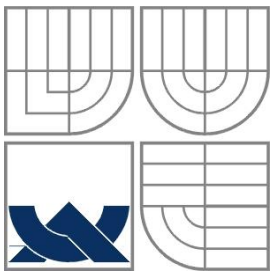
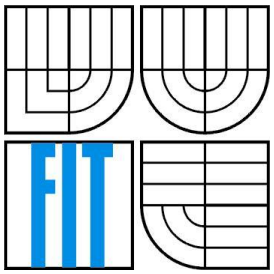


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

# DETEKCE SÍŤOVÝCH ANOMÁLIÍ ZALOŽENÁ NA PCA

DETECTION OF NETWORK ANOMALIES BASED ON PCA

BAKALÁŘSKÁ PRÁCE
BACHELOR‘S THESIS

AUTOR PRÁCE
AUTHOR

PAVEL KROBOT

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. VÁCLAV BARTOŠ

BRNO 2013

# Abstrakt

Tato práce se zabývá detekcí anomálií v počítačových sítích. Konkrétní metoda, jež bude dále popsána, je založena na analýze hlavních komponent. V rámci této práce byl studován původní návrh této metody a jeho další dvě rozšíření. Základní verze a poslední rozšíření pak byly implementovány společně s jedním dalším malým rozšířením, které bylo navrženo v rámci této práce. Nad výslednou implementací byla následně provedena série testů. Tyto testy přinesly dva hlavní poznatky. První z nich poukazuje na možnou použitelnost analýzy hlavních komponent pro detekci anomálií v síťovém provozu. Druhý pak poznamenává, že přestože se metoda v jistých ohledech ukázala jako funkční, je ještě nedokonalá a je potřeba dalšího výzkumu pro její vylepšení.



# Abstract

This thesis deals with subject of network anomaly detection. The method, which will be described in this thesis, is based on principal component analysis. Within the scope of this thesis original design of this method was studied. Another two extensions of this basic method was studied too. Basic version and last extension was implemented with one little additional extension. This one was designed in this thesis. There were series of tests made above this implementation, which provided two findings. First, it shows that principal component analysis could be used for network anomaly detection. Second, even though the proposed method seems to be functional for network anomaly detection, it is still not perfect and additional research is needed to improve this method.



# Klíčová slova

Detekce anomálií, detekce útoků, podprostorová metoda, analýza hlavních komponent, PCA, bezpečnost počítačových sítí

# Keywords

Anomaly detection, detection of attacks, subspace method, multiway subspace method, sketch subspace, principal component analysis, PCA, network security


# Citace

Pavel Krobot: Detekce síťových anomálií založená na PCA, bakalářská práce, Brno, FIT VUT v Brně, 2013.

# Detekce síťových anomálií s využitím NetFlow dat

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Václava Bartoše a uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……

Pavel Krobot

14. 5. 2013

## Poděkování

Rád bych poděkoval zejména Ing. Václavu Bartošovi za odbornou pomoc, dále pak rodině a přátelům za podporu.

# Obsah


1 Úvod ... 3
2 Útoky a jiné anomálie v počítačových sítích ... 4
2.1 Denial of Service (DoS) ... 4
2.1.1 Záplavové (Flood) útoky ... 5
2.1.2 Útoky využívající chyb a vyčerpání systémových prostředků ... 5
2.1.3 Distribuované DoS útoky (DDoS) ... 5
2.1.4 Reflektivní útoky (DRDoS) ... 6
2.2 Skenování portů ... 7
2.3 Skenování sítí ... 8
2.3.1 Červy ... 8
2.4 Alfa toky (alpha flows) ... 8
2.5 Přeplnění vyrovnávací paměti a výpadky ... 8
3 Protokol NetFlow a toky v počítačových sítích ... 10
4 Detekce anomálií v počítačových sítích ... 11
4.1 Analýza hlavních komponent ... 11
5 Popis metody detekce založené na PCA ... 13
5.1 Detekce objemových anomálií ... 13
5.1.1 Vstupní data ... 14
5.1.2 Použití PCA ... 14
5.1.3 Detekce ... 16
5.1.4 Identifikace ... 16
5.2 Detekce založená na analýze rozložení rysů provozu ... 17
5.2.1 Rozložení rysů provozu ... 18
5.2.2 Vstupní data a detekce ... 21
5.3 Detekce využívající náhodné agregace IP toků ... 21
5.3.1 Vstupní data, vytváření náhodných uskupení a detekce ... 22
5.4 Předzpracování dat ... 23
6 Návrh a implementace aplikace ... 24
6.1 Struktura programové soustavy ... 24
6.2 Zpracování vstupních dat ... 25
6.2.1 Mód načítání dat z objemu provozu ... 26
6.2.2 Mód načítání NetFlow dat ... 26
6.2.3 Předzpracování dat ... 27
6.3 Detekce anomálií ... 27

6.4 Identifikace anomálií ..... 28
7 Vyhodnocení výsledků ..... 30
7.1 Testovací data ..... 30
7.2 Metodika testování a způsob vyhodnocení ..... 30
7.3 Výsledky objemové varianty ..... 31
7.4 Výsledky varianty náhodných uskupení ..... 34
8 Závěr ..... 35

# 1 Úvod

Počítačové sítě dnes zasahují do mnoha oblastí každodenního života. S největší a nejznámější z nich, s Internetem, se každý den setkává mnoho lidí, ať už za účelem práce, získávání informací, komunikace s přáteli, zábavy či jiných. Některé z těchto možností využití jsou pak nezbytnou součástí běžného života lidí, chodu firem nebo třeba státní infrastruktury. Vedle veřejného Internetu je zde pak velké množství sítí privátních, které mohou fungovat izolovaně nebo mohou být prostřednictvím brány opět připojeny do Internetu. Popularita počítačových sítí a Internetu navíc stále roste.

Spolu s tímto nárůstem popularity však roste i množství útoků. Útoky v počítačových sítích mohou mít různé cíle. Někdy může jít jen o touhu útočníka „něco si dokázat“, jindy může jít o pomstu nebo nenávist k cíli útoku vyjádřenou znepřístupněním cílem poskytované služby. Stejný typ útoku může být i jistou formou demonstrace. Dále se pak může jednat o útoky s účelem osobního obohacení či likvidace konkurence. Důvodů k útoku je zkrátka mnoho a někdy mohou být v sázce velmi citlivé informace nebo dokonce lidské životy.

Z tohoto důvodu je bezpečnost počítačových sítí velmi důležitým a aktuálním tématem. Zabezpečení dat, ať už těch uložených na serverech nebo těch, která sítí prochází, dostupnost různých služeb a linek počítačových sítí – to jsou hlavní aspekty zabezpečení počítačových sítí. Počítačové útoky se stále vyvíjejí a jsou více a více sofistikovanější. Navíc narůstá kapacita linek a s tím i množství přenášených dat, což z hlediska zabezpečení síťového provozu znamená hledání menší „jehly“ ve stále větší „kopě sena“.

Jak je vidět počítačové sítě a zejména pak Internet jsou součástí každodenního života mnoha z nás. Poskytují nám zábavu, usnadňují nám práci a jiné běžné činnosti jako je třeba nakupování či bankovnictví. Bohužel stejně jako v běžném životě i na Internetu existují lidé, kteří se snaží ostatní lidi okrást či jim znepříjemnit život. Tato práce se zabývá jednou z oblastí bezpečnosti počítačových sítí, a to hledáním anomálií v síťovém provozu. K dosažení tohoto cíle je použita metoda, kterou popsali Anukool Lakhina a další v [9]. Tato metoda hledá anomálie v rámci celé rozlehlé počítačové sítě, nikoliv jen na jedné lince. Jejím základem je rozdělení provozu z více linek, který tvoří prostor o velkém počtu dimenzí, na dva podprostory – jeden normální a druhý anomální. Tato metoda bude, po stručném představení některých počítačových útoků v kapitole 2, základních termínech používaných v rámci popisované metody v kapitole 3 a způsobech detekce síťových anomálií v kapitole 4, popsána v 5. části této práce. V této kapitole jsou také popsána další rozšíření, čerpající z [10] a [11]. Dále je v kapitole 6 popsán návrh a implementace aplikační soustavy, která byla v rámci této práce na základě tří výše uvedených článků vytvořena. V kapitole 7 pak budou popsány provedené testy, jež byly provedeny díky této implementaci a zhodnoceny jejich výsledky.

# 2 Útoky a jiné anomálie v počítačových sítích

V počítačových sítích se dnes můžeme setkat s velkým množstvím různorodých anomálií. Některé z nich mohou vznikat přirozeně, bez cizího záměru (např. výpadek zařízení). Jiné však mohou být cílenými útoky, pomocí kterých se snaží útočník dosáhnout svého cíle. Při detekci síťových anomálií je potřeba znát všechny druhy síťových anomálií – jednak ty, které vznikají běžným používáním počítačových sítí a nemusí znamenat velkou (nebo dokonce žádnou) hrozbu, jednak také ty, které jsou hrozbou a proti kterým je potřeba rychlá a efektivní obrana.

V následujících podkapitolách budou popsány ty nejběžnější, které jsou v souvislosti s detekcí anomálií, pomocí dále popsané metody, relevantní. Popis jednotlivých anomálií je zaměřen zejména na jejich projevy v sítích a na jejich cíle, pakliže se jedná o útoky. Popsány jsou následující anomálie: DoS útoky (Denial of Service – odmítnutí služby), skenování portů a sítí, červy, alfa toky a různé výpadky. Společné pak pro tyto anomálie je to, že se nějakým způsobem odlišují od běžného provozu na síti, ať už zvýšením objemu přenášených dat nebo třeba nárůstem paketů se specifickými hodnotami v hlavičkách IP paketů. Zde bych zavedl termín, který budu dále používat a který vznikl volným překladem z původního anglického článku [10], popisujícího metodu, jenž je předmětem této práce. Jedná se o termín *rys provozu* (z anglického *traffic feature*). *Rys provozu* je tedy údaj z pole hlavičky IP paketu. V rámci této práce se pak jedná o pole zdrojové a cílové IP adresy (značeno zIP a cIP) a zdrojového a cílového portu (zPort a cPort) [10].

## 2.1 Denial of Service (DoS)

Tento útok, jak už sám název napovídá, si klade za cíl znepřístupnění nějaké služby, počítače, síťového segmentu aj. (v textu dále budu uvádět jen služby, jsou tím ale myšleny všechny tyto případy). Záměry tohoto útoku mohou být různé, může se jednat pouze o „žert“ mezi kamarády, dále může být podnětem k útoku vztek či nenávist vůči danému cíli útoku nebo může jít třeba o zviditelnění sebe sama. Důvodů je mnoho, a jelikož některé typy útoků z této kategorie nevyžadují mnoho znalostí z oboru a kvůli různým nástrojům ani mnoho šikovnosti, jsou tyto útoky velice populární. Navíc jimi lze způsobit cíli útoku značné škody.

Za nejtypičtější útok v této kategorii lze považovat zahlcení cíle útoku velkým objemem dat, čímž je znemožněn přístup dalších legitimních uživatelů, požadujících danou službu. Dnes se však v této kategorii vyvinulo mnoho jiných způsobů jak dosáhnout požadovaného záměru – znepřístupnění služby. Dnes jsou více na popředí typy DoS útoků, které se zaměřují na chyby v jednotlivých aplikacích a programech (třeba i v operačních systémech) [4][12].

DoS útoky se v síťovém provozu projevují zvýšením koncentrace cílové IP adresy směrem k útočníkovi a také změnou v rozložení zdrojových IP adres [10]. V následujících podkapitolách budou popsány některé z typů DoS útoků.

### 2.1.1 Záplavové (Flood) útoky

Tyto útoky patří mezi nejprostší. Útočník je jeden a jeho snaha spočívá ve vygenerování co největšího toku tak, aby zahltil linku oběti. Obrana proti těmto typům útoku je náročná. Avšak v dnešní době, kdy jsou služby ve většině případů připojeny do sítě pomocí linek, které disponují výrazně větším přenosovým pásmem, než kterým je připojen běžný uživatel (a tedy i útočník), nejsou tyto útoky příliš aktuální. Stejného efektu lze však dosáhnout za použití distribuovaného DoS útoku (DDoS), který bude popsán dále.

Mezi nejznámější typy útoků, spadající do této kategorie patří: ICMP flood, UDP flood a TCP flood. Záplavové TCP útoky (TCP flood) lze pak ještě dále rozdělit podle použitých příznaků hlavičky TCP, z nichž nejznámější je útok SYN flood [4].

### 2.1.2 Útoky využívající chyb a vyčerpání systémových prostředků

Typy útoků spadající do této podkategorie DoS útoků, využívají zranitelnosti v software nebo hardware oběti. Jedná se většinou o různé chyby, které bývají často rychle opraveny, a proto konkrétní útoky z této kategorie nemají dlouhou životnost. Oproti záplavovým útokům však nejsou náročné na prostředky. Útok totiž většinou spočívá v zaslání malého množství speciálních paketů, které cíl útoku znepřístupní. Z tohoto důvodu je také těžké detekovat tento útok z jiného místa, nežli na zařízení oběti (stačí jen malé množství paketů). Další nebezpečí těchto útoků spočívá v přístupu výrobců (hardware nebo software) k chybám v jejich produktech, kdy daný výrobce s opravami těchto chyb nijak nespěchá.

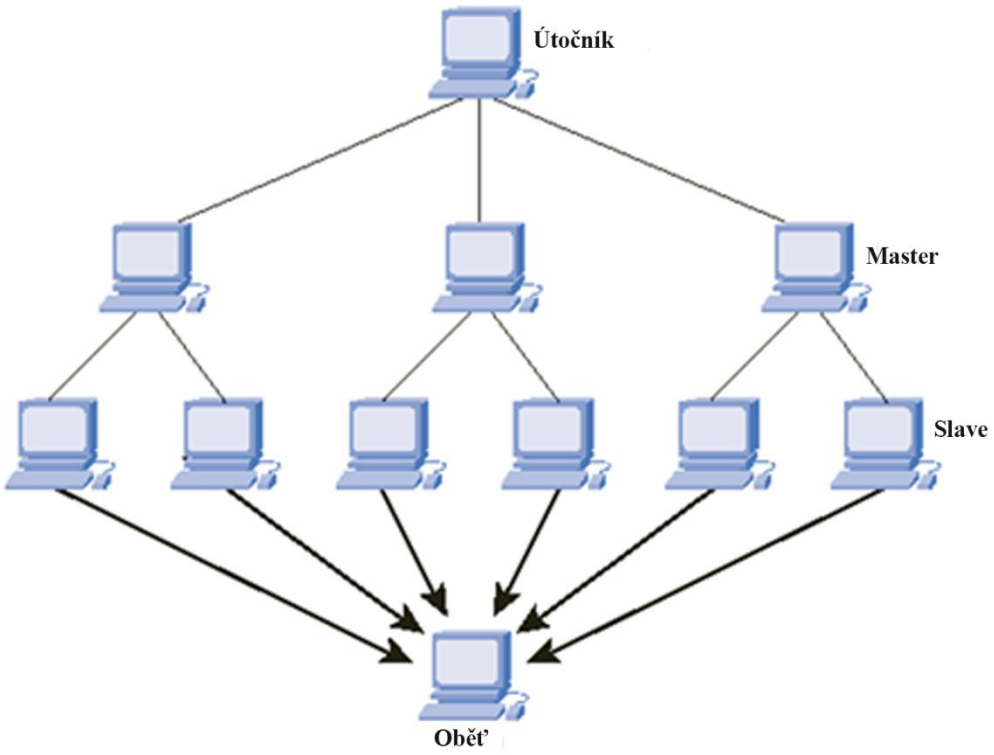
Trochu specifičtějšími útoky z této kategorie jsou útoky vyčerpávající systémové prostředky. Tyto útoky využívají implementačních chyb, tedy chyb způsobených špatným návrhem. Tyto chyby se mohou projevovat různě, např. nějaký program začne spotřebovávat více paměti či více vytíží procesor při obdržení určitého paketu. Zde je pak zapotřebí většího množství paketů, které musí být zasílány po celou dobu útoku. Zástupci této kategorie jsou např. útoky Teardrop (Slza) nebo dnes již neaktuální, přesto dobře známý Ping of death (Ping smrti) [4].

### 2.1.3 Distribuované DoS útoky (DDoS)

Distribuované DoS útoky lze jednoduše popsat jako DoS útoky prováděné více než jedním útočníkem. Dříve tomu tak skutečně bývalo – útok probíhal pomocí několika (stovek i tisíců) spolupracujících útočníků, tyto útoky se však také dále vyvíjely a dnes je znám i typ DDoS útoku, kdy skutečný útočník je pouze jeden. Stále se však jedná o distribuovaný útok, protože útočících stanic je velké množství.

Tento typ DDoS útoku začíná přípravnou fází, kdy si útočník nejdříve vybuduje síť stanic, které budou pro útok použity. Pro vytvoření této sítě útočník vyhledává zranitelné stanice. To mohou být např. nezabezpečené počítače nebo stanice se zastaralým či neaktualizovaným software. Jakmile získá útočník přístup na takovouto stanici, začne na ní instalovat potřebné nástroje pro provedení útoku. Těmto napadeným stanicím se říká zombie. Vzhledem k tomu, že jsou dostupné nástroje, které tuto fázi obstarají automaticky – tedy vyhledávají zranitelné stanice, instalují na ně potřebné nástroje a navíc se pak dále šíří z těchto stanic na další – může být rozsáhlá síť útočníků (zombie) vytvořena velmi rychle. Ve výsledku se pak tato síť pro útok skládá ze dvou typů zombie stanic – handler

(master) a agent (slave). Dokonce se může stát, že během této přípravné fáze dojde k nechtěnému DDoS útoku, jelikož proces přípravy vytváří značný provoz na síti.
Po vytvoření sítě pro distribuovaný útok určí útočník pomocí handlerů typ útoku a adresu oběti. Poté už jen čeká na správný okamžik pro zahájení útoku. Po zahájení útoku začnou všechny agentní stanice současně vysílat nepřetržitý proud paketů směrem k cíli útoku, čímž jej zahltí velkým objemem dat, která jsou k ničemu a vyčerpají takto jeho zdroje. Důvod použití zombie stanic, ať už v roli master nebo slave, pak spočívá jednak ve vytvoření velkého množství „útočníků“, jednak ve znemožnění vystopování útočníka [1][12].

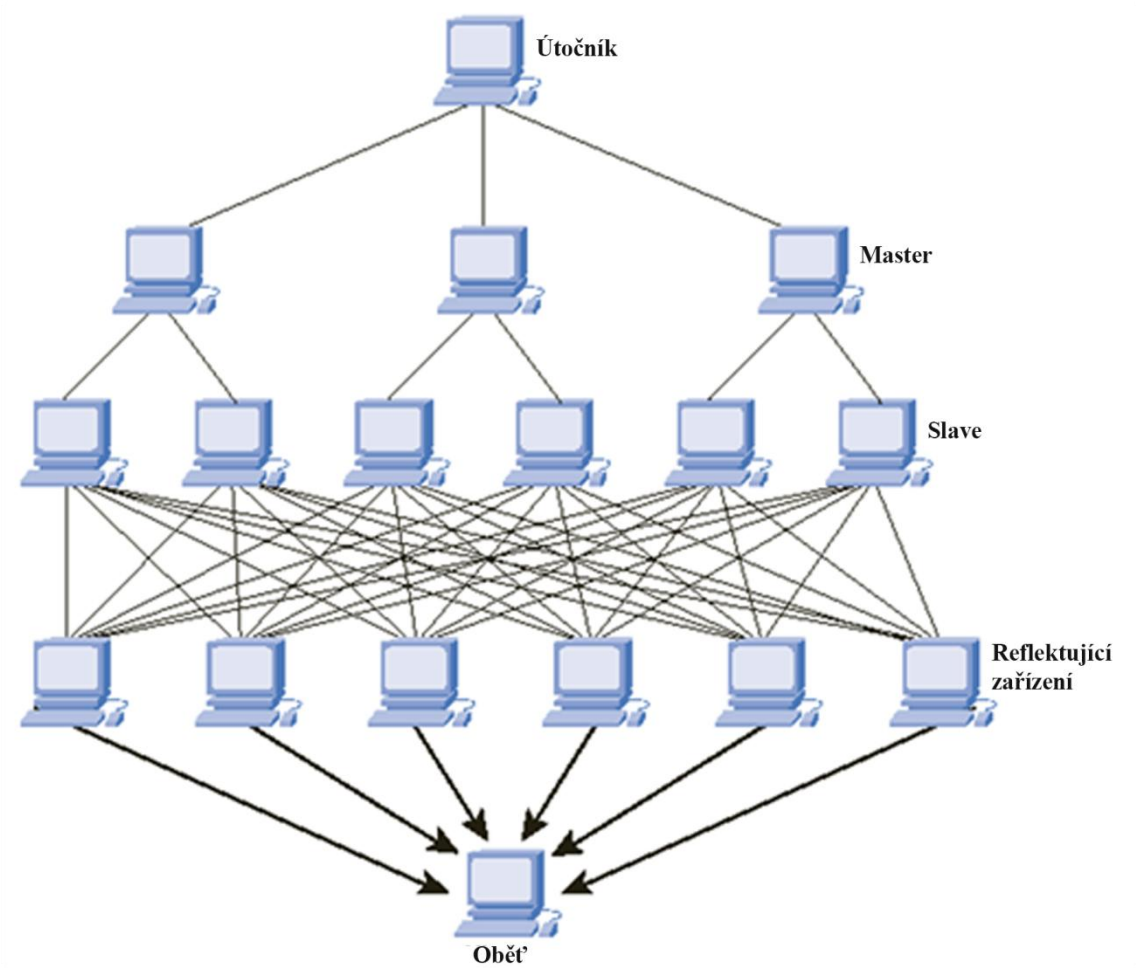



*Obrázek 2.1 Situace při DDoS útoku [12]*

### 2.1.4 Reflektivní útoky (DRDoS)

Reflektivní typy útoků spadají zejména do kategorie distribuovaných DoS útoků. Vyskytují se zde opět master a slave zombie stanice, nové jsou zde však reflektující stanice. Tak jako u klasických DDoS útoků má útočník kontrolu nad handlery, které pak kontrolují agentní stanice. Oproti klasickým DDoS útokům však neprovádí útok přímo agenti. Prostřednictvím master stanic instruuje útočník slave stanice, aby zaslali proud paketů s podvrženou IP adresou útočníka jako IP adresou zdroje na další zařízení, která nejsou útočníkem infikována. Tato zařízení právě slouží jako reflektující stanice, jež jsou tímto způsobem přiměny k tomu, aby zaslali oběti množství dat, jakožto odpověď na dotaz, o kterém si kvůli podvržené adrese myslí, že přišel právě od cíle útoku.

Tento typ DoS útoku je ještě zákeřnější než klasický DDoS útok. Zejména proto, že útok neprobíhá pomocí útočníkem napadených stanic. Je tedy výrazně obtížnější útočníka dohledat [1][12].




*Obrázek 2.2 Situace při DRDoS útoku [12]*

V kategorii DoS útoků se vyskytují i další typy a stále nové přibývají. Zde uvedené byly jedny z hlavních typů. Z hlediska detekce a ochrany jsou tyto útoky velmi nebezpečné, protože obrana proti nim je velmi náročná a většinou i drahá, zejména pak na cílových stanicích.

## 2.2 Skenování portů

Skenování portů slouží pro zjišťování spuštěných služeb na cílové stanici, k nimž je přistupováno právě pomocí otevřených portů, které jsou předmětem tohoto skenování. Toto skenování je většinou přípravnou fází nějakého následného útoku. Skenováním portů je totiž možné zjistit software a jeho verzi, který na cílové stanici (serveru) běží. Při použití znalosti slabin různých programů či jejich verzí je pak možné následný útok provést. Principů skenování je opět několik, jsou rozdílné v závislosti na tom, zdali jsou skenovány porty protokolu UDP nebo TCP. Princip jejich provádění však není předmětem této práce [1][5][6].

Tento druh anomálie se v síťovém provozu projevuje rozptýlením v rozložení cílových portů a naopak zvýšením koncentrace cílových IP adres [10]. Kromě zákeřného úmyslu použití této techniky, může být skenování portů také vhodným pomocníkem pro síťového administrátora, který jeho pomocí může například odhalit trojské koně. Tento případ použití je zde však pouze pro úplnost a není předmětem detekce síťových anomálií, jelikož je skenování portů vyvoláno síťovým správcem [5].

## 2.3 Skenování sítí

Cílem skenování sítě v kontextu útoků v počítačových sítích je nalezení stanic, které jsou v jistém ohledu zranitelné. Jedná se o zvláštní druh skenování portů, kdy je skenován jeden či malé množství portů na mnoha stanicích, jelikož se většinou jedná o cílené vyhledávání určité slabiny. Síťové skenování není opět útokem samotným, ale je přípravnou fází pro útok, jenž bude s největší pravděpodobností následovat [14]. Skenování sítí způsobuje v síťovém provozu rozptýlení v rozložení cílových IP adres a koncentraci v rozložení cílových portů [10].

### 2.3.1 Červy

Červ je nějaký škodlivý kód, jenž mimo své hlavní zákeřné činnosti, která závisí na tom, čeho chce útočník dosáhnout, má ještě jeden důležitý cíl – šíření sebe sama po síti. Červi vyhledávají zranitelné stanice, na které se pak kvůli chybám v software dále replikují. Tohoto šíření může být využito například při DDoS útoku, popsaném v kapitole 2.1.3, kdy je pomocí červů vytvořena síť stanic, použitých pro následný útok [12]. V oblasti síťových anomálií je zde však podstatné vyhledávání napadnutelných stanic, což je speciálním případem síťového skenování a jeho projevy v síťovém provozu byly popsány v úvodu této podkapitoly (2.3).

## 2.4 Alfa toky (alpha flows)

*Alfa tok* (doslovně přeloženo z anglického *alpha flow*) je termín, který se v oblasti počítačových sítí vyskytuje především v souvislosti s detekcí anomálií. Alfa toky nejsou útoky, nýbrž legitimní provoz, který však může vyvolat v objemu provozu anomálie a je tedy potřeba jej v kontextu detekce anomálií znát. Alfa tok je neobvykle velký proud dat mezi dvěma body. Tato anomálie představuje pouze zlomek z celkového počtu toků, jež vytvářejí provoz na síti. Alfa toky se však vyznačují silnými dávkami objemu dat, což způsobuje výkyvy rozložení normálního provozu. Na rozdíl od běžného provozu, kdy se rozložení objemu provozu blíží normálnímu (gaussovskému) rozložení, se tedy tyto toky vyskytují nepravidelně [16].

## 2.5 Přeplnění vyrovnávací paměti a výpadky

Tato podkapitola pouze doplňuje výčet anomálií, které jsou relevantní pro dále popisovanou metodu. U doposud popsaných anomálií se, kromě předešlých alfa toků, jednalo výhradně o anomálie se zákeřnými úmysly (s výjimkou skenování za účelem ochrany sítě, kdy je však toto skenování vyvoláno přímo správcem sítě). Následující anomálie – přeplnění vyrovnávací paměti a různé výpadky – však mohou v síti nastávat přirozeně, při běžném používání počítačové sítě. Vyrovnávací paměti (buffery) lze nalézt v počítačových sítích na různých místech např. na přepínačích (angl. switch) či směrovačích (angl. router), kde se používají za účelem uložení dat, která tato zařízení nedokážou plynule zpracovávat. Za určitých podmínek pak může při větším vytížení dojít k jejich zahlcení, což může být způsobeno třeba i špatným nastavením velikostí těchto vyrovnávacích pamětí. Nicméně zde může docházet k záměrnému vyvolání tohoto druhu anomálie za účelem znemožnění legitimního užívání daného zařízení či služby ostatními uživateli. Může se tedy jednat o jeden z druhů

DoS útoku. Ať už se však jedná o útoku, nevhodné nastavení sítě či pouze o událost, která někdy nastane běžným používáním, měl by o ní správce sítě vědět a vhodně na ni reagovat

Dalším typem anomálií, které jsou předmětem této podkapitoly, jsou výpadky. K výpadkům na síti může docházet z různých příčin a jsou většinou způsobeny nedokonalostí programového či strojového vybavení (softwaru a hardwaru) nebo i lidským selháním. Může tedy dojít například k pádu operačního systému serveru, k vyhoření zdroje směrovače, k vypnutí zařízení z důvodu údržby nebo třeba k přerušení kabelu při rekonstrukčních úpravách. Příčin může být vskutku mnoho. Je patrné, že se výpadky na síti projeví poklesem celkového provozu (tedy výskytů všech rysů provozu) směrem od daného výpadku. Nicméně je pravděpodobné, že se o výpadku některého zařízení, linky či programu dozví síťový správce dříve z jiného zdroje, než z výsledků detekce síťových anomálií. Nemusí tomu tak však být vždy a na každý pád je nutné tyto anomálie dokázat odlišit od ostatních detekovaných anomálií, i když už je jejich přítomnost známa předem.

# 3 Protokol NetFlow a toky v počítačových sítích

V této kapitole bude stručně představen protokol Netflow a dva druhy toků, jež jsou dále v rámci této práce používány při detekci anomálií. Síťový tok obecně je definován jako posloupnost paketů mající společnou vlastnost a procházející bodem pozorování za určitý časový interval. Všechny pakety patřící do jednoho toku, mají tedy společné vlastnosti, odvozené z obsahu těchto paketů [7].

Netflow je síťový protokol, vyvinutý společností Cisco Systems, pro shromažďování informací o provozu na síti. Netflow identifikuje tok podle zdrojové a cílové IP adresy, zdrojového a cílového portu, názvu logického rozhraní, typu protokolu transportní vrstvy protokolové sady TCP/IP a hodnoty ToS určující typ služby (z angl. type of service). Z těchto toků jsou v rámci metody pro detekci anomálií, popsané v kapitole 5, vytvářeny IP toky (právě pomocí protokolu Netflow). IP tok je určen jako $\boldsymbol{\tau} = \langle \mathbf{F}, \mathbf{P}, \mathbf{T} \rangle$, což znamená, že se IP tok $\boldsymbol{\tau}$ skládá z pětice **F**, jenž obsahuje protokol transportní vrstvy, zdrojovou a cílovou IP adresu a zdrojový a cílový port, dále je pak tento IP tok určen počtem paketů **P** a výskytem v čase **T**. S těmito toky pracuje však až poslední modifikace metody [11].

Základní metoda a první její modifikace pracuje s OD toky (z angl. origin destination flow). OD toky jsou zde uvažovány na úrovni páteřních sítí, které se skládají z uzlů – bodů přístupu (angl. Point of Presence či PoP) – jenž jsou propojeny linkami. OD tok je pak provoz, který vstupuje do této páteřní sítě na zdrojovém (angl. origin) PoP a opouští síť na cílovém (angl. destination) PoP. Cesta, kterou daný OD tok putuje, je určena směrovacími tabulkami [9].

# 4 Detekce anomálií v počítačových sítích

Se vzrůstajícím uplatněním informačních technologií, a tedy i počítačových sítí, vzrůstá také nebezpečí v podobě útoků (některé z nich byly představeny v kapitole 2). Mnohé z nich se v síti projevují neobvyklým – anomálním – chováním a právě detekce těchto anomálií se v posledních letech stává čím dál více důležitější oblastí jak v komerční sféře, tak i ve výzkumu. Způsob jak chránit počítačovou síť, spočívá v charakterizování známých či neznámých anomálních vzorků útoků. Obecně je anomálie v provozu na síti definována jako událost, která se v určité míře vychyluje od normálního provozu. Avšak vzhledem k tomu, že nemáme k dispozici žádný model, který by přesně popisoval normální provoz, je velmi obtížné navrhnout systém, který by přesně detekoval všechny anomálie.

Dle složitosti charakterizování normálního chování je možné rozdělit detekci anomálií na modelově založenou a na bezmodelovou detekci. U modelově založeného systému se předpokládá, že je k dispozici předem známý model, který z určitého pohledu definuje normální chování síťového provozu. Následně je pak každá odchylka od tohoto normálního stavu, považována za anomálii. V případech, kdy není možné vytvořit model, který by výstižně definoval normální provoz, je použito bezmodelových přístupů. Tyto přístupy pak mohou být dále rozděleny podle omezení přesnosti metody, které si daný návrh a implementace vynucují [17]. Podle [17] lze přístupy pro detekci anomálií dále dělit na:

- *Statistické přístupy*:
  - *Detekce bodu změny (change-point detection)*
  - *Analýza pomocí vlnkové transformace (wavelet analysis)*
  - *Analýza pomocí kovarianční matice*
  - *Kalmanův filtr*
  - A***nalýza hlavních komponent (Principal Component Analysis)***
- *Diskrétní algoritmy pro detekci anomálií:*
  - *„Heavy-hitter" detekce*
  - *Detekce prudkých změn (Heavy-change detection)*
- *Detekce anomálií pomocí strojového učení*:
  - *Učení bez učitele* - analýza využívající adaptivní meze, shluková analýza, Bayesovské sítě
  - *Učení s dodatečnou informací (Learning with additional information)*

Dále bude podrobněji popsána detekce založená na analýze hlavních komponent, jenž je hlavní náplní této práce.

## 4.1 Analýza hlavních komponent

Analýza hlavních komponent (dále PCA z angl. Principal component analysis) je statistická metoda, jenž provádí zjednodušení lineárně závislých (korelovaných) znaků snížením počtu dimenzí dat.

Lineárně transformuje původní znaky (proměnné) na nové, *nekorelované* proměnné, nazývané *hlavní komponenty* (angl. *principal axes* nebo *principal components)*. Základní charakteristikou každé hlavní komponenty je její míra variability (rozptyl), podle níž jsou hlavní komponenty seřazeny od nejvýznamnější, tj. mající největší rozptyl, k nejméně významné komponentě. Největší část informace o variabilitě dat je tedy soustředěna do první hlavní komponenty a nejméně informace je obsaženo v komponentě poslední. PCA je použita pro snížení dimenze dat čili počtu znaků bez velké ztráty informace, užitím pouze prvních několika hlavních komponent. Vstupem PCA je zdrojová datová matice o rozměrech $n \times m$, kde $n \geq m$[1]. Tato matice obsahuje $n$ pozorování a $m$ proměnných, přičemž platí, že pro každé pozorování je známa hodnota každé proměnné. Z této zdrojové matice vrací PCA sadu $m$ ortogonálních vektorů – hlavních komponent. Pro další práci je z těchto vektorů následně vybráno prvních $k$ hlavních komponent, které tvoří $k$-prostor, zachycující maximální rozptyl v datech. Smyslem detekce anomálií pomocí PCA je pak rozlišení normálního chování od anomálního právě za pomoci redukce dimenzí [8][17].

Základní myšlenka detekce anomálií pomocí PCA je následující: $k$-prostor získaný prostřednictvím PCA odpovídá normálnímu průběhu provozu, zatímco zbývající $(n - k)$-prostor odpovídá buď anomáliím, nebo anomáliím a šumu. Následně je každý vektor měření provozu (řádek zdrojové matice) zobrazen na normální a anomální podprostor. Poté je možné pomocí různých metod založených na prazích určit, zdali se jedná o normální či anomální provoz.

Různé metody založené na PCA se ukázaly být účinnými pro detekci anomálií v síťovém provozu. Nicméně jak bude dále popsáno, PCA je citlivá při nastavování jejích parametrů při praktickém použití. Velikost normálního prostoru, určená počtem hlavních komponent použitých pro jeho popis, má velký vliv na míru falešně pozitivních detekcí. Dále bylo také zjištěno, že velké anomálie mohou zkreslit definici normálního podprostoru, čímž dále zvyšují míru falešných poplachů. Účinnost PCA je také závislá na způsobu agregace dat (OD toky, IP toky, linky apod.) [9][14][17].

[1] Je logické, že je potřeba mít více pozorování než pozorovaných proměnných, abychom mohli vynášet plnohodnotné závěry.

# 5 Popis metody detekce založené na PCA

Metodu, která bude dále popsána a je jádrem této práce, představili pánové Anukool Lakhina, Mark Crovella a Christophe Diot v článku [9]. Síla této metody spočívá v detekci anomálií napříč celou rozsáhlou sítí, nikoliv jen na jedné lince či jedné stanici. Lze tedy s její pomocí do jisté míry podchytit a eliminovat síťové útoky již na úrovni poskytovatelů připojení, tedy tam kde je to nejvíce efektivní. Tato metoda však pracuje v tomto původním návrhu pouze s off-line daty a nemůže být proto použita pro detekování anomálií v reálném čase. Nicméně jejím záměrem je spíše předložit hlavní principy pro detekci anomálií založené na PCA, nežli poskytnout funkční detekční systém, který by bylo možné ihned nasadit v praxi.

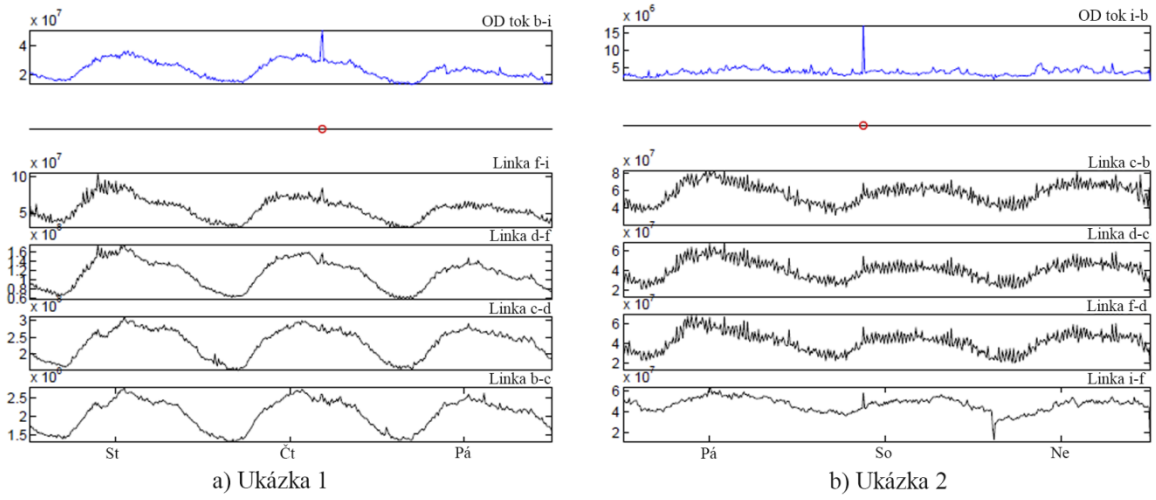
Základní varianta metody se zabývá pouze objemovými anomáliemi. Pojem objemová anomálie v rámci této práce znamená náhlou, kladnou či zápornou změnu v provozu, tedy náhlý nárůst či pokles počtu bytů, paketů či toků přenesených po síti za jednotku času. V rámci této práce pak byla studována rozšíření této metody, čerpající z článků, které navazují na původní práci autorů Lakhina, Crovella a Diot zabývající se objemovými anomáliemi. První z nich, popsané v článku [10], vylepšuje původní metodu ve smyslu použití nové, sofistikovanější metriky namísto prostého objemu dat. Důvodem tohoto vylepšení je snaha o detekování i těch anomálií, které se neprojevují výraznějšími změnami v objemu dat v provozu na síti. Další rozšíření je popsáno v článku [11] a jeho podstatou je vytváření náhodných agregací z IP toků a vícenásobná detekce, s cílem získání spolehlivějších
a přesnějších výsledků přímo nad IP toky.

## 5.1 Detekce objemových anomálií

Podprostorová metoda pro detekci objemových anomálií, popsaná v [9], pracuje s agregací provozu do OD toků, jež byly definovány v kapitole 3. Vstupem pro tuto metodu jsou informace o směrování a data o provozu na linkách, vytvořená složením provozu z jednotlivých OD toků. Jakým způsobem jsou tato data metodě předána, bude popsáno dále. Důvodem proč jsou data agregována do OD toků na úrovni páteřní sítě je efektivita. Detekce anomálií by samozřejmě mohla být prováděna pomocí získávání dat na úrovni IP toků ze vstupních linek každého PoP a následně z nich vytvářet OD toky. Nicméně vstupních linek každého z PoP mohou být stovky a agregace dat na takové úrovni je velmi náročná na výpočetní zdroje. Stejně tak přímá práce s IP toky, kterých mohou být miliony, je výpočetně velmi náročná. Metoda zde popisovaná, předkládá jednodušší a praktičtější přístup k detekování anomálií, využívajíce toho, že objemovou anomálii je možné sledovat v průběhu její cesty od zdroje k cíli na všech linkách, přes které prochází. Anomálie jsou zde tedy detekovány pouze na základě údajů o objemu dat na linkách.

Obtížnost detekce anomálií z dat pořízených pouze z linek znázorňuje Obrázek 5.1. Horní část obou grafů ukazují průběh objemu dat jednoho OD toku v průběhu času. Tato data však nejsou vstupními daty popisované metody. Na časové ose je pak červeným kolečkem vyznačena anomálie. Pod časovou osou je znázorněn provoz na linkách, které přenášejí daný OD tok. Tato data již jsou vstupem podprostorové metody. Z obrázků je patrné, že anomálie, která se projevuje výrazným

výkyvem objemu dat v provozu, je snadno odhalitelná při pohledu na data z OD toku. Avšak je-li nahlíženo na data z linek, je velmi obtížné tyto anomálie odhalit, dokonce i pouhým okem z grafu. Další komplikací, kterou je možné na obrázku pozorovat, je průběh provozu na jednotlivých linkách v čase. Na Ukázce 2, lince i-f je možno vidět poměrně hladký průběh provozu na této lince, zatímco ostatní linky, přenášející daný OD tok, obsahují více šumu. Oddělení anomálie, představované náhlým výkyvem v provozu, z šumem narušovaného provozu na lince c-b je očividně obtížnější nežli na lince i-f. Tudíž odhalení všech linek, na kterých anomálie vyvstává je náročné.




*Obrázek 5.1 Ukázka anomálií na úrovni OD toku (horní část), jež by měly být detekovány na základě provozu z linek. [9]*

### 5.1.1 Vstupní data

Jak již bylo řečeno, vstupními daty jsou data o provozu na jednotlivých linkách a směrovací informace. Údaje o směrování jsou zachyceny ve směrovací matici **A** o rozměrech $m \times n$ kde $m$ představuje počet linek a $n$ udává počet OD toků. Tato matice nese informaci o vzájemných vztazích mezi linkami a OD toky tak, že hodnota $\mathbf{A_{ij}=1}$, pokud OD tok $j$ prochází linkou $i$, jinak je nulová. Dalšími vstupními daty jsou vektory objemu dat v OD tocích. Takový vektor je vždy právě jeden pro jedno časové okno. Zde je důležité poznamenat, že všechny vektory v této práci jsou považovány za sloupcové, nebude-li výslovně řečeno jinak. Z tohoto souboru vstupních hodnot je následně vypočítána datová matice **Y** o rozměrech $t \times m$, kde $t$ značí počet časových oken, s kterými je pracováno a hodnota $m$ udává, stejně jako u směrovací matice, počet linek v síti. Jeden řádek matice **Y** je získán transponováním vektoru vypočítaného následovně: $\mathbf{y = Ax}$. V této datové matici zachycuje tedy každý sloupec $i$ průběh objemu provozu na lince $i$ v průběhu času a každý řádek $j$ obsahuje data ze všech linek po dobu časového okna $j$.

Aby nebyly výsledky detekce zkresleny mírou vytížení linek, musí být datová matice **Y** před aplikací PCA normalizována tak, aby její sloupce měly nulovou střední hodnotu. V následujícím textu je matice **Y** uvažována, jako již takto normalizována.

### 5.1.2 Použití PCA

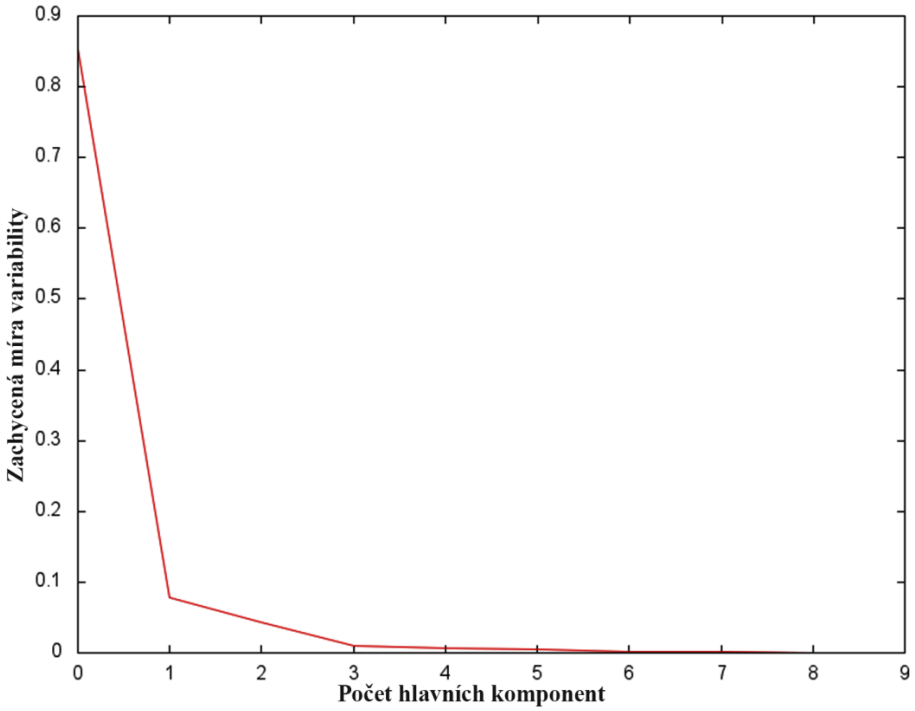
Normalizovaná matice **Y** je následně použita jako vstupní matice pro PCA. Ta vrací sadu $m$ hlavních komponent $\mathbf{v}_i$ pro $\mathbf{i = 1, \ldots, m}$. První hlavní komponenta $\mathbf{v}_1$ ukazuje ve směru největšího rozptylu ve zdrojové matici **Y**. Následně je vybráno prvních $r$ komponent, jejichž míra variability je nezanedbatelná. Tento počet prvních hlavních komponent je zpravidla výrazně menší, než celkový

počet všech komponent. Situaci názorně ukazuje Obrázek 5.2. Na tomto obrázku je graficky znázorněn podíl rozptylu každé hlavní komponenty z celkového rozptylu v datech. Graf, jenž znázorňuje Obrázek 5.2, byl vytvořen v rámci testování implementace metody a ukazuje, že většinu celkového rozptylu v datech zachycují první tři hlavní komponenty z celkového počtu osmi komponent.

Konstrukce normálního a anomálního (téže residuálního) podprostoru je provedena prostřednictvím mapování původních dat z datové matice **Y** na nové osy – hlavní komponenty. Toto mapování je dáno vztahem: $\mathbf{Yv_i}$ pro hlavní komponentu *i*. Tyto vektory jsou dále normalizovány na jednotkovou délku vydělením jejich normou. Pro každou hlavní komponentu *i* je tedy dáno mapování dat vztahem:

$$\mathbf{u_i} = \frac{\mathbf{Yv_i}}{\|\mathbf{Yv_i}\|} \qquad \boldsymbol{i = 1, \dots, m}$$

Vektory $\mathbf{u_i}$ mají velikost *t* a jsou ortogonální. Tyto vektory zachycují míru variability v průběhu času, společnou pro všechen provoz na linkách podél hlavní komponenty *i*. Opět zde platí, že jsou vektory seřazeny podle množství celkového rozptylu v datech. Pomocí těchto vektorů jsou následně rozděleny všechny hlavní komponenty do dvou skupin – normální a anomální. Sada normálních hlavních komponent zaujímá normální podprostor značený **S** a sada anomálních komponent utváří anomální podprostor $\tilde{\mathbf{S}}$. Hlavní komponenty jsou do těchto dvou souborů přiřazeny pomocí prahové separační metody. Ta pracuje tak, že bere po řadě zobrazení dat na hlavní osy do okamžiku, kdy se některá hodnota daného zobrazení vychýlí o více než $\mathbf{3\delta}$ od střední hodnoty (tedy překročí separační práh)[2]. Tato hlavní komponenta, při které došlo k překročení prahu, a všechny následující jsou přiřazeny do anomálního podprostoru. Všechny předešlé pak utváří podprostor normální. Po rozdělení prostoru všech měření objemu provozu do normálního a anomálního prostoru, je následně provedena detekce anomálií pomocí rozdělení provozu na každé lince na jeho normální a anomální složku. Princip této detekce je popsán v následující podkapitole.




*Obrázek 5.2 Podíl celkového rozptylu zachyceného každou hlavní komponentou*

[2] Tuto separační metodu dále také nazývám *δ-test*

### 5.1.3 Detekce

Detekce anomálií podprostorovou metodou spočívá v rozložení provozu na linkách, tedy vektoru **y**, v každém časovém úseku na jeho normální a anomální složku. V souladu s článkem [9] budou v této práci nazývány jako *modelovaná* a *residuální* část vektoru **y** a budou značeny $\hat{\mathbf{y}}$ (modelovaná) a $\tilde{\mathbf{y}}$ (residuální). Tyto složky jsou získány zobrazením dat vektoru **y** na podprostory **S** a $\tilde{\mathbf{S}}$. Toto zobrazení je provedeno následujícím způsobem. Ze sady normálních hlavních komponent ($\mathbf{v}_i$ _ $\mathbf{v}_r$) je vytvořena matice **P** o rozměrech $m \times r$, jejíž sloupce jsou právě tyto vektory. Připomínám, že $r$ je počet normálních hlavních komponent. Pomocí matice **P** lze následně získat vektory $\hat{\mathbf{y}}$ a $\tilde{\mathbf{y}}$ použitím vztahů:

$$\hat{\mathbf{y}} = \mathbf{P}\mathbf{P}^{\mathbf{T}}\mathbf{y} = \mathbf{C}\mathbf{y} \quad \text{a} \quad \tilde{\mathbf{y}} = \left(\mathbf{I} - \mathbf{P}\mathbf{P}^{\mathbf{T}}\right)\mathbf{y} = \tilde{\mathbf{C}}\mathbf{y}$$

kde $\mathbf{C}$ a $\tilde{\mathbf{C}}$ jsou lineární operátory, jenž provádí transformaci na normální podprostor **S** a anomální podprostor $\tilde{\mathbf{S}}$. Objemová anomálie by se pak měla projevit velkou změnou v residuální složce $\tilde{\mathbf{y}}$.

Pro detekci těchto abnormálních změn je použit čtverec chyby predikce (angl. Squared prediction error, dále značena SPE), určený vztahem:

$$\mathbf{SPE} \equiv \|\tilde{\mathbf{y}}\|^2 = \left\|\tilde{\mathbf{C}}\mathbf{y}\right\|^2$$

Pomocí prahové hodnoty $\boldsymbol{\delta}_{\propto}^{\mathbf{2}}$ pro míru spolehlivosti SPE $1-\propto$, je provoz určen jako normální, pokud splňuje podmínku:

$$\mathbf{SPE} \leq \boldsymbol{\delta}_{\propto}^{\mathbf{2}}$$

v opačném případě je provoz určen jako anomální. Jak je uvedeno v [9], $\boldsymbol{\delta}_{\propto}^{\mathbf{2}}$ je hodnota statistického testu *Q-statistic*, vyvinutého pány Jacksonem a Mudholkarem a je dána vztahem:

$$\boldsymbol{\delta}_{\propto}^{\mathbf{2}} = \boldsymbol{\phi}_{\mathbf{1}}\left[\frac{\mathbf{c}_{\boldsymbol{\alpha}}\sqrt{\mathbf{2}\boldsymbol{\phi}_{\mathbf{2}}\mathbf{h}_{\mathbf{0}}^{\mathbf{2}}}}{\boldsymbol{\phi}_{\mathbf{1}}} + \mathbf{1} + \frac{\boldsymbol{\phi}_{\mathbf{2}}\mathbf{h}_{\mathbf{0}}(\mathbf{h}_{\mathbf{0}} - \mathbf{1})}{\boldsymbol{\phi}_{\mathbf{1}}^{\mathbf{2}}}\right]^{\frac{\mathbf{1}}{\mathbf{h}_{\mathbf{0}}}} \tag{1}$$

kde

$$\mathbf{h}_{\mathbf{0}} = \mathbf{1} - \frac{\mathbf{2}\boldsymbol{\phi}_{\mathbf{1}}\boldsymbol{\phi}_{\mathbf{3}}}{\mathbf{3}\boldsymbol{\phi}_{\mathbf{2}}^{\mathbf{2}}} \quad \text{a} \quad \boldsymbol{\phi}_{\mathbf{i}} = \sum_{\mathbf{j}=\mathbf{r}+\mathbf{1}}^{\mathbf{m}} \boldsymbol{\lambda}_{\mathbf{j}}^{\mathbf{i}} \quad \text{pro } \mathbf{i} = \mathbf{1}, \mathbf{2}, \mathbf{3}$$

a kde $\boldsymbol{\lambda}_{\mathbf{j}}$ je míra variability zachycená zobrazením dat na hlavní komponentu $j$ a $c_\alpha$ je hodnota kvantilu normálního rozložení pro $1-\propto$. Nastavení hodnoty $1-\propto$ pak odpovídá míře falešně pozitivních poplachů. Takto je tedy získán soubor časových oken, v kterých nastává anomálie. V následující podkapitole bude popsáno, jakým způsobem jsou v těchto časových úsecích obsahujících anomálie, identifikovány OD toky, které tyto anomálie způsobily.

### 5.1.4 Identifikace

Podprostorová metoda uvažuje objemovou anomálii jako odchýlení stavového vektoru **y** od normálního podprostoru **S**. Konkrétní směr tohoto odsunu navíc poskytuje informaci o charakteru dané anomálie. Podstata identifikace tedy tkví v hledání takové anomálie ze souboru potencionálních anomálií, která nejlépe popisuje odchylku vektoru **y** od podprostoru **S**. Sada všech anomálií by měla být co nejúplnější. V rámci této metody je to tedy soubor všech OD toků a je značena jako množina $\{F_i, i = 1, \dots, n\}$.

Každá potencionální anomálie $F_i$ je spojena s vektorem $\theta_i$, jenž definuje míru, jakou daná anomálie přispívá k provozu na každé z linek. Zde, kde je uvažován pouze jediný OD tok jako zdroj objemové anomálie, přispívá tato anomálie ke všem linkám stejným množstvím provozu (ať už kladným či záporným). Vektor $\theta_i$ je tedy definován jako sloupec $i$ směrovací matice **A**, který je následně ještě normalizován na jednotnou velikost:

$$\boldsymbol{\theta_i} = \frac{\mathbf{A_i}}{\|\mathbf{A_i}\|}$$

Uvažujeme-li tedy normalizovaný vektor $\theta_i$, je pak v přítomnosti anomálie $F_i$ definován stavový vektor **y** vztahem:

$$\mathbf{y} = \mathbf{y}^* + \boldsymbol{\theta_i}\mathbf{f_i}$$

kde **y*** je vektor naměřených hodnot pro podmínky normálního provozu (typický vektor), který je však v přítomnosti anomálie neznámý. Hodnota $\mathbf{f_i}$ udává rozsah anomálie. Jelikož je hodnota vektoru **y*** v průběhu anomálie neznámá, je potřeba získat její odhad. Toho je dosaženo odstraněním vlivu dané anomálie, odečtením určitého množství provozu, kterým anomálie přispívá, ze všech linek s touto anomálií spojených. Nejlepší odhad vektoru **y*** v případě výskytu anomálie $F_i$ je nalezen minimalizováním vzdálenosti od normálního podprostoru **S** ve směru anomálie. Tato minimalizace vzdálenosti je dána vztahem:

$$\hat{\mathbf{f}}_\mathbf{i} = \arg\min_{\mathbf{f_i}} \|\tilde{\mathbf{y}} - \tilde{\boldsymbol{\theta}}_\mathbf{i}\mathbf{f_i}\|$$

kde $\tilde{\boldsymbol{\theta}}_\mathbf{i} = \tilde{\mathbf{C}}\boldsymbol{\theta_i}$. To určuje hodnotu $\hat{\mathbf{f}}_\mathbf{i} = \left(\tilde{\boldsymbol{\theta}}_\mathbf{i}^\mathbf{T}\tilde{\boldsymbol{\theta}}_\mathbf{i}\right)^{-1}\tilde{\boldsymbol{\theta}}_\mathbf{i}^\mathbf{T}\mathbf{y}$. Vzorec pro nejlepší odhad vektoru **y*** za předpokladu výskytu anomálie $F_i$ je tedy následující:

$$\begin{aligned}\mathbf{y}_\mathbf{i}^* &= \mathbf{y} - \boldsymbol{\theta}_\mathbf{i}\hat{\mathbf{f}}_\mathbf{i} \\ &= \mathbf{y} - \boldsymbol{\theta}_\mathbf{i}\left(\tilde{\boldsymbol{\theta}}_\mathbf{i}^\mathbf{T}\tilde{\boldsymbol{\theta}}_\mathbf{i}\right)^{-1}\tilde{\boldsymbol{\theta}}_\mathbf{i}^\mathbf{T}\mathbf{y} \\ &= \left(\mathbf{I} - \boldsymbol{\theta}_\mathbf{i}\left(\tilde{\boldsymbol{\theta}}_\mathbf{i}^\mathbf{T}\tilde{\boldsymbol{\theta}}_\mathbf{i}\right)^{-1}\tilde{\boldsymbol{\theta}}_\mathbf{i}^\mathbf{T}\tilde{\mathbf{C}}\right)\mathbf{y}\end{aligned}$$

Takto je tedy vypočítán typický vektor $\mathbf{y}_\mathbf{i}^*$ pro každou potencionální anomálii $F_i$. Následně je z těchto vektorů hledán ten, který popisuje největší množství residuálního provozu, tedy minimalizuje zobrazení $\mathbf{y}_\mathbf{i}^*$ na anomální podprostor $\tilde{\mathbf{S}}$. Jako původce anomálie je tedy nakonec určena anomálie $F_j$ kde $j$ je:

$$\mathbf{j} = \arg\min_{\mathbf{i}} \|\tilde{\mathbf{C}}\mathbf{y}_\mathbf{i}^*\|$$

Výstupem metody je tedy sada anomálních OD toků $j$ pro každé časové okno (vždy jeden OD tok pro jedno časové okno), ve kterém se vyskytuje anomálie.

Tato základní varianta byla v rámci této práce naimplementována, tak jak byla výše popsána. V následujících dvou podkapitolách budou popsána další dvě hlavní rozšíření podprostorové metody. V poslední podsekci pak bude popsáno další menší rozšíření, které bylo naimplementováno.

## 5.2 Detekce založená na analýze rozložení rysů provozu

Podprostorová metoda ve svém základním návrhu detekuje pouze objemové anomálie v rámci jednoho OD toku. Mnoho anomálií se však sledováním objemu provozu v obrovském množství dat na páteřních linkách nemusí výrazněji projevit. Naopak, jak uvádí [10], většina anomálií se v síti projevuje změnou rozložení *rysů provozu* (angl. *traffic feature*). Jak již bylo řečeno v kapitole 2, rys

provozu je údaj z hlavičky IP paketu a v rámci této metody se jedná o položky zdrojové a cílové IP adresy a zdrojového a cílového portu, které budou značeny po řadě zIP, cIP, zPort a cPort.

Tato kapitola popisuje rozšíření základní podprostorové metody na *více-cestnou podprosotorovu metodu*, popsanou v [10], která spočívá ve zkoumání rozložení rysů provozu a jejich vzájemné působení (proto *více-cestná*), s cílem dosažení většího počtu správně detekovaných událostí a menšího počtu falešných poplachů.

Pro zachycení rozložení daného rysu používá tato modifikovaná metoda hodnoty *entropie* vzorku. Entropie vyjadřuje v jediné hodnotě uspořádanost, respektive neuspořádanost, rozložení výskytů nějaké veličiny – v tomto případě rysů provozu. Je-li rozložení velmi neuspořádané, čili hodnoty daného rysu provozu se různí, je hodnota entropie vysoká. Naopak když je rozložení uspořádané, tedy hodnoty rysu se pohybují v malém rozsahu (v extrémním případě nabývají vždy pouze jediné hodnoty), je entropie nízká (ve zmíněném extrémním případě nulová).

Kromě toho, že tato modifikace umožňuje odhalení anomálií, které je pomocí objemově založené analýzy obtížné odhalit (ne-li dokonce nemožné), přináší tento nový přístup další výhodu. Struktura rozložení rysů provozu totiž nese důležité informace o charakteru anomálií, které mohou být následně použity pro automatickou klasifikaci anomálií do smysluplných kategorií. Nicméně automatické třídění anomálií do kategorií není předmětem této práce a je zde uvedeno jen pro úplnost.

### 5.2.1 Rozložení rysů provozu

Na rozdíl od původní varianty metody, jež používala pro detekci anomálií metriku objemu provozu na síti, pracuje více-cestná podprostorová metoda s informací o rozložení rysů provozu. Tato informace je zachycena pomocí *entropie*, která je zde použita jako sumarizační nástroj a shrnuje v jediné hodnotě informaci o rozložení či koncentraci každého rysu provozu. V rámci této práce i práce, z níž tato práce pramení, jsou jako rysy provozu uvažovány 4 výše zmíněné (zIP, cIP, zPort a cPort). Nicméně metoda by měla být natolik obecná, že je možné ji použít i pro další, jiné rysy provozu.

Základem pro výpočet entropie v rámci popisované metody je vektor měření $\mathbf{X} = \{\mathbf{n_i},\ \mathbf{i = 1}, \dots, \mathbf{N}\}$, kde hodnota $\mathbf{n_i}$ udává počet výskytů hodnoty **i** daného rysu a **N** je počet všech různých hodnot rysu. Entropie vzorku je pak definována jako:

$$\mathbf{H(X)} = -\sum_{\mathbf{i=1}}^{\mathbf{N}} \left(\frac{\mathbf{n_i}}{\mathbf{S}}\right) \mathbf{log_2}\left(\frac{\mathbf{n_i}}{\mathbf{S}}\right) \quad \text{kde} \quad \mathbf{S} = \sum_{\mathbf{i=1}}^{\mathbf{N}} \mathbf{n_i}$$

a kde **S** je celkový počet pozorování ve vektoru. Hodnota entropie vzorku $\mathbf{H(X)}$ leží v intervalu $\langle \mathbf{0}; \mathbf{log_2\, N} \rangle$. Minimální hodnoty 0 pak nabývá, pokud je pozorován pouze jediný výskyt rysu (tedy $n_1 = N$ a $i = 1$), a naopak maximální hodnoty $\log_2 N$ dosahuje, když jsou všechny různé výskyty stejně časté (tzn. $n_1 = n_2 = \cdots = n_N$).

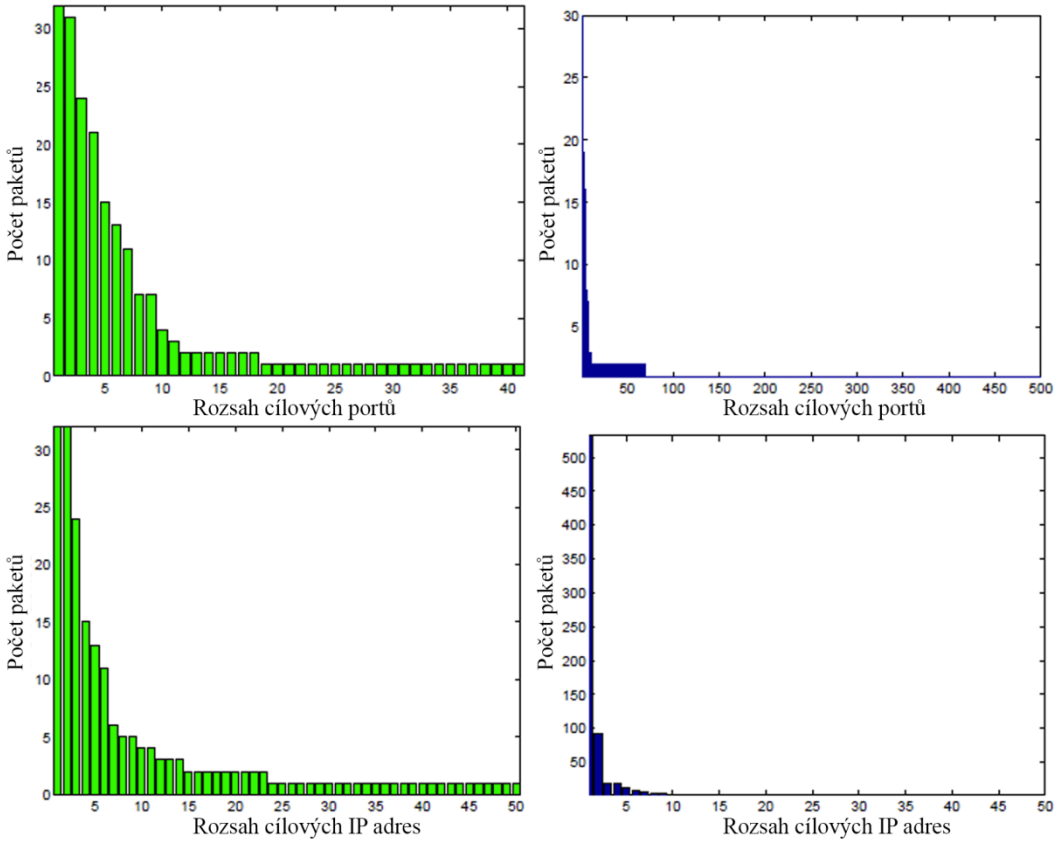
Použití rozložení rysů provozu, jakožto nástroje pro detekci, je podle [10] založeno na skutečnosti, že mnoho významných anomálií v provozu na síti způsobuje změny v rozložení adres nebo portů. Tabulka 5.1 uvádí přehled anomálií, které se běžně vyskytují v provozu na páteřních linkách. Většina z těchto anomálií byla popsána v kapitole 2. Každá z uvedených anomálií ovlivňuje rozložení určitých rysů provozu, jako například když se rozložení zdrojových IP adres stává více rozptýleným při podvrhování („spoofování“) těchto adres při DoS útocích a zároveň se rozložení cílových IP adres koncentruje směrem k cíli útoku.

| Označení anomálie | Popis | Ovlivněné rysy provozu |
|---|---|---|
| *Alfa tok* | Neobvykle velký tok velkého objemu dat mezi dvěma body | Zdrojová a cílová IP adresa (možné i porty) |
| *DoS* | Útok znepřístupňující nějakou službu (distribuovaný nebo z jednoho zdroje) | Cílová a zdrojová IP adresa |
| *Přeplnění flash* | Neobvykle velká dávka provozu k jedinému cíly z „typického“ rozložení zdrojových rysů | Cílová IP adresa, cílový port |
| *Skenování portů* | Prohledávání mnoha cílových portů na malé sadě cílových adres | Cílová IP adresa, cílový port |
| *Skenování sítí* | Prohledávání mnoha cílových adres na malé sadě cílových portů | Cílová IP adresa, cílový port |
| *Výpadek* | Změna v provozu vyvolaná selháním zařízení či údržbou | Hlavně zdrojová a cílová IP adresa |
| *Bod do více bodů (Point to multipoint)* | Provoz z jednoho zdroje do více cílů, vyvolaný např. šířením škodlivého obsahu | Zdrojová a cílová IP adresa |
| *Červ* | Skenování vyvolané červy s účelem nalezení zranitelných hostitelských stanic | Cílová IP adresa, cílový port |

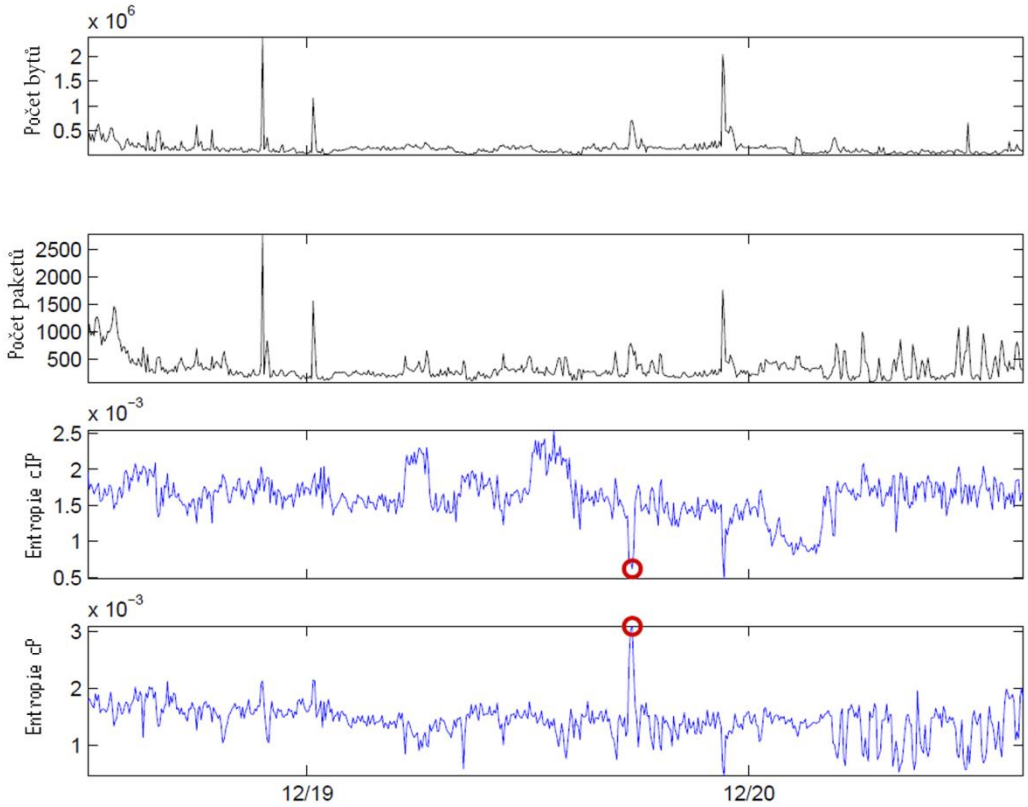
*Tabulka 5.1 Přehled způsobu ovlivnění rysů provozu různými anomáliemi [10]*

Dále Obrázek 5.3 ukazuje, jak se rozložení rysů provozu mění následkem anomálie. Na tomto obrázku je zachyceno skenování portů, probíhající na páteřní síti Abilene. Obrázek 5.3 zobrazuje v horní polovině rozložení cílových portů a v dolní polovině rozložení cílových IP adres. Nalevo je pak znázorněna situace během normálního provozu, napravo situace v průběhu anomálie. Každý z grafů tohoto obrázku zobrazuje rozložení rysů provozu v průběhu 5 minutového časového okna jako histogramy s hodnotami sestupně seřazenými podle počtu výskytů. V horní polovině obrázku mají oba grafy stejný rozsah osy *y* a liší se rozsahem osy *x*. Je zde tak možné pozorovat rozdíl v rozptylu cílových portů, kdy je celkový počet vyskytujících se portů v průběhu anomálie výrazně větší, přestože výskyt nejpoužívanějších portů zůstává přibližně stejný (okolo 30). V dolní polovině dochází u cílových IP adres k opačnému jevu. Na těchto grafech mají oba stejný rozsah osy *x* a různý na ose *y*. Zde je zhruba stejný počet různých IP adres v obou případech, ale během anomálie se rozložení těchto adres stává mnohem více koncentrované na jedinou adresu.

Obrázek 5.4 ilustruje účinnost entropie pro detekci anomálií v kontrastu s metrikami objemu dat v provozu. Tento obrázek tedy ukazuje grafy různých metrik provozu okolo času, kdy probíhá anomálie typu skenování portů, jež byla popsána v předchozím odstavci a jejíž histogramy ukazuje Obrázek 5.3. Bod v čase, kde se vyskytuje anomálie, je zobrazen červeným kroužkem. Horní dva grafy zobrazují objemové metriky počtu bytů a paketů v OD toku a ukazují, jak je obtížné detekovat skenování portů na základě těchto metrik. Avšak dolní dva grafy, zobrazující hodnoty entropie rozložení cílových IP adres a portů, ukazují, jak tato anomálie jasně vyniká při použití této metriky. Entropie cílových IP adres ostře klesá následkem koncentrace rozložení okolo jediné IP adresy a entropie cílových portů naopak ostře stoupá kvůli rozptýlenému rozložení cílových portů.




*Obrázek 5.3 Změna v rozložení rysů provozu vyvolaná skenováním portů [10]*




*Obrázek 5.4 Pohled na skenování portů z hlediska objemu dat v provozu a z hlediska entropie [10]*

### 5.2.2 Vstupní data a detekce

Více-cestná podprostorová metoda zkoumá vzájemné působení čtyř rysů provozu. Vstupem je tedy trojrozměrná matice **H** taková, že hodnota **H(t, o, k)** udává hodnotu entropie v čase *t*, OD toku *o*, rysu provozu *k*. Jednotlivé podmatice rysů provozu jsou značeny **H(zIP)**, **H(cIP)**, **H(zPort)**, **H(cPort)**. Každá tato matice má velikost $t \times p$ a prostřednictvím hodnot entropie nese data o průběhu rozložení daného rysu provozu jednotlivých OD toků, kterých je *p*, v čase o délce *t* časových oken. V těchto datech jsou následně vyhledávány výrazné odchylky hodnot entropie, které značí výskyt anomálie. Aby bylo možné k tomuto vyhledávání odchylek použít původní podprostorovou metodu, je nejdříve potřeba z této trojrozměrné matice vytvořit matici dvojrozměrnou. To je provedeno tak, že se jednotlivé podmatice rysů provozu složí vedle sebe, čímž vznikne nová matice **H** o rozměrech $t \times 4p$. Prvních *p* sloupců této sloučené matice pak obsahuje data z podmatice **H(zIP)**, v dalších *p* sloupcích ($p+1$ až $2p$) jsou data z **H(zPort)**, sloupce $2p+1$ až $3p$ obsahují data z podmatice **H(cIP)** a posledních *p* sloupců ($3p+1$ až $4p$) nese data z **H(cPort)**. Tím je tedy vytvořena dvojrozměrná matice, která může být použita jako vstup pro podprostorovou metodu, popsanou v podkapitole 5.1. Než však bude tato metoda na zdrojovou matici **H** použita, je ještě zapotřebí provést normalizaci matic na *jednotkovou varianci*[3], aby žádný z rysů nedominoval následné analýze. To je provedeno vydělením každého prvku podmatice její celkovou variancí. Takto upravená zdrojová matice může být následně použita pro podprostorovu metodu. Průběh detekce je pak stejný jako u původní varianty, pracující s objemem dat a jejím výstupem je sada časových oken, obsahujících anomálie.

V následující sekci bude popsáno další rozšíření více-cestné podprostorové metody, které si klade za cíl navýšení spolehlivosti výsledků detekce zvýšením míry pozitivních detekovaných anomálií a snížením míry falešných detekcí.

## 5.3 Detekce využívající náhodné agregace IP toků

Jak již nadpis této podkapitoly napovídá, tato další modifikace, vysvětlená v článku [11], posouvá zde popisovanou metodu dále ve smyslu použití IP toků namísto OD toků. V předchozích dvou variantách museli být po detekci a identifikaci OD toků obsahujících anomálie, dolovány IP toky právě z těchto OD toků ručně. Zde popisované vylepšení metody, které autoři nazývají *Defeat*, pracuje s IP toky přímo a nepotřebuje tedy ruční dohledávání anomálních IP toků. Navíc zde také není potřeba žádná směrovací matice, ani jiné informace o topologii síti.

Kromě použití IP toků se Defeat dále snaží o vylepšení robustnosti detekčního systému použitím násobných náhodných rozdělení IP toků do skupin (tzv. *sketches*). Toto vylepšení vychází z nového poznatku, jenž uvádí, že zmíněné náhodné přerozdělování globálního toku do skupin zachovává míru variability provozu v rámci normálního a většiny anomálního (residuálního) podprostoru. IP toky jsou do těchto skupin rozřazovány prostřednictvím rozptylovacích („hashovacích“) funkcí způsobem, který bude popsán dále. V tuto chvíli je podstatné to, že vstupní IP toky jsou náhodně rozděleny do tolika skupin, kolik je použito rozptylovacích funkcí. Následně je stejněkrát provedena i detekce, čímž je získáno několik sad anomálií. Pomocí volebního přístupu, jehož princip bude také popsán později, je pak dosaženo větší spolehlivosti a přesnosti výsledků.

[3] V původním článku je uveden pojem „*unit energy*“ bez bližšího vysvětlení, přestože se tento pojem v žádné jiné literatuře nepoužívá. Z kontextu však lze odvodit, že se s největší pravděpodobností jedná o varianci všech hodnot v matici. Energii pro matici X o rozměrech m×n lze tedy vypočítat podle těchto vzorců:
$E(X) = \sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij}\,/(m \times n),\ Energy(X) = \sum_{i=1}^{m}\sum_{j=1}^{n}\left[\left(x_{ij} - E(X)\right)^2\right]/(m \times n)$

### 5.3.1 Vstupní data, vytváření náhodných uskupení a detekce

Jak již bylo řečeno, metoda Defeat pracuje na rozdíl od předchozích dvou s IP toky namísto OD toků. K získávání informací o IP tocích je použito *netflow záznamů*, jak bylo popsáno v kapitole 3. Z každého IP toku je vytvořen klíč, který je použit pro vytváření náhodných uskupení. Tento klíč je získán konkatenací prvních 21 bitů zdrojové a cílové IP adresy (celkem tedy 42 bitů).

Počet řádků náhodného uskupení je dán předem a bude značen **s**. Podle původní práce by měla být velikost **s** volena v rozsahu hodnot 32 – 1024. Tato náhodná uskupení jsou vytvářena následovně. Postupně jsou brány IP toky $\boldsymbol{\tau}$, respektive jejich klíče, jenž jsou použity jako vstup rozptylovací funkce $\mathbf{h_i}(\cdot)$ pro $\mathbf{i = 1, \dots, m}$ (dále značí *m* počet rozptylovacích funkcí). Rozptylovací funkce $\mathbf{h_i}(\cdot)$ převede hodnotu klíče IP toku na hodnotu v rozsahu $\mathbf{1, \dots, s}$. Podle této hodnoty jsou přidány údaje o výskytu daného rysu (zIP, cIP, zPort či cPort) do každého náhodného uskupení pro každý rys (tedy do 4 uskupení pro jednu rozptylovací funkci). Z toho tedy plyne, že jednotlivé řádky v rámci jednoho náhodného uskupení mohou mít různou délku, podle hodnot výskytů daného rysu. Každý tento řádek pak tvoří nový, náhodně vytvořený pseudotok. Tyto pseudotoky budou dále použiti pro detekci anomálií výše popsanými algoritmy, obdobným způsobem jako původní OD toky. OD toky však byly pouze jedním konkrétním uskupením dat. Zde je vytvořeno *m* náhodných uskupení, což přináší dvě výhody. Jednak tu, že anomálie, jež mohou být skryté v rámci jedné agregace, budou odhaleny v jiné, což vede k vysoké míře detekcí. Jednak tu, že anomálie, detekované pouze v rámci malé podmnožiny agregací, budou pravděpodobně falešnými poplachy a mohou být tedy zahozeny.

Aby bylo možné použít na data v náhodných uskupení podprostorovou metodu, je potřeba z těchto dat vytvořit vstupní matici pro PCA z celkového množství $4 \times m \times t$ (*4* rysy provozu, *m* rozptylovacích funkcí, *t* časových oken) náhodných uskupení o velikosti *s*. Z každého řádku *i* náhodného uskupení pro časové okno *t*, určitého rysu je vypočítána hodnota entropie, která je uložena na řádek *t* sloupce *i* datové matice onoho rysu. Výsledkem těchto výpočtů je tedy $4 \times m$ datových podmatic o velikosti $t \times s$. Jednotlivé podmatice rysů dané rozptylovací funkce jsou následně složeny po stranách k sobě, tak jak bylo popsáno u skládání podmatic rysů v rámci více-cestné podprostorové metody v podkapitole 5.2.2. Vstupními daty pro podprostorovou metodu pak bude *m* těchto složených matic o velikosti $t \times 4s$. Ty jsou však před provedením detekce opět normalizovány. Tato modifikace původní podprostorové metody přináší do detekční části ještě zobecnění definice normálního podprostoru, kdy je tento prostor napevno definován prvními deseti hlavními komponentami.

Následná detekce bude provedena *m-krát* a výsledných vektorů časových oken, obsahujících anomálie bude také *m*. Z této množiny výsledných vektorů je pak vytvořen, pomocí volebního přístupu a volebního prahu *n*, jeden výsledný vektor. Hodnota *n* je z rozsahu $1 - m$[4] a určuje, v kolika dílčích vektorech anomálních časových oken musí být určitá anomálie označena, aby byla tato anomálie přidána do vektoru výsledného. Tím je navýšena spolehlivost výsledků detekční části metody.

Předchozí varianta (více-cestná podprostorová metoda) v rámci této práce implementována nebyla. Naprogramováno bylo až toto její vylepšení, využívající agregací IP toků do náhodných uskupení.

[4] Autoři původní práce doporučují $n = m - 1$

## 5.4 Předzpracování dat

Na základě podnětu z článku [14] jsem mezi vylepšení metody zahrnul ještě předzpracování vstupních dat. To má za účel odstranění obrovských anomálií, které by mohli nepříznivě ovlivňovat definici normálního podprostoru a které jsou snadno detekovatelné jakožto výrazné odchylky. Základní jednoduchou metodu pro předzpracování dat jsem vyvinul společně s vedoucím této práce - Ing. Václavem Bartošem. Algoritmus pro předzpracování dat bere datovou matici **Y**, respektive **H** ještě před normalizací a odstraní z nich výrazné odchylky. To je provedeno následovně. Z datové matice jsou postupně brány sloupce, představující časový průběh provozu v rámci dané linky, respektive daného pseudotoku. Z každého tohoto sloupce je následně vypočítána jeho střední hodnota. Dále jsou z každého sloupce nahrazeny touto střední hodnotou všechny údaje, jež se odchylují od střední hodnoty více jak o $\mathbf{x} * \boldsymbol{\delta}$ kde $x$ je volitelný parametr, určující velikost odchylky v datech k odstranění. Během testů pak nejlépe fungovaly hodnoty $x = 4$ a $x = 5$. Dále jsou o odstraněných anomáliích ukládány informace o čase a lince (pseudotoku), na které se vyskytly. Tím je tedy umírněn vliv velmi výrazných anomálií na definici normálního prostoru. Předzpracování dat bylo implementováno pro obě varianty detekce anomálií.

# 6 Návrh a implementace aplikace

V rámci této práce byla za účelem otestování detekce anomálií základní podprostorové metody, popsané v kapitole 5.1 tato metoda implementována. Dále pak bylo implementováno rozšíření této metody, využívající přerozdělování IP toků do náhodných uskupení, popsané v kapitole 5.3. Nad oběma variantami pak bylo implementováno předzpracování dat tak, jak je uvedeno v kapitole 5.4. Implementace je koncipována jako soustava konzolových aplikací, ovládaných prostřednictvím parametrů[5] a výsledná aplikace je napsána procedurálně v programovacím jazyce C/C++. Výstupy detekce anomálií jsou ukládány do souborů, případné informace o průběhu vykonávání programu jsou vypisovány na standardní chybový výstup. Tato aplikace nebyla vytvořena s cílem využití v ostrém provozu. Smyslem této implementace je vyzkoušení a ověření funkčnosti popsaných algoritmů. Za tímto účelem jsem se tedy snažil vytvořit program tak, aby byl co nejvíce parametrizovatelný a zjednodušil tak následné testování s různým nastavením velikosti normálního podprostoru, detekčních prahů apod.

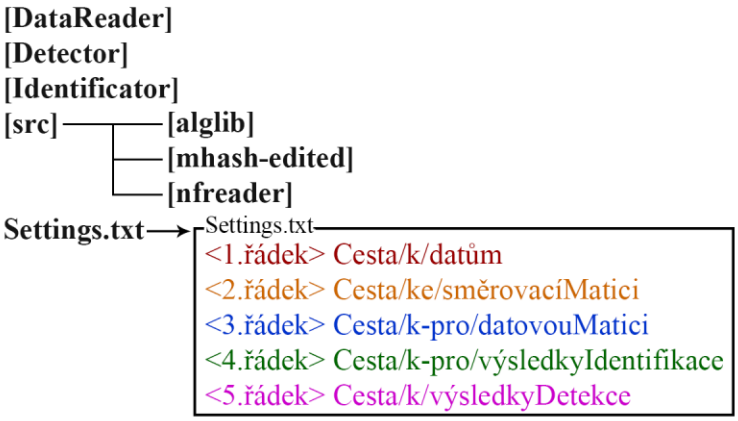
Implementovaná aplikace pro detekci anomálií je rozdělena do dvou samostatných celků. První z nich má na starosti načtení dat a jejich přípravu pro detekci, respektive pro PCA. Ta bude popsána v podkapitole 6.2. Druhá část již provádí detekci samotnou. Princip její činnosti je popsán v kapitole 6.3. Pro původní variantu podprostorové metody jsem implementoval i identifikační část, abych otestoval i její principy, přestože hlavní náplní této práce je pouze detekce anomálií. Její implementace je popsána v kapitole 6.4. Důvodem, proč je celý proces rozdělen na jednotlivé celky, je úspora času při provádění testů, kdy např. při testování různé velikosti normálního podprostoru není zapotřebí, aby byla data vždy znovu vypočítávána ze směrovací matice a dat objemu provozu na linkách (nebo dokonce znovu rozdělována rozptylovacími funkcemi), následně normalizována a eventuelně předzpracována - tento krok stačí provést jednou.

## 6.1 Struktura programové soustavy

Jak již bylo řečeno, programová soustava se skládá ze dvou, respektive ze tří hlavních částí. Chování každé z těchto částí je definováno v jednom hlavním zdrojovém souboru, který je uložen vždy v adresáři dané části. Jsou to adresáře [DataReader] pro část načítající data, [Detector] pro část detekující anomálie a [Identificator] pro identifikaci objemových anomálií. Kromě samostatných hlavních zdrojových souborů, jež definují činnost dané části, využívají všechny tři části dalších zdrojových souborů, které jsou pro ně společné a jsou uloženy v adresáři [src]. Zaprvé je zde hlavní hlavičkový soubor, jenž kromě deklarací obecných funkcí a definic abstraktních datových typů obsahuje také definici maker, jež určují nastavení celé aplikační sady. Další zdrojové soubory se nachází v adresáři [src]/[alglib]. Jedná se o soubory volně dostupné matematické knihovny *alglib*, jenž obsahuje zejména definici funkce provádějící PCA. Dalšími zdrojovými soubory jsou upravené soubory z volně dostupné knihovny *mhash*, které obsahují definici použitých rozptylovacích funkcí. Tyto soubory se nachází v adresáři [src]/[mhash-edited]. Dále se ve složce [src]/[nfreader] nacházejí zdrojové soubory knihovny `nfreader`, jenž je součástí frameworku NADEX [3] a poskytuje rozhraní pro čtení výstupních souborů nástroje NfDump, tedy pro načítání NetFlow záznamů. Posledními zdrojovými soubory jsou *matrix.cpp/matrix.h*, téže v adresáři [src]. Tyto soubory obsahují

[5] Použití parametrů je popsáno v příloze *Manuál*

definici vlastního typu pro matice (`TMatrix`) a operace nad nimi. Kromě operací nad vlastním datovým typem `TMatrix` pro matice, obsahují tyto soubory také několik operací nad datovým typem `real_2d_array`, jenž je reprezentací matice (resp. dvourozměrného pole) pro knihovnu *alglib* a několik operací hybridních (pracujících s oběma typy). Poslední částí celé soustavy je textový soubor *Settings.txt*, který uchovává výchozí nastavení cest k jednotlivým vstupům a výstupům, potřebným pro správnou funkci celé soustavy.




*Obrázek 6.1 Struktura aplikační soustavy*

Všechny výchozí cesty je možné doplnit či úplně nahradit prostřednictvím parametrů, jimiž jsou jednotlivé aplikace ovládány. Na prvním řádku souboru *Settings.txt* je uvedena výchozí cesta k datovému souboru. Na druhém řádku souboru je výchozí cesta k souboru popisujícímu směrovací matici. Tyto dva soubory jsou vstupními daty pro mód pracující s objemem dat první aplikace, v případě druhého módu této aplikace uvádí cesta na prvním řádku souboru *Settings.txt* cestu k adresáři se vstupními soubory (viz. 6.2.2). Třetí řádek souboru určuje výchozí cestu k adresáři, do kterého bude první aplikace ukládat výstupní datovou matici (varianta objemu) či matice (pro každou rozptylovací funkci – varianta náhodných uskupení). Tento řádek zároveň určuje, odkud bude druhá (detekční) část tuto matici získávat. Tato část zpracovává vždy jen jedinou vstupní matici, v případě potřeby je tedy nutné ji spouštět opakovaně s odlišnými parametry, např. pomocí jednoduchých skriptů. Čtvrtý a pátý řádek určují výchozí cesty k výsledkům. Na čtvrtém řádku je určena cesta pro výstupy identifikační části, do které zapisuje první aplikace data (mód objemu dat), potřebná pro účely identifikace. Třetí, identifikační část soustavy pak odtud tato data čte a následně na stejné místo ukládá identifikační výstupy, popsané v kapitole 6.4. Na posledním, pátém řádku je uložena výchozí cesta k výsledkům detekční aplikace. Ty budou popsány v kapitole 6.3. Celou zmíněnou aplikační strukturu shrnuje Obrázek 6.1.

## 6.2 Zpracování vstupních dat

První z aplikační sady je, jak již bylo zmíněno, aplikace provádějící zpracování vstupních dat. Ta pracuje ve dvou módech v závislosti na použité verzi metody. Následující popis je tedy rozčleněn do tří sekcí, z nichž první dvě popisují tyto dva módy, a poslední část se zabývá předzpracováním dat, které je společné pro oba tyto módy. Oba módy vytváří ze vstupních informací, jejichž forma tyto přístupy odlišuje, datovou matici, která slouží pro následnou detekci, založenou na PCA. Ta je tedy vstupem druhé (detekční) části aplikační soustavy. Formát této datové matice je pak pro oba módy stejný.

### 6.2.1 Mód načítání dat z objemu provozu

První mód je určen pro vytvoření datové matice za pomocí směrovací matice a dat o objemu provozu v jednotlivých tocích. Obě tyto sady vstupních dat jsou reprezentovány vstupními soubory, jež mají odlišný, pevně stanovený formát:

- Směrovací matice:

  1. řádek souboru:`x y` – udává rozměry směrovací matice v podobě *x řádků* × *y sloupců*.

  2. – n. řádek souboru:`a b` – udává souřadnice ve směrovací matici, na jejichž pozici mají být hodnoty 1. Souřadnice *a* udává pozici řádku a hodnota *b* určuje sloupec.

- Datová matice:

  1. řádek souboru: `x y` – udává také rozměry vstupní datové matice v podobě *x řádků* × *y sloupců*.

  2. – x. řádek souboru: $a_1$ $a_2$ … $a_y$ - udává hodnoty matice s informacemi o objemu dat v OD tocích oddělené mezerami. Každý řádek této matice obsahuje data v rámci jednoho časového okna a každý sloupec obsahuje data jednoho OD toku.

Obě tyto matice jsou načteny do paměti a následně je z nich pomocí vztahu $\mathbf{y} = \mathbf{Ax}$ (viz. kapitola 5.1.1) vypočítána základní datová matice. Pokud je zapnuto předzpracování, jehož popis bude uveden v podkapitole 6.2.3, je toto předzpracování dat provedeno. Po případném předzpracování je provedena normalizace datové matice, tak aby její sloupce měli nulovou střední hodnotu. Po tomto kroku je matice připravena pro PCA a je tedy následně uložena do souboru, do patřičného adresáře (viz. kapitola 6.1). Tento soubor odráží podobu této matice, na jeho řádcích jsou tedy hodnoty řádků datové matice, bez jakýchkoliv dodatečných informací. Cesta a základ pro název výstupního souboru je nastaven pomocí parametru (viz. *Manuál*), název je vždy ještě doplněn řetězcem „_DM“[6] (z angl. data matrix). Následně je činnost aplikace pro načítání dat, módu objemu dat z provozu ukončena.

### 6.2.2 Mód načítání NetFlow dat

Tento mód pracuje pouze s jediným druhem vstupních souborů, obsahujícím Netflow záznamy, přičemž je vždy právě jeden takový soubor pro jeden časový okamžik. Soubory obsahující Netflow záznamy jsou určeny zadáním adresáře, v němž se tyto soubory nachází. Je důležité, aby se v tomto adresáři nenacházely žádné další soubory, jelikož podle počtu souborů v tomto adresáři určuje aplikace počet časových oken, s kterými bude pracováno. Z jednotlivých souborů získává aplikace IP toky, tak jak byly definovány v kapitole 3. Kvůli obrovskému množství Netflow záznamů jsou vstupní soubory zpracovávány po jednom. Ze souboru jsou průběžně načítány IP toky, které jsou ihned zpracovány. Z načteného toku je nejprve získán klíč (viz kapitola 5.3.1). Na tento klíč je následně použita každá z rozptylovacích funkcí, jež určí pozici v sadě 4 náhodných uskupení (jedno pro každý rys), náležící k dané funkci. Těchto uskupení je tedy pro jeden soubor, který představuje jedno časové okno *t*, *4* × *m*. Do každé sady 4 uskupení pro danou rozptylovací funkci je na řádek uskupení patřičného rysu, jenž určuje vypočítaná pozice, přidán výskyt patřičného rysu. Jakmile jsou načteny a rozděleny všechny IP toky z jednoho souboru, je z řádků jednotlivých uskupení vypočítávána entropie. Hodnota entropie řádku *s* je pak uložena do datové podmatice pro danou

[6] Tento řetězec lze stejně jako všechny ostatní doplňky názvů souborů snadno modifikovat změnou makra v hlavním hlavičkovém souboru *main.h*.

rozptylovací funkci a rys provozu na řádek *t*, sloupce *s*. Po načtení všech souborů jsou datové podmatice kompletní. Tyto datové podmatice jsou opět 4 (jedna pro každý rys provozu) pro každou rozptylovací funkci. Následně jsou tyto podmatice složeny do jediné datové matice pro každou z funkcí tak, jak bylo popsáno v kapitole 5.2.2. Dále je nad každou z matic provedeno volitelné předzpracování a následně normalizace. Nakonec jsou takto upravené matice opět uloženy do souboru stejným způsobem jako u prvního módu. Název výstupní datové matice je také doplněn řetězcem, tentokrát je to však řetězec „x_DM“, kde $x = 0, \dots, (m-1)$ označuje číslo rozptylovací funkce a kde *m* značí počet rozptylovacích funkcí. Po zapsání výstupních datových matic do souborů je činnost první aplikace ukončena.

### 6.2.3 Předzpracování dat

Princip předzpracování dat byl popsán v kapitole 5.4. Implementovaný algoritmus je pro oba dva módy první aplikace stejný. Jako vstup bere funkce, jež provádí předzpracování, základní (nenormalizovanou) datovou matici. Následně jsou z této matice odstraňovány nejvýraznější anomálie. Jak moc musí být anomálie výrazné, aby byly odstraněny je opět možno nastavit[7]. O odstraněných anomáliích jsou ukládány informace, které je zpětně identifikují. Tato data jsou ukládána do souboru, jenž je zakončen řetězcem „Pre_AT“ (z angl. preprocessing – anomalous timebins) a je uložen v adresáři pro výstupy identifikace (viz. kapitola 6.1).

## 6.3 Detekce anomálií

Druhou aplikací programové soustavy je implementace podprostorové metody pro detekci anomálií, založené na PCA (viz. 5.1.3). Tato detekční část je pro obě varianty stejná. Při implementaci této části jsem se však z důvodu citlivosti PCA vůči definici normálního podprostoru popsané v [14] striktně nedržel definice normálního podprostoru podle [9]. Tuto definici jsem tedy pro snazší experimentování také učinil nastavitelnou, a to třemi možnými způsoby[8]:

- *Pevně nastavená velikost* – při použití této varianty je velikost normálního podprostoru určena napevno, počtem prvních hlavních komponent, jež budou tento prostor utvářet. Podle [11] je takto určována velikost normálního podprostoru počtem prvních 10 hlavních komponent.
- *Velikost zadána v procentech* – zde je podprostor definován počtem prvních hlavních os, jež zachycují minimálně zadané procentuální množství z celkového rozptylu v datech.
- *Velikost určena δ-testem* - v této variantě je velikost normálního podprostoru určena testem, popsaným v kapitole 5.1.2. Proměnným parametrem je zde násobitel hodnoty δ jenž je v původním návrhu roven 3 [9].

Po určení velikosti normálního podprostoru jsou následně detekována časová okna, v nichž se vyskytují anomálie. Pořadová čísla těchto anomálních časových oken jsou ukládána do výstupního souboru, v němž jsou tato čísla oddělená vždy jedním tabulátorem. Tento soubor je pak uložen v adresáři pro výstupy detekce se zakončením názvu souboru „_AT“ (z angl. anomalous timebins). Pro výpočty identifikační části, která byla implementována pro účely ověření jejích základních

[7] Toto nastavení je možné měnit opět změnou makra v hlavním hlavičkovém souboru *main.h*.
[8] Jednotlivé způsoby jsou také ovládány kombinací nastavení maker v hlavním hlavičkovém souboru *main.h*

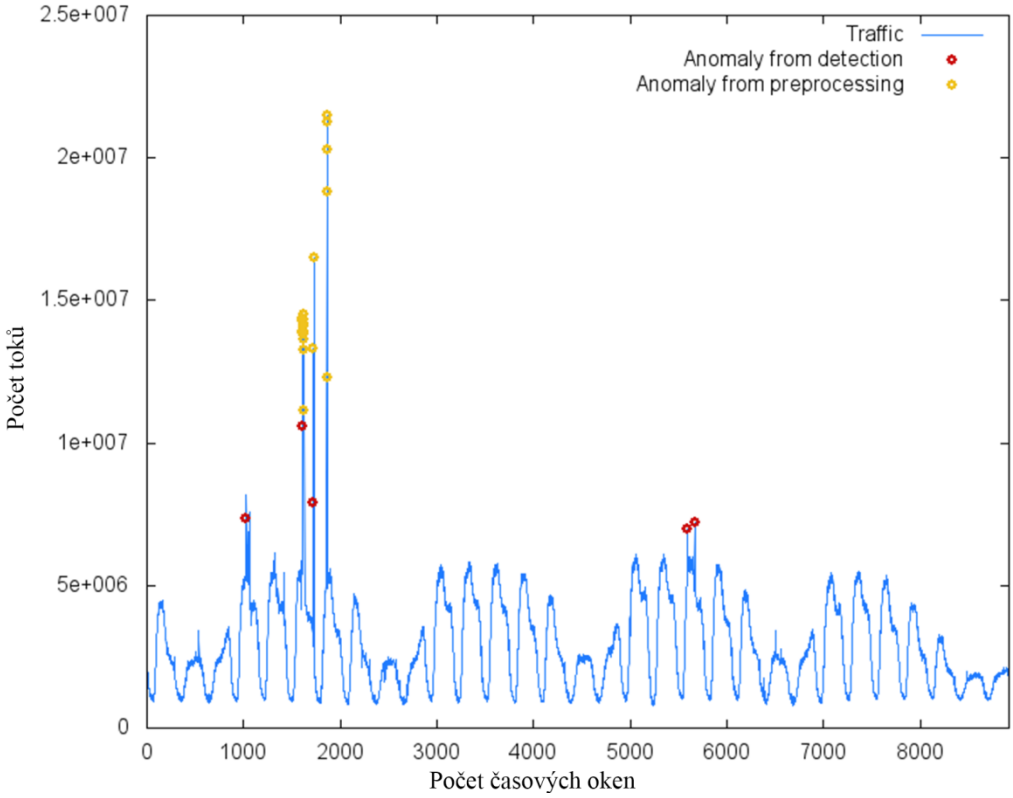
principů, vypisuje tato aplikace ještě do téhož adresáře matici $\tilde{\mathbf{C}}$. Název souboru s touto maticí je zakončen řetězcem „_Cr“ (z angl. C residual). Po zapsání těchto výstupů je aplikace ukončena.

## 6.4 Identifikace anomálií

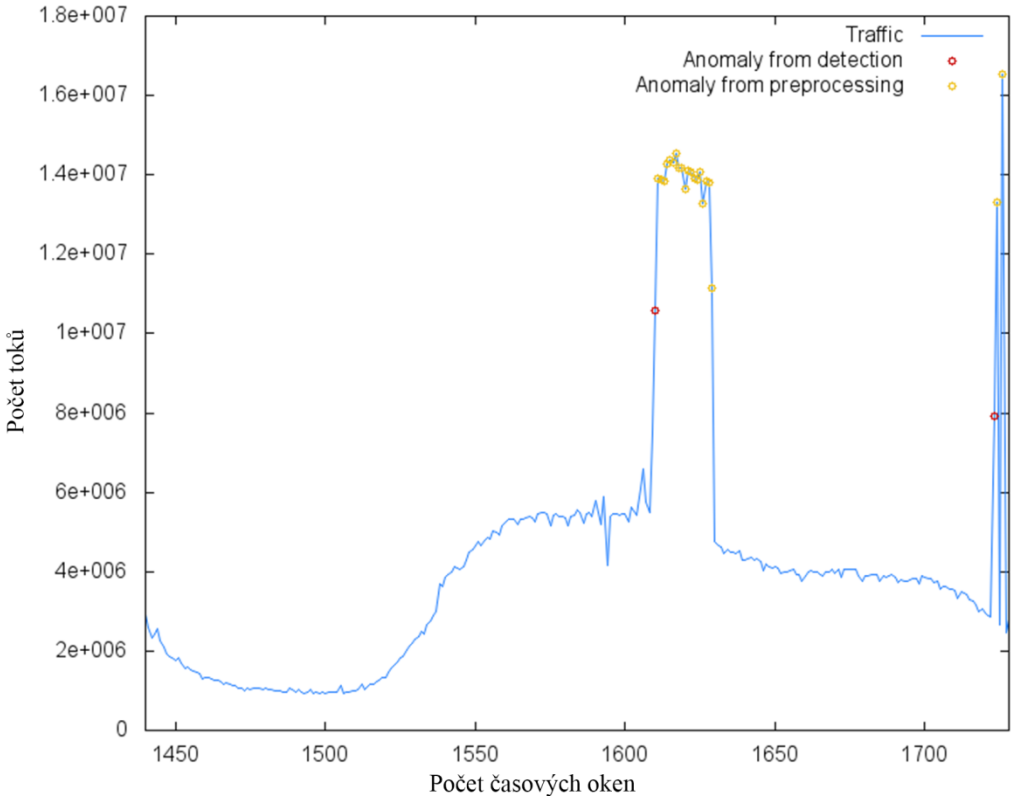
Jak již bylo řečeno v úvodu této kapitoly, identifikační metodu jsem implementoval pouze pro původní variantu podprostorové metody popsané v [9], pracující pouze s objemem dat provozu na síti.

Třetí a poslední část aplikační soustavy, provádí tedy identifikaci anomálních OD toků. Začíná opět načítáním dat ze vstupních souborů, kterých je zde ze všech tří aplikací nejvíce. Načtena je tedy směrovací matice, datová matice (tak jak byla předána detekční části), matice $\tilde{\mathbf{C}}$ a čísla identifikující anomální časová okna. Za použití těchto dat a postupů popsaných v kapitole 5.1.4, jsou následně identifikovány OD toky, které jsou za danou anomálii odpovědné. Tyto anomálie jsou uloženy také do souboru, stejným způsobem jako u předzpracování, tedy na jednom řádku souboru je vždy číslo časového okna, v němž se vyskytuje anomálie, následováno číslem anomálního OD toku, odděleným třemi tabulátory. Tři tabulátory jsou zde pro lepší optické odlišení sloupců, jelikož tento soubor již není určen pro žádnou další aplikaci, ale pro člověka. Za tyto identifikované anomálie jsou ještě připsány anomálie z předzpracování, získané ze souboru uloženého první aplikací. Soubor s těmito všemi anomáliemi je pak uložen v adresáři pro výstupy identifikace se zakončením názvu „_ANOMALIES“. Tato aplikace však poskytuje kromě textových výsledků ještě jejich grafické zobrazení. Tato funkcionalita však vyžaduje přítomnost funkční aplikace Gnuplot[9]. V grafech je vynesen vždy objem dat v provozu v průběhu času pro každou z linek. Na tomto grafu jsou pak barevnými kroužky zaznačeny anomálie, přičemž jsou barevnými odstíny odlišeny ty z předzpracování dat a ty, jež byly získány na PCA založenou detekcí. Pro případ velkého počtu časových oken, kdy se může zdát takovýto graf příliš stěsnaný, umožňuje aplikace navíc vyobrazení po určitých intervalech. Příklady těchto grafů jsou zobrazeny na následujících obrázcích. Na prvním z nich (Obrázek 6.2) je zobrazen základní graf, tedy graf provozu na jedné lince v průběhu sledovaného času (všech časových oken). Zde je tímto provozem objem dat na jedné z linek (nix3_1) sítě Cesnet za období od 1. 3. 2013 do 31. 3. 2013, rozděleného do 5 minutových časových oken. Druhý obrázek (Obrázek 6.3) zobrazuje zmiňovaný detail, kdy byly mimo hlavního grafu ukládány i grafy po jednotlivých dnech. Obrázek 6.3 pak zobrazuje údaje ze dne 5. 3. 2013, téže linky jako Obrázek 6.2. Po uložení všech těchto výstupů je identifikační část ukončena

[9] Při testování aplikace bylo pracováno s verzemi Gnuplot 4.0, 4.6 a 4.7.




*Obrázek 6.2 Objem dat v provozu na lince nix3_1 sítě Cesnet v průběhu března 2013 s vyznačenými anomáliemi detekovanými aplikační soustavou.*




*Obrázek 6.3 Detail objemu dat v provozu na lince nix3_1 sítě Cesnet ze dne 5. 3. 2013 s vyznačenými anomáliemi detekovanými aplikační soustavou.*

# 7 Vyhodnocení výsledků

Abych mohl posoudit funkčnost metody popsané v kapitole 5 a také citlivost PCA vůči nastavení parametrů, provedl jsem sérii testů s různým nastavením nad oběma implementovanými verzemi podprostorové metody. Testování mi komplikoval nedostatek dat, která by byla přesně shromažďována pro účely této metody. V následujících podkapitolách tedy budou nejdříve představena tato data. Dále budou popsány provedené testy a způsob jejich vyhodnocení. V posledních dvou podkapitolách pak budou zobrazeny výsledky každé z variant podprostorové metody, tedy jednak pro objem dat v provozu, jednak pro agregaci IP toků do náhodných uskupení.

## 7.1 Testovací data

Všechny testy byly provedeny nad daty ze sítě Cesnet. Pro objemovou variantu pak nad daty z celého měsíce března 2013 a pro variantu náhodných agregací nad daty ze dne 4. 3. 2013. V těchto datech bylo možné pozorovat na určitých linkách[10] intenzivní DoS útoky, jimž čelily české servery ve dnech 4. 3. – 7. 3. 2013. Pro objemovou variantu jsem však neměl k dispozici dvojici potřebných vstupních souborů – soubor se směrovací maticí a soubor s informacemi o objemu provozu v jednotlivých OD tocích. Měl jsem však k dispozici přímo základní datovou matici (tedy tu, která je ze dvou předchozích vytvořena), což na výsledky detekční části nemá vliv. Problém byl však s identifikační částí, jež směrovací matici vyžaduje. Pro její potřeby jsem vytvořil tuto matici uměle, jako diagonální matici o rozměrech $m \times m$, kde $m$ udává počet linek. Výsledkem této identifikace pak není určení OD toku, který je za anomálii odpovědný, ale pouze určení linky, na které anomálie probíhá. Nicméně k základnímu ověření principů identifikace se tato úprava ukázala jako postačující. Metrikou pro tuto objemovou variantu byl počet OD toků na lince.

Pro druhou variantu jsem měl k dispozici netflow záznamy o tocích ze dvou linek. Zvolil jsem data z linky nix2, na které probíhaly zmiňované DoS útoky. Tato data však byla velmi objemná, a jejich zpracování zabralo hodně času, což také komplikovalo průběh testování.

## 7.2 Metodika testování a způsob vyhodnocení

V průběhu prvotních testů a ladění aplikací programové soustavy jsem postupně vyčlenil skupinu parametrů, s jejichž nastavením bylo dosaženo poměrně dobrých výsledků. Zároveň jsem také vyřadil parametry metody, jejichž použití dávalo vždy velmi špatné výsledky. Navíc jsem ještě také vždy začlenil kombinaci parametrů, odpovídající původním návrhům obou variant podprostorové metody. Pro obě varianty zahrnovaly testy zaprvé různá nastavení předzpracování dat:

- Bez předzpracování
- Předzpracování s odstraněním anomálií, které jsou větší či menší než střední hodnota vzorku dat o 4δ.
- Předzpracování s odstraněním anomálií, které jsou větší či menší než střední hodnota vzorku dat o 5δ.

[10] Linky nix2, nix2_1,nix3, nix3_1 sítě Cesnet

Dále pak pro každou z těchto tří variant byly nastaveny následující parametry definice normálního podprostoru:

Normální podprostor definován prvními hlavními osami, obsahujícími alespoň:

- 75% celkového rozptylu v datech.
- 80% celkového rozptylu v datech.
- 85% celkového rozptylu v datech.
- 90% celkového rozptylu v datech.

Normální podprostor určen δ-testem podle 5.1.2 pro:

- 3δ.
- 4δ.
- 5δ.

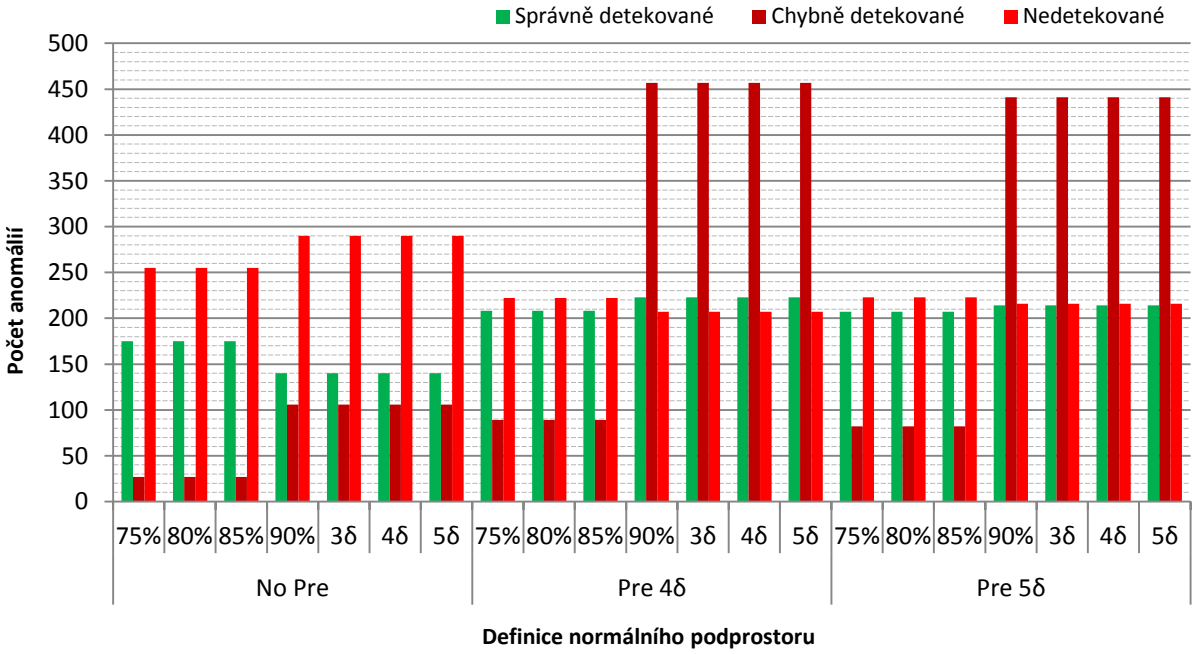
Abych mohl provést vyhodnocení detekovaných anomálií, prošel jsem podrobné grafy provozu těchto dat a ručně vyhledal výrazné odchylky v těchto grafech, představující anomálie. Kvalitativní posouzení obou variant pak bylo zachyceno pomocí hodnot míry správně detekovaných událostí (angl. true positive ratio, dále TPR) a míry falešných poplachů (angl. false positive ratio, dále FPR), vypočítaných z výsledků detekce a ručně nalezených anomálií.

## 7.3 Výsledky objemové varianty

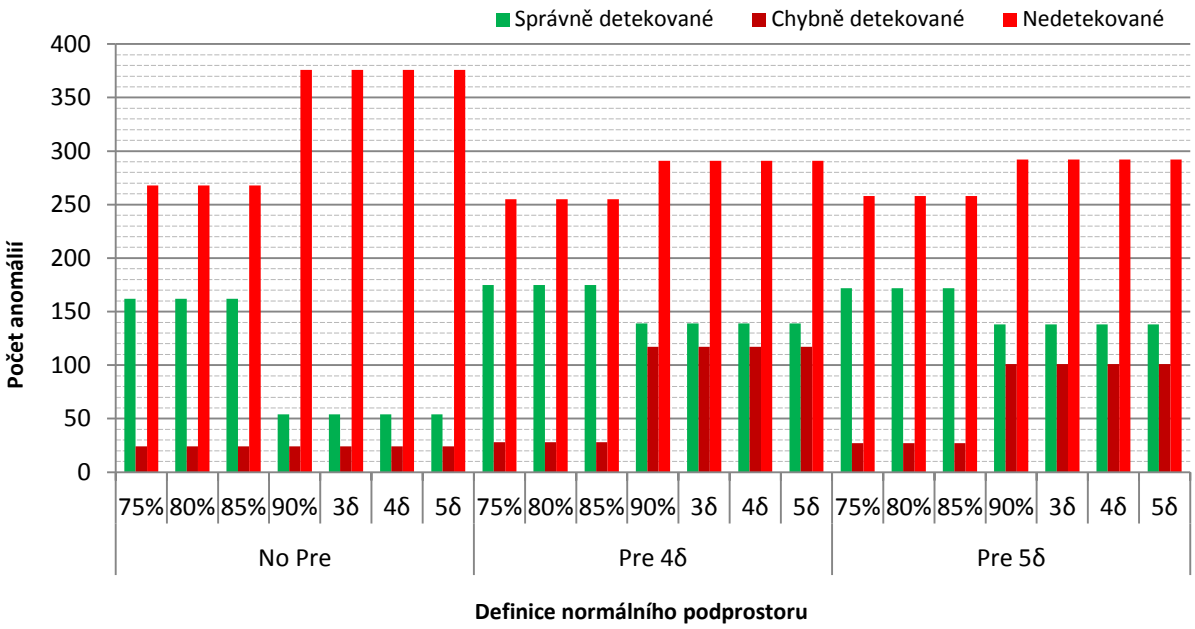
Výsledky původní varianty podprostorové metody pro objem dat v provozu zobrazuje Obrázek 7.1 a Obrázek 7.2. Grafy na těchto obrázcích zobrazují počty událostí, které metoda úspěšně určila jako anomálie (zelený sloupec), chybně označila za anomálie (tmavě červený sloupec) a těch, jež byly anomáliemi, ale nebyly detekovány metodou. Obrázek 7.1 zobrazuje počty pro míru spolehlivosti detekce[11] 95%, Obrázek 7.2 pak pro míru spolehlivosti 99%. Na obrázcích je možné vidět, že pro každé předzpracování jsou zde vždy pouze dvě sady výsledků – první pro velikost normálního podprostoru určenou prvními hlavními osami, obsahujícími alespoň 75%, 80% a 85% celkového rozptylu v datech a druhou pro velikost normálního podprostoru určenou jednak prvními hlavními osami, obsahujícími alespoň 90% celkového rozptylu v datech, jednak δ-testem pro 3δ, 4δ a 5δ. Tento jev je způsobem zmiňovaným nedostatkem dat, určených pro tuto metodu. Namísto relativně vysokého počtu OD toků je zde pracováno s linkami, kterých je pouze 9. To dává na vstupu metody PCA pouhých 9 proměnných. Je zde tedy velmi pravděpodobné, že i přes různé způsoby nastavení velikosti normálního podprostoru, bude normální podprostor nakonec definován jedním a tím samým počtem prvních hlavních komponent.

Úspěšnost detekce objemových anomálií podprostorové metody shrnuje Obrázek 7.3. Na tomto obrázku je graf, znázorňující poměr TPR a FPR. Je zřejmé, že optimální jsou vysoké hodnoty TPR a současně nízké hodnoty FPR. V tomto grafu již nejsou vyznačeny body pro všechny kombinace testů, ale pouze skupiny pro nastavení parametrů, s nimiž bylo dosaženo stejných výsledků. Tyto skupiny shrnuje Tabulka 7.1.

[11] Parametr $\alpha$ rovnice (1)




*Obrázek 7.1 Výsledky detekce podprostorové metody pro míru spolehlivosti 95%*

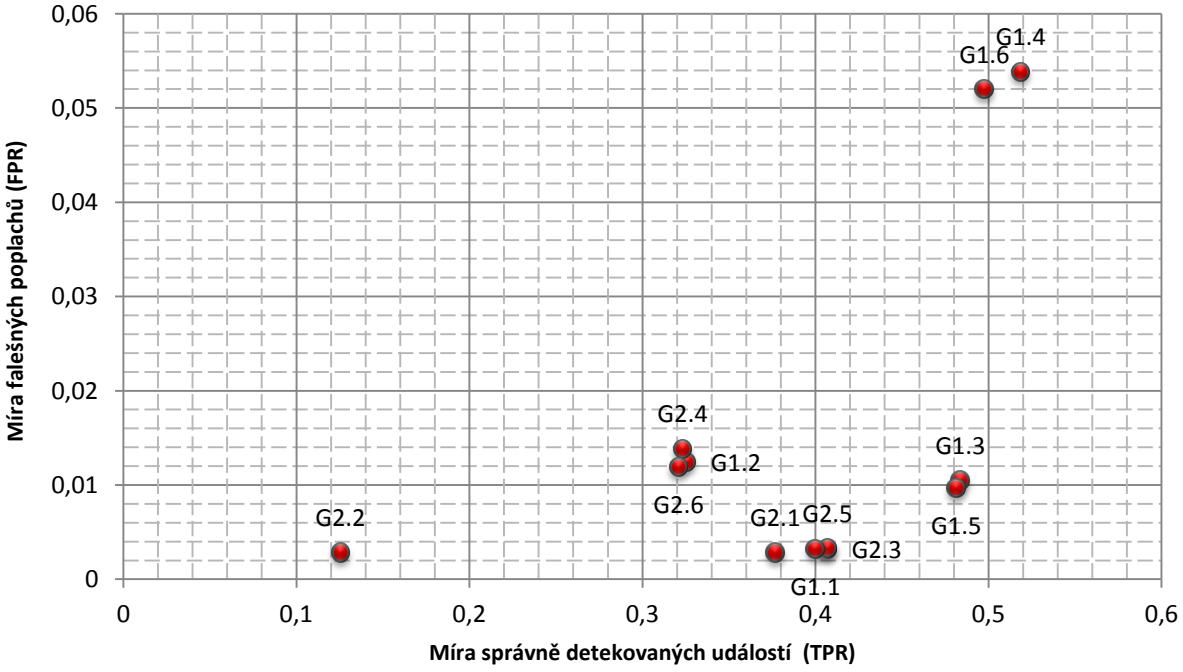



*Obrázek 7.2 Výsledky detekce podprostorové metody pro míru spolehlivosti 99%*

Přestože výsledky nejsou dokonalé, je vidět, že metoda je schopná objemové anomálie úspěšně detekovat. Zejména bych pak vyzdvihl nastavení detekce s předzpracováním v obou uvedených variantách, první skupiny definice normálního podprostoru a obou úrovní spolehlivosti. Tedy pro předzpracování odstraňující výrazné anomálie, které jsou větší či menší než střední hodnota vzorku dat o 4δ či 5δ, velikost normálního podprostoru určenou prvními hlavními osami, obsahujícími alespoň 75%, 80% a 85% celkového rozptylu v datech a pro míru spolehlivosti detekovaných událostí 95% a 99%.

| Označení skupiny | Míra spolehlivosti | Způsob předzpracování | Definice normálního podprostoru |
|---|---|---|---|
| **G1.1** | 95% | Žádné | 75%; 80%; 85% |
| **G1.2** | | | 90%; 3δ; 4δ;5δ |
| **G1.3** | | 4δ | 75%; 80%; 85% |
| **G1.4** | | | 90%; 3δ; 4δ;5δ |
| **G1.5** | | 5δ | 75%; 80%; 85% |
| **G1.6** | | | 90%; 3δ; 4δ;5δ |
| **G2.1** | 99% | Žádné | 75%; 80%; 85% |
| **G2.2** | | | 90%; 3δ; 4δ;5δ |
| **G2.3** | | 4δ | 75%; 80%; 85% |
| **G2.4** | | | 90%; 3δ; 4δ;5δ |
| **G2.5** | | 5δ | 75%; 80%; 85% |
| **G2.6** | | | 90%; 3δ; 4δ;5δ |

*Tabulka 7.1 Skupiny nastavení parametrů se shodnými výsledky*




*Obrázek 7.3 Poměr TPR a FPR se zvýrazněnými pareto-optimálními body*

Mnoho nedetekovaných anomálií nebylo detekováno při žádném z testů, tedy při žádném z nastavení podprostorové metody. To bylo pravděpodobně způsobeno tím, že mnoho těchto anomálií se vyskytuje na všech linkách současně. Metoda PCA však porovnává pouze hodnoty z linek v jednom časovém okně a nepracuje žádným způsobem s historií, nemůže tedy již z principu metody takovéto anomálie detekovat. Při praktickém nasazení by proto bylo vhodné doplnit tuto metodu nějakou další, která by byla schopna detekovat neobvyklé změny v provozu na určité lince v čase.

## 7.4 Výsledky varianty náhodných uskupení

U varianty podprostorové metody rozdělující IP toky do náhodných uskupení byla provedena stejná sada testů, jako u varianty předchozí. Navíc zde byla zařazena ještě varianta definující normální podprostor přímo počtem prvních 10 hlavních komponent, tak jak uvádí [11]. Výsledky této metody však byly velmi špatné. Správně detekovaných událostí bylo ve všech testech velmi málo (v řádu jednotek) a vždy za cenu vysoké míry falešných poplachů. Z toho lze usuzovat, že tyto správné detekce byly spíše pouhou náhodou. Z tohoto důvodu zde také neuvádím žádné další grafy, které nemají v tomto případě žádnou další informační hodnotu. Výsledkem mého studování neúspěchu této modifikace původní podprostorovém metody, které by mělo být jejím vylepšením, je zjištění dvou možných příčin.

První z nich je založena na předpokladu, že dříve zmiňovaný DoS útok, který se vyskytoval i v testovaném vzorku dat, není způsoben malou skupinou toků, nýbrž miliony různých toků. Takovýto útok se pak rovnoměrně rozloží do všech náhodných uskupení, entropie bude všude ovlivněna víceméně stejně a PCA pak logicky neoznačí žádnou anomálii.

Druhá teorie vychází z poznatku v [14], jenž uvádí, že definice normálního podprostoru PCA může být nepříznivě ovlivněna („znečištěna“) příliš intenzivní anomálií. To by mohl být i případ testovaného vzorku dat, kdy je provoz vytvořený zmiňovaným DoS útokem oproti zbývajícímu provozu velmi výrazný. Může tak ovlivnit definici normálního podprostoru takovým způsobem, že metoda považuje spíše anomálii za normální a naopak normální provoz za anomální. Tuto teorii také částečně potvrzují výsledky testování, kdy jsem zejména u variant bez předzpracování pozoroval, že většina neoznačených časových oken, náležela právě k tomuto DoS útoku.

Testování této modifikace původní podprostorové metody přineslo tedy negativní výsledky. Ty však odhalují důležité poznatky o použití metody PCA pro detekci anomálií v počítačových sítích.

# 8 Závěr

Cílem této práce byla implementace metody detekce anomálií v počítačových sítích, založená na PCA a vyhodnocení jejích výsledků. V rámci této práce byla tedy implementována metoda popsaná v [9], která se nazývá *podprostorová metoda*. Tato metoda však pracuje pouze s objemem dat v provozu na síti, jenž nepokrývá poměrně velké množství anomálií, které se nemusí ve velkém objemu dat v provozu na páteřních linkách výrazněji projevit. Byla proto studována další rozšíření této základní metody popsaná v [10] a v [11]. Tyto metody vylepšují původní podprostorovou metodu použitím sofistikovanějšího pohledu na provoz na síti a dalších přístupů, za účelem dosažení vyšší míry detekovaných anomálií a nižší míry falešných poplachů. Z těchto vylepšení pak bylo implementováno to, jež mělo být nejvyspělejší. Toto vylepšení bylo popsáno v [10] a nese název *Defeat*. Jak ale ukázaly následující testy provedené nad sadou různých parametrů detekční metody, má tento přístup velké nedostatky. Na tyto nedostatky poukazuje i článek [14]. Nicméně testování původní podprostorové metody ukázalo, že tato metoda, založená na analýze hlavních komponent, je schopna určité anomálie obstojně detekovat.

Tím se otevírají další možnosti navazující práce. Analýza hlavních komponent se ukázala pro detekci anomálií v síťovém provozu jako funkční, nicméně velmi citlivá na nastavení jejích parametrů a také ovlivnitelná velmi intenzivními anomáliemi. Vliv těchto parametrů a jejich automatické nastavení podle podmínek dané sítě je jednou z možností dalšího výzkumu. Dále je pak možné studovat způsoby potlačení vlivu intenzivních anomálií na definici normálního podprostoru, např. dokonalejší metodou předzpracování dat. Zároveň by také nová testovací data s dobře zdokumentovanými anomáliemi, přizpůsobená přímo potřebám metody, mohla přinést nové poznatky o chování této metody.

Tato práce tedy předkládá nové poznatky o metodě detekce anomálií na základě analýzy hlavních komponent. V rámci této práce je také k dispozici aplikační soustava, umožňující provádění experimentů, pro jejichž parametrizování poskytuje snadný způsob změny nastavení parametrů detekční metody. Výsledky všech testování ukazují, že metoda, jíž jsem se zabýval, je ještě nedokonalá a způsoby jejího zdokonalení ponechávám pro další výzkum.

# Literatura


[1] *Skenování portů* [online]. Naposledy upraveno: 5.4.2013 [citováno 2013-03-15]. Dostupné z: <http://cs.wikipedia.org/wiki/Skenování_portů>

[2] *Strategies to protect against distributed denial of service attacks* [online]. Naposledy upraveno 22/4/2008 [cit. 2013-03-12]. Dostupné z: <http://www.cisco.com/en/US/tech/tk59/technologies_white_paper09186a0080174a5b.shtml>

[3] BARTOŠ, Václav a ŽÁDNÍK, Martin *Framework for comparison of network anomaly detection algorithms*. Technický report č. FIT-TR-2012-02, Fakulta informačních technologií v Brně, Vysoké učení technické v Brně. 2012.

[4] HALLER, Martin *Denial of service (DoS) útoky: záplavové typy* [online]. 9/2006 [cit. 2013-03-11]. Dostupné z: <http://www.lupa.cz/clanky/denial-of-service-dos-utoky-zaplavove-typy/>

[5] HALLER, Martin *Skenování portů: teorie* [online]. 10/2006 [cit. 2013-03-15]. Dostupné z: <http://www.lupa.cz/clanky/skenovani-portu-teorie/>

[6] HALLER, Martin *Skenování portů: techniky* [online]. 10/2006 [cit. 2013-03-15]. Dostupné z: <http://www.lupa.cz/clanky/skenovani-portu-techniky/>

[7] MATOUŠEK, Petr. *12. Měření na síti pomocí NetFlow* [přednáška]. Studijní matriály FIT VUT v Brně, 2012.

[8] MELOUN, Milan. *Počítačová analýza vícerozměrných dat v oborech přírodních, technických a společenských věd* [online prezentace]. 6/2011 [cit. 2013-04-09]. Dostupné z: <http://www.crr.vutbr.cz/system/files/prezentace_05_1106_04a.pdf>.

[9] LAKHINA, Anukool; CROVELLA, Mark and DIOT, Christophe Diagnosing network-wide traffic anomalies. In: ACM SIGCOMM, Portland (Oregon, USA), 2004.

[10] LAKHINA, Anukool; CROVELLA, Mark and DIOT, Christophe Mining anomalies using traffic features distributions. In: ACM SIGCOMM, Philadelphia (Pennsylvania, USA), 2005

[11] LI, Xin; BIAN, Fang; CROVELLA, Mark and DIOT, Christophe; GOVINDAN, Ramesh; IANNACCONE Gianluca a LAKHINA, Anukool Detection and identification of network anomalies using sketch subspaces. In: ACM Internet measurement conference, Rio de Janeiro (Brazílie), 2006.

[12] PATRIKAKIS, Charalampos; MASIKOS, Michalis and ZOURARAKI, Olga Distributed denial of service attacks. *The internet protocol journal*. 2004, volume 7, number 4. ISSN 1944-1134.

[13] PŘIBYL, Tomáš. *Červ: nepřítel sítě číslo jedna* [online]. Dostupné z: <http://www.ictsecurity.cz/odborne-clanky/cerv-nepritel-site-cislo-jedna.html>

[14] RINGBERG, Haakon; REXFORD, Jennifer; SOULE, Augustin and DIOT, Christophe Sensitivity of PCA for traffic anomaly detection. In SIGMETRICS, San Diego (California, USA), 2007.

[15] ROUSE, Margaret *Network scanning*. 9/2005 [cit. 2013-16-3]. Dostupné z: <http://searchmid marketsecurity.techtarget.com/definition/network-scanning>

[16] SARVOTHAM, Shriram; RIEDI, Rudolf and BARANIUK, Richard Conection-level analysis and modeling of network traffic. In: ACM SIGCOMM Internet Measurement Workshop, San Francisco (USA), 2001.

[17] CORMODE, Graham and THOTTAN, Marina *Algorithms for Next Generation Networks*. London: Springer-Verlag London Ltd, 2010. Anomaly detection approaches for communication networks, s 239-262. ISBN 978-1-84882-764-6

# Seznam příloh

A Obsah CD
B Systémové požadavky
C Manuál

# A Obsah CD

Na přiloženém CD se nachází zejména zdrojové soubory aplikační soustavy včetně struktury Makefilů, potřebných pro překlad. Tyto soubory se nachází v adresáři [Subspace_method_sources-app_set]. V tomto adresáři se také nachází spustitelné soubory a testovací skripty, pro provedení výše uvedených testů objemové varianty. Testovací skripty pro variantu náhodných uskupení zde chybí, z důvodu velkého objemu zdrojových dat pro tuto variantu. Jedná se jednak o „batch“ soubor, pomocí kterého byly prováděny testy v prostředí Microsoft Windows 7, jednak o „shell“ skript pro, s jehož pomocí byly prováděny testy v prostředí Unix a Linux. Dále se na CD nachází dokumentace k aplikační sadě, jenž obsahuje instrukce pro její nastavení a spuštění, tato zpráva ve formátu PDF i ve zdrojovém formátu docx.

# B Systémové požadavky

Aplikační soustava, vytvořená v rámci této práce, byla testována v prostředích Microsoft Windows 7 (edice Professional, 64 bitová verze, SP1), Unix (distribuce FreeBSD 9.1) a Linux (distribuce CentOS 5.3). Pro přeložení aplikační soustavy je nutné mít nainstalovaný překladač g++. Pro účely vytvoření grafů vyžaduje identifikační část v systému přítomnost programu Gnuplot (testováno na verzích 4.0, 4.6 a 4.7).

# C Manuál

Aplikační soustava je koncipována jako soustava konzolových aplikací. Tudíž se ovládá pomocí parametrů příkazové řádky a souboru s výchozím nastavením cest *Settings.txt*. Nastavení, ovlivňující způsob chování a výpočtů aplikační soustavy (resp. podprostorové metody) je možné měnit úpravou konstant v hlavním hlavičkovém souboru *main.h* (v adresáři *src*). Popis parametrů bude uveden pro každou část aplikace zvlášť.

Pozn. `defCesta` značí vždy patřičnou výchozí cestu, nastavenou v souboru *Settings.txt*.

**DataReader:**

`-h/-H` – vypíše nápovědu k programu.

`-v/-V` – spustí program v módu *volume* (výchozí mód).

`-s/-S` – spustí program v módu *náhodných uskupení* (*sketch*).

`-d [cesta]` – nastaví doplněk výchozí cesty k souboru datové matice (mód *volume*) nebo k adresáři se soubory s NetFlow záznamy (mód *náhodných uskupení*). Výsledná cesta: `defCesta+cesta`.

`-D [cesta]` – nastaví úplně cestu k souboru datové matice (mód *volume*) nebo k adresáři se soubory s NetFlow záznamy (mód *náhodných uskupení*). Výsledná cesta: `cesta`.

`-a [cesta]` – nastaví doplněk výchozí cesty ke směrovací matici. Výsledná cesta: `defCesta+cesta`.

`-A [cesta]` – nastaví úplně cestu ke směrovací matici. Výsledná cesta: `cesta`.

`-n [cesta]` – nastaví doplněk výchozí cesty k výstupní datové matici (pro PCA) a také doplněk cesty k výstupům pro identifikační část. Výsledná cesta: `defCesta+cesta`.

`-M [cesta]` – nastaví úplně cestu pro výstupní datovou matici (pro PCA). Výsledná cesta: `cesta`.

`-I [cesta]` – nastaví úplně cestu pro výstupy pro identifikaci. Výsledná cesta: `cesta`.

**Detector:**

`-h/-H` – vypíše nápovědu k programu.

`-d [cesta]` – nastaví doplněk výchozí cesty k souboru zdrojové datové matice pro PCA. Výsledná cesta: `defCesta+cesta`.

`-D [cesta]` – nastaví úplně cestu k souboru zdrojové datové matice pro PCA. Výsledná cesta: `cesta`.

`-n [cesta]` – nastaví doplněk výchozí cesty pro detekční výstupy. Výsledná cesta: `defCesta+cesta`.

`-N [cesta]` – nastaví úplně cestu pro detekční výstupy. Výsledná cesta: `cesta`.

**Identificator:**

`-h/-H` – vypíše nápovědu k programu.

`-d [cesta]` – nastaví doplněk výchozí cesty k výstupům detekční části (anomální časová okna a C-residual). Výsledná cesta: `defCesta+cesta`.

`-A [cesta]` – nastaví úplně cestu pro k anomálním časovým oknům z detekční části. Výsledná cesta: `cesta`.

`-C [cesta]` – nastaví úplně cestu k matici C-residual z detekční části. Výsledná cesta: `cesta`.

`-r [cesta]` – nastaví doplněk výchozí cesty ke směrovací matici. Výsledná cesta: `defCesta+cesta`.

`-R [cesta]` – nastaví úplně cestu ke směrovací matici. Výsledná cesta: `cesta`.

`-m [cesta]` – nastaví doplněk výchozí cesty k výstupní datové matici (pro PCA) a také doplněk cesty k výstupům pro identifikační část. Výsledná cesta: `defCesta+cesta`.

`-M [cesta]` – nastaví úplně cestu pro výstupní datovou matici (pro PCA). Výsledná cesta: `cesta`.

`-n [cesta]` – nastaví doplněk výchozí cesty k souboru datové matice (mód *volume*) nebo k adresáři se soubory s NetFlow záznamy (mód *náhodných uskupení*). Výsledná cesta: `defCesta+cesta`.

`-N [cesta]` – nastaví úplně cestu k souboru datové matice (mód *volume*) nebo k adresáři se soubory s NetFlow záznamy (mód *náhodných uskupení*). Výsledná cesta: `cesta`.