VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA PODNIKATELSKÁ
FACULTY OF BUSINESS AND MANAGEMENT

ÚSTAV INFORMATIKY
INSTITUTE OF INFORMATICS

NÁVRH IMPLEMENTACE A SÍŤOVÉ BEZPEČNOSTI PROTOKOLU IPV6 V ORGANIZACI

IMPLEMENTATION AND SECURITY OF IPV6 PROTOCOL IN THE COMPANY

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. Jaroslav Hensl

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. Viktor Ondrák, Ph.D.

BRNO 2019

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA PODNIKATELSKÁ

Zadání diplomové práce

Ústav:	Ústav informatiky
Student:	**Bc. Jaroslav Hensl**
Studijní program:	Systémové inženýrství a informatika
Studijní obor:	Informační management
Vedoucí práce:	**Ing. Viktor Ondrák, Ph.D.**
Akademický rok:	2018/19

Ředitel ústavu Vám v souladu se zákonem č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů a se Studijním a zkušebním řádem VUT v Brně zadává diplomovou práci s názvem:

Návrh implementace a síťové bezpečnosti protokolu IPv6 v organizaci

Charakteristika problematiky úkolu:

Úvod
Cíle práce, metody a postupy zpracování
Teoretická východiska práce
Analýza současného stavu
Vlastní návrhy řešení
Závěr
Seznam použité literatury
Přílohy

Cíle, kterých má být dosaženo:

Navrhnout implementaci a zabezpečení protokolu IPv6.

Základní literární prameny:

BOLLAPRAGADA, Vijay, Mohamed KHALID a Scott WAINNER. IPSec VPN design: the definitive design and deployment guide for secure virtual private networks. Indianapolis: Cisco Press, 2005. ISBN 15-870-5111-7.

BOTT, Ed, Carl SIECHERT a Craig STINSON. Mistrovství Microsoft Windows 10. Brno: Computer Press, 2017. ISBN 978-80-251-4869-3.

MCKUSICK, Marshall Kirk a George V. NEVILLE-NEIL. The design and implementation of the FreeBSD operating system. Boston: Addison-Wesley, 2005. ISBN 02-017-0245-2.

SATRAPA, Pavel. IPv6: Internetový protokol verze 6. 3., aktualiz. a dopl. vyd. Praha: CZ.NIC, 2011. ISBN 978-80-904248-4-5.

STANEK, William. Mistrovství v Microsoft Windows Server 2008: [kompletní informační zdroj pro profesionály]. Brno: Computer Press, 2009. ISBN 978-80-251-2158-0.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2018/19

V Brně dne 28.2.2019

L. S.

doc. RNDr. Bedřich Půža, CSc. ředitel	doc. Ing. et Ing. Stanislav Škapa, Ph.D. děkan

ABSTRAKT

Diplomová práce se zabývá nasazením protokolu IPv6 ve firemním prostředí. Práce teoreticky popisuje části protokolu, které jsou pak použity v reálném prostředí s ohledem na maximální kompatibilitu a zabezpečení. Práce řeší výběr hardwaru, konfiguraci routeru se systémem FreeBSD a switchů se systémem RouterOS a navrhuje řešení pro monitoring sítě.

KLÍČOVÁ SLOVA

IPv6, FreeBSD, Mikrotik, RouterOS, firewall, pf, bezstavová autokonfigurace, DHCPv6, zabezpečení, IPSec, monitoring, NetFlow

ABSTRACT

This Master's thesis is focused on the IPv6 protocol implementation in a business environment. The thesis theoretically describes parts of the protocol which are then used in a real environment with respect to maximal compatibility and security. The thesis deals with hardware components, router configuration on FreeBSD OS, and switch configuration with RouterOS. The thesis also proposes how to monitor the network.

KEYWORDS

IPv6, FreeBSD, Mikrotik, RouterOS, firewall, pf, stateless address autoconfiguration, DHCPv6, security, IPSec, monitoring, NetFlow

BIBLIOGRAFICKÁ CITACE

HENSL, Jaroslav. *Návrh implementace a síťové bezpečnosti protokolu IPv6 v organizaci*. Brno, 2019 [cit. 2019-05-08]. Dostupné z: https://www.vutbr.cz/studenti/zav-prace/detail/116001. Diplomová práce. Vysoké učení technické v Brně, Fakulta podnikatelská, Ústav informatiky. Vedoucí práce Viktor Ondrák.

PROHLÁŠENÍ

Prohlašuji, že předložená diplomová práce je původní a zpracoval jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušil autorská práva (ve smyslu Zákona č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským).

Brno . . . .

podpis autora(-ky)

PODĚKOVÁNÍ

Rád bych poděkoval Andree Goldbergerové za to, že mi byla v průběhu psaní této práce jazykově, gramaticky a psychicky nápomocna a podržela mě v momentech, kdy jsem psaní málem vzdal.

Brno ....

podpis autora(-ky)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu práce panu Viktoru Ondrákovi a oponentovi této práce panu Petru Sedlákovi za odborné vedení, ochotu a trpělivost.

Brno . . . .

podpis autora(-ky)

OBSAH

ÚVOD

IP protokol je základním kamenem internetu – aktuálně nejrozšířenější verze protokolu, verze 4, byla definována v roce 1981, tedy více než před 35 lety, což bylo v době, kdy Apple prodával počítač Apple II plus a kdy začínala éra domácích 8bitových počítačů.

Aby tento protokol vůbec byl schopen fungovat v současném světě, muselo být přidáno mnoho rozšiřujících mechanismů, zejména NAT, který sice rozšířil limitující počet 2^{32} adresovatelných zařízení, ale zároveň rozdělil internet na internet vnější a vnitřní sítě. Nato se začaly objevovat mechanismy, které dokázaly tyto vnitřní sítě opět propojovat. Byly vytvořeny protokoly směňující dle potřeby uživatele venkovní provoz dovnitř. Společnosti provozují v internetu prostředníky, umožnující zákazníkům ve vnitřních sítí vzájemně navázat spojení. Nabízí se otázka, co se stane, až začne docházet prostor ve vnitřních sítí. Dáme do vnitřních sítí další vnitřnější sítě? Není konečně čas na změnu?

Protokol IP verze 6 dokáže řešit problémy současného internetu, poskytuje obrovský adresný prostor, poskytuje zabezpečení, anonymizaci, redundanci. Nedokáže však změnit zažitá lidská paradigmata, že musíme vytvářet malé uzavřené ostrůvky síťových zařízení a mezi nimi budovat složité mosty, místo abychom komunikovali přímo. Pokud se bojíme přímého adresování, máme použít firewall a ne schovávat jednotlivé počítače za počítače jiné a tvářit se, že jen takto je to bezpečné.

Tuto práci píši, abych demonstroval, že v roce 2019, tedy více než 20 let poté, co byly zformulovány požadavky na nový protokol, je tento protokol plně životaschopný. Protokol je použitelný v podnikovém prostředí a je schopen plnit současné požadavky kladené na sítě mnohem jednodušeji než jeho předchůdce.

CÍLE PRÁCE, METODY A POSTUPY ZPRACOVÁNÍ

Cílem této práce je návrh zavedení protokolu IPv6 v organizaci Centrum experimentálního divadla, p.o. s důrazem na naplnění požadavků organizace, zabezpečení, a monitoring sítě. Práce si klade za cíl, aby veškerá infrastruktura mohla fungovat pouze na IPv6 (*IPv6-only*). Nemělo by jít pouze o případy, kdy je jeho implementace jednoduchá a na vše podstatné je použit potokol IPv4.

Práce se zabývá i bezpečnostní stránkou. Protokol je navržen tak, aby byla co nejvíce chráněna a anonymizována koncová zařízení, což znesnadňuje správu sítí, jejichž správce je potřebuje kompletně monitorovat a mít přehled o všech zařízeních, která v ní jsou a s kým komunikují. Práce se snaží nalézt řešení i tohoto problému.

Tato práce v hojné míře cituje a odkazuje se na dokumenty RFC - Request For Comments (česky žádost o komentář), což jsou dokumenty popisující zejména internetové protokoly. Ač mají spíše charakter doporučení než norem, řídí se jimi většina internetu. Seznam zmíněných RFC je na konci práce.

1 TEORETICKÁ VÝCHODISKA

V této kapitole popisuji teoretická východiska pro zavedení Internetového protokolu verze 6 (IPv6), věnuji se rozdílům oproti staršímu protokolu IPv4, způsobu, jakým je protokol implementován, zabezpečení protokolu IPv6 a také operačnímu systému FreeBSD, který patří patří ke špičce v implementaci protokolu IPv6.

1.1 Internetový Protokol verze 6

„IPv6 se má stát následníkem nosného protokolu současného Internetu, kterým je IPv4, ve starší literatuře bývá označován též jako *IP Next Generation (IPng)*.“ [1, s. 17]

1.1.1 Historie

Cíle a vlastnosti IPv6 kořeny sahají do začátků devadesátých let, kdy začalo být zřejmé, že se adresní prostor dostupný v rámci IPv4 rychle tenčí. [1, s. 17]

„Tehdy vypracované studie ukazovaly, že s perspektivou přibližně deseti let dojde k jeho úplnému vyčerpání. Jelikož na řešení problému bylo k dispozici poměrně dost času, rozhodlo se IETF navrhnout zásadnější změnu, která by kromě rozšířeného adresního prostoru přinesla i další nové vlastnosti.“ [1, s. 17]

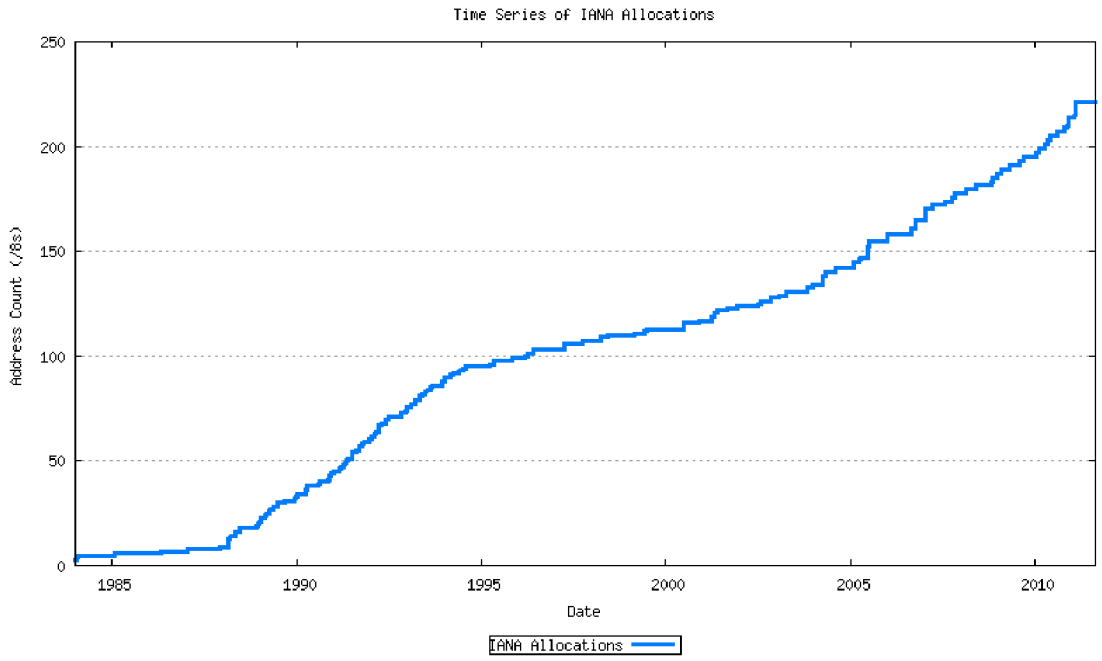
Požadavky na nový protokol byl následující:

- rozsáhlý adresní prostor, který vystačí pokud možno navždy,
- tři druhy adres: individuální (unicast), skupinové (multicast) a výběrové (anycast),
- jednotné adresní schéma pro Internet i vnitřní sítě,
- hierarchické směrování v souladu s hierarchickou adresací,
- zvýšení bezpečnosti (zahrnout do IPv6 mechanismy pro šifrování, autentizaci a sledování cesty k odesilateli),
- podpora pro služby se zajištěnou kvalitou,
- optimalizace pro vysokorychlostní směrování,
- automatická konfigurace (pokud možno plug and play),
- podpora mobility (přenosné počítače apod.),
- hladký a plynulý přechod z IPv4 na IPv6. [1, s. 17]

„Jak je vidět, cíle nebyly skromné.“ [1, s. 17] Koncem roku 1995 je vydáno RFC definujících základ IPv6. Jedná se o RFC 1883: Internet Protocol, Version 6 (IPv6) Specification a jeho příbuzné[2, s. 1]. Finální verze standardu přináší RFC 8200 z července 2017[3, s. 1].

„Oficiální specifikace protokolu tedy byla na stole a mohlo se začít s implementováním a uváděním do života. Jenže nezačalo. IPv6 bylo příliš ožehavou a nejistou půdou, zatímco na poli IPv4 čekaly zisky teď hned. Většina firem se proto věnovala raději snaze o rozvoj IPv4, než aby se angažovala v IPv6, protože návratnost investic byla v prvním případě rychlejší.“ [1, s. 17-18]

„Mimo jiné se podařilo otupit ostří největšího nože na krku IPv4 – nedostatku adres. Začalo se používat beztřídní adresování CIDR, zpřísnila se kritéria pro přidělování síťových adres a byly zavedeny mechanismy pro překlad adres (NAT).“ [1, s. 18]



Obr. 1.1: IANA - alokace /8 bloků

Zdroj: [1, s. 19]

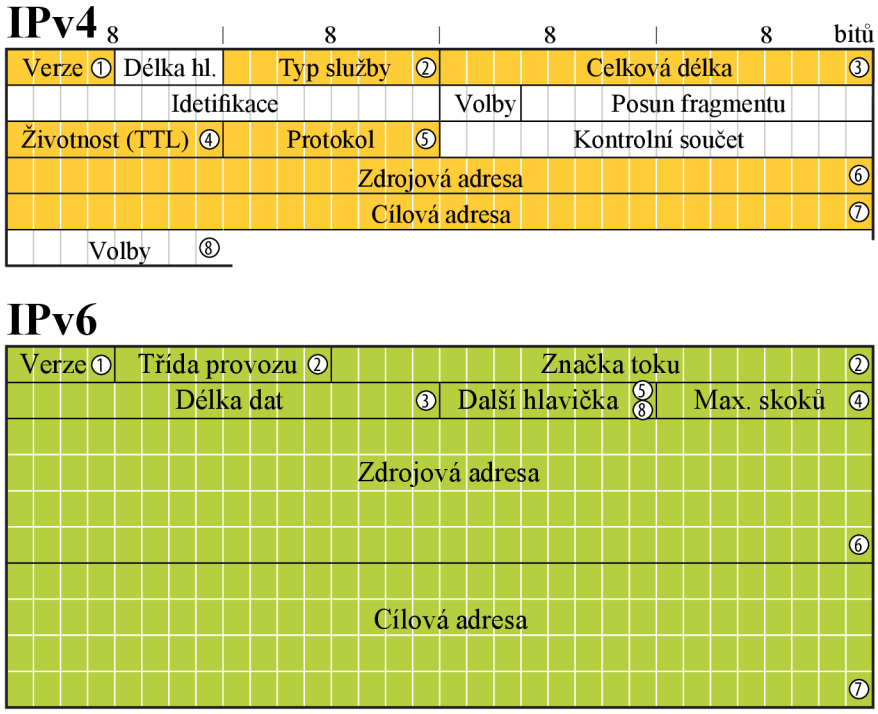
„Tím IPv6 přišlo o svou hlavní hnací sílu a jeho nasazení se začalo odkládat. Aby se dokázalo prosadit do praxe, musí nabídnout nějaké zásadní výhody. Ovšem všechny jeho lákavé vlastnosti byly mezitím implementovány i pro IPv4. Pravda, ne vždy tak elegantně

a zdaleka ne každá implementace je podporuje, ale principiálně jsou k dispozici. A jak již bylo řečeno, většina hráčů na tomto poli preferuje rychlé a velké zisky před vzdálenými a nejistými.“ [1, s. 18]

„Internet si sice našel způsob, jak zpomalit jeho konzumaci, ale i ten má své meze. Obrázek 1.1 ukazuje historický vývoj počtu osmibitových prefixů přidělených jejich centrálním správcem IANA. Je v něm vidět, jak opatření z poloviny 90. let razantně snížila tempo spotřeby, proč prognózy kolem roku 2000 ukazovaly dostatek adres na 20 let a jak později začala křivka zase ošklivě stoupat.“ [1, s. 19]

1.1.2 Paket

Základním kamenem IPv6 je dokument *RFC 1883: Internet Protocol, Version 6 (IPv6) Specification*, s aktuální podobou v RFC 8200, který obsahuje především formát paketu. Ostatním mechanismům a datovým formátům, které souvisejí s IPv6, jsou věnovány další RFC specifikace. [1, s. 35]



Obr. 1.2:

Zdroj: [1, s. 37]

Porovnání hlaviček IPv6 a IPv4 hlaviček je na obrázku 1.2. „Přestože se adresy odesilatele a příjemce prodloužily čtyřikrát, celková délka základní hlavičky pateku vzrostla ve srovnání s IPv4 jen dvojnásobně (z 20 B na 40 B, z toho 32 B zabírají adresy).“ [1, s. 35]

„Při srovnání s IPv4 je nejnápadnější absence tří informací: rozšiřujících voleb, kontrolního součtu a fragmentace. Rozšiřující volby byly nahrazeny obecnějším principem zřetězení doplňkových hlaviček. Obdobně údaje související s fragmentací byly přesunuty do těchto rozšiřujících hlaviček. Zdaleka ne každý paket je totiž fragmentován a lze očekávat, že v IPv6 bude fragmentace ještě vzácnější než v současnosti. IPv6 totiž požaduje, aby infrastruktura pro jeho přenos dovedla přenášet pakety minimálně o délce 1280 B (MTU). Vzhledem k tomu, že drtivá většina koncových zařízení je dnes připojena prostřednictvím různých variant Ethernetu s MTU 1500 B, lze očekávat, že tato maximální velikost paketů se usídlí téměř všude a fragmentace prakticky zmizí ze světa.“ [1, s. 37]

1.1.3 Rozšiřující hlavičky

„IP verze 6 používá odlišný způsob reprezentace rozšiřujících hlaviček než jeho předchůdce. Každá hlavička je nyní samostatným blokem a k jejich vzájemnému propojení slouží položka Další hlavička (Next header). Kód v ní obsažený identifikuje, jakého typu je hlavička, která následuje za tou stávající. Každá rozšiřující hlavička začíná položkou Další hlavička.“ [1, s. 38]

Prostřednictvím těchto hodnot lze za sebe zřetězit neomezeně mnoho hlaviček. Poslední z nich obsahuje v položce Další hlavička typ dat, která paket nese. Hodnota položky *další hlavička* tak zároveň zastupuje dřívější položku Protokol. Nejvýznamnější hodnoty shrnuje tabulka 1.1. [1, s. 38]

„Hlavními devizami koncepce hlaviček v IPv6 je pružnost a úspornost. Součástí paketu jsou jen ty průvodní informace, které skutečně potřebuje. Rubem mince je, že zpracování kompletních hlaviček může představovat průchod relativně dlouhým řetězcem. Pokud by se mělo odehrávat v každém směrovači na cestě mezi odesilatelem a příjemcem, mohlo by to vést k nezanedbatelné degradaci výkonu.“ [1, s. 39]

Tento pořadí hlaviček problém řeší IPv6 následovně – přesně předepisuje pořadí rozšiřujících hlaviček:

1. základní hlavička IPv6,

2. volby pro všechny (hop-by-hop opions),
3. volby pro cíl (destination options) – pro první cílovou adresu paketu,
4. případné další uvedené v hlavičce Směrování,
5. směrování (routing),
6. fragmentace (fragment),
7. autentizace (authentication),
8. šifrování obsahu (encapsulating security payload),
9. volby pro cíl (destination options) – pro konečného příjemce paketu,
10. mobilita (mobility). [1, s. 39]

	Rozšiřující hlavičky
0	volby pro všechny (hop-by-hop options)
43	směrování (routing)
44	fragmentace (fragment)
50	šifrování obsahu (ESP)
51	autentizace (AH)
59	poslední hlavička (no next header)
60	volby pro cíl (destination options)
135	mobilita (mobility)
	Typ nesených dat (protokol)
6	TCP
8	EGP
9	IGP
17	UDP
46	RSVP
47	GRE
58	ICMP

Tab. 1.1: Vybrané hodnoty položky Další hlavička

Zdroj: [1, s. 38]

Bohužel flexibilita hlaviček vytvořila nový bezpečnostní problém, například mezi hlavičku IP protokolu a hlavičku transportního protokolu útočník vloží spoustu dalších nepodstatných hlaviček, takže hlavička transportního protokolu přeteče do druhého fragmentu. Protože firewall vyhodnocuje pakety nezávisle, není z dat prvního fragmentu schopen detekovat použitý protokol. [4]

Tento problém řeší RFC 7112 deklarací, že kompletní řetězec hlaviček *musí* být obsažen v prvním fragmentu. Přijímající zařízení by *mělo* paket nesplňující tuto podmínku odmítnout s vygenerováním patřičné ICMPv6 zprávy. Stejně tak *může* takové pakety odmítat i libovolné mezilehlé zařízení, jako směrovač či přepínač. [4]

1.1.4 Adresy v IPv6

Rychle se tenčící adresní prostor byl jedním z hlavních hnacích motorů vzniku IPv6 [1, s. 55]. Z těchto důvodů byla velikost adresy stanovena na 128 b. Počet adres dostupných v IPv6 je srovnatelný s počtem atomů ve vesmíru, nebo dání každému člověku na zemi přes jednu miliardu adres [5, s. 559].

Existují tři druhy adres s odlišným chováním [1, s. 55]:

Individuální (unicast): každá z nich identifikuje jedno síťové rozhraní a data mají být dopravena právě jemu.

Skupinové (multicast): slouží pro adresování skupin zařízení. Pokud někdo odešle data na tuto adresu, musí být doručena všem členům skupiny.

Výběrové (anycast): představují novinku a nejzajímavější přírůstek v IPv6. Také výběrové adresy označují skupinu, data se však doručí jen jedinému jejímu členovi – tomu, který je nejblíže.

„Porovnání s IPv4 ukazuje, že zmizely všesměrové (broadcast) adresy. Nejsou potřeba, protože jejich funkce přebírají adresy skupinové. Jsou definovány speciální skupiny, např. pro všechny uzly na dané lince, které umožňují plošnou distribuci zpráv.“ [1, s. 55]

IPv6 umožňuje, aby rozhraní mělo libovolný počet adres různých druhů. Přikazuje dokonce několik povinných adres, které musí být přiděleny. Stejně jako v IPv4 se předpokládá, že všechny počítače v jedné fyzické síti (např. na jednom Ethernetu) budou náležet do stejné podsítě a budou tudíž mít společný prefix podsítě. [1, s. 55]

1.1.5 Podoba a zápis adresy

„Standardním způsobem jejího zápisu je osm skupin po čtyřech číslicích šestnáctkové soustavy, které vyjadřují hodnoty 16 bitů dlouhých částí adresy – celková délka adresy je 128 b. Navzájem se oddělují dvojtečkami.“[1, s. 56] Příkladem IPv6 adresy je:

`2001:db8:1234:8765:abcd:rf10:1234:4567`.

Očekává se, že uživatelé budou striktně používat DNS a ručního psaní dlouhých adres budou ušetřeni. [1, s. 56]

„Jelikož je poměrně častou hodnotou nula, nabízí se dvě možnosti pro zkrácení zápisu. Jednak v každé čtveřici můžete vynechat počáteční nuly. Místo `0000` tedy lze psát jen `0`. Někdy se dokonce vyskytuje několik nulových skupin za sebou. Ty můžete nahradit zápisem `::` (dvě dvojtečky).“ [1, s. 56] Kupříkladu adresu:

`2001:0db8:0000:0000:fedc:ba98:7654:3210`

můžete zkrátit na:

`2001:db8:0:0:fedc:ba98:7654:3210`

nebo ještě více na:

`2001:db8::fedc:ba98:7654:3210`.

Konstrukci `::` je možné v každé adrese použít jen jednou, potom by nebylo jednoznačné, jak se má adresa rozvinout do původní podoby [1, s. 56]. Například adresu

`2001:0db8:0000:0000:1234:5678:0000:0000`

lze psát jako:

`2001:db8::1234:5678:0:0`

nebo:

`2001:db8:0:0:1234:5678::`

nikoli však:

,,`2001:db8::1234:5678::`‘‘.

Variabilita zápisu adres člověku znesnadňuje jejich porovnání - VELKÁ/malá písmena a potenciálně zaměnitelné znaky `B` a `8` či `D` a `0`. To vyústilo v RFC 5952 a to definovalo kanonický zápis, jehož cílem je učinit psanou podobu adresy jednoznačnou [1, s. 57]:

- šestnáctkové číslice reprezentované písmeny se píší vždy malými znaky,
- vynechání počátečních nul ve čtveřici je povinné.

- „Konstrukce `::` musí být použita tak, aby měla největší možný efekt. Musí pohltit všechny vzájemně sousedící nulové skupiny (není povoleno `:0::` ani `::0:`) a musí být použita pro nejdelší sekvenci nulových skupin v adrese. Má-li shodnou maximální délku několik skupin, použije se `::` pro první z nich. Není povoleno ji použít pro jedinou nulovou skupinu, ta vždy zůstane jako jednoduchá nula.“ [1, s. 57]

Kanonický zápis předchozího příkladu je tedy:

`2001:db8::1234:5678:0:0`

a SW by ji vždy měl vypisovat v této podobě. [1, s. 57]

V případě URL potřeba adresu uzavřít do hranatých závorek (`[ ]`), aby bylo možné odlišit oddělovač skupin číslic v adrese od čísla portu, pokud je jiný než výchozí. Na například stránku *www.ced-brno.cz* je možné zapsat jako:

`http://[2a02:2b88:2:1::1dd7:1]/`

případně s číslem portu jako:

`http://[2a02:2b88:2:1::1dd7:1]:80/`.

1.1.6 Prefix

Příslušnost k určité síti nebo podsíti se vyjadřuje prefixem – všechna rozhraní v jedné síti mají stejný prefix (začátek adresy). Jeho délka může být různá – záleží na tom, s jakou podrobností se na adresy díváte: může vás zajímat jen prefix poskytovatele Internetu (který bude obvykle 48 – 56 bitů) nebo o poznání delší prefix určité konkrétní podsítě (obvykle nejmenší 64 bitů). [1, s. 58]

Tento přístup se používá již v současném Internetu pod názvem Classless Inter-Domain Routing (CIDR). Z něj je také převzat způsob zápisu [1, s. 58]:

`IPv6_adresa/délka_prefixu`

Část `Délka_prefixu` určuje, kolik bitů od začátku adresy je považováno za prefix. Například 60 bitů dlouhý prefix `2001 db8 0000 abc` lze zapsat několika možnými způsoby:

`2001:db8:0:abc0:0:0:0:0/60`

`2001:db8::abc0:0:0:0:0/60`

`2001:db8:0:abc0::/60`

Nejvhodnější je poslední z nich, protože odpovídá kanonickému tvaru (viz. 1.1.5) a navíc konstrukcí `::` logicky nahrazuje závěrečnou část adresy, která je z pohledu prefixu nezajímavá. Povšimněte si, že do prefixu nepatří ani závěrečná nula ve skupině `abc0`, protože při délce 60 bitů do prefixu z této skupiny patří jen 12 bitů, čili první tři šestnáctkové číslice. Tuto nulu však při zápise nelze vynechat. Prefix pochopitelně nemusí končit na hranici šestnáctkových číslic. Například prefixu `2000::/3` vyhoví všechny IPv6 adresy, jejichž první hexadecimální číslicí je 2 nebo 3. [1, s. 58]

Lze použít i zápis, který současně oznamuje jak konkrétní adresu rozhraní, tak délku prefixu (a tudíž adresu podsítě):

`2001:db8:0:cd30:123:4567:89ab:cdef/64`.

1.1.7 Typy adres

Adresní prostor zahrnuje různé typy adres, aktuální rozdělení tohoto prostoru je v tabulce 1.2.

1.1.8 Objevování sousedů

„Jedním z dobře známých problémů počítačových sítí je zjištění linkové adresy partnera. Počítač potřebuje poslat někomu data, zná jeho IP adresu, podle ní také ví, že spolu sídlí v jedné lokální síti (řekněme Ethernetu). Aby mu však mohl odeslat paket, potřebuje znát právě cílovou ethernetovou adresu.“ IPv4 k tomuto účelu používá samostatný protokol nazvaný ARP. [1, s. 103].

V IPv6 se původně dotyčný mechanismus stal jednou ze základních součástí protokolu. Tato součást se nazývá objevování sousedů (Neighbor Discovery, ND). Slouží k následujícím účelům:

- zjišťování linkových adres uzlů ve stejné lokální síti
- rychlé aktualizace neplatných položek a zjišťování změn v linkových adresách
- hledání směrovačů
- přesměrování
- zjišťování prefixů, parametrů sítě a dalších údajů pro automatickou konfiguraci adresy
- ověřování dosažitelnosti sousedů

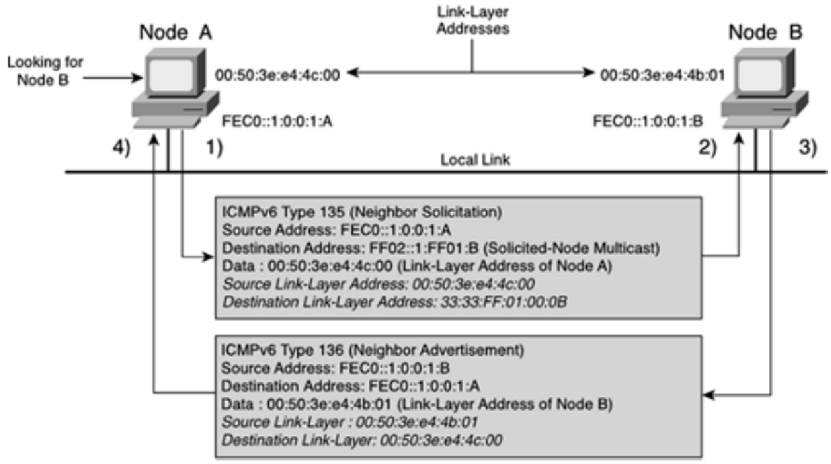
Adresní blok	Jméno	RFC	Routovatelné	Globálně dostupné
::1/128	Loopback Address	RFC 4291	ne	ne
::/128	Unspecified Address	RFC 4291	ne	ne
::ffff:0:0/96	IPv4-mapped Address	RFC 4291	ne	ne
64:ff9b::/96	IPv4-IPv6 Translat.	RFC 6052	ano	ano
64:ff9b:1::/48	IPv4-IPv6 Translat.	RFC 8215	ano	ne
100::/64	Discard-Only Address Block	RFC 6666	ano	ne
2001::/23	IETF Protocol Assignments	RFC 2928	ne	ne
2001::/32	TEREDO	RFC 4380 a RFC 8190	ano	n/a
2001:1::1/128	Port Control Protocol Anycast	RFC 7723	ano	ano
2001:1::2/128	TURN protocol	RFC 8155	ano	ano
2001:2::/48	Benchmarking	RFC 5180, Errata RFC 1752	ano	ne
2001:3::/32	AMT	RFC 7450	ano	ano
2001:4:112::/48	AS112-v6	RFC 7535	ano	ano
2001:5::/32	EID Space for LISP	RFC 7954	ano	ano
2001:10::/28	ORCHID/Deprecated	RFC 4843	-	-
2001:20::/28	ORCHIDv2	RFC 7343	ano	ano
2001:db8::/32	Documentation	RFC 3849	ne	ne
2002::/16 [6]	6to4	RFC 3056	ano	n/a
2620:4f:8000::/48	Direct Delegation AS112 Service	RFC 7534	ano	ano
fc00::/7	Unique-Local	RFC 4193, RFC 8190	ano	ne
fe80::/10	Link-Local Unicast	RFC 4291	ne	ne

Tab. 1.2: Rozdělení adresního prostoru

Zdroj: [6]

- detekce duplicitních adres. [1, s. 103]

Protokol je definován v *RFC 2461* a využívá pro přenos zpráv protokol ICMPv6, jak funguje objevování sousedů je vidět na obrázku 1.3



Obr. 1.3: Objevování sousedů

Zdroj: [7]

1.1.9 Autokonfigurace

„Správce sítě má na výběr dokonce dva typy automatické konfigurace: stavovou a bezstavovou.“ [1, s. 119]

Základem **Stavové konfigurace** je server spravující konfigurační parametry, které pak klientům na požádání sděluje. Podobné existují i v IPv4 jako BOOTP či modernější dnešnímu DHCP. Pro účely stavové konfigurace IPv6 byl navržen protokol DHCPv6. V novějších textech o IPv6 se přestává termín „stavová konfigurace“ používat (prý byl matoucí) a píše se jednoduše o DHCPv6. [1, s. 119]

Princip všech zmíněných mechanismů je podobný jako i DHCP u IPv4: počítač rozešle na obecnou adresu dotaz ohledně svých komunikačních parametrů a server mu je ve své odpovědi sdělí. Obvykle zahrnují vše potřebné pro zapojení do sítě – IP adresu, prefix podsítě, implicitní záznam do směrovací tabulky, adresu DNS serveru a případné další informace. [1, s. 119]

Naproti tomu **bezstavová konfigurace** (Stateless Address Autoconfiguration, SLAAC) představuje zcela nový způsob. Je založena na tom, že v síti jsou směrovače, které vědí vše

potřebné. Proto čas od času všem sdělí, jaká je zdejší situace. Používají k tomu ohlášení směrovače. Nově připojenému zařízení stačí jen chvíli poslouchat, případně o tyto informace aktivně požádat. [1, s. 119]

Hlavním cílem bezstavové konfigurace je automatické určení vlastní adresy uzlu. Jako taková je popsána v RFC 4862 [1, s. 119]

Bezstavová automatická konfigurace dokáže nastavit adresu rozhraní i jednoduchou směrovací tabulku. K funkčnímu zapojení zařízení to sítě však zbývá adresa místního DNS serveru, na nějž se má obracet se svými dotazy. Dlouhá léta se ovšem DNS nacházelo mimo dosah bezstavové konfigurace, jež pro ně nenabízela žádné možnosti. Při současné podobě IPv6 adres je ale počítač bez funkčního DNS téměř nepoužitelný, proto bylo třeba problém nějak řešit. [1, s. 126]

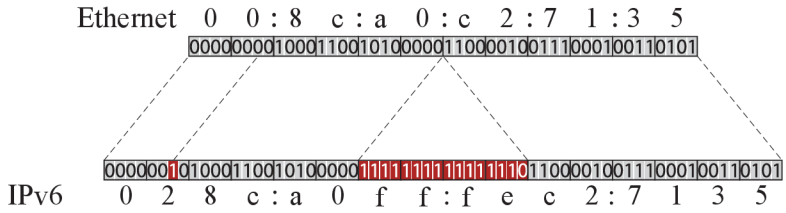
„Jedinou standardní možností bylo nechat bezstavovou konfiguraci být a informace o DNS (i případné další) do ní doplnit stavovou cestou. K tomu slouží příznak `O` v ohlášení směrovače. Nevýhoda tohoto přístupu je zjevná: musíte provozovat další server a v konfiguraci spojovat informace získané různými cestami. V roce 2010 proto vyšlo RFC 6106, jež do automatické bezstavové konfigurace doplnilo volby pro DNS. Jsou dvě, RDNSS poskytuje seznam místních rekurzivních DNS serverů, na něž se klient má obracet se svými dotazy, zatímco DNSSL obsahuje seznam přípon, které může při hledání přidávat na konec relativních doménových jmen“. [1, s. 126] RFC 6106 vylo v roce 2017 nahrazeno RFC 8106. [8, s.1]

„Hlavní nevýhodou bezstavové automatické konfigurace je velmi omezený sortiment informací, které lze jejím prostřednictvím získat. Velký problém byla absence adres místních DNS serverů, a občas by se klientům hodily i jiné věci. Proto obsahuje bezstavová konfigurace možnost, jak doplnit další informace jiným (stavovým) způsobem.“ [1, s. 134]

K tomuto účelu slouží kombinace voleb `M=0` a `O=1` v Ohlášení směrovače. Znamená, že počítač si má adresu a směrování nastavit bezstavově a doplnit k nim další informace získané stavovým protokolem. A právě k tomuto účelu slouží **bezstavové DHCPv6** definované v RFC 3736. Jedná se o velmi zjednodušenou verzi DHCPv6. Z něj přebírá formáty zpráv i pravidla chování všech účastníků, ovšem snaží se o maximální jednoduchost a díky tomu snadnou implementovatelnost. [1, s. 134]

1.1.10 Identifikátory rozhraní

Podoba identifikátoru rozhraní v IPv6 je odvozena z IEEE EUI-64, což je standard globálních identifikátorů pro rozhraní v počítačových sítích [1, s. 61]. V nejčastějších případech jsou sítě založené na Ethernetu nebo bezdrátové IEEE 802.11. V tom případě mají jednotlivá rozhraní výrobcem přidělené jedinečné 48 b MAC adresy. Jejich transformace na modifikované EUI-64 je následující: mezi třetí a čtvrtý bajt MAC adresy se vloží 16 bitů s hodnotou $FFFE_2$ a inventoruje se příznak globality. Na obrázku 1.4 je znázorněné, jak z MAC adresy `00:8c:a0:c2:71:35` se stane identifikátor rozhraní `028c:a0ff:fec2:7135`. [1, s. 62]



Obr. 1.4: Vytvoření modifikovaného EUI-64 z ethernetové adresy

Zdroj: [1, s. 62]

1.1.11 Privacy extension

Klíčová výhoda většího adresního prostoru protokolu IPv6 je v tom, že oproti IPv4 značně komplikuje skenování slabých míst v určitých blocích IP adres. Protokol IPv6 je tedy odolnější proti útočníkům, kteří hledají zranitelné počítače. [9, s. 639].

Avšak adresa vygenerovaná pomocí bezstavové autokonfigurace obsahuje identifikátor rozhraní, který se v čase nemění. Kdykoliv je stejný identifikátor použit ve více kontextech, je možné sledovat zdánlivě nesouvisející aktivity pomocí tohoto identifikátoru. [10, s. 4]

Protože identifikátor je součástí IPv6 adresy, které je základním kamenem komunikace, není snadné tento identifikátor skrýt. Proto RFC 4941 stanovuje způsob generování náhodných identifikátorů, které nemají dlouhou životnost. [10, s. 5]

Důležité je, že RFC 4941 stanovuje, že koncový uživatel má možnost toto chování vypnout. [10, s. 15]

1.1.12 Bezpečnost

Protokol IPv6 má z pohledu jednoduchého přenosu paketů podobné bezpečnostní prvky jako IPv4, bezpečnostní problémy zahrnují[3, s. 30]:

- **Odposlech**, kdy zařízení na cestě může zachytit celý paket (tzn. hlavičky včetně dat).
- **Opakování**, kdy útočník zaznamená sekvenci paketů a pak je ještě jednou pošle straně, která je už jednou dostala.
- **Vkládání**, kdy útočník může do cesty vkládat další pakety s jeho vlastním obsahem.
- **Mazání**, kdy útočník na cestě blokuje některé pakety na neposílá je dál.
- **Modifikaci**, kdy útočník zachytí paket, modifikuje ho a pošle dál modifikovaný.
- **Man-in-the-middle útok**, kdy útočník podvrací komunikační proud mezi odesílatelem a příjemcem a zpět mezi příjemcem a odesílatelem.
- **Denial-of-service (DoS)**, kdy útočník vytváří velké množství (zdánlivě) legitimního provozu a zahltí cíl.

Pakety mohou být proti odposlechu, opakování, vkládání, mazání a modifikaci v IPv6 chráněny mechanismem IPSec[3, s. 30]. Kromě toho protokoly vyšší vrstvy jako Transport Layer Security (TLS) nebo Secure Shell (SSH) mohou být použity pro ochranu provozu, který běží nad IPv6[3, s. 30].

Proti útoku DoS neexistuje žádný mechanismus, obrana proti tomuto typu útoku je mimo specifikaci protokolu IPv6. [3, s. 30]

Protože IPv6 poskytuje *mnohonásobně*[1] větší prostor, než protokol IPv4, je mnohem těžší ho celý proskenovat a dokonce je problematické i proskenování jediné sítě. [3, s. 30]

Předpokládá se, že adresy koncových uzlů budou více viditelné z internetu v porovnání s IPv4 díky odstranění mechanismu NAT, což vytváří další problémy se soukromím a také je je jednodušší rozlišit koncové body. [3, s. 31]

Design rozšiřujících hlaviček v IPv6 přidává hodně flexibility, ale zároveň vytváří další bezpečnostní problémy. Problémy s fragmentací hlaviček považuje dokument RFC 8200 za vyřešenou, vytváření nových budoucích rozšiřujících hlaviček bude vyžadovat zvážení bezpečnostních dopadů na to, jak nová rozšiřující hlavička koexistuje s stávájícími rozšířeními. [3, s. 31]

[1] Rozdíl mezi 2^{32} a 2^{128}

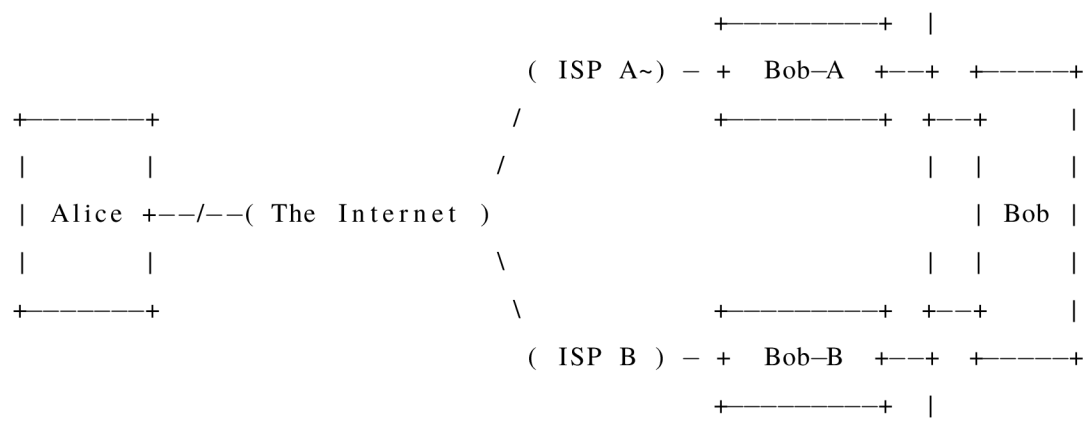
1.1.13 Redundance

Jedno rozhraní může mít více adres a více výchozích bran. Ve výchozím nastavení se zařízení naučí všechny aktivní prefixy z ICMPv6 zpráv typu *router advetisment*. [11, s. 15]

Aby koncové zařízení poznalo, kterou cestu použít, využije podle RFC 4861 tzv. detekci nedostupnosti souseda (Neighbor Unreachability Detection), což není algoritmus, ale spíše soubor doporučení. Podle tohoto doporučení by každá cesta v mezipaměti zařízení měla mít jeden z pěti stavů: nekompletní (incomplete), dostupná (reachable), stará (stale), zpožděná (delay) a prozkoumávat (probe). [11, s. 70]

Informaci o dostupnosti cesty lze získat dvěma způsoby: 1) z vyšších vrstev (např. TCP), pokud se podaří spojit spojení přes tuto cestu. 2) zařízení dostalo ICMPv6 zprávu *Neighbor Advertisement* v reakci na zprávu *Neighbor Solicitation*[11, s. 69]. Jak často má koncové zařízení zkoumat stav svého souseda (routeru) je možné ovlivnit konfigurační volbou `ReachableTime` udávanou v milisekundách.[11, s. 72]

Podle RFC 4861 zprávy *Router Advertisement* neobsahují priority, protože priorita není potřebná pro vypořádání se s routery s různou stabilitou [11, s. 16]. Ale starší RFC 4191 stejně zavádí do RA hlavičky prioritu (velikost 2 b)[12, s.4]: 01_2 vysoká, 00_2 střední a 10_2 nízká[12, s. 5]. Celkový výběr routeru z pohledu koncového zařízení shrnuje RFC 8028[13, s. 1].



Obr. 1.5: Příklad z RFC 8028

Zdroj: [13, s. 7]

Pokud vezmeme zapojení z obrázku 1.5. Pokud Bob odpovídá na zprávu od Alice,

musí použít adresu, na kterou mu přišla. Pokud by router, ze kterého byl prefix, z něhož pocházela tato adresa nedostupný, nemůže jí odpovědět. Pokud Bob inicializuje komunikaci, může si vybrat, který router (Bob-A nebo Bob-B) použije. Vybírá podle kritéria, zda je router dostupný, zda mu sděluje prefix (životnost prefixu nesmí být nulová) nebo podle priority. Bob ale musí použít tu adresu jako zdrojovou, kterou mu přidělil vybraný router. [13, s. 7]

1.2 Operační systém FreeBSD

Systémy založené na jádře *BSD* (FreeBSD, OpenBSD, NetBSD) patří ke špičce s implementací protokolu IPv6[1, s. 317]. V této práci popisuji systém FreeBSD, který se z rodiny *BSD* systémů vyznačuje nejlepší podporou aktuálního a nového HW.

Pokud má čtenář zkušenosti s distribucemi na jádře Linux, je tu zásadní rozdíl mezi linuxovými distribucemi a BSD systémy – linuxové distribuce jsou celé tvořeny tzv. balíčky – SW upravený na míru distribuce a je možné s nimi manipulovat samostatně. Jednotlivé balíčky se instalují do kořenových adresářů (`/usr`) a je úkolem balíčkovacího systému poznat, který soubor patří je kterému balíčku, a řešit aktualizace a konflikty. Kdežto BSD systémy obsahují základní systém, který je distribuovaný jako jeden celek. Další externí software je instalovaný mimo tento základní systém (obvykle `/usr/local`). Základní systém je tak přesně daný a nevzniká tak obrovské množství nekompatibilních konfigurací.

1.2.1 Historie

Počátky systému FreeBSD můžeme vysledovat někdy na začátku roku 1993, kolem neoficiálního *386BSDPatchkit*. Základní myšlenkou bylo opravit problémy, které samotný *patchkit* nebyl schopen vyřešit. [14]

Aby se systém odlišil od *386BSD*, osvojil si jméno FreeBSD a jeho první vydání na CD médiích se objevilo v prosinci 1993 pod jménem FreeBSD 1.0. Bylo částečně založené na 4.3BSD-Line („Net/2“) a 386BSD a částečně také obsahovalo SW od Free Software Foundation. [14]

Vzhledem k právním problémům ohledně vlastnictví kódu 4.3BSD-Line (nárokovala si ho společnost Novell, která ji odkoupila od AT&T) byly tyto části přepsány a vzniklo tak v prosinci 1994 FreeBSD 2.0. [14]

FreeBSD 2.0 bylo úspěšné a bylo následované FreeBSD 2.0.5 vydané v červnu 1995, které bylo robustnější a mělo jednoduší instalaci. [14]

Od té doby FreeBSD udělalo sérii několika vydání a každé nové vydání by dle slov vývojářů mělo poskytovat lepší stabilitu, rychlost a širší možnosti než předchozí vydání. [14]

1.2.2 Současnost

FreeBSD momentálně poskytuje dvě stabilní vydání, označovaná jako 11 a 12. U verze 11 se očekává ukončení podpory minimálně 30. září 2021 a verze 12 minimálně 30. června 2020. Tato data jsou orientační a mohou být prodloužena, nebo může být k verzi poskytována prodloužená podpora[2].

Tato práce je koncipována tak, že používá FreeBSD ve verzi 12 a použitý HW by měl rovněž fungovat právě s touhle verzí.

1.2.3 Software

Jak bylo uvedeno výše, FreeBSD obsahuje část SW vybavení v základní instalaci a část SW je nutné doinstalovat.

IPSec je přímo implementován ve IPv4 a IPv6 stacku [5, s. 579]. Uživatelské aplikace ale nemohou využít protokol IPSec stejně jako využívají protokoly TCP a UDP, místo toho mohou používat zvláštní protokol a jím obsluhovat management klíčů [5, s. 574]. IKE a protokol výměny klíčů (např. *Racoon*) je možné tak implementovat přes uživatelskou aplikaci a tato obsluha není součástí FreeBSD jádra[5, s. 579].

pf je (jeden z firewallů) integrovaný ve FreeBSD původně pocházející z OpenBSD. Ovšem vývoj se rozešel a pf ve FreeBSD používá stejné konfigurační direktivy jako pf ve OpenBSD 5.3. Jelikož se ale OpenBSD momentálně nachází ve verzi 12, konfiguraci pro OpenBSD není bez úprav možné použít ve FreeBSD a naopak. [15]

radvd je router advertisment služba pro IPv6. Naslouchá požadavkům na objevování routerů a posílá na něj odpovědi, jak je definováno v RFC 4861. Umí pracovat i v režimu klienta a nastavovat tak výchozí routy [16]. Je součástí základního systému.

ISC DHCP server nabízí kompletní open source řešení pro implementaci DHCP serverů. Podporuje oba protokoly IPv4 a IPv6. Je vhodný pro aplikace s vysokým provozem a vysokou dostupností [17]. Není k dispozici v základní systému, ale na většině platforem je dostupný ve formě binárního balíčku.

[2]Prodloužená podpora se obvykle nevztahuje na veškerý software, ale obvykle jen na serverový software, u něhož se najdou zákazníci, kteří jsou ochotni prodloužení podpory financovat

1.3 Systémy MS Windows a IPv6

Systémy Microsoft Windows podporují kromě protokolu IPv4 i protokol IPv6. Protokol standardně IPv6 je povolen na všech systémech Windows (Server i Desktop) již téměř 10 let (od verze Windows Vista). Je také nastaven jako upřednostňovaný protokol. [18, s. 138]

„Zkušení uživatelé systému Windows se mohou protokolu IPv6 vyhýbat a upřednostňovat IPv4, kterou znají lépe. Podle našich zkušeností však takový přístup není správný. Budoucnost patří protokolu IPv6. Seznamte se a zvykejte se si na něj.“ [18, s. 138]

Velcí poskytovatelé obsahu rovněž podporují IPv6, Google podporuje IPv4 i IPv6, sít operátora Netflix je s IPv6 kompatibilní a například datové centra Facebooku používá výhradně protokol IPv6. [18, s 138]

Firewall na Windows si udržuje samostatný profil (tj. kompetní kolekci nastavení, včetně pracovních pravidel pro různé programy služby a porty) pro každý ze tří typů:

Doména: používá se, když je počítač připojen do domésy služby active directory, v takovém případě řídí firewall obvykle správce sítě. [18, s. 198]

Privátní: pokud je počítač připojen do domácí sítě (skupiny) nebo sítě pracovní skupiny. [18, s. 198]

Host nebo veřejný: používá se, pokud je připojen do veřejné sítě. Toto nastavení je výchozí. [18, s. 198]

Pokud je počítač připojen k více sítím, může pro každou využít jiný profil[18, s. 198]. Windows si pamatuje zvolený profil pro každou síť [18, s. 139].

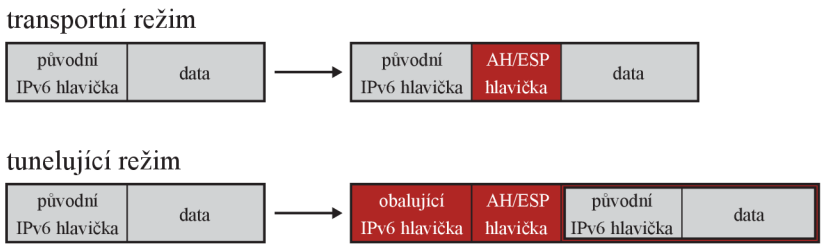
1.4 IPSec

IPv4 neobsahovalo vůbec žádné bezpečnostní mechanismy. Je to překvapující, protože se jednalo o technologii vyvíjenou původně pro armádu. Postupem času se ukázalo, že je bezpečnost nutná, a začaly se hledat cesty jak komunikaci v Internetu zabezpečit. [1, s. 199]

IPSec nachází v síťové vrstvě a je standardně implementován v rámci protokolu IPv6 a standardně zajišťuje šifrování a ověřování jako integrální součást protokolu. Tím odpadá zbytečná režie na samotné šifrování a dešifrování paketů při použití samostatné funkce IPSec. [9, s. 641]

„Pro uživatele či aplikace nabízí IPsec dvě základní služby: autentizaci a šifrování.“ „Současná definice bezpečnostní architektury IP je již třetí generací. První byla soustředěna kolem RFC 1825 vydaného v roce 1995. O tři roky později ji nahradila generace kolem RFC 2401. Zatím poslední specifikaci představuje RFC 4301 vydané koncem roku 2005. Během vývoje nedocházelo k žádným radikálním změnám, kroky jsou spíše evoluční a reagují na zkušenosti získané s implementacemi IPsec a reálnými požadavky na ně kladenými.“ [1, s. 199]

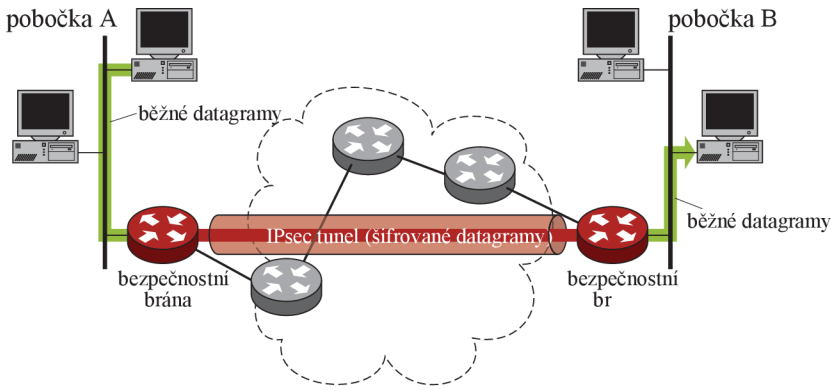
Režimy ochrany lze aplikovat ve dvou režimech. V transportním režimu se vkládají hlavičky přímo jako součást paketu mezi jeho rozšiřující hlavičky. Oproti tomu tunelující režim pracuje tak, že celý stávající paket zabalí jako data do nového paketu, který opatří novými hlavičkami, včetně bezpečnostních. Rozdíl je na obrázku 1.6. [1, s. 200]



Obr. 1.6: Režimy IPSec

Zdroj: [1, s. 200]

„Aby to bylo ještě komplikovanější, nemusí být pakety chráněny po celé své cestě. Vezměme si jako příklad firmu, která bude mít dvě filiálky v různých městech, propojené běžným Internetem. Aby chránila svá interní data, vytvoří mezi těmito pobočkami šifrovaný tunel (obrázek 1.7).“ [1, s. 200]



Obr. 1.7: Použití IPSec

Zdroj: [1, s. 201]

1.4.1 Bezpečnostní asociace

„Důležitou roli v IPsec hraje tak zvaná bezpečnostní asociace (Security Assoasociace ciation, SA). Jedná se o jakési virtuální spojení dvou počítačů, které zajišťuje zabezpečený přenos dat. Součástí bezpečnostní asociace jsou všechny potřebné informace – použitý bezpečnostní protokol (AH nebo ESP, nikoli oba) a jeho režim, šifrovací algoritmus a klíče platné pro toto spojení, čítače, doba životnosti, ochranné prvky proti opakování a podobně.“ [1, s. 201]

„Bezpečnostní asociace jsou jednosměrné. Při komunikaci je nutno navazovat je vždy po dvojicích – jednu pro vysílání, druhou pro příjem. Praktickým dopadem tohoto opatření je, že se v každém směru používají jiné klíče. Navíc jsou asociace jednoúčelové. Chcete-li opatřit pakety jak AH, tak ESP hlavičkou, budete potřebovat samostatnou asociaci pro každou z těchto služeb.“ [1, s. 201]

1.4.2 Databáze bezpečnostní politiky

Řazení má na starosti takzvaná databáze bezpečnostní politiky (Security Policy Database, SPD). Jedná se o sadu pravidel, podle kterých jsou posuzovány všechny přicházející či odesílané pakety. Uplatnění pravidel vede k jednomu z následujících rozhodnutí:

- zahodit paket,
- zpracovat paket – přijmout či odeslat, žádné IPsec služby se na něj nevztahují,
- podrobit paket IPsec – v tom případě databáze vydán odkaz na bezpečnostní asociaci, která na něj má být uplatněna. [1, s. 202]

1.4.3 IPSec spojení

Samotné vytvoření zabezpečeného spojení technologií IPSec je možné následujícími způsoby:

- IPSec model,
- GRE model,
- Remote access client model. [19, s. 109]

IPSec model jedná se o nejjednodušší model připojení, které vytváří bezpečný tunel mezi dvěma stranami. Připojení může být vedeno mezi privátními sítěmi, stejně tak jako přes ATM, nebo přes veřejný internet. Model je vhodný pro svoji jednoduchost, jeho nevýhodou je, že administrátor musí nakonfigurovat bezpečnostní profil pro každý subnet sítě a v rozsáhlých VPN sítích může být konfigurace docela komplexní. [19, s. 110]

Další nevýhodou je, že tato technologie nepodporuje IP multicast, což způsobuje problémy s routovacími protokoly (OSPF, RIP). Toto vede k nutnosti u těch částí sítě spojených protokolem IPSec routovat staticky. [19, s. 110]

GRE model Tento model počítá s vytvořením GRE tunelu, který je zabezpečen pomocí IPSec a jím prochází provoz. Postup je zhruba následující:

- vytvořit GRE tunel,
- zabezpečit GRE tunel pomocí IPSec,
- zajistit IP konektivitu mezi GRE tunelem a IPSec konci (routouvání mimo VPN),

- zajistit routovací cesty pro koncové systémy skrz GRE tunel (routování přes VPN). [19, s. 111]

U tohoto modelu je výhoda, že jím prochází všechen provoz a minimalizuje se tak množství pravidel IPSec, který by bylo potřeba nastavovat pro každý segment sítě. Nevýhodou je nutnost používat 2 routovací plány:

- routování zabezpečených paketů mezi VPN bránami,
- routování provozu skrz tunel mezi koncovými subsítěmi. [19, s. 111]

Remote access client mode: předchozí modely fungují dobře, pokud jsou adresy koncových bodů známé, ale už nefungují pro pracovníky, kteří se musí vzdáleně připojovat do firemní sítě. [19, s. 112]

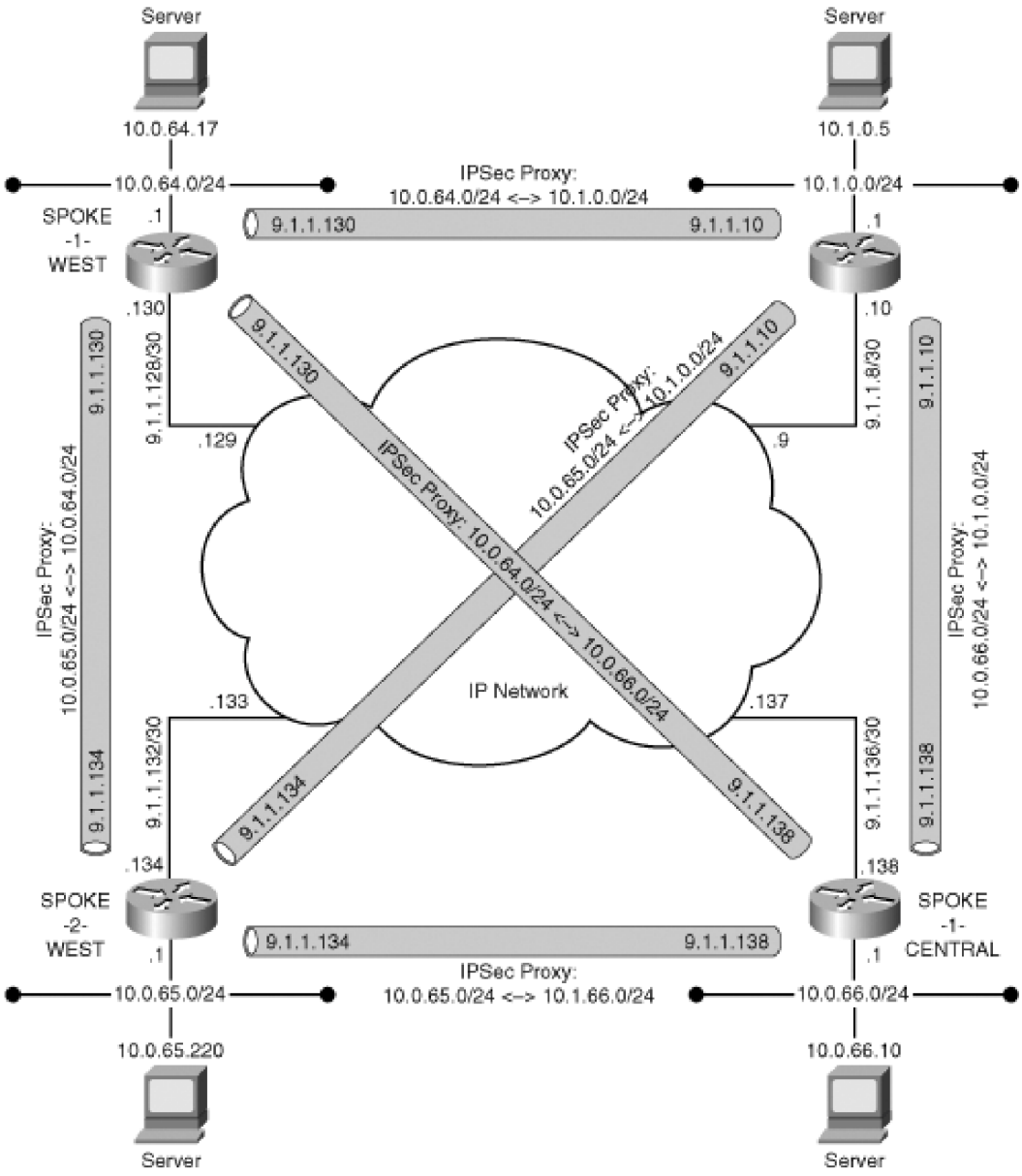
Pro tento požadavek existuje protokol IKE a je možné ho použít pro automatickou konfiguraci takto připojovaných zařízení. Typické atributy pro konfiguraci jsou následující: [19, s. 112]

- vnitřní adresa,
- vnitřní DNS server,
- vnitřní WINS server,
- doménové jméno vnitřní sítě.

Tento model má i své nevýhody: Za prvé – spojení může být navázáno jen ze strany klienta k serveru. Za druhé – takto vytvoření spojení nepodporuje multicast. [19, s. 112]

1.4.4 Architektura Full-mesh

Tato architektura vytváří zabezpečené spojení s každým bodem v síti. Požadavkem je obvykle to, aby provoz šel vždy nejkratší cestou, a odolnost proti výpadku jednotlivých cest [19, s. 156]. Tento model je na obrázku 1.8.



Obr. 1.8: IPSec tunely pro architekturu Full-Mesh

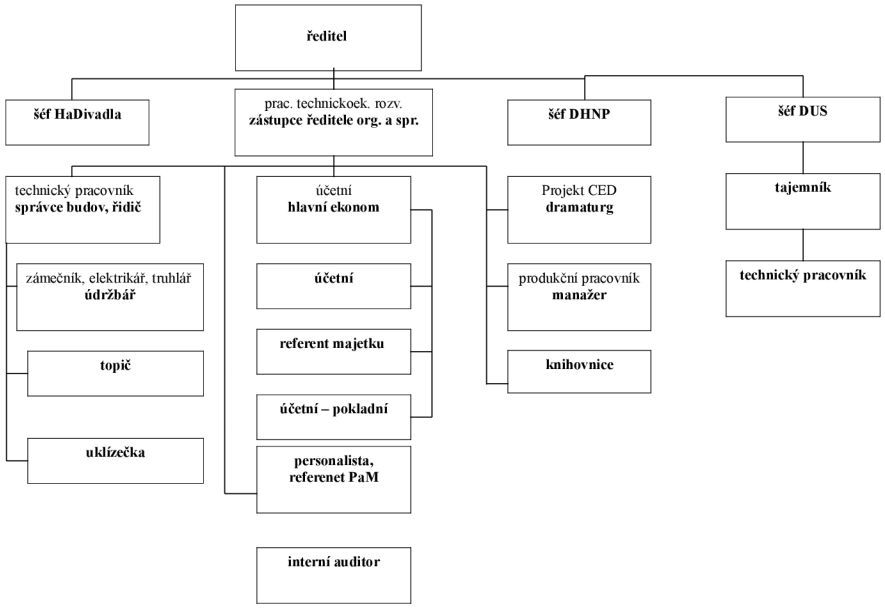
Zdroj: [19, s. 157]

2 ANALÝZA SOUČASNÉHO STAVU

2.1 O organizaci

Centrum experimentálního divadla je příspěvková organizace, sdružující soubory Divadlo Husa na provázku, HaDivadlo a Projekt CED:

„Centrum experimentálního divadla je nejen svazkem několika divadel rozvíjejících se vedle sebe a pod svými vlastními jmény své vyhraněné tvůrčí programy, ale chce poskytovat přístřeší i jiným pokusům. Ve svých divadelních prostorách chce hostit příbuzná divadla česká i zahraniční. Chce být prostorem pro mezinárodní festivaly a duchovním podnikem, v němž je divadelní tvorba vřícím epicentrem, který však svou aktivitu rozvíjí i v širším kulturním prostoru.“ [20]



Obr. 2.1: Organizační struktura CED, p.o. (k 31. 7. 2007)

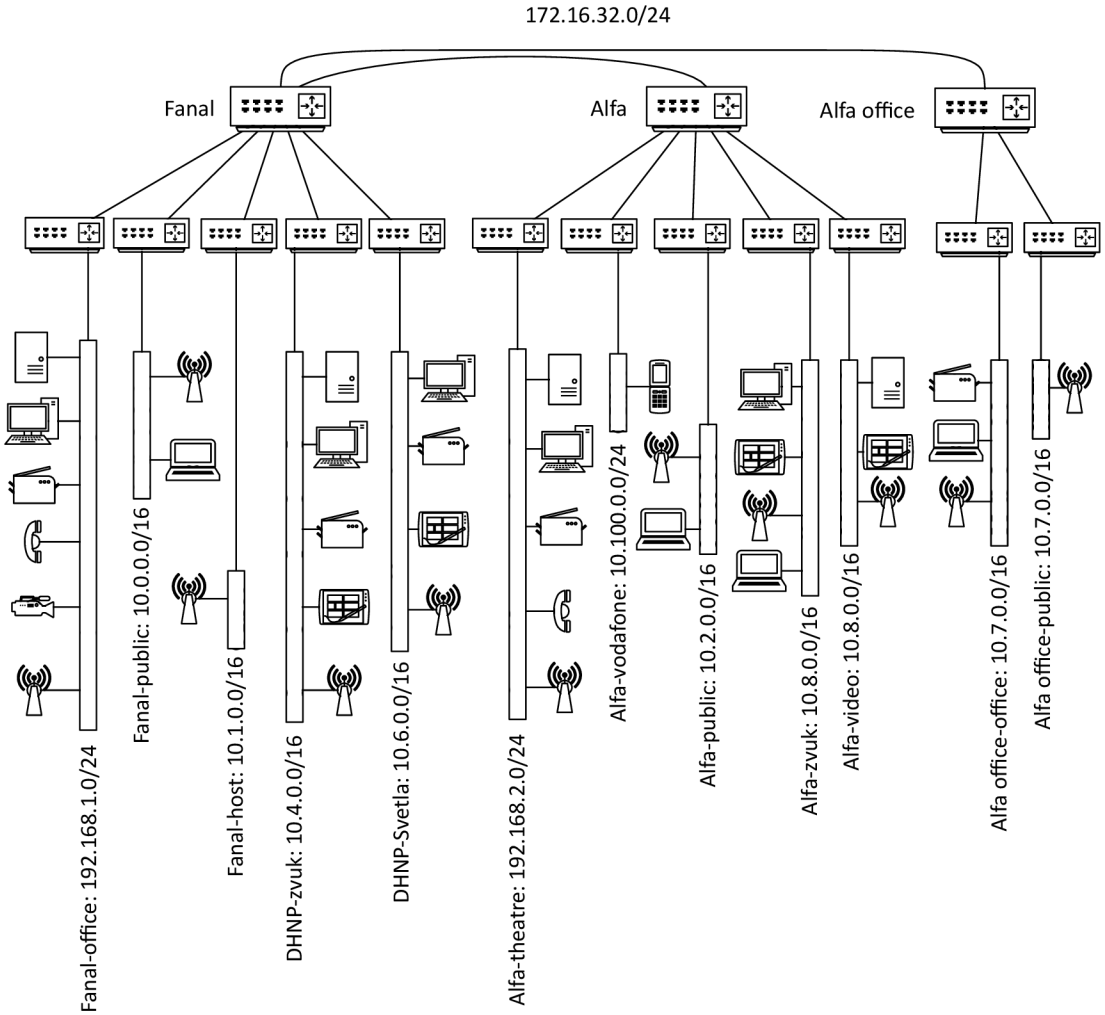
Zdroj: [21]

Organizace byla založena 1. 1. 1992 [22, s. 1], avšak existence provozovaných uměleckých souborů se datuje do roku 1968 v případě Divadla Husa na provázku [23] a od roku 1974 v případě HaDivadla [24].

K 31. 12. 2018 měla organizace 98 zaměstnanců [25, s. 86] a rozpočet se pohybuje kolem 60 mil. Kč [25, s. 92]. Organizační struktura organizace je zobrazena obrázku 2.1.

2.2 Topologie počítačové sítě organizace z pohledu síťové vrstvy modelu ISO/OSI

Počítačovou síť organizace tvoří 3 geograficky oddělené sítě a dohromady 12 podsítí (na obrázku 2.2). První ze sítí, interně pojmenovaná jako *Fanal*, se geograficky nachází v prostorách Domu pánů z Fanalu a Nové scény; druhá, která interně označovaná jako *Alfa*, se nachází v prostorách HaDivadla v Alfa pasáži. Třetí, pojmenovaná *Alfa Office* je kancelářský prostor v Alfa pasáži, který ale přímo nesousedí s prostory HaDivadla a není možné je přímo propojit.



Obr. 2.2: Topologie z pohledu síťové vrstvy modelu ISO/OSI

Zdroj: *vlastní tvorba*

2.2.1 Síť: „Fanal“

Tato síť se nachází v budovách Domu pánů z Fanalu a Nové scény, jejichž 5 podsítí je popsáno níže.

2.2.2 Podsíť: „Fanal-office“

V této síti je umístěna většina pevných počítačů a slouží zejména administrativě, jsou do ní připojené síťové tiskárny, nachází se zde souborový/aplikační server a zálohovací server. Rovněž je do ní připojené záznamové zařízení kamerového systému a telefonní ústředna.

Bezdrátově je možno se k síti připojit přes SSID *CED-RAD*. Lze k ní připojovat jen schválená zařízení. Zařízení v této síti mohou komunikovat se zařízeními v síti „Alfa-office“ a „Alfa-theatre“.

2.2.3 Podsíť: „Fanal-public“

Veřejná síť pro zaměstnance je firewallem oddělená od sítě „Fanal-office“ a nelze se z ní dostat na servery a pevné počítače. Slouží prakticky jen k připojení k internetu. Zaměstnanci si do ní mohou připojovat svá zařízení, je switchována s `VLAN ID 1` a bezdrátově je možno se k ní připojit přes SSID *CED*.

2.2.4 Podsíť: „Fanal-host“

Veřejná síť, bez hesla (a bez šifrování) – je určena veřejnosti, provoz má menší prioritu, než ostatní sítě. Je switchována s `VLAN ID 2`. SSID bezdrátově sítě je *CED-Faster.cz*.

2.2.5 Podsíť: „DHNP-zvuk“

Síť určená zvukařům Divadla Husa na provázku, je do ní připojen jejich vlastní datový server, zvukový pult a obslužné počítače na scéně a ve zvukové dílně. Také je sem připojena síťová tiskárna ve zvukové dílně. Bezdrátově se lze připojit k SSID *DHNP-Zvuk*, síť switchována s `VLAN ID 3`. Tuto síť bude v budoucnosti potřeba rozdělit na 3 podsítě: „public“, „control“ a „device“. Moje práce v části s návrhem řešení s tímto rozdělení počítá.

2.2.6 Podsíť: „DHNP-svetla“

Určena osvětlovačům Divadla Husa na provázku, je do ní připojen jejich světelný pult a počítač ve světelné dílně. Bezdrátově se lze připojit k SSID *DHNP-Svetla*, síť switchována s `VLAN ID 4`.

2.2.7 Síť: „Alfa“

Síť umístěna v prostorách bývalého kina v Alfa pasáži v nynějšímu sídle HaDivadla - 5 podsítí jsou popsané níže.

2.2.8 Podsíť: „Alfa-theatre“

Podsíť pro administrativu HaDivadla, jsou do ní připojeny pracovní stanice, VOIP telefony a datový server. SSID bezdrátově je *CED RAD*, jelikož není technicky možné zároveň chytat bezdrátové varianty sítě „Fanal-office“ a „Alfa-theatre“, může se bezdrátová síť jmenovat stejně. Rovněž to umožňuje řídícím pracovníkům pohybovat se v různých prostorách a připojovat se stále k jedné síti. Do sítě je možné připojovat jen schválená zařízení.

2.2.9 Podsíť: „Alfa-public“

Veřejná síť pro zaměstnance, slouží pouze k přístupu na internet, je switchována s `VLAN ID 3`. Bezdrátové SSID je *HADI*.

2.2.10 Podsíť: „Alfa-Vodafone“

Tato síť je určena pro 3G vysílače signálu, které pokrývají část budovy, kde organizací používaný operátor nemá signál. Vzhledem k tomu, že organizace nemá nad těmito zařízeními žádnou kontrolu, byla tato zařízení umístěna ve vlastní separované síti. Není nutné mít do budoucna pro tyto účely speciální síť a proto v návrhu místo s touto operuji s univerzálnější „Alfa-Host“, která má sloužit pro tato a podobná zařízení.

2.2.11 Podsíť: „Alfa-video“

Síť určená pro video produkci HaDivadla, je do ní připojen NAS a potřebná zařízení. Síť je switchována s `VLAN ID 4`. Bezdrátové SSID je *HADI-Zvuk*.

2.2.12 Podsíť: „Alfa-zvuk“

Síť určená pro zvukaře HaDivadla, jsou do ní připojeny jejich pracovní stanice, zvukový pult a jejich notebooky. Síť je switchována s `VLAN ID 5`. Bezdrátové SSID je *HADI-Video*.

2.2.13 Síť: „Alfa office“

Síť je umístěna v Alfa pasáži mimo prostory HaDivadla (nachází se nad Kavárnou Švanda), slouží především administrativě HaDivadla a má dvě podsítě.

2.2.14 Podsíť: „Alfa office-office“

Do této podsítě je připojená síťová multifunkční tiskárna a SSID bezdrátově je *CED RAD*, do níž lze připojovat jen schválená zařízení.

2.2.15 Podsíť: „Alfa office-public“

Veřejná síť pro zaměstnance, slouží pouze k přístupu na internet, je switchována s `VLAN ID 1`. Bezdrátové SSID je *HADI*.

2.2.16 Propojení sítí

Sítě „Fanal“, „Alfa“ a „Alfa office“ jsou propojeny pomocí VPN, primární trasa využívá jako VPN server router v síti „Fanal“, sekundární využívá jako VPN server router v síti „Alfa“. Volba směrování je řešena pomocí dynamického routování s protokolem OSPF.

2.2.17 Adresní plán

Adresní plán je vidět v tabulce 2.1 pro síť Fanal, pro síť Alfa je v tabulce 2.2 a v tabulce 2.3 pro síť Alfa office.

Rozsah	Vyhrazené adresy	Použití
192.168.1.0/24	192.168.1.5-15	servery
	192.168.1.15-29	VOIP
	192.168.1.30-199	pracovní stanice a notebooky
	192.168.1.200-220	ostatní síťová zařízení
	192.168.1.230-249	bezdrátová AP
	192.168.1.250	kamerový systém
	192.168.1.252	switch - nová scéna
	192.168.1.253	kontrolér bezdrátových AP
	192.168.1.254	brána, DNS
10.0.0.0/16	10.0.0.1	brána, DNS
	10.0.100.1-199.254	klientská zařízení
10.1.0.0/16	10.1.0.1	brána, DNS
	10.1.100.1-199.254	klientská zařízení
10.4.0.0/16	10.4.0.1	brána, DNS
	10.4.0.10-0.20	servery
	10.4.100.1-199.254	klientská zařízení
10.5.0.0/16	10.5.0.1	brána, DNS
	10.5.100.1-199.254	klientská zařízení
176.16.0.0/24	176.16.0.2-254	VPN pro zaměstnance
176.16.32.0/24	176.16.32.2-254	VPN propojovací

Tab. 2.1: Adresní plán IPv4, Fanal

Zdroj: *vlastní tvorba*

Rozsah	Vyhrazené adresy	Použití
192.168.2.0/24	192.168.2.1	brána, DNS
	192.168.2.20-199	pracovní stanice a notebooky
	192.168.2.200-220	VOIP telefony
	192.168.2.230-239	bezdrátová AP
	192.168.2.240-250	ostatní síťová zařízení
10.2.0.0/16	10.2.0.1	brána, DNS
	10.2.100.1-199.254	klientská zařízení
10.100.0.0/24	10.100.0.1	brána, DNS
	10.100.0.10-200	3G vysílače
10.6.0.0/16	10.6.0.1	brána, DNS
	10.6.100.1-199.254	klientská zařízení
176.16.64.0/24	176.16.64.2-254	VPN záložní

Tab. 2.2: Adresní plán IPv4, Alfa

Zdroj: *vlastní tvorba*

Rozsah	Vyhrazené adresy	Použití
192.168.3.0/24	192.168.3.1	brána, DNS
	192.168.3.20-199	notebooky
	192.168.3.230-239	bezdrátová AP
	192.168.3.240-250	ostatní síťová zařízení
10.7.0.0/16	10.7.0.1	brána, DNS
	10.7.100.1-199.254	klientská zařízení

Tab. 2.3: Adresní plán IPv4, Alfa office

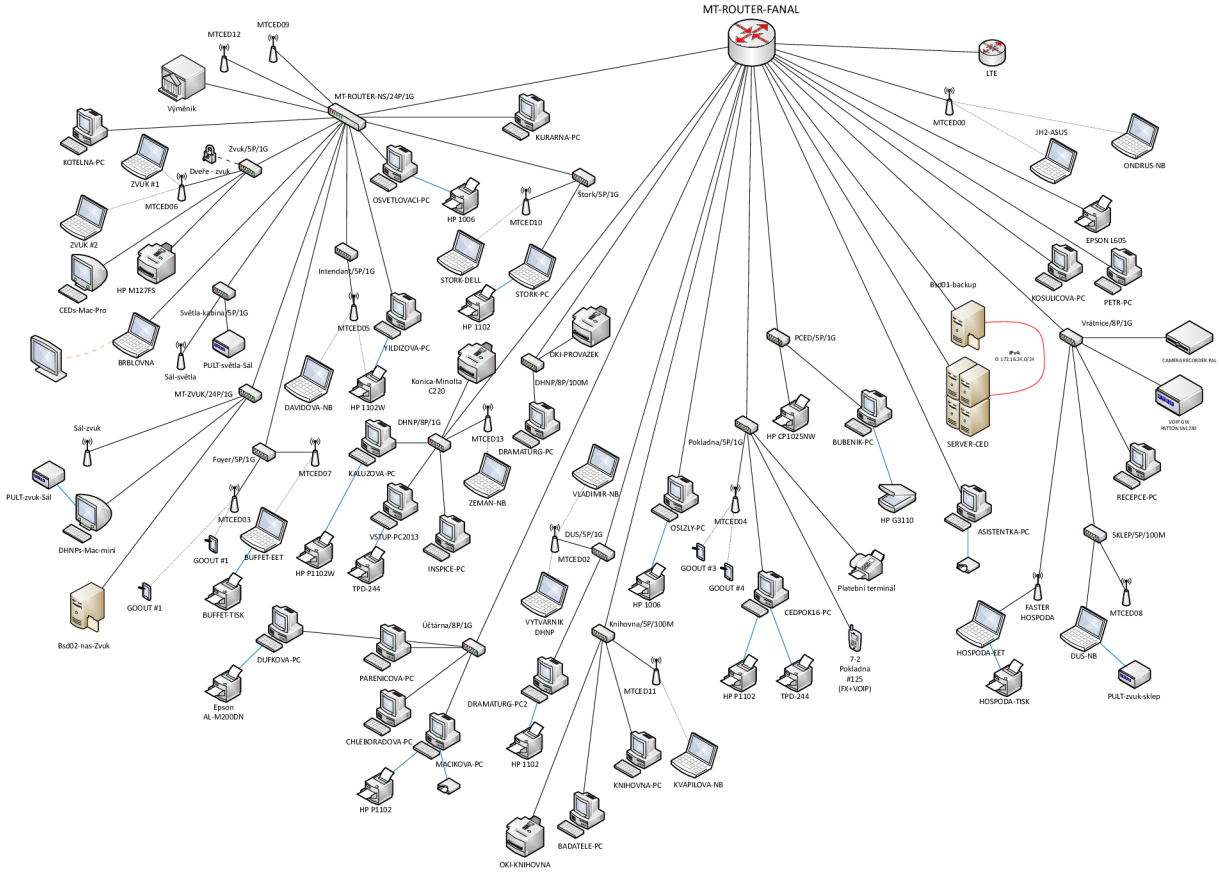
Zdroj: *vlastní tvorba*

2.3 Topologie počítačové sítě organizace z pohledu linkové a fyzické vrstvy modelu ISO/OSI

Fyzické zapojení sítě **„Fanal“** je na obrázku 2.3. Jako router je používán Mikrotik CRS326, s tím že port 1 slouží jako WAN a je routován a ostatní porty jsou switchovány. Mapa portů je na obrázku 2.5.

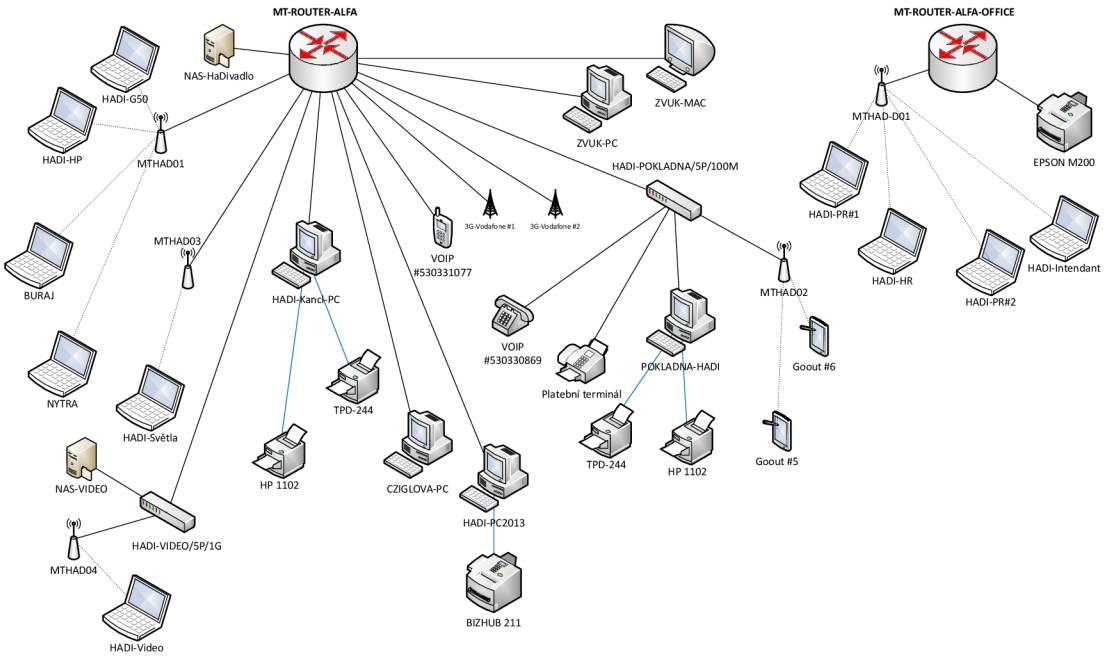
Pro druhý z centrálních switchů je rovněž používáno zařízení Mikrotik CRS326 s tím, že jsou switchovány všechny porty a nejsou využívány žádné routovací funkce. Třetí centrální switch (opět Mikrotik CRS326) slouží pro různá zařízení, které obsluhuje technika Divadla Husa na provázky a jeho porty jsou zapojené do VLAN sítí (`VLAN ID` 3 a 4).

Fyzické zapojení sítě **„Alfa“** a **„Alfa office“** je na obrázku 2.4. Jako router rovněž slouží Mikrotik CRS326.



Obr. 2.3: Síť Fanal z pohledu fyzické vrstvy

Zdroj: *vlastní tvorba*



Obr. 2.4: Síť Alfa a Alfa officez pohledu fyzické vrstvy

Zdroj: *vlastní tvorba*

2.4 Použité technologie a jejich konfigurace

Následující části práce popisuji použité síťové technologie v organizaci.

2.4.1 VLAN

Organizace používá VLAN podle standardu *801.1q*, s pomocí VLAN jsou odděleny jednotlivé podsítě, je ale míchán provoz netagované VLAN a VLAN tagovaných.

2.4.2 Bezdrátová síť

Bezdrátová síť je postavena na technologii *802.11n* a *802.11a* - jednotlivá AP[1] jsou řízena protokolem `CAPsMAN` od firmy Mikrotik. Řídící zařízení[2] je zařízení `MTCED00`. Protokol umožňuje řídícímu zařízení řídit jak komunikaci, tak data. V současné době jsou komunikace i data posílány na řídící zařízení a to je posílá přes příslušnou VLAN.

[1] v terminologii Mikrotiku označována CAP

[2] v terminologii Mikrotiku označována CAPsMAN

zařízení/místnost	port	port	zařízení/místnost
WAN	1	2	MTCED00
MT-ROUTER-NS	3	4	F16
SERVER-CED	5	6	BSD01-BACKUP
F16	7	8	F16
F515	9	10	F15
–	11	12	F19
F18	13	14	F18
F19	15	16	F19
F20	17	18	F20
F20	19	20	F7
F200	21	22	F16
F19	23	24	F102
–	sfp1	sfp2	–

Legenda

10M	100M	1G
10G	nezapojeno	zakázáno

Obr. 2.5: Zapojení portů MT-Router-Fanal

Zdroj: *vlastní tvorba*

2.4.3 Routování

Routování je dynamické s vyžitím protokolu OSPF.

2.4.4 VPN

Pro VPN je požíván protokol `OpenVPN`, jako transportní protokol je používán TCP, protože zařízení Mikrotik neumí používat OpenVPN přes (vhodnější) UDP.

2.4.5 Autokonfigurace

Pro automatickou konfiguraci se používá protokol DHCP. DHCP server běží přímo na routerech. Většina zařízení nemá nakonfigurovanou pevnou IP adresu, ale získá předefinovanou podle své MAC.

2.4.6 DNS

DNS server běží na zařízení SERVER-CED, server je rekurzivní. Na routerech běží pouze DNS cache. V případě, že zařízení SERVER-CED není k dispozici (selhání VPN, server instaluje aktualizace apod.), jsou pro překlad použity servery firmy Google. Přepínací skript je následující:

```
1 /tool netwatch
2 add comment="DNS swicth on server down" down-script=\
3     "/ip dns set servers=\"8.8.8.8\"\r\
4     \n" host=192.168.1.12 interval=30s up-script=\
5     "/ip dns set servers=\"192.168.1.12\"\r\
6     \n/ip dns cache flush\r\
7     \n"
```

2.5 Zabezpečení

Organizace se i přes velké množství externistů a nepravidelných uživatelů snaží svoji síť co nejvíce zabezpečit. Vzhledem k délce vývoje místní počítačové sítě a omezeným finančním prostředkům se potýká se zastaralými zařízeními, zařízeními s omezenou schopností monitoringu a ne vždy zcela vhodnou topologií.

2.5.1 Fyzická bezpečnost

Fyzické zabezpečení funguje na následujících principech:

1. Žádné síťové prvky v prostorech přístupných veřejnosti.
2. Vypnuté nevyužité porty (pokud to zařízení umožňuje).
3. Pokud pracovník nepotřebuje přístup k vnitřní síti, není do ní připojen.
4. Všechna nová síťová zařízení musejí být konfigurovatelná a monitorovatelná.

2.5.2 Firewall

Firewall běží na routerech (*MT-ROUTER-FANAL*, *MT-ROUTER-ALFA* a *MT-ROUTER-ALFA-OFFICE*). Následuje zkrácený výpis konfigurace:

```
1  /ip firewall filter
2  add action=accept chain=input comment="Opened connections" connection-
       state=established,related
3  add action=accept chain=input comment="Mikrotik winbox" dst-port=8291
       protocol=tcp
4  add action=accept chain=input comment=DNS dst-port=53 in-interface-list
       =lan protocol=udp
5  add action=accept chain=input comment=DNS dst-port=53 in-interface-list
       =lan protocol=tcp
6  add action=accept chain=input comment=DHCP dst-port=67 in-interface-
       list=lan protocol=udp
7  add action=accept chain=input comment=VPN dst-port=4443 protocol=tcp
8  add action=accept chain=input comment=OpenVPN dst-port=1194 protocol=
       tcp
9  add action=accept chain=input comment=PING protocol=icmp
10 add action=accept chain=input comment="Internal input INTERFACE" in-
       interface-list=private
11 add action=accept chain=input comment="Internal input ADDRESS" src-
       address-list=private
12 add action=drop chain=input comment="ALL others"
13 add action=accept chain=forward comment=ESTABLISHED connection-state=
       established,related
14 add action=accept chain=forward comment="servers: PING" dst-address-
       list=servers protocol=icmp
15 add action=accept chain=forward comment="servers: TCP" dst-address-list
       =servers dst-port=3389,80,443,20,21,22,8000-8100 protocol=tcp
16 add action=accept chain=forward comment="servers: UDP" dst-address-list
       =servers dst-port=3389 protocol=udp
17 add action=accept chain=forward comment=RDP-TCP dst-address-list=rdp
       dst-port=3389 protocol=tcp
18 add action=accept chain=forward comment=RDP-UDP dst-address-list=rdp
       dst-port=3389 protocol=udp
```

```
19 add action=accept chain=forward comment="ALL -> WAN" out-interface-list
   =wan
20 add action=accept chain=forward comment="private -> private (INTERFACE)
   " in-interface-list=private out-interface-list=private
21 add action=accept chain=forward comment="private -> private (ADDRESS)"
   dst-address-list=private src-address-list=private
22 add action=drop chain=forward
```

Z tohoto je vidět, že firewall operuje se třemi druhy sítí WAN, LAN-vnější a LAN-vnitřní. Pro segmenty LAN-vnitřní jsou routování mezi segmenty LAN-vnitřní povoleno, jinak je povoleno pouze routování do internetu.

Kromě toho ještě firewall operuje se seznamy: *servers*, kde jsou servery, jejichž vybrané služby mohou využívat i zařízení v LAN-vnější, a seznam *RDP*, kde jsou umístěné zařízení s přístupem přes protokol RDP (vzdálená plocha).

2.5.3 VPN

Mezi stále připojenými zařízeními je použita VPN na protokolu OpenVPN a ověřují se certifikátem, zaměstnanci se mohou připojit před protokol SSTP a ověřují se jménem a heslem. Veškerý provoz přes VPN sítě je šifrován.

2.5.4 SSL, EAP a použité šifrování

Pro VPN používá organizace certifikáty šifrovaní klíče RSA s délkou klíče 2048 b (starší zařízení) a 4096 b (novější zařízení), pro projení je pak použita šifra AES od délce 256 b. Protokoly vyšších vrstev používají protokol TLS 1.2, RSA s délce 4096 b s ECDH a AES o délce 256 b.

2.6 Servery, stanice zařízení a tiskárny

V tabulce 2.4 se nachází seznam operačních systému, které běží na serverech a stanicích. Rovněž se v ní nachází seznam síťových tiskáren, které má organizace zapojené v síti.

Výrobce	Produkt	Verze
Servery		
Microsoft	Windows Server	2012
FreeBSD Project	FreeBSD	11, 12
Debian Project	Debian GNU/Linux	9
Synology	Synology DiskStation Manager	6
Pracovní stanice a notebooky		
Microsoft	Windows	7, 10
Apple	macOS	10.6, 10.10-10.14
Mobilní zařízení		
Apple	iOS	různé
Google	Android	různé
Síťové tiskárny		
Konica Minolta	bizhub	C220
OKI	MB400	MB461, MB441
HP	LaserJet	1102W, CP 1025nw
Epson	–	L605, M200

Tab. 2.4: Operační systémy v organizaci

Zdroj: *vlastní tvorba*

2.7 Známé chyby návrhu

Počítačová síť Centra experimentálního divadla prošla poměrně chaotickým vývojem, uvádím zde proto několik chyb, které se v návrhu objevily a v případě zavedení IPv6 způsobují problémy.

2.7.1 Agregace rozsahů privátních sítí

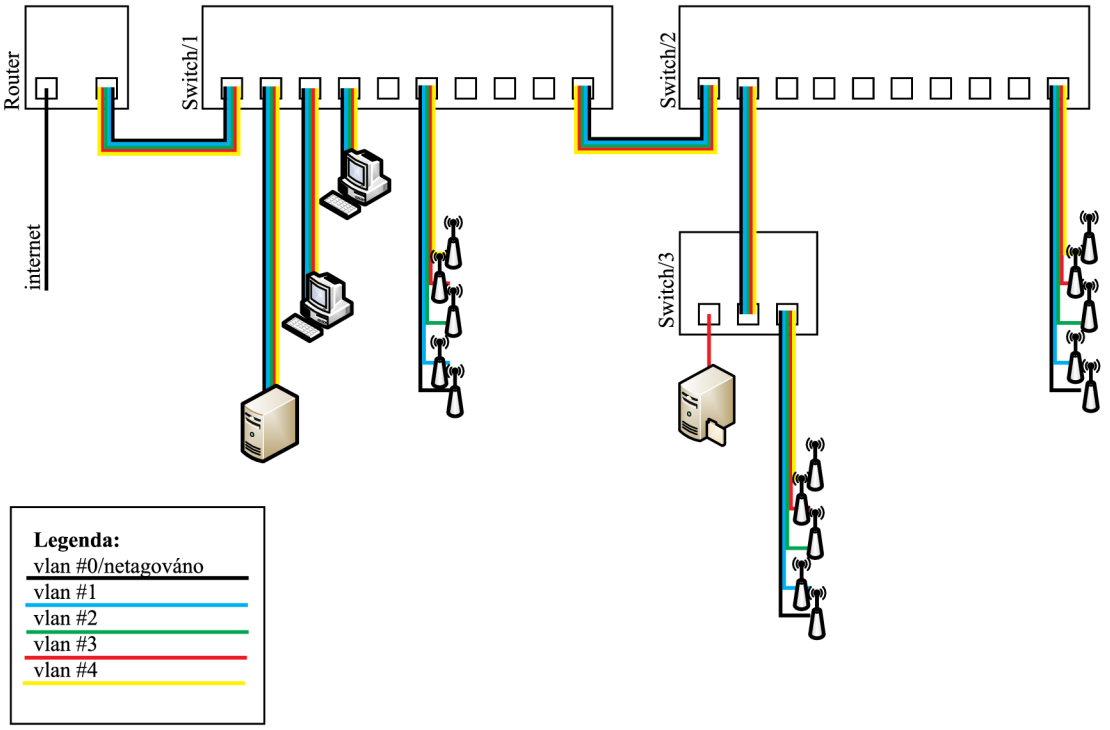
Pokud porovnáme tabulky 2.1 a 2.2, vidíme, že jednotlivé podsítě z rozsahu `10.0.0.0/8` jsou alokovány za sebou a jsou přímo routovatelné – kupříkladu síť `10.1.0.0/16` se nachází v budově Domu pánů z Fanalu a síť `10.2.0.0/16` v Alfa pasáži a `10.3.0.0/16` je

opět v budově Domu pánů z Fanalu. Má to za následek delší routovací tabulky.

Synchronizace routovacích tabulek je řešitelná pomocí dynamického routování a vliv delších routovací tabulek na výkon nebyl zatím pozorován. Úzkým hrdlem nadále zůstává připojení k internetu.

2.7.2 Míchání tagovaného a netagovaného provozu

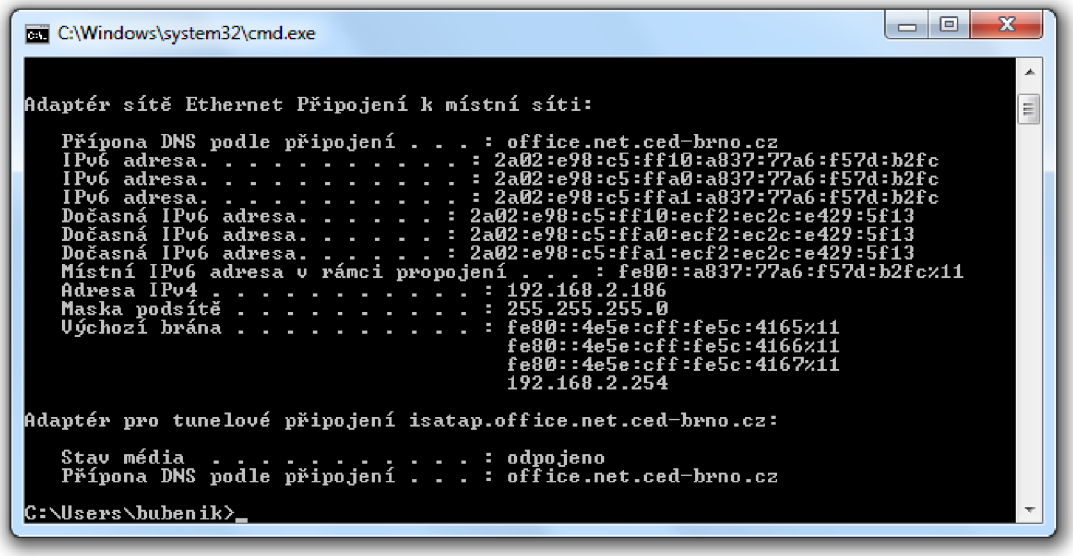
Jak ilustruje obrázek 2.6, většina zařízení připojená do vnitřní sítě dostává i tagované VLAN rámce, což by nevadilo v případě, že zařízení mají podporu VLAN, nebo tagované rámce zahazují. Bohužel se stále nepodařilo vyřadit se sítě všechny nekonfigurovatelné switche, které se v případě VLAN rámců chovají jako huby a problémy se vyskytují i na koncových zařízeních, která ve výchozí konfiguraci akceptují VLAN tagovaný rámec, přestože nemají nastavenou VLAN.



Obr. 2.6: Míchání tagovaného a netagovaného VLAN provozu

Zdroj: *vlastní tvorba*

Bohužel v IPv6 tato zařízení dostanou Router Advertisement a přiřadí si adresu, i když není možné přes ni komunikovat. Na obrázku 2.7 je výpis příkazu `ipconfig` na systému Microsoft Windows 7 ve zmíněné konfiguraci.



Obr. 2.7: Chybné přiřazení adres z různých VLAN

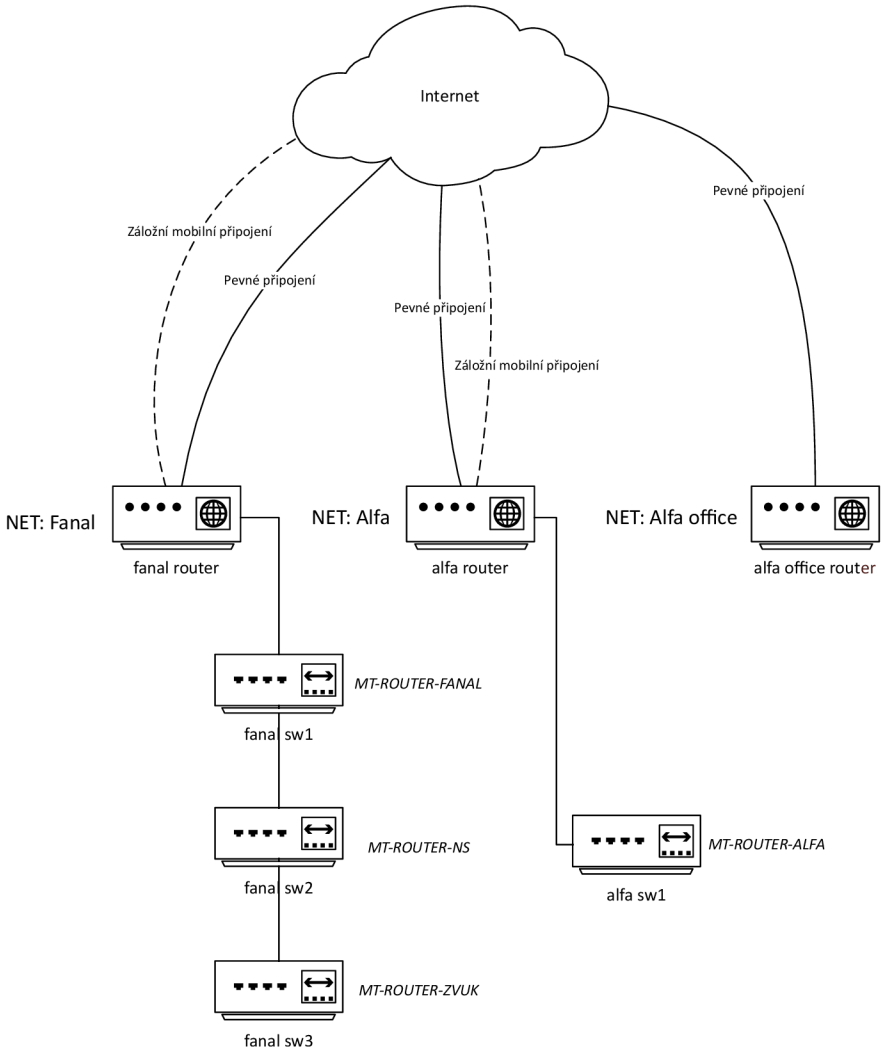
Zdroj: *vlastní tvorba*

2.7.3 DNS infrastruktura

Zařízení Mikrotik podporují pouze DNS cache a nejsou schopna fungovat jako sekundární DNS resolver, viz 2.4.6. Následkem je, že zařízení nejsou schopna překládat adresy, pokud nemají nějaký DNS server pro předávání dotazů. Pokud je použit veřejný, tak není možné přeložit názvy místních zařízení.

3 NÁVRH ŘEŠENÍ

V této kapitole popisuji svůj návrh řešení. Protože jsem si vědom, že ne všechna zařízení a všechny služby plně podporují IPv6, použiji tzv. dual stack - provozování protokolu IPv6 i IPv4 současně.



Obr. 3.1: Propojení do internetu

Zdroj: *vlastní tvorba*

3.1 Předpoklady

Návrh řešení předpokládá, že máme dva hlavní objekty (Fanal a Alfa), jejichž hlavní router je připojen do internetu pomocí pevného připojení a záložního mobilního připojení.

Kromě toho má organizace ještě kancelářské prostory s pevným připojením (Alfa office) bez možnosti ji přímo propojit s některým s hlavních objektů. Na obrázku 3.1 je zobrazeno toto zapojení včetně hlavních ethernetových switchů.

Rovněž předpokládám, že internetoví provideři jsou schopni zajistit přidělení IPv6 prefixu o minimální délce **/48** pro hlavní objekty a pro **/56** pobočku[1].

3.2 Adresní plán

V tabulce 3.1 jsou veřejné adresy, které jsou k dispozici v jednotlivých budovách. Plán adres pro jednotlivé podsítě se nachází v tabulce 3.2.

Objekt	Verze protokolu	Typ připojení	Adresa
Fanal	6	Primární připojení	2001:db8:ced1::/48
Fanal	6	Záložní připojení	2001:db8:ced1::/48
Fanal	4	Primární připojení	192.0.0.1
Fanal	4	Záložní připojení	192.0.0.2
Alfa	6	Primární připojení	2001:db8:afa1::/48
Alfa	6	Záložní připojení	2001:db8:afa2::/48
Alfa	4	Primární připojení	192.0.0.101
Alfa	4	Záložní připojení	192.0.0.102
Alfa office	6	Primární připojení	2001:db8:ffce:aa00::/56
Alfa office	4	Primární připojení	neveřejná adresa

Tab. 3.1: Veřejné adresy pro jednotlivé prostory

Zdroj: *vlastní tvorba*

[1] Teoreticky nejnižší rozsah, pro který je možné použít můj návrh řešení, je **/60**, důrazně ovšem doporučuji si nechat přidělit rozsah co největší, aby byl prostor k rozšiřování sítě

Název	VLAN id	ipv4	ipv6	DNS přípona
Objekt: Fanal (Dům pánů z Fanalu a Nová scéna)				
office	1	192.168.1.0/24	PREFIX:1000::/64	office.net.ced-brno.cz
public	2	10.0.0.0/16	PREFIX:a000::/64	public.net.ced-brno.cz
host	3	10.1.0.0/16	PREFIX:a100::/64	host.net.ced-brno.cz
public_zvuk	4	10.4.1.0/24	PREFIX:b000::/64	public.zvuk.net.provazek.cz
control_zvuk	5	10.4.2.0/24	PREFIX:b100::/64	control.zvuk.net.provazek.cz
device_zvuk	6	10.4.3.0/24	PREFIX:b200::/64	device.zvuk.net.provazek.cz
světla	7	10.6.0.0/16	PREFIX:c000::/64	svetla.net.provazek.cz
Objekt: Alfa (Alfa pasáž - divadlo)				
theatre	1	192.168.2.0/24	PREFIX:1000::/64	theatre.net.hadivadlo.cz
public	2	10.2.0.0/16	PREFIX:a000::/64	public.net.hadivadlo.cz
host	3	10.3.0.0/16	PREFIX:a100::/64	host.net.hadivadlo.cz
video	4	10.5.0.0/16	PREFIX:d000::/64	video.net.hadivadlo.cz
zvuk	5	10.8.0.0/16	PREFIX:b000::/64	zvuk.net.hadivadlo.cz
Objekt: Alfa office (Alfa pasáž - kancelář)				
office	0[2]	192.168.3.0/24	PREFIX00::/64	office.net.hadivadlo.cz
public2	1	10.7.0.0/16	PREFIXa0::/64	public2.net.hadivadlo.cz

Tab. 3.2: Plán adres pro jednotlivé podsítě

Zdroj: *vlastní tvorba*

[2] netagováno

3.3 Výběr hardwaru

Vzhledem k tomu, že podpora IPv6 v routerech od firmy Mikrotik není dostatečná, musel jsem nalézt jiné řešení a vybrat jinou platformu. Kromě toho, že implementace není úplně dotažená a nepodporuje mnohé vlastnosti protokolu, objevují se zde i bezpečnostní zranitelnosti:

„Komunita doufá, že MikroTik stihne do oznámení vydat opravenou verzi RouterOS. Je pravděpodobné, že chyba začne být okamžitě zneužívána k DoS útokům na sítě využívající MikroTiky jako hraniční routery. Je také načase, aby IPv6 přestal být v RouterOS protokolem druhé kategorie a dostalo se mu stejné péče jako v případě IPv4.“ [26]

Stávající zařízení (Mikrotik CRS 326) je však možno dál používat jako konfigurovatelné 24 portové switche. Tyto switche jsou na obrázku 3.1 napsané kurzívou.

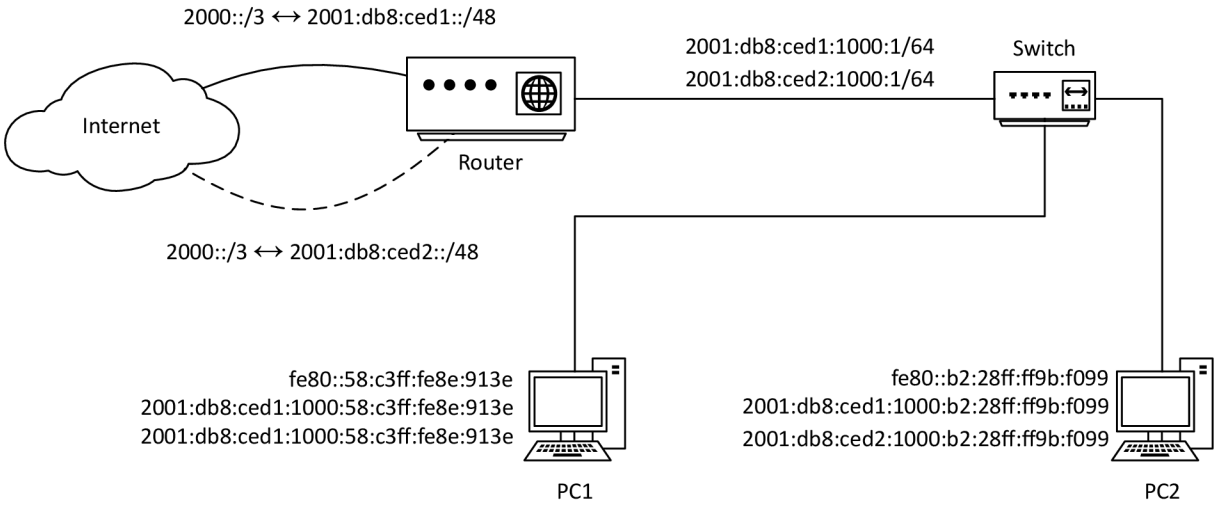
Vzhledem k výborné implementaci v operačním systému FreeBSD jsem se rozhodl využít právě tento systém, jenž je provozovatelný na libovolném PC hardwaru. K extra vybavení potřebuji minimálně 2 síťové karty a dostatečný diskový prostor na uchování informací o provozu v síti, popř. uchovávání části provozu.

Jako HW pro router jsem zvolil HPE ProLiant MicroServer Gen10 (s procesorem AMD Opteron X3418 a 8 GB ram) s 4 TB úložištěm (disk 2x 4 TB) pro data a 128 GB pro systém. Velikost informací o datovém toku není nijak velká (cca 200 MB/síť/den), organizace musí ale počítat s tím, že bude nutné tyto informace zálohovat (nejlépe mezi routery navzájem) a možná bude nutné do budoucna uchovávat i část skutečného provozu. Datový prostor 4 TB vychází na cca 1 měsíc uchování veškerého provozu, který proteče do a z internetu. Organizace bude tento router (jak je vidět na obrázku 3.1) potřebovat 3x.

3.4 Zvolení redundance připojení

Než přejdu ke konfiguraci, musím zvolit vhodnou technologii redundance internetového připojení. Oba hlavní objekty organizace mají k dispozici jedno pevné internetové připojení přes ethernet a druhé záložní přes mobilního operátora, jenž je řešeno USB 4G modemem (obrázek 3.1). U IPv4 byl výpadek hlavní linky a přepnutí na záložní řešen změnou výchozí brány (a měnila se tak i veřejná IP adresa), na IPv6 volím řešení, kdy router bude odesílat v bezstavové autokonfiguraci informace o obou veřejných prefixech.

Aby bylo preferováno primární připojení, bude mít záložní prefix nižší prioritu.



Obr. 3.2: Návrh redundance internetového připojení

Zdroj: *vlastní tvorba*

Zjednodušené schéma je na obrázku 3.2, jenž ukazuje, jaké adresy budou mít koncová zařízení a jak bude provoz směrován.

3.5 Konfigurace routerů

V následující části popisuji návrh konfigurace routeru. Jak konkrétní kroky implementovat na fyzickém zařízení si lze přečíst v příloze A.

3.5.1 Instalace

Instalace systému bude provedena na SSD disk, doporučené rozložení je v tabulce 3.3 – je využito 48 GB pro systém, 16 GB odkládací prostor a zbytek bude využit jako cache pro rotační datové disky. Kromě toho je nutné jednotlivé oddíly pojmenovat nebo je připojovat přes UUID, jinak se v budoucnu s přidáním potenciálního dalšího disku změní pořadí disků a systém nenaběhne.

Po instalaci vytvořím na datových discích oddíl `/monitor` s využitím zrcadlení na ZFS a s cache na posledním oddílu SSD disku. Jak takovéto oddíly vytvářet se lze dočíst v mé bakalářské práci [27].

Zařízení	Velikost	Typ	Přípojný bod
ada2	128 GB	GPT	
ada2p1	512 KB	freebsd-boot	
ada2p2	48 GB	freebsd-ufs	`/`
ada2p3	16 GB	freebsd-swap	`none`
ada2p4	64 GB	freebsd-zfs	`none`

Tab. 3.3: Rozložení systémového disku

Zdroj: *vlastní tvorba*

3.5.2 Konfigurace síťových adaptérů

Konfigurace adaptérů bude následovná: jako WAN bude sloužit port 1 a jako LAN (se všemi virtuálními sítěmi – VLAN) port 2. Na záložní připojení je využit LTE modem, který je připojen přes USB. Rozhraním nastavím příslušné adresy a jako výchozí bránu nastavuji bránu pro primární připojení. Poté vytvořím pro každou virtuální síť VLAN zařízení a bridge, spojím je a na bridge nastavím adresu podle plánu v tabulce 3.2. Adresy jsou statické a pro případ manuálního nastavení jsou IPv6 adresy vždy ve tvaru `PREFIX::1`. Každému bridge navíc vygeneruji jinou MAC adresu, aby automatické nastavení v počítačích vyhodnotilo každou podsíť síť jako jinou síť a použilo jiné nastavení firewallu.

3.5.3 IPSec

Pro řešení zabezpečeného spojení jsem se rozhodl použít *IPSec model*, protože potřebuji propojit pouze 5 nezávislých segmentů přes internet a jednoduchost konfigurace a obsluhy převažuje nad nevýhodou potenciálně delší konfigurace. Jelikož každá z přípojek může případně vypadnout, využiji pro zabezpečení architekturu full-mesh – v IPv6 navíc není nutné jednotlivým spojení přiřazovat privátní adresy, proto se tato síť bude jevit pro koncová zařízení zcela transparentně, přestože je šifrovaná. Použit bude tedy transportní režim s ESP hlavičkou (šifrování i datové části).

Pro šifrování jsem zvolil šifrovací algoritmus `aes-ctr` s délkou klíče 288 bitů. IPSec umožňuje použít jiný klíč pro každou cestu, a jelikož existuje 10 kombinací možných spo-

jení, je celkem potřeba vygenerovat 20 klíčů. Přiřazením jednotlivých klíčů k jednotlivým cestám vzniknou bezpečnostní asociace. Bezpečnostní asociace ukládám do zvláštního souboru, neb jsou stejné pro všechny routery.

Pak zbývá zadat bezpečnostní politiky, ty už jsou zadávané pro každý router zvlášť a znamená to 18 pravidel u každého routeru. Nastavuji vyžadování[3] transportního režimu.

3.5.4 Autokonfigurace

Autokonfiguraci jsem se rozhodl provést dle RFC 3736, což znamená bezstavové získání adresy s tím, že adresa DNS serveru je získána stavově z DHCPv6 serveru. Protože se vyskytují i zařízení, která DHCPv6 nepodporují, budu i adresu DNS serveru šířit i bezstavovým způsobem, což by mělo být kompatibilní s většinou zařízení.

Pokud má zařízení více než jedno připojení, je potřeba šířit oba prefixy – primárnímu připojení nastavuji střední prioritu a záložnímu připojení nízkou prioritu. Podle RFC 3736 router šíří následující konfigurační volby `M=0` a `O=1`, díky nimž zařízení podporující DHCP6 protokol získají adresu DNS serveru a doménové jméno pro hledání místních zařízení.

Jelikož ale ne všechna zařízení DHCP6 protokol podporují, je šířena i bezstavová konfigurace ohledně DNS serveru a doménového jména.

Abych docílil, že zařízení bude zjišťovat dostupnost/nedostupnost výchozí brány, použiji konfigurační volbu *ReachableTime*, jež používá algoritmus pro detekci nedostupnosti sousedů v definovaný RFC 4861. Tento čas se nastavuje v ms a neměl by přesáhnout 1 hodinu. Na druhou stranu čím menší hodnota, tím se zvyšuje provoz v síti, poněvadž se jednotlivá zařízení stále dotazují na existenci brány. Protože tento algoritmus funguje tak, že zjišťuje pouze existenci souseda (a nikoliv celé trasy), použiji jednoduchý program, který (podobně jako v IPv4) zjišťuje průchodnost celé trasy a pokud není průchodná, nastaví ve firewallu pravidlo, aby zařízení byla poslána ICMPv6 zpráva „no route to destination“ (ICMPv6 typ `1`, hodnota `0`), z čehož by zařízení mělo poznat, že má použít jinou trasu a zároveň vypne šíření tohoto prefixu pomocí bezstavové autokonfigurace.

[3] pakety které dorazí nešifrované budou zahozeny

3.5.5 Firewall

Zde je dobré podotknout, že v tomto místě se nenastavují pravidla pro IPSec, konfiguraci IPSec totiž obsluhuje přímo jádro a do firewallu se už dostávají pakety, na které byly aplikovány bezpečnostní asociace.

Moje konfigurace firewallu se skládá ze 4 částí: sdílené konfiguraci tabulek adres, sdílené konfigurace překladu adres (NAT), sdíleného pravidla pro filtrování paketů a individuální konfigurace pro daný segment sítě.

Sdílená konfigurace tabulek adres: firewall pf má dvě metody, jak používat v pravidlech více adres. První z nich jsou seznamy, kdy je každé pravidlo expandované kartézským součinem použitých seznamů, což má pak zásadní vliv na výkon při průchodu paketu firewallem. Další možností jsou tabulky adres, které mají zpracování podstatně rychlejší, ale nelze už z vytvořené tabulky tvořit tabulky další. Řešením je adresy zadávat jako seznamy, z nich vytvářet tabulky a (pokud je to možné) ty teprve používat v pravidlech. Firewall pf umí kombinovat IPv4 a IPv6 pravidla v jedné tabulce a není tak potřeba mít zvlášť pravidla pro každou verzi protokolu, což výrazně zjednodušuje konfiguraci a může zabránit bezpečnostním chybám, kdy administrátor něco zakáže pouze pro jeden protokol.

Sdílená konfigurace překladu adres: překlad adres, neboli NAT, používám pouze v protokolu IPv4. V případě IPv6 ale v těchto místech nastavuji routování podle zdroje – provoz z hlavního a záložního prefixu se musí posílat na příslušnou bránu. Na rozdíl od routovacích tabulek ale neprobíhá rozhodnutí podle cíle, ale podle zdroje, proto se tato část realizuje na firewallu.

Sdílená konfigurace filtru: obsahuje pravidla pro filtrování provozu, např.: DNS pouze zevnitř sítě, zakázání provozu mezi jednotlivými sítěmi a otevření portů pro správu. K filtrování přistupuji s filosofií, že bude všechno zakázáno a jen potřebné věci se budou jednotlivě povolovat. Firewall je nastaven tak, aby ze všech sítí byl povolen provoz směrem do internetu, ale zakázán je provoz z internetu do vnitřní sítě a provoz mezi jednotlivými sítěmi.

Individuální konfigurace: konfigurace, která je odlišná pro různé routery. Zejména se bude jednat o směřování NAT těch zařízení, která musí být dostupná z IPv4 sítě.

3.5.6 DNS

Kromě toho, že systém DNS musí umět překládat doménová jména v internetu, bude sloužit také pro překlad jmen místních zařízeních. Jelikož informace o jménech a adresách místních zařízení není možné v případě IPv6 získat z jednoho centrálního zdroje (např. jako DHCP serveru v IPv4), samotné získání jmen a adres místních zařízení bude dále popsáno v sekci 3.10.3.

Konfigurace v tomto bodě bude obsahovat primární DNS zónu pro domény obsluhované routerem v konkrétní budově (v tabulce 3.2) vč. zóny pro zpětný překlad adres. U této zóny musí být povolen transfer domény z ostatních routerů (aby mohly vždy replikovat celou zónu a nemuseli se individuálně dotazovat).

Dále konfigurace bude obsahovat sekundární DNS zóny pro překlad z ostatních routerů s příslušným primárním DNS serverem – což je v mém případě příslušný router.

Další nastavení, co by měla konfigurace obsahovat, je zapnutí zabezpečení DNSSec pro všechny ostatní domény a DNS servery pro dotazování získané od poskytovatele připojení[4]. Nakonec v konfiguraci zapnu povolení rekruzivního dotazování, z bezpečnostních důvodů je ve firewallu zakázáno přijímat DNS dotazy z internetu.

3.5.7 Dodatečná konfigurace

Router s konfigurací, jak jsem ji výše popsal, bude plnit požadavky zadání, ale pouze na IPv6 protokolu. Aby síť fungovala i na IPv4 protokolu, musím ještě nakonfigurovat VPN mezi sítěmi a musím na vnitřní adresy nějakým způsobem šířit informace o routách mezi jednotlivými routry.

Pro realizaci VPN použiji protokol *OpenVPN* v režimu tunelování paketů[5] a jako transportní protokol použiji UDP.

Routování na IPv4 nechám dynamické, pouze jako protokol použiji OSPFv3, který bude oproti používanému protokolu OSPF mít podporu i IPv6 ač ho používáme pouze pro protokol IPv4.

[4] to ale není absolutně nutné, pokud tyto servery nebudou zadány, bude se DNS server dotazovat přímo kořenových serverů

[5] v terminologii OpenVPN: *tun* mód

Poslední z dodatečné konfigurace je nastavení DHCP serveru, jehož nastavení a funkcionalita bude stejná jako v případě aktuálního řešení.

3.6 Konfigurace serverů

Serverům lze přiřazovat adresy dvěma způsoby, buď si ji daný server vygeneruje z MAC pomocí EUI-64, nebo mu bude přidělena staticky. Navrhuji pro server, který je k dispozici veřejnosti statickou krátkou adresu (aby si ji bylo možno v rozumném čase opsat/napsat), a pro ostatní servery, které slouží organizaci interně, aby si adresu generovaly z MAC adresy a prefixu získaném z bezstavové autokonfigurace.

Servery budou k dispozici i pomocí DNS, tudíž pro uživatele nebude použití IPv6 adres znamenat žádný diskomfort a ani nepoznají, že se připojují pomocí jiné technologie.

3.7 Konfigurace switchů

Organizace používá konfigurovatelné switche/routery **MikroTik CSS326**, z nichž některé byly původně nakonfigurované jako routery. Můj návrh počítá s využitím těchto zařízení pouze jako switchů. Zásadní změna ve switchování je v tom, že nově se nebude míchat provoz tagovaného a netagovaného provozu. U každého portu tak bude nastaveno, která VLAN bude pro něj netagovaná a ostatní VLAN na tento port nebudou posílány. Výjimkou je spojení mezi routerem a switchem a mezi centrálními switchy, kde je naopak veškerý provoz přenášen ve jako tagovaný.

3.8 Konfigurace koncových zařízení

Konfiguraci koncových zařízení zajišťuje autokonfigurace. Koncová zařízení však generují pomocí RFC 4941 takzvanou dočasnou adresu a adresu trvalou randomizují, takže není možné ji získat výpočtem z MAC adresy.

U některých zařízení, například těch, na které se zaměstnanci přihlašují z domu pomocí vzdálené plochy, preferuji možnost, aby se tato adresa neměnila. Uživatel i tak bude komunikovat se světem pomocí dočasné adresy a zmenšuje se tak riziko, že útočník zatočí na jeho pracovní stanici na základě historicky zachycené adresy. Postup tohoto nastavení je v příloze C.

Pro zaměstnance to znamená komfort, že se mohou ke své pracovní stanici připojit přímo z domu (pokud jejich poskytovatel podporuje IPv6), aniž by se předtím museli připojit na prostředníka, nebo zadávat jim přidělený NATtovaný port. Počítače, ke kterým se lze takto připojit, specifikuji v seznamu ve firewallu, povoleno je připojení na TCP a UDP port 3389 (protokol RDP). Není možné povolit tento port globálně, jelikož by se pak útočník mohl zmocnit kontroly nad strojem se slabým heslem.

3.9 Konfigurace ostatních síťových zařízení

Ostatní síťová zařízení v organizaci, především Wifi AP od výrobce Mikrotik, síťové tiskárny a VOIP zařízení.

3.9.1 Podpora v zařízeních Mikrotik

Zásadním problémem zařízení s operačním systémem Mikrotik RouterOS, na kterých je síť původně stavěná, je, že jako DHCPv6 server neumí přidělovat koncovou adresu, pouze prefix sítě s maximální délkou prefixu `/64`. Zařízení rovněž neakceptuje adresu z bezstavové autokonfigurace, proto je nutné ji nastavit manuálně (stačí prefix).

3.9.2 Podpora v síťových tiskárnách

Všechna zařízení v organizaci podporují IPv6 do té míry, že akceptují prefix získaný automatickou konfigurací a to k jejich provozování postačuje.

3.9.3 Podpora ve VOIP zařízeních

Podpora IPv6 protokolu je ve VOIP zařízeních dobrá, právě u VOIP je možné plně docenit výhody IPv6 protokolu bez existence technologie NAT. Zařízení vedu v seznamu ve firewallu, povolen je port UDP+TCP 5060 (SIP) a rozsah portů pro RTP protokol (například UDP 11000-12000). V konfiguraci zařízení je nutné nastavit stejný rozsah portů. Povolení konkrétních portů je nutné kvůli bezpečnosti, jinak by mohl útočník útočit na administraci tohoto zařízení.

3.10 Monitoring provozu

Jak může být z kapitoly ohledně autokonfigurace zřejmé, autoři IPv6 preferovali anonymitu koncových stanic před správou sítě, obzvláště sítí, kde její správce nemůže plně ovlivňovat koncová zařízení. Na následujících stránkách se pokusím nastínit řešení tohoto problému. Pro monitoring provozu plánuji použít použití protokolu NetFlow, obvyklá infrastruktura se pro monitoring se skládá ze tří částí: Flow exporter, Flow collector a Analysis application.

3.10.1 Flow exporter

Exportování informací o provozu je ve FreeBSD integrováno přímo v jádře systému pomocí subsystému netgraph. Tento subsystém umožňuje kompletní manipulaci se síťovými zařízeními a exportování informací o provozu je jedna z mnoha jeho schopností. Tento systém konfiguruji tak, aby informace z každého bridge exportoval jako jiný datový proud, abych potom mohl přesně identifikovat konkrétní rozhraní v případě problémů.

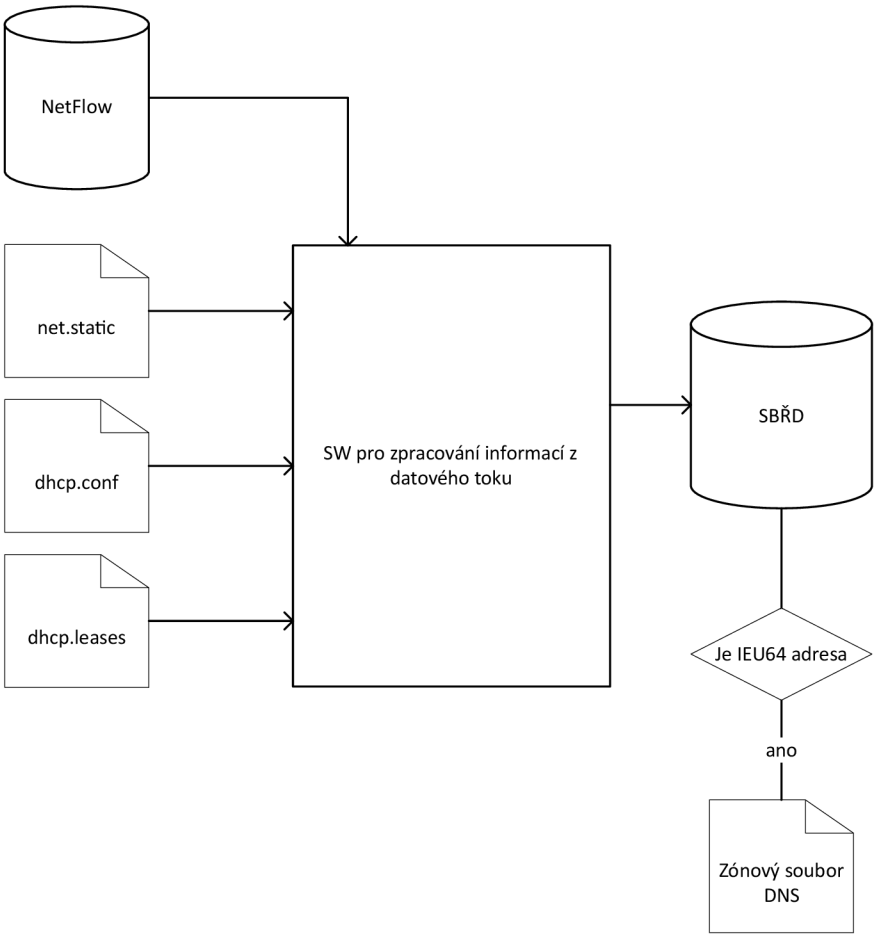
3.10.2 Flow collector

K ukládání dat může sloužit ve FreeBSD SW *nfcapd*, není nutné ho manuálně spouštět a lze k komu použít SW jménem *nfsen*. Samotný tento software je možné použít i pro analýzu, ale neposkytne mám úplně všechny informace o síti, které potřebuji. Konfiguraci *nfcapd* provedu tak, aby se ukládal každý proud do zvláštní databáze.

Alternativou je SW *ipfixcol* od sdružení CESNET, bohužel FreeBSD (a obecně BSD systémy) není podporovaný.

3.10.3 Analysis application

Jednou z aplikací na analýzu informací o datovém toku je i již zmíněný *nfsen*, který umí kreslit grafy a poskytuje tak informace o zatížení sítě. Po aplikaci určitých filtrů umí detekovat i některé z DoS útoků. Já však potřebuji něco víc, v IPv6 mi chybí nějaká centrální služba, která by byla schopna poskytovat informace o zařízeních, která se v síti vyskytují. V IPv4 to byl například DHCP server, nicméně v IPv6 je úloha DHCPv6 serveru patrně trochu jiná. Z těchto důvodů jsem se rozhodl v návrhu řešení navrhnou SW, který bude řešit tento problém. Jak bude daný SW fungovat a jak bude bude získávat potřebné informace, je zobrazeno na obrázku 3.3



Obr. 3.3: Analytický SW pro IPv6 síť

Zdroj: *vlastní tvorba*

Činnost tohoto SW je jednoduchá: pomocí **nfdump** jsou načítána data z NetFlow a k nim jsou i načteny informace z DHCP serveru a z jeho konfigurace (přidělení statické adresy) a ze souboru `net.static`, kde budou umístěny MAC adresa a název zařízení se statickou IP adresou. Informace o adresách (IPv4 - lokální; IPv6 - globální; IPv6 linková a MAC adresa) a k nim přiřazené jméno hostitele (získané z DHCP nebo staticky přidělené) se uloží do databáze. K dané adrese se uloží informace o poslední komunikaci a tím dokážeme odhalit, jestli je zařízení online a také kdy online skutečně bylo.

Databáze nám bude poskytovat informaci o zařízeních v dané síti a také o čase, kdy v síti opravdu byla online a komunikovala.

Další vlastností, kterou bude tento SW poskytovat, je generování zónového souboru DNS, aby bylo možné se na jednotlivá zařízení připojovat pomocí DNS jména.

3.11 Bezpečnost

IPv6 má na rozdíl od svého předchůdce[6] integrovanou bezpečnost, bohužel postupem času se našly chyby v návrhu a rovněž se objevují bezpečnostní chyby v některých implementacích.

3.11.1 Rotování typu 0, RFC 5095

Stará chyba, podobná k ekvivalentní v IPv4, zmiňuji ji jen z historických důvodů a v aktuálních implementacích by se již neměla nacházet a podle RFC 5095 by měly routery zahazovat paket pokud obsahuje tuto hlavičku.

3.11.2 Fragmentace hlaviček, RFC 7112

Jak bylo popsáno výše (1.1.3), díky fragmentaci hlaviček bylo možné způsobovat chybu typu DoS. Nápravu této chyby popisuje dokument RFC 7112 a v aktuálních implementacích by měla být opravena, správce sítě by si měl dát pozor na neaktualizovaný SW ke staršímu HW, který může být k chybě náchylný.

3.11.3 Falešné routery, RFC 6104

Mechanizmus útoku je podobný jako u falešného DHCP serveru na IPv4, na rozdíl od IPv4 mohou mít rozhraní více adres a pomocí protokolu ICMPv6 se mnohem lépe tento útok propaguje do konfigurace jednotlivých zařízení. Pakety (resp. rámce) navíc neprocházejí routerem a šíří je switche, které ze své podstaty nezajímá obsah datové části rámců, jež šíří sítí. Jsou v podstatě možná 4 řešení:

Pouze statické adresy: toho řešení je vhodné pro malé sítě, nebo pro sítě, kde je prioritou výkon/bezpečnost nad flexibilitou konfigurace. Toto řešení však nevyhovuje požadavkům organizace, takže nemůže být v mém případě použito.

Kryptografické bezpečné objevování sousedů: RFC 3971 definovalo protokol *SEND*, který má za úkol zajistit bezpečné výměny zpráv. Toto řešení má tu nevýhodu, že není k dispozici automaticky na koncových zařízení, dá se dokonce tvrdit, že podpora je mizerná

[6] v IPv4 je možno také používat IPSec, ale je jen dodatečným protokolem a nemusí být implementován vůbec

[1, s.110]. Vzhledem k tomu, jak špatně je implementována autokonfigurace v současné době, bude patrně trvat dlouho, než se tato možnost dostane do koncových zařízení.

RA Guard: toto není samotný software, ale soubor postupů, jak ochránit zařízení, která sama sebe ochránit nedokážou. Toto řešení lze realizovat pomocí L2 firewallu, který má za cíl blokovat všechny rámce[7], které obsahují IPv6 paket, který obsahuje protokol ICMPv6 s typem Router advertisement[8] nebo Redirect[9] a není z důvěryhodného zdroje. Nevýhodou je jistý výkonnostní deficit na rychlost switchování, protože na většině zařízení toto nastavení vypíná HW obvody pro rychlé switchování rámců a přepíná je softwarově pomocí CPU. Navrhuji toto nastavení použít pro veřejné a zaměstnanecké sítě, kde není velký datový tok a je poměrně velké riziko, že zde bude připojeno zařízení útočníka.

Aktuálně u některých switchů Mikrotik (např. organizací užívané CRS326) existuje hardwarová ochrana před falešnými DHCP servery, který nemá vliv na výkon, předpokládám, že se objeví stejný mechanismus, určený pro IPv6, jelikož princip je stejný, jen se liší protokoly.

Eliminace neplatných routerů: princip spočívá v detekci falešných oznámení, kdy se pošle stejné oznámení s nulovou životností, zároveň je možné informaci logovat, nebo spustit nějakou akci (kupříkladu přidat zdrojovou MAC adresu na černou listinu). Na realizaci toho řečení lze použít SW jménem Ramond[28]. Toto řešení navrhuji použít pro ethernetovou sít, kde by L2 firewall způsoboval výkonností problémy.

3.11.4 Mikrotik a CVE-2018-19298, CVE-2018-19299

Routery Mikrotik donedávna[10] obsahovaly chybu, kdy bylo možné způsobit DoS útok upraveným paketem. Proto je nutné používat všude verzi 6.44.2 – v opačném případě je síť vystavena vážnému bezpečnostnímu riziku.

Chyba byla opravena jen na zařízeních, které mají 64 MB a více RAM, u starších zařízeních, která mají méně, lze sice IPv6 kompletně vypnout, ale pak zase není možné bránit útoku popsaném v RFC 6104 (popsaném výše v 3.11.3).

[7] switch opravdu pracuje s rámcem a zkoumá obsah jeho datové části

[8] číslo typu: 134

[9] číslo typu: 137

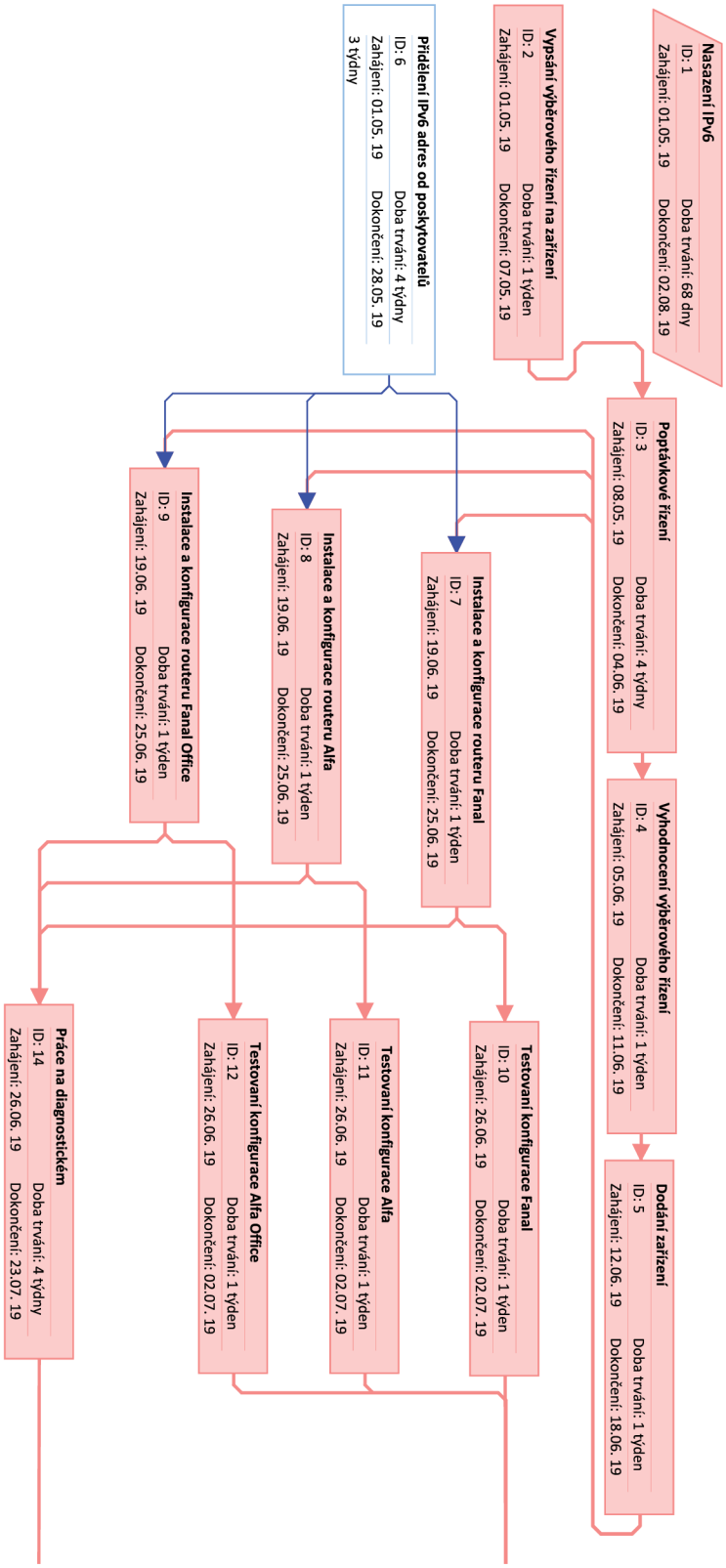
[10] Psáno v dubnu 2019, kdy oprava vyšla 1. dubna 2019

3.12 Časový plán

V této části práce se věnuji časovému plánování realizace, tabulka jednotlivých činností v tabulce 3.4 a pomocí CPM analýzy vychází, že projekt bude trvat **68 dní**. Síťový graf je na obrázcích 3.4 a 3.5. Jak je vidno ze síťového grafu, téměř všechny činnosti leží na kritické cestě a je velké riziko, že se projekt zpozdí. Koncové datum (2. 8.) vychází na divadelní prázdniny, tudíž případné zdržení projektu nebude znamenat žádnou katastrofu.

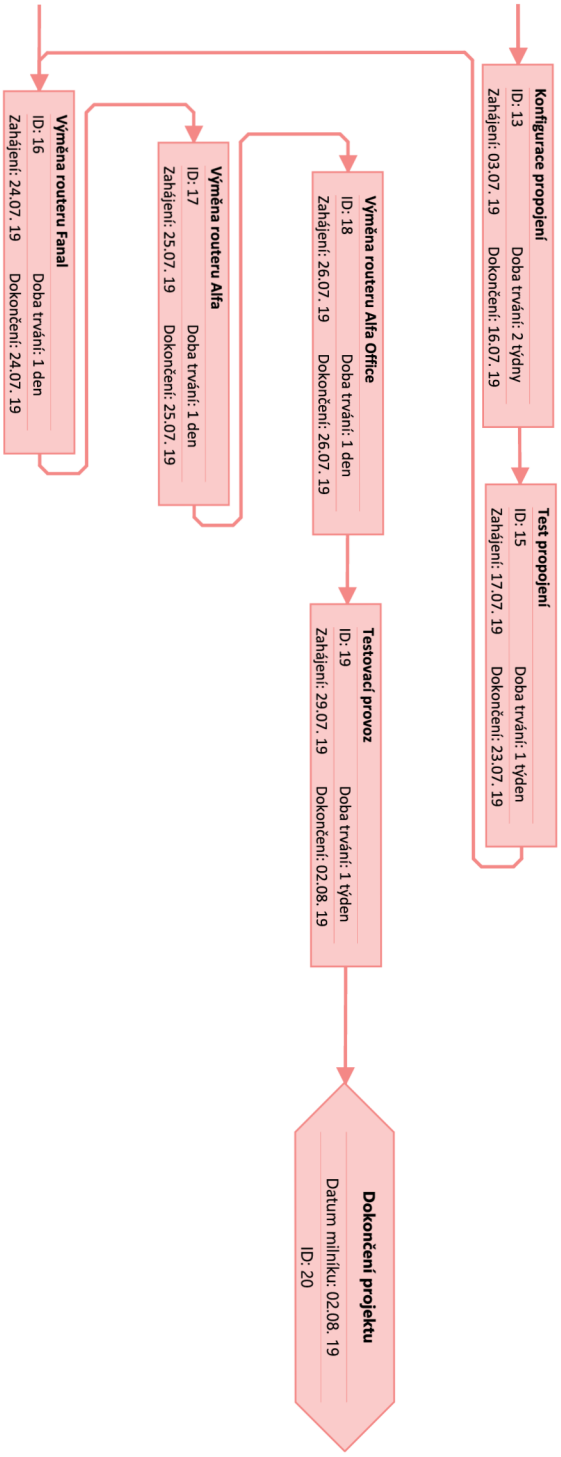
ID	Název	Doba trvání	Zahájení	Dokončení	Předchůdci
1	Nasazení IPv6	68 dny	1. 5. 2019	2. 8. 2019	
2	Vypsání výběrového řízení na zařízení	1 týden	1. 5. 2019	7. 5. 2019	
3	Poptávkové řízení	4 týdny	8. 5. 2019	4. 6. 2019	2
4	Vyhodnocení výběrového řízení	1 týden	5. 6. 2019	11. 6. 2019	3
5	Dodání zařízení	1 týden	12. 6. 2019	18. 6. 2019	4
6	Přidělení IPv6 adres od poskytovatelů	4 týdny	1. 5. 2019	28. 5. 2019	
7	Inst. a konfig. routeru Fanal	1 týden	19. 6. 2019	25. 6. 2019	5;6
8	Inst. a konfig. routeru Alfa	1 týden	19. 6. 2019	25. 6. 2019	5;6
9	Inst. a konfig. routeru Fanal Office	1 týden	19. 6. 2019	25. 6. 2019	5;6
10	Testovaní konfigurace Fanal	1 týden	26. 6. 2019	2. 7. 2019	7
11	Testovaní konfigurace Alfa	1 týden	26. 6. 2019	2. 7. 2019	8
12	Testovaní konfigurace Alfa Office	1 týden	26. 6. 2019	2. 7. 2019	9
13	Konfigurace propojení	2 týdny	3. 7. 2019	16. 7. 2019	10;11;12
14	Práce na diagnostickém SW	4 týdny	26. 6. 2019	23. 7. 2019	7;8;9
15	Test propojení	1 týden	17. 7. 2019	23. 7. 2019	13
16	Výměna routeru Fanal	1 den	24. 7. 2019	24. 7. 2019	14;15
17	Výměna routeru Alfa	1 den	25. 7. 2019	25. 7. 2019	16
18	Výměna routeru Alfa Office	1 den	26. 7. 2019	26. 7. 2019	17
19	Testovací provoz	1 týden	29. 7. 2019	2. 8. 2019	18
20	Dokončení projektu	0 dny	2. 8. 2019	2. 8. 2019	19

Tab. 3.4: Tabulka činností



Obr. 3.4: Síťový graf, část I.

Zdroj: *vlastní tvorba*



Obr. 3.5: Síťový graf, část II.

Zdroj: *vlastní tvorba*

3.13 Ekonomické zhodnocení

Pro celkové ekonomické zhodnocení je potřeba znát množství hodin, cenu HW a platové tabulky (příspěvková organizace stanovuje mzdu pomocí mzdových tabulek pro státní zaměstnance). Množství hodin je v tabulce 3.5, a po přepočtu pomocí platových tabulek vyjdou přímé mzdy (v tab. 3.6).

ID	Název	Hodiny
2	Vypsání výběrového řízení na zařízení	6
3	Poptávkové řízení	4
4	Vyhodnocení výběrového řízení	4
5	Dodání zařízení	1
6	Přidělení IPv6 adres od poskytovatelů	2
7	Inst. A konfig. routeru Fanal	40
8	Inst. A konfig. routeru Alfa	40
9	Inst. A konfig. routeru Fanal Office	40
10	Testovaní konfigurace Fanal	40
11	Testovaní konfigurace Alfa	40
12	Testovaní konfigurace Alfa Office	40
13	Konfigurace propojení	80
14	Práce na diagnostickém SW	160
15	Test propojení	40
16	Výměna routeru Fanal	8
17	Výměna routeru Alfa	8
18	Výměna routeru Alfa Office	8
19	Testovací provoz	40
20	Dokončení projektu	0
1	Nasazení IPv6	601

Tab. 3.5: Množství hodin

Zdroj: *vlastní tvorba*

Člověko-hodin	601
Tabulkový plat (třída 10, stupeň 5)[29, Příloha č. 1]	22 180,00
Cena za hodinu	132,02
Celkem	79 344,02
Daň (za zaměstanvatele)	26 976,98
Celkem mzdy (Kč)	106321,00

Tab. 3.6: Přímé mzdy

Zdroj: *vlastní tvorba*

Položka	nabídka			Průměr	Počet	Cena
	1	2	3			
HPE MicroServer Gen10 (X3418)	9 762[11]	10 322[12]	10 322[13]	10 135	3	30 406
Intel SSD DC S3110 - 128GB	1 341[14]	1 160[15]	1 264[16]	1 255	3	3 765
Seagate IronWolf 6TB	4 016[17]	4 131[18]	4 049[19]	4 065	6	24 392
Celkem (Kč)						58 563

Tab. 3.7: Cena za HW, bez DPH

Zdroj: *vlastní tvorba*, *zdroje cen v poznámce*

Cena za HW zahrnuje samotný server a k němu disky určené pro 24hodinový provoz. Datové disky není nutné používat přímo serverové, ale měly by mít 7200 ot./min. Nejlevnější disk těchto parametrů (WD RED PRO 4 TB) v době psaní této práce[20] byl dražší,

[11] zdroj: czc.cz
[12] zdroj: bscom.cz
[13] zdroj: alza.cz
[14] zdroj: czc.cz
[15] zdroj: xevos.store
[16] zdroj: lan-shop.cz
[17] zdroj: czc.cz
[18] zdroj: alza.cz
[19] zdroj: tsbohemia.cz
[20] leden až květen 2019

nebo vycházel cenově hodně podobně jako 6 TB disk těchto parametrů. Do cenové kalkulace jsem tedy zahrnul větší variantu.

Přímé mzdy	106321,00
Pořízení majetku	58 563,00
Celkem (Kč)	164 884,00

Tab. 3.8: Rekapitulace nákladů

Zdroj: *vlastní tvorba*

Součet mezd a ceny za HW je v tabulce 3.8. Jelikož se jedná o příspěvkovou organizaci, náklady investice nelze použít ke snížení základu daně z příjmu. Celkové náklady na změnu jsou podle mého návrhu **164 884 Kč**.

3.14 Doporučení pro budoucí činnosti

Navrhl jsem ucelenou, funkční a bezpečnou variantu. Doporučuji v případě dalšího rozvoje implementovat dále následující technologie:

3.14.1 IPv6-only sít

Jak už název napovídá, do budoucna bude zajímavé transformovat síť, který podporuje pouze IPv6 protokol a zařízení v ní mají pouze IPv6 adresu. Konfigurace je překvapivě jednodušší, jelikož nám odpadne například celá část 3.5.7. Aby byl přístupný i IPv4-only internetu, je potřeba ještě nasadit další dvě služby:

NAT64

NAT64 (neplést s technologií NAT z IPv4) překládá IPv4 adresy na IPv6 a tak umožňuje z IPv6-only sítí obsáhnout ty části internetu, které nemají IPv6 adresu.

DNS64

Obdobně jako NAT64, DNS64 transformuje IPv4 adresy z DNS na IPv6 adresy. DNS64 má však pár problémů, prvním z nich je, že u takto přeložených adres neprojde validace pomocí DNSSec, což ale vzhledem k tomu, že tuto validaci dělá resolver, není obvykle potřeba tento problém řešit.

Další problém se týká technologie **DNS-over-HTTPS**, kterou někteří tvůrci webových prohlížečů obcházejí klasický překlad adres. Na druhou DNS64 je jen minoritním problémem této technologie, mnohem větším je nedostupnost místních jmen zařízení předkládaný pouze lokálním resolverem a (paradoxně) i problém se soukromím.

ZÁVĚR

Moje práce ukazuje, že protokol IPv6 je možné nasadit ve firemním prostředí a pokud jsou zvoleny jeho správné součásti, poskytuje protokol mnohem větší flexibilitu než jeho stále ještě rozšířený zástupce IPv4. Cíle práce se tak podařilo naplnit.

Vzhledem k tomu, že se autoři v protokolu IPv6 snažili vyřešit problémy všech počítačových sítí, stalo se zIPv6 bohužel monstrum, u kterého není snad žádná implementace úplná, což správcům sítí dává za povinnost vybrat a používat ty části, které jsou v zařízeních v jeho síti správně implementovány. Také je vidět, že nebylo nalezeno ekvilibrium mezi anonymitou koncových stanic a monitoringem sítě.

Závěrem autor doufá, že se protokol IPv6 bude dostávat do podnikových sítí mnohem častěji a povede to tak k tvorbě a rozšíření nástrojů, které si momentálně musí správci sítí vytvářet sami.

SEZNAM POUŽITÉ LITERATURY

1. SATRAPA, Pavel. *IPv6: Internetový protokol verze 6*. 3., aktualiz. a dopl. vyd. Praha: CZ.NIC, 2011. ISBN 978-80-904248-4-5.
2. DEERING, S.; HINDEN, R. *Internet Protocol, Version 6 (IPv6) Specification* [RFC 1883 (Proposed Standard)]. Fremont, CA, USA: RFC Editor, 1995. Internet Request for Comments, č. 1883. ISSN 2070-1721. Dostupné z DOI: 10.17487/RFC1883. Obsoleted by RFC 2460.
3. DEERING, S.; HINDEN, R. *Internet Protocol, Version 6 (IPv6) Specification* [RFC 8200 (Internet Standard)]. Fremont, CA, USA: RFC Editor, 2017. Internet Request for Comments, č. 8200. ISSN 2070-1721. Dostupné z DOI: 10.17487/RFC8200.
4. CALETKA, Ondřej. *Nové RFC řeší problém zřetězených IPv6 hlaviček* [online]. Praha: Internet Info, 2017 [cit. 2019-05-08]. Dostupné z: https://www.root.cz/zpravicky/nove-rfc-resi-problem-zretezenych-ipv6-hlavicek/.
5. MCKUSICK, Marshall Kirk.; NEVILLE-NEIL, George V. *The design and implementation of the FreeBSD operating system*. Boston: Addison-Wesley, 2005. ISBN 02-017-0245-2.
6. *IANA IPv6 Special-Purpose Address Registry* [online]. Los Angeles: IANA, 2017 [cit. 2019-05-08]. Dostupné z: https://www.iana.org/assignments/iana-ipv6-special-registry/iana-ipv6-special-registry.xhtml.
7. KHANNA, Sunil. *IPv6: Neighbor Discovery Protocol (NDP)* [online]. San Jose: Cisco Systems, 2011 [cit. 2019-05-08]. Dostupné z: https://community.cisco.com/t5/networking-documents/ipv6-neighbor-discovery-protocol-ndp/ta-p/3125000.
8. JEONG, J.; PARK, S.; BELOEIL, L.; MADANAPALLI, S. *IPv6 Router Advertisement Options for DNS Configuration* [RFC 8106 (Proposed Standard)]. Fremont, CA, USA: RFC Editor, 2017. Internet Request for Comments, č. 8106. ISSN 2070-1721. Dostupné z DOI: 10.17487/RFC8106.
9. STANEK, William R. *Mistrovství v Microsoft Windows Server 2008: [kompletní informační zdroj pro profesionály]*. 1. vydání. Brno: Computer Press, 2009. ISBN 978-80-251-2158-0.

10. NARTEN, T.; DRAVES, R.; KRISHNAN, S. *Privacy Extensions for Stateless Address Autoconfiguration in IPv6* [RFC 4941 (Draft Standard)]. Fremont, CA, USA: RFC Editor, 2007. Internet Request for Comments, č. 4941. ISSN 2070-1721. Dostupné z DOI: 10.17487/RFC4941.

11. NARTEN, T.; NORDMARK, E.; SIMPSON, W.; SOLIMAN, H. *Neighbor Discovery for IP version 6 (IPv6)* [RFC 4861 (Draft Standard)]. Fremont, CA, USA: RFC Editor, 2007. Internet Request for Comments, č. 4861. ISSN 2070-1721. Dostupné z DOI: 10.17487/RFC4861. Updated by RFCs 5942, 6980, 7048, 7527, 7559, 8028, 8319, 8425.

12. DRAVES, R.; THALER, D. *Default Router Preferences and More-Specific Routes* [RFC 4191 (Proposed Standard)]. Fremont, CA, USA: RFC Editor, 2005. Internet Request for Comments, č. 4191. ISSN 2070-1721. Dostupné z DOI: 10.17487/RFC4191.

13. BAKER, F.; CARPENTER, B. *First-Hop Router Selection by Hosts in a Multi-Prefix Network* [RFC 8028 (Proposed Standard)]. Fremont, CA, USA: RFC Editor, 2016. Internet Request for Comments, č. 8028. ISSN 2070-1721. Dostupné z DOI: 10.17487/RFC8028.

14. *About the FreeBSD Project* [online]. The FreeBSD Project [cit. 2019-05-08]. Dostupné z: https://www.freebsd.org/doc/en_US.ISO8859-1/books/handbook/history.html.

15. *PF* [online]. The FreeBSD Project [cit. 2019-05-08]. Dostupné z: https://www.freebsd.org/doc/handbook/firewalls-pf.html.

16. *RADVD.CONF(5)* [online]. Boulder: FreeBSD Project, 2011 [cit. 2019-05-08]. Dostupné z: https://www.freebsd.org/cgi/man.cgi?query=radvd.conf.

17. *ISC DHCP* [online]. Redwood City: Internet Systems Consortium, c2019 [cit. 2019-05-08]. Dostupné z: https://www.isc.org/downloads/dhcp/.

18. BOTT, Ed; SIECHERT, Carl; STINSON, Craig. *Mistrovství Microsoft Windows 10*. 1. vydání. Brno: Computer Press, 2017. ISBN 978-80-251-4869-3.

19. BOLLAPRAGADA, Vijay; KHALID, Mohamed; WAINNER, Scott. *IPSec VPN design: the definitive design and deployment guide for secure virtual private networks*. Indianapolis: Cisco Press, 2005. ISBN 15-870-5111-7.

20. *O nás* [online]. Brno: Centrum experimentálního divadla, 2019 [cit. 2019-05-08]. Dostupné z: https://www.ced-brno.cz/cs/ced/about/.

21. *Organizační struktura* [online]. Brno: Centrum experimentálního divadla, 2009 [cit. 2019-05-08]. Dostupné z: https://www.ced-brno.cz/toolkit/download.php?file=2024.

22. *Zřizovací listina Centra experimentálního divadla, p. o.* [online]. Brno: Centrum experimentálního divadla, 2009 [cit. 2019-05-08]. Dostupné z: http://www.ced-brno.cz/toolkit/download.php?file=978.

23. *Historie* [online]. Brno: Divadlo Husa na provázku, 2014 [cit. 2019-05-08]. Dostupné z: https://www.provazek.cz/historie.

24. *Historie* [online]. Brno: HaDivadlo, 2019 [cit. 2019-05-08]. Dostupné z: https://www.hadivadlo.cz/info/sekce-3/.

25. *Zpráva o výsledku hospodaření a činnosti 2018* [online]. Brno: Centrum experimentálního divadla, 2019 [cit. 2019-05-08]. Dostupné z: https://www.ced-brno.cz/toolkit/download.php?file=2028.

26. KRČMÁŘ, Petr. *MikroTik má chybu v IPv6 stacku, upravený paket může shodit zařízení* [online]. Praha: Internet Info, 2019 [cit. 2019-05-08]. Dostupné z: https://www.root.cz/zpravicky/mikrotik-ma-chybu-v-ipv6-stacku-upraveny-paket-muze-shodit-zarizeni/.

27. HENSL, Jaroslav. *Zálohování dat a datová úložiště s využitím pokročilých funkcí souborových systémů*. Brno, 2016. Bakalářská práce. Vysoké učení technické v Brně, Fakulta podnikatelská, Ústav informatiky. Vedoucí práce Jiří Kříž.

28. *Router Advert MONitoring Daemon* [online]. Southampton: University Of Southampton, 2011 [cit. 2019-05-08]. Dostupné z: http://ramond.sourceforge.net/.

29. *Nařízení vlády č. 263/2018 Sb., kterým se mění některá nařízení vlády v oblasti odměňování zaměstnanců ve veřejných službách a správě a státních zaměstnanců.* Praha: Sbírka zákonů, 2018. Č. 263. ISSN 1211-1244.

30. *Basic VLAN switching* [online]. Riga: SIA Mikrotīkls, 2019 [cit. 2019-05-08]. Dostupné z: https://wiki.mikrotik.com/wiki/Manual:Basic_VLAN_switching.

SEZNAM OBRÁZKŮ

SEZNAM TABULEK

SEZNAM ZKRATEK

AES Advanced Encryption Standard, standardizovaný algoritmus používaný k šifrování dat, symetrická šifra. 52

ARP Address Resolution Protokol. 24

CPM Metoda kritické cesty = Critical Path Method, algoritmus pro plánování množiny úkolů v projektu. 71

DNS Domain name system. 22

DoS Denial of service, útok jehož cílem je znepřístupnit danou službu ostatním uživatelům. 29, 67

ECDH Elliptic-curve Diffie–Hellman, šifrovací protokol, který umožňuje dvěma stranám, které spolu nikdy nekomunikovaly, přenášet informace po nezabezpečeném kanálu. 52

GUI Graphic user interface = grafické uživatelské rozhraní. 111

HW Hardware. 32, 33, 59, 69, 70, 74–76, 111

ICMPv6 Internet Control Message Protocol Version 6, protokol pro šíření chyb a diagnostiku sítě. 26, 30, 62, 69, 70, 107, 110

IKE Internet Key Exchange = internetový protokol na výměnu kryptografických klíčů. 33, 38

IPv4 Internetový protokol verze 4. 15–17, 19, 21, 24, 26, 28, 29, 33–35, 56, 59, 62–64, 67–69, 77, 78, 93–95, 101, 103, 107

IPv6 Internetový protokol verze 6. 9, 13, 15–19, 21, 22, 24, 26–29, 32–35, 53, 55–57, 59, 61, 63–70, 77, 78, 89, 94, 95, 103, 107, 109, 111, 113

NAS Network Attached Storage = datové uložitelně v sítí. 43

NAT Network Address Translation, mechanizmus pro překlad síťových adres. 29, 63, 66, 77

NAT64 NAT64. 77

pf Packet filter, firewall pro BDS systémy. 33, 63, 101, 103, 107

RFC Request For Comments = žádost o komentáře, dokument(y) popisující zejména internetové protokoly a zajišťující jejich standardizaci. 15

RSA Rivest–Shamir–Adleman, šifrovací algoritmus v veřejným a soukromím klíčem, asymetrická šifra. 52

SW Software. 23, 32, 33, 67–70, 90, 98, 101, 107

URL URL = Uniform Resource Locator. 23

VPN Virtual private network, virtuální počítačová síť bez vazby na fyzické zapojení. 44, 49, 52, 64

SEZNAM POUŽITÝCH RFC

RFC 1752 The Recommendation for the IP Next Generation Protocol. 25

RFC 1825 Security Architecture for the Internet Protocol. 35

RFC 1883 Internet Protocol, Version 6 (IPv6) Specification. 17, 18

RFC 2401 Security Architecture for the Internet Protocol. 35

RFC 2928 Initial IPv6 Sub-TLA ID Assignments. 25

RFC 3056 Connection of IPv6 Domains via IPv4 Clouds. 25

RFC 3736 Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6. 27, 62

RFC 3849 IPv6 Address Prefix Reserved for Documentation. 25

RFC 3971 SEcure Neighbor Discovery (SEND). 69

RFC 4191 Default Router Preferences and More-Specific Routes. 30

RFC 4193 Unique Local IPv6 Unicast Addresses. 25

RFC 4291 IP Version 6 Addressing Architecture. 25

RFC 4301 Security Architecture for the Internet Protocol. 35

RFC 4380 Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs). 25

RFC 4843 An IPv6 Prefix for Overlay Routable Cryptographic Hash Identifiers (ORCHID). 25

RFC 4861 Neighbor Discovery for IP version 6 (IPv6). 30, 33, 62

RFC 4862 IPv6 Stateless Address Autoconfiguration. 27

RFC 4941 Privacy Extensions for Stateless Address Autoconfiguration in IPv6. 28, 65

RFC 5095 Deprecation of Type 0 Routing Headers in IPv6. 69

RFC 5180 IPv6 Benchmarking Methodology for Network Interconnect Devices. 25

RFC 5952 A Recommendation for IPv6 Address Text Representation. 22

RFC 6052 IPv6 Addressing of IPv4/IPv6 Translators. 25, 108

RFC 6106 IPv6 Router Advertisement Options for DNS Configuration. 27

RFC 6666 A Discard Prefix for IPv6. 25

RFC 7112 Implications of Oversized IPv6 Header Chains. 21, 69

RFC 7343 An IPv6 Prefix for Overlay Routable Cryptographic Hash Identifiers Version

2 (ORCHIDv2). 25

RFC 7450 Automatic Multicast Tunneling. 25

RFC 7534 AS112 Nameserver Operations. 25

RFC 7535 AS112 Redirection Using DNAME. 25

RFC 7723 Port Control Protocol (PCP) Anycast Addresses. 25

RFC 7954 Locator/ID Separation Protocol (LISP) Endpoint Identifier (EID) Block. 25

RFC 8028 First-Hop Router Selection by Hosts in a Multi-Prefix Network. 30, 83

RFC 8106 IPv6 Router Advertisement Options for DNS Configuration. 27

RFC 8155 Traversal Using Relays around NAT (TURN) Server Auto Discovery. 25

RFC 8190 Updates to the Special-Purpose IP Address Registries. 25

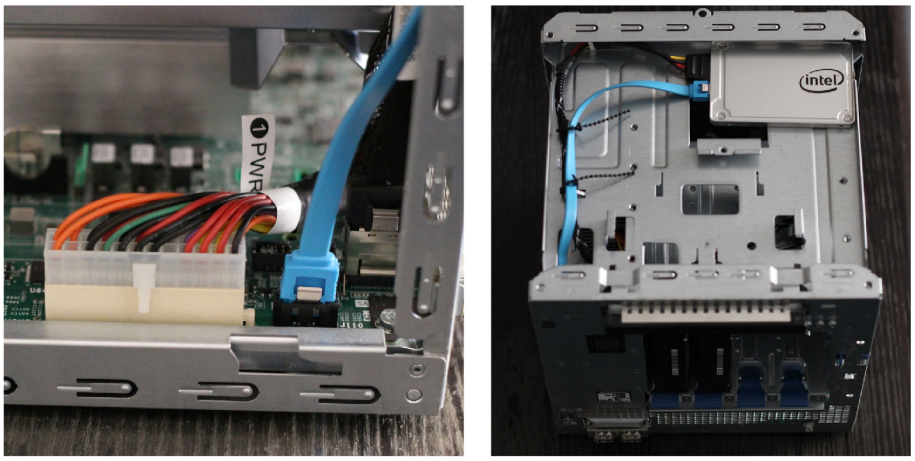
RFC 8200 Internet Protocol, Version 6 (IPv6) Specification. 17, 18, 29

RFC 8215 Local-Use IPv4/IPv6 Translation Prefix. 25

SEZNAM PŘÍLOH

A PŘÍLOHA: PODROBNÁ KONFIGURACE ROUTERU/FREEBSD

V této příloze se věnuji konkrétní konfiguraci jednotlivých částí z implementace na FreeBSD ve verzi 12.0. Kromě samotné konfigurace popisuji i stručně instalaci systému, která není úplně triviální. Většina SW se instaluje z repositáře binárních balíčků. V místech, kde je potřeba něco kompilovat ze zdrojových kódů, na tuto skutečnost upozorňuji. Konfigurace je pokud možno zestručněna, aby čtenář z ní pochopil princip a byl ji schopen aplikovat na řešení svého vlastního problému.



Obr. A.1: Zapojení pevných disků na HPE MicroServer Gen 10

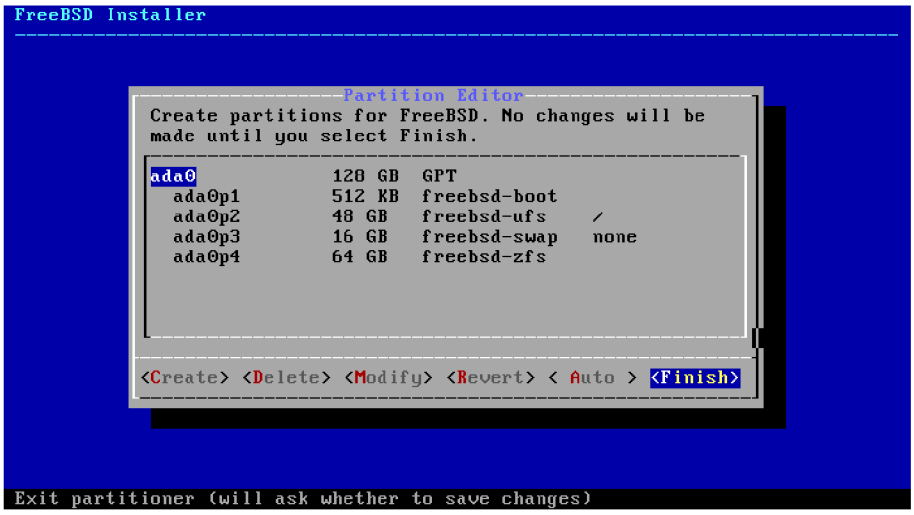
Zdroj: *vlastní tvorba*

A.1 Instalace systému

FreeBSD si úplně nerozumí s AMD APU[1], a proto je potřeba zavést jádro s parametrem `hw.pci.realloc_bars=1`. Jednoduše to lze provést tak, že je při bootování zvolena volba 3 (escape to loader promt) a jsou napsány tyto dva příkazy:

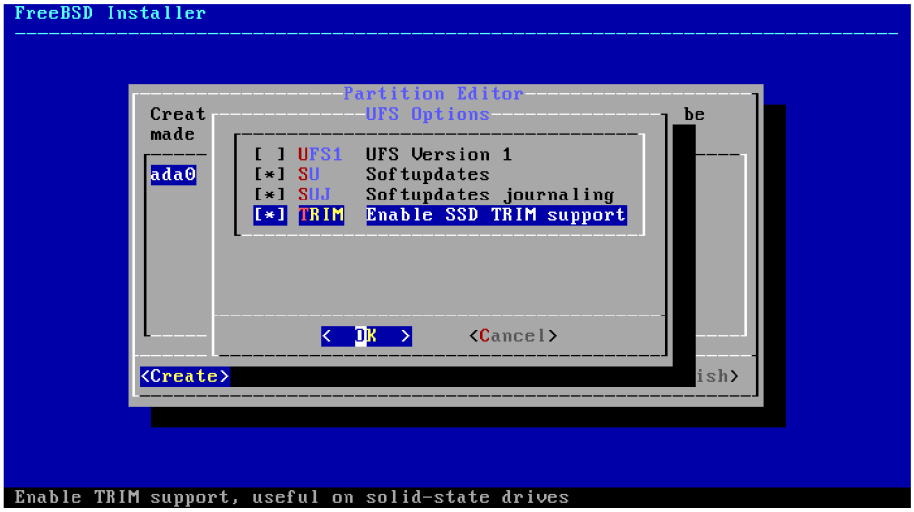
```
1 set hw.pci.realloc_bars=1
2 boot
```

[1] Grafická karta a CPU integrována v jednom pouzdře



Obr. A.2: Rozdělení pevného disku

Zdroj: *vlastní tvorba*



Obr. A.3: Zapnutí podpory technologie trim

Zdroj: *vlastní tvorba*

Zbytek instalace je pomocí průvodce docela snadné provést (rozdělení disku je na obrázku A.2 – důležité je zapnout podporu technologie trim, obrázek A.3), jen před koncem doporučuji, aby se systém nerestartoval. Místo toho je vhodné přejít do konzole a přidat parametr, který jsme zadávali manuálně do nastavení zavaděče, a další užitečná věc je pojmenovat jednotlivé diskové oddíly, aby se daly připojovat, i když se změní pořadí disků.

FreeBSD disky čísluje podle pořadí detekce. Jelikož systémový disk je v portu číslo 5 (viz obrázek A.1), je vhodné oddíly pojmenovat[2], aby nebylo nedošlo přidáním/odebráním disků k jejich posunutí. Je to nutné udělat zde z toho důvodu, že to není možné provést, pokud jsou oddíly připojené. Pokud se systémový disk jmenuje `ada`, lze to provést následujícími příkazy:

```
1 echo 'hw.pci.realloc_bars=1' >> /boot/loader.conf
2 tunefs -L gwsys /dev/ada2
3 tunefs -L gwswap /dev/ada3
4 glabel label gwsys /dev/ada2
5 glabel label gwswap /dev/ada3
6 glabel label gwcache /dev/ada4
```

Nakonec udělám editaci souboru `/etc/fstab`. K tomu lze použít například editor `vi`. Jakmile tento krok dokončím, můžu operační systém restartovat (příkazem `exit`). Obsah souboru `/etc/fstab` je následující:

```
1 # Device          Mountpoint      FStype  Options Dump    Pass#
2 /dev/label/gwsys  /               ufs     rw      1       1
3 /dev/label/gwswap none            swap    sw      0       0
```

V nově naběhlém systému je nutné ještě provést několik úprav. Například přepnout systém do UTF-8, stáhnout možné bezpečnostní aktualizace a aktualizovat balíčkovací systém, ze kterého budeme instalovat software.

Pro podporu UTF-8 je třeba editovat soubor `/etc/login.conf`, administrátoři, kteří se úplně necítí při práci v editoru `vi`, mohou použít uživatelsky přívětivější editor `nano`. V tomto souboru je potřeba doplnit do sekce default kódování a jazyk a mělo by to vypadat takto:

```
1 default:\
2                 [...]
3                 :charset=UTF-8:\
4                 :lang=en_US.UTF-8:\
5                 :umask=022:
```

Pak je potřeba ještě aktualizovat profil příkazem:

```
cap_mkdb /etc/login.conf
```

[2] lze použít i připojování oddílů pomocí uuid, ale instalátor FreeBSD to z nějakých důvodů nevytváří

Aby fungoval všechen software, je ještě nutné přidat jméno systému do souboru `/etc/hosts`, to lze automaticky provést příkazem:

```
sh -c 'echo 127.0.0.1 $(hostname);echo ::1 $(hostname)' \
        >> /etc/hosts
```

Stáhnu a nainstaluji aktualizace systému:

```
freebsd-update fetch
freebsd-update install
```

Nakonec nainstaluji a aktualizuji balíčkovací systém.

```
pkg update
```

Po tomto kroku je instalace hotova.

A.2 Sdílená konfigurace

Aby byla konfigurace co nejvíce přenositelná, vytvářím následující souborovou strukturu:

```
/etc
 |---shared
       |--- pf.tables.conf
       |--- pf.nat.conf
       |--- pf.server.conf
       |--- setkey.conf
       |--- openvpn
              |--- ca.crt
```

Význam souborů je následující:

- `/etc/shared/pf.tables.conf` seznamy (tables) adres pro firewall.
- `/etc/shared/pf.nat.conf` společná konfigurace NAT pro IPv4.
- `/etc/shared/pf.server.conf` veřejných protokolů a portů pro servery
- `/etc/shared/setkey.conf` soubor pro ipsec obsahující šifrovací klíče.
- `/etc/shared/openvpn` složka obsahující veřejné klíče pro openvpn (pro IPv4)
- `/etc/shared/openvpn/ca.crt` veřejný certifikát certifikační autority pro openvpn.

A.3 Konfigurace síťových adaptérů

FreeBSD pojmenovává jednotlivé adaptéry podle konkrétního HW ovladače – v mém případě je to *bgeX* pro ethernet, kde *X* označuje pořadové číslo adaptéru a *ueX*[3] pro USB modem (*X* opět označuje pořadové číslo modemu). Jednotlivé porty jsou na obrázku A.4.

Obr. A.4: Zadní panel na HPE MicroServer Gen 10

Zdroj: *vlastní tvorba*

Veškeré nastavení se provádí v souboru `/etc/rc.conf`, nastavení portu do internetu bude pro *router Fanal* následující (IPv6 a IPv4):

```
1 ifconfig_bge0_ipv6="2001:db8:ced1::1/32"
2 ipv6_defaultrouter="2001:db8:aaaa::1"
3
4 ifconfig_bge0="inet 192.0.0.1/30"
5 defaultrouter="192.0.0.2"
```

Stejně tak provedeme nastavení USB modemu[4].

```
1 ifconfig_ue0_ipv6="2001:db8:ced2::1/32"
2 ifconfig_ue0="inet 192.0.0.3/30"
```

Není možné mít více výchozích bran. Mechanizmus jejich přepínání nebo směřování podle zdroje (nikoliv podle cíle) musíme měnit jiným způsobem – více o tom v sekci firewall. Nyní přichází nastavení vlan a vnitřních adres. Následujícím záznamem v `/etc/rc.conf` vytvořím požadované vlan, zařízení obsluhující VLAN má bude mít název ve tvaru *bgeX.Y*,

[3] Záleží na ovladači, v některých případech může být zařízení označeno jako *ndisX*.

[4] V některých případech dojde k automatickému získání IP adresy, ne však všech a také občas nedojde k automatickému získání IPv6 i IPv4 adresy, ale jen jedné u nich.

kde *X* je pořadové číslo fyzického adaptéru a *Y* je číslo VLAN. Ve FreeBSD je zařízení automaticky aktivováno, pokud je mu přidělena IP adresa, v mém případě to ale tak není, proto zařízení musím aktivovat manuálně:

```
vlans_bge1="1 2 3 4 5 6 7"
ifconfig_bge1="up"
```

Firewally v některých systémech detekují typ sítě podle mac adresy brány, rovněž většina monitorovacího software je schopna třídit informace podle mac adresy zachytávajícího zařízení. Protože potřebuji mac adresu pro každou VLAN jinou, mohu ji náhodně vygenerovat následujícím příkazem.

```
sh -c 'echo 02:$(openssl rand -hex 5 | sed "s/\(..\)/\1:/g ; s/:$//")'
```

První oktet vygenerované adresy je vždy 02_{16} neboli 00000010_2, nejméně důležitý bit je 0, což indikuje unicastovou adresu a druhý nejméně důležitý bit je 1 což indikuje tzv. lokálně generovanou adresu a nikoliv unikátní adresu přidělenou výrobcem.

Dále potřebuji vytvořit bridge, přidělit mu MAC adresu, spojit ho s VLAN a přidělit mu IPv4 a IPv6 adresu. Lze toho dosáhnou přidáním následujících řádků do `/etc/rc.conf`:

```
ifconfig_bge1_1="up"
ifconfig_bridge0="ether 02:3a:44:7f:d2:31"
ifconfig_bridge0_alias0="addm bge1"
ifconfig_bridge0_alias1="inet 192.168.1.254/24"
ifconfig_bridge0_alias2="inet6 auto_linklocal"
ifconfig_bridge0_ipv6="inet6 2001:db8:ced1:1000::1/64"
ifconfig_bridge0_alias3="inet6 2001:db8:ced2:1000::1/64"
```

Takto vypadá první bridge (`VLAN ID 1`) na zařízení *router Fanal*, pro další VLAN a zařízení lze postupovat analogicky, doporučuji věnovat pozornost tomu, že bridge jsou číslovány od 0.

Změny provedené v `/etc/rc.conf` se projeví až po restartu zařízení (nebo restartu příslušné služby).

A.4 IPSec

Konfigurace IPSec se nachází v souboru `/etc/setkey.conf` a příslušný program se jmenuje `setkey`. Konfigurační soubor ale nemá direktivu *include*, případně nějaký její ekvi-

valent. Pro mé účely budu potřebovat konfigurační soubory dva. Provedu drobnou změnu v souboru `/etc/rc.d/ipsec`. Funkce start by nově měla vypadat takto:

```
ipsec_start()
{
        echo "Installing ipsec manual keys/policies."
        ${ipsec_program} -f $ipsec_file
        if [ -n "$ipsec_file2" ]; then
            ${ipsec_program} -f $ipsec_file2
        fi
}
```

Je potřeba vygenerovat klíče pro šifrování provozu, zvolil jsem šifrovací algoritmus `aes-ctr` s délkou klíče 288 bitů. Vygenerovat náhodný klíč je možné následujícím příkazem:

```
openssl rand -hex 36
```

Vzhledem k tomu, že potřebuji propojit 5 bodů – což dává 10 cest – a šifrujeme každý směr zvlášť, je třeba 20 klíčů. Ke každému možnému propojení přiřadím jeden klíč - vniká nám tím tzv. bezpečnostní asociace (SA). Tyto bezpečnostní asociace se ukládají do souboru `/etc/shared/setkey.conf`, pro první kombinaci, tj. **fanal** *primární připojení* se všemi ostatními, vypadá tento konfigurační soubor takto:

```
add 2001:db8:ced1::/48 2001:db8:ced2::/48 esp 0x101
-m transport -E aes-ctr
0x8e44f91a864339f58849205bf1a254da667
b1b37af6574083f94716a60f9a112596deaa7 ;

add 2001:db8:ced1::/48 2001:db8:afa1::/48 esp 0x102
-m transport -E aes-ctr
0xc64b40123d01e53510aea9c385e162abf0c
fd743ef4ea2c5e8cc3032b420e87141e12837 ;

add 2001:db8:ced1::/48 2001:db8:afa2::/48 esp 0x103
-m transport -E aes-ctr
0xc38c7902768d3d9c360e1c6e2be1f0d5aa8
ba6a0be641a4923fcb2b84ecebe17a04241bd ;

```

```
16 add 2001:db8:ced1::/48 2001:db8:ffce:aa00::/56 esp 0x104
17 -m transport -E aes-ctr
18 0x5d9785c312de901edce1a26cf8b7dba31052
19 6723ae275ba5dbadeb2036b93e0a4532939e ;
```

Čísla 0x101 - 0x104 (257_{10} - 260_{10}) jsou označením čísla pravidla, toto číslo musí být větší než 255. Celková konfigurace je zkrácena, databáze bezpečnostních asociací (SAD) bude mít celkem 20 pravidel.

Nyní musím ještě vytvořit databázi bezpečnostní politiky (SPD), k tomu slouží soubor `/etc/setkey.conf` a jeho obsah se bude lišit pro každý router, pro *router Fanal* bude konfigurace následující:

```
1  spdadd 2001:db8:afa1::/48 2001:db8:ced1::/48
2  any -P in ipsec esp/transport//require ;
3  spdadd 2001:db8:ced1::/48 2001:db8:afa1::/48
4  any -P out ipsec esp/transport//require ;
5
6  spdadd 2001:db8:afa2::/48 2001:db8:ced1::/48
7  any -P in ipsec esp/transport//require ;
8  spdadd 2001:db8:ced1::/48 2001:db8:afa2::/48
9  any -P out ipsec esp/transport//require ;
10
11 spdadd 2001:db8:afa1::/48 2001:db8:ced2::/48
12 any -P in ipsec esp/transport//require ;
13 spdadd 2001:db8:ced2::/48 2001:db8:afa1::/48
14 any -P out ipsec esp/transport//require ;
15
16 spdadd 2001:db8:afa2::/48 2001:db8:ced2::/48
17 any -P in ipsec esp/transport//require ;
18 spdadd 2001:db8:ced2::/48 2001:db8:afa2::/48
19 any -P out ipsec esp/transport//require ;
20
21 spdadd 2001:db8:afa2::/48 2001:db8:ced2::/48
22 any -P in ipsec esp/transport//require ;
23 spdadd 2001:db8:ced2::/48 2001:db8:afa2::/48
24 any -P out ipsec esp/transport//require ;
25
26 spdadd 2001:db8:ffce:aa00::/56 2001:db8:ced1::/48
```

```
27 any -P in ipsec esp/transport//require ;
28 spdadd 2001:db8:ced1::/48 2001:db8:ffce:aa00::/56
29 any -P out ipsec esp/transport//require ;
30
31 spdadd 2001:db8:ffce:aa00::/56 2001:db8:ced2::/48
32 any -P in ipsec esp/transport//require ;
33 spdadd 2001:db8:ced2::/48 2001:db8:ffce:aa00::/56
34 any -P out ipsec esp/transport//require ;
35
36 spdadd 2001:db8:ced1::/48 2001:db8:ced2::/48
37 any -P in ipsec esp/transport//require ;
38 spdadd 2001:db8:ced1::/48 2001:db8:ced2::/48
39 any -P out ipsec esp/transport//require ;
40
41 spdadd 2001:db8:ced2::/48 2001:db8:ced1::/48
42 any -P in ipsec esp/transport//require ;
43 spdadd 2001:db8:ced2::/48 2001:db8:ced1::/48
44 any -P out ipsec esp/transport//require ;
```

Poslední 4 pravidla nejsou důležitá ve smyslu, že by jimi měl téct nějaký provoz, ale pokud by tam nebyla, mohl by toho vyžít útočník na cestě a posílat takové pakety, které by router akceptoval.

Nakonec musím ještě přidat do souboru `rc.conf` následující řádky:

```
1 ipsec_enable="YES"
2 ipsec_file="/etc/shared/setkey.conf"
3 ipsec_file2="/etc/setkey.conf"
```

Tímto příkazem tyto bezpečnostní politiky aplikujeme:

```
service setkey start
```

Pokud bychom se chtěli vyhnout manuálnímu generování klíčů a využít jejich automatickou obnovu, můžeme pro to použít SW, který implementuje IKE a jmenuje se **racoon**. Vzhledem k tomu, jaké množství tras se snažíme zabezpečit, konfigurace nebude jednoduší, ale lze tím zvýšit bezpečnost v případě, že by se ke klíčům nějakým způsobem dostal útočník.

A.5 Autokonfigurace

Nejdříve nainstaluji příslušný software, což je DHCP server (implementace DHCP od ISC), a Router advertisement daemon (radvd), který bude šířit bezstavovou autokonfiguraci:

```
pkg install isc-dhcp44-server
pkg install radvd
```

Začnu konfigurací radvd, která se nachází v souboru `/usr/local/etc/radvd.conf`. Pro bridge s `VLAN 1` na zařízení *router Fanal* bude vypadat takto.

```
interface bridge0 {
        MaxRtrAdvInterval 600;
        MinRtrAdvInterval 60;
        AdvSendAdvert on;

        prefix 2001:db8:ced1:1000::/64 {
                AdvOnLink               on;
                AdvAutonomous           on;
                AdvManagedFlag          off;
                AdvOtherConfigFlag      on;
                AdvDefaultPreference    medium;
        };

        prefix 2001:db8:ced2:1000::/64 {
                AdvOnLink               on;
                AdvAutonomous           on;
                AdvManagedFlag          off;
                AdvOtherConfigFlag      off;
                AdvDefaultPreference    low;
        };

        RDNSS 2001:db8:ced1:1000::1 {
                AdvRDNSSLifetime 3600;
        };
};
```

DNS server leží v naší vlastní síti, nemusíme proto pro něj uvádět alternativní prefix.

Podobně konfiguruji další můstky. Přejdeme ke konfiguraci DHCPv6 serveru, jež se nachází v souboru `/usr/local/etc/dhcpd6.conf`. Ukázkovou konfiguraci a doporučené výchozí hodnoty si lze přečíst v souboru `dhcpd6.conf.sample` ve stejné složce. DHCP se nemapuje podle jednotlivých zařízení, ale podle toho v jakém rozsahu mají adresu, v následující ukázce je moje konfigurace DHCPv6 serveru včetně nastavení pro *router Fanal*, můstek s vlan 1, ostatní můstky lze přidat analogicky.

```
1  # IPv6 address valid lifetime
2  #  (set to 30 days, the usual IPv6 default)
3  default-lease-time 2592000;
4
5  # IPv6 address preferred lifetime
6  #  (set to 7 days, the  usual IPv6 default)
7  preferred-lifetime 604800;
8
9  # T1, the delay before Renew
10 #  (set to 1 hour)
11 option dhcp-renewal-time 3600;
12
13 # T2, the delay before Rebind (if Renews failed)
14 #  (default is 3/4 preferred lifetime)
15 option dhcp-rebinding-time 7200;
16
17 # Enable RFC 5007 support (same than for DHCPv4)
18 allow leasequery;
19
20 # The delay before information-request refresh
21 #  (set to 6 hours)
22 option dhcp6.info-refresh-time 21600;
23
24 # The path of the lease file
25 dhcpv6-lease-file-name "/var/db/dhcpd6.leases";
26
27 subnet6 2001:db8:ced1:1000::/64 {
28     option dhcp6.domain-search "office.net.ced-brno.cz";
29     option dhcp6.name-servers 2001:db8:ced1:1000::1;
30 }
```

Kromě toho je ještě potřeba nakonfigurovat DHCP server i pro IPv4, ta se nachází v souboru `/usr/local/etc/dhcpd.conf`[5]. Opět globální konfigurace a konfigurace pro *router Fanal*, můstek s vlan 1:

```
default-lease-time 86400;
max-lease-time     259200;

authoritative;

lease-file-name "/var/db/dhcpd.leases";

subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.40 192.168.1.199;
    option domain-name "office.net.ced-brno.cz";
    option domain-name-servers 192.168.1.254;
    option routers 192.168.1.254;
    option broadcast-address 192.168.1.255;
};
```

Pro aplikaci konfigurace je ještě nutno doplnit do souboru `/etc/rc.conf` následující řádky:

```
dhcpd_enable="YES"
dhcpd_ifaces="bridge0 bridge1 bridge2 bridge3 bridge4 bridge5 bridge6"
dhcpd6_enable="YES"
dhcpd6_ifaces="bridge0 bridge1 bridge2 bridge3 bridge4 bridge5 bridge6"
rtadvd_enable="YES"
rtadvd_interfaces="bridge0 bridge1 bridge2 bridge3 bridge4 bridge5
    bridge6"
```

A.6 Firewall

FreeBSD podporuje několik SW firewallů, v této práci jsem zvolil firewall pf původem z OpenBSD. Požadavky organizace jsou následující: povolit provoz mezi zařízeními v sítích office.net.ced-brno.cz, theatre.net.hadivadlo.cz a office.net.hadivadlo.cz. Dále musí

[5]Direktivy pro DHCP a DHCPv6 konfiguraci nelze mít v jednom souboru.

být povolen provoz mezi sítí control.zvuk.net.provazek.cz a device.zvuk.net.provazek.cz. Ve všech ostatních nesmí být provoz mezi sítěmi umožněn.

Značnou část konfigurace je možné sdílet mezi jednotlivými routery, takto bude vypadat obsah souboru `/etc/shared/pf.tables.conf`

```
addr_office = "{ 192.168.1.0/24 192.168.2.0/24 192.168.3.0/24
    172.16.32.0/24 172.16.65.0/24 2001:db8:ced1:1000::/64 2001:db8:ced2
    :1000::/64 2001:db8:afa1:1000::/64 2001:db8:afa2:1000::/64 2001:db8
    :ffce:aa00::/64 }"
addr_fanal_public = "{ 10.0.0.0/16 2001:db8:ced1:a000::/64 2001:db8:
    ced1:a000::/64 }"
addr_fanal_host = "{ 10.1.0.0/16 2001:db8:ced1:a100::/64 2001:db8:
    ced1:a100::/64 }"
addr_fanal_zvuk_public = "{ 10.4.1.0/16 2001:db8:ced1:b000::/64
    2001:db8:ced2:b000::/64 }"
addr_fanal_zvuk_control = "{ 10.4.1.0/16 2001:db8:ced1:b100::/64
    2001:db8:ced2:b100::/64 }"
addr_fanal_zvuk_device = "{ 10.4.1.0/16 2001:db8:ced1:b200::/64
    2001:db8:ced2:b200::/64 }"
addr_fanal_svetla = "{ 10.5.0.0/16 2001:db8:ced1:c000::/64 2001:db8:
    ced2:c000::/64 }"
addr_alfa_public = "{ 10.2.0.0/16 10.7.0.0/16 2001:db8:afa1:a000::/64
     2001:db8:afa2:a000::/64 ${prefix_alfa_office}A0::/64 }"
addr_alfa_host = "{ 10.3.0.0/16 2001:db8:afa1:a100::/64 2001:db8:
    afa2:a100::/64 }"
addr_alfa_zvuk = "{ 10.8.0.0/16 2001:db8:afa1:b000::/64 2001:db8:
    afa2:b000::/64 }"
table <net_office> const { $addr_office }
table <net_fanal_public> const { $addr_fanal_public }
table <net_fanal_host> const { $addr_fanal_host }
table <net_fanal_zvuk_device> const { $addr_fanal_zvuk_device }
table <net_fanal_zvuk_control> const { $addr_fanal_zvuk_control }
table <net_fanal_zvuk_public> const { $addr_fanal_zvuk_public }
table <net_fanal_svetla> const { $addr_fanal_svetla }
table <net_alfa_public> const { $addr_alfa_public }
table <net_alfa_host> const { $addr_alfa_host }
table <net_alfa_video> const { $addr_alfa_video }
```

```
21 table <net_alfa_zvuk>          const { $addr_alfa_zvuk }
22 table <net_locals> const { $addr_office $addr_fanal_public
       $addr_fanal_host $addr_fanal_zvuk_public $addr_fanal_device
       $addr_fanal_zvuk_control $addr_fanal_svetla $addr_alfa_public
       $addr_alfa_host $addr_alfa_video }
23
24 table <servers> const { 192.168.12 2001:db8:ced1:1000::12 2001:db8:ced2
       :1000::12 192.168.1.200 2001:db8:ced1:1000:4261:86ff:fe86:7481 }
```

Jedná se o seznamy adres, které patří k určitému typu sítí a z těch jsou vytvořeny tabulky adres – pf neumí vytvářet tabulky z tabulek, tudíž vytvářím makra obsahující seznam a z těch potom vytvářím tabulky adres. Dále budu věnovat pozornost konfiguračnímu souboru `/etc/shared/pf.nat.conf`, kde konfiguruji nat. Kromě konfigurace IPv4 jsem do tohoto souboru umístil směrování IPv6, aby podle zdrojové adresy byla komunikace směrována na konkrétní rozhraní a konkrétní bránu.

```
1 table <rfc1918>  const { 192.168.0.0/16, 172.16.0.0/12, 10.0.0.0/8 }
2
3 nat on $if_wan from <rfc1918> to !<rdc1918> -> ($if_wan)
4 nat on $if_wan_backup from <rfc1918> to !<rdc1918> -> ($if_wan_backup)
5
6 pass in on ! $if_wan route-to ( $if_wan $gw6_wan ) from $range6
7 pass in on ! $if_wan_backup route-to ( $if_wan_backup $gw6_wan_backup
      ) from $range6_backup
```

Poslední ze sdílené konfigurace je `/etc/shared/pf.filter.conf`, kde jsou umístěna pravidla pro filtrování paketů. Obsah souboru vypadá takto:

```
1 block all
2
3 pass out      all      keep state
4 pass in       proto    icmp
5 pass from     <net_office>        to <net_office>         keep state
6 pass from     <net_fanal_public>  to <net_fanal_public>   keep state
7 pass from     <net_fanal_host>    to <net_fanal_host>     keep state
8 pass from     <net_fanal_zvuk_device>    to <net_fanal_device>     keep
      state
9 pass from     <net_fanal_zvuk_control>    to <net_fanal_zvuk_control>
      keep state
```

```
10 pass from    <net_fanal_zvuk_public>   to <net_fanal_zvuk_public>
       keep state
11 pass from    <net_fanal_zvuk_control>   to <net_fanal_zvuk_device>
       keep state
12 block quick from    <net_fanal_zvuk_devicel>   to ! <
       net_fanal_zvuk_control>    keep state
13 pass from    <net_fanal_svetla>  to <net_fanal_svetla>  keep state
14 pass from    <net_alfa_public>   to <net_alfa_public>   keep state
15 pass from    <net_alfa_host>     to <net_alfa_host>     keep state
16 pass from    <net_alfa_video>    to <net_alfa_video>    keep state
17 pass from    <net_alfa_vodafone> to <net_alfa_vodafone> keep state
18
19 pass from <net_locals>    to !<net_locals>   keep state
20 pass proto tcp from any   to  <servers> port { 22, 3389, 80, 443 } keep
       state
21 pass proto tcp form any   to  <voip> port { 5060 } keep state
22 pass proto udp form any   to  <voip> port { 5060, 11000-12000 } keep
       state
23 pass proto tcp from any   to  <rdp> port { 3389 } keep state
```

Velmi jednoduše se dá říct, že je povolen provoz pouze v určité síti a zakázán do ostatních síti.

Nakonec obsah souboru `/etc/pf.conf` bude vypadat takto, tohle je verze pro zařízení *router Fanal* pro ostatní zařízení se bude konfigurace lišit jen v adresách:

```
1  if_wan = "bge0"
2  if_wan_backup = "ue0"
3
4  $gw6_wan = "2001:db8:aaaa::1"
5  $range6 = "2001:db8:ced1::/48"
6
7  $gw6_wan_backup = "2001:db8:bbbb::2"
8  $range6_backu = "2001:db8:ced2::/48"
9
10 table <voip> const { }
11 table <rdp> const { }
12
13 include "/etc/shared/pf.tables.conf"
```

```
14 include "/etc/shared/pf.nat.conf"
15 include "/etc/shared/pf.filter.conf"
16
17 pass     in from <net_locals>  proto   tcp  to any  port { domain }
       keep state
18 pass     in from <net_locals>  proto  udp  to any  port { domain }
            keep state
19
20 pass     in       proto   icmp
21 pass     in       proto   icmp6
22
23 pass     in       inet6   proto   ipv6-icmp       all keep state
24 pass     in       inet6   proto   udp            to any  port { 546, 547
       }
```

A.7 Konfigurace netgraph

Ve FreeBSD je integrovaný systém v jádře zvaný **netgraph**, který umožňuje komplexní manipulaci se síťovými zařízeními. Z něho v této práci využívám pouze modul ng_netflow, který mi umožní informace o provozu posílat na určené zařízení. Ač by to mohlo být i jiné zařízení, zvolený router má dost výkonu a kapacity, aby bylo možné informace o provozu zpracovávat přímo na routeru.

```
1 netgraph_load="YES"
2 ng_ether_load="YES"
3 ng_socket_load="YES"
4 ng_ksocket_load="YES"
5 ng_tee_load="YES"
6 ng_netflow_load="YES"
```

Vytvořím soubor `/etc/netflow.conf`, pro vlan 1 a 2 bude konfigurace následující.

```
1 mkpeer bge1.1: netflow lower iface0
2 name bge1.1:lower nfbg1_1
3 connect bge1.1: nfbg1_2: upper out0
4 mkpeer nfbg1_1: ksocket export9 inet/dgram/udp
5 msg nfbg1_1: setconfig {iface=0 conf=7}
6 msg nfbg1_1:export9 connect inet/127.0.0.1:9991
```

```
7
8  mkpeer bge1.2: netflow lower iface0
9  name bge1.2:lower nfbg1_2
10 connect bge1.2: nfbg1_2: upper out0
11 mkpeer nfbg1_2: ksocket export9 inet/dgram/udp
12 msg nfbg1_2: setconfig {iface=0 conf=7}
13 msg nfbg1_2:export9 connect inet/127.0.0.1:9992
```

Přidání dalších vlan je samozřejmě analogické. Kromě toho vytvoříme spouštěcí soubor `/etc/rc.d/netflow.sh`:

```
1  #!/bin/sh
2  case "$1" in
3  'star')
4  /usr/sbin/ngctl -f /etc/netflow.conf
5  ;;
6  'stop')
7  ;;
8  *)
9  echo unknown directive $1
10 echo $0 "(start|stop)"
11 ;;
12 esac
```

A nastavíme ho pro spuštění příkazem:

```
chmod a+x /etc/rc.d/netflow.sh
```

A.8 Přepínání na záložní připojení

FreeBSD může mít více routovacích tabulek a ty pak přepínat příkazem `setfib`. Ze všeho nejdříve ale musíme specifikovat, kolik routovacích tabulek vlastně systém má mít – já kromě výchozí potřebuji 2 další, proto do souboru `/boot/loader.conf` doplním tento řádek:

```
1  net.fibs=3
```

Princip samotného programu je jednoduchý, na začátku pomocí příkazu `setfib` nastavím alternativní routovací tabulky:

```
setfib 1 route -4 add default $brana_ipv4
setfib 1 route -6 add default $brana_ipv6
setfib 2 route -4 add default $brana_ipv4_backup
setfib 2 route -6 add default $brana_ipv6_backup
```

Pak v kombinaci se `setfib` vyzkouším nějaký server v internetu, například pomocí programu `ping`

```
setfib 1 ping -c 5 -n -o 2001:4860:4860::8888
```

Příkaz skončí se stavovým kódem 0, pokud server alespoň jednou odpověděl. Zkouší to celkem 5x. Pokud primární připojení nefunguje, přehodím výchozí bránu:

```
route -4 add default $brana_ipv4_backup
route -6 add default $brana_ipv6_backup
```

Přidávám do firewallu pf následující pravidlo, které dá zařízení ICMPv6 najevo, že tato trasa není průchodná a má se použít jiná.

```
block return-icmp6 in from $prefix_wan quick
```

Nastavím, aby *radvd* daný prefix distribuoval s konfigurační volbou `AdvDefaultLifetime 0`, což znamená, že daný prefix není výchozí.

Pokud se stav změní a připojení bude zase online, stejným způsobem se přepnu na primární připojení (a odstraním pravidla pro pf a *radvd*). Pokud nefunguje záložní připojení, použiji obdobné pravidlo v pf a v *radvd*, abych dal zařízením v síti vědět, že alternativní cesta není průchodná. Rovněž testuji více serverů, protože je možné, že i server velkého poskytovatele může mít výpadek.

A.9 NAT64

Ve FreeBSD je k dispozici SW **tayga**, který poskytuje tuto funkcionalitu[6]. Software je k dispozici v balíčkovacím systému:

```
pkg install tayga
```

Nyní přikročím k editaci souboru `/usr/local/etc/tayga.conf`. Je v něm třeba zvolit privátní rozsah do kterého budou mapovány IPv6 adresy, jež se budou zasílat pakety do IPv4 světa. Dále musí být vyhrazen prefix o délce **/96**, to kterého budou mapovány IPv4

[6] poskytuje ji i firewall ipfw, pf ve FreeBSD verzi ale ještě ne

adresy, je možné je (dle manuálu) vyhradit ze stávajícího veřejného prefixu, nebo podle RFC 6052 použít prefix `64:ff9b::/96`.

```
tun-device nat64
ipv4-addr 10.255.0.1
prefix 64:ff9b::/96
dynamic-pool 10.255.0.0/16
data-dir /var/db/tayga
```

Hostitel s adresou `192.0.0.8` bude tedy dostupny jako

`64:ff9b::192.0.0.8`,

a je možné i adresy takto zapisovat.

A.10 DNS64

Konfigurace je triviální stačí do souboru `/usr/local/etc/namedb/named.conf` doplnit vyhrazený prefix:

```
dns64  64:ff9b::/96 {
    suffix ::;
};
```

B PŘÍLOHA: KONFIGURACE SWITCHŮ/ROUTEROS

V této příloze vysvětluji některé popisuji některé specifické nastavení na RouterOS od firmy Mikrotik.

B.1 Nastavení IPv6 adresy

Jak už bylo v této práci řečeno, zařízení s operačním systémem RouterOS neumí získat z bezstavové autokonfigurace. Pokud znám prefix a chci vygenerovat adresu pomocí EUI-64 identifikátoru, použiji tento příkaz:

```
1 /ipv6 address
2 add address=2001:db8:ced1:1000::/64 advertise=no eui-64=yes interface=
    ether1
```

Důležité je parametr `advertise=no`, jinak začne zařízení posílat *Router advertisment* zprávy, že je routerem. Pokud chci zadat konkrétní adresu, vynechám nastavím `eui-64=no`. Například:

```
1 /ipv6 address
2 add address=2001:db8:ced1:1000::2/64 advertise=no eui-64=no interface=
    ether1
```

B.2 Nastavení VLAN

Nastavení můstku a konfigurace prvního portu (je na něm připojen router) bude následovná:

```
1 /interface bridge
2 add name=bridge1
3 /interface bridge port
4 add bridge=bridge1 interface=ether1 hw=yes
5 add bridge=bridge1 interface=ether2 hw=yes
6 add bridge=bridge1 interface=ether3 hw=yes pvid=1
7 add bridge=bridge1 interface=ether4 hw=yes pvid=1
8 add bridge=bridge1 interface=ether5 hw=yes pvid=2
9 /interface bridge vlan
```

```
10 add bridge=bridge1 tagged=ether1,ether2,bridge1 untagged=ether3,ether4
       vlan-ids=1
11 add bridge=bridge1 tagged=ether1,ether2 untagged=ether5 vlan-ids=2
12 add bridge=bridge1 tagged=ether1,ether2 vlan-ids=3
13 add bridge=bridge1 tagged=ether1,ether2 vlan-ids=4
14 add bridge=bridge1 tagged=ether1,ether2 vlan-ids=5
15 add bridge=bridge1 tagged=ether1,ether2 vlan-ids=6
16 add bridge=bridge1 tagged=ether1,ether2 vlan-ids=7
17 /interface vlan
18 add interface=bridge1 vlan-id=1 name=MGMT
19 /ip address
20 add address=192.168.1.252/24 interface=MGMT
21 /interface bridge
22 set bridge1 vlan-filtering=yes
```

Zde je však nutno zdůraznit, že tato konfigurace není univerzální pro všechny routery od firmy Mikrotik a funguje pouze s řadou **CRS3xx**. Pokud má v plánu čtenář této práce využít router jiný, nechť upraví můj kód dle dokumentace[30]. Pro testovací účely je možné místo parametru `hw=yes` použít `hw=no`, což konfiguraci udělá přenositelnou. Rámce budou switchované softwarově a je třeba počítat s masivní ztrátou výkonu.

B.3 Nastavení L2 firewallu

Nejprve je potřeba zapnout L2 firewall:

```
/interface bridge settings set use-ip-firewall=yes
```

Aby toto nastavení bylo aplikováno, je však třeba router restartovat. Přikročím k samotnému L2 firewallu, nemůžeme však zahodit všechny ICMPv6 zprávy, proto musíme daný paket postoupit L3 firewallu a to tak, že daný paket označíme (bereme všechny rámce, které nepřichází z důvěryhodného portu, než je v příkladu *ether1*):

```
1 /interface bridge filter
2 add action=mark-packet chain=forward in-interface=!ether1 ip-protocol=
      icmpv6 log=yes mac-protocol=ipv6 new-packet-mark=test_icmpv6
```

A pak stačí překročit k IP firewallu a takto označený paket dál filtrovat.

```
1 /ipv6 firewall filter
```

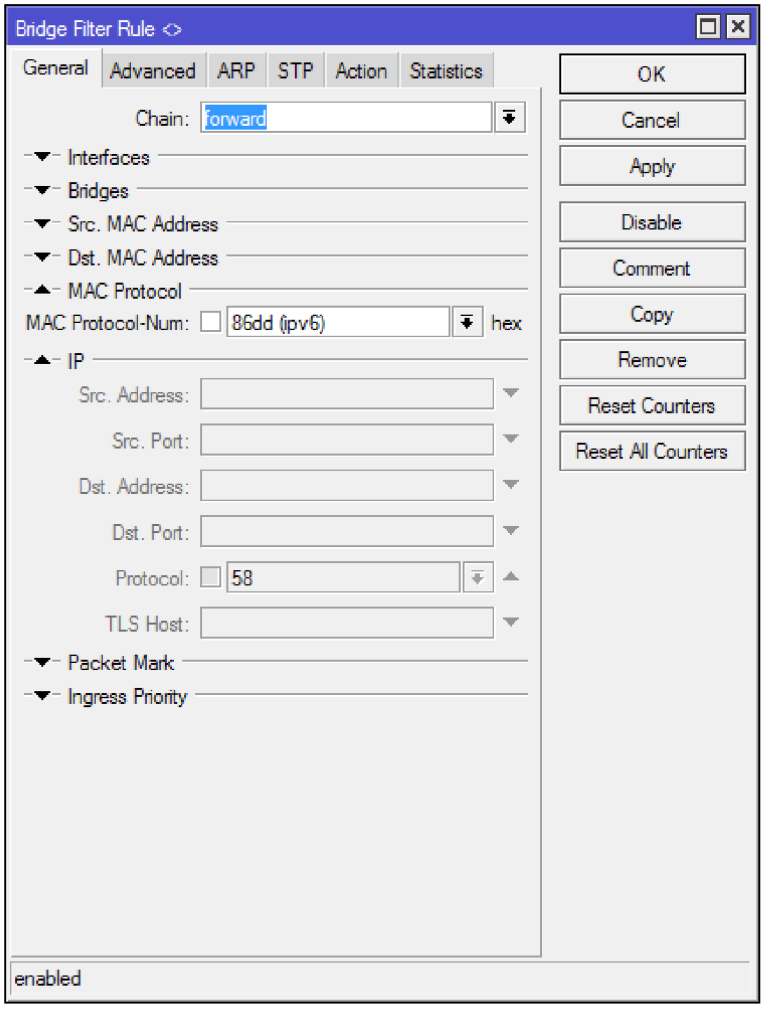
```
2 add action=drop chain=forward icmp-options=134:0-255 packet-mark=
    test_icmpv6 protocol=icmpv6
3 add action=drop chain=forward icmp-options=137:0-255 packet-mark=
    test_icmpv6 protocol=icmpv6
```

Není to však zadarmo. Filtrování rámců nefunguje pokud se používá HW switch, proto je potřeba na daných portech switchovat softwarově (v příkladu port *ether2*):

```
1 /interface bridge port
2 set hw=no where interface=ether2
```

Degradace výkonu je příliš masivní, aby toto nastavení bylo použito pro všechny porty. Toho lze docílit v kombinací s fyzickou bezpečností, kdy důvěryhodná zařízení jsou připojena do důvěryhodných portů a prázdné porty jsou vypnuty a případně i fyzicky blokovány. Toto nastavení lze použít i na WLAN, kde jsou přece jen jižní přenosové kapacity.

Kromě toho ještě existuje chyba v GUI, kvůli které nelze zadávat další parametry v L2 firewallu, pokud je typ rámce IPv6 paket (viz obrázek B.1), zadání příkazu do konzole je však úplně funkční.



Obr. B.1: Chyba v GUI v Mikrotiku

Zdroj: *vlastní tvorba*

C PŘÍLOHA: KONFIGURACE ZAŘÍZENÍ S OS WINDOWS

Pro stanice, které by měly mít statické ip adresy, vypínám randomizování identifikátoru následujícím příkazem (Windows 7, 10):

```
1 netsh interface ipv6 set global randomizeidentifiers=disabled store=
    active
2 netsh interface ipv6 set global randomizeidentifiers=disabled store=
    persistent
```

Pokud by tato stanice sloužila i jako server[1], je možné dočasné adresy úplně vypnout:

```
1 netsh interface ipv6 set privacy state=disabled store=active
2 netsh interface ipv6 set privacy state=disabled store=persistent
```

Pokud se stanice drží (chybné) adresy, lze vynutit získání nové adresy příkazem[18, s. 641]:

```
1 ipconfig release6
2 ipconfig renew6
```

Pokud tento postup nepomůže, je potřeba vypnout IPv6 protokol v nastavení síťové karty a potom ho zase zapnout. Obvyklým způsobem řešení problém na systému Windows, tj. restartem, toho nelze vždy docílit, protože získané IPv6 adresy si počítač drží i po restartu.

[1] momentálně se v organizaci nic takového nevyskytuje, uvádím jako příklad