

Protokol Spanning Tree (802.1d) v OS GNU/Linux

Zpracovali: Tomáš Hrabálek a Jiří Kotraba, [Vysoká škola báňská – Technická univerzita Ostrava](#)
Datum vydání: 4.4.2006

Cíl projektu

Úkolem tohoto projektu bylo zjistit, jakým způsobem se konfiguruje ethernetové bridge v OS GNU/Linux a jak se ním provozuje protokol *Spanning Tree*. Dále bylo cílem vyzkoumat, jakým způsobem pracuje STP v různých VLAN a je-li možno takto spustit více oddělených instancí.

Obecné informace

Protokol IEEE 802.1d, neboli Spanning Tree Protocol (STP) je dlouhou dobu znám na mnoha „lepších“ ethernetových přepínačích. Ovšem pořídit tato zařízení je nákladné, a tak je zcela jistě na místě myšlenka, zda by nebylo možno tento protokol používat i na osobních počítačích s několika ethernetovými kartami.

Roku 2000 vydal [Matematický institut na univerzitě v Leidenu](#) v Nizozemsku první verzi balíku Bridge, který obsahoval patche jádra OS Linux a utility, které modul jádra spravovaly. Tento balík se postupně vylepšoval, až byl zařazen přímo do jádra OS Linux. Sídlo projektu je nyní na webových stránkách <http://linux-net.osdl.org/index.php/Bridge>, kde najdete veškerou potřebnou dokumentaci i soubory ke stažení.

Příprava operačního systému

Před vytvořením bridge nejdříve zkontrolujte, zda máte nainstalováno jádro Linuxu verze 2.3.47 a pozdější¹. Ověřte, že do jádra byla přidána podpora 802.1d a v případě, že zde není, upravte konfiguraci a jádro zkompilejte a spusťte Linux s tímto funkčním jádrem².

Nástroje na konfiguraci bridge

Instalace

Jestliže máme jádro s podporou bridgování, pak si připravíme ovládací software. K veškeré konfiguraci používáme utility s názvem **brctl**. Zkuste, zda-li je v systému nainstalována. Není-li, potom si stáhněte zdrojové kódy balíčku s názvem bridge-utils a zkompilejte je. Případně zkuste nalézt binární balíček pro vaši distribuci, který se bude jmenovat buď *bridge-utils* (např. Fedora), nebo *brctl* (např. Ubuntu). Pokud se vám s balíčkem nenainstalovaly manuálové stránky, pak je doporučeno je také nainstalovat.

¹ Samozřejmě nejlépe stabilní verze 2.4.x nebo 2.6.x. Nejnovější najdete na [webových stránkách](#).

² Nejběžnější distribuce jako [Ubuntu](#) nebo [Fedora](#) tuto podporu zakompilovanu mají.

Příkaz brctl

Syntaxe příkazu *brctl*:

brctl	addbr	<bridge>	vytvoř bridge
	delbr	<bridge>	smaž bridge
	addif	<bridge> <zařízení>	přidej rozhraní k bridge
	delif	<bridge> <zařízení>	smaž rozhraní z bridge
	setageing	<bridge> <čas>	nastav ageing time
	setbridgeprio	<bridge> <prio>	nastav bridge priority
	setfd	<bridge> <čas>	nastav bridge forward delay
	sethello	<bridge> <čas>	nastav hello time
	setmaxage	<bridge> <čas>	nastav max message age
	setpathcost	<bridge> <port> <cena>	nastav cenu cesty
	setportprio	<bridge> <port> <prio>	nastav prioritu portu
	show		ukaz seznam bridgů
	showmacs	<bridge>	ukaz seznam mac adres
	showstp	<bridge>	ukaz bridge stp info
	stp	<bridge> <stav>	přepni stp on/off

Vytvoření bridge

Most, neboli bridge, je v linuxové implementaci virtuální síťové rozhraní, na které se navazují již existující (fyzická) síťová zařízení. Lze tedy vytvořit několik mostů a na každý z nich navázat libovolný síťový interface a to včetně i dalšího mostu. Je rovněž možné, aby jedno síťové rozhraní bylo navázáno na několik mostů zároveň.

Každý most je jednoznačně identifikován svým názvem, který si v Linuxu můžeme zvolit libovolně³. Přes jméno tohoto mostu jej potom můžeme konfigurovat. Jelikož se jedná o síťový interface (i když virtuální), jeho funkce jsou stejné jako u fyzického. Lze tedy tomuto rozhraní přiřadit IP adresu a provozovat na něm síťové služby.

K zapnutí bridgování je třeba vytvořit alespoň jednu instanci mostu. Tu vytvoříme:

```
brctl addbr jmenomostu
```

K němu související příkaz je:

```
brctl delbr jmenomostu
```

který rozhraní mostu smaže. To, že se virtuální rozhraní skutečně vytvořilo můžeme ověřit zavoláním známého příkazu:

```
ifconfig -a
```

Ve výpisu by se měl váš most objevit.

Přidání síťového rozhraní do bridge

Provedením příkazu:

```
brctl addif jmenomostu zarizeni
```

připojíme síťové zařízení k mostu. Všechna připojená zařízení vytvoří jakoby jednu ethernetovou síť. Je samozřejmě nemožné a nesmyslné přidávat jedno zařízení k více mostům. Souvisejícím příkazem je:

```
brctl delif jmenomostu zarizeni
```

který dané síťové zařízení odpojí od mostu.

Příklad:

```
brctl addbr most1      #vytvoříme instanci mostu

brctl addif most1 eth0 #přidáme eth0 k našemu bridge
```

³ Toto není standardizováno. Lze setkat s označením *br0*, *br1*, ..., ale my budeme používat *most0*, *most1*, ...

```
brctl addif most1 eth1 #přidáme eth1 k našemu bridge

ifconfig eth0 0.0.0.0 #vymažeme ip adresu a tím vypneme tcp/ip na eth0
ifconfig eth1 0.0.0.0 #na bridge není tcp/ip potřeba

ifconfig most1 up #zapneme náš bridge
```

Zapnutí podpory Spanning Tree Protocol (STP)

Implicitně je podpora protokolu Spanning Tree u mostů vypnuta. Zapnutí, provedeme příkazem:

```
brctl stp jmenomostu on
```

Tímto spustíme instanci STP na daném mostu. Je samozřejmě možné, máme-li více mostů, spustit na každém z nich zvláštní instanci STP. Potřebujeme-li STP vyřadit, napíšeme příkaz:

```
brctl stp jmenomostu off
```

Praktické použití mostů

Mosty použijeme v případech, potřebujeme-li například fyzicky oddělit dvě síťové topologie – například *Ethernet-Ethernet*, bez nutnosti směrování ve vyšší vrstvě. Most mezi stejnými síťovými technologiemi můžeme s úspěchem použít třeba u malého směrovače pro několik počítačů z nichž směřujeme třeba do Internetu, kdy můžeme několik síťových rozhraní svázat do mostu a vytvořit tak malý switch a jedno rozhraní půjde do vnější sítě. Ovšem tato varianta je z úspěchem použitelná jen pro velmi malý počet počítačů (obecně v domácnosti), které mezi sebou příliš nekomunikují a spíše je jejich datový tok spojen s vnější sítí – v našem případě s Internetem.

Spojení STP a VLAN

To, že na každém z mostů může běžet nezávislá instance STP, můžeme v praxi velmi dobře využít. Máme-li v síti několik VLANů, potom je možné na každém z nich takovouto instanci provozovat a lze teoreticky dosáhnout různě spočítaných topologií pro různé VLAN. Toto ukážeme dále.

Vytvoření VLAN rozhraní

Máme-li k síťové kartě připojenu *trunk* linku, kde běží několik VLANů, pak lze pomocí příkazu *vconfig* rozdělit síťové rozhraní této karty na logická rozhraní, kde každé náleží jednomu VLANu. Toto provedeme příkazem:

```
vconfig add síťové_rozhraní číslo_vlan
```

Takto nám vznikne nové rozhraní s označením „*síťové_rozhraní.číslo_vlan*“, kam bude chodit výhradně provoz daného VLANu.

Příklad:

```
#k rozhraní eth0 je připojena trunk linka s VLANy 2 a 3

vconfig add eth0 2 #vznikne eth0.2
vconfig add eth0 3 #vznikne eth0.3

ifconfig -a #prohlédneme si nová rozhraní
```

Vytvoření mostů s různými VLAN na počítači se dvěma síťovými kartami

Předpokládejme, že máme počítač se dvěma ethernetovými kartami – eth0 a eth1. Ke každé z nich je připojena trunk linka a běží zde VLANy 2 a 3. Úkolem je vytvořit mosty mezi stejnými VLANy.

Provedení:

```
vconfig add eth0 2 #vznik eth0.2 (vlan 2)
```

```

vconfig add eth0 3      #vznik eth0.3 (vlan 3)
vconfig add eth1 2      #vznik eth1.2 (vlan 2)
vconfig add eth1 3      #vznik eth1.3 (vlan 3)

brctl addbr most2       #vytvoření mostu pro vlan 2
brctl addif most2 eth0.2 #přidání vlan 2 zařízení
brctl addif most2 eth1.2

brctl addbr most3       #vytvoření mostu pro vlan 3
brctl addif most3 eth0.3 #přidání vlan 3 zařízení
brctl addif most3 eth1.3

```

Nezávislé STP pro VLAN mosty

Jednoduchým zadáním

```

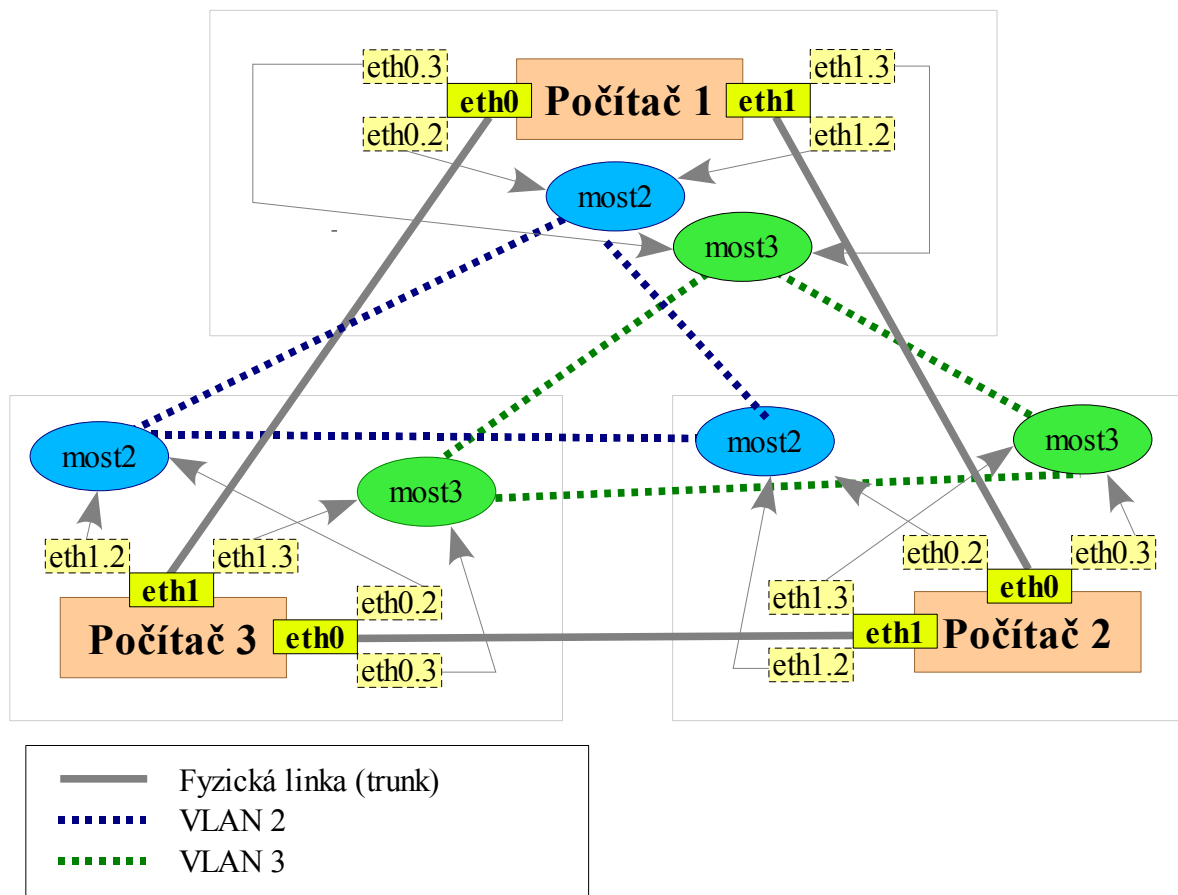
brctl stp most2 on      #STP pro vlan 2
brctl stp most3 on      #STP pro vlan 3

```

spustíme na obou mostech zvláštní instance STP.

Test nezávislých STP na síti se smyčkami

Mějme tři počítače nakonfigurované způsobem uvedeným v předchozí kapitole – tedy dva mosty a každý z nich připojený na stejný VLAN k různým rozhraním. Tyto počítače spojíme do trojúhelníku. Vznikne nám tak síť se smyčkami tvořená trunk linkami. Nebyl-li by spuštěn protokol STP, brzy by se síť zahltila kruhovým obíháním paketů. Protože STP zapnuté máme, některé z rozhraní se vyblokuje a k zahlcení nedojde. Díky naší konfiguraci se ovšem vyblokuje zvlášť pro každý VLAN. Tak může dojít k tomu, že provoz VLAN 2 jde jinou trasou než VLAN 3. Díky tomu využijeme i doposud nepoužívané spoje a nezatěžujeme méně linky dosud používané.



Výsledky testů

Po zapojení sítě – jak je to ukázáno na obrázku, jsme vysílali ping z počítače 3 do počítače 1. Spuštěním příkazu

```
brctl showstp
```

a také odposloucháváním rozhraní eth0 a eth1 na všech počítačích programem *Ethereal* bylo patrné, že na obou VLANech se vyblokovaly linky přímé mezi 1 a 3, tudíž provoz chodil přes počítač dvě. Když jsme uměle vypnuli rozhraní eth0.2 na počítači 3, provoz VLAN 2 šel přímo, zatímco provoz VLAN 3 pokračoval původní trasou.

Když jsme vypojoili kabel, potom oba VLANy chodily novou trasou (jak se ukázalo – stejnou, což je i předpokládatelné).

Podíváme-li se, jak vypadají pakety, které chodí sítí, pak každý odchycený paket byl zabalen a označován podle svého VLANu. Takto zabalené chodily i zprávy STP, proto je možno spustit několik instancí.

Změna topologie pro každý VLAN zvlášť

Chceme-li, aby data každého VLANu chodila jinou cestou, potom toto provedeme nastavením cen linek – a to pro každou VLAN jinak. Syntaxe nastavení je popsána v kapitole *Příkaz brctl*. Proto uvedeme příklad:

```
# VLAN 2
brctl setpathcost most2 eth0.2 20
brctl setpathcost most2 eth1.2 30
# VLAN 3
brctl setpathcost most3 eth0.3 30
brctl setpathcost most3 eth1.3 20
```

Tímto způsobem jsem nastavili STP, aby pro VLAN2 priorizoval jinou cestu, než pro VLAN3.

Pokusy s přepínači

Při přidání přepínače mezi počítače – místo přímého spojení dvou rozhraní jedním kříženým UTP, přestala síť fungovat. Zařízením sice pakety procházely, ale to se netýkalo protokolu STP. Toto chování patrně závisí na typu a výrobci switchu a jeho nastavení.

Nevýhody zvláštních STP pro různé VLANy

Obrovskou nevýhodou provozu více STP najednou je režie. Sítí totiž neustále putují STP pakety a uvážíme-li, jak často jsou vysílány a vynásobíme-li si to počtem VLANů, zjistíme, že dost síťové kapacity je zabráno STP, protože si každá instance přepočítává topologii zvlášť.

Další nevýhodou je již zmiňované nepřeposílání zabalených STP některými přepínači. Je nutno se podrobně seznámit s dokumentací přepínače a zjistit, je-li možno takový druh provozu přes něj protlačit.

Závěr

Mosty v OS Linux jsou velmi prakticky použitelné a jak známo i používané (například v ADSL routerech pro domácnosti), mohou sloužit jako oddělovač různých síťových technologií nebo jako malý přepínač. Ve spojení s VLANy se v Linuxu objevují zajímavé možnosti provozu Spanning Tree Protocolu v různých instancích při jedné fyzické topologii a mohou posloužit pro odlehčení využívaných linek pomocí redundantních spojů.