Jihočeská univerzita

Přírodovědecká fakulta

Ústav aplikované informatiky

Bakalářská práce

Automatické zjištění topologie a monitoring sítě na bázi FreeBSD

Autor: Petr Macek

Vedoucí práce: Mgr. Michal Kočer, Dr.rer.nat.

Garant práce: Ing. Rudolf Vohnout

České Budějovice

2013

Bibliografické údaje

MACEK,P.:2013: Automatické zjištění topologie a monitoring sítě na bázi FreeBSD [Automatic topology recognition and network monitoring based on FreeBSD] – 53 p., Faculty of Science, The University of South Bohemia, České Budějovice, Czech Republic.

Abstrakt

Cílem práce je vytvoření systémového nástroje pro zjištění topologie a následný monitoring LAN. Tento nástroj bude postaven na OS FreeBSD a vytvořen jako bootovatelný obraz vhodný pro provoz z USB disku. Pro zjištění údajů topologie a monitoringu sítě bude kromě protokolů nižších vrstev TCP/IP využito protokolu SNMP. Vedle zobrazení topologie sítě bude u aktivních prvků monitorován jejich stav, síťové přenosy a další hodnoty. Komunikace s uživatelem bude probíhat přes www rozhraní. Snahou je vytvořit nástroj spustitelný bez instalace na jakémkoli počítači v síti a provést rychlé zjištění topologie, její zobrazení a základní monitoring. Má být opakem nástrojů typu NMS, které vyžadují instalaci na server, databázový server a další prostředky.

Klíčová slova

Topologie sítě, FreeBSD, SNMP, IPv4

Abstract

The goal of the Bachelor Thesis is the creation of system tool for topology recognition and following LAN monitoring. This tool will be built on OS FreeBSD and formed like bootable image suitable for running from the USB stick. Besides the lower layers protocols TCP/IP, the protocol SNMP will be used for topology recognition and network monitoring. The tool will monitor state, network transmissions and other values at active network components too. Communication with the user will be realized via www interface. The goal is to create the tool executable on any computer without installation, to perform quick topology recognition, its visualisation and basic monitoring. It is supposed to be the opposite of the tools of NMS type that need installation on server, database server and have other requirements.

Keywords

Network topology, FreeBSD, SNMP, IPv4

Poděkování

Rád bych poděkoval vedoucímu práce Michalovi Kočerovi a garantovi práce Rudolfovi Vohnoutovi za připomínky a postřehy.

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

České Budějovice 18. 4. 2013

Obsah

1. Úvod

V dnešní době jsou na počítačové sítě kladeny stále vyšší nároky jak z hlediska propustnosti, tak i stability a odolnosti proti výpadkům. Aby bylo možné takové sítě provozovat, je nutné se věnovat monitoringu sítě a síťových zařízení. Monitoring umožňuje získat informace o aktuálním stavu sítě, slabých místech a prvcích vykazujících chyby. Nutná je také znalost topologie sítě, která usnadňuje rozšiřování sítě a změny. Bakalářská práce se zabývá právě zjišťováním síťové topologie a následným monitoringem LAN sítě.

2. Teoretická část

2.1 Síťové okolí

Aktivní prvky (přepínače, routery, bridge) ke své správné činnosti nutně nemusí znát kompletní síťové okolí. Každý typ zařízení může mít naučené různé informace. Přepínači postačuje znalost druhé vrstvy, proto může vědět jen o zařízeních dostupné jen na této vrstvě a neví nic o IP sítích. Přepínač se učí z přicházejících rámců, na kterém portu je která MAC adresa. Tuto informaci je možné použít pro zjištění přímo připojených sousedů.

Naopak routeru pracuje na 3. vrstvě, směruje IP provoz přes různé sítě a údaje z druhé vrstvy pro směrování nevyužívá. Pro předávání ale používá druhou vrstvu a může tedy mít podobné informace jako přepínač. Dále může mít např. informace o VLANech, výpadcích linek naučených přes routovací protokoly. I tyto informace mohou být pro topologii důležité.

Pro potřebu větších sítí dále vznikly tzv. discovery protokoly, které zjišťují nejbližší síťové okolí prvku - přímo připojené sousedy. Dnešní aktivní prvky tyto discovery protokoly často používají a poskytují mnoho důležitých informací jako je například číslo a název portu, IP adresu protistrany apod.

Pro zjištění celé topologie je tedy nutné získat informace od co největšího počtu zařízení a zkombinovat je. Níže uvádím seznam protokolů a zdrojů informací, z kterých je možné síťové okolí zjistit.

2.1.1 LLDP (Link Layer Discovery protocol)

LLDP *[1]* je protokol linkové vrstvy. Jedná se o jednocestný protokol, který pouze vysílá informace a nedochází k žádnému potvrzování přijaté informace nebo k navazování spojení. Aktivní prvek odesílá přes své porty informace o sobě ostatním LLDP zařízením v síti. K odesílání dochází periodicky nebo při změně na aktivním prvku. K šíření LLDP oznámení se využívá multicast 01-80-C2-00-00-0E (Ethernet type0x88cc). Tuto zprávu přijímají nejbližší sousedé, informace zpracují a dále nepředávají. Pomocí LLDP je tedy možné zjistit pouze přímo připojené sousedy. Údaje zjištěné LLDP oznámeními se uchovávají pouze po určitou dobu, s každým dalším doručením stejné informace se pouze aktualizuje čas zjištění informace. Pokud oznámení přestane přicházet a vyprší čas aktualizace, je informace smazána. Časy mohou být ovlivněny i hodnotou TTL v oznámení.

LLDPDU obsahuje vždy:

1) **chassis ID**

2) **port ID**

3) **Time to live** - po jakou dobu si odesílací prvek myslí, že budou informace validní. Pomocí nastavení TTL rovno 0, je možné u příjemce vynutit smazání informace z LLDP databáze. Toho se využívá například při vypnutí portu

4) **end of lldpdu**

Další nepovinné informace, které mohou být posílané jsou specifikováné v IEEE normě. Zařízení často posílají i informace o VLAN ID, VLAN name, Link aggregaci, POE, moznosti systemu (switching, routing).

Ve výchozím stavu aktivní prvek odesílá LLDP oznámení na všech svých portech. Konfigurací lze ovlivnit čas odesílání, vypršení, vyjmenovat porty, kde nebude LLDP odesíláno nebo dokonce zákaz veškerého rozesílání LLDP oznámení. V takovém případě má prvek přehled o svém okolí, ale sousedé jej nevidí - v tomto případě prvek jejich oznámení přepošle beze změny, tzn. chová se jako aktivní prvek bez managementu. Lze nakonfigurovat i opačný případ, kdy prvek pouze vysílá, ale neukládá si informace o okolí. Takové konfigurace mohou značně ovlivnit zjišťování topologie.

Protože je protokol standardizovaný normou IEEE, používají ho různí výrobci ve svých zařízeních a existuje i softwarová implementace pro Linux/Unix a Windows. Pokud je použit na tkaobých systémech, můžeme získat i informace o koncových zařízeních. Existuje rozšíření LLDP-MED (media endpoint discovery), které používají VoIP zařízení a vysílají o sobě LLDP informaci. Switch jim potom dle těchto informací může na jejich portech nastavit prioritu a další parametry pro takové koncové zařízení. Toto rozšíření nemá pro zjištění topologie velký význam, protože neobsahuje informaci o aktivních prvcích, ale pouze o koncových zařízeních.

Pro zjištění topologie postačují základní informace:

- na jakém portu bylo oznámení přijato
- jaké zařízení informaci vyslalo (IP adresa a popis)
- přes jaký port byla informace odeslána

2.1.2 CDP (Cisco Discovery Protocol)

CDP *[2]* je obdobný protokol Linkové vrstvy, ale jedná se o proprietární protokol společnosti Cisco. Dnes obsahuje podobná rozšíření jako je LLDP-MED a umožňuje i konfiguraci sousedních zařízení. Pro zjištění topologie postačuje základ tohoto protokolu fungující obdobně jako LLDP.

Ač se jedná o proprietární protokol, je dnes využíván i dalšími výrobci, například jej používá software Winbox k dohledání zařízení s operačním systémem Mikrotik. Někteří výrobci síťových prvků (např. 3com, později HP) využívají standardní LLDP protokol, ale rozpoznají CDP a umožňují s nimi pracovat.

Pomocí CDP oznámení může být distribuována i informace týkající se pouze Cisco zařízení, např. VTP doména. Pro zjištění topologie jsou ale tyto informace nepodstatné. Pro zjištění topologie dostačují základní informace, které jsou shodné s LLDP protokolem.

2.1.3 Další discovery protokoly

Následující protokoly také řeší objevování sousedů, ale nepoužil jsem je kvůli nedostatku relevantních informací nebo nedostupnosti přes monitorovací protokol, který ve své práci používám, tj. SNMP.

NDP (Nortel Discovery Protocol) - proprietární protokol firmy Nortel

Plug and Play *[3]* - u síťových zařízení nebývá použit, slouží spíše k nastavení domácích zařízení.

NDP (Neighbor Discovery Protocol) *[4]* - součást IPv6, využívá ICMPv6, nachází okolní prvky, dostupné routery a dns servery

TRILL (TRansparent Interconnection of Lots of Links) *[5]*- jedná se o nový linkový protokol, který má vyřešit nedostatky Spanning tree protokolu, hlavně využití linek, které jsou u STP blokované a nevyužívané. Zatím není rozšířen.

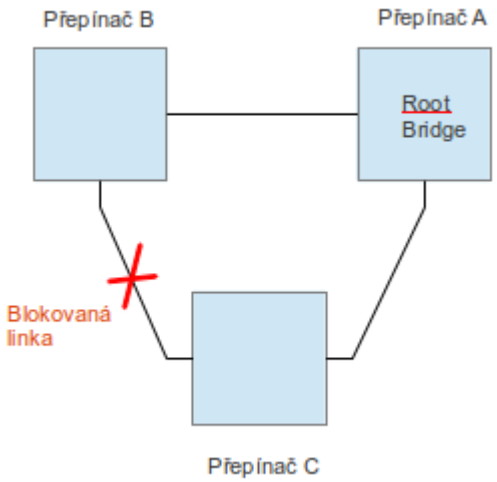
LLTD (Link Layer Topology Discovery) *[6]* - využívaný v OS Windows, podporuje drátové i bezdrátové sítě. Je obsažen v OS Windows od verze Vista výše. Tyto informace ale není jak získávat, protože neexistuje SNMP MIB.

2.1.4 STP (Spanning tree protocol)

Nejedná se o protokol zjišťující síťové okolí, ale některé informace o sousedech obsahuje. Spanning tree protocol slouží k detekci a blokování smyček v L2 síti. Smyčka může způsobit zahlcení přepínače i celé sítě.

Princip přepínače je takový, že se postupně učí z příchozích rámců, která MAC adresa je na kterém portu. Tyto údaje si po omezenou dobu uchovává v paměti. Když přijde rámec, prohledá tuto tabulku a v případě nálezu záznamu, přepošle rámec pouze na příslušný port. Pokud záznam pro MAC adresu v tabulce není, je rámec rozeslán na všechny porty kromě portu kterým přišel. Smyčka v síti může způsobit, že se rozeslaný rámec vrátí přes jiný port a je znovu rozeslán na všechny porty. Stejně se zachovají ostatní přepínače a dojde k mnohonásobnému rozmnožení rámců až do úrovně zahlcení linek nebo vytížení přepínačů. Dalším problémem může být zahlcení tabulky naučených MAC adres, protože u smyčky se rámec vrací mnoha porty. STP protokol smyčky rozpojuje a tím brání takovému zahlcení.

Na druhou stranu ale často redundanci linek požadujeme, protože nám může zajistit fungování sítě v případě výpadku přepínače nebo nějaké linky. Záměrně je vytvořena topologii se smyčkou, kterou spanning tree zablokuje a v případě výpadku aktivní cesty umožní fungování přes dříve blokovanou linku.



Obr. 1: Rozpojení smyčky pomocí spanning tree protokolu

Pro zjištění smyčky musí mít STP přehled o topologii sítě. Je zvolen tzv. Root bridge a přepínače v síti si na základě rychlosti linek naleznou nejvýhodnější cestu k němu. Ostatní cesty k root bridge zablokují a tyto informace uchovávají. U portu, který je směrem k root bridge je zde údaj o protistraně - buď přímo root bridge nebo nadřazeném přepínači.

Existuje několik standardizovaných verzí STP protokolu (STP, RSTP, MSTP), Cisco používá vlastní modifikaci PVST+ a RPVST+. Ač existuje větší množství verzí a modifikací, uchovávané informace jsou shodné *[7]*.

2.1.5 ARP a CAM tabulka

Naučené okolí je možné zjistit z CAM (Content addressable memory) tabulky přepínače. Do této tabulky jsou ukládány informace o naučených MAC adresách zařízení na jednotlivých portech. Tyto informace využívá přepínač ke své činnosti a získává je z běžné L2 komunikace. Přepínač si může také uchovávat informace v ARP tabulce. Zde jsou udržovány informace o párování MAC a IP adres. Při spojení těchto tabulek získáme informaci o naučených IP adresách na portech přepínače.

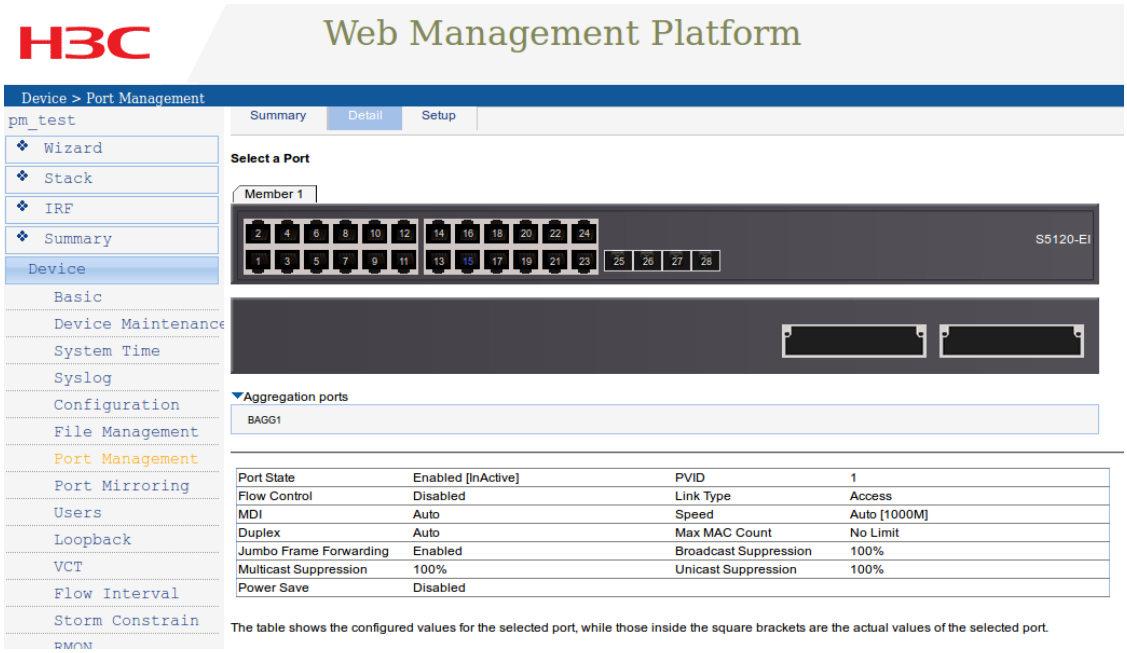
2.2 Zjišťování informací z aktivního prvku

Většina síťových zařízení umožňuje zjišťování informací a konfiguraci více cestami. Výhodou různých způsobů může být přehlednější zobrazení nebo snadnější práce. Získáváme informace, které zařízení samo používá ke své činnosti a proto jsou uložené tak, aby byly pro zařízení co nejsnáze použitelné. To je například zápis adres v šestnácková soustavě, číselné indexy a další. Pro člověka nemusí být takový zápis na první pohled čitelný a přehledný. Proto se používají i další metody, které jsou pro běžnou práci výhodnější, ale pro automatické zjišťování informací jsou nevhodné. Tyto metody také pracují se stejnými informacemi, ale upravují je a formátují pro dané užití.

2.2.1 WWW

Výhodou grafického znázornění a velké množství informací zobrazitelných najednou. Pro automatické zpracování je nevýhodné, protože není jednotné mezi výrobci, často se liší i v různých verzích softwaru stejného zařízení. Pro zjišťování informací by bylo nutné parsovat html stránky. Pokud jsou informace zobrazeny v obrázku, bylo by velmi obtížné

se k informacím dostat.



Obr 2: Ukázka www managementu

2.2.2 Telnet a SSH

Většinou umožňuje přístup na konzoli zařízení, kde je možné v textovém režimu zadávat příkazy a číst odpovědi. Problém se zjišťováním informací je stejný jako u www managementu. Každý výrobce používá vlastní sadu příkazů nebo přes telnet zobrazí jen menu s výběrem, kterým se uživatel pohybuje jen volbou čísla.

Zjišťování informací by znamenalo parsovat textový výstup. SSH nabízí pro zvýšení bezpečnosti šifrování komunikace. Dalším problémem je, že šifrování a autentizace uživatele SSH klíčem je pro přepínač výkonově poměrně náročná operace. Při použití by docházelo ke zbytečnému vytěžování CPU přepínačů.

2.2.3 SNMP

Výhodou je jeho velké rozšíření a podpora všemi výrobci. Data jsou uložené v přesně daných formátech a je standardizován způsob komunikace, proto není nutné řešit výjimky nebo hledat řešení pro každého výrobce. Navíc získáváme data ve formátech a strukturách,

se kterými pracuje zařízení a nejsou ovlivněny dalším zpracováním. SNMP je patrně jedinou možnou cestou, jak snadno zjistit informace ze zařízení různých výrobců. Protože je důležitou částí práce, je popsán v následující kapitole.

2.2.4 ICMP

Dostupnost zařízení je také možné zjišťovat pomocí ICMP *[8]* protokolu, konkrétně nástroje ping a traceroute. Existují i nástroje, které spouští ping paralelně a jsou schopné velice rychle ověřit stovky i tisíce adres. Metoda ale není příliš použitelná z několika důvodů. Chování ICMP lze nastavit tak, že zařízení na ping nebo traceroute neodpovídá. Přepínače dnes často kvůli ochraně proti zahlcení ICMP požadavky nezpracovávají nebo je zpracovávají s nejnižší prioritou a nemusí odpovědět včas. V případě traceroute nesnižují TTL paketu, takže v cestě vypsané programem traceroute nebude takové zařízení vidět.

2.2.5 Odposlech síťové komunikace

Další možnost zjišťování síťového okolí je odposlech síťového provozu. Je možné zachytit LLDPDU a z něj vyčíst informace. Problémem je ale množství míst, kde by bylo nutné odposlech provádět. Druhým problémem je množství zachycených dat a jejich složité zkoumání. Přepínače nabízí možnost vytvářet kopii provozu z vyjmenovaných portů a posílat ji na zvolený port. Zde by se musel provoz analyzovat. Port mirroring ale není možné nasadit na všech prvcích a všech portech, protože by došlo k přetížení aktivních prvků.

2.3 SNMP (Simple network management protocol)

Jak vyplývá z názvu, jedná se o jednoduchý protokol aplikační vrstvy určený ke správě síťových zařízení. Správa ale neznamená pouze nastavování prvků, ale i sledování a reakce na neobvyklé chování nebo stavy.

Jedná se standardizovaný protokol a první RFC, které se ho týká pochází již z roku 1988. Protože jiné management protokoly se neprosadily, je SNMP *[11]* podporován drtivou většinou výrobců a je to prakticky jediná cesta, jak spravovat a dohlížet celou síť.

I když je možné pomocí SNMP zařízení konfigurovat, častěji se používá pouze k monitoringu (čtení dat ze zařízení) a konfigurace se provádí jinou cestou., např. Výše zmiňované metody www, telnet, ssh.

SNMP má jednoduchou architekturu. Existují dva typy objektů:

agent:

- snmp server běžící na sledovaném zařízení
- monitoruje zařízení a sbírá z něho informace
- poslouchá na UDP portu 161

manager:

- snmp klient, u dohledových sw se nazývá NMS (network management system)
- posílá dotazy agentovi a přijímá odpovědi
- čeká na snmp trapy na UDP portu 162

SNMP pracuje ve dvou režimech. První z nich je vyžádaná klient/server komunikace. Manager pošle dotaz na agenta, ten požadavek zpracuje a odpoví. Druhou možností je SNMP trap. U toho neexistuje požadavek, ale agent může sám od sebe při námi definovaných událostech (např. výpadek napájení, plná paměť, ..) poslat zprávu manageru s informací. Tato zpráva se nazývá Trap a jedná se o UDP datagram, které očekává manager na portu 162.

Máme tedy dvě možnosti monitoringu - pasivně vyčkávat na trapy při dosažení kritických hodnot nebo se aktivně periodicky vyptávat na stav. Často se používá obou režimů zároveň - periodicky se vyčítají data, ale interval dotazování není příliš častý, aby zbytečně nedocházelo k zatěžování prvků nebo sítě a zároveň se přednostně reaguje na přijaté trapy.

SNMP používá výhradně UDP protokol, který nezaručí doručení dat. Je to z důvodu jednoduchosti UDP a snadné implementace v zařízeních. UDP má i podstatně menší nároky při komunikaci (u TCP se musí vytvořit spojení, potvrzovat přijaté TCP segmenty, ...). Nevýhodou je, že například při odeslání trapu nemá agent šanci zjistit, zda trap dorazil nebo se doručení nepodařilo. Stejně tak agent po dotazu, na který nepřijde odpověď neví, zda zařízení z nějakého důvodu neodpovídá nebo se jen ztratil UDP datagram.

Při komunikaci mezi agentem a managerem nebo mezi dvěma managery je možné použít tyto metody:

GET – slouží pro čtení jedné hodnoty, využívá ji manager

GET-NEXT – slouží pro čtení následující hodnoty, je možné použít k sekvenčnímu procházení tabulek, využívá ji manager

GET-RESPONSE – odpověď agenta na GET nebo GET-NEXT

SET – slouží k nastavení jedné hodnoty, posílá ji manager

TRAP – zpráva, kterou posílá agent

GETBULK – slouží ke získání většího množství informací na jeden dotaz, umožňuje získat celou tabulku najednou, odesílá manager

Inform – používá se ke komunikaci mezi NMS nebo managery. Může být generován i agentem pro odeslání informací manageru. Manager na přijatou zprávu posílá inform s potvrzením.

Notification – Protože SNMPv1 používal pro trapy jiný format PDU, došlo ve verzi 2 ke sjednocení a notification PDU má stejný formát jako Get a Set

Dnes existují více verzí SNMP protokolu:

SNMPv1 - pro monitoring větších sítí bylo nevýhodné, že neexistovala podpora pro komunikaci mezi managery. Nepodporoval získání více odpovědí v jednom dotazu a přenos byl téměř nezabezpečený. Podporuje metody GET, SET, GET-NEXT, TRAP. Protokol je definován v RFC 1052, 1065, 1067, 1155, 1212, 1155, 1213, 1157.

SNMPv2 - přidává GET-BULK (pro čtení tabulek) a INFORM. Zároveň přidává výšší zabezpečení, ale kvůli složitější implementaci se prosadila verze 2c. RFC 1441, 1452.

SNMPv2c - jedná se o verzi 2, ale používá stejné zabezpečení jako v1. RFC 1901.

SNMPv2u - není tak složitý jako v2, ale poskytuje vyšší zabezpečení než v1. RFC 1910.

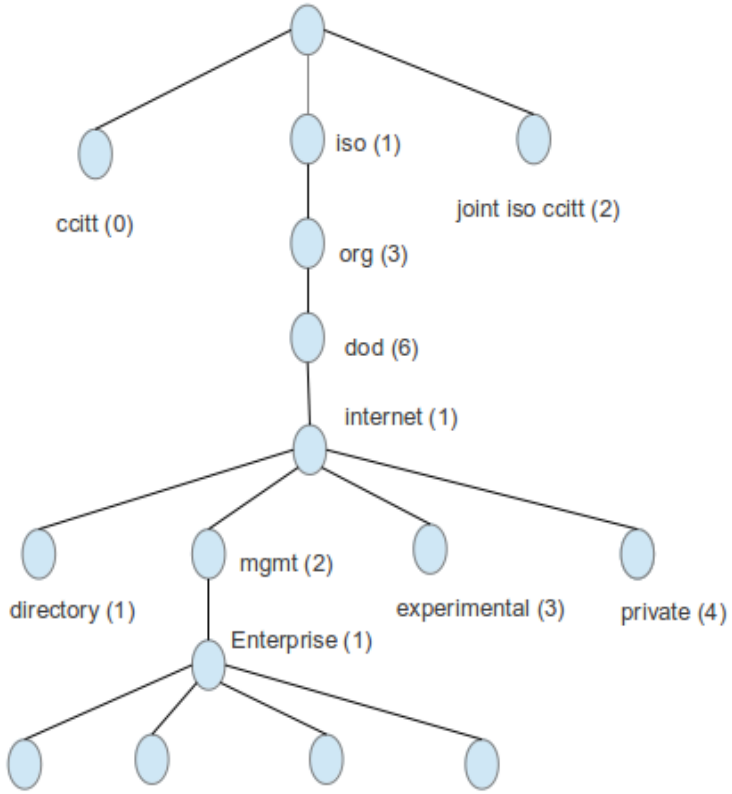
SNMPv3 - přidává šifrování, integritu zpráv a autentifikaci. RFC 3411 – 3418.

V praxi se používá převážně verze 1, 2c a 3. Verze 3 se liší hlavně v zabezpečení. Verzi 1 obsahuje velice jednoduché zabezpečení formou tzv. community stringu. Jedná se o textový řetězec (v podstatě heslo), které je přenášeno spolu s dotazem. Na základě tohoto řetězce jsou definovaná přístupová práva k dotazovaným objektům. Verze 1 tento řetězec nešifruje, proto se nedá o zabezpečení moc mluvit. Velmi často jsou používané community stringy "public" pro přístup pouze pro čtení a "private" pro čtení i zápis.

Verze 2 zavedla složitý systém zabezpečení a právě díky složité implementaci zabezpečení nikdy neprosadila. Vznikla verze 2c, která je kombinací verze 2 a zabezpečení pomocí community stringů z verze 1. Verze 3 nemění protokol, pouze přidává zabezpečení. Umožňuje

šifrování zpráv, integritu zpráv a autentizaci odesílatele. Jejich použití ale není povinné. Je zde i jemnější dělení práv, uživatele se přiřazují do skupin a skupinám se přidělují tzv. pohledy z kterých je možné vyřadit libovolné podstromy a tím omezit uživatelům přístup.

Pro uchovávání informací agent používá tzv. MIB (management information base). RFC 1155, RFC 1212, RFC 1215. Jedná se o stromovou strukturu, která obsahuje kořen, podvětve a listy. Byla definována základní struktura a jednotlivým výrobcům jsou přidělovány podstromy. Díky tomu si každý výrobce může svůj podstrom definovat do šířky nebo výšky libovolně aniž by ovlivnil zbytek stromu. Díky tomu může zařízení bez problémů implementovat více MIBů, kromě standardizovaných i od jednotlivých výrobců.



Obr 3: Ukázka struktury MIB

Všechny objekty ve stromu kromě kořenu (tedy větve a listy) mají číselný identifikátor, který je na dané úrovni jedinečný. Každý objekt ve stromové struktuře musí mít jednoznačný identifikátor. Ten se nazývá - OID (Object Identifier) a je tvořen zápisem čísel jednotlivých uzlů od kořenu až k požadovanému objektu, čísla se oddělují tečkou.

Ukázka č. 1 – číselné OID:

[root@mail /]# snmpget -On -v1 -c public 192.168.222.108 1.3.6.1.2.1.1.3.0

.1.3.6.1.2.1.1.3.0 = Timeticks: (1417212507) 164 days, 0:42:05.07

OID mohou mít i slovní identifikátor, s kterým se pracuje stejně jako s číselným. Výhodou je větší přehlednost

Ukázka č. 2 – jmenné OID:

[root@mail /]# snmpget -On -v1 -c public 192.168.222.108 iso.org.dod.internet.mgmt.mib-2.system.sysUpTime.sysUpTimeInstance

.1.3.6.1.2.1.1.3.0 = Timeticks: (1417245530) 164 days, 0:47:35.30

MIB dále může obsahovat popis hodnot, to může často pomoci při hledání požadovaných OID. Jedná se o textový soubor zapsaný pomocí ASN.1.

List ve stromové struktuře může obsahovat buď skalární hodnotu nebo tabulku. Skalárních typů je několik:

- **integer** - jednoduché číslo
- **counter** - počítadlo, jedná se o nezáporné číslo. Pokud přesáhne max. hodnotu, začne počítat od nuly, použití např. jako počítadlo přenesených paketů
- **gauge** - jednoduché číslo, které ale může i klesat, použití např. u teploty
- **timeticks** - čas v setinách vteřiny
- **ipAddress** - ip adresa
- **string, hexString** - řetězec

Druhou možností je tedy tabulka, která v jednotlivých buňkách obsahuje výše uvedené skalární hodnoty. Pro práci s tabulkou ale nemá SNMP žádný nástroj, je možné tabulku procházet sekvenčně pomocí get-next nebo náhodně, pokud známe index žádaného objektu.

Tabulka je složitější struktura, ukážeme si ji na seznamu síťových rozhraní. Jedná se o IF-MIB, seznam portů je dostupný přes OID 1.3.6.1.2.1.2.2 (.iso.org.dod.internet.mgmt.mib-2.interfaces.ifTable). Záznamy tabulky jsou dostupné přes OID 1.3.6.1.2.1.2.2.0 (.iso.org.dod.internet.mgmt.mib-2.interfaces.ifTable.ifEntry). Zde jsou pak jednotlivé sloupce tabulky, například název rozhraní 1.3.6.1.2.1.2.2.1.2 (.iso.org.dod.internet.mgmt.mib-2.interfaces.ifTable.ifEntry.ifDescr) nebo velikost MTU 1.3.6.1.2.1.2.2.1.4 (.iso.org.dod.internet.mgmt.mib-2.interfaces.ifTable.ifEntry.ifMtu). Poslední číslo tedy určuje sloupec. Dále má každá tabulka index. Ten může být tvořen jednou nebo více hodnotami. Zde je indexem číslo portu. Tím nám tedy vzniká OID .1.3.6.1.2.1.2.2.1.4.X, kde X je číslo portu.

Pokud známe nebo můžeme odvodit index, není již nutné procházet celou tabulkou, ale můžeme se zeptat přesně.

Ukázka č. 3 – zjištění velikosti MTU na portu 3:

[root@mail /]# snmpwalk -On -v1 -c public 192.168.222.181
.iso.org.dod.internet.mgmt.mib-2.interfaces.ifTable.ifEntry.ifMtu.3

.1.3.6.1.2.1.2.2.1.4.3 = INTEGER: 9216

Ukázka č. 4 – zjištění velikosti MTU všech portů:

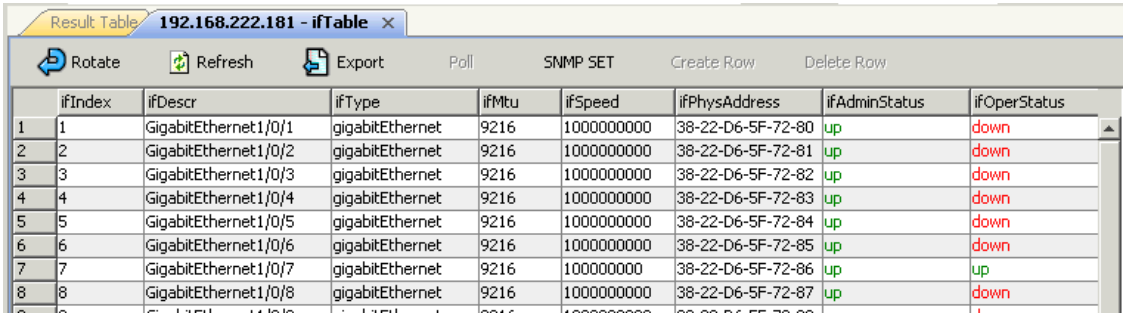
[root@mail /]# snmpwalk -On -v1 -c public 192.168.222.181
.iso.org.dod.internet.mgmt.mib-2.interfaces.ifTable.ifEntry.ifMtu

.1.3.6.1.2.1.2.2.1.4.1 = INTEGER: 9216

.1.3.6.1.2.1.2.2.1.4.2 = INTEGER: 9216

.1.3.6.1.2.1.2.2.1.4.3 = INTEGER: 9216

…



Result Table | 192.168.222.181 - ifTable

Rotate | Refresh | Export | Poll | SNMP SET | Create Row | Delete Row

	ifIndex	ifDescr	ifType	ifMtu	ifSpeed	ifPhysAddress	ifAdminStatus	ifOperStatus
1	1	GigabitEthernet1/0/1	gigabitEthernet	9216	1000000000	38-22-D6-5F-72-80	up	down
2	2	GigabitEthernet1/0/2	gigabitEthernet	9216	1000000000	38-22-D6-5F-72-81	up	down
3	3	GigabitEthernet1/0/3	gigabitEthernet	9216	1000000000	38-22-D6-5F-72-82	up	down
4	4	GigabitEthernet1/0/4	gigabitEthernet	9216	1000000000	38-22-D6-5F-72-83	up	down
5	5	GigabitEthernet1/0/5	gigabitEthernet	9216	1000000000	38-22-D6-5F-72-84	up	down
6	6	GigabitEthernet1/0/6	gigabitEthernet	9216	1000000000	38-22-D6-5F-72-85	up	down
7	7	GigabitEthernet1/0/7	gigabitEthernet	9216	100000000	38-22-D6-5F-72-86	up	up
8	8	GigabitEthernet1/0/8	gigabitEthernet	9216	1000000000	38-22-D6-5F-72-87	up	down

Obr 4: Zobrazení IF-MIB pomocí aplikace SNMP MIB browser

Kromě získané hodnoty nebo hodnot je často důležitý i index objektu, vždy nejde odvodit a může v něm být například uvedena MAC adresa. Ta se většinou zapisuje v šestnáctkové soustavě a protože index musí být numerický, bývá v indexu zapsána v desítkové soustavě. Na dalším výstupu vidíte, že MAC adresa je součástí indexu i vlastní odpovědi jen pokaždé zapsaná v jiné soustavě.

Ukázka č. 5 – výpis OID včetně indexu:

[root@mail /]#snmpwalk -On -v1 -c public 192.168.222.108 1.3.6.1.2.1.17.4.3.1.1

.1.3.6.1.2.1.17.4.3.1.1.0.4.117.232.65.11 = Hex-STRING: 00 04 75 E8 41 0B

.1.3.6.1.2.1.17.4.3.1.1.0.8.2.28.203.114 = Hex-STRING: 00 08 02 1C CB 72

.1.3.6.1.2.1.17.4.3.1.1.0.21.93.222.17.10 = Hex-STRING: 00 15 5D DE 11 0A

2.4 Hloubka zkoumání , typy zařízení

Pro zjištění a vykreslení topologie jsou nutné informace pouze z přepínačů. Podmínkou jsou prvky s managementem a nakonfigurovaným SNMP. V síti je často mnoho dalších zařízení, které podporují SNMP a používají také discovery protokoly. Lze z nich tedy vyčíst často stejné množství informací jako z přepínačů. Pro zjištění topologie nemají význam, ale je možné získat podrobnější mapu. V případě wifi controllerů, AP, bridge je dobré toto do topologie zaznamenat, protože se často jedná o důležitá zařízení.

Další skupinou jsou zařízení také podporující SNMP, ale jedná se o méně důležité prvky jako jsou IP a VoIP telefony, printservery. Informace z nich jsou důležité pouze v případě, kdy chceme lokalizovat koncová zařízení.

Protože cílem práce je zmapovat celou síť a získat přehled o co největším množství zařízení, je nutné zpracovávat informace i z těchto prvků a snažit se rozpoznat typ zařízení. Neexistuje ale žádný standardizovaný způsob, kterým lze tuto informaci získat. V případě zařízení odpovídajího na SNMP lze typ odhadnout. Pokud např. odpovídá na dotazy do Printer MIBu, bude se patrně jednat o tiskárnu. Kombinace MIBu některého z discovery protokolů a většího množství portů ukazuje na přepínač.

U zařízení nepodporujících SNMP je situace ještě složitější. Detekci je možné provádět pouze dle otevřených portů. Dotazem na nejčastěji používané porty jako je sdílení v sítích Microsoft, SSH, DNS apod. je možné zjistit seznam běžízích služeb. Z něho je potom možné odvodit typ zařízení.

2.5 Zařízení a protokoly ovlivňující topologii

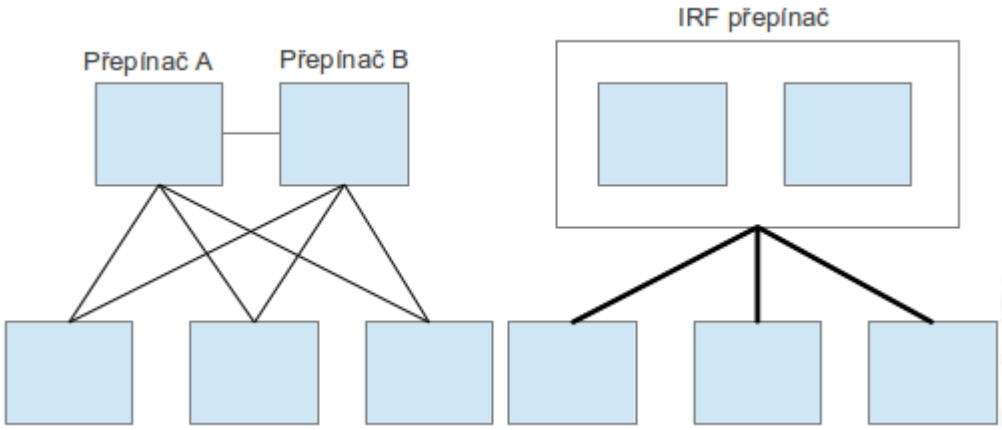
Topologie v dnešních sítích není jen prosté zapojení kabelů. Mnoho zařízení a zapojení topologii ovlivňují a fyzická se od logické může značně lišit. Dále uvádím seznam technologií ovlivňujících topologii.

2.5.1 Virtualizaci přepínačů

Ze dvou nebo více přepínačů vytvoříme jeden virtuální. Takové řešení má několik výhod:

- zjednodušení sítě, v rámci virtuálního zařízení není nutný STP, VRRP a další vše si řeší přepínač interně sám
- velmi rychlá konvergence při výpadku jednoho z přepínačů (méně než 40 ms)
- management jednoho virtuálního zařízení namísto několika samostatných
- možnost linkové agregace přes více fyzických zařízení
- využití všech linek, nedochází k blokaci jako u STP

IRF (Intelligent Resilient Framework) je technologie vyvinutá firmami 3com a H3C. Obě firmy byly odkoupeny firmou HP, takže se s IRF dnes setkáme v jejích přepínačích. Cisco používá svou technologii nazvanou Cisco Virtual Switching Systems (VSS).



Obr. 5: Virtualizace přepínačů

Jak je vidět na obrázcích, virtualizace přepínačů zásadně ovlivňuje topologii sítě. U IRF lze virtualizovat až 10 zařízení do jednoho a při zjišťování údajů ze SNMP vidíme pouze tento jeden virtuální přepínač a není možné zjistit žádné informace o tom, jak je vnitřně zapojen.

2.5.2 Klasické stohování přepínačů

Jedná se o starší, ale stále používaný způsob zapojení více přepínačů do jednoho celku. Toto stohování neumožňuje tak pokročilé funkce jako virtualizace. Výhodou zůstává řízení jednoho prvku a zjednodušení sítě, ale není možná linková agregace přes více fyzických zařízení a ve stohu se někdy používá STP protokol pro blokování smyček. Při zjišťování informací pomocí SNMP je stejně jako u virtualizace stoh rozpoznán jako jedno zařízení.

2.5.3 Virtualizace serverů

Virtualizace serverů je dnes velmi rozšířená a často nasazována. Její výhodou jsou mimo jiné úspory elektrické energie, lepší využití výkonu HW. Na jednom hardwaru je tedy více serverů a proto virtualizace musí obsahovat virtuální přepínač, do kterého jsou jednotlivé virtuální servery servery zapojené.

S rostoucími nároky na dostupnost a zabezpečení virtuálních strojů rostou i požadavky na virtuální přepínače. Platforma VMWare dnes nabízí možnost jako virtuální přepínač použít Cisco IOS, který je v běžně v hardwarových přepínačích. Zde nám tedy z pohledu topologie vzniká plnohodnotný přepínač, který je ale opět virtuální.

Protože virtualizovaných serverů bývá na jednom hardwaru více, je zde požadována i vyšší propustnost. Proto virtualizace dnes umožňují link agregace, takže virtuální přepínač bývá

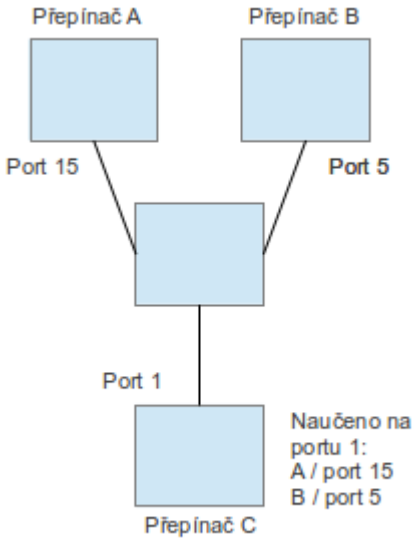
propojen s běžnými přepínači agregovanými linkami. Dále je možné před virtuální stroje ještě umístit virtuální firewall, který se v topologii sítě jeví jako další HW zařízení i když je pouze virtuální.

2.5.4 VRRP (Virtual Router Redundancy Protocol)

Protokol pro vytvoření virtuálního routeru ze dvou nebo více fyzických. Jeho nasazení je podobné jako u virtualizace přepínačů *[11]*.

2.5.5 Prvky se základním nebo bez managementu

Levné přepínače často neobsahují podporu žádného discovery protokolu nebo nemají management vůbec. Takové zařízení nelze přes SNMP detekovat a lze jej odhalit pouze díky ostatním zařízením s discovery protokoly. Tyto protokoly mají přehled pouze o přímo připojených sousedech, od kterých dostaly oznámení. V případě, že mezi více přepínači bude zapojený jiný přepínač bez těchto protokolů, dojde sice ke zjištění informací o sousedech, ale bude zkreslené.



Obr. 6: Ukázka sítě s prvkem bez managementu

Při podobném zapojení se přepínač 1 naučí na jednom svém portu 2 další přepínače. Stejně tak se i ostatní přepínače naučí přes jeden port 2 zařízení a o přepínači uprostřed nic neví.

Při kreslení topologie je nutné toto zohledňovat a při detekci více naučených sousedů přes jeden port vykreslit neznámé zařízení.

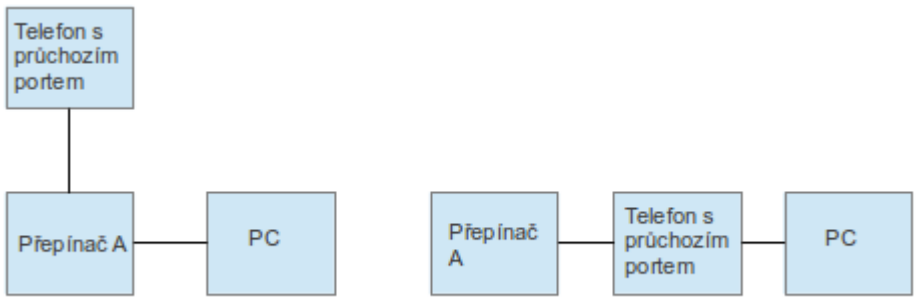
2.5.6 Špatně nakonfigurovaný přepínač

Konfigurace přepínačů umožňuje nastavit mnoho kombinací a způsobů chování. Lze nastavovat i discovery protokoly CDP a LLDP. Výrobci je ve většině případů v základní konfiguraci ponechávají zapnuté. V případě nevhodné konfigurace jako je např. zahazování LLDP rámců nebo jejich povolení, ale zakázání ukládat zjištěné informace, mohou chybět důležité informace nutné pro zjištění topologie. V krajním případě, kdy na přepínači vypneme vše týkající se SNMP a discovery protokolu, degradujeme přepínač na jednoduchý přepínač popsaný výše a topologie se stane nezjistitelnou.

2.5.7 Průchozí zařízení

Většinou se jedná o IP nebo VoIP telefony, které obsahují malý dvouportový přepínač a nepodporují SNMP. Detekce je podobná jako u přepínačů bez managementu, ale přes údaje v MIBech bridge a qbridge. Z přepínače zjistíme informaci o 2 zařízeních na jednom portu a je nutné nakreslit mezi zařízení a přepínač další přepínač.

Taková topologie ale není zcela přesná, protože reálné zapojení je trochu jiné, protože telefon a přepínač je jedno zařízení. U podobných zařízení bude topologie vždy zkreslená. Nejedná se ale o kritickou chybu, protože se dotýká koncových zařízení.



Obr. 7: Ukázka zjištěné a reálné topologie

2.5.8 Link agregace

Jedná se o spojení více linek mezi přepínači do jedné virtuální. Zvyšuje se tím propustnost a odolnost proti výpadku. Propustnost je zvýšena v případě, že se v link agregaci aktivně používá více než jedna linka. Provoz je pak posílán dvěma nebo více linkami mezi přepínači. Rozložení zátěže mezi linky se provádí na základě zdrojových a cílových MAC nebo IP adres. Odolnost proti výpadku je možné zvýšit zapojením dalších linek, které ale data nepřenáší a jsou připravené pro případ výpadku aktivní linky. Z hlediska reálného zapojení jsou přepínače spojeny 4 linkami o rychlosti 1 Gb, z pohledu topologie se ale jedná o propojení jednou linkou o rychlosti 4x 1Gb *[11]*.

2.5.9 NIC teaming

Link agregace řeší zvýšení propustnosti mezi přepínači, NIC teaming se používá podobně, ale na straně serverů. Více síťových karet serveru je zapojených více kabely do přepínače, ale navenek se jeví jako jedna síťová karta. Její maximální rychlost je dána součtem rychlostí jednotlivých linek. Opět dochází k rozkládání zátěže mezi jednotlivé karty.

2.6 Přehled existujících aplikací

V případě sítě postavené na zařízeních jediného výrobce je většinou možné použít pro správu sítě software daného výrobce (např. HP iMC), který nám nabídne vysoký komfort. V některých případech částečně podporují i zařízení jiných výrobců alespoň v režimu čtení informací. Tento sw většinou také využívá protokol SNMP a zároveň zjišťování údajů jinými cestami (telnet, ssh). Poskytne nám mnoho informací. Jeho nevýhodou bývá ale složitost, nutnost instalace a cena.

Protože zařízení podporují standardní protokol SNMP, je další možností využít pouze ten a jiný management software (např. Cacti, Nagios). Takový software většinou neumožňuje konfiguraci, ale pouze vyčítání informací. Níže je uvedena tabulka často používaných nástrojů.

Software	Výhody	Nevýhody
HP iMC	Možnost konfigurace, podporuje různé výrobce, notifikace, využívá i CLI	Velká instalace (vyžaduje SQL a Win server), cena, složitost
Cisco network assistant	Možnost konfigurace, update firmwaru, využívá i CLI	Instalace, není jeden sw pro všechny Cisco zařízení
Nagios	Zdarma, notifikace, reakce na události	Neumí topologii
Cacti	Zdarma, notifikace	Pouze notifikace, neumí topologii
Solarwinds (více sw)	Nízká cena	Více samostatných aplikací

Tabulka 1: Přehled používaných nástrojů

2.7 Metodika a cíle práce

Byla provedena analýza existujích řešení. Při ní bylo zjištěno, že pro rychlé zmapování sítě tyto nástroje nemusí být vždy vhodné kvůli nutnosti instalace, ceně apod. Dále z analýzy vyplynuly požadavky na nový nástroj pro zjišťování topologie sítě. Dle zjištěných požadavků bude vytvořen návrh aplikace. Tento návrh bude rozšířen o funkci jednoduchého monitoringu. Dle tohoto návrhu bude vytvořena aplikace. Ta bude následně testována v různých sítích a výsledky budou porovnávány s výstupy výše zmíněných aplikací a reálným zapojením sítě.

2.8 Návrh aplikace

Tvorba nástroje se skládá ze dvou důležitých částí. První je modifikace operačního systému FreeBSD pro běz z flash disku. Samotné spuštění systému by ale nestačilo, proto je systém částečně modifikován a překonfigurován. Pro běh aplikace je nutné nainstalovat některé nástroje, které běžně distribuovaný systém neobsahuje a po startu systému musí být automaticky spuštěn www a databázový server s připravenou databází.

Druhou částí je vytvoření aplikace, která volá systémové programy a další utility, výsledky ukládá do databáze a vše prezentuje pomocí www stránek. Aplikační část je rozdělena do 4 samostatných částí.

2.8.1 Nastavení

Je to první část, s kterou se uživatel setká a umožní mu nastavit základní parametry zobrazování, způsob objevování a další volby. Některé z nich je možné změnit i v aplikaci.

2.8.2 Hledání aktivních prvků a zkoumání jejich okolí

Zde je uživateli nabídnut formulář pro zadání rozsahu prohledávaných IP adres a přihlašovacích údajů pro SNMP. Po odeslání jsou IP adresy zkoumány a v případě funkčního SNMP je prvek prozkoumán a zobrazen se základními informacemi v tabulce. Dále je zde zobrazován log, do kterého jsou průběžně zaznamenávány události jako nález nového zařízení, počátky a konce prohledávání apod.

2.8.3 Zobrazeni topologie

Po nalezení všech zařízení je ze zjištěných dat zobrazena mapa sítě. Mapu je umožněno přibližovat a oddalovat, prvky je možné posunovat. U každého zařízení je možné zobrazit detailní informace. Zobrazení lze ovlivnit několika přepínači, je možné například vypnout legendu nebo zobrazování některých typů zařízení.

2.8.4 Monitoring

Periodicky se nalezených zařízení doptává na změny a pokud nějaké nastanou, upozorní na to v předchozí obrazovce s mapou. Upozorňuje např. na smyčku v síti, změnu stavu portu nebo větší množství chyb na portu. Spouští se každých 5 minut.

2.9 Použité technologie

PHP - skriptovací jazyk určený hlavně pro tvorbu www stránek. Protože je možné jej pustit i jako shell skript, naprogramoval jsem v něm i monitoring, zkoumání IP, zjišťování otevřených portů a další. Výhodou je, možnost použití stejných funkcí, připojení k databázi apod.

FreeBSD - serverový operační systém. Pro mou práci je vhodný kvůli snadné modifikaci. Při práci aplikace na zjišťování topologie často z PHP volá některé jeho příkazy a utility jako například arp pro vypsání ARP tabulky.

Pseudoparalelní zpracování - při prohledávání více zařízení je seriový chod nepraktický,

proto jsem se rozhodl pro paralelizaci. Nejsnadnější cestou se mi jeví spustit PHP skript na pozadí a nechat rozdělení zdrojů přímo na operačním systému.

Apache web server - nejrozšířenější www server. Výhodou je snadná konfigurace a snadnost použití PHP.

MySQL - nejčastěji používaná databáze při tvorbě www stránek. Výhodou je snadná instalace a jednoduché zálohy nebo obnovy dat a snadná práce z příkazové řádky.

Sudo - program, který umožní spustit program s vyššími právy než má uživatel. V některých případech potřebuji vytvořit v PHP ICMP socket a to bez vyššího oprávnění není možné. Proto je nainstalováno sudo a uživatel www má práva jako uživatel root.

Ajax - moderní webová technologie, která umožňuje změnu obsahu stránky bez nutnosti načtení celé stránky. Samotné znovu načtení stránky není při dnešních odezvách www serveru problém, nepříjemně se ale projeví překreslování celé stránky. Proto jsem se rozhodl AJAX použít a načítat jím logy, nalezená zařízení a další.

SVG - vektorový grafický formát zapsaný pomocí XML. Výhodou je jeho podpora ve všech nejrozšířenějších prohlížečích, lze jej stylovat pomocí CSS a podporuje Javascript. Velkou výhodou jsou transformace objektů (posuny, rotace), proto je vhodný pro vykreslení topologie sítě. Použití transformací mi usnadňuje pozicování objektů.

Javascript – skriptovací jazyk, je použitý při práci s SVG a umožňuje interaktivitu s obrázkem - přesouvání objektů, zobrazení detailních informací apod.

3 Praktická část

3.1 SNMP – vyčítání informací z konkrétních MIBů

Každé zařízení může používat jeden nebo více MIBů. V případě některých (např. LLDP) jsou požadované informace pouze v rámci tohoto jendnoho MIBu. V jiných případech (např. naučené MAC adresy na portech) se mohou vyskytovat ve více MIBech. Někdy je dokonce nutné zkombinovat informace z více MIBů, rozdíly v uložení informací se liší i v rámci jednoho výrobce, proto je nutné doptávat se na všechna OID zmíněné v teoretické části práce. U některých prvků to může znamenat, že získám duplicitní informace přes různé MIBy.

Jak bylo uvedeno v teretické části – při zjišťování dat ze SNMP tabulek může být důležitý i index tabulky, který může obsahova informace např. o MAC adresách, číslech portů apod. Proto se v PHP používají dvě různé funkce.

Ukázka č. 6 – výpis snmpwalk:

print_r(snmpwalk("localhost", "public", "IF-MIB::ifName"));

Array ([0] => STRING: vr0 [1] => STRING: lo0)

Ukázka č. 7 – výpis snmprealwalk:

snmprealwalk – vrací pole, které obsahuje i index každé hodnoty

print_r(snmprealwalk("localhost", "public", "IF-MIB::ifName"));

Array ([IF-MIB::ifName.1] => STRING: vr0 [IF-MIB::ifName.2] => STRING: lo0)

3.1.1 LLDP MIB

Tento MIB je poměrně obsáhlý a poskytuje mnoho důležitých informací. Je zde podstrom, kde je uvedena konfigurace LLDP protokolu. Z té přes OID 1.0.8802.1.1.2.1.3.2.0 zjišťujeme LLDP ID zařízení. Odpovídá nejnižší MAC adrese zařízení obsažené v BRIDGE MIB přes OID 1.3.6.1.2.1.17.1.1.0 a je použito i u STP protokolu. Tuto MAC adresu bude použita jako identifikátor zařízení.

Dále je zde tabulka lokálních portů dostupná přes OID 1.0.8802.1.1.2.1.3.7. Součástí indexu tabulky je index portu. Tento index se ale nemusí rovnat indexu portu v tabulce Bridge MIB (1.3.6.1.2.1.17.1.4.1). Přes něj je provedeno párování do následující tabulky, která obsahuje naučené okolí.

Ta je dostupná přes OID 1.0.8802.1.1.2.1.4.1.1. Konkrétně se ptáme na OID 1.0.8802.1.1.2.1.4.1.1.5, které obsahuje LLDP ID (výše zmíněná MAC adresa) protistrany. Předposlední číslo indexu tabulky je opět index portu odpovídající tabulce lokálních portů výše. LLDP ale posílá i informaci o portu protistrany. Ta je uložená v OID 1.0.8802.1.1.2.1.4.1.1.7. Zde ale může být uložena rozdílná informace, např. index portu nebo jeho popis. Jaký typ informace je uložen říká OID 1.0.8802.1.1.2.1.4.1.1.6. Pokud obsahuje hodnotu 7, bude uložen index portu, 5 znamená název portu atd. OID 1.0.8802.1.1.2.1.4.1.1.8 obsahuje vždy název portu. Vývojový diagram viz. příloha B.

3.1.2 STP MIB

STP používá jako ID nejnižší MAC adresu zařízení uloženou v OID 1.3.6.1.2.1.17.1.1.0. STP MIB opět obsahuje informace o konfiguraci STP protokolu a tabulku se stavem portů. Jediným indexem tabulky je index portu, který odpovídá indexu portu v IF-MIBu v tabulce portů (OID 1.3.6.1.2.1.2.2.1).

Z celé tabulky je zajímavý sloupec 1.3.6.1.2.1.17.2.15.1.8. Ten obsahuje buď svou nejnižší MAC adresu, pokud je prvek root bridge nebo MAC adresu protistrany, která vede k root bridge pro síťový segment. Proto jsou zajímavé pouze záznamy, kde je uložena jiná MAC adresa než nejnižší MAC tohoto prvku. Pokud takový záznam najdeme, s pomocí indexu se doptáme na jméno portu přes Bridge MIB a IF-MIB. Vývojový diagram viz. příloha C.

3.1.3 CDP MIB

CDP cache je dostupná přes OID 1.3.6.1.4.1.9.9.23.1.2.1.1. Jsou zde informace o VLAN, MTU, a dalších. Protože zkoumáme pouze IPv4, ptáme se na hodnoty OID 1.3.6.1.4.1.9.9.23.1.2.1.1.3, které obsahuje typ naučené adresy. Hodnota 1 znamená IPv4, 20 je IPv6. OID 1.3.6.1.4.1.9.9.23.1.2.1.1.7 obsahuje název portu protistrany, OID 1.3.6.1.4.1.9.9.23.1.2.1.1.4 pak jeho IP adresu. Index tabulky tvoří index portu a index zařízení protistrany. Pomocí indexu portu se doptávám do IF-MIBu na jméno lokálního portu. Celým indexem zjišťuji přes OID 1.3.6.1.4.1.9.9.23.1.2.1.1.4 IP adresu protistrany zapsanou v šestnáctkové soustavě. Vývojový diagram viz. příloha D.

3.1.4 BRIDGE MIB

Bridge MIB mimo jiné obsahuje informace o naučených MAC adresách na jednotlivých portech. Tabulka je dostupná přes OID 1.3.6.1.2.1.17.4.3.1.1. Indexem této tabulky je naučená MAC adresa zapsaná v desítkové soustavě. Druhý sloupeček tabulky (OID 1.3.6.1.2.1.17.4.3.1.2) obsahuje číslo lokálního portu. Přes toto číslo se doptáváme do tabulky se seznamem všech bridge portů. Pomocí OID 1.3.6.1.2.1.17.4.3.1.2 zjistíme index rozhraní a pomocí něj už se můžeme doptat na název portu do IF-MIBu. Vývojový diagram viz. příloha E.

3.1.5 QBRIDGE MIB

Naučené MAC adresy na portech mohou být uloženy i v QBRIDGE MIBu. Je to méně typické, ale třeba přepínače H3C využívají právě tyto tabulky a ne BRIDGE MIB. Struktura tabulky je podobná jako u BRIDGE MIBu, jen je dostupná přes jiné OID, a to 1.3.6.1.2.1.17.7.1.2.2.1.2. Indexem tabulky je opět MAC adresa zapsaná v desítkové soustavě společně s číslem portu. Protože víme číslo portu, zbytek už je shodný jako u BRIDGE MIBu.

3.1.6 IF (Interface) MIB

Přes OID 1.3.6.1.2.1.2.2 je dostupná tabulka síťových rozhraní zařízení. Už byla popsána v teoretické části. Z této tabulky používáme všechny informace. Ve fázi zjišťování topologie čteme informace jako např. název portu, rychlost a popis. Při monitoringu sítě pak ostatní informace jako je množství prošlých paketů, chyb na portu apod.

3.1.7 RFC1213 MIB

Je rozšířením RFC1156 MIB, který se zabývá TCP/IP. Obsahuje mimo jiné tabulky IP a MAC adres. Dostupné jsou přes OID 1.3.6.1.2.1.4.20 a OID 1.3.6.1.2.1.2.2. Indexem obou tabulek je index portu a tím spoji MAC a IP adresu s portem.

Další tabulka má OID 1.3.6.1.2.1.4.22 a jsou v ní uložena ARP tabulka. Součástí indexu tabulky je i index prvku. Podobná tabulka je OID 1.3.6.1.2.1.3.1.1.2. Ta pochází z RFC1156 a některá zařízení ji stále používají.

3.1.8 PRINTER MIB

Dostupný přes OID 1.3.6.1.2.1.43.5.1.1.1.1. Používáme jej jen pro určení, zda se jedná

o tiskárnu. Pokud zařízení na dotaz do tohoto MIBu odpoví, pokládáme jej za tiskárnu.

3.2 Databázový návrh

Databáze je poměrně jednoduchá, obsahuje pouze 10 tabulek, neobsahuje žádné serverové procedury, triggery nebo jiné pokročilejší databázové postupy. Seznam a popis tabulek:

- **discovery_devices** - obsahuje informace o všech nalezených zařízeních, SNMP přihlašovací údaje, seznam podporovaných MIBů a switch_id, které se používá jako identifikátor zařízení u discovery protokolů. Může obsahovat i DNS název, rychlosti odpovědí a zjištěný typ zařízení
- **ports** - ze zařízení podporujících SNMP IF-MIB se zde ukládají informace o všech portech. Je zde např. index portu, název, status, rychlost a důležitý údaj - MAC adresa portu.
- **neighbourhood** - obsahuje všechny nalezené hrany a MIB, kterým byla hrana naučena.
- **log** - jednoduchá tabulka pro zaznamenávání informací o procesu prohledávání zařízení a monitoringu
- **mac_vendor** - předem naplněná tabulka. Je to seznam rozdělení MAC adres jednotlivým výrobcům
- **settings** - předem vyplněná tabulka s nastaveními
- **style** - zde se při vykreslování ukládají informace o aktuálním umístění objektů a jejich stylů. Využívají se v SVG zobrazovaní
- **open_ports** - obsahuje informace o otevřených portech jednotlivých zařízení
- **mon_devices** a **mon_ports** - podobné tabulky jako jsou discovery_devices a ports. Zde se ukládají informace pro monitorig společně s časovým údajem

3.3 Algoritmus prozkoumání zařízení

Jedná se o llgoritmus pro prozkoumání jedné IP adresy. Tento proces je spuštěn z PHP jako úloha na pozadí. Nejprve zjistí informace o zařízení jako je podpora SNMP, jeho DNS

jméno a odpověď na ping.

V případě, že zařízení komunikuje přes SNMP, je zahájeno zjišťování informací o jeho portech a naučeném okolí. Na závěr je pro každou nalezenou a dosud neprozkoumanou IP adresu, zjištěnou z tohoto prvku, spuštěn stejný proces zkoumání. Ten je opět spuštěn na pozadí. Vývojový diagram viz. příloha F.

3.4 Algoritmus vykreslení sítě

Při vykreslení se z databáze vyberou všechna zařízení. U každého se prohledá tabulka style, zda již nejsou určené jeho souřadnice z dřívějšího vykreslení. Pokud ne, vygeneruje se jeho náhodné umístění. Tyto údaje se ukládají do pole nodů. Poté se vyberou z tabulky neighbourhood všechny hrany naučené přes LLDP. U všech se zjistí rychlost, popis portu a uloží se do pole hran. Při ukládání se zkoumá, zda již v poli není uložena opačně orientovaná hrana. V takovém případě se nová ignoruje. Stejně se postupuje i případě hran naučených dalšími způsoby (STP, CDP, ..), ale před uložením hrany do pole hran se zjišťuje, zda už není stejná hrana naučena a zaznamenána některým z předchozích protokolů. Pokud ano, je pouze doplněn popisek hrany o tento protokol. Když nebyla nalezena, je uložena jako v předchozích případech. U hran naučených z MIBů BRIDGE a QBRIDGE ještě ignorujeme hrany, kdy je koncové zařízení naučeno přes UPLINK hranu některého z protokolů LLDP, CDP nebo CDP. V takovém případě totiž přepínač ví o zařízení, které je ale fyzicky připojeno k jinému přepínači a vykreslení takové hrany by nedávalo smysl. Nakonec se vykreslí všechny hrany z pole hran a všechny nody z pole nodů.Vývojový diagram viz. příloha G.

3.5 Algoritmus monitoringu

Monitoring již nevykonává opětovné zkoumání SNMP zařízení ani zkoumání nových. Pouze na stávajících monitoruje události a změny jako je třeba změna topologie ohlášená spanning tree protokolem. Každých 5 minut se spustí, načte si seznam SNMP zařízení a ty prozkoumá. Do databáze ukládá stavy portů, rychlost, přenesená data a chyby na portech. Dále vyčte změny CDP, LLDP a STP protokolů. I tyto údaje uloží s časovým údajem do databáze. Tyto údaje potom čte skript volaný Ajaxem a hledá změný oproti předchozímu stavu. Pokud ke změně došlo, je to indikováno v obrazovce se zobrazenou topologií. Vývojový diagram viz. příloha H.

3.6 Úprava OS na spuštění z flash disku

Pro vytvoření nástroje jsme vybrali operační systém FreeBSD. Je často používaný jako serverový operační systém a jeho výhodou je snadná modifikace a kompilace všech komponent dle požadavků. Další výhodou je jeho snadná instalace na flash disk.

Cílem práce je vytvořit obraz modifikovaného systému, který stačí přenést na flashdisk. Proto musí obraz řešit i zavádění systému. Metod, jak toho dosáhnout, je více. Použili jsem současnou produkční verzi 9.1 v 32 bitové verzi. 64 bitová by nám nepřinesla žádné zásadní výhody a neumožnila by spuštění na starších počítačích. Vytvoření obrazu se nejsnáze provádí přímo ze systému FreeBSD, proto jsem výše zmiňovanou verzi nainstaloval na PC. Postačuje základní instalace.

Potom je nutné zkompilovat jádro a celý operační systém, který se po kompilaci uloží do paměťového disku. Na tomto systému se provede konfigurace, instalace potřebných programů, aplikace NetSentinel a databáze. Nakonec je soubor přenesen na flash disk. Detailní popis instalace a konfigurace – příloha A.

3.7 Řešení očekávaných problémů

V teoretické části byl zmíněn seznam očekávaných problémů vycházejících ze zařízení ovlivňujících topologii sítě a jejich řešení. V případě virtuálních přepínačů nemám pomocí běžných SNMP MIBů žádnou možnost zjistit reálné fyzické zapojení. Otázkou je, zda by taková informace byla přínosem. Virtualizací přepínačů se snažíme vytvořit jeden virtuální a tomu odpovídá i námi zjištěná topologie. Podobně je tomu i u virtuálních přepínačů používaných k připojení virtuálních serverů. Výrobci postupně uvolňují SNMP MIBy nesoucí informace o virtuálních zařízeních. Protože se jedná ale o proprietární řešení, nebyly tyto MIBy zkoumány.

V případě link agregace je situace snažší, protože SNMP obsahuje informace o samostatných linkách a zároveň i o agregaci. Je pouze nutné informace správně spojit. MAC adresa link agregace je rovna nejnižší MAC adrese portu, z kterých je agregace složena. To je nutné zohlednit při vykreslování takové hrany. Z hlediska zobrazení se link agregace neliší od běžných hran, jiná je pouze rychlost a popis. K vytvoření správného popisu k hraně dochází při kreslení diagramu, konkrétně při detekci duplicitních hran. Zpracování hran bylo vysvětleno již dříve.

U levných zařízení nebo u typů pouze se základním managementem je možné se setkal

s neúplnou podporou MIBu. V některých případech tak nemáme důležité informace jako je index portu, ale např. jen popis portu. Pomocí takové informace by v některých případech bylo možné vytvořit hranu odvozenou z části řetězce popisu portu. To by ale mohlo do zjištěné topologie zanášet nepřesnosti jako nové, reálně neexistující, hrany apod. Proto raději neúplné informace nevyužijeme.

Segmentace sítě pomocí VLAN mohla dle předpokladů způsobovat problém. Nakonec se ukázalo, že to tak není. Pokud se pomocí SNMP dostaneme přes jakékoli IP rozhraní přepínače, získáme stejné informace.

3.8 Řešení problémů zjištěných při vytváření aplikace

Při vytváření aplikace se objevily další problémy a okolnosti, které bylo nutné řešit. V dalších kapitolách jsou uvedena řešení nevětších problémů.

3.8.1 Špatná odpověď na SNMP dotaz

Při testování praktické části byla nalezena zařízení, které při neznalosti dotazovaného OID odpovídají nestandardním způsobem. SNMP response normálně obsahuje pole Request-ID, Error-status, Error-index a Variable-bindings.

V případě zpracování OID, které zařízení zná, se Error-status i Error-index rovná nule.

Ukázka č. 8 – správná odpověď a její návratová hodnosta

root@mail ~ # snmpget -On -v1 -c public 192.168.222.108
1.3.6.1.2.1.17.2.3.0.1.3.6.1.2.1.17.2.3.0 = Timeticks: (1250468684) 144 days, 17:31:26.84

root@mail ~ # echo $?
0

Když při zpracování dojde k chybě, je v Error-status uvedeno číslo chyby (např. 1 - odpověď by byla příliš velká, 2 - neexistující OID, ..) a v Error-index je vráceno chybné OID.

Ukázka č. 9 – správná odpověď při chybném dotazu
root@mail ~ # snmpget -On -v1 -c public 192.168.222.108 1.3.6.1.2.1.17.2.3.0.6.7.8

Error in packet

Reason: (noSuchName) There is no such variable name in this MIB.

Failed object: .1.3.6.1.2.1.17.2.3.0.6.7.8

root@mail ~ # echo $?

2

Problematické zařízení se ale chová tak, že i v případě neznámého OID nastaví Error-status a Error-index na nula a zprávu o neznámém OID vloží do odpovědi. Takže místo chyby dostáváme kladnou odpověď, která ale nemá nic společného s dotazovanou hodnotou.

Ukázka č. 10 – špatná odpověď

root@mail ~ # snmpget -On -v1 -c public 192.168.222.180
1.3.6.1.2.1.17.2.3.0.1.3.6.1.2.1.17.2.3.0 = String: OID not found

root@mail ~ # echo $?

0

Z tohoto důvodu je nutné po získání odpovědi provádět testování vrácené hodnoty, zda odpovídá očekávané hodnotě. Toto testování provádíme regulárními výrazy a testujeme například, zda vrácená hodnota odpovídá MAC adrese.

Ukázka č. 11– špatná odpověď

```
if ( preg_match ("/^([0-9A-F]{2}:){5}[0-9A-F]{2}$/", $mac))
```

3.8.2 Zjištění typu zařízení

Pro zjištění typu zařízení je vhodné použít program NMAP, který má mimo jiné funkci zjištění údajů o IP adrese, konkrétně typu zařízení, jeho operačního systému a verze. NMAP toto řeší pomocí detailního zkoumání odpovědí na TCP a UDP požadavky. V těchto odpovědích se zjišťují např. komunikační parametry jako je počáteční velikost TCP window size, podporované hlavičky a jejich pořadí. Výsledek je potom porovnán s databází již zjištěných "otisků operačních systémů".

Ač je NMAP patrně nejlepší program pro detekci OS, jeho výsledky jsou v některých případech značně nepřesné. U HP tiskárny NMAP zobrazil více naprosto rozdílných výsledků - OS Windows, Unix, Linux. Pokud NMAP systém rozezná, je výsledek snadno parsovatelný, v případě nerozhodnosti je ale výsledek zobrazen špatně formátovaný a jeho následné zpracování je složitější.

V případě výše zmíněné tiskárny přitom stačí pouze jeden SNMP dotaz na Printer MIB. Druhým problémem je doba detekce NMAPem, která v některých připadech trvá několik minut a výsledkem může být opět nejednoznačná odpověď. Časová náročnost by umožnila pouze jeho spouštění na pozadí nezávisle na zkoumání zařízení a v případě jednoznačné odpovědi upřesnění typu.

Jednodušší, časově méně náročná detekce, ale také ne zcela přesná je zkoumání otevřených portů v kombinace se SNMP dotazy, jak již bylo popsáno v předchozím odstavci. Vytvoříme seznam nejčastěji používaných službeb a zkoumáme porty, které používají, např. Netbios, ssh, www, DNS, ...

Dále musíme vytvořil pravidla, která odpovídají jednotlivých systémům. Otevřené porty Netbios nebo portu 445 většinou značí OS Windows. Otevřený portem 22 (ssh) značí, že se pravděpodobně jedná o Linux apod. Protože detekce OS není nejdůležitější, můžeme si takové zjednodušení dovolit.

Setkat se můžeme i se zařízeními, které nedetekuje NMAP vůbec a zkoumání otevřených portů mylně detekovalo typ zařízení. Například 3com NBX IP telefon má SNMP, ale nepoužívá žádný MIB. Proto na každý SNMP požadavek odpovídá, že dané OID nezná. Zároveň má ale otevřený UDP port 135, který používá sdílení Microsoft. Proto takové zařízení budeme identifikovat špatně.

3.8.3 Špatná nebo neúplná implementace SNMP MIBů STP, LLDP, ...

U levných zařízení pouze se základním managementem nebývá vždy ani implementace konkrétního MIBu úplná, popřípadě obsahuje chyby. Setkat se můžeme s levnými přepínači, které při LLDP oznámeních zaměňují hodnoty index a popis portu. Pro taková zařízení by bylo nutné řešit výjimky a algoritmus hledání topologie by se stal složitým. Proto někdy můžeme získat informace o hranách, které ale není možné jednoduše použít.

Nejmenším problémem se ukázala zařízení, která SNMP podporují, ale neobsahují žádný MIB. Nezískáváme z nich žádné výsledky, ale dotazování takových zařízení zdržuje.

3.9 Ověření systému v praxi

Při testování funkčnosti byla aplikace testována od malých po středně velké sítě. Testování proběhlo na reálných sítích. Výsledky programu NetSentinel byly porovnány s topologií zobrazenou nástroji uvedenými v teoretické části a dále pak i se skutečným stavem.

3.9.1 Testované sítě

- **Síť 1, malá firemní síť -** jedná se o síť s přibližně 60 koncovými zařízeními, přepínače jsou dva, z toho jeden je stoh sestavený ze 3 přepínačů. Celkový počet fyzických portů je přibližně 100. Prvky jsou značek HP a H3C, síť obsahuje několik Unix a Windows serverů, print serverů a IP telefonů.
- **Síť 2, škola -** síť je postavena na levných přepínačích se základním managementem. Prvky jsou značky HP a Allied telesis, celkem asi 120 portů. V síti je zapojeno několik IP telefonů.
- **Síť 3, výrobní firma -** jedná se o větší síť postavou výhradně na prvcích Cisco. Síťová topologie je hvězda, pateřním prvkem je stoh dvou přepínačů. Z pateřního je pomocí linkových agregací zapojeno 6 přepínačů, celkem asi 240 portů, síť je segmenovaná pomocí VLAN.
- **Síť 4, nemocnice -** jedná se o středně velkou síť s přepínači značek 3com, H3C a HP. Aktivních prvků je více než 20, jsou zde i přepínače bez managementu. V síti je zapojeno mnoho různých lékařských zařízení, dále VPN koncentrátor a několik routerů. Základem topologie je hvězda, z centrálního prvku jsou okolní přepínače připojené optickými kabely. Síť pokrývá několik budov.

3.9.2 Testování aplikace, porovnání se skutečným stavem

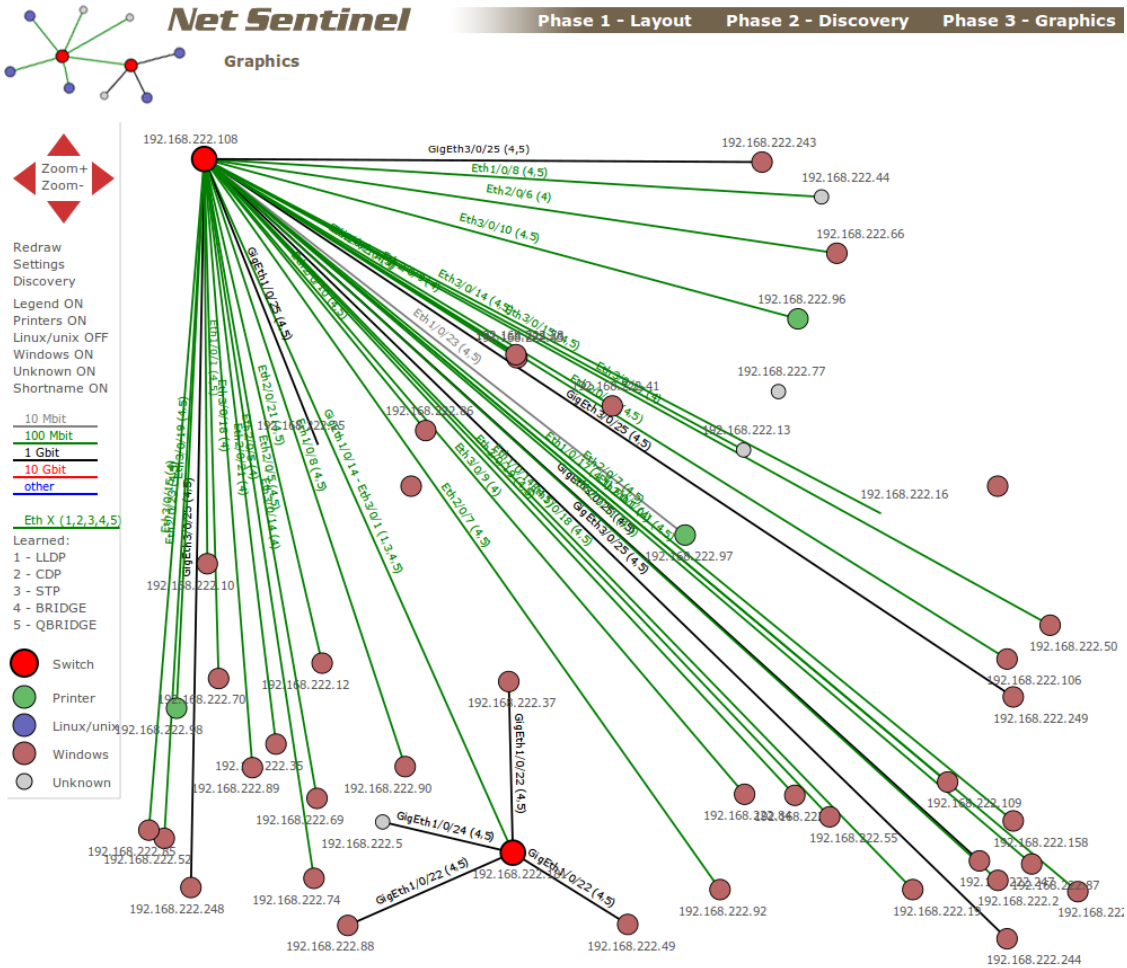
Aplikaci byla testována na všech 4 výše zmíněných sítích. Doba prohledávánní sítě se liší v závislosti na množství nalezených zařízení a jejich konfiguraci. Pokud je na zařízení firewall, který na testování otevřených TCP portů neodesílá informaci o filtrování komunikace, testovací skript čeká na timeou a doba zkoumání se prodlužuje. Podobné chování je i u UDP služeb. Pokud zařízení na daném portu neposlouchá, má odeslat ICMP odpověď. Pokud je nakonfigurované jinak, odpověď se nepošle a skript čekání ukončí až po stanoveném limitu. Díky spouštění testů na pozadí není doba prohledávání sítě příliš dlouhá.

Při práci s aplikací NetSentinel se můžeme setkat s nepříjmenou vlastností, která se projevuje

u velkého množství zařízení. Při každém přetažení objektu se znovu generuje SVG, to znamená hledání všech hran, vynechání duplicit apod. Je nutná další optimalizace při vykreslování a pravděpodobně ještě před prvním vykreslením. Kdybych předem odstranil duplicity, ubylo by průchodů cyklů při hledání hran asi až o 50%.

Síť 1 - malá firemní síť

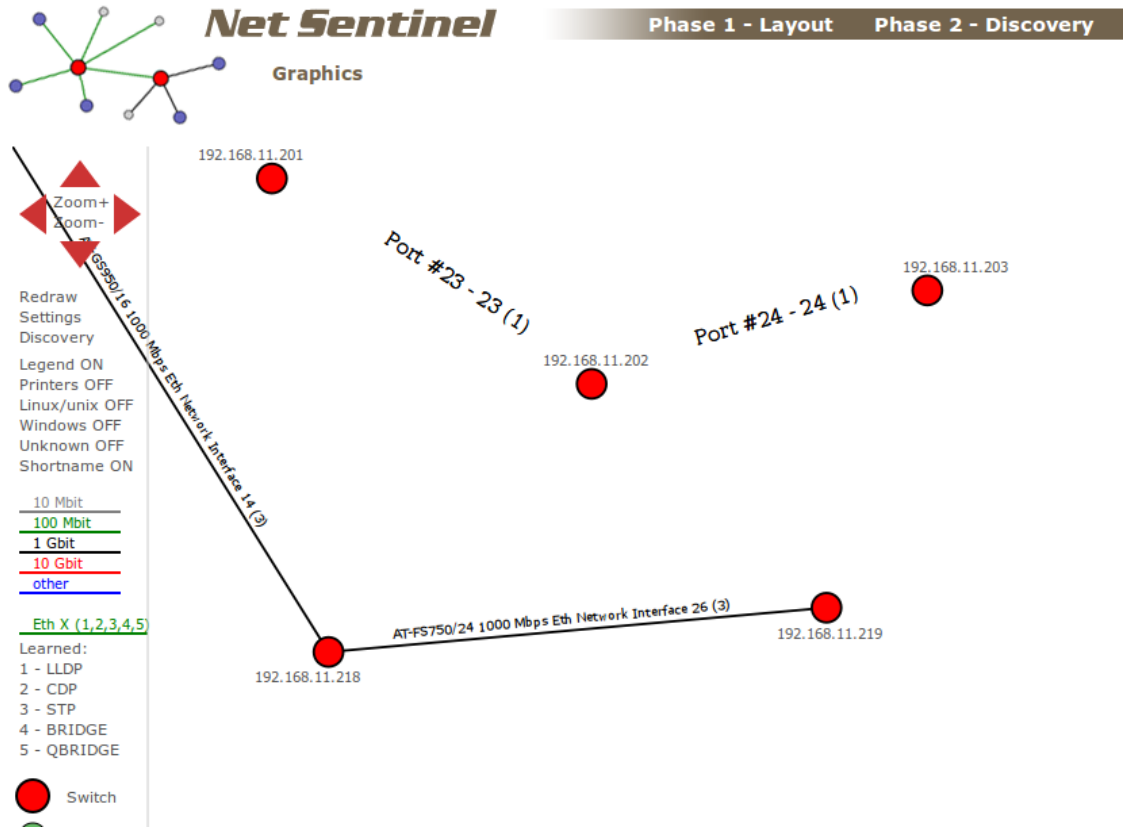
Doba zjišťování síťové topologie trvá v řádech jednotek minut. Správně jsou nalezeny oba přepínače a několik desítek zařízení. Vykreslení SVG je rychlé, periodický monitoring běží vždy jen pár vteřin a odezvy aplikace jsou v normálu.



Obr. 8: Topologie získaná aplikací NetSentinel – malá firemní síť

Síť 2 - škola

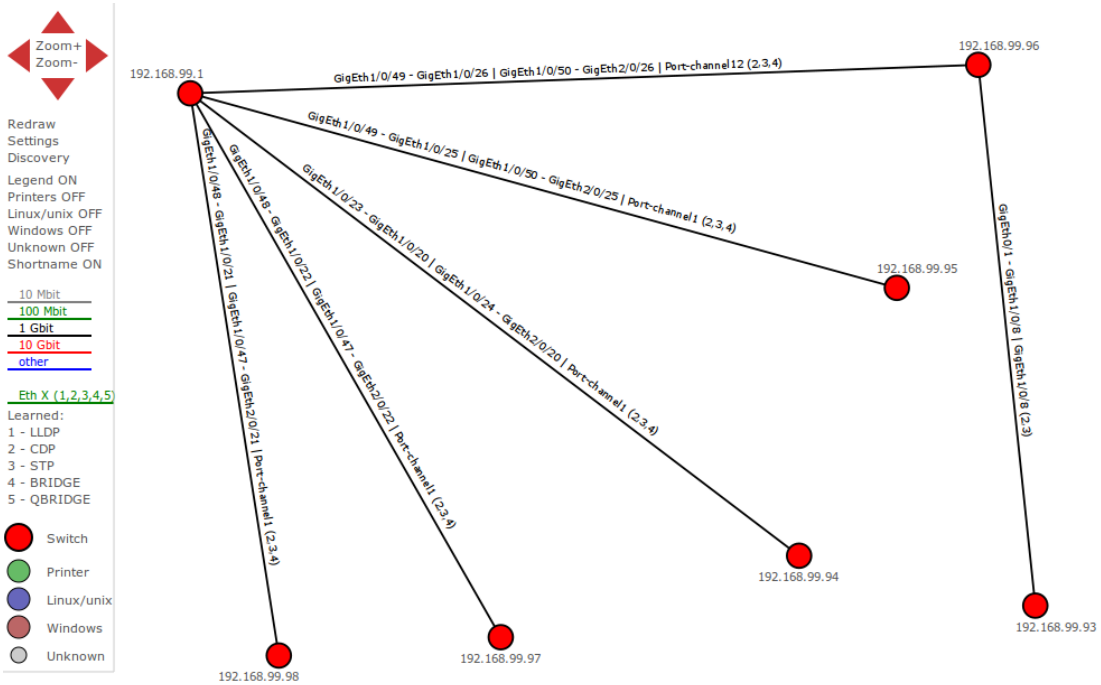
Zde byla doba zjišťování kratší z důvodu pouze základního maangementu přepínačů. Část přepínačů podporuje pouze LLDP a část STP. Proto není vykreslena ani celá topologie na úrovní přepínačů. Není možné zjistit ani koncové stanice, protože přepínače nemají přes SNMP dostupnou CAM, ARP ani IP tabulku. Kvůli málo zařízením je vykreslení, monitoring i práce s SVG rychlá.



Obr. 9: Topologie získaná aplikací NetSentinel - škola

Síť 3 - síť ve výrobní firmě

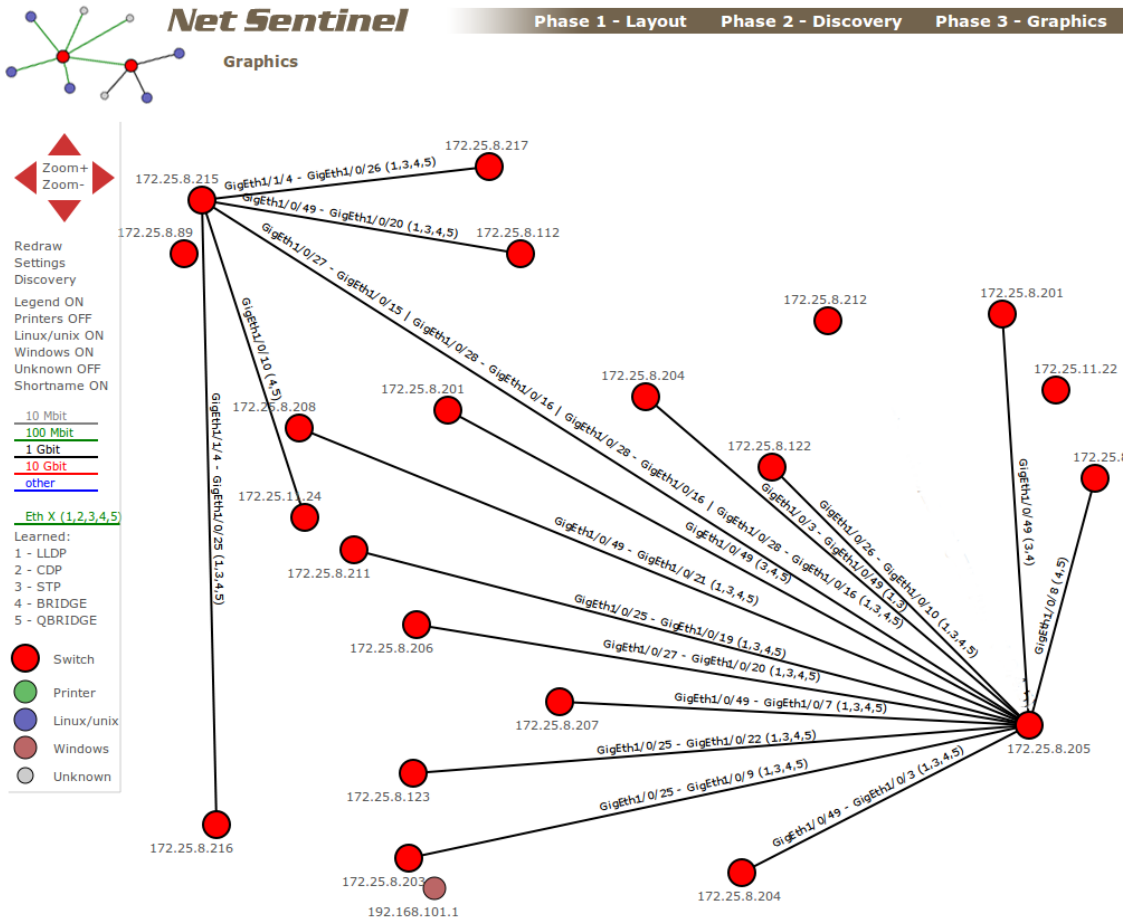
Zde zjištění sítě trvalo asi 15 minut. Zobrazená topologie odpovídá realitě. Nebyla nalezena část koncových zařízení v jiných VLAN sítích. Byl nalezen i jeden VRRP router sestavený ze dvou fyzických routerů. To odpovídá předpokladu z kapitoly očekávaných problémů. Vykreslení mapy trvá kolem 3 – 5 vteřin.



Obr. 10: Topologie získaná aplikací NetSentinel – síť ve výrobní firmě

Síť 4 - nemocnice

Prohledání sítě trvalo asi 20 minut. Byla nalezena celá topologie na úrovni přepínačů. U koncových zařízení je situace horší, pravděpodobně je to způsobeno částí přepínačů, které jako v případě školní sítě neumožňují vyčíst potřebné informace. Další část zařízení jsou prvkym podporující SNMP, ale nebylo možné zjistit SNMP community. Zde se již značně projevuje pomalé vykreslování SVG zmiňované výše.

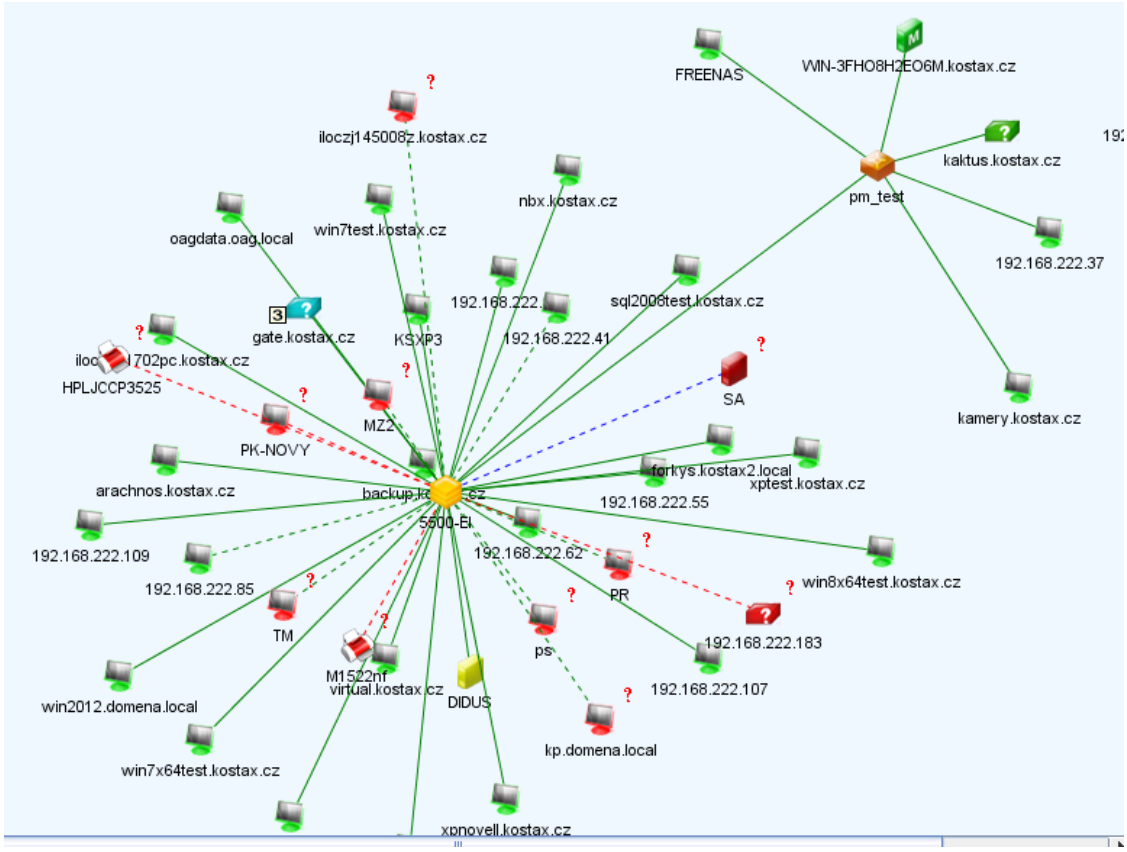


Obr. 11: Topologie získaná aplikací NetSentinel – síť nemocnice

3.9.3 Porovnání s HP iMC a Cisco network assistant

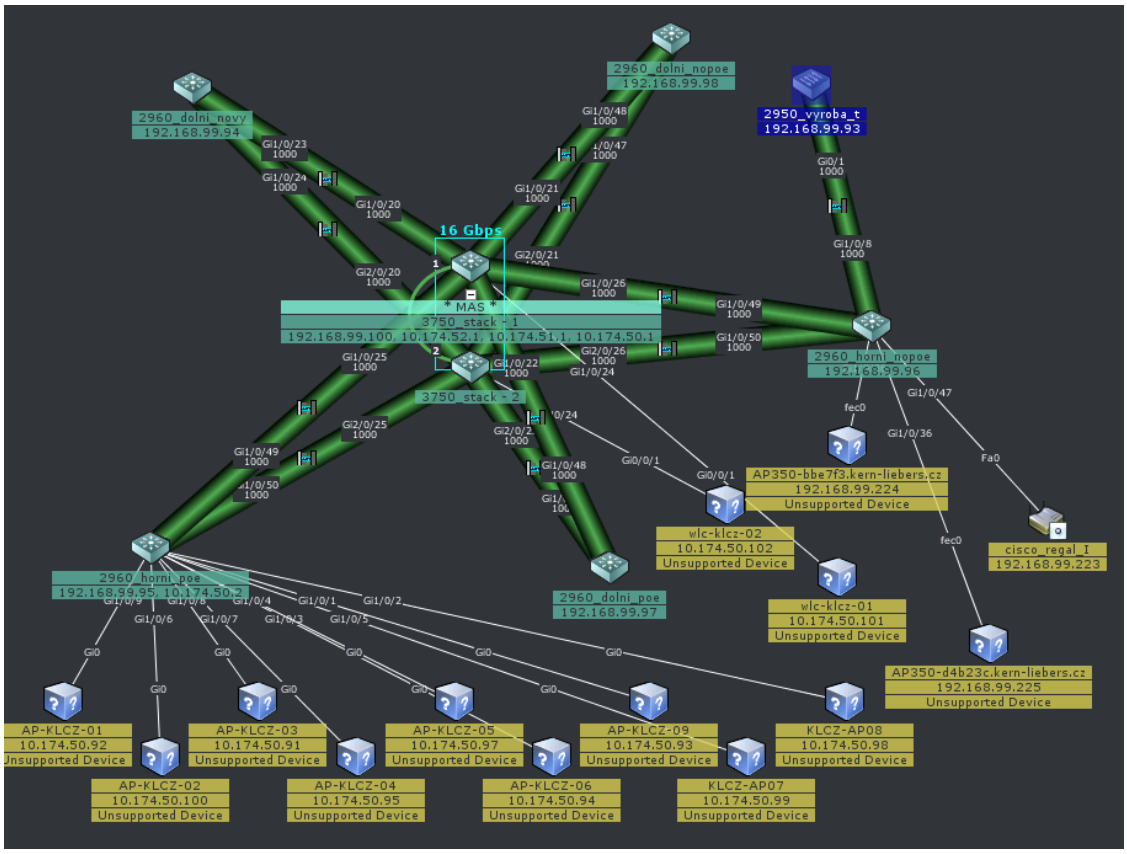
Při porovnání výsledků zohledňujeme pouze základní topologii, protože Cisco network assistant se nezabývá koncovými stanicemi. U větších sítí se obrázek stává nepřehledným a bylo by náročné ověřit velké množství zařízení.

HP iMC je profesionální komerční nástroj pro monitoring celé sítě včetně koncových zařízení. Pro zkoumání zařízení kromě SNMP využívá i přístup přes Telnet nebo SSH a získává tím podstatně více informací než aplikace NetSentinel. Pokud je o topologii, jsou zjištěné a zobrazené informace shodné v obou aplikacích.



Obr. 12: Zobrazení topologie nástrojem HP iMC – malá firemní síť

Druhé porovnání topologie je s aplikací Cisco network assistant. Porovnání proběhlo na síti č. 3 - výrobní firma. Tento program zobrazí shodnou základní topologii jako aplikace NetSentinel. Stejně jako v případě HP iMC zjišťuje údaje i z CLI a proto má navíc například informace o vytížení CPU.



Obr. 13: Zobrazení topologie nástrojem Cisco network assistant – síť ve výrobní firmě

4 Závěr

Cílem práce bylo vytvořit jednoduchý a snadno použitelný nástroj pro zjištění topologie sítě a její základní monitoring. Snadnou použitelnost vidím v možnosti spuštění aplikace z flash disku na libovolném počítači v síti., podporujícím spuštění systému z USB zařízení. To je jediná podmínka pro běh aplikace, nevyžaduje místo na pevném disku ani žádnou databázi. Protože aplikace řeší pouze topologii a jednoduchý monitoring, je její ovládní snadné a přehledné.

Po otestování aplikace na několika různých sítích si dovolím tvrdit, že cíl práce jsem splnil. Nalezené topologie s shodují s realitou i s výstupy z profesionálních NMS systémů jako je HP iMC.

Výsledný produkt ale není možné srovnávat s takovými programy, je určen pouze na základní seznámení se sítí a rychlé nalezení problémů.

Aplikaci se hodlám dále věnovat a rozvíjet ji. Je možné vylepšit např. detekci OS, lépe využít program NMAP a optimalizovat vykreslování. Výsledek chci nabídnout pod BSD licencí k volnému použití.

5 Seznam použitých zdrojů

KABELOVÁ, Alena; DOSTÁLEK, Libor. Velký průvodce protokoly TCP/IP a systémem DNS. Brno: Computer Press, 2008. ISBN 978-80-251-2236-5.

The FreeBSD Documentation Project. FreeBSD Handbook [online]. 2013 [cit. 2013-03-28]. Dostupné z: http://www.freebsd.org/doc/en/books/handbook/

[1] IEEE Computer Society. IEEE802.1ab: Station and Media Access Control Connectivity Discovery [online]. 2009 [cit. 2013-03-28]. Dostupné z: http://standards.ieee.org/getieee802/download/802.1AB-2009.pdf

[2] Cisco Systems. Cisco Discovery Protocol [online]. 2006 [cit. 2013-03-28]. Dostupný z:

http://www.cisco.com/en/US/tech/tk648/tk362/tk100/tsd_technology_support_s ub-pro tocol_home.html

[3] Microsoft. Plug and Play for Windows 2000 and Windows XP [online]. 2001 [cit.2013-03-28].

Dostupné z: http://msdn.microsoft.com/en-us/windows/hardware/gg463317

[4] Network Working Group. RFC 4861: Neighbor Discovery for IP version 6 (IPv6)

[online]. 2007 [cit. 2013-03-28]. Dostupné z: http://tools.ietf.org/html/rfc4861

[5] Network Working Group. RFC 5556: Transparent Interconnection of Lots of Links (TRILL): Problem and Applicability Statement [online]. 2009 [cit. 2013-03-28]. Dostupné z: http://tools.ietf.org/html/rfc5556

[6] Microsoft. Link Layer Topology Discovery Protocol Specification [online]. 2010 [cit. 2013-03-29]. Dostupné z:

http://msdn.microsoft.com/en-US/windows/hardware/gg463024

[7] IEEE Computer Society: IEEE 802.1D: Media Access Control (MAC) Bridges [online]. 2004 [cit. 2013-03-29]. Dostupné z:

http://standards.ieee.org/getieee802/download/802.1D-2004.pdf

[8] Network Working Group. RFC 792: Internet Control Message Protocol [online]. 1981 [cit. 2013-03-29]. Dostupné z: http://www.ietf.org/rfc/rfc792.txt

[9] MAURO, D. R., SCHMIDT, K. J. Essential SNMP. Sebastopol: O'Reilly Media, 2005. ISBN 978-0-596-00840-6.

[10] Network Working Group. RFC 3768: Virtual Router Redundancy Protocol (VRRP) [online]. 2004 [cit. 2013-03-29]. Dostupné z: http://tools.ietf.org/html/rfc3768

[11] IEEE Computer Society: IEEE 802.1AX: Standard for Local and metropolitan area networks – Link Aggregation [online]. 2008 [cit. 2013-03-29]. Dostupné z:

http://standards.ieee.org/getieee802/download/802.1AX-2008.pdf

6 Seznam příloh

Příloha A – příprava OS pro provoz z flash disku
Příloha B – vývojový diagram čtení LLDP MIB
Příloha C – vývojový diagram čtení STP MIB
Příloha D – vývojový diagram čtení CDP MIB
Příloha E – vývojový diagram čtení Bridge MIB
Příloha F – vývojový diagram zjišťování informací z jednoho SNMP zařízení
Příloha G – vývojový diagram vykreslení topologie
Příloha H – vývojový diagram monitoringu

Příloha A – příprava OS pro provoz z flash disku

Je nutné zaktualizovat tzv. Ports tree. Zde jsou informace o zdrojových kódech aplikací a způsoby instalace:

porstnap fetch extract

Nainstalovat aplikaci svnup. Ta je vhodná pro snadnou aktualizaci zdrojových kódů OS

cd /usr/ports/net/svnup
make install

Aktulizace zdrojů systému:

svnup -h svn0.us-west.freebsd.org -b base/releng/9.1 -l /usr/src

Dále je vhodné upravit konfiguraci kernelu, výchozí obsahuje mnoho nepotřebných věcí jako je podpora zvukových karet, PCMCIA apod. Proto si zkopírujeme výchozí konfigurační soubor kernelu do souboru MYKERNEL a ten upravíme:

cd /usr/src/sys/i386/conf
cp GENERIC MYKERNEL

Je potřeba upravený kernel zkompilovat

cd /usr/src
make buildkernel KERNCONF=MYKERNEL

Dále je potřeba zkompilovat operační systém

make buildworld

Vytvoříme soubor pro img o velikosti 2GB a z něho si vytvoříme paměťové zařízení.

dd if=/dev/zero of=./memstick_sentinel.img bs=1m count=2000
mdconfig -a -t vnode -f ./memstick_sentinel.img -u 0

Protože řešíme i bootování, vytvoříme diskový oddíl, boot sektor a nakonec oddíl

naformátujeme:

fdisk -iI /dev/md0
bsdlabel -wB /dev/md0s1
newfs -L FreeBSD /dev/md0s1a

Výsledný oddíl si připojíme do souborového systému:

mount /dev/md0s1a /mnt

Do tohoto oddílu přeneseme zkompilovaný kernel a celý OS:

make installkernel KERNCONF=MYKERNEL DESTDIR=/mnt
make installworld DESTDIR=/mnt

Je nutné upravit konfiguraci systému, nejjednodušší cestou asi je zkopírovat konfiguraci funkčního systému a tu jen upravit. Největší změnou je vytvoření oddílu /tmp a /var pouze v paměti. Omezí se tak časté zápisy na flash paměť.

cp /etc/* /mnt/etc

Pro instalaci aplikací potřebujeme Ports tree a adresář /dev, proto si oba připojíme:

mount -t devfs devfs /mnt/dev
mkdir /mnt/usr/ports
mount_nullfs /usr/ports /mnt/usr/ports

Potom už je možné instalovat aplikace. Protože systém ještě není spuštěn, je nutné provést změnu kořenového adresáře do vytvářeného OS pomocí chroot:

chroot /mnt /bin/sh
cd /usr/ports/www/apache22
make install
...

Po instalaci aplikací a konfiguraci systému musíme ukončit chroot:

exit

Nakopírovat aplikaci Net Sentinel do příslušného adresáře www serveru a stejným způsobem připravit i databázi.

Pro funkční aplikaci je nutné správné nastavení IP, DNS, případně routování. Abych uživateli usnadnil konfiguraci a nemusel ručně nastavovat údaje, vytvořil jsem jednoduchý shell skript, kterým jsem nahradil první virtuální terminál. Po spuštění systému je tedy uživateli zobrazena nabídka z tohoto skriptu a může vše nastavit bez znalosti konkrétních příkazů tohoto unixového systému. Náhrada se provádí v souborech /mnt/etc/gettytab a /mnt/etc/ttys.

Už zbývá pouze připravený systém odpojit a přenést na flash disk.

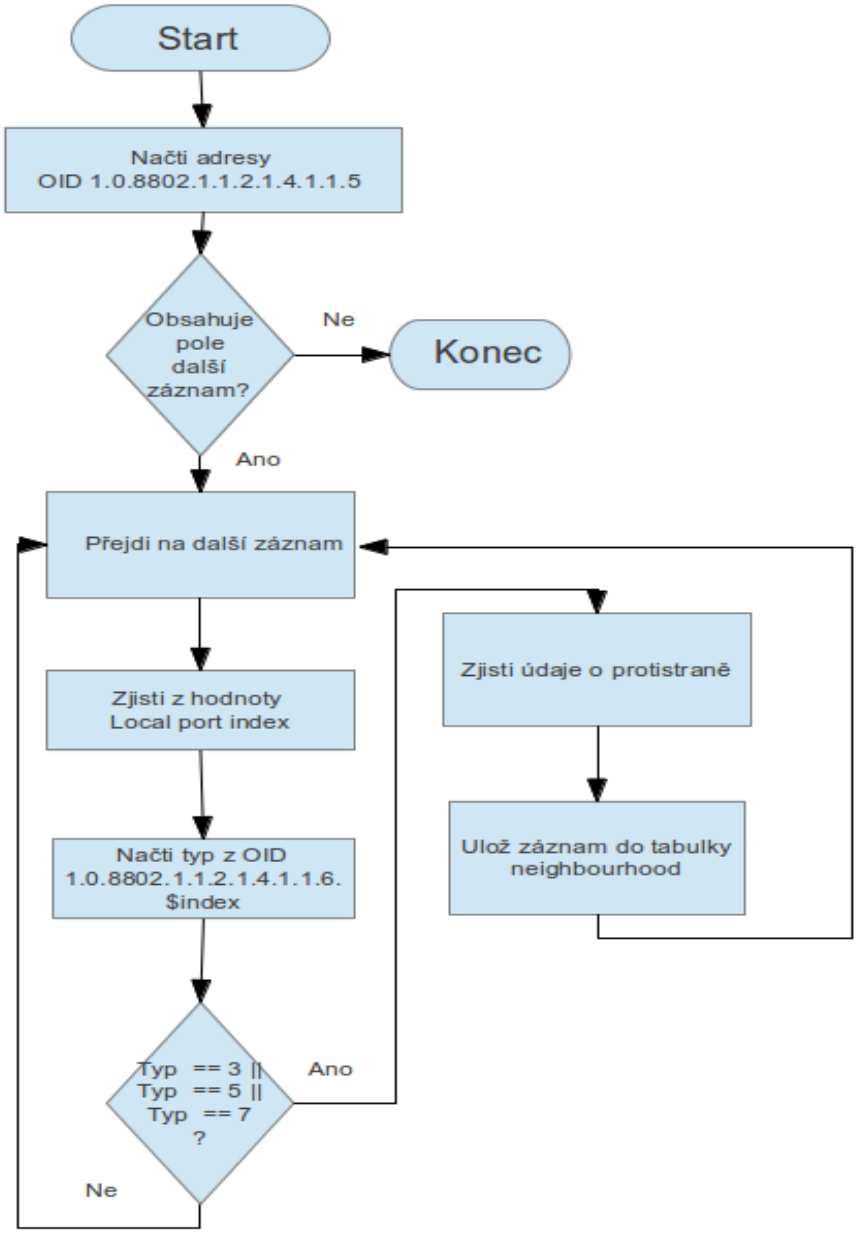
umount /mnt/dev
umount /mnt/usr/ports

umount /mnt

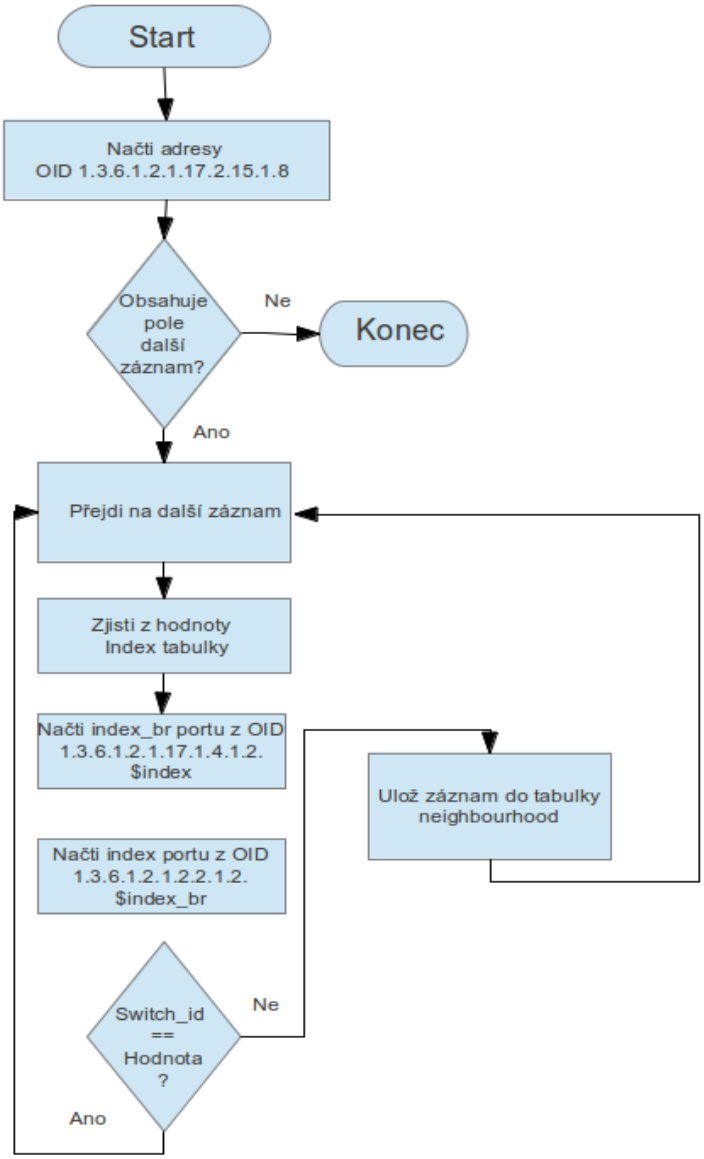
dd if=./memstick_sentinel.img of=/dev/da0 bs=1m

Tím jsme vytvořil flashdisk s opraveným operačním systémem a nainstalovanou aplikací NetSentinel.

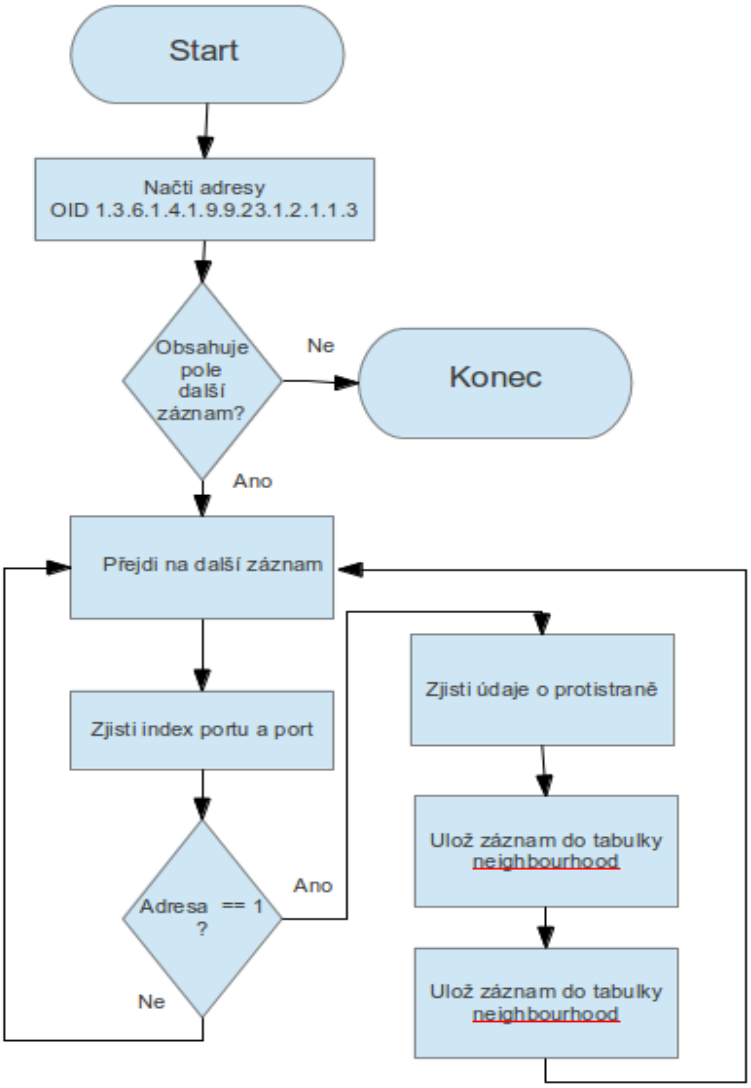
Příloha B – vývojový diagram čtení LLDP MIB



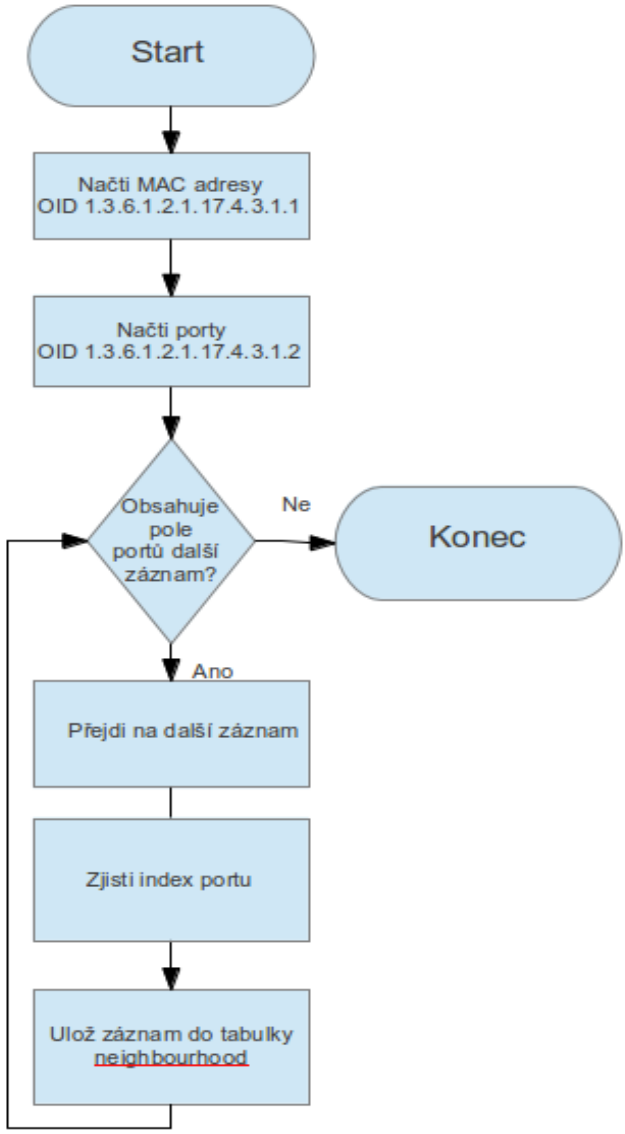
Příloha C – vývojový diagram čtení STP MIB



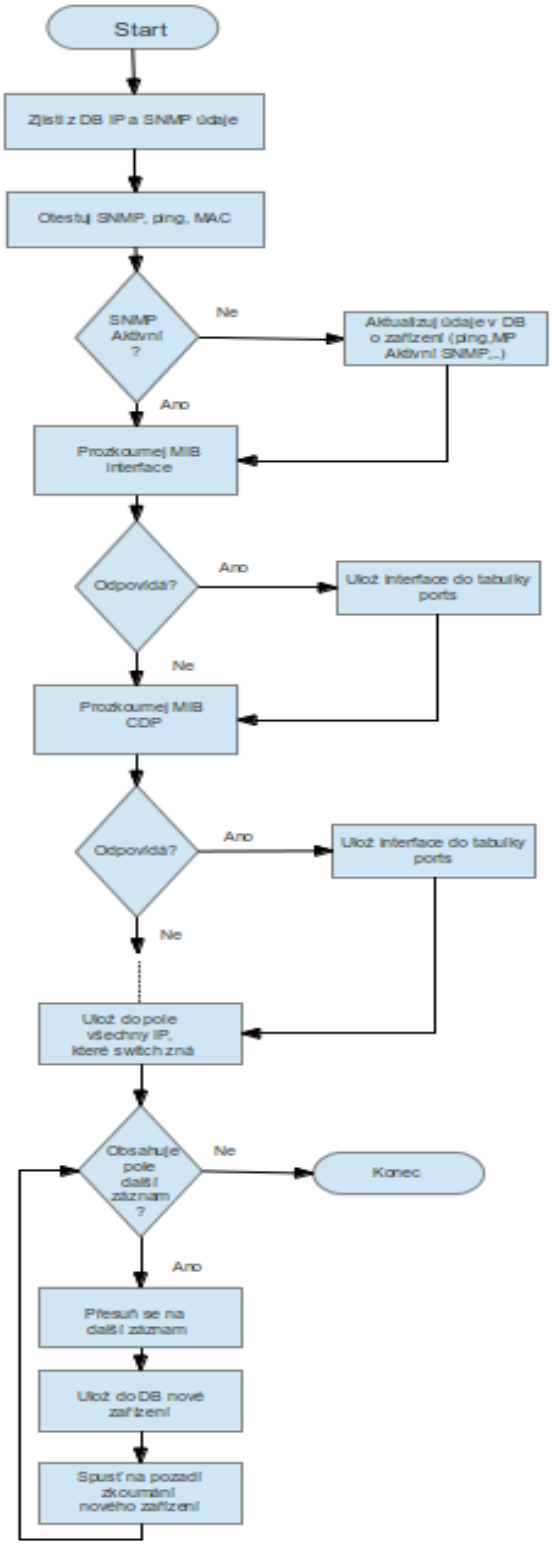
Příloha D – vývojový diagram čtení CDP MIB



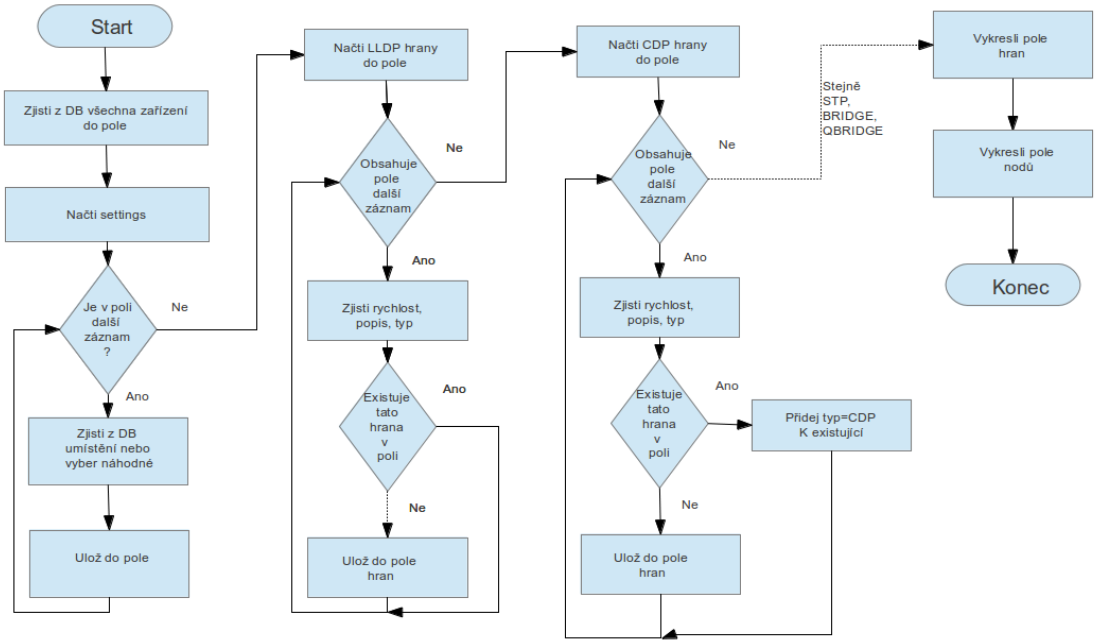
Příloha E – Vývojový diagram čtení Bridge MIB



Příloha F – Vývojový diagram zjišťování informací z jednoho SNMP zařízení



Příloha G – vývojový diagram vykreslení topologie



Příloha H – vývojový diagram monitoringu