

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

**Príspevok k problematike neintruzívneho vyhodnocovania  
parametrov QoS**

Vedúci diplomovej práce:  
Ing. František Jakab

Diplomant:  
Róbert Jakab

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 02.05.2005

.....  
*vlastnoručný podpis*

Na tomto mieste bude vložené zadanie diplomovej práce

## **Pod'akovanie**

Týmto by som chcel poďakovať svojej rodine za vytvorenie optimálnych podmienok počas celého štúdia na vysokej škole.

Moja vďaka patrí aj Ing. Františkovi Jakobovi za cenné rady a pripomienky, ako aj Ľubošovi Koščovi za príjemnú spoluprácu pri vývoji programového nástroja.

Názov práce : Príspevok k problematike neintruzívneho vyhodnocovania parametrov QoS

Katedra : Katedra počítačov a informatiky, TU FEI Košice

Autor : Róbert Jakab

Vedúci DP : Ing. František Jakab

Konzultant DP : Ing. František Jakab

Dátum : 02.05.2005

Kľúčové slová : QoS, IPFIX, tok, šablóna, NetFlow, BasicMeter, šírka pásma, priepustnosť paketov

Anotácia : Práca sa zaoberá neintruzívnym meraním a vyhodnocovaním parametrov QoS. Popisuje štandard IPFIX, ktorý definuje architektúru meracej platformy ako aj protokoly podporujúce tento štandard. Na záver je popísaná koncepcia meracej platformy s využitím protokolu NetFlow9, pričom je kladený dôraz na návrh analyzujúcej aplikácie.

Thesis title : Contribution to The Problematics of Non-Intrusive Measurement of QoS

Department : Department of Computers and Informatics, TU FEI Košice

Author : Róbert Jakab

Supervisor : Ing. František Jakab

Tutor : Ing. František Jakab

Date : 02.05.2005

Keywords : QoS, IPFIX, flow, template, NetFlow, BasicMeter, bandwidth, packet rate

Annotation : The thesis deals with non-intrusive measurement and evaluation of QoS parameters. It describes standard IPFIX, which defines the architecture of measuring platform, and also describes protocols supporting this standard. At last, the concept of measuring platform is described, which uses protocol NetFlow9 with an emphasis on design of analyzing application.

# Obsah

1.	Úvod.....	1
1.1	Formulácia úlohy .....	1
1.2	Motivácia .....	2
2.	QoS (Quality of Services).....	3
2.1	Kvalita služieb v IP technológiách.....	3
2.2	Parametre QoS .....	5
2.3	Metódy merania parametrov QoS .....	5
2.4	Pasívne merania .....	6
3.	IP Flow Information Export (IPFIX) .....	7
3.1	Dôvody potreby štandardizácie a výhody tohto riešenia .....	7
3.2	Možnosti aplikácie IPFIX .....	8
3.3	Terminológia .....	10
3.4	Architektúra IPFIX .....	14
3.5	IPFIX protokol .....	17
3.6	Exportné modely .....	17
3.7	Rozlišovanie tokov .....	19
3.8	Časové značky a synchronizácia meracích bodov .....	20
3.9	Expirácia tokov .....	20
3.10	Nepovinné vlastnosti meracieho procesu.....	20
3.11	Export záznamov tokov .....	21
3.12	Požiadavky na možnosti konfigurácie .....	23
3.13	Všeobecné požiadavky kladené na implementáciu riešení .....	24
3.14	Typy zariadení v architektúre IPFIX.....	24
3.15	Bezpečnosť .....	25
4.	Protokoly vyhovujúce štandardu IPFIX.....	28
4.1	CRANE [29] .....	28
4.2	Diameter [30] .....	29
4.3	LFAP .....	30
4.4	Streaming IPDR .....	31
4.5	NetFlow9 [32] .....	31
4.6	Porovnanie vhodnosti využitia jednotlivých protokolov .....	34
5.	Existujúce implementácie protokolu NetFlow.....	34
5.1	Flow-tools [34].....	34
5.2	Stager [35].....	35
5.3	Softflowd [36], flowd [37] .....	35
5.4	nProbe, nTop [38] .....	36
5.5	Ostatné nástroje.....	36
6.	sFlow [28][42].....	37
7.	Projekt BasicMeter.....	38
7.1	Vznik a vývoj .....	38
7.2	Architektúra nástroja.....	38
7.3	BM Analyzer .....	40
7.4	Ďalší vývoj nástroja BasicMeter .....	51
8.	Experimentálne merania.....	52
9.	Zhodnotenie riešenia .....	56
10.	Zoznam použitej literatúry .....	57
11.	Zoznam príloh .....	61
12.	Zoznam obrázkov a tabuliek .....	61

# 1. Úvod

Diplomová práca sa zaoberá problematikou neintruzívnych meraní parametrov kvality služieb (Quality of Service, QoS) v počítačových sieťach s dôrazom na štandard IPFIX (IP Flow Information Export) a jeho implementáciu prostredníctvom NetFlow9.

Pojem QoS sa vzťahuje na schopnosti siete zabezpečiť lepšie služby pre určitý typ sieťovej prevádzky nad rozličnými sieťovými technológiami založenými na protokole IP. V praxi to znamená prioritné spracovanie pre túto prevádzku, ako aj kontrolované oneskorenie (latency), kolísanie oneskorenia (jitter) a zabezpečenie ďalších prevádzkových parametrov definovaných QoS.

Vzhľadom na rastúcu potrebu zabezpečenia prevádzkových parametrov, ktoré sú požadované špecifickými typmi prevádzky ako je prenos hlasu alebo videa, bol vyvinutý štandard, ktorý popisuje architektúru komplexného nástroja na sledovanie a analýzu týchto parametrov. Existuje viacero konkrétnych návrhov, ktoré sa viac či menej tomuto štandardu približujú, v tejto práci budú popísané najpodstatnejšie z nich. Najväčší dôraz bude kladený na popis protokolu NetFlow9, ktorý bol vyvinutý spoločnosťou Cisco Systems a je najhorúcejším kandidátom na splnenie podmienok definovaných štandardom IPFIX.

Bude navrhnutý a predstavený nástroj na vyhodnocovanie nameraných údajov ako súčasť architektúry IPFIX. Ďalej bude priblížený nástroj na zber nameraných dát, ktoré je možné vyhodnocovať. So spomenutými nástrojmi budú realizované aj experimentálne merania a bude možné overiť ich využiteľnosť v praxi.

## 1.1 Formulácia úlohy

Cieľom diplomovej práce je návrh a implementácia nástroja pre vyhodnocovanie nameraných resp. zozbieraných údajov na úrovni tokov. V rámci tejto úlohy je potrebné:

- Popísať problematiku neintruzívnych meraní parametrov QoS s dôrazom na štandardizáciu procesu merania.
- Analyzovať koncepciu štandardu IPFIX.
- Zistiť možnosti implementácie štandardu IPFIX s dôrazom na existujúce implementácie (NetFlow9) a experimentálne riešenia (BasicMeter).

- Navrhnuť a implementovať aplikáciu určenú na analýzu nameraných údajov, získaných prostredníctvom protokolu NetFlow9.
- Experimentálne overiť funkčnosť riešenia DP.

## 1.2 Motivácia

V súčasnej dobe sa kladú čoraz väčšie nároky na kvalitu poskytovaných služieb. Keďže postupne dochádza ku konvergencii klasických telekomunikačných a dátových sietí, je často potrebné kontrolovať parametre v sieťach, ktoré sú realizované na vybudovanej infraštruktúre. V súčasnej dobe dochádza k masovému nástupu resp. rozširovaniu nových technológií na prenos hlasu a videa. Sieť, ktorá úplne postačuje obyčajnému prenosu dát, nemusí vyhovovať prísnejším požiadavkám, ktoré si kladú napr. technológia VoIP (Voice over IP) alebo realizácia videokonferencie. Práve z tohto dôvodu vznikla požiadavka na meranie určitých prevádzkových parametrov, ktoré sú definované QoS. Na základe ich monitorovania je možné prispôbením nastavení tieto parametre dosiahnuť.

Tieto parametre sa taktiež stávajú podkladom pre zmluvy medzi poskytovateľmi služieb a zákazníkmi. Vznikajú tzv. SLA (Service Level Agreement), na základe ktorých sa poskytovateľ zaväzuje poskytovať služby s dohodnutými prevádzkovými parametrami.

Momentálne existuje viacero riešení, ktoré umožňujú či už čiastočné alebo úplné sledovanie týchto parametrov. Postupom času sa objavila potreba jednotného riešenia, ktoré by bolo akceptované aj medzi výrobcami sieťových zariadení. V prípade, že by takéto riešenie bolo v týchto zariadeniach implementované, bolo by možné získavať údaje potrebné pre analýzu prevádzkových parametrov priamo z reálnej prevádzky, čo je vlastne podstatou neintruzívnych meraní.

Analýzou meraní by bolo možné optimalizovať prevádzku tak, aby poskytované služby mali požadovanú kvalitu a mohli by tiež slúžiť ako podklad pri rozhodovaní o prípadnom rozširovaní siete, ako aj účtovaní služieb.

## 2. QoS (Quality of Services)

### 2.1 Kvalita služieb v IP technológiách

Architektúra Internetu je založená na modeli najlepšej prevádzky s minimálnym úsilím (best-effort traffic model). Tento model je dostatočne funkčný a vhodný pre tradičné sieťové aplikácie ako sú elektronická pošta a prenos súborov. Existujú však aplikácie, ktoré si pre svoju správnu a spoľahlivú činnosť vyžadujú zabezpečenie určitých parametrov prenosových liniek. Ide napríklad o aplikácie ako prenos videa na požiadanie (Video On Demand, VoD), IP telefónia, telekonferencie, vzdialené vyučovanie, digitálna televízia s vysokou rozlišovacou schopnosťou (High Definition TV, HDTV), ktoré si okrem šírky prenosového pásma vyžadujú aj garanciu časových charakteristík. Zabezpečenie poskytovania služieb s požadovanou kvalitou vyjadruje tzv. kvalitu služby, ktorá je popísaná parametrami QoS.

Definícia QoS: „Quality of Services“ – sú chápané ako komplexné opatrenia vedúce k zabezpečeniu doručenia informácie koncovému používateľovi v požadovanej kvalite.

QoS predstavuje jednu z najväčších výziev pri komunikácii v dnešnom Internete. Zámerom QoS je zabezpečiť rozdielne požiadavky jednotlivých služieb využitím spoločnej infraštruktúry, pričom ponúka možnosť zabezpečiť kvalitatívne ako aj kvantitatívne vlastnosti poskytovanej služby. Na zabezpečenie týchto vlastností sa vyžaduje stabilné a predvídateľné „správanie“ komunikačných kanálov.

**Integrované služby (Integrated Services [46])** – rozširujú základný model IP a predstavujú nové služby pre zabezpečenie parametrov pre jednotlivé toky. Poskytujú diferenciáciu na základe explicitného vyjadrenia požiadaviek na kvalitu služby pomocou špeciálneho protokolu RSVP (Resource Reservation Protocol). Vzhľadom na to, že sú požadované stavové informácie o každom toku v každom routeri, škálovateľnosť riešenia je problémom. Existujú dva základné modely:

- Služby kontrolovanej záťaže (Controlled Load Services [47]) – ponúkajú rovnakú kvalitu ako nezťažené alebo veľmi málo zaťažené siete so službou best effort. Sieťové zdroje sú rezervované pre každý dátový tok a preto jednotlivé

toky nepreťažujú sieťové uzly. Táto služba neponúka žiadnu kvantitatívnu záruku.

- Garantované služby (Guaranteed Services [9]) – ponúkajú garanciu horného limitu latencie tokov zodpovedajúcich špecifikácii určitej prevádzky. Princíp spočíva v rezervovaní šírky prenosového pásma a fronty určitej dĺžky pre každý dátový tok v každom sieťovom uzle. Služba je podobná vyhradenému spojeniu (dedicated wire).

Použitie integrovaných služieb nie je veľmi rozšírené a nepredstavuje perspektívne riešenie zabezpečenia QoS v IP sieťach.

**Diferencované služby (Differentiated Services [48])** – zabezpečujú kontrolu pre skupiny tokov. Vzhľadom nato, že toky sú rozoznávané a zoskupované len na okrajových zariadeniach, nie sú na ich smerovanie vo vnútri domény potrebné stavové informácie každého toku. Každý IP paket obdrží na okrajovom routri identifikátor DSCP (DiffServ Code Point), ktorý určuje jeho zaobchádzanie na jednotlivých uzloch (per-hop behaviour). DSCP má veľkosť 6 bitov, je preto možné špecifikovať 64 odlišných hodnôt. Štyri základné typy PHB:

- Default PHB [49] – klasické spracovanie paketu metódou best-effort delivery.
- Class Selector (CS) PHB [49] – zahŕňa prioritný model na princípe používania ToS (Type of Service).
- Assured Forwarding (AF) PHB [54] – ponúka rôzne úrovne forwardovania paketov patriacich do agregovaného toku. Sieťové zdroje sú alokované nezávisle pre každú skupinu AF. Existujú štyri triedy AF (Premium, Gold, Silver, Bronze), každá s tromi úrovňami priority zahadzovania paketov.
- Expedited Forwarding (EF) PHB [55] – garantuje pevne ohraničené hodnoty parametrov oneskorenia, kolísania oneskorenia a pomeru stratovosti vyhovujúcich IP paketov. Tento prístup si vyžaduje, aby každý router splňal tieto podmienky. Z pohľadu koncového používateľa EF emuluje vlastnosti virtuálneho okruhu.

## 2.2 Parametre QoS

Parametre QoS je možné prezentovať ako charakteristiky dátových tokov a sieťových liniek vyjadrujúce ich výkonnosť a kvalitu. Definíciou a meraním týchto parametrov sa zaoberajú organizácie ITU-T a IETF. ITU-T [50] vo svojom odporúčaní navrhuje štatistické definície QoS parametrov, zatiaľ čo IETF [51] sa zameriava na presný popis procedúr merania každého parametra. Procedúry merania týchto parametrov sa nazývajú metriky. IETF definuje nasledujúce metriky:

- **Šírka pásma (bandwidth)** je množstvo bitov prenesených za jednotku času zo zdroja do cieľa [52].
- **Stratovosť paketov (packet loss)** je množstvo nedoručených paketov, ktoré neboli prijaté v cieľi, alebo boli prijaté s chybou [56].
- **Jednosmerné oneskorenie (one-way delay)** je čas, ktorý uplynie od odoslania paketu zo zdroja až po jeho prijatie v cieľi [51]. Skladá sa z dvoch častí:
  - času potrebného na prenesenie paketu cez fyzické médium, čo je funkcia prenosovej rýchlosti linky a
  - času, ktorý predstavuje oneskorenie spôsobené radením do front, spracovaním v sieťových zariadeniach a preťažením liniek.
- **Kolísanie oneskorenia (delay variation, jitter)** je definované pre dva pakety ako rozdiel medzi jednosmerným oneskorením jedného paketu a jednosmerným oneskorením druhého paketu [53].
- **Spätočné oneskorenie (round trip time, RTT)** je čas potrebný k odoslaniu paketu zo zdroja, jeho prijatiu v cieľi, okamžitému odoslaniu naspäť k zdroju a jeho prijatiu v zdroji [57].
- **Priepustnosť paketov (packet rate)** je množstvo prenesených paketov za jednotku času.

## 2.3 Metódy merania parametrov QoS

**Aktívne (intruzívne) merania** – pri meraní charakteristík využívajú dodatočnú prevádzku, vygenerovanú špeciálne pre meracie účely. Ide o kontrolovateľné merania, ktoré môžu byť uskutočnené v ľubovoľnom čase pre ľubovoľný typ prevádzky.

**Pasívne (neintruzívne) merania** – pri meraní sa negeneruje žiadna dodatočná prevádzka, na tieto účely sa využíva výhradne existujúca prevádzka. Tento druh meraní predstavuje určité výhody oproti aktívnym. Odpadá problém emulácie prevádzky s určitými charakteristickými vlastnosťami reálnej, vzhľadom na neexistenciu umelej prevádzky nemôže dôjsť k ovplyvneniu výsledkov merania. Na druhej strane, medzi nevýhody tohto prístupu patrí fakt, že ide o neriaditeľné merania. Ďalšou nevýhodou je nutnosť prenášania riadiacich dát inou cestou, aby ani tieto neovplyvňovali skutočný tok dát. Pri meraní časových charakteristík je navyše potrebné zabezpečiť synchronizáciu času jednotlivých meracích bodov.

**Semi-aktívne merania** – na účely merania používajú existujúcu (reálnu) prevádzku, pridávajú k nej však doplňujúce údaje, resp. ju za účelom merania nejakým spôsobom modifikujú. Takéto informácie môže predstavovať napríklad časová značka paketu alebo identifikátor paketu.

## 2.4 Pasívne merania

### 2.4.1 Merania objemových charakteristík

Pre merania objemových charakteristík postačuje použitie jedného meracieho bodu. Medzi tieto charakteristiky radíme:

- šírku prenosového pásma (bandwidth),
- priepustnosť paketov (packet rate).

Pre výpočet objemových charakteristík platí vzťah **rate = count / time**, pričom význam jednotlivých parametrov je nasledovný:

- **rate** – požadovaný parameter kvality,
- **count** – v závislosti od typu merania je to buď počet bitov resp. bajtov pre meranie šírky pásma, alebo počet paketov pre meranie priepustnosti paketov,
- **time** – časová jednotka, dĺžka trvania merania.

### 2.4.2 Merania časových charakteristík

Pre merania časových charakteristík je s výnimkou merania spätného oneskorenia nutné použiť dva meracie body. K časovým charakteristikám patria:

- jednosmerné oneskorenie (one-way delay),

- spätné oneskorenie (round trip time),
- kolísanie oneskorenia (delay variation, jitter).

Dva základné problémy, ktoré môžu nastať pri týchto meraniach, sú synchronizácia času na jednotlivých meracích bodoch a generovanie jednoznačných identifikátorov paketov. Čo sa týka synchronizácie času, je možné použiť nasledujúce riešenia:

- sieťový časový protokol (Network Time Protocol, NTP) [58],
- globálny pozičný systém (Global Positioning System, GPS),
- hodinové rádiové signály (DCF77).

Pre generovanie identifikátorov paketov sa musí zabezpečiť ich jedinečnosť počas určitého časového intervalu, ktorý zahŕňa prenos paketu sieťou a jeho následné spracovanie. Ak paket nedorazí na požadované miesto v tomto definovanom čase, môžeme ho považovať za stratený. Na zabezpečenie jedinečnosti je vhodné zvoliť pre výpočet položky z hlavičky paketu, ako aj časť dátovej časti paketu. Na vygenerovanie takéhoto identifikátora sa potom môže použiť napríklad hašovacia funkcia MD5 so spomínanými vstupnými údajmi. Funkcia nesmie byť veľmi náročná, aby merací proces stíhal generovať tieto identifikátory v reálnom čase.

V nasledujúcej kapitole bude predstavený štandard popisujúci meraciu platformu využiteľnú okrem iného aj na meranie parametrov QoS.

### **3. IP Flow Information Export (IPFIX)**

#### **3.1 Dôvody potreby štandardizácie a výhody tohto riešenia**

Vzhľadom na potreby zabezpečenia parametrov QoS v sieťach postupne vznikali nástroje, ktoré umožňovali sledovanie a vyhodnocovanie prevádzky. Takýchto riešení existuje niekoľko. Nie všetky boli pôvodne vyvinuté pre potreby QoS, ale poskytovali aspoň čiastočné možnosti využitia aj v tejto oblasti. Veľkou nevýhodou týchto riešení je ich rozmanitosť, jednoúčelovosť, neškálovateľnosť a hlavne nejednotnosť. Predovšetkým potreba jednotnosti a univerzálnosti riešenia mala za následok vznik štandardu, ktorý mal popísať architektúru meracieho nástroja vhodného na sledovanie parametrov QoS.

Štandard je koncipovaný všeobecne, čo znamená, že popisuje požiadavky na funkčnosť riešenia, nie však implementačné detaily. Za základ pre vytvorenie štandardu bol zobrať protokol NetFlow verzie 9, ktorý bol vyvinutý spoločnosťou Cisco Systems, boli však špecifikované dodatočné požiadavky, ktoré riešenie musí spĺňať, aby mohlo byť označené ako vyhovujúce IPFIX. Paralelne s vývojom NetFlow9 boli vyvíjané nové, resp. vylepšované už existujúce riešenia, ktoré sa tiež usilovali o získanie toho štatútu. Ak doteraz správcovia sietí súperili s problémom možnosti analýzy dát exportovaných zariadeniami v proprietárnych formátoch, vznik tohto štandardu a jeho implementácia výrobcami sieťových prvkov umožní jednoduchú analýzu prevádzky aplikáciami, ktoré budú tento štandard podporovať. Ako bude ukázané neskôr, exportný formát je modifikovateľný, čo poskytuje možnosť exportu a následnej analýzy požadovaných údajov bez potreby modifikácie či už exportného zariadenia, zberacieho zariadenia alebo vyhodnocovacej aplikácie.

Export informácií o prevádzke z routra a zobrazovanie štatistík na báze tokov poskytuje sieťovým správcom informácie, ktoré môžu tiež pomôcť pri kľúčových rozhodnutiach ako je optimalizácia a prípadné rozširovanie siete.

## **3.2 Možnosti aplikácie IPFIX**

### **3.2.1 Účtovanie podľa typu použitia (Usage based accounting)**

V súčasnosti sa vyvíja niekoľko modelov spoplatňovania IP služieb. Okrem pripojenia typu flat rate, ktoré nepotrebuje špeciálny prístup k účtovaniu, účtovanie môže byť založené na čase prístupu k službe, ako aj na množstve prenesených údajov. Ďalším prístupom je kategorizácia na základe typu doručeného obsahu.

### **3.2.2 Profilovanie prevádzky (Traffic profiling)**

Ide o proces charakterizovania IP tokov kľúčovými parametrami, takými ako je napríklad dĺžka trvania toku, objem prenesených údajov, čas vzniku toku. Je neodmysliteľnou súčasťou návrhu a dimenzovania sietí. Typické informácie potrebné pre profilovanie prevádzky sú distribúcia služieb a protokolov v sieti, množstvo paketov určitého druhu (napr. podiel IPv6 paketov) a špecifické profily tokov.

Nakoľko účel profilovania môže byť odlišný, merania vyžadujú vysoko flexibilnú meraciu infraštruktúru najmä s možnosťami konfigurácie meraní a klasifikácie prevádzky.

### **3.2.3 Traffic engineering**

Zahŕňa metódy pre meranie, modelovanie, kontrolu a riadenie sietí. Cieľom tohto procesu je optimalizácia využitia zdrojov a výkonu, resp. priepustnosti sietí. Typické parametre, požadované pre tento druh činnosti, sú využitie prenosových liniek, záťaž liniek medzi jednotlivými uzlami siete, počet, veľkosť a vstupné/výstupné uzly jednotlivých tokov, ako aj smerovacie informácie. Údaje získané prostredníctvom IPFIX poskytujú detailné informácie o prevádzke a môžu byť použité pri optimalizácii smerovacej politiky siete s využitím techník ako load balancing, alebo smerovanie určitého druhu prevádzky určitými trasami v sieti, ako aj prioritizácia prevádzky.

### **3.2.4 Detekcia útoku/prieniku (Attack/intrusion detection)**

Zber informácií tokov môže hrať dôležitú úlohu pre sieťovú bezpečnosť, či už pri detekcii porušenia bezpečnosti alebo pri následnej ochrane. Monitorovanie tokov umožní detekciu nezvyčajných situácií a podozrivých tokov, ktoré nie sú v súlade s bezpečnostnou politikou siete. V ďalšom kroku môžu informácie o atakujúcom toku pomôcť pri voľbe obrannej stratégie.

Na druhej strane, detekcia prienikov je náročnejšia úloha, ktorá okrem sledovania špecifických charakteristík tokov vyžaduje aj sledovanie stavových informácií jednotlivých paketov patriacich do tokov. Podozrivé aktivity, alebo príliš frekventované aktivity určitého druhu môžu byť charakterizované špecifickými komunikačnými modelmi, ktoré sú detekovateľné na základe sekvencií určitých typov paketov.

### **3.2.5 Plánovanie sietí**

Údaje IPFIX, zozbierané počas dlhého časového intervalu, môžu byť použité na predvídanie rastu siete a plánovanie rozširovania siete prostredníctvom zvyšovania počtu aktívnych sieťových prvkov alebo sieťových rozhraní týchto prvkov. Údaje získané prostredníctvom IPFIX pomôžu optimalizovať strategické plánovanie siete (plánovanie upgradovania backbonu siete, plánovanie politiky smerovania), ako aj

taktické rozhodnutia z oblasti sieťového inžinierstva (upgrade routra alebo zvýšenie kapacity prenosovej linky). Výsledkom je minimalizácia nákladov a maximalizácia priepustnosti a spoľahlivosti siete.

### **3.2.6 Ukladanie a dolovanie dát (data warehousing and mining)**

Zozbierané údaje môžu byť uložené pre neskoršiu analýzu. Typickým príkladom využitia je poskytovateľ komunikačných služieb, ktorý tieto údaje môže využiť na odhalenie najpoužívanejších aplikácií a služieb určitou skupinou zákazníkov a optimalizácia spracovania daného typu prevádzky. Nakoľko IPFIX poskytuje aj potrebné časové informácie, túto optimalizáciu je možné zabezpečovať aj dynamicky pre rôzny typ prevádzky v rôznych časových intervaloch.

### **3.2.7 Monitorovanie QoS**

IPFIX sa v neposlednom rade dá využiť aj na monitorovanie parametrov QoS. Na druhej strane si treba uvedomiť, že požiadavky špecifikované štandardom IPFIX nepokrývajú monitorovanie všetkých parametrov QoS. Nakoľko informácie sú vyhodnocované na úrovni tokov a nie paketov, ktoré síce definujú charakter paketov patriacich do toku, nezahŕňajú však informácie o samotných paketoch. Typickým príkladom je meranie kolísania oneskorenia (jitter), kde sa vyžaduje pridelenie časových značiek (timestamps) pre jednotlivé pakety. Naproti tomu požiadavky IPFIX definujú časové značky len pre prvý a posledný paket patriaci do toku (z ktorých sa dá určiť kedy tok začal a ako dlho tok trval). Na druhej strane je možné za účelom takéhoto merania nakonfigurovať exportný proces tak, aby každý tok obsahoval práve jeden paket. Tým sa však stratí výhoda kategorizácie prevádzky do tokov a rapídne stúpne množstvo exportovaných údajov. Podobné riešenia by mali byť použité len za účelom vykonania špecifických meraní. Aj keď IPFIX nie je primárne určený na tieto účely, možnosť využitia aj v tejto oblasti dokazuje veľkú flexibilitu návrhu.

## **3.3 Terminológia**

**IP tok prevádzky (IP traffic flow)** – Skupina IP paketov prechádzajúcich cez pozorovací bod v sieti počas nejakého časového intervalu. Všetky pakety patriace

danému toku majú množinu spoločných vlastností. Každá vlastnosť je definovaná ako výsledok aplikovania funkcie na hodnoty:

- jedného alebo viacerých polí hlavičiek paketu (napr. cieľová IP adresa)
- jednej alebo viacerých vlastností samotného paketu (napr. dĺžka paketu)
- jedného alebo viacerých polí odvodených od spracovania paketu

Paket patrí do toku ak spĺňa všetky uvedené kritériá. Táto definícia pokrýva rozsah od toku obsahujúceho všetky pakety, ktoré prešli sieťovým rozhraním, až po tok obsahujúci jediný paket. Toky môžeme definovať rôzne:

- Na odlišenie rôznych tokov definujeme rôzne polia, ktorých rôzne kombinácie potom znamenajú rôzne toky. Ak definujeme kľúčové polia {zdrojová IP adresa, cieľová IP adresa, ToS (Type Of Service)}, potom každý z paketov:
  - {192.1.40.1, 171.6.23.5, 4}
  - {192.1.40.23, 171.6.23.67, 4}
  - {192.1.40.23, 171.6.23.67, 2}
  - {198.20.9.200, 171.6.23.67, 4}
- patrí do iného toku.
- Pakety budú patriť do jedného toku, ak budú prechádzať určitým pozorovacím bodom. Toto môžeme docieľiť definovaním kľúčových polí {zdrojová IP adresa, cieľová IP adresa, ToS}, ako v predošlom prípade, a aplikovaním funkcie, ktorá uplatní masku na spodných 8 bitov zdrojovej a cieľovej IP adresy (výsledok je adresa v tvare /24). Štyri toky z predošlého prípadu sa zlúčia do troch, pričom pakety 1 a 2 sa zaradia do toho istého toku:
  - {192.1.40.0/24, 171.6.23.0/24, 4}
  - {192.1.40.0/24, 171.6.23.0/24, 2}
  - {198.20.9.0/24, 171.6.23.0/24, 4}
- Ďalším spôsobom je filtrácia niektorých polí všetkých paketov, ktoré prejdú pozorovacím bodom s cieľom zvoliť len určité toky. Filter je definovaný zvolením konštantných hodnôt niektorých polí paketu. Napr. sa vezmú do úvahy všetky pakety, ktoré idú zo siete 192.1.40.0/24 do siete 171.6.23.0/24 s hodnotou ToS rovnou 4. Ostatné sa neberú do úvahy. V tomto prípade sa 3 toky z predošlého prípadu zredukujú na 1 tok, pričom sa odfiltrujú pakety 3. a 4. toku. Uvedený prípad je príkladom toho, ako funkcia F so vstupom {zdrojová IP

adresa, cieľová IP adresa, ToS} označí len pakety, ktoré spĺňajú nasledovné 3 podmienky:

- uplatní masku na spodných 8 bitov zdrojovej IP adresy a porovná s 192.1.40.0
  - uplatní masku na spodných 8 bitov cieľovej IP adresy a porovná s 171.6.23.0
  - hodnota ToS je rovná 4
- V závislosti od hodnôt uvedených polí {zdrojová IP adresa, cieľová IP adresa, ToS} rôznych paketov porovnávací mechanizmus funkcie F buď zvolí, odfiltruje, alebo zlúči pozorované pakety, čím sa vytvoria rôzne toky. Rôzne kombinácie hodnôt F(zdrojová IP adresa, cieľová IP adresa, ToS) vedú k definícii jedného alebo viacerých tokov.

**Pozorovací bod (Observation point)** – Miesto v sieti, kde môžu byť IP pakety pozorované. Môže sa jednať o linku, na ktorej je pripojená sonda, zdieľané médium (napr. sieť typu Ethernet), port alebo skupina portov routra. Pozorovací bod môže byť aj skupina niekoľkých ďalších pozorovacích bodov (napr. celý zásuvný modul routra obsahujúci niekoľko sieťových rozhraní, môže byť chápaný ako jeden pozorovací bod).

**Pozorovacia doména (Observation domain)** – Najväčšia možná skupina pozorovacích bodov, pre ktorú môžu byť informácie o tokoch agregované meracím procesom. Pozorovacia doména sa prezentuje zberaciemu procesu jedinečným ID kvôli identifikácii správ, ktoré vygeneruje. Typickým príkladom je zásuvný modul routra, ktorý pozostáva z viacerých sieťových rozhraní, pričom každé rozhranie môže byť pozorovacím bodom. Každý pozorovací bod je pridružený k nejakej pozorovacej doméne.

**Merací proces (Metering process)** – Merací proces generuje záznamy tokov. Vstupom sú hlavičky paketov získané v pozorovacom bode a jeho spracovanie (napr. voľba výstupného rozhrania). Merací proces pozostáva z funkcií, ktoré zabezpečujú spracovanie hlavičiek paketov, pridelovanie časových značiek, vzorkovanie, triedenie, ako aj udržiavanie záznamov jednotlivých tokov.

Udržiavanie záznamov môže zahŕňať vytváranie nových záznamov, aktualizáciu existujúcich, výpočet štatistík jednotlivých tokov, odvádzanie ďalších vlastností, detekciu expirácie tokov, predávanie záznamov exportnému procesu a rušenie

záznamov. Vzorkovacie a klasifikačné funkcie môžu byť aplikované viackrát s rôznymi parametrami.

**Kľúč toku** – Mohol by sa tiež označiť kľúčový atribút toku. Je to každý údaj z hlavičky paketu, alebo nejaký z nej odvoodený, použitý na identifikáciu toku.

**Záznam toku (Flow record)** – Obsahuje informácie o danom toku, ktorý bol meraný v pozorovacom bode. Obsahuje namerané veličiny (napr. počet všetkých bajtov obsiahnutých vo všetkých paketoch patriacich do daného toku) a zvyčajne tiež charakteristické vlastnosti daného toku (napr. zdrojová IP adresa).

**Exportný proces (Export process)** – Jeho úlohou je zasielanie záznamov tokov jednému alebo viacerým zberacím procesom. Záznamy sa získavajú z jedného alebo viacerých meracích procesov.

**Exportér** – Zariadenie, na ktorom beží jeden alebo viac exportných procesov.

**IPFIX zariadenie** – Zariadenie, ktoré reprezentuje aspoň jeden pozorovací bod, merací proces a exportný proces.

**Zberací proces (Collecting process)** – Prijíma a sústreďuje údaje od jedného alebo viacerých exportných procesov, ktoré následne ukladá alebo ďalej spracováva.

**Kolektor** – Zariadenie, na ktorom beží jeden alebo viac zberacích procesov.

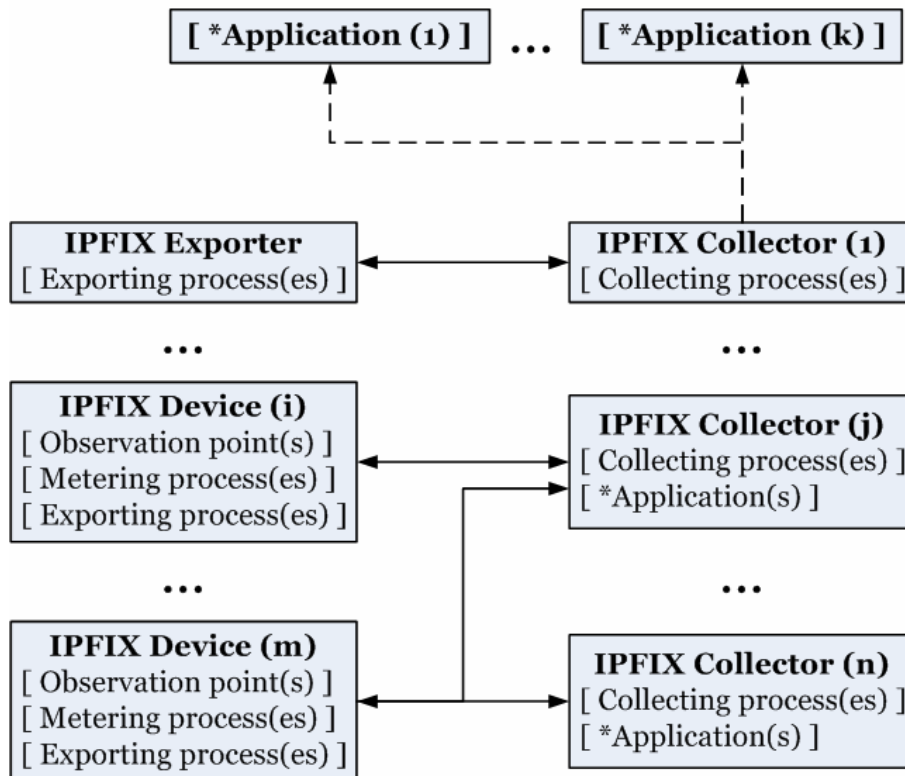
**Šablóna (Template)** – Ide o usporiadanú dvojicu <typ,dĺžka>, ktorá slúži na presnú identifikáciu štruktúry a sémantiky čiastkovej informácie, ktorú je potrebné preniesť z IPFIX zariadenia do kolektora. Každá šablóna je jedinečne identifikovateľná prostredníctvom ID šablóny.

**Riadiace informácie (Control information)** – zahŕňajú definíciu toku, výberové kritériá paketov vrámci toku, poslané exportným procesom, a šablóny popisujúce exportované údaje. Vo všeobecnosti tieto informácie zahŕňajú všetky informácie určené koncovým bodom potrebné pre „pochopenie“ protokolu IPFIX a špeciálne pre adresáta (kolektor), aby dokázal interpretovať dáta poslané odosielateľom (exportérom).

**Dátový tok** – Zahŕňa dátové záznamy vzťahujúce sa k informáciám o rôznych pozorovaných tokoch na každom z pozorovacích bodov.

## 3.4 Architektúra IPFIX

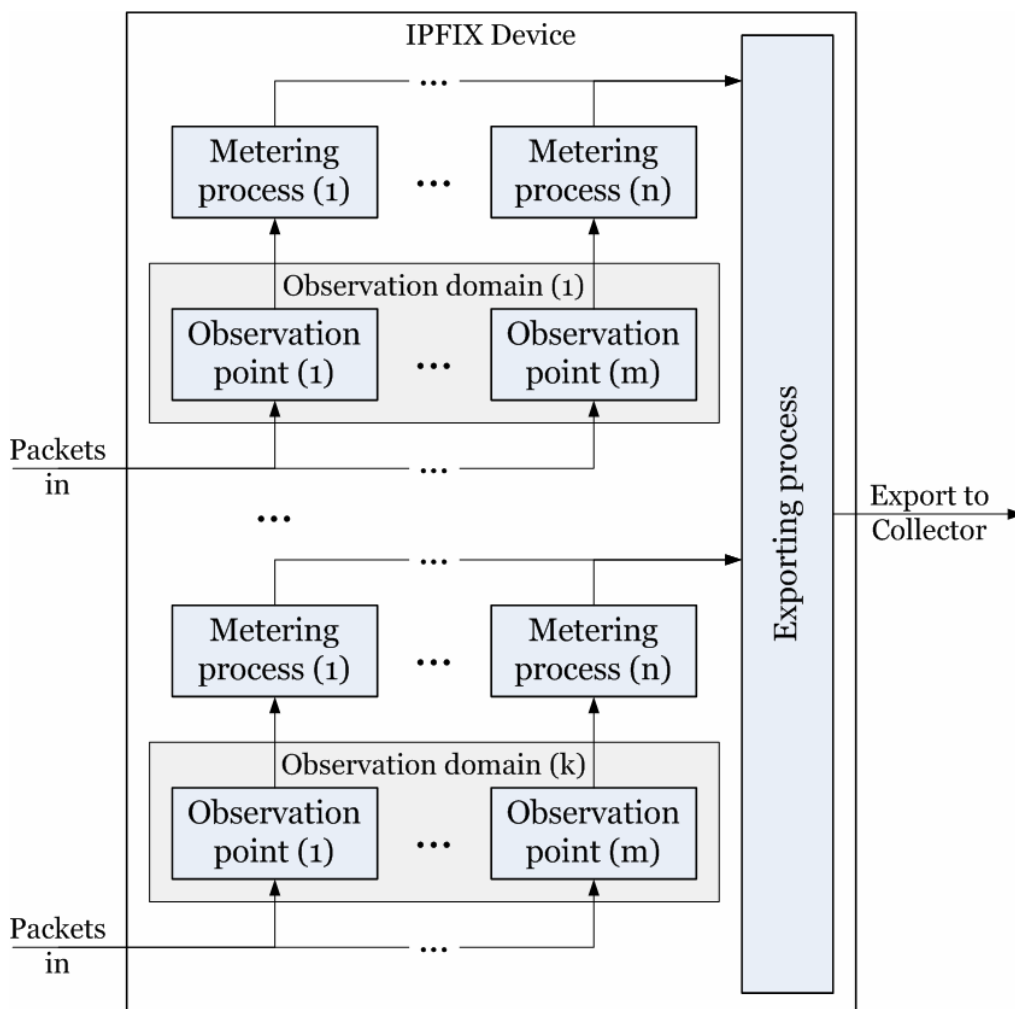
### 3.4.1 Referenčný model



Obr. 3.1: Model architektúry IPFIX.

Na Obr. 3.1 je zobrazený model architektúry IPFIX. Jednotlivé funkčné komponenty sú zobrazené v hranatých zátvorkách, komponenty označené hviezdičkou nie sú súčasťou architektúry IPFIX. Podobne rozhrania označené plnou čiarou sú definované architektúrou IPFIX, tie označené čiarkovane nie sú.

Obr. 3.2 zobrazuje blokovú schému IPFIX zariadenia, kde sú v jednotlivých blokoch zobrazené komponenty architektúry IPFIX.



Obr. 3.2: Schéma IPFIX zariadenia.

### 3.4.2 Merací proces

Každý pozorovací bod, ktorý sa podieľa na meraní tokov, musí byť priradený aspoň k jednému meraciemu procesu. Merací proces je funkčný blok, ktorý spravuje všetky toky vygenerované pozorovacou doménou. Jeho typické funkcie sú:

- udržiavať databázu všetkých záznamov tokov z pozorovacej domény, čo zahŕňa vytváranie nových záznamov, aktualizovanie (udržiavanie) existujúcich, výpočet štatistík týkajúcich sa jednotlivých tokov a generovanie ďalších charakteristík na základe informácií o tokoch, a pridávanie ďalších informácií, ktoré sa priamo nedotýkajú tokov;
- udržiavanie štatistických údajov o samotnom meracom procese, napr. počet vygenerovaných záznamov, počet paketov zahrnutých do meraní, atď.

Merací proces musí byť spoľahlivý v takom zmysle, že musí byť schopný spracovať každý paket v závislosti na konfigurácii (s ohľadom na vzorkovanie resp. filtrovanie). V prípade preťaženia, alebo ak z nejakého iného dôvodu všetky pakety nemôžu byť zaradené do meracieho procesu, merací proces musí tento stav zistiť a nejakým spôsobom oznámiť. Možné reakcie na tento stav sú nasledovné:

- Zredukovať počet meraných tokov;
- Vzorkovať pakety ešte pred tým, než sú spracované meracím procesom (ak už je vzorkovanie zapnuté, znížiť vzorkovaciu frekvenciu);
- Zastaviť meranie;
- Znížiť využitie dostupných prostriedkov ostatných procesov na danom zariadení (napr. zníženie priepustnosti posielania paketov).

V prípade preťaženia sa pripúšťajú aj iné možnosti ošetrenia takéhoto stavu, v každom prípade však musia byť dodržané kritériá korektného ukončenia tokov a ich záznamy musia byť vyexportované separátne od tokov vytvorených po zmene správania (napr. po zmene klasifikácie paketov). Ďalšou možnosťou po odhalení takejto situácie je samozrejme zásah administrátora.

### **3.4.3 Exportný proces**

Exportný proces je funkčný blok, ktorý pôsobí medzi pozorovacou doménou na jednej strane a kolektorom na strane druhej. Jeho úlohou je zabezpečiť posielanie dát spracovaných meracím procesom zberaciemu procesu. Export môže zahŕňať dodatočné filtračné alebo vzorkovacie kritériá, ktoré kategorizujú typ prevádzky ktorá má/nemá byť exportovaná. Tieto kritériá budú spomenuté neskôr.

### **3.4.4 Zberací proces**

Získava údaje od jedného alebo viacerých exportných procesov. Na správne interpretovanie týchto údajov používa ID šablóny. Exportér by mal zaručiť doručenie popisu šablóny spolu s jej ID, a to periodickým zasielaním v určitom časovom intervale. Ďalším dôvodom na takéto periodické zasielanie je obmedzená časová platnosť šablóny. Ak kolektor nemá popis šablóny, mal by si záznamy tokov, ktoré nevie interpretovať, uchovať a po prijatí príslušnej šablóny by ich mal následne spracovať. Takto sa zabráni strate informácií o tokoch. Na druhej strane, interval exportu popisu šablón musí byť

zvolený optimálne, pretože pri príliš malej periodicite môže dôjsť k zahlteniu kolektora záznamami tokov, ktoré nie je schopný interpretovať.

### **3.5 IPFIX protokol**

IPFIX zariadenie pozostáva zo skupiny navzájom kooperujúcich procesov, ktoré implementujú funkčné bloky popísané v predošlej časti. Z iného pohľadu, IPFIX zariadenie predstavuje sieťový prvok, ktorý implementuje IPFIX protokol. Takéto zariadenie vykonáva nasledujúce funkcie:

- transformuje riadiace informácie do šablón,
- zahŕňa pakety zachytené pozorovacím bodom do záznamov tokov,
- konvertuje šablóny a záznamy tokov na správy protokolu IPFIX,
- posiela tieto správy kolektoru.

Použitím mechanizmu šablón nové atribúty popisujúce prevádzku môžu byť pridané do záznamov tokov bez nutnosti zmeny štruktúry exportovaných dát. Ďalšou výhodou je, že aj keď kolektor nepozná sémantiku niektorých atribútov v šablóne, aj napriek tomu dokáže záznam toku interpretovať. Aj keď bude tieto neznáme atribúty ignorovať, stále môže dekódovať niektoré relevantné informácie o danom toku. V neposlednom rade je tu možnosť konfigurácie šablón, čo sa umožňuje exportovať len potrebné údaje, čím dochádza k šetreniu prenosových kapacít siete na jednej strane, ako aj hardvérových prostriedkov (napr. pamäť, výpočtová kapacita) exportného a zberacieho procesu na strane druhej.

### **3.6 Exportné modely**

#### **3.6.1 Spôsoby exportu údajov: push a pull**

Vo všeobecnosti existujú dva spôsoby exportu údajov: push a pull. Pri prvom spôsobe exportný proces po odštartovaní autonómne posiela záznamy tokov podľa nastavených kritérií bez potreby nejakého impulzu zvonku. Toto je základný spôsob, ktorý definuje štandard IPFIX. Naproti tomu pri pull spôsobe sú záznamy exportované na vyžiadanie – exportný proces teda čaká na externý pokyn zvyčajne od zberacieho procesu, na základe ktorého dôjde k exportu nazhromaždených záznamov.

Pri push spôsobe exportný proces musí byť schopný exportovať záznamy v pravidelných nastaviteľných intervaloch.

Merací proces môže počas komunikácie posilať upozornenia zberaciemu procesu, v prípade, že nastane nejaká špecifická udalosť (napr. vytvorenie nového toku, čo zodpovedá príchodu prvého paketu daného toku, alebo expirácia toku po vypršaní časového limitu).

### 3.6.2 Export so spoľahlivým riadením spojenia

Ak sieť, v ktorej sa nachádza IPFIX zariadenie a kolektor negarantuje spoľahlivosť prenosu, aspoň riadiace informácie majú byť exportované použitím spoľahlivého protokolu. Na samotný prenos dát potom môže byť využitý ako spoľahlivý tak aj nespoľahlivý protokol. Možné protokoly použiteľné pre prenos:

- SCTP [23] – podporuje spoľahlivý aj nespoľahlivý prenos,
- TCP – poskytuje len spoľahlivý prenos,
- UDP – podporuje len nespoľahlivý prenos.

Vytvorenie spojenia (v prípade spojovo-orientovaného protokolu) je čisto v réžii daného protokolu, IPFIX protokol neposkytuje žiadne mechanizmy konfigurácie exportného alebo zberacieho procesu. Po pripojení strana kolektor prijíma riadiace informácie a používa ich k interpretovaniu záznamov tokov. IPFIX zariadenie by malo nastaviť hodnotu *čas vypršania funkčnosti* (keepalive timeout v prípade protokolu TCP, heartbeat interval v prípade SCTP, alebo keepalive na úrovni protokolu IPFIX - ak existuje) na primerane malú hodnotu, aby mohlo čím skôr detekovať poruchu kolektora.

Porucha kolektora väčšinou spôsobí reštartovanie zberacieho procesu, alebo celého kolektora. Pri zistení takejto poruchy by malo IPFIX zariadenie okamžite prestať posilať údaje kolektoru a malo by sa pokúsiť znovu nadviazať spojenie. Tento postup platí v prípade, že exportér posila údaje jednému kolektoru.

Pri existencii viacerých kolektorov sú ďalšie považované za takzvané sekundárne a údaje sa k nim začnú exportovať len vtedy, ak dôjde k zlyhaniu spojenia s primárnym kolektorom. Exportér by v prípade viacerých kolektorov mal udržiavať ich zoznam, aby v takejto situácii mohol rýchle zareagovať a minimalizovať tak dobu prerušenia exportu dát. V závislosti od konfigurácie sa buď nový kolektor stane primárnym kolektorom, alebo sa exportér pokúša naďalej nadviazať spojenie s

primárnym kolektorom, avšak už počas exportu údajov sekundárnemu kolektorovi. Táto druhá možnosť prichádza do úvahy v prípade, že primárny kolektor poskytuje väčší výkon alebo kapacitu ako sekundárne.

### 3.7 Rozlišovanie tokov

Pakety sú priradzované do tokov na základe ich charakteristických vlastností, ktoré sú definované v hlavičke paketov. Pri vyhodnocovaní tohto procesu paket, ktorý sa odlišuje čo i len jedným parametrom, sa považuje za paket iného toku. Štandard definuje skupinu vlastností, ktoré by merací proces mal byť schopný rozlíšiť. Sú definované tri úrovne, a to: povinné, odporúčané a voliteľné. Pri rozlišovaní tokov sa nemusia nutne použiť všetky kritériá, práve naopak – niekedy je pre určitý účel vhodné použitie len niektorých zvolených vlastností. To, na základe ktorých vlastností sa budú pakety zaradzovať do tokov, závisí na konfigurácii meracieho bodu, čo predstavuje jednu zo základných výhod IPFIX.

V niektorých špeciálnych prípadoch (šifrovanie) sa môže stať, že merací proces nie je schopný dekodovať niektoré údaje z hlavičky paketu, vtedy je samozrejme možné kategorizovať pakety len na základe údajov, ktoré sú k dispozícii. Každý merací proces musí byť schopný separovať toky na základe:

- vstupného alebo výstupného rozhrania, resp. na základe oboch;
- nasledujúcich políček hlavičky IP paketu:
  - zdrojová IP adresa,
  - cieľová IP adresa,
  - typ protokolu (TCP, UDP, ICMP),
  - verzia IP protokolu, ak je pozorovací bod umiestnený na zariadení, ktoré podporuje viac verzií IP protokolu;
- čísla zdrojového a cieľového portu hlavičky transportnej vrstvy v prípade, že je ako transportný protokol použitý TCP, UDP alebo SCTP [23];
- MPLS návestia (label), ak je pozorovací bod umiestnený na zariadení podporujúcom MPLS (Multi Protocol Label Switching) [59];
- DSCP (Differentiated Services Code Point), ak zariadenie, na ktorom je umiestnený pozorovací bod podporuje Differentiated Services [49].

### 3.8 Časové značky a synchronizácia meracích bodov

Merací proces musí byť schopný vygenerovať časové značky pre prvý a posledný paket patriaci do toku so schopnosťou rozlíšenia minimálne 0,01 sekundy. Rovnako dôležitá je synchronizácia meracích bodov s UTC, z čoho vyplýva aj možnosť synchronizácie časových značiek rôznych meracích procesov. Nakoľko časové značky môžu byť modifikované na strane zberacieho procesu, nie je nevyhnutné, aby exportný proces generoval časové značky priamo vo formáte UTC. Tieto môžu byť exportované v lokálnom čase a zberací proces môže prerátať túto informáciu podľa rozdielu lokálneho času a UTC. V každom prípade je však nevyhnutná synchronizácia meracích bodov, inak by časové značky stratili svoj význam.

### 3.9 Expirácia tokov

Tok sa považuje za expirovaný, ak počas nastaveného časového intervalu nebol zaznamenaný žiadny paket patriaci do daného toku. Sú spôsoby ako detekovať vypršanie toku aj pred uplynutím časového intervalu (napr. overovaním FIN alebo RST bitu v TCP spojení). Merací proces musí byť schopný expiráciu toku zistiť a záznam takého toku musí byť následne exportným procesom vyexportovaný.

### 3.10 Nepovinné vlastnosti meracieho procesu

Pre multicastové toky obsahujúce pakety replikované na viacero výstupných rozhraní, merací proces by mal udržiavať rôzne záznamy pre každé výstupné rozhranie. Ak napríklad príde na vstupný port routra multicastový paket, ktorý bude presmerovaný na štyri výstupné porty, merací proces by mal registrovať štyri rôzne záznamy tokov, ktoré sa budú líšiť vo výstupných rozhraniach.

V prípade fragmentácie paketov, jedine prvý fragment bude obsahovať údaje potrebné na klasifikáciu paketu a jeho zaradenie do toku. V takomto prípade je síce zaručené, že pôvodca fragmentácie odošle tento prvý fragment ako prvý, nie je však zaručené, že tento bude zachytený meracím procesom ako prvý. Merací proces preto môže uchovávať informácie o stave fragmentácie, aby bol schopný korektne zaradiť do tokov aj fragmenty neobsahujúce v hlavičke dostatočné informácie.

Ďalším prípadom je správna detekcia a následné ignorovanie paketov vygenerovaných funkciou port copy na zariadení kde sa nachádza pozorovací bod.

## 3.11 Export záznamov tokov

### 3.11.1 Informačný model

Zahŕňa zoznam atribútov, ktoré musí byť schopný exportovací proces exportovať v takom zmysle, že musí existovať možnosť konfigurácie zariadenia aby exportovalo ľubovoľný z týchto atribútov:

- verzia IP protokolu (len v prípade ak na danom pozorovacom bode je podporovaných viacero verzií)
- zdrojová IP adresa
- cieľová IP adresa
- typ protokolu (TCP, UDP, ICMP, ...)
- zdrojový a cieľový port (v prípade ak je použitý TCP alebo UDP protokol)
- počet prenesených paketov (ak je paket fragmentovaný, započítava sa každý fragment)
- počet prenesených bajtov (zahŕňa celý paket vrátane hlavičky)
- ToS (Type Of Service) – v prípade IPv4, alebo TC (Traffic Class) pri IPv6
- návestie toku (flow label) pri IPv6 protokole
- návestie MPLS (ak je MPLS na pozorovacom bode podporované)
- časová značka prvého paketu patriaceho do toku
- časová značka posledného paketu patriaceho do toku
- informácia o použitom vzorkovaní (ak je nejaké použité)
- jedinečný identifikátor pozorovacieho bodu
- jedinečný identifikátor exportného procesu

Odporúčané atribúty, ktoré by exportný proces mal vedieť exportovať:

- typ a kód ICMP, ak sa jedná o ICMP správu
- číslo (index) vstupného/výstupného rozhrania – neplatí v prípade, že pozorovacím bodom je sonda
- replikačný faktor – počet výstupných paketov ktoré vzniknú zo vstupného paketu (jedná sa o dynamickú charakteristiku multicastových tokov, v prípade unicastového toku má hodnotu 1)

Voliteľné atribúty, ktoré exportný proces môže byť schopný exportovať:

- TTL (Time To Live) v prípade IPv4, HL (Hop Limit) v prípade IPv6

- príznaky (flags) hlavičky IP paketu
- príznaky hlavičky TCP segmentu
- počet zahodených paketov (v prípade fragmentácie sa počíta každý fragment)
- počet fragmentovaných paketov (počet všetkých paketov, ktoré majú nastavený príznak fragmentácie)
- IP adresa ďalšieho uzla resp. stanice (next hop)
- číslo zdrojového autonómneho systému BGP
- číslo cieľového autonómneho systému BGP
- číslo ďalšieho nasledujúceho (next hop) autonómneho systému BGP

Zoznam voliteľných atribútov nemusí byť konečný, nakoniec, jedná sa o výhodu ak je merací proces schopný exportovať čo najviac údajov.

### 3.11.2 Dátový model

Popisuje ako sú dáta reprezentované v záznamoch tokov. Tento formát musí byť rozširovateľný aby sa v budúcnosti mohli pridávať nové atribúty, a konfigurovateľný, aby sa exportovali len atribúty potrebné pre dané meranie, resp. sledovanie prevádzky. Musí existovať možnosť popísať poradie a typ atribútov, ktoré budú do exportu zahrnuté, a taktiež musí byť podporovaný protokol, ktorý dokáže indikovať zahltenie alebo preťaženie siete, aby nedochádzalo k strate exportovaných údajov bez upozornenia.

### 3.11.3 Spoľahlivosť

Nakoľko sa strate exportovaných paketov nedá principiálne zabrániť, musí byť aspoň indikovaná, a to na strane zberacieho procesu. Tento musí byť schopný udržiavať informácie o počte stratených záznamov. Príčiny takejto straty môžu byť nasledovné:

- limity na strane meracieho alebo exportovacieho zariadenia, napr. nedostatok pamäte, výpočtového výkonu, a pod.,
- problém s prenosom údajov v sieti – záznamy (resp. pakety vo všeobecnosti) sú z nejakých dôvodov zahadzované, a
- limity na strane zberacieho procesu – neschopnosť dostatočne rýchle spracovávať záznamy pri zahltení siete resp. pri pripojení väčšieho počtu exportovacích procesov.

V prípade, že sa na prenos používa nespoľahlivý protokol (napr. UDP), je nutné zabezpečovať spoľahlivosť na vyšších vrstvách. Prvky komunikácie medzi exportným a zberacím procesom na prídavné zabezpečenie spoľahlivosti môžu zahŕňať:

- opätovný prenos stratených záznamov,
- detekcia odpojenia alebo poruchy, a
- potvrdzovanie príjmu záznamov tokov zberacím procesom

#### **3.11.4 Anonymita tokov**

Exportný proces môže podporovať anonymizáciu zdrojových a cieľových IP adries ako aj čísel portov a iných atribútov. V prípade, že pri meraniach alebo monitorovaní prevádzky sú tieto údaje potrebné, nie je tento proces žiadúci, avšak vo všetkých ostatných prípadoch by sa tento proces mal aplikovať kvôli ochrane súkromia v sieti. Vo veľa prípadoch sú údaje získané z takto upravených záznamov postačujúce. Tento aspekt, ako aj algoritmy zabezpečujúce anonymizáciu sú stále objektom bádania, je však podmienkou, že takéto toky musia byť indentifikované, aby boli zberacím procesom odlišiteľné od bežných, nemodifikovaných tokov.

### **3.12 Požiadavky na možnosti konfigurácie**

Parametre, ktoré by musia byť konfigurovateľné na meracom procese:

- špecifikácia pozorovacieho bodu – voľba sieťového rozhrania (jedného alebo viacerých), ktoré má byť monitorované,
- špecifikácia tokov, ktoré majú byť monitorované a
- časový limit pre expiráciu tokov.

Nepovinné konfigurovateľné parametre pre merací proces:

- voľba metódy a parametrov vzorkovania, ak je podporované
- definovanie správania pri stave preťaženia

Pre exportný proces sa vyžaduje, aby bol nejakým spôsobom konfigurovateľný. Odporúčané parametre konfigurácie pre tento proces sú:

- špecifikovanie formátu exportovaných údajov, v zmysle NetFlow9 – použitie výstupu na základe šablón,

- nastavenie zberacieho procesu (jedného alebo viacerých), na ktorý majú byť údaje exportované,
- interval zasielania záznamov,
- upozornenia (notifications), ktoré majú byť zasielané zberaciemu procesu, a
- anonymizácia tokov.

Pozn.: Aj napriek architektonickému členeniu je možné, že merací a exportný proces splynú do jedného celku.

### 3.13 Všeobecné požiadavky kladené na implementáciu riešení

**Otvorenosť** – zahŕňa otvorenosť pre budúce technológie, ako aj rozšíriteľnosť konfiguračných možností meracieho a exportného procesu, a rozšíriteľnosť v zmysle dátového modelu.

**Škálovateľnosť** – musí byť podporované spracovanie dát z niekoľkých stoviek exportných procesov (samozrejme, v rámci hardvérových možností) a tieto procesy musia byť rozlíšiteľné zberacím procesom na základe jednoznačného identifikátora.

**Redundancia** – exportný proces musí byť schopný exportovať záznamy viacerým zberacím procesom, na druhej strane musí zabezpečiť aby sa tieto záznamy dali nejako identifikovať, aby sa mohli detekovať duplikované záznamy a predišlo sa tak viacnásobnému napočítavaniu tých istých dát.

### 3.14 Typy zariadení v architektúre IPFIX

Štandard nijakým spôsobom neobmedzuje architektúru zariadení ako sú sondy, routre, a iné zariadenia, ktoré vystupujú v úlohe pozorovacích bodov, meracích alebo exportných procesov. Aj keď typicky budú pozorovací bod, merací a exportný proces umiestnené na jednom zariadení, v praxi sa nevylučuje ani iná konfigurácia. Príklady rôznych riešení sú znázornené na Obr. 3-1.

**Sonda** – najjednoduchšie zariadenie na meranie a export údajov obsahuje jeden pozorovací bod, merací proces a exportný proces. V praxi si ju môžeme predstaviť ako zariadenie, ktoré sa pripojí na monitorované sieťové rozhranie.

**Základný router** – typický router, ktorý obsahuje na rozdiel od sondy viacero pozorovacích bodov.

**Rozšírený router** – môže obsahovať viacero meracích procesov, napr. pre každý zásuvný modul alebo skupinu sieťových rozhraní, pričom môže tiež obsahovať aj viac exportných procesov pre generovanie záznamov tokov odchytených rôznymi meracími procesmi.

**Konvertor protokolov** – môže prijímať údaje exportované inými protokolmi (napr. SNMP) a preposielať ich ďalej vo forme IPFIX.

**Koncentrátor** – prijíma záznamy od viacerých exportných procesov, následne tieto záznamy agreguje a exportuje ich v takejto forme.

**Proxy** – toto zariadenie jednoducho prijíma záznamy vo forme protokolu IPFIX a exportuje ich bez modifikácie ďalej. Takéto zariadenie nájde uplatnenie v prípade firewallov.

### 3.15 Bezpečnosť

Protokol IPFIX musí byť schopný prenášať údaje aj v prostredí verejného Internetu, čo so sebou prirodzene prináša aj bezpečnostné riziká, ako je odchytenie a modifikácia paketov obsahujúcich záznamy tokov potenciálnym útočníkom. Získanie takýchto záznamov a prístup k ich obsahu môže znamenať vážne riziko, pretože dáva útočníkovi informácie o aktívnych tokoch v sieti, koncových uzloch komunikácie ako aj informácie o type prevádzky. Tieto informácie môžu byť použité nielen na „špionážne“ účely, ale aj na dôkladné naplánovanie cieleného útoku na danú sieť. Je preto veľmi dôležité aby takéto údaje boli pri prenose sieťou nejakým spôsobom chránené, čo sa dá dosiahnuť šifrovaním. Ďalšou cestou je anonymizácia tokov, ktorú však, ako už bolo uvedené, nie je možné aplikovať za každých podmienok.

Taktiež musí byť chránené súkromie užívateľov prístupujúcich prostredníctvom monitorovanej siete k rôznym službám. Táto ochrana súkromia je v niektorých krajinách dokonca zákonne podchytená. Je to taktiež jedno z uplatnení anonymizácie tokov.

Ďalší aspekt môžu predstavovať falošné záznamy. Ak sa pre účtovanie používajú záznamy tokov, môže existovať motivácia na oklamanie tohto systému, čo sa dá realizovať buď modifikáciou exportovaných záznamov, alebo podstrčením falošných záznamov, ktoré by vyzerali ako keby ich vyexportoval pravý exportný proces.

Zvláštna pozornosť musí byť tejto problematike venovaná, ak sa na tieto údaje spoliehajú systémy, ktoré sledujú a strážia bezpečnosť v sieti. V takomto prípade by útočník podstrčením špeciálne pripravených falošných záznamov vedel oklamať aplikáciu monitorujúcu výskyt rôznych typov útokov. V tejto súvislosti ešte treba spomenúť možnosť útoku na samotné zariadenie architektúry IPFIX. Typickým príkladom je útok typu DoS (Denial Of Service), kde pri „push“ modeli môže byť zberací proces, nakoľko stále očakáva údaje, zahltený množstvom falošných záznamov generovaných útočníkom.

Aby bolo možné vyhovieť bezpečnostným požiadavkám rôznych používateľov, IPFIX musí poskytovať rôzne úrovne zabezpečenia. IPFIX poskytuje štyri úrovne zabezpečenia, pričom prvá úroveň predstavuje úroveň bez zabezpečenia. V prípade, že komunikácia medzi IPFIX zariadením a kolektorom je považovaná za bezpečnú, nemusí byť vyžadované žiadne ďalšie zabezpečenie. Táto možnosť umožňuje najefektívnejšiu cestu komunikácie, pretože nie sú potrebné žiadne špeciálne bezpečnostné operácie, ktoré by predstavovali zvýšenú záťaž.

### **3.15.1 Bezpečnosť na základe autentifikácie**

Táto úroveň bezpečnosti poskytuje záruku integrity a autentičnosti dát. Údaje vymieňané medzi IPFIX zariadením a kolektorom sú chránené signatúrou, alebo podpisom pravosti. Každá modifikácia týchto dát bude odhalená príjemcom a bude viesť k ich odmietnutiu. Aj keď táto metóda dokáže zabezpečiť detekciu neplatnosti dát, nedokáže zabezpečiť ich utajenie. Používateľ IPFIX by nemal použiť túto možnosť v prípade, že sa prenášajú citlivé alebo dôverné údaje. Túto úroveň zabezpečenia je možné dosiahnuť prostredníctvom:

- TCP s použitím hašovacej funkcie MD5 alebo
- autentifikačnej hlavičky IP paketu (IP authentication header [21]).

### **3.15.2 Šifrovanie**

Šifrovanie dát predstavuje najúčinnější prostriedok ochrany prenášaných dát. Dáta IPFIX sú šifrované odosielateľom a jedine správny príjemca ich môže dešifrovať a pristupovať k ich obsahu. Táto úroveň zabezpečenia musí byť použitá keď prenos medzi exportérom a kolektorom nie je bezpečný a dáta IPFIX musia byť chránené. Na tento

účel sa odporúča využiť bezpečnostné funkcie transportnej vrstvy. Prostriedky na dosiahnutie tejto úrovne ochrany:

- ESP (Encapsulating Security Payload [22]),
- použitie bezpečného protokolu transportnej vrstvy.

Šifrovanie dát pridáva režiu samotnému prenosu IPFIX dát. Môže teda obmedziť rýchlosť, akou je exportér schopný odosielať dáta kolektoru vzhľadom na zvýšené nároky na hardvérové prostriedky zariadenia.

### **3.15.3 Autentifikácia koncového bodu**

Je dôležité pre IPFIX zariadenie uistiť sa, že komunikuje s pravým kolektorom a nie s falošným. Rovnaký prístup platí aj z pohľadu kolektora, ktorý môže požadovať overenie pravosti IPFIX zariadenia, od ktorého prijíma údaje. Architektúra IPFIX by mala zabezpečiť možnosť autentifikácie tak, aby sa mohla vykonať buď jednosmerná alebo vzájomná autentifikácia medzi IPFIX zariadením a kolektorom. Na splnenie týchto požiadaviek by sa mali využiť existujúce protokoly transportnej vrstvy ako je TLS (Transport Layer Security [60]) alebo IPSEC (IP Security [61]).

### **3.15.4 Prevencia útokov typu DoS**

Nakoľko jedným z potenciálnych využití IPFIX je detekcia prienikov do sietí, je dôležité aby architektúra IPFIX bola nejakým spôsobom odolná voči útokom typu DoS. V prípade, že sa monitorovaná sieť stane terčom útoku, môže to viesť k záplave siete množstvom IPFIX správ. IPFIX systém by sa mal pokúsiť odchytiť a spracovať čo najviac takýchto správ, avšak ich obrovské množstvo môže spôsobiť jeho preťaženie. Zariadenie IPFIX sa môže stať terčom útoku typu DoS rovnako ako každé iné zariadenie v sieti. V takomto prípade hovoríme o všeobecnom DoS útoku. O IPFIX špecifickom DoS útoku môžeme hovoriť vtedy, ak sa útočník pokúsi zaplaviť kolektor množstvom falošných IPFIX správ. Jednou možnosťou ako vyriešiť tento problém je periodická synchronizácia sekvenčných čísel jednotlivých záznamov tokov medzi exportnými a zberacími procesmi. Problém môže byť do značnej miery vyriešený pri použití šifrovania riadiacich správ.

## 4. Protokoly vyhovujúce štandardu IPFIX

### 4.1 CRANE [29]

Protokol vyvinula spoločnosť XACCT Technologies. Názov reprezentuje skratku Common Reliable Accounting for Network Element. Protokol umožňuje efektívny a spoľahlivý prenos dát ľubovoľného typu, primárne sa však jedná o údaje používané na účtovanie služieb, zo sieťových prvkov. Protokol je navrhnutý tak, aby dokázal splniť kritické požiadavky kladené na export veľkých objemov dát takéhoto typu zo sieťových prvkov pri efektívnom využívaní sieťových, úložných a výpočtových prostriedkov.

Exportovacia strana sa nazýva CRANE klient, prijímacia strana CRANE server. Vzhľadom na klasickú funkciu architektúry klient-server v tomto prípade netradične server má za úlohu nadviazať spojenie s klientom. Kvôli robustnosti zo strany klienta môže existovať viacero simultánných spojení s rôznymi servermi. V takomto prípade má každý server priradenú určitú prioritu a klient exportuje údaje len funkčnému serveru s najvyššou prioritou. Klient a server medzi sebou komunikujú zásadne prostredníctvom spoľahlivého protokolu (TCP alebo SCTP). Aj keď TCP spĺňa minimálne požiadavky, preferovaný je protokol SCTP, a to z nasledujúcich dôvodov:

- autentifikácia na úrovni spojenia,
- záznamovo orientované (message based) posielanie údajov, oproti prúdovo orientovanému (stream oriented) v prípade TCP,
- rýchla detekcia zlyhania (poruchy) spojenia.

Komunikácia je obojstranná počas celej doby trvania spojenia. Existuje 20 rôznych typov správ, je podporovaná negociácia šablón, a to nielen pri nadviazaní, ale aj uprostred spojenia. Protokol podporuje správy s max. dĺžkou až  $2^{32}$  bajtov, čo si môže vyžadovať schopnosť fragmentácie správ na transportnej vrstve v prípade použitia protokolu IP. Spoľahlivosť je dosahovaná kombináciou služieb transportnej vrstvy a vlastností samotného protokolu. Transportná vrstva zabezpečuje spoľahlivé doručovanie správ a detekciu poruchy spojenia, CRANE protokol zabezpečuje posielanie potvrdení spracovania týchto správ po bezpečnom uložení obsahu na trvalom úložisku. V prípade použitia protokolu TCP musia byť splnené nasledujúce požiadavky:

- CRANE klient musí čakať na spojenie na špecifickom TCP porte,

- CRANE server sa musí pripojiť na tento port klienta a je zodpovedný za opätovné nadviazanie spojenia v prípade výskytu chyby,
- správy sú posielané ako sekvencie bajtov, pričom veľkosť správy je špecifikovaná v hlavičke správy CRANE

Po nadviazaní spojenia ešte pred začiatkom samotného prenosu dát sa musia obidve strany dohodnúť na použitej šablóne. Ak vznikne potreba zmeny šablóny, obidve strany sú schopné dohodnúť sa na novej šablóne aj počas spojenia. Je veľmi dôležité, aby všetci klienti a servery mali k dispozícii rovnakú množinu šablón.

Po bezpečnostnej stránke predstavuje protokol dva koncové systémy, ktoré spolu komunikujú cez spoľahlivý transportný protokol. Pred uskutočnením akejkoľvek výmeny dát, musí byť na strane klienta vytvorený zoznam adries jednotlivých CRANE serverov, s ktorými bude možné komunikovať. Na strane servera podobne existuje zoznam platných CRANE klientov. Vzhľadom na to, že táto schéma je statická, je systém náchylný na útoky typu address spoofing. Protokol sám osebe neposkytuje silné bezpečnostné mechanizmy a nedokáže zaistiť ochranu prenášaných dát. Výrazne sa preto odporúča použitie bezpečnostných stratégií na zabezpečenie takejto ochrany napríklad využitím IPSEC na sieťovej, alebo TLS na transportnej vrstve.

## 4.2 Diameter [30]

Ide o ďalší vývojový stupeň protokolu RADIUS (Remote Authentication Dial In User Service), ktorý je používaný na overovanie autentifikačných a autorizačných údajov. Diameter je zovšeobecnený a rozširovateľný protokol určený na podporu AAA (Authentication, Authorization, Accounting) požiadaviek rôznych aplikácií. Typickým príkladom sú vytáčané (dialup) služby a technológia Mobile IPv4.

Z pohľadu architektúry sa jedná o protokol typu peer-to-peer, je definovaných 14 riadiacich správ, organizovaných ako 7 párov požiadavka/odpoveď, z ktorých sa v čistej aplikácii IPFIX použijú len niektoré. Diameter zahŕňa možnosti negociácie schopností a mechanizmus informovania o chybách, taktiež definuje pravidlá pre bezpečnosť komunikácie medzi dvoma koncovými systémami. Na zaistenie autentifikácie a šifrovania môžu byť použité TLS alebo IPSEC.

Dáta sú posielané vo forme párov atribút/hodnota (Attribute Value Pair, AVP). AVP pozostáva z 8-bajtovej hlavičky a tela, ktorého dĺžka závisí od formátu

prenášaných dát. Je preddefinované množstvo takýchto formátov, ktoré zahŕňajú prenos 32 alebo 64 bitových, znamienkových alebo neznamienkových hodnôt, ako aj podpora prenosu adres IPv4 a IPv6, a mnoho ďalších. Rovnako je možné dedefinovať nové formáty AVP, ktoré by napríklad mohli reprezentovať informácie o tokoch v sieti.

Diameter predstavuje vysoko flexibilný protokol určený pre splnenie rozličných požiadaviek AAA, ktorý v určitom zmysle vysoko prekračuje potreby IPFIX.

### 4.3 LFAP

Pôvodne ide o skratku Lightweight Flow Admission Protocol, a bol používaný na vyhodnocovanie rozhodovacích kritérií na tokovo založených routoch, ako aj na poskytovanie štatistík pre jednotlivé toky. Neskoršia verzia obmedzila funkčnosť na poskytovanie štatistík a protokol bol premenovaný na Lightweight Flow Accounting Protocol.

Exportér v LFAP architektúre sa nazýva CCE (Connection Control Entity), kolektor je FAS (Flow Accounting Server). Tieto prvky komunikujú medzi sebou prostredníctvom protokolu TCP. LFAP definuje 13 typov správ, ktoré zahŕňajú operácie pre menežovanie spojenia, negociáciu verzií, správy týkajúce sa informácií o tokoch a administratívne príkazy. V prípade požadovania autentifikácie alebo šifrovania sa môže použiť IPSEC alebo TLS. Navyše, LFAP poskytuje štyri úrovne zabezpečenia použitím autentifikácie HMAC-MD5 a DES-CBC metódy šifrovania.

Pozn.: V súčasnosti sa DES nepovažuje za adekvátnu ochranu, pretože kvôli nedostatočnej veľkosti kľúča je náchylná na útoky typu brute-force.

Dáta sú kódované vo forme TLV (Type, Length, Value) so 4-bajtovou réžiou na každú dátovú položku (2 bajty označujúce typ správy a ďalšie 2 bajty jej dĺžku). LFAP rozoznáva dva typy správ určené pre informácie o tokoch. Prvou z nich je správa FAR (Flow Accounting Request), ktorá je posielaná vždy pri vzniku nového toku a slúži len na informovanie o vzniku jednotlivých tokov. Na samotné posielanie účtovacích informácií sa používa správa FUN (Flow Update Notification). Kolektor musí udržiavať databázu tokov podľa prijatých správ typu FAR a aktualizovať informácie pre príslušné toky po prijatí správ FUN.

Popis protokolu LFAP taktiež definuje množinu užitočných štatistík pre účely účtovania. Zaujímavou možnosťou je možnosť menežovania prvkov LFAP architektúry prostredníctvom protokolu SNMP.

#### **4.4 Streaming IPDR**

Ide o produkt organizácie Internet Protocol Detail Record (IPDR). Na komunikáciu sa používa protokol TCP. Zasielanie správ je realizované v jednom z dvoch režimov: triviálne TCP doručovanie, alebo potvrdzované TCP doručovanie. Druhý režim používa na zvýšenie spoľahlivosti potvrdzovanie na úrovni aplikačnej vrstvy.

Spojenie medzi exportérom a kolektorom je jednosmerné. Akonáhle dôjde k vytvoreniu spojenia, exportér pošle kolektoru hlavičku s množinou deskriptorov popisujúcich záznamy a po potvrdení príjmu tejto správy kolektorom začne kolektoru posilať dáta vo forme ktorá zodpovedá odoslanému popisu. V ľubovoľnom okamihu komunikácie môže exportér poslať novú množinu deskriptorov. Tento typ komunikácie pretrváva až do ukončenia spojenia. Správy posielané exportérom sú číslované, aby sa predišlo viacnásobnému spracovaniu tej istej správy v prípade poruchy pri spojení. Na toto číslovanie sa odvoláva aj potvrdzovanie správ aplikačnou vrstvou, ak je použitý režim potvrdzovaného doručovania.

#### **4.5 NetFlow9 [32]**

Tento protokol bol vyvinutý firmou Cisco Systems a číslo 9 v názve hovorí o jeho deviatej verzii. Ide o zovšeobecnenú a vylepšenú piatu verziu, ktorá bola široko implementovaná a využívaná na export a zber informácií o tokoch v sieťach. Značná výhoda spočívala v tom, že firma Cisco je popredný svetový výrobca aktívnych sieťových prvkov a implementácia tohto protokolu v routroch umožnila získavanie údajov priamo z týchto zariadení. Piata verzia protokolu mala napevno nadefinované typy informácií, ako aj formu, v ktorej boli tieto informácie exportované.

##### **4.5.1 Transportný protokol**

NetFlow9 ako jediný zo spomínaných protokolov primárne používa na export dát protokol UDP, ktorý nie je spojovo-orientovaný. Je však navrhovaný tak, aby bol

---

nezávislý na transportnom protokole a vzhľadom na jednosmerný charakter komunikácie medzi exportérom a kolektorom by malo byť relatívne jednoduché pridať podporu ďalších protokolov, napr. TCP alebo SCTP. V prípade SCTP sa ďalej odporúča namapovanie riadiacich a údajových správ na rôzne prúdy v rámci jedného spojenia. V takomto prípade by nadväzovanie spojenia bolo realizované exportérom. Využitie protokolu UDP má oproti ostatným spojovo-orientovaným protokolom jednu výhodu, ktorá spočíva v jeho menších nárokoch na systémové aj sieťové prostriedky. Dôvodom tejto voľby bola myšlienka možnosti exportu veľkých objemov dát. Na druhej strane, UDP má viacero nevýhod a v konečnom dôsledku nevyhovuje ani požiadavkám IPFIX.

#### 4.5.2 Princíp komunikácie, šablóny

Exportér posiela kolektoru dáta vo forme správ FlowSet. FlowSet je skupina jedného alebo viacerých záznamov tokov, resp. šablón, ktoré ich popisujú. Exportér je tiež zodpovedný za vytváranie týchto záznamov na základe údajov z odpozorovaných paketov. Šablóny popisujúce formát záznamov tokov musia mať pridelené nejaký identifikátor ID. Jedinečnosť tohto ID nie je garantovaná medzi viacerými pozorovacími doménami. Ak z nejakého dôvodu (preťaženie, zásah administrátora, ...) dôjde k reštartu exportéra, ani v takomto prípade nie je zaručená konzistencia nových ID šablón s predošlými. Ak kolektor obdrží nový popis šablóny s existujúcim ID, musí pôvodnú šablónu zahodiť. Táto situácia môže nastať po reštarte exportéra, pretože tento je povinný pred posielaním akýchkoľvek dát poslať šablónu popisujúcu ich formát. Životnosť šablón je obmedzená, preto exportér musí periodicky posielat' popisy šablón v závislosti od konfigurácie, či už v pravidelných časových intervaloch, alebo podľa počtu odoslaných správ.

Kolektor by mal dostať popisy šablón ešte pred samotnými dátami. Ak sa však stane, že dostane dáta pre ktoré nemá popis zodpovedajúcej šablóny, mal by tieto dáta dočasne uložiť, počkať na príchod príslušnej šablóny a následne tieto dáta spracovať. To všetko sa však musí diať za súčasného príjmu a spracovávania ostatných dát, a v prípade, že kolektor nedostane šablónu v dostatočne krátkom čase, môže dôjsť k zaplneniu jeho pamäťových prostriedkov. Životnosť šablón je obmedzená a ako bolo spomenuté, exportér by sa mal postarať o ich periodické zasielanie. Ak kolektor nedostane tieto informácie v určitom nastavenom intervale, ktorý by mal byť, s

ohľadom na dobu prenosu v sieti, o niečo dlhší ako ten nastavený na exportéri, zahodí príslušné šablóny. Pri tomto procese platí prísne pravidlo, že kolektor sa nesmie pokúšať dekódovať prijaté dáta podľa šablóny, pre ktorú vypršala doba platnosti.

### 4.5.3 Bezpečnosť

Pôvodný návrh protokolu NetFlow9 nebral do úvahy žiadne bezpečnostné kritériá, pretože sa predpokladalo, že exportér a kolektor sa budú nachádzať v rámci jednej privátnej siete a zapracovanie bezpečnostných mechanizmov priamo do komunikačného protokolu by malo za následok zíženie efektívnosti komunikácie. V prípade, že budú dáta prenášané verejnými sieťami (napr. Internetom), je potrebné dodržať pravidlá bezpečnosti, ktoré špecifikuje štandard IPFIX.

Jedno z hľadísk, ktoré neboli v IPFIX štandarde podrobnejšie rozobraté je odchytenie a modifikácia záznamov obsahujúcich definície šablón. Odchytenie takéhoto záznamu poskytuje útočníkovi schopnosť interpretovať všetky záznamy, ktoré budú exportované vo formáte, ktorý táto šablóna špecifikuje. Na druhej strane, modifikácia šablóny a jej „podsunutie“ kolektoru môže ohroziť správnu interpretáciu dát kolektorom. Špeciálne pripraveným obsahom takejto falošnej šablóny môže nastať viacero situácií:

- Útočník modifikuje poradie jednotlivých položiek v šablóne, resp. ich zamení za iné, pričom celková dĺžka ostane nezmenená. Túto situáciu nie je možné na strane kolektora odhaliť a má za následok chybnú interpretáciu dátových záznamov a tým pádom ich znehodnotenie. V úložisku, kam kolektor takéto údaje ukladá, sa objavia nezmyselné dáta;
- Útočník modifikuje jednotlivé položky v šablóne tak, že jej dĺžka nezodpovedá dĺžke pôvodnej šablóny. Odhalenie tejto situácie závisí na type protokolu použitého na prenos dát medzi exportérom a kolektorom:
  - V prípade, že je použitý protokol UDP alebo SCTP, je túto situáciu možné jednoducho odhaliť;
  - V prípade, že je použitý protokol TCP, si jej odhalenie vyžaduje dodatočné kontrolné mechanizmy na strane kolektora a v prípade ich absencií je možné úplné zlyhanie prijímania dátových záznamov od exportéra v zmysle nesprávnej interpretácie aj záznamov, ktoré sa týkajú iných šablón.

Z uvedených faktov vyplýva dôležitosť použitia bezpečného protokolu na komunikáciu medzi exportérom a kolektorom, minimálne na účely prenosu riadiacich informácií, ktoré v tomto prípade predstavujú definície šablón.

## 4.6 Porovnanie vhodnosti využitia jednotlivých protokolov

- Vysoko-výkonné merania tokov. Na tento účel sú najvhodnejšími kandidátmi protokoly NetFlow9 a LFAP. Výhoda protokolu LFAP spočíva v menších pamäťových nárokoch na exportér, ktoré vyplývajú z priebežného posielania atribútov kolektoru. NetFlow naopak vyžaduje, aby množina týchto atribútov bola udržiavaná v pamäti až do okamihu ich exportu. Naopak, pri výskyte veľkého počtu „krátko žijúcich“ tokov bude v prípade protokolu LFAP počet exportovaných správ výrazne vyšší a kolektor bude musieť udržiavať stav aktívnych tokov v pamäti až do doby ich ukončenia.
- Viacúčelové účtovanie. Sem sa svojimi možnosťami najlepšie hodia protokoly Streaming IPDR a CRANE. Oba protokoly sú navrhnuté tak, aby boli rozširovateľné najmä vzhľadom na veľký rozsah úloh zameraných na účtovanie.
- Všeobecnoúčelové AAA. Protokol diameter má ako jediný veľmi široké využitie pri autentifikácii, autorizácii ako aj účtovaní. Je to najmä dôsledok toho, že za základ bol zobraňovaný protokol RADIUS, ktorý je určený na účely autentifikácie a autorizácie.

## 5. Existujúce implementácie protokolu NetFlow

### 5.1 Flow-tools [34]

Ide o skupinu nástrojov určených na zber, posielanie, spracovanie a generovanie výsledkov z dát získaných prostredníctvom protokolu NetFlow. Nástroje môžu byť použité na jednom stroji, alebo môžu byť rozmiestnené na viacerých prvkoch v sieti. Verzie protokolu, ktoré sú podporované v súčasnosti sú, 1, 5, 6, 7 a 8. Čo sa týka 8.verzie, tá sa ako prvá priblížila myšlienke používania šablón, tie sú však nadefinované napevno a sú chápané ako subverzie. Tento nástroj podporuje zber 14 týchto subverzií. Vzhľadom na to, že nie je podporovaná 9.verzia protokolu NetFlow, nebudem sa týmto

nástrojom zaoberať podrobnejšie, aj keď každopádne ide o zaujímavé riešenie. Tento projekt sa stále vyvíja, takže je predpoklad budúcej podpory aj tejto verzie.

## 5.2 Stager [35]

Je to všeobecný nástroj na ukladanie, agregáciu a prezentáciu štatistík týkajúcich sa sieťovej prevádzky. Pozostáva z webovskej aplikácie na prezentáciu dát a perlovskej back-endovej časti na ukladanie a agregáciu dát. Práve druhá spomínaná súčasť je určená na spracovanie dát o tokoch v sieti vo formáte NetFlow. Táto časť je však riešená modulárne, je preto možné rozšíriť možnosti o odchytyvanie aj z iných zdrojov. Vyhodnocovacia časť je vybavená množstvom funkcií a je rozširovateľná. Zobrazovanie štatistík je možné v grafickej alebo tabuľkovej forme a typy generovaných štatistík zahŕňajú zobrazenie podľa:

- cieľových sieťových rozhraní,
- IP protokolov,
- IP ToS (Type Of Service),
- zdrojových a cieľových IP adries,
- zdrojových a cieľových autonómnych systémov,
- zdrojových a cieľových portov transportnej vrstvy,
- sumárne štatistiky.

## 5.3 Softflowd [36], flowd [37]

Softflowd je analyzátor sieťovej prevádzky s podporou tokov, schopný exportovať dáta vo formáte NetFlow verzie 1, 5 a 9. Podporované štatistiky zahŕňajú sledovanie min., max., priemerného a sumárneho počtu bajtov a paketov na úrovni jednotlivých tokov, resp. protokolov. Plne je podporovaný protokol IPv6, a to jednak čo sa týka sledovania prevádzky, tak aj podpora exportu na IPv6 kolektory. Akýkoľvek štandardný kolektor by mal byť schopný prijímať údaje od tohto programu. Do tejto doby je verzia 0.9 určená pre operačné systémy Linux a BSD, je zahrnutá aj experimentálna podpora Solarisu. V prípade sledovania prevádzky Softflowd beží ako démon. K programu je pribalený aj program **softflowctl**, ktorý slúži na vzdialené ovládanie a extrakciu štatistík z bežiaceho démona softflowd. Zaujímavou možnosťou je možnosť exportovania údajov zo súborov vytvorených programom tcpdump.

Flowd je popísaný ako malý, rýchly a bezpečný NetFlow kolektor. Dokáže rozpoznávať verzie 1, 5, 7 a 9 s podporou IPv4 aj IPv6. Podpora IPv6 sa týka tokov, ale aj exportovaných dát. Podporuje filtrovanie a značkovanie tokov (použitím syntaxe linuxového packet filtra). Umožňuje ukladať zozbierané informácie v kompaktnom binárnom formáte, pričom je možné za behu programu meniť zoznam atribútov, ktoré sa majú uložiť. Na čítanie týchto súborov je k dispozícii programové rozhranie v jazykoch Perl a Python. Zaujímavou vlastnosťou je podpora zberu multicastových exportov (pre multicastové skupiny IPv4 aj IPv6), čím je možné vytvárať systémy redundantných kolektorov. Ide o jednoúčelový nástroj na zber exportovaných dát, ktorý ich analýzu prenecháva iným programovým prostriedkom.

## 5.4 nProbe, nTop [38]

Ako už vyplýva z názvu nástroja, ide o softvérovú sondu, určenú pre Unixové OS (vrátane MacOS X) a Windows. Dokáže exportovať údaje o prevádzke v sieti vo formáte NetFlow verzie 5 a 9 s plnou podporou IPv4 a IPv6. Ponúka tiež možnosť ukladania dát na disk pre neskoršiu analýzu. Nespornou výhodou je možnosť exportu dát vo formáte nFlow, ktorý bude spomenutý neskôr. Je súčasťou programového balíka nTop, ktorý ďalej obsahuje kolektor podporujúci NetFlow a sFlow. Možnosti vyhodnocovania prevádzky zahŕňajú:

- zotriedenie dát sieťovej prevádzky podľa protokolov a rôznych iných kritérií,
- zobrazovanie štatistík o sieťovej prevádzke,
- ukladanie štatistík v RRD (Round Robin Database) formáte,
- identifikáciu hostiteľského OS,
- zobrazenie distribúcie prevádzky vzhľadom na jednotlivé protokoly,
- zobrazenie komunikačnej matice prevádzky pre jednotlivých hostiteľov aj siete,
- a iné.

## 5.5 Ostatné nástroje

**NFDUMP** [39] – sada nástrojov podporujúcich zber a spracovanie dát exportovaných protokolom NetFlow verzie 5 a 7.

**NFSen** [40] – sratka od NetFlow Sensor, je to grafické rozhranie k programu nfdump, založené na prístupe cez web.

**ManageEngine™ NetFlow Analyzer 4** – webová aplikácia na monitorovanie šírky pásma (bandwidth).

## 6. sFlow [28][42]

Tomuto riešeniu je vzhľadom na jeho jedinečnosť venovaná zvláštna kapitola. Avšak vzhľadom nato, že nespĺňa požiadavky architektúry IPFIX, sa zameriam len na stručný popis a nebudem mu venovať väčšiu pozornosť. Ide o komplexný nástroj implementujúci vzorkovaciu technológiu vyvinutý viacerými výrobcami určený pre merania parametrov prevádzky vo vysokorýchlostných sieťach. Ponúka možnosť kontinuálneho monitorovania prevádzky a detailnú analýzu tokov od linkovej až po aplikačnú vrstvu so žiadnym ovplyvnením forwardovacích rozhodnutí paralelne na všetkých sieťových rozhraniach pri plnej rýchlosti. Architektúra je reprezentovaná takzvanými sFlow agentmi, ktoré sú implementované v sieťových zariadeniach (route, switche, sondy) a vystupujú v úlohe exportérov, a sFlow kolektora, ktorý zbiera dáta vyslané týmito agentmi. Uvádza sa, že systém je schopný monitorovať siete s priepustnosťou za hranicou 10 až 100 Gbps. Jedným sFlow kolektorom sa dajú monitorovať tisíce zariadení. Z hľadiska architektúry sa jedná o riešenie sčasti odlišné od štandardu IPFIX, pretože dáta sa vyhodnocujú priamo na kolektore. Z tohto riešenia postupom času vznikol štandard, ktorý je čoraz viac implementovaný výrobcami sieťových prvkov. Jeho výhoda oproti riešeniu na báze NetFlow je, že dokáže analyzovať prevádzku nielen z routrov ale aj zo switchov, pretože spracováva informácie už na linkovej vrstve, na rozdiel od NetFlow-u, ktorý síce dokáže do záznamov tokov zahŕňať niektoré informácie z druhej vrstvy (číslo sieťového rozhrania), pracuje však na sieťovej vrstve, čo ho predurčuje výhradne na použitie exportu dát z routrov. Vzhľadom na túto skutočnosť je možné pomocou sFlow-u monitorovať okrem sietí bežiacich na protokole IP aj protokoly IPX a AppleTalk. Ďalšou nespornou výhodou je okrem statickej konfigurácie aj možnosť konfigurácie cez SNMP.

## 7. Projekt BasicMeter

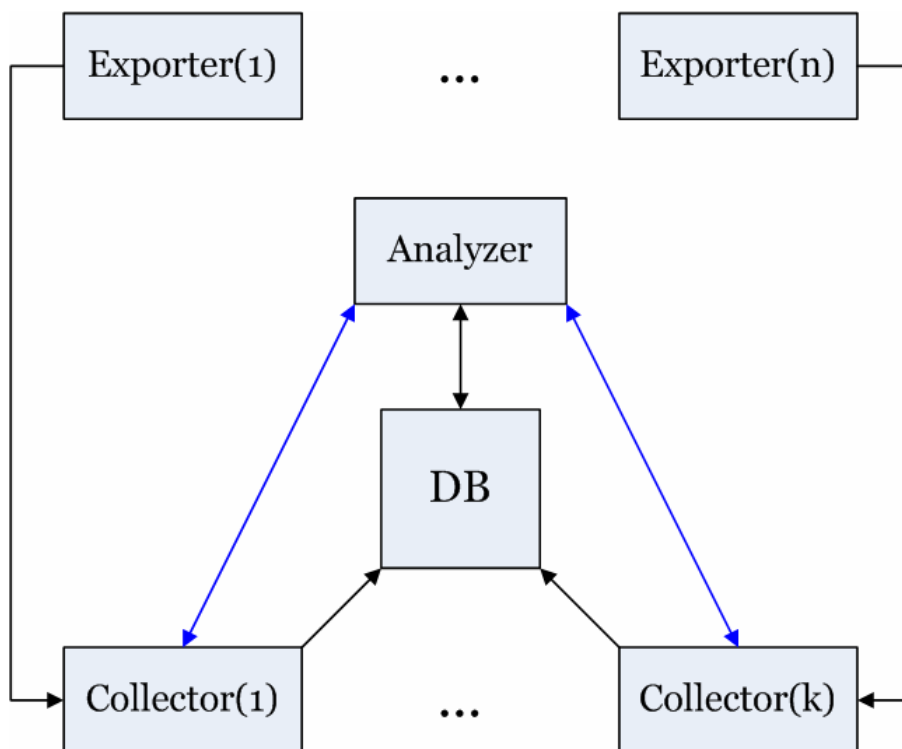
### 7.1 Vznik a vývoj

Tento nástroj sa začal vyvíjať zhruba pred dvomi rokmi a dôvodom jeho vzniku bola potreba nástroja na meranie prevádzkových parametrov počítačových sietí za účelom zabezpečenia požiadaviek QoS. Vtedajší basicmeter predstavoval merací nástroj určený pre UNIX-ové operačné systémy, schopný odchytať a zaznamenávať údaje o sieťovej prevádzke. Samotné zobrazovanie časových závislostí parametrov QoS bolo realizované externými programami (gnuplot). Neskôr bol nástroj od základu prepracovaný a dá sa povedať, že okrem ponechania si základnej myšlienky sa jednalo o úplne nový program, pričom sa už počítalo s jeho budúcim začlenením do vtedy ešte vyvíjajúceho sa štandardu IPFIX. Aby mohol splniť toto kritérium, musela byť pridaná podpora exportu zaznamenaných dát. Nástroj bol navrhnutý modulárne, pretože bola plánovaná implementácia vzorkovacích algoritmov, aby bolo možné nástroj využiť aj na merania vo vysokorýchlostných sieťach.

### 7.2 Architektúra nástroja

Približne v čase, keď bol IPFIX uznaný ako štandard, došlo k vzniku projektu BasicMeter, ktorého cieľom je vytvoriť komplexnú meraciu platformu postavenú na báze architektúry IPFIX spolu s vyhodnocovacou aplikáciou. V súčasnosti sa pracuje na rozširovaní prepracovanej verzie pôvodného basicmetra, implementuje sa podpora exportovania dát vo formáte NetFlow9 ako aj podpora niektorých vzorkovacích techník. V čase odovzdania tejto práce by mala byť podpora NetFlow9 plne funkčná. Ako programovací jazyk bol použitý C++. Nástroj sa nazýva jednoducho – BM exporter (BasicMeter sa zvykne označovať skratkou BM). Ďalším prvkom meracej architektúry je nástroj na zachytávanie dát posielaných exporterom – kolektor. Ide o aplikáciu vytvorenú v jazyku Java, ktorá je schopná prijímať záznamy tokov vo formáte NetFlow verzie 5 a 9 súčasne od viacerých exportérov a tieto dáta ukladať do databázy, resp. posielat' analyzujúcej aplikácii. Názov programu – JXColl, je odvodený od skratky Java XML Collector. XML značí možnosť rozpoznávania a využívania obsahu prichádzajúcich paketov na základe ich popisu pomocou XML. Poslednou časťou tejto skupiny nástrojov je program určený na analýzu dát zozbieraných kolektorom a jeho názov je

BM Analyzer. Zobrazenie architektúry meracej platformy BasicMeter je na Obr. 7.1. Obrázok odráža funkčnosť riešenia v čase písania tejto práce. Pri spustení exportera si tento načíta konfiguráciu zo súboru, kde má špecifikovanú IP adresu a port, na ktorom bude kolektor zberať exportované údaje, ako aj ostatné parametre, ako je napríklad formát šablóny, podľa ktorej bude exportovať dáta, prípadne typ vzorkovania, a iné. Čierne šípky teda znázorňujú prenos dátových informácií smerom od exportera ku kolektoru. Modré šípky znázorňujú prenos dátových a riadiacich informácií medzi kolektorom a analyzerm. Tento proces, ako aj protokol používaný na túto komunikáciu bude vysvetlený neskôr. Obojsmerná šípka medzi analyzerm a databázou značí jednak autentifikačný proces, kde sa analyzer autentifikuje voči databáze, na druhej strane odráža možnosť analyzera vyžiadať si požadované informácie z databázy a ich následný prenos.



**Obr. 7.1: Architektúra meracej platformy BasicMeter.**

### 7.3 BM Analyzer

Ako bolo uvedené v časti 3.4.1 (Obr. 3.1), aplikácie určené na vyhodnocovanie nie sú popísané v samotnej architektúre, čo sa pri vývoji tejto aplikácie ukázalo ako dobrá voľba. Program, ktorý sa dnes vyvíja, bol od svojej prvej verzie značne prepracovaný a jedným z hlavných dôvodov boli práve výkonnostné problémy pri väčšom množstve nazbieraných dát. Ako primárne využitie programu bolo zamýšľané vyhodnocovanie dát uložených v databáze, mal teda slúžiť na neskoršiu analýzu. Neskôr sa ukázalo ako veľmi potrebné môcť vyhodnocovať dáta, ktoré pretečú sieťovými prvkami priamo za behu, teda v reálnom čase. Túto požiadavku nie je možné celkom uspokojiť, pretože už zo samotnej architektúry IPFIX vyplýva určité oneskorenie spracovania monitorovaných dát. Toto oneskorenie je ovplyvnené spracovaním dát v exportéri, prenosom týchto dát v sieti, ich spracovaním v kolektore a následným prenosom a spracovaním v analyzujúcej aplikácii. Napriek tomu má význam sa týmto riešením zaoberať, ak sa podarí analyzovať údaje v dostatočne krátkom čase, teda s prijateľným oneskorením. Dalo by sa polemizovať o tom, aké oneskorenie je prijateľné a ako odpoveď na to si dovoľím tvrdiť, že to závisí od aplikácie, na akú bude toto monitorovanie využívané. Toto oneskorenie je možné je možné v prípade potreby toto znížiť použitím výkonnejších zariadení a rýchlejších prenosových liniek. Pôvodná myšlienka bola taká, že sa dáta zozbierané kolektorom budú ukladať do databázy, odkiaľ ich bude analyzer čítať a spracovávať. Toto riešenie sa ukázalo pre prípad vyhodnocovania v reálnom čase absolútne nevhodné, pretože s nárastom množstva údajov v databáze sa realizácia dotazov stávala časovo čoraz náročnejšia. Dospelo sa do stavu, kedy pri naplnení databázy údajmi o prevádzke monitorovanej v časovom rozpätí jedného týždňa, bolo potrebných na získanie výsledku niekoľko desiatok sekúnd. Tento čas je už na hranici únosnosti pre vyhodnocovanie histórie prevádzky, aj keď v tomto prípade má vplyv len na samotnú odozvu aplikácie. Pre prípad kontinuálneho vyhodnocovania bol zvolený prístup získavania údajov priamo z kolektora, a to bez narušenia existujúcej koncepcie návrhu. Kolektor by mal permanentne zabezpečovať ukladanie dát do databázy, navyiac by v prípade potreby paralelne posielal tie isté údaje (resp. časť z nich) smerom analyzeru. Toto riešenie predstavuje istú redundanciu posielaných dát, avšak čiastočne odbúrava problém s výkonnosťou databázového systému. Na zefektívnenie komunikácie medzi kolektorom a analyzerom sa používa

vlastný komunikačný protokol, ktorý kladie nižšie nároky na systémové a sieťové prostriedky, ako protokol NetFlow9. V takomto prípade je možné monitorovať prevádzku s určitým, približne konštantným, časovým oneskorením.

### 7.3.1 Návrh a analýza riešenia

Ako programovací jazyk bol zvolený jazyk Java, v budúcnosti sa navyše počíta s využitím technológie Java Web Start. Výhoda takéhoto riešenia spočíva najmä v jeho multiplatformovom využití, na druhej strane technológia Web Start umožňuje spúšťanie aplikácie priamo z prehliadača webovských stránok a hlavne zabezpečuje automatickú aktualizáciu v prípade, že je k dispozícii nová verzia programu.

Pri kontinuálnom monitorovaní prevádzky sa dáta získavajú priamo z kolektora. Na komunikáciu je vyvinutý vlastný aplikačný protokol, ktorý je navrhnutý s ohľadom na jednoduchosť a efektívnosť komunikácie. Z hľadiska bezpečnosti je použitá autentifikácia voči kolektoru, aj keď do budúcnosti sa počíta s obojsmernou autentifikáciou. Žiadne ďalšie bezpečnostné mechanizmy nie sú implementované. Na komunikáciu sa používa TCP, ale v bolo by možné použiť ľubovoľný spojovo-orientovaný spoľahlivý protokol. Komunikácia je obojsmerná a funguje na princípe komunikácie klient-server, pričom klientskú stranu predstavuje analyzer. Kolektor teda čaká na pripojenie na dohodnutom porte (štandardne 2138). Po pripojení analyzera na kolektor, analyzer automaticky posielajú meno a heslo v šifrovanej podobe (MD5). V prípade úspešnej autentifikácie dôjde k nadviazaniu spojenia. V takomto stave kolektor neposiela žiadne dáta, je však pripravený na príjem jednej zo štyroch riadiacich správ:

- nastavenie šablóny,
- nastavenie filtra,
- pozastavenie posielania dát,
- obnovenie posielania dát.

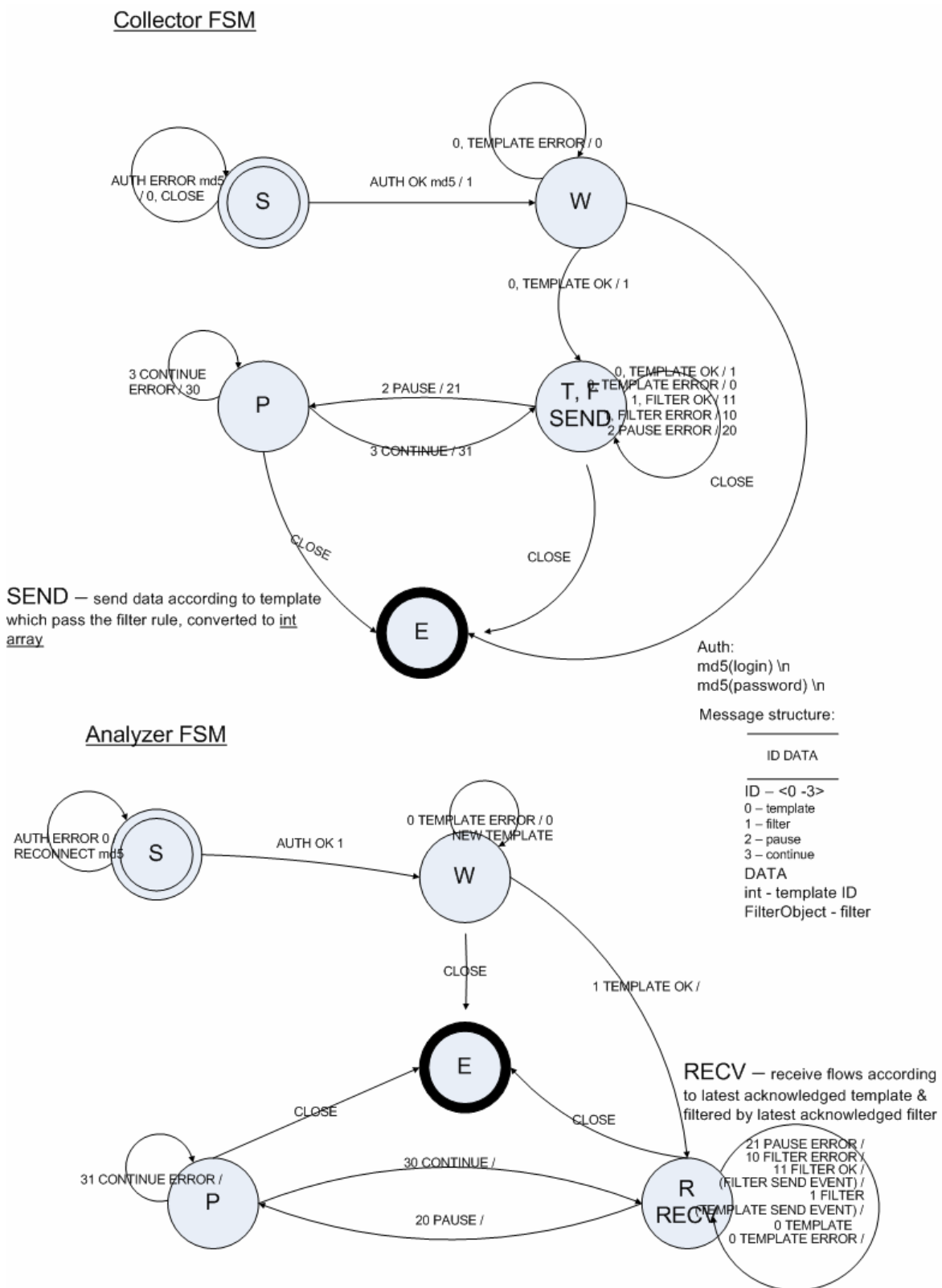
Správa určená pre nastavenie šablóny obsahuje ID šablóny, podľa ktorej má kolektor posielajú dáta. Tieto šablóny sú zatiaľ v programe napevno zadefinované, v budúcnosti sa však počíta priamo so zasielaním binárneho popisu šablóny. Po prijatí šablóny kolektor odpovie jej akceptovaním/zamietnutím. V prípade, že kolektor šablónu akceptoval, začne okamžite posielajú dáta vo formáte špecifikovanom touto šablónou.

Šablónu je možné kedykoľvek počas spojenia zmeniť a ak kolektor akceptuje novú šablónu, začne hneď po jej potvrdení posielat' dáta vo formáte tejto zmenenej šablóny. Šablóna posielaaná kolektoru je zvolená, resp. zostavená programom automaticky podľa typu merania. Súčasné možnosti programu umožňujú príjem a spracovanie týchto atribútov:

- IP adresa meracieho bodu,
- zdrojová a cieľová IP adresa pre daný tok,
- zdrojový a cieľový port,
- protokol,
- čas vzniku a zániku toku,
- počet prenesených bajtov a paketov pre daný tok.

Ďalšia správa, ktorá slúži na nastavenie filtra, určuje kolektoru, ktoré dáta má posielat' a ktoré nie. Pre jedno takéto spojenie musí byť nastavená práve jedna šablóna a môže, ale nemusí, byť špecifikovaný filter. V prípade, že nie je nastavený žiadny filter, kolektor posielat' všetky údaje. Filter je zostavený podľa filtračných kritérií nastavených používateľom. Program v súčasnej verzii umožňuje filtrovat' a kategorizovat' toky na základe nasledujúcich kritérií:

- IP adresa meracieho bodu – zatiaľ je možné vybrať merací bod zo zoznamu načítaného z databázy, alebo je možné zvoliť adresu 0.0.0.0/0, ktorá reprezentuje ľubovoľný merací bod;
- Zdrojová a cieľová IP adresa/maska – kombináciou IP adresy a sieťovej masky je možné dosiahnuť špecifikovanie nielen jedného konkrétneho hostiteľa, ale aj celú podsieť. Vzhľadom na to, že je možné zadať kombináciu viacerých IP adries, je možné jedným pravidlom špecifikovat' aj viacero podsietí, prípadne hostiteľov;
- Zdrojový a cieľový port – toto kritérium umožňuje špecifikovat' konkrétny typ prevádzky na základe čísla portu transportnej vrstvy, je tiež možné zadať viacero hodnôt, rozsah hodnôt ako aj viacero rozsahov hodnôt;
- Protokol – jedná sa o hodnotu nesenú v hlavičke IP paketu, ktorá označuje typ protokolu, zatiaľ je podporované rozoznávajúce podľa čísla a rovnako ako v prípade portov je možné zadať ľubovoľnú kombináciu hodnôt a rozsahov hodnôt, neskôr sa plánuje prehľadnejší výber protokolu zo zoznamu.



**Obr. 7.2: Komunikačný protokol medzi kolektorom a analyzerm.**

Pri návrhu nástroja bol kladený dôraz na maximálnu efektivitu komunikácie a vyhodnocovania filtračných kritérií. Platí pravidlo, že s jedným kolektorom je vytvárané len jedno spojenie, aj keď je realizovaných viac druhov meraní naraz. Po zadaní filtračných kritérií užívateľom dôjde ešte pred samotným vytvorením spojenia k ich optimalizácii nasledujúcim spôsobom:

- Zlučujú sa IP adresy jednotlivých hostiteľov alebo podsietí (supernetting), napr. pravidlá pre podsiete 192.168.0.0/24 a 192.168.1.0/24 sa dajú zlúčiť do jedného pravidla 192.168.0.0/23, podobne IP adresy hostiteľov 192.168.0.2 a 192.168.0.3 sa dajú zlúčiť do pravidla 192.168.0.2/31. Toto zlučovanie je možné realizovať, pretože pri vyhodnocovaní pravidiel sa testuje, či sa určitá IP adresa zhoduje s tou, ktorá je definovaná vo filtri, prípadne, či patrí do podsiete, ktorá je vo filtri definovaná. V súvislosti s prvým uvedeným príkladom napr. platí, že ak IP adresa patrí do podsiete 192.168.0.0/23, zaručene bude patriť do jednej z podsietí 192.168.0.0/24 a 192.168.1.0/24.
- Zlučujú sa aj čísla resp. rozsahy portov a protokolov, napr. ak je uvedené pravidlo reprezentujúce sledovanie požadovaných portov zadané v tvare 20,21,22,23,25, 10-19,80, potom dôjde k jeho úprave na tvar 10-23,25,80.

Ako bolo naznačené, v určitých prípadoch je možné takýmto spôsobom výrazne redukovať počet pravidiel, čo má za následok znížený počet testov pri overovaní voči jednotlivým filtračným kritériám. Pre vyhodnocovanie logicky platí, že ak dôjde k zhode s jedným pravidlom, ďalšie sa už netestujú. Napr. ak dorazí záznam toku s cieľovým portom 80, je nutné pre uvedený príklad otestovať postupne všetky tri podmienky, kým sa zistí, že tok vyhovuje podmienke. Ak však číslo portu bude 22, stačí otestovať prvú podmienku. Rýchlosť vyhodnocovania je teda nepriamo ovplyvnená aj poradím zadaných filtračných kritérií v súvislosti s typom monitorovanej prevádzky. Poradie filtračných kritérií je pri optimalizácii vždy zachovávané. Rovnaký princíp platí pre pravidlá týkajúce sa IP adries. K takémuto zlučovaniu dochádza nielen pre jednotlivé pravidlá filtra, ale dochádza tiež ku spájaniu jednotlivých pravidiel.

Takto optimalizovaný filter je potom poslaný kolektoru. Ak sa počas behu programu pridávajú nové filtračné pravidlá, optimalizačný proces sa zakaždým opakuje a kolektor dostáva vždy nový – prepočítaný filter. Pre vyhodnotenie dát z jedného

kolektora sa s daným kolektorom vytvorí vždy len jedno spojenie, kde je platný jeden (prípadne žiadny) filter. Týmto spôsobom dochádza k rozloženiu záťaže, ktorú prináša proces vyhodnocovania, medzi kolektor a analyzer. Hlavným dôvodom pre takéto rozhodnutie bolo posúdenie faktu, že kolektor je zaťažovaný parsovaním záznamov tokov, ktoré získava od jedného alebo viacerých exporterov a pri zníženom počte filtračných pravidiel, ktoré musí overovať je schopný prijímať a spracovávať viac informácií.

Nevýhodou tohto riešenia je, že pri tomto spôsobe dostáva analyzer potrebné údaje spoločne v jednom prúde dát a preto je potrebné tieto dáta ďalej analyzovať. Program si uchováva zoznam pôvodných optimalizovaných filtrov pred ich zlúčením a na základe toho potom aplikovaním týchto filtrov na záznamy tokov získané od kolektora je schopný ich správne zaradzovať pre konkrétne „spustené“ merania.

Po zvážení situácie sa môže načrtnúť otázka, prečo vlastne dochádza k filtrovaniu na strane kolektora, keď sa tento proces musí aj tak opakovať pre všetky dáta na strane analyzera. Dôvodom je zbytočné zaťaženie siete preposielaním všetkých dát prijatých kolektorom. Predstavme si modelovú situáciu, že kolektor prijíma záznamy tokov od piatich exporterov a nás zaujíma meranie počtu prenesených paketov separátne z dvoch rôznych podsietí. Záznamy tokov z týchto podsietí budú pravdepodobne predstavovať len nepatrný zlomok všetkých záznamov zachytených kolektorom. Nemá preto význam preposielať analyzeru všetky zozbierané dáta. Vytvoria sa filtre pre jednotlivé podsiete, zlúčia sa, a výsledný filter sa pošle kolektorovi. Následne od kolektora dostávame už len záznamy týkajúce sa týchto dvoch podsietí, ktoré si analyzer ešte ďalej podľa potreby separuje na základe čiastkových filtrov, v našej situácii pre jednu a druhú podsieť.

Posledné dva typy riadiacich správ, ktoré analyzer môže poslať kolektorovi umožňujú pozastavenie, resp. opätovné spustenie posielania údajov.

Vzhľadom na možnosť kontinuálneho monitorovania rôznych parametrov, ktoré môžu byť navyše špecifikované filtrami, môže byť náročné, ak nie nemožné sledovanie týchto meraní jednou osobou, napríklad sieťovým administrátorom. Pre takýto účel sa žiada nejaký mechanizmus, ktorý by bol schopný detekovať z týchto meraní určité špecifické situácie a nejakým spôsobom ich zaznamenávať, resp. na ne upozorňovať. BM Analyzer má zabudovaný mechanizmus takzvaných udalostí (events), ktoré plnia presne túto funkciu. Používateľ si prostredníctvom boolovského výrazu zadefinuje podmienky vzniku takejto udalosti ako aj reakciu na túto udalosť. Výskyt takýchto

---

udalostí sa potom overuje v určitých pravidelných časových intervaloch. Príklad takéhoto výrazu:  $\%V > 1M$ , kde  $\%V$  predstavuje premennú, ktorá reprezentuje aktuálnu hodnotu sledovaného parametra,  $1M$  je číselná hodnota zodpovedajúca jednému megabajtu. Na parsovanie boolovských výrazov bol použitý nástroj JavaCC.

Predstavme si nasledujúcu modelovú situáciu. Sieťový administrátor firmy má sťažnosť na malú priepustnosť siete, pričom má podozrenie, že určití užívatelia používajú podnikovú sieť v pracovnom čase na sťahovanie rôzneho obsahu prostredníctvom peer-to-peer klientov typu Kazaa, DirecConnect a iných. Túto situáciu si môže pomerne jednoducho overiť. Zadefinuje si nový typ merania na sledovanie šírky prenosového pásma, vytvorí si filter pre zachytávanie celej prevádzky v podozrivej podsieti, ako aj filter pre túto podsieť, ktorý sleduje len čísla portov, na ktorých pracujú programy peer-to-peer. Pre druhý spomínaný filter si nadefinuje napríklad takúto udalosť:  $(\%V > 500k) \text{ and } (\%dw > 0) \text{ and } (\%dw < 6) \text{ and } (\%dh > 8) \text{ and } (\%dh < 16)$ , ktorá sa vyvolá zakaždým, ak počet prenesených bajtov stúpne nad 500kB, ostatné kritériá určujú pracovné dni a pracovnú dobu. K tejto udalosti si nadefinuje ako reakciu zápis do logovacieho súboru, kde je možné špecifikovať formát zápisu napr. takýmto reťazcom:  $(\%dd.\%dm.\%dy-\%dh:\%dM:\%ds) \ \%a:\%p->\%A:\%P \ [%V]$ , ktorý uloží údaje o dátume výskytu udalosti, zdrojovej a cieľovej IP adrese a porte, a množstve prenesených bajtov. Ak bude takéto meranie spustené týždeň, je možné získať celkom prehľadný záznam o tom, kto a kedy použil príslušnú službu (program), a množstvo prenesených dát. Jedinou podmienkou pre úspešnosť takéhoto merania je správne nastavenie exportovacích časov záznamov jednotlivých tokov, čo však už spadá do oblasti konfigurácie exportera.

Ďalšou možnosťou aplikácie je vyhodnocovanie nameraných dát uložených v databáze. Na prístup do databázy je použité rozhranie JDBC (Java DataBase Connectivity), pričom sú podporované databázové systémy PostgreSQL a MySQL (oba systémy sú voľne dostupné). Pri porovnávaní uvedených databázových systémov ponúka PostgreSQL jednu veľkú výhodu, ktorou je natívna podpora dátového typu Inet, ktorý je určený na uchovávanie IP adries [43]. Je podporovaný protokol IPv4 aj IPv6. Na prácu s týmito adresami existuje sada funkcií a operátorov, pomocou ktorých je možné zistiť zhodnosť IP adries, prípadne príslušnosť IP adresy do siete, a iné. Sú obsiahnuté všetky funkcie potrebné pre náš účel. Zaujímavosťou je aj podpora MAC

adres. MySQL žiadny podobný typ neposkytuje, preto nie je pre naše účely vhodná. Tento nedostatok by sa dal odstrániť navrhnutím vlastných funkcií pre prácu s IP adresami, pričom na ich uloženie by sa použil vhodný celočíselný dátový typ. Na druhej strane by to mohlo priniesť výrazné spomalenie vyhodnocovania dotazov.

Z hľadiska optimalizácie prístupu do databázy je treba zvážiť niekoľko aspektov. Pri vyhodnocovaní výsledkov z nameraných údajov závisí od konkrétneho merania, aké údaje potrebujeme preniesť z databázy ku klientovi, čo v prípade vzdialeného prístupu k databáze (mimo lokálnej siete) môže hrať veľmi dôležitú úlohu. Ideálnym riešením by bolo, ak by mohol byť celý výpočet realizovaný na strane databázového servera a aplikácii by boli odoslané už len údaje reprezentujúce výsledok. Typickým príkladom môže byť situácia, keď nás zaujíma využitie šírky pásma za posledný týždeň, s priemernými hodnotami za jednu hodinu. Výsledná krivka grafu bude mať 168 krokov na časovej osi, čo pri použití 8-bajtového typu pre jednotlivé hodnoty zodpovedá 1344 bajtom. Ako je vidieť, z hľadiska objemu dát sa jedná o zanedbateľnú hodnotu, ktorú nie je v krátkom čase problém preniesť aj cez veľmi pomalú linku (napr. aj pre vytáčané pripojenie s rýchlosťou 28.8kbps by bol čas prenosu kratší ako jedna sekunda). Je teda zrejmé, že prenos údajov pre prakticky ľubovoľný časový interval by bol aj pri takejto pomalej prenosovej linke v extrémnych prípadoch záležitosťou rádovo niekoľkých sekúnd. Aby bolo možné zvoliť takýto prístup, musí databázový systém podporovať takzvané storované procedúry – programovateľné časti kódu, ktoré sa spúšťajú na strane databázového servera. Databáza PostgreSQL takúto možnosť ponúka a podporuje aj niekoľko jazykov, v ktorých je možné takéto funkcie programovať [62]. Jedným z týchto jazykov je procedurálny jazyk PL/SQL, pomocou ktorého bola naprogramovaná funkcia, ktorá pracuje nasledovným spôsobom: Klientská aplikácia (v našom prípade analyzer) predá tejto funkcii sformulovaný dotaz pre databázu, ktorý obsahuje filtračné kritériá zadané používateľom a na základe výsledku, ktorý databáza pre tento dotaz vráti, vyráta šírku pásma alebo priepustnosť paketov a klientskej aplikácii odošle už len hodnoty týchto metrík pre jednotlivé čiastkové intervaly. Tieto hodnoty sú priamo vstupnými dátami pre vykreslenie krivky v grafe. Pri testovaní tohto riešenia sme však narazili na zanedbateľný problém, a tým je slabá výkonnosť takéhoto riešenia, ktorá je pravdepodobne spôsobená buď slabou výkonnosťou databázy pri vyhodnocovaní storovaných procedúr, alebo len slabou

výkonnosťou konkrétneho jazyka PL/SQL. Riešením by mohla byť voľba iného jazyka, resp. zaradením nejakej middle ware aplikácie medzi databázu a klienta, ktorá by bola umiestnená na stroji s databázovým serverom, a klientská aplikácia by k údajom v databáze pristupovala prostredníctvom nej. Momentálne sa však od daného riešenia ustúpilo a všetok výpočet je riešený na strane aplikácie. Aplikácia teda pošle dotaz definujúci požadovanú prevádzku vrátane filtračných kritérií zadaných používateľom a databáza pošle všetky tieto údaje naspäť klientskej aplikácii, ktorá ich následne spracuje a výsledok zobrazí v grafickej podobe. A neberieme do úvahy časovú náročnosť, z užívateľského hľadiska je tento proces transparentný. Pre porovnanie oboch riešení uvediem, že pri použití hardvérovo výkonnejšieho stroja pre databázový server v porovnaní s počítačom, na ktorom bola spustená klientská aplikácia, bol v prostredí 100Mbit-ovej siete výsledok vypočítaný na strane klienta niekoľkokrát rýchlejšie, ako v prípade použitia storovanej procedúry. Nevýhodou súčasného riešenia je nepoužiteľnosť pri pomalých prenosových linkách, pretože kým v prípade použitia storovanej procedúry má výsledok prenášaný po sieti rádovo maximálne niekoľko kB, v druhom prípade sa teoreticky môže jednať rádovo o stovky až tisíce MB. Takéto riešenie je teda predurčené výhradne na použitie v lokálnej sieti.

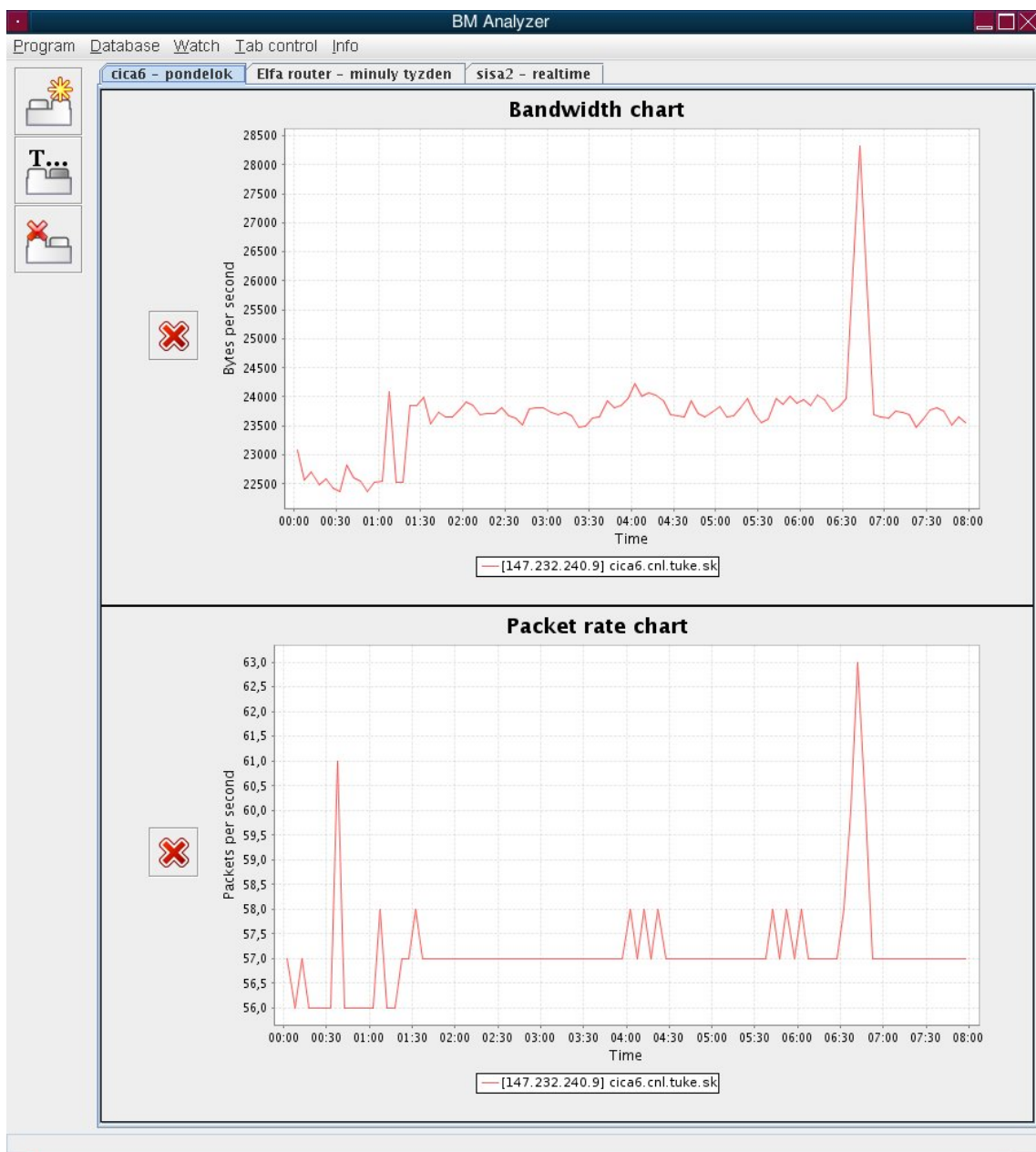
Celkové zvýšenie výkonu by mohlo byť dosiahnuté hlavne zvýšením výkonu diskového systému a optimalizáciou indexovania v databáze. Ďalšou zaujímavou voľbou by mohlo byť použitie MainMemory databázy, prípadne kombinácia riešenia klasickej a MainMemory databázy. Uvedené riešenia sú navrhnuté len z teoretického hľadiska, pretože v čase testovania boli za hranicou našich hardvérových možností.

### **7.3.2 Možnosti aplikácie, používateľské prostredie**

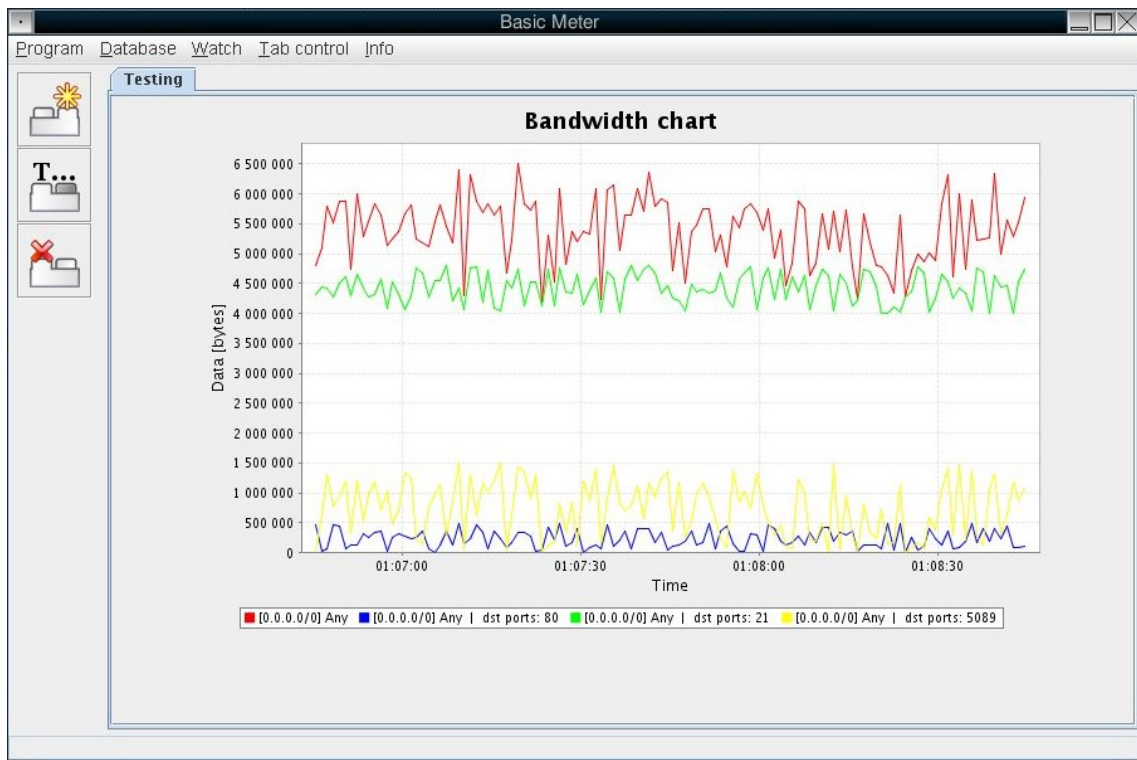
V súčasnom stave je možné kontinuálne vyhodnocovanie šírky pásma a priepustnosti paketov. Taktiež je možné tieto parametre vyhodnocovať spätne z údajov uložených v databáze. Program je schopný pripojiť sa súčasne na viacero databázových serverov a viacero kolektorov, z ktorých je schopný zobrazovať uvedené parametre. Ďalšie obmedzenia, ako je počet súčasne realizovaných meraní alebo maximálny počet pripojených kolektorov sú dané hardvérovými parametrami použitého počítača.

Používateľské prostredie bolo navrhované s ohľadom na fakt, že v praxi bude potrebné mať naraz spustených viacero meraní (reprezentovaných grafmi). Bol preto

zvolený systém „ukladania“ grafov na záložky, pričom jednotlivé záložky je možné pomenovať a na každú záložku je možné uložiť viacero grafov (meraní). Týmto spôsobom je možné jednotlivé merania prehľadne zoskupovať. Situáciu ilustruje Obr. 7.3. V jednom grafe je pritom možné definovať viacero filtrov, kde každý filter je reprezentovaný jednou krivkou (Obr. 7.4). Ďalším dôležitým prvkom používateľského rozhrania je dialógové okno určené pre definíciu jednotlivých meraní a špecifikáciu filtračných kritérií, ktoré je vyobrazené na Obr. 7.5.



**Obr. 7.3:** BM Analyzer – ukážka umiestnenia viacerých meraní na jednu záložku.



Obr. 7.4: BM Analyzer – ukážka použitia viacerých filtrov v jednom grafe.

The screenshot shows the 'Bandwidth' dialog box. It is divided into several sections: 'Database' with a dropdown menu showing '1 bm@localhost:5432/bm'; 'Measurement point' with a dropdown menu showing '[147.232.240.9] cica6.cnl.tuke.sk'; 'Measure type' with a list containing 'Total bytes' (selected) and 'Total packets'; 'Filter' section with fields for 'Source', 'Destination', and 'Protocols', and a 'Ports' field set to '21'; 'Time period' section with 'Start' (25.4.2005 00:00:00) and 'End' (25.4.2005 08:00:00) time pickers, and 'Time interval' (1) and 'Time step' (5) fields, both with 'minutes' units. At the bottom, there is a table with two rows: '[147.232.240.9] cica6.cnl.tuke.sk' and '[147.232.240.9] cica6.cnl.tuke.sk | src ports: 80'. To the right of the table are 'Add' and 'Remove' buttons. At the very bottom are 'OK' and 'Cancel' buttons.

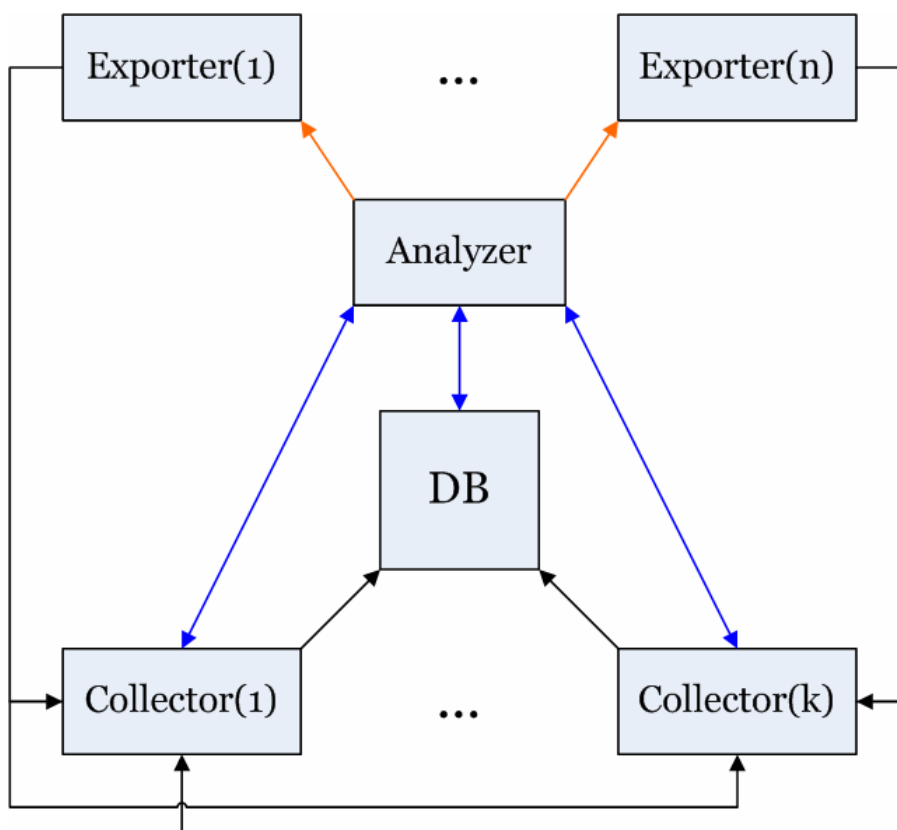
Obr. 7.5: BM Analyzer – dialógové okno na definovanie nových meraní.

## 7.4 Ďalší vývoj nástroja BasicMeter

Aby meracia platforma zodpovedala architektúre znázornenej na Obr. 7.6, sú potrebné nasledujúce rozšírenia:

- rozšírenie exportéra o podporu exportu na viacero kolektorov súčasne,
- implementácia komunikačného protokolu medzi exporterom a analyzerom,
- prípadné rozšírenie komunikačného protokolu medzi kolektorom a analyzerom,
- rozšírenie analyzera o funkcie na prácu s databázou v zmysle správy prístupu k nazbieraným údajom,
- implementácia bezpečnostných prvkov v komunikácii (použitie protokolu IPsec).

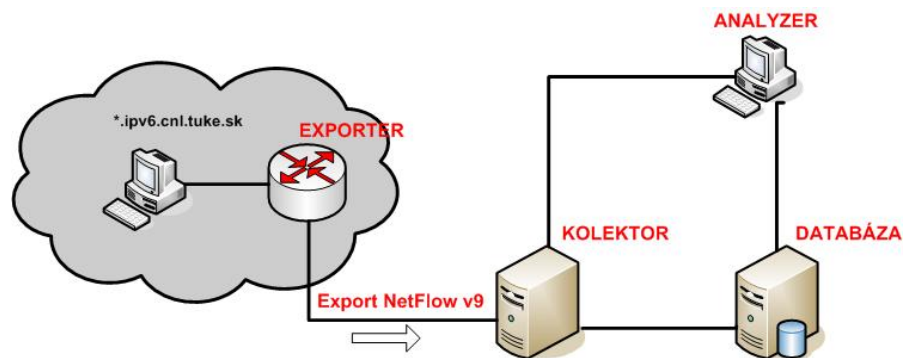
Spomenuté rozšírenia by umožnili kontrolu exportného procesu priamo z analyzujúcej aplikácie, ktorá by v takomto prípade vystupovala aj v úlohe centrálného riadiaceho bodu celej meracej platformy. Ďalej by bolo možné riadiacou aplikáciou manažovať prístupové práva k nazbieraným údajom a spravovať autentifikačné a autorizačné dáta pre exporter a kolektor.



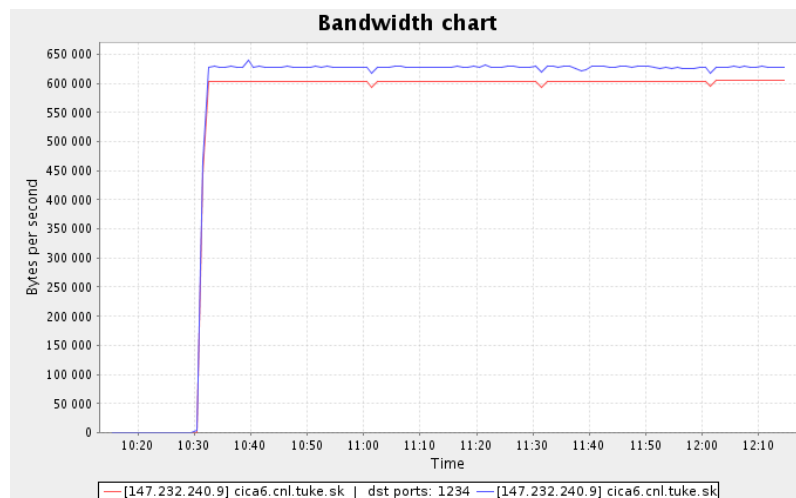
Obr. 7.6: Plánované rozšírenie meracej platformy BasicMeter.

## 8. Experimentálne merania

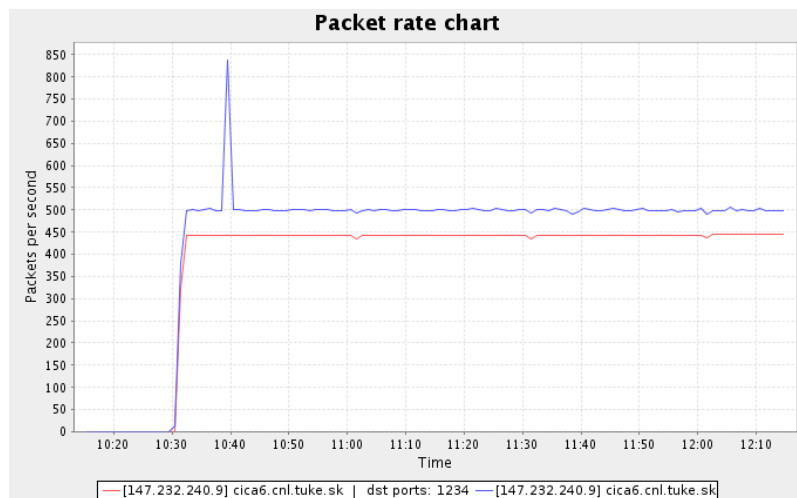
Za účelom overenia funkčnosti nielen analyzujúcej aplikácie, ale aj celej meracej platformy boli realizované dva druhy experimentálnych meraní. V prvom prípade išlo o monitorovanie prevádzky na IPv6 segmente v laboratóriu počítačových sietí. V tomto prípade bol ako exportér použitý Cisco router. Topológia zapojenia je znázornená na Obr. 8.1. V čase merania bolo popri bežnej sieťovej prevádzke realizovaných niekoľko typov dátových prenosov – video a audio streaming a prenos dát prostredníctvom protokolu FTP. Namerané výsledky sú znázornené na Obr. 8.2 až 8.5. Výsledky týchto meraní boli vyhodnocované z databázy. Vzhľadom na možnosti analyzujúcej aplikácie boli merané objemové charakteristiky QoS, teda šírka pásma a priepustnosť paketov.



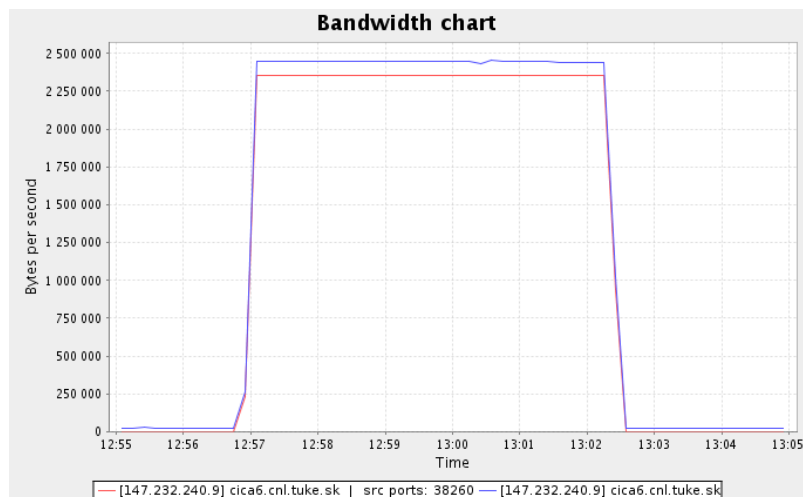
Obr. 8.1: Topológia zapojenia BasicMetra pri experimente na segmente IPv6.



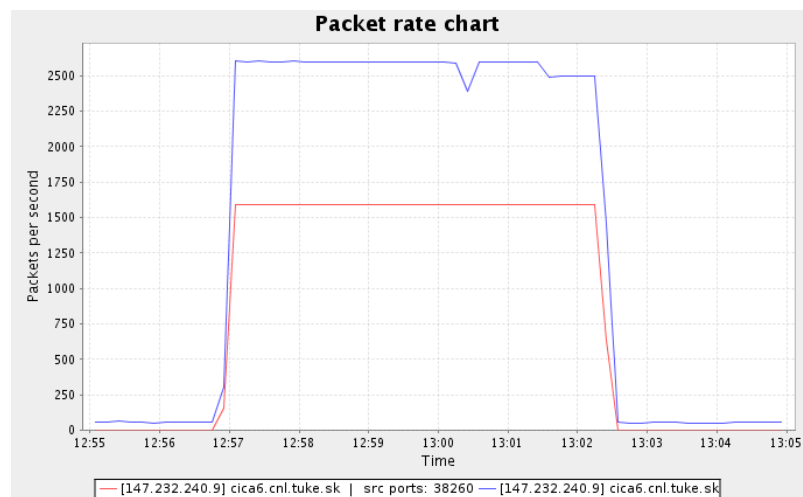
Obr. 8.2: Šírka pásma – video streaming a celková prevádzka.



Obr. 8.3: Priepustnosť paketov – video streaming a celková prevádzka.

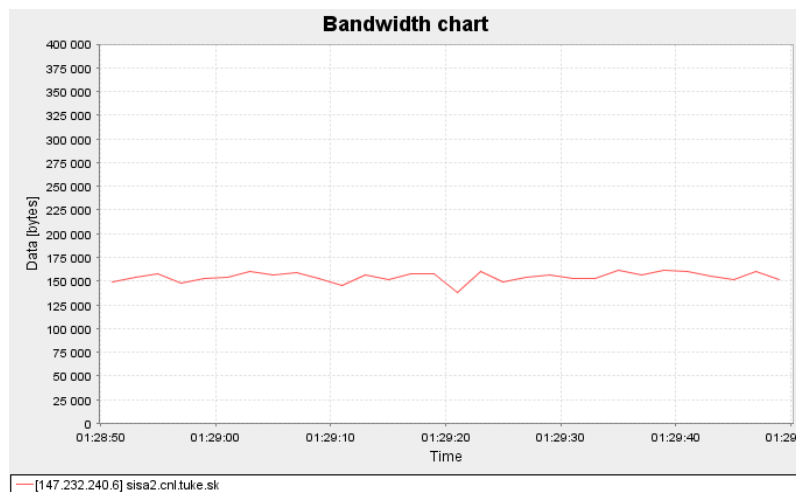


Obr. 8.4: Šírka pásma – ftp prenos a celková prevádzka.

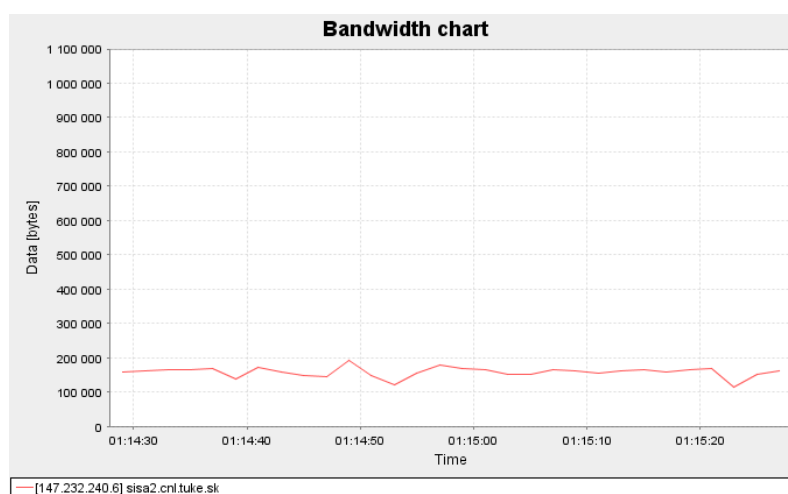


Obr. 8.5: Priepustnosť paketov – ftp prenos a celková prevádzka.

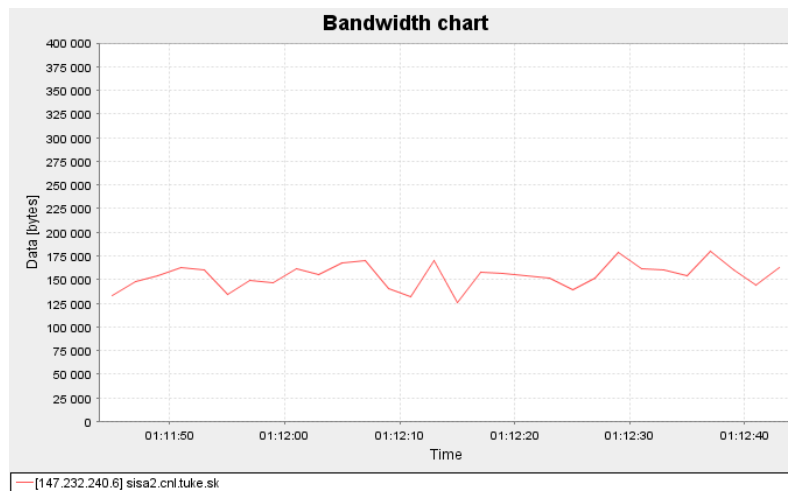
Ďalšie merania boli realizované za účelom overenia funkčnosti implementácie vzorkovacích algoritmov v exporteri. Tieto merania boli overované v reálnom čase, čím bola zároveň overená funkčnosť komunikácie analyzera s kolektorom a schopnosť aplikácie dynamicky analyzovať prevádzku sietí. Konfigurácia meraní zahŕňala nastavenie doby exportovania údajov v časových intervaloch jednej sekundy, pričom na strane analyzera boli údaje vyhodnocované v dvojsekundových intervaloch, čím sa čiastočne vykompenzovalo oneskorenie, ktoré je dané dobou prenosu údajov v sieti a ich spracovaním v jednotlivých prvkoch meracej platformy. Parametre vzorkovacích algoritmov boli volené tak, aby cieľová vzorka tvorila 20 % z celkovej populácie. Úspešnosť vzorkovacích algoritmov je vidieť z porovnania Obr. 8.6 – 8.9 s Obr. 8.10.



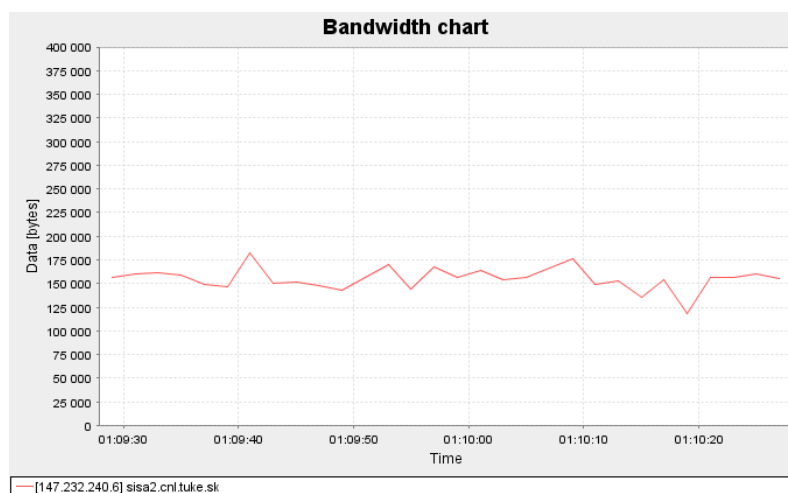
**Obr. 8.6: Systematické vzorkovanie založené na počte.**



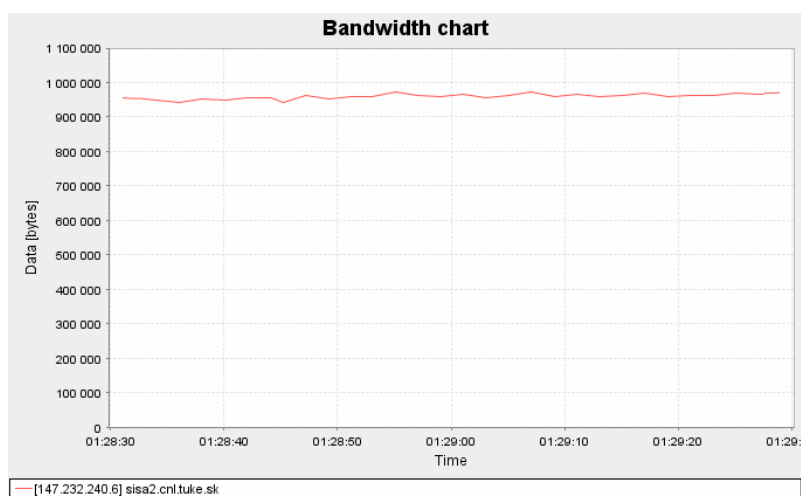
**Obr. 8.7: Vzorkovanie  $n$  z  $N$ .**



Obr. 8.8: Uniformné náhodné pravdepodobnostné vzorkovanie.



Obr. 8.9: Neuniformné náhodné pravdepodobnostné vzorkovanie založené na počte.



Obr. 8.10: Tok celkovej populácie.

## 9. Zhodnotenie riešenia

Táto práca sa zaoberá vyhodnocovaním parametrov QoS s využitím pasívnych metód. Pre podobné účely bol vyvinutý štandard IPFIX, ktorý špecifikuje architektúru meracej platformy použiteľnej pre tieto účely. Hlavným cieľom bolo priblížiť tento štandard ako aj jeho existujúce implementácie a navrhnúť nástroj na vyhodnocovanie meraných údajov zozbieraných pomocou protokolu NetFlow9, ktorý predstavuje najrozšírenejšiu implementáciu štandardu IPFIX.

Vytvorená aplikácia je súčasťou komplexnej meracej platformy BasicMeter, ktorá si kladie za cieľ splniť požiadavky definované štandardom IPFIX. Aj keď samotná analyzujúca aplikácia nie je súčasťou architektúry tohto štandardu, spolu s programami určenými na export a zber informácií tvorí plnohodnotný merací nástroj. Pri návrhu riešenia boli zohľadnené požiadavky na výkonnosť riešenia, čo malo za následok vznik jednoduchého aplikačného protokolu na komunikáciu medzi prvkom určeným na zber a prvkom určeným na analýzu údajov. Pri samotnej komunikácii takto dochádza k rozloženiu záťaže medzi tieto dva prvky, čo v konečnom dôsledku umožňuje spracovávanie a vyhodnocovanie väčšieho množstva informácií. Aplikácia je v súčasnom stave schopná analyzovať objemové prevádzkové parametre QoS – šírku pásma a priepustnosť paketov, pričom je možné kategorizovať toky na základe rôznych parametrov (IP adresa, port, protokol).

Z hľadiska budúcnosti je nevyhnutná implementácia protokolu IPv6 minimálne na úrovni rozlišovania tokov. Na druhej strane použitie tohto protokolu na samotný prenos exportovaných dát by vyriešilo niektoré otázky týkajúce sa bezpečnosti, pre ktoré sa inak musí použiť dodatočné zabezpečenie. Ďalším cieľom je dopracovanie podpory vyhodnocovania udalostí na základe zadefinovaných parametrov, ktoré môžu z aplikácie vytvoriť čiastočne inteligentný nástroj monitorovania sieťovej prevádzky. Táto funkcia by si našla svoje uplatnenie nielen pri detekcii výskytu neobvyklých udalostí v sieťovej prevádzke, ale najmä pri rôznych metódach účtovania služieb.

## 10. Zoznam použitej literatúry

- [1] Sučík, J.: Príspevok k problematike merania a vyhodnocovania parametrov kvality služieb (QoS) v počítačových sieťach. Diplomová práca (vedúci Ing. F. Jakab), Košice: KPI FEI TU, 2003
- [2] André, M.: Meranie a vyhodnocovanie prevádzkových parametrov v počítačových sieťach. Diplomová práca (vedúci Ing. F. Jakab), KPI FEI TU, Košice, 2004
- [3] NetFlow Services and Applications, White Paper, Cisco Systems Inc., 1999.  
[online] [http://www.cisco.com/warp/public/cc/pd/iosw/ioft/neflet/tech/napps\\_wp.htm](http://www.cisco.com/warp/public/cc/pd/iosw/ioft/neflet/tech/napps_wp.htm)
- [4] Quittek, J. – Zseby, T. – Claise, B. et al.: Requirements for IP Flow Information Export. RFC 3917, October 2004
- [5] Sadasivan, G. – Brownlee, N. – Claise, B. et al.: “Architecture Model for IP Flow Information Export”. Internet draft draft-ietf-ipfix-architecture-07.txt, work in progress, March 2005
- [6] Brownlee, N. – Mills, C. – Ruth, G.: Traffic Flow Measurement: Architecture. RFC 2063, January 1997
- [7] Laine, J.; Saaristo, S.; Prior, R.: “RUDE & CRUDE: Real-time UDP Data Emitter and Collector”, [online] <http://rude.sourceforge.net>.
- [8] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981
- [9] Shenker, S. – Partridge, C. – Guerin, R.: Specification of Guaranteed Quality of Service. RFC 2212, September 1997
- [10] Claise, B.: Cisco Systems NetFlow Services Export Version 9. RFC 3954, October 2004
- [11] Shenker, S., et al.: Specification of Guaranteed Quality of Service. RFC 2212, September 1997
- [12] Ubik, S. – Smotlacha, V. et al.: Low-cost Precise QoS Measurement Tool, technická zpráva Cesnet 7/2001, [online]  
<http://www.cesnet.cz/doc/techzpravy/2001/07>
- [13] Internetworking Technologies Handbook – QoS Management, Cisco Systems Inc.  
[online] [http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito\\_doc/qos.htm](http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/qos.htm)

- [14] Kohler, P. – Claise, B.: IPFIX fine-tunes traffic analysis. Network World, 08/11/03. [online]  
<http://www.networkworld.com/news/tech/2003/0811techupdate.html>
- [15] Zseby, T. – Boschi, E. – Brownlee, N.: IPFIX Applicability. Internet draft draft-ietf-ipfix-as-04.txt, work in progress, February 2005
- [16] Quittek, J. – Bryant, S. – Meyer, J.: Information Model for IP Flow Information Export. Internet draft draft-ietf-ipfix-info-06.txt, February 2005
- [17] Claise, B.: IPFIX Protocol Specification. Internet draft draft-ietf-ipfix-protocol-11.txt, March 2004
- [18] Leinen, S.: IP Flow Information Export (IPFIX) Over TCP. Internet draft draft-leinen-ipfix-tcp-01.txt, October 24, 2004
- [19] Schulzrinne, H. – Casner, S. et al.: “RTP: A Transport Protocol for Real-Time Applications“. RFC 1889, January 1996
- [20] Bellovin, S.: Defending Against Sequence Number Attacks. RFC 1948, May 1996
- [21] Kent, S. – Atkinson, R.: IP Authentication Header. RFC 2402, November 1998
- [22] Kent, S. – Atkinson, R.: IP Encapsulating Security Payload (ESP). RFC 2406, November 1998
- [23] Xie, Q. – Sharp, C. – Taylor, T.. et al.: Stream Control Transmission Protocol. RFC 2960, October 2000
- [24] Stewart, R. – Tuexen, M. – Conrad, P. et al.: Stream Control Transmission Protocol (SCTP) Partial Reliability Extension. RFC 3758, May 2004
- [25] Brownlee, N. – Blount, A.: Accounting Attributes and Record Formats. RFC 2924, September 2000
- [26] Mitton, D. – Barkley, S – Nelson, D. et al.: “Authentication, Authorization, and Accounting: Protocol Evaluation“. RFC 3127, June 2001
- [27] Phaal, P et al.: “InMon Corporation's sFlow: A Method for Monitoring Traffic in Switched and Routed Networks“. RFC 3176, September 2001
- [28] Zhang, K. – Elkin, E.: XACCT's Common Reliable Accounting for Network Element (CRANE). RFC 3423, November 2002
- [29] Calhoun, P. – Loughney, J. – Guttman, E. et al: Diameter Base Protocol, RFC 3588, September 2003

- [30] Loughney, J: Diameter Command Codes for Third Generation Partnership Project (3GPP) Release 5. RFC 3589, September 2003
- [31] Claise, B et al.: Cisco Systems NetFlow Services Export Version 9. RFC 3954, October 2004
- [32] Leinen, S.: Evaluation of Candidate Protocols for IP Flow Information Export (IPFIX). RFC 3955, October 2004
- [33] [online] <http://www.splintered.net/sw/flow-tools/>
- [34] [online] <http://software.uninett.no/stager/>
- [35] [online] <http://www.mindrot.org/softflowd.html>
- [36] [online] <http://www.mindrot.org/flowd.html>
- [37] [online] <http://www.ntop.org/ntop.html>
- [38] [online] <http://nfdump.sourceforge.net/>
- [39] [online] <http://nfsen.sourceforge.net/>
- [40] [online] <http://manageengine.adventnet.com/products/netflow/>
- [41] [online] <http://www.sflow.org/>
- [42] PostgreSQL 8.0.2 Documentation.  
[online] <http://www.postgresql.org/docs/8.0/static/datatype-net-types.html>
- [43] Crawley, E. – Nair, R. – Rajagopalan, B et al.: A Framework for QoS-based Routing in the Internet. RFC 2386, August 1998
- [44] [online] <http://www.cisco.com/warp/public/732/Tech/nmp/netflow/v9/>
- [45] System Management Guide: Communications and Networks.  
[online] [http://www.unet.univie.ac.at/aix/aixbman/commadmn/tcp\\_qos.htm](http://www.unet.univie.ac.at/aix/aixbman/commadmn/tcp_qos.htm)
- [46] Braden, R. – Clark, D. – Shenker, S: “Integrated Services in the Internet Architecture: an Overview”. RFC 1633, June 1994
- [47] Wroclawski, J: Specification of the Controlled-Load Network Element Service. RFC 2211, September 1997
- [48] Blake, S. – Black, D. – Carlson, M et al.: An Architecture for Differentiated Services. RFC 2475, December 1998
- [49] Nichols, K. – Blake, S. – Baker, F. et al: Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. RFC 2474, December 1998
- [50] ITU-T Recommendation I.380. Internet protocol data communication service - IP packet transfer and availability performance parameters, February 1999.

- [51] Paxson, V. – Almes, G. – Mahdavi et al: Framework for IP Performance Metrics. RFC 2330, May 1998
- [52] Mathis, M. – Allman, M.: A Framework for Defining Empirical Bulk Transfer Capacity Metrics. RFC 3148, July 2001
- [53] Demichelis, C. – Chimento, P: IP Packet Delay Variation Metric for IP Performance Metrics (IPPM). RFC 3393, November 2002
- [54] Heinanen, J – Baker, F – Weiss, W. et al.: Assured Forwarding PHB Group. RFC 2597, June 1999
- [55] Jacobson, V. – Nichols, K. – Poduri, K.: An Expedited Forwarding PHB. RFC 2598, June 1999
- [56] Almes, G. – Klidindi, S. – Zekauskas, M.: A One-way Packet Loss Metric for IPPM. RFC 2680, September 1999
- [57] Almes, G. – Klidindi, S. – Zekauskas, M.: A Round-trip Delay Metric for IPPM. RFC 2681, September 1999
- [58] Mills, D.: Network Time Protocol (v3). RFC 1305, April 1992
- [59] Rosen, E. – Viswanathan, A. – Callon, R.: Multiprotocol Label Switching Architecture. RFC 3031, January 2001
- [60] Dierks, T. – Allen, C: The TLS Protocol Version 1.0. RFC 2246, January 1999
- [61] Thayer, R. – Doraswamy, N. – Glenn, R.: IP Security Document Roadmap. RFC 2411, November 1998
- [62] PostgreSQL 8.0.2 Documentation.  
[online] <http://www.postgresql.org/docs/8.0/static/server-programming.html>

## 11. Zoznam príloh

1. CD médium – diplomová práca v elektronickej podobe, prílohy v elektronickej podobe.
2. Používateľská príručka
3. Systémová príručka

## 12. Zoznam obrázkov a tabuliek

### Zoznam obrázkov

Obr. 3.1: Model architektúry IPFIX .....	14
Obr. 3.2: Schéma IPFIX zariadenia .....	15
Obr. 7.1: Architektúra meracej platformy BasicMeter .....	39
Obr. 7.2: Komunikačný protokol medzi kolektorom a analyzerm .....	43
Obr. 7.3: BM Analyzer – ukážka umiestnenia viacerých meraní na jednu záložku.....	49
Obr. 7.4: BM Analyzer – ukážka použitia viacerých filtrov v jednom grafe .....	50
Obr. 7.5: BM Analyzer – dialógové okno na definovanie nových meraní.....	50
Obr. 7.6: Plánované rozšírenie meracej platformy BasicMeter.....	51
Obr. 8.1: Topológia zapojenia BasicMetra pri experimente na segmente IPv6 .....	52
Obr. 8.2: Šírka pásma – video streaming a celková prevádzka .....	52
Obr. 8.3: Prieupustnosť paketov – video streaming a celková prevádzka.....	53
Obr. 8.4: Šírka pásma – ftp prenos a celková prevádzka.....	53
Obr. 8.5: Prieupustnosť paketov – ftp prenos a celková prevádzka .....	53
Obr. 8.6: Systematické vzorkovanie založené na počte.....	54
Obr. 8.7: Vzorkovanie $n$ z $N$ .....	54
Obr. 8.8: Uniformné náhodné pravdepodobnostné vzorkovanie .....	55
Obr. 8.9: Neuniformné náhodné pravdepodobnostné vzorkovanie založené na počte...	55
Obr. 8.10: Tok celkovej populácie.....	55