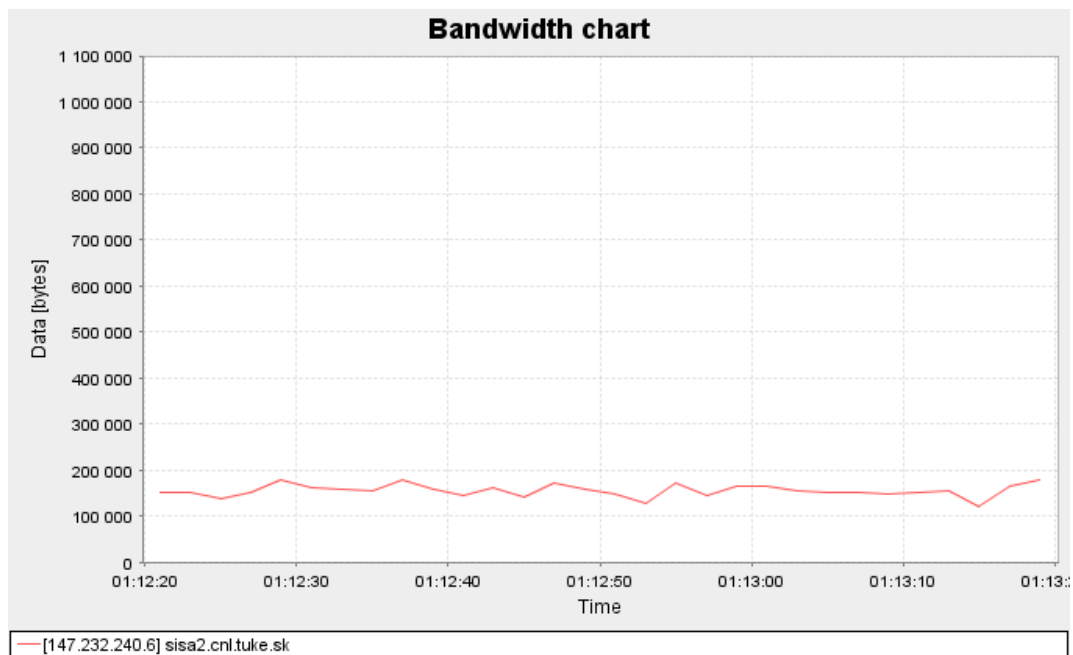
Obr. 6.6: Vzorkovanie n-z-N s maximom v 400KB/s



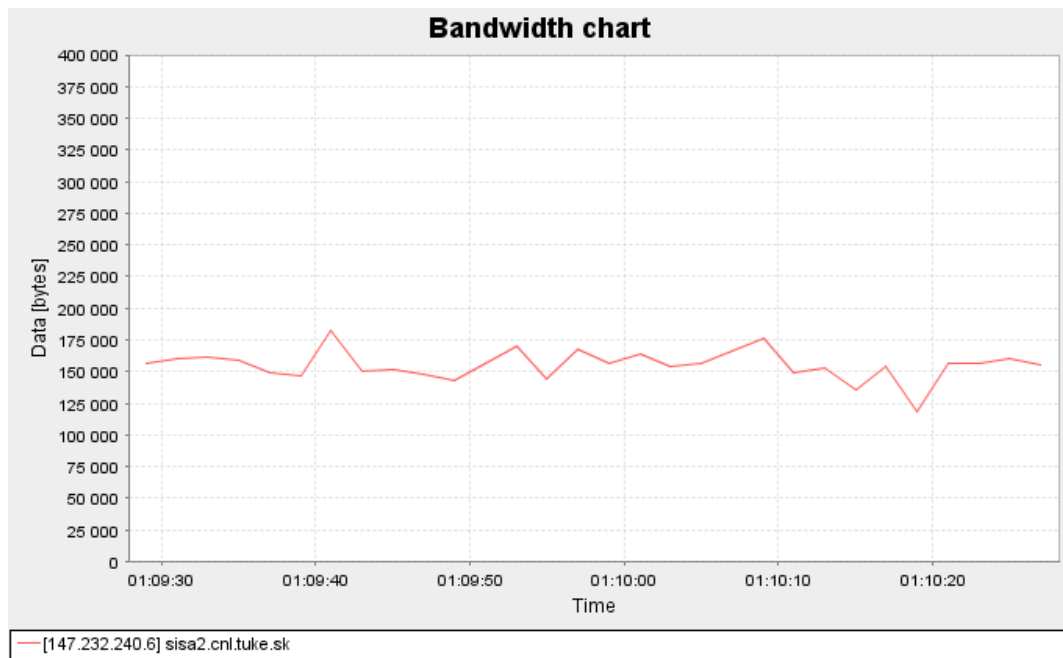
Obr. 6.7: Vzorkovanie n-z-N s maximom v 1100KB/s



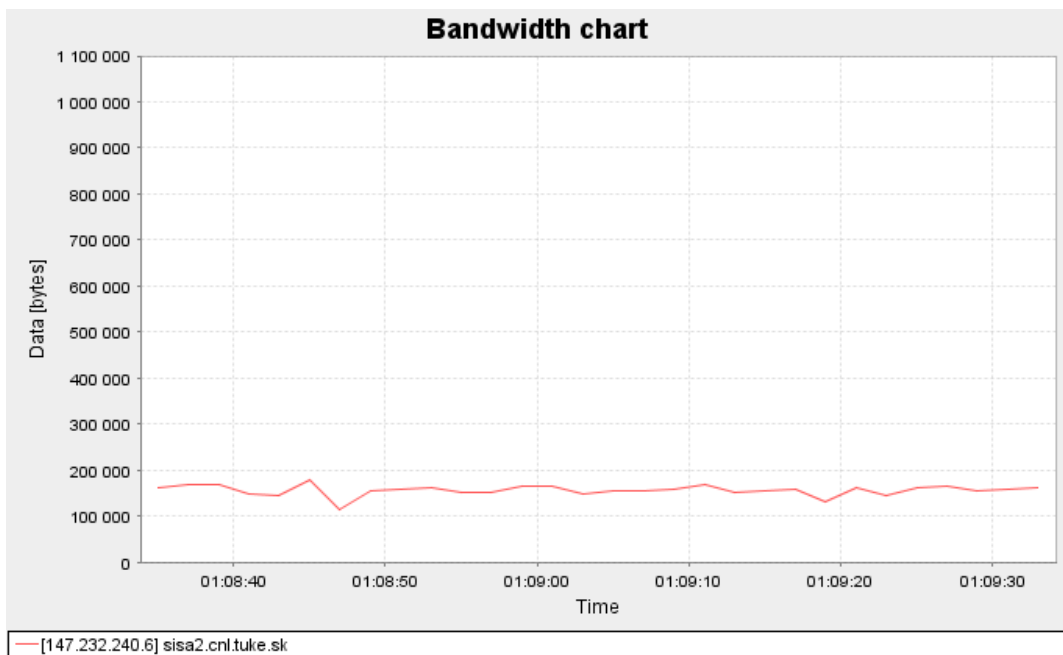
Obr. 6.8: Uniformné náhodné pravdepodobnostné vzorkovanie s maximom v 400KB/s



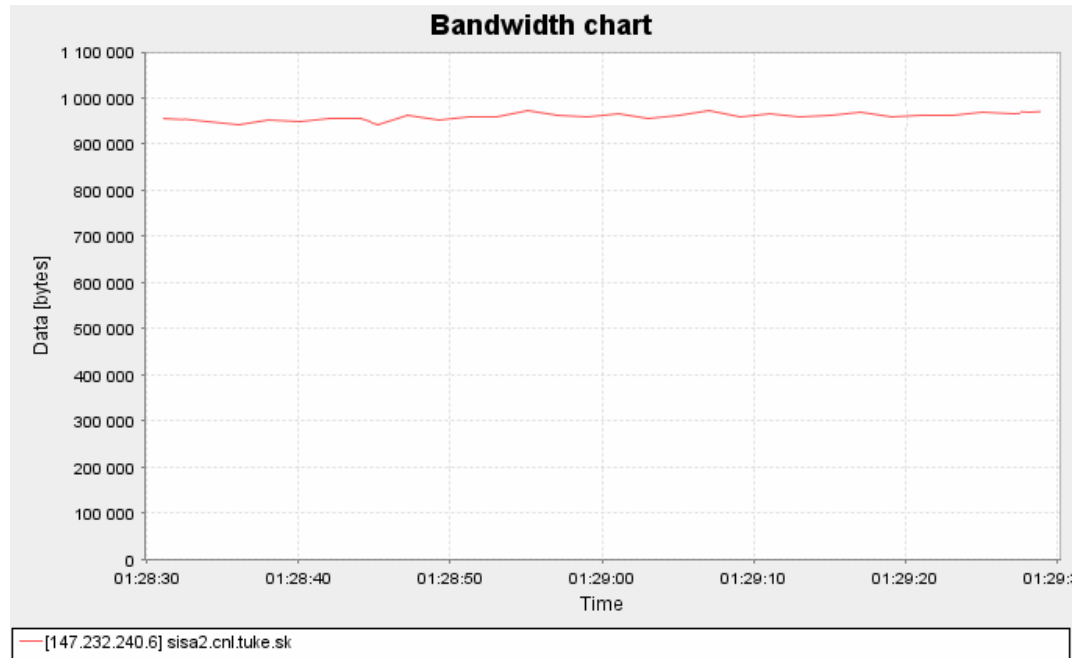
Obr. 6.9: Uniformné náhodné pravdepodobnostné vzorkovanie s maximom v 1100KB/s



Obr. 6.10: Neuniformné náhodné pravdepodobnostné vzorkovanie založené na počte s maximom v 400KB/s



Obr. 6.11: Neuniformné náhodné pravdepodobnostné vzorkovanie založené na počte s maximom v 1100KB/s



Obr. 6.12: Tok celkovej populácie s maximom v 1100KB/s

Z nameraných výsledkov vyplýva, že systematické vzorkovanie založené na čase má kvôli dlhým časovým intervalom vzorkovania a intervalom kedy sa nevzorkuje svoj charakteristický pilovitý priebeh, ktorý možno optimalizovať zmenšením intervalov, resp. zväčšením ich rozlišovacej schopnosti zo sekúnd na milisekundy. Tento priebeh však možno upraviť aj pomocou vyhodnocovacej aplikácie, ak nastavíme intervaly zobrazovania na väčšiu hodnotu.

Systematické vzorkovanie založené na počte veľmi dobre kopírovalo tok celkovej populácie nakoľko veľkosti paketov a frekvencia ich prenosu boli takmer konštantné. V prípade, že by sa vyskytla v sieti prevádzka s istou systematickosťou, tak systematické vzorkovanie na základe zvolených parametrov korelujúcich zo systematickosťou sledovaného toku, poskytovalo výsledky značne sa odlišujúcich od toku celkovej populácie.

Vzorkovanie n -z- N poskyto výsledky mierne sa odlišujúce od celkovej populácie, avšak pri stochastickom celkovom toku by jeho výsledky boli podstatne presnejšie ako pri systematickom vzorkovaní.

Uniformné náhodné pravdepodobnostné vzorkovanie poskyto výsledky v súlade s predpokladom, a to že jeho priebeh nebude presne zodpovedať prevádzke k približne konštantnou veľkosťou paketu a rovnakou frekvenciou ich vysielania.

Neuniformné náhodné pravdepodobnostné vzorkovanie založené na počte paketov poskyto veľmi podobné výsledky ako uniformné náhodné pravdepodobnostné vzorkovanie. Jeho použitie na akýkoľvek typ prevádzky vzhľadom na jeho náhodnosť selekcie paketu pozostávajúcu z dvoch navzájom nezávislých procesov možno len odporúčať.

Cieľom tohto experimentu bolo poukázať na náhodnosť selekcie paketu pri použití algoritmov vzorkovania ktoré pracujú s generovaním náhodných čísel, čo sa nám podarilo preukázať, nakoľko tieto algoritmy nekopírovali pôvodnú celkovú populáciu presne, ale s istou odchýlkou a práve táto odchýlka umožňuje ich použitie aj na sledovanie systematických tokov.

Presnosť tohto merania by sa dala zvýšiť, ak by jednotlivé algoritmy vzorkovania prebehli na identickej celkovej populácii načítanej zo súboru, avšak súčasný stav basicmetra nám neumožňuje takéto meranie z dôvodu neúplnej implementácie štandardov PSAMP a IPFIX.

7. Zhodnotenie riešenia

Predložená práca je venovaná analýze a popisu použitia vzorkovacích algoritmov pre pasívne merania v počítačových sieťach. Pozornosť je zameraná na implementáciu novo vznikajúceho štandardu PSAMP a IPFIX. V rámci práce boli analyzované rôzne typy vzorkovaní, pričom sa vychádzalo z použitia matematického popisu pomocou štatistických metód.

Praktickým výsledkom diplomovej práce je vytvorenie modulárnej architektúry pre realizáciu neintruzívnych meraní parametrov kvality služieb s podporou pre vzorkovanie. Program v súčasnosti dokáže merať objemové a časové charakteristiky prevádzky. Návrh architektúry basicmetra je založený na novo vznikajúcich štandardoch PSAMP a IPFIX, avšak vzhľadom na komplexnosť týchto štandardov nie je basicmeter ich presnou implementáciou. Modul pre podporu vzorkovania bol navrhnutý tak, aby v budúcnosti bol jednoducho rozšíriteľný o ďalšie vzorkovacie algoritmy.

S Basicmetrom boli realizované merania QoS parametrov dátovej prevádzky z Laboratória počítačových sietí. Pri experimentoch sa použila simulovaná prevádzka pomocou protokolu ftp. Merač nástroj poskytoval reálne výsledky o dátových tokoch v laboratóriu. Pri použití implementovaného vzorkovania je možné tento nástroj používať ako merač QoS parametrov vo vysokorýchlostných sieťach typu 1 Gbit a 10Gbit Ethernet.

8. Zoznam použitej literatúry

- [1] BRAY, T. et al.: Extensible Markup Language (XML) 1.1 [online] Publikované vo apríli 2004. [citované 7.4.2005] URL <http://www.w3.org/TR/xml11/>
- [2] Steven McCanne - Van Jacobson. The BSD Packet Filter: A New Architecture for User-level Packet Capture. Lawrence Berkeley Laboratory. Berkeley. December 1992.
- [3] NetFlow Services and Applications, White Paper, Cisco Systems, 1999, URL http://www.cisco.com/warp/public/cc/pd/iosw/ioft/neflct/tech/napps_wp.htm
- [4] Clark, J., Murata, M.: RELAX NG Specification [online] Publikované v decembri 2001. [citované 4.2.2005]. URL <http://www.relaxng.org/spec-20011203.html>
- [5] Duffield, N., Grossglauser, M.: Trajectory Sampling for Direct Traffic Observation. In: Proceedings of ACM SIGCOMM 2000, Stockholm, Sweden, august 2000
- [6] J. Quittek, J., Briant, S., Molina, Information Model for IP Flow Information Export [online] Publikované vo februári 2005. [citované 9.4.2005] URL <http://www.ietf.org/internet-drafts/draft-ietf-ipfix-info-06.txt>
- [7] McCANNE, S. et al.: Libpcap — library for capturing packets [knižnica] Ver 0.8.3 URL: <http://www.tcpdump.org/release/libpcap-0.8.3.tar.gz>
- [8] SADASIVAN, G. — BROWNLEE, N.: Architecture for IP Flow Information Export [online] Publikované v marcii 2005. [citované 7.4.2005] URL <http://www.ietf.org/internet-drafts/draft-ietf-ipfix-architecture-07.txt>
- [9] <http://www.tcpdump.org/daily/tcpdump-current.tar.gz>
- [10] VEILLARD, D.: The XML C parser and toolkit of Gnome — libxml [knižnica] Ver2.6.19 [citované 9.4.2005]. URL <http://xmlsoft.org/sources/libxml2-2.6.19.tar.gz>
- [11] HEDENFALK, M.: Confuse - simple configuration file parser library [online] Publikované 17.10.2004. [citované 3.2.2005] URL www.nongnu.org/confuse/

- [12] Duffield N., A Framework for Passive Packet Measurement [online] Publikované vo februári 2002. [citované 14.4.2005] URL <http://psamp.ccrle.nec.de/drafts/draft-duffield-framework-papame-01.txt>
- [13] <http://www.tcpdump.org/release/libpcap-0.8.3.tar.gz>
- [14] Zseby, T., Molina, M., Sampling and Filtering Techniques for IP Packet Selection [online] Publikované vo februári 2005. [citované 9.4.2005] URL <http://www.ietf.org/internet-drafts/draft-ietf-psamp-sample-tech-06.txt>
- [15] Mills, D., "Network Time Protocol (v3)", RFC 1305, April 1992.
- [16] Rivest, R., The MD5 Message-Digest Algorithm, RFC 1321, MIT and RSA Data Security, Inc., April 1992.
- [17] Eastlake, D., Jones, P., US Secure Hash Algorithm 1 (SHA1), RFC 3174, Motorola, Cisco Systems, September 2001.
- [18] Klíma, V., jak se melou data, CHIP, str. 44-46, apríl 1999
- [19] J. Quittek, On the Relationship between PSAMP and IPFIX [online] Publikované vo októbri 2002. [citované 14.4.2005] URL <http://psamp.ccrle.nec.de/drafts/draft-quittek-psamp-ipfix-00.txt>
- [20] Zander, S., Zseby, T., Sampling Schemes for Validating Service Level Agreements [online] Publikované vo júli 2004. [citované 10.4.2005] URL <http://caia.swin.edu.au/reports/040706A/CAIA-TR-040706A.pdf>
- [21] Zseby, T., Deployment of Sampling Methods for SLA Validation with Non-Intrusive Measurements [online] Publikované vo februári 2002. [citované 10.4.2005] URL http://www.labs.agilent.com/pam2002/proceedings/Deployment_of_Sampling_Methods_for_SLA_Validation.pdf
- [22] Almes, G., Kalidindi, S., Zekauskas, M.: A Round-trip Delay Metric for IPPM, RFC 2681, September 1999
- [23] Duffield N., A Framework for Packet Selection and Reporting [online] Publikované vo januári 2005. [citované 20.4.2005] URL <http://ietfreport.isoc.org/ids/draft-ietf-psamp-framework-10.txt>
- [24] Cisco Systems: NetFlow Services and Applications [online] Publikované vo júli 2002. [citované 20.4.2005] URL http://www.cisco.com/warp/public/cc/pd/iosw/ioft/neflct/tech/napps_wp.htm

- [25] Dietz, T., Information Model for Packet Sampling Exports [online] Publikované vo júli 2004. [citované 20.4.2005] URL <http://www.ietf.org/proceedings/04aug/I-D/draft-ietf-psamp-info-02.txt>

9. Zoznam príloh

1. používateľská príručka
2. systémová príručka
3. zdrojové texty
4. CD médium obsahujúce:
 - diplomovú prácu s prílohami v elektronickej podobe vo formáte PDF, RTF a DOC
 - funkčný program s dokumentáciou

10. Zoznam obrázkov a tabuliek

Zoznam obrázkov

Obr. 2.1 Všeobecná architektúra merania oneskorenia.....	7
Obr. 3.1: Architektúra meracej platformy.. ..	17
Obr. 3.2: Architektúra meracieho nástroja.. ..	20
Obr. 4.1: Architektúra PSAMP zariadenia.....	34
Obr. 5.1: Schéma vzorkovania . ..	42
Obr: 5.2: Presnosť vzorkovacích algoritmov	52
Obr: 5.3: Celková časová náročnosť algoritmov vzorkovania	55
Obr. 6.1: Algoritmus spracovania paketu.....	57
Obr. 6.2: Systematické vzorkovanie založené na počte s maximom v 400KB/s.....	63
Obr. 6.3: Systematické vzorkovanie založené na počte s maximom v 1100KB/s.....	63
Obr. 6.4: Systematické vzorkovanie založené na čase s maximom v 400KB/s.....	64
Obr. 6.5: Systematické vzorkovanie založené na počte s maximom v 1100KB/s.....	64
Obr. 6.6: Vzorkovanie n-z-N s maximom v 400KB/s.....	65
Obr. 6.7: Vzorkovanie n-z-N s maximom v 1100KB/s.....	65
Obr. 6.8: Uniformné náhodné pravdepodobnostné vzorkovanie s maximom v 400KB/s.....	66
Obr. 6.9: Uniformné náhodné pravdepodobnostné vzorkovanie s maximom v 1100KB/s.....	66
Obr. 6.10: Neuniformné náhodné pravdepodobnostné vzorkovanie založené na počte s maximom v 400KB/s.....	67
Obr. 6.11: Neuniformné náhodné pravdepodobnostné vzorkovanie založené na počte s maximom v 1100KB/s.....	67
Obr. 6.12: Tok celkovej populácie s maximom v 1100KB/s.....	68

Zoznam tabuliek

Tab. 3.1: Popis jednotlivých častí architektúry..	18
Tab. 4.1: Tabuľka popisujúca charakteristiky vzorkovania a filtrovania...	32
Tab. 5.1: Terminológia..	48
Tab. 5.2: Vzorkovacie algoritmy a ich požiadavky na implementáciu..	51
Tab. 5.3: Časová náročnosť jednotlivých algoritmov vzorkovania..	54-55
Tab. 6.1: Popis konfiguračných direktív vzorkovania.....	62