**BGP**

פרוטוקול זה הוא מסוג EGP שאומר שהוא פרוטוקול חוץ ארגוני והיחיד מסוגו.  
לעומתו יש פרוטוקולים מסוג IGP כמו IS\_IS או OSPSF שהם רצים בתוך פרוטוקולים מסוג EGP

AS – מספר של קבוצה של נתבים שיש להם יכולת לצאת החוצה מהארגון.

* המספר של ה-AS בנוי מ-4 BYTES כלומר יש אפשרות ל 4,294,967,295
* שכנות בין נתבים באותו AS נקרא IBGP – INTERNAL, AD=20
* שכנות בין נתבים ב-AS שונים נקרא EBGP –EXTERNAL, AD=200
* כאשר יש לי שכנות בין שני נתבים ב-BGP היא רק שכנות ולא בהכרח שאני אפיץ את כלל הכתובות שלי או את כל הסגמנטים שאני מכיר
* ניתן להגדיר לנתב איזה סגמנטים להפיץ לשכני IBGP ואיזה סגמנטים להפיץ לשכני EBGP
* על מנת שנתב יעביר ניתובים משכן EBGP לנתב אחר ב-AS הוא צריך שכנות IBGP עם אותו נתב
* לפרוטוקול זה יש מספר Attributes שניתן לשנות אותם, יש בערך 13 כאלו ויש Attributes שהם קניינים לVENDOR מסוים שVENTOR-ים אחרים לא מתייחסים אליהם.
* פרוטוקול זה משתמש באלגוריתמים שונים על מנת למנוע לולאות לפני שהוא בוחר ניתוב, כל ניתוב שומר באיזה AS הוא עובר וכך הנתב הפרוטוקול יודע לא לעבור שוב באותו AS
* פרוטוקול זה תומך ב-Address Family כלומר הוא תומך בכמה קטגוריות כמו IPV4,IPV6 וכו'.
* פרוטוקול זה לא שולח הודעות HELLO על מנת להקים שכנות, השכנים מוגדרים לו סטטיים בתוך ה-PROCCESS של ה-BGP ובאחד מהשלבים של השכנות מתבצע הבדיקה של התנאים לשכנות.
* השכנות היא TCP CONNECTION בפורט 179.
* ה-TTL ב-IBGP הוא 255 דיפולטיבי
* ה-TTL ב-EBGP הוא 1 דיפולטיבי, כאשר נדרש ניתן לשנות את ה-TTL
* הודעות ב-BGP
  + OPEN – בתחילת הקמת השכנות
  + UPDATE – עדכון של BGP
  + NOTIFIACTION – שגיאות
  + KEEPALIVE – טיימר של 3 דקות שבודק האם השכן למעלה.  
    הזמן מחולק ל-3, כל 60 שניות נשלחת ההודעה על מנת לבדוק האם השכן חיי ואחרי 3 פעמים של כישלון מבין שהשכן נפול.
* מצבי שכנות
  + שלבי מצבים
    - IDLE
    - CONNRCT
    - אם CONNECT לא מצליח אז יעבור ל-ACTIVE מנסה שוב ואם לא מצליח אז חוזר לIDLE
    - אם CONNECT מצליח עובר לOPENSET
    - אם OPENSET לא מצליח עובר לIDLE
    - אם OPENSET מצליח עובר לOPEN CONFIRM
    - אם OPEN CONFIRM מצליח עובר לESTABLISHED
    - אם OPEN CONFIRM לא מצליח עובר לIDLE
  + הסבר על המצבים
    - IDLE – מקים את הקשר הראשוני והתחלת הקמת הקשר TCP
    - CONNECT – ביצוע Tree-way handshake על מנת להקים את ה-TCP CONNECTION, אחרי ששלב זה מצליח ה-TCP CONECTION יהיה למעלה ומסונכרן.
    - ACTIVE - אם CONNECT לא מצליח השכנות תהיה במצב זה למשך 60 שניות שבמהלכן הוא מנסה לבצע שוב את ה- Tree-way handshake
    - OPEN SENT – בשלב זה מתבצע הבדיקה לתנאים של הקמת שכנות
    - OPEN CONFIRM – שני אפשרויות למצב זה
      * אם קיבל KEEPALIVE אז זה מצב תקין ואני בודק שהשכן חי
      * אם קיבל NOTIFICATION כלומר הודעת שגיאה הוא יחזור לIDLE
    - ESTABLISHED – מצב שהשכנות למעלה, ומתחיל להפיץ סגמנטים במידה ומוגדר לו להפיץ.
* ניתן להגיד לנתב(1) לשלוח את הניתובי EBGP שלו שנלמדו משכן(5) אחר לשכן(2) שלו עם שה-NH זה הנתב(1) ולא נתב(5) על ידי הפקודה next-hop-self שמשנה את ה-NH של הניתוב

ROUTE REFLECTOR

* על מנת לחסוך שכנויות בAS ניתן להגדיר ROUTR שיהיה שכן (נתב 7) של כולם ויעביר ניתובים מנתב אחד לשתיים למרות שאין שכנות בין אחד לשתיים, נתב זה נקרא

ROUTE REFLECTOR וזה פקודה שמגדיר על הנתב שיהיה זה שיהיה שכן של כולם.

* נדרש להקים שכנות בין הREFLECTOR לכולם בצורה הרגילה ואז להגדיר לנתב שנרצה שיהיה REFLECTOR את הפקודה שהופכת אותו לREFLECTOR
* ניתן להגדיר יותר מנתב אחד שיהיה REFLECTOR
* כאשר ה-REFLECTOR לומד ניתוב משכן IBGP הוא יעביר אותו לשאר השכני IBGP שלו ב-AS עם NH של השכן IBGP שהעביר לו את הניתוב.

על מנת לחסוך בREFLECTORים בAS גדול נשתמש בשיטה שנקראת CONFEDERATIONS

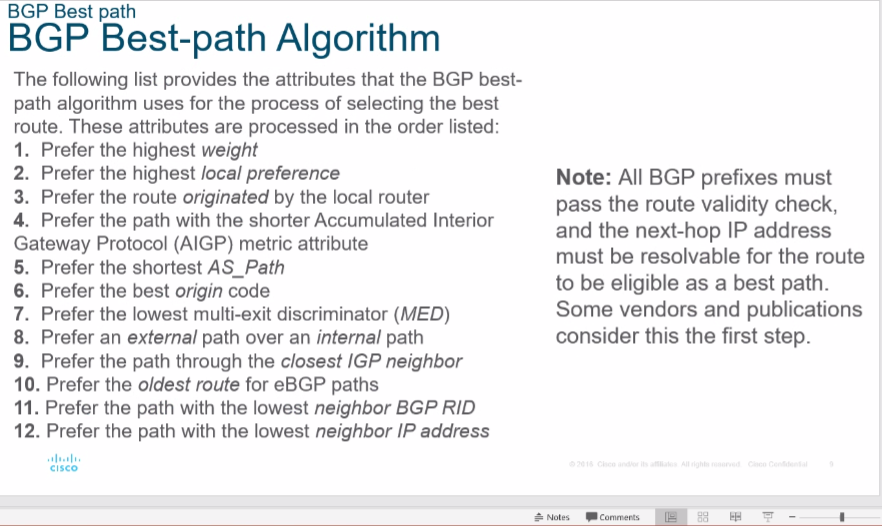
* שיטה זו מחלקת AS אחד גדול למספר AS-ים קטנים כדי לחסוך בשכנויות IBGP ובREFLECTOR-ים
* נגדיר על הנתבים בAS הגדול את הפקודה bgp confederation identifier <big as number>
* ונגדיר את המספר של הAS הקטן עםbgp confederation peer <small as number>

**MP-BGP for IPV6**

כאשר מגדירים BGP הכול אוטומטית מוגדר ל-address family של IPV4 ועל מנת לאפשר את ה-IPV6 יש לבצע את הפקודה no bgp default ipv4-unicast

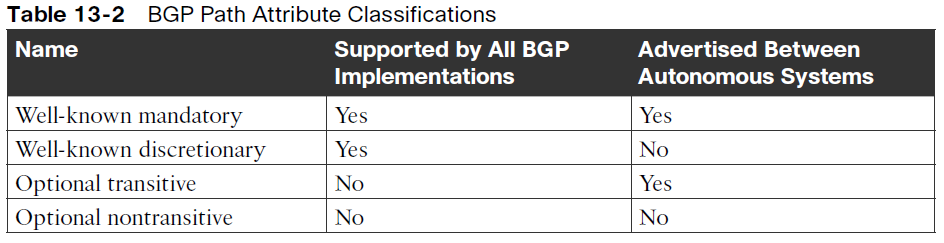
ואז יהיה ניתן להגדיר את הפקודות עם IPV6 תחת address family ipv6

יהיה ניתן גם במקביל להגדיר שכנויות של IPV4 תחת address family ipv4



**BGP Path Selection**

* בחירת הדרך הטובה ביותר מתבצעת על פי מספר ערכים, חלק מן המשתנים הם קנייניים ל-VENDOR מסוים.
* חלק מהערכים הם רק לאותו AS וחלקם נשמרים כאשר הניתוב עובר ל-AS אחר.



**Well-known mandatory** – ערכים שנתמכים בכלל ה-VENDOR-ים, ועוברים בין AS-ים שונים. לדוגמא: AS PATH

**Well-known discretionary** – ערכים שנתמכים בכלל ה-VENDOR-ים, לא עוברים בין AS-ים שונים. לדוגמא: LOCAL PREFERENCE

**Optional transitive** – ערכים שלא נתמכים בכלל ה-VENDOR-ים, עוברים בין AS-ים שונים. לדוגמא:

**Optional nontransitive** – ערכים שלא נתמכים בכלל ה-VENDOR-ים, לא עוברים בין AS-ים שונים. לדוגמא: WEIGHT(ערך זה משפיע ספציפית על הנתב ולא על כל ה-AS)

**BGP Attributes – רשימת ערכית שהם הקריטריונים לבחירת ניתוב ליעד מסויים**

**הסדר של הרשימה חשוב מאוד לבחירת הניתוב הטוב ביותר!!**

1. HIGHEST WEIGHT
   1. קנייני לסיסקו, יהיה ראשון בסדר הערכים רק בציודים של סיסקו.
   2. משפיע לוקאלי על הנתב בלבד בו הניתוב נמצא
   3. ערך מספרי בין 0 ל65535. הערך דיפולטיבי הוא 0
   4. ניתובים שאני(נתב מסוים) מפיץ יהיו עם ערך דיפולטיבי של 32768
   5. ככל שהערך שלהם יותר גבוה אז הניתוב יותר עדיף(ככה אני אעדיף ניתובים שאני הפצתי )
   6. אני יכול להגדיר ערך יותר גבוה לשכן ספציפי וכך הניתובים ליעד מסוים דרכו יבחרו על פני ניתובים לאותו יעד משכנים אחרים.
   7. ניתן להגדיר ערך אחר לסגמנט ספציפי וככה לתעדף ניתוב ליעד מסוים דרך נתב מסוים
2. HIGHEST LOCAL PREFERENCE
   1. במידה והנתב הוא של סיסקו אז אם ה-WEIGHT שווה הוא ישווה את הערך הזה
   2. נתב אשר לא קנייני לסיסקו הוא יתחיל להשוות ניתובים החל מערך זה
   3. ערך מספרי, בין 0 ל255
   4. הערך הדיפולטיבי הוא 100
   5. הערך היותר גבוה יועדף
   6. לא עובר בין AS-ים, משפיע על ה-AS שלי-> עובר מנתב לנתב
   7. כאשר אני הגדיר על נתב מסוים LP לניתוב מסוים, כלל הנתבים ב-AS יעברו דרך אותו נתב שהפיץ את ה-LP הגבוה ביותר.
3. ORIGINATED BY THE LOCAL ROUTER
   1. ניתוב אשר אני מפיץ יהיה עדיף על פני ניתוב שלמדתי מאחד השכנים
4. SHORTER AIGP METRIC
   1. לא בטוח שערך זה בשימוש מכיוון שהוא קורה במקרים מאוד ספציפיים
   2. המקרה: כאשר יש לי מספר AS-ים אשר מנוהלים על ידי אוטו ארגון ושהפרוטוקולים הIGP (בתוך ה-AS) של כל ה-AS-ים יהיו אותו דבר(4 AS שבתוך כל אחד רץ OSPF) אז ערך ה-AIGP יועבר בין ה-ASים וכך הנתבים בכל AS ידעו את החישוב של כל הדרך כולל ה-AS האחרים(כלומר הפרוטוקול שרץ בתוך ה-AS ילמד METRIC של נתבים ב-AS אחר בעזרת ה-AIGP)
   3. על מנת שערך זה יהיה רלוונטי לבחירת הניתוב זה צריך להיות מוגדר
   4. אם מוגדר בדרך אחת AIGP ובדרך אחת לא הפרוטוקול יעדיף את הדרך שבה מוגדר על פני הדרך שלא מוגדר בה בלי קשר לערך שלו.
5. SHORTER AS\_PATH
   1. ערך נצבר בין ה-ASים ותומך בכל VENDOR
   2. כאשר הניתוב עובר בכמה נתבים באותו AS אני אראה את הAS הזה פעם אחת כי ערך זה מחשב רק באיזה אזורים(AS) אני עובר ולא איזה נתבים אני עובר
6. BEST ORIGIN CODE
   1. ערך זה אומר האם הניתוב נוצר מ-IGP(i) או EGP(e) או ממשהו לא ידוע(?) והוא מופיע אחרי כל הAS-ים שאני עובר
   2. אני עובר מספר AS-ים ואז בAS האחרון אני לומד את הניתוב הזה באחת משני דרכים, IGP – מתוך הניתוב שרץ לי בAS האחרון שלי, EGP – הוזרק לAS האחרון שלי (מה שלא קורה בכלל, אולי בREDISTARBUTE), ואם לא נלמד באחד מהדרכים הללו יהיה ? שאומר incomplete
   3. IGP מועדף על פני EGP
   4. כאשר אני לומד ניתוב מהפרוטוקול IGP שחי אצלי בAS אז אני אעדיף אותו על פני הניתוב שהוזרק אלי
7. LOWEST MED
   1. זה ה-METRIC של ה-IGP (מופיע בטבלה של sh ip bgp בתור METRIC), מכיוון שהLOOPBACK מופץ בתוך ה-IGP הMETRIC יהיה 0
8. EXTERNAL PATH OVER INTERNAL PATH
   1. אני העדיף ניתוב שנלמד משכן EBGP על פני ניתוב שנלמד משכן IBGP
9. CLOSEST IGP NEIGHBOR
   1. הMETRIC היותר נמוך לNH של הניתוב BGP, הMETRIC של הפרוטוקול שרץ בתוך הAS שלי
   2. אני בודק בטבלת הניתוב שלי איך אני מגיע לNH של הניתוב (אותו אני בוחן) דרך הפרוטוקול IGP שרץ בתוך הAS
10. OLDEST ROUTE FOR EBGP PATH
    1. לכל ניתוב יש טיימר שנספר החל מהרגע שלמדתי אותו
    2. הנתב יעדיף את ניתוב יותר ישן

* ישנו ערך שנקרא minimum cluster list length
  + כאשר אנחנו משתמשים בroute-reflector הנתב שהוא הRR יכנס לרשימה שנקראת cluster-list על מנת שנוכל לדעת בכמה RR כל ניתוב עובר
  + הנתב יעדיף ניתוב עם RR נמוך יותר

1. LOWEST NEIGHBOR BGP RID
   1. הנתב יעדיף את הניתוב שמגיע מהשכן עם הROTER-ID הנמוך יותר
2. LOWEST NEIGHBOR IP ADDRESS
   1. הנתב יעדיף את הניתוב שמגיע מהשכן עם הכתובת הנמוכה יותר(הכתובת עליה מוגדרת השכנות)

**Route summarization**

הפצה של סגמנטים כסגמנט אחד.

במידה ויש לנו הפצה של כמה סגמנטים ונרצה להפיץ את כולם כסגמנט אחד אשר כולם בטווח שלו ניתן לעשות זאת בכמה דרכים:

* הפצה של הסגמנט הכולל אבל יהיה ניתן לפרסם סגמנט בטווח שלו בנוסף לסגמנט הכללי
  + Aggregate-address <network ip> <S.M>
* הפצה של הסגמנט הכללי אבל למנוע הפצה של סגמנטים אשר בטווח שלו (מונע הפצות מיותרות)
  + Aggregate-address <network-ip> <S.M> summary-only
  + במידה ונעשה את הפקודה הזו על נתב אשר הסגמנט שאנו רוצים להפיץ נלמד מAS אחר, לא נראה את כל הASים בהם הניתוב עובר. על מנת להפיץ את הניתוב ביחד עם רשימת ה-ASים שלנו נוסיף לפקודה as-set
  + Aggregate-address <network-ip> <S.M> summary-only as-set
* ניתוב שלמדתי מנתב אחר אשר הפיץ אותו על ידי SUMMERY וללא as-set יופיע עם תג של atomic-aggregate כנבצע את הפקודה: sh ip bgp <ip> <sm>

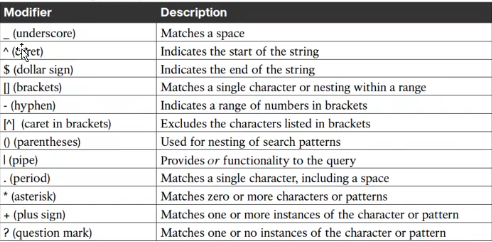
**Route filtering and manipulation**

בעולם ה-BGP יש 4 RIB-ים

* RIB-in: עדכוני ניתוב שמגיעים אלי, אחרי שניתובים נכנסים לנתב הם מגיעים לInbound route police שבו ניתן לבצע בדיקות ושינויים על הניתובים (שינוי attribute וכדומה) ומשם ממשיכים לBGP DB שנקרא גם RIB-local
* RIB-local: ה-DB של הBGP, יכול לקבל ניתובים משני מקומות
  + Inbound route police – ניתובים אשר נשלחים אלי
  + Network statement – ניתוב שאני מפיץ, הם עוברים בדיקה ע"י RIB CHECK לפני שנכנסים לRIB-LOCAL
* כאשר בsh ip bgp נראה ניתוב עם r זה אומר שהוא לא עבר את הrib-check
* אחרי הRIB-LOCAL הניתובים מעוברים לבדיקה האם הניתובים תקינים וגם בדיקת ה-NH של הניתוב
* אחרי הבדיקות מתבצעת הבחירה של הניתוב הטוב ביותר
* global-RIB: הRIB אשר בסופו משמש את הטבלת ניתוב
* RIB-out: עדכוני ניתוב אשר אני אעביר לשכנים שלי, לפני הכניסה לRIB זה הניתובים עוברים בoutbound route police שבו ניתן לבצע בדיקות ושינויים על הניתובים (שינוי attribute וכדומה)

\*\*לא ניתן לראות את rib-in ואת הrib-out

ניתן לבצע FILTERING על ידי 4 כלים:

1. Distribution list – למנוע כניסה או יציאה של ניתובים
   1. נגדיר ACL standart ובו נחסום את מה שנרצה שלא יכנס או יצא
   2. ונגדיר לכיוון השכן שנרצה את החסימה עם הפקודה:
      1. neighbor <neighbor Ip> distribute-list <ACL name/number> <out/in>
2. Prefix list
   1. נגדיר Prefix-list אשר נחסום בה את כלל הסגמנטים שנרצה שלא יכנסו או ייצאו
   2. בסוף נגדיר permit לכל שאר הדברים עם הפקודה
      1. ip prefix-list <prefix-list name> permit 0.0.0.0/0 le 32
   3. ונגדיר לכיוון השכן שנרצה את החסימה עם הפקודה:
      1. neighbor <neighbor Ip> prefix-list < prefix-list name > <out/in>
3. AS\_Path ACl/filtering
   1. 
   2. ניתן לסנן ניתובים על סמך הAS שלהם על ידי שפת איפון שנקראת REGEX
   3. ניתן להשתמש בregex בBGP בFILERING רק על ה-AS ולא על שאר הattribute-ים
   4. נשתמש בפקודה: show ip bgp regex < regex line> כדי לראות ניתובים על ידי סינון עם REGEX, היא מסננת רק את החלק של הAS-PATH בתצוגה של הפקודה (לא כולל הorigin code)
      1. ^ אומר תחילת שורה
      2. $ אומר סיום שורה
      3. \_ אומר שיש משהו בהמשך, אם נרשום לפני הטקסט יגיד שיש משהו לפני, אם נרשום אחרי טקס יגיד שיש משהו אחרי
   5. ניתן למנוע הפצה\קבלה של ניתובים בעזרת REGEX שמסנן רק את הAS-PATH של הניתוב שנכנס ויוצא, כלומר אם בסינון אמרתי לו את $300^ אז כל הניתובים שהAS-PATH שלהם הוא 300(נמצאים בAS 300) לא יופצו או יכנסו אלי משכן אשר אני אגדיר לכיוונו
      1. נגדיר AS-path access-list אשר יאפשר או יחסום סינון של AS-ים
         1. ip as-path access-list <number> deny/permit <regex line>
         2. ip as-path access-list 1 deny ^$
      2. נגדיר את הסינון מול השכן
         1. neighbor <neighbor ip> filter-list <as-path number> out/in
         2. neighbor 10.10.9.0 filter-list 1 out
4. Route map
   1. יפורט בהמשך כנושא בפני עצמו

**BGP Communities**

* Attribute נוסף של BGP
* הוא לא עובד דיפולטיבי ונדרש להפעיל את השימוש
* לקחת קבוצה של נתבים ולהחל עליהם איפון של ניתובים
* מבוסס address-family
* ישנם מספר סוגים של communities, נדבר על חלק מהם
  + Internet – להפיץ לכולם
  + Not\_Advertise – ניתובים אשר תפסתי והשמשתי עליהם את הcommunities מסוג זה לא יופצו, כלומר לא יועברו לשכנים שלי (IBGP)(תקף גם לניתובים שקיבלתי והם בcommunity מסוג זה)
  + Not\_Export – ניתובים מסוג זה לא יופצו לשכני EBGP שלי על פי הגדרה, נגדיר לו לא להוציא ניתובים בcommunities הזה לשכן ספציפי)(תקף גם לניתובים שקיבלתי והם בcommunity מסוג זה)
  + Local\_AS – כמו Not\_Export רק ב CONFEDERATIONS(חלוקה של AS למספר ASים), ואז לא יוציא לשכן EBGP שנמצא איתי ב CONFEDERATIONS למרות שהוא איתי בprivate as הוא שכן EBGP)(תקף גם לניתובים שקיבלתי והם בcommunity מסוג זה)

**Maximum Prefix**

* על מנת לא ליצור עומס על הנתב ניתן להקביל את כמות הסגמנטים שהנתב יקבל מכל שכן

**Peer groups**

ניתן להגדיר TEMPLATE של הגדרות ל-AS מסוים( remote-as, update-sourceוכ'ו) ואז הגדיר לכל שכן להשתמש בtemplate שהכנתי וככה לחסוך בהגדרות.