MASARYKOVA UNIVERZITA
FAKULTA INFORMATIKY

Offline vizualizace NetFlow dat

BAKALÁŘSKÁ PRÁCE

Aleš Novák

Brno, jaro 2007

Prohlášení

Prohlašuji, že tato bakalářská práce je mým původním autorským dílem, které jsem vypracoval samostatně. Všechny zdroje, prameny a literaturu, které jsem při vypracování používal nebo z nich čerpal, v práci řádně cituji s uvedením úplného odkazu na příslušný zdroj.

Vedoucí práce: Mrg. Pavel Minařík

Poděkování

Chtěl bych poděkovat Mgr. Pavlovi Minaříkovi a Mgr. Tomáši Dymáčkovi za to, že mně umožnili dělat tuto práci a za jejich přístup a podporu, které ji provázely.

Shrnutí

Cílem této práce je zhodnotit dosavadní techniky analýzy vizualizovaných dat, která vznikají jako výstup z monitorování sítí a vytvořit nástroj, který takovou vizualizaci provádí.

Klíčová slova

NetFlow, nfdump, vizualizace, infoVis, prefuse, java, Flow

Obsah

Kapitola 1

Úvod

1.1 Rozsáhlé sítě a jejich monitorování

Propustnost sítí rapidně roste a s tím se nese zvyšování počtu služeb sítěmi poskytovaných. Je zřejmé, že větší objemy různých typů služeb přináší větší riziko narušení integrity sítě. Proto je nutné monitorovat přenosy na síti, aby se poznalo, jakým způsobem a kam směřují.
Výsledky monitorování nastiňují možnosti, které pomáhají zvyšovat efektivitu a spolehlivost sítí a redukovat investice nutné pro jejich výstavbu, počínaje návrhem a optimalizací služeb na síti a a konče detailním pochopením chování sítě. Pokud známe podrobnosti o chování sítě, jsme schopni s jistotou implementovat nové IP služby.
Sledování sítě je možné rozdělit na dva různé typy dané cílem, kterého chceme dosáhnout, respektive anomáliemi, kterým chceme zabránit. Monitorovací software je postaven pro sledování sítě za účelem odhalení narušení systému, měl by tedy poukázat na možné hrozby ať už zvenčí nebo zevnitř sítě. Na druhé straně také monitoruje síť, aby zjistil příčinu přetížení nebo havárie serveru a jiných zařízení, případně porušení síťových spojení.
Řešení těchto požadavků nabízí technologie NetFLow[4] vyvíjená společností Cisco Systems[1]. Jedná se o techniku sběru a měření dat, procházejících routery. Analyzování NetFlow dat umožňuje zjistit příčinu zahlcení, určit CoS (Class of Service - 3 bitové číslo, specifikující prioritu mezi hodnotami 0 a 5 včetně[2]) pro každého uživatele a aplikaci a identifikovat zdrojovou a cílovou síť přenosu. O této technologii pojednává podrobněji část 1.3.

1.2 Zpracování výsledků monitorování

Výsledkem monitorování jsou logovací záznamy o milionech řádků. Tyto výsledky jsou tedy dostupné, ale pro složitější analýzy obtížně použitelné. Řešením je jejich inteligentní zpřístupnění, především vizualizace. A tato vizualizace je cílem této práce.
Základní motivací pro vývoj vizualizací je fakt, že člověk je schopen kognitivně zpracovat jen omezenou míru informací. Tato míra u většiny lidí určitě není tak velká, aby člověku umožnila pojmout milion řádků výstupních dat. Proto je nutné data zpřehlednit tak, aby se s omezenou mírou kognitivních schopností dalo pracovat s větším množstvím informací. Problém se netýká jen většího množství dat. Každý preferuje jiný typ výstupu jako popis zkoumaných dat a pokud by řešením bylo generování rozdílných výstupů pro jednotlivé žádosti, mohlo by docházet k odlišným interpretacím stejné analýzy. Čím více se počítače zrychlují a způsoby návrhu i aplikace jimi vytvořené se zdokonalují, zvyšuje se také snaha o úsporu lidských prostředků nahrazením počítačovým výpočtem. Proto není vždy nezbytné požadovat po správcích systémů, aby se vyznali v různých formátech reprezentací výstupních dat a byli schopní hledat v nich souvislosti.

1.3 Principy vizualizace

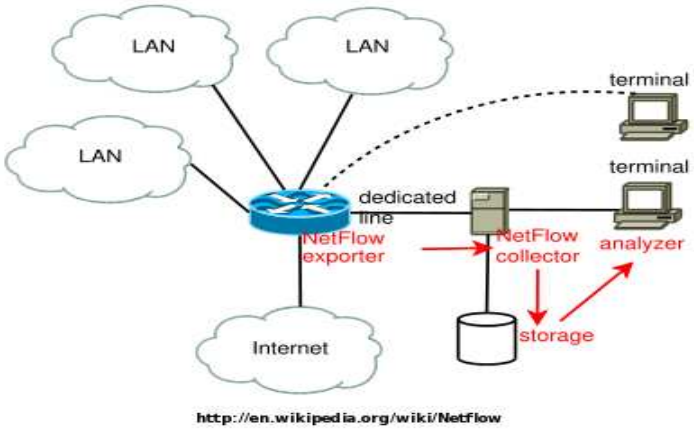
Vizualizace je proces transformací informací do vizuální formy, umožnující uživateli pozorovat a procházet požadovaná data. Vizualizace se snaží o to, aby její zobrazení dávalo bez dlouhého zkoumaní smysl a aby mu uživatel rozuměl a dokázal si udělat celkový obraz o vizualizovaných datech. Je proto důležité, aby bylo na první pohled jasné, co který prvek reprezentuje.
Aby bylo možné dosáhnout co nejpřesnější interpretace informace, je nutné docílit toho spojením člověka a počítače ve smyslu použití počítačové schopnosti a kapacity pro zpracovaní dat a jejich relevantní zobrazení a využití lidského potenciálu pro intuitivní vytváření souvislostí. Není možné se spolehnout na to, že vizualizace správně vystihne a vyzdvihne všechny hledané souvislosti. To proto, že

pravidla pro hledání takových souvislostí jsou často velmi složitá a těžká (nebo spíše často nemožná) definovat. Navíc je to dáno tím, že se tyto pravidla mohou měnit podle povahy zobrazovaných dat. Proto využíváme vizualizace pro zobrazení souvislostí, vycházejících ze známých pravidel a vztahů.
A uživateli dáváme prostor pro vstřebání těchto souvislostí a zároveň pro nalezení dalších, metodami jemu zcela přirozenými, ale neimplementovatelnými.

1.4 NetFlow

1.4.1 Základní přehled

NetFlow je otevřený, ale patentovaný síťový protokol vyvíjený společností Cisco Systems pro zařízení podporované systémem **Cisco IOS** (Internetwork Operating System[3]). Představuje poměrně jednoduchou cestu, jak získat komplexní informace o přenosu IP paketů v síti. Analyzováním těchto informací je možné identifikovat příčinu zahlcení sítě, určit CoS pro každou aplikaci a uživatele, identifikovat zdrojovou a cílovou síť provozu. atd. Tato analýza tedy nabízí možnosti například pro evidenci síťového provozu, účtování na základě míry používání, plánovaní rozvoje sítí, zajištění bezpečnosti, monitorování útoků typu DOS (Denial Of Service) a monitorování provozu na síti.
Postup této metody začíná u routeru, který generuje **NetFlow záznamy**. Ty se poté z routeru exportují pomoci protokolu **UDP** nebo **SCTP** (Stream Control Transmission Protocol), pokud je požadována vetší spolehlivost a hlavně bezztrátovost přenosu.
Tento export sbírá netflow kolektor, který jej hromadí a poskytuje k analýze.



Následují důležité verze NetFlow, ve kterých jsou informace exportovány.
Verze **1** je původní verze podporovaná prvním vydáním NetFlow. Další je Verze **5**, která je rozšířena o **BGP** (Border Gateway Protocol), obsahuje tedy navíc informace o komunikaci autonomních systémů. Navíc je přidáno sekvenční číslování každého toku. Verze **7** je používaná exkluzivně pro Cisco switche a je tedy nepoužitelná pro routery. **8**. verze přidává možnost agregačních schémat, seskupujících toky podle určených kriterií, například pár zdrojová adresa a číslo portu. To snižuje požadavky na šířku pásma mezi routerem a stanicí, která NetFlow exporty spravuje, také počet takových stanic nutných na zpracovaní exportů a dovoluje snazší rozšiřitelnost vysokorychlostních routerů.

1.4.2 Network flows, Netflow record, nfdump

Pokud se zabýváme NetFLow exportem je důležité znát formát, ve kterém NetFlow kolektory výstup z routerů přijímají. Základní prvek takového exportu je NetFlow záznam (NetFlow record). Netflow záznam může obsahovat široké množství různých informaci pro daný Flow (tok).

Například struktura záznamu nejpoužívanější verze - NetFlow verze 5 - vypadá takto :

- Číslo verze
- Sekvenční číslo
- Vstupní a výstupní SMNP (Simple Network Management Protocol) indikátory
- Časová značka započatí a ukončení tohoto Flow
- Počet bytů a paketů v tomto Flow pozorovaných
- Hlavičky síťové vrstvy (vrstva 3)
    - Zdrojová a cílová adresa
    - Zdrojový a cílový port
    - IP protokol
    - Hodnota ToS (Type of Service)

Router generuje na výstup NetFlow záznam poté, co je záznam ukončen. To nastává při ukončení TCP sezení. Ukončení může způsobit také zestárnutí Flow (Flow agging). V tomto případě je pro každý Flow uchováván tzv. čítač stáří (agging counter), který je resetován pokaždé, když router zaznamená nový provoz pro daný Flow. Při použití FNF (Flexible NetFLow) mohou být routery nastaveny na exportování Flow v určených časových intervalech, i když Flow ještě není ukončen. Administrátor může měnit nastavení toho intervalu na routeru. V případě omezené paměti je export proveden také při jejím

naplnění.

Export ze sond, podporujících NetFlow formát je zachytávám démonem nfcapd a ukládán ve specifickém formátu. Pro extrakci z toho formátu se používá program pojmenovaný nfdump[24]. Tento program generuje výstup v čitelném nebo strojově zpracovatelném formátu typu CSV. Způsoby použití a možnosti přepínačů jsou vyčerpávajícím způsobem popsány v manuálu k tomuto programu.

Kapitola 2

Existující vizualizační techniky a jejich nedostatky

2.1 Požadavky kladené na kvalitní vizualizační software

Požadavky na vizualizační software se liší v závislosti na cílech výsledné analýzy. Přesto je možné definovat několik základních kritérií, ze kterých bude vycházet následující hodnocení. Kvalitní vizualizace by měla nabídnout přehledný pohled i na řádově desítky tisíc uzlů.
Kvalitní vizualizační nástroj musí směřovat uživatelovu pozornost. Aby toho dosáhl, musí umožnit proces analýzy podle modelu *level of detail[5]*. Základní proces analýzy v tomto modelu se skládá ze tří kroků :

1. Přehled nad celou vizualizací, zobrazovány jsou jen ty nejdůležitejší aspekty vizualizovaného
2. Přiblížení zkoumané části nebo zobrazení vybrané podčásti vizualizovaného
3. Zobrazení detailních informací

Je také nutné poskytnout tento model procesu v obou směrech, kdy je uživatel schopný nazírat na nejpodrobnější detaily jednotlivých položek a ty vidět v souvislosti s ostatními.
Pro směřování pozornosti je také nevyhnutelně nutné implementovat nějaký systém filtrování zobrazovaných dat. Požadovaný vizualizační systém by měl být schopný filtrovat zobrazení podle uživatelem zadaných parametrů zahrnujících nejen filtraci podle IP adresy nebo podsítě, ale také podle všech parametrů vstupních flow. Toto filtrování by nejlépe mělo probíhat jen nad vizualizací bez vlivu na zdrojová data a bez nutnosti tyto data podle nových parametrů konvertovat.

Při zobrazení takovým způsobem by měla být aplikace také dostatečně rychlá a schopná pružné interakce. To se částečně týká toho, co bylo řečeno v úvodu. Aby byl uživatel schopen vyvodit z vizualizace nějaké výsledky a nalézt nové souvislosti, vizualizační software, s kterým pracuje, musí mít dostatečně nízkou dobu odezvy.
Důležité je také dosáhnout intuitivní formy ovládání. Esteticky příjemný vzhled je jen podružná vlastnost, ale například animace používané při změně rozvržení zobrazení jsou užitečné při sledování změn.

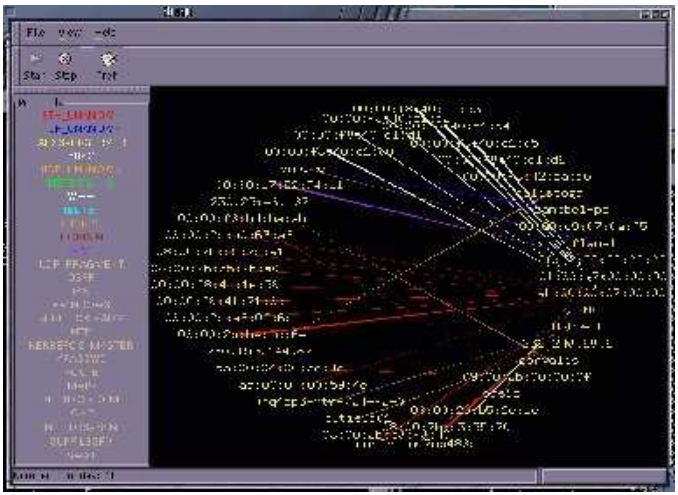
Po konzultacích se síťovým administrátory Ústavu výpočetní techniky Masarykovy univerzity byly určeny také specifické požadavky, které musí vizualizační nástroj nutně splňovat.
V první řadě musí umět podporovat NetFlow exporty jako zdroj dat, protože hledaný program je určen na vizualizaci výstupu ze sond, které s tímto formátem pracují.
Také musí umět odrazit topologii sítě, tedy zdůraznění vztahů a relací, díky kterým je možné najít struktury v přenosu dat, například ring sítě[6], struktury dat, které upozorní na pravděpodobnost výskytu wormů a zatížení sítě použitím p2p stahovačů.
Vizualizace musí podporovat agregaci zobrazované komunikace podle subnetů, IP adres, protokolů a jednotlivých toků.
Filtrování by mělo být řešeno jako omezující podmínky. Filtrování by tak mělo být schopno omezit zobrazení skrytím toků s menší než nastavenou prahovou velikostí, omezit zobrazení na vybrané protokoly a obecně umožnit filtrování dle údajů, které jsou součástí NetFlow dat. V uživatelském rozhraní by to mělo být řešeno dotazovacím jazykem, například se syntaxí TCPdump[7] utility, připraveným formulářem nebo nejlépe kombinací obou způsobů.
Vizualizace musí umět zobrazit detaily jednotlivých toků, nejlépe formou interaktivní tabulky s možností řazení jednotlivých řádků výběrem kritéria (sloupce, dle kterého budou toky seřazeny). Musí ale také umožnit rychlý přehled o stroji zobrazením několika nejčastěji užívaných portů na daném stroji, například zobrazit všechny ve formě hintů dle četnosti použití.

Jako další by měl kvalitní nástroj poskytnout možnost zobrazit stroj a tok s velkým počtem flow ve větší velikosti a vyjádřit tak na první pohled důležitou informaci. IP adresy, které komunikovaly s vybranou IP adresou by bylo vhodné vizualizovat také formou tabulky, ne pouze dynamickou myšlenkovou mapou. Také by bylo zajímavé umožnit definovat významné IP adresy (brány, DNS, poštovní servery, …), které budou vizualizovány odlišným způsobem tak, aby byly na první pohled rozpoznatelné, například změnou tvaru, který je reprezentuje, nebo jeho velikosti, barvy, atd.

2.2 Existující nástroje

2.2.1 EtherApe

EtherApe[8] je opensource grafický monitor sítě, napsaný v c++.
Data mohou být čteny přímo ze sítě – z Ethernet, z FDDI (Fiber distributed data interface)[9], PPP (Point-to-Point Protocol)[10] a SLIP (Serial Line Internet Protocol)[11] zařízení - nebo z výpisu programu tcpdump. Soustředí se na zobrazení komunikace mezi jednotlivými účastníky sítě.
Uzly a hrany, tedy účastníci komunikace a datové toky, jsou zobrazeny barevně podle protokolu, který představují. Velikost uzlů je dynamicky nastavena podle velikosti přenosů, kterých se ten uzel účastnil. Jeden uzel reprezentuje jednoho účastníka, může být ale také rozdělen mezi více uzlů podle jednotlivých portů. Zobrazení uzlů a hran může být také omezeno použitím filtrů. Zakomponovaný je také automatický převod IP adres na odpovídající jména. Pro uzel by měly být poskytnuty detailní informace a statistiky při jeho poklepání myší.



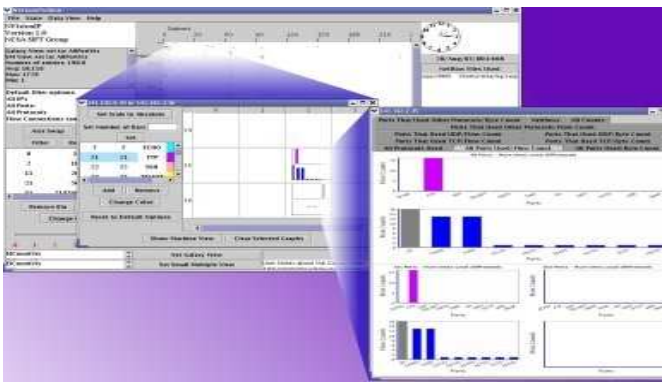
EtherApe
http://etherape.sourceforge.net/images/v0.5.5.png

2.2.2 NvisionIP

NvisionIP[13] je opensource vizualizační nástroj napsaný v javě. Jako zdroj dat používá data v NetFlow formátu. Vstupní data jsou čtena pomocí D2K (Data to Knowledge[12]) data mining aplikačního prostředí. Umí zobrazit množství užitečných informací o celé síti na jediné obrazovce, zobrazit informace o vybraných hostech a detailní informace o jednotlivcích. To díky třem úrovním pohledu : Galaxy View pro pohled na celou síť typu B, Small Multiple View pro zobrazení vybrané množiny hostů a Machine View pro zobrazení detailních informaci o jediném hostu.
Tyto informace obsahují počet bytů a flow pro všechny :

- protokoly
- porty

- TCP přenosy
- UDP přenosy
- přenosy používající jiný protokol než TCP nebo UDP

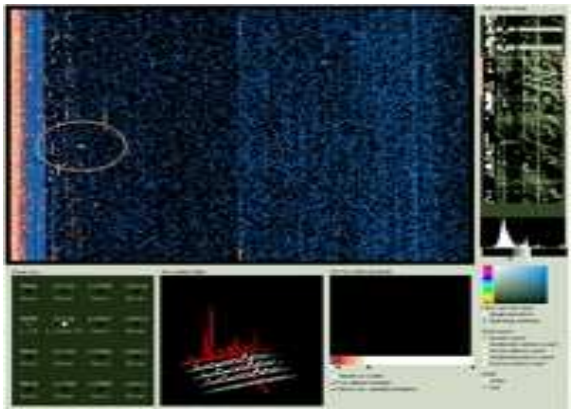


NvisionIP
http://security.ncsa.uiuc.edu/distribution/NVisionDownLoad/pix/AllViewBig.JPG

2.2.3 PortVis

PortVis[14] je zajímavý vizualizační nástroj, který zobrazovaná data agreguje podle TCP portů. Je zajímavý cílem, který si tento projekt předsevzal a to vytvořit vizualizaci vhodnou pro analýzu provozu při nedostatku detailních informací jako například IP adres. PortVis tedy jako vstup v časových intervalech přijímá hrubá data – základní shrnuté informace o aktivitách na jednotlivých TCP portech a snaží se je vizualizovat tak, aby bylo možné odhalit důležité bezpečnostní události.

Bohužel je tento program ještě ve stádiu vývoje a jediné, co se dá o jeho funkcích, kromě výše zmíněného stručného popisu, získat je jeden screenshot demonstrující jeho funkčnost.



PortVis
http://www.cs.ucdavis.edu/~ma/SecVis/ss.jpg

2.2.4 cAIDA

CAIDA[15] vyvíjí mnoho nástrojů pro vizualizaci a analýzu dat na internetu, včetně kolekce, analýzy a zobrazení dat pasivní i interaktivní cestou. CAIDA je projekt internetového atlasu s cílem vyvinout techniky, software a protokoly pro mapování internetu. Projekt kombinuje práce na vývoji s rozsáhlými inženýsrkými a publikačními pracemi.

GeoPlot

Geoplot je javový applet, který uživateli umožňuje vytvořit geografický obraz sady dat. Applet uživateli poskytuje mnoho možností, jak reprezentovat tyto data. Základní funkcionalita je vykreslení uzlů a hran, které je spojují na obrázek definovaný uživatelem.

Gtrace

GTrace je grafický front-end k programu traceroute. Používá kombinaci metod, aby zjistil nebo odhadl fyzickou polohu uzlu v ceste traceroute.

LibSea

LibSea je formát souboru a také Java knihovna pro reprezentaci orientovaných grafů na disku a v paměti.

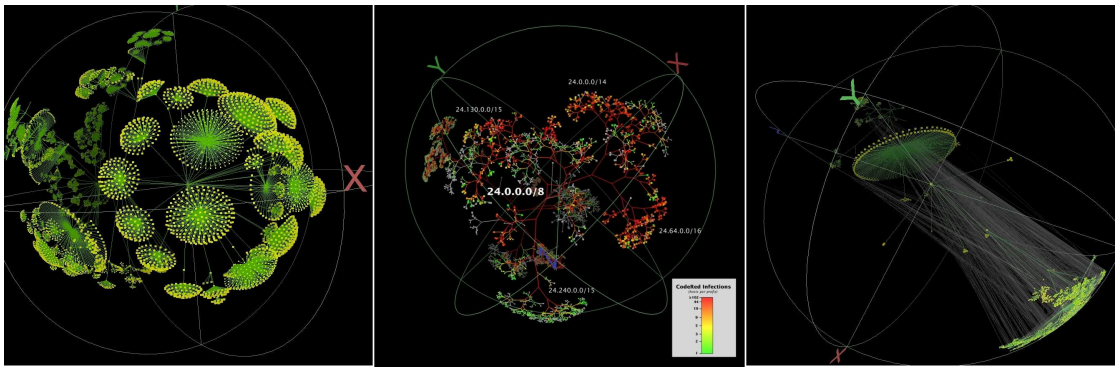
Hlavní cíl byla škálovatelnost grafů čítajících až milióny uzlů. Další cíle tohoto projektu byly kompaktní a smysluplná vizualizace analyzovaných dat a podpora pro specifické aplikační konvence a strategie.

Plot-latlong

PlotPath zobrazuje dopřednou a obrácenou cestu dat uvnitř sítě směřující z jednoho zdroje k více cílům.

Walrus

Tento nástroj je nabízí interaktivní vizualizaci velmi rozměrných orientovaných grafů. Walrus je zajímavý svou velice kvalitní prostorovou reprezentací vizualizovaných dat. Pro jeho spuštění je třeba použít OpenGL verzi Java3D knihovny. Mezi užitečné rysy tohoto programu patří možnost přidat uzlům popisek, zobrazit detailní informace o uzlu, zobrazení vybrané množiny uzlů k analýze a přibližování a oddalování zobrazení.



Walrus
http://www.caida.org/tools/visualization/walrus/gallery1/

2.3 Shrnutí

V předchozím bodě bylo popsáno několik nejpoužívanějších a pro dané téma vhodných nástrojů. PortVis je velice perspektivní projekt, ale zatím je ve vývoji a nedají se zjistit přesné parametry jeho vizualizace. Navíc tento program není vhodný pro požadovanou analýzu, jde spíše o specifický vizualizační nástroj pro reprezentaci dat ochuzených o detailní informace. Kromě Walrus jsou aplikace vyvýjené v rámci projektů cAIDA jednoduší nástroje pro specifické zobrazení, nevhodné pro složitější komplexní analýzu. Ze všech výše popsaných vynikají tři nástroje vizualizace : EtherApe, NvisionIP a Walrus.

EtherApe je chytrý vizualizační program schopný poskytnout jeho uživateli množství užitečných informací v relevantním kontextu, ale pro záměry této práce je nevhodný nedostatkem některých vlastností. Podporuje několik typů vstupních dat, ale nepatří mezi ně podpora pro zobrazení NetFlow informací, což ho zcela vyřazuje z uvažování. Chybí možnost kombinovat více způsobů zobrazení, možné jsou jen některé specifické sledující odhalení určitých anomálií. Tento nástroj také není vhodný pro vizualizaci velkého množství uzlů, což je jeden základních požadavků, popisovaných v bodě 2.1.

NvisionIP je vizualizační nástroj, který dokáže prezentovat širokou škálu informací o charakteristikách celé sítě typu B. Data pro vizualizaci načítá z NefLow záznamů. Zobrazovaná data umožňuje filtrovat a agregovat na základě množství parametrů, které jsou důležité pro analýzu bezpečnosti. Implementuje také model level of detail, kdy může být zobrazena celá síť, množina vybraných uzlů nebo detailní informace o jednotlivých z nich. Pro operace související s těmito procesy nabízí velmi intuitivní ovládání. Na druhou stranu detailní informace jsou zobrazeny jen sloupcovými a jinými statistickými grafy, přičemž se strácejí souvislosti s jinými detailními statistikami. I když tento nástroj implementuje model level of detail, každý krok procesní analýzy umožňuje náhled na anlyzovaná data spíše statisticky a neodráží topologii sítě. Jedním z nevýhod tohoto programu je pak také nemožnost exportu do jakéhokoliv formátu obrázku, čímž se také snižují možnosti prezentování analyzovaných dat. Při velkém množství dat je tento nástroj také velmi pomalý.

Walrus je velmi zajímavý nástroj pro vizualizaci. V tomto přehledu je ale spíše jen jako zajímavý pohled na trochu alternativní cestu vizualizace. Neposkytuje funkce pro načtení dat z jiných formátů souborů než LibSea grafický formát[16]. Také neumí zobrazovat násobné hrany mezi uzly. Tento projekt se také nezdá být aktivní, poslední aktualizace na webových stránkách je 2. března 2006, proto se nedá počítat s jeho živým vývojem ani s podporou komunity. Zobrazován je tady také moc velký kontext, který je vhodný pro specifickou analýzu dat. Jedná se o takzvaný citový náhled, kdy si uživatel začíná pokládat otázky až poté, co výslednou vizualizaci vidí.

Žádný z prozkoumaných nástrojů nevyhovuje všem požadavkům uvedeným v bodě 2.1. Proto se jako nelepší způsob nabízí vytvoření vlastního vizualizačního nástroje, uspůsobeného přesným požadavků. V rámci jeho vývoje se bude usilovat nejdříve o splnění nejdůležitejších požadavků a poté bude vývoj pokračovat se zaměřením na ty ostatní. Velká výhoda tohoto způsobu je také interakce s cílovými uživateli během vývoje, jehož výsledkem by měla být velka informační hodnota výsledného vizualizačního nástroje a také jeho rozšiřitelnost, flexibilita a schopnost odrážet nové trendy v analýzách sítě. Praktická použitelnost by měla být dokázána dosavadní i následující aktivní komunikací s cílovými uživateli, tedy se členy řešitelského týmu projektu CAMNEP[26].

Kapitola 3

Průzkum platforem pro vizualizaci

3.1 Požadavky na knihovny

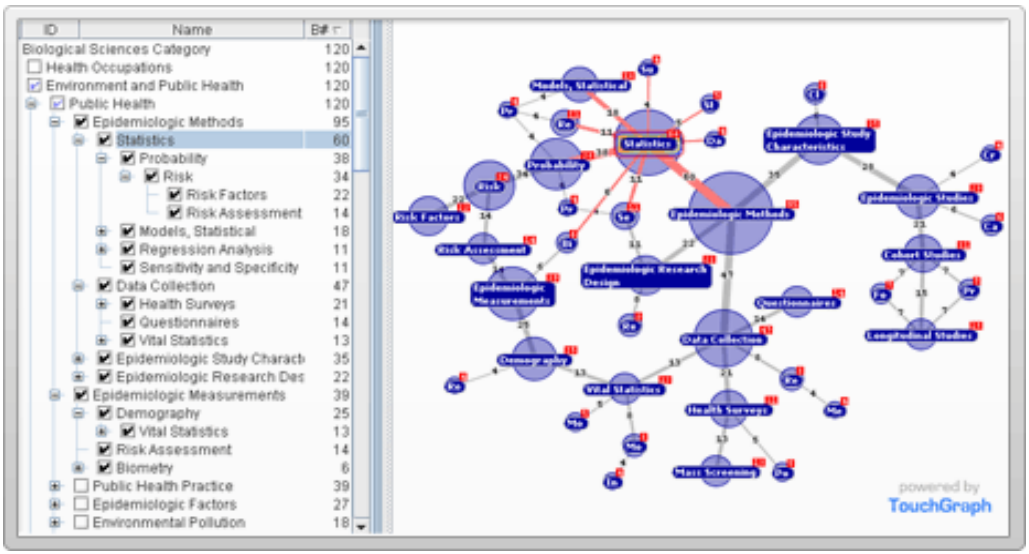
Výsledkem praktické části bude aplikace, která jako svůj vstup zpracuje offline NetFlow data a vytvoří vhodnou reprezentaci těchto dat pomocí vizualizace, která se bude řídit principy popsanými v úvodu. Proto je nutné, aby hledaná knihovna implementovala přehledný způsob vizuální reprezentace dat a byla uspůsobená na vizualizaci offline NetFlow dat. Tato knihovna by měla respektovat standard modelu InfoVis a model MVC (Model View Controller)[25]. Tím se zvýší transparentnost kódu a fungování aplikace a také možnosti dalšího rozšíření.
Do této aplikace bude nutné zakomponovat výstupní NetFlow data konvertovaná programem nfdump. To bude provedeno nejlépe generováním grafu síťového provozu ze souboru ve formátu csv.
Výsledná aplikace musí maximálně využívat existující programové zázemí dané knihovny. Knihovna proto musí implementovat standardní způsoby vizualizace a umožnit jejich rozšíření a implementaci dalších funkcí. Pro další použití této aplikace bude potřeba použít nějaký standardní formát souboru jako univerzální rozhraní mezi touto aplikací a jakýmkoliv vstupem NetFlow dat. Knihovna by tedy měla implementovat generování grafu z takového souboru, jehož formát bude postaven nejlépe nad xml.
Knihovna by již měla implementovat metody pro přispůsobení vizualizace parametrům zobrazovaných dat, například velikost nebo barva uzlů nebo hran podle množství přenesených dat nebo čísla portu.
Důležitá je možnost nějakého způsobu filtrovnáni vizualizace. Základní je filtrace podle přenesených dat, portů a délky cesty od vybraného uzlu. Jeho rozšířením je filtrování podle všech parametrů flow. Bylo by také vhodné mít možnost provádět toto filtrování pomocí regulárních výrazů a také podle ostatních dat, například whois informací. Toho by mělo být dosáhnuto nejlépe použitím nějakého fulltextového

vyhledávání, přičemž vstupní data by se v tomto případě indexovala v samostatném vlákně po jejich načtení. Jakékoliv filtrování dat musí probíhat nezávisle na vstupních datech. Důležitý požadavek je tedy to, aby nebylo nutné při filtrování tyto data znovu konvertovat podle nově zadaných parametrů, ať už spouštěním programu nfdump nebo vnitřní implementovanou funkcí.
Při filtraci, změnách zobrazení a uživatelské interakci je požadována dostatečně nízká prodleva. Proto by dynamické změny zobrazení, zpracování uživatelské interakce a filtrování měl zajišťovat operátor běžící v samostatném vlákně.
Pro splnění požadavku rozšiřitelnosti je nutné, aby hledaná knihovna byla vydávána jako opensource a nejlépe v Javě pro snadnou přenositelnost mezi operačními systémy. Projekt takové knihovny musí mít kvalitní dokumentaci a aktivní komunitu, aby bylo možné udělat si přehled o možnostech, které tato knihovna nabízí a mít podporu při řešení nestandardních úkolů.

3.2 Dostupné knihovny

3.2.1 TouchGraph

TouchGraph[17] poskytuje zobrazení síťě jako interaktivni graf. To dovoluje množství transformací zobrazení a uživateli to umožňuje orientovat se v rozhlehlém grafu a také prozkoumávat vybrané cesty grafem. Vývojáři této platformy mají snahu o co nejvíce staticky vypadající graf, s tím, aby změny rozvržení a zobrazení byly předvídatelné, aby je uživatel mohl zopakovat a aby se mohl vrátit zpět na stav před těmito změnami. TouchGraph také poskytuje funkce pro vytváření reportů a jejich export do formátů jako je například excelovský list nebo obrázek.



TouchGraph
http://www.touchgraph.com/screens/products-technology-04.gif

3.2.2 GraphVis

GraphVis[18] je opensource nástroj, ale také knihovna pro vizualizaci grafů. Poskytuje množství typů rozvržení zobrazení, navíc kromě interaktvivního grafického prostředí také web rozhraní.
Jako vstup je použit textový soubor popisující zobrazovaný graf. Výstup této knihovny může být buď ve formátu zmiňované interaktivní vizualizace nebo ve formátu obrázku, Postscriptu , PDF atd.

3.2.3 JUNG

JUNG[19] architektura je navržena, aby podporovala množství reprezentací objektů a jejich relací, jako například neoriantované a orientované grafy, multimodální grafy, grafy s násobnými hranami, atd. Poskytuje mechanismus pro popisování grafů, objektů a relací pomocí metadat. To usnadňuje vytváření analytických nástrojů komplexní datové struktury, jejijchž relace a metadata mohou být zkoumány.

3.2.4 Prefuse

Výbornou platformou je projekt prefuse. Ten byl také vybrán pro výsledný vizualizační systém je podrobě popsán v kapitole 4.

3.3 Srovnání

Z existujících platforem byly vybrány tři nejznámější, ty které reprezentují vzorek existujících platforem. Knihovna JUNG je základní příklad knihovny pro vizualizaci. Je ale také základní v množství nabízených funkcí. Jeho interaktivita je velmi slabá, nepodporuje žádnou formu dynamického zobrazení. GraphVis poskytuje zajímavé a přehledné zobrazení, dobré zobrazení kontextu. Nepodporuje ale NetFlow export jako vstupní data.
Třetí platforma se jmenuje TouchGraph. Z těchto třech je určitě nejkvalitnější a k danému účelu nejvhodnější. Bohužel je nyní vyvýjena jako komerční projekt, takže nesplňuje požadavek, aby cílená platforma byla použitelná zdarma. Protože není vyvýjena jako opensource , není možné dále rozšiřovat její funkcionalitu podle kokrétních požadavků. Protože to není jasné ze stránek projektu, není možné určit přesnou funkcionalitu. Jako podpora vývoje založeného na této platformě chybí také aktivní komunita .

3.4 Shrnutí

Žádná z uvedených knihoven s vyjímkou prefuse nesplňuje všechny požadavky stanovené v bodě 3.1. Chybí implementované standardní funkce pro práci s daty, ovládání a vizualizace a neposkytují snadnou možnost rozvoje a přispůsobení danému úkolu. Dobře použitelná a velice kvalitní knihovna je aktuální verze TouchGraph, ta je však nyní vyvíjena jako komerční projekt. Není také jasná její plná funkcionalita a možnosti rozšíření jsou v podstatě nulové. Firma, která tento projekt vyvýjí nabízí pouze řešení na míru a to je pro tento účel naprosto nepřijatelné.
Z těchto důvodů byla vybrána knihovna vyvýjená v rámci projektu prefuse[21]. Tato knihovna splňuje, až na nedůležité detailny, všechny požadavky, stanovené v bodě 3.1 a je popsána v další kapitole.

Kapitola 4

Volba platformy - prefuse API

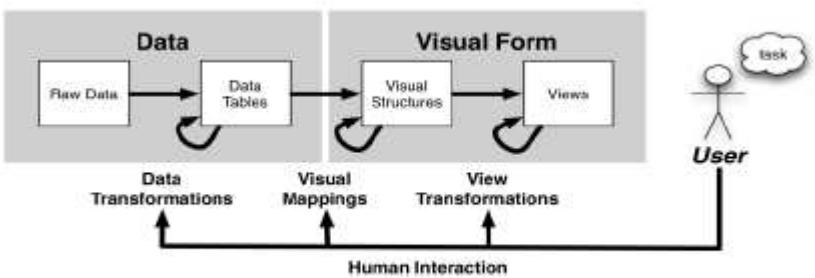
4.1 Základní vlastnosti

Knihovna prefuse je vyvíjena a distribuována jako opensource projekt. Je napsaná v javě, vizualizace pomoci ní vytvořené lze tedy spouštět na jakémkoliv operačním systému, pro který existuje interpretr javy. Součástí projektu je výborna dokumentace a aktivní komunita například ve formě fóra[20]. Na stránkách projektu *prefuse*[21] jsou ukázky vytvořených vizualizací, které tuto knihovnu používají a také dema od tvůrců prefuse, takže je možné udělat si rychle jasnou představu o schopnostech této knihovny. Prefuse poskytuje množství rozhraní pro uživatelskou implementaci a také třídy, které je možné použít nebo rozšířit a tím vytvořit vizualizaci na míru podle potřeb projektu. Knihovna je navržena podle standardu referenčního modelu InfoVis, dodržuje tedy model MVC a proto je výborný přehled nad funkci knihoven a také je možné jednoduše sledovat data a struktury na cestě od abstraktních dat k vytvořeni a zobrazeni vizualizace.
Knihovna obsahuje spoustu implementovaných tříd pro zápis a čtení formátovaných souboru, čtení z relační databáze, komponenty pro tvar, barvu, velikost, způsob rozmístění a vzájemnou vzdálenost prvků vizualizace, reakce na uživatelskou interakci nebo také třídy pro animaci změn rozvržení zobrazení.
Je zde implementován engine pro simulaci fyzikálních sil, věrně napodobující odpudivé a přitažlivé sily mezi jednotlivými prvky vizualizace. Zajímavé jsou také jednoduchý dotazovací jazyk pro interaktivní filtrování výstupu vizualizace a třídy pro indexování dat asociovaných s uzly a hranami a vyhledávání v nich, používající regulární výrazy a fulltextový vyhledávací engine Apache Lucene[23]. Poskytují pohodlný způsob výběru relevantních dat. Všechny změny se filtrují z vizualizace nad konstantními daty, není tedy třeba opakovat proces načítání a konvertování dat.

Poslední a důležitá přednost je plánovač akcí, který řídí všechny změny vizualizace, počínající jejím spuštěním a pokračující interaktivními změnami popsanými výše. Jeho neopomenutelná výhoda je to, ze běží v dedikovaném vlákně a ovládá všechny vizualizace. Nemůže tak dojit k náhodné interakci mezi souběžnými akcemi a vetším počtem vizualizaci.

4.2 InfoVis referenční model

Jedna z výhod knihovny *prefuse* je fakt, že je navržena v souladu s referenčním modelem InfoVis. Tento model představuje tři úrovně v konceptu MVC (Model View Controller) a usnadňuje tak přehlednost a oddělení zobrazovací casti od logické. Je proto jednoduché napsat program tak, aby byl flexibilní vůči změnám typu zdroje dat nebo napsat více vizualizací, používající stejný logicky základ jako zdroj informací.
Jednotlivé úrovně jsou zobrazeny na následujícím obrázku :



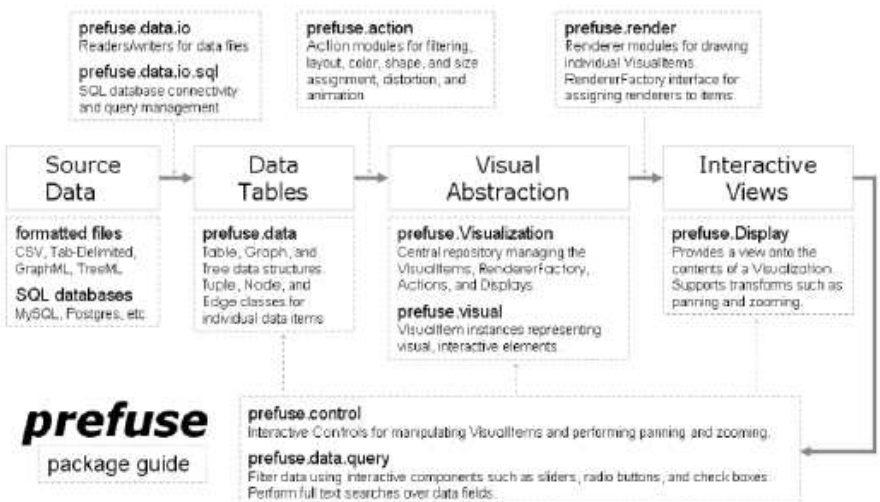
http://www.infovis-wiki.net/index.php/Image:Prefuse-tutorial-20061127-handout.pdf

Nejdříve jsou knihovně poskytnuta nezpracovaná **data** ve formátu charakteristickém pro typ vstupních

dat. Tyto data jsou pote transformovány na formát zdůrazňující strukturu dat a určující jejich uspořádání. Výsledkem teto transformace jsou datové **tabulky** - organizovaná data s vazbami a metadaty. Další krok je vizuální mapování, které přidá strukturovaným datům vizuální informaci a generuje tak vizuální struktury. Nakonec přichází na řadu vizuální transformace umožňující zobrazení vizuálních informací pod úhlem pohledu, zajištujícím čitelnost pro uživatele. Jsou vytvořeny tzv. **Pohledy (Views)**, umožňující uložení takových informací.
Tento referenční model počítá z uživatelským ovládáním vizualizace, poskytujícím funkce na přeskupování uzlu, otáčení, zvětšování zobrazení, filtrování, atd.

4.3 Implementace InfoVis modelu v knihovně *prefuse*

4.3.1 Přehled



http://www.infovis-wiki.net/index.php/Image:Prefuse-tutorial-20061127-handout.pdf

4.3.2 Zdrojová data

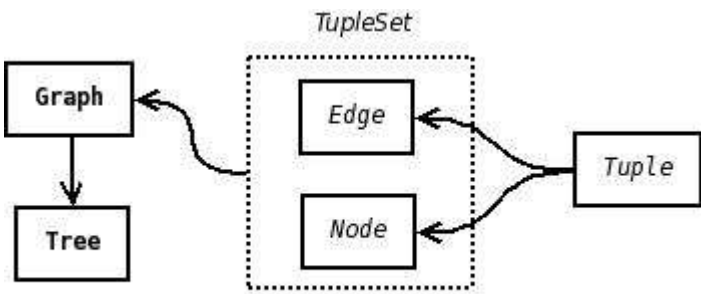
Na začátku jsou specifická zdrojová data. Mohou to být formátované soubory, databáze nebo jakákoliv jiná predstavitelná forma dat. Prefuse API poskytuje základní rozhraní a implementace struktur dat pro nespecifikovaná data, graf a strom.
Základní datový typ je Entita (rozhraní **Entity**), která podporuje jakýkoliv pojmenovaný atribut. Z ní vychází strukturované typy jako **Node**, **TreeNode** a **Edge**.
Jeden z nejpoužívanějších zdrojů dat je například **Graph ML**[22], který představuje formát souboru pro popisování grafu. Je postavený na XML a jeho struktura je navržena pro reprezentaci neorientovaných, orientovaných a smíšených grafu, také hypergrafů, externích dat, ale i jednoduchých syntaktických analyzátorů. Tento formát je použitý jako zdroj dat také v aplikaci, popisované v části 4.5.

4.3.3 Načtení ze zdroje

Při načítání dat ze zdroje je nutné převést zdrojová data v uživatelsky definovaném formátu na strukturu, která je vhodná pro další fáze zpracování dat. V tomto bodě závisí na programátorovi, který implementuje konkretní použití této knihovny, jestli zvolí cestu vytvoření vlastních funkcí a napíše je na míru vstupním datum nebo použije již implementované funkce, které poskytují balíčky prefuse.data.io a prefuse.data.io.sql.
Balíček **prefuse.data.io** poskytuje rozhraní a abstraktní a implementované třídy pro čtení a zápis formátovaných souboru. Implementací rozhraní GraphReader, GraphWriter, TableReader, TableWriter nebo jejich abstraktních potomků je možné vytvořit vlastní třídu pro čtení nebo zápis. Druhý způsob je použití jedné z již implementovaných tříd pro čtení a zápis csv, graphML nebo treeML formátu.
Naproti tomu **prefuse.data.io.sql** poskytuje způsob, jak jednoduše zařadit do vytvářené vizualizace relační databázi jako zdroj vstupních dat. Proto je lehko přístupná cesta použití knihovny pro vizualizaci nějakých centrálních dat nebo statistické zobrazení dat uložených v databázi.

4.3.4 Datové tabulky

Datové tabulky jsou pro vnitřní potřebu určená úložiště dat. To usnadňuje manipulaci s daty. Každý záznam v takové tabulce představuje n-tici pojmenovaných atributů specifických datových typů. Tyto datové struktury ještě neobsahují žádné vizuální informace a jsou proto jen strukturovanou reprezentací vstupních dat. Pro tuto část zpracování je určen balíček **prefuse.data**..
Obsahuje rozhraní *Node*, reprezentující informace o uzlu a *Edge* pro hranu. Obě rozhraní jsou rozšířením třídy Tuple, která reprezentuje n-tici pojmenovaných atributu. Rozhraní *TupleSet* obsahuje seznam n-tic s významem seznamu uzlů a hran. Toto rozhraní implementuje třída **Graph** (a také její speciální podtřída **Tree**), která se používá jako výchozí struktura pro další zpracování. V teto fázi je proto vytvořena třída pro uložení a manipulaci s daty grafu nebo stromu.

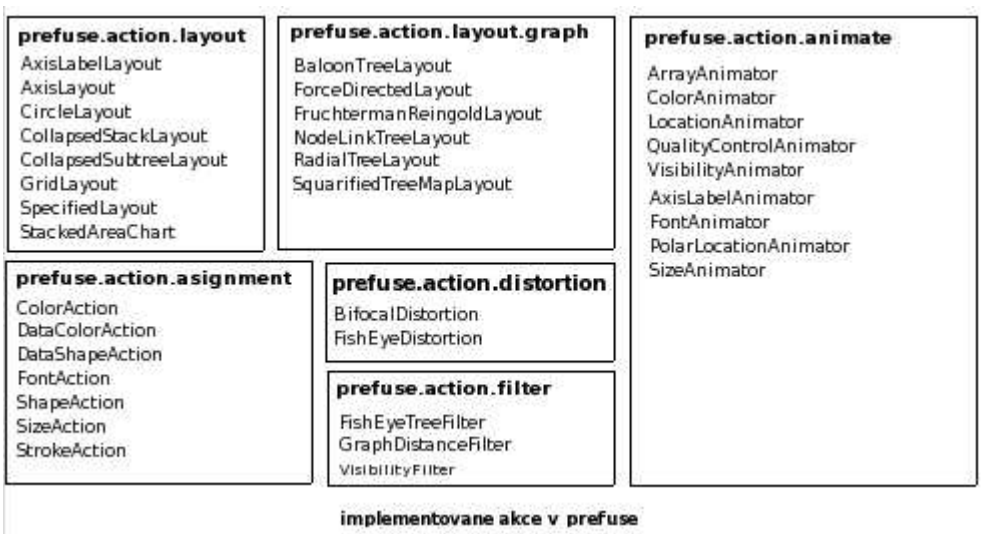


Important interfaces and classes from package prefuse.data

4.3.5 Filtrování

Máme-li již data reprezentovaná nějakou strukturou, která uchovává zdroj dat zvolený pro zobrazení, potřebujeme je nějakým způsobem přetransformovat na strukturu, která bude výchozí zdroj pro zobrazení vizualizace a přidat základní vizuální informace jako pozici, barvu a velikost.
V teto fázi nastupuje filtrování, chápané jako mapování abstraktních dat na formu vyhovující vizualizaci. Nejdříve se vybere region zájmu - například oblast grafu nebo určitá úroveň stromu. Z něj jsou poté vygenerovány vizuální jednotky (Visual Items), jejichž výpočet je založený na charakteru dat a uživatelově výběru typu zobrazení.
V této fázi nastupuje balíček **prefuse.action**, který obsahuje moduly pro zpracování dat a priřazení vizuálních informací. Pro pochopení principu funkce tohoto balíčku - funkce akcí v API *prefuse* je nutné nejdříve popsat nejdůležitejší třídu **ActionList**.
Akční seznam je třída, která slouží jako spustitelná kolekce akcí (třída **Action**). Sám je také akcí a dovoluje tedy, aby obsahoval další akční seznamy. Pokud je spuštěn, zařadí tedy do plánovače všechny akce a spustí všechny akční seznamy, které obsahuje. Může být spuštěn buď jako reakce na systémovou událost nebo uživatelskou interakci. Seznamy se do vizualizace přidávají s popisujícím identifikačním retězcem. Vizualizace také umožňuje nastavit závislosti akčních listů. To je například použito pro zajištění spuštění akčního seznamu pro překreslení vždy po změně proporcí zobrazení.
Spouštění seznamu akci zajišťuje implementovaný plánovač aktivit, který přijímá objekty **Activity** (nadtřída třídy ActionList) s parametry času startu, trvání aktivity, četnost opakování za jednotku času a podle těchto kriterií je spouští. Tento plánovač běží ve vyhrazeném vlákně a spouští akce nad všemi vizualizacemi. Tím zajišťuje atomicitu jednotlivých akcí a zabraňuje problemům se soubežnými akcemi, které by se mohly navzájem ovlivňovat nečekaným způsobem.
První akce, se kterými se programátor setká, jsou tzv. layouty, neboli třídy, které, zařazené do akčních listů a vizualizace, určují způsob rozvržení zobrazení a jeho vlastnosti. Například v aplikaci, popisované v bodě 4.5, je použit layout prefuse.layout.graph.ForceDirectedLayout, který implementuje fyzikální

simulaci působení sil uzlů navzájem se ovlivňujících. Tak je jednoduše možné vykreslit zobrazení, které se chová, jakoby se jednotlivé uzly přitahovaly pomocí gravitačních sil. Dva balíčky, které jsou v distribuci pro tyto účely implementované jsou **prefuse.action.layout** a **prefuse.action.layout.graph**. Vzhled ovlivňují akce - třídy z balíčku prefuse.action.assignment. Takto je možné staticky nebo dynamicky v závislosti na datech určit tvar, barvu, velikost a popisek. Dale jen ve stručnosti zmíním akce z balíčku **prefuse.action.filter** pro filtrování, **prefuse.action.distortion** pro pokřivení části zobrazení a **prefuse.action.animation** pro animaci změn zobrazení.



4.3.6 Vizuální abstrakce

Výsledkem filtrování jsou tedy visuální abstrakce. Balíček **prefuse** obsahuje třídu **Visualization**, která tuto abstrakci reprezentuje. Tato třída je centrální datovou strukturu, reprezentující interaktivní vizualizaci. Ta je svázána se všemi akčními listy a s daty grafu. Tato třída již obsahuje všechny informace pro určení vzhledu a chování vizualizace. Každé n-tici, tedy Tuple, Node nebo Edge opdovídá v této třídě instance třídy **VisualItem**. Ta poskytuje přístup jak k jejím vizuálním parametrům, tak k výchozím datům.

4.3.7 Zobrazení vizualizace

Aktuální vzhled instance **VisualItem** je určen třídou implementující rozhraní **Renderer**. Ta je zodpovědná za vykreslování všech položek vizualizace a počítání jejich mezí (kolik místa je potřeba pro zobrazení této položky). *Prefuse* poskytuje třídy pro vykreslení různých druhů tvarů, popisků a také pro zobrazení obrazků. Výběr Rendereru je určen třídou **RendererFactory**, která je přiřazena vizualizaci. Při každém zobrazení se jí vizualizace dotazuje, jakou vykreslovací třídu použít. Pro zobrazení se používá třída **Display**. Ta zde figuruje jako pohled na určitý kontext vizualizace. Tato třída je komponena, rozšiřující **javax.swing.JComponent**, která je použitelná v grafických Java aplikacích a appletech. Zobrazuje vizualizace v konkrétním pohledu a poskytuje zpětnou vazbu pro uživatelskou interakci. To umožňuje provádět jakékoliv uživatelsky implementované akce. Je také možné použít již implementované reakce na uživatelské akce, například zvětšování, zmenšování, otáčení pohledu a změna aktuálního kontextu. Jedna vizualizace může být také asociována s více displeji za účelem vytvoření poddispleje s přehledem o celé vizualizaci nebo zobrazení dvou rovnocenných displejů, které každý zobrazují jiný kontext vizualizace.

4.3.8 Interaktivní zobrazení - Control akce a query language

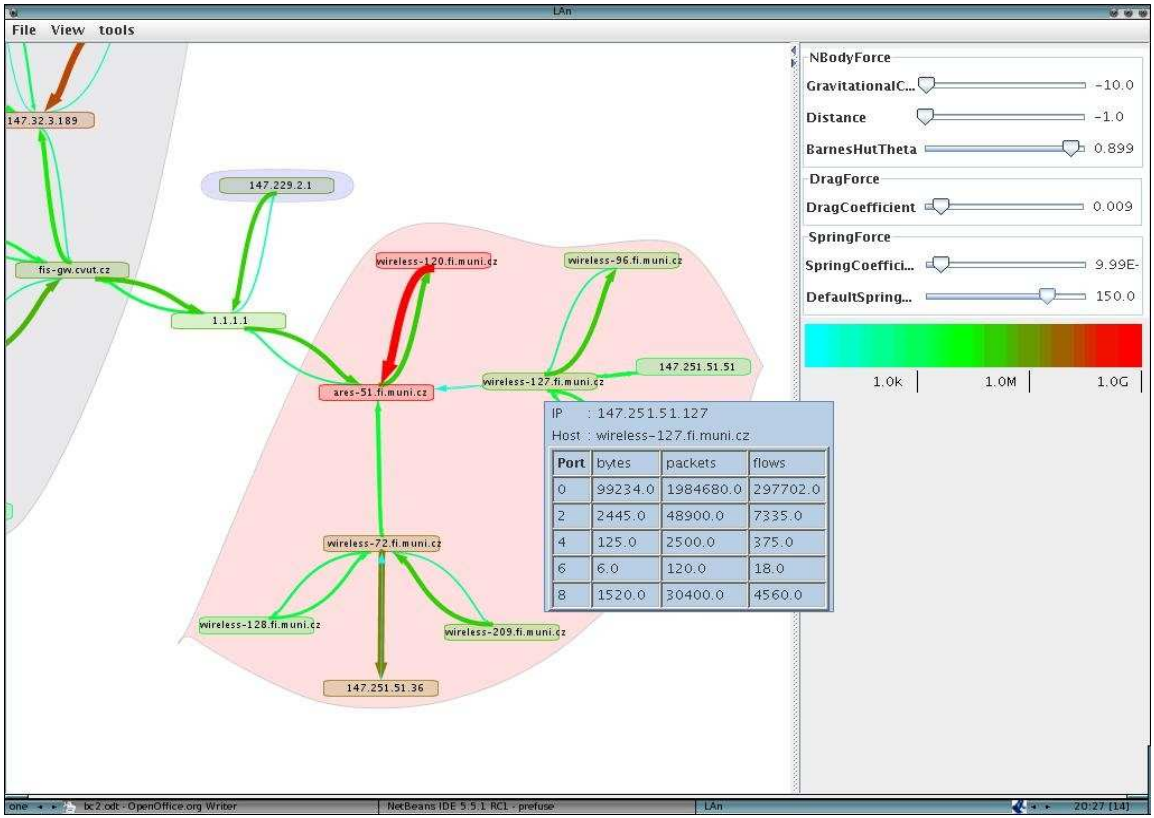
Pokud se programátor dostal po jednotlivých bodech až do této fáze, může být zmatený a obávat se složitosti implementace ovládání do dosavadní struktury programu. Při návrhu pomocí prefuse je program velmi modulární a každá jeho část je v rámci mezí univerzální a nezávislá na ostatních použitých modulech. Ve stejném duchu se nese také ovládání vizualizace, ve kterém je každá funkce implementována jako reakce na uživatelskou interakci.
Základní a v této fázi nejdůležitejší je znát rozhraní **Control** v balíčku **prefuse.control**. Toto rozhraní je zde jako posluchač událostí vyvolaných uživatelskými akcemi. Obsahuje základní funkce, jejichž implementací je možné jednoduše tyto akce zpracovávat. Jako názorný příklad je funkce, deklarovaná takto : void itemClicked(VisualItem, MouseEvent). Její implementace tedy spočívá v programování akce, která se spustí při kliknutí myší na položku vizualizace a přes parametry funkce má k dispozici instanci objektu, který položku reprezentuje a objekt události akce, způsobené myší. Její základní implementaci nabízí třída **ControlAdapter**, která všechny akce implementuje jako prázdné akce. Balíček prefuse.control také poskytuje několik tříd s již implementovanými akcemi pro základní interakci. To jsou například HoverActionControl pro spuštění parametrem určené akce, pokud je kurzor myši nad nějakou položkou, ToolTipControl pro zobrazení krátké tooltip informace, RotationControl pro otáčení celé vizualizace, atd.
Instance kontrolních tříd jsou poté zaregistrovány ve vizualizaci jako posluchači.
Ve fázi návrhu uživatelského rozhraní je také možné zakomponovat filtrování vizuálních dat. Výsledkem takového filtrování může být filtrace zobrazení vizuálních položek, zdůraznění výběru parametricky určené množiny položek nebo také statistická data, která se vybraných položek týkají. Filtrování je nejlépe uskutečňováno spojením dotazovacího jazyku v prefuse implementovaného a použitím nějaké implementace abstraktní třídy SearchTupleSet z balíčku prefuse.data.search. Tento balíček nabízí implementace pro vyhledávání podle předpony - PrefixSearchTupleSet, regulárního výrazu – RegexSearchTupleSet a fulltextové vyhledávání - LuceneSearcher. Vyhledávání podle předpony používá

vnitřní indexovanou datovou strukturu. Regulární vyhledávání prochazí všechny n-tice a porovnává je standardními funkcemi pro regulární výrazy. Fulltextové vyhledávání je postaveno na technologii Apache Lucene, probíhá nad vlastním specifickým indexem s tím rozdílem, že v tomto případě je index uložen v paměti.

Kapitola 5

Výsledná aplikace

Vytvořená aplikace usiluje o přehlednou vizualizaci při použití dynamického rozvržení, které podporuje agregaci zobrazování komunikace podle IP adres, podsítí a protokolů. Přehlednost zobrazení je rovněž podpořena také vhodnou kombinací barev, tvarů a velikostí. Počet flow, množství přenesených bytů a paketů je znázorněno právě odlišnou velikostí a barevnou reprezentací odvozující barvu z jednotného schématu. Přehlednost vizualizace jednotlivých podsítí je zajišťována podbarvením množin uzlů, které danou síť reprezentují. Aplikace implementuje několik funkcí jako je zobrazení WHOIS informací pro každý jednotlivý uzel a to pro libovolné množství WHOIS serverů nadefinovaných v konfigurační třídě aplikace. Automatické doplňování DNS jmen strojů je řešeno samostatným vláknem, které si do fronty ukládá požadavky na DNS dotazy a postupně je uskutečňuje.
Jako vstup je přijímán GraphML soubor, csv soubor se syntaxí, popsanou v dokumentaci programu nebo nfcapd logovací soubor, na který aplikace reaguje spuštěním nástroje nfdump a vytvořením dočasného csv souboru.
Implementováno bylo také filtrování zobrazovaných položek podle množství přenesených dat, čísla portu a podle parametrů jednotlivých flow nebo jmen uzlů. Model level of detail je zde implementován třemi kroky procesního modelu. Od přehledu nad celou vizualizací, kdy jsou zobrazovány jen ty nejdůležitejší aspekty, přes přiblíženou vybranou podčást vizualizace, až po detailní informace o jednotlivých uzlech. IP adresy, které komunikovaly s vybranou IP adresou se vizualizují také formou tabulky (ne pouze dynamickou myšlenkovou mapou). Uživatel si také může definovat významné IP adresy (brány, DNS, poštovní servery, ...), které budou vizualizovány odlišným způsobem tak, aby byly na první pohled rozpoznatelné..
Detailní popis výsledné aplikace je přiložen na CD spolu se zdrojovými kódy.


LAn
File View tools
147.32.3.189
147.229.2.1
fis-gw.cvut.cz
1.1.1.1
wireless-120.fi.muni.cz
wireless-96.fi.muni.cz
ares-51.fi.muni.cz
wireless-127.fi.muni.cz
147.251.51.51
wireless-72.fi.muni.cz
wireless-128.fi.muni.cz
wireless-209.fi.muni.cz
147.251.51.36
IP : 147.251.51.127
Host : wireless-127.fi.muni.cz
Port bytes packets flows
0 99234.0 1984680.0 297702.0
2 2445.0 48900.0 7335.0
4 125.0 2500.0 375.0
6 6.0 120.0 18.0
8 1520.0 30400.0 4560.0
NBodyForce
GravitationalC... -10.0
Distance -1.0
BarnesHutTheta 0.899
DragForce
DragCoefficient 0.009
SpringForce
SpringCoeffici... 9.99E-
DefaultSpring... 150.0
1.0k
1.0M
1.0G
bc2.odt - OpenOffice.org Writer
NetBeans IDE 5.5.1 RC1 - prefuse
LAn
20:27 (14)

Kapitola 6

Závěr

V úvodu této práce byly představeny základní důvody sledování sítě a vytváření vizuálních reprezentací výstupů z těchto sledování. Dále byly stručně popsány požadavky na tyto vizualizace a naznačeny základní postupy, kterými lze tyto požadavky naplnit.

Druhá část popisuje jak obecné, tak pro tento projekt zcela konkrétní požadavky kladené na kvalitní vizualizační software. Po následném srovnání vyšlo najevo, že mezi těmito nástroji není takový, který by všechny splňoval. Jako nejlepší rešení se ukázalo vytvoření vlastního vizualizačního nástroje postaveného na jedné z existujících knihoven. Po provedení průzkumu byla vybrána vizualizační platforma projektu prefuse. Čtvrtá kapitola detailně popisuje důvody výběru této platformy a principy jejího fungování.

Výslednou aplikaci, vzniklou v rámci této bakalářské práce stručně prezentuje kapitola pátá. Podrobný popis této aplikace je spolu se zdrojovými kódy přiložen na CD.
Cílem práce bylo vytvořit vizualizační systém, který splňuje základní požadavky kladené na kvalitní vizualizační software. Systém je navržen tak, aby bylo možné zbylé funkce jednoduše doplnit formou pluginů. Tato práce vedla k vytvoření nástroje, který vizualizuje provoz na síti a reflektuje potřeby těchto uživatelů. Výsledky práce byly verifikovány síťovými experty v projektu CAMNEP.

Literatura

[1] Cisco System, Inc <http://www.cisco.com>

[2] Wikipedia – Class of Service <http://en.wikipedia.org/wiki/Class_of_Service≥

[3] Cisco Internetwork Operating System <http://en.wikipedia.org/wiki/Cisco_IOS>

[4] Cisco IOS NetFlow
<http://www.cisco.com/en/US/products/ps6601/products_ios_protocol_group_home.html>
<http://en.wikipedia.org/wiki/Netflow>

[5] Level of detail
<http://www.cs.virginia.edu/~gfx/Courses/2004/RealTime/lecture06.LOD1.ppt>
<http://en.wikipedia.org/wiki/Level_of_detail_(programming)>

[6] Ring network definition
<http://searchsmb.techtarget.com/sDefinition/0,,sid44_gci870770,00.html>

[7] TCPDUMP <http://www.tcpdump.org>

[8] EtherApe <http://etherape.sourceforge.net>

[9] Fiber Distributed Data Interface
<http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/fddi.htm>

[10] Point-to-Point Protocol <http://www.cisco.com/univercd/cc/td/doc/cisintwk/ito_doc/ppp.htm>

[11] Serial Line Internet Protocol
<http://en.wikipedia.org/wiki/Serial_Line_Internet_Protocol>

[12] D2K - Data To Knowledge <http://alg.ncsa.uiuc.edu/do/tools/d2k>

[13] NVisionIP: NetFlow Visualizations of System State for Security Situational Awareness
<http://www.ncassr.org/projects/sift/papers/vizsec04-nvisionip.pdf>

[14] Security Vizualization <http://www.cs.ucdavis.edu/~ma/SecVis>

[15] CAIDA <http://www.caida.org/tools/measurement/skitter/visualizations.xml>

[16] The LibSea Graph File Format and Java Graph Library
<http://www.caida.org/tools/visualization/libsea>

[17] TouchGraph <http://www.touchgraph.com>

[18] Graphviz - Graph Visualization Software <http://www.graphviz.org>

[19] JUNG - Java Universal Network/Graph Framework <http://jung.sourceforge.net>

[20] The prefuse visualization toolkit forum
<http://sourceforge.net/forum/forum.php?forum_id=343013>

[21] The prefuse visualization toolkit <http://prefuse.org>

[22] The GraphML File Format <http://graphml.graphdrawing.org>

[23] Apache Lucene <http://lucene.apache.org>

[24] NFDUMP <http://nfdump.sourceforge.net>

[25] Model-View-Controller Song <http://csl.ensm-douai.fr/noury/20#mp3>

[26] CAMNEP (Cooperative Adaptive Mechanism for Network Protection)
<http://kirlab.fi.muni.cz/cs:apobl>

Příloha

Na přiloženém CD je výsledná aplikace s podrobným popisem, ukázkami použití a doplňující materiály o projektu prefuse.