VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

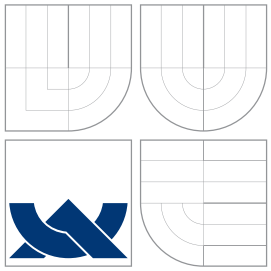
ADAPTIVNÍ VZORKOVÁNÍ PAKETŮ IMPLEMENTOVANÉ V SONDĚ FLOWMON

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

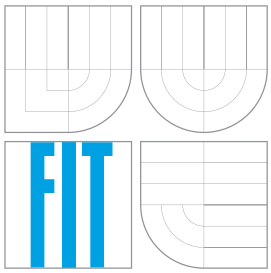
AUTOR PRÁCE
AUTHOR

PETR KAŠTOVSKÝ

BRNO 2007



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

ADAPTIVNÍ VZORKOVÁNÍ PAKETŮ IMPLEMENTOVANÉ V SONDĚ FLOWMON

ADAPTIVE SAMPLING OF INPUT PACKETS IMPLEMENTED IN FLOWMON PROBE

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — PETR KAŠTOVSKÝ
AUTHOR

VEDOUCÍ PRÁCE — Ing. JAN KOŘENEK
SUPERVISOR

BRNO 2007

Abstrakt

V rámci projektu *Liberouter* je vyvíjena sonda *FlowMon* určená pro pasivní monitorování sítí. Sonda na rozdíl od programových řešení poskytuje vysokou stabilitu a přesnost výsledků i pod nadměrnou zátěží či útokem. Pro zajištění kvality výsledků je třeba redukovat množství zpracovávaných dat tak, aby nedošlo k přetížení měřícího systému. Způsobů používaných ke snížení objemu vstupních informací existuje celá řada. Metoda redukce dat použitá v sondě *FlowMon* se nazývá vzorkování. *Adaptivní vzorkovací jednotka* pak zaručuje, že aktuální vzorkování (poměr zpracovaných a zahozených paketů) se přizpůsobí okamžitému stavu monitorované sítě.

Klíčová slova

Liberouter, FlowMon, VHDL, siťový tok, vzorkování, agregace, filtrování, adaptivní vzorkování, redukce dat, kolektor, sonda, NetFlow

Abstract

There is a *FlowMon* probe being developed in a *Libeouter* project that is used for passive network measurements. The probe has better stability and accuracy than sofware based solutions even under a heavy load or network attack. To guarantee a precision of results there is a need to data reduction to prevent measuring system overload. There are few kinds of data reduction. Method used in the *FlowMon* probe is called sampling. *Adaptive sampling unit* sets the sampling rate (rate of processed and discarded packets) according to actual state of measured network.

Keywords

Liberouter, FlowMon, VHDL, ip flow, sampling, aggregation, filtering, adaptive sampling, data reduction, collector, probe, NetFlow

Citace

Petr Kaštovský: Adaptivní vzorkování paketů implementované v sondě FlowMon, bakalářská práce, Brno, FIT VUT v Brně, 2007

Adaptivní vzorkování paketů implementované v sondě FlowMon

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Jana Kořenka. Další informace mi poskytl Martin Žádník, můj vedoucí na projektu Liberouter. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Petr Kaštovský
13. května 2007

Poděkování

Za odborné vedení a vstřícnost při řešení problémů bych chtěl poděkovat Ing. Janu Kořenkovi a Martinovi Žádníkovi.

Zadání

1. Nastudujte problematiku síťových toků na bázi protokolů NetFlow verze 5 a 9.
2. Seznamte se s architekturou sondy FlowMon vyvinuté v rámci projektu Liberouter.
3. Proveďte analýzu chování sondy v případě útoků nebo nadměrného zatížení. Snažte se detekovat situace, kdy dochází k nekontrolovanému zahazování paketů.
4. Na základě provedené analýzy navrhněte mechanizmus adaptivního vzorkování provozu s cílem odstranit nekontrolované zahazování paketů. Snažte se také o minimalizaci počtu paketů zahozených v důsledku vzorkování.
5. Proveďte simulace navrženého řešení pro různé zatížení a typy útoků.
6. Implementujte mechanizmus adaptivního sámplování v jazyce VHDL a ověřte funkci na sondě FlowMon.
7. V závěru diskutujte vlastnosti navrženého řešení.

Licenční smlouva

Licenční smlouva je uložena v archivu Fakulty informačních technologií Vysokého učení technického v Brně.

Obsah

Kapitola 1

Úvod

Monitorování počítačových sítí motivuje rostoucí zájem o služby a aplikace dostupné především skrze internet. S rostoucím počtem soukromých uživatelů a organizací využívajících vzájemnou komunikaci pomocí počítačů roste množství přenášených dat. Někdy jde o nezabezpečené informace nevelké hodnoty, jindy zase o důvěrná data. Například elektronické bankovnictví se stalo velice oblíbenou službou. To přináší zvýšenou hrozbu zneužití předávaných informací.

Nejen bezpečnost vyžaduje sledování toho, co se na komunikačních veletocích děje. Každá domácnost vybavená internetem využívá nabídky některého z poskytovatelů připojení. Přehled a kontrola kvality služeb tvoří další velkou oblast využití síťového monitoringu.

Počet koncových uživatelů internetu v současnosti nabývá obrovských čísel. To přináší i vyšší zátěž na páteřní spoje, které nyní musí obsloužit větší počet klientů. Využívají se nové technologie s vyššími přenosovými rychlostmi a vzniká poptávka po nástrojích schopných tyto důležitá místa sledovat. Dosud si poskytovatelé vysokorychlostních přípojení vystačili s počítačem vybaveným speciálním programem. Ukazuje se však, že tato řešení se stávají nedostačujícími.

Sonda *FlowMon* vyvíjená v rámci projektu *Liberouter* nabízí přístup založený na využití hardwarové akcelerace výpočtů. Akcelerované zpracování dokáže sledovat vyšší přenosové rychlosti, a proto podává přesnější informace. Dojde–li však k extrémní situaci, například útoku či nadměrnému zatížení, je třeba, i u hardwarem urychlených prostředků, redukovat množství zpracovávaných dat a zabránit tak přetížení, selhání monitorovacího systému.

Existuje několik metod redukce dat. Často používanou metodou se zajímavými vlastnostmi je vzorkování. Adaptivní vzorkování tvoří nástavbu, která dynamicky mění parametry vzorkování tak, aby reflektovalo aktuální stav monitorované sítě.

V této práci se snažíme odhalit a zamezit nekontrolovanému zahazování paketů v důsledku redukce dat. Proto je do vstupní části sondy *FlowMon* zařazena adaptivní vzorkovací jednotka. Činností jednotky je v reálném čase přizpůsobovat zahazování paketů tak, aby nedocházelo k přetížení monitorovacího nástroje. Zároveň požadujeme minimalizaci množství zahozených paketů.

Nejprve je vysvětlena teorie monitorování síťových toků a především vhodné metody redukce objemu zpracovávaných dat. Velké objemy dat tvoří překážku při snaze sledovat provoz na vysokorychlostních páteřních spojích. Další kapitola popisuje architekturu sondy *FlowMon* a odhaluje slabá místa, která způsobují nekontrolované zahazování paketů. Ať už v důsledku nadměrného zatížení či útoku. Spolu s analýzou chování sondy v uvedených situacích tak dává základ pro návrh řešení zabraňujícího nežádoucímu nekontrolovanému zahazování paketů.

Po provedení návrhu jednotky následuje implementace a simulační model. Implementace tvoří praktickou část práce. Simulační model slouží k ověření platnosti hypotéz. Poté je implementace testována přímo v hardwaru. V závěru diskutujeme získané řešení, jeho přednosti a omezení, vhodnost a možnosti nasazení.

Kapitola 2

Počítačové sítě a monitorování síťových toků

V první, teoreticky zaměřené, části si vysvětlíme základní pojmy z oblasti počítačových sítí a monitorování síťových toků. Co je to počítačová síť, vrstvový model ISO/OSI. Jaké se používají komunikační protokoly. Proč vůbec monitorovat a jaké nástroje a technologie jsou v současné době využívány. Ozřejmíme si také návaznost na sondu *FlowMon*.

2.1 Počítačová síť

Počítačová síť [5] je souhrnné označení pro technické prostředky, které realizují spojení a výměnu informací mezi počítači. Umožňují tedy uživatelům komunikaci podle určitých pravidel, za účelem sdílení, využívání společných zdrojů nebo výměny zpráv.

2.1.1 Architektura počítačové sítě

Architektura počítačové sítě podle [10] zahrnuje celkové uspořádání sítě, tj. její topologii, formu komunikace, použité komunikační protokoly a základní služby poskytované komunikujícím uzlům.

Postupem času vzniklo několik architektur. Uvedeme si ty nejpoužívanější :

- *SNA* – firemní architektura **IBM** vytvořená s cílem propojení terminálů s centrálními počítači.
- *OSI* – referenční model otevřených systémů tvořící mezinárodní standard a základní koncept architektury počítačové sítě [10].
- *TCP/IP* – vytvořená v rámci výzkumných prací na pokusné akademické síti **ARPANet**. Slouží jako architektura pro současnou nejrozsáhlejší síť Internet.

Architektury TCP/IP i ISO/OSI tvoří několik oddělených vrstev. Každá vrstva reprezentuje jistou funkci poskytovanou vyšší vrstvě a obsahuje protokoly pracující na dané vrstvě. Srovnání vrstev obou modelů můžeme vidět v tabulce 2.1.

V tabulce 2.1 jsou uvedeni i zástupci protokolů příslušných vrstev. Popišme stručně nejdůležitější z nich :

- *TCP* – vytváří, udržuje a ruší transportní spojení, prostřednictvím nichž komunikují aplikační procesy v koncových uzlech sítě.

Model OSI		Model TCP/IP		
č.	vrstva	č.	vrstva	
7	aplikační	4	aplikační	TELNET
6	prezentační			FTP
5	relační			HTTP, ...
4	transportní	3	transportní	TCP, UDP
3	síťová	2	internet	IP, ICMP, ARP
2	linková	1	řadič komunikace	
1	fyzická			

Tabulka 2.1: Srovnání vrstev modelů ISO/OSI a TCP/IP

- *UDP* – poskytuje rychlý a nezabezpečený přenos paketů bez potvrzování příjmu a opakování přenosu (datagramová služba).
- *IP* – směřuje pakety s údaji protokolů vyšší vrstvy (TCP, UDP) od zdrojového k cílovému uzlu na síti, která se skládá z více vzájemně propojených počítačových sítí.
- *ICMP* – přenáší chybové a řídící zprávy mezi uzly a směrovači sítě TCP/IP.
- *ARP* – doplňkový protokol, který zabezpečuje přiřazení IP adres fyzickým adresám linkové vrstvy.

2.2 Síťový tok

Síťový tok (IP flow) je podle [12] množina paketů procházejících pozorovacím bodem v síti v daném časovém intervalu. Všechny pakety příslušící k nějakému toku sdílejí jisté společné vlastnosti. Těmito vlastnostmi mohou být:

- jedno nebo více polí v záhlaví paketu (např. zdrojová/cílová IP adresa, zdrojový/cílový port, druh protokolu transportní vrstvy, atd.)
- jedna nebo více charakteristik samotného paketu (např. počet MPLS značek, a další)
- jedno nebo více polí odvozených od zpracování paketu (např. IP adresa následujícího směrovače, vstupní/výstupní rozhraní směrovače, atd.)

Obecně se neurčuje příslušnost paketu k toku pouze na základě shody hodnot uvedených vlastností. Může to být prakticky libovolná rozhodovací funkce. Například skupina paketů jejichž zdrojová adresa leží v daném rozsahu.

2.3 Monitorování

2.3.1 Aktivní vs. pasivní

Monitorování počítačových sítí dělíme na dva typy (viz. [13]) :

- aktivní – měřený provoz je účelově vytvořen pro potřeby měření
- pasivní – provoz na síti je pouze sledován a není ovlivněn* samotným měřením

Add *) Při pasivním monitorování je třeba odesílat nashromážděná data ze sond rozložených v měřené síti do sběrného místa. Přenos probíhá po stejném médiu a zvyšuje zatížení sítě, neovlivňuje však samotné měření. Snahou samozřejmě zůstává zvýšení zátěže minimalizovat.

Sonda *FlowMon* realizuje právě druhý z výše zmiňovaných typů. Proto se tato práce bude dále zabývat jenom pasivním monitorováním.

2.3.2 Motivace

Požadavky na sledování využití síťových zdrojů pochází zejména od provozovatelů a správců sítí. Původní „best effort“ model internetu nevyhovuje požadavkům současných aplikací. Je to způsobeno tím, že zachází s veškerým provozem stejně. Což je nevhodné, například když chce poskytovatel připojení zajistit kvalitu služby, která je citlivá na zpoždění paketů.

Dalším důvodem pro monitorování může být účtování založené na množství přenesených dat a typu využívaných služeb, také snaha předejít zahlcení a tím porušení podmínek dohodnutých s uživateli. Velmi důležité je také sledování za účelem odhalení bezpečnostních rizik, případně probíhajících útoků.

2.3.3 Množství dat

Mnoho různých požadavků motivujících monitorování sítě vytváří rozdílné požadavky na množství zpracovávaných dat. Například pokud chce poskytovatel připojení účtovat své služby na základě objemu přenesených dat, stačí mu k tomuto účelu souhrnná (tzv. *agregovaná*) informace udávající počet přenesených bajtů. Na druhou stranu, chce–li mít detailnější přehled o službách, které zákazník využívá, není předchozí agregovaná informace dostačující.

Další problematickou částí je sledování stavu sítě, stavu jednotlivých linek. Obvyklé je měření mezi dvěma koncovými body. Tento přístup, ale neodhalí případné problematické spojení na cestě. Cestu obvykle tvoří posloupnost spojení mezi dvěma uzly. Pro vyhledání problematického spoje by tedy bylo třeba umístit monitorovací zařízení do každého uzlu po celé délce cesty. To sebou přináší nárůst dat určených ke zpracování.

Zpracování může probíhat přímo v uzlu sítě, například ve směrovači nebo přepínači. Tato zařízení směřují, přepínají pakety a jsou touto činností vytížena. Nezbývá zde tedy prostor pro výkonově náročné zpracování. Navíc zařízení pro analýzu a uložení získaných informací jsou drahá.

Popsaná omezení vedou k jakési hierarchii prostředků (viz. obrázek 2.1) určených k monitorování sítí. Ve sledované síti jsou rozmístěny sondy, které provádí primární úkony spojené s analýzou provozu. Získané výsledky pak odesílají do sběrných míst určených k sekundárnímu zpracování. Tyto jsou nazývány *kolektory*.

Se vzniklou hierarchií vyvstává problém, jakým způsobem mezi prvky ležícími na jiných vrstvách hierarchie předávat informace. Vhodným přístupem je použít standardizovaný formát, který umožní spolupráci zařízení od různých výrobců. Uvedená kritéria splňují veřejně přístupné formáty společnosti **Cisco** : *NetFlow verze 5* a *NetFlow verze 9*.

2.3.4 Kolektor

Ještě než přejdeme k popisu formátů používaných k exportu dat, zmíníme se krátce o jejich destinaci, tj. kolektoru.

Obrázek 2.1: Blokové schéma monitorovacího systému

Zpravidla není kolektor nic jiného než počítačový systém s možností ukládat velké objemy dat. Dovoluje tak uchovávat dlouhodobější sledování. Vzhledem k tomu, že kolektor sbírá data ze sond rozmístěných po síti, poskytuje komplexnější pohled na monitorovaný objekt (počítačovou síť).

Na kolektoru (počítačovém systému) běží program, který se stará o samotné vyhodnocování získaných informací. V případě sondy *FlowMon* je použito WWW grafické rozhraní *NFSen*.

2.4 NetFlow verze 5

Jedná se o formát *UDP datagramu*, který slouží pro export dat směrem od sondy do kolektoru. Skládá se z hlavičky (viz. tabulka 2.2) a jednoho nebo více záznamů (viz. tabulka 2.3) o síťovém toku (tzv. *flow*). V prvním poli hlavičky datagramu je uvedena verze formátu (1 nebo 5; verze 5 je rozšířením verze 1). Druhé pole hlavičky obsahuje počet následujících záznamů (viz. [1]).

Vzhledem k tomu, že se pro přenos záznamů používá protokol UDP, může dojít ke ztrátám datagramů. Informaci o ztrátě je třeba předat programu zpracovávajícímu data na kolektoru. Proto má hlavička datagramu pole označované *flow sequence*, nebo–li pořadové číslo záznamu. Vznikne součtem předchozího pořadového čísla a počtu záznamů v předchozím datagramu. Po přijetí může vyhodnocující program odečtením očekávané hodnoty od hodnoty v datagramu zjistit množství ztracených záznamů. Tato informace je důležitá pro statistické zpracování informací získaných na kolektoru.

2.5 NetFlow verze 9

Podle [4] se jedná o nejnovější formát používaný pro export dat. Je založen na šablonách, které umožňují definovat tvar exportovaného záznamu. Tento přístup je odolný vůči změnám protokolů. Pokud by se například začali používat nové protokoly, stačí vytvořit novou šablonu pro záznam bez nutnosti změny verze formátu NetFlow. Stejně lze do záznamu přidat nové vlastnosti bez porušení funkčnosti stávajících implementací pouze úpravou šablony.

Byty	Obsah	Popis
0 – 3	verze a počet	verze formátu NetFlow a počet záznamů v datagramu (1 – 30)
4 – 7	SysUptime	současný čas od nastartování směrovače v milisekundách
8 – 11	unix_secs	počet sekund od 0000 UTC 1970
12 – 15	unix_nsecs	zbylé nanosekundy od 0000 UTC 1970
16 – 19	flow_sequence	čítač všech dosud monitorovaných toků
20 – 23	vyhrazeno	nepoužito

Tabulka 2.2: Formát hlavičky datagramu pro NetFlow verze 5

Byty	Obsah	Popis
0 – 3	srcaddr	zdrojová IP adresa
4 – 7	dstaddr	cílová IP adresa
8 – 11	nexthop	IP adresa následujícího směrovače
12 – 15	vstup a výstup	SNMP index vstupního a výstupního rozhraní
16 – 19	dPkts	počet paketů v toku
20 – 23	dOctets	celkový počet bytů náležících do vrstvy L3 v paketech daného toku
24 – 27	first	SysUptime v době začátku toku
28 – 31	first	SysUptime v době přijetí posledního paketu náležícího toku
32 – 35	src a dst port	TCP/UDP zdrojový a cílový port
36 – 39	pad1, tcp_flags, prot a tos	nulový byte, souhrnný OR všech TCP příznaků, IP protokol, typ služby protokolu IP
40 – 43	src_as a dst_as	autonomní systém zdroje a cíle
44 – 47	src_mask, dst_mask a pad2	počet bitů v masce adresy zdroje a cíle, 2 byty nulové

Tabulka 2.3: Formát záznamu o toku pro NetFlow verze 5

Struktura zprávy přenášející záznamy o tocích je oproti verzi 5 složitější. Na druhé straně tak přináší zmíněné výhody šablon. I přesto lze najít společné rysy v obou verzích. Stejně jako verze 5 má i verze 9 hlavičku. Ta vychází ze svého předchůdce a nedoznala výraznějších změn. Naopak část obsahující záznamy o tocích je obohacena o možnost přenosu šablon a stává se tak výrazně komplikovanější. Pro naši potřebu bude postačující vědět, že má zpráva hlavičku a po té následuje datová část obsahující záznamy o tocích a šablony pro popis formátu jednotlivých záznamů.

2.6 Redukce objemu dat

Během monitorování je třeba zpracovat velké množství dat a získané informace pak přenést po měřené síti ke kolektoru. Pokud by se množství dat nijak neredukovalo, mohlo by dojít k situaci, kdy naměřené informace síť zahltí. To je zjevně nežádoucí. Dále je potřeba odlehčit zátěž na směrovačích, které jsou již zatíženy směrováním paketů.

Zmíněné okolnosti vedou k redukci dat, ačkoliv je to v rozporu se zájmem o co nejpřesnější

a detailní měření. Přitom je preferováno jednoprůchodové zpracování, které potlačuje ukládání dat a následné opětovné výpočty. Jednoprůchodový přístup také zvyšuje propustnost měřících zařízení.

Obvykle používané metody pro redukci dat jsou podle [6] :

- *agregace* – spojení dat z dílčích komponent do jednoho celku tak, že komponenty jsou zahozeny. Agregace je obvykle *aditivní*, tzn. součet komponent. Například nalezení celkového provozu z několika zdrojů za časový interval.

  Agregáty poskytují složenou informaci bez možnosti rozlišit příspěvek jednotlivých součástí.

- *filtrování* – výběr podmnožiny dat na základě jejich obsahu. Nevyhovující data jsou zahozena. Například provoz z daného zdroje. Filtrování lze použít k zaměření na jistou část provozu v okamžiku, kdy víme, jaké má vlastnosti.

- *vzorkování* – náhodný nebo pseudonáhodný výběr podmnožiny. Nevybrané položky (pakety) jsou zahozeny. Vzorkování a filtrování se svou definicí do jisté míry překrývají. Je možné implementovat vzorkování pomocí filtrování za cenu složitého pravidla.

Klíčový atribut, který odlišuje agregaci a filtrování od vzorkování, je znalost vlastností síťového provozu. Jinými slovy, pokud chceme provoz agregovat nebo filtrovat musíme předem znát vlastnosti, které sledovaná podmnožina má.

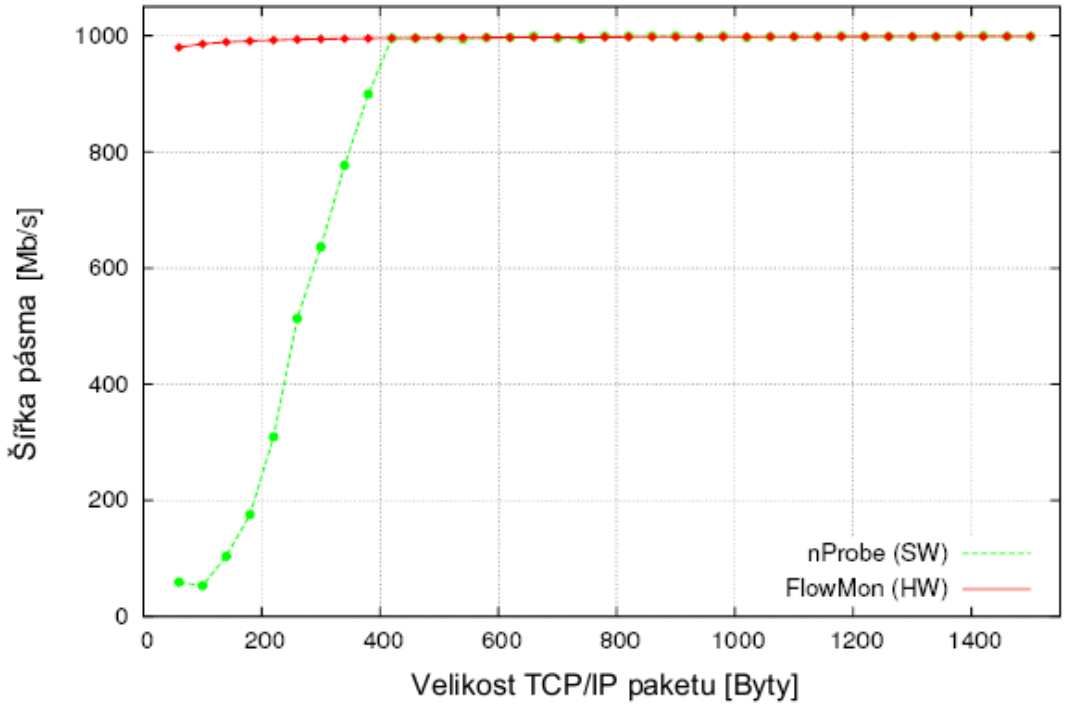
Naneštěstí potřeba redukce dat klesá v místech v monitorovací hierarchii vzdálených od samotných sond. Především pak v úložištích, kde je možné opakované vyhodnocení skladovaných dat.

2.7 Hardwarová akcelerace monitorování

Jak již bylo uvedeno, prvotní zpracování paketů probíhá přímo ve směrovačích. Pokud je u směrovače zapnuta podpora pro zpracování a export dat, znamená to pro produkty společnosti **Cisco** (v nejlepším případě 50 %, v nejhorší případech až 300 %) nárůst zatížení vestavěného procesoru (viz. [2]).

Takováto řešení postačují pro monitorování v běžných uzlech sítě. Chceme–li však monitorovat páteřní spoj, nezbývá než použít PC vybavené síťovou kartou a vhodným programem nebo specializované hardwarové řešení. Hardwarová řešení zastupuje právě sonda *FlowMon*. Mluvíme pak o hardwarově akcelerovaném měřícím zařízení, u něhož je snaha rozložit zátěž na vysoce optimalizované výpočty řešené nízkoúrovňovým přístupem (hardware) a méně optimální výpočty s vyšším stupněm abstrakce (software). Software pak řeší úlohy jako statistické zpracování a vizualizace výsledků; jen obtížně implementovatelné na úrovni hardwaru.

Rozdíl mezi oběma přístupy je pak stabilita, robustnost a zejména vyšší propustnost obvodového řešení (viz. obrázek 2.2). Na rozdíl od softwaru, který obecně vykazuje nižší spolehlivost v mezních případech. Takové chování nemusí vždy znamenat velkou nevýhodu. Především vzhledem k pestrosti síťového provozu. V běžném provozu prakticky nenastává situace, kdy registrujeme jen nejkratší pakety na maximální rychlosti linky.



Obrázek 2.2: Srovnání propustnosti softwarového řešení *nProbe* se sondou *FlowMon* viz. [9]

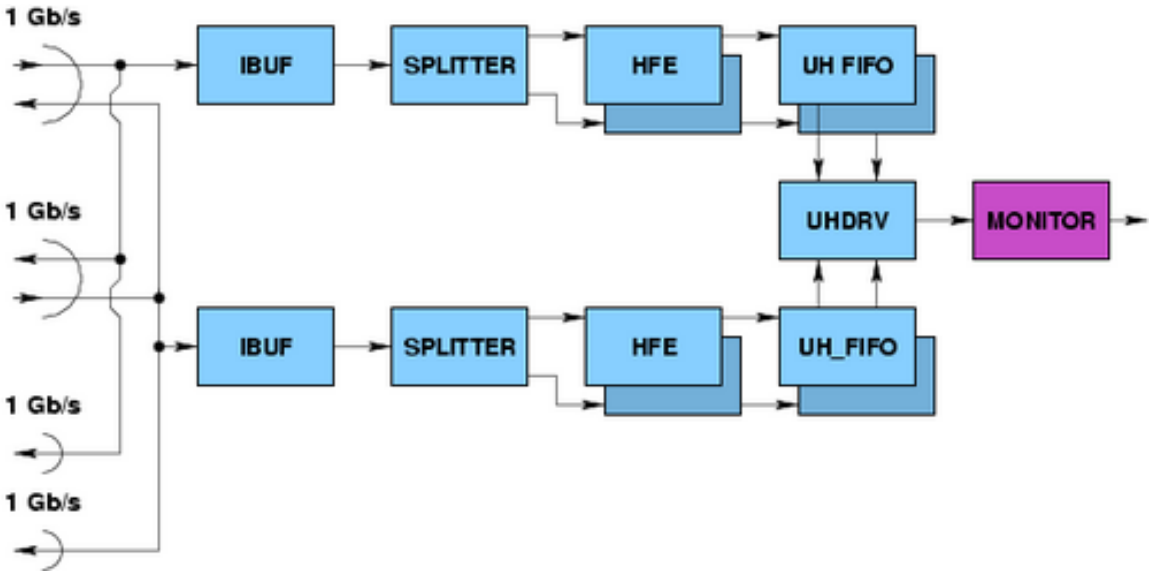
Kapitola 3

Architektura sondy FlowMon

Následující kapitola se věnuje architektuře sondy *FlowMon* vyvíjené v rámci projektu *Liberouter*. Architektura popíšeme na úrovni bloků. Zajímá nás zejména vstupní část, do níž bude zařazena *jednotka adaptivního vzorkování*.

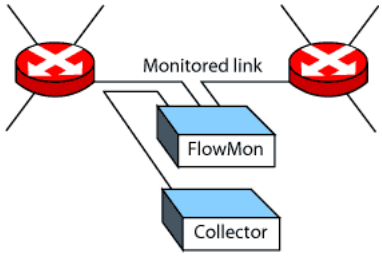
Popisovaná architektura je již druhou vývojovou fází. V této fázi je sonda odladěna ve stabilní verzi a používá se a testuje na několika místech. V současnosti vzniká na projektu *Liberouter* třetí vývojová fáze. Dospěla však zatím pouze do stádia návrhu.

3.1 Bloková struktura



Obrázek 3.1: Bloková struktura sondy *FlowMon* viz. [3]

Z obrázku 3.1 je patrný paralelismus, který ve výkonově náročných aplikacích zvyšuje propustnost nejen procesorů, ale i jiných obvodů. Také můžeme vidět, že sonda obsahuje dvě vstupní rozhraní, na kterých je schopna monitorovat síťový provoz. Zároveň umožňuje zrcadlit provoz z jednoho rozhraní na druhé. Typické zapojení pak vypadá jako na obrázku 3.2.



Obrázek 3.2: Typické zapojení sondy *FlowMon* viz. [14]

Sonda je implementována v programovatelných polích (*FPGA*) firmy **Xilinx**, které jsou umístěny na základní kartě *COMBO6X* a rozšiřující kartě *COMBO-4SFPRO*. Tyto karty jsou rovněž vyvíjeny v rámci projektu *Liberouter*.

Podívejme se nyní na jednotlivé bloky :

- *IBUF* – zkratka IBUF zastupuje vstupní vyrovnávací paměť (z angl. *input buffer*). Tento blok je z hlediska adaptivního vzorkování nejdůležitější. Stará se o příjem *ethernetových rámců*. Nad každým přijatým rámcem provede řadu kontrol a pokud neskončí pozitivním výsledkem, obsah rámce zahodí. V opačném případě postupuje k dalšímu zpracování. Tímto zpracováním může být i vzorkování, které se tak provádí jen s platnými pakety, což chápeme jako požadovanou vlastnost.

- *SPLITTER* – slouží k rozdělení vstupního toku do dvou paralelních výpočetních jednotek. Tímto přístupem dosahujeme vyšší propustnosti. Na druhou stranu je tak zabráno více zdrojů v programovatelném poli. To také vytváří jeden z nejvíce omezujících faktorů hardwarové implementace v *FPGA*.

- *HFE* – (z angl. *header field extractor*) 16bitový procesor architektury s redukovanou instrukční sadou (*RISC*). Instrukční sada je optimalizována pro zpracování toků dat. Procesor analyzuje hlavičky paketů a získaná data ukládá do struktury zvané unifikovaná hlavička. Tato datová struktura pak slouží pro další výpočty a monitorování toků (viz. [3]).

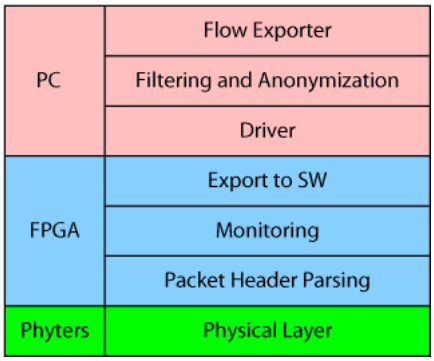
  V současné verzi sondy existuje pro každé síťové rozhraní dvojice procesorů. Každý procesor dokáže zpracovat milion paketů za sekundu. Dvojice pak zvládne zpracovat plně vytíženou 1Gb/s linku v obou směrech.

- *UH_FIFO* – tvoří vyrovnávací paměť pro unifikované hlavičky před jejich dalším zpracováním. Ke každému HFE procesoru přísluší jedno UH_FIFO. V sondě jsou tedy celkem 4.

- *UHDRV* – řídí vyčítání dat ze všech 4 vyrovnávacích pamětí. O výběru paměti rozhoduje spravedlivý algoritmus *round-robin*.

- *MONITOR* – blok by bylo možné dále rozdělit. To však není nezbytné. Postačí vědět, že pro každou unifikovanou hlavičku vypočítá *hash* identifikující tok, jemuž původní paket náležel. Záznam o toku je následně aktualizován (pokud existoval) nebo vytvořen (v opačném případě). Pokud po uplynutí jistého časového intervalu nedojde k obnovení záznamu o toku, smaže se z paměti a odešle do softwaru.

3.2 Vrstvový model



Obrázek 3.3: Rozdělení sondy *FlowMon* do vrstev viz. [14]

Obrázek 3.3 ukazuje sondu v hlubším kontextu. Vazbu na linkovou vrstvu síťového modelu ISO/OSI a zároveň vazbu na programovou část, která je nezbytná pro získání výsledků zpracovaných v sondě.

- *fyzická vrstva (physical layer)* – fyzické připojení k lince; obsluha na úrovni ethernetových rámců
- *zpracování hlaviček paketů (packet header parsing)* – odpovídá vstupní části blokového schématu až po blok MONITOR
- *monitorování (monitoring)* – odpovídá části MONITOR blokového schématu
- *export do softwaru (export to sw)* – v blokovém schématu součást MONITORu; realizován výstupní vyrovnávací pamětí
- *ovladač (driver)* – modul jádra operačního systému umožňující komunikaci uživatelských programů s hardwarem
- *filtrování a anonymizace (filtering and anonymization)* – anonymizace je důležitá k ochraně uživatelských dat a soukromí; filtrování slouží k výběru záznamů následně odesílaných na daný kolektor
- *odesílání záznamů o toku (flow exporter)* – zapouzdřuje odesílané záznamy do zpráv ve formátu NetFlow v5 nebo NetFlow v9 a následně přenáší na kolektor

3.3 Parametry sondy

Realizace sondy na kartách řady *COMBO* přináší jistá omezení daná hardwarovými zdroji. Jak již bylo zmíněno dříve, sonda *FlowMon* je implementována na dvojici karet *COMBO6X* a *COMBO-4SFPRO*. Druhá z karet je označována jako tzv. *interfaceová*. Rozšiřuje základní kartu o 4 gigabitové síťové porty. Na obou kartách se nachází programovatelné pole (*FPGA*) s pevně danými zdroji.

3.3.1 Zpracování hlaviček paketů

První z omezení, při cestě dat od vstupního rozhraní k výstupní vyrovnávací paměti (viz. část 3.1), tvoří jednotka zpracování hlaviček paketů (*HFE*). Pro každé monitorované rozhraní je třeba mít dvojici těchto jednotek, aby bylo možné zvládnout plně zatíženou gigabitovou linku. Uvedené řešení však není dostačující pro linku 10 GB/s.

Možným řešením by mohlo být doplnění návrhu o další jednotky *HFE* a škálovat tak výkon. Tento přístup však naráží na množství zdrojů dostupných v *FPGA*. Zároveň úzce souvisí s následujícím problémem, který by se tak dále vystupňoval.

3.3.2 Přenos mezi kartami

Zpracovaná data je třeba z rozšiřující karty přenést na kartu základní, kde se vyhodnocují informace o síťových tocích. Karty spojují konektory. Konektorů je omezené množství a jejich fyzikální konstrukce také předurčuje maximální frekvenci, na niž jsou schopny spolehlivě pracovat. Počet konektorů a pracovní frekvence udává maximální propustnost spojení. V současném návrhu tato kapacita nedostačuje pro monitorování 10Gb/s linky.

Problém by měl být odstraněn v následující generaci přechodem k použití jediné karty namísto stávající dvojice. Rozvody přímo po desce jsou spolehlivější s lepšími fyzikálními vlastnostmi než mechanicky spojené kontakty.

3.3.3 Paměť pro uchování toků

Úspěšně přenesené informace je třeba zpracovat. Jak bylo zmíněno (viz. 15), vyhodnocuje se příslušnost paketu k síťovému toku. Informace o tocích, které již byly zachyceny, se ukládají v pamětech na základní (také mateřské) kartě. Kapacita a rychlost paměti představuje další omezující prvek.

Paměť leží na cestě od vstupu nejdál a zahltí se i v případě, kdy dvě předchozí omezení neprojeví svůj vliv. Lze ji proto chápat jako omezení největší.

3.3.4 Redukce vstupních dat

První dvě omezení můžeme vyřešit úpravou návrhu, případně jiným přizpůsobením. Nejsou ovlivněna skladbou síťového provozu. Jedním z možných způsobů jak odstranit problémy, by bylo zjistit maximální teoretickou propustnost přenosového kanálu mezi kartami. Podle ní upravit vstupní část sondy tak, aby nikdy nedošlo ke zpracování většího množství dat, než kanál dovoluje přenést.

Pevně nastavená redukce vstupních dat však také není optimálním řešením. Ne vždy jsou linky využity na 100 %. Pak může dojít k redukci dat v okamžiku, kdy není sonda plně vytížena. Ztrácí se tak informace, která může být zpracována. Vzniklé ztráty nejsou přijatelné z pohledu koncového uživatele a je třeba jim předejít vhodnými opatřeními.

Omezení vytvářené velikostí paměti je přímo závislé na profilu síťového provozu. Proto není možné vytvořit ideální řešení založené na vylepšení paměťového bloku, které by mělo dlouhodobější platnost. Řešení by bylo závislé na analýze provozu. Příkladem úpravy může být vylepšená organizace paměti. Taková optimalizace je možná pro zefektivnění práce sondy ve většině případů; není však obecně platná. Stačí například nasadit sondu v místě s novým typem služby vytvářejícím statisticky více síťových toků. Následkem bude zhoršené chování (rychleji zaplněná paměť, dlouhá doba přístupu k uloženým tokům), což není vhodné.

Všechna zmíněná omezení vedou k řešení založeném na redukci množství zpracovávaných dat. Vstupní část bude vždy zpracovávat jen takový počet paketů, který zpracovat dokáže. Přenosový kanál bude přenášet také jen objemy, které přenést zvládne a obdobně to platí i pro paměť toků.

V části 2.6 byly uvedeny tři metody používané za tímto účelem. Pro sondu *FlowMon* se jeví nejvhodnější vzorkování právě proto, že není třeba znát vlastnosti monitorovaného provozu, aby mohl být redukován.

Kapitola 4

Analýza chování sondy

V této kapitole se budeme zabývat mezními situacemi, ve kterých se může sonda *FlowMon* ocitnout. Cílem je odhalit okamžiky, kdy dochází k nekontrolovanému zahazování paketů. Tato informace bude následně důležitá pro návrh a realizaci řešení, které nekontrolovanému zahazování paketů zabrání.

V části 3.3 byly diskutovány omezení vyplývající z architektury a fyzických zdrojů. Každé z omezení se projeví při obecně různých (přesto často podobných) okolnostech. Podíváme se proto na každé omezení v kontextu zkoumaného jevu.

4.1 Nadměrné zatížení

Stav, kdy sonda čelí vysokému síťovému provozu. Data odpovídají běžnému profilu síťového provozu.

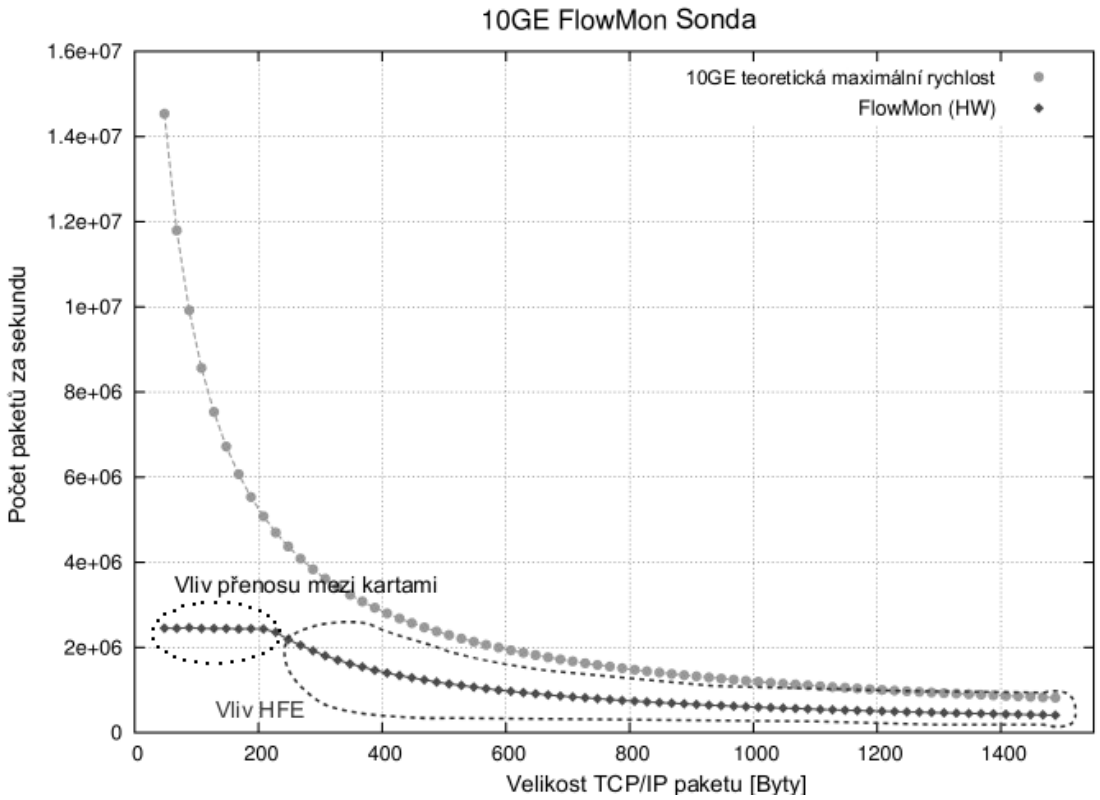
- *HFE* – architektura sondy je vytvořena s cílem zvládnout plně vytíženou 1Gb/s linku. Tuto rychlost zvládá zpracovat. Připojíme-li však linku 10Gb/s a provoz bude vyšší než přibližně 2,5Gb/s, nebudou jednotky HFE schopny zpracovat příval dat a dojde k nekontrolovanému zahazování paketů (viz. obrázek 4.1).

- *přenos mezi kartami* – omezení se projeví v případě popsaném v předchozím bodě, tentokrát však s krátkými pakety na vstupu. Ty zvládá HFE zpracovat. Ze všech vytváří již zmíněnou unifikovanou hlavičku (délka hlavičky je stejná pro různě dlouhé pakety; pro nejkratší pakety je pak UH záznam delší než samotný paket). Počet vytvořených hlaviček je ale vyšší, než kapacita přenosového kanálu mezi kartami (viz. obrázek 4.1). Opět dochází k nekontrolované ztrátě dat; zahazování paketů.

- *kapacita a rychlost paměti* – k vyčerpání dostupné paměti dojde v okamžiku, kdy bude, v převážné většině po sobě jdoucích paketů, paket náležet novému/jinému síťovému toku. Tato situace však za normálních okolností nenastává. Naopak je to vlastnost jistých druhů útoků.

port	toky	pakety	provoz
test0p3000	6,2 k/s	146,3 k/s	859,8 Mb/s
test0p3001	6,4 k/s	132,8 k/s	786,3 Mb/s

Tabulka 4.1: Hodnoty odpovídající grafu na obrázku 4.2 v čase 12:00



Obrázek 4.1: Propustnost 10gigabitové verze sondy v závislosti na délce paketů

Na obrázku 4.2 můžeme vidět běžný síťový provoz. Obrázek doplňuje tabulka 4.1, v níž jsou údaje o počtu toků, paketů a celkovém provozu. Z těchto údajů vyplývá, že průměrný počet paketů na jeden tok je vyšší než 20. Na rozdíl od útoku, kde by se tato hodnota buď přímo rovnala jedné nebo se jedné blížila.

4.2 Útoky

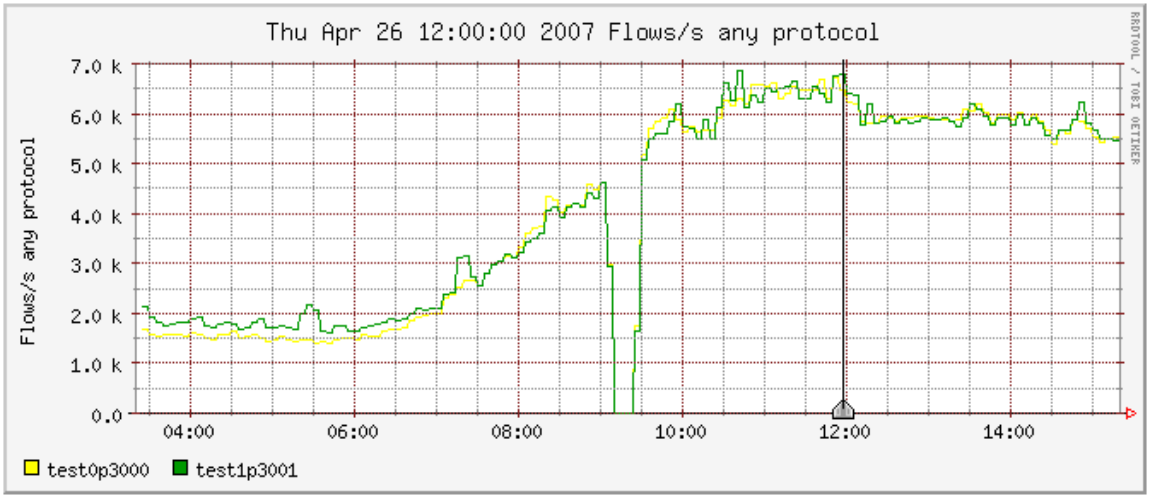
Během síťového útoku je provoz uměle vytvořen za určitým cílem. Existuje ještě řada dalších útoků mimo níže uvedené. Ty jsou ale ve většině případů zaměřeny na uživatelské služby a chyby v nich obsažené. Nevedou ke zvýšené zátěži sítě ani k velkému množství nově vznikajících síťových toků. Neovlivňují činnost sondy, a proto nebudou diskutovány.

4.2.1 DoS

Útok typu *DoS* (z angl. Denial Of Service), má za cíl přetížit vybraný objekt(služba, stroj) a znemožnit tak jeho funkčnost. Existuje celá řada technik jak tento útok vyvolat (viz. [7] a [8]).

Podívejme se na dva způsoby vedení útoku. Jeden s cílem zatížit síť na její plnou kapacitu. Druhý s cílem přetížit sondu *FlowMon*.

První z uvedených je obdobou nadměrného zatížení. Vzhledem k tomu, že se však jedná o útok, budou mít pakety tvar odpovídající nejhoršímu případu, tj. nejkratší možnou délku.



Obrázek 4.2: Počet toků v čase. Na port test0p3000 a test0p30001 je zapojen páteřní spoj dle obrázku 3.2. Snímek je pořízen z kolektoru *NfSen*.

V případě gigabitové linky sonda situaci zvládne (viz. obrázek 2.2). V případě 10gigabitové linky sonda bude nekontrolovaně zahazovat pakety (viz. obrázek 4.1).

Druhý typ útoku, zaměřený proti sondě, je založen na znalosti problému s uchováním informací o velkém množství toků. Provoz se tak bude skládat z nejkratších paketů. Každý paket bude náležet jinému síťovému toku. To povede k vyčerpání dostupné paměti pro uchování záznamů o toku. Sonda se bude snažit nuceně expirovat nejstarší záznamy. Tato operace je časově náročná a při plném vytížení vstupní části se stane úzkým místem. V takovémto případě bude sonda nekontrolovaně zahazovat pakety.

4.2.2 Skenování portů

Skenování portů charakterizují pokusy o vytvoření spojení na porty vybraného stroje. Tímto postupem útočník zjistí přítomnost spuštěných služeb, jejichž prostřednictvím se následně pokusí získat kontrolu, tj. přístup k datům, správcovská hesla. Přitom se snaží využít špatně zabezpečených nebo zcela nezabezpečených služeb.

Profil síťového provozu během tohoto útoku tak obsahuje velké množství paketů s různými cílovými porty. To znamená i velké množství síťových toků. A stejně jako u útoku DoS následuje zahlcení paměti a nekontrolované zahazování paketů.

4.3 Důležité poznatky

Jak vyplývá z provedené analýzy, dochází v mezních situacích k nekontrolovanému zahazování paketů. Největším negativem nekontrolovaného zahazování je nemožnost zpětně zjistit množství nezpracovaných dat. Zároveň tak není možné statisticky upravit získané výsledky a korigovat chybu vzniklou redukcí dat.

Řešení vede cestou řízené redukce dat. V části 2.6 byly vyjmenovány používané metody a také odvozen závěr, že vhodná metoda pro sondu *FlowMon* je vzorkování dat. Pokud jsou pakety zahazovány v daném pravidelném poměru, lze statistickými metodami určit původní množství dat. Získaný výsledek samozřejmě zatěžuje chyba, ta je však zanedbatelná

v porovnání se ztrátou informací v okamžiku nekontrolovaného zahazování, kdy ani nelze vzniklou chybu určit.

Redukce dat se stává nezbytnou především u 10gigabitového řešení, kdy ji potřebujeme i při ne zcela zatížené lince. Zlepšení situace může přinést jednotka s možností nastavit vzorkovací poměr (tj. poměr zpracovaných a zahozených paketů). Poměr by se pak odvodil od nejvyššího možného toku a propustnosti sondy.

Takové řešení je však suboptimální v okamžiku, kdy by nebyla linka plně vytížena (viz. část 3.3). Z tohoto pohledu se nejlepším přístupem jeví nastavovat vzorkování podle aktuálního zatížení a využití zdrojů sondy. Nastavení by tak mělo reflektovat okamžité vytížení monitorované linky (tj. počet paketů za sekundu), zatížení přenosového kanálu mezi kartami a současné zaplnění paměti.

Kapitola 5

Adaptivní vzorkovací jednotka

V předchozích kapitolách jsme uvedli teorii monitorování síťových toků, popsali architekturu sondy *FlowMon* a provedli analýzu chování sondy v mezních situacích. Získané závěry zhodnotíme při návrhu řešení, které má zabránit nekontrolovanému zahazování paketů.

5.1 Návrh

V průběhu vývoje sondy byla vytvořena jednotka umožňující pravidelné, náhodné vzorkování a vzorkování podle délky (namísto paketů jsou počítány byty; je–li napočítána daná hodnota, je paket obsahující tento byte zpracován). Tuto jednotku (dále vzorkovací jádro) lze programově nastavit. Tato operace je však časově náročná a vyžaduje interakci se softwarovou vrstvou, která vnáší velké zpoždění. Proto je vhodná pouze pro jednorázové nastavení. Jako výsledek dostaneme neoptimální vzorkování, které nereaguje na změny v monitorovaném provozu (viz. kapitola 4).

Potřebujeme vytvořit jednotku, která odstraní velké zpoždění při řízení programem. Přitom je třeba, aby nová hodnota měla platnost alespoň po nějaký časový interval. Rozdělíme si proto měřenou veličinu na řadu intervalů, ve kterých zůstane nastavení platné.

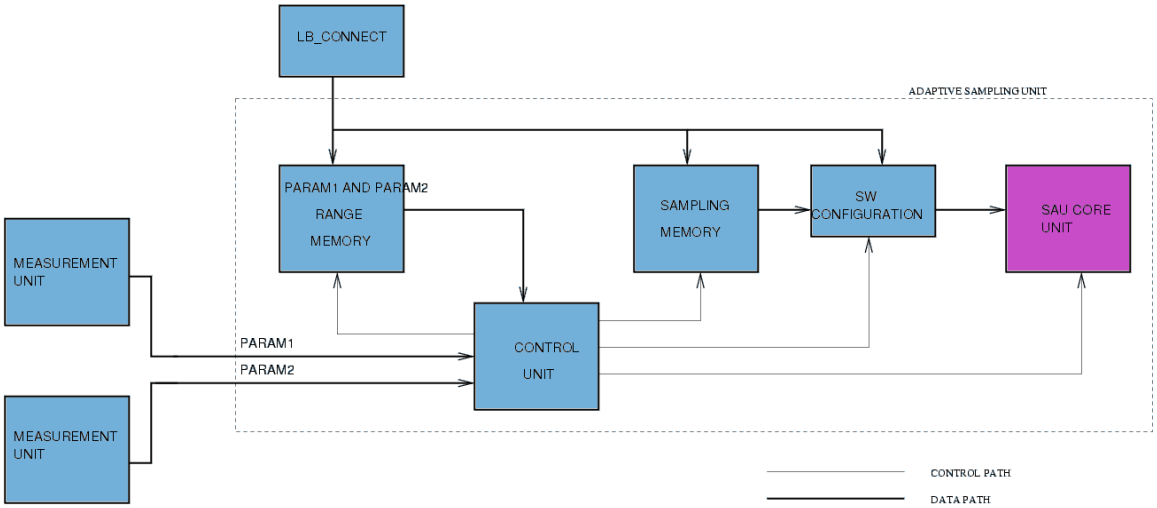
Navrhovaná jednotka pak v čase mění konfiguraci vestavěného vzorkovacího jádra. Chování řídí zadané hodnoty; jednotka je nejprve inicializována a na základě inicializace již autonomně řídí jádro v závislosti na vstupních parametrech.

Současně je umožněno přejít do režimu programového ovládání, kdy je kontrola plně na straně softwaru. Režim programového řízení se používá především pro ladící účely, neboť sebou nese neduhy zmíněné výše, tj. velkou odezvu, pomalé reakce.

Jako vstupní parametry jednotky slouží dvě 16bitové hodnoty. Pro každý vstupní parametr existuje měřící blok, který zajišťuje převod stavu zkoumaného jevu do informace v odpovídajícím tvaru. V prvním návrhu byl jeden měřící blok zaintegrován přímo v adaptivní vzorkovací jednotce. To však nebylo dostatečně obecné řešení. Přesunutím mimo jednotku vzniklo univerzální rozhraní umožňující vytvářet další měřící bloky. Tak se do budoucna zvýšila možnost pro úpravy, vylepšení a změny.

5.1.1 Blokové schéma

Na obrázku 5.1 vidíme blokové schéma návrhu. Popišme si nyní význam a funkci jednotlivých bloků :



Obrázek 5.1: Blokové schéma adaptivní vzorkovací jednotky

- *připojení ke sběrnici (lb_connect)* – jak bylo uvedeno výše, jednotka musí být inicializována. Inicializace je řízena programově, proto zde musí existovat možnost komunikace s registry a pamětmi umístěnými v jednotce. K tomuto účelu slouží právě popisovaný blok.

  Zprostředkovává komunikaci mezi systémem sběrnic navržených pro komunikaci uvnitř *FPGA*. To je pak svými vývody napojeno na další podpůrné obvody umožňující připojení k systémové sběrnici PCI/PCI–X/PCI–Express (záleží na typu karty, nyní je platná verze PCI–X).

- *param1 a param2* – vstupní signály. Na základě hodnoty signálů se provádí nastavení samotné vzorkovací jednotky.

- *paměť rozsahů pro param1 a param2 (param1 and param2 range memory)* – slouží pro uchování hodnot rozsahů pro jednotlivé vstupní signály. Vnitřně sestává ze dvou oddělených pamětí. Plnění probíhá ve fázi inicializace prostřednictvím bloku připojení ke sběrnici. Nastavené hodnoty je možno kdykoliv znovu inicializovat.

- *paměť vzorkovacích poměrů (sampling memory)* – udržuje zadané hodnoty poměrů zpracovaných a zahozených paketů; tvořen jednou pamětí. Stejně jako předchozí blok je připojen ke sběrnici a nastavován v době inicializace.

- *programová konfigurace (sw configuration)* – slouží k programovému ovládání vzorkovací jednotky prostřednictvím sady řídících a datových registrů. Stejně jako předchozí dva bloky připojen ke sběrnici.

- *výpočetní jednotka (control unit)* – vyhodnocuje příslušnost vstupních parametrů do intervalů zadaných v pamětech rozsahů pro param1 a param2. Podle výsledku inkrementuje/dekrementuje řádkový index společný pro všechny paměti (paměť rozsahů pro param1 a param2, paměť vzorkovacích poměrů). V případě potřeby pak vysílá posloupnost řídících signálů provádějících změnu nastavení vzorkovacího jádra.

- *vzorkovací jádro(sau core unit)* – vzorkuje vstupní impulsový signál v právě nastaveném poměru zvolenou metodou (ať už pravidelnou či náhodnou).

  Každý příchozí impuls odpovídá přijatému paketu. V návaznosti na přijatý impuls jádro nastaví/nenastaví odpovídající výstupní signál a rozhodne, jestli se paket zpracuje/zahodí.

- *měřící blok(measurement unit)* – zpracovává jev, na jehož základě je vzorkování řízeno, do podoby skalární hodnoty (16bitů). Vnitřní architekturou se mohou jednotlivé měřící bloky výrazně lišit. Například blok vyhodnocující aktuální průměrný počet paketů za sekundu bude zřejmě zcela odlišný od bloku poskytujícího informaci o zaplnění paměti toků.

5.1.2 Princip funkce

Samotné jádro adaptivní vzorkovací jednotky tvoří výpočetní jednotka. Skládá se ze čtyř 16bitových srovnávacích obvodů a vyhledávací tabulky (obsahem adresovaná paměť). Neustále vyhodnocuje porovnání vstupních parametrů s jejich příslušnými mezemi uloženými v pamětech (viz. obrázek 5.2) :

$$\begin{array}{lcl} param1 & < & \text{dolní limit pro param1} \\ param1 & > & \text{horní limit pro param1} \\ param2 & < & \text{dolní limit pro param2} \\ param2 & > & \text{horní limit pro param2} \end{array}$$

Výsledkem těchto porovnání dostáváme čtyři logické hodnoty. Konkatenací pak získáme 4bitový vektor. Vektor slouží jako vstupní adresa již zmíněné vyhledávací tabulky. Ta reprezentuje logickou funkci. Výstup logické funkce je 2bitový. Vzniká tedy mapování, kdy pro každý možný 4bitový vektor existuje odpovídající 2bitová hodnota. Tato hodnota ovládá čítač řádkového indexu (viz. obrázek 5.2)

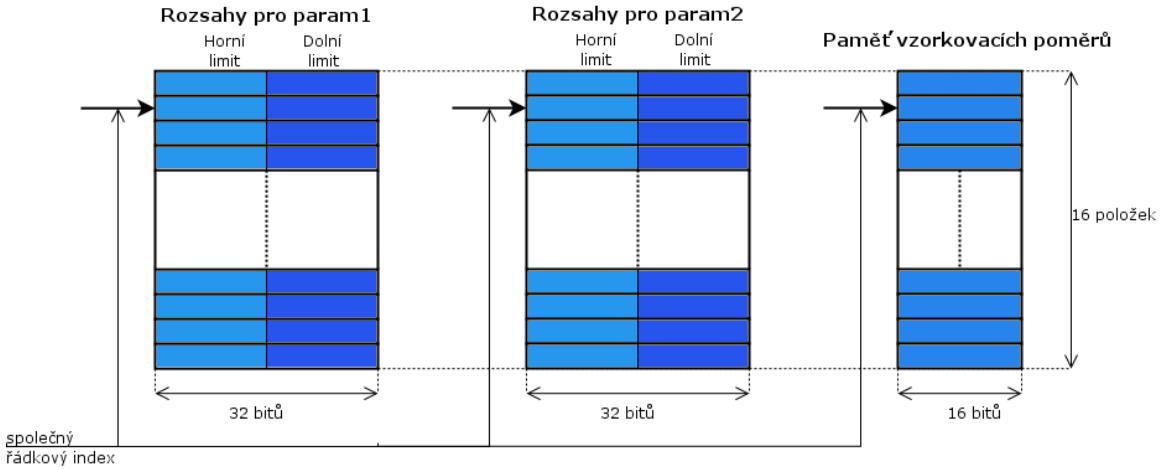
- *1. bit* pokud má hodnotu logická 1, bude se měnit hodnota čítače,
- *2. bit* logické úrovně říkají, kterým směrem (nahoru či dolů) se čítá.

Změna čítače řádkového indexu způsobí změnu rozsahů pro vstupní parametry a také vzorkovacího poměru. Pokud k takové události dojde, vyvolá řídící jednotka posloupnost signálů, které vedou k obnovení nastavení vzorkovacího jádra. Tato akce se opakuje pokaždé, dojde–li ke změně poměru; buď následkem posunu řádkového indexu nebo přepsáním hodnot v příslušné paměti (inicializací).

5.1.3 Měřící jednotky

Měřící bloky jsou integrální součástí adaptivní vzorkovací jednotky. Výše jsme popsali i jejich funkci, tj. snaží se ohodnotit stav nějakého děje skalární hodnotou. V našem případě vyjadřují počet paketů za sekundu nebo zaplnění paměti aj. Obecně však mohou popisovat libovolnou vlastnost sondy, která je nějakým způsobem reprezentovatelná skalární 16bitovou hodnotou.

Vychází z analýzy provedené v předchozí kapitole :



Obrázek 5.2: Detail pamětí použitých v adaptivní vzorkovací jednotce

- *průměrný počet paketů* – počítá aktuální průměrný počet paketů v uplynulém časovém intervalu. Intervaly jsou krátké a je tak zajištěna rychlá reakce na změny. Tento blok má za cíl umožnit sondě, pomocí změny vzorkovacího poměru, vypořádat se s nadměrným zatížením. Nastavení vzorkovacích poměrů odvodíme od zjištěné maximální propustnosti. Výsledkem získáme maximální využití sondy spolu s kontrolovaným zahazováním paketů.

- *zaplnění vyrovnávací paměti pro přenos mezi kartami* – také jeden z možných parametrů ovlivňujících úroveň vzorkování; v současné době však není pro přenos vyrovnávací paměť používána.

- *zaplnění paměti toků* – spolu s průměrným počtem paketů tvoří původně plánované vstupní parametry pro řízení adaptivního vzorkování.

  Podává informaci o zbývající kapacitě paměti a předchází tak jejímu vyčerpání změnou vzorkovacího poměru.

Prozatím je implementována pouze jedna měřící jednotka a to jednotka pro výpočet průměrného počtu paketů za sekundu. Při návrhu byl brán ohled na možnost měnit vlastnosti pomocí generických parametrů. Jednotka tak umožňuje v době překladu nastavit časový interval z něhož se určuje průměrný počet paketů. Dále lze měnit počet použitých čítačů a ovlivnit tak množství zabraných zdrojů, ale především rychlost aktualizace měřené hodnoty. Čítače jsou rovnoměrně překryty a proto se údaj aktualizuje po uplynutí doby $\frac{\text{měřený interval}}{\text{počet čítačů}}$.

5.2 Implementace

Vzhledem k tomu, že hardwarovou platformou pro realizaci sondy je programovatelné pole (*FPGA*), byl zvolen jazyk VHDL. V něm lze vytvářený systém definovat na úrovni strukturální, kdy se popíše propojení jednotlivých stavebních prvků, nebo behaviorální, kdy se naopak popíše chování architektury. V případě adaptivní vzorkovací jednotky se jedná o behaviorální popis.

Implementačním problém se vyskytl při synchronizaci paměti rozsahů s rozhodovací funkcí. Při čtení dat z paměti rozsahů dochází ke zpoždění, následně i při porovnání se vstupními parametry. Teprve výsledek porovnání slouží jako vstup logické funkce. Po dobu než jsou data vyčtena a provedeno porovnání je třeba blokovat čítač řádkového indexu.

Problém se stupňuje se snahou zvýšit maximální frekvenci synchronizačních hodin. Popsaná cesta od pamětí rozsahů po rozhodovací funkci tvoří nejdelší logickou cestu. Problém vyřešíme vložením registru do cesty a umožníme tak zpracování ve více krocích. Zkrátí se délka cesty a zvýší maximální použitelná frekvence hodin. Na druhou stranu se zvětší zpoždění a je nutno prodloužit dobu, po kterou musí být čítač řádkového indexu blokován.

Naštěstí tyto prodlevy nejsou kritické. Změna vstupních parametrů trvá výrazně delší dobu (samotný příjem paketu trvá déle). Zároveň ale negativně působí na dobu, po kterou se jednotka ustaluje v okamžiku spuštění (inicializaci).

5.3 Simulace

Nezbytným krokem při realizaci této práce je simulace chování adaptivní vzorkovací jednotky a také analýza vlivu na zbytek sondy. Pomocí simulace jsme schopni odhalit zda navržené řešení splňuje kladené požadavky.

5.3.1 Simulační model

Paměť mezí pro param1		**Vzorkovací poměr**
horní mez	dolní mez	
0	50	0
48	100	1
98	150	2
148	200	3
198	250	4
248	300	5
298	350	6
348	400	7
398	450	8
448	500	9
498	550	10
548	600	11
698	650	12
648	700	13
698	750	14
748	800	15

Tabulka 5.1: Nastavení adaptivního vzorkování odpovídající grafu z obrázku 5.3

Simulační model využívá rozšíření jazyka C++, konkrétně knihovny SIMLIB (viz. [11]). Nástroj je vhodný jak pro spojitou tak i pro diskrétní simulaci. V našem případě se jedná o diskrétní model řízený synchronizačními impulsy (odpovídají hodinovému signálu v obvodovém řešení).

Model popisuje základní bloky uvedené v blokovém schéma (viz. obrázek 5.1). Neobsahuje však jednotku pro připojení ke sběrnici, která je v programovacím jazyce zastoupena inicializací proměnných.

Použité nastavení adaptivní vzorkovací jednotky ukazují tabulky 5.1 (meze vstupních parametrů a odpovídající vzorkovací poměry) a 5.2 (logická funkce).

Aby simulační model odpovídal současné vývojové fázi jednotky a měřících bloků, využívá pouze jeden vstupní parametr. Proto uvedená tabulka rozsahů obsahuje pouze jeden sloupec pro param1 a zároveň se zjednodušila řídící funkce, která reaguje pouze na hodnoty prvního parametru; hodnoty druhého zanedbává.

vstup	**měnit**	**nahoru**	—	**vstup**	**měnit**	**nahoru**
0000	false	false	—	1000	true	false
0001	false	false	—	1001	true	false
0010	false	false	—	1010	true	false
0011	false	false	—	1011	true	false
0100	true	true	—	1100	false	false
0101	true	true	—	1101	false	false
0110	true	true	—	1110	false	false
0111	true	true	—	1111	false	false

Tabulka 5.2: Logická funkce odpovídající grafu z obrázku 5.3

Logická funkce (viz. tabulka 5.2) poskytuje široké prostředky pro nastavení chování. Tím, že jsme schopni pro každou vstupní kombinaci definovat výstupní hodnotu, lze dosáhnout prakticky libovolného chování. Zřejmě nejzajímavější se jeví varianta, kdy dáme prioritu řízení jednomu ze dvou vstupních parametrů. Druhý pak slouží jako doplňkové řízení v případě, kdy první nevyžaduje a naopak povoluje změny.

Vzniká tak možnost reflektovat důležitost jednotlivých omezujících vlivů do nastavení adaptivní vzorkovací jednotky. Prioritu by získal parametr vypovídající o zaplnění paměti a jako doplňující by byl využit průměrný počet paketů.

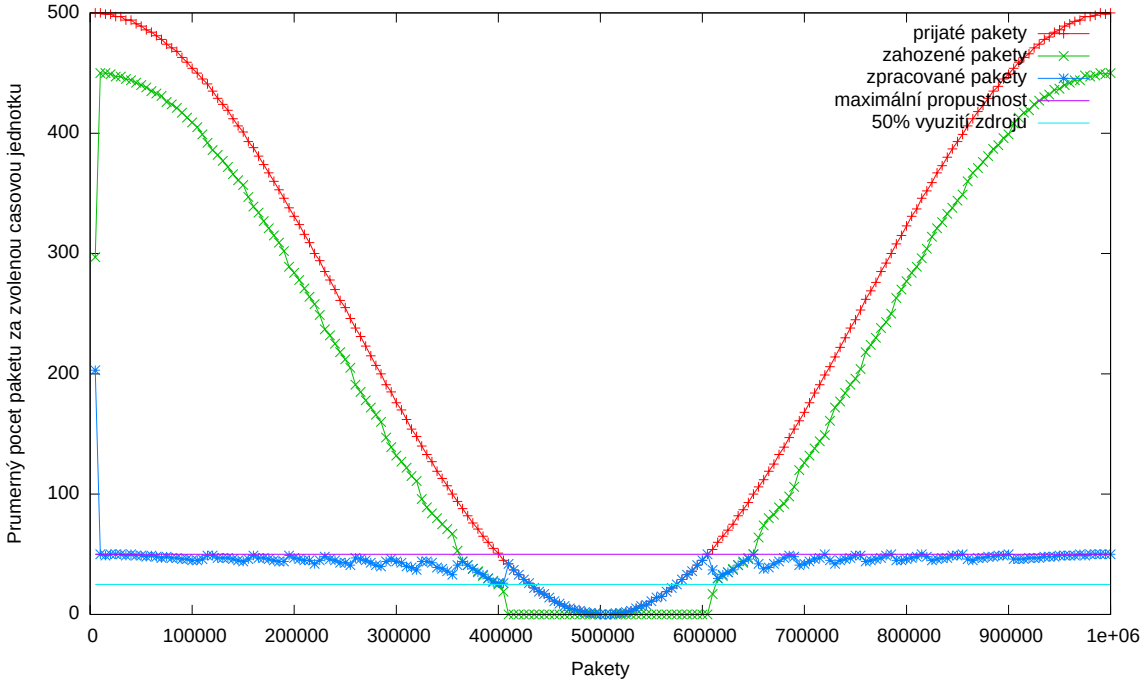
Alternativní varianta : oba parametry mohou samostatně zvýšit vzorkovací poměr bez ohledu na druhý. V opačném případě, tj. snížení poměru, dojde ke změně jen při shodě obou parametrů. Oba nabývají stejné váhy a jedná se tak o spravedlivý přístup na místo prioritního v prvním případě.

5.3.2 Výsledky získané simulací

Výsledky získané simulací zobrazuje graf na obrázku 5.3. Rozložení příchodu paketů udává funkce $cos(x)$ posunutá nad osu x. Uprostřed zobrazeného intervalu má minimální hodnotu. Dojde tedy ke dvěma průnikům s přímkou udávající maximální zatížení. Jednou jako funkce klesající, podruhé rostoucí. Právě proto byla vybrána; umožňuje sledovat chování v okamžicích, kdy je třeba začít/přestat vzorkovat pakety.

Maximální hodnoty funkce příchodu paketů nabývá přibližně desetinásobku maximálního zatížení. Toto je opět požadovaná vlastnost a ukazuje, jak by se sonda chovala při použití adaptivní vzorkovací jednotky na 10Gb/s lince (sonda bezpečně zvládá 1GB/s).

V těsné blízkosti svislé osy vidíme velké množství paketů označených za zpracované. Množství však výrazně přesahuje přes hladinu maximální propustnosti. Tyto pakety by v reálném provozu byly nekontrolovatelně zahozeny. Jedná se však pouze o okamžik spuštění



Obrázek 5.3: Graf vyjadřující počet přijatých, zpracovaných a zahozených paketů

sondy, kdy čelí velkému síťovému provozu. Situace se rychle ustálí a k dalším obdobným výkyvům nadále nedochází.

Popsaná negativní vlastnost se projeví pouze v případě spuštění nebo opakované inicializace adaptivní vzorkovací jednotky. Navíc se jednotka přizpůsobí velmi rychle, proto popsané chování nebrání použití v reálné aplikaci.

Velmi důležitou součástí grafu je i přímka udávající hladinu 50% využití sondy. Požadované minimální zatížení lze chápat jako vstupní proměnnou při výpočtu mezí pro jednotlivé parametry. V grafu vidíme, že počet zpracovaných paketů neklesne pod tuto hranici pokud samotný monitorovaný provoz nepoklesne. Jinými slovy můžeme říct, že využití sondy je vždy nejméně 50%. Tato hranice lze nastavit i vyšší podle potřeby. Minimalizuje se tak množství paketů zahozených v důsledku vzorkování a udržuje se optimální zatížení sondy.

5.4 Ověření funkce

Adaptivní vzorkovací jednotka se používá jak v 1GB/s verzi tak 10GB/s verzi sondy *FlowMon*. V reálných nasazeních využíváme prozatím softwarového řízení vzorkování. Ve zkušebním provozu však již byla jednotka testována a vykazovala požadovanou funkcionalitu.

Zkušební provoz byl zachycen nástrojem *tcpdump* a do sondy odeslán pomocí příkazu *tcpreplay*. Vše proběhlo na jednom z pracovních počítačů projektu *Liberouter* osazeného kartami *COMBO6X* a *COMBO-4SFPRO*.

Příkaz tcpreplay umožňuje nastavit rychlost odesílání zachyceného vzorku dat. Tak bylo možné srovnat reakce vytvořené jednotky se skutečností.

Po zahájení přenosu paketů z pomocného stroje zapojeného proti sondě *FlowMon* byly

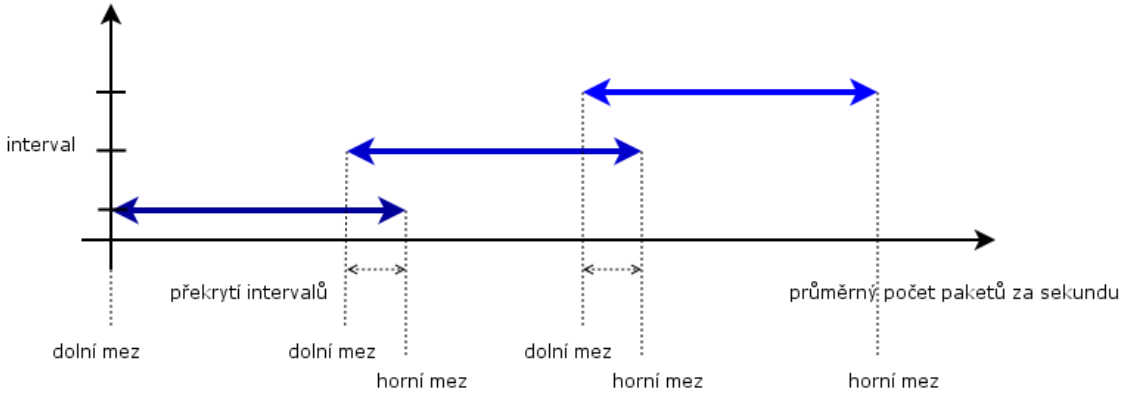
vyčítány aktuální hodnoty ladících registrů, které prokázaly shodu s předpokládanými hodnotami. Chování odpovídalo jak simulacím modelu vytvořeného v jazyce C++, tak simulacím samotné implementace v jazyce VHDL.

5.5 Diskuze

Velký prostor pro práci s jednotkou nabízí její konfigurační možnosti, kdy můžeme provést řady analýz a testů optimálních vzorkovacích poměrů a odpovídajících rozsahů. Neméně důležité pak bude nastavení měřících bloků (mohou významným způsobem změnit reakce sondy).

Na druhou stranu přináší velká variabilita prostor pro vznik nevhodných a špatně fungujících natavení. V průběhu vývoje bylo při simulacích VHDL kódu zjištěno, že může například dojít k neustálému přepínání mezi dvěma vzorkovacími poměry. K takovému jevu dochází v případě, kdy mají stejnou váhu oba vstupní parametry. Jinými slovy je nevhodným způsobem sestavena řídící logická funkce.

Doporučením pro návrh mezí je vytvořit překrytí jednotlivých intervalů. Nastavení se tak stane více odolným proti oscilaci vstupního parametru kolem jedné z mezí. Příklad takového překrytí lze spatřit i v tabulce 5.1. Tam vidíme překrytí jen minimální – vychází ze znalosti průběhu rozložení paketů (viz. obrázek 5.3). Pro reálné použití se doporučuje dodržet překrytí, které názorněji ukazuje obrázek 5.4.



Obrázek 5.4: Ukázka vhodného překrytí mezí nastavených pro vstupní parametry

Pro správnou funkci je nezbytné korektní nastavení. Při splnění tohoto předpokladu bude adaptivní vzorkovací jednotka plnit svůj účel, tj. zamezovat nekontrolovanému zahazování paketů a zároveň minimalizovat množství zahozených paketů. Je třeba připomenout nekontrolované zahazování, které nastává po spuštění jednotky při velkém provozu. Ale jak již bylo řečeno, jedná se o velmi krátký jev, a proto lze ve výsledku zanedbat.

Dalším krokem ve vývoji by mělo být vytvoření nových měřících bloků. Důležité bude zejména vytvořit blok pro zpracování aktuálního stavu paměti toků.

Kapitola 6

Závěr

Cílem této práce bylo vytvořit jednotku adaptivního vzorkování, která bude zároveň minimalizovat množství paketů zahozených v důsledku vzorkování. Jednotka byla implementována, odzkoušena v hardwaru, a tudíž byl cíl splněn.

Pro úspěšnou realizaci byla nastudována problematika síťových toků, metody redukce dat a druhy síťových útoků. Tyto jsou popsány v druhé kapitole s názvem *Počítačové sítě a monitorování síťových toků*.

Jednotka byla vytvářena jako součást sondy *FlowMon*, proto bylo třeba se seznámit s architekturou sondy. Architekturu popisuje třetí kapitola nazvaná *Architektura sondy FlowMon*. Jako výchozí bod pro samotnou implementaci posloužila analýza chování sondy v krajních případech, jakými jsou útoky a nadměrné zatížení (kapitola *Analýza chování sondy*). Návrh a následná implementace byla provedena s ohledem na vysokou variabilitu použití. Popis návrhu, implementace, simulace i ověření funkce obsahuje kapitola *Adaptivní vzorkovací jednotka*.

Největší předností adaptivní vzorkovací jednotky pak jsou téměř neomezené možnosti měnit nastavení hardwaru bez nutnosti změn samotné obvodové realizace. Zároveň poskytuje nízkoúrovňové řešení vysoký výkon a rychlou odezvu na změny.

Možnosti nasazení jsou široké. Při použití adaptivní vzorkovací jednotky lze provozovat sondu jak na běžných spojích lokální úrovně, kdy vhodné nastavení zajistí, že se pakety nebudou zahazovat, tak i na páteřních spojích, kdy bude jednotka měnit vzorkovací poměry podle aktuálního zatížení a využití zdrojů.

Adaptivní vzorkování paketů zvyšuje náskok sondy *FlowMon* před konkurenčními softwarovými nástroji. Především v okamžicích útoků a nadměrné zátěže. Současně minimalizuje množství řízeně zahozených paketů a udržuje tak vysokou kvalitu nabízených služeb.

Dovoluji si tedy říct, že práce splnila svůj účel a přinesla řadu zajímavých poznatků, které budou zhodnoceny zejména ve fázi nasazení do skutečného provozu. Přispěje nejen k lepšímu přehledu o používaných síťových službách, ale také ke zvýšení bezpečnosti počítačových sítí.

Do budoucna by měla být jednotka doplněna o další měřící bloky. Zároveň by práci s jednotkou usnadnil nástroj, který by na základě zadaných parametrů sítě a požadovaného využití sondy připravil kompletní konfiguraci jednotky. Případně i složitější regulační systém, který by v pravidelných intervalech hodnotil kvalitu aktuálního nastavení a přizpůsoboval jej potřebám uživatele.

Literatura

[1] Netflow Overview. [online], Leden 2004, [cit. 2007-04-20].
URL http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/switch_c/xcprt3/xcovntfl.htm

[2] Netflow Performance Analysis. [online], 2005, [cit. 2007-04-20].
URL http://www.cisco.com/application/pdf/en/us/guest/netsol/ns583/c654/cdccont_0900aecd80308a66.pdf

[3] FlowMon monitoring probe project. [online], 2006, [cit. 2007-04-23].
URL
http://www.liberouter.org/vhdl_design/generated/HEAD_FLOWMON_hw.php

[4] Netflow Version 9 Flow-Record Format. [online], Únor 2007, [cit. 2007-04-18].
URL http://www.cisco.com/en/US/products/ps6601/products_white_paper09186a00800a3db9.shtml

[5] Počítačová síť. [online], Duben 2007, [cit. 2007-05-02].
URL http://cs.wikipedia.org/wiki/Po%C4%8D%C3%ADta%C4%8Dov%C3%A1_s%C3%AD%C5%A5

[6] Duffield, N.: *Sampling for Passive Internet Measurement : A Review*. Institute of Mathematical Statistics, 2004.

[7] Haller, M.: Denial of Service útoky: reflektivní a zesilující typy. [online], Říjen 2006, [cit. 2007-04-25].
URL http://www.lupa.cz/clanky/denial-of-service-utoky-reflektivni-a-zesilujici-typy/

[8] Haller, M.: Denial of Service útoky: záplavové typy. [online], Září 2006, [cit. 2007-04-25].
URL
http://www.lupa.cz/clanky/denial-of-service-dos-utoky-zaplavove-typy/

[9] Ivanko, J.: *FlowMon testing*. Liberouter, Duben 2007.

[10] Kállay, F.; Peniak, P.: *Počítačové sítě a jejich aplikace*. Praha: Grada Publishing, 1999, ISBN 80-7169-407-X, 312 s., recenzenti Zavoral, M.; Jeger, D.

[11] Peringer, P.: SIMLIB. [online], Listopad 2006.
URL http://www.fit.vutbr.cz/~peringer/SIMLIB/index.html.cz

[12] Quittek, J.; Zseby, T.; Claise, B.; aj.: *Requirements for IP Flow Information Export*. Network Working Group, Říjen 2004.
URL http://www.ietf.org/rfc/rfc3917.txt

[13] Veiga, H.; Pinho, T.; Oliveira, J. L.; aj.: *Active traffic monitoring for heterogeneous environments*. University of Aveiro, Institute of Telecommunications, Campus Santiago, Aveiro, Portugal.
URL http://www.av.it.pt/jowamp/index_files/ICN05_J-OWAMP_paper.pdf

[14] Čeleda, P.; Kováčik, M.; Koníř, T.; aj.: FlowMon probe. [online], Prosinec 2006, [cit. 2007-04-23].
URL http://www.cesnet.cz/doc/techzpravy/2006/flowmon-probe/