UNIVERZITA PARDUBICE
Fakulta elektrotechniky a informatiky

Analýza využití protokolu TRILL v podnikové síti

Bc. Tomáš Kmoníček

Diplomová práce
2015

Univerzita Pardubice
Fakulta elektrotechniky a informatiky
Akademický rok: **2014/2015**

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Bc. Tomáš Kmoníček**
Osobní číslo:	**I13415**
Studijní program:	**N2646 Informační technologie**
Studijní obor:	**Informační technologie**
Název tématu:	**Analýza využití protokolu TRILL v podnikové síti**
Zadávající katedra:	**Katedra softwarových technologií**

Zásady pro vypracování:

Cílem práce je podrobně popsat principy protokolu TRILL a analyzovat možnosti jeho nasazení v podnikové síti. Autor podrobně představí principy fungování protokolu TRILL. Autor připraví simulaci fungování protokolu TRILL a jeho využití v podnikové síti. V praktické části autor vytvoří case study pro nasazení protokolu TRILL s důrazem na vysokou dostupnost požadovaných služeb.

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

HOODA, Sanjay K, Shyam KAPADIA a Padmanabhan KRISHNAN. Using TRILL and FabricPath: [designing massively scalable data centers (MSDC) with overlays]. xviii, 344 pages. ISBN 15-871-4393-3.
TISO, John, Shyam KAPADIA a Padmanabhan KRISHNAN. Designing Cisco network service architectures (ARCH): foundation learning guide. 3rd ed. Indianapolis: Cisco Press, c2012, xxxiv, 698 s. Foundation learning guide series. ISBN 978-1-58714-288-8.

Vedoucí diplomové práce: **Mgr. Josef Horálek, Ph.D.**
Katedra softwarových technologií

Datum zadání diplomové práce: **31. října 2014**
Termín odevzdání diplomové práce: **15. května 2015**

Univerzita Pardubice Česká republika

L.S.

prof. Ing. Simeon Karamazov, Dr.
děkan

prof. Ing. Antonín Kavička, Ph.D.
vedoucí katedry

V Pardubicích dne 15. listopadu 2014

Prohlášení autora

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

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

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 12. 5. 2015 Bc. Tomáš Kmoníček

Poděkování

Děkuji vedoucímu diplomové práce Mgr. Josefu Horálkovi, Ph.D. za veškeré rady, věcné připomínky a pomoc při zpracování této práce.

Anotace

V práci je podrobně popsána problematika přepínaných počítačových sítí. Jsou představeny protokoly Spanning Tree Protocol a TRILL, které v těchto sítích řídí přenos dat. Dalším výstupem práce je didaktický model, který umožňuje modelování přepínaných sítí s využitím těchto protokolů a názorně zobrazuje výhody protokolu TRILL pro nasazení v podnikových sítích.

Klíčová slova

počítačové sítě, přepínače, Spanning Tree Protocol, TRILL, IS-IS

Title

Analysis of Using TRILL Protocol in Enterprise Network.

Annotation

This work discusses detailed description of switched networks. The Spanning Tree Protocol and TRILL protocol (for managing data transfer in these networks) are presented. Another outcome of this work is didactic model that allows modeling of switched networks using these protocols and shows the benefits of TRILL protocol for deployment in enterprise networks.

Keywords

computer networks, switches, Spanning Tree Protocol, TRILL, IS-IS

Obsah

Seznam zkratek

ADSL	Asymmetric Digital Subscriber Line
ANSA	Automated Network Simulation and Analysis
API	Application Programming Interface
ARP	Address Resolution Protocol
ARQ	Automatic Repeat Request
ATM	Asynchronous Transfer Mode
b	Bit
B	Byte
CAM	Content Addressable Memory
CCITT	The International Telegraph and Telephone Consultative Committee
CDP	Cisco Discovery Protocol
CFI	Canonical Format Indicator
CRC32	Cyclic Redundancy Check 32
CSMA	Carrier Sense Multiple Access
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CSMA/CD	Carrier Sense Multiple Access with Collision Detection
DHCP	Dynamic Host Configuration Protocol
DMZ	Demilitarized Zone
DSAP	Destination Service Access Point
ECMP	Equal-Cost Multi-Path Routing
ESADI	End System Address Distribution Information
FDDI	Fiber Distributed Data Interface
FCS	Frame Check Sequence
FGL	Fine-Grained Labeling
GVRP	Generic VLAN Registration Protocol
HDLC	High-Level Data Link Control
ID	Identifikace
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protocol
IGP	Interior Gateway Protocols
IP	Internet Protocol
IPX	Internetwork Packet Exchange
ISDN	Integrated Services Digital Network
IS-IS	Intermediate System to Intermediate System
ISO	International Organization for Standardization
ISO/OSI	International Organization for Standardization / Open Systems Interconnection
ITU	International Telecommunication Union
LAN	Local Area Network

LAPD	Link Access Procedures - D Channel
LLC	Logical Link Control
LSAP	Link Service Access Point
MAC	Media Access Control
MST	Multiple Spanning Tree
MTP	Message Transfer Part
OSI	Open Systems Interconnection
OUI	Organizationally Unique Identifier
PCP	Priority Code Point
PVST	Per-VLAN Spanning Tree Protocol
PVSTP	Per-VLAN Spanning Tree Protocol Plus
RB	Root Bridge
RBridge	Routing Bridge
RFC	Request For Comments
RPF	Reverse Path Forwarding
RSTP	Rapid Spanning Tree Protocol
SDLC	Synchronous Data Link Control
SDN	Software-Defined Networking
SFD	Start of Frame Delimiter
SNAP	Subnetwork Access Protocol
SPB	Shortest Path Bridging
SPF	Shortest Path First
SSAP	Source Service Access Point
STP	Spanning Tree Protocol
TCA	Topology Change Acknowledgment
TCN	Topology Change Notification
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol / Internet Protocol
TRILL	Transparent Interconnection of Lots of Links
TTL	Time to Live
UDP	User Datagram Protocol
UML	Unified Modeling Language
VID	VLAN Identifier
VLAN	Virtual LAN
VMPS	VLAN Management Policy Server
VPN	Virtual Private Network
VTP	VLAN Trunking Protocol
Wi-Fi	Wireless LAN

Seznam obrázků

Úvod

Oblast informačních technologií v posledních letech zažívá jasný trend přesunu služeb do oblasti Internetového cloudu. V praxi to velmi často zahrnuje i přesun služeb ze serverů jednotlivých společností na servery datových center, která jsou díky tomu rozvíjena. Servery datových center jsou virtualizovány, aby mohly dynamicky reagovat na aktuální poptávku po službách. Stále vyšší požadavky jsou kladeny i na technologie počítačových sítí, které musí zajistit vysokou dostupnost služeb, musí pružně reagovat na změny v topologiích a virtuálním serverům umožnit například i přesun na jiný fyzický server bez výpadku služeb.

Technologie počítačových sítí se tak neustále rozvíjejí. Datová centra jsou dnes navrhována především jako velké ploché sítě postavené na přepínačích. V posledních 25 letech se pro řízení toku dat v přepínačích používá Spanning Tree Protocol (STP), který ale v dnešní době již nedokáže plně uspokojit extrémní požadavky datových center. V posledních letech proto bylo představeno několik nových technologií s cílem nahradit STP a vylepšit vlastnosti přepínačů. Jednou z těchto technologií je i protokol Transparent Interconnection of Lots of Links (TRILL), kterému je věnována tato práce.

Cílem teoretické části práce je podrobně popsat principy protokolu TRILL a analyzovat možnosti jeho nasazení v podnikové síti. Cílem praktické části práce původně bylo vytvoření simulátoru chování protokolu. Tento cíl byl ale v průběhu analýzy a návrhu programu po dohodě s vedoucím práce změněn. Z důvodu velké časové náročnosti vytvoření simulátoru byl cíl změněn na vytvoření didaktického modelu chování protokolu. Díky této změně je v rozsahu diplomové práce reálné implementovat všechny důležité vlastnosti protokolu TRILL. Dalším cílem práce je vytvoření případové studie nasazení protokolu v podnikové síti.

V první kapitole práce je stručně popsán současný stav problematiky. Jsou v ní uvedeny informace o vědeckých pracích, které se protokolu TRILL věnují. V další kapitole jsou uvedeny informace o problematice počítačových sítí, které slouží jako základ pro vysvětlení funkce přepínačů. Třetí kapitola podrobně popisuje spojovou vrstvu modelu Open Systems Interconnection (OSI) a fungování přepínačů. Čtvrtá kapitola je věnována možnostem řízení toku dat v přepínaných sítích. Podrobněji je popsán především STP a jeho nedostatky. V dalších podkapitolách jsou uvedeny možnosti a rozdíly technologií, které se rozvíjejí v posledních letech a mají za cíl STP nahradit.

V dalších dvou kapitolách jsou informace o protokolu TRILL. Pátá kapitola podrobně představuje všechny jeho principy a vlastnosti, které lze v počítačových sítích využít. Šestá kapitola analyzuje možnosti využití protokolu nejen v datových centrech ale i v podnikových sítích. Zároveň je nastíněn plán jeho dalšího vývoje.

Poslední kapitola popisuje výsledek praktické části práce. Tím je vytvořený interaktivní didaktický model, který názorně zobrazuje chování protokolů STP a TRILL v počítačových sítích. V kapitole jsou informace o programu, implementovaných možnostech modelu a zpracovaná případová studie nasazení protokolu TRILL v podnikové síti.

1 Současný stav problematiky protokolu TRILL

Po sedmiletém vývoji byl v roce 2011 vydán standard, který definuje protokol TRILL (RFC 6325, 2011). Cílem protokolu je významné vylepšení vlastností přepínaných počítačových sítí proti stávajícím řešením s využitím STP. Stejný cíl má i několik konkurenčních technologií.

Před popisem samotné problematiky protokolu TRILL, kterému jsou věnovány další kapitoly, je v této kapitole uveden popis několika odborných prací zabývajících se různými možnostmi využití tohoto protokolu. Vývoj protokolu byl přizpůsoben tomu, aby mohl být nejdříve nasazován v datových centrech, a další vlastnosti jsou přidávány postupně během aktualizací standardu (Matuška, 2010). Z tohoto důvodu se nejvíce prací zabývá právě využitím protokolu v datových centrech.

Popis potřeby změn proti současným řešením postaveným na STP v datových centrech a popis možností, které poskytuje TRILL a další nové technologie, nabízí zdroje (Scarfò, 2011; Dhanagopal, 2011). Velkou pozornost tyto zdroje věnují virtualizaci datových center, pro kterou TRILL nabízí podporu.

Využitím protokolu v datových centrech, ale z jiného pohledu, se zabývá i zdroj (Lu, 2013). Autoři se zaměřili na efektivitu doručení rámců pomocí protokolu TRILL a porovnali ji s efektivitou současných řešení se základem na STP. V příspěvku je popsáno provedené měření, které bylo zaměřeno na využití protokolů v datovém centru.

Vzdálenému přístupu uživatelů do datových center a propojení více datových center přes Internet se věnuje zdroj (Coudron, 2013). Popisuje možnosti využití více cest ve spojeních založených na různých vrstvách modelu OSI. Zdroj představuje řešení tohoto problému s využitím spolupráce více protokolů, kdy jedním z nich je TRILL.

Zdroj (Selga, 2013) se věnuje možnosti využití protokolu TRILL v oblasti inteligentních sítí. Ty umožňují regulaci výroby a distribuce elektrické energie v rozvodné síti. Zdroj popisuje možnosti, které TRILL může do inteligentních sítí přinést a porovnává je se současnými technologiemi, které se v inteligentních sítích používají.

Nestandardizovanou alternativu k protokolu TRILL a dalším protokolům, které jsou založeny na výběru nejlepších cest v síti, představuje zdroj (Ibanez, 2013). Místo výpočtů cest v síti je využita metoda založená na průzkumu cest pro doručení rámců, která má výhody v jednoduché implementaci, nízké latenci a adaptaci cest podle zatížení spojů. Výsledky experimentů zdroj porovnává s protokolem TRILL a dalšími technologiemi.

I přes poměrně velkou pozornost, kterou protokol TRILL vyvolává, se však jen málo lidí na světě zabývá možnostmi jeho modelování nebo simulace. Funkce protokolu je v zařízeních počítačových sítí implementována pomocí hardwaru i softwaru. Proto nelze využít produkční řešení pro modelování a simulace softwarovými nástroji, ale je nutné jednoúčelově implementovat softwarové řešení protokolu. To je poměrně složitý problém, který se v rámci projektu Automated Network Simulation and Analysis (ANSA) pokusil

řešit (Hrnčiřík, 2012). Autor ale v průběhu své práce změnil z důvodu náročnosti její cíle a zaměřil se pouze na implementaci protokolu Intermediate System to Intermediate System (IS-IS), která je pro funkci protokolu TRILL nezbytná. V projektu ANSA vyvíjí skupina autorů na Vysokém učení technickém v Brně komplexní simulační model počítačových sítí (Veselý, 2012; Networked and embedded systems research group, 2012). Jak je blíže popsáno v sedmé kapitole této práce, existuje i malé množství softwarových implementací protokolu TRILL, které ale nejsou pro simulace a modelování úplně vhodné.

2 Základní teorie počítačových sítí

Problematika počítačových sítí je poměrně složitá a zahrnuje mnoho aspektů, které se jí týkají. Počítačová síť v dnešní době slouží ke vzájemné komunikaci různých typů koncových zařízení, jako jsou například počítače, tablety, mobilní telefony, ale na druhé straně i servery – internetové, firemní nebo i servery v domácnostech (např. datová úložiště). Pokud je v jedné síti více zařízení, nelze se obejít bez dalšího typu zařízení – síťových prvků, které zajišťují členění sítě, přepínání a směrování dat mezi koncovými zařízeními.

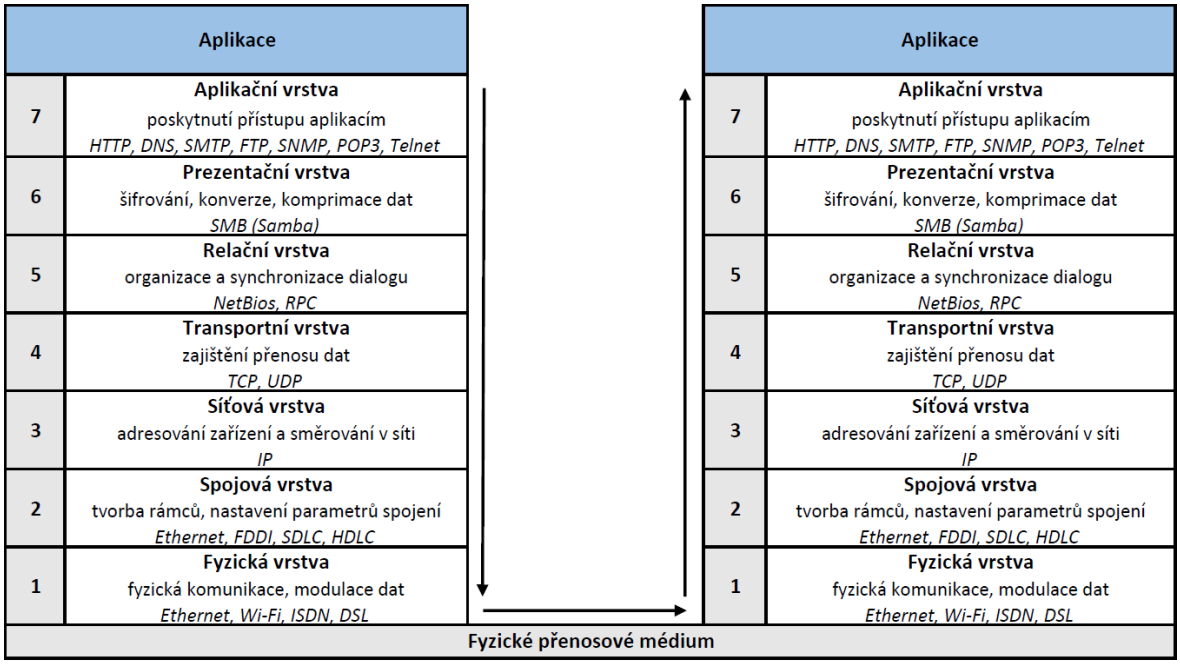
Přenos dat v počítačových sítích se řídí pravidly různých protokolů. Obecně je pro správný přenos dat potřeba součinnosti více protokolů, kdy každý z nich zajišťuje jinou část procesu přenosu. Spolupráce protokolů je v dnešní době založena především na architektuře TCP/IP (Transmission Control Protocol / Internet Protocol), která je využívána v Internetu. V této práci je však popsána a využita konkurenční architektura počítačových sítí založená na referenčním modelu ISO/OSI (International Organization for Standardization / Open Systems Interconnection). Tento model síťové protokoly rozděluje na více skupin a díky tomu podrobněji a přehledněji popisuje jejich odpovědnosti. Všechny popsané poznatky lze však snadno využít i v architektuře TCP/IP.

2.1 Referenční model ISO/OSI

Mezinárodní organizace pro standardizaci (International Organization for Standardization – ISO) vydala v roce 1984 normu ISO 7498 nazvanou OSI[1], která popisuje možný standard komunikace v počítačových sítích. Kompletně tuto normu následně v roce 1988 jako doporučení X.200 přejal i Mezinárodní poradní sbor pro telegrafii a telefonii (The International Telegraph and Telephone Consultative Committee – CCITT), který je dnes součástí Mezinárodní telekomunikační unie (International Telecommunication Union – ITU). V roce 1994 byly norma OSI i doporučení X.200 aktualizovány. (Telecommunication Standardization Sector of ITU, 1994)

Hlavní částí normy OSI je referenční model, který popisuje řešení komunikace v počítačové síti pomocí 7 na sobě nezávislých vrstev. Data, vytvořená v koncovém zařízení, jsou postupně zpracovávána od 7. k 1. vrstvě, která zajišťuje přenos do dalšího zařízení. V cílovém zařízení jsou přijatá data zpracována opačným postupem od 1. k 7. vrstvě. Model je znázorněn na následujícím obrázku (Obrázek 1).

[1] http://www.iso.org/iso/home/store/catalogue_ics/catalogue_detail_ics.htm?csnumber=14252 [cit. 2015-02-16]



Obrázek 1 – Referenční model ISO/OSI, zpracováno podle (Novell, 2007)

2.1.1 Fyzická vrstva

Fyzická vrstva je první vrstvou modelu a zajišťuje přenos dat mezi zařízeními na bitové úrovni, tato vrstva tedy nerozeznává žádnou strukturu dat. Model definuje možnosti fyzických spojů, elektrické a fyzikální vlastnosti zařízení. Fyzický spoj může být přímý (např. sériová linka mezi dvěma zařízeními), mnohobodový (např. Ethernet) nebo bezdrátový.

V dnešní době nejvíce rozšířenou technologií, která funguje na fyzické vrstvě, je Ethernet, definovaný normou 802.3 společnosti Institute of Electrical and Electronics Engineers (IEEE). Tato technologie definuje přenos dat v lokálních sítích (Local Area Network – LAN). Ethernet v průběhu historie prošel velkým vývojem, díky kterému je definováno mnoho verzí Ethernetu, které se liší použitými kabely, využitelnými rychlostmi přenosu, maximální délkou kabelů a dalšími vlastnostmi. Některé verze Ethernetu jsou mezi sebou kompatibilní (dvě zařízení s různými typy Ethernetu použijí verzi, se kterou obě dokáží pracovat). (IEEE Std 802.3-2012, 2012)

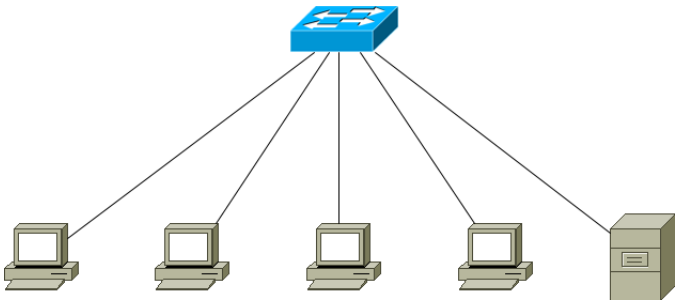
Jednou z alternativ Ethernetu je bezdrátová technologie Wireless LAN (Wi-Fi). Wi-Fi pro přenos dat používá jako médium vzduch. Do fyzické vrstvy modelu patří i Wi-Fi vysílače a přijímače.

2.1.2 Spojová vrstva

Druhá vrstva modelu je odpovědná za spojení mezi sousedními systémy. Oznamuje chyby a stará se o nastavení parametrů přenosu jako je rychlost přenosu nebo velikost přenášených bloků dat. Výstupem spojové vrstvy (který je předán fyzické vrstvě) je v dnešní době nejčastěji rámec – struktura dat, která obsahuje řídicí data a místo pro libovolná data připravená třetí (síťovou) vrstvou modelu OSI.

Ethernet i Wi-Fi kromě fyzické vrstvy zasahují i do spojové vrstvy modelu. Na spojové vrstvě definují skladbu rámců. Ty jsou podle těchto norem uspořádány do rámců s přesně strukturovanou hlavičkou (obsahující řídící informace) a různě velkým místem pro data vyšších vrstev modelu. Díky tomu tyto protokoly odpovídají definici modelu OSI a požadavku na nezávislost jeho vrstev.

Na spojové vrstvě pracují v počítačových sítích především dva druhy zařízení, které do vyšších vrstev modelu nezasahují: mosty a přepínače. Zařízení spojové vrstvy má obecně více fyzických připojení s dalšími zařízeními. Úkolem zařízení je pomocí využití (někdy i změn) řídicích informací v přijatém rámci poslat tento rámec dalšímu zařízení tak, aby pro rámec byla zajištěna cesta k cílovému zařízení. Cílovým zařízením na spojové vrstvě může být pouze zařízení připojené do stejné sítě jako zdrojové zařízení – obvykle to jsou zařízení v jedné domácnosti nebo jedné firemní budově. Na obrázku (Obrázek 2) je jednoduchá síť složená ze čtyř osobních počítačů a jednoho serveru propojená přepínačem. Pro přenos mezi zařízeními v různých počítačových sítích je potřeba navíc využít síťové vrstvy modelu.



Obrázek 2 – Propojení sítě zařízením druhé vrstvy modelu OSI (přepínačem)

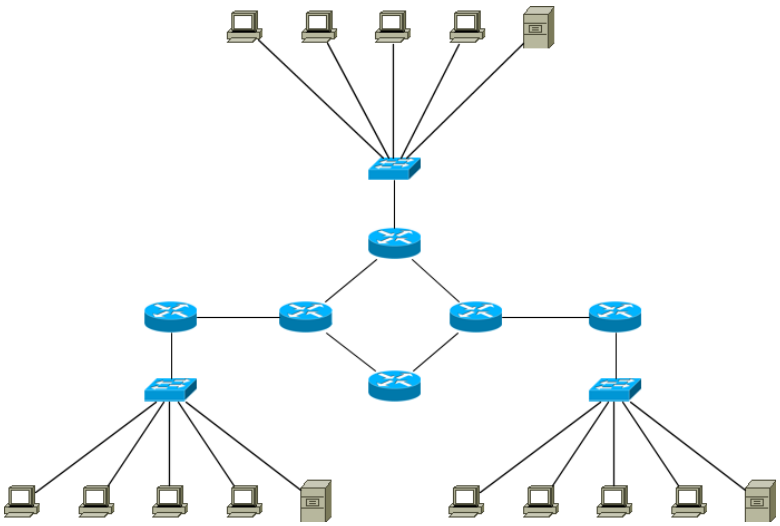
2.1.3 Síťová vrstva

Třetí vrstva modelu je odpovědná za směrování dat mezi různými systémy, které spolu nemusí sousedit a mohou fungovat na různých technologiích. Pokud je jako samostatný systém označena domácnost (ve které počítače mezi sebou komunikují prostřednictvím přepínače), síťová vrstva může zajistit komunikaci mezi různými domácnostmi, různými městy, státy, kontinenty. Síťová vrstva zajišťuje směrování dat v Internetu.

Nejpoužívanějším protokolem této vrstvy je Internet Protocol (IP). Podle něj má každé zařízení IP adresu a data jsou mezi nimi posílána v paketech. Paket je datová struktura podobná rámcům – obsahuje hlavičku s řídicími daty a variabilní místo pro data získaná z vyšších vrstev modelu OSI.

Zařízení, které na třetí vrstvě pracují a směrují data v síti, se nazývají směrovače. Obdobně jako zařízení spojové vrstvy směrují data podle řídicích informací v paketu. Cílovými zařízeními, mezi kterými je směrování provedeno, jsou takzvané brány – obvykle tuto funkci zastávají směrovače, které jsou nejblíže zařízením, které data odeslalo a které má data přijmout. Zasílání dat mezi bránou a těmito zařízeními je záležitostí druhé vrstvy.

Na obrázku (Obrázek 3) jsou zobrazeny tři malé sítě se směrovači, které zastávají funkci bran. Komunikace mezi sítěmi pak funguje díky jejich propojení přes další tři směrovače.



Obrázek 3 – Propojení tří sítí přes 6 směrovačů

Každé zařízení pracující na třetí vrstvě je zároveň i zařízením obou nižších vrstev, protože veškerá komunikace v síti probíhá na první vrstvě. Zařízení třetí vrstvy tedy nejdříve vytvoří paket, na druhé vrstvě ho zabalí do rámce a na první vrstvě pošle data sousednímu zařízení.

2.1.4 Transportní a další vrstvy

Pomocí prvních tří vrstev modelu lze vytvořit komunikační spojení mezi všemi zařízeními v počítačové síti, případně v Internetu. Další vrstvy tuto možnost komunikace využívají a dále rozvíjejí – přidávají další možnosti.

Transportní vrstva zajišťuje vyšším vrstvám určitý komfort pro práci s přenosem dat v síti a umožňuje doručení dat konkrétním aplikacím v koncových zařízeních. Nejrozšířenějšími protokoly jsou User Datagram Protocol (UDP) a Transmission Control Protocol (TCP). Využití UDP je rychlejší, ale není zajištěno doručení dat. Tento protokol je vhodný například pro přenos videa, zvuku apod., kde občasné vynechání dat nemá fatální důsledky. Naopak doručení dat pomocí TCP je spolehlivé. Protokol používá mechanismy pro detekci nedoručených dat, které následně automaticky posílá znovu. Nevýhodou je pomalejší přenos dat kvůli kontrole spolehlivého přenosu.

Pátá až sedmá vrstva odpovídají za synchronizaci přenosu (např. správné řazení dat), šifrování, komprimaci, různé konverze dat a také vytvářejí rozhraní pro přístup aplikacím ke komunikačnímu systému, který je tvořen z těchto vrstev. (Novell, 2007)

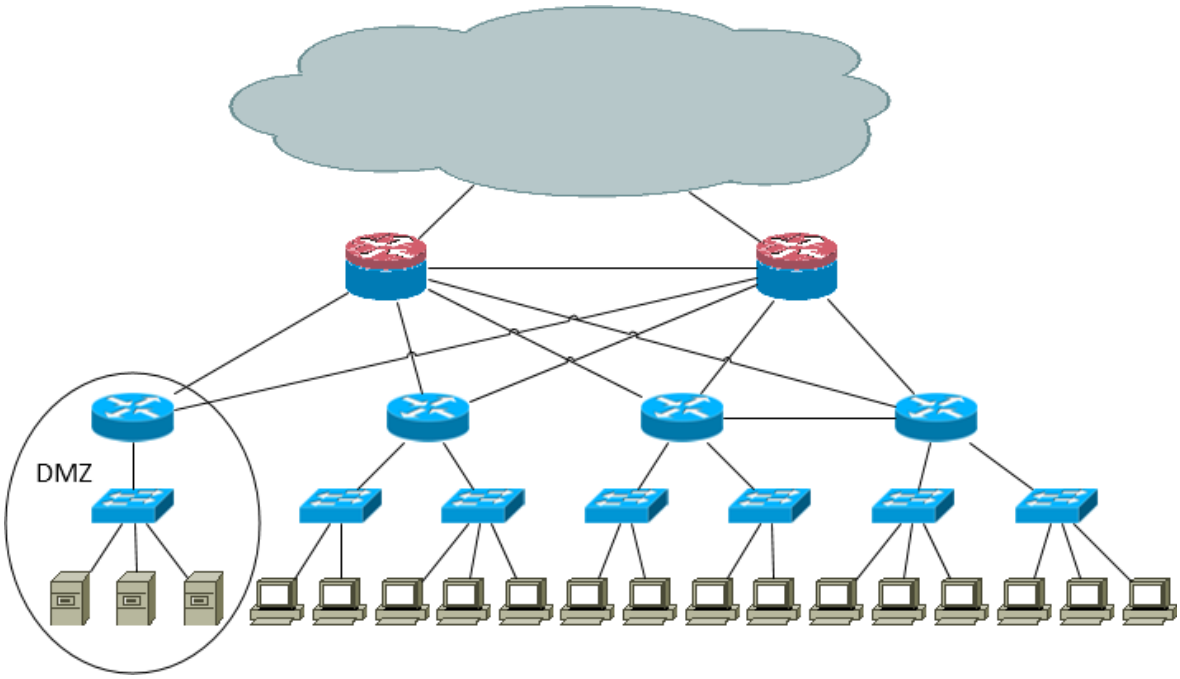
2.2 Podniková síť

Tato práce se zabývá analýzou v podnikových sítích, které mohou mít mnoho podob, ale obecně mají podnikové sítě určité společné znaky, podle (Microsoft, 2005):

- Připojení k Internetu, zabezpečené pomocí firewallu,

- servery jsou přístupné z Internetu i z vnitřní sítě a jsou umístěny v tzv. demilitarized zone (DMZ),
- podniková síť je složena z několika segmentů sítě LAN,
- do sítě je připojeno velké množství koncových zařízení přes různé přístupové technologie (Ethernet, Wi-Fi),
- požadavky na zabezpečení jsou odlišné pro různé skupiny uživatelů (management, obchodní oddělení, právní oddělení),
- směrování uvnitř podnikové sítě je zajištěno vnitřními směrovacími protokoly (Interior Gateway Protocols – IGP),
- do sítě je možné se vzdáleně připojit pomocí virtuální privátní sítě (Virtual Private Network – VPN),
- podniková síť může být hierarchicky uspořádaná do tří vrstev (páteřní, distribuční, přístupová).

Podle velikosti a požadavků na rozdělení sítě je potřeba navrhnout architekturu sítě, která obsahuje větší nebo menší množství aktivních síťových prvků pracujících na síťové nebo spojové vrstvě modelu OSI (směrovačů nebo přepínačů). Nejen podle počtu daných zařízení je potřeba zvolit vhodné protokoly, které budou zasílání dat v síti řídit. Na obrázku (Obrázek 4) je znázorněna třívrstvá architektura podnikové sítě s DMZ a dvěma připojeními k Internetu. Oba směrovače, které jsou do Internetu připojeny, zastávají i funkci firewallů.



Obrázek 4 – Jedna z možných architektur podnikových sítí

3 Spojová vrstva modelu ISO/OSI

V předchozí kapitole byly popsány základní informace o spojové vrstvě modelu OSI. Protože všechny další kapitoly práce se zabývají technologiemi, které na této vrstvě pracují, je potřeba spojovou vrstvu popsat podrobněji. Pro lepší přehled o funkcích vrstvy je potřeba ji rozdělit na dvě podvrstvy, kdy každá z nich odpovídá za část funkcí a ke správné činnosti spojové vrstvy je potřeba spolupráce obou podvrstev.

3.1 Podvrstva MAC

Podvrstva přístupu k médiu (Media Access Control – MAC) je v kontextu modelu OSI na nižší úrovni. Podvrstva je implementována hardwarem zvaným Media Access Controller, který je vyroben pro konkrétní použitou technologii (Token Ring, Ethernet, Wi-Fi apod.). Podvrstva působí jako rozhraní mezi fyzickou vrstvou modelu OSI a softwarovou částí spojové vrstvy a podle standardu (IEEE Std 802-2014, 2014, s. 14) poskytuje následující funkce:

- Vymezení a rozpoznání rámců,
- adresování cílových stanic (individuální i skupinové adresování),
- předávání adresních informací zdrojového zařízení,
- transparentní přenos dat přijatých z vyšší podvrstvy,
- kontrola vzniklých chyb, obvykle pomocí vložení a ověření kontrolních součtů,
- řízení přístupu k fyzickému médiu (dále popsané přístupové metody).

3.1.1 Adresování zařízení

Při komunikaci na dvoubodových spojích nejsou pro komunikaci na spojové vrstvě potřebné žádné adresy zdrojového nebo cílového zařízení. Toho využívá například protokol Message Transfer Part (MTP) ve všech svých verzích.

V sítích typu point-to-multipoint, kde komunikace probíhá vždy pouze mezi jedním řídicím a jedním z více podřízených zařízení, lze použít pouze jednu adresu podřízené stanice. Adresa řídicího zařízení není potřebná, protože v síti je pouze jedno řídicí zařízení. Tímto způsobem je navržen protokol pro synchronní řízení datového spoje (Synchronous Data Link Control – SDLC) a procedury z něj odvozené, např. High-Level Data Link Control (HDLC) a Link Access Procedures – D Channel (LAPD), které jsou používány v hojně rozšířených sítích Integrated Services Digital Network (ISDN). (Telecommunication Standardization Sector of ITU, 1997)

V sítích typu LAN je nutné vždy spolu s daty přenášet adresy zdrojového i cílového zařízení, protože v LAN sítích je obvykle připojeno větší množství rovnocenných zařízení a je potřeba jednoznačně identifikovat ta vzájemně komunikující.

3.1.2 Přístupové metody k fyzickému médiu

Jak bylo výše napsáno, další z funkcí podvrstvy MAC je řízení přístupu k fyzickému médiu. Pokud je médium pouze dvoubodové, není potřeba složité řešení přístupu. Pokud lze k médiu připojit více než dvě zařízení, vzniká problém, pokud více zařízení začne vysílat data ve stejném okamžiku. Tento problém lze řešit pomocí různých metod. V následujícím seznamu je uvedeno několik z nich podle (Tanenbaum, 2011):

- Token Ring (IEEE 802.5) je protokol, který zabraňuje vzniku kolizí tím, že v danou chvíli má právo vysílat pouze jedno zařízení v síti. Všechna zařízení jsou zapojena do kruhové topologie, data jsou v kruhu zasílána jen jedním směrem a pouze jedno zařízení má právo data vysílat (token). Právo vysílat je postupně předáváno dalším zařízením v kruhu.
- Token Bus (IEEE 802.4) má podobné vlastnosti jako Token Ring, ale zařízení jsou připojeny k jedné společné sběrnici, takže nejsou spojeny do kruhu. Kruhová topologie je vytvořena logicky a token je předáván podobně jako v případě protokolu Token Ring.
- Protokol naslouchání nosné při vícenásobném přístupu (Carrier Sense Multiple Access – CSMA) zařízením na společné sběrnici předepisuje, že před začátkem vysílání musí naslouchat, zda na médium nevysílá jiné zařízení. Zařízení může začít vysílat pouze ve chvíli, kdy nedetekuje žádný signál. Ve stejnou chvíli tak může začít vysílat více zařízení. Tím nastane kolize, kterou ale zařízení nedokážou rozpoznat, a odešlou na médium celý rámec. Současně vysílané rámce jsou porušené a zařízení musí individuálně řešit, jakým způsobem na to budou reagovat.
    - Wi-Fi a jiné bezdrátové sítě používají rozšířenou verzi CSMA s předcházením kolizí (CSMA with Collision Avoidance – CSMA/CA). Přesná implementace se může u různých technologií lišit. Například v LocalTalk zařízení nejdříve ohlašuje, že bude vysílat, ostatní zařízení s tím následně počítají a vysílat nebudou. Wi-Fi vysílání neohlašuje, ale každé zařízení začne vysílat po náhodném čase od ukončení vysílání ostatních zařízení. Tím je velmi zmenšena pravděpodobnost, že začne vysílat více zařízení najednou. I při uplatnění CSMA/CA kolize nastávají a mají stejné důsledky jako při použití CSMA, ale u CSMA/CA je výskyt kolizí mnohem menší.
    - Ethernet používá rozšířený protokol CSMA s detekcí kolizí (CSMA with Collision Detection – CSMA/CD), který dokáže okamžitě rozpoznat vzniklou kolizi v síti, po které zařízení ihned přestanou vysílat (nedokončí vysílání celého rámce). Následně zařízení čeká náhodnou dobu (aby dvě zařízení opět nezačala vysílat současně), a pokud je médium volné, začne vysílat rámec znovu.

3.1.3 Podvrstva MAC v technologii Ethernet

Každá technologie může funkce podvrstvy MAC implementovat různým způsobem. V případě Ethernetu jsou na této podvrstvě vytvořeny rámce, které mimo jiné obsahují i synchronizační data (preambuli, oddělení začátku rámce, mezirámcovou mezeru) a sekvenci dat pro ověření správnosti rámce (Frame Check Sequence – FCS). Poškozené rámce (se špatným FCS) jsou vyřazeny. V hlavičce Ethernetového rámce jsou vloženy také adresy, které jedinečně identifikují zařízení – MAC adresy zdrojového a cílového zařízení.

3.2 Podvrstva LLC

Podvrstva logického řízení linky (Logical Link Control – LLC) tvoří mezivrstvu MAC a síťové vrstvy modelu OSI. Je definována standardem IEEE 802.2. Hlavními úkoly LLC jsou podle (IEEE Std 802-2014, 2014):

- Multiplexní mechanismus, který umožňuje společnou existenci více protokolů (např. IP, Internetwork Packet Exchange – IPX, Decnet a Appletalk) v jedné počítačové síti,
- řízení toku dat (určuje maximální rychlost vysílání dat, aby sousední zařízení nebylo zahlceno),
- řízení chyb pomocí metody automatického opakování požadavků (Automatic Repeat Request – ARQ), která zajišťuje opakované zaslání rámců, u kterých sousedním zařízením nebylo potvrzeno doručení.

Funkce této podvrstvy je zajištěna přidáním hlavičky s řídicími informacemi k datům, které jsou zasílány vyšší vrstvou modelu OSI. Hlavička LLC, která je zobrazena na následujícím obrázku (Obrázek 5), obsahuje 3 pole (IEEE Std 802-2014, 2014):

- Přístupový bod zdrojové služby (Source Service Access Point – SSAP) je 8 bitové (b) pole, které určuje logickou adresu služby vyšší vrstvy, která vytvořila data (např. IP, IPX apod.). Logická adresa služby se označuje Link Service Access Point (LSAP).
- Přístupový bod cílové služby (Destination Service Access Point – DSAP) je 8b pole, které určuje LSAP cílové služby na vyšší vrstvě.
- Kontrolní pole Control může být velké 8b nebo 16b (podle definovaného formátu paketu) a slouží k přenosu dalších pomocných informací, například k řízení přenosu dat. V celé práci je používán vztah 1 byte (B) = 8b.

DSAP	SSAP	Control
1B	1B	1B nebo 2B

Obrázek 5 – Struktura LLC hlavičky

Pro řízení toku dat a řízení chyb existuje několik různých použitelných metod. V dnešní době ale tyto funkce z velké části přebraly protokoly transportní vrstvy (např. TCP), které provádí řízení mezi koncovými zařízeními (LLC provádí řízení mezi sousedními zařízeními).

Například Ethernet řízení toku dat a řízení chyb na spojové vrstvě vůbec nepodporuje (kromě CSMA/CD). Odpovědnost LLC se tímto zmenšuje na multiplex protokolů, který je zajištěn pomocí příznaku v řídicích datech zasílaného rámce.

3.3 Protokol SNAP

Protokol přístupu k podsíti (Subnetwork Access Protocol – SNAP) byl vytvořen pro podporu většího množství protokolů, než které podporuje LLC. SNAP rozšiřuje možnosti LLC tím, že k hlavičce přidává dalších 5B řídicích dat. V původní LLC hlavičce je pomocí hodnot LSAP označen protokol vyšší vrstvy jako SNAP (šestnáctkové hodnoty AA nebo AB). Hlavička je zobrazena na následujícím obrázku (Obrázek 6).

LLC hlavička			SNAP rozšíření	
DSAP	SSAP	Control	OUI	Protocol ID
1B	1B	1B nebo 2B	3B	2B

Obrázek 6 – Struktura LLC hlavičky s rozšířením SNAP

Dvě pole, které SNAP do hlavičky přidává, jsou definována (IEEE Std 802-2014, 2014):

- Jedinečný identifikátor organizace (Organizationally Unique Identifier – OUI), který určuje organizaci spravující protokol využitý ve vyšší vrstvě. OUI je přidělováno organizací IEEE. Pokud je šestnáctková hodnota OUI nastavena na 000000, je ve vyšší vrstvě použit protokol s přiřazenou hodnotou EtherType (velké množství běžně používaných protokolů, např. IP, IPX, AppleTalk, Wake on LAN).
- Identifikace (ID) protokolu Protocol ID je pole, které uchovává číslo určující protokol vyšší vrstvy. Může to být hodnota EtherType nebo hodnota, kterou si pro své protokoly určuje konkrétní organizace označená v poli OUI.

3.4 Přehled nejpoužívanějších technologií na spojové vrstvě modelu OSI

Na druhé vrstvě modelu je používáno velké množství různých protokolů a mnoho firem využívá možnosti vytváření vlastních protokolů (např. pro zajištění aktualizace firmware svých zařízení). V následujícím přehledu je tedy pro přehled o různorodosti technologií na druhé vrstvě uvedeno jen několik z nich, které jsou všeobecně rozšířené nebo souvisí s tématem této práce:

- Cisco Discovery Protocol (CDP) je používaný pro zjištění informací o sousedních zařízeních.
- Asynchronous Transfer Mode (ATM) je technologie v 80. a 90. letech označována jako telefonie další generace, která dokáže přenášet data vyšších vrstev v malých i mezinárodních sítích. V mnoha případech je ATM využíváno technologií Asymmetric Digital Subscriber Line (ADSL), kterou velké množství domácností dodnes využívá pro připojení k Internetu.

- Token Ring a Fiber Distributed Data Interface (FDDI) jsou použitelné v LAN sítích, kde využívají kruhovou topologii zařízení.
- Frame Relay je technologie pro přenos dat v rozlehlých sítích.
- Ethernet, který je více popsán v další podkapitole.
- Wi-Fi umožňuje vytvoření bezdrátové sítě LAN.
- Také to jsou řídicí protokoly druhé vrstvy, které jsou popsány v další kapitole (Možnosti řízení přepínání rámců v počítačové síti) této práce.

3.5 Ethernet

Jak již bylo napsáno, každá z technologií implementuje požadavky modelu OSI jiným způsobem. Protože další kapitoly této práce předpokládají provoz počítačové sítě převážně pomocí technologie Ethernet, je potřeba vysvětlit základní vlastnosti implementace, jakými Ethernet řeší požadavky modelu OSI.

Ethernet je definovaný standardem IEEE 802.3 a jedná se o skupinu technologií fungující na první až třetí vrstvě modelu. Pro přenos dat je možné využít koaxiální kabely (první verze Ethernetu), kabely s kroucenou dvoulinkou nebo optické kabely. Pro každý druh kabelu byl postupným vývojem definován větší počet verzí Ethernetu, které se liší maximální rychlostí přenosu dat, maximální délkou kabelu a jeho přesnější specifikací (např. využití jednovidového nebo vícevidového optického kabelu).

Na spojové vrstvě si Ethernetová zařízení zasílají rámce. Formát rámce je přesně daný (i když ho různé verze mohou používat trochu odlišně) a díky tomu mohou všechna Ethernetová zařízení vyrobená po roce 1985 komunikovat v jedné společné síti s jediným omezením – některá zařízení nelze propojit přímo (pokud používají jiné přenosové médium nebo nepodporují stejné rychlosti přenosu), ale je ale nutné je propojit přes aktivní síťové prvky. Formát rámce je zobrazen na následujícím obrázku (Obrázek 7) a význam jednotlivých polí je vysvětlen dále. (IEEE Std 802.3-2012, 2012)

72B - 1526B								
8B		64B - 1518B						
Preambule	SFD	MAC cíle	MAC zdroje	Typ / délka	LLC+SNAP	Data a výplň	FCS	Mezera mezi rámci
7xbyte 10101010	1xbyte 10101011	6B	6B	2B	8B	38B - 1492B	4B	12B

Obrázek 7 – Struktura Ethernetového rámce

Ethernetový rámec obsahuje tato pole:

- Preambule slouží k synchronizaci hodin příjemce, nenese žádné informace.
- Označení začátku rámce (Start of Frame Delimiter – SFD) se liší od preambule v 1b.
- MAC adresa cíle identifikuje cílové zařízení rámce.
- MAC adresa zdroje identifikuje zařízení, které rámec vytvořilo.

- Typ/délka je pole, které má dvě různé funkce podle verze Ethernetu, viz dále.
- Hlavičky LLC a SNAP jsou součástí Ethernetové hlavičky pouze v některých případech a zkracují prostor pro data, viz níže.
- Data a výplň je pole obsahující data z vyšší vrstvy modelu OSI (např. IP paket). Minimální délka pole 46B je nutná kvůli správné detekci kolizí v rámci segmentu sítě. Pokud je dat méně než 46B, je pole do této velikosti doplněno výplní. V případě využití hlaviček LLC a SNAP je prostor pro data o 8B zmenšen a minimální délka pole je tak 38B. Je tak zajištěna minimální délka celého rámce 64B.
- FCS obsahuje kontrolní součet vypočítaný z polí od MAC cíle po data a výplň. Ve chvíli, kdy zařízení přijme rámec, spočítá si podle dat hodnotu FCS. Pokud se tato hodnota neshoduje s hodnotou FCS v přijatém rámci, je rámec vyřazen. Výpočet FCS je v Ethernetu proveden vždy algoritmem Cyclic Redundancy Check 32 (CRC32).
- Mezera mezi rámci je čas odpovídající době přenosu 12B dat. Během této doby nemůže vysílat žádné zařízení v síti. Mezera mezi vysíláním je nutná, aby zařízení mohlo zpracovat přijímaný rámec, poté vyčistilo svou paměť a připravilo se na příjem dalšího rámce. (IEEE Std 802.3-2012, 2012, s. 31)

MAC adresa je 48b celosvětově unikátní číslo, které se zařízením přiřazuje při jejich výrobě. První polovina adresy je číslo OUI, které identifikuje výrobce zařízení, který následně určuje druhou polovinu adresy. Systém MAC adres byl poprvé definován v původním standardu Ethernetu od firmy Xerox. Tuto myšlenku následně přejaly i další technologie a MAC adresy tak využívají např. Wi-Fi, Bluetooth, Token Ring, ATM a další.

Pole Typ/délka v Ethernetovém rámci má dva možné významy. Standard Ethernet v2 (v dnešní době využívanější) v tomto poli uchovává kód protokolu vyšší vrstvy EtherType. Ethernet podle standardu 802.3 přenáší informaci o velikosti dat (v poli Data a výplň), pokud je hodnota menší nebo rovna 1500. V případě, že hodnota je alespoň 1536, informace obsahuje také typ protokolu vyšší vrstvy. Oba standardy jsou dnes součástí IEEE 802.3x a jsou spolu kompatibilní.

Ethernet, díky možnosti přenášet hodnotu EtherType v poli původně určeném pro uchování délky rámce, jako jediný ze skupiny protokolů IEEE 802 nemusí používat hlavičku LLC. V případě, že je v tomto poli přenášena informace o délce rámce, musí být hlavička LLC a případně SNAP vložena do pole pro data. Tím je zmenšen prostor pro data (např. IP paket) o 8B na 1492B. V dnešní době se tedy pro EtherType protokoly (např. IP) obvykle využívá Ethernet v2 a pro ostatní protokoly vyšší vrstvy je potřebné využít Ethernet s hlavičkami LLC a SNAP. V případě přenosu dat přes více segmentů sítě tím může docházet k situacím, kdy je potřeba 1500B dat zaslat rámcem, který povoluje maximálně 1492B dat. V tomto případě musí zařízení síťové vrstvy (např. směrovač) rozdělit data a odeslat je ve 2 rámcích. Další komplikace ale může nastat, pokud data rozdělit nelze (např. protokol TCP nastavuje v hlavičce paketu, že jeho obsah nemůže být rozdělen). Směrovač tak nemůže

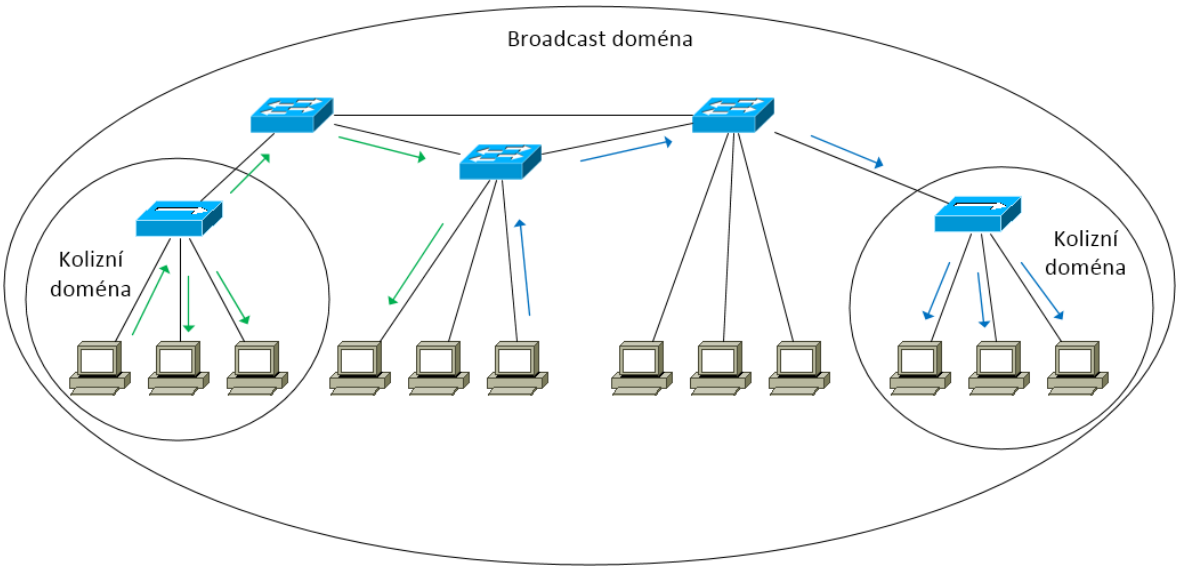
menší rámec vytvořit, vyřadí ho a zdrojovému zařízení pošle požadavek, aby data zaslal znovu v menších dávkách. (IEEE Std 802.3-2012, 2012)

Další možností, kterou Ethernet nabízí pro přenos dat, je využití Jumbo Frame. Jedná se o rámec, který může místo původních 1500B dat přenášet až 9000B a tím umožňuje zvýšit propustnost sítě cca o 4% (s jednou hlavičkou rámce je přeneseno větší množství užitečných dat). Jumbo Frame má stejnou strukturu jako rámec Ethernet v2, jen pole pro data je delší. V síti, kde některá zařízení tuto možnost nepodporují, nastává k časté fragmentaci rámců a propustnost sítě se tak může proti použití standardní velikosti rámců i snížit. Dalším krokem evoluce je možnost využití Super Jumbo Frame, u kterého není velikost dat omezena, ale u zařízení zatím vznikají problémy se sjednocením velikosti rámce v celé síti a se zpracováním těchto rámců z důvodu nedostatečně dimenzovaných pamětí, které zpracovávají aktuální rámec. (Tanenbaum, 2011)

Přístupová metoda Ethernetu ke společnému médiu CSMA/CD již byla popsána. Díky této metodě je po kolizi možné zaslat nedoručený rámec znovu. Je to ale jediný případ, kdy Ethernet řeší spolehlivost přenosu dat. V ostatních případech je pouze kontrolována správnost dat pomocí FCS a v případě zjištění porušení dat je daný rámec vyřazen. Stejně se síťová zařízení mohou chovat, když jsou zahlcená a nestíhají provoz sítě zpracovat. Pokud je nutné, aby byla cílovému zařízení doručena všechna data, která zdrojové zařízení odeslalo, je nutné kontrolovat doručení dat pomocí protokolů vyšších vrstev (konkrétně transportní vrstvy), např. TCP. Ethernet sám o sobě není spolehlivý.

3.6 Přepínače

Na spojové vrstvě modelu OSI pracuje v Ethernetu několik různých druhů zařízení. Nejvíce funkcí nabízí přepínače, které tak jsou nejpoužívanější. Tím, že zařízení pracuje na spojové vrstvě, je myšleno, že nepracují na vyšších vrstvách, a tak data vyšších vrstev nemění a mají za úkol je v původní podobě přenést mezi zařízeními, které na vyšších vrstvách pracují.



Obrázek 8 – Funkce přepínačů v síti

Hlavním úkolem přepínačů (Obrázek 8) je doručení rámce v jednom segmentu sítě. Na rozdíl od rozbočovačů (na obrázku vlevo a vpravo) nerozesílají rámce všem sousedním zařízením, ale pro rámec vybírají pouze jednu cestu. Tím segment sítě rozdělují na více menších kolizních domén (oblast, kde může dojít ke kolizi, když více zařízení začne současně vysílat) a ke kolizím nedochází tak často. Přepínače mají ale i další funkce, na podvrstvě MAC to podle (IEEE Std 802-2014, 2014) jsou:

- Zajištění přístupové metody k médiu (CSMA/CD).
- Fyzické adresování (pomocí vložení MAC adres do hlavičky rámce).
- Přepínání rámců (jak bylo popsáno v předchozím odstavci), které navíc může zahrnovat filtrování rámců (např. pomocí zakázaných MAC adres) a řízení přepínání rámců, které je popsáno v dalších kapitolách.
- Možnost zařazení rámce do fronty, když ho nelze odeslat ihned.
- Vytvoření virtuálních sítí LAN (Virtual LAN – VLAN).
- Některé přepínače kromě obvyklé metody přepínání (Store and forward) podporují na podvrstvě MAC i rychlejší metody přepínání, kdy ale není kontrolován FCS (metody Cut-through a Fragment-free).

Ve většině případů přepínač rámce žádným způsobem neupravuje, proto kromě původní metody přepínání (Store and forward), podle které přepínač přijme celý rámec, uloží si ho do paměti a po kontrole FCS začne vysílat rámec výchozím portem, byly vyvinuty další dvě metody. Cut-through umožňuje vyslat rámec co nejdříve – přepínač po přijetí prvních 14B rámce zná cílovou MAC adresu a v tuto chvíli může začít rámec vysílat. Tímto způsobem jsou ale přeposílány i chybné a neúplné rámce. Jedna z příčin přeposlání neúplného rámce může nastat ve chvíli, kdy se rámec v Ethernetu na příchozím portu přepínače dostane do kolize s jiným rámcem a zdrojové zařízení ho přestane po zjištění kolize vysílat. Jistota, že se rámec v Ethernetu nedostal do kolize, nastává až ve chvíli, kdy je z něj přijato 64B. Tuto vlastnost využívá metoda Fragment-free, pomocí které začne přepínač přeposílat rámec po přijetí právě prvních 64B dat. Odstraňuje se tak zasílání rámců, které se dostaly do kolize, ale stále je možné přeposílání poškozených rámců, protože není kontrolován FCS. Ten je kontrolován pouze při využití metody Store and forward.

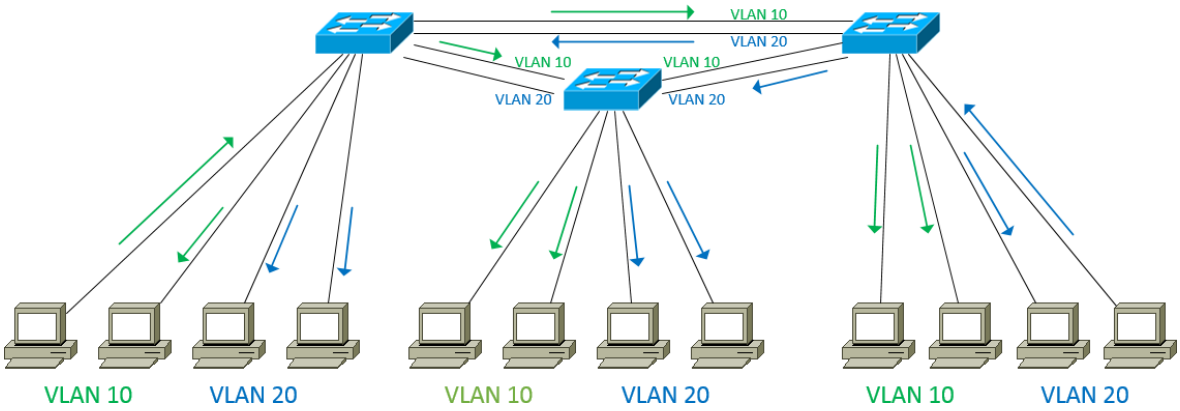
Aby přepínač mohl odesílat rámce správnou cestou k cílovému zařízení, je potřeba, aby znal základní informace o umístění cílových zařízení. Proto si přepínač vytváří tabulku (Content Addressable Memory – CAM), do které si ke každé MAC adrese zařízení ukládá port, přes který je možné se zařízením komunikovat. Přepínač po připojení do sítě má CAM tabulku prázdnou a začíná ji naplňovat po přijetí prvních rámců. Z každého přijatého rámce si do CAM tabulky uloží adresu zdrojového zařízení a port, přes který byl rámec přijat. Pokud je poté naopak zaslán rámec adresovaný pro toto zařízení, přepínač již k němu zná cestu. Pokud ale v CAM tabulce zatím není záznam o cílovém zařízení rámce, přepínač rozesílá rámec na všechny porty (kromě příchozího), aby rámec mohl být k cílovému zařízení doručen. Tímto způsobem se vždy chovají síťové rozbočovače a pro přepínače

je to typické chování po zapnutí. Je nutné dodat, že v CAM tabulce může být k jednomu portu přiřazeno více MAC adres. Přes port nemusí být dostupné jen jedno koncové zařízení, ale může k němu být připojen např. další přepínač nebo rozbočovač. V CAM tabulce jsou pak uloženy adresy všech koncových zařízení v podsíti (např. adresy všech počítačů), které jsou přes tento port dostupné. (Tanenbaum, 2011)

Kromě přenosu rámců s jedinečně danou cílovou adresou musí přepínače umět rozesílat všesměrové rámce (broadcast), které jsou účelně rozeslány všem zařízením v podsíti. Mají jedinečnou zdrojovou adresu a speciální formát cílové MAC adresy (FF:FF:FF:FF:FF:FF). Tyto rámce jsou již z principu postupného plnění CAM tabulky přepínačem rozesílány na všechny porty, protože přepínač nikdy neobdrží rámec ze všesměrové adresy, a tak si ji do CAM tabulky neuloží. Přepínače tedy pro všesměrové rámce nemusí mít implementované žádné speciální funkce. Na stejném principu funguje na spojové vrstvě i zasílání skupinových rámců. Např. skupina zařízení podporujících CDP se adresuje pomocí cílové MAC adresy 01:00:0C:CC:CC:CC a přepínače každý přijatý rámec s touto adresou rozesílají všemi porty (protože ji nemají v CAM tabulce). Rozhodnutí, jak bude rámec zpracován (nebo nezpracován), je až na koncovém zařízení. Přepínače mají za úkol rámce pouze doručit.

3.7 Virtuální sítě VLAN

Jednou z pokročilých vlastností přepínačů je možnost vytvoření VLAN. Jejich využití přináší do podnikových sítí několik výhod. Při využití stejného množství síťových prvků a kabeláže je to v první řadě větší bezpečnost a dále snížení počtu všesměrových rámců v podsíti, jednodušší změny síťové infrastruktury a oddělení speciálního provozu. (Bouška, 2007)



Obrázek 9 – Provoz ve dvou VLAN (všesměrové rámce)

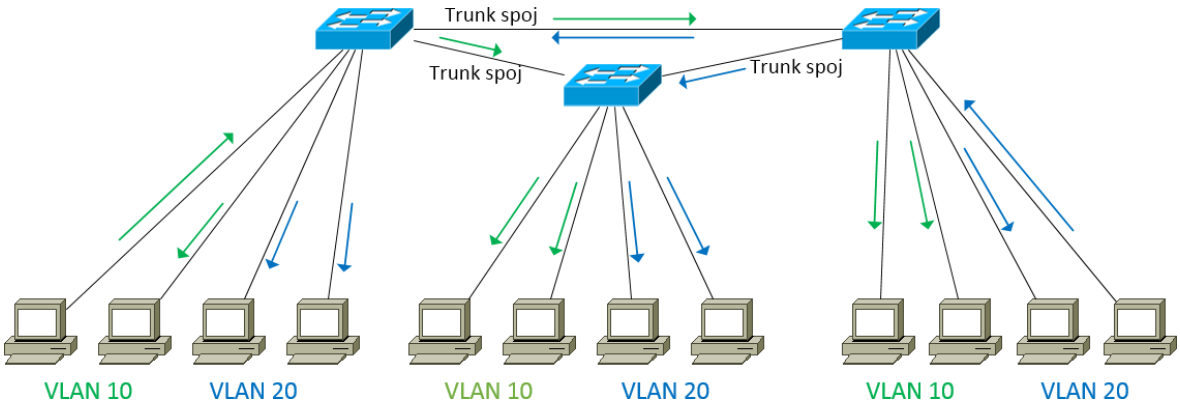
VLAN je jedna podsíť, oddělená od ostatních. Hlavní myšlenkou VLAN je vytvoření více oddělených VLAN na přepínači, kdy mezi sebou mohou komunikovat pouze zařízení ze stejné VLAN (Obrázek 9). K jednomu přepínači tak lze připojit např. dvě nebo více různých oddělení podniku, které mezi sebou nemohou komunikovat napřímo pomocí spojové vrstvy modelu. Tím je dosaženo všech výhod, které byly popsány v předchozím odstavci. Stejných možností lze dosáhnout i bez využití VLAN, ale pouze

s několikanásobným počtem síťových prvků. Proto je využití této technologie velmi užitečné hlavně ve velkých podnicích a datových centrech. Pokud je potřeba, aby mezi sebou komunikovala zařízení z různých VLAN, je to možné pomocí směrování na síťové vrstvě modelu OSI, která ale nabízí mnohem více možností pro omezení a sledování provozu. Provoz tak lze daleko lépe kontrolovat.

Samotné přiřazení koncového zařízení do konkrétní VLAN je možné několika způsoby (Hucaby, 2010):

- Podle portu přepínače, do kterého je zařízení připojeno.
- Podle MAC adresy koncového zařízení, která je uložena v databázi serveru pro politické řízení VLAN (VLAN Management Policy Server – VMPS), který může spravovat MAC adresy celé podnikové sítě a uživatel se tak může připojovat k různým přepínačům v síti (např. se svým notebookem) a vždy bude mít přístup do své VLAN.
- Podle autentizace (přihlašovacích údajů), které jsou protokolem IEEE 802.1x ověřeny na RADIUS serveru. Přes jedno koncové zařízení se tak může v různou dobu připojovat více různých uživatelů do různých VLAN podle toho, jaká mají definována přístupová práva na základě svých přihlašovacích údajů.

V podnikové síti ale velmi často nelze zařízení jedné VLAN propojit na jednom fyzickém místě (k jednomu přepínači), proto je potřeba propojení jedné VLAN mezi více přepínači. To lze udělat dvěma způsoby – pro každou VLAN vytvořit fyzický spoj (kabel) mezi přepínači a nastavit mu příslušnost k dané VLAN nebo po jednom fyzickém spoji přenášet data z různých VLAN pomocí spoje zvaného Trunk (Obrázek 10). Ethernetový rámec v základním formátu ale nedovoluje vložení dalších informací, aby přepínač rozeznal, do které VLAN patří přijatý rámec. Tento problém je v dnešní době řešen dvěma možnostmi – přes protokol Inter-Switch Link (ISL) a standard IEEE 802.1Q.



Obrázek 10 – Vytvoření Trunk spojů mezi přepínači

3.7.1 Protokol ISL

ISL je proprietární protokol společnosti Cisco, takže je vyvinut pouze pro zařízení tohoto výrobce. Na Trunk spojích přidává k Ethernetovým rámcům další informace tím,

že před rámec vloží 26B hlavičku a na konec rámce přidává 4B FCS (Obrázek 11). Rámec se tak může zvětšit až na 1548B. S nestandardní délkou obvykle problémy nebývají, protože ISL Trunk spoj je vytvářen pouze mezi Cisco zařízeními, které jsou k tomuto účelu vyrobeny. Přepínač, který přijme ISL rámec, přidané informace smaže a cílovému zařízení doručí původní Ethernetový rámec.

ISL hlavička	Ethernetový rámec	FCS
26B	64B - 1518B	4B

Obrázek 11 – Struktura ISL rámce

Nová hlavička obsahuje kromě 15b VLAN identifikátoru (VLAN Identifier - VID) např. 40b skupinovou cílovou adresu, 48b zdrojovou MAC adresu, 4b určující prioritu rámce, 4b označující typ rámce (kromě Ethernetu to může být např. Token Ring a další), LLC a SNAP hlavičku a několik dalších řídicích polí. (Hucaby, 2010)

3.7.2 Standard 802.1Q

Na rozdíl od ISL je standard IEEE 802.1Q rozšířen a podporován mezi všemi výrobci pokročilých přepínačů. Tento standard také upravuje Ethernetovou hlavičku, ale jiným způsobem než ISL. Přímo do hlavičky vkládá 32b dat (Obrázek 12), které jsou složeny z 16b hodnoty EtherType definované pro 802.1Q (0x8100), 3b pro definování priority (Priority Code Point – PCP), 1b pro rozlišení technologií Ethernet nebo Token Ring (Canonical Format Indicator – CFI) a 12b VID. Ethernetový rámec se přidáním těchto dat může zvětšit na 1522B, což může v síti způsobit problémy. Pokud některý z přepínačů podporuje pouze původní standard Ethernet v2, může rámec označit kvůli větší délce jako chybný. (Hucaby, 2010)

				802.1Q					
Preambule	SFD	MAC cíle	MAC zdroje	EtherType	PCP+CFI+VID	Typ / délka	Data a výplň	FCS	Mezera mezi rámci
7xbyte 10101010	1xbyte 10101011	6B	6B	2B - 0x8100	2B	2B	42B - 1500B	4B	12B

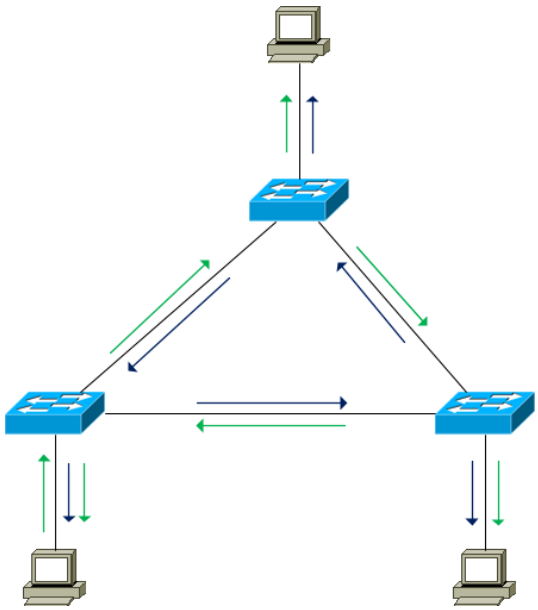
Obrázek 12 – Struktura Ethernetového rámce s 802.1Q

3.7.3 Protokoly pro správu VLAN v síti

Pro úplnost by měly být zmíněny i protokoly VLAN Trunking Protocol (VTP) a Generic VLAN Registration Protocol (GVRP). Oba mají za cíl usnadnit správu VLAN v celé síti. Přepínače si díky těmto protokolům vyměňují informace o svých VLAN a mohou si tak do své pamětí uložit informace o všech VLAN v síti. Konfigurace VLAN může být díky těmto protokolům provedena pouze na jednom přepínači, ostatní přepínače si od něj stejné informace převezmou a koncová zařízení se mohou připojit k jakémukoli přepínači. VTP je proprietární protokol společnosti Cisco, nelze ho tedy použít v síti s přepínači od různých výrobců. GVRP je protokol vyvinutý asociací IEEE a díky tomu je určen pro zařízení všech výrobců. (Hucaby, 2010)

4 Možnosti řízení přepínání rámců v počítačové síti

Již bylo vysvětleno, jakým způsobem přepínače zasílají rámce směrem k cílovému zařízení. K základnímu přehledu o topologii podsítě slouží přepínači CAM tabulka. Princip ukládání MAC adres do tabulky je velmi jednoduchý, ale nemůže zabránit vzniku některých problémů. Základní problém je zobrazen na následujícím obrázku (Obrázek 13). V případě zapojení přepínačů s prázdnými CAM tabulkami do smyček mohou být rámce v podsíti rozesílány stále dokola ve všech směrech. Protože na rozdíl od síťové vrstvy nedisponuje spojová vrstva mechanismem, který by vyřadil rámce kolující v podsíti bez dosažení cíle, je nutné zabránit vzniku tohoto problému preventivním opatřením.



Obrázek 13 – Zacyklení rozeslaného rámce v síti

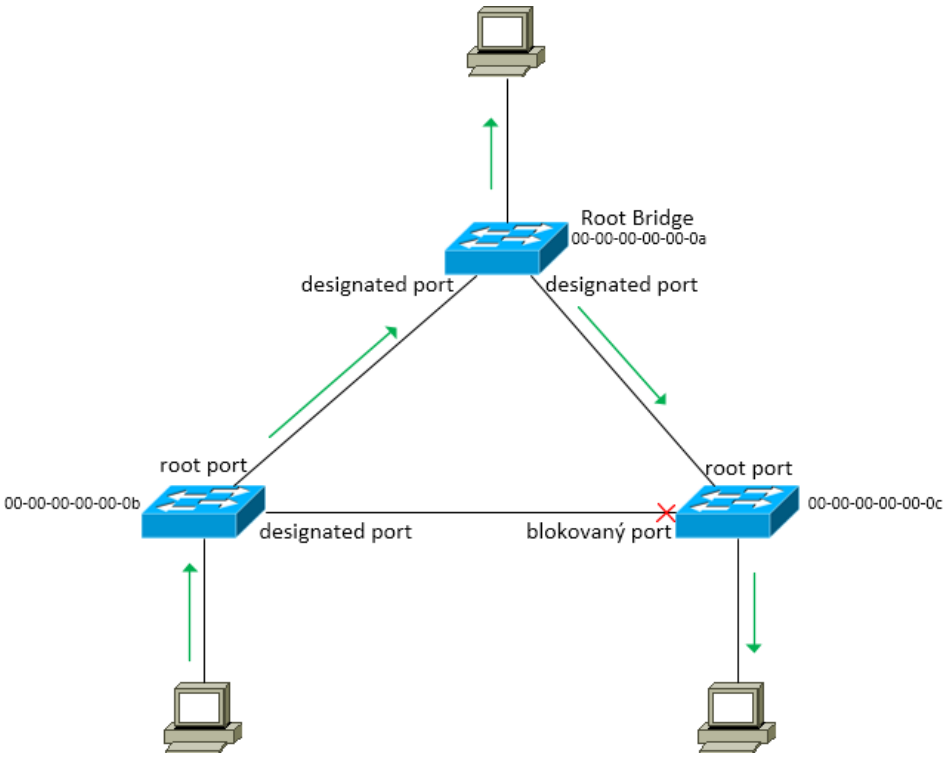
Nejjednodušším preventivním opatřením je samozřejmě nezapojování přepínačů do smyček, protože i bez nich může podsíť plnohodnotně fungovat. To ale často není vyhovující ve větších sítích, kde zapojení přepínačů do smyček plní důležité funkce redundance a rozložení zátěže. Díky redundanci může podsíť fungovat bez velkého výpadku i v případě poruchy některého z přepínačů. Bez rozložení zátěže by větší podsíť mohla mít problémy s přetížením, a tím by se mohla stát nespolehlivou. Z těchto důvodů byly vytvořeny mechanismy, které stále využívají CAM tabulky přepínačů, ale umožňují i pokročilejší metody řízení pro přepínání rámců.

4.1 Spanning Tree Protocol

Protokol kostry grafu (Spanning Tree Protocol – STP) je nejdéle používanou technologií pro zabránění vzniku smyček. STP je standardizován jako IEEE 802.1D, první verze byla zveřejněna roku 1990 a poslední verze standardu z roku 2004 je dnes v praxi postupně nahrazována novým standardem 802.1aq (protokol Shortest Path Bridging – SPB), který byl publikován v roce 2012, nebo jinými alternativními protokoly. Protože většina podnikových

sítí využívala (případně stále využívá) STP, je v této práci důležité nastínit jeho základní vlastnosti a principy chování.

Základní verze STP vybírá mezi přepínači vždy pouze jednu možnou cestu a alternativní cesty blokuje (Obrázek 14). Selekce aktivních cest funguje na principu výběru jednoho hlavního přepínače (Root Bridge – RB), ke kterému jsou aktivovány všechny nejlepší cesty, zatímco ostatní cesty jsou zablokovány. RB obvykle nemá žádný blokovaný port a je kořenem ve stromové topologii. V podsíti je časté, že většina provozu prochází přes RB, který je vybírán na základě konfigurace nebo nižší hodnoty MAC adresy. Po výběru RB jsou k němu z každého přepínače aktivovány nejlepší cesty vybrané na základě větších rychlostí spojů. Přepínač tak může mít porty v několika různých režimech: root port (směrem k RB), designated porty (aktivní) a blokované porty. Pokud je více přepínačů připojeno ke stejné LAN nebo jsou dva přepínače spojeny přímo mezi sebou, je vždy vybrán jeden přepínač, který bude svým portem (designated) naslouchat a přeposílat data z této LAN do dalších částí sítě. Ostatní přepínače musí mít připojené porty v blokovaném stavu, aby zasílaná data nebyla duplikována. (Hucaby, 2010, s. 130-134)



Obrázek 14 – Funkce STP

RB v podsíti pravidelně odesílá ostatním přepínačům se STP rámce nazvané Bridge Protocol Data Unit (BPDU), které jsou adresovány se skupinovou MAC adresou (01:80:C2:00:00:00). BPDU obsahuje hlavně informace o RB a cestě k němu. Každý přepínač po přijetí BPDU upraví informace o cestě k RB (vloží informace o vlastní nejlepší cestě k němu) a odešle BPDU do dalších částí podsítě. V souvislosti se zasíláním informačních zpráv má každý přepínač několik nastavených časovačů. Každý port přepínače může být v blokovaném stavu, nebo se přes stavy naslouchání a učení může stát aktivním

(pokud podle dostupných informací nemůže vzniknout smyčka v podsíti). Časovače se mimo jiné týkají i stavů naslouchání a učení, takže konvergence podsítě není okamžitá. V základním nastavení se vždy (i po změně topologie) zkonverguje maximálně do 52 sekund. Ve velkých podsítích, kde některý z přepínačů musí s RB komunikovat přes více než 5 jiných přepínačů, je doporučeno nastavit větší časové intervaly. Tím konvergence podsítě může trvat delší dobu. (Hucaby, 2010, s. 140-145)

Každý přepínač ve chvíli, kdy v síti zaznamená změnu topologie, zasílá směrem k RB speciální typ BPDU – Topology Change Notification (TCN), kterým o změně v síti informuje ostatní přepínače. Každý přepínač, který dostane TCN, musí po jeho zpracování zaslat zpět potvrzení Topology Change Acknowledgment (TCA). Poté, co RB přijme TCN, zašle RB všem přepínačům BPDU s nastaveným příznakem změny topologie. Po přijetí této informace si přepínače nastaví interval platnosti aktuálních informací v tabulce STP přepínačů na kratší čas, aby mohly na změnu topologie rychleji reagovat. Bez TCN by konvergence sítě trvala mnohem déle (přes 5 minut). (Hucaby, 2010)

4.1.1 Pokročilé typy STP

S postupným technologickým vývojem počítačových sítí bylo vytvořeno několik úprav STP, které částečně odstraňují jeho nevýhody (dlouhá doba konvergence sítě, kompletní zablokování některých spojů, otevření portů při nejasném stavu podsítě – vzniká možnost smyček). Nejpoužívanější z pokročilých typů jsou (Hucaby, 2010):

- Per-VLAN Spanning Tree (PVST) a PVST+ (někdy také PVSTP), které jsou proprietárními protokoly společnosti Cisco, pro každou VLAN vytvářejí jednu instanci STP a tím lze pro každou VLAN blokovat jiné spoje v podsíti. Žádný spoj podsítě tak nemusí být zcela bez provozu jako v případě STP.
- Rapid Spanning Tree Protocol (RSTP) definovaný normou 802.1w je zaměřen na rychlejší konvergenci sítě. Nový stav nazvaný vyřazování nahrazuje stavy portů zakázaný, blokování a naslouchání z protokolu STP. Zároveň snižuje časovače při procházení stavů, které ale ve většině případů ani nejsou použity a podsíť se může zkonvergovat během jedné sekundy (čas pro doručení rámců s informací o změně, pokud všechny přepínače využívají RSTP). V ostatních případech je maximální čas konvergence 6 sekund (vypršení časovače pro čekání na BPDU).
- Multiple Spanning Tree (MST) definovaný normou 802.1s využívá RSTP a přidává možnost vytvoření více instancí protokolu. Každá instance blokuje některé spoje podsítě, takže výsledný provoz v podsíti je podobný použití PVST. MST ale může v jedné instanci řídit přepínání paketů ve více VLAN zároveň a tím šetří výpočetní zdroje přepínačů.

S dalším rozvojem ale žádná z verzí STP nedokáže efektivně uspokojit všechny potřeby dnešního světa počítačových sítí, takže vzniklo několik nových technologií, které jsou navrženy s ohledem na stále narůstající požadavky dnešní doby. Nevyhovující je zejména

stromová topologie STP, stále příliš dlouhá konvergence sítě (i s využitím RSTP) a poměrně složitá konfigurace přepínačů.

4.2 Shortest Path Bridging

SPB je IEEE standard nahrazující STP. Byl vytvořen s cílem zjednodušit konfiguraci a umožnit vyvážený tok dat sítí (rovnoměrné využití spojů). Navíc nabízí vytvoření mnohem větších podsítí (až 16 milionů VLAN proti 4096 u STP) a mnohem rychlejší konvergenci sítě. Stromová topologie je změněna na mnohem lépe škálovatelnou topologii mesh. SPB na technologii Ethernet vytváří logické sítě pomocí zapouzdření rámců metodami MAC-in-MAC (802.1ah, dvojité zapouzdření do Ethernetového rámce) nebo Q-in-Q (802.1ad, dvojité označení VLAN). Pro výběr cest rámců v podsíti je využit protokol IS-IS, který využívá Dijkstrův algoritmus a původně byl vytvořen pro směrování paketů v síťové vrstvě modelu OSI. Vzhledem k tomu, že SPB využívá již existující technologie a nevyžaduje pro přepínače vytvořit specializovaný hardware, lze jej využít i na některých přepínačích, které byly vytvořeny před vytvořením standardu. (IEEE Standards Association, 2012)

4.3 Protokol TRILL

Popis protokolu TRILL je jedním z cílů této práce, proto je TRILL podrobně popsán v dalších kapitolách. Na tomto místě jsou pro porovnání s ostatními protokoly uvedeny jen jeho základní charakteristiky. TRILL je standard komise techniky Internetu (Internet Engineering Task Force – IETF), která se mimo jiné zabývá internetovými protokoly a architekturou TCP/IP. První definice protokolu byla vydána v roce 2011 v IETF žádosti o komentáře (Request For Comments – RFC) číslo 6325. (RFC 6325, 2011)

TRILL je nejvážnějším konkurentem SPB, protože výsledná funkce těchto protokolů je přes odlišnou implementaci velmi podobná. Již od vzniku obou standardů se mezi odborníky vedou dlouhé diskuse o tom, který z protokolů bude v praxi více nasazován a zda se některý z nich stane de facto jediným využívaným protokolem na spojové vrstvě stejně, jako se tomu stalo v případě STP.

TRILL je zčásti hardwarově implementovaný protokol, proto jej lze provozovat pouze na přepínačích, které mají specializovaný hardware. Stejně jako SPB využívá pro výběr cest protokol IS-IS a může pro určité případy (vysvětleno v dalších kapitolách) v podsíti fungovat se staršími přepínači, které TRILL nepodporují. Zároveň do spojové vrstvy přináší některé mechanismy, které jsou velmi známé a používané na síťové vrstvě, například ping a traceroute. Celkově je TRILL komplexnější než SPB a díky tomu nabízí větší odolnost proti smyčkám a také větší univerzalitu implementace (např. jen v části sítě). (Matuška, 2010)

4.4 Cisco FabricPath

Proprietární protokol FabricPath společnosti Cisco má velmi podobné cíle a možnosti využití jako protokol TRILL a k výpočtům také využívá protokol IS-IS. FabricPath

ale používá jinou hlavičku rámců, takže není s protokolem TRILL kompatibilní. Při využití přepínačů různých výrobců je tak na Cisco přepínačích potřeba vypnout FabricPath, aby byl využit standardní TRILL, který je na nich také implementován. Nevýhodou FabricPath proti protokolu TRILL je nutnost celistvosti FabricPath sítě – mezi dvěma FabricPath přepínači nesmí být přepínač, který tento protokol nepodporuje. Výhodou FabricPath je ale větší funkcionalita, protože nové možnosti protokolu jsou zaváděny podle vnitřních zvyklostí společnosti Cisco a nemusí procházet zdlouhavými schvalovacími procesy IETF. Ve výsledku by TRILL i FabricPath měly podporovat téměř totožnou funkcionalitu, ale FabricPath přináší nové funkce vždy o několik let dříve. V době psaní této práce například podporuje vytvoření více virtuálních topologií v jedné síti a technologii Cisco Virtual Port Channel. Pro TRILL jsou vyvíjena alternativní řešení, ale zatím existují pouze ve stádiu návrhů. (Hooda, 2014)

4.5 QFabric

Firma Juniper vyvinula vlastní technologii QFabric, která je zaměřena na rozsáhlé ploché podsítě a nabízí jejich velkou škálovatelnost. Všechny síťové prvky tvoří jeden logický přepínač, díky kterému je možná jednoduchá konfigurace a všechna koncová zařízení mezi sebou mohou komunikovat s velmi malou latencí. Jedna podsíť může obsahovat až 6 144 portů, které jsou logicky umístěny na jednom přepínači. Technologie QFabric používá tři druhy síťových prvků: uzly, spojovač a manažer. Uzly jsou přepínače s neměnnou konfigurací, které jsou připojeny ke koncovým zařízením, spojovač je velmi rychlé přenosové zařízení, které propojuje všechny uzly do topologie mesh. Manažer poskytuje kontrolní a servisní služby pro celou QFabric síť. (Juniper Networks, 2015)

4.6 Virtual Cluster Switching

Další proprietární technologií určenou ke správě plochých podsítí je Virtual Cluster Switching, kterou v roce 2010 představila společnost Brocade. Funguje na podobném principu jako QFabric, všechny přepínače v podsíti se logicky chovají jako jeden. Po připojení nových přepínačů není potřeba jejich manuální konfigurace a i díky tomu je škálovatelnost sítě na velmi vysoké úrovni. Technologie podporuje rozložení zátěže všemi spoji a velmi rychlou konvergenci sítě. (Brocade, 2015)

4.7 Software-Defined Networking

Technologie softwarově definovaných sítí (Software-Defined Networking – SDN) byla vytvořena se snahou o komplexní změnu systému řízení provozu v počítačových sítích na druhé a třetí vrstvě modelu OSI. Hlavními přednostmi SDN jsou otevřenost, virtualizace síťových funkcí, centrální řízení a programovatelnost. Hlavním prvkem v síti je kontrolér, který centrálně řídí všechny směrovače, přepínače a další síťové prvky. Komunikace mezi prvky a kontrolérem probíhá obvykle pomocí protokolu OpenFlow.

SDN díky těmto principům odděluje směrovací a řídicí logiku na dvě nezávislé abstraktní vrstvy. Každá z nich nabízí otevřené rozhraní pro programování aplikací (Application

Programming Interface – API). Díky tomu je možné lépe řešit dohled nad sítí, připojovat nové softwarové aplikace a určitým způsobem naprogramovat i nestandardní funkce prvků. Výhodou SDN je i možnost hybridního fungování s počítačovou sítí, která SDN nepodporuje. (BusinessIT.cz, 2013; Zloch, 2014)

V roce 2011 byla založena nezisková organizace Open Networking Foundation, která vytváří nezávislé standardy SDN a OpenFlow. V organizaci je dnes zapojeno více než 170 firem, mimo jiné i Intel, Brocade, Facebook, Google, Huawei, Juniper a Microsoft. (Open Networking Foundation, 2015)

5 Protokol TRILL

V roce 2004 byla společností IETF vytvořena pracovní skupina nazvaná TRILL, která v červenci roku 2011 vydala RFC 6325, který představuje první standard vznikajícího protokolu. Postupně bylo vydáno dalších 7 RFC, které se protokolu týkají. Poslední RFC před vznikem této práce bylo vydáno v květnu 2014[2] a obsahuje různá objasnění, opravy a aktualizaci standardu.

Cíle protokolu TRILL vycházejí z nutnosti nahradit STP lepším řešením a byly definovány v květnu 2009 (RFC 5556, 2009):

- TRILL při nejasném stavu musí raději zablokovat port, než aby byl otevřený,
- musí spolehlivě bránit zacyklení rámců v síti,
- podporovat libovolnou topologii podsítě na spojové vrstvě modelu OSI,
- používat nejvýhodnější cesty k cílovému zařízení,
- přenášet data mezi stejnými zařízeními přes více souběžných spojů,
- podporovat nástroje pro analýzu sítě,
- nesmí blokovat redundantní trasy v síti,
- dále je potřeba zachovat určité vlastnosti Ethernetových sítí:
    - rámec musí být cílovému zařízení doručen v nezměněné podobě (přesně jak byl vytvořen zdrojovým zařízením),
    - v síti je možné využívat rámce se skupinovou nebo všesměrovou cílovou MAC adresou,
    - jednoduchost konfigurace přepínačů – v praxi nesmí být pro základní funkčnost protokolu potřeba žádná konfigurace,
    - možnost společného fungování klasických i TRILL přepínačů v jedné síti. (Matuška, 2010; RFC 5556, 2009)

V RFC dokumentech jsou zavedeny termíny specifické pro TRILL. V následujícím textu bude v souladu s těmito dokumenty používán termín směrující přepínač (Routing Bridge – RBridge), který označuje přepínač implementující TRILL. Termín přepínač bude používán pro klasické přepínače, které TRILL neimplementují.

5.1 Základní principy protokolu

Již označení směrující přepínač (RBridge) napovídá, že TRILL využívá principy směrování, které se již delší dobu uplatňují v síťové vrstvě modelu OSI. V první řadě to jsou:

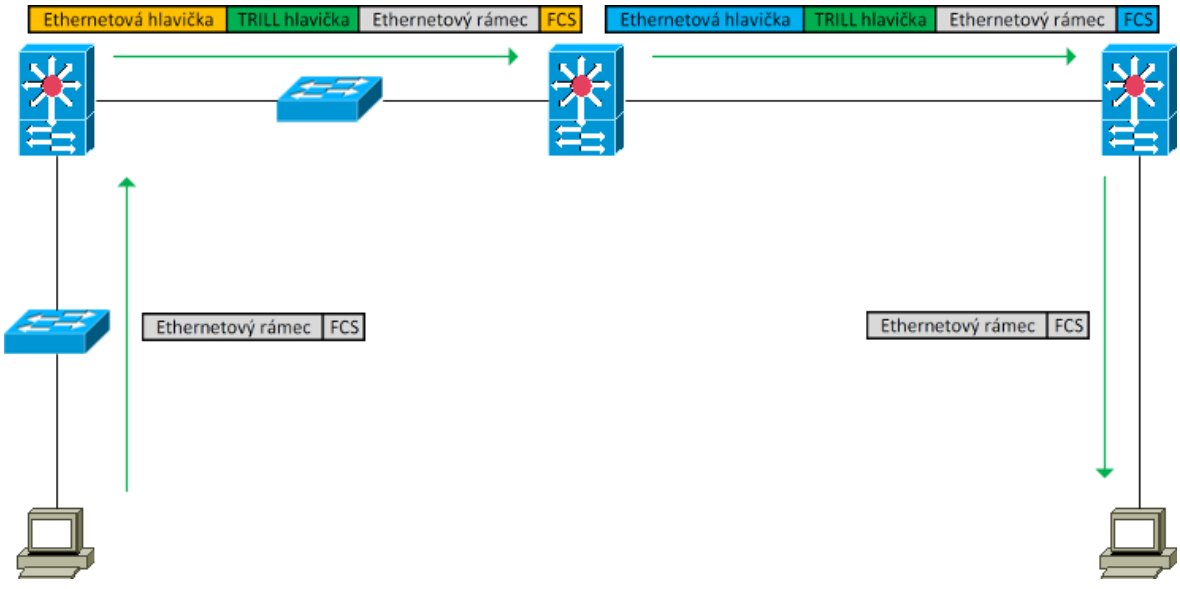
[2] RFC 7180, dostupné z: https://tools.ietf.org/html/rfc7180 [cit. 2015-03-10]

- Na síťové vrstvě velmi známý a používaný protokol IS-IS, který je využit pro hledání nejlepších cest v podsíti,
- distribuční stromy a kontrola směrování opačnou cestou (Reverse Path Forwarding – RPF) při zasílání skupinových rámců,
- využití příznaku omezené platnosti dat (Time to Live – TTL), který je využíván v IP paketech i pro provoz na spojové vrstvě modelu OSI. (Matuška, 2010)

Pro splnění všech cílů pomocí těchto principů je potřeba spolu s daty přenášet další informace, které se v Ethernetových rámcích nevyskytují. Zároveň ale nesmí být hlavička rámce upravena, aby TRILL mohl fungovat se všemi současnými Ethernetovými zařízeními. Vložení nových dat k rámci je vyřešeno pomocí zapouzdření celého Ethernetového rámce do nové hlavičky protokolu TRILL. Aby bylo možné mezi dva RBridge umístit jeden nebo více přepínačů a umožnit tak obvyklý Ethernetový provoz i mezi různými RBridge, je celý rámec včetně přidané TRILL hlavičky obalen další Ethernetovou hlavičkou (metoda MAC-in-MAC zapouzdření). RBridge tak pracuje se dvěma typy datových rámců:

- Klasické Ethernetové rámce, které jsou přijímány a zasílány koncovým zařízením,
- rámce s dvojitou Ethernetovou hlavičkou a vloženou TRILL hlavičkou.

Výhody protokolu TRILL jsou v podsíti patrné již při využití dvou RBridge i bez výměny ostatních přepínačů. Jedna z možných cest rámce v podsíti je znázorněna na následujícím obrázku (Obrázek 15).



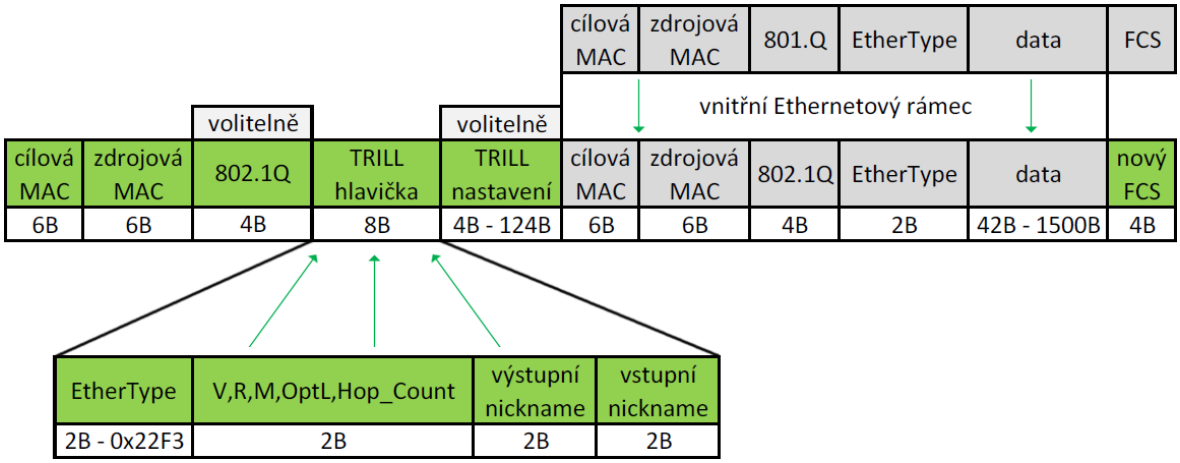
Obrázek 15 – Cesta rámce přes RBridge

Jak lze vidět na obrázku, koncová zařízení vždy odesílají nebo přijímají tradiční Ethernetový rámec, nepotřebují se tak protokolu TRILL žádným způsobem přizpůsobovat a protokol pro ně působí transparentně. Všechny klasické přepínače fungují také běžným způsobem – podle hlavičky rámce přeposílají tento rámec cílovému zařízení, které pro ně může být

i určitý RBridge. Skutečnost, že obsahem přeposílaného Ethernetového rámce je hlavička TRILL s vnořeným Ethernetovým rámcem, klasický přepínač nerozpozná a ke své práci to rozpoznat nepotřebuje. Klasický přepínač se orientuje pouze podle vnější Ethernetové hlavičky a případně podle FCS. Pouze některé zastaralé přepínače, které nepodporují rámce s daty větší než 1500B, by mohly mít problém s velikostí celého rámce, protože TRILL hlavička s vnější Ethernetovou hlavičkou velikost celého rámce zvětšují. Pokud přepínač takto velký rámec nedokáže zpracovat, nepřeposílá ho dál a zruší ho. Je odpovědností vyšších vrstev modelu OSI, aby rámec rozdělily a zaslali ho znovu. Přesto TRILL testuje maximální možnou velikost rámců na cestě mezi dvěma RBridge, ale pouze pro svoji potřebu, aby nevytvářel kontrolní rámce s větší velikostí, než jaká je na spoji přípustná. (RFC 6325, 2011)

Jediným polem, které TRILL v původním Ethernetovém rámci upravuje, je kontrolní součet FCS. Aby sítí nebyly přenášeny dva FCS za sebou, je původní FCS nahrazen novým, který je vygenerován z celého zasílaného rámce (a řeší tak kontrolu celého obsahu včetně vnořeného rámce). Poslední RBridge na cestě, který poté zasílá pouze původní rámec koncovému zařízení, celkový FCS maže a k původnímu rámci vypočítá a přiřadí nový FCS, který se v případě neporušeného rámce musí shodovat s původním FCS, který byl vypočítán při vytvoření původního rámce.

5.2 Formát rámce



Obrázek 16 – Formát TRILL rámce, zpracováno podle (Hooda, 2014)

Na obrázku (Obrázek 16) je znázorněna struktura rámce, který je zasílán mezi dvěma RBridge. Vnitřní Ethernetová hlavička obsahuje MAC adresu zdrojového zařízení a cílovou MAC adresu např. směrovače nebo jiného zařízení vyšší vrstvy v podsíti. Vnější Ethernetová hlavička pak obsahuje MAC adresy sousedních RBridge a volitelně (důležité pokud je více RBridge připojeno k LAN) hodnotu EtherType specifickou pro protokol TRILL. Každý RBridge rozhoduje o přeposlání rámce podle cílové adresy vnitřního Ethernetového rámce a TRILL hlavičku a vnější Ethernetovou hlavičku vždy změní tak, aby celý rámec byl doručen dalšímu RBridge na cestě k cílovému zařízení. Vložená hlavička TRILL obsahuje následující pole:

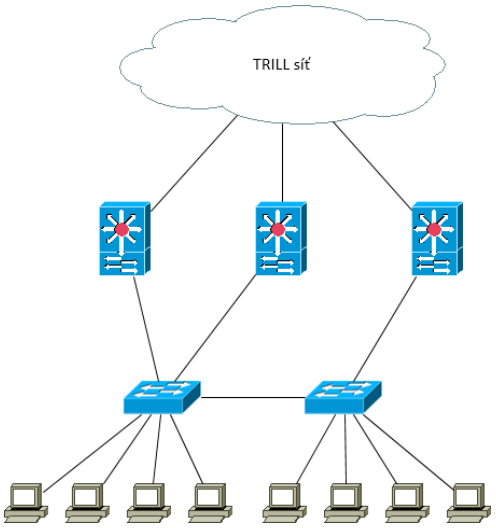
- EtherType obsahuje hodnotu 0x22F3, která identifikuje TRILL.
- V je 2b pole pro označení verze protokolu. Pokud RBridge danou verzi nepodporuje, rámec je zrušen.
- 2b R zatím nebyly využity a jsou rezervovány pro případné využití v budoucnu.
- M bit označuje možnost skupinového rámce – pokud je nastavený na hodnotu jedna, výstupní nickname je název distribučního stromu, jinak je rámec určen jednomu příjemci.
- OptL je 5b pole obsahující délku nepovinné části TRILL hlavičky – nastavení. Ta nemusí být v rámci vůbec (hodnota 0), nebo ve velikosti 4B – 124B.
- Hop count nastavuje zdrojový RBridge na určitou hodnotu, která je na každém dalším RBridge snížena o jedničku. Pokud hodnota klesne na nulu, je informován RBridge se vstupním nickname a rámec je vyřazen. Účel tohoto pole je stejný jako TTL v IP paketech a zabraňuje rámcům v nekonečném kolování sítí.
- Výstupní nickname označuje cílový RBridge nebo distribuční strom. Pokud RBridge rámec pouze přeposílá, není pole měněno.
- Vstupní nickname označuje zdrojový RBridge. Toto pole také není při průchodu sítí změněno.
- TRILL nastavení může mít různou velikost. V hlavičce nemusí být vůbec nebo může obsahovat 4B – 124B dat podle hodnoty OptL. Kromě následujících dvou příznaků, které jsou součástí prvního B, v době psaní této práce nebyla definována žádná pole.
- CHBH bit (Critical Hop by Hop bit) má hodnotu jedna v případě, že v rámci jsou vložena kritická nastavení ovlivňující přeposílání dat na každém RBridge. V případě, že RBridge některé z nastavení nezná, musí rámec vyřadit. Pokud je hodnota nula, RBridge přeposílá rámec i v případě, že nezná všechna nastavení rámce.
- CItE bit (Critical Ingress to Egress bit) plní stejnou funkci jako předchozí bit, ale je specializovaný na nastavení, která se týkají vstupně-výstupních nastavení. Ty jsou součástí všech nastavení, takže v případě hodnoty jedna v předchozím poli CHBH může být rámec vyřazen bez ohledu na hodnotu CItE. (RFC 7179, 2014)

5.3 Využití nickname RBridge

V TRILL hlavičce se nachází vstupní a výstupní nickname, které identifikují jeden nebo více RBridge. Nickname jsou použity pro směrování rámců směrem k cílovému zařízení. Každá nickname je 16b dynamicky přiřazované číslo, které je využito protokolem IS-IS při výpočtu nejlepších cest. Hodnota nula je rezervována pro označení nespecifikované nickname, hodnoty 0xFFC0 až 0xFFFE jsou rezervovány pro definování speciálních účelů v budoucnu a hodnota 0xFFFF nebude nejspíše nikdy specifikována. Zvolení nickname je poměrně složitý proces zajišťovaný protokolem IS-IS. Každý RBridge má obvykle pouze jednu, ale může mít i více nickname, které by se měl pokusit využít i po restartu zařízení. Pokud

je nickname konfigurována administrátorem, má větší prioritu než ostatní, které jsou vybrány automaticky. V případě zjištění více stejných nickname v podsíti, je spuštěna procedura, díky které si jeden z RBridge vytvoří novou nickname, aby byla duplicita odstraněna. Vzniku duplicit se předchází dvěma způsoby – RBridge obvykle volí svou nickname pseudonáhodně nebo hashem svých parametrů a podle doporučení by před zvolením nickname měl čekat do chvíle, než má k dispozici kompletní databázi všech RBridge v podsíti, aby nezvolil nickname, kterou již používá jiný RBridge. (RFC 6325, 2011)

5.4 Připojení RBridge k LAN



Obrázek 17 – Připojení více RBridge k jedné LAN

V situaci znázorněné na obrázku (Obrázek 17), kdy je k jedné LAN připojeno více RBridge, je nutné, aby provoz z LAN byl odesílán do dalších částí sítě (a zpět do této LAN) pouze jedním RBridge a ostatními nebyl duplikován. Mezi všemi připojenými je vybrán jeden RBridge označovaný jako designated RBridge. Ten vybere několik VLAN, pro které bude sám přenášet provoz, a může pověřit ostatní RBridge připojené k LAN pro přenos jiných konkrétních VLAN, aby byl provoz rozložen. (Hooda, 2014)

5.5 Řídicí vrstva

Vzhledem ke komplexnosti protokolu TRILL je potřeba jeho funkci rozdělit na tři vrstvy – datovou, řídicí a třetí vrstvu managementu, která umožňuje administrátorům konfiguraci protokolu. Řídicí vrstva řeší především výpočet nejlepších cest v síti, jehož výsledky jsou použity datovou vrstvou pro přeposílání rámců. K výpočtu je využit modifikovaný protokol IS-IS. V rozsahu této práce nelze popsat všechny jeho detaily, jsou zde popsány pouze základní mechanismy, které protokol TRILL při výpočtu cest využívá.

5.5.1 Konvergence

Pro přesné směrování rámců je potřeba, aby všechny RBridge v podsíti znaly její kompletní topologii. Proto je v první řadě nutné, aby každý RBridge znal své nejbližší sousedy.

Protokol IS-IS tedy zajišťuje odesílání tzv. Hello skupinových rámců všemi porty, na kterých je protokol aktivován. Když tento rámec přijme jiný RBridge, odpoví na něj a pomocí třícestného (v LAN pouze pomocí dvoucestného) handshake procesu se dva RBridge stávají svými sousedy. Dále jsou mezi nimi zasílány kontrolní Hello rámce pro udržení sousedství – pokud RBridge během určitého intervalu nedostane od souseda kontrolní rámec, sousedství je zrušeno.

Aby všechny RBridge měly stejné informace o topologii, sousedi si vyměňují záznamy, které mají ve svých databázích. Ty se skládají především z informací o jednotlivých RBridge a vlastnostech spojů mezi nimi. Stejné informace se postupně dostanou ke všem RBridge. Obvykle jsou mezi RBridge zasílány pouze informace o změnách topologie, aby síť nebyla zahlcena periodickým zasíláním stále stejných informací. Pokud RBridge má svoji databázi prázdnou, může si od souseda vyžádat zaslání jeho kompletní databáze. (Hooda, 2014)

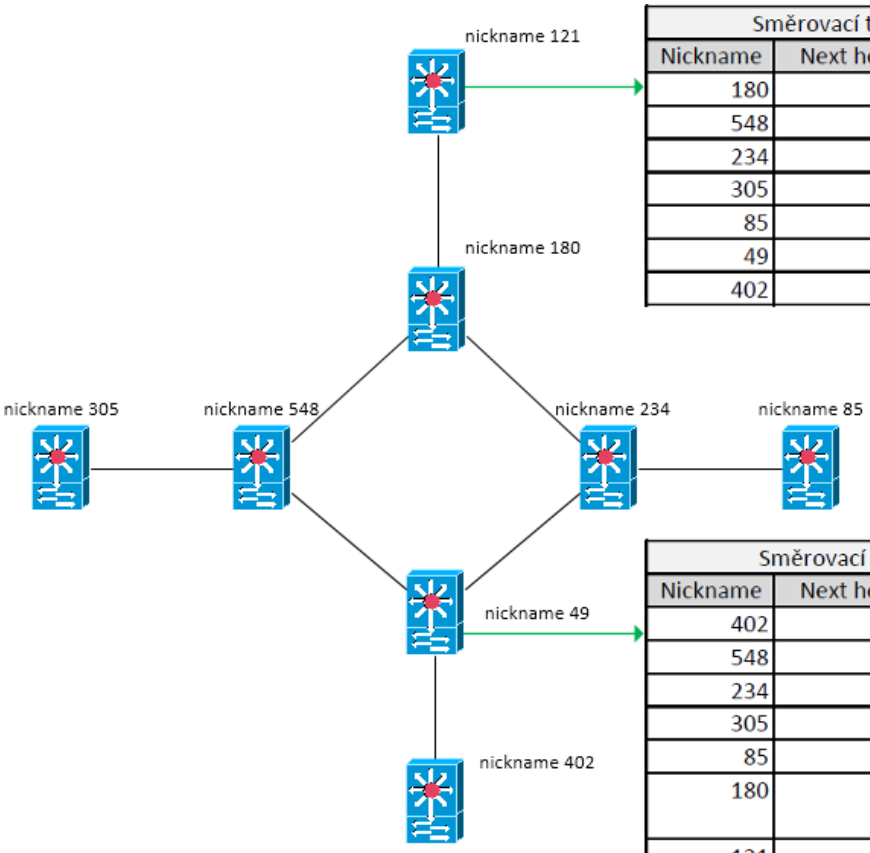
Ve chvíli, kdy RBridge má kompletní informace o topologii, probíhá na každém RBridge samostatně výpočet cest, který díky stejným výchozím informacím končí na všech RBridge se stejnými výsledky. Po každé změně topologie je rozeslána informace o této změně všem RBridge a každý z nich pak musí všechny cesty přepočítat. V rozsáhlejších topologiích to může způsobovat problémy, protože výpočty cest z velkých databází jsou náročné na paměť a procesor RBridge. Protokol IS-IS proto umožňuje rozdělit síť na více oblastí, mezi kterými jsou zasílány pouze omezené informace, a kompletní výměna informací probíhá jen uvnitř jedné oblasti. Tato možnost ale není v současné době protokolem TRILL implementována. (RFC 6325, 2011)

TRILL v případě výpadku některého RBridge nebo spoje zajišťuje rychlou konvergence podsítě. Díky TTL v hlavičce rámce mohou bez následků dokonce být na malou chvíli povoleny smyčky v podsíti, dokud protokol IS-IS nezaplaví všechny RBridge informací o změně topologie. I při propojení více než 500 RBridge by měla konvergence proběhnout za méně než 500 ms. V menších sítích je konvergence mnohem rychlejší. (Weiguo, 2012)

5.5.2 Rámce s jedním cílem

Ke směrování rámců jsou využity nickname jednotlivých RBridge. Každý RBridge vypočítává cesty k ostatním RBridge metodou „nejdříve nejkratší cesta“ (Shortest Path First – SPF) pomocí Dijkstrova algoritmu. Z vypočítaných údajů TRILL vytváří směrovací tabulku všech RBridge nickname v podsíti, ke kterým přiřazuje nickname svého nejbližšího RBridge na cestě k němu (Next Hop) a výstupní port, kterým je možné poslat rámec směrem k danému Next Hop RBridge. Protokol se inspiroval směrovací tabulkou, která je využita na síťové vrstvě modelu OSI (kde se místo nickname využívají IP adresy). Nejkratší cestou je myšlena cesta s nejlepšími přenosovými parametry (výpočet je založen na rychlosti přenosu spoje a počtu RBridge na cestě k cíli). (RFC 6325, 2011)

Pokud je k cílovému zařízení nalezeno více nejkratších cest (cest se stejným ohodnocením), může být využito mechanismu vícecestného směrování (Equal-Cost Multi-Path Routing – ECMP), který pro přenos rámců může využít až 16 různých cest k cílovému zařízení a může tak rozložit zátěž podsítě na více spojů. (Hooda, 2014)



Směrovací tabulka RBridge 121

Nickname	Next hop	Hop count	Cost
180	180	1	10
548	180	2	20
234	180	2	20
305	180	3	30
85	180	3	30
49	180	3	30
402	180	4	40

Směrovací tabulka RBridge 49

Nickname	Next hop	Hop count	Cost
402	402	1	10
548	548	1	10
234	234	1	10
305	548	2	20
85	234	2	20
180	548 234	2	20
121	548 234	3	30

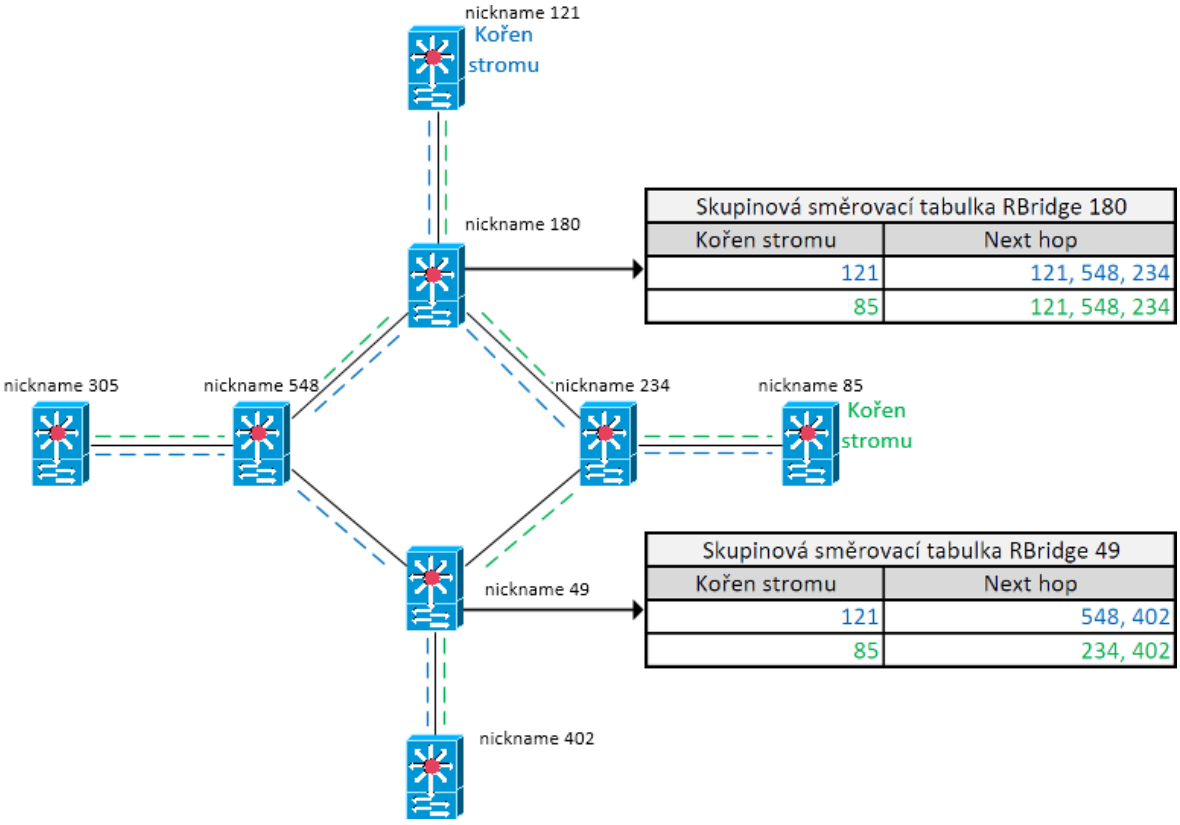
Obrázek 18 – Směrovací tabulky RBridge, zpracováno podle (Hooda, 2014)

Na obrázku (Obrázek 18) je zobrazena jednoduchá topologie RBridge se zobrazením směrovacích tabulek ve dvou z nich. Směrovací tabulka obsahuje nickname cílových RBridge, další RBridge na cestě k němu (Next hop) a informace o cestě – počet RBridge na cestě (Hop count) a celkové ohodnocení cesty (Cost). Pro názornost je ohodnocení všech spojů na obrázku 10. Ve směrovací tabulce jsou uloženy jen nejlepší cesty podle ohodnocení. Pokud RBridge zná více cest se stejným ohodnocením, může jich být uloženo až 16.

5.5.3 Skupinové rámce

Kromě výpočtu směrovací tabulky pro rámce s jedinečným cílem si každý RBridge počítá několik distribučních stromů pro zasílání skupinových rámců. To mohou být skupinové rámce vytvořené protokolem síťové vrstvy nebo i všesměrové rámce např. protokolů Address Resolution Protocol (ARP), Dynamic Host Configuration Protocol (DHCP) nebo UDP. Zároveň jsou tímto způsobem rozesílány rámce, které mají jeden cíl, ale RBridge nezná umístění cílového zařízení v podsíti (adresa cílového zařízení zatím není v CAM tabulce ani v tabulce cílových zařízení). Podle protokolu IS-IS je mezi všemi RBridge v podsíti vybrán jeden kořenový RBridge (root RBridge), který určuje počet distribučních stromů v podsíti, dále určuje kořenový RBridge každého jednotlivého distribučního stromu a každému distribučnímu stromu přiřazuje identifikační číslo. Všechny RBridge jsou součástí každého distribučního stromu a všechny RBridge v podsíti počítají distribuční stromy samostatně, ale protože mají stejné informace o topologii, mají i stejně vypočítané

distribuční stromy. Existují pravidla, podle kterých se RBridge řídí ve chvíli, kdy je více možností, jak distribuční strom vytvořit. V distribučním stromu tak nelze využít ECMP, ale na každém RBridge může mít strom více odchozích portů, aby byl skupinový rámec doručen do všech částí podsítě. Dva distribuční stromy (modrý a zelený) jsou vytvořeny v topologii na následujícím obrázku (Obrázek 19).



Obrázek 19 – Skupinové směrovací tabulky RBridge, zpracováno podle (Hooda, 2014)

Každý distribuční strom je vypočítán pomocí nejkratších cest z jeho kořene, takže cesty stromu mezi ostatními uzly již nemusí korespondovat s nejkratšími cestami mezi nimi. V nejlepším případě by tak každý RBridge v podsíti měl být kořenem jednoho distribučního stromu, aby skupinové rámce, které RBridge vytvoří, byly doručeny přes tento distribuční strom. Každá malá změna topologie ale způsobí přepočítání všech distribučních stromů a ve větších podsítích by velký počet distribučních stromů mohl při přepočítání způsobovat problémy. Proto je v podsíti obvykle méně distribučních stromů než RBridge a pokud RBridge není kořenem stromu a potřebuje odeslat skupinový rámec, obvykle (záleží na konkrétní implementaci) jej rozešle přes distribuční strom, k jehož kořenu má RBridge nejkratší cestu ze všech kořenů různých distribučních stromů. (Hooda, 2014)

5.5.4 Pruning

Pruning je metoda, která omezuje šíření skupinových a všesměrových rámců do těch částí sítě, kde nejsou žádní příjemci, kteří by daný rámec přijali. Přestože je pro skupinové rámce vytvořen kompletní distribuční strom se všemi RBridge, rámec není všem RBridge odeslán, pokud to není nezbytné. RBridge implementují chování protokolu síťové vrstvy, který

spravuje management internetových skupin (Internet Group Management Protocol – IGMP). Tento protokol umožňuje cílovým zařízením přihlásit se k odběru skupinových rámců. RBridge zpracovává všechny IGMP rámce a ostatním RBridge odesílá informace o odebíraných skupinách ve vlastních řídicích IS-IS rámcích. Kromě toho ale původní IGMP rámec přeposílá na zařízení síťové vrstvy, aby se o odebírané skupině dozvěděl i směrovač, který má do podsítě zasílat skupinové pakety. Stejným způsobem lze omezit i provoz všesměrových rámců v konkrétní VLAN. Pokud nejsou k RBridge připojena žádná zařízení z VLAN, pruning zajišťuje, že na RBridge nebude z této VLAN odeslán žádný všesměrový rámec. (Hooda, 2014)

5.6 Datová vrstva

Hlavním úkolem datové vrstvy protokolu TRILL je směrování rámců směrem k cílovému zařízení. To je většinou řešeno pomocí cílové MAC adresy a příslušnosti zdrojového zařízení ke konkrétní VLAN. RBridge může podle hardwarové implementace nabízet i další možnosti pro směrování paketů, obvykle např. podle zdrojového portu. RBridge rozlišuje směrování klasických rámců, rámců s TRILL hlavičkou, skupinových rámců nebo rámců s jednoznačně určeným cílem. Pokud RBridge nezná umístění cílového zařízení v podsíti (v CAM tabulce ani v tabulce cílových zařízení zatím nemá uloženou příslušnou MAC adresu), je rámec zpracován stejným způsobem jako všesměrový. (Hooda, 2014)

5.6.1 Rámce s jedním cílem

Rámec s jasným cílem, který je dán MAC adresou, je v TRILL podsíti zpracován třemi různými způsoby. Na prvním (vstupním) RBridge je rámec obalen TRILL hlavičkou, další RBridge (v jádru podsítě) TRILL hlavičku upravují a poslední (výstupní) RBridge na cestě ji odstraňuje. Jiná situace samozřejmě nastává ve chvíli, kdy zdrojové i cílové zařízení je připojeno k jednomu RBridge – rámec je zpracován stejným způsobem jako u klasických přepínačů. V dalším popisu je předpokládáno, že rámec prochází větší TRILL sítí.

První RBridge přijme klasický Ethernetový rámec, ze kterého si do CAM tabulky může (pokud to již neučinil) uložit zdrojovou MAC adresu společně s portem, ze kterého byl rámec přijat. Toto chování je stejné jako u klasických přepínačů, které následně podle cílové MAC adresy rámce zjišťují port, kterým mají rámec odeslat. Pro RBridge je ale důležitá jiná informace – podle cílové MAC adresy vybere nickname výstupního RBridge v tabulce cílových zařízení. Tu vkládá do TRILL hlavičky jako výstupní nickname. Dále podle této nickname vyhledává Next Hop RBridge (v případě ECMP vybere jeden z více možností), jehož MAC adresu vloží do vnější Ethernetové hlavičky a odesílá rámec příslušným portem. Zdrojové adresy TRILL a vnější Ethernetové hlavičky nastaví RBridge na vlastní nickname a MAC adresu.

Všechny RBridge na cestě mezi vstupním a výstupním RBridge pouze upraví TTL v TRILL hlavičce a ve vnější Ethernetové hlavičce nastaví vlastní zdrojovou MAC adresu a MAC adresu cíle změní na adresu Next Hop RBridge. Pokud je hodnota TTL po snížení rovna nule, rámec je vyřazen a o jeho vyřazení je zaslána informace zdrojovému RBridge. Next

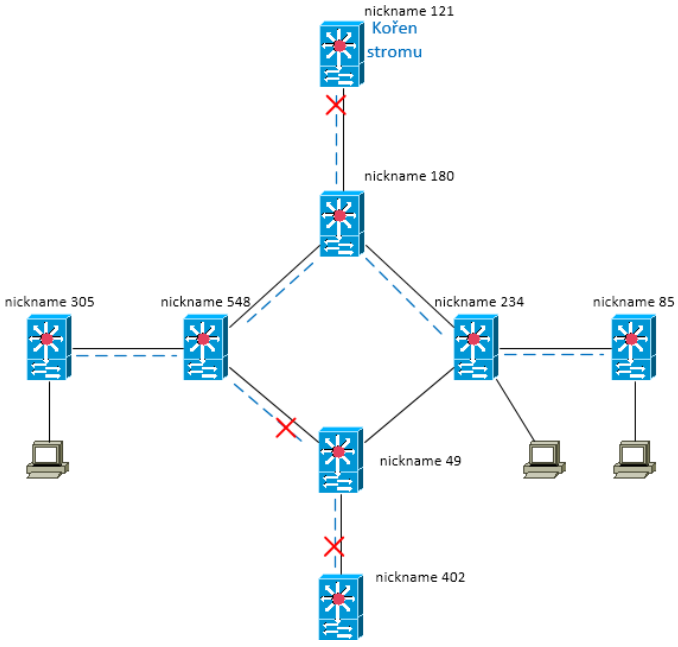
Hop RBridge musí být vyhledán podle nickname výstupního RBridge v TRILL hlavičce. Tyto RBridge se nezajímají o vnitřní Ethernetovou hlavičku.

Výstupní RBridge si podle zdrojové MAC adresy ve vnitřním Ethernetovém rámci a podle zdrojové nickname v TRILL hlavičce uloží informaci o umístění zařízení v podsíti, kterou využije při směrování rámců v opačném směru – tedy při směrování rámců k (v tuto chvíli) zdrojovému zařízení. Kromě uložení neznámých adres z rámce odstraní přidanou hlavičku, přepočítá FCS a odešle rámec v původním stavu klasickým způsobem na určitý svůj port. Pokud výstupní RBridge ve své CAM tabulce nenalezne MAC adresu cílového zařízení, nezasílá rámec zpět ostatním TRILL zařízením, ale odesílá ho do LAN všemi svými ostatními porty. (Hooda, 2014)

5.6.2 Skupinové rámce

Přes distribuční stromy jsou rozesílány tři druhy rámců: všesměrové, skupinové a rámce s jedním cílem, pokud RBridge nezná jeho umístění v podsíti. Všesměrové rámce mají cílovou MAC adresu se speciální hodnotou FF:FF:FF:FF:FF:FF, skupinové rámce mají nejméně významný bit v prvním byte MAC adresy nastaven na jedničku.

Na vstupním RBridge jsou všechny tyto rámce označeny jako skupinové jedničkou v bitu M v TRILL hlavičce. Výstupní nickname pak označuje vybraný distribuční strom, kterým je rámec šířen. Vstupní RBridge může vybrat distribuční strom, ve kterém je sám kořenem, nebo strom, kde je kořenem jiný RBridge. Výběr konkrétního stromu není protokolem definován a záleží na konkrétní implementaci. Obvykle je strom vybírán tak, aby kořen stromu byl co nejblíže vstupnímu RBridge. Pro rozložení zátěže je možné vybrat i více stromů – každý rozesílaný rámec může být zaslán přes jiný distribuční strom. V distribučním stromu nezáleží na směru rozesílání rámce a díky pruningu může být i kořen stromu odříznut od příjmu rámců, jak je znázorněno na obrázku (Obrázek 20).



Obrázek 20 – Pruning skupinových rámců, zpracováno podle (Hooda, 2014)

Rozeslání rámce v podsíti probíhá podobným způsobem jako v případě rámce s jedním určitým cílem, ale RBridge hledá porty pro rozeslání rámců podle nickname distribučního stromu ve skupinové směrovací tabulce (a poté porovnává s možným pruningem a také rámec neodesílá zpět portem, kterým ho přijal). V TRILL hlavičce jsou provedeny dvě změny, M bit je nastaven na jedničku a výstupní nickname označuje distribuční strom. Ve vnější Ethernetové hlavičce není MAC adresa cíle nastavena na sousední RBridge, ale je nastavena na jednu ze speciálních skupinových MAC adres, které jsou rezervovány pro protokol TRILL. Hodnota může být od 01-80-C2-00-00-40 do 01-80-C2-00-00-4F. (RFC 6325, 2011)

Každý RBridge po přijetí skupinového rámce od sousedního RBridge provádí RPF, aby zjistil, jestli byl rámec přijat z portu, který je součástí distribučního stromu. Pokud by rámec byl přijat na jiném portu, je rámec vyřazen. Kromě rozeslání rámce distribučním stromem musí každý RBridge zaslat rámec i koncovým zařízením, které jsou k němu připojeny. V tomto případě jsou odstraněny vnější Ethernetová a TRILL hlavičky, aby koncová zařízení přijala rámec v původním formátu.

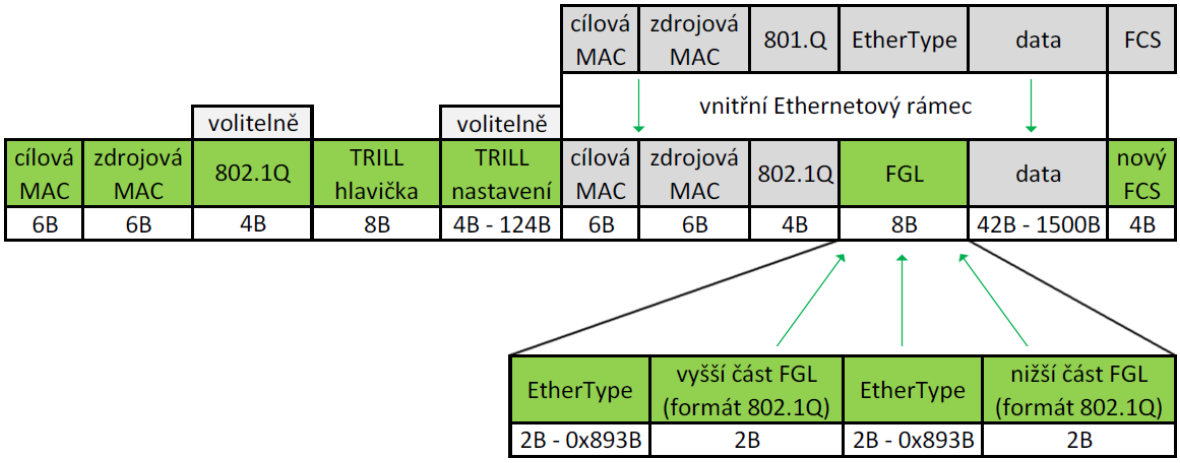
5.7 Učení MAC adres na RBridge

Tabulka s MAC adresami a informacemi o umístění jednotlivých zařízení v podsíti může být na RBridge naplňována dvěma způsoby. Plnění tabulky na datové vrstvě bylo již popsáno – RBridge si ukládá zdrojovou MAC adresu zařízení, jeho VLAN a nickname vstupního RBridge, přes který pak zasílá rámce určené danému zařízení. Aby se předešlo přeplněnosti tabulek s adresami, používá TRILL několik metod, které počet záznamů omezují. První z nich je ukládání záznamů pouze na výstupním RBridge. Všechny RBridge v jádru podsítě, které rámce pouze přenáší mezi dvěma koncovými RBridge, nepotřebují znát umístění koncových zařízení v síti, protože pracují podle informací v TRILL hlavičce (vstupní a výstupní nickname). Každý výstupní RBridge si v tabulce vytváří záznamy i ze skupinových rámců (včetně rámců s jedním cílem, které jsou rozeslány po celé podsíti ve chvíli, kdy není známé umístění cílového zařízení). Přeplnění tabulek výstupních RBridge částečně brání pruning VLAN, díky kterému nejsou doručovány rámce k RBridge, které danou VLAN nevyužívají. Při návrhu sítě je tak důležité myslet na omezení skupinového a všesměrového provozu, který je generován i různými protokoly, např. ARP.

Druhou možností učení MAC adres je plnění tabulky na řídicí vrstvě. TRILL k tomu využívá protokol pro distribuci adres koncových zařízení (End System Address Distribution Information – ESADI), jehož použití je možné na RBridge zapnout pro specifikované VLAN. ESADI v každé VLAN, pro kterou je nakonfigurován, vytvoří skupinu všech RBridge, které mají protokol pro danou VLAN zapnutý. ESADI mezi těmito RBridge rozesílá skupinové rámce s informacemi o umístění koncových zařízení v podsíti. Všechny RBridge díky ESADI mohou mít pro danou VLAN naprosto stejné informace o koncových zařízeních. V podsíti mohou fungovat RBridge se zapnutým ESADI (vhodné pro výstupní RBridge) spolu s RBridge, které ESADI nepoužívají (vhodné pro RBridge v jádru, které nemají připojeny žádná koncová zařízení). (Hooda, 2014)

5.8 Fine-Grained Labeling

TRILL podle již popsaného standardu umožňuje vytváření VLAN podle standardu 802.1Q, který byl v této práci již také popsán. Ten využívá 12b pro označení VLAN, kterých tak v jedné síti může být 4096. Především ve velkých datových centrech nemusí tento počet VLAN dostačovat. Proto byl v květnu 2014 vydán RFC 7172, který aktualizuje původní TRILL standard a rozšiřuje vnitřní Ethernetovou hlavičku o nové pole Fine-Grained Labeling (FGL), jak je vidět na obrázku (Obrázek 21).

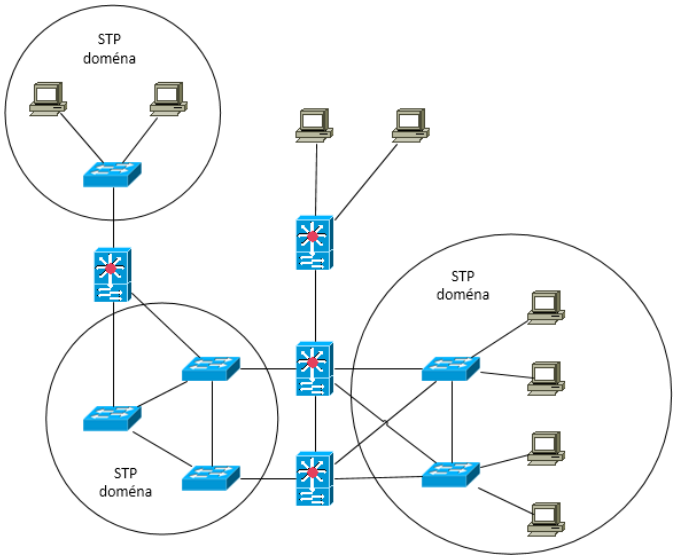


Obrázek 21 – Formát TRILL rámce s FGL, zpracováno podle (RFC 7172, 2014)

FGL má stejný formát jako pole 802.1Q (s novou hodnotou EtherType), ale tento formát je dvakrát zopakován. Tím je zvětšen prostor pro adresaci VLAN na 24b a počet VLAN tak může být větší než 16 milionů. Nevýhodou úpravy hlavičky je fakt, že rámce využívající FGL nemohou být směrovány přes RBridge, který FGL nepodporuje. Problémy v síti s kombinací různých RBridge, které FGL podporují a nepodporují, jsou špatně fungující pruning VLAN a vyřazování rámců s FGL na RBridge, které FGL nepodporují. Kombinace RBridge s podporou a bez podpory FGL je sice v síti možná, ale je nutné zajistit, aby FGL rámce byly směrovány pouze přes RBridge s podporou FGL (včetně ECMP provozu). Průchod rámců bez FGL je bezproblémový na všech RBridge. (RFC 7172, 2014)

6 Možnosti využití protokolu TRILL

Velmi užitečnou vlastností protokolu je možnost společného fungování RBridge a klasických přepínačů. Díky tomu je možné ve fungující síti nahrazovat přepínače pomocí nových RBridge postupně. Již při využití dvou RBridge je v síti znatelný přínos pro řízení přenosu dat, pokud jsou RBridge vhodně umístěny.

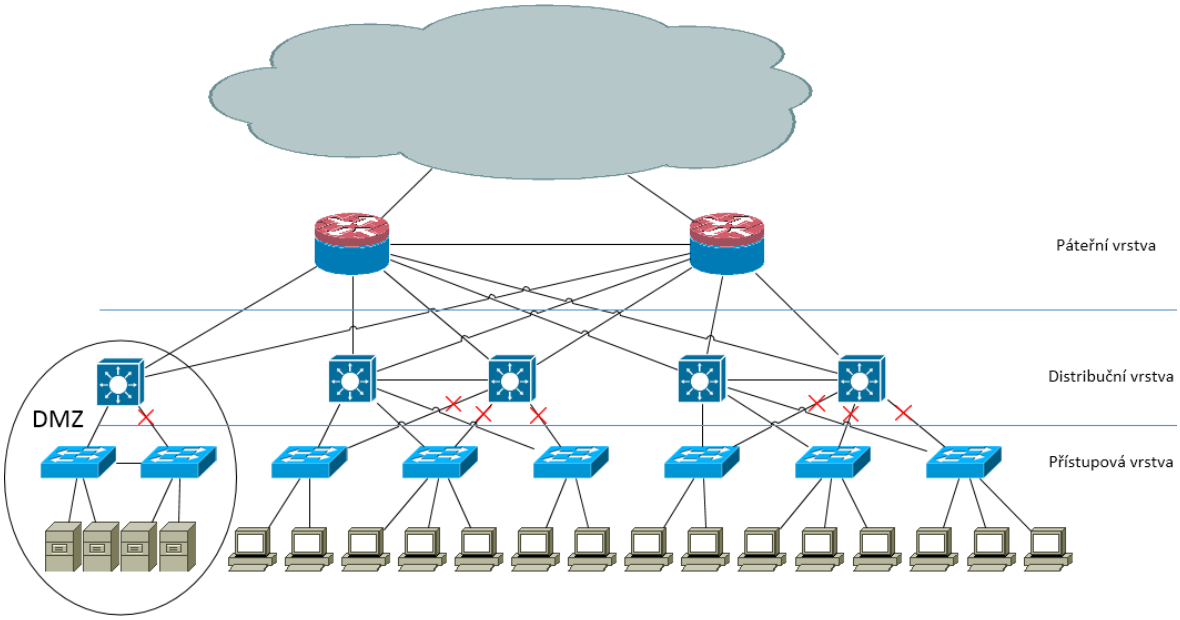


Obrázek 22 – Možnost společného fungování RBridge a klasických přepínačů v síti

Každý RBridge ukončuje STP oblast (Obrázek 22), takže při změně topologie (např. výpadku spoje) v této oblasti není ovlivněna celá podsíť, ale ostatní oblasti fungují bez výpadku. RBridge navíc mohou využít STP domény pro ECMP a využít tak i spojů, které by samotný STP zablokoval. S větším počtem RBridge se postupně potlačuje i stromová topologie STP, kdy většina provozu může procházet přes jeden klasický přepínač, i přestože by v podsíti vedli mnohem lepší cesty. Klasické přepínače jsou pro RBridge jako zařízení nižší vrstvy, které mezi nimi dokáží transparentně přenést data. K tomu navíc RBridge na každém portu naslouchá (ale obvykle negeneruje) řídicí provoz STP, aby mohl reagovat např. na změnu DR. Naslouchání STP je omezeno na daný port, ostatní porty nebo globální funkce RBridge nejsou ovlivněny. (Matuška, 2010)

6.1 Podnikové sítě

Podnikové sítě jsou obvykle budovány podle třívrstvé architektury – prvky jsou rozděleny do páteřní, distribuční a přístupové vrstvy. Síť obsahuje mnoho malých podsítí rozdělených pomocí zařízení síťové vrstvy (směrovačů) a komunikace probíhá většinou ve směru sever-jih (koncové zařízení-Internet). STP oblasti nejsou velké, případné problémy STP se tak projeví pouze v omezené části podnikové sítě, kde např. výpadek na několik vteřin, než RSTP zkonverguje podsíť, nemusí znamenat velký problém, který je potřeba řešit velkou investicí do nasazení RBridge. Na obrázku (Obrázek 23) je zobrazena podniková síť, která v distribuční vrstvě má přepínače pracující i na síťové vrstvě modelu OSI. Ty spolu s přepínači přístupové vrstvy tvoří STP domény, v nichž jsou blokovány přeškrtnuté spoje.



Obrázek 23 – Třívrstvá architektura podnikové sítě

I v podnikové třívrstvé architektuře ale jsou místa, kde jsou výhody protokolu TRILL zřejmé. I krátký výpadek v podsíti s podnikovými servery (DMZ) může znamenat velké finanční ztráty, takže zde využití RBridge může být výhodné. V případě nedostatečné kapacity spojů v podnikové síti (z důvodu blokovaných spojů STP) a nedostatku financí na nasazení RBridge v celé síti, lze TRILL využít dvěma způsoby: na spojích mezi přístupovou a distribuční vrstvou nebo na spojích mezi distribuční vrstvou a páteří podnikové sítě.

Pořízení nových RBridge na přístupové a distribuční vrstvě může být finančně nákladné, ale žádný ze spojů již nebude blokovaný (jako v případě STP) a bude využito i ostatních výhod protokolu včetně snadné konfigurace. Spoje mezi distribuční a páteřní vrstvou mohou zůstat na síťové vrstvě s využitím přepínačů, které na třetí vrstvě umí pracovat.

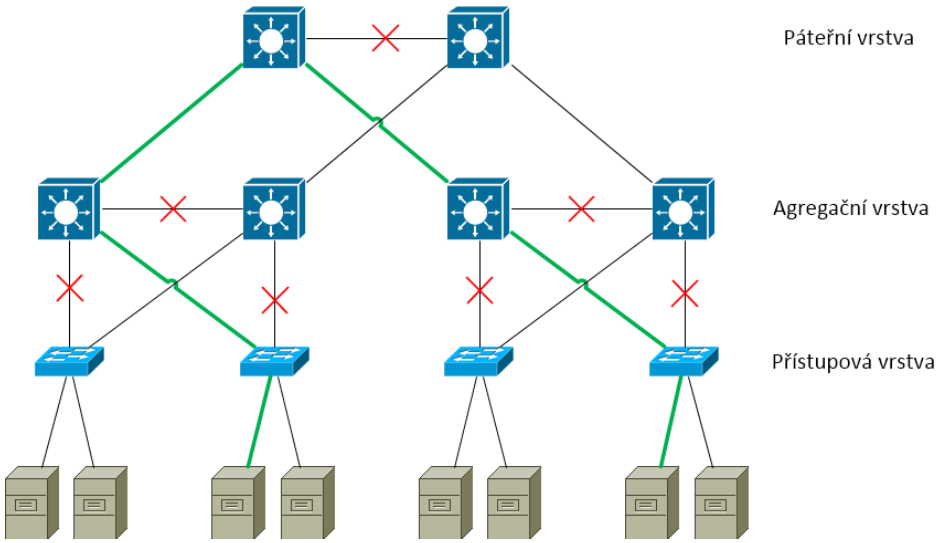
Využití protokolu TRILL na spojích mezi distribuční a páteřní vrstvou může přinést velké výhody při využití menšího počtu nových RBridge, které by ale v nejlepším případě měly umět přepínání paketů na síťové vrstvě a pro komunikaci s přístupovou vrstvou by měly umět využívat i jiné technologie (např. Cisco Virtual Switching System), které dokáží omezit některé nedostatky STP. Díky tomuto přístupu a využití VLAN lze rozšířit podsítě napříč všemi částmi podnikové sítě a určitá podsíť tak nemusí být omezena jen na lokální fyzické umístění. RBridge pracující kromě spojové vrstvy i na síťové vrstvě pak mohou řídit i směrování paketů mezi různými VLAN podle konfigurace a nastavených politik pro přístup mezi různými podsítěmi. (Hooda, 2014)

6.2 Datová centra

Mnohem větší přínos než v podnikových sítích přináší TRILL do datových center. V nich je obvykle mnohem důležitější omezit výpadky služeb, využití více cest mezi servery a hlavně využití velkých podsítí na druhé vrstvě modelu OSI. Moderní datová centra obvykle

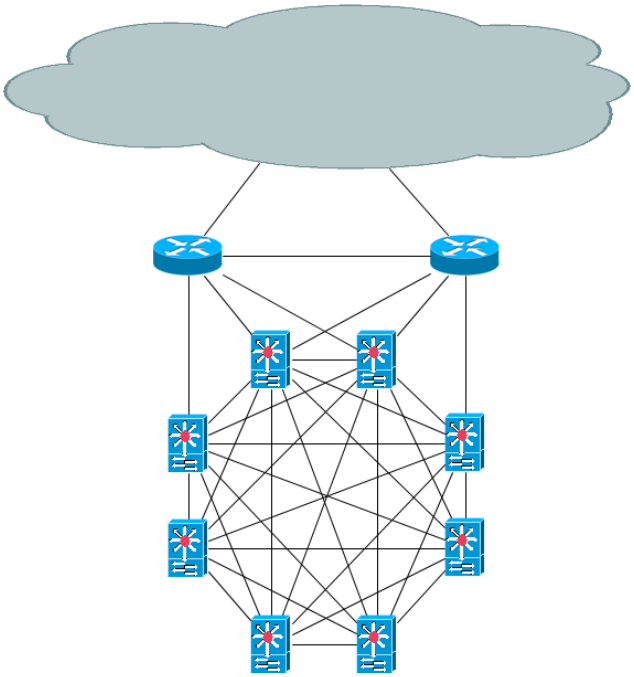
nabízí velké množství virtuálních serverů, které mohou měnit své umístění mezi různými fyzickými servery, a přesto potřebují mít stále stejnou IP adresu. Zároveň je zde velký provoz mezi jednotlivými servery (označovaný jako komunikace východ-západ). Proto jsou dnes v datových centrech budovány velké ploché sítě místo sítí strukturovaných do více vrstev. I z těchto důvodů je TRILL od začátku vývoje více zaměřen na využití v datových centrech a až později jsou a budou přidávány další možnosti využitelné spíše v podnikových sítích. (Matuška, 2010)

V minulosti byla datová centra obvykle budována pomocí třívrstvé architektury, která je podobná architektuře podnikových sítí, ale je složena z páteřní, agregační a přístupové vrstvy. V těchto topologiích, které využívají STP, nastávají stále stejné problémy a síť nemůže pro komunikaci mezi jednotlivými servery nabídnout řešení, které by uspokojilo potřeby moderních datových center. Na obrázku (Obrázek 24) je znázorněna cesta dat mezi dvěma servery a jsou označeny všechny spoje, které STP blokuje jako prevenci před vznikem smyček v síti.



Obrázek 24 – Síť datového centra s využitím STP, zpracováno podle (Brocade, 2009)

Na dalším obrázku (Obrázek 25) je znázorněna topologie moderního datového centra, které využívá protokol TRILL a serverům poskytuje všechny možnosti potřebné ke splnění dnešních náročných požadavků datových center. Architektura je složena pouze ze dvou vrstev. Páteřní vrstva pracuje na síťové vrstvě díky směrovačům připojeným k Internetu. Ostatní síťové prvky v datovém centru jsou RBridge pracující na spojové vrstvě a vytvářejí jednu velkou plochou síť složenou z mnoha VLAN, které umožňují připojení tisíců virtualizovaných serverů, které je možné libovolně přesouvat mezi různými fyzickými servery bez nutnosti další konfigurace síťových prvků. Tato architektura umožňuje velké a snadné škálování i díky tomu, že TRILL podporuje připojení nových RBridge do sítě bez potřeby manuální konfigurace. (Amamou, 2014)



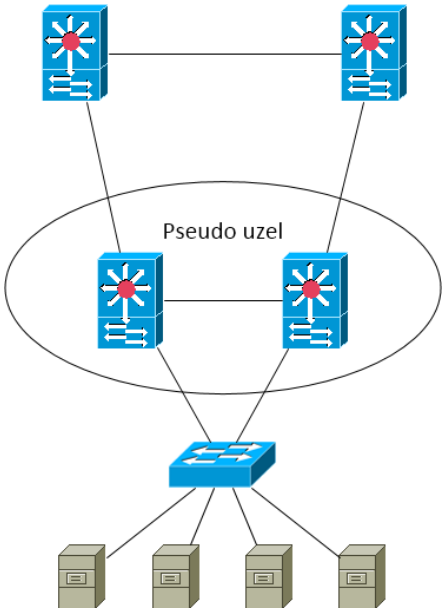
Obrázek 25 – Topologie datového centra s RBridge, zpracováno podle (Amamou, 2014)

6.3 Další vývoj protokolu

Protokol TRILL je organizací IETF dále rozvíjen a postupně jsou přidávány nové možnosti, které TRILL může využívat. Aktuální vývoj je možné sledovat na webových stránkách organizace IETF[3].

V průběhu psaní této práce[4] již měl vyjít standard pro vytvoření pseudo uzlů, který je ale zatím k dispozici pouze ve fázi návrhu (draft). Tento standard má definovat možnost vytvoření jednoho virtuálního RBridge ze dvou nebo více fyzických RBridge, které jsou připojeny paralelními spoji k další části sítě. Pseudo uzel, složený z více RBridge, se jeví jako jednotný a tato koncepce zajišťuje zálohu spojů a možnost rozložení zátěže při zasílání dat již v části sítě, která je za hranicemi oblasti spravované protokolem TRILL. Stejnou funkci plní například implementace Cisco Virtual Port Channel. Na dalším obrázku (Obrázek 26) je zobrazeno vytvoření pseudo uzlu ze dvou RBridge. Pseudo uzel z pohledu klasického přepínače představuje jeden síťový prvek a síťový provoz serverů tak může být rozložen mezi oba Rbridge.

[3] http://datatracker.ietf.org/wg/trill/charter/ [cit. 2015-03-14]
[4] Aktuálnost informací k datu 14. března 2015



Obrázek 26 – Vytvoření pseudo uzlu ze dvou RBridge

Jedna z důležitých aktualizací protokolu by měla být definována v srpnu tohoto roku a měla by přinést možnost vytvoření více logických topologií v jedné fyzické TRILL síti. To může přinést užitek při potřebě směrování určitých skupin rámců jiným způsobem, než jakým jsou směrovány ostatní rámce. Příkladem může být směrování skupiny rámců, která má větší prioritu nebo potřebuje zajistit větší bezpečnost. Pro takovou skupinu rámců je vytvořena oddělená logická topologie sítě a každý RBridge si vytvoří novou samostatnou směrovací tabulku pro tuto logickou topologii. Zatím ale není jasné, jakým způsobem bude v hlavičce TRILL rámce označena využitá logická topologie (v návrhu je zatím několik různých možností). (Hooda, 2014)

7 Model chování přepínačů podle protokolu TRILL

Protokol TRILL je v RBridge implementovaný zčásti softwarově a zčásti hardwarově. Poté, co jsou softwarově (s pomocí protokolu IS-IS) naplněny speciální paměti tabulkami s informacemi o cestách v síti, jsou přijaté rámce řízeny již pouze hardwarem, aby jejich rozesílání v síti bylo co nejrychlejší. V praxi RBridge rozesílá rámce rychlostí daných portů (např. 1 Gbps). Dochází pouze k malému zpoždění, než RBridge přijme celý rámec nebo pouze jeho hlavičku (metody Store and forward, Cut-through, Fragment-free), aby mohl rozhodnout, jakou cestou rámec odešle dále.

V dnešní době není dostupná žádná aplikace, která by uživateli názorně zobrazila užitečnost protokolu TRILL, případně by mu umožnila modelování nebo simulaci vývoje vlastní podnikové sítě, ve které by uživatel mohl postupně zaměňovat klasické přepínače za RBridge. K dispozici je pouze malé množství softwarových implementací protokolu (v operačním systému Solaris[5], Linux Implementation of TRILL Protocol[6]). Tyto implementace umí simulovat RBridge a počítač s jednou z těchto implementací je možné připojit do fyzické sítě namísto RBridge. Tyto aplikace ale nejsou určeny pro názornou ukázku chování RBridge nebo pro simulaci provozu sítě, ve které je více zařízení. Jedinou možností simulace větší sítě s využitím těchto aplikací je vzájemné propojení více počítačů (mohou být virtuální). Takové řešení ale rozhodně není ideální a uživatelsky přívětivé.

Praktickou částí této práce je vytvoření didaktického modelu, který by uživateli dokázal představit užitečnost protokolu TRILL a bylo by možné ho využít k modelování různých topologií počítačových sítí, které by si experimentátor mohl v průběhu modelování modifikovat. Vzhledem k nutnosti velkých investic do zakoupení RBridge je užitečnost modelu zřejmá. Každý síťový administrátor uvažující o nasazení protokolu TRILL si díky modelu může udělat přesnou představu o přínosu zakoupení například dvou, tří nebo dvaceti RBridge pro podnikovou síť. Cílem modelu naopak není testování výkonu nebo testování skutečné propustnosti sítě, protože k tomu by bylo nutné mít k dispozici fyzická síťová zařízení nebo velmi propracovaný simulátor, který není možné vytvořit v rozsahu této práce.

7.1 Metody zpracování projektu

Na fyzických zařízeních počítačové sítě byl vymezen systém, který zajišťuje přenos dat v síti pomocí protokolů STP a TRILL. Tento systém zahrnuje i části technologie Ethernet a protokolu IS-IS. V modelovaném systému jsou zanedbány časy potřebné k výpočtům a zpracování informací, proto se jedná o statický modelovaný systém. Všechny prvky v systému mohou být odstraněny, přesunuty nebo mohou být přidány další prvky v průběhu modelování, proto jsou všechny prvky temporární a mobilní. Všechny prvky modelu vznikají a zanikají uvnitř modelovaného systému a jsou tedy endogenní.

[5] https://docs.oracle.com/cd/E36784_01/html/E37516/rbridges.html [cit. 9. 4. 2015]
[6] http://wisnet.seecs.nust.edu.pk/projects/trill/index.html [cit. 9. 4. 2015]

Podle modelovaného systému byl vytvořen modelující systém, který je v této práci označován jako model (přestože tento pojem není přesný, jedná se o ustálené označení modelujícího systému). Model je diskrétní (evoluce je modelována po krocích) a jeho velmi důležitou součástí je on-line animace. Ta umožňuje sledování průběhu modelování a interaktivní zásahy experimentátora, které mohou průběh modelování ovlivnit.

Model byl po vytvoření verifikován proti předpokladům, které jsou popsány v teoretické části této práce a jsou součástí modelovaného systému. Při verifikaci byla velmi užitečná on-line animace, která přesně ukazuje stav modelu po krocích v průběhu modelování. V další fázi byl model validován nezávislými odborníky empirickou metodou a porovnáním proti provozu generovanému na fyzických zařízeních v několika různých topologiích.

7.1.1 Agentová architektura

„*Agent je zapouzdřený počítačový systém zasazený do nějakého okolí, který v něm pružně a autonomně působí za účelem plnění daného cíle.*“ (Kavička, 2014)

Všechna aktivní síťová zařízení se ve fyzické síti chovají podle paradigma autonomních agentů zasazených do počítačové sítě a jsou ze své podstaty reaktivní (po přijetí Ethernetového rámce vykonají nějakou akci). Pokročilejší síťová zařízení (přepínače, mosty, RBridge, směrovače) se po interakci s dalšími připojenými agenty učí topologii sítě, o které v první chvíli nemají moc informací (některé informace lze předem konfigurovat). Implementace různých protokolů v zařízení (např. CDP, STP, IS-IS atd.) způsobí, že se zařízení stává iniciativním agentem, který sám generuje provoz v počítačové síti (obvykle s cílem dozvědět se nové informace o topologii nebo o ní informovat ostatní zařízení).

Z těchto důvodů je model vytvořen pomocí agentově-orientované architektury. Jako základ pro vývoj modelu byl zvolen framework Repast Simphony 2.2.

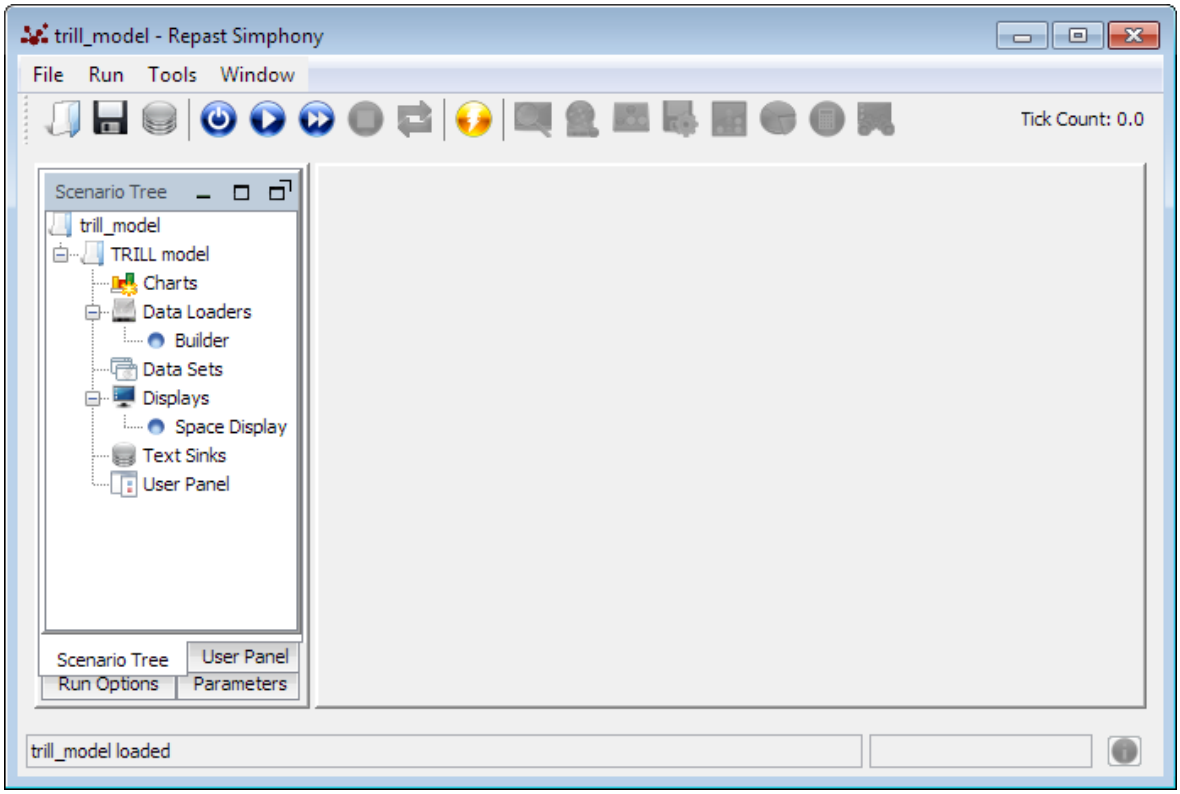
7.1.2 Framework Repast Simphony 2.2

Tento framework je specializovaný na agentově-orientované modelování a simulace. Je šířen pod licencí New BSD jako open-source. Pro vytvoření modelu byl vybrán, protože nabízí základ pro tvorbu jakéhokoli agentově-orientovaného modelu nebo simulace. Je naprogramován v jazyce Java a vývojové prostředí pro vytvoření projektů je založeno na upraveném prostředí Eclipse Kepler.

Základní okno pro vytvořené projekty je zobrazeno na následujícím obrázku (Obrázek 27). Pro vytvoření modelu nebyly využity všechny možnosti frameworku, ale velmi důležité jsou pro model následující vlastnosti:

- Ovládání kroků modelování – spuštění, zastavení, nastavení rychlosti, krokování, případně restart modelování.
- Možnost vložení více druhů agentů.
- Možnost vytvoření spojení mezi agenty a tím vytvoření grafu.

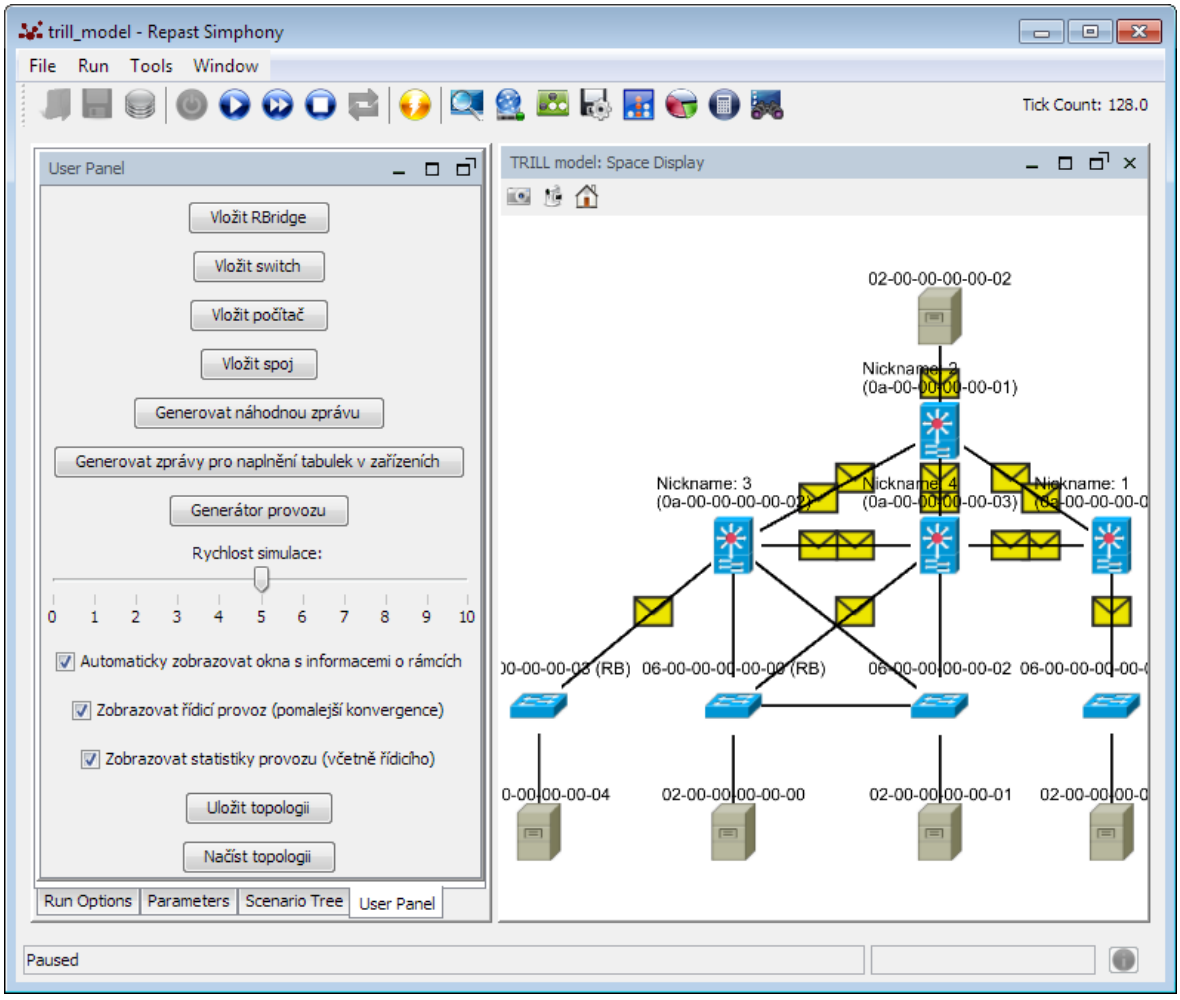
- Zobrazení modelování na pracovní ploše, která umožňuje zobrazení grafu s různými druhy agentů.
- Interakce agentů mezi sebou díky grafu nebo s využitím souřadnic, které určují umístění agentů na pracovní ploše.
- Plánování akcí, které agenti mohou vykonat v určitých okamžicích nebo intervalech.



Obrázek 27 – Okno prázdného projektu vytvořeného ve frameworku Repast Simphony

7.2 Základní struktura programu

Výsledný model po spuštění programu zobrazuje základní okno využitého frameworku. V horní liště je potřeba spustit modelování modrým tlačítkem pro spuštění. Po načtení programu je vidět základní model sítě (Obrázek 28). Hned po zapnutí modelu rozesílají všechny RBridge Hello rámce, aby rozeznali sousedy a mohly naplnit svou paměť informacemi o topologii. Levé menu programu nabízí uživateli možnost vložení nových zařízení a spojů do topologie, úpravu rychlosti modelování, generování zpráv v síti, nastavení zobrazování provozu a možnosti uložení a načtení jiných topologií. Všechny úpravy topologie lze provádět dynamicky v průběhu modelování, je tak možné průběžně sledovat změny provozu při přidání nebo odebrání zařízení a spojů mezi nimi.



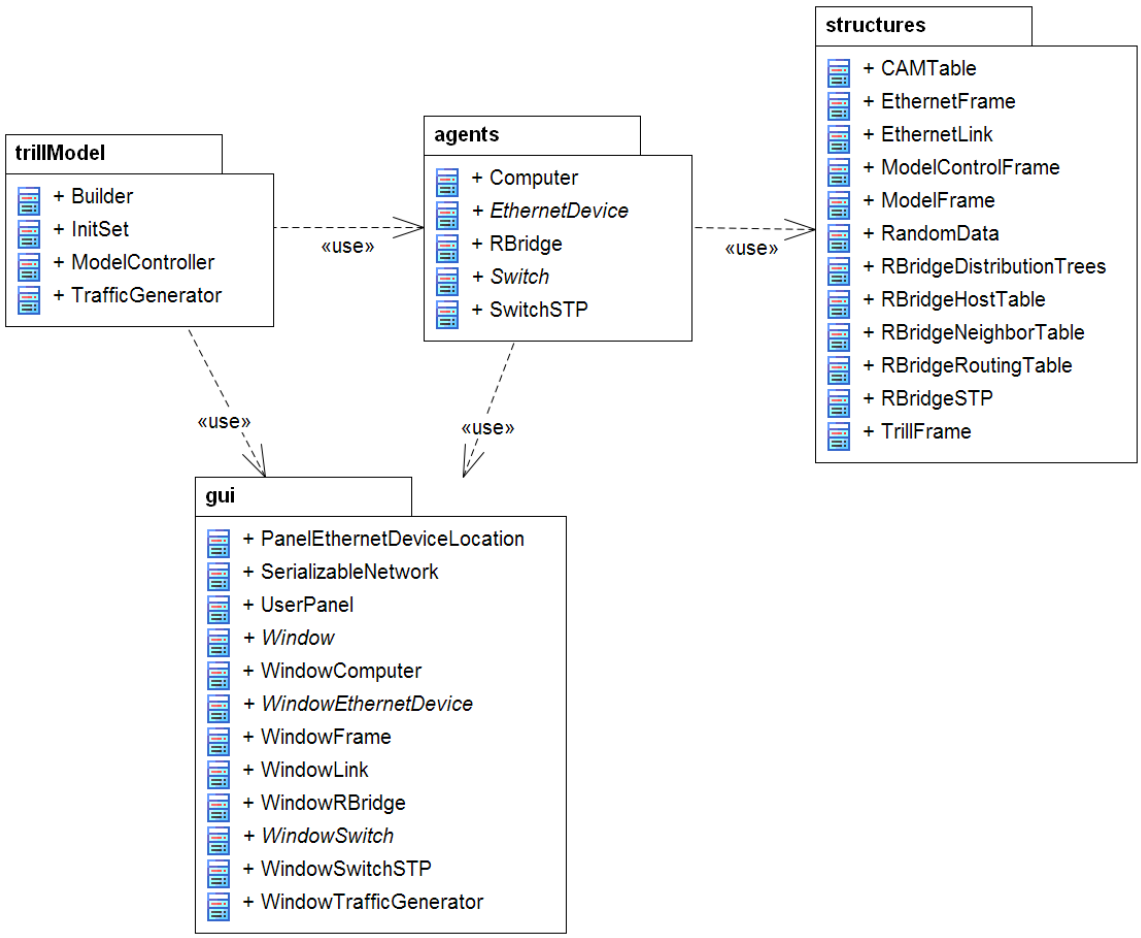
Obrázek 28 – Okno programu po spuštění modelování

Po dvojkliku na určité zařízení, spoj nebo zprávu se zobrazí okno s informacemi o daném objektu. Pokud to objekt umožňuje, lze v okně upravit jeho parametry, odeslat z objektu novou zprávu nebo objekt smazat. Okna jednotlivých zařízení mohou obsahovat několik záložek s různými informacemi, které dané zařízení využívá ke správné činnosti v síti. Provoz v modelu je rozlišen na řídicí rámce (označené žlutě) a datové rámce (označené bíle). Řídicí rámce slouží ke konvergenci sítě a jsou zasílané automaticky mezi síťovými prvky. Datové rámce představují užitečná data zasílaná mezi koncovými zařízeními. Obsah každé zprávy je možné zobrazit po dvojkliku na ikonu zprávy. Při průchodu zprávy TRILL sítí je v nově otevřeném okně vidět, jakým způsobem se obsah zprávy na každém RBridge mění.

Jak již bylo naznačeno, model je vytvořen v programovacím jazyce Java Standard Edition. Zdrojové soubory projektu jsou uloženy na přiloženém CD a jsou rozděleny do 4 balíčků, jejichž vztahy jsou zobrazeny i na obrázku (Obrázek 29):

- Agents obsahuje třídy s agenty, které definují jejich logiku.
- Gui obsahuje třídy použité pro grafické uživatelské rozhraní (okna, panel pro ovládání modelu, třídu potřebnou pro uložení topologie do souboru).

- Structures jsou třídy s datovými strukturami, které jsou využity agenty (zprávy a tabulky se sousedy, hosty, distribučními stromy apod.).
- TrillModel poskytuje třídy pro vytvoření a řízení modelování, které pracují např. s grafem a pracovní plochou, které jsou vytvořeny pomocí frameworku.



Obrázek 29 – UML diagram balíčků programu

V příloze této práce (Příloha A) je přiložen UML (Unified Modeling Language) diagram tříd, na kterém jsou znázorněny pouze nejdůležitější třídy modelu. Z balíčku trillModel jsou zahrnuty všechny třídy. Třída Builder vytváří prostředí pro modelování, ModelController využívá návrhový vzor Singleton a agentům nabízí možnost správy objektů v modelu, třída InitSet zajišťuje vytvoření základní topologie a zajišťuje pokračování modelování i v případě vymazání všech modelovaných objektů a třída TrafficGenerator se stará o nastavení a generování provozu, který slouží pro analýzu vlastností sítě.

V balíčku agents je celkově 5 tříd.

- EthernetDevice je abstraktní třída obsahující metody, které musí implementovat každé zařízení Ethernetu. Jsou to metody pro výpočet FCS a příjem Ethernetového rámce. Příjem rámců ale každé zařízení implementuje jinak, proto je tato metoda abstraktní.

- Computer představuje jednoduchý počítač, který může vytvářet a přijímat rámce. Třída je potomkem EthernetDevice a v modelu kromě osobního počítače nebo serveru může představovat i bránu sítě do internetu. Může tak omezeným způsobem nahradit směrovač, který obvykle v podnikových sítích plní funkci brány.
- Třída Switch také dědí od třídy EthernetDevice, ale zároveň je sama abstraktní. Implementuje metody jednoduchého přepínače, který podle CAM tabulky přepíná rámce v síti.
- SwitchSTP je potomkem třídy Switch a rozšiřuje ji o další metody, které modelují protokol STP. Díky tomu je zajištěna topologie sítě bez vzniku smyček. Lze tak modelovat i rozsáhlé sítě založené na klasických přepínačích se STP a názorně tak zobrazit jeho nevýhody.
- RBridge dědí opět od třídy Switch a přidává klasickému přepínači pokročilé funkce protokolu TRILL a díky tomu také zajišťuje topologii bez vzniku smyček, ale mnohem sofistikovanějším způsobem. Při propojení s agenty typu SwitchSTP musí RBridge naslouchat informace zasílané protokolem STP a přizpůsobit své chování těmto připojeným agentům, aby bylo zabráněno vzniku smyček i na rozhraní protokolů TRILL a STP.

Balíček structures obsahuje třídy s datovými strukturami, které jsou využity agenty a mají funkce tabulek CAM, hostů, sousedů, směrovací tabulky, tabulky s distribučními stromy a datové struktury s uloženými informacemi, které RBridge získává naslouchání STP. Všechny tyto datové struktury využívají pro vyhledávání hash klíčových hodnot (implementace pomocí hash map) stejně jako hash využívají skutečná zařízení při vyhledávání ve svých speciálních fyzických pamětích. V balíčku structures jsou navíc třídy, které slouží pro uchování informací o spojích mezi zařízeními a uchování obsahu Ethernetových rámců bez nebo včetně TRILL hlaviček. Pro zobrazení Ethernetového provozu v modelu jsou zde ještě dvě třídy, které umožňují zobrazit datový a řídicí rámec.

Balíček gui slouží pro vytvoření zobrazovacích prvků, ovládacích prvků, oken modelu a umožnění uložení a načtení topologií do souboru. Tento balíček neobsahuje žádnou řídicí logiku aplikace, proto zde nejsou popsány jeho konkrétní třídy. Kompletní přehled o balíčcích a všech třídách modelu je z důvodu rozsáhlosti uložen jako UML diagram tříd na přiloženém CD.

7.3 Implementované možnosti modelu

Model je zaměřený na představení možností protokolu TRILL a jeho výhod, které přináší proti protokolu STP. Pro dosažení těchto cílů model implementuje většinu vlastností protokolu TRILL včetně potřebného použití protokolu IS-IS a také základní vlastnosti protokolu STP.

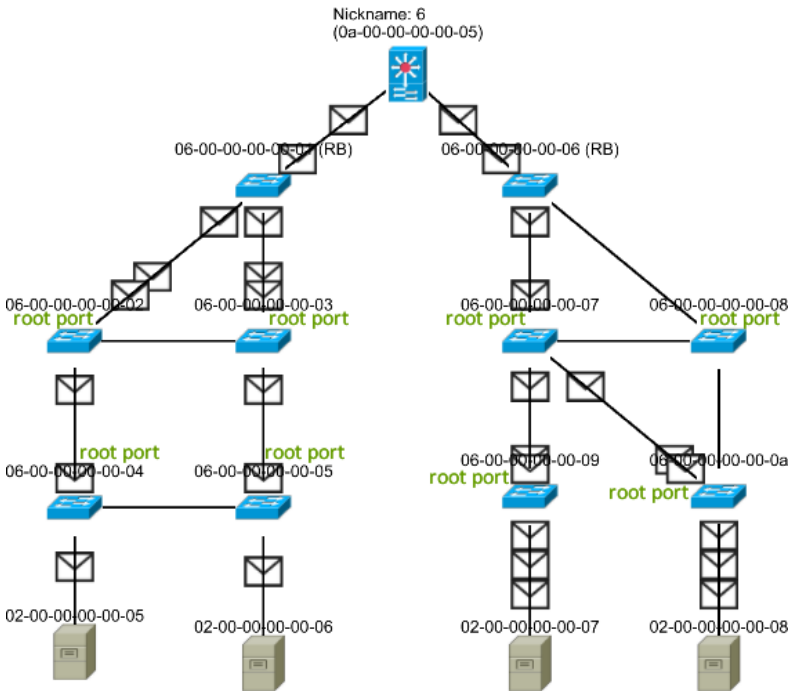
7.3.1 Zařízení modelované sítě a spoje mezi nimi

Všem zařízením (počítač, přepínač, RBridge), která lze v modelu využít, jsou přiděleny MAC adresy, které jsou podle standardu určeny pro lokální využití. Tyto adresy nejsou přiděleny žádnému výrobci Ethernetových zařízení a jsou vyčleněny pro využití v soukromých sítích, kde mohou být přiděleny např. virtuálním počítačům nebo virtuálním síťovým prvkům. (IEEE Std 802-2014, 2014)

Stejně jako lze ve fyzické počítačové síti propojit Ethernetovým kabelem všechny druhy zařízení, lze i v modelu vytvořit spoj mezi jakoukoli dvojicí zařízení. Každý spoj určuje možnosti rozesílání rámců v síti a ovlivňuje chování připojených zařízení.

7.3.2 Přepínače se STP

Protokol STP zajišťuje provoz sítě složené z klasických přepínačů bez vzniku smyček. Model implementuje nejdůležitější možnosti tohoto protokolu. Po spojení více přepínačů je podle nejnižší hodnoty MAC adresy vybrán RB přepínač. Ostatní přepínače postupně od RB volí své root porty, kterými mohou nejsnadněji dosáhnout RB a rozesílají přes něj velkou část provozu. Ostatní porty přepínačů se chovají jako designated. RB je vybrán v každé STP doméně, která je vždy ohraničena zařízeními vyšší vrstvy modelu OSI. V modelu tak může být více STP domén oddělených pomocí RBridge (Obrázek 30).

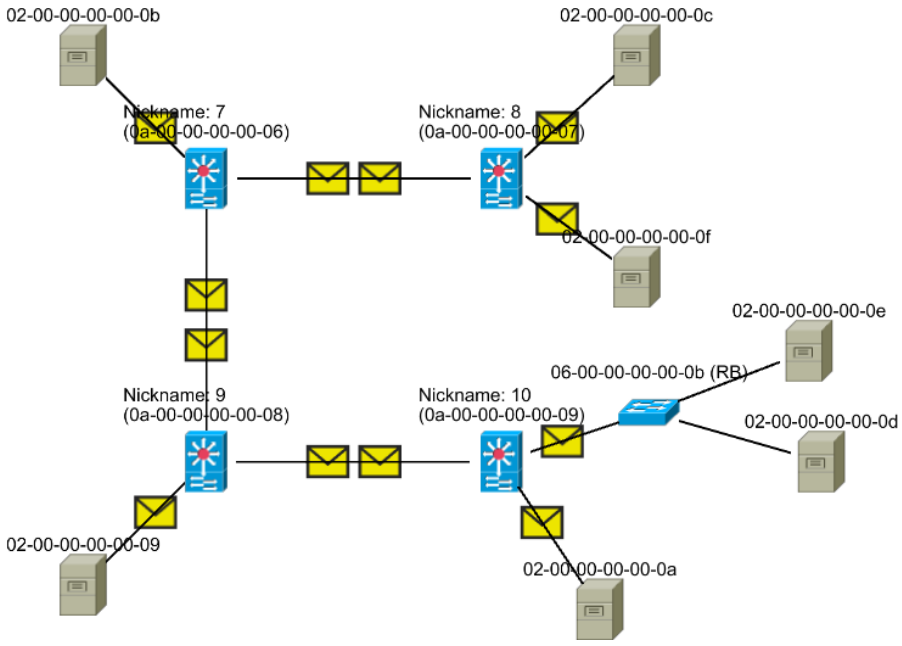


Obrázek 30 – Zvolení RB a root portů, provoz ve dvou STP doménách

Ke své funkci přepínače využívají CAM tabulku, kterou tvoří dvojice informací o MAC adrese zařízení a portu přepínače, přes který je zařízení v síti dostupné. Funkce STP jsou zajištěny uchováním informace o zvoleném RB a vybráním nejkratší cesty k RB z různých nabídek cest od okolních přepínačů. Po zvolení root portu přepínač nabízí svou cestu dalším přepínačům. Ty ji využijí v případě, že tato cesta je nejkratší i pro ně.

7.3.3 Vytvoření RBridge sousedství

RBridge stejně jako klasický přepínač využívá CAM tabulku, ale místo STP a jeho informací využívá postupy definované protokolem TRILL. Každý RBridge pravidelně přes své porty rozesílá skupinové rámce protokolu IS-IS (označované jako Hello rámce), které slouží ke vzájemnému rozpoznání RBridge v síti (Obrázek 31). Pokud si dva RBridge vymění tyto rámce, stávají se svými sousedy. Každý RBridge si vytváří tabulku sousedů, ve které si o sousedech udržuje všechny dostupné informace. Sousedé mohou být spojeny přímým spojem nebo spojením přes zařízení nižší vrstvy (např. přes přepínač nebo rozbočovač). Přijaté Hello rámce RBridge nerozesílá dál, sousedy se tak stávají jen RBridge, které jsou k sobě nejblíž. I po vytvoření sousedství jsou rozesílány Hello rámce pro udržení daného sousedství. Po určitém čase bez přijetí rámce od souseda vytvořené sousedství zaniká.



Obrázek 31 – Rozeslání Hello rámců pro objevení sousedních RBridge

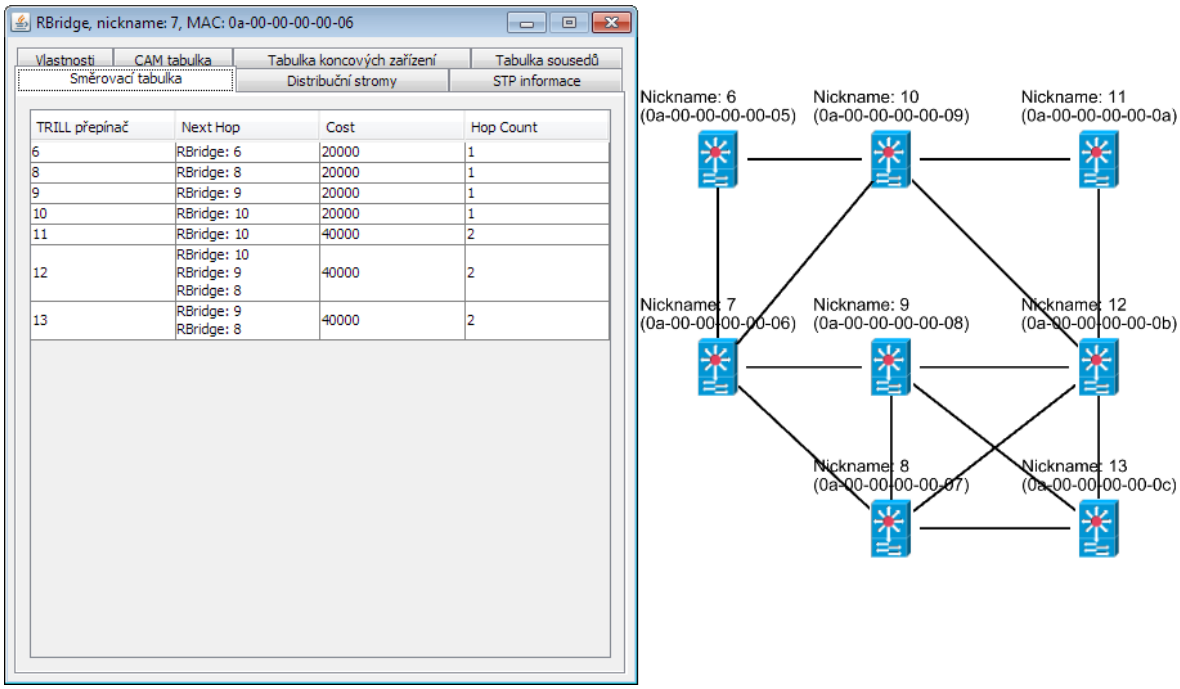
Vytvoření sousedství je základem pro další výměnu informací mezi RBridge. V reálné síti IS-IS protokol mezi sousedy zajišťuje výměnu informací o celé topologii a postupným přeposláním mezi jednotlivými sousedy se tak všechny RBridge v síti dozví informace o celé síti. Každý RBridge poté spouští vlastní výpočet cest. V modelu je výměna informací proti reálné síti zjednodušena. RBridge k vytvoření sousedství stačí přijetí Hello rámce (místo dvou nebo třícestného handshake postupu) a informace o topologii každý RBridge získává ze struktur uložených modelem (RBridge si v modelu nevyměňují své databáze).

Pokud RBridge zjistí změnu topologie (připojí se nový soused nebo se odpojí starý soused), zasílá řídicí rámec přes protokol IS-IS s informací o změně topologie všem RBridge v síti. Po přijetí této zprávy každý RBridge aktualizuje svou databázi a spustí nový výpočet cest v síti. Informace o změně rozesílá RBridge spolu se všemi informacemi potřebnými k výpočtům všemi porty s aktivovaným TRILL protokolem a sousední RBridge přeposílá tyto rámce stejným způsobem dál. Každý RBridge tak může stejné informace přijmout několikrát, ale podle identifikačního čísla rozezná, zda už tento rámec zpracoval a odeslal

ostatním. RBridge v modelu rozesílá informace stejným způsobem, ale v obsahu rozesílaného rámce je pouze identifikační číslo změny bez dalších informací o této změně. Bližší informace si ostatní RBridge zjišťují ze struktur modelu. I přes malé odlišnosti v chování modelovaných zařízení každý RBridge spouští vlastní výpočet, který na všech RBridge končí se stejným výsledkem a odpovídá výsledkům dosaženým v reálných RBridge.

7.3.4 Směrovací tabulka RBridge

Jedním z výsledků výpočtů každého RBridge je směrovací tabulka. Ta obsahuje jako cíl všechny RBridge v síti a informace, přes které RBridge jim zasílat rámce. Tabulka také obsahuje informace o kvalitě cest, kterou vyjadřuje její číselné ohodnocení. Další důležitou informací je počet RBridge, který se na cestě nachází. Toto číslo je vloženo do TRILL hlavičky při jejím vytvoření jako Hop count a je na každém průchozím RBridge snižováno, aby byl rámec zničen v případě, kdy nedorazí do cílového RBridge a místo toho bude zaslán do jiné části sítě (situace může nastat při změnách topologie). Ukázka topologie a obsahu směrovací tabulky RBridge je na následujícím obrázku (Obrázek 32).



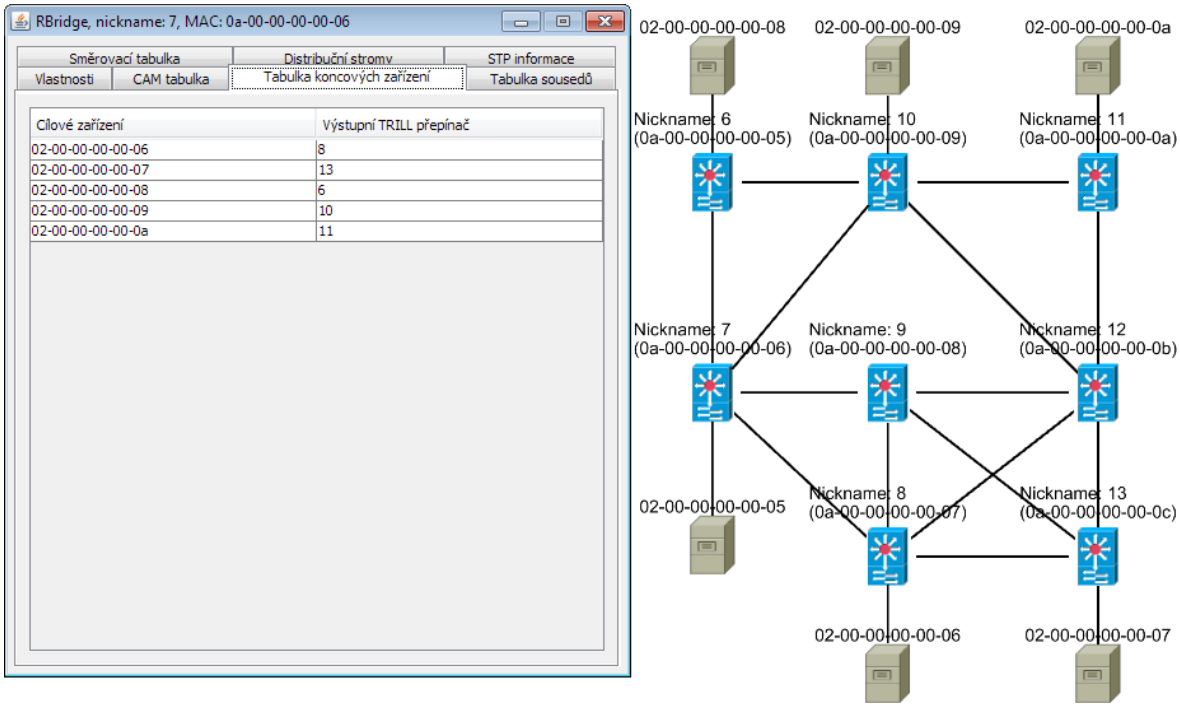
Obrázek 32 – Směrovací tabulka RBridge (nickname 7)

Cesty v síti jsou vypočítány Dijkstrovým algoritmem a umožňují ECMP, tedy rozložení zasílání rámců více cestami, pokud je jejich ohodnocení stejné. V tabulce tak může být více RBridge, které jsou označeny jako další na cestě k cíli. Rámce tak jsou zasílány postupně různými cestami. Po výběru cesty (tedy dalšího RBridge na cestě) je vybrán port pro odeslání podle informací z CAM tabulky podle MAC adresy sousedního RBridge.

7.3.5 Tabulka cílových zařízení

Další tabulka v modelovaném RBridge obsahuje informace o cílových zařízeních v síti stejně, jako jsou uloženy v reálném RBridge. Záznam v tabulce obsahuje MAC adresu

cílového zařízení a nickname cílového RBridge, přes který je cílové zařízení připojeno (Obrázek 33). RBridge v jádru sítě tyto informace znát nepotřebují, proto si je ukládají pouze RBridge, ke kterým jsou připojena jiná cílová zařízení. Výstupní RBridge před odstraněním TRILL hlavičky a odesláním původního rámce koncovému zařízení ukládá dvojici zdrojové MAC adresy ve vnitřním Ethernetovém rámci a nickname vstupního RBridge. Při zaslání rámce opačným směrem RBridge posílá TRILL rámec konkrétnímu výstupnímu RBridge místo toho, aby odesílal všesměrový rámec přes distribuční strom.

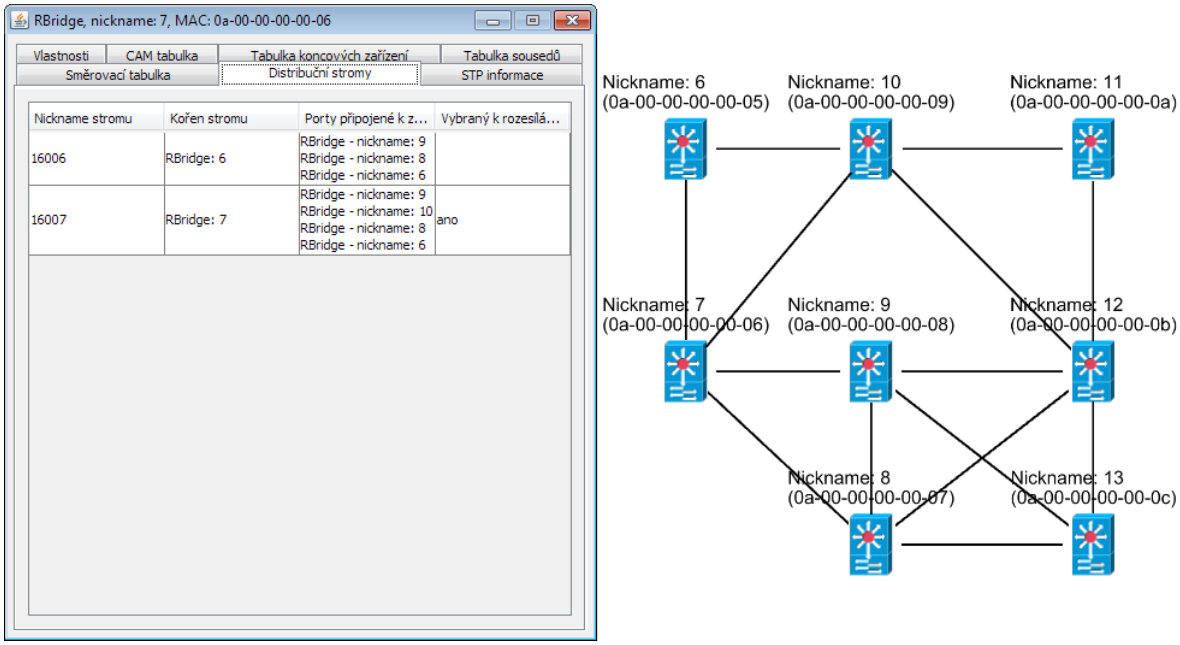


Obrázek 33 – Tabulka koncových zařízení RBridge (nickname 7)

7.3.6 Distribuční stromy

Stejně jako v reálné síti jsou v modelovaných RBridge počítány distribuční stromy používané při rozesílání skupinového a všesměrového provozu. Stejně jsou distribuční stromy použity pro rozeslání rámců s jedním cílem, ke kterému není nalezen záznam v tabulce cílových zařízení a není tak známa jeho pozice v síti. Pro každou propojenou topologii je v modelu vypočítán minimálně jeden distribuční strom. Další distribuční stromy mají kořeny na jednom ze sedmi RBridge. Všechny RBridge musí znát pouze kořenové zařízení distribučního stromu, ze kterého jsou Dijkstrovým algoritmem počítány nejkratší cesty ke všem ostatním zařízením. Výpočet cest probíhá stejně jako u ostatních tabulek na každém RBridge zvlášť.

Pro RBridge jsou důležité vlastní porty, které jsou součástí distribučního stromu (Obrázek 34). Ve chvíli, kdy jedním z těchto portů je přijat rámec, je zpracován a následně odeslán ostatními porty v distribučním stromu. Pokud je rámec vytvořen zařízením, je odeslán všemi porty distribučního stromu. Pokud zařízení není kořenem vlastního distribučního stromu, nemusí být rámce rozeslány nejkratšími cestami k ostatním RBridge, ale méně stromů v topologii usnadňuje výpočty při změnách topologie a umožňuje rychlejší konvergenci sítě.

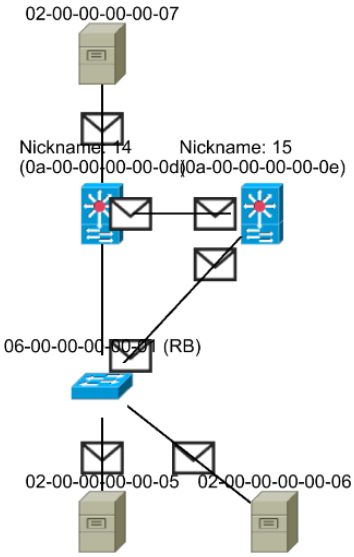


Obrázek 34 – Tabulka distribučních stromů RBridge (nickname 7)

7.3.7 Implementace STP v RBridge

RBridge ukončuje STP doménu, která je složena z klasických přepínačů. Pokud je RBridge zapojen mezi dva klasické přepínače, které mezi sebou neznají jinou cestu, rozděluje RBridge původní STP doménu na dvě STP domény. Takové zapojení do počítačové sítě přináší několik výhod. V případě změn nebo chyb v jedné STP doméně není žádným způsobem ovlivněna druhá doména. V menších doménách také dochází k rychlejší konvergenci STP. Propojení domén jedním spojem s sebou ale nese riziko přetížení spoje a v případě poruchy i výpadky spojení.

Při spojení domén přes více RBridge je nutná implementace dalších funkcí, které zabrání vícenásobnému šíření datových rámců mezi doménami. RBridge v tomto případě postupují podobným způsobem jako při připojení ke sdílenému médiu, kdy je z nich zvolen designated RBridge, který jediný může přeposílat provoz z jedné domény (v reálných zařízeních je provoz navíc rozdělen podle VLAN). RBridge navíc musí naslouchat řídicí provoz STP domén, aby v případě více připojení k jedné doméně nerozesílal provoz vícekrát. Vždy je nutné zasílat provoz pouze jedním portem, který je zvolen stejným způsobem, kterým klasické přepínače volí svůj root port. Ostatními porty RBridge nesmí přijímat ani odesílat žádné datové rámce. Žádný RBridge zároveň nesmí STP doménu ovlivňovat, proto nevytváří žádný STP řídicí provoz a pouze naslouchá informace vytvořené klasickými přepínači. Provoz STP domény tak vždy zajišťuje jeden RBridge svým jedním portem (Obrázek 35). Na ostatních portech připojených k doméně (včetně portů na ostatních RBridge) není žádný provoz generován a přijatý provoz je zrušen.

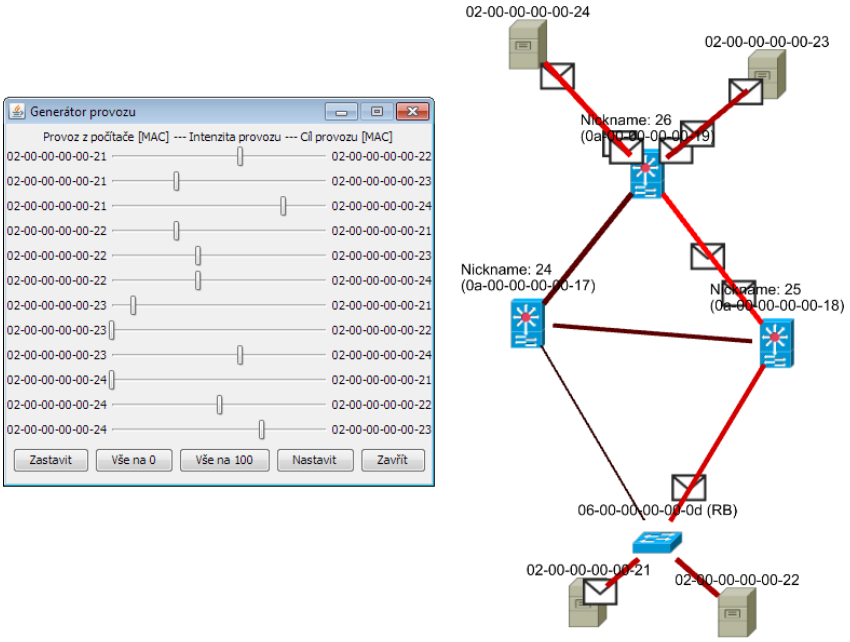


Obrázek 35 – Ukázka provozu přes designated RBridge (nickname 15)

Jak již bylo napsáno v předchozí kapitole, ve stádiu návrhu je rozšíření protokolu TRILL, které by mělo přinést možnost vytvoření pseudo uzlů s cílem vylepšit vlastnosti podobných připojení.

7.3.8 Generátor provozu a vytíženost spojů

Pro základní představu o provozu v modelované síti poskytuje model možnost relativního porovnání provozu na různých spojích v síti. Levé menu nabízí uživateli zvolit možnost zobrazení statistik provozu. Od chvíle zvolení této možnosti se počítá počet průchozích rámců všemi spoji modelu. Pro každý spoj je následně počítán poměr průchozích rámců proti nejvytíženějšímu spoji modelu. Spoje s relativním provozem do 25% jsou označeny slabými černými linkami, spoje s provozem nad 25% jsou označeny tlustšími linkami s odstíny červené barvy. Nejvytíženější spoj s relativním vytížením 100% je označen sytě červenou barvou. Po dvojkliku na daný spoj se v okně s informacemi o spoji zobrazuje počet procent relativního vytížení spoje.



Obrázek 36 – Generátor provozu a vytíženost spojů

Na obrázku (Obrázek 36) je zobrazeno okno generátoru provozu, kde lze nastavit relativní intenzitu generovaného provozu mezi jednotlivými počítači, a jednoduchá topologie sítě, na které je znázorněna intenzita provozu na jednotlivých spojích. Klasický přepínač zapojený ve spodní části sítě způsobuje, že provoz je směrován pomocí STP protokolu pouze přes spoj k RBridge s nickname 25. RBridge mezi sebou zasílají provoz nejkratší cestou, proto je využit hlavně spoj mezi RBridge 25 a 26. Další dva spoje mezi RBridge jsou využity pouze pro řídicí provoz a v tomto případě dosahuje jejich relativní vytížení 35% (řídicí provoz není zobrazován a generovaný provoz má nastavenu poměrně malou intenzitu). Při rovnoměrném rozložení provozu v síti by spoje měli stejnou tloušťku a červenou barvu.

Některé další možnosti, které vytvořený program nabízí, a bližší popis ovládání, jsou popsány v uživatelské příručce programu, která je k dispozici na přiloženém CD a v příloze této práce (Příloha B).

7.4 Ukázky zdrojových kódů

V rozsahu této práce nelze podrobně popsat všechny třídy a jejich odpovědnosti v programu. V přílohách tak je připojeno jen pár ukázek důležitých částí programu. První ukázka (Příloha C) představuje třídu TrillFrame sloužící jako datová struktura pro uchování polí TRILL hlavičky a dat vnitřního Ethernetového rámce. Datové typy atributů jsou zvoleny tak, aby pokud možno odpovídaly bitovým velikostem polí rámců zasílaných v reálných sítích. Stejným způsobem je vytvořena i třída EthernetFrame, která slouží pro uchování dat vnitřního i vnějšího Ethernetového rámce.

Další ukázka (Příloha D) obsahuje zdrojový kód metody, která zajišťuje příjem rámce v RBridge. Přijatý rámec je zpracován téměř stejným postupem jako v reálném RBridge. Nejdříve je zkontrolováno, zda má být rámec zpracován – podle FCS a portů blokovaných

protokolem STP. Pokud rámec není zrušen, ukládá si RBridge zdrojovou adresu rámce do CAM tabulky. Poté je podle hodnoty EtherType rozhodnuto, zda se jedná o TRILL rámec, případně jestli je skupinový. Podle obsahu tabulek v RBridge je rozhodnuto, zda bude rámec odeslán v původním tvaru nebo bude zpracován vstupním, páteřním, nebo výstupním TRILL procesem. Po vybrání procesu je rámec předán pro zpracování odpovídající metodě.

Část třídy, která zajišťuje funkci CAM tabulky, je přiložena v další příloze (Příloha E). V ukázce je zobrazena vnitřní třída, která slouží pro uchování jednotlivých záznamů v tabulce, a způsob, jakým jsou záznamy uloženy (HashMap s klíči, které představují MAC adresy zařízení).

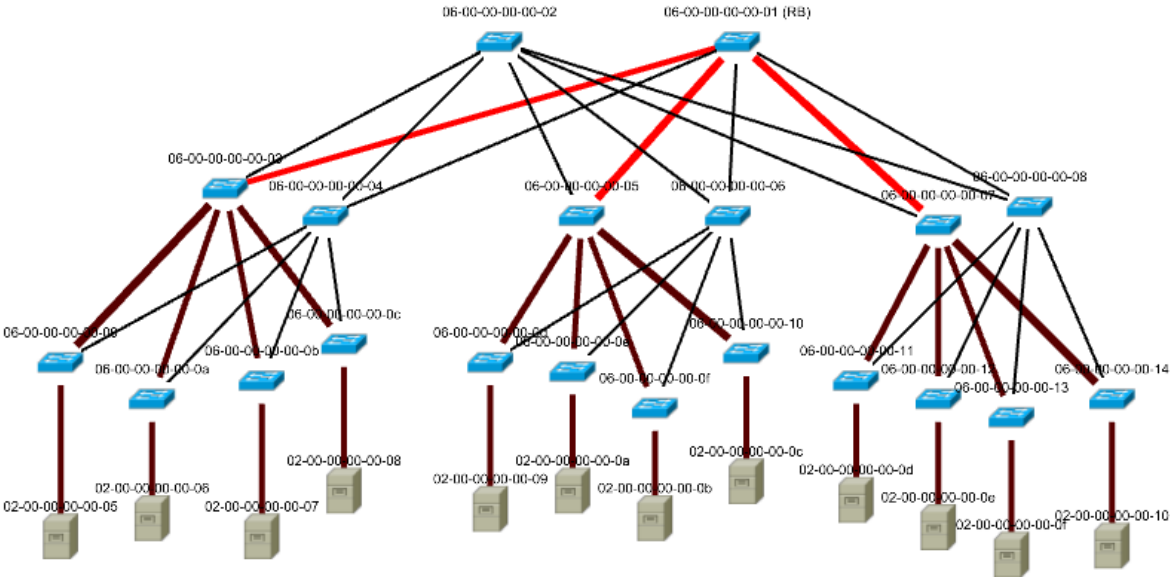
Poslední ukázka (Příloha F) je věnována dvěma metodám, které zajišťují pohyb zpráv na pracovní ploše modelu a jejich doručení do cílových zařízení. Z pohledu implementace je zpráva agentem (z pohledu modelování ale agentem není), proto v implementaci sama zajišťuje svůj pohyb a své doručení. V každém kroku počítá směr a délku pohybu podle umístění a vzdálenosti cílového zařízení. Po provedení pohybu je vypočítána nová vzdálenost k cílovému zařízení, a pokud je zpráva již tak blízko, že by se dalším krokem ocitla za zařízením, je doručena a je započítána do statistik spoje. V případě, že se jedná o řídicí rámec a zobrazení řídicího provozu je v modelu vypnuté, je využit postup rychlého doručení, který neřeší pohyb a vzdálenost k cílovému zařízení, ale zpráva je vždy doručena po třech krocích modelování.

7.5 Případová studie nasazení protokolu TRILL

Vytvořený model umožňuje poměrně snadné zpracování případových studií pro nasazení protokolu TRILL. Lze modelovat rozsáhlou topologii přepínačů se STP a postupně je nahrazovat pomocí RBridge. Pomocí generátoru provozu lze sledovat zatížení spojů a připravit tak několik variant pro nasazení protokolu TRILL do podnikové sítě.

7.5.1 Podniková síť s klasickými přepínači

V rozsáhlých podnikových sítích je velmi často vytvořena třívrstvá architektura síťových prvků, ve které jsou vrstvy: páteřní, agregační a přístupová. Přístupovou vrstvu ve většině případů tvoří síťové prvky druhé vrstvy (klasické přepínače). Agregační a páteřní vrstva může být tvořena přepínači nebo směrovači podle účelu, ke kterému má síť sloužit. Tyto vrstvy obsahují méně síťových prvků, které jsou daleko více vytížené, proto v nich jsou umístěny výkonnější zařízení. Pro případovou studii je využita třívrstvá podniková síť složená z klasických přepínačů, které tak tvoří rozsáhlou podsíť na druhé vrstvě modelu OSI. Autor (Bernstein, 2013) tuto strukturu sítě označuje jako tradiční především pro datová centra. Cílem případové studie je vytvořit několik variant nahrazení klasických přepínačů pomocí RBridge, aby byl optimalizován provoz v podnikové síti.

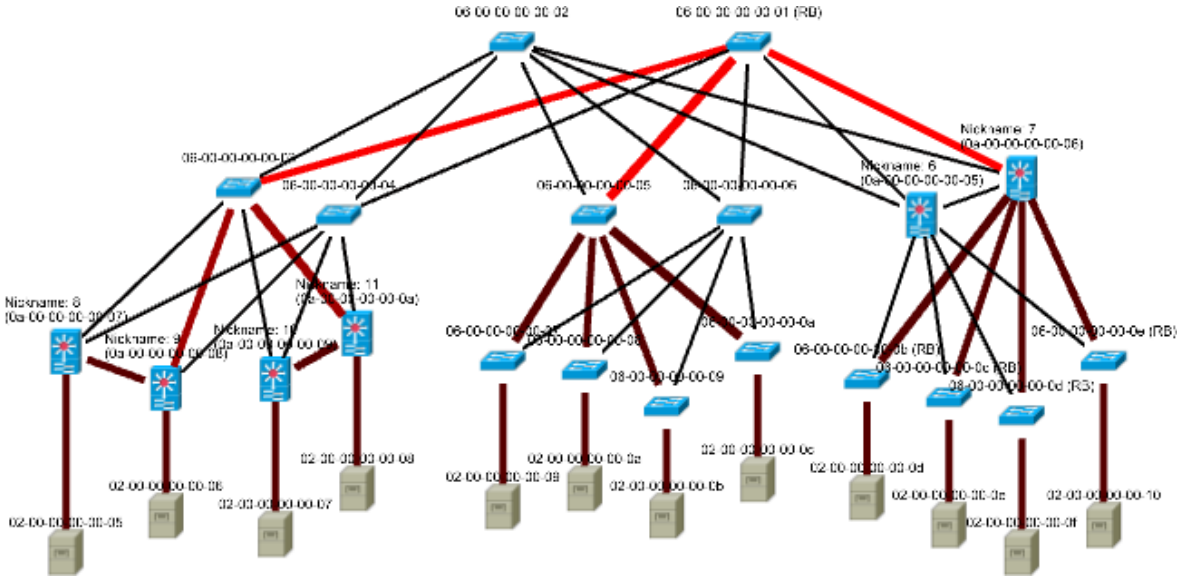


Obrázek 37 – Provoz v síti s třívrstvou architekturou s klasickými přepínači

Obrázek (Obrázek 37) znázorňuje podnikovou síť, která v páteřní vrstvě má dva, v agregační vrstvě šest a v přístupové vrstvě dvanáct klasických přepínačů. Ke každému přepínači v přístupové vrstvě je připojen jeden počítač. Ve fyzické síti mohou být místo jednoho počítače připojeny desítky počítačů nebo serverů. Počet počítačů ale neovlivní strukturu provozu v síti, proto je možné počet počítačů v modelu takto omezit. Obrázek je výstupem vytvořeného modelu a znázorňuje hustotu provozu v síti při využití jedné VLAN. Protokol STP v síti vytváří logickou stromovou topologii a rozesílá provoz pouze přes určité spoje a přepínače. Další přepínače v síti slouží jako záloha a jsou využity pouze při poruše aktivních spojů nebo přepínačů. Pouze pokročilé varianty STP mohou neaktivní spoje a přepínače využít pro přenos dat v dalších VLAN, ale konfiguraci a optimalizaci rozložení provozu v síti musí provést síťový administrátor. STP ale vždy využívá stromovou topologii a kořen stromu je potřeba v síti vhodně umístit. I po vytvoření nových spojů, například mezi přepínači v přístupové vrstvě topologie, nebudou tyto spoje protokolem STP využity a budou sloužit pouze jako záložní.

7.5.2 Ukázka neefektivního nasazení protokolu TRILL

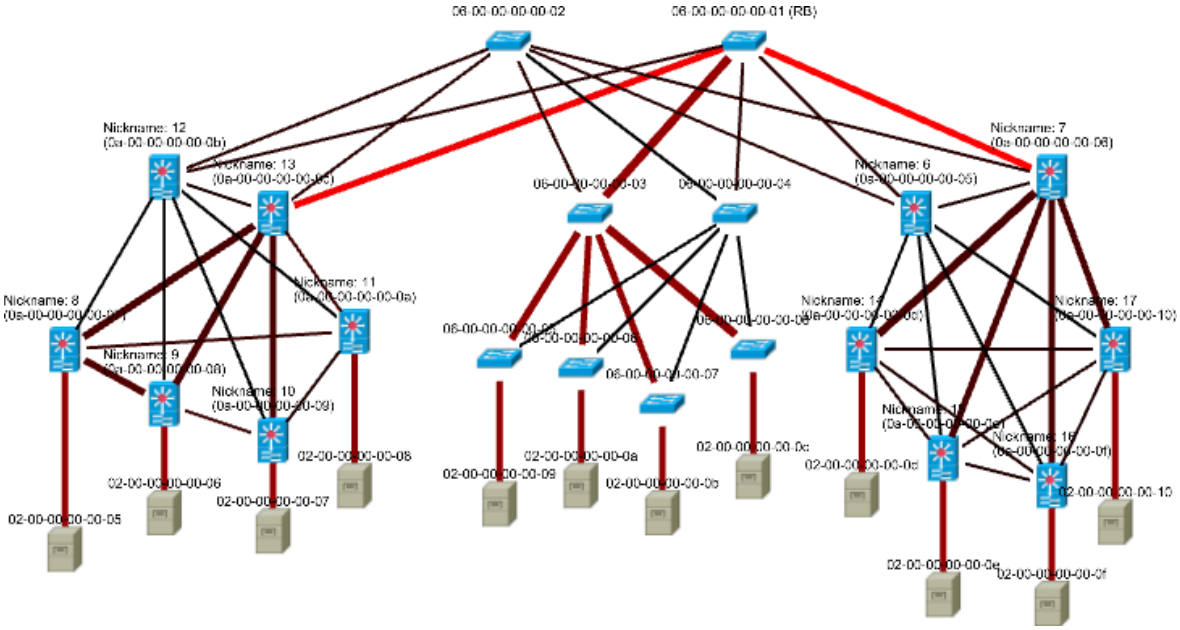
Nasazení protokolu TRILL do počítačové sítě je nutné promyslet. Na následujícím obrázku (Obrázek 38) jsou přepínače nahrazeny pomocí RBridge na dvou místech. V levé části sítě jsou nasazeny čtyři RBridge v přístupové vrstvě sítě. Každé dva mezi sebou zvolí designated RBridge, který zajišťuje provoz pro danou STP doménu. Tím je provoz soustředěn přes jedno zařízení a do páteřní sítě je přenášen pouze jedním spojem. Zde nasazení RBridge provozu spíše uškodilo. V pravé části sítě jsou nahrazeny dva RBridge v agregační vrstvě. Opět je z nich vybrán jeden designated RBridge (s vyšším nickname), který zajišťuje provoz STP domén. V tomto případě zůstává provoz stejný, jako v případě klasických přepínačů. V ostatních částech sítě je provoz řízen STP protokolem a zůstává tak stejný.



Obrázek 38 – Neefektivní umístění RBridge

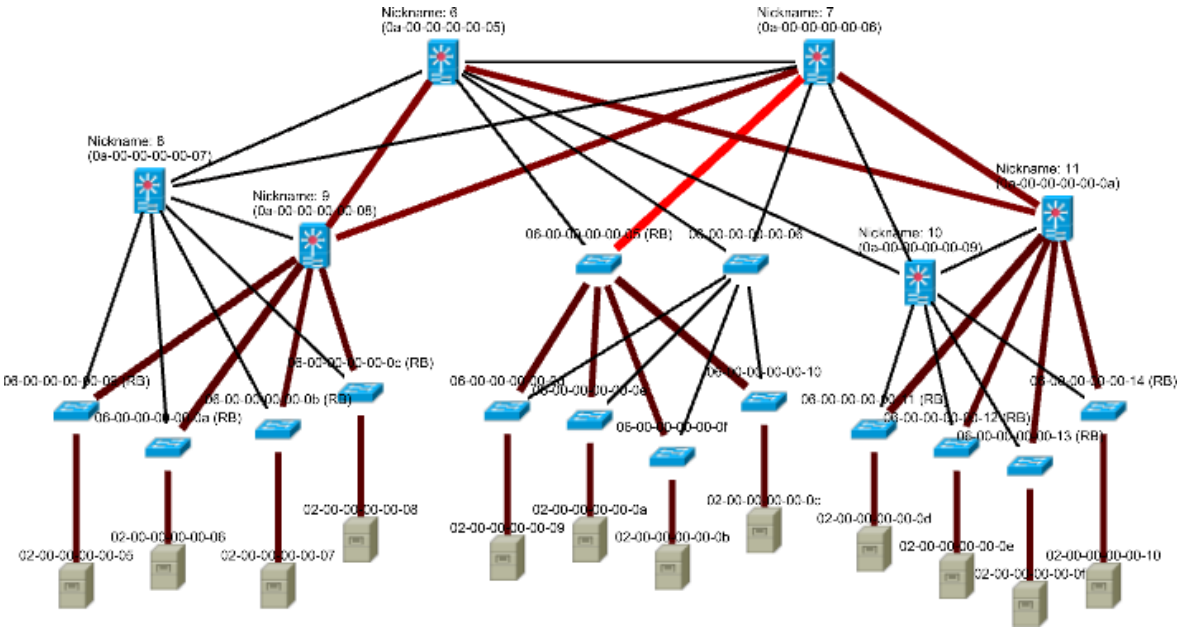
7.5.3 Varianty správného nasazení protokolu TRILL

Správné nasazení protokolu TRILL je závislé na povaze provozu v síti. Pokud je největší provoz v síti lokalizován v samostatných oblastech, které jsou propojeny přístupovou a agregační vrstvou, a přes páteřní vrstvu má provoz v síti menší hustotu, je výhodné nasadit TRILL v přístupové a agregační vrstvě. Takové nasazení je znázorněno na následujícím obrázku (Obrázek 39) včetně nových spojení mezi RBridge, které TRILL na rozdíl od STP dokáže využít. Každý počítač zde počítačům do své části sítě generuje pětkrát více provozu než počítačům, které jsou dostupné přes páteřní vrstvu sítě (v ostatních částech sítě je vždy ale více než dvakrát více počítačů). V prostřední části sítě jsou zachovány klasické přepínače, aby bylo patrné menší relativní zatížení linek mezi RBridge. Nejvytíženější spoje mezi RBridge mají relativní zatížení 29% a nejvytíženější spoje v prostřední části sítě 57%. To je dáno tím, že protokol TRILL nevyužívá pro provoz v jedné části sítě zařízení agregační vrstvy. Jak již bylo popsáno, STP by tato zařízení využil i v případě přímých spojů mezi přepínači v přístupové vrstvě sítě.



Obrázek 39 – TRILL mezi přístupovou a agregační vrstvou

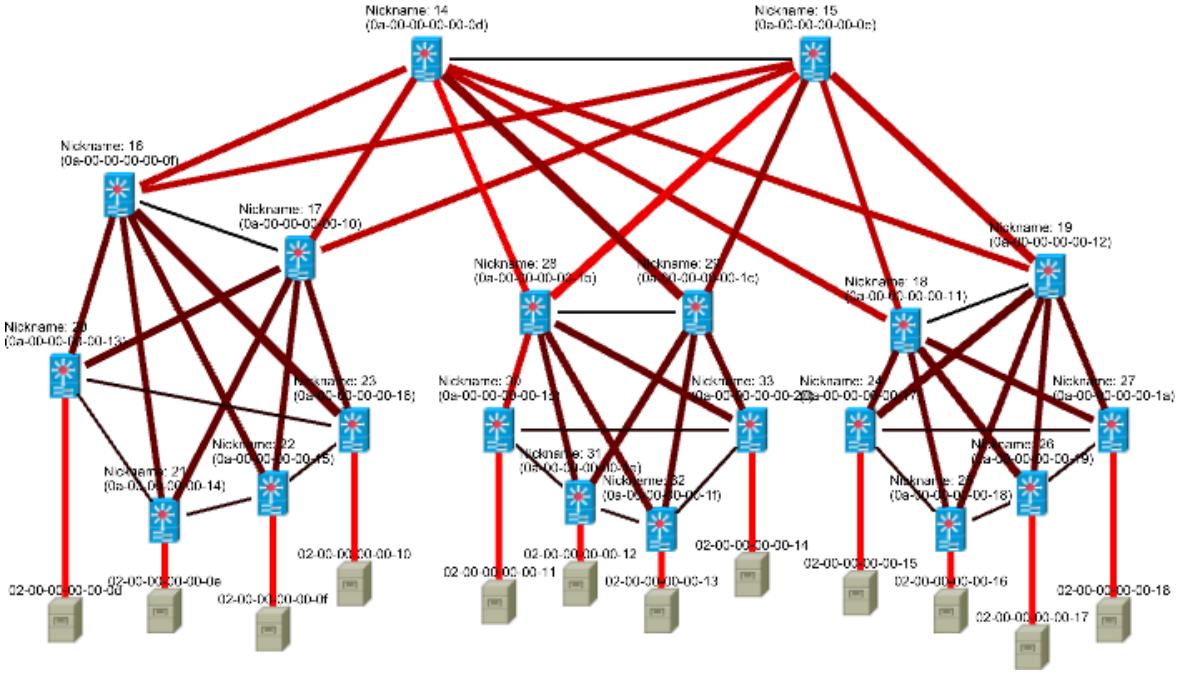
V případě hustého provozu přes páteřní část sítě je lepší nasazení protokolu TRILL mezi páteřní a agregační vrstvu sítě, jak je zobrazeno na dalším obrázku (Obrázek 40). V prostřední části sítě jsou pro názornost opět ponechány klasické přepínače. Mezi páteřní a agregační vrstvou klesá vytížení linek mezi RBridge díky ECMP na 50%. Ostatní provoz pro jednu VLAN zůstává stejný kvůli volbě designated RBridge. Využití RBridge 8 a 10 by řešilo zavedení pseudo uzlů. Tato možnost byla v této práci již popsána.



Obrázek 40 – TRILL mezi páteřní a agregační vrstvou

Pokud to dovoluje finanční rozpočet pro rekonstrukci sítě, je samozřejmě nejlepší nahradit všechny přepínače pomocí RBridge. Na dalším obrázku (Obrázek 41) je vidět rozložení zátěže v síti při výběru cest protokolem TRILL. Mírná odlišnost v hustotě provozu

mezi agregační a páteřní vrstvou sítě je pouze u části sítě v obrázku uprostřed. V této části sítě je totiž levý RBridge v přístupové vrstvě na rozdíl od ostatních spojen pouze s jedním RBridge agregační vrstvy. Nejsytější červená je v tomto případě mezi počítači a RBridge v přístupové vrstvě. Provoz je zde stejný jako v předchozích případech, ale provoz na jednotlivých linkách ve vyšších vrstvách topologie je rozložen natolik, že největší provoz je na spojích připojených k počítačům.



Obrázek 41 – TRILL v celé síti

7.6 Možnosti dalšího rozšíření modelu

Model je zaměřený na modelování chování protokolu TRILL. Ke správné funkci protokolu je nutné i využití technologií Ethernet, STP, IS-IS a dalších specifik síťových zařízení. Kompletním zpracováním všech těchto technologií by se model stal velmi rozsáhlou aplikací. Rozsah této práce není určen k vytvoření takto rozsáhlé aplikace, proto model přednostně implementuje možnosti, které jsou důležité pro komplexní modelování protokolu TRILL. Model umožňuje postupně programovat další možnosti.

Jednou z možností, kterou nabízejí reálné sítě a model ji nenabízí, je různá rychlost portů síťových zařízení a různá rychlost přenosu přes připojené kabely. Model nabízí pro všechny spoje pouze jednu stejnou rychlost přenosu dat. Další možností reálných sítí je multipoint propojení zařízení. Model umožňuje pouze častější point-to-point, tedy přímé propojení dvou zařízení, ale neumožňuje propojení přes sdílenou sběrnici nebo přes rozbočovač.

Z tohoto důvodu je implementace STP protokolu v přepínači omezena na zvolení root portu a ostatních portů, které se chovají jako designated. Implementace blocked portů by byla nutná pouze pro multipoint propojení. Blocked porty jsou ale v modelu implementovány na RBridge a jsou využity při připojení více portů (i různých RBridge) k jedné STP doméně.

Protože modelování STP není hlavním cílem práce, model nezobrazuje jeho řídicí provoz a STP komunikace mezi zařízeními probíhá na pozadí. Výběr RB pomocí STP probíhá stejným způsobem jako v reálné síti, ale model neumožňuje konfiguraci priority jednotlivých přepínačů, proto je RB vybrán vždy pouze podle nejnižší MAC adresy.

Protokol TRILL je v reálné síti plně funkční hned po zapojení zařízení. Stejným způsobem funguje i v modelu. Model nenabízí možnosti pro konfiguraci protokolu, ale v reálných zařízeních lze jeho konfiguraci upravit. Jedná se např. o možnosti konfigurace počtu vytvořených distribučních stromů a jejich kořenů. Protokol TRILL umožňuje pro rychlejší naučení adres cílových zařízení využít protokol ESADI, který není v modelu implementován a RBridge se tak učí adresy zařízení pouze z průchozího provozu. Model také neimplementuje proces RPF, který vyřazuje rámce přijaté na portech, na kterých nejsou očekávány. Podobný princip je v modelu využit pouze u blocked portů RBridge, které jsou vypočítány podle STP.

Všechny větší podnikové sítě by měly využívat možnost vytvoření VLAN, kterou model neumožňuje. Implementace VLAN problematiky s sebou kromě samotných VLAN nese další požadavky na umožnění trunk spojů a případně i možností VTP, FGL a pruningu.

Další možností rozšíření modelu by mohlo být přizpůsobení modelu pro simulaci a zobrazení reálného času v mikrosekundách. Model by se ale této vlastnosti musel přizpůsobit a při sledování průchozího provozu (při velkém zpomalení proti reálnému času) by téměř zanikl řídicí provoz zařízení. Vytvořený model kvůli didaktickému zobrazení zobrazuje řídicí provoz mnohem častěji proti ostatnímu provozu, než je běžné v reálných sítích.

Model nabízí pouze jednoduché zobrazení statistik průchozího provozu, které ukazuje relativní zatížení spojů v síti proti ostatním. V praxi by byla užitečnější důkladná statistická analýza provozu, pro kterou by ale model musel být rozšířen o další funkce, které by analýzu umožnily.

Závěr

V diplomové práci byly popsány základy přepínaných sítí a principy protokolů STP a TRILL. Zároveň byly představeny konkurenční technologie, které mají s protokolem TRILL stejné cíle. Následně byly analyzovány možnosti nasazení protokolu TRILL v podnikových sítích a v praktické části byl vytvořen interaktivní didaktický model chování protokolu, ve kterém byla provedena případová studie nasazení protokolu. Původní cíl vytvoření simulace byl v průběhu práce po dohodě s jejím vedoucím změněn na vytvoření didaktického modelu. Všechny cíle práce tak byly splněny.

Protokol TRILL je jednou z poměrně nových technologií, které jsou nasazovány v datových centrech a podnikových sítích, protože přináší mnohem lepší vlastnosti než STP, který byl dlouhou dobu jedinou možností pro řízení toku dat v přepínaných sítích. Význam protokolu TRILL bude pravděpodobně dále stoupat spolu s větší a levnější nabídkou zařízení, které ho implementují.

V průběhu psaní této práce bylo zjištěno, že modelováním a simulacemi protokolu TRILL se na světě zabývá pouze malé množství lidí a nikdo nenabízí model, který by byl podobný modelu vytvořenému v této práci.

Vytvořená aplikace implementuje určité vlastnosti technologie Ethernet a protokolů STP, TRILL a IS-IS a umožňuje interaktivní modelování libovolných topologií přepínaných počítačových sítí. Model je vytvořen pomocí agentově-orientované architektury a chování jednotlivých prvků modelu odpovídá chování počítačů, přepínačů a RBridge (přepínače implementující TRILL) v reálných počítačových sítích. Model zobrazuje obsah a cestu rámců v síti a obsah pamětí přepínačů, který se mění po jejich připojení do sítě a po rozeslání provozu. Při modelování počítačové sítě je vidět rozdíl v řízení provozu protokoly STP a TRILL, který v RBridge ukládá mnohem více informací o topologii sítě a pro výpočet cest v síti využívá protokol IS-IS a Dijkstrův algoritmus.

Model tak lze využít pro didaktické zobrazení chování síťových prvků, pro zobrazení výhod, které přináší protokol TRILL a jako pomůcku pro návrh nové topologie sítě. Generátor provozu a možnost sledování relativního zatížení jednotlivých linek umožňují experimentátorovi analyzovat provoz v modelu počítačové sítě a pomocí úprav topologie a nastavení generátoru může aplikace experimentátorovi pomoci navrhnout správnou topologii pro reálnou podnikovou síť.

Aplikace implementuje většinu důležitých vlastností protokolů, ale protože jejich problematika je velmi rozsáhlá, v poslední kapitole diplomové práce jsou rozepsány i možnosti dalšího rozšíření modelu. Ty obsahují například rozsáhlou problematiku virtuálních sítí, větší možnosti konfigurace přepínačů a RBridge, možnost rozšířit model na simulátor nebo vytvoření dalších nástrojů pro analýzu provozu v síti.

Literatura

AMAMOU, Ahmed, Kamel HADDADOU a Guy PUJOLLE, 2014. A TRILL-based multi-tenant data center network. In: *Computer Networks: The International Journal of Computer and Telecommunications Networking* [online]. [Cit. 2015-03-14]. Dostupné z: http://dx.doi.org/10.1016/j.comnet.2014.02.019

BERNSTEIN, Gary, 2013. New Switch Architectures and the Impact to the 40/100GbE Transition in the Data Center. In: *The Data Center Journal* [online]. [Cit. 2015-04-20]. Dostupné z: http://www.datacenterjournal.com/it/switch-architectures-impact-40100gbe-transition-data-center/

BOUŠKA, Petr, 2007. VLAN - Virtual Local Area Network. In: *SAMURAJ-cz.com* [online]. [Cit. 2015-02-28]. Dostupné z: http://www.samuraj-cz.com/clanek/vlan-virtual-local-area-network/

BROCADE, 2009. TRILL Brings a Thrill to Data Centers. In: *Brocade Advantage* [online]. [Cit. 2015-03-15]. Dostupné z: http://www.brocade.com/company/news-events/newsletters/BA1209/0912_technology_showcase.html

BROCADE, 2015. Brocade VCS Fabric Technology. In: *Brocade - Network Provider for Data Centers Everywhere* [online]. [Cit. 2015-03-05]. Dostupné z: http://www.brocade.com/solutions-technology/technology/vcs-technology/details.page

BUSINESSIT.CZ, 2013. SDN: Software-defined networking slibuje efektivnější sítě. In: *BusinessIT: Informační technologie pro profesionály* [online]. [Cit. 2015-03-05]. ISSN 1805-0522. Dostupné z: http://www.businessit.cz/cz/sdn-software-defined-networking-slibuje-efektivnejsi-site.php

COUDRON, Matthieu et al., 2013. Boosting Cloud Communications through a Crosslayer Multipath Protocol Architecture. In: *Future Networks and Services (SDN4FNS), 2013 IEEE SDN for* [online]. Trento: IEEE, s. 1-8 [cit. 2014-12-05]. Dostupné z: http://www.researchgate.net/profile/S_Secci/publication/261447758_Boosting_Cloud_Communications_through_a_Crosslayer_Multipath_Protocol_Architecture/links/53dac6ec0cf2e38c633978b7.pdf

DHANAGOPAL, Sudhakar, 2011. Converged Virtualized Data Center Networks – Reasons for Non-Deterministic Nature and Possible Solutions. In: *Advanced Networks and Telecommunication Systems (ANTS), 2011 IEEE 5th International Conference on*. Bangalore: IEEE, s. 1-5. ISBN 978-1-4673-0093-3.

HOODA, Sanjay, Shyam KAPADIA a Padmanabhan KRISHNAN, 2014. *Using TRILL, FabricPath, and VXLAN: Designing Massively Scalable Data Centers with Overlays*. Indianapolis: Cisco Press. 344 s. ISBN 1-58714-393-3.

HRNČIŘÍK, Matej, 2012. *Modelování L2 protokolů zajišťujících bezsmyčkovost* [online]. Brno. Diplomová práce. Vysoké učení technické v Brně, Fakulta informačních technologií [cit. 2014-12-07]. Dostupné z: https://nes.fit.vutbr.cz/ansa/uploads/Main/xhrnci00_dip_final.pdf

HUCABY, David, 2010. *CCNP SWITCH 642-813 Official Certification Guide*. Indianapolis: Cisco Press. 459 s. ISBN 1-58720-243-3.

IBANEZ, Guillermo a Elisa ROJAS, 2013. All-path bridging: Path exploration as an efficient alternative to path computation in bridging standards. In: *Communications Workshops (ICC), 2013 IEEE International Conference on*. Budapest: IEEE, s. 1280-1285.

IEEE STANDARDS ASSOCIATION, 2012. IEEE Approves New Ieee 802.1aq™ Shortest Path Bridging Standard. In: *IEEE Standards Association* [online]. [Cit. 2015-03-20]. Dostupné z: http://standards.ieee.org/news/2012/802.1aq.html

IEEE STD 802.3-2012, 2012. *IEEE Standard for Ethernet* [online]. The Institute of Electrical and Electronics Engineers, Inc. [Cit. 2015-04-27]. ISBN 973-07381-7312-2. Dostupné z: http://standards.ieee.org/about/get/802/802.3.html

IEEE STD 802-2014, 2014. *IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture* [online]. The Institute of Electrical and Electronics Engineers, Inc. [Cit. 2015-04-20]. ISBN 978-0-7381-9219-2. Dostupné z: http://standards.ieee.org/getieee802/download/802-2014.pdf

JUNIPER NETWORKS, 2015. QFabric System. In: *Juniper Networks* [online]. [Cit. 2015-03-04]. Dostupné z: http://www.juniper.net/us/en/products-services/switching/qfabric-system/

KAVIČKA, Antonín, 2014. *Pokročilé techniky modelování a simulace*. (přednáška) Pardubice: Univerzita Pardubice.

LU, Chunyang et al., 2013. Understanding the Overhead of Large Layer 2 Data Center Networking: Measurement and Analysis of TRILL as an Example. In: *Innovative Computing Technology (INTECH), 2013 Third International Conference on*. London: IEEE, s. 555-560. ISBN 978-1-4799-0047-3.

MATUŠKA, Miroslav, 2010. Seriál TRILL: Konečně náhrada za Spanning Tree?. In: *Lupa.cz* [online]. [Cit. 2015-03-04]. ISSN 1213-0702. Dostupné z: http://www.lupa.cz/serialy/trill/

MICROSOFT, 2005. Corporate Network. In: *Resources and Tools for IT Professionals | TechNet* [online]. [Cit. 2014-12-28]. Dostupné z: https://technet.microsoft.com/en-gb/library/cc782833%28v=ws.10%29.aspx

NETWORKED AND EMBEDDED SYSTEMS RESEARCH GROUP, 2012. *Project ANSA* [online]. Faculty of Information Technology, Brno University of Technology [Cit. 2014-12-05]. Dostupné z: https://nes.fit.vutbr.cz/ansa/

NOVELL, 2007. Novell's Networking Primer: Data Transmission. In: *NOVELL Worldwide* [online]. [Cit. 2015-02-16]. Dostupné z: https://www.novell.com/info/primer/prim05.html

OPEN NETWORKING FOUNDATION, 2015. *Open Networking Foundation* [online]. [Cit. 2015-03-05]. Dostupné z: https://www.opennetworking.org/

RFC 5556, 2009. *Transparent Interconnection of Lots of Links (TRILL): Problem and Applicability Statement* [online]. Network Working Group [Cit. 2015-03-10]. Dostupné z: https://tools.ietf.org/html/rfc5556

RFC 6325, 2011. *Routing Bridges (RBridges): Base Protocol Specification* [online]. Internet Engineering Task Force [Cit. 2015-03-06]. ISSN: 2070-1721. Dostupné z: https://tools.ietf.org/html/rfc6325

RFC 7172, 2014. *Transparent Interconnection of Lots of Links (TRILL): Fine-Grained Labeling* [online]. Internet Engineering Task Force [Cit. 2015-03-15]. ISSN: 2070-1721. Dostupné z: https://tools.ietf.org/html/rfc7172

RFC 7179, 2014. *Transparent Interconnection of Lots of Links (TRILL): Header Extension* [online]. Internet Engineering Task Force [Cit. 2015-03-12]. ISSN: 2070-1721. Dostupné z: https://tools.ietf.org/html/rfc7179

SCARFÒ, Antonio, 2011. The Evolution of Data Center Networking Technologies. In: *Data Compression, Communication, and Processing (CCP), 2011 First International Conference on* [online]. Palinuro: IEEE, s. 172-176 [cit. 2014-12-03]. ISBN 978-1-4577-1458-0. Dostupné z: http://www.ece.jhu.edu/~cooper/Oct2012/References/Data%20Centers/The%20evolution%20of%20Data%20Center%20networking%20technologies.pdf

SELGA, Josep M., Augustin ZABALLOS a Joan NAVARRO, 2013. Solutions to the Computer Networking Challenges of the Distribution Smart Grid. In: *Communications Letters, IEEE, 17.3* [online]. IEEE, s. 588-591 [cit. 2014-12-05]. ISSN 1089-7798. Dostupné z: http://www.researchgate.net/profile/Joan_Navarro/publication/246546014_Solutions_to_the_Computer_Networking_Challenges_of_the_Distribution_Smart_Grid/links/00b7d51d9ac5605d71000000.pdf

TANENBAUM, Andrew S. a David J. WETHERALL, 2011. *Computer networks*. 5th ed. Boston: Pearson Prentice Hall. 933 s. ISBN 0-13-212695-8.

TELECOMMUNICATION STANDARDIZATION SECTOR OF ITU, 1994. ITU-T Recommendation X.200: Information Technology – Open Systems Interconnection – Basic Reference Model: The Basic Model. In: *ITU: Committed to connecting the world* [online]. [Cit. 2015-02-16]. Dostupné z: https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-X.200-199407-I!!PDF-E&type=items

TELECOMMUNICATION STANDARDIZATION SECTOR OF ITU, 1997. ITU-T Recommendation Q.921: ISDN user-network interface – Data link layer. In: *ITU: Committed to connecting the world* [online]. [Cit. 2015-01-15]. Dostupné z: https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-Q.921-199709-I!!PDF-E&type=items

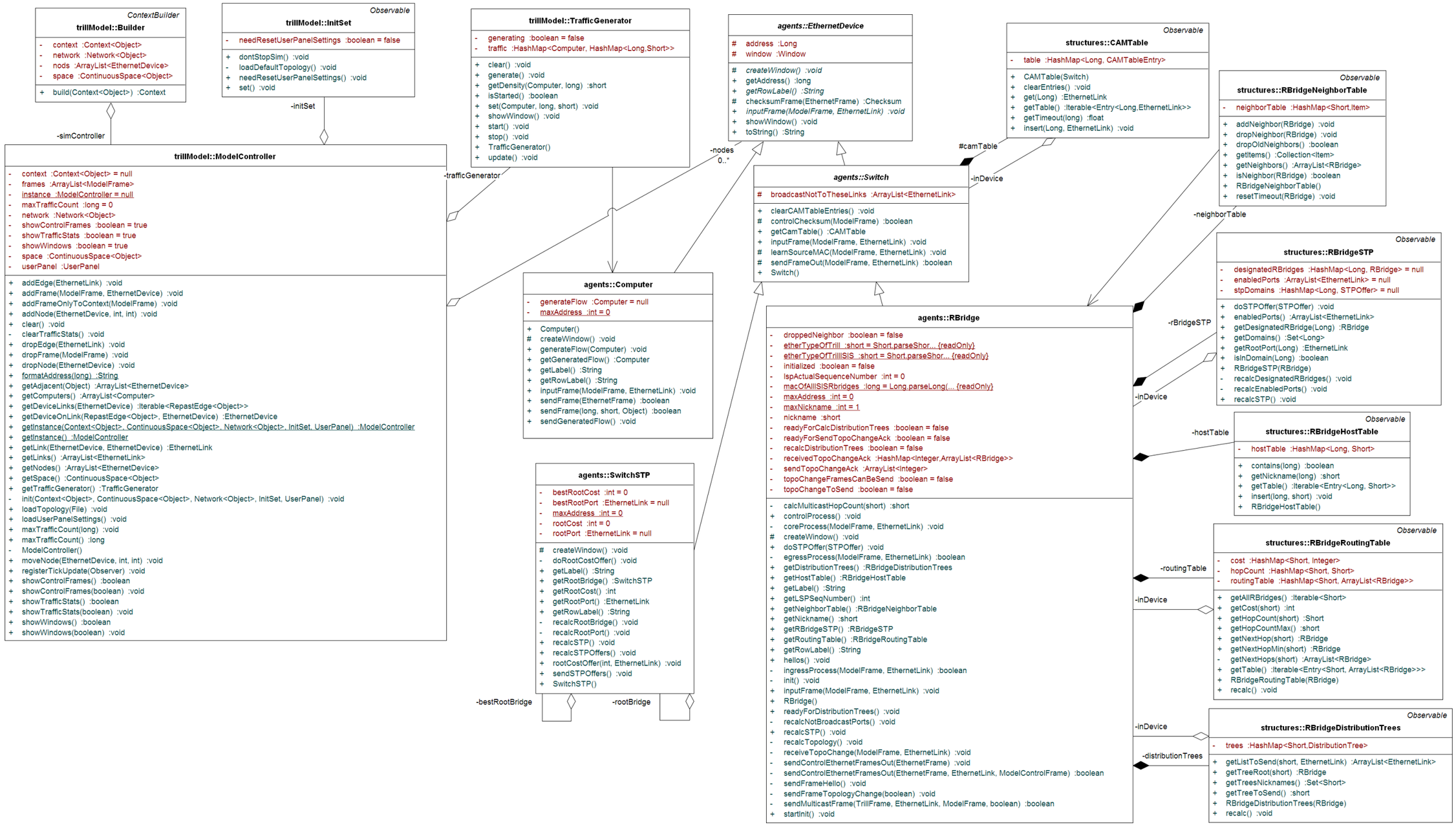
VESELÝ, Vladimír a Miroslav ŠVÉDA, 2012. L2 protocols in OMNeT++. *IP Networking 1 -- Theory and Practice.* Žilina: Žilina University Publisher, s. 37-40. ISBN 978-80-554-

0494-3. Dostupné z:
http://www.researchgate.net/publication/236889962_L2_protocols_in_OMNeT

WEIGUO, Hao, 2012. Data Center High-Speed Bus: TRILL-based Large Layer 2 Network Solution. In: *Better Connected World: Huawei Enterprise ICT Solutions* [online]. [Cit. 2015-03-12]. Dostupné z:
http://enterprise.huawei.com/ilink/enenterprise/download/hw_198092&ei=vn8bvzquiot4ygpvjykida&usg=afqjcnflpzwvzugmsy1ferr96x6babdwsg&sig2=ai2-rkmwcvt36dggv6-qhg&bvm=bv.87920726,d.bgq&cad=rja

ZLOCH, Tomáš, 2014. SDN – softwarově definované sítě. In: *SystemOnLine* [online]. [Cit. 2015-03-05]. ISSN 1802-615X. Dostupné z: http://www.systemonline.cz/clanky/sdn-softwarove-definovane-site.htm

Příloha A – UML diagram nejdůležitějších tříd programu



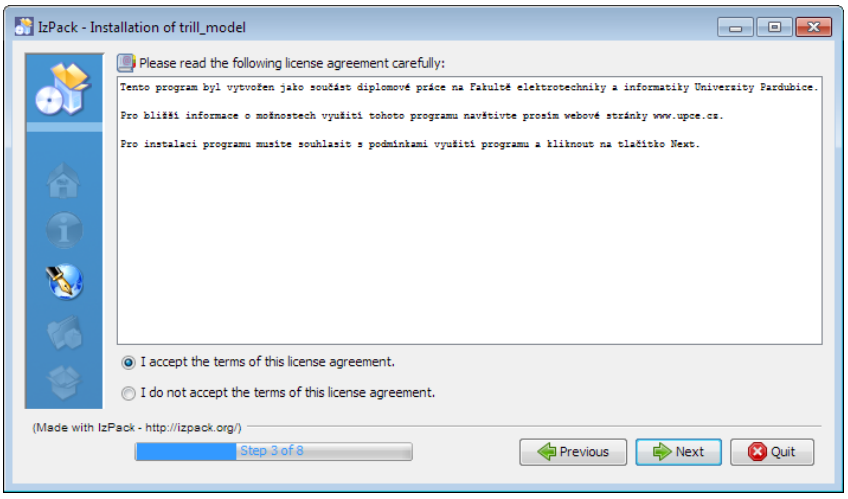
Příloha B – Uživatelská příručka k programu

Tato krátká příručka slouží k popisu ovládání programu vytvořeného v rámci diplomové práce s názvem „Analýza využití protokolu TRILL v podnikové síti“. Program umožňuje modelování libovolné topologie přepínané počítačové sítě využívající protokoly STP a TRILL a je blíže popsán v dané práci. Možnosti programu, které popisuje diplomová práce, nejsou v této příručce znovu opakovány.

Instalace programu

Na CD přiloženém k diplomové práci je uložen instalační soubor „instalace.jar“. K jeho spuštění je nutné mít v počítači nainstalovanou Javu SE verze 7 nebo vyšší. Pokud uživatel chce instalovat program do systémových adresářů Windows (např. Program files), je nutné, aby Java měla přidělena administrátorská práva (pro tuto možnost je nejjednodušší spustit instalaci přes příkazový řádek). Pro instalaci do jiných adresářů není potřeba žádný zvláštní postup a instalaci lze spustit běžným způsobem.

Po spuštění instalačního souboru je zobrazeno okno, které uživatele provede instalací, jaká je u počítačových programů obvyklá (Obrázek 1). Průvodce má několik kroků, je potřeba odsouhlasit licenční podmínky, nastavit cílový adresář pro instalaci a zvolit možné volitelné součásti instalace (zdrojové kódy a dokumentaci), které ale pro funkci modelu nejsou potřeba. Další možností instalace je vytvoření zástupců programu v nabídce start.



Obrázek 1 – Instalační průvodce programu

Projekt Repast Simphony

Na CD přiloženém k diplomové práci jsou kromě instalačního souboru uloženy i zdrojové soubory včetně kompletního projektu, který je vytvořen ve vývojovém prostředí Repast Simphony 2.2. Po instalaci vývojového prostředí, které je ke stažení zdarma dostupné na webových stránkách http://repast.sourceforge.net, v něm lze projekt otevřít a případně upravovat.

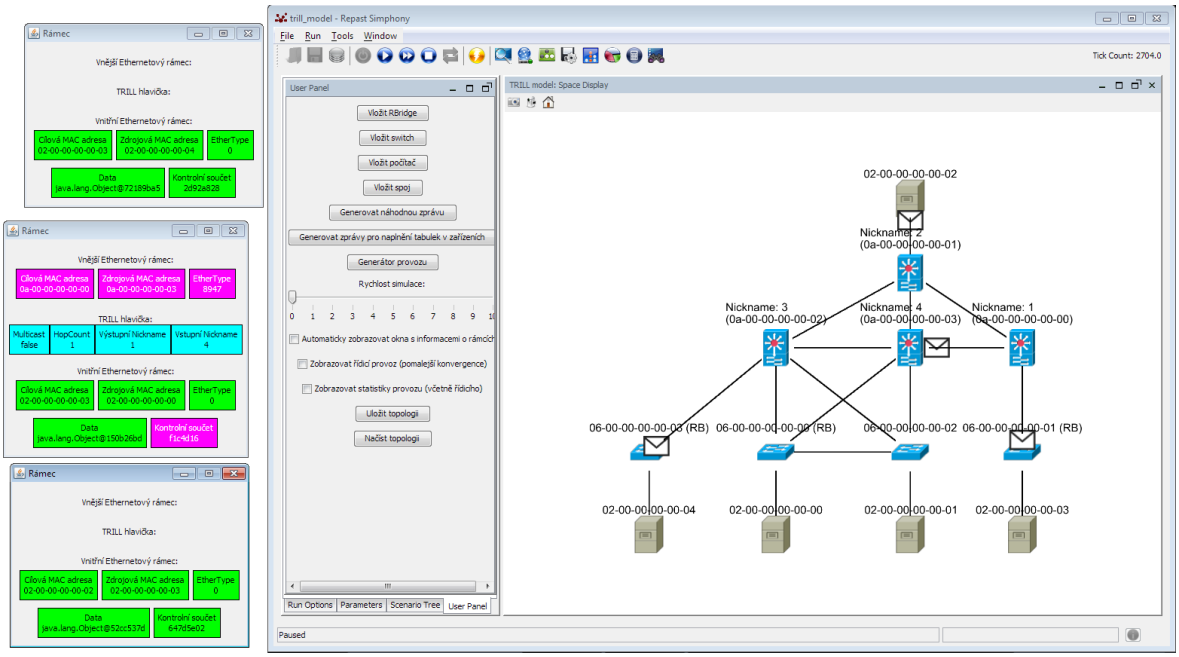
Spuštění programu

Po nainstalování lze program spustit souborem „start_model.bat“ (spuštění může trvat i několik desítek sekund). Případně lze program spustit i přímo ve vývojovém prostředí Repast Simphony po otevření projektu. Po spuštění programu je zobrazeno okno s ovládacími prvky modelu. Modelování je potřeba spustit modrým spouštěcím tlačítkem v horní liště. Poté je zobrazen základní model sítě s několika RBridge, přepínači a počítači.

Základní tipy pro ovládání programu

Prostor pro vytvoření modelu je ohraničen modrou čarou. Základní pohyb po pracovní ploše je umožněn pohybem myši při stisku jejího pravého tlačítka. Oddálení a přiblížení plochy je možné kolečkem myši nebo současným stisknutím klávesy shift a pravého tlačítka myši spolu s pohybem myši nahoru a dolů.

Nově otevíraná okna programu se zobrazují vždy v levé části monitoru. Proto je pro pohodlnou práci dobré umístit hlavní okno programu k pravému okraji obrazovky a roztáhnout ho maximálně na tři čtvrtiny šířky monitoru (Obrázek 2), aby nová okna nepřekrývala ovládací prvky a pracovní plochu programu.

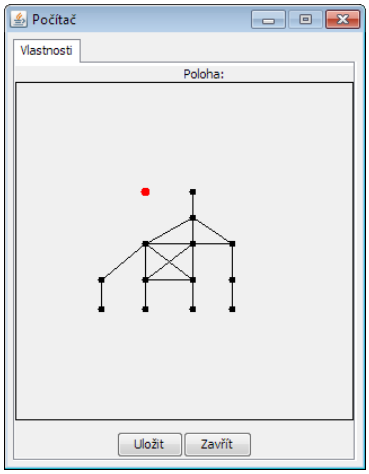


Obrázek 2 – Zobrazení nových oken programu

Program obsahuje několik ukázkových topologií, které lze načíst po kliknutí na tlačítko „načíst topologii“ v levém menu. Při načítání topologií musí být spuštěno modelování (při pauze se topologie nenačte). Topologie nazvaná clear je prázdná. Jejím načtením tak lze vyčistit pracovní plochu a připravit ji pro vytvoření nové topologie. Topologie s názvem začínajícím na cs_01 byly využity pro případovou studii v diplomové práci.

Práce se síťovými prvky

Každý síťový prvek (RBridge, přepínač, počítač, spoj) lze vytvořit pomocí tlačítka v levém menu. Spoje lze vytvořit pomocí volby dvou prvků, které mají být propojeny. Každý počítač může mít pouze jeden spoj do sítě, ostatní prvky mohou mít spojů neomezeně. Všechny prvky kromě spojů pak lze umístit kamkoli na pracovní plochu pomocí panelu pro umístění prvku, kde lze jedním klikem označit požadované umístění (Obrázek 3).



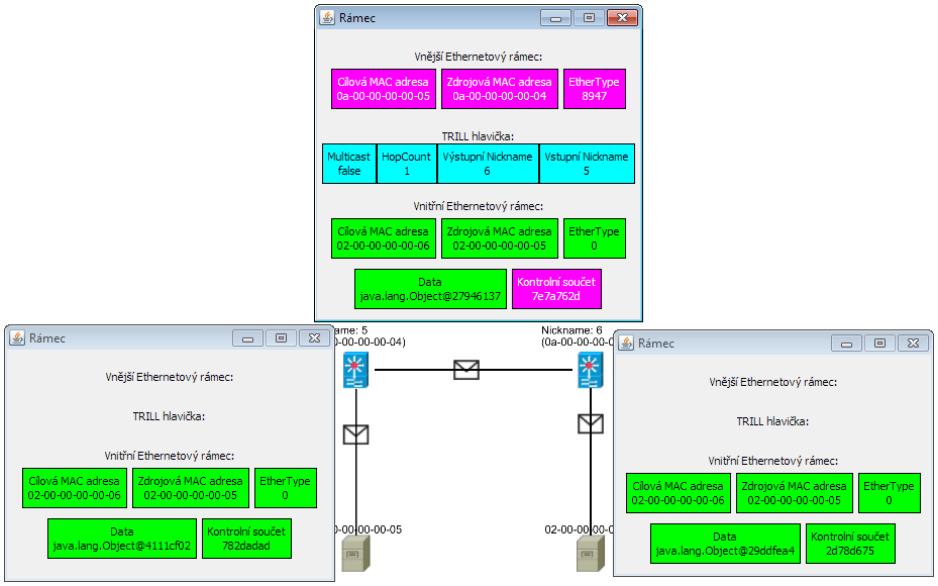
Obrázek 3 – Výběr polohy prvku na pracovní ploše

Po dvojkliku na prvek v topologii se otevře okno s možnostmi prvek upravit, přesunout nebo smazat. Při dvojkliku na spoj je dobré mít pohled na topologii více zvětšený, při oddálení pracovní plochy je potřeba spoj myší zaměřit velmi přesně, aby bylo okno zobrazeno. Dvojklik na ostatní prvky je jednodušší, protože jsou větší a není tak potřeba tolik přesné zaměření myší.

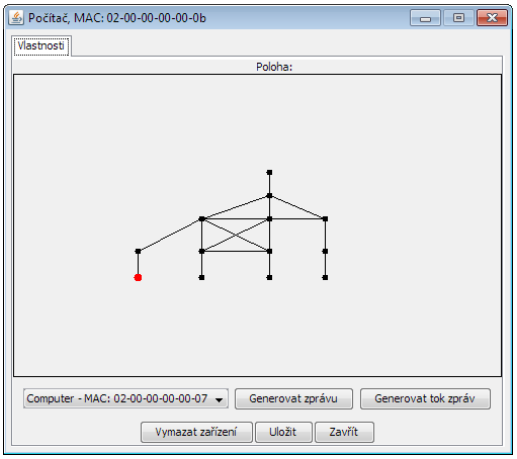
Objekty RBridge, přepínač, počítač a zpráva mají určité vlastnosti, které lze v otevřeném okně zobrazit nebo změnit. Okno zprávy zobrazuje jeho obsah – tím je buď jednoduchý Ethernetový rámec nebo TRILL rámec. Obsah rámce se při průchodu sítí může měnit (Obrázek 4).

Okno každého počítače (Obrázek 5) nabízí kromě jeho přesunu nebo vymazání ještě možnost generování rámců, které jsou zaslány jiným počítačům v síti. Vždy lze vygenerovat jen jeden rámec (po jeho odeslání se okno zavře) nebo tok rámců, který lze využít např. pro sledování ECMP provozu.

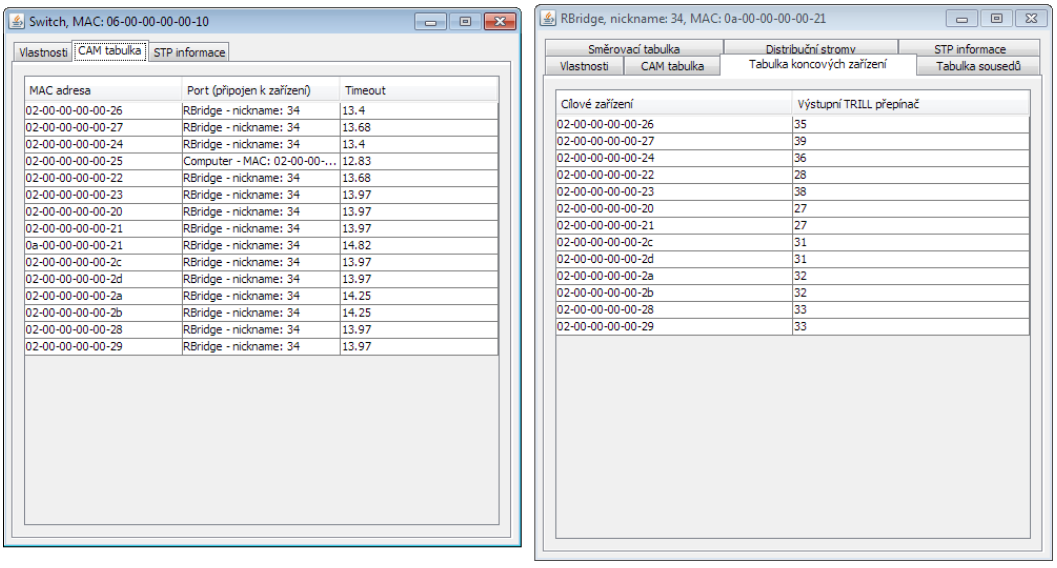
Okna přepínačů a RBridge nabízí mnohem více informací. Na dalším obrázku (Obrázek 6) jsou zobrazena okna s informacemi přepínače i RBridge. Okna mají několik záložek, díky kterým lze kromě obecných vlastností zobrazit informace z CAM tabulky, protokolu STP, směrovací tabulky, skupinové směrovací tabulky, tabulky koncových zařízení a tabulky sousedů.



Obrázek 4 – Zobrazení obsahu rámců při průchodu sítí



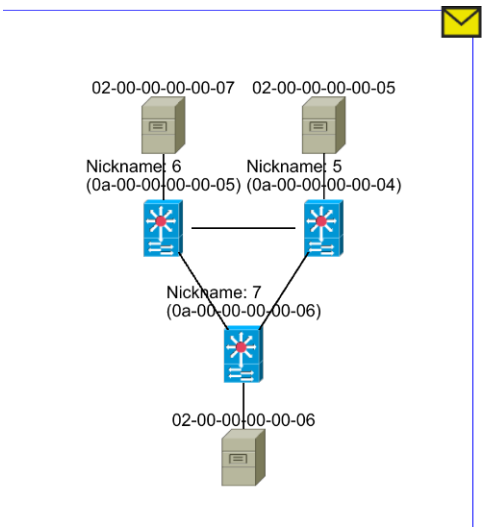
Obrázek 5 – Okno počítače s možností generovat provoz v síti



Obrázek 6 – Možnosti přepínačů (vlevo) a RBridge (vpravo)

Konvergence sítě

Stejně jako v reálných sítích je v modelu pro přepínání rámců potřeba, aby byla síť zkonvergovaná (aby přepínače a RBridge měly správné informace o topologii). Konvergence STP (na přepínačích i RBridge) probíhá na pozadí každých 20 kroků (počet kroků od začátku modelování je v programu vidět vpravo nad pracovní plochou). Konvergence protokolu TRILL (na RBridge) probíhá v modelu dvěma způsoby. Pokud je v levém menu zaškrtnuta volba zobrazení řídicího provozu, začíná konvergence sítě vždy po 120 krocích modelování. RBridge mezi sebou rozesílají žluté řídicí zprávy – Hello rámce, případně následně informace o změnách v topologii. Síť je po změnách zkonvergovaná až ve chvíli, kdy nejsou řídicí rámce dál rozesílány. Pokud volba zobrazení řídicích rámců není zaškrtnuta, probíhá rozeslání Hello rámců každých 30 kroků. Doručení řídicích rámců je v tomto případě mnohem rychlejší, než pokud jsou zobrazovány. Přesto konvergence probíhá několik kroků modelování (záleží na velikosti sítě) a proto je konvergence v jejím průběhu signalizována pomocí žluté obálky v pravém horním rohu pracovní plochy (Obrázek 7).



Obrázek 7 – Signalizace konvergence protokolu TRILL při vypnutém zobrazování řídicího provozu

Stejně jako v reálných sítích jsou některé změny topologie v modelu zařízeními zjištěny až po vypršení určitého času, který umožňuje zachování topologie i v případě krátkodobého výpadku (tzv. timeout). Jedná se např. o zrušení sousedství mezi RBridge nebo vymazání záznamu z CAM tabulky. Čas zbývající před smazáním těchto záznamů se zobrazuje u každého záznamu, kterého se timeout týká, v okně daného zařízení. Konvergence po smazání RBridge tak trvá delší dobu – informace o změně topologie jsou rozeslány až po vypršení daného času.

Pokud síť není zkonvergovaná, nemůže přepínání rámců fungovat správným způsobem, proto je potřeba po každé změně topologie počkat (případně zrychlit modelování), než konvergence sítě proběhne do svého konce.

Příloha C – Zdrojový kód TRILL rámce

```
package structures;

public class TrillFrame implements Cloneable {
    public byte version=1;
    public byte reserved=0;
    public boolean multicast;
    public byte opLength=0;
    public short hopCount;
    public short egressNickname;
    public short ingressNickname;
    public EthernetFrame payload;

    public TrillFrame(boolean multicast, short hopCount, short egressNickname,
short ingressNickname, EthernetFrame payload) {
       this.multicast = multicast;
       this.hopCount = hopCount;
       this.egressNickname = egressNickname;
       this.ingressNickname = ingressNickname;
       this.payload = payload;
    }

    @Override
    public Object clone() {
       TrillFrame newObject=new TrillFrame(multicast, hopCount, egressNickname,
ingressNickname, (EthernetFrame)payload.clone());
       return newObject;
    }
}
```

Příloha D – Zdrojový kód metody pro příjem rámce v RBridge

```
…

public class RBridge extends Switch {
…

@Override
public void inputFrame(ModelFrame sFrame, EthernetLink fromLink) {
    //  destroy corrupted frame
    if(!controlChecksum(sFrame)){
        sFrame.destroy();
        return;
    }
    //  get frames only from STP root ports and only on designated RBridge
    if(ModelController.getInstance().getDeviceOnLink(fromLink, this) instanceof
SwitchSTP){
        if(!rBridgeSTP.enabledPorts().contains(fromLink)){
            sFrame.destroy();
            return;
        }
    }
    //  add address to CAM table
    learnSourceMAC(sFrame, fromLink);
    //  TRILL frames
    if(sFrame.getFrame().etherType==etherTypeOfTrill){
        if(sFrame.getFrame().payload instanceof TrillFrame){
            TrillFrame tFrame=(TrillFrame)sFrame.getFrame().payload;
            //  distribution tree for host, which is not in host tables
            if(tFrame.multicast){
                //  decrease hopCount and destroy frames with hopCount==0
                if(0==--tFrame.hopCount){
                    sFrame.destroy();
                    return;
                }
                //  update info window
                sFrame.frameUpdate();
                //  send to non TRILL ports
                boolean sendedOut=egressProcess(sFrame, fromLink);
                if(sendedOut){
                    sFrame=(ModelFrame)sFrame.clone();
                }
                //  send to another parts of distribution tree
                boolean sendedMulticast=sendMulticastFrame(tFrame, fromLink,
sFrame, false);
                if(!sendedMulticast){
                    sFrame.destroy();
                }
            //  egress process - remove TRILL header and send orig frame to host
            }else if(tFrame.egressNickname==getNickname()){
                if(!egressProcess(sFrame,fromLink)){
                    sFrame.destroy();
                }
            //  core process - send to another RBridge
            }else{
                coreProcess(sFrame,fromLink);
            }
```

```
        }
    //  classic Ethernet frame
    }else{
        //  multicast TRILL IS-IS frame
        if(sFrame.getFrame().destination == macOfAllISISRbridges){
            if(sFrame.getFrame().etherType == etherTypeOfTrillISIS){
                //  Hello frame
                if(sFrame.getFrame().payload instanceof Hello){
                    RBridge
neighbor=((Hello)sFrame.getFrame().payload).getRBridge();
                    //  from actual neighbor
                    if(neighborTable.isNeighbor(neighbor)){
                        neighborTable.resetTimeout(neighbor);
                    //  new neighbor
                    }else{
                        neighborTable.addNeighbor(neighbor);
                        recalcTopology();
                        sendFrameTopologyChange(false);
                    }
                    sFrame.destroy();
                //  topology change
                }else if(sFrame.getFrame().payload instanceof TopologyChange){
                    receiveTopoChange(sFrame,fromLink);
                }
            }
        //  send to host out of TRILL network
        }else if(null!=camTable.get(sFrame.getFrame().destination)){
            if(!sendFrameOut(sFrame, fromLink)){
                sFrame.destroy();
            }
        //  ingress process - send to host via TRILL network
        }else if (hostTable.contains(sFrame.getFrame().destination)) {
            if(!ingressProcess(sFrame, fromLink)){
                sFrame.destroy();
            }
        //  don't know where is host
        //  send everywhere – out of TRILL + distribution tree in TRILL network
        }else{
            boolean sendedOut=sendFrameOut(sFrame, fromLink);
            if(sendedOut){
                sFrame=(ModelFrame)sFrame.clone();
            }
            boolean ingressed=ingressProcess(sFrame, fromLink);
            if(!ingressed){
                sFrame.destroy();
            }
        }
    }
}
…

}
```

Příloha E – Zdrojový kód části třídy představující CAM tabulku

```
package structures;

...

public class CAMTable extends Observable {
    private Switch inDevice;

    // key of HashMap is MAC address of host device
    private HashMap<Long, CAMTableEntry> table;

    public CAMTable(Switch inDevice){
       table=new HashMap<Long, CAMTableEntry>();
       this.inDevice=inDevice;
    }

    ...

    private class CAMTableEntry{
       private EthernetLink link;
       private int inserted;

       public CAMTableEntry(EthernetLink link) {
              this.link=link;
              resetTimeout();
       }

       public EthernetLink getLink(){
              return link;
       }

       //     refresh
       public void resetTimeout(){
              inserted=(int)RepastEssentials.GetTickCount();
       }

       //     900 - tick count from latest refresh
       public int getTimeout(){
              int timeout=900-((int)RepastEssentials.GetTickCount()-inserted);
              return timeout>0 ? timeout : 0;
       }

    }

}
```

Příloha F – Zdrojový kód zajišťující modelování pohybu zpráv

```
package structures;
...

public class ModelFrame extends Observable implements Cloneable {
    ...

    @ScheduledMethod(start=1,interval=1)
    public void move(){
        if(inMove){
            //  control frames if control traffic is hidden
            //  deliver after 3 ticks
            if(fastDelivery){
                if(3==++fastDeliveryCount){
                    deliver();
                }
            }else{
                //  distance and angle to target device
                NdPoint pt=space.getLocation(moveTo);
                NdPoint ptLast=space.getLocation(lastDevice);
                double distance=Math.sqrt(Math.pow(pt.getX()-ptLast.getX(),
2)+Math.pow(pt.getY()-ptLast.getY(), 2));
                double moveDistance=distance/17;
                //  move message
                space.moveByVector(this, moveDistance,
getAngleToDestination(),0);
                //  control actual distance to target
                NdPoint myLoc=space.getLocation(this);
                if(pt.equals(myLoc) || space.getDistance(pt,
myLoc)<0.9*moveDistance){
                    deliver();
                }
            }
        }
    }

    private void deliver(){
        inMove=false;
        EthernetLink actLink=ModelController.getInstance().getLink(lastDevice,
moveTo);
        //  change actual device
        lastDevice=moveTo;
        fastDeliveryCount=0;
        //  statistics of link
        actLink.addTrafficCount();
        //  input to device
        moveTo.inputFrame(this,actLink);
    }
    ...

}
```