VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2020 Jiří Březina

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

EVOLUCE PROTOKOLŮ PRO ZAMEZENÍ VZNIKU SMYČEK NA LINKOVÉ VRSTVĚ
EVOLUTION OF LINK LAYER LOOP PREVENTION PROTOCOLS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **Jiří Březina**
AUTHOR

VEDOUCÍ PRÁCE — **Mgr. Karel Slavíček, Ph.D.**
SUPERVISOR

BRNO 2020

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní program **Informační bezpečnost**

Ústav telekomunikací

Student: Jiří Březina ***ID:*** 198142
Ročník: 3 ***Akademický rok:*** 2019/20

NÁZEV TÉMATU:

Evoluce protokolů pro zamezení vzniku smyček na linkové vrstvě

POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je zmapovat vývoj protokolů typu spanning tree od prvopočátků po moderní protokoly, popsat rozdíly jednotlivý verzí, jejich výhody a nevýhody a rovněž spektrum aktivních síťových prvků, které jednotlivé protokoly podporují. Věcným výstupem práce bude přehled využití jednotlivých variant protokolu spanning tree u předních světových výrobců aktivních síťových prvků a připravit laboratorní úlohu, na které by bylo možné demonstrovat vlastnosti klíčových protokolů.
Úkolem semestrální práce je vypracovat seznam existujících spanning tree protokolů, podrobněji popsat alespoň tři z nich, vybrat protokoly, které budou použity pro laboratorní úlohy a připravit návrh zapojení laboratorních úloh.

DOPORUČENÁ LITERATURA:

[1] IEEE 802 LAN/MAN Standards Committee [online]. -: IEEE, 2019 [cit. 2019-10-31]. Dostupné z: http://www.ieee802.org/

[2] Transparent Interconnection of Lots of Links (TRILL) Use of IS-IS. RFC 6326. -: Internet Engineering Task Force (IETF), 2011.

Termín zadání: 3.2.2020 ***Termín odevzdání:*** 8.6.2020

Vedoucí práce: Mgr. Karel Slavíček, Ph.D.

doc. Ing. Jan Hajný, Ph.D.
předseda rady studijního programu

ABSTRAKT

Tato bakalářská práce se zabývá možností eliminace tvorby smyček a na linkové vrstvě OSI modelu. Nejprve bylo potřeba určit, co to vůbec smyčky jsou a jak se tvoří. Následně bylo představeno řešení problému se smyčkami v podobě protokolu Spanning Tree, jeho vznik, fungování, výhody a nevýhody. Dále byly probrány další varianty originálního STP, a také jeho alternativy například v podobě protokolu TRILL nebo SPB. Praktická část této práce navazuje na teoretickou a ukazuje, jak fungují protokoly Spanning Tree v praxi na reálných zařízeních. Tato část se věnuje především testování rychlostí konvergence jednotlivých protokolů a také jejich rozšířením. Ke konci je zmíněna laboratorní úloha, která je součástí příloh a tvoří hlavní přínos celé práce, protože si v díky ní budoucí studenti předmětu „architektura sítí“, mohou vyzkoušet konfigurovat výše zmíněné protokoly a lépe tak porozumět tomu, jak tyto protokoly fungují.

KLÍČOVÁ SLOVA

BPDU, instance, konvergence, MSTP, přepínač, PVST, region, RSTP, smyčka, spanning-tree, TRILL, VLAN

ABSTRACT

This bachelor thesis deals with the possibility of eliminating and creation loops on the Data Link layer of the OSI model. At first, it was necessary to determine what are the loops and how they are formed. Then the solution of the problem with loops was presented as the Spanning Tree protocol, where did the protocol come from, protocol operation, his advantages and disadvantages. In next section were discussed other variants of original STP, as well as his alternatives, like for example the TRILL or the SPB protocol. The practical part of this work is based on the theoretical and shows how Spanning Tree protocols work in practice on real devices. This part is mainly focused to testing the speed of convergence each of protocols and also their extensions. In the end is mentioned a laboratory task, which is part of the appendices and forms the main benefit of the whole thesis, because future students at school subject “network architecture” can try to configure protocols that was mentioned above and better understand how these protocols work.

KEYWORDS

BPDU, instance, convergence, MSTP, switch, PVST, region, RSTP, loop, spanning-tree, TRILL, VLAN

BŘEZINA, Jiří. *Evoluce protokolů pro zamezení vzniku smyček na linkové vrstvě*. Brno, Rok, 67 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: Mgr. Karel Slavíček, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Evoluce protokolů pro zamezení vzniku smyček na linkové vrstvě“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autora

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu Mgr. Karlu Slavíčkovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci. Dále také děkuji za poskytnutí fyzických zařízení, která byla nutná k vypracování praktické části práce.

Obsah

Seznam obrázků

Seznam tabulek

Úvod

V dnešní době existují různé velké počítačové sítě, od těch nejmenších například u kohokoli doma, přes různě velké sítě v různě velkých společnostech, školách, úřadech atd., až po ty opravdu velké, v podobě obřích datových center nebo infrastruktury velkých poskytovatelů služeb. Tato práce se bude primárně zaobírat technikami eliminace vzniku smyček na linkové vrstvě OSI modelu, které se tvoří právě u sítí většího rozsahu. Dnes by bez těchto technik prakticky nemohla fungovat síť internet.

První kapitola se bude věnovat stručné historii technologie Ethernet a popisu fungování základních síťových zařízení pracujících na 2. vrstvě ISO/OSI modelu. Dále se popíše princip smyček v síti a jak dojde k vytvoření takové smyčky a pochopitelně proč představují značný problém.

Druhá kapitola ukáže již možnosti eliminace smyček pomocí Spanning Tree protokolu. Je zde popsán jeho vývoj a jeho fungování včetně detailního popisu parametrů nutných proto, aby pracoval správně.

Ve třetí kapitole se práce více zaměří na existující verze Spannig Tree protokolu a také dostupné alternativy. U verzí STP je popsáno jak se konkrétně liší od původního protokolu a jaké mají výhody vůči originálnímu protokolu. Zmíněna budou i proprietární řešení síťových výrobců. Druhá část této kapitoly je věnována výhradně alternativním řešením s podrobnám popisem protokolu TRILL, a především toho, proč je navržen jako nástupce protokolu STP. V neposlední řadě ještě bude zmíněno několik dalších alternativních řešení, která fungují na obdobném principu jako TRILL.

Čtvrtá kapitola bude prezentovat využití protokolů typu Spanning Tree v praxi. Navazuje tak na teoretickou část práce a pokusí se problematiku různých verzí protokolů Spanning Tree víc přiblížit a ukázat jejich fungování na reálných zařízeních od předního světového výrobce aktivních síťových prvků.

Poslední pátá kapitola se bude věnovat pouze stručnému popisu laboratorní úlohy, která je obsažena v přílohách této práce a tvoří hlavní výstup této práce.

1 Historie a smyčky

V počátcích technologie Ethernet byl rozsah zařízení spojové vrstvy určen maximem kompletního zpoždění, které ještě dovolovalo detekci kolizí ve sdíleném segmentu. Maximální zpoždění se sestávalo z maximálního počtu rozbočovačů, opakovačů a dalších (zpoždění způsobujících) prvků v segmentu. Postupně se tyto omezení začaly uvolňovat, když společnost DEC (Digital Equipment Corporation)[1] představila svůj první 2-portový ethernetový most[2] přibližně v polovině 80. let minulého století. Byl to první krok ke stavbě libovolně velkých L2 sítí, které vznikaly spojením více kolizních domén, do větší broadcastové domény. Nedlouho poté následovaly i první síťové prvky dnes známé, jako přepínače.[1]

1.1 Ethernetový rozbočovač

Rozbočovač (Hub) je primitivní aktivní zařízení a základní prvek sítě s hvězdicovou topologií, sloužící k jejímu větvení. Rozbočovač kopíruje veškerý provoz z jednoho portu na všechny ostatní porty aniž by zkoumal komu data skutečně náleží. Jinými slovy se jedná o broadcastové vysílání, které má jednak za následek značné zahlcování sítě a jednak z bezpečnostního hlediska je použití rozbočovače nevhodné, protože jakákoliv stanice, v téže rozbočovači, může číst obsah pro jinou stanici. Kvůli těmto nedostatkům byl postupem času nahrazen přepínačem a dnes se již ani nevyrábí, ani nepoužívají.[2]

1.2 Ethernetový přepínač

Na rozdíl od staršího rozbočovače, dokáže přepínač (switch) třídit provoz a posílat ethernetové rámce na konkrétní rozhraní, a pro konkrétního adresáta na základě MAC (Media Access Control) adresy, která je uvedena v CAM (Content-addressable memory) tabulce přepínače. Tabulku si přepínač sestavuje sám automaticky podle MAC adres obsažených v příchozích rámcích asociovaných s portem přepínače odkud rámec přišel. Přepínač tedy funguje na druhé (spojové) vrstvě ISO/OSI modelu. Problém pochopitelně nastane pokud přepínač cílovou MAC adresu nezná (ještě se ji nenaučil), případně je společná pro více koncových stanic (adresa typu *multicast*). V

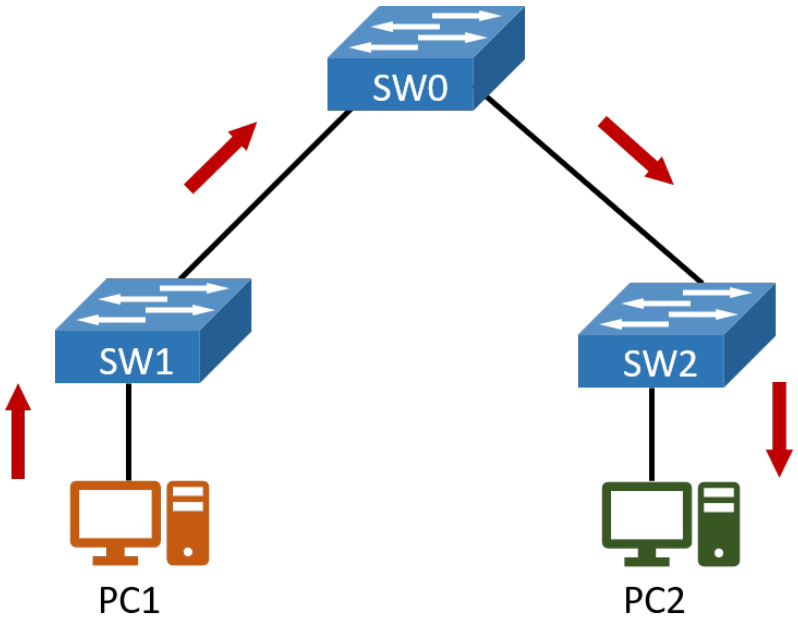
[1] Jedna z vůbec prvních společností v americkém počítačovém průmyslu. V minulosti byla odkoupena společností Compaq (ta dále potom spol. Hewlett-Packard) a dnes stále vydává produkty pod značkou HP.

[2] Most nebo-li anglicky bridge je zařízení sloužící k propojení LAN sítí a od přepínače (switche) se liší hlavně počtem portů a funkcionalitou, kdy přepínač je ve své podstatě více mostů, které fungují společně.

tomto případě se přepínač zachová jako rozbočovač a odešle rámec na všechny ostatní porty mimo port, ze kterého rámec přišel (tedy broadcastem) a následně čeká na odpověď adresáta. Hlavním problémem tohoto řešení je, když se replikovaný rámec vrátí zpět na stejný přepínač, kterým už předtím prošel. Tento proces se totiž bude opakovat a opakovat, dokud nedojde ke kompletnímu zahlcení sítě kopiemi a kopiemi stejného rámce. Tímto lze tedy říci, že smyčky jako takové naprosto nežádoucím prvkem v ethernetové síti.[2]

1.3 Smyčky v síti

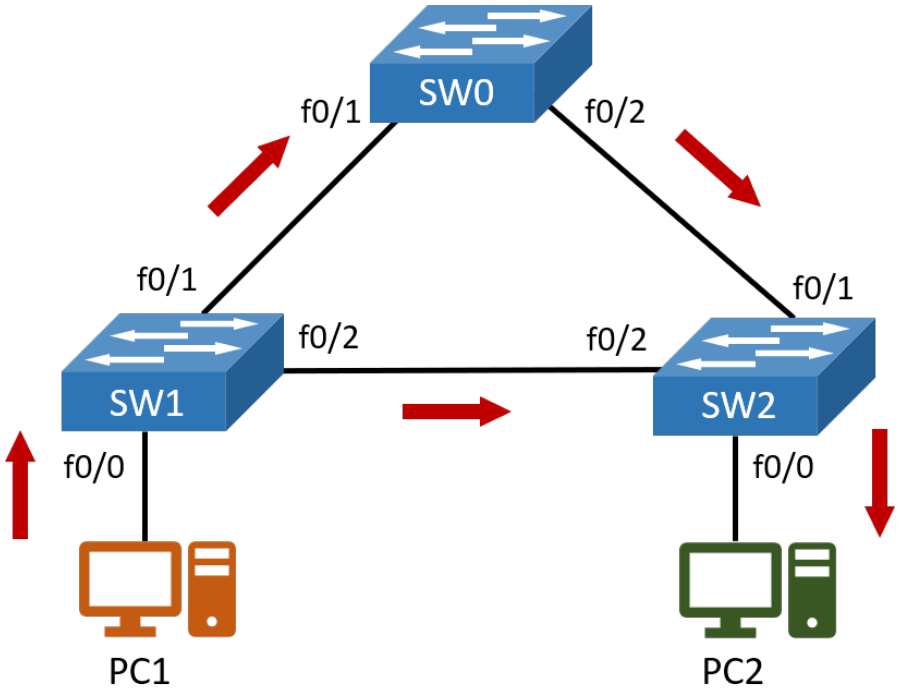
V ethernetové lokální síti je dnes je běžné zapojení do rozšířené topologie hvězda. Vznikne tím stromová struktura, což znamená, že mezi každým jedním prvkem v síti je jen jedna cesta viz příklad na obrázku 1.1.



Obr. 1.1: Ukázka zapojení bez smyčky

Samozřejmě je v dnešní době kladen velký důraz na dostupnost služeb, takže se lokální sítě rozšiřují a vznikají tak redundantní (záložní) cesty k cíli. Toto řešení má výhodu v tom, že pokud dojde k výpadku nějakého jednoho prvku a nebo třeba linky, pak je možné stále provozovat větší část sítě a směrovat komunikaci jinou cestou, viz obrázek 1.2.

Nevýhodou je ovšem v takovém případě právě vznik smyčky mezi přepínači, reps. zapojení linek mezi přepínači do smyčky. [3]



Obr. 1.2: Ukázka zapojení se smyčkou

1.4 Ukázka vzniku smyčky

V dnešních rozsáhlých lokálních sítích už v naprosté většině případů vzniknou logické smyčky buď chybou obsluhy, nebo neodbornou manipulací s přepínači. Následující popis fungování zařízení zapojených do smyčky se věnuje obrázku 1.2.

1. PC1 odešle rámec pro PC2
2. SW1 přijme rámec na f0/0, uloží si záznam (o PC1) do CAM tabulky a rámec odešle na f0/1 a f0/2.
3. SW0 přijme rámec na f0/1, uloží si záznam (o PC1) do CAM tabulky a rámec odešle na f0/2.
   SW2 přijme rámec na f0/2, uloží si záznam (o PC1) do CAM tabulky a rámec odešle na f0/0 a f0/1.
4. *PC2 v tuto chvíli již obdrželo zprávu, ale přepínače o tom neví.*
5. SW0 přijme rámec na f0/2, opraví si záznam (o PC1) v CAM (protože si myslí, že se PC1 přesunulo) a odešle na f0/1
   SW2 přijme rámec na f0/1, opraví si záznam (o PC1) v CAM (protože si myslí, že se PC1 přesunulo) a odešle na f0/0 a f0/2
6. SW1 přijme rámec na f0/2, opraví si záznam (o PC1) v CAM (protože si myslí, že se PC1 přesunulo) a odešle na f0/0 a f0/1
7. *PC1 pozná, že rámec není pro něj a tedy jej zahodí.*

8. SW1 přijme rámec na f0/1, opraví si záznam (o PC1) v CAM (protože si myslí, že se PC1 přesunulo) a odešle na f0/0 a f0/2

Takto se neustále bude rámec cyklit v síti, protože na druhé vrstvě ISO/OSI, konkrétně v ethernetovém rámci neexistuje parametr podobný hodnotě TTL (Time to Live) ze třetí vrstvy, který by vyslaný rámec po určité době zlikvidoval. Pokud vezmeme do úvahy fakt, že se takto v síti začne cyklit desítky, stoky nebo i tisíce rámců, tak může i v malé síti dojít velice rychle ke kolapsu.

1.5 Problémy smyček

Nejčastějším problémem smyčky je, že dojde k tzv. broadcastové bouři. Broadcastová bouře znamená nekontrolovatelné šíření broadcastových (případně i jiných) rámců celou sítí, která postupem času přestane být schopna tyto rámce zpracovávat. Už jenom z principu funkce přepínačů musí zákonitě dojít k takovémuto efektu. Což nám taky krásně demonstroval příklad výše, a to se v tomto případě jednalo pouze o příklad s jediným rámcem a topologií čítající 2 počítače a 3 přepínače. V případě rozlehlejší topologie a při zasílaní většího počtu rámců, by došlo k úplnému zahlcení sítě a následnému kolapsu už během několika málo minut.

Další problémy, co mohou nastat, vyplývají z broadcastové bouře a tedy: Problémy s konektivitou - díky smyčce dochází na přepínač jeden rámec postupně z několika portů a on si neustále bude přepisovat zdrojovou MAC adresu rámce ve své CAM tabulce. Několikanásobné doručení rámce – rámec bude sítí kolovat prakticky donekonečna.[3]

2 Spanning Tree protocol

Za samotným zrodem Spanning Tree protokolu stojí Radia Joy Perlman, která studovala na MIT (Massachusetts Institute of Technology) a svá studia zakončila titulem Ph.D v oboru informatika. Poté pokračovala od roku 1980 v práci jako vývojář výpočetní techniky ve společnosti DEC (Digital Equipment Corporation). Zatímco zde pomáhala transformovat ethernetové sítě ze sítí malého rozsahu, do sítí, které dokáží zvládnout stovky tisíc síťových uzlů rozšířených v různých oblastech, tak tady taky také započala svoji práci na algoritmu pro Spanning Tree protokol.

V roce 1985 Perlmanová dokončila svoji práci na onom algoritmu ten je poprvé představen v dou-portovém ethernet mostu společnosti DEC a do prodeje uveden přibližně ve stejném roce. Původní označení bylo DEC STP.

V době svého vzniku byl STP revoluční, protože umožňoval budovat síťové architektury typu „plug-and-play“, ve kterých bylo možno připojovat další přepínače aniž by docházelo k tvorbě smyček jako tomu bylo v minulosti. STP byl rovněž schopen podporovat celou řadu budoucích sítových aplikací a služeb a jejich schopnosti pracovat se staršími technologiemi.

V roce 1990 je STP včleněno do nově vznikajícího standardu organizace IEEE[1] 802.1D, který řeší kontrolu přístupu k médiím a slouží jako základní součást síťových architektur typu LAN.[4]

2.1 Fungování STP

Tato sekce je věnována popisu fungování původního Spanning Tree protokolu definovaného normou IEEE 802.1D. STP protokol pracuje na principu teorie grafů, kde ohodnoceným grafem označujeme síť, ke které algoritmus hledá kostru. Nebo-li snaží se najít nejkratší cestu vždy mezi každými dvěma sousedícími přepínači. Pro vytvoření databáze topologie používá Spanning Tree Algorithm (STA), následně se snaží najít redundantní (nadbytečné) spoje na přepínačích, u kterých pak blokuje porty těchto redundantních spojení. Zjednodušeně STP funguje tak, že si na fyzické topologii, která většinou (vzhledem k dnešním sítím) obsahuje smyčky, vytvoří svoji vlastní virtuální topologii, která je již bez smyček. Vzhledem k tomu, že je STP dynamickým protokolem, tak je schopen změnám v síti přizpůsobovat a pakliže vznikne nová smyčka, tak dojde k rekonfiguraci, aby se smyčka redukovala.[3, 5]

[1] Mezinárodní nezisková a profesní organizace Institute of Electrical and Electronics Engineers, usilující o vzestup technologií spojených s elektrotechnikou.

2.1.1 Určení nejkratší cesty

STP určí nejkratší cestu tak, že si nejprve vytvoří strom všech cest (kostru grafu). Poté určí nejkratší cestu podle kumulativní ceny linky, která je dána její propustnosti (anglicky bandwith). Cena je určována podle tabulky níže.[3]

Tab. 2.1: Ceny cest v závislosti na rychlosti linky

Rychlost linky	**Původní cena**	**Cena od 1998**	**Cena od 2001**
10 Gb/s	1	2	2000
2 Gb/s	1	3	10000
1 Gb/s	1	4	20000
100 Mb/s	10	19	200000
10 Mb/s	100	100	2000000

Pozn.: V době vzniku STP nedosahovaly ještě lokální sítě takových rychlostí, jako je tomu v dnešní době, a proto se nepočítalo s vyššími rychlostmi jak 1 Gb/s. Postupně tedy byla tabulka aktualizována aby odpovídala dnešním standardům.

2.1.2 Základní parametry pro fungování STP

Aby mohl spannig tree protokol správně fungovat, musí určitým způsobem dorozumívat s přepínači a zároveň musí být zvolen kořenový (hlavní) přepínač (anglicky Root Bridge).

Bridge ID (BID)

Jedná se o jednoznačné určení (identitu) každého přepínače, který je součástí daného segmentu sítě. Hodnota BID je 8 bajtů a skládá se ze dvou částí. První 2 bajtová část udává tvz. Bridge Priority, jejíž výchozí hodnota (převedeno do desítkové soustavy) je 32768, ale může být změněna. Druhá 6 bajtová část BID se skládá z MAC adresy přepínače, která je unikátní mezi všemi přepínači. Přepínač s nejnižší hodnotou BID se stane tzv. **Root Bridgem**.[3, 5]

Bridge Protocol Data Units (BPDU)

BPDU jsou speciální rámce, které využívá STP pro komunikaci mezi přepínači. BPDU se dělí na tři části. První část obsahuje globální informace (např. o verzi STP). Druhá obsahuje informace o dané instanci STP (např. root BID nebo cenu cesty). Poslední část obsahuje časové parametry, které mimo jiné určují i Hello Time interval, kdy se po uplynutí určité doby (výchozí hodnota je 2 sekundy) budou znovu posílat BPDU rámce.[3, 5]

Tab. 2.2: Složení BPDU rámce

Velikost v bajtech	**Položka**
2	Protocol ID
1	Protocol version
1	BPDU type
1	Flags
8	Root BID
4	Root path cost
8	sender BID
2	sender port ID
2	Message Age
2	Max Age
2	Hello Time
2	Forward delay

2.1.3 Root Bridge

Je jím takový přepínač, který bude mít nejnižší hodnotu BID. Zároveň by to měl být vždy nejvýkonnější přepínač v síti, protože to bude právě on, kdo bude rozhodovat o veškerých dalších rozhodnutích (například které porty budou zvoleny jako kořenové). Pokud chceme manuálně určit Root Bridge, pak je potřeba upravit hodnotu Bridge Priority na nižší, než je ta výchozí. Automatické určení Root Bridge probíhá následovně:

1. Každý (např. nově) připojený přepínač odešle broadcastem (tzn. na všechny ostatní porty) rámec BPDU, ve kterém má nastaveno **svoje BID** jako **root BID**.
2. Přepínače zjišťují hodnotu BID obsaženou v **přijatém rámci BPDU** a pokud je **vyšší**, než jejich vlastní, tak ji zamění na svoji vlastní a odešlou.
3. Opačně pokud je hodnota BID v přijatém rámci **nižší**, než jejich vlastní, pak přepínač, **od kterého rámec přišel**, uznají za Root Bridge.[3, 5]

2.1.4 Role portů

V základním (802.1D) spanning tree protokolu jsou definovány tři možné typy portů, které přepínače mohou mít.

- **Root port (kořenový port)** – tento port má nejnižší cenu a slouží k přímému propojení s Root Bridgem (případně s nejkratší cestou k Root Bridgi)
- **Designated port (vyhrazený port)** – druhý typ portu STP topologie, připojuje další segment sítě. Root Brigde má tyto typy portů.

- **Non-designated port (nepředávající nebo taky blokovaný port)** – STP blokované porty, které slouží pro záložní cestu v případě výpadku té hlavní. Porty jsou stále zapnuty jen je jim blokováno předávání provozu.[3, 5]

2.1.5 Stavy portů

Aby mohl port na přepínači, na kterém pracuje STP fungovat (přenášet data), tak musí postupně projít několika stavy, kdy výsledným stavem je vždy poslední (pokud má port sloužit pro aktivní komunikaci a ne pro zálohu). K tomuto procesu volby stavu portu dochází pokaždé při konvergenci (změně topologie) sítě. Takže pokaždé když dojde byť ke krátkému výpadku jedné z linek. Mezi stavy je vždy několika sekundové prodlení (u 802.11D).

- **Disabled (Vypnuto)** – V tomto stavu se port nachází pakliže je vypnut, nebo nemůže být používán po fyzické stránce (byl správcem deaktivován). Neodesílá ani nepřijímá žádná data včetně BPDU rámců.
- **Bloking (Blokování)** – Do tohoto stavu automaticky přejde port, který byl zapnut. Všechna komunikace je blokována/zakázana a jediné co port dělá je, že přijímá BPDU rámce, ale dál už je nevysílá. Případně ještě reaguje na zprávy správy sítě. Tento stav trvá nejdéle 20 sekund, poté přechází na dlaší. Standardně se tomto stavu se nachází non-designated porty.
- **Listening (Naslouchání)** – Oproti předchozímu stavu už i posílá rámce BPDU (nejenom přijímá), nicméně stále je ostatní komunikace blokována. Tento stav trvá nejdéle 15 sekund a bývá označován jako první směrovací zpoždění.
- **Learning (Učení se)** – U tohoto stavu se port připravuje na plný provoz proto zachycuje příchozí rámce a učí se MAC adresy, které si ukládá do CAM tabulky. Tento stav trvá nejdéle 15 sekund a bývá označován jako druhé a poslední směrovací zpoždění.
- **Forwarding (Přeposílání)** – Port přechází do plného provozu, konvergence je dokončena. V tomto stavu se nachází root i designated porty.[3, 5]

2.1.6 Konvergence

Konvergence sítě je časový interval, za který dokáže port přepínače přejít ze stavu bloking do stavu forwarding. Z popisu stavů výše je zřejmé, že konvergovaná je síť teprve v momentě, kdy jsou porty přepínačů v jednom z koncových stavů. Celkový čas, který je pro toto potřeba, je v případě původního STP (802.11D) maximálně 50 sekund (v praxi je to o něco méně, a sice okolo 30 sekund), což je tedy i doba potřebná pro zprovoznění záložní linky v případě výpadku té hlavní.[3]

3 Varianty a alternativy STP

V předchozí kapitole bylo vyjasněno jak vlastně funguje klasický a původní STP, který byl definován normou IEEE 802.1D. Tato kapitola se bude zabývat dalšími variantami protokolu Spanning Tree, které vznikly ať už jako další (novější) verze původního standardu, anebo proprietární řešení různých síťových výrobců. Dále budou v textu zmíněny alternativní protokoly, které fungují odlišně od Spanning Tree a jsou v současnosti na vzestupu.

3.1 Rapid Spanning Tree Protocol

RSTP vznikl v roce 2001 z důvodu potřeby rychlejší konvergence sítě. U standardního protokolu STP se s postupem času ukázalo, že čas konvergence je příliš dlouhý pro použití v praxi. Zatímco u STP se běžná doba konvergence pohybuje okolo 30 sekund, u protokolu RST je tato doba běžně nižší řádově o desítky sekund a sice je to přibližně 1-3 sekundy. Novější protokol se tedy dokáže změnám v síti přizpůsobovat mnohem rychleji a efektivněji než starší STP. RSTP je zpětně kompatibilní s STP, nicméně samotné fungování potom bude degradováno na úroveň staršího z protokolů.

Původně byl protokol definován v samostatné normě 802.1w, avšak v roce 2004 došlo organizací IEEE k revidování normy 802.1D, kdy byly do této normy sloučeny právě norma 802.1w a norma 802.1t. Tímto došlo de facto k nahrazení původního STP právě novějším RSTP.[7, 5].

3.1.1 Změny proti STP

Role portů

U RSTP došlo ke zpřesnění jednotlivých rolí portů, zatímco role Root a Designated portů zůstaly beze změn vůči STP, tak původní role „Non-designated“ se zde rozdělila na nové dvě:

- **Alternate port** – RSTP blokovaný port sloužící jako alternativní spojení s Root Bridgem v případě výpadku root portu.
- **Backup port** – RSTP blokovaný port sloužící jako záložní cesta pro daný síťový segment. V podstatě jde o zálohu designated portu.[7]

Stavy portů

Stavy portů se dočkaly zjednodušení a z původních 5-ti se snížil jejich počet na pouhé 3 stavy.

- **Discarding (Vyřazeno)** – Tento stav kombinuje 3 stavy z původního protokolu STP, a sice stavy: disabled, bloking, listening. Nachází se v něm tedy jak vypnuté, tak zapnuté porty. U těch zapnutých se přijímají i posílají BPDU rámce, ale ostatní komunikace je blokována.
- **Learning (Učení se)** – Stejná funkcionalita jako v případě STP, port se připravuje na plný provoz a učí se MAC adresy.
- **Forwarding (Přeposílání)** – Opět stejná funkcionalita, jako v případě STP - port v tomto stavu přechází do plného provozu (dokončení konvergence).[7, 5].

BPDUv2

BPDU rámce jsou nově ve verzi 2. Další změnou je nově využití všech 8 flag bitů oproti jeho předchůdci, který využíval jen dva na změnu topologie a potvrzení změny topologie. Nově jsou využívány všechny možnosti včetně pole „Port Role“, neboli role portu. Další změnou je, že BPDU rámce už nejsou generovány pouze root bridgem, ale generují je všechny přepínače a to (ve výchozím nastavení) každé 2 sekundy (hello time).

Klasické spanning tree nakládalo s BPDU rámci tak, že pokud uplynul čas životnosti (maximum bylo 20 sekund), rámec byl zahozen. RSTP v tomto ohledu funguje jinak, BPDU rámce používají udržovací mechanismus podobně jako například směrovací protokoly OSPF nebo EIGRP. Pokud přepínač nezachytí poslední 3 vyslané rámce BPDU od svého souseda, tak usoudí, že konektivita k tomuto sousednímu přepínači je ztracena.

Rovněž RSTP akceptuje méněcenné BPDU rámce, což v podstatě implementovaná funkce backbone fast feature, která byla po sléze přidána do klasického STP.[7, 6].

Konvergence

Rychlost konvergence sítě je v dnešní době zásadní faktor, a proto RSTP již nepoužívá pomalou metodu založenou na časovačích, jako tomu bylo u STP, kdy musel port projít několika stavy vždy s několika sekundovým zpožděním. RSTP používá nově metodu tzv. smlouvání, která umožňuje přeskočit rovnou na stav forwarding, a tím značně zrychlit proces konvergence.[8]

3.2 Per VLAN Spanning Tree

PVST nebo-li Per VLAN Spannig Tree je proprietární protokol vyvinutý společností Cisco a používaný ve výchozím nastavení na přepínačích tohoto výrobce. Termínem

„VLAN“ je myšleno Virtual Local Area Network. Tento protokol se postupem času dočkal vylepšení a nových verzí, které jsou uvedeny dále v textu.[3]

Cisco Systems

Cisco Systems, Inc. je v dnešní době jedna z největších počítačových společností a dominantní společnost na trhu se síťovými prvky. Založena byla v roce 1984 manželským párem Lena Bosacka a Sandy Lernerovou, kteří v té době působili na Stanfordově Univerzitě. Cisco se soustředí především na výrobu kvalitních síťových prvků a vytváření dlouhodobých partnerství se zákazníky, kterým tak můžou poskytovat řešení na míru.

V roce 1997 Cisco darovalo nějaké síťové prvky a vybavení jedné Kalifornské škole, jenže tam nebyl nikdo, kdo by byl vyškolen a věděl, jak správně s tímto vybavením pracovat. Ve stejném roce tedy vznikla tzv. Cisco Akademie (Cisco Networking Academy), která měla a stále má za cíl vzdělávat studenty jak v teoretické, tak i v praktické oblasti počítačových sítí. Cisco Akademie je dnes velice oblíbená na celé řadě technických škol ve 180 zemích.[10, 11]

VLAN

VLAN je technologie, která spatřila světlo světa v roce 1995 a její klíčovou vlastností je, že umožňuje oddělit fyzické zapojení od logického. Pokud byl dříve požadavek na připojení uživatelů do více různých LAN sítí (například kvůli oddělení zaměstnanců firmy různých pozic na stejném pracovišti), musela se každá skupina těchto uživatelů připojovat do samostatných přepínačů. Díky VLAN sítím je možnost připojit různé skupiny uživatelů zapojit fyzicky do jednoho přepínače a mít je logicky oddělené. Aby spolu mohli uživatelé z rozdílných VLAN komunikovat, tak je potřeba buď směrovače a nebo L3 přepínače pro inter-VLAN routing. Samozřejmě potom je potřeba určit, ze které VLAN a hlavně do jaké VLAN daný rámec směřuje. Jeden z nejrozšířenějších způsobů, jak docílit správného označení rámce (tzv. VLAN tagging). Jedním ze způsobů, jak rámec správně označit, je dán normou 802.1Q, která bude podrobněji rozebírána v části o PVST+.[12]

Důvody vzniku a výhody VLAN

VLAN vznikly především (jak už vyplynulo z předchozího odstavce) pro seskupování uživatelů z různých fyzicky oddělených skupin do logicky spojených celků. Dále potom, aby se snížily počty broadcastů v sítích a zmenšil se počet kolizních domén ve srovnání s dobou, kdy se ještě používaly místo přepínačů rozbočovače.[12]

Ačkoliv se dnes důvody vzniku VLAN neliší od těch v roce 1995, tak postupem času došlo k jejich aktualizaci, úpravě, nebo se objevil nový důvod. Seznam hlavních výhod VLAN, používaných dnes:

- **Snižování broadcastů** – toto se od vzniku nezměnilo jen při dnešních počtech zařízení připojujících se do sítí, je o to víc žádoucí v rámci zachování svižného provozu a vysokého výkonu.
- **Zjednodušení správy** – není potřeba fyzicky měnit pozici/zapojení zařízení, pouze se virtuálně změní zařezení do patřičné VLAN.
- **Zabezpečení** - díky VLAN sítím je možné oddělovat komunikaci a například omezit přístup.
- **Oddělení provozu** – v dnešní době je žádoucí oddělovat různé druhy provozu v síti a rovněž je třeba i prioritizovat. Jako příklad může posloužit IP telefonie, kdy je potřeba jednak oddělit telefony od zbytku sítě.
- **Nižší nároky na hardware** – s pomocí VLAN klesají nároky na hardware takovým stylem, že není nutné pro různé podsítě používat více přepínačů.[12]

3.2.1 PVST

PVST původně vychází z normy 802.1D (tedy původního STP), nicméně Cisco jej upravilo o možnost fungovat v rámci každé jednotlivé VLAN na přepínači. Zjednodušeně řečeno je PVST schopen pro každou VLAN spustit vlastní instanci spanning tree. V praxi se tato funkcionalita hodí hlavně pokud potřebuji mít například pro 2 skupiny VLAN sítí dva různé root brigde. Dále Cisco definovalo pro PVST několik rozšíření, které slouží například pro zvýšení rychlosti a bezpečnosti sítě.

3.2.2 VLAN Trunking Protocol

Nastavovat ručně na přepínačích všechny VLAN sítě, tak aby měl každý přepínač v jedné dom=ně informace o všech VLAN sítích je při menším počtu zařízení sice proveditelné, nicméně i tak je to proces poměrně pracný a navíc existuje možnost, že se člověk při konfiguraci jednoho zařízení přepíše a bude potřeba lokalizovat a opravit chybu. Pro jistou míru zjednodušení a automatizace lze použít VLAN Trunking Protokol (VTP), což je proprietární řešení od společnosti Cisco sloužící pro přenos informací o VLAN sítích (od určité verze ne jen o nich) mezi jednotlivými přepínači v síťovém segmentu. VTP se tak může starat o kompletní správu v případě vytvoření, přejmenování nebo vymazání VLAN sítě/sítí uvnitř jedné VTP domény.[13, 14]

VTP role

VTP může fungovat na přepínači v jedné z celkem tří (čtyř u v3) rolí.

- **Server** – výchozí role, díky které se přepínač stane řídícím přepínačem v doméně a stará se o správu (vytváření a mazání) VLAN sítí, které má uložené v NVRAM a rozesílá informace o těchto VLAN sítích pro celou VTP doménu.
- **Klient** – obdrží konfiguraci od serveru, kterou si sám nastaví a dále tuto konfiguraci přeposílá. Nemůže si vytvářet vlastní konfigurace sám.
- **Transparentní** – Původně jediná role, ve které se přepínač VTP přímo neúčastní a může si vyvářet vlastní VLAN sítě. Změny ovšem budou jen lokálního charakteru (neodesílají se dál). Od VTP verze 2 už přeposílá VTP zprávy s konfigurací od serveru na další přepínače, sám si je ovšem neaplikuje na rozdíl od klienta.
- **Off** – přibyla s VTP verzí 3 a podobá se transparentní roli, jen VTP zprávy **nepřeposílá** dál jinak pracuje stejně jako transparentní role.[13, 14]

Verze VTP

VTP protokol existuje celkem ve třech verzích.

1. verze – nejstarší
   - Podporuje základní číselný rozsah VLAN (od 1 do 1005)
   - Funguje v Ethernetu a v FDDI
2. verze
   - Stejný podporovaný rozsah VLAN sítí jako u v1
   - Přidána podpora pro sítě typu Token Ring
   - Transparentní role nyní přeposílá VTP zprávy dál
3. verze
   - Přidána podpora pro extended rozsah VLAN sítí (tedy až do 4096)
   - Udržuje částečnou kompatibilitu s předchozími verzemi
   - Přidány prvky pro ochranu databáze VLAN sítí (možnost nastavit heslo)
   - **Přidána podpora pro synchronizaci i ostatních databází**[13, 14]

3.2.3 Rozšíření PVST

PortFast

Určitě jedno z nejužitečnějších rozšíření původního STP, protože umožňuje přeskočit stavy listening a learning a přejít s portem rovnou do stavu forwarding. Toto se hodí hlavně u portů, které jsou koncové, nebo-li je v nich připojená koncová stanice, protože se nemusí čekat na delší časové intervaly stavů listening a learning což kladně přispívá k rychlosti konvergence celé sítě. Samozřejmě může nastat problém, pokud by do portu, na kterém je aktivován PortFast, byl připojen (například omylem) další přepínač, jelikož by mohlo dojít k vytvoření smyčky, která by nemusela být

detekována STP. Toto je možné řešit pomocí dalšího rozšíření BPDUguard – bude popsáno níže.[3]

UplinkFast

Funkce slouží (podobně jako PortFast) k okamžitému přepnutí portu do stavu forwarding, nicméně tentokrát ne pro koncové porty, ale porty, které jsou ve stavu bloking a jsou určeny jako záložní cesta v případě výpadku root portu přepínače. Funkce se dá použít pouze v případě, že přepínač má nějaké porty ve stavu bloking. Toto se týká hlavně přepínačů na přístupové (access) vrstvě, protože u nich se předpokládá, že budou mít blokovaná záložní spojení. UplinkFast zle použít pouze pro celý přepínač, ne v jednotlivých VLAN.[3, 15]

BPDUguard

Nelze jej nastavit samostatně, ale už z principu funkce se používá společně s funkcí PortFast a slouží k ochraně koncového portu proti připojení nežádoucího přepínače do tohoto typu portu. Pokud na port chráněný BPDUguardem přijde BPDU rámec, tak se port okamžitě vypne (přejde do stavu error-disabled) a znemožní tak další provoz. Funkce se dá nastavit buď na konkrétní porty, a nebo globálně pro všechny porty na přepínači (na kterých je zároveň nastaven PortFast).[3]

BPDUfilter

BPDUfilter také slouží pro k ochraně portu proti připojení nežádoucího přepínače. Funguje ovšem poněkud odlišně, protože BPDUfilter na rozdíl od funkce BPDUguard port nevypne při příchozím BPDU rámci, ale pouze filtruje provoz těchto příchozích rámců. Rovněž přepínač na daný port přestane vysílat vlastní BPDU rámce, což je výhoda především z bezpečnostního hlediska (případný útočník nezachytí BPDU rámec a tím pádem se nedozví nic o topologii sítě). Zároveň je to ovšem problém, pokud by byla funkce nesprávně nastavena na port vedoucí k jinému přepínači, jelikož se tímto prakticky zruší Spanning Tree a nebude zabráněno tvorbě smyček v síti.[16]

3.2.4 PVST+

Jak už bylo zmíněno v kapitole o VLAN sítích, pokud chceme zachovat informaci o zařazení do konkrétní VLAN sítě, tak je potřeba označit odchozí rámce. PVST řešilo označování rámců dnes už poněkud zastaralou metodou Cisco Inter-Switch Link zkráceně ISL, která rozšiřuje rámec o 26-bajtovou hlavičku s 10-bitovým VLAN ID + ještě 4 bajtový kontrolní součet na konci rámce. Tento proces poměrně

„nabobtnával“ samotný rámec, takže se jej Cisco rozhodlo nahradit jednodušší metodou která je definována normou 802.1Q.

Ono přidané plus u PVST+ definuje právě použití normy 802.1Q místo ISL. Norma přidává do rámce tzv. „tag“ což je 32-bitová značka obsahující především příslušnost k VLAN. Definovat ovšem jde i prioritu komunikace (například VLAN pro internetovou telefonii bude mít vyšší prioritu). Tento tag může být vložen do rámce už samotnou stanicí, nebo (častěji) přepínačem, do kterého je stanice připojena a který definuje do jaké VLAN stanice spadá. Dnes je tato verze protokolu používána ve výchozím nastavení na všech Cisco přepínačích.[17]

3.2.5 Rapid PVST+

Rapid PVST+ rozšiřuje standard RSTP (původně 802.1w) o funkcionalitu per-VLAN spanning tree, takže opět (jako v případě klasického PVST) existuje pro každou VLAN jiná instance STP. V ostatních parametrech je Rapid PVST+ prakticky shodný s RSTP.[18]

3.3 Multiple Spanning Tree Protocol

Určitou reakcí na Cisco PVST protokol, byl organizací IEEE vytvořen protokol Multiple Spanning Tree, který rozšiřuje RSTP o konfiguraci pro jednotlivé VLAN, podobě jako tomu je u PVST protokolu. PVST vytváří pro každou jednotlivou VLAN jednu spannnig-tree instanci. MSTP na to jde jinak a seskupuje více VLAN sítí do skupin a těm teprve přiděluje spanning-tree instance. Díky tomuto řešení je možné využívat více možných cest a rovněž rozdělovat zátěž, byť staticky. Standardně MST protokol umožňuje vytvořit až 65 různých spanning-tree instancí. MSTP byl původně definován normou 802.1s, nicméně v roce 2003 byl sloučen do normy 802.1Q společně s VLAN sítěmi.[7, 19]

3.3.1 MST region

MST protokol zavedl nové vylepšení v podobě MST regionů, což je skupina určitého počtu přepínačů, které mají stejnou MST konfiguraci (stejně zadané jméno regionu, stejné číslo revize a stejně namopované instance na VLAN sítě). Připodobnit by se to dalo k autonomním systémům, které používá směrovací protokol BGP (Border Gateway Protocol) na 3. vrstně OSI modelu. Skupina těcho přepínačů se potom bude tvářit jako jeden virtuální přepínač a jejich počet není nikterak omezen.[7, 19]

Hranice MST regionů

Jak již bylo zmíněno výše, tak pro zařazení určitého počtu přepínačů do stejného MST regionu je potřeba, aby měly přepínače stejnou konfiguraci. MSTP nepropaguje do BPDU rámce jak přesně jsou VLAN namapovány do konkrétních instancí (pokud mají být namapovány na všech přepínačích stejně, tak to ani není potřeba), ale propaguje kontrolní součet (matematickou funkcí spočítaná hodnota z tabulky kde se mapují VLAN sítě do instancí), číslo revize a název MST konfigurace. Přepínače, které obdrží BPDU rámec, porovnávají hodnotu kontrolního součtu se svoji vlastní a pakliže se liší, tak port, na kterém přepíanč obdržel BPDU rámec leží na hranici MST regionu.[7, 19]

3.3.2 MST instance

Uvnitř **jednoho** MST regionu vytváříme MST instance, což je mapování několika VLAN do skupin (nemáme pro každou VLAN jednu instanci jako u PVST), kterých může být vytvořeno maximálně 65 (číslované 0 - 4094). Ve výchozím nastavení se mapují VLAN do jediné MST instance 0.[7]

3.4 TRILL

Spanning Tree je dnes hojně rozšířeno a postupem času i vývoje, v podobě novějších módů, bylo značně vylepšeno oproti původní verzi definované v normě 802.1D, viz předchozí sekce. Nicméně počátkem tisíciletí a se stále složitějšími sítěmi, vznikla potřeba nahradit mechanizmus Spanning Tree protokolů novým řešením, které by vyřešilo jejich nedostatky a zefektivnilo provoz.

3.4.1 Nedostatky STP

Spanning tree trpí i přes léta vývoje na nedostatky, které se projeví především u větších a komplexnějších L2 sítí, které se v současnosti stále častěji budují.

- Původní STP (802.1D), kterému je udržována kompatibilita i u novějších verzí, měl charakter „fail-open“, kdy mohla nastat taková situace, že se krom aktivních spojů, aktivují i ty záložní (které jsou ve stavu bloking) a dojde tak k zahlcení sítě.
- Nedokonalá stabilita a navíc obtížně dohledatelné příčiny případné nestability – na druhé vrstvě OSI modelu neexistující nástroje *ping* nebo *traceroute* obvykle používané pro diagnostiku na vrstvě třetí.

- Plýtvání zdroji v podobě nevyužití kapacity všech možných cest v síti. Toto částečně řeší PVST, nicméně to vyžaduje detailní plánování sítě a ruční konfiguraci pro každou VLAN.
- Doba konvergence i nejrychlejších případech (u konkrétních módů) v řádu jednotky/jednotek sekund, což může být stále hodně v určitých situacích (např. online gaming)

Hlavní nedostatek STP je především v jeho samotném konceptu fungování, kdy výsledkem konvergence sítě je vždy stromová topologie. Totiž stromová topologie značně omezuje design celé sítě a (už zmíněno výše) nedovoluje využít všech možností a kapacity jednotlivých linek.

Čím větší segment sítě, tím závažnější mohou nedostatky být, takže lze tedy říci, že je lepší stavět menší L2 segmenty sítě, které potom propojovat aktivními L3 prvky, čímž omezíme případné komplikace na danou doménu. Menší L2 segmenty by dále neměly být příliš rozlehlé (jak geograficky tak počtem stanic) a příliš komplexní bez ohledu na technologický vývoj.[20, 22]

Nutnost velké L2 sítě

V dnešní době je ovšem potřeba dělat pravý opak výše zmíněných doporučení a budovat rozlehlé L2 sítě. Typicky u velkých datových center a nebo u internetových providerů (poskytovatelů služeb). Například u data center je dnes nutností mít záložní spojení, pro případ výpadku serverových farem umístěných geograficky daleko od sebe. Případně pakliže je využívána virtualizace a potřeba migrace dat mezi servery v různých lokalitách. U providerů je zase snaha o to se vyhnout stavbě tranzitních L3 sítí, kde bylo by nutné řešit přílišnou režii protokolů, nebo adresaci sítě. Primárně se internetoví provideři musí zaměřit na škálovatelnost sítě (především MAC adres a VLAN indetifikátorů) tak, aby mohly zajistit kvalitu různých služeb různým zákazníkům s ohledem na logické oddělení v síti.[20]

3.4.2 Náhrada STP

Transparent Interconnection of Lots of Links zkráceně TRILL je novým protokolem, který by měl odstranit nedostatky STP. Na novém standardu se postupně začalo pracovat v roce 2004, kdy v IETF vnikla pracovní skupina, která měla za úkol vytyčit si cíle pro nový protokol, který by vyřešil nedostatky Spanning Tree protokolů. Tyto cíle byly popsány v RFC normě 5556. Za zmínku stojí, že v čele této pracovní skupiny tehdy stanula i Radia Joy Perlman což je autorka původního Spanning Tree protokolu. Požadavky na nový protokol:

- Musí mít „fail-closed“ charakter oproti „fail-open“ tzn. pokud nebude stav portu zcela jasný, tak musí být port okamžitě zablokován.

- Zajistit stabilitu v síti, aktivně bránit zacyklení rámců v síti.
- Fungovat na libovolné topologii sítě.
- Využívat přenosové kapacity všech linek v síti a zároveň sestavovat nejvýhodnější cesty k danému cíli – jinými slovy se oprostit od stromové topologie.
- Využívat multipathing tzn. pro přenos od toho samého zdroje k tomu samému cíli používat souběžných linek.
- Získat možnost analýzy sítě na L2 pomocí formalizovaných nástrojů.

Rovněž protokol musí zachovat následující:

- Charakter přepínané L2 sítě vůči koncovými stanicím – musí být zachován standardní ethernetový rámec od zdroje k cíli, tak aby cílová stanice nepoznala rozdíl.
- Možnost posílání broadcastových, multicastových i unicastových zpráv celou síťovou doménou.
- Snadnost konfigurace přepínačů, TRILL přepínač musí být stejně snadno konfigurovatelný, jako jakýkoliv jiný přepínač dosud.
- Možnost koexistence přepínačů TRILL a STP uvnitř jedné L2 infrastruktury.

RFC 5556 dále definuje i název přepínače, který implementuje (pracuje) s TRILL protokolem a sice jako „RBridge“ (zkratka pro „Routing Bridge“). V českém překladu se používá jednoduššího odlišení, kdy pojem „přepínač“ označuje klasický přepínač a „TRILL přepínač“ označuje přepínač implementující TRILL protokol.[21, 22, 23]

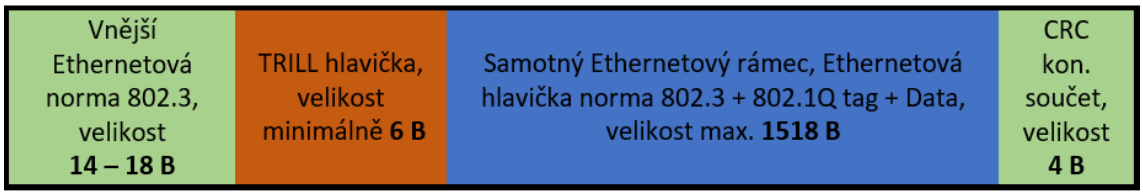
3.4.3 Základní principy

Už z označení přepínačů vyplývá, že TRILL pracuje značně odlišně od STP. Funguje technikou směrování známé ze třetí vrstvy OSI modelu. Využívá protokol IS-IS (který pracuje na Dijkstrově algoritmu) a pomocí něj vypočítává nejvýhodnější cestu k jiným TRILL přepínačům. Další novinky:

- Využití distribučních stromů a kontroly směrování opačnou cestou (Reverse Path Forwarding, zkráceně RPF), pro distribuci provozu více příjemcům.
- Přidání parametru pro omezení životnosti dat (Time to Live - TTL), který je znám z IPv4 packetu.

Aby mohl TRILL protokol fungovat s splňovat výše zmíněné principy, tak je potřeba přenášet další nezbytné informace, na které ovšem není klasický ethernetový rámec stavěn. A protože nesmí dojít k úpravě hlavičky rámce (kvůli kompatibilitě ostatních ethernetových zařízení), tak toto TRILL řeší přidáním nové hlavičky, do které je celý původní rámec zabalen. Pokud by byl umístěn mezi dva TRILL přepínače i klasický přepínač, tak pro zajištění funkčnosti bude celý rámec (včetně TRILL hlavičky) zabalen ještě do další ethernetové hlavičky (zapouzdření metodou MAC-in-MAC).

Pro lepší pochopení je znázorněn upravený ethernetový rámec na následují obrázku 3.1.[21, 23]



Obr. 3.1: Ukázka zapouzdření protokolem TRILL do kompletního rámce

3.4.4 Princip směrování na L2

TRILL přepínač se po fyzické stránce příliš neodlišuje od klasického přepínače. Disponuje ethernetovými porty a rozlišuje dva základní typy portů:

- Porty pro ostatní TRILL přepínače.
- Standardní porty pro koncové stanice.

TRILL přepínače jsou tedy schopny pracovat jak s klasickými ethernetovými rámci na standardních portech, tak taky s rámci, které mají dvojitou ethernetovou hlavičku společně s vloženou TRILL hlavičkou (na portech pro TRILL přepínače).

Ihned po zapojení do sítě a následném zapnutí TRILL přepínače, začne přepínač hledat zda-li má nějaké připojené sousední TRILL přepínače. Pomocí protokolu IS-IS si potom buduje topologickou mapu sítě na základě čehož si vypočítá nejvýhodnější cesty k ostatním TRILL přepínačům, které posléze použije pro první komunikaci. Každý TRILL přepínač si dále vytvoří distribuční strom, který bude každý TRILL přepínač používat na posílání rámců neznámých adresátů přes porty k ostatním TRILL přepínačům. U standardních portů se pro stejný typ rámců použije klasický postup odeslání broadcastem jako používají i klasické přepínače. Učení nových MAC adres probíhá současně s posíláním a přeposíláním rámců opět obdobně jako u klasického přepínače. Novinkou je, že se TRILL přepínače učí i zdrojové adresy v rozbalených TRILL rámcích, což znamená, že je zapsána (jako hodnota „next-hop“) i MAC adresa TRILL přepínače, který provedl prvotní enkapsulaci původního rámce. Toto se hodí především díky možnosti zasílání různých doplňujících informací prvním TRILL přepínačem, které v budoucnu mohou omezit například zbytečné zasílání ARP dotazů, protože koncový TRILL přepínač už bude vědět, kde se cílové stanice nacházejí. Z uživatelského pohledu je používání TRILL protokolu jednodušší díky automatické konfiguraci bez nutnosti toto dělat ručně.[21, 22]

Alternativy protokolu TRILL

V průběhu vývoje protokolu TRILL i po něm když byl standardizován se objevovaly/objevují jeho alternativy ať už v podobě jiných standardů a nebo proprietárních řešení síťových výrobců.

3.4.5 Shortest Path Bridging

Protokol Shortest Path Bridging zkráceně SPB, byl vyvíjen organizací IEEE, která stojí například za protokoly STP. Jeho cíle nebo funkcionalita se prakticky neliší od TRILL, takže i tento protokol funguje na stejném algoritmu směrovacího protokolu IS-IS. SPB byl finálně uveden o pár měsíců později vůči protokolu TRILL, začleněn byl do normy 802.1aq a to konkrétně v březnu roku 2012.[22]

3.4.6 FabricPath

Vlastní implementaci protokolu TRILL vyvinula i firma Cisco, která ji pojmenovala jako FabricPath. Základní funkcionalita protokolu TRILL je stejná, nicméně Cisco si přidalo pár vlastních vylepšení. Jedním z nic je přidaná podpora proprietárního Virtual PortChannel+, které podporují Port-Channel mezi zařízeními. Dále byl rozšířen o podporu více topologií, takže lze omezit přepínače tak aby znali jen určité VLAN sítě a tím pádem dokázali naučit jen omezené množství MAC adres. Toto řešení má obrovskou výhodu například u dnešních virtualizovaných center, jelikož tam může být počet koncových stanic v řádech desítek tisíc. V neposlední řadě ještě u novějších Nexus přepínačů řady 5000 a 7000 podporuje i možnost aby FabricPath běžel v tzv. TRILL kompatibilním módu, tedy aby bylo možné na s přepínačem nakládat jako s klasickým TRILL přepínačem. Samozřejmě bez výše zmíněných funkcí a vylepšení. Na jednom přepínači nelze provozovat zároveň FabricPath a FabricPath v TRILL kompatibilním módu.[22]

3.4.7 QFabric

QFabric je opět proprietární řešení, tentokrát od firmy Juniper Networks, které si klade stejné cíle jako protokol TRILL. Nicméně Juniper odmítl podporovat standardizovaná řešení v podobě protokolů TRILL a SPB.[22]

4 Spanning Tree v praxi

V předchozích částech se tato práce zaměřovala zejména na teoretický popis fungování Spanning Tree protokolu a jeho alternativ v podobě protokolů TRILL a SPB. Tato část práce bude naváže na předešlé a pokusí se ukázat fungování protokolů Spanning Tree v praxi na reálných zařízeních. Konkrétně se tato část zaměří na PVST protokol od společnosti Cisco[1] a především potom na novější varianty v podobě RSTP (resp. RPVST na Cisco přepínačích) a také na MSTP.

4.1 Použitá zařízení

Jak již bylo zmíněno výše, tak v této části budou použity reálné přepínače od předního výrobce síťových prvků a sice přepínače od společnosti Cisco. Jedná konkrétně o modely Cisco Catalyst 3750G (3 kusy) v jejich 28-portových (24 klasických + 4 SFP porty) variantách a jeden model 3560X v 24-portové variantě.

4.2 Základní příkazy

Právě zakoupená zařízení a nebo zařízení, která byla nastavena do továrního nastavení, mají již dnes standardně zapnutý STP v nějaké jeho variantě nebo proprietární formě. Nejinak tomu je i zde u přepínačů Cisco, kde je ve výchozím nastavení spuštěn PVST protokol. Toto je možné ověřit pomocí několika příkazů, která u těchto přepínačů slouží pro kontrolu a dohled nad Spanning Tree protokolem.

Úplně základní příkaz `show spanning-tree` zobrazí informace ohledně Bridge ID (BID) a s tím související informaci, kdo je v dané VLAN nastaven jako Root Bridge a jakou má konkrétní hodnotu Bridge priority. Následují informace o Spanning Tree instancích, a to v případě PVST pro každou z VLAN, pro kterou existuje aktivní instance, které porty jsou zapojeny v rámci dané VLAN, jejich role a aktuální stav. V případě přepínače, který nemá žádnou aktivní instanci (tzn. neúčastní se ST nebo není připojen k některému dalšímu zařízení), bude pochopitelně výpis prázdný (resp. zobrazí se hlášení o neexistující ST instanci). Níže uveden příklad z Cisco přepínače, který je pro výchozí VLAN (VLAN 1) vedený jako Root Bridge:

```
SW1_CIS#show spanning-tree
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    32769
             Address     0008.30f0.0600
```

[1] Společnost Cisco již byl krátce zmíněna na straně 19

```
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
             Address     0008.30f0.0600
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  300 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------
Gi1/0/7             Desg FWD 4         128.7    P2p
Gi1/0/8             Desg FWD 4         128.8    P2p
Gi1/0/14            Desg FWD 19        128.14   P2p
```

Je zde krásně vidět výchozí hodnota BID 32768 + 1 (jako + VLAN 1) a nastavení výchozích časovačů (v sekundách). Informace je možné zobrazit pro jednu konkrétní VLAN i rozmezí, či výběr VLAN sítí (např. VLAN 5-10; 12,13), a to jednoduše doplněním `vlan` resp. `vlan 5-10, 12,13` za `spanning-tree`.

Pakliže je potřeba si zobrazit detailnější výpis informací o ST, lze použít příkaz `show spanning-tree detail`, který zobrazí podrobné údaje o jednotlivých portech a VLAN sítích operujících na těchto portech. Dále je možné vyčíst i například počet přijatých a odeslaných BPDU rámců na konkrétním portu, co jest užitečné například při kontrole správného fungování BPDUfilteru.

```
SW1_CIS#show spanning-tree detail

 VLAN0001 is executing the ieee compatible Spanning Tree protocol
  Bridge Identifier has priority 32768, sysid 1, address 0008.30f0.06
  Configured hello time 2, max age 20, forward delay 15
  We are the root of the spanning tree
  Topology change flag not set, detected flag not set
  Number of topology changes 1 last change occurred 02:16:27 ago
          from GigabitEthernet1/0/7
  Times:  hold 1, topology change 35, notification 2
          hello 2, max age 20, forward delay 15
  Timers: hello 1, topology change 0, notification 0, aging 300

 Port 7 (GigabitEthernet1/0/7) of VLAN0001 is designated forwarding
    Port path cost 4, Port priority 128, Port Identifier 128.7.
    Designated root has priority 32769, address 0008.30f0.0600
    Designated bridge has priority 32769, address 0008.30f0.0600
```

```
   Designated port id is 128.7, designated path cost 0
   Timers: message age 0, forward delay 0, hold 0
   Number of transitions to forwarding state: 1
   Link type is point-to-point by default
   BPDU: sent 4085, received 2
```

Ve chvíli, kdy je potřeba zobrazit podrobnější údaje o rozšíření PVST, jako třeba přehledný výpis, pro které VLAN sítě je daný přepínač Root Bridgem, lze použít příkaz `show spanning-tree summary`. Mimo jiné vypíše i obecně přehlednější informace o počtu VLAN, které tvoří jednotlivé instance a dále jestli jsou defaultně zapnuta rozšíření pro PVST. Příklad:

```
SW1_CIS#show spanning-tree summary
Switch is in pvst mode
Root bridge for: VLAN0001, VLAN1059
Extended system ID           is enabled
Portfast Default             is disabled
PortFast BPDU Guard Default  is disabled
Portfast BPDU Filter Default is disabled
Loopguard Default            is disabled
EtherChannel misconfig guard is enabled
UplinkFast                   is disabled
BackboneFast                 is disabled
Configured Pathcost method used is short

Name               Blocking Listening Learning Forwarding STP Active
---------------- - -------- --------- -------- ---------- ----------
VLAN0001                 0         0        0          3          3
VLAN1059                 0         0        0          2          2
------------------ -------- --------- -------- ---------- ----------
2 vlans                  0         0        0          5          5
```

4.3 PVST a konvergence v praxi

Vzhledem k tomu, že PVST vychází z klasického STP (802.1D), tak nepřekvapí, že doba konvergence se reálně pohybuje v rozpětí 30–50 sekund. Na příkladu níže je ukázáno, že doba konvergence, v případě jedné VLAN (tedy jedné instance) a jednoho portu, trvala přesně 30 sekund. Příklad:

```
SW1_CIS#
*Mar 28 13:35:37.539: set portid: VLAN0010 Gi1/0/14: new port id 800E
```

```
*Mar 28 13:35:37.539: STP: VLAN0010 Gi1/0/14 -> listening
SW1_CIS#
*Mar 28 13:35:39.527: %LINK-3-UPDOWN: Interface GigabitEthernet1/0/14
changed state to up
*Mar 28 13:35:40.534: %LINEPROTO-5-UPDOWN: Line protocol on Interface
GigabitEthernet1/0/14, changed state to up
SW1_CIS#
*Mar 28 13:35:52.546: STP: VLAN0010 Gi1/0/14 -> learning
SW1_CIS#
*Mar 28 13:36:07.553: STP[10]: Generating TC trap for port
GigabitEthernet1/0/14
*Mar 28 13:36:07.553: STP: VLAN0010 Gi1/0/14 -> forwarding
```

Když na přepínači bude větší počet VLAN sítí, a to se v tomto případě jedná o testovacích 50 VLAN (tzn. 50 instancí ST), je čas prakticky stejný, resp. jedná se o 31 sekund. Pro moderní vysokorychlostní sítě je už samozřejmě tato hodnota krajně nedostatečná a je tedy žádoucí využívat novější variantu protokolu (např. RSTP nebo MSTP). Pro úplnost je potřeba ještě uvést, že podobného času konvergence (v tomto případě totožného), bude docíleno i pakliže jeden z přepínačů poběží na RPVST (RSTP) a druhý na klasickém PVST. Pochopitelně zde nedojde k využití novějších mechanismů, jelikož se RPVST se bude muset přizpůsobit staršímu PVST, čímž celý proces kovergování sítě zůstane na úrovni staršího standardu. Demonstruje ukázka níže:

```
SW1_CIS(config)#
*Mar 29 06:02:39.306: RSTP(10): initializing port Gi1/0/8
*Mar 29 06:02:39.306: RSTP(10): Gi1/0/8 is now designated
*Mar 29 06:02:39.315: RSTP(10): transmitting a proposal on Gi1/0/8
*Mar 29 06:02:40.002: RSTP(10): transmitting a proposal on Gi1/0/8
SW1_CIS(config)#
*Mar 29 06:02:41.303: %LINK-3-UPDOWN: Interface GigabitEthernet1/0/8,
changed state to up
*Mar 29 06:02:42.016: RSTP(10): transmitting a proposal on Gi1/0/8
SW1_CIS(config)#
*Mar 29 06:02:42.309: %LINEPROTO-5-UPDOWN: Line protocol on Interface
GigabitEthernet1/0/8, changed state to up
SW1_CIS(config)#
*Mar 29 06:02:44.029: RSTP(10): transmitting a proposal on Gi1/0/8
SW1_CIS(config)#
*Mar 29 06:02:54.313: RSTP(10): Gi1/0/8 fdwhile Expired
SW1_CIS(config)#
*Mar 29 06:03:09.321: RSTP(10): Gi1/0/8 fdwhile Expired
```

```
*Mar 29 06:03:09.321: STP[10]: Generating TC trap for port
GigabitEthernet1/0/8
```

4.3.1 Manipulace s Root Bridgem

Jak již bylo vysvětleno v teoretické části práce, tak Root Bridge (tedy kořenový/hlavní přepínač) se volí podle hodnoty Bridge ID. Právě první 2 bajtovou část BID je možné nastavit, jelikož jde o upravitelnou hodnotu Bridge Priority. U Cisco PVST ji je možné změnit dvěma způsoby.

Prvním způsobem je volit tuto hodnotu manuálně příkazem `spanning-tree vlan X priority Y`, kdy hodnota X odpovídá číslu (případně rozsahu) VLAN sítě a hodnota Y je samotné číslo Bridge Priority jež odpovídá násobkům čísla 4096 (nebo taky číslo 0 což je absolutní priorita).

Druhou možností, je nechat prioritu přiřadit automaticky příkazem `spanning-tree vlan X root primary/secondary`, kdy hodnota X je znovu číslo (nebo rozsah) VLAN a slovo primary nastaví první nejnižší hodnotu, resp. slovo secondary nastaví druhou nejnižší hodnotu Bridge Priority. Logika u tohoto automatického výběru je taková, že příkazy primary a secondary nastaví prioritu jen tak nízko, aby priorita byla nejnižší (resp. druhá nejnižší) mezi ostatními přepínači a stačilo to tedy pro „vyhrání“ volby Root Bridge a zároveň ještě zbylo dostatek místa pro sekundární Root Bridge. Výchozími hodnotami Bridge Priority jsou: **24576** pro `primary` a **28672** pro `secondary`. Je nutné ještě poznamenat, že automatický výběr bere vždy do úvahy celé BID přepínače při počítání priority (tedy i včetně 6 bajtové části s MAC adresou). Ukázka nastavené Bridge Priority při použití `root primary`:

```
SW1_CIS#sh span
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    24577
             Address     0008.30f0.0600
             This bridge is the root
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    24577  (priority 24576 sys-id-ext 1)
             Address     0008.30f0.0600
             Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
             Aging Time  20
```

4.3.2 Rozšíření PVST v praxi

Cisco PVST protokol obsahuje i proprietární rozšíření, která především v době před RSTP a novějšími protokoly, byla potřeba pro docílení rychlejší doby konvergence (PortFast) a k zabezpečení proti nechtěnému přijímání BPDU rámců (BPDUguard/filter). Ačkoliv byla rozšíření vytvořena pro PVST protokol, tak najdou využití i u novějších protokolů, kde už je sice doba konvergence výrazně lepší, ale například zabezpečení portu (BPDUguardem nebo filterem) je stále aktuální a doporučované. Pochopitelně jelikož se jedná o proprietární řešení, tak využití je výhradně omezeno na přepínače spol. Cisco. Teoreticky byla rozšíření popsána v kapitole 3.2 a to konkrétně na stranách 20 a 21. Prakticky jsou jejich použití ukázána níže i včetně zajímavostí z praxe.

PortFast

Díky rozšíření PortFast je možné přeskočit standardní stavy původního ST, a tím urychlit možnost připojení především pro koncové stanice případě také servery. PortFast lze nakonfigurovat buď ručně na konkrétní port příkazem `spanning-tree portfast` anebo příkazem `spanning-tree portfast default`, kdy se rozšíření nastaví na všechny porty vyjma portů, které jsou v režimu trunk. Zajímavé je jednak to, že se portfast nastaví i na port, který není v režimu access (kde by to principiálně dávalo největší smysl), ale rovněž to, že sám přepínač zobrazuje po zadání příkazu na portfast default také toto upozornění:

```
SW1_CIS(config)#spanning-tree portfast default
%Warning: this command enables portfast by default on all interfaces.
 You should now disable portfast explicitly on switched ports leading
 to hubs, switches and bridges as they may create temporary bridging
 loops.
```

Z textu upozornění by se mohlo zdát, že se PortFast nastaví opravdu na každý interface (včetně trunk portu) a je na administrátorovi, aby jej manuálně vypnul na trunk spojích k dalším přepínačům. Nicméně toto bylo vyvráceno při praktické konfiguraci na zařízení C3750G a verzi firmwaru 12.2(58)SE2.

V případě, že by bylo z nějakého důvodu potřeba nastavit PortFast rozšíření i na trunk spoj (např. pokud je na konci spoje server), tak to lze učinit příkazem `spanning-tree portfast trunk`.

BPDUguard a BPDUfilter

Dalšími rozšířeními, které mohou být nastaveny buď společně s předchozím PortFast, nebo také samostatně přímo na port, jsou BPDUguard a BPDUfilter. Nastavení na

konkrétní port se provede jednoduše příkazem `spanning-tree bpduguard enable` (resp. `bpdufilter`) přímo na daném portu. Druhým způsobem je nastavit BPDUguard/filter hromadně na všechny porty, na kterých je už nastaveno rozšíření PortFast, a to příkazem `spanning-tree portfast bpduguard default` (resp. bpdufilter). Zde dává smyl využití rozšíření PortFast i u novějších protokolů, pakliže je PortFast nutným pro hromadné nastavení BPDUguardu/filteru na koncové porty.

Ukázka níže demonstruje zablokování portu (port přejde do error-disabled stavu) s nastaveným BPDUguard poté, co na něj přijde BPDU rámec:

```
SW1_CIS#
*Mar 31 00:55:49.451: set portid: VLAN0001 Gi1/0/24: new port id 8018
*Mar 31 00:55:49.451: STP: VLAN0001 Gi1/0/24 -> listening
*Mar 31 00:55:49.652: %SPANTREE-2-BLOCK_BPDUGUARD: Received BPDU
on port Gi1/0/24 with BPDU Guard enabled. Disabling port.
SW1_CIS#
*Mar 31 00:55:49.652: %PM-4-ERR_DISABLE: bpduguard error detected on
Gi1/0/24, putting Gi1/0/24 in err-disable state
SW1_CIS#
```

Zatímco BPDUguard funguje čistě jako ochrana proti nežádoucím BPDU rámcům, tak BPDUfillter slouží spíš jako prevence a po jeho nastavení se stará o filtrování BPDU rámců (tzn. žádné na port nevysílá a příchozí zahazuje). Zajímavost z praxe je, že při nastavení přímo na port dojde k okamžitému zastavení posílání rámců BPDU, ale při globálním nastavení (společně s PortFast) k tomuto zastavení dojde až po několika odeslaných rámcích. Toto bylo ověřeno opět na přepínači C3750G s verzí firmwaru 12.2(58)SE2.

4.4 RPVST a konvergence v praxi

Teprve v případě, že je na zařízeních nastaveno shodně RPVST, dojde k využití jeho potenciálu a při stejné jedné instanci (jako v případě testování PVST). Doba konvergence odpovídá času mírně přesahujícím 1 sekundu, přičemž PVST protokol na stejnou operaci potřeboval standardních 30 sekund, a to je skutečně značný rozdíl. Demonstruje ukázka níže:

```
SW1_CIS#
*Apr 6 13:49:00.945: RSTP(1): initializing port Gi1/0/7
*Apr 6 13:49:00.945: RSTP(1): Gi1/0/7 is now designated
*Apr 6 13:49:00.945: RSTP(1): updt roles, received superior bpdu on
                     Gi1/0/7
*Apr 6 13:49:00.945: RSTP(1): Gi1/0/7 is now root port
```

```
*Apr 6 13:49:00.945: RSTP(1): Gi1/0/8 blocked by re-root
*Apr 6 13:49:00.945: RSTP(1): synced Gi1/0/7
*Apr 6 13:49:00.953: RSTP(1): Gi1/0/8 is now designated
*Apr 6 13:49:00.962: RSTP(1): transmitting an agreement on Gi1/0/7 as a
                      response to a proposal
*Apr 6 13:49:00.962: RSTP(1): transmitting a proposal on Gi1/0/8
*Apr 6 13:49:00.970: RSTP(1): received an agreement on Gi1/0/8
*Apr 6 13:49:01.146: RSTP(1): updt roles, received superior bpdu on
                      Gi1/0/8
*Apr 6 13:49:01.146: RSTP(1): Gi1/0/8 is now alternate
```

Mírně pomalejší již RPVST protokol bude ve chvíli, kdy na přepínači poběží více VLAN sítí a tím pádem i více instancí. V tomto konkrétním případě je nakonfigurováno na přepínači 50 VLAN sítí a shodně tedy i 50 ST instancí. Následující příklad se zaměří na dobu potřebnou pro konvergenci sítě na přepínači, který navázal opětovnou komunikaci s primárním Root Bridgem a bude měnit role portů. Příklad výpisu[2] debuggu Spannig Tree:

```
SW4_CIS#
*Apr 6 13:26:02.479: %LINK-3-UPDOWN: Interface GigabitEthernet0/23, changed state
                     to up
*Apr 6 13:26:02.739: RSTP(1): updt roles, received superior bpdu on Gi0/24
*Apr 6 13:26:02.747: RSTP(1): synced Gi0/24
*Apr 6 13:26:02.747: RSTP(2): updt roles, received superior bpdu on Gi0/24
*Apr 6 13:26:02.747: RSTP(2): synced Gi0/24
...
*Apr 6 13:26:02.797: RSTP(49): updt roles, received superior bpdu on Gi0/24
*Apr 6 13:26:02.797: RSTP(49): synced Gi0/24
*Apr 6 13:26:02.797: RSTP(50): updt roles, received superior bpdu on Gi0/24
*Apr 6 13:26:02.797: RSTP(50): synced Gi0/24
*Apr 6 13:26:02.797: RSTP(1): transmitting an agreement on Gi0/24 as a
                              response to a proposal
...
*Apr 6 13:26:02.814: RSTP(49): transmitting an agreement on Gi0/24 as a
                     response to a proposal
*Apr 6 13:26:02.814: RSTP(50): transmitting an agreement on Gi0/24 as a
                     response to a proposal
*Apr 6 13:26:03.586: RSTP(1): initializing port Gi0/23
*Apr 6 13:26:03.586: RSTP(1): Gi0/23 is now designated
*Apr 6 13:26:03.586: RSTP(2): initializing port Gi0/23
*Apr 6 13:26:03.586: RSTP(2): Gi0/23 is now designated
...
```

[2] Ačkoliv se to nemusí na první pohled zdát, tak je výpis výrazně zkrácen vzhledem k opravdu velkému množství informací, které se vypisuje pro 50 instancí RPVST

```
*Apr 6 13:26:03.611: RSTP(49): initializing port Gi0/23
*Apr 6 13:26:03.611: RSTP(49): Gi0/23 is now designated
*Apr 6 13:26:03.619: RSTP(50): initializing port Gi0/23
*Apr 6 13:26:03.619: RSTP(50): Gi0/23 is now designated
*Apr 6 13:26:03.661: RSTP(1): transmitting a proposal on Gi0/23
*Apr 6 13:26:03.661: RSTP(2): transmitting a proposal on Gi0/23
...
*Apr 6 13:26:03.678: RSTP(49): transmitting a proposal on Gi0/23
*Apr 6 13:26:03.678: RSTP(50): transmitting a proposal on Gi0/23
*Apr 6 13:26:03.972: RSTP(1): updt roles, received superior bpdu on Gi0/23
*Apr 6 13:26:03.972: RSTP(1): Gi0/23 is now root port
*Apr 6 13:26:03.980: RSTP(1): Gi0/24 blocked by re-root
*Apr 6 13:26:03.980: RSTP(1): synced Gi0/23
*Apr 6 13:26:03.980: RSTP(1): Gi0/24 is now alternate
...
*Apr 6 13:26:04.022: RSTP(50): updt roles, received superior bpdu on Gi0/23
*Apr 6 13:26:04.022: RSTP(50): Gi0/23 is now root port
*Apr 6 13:26:04.022: RSTP(50): Gi0/24 blocked by re-root
*Apr 6 13:26:04.022: RSTP(50): synced Gi0/23
*Apr 6 13:26:04.022: RSTP(50): Gi0/24 is now alternate
*Apr 6 13:26:04.299: RSTP(1): transmitting an agreement on Gi0/23 as a
                     response to a proposal
*Apr 6 13:26:04.299: RSTP(2): transmitting an agreement on Gi0/23 as a
                     response to a proposal
...
*Apr 6 13:26:04.626: %LINEPROTO-5-UPDOWN: Line protocol on Interface
                     GigabitEthernet0/23, changed state to up
SW4_CIS#
```

Z výpisu je patrné, že už přibližně za sekundu přechází port G0/23 (port vedoucí k Root Bridgi) do role designated a (ve výpisu zahrnuto není, ale bylo ověřeno) už je provozu schopný. Nicméně ještě se musí port ustanovit jako root port (vzhledem k tomu, že vede k Root Bridgi) a port G0/24 určit jako alternate port. Celá doba konvergence tedy přesahuje dobu 2 sekund přibližně o 200 ms, což je stále ovšem výborný výsledek v porovnání se starším PVST.

4.5 MSTP v praxi

Po přepnutí do MST protokolu si přepínač nastaví výchozí hodnoty což znamená, že se nastaví výchozí revizní číslo (tj. hodnota[3] označující konkrétní MST region), dále se všechny VLAN sítě namapují do jediné instance (defaultně instance 0) a v neposlední řadě se nastaví, resp. spíš nenastaví jméno MST regionu (region zůstane nepo-

[3]Většinou používaná jako rychlé rozpoznání pro administrátory

jmenován). Toto nastavení je možno ověřit příkazem `sh span mst configuration`, opět ukázka zde:

```
SW1_CIS#sh span mst configuration
Name      []
Revision  0     Instances configured 1

Instance  Vlans mapped
--------  ----------------------------------------------------
0         1-4094
-------------------------------------------------------------------
```

Zařízení, která mají spolu komunikovat v jednom MST regionu, musí mít shodně nastaveny všechny tři zmíněné parametry. Název regionu je „case sensitive“ tzn. záleží na velikosti písmen a je potřeba jej zadat správně (př. *BARS* a *bars* jsou dva rozdílné regiony). Dále je třeba zadat stejné revizní číslo a nakonec musí být i stejně namapované VLAN sítě v regionu. Konfigurace MST protokolu se provádí příkazem `spanning-tree mst configuration`, který vyvolá nabídku pro nastavení výše zmíněných parametrů. Dokud se **neopustí** nabídka s nastavením, tak se změny neaplikují. Důvodem, proč nedojde k aplikování změn ihned po zadání příkazů (jak je obecně u Cisca mimochodem zvykem), ale až po opuštění nabídky je, aby se zadaná konfigurace mohla ještě ověřit a zda-li opravdu parametry odpovídají zadaným a někde se nestala chyba. Ke kontrole slouží dvojice příkazů, z nichž první `show current` zobrazí vždy aktuální konfiguraci (aktuálně používaná/běžící konfigurace) a druhý `show pending` zobrazí konfiguraci, tak jak bude vypadat právě po opuštění nabídky. Příklad nastavení parametrů MST regionu a ověření:

```
SW1_CIS(config)#spanning-tree mst configuration
SW1_CIS(config-mst)#name BARS
SW1_CIS(config-mst)#revision 2
SW1_CIS(config-mst)#instance 1 vlan 10-30
SW1_CIS(config-mst)#show current
Current MST configuration
Name      [FIRST]
Revision  1     Instances configured 2

Instance  Vlans mapped
--------  ----------------------------------------------------
0         31-4094
1         1-30
-------------------------------------------------------------------
```

```
SW1_CIS(config-mst)#show pending
Pending MST configuration
Name      [BARS]
Revision  2     Instances configured 2

Instance  Vlans mapped
--------  ----------------------------------------------------
0         31-4094
1         1-30
-------------------------------------------------------------------
SW1_CIS(config-mst)#
```

4.5.1 Manipulace s Root Bridgem

U MST protokolu je možné nastavovat Root Bridge velmi podobně jako u PVST protokolu. Opět lze nastavit prioritu buď manuálně a nebo automaticky, kdy se určuje primární a sekundární Root Brigde. Příkazy jsou strukturou velmi podobné, pochopitelně se nenastavují jednotlivé VLAN (tak jako tomu bylo u PVST), ale MST instance, ve kterých jsou VLAN sítě zařazeny. Pro manuální nastavení na přepínači se použije příkaz `spanning-tree mst X priority Y`, za hodnotu X se dosadí číslo MST instance nebo rozsah více instancí a za Y se opět dosazuje samotná hodnota priority (zde se nastavení neliší od PVST). Automatické nastavení příkazem `spanning-tree mst X root primary` případně `root secondary` určuje priority stejným způsobem jako u PVST. Ověření se provede příkazem `show spanning-tree`, kdy je vidět nastavená priorita, a ještě dodatečná informace, že daný přepínač je skutečně Root Bridgem pro konkrétní instanci.

```
SW2_CIS(config)# do sh span
MST1
  Spanning tree enabled protocol mstp
  Root ID    Priority    24577
             Address     64a0.e773.f080
             This bridge is the root
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    24577  (priority 24576 sys-id-ext 1)
             Address     64a0.e773.f080
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
```

4.5.2 Konfigurace VTP verze 3

VLAN trunking protocol byl zmíněn v teoretické části práce a nyní přijde na řadu jeho praktická implementace v podobě třetí verze, která se vyznačuje především možností synchronizace i ostatních databází kromě standardní VLAN databáze. Toho lze využít pro nastavení MSTP napříč přepínači v celé síti, kdy se nastaví jeden přepínač (do role primárního serveru), na něm se nakonfiguruje MST region, a tato konfigurace se potom rozšíří snadno na ostatní přepínače (v roli klientů nebo sekundárních serverů). Základem ovšem je aby nejprve fungovala synchronizace VLAN databáze napříč všemi přepínači, jelikož ty musí mít aktualizované záznamy s VLAN sítěmi pro jejich úspěšné zařazení do MST instance.

Postup pro vytvoření VTP serveru

Ve výchozím nastavení je VTP zapnut a přepínač je v transparentní roli, doména není nastavena (resp. je nastavena hodnota NULL, tedy nic) a VTP je ve verzi 1. Pro ověření parametrů je k dispozici příkaz `show vtp status`. Následuje úprava parametrů:

```
SW4_CIS(config)#vtp version 3
SW4_CIS(config)#
*Apr  7 14:22:24.048: %SW_VLAN-6-OLD_CONFIG_FILE_READ: Old version 2
VLAN configuration file detected and read OK.  Version 3
    files will be written in the future.

SW4_CIS(config)#vtp domain ABC
Changing VTP domain name from NULL to ABC
SW4_CIS(config)#
*Apr  7 14:22:34.466: %SW_VLAN-6-VTP_DOMAIN_NAME_CHG: VTP domain
name changed to ABC.
```

V tomto případě je nastavena verze VTP 3 a změněna doména na „ABC“. Následně je ještě potřeba udělit přepínači roli VTP serveru což se provede příkazem:

```
SW4_CIS(config)#vtp mode server
Setting device to VTP Server mode for VLANS.
```

U starších verzí VTP (verze 1 a 2) ještě hrálo roli tzv. konfigurační revizní číslo[4]. Pokud je přepínačů v roli serveru více, tak přepínač s nevyšší hodnotou revizního

[4]Revizní číslo označuje počet změn konfigurace, kdykoliv se konfigurace změní, tak se revizní číslo navýší o jedna (začíná na 0)

čísla, bude označen jako přepínač s nejnovější konfigurací, takže se stane primárním serverem VTP domény (tzn. právě takový přepínač, který bude rozhodovat o konfiguraci). Od verze 3 už nicméně revizní číslo tento účel neplní a na nastavení primárního VTP serveru slouží speciální příkaz: `vtp primary`, který napevno nastaví daný přepínač jako primární VTP server. Ostatní přepínače (ať už v roli serveru nebo klienta) se tomu musí přizpůsobit, právě nehledě na revizní číslo. V tomto ohledu je nastavení VTP jednodušší, protože administrátor má tímto větší kontrolu nad tím, kdo se stane primárním serverem v VTP doméně.

VTP server pro VLAN sítě je tedy tímto úspěšně nastaven a obdobným způsobem lze nastavit i VTP klienty, u kterých se jen místo příkazu `vtp mode server` pochopitelně použije příkaz pro `clienta`, případně pro režim `transparent` pokud má například přepínač sloužit jen jako „spojka“ pro další přepínače. Příklad nastaveného primárního VTP serveru:

```
SW1_CIS#sh vtp status
VTP Version capable             : 1 to 3
VTP version running             : 3
VTP Domain Name                 : ABC
VTP Pruning Mode                : Disabled
VTP Traps Generation            : Disabled
Device ID                       : 0008.30f0.0600

Feature VLAN:
--------------
VTP Operating Mode              : Primary Server
Number of existing VLANs        : 57
Number of existing extended VLANs : 1
Maximum VLANs supported locally   : 1005
Configuration Revision          : 3
Primary ID                      : 0008.30f0.0600
Primary Description             : SW1_CIS
MD5 digest                      : 0xFD 0x63 0x62 0xA4 0x4B 0xC6 0x4A 0x58
                                  0x37 0xE8 0x10 0x17 0x3D 0xC8 0x3B 0x1E
```

4.5.3 Konfigurace VTP pro propagaci nastavení MSTP

Za předpokladu, že je již VTP ve verzi 3 nastaven na všech přepínačích, tak lze jen jednoduše přidáním příkazu `vtp mode server mst` (resp. client nebo transparent) připravit synchronizaci konfigurace MSTP. Poslední, co je potřeba určit je, který přepínač bude mít roli primárního mst serveru (tzn. který přepínač bude určovat

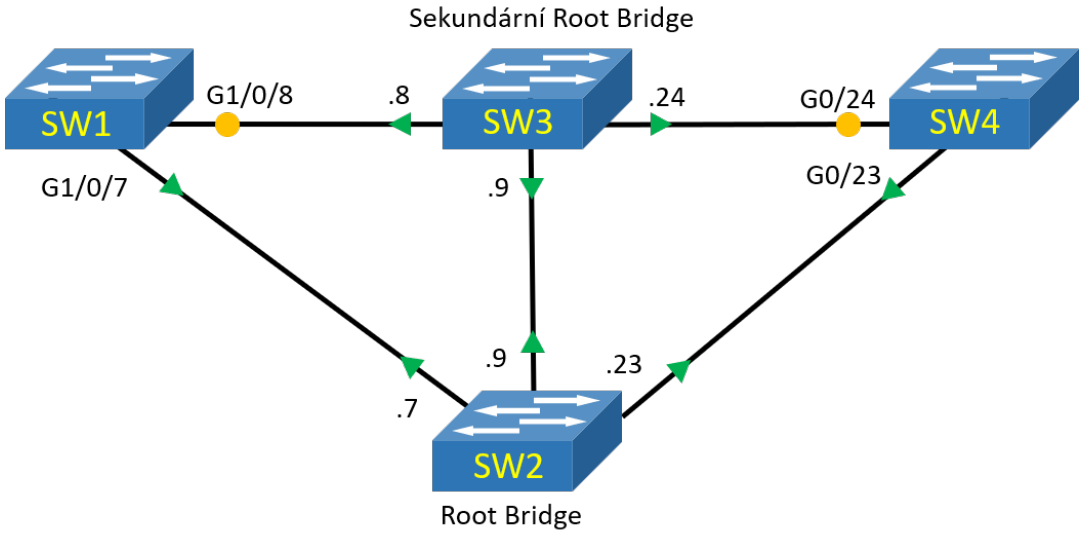
MST konfiguraci), což připadá v naprosté většině případů na primární Root Bridge. Ukázka nastavení:

```
SW1_CIS(config)#vtp mode server mst
Setting device to VTP Server mode for MST.
SW1_CIS(config)#exit
SW1_CIS#vtp primary mst
This system is becoming primary server for feature  mst
No conflicting VTP3 devices found.
Do you want to continue? [confirm]
```

Synchronizaci je poté možné na přepínačích ověřit klasicky již výše zmíněným příkazem `show vtp status`.

4.5.4 Čas konvergence u MSTP

Rychlost konvergence MST protokolu je možné si ověřit na následujícím příkladu. Na obrázku 4.1 je ukázána topologie zapojení jednotlivých přepínačů (všechny přepínače jsou zástupci spol. Cisco) včetně číslování a stavů jednotlivých portů. Zelená šipka značí forwarding stav (port operuje normálně) a oranžová značí stav bloking (kdy je port zablokován Spanning Tree protokolem).



Obr. 4.1: Zapojená topologie s fungujícím MST protokolem

Dále je z obrázku 4.1 patrné, že přepínač SW2 je nastaven jako `root primary`, tedy s nejnižší prioritou v síti což z něj dělá Root Bridge. Přepínač SW3 je potom

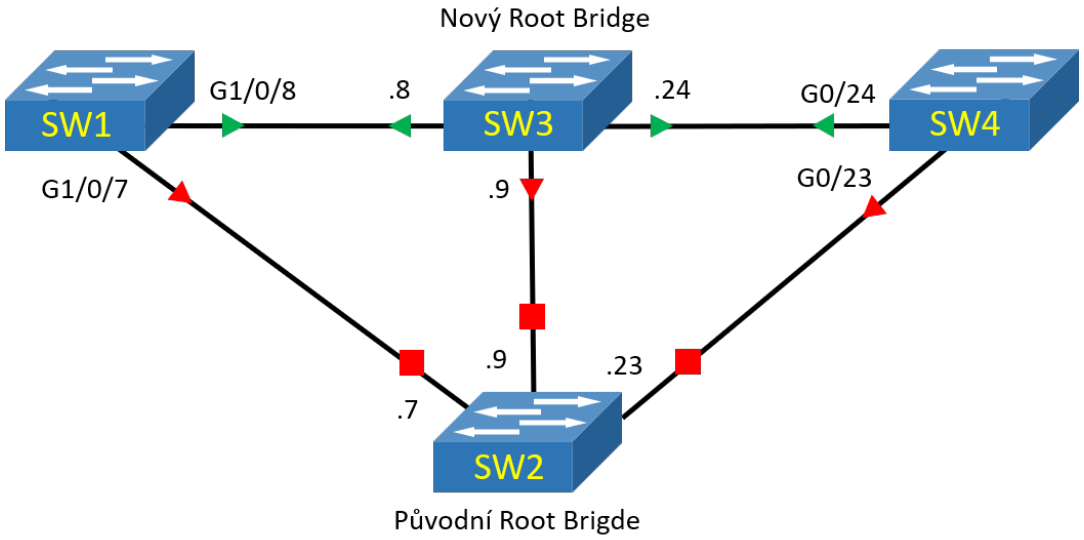
nastaven jako `root secondary`, což znamená, že má druhou nejnižší prioritu a v případě výpadku Root Bridge, zaujme jeho místo. Na přepínačích je nastaven VTP ve verzi 3 se synchronizací MSTP a jako Primary Server je určen SW2, což je i z logiky věci ideální vzhledem k tomu, že se jedná o Root Bridge (ostatní přepínače jsou v režimu klienta). Nakonfigurovány jsou celkem tři MST instance (každá s obsahem přibližně 20 VLAN sítí) plus je aktivní výchozí instance 0 (která obsahuje zbytek nepoužitých VLAN) a v neposlední řadě – porty mezi přepínači jsou v režimu trunk. Klasickým příkazem `show spanning-tree` je možné si ověřit hodnotu Priority pro Root Bridge (SW2), která je rovna 24577 u některé z nakonfigurovaných instancí, a to na kterémkoliv přepínači (níže zobrazen konkrétně SW1).

```
MST1
  Spanning tree enabled protocol mstp
  Root ID    Priority    24577
             Address     64a0.e773.f080
             Cost        20000
             Port        7 (GigabitEthernet1/0/7)
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
             Address     0008.30f0.0600
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------
Gi1/0/7             Root FWD 20000     128.7    P2p
Gi1/0/8             Altn BLK 20000     128.8    P2p
```

Obrázek 4.2 ukazuje simulaci výpadku Root Bridge, a to vypnutím portů 7,9 a 23 na SW2. Dochází ke změně topologie a změna role portů G1/0/8 a G0/24 na SW1, resp. SW4 z alternate do root, vzhledem ke ztrátě původního spojení s původním Root Bridgem a navázání spojení s novým Root Bridgem, kterým se stává SW3. Kkonvergence sítě pro všechny čtyři instance (3 nakonfigurované + 1 výchozí) v tomto případě trvá přibližně 1 sekundu pro SW4, resp. přibližně 2 sekundy pro SW1 (od první změny role portu po definitivní oznámení o vypnutí původního Root portu).



Obr. 4.2: Změna topologie po výpadku Root Brigde

```
SW1_CIS#
*Apr 12 13:43:24.746: MST[1]: updt roles, root port Gi1/0/7 going down
*Apr 12 13:43:24.746: MST[1]: Gi1/0/8 is now root port
*Apr 12 13:43:24.746: MST[2]: updt roles, root port Gi1/0/7 going down
*Apr 12 13:43:24.746: MST[2]: Gi1/0/8 is now root port
*Apr 12 13:43:24.754: MST[3]: updt roles, root port Gi1/0/7 going down
*Apr 12 13:43:24.754: MST[3]: Gi1/0/8 is now root port
*Apr 12 13:43:24.829: STP[1]: Generating TC trap for port Gi1/0/8
*Apr 12 13:43:24.838: STP[2]: Generating TC trap for port Gi1/0/8
*Apr 12 13:43:24.838: STP[3]: Generating TC trap for port Gi1/0/8
SW1_CIS#
*Apr 12 13:43:24.947: MST[1]: updt roles, received superior bpdu on
Gi1/0/8
*Apr 12 13:43:24.947: MST[2]: updt roles, received superior bpdu on
Gi1/0/8
*Apr 12 13:43:25.106: MST[3]: updt roles, received superior bpdu on
Gi1/0/8
SW1_CIS#
*Apr 12 13:43:25.744: %LINEPROTO-5-UPDOWN: Line protocol on Interface
GigabitEthernet1/0/7, changed state to down
SW1_CIS#
*Apr 12 13:43:26.801: %LINK-3-UPDOWN: Interface GigabitEthernet1/0/7,
changed state to down
```

Rovněž se změnila i hodnota Priority na 28673 pro Root Bridge, vzhledem k

tomu, že se jím stal SW3, což je možné ověřit následujícím výpisem:

```
MST1
  Spanning tree enabled protocol mstp
  Root ID    Priority    28673
             Address     2c3f.3802.0080
             Cost        20000
             Port        8 (GigabitEthernet1/0/8)
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
             Address     0008.30f0.0600
             Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
Interface           Role Sts Cost      Prio.Nbr Type
------------------- ---- --- --------- -------- --------------------------
Gi1/0/8             Root FWD 20000     128.8    P2p
```

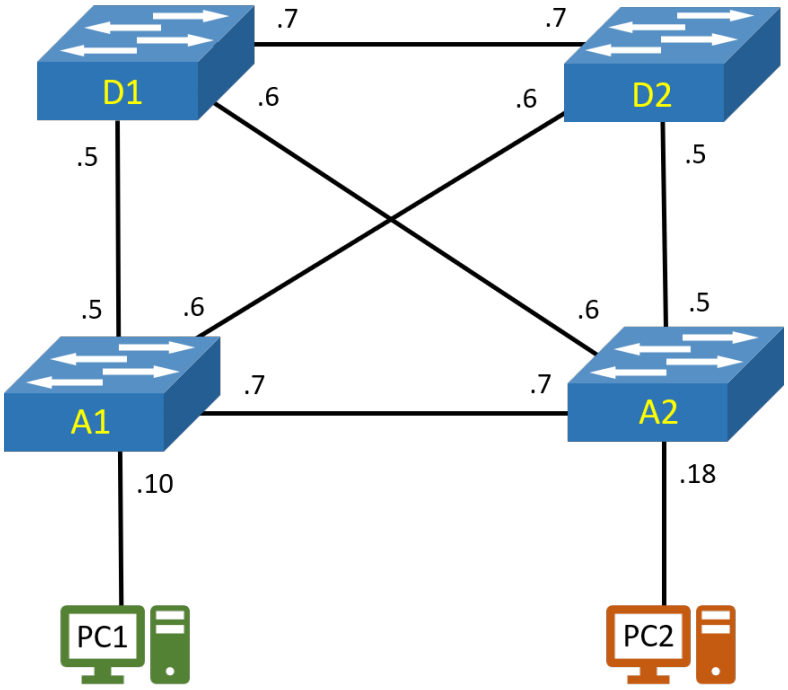
Po opětovném zprovoznění původního Root Bridge (zapnutí portů na SW2) se změní topologie zpět na původní. To znamená, že porty G1/0/8 a G0/24 se přepnou zpátky do role alternate (tedy záložní cesty k Root Bridgi) a původní porty G1/0/7 a G0/23 se zase stanou root porty ve forwarding stavu. Celá operace (konvergence) od naběhnutí portu do aktualizace rolí na portech, trvá shodně u obou přepínačů (SW1 a SW4) přibližně 2 sekundy viz ukázka:

```
*Apr 12 13:48:36.164: %LINK-3-UPDOWN: Interface GigabitEthernet0/23,
changed state to up
SW4_CIS#
*Apr 12 13:48:37.280: MST[0]: Gi0/23 is now designated port
*Apr 12 13:48:37.280: MST[0]: Gi0/23 becomes designated - clearing
BOUNDARY flag
*Apr 12 13:48:37.280: MST[1]: Gi0/23 is now designated port
*Apr 12 13:48:37.280: MST[2]: Gi0/23 is now designated port
*Apr 12 13:48:37.297: MST[3]: Gi0/23 is now designated port
*Apr 12 13:48:37.297: MST[1]: updt roles, received superior bpdu on Gi0/24
*Apr 12 13:48:37.297: MST[2]: updt roles, received superior bpdu on Gi0/24
*Apr 12 13:48:37.297: MST[3]: updt roles, received superior bpdu on Gi0/24
*Apr 12 13:48:38.303: %LINEPROTO-5-UPDOWN: Line protocol on Interface
GigabitEthernet0/23, changed state to up
*Apr 12 13:48:38.370: MST[0]: updt roles, received superior bpdu on Gi0/23
*Apr 12 13:48:38.370: MST[0]: Gi0/23 is now root port
*Apr 12 13:48:38.370: MST[0]: Gi0/24 is now alternate port
*Apr 12 13:48:38.370: MST[1]: updt roles, CIST reconcile on Gi0/23
```

```
*Apr 12 13:48:38.379: MST[2]: updt roles, CIST reconcile on Gi0/23
*Apr 12 13:48:38.379: MST[3]: updt roles, CIST reconcile on Gi0/23
*Apr 12 13:48:38.379: MST[1]: updt roles, received superior bpdu on Gi0/23
*Apr 12 13:48:38.379: MST[1]: Gi0/23 is now root port
*Apr 12 13:48:38.379: MST[1]: Gi0/24 is now alternate port
*Apr 12 13:48:38.379: MST[2]: updt roles, received superior bpdu on Gi0/23
SW4_CIS#
*Apr 12 13:48:38.379: MST[2]: Gi0/23 is now root port
*Apr 12 13:48:38.379: MST[2]: Gi0/24 is now alternate port
*Apr 12 13:48:38.379: MST[3]: updt roles, received superior bpdu on Gi0/23
*Apr 12 13:48:38.379: MST[3]: Gi0/23 is now root port
*Apr 12 13:48:38.379: MST[3]: Gi0/24 is now alternate port
```

5 Tvorba laboratorní úlohy

Laboratorní úloha navazuje na teoretickou i praktickou část práce a studenti se zde seznámí hlavně s konfigurací Spanning Tree protokolu na platformě Cisco. V úloze studenti postupují podle návodu, kdy nejprve nakonfigurují základní nastavení přepínačů, jako trunk spoje, VTP protokol atd. Posléze konfigurují jednotlivé verze Spanning Tree od nejstarší, po tu nejnovější a na konci by měli být schopni poznat rozdíly mezi těmito verzemi. Většina ověřování probíhá přímo na přepínačích, a to ať už podle statických výpisů běžně dostupných pomocí příkazů `show`, nebo pomocí dynamických výpisů (debuggů), které si spustí na některém z přepínačů. Některé ověření probíhají i přes příkazový řádek na počítači, případně i s využitím programu Wireshark.



Obr. 5.1: Topologie zapojení laboratorní úlohy

Obrázek 5.1 ukazuje jak si studenti mají zapojit laboratorní úlohu. Úloha je realizována na čtyřech přepínačích, které jsou konfigurovány ze dvou PC stanic, na kterých je ještě zároveň umístěno virtualizační prostředí potřebné pro zaintegrování stanic do topologie. Samotný návod laboratorní úlohy je umístěn v příloze jako příloha A.

Závěr

Tato práce měla za úkol zmapovat vývoj protokolu Spanning Tree a jeho možných alternativ v dnešní době. Na začátku práce jsem se věnoval základním Ethernetovým zařízením tak abychom si mohli ujasnit na čem vlastně Spannig Tree funguje. Dále byla probrána problematika smyček v síti, včetně jejich vznikání a co dokáže taková smyčka způsobit. Poukázalo se na to, proč jsou vlastně nežádoucí a je potřeba je eliminovat.

Ve druhé kapitole byl podrobně rozebrán původní Spannig Tree protokol, bylo objasněno, jak funguje a jak si volí potřebné parametry a byly zmíněny výhody a nevýhody tohoto protokolu.

Ve třetí kapitole se práce zaobírala evolucí a kompletním zmapováním protokolů z rodiny Spanning Tree, jako jsou například Rapid Spanning Tree Prokotol nebo Multiple Spanning Tree protokol. Představeny byly i alternativy v podobě protokolů TRILL a SPB, které pracujících za stejným účelem, jako protokol Spanning Tree, ale fungují poněkud odlišně. Zmínily si i proprietarní řešení různých síťových výrobců včetně výhod a úskalí jednotlivých variant.

Čtvrtá kapitola se zaměřovala především na praktické ukázání fungování a konfigurování protokolů Spanning Tree na reálných zařízeních společnosti Cisco. Byly zde dopodrobna ukázány příklady konfigurace, ukázky výpisů konfigurací a porovnávána rychlost konvergence sítě pro jednotlivé protokoly.

Poslední pátá kapitola obsahuje popis laboratorní úlohy, která je součástí příloh. Tato úloha byla vytvořena především pro budoucí studenty předmětu BARS, kteří si díky ní mohou vyzkoušet pracovat s protokoly Spanning Tree, zkoumat rozdíly jednotlivých verzí a rozšířit si tak svoje teoretické znalosti.

Literatura

[1] *Evolution of the Spanning Tree Protocol* [online]. [cit. 2019-11-04]. Dostupné z: <http://www.force10networks.com/whitepapers/pdf/F10_wp19_v1%201.pdf>

[2] BOUŠKA, Petr. *Víte, jak pracuje switch?* Samuraj-cz. [online]. [cit. 2019-11-04]. Dostupné z: <https://www.samuraj-cz.com/clanek/vite-jak-pracuje-switch/>

[3] BOUŠKA, Petr. *Cisco IOS 9 - Spanning Tree Protocol* Samuraj-cz. [online]. [cit. 2019-11-04]. Dostupné z: <https://www.samuraj-cz.com/clanek/cisco-ios-9-spanning-tree-protocol/>

[4] BRACHMANN, Steve *The Evolution of the Internet: The spanning tree protocol, a major achievement in Internet routing.* Ipwatchdog [online]. 2016, [cit. 2019-12-21]. Dostupné z: <https://www.ipwatchdog.com/2016/02/04/spanning-tree-protocol-internet-routing/id=65051/>

[5] VODEHNAL, Stanislav. *Návrh datových sítí poskytovatelů připojení k Internetu.* [online]. Brno, 2016 [cit. 2019-12-21]. Dostupné z: <https://www.vutbr.cz/www_base/zav_prace_soubor_verejne.php?file_id=129665> Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Vedoucí práce Doc. Ing. Vít Novotný, Ph.D.

[6] MANDÁK, Vladimír. *Vizualizace Spanning Tree Protocol.* [online]. Plzeň, 2015 [cit. 2019-12-21]. Dostupné z: <https://otik.zcu.cz/bitstream/11025/17873/1/DPA12N0030P.pdf> Diplomová práce. Západočeská univerzita v Plzni, Fakulta aplikovaných věd. Vedoucí práce Ing. Michal Petrovič.

[7] BOUŠKA, Petr. *Cisco IOS 10 - Rapid Spanning Tree Protocol* Samuraj-cz. [online]. [cit. 2019-12-21]. Dostupné z: <https://www.samuraj-cz.com/clanek/cisco-ios-10-rapid-spanning-tree-protocol/>

[8] ROHÁČ, Michal a Roman KUBÍN, *Rapid Spanning Tree Protocol (802.1w)* [online]. Ostrava, 2005 [cit. 2019-12-21]. Dostupné z: <http://www.cs.vsb.cz/grygarek/SPS/projekty0405/RSTP-Kubin-Rohac.pdf> Projekt. VŠB - Technická univerzita Ostrava, FEI.

[9] BOHÁČ, Daniel a Jakub PRÁŠIL, *Rapid Spanning Tree Protocol* [online]. Ostrava, 2015 [cit. 2019-12-21]. Dostupné z: <http://wh.cs.vsb.cz/sps/images/7/72/Rstpm.pdf> Projekt. VŠB - Technická univerzita Ostrava, FEI.

[10] *Cisco Overview* Cisco.com [online]. [cit. 2020-03-29]. Dostupné z: <https://newsroom.cisco.com/overview/>

[11] *Networking Academy* Www.netacad.com [online]. [cit. 2020-03-29]. Dostupné z: <https://www.netacad.com/about-networking-academy>

[12] BOUŠKA, Petr *VLAN - Virtual Local Area Network* Samuraj-cz. [online]. 2007, [cit. 2019-11-15]. Dostupné z: <https://www.samuraj-cz.com/clanek/vlan-virtual-local-area-network//>

[13] *Catalyst 2960-X Switch VLAN Configuration Guide, Cisco IOS Release 15.0(2)EX.* Cisco.com [online]. [cit. 2020-06-02]. Dostupné z: <https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst2960x/software/15-0_2_EX/vlan/configuration_guide/b_vlan_152ex_2960-x_cg/b_vlan_152ex_2960-x_cg_chapter_010.html>

[14] KORITAR, Lukáš a Jan WASSERBAUER *Možnosti protokolu Cisco VTP v3* [online]. [cit. 2020-06-02]. Dostupné z: <http://wh.cs.vsb.cz/sps/images/archive/9/9e/20150520163705!Vtpv3m.pdf>

[15] *Understanding and Configuring the Cisco UplinkFast Feature.* Cisco [online]. 2008 [cit. 2019-12-21]. Dostupné z: <https://www.cisco.com/c/en/us/support/docs/lan-switching/spanning-tree-protocol/10575-51.html>

[16] *BPDU GUARD AND BPDU FILTER* In: Youtube [online], 2016 [cit. 2019-12-21]. Dostupné z: <https://www.youtube.com/watch?v=-jLFtxgmA2g>

[17] MOLENAAR, Rene. *MOLENAAR, Rene. 802.1Q Encapsulation Explained.* NetworkLessons [online]. [cit. 2019-12-21]. Dostupné z: <https://networklessons.com/switching/802-1q-encapsulation-explained>

[18] *Cisco Nexus 5000 Series NX-OS Software Configuration Guide* Cisco [online] [cit. 2019-12-21]. Dostupné z: <https://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus5000/sw/configuration/guide/cli/CLIConfigurationGuide/RPVSpanningTree.html>

[19] KRÁLÍK, Ondřej a Jiří ŠEBESTA *Multi-instance Spanning Tree - norma 802.1s* [online]. Ostrava, - [cit. 2019-12-21]. Dostupné z: <http://www.cs.vsb.cz/grygarek/SPS/projekty0405/MST/mst.htm> Projekt. VŠB - Technická univerzita Ostrava, FEI.

[20] MATUŠKA, Miroslav. *TRILL: Konečně náhrada za Spanning Tree?* LUPA [online]. 2010, 1 [cit. 2019-12-21]. Dostupné z: <https://www.lupa.cz/clanky/trill-konecne-nahrada-za-spanning-tree/>

[21] MATUŠKA, Miroslav. *TRILL 2. část – Základní principy* LUPA [online]. 2010, 1 [cit. 2019-12-21]. Dostupné z: <https://www.lupa.cz/clanky/trill-2-cast-zakladni-principy/>

[22] SURÁK, Adam. *Transparent Interconnection of Lots of Links (TRILL) jako náhrada Spanning Tree* [online]. Ostrava, - [cit. 2019-12-21]. Dostupné z: <http://wh.cs.vsb.cz/sps/images/8/88/TRILL.pdf> Projekt. VŠB - Technická univerzita Ostrava, FEI.

[23] KMONÍČEK, Tomáš. *Analýza využití protokolu TRILL v podnikové síti* [online]. Pardubice, 2015 [cit. 2019-12-21]. Dostupné z: <https://dk.upce.cz/bitstream/handle/10195/60413/KmonicekT_AnalyzaVyuziti_JH_2015.pdf?sequence=1&isAllowed=y> Diplomová práce. UNIVERZITA PARDUBICE, Fakulta elektrotechniky a informatiky. Vedoucí práce Mgr. Josef Horálek, Ph.D.

Seznam symbolů, veličin a zkratek

BID Bridge ID
BPDU Bridge Protocol Data Unit
CAM Content-addressable memory
DEC Digital Equipment Corporation
EIGRP Enhanced Interior Gateway Routing Protocol
IEEE Institute of Electrical and Electronics Engineers
IETF Internet Engineering Task Force
IS-IS Intermediate System to Intermediate System
ISL Inter-Switch Link
LAN Local Area Network
MAC Media Access Control
MIT Massachusetts Institute of Technology
MSTP Multiple Spanning Tree Protocol
OSI Open Systems Interconnection
OSPF Open Shortest Path First
PVST Per-VLAN Spanning Tree
RSTP Rapid Per-VLAN Spanning Tree
SPB Shortest Path Bridging
STA Spanning Tree Algorithm
STP Spanning Tree Protocol
TRILL Transparent Interconnection of Lots of Links
VTP VLAN Trunking Protocol

Seznam příloh

A Laboratorní úloha – konfigurace STP na platformě Cisco

A.1 Cíl

Cílem laboratorní úlohy je blíže seznámit studenty s protokolem Spannig Tree, a především s jeho novějšími variantami v podobě protokolů (RSTP a MSTP) a s jejich konfigurací na přepínačích Cisco.

A.2 Vybavení pracoviště

2x Počítač s OS Windows 10 ve virtualizačním prostředí VirtualBox, 3x přepínač Cisco Catalyst WS-C3750G-24TS-S1U, 1x Cisco Catalyst WS-C3560X-24P-S.

A.3 Úkoly

1. Připravit si topologii a zprovoznit základní konfiguraci včetně VLAN sítí a VTP.
2. Krátce prozkoumat klasický PVST protokol a rozšíření toho protokolu.
3. Přepnout na RPVST a krátce si ověřit rozdíl proti PVST
4. Přepnout zařízení do MSTP a nastavit MST region
5. Nastavit synchronizaci VTP v3 s MST databází
6. Analyzovat chování MSTP při změnách topologie

A.4 Teoretický úvod

Dnešní počítačové sítě již zpravidla fungují výhradně na technologii Ethernet a tvoří je různé síťové prvky. Páteřní sítě větších poskytovatelů k internetu tvoří spíše síťové prvky pracující na 3. vrstvě OSI modelu (jako například směrovače). Zatímco distribuční a přístupovou vrstvu, tvoří většinou výhradně prvky 2. vrstvy a sice přepínače. Ty dokáží třídit provoz pro konkrétní adresáty a na konkrétní porty i díky práci s MAC adresami adresátů, které jsou obsaženy v příchozích rámcích. Přepínač se postupně učí kam má posílat zprávu pro konkrétní adresáty podle toho, na kterém portu má připojenou konkrétní MAC adresu. V případě, že adresáta ještě nezná (ještě s ním nekomunikoval), tak zprávu odešle hromadně na všechny porty a čeká, až mu z jednoho z nich přijde odpověď. Problémem tohoto řešení je ovšem to,

že pokud bude připojeno více přepínačů pospolu, tak se vyslaná zpráva může mezi přepínači replikovat až donekonečna a kompletně přepínače zahltí.

A.4.1 Smyčka v síti

Vzhledem k tomu, že dnes jsou kladeny velké nároky na dostupnost internetových služeb, tak se rozšiřují i lokální sítě a tvoří se tak redundantní spojení, aby v případě výpadku jedné linky/přepínače, byla stále schopna zařízení (typicky PC stanice) v síti stále komunikovat. Budování těchto redundantních spojení ovšem nevyhnutelně vede k tvorbě smyček v síti. Vzhledem k tomu, že na 2 vrstvě nemají rámce hodnotu podobnou hodnotě TTL (Time to live), kterou má každý paket na vrstvě třetí, tak by v případě redundatního zapojení několika přepínačů k sobě, docházelo k tzv. broadcastovým bouřím. Během několika minut by síť byla kompletně zaplněna kopiemi a kopiemi stejných rámců až by nakonec zkolabovala.

A.4.2 Spanning Tree Protocol

Aby se zabránilo právě tvorbě takovýchto bouří, tak byl vytvořen Spanning Tree protokol, který dokáže smyčky v síti eliminovat. Funguje na základě vytvoření kostry grafu, kdy se snaží vždy pomocí STA (Spanning Tree Algoritmu) vypočítat nejkratší cenu cesty k dalšímu přepínači a pakliže vede k dalšímu přepínači více cest, tak vybere tu „nejkratší“ podle ceny cesty a ostatní zablokuje. Vytvoří se tím tedy nad fyzickou topologií (která obsahuje smyčky) novou virtuální topologii, která je již bez smyček. Protokol Spanning Tree musí nějakým způsobem komunikovat s přepínači a především stanovit jeden hlavní přepínač (Root Bridge), který bude řídit provoz a stanovovat virtuální topologie, právě z jeho úhlu pohledu.

Základní parametry

Základním parametrem, který má každý přepínač je hodnota Bridge ID (BID). Ta se skládá ze dvou částí, z nichž je jedna upravitelná (vyzkoušíte s v úloze) a druhá je pevně daná, jelikož se jedná o MAC adresu přepínače. Root Bridgem se vždy stává přepínač s nejnižší hodnotou BID. V základ je tato hodnota na všech přepínačích stejná, a tedy (pakliže se do procesu nezasáhne) rozhoduje nižší hodnota MAC adresy (v praxi se tím pádem může stát, že se Root Bridgem stane ten nejstarší a nejpomalejší přepínač v síti – nežádoucí).
Aby se přepínače (s aktivním STP) mezi sebou mohli nějakým způsobem dorozumívat, sdělovat si jak se topologie mění, kdo je Root Bridgem atd., tak existují rámce BPDU (Bridge Protocol Data Units). Struktura takového rámce je zobrazena v tab.

A.1. Rámce se ve výchozím nastavení posílají každé 2 sekundy.

Tab. A.1: Složení BPDU rámce

Velikost v bajtech	**Položka**
2	Protocol ID
1	Protocol version
1	BPDU type
1	Flags
8	Root BID
4	Root path cost
8	sender BID
2	sender port ID
2	Message Age
2	Max Age
2	Hello Time
2	Forward delay

Porty přepínače mohou celkem tři role a sice: Root port (port s nejnižší cenou cesty vedoucí k Root Bridgi), Designated port (předávající port vedoucí k dalším přepínačům), Bloking (Non-designated port, který je blokován bude zapnut v případě výpadku jiného portu). Dále každý port prochází celkem pěti za sebou jdoucími stavy, než začne fungovat a předávat komunikaci (případně zůstane jako záloha blokován). Stavy mají každý svůj časový interval:

1. Disabled – vypnuté porty nebo porty, ve kterých není nic zapojeno.
2. Bloking – přejde do něj port po zapnutí (zapojení kabelu), pouze přijímá BPDU rámce po dobu 20 sekund, jinak nic nevysílá.
3. Listening – Port nadále přijímá BPDU rámce a začíná je i vysílat po dobu 15 sekund
4. Learning – Stále posílá a přijímá BPDU rámce a do toho se začal učit MAC adresy obsažené v rámcích
5. Forwarding – port dokončuje konvergenci[1] a přechází do plného provozu

A.4.3 Rapid Spanning Tree Protocol

RSTP je novější verzí klasického STP a hlavním rozdílem (už vyplývajícím z názvu) je rychlejší doba konvergence sítě, která srazila běžných 30 sekund u STP, na dobu do 5 sekund. Další změny následovaly například u rolí portů, kdy Bloking porty

[1]Časový interval, za který dokáže port projít všechny stavy přejí do plného provozu

byly nově rozděleny na dvě kategorie, a sice: Alternate port (port, který je určen jako záložní spojení s Root Bridgem v případě výpadku root portu) a Backup port (záložní cesta pro další síťový segment v případě výpadku Designated portu).
Redukcí prošly i samotné stavy portů, kdy z původních 5-ti, zbyly jen 3.

1. Discarding – kombinuje první tři stavy z původního STP a jedná se o buď vypnuté porty a nebo porty blokované.
2. Learning – Stále posílá a přijímá BPDU rámce a do toho se začal učit MAC adresy obsažené v rámcích
3. Forwarding – port dokončuje konvergenci a přechází do plného provozu

Vylepšením prošel i BPDU rámec, který využívá plnou hodnotu „Flag“ pole (tedy plných 8 bitů z původních 2 co využíval STP) a právě i díky tomu se mohlo RSTP odprostit od techniky časovačů a nově využívat techniku smlouvání s ostatními přepínači, což je právě ten zásadní rozdíl, díky kterému je možné konvergovat síť rychle.

A.4.4 PVST a RPVST protokoly

Per VLAN Spannig Tree a Rapid Per VLAN Spanning Tree jsou proprietární protokoly společnosti Cisco, které oba vycházejí z obecných standardů (PVST z STP a RPVST z RSTP) a doplňují je o funkcionalitu „per-vlan“, což znamená, že můžete na zařízeních Cisco konfigurovat pro každou jednotlivou VLAN síť jinou instanci STP a můžete mít i jiný Root Bridge pro každou VLAN případně skupinu VLAN sítí.

Rozšíření PVST

Další věcí, kterou proprietární verze protokolu Spanning Tree přinesly, jsou rozšíření pro rychlejší dobu konvergování sítě a nebo zabezpečení na úrovni linkové vrstvy. S rozšířeními si vyzkoušíte pracovat přímo v úloze.

A.4.5 Multiple Spanning Tree protokol

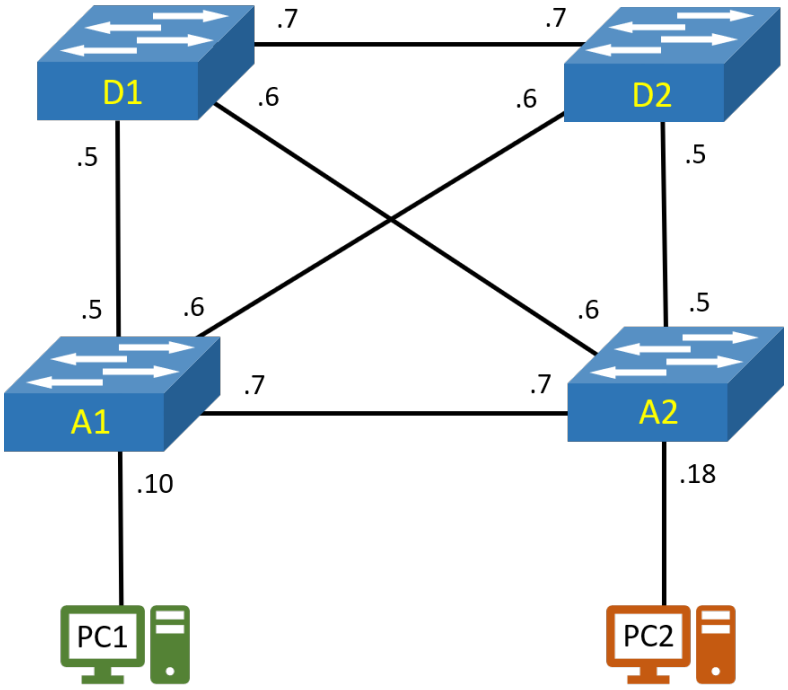
MSTP přišel jako standardizovaný protokol, který určitým způsobem právě reagoval na PVST protokoly od Cisca a posunul onu myšlenku více instancí dál. Nyní je možné v síti vytvářet MST regiony, což je ve své podstatě skupina přepínačů, která se tváří jako jeden virtuální přepínač. Aby byl přepínač přiřazen do stejného regionu, tak musí splňovat tři podmínky: mít stejný název regionu jako ostatní přepínače v regionu, stejné revizní číslo a rovněž stejné mapování VLAN sítí do instancí.
Právě MST instance jsou určité vylepšení proti PVST protokolům. PVST totiž vytváří pro každou aktivní VLAN vlastní instanci Spannig Tree, a tím pádem může

být při vyšším počtu aktivních VLAN sítí (typicky desítky až stovky), protokol pomalý, případně příliš náročný na slabší přepínače. MSTP toto řeší tak, že umožňuje přiřadit do jedné instance určitý počet VLAN sítí.

A.5 Postup řešení

A.5.1 Úkol 1

1. Začněte zapojením zařízení dle topologie na obrázku A.1 a pokračujte k základní konfiguraci přes konzoli na počítačích. Můžete použít buď přístup přes program PuTTY, případně přes program Tera Term. Přístupové údaje (především číslo portu) použijte dle tabulky A.2.



Obr. A.1: Topologie zapojení laboratorní úlohy

2. Po připojení na přepínače začněte se zadáním příkazu `enable`, dostanete se tak do „privilegovaného“ EXEC režimu, zde v tomto režimu je možné si prohlížet konfiguraci a debuggovat chování Spaning Tree protokolu – toto se bude hodit v dalších částech. Příkazem `configure terminal` (zkráceně i `conf t`) se dostanete do poslední úrovně, kde lze na zařízení začít konfigurovat fyzická zařízení, VLAN sítě atd. Začněte nastavením trunk portů na přepínači **D1**, a to posloupností příkazů:

Tab. A.2: Přístup na přepínače

Adresace	Číslo portu	Název zařízení
10.0.99.101	5001	D1 (Cisco 3750G)
10.0.99.102	5002	D2 (Cisco 3750G)
10.0.99.103	5003	A1 (Cisco 3750G)
10.0.99.104	5004	A2 (Cisco 3560X)

```
D1(config)#int range g1/0/5-7
D1(config-if-range)#switchport trunk encapsulation dot1q
D1(config-if-range)#switchport mode trunk
D1(config-if-range)#switchport nonegotiate
D1(config-if-range)#exit
```

Stejným způsobem nastavte i ostatní přepínače (tzn. D2, A1 a A2), **Pozor u přepínače A2, kde je rozdílné číslování portů!** U tohoto přepínače použijete místo příkazu `int range g1/0/5-7` příkaz `int range g0/5-7`.
Pozn.: z konfigurace portů, případě jindy, kdy se konfigurace „noří“ do dalšího separátního režimu, se vrací zpět vždy příkazem `exit`. *Toto platí i pro návrat do privilegovaného režimu.*

3. Nyní pokračujte konfigurací VLAN Trunking Protokolu v jeho třetí verzi. Nejprve nastavte roli přepínače **D2** jako primárního serveru příkazy `vtp mode server` a `vtp primary vlan`, poté nastavte doménu příkazem `vtp domain BARS` a v neposlední řadě i verzi protokolu příkazem `vtp version 3`. Příklad nastavení na přepínači D2:

   ```
   D2(config)#vtp mode server
   D2(config)#vtp domain BARS
   D2(config)#vtp version 3
   D2(config)#exit
   D2#vtp primary vlan
   ```

   Obdobně nastavte i ostatní přepínače, ale ty ponechte v roli klientů (`vtp mode client`) a pochopitelně jim ani **nenastavujte** roli primárního serveru příkazem `vtp primary vlan`.
4. Pokračujte v konfiguraci VLAN sítí přidáním VLAN sítě **10**, **20** a **30** příkazem `vlan 10`, resp. 20 a 30 v konfiguračním režimu. VLAN sítě nastavte na přepínači **D2**. Po zadání prvního příkazu se přesunete do prostředí konfigurace dané VLAN sítě, kde lze nastavit další parametry pro danou VLAN, jako je například název (příkazem `name NazevVLAN`). Můžete si sítě pojmenovat pro větší přehlednost, ale není to vyžadováno pro další úkoly.

Pozn.: pro vytvoření další VLAN není nutné opouštět režim konfigurace předchozí VLAN a je možné napsat příkaz na vytvoření další VLAN přímo v tomto režimu viz příklad níže

```
D2(config)#vlan 10
D2(config-vlan)#vlan 20
D2(config-vlan)#vlan 30
D2(config-vlan)#exit
D2(config)#
```

Ověřte, že se VLAN sítě nachází po přidání skutečně na přepínačích příkazem `sh vl br` (zkrácená podoba příkazu) v privilegovaném režimu, případně v konfiguračním režimu, kdy přidáním slova `do` před `sh`, můžete vyvolat výpis i v konfiguračním režimu.

5. Teď když jsou nastaveny VLAN sítě na všech přepínačích, pokračujte nastavením **access portů** na porty vedoucí k počítačům u přepínačů **A1** a **A2** a zařaďte je do VLAN síti 10 v případě PC1 a 20 v případě PC2. Posloupnost příkazů pro nastavení portu G0/18 na A2 do režimu access:

```
A2(config)#int g0/18
A2(config-if)#switchport mode access
A2(config-if)#switchport access vlan 20
A2(config-if)#exit
```

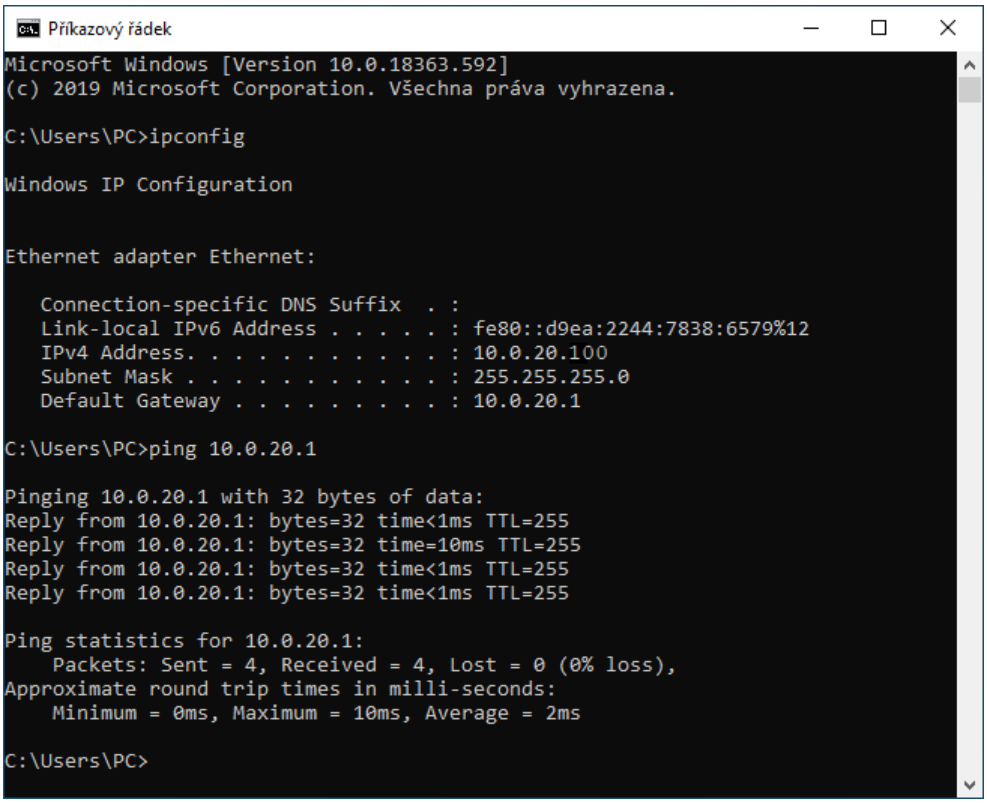
Proveďte obdobné nastavení i na přepínači **A1** a ujistěte se, že obě PC jednak dostaly adresu od DHCP serveru (jímž je přepínač D2) příkazem `ipconfig` a jednak, že komunikuje se svojí výchozí bránou (dle tabulky A.3) příkazem `ping 10.0.20.1` (pro PC2), a to v příkazové řádce ve virtuálním PC.

Tab. A.3: Adresace počítačů

Počítač	**IP adresa**	**Maska**	**Výchozí brána**	**Port na Přepínači**
PC1	10.0.10.100	/24	10.0.10.1	G1/0/10
PC2	10.0.20.100	/24	10.0.20.1	G0/18

A.5.2 Úkol 2

1. Na všech přepínačích ve výchozím nastavení funguje PVST protokol. Identifikujte příkazem `sh spanning-tree`, který přepínač se stal Root Bridgem, které porty a v jakém stavu jsou ve Spannig Tree zapojeny.



Obr. A.2: Ověření přidělená IP adresa a test dostupnosti výchozí brány na PC2

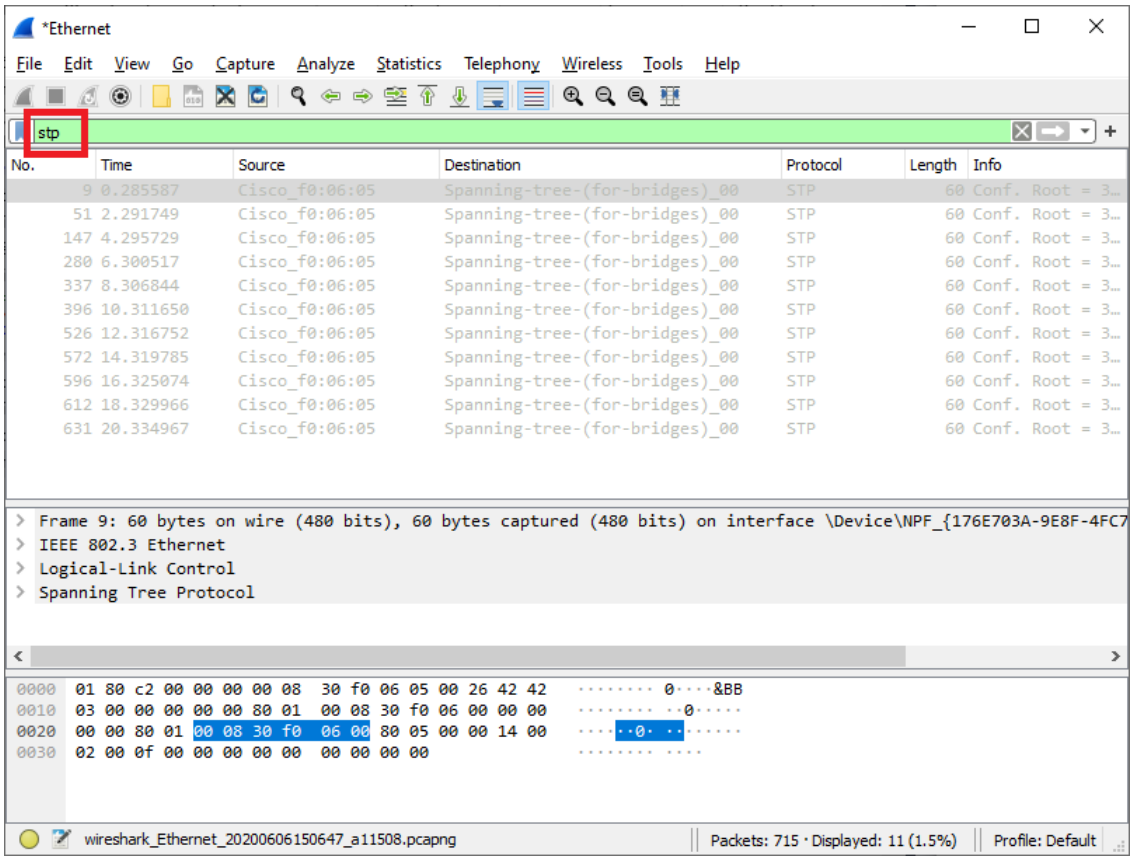
2. Upravte hodnotu Bridge Priority pomocí příkazu `spannin-tree vlan 1-30 root primary` na přepínači **D1** a nastavte přepínač **D2** jako sekundární Root Bridge příkazem `spanning-tree vlan 1-30 root secondary`.
3. Přejděte na přepínač **A1** a v privilegovaném režimu zapněte debuggování událostí o Spanning Tree příkazem `debug spanning-tree events`. Nyní na přepínači **D1** vypněte všechny tři trunk porty příkazem `shutdown` (v jejich konfiguračním režimu) a sledujte změny na přepínači **A1** ve výpisu zpráv z debuggingu. Po dokončení konvergence (přibližně 30 sekund) si ověřte příkazem `sh span`, že došlo ke změně Root Bridge (měl by se jím sát D2) a že se změnily role a stavy portů. Měli byste vidět, že port G1/0/5 zmizel (byl vypnut) a novým root portem se stal G1/0/9, který byl předtím ve stavu bloking.
4. Znovu zapněte ony trunk porty na **D1**, což by mělo mít za následek, že se po přibližně 30 sekundách všechno vrátí do původního stavu. Opět ověřte z výpisu debuggu a zadáním příkazu `sh spanning-tree` v privilegovaném režimu na **A1**.
5. Na PC1 si pusťte nepřetržitý ping na jeho výchozí bránu (příkazem `ping 10.0.10.1 -t`), a poté odpojte a znovu zapojte kabel vedoucí k přepínači. Mělo by trvat opět přibližně 30 sekund, než bude ping úspěšný.

Zprovozněte rozšíření PortFast na obou přepínačích **A1** a **A2** a nastavte jej na access porty (porty vedoucí k počítačům) těchto přepínačů. Nastavení provedete příkazem `spanning-tree portfast` v konfiguraci daného portu a po nakonfigurování zopakujte odpojení a připojení kabelu u PC1 (můžete si vyzkoušet zároveň i PC2, ale výsledek bude totožný) a sledujte rozdíl v rychlosti znovu obnovení portu do forwarding stavu (obnovení dostupnosti vých. brány).

6. Nyní si vyzkoušejte fungování rozšíření **BPDUfilter**. Na **PC2** si spusťte program Wireshark, zvolte zachytávání na Ethernet rozhraní a vyfiltrujte si v horním řádku zachytávání pouze na **stp rámce** viz Obr. A.3. Ověřte, že každé 2 sekundy přichází na port BPDU rámce. Následně nastavte BPDUfilter, což můžete učinit dvěma způsoby: buď přímo v konfiguraci daného portu příkazem `spanning-tree bpdufilter enable` anebo globálně v hlavním konfiguračním režimu příkazem `spanning-tree portfast bpdufilter default`, kdy se BPDUfilter nastaví automaticky na všechny porty, kde je již nakonfigurován PortFast. Vyzkoušejte si oba způsoby kdy nejprve nastavte BPDUfilter přímo na port a sledujte téměř okamžité zastavení přijímání BPDU rámců ve Wiresharku. Poté příkaz z portu odstraňte (zadejte příkaz znovu, ale před něj napište slovo `no`), vyzkoušejte druhý způsob a porovnejte dobu za jak dlouho přestanou na PC chodit BPDU rámce.
7. Další praktické rozšíření, které lze vyzkoušet je BPDUguard. Nastavte BPDUguard na přepínači **A2** na **kterýkoliv nevyužitý a zapnutý port** (např. G0/24) příkazem `spanning-tree bpduguard enable`. Poté do portu připojte kabel, který následně zapojíte do **kteréhokoliv volného a zapnutého portu** na jednom z přepínačů (Např. A1). Pozorujte jak BPDUguard zareaguje na přepínači **A2**. Můžete si podobný postup vyzkoušet i s globálním nastavením BPDUguardu v konfiguračním režimu přepínačů A1 a A2 příkazem `spanning-tree portfast bpduguard default`, kdy se BPDUguard nastaví na všechny PortFast porty obdobně, jako tomu bylo u BPDUfilteru.

A.5.3 Úkol 3

1. Nastavte na všech přepínačích RPVST protokol příkazem `spanning-tree mode rapid-pvst` v hlavním konfiguračním režimu, čímž docílíte změny ze staršího STP na novější RSTP. Ověřte si přepnutí z STP do RSTP příkazem `sh span summary` v privilegovaném režimu. Jelikož se v základu stále jedná o PVST protokol, tak zůstane platné i nastavení primárního i sekundárního Root Bridge.
2. Vraťte se ke krokům 3 a 4 z předchozího úkolu a zopakujte je, ovšem tentokrát s aktuálně nastaveným RPVST protokolem a porovnejte rychlost konvergence



Obr. A.3: Spuštěný program Wireshark na PC1

proti protokolu PVST. Zaměřte se také na změny u rolí jednotlivých non-designated (blokovaných) portů, když si zobrazíte standardní výpis `sh span`.

3. Na přepínači **D2** přidejte dalších 50 VLAN sítí příkazem `vlan 50-100` v hlavním konfiguračním režimu přepínače. Díky VTP by se VLAN sítě měly přidat i na ostatní přepínače – ověřte příkazem `sh vlan brief` v privilegovaném režimu na všech přepínačích.
4. Nyní zopakujte krok 2 a pokuste se porovnat dobu konvergence sítě s přidanými VLAN sítěmi.

A.5.4 Úkol 4

1. Nyní nastavte na všech přepínačích MST protokol příkazem `spanning-tree mode mst` v hlavním konfiguračním režimu. Ověřte si běh MSTP příkazem `sh spanning-tree mst` a příkazem `sh spanning-tree mst configuration`, že je přepínač ve výchozím nastavení MST regionu a všechny VLAN sítě jsou zařazeny do výchozí instance 0.
2. Vraťte se na přepínač **D2** a nastavte MST region podle následujících parametrů, jménu regionu: ARS1, revizní číslo: 1 a celkem tři instance a zařazení

VLAN sítí do těch instancí podle následující tabulky:

Tab. A.4: Zařazení VLAN sítí do MST instancí

MST instance	**VLAN sítě**
instance 1	10, 30
instance 2	20
instance 3	50 - 100

Nastavení proveďte touto posloupností příkazů:

```
A2(config)#spanning-tree mst configuration
A2(config-mst)#name ARS1
A2(config-mst)#revision 1
A2(config-mst)#instance 1 vlan 10,30
A2(config-mst)#instance 2 vlan 20
A2(config-mst)#instance 3 vlan 50-100
```

Jednotlivé parametry se neaplikují, dokud režim konfigurace MST regionu neopustíte, což zatím ale **nedělejte**! Nejprve si vyzkoušejte (přímo v tomto režimu) ověřit zadané údaje pomocí příkazu `sh pending`, kdy si zobrazíte přehledně konfiguraci MST regionu a můžete si zde překontrolovat správnost zadaných údajů. Vzhledem k tomu, že teď měníte konfiguraci regionu z výchozí, tak netřeba porovnávat konfigurace. Nicméně pakliže byste měnili konfiguraci regionu znovu, tak je dobré si příkazem `sh current` zobrazit aktuálně běžící konfiguraci, kterou následně porovnáte s tou, kterou se chystáte nastavit. Po kontrole správnosti zadaných údajů, opusťte konfigurační režim standardním příkazem `exit`.

3. Na přepínači **D2** nakonfigurujte VTP pro propagaci MST regionů napříč ostatními přepínači. Nastavte přepínač **D2** jako primární server pro MST, nejprve příkazem `vtp mode server mst`, a poté v privilegovaném režimu příkazem `vtp primary mst`, následně potvrďte, že chcete pokračovat klávesou enter.
4. Obdobně nakonfigurujte i **ostatní přepínače**, jen místo role serveru jim přiřaďte roli **klienta** a samozřejmě **ne**konfigurujte další přepínače jako primary (ono to ani nepůjde, pakliže zařízení pracují v roli klienta).
5. Teď by se mělo díky VTP synchronizovat nastavení MST regionu napříč všemi přepínači. **Ověřte**, že tomu tak skutečně je pomocí zadání příkazu `sh span mst config` v privilegovaném režimu, na všech ostatních přepínačích. Rovněž si na všech přepínačích zobrazte příkazem `sh span` informace o Spanning Tree a identifikujte Root Bridge.

6. Vzhledem k tomu, že pro MSTP se pochopitelně neaplikuje nastavení Root Bridge ještě od protokolů rodiny PVST, tak opět budete muset **přenastavit** výchozí hodnoty Bridge Priorty. Příkazy pro nastavení jsou ovšem velice podobné těm, které jste použili pro PVST, takže logika zadávání bude stejné, jen se liší syntax. Jak Root Bridge nastavte opět přepínač **D2**, a to pro všechny instance, příkazem `spanning-tree mst 0-3 root primary`. Obdobně nastavte i přepínač **D1**, jako sekundární Root Bridge (pro případ výpadku toho hlavního), příkaz pro nastavení je stejný s tím rozdílem, že místo slova `primary`, použijte slovo `secondary`. Priority si opět ověřte.
7. Proveďte stejný test rychlosti konvergence sítě, který jste prováděli u RPVST protokolu (tzn. úkol 2) a porovnejte dosažené výsledky.

A.5.5 Úklid pracoviště

- Rozpojte všechny kabely a ukliďte je zpět na místo odkud jste je vzali.
- Na všech přepínačích smažte soubor vlan.dat příkazem `delete vlan.dat` v privilegovaném režimu, a poté přepínač restartujte příkazem `reload`. Při dotázání zda-li máte uložit konfiguraci, konfiguraci **NEukládejte**!

Kontrolní otázky

1. Proč je již v dnešní době naprosto nevyhovující používat původní STP 802.1D (případně ekvivalent v podobě PVST u Cisca)?
2. Proč je nebezpečné používat rozšíření PortFast na trunk spoji mezi přepínači?
3. Kde by se podle vás měl použít RSTP (potažmo RPVST) a kde už je vhodné zvážit použití MSTP?
4. Jak problém se smyčkami na 2. vrstvě OSI modelu řeší STP již teď víte, dokázali byste ale říci, jak se smyčky řeší na 3. vrstvě?