Cisco IOS 9 - Spanning Tree Protocol

Vložil/a Samuraj [1], 21 Listopad, 2011 - 11:45

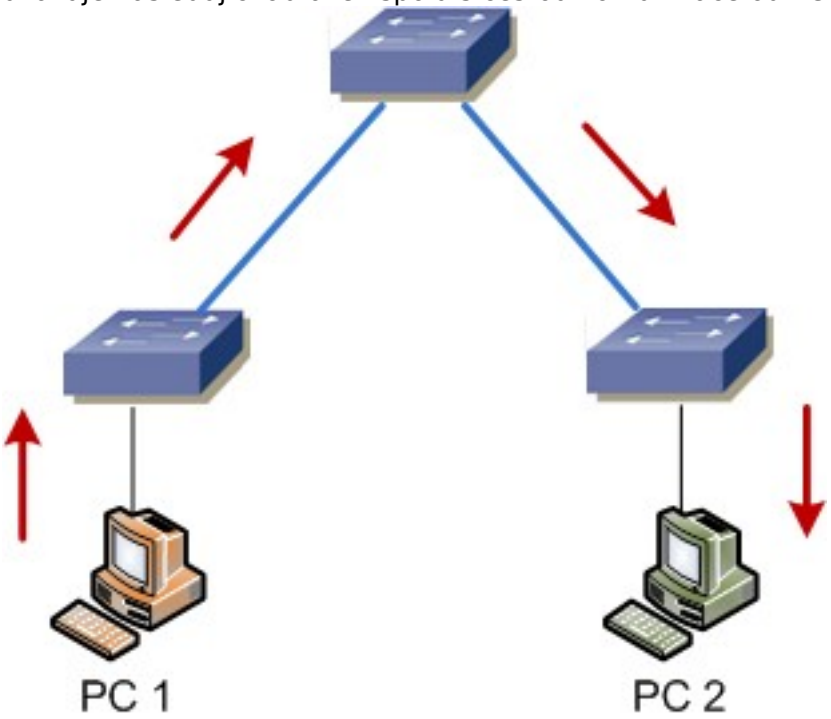
- Networks & Protocols [2]

Další, značně teoretická, část popisu Cisco IOSu se věnuje tématu smyček v síti (tedy ne-stromové topologii). Nejprve zmiňuji jaké nevýhody a naopak jaké výhody nám smyčky přináší. Následně popisuji řešení souvisejících problémů pomocí Spanning Tree Protocolu (STP). A v závěru je popsána konfigurace PVSTP pomocí příkazů IOSu.

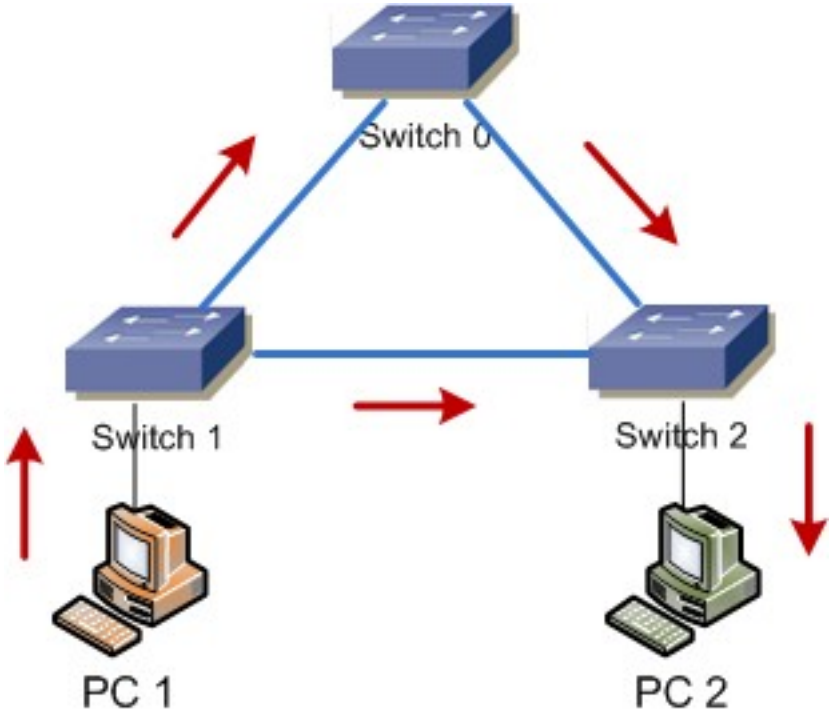
Myslím, že správce menší sítě, ani nemusí vědět, jak přesně STP funguje (i když je to vždy lepší), ale důležité je vědět, co může smyčka v síti způsobit, a že je možno využít STP. Můj popis ani nezachází do úplných podrobností, není problém najít, např. přímo u Cisca, více detailní informace. Tento popis se věnuje obecně STP (řekněme dle normy IEEE 802.1d) a pro konfiguraci jeho Cisco verzi PVSTP. Pokud to praxe dovolí, tak je lepší použít novější variantu Rapid STP či Multiple STP, kterým se budu věnovat v dalším díle.

Smyčky v síti

Pro běžnou ethernetovou síť se používá zapojení do rozšířené topologie hvězda. Jedná se o stromovou strukturu, kde mezi každými prvky existuje pouze jedna cesta. Jednoduchý příklad ukazuje následující obrázek spolu s cestou komunikace od PC 1 k PC 2.



Pokud však propojíme Switch 1 a 2, tak vznikne smyčka a mezi stanicemi bude existovat více než jedna cesta.



Problémy smyček

Smyčky mohou způsobit několik problémů

- broadcastová bouře - u broadcastů se tyto budou rozmnožovat, až dosáhnou kritického množství
- problémy s konektivitou nebo nestabilita tabulky MAC adres (CAM) - díky smyčce zpráva přijde na switch z více portů a on si stále mění adresu zdroje, v určitém případě může dojít k tomu, že si switch myslí, že je stanice připojena ke špatnému portu a nikdy ji nedoručí zprávu
- několikanásobné doručení - zpráva koluje v síti stále dokola a stále se doručuje

Nejčastější problém, pokud v běžné ethernetové LAN existuje smyčka, je, že dojde k tzv. broadcastové bouři (broadcast storm), která většinou skončí úplným zahlcením sítě. Broadcastová bouře znamená, že se v síti šíří více broadcastových (nebo i jiných) rámců, než je síť (aktivní prvky) schopna zpracovat. Pokud máme v síti smyčku, tak z principu funkce switchů dojde k tomuto efektu.

Připomeňme, jak funguje switch. Pokud dostane rámec pro neznámý cíl, tak jej přepošle na všechny porty mimo toho odkud rámec přišel. Stejně pracuje i s broadcastem. Dále si také uloží zdrojovou MAC adresu do CAM tabulky s přiřazením portu, ze kterého přišel rámec.

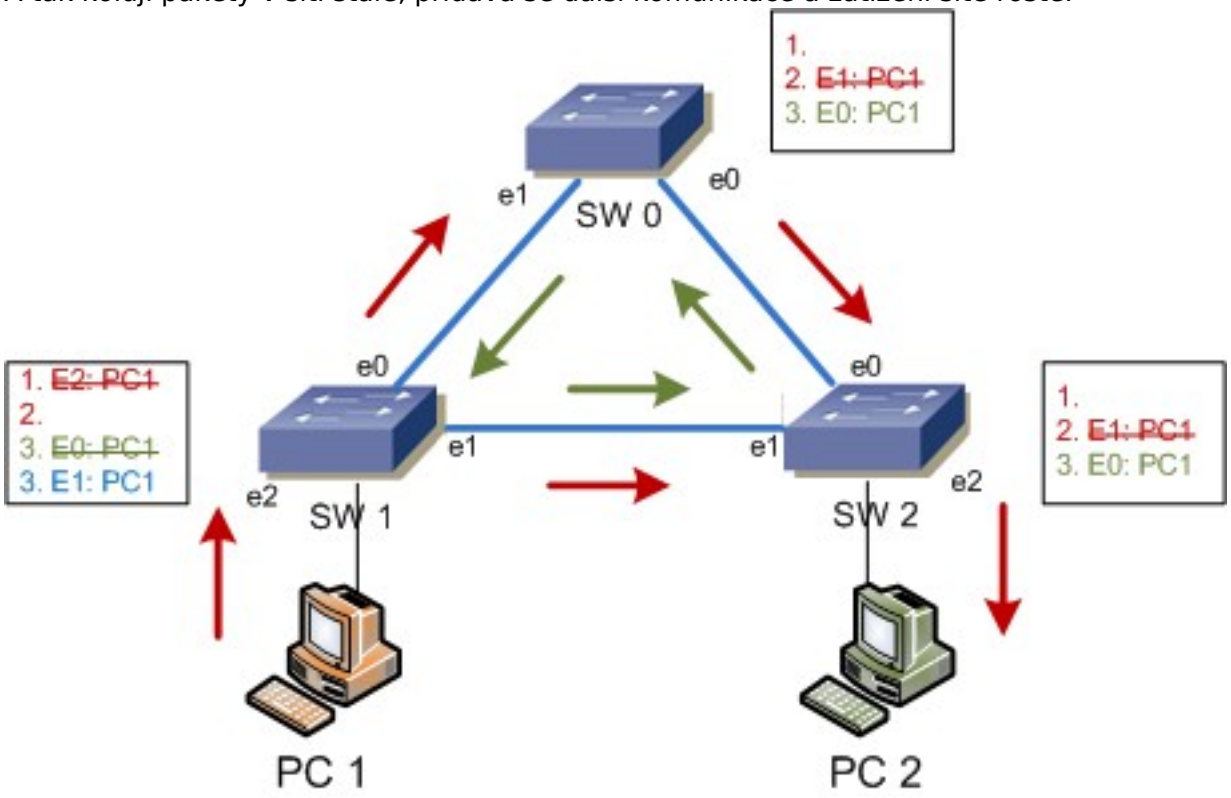
Pozn.: Na 3. vrstvě ISO/OSI (IP) máme TTL (time to live), takže kolování zprávy po určité době skončí, ale na 2. vrstvě nic takového není.

Takže na našem příkladu probíhá komunikace následovně, pokud PC1 posílá zprávu pro PC2 (dosud spolu nekomunikovali)

- krok 1 - SW1 přijme na portu e2 a odešle na ostatní (e0, e1), také si uloží záznam do CAM
- krok 2 - SW0 přijme na portu e1 a odešle na e0, také vytvoří záznam ve své CAM
- krok 2 - SW2 také přijme na portu e1 a odešle na e0, e2, cílové PC2 tedy již obdrželo zprávu, ale switche to neví
- krok 3 - SW0 přijme na portu e0 a odešle na port e1, přitom si opraví záznam v CAM, protože si myslí, že bylo PC přepojeno
- krok 3 - SW2 přijme na portu e0 a odešle na port e1, e2, přitom si opraví záznam v CAM
- krok 4 - SW1 přijme na portu e1 a odešle na e0, e1, PC1 pozná, že to není zpráva pro něj a

zahodí ji, SW1 si opraví CAM
- krok 4 - SW1 přijme na portu e0 a odešle na e1, e2, opraví CAM

A tak kolují pakety v síti stále, přidává se další komunikace a zatížení sítě roste.



Pozn.: Všude se uvádí, že Broadcastová bouře vznikne při zaslání broadcastu, ale já myslím, že zaslání unicastu může mít stejný efekt.

Proč vznikají smyčky

V dnešních lokálních sítích, které jsou často velmi rozsáhlé, může dojít ke vzniku smyčky ze dvou důvodů. Jedna možnost je chyba obsluhy či neodborná manipulace. Ve větší síti není problém omylem propojit dva switche dohromady, místo, abychom připojili nějakou stanici. Může se také stát, že někdo připojí do sítě switch místo stanice a připojí jej do dvou zásuvek.

Druhý důvod je asi důležitější a jedná se o redundanci nebo load balancing. Protože je dnes velmi důležitá vysoká dostupnost, tak se vytvářejí redundantní (nadbytečné) spojení. Potom, když dojde k výpadku některé linky nebo aktivního prvku, tak pořád vetší část sítě funguje po jiné cestě. V tomto případně slouží redundantní spojení jako záloha. Jiný případ je, kdy využíváme redundantní zapojení pro zvýšení výkonu (propustnosti) a jedná se o vyvažování zátěže. V tom případě jsou využívány všechny spoje zároveň.

Pozn.: Jednoduchým řešením rozložení zátěže může být spojování linek Cisco EtherChannel (používá normu IEEE 802.3ad či PAgP).

Spanning Tree Protocol

Proto, abychom zabránili smyčkám v síti, slouží Spanning Tree Protocol - STP. Můžeme říci, že pracuje na principu teorie grafů, síť je ohodnocený graf a algoritmus hledá kostru tohoto grafu. Jinak řečeno, hledá nejkratší cesty mezi každými dvěma switchi. Používá Spanning Tree Algorithm (STA) pro vytvoření databáze topologie a pak hledá a ruší redundantní spoje (blokuje porty - ty nevysílají a přijatá data zahazují). STP je definován normou IEEE 802.1d a je označován jako Common Spanning Tree (CST).

Pozn.: Doplněno díky Tomfimu. Originální STP (řekněme CSTP) již dnes neexistuje. V roce 2004 byla revidována norma IEEE 802.1D a byla sloučena s rozšířeními 802.1t a 802.1w, přičemž originální STP bylo nahrazeno pomocí RSTP. Přesto se v dalším popisu věnuji původnímu STP, které je nejbližší (a také nejjednodušší) k defaultní Cisco verzi PVSTP.

STP na fyzické topologii, která může obsahovat smyčky, vytvoří virtuální topologii, která již smyčky neobsahuje. Je to dynamický protokol, pokud tedy vznikne smyčka, tak se překonfiguruje, aby jí zabránil. Stejně tak, pokud se přeruší některá linka, tak se pokusí vytvořit alternativní cestu (povolením dříve blokovaného portu), pokud je to možné.

Určení nejkratší cesty

STP vytváří strom nejkratších cest (kostru grafu). Nejkratší cesta je určována na základě kumulativní ceny linek. Cena linky je dána její propustností (bandwith), podle následující tabulky. V původní specifikaci se počítalo s maximální rychlostí 1Gbs, takže byla aktualizována, aby zahrnovala i linky 10Gbps.

rychlost linky	cena od 2001	cena od 1998	cena p?vodní
10 Gbps	2000	2	1
2 Gbps	10000	3	1
1 Gbps	20000	4	1
100 Mbps	200000	19	10
10 Mbps	2000000	100	100

Bridge ID - BID

Bridge ID (BID) je základní hodnota každého switche a skládá se z priority (2B), defaultní je 0x8000, a MAC adresy switche (6B). Switch, který má nejnižší BID se stává Root Bridgem. BID můžeme změnit tím, že změníme prioritu switche.

Bridge Protocol Data Units - BPDU

STP využívá zasílání speciálních zpráv mezi zařízeními. Tyto zprávy se jmenují BPDU (bridge protocol data units) a jsou přijímány i blokovanými porty. Na začátku komunikace se používají konfigurační BPDU, následně Topology Change Notification - TCN BPDU (oznamují změnu v síťové topologii) a Topology Change Notification Acknowledgment - TCA BPDU. BPDU rámce používají jako zdrojovou MAC adresu adresu portu a odesílají se na STP multicast adresu 01:80:C2:00:00:00.

BPDU má tři hlavní části. Globální informace o STP (verze apod.), informace dané instance STP pro konfiguraci a časové parametry (STP timers). Hello Time je interval, po kterém se zasílají BPDU (default 2s). Max age (default 20s) a Forward delay (default 15s) jsou doby mezi stavy.

velikost [B]	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

Superior BPDU (nadřazené BPDU) je takové BPDU, které má nižší hodnoty Root BID, cenu cesty k

rootu, odesílací BID a odesílací port ID než ostatní.

Root Bridge

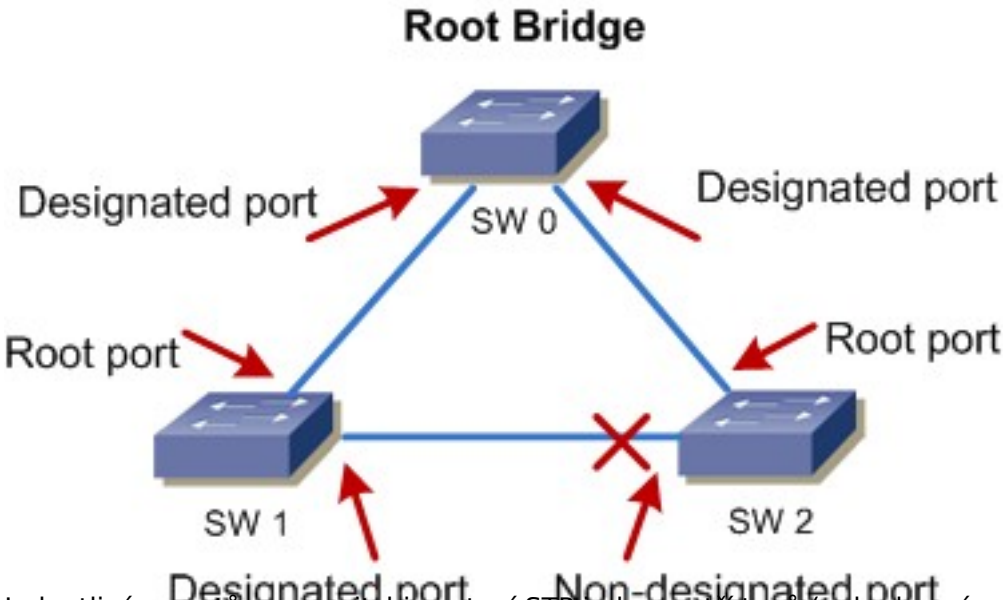
- má nejnižší BID
- všechny jeho porty jsou ve stavu forwarding (jsou komunikující) a jsou typu designated
- je to kořen stromu
- všechna rozhodnutí se dějí z jeho pohledu
- běžně je dobré zajistit, aby Root Bridge byl nejvýkonnější switch (což bývá zároveň centrální prvek)

Volba Root Bridge

Pokud nastavíme prioritu switche na nižší hodnotu, tak můžeme určit, který switch bude Root Bridgem. Volba Root Bridge probíhá následovně:

- switch (například nově připojený) odešle BPDU (jako broadcast), kde nastaví svoje BID jako root BID
- každý switch přijme BPDU a pokud je jeho BID menší než root, tak je opraví na svoje a odešle
- pokud přijme BPDU s nižším root BID, než je jeho, tak jej uzná za Root Bridge

Typy portů



Jednotlivým portům na switchi nastaví STP jeden ze tří typů (pokud není port disabled):

- **root port** - port s nejnižší cenou, buď linka přímo spojená s Root Bridgem nebo s nejkratší cestou k němu.
- **designated port** - port, který je členem STP topologie a připojuje segment.
- **non-designated port** - blokovaný port, redundantní cesta.

Root a designated port jsou porty, které posílají data, jsou ve stavu forwarding. Non-designated port je blokující, tedy ve stavu blocked.

Stavy portů

Při konvergenci (změně topologie, například připojení switche do sítě) prochází jednotlivé porty

několika stavy. Mezi každým přechodem je určitý maximální časový interval.

```
stav portu      popis   ?as [s]
Blocking (blokující)  p?ijímá pouze BPDU, nevysílá
|                               20      Max-Age
Listening (naslouchající)     posílá a p?ijímá BPDU, nic jiného
|                               15      Forward Delay 1
Learning (u?ící se)  posílá a p?ijímá BPDU a u?í se MAC adresy
|                               15      Forward Delay 2
Forwarding (p?eposílající)  posílá a p?ijímá vše
```

Konvergence STP

Přepínaná počítačová síť je konvergovaná ve chvíli, kdy všechny porty switchů jsou buď ve stavu blocking nebo forwarding. Tedy konvergence je čas, než port projde ze stavu blocking do forwarding, standardně je to max. 50s. Ke konvergenci dochází vždy při změně topologie, tedy připojení či odpojení switche/portu nebo změně konfigurace STP. Standardně tedy každý nově připojený port začne komunikovat až po 50s. Stejně tak při výpadku jedné linky dojde k překlopení na záložní linku až po této době.

Průběh STP

- volí se Root Bridge
- určují se Root Porty
- určují se Designated Porty
- ostatní se nastaví jako Non-designated

Módy/typy STP

To co jsem doposud popisoval, byl klasický STP, který se označuje jako Common Spanning Tree (CST) a je dán normou IEEE 802.1d. V průběhu času vzniklo několik dalších typů STP, které vylepšují některé vlastnosti. Na Cisco zařízeních se nemluví o typu, ale módu, ve kterém STP pracuje. Navíc Cisco používá většinu STP ve vlastní upravené verzi.

Různé typy STP

- **Common Spanning Tree (CST)** - IEEE 802.1d, pro všechny VLANy běží jediná instance STP. Norma vznikla v roce 1998 a CST byl zrušen revizí v roce 2004.
- **Per-VLAN Spanning Tree (PVST)** - Cisco, vychází z IEEE 802.1d, ale pro každou VLANu běží samostatná instance STP. Výhodou je, že mohu rozdělit zátěž, že každá VLANa komunikuje jinou cestou. Používá ISL trunk.
- **Per-VLAN Spanning Tree Plus (PVST+)** - Cisco, rozdíl oproti PVST je v tom, že používá 802.1q trunk.
- **Rapid Spanning Tree (RST)** - IEEE 802.1w, hlavní rozdíl je v rychlé konvergenci (okolo 1s). Revizí v roce 2004 byl sloučen do normy 802.1d.
- **Rapid per-VLAN Spanning Tree Plus (RPVST+)** - Cisco, vychází z IEEE 802.1w, RST běží pro každou VLAN zvlášť.
- **Multiple Spanning Tree (MST)** - IEEE 802.1s, rychlé jako RST a umožňuje mapovat několik VLAN do jedné STP instance, tedy umožní ušetřit počet STP pro velký počet VLAN. MSTP běží navrchu nad RSTP, takže vždy musí existovat oboje. Používá se na páteř sítě. Revizí v roce 2003 byl sloučen do normy 802.1q, která se věnuje VLANám.

STP Load Balancing – vyvažování zátěže mezi trunk porty

Pomocí priority portu – port priority

Spanning-Tree protokol můžeme použít pro určité vyvažování zátěže VLAN na trunk portech. Vychází se z toho, že switche máme přímo propojeny více než jedním trunkem (což je běžné kvůli redundanci). Tehdy je jedna linka blokována a komunikuje se pouze přes jednu. Protože pro obě linky je stejné Root Bridge ID, cena cesty i BID odesílajícího switche, tak se blokovaný port volí pouze podle odesílajícího port ID.

Port ID (16ti bitová hodnota) se skládá z priority portu a jeho indexu. Priorita portu je defaultně 128, ale můžeme ji změnit konfigurací a to i jen pro některou VLANu. Validní hodnoty jsou násobky 16-ti do hodnoty 240, ostatní hodnoty jsou odmítnuty. Vyšší prioritu má Port ID s nižší hodnotou (tedy i nižší port priority).

```
SWITCH(config-if)#spanning-tree port-priority 48         // priorita celého interfacu
SWITCH(config-if)#spanning-tree vlan 3 port-priority 48  // priorita portu pro danou
VLAN
```

Pomocí ceny cesty – path cost

Druhá možnost vyvažování zátěže pomocí STP je využití ceny cesty (path cost). Při této metodě mohou být různé trunk linky zapojeny do různých switchů. Cena cesty se standardně určuje podle rychlosti linky. Cenu můžeme také zadat ručně, nižší hodnota má větší prioritu (pokud jsou hodnoty stejné, tak se určuje podle BID a port ID).

```
SWITCH(config-if)#spanning-tree cost 4          // cena celého interfacu
SWITCH(config-if)#spanning-tree vlan 10 cost 4  // cena pro VLAN na interfacu
```

Konfigurace STP v Cisco IOS

Pozn.: Myslím, že v řadě případů v praxi, nám stačí konfigurovat pouze dvě věci. Určit Root Bridge a na každý koncový port (kde je připojen počítač) nastavit PortFast.

Dnešní Cisco switche podporují STP v módu PVST+, Rapid PVST+ a MSTP. Pro PVST+ a RPVST+ může existovat (většinou) max. 128 STP instancí. Pro MSTP bývá limit 65 instancí. Jedná se tedy pouze o verze STP, které obsahují Cisco rozšíření.

```
Pozn.: STP je na Cisco switchích standardn? zapnuté (v módu PVST+) a nedoporu?uje se
jej vypínat.
```

Základní konfigurace parametrů STP nezáleží na použitém módu, pro vyšší režimy pouze přibývají další vlastnosti. V této kapitole budu popisovat pouze konfiguraci STP v módu PVST+. V tomto případě většinou nepotřebujeme konfigurovat téměř nic.

Pozn.: Možné konfigurace a vlastnosti se liší podle verze IOSu.

Na začátku konfigurace můžeme zvolit, v jakém módu má STP pracovat.

```
SWITCH(config)#spanning-tree mode pvst    // nastavení módu STP
```

Pozn.: Pokud změníme mód STP, tak jsou všechny instance znovu inicializovány a může dojít k přerušení komunikace.

Dále můžeme zapnout či vypnout STP pro jednotlivou VLANu.

```
SWITCH(config)#spanning-tree vlan 10      // zapne STP pro VLAN 10
SWITCH(config)#no spanning-tree vlan 10   // vypne STP pro VLAN 10
```

Potom můžeme konfigurovat jednotlivé parametry STP pro každou VLANu zvlášť. Myslím, že nejdůležitější je nastavení priority, protože tím určujeme Root Bridge. Příkaz s root primary zjistí aktuální nejnižší prioritu v STP instanci a nastaví nižší, takže je ještě lepší než nastavování přímo priority.

Pozn.: STP instance pro VLANu vzniká automaticky, když je první port zařazen do VLANy a ruší se, když se poslední port vyřadí.

```
SWITCH(config)#spanning-tree vlan 10 priority 32768 // nastaví prioritu switche,
násobky 4096
SWITCH(config)#spanning-tree vlan 10 root primary   // nastaví switch jako root
```

Místo jedné VLANy můžeme definovat i několik oddělených čárkou nebo rozsah pomocí pomlčky.

Také můžeme použít volitelné klíčové slovo diametr a definovat maximální průměr sítě. V praxi většinou jako Root Bridge volíme centrální prvek (core switch) a poloměr sítě bývá 2 (k centru jsou připojeny rovnou access switch) nebo 3 (máme ještě distribuční vrstvu). Switch potom dopočítá optimální hodnoty pro hello time, forward-delay time, a maximum-age time.

```
SWITCH(config)#spanning-tree vlan 1-4094 root primary diameter 2
```

Pro kontrolu a dohled na STP slouží řada show příkazů.

```
SWITCH#show spanning-tree                // zobrazí info o STP pro každou VLANu
SWITCH#show spanning-tree summary        // stru?né info o STP
SWITCH#show spanning-tree detail         // detailní info o STP
SWITCH#show spanning-tree vlan 100       // info o STP pouze pro danou VLANu
SWITCH#show spanning-tree interface f0/1 // info o STP pouze pro daný interface
SWITCH#show spanning-tree bridge detail  // stru?né p?ehled STP instancí
```

Jak zjistím, který switch je Root Bridge?

Pomocí show spanning-tree summary vidím, pro které VLANy je daný switch Rootem. Také to poznám podle toho, že pro danou VLANu jsou všechny porty ve stavu Designated.

Abych dohledal správný switch, tak se na jakémkoliv switchi podívám na show spanning-tree vlan 100, který port je Root a přejdu na switch, který je do něj připojen. Postupně se dostanu až na Root Bridge.

Rozšíření STP

Cisco má řadu rozšíření pro běžné STP. Většinou se jedná o zvýšení rychlosti nebo zlepšení bezpečnosti. Pouze stručně se zmíním o několika z nich. Nejdůležitější je, dle mého názoru, portfast.

PortFast

Normálně se po připojení zařízení k portu musí projít celý cyklus od blokovaného stavu k forwarding. Pokud víme, že na portu je připojen pouze počítač a nemůže dojít ke smyčce, tak můžeme nastavit port jako portfast, kdy po zapnutí přejde rovnou do stavu forwarding. Nastavit můžeme buďto na port nebo globálně pro všechny porty (kde není určeno jinak).

```
SWITCH(config-if)#spanning-tree portfast       // pro jeden port
SWITCH(config)#spanning-tree portfast default  // pro všechny
```

Pozn.: Pokud není nastaven portfast, tak se často stane, že připojené PC (třeba s Windows XP) nabootuje dříve, než port přejde do forwarding stavu, takže při odeslání žádosti DHCP o adresu nedostaneme odpověď a nastává řada problémů. Portfast můžeme nastavit i na trunk port, pokud je zde připojen server.

UplinkFast

Používá se převážně na přístupových switchích (access switch). Při výpadku hlavní linky (Root Port), odblokuje záložní linku a zajistí její okamžité přepnutí do forwarding stavu (vynechává stavy listening a learning). Nastavuje se pro celý switch.

```
SWITCH(config)#spanning-tree uplinkfast
```

BPDUguard a BPDUfilter

Obě funkce můžeme nastavit buďto per port nebo globálně, jako defaultní chování portu, tehdy se však týká pouze portů, které mají nastavený portfast.

BPDU guard ochraňuje port, který je určený pro koncovou stanici (nebo server). Pokud přes tento port přijde BPDU, tak se port vypne (přepne se do stavu error-disable). Většinou to znamená, že někdo připojil nepovolený switch.

```
SWITCH(config-if)#spanning-tree bpduguard enable         // pro jeden interface
SWITCH(config)#spanning-tree portfast bpduguard default  // pro všechny
```

BPDU filter slouží k filtrování STP provozu na portech určených pro koncovou stanici (nebo server). Zabrání přijímání a odesílání BPDU paketů, což je dobré nastavit, aby klientské stanice nedostávaly tuto komunikaci. Pokud na port dorazí BPDU, tak se vypne portfast (pokud bylo zapnuto) a také BPDU filter.

```
SWITCH(config-if)#spanning-tree bpdufilter enable         // pro jeden interface
SWITCH(config)#spanning-tree portfast bpdufilter default  // pro všechny
```

STP Guard

Můžeme použít Root Guard, který chrání síť, aby se nestal nechtěný switch Root Bridgem. Pokud by například někdo připojil switch s prioritou 0 a nízkou MAC adresou. Vynucuje, aby port, na který je nastavený Root Guard byl designated portem, pokud by se měl stát root portem, tak se zablokuje (přepne se do root-inconsistent stavu, kdy neposílá data, ale přijímá BPDU).

```
SWITCH(config-if)#spanning-tree guard root
```

Dodatečnou obranu před vznikem smyček nabízí Loop Guard.

```
SWITCH(config-if)#spanning-tree guard loop     // pro jeden interface
SWITCH(config)#spanning-tree loopguard default // pro všechny
```

URL článku: https://security-portal.cz/clanky/cisco-ios-9-spanning-tree-protocol

Odkazy:
[1] https://security-portal.cz/users/samuraj
[2] https://security-portal.cz/category/tagy/networks-protocols