**uVysoká škola báňská – Technická univerzita Ostrava**
**Fakulta elektrotechniky a informatiky**

# Projekt do SPS

***Otestování speciálních vlastností přepínačů Cisco Catalyst:***

- *port security*
- *Unidirectional Link Detection (UDLD)*
- *protected port*
- *Flexlink*
- *traffic storm control*
- *private VLAN*

Tomáš Jílek – JIL022
Lukáš Kašpar – KAS167
Ladislav Rozsíval – ROZ089

**Cíl projektu:**

Cílem našeho projektu je prozkoumat speciální vlastnosti přepínačů Cisco Catalyst. Vlastnosti otestovat na dostupných přepínačích a zjištěné skutečnosti sepsat. K dispozici jsme měli přepínače typu Cisco Catalyst 2950 a 3560. Tento dokument obsahuje výsledky naší práce a to tak, že každá kapitola se věnuje jedné z šesti testovaných vlastností. Na začátku kapitoly je vždy daná vlastnost přepínače popsána a naznačen její význam. Dále následuje popis výchozího nastavení a následně jednotlivé příkazy, jak danou vlastnost nakonfigurovat. Poté prezentujeme námi navržený příklad, na kterém jsme danou vlastnost testovali, popis konfigurace a na závěr každé kapitoly naše zhodnocení.

První z testovaných vlastností byl Security port, který by podle specifikace výrobce měl sloužit k řízení přístupů k jednotlivým portům přepínače. Tj. na základě zadaných MAC adres povolit přístup k portu a následně tedy třeba k celé síti.

Druhou speciální vlastností přepínačů, kterou jsme testovali, byl UDLD tj. Unidirectional Link Detection. Tato vlastnost by měla zabránit problémům s jednosměrným spojením, které může být způsobeno například poškozením optického kabelu nebo jednoho páru vodičů UTP kabelu. Jednosměrné spojení může být nebezpečné kvůli možnému tvoření smyček ve spanning tree a proto je důležité mít nástroj pro obranu před tímto problémem.

Dalšími dvěmi testovanými vlastnostmi jsou Protected port a Flex links, které jsou poměrně známé a často využívány poskytovateli připojení k internetu. Protected port umožňuje blokovat provoz mezi dvěmi a více porty přepínače, čehož se hlavně využívá, když poskytovatel připojení chce zabránit komunikaci mezi jednotlivými porty (a tedy uživateli) a chce jim povolit pouze komunikovat s nějakým hlavním portem (většinou nějakým serverem nebo páteřním routerem). Flex links poskytují možnost zálohovat nějakou důležitou trasu (např. připojení do internetu) tj. zajistit provoz s co nejméně a s co nejkratšími výpadky. To by měla zajišťovat schopnost Flex links: když vypadne hlavní linka, plynule ji nahradí záložní linka a naopak.

Pátou testovanou vlastností byla Traffic storm control tj. vlastnost umožňující přepínači při jeho velkém zatížení nějakým způsobem regulovat provoz přes něj procházející a tím zabránit úplnému zhroucení všech spojení. Je jasné, že je lepší, když přepínač v době vysokého provozu omezí nějaké pakety nastavené jako „méně“ důležité, než kdyby se snažil přenést vše a při zahlcení nepřenesl vůbec nic.

Poslední námi testovanou vlastností byly Private VLANs, což podle specifikace výrobce by měl být nástupce Protected portů, jelikož umožňuje detailnější možnosti nastavení komunikace mezi jednotlivými porty. Tj. podobně jako Protected port brání komunikaci mezi různými druhy portů, ale navíc poskytuje možnost nastavit různé skupiny portů, které spolu mohou komunikovat (community porty), hlavní port, který má možnost komunikovat se všemi porty (promiscuous port) a porty (isolated porty), které mohou komunikovat jen s hlavními porty.

## Security port

Prvním pojmem, kterým jsme se v projektu zabývali, byl security port. Security portů se užívá k blokování vstupu na jednotlivých portech přepínače. K blokování portu přepínače dojde, pokud se o přístup na daný port pokusí jiná než povolená MAC adresa. Na každém portu přepínače lze nastavit jednu výchozí MAC adresu. Kromě této adresy lze portům přidělit ještě další povolené adresy. Počet povolitelných adres na port závisí na typu přepínače. Například přepínač Cicso Catalyst 2950 umožňuje jednotlivým portům přidělit 1-132 povolených adres. U novějších typů má přepínač paměť pro uložení dalších MAC adres, které lze rozdělit mezi porty podle potřeby (počet adres opět závisí na typu přepínače). Tyto adresy mohou být přiděleny například jednomu portu, anebo rozloženy po celém přepínači.

### Postup při nastavování adres portům

Nejprve si alokujeme pro daný port požadovaný počet MAC adres, následně přiřadíme portu jednotlivé MAC adresy. To můžeme provést buď manuálně nebo nechat port ať se naučí MAC adresy dynamicky z připojených zařízení. Případně některé nastavit manuálně a zbytek nechat naučit automaticky. Pokud je port odpojen, tak se dynamicky naučené MAC adresy vymažou. Tato vlastnost je v novější verzi security portu rozšířena, tak aby si přepínač naučené MAC adresy pamatoval (je popsána níže). Port na přepínači může být nastaven ve třech režimech: shutdown, protect, restrict mode (viz. Tabulka 1).

Novější verze security portu dává uživateli možnost přepnout port do režimu Sticky. V tomto režimu je oproti klasickému security portu výhodou, že dynamicky naučené bezpečné MAC adresy se i po odpojení linky udrží v paměti a přepínač se je nemusí znovu učit. Tyto dynamicky naučené adresy se uloží do running configuration souboru a pří startování přepínače jsou znovu načteny.

Aging – je to volitelná vlastnost security portu, která nastavuje dobu platnosti povolených MAC adres. Tyto adresy jsou po uplynutí nastavené doby vymazány automaticky bez nutnosti je mazat ručně.

## Výchozí nastavení portů

| **Vlastnosti portu** | **Výchozí nastavení** |
|---|---|
| Port security | Zakázáno |
| Maximum number of secure MAC addresses | 1 |
| Violation mode | Vypnuto |
| Sticky address learning | Zakázáno |
| Port security aging | Zakázáno (0) |

**Tabulka 1**

V následující Tabulce 2 je uveden postup a možnosti nastavení security portu. V levém sloupci je uveden příkaz a v pravém sloupci je uveden význam příkazu. Pokud má příkaz více možností nastavení, tak je zde uveden význam jednotlivých možností.

### Nastavení security portu

| | Příkaz | Význam |
|---|---|---|
| Step 1 | **configure terminal** | Vstup do konfiguračního režimu |
| Step 2 | **interface** *interface-id* | Konfigurace rozhraní |
| Step 3 | **switchport mode access** | Nastavení módu jako access |
| Step 4 | **switchport port-security** | Nastavení portu do módu Security port |
| Step 5 | **switchport port-security maximum** *value* | Nastavení počtu MAC adres, kterým povolíme současně přístup na port. |
| Step 6 | **switchport port-security violation** {**protect** \| **restrict** \| **shutdown**} | Pokud na port přistupuje jiná než povolená MAC adresa, pak lze nastavit jedna z těchto možností:<br><br>**protect**<br>- všechny rámce s neznámou adresou jsou zahazovány<br><br>**restrict**<br>- stejný jako protect, avšak navíc je oznámeno, že o přístup se pokusila nepovolená MAC adresa (zpráva zapsána do syslogu)<br><br>**shutdown**<br>- v tomto módu port security způsobuje, že pokud se o přístup pokusí nepovolená MAC adresa, tak po zjištění se port přepne do chybového stavu a vypne se LED dioda portu. Událost je oznámena správci sítě SNMP zprávou a je zapsána do syslogu |
| Step 7 | **switchport port-security mac-address** *mac-address* | Tento příkaz slouží pro statické nastavení povolené MAC adresy na daném portu. Příkaz opakujte pro každou požadovanou MAC adresu. Pokud zadáte méně než maximum MAC adres, tak se zbytek do maxima naučí dynamicky. |
| Step 8 | **switchport port-security mac-address sticky** | Zapnutí Sticky módu na portu |

| | | |
|---|---|---|
| Step 9 | **switchport port-security aging** {**static \| time** *time* \| **type** {**absolute \| inactivity**}} | Povolí nebo zakáže časovou platnost MAC adres.<br><br>Parametr **static** značí, že vlastnost platí pouze pro staticky nastavené MAC adresy.<br><br>Parametr **time** je doba platnosti MAC adresy. Je možno ji nastavit v rozmezí 0-1440 minut. Pokud je nastavena na nulu, tak to znamená, že tato vlastnost je zakázána pro tento port.<br><br>Parametr **type**, může nabývat dvou hodnot:<br>- **absolute** – MAC adresa je z tabulky portu vymazána po vypršení nastavené doby<br>- **inactivity** – MAC adresa je z tabulky portu vymazána, pokud není z této adresy po nastavenou dobu žádný datový provoz |
| Step 10 | **end** | Návrat |
| Step 11 | **show port-security** | Ověření konfigurace |

**Tabulka 2**

## Příklad

Příklad jak nastavit max. 50 adres, adresy budou naučeny dynamicky a bude zapnut sticky mód, aby si adresy přepínač pamatoval i po restartu.

```
Switch# configure terminal
Switch(config)# interface fastethernet0/1
Switch(config-if)# switchport mode access
Switch(config-if)# switchport port-security
Switch(config-if)# switchport port-security maximum 50
Switch(config-if)# switchport port-security mac-address sticky
Switch(config-if)# end
```

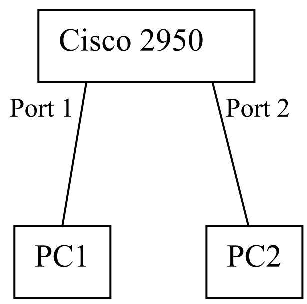
Příklad jak nastavit jednu statickou MAC adresu a zbytek se naučí dynamicky a bude zapnut sticky mód, aby si adresy přepínač pamatoval i po restartu.

```
Switch# configure terminal
Switch(config)# interface fastethernet0/2
Switch(config-if)# switchport mode access
Switch(config-if)# switchport port-security
Switch(config-if)# switchport port-security maximum 50
Switch(config-if)# switchport port-security mac-address 00:E0:4C:73:14
Switch(config-if)# switchport port-security mac-address sticky
Switch(config-if)# end
```

## Zhodnocení

Port security byl odzkoušen na přepínači Cisco Catalyst 2950 v módu shutdown v zapojení podle obrázku 1. Test byl proveden tak, že jsme připojili 2 PC k přepínači na port 1 a 2. Port 1 jsme přepnuli do security módu s povolenou MAC adresou prvního PC. Dostupnost PC byla prověřena echo zprávami. Po přepojení prvního PC do portu 2 a druhého PC do portu 1, echo zprávy už neměly odezvu, protože MAC adresa druhého PC nebyla povolena pro port 1. Přičemž port 1 byl podle očekávání odpojen a jeho LED dioda zhasla. Port opět „oživíme“ zadáním příkazů **shutdown** a **no shutdown**.

```
Switch# configure terminal
Switch(config)# interface fastethernet0/1
Switch(config-if)# switchport port-security
Switch(config-if)# switchport port-security mac-address 00:E0:4C:73:14
Switch(config-if)# end
```




Obr. 1

## UDLD - Unidirectional Link Detection

UDLD protokol pracuje na druhé vrstvě a spolupracuje s mechanismy první vrstvy, které slouží k fyzickému zjištění stavu linky. Jedná se o detekci jednosměrných spojení a následné odpojení portu. Jednosměrné spojení vznikne porušením optického kabelu nebo UTP kabelu tak, že jedna strana přijímá a nemůže vysílat, tedy pokud jedna strana zašle nějakou zprávu, tak ji druhá strana nepřijme. Toto je nebezpečné, protože můžou vznikat smyčky ve spanning tree topologii. Pokud je tedy provoz pouze jednosměrný, tak je port odpojen úplně.

Aby bylo jednosměrné spojení správně detekováno, je nutné, aby všechna připojená zařízení UDLD podporovala.

Možnost konfigurace UDLD je buď globální (viz. Tabulka 4) pro celý přepínač, anebo pro jednotlivé porty (viz. Tabulka 5). Konfigurace pro jednotlivé porty má vyšší prioritu než konfigurace globální.

**Normální mód** – v tomto režimu UDLD detekuje jednosměrné spojení, pokud jsou optická vlákna v optickém portu odpojena. Pokud jsou ovšem porty zapojeny správně, ale spojení je přesto jednosměrné, tak to UDLD nerozpozná, vyhodnotí port, že se nachází v neurčitém stavu a neodpojí ho.

**Agresivní mód** – pokud se port nachází v tomto režimu, tak UDLD detekuje jednosměrné spojení jako v předchozím případě. Navíc však detekuje jednosměrné spojení na lince mezi dvěma zařízeními, kde nejsou povoleny žádné chyby. Spojení tedy vyhodnotí jako chybové, pokud nastane jeden z následujících případů:

- na optické nebo na UTP lince jeden z portů nevysílá nebo nepřijímá provoz
- na optické nebo na UTP lince je jeden z portů odpojen zatím co druhý je aktivní

- jedno z vláken v optickém kabelu je odpojeno

V těchto případech potom UDLD odpojí zasažený port.

**Postup zjišťování jednosměrných spojení**

UDLD používá pro detekci dva mechanismy:

1. Udržování databáze sousedů – UDLD periodicky zasílá hello pakety na každé aktivní rozhraní, aby se dozvědělo o dalších sousedech, kteří provozují UDLD protokol. Když přepínač přijme hello zprávu, uloží si informaci do své cache paměti dokud nevyprší čas platnosti zprávy (message time – viz. Tabulka 4). Pokud přepínač přijme novou hello zprávu předtím, než starší uložený záznam vyprší, přepínač nahradí starší záznam novým. UDLD vymaže všechny existující záznamy pro rozhraní ovlivněná změnou konfigurace, pokud nastane jeden z následujících případů:
   - rozhraní je zakázáno, ale UDLD běží
   - UDLD je na rozhraní zakázáno
   - pokud je přepínač restartován

   Potom UDLD posílá nejméně jednu zprávu, aby informoval sousedy o vzniklé změně a ti si vymazali část jejich paměti, kterou změna ovlivnila. Zpráva je určená, aby udržela paměti synchronizované.

2. Událostmi řízená detekce a ozývání – UDLD založené na ozývání jako detekčním mechanismu. Kdykoliv se UDLD zařízení dozví o novém sousedu nebo přijme požadavek na opětovnou synchronizaci od nesynchronizovaného souseda, restartuje detekční okno na své straně spojení a pošle echo zprávu jako odpověď. Protože toto chování je stejné na všech UDLD sousedech, odesílatel echo zprávy očekává přijmutí echo zprávy jako odpověď. Jestliže detekční okno skončí a není přijata žádná platná odpověď, linka může být odpojena v závislosti na módu v jakém se UDLD nachází. Pokud je UDLD v normálním módu, tak je linka shledána jako nerozpoznaná a rozhraní nebude odpojeno. Pokud je UDLD v agresivním módu, tak bude linka shledána jako jednosměrná a rozhraní bude odpojeno.

## Výchozí nastavení UDLD

| **Vlastnost** | **Výchozí nastavení** | **Význam** |
|---|---|---|
| UDLD global enable state | Zakázáno | Globální stav UDLD |
| UDLD per-interface enable state for fiber-optic media | Zakázáno pro optická rozhraní | Stav nastavení UDLD pro optická rozhraní |
| UDLD per-interface enable state for twisted-pair (copper) media | Zakázáno pro Ethernet 10/100 a 1000BASE-TX rozhraní | Stav nastavení UDLD pro Ethernet rozhraní |
| UDLD aggressive mode | Zakázáno | Stav UDLD v agresivním módu |

**Tabulka 3**

### Globální nastavení UDLD pro celý přepínač

| | **Příkaz** | **Význam** |
|---|---|---|
| Step 1 | **configure terminal** | Vstup do konfiguračního režimu |
| Step 2 | **udld** {**aggressive** \| **enable** \| **message time** *message-timer-interval*} | Specifikace módu UDLD:<br><br>• **aggressive -** povolení UDLD v agresivním režimu pro všechna optická rozhraní<br><br>• **enable -** povolení UDLD v normálním režimu pro všechna optická rozhraní. UDLD je ve výchozím nastavení zakázán.<br><br>• **message time** *message-timer-interval* - Čas mezi posílanými hello pakety mezi sousedy ke zjištění jejich aktuálního stavu. Je možno ho nastavit od 7 do 90 sekund.<br><br>Pozn. Toto nastavení platí pouze pro optická rozhraní. Pro další rozhraní je nutno použít konfiguraci přímo na rozhraní. |
| Step 3 | **end** | Návrat |
| Step 4 | **show udld** | Ověření konfigurace |

**Tabulka 4**

### Povolení UDLD na rozhraní

| | **Příkaz** | **Význam** |
|---|---|---|
| Step 1 | **configure terminal** | Vstup do konfiguračního režimu |
| Step 2 | **interface** *interface-id* | Konfigurace rozhraní |
| Step 3 | **udld port** [**aggressive**] | Ve výchozím nastavení je UDLD vypnuté.<br>- **udld port -** aktivuje UDLD v normálním módu na daném rozhraní<br>- **udld port aggressive -** aktivuje na daném portu UDLD v agresivním módu |
| Step 4 | **end** | Návrat |
| Step 5 | **show udld** *interface-id* | Ověření konfigurace |

**Tabulka 5**

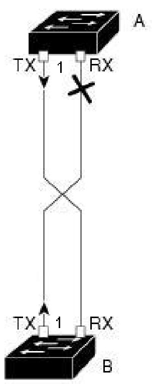
### *Reset konfigurace nastavení UDLD*

- znovu povolení všech rozhraní, které byly vypnuty UDLD

| | **Příkaz** | **Význam** |
|---|---|---|
| Step 1 | **udld reset** | Reset pro všechna rozhraní |
| Step 2 | **show udld** | Ověření konfigurace |

**Tabulka 6**

### Příklad funkce




Obr. 2

Když je UDLD v agresivním režimu, při detekci problému se zavře port. Pokud se nachází UDLD v normálním režimu, tak UDLD nerozpozná chybu a interface nebude zavřen.

Na obrázku 2 je propojen port 1 přepínače A s portem číslo 1 v přepínači B. Port v přepínači A je nastaven jako UDLD v agresivním módu. Pokud je zjištěn jen jednosměrný provoz na lince, tak je port 1 přepínače A uzavřen.

```
SwitchA# configure terminal
SwitchA(config)# interface fastethernet0/1
SwitchA(config-if)# udld port aggressive
SwitchA(config-if)# end
```

```
SwitchB# configure terminal
SwitchB(config)# interface fastethernet0/2
SwitchB(config-if)# udld port aggressive
SwitchB(config-if)# end
```

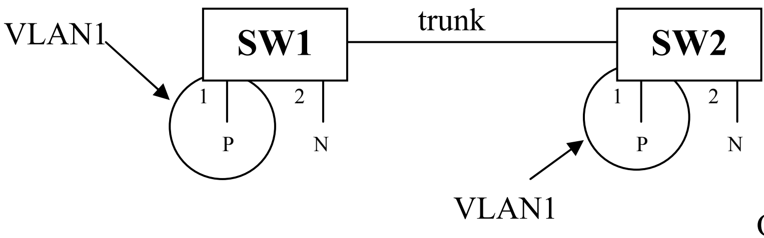
### Zhodnocení

Při testování této funkce jsme narazili na problém, který znemožnil řádně otestovat tuto vlastnost přepínače. I když na všech portech přepínače i na něm jako celku bylo UDLD vypnuto, při přerušení jednoho směru linky byl téměř okamžitě port odpojen, což znemožnilo testování. Konfiguraci jsme přesto vyzkoušeli (podle obr. 2) a bylo potvrzeno, že jak přepínač A tak i B znal kam je jeho port s nastaveným UDLD připojen. Konkrétně tedy rozhraní fastethernet0/1 přepínače A vědělo, že je připojeno na fastethernet0/2 přepínače B a naopak. Z toho vyplývá, že předávání informaci mezi dvěmi zařízeními podporující UDLD funguje (testováno na přepínači 2950 i 3650).

## Protected port

V překladu chráněný port, se užívá, pokud chceme zabránit v komunikaci některým zařízením připojeným do jednoho přepínače. Dva protected porty mezi sebou nemohou navzájem komunikovat. Komunikace je povolena jen mezi protected portem s neprotected portem. Kdybychom chtěli komunikovat dvěma protected porty mezi sebou, tak musí jít komunikace přes L3 zařízení. Jedná se tedy o omezení komunikace na druhé vrstvě (obrázek. 4).
Pokud chceme zamezit komunikaci mezi dvěma chráněnými porty, ale na různých přepínačích, tak musíme na každém přepínači vytvořit VLAN (obrázek 3). Do této VLAN přidáme porty,

jimž chceme zamezit v komunikaci a poté mezi přepínači vytvoříme trunk linku. Tím, že budou porty v jedné VLAN docílíme stavu, jako by byly dva protected porty na jednom přepínači a tak jim zamezíme vzájemnou komunikaci.



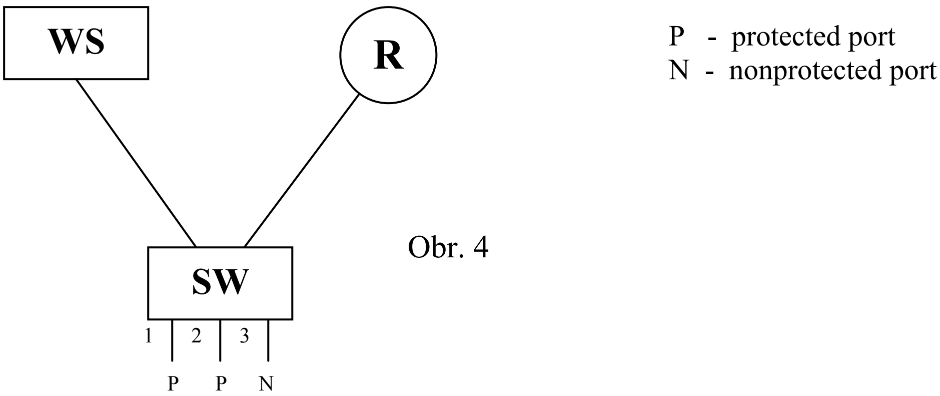

Obr. 3

### Nastavení protected na rozhraní

| | Příkaz | Význam |
|---|---|---|
| Step 1 | **configure terminal** | Vstup do konfiguračního režimu |
| Step 2 | **interface** *interface-id* | Konfigurace rozhraní |
| Step 3 | **switchport protected** | Nastavení port jako protecte port |
| Step 4 | **end** | Návrat |
| Step 5 | **show interfaces** *interface-id* **switchport** | Ověření nastavení |

**Tabulka 7**

### Příklad užití protected portů




Obr. 4

Na obrázku 4 jsou nastaveny porty jedna a dva jako protected a port tři nikoliv. Pokud budou chtít komunikovat port jedna s portem dva, tak komunikace neproběhne. Pokud bude komunikovat port tři s prvním nebo druhým portem, bude komunikace probíhat.

```
Switch# configure terminal
Switch(config)# interface fastethernet0/1
Switch(config-if)# switchport protected
Switch(config-if)# end
Switch(config)# interface fastethernet0/2
Switch(config-if)# switchport protected
Switch(config-if)# end
```

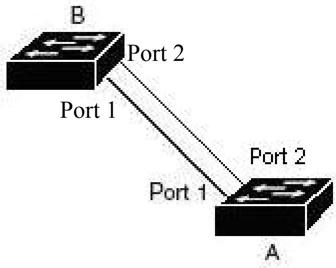
### Zhodnocení

Výše uvedený příklad byl odzkoušen na přepínači Cisco 2950. Test byl proveden tak, že jsme připojili 3 PC k přepínači na porty 1, 2 a 3. Port 1 a 2 jsme nastavili jako protected porty. Dostupnost PC byla prověřena echo zprávami. Vše fungovalo tak jak bylo očekáváno, jelikož komunikace probíhala pouze mezi portem 3 a některým ze zbylých dvou portů a komunikace mezi porty 1 a 2 nebyla možná.

## Flex links

Flex linky – používají se pro redundantní připojení. Nejčastějším využitím je připojení k poskytovateli internetu nebo ve firmách, kde nechtějí provozovat spaning tree. Jedna linka je zálohou druhé. Když hlavní linka přestane komunikovat, tak druhá se stane hlavní a veškerý provoz jde přes ni. Až druhá linka opět naběhne, tak se stane záložní linkou pro hlavní linku. Jedna hlavní linka může mít pouze jednu linku záložní. Jedna záložní linka může být zálohou pro více hlavních linek.
Flex linky pracují na druhé vrstvě portu a jestli je linka aktivní se pozná z existujícího spojení.
Pokud přestane primární linka fungovat, je přechod na záložní linku okamžitý, při testování podle uvedené architektury nebylo žádné zpoždění zaznamenáno.

### Příklad užití flex linků



Obr. 5

Switche A, B jsou zapojeny podle topologie na obrázku 5. Porty 1 a 2 z přepínače A jsou připojeny k portům 1 a 2 v přepínači B. Port 1 je nakonfigurován jako primární linka. Port 2 je nakonfigurován jako její záložní linka. Protože porty 1 a 2 jsou nakonfigurovány jako Flex linky, pouze jeden port propouští provoz a druhý je v záložním režimu. V našem případě je to Port 1 a tedy provoz mezi A a B jde přes port jedna. Pokud rozpojíme linku z portu jedna, tak se zapne port 2 a provoz začne procházet přes něj. Pokud port 1 znovu zapojíme, přejde do záložního režimu a provoz bude dál procházet portem 2.

### Nastavení flex link na rozhraní

| | **Příkaz** | **Význam** |
|---|---|---|
| Step 1 | **configure terminal** | Vstup do konfiguračního režimu |
| Step 2 | **interface** *interface-id* | Konfigurace rozhraní |
| Step 3 | **switchport backup interface** *interface-id* | Konfigurace Layer 2 rozhraní (nebo port channel). Když linka je v forwarding traffic, nebo rozhraní pak je v módu standby. |
| Step 4 | **end** | Návrat |
| Step 5 | **show interface** [*interface-id*] **switchport backup** | Ověření konfigurace |

**Tabulka 8**

```
Switch# configure terminal
Switch(conf)# interface fastethernet0/1
Switch(conf-if)# switchport backup interface fastethernet0/2
Switch(conf-if)# end
```

### Zhodnocení

Flex linky byly odzkoušeny na dvou přepínačích Cisco Catalyst 3560, protože přepínače typu 2950 je nepodporují. Test byl proveden podle výše uvedeného příkladu. Na každý z přepínačů jsme připojili jedno PC a pustili mezi nimi echo zprávy. Komunikace bez problému probíhala. Odpojili jsme primární linku z portu 1 a provoz během několika milisekund přešel na záložní linku a komunikace mezi PC probíhala dál bezproblémově. Po zpětném připojení linky do portu 1 se tato linka opravdu nastavila jako záložní k lince z portu 2 a na komunikaci mezi PC to nemělo žádný vliv. Při odpojení a následném připojení linky z portu 2 se linky vrátily opět během několika milisekund do původního nastavení. Vše fungovalo podle předpokladů a test proběhl úspěšně.

## TSC - Traffic storm control

Traffic storm nastane, když velké množství broadcast, unicast, nebo multicast rámců přichází na port. Přeposílání těchto rámců může způsobit zpomalení nebo celkovou nedostupnost sítě. Storm control sleduje rámce procházející portem na sběrnici přepínače a rozlišuje, jestli je rámec unicast, multicast nebo broadcast. Přepínač počítá přicházející rámce jednotlivých typů během intervalu jedné vteřiny a porovnává naměřenou hodnotu s přednastavenou hodnotou.
Storm control používá kapacitu linky k změření intenzity provozu. Hranice jsou vyjádřené jako procento celkové šířky pásma, které může být použito k provozu broadcast, unicast, nebo multicast rámců. Pokud je nastavené procento překročeno, tak je port blokován a rámce daného typu (závislé na nastavení, který typ rámců chceme sledovat) zahazovány dokud provoz neklesne opět pod danou hranici. V případě použití hysterezní charakteristiky (tj. rozdílné horní a dolní hranice) je dolní hranice obnovení provozu menší než horní hranice. Hysterezní charakteristika má vůči nastavení stejné horní i dolní hranice tu výhodu, že v případě, kdy provoz kolísá kolem nastavené horní hranice, nedochází k tak častému střídání mezi stavy blokování a propouštění provozu, neboť provoz je obnoven až po jeho klesnutí pod dolní hranici.
Rámce spanning tree jsou přeposílány i při blokování provozu na portu.
Ve výchozím nastavení je traffic storm control vypnutá. Nastavení traffic storm control na konkrétním rozhraní je uvedeno v Tabulce 9.
U non-LRE Catalyst 2950 přepínačů běžících s Cisco IOS Release 12.1(14)EA1 nebo novějším je možnost hranici udávat také v počtu rámců za vteřinu.

**Nastavení traffic storm na rozhraní**

| | **Příkaz** | **Význam** |
|---|---|---|
| Step 1 | **Configure terminal** | Vstup do konfiguračního režimu |
| Step 2 | **interface** *interface-id* | Konfigurace rozhraní |
| Step 3 | **storm-control** {**broadcast** \| **multicast** \| **unicast**} **level** {*level* [*level-low*] \| **pps** *pps* [*pps-low*]} | **level** – specifikuje horní hranici pro broadcast, multicast nebo unicast provoz na portu jako procento z celkové šířky pásma. Při dosažení této hranice port blokuje provoz.<br><br>**level-low** – je nepovinný parametr určující dolní hranici obnovení provozu. Pokud je nastaven, jeho hodnota musí být menší nebo rovna parametru **level**. Opět se nastavuje jako procento celkové šířky pásma. Pokud provoz daného typu zpět klesne pod tuto hranici, port jej opět začne propouštět.<br><br>**pps** - specifikuje horní hranici pro broadcast, multicast nebo unicast v rámcích za vteřinu. Provoz je blokován, když počet rámců přesáhne tuto hranici.<br><br>**pps-low** - je nepovinný parametr určující dolní hranici obnoveni provozu. Pokud je nastaven, jeho hodnota musí být menší nebo rovna parametru **pps**. Opět se nastavuje jako počet rámců daného typu za vteřinu. Pokud provoz daného typu zpět klesne pod tuto hranici, port jej opět začne propouštět. |
| Step 4 | **storm-control action** {**shutdown** \| **trap**} | Specifikuje akci, která se použije, když nastane storm. Standardní je odfiltrováni provozu.<br><br>**shutdown** – port přejde do error-disable stavu<br><br>**trap** - generování SNMP trap, když nastane storm. |
| Step 5 | **end** | Návrat |
| Step 6 | **show storm-control** [**interface**] [{**broadcast** \| **history** \| **multicast** \| **unicast**}] | Ověření konfigurace |

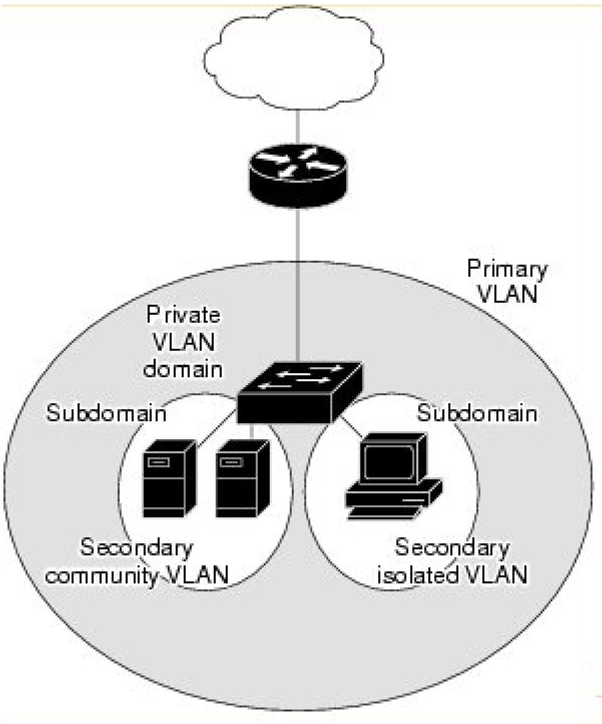
**Tabulka 9**

**Příklad konfigurace traffic storm control** (zabránění překročení hranice 70,5% celkové šířky pásma. Je sledován provoz multicast na portu 1)

```
Switch # configure terminal
Switch (config)# interface fastethernet 0/1
Switch (config-if)# storm-control multicast level 70.5
Switch (config-if)# end
```

## Private VLANs (PVLANs)

Privátní VLANy poskytují mechanismus k řízení, která zařízení smí vzájemně komunikovat uvnitř podsítě. Privatní VLANy dělí obvyklou VLAN oblast do podoblastí. Podoblast (subdomain) je reprezentována dvojicí hlavní (primary) a vedlejší (secondary) VLAN. Privátní VLAN může mít více dvojic. Všechny dvojice v privátním VLANu sdílí stejný hlavní VLAN. Více obr. 6




Obr. 6

Vedlejší VLANy mohou být dvou typů:

1. **Isolated VLANs** – porty uvnitř tohoto VLAN nemůžou vzájemně komunikovat
2. **Community VLANs** – porty uvnitř tohoto komunitního VLANu můžou vzájemně komunikovat, ale nemůžou komunikovat s porty v jiném komunitním VLANu.

Privatní VLANy poskytují izolaci na 2. vrstvě mezi porty ve stejném privatním VLANu. Porty privátních VLANu jsou tří typů:

1. **Promiscuous port** – patří do primárního VLANu a může komunikovat se všemi rozhraními (včetně isolated i community portů)

2. **Isolated port** – patří do vedlejšího VLANu typu Isolated. Je kompletně oddělen od ostatních portů ve stejném privátním VLANu s výjimkou promiscuous portů. Tzn. pouze provoz z/na promiscuous port je propouštěn na/z Isolated port.
3. **Community port** - patří do vedlejšího VLANu typu Community. Tyto porty komunikují s ostatními porty ve stejném privatním VLANu a s promiscuous porty. Jsou na 2. vrstvě izolovány od všech ostatních portů v jiném Community vedlejším VLANu a od Isolated portů ve vlastním vedlejším VLANu.

Charakteristiky VLANu
- privátní VLAN má pouze jeden hlavní VLAN.
- každý port v privátním VLANu přísluší hlavnímu VLANu
- privátní VLAN posílá jednosměrný provoz přes promiscuous porty k isolated a community portům
- privátní VLAN má pouze jeden vedlejší Isolated VLAN
- vedlejší Isolated VLAN posílá jednosměrný provoz z isolated portů přes promiscuous porty
- vedlejší Community VLAN posílá provoz z community portů k promiscuous portům a ostatním portům ve svém vedlejším VLANu. je možno definovat více vedlejších Community VLAN promiscuous port může sloužit pouze jednomu hlavnímu VLANu, jednomu vedlejšímu Isolated VLANu a více vedlejším Community VLANům

**Nastavení a sjednocení VLANů v privátní VLAN**

| | **Příkaz** | **Význam** |
|---|---|---|
| Step 1 | **configure terminal** | Vstup do konfiguračního režimu |
| Step 2 | **vlan** *vlan-id* | Konfigurace VLANu |
| Step 3 | **private-vlan primary** | Nastavení VLAN jako hlavní VLAN. |
| Step 4 | **exit** | Návrat |
| Step 5 | **vlan** *vlan-id* | Konfigurace VLANu |
| Step 6 | **private-vlan isolated** | Nastavení VLAN jako isolated VLAN. |
| Step 7 | **exit** | Návrat |
| Step 8 | **vlan** *vlan-id* | Konfigurace VLANu |
| Step 9 | **private-vlan community** | Nastavení VLAN jako community VLAN. |
| Step 10 | **exit** | Návrat |
| Step 11 | **vlan** *vlan-id* | Konfigurace hlavního VLANu |
| Step 12 | **private-vlan association** [**add** \| **remove**] *secondary_vlan_list* | Přidružení vedlejších VLANů k tomuto hlavnímu VLANu |
| Step 13 | **end** | Návrat |
| Step 14 | **show vlan private-vlan** | Ověření konfigurace |

**Tabulka 10**

### Konfigurace portu jako port vedlejšího VLANu

| | Příkaz | Význam |
|---|---|---|
| Step 1 | **configure terminal** | Vstup do konfiguračního režimu |
| Step 2 | **interface** *interface-id* | Konfigurace rozhraní |
| Step 3 | **switchport mode private-vlan host** | Konfigurace portu jako portu vedlejšího VLANu |
| Step 4 | **switchport private-vlan host-association** *primary_vlan_id secondary_vlan_id* | Přiřazení portu k hlavnímu a vedlejšímu VLANu |
| Step 5 | **end** | Návrat |
| Step 6 | **show interfaces** [*interface-id*] **switchport** | Ověření konfigurace |

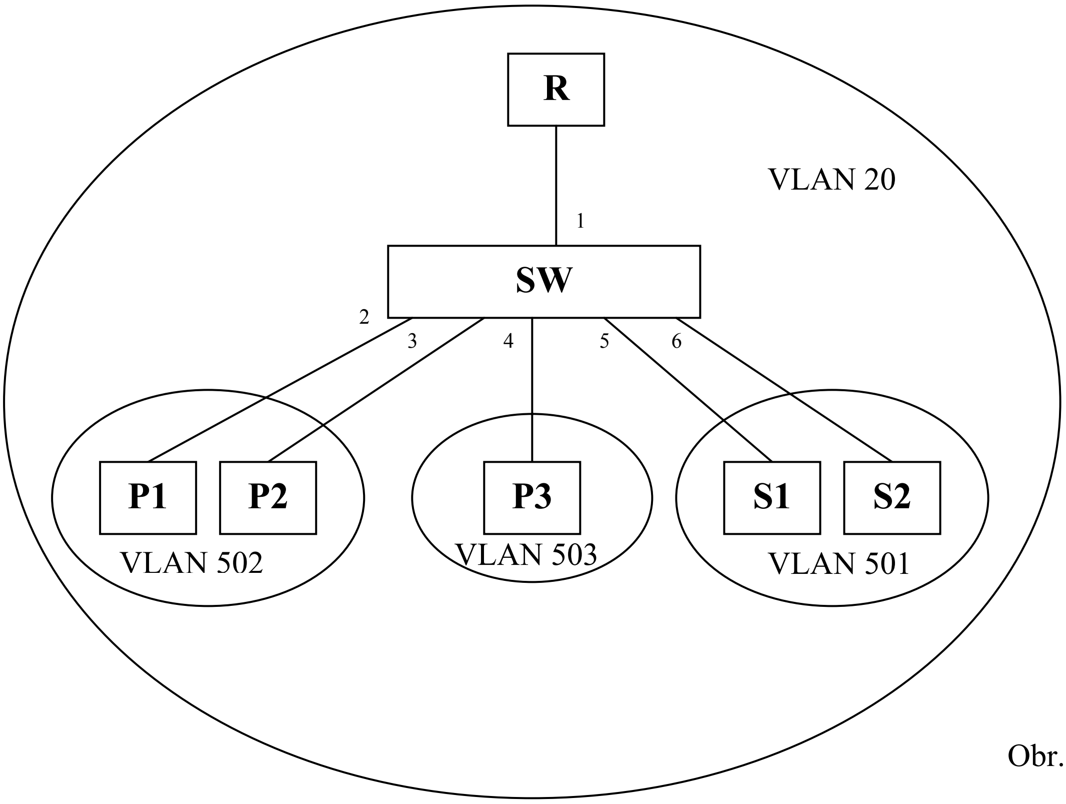
**Tabulka 11**

### Konfigurace portu jako promiscuous portu

| | Příkaz | Význam |
|---|---|---|
| Step 1 | **configure terminal** | Vstup do konfiguračního režimu |
| Step 2 | **interface** *interface-id* | Konfigurace rozhraní |
| Step 3 | **switchport mode private-vlan promiscuous** | Konfigurace portu jako promiscuous port |
| Step 4 | **switchport private-vlan mapping** *primary_vlan_id* {**add** \| **remove**} *secondary_vlan_list* | Mapování promiscuous portu k hlavnímu VLANu a označeným vedlejším VLANům. |
| Step 5 | **end** | Návrat |
| Step 6 | **show interfaces** [*interface-id*] **switchport** | Ověření konfigurace |

**Tabulka 12**

Jednoduchý příklad použití základních funkcí privátních VLANů:




Obr. 7

V příkladě na obrázku číslo 7 předpokládáme, že využijeme 6 portu přepínače SW. Na port 1 bude připojen router R, který bude představovat připojení k poskytovateli internetu, tzn., bude promiscuous portem (hlavní VLAN 20), neboť zbylých 5 využitých portů bude jakoby využívat internetové připojení. Na portu 2 a 3 budou počítače P1 a P2 , které spolu mohou komunikovat, tzn., jsou ve stejné komunitě (Community VLAN 502). Na portu 4 bude připojen počítač, který bude mít vlastní komunitu (Community VLAN 503). Na portu 5 a 6 budou připojeny dva servery, které mají být přístupné z internetu, ale nesmí spolu ani s ostatními počítači vnitřní sítě vzájemně komunikovat, tzn., budou izolovány od sebe i vnitřní sítě (Isolated VLAN 501).

Příklad konfigurace na přepínači Cisco 3560 Catalyst, Software 12.2.(20)SE:

Vytvoříme hlavní VLAN 20 a vedlejší VLANy 501,502 a 503, které přiřadíme hlavnímu VLANu. Konfigurace je následující:

```
Switch# configure terminal
Switch(config)# vlan 20
Switch(config-vlan)# private-vlan primary
Switch(config-vlan)# exit
Switch(config)# vlan 501
Switch(config-vlan)# private-vlan isolated
Switch(config-vlan)# exit
Switch(config)# vlan 502
Switch(config-vlan)# private-vlan community
Switch(config-vlan)# exit
Switch(config)# vlan 503
Switch(config-vlan)# private-vlan community
Switch(config-vlan)# exit
Switch(config)# vlan 20
```

```
Switch(config-vlan)# private-vlan association 501-503
Switch(config-vlan)# end
```

Dále přiřadíme jednotlivé porty k vytvořeným vedlejším VLANům podle zadání:

```
Switch# configure terminal
Switch(config)# interface fastethernet0/2
Switch(config-if)# switchport mode private-vlan host
Switch(config-if)# switchport private-vlan host-association 20 502
Switch(config-if)# end
Switch(config)# interface fastethernet0/3
Switch(config-if)# switchport mode private-vlan host
Switch(config-if)# switchport private-vlan host-association 20 502
Switch(config-if)# end
Switch(config)# interface fastethernet0/4
Switch(config-if)# switchport mode private-vlan host
Switch(config-if)# switchport private-vlan host-association 20 503
Switch(config-if)# end
Switch(config)# interface fastethernet0/5
Switch(config-if)# switchport mode private-vlan host
Switch(config-if)# switchport private-vlan host-association 20 501
Switch(config-if)# end
Switch(config)# interface fastethernet0/6
Switch(config-if)# switchport mode private-vlan host
Switch(config-if)# switchport private-vlan host-association 20 501
Switch(config-if)# end
```

Ověřením konfigurace pomocí příkazu **show vlan private vlan** získáme:

```
Primary Secondary Type              Ports
------- --------- ----------------- ------------------------------------
20      501       isolated          Fa0/5, Fa0/6
20      502       community         Fa0/2, Fa0/3
20      503       community         Fa0/4
```

Dále nastavíme port 1 jako promiscuous a namapujeme ho k vedlejším VLAN 501, 502 a 503:

```
Switch# configure terminal
Switch(config)# interface fastethernet0/1
Switch(config-if)# switchport mode private-vlan promiscuous
Switch(config-if)# switchport private-vlan mapping 20 add 501-503
Switch(config-if)# end
```

## Zhodnocení

Příklad byl odzkoušen a vše fungovalo bez problému. Ověřili jsme si, že z izolovaného portu 5 lze komunikovat pouze s hlavním portem. Porty 2 a 3 vzájemně komunikovali stejně, jako šla komunikace s hlavním portem, avšak odpovědi od izolovaného portu 5 i od portu 4 z jiné komunity byly blokovány. Hlavní port 1 mohl komunikovat se všemi porty.