Tunelování VLAN a servisních protokolů 2. vrstvy v síti poskytovatele

Jan Vavříček, Jan Gaura

10.6.2006

Obsah

1 Úvod aneb „K čemu je tunelování protokolů dobré“

Díky „technologii“ tunelování jsme schopni před směrovači, nebo přepínači „zamaskovat“ zařízení, které se nachází na spoji mezi nimi. Představme si tuto situaci na příkladu (který je použit i v dalších odstavcích) firmy, které má na dvou geograficky rozdílných místech svoje pobočky, ale z nějakého důvodu požaduje, aby celá její síť (v první a druhé pobočce dohromady) tvořila jednu broadcast doménu (spojitá na 2. vrstvě) a přitom aby v síti nebylo možno „odhalit“ zařízení poskytovatele (realizující propojení poboček).

Ono „zamaskování“ zařízení poskytovatele spočívá v podstatě v „nevměšování se“ do protokolů – např.: STP (Spanning Tree Protocol), CDP (Cisco Discovery Protokol), ... Není těžké si uvědomit, že STP běžící na síťové topologii složené ze dvou komponent (v našem příkladu – LAN jednotlivých poboček) a prvku je spojující, dopadne rozdilně, pokud tento spojovací prvek bude do STP vstupovat, nebo bude tvořit pouze spojovací tunel. Podobná situace je i u dalších protololů 2. vrstvy.

Než přistoupíme k vlastní konfiguraci, připoměnme si co jsou a k čemu slouží VLANy (norma 802.1q) - budeme je při konfiguraci potřebovat.

2 802.1q

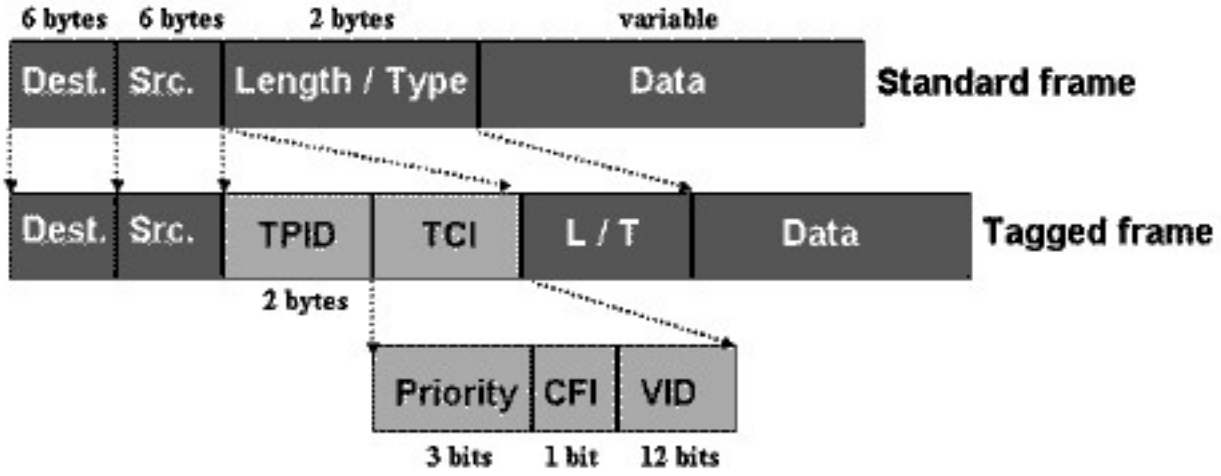
2.1 Trocha historie

Historie virtuálních sítě začíná rokem 1995. Tehdy, rok po uvedení prvních přepínačů, se objevily první proprietární implementace VLAN. Ihned si získaly pozornost uživatelů tím, že umožní správcům snadnou segmentaci sítě do logických subsítí, které jsou nezávislé na fyzické vrstvě, virtuální sítě mohou zjednodušit úlohy managementu, jako jsou např. přesun či přidání pracovní stanice a vytváření logických pracovních skupin.

2.2 Tvar 802.1q rámce

Technologie, která stojí za virtuálními sítěmi, je tedy založena na přepínačích. Virtuální sítě posouvají segmentaci sítí o krok dále tím, že oddělují fyzickou vrstvu sítě (první vrstvu dle OSI modelu) od logické síťové topologie.

Fungování 802.1q je poměrně triviální – ethernetový rámec, který prochází z jednoho přepínače do druhého, aby dosahl cílové stanice, která se nachází ve stejném VLANu, je označen značnou (tagem) – to jsou čtyři byty ke každému rámci navíc (čili overhead této technologie je víceméně zanedbatelný - viz obrázek 1).



Obrázek 1: Formát ethernetového rámce podle normy 802.1q

Popis položek 802.1q rámce:

TPID Pole TPID má definovanou hodnotu 8100 (hex). Má-li tedy toto pole (EtherType) hodnotu 8100, rámec nese informace IEEE 802.1q/ 802.1p

Priority umožňuje zakódování až osmi úrovní priority rámců (0-7, nula je přitom nejnižší priorita) dle standardu 802.1p.

VID jednoznačně identifikuje VLAN, do které rámec přísluší. Z velikosti pole (12 bitů) plyne teoretický max. počet virtuálních sítí ($2^{12} = 4096$).

Při používání 802.1q můžeme identifikovat dva základní režimy provozu portů na přepínačích – jednak tzv. přístupové (access porty), po nichž teče provoz již „netagovaný“ (typicky přepínač u odchozího rámce tag odstraní a naopak příchozí rámec tagem označí), a pak tzv. trunk porty, na nichž veškerý provoz naopak „otagovaný“ je. Označování rámců má pochopitelně smysl jen při propojení více přepínačů dohromady.

Pro snadnější pochopení uveďme jednoduchý příklad: mějme dva přepínače, které jsou propojeny dohromady (např.: každý náleží jednomu patru budovy, přičemž potředujeme mít jednotlivá patra propojena). V některých portech přepínačů (a tedy i různých patrech) jsou připojeni například administratoři a v jiných ekonomové. Chceme zajistit, aby „provoz“ administrátorů a ekonomů byl od sebe oddělen a přitom byli propojeni všichni administrátoři, respektive ekonomové. Řešení nám nabízí 802.1q: na obou přepínačích přidělíme portům administratorům jedno VID (VLAN ID) a ekonomům druhé (např.: `0x111` a `0x666`) – tím z nich uděláme access porty. Tímto zajístíme, že na **JEDNOTLIVÝCH** přepínačích se provoz „nebude míchat“. My ale potřebujeme mít propojeny administrátory respektive ekonomy z více přepínačů (pater)! Toho dosáhneme propojením přepínačů trunk portem. Teď již máme síť v požadavaném stavu.

Za jediné úskalí používání 802.1q lze považovat délku rámce, která může překročit velikost bufferu ovladače síťových karet nebo přímo HW, jehož obvyklá velikost pro 10/100Base-T činí 1518 bajtů.

2.3 Tunelování VLANů

Mějme znovu příklad naší firmy se dvěmi oddělenými pobočkami. Navíc tato firma používá VLANy a tedy potřebuje, aby otagované rámce „procházely“ mezi pobočkami. Spojení ale ve skutečnosti prochází přes poskytovatele připojení a ten nejspíš také využívá VLANy. Takže pokud by bylo připojení realizováno pouze trunk portem, provoz zákazníka a poskytovatele by se „míchal“ a provoz by nemusel dorazit ke správnému cíli.

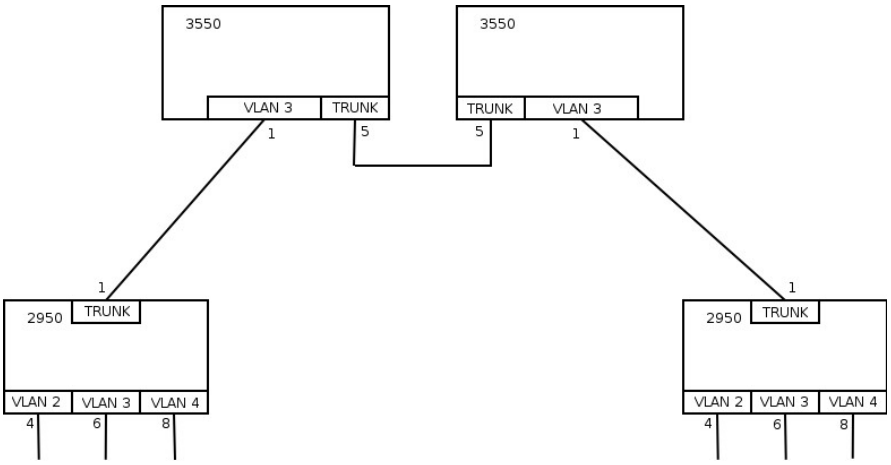
Když poskytovatel, který vlastní 802.1q infrastrukturu, prodá zákazníkovi spoj v rámci své sítě, prodává mu vlastně jen jednu VLAN. Nicméně zákazník také využívá 802.1q a chce přes tento spoj přenášet své, již tagované, rámce – a tedy více VLANů. Existují samozřejmě polovičatá řešení tohoto problému, jako je varianta nabídnout zákazníkovi k dotyčnému spoji více VLAN (a fyzickou realizaci přípojky provést jako trunk), ale to je záležitost z hlediska administrace a zejména koordinace se zákazníkem poměrně náročná – a to jak při samotné instalaci, tak samozřejmě i při řešení provozních problémů či údržbě. Takové řešení není ideální pro zákazníka a už vůbec ne pro poskytovatele, mimo jiné i proto, že VLANů je pouze 2^{12}.

Řešení problému však existuje – je jím technologie **802.1q over 802.1q**, zkráceně Q-in-Q. O co jde? Myšlenka je poměrně jednoduchá – když mohu označit rámec jednou, proč jej neoznačit vícekrát? Již otagovaný rámec je tedy otagován znovu. První otagování rámce provede kupříkladu zařízení zákazníka, druhé otagování přístupové zařízení poskytovatele.

3 802.1q over 802.1q

Co je na celé věci zajímavé, je skutečnost, že Q-in-Q nemusejí podporovat ani zařízení zákazníka, ani všechna zařízení poskytovatele na trase, ale pouze ta zařízení, která provádějí druhé označení rámce (a pochopitelně zároveň odstranění značky v opačném směru) tzv. **edge** přepínač (zapojení na obrázku 2 bylo použito k testování Q-in-Q). Ostatní zařízení si přečtou pouze druhý (tedy přesněji v dotyčném rámci v pořadí první, jako druhý byl přidán) tag, považují tento rámec za součást jedné VLAN a další obsah rámce je nezajímá. Na edge zařízeních přibývá kromě režimu linek trunk a access další režim fungování portu – režim **dot1q-tunnel**, v němž se port přepínače chová víceméně jako v režimu access, s rozdílem že otaguje všechny rámce nezávisle na tom, zda již nějaký tag mají či nikoli (čili již otagované rámce otaguje znovu respektive odstrani „vnější“ tag z 2x tagovaných rámců). Zařízení zákazníka, zapojené proti portu v režimu dot1q-tunnel, je nakonfigurováno v běžném režimu trunk. Jediné, co je nezbytné vyladit, je délka rámců, která nám o další čtyři byty roste každým „zabalením“.

Na portu přepínače, nastaveného v režimu dot1q-tunnel, je možné zapnout tunelování různých L2 protokolů, jako je STP, CDP, apod., takže pro zákazníka se Q-in-Q tunel může



Obrázek 2: Testované zapojení, realizující tunelování

tvářit jako čistý point-to-point spoj bez další L2 infrastruktrury na trase.

Teď se pochopitelně nabízí otázka, zda je možné otagovat rámec ještě vícekrát (tedy Q-in-Q-in-Q nebo Q-in-Q-in-Q-in-Q atd). Pochopitelně to možné je a víceméně v dalším tagování nikomu nic nebrání, kromě nutnosti postupného rozbalování rámců v kaskádě přepínačů a také hlídání maximální délky rámce. Teoreticky je tedy při MTU 9 kB možné 1518 bajtů dlouhý netagovaný rámec otagovat zhruba 1750x, přičemž overhead v takovém případě činí cca 800 %, u kratších rámců i mnohem více. Otázka pochopitelně zní, k čemu je to dobré.

4 Konfigurace Q-in-Q na přepínačích Cisco

Rozdělme si přepínače na dvě skupiny – přepínače zákazníka a poskytovatele. Naši snahou bude nakonfigurovat Q-in-Q pokud možno tak, aby zákazník NIC nevědel a tedy nemusel nic specialního nastavovat.

Na obrázku 2 je naznačeno testovací zapojení. Zákazník má na svých přepínačích nastaveny pouze VLANy – a to podle svých potřeb. Port vedoucí k poskytovateli (v obrázku 2 a testovaných konfiguracích má číslo 1) je v režimu trunk. To je veškeré požadované nastavení na straně zákazníka – postačí mu tedy přepínače „rozumící“ si pouze s VLANy (v testovacím zapojení použity přepínače Cisco 2950).

Konfigurace přepínače poskytovatele není nijak obtížná. Port příslušející konkrétnímu zákazníkovi patří do VLANu, kterým si tohoto zákazníka označíme (jeho specifická identifikace). Tímto identifikátorem VLANu je označen znovu 802.1q rámec přicházející od zákazníka (který má rámec označet svým VLANem – jde tedy o zmíněný „vnější“ tag) – označován jako **metro VLAN**. Takový Q-in-Q rámec odchází trunk portem do sítě poskytovatele. Na cestu k druhéhu edge přepínači jsou kladeny dva nároky: MTU (Maximal Transfer Unit) musí být zvětšena (na edge přepínačích i přepinačích na cestě) a provoz musí procházet skrze trunk porty.

Část konfigurace „zákaznického“ přepínače (kompletní konfigurace jsou v přílohách):

```
vlan 2
name customer-VLAN2
vlan 3
name customer-VLAN3
vlan 5
name customer-VLAN5
interface FastEthernet0/1
switchport mode trunk
interface FastEthernet0/4
switchport access vlan 2
interface FastEthernet0/6
switchport access vlan 3
interface FastEthernet0/10
switchport access vlan 5
```

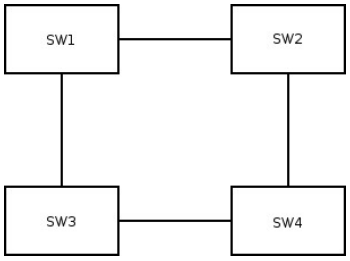
Část konfigurace přepínače poskytovatele:

```
system mtu 1525
vlan 3
name MyCustomer-VLAN3
interface FastEthernet0/1
switchport access vlan 3
switchport mode dot1q-tunnel
l2protocol-tunnel cdp
l2protocol-tunnel stp
l2protocol-tunnel vtp
interface FastEthernet0/5
switchport trunk encapsulation dot1q
switchport mode trunk
interface FastEthernet0/10
switchport access vlan 3
switchport mode dot1q-tunnel
l2protocol-tunnel cdp
l2protocol-tunnel stp
l2protocol-tunnel vtp
```

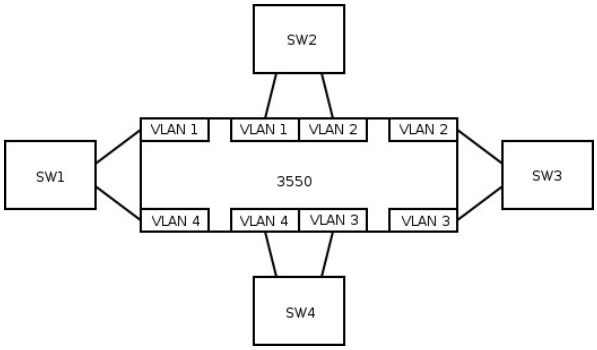
Příkazy **l2protocol-tunnel** slouží k tunelování protolů 2. vrstvy. Lze je použít i bez tunelovaní VLANů (Q-in-Q) a obdobně Q-in-Q lze taky provozovat bez tunelování protokolů 2. vrstvy – viz další odstavce.

5 Tunelování STP a CDP

Na obrázku 3 je zapojení, které požadujeme – co si ale počít, pokud jsou přepínače zapojený do nějakého centrálního přepínače (kterým chceme podobné topologie vytvářet elektronicky bez nutnosti fyzického zásahu do propojení přepínačů), jak je to zobrazeno v obrázku 4? Řešení je jednoduché – využijeme tunelování.



Obrázek 3: Požadované „logické“ zapojení přepínačů



Obrázek 4: Realizace propojení přepínačů z obrázku 3

Nejprve propojíme „sousední“ přepínače pomocí VLANů definovaných na centrálním přepínači, jak bylo naznačeno na obrázku 4. Teď již máme přepínače propojený, ale centralní prvek se nám bude „plest“ do STP (Spanning Tree Protocol) a bude se objevovat v CDP (Cisco Discovery Protocol) namísto „sousedních“ přepínačů.

Touto konfigurací dosáhneme potřebného efektu – teď se centrálni prvek nebude projevovat v STP a nebude se propagovat do CDP, ale bude tyto protokoly „propuštět“.

Konfigurace jednoho z portů na centrálním prvku:

```
!
interface FastEthernet0/1
switchport access vlan 3
switchport mode access
l2protocol-tunnel cdp
l2protocol-tunnel stp
l2protocol-tunnel vtp
no cdp enable
!
```

6 Závěr

Pokud se vrátíme k našemu příkladu s firmou, uvedením příkazů pro tunelování protokolů 2. vrstvy získamé na edge přepínači (spolu s již nastaveným Q-in-Q) tunel, kterým nezměněně prochází SPT, CDP a 802.1q – takže celý spoj mezi pobočkami (který může být osazen mnoha prvky) se pro nás bude jevit jako spojení „point-to-point“.

A Konfigurace zákaznického přepínače

```
  !
version 12.1
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname SW3
!
ip subnet-zero
!
no ip domain-lookup
vtp mode transparent
!
spanning-tree mode pvst
no spanning-tree optimize bpdu transmission
spanning-tree extend system-id
!
vlan 2
name customer-VLAN2
!
vlan 3
name customer-VLAN3
!
vlan 5
name customer-VLAN5
!
interface FastEthernet0/1
switchport mode trunk
!
interface FastEthernet0/4
switchport access vlan 2
!
interface FastEthernet0/5
!
interface FastEthernet0/6
switchport access vlan 3
!
interface FastEthernet0/10
switchport access vlan 5
!
ip http server
!
line con 0
logging synchronous
line vty 0 4
login
line vty 5 15
login
!
end
```

B Konfigurace přepínače poskytovatele

```
 version 12.1
no service pad
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname SW-RS2-ISP
!
ip subnet-zero
!
no ip domain-lookup
ip ssh time-out 120
ip ssh authentication-retries 3
vtp domain cisco
vtp mode transparent
!
spanning-tree mode pvst
spanning-tree extend system-id
!
vlan 3
name MyCustomer-VLAN3
!
interface FastEthernet0/1
switchport access vlan 3
switchport mode dot1q-tunnel
l2protocol-tunnel cdp
l2protocol-tunnel stp
l2protocol-tunnel vtp
!
interface FastEthernet0/2
switchport mode dynamic desirable
!
interface FastEthernet0/3
switchport mode dynamic desirable
!
interface FastEthernet0/4
switchport mode dynamic desirable
!
interface FastEthernet0/5
switchport trunk encapsulation dot1q
switchport mode trunk
!
interface FastEthernet0/6
switchport mode dynamic desirable
!
interface FastEthernet0/7
switchport mode dynamic desirable
!
interface FastEthernet0/8
switchport mode dynamic desirable
!
interface FastEthernet0/9
switchport mode dynamic desirable
!
interface FastEthernet0/10
switchport access vlan 3
switchport mode dot1q-tunnel
!
interface FastEthernet0/11
switchport mode dynamic desirable
!
interface FastEthernet0/12
switchport mode dynamic desirable
!
interface FastEthernet0/13
switchport mode dynamic desirable
!
interface FastEthernet0/14
switchport mode dynamic desirable
!
interface FastEthernet0/15
switchport mode dynamic desirable
!
ip classless
ip http server
!
line con 0
logging synchronous
line vty 0 4
login
line vty 5 15
login
!
end
```