Border Gateway Protocol

BSCI Module 6

Autonómny systém

- Autonómny systém (AS) je skupina sietí a smerovačov, ktorá používajú spoločnú smerovaciu politiku a patria pod spoločnú administratívnu doménu
  - Smerovacia politika: spôsob výberu ciest do rôznych cieľov, filtrovanie smerovacích informácií, oznamovanie smerov...
  - Administratívna doména: dosah administratívnej právomoci správcu
- Vo vnútri AS môže pracovať jeden alebo niekoľko IGP, AS však ako celok patrí jednej organizácii
- Zvonku je AS vnímaný ako jedna nerozdelená entita
  - Všetky členské siete v AS sú v ňom z pohľadu iných AS priamo dostupné

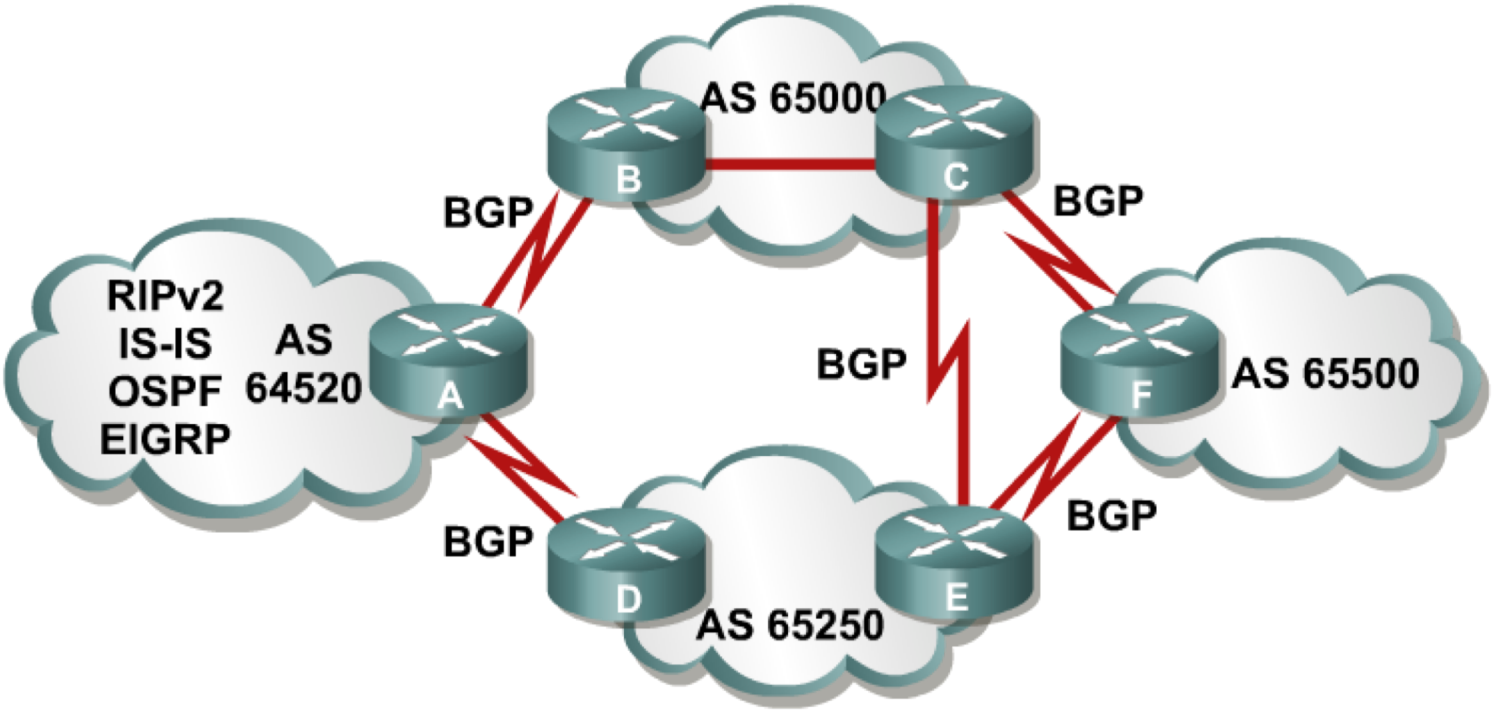
Autonómny systém

- AS sú číslované – ASN
  - Čísla AS rozdeľuje IANA na regionálne internetové registre, tie následne prideľujú AS jednotlivým žiadateľom
  - Regional Internet Registry:
    - ARIN (Severná Amerika)
    - RIPE NCC (Európa, Stredný Východ, Stredná Ázia)
    - APNIC (Ázia, Pacifická oblasť)
    - LACNIC (Latinská Amerika, Karibik)
    - AfriNIC (Afrika)
  - V súčasnosti sa používajú 2B čísla (0 – 65535)
  - RFC 4893 špecifikuje použitie 4B čísel (v dekadickom zápise 2B.2B)
  - Časť priestoru od 64512 po 65535 je vyhradená pre privátne ASN
- IANA nástojí na tom, aby organizácie, ktoré chcú mať vlastné číslo AS, avšak majú iba jediného ISP a zdieľajú jeho smerovacie politiky, zásadne používali privátne ASN
  - Privátne čísla AS sa objavujú len v sieti ISP a sú zamenené za ASN providera, keď sa prenášajú do iných AS

Autonómny systém

- Autonómne systémy sa tradične rozdeľujú na 3 druhy
- Single-homed
  - AS, ktorý má jediný hraničný router do ostatného sveta
  - Single-homed AS častokrát vôbec nepotrebujú EGP routing
- Multihomed
  - AS, ktorý má viacero hraničných routerov do ostatného sveta
  - Napriek tomu, že sa pripája viacerými výstupnými bodmi, nedovoľuje, aby cez neho tiekla cudzia prevádzka
- Transit
  - AS, ktorý má viacero hraničných routerov do ostatného sveta a slúži na prenos tranzitného trafficu

Smerovanie medzi AS



- Vo vnútri AS sa používajú IGP
- Medzi AS sa informácie vymieňajú pomocou BGP
- Smerovanie medzi AS
  - V ktorom AS sa nachádza cieľová sieť?
  - Akou cestou (cez ktoré medziľahlé AS) sa k nej dostanem?

Smerovanie medzi AS

- Smerovanie medzi AS sa zásadne líši od smerovania vo vnútri AS
- IGP protokoly:
  - Susedné smerovače sa navzájom objavujú automaticky
  - Snahou IGP je vymeniť si čo najkompletnejšiu informáciu o vnútornej topológii AS a jeho členských sieťach
  - Svet za hranicami AS je „zahmlený“ – nahradený sumárnymi smermi alebo využitím default route, bez topologickej predstavy
  - Metrika odráža výhodnosť trasy na základe počtu hopov, prenosovej rýchlosti, oneskorenia, záťaže, teda jej prenosové vlastnosti

Smerovanie medzi AS

- Smerovanie medzi AS sa zásadne líši od smerovania vo vnútri AS
- EGP protokoly:
  - Susedné smerovače musia pre vzájomnú komunikáciu byť explicitne nakonfigurované
  - EGP protokoly sa nezaujímajú o vnútornú topológiu AS, riešenie vnútornej dosiahnuteľnosti prenechávajú IGP
  - EGP protokoly sa zaujímajú o hraničné smerovače na okrajoch AS a o vzájomné prepojenie AS medzi sebou
  - Metrika sa skladá z parametrov, ktoré vyjadrujú pôvod siete a cestu cez tranzitné AS, jej lokálnu preferenciu – neodráža nutne fyzický charakter cesty, ale jej administratívne vlastnosti

Smerovanie medzi AS

- Smerovanie medzi AS musí byť zaručene bezslučkové
- Objem vymieňaných informácií je obrovský – desiatky až stovky megabajtov informácií obsiahnutých v smerovacích tabuľkách
- Výber ciest sa nerealizuje na základe ich prenosových charakteristík, ale na základe dohodnutých smerovacích politík a administratívnych rozhodnutí

BGP
Základné pojmy

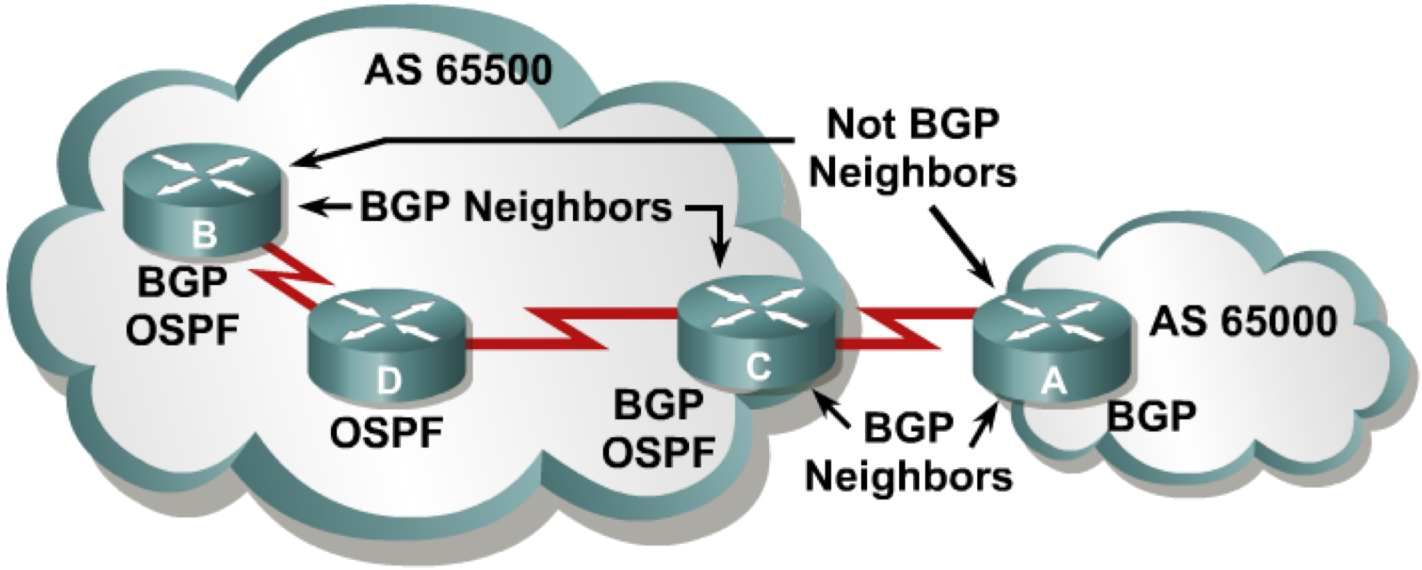
Border Gateway Protocol

- BGP je v súčasnosti prakticky jediný používaný smerovací protokol pre inter-AS smerovanie
- Aktuálna verzia: BGPv4 špecifikovaná v RFC 4271
  - Početné ďalšie RFC rozširujú schopnosti BGP o smerovanie multicastov, podporu MPLS a ďalšie
- BGP využíva TCP protokol pre komunikáciu, cieľový port 179
- BGP patrí do rodiny EGP protokolov, no pojem EGP je zároveň aj meno staršieho externého smerovacieho protokolu, s ktorým BGP nesúvisí

Tabuľky v BGP

- Tabuľka susedov – Neighbor table
  - Obsahuje zoznam a stav BGP susedov
- BGP tabuľka (forwarding database)
  - Obsahuje zoznam všetkých sietí získaných od každého suseda
  - K jednému cieľu môže obsahovať niekoľko záznamov
  - Ku každej ceste si eviduje jej BGP atribúty
- Smerovacia tabuľka – IP routing table
  - Zoznam najlepších ciest do cieľových sietí

BGP speaker, sused (peer, neighbor)

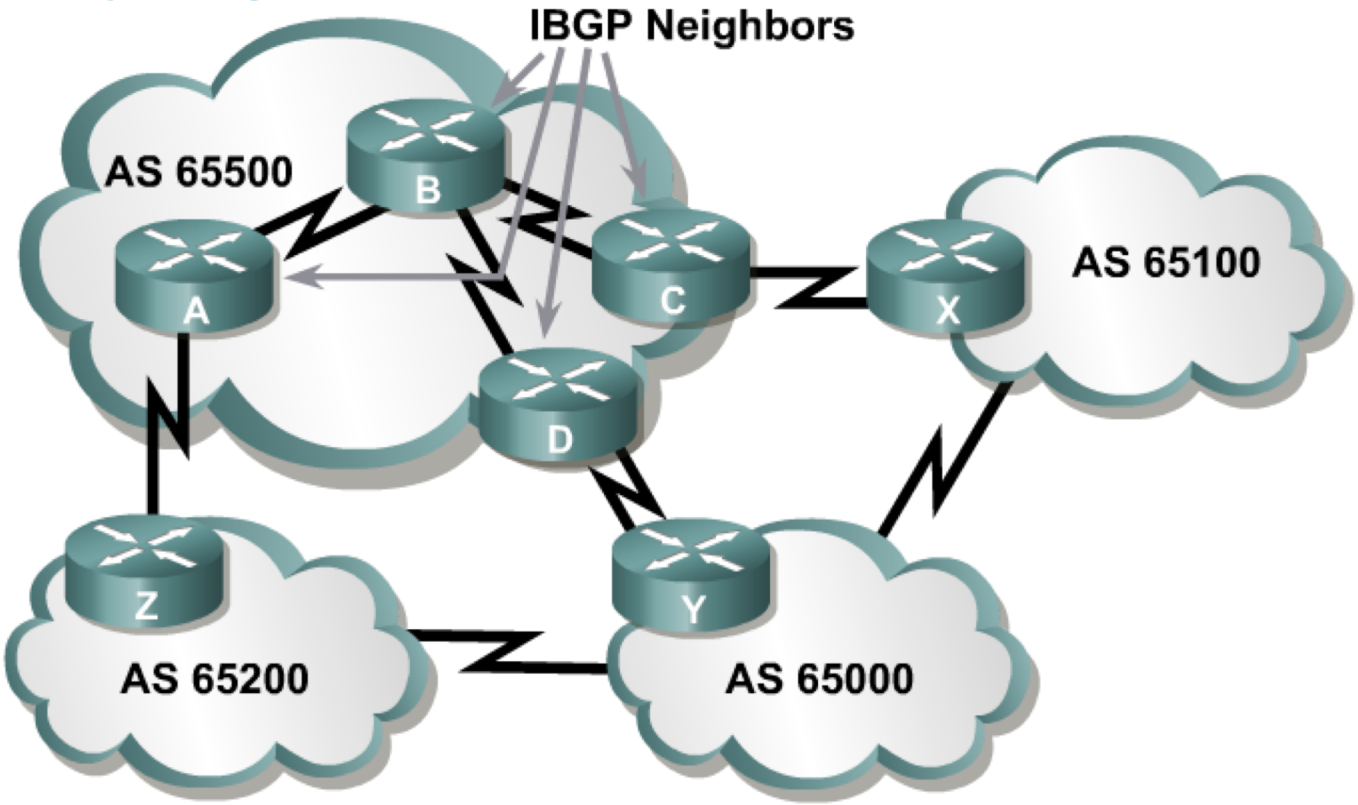


- BGP speaker je každý router, ktorý hovorí BGP protokolom
- BGP susedia (peers = neighbors) je pojem, ktorý označuje dvojicu vzájomne komunikujúcich BGP speakerov

Komunikácia v BGP

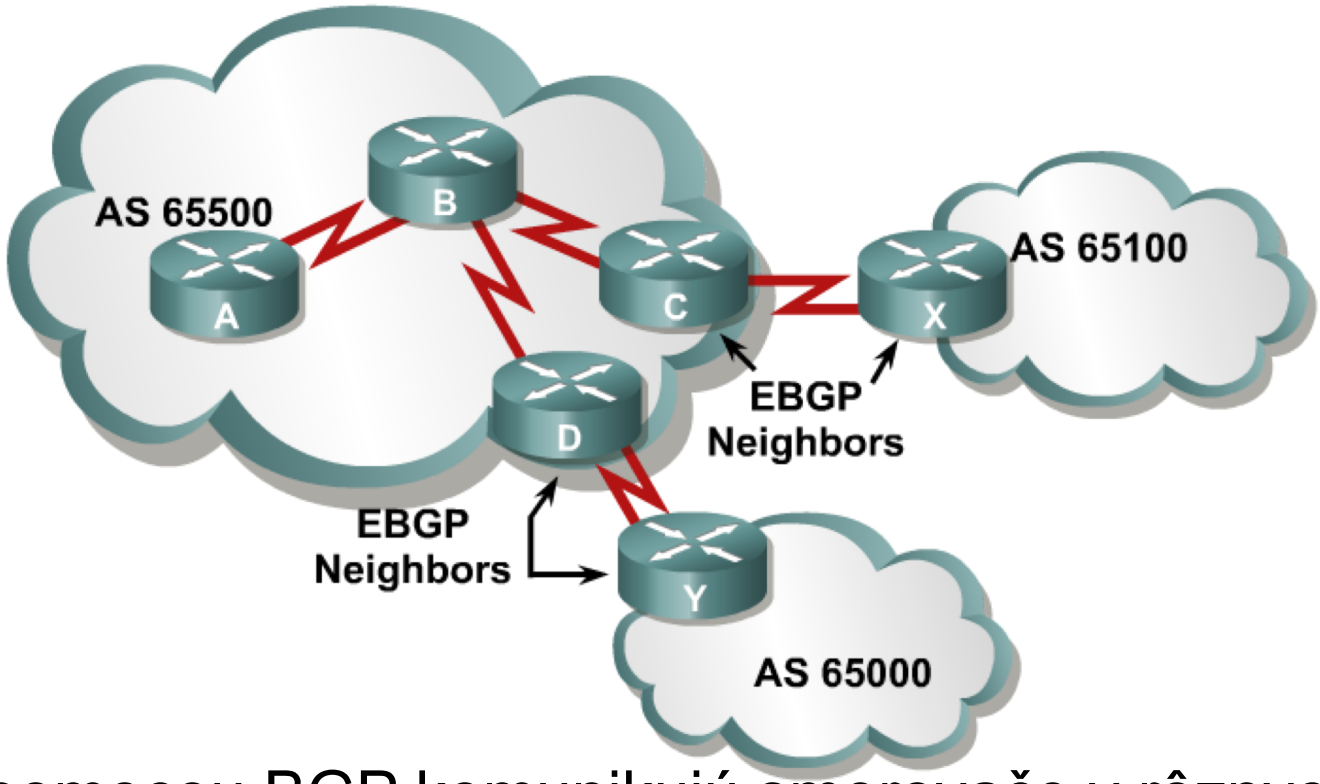
- Keď BGP susedia vytvoria spojenie, vzájomne sa synchronizujú – navzájom si oznámia všetky najlepšie smery zo svojich BGP tabuliek
- Po úvodnej synchronizácii sa posielajú iba inkrementálne aktualizácie – len zmeny (pridanie alebo odobranie smeru)
  - Inkrementálne aktualizácie sú efektívnejšie než prenosy úplných smerovacích tabuliek
  - Pri BGP sa jedná o obzvlášť zásadnú záležitosť, keďže veľkosť smerovacích tabuliek na chrbticových smerovačoch dosahuje rádovo desiatky až stovky MB

Interné BGP



- Ak pomocou BGP komunikujú smerovače v tom istom AS, hovoríme o IBGP (internal BGP)
- Susedia nemusia byť priamo spojení

Externé BGP



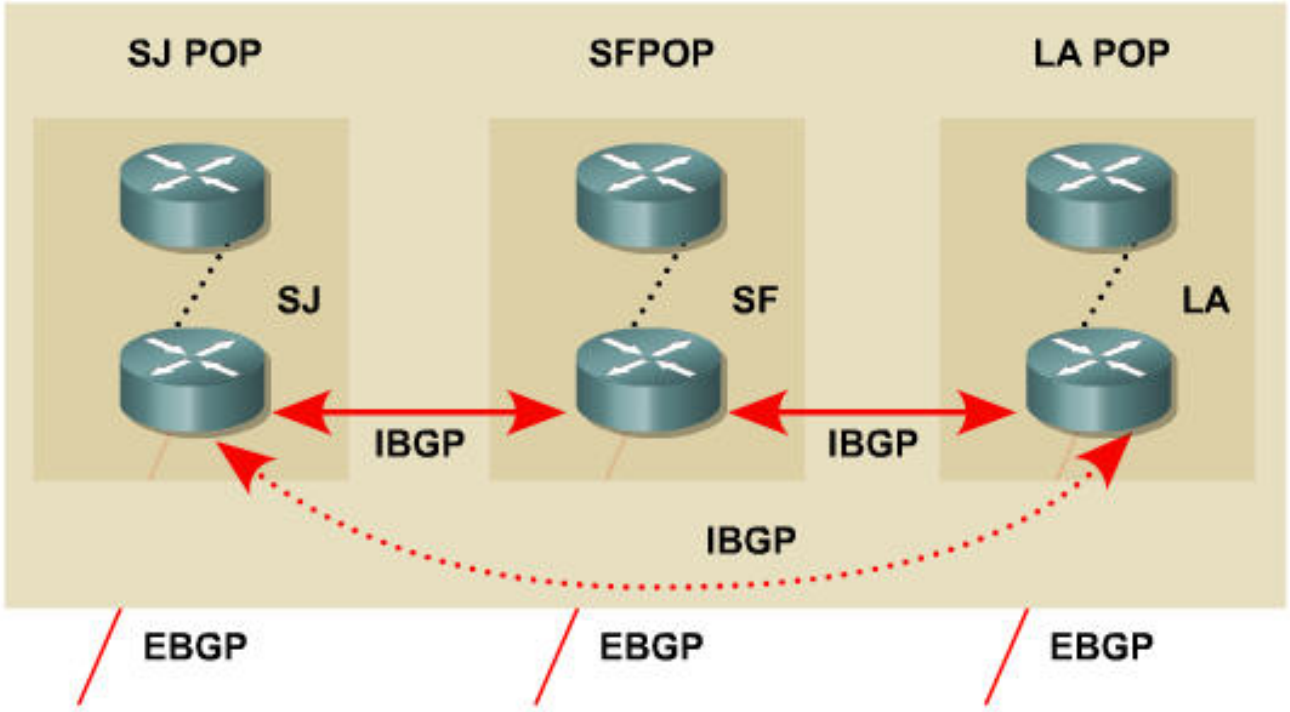
- Ak pomocou BGP komunikujú smerovače v rôznych AS, hovoríme o EBGP (external BGP)
- EBGP susedia musia byť za normálnych okolností priamo spojení

Prečo sa rozlišuje IBGP a EBGP?

- Správanie BGP sa líši v závislosti od toho, či komunikácia prebieha na úrovni IBGP alebo EBGP
- Zásadný rozdiel:
  - EBGP susedia si navzájom odovzdávajú BGP smery obvyklým spôsobom (čo som sa cez EBGP naučil, to cez ľubovoľné BGP odovzdám, a obrátene)
  - IBGP susedia si odovzdávajú informáciu len priamo, nikdy nie sprostredkovane
    - Ak sa router o nejakej ceste dozvie cez IBGP, neodovzdá túto informáciu nijakému ďalšiemu susedovi cez IBGP (čo som sa cez IBGP naučil, to si v IBGP nechám len pre seba, smiem to však povedať EBGP susedom)
    - Je to prísna, ale účinná ochrana pred vznikom smerovacích slučiek

Prečo sa rozlišuje IBGP a EBGP?

- Správanie sa IBGP si vynucuje závažný architekturálny rys pri implementácii BGP vo vnútri AS
  - Všetky BGP routery musia byť navzájom susedmi (konfiguračne, nie fyzicky)



Prečo sa rozlišuje IBGP a EBGP?

- Existujú ešte ďalšie rozdiely medzi IBGP a EBGP, ale o nich až na vhodnom mieste
- Pri veľkom počte BGP speakerov v AS je ich full meshing konfiguračne náročný a neefektívny
- Tento problém sa rieši v praxi dvomi spôsobmi:
  - Použitím tzv. route reflectorov
  - Rozdelením AS na podčasti – vnútorné AS a vytvorením tzv. konfederácií

Stavy komunikácie v BGP

BGP správy

- BGP definuje niekoľko jednoduchých typov správ
  - Open
    - Obsahuje číslo verzie, ASN, holdtime, BGP router ID
    - Posiela sa pri otváraní BGP spojenia medzi susedmi
  - Keepalive
    - Posiela sa periodicky na overenie, či sused žije, zodpovedajúco podľa dohodnutého holdtime
  - Update
    - Prenáša informáciu o jednej ceste (cestou sa rozumie postupnosť AS, táto cesta môže viesť k rozličným cieľovým sieťam)
    - Obsahuje atribúty cesty a tzv. Network Layer Reachability Information (NLRI – zoznam sietí dostupných touto cestou)
  - Notification
    - Posiela sa v prípade chyby a obsahuje jej popis
    - BGP spojenie sa po odoslaní Notification ukončí

Stavy v komunikácii so susedom v BGP

- Idle: Štartovací stav. Sused je definovaný, ale zatiaľ sme sa nepokúsili kontaktovať ho
- Connect: So susedom sme úspešne nadviazali TCP spojenie
- Open sent: Susedovi sme poslali správu OPEN obsahujúcu parametre spoločnej relácie
- Open confirm: Od suseda sme prijali súhlasnú OPEN správu, v ktorej sused vyjadruje súhlas pre peering s nami
  - Ak na odoslanú OPEN správu v stave Open Sent do 5 sekúnd nepríde potvrdenie ani zamietnutie, presúvame sa do stavu Active
- Established: Sme úspešní susedia, môže začať výmena smerovacích informácií

BGP stavy Established a Idle

- Idle: Ak sused zostáva v stave Idle, z nejakého dôvodu nie je možné vytvoriť s ním TCP spojenie
  - Existuje v našej smerovacej tabuľke cesta k tomuto susedovi?
  - Nie je v IP adrese suseda preklep?
- Established: Korektný stav pre suseda, kedy je možné vymieňať si s ním obsah smerovacích tabuliek
- Ak vo výpise príkazu **`show ip bgp summary`** sú v stĺpci „State/PfxRcd“ čísla, znamená to, že so susedom sme v stave Established. Číslo označuje počet smerov získaných od daného suseda.

Riešenie stavu BGP Active

- Active: Router poslal OPEN správu susedovi a čaká (zatiaľ neúspešne) na odpoveď alebo na vytvorenie TCP spojenia z druhej strany
- Stav môže oscilovať medzi Active a Idle.
- Táto situácia naznačuje na problém vo vzájomnej komunikácii medzi susedmi. Niektoré možné príčiny:
  1. Sused nemá cestu nazad k nám alebo my k nemu
  2. Nesprávne adresy susedov v konfigurácii BGP
  3. Sused nemá nás nakonfigurovaných ako svojho suseda
  4. Nezhoda v číslach AS
  5. Firewall blokuje komunikáciu medzi nami a susedom

Konfigurácia BGP

Spustenie BGP

```
Router(config)#
router bgp autonomous-system
```

- Príkaz definuje, v akom AS sa router nachádza, a otvorí konfiguračný kontext pre protokol BGP
- Na routeri môže bežať najviac jedna inštancia protokolu BGP
- Číslo AS v záhlaví príkazu sa porovná s číslami AS definovanými pri jednotlivých susedoch. Tak sa zistí, či je sused v tom istom alebo v inom AS, než sme my. Podľa toho sa so susedom vytvorí IBGP alebo EBGP peering
- BGP má svoje RouterID, ktoré sa vyberá rovnakým algoritmom ako pri OSPF resp. EIGRP

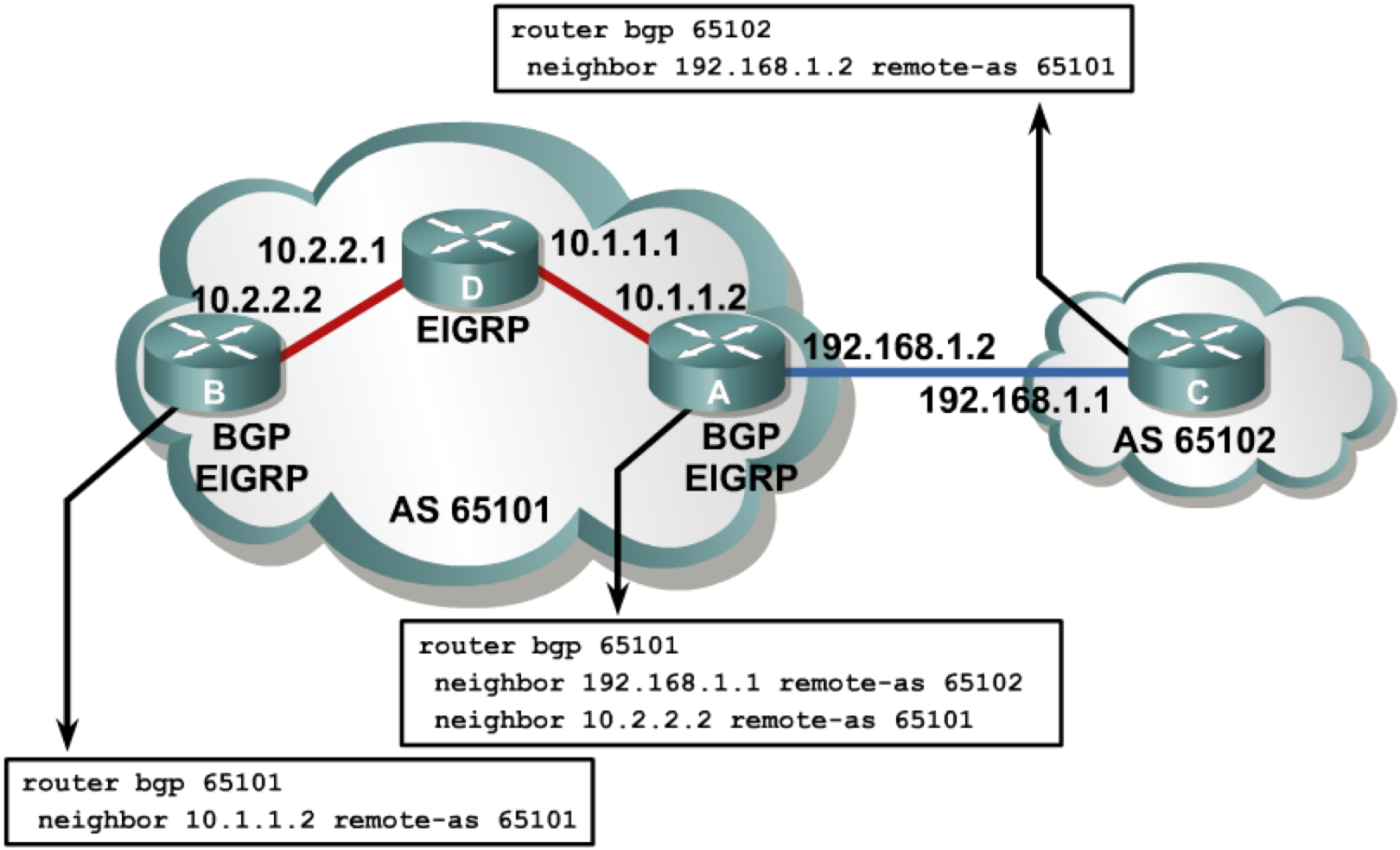
BGP príkaz `neighbor remote-as`

```
Router(config-router)#
```

```
neighbor {ip-address | peer-group-name}
  remote-as autonomous-system
```

- Príkaz **`neighbor`** definuje suseda a aktivuje peering s ním
- IP adresa špecifikuje cieľovú adresu, na ktorú sa budú posielať BGP pakety pre tohto suseda
- K danej IP adrese musí existovať v našej smerovacej tabuľke nejaká cesta
  - Pozor – default route sa na dosiahnutie suseda nikdy nepoužije!
- Argument **`remote-as`** hovorí, v akom AS sa nachádza príslušný sused
- Týmto príkazom sa definujú všetci susedia – externí aj interní
- Medzi dvojicou susedov musia IP adresy uvedené v príkaze **`neighbor`** vzájomne korešpondovať
  - **Zdrojová IP adresa BGP paketov od jedného suseda musí zodpovedať IP adrese v príkaze `neighbor` u druhého suseda, a obrátene**

Príklad: BGP príkaz `neighbor`



Sumarizácia – agregácia v BGP

- Do BGP sa siete vnášajú spravidla pomocou redistribúcie
- Sumarizácia (v BGP terminológii sa tento proces nazýva agregácia) sa realizuje príkazom

```
Router(config-router)#
aggregate-address network mask [summary-only]
```

- Parameter **summary-only** zabezpečí, že sa rozpošle iba agregovaná sieť, nie aj jej komponenty
  - Potrebné v prípadoch, že niekomu chceme poslať agregát, inému zasa špecifickejšie komponenty

Posielanie default route v BGP

- V BGP sa dá poslať default route vybranému susedovi príkazom

```
Router(config-router)#
neighbor {ip-address | peer-group-name} default-originate
```

- Default route na aktuálnom BGP routeri nemusí existovať (na chrbtici internetu neexistuje default route)

BGP `neighbor update-source` Command

```
Router(config-router)#
```

```
neighbor {ip-address | peer-group-name} update-source
interface-type interface-number
```

- Tento príkaz hovorí, aby zdrojová IP adresa BGP paketov odosielaných danému susedovi bola nastavená na IP adresu uvedeného rozhrania
- Najvhodnejšie je použiť loopback, ktorý je samozrejme potrebné ohlásiť v IGP, aby sused vedel odpovedať
- IP adresa v príkaze **`neighbor`** u *suseda* bude cieľovou adresou jeho BGP paketov, a teda musí byť nastavená na IP adresu *nášho* loopbacku
- Príkaz **`neighbor update-source`** sa zvykne používať len pri IBGP peeroch
- EBGP peer musí byť by default priamo pripojený – jeho loopback nie je priamo pripojený

Príkaz: BGP s použitím loopbackov


```
router bgp 65101
 neighbor 172.16.1.1 remote-as 65100
 neighbor 3.3.3.3 remote-as 65101
 neighbor 3.3.3.3 update-source Loopback0
!
router eigrp 1
network 10.0.0.0
network 2.0.0.0
```

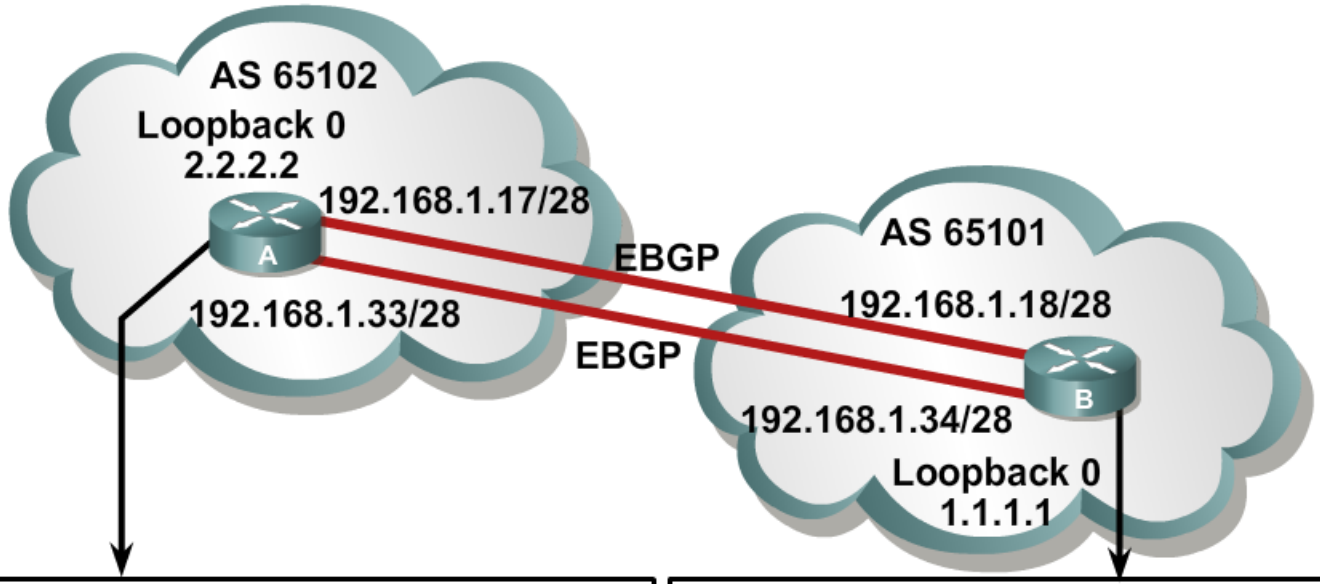
```
router bgp 65101
 neighbor 192.168.1.1 remote-as 65102
 neighbor 2.2.2.2 remote-as 65101
 neighbor 2.2.2.2 update-source Loopback0
!
router eigrp 1
network 10.0.0.0
network 3.0.0.0
```

BGP príkaz `neighbor ebgp-multihop`

```
Router(config-router)#
neighbor {ip-address | peer-group-name} ebgp-multihop [ttl]
```

- Tento príkaz umožňuje zvýšiť počet hopov medzi nami a EBGP peerom
- Počet hopov sa rieši elegantne využitím hodnoty TTL v IP paketoch
- Ak sa **`ttl`** neuvedie, použije sa hodnota 255

Example: ebgp-multihop Command




```
router bgp 65102
Neighbor 1.1.1.1 remote-as 65101
Neighbor 1.1.1.1 update-source Loopback 0
Neighbor 1.1.1.1 ebgp-multiho 2
!
ip route 1.1.1.1 255.255.255.255 192.168.1.18
ip route 1.1.1.1 255.255.255.255 192.168.1.34
```

```
router bgp 65101
Neighbor 2.2.2.2 remote-as 65102
Neighbor 2.2.2.2 remote-as 65101
Neighbor 2.2.2.2 update-source Loopback 0
Neighbor 2.2.2.2 ebgp-multiho 2
!
ip route 2.2.2.2 255.255.255.255 192.168.1.17
ip route 2.2.2.2 255.255.255.255 192.168.1.33
```

BGP is not designed to perform load balancing; paths are chosen because of policy, not based on bandwidth. BGP will choose only a single best path. Using the loopback addresses and the neighbor ebgp-multihop command as shown in this example allows load balancing, as well as redundancy, across the two paths between the autonomous systems.

BGP príkaz `neighbor shutdown`

```
Router(config-router)#
```

```
neighbor {ip-address | peer-group-name} shutdown
```

- Administratívne deaktivuje vybraného suseda
- Využíva sa pri údržbe konfigurácie a zmenách smerovacích politík

```
Router(config-router)#
```

```
no neighbor {ip-address | peer-group-name} shutdown
```

- Opätovne aktivuje suseda, ktorý bol deaktivovaný
  - Pozor – príkaz **`neighbor activate`** slúži na mierne iný účel: aktivácia suseda pre konkrétnu adresovú rodinu (nebudeme sa učiť)

BGP príkaz `network`

```
Router(config-router)#
network network-number [mask netmask]
```

- BGP zaradí sieť s presne danou adresou siete a presnou maskou do zoznamu sietí, ktoré oznámi svojim susedom
- Správanie príkazu je zásadne odlišné od jeho významu v IGP protokoloch
  - IGP prehľadá rozhrania – ak nejaké rozhranie má IP adresu z rozsahu adries daného týmto príkazom, potom do IGP bude zaradená celá sieť tohto rozhrania
  - BGP prehľadá smerovaciu tabuľku – musí v nej nájsť sieť so zhodným číslom a maskou (nezávisí na pôvode informácie)

Príklad: funkčný BGP peering

```
RouterA# show ip bgp summary
BGP router identifier 10.1.1.1, local AS number 65001
BGP table version is 124, main routing table version 124
9 network entries using 1053 bytes of memory
22 path entries using 1144 bytes of memory
12/5 BGP path/bestpath attribute entries using 1488 bytes of memory
6 BGP AS-PATH entries using 144 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 3829 total bytes of memory
BGP activity 58/49 prefixes, 72/50 paths, scan interval 60 secs

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd

10.1.0.2        4 65001      11      11      124    0    0 00:02:28        8
172.31.1.3      4 64998      21      18      124    0    0 00:01:13        6
172.31.11.4     4 64999      11      10      124    0    0 00:01:11        6
```

OPAKOVANIE z minula:

- BGP je path vector protocol – nejaké atribúty s cestou spoj.
- beží na TCP port=179 – pozor na firewally
- sh ip bgp summary
- rozlišujeme iBGP a eBGP podľa toho ako sú poprepájaní
- ak je AS rovnaké tak sa jedna o iBGP a ak rôzne eBGP
- podstatne rozdiely : eBGP susedia (externí) sú priamo pripojení lebo TTL byDefault=1, takže ak by sme potrebovali ist voči loopbacku tak potrebujete doplniť **neighbor ebgp multihop**
- pri eBGP spojeniach sa do interného atribútu AS_PATH dopĺňal jednoducho zoznam AS-iek cez ktoré potrebujem prejsť, aby som cieľovú sieť dosiahol, takže sa tam vždycky čosi pri kopírovalo
- prave tento mechanizmus mal jednu výhodu, že práve toto pri kopírovanie mi pomohlo pri detekcii slučiek

- ak som zbadal svoje vlastne AS viem, že je to slučka – taký update zahodím
- pri iBGP mala komplikácia, lebo tam sme stále v tom istom AS – konfiguračný full-mesh – neexistoval aby interný peer oznámil prefix ktorý sa naučil od iného peera – striktná podmienka, že susedia si môžu odovzdávať informácie vždy len priamo a nikdy nie sprostredkovane
- stavy: Active a Idle - kontrolovať ci mam routu k nemu a ci on ma routu ku mne
- netvork statement –> pridavam siete ... je to prenášač ...
- redistribúcie fungujú – hlavne to robíme ak nie je možné rozbehať v danom AS iBGP na všetkých routeroch – byDefault ale toto redistr nie je povolene -> bgp redistribute internal
- manualne objavovanie susedov cez neighbor comand cez IP a AS_number

Reset BGP spojení

Reštart BGP spojenia

- Po zmene politík (ACL pre filtrovanie, zmena hodnôt atribútov) sa zmeny prejavia len na nových prijatých alebo odoslaných prefixoch.
- Zmena na doposiaľ odoslaných a prijatých prefixoch sa sama od seba neuskutoční.
- Administrátor musí ručne vyvolať akciu, ktorá spôsobí aplikovanie nových politík
- Spôsoby na vynútenie aktualizácie prefixov:
  - Hard reset
  - Soft reset
  - Route refresh

Hard Reset BGP spojení

```
Router#
clear ip bgp *
```

- Zruší BGP spojenia so všetkými susedmi
- Celá BGP tabuľka sa zahodí
- Všetky spojenia prejdú do stavu Idle a informácie sa kompletne musia preniesť od susedov nanovo

```
Router#
clear ip bgp neighbor-address
```

- Zruší BGP spojenie s daným susedom
- Spojenie s týmto susedom prejde do stavu Idle a je potrebné nanovo sa s ním synchronizovať
- Menej drastické než **`clear ip bgp *`**

Outbound Soft Reset

```
Router#
clear ip bgp {* | neighbor-address} soft out
```

- Cesty od daného suseda sa ponechajú v našich tabuľkách
- Prepošleme však susedovi všetky BGP smery nanovo bez resetu spojenia
- Spojenie zostane v stave Established
- Táto voľba je ideálna pre situácie, keď sa mení outbound policy
- Príkaz **`soft out`** nemá efekt, ak sa mení inbound policy

Inbound Soft Reset

```
Router(config-router)#
neighbor [ip-address] soft-reconfiguration inbound
```

- Idea Inbound Soft Reset je pamätať si *všetky* informácie od BGP suseda *nefiltrované* v osobitnej databáze
  - Pri zmene inbound policy sa využije obsah tejto databázy namiesto opätovného stiahnutia dát od suseda
- Vyžaduje si dodatočnú konfiguráciu pre daného suseda
- Táto funkcionalita je pamäťovo náročná

```
Router#
clear ip bgp {* | neighbor-address} soft in
```

- Príkaz spôsobí zabudnutie naučených informácií od daného suseda a aplikuje aktuálnu inbound policy na údaje v osobitnej databáze

Route Refresh: Dynamický Inbound Soft Reset

```
Router#
clear ip bgp {* | neighbor-address} [soft in | in]
```

- Táto funkcionalita umožňuje nanovo si vyžiadať od suseda odoslanie informácií, pričom budú podrobené aktuálnej inbound policy
  - Pôvodná špecifikácia BGP to neumožňovala, pri Route Refresh sa jedná o dodatočné RFC 2918
- Smery odoslané danému susedovi nie sú dotknuté
- Nevyžaduje kópiu informácií v osobitnej databáze
- Spojenie zostáva v stave Established
- Podporované od verzie IOS 12.0(2)S and 12.0(6)T
- Ak všetci susedia podporujú Route Refresh, netreba písať **`soft`**

Príklad: debug ip bgp updates

```
RouterA#debug ip bgp updates
Mobile router debugging is on for address family: IPv4 Unicast
RouterA#clear ip bgp 10.1.0.2
<output omitted>
*Feb 24 11:06:41.309: %BGP-5-ADJCHANGE: neighbor 10.1.0.2 Up
*Feb 24 11:06:41.309: BGP(0): 10.1.0.2 send UPDATE (format)
10.1.1.0/24, next 10.1.0.1, metric 0, path Local
*Feb 24 11:06:41.309: BGP(0): 10.1.0.2 send UPDATE (prepend, chgflags:
0x0) 10.1.0.0/24, next 10.1.0.1, metric 0, path Local
*Feb 24 11:06:41.309: BGP(0): 10.1.0.2 NEXT_HOP part 1 net
10.97.97.0/24, next 172.31.11.4
*Feb 24 11:06:41.309: BGP(0): 10.1.0.2 send UPDATE (format)
10.97.97.0/24, next 172.31.11.4, metric 0, path 64999 64997
*Feb 24 11:06:41.309: BGP(0): 10.1.0.2 NEXT_HOP part 1 net
172.31.22.0/24, next 172.31.11.4
*Feb 24 11:06:41.309: BGP(0): 10.1.0.2 send UPDATE (format)
172.31.22.0/24, next 172.31.11.4, metric 0, path 64999
<output omitted>
*Feb 24 11:06:41.349: BGP(0): 10.1.0.2 rcvd UPDATE w/ attr: nexthop
10.1.0.2, origin i, localpref 100, metric 0
*Feb 24 11:06:41.349: BGP(0): 10.1.0.2 rcvd 10.1.2.0/24
*Feb 24 11:06:41.349: BGP(0): 10.1.0.2 rcvd 10.1.0.0/24
```

Atribúty v BGP

BGP atribúty

- Atribút je vlastnosť smeru, ktorá charakterizuje nejakú jeho vlastnosť, a na základe ktorej si BGP vyberá najvhodnejšiu cestu
- BGP pozná mnoho atribútov, zaraďujeme do 4 základných druhov:
  - Well-known mandatory: atribút, ktorý musí povinne podporovať každá implementácia BGP (well-known) a ktorý musí byť prítomný pri každom popise nejakej cesty (mandatory)
  - Well-known discretionary: atribút, ktorý musí povinne podporovať každá implementácia BGP (well-known), ale ktorý nemusí byť prítomný v popise cesty (discretionary)
  - Optional transitive: atribút, ktorý nemusí podporovať každá implementácia BGP (optional), avšak musí ho preposlať susedom napriek tomu, že mu nerozumie (transitive)
  - Optional nontransitive: atribút, ktorý nemusí podporovať každá implementácia BGP (optional), a ak mu nerozumie, nesmie ho preposlať susedom (nontransitive)
- Všetky well-known atribúty majú tranzitívnu povahu

BGP atribúty

- Well-known mandatory: (pomôcka: MONA)
  - ORIGIN: pôvod smeru (iBGP, EGP, neznámy)
  - NEXT_HOP: next hop pre daný smer
  - AS_PATH: zoznam AS na ceste k danému smeru
- Well-known discretionary: (pomôcka: DALA)
  - ATOMIC_AGGREGATE: info o nerozdeliteľnej sieti
  - LOCAL_PREF: vyjadrenie preferencie
- Optional Transitive:
  - AGGREGATOR: zdroj agregovanej informácie
- Optional Nontransitive:
  - MULTI_EXIT_DISC: odporúčanie pre výstup z jedného AS do druhého

BGP atribúty

- Okrem toho Cisco zavádza vlastný atribút **Weight**, ktorý je lokálny pre daný router a nepreposiela sa nikomu
- BGP atribúty možno nastavovať alebo kontrolovať pomocou route-map konštruktov
- Každá kontrola, nastavovanie a filtrovanie sa deje pomocou príkazu **`neighbor`** pre každého suseda alebo peer group nezávisle
  - Na rozdiel od IGP, kde sa filtrovanie robí spravidla hromadne, nezávisle od odosielateľa informácie, v BGP sa zasa spravidla realizuje filtrovanie pre každého suseda zvlášť
- BGP má stanovené poradie, v akom vyhodnocuje jednotlivé atribúty pre výber najlepšej cesty

BGP atribút AS_PATH

- Atribút AS_PATH predstavuje zoznam AS, cez ktoré treba prejsť, než sa dostaneme do cieľovej siete
- Keď informácia o nejakej sieti prechádza medzi eBGP susedmi, odosielateľ pripojí číslo svojho AS na začiatok tohto zoznamu (prepending)
- AS_PATH slúži ako prostriedok na eliminovanie smerovacej slučky
  - Router v istom AS nemôže akceptovať cez eBGP informáciu, ktorá už toto číslo AS v atribúte AS_PATH obsahuje
- pri filtrovaní AS_PATH je možné využívať: regulárne výrazy

BGP atribút NEXT_HOP

- Atribút NEXT_HOP je IP adresa nasledujúceho hopu
  - BGP vidí cestu ako poradie AS systémov, nie ako poradie routerov
  - NEXT_HOP vyjadruje IP adresu hraničného routera v nasledujúcom AS
- Z toho logicky vyplýva správanie sa atribútu NEXT_HOP
  - Pri eBGP (medzi AS) bude nastavený odosielateľom na IP adresu odosielateľa danej informácie (jeho update-source)
    - Za istých okolností na multiaccess sieťach môže byť hodnota NEXT_HOP odlišná, vždy však bude na spoločnej IP sieti s eBGP peerom
  - Prechodom cez iBGP (vo vnútri AS) sa jeho hodnota nebude meniť
- To má závažný dôsledok
  - Daný AS musí poznať cestu k príslušnému hraničnému routeru v inom AS, inak nebude možné vložiť tento BGP smer do smerovacej tabuľky

BGP atribút NEXT_HOP

- Táto vlastnosť môže komplikovať situáciu pri NBMA sieťach alebo v topológiách, kde nie je všetkým BGP speakerom známa cesta k príslušnému hraničnému routeru
- Riešenie: **`neighbor A.B.C.D next-hop-self`**
  - V smeroch získaných z eBGP a preposielaných na daného iBGP suseda A.B.C.D zmeníme NEXT_HOP na seba
  - Príkaz neplatí pri route reflectoroch – v takom prípade sa dá NEXT_HOP modifikovať pomocou route-map

BGP atribúty ORIGIN a WEIGHT

- Atribút ORIGIN vyjadruje pôvod cesty
  - „i“ – sieť má pôvod v súčasnom AS (vnesená do BGP príkazom network)
  - „e“ – sieť je redistribuovaná z historického EGP protokolu
  - „?“ – pôvod siete nie je známy (redistribúcia)
- Atribút WEIGHT vyjadruje „mikrolokálnu“ preferenciu smeru
  - Cisco proprietárny atribút lokálny pre router, nepreposiela sa
  - Čím vyššia hodnota, tým viac preferovaný smer
  - Smery, ktoré do BGP vnášame my, majú WEIGHT 32768
  - Smery, ktoré sme sa cez BGP naučili, majú WEIGHT 0

BGP atribút LOCAL_PREF a MED

- Atribút LOCAL_PREF označuje preferenciu cesty
  - Atribút LOCAL_PREF sa odovzdáva len cez iBGP, neprechádza cez eBGP, je teda uzatvorený vo vnútri AS
  - Čím vyššia hodnota, tým lepšie
  - Implicitná hodnota je 100
- MED je indikácia pre susedný AS, ktorú z viacerých možných ciest do nášho AS má použiť
  - MED sa iniciálne prenesie z nášho AS cez eBGP do susedného AS a v ňom sa rozšíri, avšak z neho už nevychádza ďalej (z neho sa ďalej preposiela s hodnotou 0)
  - MED sa zvykne nazývať aj „metrika“ a v tomto zmysle platí: čím menšia, tým lepšie
  - Štandardne sa MED porovnáva medzi rôznymi cestami do toho istého cieľa len vtedy, ak tieto cesty prišli z rovnakého susedného AS

Kontrola/nastavovanie jednotlivých atribútov

- Hodnota atribútov a ich manipulácia sa spravidla realizuje pomocou route-map
- NEXT_HOP
  - match ip address {1-99 | 1300-1999 | MENO | prefix-list MENO}
  - set ip next-hop A.B.C.D
- ORIGIN
  - match route-type local
  - set origin {igp | egp | incomplete}
- AS_PATH
  - match as-path 1-500 (číslo označuje tzv. AS path list)
  - set as-path prepend N N N ...

Kontrola/nastavovanie jednotlivých atribútov

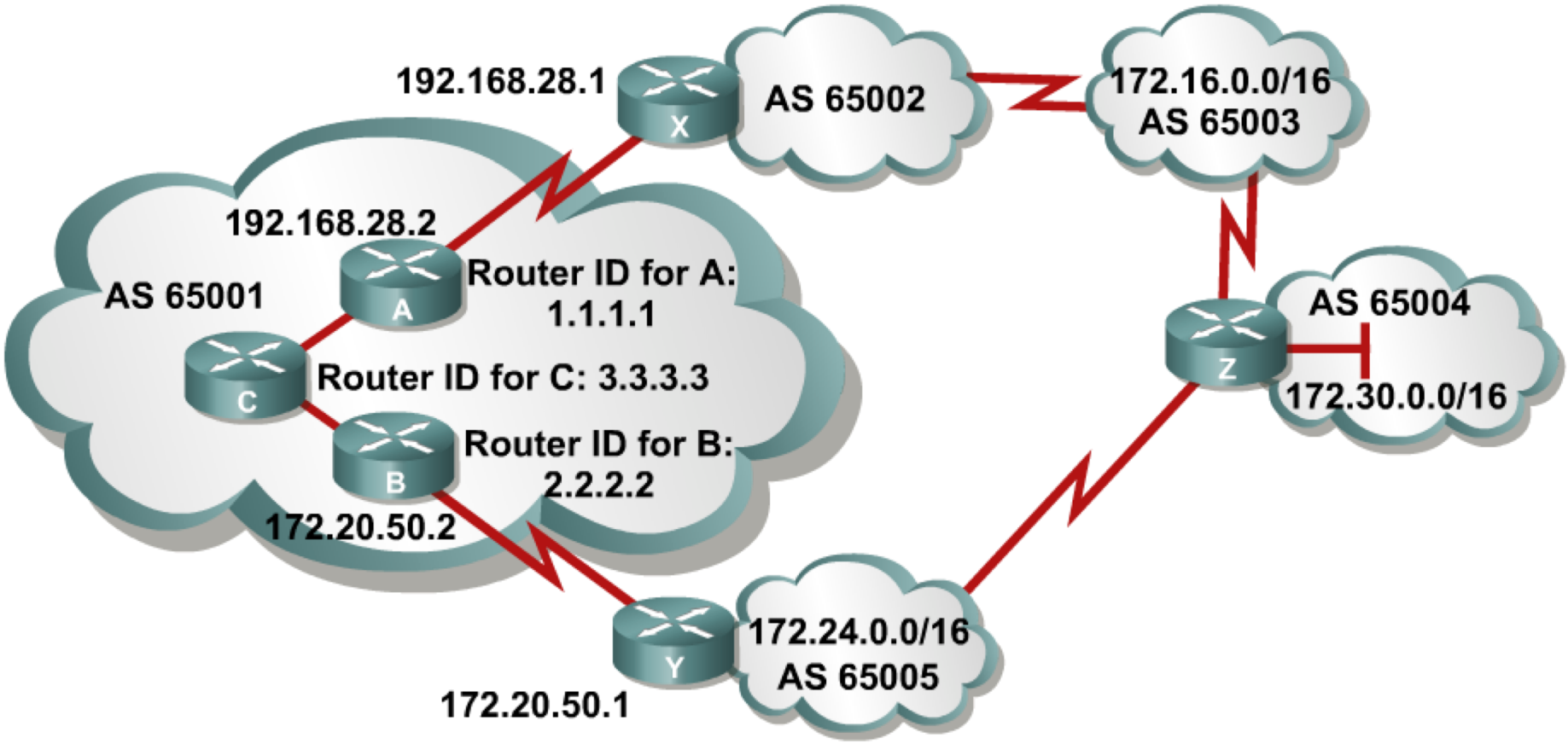
- LOCAL_PREF:
  - match local-preference N
  - set local-preference N
- MULTI_EXIT_DISC:
  - match metric N
  - set metric N
- WEIGHT:
  - match neexistuje (atribút sa neposiela)
  - set weight N

Zmena štandardnej LOCAL_PREF hodnoty

```
Router(config-router)#
bgp default local-preference value
```

- Zmení hodnotu LOCAL_PREF zo štandardnej hodnoty 100 na definovanú hodnotu
- Všetky smery ohlásené iBGP susedom budú mať nastavenú danú hodnotu LOCAL_PREF

Použitie LOCAL_PREF



Použitie LOCAL_PREF: Route Map na smerovači A

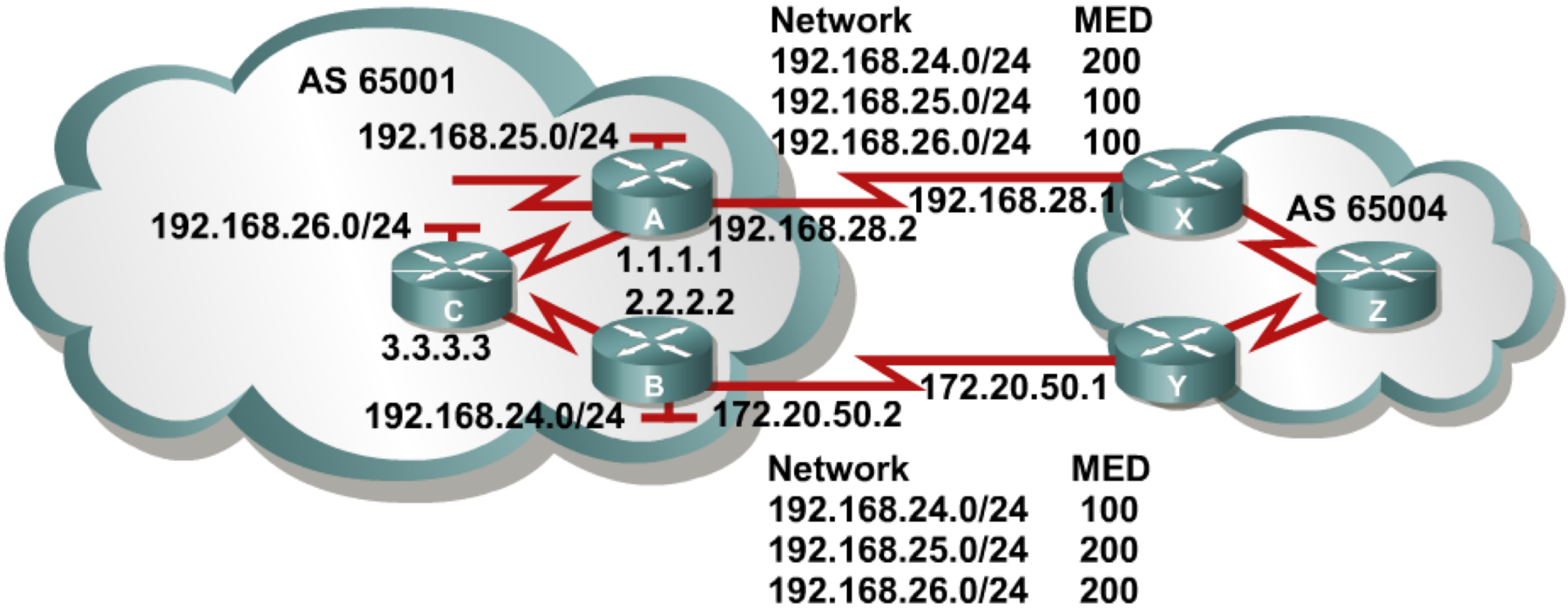
```
router bgp 65001
neighbor 2.2.2.2 remote-as 65001
neighbor 3.3.3.3 remote-as 65001
neighbor 2.2.2.2 remote-as 65001 update-source loopback0
neighbor 3.3.3.3 remote-as 65001 update-source loopback0
neighbor 192.168.28.1 remote-as 65002
neighbor 192.168.28.1 route-map local_pref in
!
access-list 65 permit 172.30.0.0 0.0.255.255
!
route-map local_pref permit 10
match ip address 65
set local-preference 400
!
route-map local_pref permit 20
```

Zmena štandardnej hodnoty MED

```
Router(config-router)#
default-metric number
```

- Zmení hodnotu MED zo štandardnej hodnoty 0 na definovanú hodnotu
- Všetky smery ohlásené eBGP susedom budú mať nastavenú hodnotu MED (metriku) na definovanú hodnotu

Použitie MED



Použitie MED: Route Map pre Router A

```
router bgp 65001
neighbor 2.2.2.2 remote-as 65001
neighbor 3.3.3.3 remote-as 65001
neighbor 2.2.2.2 update-source loopback0
neighbor 3.3.3.3 update-source loopback0
neighbor 192.168.28.1 remote-as 65004
neighbor 192.168.28.1 route-map med_65004 out
!
access-list 66 permit 192.168.25.0.0 0.0.0.255
access-list 66 permit 192.168.26.0.0 0.0.0.255
!
route-map med_65004 permit 10
match ip address 66
set metric 100
!
route-map med_65004 permit 100
set metric 200
```

Použitie MED: Route Map pre Router B

```
router bgp 65001
neighbor 1.1.1.1 remote-as 65001
neighbor 3.3.3.3 remote-as 65001
neighbor 1.1.1.1 update-source loopback0
neighbor 3.3.3.3 update-source loopback0
neighbor 172.20.50.1 remote-as 65004
neighbor 172.20.50.1 route-map med_65004 out
!
access-list 66 permit 192.168.24.0.0 0.0.0.255
!
route-map med_65004 permit 10
match ip address 66
set metric 100
!
route-map med_65004 permit 100
set metric 200
```

Použitie MED: čo vidí Router Z

```
RouterZ# show ip bgp
BGP table version is 7, local router ID is 122.30.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
r RIB-failure, S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*>i192.168.24.0     172.20.50.2              100    100        0 65001 i
* i                 192.168.28.2             200    100        0 65001 i
* i192.168.25.0     172.20.50.2              200    100        0 65001 i
*>i                 192.168.28.2             100    100        0 65001 i
* i192.168.26.0     172.20.50.2              200    100        0 65001 i
*>i                 192.168.28.2             100    100        0 65001 i
```

Poradie vyhodnocovania atribútov

Uvažujú sa len (synchronizované) smery bez AS slučiek a dosiahnuteľným next hop routerom:

- Cesta s najvyšším atribútom WEIGHT (lokálny pre router)
- Cesta s najvyšším atribútom LOCAL_PREF (globálny v rámci AS)
- Cesta, ktorú sme do BGP my sami vniesli (next hop = 0.0.0.0)
- Cesta s najmenším počtom AS v zozname AS_PATH
- Cesta s najnižším kódom pôvodu (IGP < EGP < incomplete)
- Cesta s najnižším MED
- Preferujú sa cesty naučené cez EBGP voči cestám naučeným cez IBGP
- Preferujú sa cesty cez najbližšieho IGP suseda
- Pre EBGP sa preferujú najstaršie (prvé naučené) cesty
- Preferujú sa cesty od suseda s najnižším BGP router ID
- Preferujú sa cesty od suseda s najnižšou IP adresou

BGP Peer Groups, Router Reflectors & Autentifikácia

Používanie Peer Group

```
Router(config-router)#
neighbor peer-group-name peer-group
```

- Týmto príkazom sa vytvorí peer group

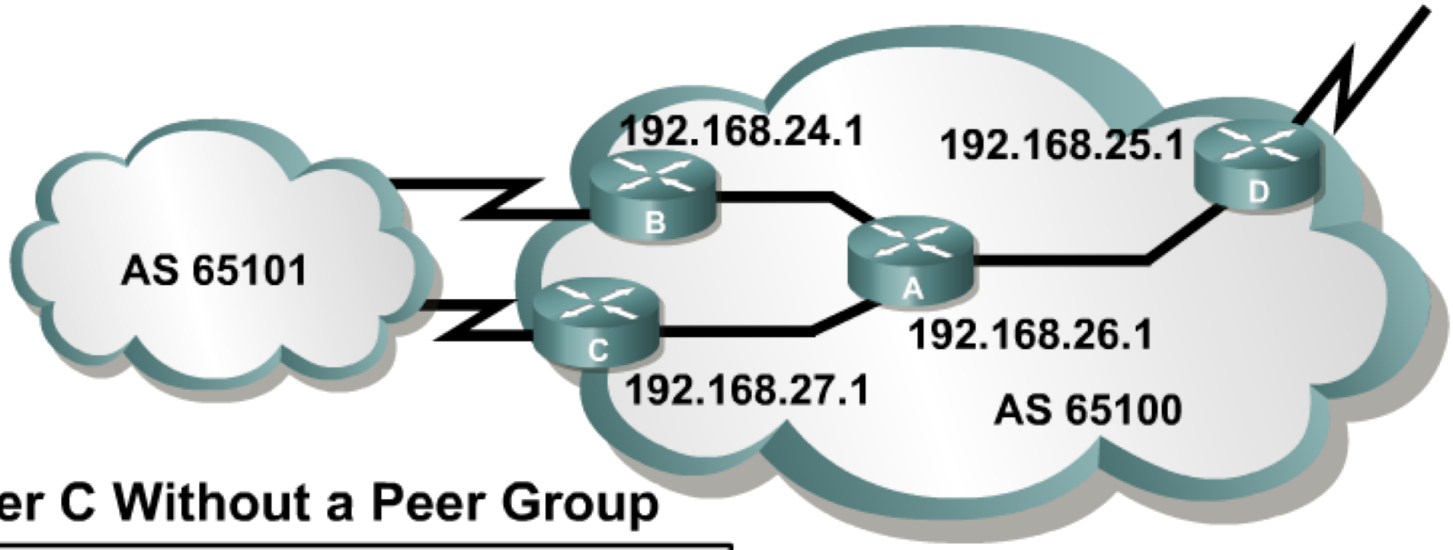
```
Router(config-router)#
neighbor ip-address peer-group peer-group-name
```

- Tento príkaz zaradí suseda do vybranej peer group
- Peer groups je možné s výhodou použiť, ak máme skupinu susedov, ktorí majú spoločnú „outbound policy“ – politiku, ktorá filtruje prenos informácie od nás k nim
- Členovia peer group môžu mať rozličné „inbound policies“ – vstupné politiky, ktoré filtrujú prenos informácie od nich k nám

Používanie Peer Group

- Peer Groups sú veľmi výhodné, pretože
  - Aktualizácie sú vygenerované pre celú grupu iba raz
  - Zjednodušuje sa konfigurácia – všetky filtre, route-mapy a podobné konštrukcie sa aplikujú na grupu, netreba na jednotlivých členov
  - Šetrí sa procesorový čas a pamäť, pretože smerovacia tabuľka sa pre peer group kontroluje len raz, takisto aktualizácie sa generujú len raz a replikujú sa
- Peer Group zjednodušujú konfiguráciu, avšak stále je zachovaná požiadavka na full-mesh peerov

Príklad: Použitie Peer Group



Router C Without a Peer Group

```
router bgp 65100
 neighbor 192.168.24.1 remote-as 65100
 neighbor 192.168.24.1 update-source Loopback 0
 neighbor 192.168.24.1 next-hop-self
 neighbor 192.168.24.1 distribute-list 20 out
 neighbor 192.168.25.1 remote-as 65100
 neighbor 192.168.25.1 update-source Loopback 0
 neighbor 192.168.25.1 next-hop-self
 neighbor 192.168.25.1 distribute-list 20 out
 neighbor 192.168.26.1 remote-as 65100
 neighbor 192.168.26.1 update-source Loopback 0
 neighbor 192.168.26.1 next-hop-self
 neighbor 192.168.26.1 distribute-list 20 out
```

Router C Using a Peer Group

```
router bgp 65100
 neighbor internal peer-group
 neighbor internal remote-as 65100
 neighbor internal update-source Loopback 0
 neighbor internal next-hop-self
 neighbor internal distribute-list 20 out
 neighbor 192.168.24.1 peer-group internal
 neighbor 192.168.25.1 peer-group internal
 neighbor 192.168.26.1 peer-group internal
```

Route reflector

- Route reflector (RR) je BGP router, ktorý obchádza pravidlo, že cez iBGP sa nesmie odovzdať informácia, ktorá bola práve cez iBGP naučená
- Pomocou RR je možné zásadne zjednodušiť konfiguráciu BGP speakerov v AS: z full meshed logickej topológie vznikne logická hub-and-spoke
  - RR si nakonfiguruje všetky ostatné routery ako svojich susedov a pridá si ku každému z týchto susedov riadok

```
Router(config-router)#
neighbor {ip-address | pg-name} route-reflector-client
```

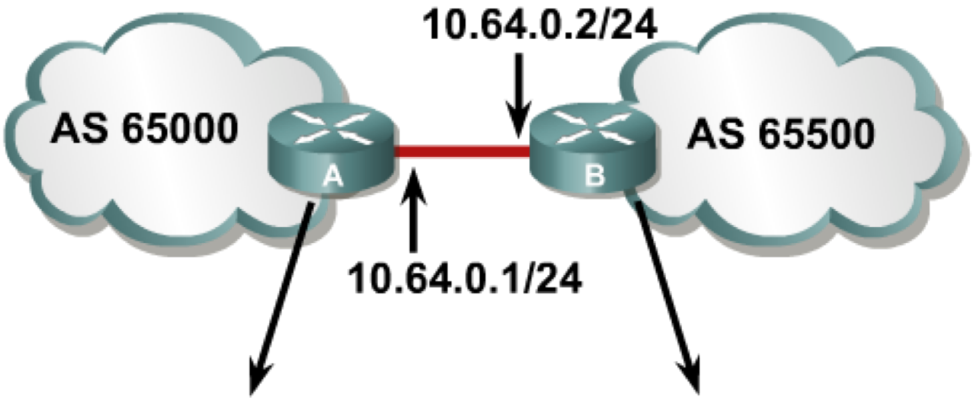
  - Všetky ostatné BGP routery si nakonfigurujú RR router ako svojho suseda

Autentifikácia v BGP

```
Router(config-router)#
```

```
neighbor {ip-address | peer-group-name} password string
```

- BGP používa MD5 autentifikáciu
- Pre každého suseda sa môže definovať nezávislý kľúč (heslo)




```
router bgp 65500
 neighbor 10.64.0.2 remote-as 65500
 neighbor 10.64.0.2 password v61ne0qkel33&
```

```
router bgp 65500
 neighbor 10.64.0.1 remote-as 65000
 neighbor 10.64.0.1 password v61ne0qkel33&
```