

Laborator 4

Rutarea dinamică inter-AS folosind BGP; *Policy routing*

Partea 1: prezentarea BGP

1.1 Introducere

Protocoalele de rutare dinamică discutate pînă acum (RIP, OSPF) au toate în comun rutarea pe baza unei metrici cu caracter *tehnic*: numărul de hopuri, banda disponibilă, etc. Adică, procesul de rutare determină singur ruta optimă între oricare 2 hosturi din domeniu pe baza acestei metrici.

Chiar presupunînd că un astfel de protocol este scalabil la nivelul întregului Internet (de fapt nu este), se observă că aceasta abordare introduce o problemă majoră.

Ca un exemplu, să presupunem că un client este conectat la 2 furnizori de Internet ISP1 și ISP2, pentru redundanță, ca în figura 1:

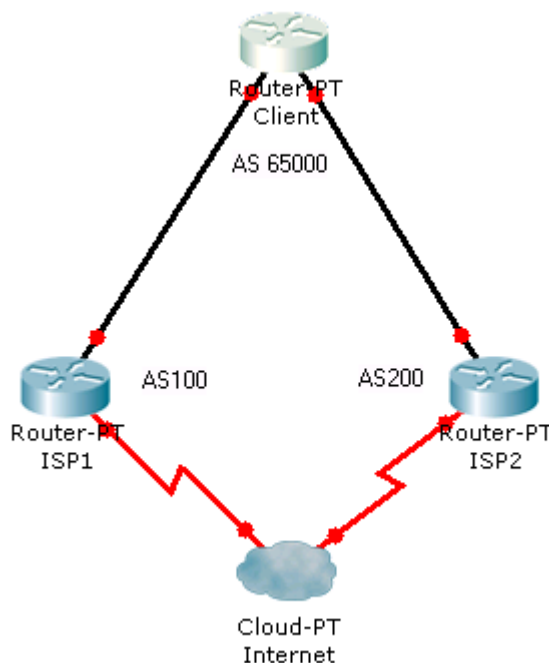


Fig. 1 Un client conectat la 2 ISP

Presupunînd că cele 3 domenii (client C, ISP1 și ISP2) au activat un protocol de rutare, de exemplu OSPF, atunci nu numai C va putea trimite trafic către ISP1 și ISP2, dar și ISP1 va putea comunica cu ISP2 prin C. Presupunînd în continuare că legătura la Internet a ISP2 este mult mai rapidă decît a ISP1, OSPF din ISP1 ar putea lua decizia ca, pentru destinațiile din Internet, traficul să nu mai fie rutat prin legătura Internet a lui ISP1, ci prin legătura cu C, și mai departe prin ISP2! Altfel zis, ISP2 ar ajunge să suporte tot traficul Internet al C, dar și al ISP1. Avînd în vedere că ISP1 și ISP2 plătesc pentru conectarea lor la Internet, s-ar ajunge ca ISP1 să nu mai plătească nimic și ISP2 să plătească aproximativ dublu.

Se observă în acest caz că metrica tehnică nu mai este de dorit. Între ISP-uri trebuie implementat un protocol de rutare bazat nu pe capabilitati tehnice, ci pe reguli politice, configurate de administratori. În acest exemplu, ISP1 și ISP2 vor accepta trafic Internet de la C, dar nu de la celălalt ISP. Singurul trafic între ISP-uri acceptat prin legătura cu C ar putea fi atunci cînd sursa este într-unul din ISP-uri și destinația este în celălalt ISP. Dar, și aceasta situație este acceptabilă numai dacă administratorii lui C sînt de acord cu acest lucru (practic, legăturile C cu cele 2 ISP-uri sînt

folosite în interesul ISP-urilor și nu al sau propriu!). Dacă nu, atunci traficul între o sursă din ISP1 și o destinație din ISP2 trebuie să urmeze o cale ocolitoare prin Internet, deși există o cale mai scurtă și mai bună prin C!

Reamintim conceptul de *Autonomous System* (AS): este o rețea aflată sub o administrare comună, indiferent cât de mare este această rețea. În interiorul unui AS, traficul între nodurile proprii evident nu se plătește, și se folosește un protocol de rutare dinamică din cele discutate. Aceste protocoale se numesc IGP (*Interior Gateway Protocols*). Între AS-uri însă, se implementează un protocol care să permită numai traficul definit prin reguli, care de obicei se stabilesc prin înțelegeri politice sau financiare. Acest protocol face parte din familia EGP (*Exterior Gateway Protocols*). Protocolul din aceasta familie folosit în Internet este BGP4 (*Border Gateway Protocol, version 4*). În figura de mai jos este ilustrat un exemplu de conectare a mai multor AS prin BGP:

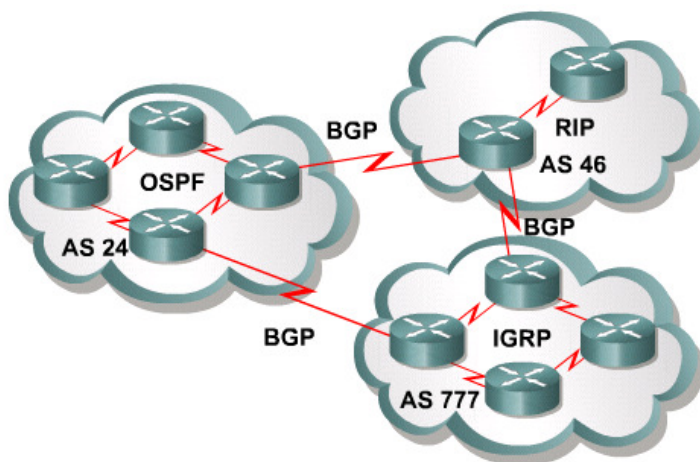


Fig. 2 Exemplu de rețea multi-AS (sursa: [1])

Protocolul BGP4 este “motorul” Internetului curent; în esență, Internetul este o sumă de AS-uri care schimbă rute prin BGP, în timp ce în interiorul AS-urilor se folosesc diferite protocoale de rutare interne astfel încât (1) se “ascund” toate rețelele interne care nu trebuiesc cunoscute în exterior, și (b) se sumarizează rețelele care trebuiesc cunoscute, în cât mai puține rețele de mască cât mai scurtă.

Pe baza exemplului precedent, extindem definiția AS ca fiind o rețea aflată sub administrare comună și care are legături cu mai mult de un alt AS. O organizație având o rețea oricât de mare, dar conectată printr-un singur ISP la Internet nu are nici una din problemele din exemplu și de aceea nu intră în categoria AS. Rețelele acestei organizații vor fi văzute din Internet ca făcând parte din AS-ul ISP-ului lor. O astfel de rețea cu o singură ieșire se mai numește *single-homed network* sau *stub network*. O rețea cu mai multe ieșiri, care satisface definiția AS, se cheama *multi-homed*.

Fiecare AS cere și primește un număr de AS unic. Datorită restricției precedente, rezultă că numărul AS-urilor din Internet este mult mai mic decât numărul adreselor IP (practic, numai operatorii de tip ISP și un număr redus de organizații non-ISP), s-a ales ca numărul de AS să fie pe 16 biți, deci maxim 65535 AS-uri în tot Internetul. AS-urile începând cu 64512 sînt private (folosite pentru teste, oarecum similare cu adresele IP nerutabile 192.168. etc.).

BGP poate fi folosit în interiorul unui AS, în care caz se numește Interior BGP (IBGP), sau între AS-uri, în care caz se numește EBGP. De notat că IBGP și EBGP nu sînt 2 protocoale diferite; BGP va detecta automat, pe baza numărului AS a vecinului, dacă rulează în mod interior sau exterior.

2 rutere care comunică între ele prin BGP realizează o conexiune TCP pe portul 179. Prin comparație, OSPF folosește doar IP, iar RIP trimite updates folosind UDP pe portul 520.

La stabilirea comunicației între 2 rutere BGP, fiecare informează pe celălalt asupra *rețelelor disponibile* (rețelele pentru care face rutare, de obicei rețelele din “spatele” său). Cei 2 vecini BGP se numesc de obicei *peers* iar procesul se numește *BGP peering*. După aceea, ruterele trimit în *routing updates* doar schimbările survenite. În lipsa oricărei schimbări, BGP trimite mesaje *keepalive* la fiecare 60s. Aceasta prin contrast cu RIP/OSPF, care trimit în *updates* în permanență aceeași informație, indiferent dacă s-a schimbat sau nu.

1.2 Conținutul mesajelor BGP

În mesajele BGP există mai multe câmpuri, dintre care un header. Acesta conține un câmp care descrie tipul mesajului; sînt 4 tipuri de mesaje: stabilire a conexiunii (OPEN), menținere în funcție a conexiunii (KEEPALIVE – un mesaj de 19 octeți, care reprezintă lungimea minimă a headerului+tipului, fără alte date), semnalare a unei erori (NOTIFICATION) și informare (UPDATE).

Mesajele UPDATE sînt cele mai importante, căci conțin cele mai multe date și pe baza lor BGP crează o imagine (un graf) al întregii rețele, pe baza căruia se stabilește ruta optimă către orice destinație. Cele 3 componente ale unui mesaj UPDATE sînt:

1. NLRI (*Network Layer Reachability Information*): un câmp care informează despre rutele disponibile pe ruterul respectiv, prin enumerarea fiecărei perechi (*maskă, rețea*). Pentru economie, măștile se transmit sub forma prescurtată /N. În terminologia BGP ele se numesc *prefix length* (prin analogie cu rețelele care se numesc *prefixe* întrucît reprezintă prima parte a unei adrese IP). De exemplu, o rută către rețeaua 192.168.1.0/255.255.255.0 se scrie simplificat ca 192.168.1.0/24 și se transmite ca:

<24, 192.168.1.0>

2. attribute (*Path Attributes*) ale fiecărei rute. Dintre acestea, cele mai importante sînt:

Cod și denumire	tip
1. ORIGIN	<i>Well-known mandatory</i>
2. NEXT_HOP	<i>Well-known mandatory</i>
3. AS_PATH	<i>Well-known mandatory</i>
4. MED (MULTI_EXIT_DISCRIMINATOR)	<i>Optional nontransitive</i>
5. LOCAL_PREF	<i>Well-known discretionary</i>
6. ATOMIC_AGGREGATE	<i>Well-known discretionary</i>
7. AGGREGATOR	<i>Well-known discretionary</i>
8. COMMUNITY	<i>Optional transitive (Cisco)</i>
9. ORIGINATOR_ID	<i>Optional nontransitive (Cisco)</i>
10. CLUSTER LIST	<i>Optional nontransitive (Cisco)</i>

Semnificația tipurilor (al căror nume a fost păstrat conform standardului) este următoarea:

- *Well-known mandatory*: atributul trebuie să existe în orice pachet UPDATE și este recunoscut de toate implementările BGP
- *Well-known discretionary*: poate să existe sau poate să lipsească, și este recunoscut de toate implementările BGP
- *Optional transitive*: poate să existe sau poate să lipsească, nu este recunoscut de toate implementările BGP, dar chiar și ruterele care nu știu să-l interpreteze trebuie să-l trimită mai departe (tranzitiv)
- *Optional nontransitive*: față de cazul de mai sus, nu este transmis mai departe.

Atributele rutelor sînt folosite extensiv în BGP pentru a stabili politicile de rutare; pe baza lor administratorii pot filtra anumite rute, pot prefera unele rute în locul altora, etc.

3. informații despre rutele care nu mai sînt disponibile (*withdrawn routes*)

1.3 Configurarea BGP

În configurarea unui ruter BGP trebuie precizate cel puțin următoarele 3 informații: (1) AS-ul propriu, (2) rețelele anunțate (*advertised networks*) de ruterul nostru, și (3) ruterele BGP vecine (*neighbors*) - adresa și AS-ul acestora. Sintaxa este:

```
router bgp AS-number
network network-number [mask netmask]
network ...
neighbor neighbor-address remote-as AS-number
neighbor ...
```

Rețelele anunțate sînt doar acele rețele pe care *dorim* să le anunțăm. Este o deosebire de cazul RIP/OSPF în care, prin configurarea unor rețele, automat configurăm interfețele direct conectate ca “participînd” la protocolul de rutare respectiv și deci efectul este de a trimite *updates* prin acele interfețe. În BGP, anunțarea acestor rețele nu are efect asupra interfețelor. Se poate specifica manual o interfață de comunicație cu fiecare vecin, iar dacă nu, ruterul va alege interfața cea mai apropiată. Rețelele anunțate vor fi incluse în tabela de rutare BGP.

Prin specificarea *AS-number* BGP cunoaște AS-ul propriu; în continuare, în funcție de *AS-number* specificat pentru fiecare *neighbor*, BGP determină singur dacă respectivul este în același AS (și deci protocolul va rula ca IBGP) sau în alt AS (EBGP).

În afara acestei configurații obligatorii, BGP suportă opțiuni de configurare foarte complexe, în funcție de politicile pe care administratorii doresc să le implementeze (în general prin manipularea atributelor rutelor); vom studia doar cîteva din posibilități pe baza unor scenarii. Mult mai multe detalii găsiți în bibliografie.

Partea 2. Desfășurare; configurare; exerciții

OBSERVAȚII

- **1:** reamintim că pentru a întrerupe o comanda *ping*, *traceroute* etc care nu merge se va folosi secvența **CTRL-SHIFT-6**
- **2:** pentru a forța ruterul să nu ne mai scoată automat din modul “enabled” după un timp de inactivitate, se dau comenzile:

```
line con 0
no exec-timeout
```

- **3:** Pentru a putea face telnet de pe un ruter pe altul se fac următoarele:

- se pune parola pe telnet (*cisco*)
- se pune parola de enable (*class*)
- pentru a nu scrie mereu telnet adresa_ip se poate defini un nume de host, de exemplu pe R0:

```
R0(config)# ip host 10.1.1.2 isp1
```

și de acum în loc de telnet 10.1.1.2 se poate da simplu telnet isp1 sau chiar isp1 (dacă nu recunoaște numele ca o comandă, el presupune implicit că se dorește telnet la hostul cu numele respectiv). Același no exec-timeout se poate da și pe telnet (line vty 0 4)

Faza 1: Conectarea unui client la 2 ISP folosind BGP

Se realizează topologia ca mai jos: interfețele care leagă între ele ISP1 și ISP2 nu se configurează momentan, dar se face conexiunea serială (ca în figura 4) și se lasă în starea “administratively down”.

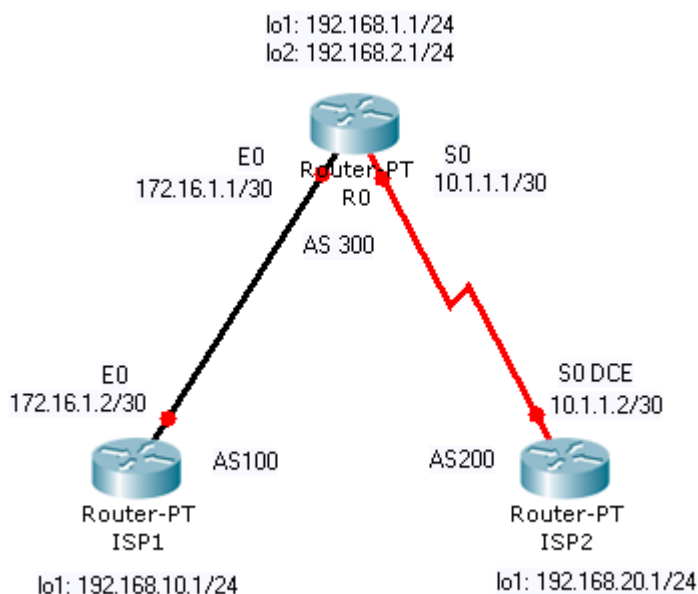


Fig. 3 Topologia inițială

Se începe configurarea BGP. Cele 3 routere aparțin de 3 AS-uri diferite. Se urmărește ca rețelele clientului (simulate de cele 2 interfețe loopback) să aibă acces la internet (simulat de cele 2 loopback-uri ale ISP).

a) Se configurează BGP pe ruterul R0 al clientului, care aparține AS 300; se vor anunța cu comanda *network* **numai** cele 2 rețele ale loopbackurilor. Nu vom anunța rețelele către ruterele ISP, acestea nu aparțin clientului. BGP are altă utilizare decât RIP, și anume să anunțe rutele “dorite”. Se pot anunța cu *network* rute direct conectate, sau rute statice/dinamice cunoscute de ruter. BGP va anunța în mod implicit alte rute BGP (aflate de la vecinii BGP) dar acestea pot fi filtrate. Reamintim că BGP este folosit pentru a face rutare bazată pe politici, nu pentru a determina automat cea mai bună rută din toate rutele posibile.

```
R0(config)# router bgp 300
R0(config-router)# network 192.168.1.0
R0(config-router)# network 192.168.2.0
R0(config-router)# neighbor 172.16.1.2 remote-as 100
R0(config-router)# neighbor 10.1.1.2 remote-as 200
```

Se configurează similar cele 2 ISP, de exemplu pentru ISP1:

```
ISP1(config)# router bgp 100
```

```
ISP1(config-router)# network 192.168.10.0  
ISP1(config-router)# neighbor 172.16.1.1 remote-as 300
```

b) se așteaptă pînă cînd converge BGP-ul; se vizualizează tabela de rutare de pe cele 3 rutere:

```
R0# sh ip route
```

Q1. Ce rețele se văd pe fiecare ruter prin BGP (prefixul B)?

Se verifică posibilitatea de a da ping de la R0 către ISP1 și ISP2.

c) se verifică funcționarea BGP pe R0 folosind `show ip bgp`. Verificați în legendă ce semnifică simbolurile “*” și “>”

Verificați pe R0 detalii despre vecini cu `show ip bgp neighbor`.

Q2. Care este starea BGP (BGP State) ? Ce linie afișată indică faptul că este IBGP sau EBGP?

Ultima comandă mai suportă și 2 parametri adiționali, pentru a vedea ce rute sînt primite de la vecini (*received*) sau trimise către vecini (*advertised*).

```
sh ip bgp neighbor [neighbor-address] [received-routes | routes |  
advertised-routes]
```

Alte comenzi pentru verificare:

```
sh ip bgp paths  
sh ip bgp summary
```

Q3. Folosind ultima comandă pe R0, determinați cîtă memorie consumă BGP pentru rețele (*networks*) și în total.

Pe baza acestei informații, puteți estima cîtă memorie este necesară pentru ruterele BGP din rețelele *backbone* ale internetului, și de ce sumarizarea și agregarea rutelor trebuiesc folosite cît mai mult!

Faza 2: Transformarea AS300 din domeniu de tranzit în non-tranzit

a) În configurația precedentă, observați cu `sh ip route` în tabela de rutare a ISP1 că apare ruta către loopback-ul ISP2; aceasta deoarece ISP2 anunță ruta către R0, și R0 o anunță către ISP1. Se verifică că merge ping-ul între ISP1 și ISP2. Acest lucru este nedorit în acest caz, căci semnifică că clientul R0 poate fi folosit ca *tranzit* de către ISP1 și ISP2 (vezi paragraful 1.1).

Pentru a verifica că un host din rețeaua ISP1 poate da ping unui host din rețeaua ISP2, trebuie folosit *extended ping* (comanda ping fără parametri, și la “extended commands” se va raspunde “yes” pentru a ne permite să alegem noi adresa sursă). Protocolul va fi IP.

Q4. Care este interfața de rețea a lui ISP1 (din cele 2 definite: Ethernet și Lo1) folosită ca sursă dacă se dă ping către o destinație din ISP2? De ce nu este suficient ping simplu în acest caz ? A avut succes ping/ extended ping de la ISP1 la ISP2 ? de ce ?.

b) Pentru a interzice tranzitul între ISP1 și ISP2 prin R0 se crează un access-list pe R0 care permite doar anunțarea celor 2 rețele ale R0 (cele 2 loopback-uri):

```
R0(config)# access-list 1 permit 192.168.0.0 0.0.3.255
```

Q5. Explicați alegerea wildcard mask-ului 0.0.3.255. Ce rețele sînt permise ? Ce regulă implicită consideră ruterele Cisco că există la sfîrșitul unui access-list ?

Se instalează access-list-ul pe R0 ca filtru (*route filter*), folosind cuvîntul cheie *distribute-list*:

```
R0(config)# router bgp 300
R0(config-router)# neighbor 172.16.1.2 distribute-list 1 out
R0(config-router)# neighbor 10.1.1.2 distribute-list 1 out
R0(config-router)# CTRL-Z
```

Se verifică pe ISP1 existența rețelei 192.168.20.0 de pe ISP2 în tabela de rutare;

Q6. Mai apare rețeaua lui ISP2 (192.168.20.0) ?

Pe R0 se da următoarea comandă, care șterge toate rutele BGP:

```
R0# clear ip bgp *
```

și se așteaptă ca BGP-ul să ajungă în starea Established (folosind `sh ip bgp neighbor`). Apoi se examinează din nou tabela de rutare de pe ISP1.

Q7. Mai apare rețeaua lui ISP2 (192.168.20.0) ? de ce ?

În continuare, se anulează cu *no* cele 2 comenzi `neighbor ... distribute-list ...` pentru a reveni la situația inițială. Se repornește BGP pe R0 cu `clear ip bgp *`

Faza 3: Sumarizarea și agregarea rutelor

Reamintim 2 concepte:

- sumarizarea se referă la concentrarea a mai multe subneturi într-un singur net, de obicei la limita de clasă
- agregarea este similară cu sumarizarea, dar la o limită arbitrară; dacă limita este mai mică decît limita de clasă, prin agregare se obține un *supernet* (de exemplu 2 neturi clasă C, adică /24 se combină într-un supernet /23).

Q8. RIP face sumarizarea automată a rutelor la limita de clasă ? cum se controlează acest lucru ? dar OSPF ?

Pentru a afla comportamentul BGP, se definesc 2 noi rute statice pe ISP2. Întrucît ne interesează doar distribuirea acestor rute prin BGP, le vom atașa unei interfețe virtuale:

```
ISP2(config)# ip route 192.168.5.0 255.255.255.192 null0
ISP2(config)# ip route 192.168.5.64 255.255.255.192 null0
```

Q9. Ce lungime are masca pentru aceste rute? este vorba de un supernet ?

Se redistribuie rutele statice în BGP:

```
ISP2(config)# router BGP 200  
ISP2(config-router)# redistribute static
```

Se așteaptă pînă BGP propagă rutele pe R0 și ISP1. Se verifică cu `sh ip bgp` sau `sh ip route` pe aceste rutere dacă apar ambele rute sau un *summary* al lor.

Q10. Din rezultatul afișat, rezultă că BGP face sau nu sumarizarea automată la limita de clasă?

Comportamentul BGP poate fi controlat folosind comenzile `auto-summary`, respectiv `no auto-summary`

Pe R2 dați comanda:

```
ISP2(config)# router BGP 200  
ISP2(config-router)# no auto-summary
```

Pe R0 ștergeți din nou rutele cu `clear ip route *` și așteptați să convergă

Q11. Ce rute se transmit pe R0 ?

Pentru a controla manual agregarea vom da comanda:

```
ISP2(config-router)# aggregate-address rețea mască
```

unde rețeaua și masca sînt ale rețelei agregate (sau a supernetului) dorite.

Q12. Scrieți rețeaua și masca necesare pentru a agrega cele exact 2 rute ale subneturilor definite mai sus într-o singură rută.

Configurați pe ISP2 ruta agregată cu comanda de mai sus; ștergeți tabela de rutare de pe R0, așteptați să convergă și verificați ce s-a întîmplat.

Q13. Ce schimbare apare pe R0?

Pentru a forța BGP să transmită *doar* ruta agregată, nu și subneturile mai specifice, se dă comanda:

```
ISP2(config-router)# aggregate-address rețea mască summary-only
```

Configurați ISP2 astfel: ștergeți cu *no* comanda precedentă (cea fără `summary-only`) și introduceți comanda cu parametrul adițional `summary-only`. Ștergeți, așteptați și vizualizați din nou ce se întîmplă pe R0.

Q14. Ce schimbare apare acum pe R0?

Observați că în acest caz, agregarea e mai bună decît sumarizarea, întrucît sînt doar 2 subneturi /26 din cele 4 posibile, deci prin sumarizarea la limita de clasă nu se transmite o informație “exactă”.

Q15. Ce lungime are masca *summary*, respectiv a rutei sumarizate, care a apărut/apare pe R0?

Faza 4: Definirea rutelor implicite (*default route, default network*)

Pentru a ilustra conceptul de rută *default* se mai crează o rețea, folosind câte o interfață serială pe cele 2 rutere ISP, care se configurează ca în figura 4:

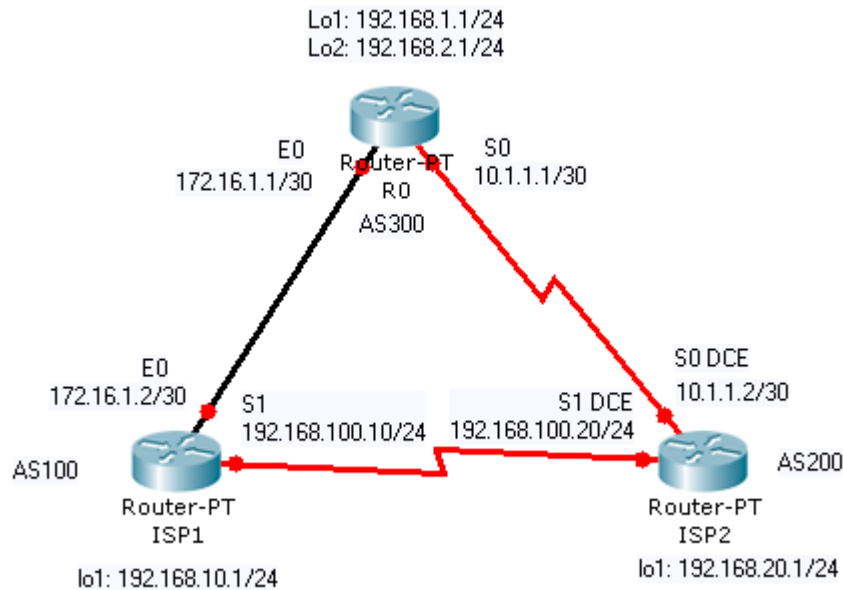


Fig. 4 Completarea topologiei pentru faza 4

Rețeaua 192.168.100.0 “simulează” o rețea din internet, care va trebui să fie accesibilă lui R0 atât prin ISP1 cât și prin ISP2 (prin definiție, orice rețea din Internet trebuie să fie accesibilă oricărui ISP).

O rută *default* se poate obține și fără rute statice, folosind *default-network*. Mai întâi configurăm ISP1 și ISP2 să anunțe rețeaua 192.168.100.0 prin BGP, adăugând o nouă intrare *network*:

```
ISP1(config)# router bgp 100  
ISP1(config-router)# network 192.168.100.0
```

Și similar pe ISP2.

Se mai adaugă pe ISP1 și pe ISP2 o interfață loopback *lo2* cu adresa 192.168.30.1/24, care *nu* se anunță în procesul BGP. Această interfață simulează o adresă oarecare din internet, necunoscută clientului (nu mai are sens să testăm cu ping că putem ajunge la 192.168.100.x întrucât această rețea este anunțată prin BGP de către ambii ISP, deci este cunoscută clientului). Evident în mod normal nu se alocă aceeași adresă la *două* interfețe de pe rutere diferite, dar aici scopul e ca adresa să fie accesibilă atât de către ISP1, cât și de către ISP2 (și nu se detectează suprapunerea, căci nici unul nu o anunță).

Q16. Funcționează extended ping de pe R0 (192.168.1.1) pe 192.168.30.1 ? de ce ?

Se configurează o rută *default* către rețeaua 192.168.100.0:

```
R0(config)# ip default-network 192.168.100.0
```

Se examinează tabela de rutare de pe R0 cu `sh ip route`

Q17. Cine apare ca *gateway of last resort* ?

Q18. Cât este distanța administrativă a unei rute EBGp? Între o rută *default* statică cu distanța administrativă 220 și una *default* EBGp, care va fi preferată ? ce utilizare ar avea, în acest context, o rută statică *default*?

Q19. Apare în tabela lui R0 o rută către rețeaua 192.168.30.0 ? Funcționează *extended ping* de pe R0 (192.168.1.1) pe 192.168.30.1 ? de ce ?

Faza 5: Policy routing folosind *weight*; atributul AS_PATH

Atunci când există mai multe rute către o destinație, procesul de selecție este algoritmic și constă în verificarea, pe rând, a mai multor condiții [1]. Dacă toate condițiile sînt la egalitate, va fi preferată ruta prin ruterul avînd ID *mai mic*. ID-ul unui ruter este dat de adresa IP de pe care vine un *update BGP*.

În exemplul nostru, clientul poate ajunge la rețeaua 192.168.100.0 atît prin ISP1 cît și ISP2. Verificați în tabela BGP (folosind `sh ip bgp`) care este ruta preferată (*best*, cea marcată cu ">")

Q20. Care este ruta preferată pe R0 către rețeaua 192.168.100.0 ?

Observați că s-a ales Next Hop cu adresa IP mai mică.

Observație: e posibil ca rezultatul să nu fie acesta, deoarece la punctul precedent s-au manipulat rutele în timp ce BGP era deja activ. În acest caz, se va ajunge la acest rezultat prin reinițializarea procesului BGP pe toate ruterile cu `clear ip bgp *` și eventual `clear ip route *`

BGP permite alegerea rutei preferate în mai multe moduri; un mod este folosirea atributului *weight* care este implicit 32768 pentru rute pe care ruterul le anunță el-însuși și 0 în rest.

Observație: manipularea tabelii de rutare în orice scop care nu e pur tehnic poartă numele generic de *policy routing* (rutare bazată pe politici).

Q21. Cât este *weight* pentru rețelele 192.168.1.0, respectiv 192.168.100.0 în tabela BGP ?

Vom alege o rută și vom specifica AS-ul vecinului ales de noi, folosind atributul BGP AS_PATH. Acest atribut conține calea dintre sursă și destinație, obținută prin concatenarea AS-urilor prin care se trece; în acest caz, de la AS100 la AS300 este un singur hop, deci AS_PATH va conține doar AS-ul original și anume 100.

Q22. De ce tip este atributul BGP AS_PATH? (vezi lista de atribute).

Q23. Cât este cîmpul *path* pentru fiecare dintre cele 2 intrări BGP care specifică ruta către 192.168.100.0 ?

Atributul *weight* este local pentru un ruter și nu este propagat în *update*-urile BGP. Pentru a schimba *weight*-ul unei rute către o destinație, învățate de la alt ruter, se pot folosi mai multe metode. Una constă în folosirea *route-maps*; o *route-map* permite stabilirea unuia sau mai multor atribute cu *set* atunci când se îndeplinește condiția după cuvîntul cheie *match*. În acest caz, vom seta atribute BGP.

Se definește pe R0 un *route-map* numit **abc** care să aloce un *weight* de 33000 (mai mare decît 32768) tuturor rutelor care au atributul AS_PATH 100, adică provin de la AS100.

Pentru a specifica *exact* numărul 100 și nu *orice* număr care conține 100 (de exemplu 11002) se folosesc caracterele speciale `^` (marchează începutul) și `$` (marchează sfîrșitul). Așadar `^100$` înseamnă un număr care începe cu 100 și nu mai are altceva la sfîrșit.

OBSERVAȚIE: Aceste caractere formează ceea ce se numește *regular expressions* și nu sînt specifice Cisco; le puteți întâlni și în limbaje ca Perl, awk, în utilitare de Unix precum grep, etc.

Selectarea unui AS-PATH se face cu o formă particulară de access-list:

```
R0(config)# ip as-path access-list 1 permit ^100$
```

apoi se definește un *route-map* care are ca efect atribuirea noului *weight* (folosind cuvîntul-cheie *set*) atunci cînd se îndeplinește condiția din ACL-ul 1 (folosind cuvîntul-cheie *match*); un *route-map*, ca și un ACL, poate avea mai multe reguli, dar spre deosebire de acesta din urmă, regulile sînt numerotate pentru a putea fi modificate independent și inserate în poziția dorită, nu doar la sfîrșit (în acest caz există o singură regulă cu numărul 10):

```
R0(config)# route-map abc permit 10  
R0(config-route-map)# match as-path 1  
R0(config-route-map)# set weight 33000
```

Și se introduce această regulă în protocolul de rutare:

```
R0(config)# router bgp 300  
R0(config-router)# neighbor 172.16.1.2 route-map abc in
```

Se observă că sensul este “in” deoarece se aplică pentru rutele învățate de ruter, deci cele care “intră”.

După ce se configurează BGP astfel, se șterge tabela cu `clear ip bgp *` și, după convergență, se examinează din nou cu `sh ip bgp`.

Q24. Cît este valoarea *weight* în noua tabelă pentru *path = 100* ? ce AS apare ca destinație favorită (simbolul “>”) pentru ruta 192.168.100.1 ?

Se verifică că funcționează în continuare un extended ping de pe 192.168.1.1 către 192.168.100.1.

Observație: în mod similar, se poate modifica ruta aleasă folosind atributul LOCAL_PREF; detalii în bibliografie.

Faza 6: Numere private de AS

Se realizează topologia de mai jos, care corespunde unui client conectat la un singur ISP:

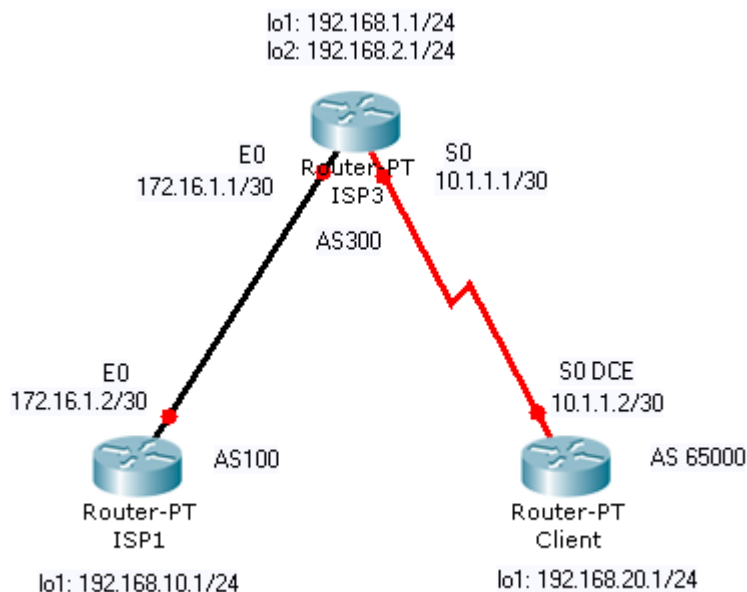


Fig. 5 Topologia pentru faza 6

Pentru a obține aceasta topologie din cea veche se fac următoarele modificări:

- de pe ruterul ISP1 se scoate din BGP rețeaua 192.168.100.0:

```
ISP1(config)# router bgp 100
ISP1(config-router)# no net 192.168.100.0
```

și se dă shutdown la interfața serială care îl lega de ISP2

- pe ruterul R0 (noul ISP3) se șterge procesul de rutare BGP și se mai pune o dată (similar cu Faza 1, dar atenție că vecinul din dreapta are acum AS65000) pentru a scăpa de toate opțiunile introduse în scenariile anterioare:

```
R0(config)# hostname ISP3
ISP3(config)# no router bgp 300
ISP3(config)# router bgp 300
ISP3(config-router)# nei 10.1.1.2 remote-as 65000
... etc ...
```

- pe ruterul ISP2 (noul Client) se oprește procesul de rutare BGP folosind comanda:

```
no router bgp 200
```

și se repornește cu AS 65000, în rest ca în Faza 1

Se verifică că de pe ISP1 se poate da (extended) ping pe rețeaua de pe client (192.168.20.1).

Pe ISP1 se verifică tabela bgp folosind `sh ip bgp`. Câmpul *Path* din tabelă cuprinde calea prin care se ajunge la fiecare rețea destinație, sub forma lanțului de AS-uri parcurs (reamintim că se obține pe baza atributului AS_PATH din mesajele de UPDATE; acestea se concatenează la fiecare "hop" parcurs, astfel încât în permanență se dispune de întreaga cale).

Q25. Ce valoare are *Path* pentru ruta 192.168.20.0 de pe Client ?

Această configurație este incorectă din următorul motiv: clientul este o rețea de tip *stub* (*single-homed*) deci nu are nevoie de un AS rutabil, ci unul privat. Se observă însă că acest număr de AS se vede de pe ISP1 (din internet). Numerele private de AS nu trebuie să fie vizibile, așa cum nu trebuie să fie vizibile adresele IP private din RFC1918 (10.0.0.0, 172.16.0.0-172.31.0.0, 192.168.0.0).

Se configurează routerul ISP3 (care este ISP-ul clientului) să elimine numerele de AS private din mesajele către ISP1 (deci spre internet), astfel:

```
ISP3(config)# router bgp 300  
ISP3(config-router)# neighbor 172.16.1.2 remove-private-as
```

Se revine pe ISP1, se dă `clear ip bgp *`, se verifică din nou tabela BGP.

Q26. Care este noua valoare *Path* pentru ruta 192.168.20.0 ?

Bibliografie

- [1] Cisco CCNP1 v3.0, cap. 9
- [2] Using the Border Gateway Protocol for Interdomain Routing (document online Cisco - *ics-bgp4.pdf*)
- [3] Cisco IOS Configuration Guide (document online Cisco – *Icfbook.pdf*)