



ZADÁNÍ DIPLOMOVÉ PRÁCE

Název: Centrální správa sít Národní technické knihovny
Student: Bc. Miroslav Brabenec
Vedoucí: Ing. Tomáš ejka
Studijní program: Informatika
Studijní obor: Po íta ové systémy a sít
Katedra: Katedra po íta ových systém
Platnost zadání: Do konce letního semestru 2017/18

Pokyny pro vypracování

Prostudujte aktuální ešení správy a dohledu (aplikace adminator [1]) nad prvky po íta ové sít Národní technické knihovny (NTK).

Navrhn te rozší ení aplikace o chyb ící funkcionalitu automatického zjiš ování stavu používaných aktivních sí ových prvk (p epína a analyzátor) a zapojení infrastruktury (kabeláže z analyzátor). Navrhn te uživatelské rozhraní, p es které bude možné prohlížet a vyhledávat zjišt né informace. Navržené rozší ení systému implementujte v etn automatizovaných test pokrývajících nov p idané funkce.

Po otestování funk nosti systému prove te nasazení nové verze v rámci NTK.

Seznam odborné literatury

[1] <https://github.com/techlib/adminator>

prof. Ing. Róbert Lórencz, CSc.
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.
d kan

V Praze dne 5. ledna 2017

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
KATEDRA POČÍTAČOVÝCH SYSTÉMŮ



Diplomová práce

Centrální správa sítě Národní technické knihovny

Bc. Miroslav Brabenec

Vedoucí práce: Ing. Tomáš Čejka

4. května 2017

Poděkování

Mé poděkování patří panu Ing. Tomáši Čejkovi za odborné vedení, rady a připomínky poskytnuté během tvorby této práce. Dále bych také rád poděkoval své rodině a přátelům za jejich psychickou podporu během mého studia.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 4. května 2017

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2017 Miroslav Brabenec. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.

Odkaz na tuto práci

Brabenec, Miroslav. *Centrální správa sítě Národní technické knihovny*. Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2017.

Abstrakt

Adminator je aplikace, která je v Národní technické knihovně využívána ke správě sítě. Momentálně je aplikací spravováno DHCP, DNS a přiřazování VLAN. Cílem této práce je rozšíření funkcí o monitoring aktivních síťových prvků (přepínačů a analyzátorů) a zapojení infrastruktury (kabeláže z analyzátorů). Nově vytvořené funkce umožňují zjistit, které síťové zásuvky jsou aktivní, jaká je konfigurace rozhraní přepínače a kde bylo spatřeno zařízení s danou MAC adresou. V každodenním provozu pak tyto funkce usnadňují přesun zařízení, detekci nekorektně nastavených portů, nalezení zařízení s vadným zdrojem.

Klíčová slova přepínač, MAC adresa, SNMP, Python, monitoring sítě, počítačová síť

Abstract

Adminator is an application that is used to manage the network in the National Library of Technology. The application is responsible for management of DHCP, DNS and dynamic VLAN assignment. The aim of this work is to extend the functionality of the application. Planned extensions are monitoring of active network components and monitoring of network infrastructure

wiring. It is planned to add functions which allow you to determine which network sockets are active, what is the configuration of interface and where the selected device have been spotted. Tasks like moving of equipment, detection of incorrectly configured interfaces and finding a device with a faulty power source will be easier due to this features.

Keywords switch, MAC address, SNMP, Python, network monitoring, computer network

Obsah

Úvod	1
1 Analýza	3
1.1 Síť v NTK	3
1.2 Adminator	13
1.3 Případy užití	21
1.4 Zdroje dat	25
2 Návrh	29
2.1 Fyzická struktura sítě	29
2.2 Data z přepínačů	31
2.3 Vzory konfigurace přepínačů	35
2.4 Uživatelské rozhraní	36
3 Realizace	41
3.1 Topologie	41
3.2 SNMP data z přepínačů	43
3.3 Konfigurace přepínačů	47
3.4 Vzorová nastavení rozhraní přepínače	49
3.5 Uživatelské rozhraní	52
4 Vyhodnocení	57
4.1 Adminator	57
4.2 Případy užití	59
4.3 Testování	60
Závěr	61
Literatura	63

A Seznam použitých zkratek	67
B Obsah přiloženého CD	71

Seznam obrázků

1.1	Přepínač 3Com typ 5500G	4
1.2	Přepínač HP typ 5120	5
1.3	Přepínače 3Com rodiny S7900	6
1.4	Přepínač HP typ 5820X	7
1.5	Zjednodušený diagram sítě v budově NTK	8
1.6	Analyzátor a dvojitá prezentace	10
1.7	Promo snímek aplikace AMPTrac ICM	11
1.8	Diagram nasazení aplikace	15
1.9	Architektura adminator-daemon	17
1.10	Rozhraní pro správu zařízení	18
1.11	Rozhraní pro správu DHCP	19
1.12	Rozhraní pro správu zapůjčených IP adres	20
1.13	Rozhraní pro správu domén	20
1.14	Rozhraní pro správu DNS záznamů	21
1.15	Výstup příkazu display brief interface na 3Com 5500G	23
1.16	Webové rozhraní analyzátoru	26
1.17	Přihlašovací formulář do přepínače HP 5120	27
1.18	Výběh portu ve webovém rozhraní 3Com 5500G	28
2.1	Databázový diagram fyzické struktury sítě	30
2.2	Databázový diagram přepínačů a rozhraní	32
2.3	Databázový diagram vzorových konfigurací rozhraní	36
2.4	Wireframe pohledu na topologii	37
2.5	Wireframe pohledu na detekované MAC adresy	38
2.6	Wireframe pohledu na rozhraní	39
2.7	Wireframe pohledu na detail rozhraní	40
3.1	Distribuce trvání aktualizací topologie	42
3.2	Dělení dat získaných přes SNMP	44
3.3	Skládání dat o jednotlivých rozhraních	45

3.4	Časová náročnost získání jednotlivých SNMP podstromů	46
3.5	Distribuce časů pro získání dat z přepínačů přes SNMP	48
3.6	Ukázka parsování konfiguračních souborů	50
3.7	Počet vzorů, které má rozhraní	51
3.8	Počet rozhraní odpovídajících danému vzoru	52
3.9	Rozhraní pro správu přepínačů	53
3.10	Rozhraní pro správu vzorových konfigurací	53
3.11	Rozhraní pro zobrazení topologie	54
3.12	Rozhraní pro zobrazení detekovaných MAC adres	55
3.13	Souhrnné zobrazení rozhraní	55
3.14	Detail rozhraní	56
4.1	Diagram nasazení aplikace	58

Úvod

Při provozu počítačové sítě v Národní technické knihovně byl vyvozen závěr, že provoz a správu sítě si lze výrazně usnadnit použitím systému, který umožní centrálně ovládat základní síťové služby. Představa byla taková, že by bylo příjemné mít možnost nastavovat DHCP, DNS a RADIUS ověřování na jednom místě. Dále by také mělo být možné, aby mohli konfiguraci provádět i méně zkušení administrátoři.

Výsledkem bylo vytvoření aplikace adminator, kterou vytvořili Ondřej Koch, Jakub Koudelka a Jan Dvořák[1]. Aplikace umožňuje skrz webové rozhraní ovládat výše zmíněné síťové služby. V porovnání s editací konfiguračního souboru je pak administrace přes webové rozhraní jednodušší.

Při používání aplikace bylo zjištěno, že by bylo vhodné mít k dispozici i další informace. Zejména se jednalo o možnost odpovědět na otázky:

- Je daná síťová zásuvka připojena do přepínače nebo končí na patch panelu v rozvodně?
- Kdy byla naposledy aktivní daná síťová zásuvka?
- Jak je nastaveno rozhraní přepínače, do kterého je připojeno dané zařízení?
- Kde bylo naposledy připojeno dané zařízení?

Diplomová práce se zabývá rozšířením funkcí aplikace adminator tak, aby bylo možné na tyto otázky odpovědět. Práce je rozdělena do několika částí. Každá část je zaměřena na jinou fázi realizace rozšíření.

V první části se zaměřuji na analýzu stávajícího stavu. Vymezuji, co adminator umí, jaké technologie využívá a co za data obsahuje. Popisuji zde stav, strukturu a jednotlivé komponenty sítě, na které bude třeba brát ohled. Zjišťuji, jaká data jsou dostupná ve strojově zpracovatelné formě a jaká data

bude potřeba ručně vytvořit nebo převést. V neposlední řadě zde také řeším formálnější specifikaci jednotlivých případů užití.

Druhá část je zaměřena na návrh jednotlivých rozšíření aplikace adminator. Diskutuji zde možnosti získání potřebných informací z konkrétních zdrojů dat. Řeším jakou formou data uložit do databáze a jak na sebe budou napojeny jednotlivé nově implementované komponenty.

Třetí část je zaměřena na realizaci jednotlivých rozšíření aplikace adminator. Věnuji se zde detailům jednotlivých komponent. Řeším zde reálné chování jednotlivých částí sítě. Rozebírám zde například interpretaci dat, která o sobě jednotlivá zařízení sdělují, časovou náročnost jednotlivých úloh (co lze dělat sekvenčně, kde je třeba postupovat paralelně) a realizaci uživatelského rozhraní.

Poslední část je zaměřená na sumarizaci výsledků této práce. Rozebírám zde, jak byly rozšířeny jednotlivé části aplikace adminator, zmiňuji zde, jak byly realizovány jednotlivé případy užití a popisuji zde, jak vypadá aktuální nasazení aplikace adminator.

Analýza

1.1 Síť v NTK

Součástí práce bylo zmapovat aktuálně používaná zařízení v infrastruktuře NTK. Tato sekce shrnuje zjištěné informace.

1.1.1 Síťové prvky

Část sítě v NTK, kterou se budu primárně zabývat v této práci, je složena z několika typů přepínačů. V této sekci se věnuji jednotlivým přepínačům, se kterými se lze v NTK setkat. U každého typu přepínače zmiňuji jednak vlastnosti, které jsou potřebné pro komunikaci s přepínačem (verze a typ OS, podporované technologie), jednak vlastnosti, které se mohou hodit při fyzické manipulaci s přepínači (vzhled, typ stohovacích kabelů). Jednotlivé typy přepínačů jsou různě staré, mají různé parametry a jsou určeny pro různé použití. Jediná společná vlastnost je operační systém Comware. Verze Comware je však rozdílná v závislosti na stáří přepínače.

1.1.1.1 3Com 5500G

Jedná se o spravovatelný přepínač (viz obr. 1.1) určený pro *access* vrstvu podnikové sítě, který zvládá poskytovat konektivitu o rychlosti 1 Gbit/s, lze na něm využívat PoE¹, stohování a lze nastavit *link aggregation*. Tyto přepínače jsou využívány v patrových rozvodnách a zajišťují konektivitu pro síťové zásuvky na jednotlivých patrech.

Souhrn parametrů[2][3]:

- datum vydání mezi 2005 - 2006
- operační systém Comware 3

¹Power over Ethernet (PoE)



Obrázek 1.1: Přepínač 3Com typ 5500G

- podpora L2 a L3 přepínání
- možnost stohování (maximálně 8 jednotek)
- stohování přes *cascade* rozhraní, které je součástí rozšiřovacího modulu
- možnost správy přes CLI lokálně nebo vzdáleně přes SSH nebo telnet
- možnost správy přes webové rozhraní
- možnost monitoringu a správy přes SNMP
- 4x SFP port v režimu *dual-personality*²

1.1.1.2 HP 5120

Jedná se o spravovatelný přepínač (viz obr. 1.2) určený pro *access* vrstvu podnikové sítě, který zvládá poskytovat konektivitu o rychlosti 1 Gbit/s, lze na něm využívat PoE, stohování a lze nastavit *link aggregation*. Tyto přepínače jsou využívány v patrových rozvodnách a zajišťují konektivitu pro síťové zásuvky na jednotlivých patrech.

Souhrn parametrů[4][5]:

- datum vydání mezi 2011
- operační systém Comware 5
- podpora L2 a L3 přepínání

²dva porty chovající se jako jeden (nastavení se propisuje, pokud je jeden zapnutý druhý musí být vypnutý)



Obrázek 1.2: Přepínač HP typ 5120

- možnost stohování (maximálně 9 jednotek)
- stohování přes SFP+ rozhraní, které součástí rozšiřovacího modulu
- možnost správy přes CLI lokálně nebo vzdáleně přes SSH nebo telnet
- možnost správy přes webové rozhraní
- možnost monitoringu a správy přes SNMP

1.1.1.3 3Com S7900

Jedná se o spravovatelný modulární přepínač (viz obr. 1.3) určený pro *core* vrstvu podnikové sítě. Přepínač podporuje vysokou dostupnost, čili má redundantní zdroje napájení a ventilátory s možností výměny za běhu. Možnosti konektivity závisí na instalovaných kartách. Přepínače tohoto typu jsou v NTK osazeny převážně kartami s porty o rychlosti 1 Gbit/s a jsou použity v serverovně, kde zajišťují konektivitu patrových přepínačů a některých serverů. Z pohledu sítě se používají částečně na *core* vrstvě a částečně na distribuční vrstvě.

Souhrn parametrů[6]:

- datum vydání mezi 2008 - 2009
- operační systém Comware 3
- podpora L2, L3 a L4 přepínání a směrování
- vysoká dostupnost
 - redundantní zdroje a ventilátory
 - možnost výměny HW za běhu



Obrázek 1.3: Přepínače 3Com rodiny S7900

- možnost správy přes CLI lokálně nebo vzdáleně přes SSH nebo telnet
- možnost správy přes webové rozhraní
- možnost monitoringu a správy přes SNMP

1.1.1.4 HP 5820X

Jedná se o spravovatelný přepínač (viz obr. 1.4) určený pro *core* vrstvu podnikové sítě. Přepínač podporuje vysokou dostupnost, čili má redundantní zdroje napájení a ventilátory s možností výměny za běhu. Jednotlivé přepínače tohoto typu lze spojit pomocí stohování. Jedná se o přepínač, který poskytuje konektivitu s rychlostí 10 Gbit/s. Zařízení tohoto typu jsou použita v serverovně, kde propojují uplinky, router, distribuční vrstvu a některé servery.

Souhrn parametrů[7][8]:

- datum vydání 2011
- operační systém Comware 5 případně Comware 7
- podpora L2, L3 přepínání a směrování
- vysoká dostupnost
 - redundantní zdroje a ventilátory



Obrázek 1.4: Přepínač HP typ 5820X

- možnost výměny HW za běhu
- možnost správy přes CLI lokálně nebo vzdáleně přes SSH nebo telnet
- možnost správy přes webové rozhraní
- možnost monitoringu a správy přes SNMP
- možnost stohování (maximálně 9 jednotek)
- podpora OpenFlow a SDN (Comware 7)
- možnost správy přes NETCONF[9] (Comware 7)
- možnost skriptování v Python (Comware 7)

1.1.2 Struktura sítě

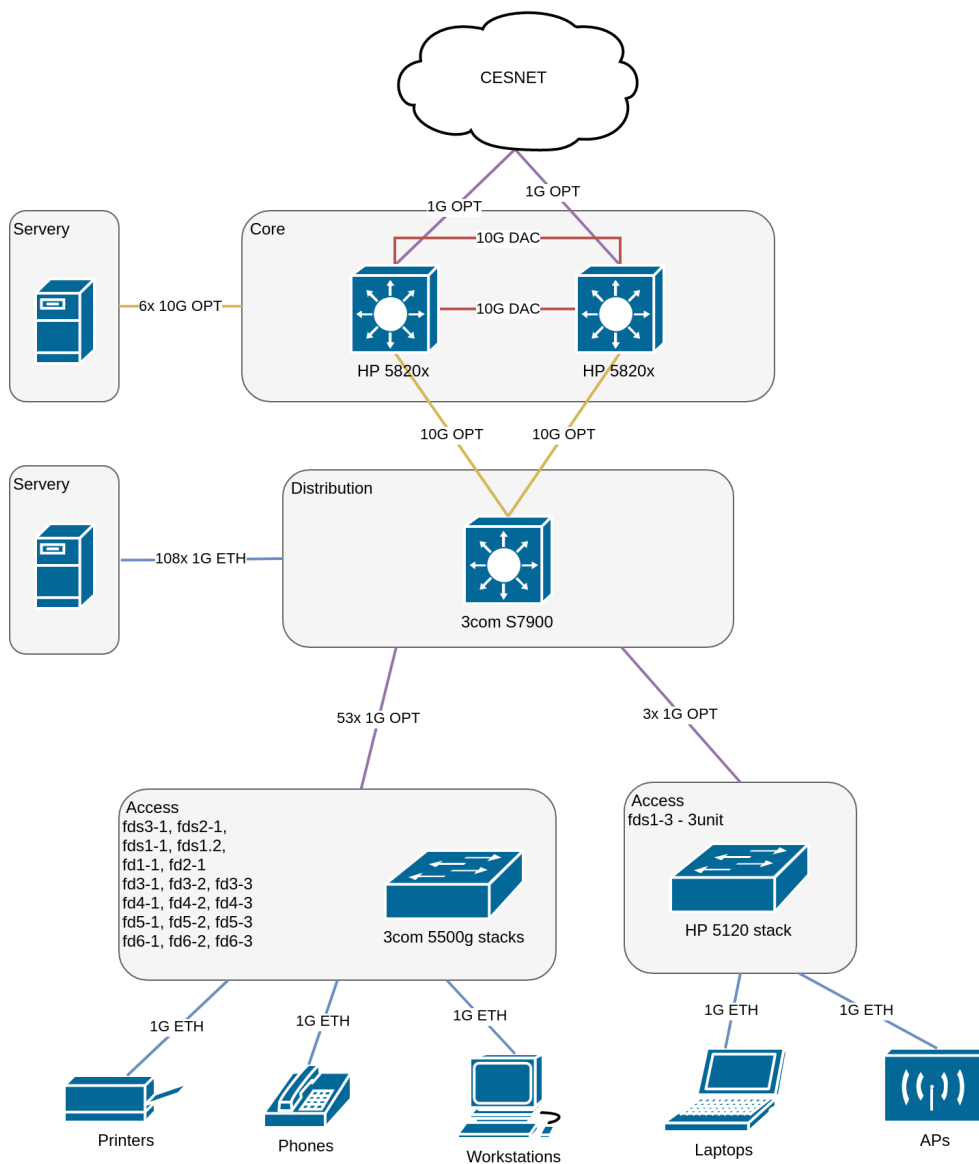
Pro účely diplomové práce si vystačíme se zjednodušenou strukturou sítě v NTK (viz obr. 1.5). Zjednodušením mám na mysli odstranění a zanedbání některých zařízení jako firewall, WIFI controller apod. Dále také z důvodů tunelování vynechávám problematiku zařízení připojených přes WIFI.

Core vrstvu sítě tvoří dvojice přepínačů HP 5820X spojených do stohu skrz dva propoje každý o rychlosti 10 Gbit/s. Do této vrstvy jsou připojeny servery, jejichž síťové karty podporují přenosovou rychlost 10 Gbit/s. Dále jsou do těchto zařízení připojeny uplinky do sítě CESNET.

Distribuční vrstva je do *core* vrstvy propojena dvěma 10 Gbit/s propoji. Konektivitu zde pak zajišťuje modulární přepínač 3Com S7900. Většina modulů tohoto přepínače je osazena kartami, které podporují rychlost 1 Gbit/s. Zařízení připojená do této vrstvy lze rozdělit do dvou skupin. První jsou servery, které mají síťové karty podporující maximální přenosovou rychlost 1 Gbit/s. Druhou skupinou jsou přepínače *access* vrstvy.

Access vrstva je tvořena stohy přepínačů typu 3Com 5500G a HP 5120. Většina stohů je složena ze 2 a více přepínačů. V místech s nízkými nároky na

1. ANALÝZA



Obrázek 1.5: Zjednodušený diagram sítě v budově NTK

konektivitu však lze nalézt i stohy složené pouze z jednoho přepínače. Každý stoh má pak z důvodu redundance nejméně 2 uplinky o rychlosti 1 Gbit/s do distribuční vrstvy.

Do *access* vrstvy jsou pak připojena veškerá koncová zařízení. Krom zařízení zobrazených na diagramu lze zmínit ještě hodiny, AV zařízení (kamery, televize, encodéry, . . .), tenké klienty, infokiosky, faxy, zákaznická zařízení, . . .

Tato práce se následně zaměřuje pouze na *access* vrstvu. Toto omezení má několik důvodů. Prvním důvodem je velikost této vrstvy. Tato vrstva je jednoznačně největší. Nachází se zde desítky síťových prvků a stovky fyzických zařízení. Ve zbylých dvou vrstvách se nachází jednotky síťových prvků a přibližně sto serverů. Druhým důvodem pro omezení na *access* vrstvu je fakt, že zařízení připojená do této vrstvy jsou přesouvána výrazně častěji (PC nebo tiskárna versus server v racku). Posledním důvodem pro omezení na *access* vrstvu je jednoznačnost informace, kde je připojeno dané zařízení. Zařízení připojená do *access* vrstvy mají typicky pouze jedno aktivní síťové rozhraní. Servery jsou typicky připojeny přes několik linek, které jsou využívány najednou. Další věc kterou by zde bylo třeba brát v úvahu jsou virtuální servery, které se pohybují mezi jednotlivými uzly virtualizační platformy, nebo klienti připojení tunelem přes virtualizovaný VPN server.

1.1.3 Chytrá kabeláž

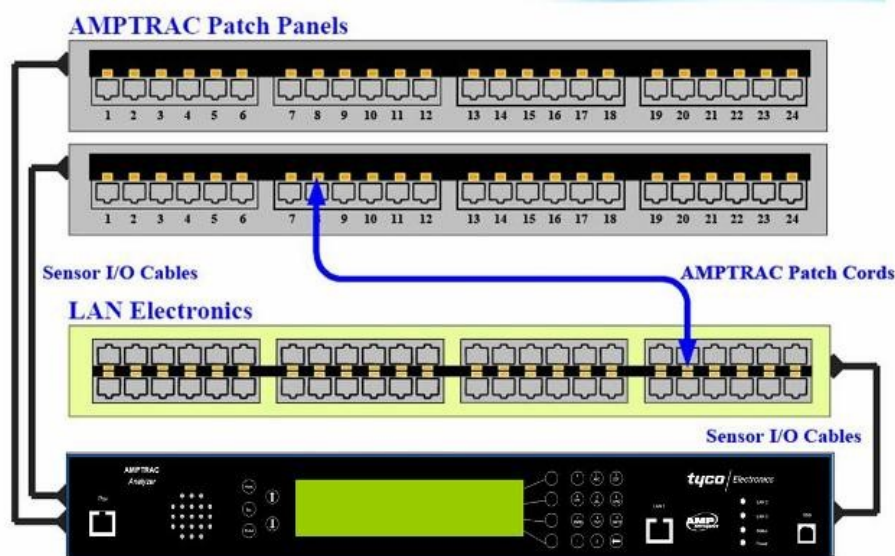
1.1.3.1 Teorie

Další součást síťové infrastruktury, která v NTK existuje, je chytrá kabeláž. Jedná se o systém AMPTrac od firmy TE connectivity. Účelem systému je monitorování fyzické vrstvy počítačové sítě. V praxi to znamená vědět odkud a kam vede daný kabel nebo které dva porty na patch panelu jsou propojeny. Systém se skládá z několika částí[10].

První částí systému chytré kabeláže jsou analyzátoři. Analyzátor (viz obr. 1.6) slouží k detekci propojení dvou portů na patch panelu. Aby mohl detekovat propoj, je potřeba splnit několik předpokladů. Nejprve je nutné mít chytrý patch panel, který je připojený do daného analyzátoru. Chytrý patch panel vypadá jako klasický patch panel s tím rozdílem, že nad každým portem je malá měděná kontaktní ploška. Druhým předpokladem je použití chytrého kabelu. Ten vypadá jako běžný UTP nebo STP kabel, který má jednu linku navíc. Tato linka je na obou konektorech vyvedena do měděného zobáčku, který se po zapojení dotýká měděné plošky na patch panelu. Pokud je toto splněno analyzátor detekuje uzavřený okruh a tím pádem i propoj.

Množství patch panelů na jeden analyzátor je omezeno na 14. Pokud je potřeba více patch panelů je třeba jeden analyzátor nastavit jako *master* a další k němu připojit ve *slave* módu[11].

Způsob zapojení, který je potřeba, aby fungovala detekce propojů, je dvojitá prezentace (viz obr. 1.6). Tento pojem znamená, že je třeba mít na patch



Obrázek 1.6: Analyzátor a dvojitá prezentace

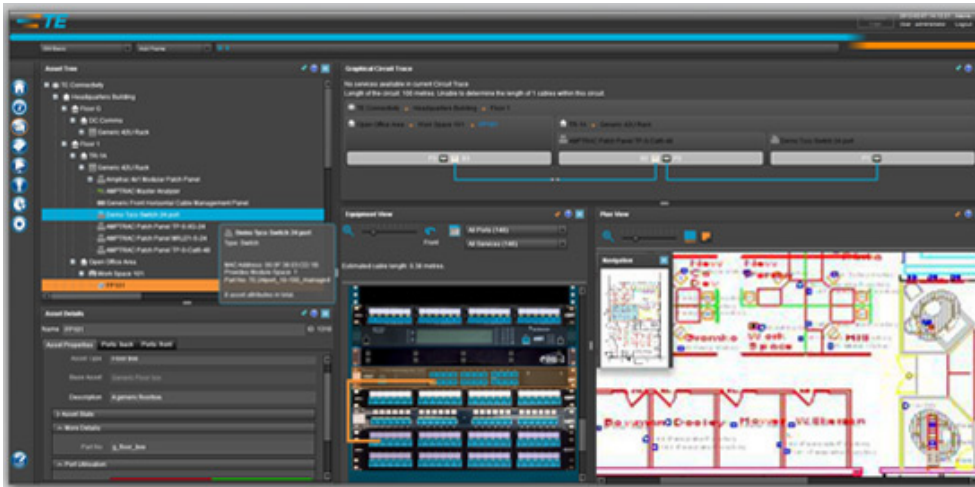
panely vyvedeny jak patrové zásuvky, tak rozhraní přepínače. Máme-li tedy například jeden přepínač, který má 24 portů a 48 síťových zásuvek znamená to, že potřebujeme 1 patch panel pro přepínač a 2 patch panely pro zásuvky. Dále je nutné mít všechny tři patch panely připojené do stejného analyzátoru.

Druhou částí systému chytré kabeláže je aplikace AMPTrac ICM (viz obr. 1.7). Tento software slouží k vizualizaci dat z analyzátorů. Jedná se o systém s plně přizpůsobitelným grafickým rozhraním pro zobrazování stavu fyzické síťové infrastruktury. Tento software umožňuje vizualizovat vzhled jednotlivých racků, lze do něj nahrát plány budovy a stromová struktura umožňuje prakticky neomezené nastavení struktury sítě. Dále je zde také funkce *automatic discovery*, která umí automaticky detekovat a následně zobrazovat zařízení připojené do sítě[12][13].

1.1.3.2 Praxe

Z popisu lze usoudit, že systém chytré kabeláže je robustní nástroj, který pokrývá takřka všechny požadavky na funkce o které chceme rozšířit adminator. Reálné nasazení však ukázalo několik problémů.

V případě chytrých patch panelů a kabelů, je zde problém, jelikož se jedná o poměrně atypické zboží vyráběné pouze jedním výrobcem. To má za důsledek poměrně vysokou pořizovací cenu těchto komponent. Cena 4 patch panelů (4x 24 portů) a příslušného množství propojovacích kabelů je srovnatelná s cenou 48 portového spravovatelného přepínače s podporou PoE. Atypičnost,



Obrázek 1.7: Promo snímek aplikace AMPTrac ICM

vázanost na jednoho výrobce a vyšší náklady pak komplikují rozvoj síťové infrastruktury.

Dvojitá prezentace má za následek zvýšení nároků na prostor v racku. Dále je zde také problém s omezeným množstvím patch panelů na jeden analyzátor, navýšení kapacity znamená prakticky vždy 2 nové patch panely.

Nasazení aplikace AMPTrac ICM v NTK také nebylo ideální. Interaktivní plány budovy se zásuvkami nebyly do aplikace nikdy nahrány. Stromová struktura byla nevhodně zvolená, jelikož pro nalezení požadovaného portu bylo třeba projít několik úrovní stromu (budova, podlaží, číslo místnosti, analyzátor a patch panel).

Samotná aplikace pak vyžaduje Adobe Flash Player, který lze dnes pokládat za ustupující technologii, která je vytlačována HTML5 a která je již v některých prohlížečích blokována[14]. Přítomnost plně přizpůsobitelného grafického rozhraní znamenala spíše potřebu vytvořit vlastní grafické rozhraní. Schopnost a snaha mnoho věcí vizualizovat vedla k neúnosně vysoké době odezvy. Editace záznamů v aplikaci byla poměrně složitá a funkce *automatic discovery* byla velmi nespolehlivá a měla velmi vysoké hardwarové nároky.

Momentálně je aplikace v NTK nasazena, ale lze ji prohlásit za nefunkční, jelikož nebylo možné ani vytvořit vlastní snímky obrazovky. Při snaze vytvořit vlastní snímky obrazovky, jsem narazil na několik problémů. První problém nastal ve špatném nastavení šifrování, kde je využíváno slabých klíčů. Aktuální verze prohlížečů v základním nastavení neumožňují komunikaci s weby, které mají takto nastavené šifrování. Po vyřešení problémů se zabezpečením jsem však narazil na nefunkční přihlášení, které se mi nepodařilo opravit.

Vzhledem k aktuálnímu stavu aplikace AMPTrac ICM není možné ji po-

užívat. Pokud je potřeba pracovat s informacemi o propojení portů na patch panelech, je třeba se fyzicky dostavit do rozvodny.

1.1.3.3 Pojmenování portů

Názvy portů jsou do jisté míry samopopisné, lze je rozdělit do tří skupin.

První skupina

První skupina jsou propoje, které vedou z rozvodny po daném patře. Jejich formát je x.y.z (např. 5.1.138) kde:

- x - reprezentuje patro na kterém je rozvodna, 5 pro 5. nadzemní podlaží, S2 pro 2. podzemní podlaží
- y - reprezentuje sekci patra, ve které je rozvodna umístěna, číslo z rozsahu 1-3 pro většinu pater, na patrech s nižším množstvím zásuvek čísla 1-2 nebo pouze číslo 1
- z - reprezentuje číslo zásuvky, je ve formátu trojčiferného čísla doplněného o nuly, tzn. jde o čísla 001-999

Druhá skupina

Druhou skupinou jsou porty v rozvodně, jež končí v přepínači. Jejich formát je x.y.Su.i (např. 5.1.S2.25) kde:

- x - reprezentuje patro na kterém je rozvodna, 5 pro 5. nadzemní podlaží, S2 pro 2. podzemní podlaží
- y - reprezentuje sekci patra ve které je rozvodna umístěna, číslo z rozsahu 1-3 pro většinu pater, na patrech s nižším množstvím zásuvek čísla 1-2 nebo pouze číslo 1
- u - reprezentuje ID jednotky stohu přepínačů v dané rozvodně, rozsah odpovídá velikosti stohu, která může mít maximálně 9 jednotek
- i - reprezentuje číslo portu na dané jednotce stohu

Třetí skupina

Třetí skupinou jsou porty reprezentující propoje mezi rozvodnami. Jejich formát je x.y.z (např. S1.2.B1A, 3.2.P1, ...), kde:

- x - reprezentuje patro cílové rozvodny, 5 pro 5. nadzemní podlaží, S2 pro 2. podzemní podlaží

- y - reprezentuje sekci patra cílové rozvodny, číslo z rozsahu 1-3 pro většinu pater, na patrech s nižším množstvím zásuvek čísla 1-2 nebo pouze číslo 1
- z - reprezentuje identifikátor propoje

Porty první a druhé skupiny mají unikátní názvy. Porty třetí skupiny nemají unikátní název, jelikož je například potřeba vést propoje ze všech patrových rozvodů do serverovny. Nezávisle na tom, ve které patrové rozvodně jsem, bude propoj do rozvodny pojmenovaný S1.2.P1, S1.2.P2 atd. Druhá strana těchto propojů bude následně v rozvodně označena 1.1.P1, 2.1.P1, 2.2.P1, ...

1.2 Adminator

1.2.1 Technologie

Aplikace adminator je postavena na několika základních technologiích. V této sekci krátce popisují některé z nich.

PostgreSQL

PostgreSQL je objektově-relační databázový systém, který má vlastnosti tradičních databázových systémů spolu s vylepšeními, které lze nalézt v databázových systémech příští generace. Jedná se o opensource software vydaný pod licencí MIT, který je vyvíjen globální komunitou vývojářů a firem[15]. Nejnovější vydaná verze má označení 9.6 a vyšla 29.9.2016[16]. Aplikace adminator je částečně závislá na tomto typu databáze, jelikož využívá datové typy *jsonb*, *inet* a *macaddr*, které v jiných databázových systémech nemusí být dostupné nebo se zde mohou chovat jinak.

KEA DHCP Server

KEA je opensource DHCP server, vydaný pod licencí MPL2.0. V projektu je využíván tento DHCP server kvůli následujícím vlastnostem[17]:

- existující a zdokumentované API
- možnost online konfigurace (lze změnit konfigurační soubor a požádat server o přenačtení konfigurace bez nutnosti restartu)
- podpora PXE³
- Přidělené IP adresy lze ukládat do PostgreSQL
- Rezervace IP lze ukládat do PostgreSQL

³Preboot eXecution Environment - síťový boot

PowerDNS

PowerDNS je opensource DNS server, šířený pod licencí GNU GPLv2. Jedná se o DNS server, jehož první klíčovou vlastností je možnost použití mnoha různých back-endů (databáze, soubory ve stylu DNS serveru BIND, ...). Druhou klíčovou vlastností je pak možnost provádět změny konfigurace bez nutnosti restartu[18].

Twisted

Twisted je opensource síťový framework, napsaný v programovacím jazyku Python, šířený pod licencí MIT. Twisted podporuje široké spektrum protokolů a je složen z řady dílčích projektů. Obsahuje webový server, mnoho chat klientů, chat servery, poštovní servery a další[19].

Flask

Flask je opensource microframework pro Python, založený na projektech Werkzeug (sada nástrojů pro WSGI) a Jinja2 (šablonovací jazyk). Flask je šířený pod BSD licencí[20]. Micro v kontextu této technologie znamená, že framework obsahuje pouze minimální nutné množství základních funkcí. V základu se nenachází například žádná abstraktní databázová vrstva či formulářová validace. Díky minimalistickému designu je však Flask snadno rozšiřitelný o funkcionalitu dle vlastního výběru[21].

SQLSoup

SQLSoup je opensource projekt, šířený pod MIT licencí, který poskytuje pohodlný způsob, jak mapovat Python objekty na tabulky relační databáze bez nutnosti psaní deklarativního kódu v jakékoli formě. Tato knihovna je postavena nad SQLAlchemy ORM a poskytuje minimalistické rozhraní k existující databázi[22]. V projektu plní funkci abstraktní databázové vrstvy, která není obsažena v základní verzi Flask.

React

React je opensource JavaScript knihovna určená pro tvorbu uživatelských rozhraní, šířená pod licencí BSD. Zjednodušeně lze říci, že React prezentuje V (view) v MVC. Za vývojem Reactu stojí firmy Facebook a Instagram, které jej otevřely světu v roce 2013[23]. React je momentálně využíván na stránkách Netflix, PayPal, Imgur, Airbnb atd[24].

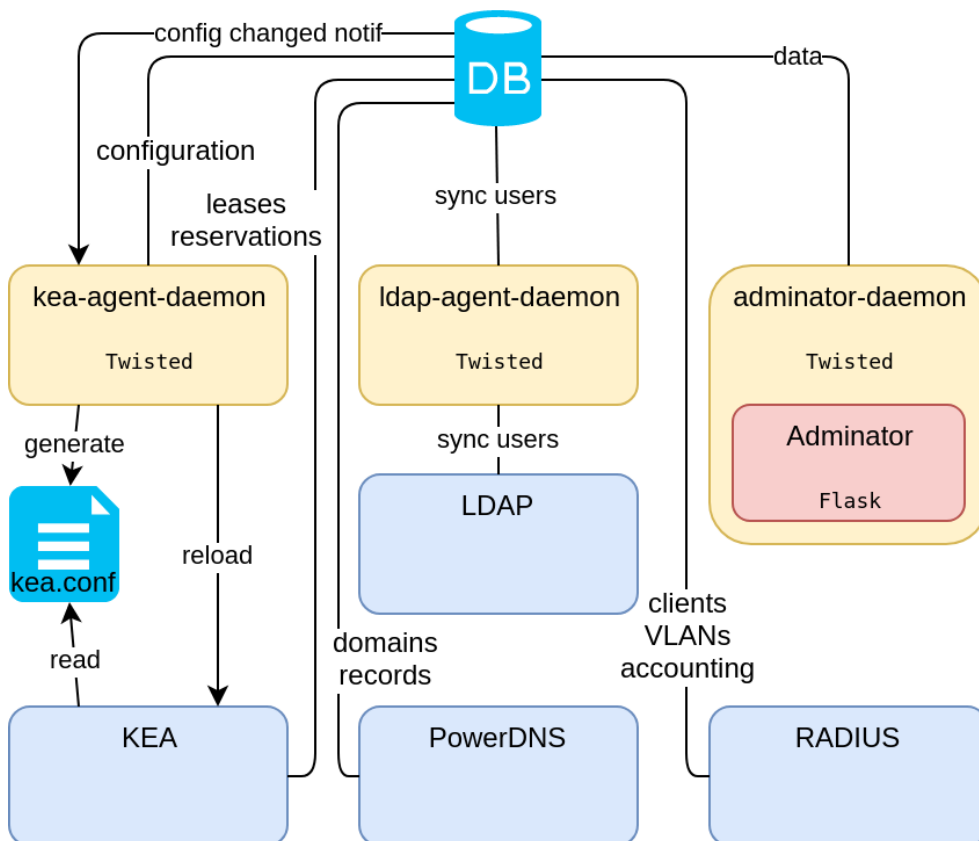
Griddle

Griddle je mimořádně přizpůsobitelná *datagrid* komponenta pro React. Jedná se o opensource JavaScript knihovnu šířenou pod licencí MIT. Griddle je primárně určen na zobrazování tabulkových dat. Díky návrhu však lze Griddle přizpůsobit na komponentu pro obecné zobrazení seznamů dat[25].

PatternFly

PatternFly je opensource komunitní projekt zaměřený na uživatelské rozhraní a UX. Obsahuje v sobě ukázky kódu, styly a návrhové vzory, které řeší časté problémy uživatelského rozhraní, ukázky[26].

1.2.2 Architektura aplikace



Obrázek 1.8: Diagram nasazení aplikace

Aplikace se skládá z několika oddělených částí (viz obr. 1.8). Centrálním bodem systému je databáze. Je zde použita upravená verze databázového systému PostgreSQL. Jedná se o Postgres-BDR ve verzi 9.4. Rozdíl mezi BDR (Bi-Directional Replication) a normální verzí spočívá v možnostech replikace databáze. BDR verze umožňuje použít replikaci databáze v *multi-master* režimu. Tento replikační režim pak lze použít pro clustery od 2 do 48 uzlů[27].

Funkci DHCP serveru obstarává KEA. Tato služba má konfiguraci rozdělenou na dvě místa. První část konfigurace je uložena v konfiguračním souboru (*kea.conf* typicky v */etc/kea/*). Zde se nachází definice a nastavení jednotlivých sítí, rozsahy IP poolů, intervaly expirace a obnovení zapůjčených IP adres (DHCP lease), specifikace back-endu pro ukládání zapůjčených IP adres atd. Druhá část konfigurace je samotné úložiště zapůjčených IP adres. V našem případě je jako úložiště použita databáze.

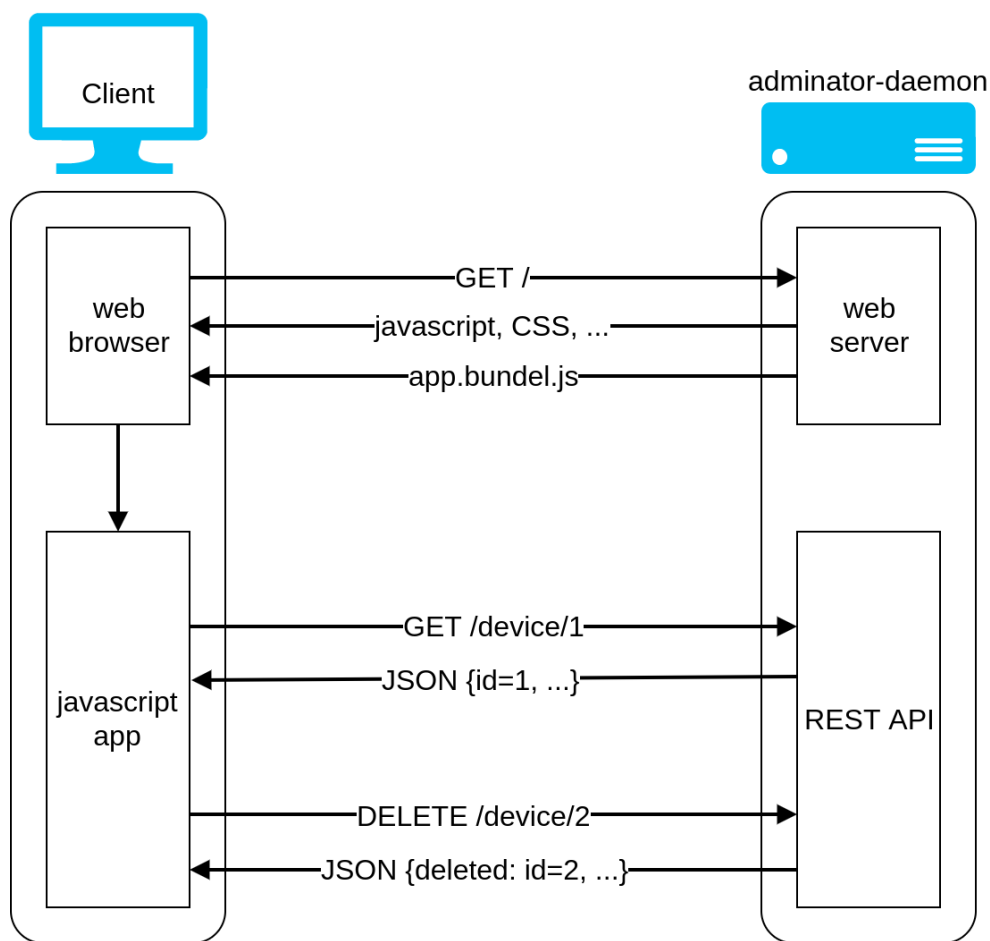
S obsluhou DHCP je spojen *kea-agent-daemon*. Tento *daemon* obstarává propagaci změn v konfiguraci do běžící služby KEA. V případě, že dojde ke změně, je v databázi nastaven trigger, který pošle notifikaci na *kea-agent-daemon*. Ten následně načte aktuální parametry z databáze a vygeneruje nový konfigurační soubor *kea.conf*. Když je soubor s novou konfigurací nagenеровán, *daemon* pošle signál *reload* do KEA služby, která si na základě signálu načte novou konfiguraci.

Funkci DNS serveru obstarává PowerDNS. Tato služba má vše potřebné uloženo v databázi. V databázi lze nalézt jak domény, tak i jednotlivé záznamy pro dané domény. V produkčním prostředí je pro zajištění funkce DNS nasazen ještě cache DNS server.

Funkci RADIUS serveru v systému obstarává FreeRADIUS server. Veškerá potřebná data má tato služba uložena v databázi. Přidáme-li tedy nové zařízení nebo změníme-li nastavení již existujícího zařízení, dojde k projevení změn okamžitě po uložení úprav. Poté již stačí jen, aby koncový síťový prvek znovu autorizoval zařízení vůči RADIUS serveru.

Zařízení se dělí do třech skupin. První skupina jsou zařízení, která nelze spárovat s konkrétním člověkem (bezpečnostní kamery, tiskárny, ...). Druhá skupina jsou zařízení, která lze spojit s nějakým evidovaným uživatelem (pracovní notebook, pracovní mobil, ...). Poslední skupinou jsou pak zařízení, která patří návštěvníkům, kteří nemají vztah k NTK (servisní technik, dodavatel, ...). Kvůli druhé skupině je třeba znát aktuální seznam evidovaných uživatelů. Tento seznam je tedy třeba synchronizovat s centrálním LDAP serverem. Tuto synchronizaci zajišťuje *ldap-agent-daemon*, který zároveň zajišťuje i odebrání přístupů zařízením, která patří expirovaným osobám (bývalí zaměstnanci, ...).

Poslední částí je *adminator-daemon*, ten realizuje web server, ve kterém běží webová aplikace *adminator*. Aplikaci lze rozdělit do dvou částí (viz obr. 1.9). První část je back-end napsaný ve Flask, který zajišťuje dvě věci. První z nich je doručení front-end aplikace klientům. Tato část je realizována tak, že na back-end přijde GET požadavek na URL /. Odpovědí na tento požadavek



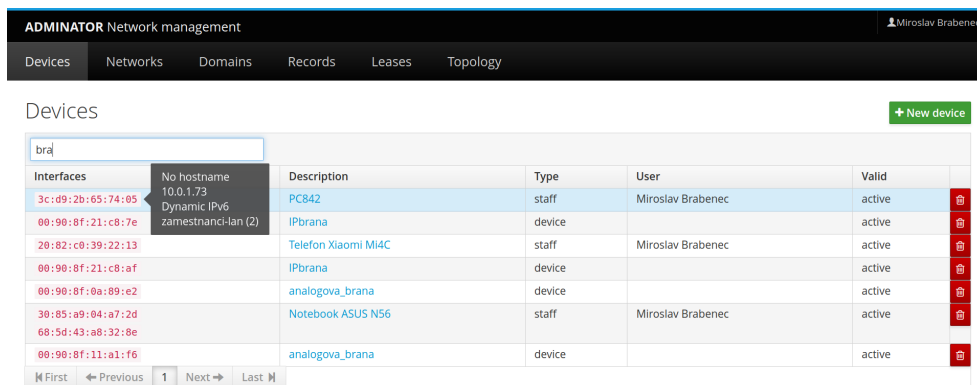
Obrázek 1.9: Architektura adminator-daemon

je HTML kód, který v sobě obsahuje informace o tom, kde najít potřebné JavaScript skripty a CSS soubory. Jedním z JavaScript souborů, které si klient na základě odpovědi stáhne, je front-end aplikace. Druhou věcí, která je realizována na back-endu, je RESTful API. Toto rozhraní je potřeba pro čtení a manipulaci s daty. Pro určení operace slouží typ požadavku a pro určení objektu slouží URL. Pro načtení zařízení s $id = 1$ se bude jednat o požadavek GET na URL `/device/1`. Pokud bychom chtěli smazat zařízení s $id = 2$, bude se jednat o požadavek DELETE na URL `/device/2`. Odpověď back-endu vždy obsahuje data ve formátu JSON.

To, co uživatel vidí na svém monitoru, je JavaScript aplikace, která běží v jeho prohlížeči. Veškeré akce, které v ní provádí, jsou zpracovány na straně klienta a na back-end přichází pouze požadavky na RESTful API. Komunikace mezi front-end a back-end probíhá asynchronně s využitím AJAX.

1. ANALÝZA

1.2.3 Funkce



Obrázek 1.10: Rozhraní pro správu zařízení

Správa RADIUS ověřování je zde realizována pomocí webového rozhraní (viz obr. 1.10), které dělá front-end ke CRUD aplikaci. Základní položkou pro správu je objekt device (zařízení). Parametry zařízení jsou název, který není unikátní, a typ zařízení (staff, device, visitor). Zařízení typu device nemají již další parametry. U zařízení typu staff je uvedena osoba, která za zařízení zodpovídá (zaměstnanec). Poslední typ zařízení je visitor, zde není obsaženo jméno osoby, ale pouze časový interval po který má být danému zařízení umožněn přístup do sítě.

K zařízení se pak pojí objekt interface (síťové rozhraní). Jedno zařízení může mít 0 až N rozhraní. Základním parametrem rozhraní je jeho MAC adresa, jež musí být unikátní. Dalším nutným parametrem rozhraní je síť (VLAN) do které patří. Volitelně, pak lze přiřadit statickou IPv4 adresu, IPv6 adresu a *hostname*. Tyto parametry se projeví do nastavení dané sítě na DHCP serveru.

Z pohledu věcí souvisejících s RADIUS ověřováním se změny provedené skrz webové rozhraní projeví okamžitě, jelikož RADIUS bere data přímo z databáze. Stačí tedy pak donutit koncový síťový prvek, aby znovu provedl ověření vůči RADIUS serveru. Změnám, které souvisí s DHCP, se budu věnovat v následující části.

Správa DHCP je rozdělena do několika částí. První z nich správu rezervací, která se provádí přes rozhraní pro správu zařízení, jsem zmínil již v předchozím odstavci.

Druhou částí je správa sítí (viz obr. 1.11). Zde lze se sítěmi provádět klasické CRUD operace. Záznam sítě se skládá z názvu, čísla příslušné VLAN, IPv4 prefixu a IPv6 prefixu. V daných prefixech je následně možné definovat rozsahy pro dynamické přidělování IP adres. Není tedy problém rozdělit roz-

The screenshot shows the Adminator Network management interface. At the top, there is a navigation bar with tabs for Devices, Networks, Domains, Records, Leases, and Topology. The 'Networks' tab is active. Below the navigation bar, there is a search bar labeled 'Filter Results' and a '+ New network' button. The main content is a table with the following columns: Description, VLAN, Prefix IPv4, Prefix IPv6, and Max. lease. The table lists various network configurations, including CVUT FIT, admini-lan, admini-vpn, admini-wlan, ballinguv sal, bezovereni-wlan, cctv, dmz, dmzI, e-reading, eduroam, garaze, hoste-bezovereni-lan, hoste-overeni-lan, and instalace.

Description	VLAN	Prefix IPv4	Prefix IPv6	Max. lease
CVUT FIT	203			3600
admini-lan	115	10.0.115.0/24	2001:718:7:115::/64	3600
admini-vpn	117	10.0.117.0/24	2001:718:7:117::/64	3600
admini-wlan	116	10.0.116.0/24	2001:718:7:116::/64	3600
ballinguv sal	219	10.8.0.0/24		259200
bezovereni-wlan	11	10.1.32.0/20		259200
cctv	5	10.0.5.0/24		532000
dmz	8	10.0.8.0/24		532000
dmzI	12	10.0.12.0/24		532000
e-reading	27	10.1.194.0/23		532000
eduroam	204	10.4.0.0/16		259200
garaze	501			3600
hoste-bezovereni-lan	10	10.1.16.0/20		259200
hoste-overeni-lan	14	10.2.4.0/22		3600
hoste-overeni-wlan	9	10.1.0.0/20		3600
instalace	24	10.100.0.0/23		532000

Obrázek 1.11: Rozhraní pro správu DHCP

sah na dvě poloviny, z nichž se jedna využívá pro dynamické přidělování adres a druhá je vyhrazena pro staticky přidělené adresy. Pro danou síť lze nastavit mnoho DHCP parametrů jako jméno domény, adresu výchozí brány, adresu NTP serveru (síťový čas), parametry pro PXE (boot ze sítě), ...

V rámci DHCP lze také spravovat zapůjčené IP adresy (viz obr. 1.12). Tato část podporuje pouze operace čtení a mazání. Snadno zde lze dohledat, jaká IP adresa je přidělena zařízení s danou MAC adresou, jaký je stav dané zápůjčky IP adresy, případně kdy bude daná zápůjčka IP adresy expirovat. Toto rozhraní v praxi usnadňuje zásahy, kdy je potřeba zařízení změnit IP adresu (přiřazení nebo změna statické adresy), jelikož je možné snadno smazat aktuálně zapůjčené IP adresy.

V případě mazání zapůjčených IP adres se změna projeví okamžitě. Pro ostatní změny propagace nového nastavení chvíli trvá. Je zde totiž nutné, aby došlo k vygenerování nového konfiguračního souboru a aby se poslal signál KEA službě, že si má znovu načíst konfiguraci. Načtení nové konfigurace pak může také chvíli trvat. Celkově by se však změny měly projevit v řádu jednotek maximálně desítek sekund.

Správa DNS je rozdělena do dvou částí a to na správu domén (viz obr. 1.13) a na správu záznamů (viz obr. 1.14) pro danou doménu. V obou případech jsou podporovány všechny operace CRUD. Rozhraní pro správu jednotlivých záznamů se pak liší v závislosti na typu záznamu.

Změny provedené v nastavení DNS se projeví okamžitě, jelikož PowerDNS využívá jako back-end přímo databázi. V produkčním prostředí pak může čas potřebný k propagaci změn ovlivnit DNS cache.

1. ANALÝZA

ADMINATOR Network management Miroslav Brabec

Devices Networks Domains Records **Leases** Topology

Leases

IPv4 IPv6

Filter Results

IP	MAC	Expire	Valid lifetime	State	Hostname	
10.1.5.66	c8:69:cd:97:cc:c8	2017-02-22 20:10:02	4000	expired-reclaimed		
10.4.13.143	34:c0:59:e8:d4:90	2017-02-22 21:12:34	4000	default		
10.1.200.249	c8:f6:50:91:07:49	2017-02-22 21:24:49	4000	default		
10.4.13.180	80:65:6d:bd:2a:e9	2017-02-22 20:27:40	4000	default		
10.4.13.183	2c:f0:a2:53:33:f5	2017-02-22 21:17:45	4000	default		
10.1.6.39	ac:cf:85:5b:09:12	2017-02-22 21:25:16	4000	default		
10.4.13.86	84:98:66:a1:4c:25	2017-02-22 19:42:41	4000	expired-reclaimed		
10.1.5.238	0c:6b:bd:a1:c3:a9	2017-02-22 20:46:49	4000	default		
10.1.200.197	5c:8d:4e:17:d3:12	2017-02-22 21:24:01	4000	default		
10.4.13.141	7c:d1:c3:e3:cb:61	2017-02-22 21:22:16	4000	default		
10.4.13.230	88:70:8c:db:7a:b9	2017-02-22 21:20:46	4000	default		
10.1.5.221	8c:eb:c6:12:95:61	2017-02-22 21:14:40	4000	default		
10.1.200.173	14:1f:78:2d:65:86	2017-02-22 21:07:30	4000	default		
10.4.10.160	e8:50:8b:d4:cd:e4	2017-02-22 20:53:10	4000	default		
10.4.13.140	40:0e:85:67:1b:5a	2017-02-22 21:05:17	4000	default		

Obrázek 1.12: Rozhraní pro správu zapůjčených IP adres

ADMINATOR Network management Miroslav Brabec

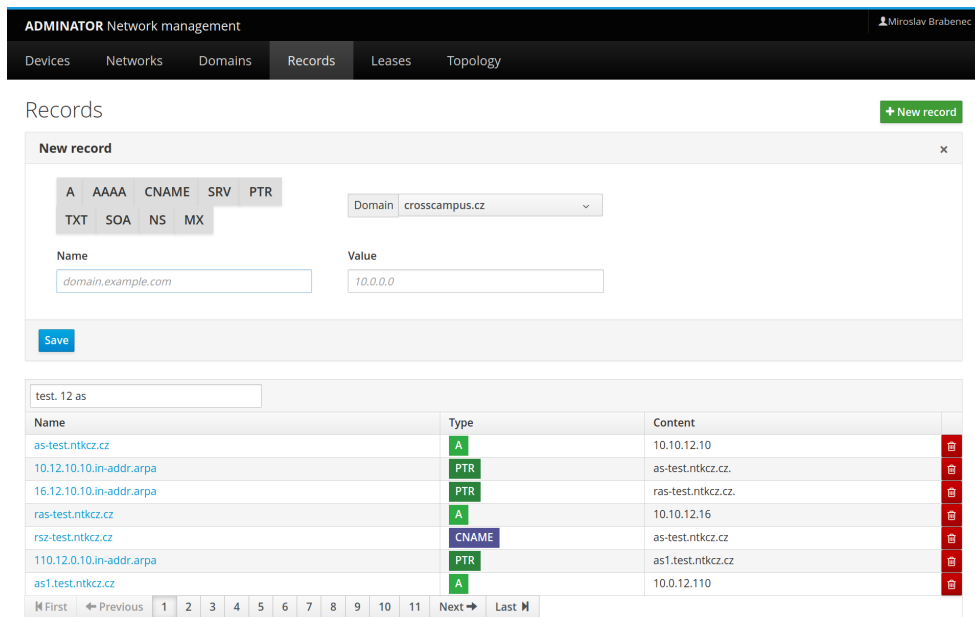
Devices Networks **Domains** Records Leases Topology

Domains

+ New domain

Name	
vpk.cz	
tedxnarodnitechickaknihovna.cz	
chemtk.cz	
crosscampus.cz	
festival-vedy.cz	
galerientk.cz	
issn.cz	
kampusdejvice.cz	
naviga.cz	
nusi.cz	
stk.cz	
7.0.0.8.1.7.0.1.0.0.2.ip6.arpa	
czechelb.cz	
techlib.cz	
ntkcz.cz	
0.10.in-addr.arpa	
1.10.in-addr.arpa	
10.10.in-addr.arpa	

Obrázek 1.13: Rozhraní pro správu domén



Obrázek 1.14: Rozhraní pro správu DNS záznamů

1.3 Případy užití

Rozšíření aplikace adminator lze rozdělit do několika skupin.

1.3.1 Topologie

První skupinou jsou rozšíření, která by měla nahrazovat aplikaci AMPTrac ICM. Primárně se jedná o možnost zjistit, jak jsou propojeny porty na patch panelech.

UC1 - Vyhledání propoje na základě názvu

Umožňuje uživateli vyhledat informace o propoji mezi patch panely na základě názvu propoje nebo jména rozvodny. Většina jmen jednoznačně určuje jeden port. Kombinace jméno rozvodny, jméno portu vždy jednoznačně určuje jeden port. Jména propojů a rozvodnů mohou být zadána celým názvem nebo jen jeho částí. Primárně se pak jedná o propoje v rámci jedné rozvodny mezi porty první a druhé skupiny (5.1.138 <-> 5.1.S2.25 - 5.1.138 je připojen do přepínače v rozvodně 5.1 do portu 2/0/25). Propoje mezi rozvodnami jsou využívány zřídka a není třeba podporovat zobrazení propojů skrz několik rozvodnů.

UC2 - Vyhledání propoje pomocí lokace

Umožňuje uživateli vyhledat informace o propoji mezi patch panely na základě umístění propoje nebo umístění rozvodny. První způsob zadání umístění je pomocí názvu místnosti. Ten může být zadán celým názvem nebo jen jeho částí. Druhým způsobem zadání umístění je přibližné vybrání místa na mapě budovy.

1.3.2 Porty přepínače

Druhá skupina rozšíření by měla umožňovat aplikaci adminator zobrazit informace o rozhraních daného přepínače.

UC3 - Zobrazení stavu daných rozhraní

Umožňuje uživateli vyhledat stavové informace o daném rozhraní na základě jeho názvu. Název může být zadán celý nebo jen částečně. Mělo by se jednat o stavové informace v podobném rozsahu v jakém jsou obsaženy ve stručném výpisu (Cisco IOS - show ip interface brief, Comware - display brief interface, viz obr. 1.13). Nejzajímavější jsou následující informace:

- administrativní stav
- operativní stav
- přenosová rychlost
- aktuální VLAN
- datum poslední změny stavu

UC4 - Zobrazení konfigurace daných rozhraní

Umožňuje uživateli zobrazit aktuální nastavení daného rozhraní. Dále by mělo být také možné filtrovat všechna rozhraní, která odpovídají dané vzorové konfiguraci a filtrovat rozhraní, která neodpovídají žádné definované konfiguraci. Bylo by dobré zde zvážit možnost práce s nastavením daného rozhraní přímo z aplikace adminator. Tato možnost je zajímavá, i pokud by se jednalo jen o změny konfigurace z definovaného vzoru do jiného definovaného vzoru.

UC5 - Zobrazení síťových zásuvek propojených do daného rozhraní

Umožňuje uživateli zjistit, která síťová zásuvka je propojena do daného rozhraní přepínače. Rozhraní je nutné v databázi propojit s porty na patch

Interface	Link	Speed	Duplex	Type	PVID	Description
Aux1/0/0	UP	--	--	--	--	
Cas1/2/1	DOWN	24G	full	--	--	
Cas1/2/2	DOWN	24G	full	--	--	
GE1/0/1	UP	A1000M	Afull	trunk	2	
GE1/0/2	UP	A100M	Afull	trunk	1	
GE1/0/3	UP	A1000M	Afull	trunk	2	
GE1/0/4	DOWN	A	A	trunk	1	
GE1/0/5	DOWN	A	A	trunk	1	
GE1/0/6	DOWN	A	A	trunk	1	
GE1/0/7	UP	A100M	Afull	trunk	1	

Obrázek 1.15: Výstup příkazu display brief interface na 3Com 5500G

panelu z druhé skupiny (názvy 5.1.S2.25, ...). Každé rozhraní přepínače pak může nebo nemusí být vyvedeno na patch panel.

1.3.3 Lokalizace zařízení

Třetí skupina rozšíření by měla poskytovat aplikaci adminator možnost zobrazit informace o zařízeních, která jsou nebo v minulosti byla připojena na síť.

UC6 - Lokalizace aktivního zařízení

Umožňuje uživateli zjistit, kde se momentálně nalézá aktivní zařízení s danou MAC adresou. Za informaci o poloze zařízení se považuje síťová zásuvka nebo rozhraní přepínače.

UC7 - Lokalizace neaktivního zařízení

Umožňuje uživateli zjistit, kde bylo naposledy spatřeno zařízení s danou MAC adresou. Za informaci o poloze zařízení se považuje síťová zásuvka nebo rozhraní přepínače.

UC8 - Poslední aktivita zařízení

Umožňuje uživateli zjistit, kdy bylo naposledy spatřeno zařízení s danou MAC adresou.

UC9 - Poslední zařízení na daném rozhraní

1. ANALÝZA

Umožňuje uživateli zjistit poslední zařízení (identifikované MAC adresou) spatřené na daném rozhraní přepínače. Každý záznam nese informaci o tom, kdy zde bylo dané zařízení spatřeno.

UC10 - Přehled detekovaných zařízení

Umožňuje uživateli zobrazit přehled, kde jsou vidět jednotlivá detekovaná zařízení spolu s informacemi o přepínači, rozhraní, síťové zásuvce a uživateli. V přehledu je možné vyhledávat pomocí všech těchto hodnot. Každá hodnota může být zadána celá nebo jen částečně.

1.3.4 Komplexní dotazy

Data by měla být uložena tak, aby bylo následně možné na úrovni databáze provádět komplexnější dotazy. Bude tedy vhodné používat kompatibilní datové typy, aby bylo možné snadno provádět SQL operace *join*.

UC11 - Kde bylo naposledy spatřeno zařízení daného uživatele

Umožňuje uživateli zjistit, kde bylo naposledy spatřeno zařízení daného uživatele. Za informaci o poloze zařízení se považuje síťová zásuvka nebo rozhraní přepínače.

UC12 - Kdy bylo naposledy spatřeno zařízení daného uživatele

Umožňuje uživateli zjistit, kdy bylo naposledy spatřeno zařízení daného uživatele.

UC13 - Konzistence VLAN daného zařízení

Umožňuje uživateli zjistit zda-li dané aktivní zařízení spadá do VLAN (nastavené v RADIUS serveru), která se shoduje s VLAN detekovanou na rozhraní přepínače, do kterého je zařízení připojeno.

UC14 - Konzistence nastavení rozhraní ve kterých jsou připojena daná zařízení

Umožňuje uživateli zjistit nastavení rozhraní, do kterých jsou připojena daná aktivní zařízení. Jako příklad lze uvést dotaz, kdy se uživatel snaží zjistit, zda-li jsou všechny tiskárny připojeny do shodně nastavených rozhraní.

1.4 Zdroje dat

Součástí práce bylo zmapovat zdroje, ze kterých lze získávat data o síti. Tato sekce shrnuje zjištěné informace.

1.4.1 AMPTrac ICM

Prvním zdrojem dat je databáze k aplikaci AMPTrac ICM. Tato databáze v sobě musí mít v nějaké formě uložená data, která popisují fyzickou strukturu chytré kabeláže. Jedná se o následující informace:

- vzájemné propojení analyzátorů (omezený počet patch panelů na jeden analyzátor)
- přiřazení patch panelů k analyzátoru včetně portu do kterého je patch panel připojen
- přiřazení jména portu k pozici na patch panelu

Tato databáze by také mohla poskytovat informace o propojích mezi patch panely. V obou případech však platí, že vzhledem ke komplexnosti celé aplikace mohou být data velmi obtížně interpretovatelná.

Databáze k této aplikaci se skládá z 56 tabulek a 6 pohledů (view). Z 56 tabulek se využívá pouze 5. Většina dat v této databázi je uložena v tabulce `asset`, která obsahuje prakticky všechny entity, které jsou v aplikaci AMPTrac ICM využity. V této tabulce lze nalézt:

- abstraktní třídy - Generic Switch, Generic Building, Generic Cable, ...
- druhy kabelů - Fibre Patch Cord, Horizontal Cable, Vertical Cable, ...
- druhy lokalit - Generic Factory, Generic Cube, Generic Desk
- druhy virtuálních strojů - Generic VM, VMware, Xen, ...
- lokality - Budova NTK, 5NP, 5.62.01 Schodiste, ...
- patch panely
- názvy portů
- a mnoho dalšího

Párování (port patří k patch panelu, místnost patří k patru, patro patří k budově, ...) je zde řešeno tak, že každý řádek tabulky obsahuje pole, ve kterém jsou uložena ID všech řádků, ke kterým má nějaký vztah. Pro párování portů k patch panelům je následně potřeba využít ještě tabulku `external_connections`, v níž jsou obsažena data o příslušnosti portu k patch panelu včetně pozice portu na patch panelu.

1.4.2 Analyzátoři

Druhou možností, jak získat data o propojích mezi patch panely, je zeptat se přímo konkrétních analyzátorů. Prvním způsobem komunikace může být SNMP. V případě SNMP bude ovšem potřeba dohledat informace o parametrech. Na analyzátořích v menu pro konfiguraci SNMP, lze nastavit pouze jméno, lokaci a kontakt.

Druhým způsobem komunikace může být webové rozhraní (viz obr. 1.16). Na webové rozhraní se lze připojit přes HTTPS s ověřením pomocí jednoduchého ověřování (basic access authentication) a umožňuje zobrazit mapu propojů. Na první pohled je vidět pouze informace, zda je propoj někam připojen. Při pohledu do HTML kódu však lze nalézt i informaci o tom kam propoj vede. Vzhledem k tomu, že webové rozhraní vrací klasické HTML, by mělo být poměrně snadné informace programově zpracovat.

The screenshot shows the TE Enterprise Networks web interface. The main content is a 'Connection Map' table. The table has 14 rows, one for each database unit (DB#01 to DB#14), and 24 columns representing ports (01 to 24). Green highlights in the cells indicate active connections. For example, DB#01 is connected to ports 03, 04, 05, 06, 07, 08, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, and 24. A 'Refresh' button is located below the table.

DB#	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
DB#01			03	04	05	06	07	08					13	14	15	16	17	18	19	20	21	22	23	24
DB#02	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
DB#03	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
DB#04	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
DB#05	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
DB#06	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
DB#07	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
DB#08	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
DB#09	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
DB#10	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
DB#11	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
DB#12	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
DB#13	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
DB#14	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24

Obrázek 1.16: Webové rozhraní analyzátoru

1.4.3 Mapové podklady

Pro získání informací o rozmístění jednotlivých síťových zásuvek a jejich párování k místnostem by bylo potřeba vycházet z plánů datových rozvodů jednotlivých pater budovy. Veškeré plány datových rozvodů, které se podařilo dohledat, jsou dostupné pouze v papírové podobě nebo ve formátu PDF. Verze v PDF je řešena formou vloženého bitmapového obrázku, takže v ní nelze textově vyhledávat. Všechny nalezené plány datových rozvodů jsou pak z revize,

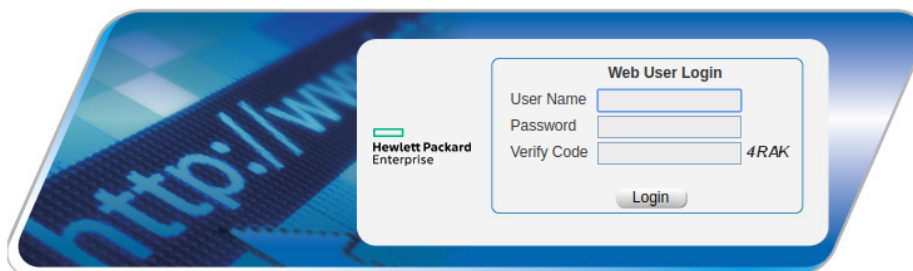
kteřá proběhla 30. 11. 2008. Nově vzniklé síťové zásuvky nebo síťové zásuvky, které se přesouvaly, v nich tedy nejsou zaneseny.

Pokud by bylo potřeba používat informace o přesné poloze zásuvek, bylo by třeba nejprve provést revizi reálného stavu datových rozvodů. Následně by bylo třeba zjištěné informace zanešt do plánů. Jelikož se mi nepodařilo nikde dohledat zdrojové soubory s plány budovy (soubory pro CAD), předpokládám, že by bylo nutné obrátit se na dodavatele stavby. Pokud by docházelo k revizi plánů budovy, předpokládám, že by nezůstalo pouze u datových rozvodů, ale došlo by také k zanesení dalších změn (stavební úpravy apod.). Revize rozvodů nebo případné překreslování plánů je však nad rámec této práce.

1.4.4 Přepínače

Pro komunikaci s přepínači existují 3 možnosti SNMP, webové rozhraní a přístup na CLI po SSH nebo Telnet. Možnost použití NETCONF nebo skriptování v Pythonu (vzdálená správa pomocí Ansible) je obsažena až v Comware 7, který není dostupný pro žádný přepínač použitý na *access* vrstvě.

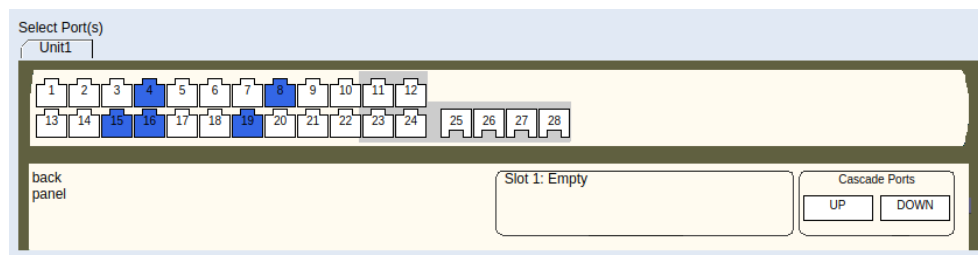
Přítomnost webového rozhraní naznačuje, že by přepínače mohly mít API, přes které by bylo možné je monitorovat a spravovat. Podrobnosti k webovému rozhraní bohužel nejsou nikde zdokumentovány. Rozhraní samotné je pak interaktivní. Jsou zde prvky jako *user verification* (nutnost přepsat verifikační kód při přihlášení, viz obr. 1.17), výběr portu pomocí obrázku (viz obr. 1.18) apod. Variantu automatizovat komunikaci s využitím částí webového rozhraní vidím jako velmi problematickou. Toto řešení by pak bylo nutné upravovat pro každou verzi webového rozhraní.



Obrázek 1.17: Přihlašovací formulář do přepínače HP 5120

SNMP se jeví jako nejlepší varianta pro získání stavových informací o přepínači. Je zde možnost získávat informace buď pomocí dotazů, kdy je na SNMP agenta zaslán dotaz a agent následně vrací odpověď, nebo pomocí oznámení (SNMP trap), kdy agent odesílá informace o změnách daných hodnot. Nejproblematictější část je v tomto případě „dekódovat“ data. Data jsou zde zasílána

1. ANALÝZA



Obrázek 1.18: Výběr portu ve webovém rozhraní 3Com 5500G

ve dvojicích klíč, hodnota. Nejprve je potřeba vybrat z MIB správné klíče a následně je nutné vyřešit párování a propojení dat. Podle specifikace by mělo být možné přepínače přes SNMP také spravovat. K této funkci se mi bohužel nepodařilo najít žádný příklad ani dokumentaci.

Pro získání informací o konfiguraci přepínače pak bude pravděpodobně nutné využít CLI. Bude zde nutné vytvořit skript, který se připojí na CLI, pošle následné příkazy a provede zpracování odpovědí. Z pohledu získání konfigurace se zde nabízejí 2 možnosti. Nejprve je možné zaslat příkaz na vypsání konfigurace a následně zpracovat odpověď. Druhou možností je použít příkaz pro nahrání konfigurace na vzdálený server a následně vycházet ze souborů, které přepínač uložil na zadaný server. Změna konfigurace přepínače by pak vyžadovala zasílat příkazy na CLI a následně uložit novou konfiguraci.

Návrh

2.1 Fyzická struktura sítě

2.1.1 Existující data

K aplikaci AMPTrac ICM se nepodařilo dohledat žádnou vývojářskou dokumentaci. Vzhledem ke způsobu uložení dat v databázi této aplikace jsou data velmi obtížně interpretovatelná. Bohužel se nepodařilo propojit data o patch panelech k příslušným analyzátorům ani najít něco, z čeho by bylo možné určit alespoň pozici patch panelu v rámci analyzátoru. S ohledem na možnou neaktuálnost dat v aplikaci (chybějící nově vytvořené propoje) bylo nejlepším řešením vytvořit data o aktuálním stavu ručně.

2.1.2 Uložení dat

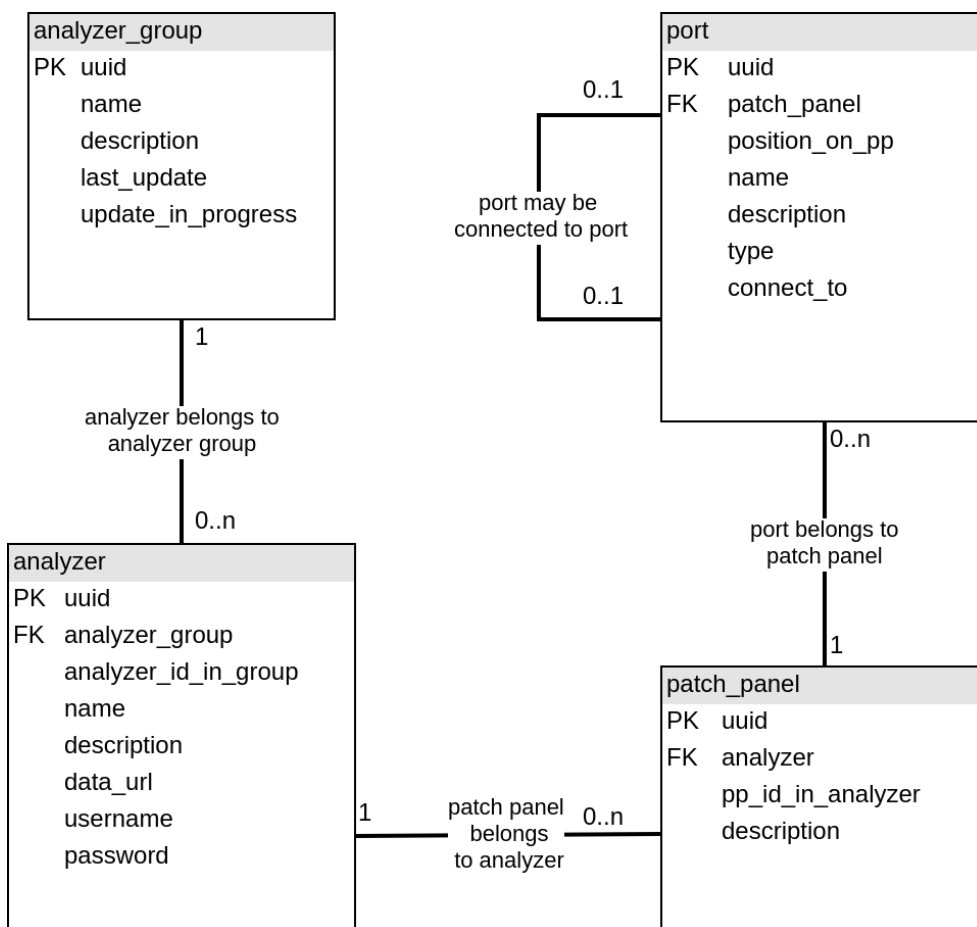
Fyzická struktura sítě je v databázi reprezentovaná pomocí 4 tabulek (viz obr. 2.1).

Základem je tabulka port. Základním parametrem portu je jméno. Jedná se o název, kterým je port označen na patch panelu. Jmenné konvence portů jsem rozebíral v předchozí části mé práce. Druhým parametrem je typ. Porty se dělí do následujících skupin:

- wall - propoj v rámci patra
- cross - metalické propoje mezi rozvodnami
- opt - optické propoje mezi rozvodnami
- sw - propoj, který končí na přepínači v portu, který nemá PoE
- sw-pwr - propoj, který končí na přepínači v portu, který má PoE

Jednotlivé porty patří k nějakému patch panelu. V rámci patch panelu pak mají unikátní pozici. Každý port tedy musí mít vyplněno ke kterému patch

2. NÁVRH



Obrázek 2.1: Databázový diagram fyzické struktury sítě

panelu patří a jaká je jeho pozice v rámci tohoto patch panelu. Dvojice patch panel a pozice pak musí být unikátní.

Propoj mezi porty je realizován pomocí sloupečku `connect_to`. Ten obsahuje ID portu do kterého je daný port připojen.

Druhá tabulka obsahuje záznamy o patch panelech. Za zmínku stojí pouze sloupeček `analyzer`, který určuje, ke kterému analyzátoru patch panel patří, a sloupeček `pp_id_in_analyzer`, který určuje identifikátor patch panelu v rámci analyzátoru. Dvojice `analyzer` a `pp_id_in_analyzer` je pak unikátní.

Záznamy o analyzátorech obsahují jméno analyzátoru, které slouží jako identifikátor, který je snadno čitelný pro člověka. Dále jsou se zde data potřebná k získání dat z webového rozhraní. Prvním z těchto parametrů je `data_url`. Hodnota v tomto sloupečku je url adresa, ke které je potřeba přistoupit, aby bylo možné dostat se k vizualizaci mapy propojů ve webovém

rozhraní analyzátoru. Následují pak parametry `username` a `password`, které jsou použity pro ověřování. Jednotlivé analyzátory se následně shlukují do skupin. Daný analyzátor má pak ve skupině unikátní identifikátor.

Tabulka `analyzer_group` je pak poměrně jednoduchá. Parametr `name` zde opět reprezentuje identifikátor snadno čitelný pro člověka. Dále jsou zde parametry vztahující se k získávání dat ze skupiny analyzátorů. Položka `update_in_progress` je zde, aby bylo možné označit skupiny analyzátorů na níž momentálně probíhá aktualizace. Položka `last_update` pak obsahuje časové razítko z okamžiku, kdy se naposledy podařilo aktualizovat informace o portech připadajících do této skupiny analyzátorů.

2.1.3 Získávání dat

Nejllepší způsob, jak z analyzátorů získat data, je zaslání dotazu na webové rozhraní. Pro tyto účely lze v Pythonu použít modul `request` z knihovny `urllib` nebo knihovnu `requests`. Pro naše potřeby jsou tato dvě řešení z pohledu funkcionality ekvivalentní.

Modul `request` z knihovny `urllib` je sada funkcí a tříd, které usnadňují přístup k URL. Za drobnou výhodu `urllib` lze považovat fakt, že je součástí standardních knihoven pro Python[28].

`Requests` je knihovna pro Python, určená pro komunikaci přes protokol HTTP, vytvořená tak, aby byla snadno použitelná pro lidské bytosti. Její výhodou je uživatelsky přívětivé API. Jelikož není součástí standardní distribuce je nutné ji doinstalovat například z PyPI. Knihovna `requests` je pak doporučována oficiální dokumentací k `urllib.request` pro použití na vysoko-úrovňové HTTP klienty[29].

Parsování HTML je v Pythonu rovněž velmi jednoduché. Lze využít například modul `parse` knihovny `html` nebo knihovnu `BeautifulSoup`. Pro naše potřeby jsou tato dvě řešení z pohledu funkcionality ekvivalentní.

Modul `parse` knihovny `html` definuje nástroje, které lze využít jako základ pro parsování textových souborů ve formátu HTML nebo XHTML. Za výhodu tohoto řešení lze považovat fakt, že je součástí standardních knihoven pro Python[30].

`BeautifulSoup` je knihovna pro získávání dat HTML a XML souborů. Mimo jiné umožňuje snadno pracovat s těmito dokumenty ve stromové podobě. Knihovna není součástí standardní distribuce Pythonu a je tedy nutné ji doinstalovat z PyPI[31].

2.2 Data z přepínačů

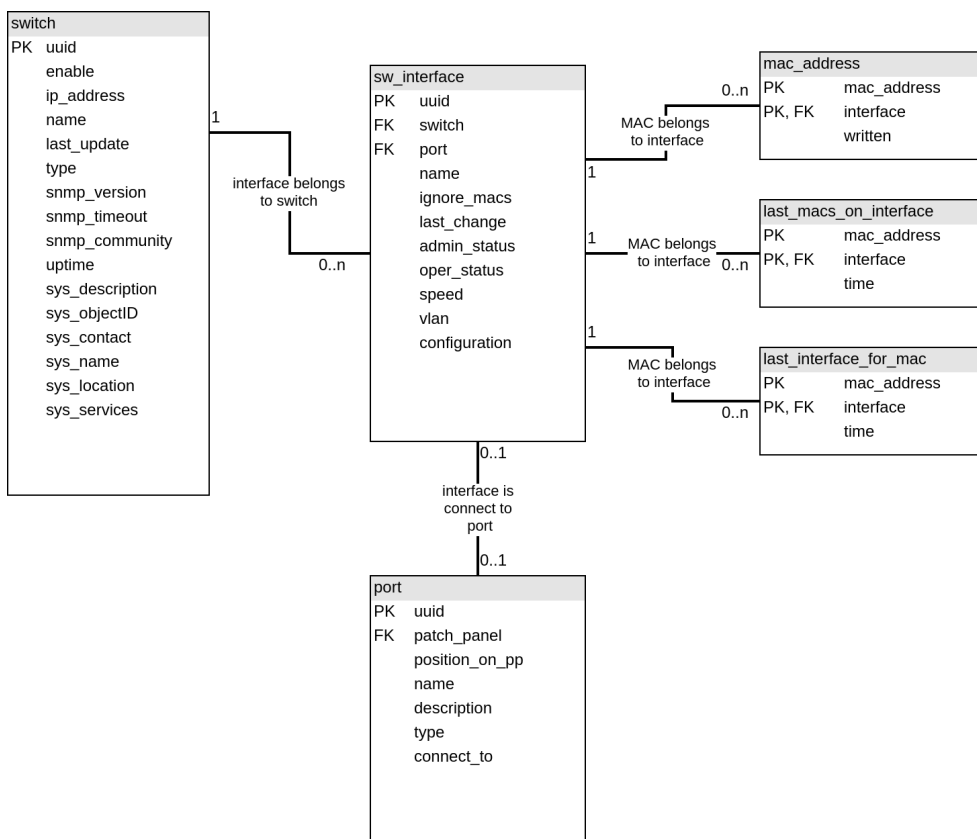
2.2.1 Existující data

Momentálně neexistují žádná data, která by bylo možné transformovat a použít pro inicializaci databáze. Data bude tedy nutné vytvořit částečně nebo

2. NÁVRH

plně ručně.

2.2.2 Uložení dat



Obrázek 2.2: Databázový diagram přepínačů a rozhraní

První položkou kterou bude potřeba uložit je přepínač. Sloupečky tabulky switch lze rozdělit do tří skupin. První skupinou jsou identifikátory. Do této skupiny se řadí sloupečky name, ip_address a type. Sloupeček type by měl obsahovat údaje o typu přepínače. Tyto údaje budou následně využívány při získávání dat. Myšlenka tohoto parametru je taková, že přepínače stejného typu se shodným firmwarem by měly mít stejný typ.

Druhou skupinou jsou parametry vztahující se k nastavení SNMP daného přepínače. Jedná se o sloupečky jejichž název má prefix snmp_ a sloupeček enable. Version a community netřeba blíže vysvětlovat, jelikož se jedná o běžné parametry SNMP, které lze snadno vyčíst přímo z konfiguračního souboru. Význam parametru timeout je délka doby, po kterou se má čekat na první data z daného zařízení. Hodnota tohoto parametru záleží na prostředí a jeho

hodnotu je potřeba určit experimentálně. Vliv na jeho hodnotu, může mít stáří přepínače, velikost stohu nebo parametry síťového připojení. Parametr `enable` nám následně říká, zda-li se mají provádět aktualizace informací o daném přepínači nebo ne.

Poslední skupinou jsou parametry, jejichž hodnota se zjišťuje při aktualizaci. Jedná se o parametry s předponou `sys_` a parametr `uptime`. Jedná se pak o několik základních informací, které je o sobě zařízení schopno sdělit přes SNMP.

Další tabulkou je `sw_interface`, jedná se o tabulku, ve které jsou uloženy informace o jednotlivých rozhraních. Sloupeček `name` v sobě nese informaci, jak se jmenuje dané rozhraní na daném přepínači. Parametr `ignore_macs` nám říká, zda-li se mají na daném rozhraní detekovat MAC adresy. Pro koncová rozhraní bude ve většině případů hodnota `false`. Pro rozhraní, která slouží k vzájemnému propojení jednotlivých přepínačů, bude naopak typická hodnota `true`, jelikož přepínač na svém uplinku vidí všechny zařízení, která nejsou připojena do jeho rozhraní. Parametr `switch` slouží pro párování k přepínači a parametr `port` slouží pro párování k portu na patch panelu. Ostatní sloupečky jsou vyplňovány automaticky při aktualizaci.

Ukládání historie MAC adres je rozděleno do tří tabulek. První tabulkou je `mac_address`, která v sobě nese data z poslední úspěšné aktualizace informací i daném přepínači. Záznam v této tabulce zůstane tedy jen tak dlouho, jak dlouho je MAC adresa aktivní. Pokud tedy zařízení nesoucí danou MAC adresu vypnu, záznam zmizí.

Tabulka `last_macs_for_interface` v sobě nese poslední MAC adresy detekované na daném rozhraní. Pokud tedy například přepojím své PC do jiného rozhraní, tak zde záznam zůstane a bude smazán až v okamžiku, kdy někdo připojí do daného rozhraní jiné zařízení.

Poslední tabulkou je `last_interface_for_mac`, která v sobě nese informace o tom, kde byla naposledy spatřena daná MAC adresa. Je tedy možné například dohledat místo, kde je připojeno vypnuté zařízení nebo zařízení s vadným zdrojem.

2.2.3 Získávání dat

Data z přepínače lze získat přes SNMP pomocí operace *walk*. Operace *walk* se chová tak, že dostane hodnotu kořenu a následně vrátí celý SNMP podstrom případající k tomuto kořenu[32]. Nejsnazším způsobem, jak data získat je použít v příkazové řádce příkaz `snmpwalk` nebo `snmpbulkwalk`. Varianta s použitím příkazové řádky je poměrně problematická, jelikož je z kódu nutné spouštět nový program a následně zpracovávat jeho výstup. Lepší řešení je použít některou z Python knihoven, které slouží pro komunikaci se SNMP. Těmito knihovnami jsou například PySNMP a Easy SNMP. Obě tyto knihovny by měly být pro potřeby aplikace ekvivalentní.

2. NÁVRH

Přepínač je schopný o sobě sdělit poměrně velké množství dat. Pokud se zeptáme na celý strom dostaneme řádově jednotky MB dat. Nás ovšem zajímá pouze několik podstromů. Těmito podstromy jsou[33][34][35][36]:

- 1.3.6.1.2.1.1 - systémové proměnné
 - parametry přepínače začínající předponou sys
 - uptime přepínače v jednotkách Timeticks (10ms, maximum 497 dní)
- 1.3.6.1.6.3.10.2.1.3 - uptime SNMP enginu
 - hodnoty v sekundách (maximum 136 let), větší rozsah oproti Timeticks
 - lze využít jako alternativu k uptime s větším rozsahem
- 1.3.6.1.2.1.2.2.1.5 - rychlost jednotlivých portů
 - datový typ Gauge32
 - 1.3.6.1.2.1.2.2.1.5 . 4227625 = Gauge32: 1000000000
- 1.3.6.1.2.1.2.2.1.7 - administrativní status jednotlivých portů
 - datový typ integer
 - 1.3.6.1.2.1.2.2.1.7 . 4227625 = INTEGER: 1
- 1.3.6.1.2.1.2.2.1.8 - operativní status jednotlivých portů
 - datový typ integer
 - 1.3.6.1.2.1.2.2.1.8 . 4227625 = INTEGER: 1
- 1.3.6.1.2.1.2.2.1.2 - popisky jednotlivých portů
 - datový typ string
 - 1.3.6.1.2.1.2.2.1.2 . 4227625 = STRING: GigabitEthernet1/0/1
- 1.3.6.1.2.1.2.2.1.9 - časové razítko poslední změny stavu portu
 - hodnota uptime (1.3.6.1.2.1.1) přepínače v okamžiku, kdy ke změně došlo
 - datový typ Timeticks (10ms, maximum 497 dní)
 - 1.3.6.1.2.1.2.2.1.9 . 4227625 = 2495809143
- 1.3.6.1.2.1.31.1.1.1.1 - názvy jednotlivých portů
 - datový typ string

- 1.3.6.1.2.1.31.1.1.1.1 . 4227625 = STRING: GigabitEthernet1/0/1
- 1.3.6.1.2.1.17.7.1.4.5.1.1 - hodnota aktuální VLAN na jednotlivých portech
 - datový typ Gauge32
 - 1.3.6.1.2.1.17.7.1.4.5.1.1 . 4227625 = Gauge32: 1
- 1.3.6.1.2.1.17.1.4.1.2 - mapování záznamu v CAM tabulce na rozhraní
 - datový typ integer
 - 1.3.6.1.2.1.17.1.4.1.2 . 1 = INTEGER: 4227625
- 1.3.6.1.2.1.17.4.3.1.2 - mapování záznamu v CAM tabulce na rozhraní
 - datový typ integer
 - 1.3.6.1.2.1.17.4.3.1.2 . 0.36.140.82.153.60 = INTEGER: 1
- 1.3.6.1.2.1.17.4.3.1.1 - záznam MAC adresy v CAM tabulce
 - datový typ hex-string
 - 1.3.6.1.2.1.17.4.3.1.1 . 0.36.140.82.153.60 = Hex-STRING: 00 24 8C 52 99 3C

2.3 Vzory konfigurace přepínačů

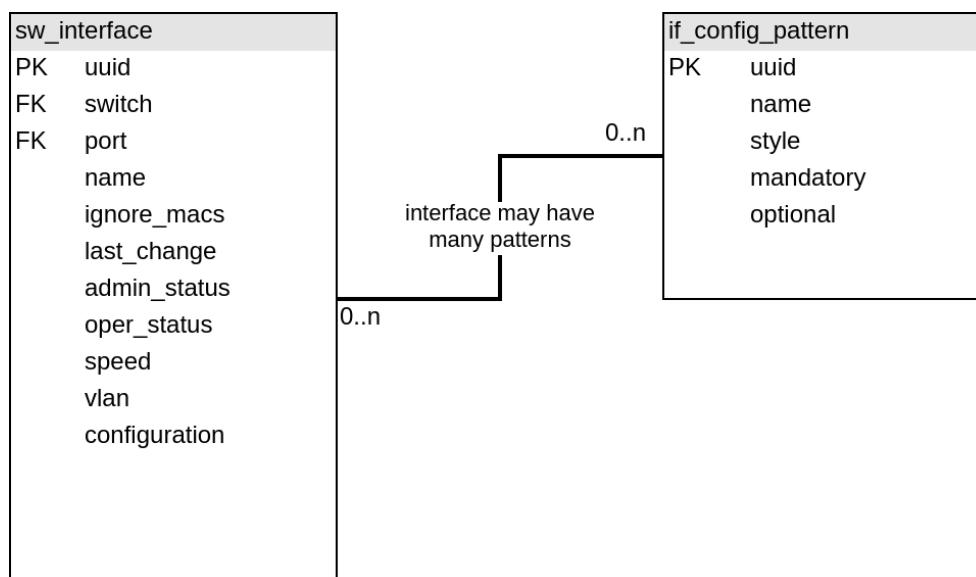
2.3.1 Existující data

Momentálně neexistuje žádný seznam vzorových konfigurací rozhraní přepínačů používaných v NTK. Bude tedy nutné projít konfigurace jednotlivých přepínačů a určit, které nastavení je správné. Ze správných nastavení bude následně nutné sestavit skupiny, které budou odpovídat jednotlivým vzorům.

2.3.2 Uložení dat

Každá konfigurace (viz obr. 2.3) má své jméno, které ve zkratce vystihuje k čemu dané nastavení slouží (tiskárna, WIFI AP, CCTV kamera, ...). Dalším parametrem je položka style. Tento parametr se využívá při zobrazení, kdy se na základě tohoto parametru vzoru přiřadí příslušná CSS třída.

Položky mandatory a optional obsahují pole regulárních výrazů. Regulární výrazy jsou následně využívány při párování rozhraní ke vzorové konfiguraci. Výrazy obsažené v mandatory musí být obsaženy v konfiguraci daného rozhraní. Položky obsažené v optional, pak rozšiřují množinu o položky, které v konfiguraci mohou a nemusí být. Typickým příkladem optional položky je například popis rozhraní (description) nebo povolení PoE.



Obrázek 2.3: Databázový diagram vzorových konfigurací rozhraní

2.3.3 Získávání dat

K získání konfigurace přepínačů lze přistoupit dvěma způsoby. Prvním je získání dat přímo z přepínače. Zde by se jednalo o nutnost automatizovaně se připojit na přepínač a stáhnout si z něj aktuální konfiguraci (*current config*, *running config*) nebo perzistentní konfiguraci (*startup config*). Vzhledem k tomu, že žádný z přepínačů nemá žádné zdokumentované API pro automatizované získání konfigurace, může tato činnost být mírně komplikovaná. Proto je zde druhá možnost, kdy se na data neptáme přímo přepínače, ale nějaké mezivrstvy. Takovou mezivrstvou může být například API, které běží nad úložištěm, kam se zálohuje konfigurace přepínačů.

Získané konfigurace bude následně potřeba nějak zpracovat. Pro zpracování bude nejlepší napsat si pro daný formát jednoduchý parser, který bude umět převést data do nějaké stromové podoby. Pro tyto účely bych použil knihovnu PLY. Pro párování vzorových konfigurací by nám následně měla stačit knihovna re, která je součástí standardní distribuce Pythonu.

2.4 Uživatelské rozhraní

2.4.1 Manipulace s daty

Nově implementované funkce budou vyžadovat rozšíření webového uživatelského rozhraní. Prvním rozšířením, které bude potřeba vyrobit, je rozhraní pro CRUD manipulaci s daty, které je potřeba spravovat ručně.

Za úvahu by stála možnost vytvořit rozhraní pro každou nově vzniklou tabulku. Dle mého názoru má však smysl vytvořit toto rozhraní pouze pro věci, které nejsou statické nebo je nelze detekovat automaticky. Za statické položky považuji například věci spojené s topologií (analyzátoři, patch panely, porty), jelikož od otevření NTK v roce 2009 došlo k minimálnímu množství změn. Za automaticky detekovatelné pak pokládám věci kolem MAC adres. Částečně automaticky, lze také detekovat kolik rozhraní má který přepínač.

Plné rozhraní podporující všechny CRUD operace bude potřeba pouze pro správu přepínačů a pro správu vzorových konfigurací. V případě vzorových konfigurací bude třeba rozhraní ještě rozšířit o možnost přepočítání párování, které se může hodit například při ladění regulárních výrazů apod.

2.4.2 Topologie

V aplikaci AMPTrac ICM bylo zobrazení propojení portů na patch panelech poměrně komplikované (grafická vizualizace cesty, ...). Vzhledem ke zkušenostem s používáním této aplikace bych při zobrazování informací o topologii volil maximálně minimalistický a jednoduchý přístup. Pro zobrazení dat by mělo stačit tabulkové zobrazení rozšířené o možnost filtrovat (viz obr. 2.4). Řádek tabulky by pak měl obsahovat informace o spojení tzn. odkud a kam vede. Dalšími sloupečky, které by tabulka měla obsahovat, by měly být identifikátor rozvodny, typ portu (přepínač, PoE přepínač, optika, ...) a pozice na patch panelu. Za úvahu by stálo uvést také označení patch panelu. Patch panely však nejsou v rozvodnách nijak označeny a není zavedeno žádné pravidlo pro jejich uspořádání.

ADMINATOR Network management someone

Devices Networks Domains Records Leases **Topology**

Topology

5.1.11 🔍

SRC Switch	SRC Name	SRC position	SRC type	DST Switch	DST Name	DST position	DST type
fd5-1	5.1.111	15	wall	fd5-1	5.1.S2.20	20	sw
fd5-1	5.1.S2.20	20	sw	fd5-1	5.1.111	15	wall
fd5-1	5.1.S1.10	10	sw-pwr	fd5-1	5.1.112	16	wall
fd5-1	5.1.113	17	wall				
fd5-1	5.1.112	16	wall	fd5-1	5.1.S1.10	10	sw-pwr

[<< Prev](#) [1](#) [Next >>](#)

Obrázek 2.4: Wireframe pohledu na topologii

2.4.3 Detekované MAC adresy

Pohled na data týkající se detekovaných MAC adres bude asi nejsložitější z pohledu databáze. Nejprve bude potřeba vyřešit to, že máme MAC adresy rozdělené do tří tabulek (`last_interface_for_mac`, `last_macs_for_interface` a `mac_address`). Vyrábět pohled pro každou tabulku zvlášť považuji za zbytečně komplikované. Podíváme-li se, jak vypadají data v jednotlivých tabulkách zjistíme, že pokud sloučíme data v tabulkách `last_interface_for_mac` a `last_macs_for_interface` bude výsledek obsahovat všechna data, která lze nalézt v tabulce `mac_address`. Jako řešení se tedy nabízí vycházet z dat, která vzniknou sloučením těchto dvou tabulek. K těmto datům bude třeba následně připojit informace o rozhraní, přepínači a síťové zásuvce ke kterým se záznam váže. Dále bude také třeba doplnit k MAC adresám informace, které o nich máme ve správě RADIUSu. Výsledné zobrazení dat pak bude ve formě tabulky s možností filtrování (viz obr. 2.5). Každý záznam bude obsahovat poměrně velké množství dat, bude tedy nutné některá data schovat do elementů, které se zobrazí po najetí kurzoru myši.

ADMINATOR Network management someone

Devices Networks Domains Records Leases **MAC history**

MacHistory

91:1b:0e:1c:a2:4

MAC address	Switch	Interface	Port	Device	User	Detected
91:1b:0e:1c:a2:4b	fd6-1	GigabitEthernet1/0/37	6.1.051	PC101	Name Surname	2017-01-14 10:31:03
91:1b:0e:1c:a2:4e	fd5-1	GigabitEthernet2/0/2	5.1.113	PC820		2017-01-14 10:31:09
91:1b:0e:1c:a2:40	fd2-1	GigabitEthernet3/0/8		encoder		2017-01-14 10:31:09
91:1b:0e:1c:a2:44	fd3-3	GigabitEthernet3/0/8	3.3.110	old PC	Name Surname	2016-09-14 10:31:09
91:1b:0e:1c:a2:48	fd3-3	GigabitEthernet3/0/8	3.3.110	new PC	Name Surname	2017-01-14 10:31:09

[<< Prev](#) [1 Next >>](#)

Obrázek 2.5: Wireframe pohledu na detekované MAC adresy

2.4.4 Zobrazení rozhraní přepínače

Pohled pro zobrazení rozhraní přepínače by měl na jedné obrazovce ukazovat data, která by měla shrnovat stav všech rozhraní přepínače (viz obr. 2.6). Zobrazení bude opět formou tabulky s možností vyhledávání. V každém záznamu by pak mělo být vidět jméno přepínače, jméno rozhraní, stav rozhraní, doba

2.4. Uživatelské rozhraní

ADMINATOR Network management someone

Devices Networks Domains Records Leases **Interfaces**

Interfaces

1/0/8 🔍

Switch	Interface	Link	Last change	Speed	Vlan	Port	Patterns	Last data
fd6-1	GigabitEthernet1/0/8	Up	a day ago	1G	3	6.1.051	Standard	2017-01-14 10:31:08
fd5-1	GigabitEthernet1/0/8	Adm. down	a year ago		1		Down	2017-01-14 10:31:07
fd2-1	GigabitEthernet1/0/8	Down	10 months ago		1			2017-01-14 10:31:06
fd3-1	GigabitEthernet1/0/8	Up	35 minutes ago	100M	8	3.1.110	Printer	2017-01-14 10:31:09
fd4-1	GigabitEthernet1/0/8	Down	5 hour ago		1	4.1.110	CCTV	2017-01-14 10:31:09

[<< Prev](#) [1 Next >>](#)

Obrázek 2.6: Wireframe pohledu na rozhraní

od poslední změny stavu, rychlost linky, aktuální VLAN, zásuvka do které rozhraní vede a vzory jimž odpovídá konfigurace rozhraní přepínače.

Zobrazení dalších informací o rozhraní by pak mělo být vidět v detailu (viz obr. 2.7). Detailní pohled by měl umožňovat zobrazit i informace jako konfiguraci, detailnější informace o přepínači (IP, lokace, typ) a MAC adresy detekované na daném rozhraní. Dále by v detailu také mělo být možné vypínat a zapínat detekci MAC adres.

2. NÁVRH

ADMINATOR Network management someone

Devices Networks Domains Records Leases Interfaces

fd6-1 GigabitEthernet2/0/2

Interface	
Name	GigabitEthernet2/0/2
Link	Up
Speed	100M
Last change	10 minutes ago
VLAN	2
Last data	2017-02-24 10:31:03
Port	6.1.005
Patterns	Standard, POE
MAC detection	<input checked="" type="checkbox"/>

Switch	
Name	fd6-1
System name	5500G-FD6.1
IP address	1.2.3.4
Uptime	a year
Last data	2017-02-24 10:31:03
Location	6.25.15
Enable	<input checked="" type="checkbox"/>

MAC history			
MAC address	Device	User	Detected
90:1b:0e:1c:a2:e1	PC123	user 1	2017-02-24 10:31:03
92:1b:0e:1c:a3:ff			2017-02-24 10:31:03
92:1b:0e:1c:a3:22	old PC		2016-09-24 11:31:03

Configuration
<pre>poe enable stp edged-port enable port link-type trunk port trunk permit vlan 1 loopback-detection enable port-security port-mode mac-else-userlogin</pre>

Obrázek 2.7: Wireframe pohledu na detail rozhraní

Realizace

3.1 Topologie

3.1.1 Naplnění databáze

Prvním krokem, který bylo potřeba provést v rámci věci spojených s topologií, bylo naplnění databáze. Vzhledem ke stavu, v jakém byly podklady o stavu sítě, bylo nutné obejít rozvodny a sestavit tato data ručně. V rámci této aktivity vznikly záznamy o:

- 18 rozvodnách
- 25 analyzátořech
- 257 patch panelech
- 5 803 portech

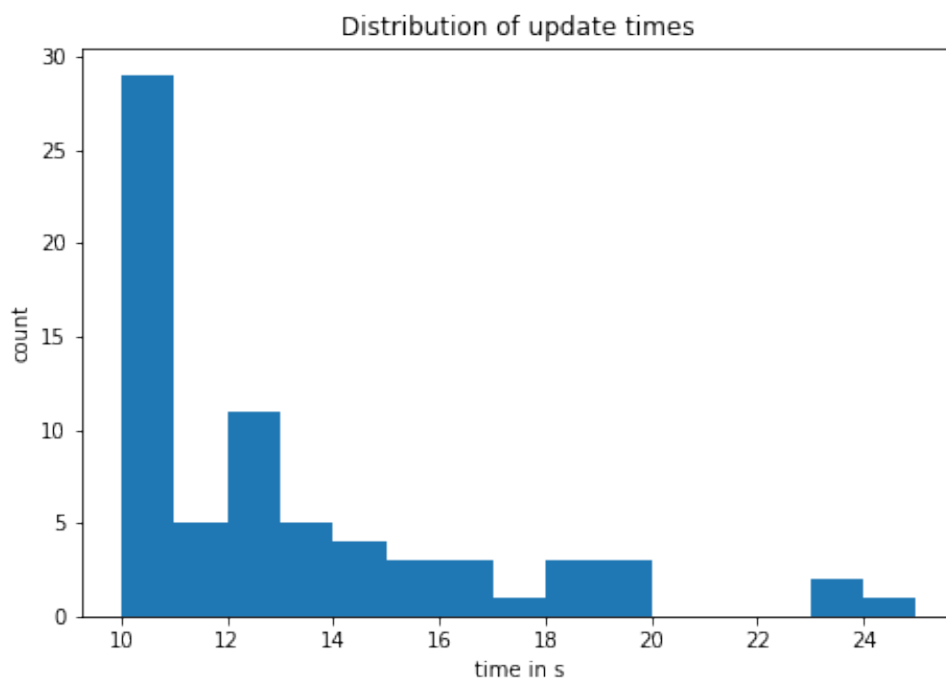
Tato data obsahovala pouze fyzickou reprezentaci statického zapojení sítě. Jednalo se tedy o data které analyzátořy jsou ve které rozvodně, které patch panely patří k danému analyzátořu a který port patří k danému patch panelu. Nejsou zde tedy obsaženy informace o tom, které dva porty jsou propojeny.

3.1.2 Detekce změn

Pro detekci propojení portů se využívá nově vytvořená komponenta topology-daemon. Tato komponenta se periodicky dotazuje jednotlivých analyzátořů a následně aktualizuje záznamy v databázi. Periodu mezi jednotlivými dotazy lze nastavit v konfiguračním souboru.

Dotaz lze rozdělit do několika fází. První fází je otevření *session* a autorizace pomocí základního ověřování (basic access authentication). Po úspěšném ověření je možné provést druhou fází, kterou je získání dat o propojení portů.

3. REALIZACE



Obrázek 3.1: Distribuce trvání aktualizací topologie

V našem případě jsou data ve formě HTML kódu. V obou těchto fázích se využívá knihovny requests.

Po získání dat následuje parsování. Parsování v našem případě znamená získat z HTML dat tabulku s informacemi o patch panelech a portech a tuto tabulku následně převést do podoby, se kterou se v Pythonu lépe pracuje. Parsování je realizováno pomocí modulu html z knihovny parser.

Poslední fází je pak uložení získaných dat do databáze.

Všechny operace jsou výpočetně nenáročné a webové rozhraní analyzátorů odpovídá na dotazy prakticky okamžitě. Aktualizační proces v topology-daemon je tedy realizován sekvenčně, kdy se postupně aktualizují všechny rozvodny.

Jelikož změna propojení vyžaduje fyzickou přítomnost v rozvodně, byla obnovovací perioda zvolena na 5 minut. Čas sekvenční aktualizace při reálném testu (70 měření, 25 analyzátorů) nikdy nepřesáhl 25 vteřin, tudíž by se vše mělo v daném čase stihnout (viz obr. 3.1).

3.2 SNMP data z přepínačů

3.2.1 Naplnění databáze

Pro získávání SNMP dat z přepínačů je potřeba nejprve ručně naplnit tabulku s přepínači. U každého záznamu je nutné vyplnit jméno, IP adresu, typ a parametry pro SNMP. Při první úspěšné aktualizaci dojde k automatickému přidání všech rozhraní přepínače. V případě rozšíření přepínače taktéž dochází k automatickému přidání nových rozhraní. Mazání je nutné provádět ručně, jelikož by bylo velmi nepraktické, kdyby došlo k automatickému smazání všech rozhraní (například při výpadku jedné jednotky stohu).

Další věcí, kterou je třeba udělat ručně je párování rozhraní na port na patch panelu. Tato činnost je prováděna ručně, jelikož není jisté, která rozhraní jsou vyvedena na patch panel a která ne. Následně je třeba u všech rozhraní správně nastavit detekování MAC adres. Na základě dat totiž nelze rozpoznat, která rozhraní přepínače slouží pro koncová zařízení a která se využívají jako uplink nebo k propojení mezi jednotkami stohu. Ve výchozím nastavení je detekce povolena. Pro správné fungování, je třeba ji vypnout na rozhraních, která slouží jako uplink nebo pro stohování. V opačném případě dochází k velké duplikaci detekovaných MAC adres.

Veškerá detekce MAC adres je pak prováděna automaticky. V tabulce `mac_address` dochází k přepisování dat, kdy jsou data vždy přepsána aktuálním stavem. V tabulce `last_macs_for_interface` zůstávají uloženy poslední MAC adresy, které byly na daném rozhraní spatřeny. Poslední tabulka `last_interface_for_mac`, pak drží informaci o posledním místě, kde byla spatřena daná MAC adresa. Byla-li tedy MAC adresa v systému někdy spatřena, máme o ní záznam.

Počty jednotlivých entit pak jsou:

- 23 přepínačů (vlozeno manuálně)
- 2597 rozhraní přepínače (vlozeno automaticky)
 - 464 rozhraní s vypnutou detekcí MAC adres (upraveno manuálně)
 - 1968 rozhraní, které jsou vyvedené na patch panel (částečně automatizováno pomocí skriptu)
- 1350 spatřených MAC adres
- cca 1000 aktivních MAC adres během dne

3.2.2 Detekce změn

K detekci změn dochází tak, že nově vytvořené komponenty provádí periodicky SNMP dotazy na jednotlivé přepínače. Dotazování probíhá pomocí operace *walk*. Za tímto účelem vznikly dvě nové komponenty `snmp-3com-agent-daemon` a `snmp-hp-agent-daemon`. Dotazování v obou komponentách vypadá

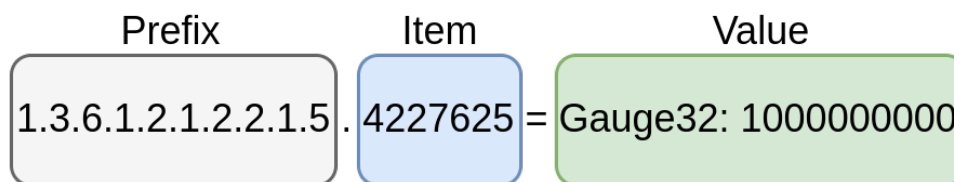
3. REALIZACE

identicky. Jediný rozdíl je ve zpracování dat, jelikož některé konstanty mají různý význam v závislosti na typu přepínače.

Dotazování probíhá pomocí volání příkazu *snmpbulkwalk*. Použití knihoven zde nebylo možné, jelikož si žádná ze zkoušených knihoven nedokázala poradit s OID, která nejsou v rostoucím pořadí (podstromy 1.3.6.1.2.1.17.4.3.1.1 a 1.3.6.1.2.1.17.4.3.1.2). V případě knihovny Easy SNMP se nepodařilo najít parametry, kterými by bylo možno ovládat chování pro tento případ. Bez ošetření tohoto případu, pak knihovna Easy SNMP nebyla schopna dostat některá data.

Knihovna PySNMP tento případ řeší[37]. Řešení uvedené v dokumentaci PySNMP však způsobilo, že dotazování probíhalo velmi pomalu (desítky minut na jeden přepínač) při neúměrném vytížení CPU (jedno jádro 100%). Jediný nalezený způsob, jak bylo možné dostat data v rozumném čase bylo použít *snmpwalk* nebo *snmpbulkwalk*.

Po získání dat probíhá jejich zpracování po jednotlivých podstromech. Nejprve dojde ke zpracování podstromů, které se týkají celého přepínače (1.3.6.1.2.1.1 a 1.3.6.1.6.3.10.2.1.3). V případě těchto podstromů stačí pouze správně namapovat a transformovat hodnoty získané pomocí SNMP na sloupečky databáze.

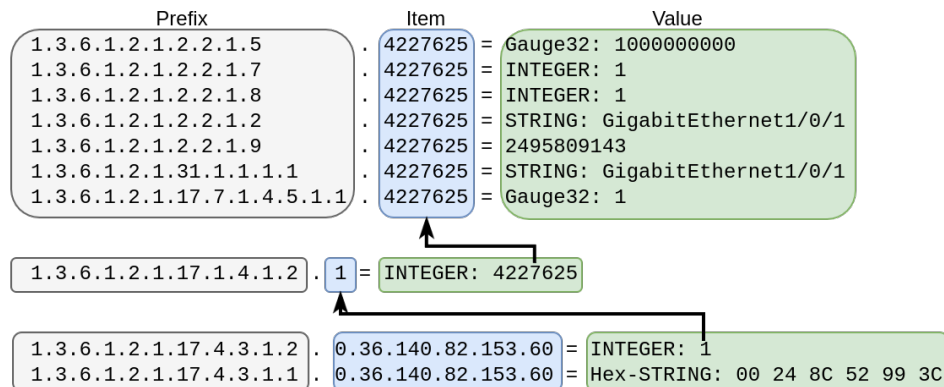


Obrázek 3.2: Dělení dat získaných přes SNMP

Data týkající se rozhraní je nutné nejprve správně poskládat. Každý řádek informací, který dostaneme, lze rozdělit na tři části (viz obr. 3.2)[38]. První částí je prefix, který nám říká, z jakého podstromu hodnota je a tím i co daný řádek znamená. Druhou částí je prvek (item), který nám umožňuje spojovat různé prefixy. Poslední položkou je pak hodnota (value), která nám říká jakou hodnotu má daná vlastnost (prefix) daného prvku. V první fázi skládání se pak provádí složení dat, která mají stejnou indexaci prvků. V druhé fázi pak dochází ke sloučení jednotlivých množin přes mapovací hodnoty (viz obr. 3.3).

Poslední fází je pak uložení získaných dat do databáze.

Při nasazení se ukázalo, že celková doba potřebná pro získání dat z přepínače pomocí *snmpwalk* je poměrně velká. Jako nejproblematictější v tomto ohledu se ukázaly přepínače 3Com 5500G, které jsou poměrně staré a mají celkem slabý procesor. V případě malého nezatíženého přepínače (1 jednotka, 24 portů) se celková doba potřebná pro získání dat pohybovala kolem 30 s. Pro standardní patrový přepínač (3 jednotky, 120 portů) v reálném provozu



Obrázek 3.3: Skládání dat o jednotlivých rozhraních

nebyl problém dostat se na časy kolem 300 s. V případech zvýšené zátěže na daném přepínači nebyl problém, aby se čas potřebný pro získání dat zvýšil dvakrát až třikrát. Celkový čas pro sekvenční získání dat ze všech přepínačů za normální zátěže byl přibližně 45 minut.

Sekvenční řešení tedy nepřipadalo v úvahu a bylo tedy nutné se podrobněji podívat na celý proces získávání dat a maximálně ho optimalizovat.

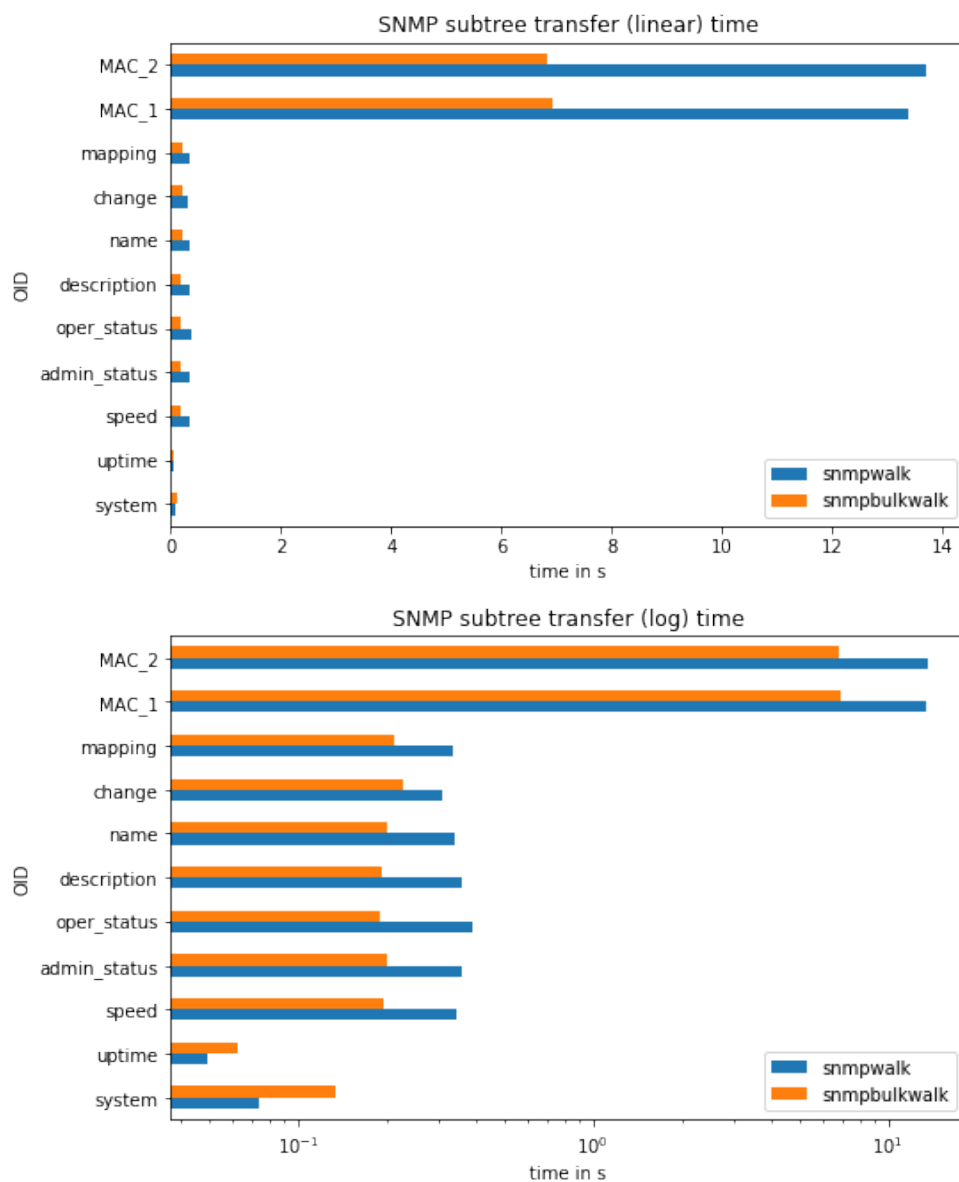
3.2.3 Optimalizace

Prvním krokem byla optimalizace samotného dotazování. Dotazování na SNMP data lze provádět více způsoby. Základní dotazování lze provést pomocí dotazů typu GETNEXT, které v praxi znamenají, že na jeden dotaz obdržíme jednu odpověď. Pokud tedy chceme například získat informaci o rychlosti 24 rozhraní, je potřeba zaslat 24 dotazů a obdržíme 24 odpovědí. Druhou možností je příkaz GETBULK, který umožňuje ptát se na více záznamů najednou[39].

Použití dotazování pomocí GETBULK umožnilo snížit čas potřebný pro získání všech dat přibližně na polovinu (viz obr. 3.4). Tento způsob dotazování umožňuje také manipulaci s parametrem, který říká kolik položek může odpověď obsahovat. Pro hodnoty vyšší než 3 však tento parametr neměl na rychlost měřitelný vliv.

Z grafů je patrné, že většinu času zabere zjišťování MAC adres. Při dotazování se bohužel nelze omezit pouze na rozhraní, která nás zajímají (zapnutá detekce MAC adres), jelikož informace o rozhraní, ke kterému řádek patří, je obsažena v hodnotě (podstrom 1.3.6.1.2.1.17.4.3.1.2). Vzhledem k tomu, že přepínač vidí na uplinku (vypnutá detekce MAC adres) většinu sítě, dochází zde k poměrně velkému plýtvání časem, jelikož spoustu získaných záznamů zahodíme. Fakt, že je nutné ptát se na každou MAC adresu dvakrát (podstrom 1.3.6.1.2.1.17.4.3.1.2 a 1.3.6.1.2.1.17.4.3.1.1), je také problematický.

3. REALIZACE



Obrázek 3.4: Časová náročnost získání jednotlivých SNMP podstromů (malý přepínač 3Com 5500G)

Podíváme-li se však podrobněji na záznamy zjistíme, že prvek v podstromech týkajících se MAC adres je složen ze 6 čísel zapsaných v desítkové soustavě. Po převodu těchto hodnot do hexadecimální číselné soustavy zjistíme, že prvek má stejnou hodnotu jako MAC adresa. Lze tedy odebrat dotaz na podstrom 1.3.6.1.2.1.17.4.3.1.1 a tím snížit čas potřebný pro dotazování přibližně na polovinu.

Výsledkem jsou dvě nové komponenty (`snmp-3com-agent-daemon` a `snmp-hp-agent-daemon`), které periodicky zasílají dotazy na jednotlivé přepínače. Perioda pro zasílání dotazů je 10 minut. Maximální délka dotazu na jeden podstrom je omezena na 2 minuty. Celková doba v takovém případě může být teoreticky až 20 minut (10 podstromů). Kód je však napsán tak, aby se dotazoval nejprve na MAC adresy. Vzhledem k předchozím měřením, ze kterých vyšly dotazy na MAC adresy jako nejdéle trvající (řádově vyšší hodnoty), předpokládám, že pokud doběhne tento dotaz, tak všechny ostatní budou bez problému. Oba parametry (perioda a doba na jeden podstrom) byly voleny s ohledem na to, aby měl vytížený přepínač dostatek času mezi jednotlivými aktualizacemi. Tyto parametry je možné změnit v konfiguračním souboru.

Čas pro sekvenční řešení za normální zátěže byl přibližně 45 minut. Celkový čas pro paralelní získání dat a zapsání do databáze, je pak za normální zátěže obvykle nižší než 3 minuty (doběhnutí aktualizace posledního přepínače). Z distribuce časů potřebných pro získání dat z jednotlivých přepínačů je patrné, že většina aktualizací proběhne za méně než 100 vteřin (viz obr. 3.5).

3.3 Konfigurace přepínačů

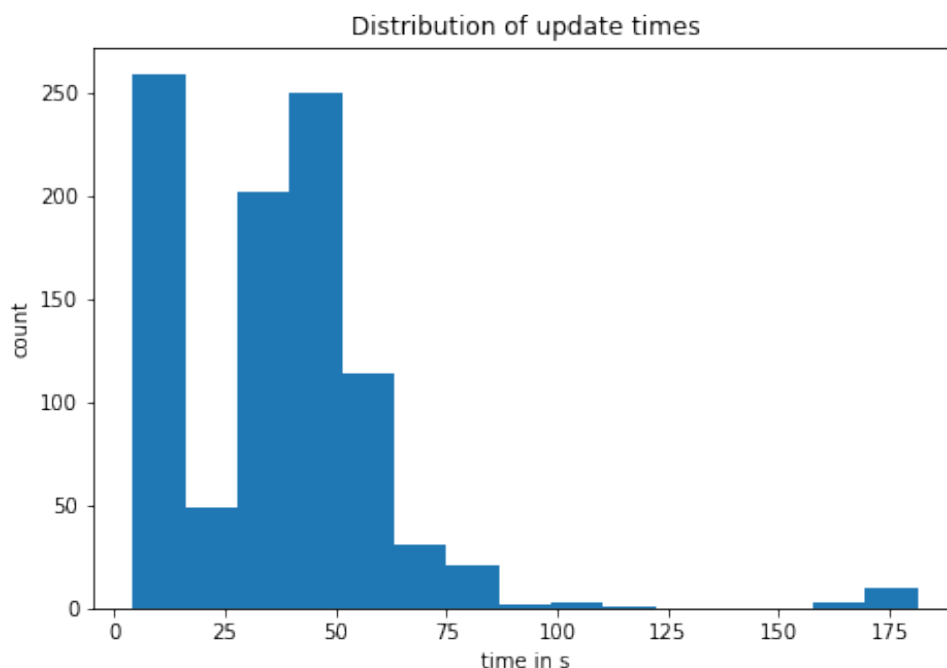
3.3.1 Naplnění databáze

Data potřebná pro tuto část aplikace jsou informace o přepínačích a informace o rozhraních. V obou případech platí, že jsou tato data shodná jako data využívaná v předchozí části.

3.3.2 Získání konfigurace přepínače

Pro získání konfigurace přepínačů byla zvolena možnost využívat middleware. Jako middleware zde slouží vrstva, která zajišťuje zálohování konfigurace přepínačů, nad kterou je postaveno jednoduché RESTful API. Tento middleware je oddělen do samostatného projektu (https://github.com/brabemi/sw_config_backup).

Aplikace zajišťující middleware dělá to, že se periodicky připojuje přes SSH na jednotlivé přepínače. Po úspěšném připojení zde zavolá příkazy pro zálohování *startup* konfigurace na vzdálený TFTP server, který běží na stejném stroji jako middleware. V dalším kroku se provede kopírování konfiguračních



Obrázek 3.5: Distribuce časů pro získání dat z přepínačů přes SNMP

souborů do lokálního GIT⁴ repozitáře, kde se provede automatické zapsání nové verze (*commit*).

Vzhledem k chování připojení přes SSH na přepínačích 3Com byla tato varianta jediná schůdná. Chování přepínačů 3Com je v okamžicích, kdy je CPU přepínače více zatíženo, velmi obtížně predikovatelné. Chápat co přepínač dělá a vědět, které příkazy se provedly, je mnohdy problém i pro zkušeného administrátora. Bylo tedy třeba snížit potřebné množství komunikace mezi programem a přepínačem na minimum. Pro zálohu na TFTP server se nám stačí pouze autorizovat a poslat na přepínač jediný příkaz. Jediným omezením této varianty je, že přes ní není možné získat i *running* konfiguraci, jelikož přepínače podporují pouze zálohu *startup* konfigurace.

Verzování zde pak slouží pro příjemnější práci se zálohami a lepší možnost ladění. API rozhraní následně umožňuje zjistit, kdy u daného přepínače proběhla poslední úspěšná záloha a získat poslední uloženou konfiguraci. Přes API tedy není možné přistupovat k verzi konfigurace z určitého dne apod.

API dále nabízí možnost provádět zálohy. V případě požadavku na zálohu aplikace označí daný přepínač a pošle zprávu o tom, že se má provést záloha. Pokud je přepínač označen a přijde další požadavek na zálohování té

⁴verzovací systém

hož přepínače, nová zpráva se nepošle. Pošleme-li tedy naráz 10 požadavků na zálohování jednoho přepínače záloha se provede pouze jednou. Zprávy jsou předávány pomocí fronty, ze které je odebírají pracovní vlákna, která provádí samotné zálohy.

V reálném nasazení se však ukázalo, že je tato funkce vhodná spíše pro uživatele než pro použití z jiné aplikace. Problém zde činní obtížně predikovatelné chování přepínačů 3Com 5500G. Čas potřebný pro provedení zálohy daného přepínače, zde velmi závisí na počtu jednotek a aktuálních podmínkách na síti (vytížení přepínače, souběžné operace apod.). Jako příklad lze uvést situaci, kdy je na CLI přepínače 3Com připojen někdo přes SSH. Pokud tato situace nastane, je mnohdy velmi obtížné navázat s přepínačem další SSH *session*.

3.3.3 Detekce změn

Aplikace adminator se periodicky dotazuje RESTful API, které je postavené nad zálohami konfigurace. Z tohoto API pak získává konfigurační soubory jednotlivých přepínačů.

Pro zpracování získaných konfiguračních souborů byla zvolena jednoduchá gramatika realizovaná v knihovně PLY, která na základě změn zarovnání řádků v konfiguračním souboru vytvoří strukturu viz obr. 3.6. Z výsledné struktury se momentálně využívají pouze položky vztahující se k jednotlivým rozhraním, ze kterých se bere pole options, které v sobě nese jednotlivé řádky konfigurace daného rozhraní. V případě potřeby by nebyl problém rozšířit aplikaci adminator o možnost detekce definovaných VLAN, RADIUS profilů nebo *access control* listů.

Celý proces od získání dat z API po uložení do databáze zabere přibližně 1 sekundu na jeden přepínač. Aktualizace tedy probíhá sekvenčně. Perioda dotazů na API je 10 minut. Periodické zálohování konfiguračních souborů probíhá jednou za den. Pokud tedy upravím konfiguraci na přepínači je třeba manuálně informovat API, aby provedlo zálohu.

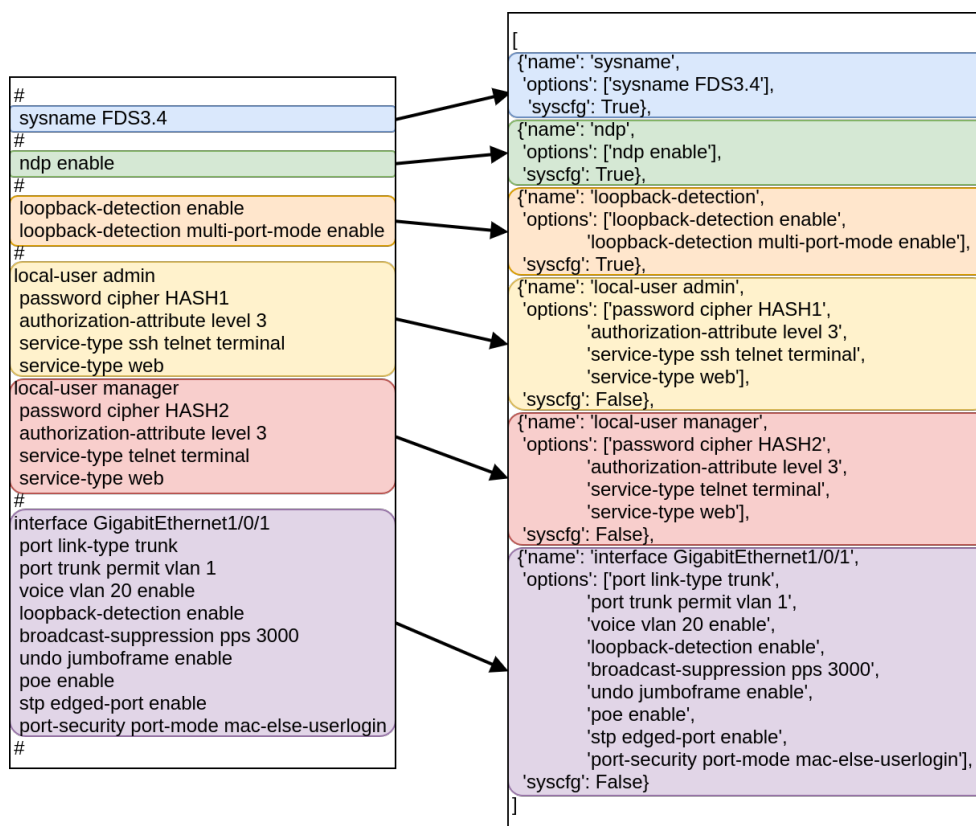
3.4 Vzorová nastavení rozhraní přepínače

3.4.1 Naplnění databáze

Na základě konfiguračních souborů bylo sestaveno 11 vzorových nastavení rozhraní přepínače. Tyto konfigurace pokrývají:

- RADIUS ověřování
- Specifická zařízení - tiskárny, IPTV, CCTV, hodiny
- WIFI AP
- Externí subjekty - ČVUT, FIT ČVUT

3. REALIZACE



Obrázek 3.6: Ukázka parsování konfiguračních souborů

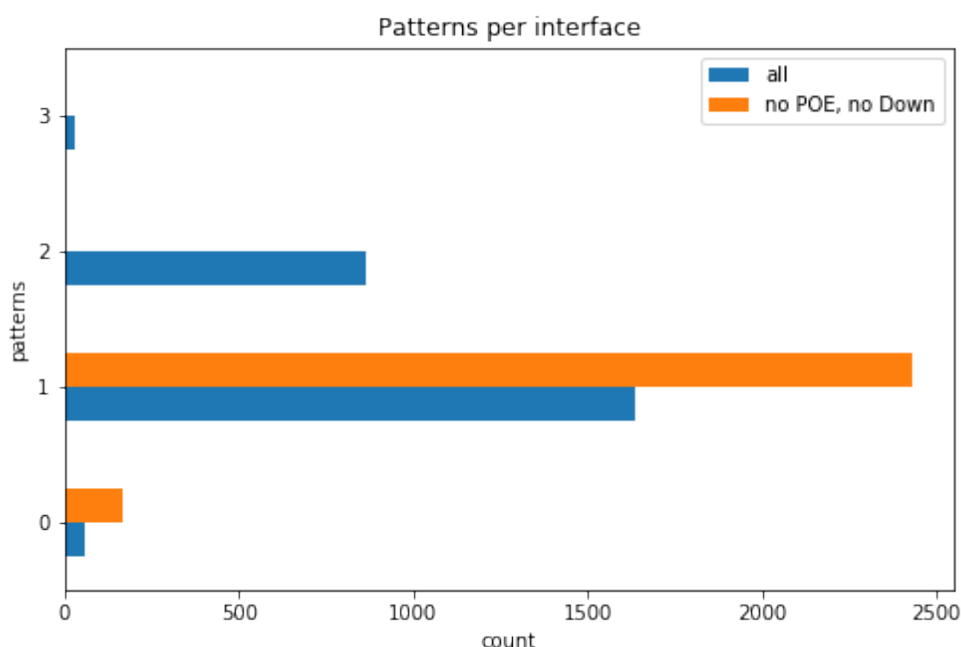
- AV zařízení - Balingův sál

Dále bylo vytvořeno dalších 5 profilů, které slouží pro specifická rozhraní nebo pro položky v konfiguraci na které je dobré upozornit. Zde se jedná o profily pro:

- PoE - rozhraní na kterých je aktivní PoE
- Down - rozhraní, která jsou manuálně vypnutá (stav *administratively down*, například kvůli *dual-personality*)
- LACP - *link aggregation*
- Vlan interface
- Special - aux interface, null interface, loopback, stohování, ...

3.4.2 Výsledky

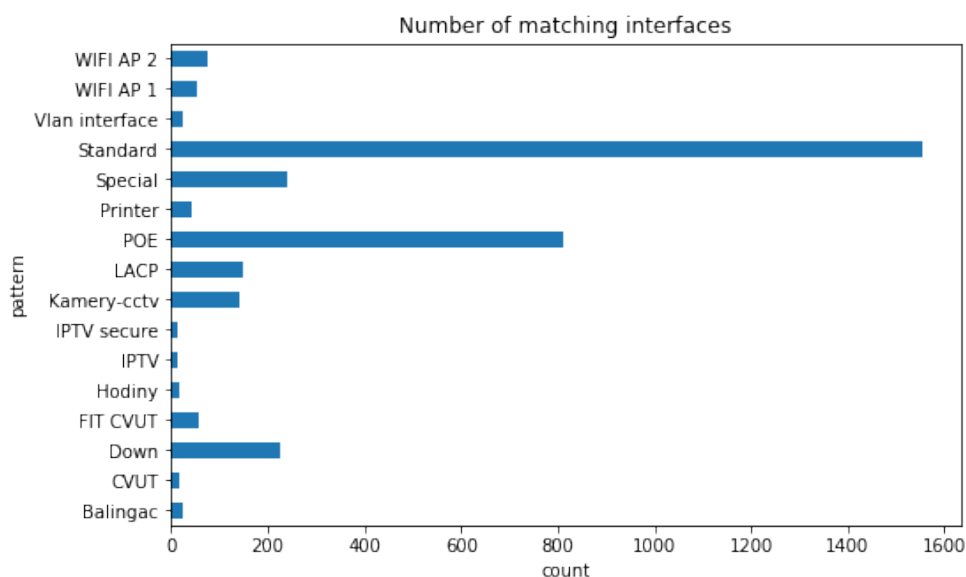
Výsledné pokrytí lze považovat za velmi dobré, jelikož podíváme-li se na počet rozhraní, jejichž konfigurace neodpovídá žádnému vzoru, jedná se o minimální množství (viz obr. 3.7). Naprostá většina rozhraní, pak odpovídá minimálně 1, maximálně 3 vzorovým konfiguracím. Zredukujeme-li množinu vzorů o konfigurace PoE a DOWN (typické hodnoty spadající do optional možností), zjistíme, že žádné rozhraní nemá více než jednu konfiguraci.



Obrázek 3.7: Počet vzorů, které má rozhraní

Rozdělení rozhraní mezi jednotlivé vzorové konfigurace, pak odpovídá reálnému stavu sítě. Největší četnost má konfigurace standard (RADIUS ověřování, viz obr. 3.8). Je zde též vidět, že přibližně třetina rozhraní (celkový počet 2597) podporuje PoE. Z dat by bylo také možné snadno udělat analýzu, která by kontrolovala, zda-li odpovídá počet zařízení s konfigurací WIFI AP skutečnému počtu těchto zařízení apod.

3. REALIZACE



Obrázek 3.8: Počet rozhraní odpovídajících danému vzoru

3.5 Uživatelské rozhraní

V rámci rozšíření UI vzniklo několik nových funkcí. První část která byla přidána je rozhraní pro manipulaci s přepínači (viz obr. 3.9) a vzorovými konfiguracemi (viz obr. 3.10). V obou případech umožňuje rozhraní vytvářet, editovat a mazat data. Za zmínku stojí ještě možnost přepočítat párování vzorové konfigurace rozhraní přímo z webového rozhraní.

Druhou částí je rozhraní pro zobrazování dat o topologii (viz obr. 3.11). Vzhledem k povaze dat je jediná činnost, kterou provádíme s těmito daty, jejich zobrazení. Hlavní funkce tzn. možnost snadno vyhledávat na základě znalosti jména síťové zásuvky je zde realizována pomocí jednoduchého filtrování na úrovni řádků, kdy se ve výsledku zobrazí všechny řádky pro které platí, že některá z jejich položek odpovídá danému výrazu.

3.5. Uživatelské rozhraní

ADMINATOR Network management Someone

Devices Networks Domains Records Leases Topology MAC history Switch interfaces Config patterns Switches

Switches + New switch

Filter Results

Name	Type	Mac detection	IP address	Uptime	System name	System location	System contact	Last update	
fd5-1-1	3com	✓	10.0.100.31	2 years	fd5-1-1	5.43.02	'ICT NTK'	2017-03-10 15:01:45	🗑️
fd4-1	3com	✓	10.0.100.20	2 months	5500G-FD4.1	4.43.02	ICT NTK	2017-03-10 15:01:40	🗑️
fd3-1	3com	✓	10.0.100.17	7 months	5500G-FD3.1	3.43.02	ICT NTK	2017-03-10 15:01:53	🗑️
fd1-2	3com	✓	10.0.100.29	a year	5500G-FD1.2-prednaskovy_sal	1.43.02	ICT NTK	2017-03-10 15:01:49	🗑️
fd2-1	3com	✓	10.0.100.15	2 years	5500G-FD2.1	2.43.01	ICT NTK	2017-03-10 15:01:52	🗑️
fd51-2	3com	✓	10.0.100.30	3 years	5500G-FD51.2-serverovna	01.32.01	ICT NTK	2017-03-10 15:01:48	🗑️
fd5-2	3com	✓	10.0.100.24	2 years	5500G-FD5.2	5.43.02	ICT NTK	2017-03-10 15:01:46	🗑️
fd1-2-1	3com	✓	10.0.100.33	a year	5500G-FD1-2-1-balingac	1.43.02	ICT NTK	2017-03-10 15:01:54	🗑️
fd5-3	3com	✓	10.0.100.25	8 days	5500G-FD5.3	5.43.01	ICT NTK	2017-03-10 15:01:42	🗑️
fd4-2	3com	✓	10.0.100.21	5 months	5500G-FD4.2	4.43.03	ICT NTK	2017-03-10 15:01:43	🗑️
fd3-2	3com	✓	10.0.100.18	2 years	5500G-FD3.2	3.43.03	ICT NTK	2017-03-10 15:01:56	🗑️
fd53-1	3com	✓	10.0.100.13	6 months	5500G-FD53.1	03.22.02	ICT NTK	2017-03-10 15:01:39	🗑️
fd4-3	3com	✓	10.0.100.22	2 years	5500G-FD4.3	4.43.01	ICT NTK	2017-03-10 15:01:54	🗑️
fd51-3	hp	✓	10.0.100.11	4 months	5120-FD51.3	01.43.44	ICT NTK	2017-03-10 13:58:59	🗑️
fd6-3	3com	✓	10.0.100.28	2 years	5500G-FD6.3	6.43.01	ICT NTK	2017-03-10 15:01:51	🗑️
fd2-2	3com	✓	10.0.100.16	2 years	5500G-FD2.2	2.11.07	ICT NTK	2017-03-10 15:01:44	🗑️
fd6-2	3com	✓	10.0.100.27	2 years	5500G-FD6.2	6.43.03	ICT NTK	2017-03-10 15:01:53	🗑️
fd5-1	3com	✓	10.0.100.23	2 years	5500G-FD5.1	5.43.02	ICT NTK	2017-03-10 15:01:39	🗑️
fd51-1	3com	✓	10.0.100.10	2 years	5500G-FD51.1	01.43.16	ICT NTK	2017-03-10 15:01:53	🗑️

Obrázek 3.9: Rozhraní pro správu přepínačů

ADMINATOR Network management Someone

Devices Networks Domains Records Leases Topology MAC history Switch interfaces Config patterns Switches

Interface configuration patterns + New pattern Recalculate all

Filter Results

Name	Style	Mandatory	Optional	
Balingac	primary	stp edged-port enable, broadcast-suppression pps 3000, port access v...	shutdown, poe enable, description .*	🗑️
Down	down	shutdown	.*	🗑️
Special	special			🗑️
CVUT	primary	stp edged-port enable, broadcast-suppression pps 3000, port access v...	shutdown, poe enable, description .*	🗑️
Printer	primary	stp edged-port enable, broadcast-suppression pps 3000, port access v...	poe enable, description .*, mac-address secu...	🗑️
IPTV security	primary	stp edged-port enable, broadcast-suppression pps 3000, port access v...	poe enable, description .*, mac-address secu...	🗑️
		stp edged-port enable, broadcast-suppression pps 3000, port access v...	.*	🗑️
		stp edged-port enable, broadcast-suppression pps 3000, port access v...	shutdown, poe enable, description .*	🗑️
		stp edged-port enable, broadcast-suppression pps 3000, port access v...	shutdown, poe enable, description .*	🗑️
Vlan interface	vlan-interface	ip address 10\.\0\.\100\.\d{1,3} 255.255.255.0		🗑️
Standard	primary	stp edged-port enable, port link-type trunk, port trunk permit vlan ...	shutdown, poe enable, description .*	🗑️
IPTV	primary	stp edged-port enable, broadcast-suppression pps 3000, port access v...	shutdown, poe enable, description .*	🗑️
LACP	lACP	lACP enable	.*	🗑️
Hodiny	primary	stp edged-port enable, broadcast-suppression pps 3000, port access v...	shutdown, poe enable, description .*	🗑️
WIFI AP 1	primary	stp edged-port enable, port link-type trunk, undo port trunk permit ...	shutdown, poe enable, description .*	🗑️
WIFI AP 2	primary	stp edged-port enable, broadcast-suppression pps 3000, port access v...	shutdown, poe enable, description .*	🗑️

🗑️ First ◀ Previous 1 Next ▶ Last 🗑️

Obrázek 3.10: Rozhraní pro správu vzorových konfigurací

3. REALIZACE

SRC switchroom	SRC name	SRC position on panel	SRC type	DST switchroom	DST name	DST position on panel	DST type
fd5-1	5.1.125	5	wall				
fd5-1	5.1.126	6	wall				
fd5-1	5.1.127	7	wall				
fd5-1	5.1.128	8	wall				
fd5-1	5.1.S2.04	4	sw	fd5-1	5.1.124	4	wall
fd5-1	5.1.124	4	wall	fd5-1	5.1.S2.04	4	sw
fd5-1	5.1.S1.04	4	sw-pwr	fd5-1	5.1.123	3	wall
fd5-1	5.1.S2.22	22	sw	fd5-1	5.1.120	24	wall
fd5-1	5.1.S1.03	3	sw-pwr	fd5-1	5.1.121	1	wall
fd5-1	5.1.123	3	wall	fd5-1	5.1.S1.04	4	sw-pwr
fd5-1	5.1.S3.14	14	sw	fd5-1	5.1.129	9	wall
fd5-1	5.1.122	2	wall	fd5-1	5.1.S2.03	3	sw
fd5-1	5.1.129	9	wall	fd5-1	5.1.S3.14	14	sw
fd5-1	5.1.120	24	wall	fd5-1	5.1.S2.22	22	sw
fd5-1	5.1.121	1	wall	fd5-1	5.1.S1.03	3	sw-pwr
fd5-1	5.1.S2.03	3	sw	fd5-1	5.1.122	2	wall

Obrázek 3.11: Rozhraní pro zobrazení topologie

Zobrazení historie MAC adres (viz obr. 3.12) vychází z dat, která vzniknou spojením informací z devíti tabulek v databázi. Vzhledem ke komplexnosti dat, která jsou zde potřeba, je možné data pouze zobrazovat. Vezmeme-li v úvahu, že MAC adresy jsou detekovány automaticky, nedává zde případná možnost editace dat příliš smysl. Z pohledu mazání, pak lze předpokládat, že vznikne spíše potřeba odmazávat staré záznamy. Tuto variantu bude lepší realizovat pomocí periodické operace nebo přidáním dialogu pro smazání záznamů starších než zadané datum. Možnost ručně mazat jednotlivé záznamy tedy také není potřeba. Rozhraní pak umožňuje filtrovat záznamy na základě znalosti:

- MAC adresy zařízení
- statické IP adresy nebo *hostname* zařízení
- rozhraní přepínače do kterého je zařízení připojeno
- síťové zásuvky do které je zařízení připojeno
- jména zařízení
- jména uživatele

Poslední část se týká zobrazení rozhraní. Souhrnné zobrazení (viz obr. 3.13) poskytuje v jedné tabulce přehled o stavu a konfiguraci. Lze zde poměrně snadno vyčíst v jakém stavu (rychlost, VLAN, link status) se rozhraní nachází,

3.5. Uživatelské rozhraní

ADMINATOR Network management Someone

Devices Networks Domains Records Leases Topology **MAC history** Switch interfaces Config patterns Switches

MacHistory

br

MAC address	Switch name	Interface	Port	Description	Owner	Detected
90:1b:0e:1c:a2:e5	fd3-1	GigabitEthernet2/0/5	3.1.046	PC922	Pavel Brhel	2017-03-10 15:01:52
90:1b:0e:1c:a2:e5	No hostname 10.0.1.73	GigabitEthernet2/0/16	4.1.008	PC922	Pavel Brhel	2017-03-10 13:52:12
3c:d9:2b:65:74:09	Dynamic IPv6	GigabitEthernet1/0/41	5.1.107	PC842	Miroslav Brabeneč	2017-01-31 16:22:10
00:90:8f:21:c8:af	zamestnanci-lan (2)	GigabitEthernet2/0/24	3.1.081	IPbrana		2017-03-10 15:01:52
00:90:8f:21:c8:7e	fd6-1	GigabitEthernet2/0/15	6.1.024	IPbrana		2017-03-10 15:01:46
3c:d9:2b:67:22:5a	fd3-1	GigabitEthernet2/0/5	3.1.046	PC848	Miroslav Subrt	2017-03-10 15:01:52
3c:d9:2b:67:22:5a	fd5-1	GigabitEthernet2/0/3	5.1.122	PC848	Miroslav Subrt	2017-03-08 17:38:54
90:1b:0e:1c:a2:e5	fd3-1	GigabitEthernet2/0/43	3.1.133	PC922	Pavel Brhel	2017-02-02 17:16:53
30:85:a9:04:a7:2d	fd5-1	GigabitEthernet2/0/6	5.1.108	Notebook ASUS N56	Miroslav Brabeneč	2017-03-10 15:01:37

1 | First | Previous | Next | Last

Obrázek 3.12: Rozhraní pro zobrazení detekovaných MAC adres

ADMINATOR Network management Someone

Devices Networks Domains Records Leases Topology **MAC history** **Switch interfaces** Config patterns Switches

SwitchInterface

/fd5-1 1/0/1

Switch name	Interface	Link	Last link change	Speed	VLAN	Port	Patterns	Last update
fd5-1-1	GigabitEthernet1/0/1	Up	a day ago	1G	2		Exotic	2017-03-10 15:01:45
fd5-1	GigabitEthernet1/0/1	Up	3 months ago	10M	99	5.1.064	POE Hodiny	2017-03-10 15:01:39
fd5-1	GigabitEthernet1/0/10	Up	3 months ago	100M	1	5.1.063	POE Standard	2017-03-10 15:01:39
fd5-1-1	GigabitEthernet1/0/10	Down	10 months ago				Standard	2017-03-10 15:01:45
fd5-1-1	GigabitEthernet1/0/11	Down	10 months ago				Down LACP	2017-03-10 15:01:45
fd5-1	GigabitEthernet1/0/11	Up	7 months ago			5.1.025	POE WIFI AP 1	2017-03-10 15:01:39
fd5-1	GigabitEthernet1/0/12	Up	3 hours ago	1G	2	5.1.036	POE Standard	2017-03-10 15:01:39
fd5-1-1	GigabitEthernet1/0/12	Adm. down	10 months ago				Down LACP	2017-03-10 15:01:45
fd5-1	GigabitEthernet1/0/13	Up	a month ago	100M	20	5.1.065	POE Standard	2017-03-10 15:01:39
fd5-1-1	GigabitEthernet1/0/13	Down	10 months ago				Standard	2017-03-10 15:01:45
fd5-1	GigabitEthernet1/0/14	Up	7 months ago	100M	2	5.1.071	POE Standard	2017-03-10 15:01:39
fd5-1-1	GigabitEthernet1/0/14	Down	10 months ago				Standard	2017-03-10 15:01:45
fd5-1	GigabitEthernet1/0/15	Up	a year ago	100M	1	5.1.076	POE Standard	2017-03-10 15:01:39
fd5-1-1	GigabitEthernet1/0/15	Down	10 months ago				Standard	2017-03-10 15:01:45
fd5-1	GigabitEthernet1/0/16	Up	2 months ago	1G	2	5.1.082	POE	2017-03-10 15:01:39
fd5-1-1	GigabitEthernet1/0/16	Down	10 months ago				Standard	2017-03-10 15:01:45
fd5-1	GigabitEthernet1/0/17	Down	a day ago			5.1.086	POE Standard	2017-03-10 15:01:39
fd5-1-1	GigabitEthernet1/0/17	Down	10 months ago				Standard	2017-03-10 15:01:45
fd5-1	GigabitEthernet1/0/18	Up	a month ago	1G	2	5.1.092	POE Standard	2017-03-10 15:01:39

Obrázek 3.13: Souhrnné zobrazení rozhraní

3. REALIZACE

ADMINATOR Network management Someone

Devices Networks Domains Records Leases Topology MAC history Switch interfaces Config patterns Switches

fd5-1 GigabitEthernet1/0/41

Interface

Name: GigabitEthernet1/0/41
Link status: **Down**
Link speed:
Last link change: 8 days ago
Detected VLAN: 1
Last update: 2017-03-10 15:01:39
Port: 5.1.107
Config patterns: **POE** **Standard**
MAC detection:

Configuration

```
poe enable
stp edged-port enable
port link-type trunk
port trunk permit vlan 1
broadcast-suppression pps 3000
undo jumboframe enable
voice vlan enable
loopback-detection enable
port-security port-mode mac-else-userlogi
n-secure
```

Switch

Name: fd5-1
System name: 5500G-FD5.1
IP address: 10.0.100.23
Uptime: 2 years
Last update: 2017-03-10 15:01:39
Enable:
Location: 5.43.02

Last interface for MACs

Filter Results

MAC address	Description	Owner	Detected
3c:d9:2b:65:74:05	PC842	Miroslav Brabeneč	2017-01-31 16:22:10

First Previous 1 Next Last

Last MACs on interface

Filter Results

MAC address	Description	Owner	Detected
3c:d9:2b:65:74:05	PC842	Miroslav Brabeneč	2017-01-31 16:22:10

First Previous 1 Next Last

Obrázek 3.14: Detail rozhraní

kdy na něm došlo k poslední změně a jaká je jeho konfigurace (štítky pro jednotlivé vzory). Každé rozhraní je pak odkaz, přes který se lze dostat na detail.

Detailní pohled (viz obr. 3.14) na rozhraní přepínače zobrazuje kromě informací, které byly již v souhrnném pohledu, také přesné znění konfigurace. Dále je zde možnost změnit nastavení detekce MAC adres (jediná věc, kterou je potřeba upravovat ručně). V pohledu je také k vidění stručný souhrn informací o přepínači, ke kterému rozhraní patří, a přehled MAC adres, které byly na daném rozhraní detekovány.

Vyhodnocení

4.1 Adminator

V rámci této práce byla aplikace adminator rozšířena o přibližně 5 500 řádek kódu. Nově vzniklá funkcionality byla pokryta testy napsanými v knihovně pytest.

4.1.1 Back-end

Prvním rozšířením, ke kterému došlo na back-endu, bylo rozšíření databáze. Vzniklo jedno nové schéma, které se skládá z 11 tabulek a 5 pohledů. Toto schéma momentálně obsahuje 16 000 řádků, z nichž bylo přibližně 6 000 řádků vytvořeno manuálně, přibližně 2 500 řádků bylo nutné manuálně upravit a 7 500 řádků bylo vytvořeno automaticky.

Druhým rozšířením, ke kterému došlo na back-endu, bylo vytvoření služeb pro automatickou aktualizaci dat v databázi. Jedná se o služby, které zajišťují získávání dat o propojích, získávání dat o stavu přepínačů a získávání dat o konfiguraci přepínačů. Jedna z těchto služeb pak využívá nástroj, který automaticky zálohuje a verzuje konfigurace jednotlivých přepínačů. Tento nástroj rovněž vznikl v rámci této práce.

Třetím rozšířením, ke kterému došlo na back-endu, bylo rozšíření API. API bylo rozšířeno tak, aby dokázalo správně pracovat s vybranými daty, která jsou nově uložena v databázi.

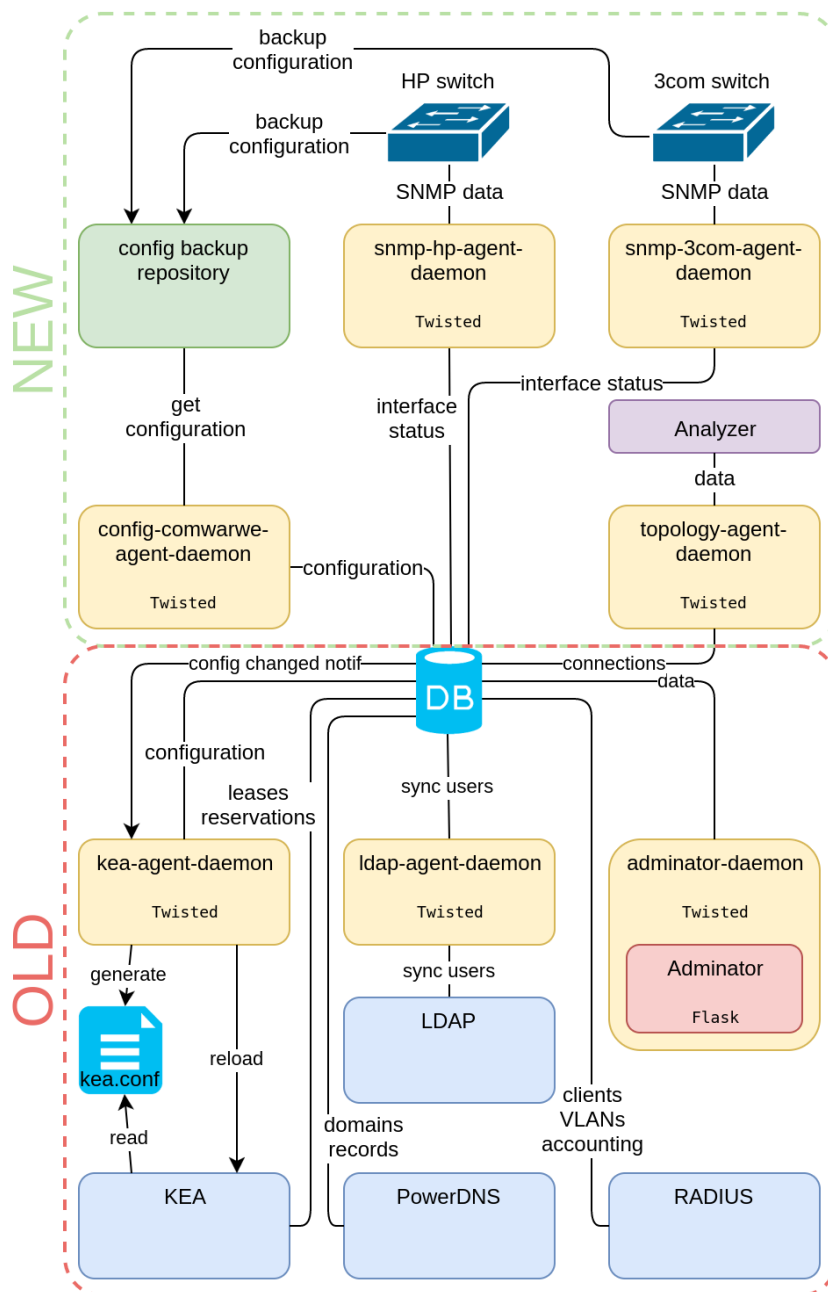
4.1.2 Uživatelské rozhraní

Uživatelské rozhraní aplikace adminator bylo rozšířeno o řadu pohledů. V aplikaci je nyní možné zobrazit a vyhledávat v souhrnných informacích o propojích mezi porty, detekovaných MAC adresách, přepínačích, rozhraních a vzorech konfigurace. V případě přepínačů, rozhraní a vzorových konfigurací je možné zobrazit detail jednotlivých položek.

4. VYHODNOCENÍ

4.1.3 Nasazení

Výsledná aplikace je nasazena na produkčním serveru a struktura aplikace odpovídá diagramu nasazení (viz obr. 4.1).



Obrázek 4.1: Diagram nasazení aplikace

4.2 Případy užití

UC1 - Vyhledání propoje na základě názvu

V propojích je možné vyhledávat pomocí pohledu pro zobrazení topologie. Zobrazené položky je možné filtrovat pomocí jednoduchého formuláře. Filtr vrací všechny řádky, v nichž se nachází daný výraz.

UC2 - Vyhledání propoje pomocí lokace

Vzhledem k problémům s mapovými podklady nebylo možné splnit.

UC3 - Zobrazení stavu daných rozhraní

Stav rozhraní je vidět v souhrnném pohledu. Zobrazené položky je možné filtrovat pomocí jednoduchého formuláře. Filtr vrací všechny řádky, v nichž se nachází daný výraz.

UC4 - Zobrazení konfigurace daných rozhraní

Konfigurace rozhraní je v souhrnném pohledu zobrazena formou štítku. V detailním pohledu na rozhraní je vidět celá podoba jeho konfigurace. Filtrování je možné v souhrnném pohledu na základě názvu vzorové konfigurace. Manipulace s nastavením rozhraní není možná, jelikož žádný přepínač neumí NETCONF ani Python a varianta s použitím CLI je nereálná (chování přepínačů 3Com 5500G).

UC5 - Zobrazení síťových zásuvek propojených do daného rozhraní

Tyto informace lze vidět v souhrnném zobrazení rozhraní a v detailu rozhraní.

UC6 - Lokalizace aktivního zařízení, UC7 - Lokalizace neaktivního zařízení a UC8 - Poslední aktivita zařízení

Tyto informace lze vyhledávat v detekovaných MAC adresách. Záznamy obsahují informaci kdy bylo zařízení na síti spatřeno. Na základě hodnoty této položky pak lze rozhodnout, zda-li se jedná o aktivní zařízení nebo historická data.

UC9 - Poslední zařízení na daném rozhraní

Informace o zařízeních detekovaných na daném rozhraní lze vidět v detailu rozhraní. Tato data jsou také zobrazena v historii detekovaných MAC adres.

UC10 - Přehled detekovaných zařízení

Informace zobrazené v historii detekovaných MAC adres jsou výsledkem spojení dat z devíti tabulek. Všechny požadované informace jsou zde obsaženy. Zobrazené položky je možné filtrovat pomocí jednoduchého formuláře. Filtr vrací všechny řádky, v nichž se nachází daný výraz.

UC11 - Kde bylo naposledy spatřeno zařízení daného uživatele,
UC12 - Kdy bylo naposledy spatřeno zařízení daného uživatele,
UC13 - Konzistence VLAN daného zařízení a UC14 - Konzistence nastavení rozhraní ve kterých jsou připojena daná zařízení

Pro tyto účely nebyly v uživatelském rozhraní vytvořeny pohledy (nebylo požadováno). Dotazy tohoto typu lze v případě potřeby provést nad databází pomocí napsání příkazu v SQL. Databáze obsahuje pohled, který je využíván k zobrazení historie detekovaných MAC adres, jenž je podobně komplexní.

4.3 Testování

Automatické testy jsou napsané v knihovně pytest. Testy jsou rozděleny do souborů, které se vztahují k jednotlivým nově vzniklým modulům. V případě topologie se jedná o testy zaměřené na parsing dat a práci s různě velkými skupinami analyzátorů. V případě modulů pro získání dat přes SNMP jsou testy zaměřeny na parsing dat, skládání dat a zpracování hodnot specifických pro dané typy přepínačů. V případě modulu pro získání konfigurace jsou testy zaměřené na parsing konfiguračních souborů a testování rozpoznávání různých typů vzorů.

Uživatelské testování v klasickém slova smyslu neproběhlo. Důvodem byl velký překryv mezi uživateli, kteří aplikaci adminator používají, a jejími vývojáři. Ladění UI bylo řešeno formou konzultací během návrhu a vývoje. Rozšíření UI byla vytvářena tak, aby maximálně kopírovala vzhled a chování již existujících komponent.

Závěr

Cílem této práce bylo nejprve prostudovat aktuální řešení správy a dohledu (aplikace adminator) nad prvky počítačové sítě Národní technické knihovny a následně navrhnout rozšíření o automatické zjišťování stavu používaných aktivních síťových prvků a zapojení infrastruktury. Nově vytvořená funkcionality měla do aplikace přidat řadu informací, které by administrátorům usnadnily každodenní provoz počítačové sítě.

Z analýzy vzešlo několik případů užití, které bylo třeba realizovat. Realizace byla rozdělena do dvou částí. Nejprve bylo vyřešeno z jakých zdrojů získat data, jak je uložit a jak je udržovat aktuální. V druhé fázi pak následovala implementace jednotlivých funkcí včetně ladění v reálném provozu.

Ve výsledku se s výjimkou jednoho podařilo realizovat všechny případy užití. Případem užití, který se nepodařilo realizovat, je UC2 - Vyhledání propoje pomocí lokace. Problémem zde je formát a neaktuálnost mapových podkladů datových rozvodů v budově NTK.

Rozšířená aplikace adminator nyní umožňuje administrátorům vidět informace o zapojení infrastruktury, informace o MAC adresách detekovaných na síti a informace o konfiguraci rozhraní přepínače. Všechny záznamy je možné jednoduše filtrovat a vyhledávat v nich.

Rozšíření aplikace adminator, která v rámci této práce vzniklá, usnadňuje každodenní správu sítě v NTK. Méně zkušení administrátoři mohou v aplikaci zjistit, jak má vypadat konfigurace pro jednotlivé síťové profily. V aplikaci je možné snadno získat data o využívanosti jednotlivých rozhraní, což umožňuje například uvolnit nevyužívaná rozhraní pro nová zařízení. Vítejným rozšířením je také možnost snadné lokalizace aktivního zařízení včetně možnosti zjištění nastavení rozhraní, do kterého je připojeno. Nově vzniklá data pak mohou snadno posloužit jako podklady pro revizi sítě, podklady pro plánování rozvoje síťové infrastruktury nebo jako podklady pro analýzu využívanosti sítě.

Podklady, které vznikly v rámci této práce, mohou posloužit pro aktualizaci dokumentace stavu sítě v NTK. Jedná se například o diagramy nasazení, síťové diagramy nebo seznam profilů konfigurací rozhraní přepínače.

Do budoucna se nabízí mnoho možností, jak aplikaci adminator dále rozšiřovat. Bylo by příjemné mít v aplikaci například možnost vyhledávání pomocí interaktivní mapy. Dále by bylo zajímavé prozkoumat možnost a vyzkoušet získávání dat ze síťových prvků od různých výrobců nebo navrhnout rozšíření, která by využívala možností, které nám dává NETCONF nebo možnost skriptování v Pythonu na síťových prvcích.

Literatura

- [1] Koch, O.; Koudelka, J.; Dvořák, J.: *GitHub - techlib/admininator [online]*. [cit. 2017-02-31]. Dostupné z: <https://github.com/techlib/admininator>
- [2] 3Com Corporation: *SuperStack® 4 Switch 5500G-EI Family Configuration Guide [online]*. [cit. 2017-02-23]. Dostupné z: http://h20628.www2.hp.com/km-ext/kmcsdirect/emr_na-c02583472-1.pdf
- [3] 3Com Corporation: *3Com® Switch 5500G 10/100/1000 Family - Process Control Outlet [online]*. [cit. 2017-02-23]. Dostupné z: <http://www.pco2.com/v/vspfiles/downloadables/switch%205500g%20family.pdf>
- [4] Hewlett Packard Enterprise: *HPE 5120 EI Switch Series data sheet [online]*. [cit. 2017-02-23]. Dostupné z: <https://www.hpe.com/h20195/v2/GetPDF.aspx/4AA3-0725ENW.pdf>
- [5] Hewlett Packard Enterprise: *JG237A HP 5120-48G-PoE+ EI Switch w/2 Intf Slts [online]*. [cit. 2017-02-23]. Dostupné z: <https://www.senetic.lv/i/pdf/22040209-778.pdf>
- [6] 3Com Corporation: *3Com® Switch S7900E - Senetic [online]*. [cit. 2017-02-23]. Dostupné z: <https://h10145.www1.hp.com/downloads/SoftwareReleases.aspx?ProductNumber=JG237A>
- [7] Hewlett Packard Enterprise: *HPE FlexFabric 5820X 24XG SFP+ Switch / HPE™ [online]*. [cit. 2017-02-23]. Dostupné z: <https://www.hpe.com/us/en/product-catalog/networking/networking-switches/pip.fixed-port-l3-managed-ethernet-switches.7482126.html>
- [8] Hewlett Packard Enterprise: *HTML - HPE.com [online]*. [cit. 2017-02-23]. Dostupné z: <https://www.hpe.com/h20195/v2/gethtml.aspx?docname=c04607918>

- [9] Enns, R.; Bjorklund, M.; Schoenwaelder, J.; aj.: *RFC 6241 - Network Configuration Protocol (NETCONF) [online]*. červen 2011, [cit. 2017-04-24]. Dostupné z: <https://tools.ietf.org/html/rfc6241>
- [10] Fleischer, M.: *AMPTrac - Systém pro monitorování a management fyzické vrstvy [online]*. květen 2005, [cit. 2017-02-23]. Dostupné z: <http://www.svetsiti.cz/clanek.asp?cid=AMPTrac-System-pro-monitorovani-a-management-fyzicke-vrstvy-252005>
- [11] Tyco Electronics: *amptrac - TE Connectivity [online]*. [cit. 2017-02-23]. Dostupné z: http://www.te.com/commerce/DocumentDelivery/DDEController?Action=srchrtv&DocNm=AMPTRAC_Hardware_Manual_0706&DocType=SS&DocLang=EN
- [12] TIBA Produktions und Vertriebs-GmbH: *Infrastructure Configuration Manager (IMC) [online]*. [cit. 2017-02-23]. Dostupné z: <http://www.tiba.at/telekom.php?id=366>
- [13] Tyco Electronics: *Infrastructure Configuration Manager - SPEC SHEET [online]*. [cit. 2017-02-23]. Dostupné z: <http://www.tiba.at/images/database/2014090815235732799.pdf>
- [14] LaForge, A.: *Flash and Chrome - Google Blog [online]*. srpen 2016, [cit. 2017-04-03]. Dostupné z: <https://blog.google/products/chrome/flash-and-chrome/>
- [15] The PostgreSQL Global Development Group: *FAQ - PostgreSQL wiki [online]*. leden 2017, [cit. 2017-02-21]. Dostupné z: <https://wiki.postgresql.org/wiki/FAQ>
- [16] The PostgreSQL Global Development Group: *PostgreSQL: Documentation: 9.6: Release 9.6 [online]*. září 2016, [cit. 2017-02-21]. Dostupné z: <https://www.postgresql.org/docs/9.6/static/release-9-6.html>
- [17] Internet Systems Consortium, Inc.: *Kea Development [online]*. [cit. 2017-02-21]. Dostupné z: <http://kea.isc.org/wiki>
- [18] PowerDNS.COM BV: *PowerDNS documentation [online]*. [cit. 2017-03-14]. Dostupné z: <https://doc.powerdns.com/md/>
- [19] Twisted Community: *TwistedProject - Twisted [online]*. leden 2017, [cit. 2017-02-22]. Dostupné z: <http://twistedmatrix.com/trac/wiki/TwistedProject>
- [20] Ronacher, A.: *GitHub - pallets/flask: A microframework based on Werkzeug, Jinja2 and good intentions [online]*. [cit. 2017-02-22]. Dostupné z: <https://github.com/pallets/flask>

-
- [21] Ronacher, A.: *Foreword — Flask Documentation (0.12) [online]*. [cit. 2017-02-22]. Dostupné z: <http://flask.pocoo.org/docs/0.12/foreword/>
- [22] Bayer, M.: *zzzeek / SQLSoup — Bitbucket [online]*. [cit. 2017-02-22]. Dostupné z: <https://bitbucket.org/zzzeek/sqlsoup>
- [23] Facebook Inc.: *React - GitHub [online]*. [cit. 2017-02-22]. Dostupné z: <https://github.com/facebook/react>
- [24] Shapiro, J.; Chase, J.; Chen, J.: *Libscore [online]*. [cit. 2017-02-22]. Dostupné z: <https://libscore.com/>
- [25] Lanciaux, J.; Lanciaux, R.: *griddle-react [online]*. [cit. 2017-02-22]. Dostupné z: <http://griddlegriddle.github.io/Griddle/>
- [26] PatternFly community: *Get Started | PatternFly [online]*. [cit. 2017-02-22]. Dostupné z: <https://www.patternfly.org/get-started/>
- [27] 2ndQuadrant Ltd: *BDR - Postgres-BDR | 2ndQuadrant [online]*. [cit. 2017-02-22]. Dostupné z: <https://2ndquadrant.com/en/resources/bdr/>
- [28] Python Software Foundation: *21.6. urllib.request — Extensible library for opening URLs [online]*. [cit. 2017-03-02]. Dostupné z: <https://docs.python.org/3/library/urllib.request.html>
- [29] Reitz, K.: *Requests: HTTP for Humans [online]*. [cit. 2017-03-02]. Dostupné z: <http://docs.python-requests.org/en/master/>
- [30] Python Software Foundation: *20.2. html.parser — Simple HTML and XHTML parser [online]*. [cit. 2017-03-02]. Dostupné z: <https://docs.python.org/3/library/html.parser.html>
- [31] Richardson, L.: *Beautiful Soup Documentation [online]*. [cit. 2017-03-02]. Dostupné z: <http://docs.python-requests.org/en/master/>
- [32] Net-SNMP: *Manpage of SNMPWALK [online]*. únor 2002, [cit. 2017-03-02]. Dostupné z: <http://net-snmp.sourceforge.net/docs/man/snmpwalk.html>
- [33] Orange SA: *OID Repository [online]*. [cit. 2017-03-02]. Dostupné z: <http://oid-info.com/get/1.3.6.1>
- [34] Reynolds, J.: *RFC 3232 - Assigned Numbers: RFC 1700 is Replaced by an On-line Database [online]*. leden 2002, [cit. 2017-04-24]. Dostupné z: <https://tools.ietf.org/html/rfc3232>

- [35] IANA: *Structure of Management Information (SMI) Numbers (MIB Module Registrations)* [online]. březem 2017, [cit. 2017-04-24]. Dostupné z: <https://www.iana.org/assignments/smi-numbers/smi-numbers.xhtml>
- [36] McCloghrie, K.; Perkins, D.; Schoenwaelder, J.: *RFC 2578 - Structure of Management Information Version 2 (SMIV2)* [online]. dubem 1999, [cit. 2017-04-24]. Dostupné z: <https://tools.ietf.org/html/rfc2578>
- [37] Etingof, I.: *Dealing with OIDs not increasing error* [online]. [cit. 2017-03-08]. Dostupné z: <http://pysnmp.sourceforge.net/faq/oids-not-increasing.html>
- [38] Rose, M.: *RFC 1158 - Management Information Base for Network Management of TCP/IP-based internets: MIB-II* [online]. březem 1991, [cit. 2017-04-24]. Dostupné z: <https://tools.ietf.org/html/rfc1158>
- [39] Net-SNMP: *Net-SNMP - SourceForge* [online]. únor 2013, [cit. 2017-03-08]. Dostupné z: <http://net-snmp.sourceforge.net/>

Seznam použitých zkratk

AJAX Asynchronous JavaScript and XML

AP Access Point

API Application Programming Interface

AV Audio Video

CAD Computer Aided Design

CAM Content Addressable Memory

CCTV Closed Circuit TeleVision

CLI Command Line Interface

CPU Central Processing Unit

CRUD Create Read Update Delete

CSS Cascading Style Sheets

ČVUT České vysoké učení technické

DHCP Dynamic Host Configuration Protocol

DNS Domain Name System

FIT Fakulta informačních technologií

HP Hewlett-Packard

HTML Hypertext Markup Language

HTTP HyperText Transfer Protocol

HTTPS HTTP Secure

A. SEZNAM POUŽITÝCH ZKRATEK

IPTV Internet Protocol TeleVision

JSON JavaScript Object Notation

LACP Link Aggregation Control Protocol

LDAP Lightweight Directory Access Protocol

MB Mega Byte

MIB Management Information Base

MVC Model View Controller

NTP Network Time Protocol

ORM Object Relational Mapping

OS Operating System

PC Personal Computer

PDF Portable Document Format

PoE Power over Ethernet

PXE Preboot eXecution Environment

PyPI Python Package Index

RADIUS Remote Authentication Dial In User Service

SNMP Simple Network Management Protocol

SSH Secure Shell

STP Shielded Twisted Pair

TFTP Trivial File Transfer Protocol

UC Use Case

UI User Interface

URL Uniform Resource Locator

UTP Unshielded Twisted Pair

UX User eXperience

VLAN Virtual Local Area Network

VPN Virtual Private Network

WSGI Web Server Gateway Interface

XML Extensible Markup Language

Obsah přiloženého CD

readme.txt	stručný popis obsahu CD
src		
_ impl	zdrojové kódy implementace
_ adminator-master.zip	..	zdrojové kódy implementace adminator
_ sw_config_backup-master.zip	zdrojové kódy implementace zálohování switchů
_ example_data		
_ adminator_data.sql	ukázková data adminator
_ adminator_schema.sql	databázové schéma adminator
_ sw_config_backup.sql	..	databázové schéma a ukázková data zálohování switchů
_ thesis	zdrojová forma práce ve formátu L ^A T _E X
text	text práce
_ DP_Brabenec_Miroslav_zadani.pdf	..	zadání práce ve formátu PDF
_ DP_Brabenec_Miroslav_2017_UTF-8.pdf		text práce ve formátu PDF