VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

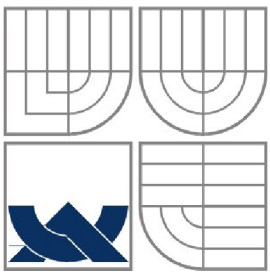
VESTAVĚNÝ TEST SONDY FLOWMON

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

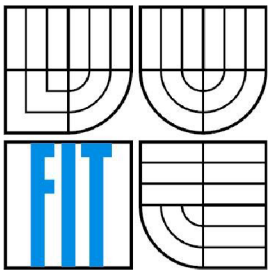
AUTOR PRÁCE
AUTHOR

BLAŽEJ KŘÍŽ

BRNO 2009



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

VESTAVĚNÝ TEST SONDY FLOWMON

SELF TEST OF FLOWMON PROBE

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

BLAŽEJ KŘÍŽ

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. JAN KOŘENEK

BRNO 2009

Abstrakt

Tato práce se zabývá vývojem vestavěného testu sondy FlowMon, což je zařízení pro monitorování síťového provozu na bázi IP toků. V úvodní části jsou shrnuty související pojmy, popsána technologie NetFlow a sonda FlowMon. Samotný vývoj testu pak zahrnuje specifikaci a analýzu požadavků, návrh obecné techniky testování, návrh jednotlivých testů, jejich implementaci a zhodnocení řešení.

Abstract

This thesis deals with development of built-in self-test for FlowMon probe, device for monitoring network traffic based on IP flows. At the begining, both NetFlow technology and the FlowMon probe are described and related terms are summarized. The development itself consists of requirements specification and analysis, design of general testing technique, desing of particular tests, their implementation and solution review.

Klíčová slova

FlowMon, sonda, NetFlow, vestavěný test, pcap, vysokorychlostní síť, monitoring, IP tok

Keywords

FlowMon, probe, NetFlow, built-in self-test, pcap, high-speed network, monitoring, IP flow

Citace

Blažej Kříž: *Vestavěný test sondy FlowMon*. Brno, 2009, bakalářská práce, FIT VUT v Brně.

Vestavěný test sondy FlowMon

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Jana Kořenka. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Blažej Kříž
18. května 2009

Poděkování

Především bych rád poděkoval panu Ing. Janu Kořenkovi za poskytnuté konzultace a odborné vedení práce a dále členům projektu Liberouter za technickou podporu.

Obsah

1 Úvod

Internet a počítačové sítě obecně se v posledních letech stávají běžnou součástí našeho života. Mnoho lidí už považuje přístup k Internetu za stejnou samozřejmost, jako pitnou vodu tekoucí z kohoutku nebo fungující ledničku. Dnešní společnost je na Internetu závislá a sebemenší chyba nebo výpadek znamenají problém. Abychom riziko vzniku těchto problémů minimalizovali, je zapotřebí počítačové sítě spravovat.

Kvalitní správa sítí se neobejde bez nástrojů pro sledování síťového provozu. Mezi nejrozšířenější z nich patří i technologie *NetFlow*, vyvinutá firmou Cisco. Umožňuje podrobné monitorování provozu v celé síti, čímž je užitečná nejen pro administrátory a manažery, kterým poskytuje informace o chování uživatelů, využití kapacity sítě, pokusech o útok na síť atd., ale i pro poskytovatele internetového připojení např. kvůli dodržování vyhlášky o elektronické komunikaci nebo kontrole FUP.

Protože neustále stoupá počet připojených zařízení a zvyšují se i nároky uživatelů na kvalitu a rozsah dostupných služeb, rychle roste objem přenášených dat. Tomuto trendu se musí přizpůsobovat veškerá síťová zařízení.

Tradiční NetFlow architektura, kdy je monitorování provozu prováděno ve směrovači, jako jedna z jeho vedlejších funkcí, získala v posledních letech zajímavou alternativu. Místo směrovače je k monitorování použito specializovaného zařízení – autonomní sondy. Toto řešení přináší řadu výhod. Oproti směrovači nabízí vyšší výkon, což umožňuje detailnější zpracování dat bez nutnosti vzorkování a možnost připojení na jakékoliv místo v síti. Sonda se navíc nestává cílem útoků, protože je neviditelná na síťové vrstvě.

Sonda FlowMon vzniká v rámci projektu Liberouter, který je součástí výzkumného záměru sdružení CESNET. Celý proces monitorování rozděluje mezi programové vybavení počítače (software) a technické vybavení (hardware), přičemž hardware vykonává výpočetně náročné často se opakující úkony, zatímco software zpracovává obecné a měnící se události. Tímto rozdělením je dosaženo vysokého výkonu a použitím programovatelných hradlových polí (FPGA) zajištěna flexibilita díky možnosti jejich rekonfigurace.

Tato práce si klade za cíl vytvoření vestavěného testu sondy, k jehož proběhnutí nebude třeba žádného specializovaného hardwaru nebo softwaru, jehož přítomnost není při spouštění testu zaručena, a který bude schopen ověřit funkčnost sondy pro její použití v běžném provozu.

Kapitola 2 začíná stručnými informacemi o síťové architektuře a principu přenosu dat v počítačové síti. Jsou zde vysvětleny pojmy jako *vrstvový model TCP/IP* nebo *protokol.* Přes popis technologie NetFlow se dostává k samotné sondě FlowMon a pojmu vestavěný test.

Třetí kapitola se zabývá vývojem samotného testu. Důraz je kladen především na první fáze vývoje, specifikaci a analýzu požadavků a z nich vycházející návrh. Je zvolena obecná technika testování a podle ní navrženy konkrétní testy, zaměřené na jednotlivé funkce sondy. Vybrané dílčí testy jsou poté zařazeny do vestavěného testu a implementovány. Pro potřeby testování slouží generátor síťového provozu, taktéž vyvinutý v rámci této práce.

Zhodnocení vyvinutého testu se spolu s vysvětlením důvodů pro výběr použité metody testování nachází v kapitole 4.

Poslední kapitola obsahuje závěrečné shrnutí práce, naznačuje možná rozšíření a směr dalšího vývoje.

2 Teoretický rozbor

2.1 Obecně o sítích

Řízení komunikace mezi síťovými zařízeními je složitý problém, proto se rozděluje do několika vrstev na oddělené a snadněji zvládnutelné menší problémy. Každá vrstva dostává přiřazen svůj úkol, spoléhá se na služby nižší vrstvy a své služby poskytuje vrstvě vyšší. Existuje několik vrstvových modelů, standardem současné sítě Internet je model TCP/IP.

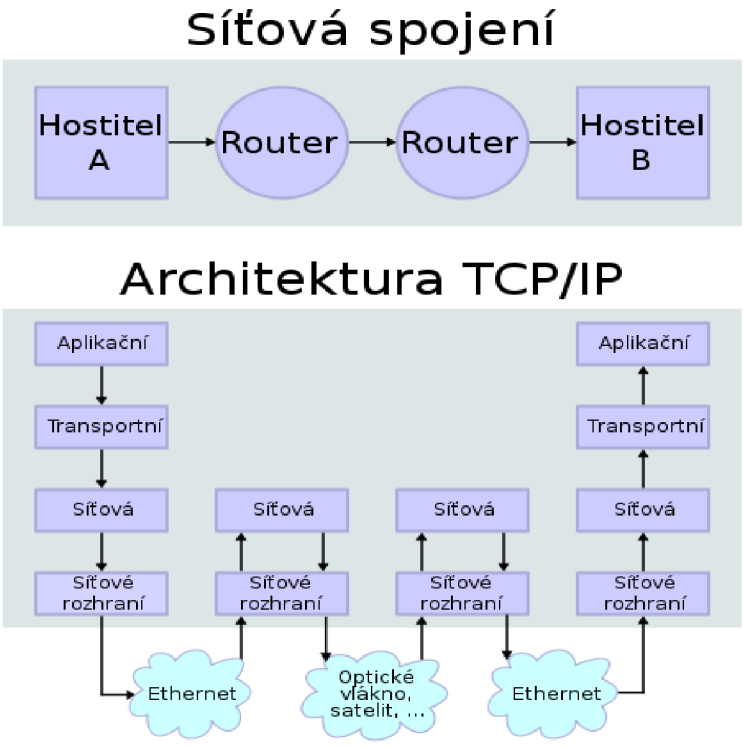
2.1.1 Model TCP/IP

Model TCP/IP se skládá ze čtyř vrstev:

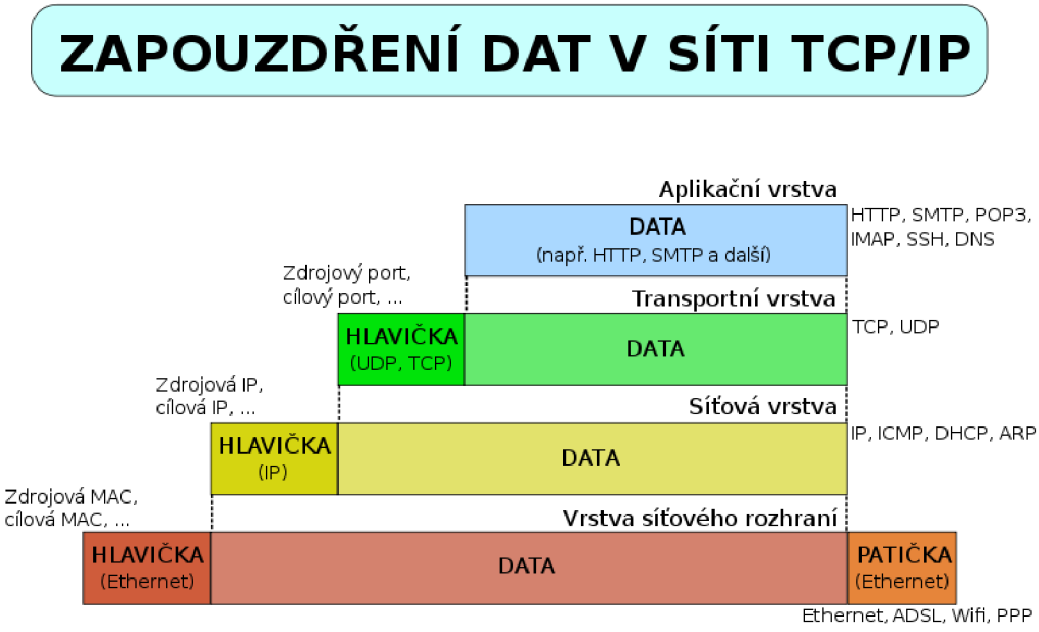
- *Vrstva síťového rozhraní*
  - nejnižší vrstva
  - umožňuje přístup k fyzickému přenosovému médiu
- *Síťová vrstva*
  - zajišťuje adresaci a směrování na místo určení
  - vytváří logické spojení mezi počítači
- *Transportní vrstva*
  - implementována až v koncových zařízeních (počítačích)
  - umožňuje přizpůsobit chování sítě potřebám aplikace
  - poskytuje spojované (spolehlivé) či nespojované (nespolehlivé) transportní služby
  - vytváří logické spojení mezi procesy
- *Aplikační vrstva*
  - tvořena aplikacemi a procesy, které komunikují po síti

Na *obrázku 2.1* je znázorněna komunikace mezi dvěma koncovými zařízeními. Horní část ukazuje strukturu síťového spojení, níže jsou pak vidět jednotlivé vrstvy modelu TCP/IP. *Směrovače (routers)* i koncová zařízení používají síťovou vrstvu k adresaci a směrování, ale jen koncová zařízení potřebují vyšší vrstvy pro odesílání a příjem aplikačních dat.

Při předávání dat z vyšší vrstvy do nižší dochází k jejich *zapouzdření (viz. obrázek 2.2)*. Je to proces, při kterém si vyšší vrstva k předávaným datům přidá své *záhlaví*, obsahující informace potřebné pro realizaci poskytované služby.



Obrázek 2.1: Princip vrstev TCP/IP



Obrázek 2.2: Zapouzdření dat v síti TCP/IP

2.1.2 Protokol

V oblasti počítačových sítí se *protokolem* nazývá norma definující syntaktická a sémantická pravidla, podle kterých probíhá komunikace a přenos dat mezi síťovými zařízeními. Protokol je svázán s vrstvou, na které pracuje. Na *obrázku 2.2* jsou u každé vrstvy vpravo uvedeny příklady jim příslušných protokolů.

Otevřeně specifikované protokoly usnadňují součinnost síťových aplikací a tím pomohly rychlému rozvoji Internetu. Základními protokoly Internetu jsou:

- *Internet Protocol (IP)*
  - základní protokol síťové vrstvy (od tohoto protokolu dostala síť Internet své jméno, původně InterNet Protocol)
  - jednotkou přenosu je *IP-datagram*, který v záhlaví obsahuje mimo jiné i *IP-adresu*, jednoznačně identifikující síťové rozhraní v rámci celého Internetu
  - nenavazuje relaci spojení, každý IP-datagram je doručován samostatně
  - doručení datagramu není zaručeno, spolehlivost musí zajistit protokoly vyšších vrstev
- *Transmission Control Protocol (TCP)*
  - protokol transportní vrstvy zaručující doručení všech dat ve správném pořadí
  - jednotkou přenosu je *paket*, v jehož záhlaví je uloženo číslo *portu* pro rozlišení adresované aplikace
- *User Datagram Protocol (UDP)*
  - protokol transportní vrstvy bez kontroly doručení
  - stejně jako TCP používá porty pro rozlišení aplikací
  - jednodušší a rychlejší než TCP, výhodný v případech, kdy není potřeba spolehlivý přenos (např. multimediální aplikace, IP telefonie, hry...)
- *Ethernet*
  - pracuje na vrstvě síťového rozhraní, nejběžnější protokol pro přenos dat po drátech v lokálních sítích
  - jako přenosové médium využívá koaxiální kabel (spíše dříve), kroucenou dvojlinku nebo optické vlákno
  - jednotkou přenosu je *rámec*
  - záhlaví rámce obsahuje *fyzickou adresu (MAC adresu)* cílového zařízení, kterou každému zařízení přiděluje výrobce, adresa je celosvětově jedinečná

Protokoly aplikační vrstvy umožňují aplikacím vzájemnou síťovou komunikaci. Mezi ty nejznámější patří např. HTTP, POP3, SMTP, FTP nebo DNS. Dle potřeby pak samy využívají protokoly transportní vrstvy TCP nebo UDP.

Z pohledu této práce je z protokolů aplikační vrstvy důležitý *NetFlow*, neboť na něm je postaven export dat z testované sondy FlowMon.

2.2 NetFlow

2.2.1 Technologie

Protokol NetFlow byl vyvinut firmou Cisco jako součást technologie pro monitorování síťového provozu. Slouží k přenosu informací o *IP tocích* ve formě *NetFlow záznamů* z *NetFlow exportérů* do *NetFlow kolektorů*. Z transportní vrstvy využívá protokolu UDP nebo SCTP. Po odeslání z exportéru je kvůli rychlosti a uvolňování paměti záznam zahozen, což při použití nespolehlivého přenosu znamená jeho ztracení.

IP tok (IP flow) je sekvence paketů se společnou vlastností procházejících bodem pozorování za určitý čas. V terminologii NetFlow je tato vlastnost definována jako pětice následujících údajů: cílová/zdrojová IP adresa, cílový/zdrojový port a číslo protokolu.

NetFlow záznam obsahuje informace o IP toku. Jeho položky se liší podle použité verze protokolu. V nejpoužívanější verzi *v5* má pevně danou strukturu, v novější verzi *v9* lze z jednotlivých položek tvořit šablony a ze záznamů se na ně odkazovat. Z verze *v9* vychází i standard IPFIX.

Obrázek 2.3: Princip technologie NetFlow

NetFlow exportér je připojen k monitorované lince. Analyzuje síťový provoz a pořizuje o něm záznamy, které odesílá na kolektor. Může být realizován jako:

- součást směrovače
- autonomní SW sonda
- autonomní HW sonda

Tradiční architektura s exportérem umístěným přímo ve směrovači trpí několika nedostatky. Nižší výkon, o který se navíc analýza musí podělit se směrováním má za důsledek menší přesnost statistik, viditelnost na síťové vrstvě z nich dělá cíle útoků a zdaleka ne každý směrovač podporuje NetFlow export. Autonomní sondy se snaží tyto nedostatky odstranit.

NetFlow kolektor přijímá záznamy z exportérů a ukládá je do databáze nebo souboru. Nad databází (souborem) pracuje aplikace zobrazující data v uživatelsky přívětivé podobě tabulek, grafů atd. Kolektorem bývá počítač, připojený k exportéru dedikovanou linkou. Také může být spolu se sondou součástí jednoho zařízení.

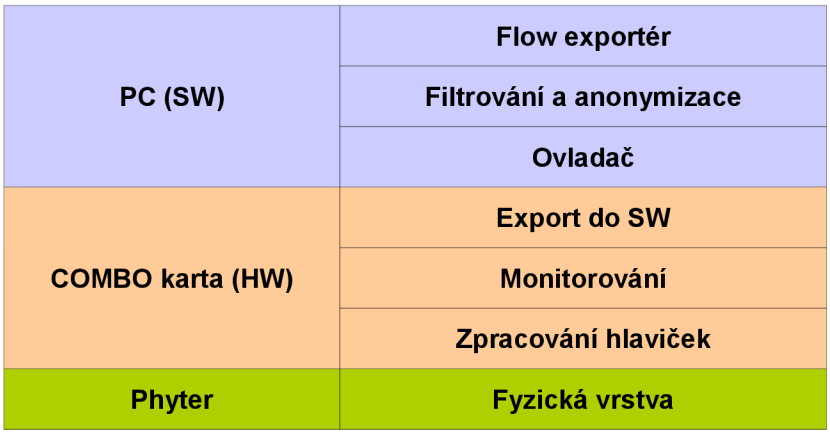
2.2.2 Použití

NetFlow poskytuje informace užitečné při řešení problémů v mnoha oblastech správy sítí. Jeho nasazení umožní:

- zvýšení bezpečnosti sítě a možnost odhalení vnějších i vnitřních útoků
- detailní sledování uživatelů a služeb
- odhalení nesprávných konfigurací
- dohledávání incidentů
- efektivní plánování kapacit sítě
- dlouhodobé uložení statistik o síťovém provozu
- dodržování vyhlášky o elektronické komunikaci
- získávání přehledných výpisů o síťovém provozu
- účtování a fakturaci na základě přenesených dat
- kontrolu dodržování FUP

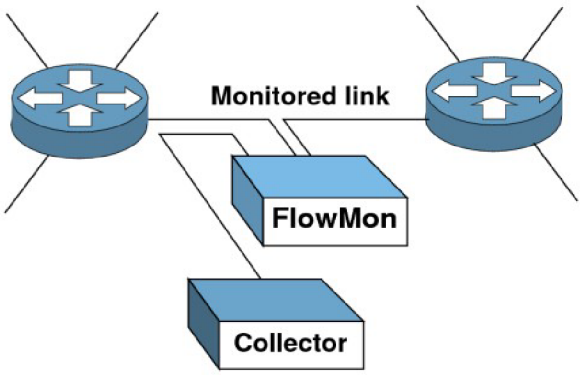
2.3 Sonda FlowMon

Pasivní monitorovací sonda FlowMon je NetFlow exportér vyvíjený v rámci projektu Liberouter a realizovaný jako PCI (PCI-X) karta doplněná o potřebné open-source programové vybavení. HW část vychází z rodiny akceleračních karet COMBO vyvíjených v rámci výzkumného záměru sdružení CESNET a mezinárodních projektů EU. Je založená na programovatelných strukturách a obvodech FPGA (Field Programmable Gate Array) a slouží k akceleraci časově kritických operací. Jedná se zejména o bezeztrátový příjem paketů a jejich agregaci do záznamů o IP tocích. SW část tvoří programové vybavení zodpovědné za obsluhu akcelerační karty, vyčítání záznamů z paměti karty a jejich odesílání protokolem NetFlow nebo IPFIX na kolektor. Systém může být rozdělen do několika vrstev se specifickými úkoly (*obrázek 2.4*).



Obrázek 2.4: vrstvový model sondy FlowMon

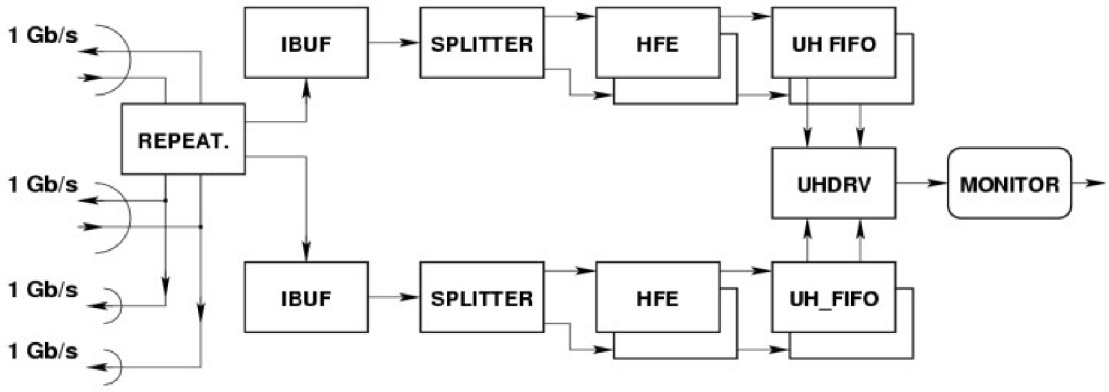
Výhodou nezávislé sondy oproti směrovačům je možnost připojit ji do libovolného bodu v síti, a to transparentním způsobem. Při zapojení dle *obrázku 2.5* se chová jako *T-splitter*. Příchozí data jsou přeposílána původnímu příjemci a jejich kopie paralelně zpracovávána. Z pohledu sítě do dat není zasahováno, proto pasivní sonda. Exportované statistiky jsou navíc na kolektor odesílány dedikovanou linkou. Díky tomu je na monitorované lince sonda zcela neviditelná (na vrstvách L2 a výše). Tento rys z ní činí velmi obtížný cíl pro případné útočníky.



Obrázek 2.5: Zapojení sondy FlowMon přímo na linku

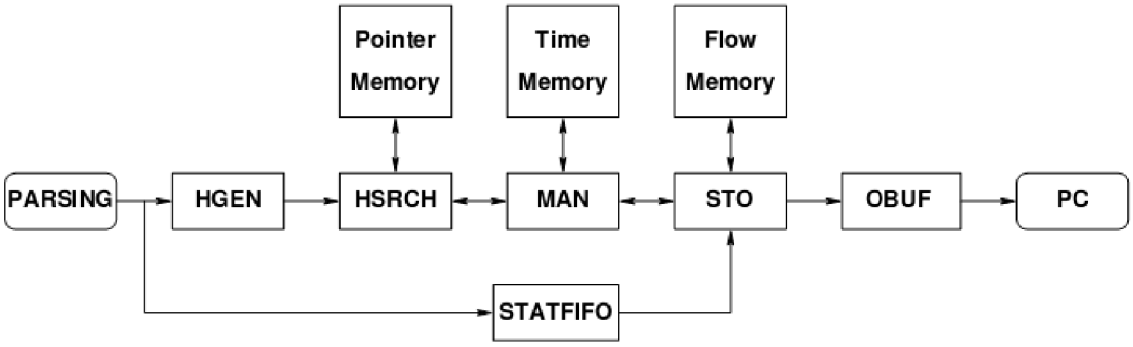
2.3.1 HW část sondy

Akcelerační karta se skládá ze dvou desek, desky základní a desky rozhraní, které jsou osazeny obvody FPGA a externími pamětmi. Právě díky obvodům FPGA je karta schopna akcelerovat mnohé aplikace v oblasti vysokorychlostních sítí, kde není možné nasadit běžné softwarové řešení. Vstupy byly navrženy tak, aby bylo možné monitorovat odchozí i příchozí provoz na lince. První dva síťové porty jsou zapojeny jako obousměrný opakovač a veškerý provoz je z nich kopírován zaprvé do karty ke zpracování a zadruhé na zbylé dva výstupy karty, umožňující připojit další síťová zařízení. Firmware tvoří do série zapojené paralelně pracující jednotky, z nichž některé jsou pro zvýšení výkonu zdvojeny. Celý firmware se dělí na dvě části, zpracování paketů a monitorování toků. *Obrázek 2.6* schematicky popisuje zapojení opakovače a jednotek pro zpracování paketů.



Obrázek 2.6: Zapojení opakovače a jednotek pro zpracování paketů

Z opakovače přicházejí pakety do vstupní vyrovnávací paměti IBUF (*Input Buffer*), kde jsou zkontrolovány pomocí CRC (*Cyclic Redundancy Check*). Validním paketům je přiřazena časová značka a jsou puštěny dále. Přijaté pakety mohou být vzorkovány, a to buď s konstantní frekvencí nebo frekvencí dynamicky se měnící podle aktuálního provozu na síti (adaptivní vzorkování). Adaptivní vzorkování zaručuje optimální využití paměti a konstantní zátěž při přesunu záznamů z karty do paměti počítače. Po navzorkování a přidělení časových značek jsou data *splitterem* rozdělena do několika větví na extrakci položek z hlaviček. Samotnou extrakci provádí HFE (*Header Field Extractor*) procesory s redukovanou instrukční sadou a periferiemi pro proudové zpracování dat (paketů). HFE z extrahovaných položek sestaví strukturu nazvanou UH hlavička (*Unified Header*) a dočasně ji uloží do jedné z jednotek UH-FIFO. *UH driver* (UHDRV) odtud následně UH hlavičky metodou round-robin vyčítá do části pro monitorování toků.

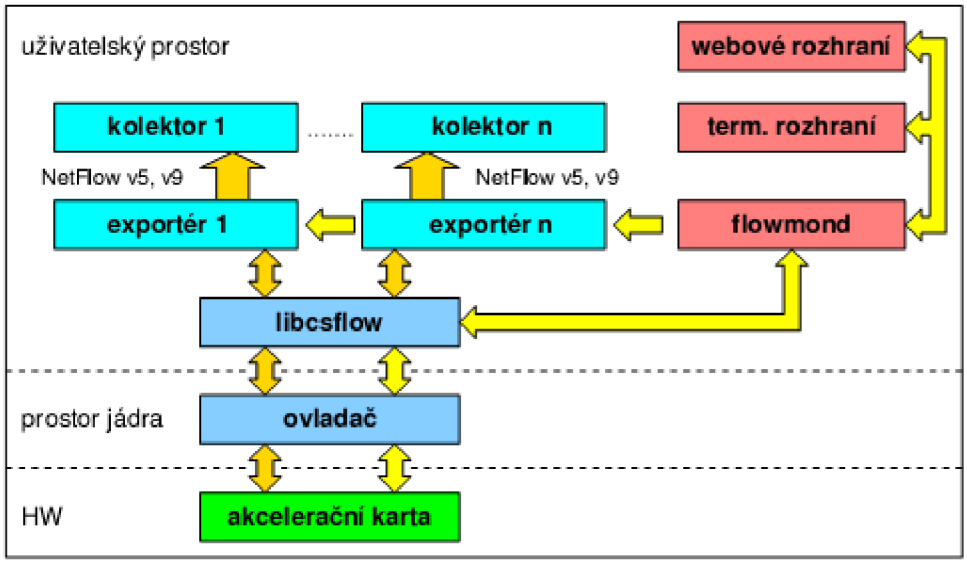


Obrázek 2.7: Zapojení jednotek pro monitorování toků

Na vstupu do monitorovací části jsou data duplikována do dvou cest (kontrolní a datové). Datová cesta je tvořena pouze frontou STATFIFO udržující UH hlavičky, než jsou zpracovány jednotkou *Storage* (STO). Kontrolní cesta začíná generátorem hash (HGEN). Na základě polí identifikujících tok spočítá 64bitovou hash, která se stává novým identifikátorem toku. Část hash hodnoty je použita jako adresa do *Pointer Memory* pro *Hash Search* (HSRCH). Ten má za úkol ověřit, zda už je o toku založen záznam a případně poskytnout jeho adresu ve *Flow Memory*. *Manager* (MAN) komunikuje s HSRCH a STO, spravuje toky a uchovává jejich stavy. Jednotka STO se stará o statistická data (počet přijatých paketů, koncová časová značka atd.). Modifikuje je dle příkazů MAN a dat z fronty STATFIFO. Také kontroluje toky z hlediska překročení aktivního timeoutu (zda-li záznam o toku není v paměti uchováván déle, než je nastavené maximum) a posílá požadavky na jejich export. Při obdržení příkazu pro smazání záznamu jej uvolní z paměti a odešle do bufferu (OBUF). Z jednotky OBUF je DMA (*Direct Memory Access*) přenosem zapsán do bufferu ovladače v počítači.

2.3.2 SW část sondy

Programové vybavení sondy se skládá z několika částí – ovladače jádra, uživatelských knihoven *libcsflow* a *libcombo*, konfiguračních programů a exportéru NetFlow dat. Struktura softwaru je viditelná na *obrázku 2.8*.



Obrázek 2.8: Struktura SW vybavení sondy

Ovladač jádra obsluhuje akcelerační karty a prostřednictvím knihoven *libcsflow* a *libcombo* poskytuje jednotné rozhraní uživatelským aplikacím. Pomocí DMA přenosů čte exportované záznamy a ukládá je do vyrovnávací paměti hostitelského počítače, kde jsou přístupny uživatelským aplikacím. Z uživatelského pohledu je práce s kartou rozdělena na dvě hlavní činnosti. První spočívá v konfiguraci karty a ovládání jejích funkčních bloků (nastavování parametrů, nahrávání firmwaru atd.), druhou tvoří čtení dat z karty, jejich následné převedení do podoby NetFlow záznamů a odeslání na kolektor.

Funkci prostředníka mezi ovladačem a aplikacemi vyšších vrstev plní knihovna *libcsflow*. Obsahuje specifická systémová volání a poskytuje standardní rozhraní jazyka C.

Po zavedení ovladače jádra jsou k nahrání firmwaru, inicializaci a spuštění sondy použity konfigurační programy. Obsluha probíhá buď z příkazového řádku pomocí sady shellových skriptů nebo přes webové rozhraní.

Nakonec je spuštěn program exportéru, který přijímá data ze sondy a ve formátu NetFlow v5, NetFlow v9 nebo IPFIX je odesílá na kolektor k uložení a dalšímu zpracování.

2.4 Vestavěný test

Problematika testování hardwaru a softwaru je velice obsáhlá a není důvod ji na tomto místě celou detailně rozebírat. Na druhou stranu je jistě vhodné zmínit alespoň základní myšlenky a cíle testování a samozřejmě si přiblížit význam pojmu *vestavěný test*, který se objevuje přímo v názvu práce.

Hlavním úkolem testování je detekovat chyby, aby mohly být nalezeny a opraveny jejich příčiny. Slavná poučka říká, že testování nemůže zaručit bezchybné fungování za jakýchkoliv podmínek, ale jen odhalit chybné fungování za určitých podmínek. A protože nelze vytvořit „dokonalý" test, existují různé testovací techniky a postupy, dramaticky se lišící s účelem testu a samozřejmě s testovaným zařízením.

Vestavěný test se vyznačuje, jak už napovídá jeho název, svým umístěním v rámci nějakého většího celku. Tímto celkem a zároveň i objektem testování bude v případě této práce sonda FlowMon.

BIST (*built-in self-test* – vestavěný test sebe sama) má sloužit ke kontrole správného fungování zařízení. Důraz je kladen především na rychlost, nezávislost na dalším vybavení a jednoduchou obsluhu tak, aby mohl být používán nejen testery při vývoji, ale i uživateli nebo automaticky samotným zařízením (např. z počítačů známý *POST – power-on self-test*).

3 Vývoj testu

Podle zadání má být vytvořen vestavěný test sondy FlowMon v podobě softwaru, který prověří její funkčnost pro nasazení v reálné síti. V případě poruchy jeho výsledky pomohou k lokalizaci a odstranění objevených chyb.

3.1 Požadavky

Specifikace a analýza požadavků je úvodní etapou při vývoji softwaru. Tato kapitola naznačuje, jakým směrem se bude ubírat další práce. Je třeba si ujasnit např. v jakých situacích se bude testu využívat, kdo s ním bude pracovat, co od něj bude vyžadovat a jaké nástroje bude mít k dispozici. Z odpovědí na podobné otázky lze potom vyčíst, na co se zaměřit v dalších vývojových etapách.

Jak již bylo naznačeno v *kapitole 2.4*, vestavěný test se bude používat především k rychlému ověření správného fungování sondy. Musí být vytvořen tak, aby mohl být spuštěn i uživatelem nebo automaticky skriptem a poskytl srozumitelné výstupní informace. Zároveň nesmí být vázán na žádný specializovaný HW nebo SW, jehož přítomnost není při spouštění testu zaručena. Dalším důležitým faktorem je snadná možnost rozšíření a přenositelnost kvůli případnému budoucímu využití i při projektu Flexible FlowMon.

Uživatel dá jistě přenost spíše jednoduššímu nástroji se základní sadou časově nenáročných testů před sice obsáhlejším a podrobnějším, ale na čas i ovládání náročnějším programem. Delší doba čekání na výsledek může vést k nechuti test používat a jeho automatické spouštění i snížit zájem o produkt jako celek.

Informace o výsledcích testování by měly být podávány srozumitelnou formou tak, aby byly užitečné i bez znalosti technických podrobností. Uživateli často stačí rozlišit, jestli zařízení funguje nebo ne. Nic neříkající výpis paměti opět spíše odrazuje, ale záznamy se základními údaji o průběhu testů mohou pomoci.

Samotná sonda se ovládá pomocí programů a skriptů spouštěných z příkazového řádku nebo přes webové rozhraní. Vestavěný test je součástí sondy, a proto by i jeho ovládání mělo být řešeno jedním z uvedených způsobů.

Z oblasti použití testu je zřejmé, že nelze spoléhat na přítomnost žádného konkrétního HW nebo SW kromě toho, který je přímo spojen s provozem sondy.

Požadavky na vyvíjený software by se daly shrnout do následujících bodu:

- rychlost
- srozumitelný výstup
- jednoduchá obsluha
- nezávislost na dalším HW nebo SW
- snadná možnost rozšíření
- přenositelnost

3.2 Návrh

V první části této kapitoly bude navržena sada testů, která dle specifikace z kapitoly 3.1 ověří správné fungování sondy FlowMon. Testy budou orientovány na rychlost, srozumitelnost výstupu, nezávislost na HW a SW, jednoduchou obsluhu a přenositelnost, aby vyhověly požadavkům uživatele. Ve druhé části kapitoly pak budou navrženy nástroje, potřebné pro realizaci těchto testů.

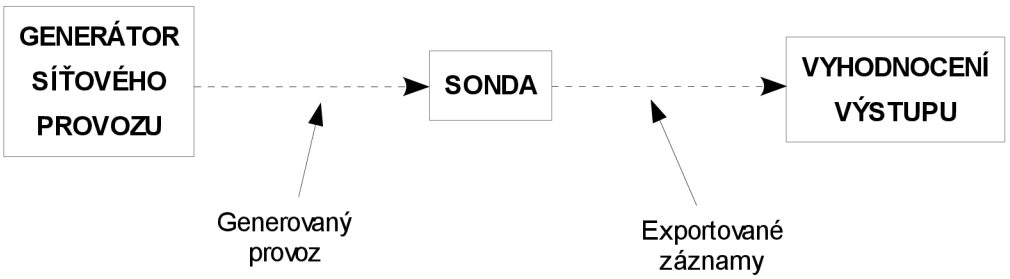
Na úvod je jistě dobré zmínit dva základní přístupy k testování, tzv. *black box testing* a *white box testing*.

Při testování metodou černé skříňky (*black box testing*) nemá testující znalosti o tom, jak testovaný systém funguje uvnitř a zaměřuje se pouze na jeho vstupy a výstupy. Ví, jak by měl systém reagovat na různá vstupní data, tato data mu poskytuje a porovnává, zda se skutečné reakce shodují s očekávanými.

Naopak při testování metodou bílé skříňky (*white box testing*) má testující přístup k objektům uvnitř systému. Může zkoumat jednotlivé části systému odděleně, přizpůsobovat vstupní data svým znalostem atd. Testování touto metodou bývá podrobnější a náročnější. Testující má sice o systému mnohem více informací, tento fakt ale nemusí nutně mít na testování pozitivní vliv, protože je svými znalostmi ovlivněn.

V praxi se mezi přístupy volí podle cílů, na které je testování zaměřeno a často se využívá kombinace obou.

Vzhledem k požadavkům na zde navrhovaný vestavěný test bude technika testování vycházet z metody černé skříňky. Při každém dílčím testu bude na sondu posílán uměle generovaný síťový provoz. Porovnáním exportovaných údajů s očekávaným výstupem se určí úspěch či neúspěch v daném testu. Schéma toku dat je znázorněno na *obrázku 3.1*. Vstupní data jednotlivých testů budou částečně přizpůsobena architektuře sondy. Tím by se mělo docílit využití dostupných prostředků a pokrytí základních funkcí sondy během několika rychlých testů.



Obrázek 3.1: Tok dat při testu

3.2.1 Test propustnosti

Tento test má za úkol ověřit, zda si sonda dokáže poradit s jakýmkoliv provozem na lince z hlediska rychlosti zpracování. Protože z příchozích dat se při monitorování zpracovávají pouze hlavičky, provoz by měl být orientován na co nejvyšší počet hlaviček za časovou jednotku. Toho se dosáhne posíláním co nejkratších rámců maximální rychlostí linky.

3.2.2 Test přetečení paměti

Kapacita paměti sondy se v závislosti na modelu a firmware pohybuje v rozmezí 64 000 – 512 000 záznamů. Každému toku je přiřazen jeden záznam, který je odeslán do výstupního bufferu a uvolněn z paměti v případě, že:

- doba trvání toku je delší než nastavená hodnota aktivního timeoutu
- toku nepřišel nový rámec po dobu delší než nastavená hodnota neaktivního timeoutu
- sonda pozná konec toku (např. podle příznaku v TCP hlavičce)

Za předpokladu, že je paměť zcela zaplněna a na vstupu se objeví nový tok, mohlo by dojít k jejímu přetečení. V takové situaci se náhodě vybere starý záznam a nahradí se novým. Fungování právě tohoto mechanismu kontroluje test přetečení paměti.

Kombinace provozu a timeoutů musí být nastavena tak, aby došlo ke spuštění testovaného mechanismu. Toho lze jednoduše docílit například odesláním jednoho milionu rámců během času kratšího, než je hodnota neaktivního timeoutu, kdy každý rámec bude patřit jinému toku.

3.2.3 Test agregace

Po dvou testech zaměřených na vysokou zátěž přichází na řadu testy jednotlivých funkcí. Prvním z nich je kontrola práce monitorovací části sondy. Ověřuje, zda se příchozí rámce správně přiřazují svým tokům a zda se adekvátně upravují sledované hodnoty v záznamech o tocích, jako např. velikost přijatých dat, čas příjmu posledního rámce, počet přijatých rámců atd.

V rámci testu agregace lze vymyslet a provést řadu dílčích testů specializovaných na sledování různých hodnot, sbíraných jak o jednotlivých tocích, tak i o provozu jako celku. Velice důležité je vždy uvážit skladbu provozu a přizpůsobit ji konkrétnímu testu. Je třeba upozornit, že návrh komplikovaných testů může vést ke složité implementaci a obtížnému vyhodnocování výsledků, a proto se příliš nehodí pro automatizované zpracování.

Jako součást vestavěného testu je vhodná kontrola správného přiřazování příchozích rámců svým tokům. Provoz by měl obsahovat dostatečné množství toků, řádově v tisících. Pozor na mechanismus ošetření přetečení paměti, který může způsobit nechtěný předčasný export záznamu a tím i znehodnocení výsledků. Stejný problém může vyvolat i vypršení aktivního nebo neaktivního timeoutu. Vhodné parametry mohou být např. 100 000 rámců odeslaných do jedné sekundy, když každá pětice rámců tvoří jeden tok a timeouty jsou nastaveny na více než jednu sekundu. Na výstupu se kontroluje počet zachycených toků a počet rámců v toku. Pro uvedený příklad by požadované hodnoty byly 20 000 zachycených toků, 5 rámců v každém toku.

Dalším příkladem dílčího testu může být kontrola objemu přenesených dat. V provozu by se měly vyskytovat rámce různých délek, ale kromě tohoto parametru postačí jednoduchá struktura provozu.

Posledním příkladem kontroly práce monitorovací části jsou testy pracující s časovými hodnotami. Odesláním několika rámců v pořadí dle *obrázku 3.2* lze ověřit přiřazování časových značek začátku a konce toku. Mělo by platit, že *T1_start* < *T2_start* < *T3_start* < *T4_start* < *T1_end* < *T2_end* < *T3_end* < *T4_end*, kde *TX_start* je časová značka začátku toku X a *TX_end* je časová

značka konce toku X. Funkci neaktivního timeoutu lze zkontrolovat odesláním dvou rámců téhož toku, mezi nimiž bude časový odstup při prvním pokusu vyšší než hodnota v nastavení sondy (exportovány by měly být dva toky) a při druhém pokusu nižší než hodnota v nastavení sondy (exportován by měl být tok jeden). Aktivní timeout ověří odesílání rámců jednoho toku po dobu delší, než je hodnota v nastavení sondy, následné odečtení času odeslání prvního rámce od času exportu toku a porovnání tohoto rozdílu s nastavenou hodnotou.

T1R1	T2R1	T3R1	T4R1	T1R2	T2R2	T3R2	T4R2

Obrázek 3.2: Pořadí odeslání rámců při ověřování přiřazování časových značek

T2R1 znamená první rámec druhého toku. Rámce jsou odesílány zleva doprava, tzn. prvním odeslaným je *T1R1*, posledním pak *T4R2*.

3.2.4 Test vzorkování

Vstupní vzorkování (*sampling*) umožňuje filtraci příchozích rámců hned na bufferu (IBUF), tedy ještě před začátkem zpracovávání jejich obsahu. Pokud uživatel nevyžaduje přesné informace o protékajícím provozu, zapnuté vzorkování ušetří práci nejen sondě, ale redukcí objemu exportovaných dat také všem zařízením, která s těmito daty poté pracují.

Funkci vstupního vzorkování lze zkontrolovat například odesláním rámců dle *obrázku 3.3*. Každý tok má 100 rámců, toků je celkem 10, tok vždy odesílán „v celku“ (posloupnost rámců jednoho toku není přerušena rámcem jiného toku). Při vzorkování 1:20 by mělo být zachyceno 10 toků, 5 rámců v každém toku.

tok1	tok2	tok3	tok4	tok5	tok6	tok7	tok8	tok9	tok10
100r	100r	100r	100r	100r	100r	100r	100r	100r	100r

Obrázek 3.3: Schéma provozu pro test vzorkování (příklad 1)

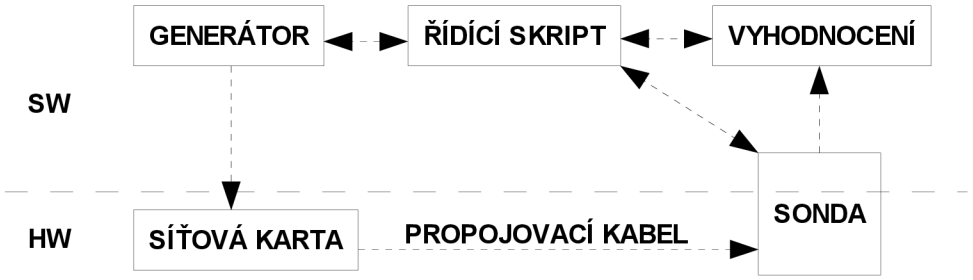
Druhým příkladem a alternativou k první možnosti je pravidelné střídání toků po každém odeslaném rámci (*obrázek 3.4*). Pokud zůstanou zachovány počty toků, rámců a vzorkování z prvního příkladu, na výstupu sondy by se měl objevit pouze 1 tok s 50ti rámci.

T1R1	T2R1	...	T9R1	T10R1	T1R2	T2R2	...	T9R100	T10R100

Obrázek 3.4: Schéma provozu pro test vzorkování (příklad 2)

3.2.5 Potřebné nástroje

Nepostradatelným nástrojem pro testování sondy je počítač se síťovou kartou a testovanou sondou FlowMon. Síťovou kartu využívá generátor síťového provozu a její výstup musí být propojen se vstupem sondy, která provoz monitoruje a pořizuje o něm záznamy. Exportované záznamy jsou poté zpracovány nástrojem pro vyhodnocení výstupu ze sondy. Na průběh celého procesu dohlíží řídící skript. *Obrázek 3.5* znázorňuje propojení jednotlivých nástrojů.



Obrázek 3.5: Propojení jednotlivých nástrojů

K provedení testů je tedy nutné mít k dispozici:

- počítač se síťovou kartou, FlowMon sondou a propojovacím kabelem
- generátor síťového provozu
- nástroj pro vyhodnocení výstupu ze sondy
- řídící skript

Počítač, který má být pro testování použit, samozřejmě musí splňovat požadavky pro instalaci sondy uvedené v příručce. Tomuto faktu bude přizpůsoben i vývoj softwarových nástrojů, potřebných pro vestavěný test. Předpokladem pro spuštění testu je stav, kdy po zadání příkazu „flowmon“ dojde k úspěšnému rozběhnutí sondy.

Kvůli zajištění vzájemné kompatibility a možnosti využít již vyvinuté nástroje pro práci se sondou, jako knihovnu *libcsflow* nebo konfigurační programy, bude k implementaci generátoru, nástroje pro vyhodnocení výstupu a řídícího skriptu použito jazyka C a shellových skriptů.

Úkolem generátoru je generování síťového provozu pro jednotlivé testy. Protože se není možné spolehnout na přítomnost síťové karty v počítači, ve kterém je instalována sonda, generátor bude tvořen samostatnou aplikací, kterou půjde spustit i na vzdáleném stroji. V takovém případě je ale třeba zajistit, aby během testování netekl linkou jiný než testovací provoz, který by samozřejmě znehodnotil výsledky testování. Proto nelze monitorované linky využít ani k ovládání aplikace generátoru.

Generátor musí být schopen vytvářet provoz, který splňuje požadavky jednotlivých testů. Jak je patrné z příkladů, je nezbytné umožnit přístup k nastavení parametrů jako počet toků, počet rámců v toku, velikost rámce, uspořádání rámců v provozu, rychlost přenosu atd. K tomuto účelu lze využít knihovnu *pcap*, která dovoluje sestavovat hlavičky transportní a síťové vrstvy a poskytuje rychlé rozhraní pro práci se síťovou kartou. Tuto knihovnu používají i známé síťové analyzéry *tcpdump* a *wireshark*.

Nástroj pro vyhodnocení výstupu zachycuje a vyhodnocuje exportované záznamy o provozu. Jedná se tak vlastě o kolektor, rozšířený o funkci vyhodnocování na základě porovnávání skutečného výstupu s očekávaným. K tomu potřebuje informace nejen o očekávaném výstupu, ale i o hodnotách, které mají být porovnávány a způsobu, jakým má porovnávání probíhat (povolené odchylky atd.). Měl by umět přijmout a zpracovat data z exportéru ve všech podporovaných formátech (NetFlow5, NetFlow9 a IPFIX). Protože detailní návrh a implementace takového nástroje jsou časově velmi náročné operace, bude v rámci této práce nahrazen použitím již existujícího kolektoru *flowmoncol* a

ručním vyhodnocením výsledků testů. Vývoj nástroje pak bude jednou z možností dalšího rozšíření softwaru vestavěného testu.

Řídící skript zajišťuje koordinaci mezi generátorem síťového provozu, nástrojem pro vyhodnocení výstupu a sondou.

3.3 Implementace

Tato kapitola popisuje implementaci navržených softwarových nástrojů, jejich ovládání a jednotlivé testy, vybrané do vestavěného testu.

3.3.1 Generátor síťového provozu

Program generátoru byl implementován dle návrhu jako samostatně spustitelná konzolová aplikace. Jazyka C bylo zvoleno s ohledem na co nejvyšší kompatibilitu s ostatními programy vyvíjenými pro práci se sondou. Využití funkcí knihovny *pcap* přináší možnost pracovat se síťovou kartou rychleji a obejít standardní knihovny jádra operačního systému. Důležitý je také přístup ke tvorbě hlaviček transportní a síťové vrstvy, díky kterému může jedno zařízení simulovat různorodý provoz, potřebný právě při testování sondy.

Aby bylo možné generátor použít i do budoucnosti pro později vyvinuté testy, je generování implementováno obecně a konkrétní parametry provozu se zadávají až při spuštění programu pomocí argumentů příkazového řádku. Takto lze nastavit:

- síťové zařízení použité k odesílání
- počet odesílaných toků
- počet rámců v jednom toku
- velikost rámce v bytech
- čas prodlevy mezi odesláním dvou rámců
- pořadí odesílaných rámců
- protokol transportní vrstvy

Podrobnější informace o použití generátoru včetně popisu všech argumentů příkazového řádku a ukázkových příkladů spuštění jsou dostupné v elektronické podobně na přiloženém paměťovém médiu.

3.3.2 Řídící skript

Řídící skript má na starost ovládání a kontrolu ostatních programů, zapojených do procesu testování. Spouští generátor a nastavuje parametry provozu pro jednotlivé testy. Stará se o chod sondy, kterou po každém provedeném testu restartuje, aby bylo zaručeno vymazání všech pamětí, nulování čítačů atd. Spouští kolektor a jeho výstup ukládá do souborů k určení úspěchu nebo neúspěchu sondy v jednotlivých testech.

3.3.3 Jednotlivé testy

Tato podkapitola se věnuje jednotlivým testům, vybraným do vestavěného testu. Z každé z navržených oblastí byl implemenován jeden test, aby byly pokryty všechny oblasti při zachování časové nenáročnosti celého testování. Záměrně jsou vždy zmíněny jen podstatné parametry testu, přesnou specifikaci nastavení provozu i sondy lze nalézt na přiloženém paměťovém médiu. U každého testu je uveden také očekávaný výstup a možné důsledky jeho neúspěchu na práci sondy v reálném provozu.

Neúspěch ve více testech značí chybu, která má vliv na celý systém. Pokud se na výstupu neobjevují vůbec žádná data, může se jednat o úplnou nefunkčnost části nebo celé sondy.

Test propustnosti

PROVOZ	
Počet odesílaných toků:	1
Počet rámců v jednom toku:	1 000 000
Velikost rámce v bytech:	64
Čas prodlevy mezi odesláním dvou rámců:	bez prodlevy
SONDA	
Vzorkování:	1:1 (vypnuto)

Z výstupních dat se sleduje počet přijatých toků a celkový počet přijatých rámců. Hodnoty by samozřejmě měly odpovídat nastaveným parametrům, tedy jeden tok a milion rámců.

Neúspěch znamená, že sonda nemusí být schopna zpracovat veškerý provoz při maximálním vytížení linky. Řešením je omezení rychlosti linky, zapnutí vzorkování nebo výkonnější sonda.

Test přetečení paměti

PROVOZ	
Počet odesílaných toků:	1 000 000
Počet rámců v jednom toku:	1
Velikost rámce v bytech:	1024
Čas prodlevy mezi odesláním dvou rámců:	bez prodlevy
SONDA	
Neaktivní timeout:	20s
Aktivní timeout:	60s
Vzorkování:	1:1 (vypnuto)

Z výstupních dat se sleduje počet přijatých toků, čas a pořadí jejich exportu. Z počtu toků lze vyčíst, zda se toky neztrácí, na čase a pořadí exportu je vidět, jak jsou staré toky v paměti nahrazovány novými.

Neúspěch znamená, že mechanismus ochrany přetečení nefunguje správně. Přetékání se dá zabránit například zkrácením timeoutů nebo zapnutím vzorkování.

Test správného přiřazování příchozích rámců svým tokům (agregace)

PROVOZ	
Počet odesílaných toků:	20 000
Počet rámců v jednom toku:	5
Velikost rámce v bytech:	1024
Čas prodlevy mezi odesláním dvou rámců:	bez prodlevy
SONDA	
Vzorkování:	1:1 (vypnuto)

Z výstupních dat se sleduje počet přijatých toků, a počet rámců v každém toku. Exportované hodnoty by se měly shodovat se zadanými parametry provozu.

Neúspěch ukazuje na chybu při extrakci hlaviček nebo v monitorovací části. Při výskytu této chyby není možno spoléhat na správnost monitoringu a chyba se bohužel nedá odstranit ani omezit změnou nastavení sondy.

Test vzorkování

PROVOZ	
Počet odesílaných toků:	1 000
Počet rámců v jednom toku:	100
Velikost rámce v bytech:	1024
Čas prodlevy mezi odesláním dvou rámců:	bez prodlevy
Pořadí odesílaných rámců:	tok v celku
SONDA	
Vzorkování:	konstantní, 1:20

Z výstupních dat se sleduje počet přijatých toků, a počet rámců v každém toku. Na výstupu by se mělo objevit 1000 toků, 5 rámců v každém toku.

Neúspěch znamená chybu vzorkování, kterou lze eliminovat pouze jeho vypnutím.

4 Zhodnocení testu

Zvolené řešení vestavěného testu vychází z potřeby otestovat připravenost sondy pro nasazení v reálné vysokorychlostní síti. Proto testování vychází z metody černé skříňky. Sonda je testována jako celek a jednotlivé testy se zaměřují na zvenku viditelné funkce. Tento způsob umožňuje zkontolovat, zda je možné sondu nasadit do provozu, ale chyby lokalizuje na úrovni nedostupnosti funkcí, a proto není příliš vhodný pro detekci chyb z hlediska jejich opravy.

Pokud by se při vývoji vycházelo z metody bílé skříňky, testování by probíhalo zcela odlišným způsobem. Každý z testů by kontroloval odděleně pouze konkrétní část hardwaru nebo softwaru. Tento přístup by sice dokázal lokalizovat chybnou část sondy, ale nebyl by schopen ověřit, jestli sonda funguje správně jako celek a potvrdit tak možnost jejího nasazení do provozu, což byl hlavní důvod pro zavržení této metody.

Navržené řešení je vhodné pro uživatele, kterým nabízí rychlé a jednoduché otestování základních funkcí pro ověření, zda je sondu možné použít v reálném provozu. Přináší širokou použitelnost, protože není závislé na parametrech jednotek sondy. Přidáním nových testů nebo úpravou těch stávajících lze snadno reagovat na měnící se požadavky a přzpůsobit tak vestavěný test aktuálním potřebám. Větší počet testů ale znamená nárůst časové náročnosti celého testování. Proto bylo implementováno jen několik základních testů a zbytek ponechán jako případná možnost rozšíření.

5 Závěr

Úkolem této bakalářské práce bylo vytvořit vestavěný test sondy FlowMon, který prověří správnost jejího fungování pro nasazení do reálné vysokorychlostní sítě. Za tímto účelem jsem nastudoval oblast monitorování síťového provozu pomocí technologie NetFlow. Zaměřil jsem se při tom hlavně na testovanou sondu. Získané znalosti jsem spolu se souvisejícími pojmy z počítačových sítí shrnul v teoretické části práce.

Praktická část poté popisuje vývoj vestavěného testu. V rámci práce bylo nutné vyvinout zaprvé samotnou sadu testů, která umožní zkontrolovat funkce sondy a zadruhé nástroje, umožňující tyto testy provést.

Navržená obecná technika testování vychází z metody černé skříňky. Na sondu je posílán testovací provoz a sledují se výstupní data sondy. Jednotlivé testy jsou rozděleny do několika oblastí. Z každé oblasti je implementován jeden test, implementace ostatních navržených testů a vývoj nových testů je jednou z možností budoucího rozšíření práce.

Pro generování síťového provozu byla v jazyce C implementována aplikace, nahrazující hardwarový tester, používaný při testování sondy v laboratoři. Pomocí nastavení parametrů provozu může aplikace přizpůsobit generovaný provoz požadavkům konkrétního testu.

K zachycení výstupních dat sondy je využito kolektoru *flowmoncol* a data jsou vyhodnocována ručně. V tomto místě vidím největší možnost rozšíření práce. Vývojem vlastního kolektoru s automatizovaným vyhodnocením výstupu by se vestavěný test posunul o velký kus dopředu.

Literatura

[1] Martin Žádník: *Síťové aplikace a správa sítí: NetFlow*. [online 5/2009].
Dostupné z WWW: <https://wis.fit.vutbr.cz/FIT/st/course-files-st.php/course/ISA-IT/lectures>

[2] CESNET, z.s.p.o.: *FlowMon Probe Handbook*. Version 1.5.0. [online 5/2009].
Dostupné z WWW: <http://www.liberouter.org/flowmon/handbook.html>

[3] Pavel Čeleda, Martin Žádník, Vojtěch Krmíček. Monitorování provozu ve vysokorychlostních sítích na bázi IP toků. In *Širokopásmové sítě a jejich aplikace.* Olomouc: Univerzita Palackého v Olomouci, 2007. ISBN 978-80-244-1687-8, s. 35-44. 29.5.2007, Olomouc.
Dostupné z WWW: <http://www.fi.muni.cz/~xkrmicek/publications/2007_ssja_flowmon.pdf>

[4] *Wikipedia: TCP/IP*. [online 5/2009].
Dostupné z WWW: <http://cs.wikipedia.org/wiki/TCP/IP>

[5] *Wikipedia: Netflow*. [online 5/2009].
Dostupné z WWW: <http://cs.wikipedia.org/wiki/Netflow>

[6] *Wikipedia: Software testing*. [online 5/2009].
Dostupné z WWW: <http://en.wikipedia.org/wiki/Software_testing>

[7] CESNET, z.s.p.o.: *Flexible FlowMon*. 2008. Architektonický dokument.

[8] Pavel Čeleda, Milan Kováčik, Tomáš Koníř, Vojtěch Krmíček, Petr Špringl, Martin Žádník: *FlowMon Probe*. CESNET technical report number 31/2006. [online 5/2009].
Dostupné z WWW: <http://www.cesnet.cz/doc/techzpravy/2006/flowmon-probe/>

[9] *FlowMon*. [online 5/2009].
Dostupné z WWW: <http://www.invea-tech.com/cs/products/flowmon>

[10] *TCPDUMP/LIBPCAP public repository*. [online 5/2009].
Dostupné z WWW: <http://www.tcpdump.org>

[11] CESNET, z.s.p.o.: *Liberouter project web page*. [online 5/2009].
Dostupné z WWW: <http://www.liberouter.org>

[12] Juraj Ivanko: *Testování sondy FlowMon*. Brno, 2007, bakalářská práce, FI MU.

[13] Petr Špringl: *Návrh a implementace programového vybavení pro ovládání a konfiguraci sondy NetFlow*. Brno, 2006, bakalářská práce, FIT VUT v Brně.

Seznam příloh

Příloha 1. CD obsahující elektronickou verzi technické zprávy, zdrojové texty a dokumentaci