UNIVERZITA PARDUBICE

FAKULTA ELEKTROTECHNIKY A INFORMATIKY

BAKALÁŘSKÁ PRÁCE

2009 Jan Horký

Univerzita Pardubice

Fakulta elektrotechniky a informatiky

Automatické generování diagramu sítě v aplikaci Dia

Jan Horký

Bakalářská práce
2009

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Katedra informačních technologií
Akademický rok: **2008/2009**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Jan HORKÝ**
Studijní program:	**B2646 Informační technologie**
Studijní obor:	**Informační technologie**
Název tématu:	**Automatické generování diagramu sítě v aplikaci Dia**

Zásady pro vypracování:

* V teoretické části práce bude věnován prostor pro popis možností monitorování sítí a protokolu SNMP. * V praktické části pak bude vytvořen open-source nástroj pro generování a aktualizace diagramu počítačových sítí, který půjde dále zpracovávat v programu Dia.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. Eric S. Raymond: Umění programování v Unixu, CP Books , 2004 2. James M. Kretchmar, Libor Dostálek: Administrace a diagnostika sítí pomocí OpenSource utilit a nástrojů, CP Books, 2004, ISBN: 80-251-0345-5 3. http://live.gnome.org/Dia/Documentation

Vedoucí bakalářské práce: **Ing. Tomáš Fidler**
Katedra softwarových technologií

Datum zadání bakalářské práce: **15. ledna 2009**

Termín odevzdání bakalářské práce: **15. května 2009**

L.S.

doc. Ing. Simeon Karamazov, Dr.
děkan

Ing. Lukáš Čegan
vedoucí katedry

V Pardubicích dne 31. března 2009

Prohlášení

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Hradci Králové dne 19.8.2009

Jan Horký

Poděkování

Na tomto místě bych chtěl poděkovat svému vedoucímu práce, panu Ing. Tomáši Fidlerovi, za cenné rady, které vedly ke zdárnému dokončení této práce.

Anotace

Tato práce se zabývá možnostmi monitorování datových sítí (konkrétně počítačových) a jejich analýzou se zaměřením na protokol SNMP. V teoretické části jsou rozebrány protokoly a techniky použitelné pro sběr informací o topologii sítě a jednotlivých prvků do ní připojených, jenž umožní automatizovanou tvorbu jejich diagramů. Dále byl vytvořen modul pro tvorbu těchto diagramů v aplikaci DIA, využívající některé nastíněné techniky.

Klíčová slova

Diagnostika, analýza a správa sítě, SNMP, diagram sítě, DIA.

Title

Automatic generation of network diagram in application DIA.

Annotation

The thesis deals with possibillities of data networks monitoring (concretely computer's) and theirs analysis focused on SNMP protocol. In the teoretical part there are discussed protocols and techniques useable for collecting information about net's topology and theirs items. This subsequently allows automatical creation of graphical diagrams of this topology. Then there was created plug-in for automatical creation of this diagrams in application DIA by using some of focused techniques.

Keywords

Network diagnostic, analyse and administration, SNMP, Network topology diagram, DIA.

Obsah

Seznam obrázků

Úvod

Datové sítě jsou založeny na propojení prvků, umožňujícím jejich vzájemnou komunikaci. Tyto prvky jsou hardwarová zařízení s obslužným programem. Stejně jako jiná zařízení jsou poruchová, a to jak na straně hardwaru, tak i softwaru. V případě, že tyto prvky jsou mezilehlými body, jejich poruchou je znemožněna nebo omezena komunikace ostatních prvků, které propojuje. Proto vyvstává potřeba jejich chod náležitě sledovat, abychom mohli v co nejkratším čase reagovat na jejich chyby a výpadky, nebo ještě lépe, díky včasným informacím, jim předcházet. S počtem připojených zařízení se zvyšuje možnost výskytu chyby a zároveň i počet tím postižených ostatních bodů. V té souvislosti se zvyšuje i náročnost na sledování stavu těchto zařízení v reálném čase. Z toho důvodu byly vyvinuty protokoly a nástroje, které nám umožňují vzdálené sledování jejich stavu, a to i více prvků najednou. Některé z nich proto budou v prvních pěti kapitolách popsány .

Pro zjištění, které prvky jsou rizikové a je proto potřeba monitorovat jejich chod, potřebujeme mít dostatečnou znalost topologie dané sítě a dostatečné informace o jednotlivých zařízeních. V případě důsledné dokumentace to nemusí být velký problém. Ovšem opět s rostoucím počtem připojených zařízeních a rychlostí jejich obměny, v dané síti, je její tvorba čím dál obtížnější. V takovém případě nám přijdou vhod nástroje, které nám co nejvíce informací zjistí automaticky, a to za kratší časový úsek než-li bychom to dokázali sami. V další části tedy budou probrány metody jak takové automatické získávání provádět a graficky interpretovat.

V poslední části je pak popsána implementace vlastního zásuvného modulu pro aplikaci DIA aplikace, jenž využívá některých z těchto metod.

Vzhledem k tomu, že většina dnešních sítí, včetně té největší – Internetu, je postavena na protokolech a zásadách z rodiny TCP/IP, jsou v této práci rozebírány metody a protokoly pro ně vhodné.

1. Servisní protokoly TCP/IP

Servisní protokoly pomáhají při přenosu dat a signalizují různé stavy při průchodu těchto dat sítí. Tyto protokoly jsou důležité i pro diagnostiku sítě, tedy pro nás zajímavé.

1.1. Protokol ICMP

ICMP (Internet Control Message Protocol) je servisním protokolem a součástí protokolu IP. Jak již bylo řečeno výše, slouží pro hlášení chyb a nestandardních situací na síti. Vzhledem k faktu, že tyto informace cestují i přes mezilehlé prvky a sítě, jsou tyto pakety dopravovány v datové části IP paketů. Protokol ICMP je povinnou součástí každé implementace IP protokolu, ovšem některé z nich implementují pouze jeho omezenou část.

Z důvodů bezpečnosti nebo upřednostňování datového provozu mohou být na některých zařízeních (směrovačích) ICMP zprávy zahazovány – tento jev nemusí znamenat jejich chybnou funkčnost.

1.2. Struktura ICMP paketu

ICMP paket obsahuje vlastní hlavičku a datovou část. Záhlaví je 8 bajtů dlouhé a dále se dělí na pevnou (4B) a proměnnou (4B) část. Pevná část obsahuje Typ ICMP zprávy, upřesňující kód a kontrolní součet. Typ a kód určují druh zprávy a formát proměnné části záhlaví a dat.

Dále budou uvedeny některé důležité, pro diagnostiku sítí nejvíce využívané, typy ICMP zpráv.

1.2.1. Echo (Ping)

Typ zprávy 8, kód 0, umožňující zjišťování dosažitelnosti jednotlivých bodů sítě. První bod vyšle zprávu - žádost o Echo. Cílový bod, v případe dostupnosti nebo pokud toto není v daném zařízení zakázáno, odpoví zprávou typu 0, kód 0 – odpověď na Echo. První bod po jejím příchodu spočítá dobu odezvy – dobu mezi odesláním žádosti a odpovědí.

Konkrétní implementací je program „Ping“ dostupný v nejpoužívanějších operačních systémech.

```
C:\>ping 192.168.11.1

Příkaz PING na 192.168.11.1 s délkou 32 bajtů:

Odpověď od 192.168.11.1: bajty=32 čas < 1ms TTL=64
Odpověď od 192.168.11.1: bajty=32 čas < 1ms TTL=64
Odpověď od 192.168.11.1: bajty=32 čas < 1ms TTL=64
Odpověď od 192.168.11.1: bajty=32 čas < 1ms TTL=64

Statistika ping pro 192.168.11.1:
Pakety: Odeslané = 4, Přijaté = 4, Ztracené = 0 (ztráta 0%),
Přibližná doba do přijetí odezvy v milisekundách:
    Minimum = 0ms, Maximum = 0ms, Průměr = 0ms
```

Obrázek 1 - Výstup programu Ping

1.2.2. Nedoručitelný paket

Typ zprávy 3 – jestliže nemůže být paket někde skrze síť doručen, je o tom odesílateli odeslána příslušná zpráva s odůvodněním uvedeným pomocí kódu zprávy.

1.2.3. Čas vypršel (Traceroute)

Typ zprávy 11 – signalizuje jeden ze dvou možných případů:

- **Kód 0** – hodnota TTL byla snížena na hodnotu 0, dříve než-li paket dosáhl svého cíle – daný směrovač odešle o tomto zprávu, s tím, že paket byl zlikvidován (zahozen).
- **Kód 1** – cílový bod není schopen sestavit z příchozích fragmentů původní paket.

První typ s kódem 0 využívá další program – v Unixu pod názvem „Traceroute“, v MS Windows® „Tracert“.

Základním principem je odesílaní IP paketů s proměnnou hodnotou TTL v jeho hlavičce. Nejprve se odešle paket s hodnotou TTL=1 – průchodem prvním směrovačem je hodnota snížena na 0 a odesílateli vrácena ICMP zpráva „Čas vypršel“ (Time exceeded), paket byl zahozen. Adresa daného směrovače je zjištěna jako zdrojová IP adresa v chybovém ICMP/IP paketu. Stejně jako u „Echo“ je spočítán čas návratu. Tímto způsobem jsou odesílány pakety s postupným zvyšováním hodnoty TTL o 1 až do chvíle kdy je přijata odpověď od cíle. V případě, že směrovač nezná další cestu k cíli, vrátí odesílateli chybovou ICMP zprávu „Nedoručitelný paket“ (ICMP typ 3).

Každý směrovač je takto testován 3x (třemi pakety) pro vyloučení chvilkových náhodných výpadků. Jednotlivé adresy si program postupně ukládá do paměti a výsledky trasování postupně zobrazuje v přehledné tabulce společně

s přeloženými DNS jmény daných adres pro lepší přehlednost. V případě, že daný směrovač neodpovídá, jsou místo těchto záznamů zobrazeny pouze znaky *

Rozdíl mezi výše uvedenými programy je v druhu odesílaných paketů.

Tracert používá stejně jako Ping ICMP paket „Žádost o echo“.

Traceroute naproti tomu posílá UDP datagramy – to je výhodnější z důvodu, kdy některé stroje mohou ICMP komunikaci blokovat. Pomocí přepínačů je pak možné zvolit vlastní číslo UDP portu, případně pomocí dalších přepínačů i jiný protokol (ICMP a další). Traceroute proti Tracertu ve výpisu může ještě zobrazovat další dodatečné znaky jako !H pro „nedostupný uzel“, !N – „nedostupná síť“ a další.

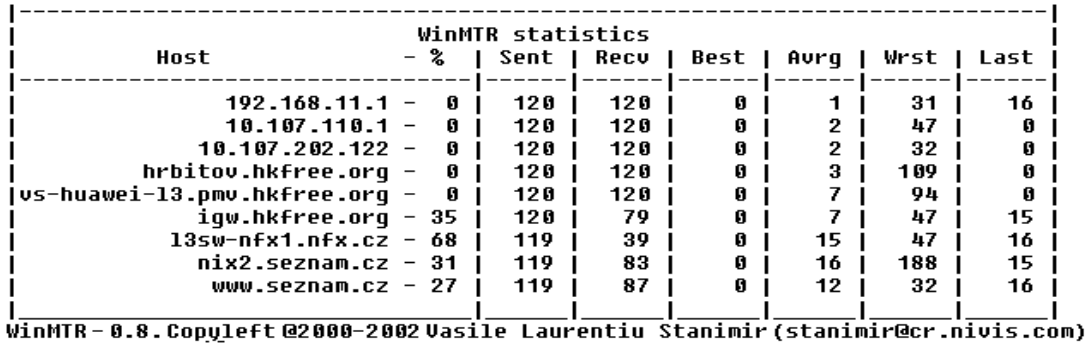
```
C:\>tracert www.seznam.cz

Výpis trasy k www.seznam.cz [77.75.76.3]
s nejvýše 30 směrováními:

  1     1 ms    < 1 ms    < 1 ms  192.168.11.1
  2     2 ms     1 ms     2 ms  10.107.110.1
  3     2 ms     2 ms     2 ms  10.107.202.122
  4     3 ms     2 ms     2 ms  hrbitov.hkfree.org [10.107.202.9]
  5     5 ms     5 ms     4 ms  vs-huawei-l3.pmv.hkfree.org [10.107.99.97]
  6     4 ms     2 ms     4 ms  igw.hkfree.org [10.107.101.129]
  7     *        *        *     Vypršel časový limit žádosti.
  8    86 ms   192 ms     6 ms  nix2.seznam.cz [194.50.100.194]
  9     5 ms     6 ms     5 ms  www.seznam.cz [77.75.76.3]

Trasování bylo dokončeno.
```

Obrázek 2 - Výstup programu Tracert

1.2.4. MTR a PathPing

MTR je diagnostický software (Unixový, existuje i verze pro MS Windows – WinMTR), kombinující jak metod Pingu, tak Traceroute. Program provede trasování a zobrazení směrovačů, ovšem na rozdíl od Traceroute tím nekončí, ale provádí nekonečné testování všech bodů a zároveň každého z bodů zvlášť. U každého z nich pak počítá a zobrazuje ztrátovost paketů a s tím i jakousi procentuální spolehlivost (jak bylo popsáno výše, zahazování ICMP paketů na některém bodu nemusí znamenat jeho poruchu).

V MS Windows novějších verzí je obsažen konzolový program PathPing fungující na stejném principu, ovšem statistiky pro jednotlivé body nezobrazuje ihned, ale provede sadu měření každého bodu na pozadí a až nakonec vypíše výsledné statistiky.



WinMTR statistics

Host	%	Sent	Recv	Best	Avrg	Wrst	Last
192.168.11.1	0	120	120	0	1	31	16
10.107.110.1	0	120	120	0	2	47	0
10.107.202.122	0	120	120	0	2	32	0
hrbitov.hkfree.org	0	120	120	0	3	109	0
vs-huawei-13.pmv.hkfree.org	0	120	120	0	7	94	0
igw.hkfree.org	35	120	79	0	7	47	15
l3sw-nfx1.nfx.cz	68	119	39	0	15	47	16
nix2.seznam.cz	31	119	83	0	16	188	15
www.seznam.cz	27	119	87	0	12	32	16

WinMTR - 0.8. Copyleft @2000-2002 Vasile Laurentiu Stanimir (stanimir@cr.nivis.com)

Obrázek 3 - Výstup programu WinMTR

1.2.5. Ostatní ICMP zprávy

ICMP protokol obsahuje řadu dalších typů zpráv, jenž jsou uvedeny v normě RFC 792.

1.3. Protokoly ARP a RARP

Tyto pomocné protokoly slouží k překladu síťové IP adresy na linkovou MAC adresu.

Ve chvíli, kdy chce zařízení na LAN síti komunikovat s druhým zařízením, musí znát kromě jeho IP adresy i jeho linkovou adresu. K jejímu zjištění slouží protokol ARP (Adress resolution protocol), jenž využívá broadcastu na linkové úrovni, kdy vytvoří linkový rámec s vlastní zdrojovou linkovou adresou a jako cílovou použije adresu broadcastovou. Díky tomu je rámec odeslán všem zařízením v daném segmentu sítě. V těle zprávy je zároveň požadavek na odpověď od zařízení s požadovanou IP adresou. Z příchozí odpovědi je poté zjištěna MAC adresa cílového zařízení, spárována s danou ip adresou a následně je již možné s ním dále běžně komunikovat. Pro případné budoucí použití si tyto informace po určitou dobu uchovává ve své paměti.

Protokol RARP slouží k opačnému překladu z linkové MAC adresy na adresu IP (její zjištění). Používá se spíše u bezdiskových stanic nebo při diagnostice, zda-li se na síti nenacházejí zařízení s duplicitní adresou. Pro případ zjišťování IP adresy se již téměř nepoužívá – byl nahrazen novějšími protokoly, jako DHCP.

2. Diagnostické softwary

V této kapitole budou uvedeny některé další známé nebo zajímavé softwary využívané při diagnostice a hledání závad v sítích, které využívají základních vlastností protokolů z rodiny TCP/IP. Díky jejich popularitě a rozšířenosti je k nim snadno dostupných dostatek návodů a informací, ať už na Internetu nebo i v knižní podobě.

2.1. TcpTraceroute

Tento software slouží podobně jako dříve zmíněné Traceroute nebo MTR k prověření trasy k danému cíli. Na rozdíl od nich však k tomu nepoužívá protokol ICMP ani UDP, ale, jak je patrné z názvu, protokol TCP. Vzhledem k dnes častému použití firewallů mezi sítěmi na trase k cíli, nemusí být předchozí metody úspěšné. Tento program se proto snaží trasování provádět skrze TCP porty známých služeb, které jsou dostupné pro příchozí spojení i skrze firewall, z vnějšku sítě. Snaží se na ně posílat TCP pakety s příznakem SYN, značící navazování spojení. V případě, že daný port je otevřený (naslouchá na něm nějaká služba), odpoví TCP paketem s příznaky SYN a ACK a spojení je tcptraceroutem ukončeno ještě před úplným navázáním spojení. V případě, že na daném portu nic nenaslouchá (je zavřený, neprůchozí), vrátí se od sledovaného zařízení TCP paket s příznakem RST pro reset, respektive ukončení spojení. Některé restriktivnější a inteligentnější firewally však mohou blokovat veškeré pokusy o komunikaci, takže jsou pak i tyto metody neúspěšné. Program je volně dostupný a je šířen pod GNU-GPL licencí.

Domovská stránka: <http://michael.toren.net/code/tcptraceroute/>

2.2. NetStat

Dalším programem využívaným při diagnostikách, již obsaženým v nejrozšířenějších operačních systémech, je program NetStat zobrazující nám všechna odchozí a příchozí spojení transportní vrstvy (TCP, UDP), případně při zvolení doplňujících parametrů i otevřené porty s naslouchajícími službami.

2.3. TcpDump

Velmi užitečný konzolový program pro diagnostiku problémů se sítí, umožňující zachytávat veškerý datový provoz procházející skrze vybrané rozhraní daného zařízení. Původně unixový program, v současné době existuje i verze pro MS Windows – WinDump, šířený pod BSD licencí. Jeho výhodou je právě konzolová jednoduchost, což nám umožňuje jeho použití skrze vzdálené připojení i na routerech a jednoduchých zařízeních bez grafického prostředí.

Domovská stránka: <http://www.tcpdump.org/>

2.4. Wireshark (Ethereal)

Stejně jako předchozí TcpDump nám program Wireshark (původní název Ethereal) slouží k zobrazení datového provozu skrze síťové rozhraní zařízení. Na rozdíl od něj je však tento určen pro grafické prostředí, čímž nám proti předchozímu dovoluje přehlednější zobrazení každého datového rámce a jeho analýzu. K tomu umožňuje, kromě vlastního zachytávání i načtení uložených záznamů spojení z mnoha podobných softwarů, mimo jiné i TcpDumpu, což nám usnadní jejich následné prohlížení a analýzy. Dále dovede ze sady zachycených paketů analyzovat a zobrazit celé jedno spojení od jeho započetí až po ukončení a mnoho dalších funkcí. Program má svojí verzi pro Unixové systémy i pro MS Windows, je zdarma šířen pod GNU GPL licencí.

Domovská stránka: <http://www.wireshark.org/>

2.5. TcpMon

Dalším programem pro odchytávání síťového provozu je TcpMon, který na rozdíl od předchozích psaný v programovacím jazyce Java, což umožňuje jeho velkou multiplatformnost. Jeho další výhodou je, že obsahuje obě části – serverovou, sloužící k odchytávání dat na vzdáleném stroji a k tomu grafickou klientskou, které jsou tyto data zasílána pro další analýzu. Je též šířen jako vlně dostupný software pod BSD licencí.

Domovská stránka: <http://tcpmon.dev.java.net/>

2.6. Nmap

Nmap je velmi silný nástroj, umožňující zjistit jaké služby běží na vzdáleném síťovém zařízení a řadu dalších informací celkově o daném zařízení, případně i takto prohledat celou síť a výsledky poté uživateli zobrazit. Tím nám umožňuje hledat případné slabiny v síti, bohužel ho však zneužívají i hackeři k následnému napadání těchto systémů . Ze získaných dat poté dovede zobrazit případně i topologickou mapu této sítě. Program je konzolový, ale existuje k němu i grafická nadstavba jménem Zenmap, je volně dostupný, šířený pod GNU GPL licencí a funguje na Unixových systémech, MS Windows a dalších platformách.

Domovská stránka: <http://nmap.org/>

3. Protokol SNMP

Vzhledem k omezeným možnostem protokolů obsažených v TCP/IP k diagnostice a správě síťových zařízení, byl vyvinut protokol SNMP (Simple Network Management Protocol) – v překladu „Jednoduchý protokol pro správu sítě“.

3.1. Historie

„Protokol SNMP vznikl na konci 80.let v za tímto účelem "ad hoc" svolaném pracovním výboru IAB (Internet Architecture Board, www.isi.edu/iab) pro správu směrovačů Internetu. Vyvinul se jako jedna varianta protokolu SGMP (Simple Gateway Monitoring Protocol), který byl navržen koncem roku 1987 právě pro výměnu informací mezi směrovači a branami této, v té době ještě akademické sítě. Druhou variantou, která pak vznikla z protokolu SGMP na půdě organizace ISO, byl protokol CMIP (Common Management Information Protocol).

I když zpočátku byla snaha společného vývoje obou vzniklých verzí - minimálně na úrovni společné struktury správcovských informací SMI a informační databáze MIB, brzy se toto řešení ukázalo nepraktické a další vývoj probíhal nezávisle. Důvodem byla hlavně objektová orientace CMIP na rozdíl od SNMP.

Přestože CMIP byl pokusem ISO vytvořit standard s maximální možnou podporou protokolů a služeb s definovanou databázovou strukturou pro přenos pomocí protokolu TCP/IP (CMOT - CMIP over TCP/IP), stejně jako celá sada protokolů OSI nenašel větší podporu u výrobců ani uživatelů a nedočkal se tak významnějšího rozšíření.“ [6]*(Klaška, 2000*)

Protokol SNMP postupně prošel dalším vývojem, v jeho průběhu vzniklo několik verzí, nejnovější je dnes verze 3 (SNMPv3).

3.2. Filozofie a architektura

Jak již bylo naznačeno výše, SNMP je jednoduchým protokolem pro správu síťových zařízení, tedy pro jejich vzdálené nastavování a diagnostiku jejich chodu. Je to aplikační protokol využívající protokol IP na síťové vrstvě a protokol UDP na vrstvě transportní. Ke komunikaci využívá architekturu klient-server s filozofií, kdy sledované zařízení obsahuje co nejjednodušší software (agent), z důvodu, aby co nejméně ovlivňoval samotný chod tohoto zařízení (zejména u těch pomalejších), který zpracovává informace o jeho chodu. Ty jsou na základě požadavků zasílány monitorovacímu zařízení (manažer), který poté provádí veškeré složitější operace. Výhodou tohoto systému je to, že máme v síti pouze minimální množství manažerů (většinou jeden), ovládaných správcem, a velké množství jednoduchých agentů. Naopak nevýhodou tohoto systému, oproti jiným, je větší zatížení sítě, které se však snaží eliminovat právě použitím jednoduššího nepotvrzovaného protokolu UDP. Tento protokol může, kromě tradičního použití v TCP/IP sítích, pracovat i v sítích s jinými protokoly jako například IPX/SPX. Jeho popis je obsažen v RFC 1157.

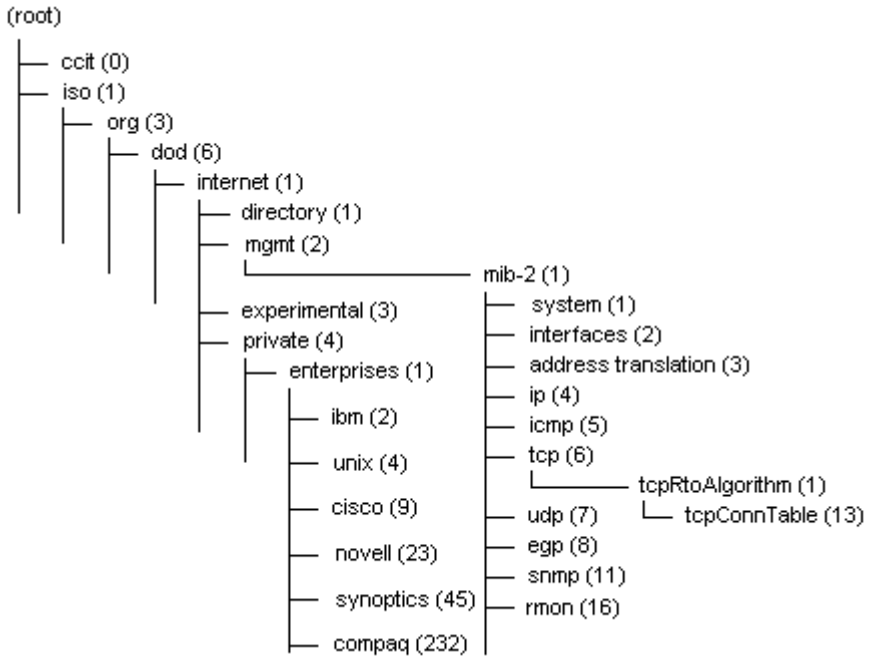
3.3. Struktura dat – MIB databáze

V rámci výměny dat mezi agentem a manažerem byla vytvořena sada identifikátorů, které nám popisují jednotlivé spravované objekty. Ta se nazývá MIB (Management Information Base) a je tvořena hierarchickou stromovou strukturou jednotlivých popisků objektů (OID – Object Identifier), které jsou pro správu daného zařízení protokolem SNMP používány. Každý objekt (proměnná) patřící do některé ze tříd stromu má v rámci ní jako identifikátor přiřazeno přirozené číslo. Pro přesnou identifikaci každého z nich je jeho OID tvořen posloupností čísel tříd od kořene stromu až k němu samotnému, které jsou odděleny tečkami. Z důvodu lepší čitelnosti má zároveň každý uzel vlastní textový popis. Zápis je pak možný jak pomocí čísel, tak pomocí slov, ovšem pro vzájemnou komunikaci se používá číselná podoba.

Tato struktura a pravidla její tvorby je popsáno v SMI (Structure of Management Information) v dokumentech RFC1155, RFC1212 a RFC1213, které definují novější MIB – 2, společně s některými úpravami.

Podstrom MIB-2 nacházející se v poduzlu 1.3.6.1.2.1 = „.iso.org.dod.internet.mgmt.mib-2“ je vrcholem SMI stromu pro mib-2 objekty a každý z nich začíná tímto číslem. Každé zařízení s podporou SNMP by mělo alespoň tuto část, s nejčastěji používanými proměnnými, podporovat. Pro specifikaci vlastních mib objektů pro různá zařízení různých výrobců je pak určena větev „.iso.org.dod.internet.private.enterprises", kde si každý výrobce může vytvářet vlastní podvětve k tomu určené.

K definici datových struktur byl pro SNMP operace použita část formálního abstraktního jazyka popsaného v normě ISO ASN.1 (Abstract syntax notation). Pro případ přenosu po síti mezi zařízeními s různou architekturou používá pravidla pro kódování dat BER (Basic Encoding Rules – ISO 8825), která určuje uspořádání bitů, kódování typů a numerických prezentací, kde každý objekt je kódován trojicí: příznak, délka, hodnota.



Obrázek 4 - MIB struktura
Zdroj: [6]

3.4. SNMP verze 1 (SNMPv1)

Zprávy protokolu SNMP jsou mezi zařízeními předávány různým způsobem a zařízení mají následující úlohy:

- **Agent** – přijímá požadavky na zpřístupnění nebo změnu MIB proměnných specifikovaných v požadavku.
- **Manažer** – vysílá požadavky na zpřístupnění hodnot MIB proměnných, které po návratu, ve formě zprávy, zpracovává. Dále posílá požadavky na změnu některých z proměnných – konfigurace zařízení.

Mezi těmito zařízeními jsou pak vyměňovány následující druhy zpráv:

- **GetRequest** – Manažer si vyžádá informaci z MIB databáze Agenta
- **GetNextRequest** – stejně jako GetRequest, ovšem umožňuje získávání dat z proměnných (objektů) bez toho abychom znali její jméno – je vyžádána následující proměnná po aktuální. Tímto způsobem je možné procházet celým MIB stromem, kdy si vyžádáme jeden objekt se známým a většinou podporovaným OID a poté procházíme celý strom.
- **SetRequest** – Manažer vyšle požadavek Agentovi na změnu hodnoty některé proměnné, který ji provede.
- **GetResponse** – Odpověď Agenta Manažerovi na jeho požadavek (GetRequest).
- **Trap** – jediný případ, kdy Agent vysílá zprávy Manažerovi bez jeho předchozího požadavku. Tyto zprávy jsou zasílány v případě splnění předem stanovených podmínek nebo v případech neočekávaných stavů (chyb zařízení, přetečení prahových hodnot atd.). Tento typ zpráv je z důvodu co nejlepší průchodnosti sítí při jejím zahlcení nebo chybovosti, posílán bez následného potvrzování Manažerem, takže Agent nemá jistotu jejich doručení.

3.4.1. Struktura SNMPv1 zpráv

SNMP protokol využívá pro přenos svých zpráv transportní protokol UDP. Na straně Agenta je standardně dostupný na UDP portu 161 a na straně Manažera je využíván UDP port 162 pro příjem „trap“ zpráv od Agenta.

Zprávy jsou děleny na dvě části – hlavičku a PDU (Protocol Data Unit). V hlavičce je uvedena verze protokolu (pro SNMPv1 obsahuje hodnotu 0) a takzvaný „Community String“. Ten slouží k identifikaci, do jaké „SNMP komunity“ komunikující strany patří. Toto slouží v první verzi SNMP jako zabezpečovací a autentizační mechanizmus pod názvem „community-based security mechanism“. PDU je pak datová část obsahující definici posílané zprávy.

Všechny verze zpráv v PDU mají stejný formát, kromě zpráv Trap jenž je odlišný. Formát základních zpráv obsahuje v PDU následující položky:

- **Typ PDU** – rozlišení typu posílané zprávy (0 = GetRequest, 1 = GetNextRequest, 2 = GetResponse, 3 = SetRequest)
- **Request ID** – číslo sloužící k párování odpovědi k příslušnému požadavku – číslo obsažené v příchozím požadavku je zkopírováno do odpovědi.
- **Error Status** – číslo, určující v odpovědi „GetResponse“ typ odpovědi (typ chyby) – hodnota 0 označuje „žádná chyba“ a používá se v běžných odpovědích, hodnoty 1 – 5 pak označují různé druhy chyb.
- **Error index** – v případě, že „Error Status“ je různý od nuly, obsahuje tato položka ukazatel do položky „Variable bindings“, jenž obsahuje seznam objektů, které se dané chyby týkají.
- **Variable Bindings** – obsahuje seznam objektů (proměnných) a jejich hodnoty, které s danou zprávou souvisí

Formát PDU zprávy „Trap“:

- **Typ PDU** – viz. výše, pro „Trap“ obsahuje hodnotu 4
- **Enterprise** – obsahuje typ objektu, který vyvolal Trap
- **Agent adress** – adresa objektu, který vyvolal Trap
- **Generic trap type** – základní typ (kód) zprávy Trap
- **Specific trap code** – upřesňující kód zprávy Trap
- **Time stamp** – uvádí čas (vzhledem k objektu sysUpTime) mezi poslední reinicializací sítě a vygenerováním Trapu
- **Variable bindings** – stejné jako u předchozích typů

3.5. SNMP verze 2 (SNMPv2)

Vzhledem k některým nedostatkům první verze byl zapokračován vývoj verze další. Ta přidává některé nové prvky:

- **SNMP Kontext** (Context) a MIB View – kolekce MIB objektů v zařízení, které spolu určitou logickou vazbou souvisí. Těchto pojmenovaných kolekcí může zařízení obsahovat více a mohou být buď lokálního nebo vzdáleného (proxy) typu.
- **SNMP Proxy** – SNMP agent, který pouze přebírá a předává požadavky pro jiného agenta – dělá prostředníka v komunikaci mezi ním a Manažerem.
- **Zabezpečení** – vzhledem k nedostatečnému zabezpečení, přidává proti předchozí verzi možnost autentizace pro přístup k objektům a s tím související zpřístupnění pouze části objektů pomocí omezení přístupu skrze „MIB view“. Dále umožňuje zvýšení bezpečnosti přenášených informací šifrováním.
- Nové SNMPv2 zprávy:
- **GetBulk** – umožňuje čtení informací z více MIB objektů najednou – usnadňuje získávání informací z tabulek
- **Inform** – slouží pro nově přidanou možnost komunikace mezi více manažery, jenž je potřebná ve větších společnostech při správě velkého množství klientů. Také využívá mechanismu požadavek-odpověď a trap zpráv, ovšem v tomto případě potvrzovaných příjemcem.

3.5.1. Zabezpečení v SNMPv2

Vzhledem k nedostatečnému zabezpečení v předchozí verzi, přidává druhá verze možnost autentizace pro přístup k objektům a s tím související zpřístupnění pouze části objektů pomocí omezení přístupu skrze „MIB view“. Metody zabezpečení a konfigurace v nové verzi byly však považovány za příliš složité. Proto vzniklo několik vzájemně nekompatibilních verzí - „Community-Based Simple Network Management Protocol version 2“ neboli „SNMPv2c“ (RFC 1901–1908) – ta využívá jednodušších metod zabezpečení pomocí „komunit“ z verze 1. Původní verze je proto označována jako „SNMPv2p“ (RFC 1445). Z důvodu opětovného zvýšení bezpečnosti vznikla verze další – „User-Based Simple Network Management Protocol version 2“ neboli „SNMPv2u“ (RFC 1909-1910), která už není tolik složitá

jako původní verze. Další verzí je pak ještě “SNMPv2 star” - SNMPv2*, která je komerční verzí a není standardizována.

3.5.2. Struktura SNMPv2 zpráv

Vzhledem k rozdílnosti jednotlivých verzí protokolu jsou různé i formáty jejich zpráv. Ty se liší ve formátu hlavičky, datová část (PDU) je pak pro všechny stejná. Z důvodu jejich velkých rozdílů, obsáhlosti a malému rozšíření v praxi je zde nebudu uvádět a odkážu na uvedené dokumenty RFC, případně popis v literatuře [2]. Z těchto několika verzí je jediná SNMPv2c kompatibilní s verzí první, díky použití stejného zabezpečovacího mechanismu.

Formát části pro základní zprávy PDU je stejný jako u první verze. K tomu přidává další čísla pro nové typy zpráv (5 – GetBulkRequest, 6 – InformRequest, 7 – Trap verze 2, 8 – Report) a rozšiřuje počet možných chybových zpráv.

Formát PDU zprávy **GetBulkRequest** pro příjem obsáhlejších dat (tabulek):

- Typ PDU a Request ID – stejné jako u základních zpráv – obsahuje typ 5
- Non Repeaters – určuje počet objektů na začátku jejich seznamu, které nemají být opakovaně zpracovávány
- Max Repetitions – určuje maximální počet opakování jednoho druhu objektu v odpovědi
- Variable Bindings – stejné jako u základních zpráv

Formát PDU zprávy **Trap verze 2**:

Na rozdíl od první verze je Trapv2 součástí základního PDU a jeho proměnné jsou uloženy v poli „Variable Bindings“. První proměnnou je „Timestamp“ se stejnou funkcí jako v první verzi. Za ní následuje „snmpTrapOID“, jenž určuje typ zprávy Trap. Popis těchto druhů zpráv je umístěn v „enterprise“ části MIB tabulky, což přináší výhodu jednoduchého přidávání nových. Za nimi následují volitelné identifikátory (objekty), které tuto zprávu vyvolali.

3.6. SNMP verze 3 (SNMPv3)

Další verze SNMP, definovaná v RFC 2271–2275, 2571–2575 a 3411-3418, vznikla z důvodu nekompatibility předchozích verzí. Je založena na základních principech z verze 1 a z verze 2 také přebírá některé prvky. Dále byl přepracován formát zpráv, vylepšeny bezpečnostní přístupové (autentizační) metody a algoritmy pro šifrování zpráv. Nový protokol byl navržen s ohledem na jednoduchost a větší modulárnost – umožňuje výběr a vývoj nových zabezpečovacích technik bez potřeby měnit základní strukturu zpráv.

Důležitou novinkou je pojem „SNMP Entita“. Ta je tvořena Manažerem nebo Agentem. Příslušné dokumenty pak obsahují doporučení pro jejich tvorbu a architekturu. SNMP Entita je rozdělena na dva základní prvky:

SNMP engine, který je dělen na následující moduly:

- **Dispatcher** – modul pro komunikaci se sítí a řízení provozu mezi moduly. Přijímá a odesílá zprávy a na základě identifikátoru verze SNMP je předává příslušnému modulu k dekódování (kódování) zprávy.
- **Message Processing Subsystem** – tento podsystém modulů slouží pro udržení zpětné kompatibility jednotlivých verzí SNMP – k dekódování informací z různých verzí SNMP. Po dekódování dané zprávy je datová část (PDU) opět předána Dispatcheru
- **Security Subsystem** – systém sloužící k zabezpečovacím účelům. Po rozbalení je zpráva v závislosti na verzi SNMP předána příslušnému bezpečnostnímu podmodulu, korespondující s danou verzí. Po projití kontrolními mechanismy je předána zpět Dispatcheru společně s výsledky analýzy
- **Access Control Subsystem** – modul pro kontrolu přístupových práv

Druhým je „SNMP Applications“, obsahující následující moduly, jenž slouží k zpracování samotné vybalené PDU zprávy:

- **Command generator** – slouží při odesílání k vytvoření příslušné PDU, jenž je předána k dalšímu zpracování
- **Command receiver** – provádí dekódování přijaté PDU
- **Command responder** – vytváří odpovědi na požadavky „Manažera“
- **Notification receiver** – přijímá zprávy Trap od agentů

- **Notification originator** – vytváří a posílá Trap zprávy Manažerovi
- **Proxy forwarder** – slouží k předávání zpráv od subagentů

3.6.1. Struktura SNMPv3 zpráv

S pozměněným systémem a vylepšením bezpečnosti souvisí i nový formát SNMP zpráv, obsahující následující položky:

- **Verze** – stejné jako u předchozích – pro tuto verzi obsahuje hodnotu „3“
- **Identifier (ID)** – obdobná funkce jako „Request ID“ v PDU. Bylo přidáno, aby mohlo být použito nezávisle na obsahu PDU a ke zvýšení bezpečnosti při přenosu.
- **Max Size** – maximální velikost zprávy, kterou odesílatel této zprávy přijme. Minimální hodnota je 484
- **Flags** – skupina řídících identifikátorů ovlivňujících zpracování a zabezpečení zpráv:
  - *Reportable Flag* – pokud je nastaven na 1, musí příjemce zprávy odeslat zpět zprávu „Report-PDU“
  - *Privacy Flag* – pokud je nastaven na 1, indikuje že zpráva používá zabezpečení (je kryptována)
  - *Authentication Flag* – pokud je nastaven na 1, indikuje že zpráva je autentizovaná
- **Security model** – číslo, určující který bezpečnostní mechanismus je použit
- **Security parameters** – dodatečné parametry související se zvoleným zabezpečením
- **ContextEngineID** a **ContextName** – určuje který SNMP Kontext má být pro tuto zprávu použit
- **PDU** – samotná datová část SNMP zprávy. Formát PDU zpráv je převzat ze SNMPv2.

3.6.2. Zabezpečení v SNMPv3

Třetí verze protokolu umožňuje vlastní výběr zabezpečovacích mechanismů. Jako základní používá pro kryptování a zabezpečení přenosu zpráv systém „User-based Security Model“ (USM) převzatý z SNMPv2u původně definovaný v RFC 1910, pro SNMPv3 pak v RFC 2274, 2574 a 3414. Pro kontrolu integrity a ověření odesílatele využívá kontrolních algoritmů MD5 a SHA a systému jména a hesla. Pro kryptování zpráv pak algoritmus DES. Dále dokument obsahuje definici MIB pro vzdálenou konfiguraci USM. Pro přenos je však možné využívat i vlastních metod jako SSH, TLS nebo další.

Ke kontrole oprávnění pro přístup k MIB objektům je využíván „View-based Access Control Model“ (VACM) definovaný v RFC 2275, 2575 a 3415. Podobně jako v předchozích verzích, používá systém skupin, ovšem nenazývá je „community“ ale „group“. Pro každou skupinu je pak definováno kam mají její členové přístup, povolené operace a vyžadovaný stupeň zabezpečení. Dokument také obsahuje definici MIB pro vzdálenou konfiguraci svých parametrů

Pozdější RFC 3584 pak definuje koexistenci jednotlivých verzí SNMP.

3.7. Podpora protokolu SNMP

Protokol SNMP je v současnosti nejznámější a nejpoužívanější protokol pro správu sítě. Svědčí o tom jeho podpora ve velkém množství jak síťových zařízení, tak i v nejpoužívanějších operačních systémech (Unix, Linux, MS Windows). Jeho podpora je implementována do nejpoužívanějších programovacích jazyků. Díky hotovým modulům je tak možné v relativně krátké době umožnit tvorbu nových softwarů a přidávat jeho podporu do již existujících.

3.8. RMON (Remote Network Monitoring)

Standard RMON slouží k monitorování vzdálených segmentů sítě skrze pomalejší WAN sítě. Není novým protokolem, ale je rozšířením protokolu SNMP a je specifikován v dokumentech RFC 1271, 1757 a 2819. Vzhledem k použití na vzdálené monitorování, používá rozdílnou architekturu – agenti umístění v pozorované síti se nazývají „sondy“. Na rozdíl od SNMP agentů nejsou pouze jednoduchou jednotkou předávající informace manažerovi, ale jsou inteligentními jednotkami pro sběr a uchovávání informací o daném segmentu sítě. Tento způsob sice zvyšuje zátěž monitorovacího zařízení, ovšem proti SNMP přináší výhodu v menším zatížení sítě. Při použití základního SNMP totiž Agent udržuje pouze

souhrnná data o každém čítači (proměnné), proto pokud chceme získávat časově závislé statistiky (například o datovém toku), musí se Manažer v pravidelných intervalech dotazovat na jejich novou hodnotu (provádět tzv. Polling), čímž se zvyšuje zatížení sítě. Princip použitý v RMON nám umožňuje v době většího zatížení sítě sbírat pouze minimum potřebných dat a další statistická a historická data o provozu je pak například možné přenášet až v době menšího vytížení sítě a následně archivovat. Stejně tak v případě výpadku připojení k Manažerovi může Sonda dál sbírat data o místní síti a po následné obnově spojení je předat. Zároveň obsahuje optimalizace pro přenos tabulkových dat po pomalých sítích.

Sonda může být implementována jako samostatný hardware připojený do segmentu sítě nebo umístěna uvnitř některého z prvků sítě. Sběr informací o provozu je prováděn pouze na linkové vrstvě (MAC) na lokálním segmentu sítě. V první verzi byl vyvinut pro sítě Ethernet, později byl rozšířen i o podporu Token Ring a další druhů sítí.

RMON není další sadou protokolů, ale pouze přidává nové objekty (převážně tabulky) do MIB databáze, potřebné pro sběr a ukládání informací a statistik o přenosech na linkové vrstvě. Pro komunikaci mezi konzolí správců (Manažery) a sondami je pak využíván protokol SNMP.

Data, která může sonda shromažďovat a poskytovat, jsou dělena do několika skupin – vedení statistik o celkovém provozu, o jednotlivých hostech (podle jejich MAC adres) a jejich historie. Dále sledování překročení hraničních hodnot a následné vyvolání Trap zpráv nebo filtrování zachytávaných paketů.

3.9. RMON2

Původní návrh RMON, vzhledem k faktu, že umožňuje monitorovat pouze linkovou vrstvu sítí (MAC), přestal po čase dostačovat. Novější sítě jsou, díky použití inteligentnějších přepínačů, již tvořeny mnoha segmenty a jsou celkově rozlehlejší, čočby znamenalo použití velkého množství sond. Proto byl vyvinut novější standard, který je schopen monitorovat aktivitu sítě i na ostatních vyšších vrstvách OSI modelu (3.síťové až po aplikační). Aby toto bylo možné, musela být rozšířena MIB databáze o nové druhy objektů pro sběr statistik o provozu na příslušných protokolech. RMON2 podporuje jak protokoly z rodiny TCP/IP, tak i IPX/SPX a řadu dalších. Nový standard je popsán v dokumentech

RFC 2021 a 4502. RMON1 se tedy zaměřuje spíše na chyby na fyzické úrovni, RMON2 na kompletní síťový provoz.

Vzhledem k faktu, že původně byl RMON koncipován pro sítě 10mbit Ethernet, vyvstala potřeba jej modernizovat i pro nové rychlejší (a nepaketové) sítě. Proto byl návrh rozšířen, pomocí RFC 3273 a 4502, přidáním nových objektů, k tomu potřebných, do MIB databáze.

Mimo návrhy dostupné v RFC si výrobci doplňují MIB databáze o nové objekty pro podporu dalších technologií, jako například podporu virtuálních sítí VLAN.

4. Další protokoly pro správu sítě

4.1. WBEM

WBEM (Web-Based Enterprise Management) je množina standardů vytvořených ke sjednocení metod pro správu výpočetních systémů a sítě, založená na webových technologiích. Společně s čím dál větším rozšiřováním webových technologií se i správa sítí a zařízení na nich připojených přesouvá směrem k webu a technologiím s ním spojených.

Webová řešení přístupu ke správě přinášejí řadu výhod, ale zároveň i rizik. Výhodou je možnost přístupu ke správní konzoli, případně přímo do konfigurace zařízení, v případě připojení do Internetu, téměř odkudkoliv. Data o provozu jednotlivých zařízení mohou být uložena na jednom centrálním serveru (databázi) s možností přístupu jak z aplikační konzole, tak pomocí webového rozhraním. Právě webové rozhraní nám umožňuje přístup téměř odkudkoliv kde je k dispozici webový prohlížeč (dnes téměř všude). Na druhou stranu však přináší právě tato otevřenost systému do veřejných sítí velké bezpečnostní riziko, proto musí být dbáno jak na zabezpečení serveru samotného, tak přístupu do jednotlivých částí administrace.

Vzhledem k různorodému přístupu výrobců k tomuto problému vyvstala potřeba nějakým způsobem, stejně jako dříve, tyto metody sjednotit - standardizovat. Z toho důvodu spojilo síly několik společností a vytvořili společné sdružení DMTF (Distributed Management Task Force), které by mělo tomuto sjednocení napomoci.

4.1.1. CIM

Jedním ze základních problémů byla integrace spravovaných dat z různých systémů do jedné databáze. Proto byla vytvořena definice CIM (Common Information Model), jenž přináší metody pro jednotný popis těchto informací bez ohledu na jejich zdroj. Vzhledem k narůstajícímu rozšiřování a oblibě objektově orientovaných prostředí, byl již tento standard navržen tak, aby umožnil implementaci do jejich prostředí a jejich vzájemnou výměnu informací (pomocí standardů jako CORBA nebo COM). Umožňuje nám tedy sběr dat z rozličných systémů, jako SNMP, DMI, CMIP/CMOT atd. a následný jednotný přístup k těmto agregovaným datům. Standard umožňuje nejen sběr dat, ale i jednotný přístup ke konfiguraci jednotlivých zařízení a jejich softwaru.

4.2. WMI

WMI (Windows Management Instrumentation) je řešení pro správu počítačových systémů, jejich operačních systémů a aplikací. Jak naznačuje název, jeho tvůrcem je společnost Microsoft, která jej implementuje do svého operačního systému Windows. Je vytvořen jako vlastní implementace standardu WBEM (Web-based Enterprise Managementu) od DMTF (Distributed Management Task Force) a z modelu CIM, jenž je součástí WBEM přebírá sadu tříd objektů pro správu těchto systémů. Díky tomu je se standardem WBEM kompatibilní.

4.3. DMI

Správa sítě neznamená pouze správu jednotlivých zařízení umožňujících její chod (switche, routery atd.), ale i koncových zařízení. Proto byl vytvořen standard DMI neboli „Desktop Management Interface“ k tomu určený. Stejně jako protokol SNMP umožňuje vzdálené získávání dat o činnosti zařízení nebo o verzích jeho softwaru, jenž jsou posílána a zpracovávána na konzoli pro správce. Pro ukládání a manipulaci s daty používá databázi MIF, jenž je obdobou SNMP MIB databáze. Zároveň umožňuje spolupráci s ostatními protokoly, jako SNMP nebo CMIP a vzájemné konverze jejich MIB a MIF databází. Tím je umožněno integrovat správu všech zařízení do jednotných správních aplikací.

Standard DMI byl navržen a spravován sdružením DMTF. Díky malé podpoře výrobců hardwaru je málo rozšířený a DMTF ukončilo jeho podporu k 31.3.2005.

4.4. NetFlow

NetFlow je otevřeným protokolem, určeným ke sledování datových toků procházejících skrze síťová zařízení a následnou tvorbu jejich statistik. Na rozdíl od SNMP slouží pouze ke sběru dat, ne ke konfiguraci zařízení. Je vyvíjen společností Cisco (Cisco Systems). Jeho architektura je podobně jako u protokolu RMON dělena na dvě části – NetFlow sonda, jenž je umístěna uvnitř sítě a provádí sběr statistických dat, která jsou poté posílána NetFlow kolektorovi, jenž provádí jejich další zpracování, archivaci a případně zobrazení správci. Sběr dat může být implementován přímo uvnitř síťového prvku, nebo jako samostatná sonda s vlastním připojením pro přenos statistických dat, tak aby nebyl ovlivněn standardní provoz. K přenosu dat je využíván protokol UDP nebo SCTP (Stream Control Transmission Protokolu).

Na základě protokolu NetFlow v9 vznikl v nedávné době nový IETF standard - Internet Protocol Flow Information eXport (IPFIX), který si již získal oblibu a podporu mezi výrobci síťových technologií.

5. Softwary pro správu sítě

V průběhu mnoha let vývoje sítí a protokolů pro jejich správu vznikla řada softwarů – některé využívající pouze jeden standard a jiné kombinující řadu metod. Některé známé, hojně rozšířené nebo zajímavé zde uvedu. Mezi nimi jsou zástupci jak z řad komerčních, tak některých zdarma s dostupnými zdrojovými kódy (open-source). Fakt, že většina z nich podporuje protokol SNMP, svědčí o jeho velké oblíbenosti.

5.1. Net-snmp

Net-SNMP je standardním balíkem pro práci s protokolem SNMP v linuxových systémech. Obsahuje různé nástroje pro práci se SNMP pro příkazový řádek i pro grafické prostředí a příslušné démony. Jeho nástroje umožňují fungovat jako SNMP agenti v zařízeních i jako SNMP manažeři pro sběr a zpracování dat. Šířen je pod BSD licencí.

Domovská stránka: <http://net-snmp.sourceforge.net/>

5.2. MRTG

MRTG neboli „Multi Router Traffic Grapher“ je nástroj původně vytvořený pro grafické zobrazení průtoku dat v počítačových sítích, získaných pomocí protokolu SNMP. K této původní metodě získávání vstupních dat byla později přidána možnost tyto data vkládat na vstup i přes vlastní skript. To umožnilo, aby mohl být používán jako univerzální nástroj pro zpracování různorodých dat do grafické podoby. Například vytížení různých částí počítače a jeho OS (vytížení CPU, paměti, pevných disků atd.), zobrazování teploty apod. Grafy jsou generovány pro poslední hodinu, den, měsíc a rok, k čemuž je využíván vlastní algoritmus díky kterému zabírají potřebná data stále stejnou velikost, což umožňuje jeho použití i na zařízeních s malou úložnou kapacitou.

Aplikace je zdarma pod open-sourcovou licencí GNU-GPL a je napsána ve skriptovacím jazyce PERL, čímž umožňuje mulitplatformnost. Některé její části jsou však v programovacím jazyce C z důvodu zrychlení některých náročnějších operací. Jejím původním autorem je Tobias Oetiker, po zveřejnění zdrojových kódů se na projektu podílí řada dalších autorů.

Domovská stránka: <http://oss.oetiker.ch/mrtg/>

5.3. RRDtool

RRDtool je nástrojem s podobným zaměřením jako MRTG, vytvořeným stejným autorem. RRD je zkratkou pro název Round Robin Database, jenž označuje nový způsob ukládání dat, kde velikost této databáze je, po prvním vytvoření, v dalším průběhu jejích aktualizací neměnná. To přináší výhodu například při použití v jednoduchých zařízeních, kde velikost paměti je velmi malá. Na rozdíl od MRTG jsou sbíraná data průběžně ukládána do RRD databáze a grafy jsou poté překreslovány nezávisle v jiném čase. Dále proti MRTG neobsahuje konfigurační soubor, kde jsou specifikovány cíle sběru dat, ale tento sběr musíme provádět vlastním skriptem jazyka PERL. Zároveň obsahuje i rozhraní k dalším programovacím jazykům jako Python, Ruby, TCL nebo PHP. Vzhledem k velké popularitě je na Internetu dostupná řada připravených skriptů, což usnadňuje jeho použití. Stejně jako předchůdce je dostupný pod open-sourcovou licencí GNU-GPL.

Domovská stránka: <http://oss.oetiker.ch/rrdtool/>

Na RRD je založena nebo jej využívá celá řada dalších softwarů. Jejich obsáhlý seznam se pak nachází online na adrese <http://oss.oetiker.ch/rrdtool/rrdworld/index.en.html>. Některé z nich budou uvedeny níže.

5.4. Cacti

Cacti je webový systém pro dohled nad síťovými prvky. Umožňuje nám zobrazení průtoku dat a informací o činnosti daných zařízení (vytížení CPU, operační paměti, diskového prostoru atd.) pomocí grafů. K tvorbě těchto grafů využívá výše uvedený software RRD a k sběru informací protokol SNMP nebo vlastní skripty. Systém je možné rozšířit o další moduly pro jiné grafické zobrazení, jako třeba Weathermap, jenž nám umožňuje zobrazovat datové toky v přehledné mapě prvků. Software je zdarma pod licencí GNU-GPL.

Domovská stránka: <http://www.cacti.net/>

5.5. Nagios

Nagios je dalším systémem pro sledování stavu sítí a služeb na ní běžících. Systém automaticky sleduje jejich dostupnost a pomocí webového grafického rozhraní umožňuje tyto informace zobrazovat správci. V případě výpadku některého zařízení nebo služby umožňuje ihned, automaticky, na to upozornit správce pomocí emailů, sms nebo jiného informačního kanálu. To je velmi užitečná vlastnost, pokud je správce mimo dosah sítě, jenž mu umožňuje rychlý zásah a nápravu nastalé situace. Tento software je také zdarma pod licencí GNU-GPL.

Domovská stránka: <http://www.nagios.org/>

5.6. Zabbix

Zabbix kombinuje funkce předchozích softwarů. Je to webový systém, jenž umožňuje sledování stavu síťových zařízení a služeb na nich běžících, zobrazování těchto dat pomocí grafů a hlášení neobvyklých stavů správci pomocí externích informačních kanalů. Systém je také šířen pod licencí GNU-GPL.

Domovská stránka: <http://www.zabbix.com/>

5.7. Zenoss

Zenoss je obdobou systému Zabbix. Na rozdíl od předchozích je však komerčním softwarem, ovšem obsahuje i open-source verzi nazvanou Zenoss Core, jenž je dostupná pod licencí GNU-GPL

Domovská stránka: <http://www.zenoss.com/>

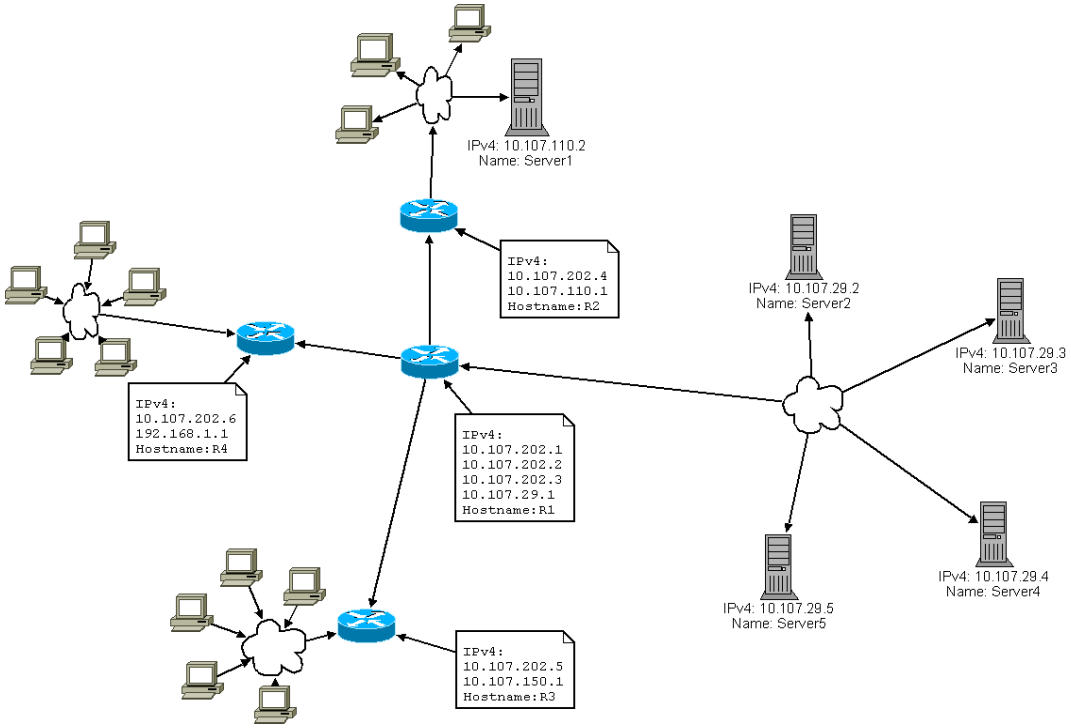
5.8. Software pro kompletní správu

Programů pro kompletní správu a sledování prvků IT infrastruktury skrze síť, včetně těch síťových, existuje velké množství. Proto zde uvedu pouze jejich seznam společně s jejich online domovskou stránkou.

- HP OpenView - <http://www.openview.hp.com/>
- IBM Tivoli - <http://www.ibm.com/tivoli/>
- Sun MicroSystems Solstice – <http://www.sun.com/software/>
- Novell ZENworks - <http://www.novell.com/products/zenworks/>
- Microsoft SMS Server - <http://www.microsoft.com/smsmgmt/>
- Compaq Insight Manger - <http://www.compaq.com/products/servers/management/>

6. Diagram sítě

Diagram sítě je grafické vyjádření (vizualizace) topologie sítě. Slouží člověku k vytvoření lepší představy o propojení jednotlivých síťových prvků dané sítě a případně zobrazení dodatečných informací o těchto prvcích a službách na nich běžících.



Obrázek 5 - Digram sítě

Důvodů, proč provádět získávání informací o topologii sítě, jejich prvcích a službách na nich běžících, může být více. Prvním z nich je jejich získávání administrátorem dané sítě pro její diagnostiku nebo zjišťování jejího zabezpečení. Druhou možností je pak použití útočníkem, jenž se snaží nějakým způsobem do sítě nabourat, ať už z důvodu jejího odstavení nebo získání citlivých informací na ní dostupných. Právě kvůli bezpečnostním důvodům je řada popsaných metod zakazována a znemožňována.

6.1. Tvorba diagramu

Návrh konkrétního řešení je závislý na důvodech jeho následného použití a místních podmínek. Vždy je však kombinací různých metod.

Jednou z možností je takzvaná ruční tvorba. Autor si nejprve zjistí informace o jednotlivých zařízeních připojených na sledovanou síť, jejich konfiguraci a jejich vzájemná propojení. Poté si ze získaných informací vytvoří představu o topologii této sítě, pomocí které následně vytvoří grafickou podobu. Ta může být vytvořena přímo ručně na papír nebo pomocí počítačového programu.

Další možností je nechat si informace o síti automaticky zjistit pomocí příslušných programů. Ty mohou, v závislosti na možnostech daného programu, být pouze ve formátu textového výstupu a grafickou podobou si následně musíme sami vytvořit. Pokročilejší programy za nás dokáží udělat vše, včetně grafického ztvárnění vybrané sítě. V následujících kapitolách budou rozebrány právě metody pro jejich automatickou tvorbu. Ty se dělí na několik základních kategorií.

6.2. Získávání informací o topologii

Jak bylo uvedeno výše, k získání dostatku informací o síti je nejprve nutné provést její analýzu. Tyto aktivity se v angličtině nazývají Network Discovery nebo Network Scan K tomu je možné využít řadu metod. Základní rozdělení je na aktivní a pasivní.

- **Aktivní** – Za aktivní jsou označovány metody, kdy je zařízení, z kterého provádíme průzkum, zapojeno do dané sítě a provádí průzkum pomocí vysílaní požadavků na odpověď ostatním zařízením v síti. Aktivní proto, že monitorovací zařízení vytváří dotazováním ostatních zařízení dodatečný provoz (aktivitu) na síti. Dotazování je možné provádět v mnoha krocích - od prozkoumání lokálního segmentu, na linkové vrstvě, až po služby aplikační vrstvy na vzdálených zařízeních. Zde se, kromě jiných, uplatní protokoly a metody pro diagnostiku sítí uvedené předchozích kapitolách.
- **Pasivní** – Při této metodě je sledovací zařízení umístěno do páteře sítě nebo je přímo jedním z páteřních prvků (směrovače, přepínače atd.). Na něm je prováděno zachytávaní síťového provozu a následně prováděna jeho analýza, která nám prozradí řadu potřebných, jinak nezjistitelných, informací. Pasivní proto, že monitorovací zařízení provádí pouze pasivní odposlech a vůči síti se tedy chová pasivně. Tato metoda též v případě delšího odposlechu odhaluje zařízení, která se objevují v síti pouze v nepravidelných intervalech a aktivní metodou by tak byla odhalena pouze malá část z nich.

6.3. Metody sběru informací

Při postupu získávání informací o síti je základem využití protokolů, které nám architektura dané sítě nabízí. Současně s faktem, že jsou tyto protokoly v modelech ISO/OSI a TCP/IP tříděny do řady vrstev, se nabízí toto třídění též využít.

6.3.1. Linková vrstva

Na linkové vrstvě probíhá výměna informací mezi zařízeními na stejném segmentu sítě, propojené zařízeními druhé vrstvy (huby, přepínače, opakovače). Na této vrstvě je používána adresace pomocí MAC adres. V tomto segmentu můžeme tedy provádět vyhledávání ostatních zařízení nezávisle na protokolech vyšších vrstev OSI modelu. K tomu můžeme využít několik protokolů a metod této vrstvy.

- **LLDP** – Protokol s názvem „Link Layer Discovery Protocol“, přijatý jako otevřený standard IEEE 802.1AB, sloužící k vyhledávání a vzájemné výměně informací mezi zařízeními na úrovni linkové vrstvy. Umožňuje výměnu informací jako jsou název zařízení, jeho konfigurace, informace o portu (a jeho konfiguraci), kterým je připojeno do sítě, informace pro protokoly vyšších vrstev (například adresy) a další. Vzhledem k otevřenosti standardu mohou výrobci vytvářet vlastní nové formáty zpráv, které mohou přenášet další informace požívané daným druhem zařízení. Jedno z mnoha rozšíření tak například umožňuje výměnu informací o konfiguraci VLAN sítě.
- **CDP** – Protokol CDP, podobně jako předchozí LLDP, je protokolem 2.vrstvy, sloužící k výměně informací o konfiguraci zařízení na lokální síti. Na rozdíl od předchozího je proprietárním řešením od firmy Cisco. Původně byl tedy implementován v zařízeních této firmy, postupně však jeho podporu implementují i další výrobci. Pracuje pomocí protokolu SNMP s vlastní kolekcí CDP MIB identifikátorů.
- **ARP a RARP** – Protokoly, jenž slouží k překladu adres 3. síťové vrstvy (IP) na adresy vrstvy linkové (MAC). Viz. Kapitola 3.2.
- **Arping** – Podobně jako Ping, používaný na třetí vrstvě, slouží k ověření dostupnosti síťových zařízení. Na rozdíl od něj však používá protokol ARP, jenž využívá broadcast na linkové vrstvě, takže i toto zjišťování dostupnosti je možné pouze na lokálním segmentu sítě. Výhodou proti tradičnímu Pingu je že tedy nebývá blokován firewally jako při použití protokolu ICMP.

- **ArpWatch –** Metoda, kdy jsou, v delším časovém úseku, zachytávány změny v mezipaměti *Arp cache*, jenž uchovává přeložené páry adres MAC-IP.

6.3.2. Síťová vrstva

Na síťové vrstvě spolu komunikují zařízení umístěná ve vzdálených segmentech sítě, propojených pomocí směrovačů (routerů).

- **Ping** – Slouží k zjištění dostupnosti zařízení na síťové vrstvě (IP). Viz. Kapitola 3.1.1. To nám poslouží k základnímu zjištění dostupnosti stanic, které poté podrobíme další analýze.
- **Traceroute** – Traceroute zjistí cestu od sledovacího zařízení ke sledovanému, přes jednotlivé směrovače. To nám může pomoci k zjištění kde se zařízení s danou adresou nachází (za kterými směrovači). Spárování a analýza jednotlivých cest nám poté pomůže při tvorbě propojení jednotlivých zařízení. Omezením této metody je fakt, že nám ve výsledku ukáže pouze cesty závislé na nastavení směrovaní jednotlivých směrovačů směrem od sledovaného zařízení. Nemusí tedy odhalit všechny cesty. Nejvíce se toto projeví při použití dynamických směrovacích protokolů na dané síti.
- **Směrovací údaje** – Získáním údajů ze směrovacích tabulek a interních databází různých směrovacích protokolů (služeb), případně odchytáváním zpráv směrovacích protokolů, získáme informace o umístění jednotlivých adresních rozsahů v síti. Zároveň získáme informaci umístění adres příslušných rozhraní směrovačů, což nám pomůže k odhalení špatně interpretovaných uzlů sítě.

6.3.3. Aplikační a Transportní vrstva

Vrstvy umístěné výše od vrstvy síťové nám umožňují komunikaci konkrétních aplikací na vzdálených zařízeních. Pro odlišení jednotlivých aplikací a jejich protokolů se používá, na transportní vrstvě, rozdělení na takzvané TCP nebo UDP porty. Každá služba má přiděleno vlastní číslo portu, na kterém následně naslouchá. Toho můžeme využít při zjišťování dostupnosti uzlů sítě. To se provádí pomocí takzvaného skenování portů, kdy je na každém z portů odeslán požadavek a čeká se zda-li přijde odpověď. Většinou se, pro zvýšení rychlosti, používá dotazování pouze na nejznámější služby (HTTP, FTP, NFS, SMB a pod.). Pokud alespoň některá služba odpoví, je na dané síťové adrese dostupné nějaké zařízení. Toho může být využito například pokud je na daném zařízení zakázáno odpovídaní

na ICMP požadavky, ovšem tato metoda může být odhalena a následně blokována chytřejšími firewally.

6.3.4. Čtení konfigurace

Další možností pro sběr dat je čtení konfigurace jednotlivých zařízení. To nám umožní získání nejvíce potřebných dat. Získáme tak přístup například k detailnějším informacím o hardwaru daného zařízení, softwaru, jednotlivých komunikačních rozhraních a jejich konfiguraci, k informacím ze směrovacích protokolů a podobně, což nám umožní tvorbu podstatně přesnějšího a detailnějšího diagramu sítě. Tento případ je ideálním řešením, ovšem má několik nástrah. Čtení konfiguračních údajů je možné provádět více způsoby.

Ruční získávání dat – Jestliže některá z použitých zařízení nepodporují protokoly pro hromadnou, standardizovanou správu, musíme použít pro vstup do konfigurace protokoly jako SSH nebo Telnet a jejich příkazovou řádku. Tento způsob je nevýhodný a zdlouhavý, jelikož musíme pro každý druh zařízení znát jeho strukturu příkazů a mít přístup do konfigurace, což také omezuje jeho použití v univerzálních automatických programech. Na druhou stranu s ním však leckdy získáme více informací než s univerzálními protokoly.

Nejlepší možností je použití standardizovaných protokolů, k tomu přímo určených, jako je SNMP a další. Ty nám umožní čtení informací v jednotném formátu a možnost čtení některých základních informací i bez přístupových práv do konfigurace.

6.4. Nástrahy při získávání informací

Některé postupy při získávání informací mají své záludnosti, které mohou způsobit vytvoření diagramu rozdílného od skutečné fyzické topologie. Proto je na ně nutné dbát zřetel.

- Zařízení mohou mít více fyzických rozhraní, které mají vlastní adresu a tyto adresy jsou obvykle zároveň z jiných adresních rozsahů. Stejně tak může být na jednom rozhraní namapováno více adres. Při použití jednodušších metod, například těch které využívají Ping nebo Traceroute, může toto způsobit situaci, kdy je skutečné fyzické zařízení s více rozhraními v následné ilustraci rozděleno na několik rozdílných zařízení. Proto je výhodnější použít výkonnějších protokolů, jako SNMP, jenž nám umožní spárování více informací o celém zařízení.

- Používání virtuálních sítí VLAN nám může zakrýt skutečnou topologii. Při metodách jako je Traceroute se pak nemusí zobrazit všechny mezilehlé prvky, jenž pak nejsou zahrnuty do výsledku nebo nám zcela mění obraz topologie. Zároveň, při větším počtu VLAN sítí na jednom zařízení, nebo jeho rozhraních, vytváří ve výsledku větší počet linek než skutečné.
- Virtualizace – V dnešní době velký fenomén virtualizace operačních systémů a serverů též napomáhá zakrytí fyzické architektury. Na jednom fyzickém zařízení tak může být mnoho virtuálních síťových rozhraní s různými adresami a příslušností k různým segmentům (a rozsahům) sítě. Kombinuje v sobě tedy předchozí problémy, ovšem v početnější míře.
- Některá zařízení mohou být, většinou z bezpečnostních důvodů, nastavena tak, aby neodpovídala na žádná příchozí (jím nevyžádaná) spojení. V případě aktivních metod dotazování tedy nemusíme získat žádnou informaci o jejich existenci. Pro tento případ je výhodnější nasazení pasivních metod odposlechu, jenž nám zobrazí i odchozí spojení a tím odhalí jejich existenci. Informace o konkrétním umístění zařízení v síti je pak zjištěno pomocí dalších metod.
- Metody překladu adres (NAT) též zakrývají skutečnou topologii a mohou způsobovat problémy při analýze dat z pasivních metod.
- Přesměrování portů transportních vrstev (portforwarding a portrelaying) též souvisí s problémem překladu adres. Pokud použijeme při průzkumu aplikační protokoly a jejich odezvy na příslušných TCP/UDP portech, může nastat situace, kdy jsou některé porty přesměrovávány na jiný stroj nebo adresu, my to však nemusíme poznat. To může opět ve výsledku vést ke špatné interpretaci zjištěných informací.
- Aplikační proxy – podobně jako NAT mění obsah procházejících paketů aplikačních protokolů, čímž zakrývá skutečnou topologii sítě.

Mobilní zařízení se mohou vyskytovat pokaždé v jiné části sítě, což též ovlivní výsledný obraz topologie.

6.5. Grafické vyjádření

Nasbíraná data o síti můžeme použít jako vstup dalších programů, pro člověka je však vhodnější jejich grafické ztvárnění. V případě druhé možnosti se pro počítačové sítě nejčastěji používá stromu nebo 2D grafu, kde uzly reprezentují jednotlivá zařízení a hrany jejich vzájemná propojení. Pro rozvržení prvků v grafu existuje celá řada jejich typů. Při ruční tvorbě diagramu sítě provádíme toto rozložení a propojení jednotlivých prvků intuitivně. Jestliže však má být diagram generován automaticky, musíme použít některý z algoritmů pro tvorbu grafů, jenž ovlivní též jeho výslednou grafickou podobu. Jednotlivé algoritmy se liší jak výsledným grafickým ztvárněním, tak složitostí a časovou náročností jeho tvorby. Výzkumná organizace *NATO Research and Technology Organisation* provedla výzkum [5] několika tradičních algoritmů vzhledem k tomu, nakolik se hodí pro generování diagramu počítačových sítí. Ve výsledku bylo zjištěno, že tradiční metody nejsou pro sítě, jenž jsou rozlehlejší a mají větší počet vzájemných propojení, příliš vhodné. Některé byly sice rychlejší, ovšem grafické ztvárnění příliš neodpovídalo realitě a bylo nepřehledné nebo naopak vizualizace byla uspokojivá ovšem rychlost a náročnost algoritmu příliš vysoká. Z tohoto důvodu vyvinuli dva nové algoritmy přímo pro tyto druhy sítí, jenž se snaží eliminovat oba předchozí problémy.

7. Softwary pro vizualizaci sítě

Programů pro tvorbu diagramů sítí je celá řada. Některé provádějí pouze sběr informací o sítích, další naopak pouze jejich grafické ztvárnění (ať už ručně či automaticky) a naposledy ty jenž zvládají obě tyto činnosti. Opět je možnost volby mezi volnými open-source a komerčními produkty. V tomto odvětví však platí, stejně jako jinde, fakt že volné programy jsou sice použitelné, ovšem jejich komerční protivníci často nabízejí lepší propracovanost a případně více funkcí. Volné programy jsou tedy použitelné spíše v menších sítích, komerční pak v těch větších, kde prvotní vynaložené prostředky jsou zanedbatelné vzhledem k jejich následnému přínosu a úspoře. Kromě specializovaných nástrojů, umožňují zjišťování nebo grafické zobrazení topologie i některé z dříve uvedených softwarů pro kompletní diagnostiku sítě (většinou jako moduly). Proto už dále uvedu pouze ty specializované.

7.1. Nmap

Nmap ("Network Mapper") je silný nástroj pro kompletní skenování sítě. Umožňuje sběr informací o jednotlivých zařízeních a jejich kompletní analýzu, jenž zahrnuje identifikaci jeho operačního systému a síťových služeb na něm běžících. Síla tohoto nástroje je v tom, že obsahuje řadu zvláštních metod a postupů, jenž dokáží, proti standardním metodám, odhalit velkou řadu informací. Proto je využíván pro analýzu slabých míst v zabezpečení sítí, a to jak administrátory, při jejím zlepšování, tak hackery při jejich napadání. Program byl původně určen pro použití v příkazové řádce, dnes je však možné použít i jeho grafické nadstavby. Je volně dostupný pod licencí GNU-GPL a je též multiplatformní, takže funguje pod nejrozšířenějšími operačními systémy, jako jsou MS Windows, Linux, Mac OS a celou řadou dalších unixů.

Domovská stránka: <http://nmap.org/>

7.2. NeDi

NeDi je dalším volně dostupným nástrojem, jenž umožňuje sběr dat o zařízeních a jejich následné grafické vyjádření. Program je zaměřen spíše na zařízení firmy Cisco a další s nimi kompatibilní, proto ke sběru využívá primárně Cisco CDP nebo LLDP protokol a CLI rozhraní určené pro jejich konfiguraci. Kromě toho však podporuje i další protokoly jako SNMP. Jeho struktura je dělena na

tři části – část pro sběr dát (jenž je psána v programovacím jazyce Perl), dále část pro ukládání získaných dat (databáze MySQL) a k nim část poslední, jenž umožňuje zobrazení těchto dat pro uživatele pomocí webového rozhraní (použit jazyk PHP). Výhoda tohoto řešení je právě v jeho modulárnosti, kdy je možné jednotlivé části měnit a upravovat dle potřeb, případně jeho část použít jako modul v jiném systému.

Domovská stránka: <http://www.nedi.ch/>

7.3. LANView

LANView je komerčním nástrojem pro analýzu sítě, určeným pro grafické prostředí systémů MS Windows. Pro sběr informací používá jak aktivních, tak pasivních metod – ping, traceroute, skenování portů, odposlech a analýza procházejících dat nebo pomocí protokolu SNMP. Některá nasbíraná data umožňuje zobrazit v grafech, případně je exportovat do různých formátů pro další použití.

Domovská stránka: <http://www.jxdev.com/>

7.4. MS Visio

Visio je komerční nástroj od společnosti Microsoft, určený pro tvorbu různých druhů diagramů. Pro potřeby datových sítí umožňuje jak ruční tvorbu diagramů jejich topologií, tak jejich automatické skenování a následnou tvorbu tohoto diagramu. Program je určen pro systémy MS Windows.

Domovská stránka: <http://www.microsoft.com/cze/office/programs/visio/>

7.5. DIA

DIA je podobně jako MS Visio univerzálním nástrojem pro tvorbu různých diagramů, na rozdíl od něj je však volně dostupný a multiplatformní. Program používá pro definici grafických prvků a ukládání dat otevřený jazyk XML, což umožňuje jeho snadnou rozšiřitelnost a přenos dat. Dále obsahuje systém pro přidávání vlastních zásuvných modulů, jenž umožňuje další rozšiřování jeho funkcí, například o export do jiných formátů nebo rozhraní pro skripty jazyka Python. Program je šířen pod licencí GNU-GPL.

Domovská stránka: <http://projects.gnome.org/dia/>

7.6. LANsurveyor

Nástroj LANsurveyor od společnosti SolarWinds (Neon Software) je nástrojem pro detekování topologie sítě pomocí SNMP. Po proskenování určeného rozsahu program zobrazí podrobné výsledky, které jsou řazeny do kategorií a ze získaných dat též umí vytvořit diagram sítě, který je možné exportovat do formátu pro MS Visio. Rozpoznaná zařízení umožňuje nadále automaticky monitorovat a případně hlásit potíže. Program je určen pro MS Windows.

Domovská stránka: <http://www.solarwinds.com/products/lansurveyor/>

7.7. Network View

Network View je další z komerčních programů, které umožňují jak sběr dat o síti, tak následné grafické zobrazení topologie. Prohledávání provádí aktivně pomocí protokolů SNMP, NetBios, WMI, DNS a skenováním portů. Po prohledání sítě umí následně sledovat a logovat chod jednotlivých zařízení, případně zasílat oznámení o vzniklých chybách. Program je funkční na operačních systémech MS Windows.

Domovská stránka: <http://www.networkview.com/>

7.8. Insightix Discovery

Insightix Discovery je komerční sada nástrojů, jenž umožňuje provádět analýzu velkých sítí pomocí kombinace aktivních i pasivních metod, následnou diagnostiku a správu jednotlivých zařízení, pomáhat zlepšovaní zabezpečení sítě a případně o provádět potřebnou dokumentaci. Zajímavostí tohoto systému jsou některé pokročilé funkce, jenž výrobce deklaruje – software by měl dokázat odhalit i zařízení skrytá za firewally nebo NATy a detekovat virtuální servery (VMWare). Lite verze tohoto produktu je dostupná pro operační systémy MS Windows, rozšířená Enteprise pak vyžaduje vlastní Linuxový stroj.

Domovská stránka: <http://www.insightix.com/>

7.9. Ipsonar

Software společnosti Lumeta je konkurenčním produktem k předchozímu a nabízí tedy obdobnou funkcionalitu pro velké sítě.

Domovská stránka: <http://www.lumeta.com/ipsonar/>

8. Vlastní aplikace

Zadáním práce bylo vytvoření nástroje pro generování diagramu počítačových sítí, který půjde dále zpracovávat v programu Dia. Pro vyřešení uvedeného problému se nabízejí dvě základní možnosti:

První je použití externího programu (ať už vlastního nebo cizího), jenž by umožňoval výstup do formátu pro aplikaci DIA. Tato možnost je výhodná v tom, že můžeme použít aplikaci, jenž máme oblíbenu nebo která nabízí větší funkčnost než-li druhé řešení.

Druhou je vytvoření vlastního modulu přímo do aplikace DIA. Tato možnost je výhodnější v tom, že nám přináší požadované funkce přímo v aplikaci, bez nutnosti instalace a použití dalších programů. Nevýhodou je předpokládaný menší počet funkcí proti specializovaným řešením (aplikacím).

Aplikace DIA je multiplatformní a je psána v programovacím jazyce C, tedy pro každou platformu (operační systém) musí být zvlášť upravena a provedeno přeložení do binární podoby. Kromě využití vlastních funkcí, nabízí možnost rozšiřování své funkčnosti pomocí aplikačního rozhraní a zásuvných modulů – pluginů. Tyto moduly, však musí být, stejně jako samotný program, kompilovány pro konkrétní platformy, což omezuje jejich přenositelnost. K aplikaci byl však vyvinut modul, jenž vytváří aplikační rozhraní pro jazyk Python. Python je dynamický, interpretovaný programovací (skriptovací) jazyk. Pro každý operační systém tak musí být v nativní podobě vytvořen pouze interpret, jenž poté poskytuje jednotné rozhraní mezi systémem a skripty. To umožňuje vytvářet pro všechny systémy doplňky se stejnými zdrojovými kódy. Nevýhodou tohoto řešení je však větší hardwarová náročnost na systém a menší rychlost zpracování aplikací proti nativním řešením.

Z výše uvedených možností jsem si pro vlastní řešení vybral jejich kombinaci, avšak s větším podílem verze vlastního interního modulu, psaného v jazyce Python. Samotný vývoj jsem prováděl na operačním systému Microsoft Windows, což ovšem v jeho průběhu přineslo některá zásadní omezení. Modul „PyDia”, jenž v základní aplikaci vytváří rozhraní pro Python, je z důvodů omezení kompilátoru, v němž je pro MS Windows vytvářen, kompilován pouze s verzí jazyka

Python 2.3, která je již relativně zastaralá. Toto omezení znemožnilo použití řady hotových skriptů, jenž by mohly být použity pro lepší funkčnost.

Aplikace je rozdělena na dvě základní části – část pro sběr dat o síti a část vykreslovací. Toto rozdělení kombinuje výhody výše uvedených možností tím, že skenovací část může být oddělena a použita jako samostatná aplikace, nezávisle na programu DIA nebo může být naopak pro DIA nahrazena zcela jiným skenovacím modulem.

8.1. Skenovací část

Skenovací část aplikace, jenž provádí prohledávání sítě a meziukládání získaných dat, používá pro svou činnost aktivních metod prohledávání. Vstupními daty je rozsah IP adres jenž má být prohledán. Aplikace vezme vždy jednu adresu z daného rozsahu a provede sérii pokusů na zjištění odezvy pomocí metody Ping, konkrétně pomocí ICMP echo zpráv. Jestliže na dané adrese nějaké zařízení odpoví, je následně provedeno testování, přes které směrovače vede cesta k této adrese, pomocí metody traceroute (zde je použita metoda s protokolem ICMP a zprávami ICMP_TIMEXCEED). Z této trasy je vybrán poslední směrovač, za nímž je testovaná adresa, a jejich adresy jsou uloženy do mezipaměti. Po proskenování celého zadaného rozsahu se ke každé nalezené adrese provede pokus o zjištění jejich doménového jména, jenž je případě doplněno k předchozím informacím.

Vzhledem k tomu, že Python v základu nemá příliš velkou podporu pro zjednodušení manipulace s pakety síťových protokolů, byly pro to použity externí skripty různých autorů.

Pro zjednodušení manipulace se zadaným vstupním rozsahem IP adres je použita sada skriptů se jménem „Ipy“, jenž je inspirována modulem „Net::IP“ z programovacího jazyka Perl. Ta dokáže provádět kontrolu a převod mezi různými druhy zápisů adresních rozsahů (192.168.1.1-192.168.1.255, 192.168.1.0/24, apod.). Pomocí nich je vytvořeno pole IP adres odpovídající zadanému rozsahu, jenž je poté použito jako vstup pro uvedenou skenovací funkci. Skript je šířen pod licencí umožňující jeho použití a její kopie je dostupná v příslušném podadresáři aplikace. Tato sada skriptů je dostupná online na adrese: <http://pypi.python.org/pypi/IPy/>

Dalším použitým cizím skriptem je „Ping“, jenž je inspirován knihovnou ping.c v Linuxu. Ten implementuje ICMP_ECHO - Ping přímo pomocí raw soketů, což umožňuje, na rozdíl od jiných implementací, přímou manipulaci s jednotlivými

bity posílaných a přijímaných zpráv. Část kódu byla upravena, část použita jako modul. Skript je šířen pod licencí GNU-GPL a je dostupný online na adrese: <ftp://ftp.visi.com/users/mdc/ping.py>

Pro zjednodušení implementace metody „traceroute“ byla použita sada skriptů se jménem „Impacket“ od společnosti CORE Security Technologies' . Ta umožňuje manipulaci s pakety řady protokolů. Zde byla jí bylo použito pro dekódování typu a kontrolu příchozích ICMP paketů. Tato sada umožňuje šíření pod licencí „Apache Software License“, jenž je dostupná v příslušném podadresáři aplikace. Skript je dostupný online na adrese: <http://oss.coresecurity.com/projects/impacket.html>

8.2. Grafická část

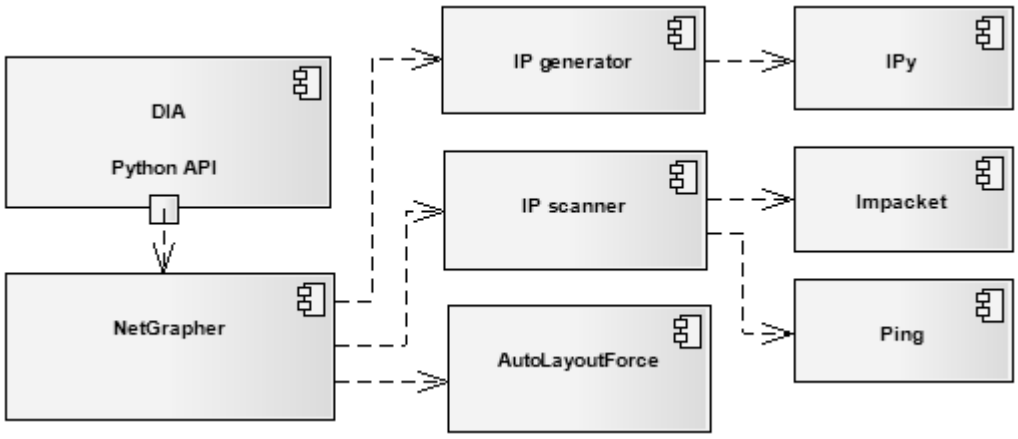
Druhá část zprostředkovává propojení a grafické zobrazení dat pro aplikaci DIA. DIA nabízí aplikační rozhraní (API) pro manipulaci s objekty na kreslící ploše, přidávání vlastních funkcí do aplikace s možností využití grafického systému GTK a pro vlastní moduly určené pro import a export do jiných formátů. Tento modul jej využívá pro zobrazení dialogového okna, v němž jsou zadávány vstupní údaje pro skener. To je předá k provedení analýzy sítě a následně jsou vrácena data a předána funkci pro grafické vykreslení. Ta provede vsazení jednotlivých grafických prvků na kreslící plochu a vyvolá další skript pro jejich rozmístění na ploše.

Pro uvedené rozmisťování prvků je použit další externí skript, jenž je již distribuován se samotnou aplikací DIA. Ten se nazývá „autolayoutforce“ a provádí rozmisťování pomocí algoritmu "force based autolayout" z teorie grafů. Jeho použití bylo zvoleno pro jeho uspokojivé výsledky. Nevýhodou je relativně dlouhá doba jeho práce a vytížení systému. Před jeho použitím byla provedena analýza možnosti použití jiných, hotových, algoritmů pro tvorbu grafů, ovšem tam jsem narazil na výše uvedené omezení, že byly implementovány pro vyšší verze jazyka Python. Jinou možností by bylo použití externí rozmísťovací aplikace jako je třeba GraphViz[1]. Toto řešení ovšem nepřináší dostatek výhod a o mnoho lepší výsledky proti použití tohoto interního skriptu.

1 GraphViz - nástroj pro automatizovanou tvorbu diagramů. Domovská stránka, online: <http://www.graphviz.org/>

8.3. Propojení jednotlivých částí aplikace

Jednotlivé části jsou propojeny a vzájemně importovány jako moduly podle schématu na obrázku č.6.



Obrázek 6 - Diagram propojení modulů

8.4. Princip funkce

Po spuštění DIA je do něj skript načten. Do textové nabídky v okně diagramu je přidána položka „IPscanner“ do podnabídky „Dialogy“. Po jejím vybrání je zobrazeno dialogové okno, jenž obsahuje pole pro zadání adresního rozsahu a potvrzovací tlačítko. Po zadání rozsahu a stisknutí tlačítka je zadaný rozsah předán jako textový řetězec IP generátoru, jenž provede kontrolu správnosti zadaného rozsahu. Pokud je rozsah zadán správně, vygeneruje a vrátí pole IP adres, jenž mu odpovídají. Pokud je nesprávný, je vyvolána výjimka a uživatel je o tom informován pomocí nového informačního dialogového okna a zpracování se zastaví až do nového zadání rozsahu. Vygenerované pole adres je předáno Scanneru, jenž provede proskenování daného rozsahu a vrátí tři související mapy(obdoba pole) s nalezenými informacemi(první hodnota – klíč, druhá – data):

- [index_bodu]:index_nadřazeného_routeru
- [IP_adresa]:index_bodu
- [IP_adresa]:doménové_jméno.

Tyto data jsou předány algoritmu, jenž podle nich provede nasázení a propojení jednotlivých prvků do diagramu (kreslící plocha). Tam je použita tato logika: Pokud je index_bodu u některého z ostatních bodů jako hodnota index_nadřazeného_routeru, je brán jako router a je mu přiřazen obrázek směrovače.

Pokud není, je brán jako koncové zařízení a je mu přiřazen obrázek počítače. Pro lepší přehlednost jsou koncová zařízení příslušející k danému směrovači spojena nejprve s obrázkem „oblak sítě“ a ten teprve s jejich směrovačem. Nakonec je zavolán modul AutoLayoutForce, jenž provede výsledné rozmístění těchto prvků.

8.5. Test funkčnosti

Výsledná aplikace byla testována na části reálné sítě. Na obrázcích č.7 a č.8 je ukázka rozdílů vygenerovaného a skutečného diagramu z provedeného testu. Jak je z výsledku patrné, při metodě pomocí traceroutu jsou zachyceny a správně identifikovány pouze adresy síťových rozhraní jednotlivých směrovačů, které jsou směrem od sledovacího zařízení.

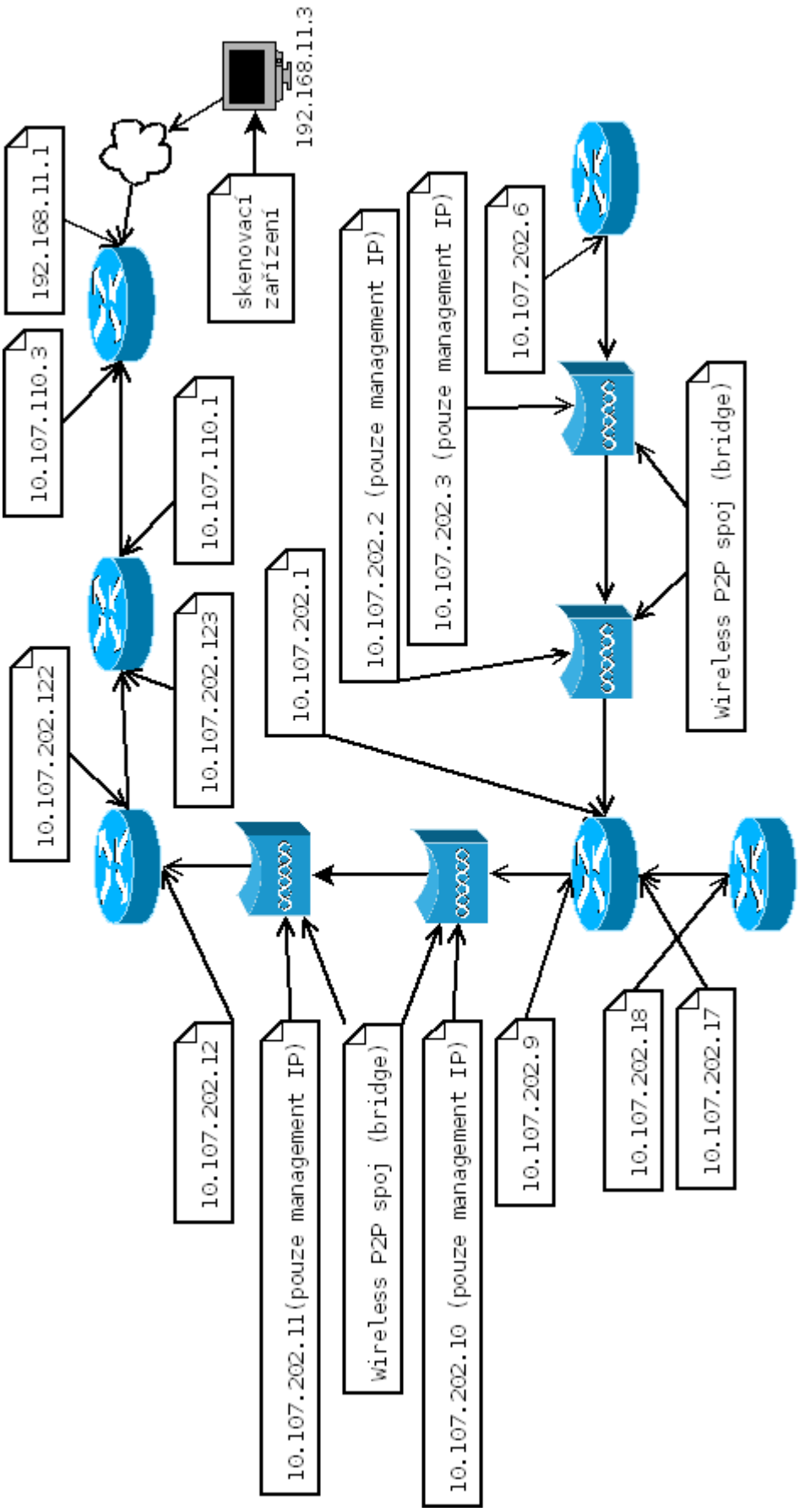
8.6. Možnosti dalšího vývoje

Použití zastaralé verze jazyka Python přinášelo v průběhu vývoje řadu omezení a v řadě případů muselo být provedeno zdlouhavé hledání řešení jednotlivých problémů právě pro tuto verzi. Ideálním řešením by byla možnost, kdy by si uživatel mohl v uvedeném dialogovém okně zvolit, zda-li chce použít interní metody nebo data získávat pomocí externích aplikací. Pro uživatele operačních systémů, kde je DIA možné použít i s novější verzí Pythonu, by tak mohly být použity některé skripty, jenž implementují pokročilejší metody vyhledávání (například PluTo[2]) a uživatelé na Windows by měli pouze možnost použití starší verze nebo externích aplikací. Vývoj takového řešení je však náročný na znalosti pokročilejších programovacích technik a tomu odpovídající čas.

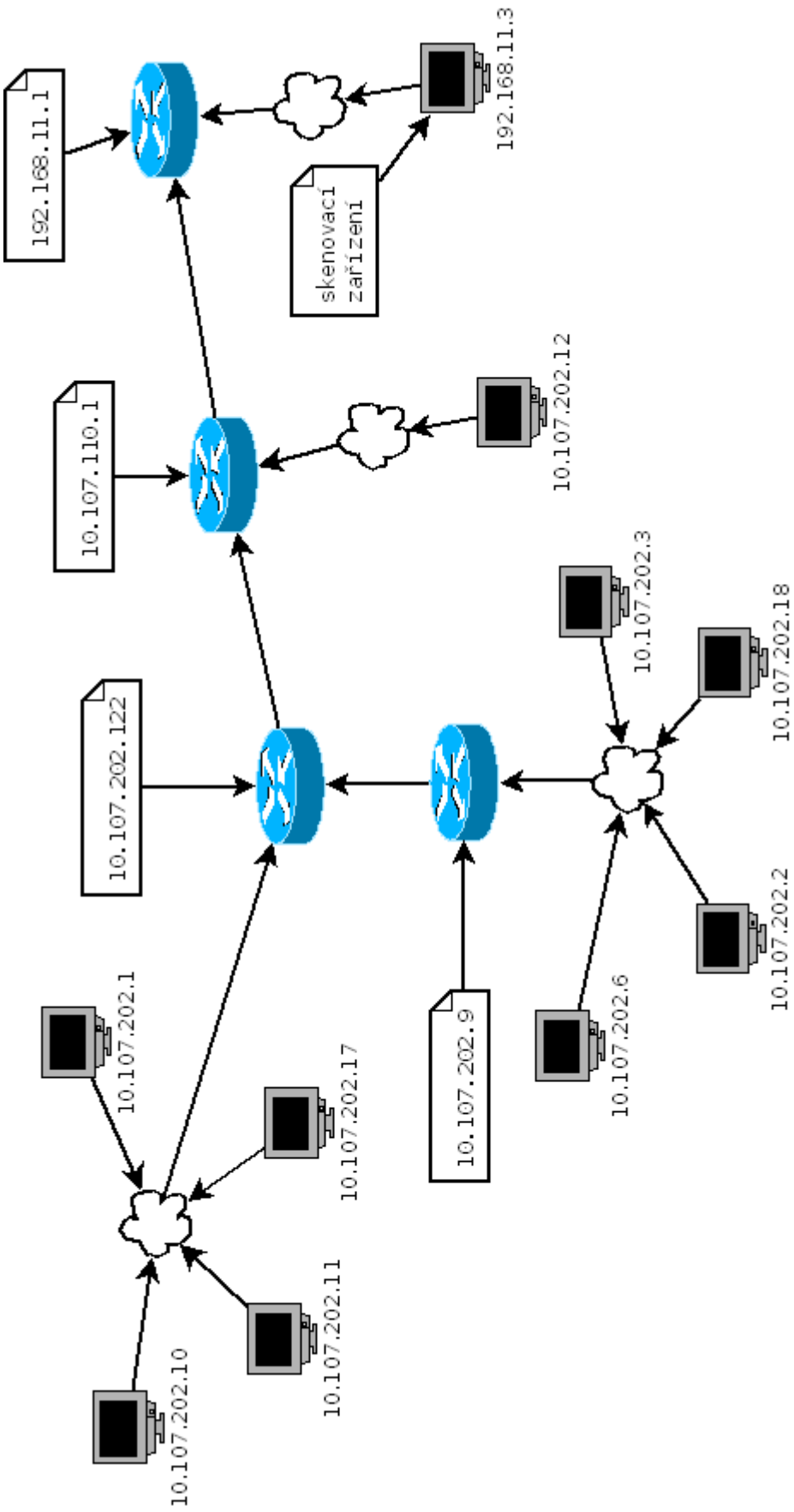
8.7. Způsob používání aplikace

Uživatelská příručka, popisující způsob instalaci a používání aplikace je uvedena v příloze A.

2 PluTo – nástroj pro skenování sítě, psaný v jazyce Python. Domovská stránka, online: <http://rainbow.cs.unipi.gr/projects/pluto>



Obrázek 7 - Příklad - diagram reálné topologie



Obrázek 8 - Příklad - vygenerovaný diagram

9. Závěr

V teoretické části práce jsem se snažil shrnout protokoly, metody a s nimi související aplikace, které nám mohou být užitečné při návrhu aplikace, jenž má automaticky provádět co nejlepší popis (a grafické vyjádření) vybrané sítě.

V praktické části bylo cílem vytvořit aplikaci, která by těchto metod využívala a výsledek byla schopna zobrazit v aplikaci DIA. Původní myšlenkou bylo vytvoření jednoduchého skriptu v Pythonu pro aplikační rozhraní programu DIA. Postupné studium jednotlivých metod prohledávání sítě však ukázalo rozsáhlost daného problému. Proto byla aplikace rozložena na dva samostatné moduly, tak aby mohly být jednotlivé části postupně nahrazovány propracovanějšími verzemi.

Ke zvolení vhodné metody konečného grafického zobrazení dané sítě, která by podávala co nejvěrnější rozřazení prvků na kreslící ploše muselo být nastudováno řada algoritmů z teorie grafů. Zajímavostí při tom bylo zjištění, že většina menších komerčních aplikací implementuje pouze jednoduché grafické metody, jenž nevytváří příliš věrné zobrazení. Pro další vývoj by bylo zajímavé se, pro toto rozmisťování, pokusit do aplikace implementovat nové algoritmy, jenž v NATO vyvinuli přímo pro počítačové sítě.

Dále by bylo vhodné, pro vyhledávací část, umožnit uživateli zvolit vlastní kombinaci různých metod, tak aby bylo dosaženo co nejlepšího výsledku pro různé případy.

Použitá literatura a ostatní zdroje

[1] PETERKA, Jiří. Co je čím v počítačových sítích [online]. 1991-1997, Dostupný z WWW: <http://www.earchiv.cz/i_coje.php3>

[2] DOSTÁLEK, L.; KABELOVÁ, A.. Velký průvodce protokoly TCP/IP a systémem DNS. 2.vyd. Praha: ComputerPress, 2000. 435 s. ISBN 80-7226-323-4

[3] KOZIEROK, M., Charles. The TCP/IP Guide, Version 3.0 [online]. 2001-2005, 20.8.2005, Dostupný z WWW: <http://www.tcpipguide.com/free/>

[4] DELCROIX, M.; LUMETTA, O.. SNMP Tutorial [online], Dostupný z WWW: <http://www.et.put.poznan.pl/snmp/>

[5] VANDENBERGHE, G; TREURNIET, J.. RTO-MP-IST-063-01: Automating the Presentation of Computer Networks [online]. Ottawa: NATO Research and Technology Organisation, 2006, 18 s., ISBN 92-837-1156-4 / 978-92-837-1156-8, Dostupný z WWW: <http://ftp.rta.nato.int/public//PubFullText/RTO/MP/RTO-MP-IST-063///MP-IST-063-01.pdf>

[6] KLAŠKA, Luboš. Správa počítačových sítí [online]. 2000, Dostupný z WWW: <http://www.svetsiti.cz/view_list.asp?rubrika=Tutorialy&temaID=23>

[7] Přehled služby WMI [online]. 2009, Dostupný z WWW: <http://technet.microsoft.com/cs-cz/library/cc753534(WS.10).aspx>

[8] Desktop Management Interface (DMI) Standards [online]. 2009, Dostupný z WWW: <http://www.dmtf.org/standards/dmi/>

[9] Web-Based Enterprise Management (WBEM) [online].2009, Dostupný z WWW: <http://www.dmtf.org/standards/wbem/>

[10] SLOAN, D., Joseph. O'Reilly : Network Troubleshooting Tools. O'Reilly Media, 2001, 364 s., ISBN 0-596-00186-X, Dostupný z WWW: <http://docstore.mik.ua/orelly/networking_2ndEd/tshoot/>

[11] MAURO, R.; Schmidt, J.. O'Reilly : Essential SNMP. O'Reilly Media, 2001, 330 s., ISBN 0-596-00020-0, Dostupný z WWW: <http://www.hell.org.ua/Docs/oreilly/tcpip2/snmp/>

[12] SNMP Portal : SNMP - The Simple Network Management Protocol [online]. RAD Data Communications, Ltd., 2009, Dostupný z WWW: <http://www3.rad.co.il/networks//applications/snmp/main.htm>

[13] KABELÁČ, Zdeněk. Správa sítě routerů: diplomová práce. Brno: Masarykova univerzita, Fakulta Informatiky, 1997, 42 s., 5 l. příl., Vedoucí diplomové práce Ing. Jiří Novotný, Dostupný z WWW: <http://is.muni.cz/th/759/fi_m/>

Použité normy

(1) POSTEL, J.: RFC 792: INTERNET CONTROL MESSAGE PROTOCOL, 1981, Dostupný z WWW: <http://tools.ietf.org/html/rfc792>

(2) CASE, J.: RFC 1157: A Simple Network Management Protocol (SNMP), 1990, Dostupný z WWW: <http://tools.ietf.org/html/rfc1157>

(3) ROSE, M; MCCLOGHRIE, K.: RFC 1155: Structure and Identification of Management Information for TCP/IP-based Internets, 1990, Dostupný z WWW: <http://tools.ietf.org/html/rfc1155>

(4) MCCLOGHRIE, K.: RFC 1213: Management Information Base for Network Management of TCP/IP-based internets: MIB-II., 1991, Dostupný z WWW: <http://tools.ietf.org/html/rfc1213>

(5) CASE, J.:RFC 1901: Introduction to Community-based SNMPv2, 1996, Dostupný z WWW: <http://tools.ietf.org/html/rfc1901>

(6) CASE, J.:RFC 1902: Structure of Management Information for Version 2 of the Simple Network Management Protocol (SNMPv2), 1996, Dostupný z WWW: <http://tools.ietf.org/html/rfc1902>

(7) CASE, J.:RFC 1903: Textual Conventions for Version 2 of the Simple Network Management Protocol (SNMPv2), 1996, Dostupný z WWW: <http://tools.ietf.org/html/rfc1903>

(8) CASE, J.:RFC 1904: Conformance Statements for Version 2 of the Simple Network Management Protocol (SNMPv2), 1996, Dostupný z WWW: <http://tools.ietf.org/html/rfc1904>

(9) CASE, J.:RFC 1905: Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2), 1996, Dostupný z WWW: <http://tools.ietf.org/html/rfc1905>

(10) CASE, J.:RFC 1906: Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2), 1996, Dostupný z WWW: <http://tools.ietf.org/html/rfc1906>

(11) CASE, J.:RFC 1907: Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2), 1996, Dostupný z WWW: <http://tools.ietf.org/html/rfc1907>

(12) CASE, J.:RFC 1908: Coexistence between Version 1 and Version 2 of the Internet-standard Network Management Framework, 1996, Dostupný z WWW: <http://tools.ietf.org/html/rfc1908>

(13) McCloghrie, K.: RFC 1909: An Administrative Infrastructure for SNMPv2, 1996, Dostupný z WWW: <http://tools.ietf.org/html/rfc1909>

(14) Waters, G.: RFC 191: User-based Security Model for SNMPv2, 1996, Dostupný z WWW: <http://tools.ietf.org/html/rfc1910>

(15) HARRINGTON, D.: RFC 2271: An Architecture for Describing SNMP Management Frameworks, 1998, Dostupný z WWW: <http://tools.ietf.org/html/rfc2271>

(16) CASE, J.:RFC 2272: Message Processing and Dispatching for the Simple Network Management Protocol (SNMP), 1998, Dostupný z WWW: <http://tools.ietf.org/html/rfc2272>

(17) LEVI, D.:RFC 2273: SNMPv3 Applications, 1998, Dostupný z WWW: <http://tools.ietf.org/html/rfc2273>

(18) BLUMENTHAL, U.: RFC 2274: User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3) 1998, Dostupný z WWW: <http://tools.ietf.org/html/rfc2274>

(19) WIJNEN, B.: RFC 2275: View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP), 1998, Dostupný z WWW: <http://tools.ietf.org/html/rfc2275>

(20) FRYE, R.; LEVI, D: RFC 3584: Coexistence between Version 1, Version 2, and Version 3 of the Internet-standard Network Management Framework, 2003, Dostupný z WWW: <http://tools.ietf.org/html/rfc3584>

(21) WALDBUSSER, S.:RFC 2819: Remote Network Monitoring Management Information Base, 2000, Dostupný z WWW: <http://tools.ietf.org/html/rfc2819>

(22) WALDBUSSER, S.:RFC 2021: Remote Network Monitoring Management Information Base Version 2 using SMIv2, 1997, Dostupný z WWW: <http://tools.ietf.org/html/rfc2021>

(23) WALDBUSSER, S.:RFC 4502: Remote Network Monitoring Management Information Base Version 2, 1997, Dostupný z WWW: <http://tools.ietf.org/html/rfc4502>

(24) WALDBUSSER, S.:RFC 3273: Remote Network Monitoring Management Information Base for High Capacity Networks, 1997, Dostupný z WWW: <http://tools.ietf.org/html/rfc3273>

Slovník pojmů

- **RFC** - zkratka anglického výrazu „Request for comments“, v překladu „žádost o komentáře“, jenž je používána pro označení standardů pro datové sítě. Název vznikl z principu jejich tvorby - nejprve je podán návrh k okomentování a následně přijat standard. Dokumenty jsou dostupné online například na adrese:<http://www.ietf.org/rfc.html>
- **ISO** – Mezinárodní organizace pro normalizaci, jenž je světovou federací národních normalizačních organizací. Organizace se zabývá tvorbou mezinárodních norem ISO. Dostupná online na adrese: <http://www.iso.org/>
- **IETF** – zkratka anglického výrazu „Internet Engineering Task Force“, jenž označuje mezinárodní organizaci, která vytváří standardy používané na Internetu. Dostupná online na adrese: <http://www.ietf.org/>
- **DMTF** - zkratka anglického výrazu „Distributed Management Task Force“, jenž označuje sdružení pro tvorbu standardů ohledně jednotné správy počítačových systémů. Dostupné online na adrese: <www.dmtf.org>
- **MD5** - zkratka anglického výrazu “Message Digest Algorithm“, která označuje sadu funkcí pro tvorbu kontrolních součtů z dat.
- **SHA** - zkratka anglického výrazu “Secure Hash Algorithm“, jenž je, podobně jako MD5, skupina funkcí a algoritmů pro tvorbu kontrolních součtů z dat.
- **DES** - zkratka anglického výrazu “Data Encryption Standard“. Ten označuje algoritmus pro symetrické šifrování dat.

PŘÍLOHA A – UŽIVATELSKÁ DOKUMENTACE APLIKACE

Obecný popis

Popisovaná aplikace je modulem do programu DIA, který je určen pro tvorbu různých diagramů. Tam slouží k proskenování vybrané sítě a následně vytvoření jejího diagramu(grafického vyjádření). Je napsána a optimalizována v jazyce Python verze 2.3.5, pro program DIA verze 0.97-pre3, pod operačním systémem Microsoft Windows XP SP2. Proto doporučuji použití této verze, vyloučena však není funkčnost i v jiných verzích. Její použití bude také proto popsáno pro MS Windows XP.

Instalace DIA

Domovskou webovou stránkou programu DIA je: <http://projects.gnome.org/dia/>, respektive v novější verzi <http://live.gnome.org/Dia>, kde je možné ji zdarma stáhnout a získat o ní více informací. Verzi pro MS Windows je možné získat z online adresy <http://dia-installer.de/>.

Vzhledem k faktu, že vytvořený modul je psán jako skript v jazyce Python, je potřeba jeho podporu zvolit již při instalaci DIA. Kromě toho je potřeba nainstalovat samotné prostředí Python (2.3.5) – dostupné na online adrese:

<http://python.org/ftp/python/2.3.5/Python-2.3.5.exe>

Dále jeho rozšíření PyCairo (1.0.2) - dostupné na online adrese:

<http://ftp.gnome.org/pub/GNOME/binaries/win32/pycairo/1.0/pycairo-1.0.2-1.win32-py2.3.exe>

A další rozšíření PyGtk (2.8.6) - dostupné na online adrese:

<http://ftp.gnome.org/pub/GNOME/binaries/win32/pygtk/2.8/pygtk-2.8.6-1.win32-py2.3.exe>

Volitelnou částí je instalace knihovny GTK, jenž je potřebná při testování vlastních skriptů přímo v prostředí Pythonu, mimo aplikaci DIA. Při testování této aplikace byla použita verze 2.2.1.1, jenž je dostupná online na adrese:

<http://sourceforge.net/projects/gtk-win/files/Legacy%20GTK%2B%20Runtime%20Env./GTK%2B-Runtime-Environment-2.2.1.1.exe/download>

Instalace skriptů

Instalace vlastních skriptů znamená pouze jejich nakopírování do příslušného adresáře a DIA si je poté samo načte při startu. Ty je možné umístit buď do adresáře, kde je DIA nainstalováno nebo do podadresáře \.dia\python v domovském adresáři uživatele. Doporučuji však verzi první, jelikož pokud je umístěn v domovském adresáři, může jej používat pouze daný uživatel. Kromě samotného skriptu je ve stejném adresáři potřeba mít umístěn skript „autolayoutforce.py“, jenž součástí standardní instalace DIA.

Použití skriptu

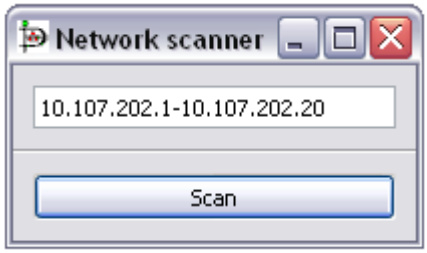
Popisovaný skript používá pro skenování sítě přímou manipulaci s pakety posílanými a přijímanými z/do ní. K tomu je však potřeba administrátorská oprávnění v operačním systému. DIA proto musí být spouštěn pod uživatelským účtem, jenž těmito oprávněními disponuje. Pokud běžně používáme účet s omezenými oprávněními, není nutné se odhlašovat – program je možné spustit jedním z následujících způsobů. Na ikonu programu klepneme pravým tlačítkem myši a zvolíme položku „Spustit jako“. Zde zadáme název účtu, jenž má potřebná oprávnění, jeho heslo a potvrdíme. Tím se program spustí pod tímto účtem a používá veškerá jeho nastavení (a soukromé adresáře). Stejného výsledku je možné dosáhnout spuštěním přes příkazovou řádku následujícím příkazem:

runas /USER:Administrator "C:\Program Files\Dia\bin\diaw.exe"

kde za /USER: je zadán uživatel s potřebnými právy a za ním cesta ke spouštěcímu souboru dané aplikace. Po jeho zadání je vyžádáno heslo tohoto uživatele a aplikace je spuštěna stejným způsobem jako v předchozím případě.

Po načtení aplikace je skript dostupný v textové nabídce v okně diagramu v podmenu „Dialogy“ pod názvem „IPscanner“. Po jeho odklepnutí se zobrazí dialogové okno, ukázané na obrázku č.9, se vstupním polem a potvrzovacím tlačítkem. Do vstupního pole se zadává IP rozsah (IP verze 4) zvolené podsítě ve formátu x.x.x.x-y.y.y.y (první_adresa-poslední_adresa) nebo x.x.x.x/z (adresa_sítě/počet_bitů_masky). Po odklepnutí potvrzovacího tlačítka je zkontrolována správnost zadaného rozsahu. Pokud je uznán za správný, je započato skenování zadané části sítě. Tato procedura je časově náročná - násobně vzhledem

k velikosti rozsahu sítě. Program při jejím průběhu neodpovídá (tváří se jako „zaseknutý“) a v některých chvílích může být náročný na výkon počítače. Po dokončení jeho práce (program opět začne reagovat) je na kreslícím plátně vytvořen diagram zadané sítě.



Obrázek 9 - Dialogové okno pro zadávání rozsahu sítě