PV177 - Laboratoř pokročilých síťových technologií
Cvičení z L2/L3 síťování
Protokol 6 - Směrování mezi autonomními systémy

Michal Šnajdr - 325268

podzim 2013

Cíl cvičení

Vyzkoušet si směrování mezi autonomními systémy. Konkrétně použití protokolu BGP, který je v praxi k tomuto účelu jediný používaný.

Použitá zařízení

- R1 - Mikrotik RouterBoard 800, označení 1, SN 3DA10298B3F9 RouterOS 5.16
- R2 - Mikrotik RouterBoard 800, označení 7, SN 3DA10204E24A, RouterOS 5.16
- R101 - Mikrotik RouterBoard 800, označení 4, SN 3DA102F59D36, RouterOS 5.16
- R102 -Mikrotik RouterBoard 800, označení 3, SN 3DA10243095A, RouterOS 5.16

Vypracování

Testovací síť logicky odpovídala obrázku 1. . Směrovače R1 a R2 byly připraveny vyučujícím jako součást zadání. Naším úkolem bylo nastavit směrovače AS 100 s adresním prostorem 100.1.0.0/20 R101 a R102 pro iBGP a navázat eBGP sezení mezi směrovači R1-R101 a R2-R102. Ostatní skupiny spravují vlastní autonomní systémy X00 s adresními prostory 100.X.0.0/20, které byly k R1 a R2 připojeny obdobně jako AS 100.

Fyzicky byly směrovače R1 a R2 propojeny mezi s sebou a ostatními AS X00 v laboratoři pomocí switche, kdy pro každou /30 linku byla vyhrazena samostatná vlan a k směrovačům R1 a R2 vedla linka typu trunk.

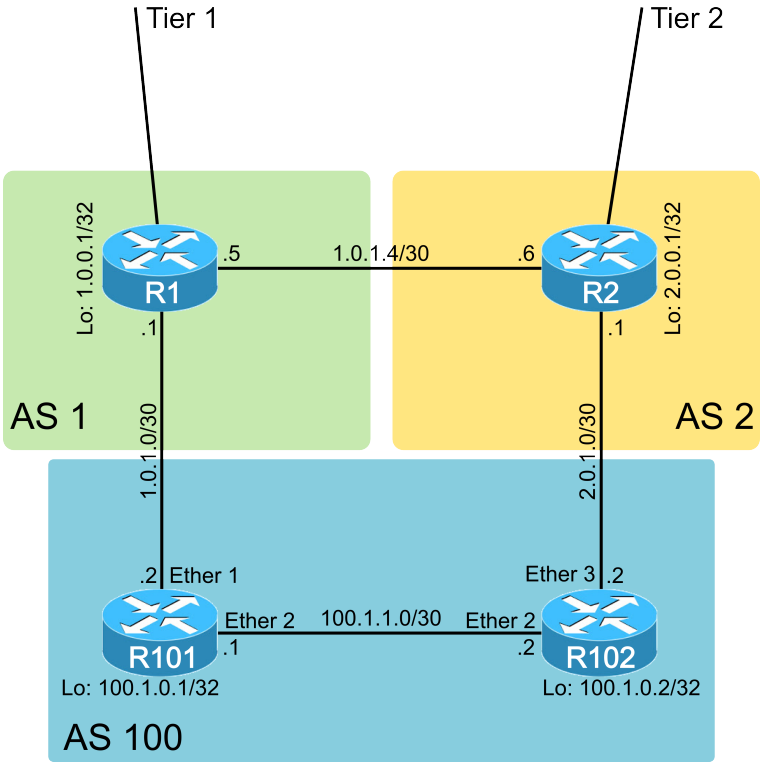
Základní nastavení

Místo zařízení *loopback*, které není v ROS přítomno jsme použili zařízení bridge, do nějž nebylo zařazeno žádné rozhraní [1]. Protože stav zařízení *bridge* není svázán s žádnou fyzickou linkou nikdy nebude nedostupný a tím splňuje charakteristiky speciálního zařízení *loopback* používaného na zařízeních jiných výrobců. Příslušným rozhraním jsme přiřadili IP adresy.

```
R101:
/interface bridge add name=loopback
/ip address add address=100.1.0.1/32 interface=loopback
/ip address add address=100.1.1.1/30 interface=ether2
/ip address add address=1.0.1.2/30 interface=ether1

R102:
/interface bridge add name=loopback0
/ip address add address=100.1.0.2/32 interface=loopback
/ip address add address=100.1.1.2/30 interface=ether2
/ip address add address=2.0.1.2/30 interface=ether1
```

Také jsme odstranili adresu *192.168.88.1/24* z rozhraní *ether1*, kde je přítomna ve výchozí konfiguraci zařízení.

Dále jsme nastavili IGP protokol našeho autonomního systému. Zvolili jsme OSPF. Problematika protokolu OSPF byla podrobně probírána v protokolu 5 a zde nebude konfigurace podrobně popisována.



Obrázek 1: Logické zapojení zadání.

```
R101:
/routing ospf instance set [ find default=yes ] router-id=100.1.0.1
/routing ospf interface add disabled=no interface=ether1 passive=yes
/routing ospf network
add area=backbone disabled=no network=100.1.1.0/30
add area=backbone disabled=no network=100.1.0.1/32
add area=backbone disabled=no network=1.0.1.0/30

R102:
/routing ospf instance set [ find default=yes ] router-id=100.1.0.2
/routing ospf interface add disabled=no interface=ether3 passive=yes
/routing ospf network
add area=backbone disabled=no network=100.1.1.0/30
add area=backbone disabled=no network=100.1.0.2/32
add area=backbone disabled=no network=2.0.1.0/30
```

Nyní je na obou směrovačích dostatečné množství směrovacích informací aby jsme mohli komunikovat mezi adresami loopbacků a navázat mezi nimi iBGP relaci. V rámci iBGP se relace navazují mezi loopback interface právě kvůli jejich vlastnosti nezávislosti na stavu linky. V reálné síti, která není tak jednoduchá jako příklad ze zadání existuje více redundantních cest a dostupnost loopback zařízení dokud existuje alespoň jedna cesta k směrovači zajistí, že iBGP spojení nespadne ani v případě výpadku některé z linek mezi peery. K zajištění směrování v rámci AS není použito BGP a musí zde běžet některý z IGP protokolů (v našem případě OSPF) nebo být nadefinovány statické cesty (neflexibilní nepraktické řešení).

Jako první jsme nastavili výchozí instanci BGP naše číslo AS a *router-id*. Poté jsme vytvořili iBGP spojení mezi R101 a R102. Důležitý je atribut *update-source*, který udává rozhraní/adresu, které bude použito jako zdrojové pro odchozí komunikaci a z výše uvedených důvodu by to měl

být loopback.

```
R101:
/routing bgp instance set default as=100 router-id=100.1.0.1
/routing bgp peer add name=R102 remote-address=100.1.0.2 remote-as=100
                update-source=loopback default-originate=if-installed

R102:
/routing bgp instance set default as=100 router-id=100.1.0.2
/routing bgp peer add name=R101 remote-address=100.1.0.1 remote-as=100
                update-source=loopback0 default-originate=if-installed
```

Tím byla navázána iBPG relace.

Dále jsme nastavili relace do AS našich ISP. Tato spojení se navazují mezi adresami rozhraní spojující sítě mezi směrovači a to z důvodu detekce výpadku linky. Při sestavení spojení mezi loopback rozhraními by při pádu linky nezískal směrovač okamžitou informaci o nedostupnosti cílové sítě (mezi různými AS typicky neběží IGP, který by tuto událost zaznamenal) a k detekci by došlo až po vypršení časovače (v ROS standardně nastaveno na 3 minuty).

```
R101:
/routing bgp peer add name=R1 remote-address=1.0.1.1 remote-as=1 update-source=1.0.1.2

R102:
/routing bgp peer add name=R2 remote-address=2.0.1.1 remote-as=2 update-source=2.0.1.2
```

Nyní máme navázány všechna potřebná spojení:

```
[admin@R101] /routing bgp peer> print
Flags: X - disabled, E - established
 #   INSTANCE                            REMOTE-ADDRESS                  REMOTE-AS
 0 E default                             100.1.0.2                       100
 1 E default                             1.0.1.1                         1

[admin@R102] /routing bgp peer> print
Flags: X - disabled, E - established
 #   INSTANCE                            REMOTE-ADDRESS                  REMOTE-AS
 0 E default                             100.1.0.1                       100
 1 E default                             2.0.1.1                         2
```

Oznamování rozsahu AS

Zadání říká, že naše síť má oznamovat pouze jednu souhrnnou síť s PI IP *100.1.0.0/20*. Toto je možné nastavit dvěma způsoby, z nichž jsme na každém z routerů R101 a R102 použili jeden z nich.

R101 - ruční nastavení oznámení

Sítě, které chceme oznamovat BGP sousedům je možné ručně vkládat v menu */routing bgp network*. Zde přidáme náš souhrnný záznam *100.1.0.0/20*

```
/routing bgp network add network=100.1.0.0/16 synchronize=no
```

Důležitý je parametr *synchronize=no*. Pokud je tento parametr pro položku nastaven na *yes*, bude síť oznamována pouze v případě, že směrovač má ve své směrovací tabulce záznam pro tuto síť [3]. Tento záznam ale naše směrovače nemají a při zapnuté synchronizaci by nedošlo k propagaci záznamu.

Odkazovaná dokumentace sice uvádí, že výchozí hodnotou pro parametr *synchronize* je *no*, naše zkušenost je ale opačná.

R102 - agregace

Na R102 jsme vyzkoušeli jiný postup. Pro BGP proces jsme nastavili, aby přeposílal informace o připojených sítích a sítích, které se naučil přes OSPF. Toto způsobý propagování jednotlivých malých rozsahů, pro které jsme následně nastavili agregaci.

```
/routing bgp instance set default redistribute-connected=yes redistribute-ospf=yes
/routing bgp aggregate add prefix=100.1.0.0/20 summary-only=yes
```

Atribut *summary-only=yes* způsobí, že bude oznamována pouze společná síť *100.1.0.0/20*. Pokud by byl nastaven na *no*, vytvořila by agregace pouze další oznamovanou síť [3].

Dle mého názoru je vhodnější první způsob, protože je přímočařejší a konfigurace je poté snáze čitelná.

Po domluvě s vyučujícím nám směrovače R1 i R2 posílají i „default gateway“ - 0.0.0.0/0.

Další konfigurace v jednotlivých úkolech jsou rozšířením této společné konfigurace.

Úkol 1

Jsme zákazníkem s PI IP adresami. Úkolem je nastavit směrovače v našem AS tak, aby používaly pouze připojení přes AS 1 a přes linku do AS 2 komunikovaly až při pádu linky k AS 1. Náš AS není tranzitivní.

Nastavili jsme následující vlastnosti:

- na R101 i R102 nastavili výstupní filtr pro BGP komunikaci s R1 respektive R2, který povolí oznamovat pouze naši síť *100.1.0.0/20*
- na R102 jsme nastavili výstupní filtr, který do oznámení naší sítě *100.1.0.0/20* směrem k R2 předřadí 4× číslo našeho AS
- na R101 jsme nastavili vstupní filtr, který pro všechny cesty přijaté od R1 nastaví hodnotu *local preference* na 200

```
R101:
/routing filter
add action=discard chain=to-R1 invert-match=yes prefix=100.1.0.0/20
add action=accept chain=from-R1 prefix=0.0.0.0/0 prefix-length=0-32 set-bgp-local-pref=200
/routing bgp peer set R1 in-filter=from-R1 in-filter=from-R1

R102:
/routing filter
add action=passthrough chain=to_R2 prefix=100.1.0.0/20 set-bgp-prepend=4
add action=discard chain=to_R2 invert-match=yes prefix=100.1.0.0/20
/routing bgp peer set R2 out-filter=to-R2
```

V tomto úkolu by k dosažení stejné funkce směrování stačilo přijímat od AS 1 a AS 2 pouze výchozí cestu. V rámci řešení cvičení pro nás bylo přehlednější přijímat kompletní směrovací informace, které sousedé nabízejí.

Výstupní filtr na R102 pro update k R2 (*chain=to-R2*) se skládá ze 2 pravidel:

- 1. pravidlo přidá náš prefix 4× do odesílané AS-PATH. Toto se aplikuje pouze na záznam naší sítě *100.1.0.0/20*. Protože je nastaveno *action=passthrough* zpracování záznamu při aplikaci tohoto pravidla nekončí. Hodnota 4 se zdá být celkem nízká, ale vzhledem k průměrné délce AS-PATH v Internetu, která je 4 [6] je prepend 4 naprosto dostatečný.
  Toto pravidlo zajistí (díky znalosti propojení mezi AS 1 a AS2) preferenci linky mezi R1

a R101 pro příchozí provoz do *AS 100*, protože i z AS 2 bude nižší *AS-PATH-LENGTH* do *AS 100* přes AS 1. Toto samozřejmě předpokládá, že správce AS 2 nenastavil směrovač R2 způsobem, aby preferoval přímou linku mezi AS 2 a *AS 100* (například nastavení vyšší *local preference* obdobně jako je níže popsáno pro R101. *Local preference* v AS 2 by mohlo ovlivnit naše záměry, protože je upřednostňováno při výběru nejlepší cesty před délkou AS-PATH [2, 5]).

- 2. pravidlo zahodí (*action=discard*) všechny prefixy, které nejsou *100.1.0.0/20* (*invert-match=yes*). Protože není explicitně nastavena hodnota atributu *prefix-length* tak se použije výchozí hodnota *32* [4] a pravidlem projde bez zahození pouze oznámení sítě *100.1.0.0/20*. Toto by vyfiltrovalo jednotlivé podsítě s maskou větší než 20 i pokud by jsme v části *R102 - agregace* nastavili *summary-only=no* (což by ale bylo zbytečné plýtvání HW prostředky směrovače).

Řetězec pravidel filtru nemá na svém konci implicitní pravidlo *discard*, jako tomu bývá například u FW pravidel. Pokud záznam při průchodu filtrem projde na konec bez toho aby byl některým pravidlem schválen dříve (*action=accept*) nebo zahozen (textitaction=discard) je propuštěn k dalšímu zpracování. Chová se tedy spíše jako implicitní *accept*.

Na R102 není žádný vstupní filtr pro informace přijaté z R2

Výstupní filtr na R101 pro update i R1 se skládá pouze z jednoho pravidla, které je shodné jako 2 pravidlo na R102 a má stejný účel.

Vstupní filtr na R101 pro informaci přijaté od R1 (*chain=from-R1*) má jediný úkol. Pro všechny informace přijaté od R1 nastavit hodnotu *local-preference* na 200. Vyšší hodnota *local-preference* zajistí preferenci cest přijatých od R1 [2]. Naše řešení předpokládá, že informace mezi *AS 1 a AS 2* nejsou filtrovány a R1 nám přeposílá vše, co se dozvěděl od R2. Toto je v rámci cvičení splněno. Pokud by byly informace mezi AS 1 a AS 2 filtrovány, nedostali by jsme stejnou množinu dostupných sítí z obou AS. Poté by mohla být jediná známá cesta do některé sítě přímo přes linku do AS 2 což by nevyhovovalo zadání cvičení a řešení takového stavu by nebylo jednoduché a pravděpodobně už vůbec ne systémové.

Že naše řešení splňuje zadání a jednotlivé filtry plní svůj účel jak jsme si představovali jsme si ověřili na následujících výstupech:

Výpis sítí, které ohlašujeme ostatním směrovačům:

```
[admin@R101] /routing bgp advertisements> print
PEER     PREFIX              NEXTHOP          AS-PATH    ORIGIN     LOCAL-PRE
R102     1.0.0.0/16          1.0.1.1          1          igp               200
R102     2.0.0.0/16          1.0.1.1          1,2        igp               200
R102     0.0.0.0/0           1.0.1.1          1          igp               200
R1       100.1.0.0/20        1.0.1.2                     incomplete

[admin@R102] /routing bgp advertisements> print
PEER     PREFIX              NEXTHOP          AS-PATH    ORIGIN     LOC
RX01     1.0.1.0/30          100.1.0.2                   igp
RX01     100.1.0.0/20        100.1.0.2                   incomplete
RX01     2.0.1.0/30          100.1.0.2                   incomplete
RX01     192.168.88.0/24     100.1.0.2                   incomplete
R2       100.1.0.0/20        2.0.1.2                     incomplete
```

Ve výpisu příkazu */routing bgp advertisements print* bohužel není ve sloupci *AS-PATH* uváděno naše číslo AS i když je do oznámení směřujícího do jiného AS přidáno na začátek [3]. Proto jsme si na R2 ověřili vlastnosti příchozích updatů od R102:

```
[admin@R2] /ip route> print detail where received-from=R102
```

```
Flags: X - disabled, A - active, D - dynamic, C - connect, S - static, r - rip, b - bgp,
       o - ospf, m - mme, B - blackhole, U - unreachable, P - prohibit
11  Db  dst-address=100.1.0.0/20 gateway=2.0.1.2 gateway-status=2.0.1.2 reachable via  VLAN102
    distance=20 scope=40 target-scope=10 bgp-as-path="100,100,100,100"
    bgp-origin=incomplete received-from=R102
```

Pomocí traceroute si ověříme, že provoz je přenášen jak požadujeme linkou k R1.

```
[admin@R101] > /tool traceroute 2.0.0.1
 # ADDRESS                                 RT1   RT2   RT3   STATUS
 1 1.0.1.1                                 1ms   1ms   1ms             < R1
 2 2.0.0.1                                 1ms   1ms   1ms             < R2

[admin@R102] /tool> traceroute 2.0.0.1
 # ADDRESS                                 RT1   RT2   RT3   STATUS
 1 100.1.1.1                               1ms   1ms   1ms             < R101
 2 1.0.1.1                                 1ms   1ms   1ms             < R1
 3 2.0.0.1                                 1ms   1ms   1ms             < R2
```

Směrovací tabulka na R2 říká, že provoz do našeho *AS 100* bude směrován přes AS 1. V tabulce je vidět i druhý záznam s přímou linkou k R102, který není právě aktivní.

```
[admin@R2] /ip route> print
Flags: X - disabled, A - active, D - dynamic, C - connect, S - static, r - rip, b - bgp,
       o - ospf, m - mme, B - blackhole, U - unreachable, P - prohibit
 #      DST-ADDRESS        PREF-SRC        GATEWAY            DISTANCE
 <output omited>
10 ADb  100.1.0.0/20                       1.0.1.5                 20
11  Db  100.1.0.0/20                       2.0.1.2                 20
```

Nyní jsme rozpojili linku mezi R101 a R1, směrovací tabulky byly automaticky upraveny na linku mezi R102 a R1:

```
[admin@R101] > /tool traceroute 2.0.0.1
 # ADDRESS                                 RT1   RT2   RT3   STATUS
 1 1.0.1.1                                 1ms   1ms   1ms              < R1
 2 2.0.0.1                                 1ms   1ms   1ms              < R2

[admin@R102] > /tool traceroute 2.0.1.1
 # ADDRESS                                 RT1   RT2   RT3   STATUS
 1 2.0.1.1                                 1ms   1ms   1ms              < R2

 [admin@R1] > /ip route print
Flags: X - disabled, A - active, D - dynamic, C - connect, S - static, r - rip, b - bgp,
       o - ospf, m - mme, B - blackhole, U - unreachable, P - prohibit
 #      DST-ADDRESS        PREF-SRC        GATEWAY            DISTANCE
  <output omited>
10 ADb  100.1.0.0/20                       1.0.1.6                 20
```

Úkol 2

Opět jsme zákazníkem s PI IP adresami a náš AS není tranzitivní. Oproti úkolu 1 není linka mezi R2 a R102 pouze záložní konektivitou, ale je požadováno, aby provoz do AS 2 a k němu přímo připojených AS směroval přes AS 2. Zbytek provozu prochází přes AS 1.

- Filtry pro zajištění netranzitnosti našeho AS jsou stejné jako v úkolu 1.
- Na R102 nastavíme filtr pro oznámení od R2 ve kterém:
  - zahazujeme prefixy s délkou AS-PATH větší než 2 (dále než AS 2 a jeho přímí sousedé)
  - přijatým prefixům nastavíme *local-preference* na 150

- Na R101 nastavíme filtr ve kterém:
  - zahazujeme prefixy s délkou AS-PATH větší než 1 (pouze prefixy AS 1)
  - přijatým prefixům nastavíme *local-preference* na 190

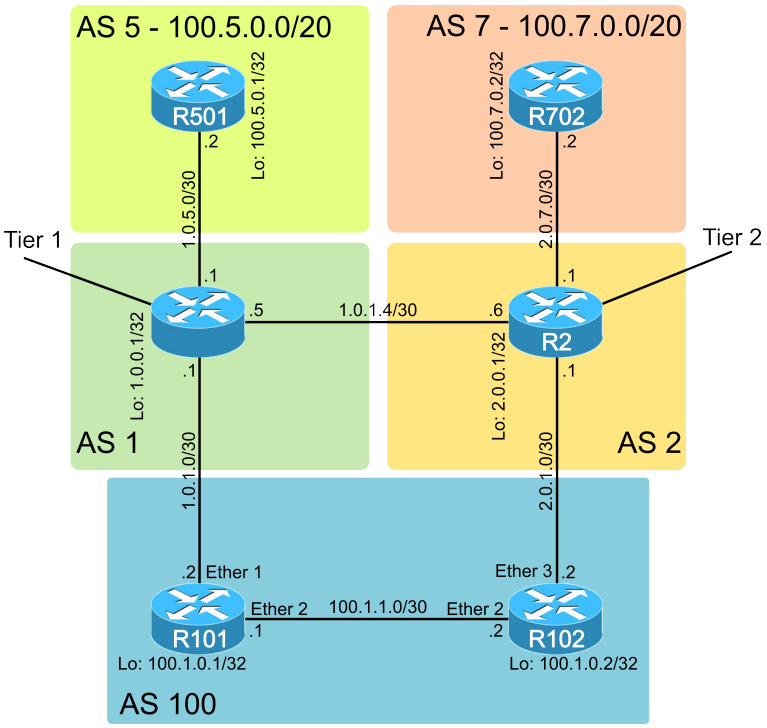
```
R101:
/routing filter
add action=discard chain=to-R1 invert-match=yes prefix=100.1.0.0/20
add action=discard chain=from-R1 bgp-as-path-length=0-1 invert-match=yes
add action=accept chain=from-R1 set-bgp-local-pref=190
/routing bgp peer set R1 in-filter=from-R1 in-filter=from-R1

R102:
/routing filter
add action=discard chain=from-R2 bgp-as-path-length=0-2 invert-match=yes
add action=accept chain=from-R2 set-bgp-local-pref=150
add action=discard chain=to_R2 invert-match=yes prefix=100.1.0.0/20
/routing bgp peer set R2 out-filter=to-R2
```

Nastavení nejvyšší priority (z námi používaných) pro prefixy patřící AS 1 (délka AS-PATH 1) nám zajistí přímé směrovaní na AS 1. Mezi těmito prefixy je i výchozí cesta. Od AS 2 přijímáme jeho prefixy a prefixy jeho sousedů (délka AS-PATH < 2). Mezi těmito prefixy je i výchozí cesta. Protože ta přijatá od AS 1 bude mít v našem AS vyšší prioritu, nebude výchozí cesta od AS 2 použita dokud nepadne linka k AS 1. Ostatní prefixy přijaté z AS 2 (kromě prefixů AS 1) budou použity, protože je od AS 1 nepřijímáme a jsou tedy jedinými záznamy o těchto sítích. Směrování do přímých sousedů AS 1 bude prováděno podle výchozí cesty (ale například do AS ostatních skupin v cvičeních bude směrováno přes AS 2 protože v tomto případě „zvítězí“ vlastnost přímého připojení k AS 2 a přímý záznam pro jejich adresní prostor přijatý od AS 2 v našich směrovacích tabulkách).

```
[admin@R101] /routing filter> /ip route print
Flags: X - disabled, A - active, D - dynamic, C - connect, S - static, r - rip, b - bgp,
       o - ospf, m - mme, B - blackhole, U - unreachable, P - prohibit
 #      DST-ADDRESS        PREF-SRC        GATEWAY            DISTANCE
 0 ADb  0.0.0.0/0                          1.0.1.1                  20
 1 ADb  1.0.0.0/16                         1.0.1.1                  20
  <output omited>
 4 ADb  2.0.0.0/16                         2.0.1.1                 200
  <output omited>
11 ADb  100.2.0.0/20                       2.0.1.1                 200
```

Pro ověření správnosti jsme si připojili další dva AS jak je naznačeno na obrázku 2. Níže uvedené příklady příkazu *traceroute* ukazují, že provoz je směrován požadovaným způsobem. Bohužel již nedokážeme ovlivnit příchozí provoz tak, aby data z AS 7 putovala do našeho AS přes AS 1, ale použijí nejkratší cestu přímo přes linku R2-R102.

```
[admin@R101] > tool traceroute 100.5.0.1
 # ADDRESS                                 RT1   RT2   RT3   STATUS
 1 1.0.1.1                                 1ms   1ms   1ms             < R1
 2 100.5.0.1                               1ms   1ms   1ms             < R501

[admin@R101] > tool traceroute 100.7.0.2
 # ADDRESS                                 RT1   RT2   RT3   STATUS
 1 100.1.1.2                               1ms   1ms   1ms             < R102
 2 2.0.1.1                                 1ms   1ms   1ms             < R2
 3 100.7.0.2                               1ms   1ms   1ms             < R702

[admin@R101] > /ip route print
Flags: X - disabled, A - active, D - dynamic, C - connect, S - static, r - rip, b - bgp,
```



Obrázek 2: Rozšířená síť pro ověření správnosti řešení.

```
       o - ospf, m - mme, B - blackhole, U - unreachable, P - prohibit
 #      DST-ADDRESS        PREF-SRC        GATEWAY            DISTANCE
 0 ADb  0.0.0.0/0                          1.0.1.1                  20
 1 ADb  1.0.0.0/16                         1.0.1.1                  20
 4 ADb  2.0.0.0/16                         2.0.1.1                 200
11 ADb  100.2.0.0/20                       2.0.1.1                 200
12 ADb  100.7.0.0/20                       2.0.1.1                 200

[admin@R101] > /ip route print detail
Flags: X - disabled, A - active, D - dynamic, C - connect, S - static, r - rip, b - bgp,
       o - ospf, m - mme, B - blackhole, U - unreachable, P - prohibit
 0 ADb  dst-address=0.0.0.0/0 gateway=1.0.1.1 gateway-status=1.0.1.1 reachable via  ether1
        distance=20 scope=40 target-scope=10 bgp-as-path="1" bgp-local-pref=190
        bgp-origin=igp received-from=R1

 1 ADb  dst-address=1.0.0.0/16 gateway=1.0.1.1 gateway-status=1.0.1.1 reachable via  ether1
        distance=20 scope=40 target-scope=10 bgp-as-path="1" bgp-local-pref=190
        bgp-origin=igp received-from=R1

 4 ADb  dst-address=2.0.0.0/16 gateway=2.0.1.1 gateway-status=2.0.1.1 recursive via 100.1.1.2 ether2
        distance=200 scope=40 target-scope=30 bgp-as-path="2"
        bgp-local-pref=150 bgp-origin=igp received-from=R102

 11 ADb  dst-address=100.2.0.0/20 gateway=2.0.1.1 gateway-status=2.0.1.1 recursive via 100.1.1.2 ether2
         distance=200 scope=40 target-scope=30 bgp-as-path="2,200"
         bgp-local-pref=150 bgp-origin=igp received-from=R102

12 ADb  dst-address=100.7.0.0/20 gateway=2.0.1.1 gateway-status=2.0.1.1 recursive via 100.1.1.2 ether2
        distance=200 scope=40 target-scope=30 bgp-as-path="2,700"
        bgp-local-pref=150 bgp-origin=igp received-from=R102
```

Nakonec jsme vyzkoušeli, jak naše konfigurace reaguje na výpadky. V obou případech výpadku linky k ISP byl veškerý provoz automaticky směrován přes zbývající linku:

Vyřazena linka R1-R101

```
[admin@R101] > tool traceroute 100.7.0.2
 # ADDRESS                                 RT1   RT2   RT3   STATUS
 1 100.1.1.2                               1ms   1ms   1ms             < R102
 2 2.0.1.1                                 1ms   1ms   1ms             < R2
 3 100.7.0.2                               1ms   1ms   1ms             < R702

[admin@R101] > tool traceroute 100.5.0.1
 # ADDRESS                                 RT1   RT2   RT3   STATUS
 1 100.1.1.2                               1ms   1ms   1ms             < R102
 2 2.0.1.1                                 1ms   1ms   1ms             < R2
 3 1.0.1.5                                 1ms   1ms   1ms             < R1
 4 100.5.0.1                               1ms   1ms   1ms             < R501
```

Vyřazena linka R2-R102

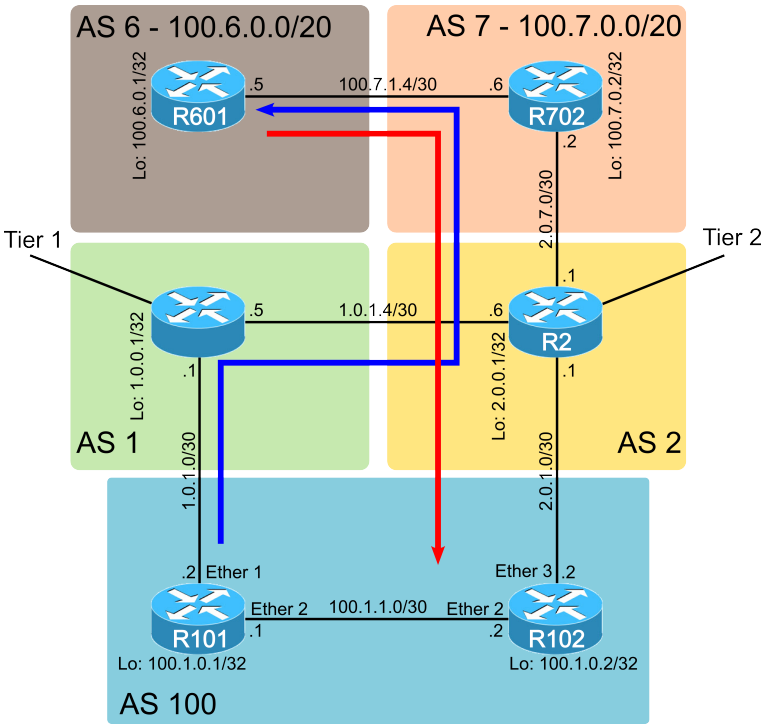
```
[admin@R101] > tool traceroute 100.7.0.2
 # ADDRESS                                 RT1   RT2   RT3   STATUS
 1 1.0.1.1                                 1ms   1ms   1ms             < R1
 2 1.0.1.6                                 1ms   1ms   1ms             < R2
 3 100.7.0.2                               1ms   1ms   1ms             < R702

[admin@R101] > tool traceroute 100.5.0.1
 # ADDRESS                                 RT1   RT2   RT3   STATUS
 1 1.0.1.1                                 1ms   1ms   1ms             < R1
 2 100.5.0.1                               1ms   1ms   1ms             < R501

[admin@R102] > /tool traceroute 100.7.0.2
 # ADDRESS                                 RT1   RT2   RT3   STATUS
 1 100.1.1.1                               1ms   1ms   1ms             < R101
 2 1.0.1.1                                 1ms   1ms   1ms             < R1
 3 1.0.1.6                                 0ms   1ms   1ms             < R2
 4 100.7.0.2                               1ms   1ms   1ms             < R702

[admin@R102] > /tool traceroute 100.5.0.1
 # ADDRESS                                 RT1   RT2   RT3   STATUS
 1 100.1.1.1                               1ms   1ms   1ms             < R101
 2 1.0.1.1                                 1ms   1ms   1ms             < R1
 3 100.5.0.1                               1ms   1ms   1ms             < R501
```

Zajímavostí je možná asymetrie mezi odchozím a příchozím provozem, která může v této konfiguraci vzniknout. Příkladem by byla například situace na obrázku 3 kdy k AS 7 bude připojen další AS. Protože AS 6 není přímým sousedem AS 2, je náš provoz k němu směrován podle výchozí cesty přes AS 1 i když to není nejkratší cesta. Provoz z AS 6 do našeho AS bude směrován nejkratší cestou přímo z AS 2.

Úkol 3

Třetím úkolem bylo rozdělit náš adresní prostor na sítě /24 a následně zajistit, aby provoz k lichým sítím přicházel z AS 1 a provoz pro sudé sítě přicházel z AS 2. Zadání nezmiňuje jak má být směrován odchozí provoz z našeho AS a proto jej necháváme best-effort. Náš AS není tranzitivní.

Naše řešení je založeno na tom, že směrovače R101 a R102 oznamují jako v předchozích úkolech souhrnou cestu, ale navíc každý z nich oznamuje příslušné /24 sítě jejichž provoz má být přes ně přijímán. Oznamováním celé ho adresního rozsahu si zajišťujeme jednoduchou redundanci spojení. V případě že fungují obě uplink linky, tak směrovací záznamy pro /24 z obou směrovačů R101 a R102 pokrývají celý náš adresní prostor 100.1.0.0/20 a jako specifičtější prefixy budou



Obrázek 3: Příklad asymetrického směrování.

použity pro směrování. V případě výpadku jedné linky k našemu ISP budou ze směrovacích tabulek R1 a R2 odebrány příslušné sudé/liché prefixy /24 a směrování do těchto prefixů bude pokryto obecnějším prefixem /20. Tímto řešením jsme se vyhnuli manipulaci s prepend.

Oproti předchozím úkolům se trochu změnil filtr který povoluje oznamování pouze našeho adresního rozsahu. V předchozích úkolech byl nastaven aby povoloval pouze souhrnné oznámení sítě 100.1.0.0/20, nyní musí povolit i /24 prefixy. Filtr je stejný na R101 i R102. Dále jsme do */routing bgp network* ručně přidali oznámení pro příslušné /24 sítě.

```
/routing filter
add action=discard chain=to-R1 disabled=no invert-match=yes
    prefix=100.1.0.0/20 prefix-length=20-24 set-bgp-prepend-path=""

/routing bgp network add network=100.0.1.0/24 synchronize=no
....
```

Nahlédnutí do směrovacích tabulek směrovačů R1 a R2 nám prozradilo, že naše konfigurace je správná.

```
[admin@R1] > /ip route print
 <output omited>
11 ADb  100.1.0.0/20                    1.0.1.2                 20
12  Db  100.1.0.0/20                    1.0.1.6                 20
15 ADb  100.1.1.0/24                    1.0.1.2                 20
18 ADb  100.1.3.0/24                    1.0.1.2                 20
22 ADb  100.1.6.0/24                    1.0.1.2                 20
23  Db  100.1.6.0/24                    1.0.1.6                 20

[admin@R2] > /ip route print
 #      DST-ADDRESS        PREF-SRC        GATEWAY            DISTANCE
  <output omited>
13 ADb  100.1.1.0/24                    1.0.1.5                 20
14 ADb  100.1.2.0/24                    2.0.1.2                 20
```

Následující výpisy ukazují průchod paketů do naší sítě z AS 200. Adresa 100.1.13.1 je nastavena loopbacku směrovače R102 a adresa 100.1.12.1 je nastavena loopbacku na směrovači R101. Adresy jsou schválně pro otestování nastaveny na opačném směrovači, než by měly do našeho AS pakety pro tyto sudé/liché sítě vstupovat. Samozřejmostí pro funkčnost směrování je správné nastavení IGP v našem AS.

```
[admin@R201] > /tool traceroute 100.1.13.1
 # ADDRESS                                 RT1   RT2   RT3   STATUS
 1 1.0.2.1                                 1ms   1ms   1ms            < R1
 2 1.0.1.2                                 1ms   1ms   1ms            < R101
 3 100.1.13.1                              1ms   1ms   1ms            < R102

[admin@R201] > /tool traceroute 100.1.12.1
 # ADDRESS                                 RT1   RT2   RT3   STATUS
 1 100.2.1.2                               1ms   1ms   1ms            < R202
 2 2.0.2.1                                 1ms   1ms   1ms            < R2
 3 2.0.1.2                                 1ms   1ms   1ms            < R102
 4 100.1.12.1                              5ms   1ms   1ms            < R101
```

Rozdělení provozu na sudé/liché sítě není ideální, protože generuje velké množství záznamů ve směrovacích tabulkách. Pro stejné rozložení zátěže by stačilo rozdělit adresní prostor na dvě /25 sítě.

Úkol 4

Posledním úkolem byla konfigurace, kdy náš AS poskytuje záložní konektivitu pro AS 2. Náš AS poskytuje konektivitu pro AS 2 v případě výpadku jiných jeho linek. Tranzitnost přes náš AS má být poslední použitou možností pro zajištění dostupnosti AS 2.

Zadání neuvádí jak se máme chovat k odchozímu provozu našeho AS. Proto jsme se rozhodli toto zadání vypracovat jako rozšíření úkolu 2, kde nahradí netranzitnost našeho AS.

Oproti konfiguraci z úkolu 2 jsme provedli dvě změny:

- v odchozím filtru na R101 a R102 už nezahazujeme cizí prefixy
  - cizím prefixům nastavíme prepend 4
- na směrovači R102 zapneme posílání výchozí cesty pro směrovač R2 s hodnotou *if-installed*. Pokud by jsme pro R2 zvolili *default-originate=always*, tak v nejhorším případě, kdy budeme tranzit pro AS2 a našel linka do AS 1 bude nedostupná, k nám bude stále od AS 2 chodit tranzitivní provoz, který budeme muset zahazovat. S hodnoutou *if-installed* při výpadku linky k AS 1 ztratíme z našich směrovacích tabulek výchozí cestu a nebudeme ji dále distribuovat k AS 2.

  Nyní při zpracování protokolu si uvědomuji, že vytvážet záznam výchozí cesty na naší straně ani nebylo nutné, protože jej přijmeme od AS 1 a AS 2 poté přepošleme. V případě výpadku spojení s AS 1 bude tato výchozí cesta odebrána s našich směrovacích tabulek a informaci o jejím zneplatnění přepošleme i AS 2.

V syntaxi konzole RouterOS tedy nastavení vypadá následovně:

```
[admin@R101] /routing filter> print
Flags: X - disabled
 0   chain=to-R1 prefix=100.1.0.0/20 invert-match=no action=accept set-bgp-prepend-path=""

 1   chain=to-R1 prefix=0.0.0.0/0 prefix-length=0-32 invert-match=no action=accept
     set-bgp-prepend=4 set-bgp-prepend-path=""
```

```
 2   chain=from-R1 bgp-as-path-length=0-1 invert-match=yes action=discard
     set-bgp-local-pref=200 set-bgp-prepend-path=""

 3   chain=from-R1 invert-match=no action=accept set-bgp-local-pref=190 set-bgp-prepend-path=""

[admin@R102] /routing filter> print
Flags: X - disabled
 0   chain=to_R2 prefix=100.1.0.0/20 invert-match=no action=accept set-bgp-prepend-path=""

 1   chain=to_R2 prefix=100.1.0.0/20 invert-match=yes action=accept set-bgp-prepend=4
     set-bgp-prepend-path=""

 2   chain=from_R2 bgp-as-path-length=0-2 invert-match=no action=accept
     set-bgp-local-pref=150 set-bgp-prepend-path=""

 3   chain=from_R2 bgp-as-path-length=0-2 invert-match=yes action=accept
     set-bgp-local-pref=100 set-bgp-prepend-path=""

[admin@R102] /routing bgp peer> print detail
 Flags: X - disabled, E - established
 1 E name="R2" instance=default remote-address=2.0.1.1 remote-as=2 tcp-md5-key=""
     nexthop-choice=default multihop=no route-reflect=no hold-time=3m ttl=255 in-filter=from_R2
     out-filter=to_R2 address-families=ip update-source=2.0.1.2 default-originate=if-installed
     remove-private-as=no as-override=no passive=no use-bfd=no
```

Pro kontrolu správné funkce jsme použili rozšířenou síť z úkolu 2 zobrazenou na obrázku 2. Ověření jsme provedli pomocí příkazu *traceroute* a kontrolou směrovacích tabulek příslušných směrovačů.

```
[admin@R702] > /tool traceroute 1.0.1.1
 # ADDRESS                                 RT1   RT2   RT3   STATUS
 1 2.0.7.1                                 1ms   1ms   1ms              < R2
 2 1.0.1.1                                 1ms   1ms   1ms              < R1

[admin@R702] > /tool traceroute 100.5.0.1
 # ADDRESS                                 RT1   RT2   RT3   STATUS
 1 2.0.7.1                                 1ms   1ms   1ms              < R2
 2 1.0.1.5                                 1ms   1ms   1ms              < R1
 3 100.5.0.1                               1ms   1ms   1ms              < R501

[admin@R2] > /ip route print  detail
Flags: X - disabled, A - active, D - dynamic, C - connect, S - static, r - rip, b - bgp,
       o - ospf, m - mme, B - blackhole, U - unreachable, P - prohibit

 1 ADb  dst-address=1.0.0.0/16 gateway=1.0.1.5 gateway-status=1.0.1.5
        reachable via  VLAN12 distance=20 scope=40 target-scope=10
        bgp-as-path="1" bgp-origin=igp received-from=R1

 2  Db  dst-address=1.0.0.0/16 gateway=2.0.1.2 gateway-status=2.0.1.2
        reachable via  VLAN102 distance=20 scope=40 target-scope=10
        bgp-as-path="100,100,100,100,1" bgp-origin=igp received-from=R102
 <output omited>

15 ADb  dst-address=100.1.0.0/20 gateway=2.0.1.2 gateway-status=2.0.1.2
        reachable via  VLAN102 distance=20 scope=40 target-scope=10 bgp-as-path="100"
        bgp-origin=incomplete received-from=R102

 <output omited>
18 ADb  dst-address=100.5.0.0/20 gateway=1.0.1.5 gateway-status=1.0.1.5
        reachable via  VLAN12 distance=20 scope=40 target-scope=10 bgp-as-path="1,500"
        bgp-origin=igp received-from=R1

[admin@R1] > /ip route print detail
Flags: X - disabled, A - active, D - dynamic, C - connect, S - static, r - rip, b - bgp,
       o - ospf, m - mme, B - blackhole, U - unreachable, P - prohibit
```

```
 <output omited>
10 ADb  dst-address=2.0.0.0/16 gateway=1.0.1.6 gateway-status=1.0.1.6
        reachable via  VLAN12 distance=20 scope=40 target-scope=10 bgp-as-path="2"
        bgp-origin=igp received-from=R2

11  Db  dst-address=2.0.0.0/16 gateway=1.0.1.2 gateway-status=1.0.1.2
reachable via  VLAN101 distance=20 scope=40 target-scope=10
        bgp-as-path="100,100,100,100,2" bgp-origin=igp received-from=R101

12 ADb  dst-address=100.1.0.0/20 gateway=1.0.1.6 gateway-status=1.0.1.6
reachable via  VLAN12 distance=20 scope=40 target-scope=10
bgp-as-path="2,100" bgp-origin=incomplete received-from=R2

13  Db  dst-address=100.1.0.0/20 gateway=1.0.1.2 gateway-status=1.0.1.2
reachable via  VLAN101 distance=20 scope=40 target-scope=10
        bgp-as-path="100" bgp-origin=incomplete received-from=R101

 <output omited>
18 ADb  dst-address=100.7.0.0/20 gateway=1.0.1.6 gateway-status=1.0.1.6
reachable via  VLAN12 distance=20 scope=40 target-scope=10
bgp-as-path="2,700" bgp-origin=igp received-from=R2

19  Db  dst-address=100.7.0.0/20 gateway=1.0.1.2 gateway-status=1.0.1.2
reachable via  VLAN101 distance=20 scope=40 target-scope=10
        bgp-as-path="100,100,100,100,2,700" bgp-origin=igp received-from=R101
```

Nyní jsme vypnuli linku mezi R1 a R2 a po konvergenci opět provedli trasování:

```
[admin@R702] > tool traceroute 100.5.0.1
 # ADDRESS                                 RT1   RT2   RT3   STATUS
 1 2.0.7.1                                 1ms   1ms   1ms             < R2
 2 2.0.1.2                                 1ms   1ms   1ms             < R102
 3 100.1.1.1                               1ms   1ms   1ms             < R101
 4 1.0.1.1                                 1ms   1ms   1ms             < R1
 5 100.5.0.1                               1ms   1ms   1ms             < R501

[admin@R702] > tool traceroute 1.0.1.1
 # ADDRESS                                 RT1   RT2   RT3   STATUS
 1 2.0.7.1                                 1ms   1ms   1ms             < R2
 2 2.0.1.2                                 1ms   1ms   1ms             < R102
 3 100.1.1.1                               1ms   1ms   1ms             < R101
 4 1.0.1.1                                 1ms   1ms   1ms             < R1
```

Závěr

Prakticky jsme si vyzkoušeli práci s protokolem BGP. Při řešení jsme narazili na některé limity RouterOS v možnostech vzorů filtrů, které by ale v úkolu 3 řešilo efektivněji navržené adresní schéma. Špatné porozumění zadání úkolu 3 a pokus o jeho implementaci opačně nám taky ukázalo některé z možných řešení včetně jejich slabých stránek.

Literatura

[1] Blake: loopback interface. [online].[citováno 15. 10. 2013]. Dostupný z WWW: <http://forum.mikrotik.com/viewtopic.php?f=14&t=56329>.

[2] Mikrotik: Manual:BGP Best Path Selection Algorithm. [online].[citováno 15. 10. 2013]. Dostupný z WWW: <http://wiki.mikrotik.com/wiki/BGP_Best_Path_Selection_Algorithm>.

[3] Mikrotik: Manual:Routing/BGP. [online].[citováno 15. 10. 2013]. Dostupný z WWW: <http://wiki.mikrotik.com/wiki/Manual:Routing/BGP>.

[4] Mikrotik: Manual:Routing/Routing filters. [online].[citováno 15. 10. 2013]. Dostupný z WWW: <http://wiki.mikrotik.com/wiki/Manual:Routing/Routing_filters>.

[5] Rohleder, D.: Směrování mezi autonomními systémy. [online].[citováno 29. 10. 2013]. Dostupný z WWW: <https://is.muni.cz/auth/el/1433/podzim2013/PV177/um/43874409/BGP.pdf>.

[6] Zmijewski, E.: Reckless Driving on the Internet. [online].[citováno 29. 10. 2013]. Dostupný z WWW: <http://www.renesys.com/2009/02/the-flap-heard-around-the-world/>.