Universitatea “Politehnica” din Bucureşti

Facultatea de Electronică, Telecomunicaţii şi Tehnologia Informaţiei

Protocolul BGP

Temă de curs

Rețele de Calculatoare și Internet

Profesor coordonator

Conf. dr. ing. Ștefan Stăncescu

Masterand

Theodor Cristian BUCULEI

2017 - 2018

Cuprins

1. Introducere……..……. 4
2. Caracteristici BGP…….……..……. 4
3. Blocurile de construcție ale protocolului BGP ……..……. 7
4. Atribute de rutare BGP……...…… 10
5. Protocolul BGP INTERN ……...…….....….. 13
6. Procesul de stabilire a rutelor…….…… 15
7. Proprietăti BGP …….… 17
   7.1. Mesajele accepate de protocol ……...…. 17
   7.2. Schimbul rutelor BGP-IGP …….……..…….. 20
8. Baza informaționala de rutare …….…….…… 20
9. Rutele de comutare ……..…….……. 21
   9.1. Comutarea proceselor ……..…… 21
   9.2. Comutarea bazată pe cache ……...……...…… 23
      9.2.1. Comutarea rapidă ……...…… 24
      9.2.2. Comutarea optimă ……..…… 24
      9.2.3. Comutare optimă distribuită …….…….. 24
      9.2.4. Comutarea NetFlow ……...…. 24
      9.2.5. Deficiențe ale metodelor de comutare bazate pe cache …….. 25
   9.3. Cisco Express Forwarding ……..……. 26
10. Proprietăti de performantă a rețelei BGP ……...……. 30
11. Concluzii …….…… 30
12. Bibliografie …….……. 31

1. Introducere

Border Gateway Protocol (BGP) este un protocol de rutare care este folosit pentru a face schimb de informații despre disponibilitatea stratului de rețea (NLRI) între domeniile de rutare. Un domeniu de rutare este numit adesea un sistem autonom (AS) deoarece diferite autorități administrative controlează domeniile respective. Actualul Internet este o rețea de sisteme autonome interconectate, în care BGP versiunea 4 (BGP4) este protocolul de rutare.

2. Caracteristici BGP

Internetul a crescut semnificativ în ultimele decenii. Actualul tabel BGP din Internet are peste 100.000 de rute. Multe întreprinderi folosesc , de asemenea, BGP pentru a-și interconecta rețelele. Aceste implementări pe scară largă au dovedit capacitatea BGP de a sprijini rețelele mari și complexe.

Protocolul BGP are următoarele caracteristici:

- Fiabilitate
- Stabilitate
- Scalabilitate
- Flexibilitate

A. Fiabilitatea BGP poate fi examinată din mai multe puncte de vedere:

➔ Realizarea conexiunii
➔ Întreținerea conexiunii
➔ Acuratețea informațiilor de rutare

BGP profită de serviciul de transport fiabil furnizat de Transmission Control Protocol (TCP). Aceasta elimină necesitatea ca BGP să implementeze fragmentarea actualizării, retransmiterea, confirmarea și secventierea, deoarece TCP are grijă de aceste funcții. În plus, orice schemă de autentificare utilizată de TCP poate fi utilizată și de BGP.

După ce sesiunea este stabilită, BGP folosește mesaje keepalive regulate pentru a menține integritatea sesiunii. Actualizarea mesajelor, de asemenea, resetează cronometrul de așteptare, care de obicei este de trei ori mai mare decât cronometrul de întreținere. O sesiune BGP este inchisă dacă sunt pierdute trei mesaje keepalive consecutive și nu sunt primite mesaje de tip Update.

Informațiile de rutare corecte sunt importante pentru transmiterea fiabilă. BGP utilizează mai multe măsuri pentru a crește precizia. Când se primesc actualizări, se verifică AS_PATH (un atribut BGP care listează sistemelor autonome traseul traversat) pentru a detecta buclele. Actualizările provenite din AS-ul curent sau care au trecut prin AS sunt refuzate. Filtrele

de intrare pot fi aplicate tuturor actualizărilor care asigură conformitatea cu politicile locale. Accesibilitatea următorului hop este verificată în mod regulat înainte ca o rutare BGP să fie considerată valabilă.

Pentru a menține precizia informațiilor de rutare, este, de asemenea, important să fie eliminate în timp util rutele inaccesibile.

B. Stabilitatea

Stabilitatea protocolului de rutare este critică pentru rețelele mari. Având în vedere mărimea actualului internet, accesarea unui număr mare de rute poate fi catastrofală.

Prin implementarea diferiților timeri, BGP suprimă impactul interfețelor sau rutează sus/jos evenimentele în rețea. De exemplu, un difuzor BGP poate genera actualizări pană la intervalul minim de publicitate. În software-ul Cisco IOS, intervalul este de 30 de secunde pentru sesiunile BGP (eBGP) externe și 5 secunde pentru sesiunile interne BGP (iBGP) .

Anularea zgomotului este o altă caracteristică BGP care suprimă instabilitatea. Router-ul urmăreste istoricul unui traseu. Căile instabile sunt penalizate și supuse suprimării.

Stabilitatea poate fi mărită dacă sesiunile nu trebuie resetate atunci când se schimbă o politică. Caracteristicile cum ar fi reconfigurarea de soft și reimprospătarea traseului, sunt utile pentru modificarea politicii BGP fără resetarea sesiunii BGP. Ambele funcții permit actualizările noi să fie solicitate sau trimise dinamic.

Dacă o sesiune trebuie resetată, toate informațiile de rutare BGP și de redirecționare pentru acea sesiune sunt șterse. Acest lucru ar putea duce la pierderea pachetelor pană când va fi construită o nouă bază de date de redirecționare. Transferul non-stop (NSF) sau restartarea (Graceful Restart) permite unui router să continue transmiterea cu informațiile existente (reținute din sesiunea anterioară) în timp ce sesiunea este resetată.

Convergența este procesul în care o rețea se sincronizează cu aceleași informații de rutare după o schimbare în rețea. O rețea care nu este convergentă poate duce la pierderi de pachete sau bucle de forwardare. Cu toate acestea, stabilitatea poate fi redusă dacă o rețea se află într-o stare constantă de convergentă. Un echilibru adecvat al stabilitătii și convergenței poate depinde de serviciile oferite de rețea. De exemplu, atunci când BGP este utilizat pentru a furniza servicii de rețea privată virtuală (VPN) printr-o rețea partajată Multiprotocol Label Switching (MPLS), este posibil că accentul să fie pus pe convergentă.

C. Scalabilitate

Scalabilitatea BGP se poate evalua în două domenii: numărul de sesiuni și numărul de rute. În funcție de configurație, platforma hardware (procesor și memorie) și versiunea Cisco

IOS, BGP a demonstrat să susțină sute de sesiuni și să mențină peste 100.000 de rute. Există mai multe măsuri pentru a crește scalabilitatea BGP.

Aceste măsuri reduc fie numărul de rute / căi care trebuie menținute, fie numărul de actualizări care trebuie generate.

Ca formă de protocol vectorial de distanță, BGP își actualizează conexiunile (peer) numai cu căile pe care le utilizează. Cu alte cuvinte, numai cele mai bune căi sunt direcționate către conexiunile sale. Când se schimbă calea cea mai bună, noua cale este anuntată, ceea ce le permite conexiunilor să-și inlocuiască cea mai bună anterioară cale cu cea găsită recent.

Când BGP este folosit pentru a schimba informații de accesibilitate în cadrul aceleiași AS, toate conexiunile BGP trebuie să fie complet conectate.

Deoarece rețelele conectate în întregime tind să limiteze scalabilitatea din cauza numărului de sesiuni care trebuie menținute pe fiecare router și a numărului de actualizări care trebuie generate, reflecția rutei este o metodă care mărește scalabilitatea rețelelor BGP.

Agregarea rutelor este un alt instrument pe care BGP îl folosește pentru a reduce numărul de prefixe care trebuie direcționate și pentru a spori stabilitatea. Reducerea numărului de actualizări care urmează a fi generate reduce utilizarea procesorului și permite o convergentă mai rapidă.

D. Flexibilitate

BGP este un protocol vectorial de cale, o formă de protocol vectorial de distanță care construiește un grafic abstract al sistemelor autonome pentru fiecare destinație. Flexibilitatea BGP este demonstrată prin numărul de atribute ale căii care pot fi utilizate pentru a defini politicile. Atribuiile căii BGP sunt parametrii care descriu caracteristicile unui prefix BGP.

Puteți defini două tipuri de politici pentru BGP: rutare și administrare. Aceste politici se suprapun adesea în funcționalitatea lor.

Se poate defini o politică de rutare BGP pentru direcția de intrare sau de ieșire care afectează calea sau selectarea traseului. De exemplu, o politică de filtrare la intrare poate fi definită pentru a accepta rute care provin numai de la furnizorul imediat existent și de la clienții acelui furnizor. Cu setarea corectă a unor atribute, o cale poate fi preferată în defavoarea altora.

Politica administrativă a BGP definește controalele administrative pentru rutele care intră în sistemul auxiliar sau care părăsesc sistemul auxiliar. De exemplu, un AS ar putea intenționa să-și protejeze routerele de frontieră prin limitarea numărului maxim de prefixe pe care le permite să le primească.

Pe partea exterioară, un ruter de frontieră al unei AS ar putea alege să își stabilească atributul astfel încât numai rutele localizate local să fie anunțate.

Pe măsură ce sunt primite actualizările de la o conexiune, acestea sunt stocate într-o bază de informații de rutare (RIB) pentru acea rută(Adj-RIB-In).

Un algoritm de selectare a traseului este apoi efectuat pentru a determina calea cea mai bună pentru fiecare prefix.

Cele mai bune căi rezultate sunt stocate în BGP RIB local (Loc-RIB) și apoi sunt trimise la tabela locală de rutare IP (IP-RIB).

Când căile multiple sunt activate, calea cea mai bună, plus toate căile de cost egal, sunt trimise pentru examinarea pentru a fi luate în considerare IP-RIB.

În completarea celor mai bune conexiuni, Loc-RIB conține, de asemenea, prefixe BGP injectate de ruterul curent (numite surse locale) care sunt selectate ca fiind cele mai bune căi. Conținutul Loc-RIB trebuie să treacă prin Output Policy Engine înainte de a fi direcționat către alte rute. Traseele care trec cu succes prin motorul de politică de ieșire sunt instalate în ieșirea RIB (Adj-RIB-Ouț).

Un proces de actualizare poate varia în funcție de implementarea și configurarea BGP. În Cisco IOS, tabelul BGP sau BGP RIB (rezultatul comenzii show ip bgp) conține toate rutele permise de Input Policy Engine, inclusiv rutele care nu sunt selectate drept cele mai bune căi. Atunci când este activată caracteristica Inbound Soft Reset IOS (reconfigurare software), rutele care sunt refuzate de Input Policy Engine sunt de asemenea păstrate (marcate ca Receive), dar nu sunt luate în considerare în procesul de selectare al traseului.

3. Blocurile de construcție ale protocolului BGP

BGP este format din trei mari procese:

- ➔ I/O
- ➔ Router
- ➔ Scanner

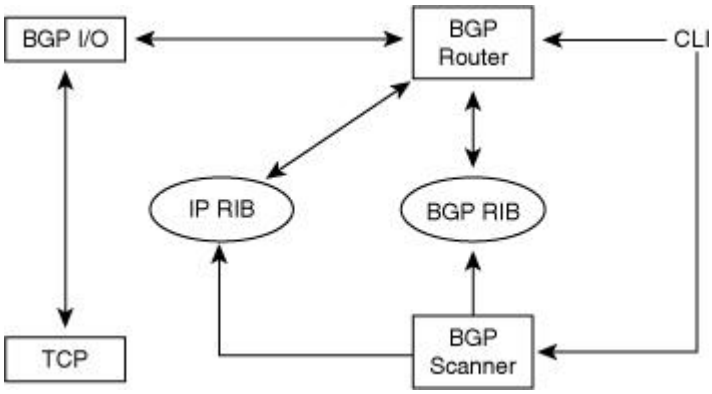

Figura 1: Procesele BGP

Procesul BGP I / O se ocupă de citirea, scrierea și executarea mesajelor BGP. Acesta oferă interfața dintre TCP și BGP. Pe de o parte, citește mesajele din socketul TCP și le pune în coada de intrare BGP (InQ) pentru a fi procesate de procesul de router BGP. Pe de altă parte, mesajele acumulate în coada de ieșire (OutQ) sunt mutate de către BGP I / O către socketul TCP.

Procesul BGP Router este un proces BGP principal care este responsabil pentru inițierea altor procese BGP, menținerea sesiunilor BGP cu vecinii, procesarea actualizărilor primite de la conexiuni și rețele din surse locale, actualizarea IP RIB cu intrări BGP și trimiterea de actualizări către conexiuni. În mod specific, procesul de router BGP primește comenzi introduse de la CLI (Common Line Interface) prin parser. Interactionează cu procesul de intrare / ieșire BGP pentru procesarea actualizării (trimitere și primire) utilizând cozile per-vecin, așa cum este arătat **in Figura 2**.

După ce toate căile valide sunt instalate în BGP RIB, Router-ul BGP rulează selecția traseului și instalează cele mai bune căi în IP RIB. Evenimentele care se petrec în IP RIB și BGP RIB pot, de asemenea, declanșa acțiuni adecvate în procesul de rutare BGP. De exemplu, atunci când o cale trebuie redistribuită dintr-un alt protocol către BGP, IP RIB notifică Router-ul BGP să actualizeze BGP RIB.

```
router#show ip bgp summary
BGP router identifier 192.168.100.6, local AS number 100
BGP table version is 8, main routing table version 8
4 network entries and 7 paths using 668 bytes of memory
3 BGP path attribute entries using 180 bytes of memory
6 BGP rrinfo entries using 144 bytes of memory
1 BGP AS-PATH entries using 24 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 activity 4/74 prefixes, 11/4 paths, scan interval 60 secs

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
192.168.100.4   4   100    1120    1119        8    0    0 17:12:34        3
192.168.100.5   4   100    1114    1111        8    0    0 00:07:35        3
```

Figura 2: Cozile BGP

Funcția principală a procesului de scanare BGP este organizarea BGP. Mai exact, scanerul BGP efectuează scanări periodice ale BGP RIB pentru a determina dacă prefixele și atributele ar trebui să fie șterse și dacă hărtile de rutare sau cache-urile de filtrare ar trebui actualizate. Acest proces scanează, de asemenea, IP RIB pentru a se asigură că toate hop-urile

BGP următoare sunt incă valabile. Dacă următorul hop nu este accesibil, toate intrările BGP care utilizează acest hop sunt eliminate din BGP RIB.

Scanarea generală se efectuează la fiecare 60 de secunde. De asemenea, scanerul BGP acceptă comenzi de la CLI prin parser pentru a schimba timpul de scanare.

Figura 3 este un prinț-screen al proceselor BGP și al utilizării memoriei într-un router Cisco 12000. Coloana Allocated arată numărul total de octeți alocați de la crearea procesului. Coloana Freed furnizează numărul de octeți folosiți de la crearea sa. Coloana Holding afisează memoria reală care este consumată de proces în acest moment.

În acest exemplu, procesul BGP router deține mai mult de 34 MB de memorie, în timp ce BGP I / O și BGP Scanner dețin 6 KB fiecare.

```
router#show process memory | include BGP

PID TTY  Allocated      Freed    Holding    Getbufs    Retbufs Process
  99   0  171331064   28799944   34023220          0          0 BGP Router
 100   0     131064   22748136       6796          0          0 BGP I/O
 101   0          0    6814116       6796          0          0 BGP Scanner
```

Figura 3: Procese și utilizarea memoriei

După cum este indicat în exemplu, procesul BGP Router contabilizează majoritatea utilizării memoriei BGP (coloana Holding).

Utilizarea memoriei pentru procesele BGP I / O și BGP Scanner **este nesemnificativă**.

Trei componente importante ale procesului **BGP Router** au contat în utilizarea lui în memorie:

- ➔ **BGP RIB**
- ➔ **IP RIB** pentru prefixele BGP deja înregistrate
- ➔ **Componenta IP de comutare** pentru prefixele BGP

Informațiile deținute în BGP RIB includ intrările de rețea, intrările de căi, atributele traseului,hărtile de rutare și cache-urile din lista de filtrare.

Memoria utilizată pentru a stoca această informație poate fi afișată utilizând comanda **show ip bgp summary**. Prefixele BGP invătate în IP RIB sunt stocate în două tipuri de structuri:

- ➔ Blocurile descriptorilor de rețea (**NDB- Network Descriptor Blocks**)
- ➔ Bloc descriptor de rutare (**RDB – Routing Descriptor Blocks**)

Fiecare rută din IP RIB necesită un NDB și un RDB pentru fiecare cale. Dacă ruta este subnetată, este necesară memorie suplimentară pentru a menține NDB. Utilizarea directă a memoriei pentru IP RIB poate fi afisată utilizând comanda show ip route summary.

Al treilea element major al procesului **BGP Router** cu o cerere semnificativă de memorie este **componența de comutare IP**.

Procesul **BGP Router** necesită, de asemenea, o cantitate mică de memorie pentru funcționarea proprie, în plus fată de ceea ce este necesar pentru stocarea informațiilor de rutare;

cu toate acestea, cantitatea de memorie pentru procesul singur este de aproximativ 40 KB și, prin urmare, este nesemnificativă în comparație cu memoria totală consumată de procesul router-ului BGP.

4. Atribute de rutare BGP

Atributele căii BGP reprezintă un set de parametri care descriu caracteristicile unui prefix BGP. Deoarece BGP este în primul rând un instrument de politică de rutare, BGP folosește des aceste atribute în influențarea selecției traseului. Utilizarea eficientă a acestor atribute este esentială în proiectarea unei arhitecturi eficiente de rutare BGP.

Următoarele atribute sunt în prezent acceptate în software-ul Cisco IOS:

- ORIGIN
- AS_PATH
- NEXT_HOP
- MULTI_EXIT_DISC
- LOCAL_PREF
- ATOMIC_AGGREGATE
- AGGREGATOR
- COMMUNITY
- ORIGINATOR_ID
- CLUSTER_LIST

- **ORIGIN**

  Acest atribut indică o sursă a prefixului. Există trei origini posibile:

➔ **IGP -** sursa pentru 0
➔ **EGP -** sursa pentru 1
➔ **INCOMPLETE -** sursa pentru 3

Este preferat un prefix cu o valoare ORIGIN inferioară în timpul unei selecții a rutei. Atributul ORIGIN al prefixului este definit automat când un prefix este injectat în BGP, dar poate fi modificat prin utilizarea unei hărti de rutare. De exemplu, dacă un prefix este redistribuit în BGP utilizând comanda redistribute, atributul ORIGIN este setat la 3; dacă un prefix este injectat în BGP prin intermediul comenzii de rețea, atributul este setat la 0.

- **AS_PATH**

  AS_PATH afisează în ordine inversă sistemele autonome traversate de un prefix, plasând ultimul AS la începutul listei. Scopul principal al parametrului AS_PATH este de a oferi o prevenție împotriva iterarii pentru rutarea inter-AS. Numărul acceptat de sisteme autonome din listă este între 1 și 255. Prefixarea aceluiași număr AS listei este o metodă comună de influențare a selecției căii de intrare, deoarece calea cu lista cea mai scurtă este preferată. Patru tipuri de segmente AS din parametrul AS_PATH sunt suportate în software-ul Cisco IOS:

➔ **AS_SET**

- ➔ **AS_SEQUENCE**
- ➔ **AS_CONFED_SET**
- ➔ **AS_CONFED_SEQUENCE**

Diferența dintre SET și SEQUENCE este că lista sistemelor autonome este neordonată (în ceea ce privește sistemele autonome traversate) într-un SET și este ordonată într-o SECVENTĂ. Cele două din urmă se aplică numai căilor aflate local. În plus, acestea sunt numărate în mod diferit în alegerea traseului.

NEXT_HOP

Acest atribut definește următoarea adresă IP din punctul de vedere al localizării unui pefix în protocolul BGP. Acest lucru nu inseamnă neapărat că următorea adresă este conectată direct. În cazul în care următoarea adresă nu este imediat următoarea, este necesară o căutare recursivă a rutelor în IP RIB. Un prefix trebuie să aibă un următor hop accesibil înainte ca BGP să o considere cea mai bună alegere.

Cu alte cuvinte, următorul hop trebuie să fie un subprefix în tabela de rutare, inclusiv în 0.0.0.0/0. Există trei considerente din perspectiva cărora atributul next-hop este stabilit:

- Când prefixul este inițial injectat în BGP, următorul hop este setat de protocolul BGP care introduce prefixul. Valoarea următoarei hop depinde de modul în care este introdus prefixul. Dacă prefixul este injectat de comanda nerwork , următorul hop IGP înainte de injectare devine următorul hop BGP. Dacă următorul hop IGP nu există (cum ar fi în cazul uneo rute care indică interfața Null0), următorul hop este conexiunea BGP însăși. Dacă conexiunea locală BGP devine următorul hop, câmpul NEXT-HOP în BGP RIB este 0.0.0.0. Următorul hop din actualizările de ieșire este setat la adresa de peering locală BGP.
- Când prefixul este anunțat prin eBGP, următorul hop este setat automat la adresa IP a peerului eBGP care trimite prefixul. În cazul în care trei sau mai multe conexiuni împărtășesc aceeași rețea multiacces, totuși, conexiunea care ruteaza stabilește ruta originala pe același segment ca și următorul hop.
- Următorul hop este modificat manual prin utilizarea unei hărți de rutare sau a unei comenzi de tip "next-hop-self". Este de retinut faptul că următorul hop nu este modificat în mod implicit pentru o sesiune BGP din cadrul aceleiași AS.

- **MULTI_EXIT_DISC**

Atributul MULTI_EXIT_DISC (MED) este folosit în mod obișnuit pe legăturile inter-AS pentru a alege între mai multe puncte de ieșire / intrare în același AS vecin.

Valorile MED sunt exprimate ca valori metrice.Astel, se preferă ruta cu valoarea MED mai mică.

Câteva dintre regulile privind stabilirea parametrului MED :

- Dacă un traseu este retinut de la o ruta iBGP, routerul de frontieră elimină MED înainte de a anunta ruta către o conexiune eBGP. Pentru a forța routerul de frontieră să anunte valoarea MED într-un asemenea caz, comanda internă **set metric-type internal** a traseului poate fi configurată pentru acel peer eBGP.

- Traseele injectate local în BGP pe un router de frontieră sunt diectionate catre un eBGP peer cu MED. Valorile metrice sunt determinate după cum urmează:
    - Dacă ruta BGP injectată, utilizând comenzile **network** sau **redistribute**, este dintr-un IGP, BGP MED este derivată din IGP.
    - Dacă ruta BGP injectată (utilizând comenzile **network** sau **redistribute**) este dintr-o rută conectată, BGP MED este setată la 0.
    - Dacă ruta este injectată prin comanda **aggregate-address**, MED nu este setată.

- **LOCAL_PREF**

LOCAL_PREF este un atribut utilizat de un router iBGP pentru a calcula un grad de preferință pentru fiecare rută externă. Este transferată între iBGP peers pentru a stabili un punct de ieșire preferat dintr-un AS. Ruta cu un LOCAL_PREF mai mare este preferată. Acest atribut nu este inclus în anunțurile prefixelor eBGP (de obicei setat administrativ în actualizările eBGP primite) și este utilizat numai în interiorul unei AS pentru manipularea selecției traseului.

În comparație, MED este trimisă de la un AS la un alt AS învecinat pe o legătură eBGP pentru a seta politica de ieșire a AS-ului primit.

- **COMMUNITY**

Parametrul **Community** este definit ca un grup de prefixe care au o proprietate comună. Mai multe grupuri pot fi folosite pe un prefix, fiecare grup fiind de 4 octeți. Există două tipuri de grupuri:

- Grupuri bine-cunoscute - Când primesc prefixe cu aceste grupuri, vecinii iau măsuri în mod automat pe baza semnificațiilor predefinite ale grupurilor. Nu sunt necesare configurații suplimentare. În **RFC 1997**, comunitățile bine-cunoscute se încadrează în intervalul de valori rezervate, care sunt de la 0xFFFF0000 până la 0xFFFFFFFF.
- Comunitățile private - Comunitățile pot fi definite de administratori și trebuie coordonate între vecinii din diferitele sisteme autonome. Acțiunile trebuie configurate în mod specific. Comunitățile private au valori în afara gamei rezervate.

În prezent, în software-ul Cisco IOS exista patru comunități bine cunoscute:

- **NO_EXPORT** – Prefixele cu această comunitate nu ar trebui să fie anunțate pentru eGBP peers, ci pot fi transmise sistemelor subautonomice din cadrul aceleiași grupari. Valoarea acestei comunități este 0xFFFFFF01.
- **LOCAL_AS** – Aceste prefixe nu tebuie directionate în afara AS-ului local. Cu gruparea, numai nodurile din același subsistem pot primi aceste prefixe.

Fără grupare, **LOCAL_AS** este tratată la fel ca **NO_EXPORT**. În **RFC 1997**, NO_EXPORT_SUBCONFED (0xFFFFFF03) este definit în acest scop.

- **NO_ADVERTISE**-Prefixele dn aceasta grupare nu trebuie directionate catre niciun nod, intern sau extern. Valoarea acestei comunități este 0xFFFFFF02.
- **INTERNET**- Acestea sunt directionate catre orice nod din comunitatea Internet. Cu alte cuvinte, nu există restricții. Această comunitate binecunoscută nu este definită explicit în RFC 1997. În software-ul Cisco IOS, comunitatea INTERNET (cu o valoare de 0) este aceea care cuprinde toate prefixele.

Comunitățile mai frecvent utilizate sunt comunitățile private. Obiectivul principal al utilizării acestora este atașarea etichetelor administrative la prefixe, astfel încât să poată fi create reguli adecvate. Comunitatea privată folosește formatul AS: număr, în care AS este numărul AS local sau un peer AS și numărul este un număr arbitrar administrat local sau cu nodurile pentru a reprezenta o grupare comunitară la care poate fi aplicată o regula.

- **ORIGINATOR_ID**

ORIGINATOR_ID este utilizat ca mecanism de prevenire a buclelor în interiorul unui AS atunci când sunt utilizate reflectorii de rută (**RR –Route Reflectors**). Acesta este creat de primul RR și nu este modificat de către următorii parametrii RR. ORIGINATOR_ID este stabilit ID-ul router-ului în oricare din următoarele situații:

- Routerul BGP care inițiază ruta în AS local, cum ar fi rutele injectate utilizând comanda **network**.
- Router-ul de frontieră BGP al aceluiași AS dacă traseul este învățat prin eBGP.

ORIGINATOR_ID are o lungime de 32 de biți și trebuie recepționat numai de la colegii iBGP. Pe un RR, **ORIGINATOR_ID** este folosit în locul identificatorului routerului în selectarea traseului. Atunci când un router iBGP primește actualizări care conțin propriul ORIGINATOR_ID, acesta elimină rutele, încălcând bucla de informații de rutare. Un router BGP nu trebuie să creeze un atribut ORIGINATOR_ID dacă acesta există deja.

- **CLUSTER_LIST**

CLUSTER_LIST este un alt mecanism de prevenire a buclelor în interiorul unui AS atunci când sunt folosiți parametrii RR. Acest atribut inregistrează lista de CLUSTER_ID-uri pe care un prefix l-a traversat într-un mediu RR. Atunci când un RR reflectă un traseu de la clienții săi către alți clienți în afara clusterului, de la non-clienți la clienți sau de la un client la alt client, propagă CLUSTER_ID local la CLUSTER_LIST. Dacă update-ul are un CLUSTER_LIST gol, RR creează unul. Folosind acest atribut, un RR poate identifica dacă informațiile de rutare sunt reluate înapoi în același cluster. Dacă CLUSTER_ID-ul local este găsit în CLUSTER_LIST, actualizarea este eliminată, incălcand buclă de informații despre rutare.

5. Protocolul BGP INTERN

BGP a fost conceput pentru a oferi o rută fără bucle de-a lungul unei serii de sisteme autonome din Internet. Mecanismul de asigurare a unei topologii fără bucle este atributul AS_PATH.

În următoarea figură sunt interconectate trei sisteme autonome. Dacă ruterul R1 din AS 65000 direcționează un prefix către R3 în AS 65001, R1 preia 65000 la lista AS_PATH pentru prefix atunci când trimite prefixul la R3. Dacă acel prefix este primit din nou de către AS 65000, un router BGP respinge prefixul, deoarece detectează o buclă în atributul AS_PATH.

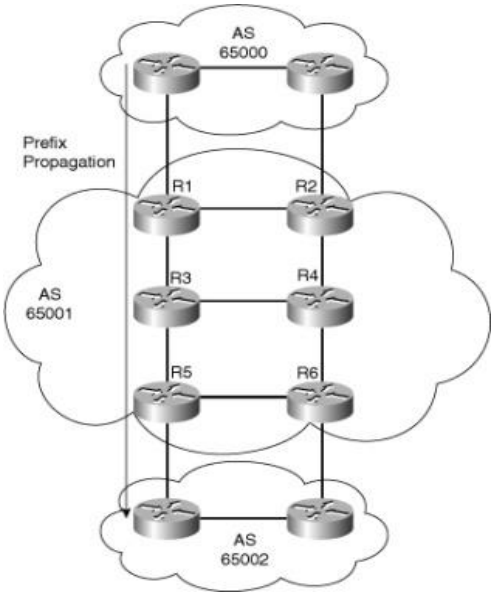

Figura 4: Propagarea prefixului intr-o topologie Multi-AS

Continuând cu Figura 4 , presupunând că R3 trebuie să propagheze prefixul la R7 în AS 65002, există câteva opțiuni pentru a realiza acest lucru.

O opțiune este ca R3 să redistribuie toate prefixele BGP în IGP, care le direcționează la R4, R5 și R6.

Apoi, R5 și R6 trebuie să redistribuie aceste prefixe în BGP și să le difuzeze către vecinii lor eBGP, respectiv R7 și R8. Există câteva probleme cu această strategie.

Parametrii IGP nu au fost concepuți să gestioneze numărul de rute care ar putea fi implicate.Tabela completă de Internet are mai mult de 100.000 de prefixe.

Actualizările periodice a informațiilor prefixate pe care multe dintre IGP-urile le solicită ar putea determina în continuare instabilitatea rețelei, consumul de resurse suplimentare al sistemului și cerințele semnificative de lățime de bandă în mod regulat pentru actualizările de rutare. Numărul crescut de prefixe are ca rezultat o probabilitate mai mare de rupere a traseului, ceea ce poate duce la probleme semnificative de stabilitate și de convergentă.

Informația BGP care este redistribuită în IGP au ca rezultat pierderea tuturor atributelor BGP, inclusiv AS_PATH. Pierderea atributului AS_PATH învinge mecanismul de prevenire a buclei BGP. De exemplu, atunci când prefixul este redistribuit înapoi în BGP în R4, același prefix este trimis înapoi la R2, deoarece AS_PATH conține numai 65001. Redistribuirea are ca rezultat pierderea oricărui atribut al politicii care a fost stabilit pentru prefixele BGP învătate.

Opțiunea preferată este utilizarea BGP internă (iBGP). Atunci când R3 direcționează prefixele către R5 prin iBGP, R3 nu adaugă propriul număr AS în AS_PATH. De fapt, software-ul Cisco IOS nu verifică nici măcar buclele AS_PATH dacă actualizările provin dintr-un peer iBGP. Fără informațiile suplimentare AS_PATH, se poate forma o buclă de informații de rutare în domeniul iBGP.

Bucla este evitată dacă R3 are permisiunea de a direcționa prefixului către R5, dar R5 nu are permisiunea de a direcționa un prefix invătat prin iBGP către un alt nod iBGP, cum ar fi R4 și R6. Cu toate acestea, această soluție necesită ca toate routerele iBGP să fie complet conectate. De exemplu, R3 este necesar să aibă sesiuni iBGP cu R4, R5 și R6. Într-un AS care are un număr mare de routere iBGP, o rețea completă poate prezenta o problemă de scalabilitate.

Utilizarea iBGP pentru a transporta informații prefixate scoate la iveală o altă problemă. Este chiar necesar un IGP dacă BGP poate transporta toate prefixele?

Un IGP este cerut în mod cert. În figură 4 , R3 nu este conectat direct la R6.

Răspunsul este că un IGP este necesar să ofere accesibilitate infrastructurii în interiorul sistemului autonom. Internal BGP nu a fost niciodată proiectat să existe fără un IGP, ci impreună cu un IGP. O rută iBGP este rezolvată adesea recursiv folosind un IGP.

6. Procesul de stabilire a rutelor

BGP trece printr-un algoritm complex pentru a determina cea mai bună cale și pentru a actualiza BGP RIB și IP RIB. Semnificația acestui lucru este cel mai bine arătată de modul în care BGP folosește atribute și alți parametri pentru a selecta cea mai bună cale.

Atunci când există mai multe căi BGP valabile pentru o anumită destinație, IOS le afisează în ordinea inversă în care au fost primite. Așadar, calea cea mai nouă este listată la început, iar cea mai veche cale este mentionată la sfârșit. Output-ul comenzii **show ip bgp**, cea mai nouă cale este listată în partea de sus, iar cea mai veche cale este listată în partea de jos. Pentru a selecta cea mai bună cale pentru o anumită destinație, BGP utilizează în general o metodă de comparare secventială. El atribuie prima cale (calea cea mai nouă) drept cea mai bună cale curentă. Apoi compară calea curentă cea mai bună cu următoarea cale din listă pană când ajunge la sfârșitul listei de căi valide. De exemplu, pentru trei căi primite secvențial-1, 2 și 3-BGP compară mai întâi căile 3 (ultima primită) și 2.

Calea rezultată cea mai buna este apoi comparată cu calea 1 (primită prima). Cea mai bună cale a celei de-a doua comparații devine calea finală cea mai bună pentru destinație.

O cale nu este un candidat valabil în procesul de selecție a celor mai bune căi dacă îndeplinește una dintre următoarele condiții:

- Atributul Next-Hop al căii este inaccesibil
- Calea nu este sincronizată și sincronizarea este activată

- Calea este refuzată de politicile BGP de intrare, iar resetarea soft-ului de intrare este configurată
- Traseul este pierdut

Selectarea traseului în software-ul Cisco IOS are în prezent 13 pași . Fiecare pas este evaluat secvențial până când se găsește o preferință:

a. **WEIGHT** este primul parametru luat în considerare. Calea cu cea mai mare valoare a parametrului este preferată. **WEIGHT** este un parametru Cisco și este local pentru routerul pe care este configurat. În mod implicit, căile locale au valoarea parametrului egală de 32768 și toate celelalte căi au valoarea egala cu 0.

b. Calea cu cel mai mare **LOCAL_PREF** este preferată. **LOCAL_PREF** implicit are valoarea 100 în software-ul Cisco IOS.

c. Rutele sunt evaluate pe baza originii, preferând calea care a fost obținută local pe router.

d. Lungimea **AS_PATH** este evaluată, preferând calea cu cea mai scurtă listă **AS_PATH**.

Se va ține cont de următoarele aspecte la evaluarea lungimii căii:

- Un **AS_SET** este numerotat cu 1, indiferent câte sisteme autonome sunt în set.
- **AS_CONFED_SEQUENCE** nu este inclus în lungimea **AS_PATH**.

e. Aici este evaluat parametrul **ORIGIN** al rutei, preferând calea cu cel mai mic tip de parametru **ORIGIN**. **IGP** este mai mic decât **EGP**, iar **EGP** este mai mic decât **INCOMPLETE**.

f. **MED** este evaluată. Calea cu cel mai mic **MED** castigă. În mod implicit, această comparație se face numai dacă primul (invecinat) AS este același în cele două căi; orice sistem subautonom este ignorat. Cu alte cuvinte, parametrii **MED** sunt comparați numai dacă primul **AS** în **AS_SEQUENCE** este același pentru mai multe căi. Orice precedent **AS_CONFED_SEQUENCE** este ignorat. Dacă comanda bgp always-compare-med este activată, parametrii **MED** sunt comparați pentru toate căile, indiferent dacă provin din același AS.

g. Căi externe (**eBGP**) sunt preferate față de căile interne (**iBGP**). Căile care conțin **AS_CONFED_SEQUENCE** sunt locale și, prin urmare, sunt tratate ca și căi interne.

h. **BGP** preferă calea cu cea mai mică valoare **IGP** pentru parametrul next-hop **BGP**. Acest pas permite luarea în considerare a topologiei locale.

i. Când ambele căi sunt externe, **BGP** preferă calea care a fost primită mai întâi (cea mai veche). Această etapă minimizează întreruperea rutei , deoarece o cale mai nouă nu înlocuiește o cale mai veche, chiar dacă aceasta este traseul preferat pe baza unor criterii suplimentare de decizie, așa cum este descris în pașii 10, 11 și 12.

j. Acest pas este sărit în oricare dintre următoarele situații:

• Comanda **bgp bestpath compare-routerid** este activată.
• ID-ul router-ului este același pentru mai multe căi, deoarece rutele au fost primite de la același router.
• Nu există cale optimă curentă. Un exemplu de pierdere a celei mai bune căi curente apare atunci când vecinul care direcționează calea este întrerupt .

k. **BGP** preferă ruta provenită de la ruterul **BGP** cu cel mai mic ID de router. Identificatorul router-ului este cea mai mare adresă IP de pe router, cu preferință acordată adreselor de tip loopback.

l. Dacă inițiatorul sau ID-ul routerului este același pentru mai multe căi, **BGP** preferă calea cu lungimea minimă a parametrului **CLUSTER_LIST**. Acest lucru este prezent numai în mediile **BGP RR**.
m. **BGP** preferă calea care vine de la adresa celui mai mic vecin. Aceasta este adresa IP folosită în configurația vecinilor **BGP** și corespunde cu cea de la distantă utilizată în conexiunea **TCP** cu ruterul local.

7. Proprietăti BGP

BGP, așa cum este definită în RFC 1771, poate transporta numai informații privind accesibilitatea IPv4 între colegii. Pentru a schimba informații de prefix de rețea, altele decât IPv4, BGP trebuie extins. Acest lucru se realizează prin schimbul de capabilitătі și prin extinderea atributelor.

7.1. Mesajele accepate de protocol

După cum este definit în RFC 1771, **BGP** acceptă următoarele patru tipuri de mesaje:

• **Open** - Acest tip de mesaj este utilizat pentru a configura conexiunile inițiale **BGP**.
• **Update** - Aceste mesaje sunt utilizate între colegi pentru a face schimb de informații despre disponibilitatea stratului de rețea.
• **Notification** - Aceste mesaje sunt utilizate pentru a comunica condițiile de eroare.
• **Keepalive** - Aceste mesaje sunt schimbate periodic între o pereche de colegi pentru a menține sesiunea.

Iată câteva dintre caracteristici care sunt suportate în software-ul Cisco IOS:

• **Capability cod 1**, **extensie Multipotocol**
• **Capability cod 2**, **Route refresh**
• **Capability cod 64**, **Graceful restart**

• **Capability cod 128**, **Old form of route refresh**
• **Codul capacității 130**, **Outbound Route Filter (ORF)**

Pentru a suporta alte adrese decât IPv4, sunt definite diferite familii de adrese (**AF) (RFC 1700**). Exemple de familii de adrese acceptate sunt IPv4 și IPv6. În cadrul fiecărei familii de adrese, identificatorii ulteriori ai familiei de adrese (**SAFI**) sunt definiți în continuare. În cadrul familiei de adrese IPv4, de exemplu, sunt definite următoarele **SAFI**:

• Unicast, codul SAFI 1
• Codul multicast, codul SAFI 2
• Etichetă IPv4, codul SAFI 4
• Etichetă VPNv4 Unicast, cod SAFI 128

Figură 5 prezintă un exemplu de capabilităti **BGP** folosind comanda **show ip bgp neighbor**. Patru capabilităti sunt schimbate (anunțate și primite): reimprospătarea rutei (formulare vechi și noi), IPv4 Unicast, etichetă IPv4 și ORF IPv4 (afișate în familia de adrese IPv4).

```
router#show ip bgp neighbor | begin cap
  Neighbor capabilities:
    Route refresh: advertised and received(old & new)
    Address family IPv4 Unicast: advertised and received
    IPv4 MPLS Label capability: advertised and received
  Received 1355 messages, 0 notifications, 0 in queue
  Sent 1354 messages, 0 notifications, 0 in queue
  Default minimum time between advertisement runs is 30 seconds

 For address family: IPv4 Unicast
  BGP table version 5, neighbor version 5
  Index 1, Offset 0, Mask 0x2
  AF-dependant capabilities:
    Outbound Route Filter (ORF) type (128) Prefix-list:
      Send-mode: advertised, received
      Receive-mode: advertised, received
  Route refresh request: received 1, sent 1
  Sending Prefix & Label
  1 accepted prefixes consume 48 bytes
  Prefix advertised 6, suppressed 0, withdrawn 0
...
```

Figura 5: Exemplificarea proprietăților BGP

Figura 6 prezintă un alt exemplu de capabilități BGP schimbate între o pereche de routere. Patru capabilităti sunt schimbate (publicate și primite): reimprospătarea rutei (formulare vechi și noi), IPv4 Unicast, VPNv4 Unicast și IPv4 Multicast. Pentru fiecare dintre cele trei familii de adrese, mai multe informații sunt furnizate în secțiunea respectivă.

```
router#show ip bgp neighbor 192.168.100.2 | begin cap
  Neighbor capabilities:
    Route refresh: advertised and received(old & new)
    Address family IPv4 Unicast: advertised and received
    Address family VPNv4 Unicast: advertised and received
    Address family IPv4 Multicast: advertised and received
  Received 1356 messages, 0 notifications, 0 in queue
  Sent 1370 messages, 0 notifications, 0 in queue
  Default minimum time between advertisement runs is 5 seconds

 For address family: IPv4 Unicast
  BGP table version 11, neighbor version 11
  Index 1, Offset 0, Mask 0x2
  Route refresh request: received 0, sent 0
  Sending Prefix & Label
  0 accepted prefixes consume 0 bytes
  Prefix advertised 10, suppressed 0, withdrawn 4

 For address family: VPNv4 Unicast
  BGP table version 1, neighbor version 1
  Index 1, Offset 0, Mask 0x2
  Community attribute sent to this neighbor
  Route refresh request: received 0, sent 0
  0 accepted prefixes consume 0 bytes
  Prefix advertised 0, suppressed 0, withdrawn 0

 For address family: IPv4 Multicast
  BGP table version 1, neighbor version 1
  Index 1, Offset 0, Mask 0x2
    Uses NEXT_HOP attribute for MBGP NLRIs
  Route refresh request: received 0, sent 0
  0 accepted prefixes consume 0 bytes
  Prefix advertised 0, suppressed 0, withdrawn 0
...
```

Figura 6: Un alt exemplu de proprietăți BGP

Figura 7 arată outputul comenzii debug ip bgp în timpul stabilirii sesiunii. În mesajul **Open**, câmpul **Capabilities** este inclus în parametrul **Option**. În câmp, toate capabilitățile acceptate sunt schimbate. Următoarele capabilităti sunt schimbate:

• Extensie multiprotocol, cod 1: IPv4 Unicast (coduri AF / SAFI 1/1), VPN IPv4 (1/128), IPv4 Multicast (1/2)
• Reimprospătarea rutei vechi, cod 128
• Reimprospătarea rutei noi, cod 2

```
*Oct 16 16:00:06.682: BGP: 192.168.100.2 went from Idle to Active
*Oct 16 16:00:06.694: BGP: 192.168.100.2 open active, delay 7887ms
*Oct 16 16:00:14.602: BGP: 192.168.100.2 open active, local address 192.168.100.1
*Oct 16 16:00:14.654: BGP: 192.168.100.2 went from Active to OpenSent
*Oct 16 16:00:14.654: BGP: 192.168.100.2 sending OPEN, version 4, my as: 100
*Oct 16 16:00:14.674: BGP: 192.168.100.2 send message type 1, length
  (incl. header) 69
*Oct 16 16:00:14.802: BGP: 192.168.100.2 rcv OPEN, version 4
*Oct 16 16:00:14.802: BGP: 192.168.100.2 rcv OPEN w/ OPTION parameter len: 32
*Oct 16 16:00:14.802: BGP: 192.168.100.2 rcvd OPEN w/ optional parameter type 2
  (Capability) len 6
*Oct 16 16:00:14.802: BGP: 192.168.100.2 OPEN has CAPABILITY code: 1, length 4
*Oct 16 16:00:14.802: BGP: 192.168.100.2 OPEN has MP_EXT CAP for afi/safi: 1/1
*Oct 16 16:00:14.802: BGP: 192.168.100.2 rcvd OPEN w/ optional parameter type 2
  (Capability) len 6
*Oct 16 16:00:14.802: BGP: 192.168.100.2 OPEN has CAPABILITY code: 1, length 4
*Oct 16 16:00:14.802: BGP: 192.168.100.2 OPEN has MP_EXT CAP for afi/safi: 1/128
*Oct 16 16:00:14.802: BGP: 192.168.100.2 rcvd OPEN w/ optional parameter type 2
  (Capability) len 6
*Oct 16 16:00:14.802: BGP: 192.168.100.2 OPEN has CAPABILITY code: 1, length 4
*Oct 16 16:00:14.802: BGP: 192.168.100.2 OPEN has MP_EXT CAP for afi/safi: 1/2
*Oct 16 16:00:14.802: BGP: 192.168.100.2 rcvd OPEN w/ optional parameter type 2
  (Capability) len 2
*Oct 16 16:00:14.802: BGP: 192.168.100.2 OPEN has CAPABILITY code: 128, length 0
*Oct 16 16:00:14.802: BGP: 192.168.100.2 OPEN has ROUTE-REFRESH capability(old)
  for all address-families
*Oct 16 16:00:14.802: BGP: 192.168.100.2 rcvd OPEN w/ optional parameter type 2
  (Capability) len 2
*Oct 16 16:00:14.802: BGP: 192.168.100.2 OPEN has CAPABILITY code: 2, length 0
*Oct 16 16:00:14.802: BGP: 192.168.100.2 OPEN has ROUTE-REFRESH capability(new)
  for all address-families
*Oct 16 16:00:14.802: BGP: 192.168.100.2 went from OpenSent to OpenConfirm
*Oct 16 16:00:14.882: BGP: 192.168.100.2 went from OpenConfirm to Established
*Oct 16 16:00:14.882: %BGP-5-ADJCHANGE: neighbor 192.168.100.2 Up
```

Figura 7: Outputul comenzii debug ip BGP

7.2. Schimbul rutelor BGP-IGP

Schimbul de rute între BGP și un IGP poate avea loc în două direcții: de la IGP la BGP și de la BGP la IGP. Există două modalități comune de a injecta rute de la un IGP în BGP:

- Utilizarea comenzii redistribute
- Utilizarea comenzii network

Rutele IGP pot fi injectate dinamic în BGP utilizând comanda redistribute.

Comanda BGP **network** operează diferit fată de o comanda IGP **network** în software-ul Cisco IOS. În majoritatea configurațiilor IGP, comanda network leagă o interfată locală la un protocol de rutare și injectează adresa interfeței în IGP.
Cu BGP, comanda **network** creează ruta în tabelul BGP numai dacă ruta este deja prezentă în tabela de rutare IP. Aceasta permite ca rutele IGP să fie injectate în BGP semistatic. Este semistatic , deoarece ruta este injectată în BGP numai atunci când există deja în tabela de rutare IP.

Redistribuirea rutelor BGP într-un IGP ar trebui să fie utilizată doar cu un mic subset din rutele BGP Internet sau atunci când numărul rutelor BGP este mic. Filtrarea corectă ar trebui implementată în timpul redistribuirii pentru a minimiza numărul de prefixe din IGP.

8. Baza informaționala de rutare

Tabelul IP RIB sau tabela de rutare IP este o bază de date critică care oferă o legătură vitală între planul de control și planul de rutare. Pe de o parte, diferite surse de rutare / protocoale precum BGP și IS-IS populează baza de rutare cu căile lor. Pe de altă parte, RIB furnizează informații pentru a construi baza de date de rutare (unele metode de comutare utilizează RIB direct pentru redirecționare).

Pe măsură ce fiecare protocol de rutare primește actualizări și alte informații, acesta alege calea cea mai bună pentru o anumită destinație și incearcă să instaleze această cale în tabela de rutare. Atunci când există mai multe căi pentru același prefix / lungime, routerul decide dacă să instaleze rutele pe baza distanțelor administrative ale protocoalelor implicate. IOS are distanțe administrative predefinite dar configurabile pentru diferite protocoale / surse de rutare. Sunt preferate prefixele dintr-o sursă de rutare care are o distanță administrativă inferioară. Rutele de rezervă sunt menținute de protocol, dacă sunt acceptate și sunt utilizate ca cele mai bune rute atunci când cele mai bune rute existente nu-și realizează scopul.

IP RIB este organizat ca o colecție de blocuri de descriptori de rețea (NDB –Network Descriptor Blocks). Fiecare NDB este o singură intrare în tabela de rutare și reprezintă un prefix de rețea obținut prin una din cele trei surse:

• O pereche de adresă / mască configurată pe o interfată locală de pe router. Aceasta devine o rută conectată, care are o distantă administrativă de valoare 0.
• Un traseu static configurat pe router. O rută statică are o distantă administrativă prestabilită de valoare 1.
• Un protocol de rutare dinamic, cum ar fi BGP.

NDB-urile conțin informații despre adresa rețelei, masca și distanța administrativă, precum și informațiile necesare pentru utilizarea protocoalelor de rutare dinamică, cum ar fi redistribuirea rutelor. Deoarece fiecare prefix dintr-un NDB poate fi atins în mod potențial prin mai multe căi, se utilizează, de asemenea, și blocuri descriptive de rutare (RDB –Routing Descriptor Blocks). Una sau mai multe RDB-uri pot fi conectate la fiecare NDB pentru a stoca informația actuală despre următorul hop. Un NDB poate avea în prezent pană la opt RDB-uri, care stabilește limita superioară a numărului de linkuri incărcate partajat pe destinație (adică opt). Deoarece NDB-urile sunt controlate de protocoalele individuale de rutare, protocoalele de rutare determină câte RDB-uri se asociază cu un NDB.

Bază de date pentru transmiterea pachetelor este construită pe baza informațiilor conținute în tabelul IP RIB și IP Address Resolution Protocol (ARP). Se efectuează o căutare pentru prefix în RIB pentru a determina adresa următorului hop și interfața de ieșire. Actualul antet Layer 2 este construit pe baza informațiilor din tabela IP ARP. Reluarea cadrelor și hărtile ATM sunt alte exemple utilizate pentru a mapa adresele Layer 3 la adresele Layer 2.

Două tipuri generale de operațiuni de căutare RIB sunt acceptate în software-ul Cisco IOS:

• Classless - Prefixul cel mai lung este căutat. Dacă nu se găsesc prefixe potrivite, se utilizează ruta implicită, dacă este prezentă.
Căutarea fără clasă IP a fost implicită (deși este incă afisată în configurația de rulare).
• Clsssful - Căutarea clasică-cea mai lungă. Supraneturile și traseul implicit nu sunt luate în considerare dacă tabela de rutare conține o subretea a rețelei principale de destinație (rețeaua clasică a adresei care este rezolvată).

9. Rutele de comutare

Sunt acceptate trei căi generale de comutare, în funcție de platformele și configurațiile hardware:
• Comutarea proceselor
• Comutarea pe cache
• Cisco Express Forwarding (CEF)

9.1. Comutarea proceselor

Procesul de comutare este cea mai elementară formă de comutare și este disponibilă universal pe toate routerele Cisco. Comutarea procesului se referă la faptul că procesorul este implicat direct în procesul necesar pentru transmiterea pachetului. Cu alte cuvinte, decizia de expediere se face printr-un proces programat de planificatorul IOS și care rulează ca peer la alte procese de pe router, cum ar fi protocoalele de rutare. Procesele care rulează în mod normal pe

router nu sunt întrerupte pentru a procesa un pachet. Pentru pachetele IP, procesul de rutare este IP Input.

Figura 8 prezintă componentele principale ale comutării tipice a procesului IP. Următoarea listă afișează procesul:

a. Un pachet IP primit de la interfața de intrare este în coadă de așteptare în memoria de pachete SDRAM – Synchronous Dynaic RAM.
b. Procesorul copiază pachetul în zona de buffer a sistemului în Dynamic RAM (DRAM), unde procesul de intrare IP începe să proceseze stratul 3 și stratul 2 al pachetului.
c. Folosind adresa IP de destinație în antetul pachetului, procesul verifică mai întâi RIB-ul pentru a determina interfața de ieșire. Apoi consultă cache-ul ARP pentru a construi antetul Layer 2.
d. În acest moment, pachetul este rescris cu noul antet Layer 2 și este copiat înapoi în memoria de pachete sau în memoria sistemului pentru a redirecționa către interfața de ieșire.

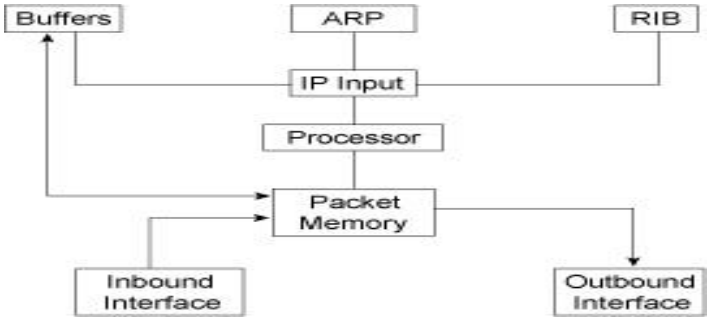

Figura 8: Comutarea proceselor IP

Comutarea proceselor este CPU-intensivă și ar putea duce la performantă scăzută a sistemului dacă un număr mare de pachete trebuie să fie examinate la nivelul procesului. Următoarele sarcini intensive din punct de vedere al procesorului sunt implicate în comutarea proceselor unui pachet IP:

• Copie a memoriei pachetului din buferul de primire către un buffer de sistem de memorie partajată.
• Căutarea tabelei de rutare . Această sarcină a devenit, în general, mai puțin o problemă de-a lungul anilor din cauza algoritmilor mai eficienți utilizați pentru a stoca informații.
• Copie a memoriei a pachetului din memoria tampon a sistemului de memorie partajată către memoria tampon de transmisie.

Limitarea procesului de comutare este exacerbată dacă routerul trebuie să gestioneze un număr mare de pachete într-o rețea instabilă, cum ar fi mediul de pe Internet. Comutarea proceselor este, de asemenea, un mecanism ineficient de comutare, deoarece informațiile despre

pachete nu sunt niciodată reutilizate. Comutarea proceselor implică efectuarea căutărilor prefixate direct în RIB, care nu este optimizat pentru căutările în tabela de rutare.

Pachetele direcționate spre router, cum ar fi pachetele BGP, sunt comutate de proces. Atunci când un pachet este destinat pentru router, procesul de intrare IP stochează pachetul pentru următorul strat superior pentru a fi prelucrat; în cazul BGP, stratul este TCP. Eficiența acestui proces poate afecta în mod direct performanța BGP. În timpul convergenței, de exemplu, TCP ar putea primi un număr mare de pachete ACK. Dacă aceste pachete nu sunt livrate către TCP la timp, este posibil ca sesiunile să nu fie stabilite.

9.2. Comutarea bazată pe cache

Comutarea bazată pe cache este un mecanism de comutare mai eficient, care folosește informațiile obținute de la primul pachet schimbat de un proces programat. În acest tip de comutare, procesul IOS care rulează în prezent pe procesor este întrerupt pentru a comuta pachetul. Pachetele sunt schimbate la cerere, în loc să fie comutate numai când se poate programa procesul de intrare IP, ca în cazul comutării proceselor.

Procesorul comută primul pachet la nivel de proces și creează o intrare în memoria cache a rutei, astfel încât pachetele ulterioare cu aceeași adresă de destinație să fie comutate pe baza intrării în memoria cache. Comutarea pachetelor pe bază cache-ului de rută necesită o procesare mai mică, ceea ce permite ca pachetul să fie comutat la nivelul întreruperii. Acesta este motivul pentru care comutarea pe cache se numește și comutare de întrerupere a contextului.

În comparație cu comutarea procesului, comutarea pe cache are următoarele avantaje:

• Comută pachetele în momentul în care ajung fără să fie nevoie să aștepte programarea procesului de redirecționare, ceea ce reduce întârzierea.
• Numai primul pachet către o destinație trebuie să fie supus comutarii procesului pentru a popula memoria cache a traseului, minimizând timpul petrecut în timpul executării sarcinilor intensive ale procesorului.
• Pachetele următoare sunt comutate pe baza informațiilor din memoria cache a traseului.

Mai multe forme de comutare pe bază de cache sunt disponibile în prezent în routerele Cisco:

- Comutare rapidă
- Comutare optimă
- Comutare optimă distribuită
- Comutare NetFlow

Rutele de comutare bazate pe cache diferă prin modul în care informațiile sunt stocate în memoria cache.

9.2.1. Comutarea rapidă

Comutarea rapidă stochează informațiile de redirecționare și șirul de rescriere a antetului MAC (noul header MAC) utilizând un arbore binar pentru o căutare rapidă și o referință rapidă.

Figura 2.6. prezintă componentele comutării rapide. Următoarea listă prezintă procesul:

a. Întrucât un pachet sosește dintr-o interfată de intrare, se efectuează o căutare pentru a determina dacă există o intrare cache pentru pachet.
b. Dacă nu există , pachetul este procesat.
c. Informațiile obținute prin comutarea primului pachet crează o intrare în memoria cache rapidă.
d. Dacă o intrare există deja când pachetul sosește, pachetul este rescris cu noua informație Layer 2 pentru interfața de ieșire și este redirecționat către interfața respectivă. Pachetul nu este copiat în bufferul de sistem, la fel ca în procesul de comutare.

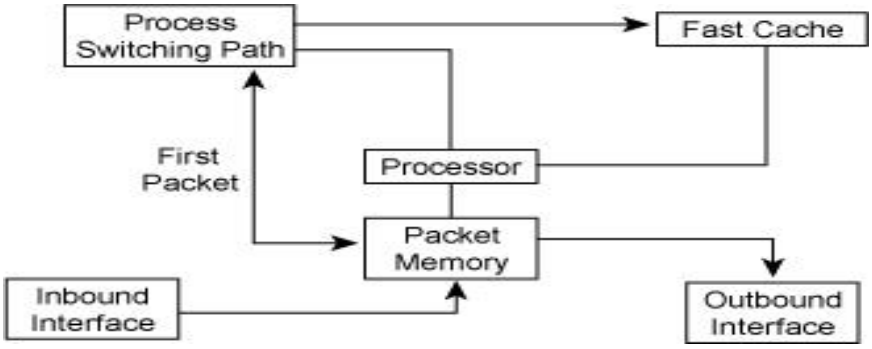

Figura 9: Comutarea IP rapidă

9.2.2. Comutarea optimă

Comutarea optimă stochează informațiile de redirecționare și informațiile de rescriere a antetului MAC într-un arbore cu 256 de direcții. Folosirea unui arbore de 256 de canale reduce numărul de pași care trebuie urmați când se cută un prefix, deși este nevoie de mai multă memorie. Comutarea optimă este acceptată numai pe platformele bazate pe procesorul de comutare a rutelor (RSP).

9.2.3. Comutare optimă distribuită

Comutarea optimă distribuită urmăreste să mute funcția de comutare a pachetelor de la procesorul principal prin mutarea deciziei de rutare la procesoarele de interfată. Acest lucru este posibil numai pe platformele de rutare care au procesoare dedicate per interfată, cum ar fi procesoare de interfață versatilă (VIP). În cazul VIP, de exemplu, cache-ul optim este populat de

RSP. Atunci când un pachet este primit, VIP incearcă să facă decizia de rutare bazată pe acel tabel. Dacă VIP poate localiza o intrare pe memoria cache a rutei locale, comută pachetul fără a întrerupe RSP-ul. Dacă aceasta nu reușește, impachetează pachetul pentru următoarea cale de comutare configurată (comutarea optimă, apoi comutarea rapidă și apoi comutarea procesului). Cu comutarea distribuită, listele de acces sunt copiate către interfața VIP, ceea ce îi permite acesteia să verifice pachetul împotriva listei de acces fără intervenția RSP.

9.2.4. Comutarea NetFlow

Comutarea NetFlow este o altă formă de comutare pe cache. Cache-ul NetFlow este construit prin procesarea primului pachet dintr-un flux prin mecanismul standard de comutare. Ca rezultat, fiecare flux este asociat cu o interfată de intrare și ieșire și cu o anumită permisiune de acces la securitate și politică de criptare. Memoria cache include, de asemenea, intrări pentru statistici de trafic care sunt actualizate prin comutarea pachetelor ulterioare.

Un flux este definit ca o conversație specifică între două gazde. Adresele sursă și destinație, porturile și tipul de pachete IP definesc un flux. Pentru comunicația TCP, o conversație începe și se oprește cu diferite mesaje de control TCP. Pentru UDP, se consideră că o conversație a încetat după expirarea unui cronometru. Pachetele următoare care se potrivesc cu eticheta de flux sunt considerate a fi membri ai aceluiași flux și sunt pur și simplu transferate prin interfața de ieșire, ocolind verificarea ulterioară a listelor de acces,de aşteptare și aşa mai departe.

Comutarea NetFlow este concepută pentru a oferi un mecanism extrem de eficient cu care se pot procesa liste de acces extinse sau complexe, fără a plăti din punct de vedere al performanței ca și alte metode de comutare. De fapt, colectarea de informații a devenit atât de importantă încât, în noile versiuni IOS, comutarea NetFlow este utilizată exclusiv în acest scop și nu mai este utilizată pentru a comuta pachete.

9.2.5. Deficiențe ale metodelor de comutare bazate pe cache

Următoarele sunt câteva dintre neajunsurile metodelor de comutare pe bază de cache:

• Toate sunt conduse de trafic, prin faptul că sunt dependente de primirea primului pachet pentru a popula memoria cache. Acest pachet este comutat în calea lentă, ceea ce duce la performanțe scăzute și la utilizarea inaltă a procesorului. Într-o rețea cu modele de trafic mari și în continuă schimbare, precum Internetul, procesarea primului pachet poate provoca degradări semnificative ale sistemului. Astfel, comutarea pe cache are probleme de scalabilitate pentru routerele core Internet. Ca un alt exemplu, eficiența comutării NetFlow depinde de lungimea fluxului. Dacă există un număr mare de fluxuri scurte, noi intrări sunt create constant, rezultând o eficiență și o performanță mai scăzute.

• Este posibil ca cache-urile să crească mai mari decât tabelele de rutare, cum ar fi atunci când există mai multe căi de cost egale. Ca rezultat, cache-ul rapid poate consuma o cantitate semnificativă de memorie.

• Comutarea pe cache nu este capabilă să efectueze partajarea incărcărilor pe pachete la un nivel de întrerupere. Deoarece comutarea pe cache se bazează în întregime pe destinație, distribuirea incărcării are loc numai pe baza unei destinații.

9.3. Cisco Express Forwarding

Atât comutarea procesului, cât și comutarea pe cache se bazează pe date sau pe cerere. Cu alte cuvinte, componentele de comutare sunt instalate numai după ce pachetele intră în router și sunt eliminate atunci când astfel de pachete nu sunt transmise de router. Dacă există un număr mare de pachete cu modele imprevizibile, performanța de comutare este degradată semnificativ. Evident, aceste căi de comutare nu sunt scalabile la nivel de Internet.

CEF a fost creat pentru a evita problemele inerente mecanismelor de comutare pe bază de cache. Acesta este proiectată pentru a se potrivi cel mai bine dinamicii rețelei de schimbare și a caracteristicilor de trafic care rezultă din creșterea numărului de fluxuri de scurtă durată asociate în mod obișnuit cu aplicațiile web și sesiunile interactive TCP.

CEF oferă următoarele avantaje:

• Scalabilitatea-CEF este bazată pe topologie și se raportează îndeaproape la tabela de rutare. CEF susține transportul asistat de hardware.
• Performantă imbunătățită-CEF este mai puțin intensivă decât procesarea în cache. Mai mult, puterea de procesare a procesorului poate fi dedicată serviciilor Layer 3, cum ar fi procesarea actualizărilor BGP.
• Resilience-CEF oferă o consistență și o stabilitate mai bună în rețelele dinamice mari. În astfel de rețele, intrările comutărilor cache rapide sunt frecvent invalide din cauza modificărilor de rutare. Aceste modificări pot determina schimbarea procesului prin folosirea tabelului de rutare, mai degrabă decât prin comutarea rapidă utilizând memoria cache a traseului. Deoarece tabela de căutare CEF conține toate rutele cunoscute care există în tabela de rutare, aceasta elimină întreținerea cache-ului de traseu.

CEF este un mecanism de comutare de tip topologie, al cărui tabel de redirecționare este legat de tabela de rutare. Ori de câte ori există modificări ale tabelului de rutare, tabela de redirecționare CEF este actualizată. În timp ce intrările sunt create, pachetele sunt comutate într-o cale de comutare mai lentă. CEF împarte funcția cache-ului rutelor în două componente principale:

• Transmiterea bazei de informare (FIB)

• Tabelul de Adiacențe

FIB

Conține toate prefixele IP din tabela de rutare. Dacă sunt menținute diferite tabele de rutare, cum ar fi într-un mediu MPLS VPN, fiecare VPN are propriul PIB. FIB nu este bazată pe date. Mai degrabă, acesta este creat și actualizat de tabela de rutare. Subsistemul FIB este

responsabil pentru a asigura că toate rutele recursive (rutele nu sunt asociate cu parametrul next-hop imediat) sunt rezolvate.

Pentru a mări consistența și a reduce timpul de căutare, FIB este organizat într-o structură de date cu mai multe căi numită mtrie. Într-o structură de date mtrie, structura arborelui este utilizată pentru a localiza datele dorite, dar datele în sine sunt stocate în altă parte. În schimb, o structură de date mtree stochează datele reale din cadrul structurii arborilor în sine. De exemplu, în memoria cache optimă de comutare, datele header-ului MAC utilizate pentru transmiterea pachetelor sunt de fapt stocate în mtree.

Două tipuri de structuri mtrie sunt utilizate frecvent în routerele Cisco:

• **8-8-8-8 :** Această formă este numită și mtrie de 256 de moduri, deoarece adresa IPv4 cu patru octeți este mapată la patru structuri pe 8 biți. Astfel, numărul maxim de căutări pentru un prefix este de patru. Acest formular este utilizat pe majoritatea routerelor Cisco.

• **16-8-8** : Aceasta este o structură mtrie pe trei nivele, unde nivelul rădăcinii are 65.536 inregistrări. Astfel, numărul maxim de căutări este de trei. Cu alte cuvinte, prima căutare rezolvă primii doi octeți și sunt necesare cel mult două căutări pentru a rezolva ultimii doi.

Fiecare nivel al structurii mtrie se numește nod. Nodul final se numește o frunză. Frunza indică spre tabelul adiacent sau către o altă structură de distribuire a incărcăturii atunci când există mai multe căi spre aceeași destinație. Conținutul IP FIB poate fi afișat cu comanda show IP CEF.

Unele dintre intrările FIB sunt următoarele:

• Attached-Prefixul este configurat pentru a fi accesibil direct prin interfață. Nu este nevoie de niciun hop IP următor pentru a crea conexiunea. Aceasta este rețeaua căreia îi aparțin interfețele locale.

• Connected-Interfața este configurată utilizând comanda de configurare a măstii adresei IP ip address. Toate intrările FIB conectate sunt atașate, dar nu sunt conectate toate intrările atașate.

• Receive-Prefixul este o adresă gazdă (/ 32) corespunzătoare uneia dintre adresele pe care ruterul le primește întotdeauna (ca gazdă). Există, în general, trei dintre acestea pe interfată: adresa de interfată reală plus subsetul all-0 și adresele de difuzare all-1.

• Recursive-Un prefix este marcat ca fiind recursiv când interfața de ieșire nu este specificată de protocolul de rutare sau de configurația statică. O intrare recursivă FIB poate fi nerezolvată atunci când nu se găseste nicio intrare FIB pentru adresa IP următoare.

Tabela de Adiacență

Se creează o tabelă pentru a conține toate next hop-urile conectate. Un nod adiacent este un nod care poate fi atins într-un singur hop. De indată ce un vecin devine adiacent, un antet de strat de legătură, numit șir MAC sau rescriere MAC, utilizat pentru a ajunge la acel vecin este

creat și stocat în tabel. Pe un segment Ethernet, de exemplu, informațiile despre antet sunt adresa MAC de destinație, adresa MAC sursă și EtherType, în ordinea respectivă.

Figura 10 prezintă un antet MAC pentru Ethernet. În acest exemplu, 00044EB31838 este adresa de destinație MAC, 0003E4BB2000 este adresa MAC sursă, iar 0800 este EtherType pentru IP.

```
router#show adjacency detail
Protocol Interface                 Address
IP       FastEthernet0/0           10.0.4.2(11)
                                   0 packets, 0 bytes
                                   00044EB31838
                                   0003E4BB20000800
                                   ARP        01:31:21
```

Figura 10: Informațiile de adiacență

De indată ce un traseu este rezolvat, acesta indică un următor next-hop adiacent. Dacă se găseste o adiacență în tabelul adiacent, în elementul FIB este memorat un indicator către adiacența apropiată. Dacă pentru aceeași destinație există mai multe căi (adică mai multe next-hop-uri sau adiacențe), la structura de distribuire a incărcărilor se adaugă un indicator pentru fiecare adiacență. Cu CEF, partajarea incărcării pe pachet este disponibilă la nivelul întreruperii.

Transmiterea incărcărilor

CEF are două forme de repartizare a incărcăturii:

• Partajarea incărcărilor pe sesiune-Aceasta este frecventă, deși incorect, numită transmiterea incărcarii per-destinație. Această formă de partajare a incărcărilor este comportamentul implicit și nu necesită o configurație specială. O sesiune este un flux de trafic care are aceeași adresă IP sursă și destinație.

• Partajarea incărcărilor pe pachete-Incărcarea este partajată pe bază de pachete . Pentru ca partajarea incărcărilor pe pachete să funcționeze corect, toate interfețele de ieșire trebuie să aibă comanda configurată.

Partajarea incărcării pe o sesiune

Permiterea partajării incărcărilor per-sesiune permite ca routerul să utilizeze mai multe căi pentru a distribui traficul. Pachetele pentru o anumită pereche de gazde-sursă-destinație sunt garantate că vor avea aceeași cale, chiar dacă sunt disponibile mai multe căi. Traficul destinat diferitelor perechi tinde să ia diferite căi. Permisiunea de impărțire a incărcărilor per-sesiune este

activată în mod prestabilit când este activat CEF. Deoarece distribuirea incărcărilor per sesiune depinde de distribuția statistică a traficului, partajarea incărcărilor devine mai eficientă pe măsură ce numărul perechilor sursă-destinație crește. Perioada de echilibrare a incărcării pe durata sesiunii poate fi utilizată pentru a se asigura că pachetele pentru o anumită pereche de gazde ajung în ordine, deoarece toate pachetele pentru aceeași pereche gazdă sunt direcționate către același link (sau linkuri).

Pentru fiecare sesiune a adreselor sursă și destinație, este alocată o cale activă. Fiecare cale are un număr egal de sesiuni incărcate. O funcție hash care utilizează adresele sursă și destinație, numărul de căi active și ID-ul router-ului este executată pentru a atribui sesiunile căilor.

Permisiunea de partajare a sarcinilor per-sesiune are o problemă potențială a polarizării traficului. Cu alte cuvinte, traficul va folosi întotdeauna același link dacă aceeași funcție hash este folosită pe toate routerele.

Pentru a vedea codul de identificare, se folosește ***show ip cef detail***.

```
router#show ip cef detail
IP CEF with switching (Table Version 5), flags=0x0
  5 routes, 0 reresolve, 0 unresolved (0 old, 0 new), peak 0
  8 leaves, 11 nodes, 12400 bytes, 8 inserts, 0 invalidations
  0 load sharing elements, 0 bytes, 0 references
  universal per-destination load sharing algorithm, id 24D5ED01
  2(0) CEF resets, 0 revisions of existing leaves
  Resolution Timer: Exponential (currently 1s, peak 1s)
  0 in-place/0 aborted modifications
  refcounts:  796 leaf, 795 node
```

Figura 11: CEF detaliat

Partajarea incărcării pe pachete

Permiterea partajării incărcărilor pe pachete permite ruterului să trimită pachete de date succesive pe căi, fără a ține seama de gazdele individuale sau sesiunile de utilizatori. Utilizează metoda round-robin pentru a determina ce cale ia fiecare pachet pană la destinație. Partajarea incărcărilor pe pachete asigură o echilibrare mai pronunțată a mai multor linkuri. Partajarea incărcărilor pe pachete este cea mai eficientă atunci când cea mai mare parte a datelor care trec prin legături paralele este pentru o singură sesiune. Permisiunea de partajare a sarcinilor în acest caz supraincarcă un singur link, în timp ce alte linkuri au un trafic foarte mic. Activarea partajării incărcării pe pachete va permite utilizarea unei căi alternative pentru aceeași sesiune ocupată. Deși utilizarea căii cu partajarea incărcărilor pe pachete este mai bună, pachetele pentru o anumită pereche de gazde-sursă-destinație pot lua diferite căi. Acest lucru poate introduce rearanjarea pachetelor, care ar putea fi necorespunzătoare pentru anumite tipuri de trafic de date care depind de pachetele care sosesc la destinație în ordine, cum ar fi traficul de voce prin IP.

10. Proprietăti de performantă a rețelei BGP

Tema performanței BGP nu se referă exclusiv la optimizarea procesului de actualizare BGP.

Eliminarea impactului de eșec al rețelei

Eroarea unui nod sau a unei legături în rețea este inevitabilă. Detectarea rapidă a defecțiunii și minimizarea impactului acesteia sunt importante pentru menținerea unei disponibilităti înalte a rețelei. Pe lângă gestionarea rapidă a eșecurilor, interacțiunea dintre IGP și BGP poate avea ca rezultat scenarii de recuperare problematică.

- Capacitatea de urmărire rapidă BGP- Caracteristica BGP oferă un mecanism pentru BGP de a identifică rapid o sesiune eBGP fără a aștepta expirarea timpului de așteptare.

- Convergența de timp IGP / BGP -Rata la care converg IGP și BGP poate crea situații în care se inregistrează pierderi de trafic. Mecanismele disponibile atât în IS-IS cât și în OSPF ajută la atenuarea acestei probleme.

- BGP Non-Stop Forwarding (NSF)-Această caracteristică se numește și repornire gratioasă. Acesta este conceput prin care se poate face restabilirea procesului BGP invizibilă pentru restul rețelei.

11. Concluzii

BGP este un protocol pe baza căruia lucrează Internetul. Datorită acestui fapt, este greu de implementat și foarte complex.

Este scalabil pentru rețele mari, însă nu este recomandat pentru rețele mici, fiind dependent de alte protocoale de rutare, cum ar fi TCP.

De asemenea este un protocol ce converge foarte greu, însă are implementate mecanisme care ajută la atenuarea problemei.

12. Bibliografie

[1] https://www.safaribooksonline.com/library/view/bgp-design-and/9781587058646/ch12.html
[2] https://www.safaribooksonline.com/library/view/bgp/9780596002541/ch12s06.html