MPLS

מאת עמית גבאי

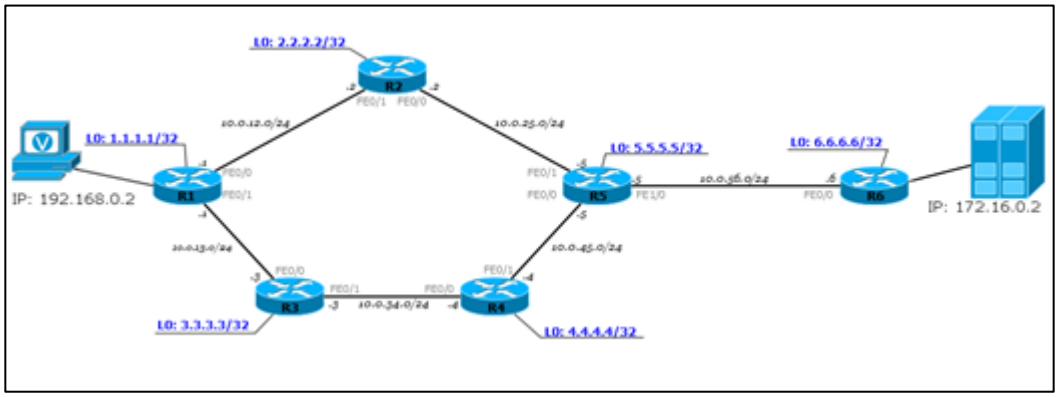
הקדמה

MPLS או **Multiprotocol Label Switching** היא טכניקת ניתוב שמאפשרת להאיץ את זרימת התעבורה ברשתות תקשורת. במקום שכל נתב יבדוק את כתובת היעד של כל חבילה בטבלת הניתוב שלו, MPLS משתמש בתוויות קצרות שמכוונות את החבילות במסלול מוגדר מראש.

עם השנים התברר כי השימוש ב-MPLS כטכנולוגיה "טהורה" - כלומר, לצורך האצת הניתוב בלבד - כמעט ואינו נפוץ. הסיבה לכך היא שהפער בביצועים מול ניתוב IP רגיל לבדו אינו משמעותי מספיק. אולם, כאשר MPLS משתלב עם שירותים נוספים, כמו VPN, ניהול איכות שירות (QoS) או Traffic Engineering, הוא הופך לכלי מרכזי בארכיטקטורות רשת מודרניות.

חשוב להדגיש: MPLS אינו בא להחליף את ניתוב ה-IP, אלא לפעול מעליו כשכבת תוויות נוספת, שמאפשרת גמישות וביצועים גבוהים בהרבה.

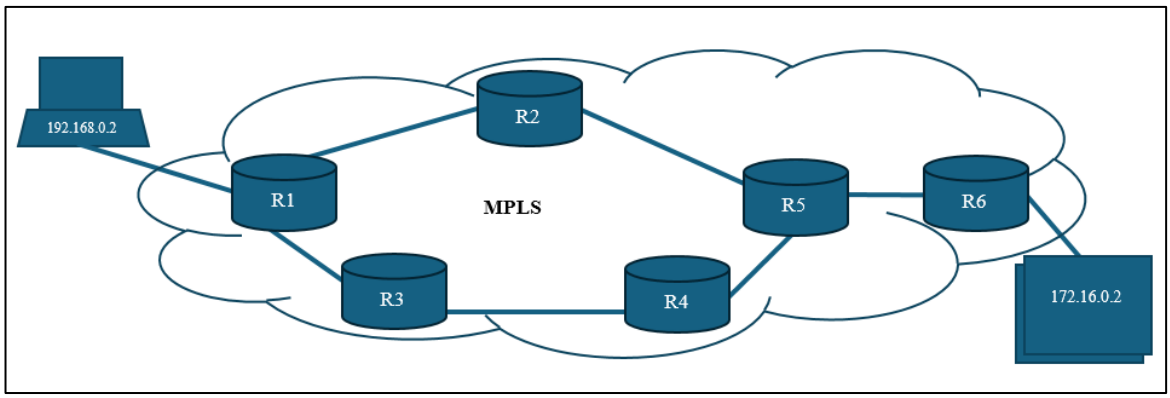
הבנת MPLS מהבסיס והצורך



[התמונה לקוחה מתוך אתר linkmeup]

כדי להבין את התרומה של MPLS, נתחיל בבחינה של התהליך המוכר לנו מרשתות IP רגילות. נשתמש בדוגמה פשוטה: מהמחשב נשלח פינג לשרת, כלומר ICMP Request ליעד שכתובתו 172.16.0.2.

לפי העקרונות הבסיסיים, החבילה תצא מ-R1 דרך הממשק FE0/0 ל-R2 ומ-R2 תמשיך ל-R3. זאת, בהתאם לבדיקה של כתובת היעד שתבוצע ע"י הנתב בעזרת טבלת ה-Forwarding Information Base (FIB) שלו, כך שתמשיך עד ליעדה. כלומר, כל נתב מחליט באופן עצמאי מה יהיה גורלה של הפקטה.



מה קורה כשפועלים תחת MPLS Domain? טבלאות של תוויות (Labels) מתמלאות מיד בנתבים ו-LSPs נוצרים. ה-Label ייכנס בין ה-IP ל-Ethernet, ברמת השכבה ה-2.5 לכאורה.

כשהפקטה מהמחשב מגיעה ל-MPLS Domain, אז הנתב הראשון מתייג לה את התוויות, ומכאן הפקטה מנותבת עד ליעד, כאשר כל נתב הבא בדרך אל היעד יסמן את הפקטה בתווית משלו, עד אשר הפקטה תעזוב את ה-MPLS DOMAIN. אז (או לפני, בהמשך נעמיק), ה-Label יוסר, ונקבל שוב פקטה שהיא Pure IP, בדיוק כפי שהגיעה מה-Client.

וזה, העיקרון הראשי של MPLS – נתבים מחליפים פקטות לפי תוויות מבלי לבצע looking לפקטת ה-MPLS, כאשר הנתב הראשון – מוסיף תווית, והאחרון – מסיר.

נבחן את הדרך שהפקטה עוברת מהמקור ועד אל היעד:

1. המחשב - ששולח פקטה לכיוון השרת המרוחק.
2. הפקטה מגיעה לנתב R1. הוא מוסיף לה במקרה הזה Label 18. פעולה הוספה זו מכונה **Push**. התווית נכנסת בין ה-IP Header ל-Ethernet.

אפשר לגזור מסקנה שזה כך מתוך טבלת ה-FIB באמצעות:

```
R1#sh ip cef detail | begin 172.16.0.0
172.16.0.0/24, version 19, epoch 0, cached adjacency 10.0.12.2
0 packets, 0 bytes
  tag information set
    local tag: 21
    fast tag rewrite with Fa0/0, 10.0.12.2, tags imposed: {18}
  via 10.0.12.2, FastEthernet0/0, 0 dependencies
    next hop 10.0.12.2, FastEthernet0/0
    valid cached adjacency
    tag rewrite with Fa0/0, 10.0.12.2, tags imposed: {18}
```

[התמונה לקוחה מתוך אתר linkmeup]

R2 מקבל את הפקטה ומסתכל על ה-Ethernet Header ומבחין שמדובר בפקטת MPLS

(שמסווג לפי EtherType 8847):

```
4457 2313.275601 192.168.0.2          172.16.0.2          ICMP      118 Echo (ping) request  id=0x0000,
▸ Frame 4457: 118 bytes on wire (944 bits), 118 bytes captured (944 bits) on interface 0
▾ Ethernet II, Src: cc:03:11:04:00:00 (cc:03:11:04:00:00), Dst: cc:04:11:04:00:01 (cc:04:11:04:00:01)
  ▸ Destination: cc:04:11:04:00:01 (cc:04:11:04:00:01)
  ▸ Source: cc:03:11:04:00:00 (cc:03:11:04:00:00)
    Type: MPLS label switched packet (0x8847)
▸ MultiProtocol Label Switching Header, Label: 20, Exp: 0, S: 1, TTL: 255
▸ Internet Protocol Version 4, Src: 192.168.0.2 (192.168.0.2), Dst: 172.16.0.2 (172.16.0.2)
▸ Internet Control Message Protocol
```

[התמונה לקוחה מתוך אתר linkmeup]

R2 קורא את התווית ומתייחס אליה בהתאם לפי טבלת התוויות שלו (Label Table).

```
R2#sh mpls forwarding-table 172.16.0.0
Local  Outgoing    Prefix           Bytes tag   Outgoing    Next Hop
tag    tag or VC   or Tunnel Id     switched    interface
18     20          172.16.0.0/24    590         Fa0/0       10.0.25.5
```

[התמונה לקוחה מתוך אתר linkmeup]

3. כעת הנתב R2 מחליט לאחר בחינת הטבלה שהפקטה צריכה לצאת עם Label=20 דרך ממשק Fa0/0. פעולת החלפת ה-Label של הפקטה נקראת **Swap**.
4. R5 מבצע פעולות זהות לזה שביצע R2, מבצע שינוי ל-Label כך שיהיה שווה 0, וקובע שהפקטה תצא דרך הממשק Fe0/1. שוב, בלי שום התייחסות ל-Routing Table, אלא רק ל-Label Table.
5. R6 מקבל את הפקטה שעטופה ב-MPLS, ומבין שצריך להסיר את ה-Label. למה? כי Label=0 הוא ערך שמור, ערך בעל משמעות (ארחיב על כך בהמשך).

בזכותו R6 יכול היה לדעת שהוא צריך לבצע את ההסרה, שנקראת גם פעולת **Pop**, ולעבור לעבודה עם IPs. הנתב מבין שהכתובת אליה מיועדת הפקטה 172.16.0.2 היא Directly Connected אליו.

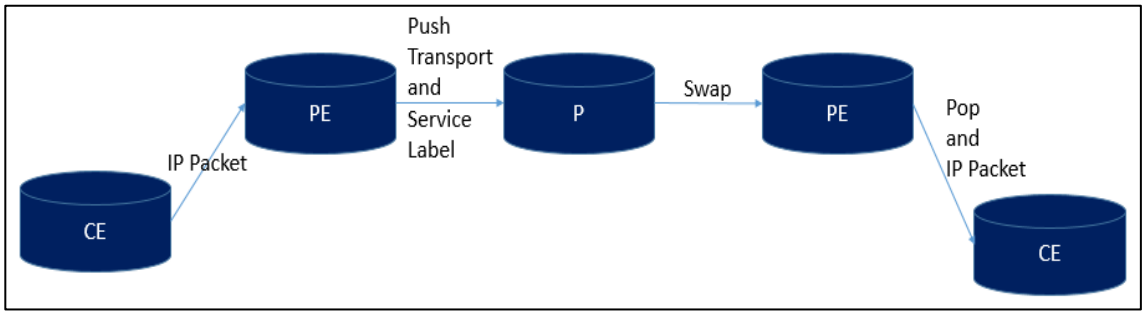
ביישומי MPLS לשירותים כגון MPLS VPN, מקובל להבחין בין שלושה סוגי נתבים עיקריים:

CE – Customer Edge - נתב הקצה של הלקוח. ה-CE אינו מפעיל MPLS ואינו מודע לתוויות. מבחינתו מדובר בחיבור IP רגיל לרשת הספק.

PE – Provider Edge - נתב הקצה של הספק. ה-PE מחבר את הלקוח אל רשת ה-MPLS, מוסיף או מסיר Service Labels (תוויות שירות) ומחזיק טבלאות נפרדות (VRF) עבור לקוחות שונים. בנוסף, ה-PE משתתף בפרוטוקולי ניתוב כמו MP-BGP, כדי להפיץ Prefix-ים של לקוחות בצירוף התוויות המתאימות.

P – Provider/Core - נתב ליבה פנימי ברשת הספק. ה-P אינו שומר מידע על Prefix-ים של לקוחות ואינו מחזיק VRFs. תפקידו היחיד הוא לבצע Label Switching על בסיס טבלת ה-LFIB - החלפה של Transport Labels לאורך המסלול.

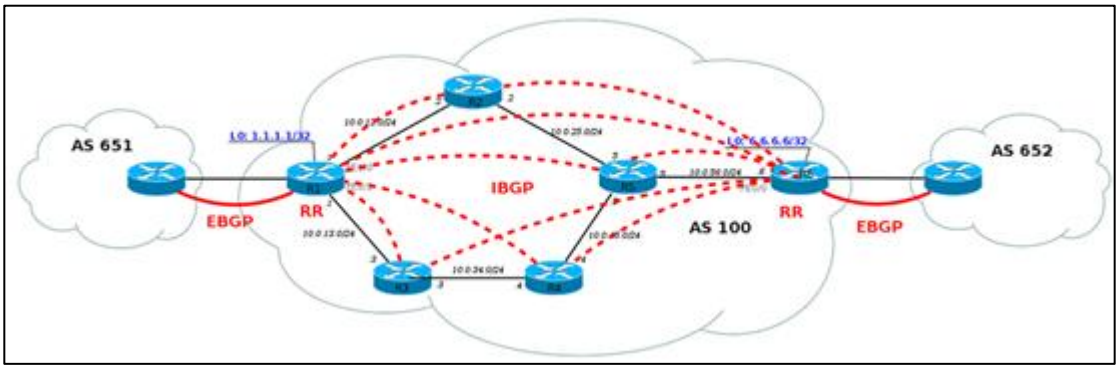
שילוב התפקידים יוצר בידול ברור: ה-CE מתייחס לרשת כסביבת IP רגילה, ה-PE אחראי על מיפוי תוויות השירות ועל ניתוב בין לקוחות, וה-P מטפל אך ורק בתוויות התעבורה. בעת שימוש במחסנית תוויות (Label Stacking), ה-P מתעסק בתווית התעבורה העליונה בעוד ה-PE מטפל גם בתוויות השירות, מה שמבטיח הפרדה יעילה בין ליבת הספק לבין שירותי הלקוח.



MPLS and BGP

על אף שאופן הפעולה שהוצג קודם נראה פשוט למדי, גם במצב הנוכחי MPLS טומן בחובו יתרונות רבים. נוכל להגדיל את מספר היתרונות אם נוסיף עבודה עם Autonomous Systems (או AS-ים בקיצור) שונים בשיתוף פעולה עם פרוטוקול כמו BGP.

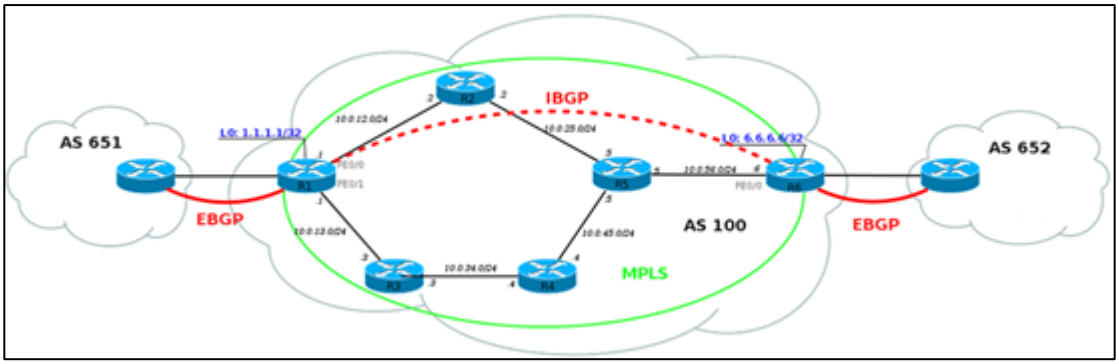
ידוע לנו, שבמצב כמו המצב המתואר בסכמה הבאה נצטרך שפרוטוקול BGP יהיה מוגדר בכל אחד מהנתבים. זאת, כדי שתתאפשר תעבורת מידע בין נתבים שונים, בין היתר בין אלה שנמצאים ב-AS-ים שונים.



[התמונה לקוחה מתוך אתר linkmeup]

ומה קורה אם נחליט לשלב את טכנולוגיית ה-MPLS?

במקרה כזה, מספיק שנגדיר את פרוטוקול ה-BGP רק על נתבים שנמצאים בקצוות ה-AS שבכל AS:



[התמונה לקוחה מתוך אתר linkmeup]

זה לא הכל - מה שנדרש מהנתבים בדרך זה בסך הכל להגיע ל-IP Address שמוגדר כ-Next-Hop. הנתבים שבדרך מ-R1 ל-R6 ישתמשו בתווית (Labels) כדי להעביר את המידע. למה? לכל נתב יש את טבלת ה--Labelים שלו, שבו הוא יודע שבכדי להגיע ל-X הוא יצמיד לפקטה את ה-Label Y, ובכדי להגיע ל-Z הוא יצמיד את ה-Label T. במקרה המתואר, הדרך מושתת כולה על אותו יעד. הנתבים בדרך לא צריכים להכיר את ה--VRFים שעומדים מאחורי כל נתב שעובד עם BGP, שכן כל תפקידם הוא לבצע פעולת Swap להעברת הפקטה עד ליעדה. מבחינת נתבי ה-**P** - לא משנה כמה ניתובי BGP יעמדו מאחורי כל נתב שעומד בקצה של ה-AS באותו מרחב MPLS-י.

מושגי בסיס

Label - ערך מספרי בטווח של 0 עד 1,048,575, המווה חלק מה-**MPLS Header**. משמשת כדי לזהות את נתיב המעבר של הפקטה (**LSP**). בהתבסס על ערך התווית, נתבי ה-**LSR** מחליטים איזו פעולה לבצע על הפקטה - **Swap/Push/Pop** - ולאן להעבירה בשלב הבא. התווית מחליפה למעשה את הצורך בבדיקת כתובת ה־**IP** של יעד הפקטה בכל תחנה בדרך.

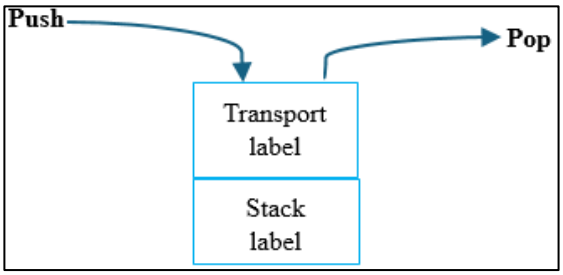
Label Stack - מדובר ב-Stack של Label-ים - יכול להיות אחד, שניים, שלושה ואפילו עשרה. פקטה שמגיעה ללא Stack משמע פקטה לא מתויגת. ה-Label שבתחתית ה-Stack נחשב כ-Level 1 Label. ה-Label שב-Top נחשב Label m. ההחלטה איזו פעולה לבצע על הפקטה מתקבלת בהתאם ל-Label שנמצא ב-Top של המחסנית, ללא תלות וקשר בשאר ה-Label-ים שנמצאים במחסנית. לכל שכבה יש תפקיד בדבר. למשל, כשפקטה מועברת, אז יהיה שימוש ב-Transit Label. קיימים כמובן סוגים נוספים, למשל Label שמעיד על שייכות הפקטה ל-VPN מסוים. מבצעים התבוננות על ה-Top Label ולפי ערכו מחליטים על:

1. ה-Next-Hop אליו תועבר הפקטה.
2. האופרציה שנבצע על ה-Label Stack.

Push - האופרציה שבה מצרפים Label לפקטת המידע ממש בהתחלה - בנתב הראשון במרחב ה-MPLS (למשל ב-R1).

Swap - פעולת ההחלפה מתחרשת בנתבי ה-Intermediate (נתבי הביניים). הנתב מקבל פקטה עם תווית אחת, ומחליף לתווית אחרת ומעביר את הפקטה הלאה. לדוגמה, המעבר בין R2 ל-R5.

Pop - פעולה שנעשית בידי הנתב האחרון טרם יציאה ממרחב ה-MPLS. הנתב מסיר את ה-Label שב-Top טרם העברה הלאה.



למעשה פעולות Push ו-Pop יכולות להתרחש בכל מקום ב-MPLS Domain, הכל תלוי בשירותים הספציפיים. הדבר הנכון יותר לומר הוא שה-Label נוסף על ידי ה-First Track Router ונמחק אחרון. כל עוד הפקטה נותרת עם תוויות, סימן שהיא פקטת MPLS.

LSR – Label switching router, כל נתב שנמצא ב-MPLS Domain.

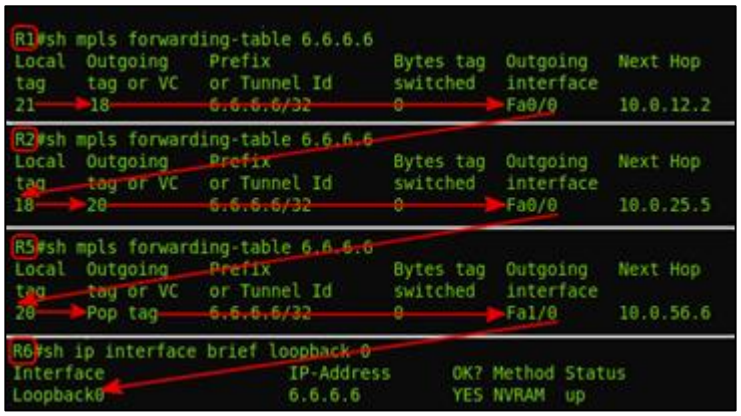
קיימים 3 סוגים של LSR:

- Intermediate ISR - הנתב האמצעי אשר מבצע את פעולת ה-Swap (כמו נתבים R1, R5).
- Ingress LSR - הנתב הראשון בקצה מרחב ה-MPLS שמבצע פעולת Push.
- Egress LSR - הנתב האחרון בקצה מרחב ה-MPLS שמבצע פעולת Pop.

LER - Label Edge Router, הנתבים שנמצאים בקצוות ה-MPLS Domain. משמע, שנתבי ה-Ingress וה-Egress נכללים תחת הגדרה זו.

LSP – Label Switched Path, הנתיב שלאורכו מתבצעת פעולת ה-Switching. מדובר במסלול חד-כיווני מה-Ingress LSR ל-Egress LSR. הנתיב שבו תעבור הפקטה לאורך מרחב ה-MPLS, שנקרא גם LSR Sequence. חשוב להבהיר את הנקודה שה-LSP הוא חד-כיווני. כלומר, מידע עובר רק בכיוון אחד, בכיוון הזה. LSP בכיוון הנגדי לאו דווקא קיים. אם קיים LSP בכיוון הנגדי זה לא אומר שהוא יעבור באותה הדרך שעבר ה-LSP בכיוון האחר - בדומה ל-GRE Tunnels.

דוגמה ל-LSP:



[התמונה לקוחה מתוך אתר linkmeup]

כל נתב LSR בדרך הזו מכיר רק את ה-Input Tags וה-Output Tags, בלי להכיר את כל הדרך שתעבור החבילה. זה אמנם נראה דומה ל-IP Routing, אבל להבדיל מהאחרון - כאן הנתיב כולו ידוע מראש. זאת אומרת, שכל LSR לא מחליט לאן תועבר הפקטה בהתאם לכתובת היעד, אלא לפי הנתיב הידוע שמוכתב על ידי ה-Ingress LSR.

FEC - אחד מהקונספטים החשובים ביותר ב-MPLS הוא ה-FEC (Forwarding Equivalence Class). בפשטות, FEC משמש לחלוקת התעבורה למחלקות (Classes). במקרה הבסיסי ביותר, מזהה ה-Class יכול להיות ה-Prefix של כתובת היעד של החבילה. עם זאת, ניתן להרחיב את ההגדרה של FEC כך שתתבסס גם על פרמטרים נוספים – למשל תגי QoS, כתובת מקור, מזהה VPN, או אפילו סוג האפליקציה.

שתי פקטות המיועדות לאותו יעד לאו דווקא יהיו שייכות לאותו FEC. לכל FEC יש את ה-LSP שלו, שנבחר במרחב ה-MPLS. למשל, בעבור WEB Surfing יקבע FEC של QoS BE, ובעוד של-VoIP יקבע FEC של EF. אפשר גם לציין שבעבור הסוג הראשון המצוין ה-LSP יהיה מגוון, ארוך ולא-מובטח. בעוד שבעבור ה-EF הוא יהיה צר באופציות, אך מהיר. הפרמטרים של החלפת ה-Label-ים מתבססים על ה-FEC. כשמידע רץ ב-LSP, אף אחד לא בוחן את ה-FEC, אלא רק את ה-Labels וה-Outgoing Interfaces. כל העבודה מבחינת ה-FEC מתבצעת בנתב אִינְטֶלֶקְטוּאָלִי, שנחשב ה-Ingress LSR. אחרי שה-Ingress מקבל פקטה "נקיה", הוא מנתח אותה, משייכה בהתאם למחלקה (Class) הרלוונטית ומשייך לה Label תואם. כל מה שנתבי ה-Intermediate LSR עושים זה פעולת Swap.

LIB (Label Information Base) - אנלוגית לטבלת הניתוב RIB (Routing Information Base). בטבלה זו מצויים צימודי הקשרים של התווית ל-Prefix. טבלה זו נכללת תחת ה-Control Plane.

LFIB (Label Forwarding Information Base) - טבלה ממנה הנתב גוזר לאיזה ממשק (פורט) צריך להפנות חבילה על סמך התווית שלה. במילים אחרות, במקום שהנתב יצטרך לבדוק את טבלת ה-FIB הוא פשוט בודק את ה-LFIB ומקבל החלטה מהירה על סמך התווית בלבד.

הרעיון האידיאולוגי החשוב שעומד מאחורי הנעת MPLS היה ביצוע פעולות מהירות, תוך שימוש מופחת מאוד ב-Control Plane וב-Data Plane לצורך ניתוב תעבורת המידע. זאת, לצד שימוש ב-Traffic Engineering מתקדם. לאורך השנים האחרונות חלה התקדמות בחומרה ומכניזם טבלת ה-FIB השתפר. שכלולים אלה הביאו לתוצאה שבה גם IP Routing נהפך מהיר ויש כאלה שכבר מעמידים בסימן שאלה את החיסכון בזמנים שמתאפשר על ידי שימוש בטכנולוגיית ה-MPLS.

כך נפגוש את החלק של MPLS בפקטה:

[התמונה לקוחה מתוך אתר linkmeup]

ה-**Label** שגודלו 20 ביטים.

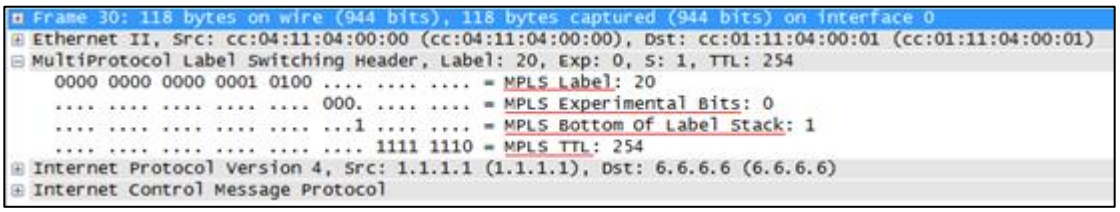
TC (Traffic control) - קובע עדיפויות לפקטות, כפי שמבצע DSCP.

מהו DSCP?

בשנת 1998 הוצג מודל Differentiated Services (DiffServ), ובמסגרתו שונה שדה ה-Type of Service או ToS ל-Differentiated Services Field או DS Field. השדה כולל 6 ביטים ראשונים ל- DSCP (Differentiated Services Code Point), שמאפשרים לסווג את התעבורה ולהקצות לה עדיפות מתאימה בכל רכיב ברשת, ושני ביטים אחרונים ל-ECN (Explicit Congestion Notification), המספקים התראה על עומס ברשת. שימוש ב-DSCP מאפשר לרכיבי הרשת ליישם Per-Hop Behavior (PHB) מותאם לכל סוג תעבורה, כגון העדפת VoIP או וידאו רגיש לעיכוב.

S – ביט אחד שמייצג את תחתית המחסנית. ה-LSR צריך להיות מודע ל-Labels שנמצאים אצלו, שהרי יש לכל Label ב-TOP של המחסנית יש השפעה על התנהגות ה-LSR מבחינת הפקטה המועברת. S יכיל את הערך 1 אם המחסנית מכילה רק Label אחד (אחרון חביב), כלומר אם הגענו לתחתית המחסנית, ו-0 אחרת – יש לפחות Label אחד במחסנית. מרגע שהוא מחליף/מסיר/מוסיף תווית ב/מה/ל מחסנית בהתאמה, ה-LSR כבר יודע מה לעשות עם הפקטה.

TTL (Time to live) – אנלוגי לחלוטין ל-IP TTL, גם מבחינת "שטח" – תופס 8 ביטים. המטרה שלו היא למנוע נדידה אינסופית של פקטה ברשת במקרה של לולאה. כשמעבירים פקטת IP דרך MPLS DOMAIN, אז הערך של ה-IP TTL יכול להיות מועתק ל-MPLS TTL ואחר כך (ביציאה מהמרחב) העתקה מה-MPLS ל-IP. לחלופין, הספירה של MPLS TTL יכולה להתחיל מ-255, ואז ביציאה ממרחב ה-MPLS ה-IP TTL ימשיך מאותה נקודה שבה הוא נעצר טרם כניסתו למרחב ה-MPLS. באופן מטאפורי, ה-MPLS Header נכלל בשכבה ה-2.5 מבחינת הטכנולוגיה שלו. אגב, לפי החלטת IETF ה-Label לא יחייב להיכלל תחת MPLS Header, אלא יכול להיות גם תחת Headers של ATM וכו', עליהם לא ארחיב.



[התמונה לקוחה מתוך אתר linkmeup]

TAG SPACE

עבור כלל התוויות יכולים להיות עד 2 בחזקת 20 ערכים שונים. עם זאת, כפי שצוין קודם, קיימים מספר ערכים שמורים שאינם ניתנים לשימוש חופשי:

0 – IPv4 Explicit Null Label – Explicit Empty Label – נעשה בזה שימוש כְּפֶסַע לפני היציאה ממרחב ה-MPLS, כלומר ב-HOP שלפני ה-Egress LSR. תווית זו יכולה להיות מוסרת גם ללא צפייה בטבלת התוויות – LFIB.

בעבר, ה-Label שמצביע על הערך 0 לא תמיד היה ה-Label האחרון במחסנית. בעקבות דרישה ולחץ שהופעל, השיטה שונתה, וה-Label עם הערך הזה הוא לאו דווקא ה-Label האחרון שנותר במחסנית.

2 - IPv6 Explicit NULL Label – דרך פעולה הזהה ל- Label 0 , רק מותאם לגרסת ה-IP.

4-15 שמורים. לא ארחיב עליהם במסגרת מאמר זה.

ערכי ה-Label-ים יכולים להשתנות בהתאם ליצרן.

קיימים פרוטוקולים ייעודיים אשר אחראיים על הפצת התוויות בין ה-Egress LSR ל-Ingress LSR, וע"י כך נבנה הלכה למעשה ה-LSP. על הפרוטוקולים בקצרה:

- LDP – הפשוט ביותר מביניהם, וזה שפועל בדרך המובנת ביותר מביניהם. מסתמך על מידע של הניתוב.
- RSVP-TE – התפתחות של הפרוטוקול הלא-פופולארי RSVP, שמשתמשים בו לצורך בניית LSPs שעונים על תנאים מסוימים. פרוטוקול זה תלוי בפרוטוקול IGP, שתומך ב-Traffic Engineering (כמו OSPF או ISIS).
- MBGP – מרוחק מעט מבחינה אידיאולוגית משני הפרוטוקולים הקודמים, שכן משמש להפצת תוויות במסגרת תכלית שונה. לא ארחיב עליו במסגרת מסמך זה.

Label 0 (Explicit Null) Vs. Label 3 (Implicit Null) – Pipe Vs. Short Pipe

ב-MPLS קיימות שתי שיטות עיקריות לטיפול בתווית האחרונה במסלול, רגע לפני שהפקטה מגיעה לנתב ה-Egress. הבחירה ביניהן קובעת האם נשמרים ערכי ה־QoS עד התחנה הסופית או לא.

Pipe – שימוש ב-Label 0 (Explicit Null)

בגישה זו הנתב שלפני ה-Egress, כלומר נתב ה-PHP, לא מסיר את התווית האחרונה, אלא מחליף אותה ב-Label 0. פעולה זו מאותתת ל-Egress: "הסרת התווית היא באחריותך". המשמעות היא שה-Egress עדיין רואה את הפקטה עם Header של MPLS, כולל ערכי QoS, ורק אז מסיר את התווית. כך נשמרת שליטה מלאה ב-QoS עד סוף הדרך.

שיטה זו חיונית כאשר רוצים לוודא שגם הנתב האחרון יעבד את הפקטה לפי ההעדפות המצוינות ב- MPLS Header - לדוגמה, בשירותים רגישים כמו קול או וידאו.

Short Pipe – שימוש ב-Label 3 (Implicit Null)

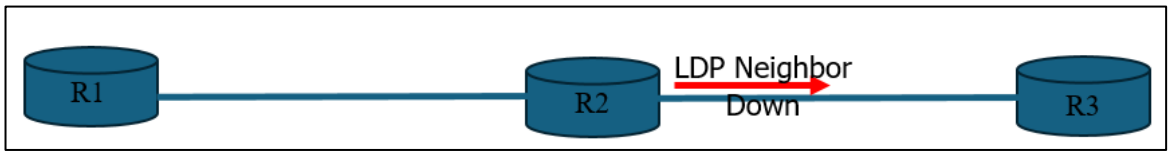
בברירת המחדל, נתב ה-PHP מסיר את התווית האחרונה בעצמו באמצעות Label 3. כך ה-Egress מקבל כבר פקטת IP "טהורה" (או פרוטוקול אחר), בלי Header של MPLS. שיטה זו יעילה יותר חישובית, אבל מוותרת על שימור ערכי QoS של MPLS עד הסוף. נעדיף אותה כאשר אין צורך ב-QoS מבוסס MPLS, או כאשר הערכים כבר הועתקו מראש לשדה אחר (כמו DSCP).

אם אסכם:

- Label 0 - Explicit Null – שימור QoS עד ה-Egress, כלומר Pipe.
- Label 3 - Implicit Null – הסרת התווית מוקדמת יותר אצל ה-PHP, כלומר Short Pipe.
- ברירת המחדל היא שימוש ב-Implicit Null, אך במקרים של שירותים קריטיים או דרישות QoS קפדניות נעדיף Explicit Null.

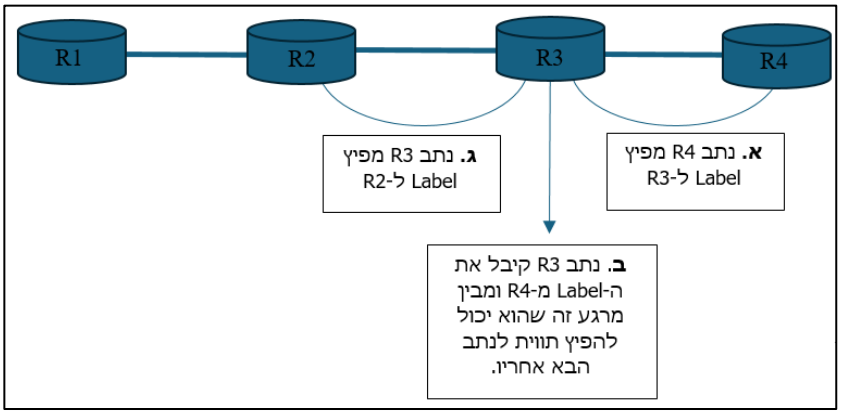
מנגנוני הפצה (LSP Distribution Modes)

במצב של **Du (Down Unsolicited)** גם אם השכנות בין R2 ל-R3 מבחינת LDP תהיה למטה, R2 עדיין יפיץ ל-R1 את התווית/Label, דבר שיגרום ל"חור שחור".



זאת, לעומת RSVP שבו ה-LSP חייב להיות בנוי מקצה לקצה (כולל כל נתבי ה-LSR בדרך). במצב של **DoD (Downstream On Demand)** רק מתי שהנתב יבקש תווית, הוא יקבלה. כלומר, R2 לא היה מפיץ את התווית ל-R1 עד ש-R1 היה מבקשה.

שיטות בקרה (LSP Control Modes)



ב-**Ordered Control** רק כשהנתב הבא יפיץ אליו את ה-Label, הוא יידע שגם הוא יכול להפיץ Label לנתב הבא אחריו. לעומת זאת, ב-**Independent Control** יש היכרות עם ה-FEC (דרך פרוטוקולי ניתוב), אך ללא הקצאת Label לשכנים באותו סדר כרונולוגי מה-Egress LSR ל-Ingress LSR.

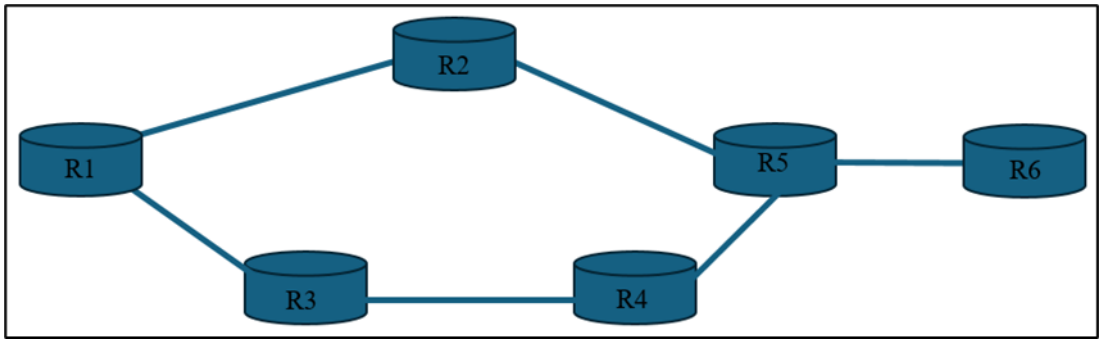
כלומר, נתב R2 , שמכיר את ה-FEC יקצה Label ל-R1 ו-R3 (לשכניו) לפני ש-R4 הקצה Label ל-R3. בזה עושים שימוש בעיקר בשילוב עם DU (Downstream Unsolicited).

מדיניות שמירת תוויות (LSP Retention Modes)

חשוב להבין כיצד ה-LSR מתמודד עם התוויות שמועברות אליו, התוויות שהוא לומד.

לדוגמה, במקרה שבו R1 מקבל תווית (Label) 20 משכנו R3, האם הנתב צריך לאחסן מידע על ה-Label הזה? שהרי לא מדובר בדרך הטובה ביותר להגיע ל-FEC (שהיא ב-R6).

זה נראה כך:



השאלה הזו מקבלת תשובה בעזרת קביעת ה-**Tag Retention Mode**:

Liberal Retention Mode – תוויות נשמרות. במקרה ש-R3 הופך להיות הצעד הבא (ולא R2), כלומר במצב שבו יש בעיות בניתוב הראשי, אז המידע יופנה מחדש דרך R3 וזה יקרה במהירות, כי ה-Label כבר שמור. עם זאת, החיסרון הוא במספר התוויות הגבוה שנאלץ להיות מאוחסן בנתב.

Conservative Label Retention – ה-Label העודף נזרק ברגע שמתקבל. פעולה/דרך זו מקטינה את מספר ה-Label-ים השמורים, אך התגובה תהיה איטית יותר במקרה שתהיה בעיה בדרך הראשית בה עובר המידע.

פרוטוקולי הפצת תוויות LDP

פרוטוקול שהשם שלו אומר הכל - **Label Distribution Protocol**.

לאחר הגדרתו בנתבי ה-LSR הוא מפיץ הודעות UDP דרך כל הממשקים שהפרוטוקול מוגדר בהם לכתובת 224.0.0.2 בפורט הלוגי 646 (היכן ש-LDP מופעל), וכך החיפוש אחר השכנים קורם עור וגידים. הודעות Hello אלה מופצות תחת TTL=1. הווה אומר, שהשכנים ב-LDP חייבים להיות Directly Connected.

אם כי יש לציין שאין זה תמיד המקרה – LDP יכול להיות מוגדר ממספר ממטרות ושכנות גם יכולה להיות מרוחקת (TTL>1) – במה שזוכה לכינוי tLDP - Targeted LDP

LDP קם על בסיס פרוטוקולי IGP כמו OSPF ו-IS-IS. לכן, הוא יוכל לפעול רק בממשקים שכבר הוגדרו לפעול באחד מהפרוטוקולים הללו, ושתומכים בטכנולוגיית ה-MPLS.

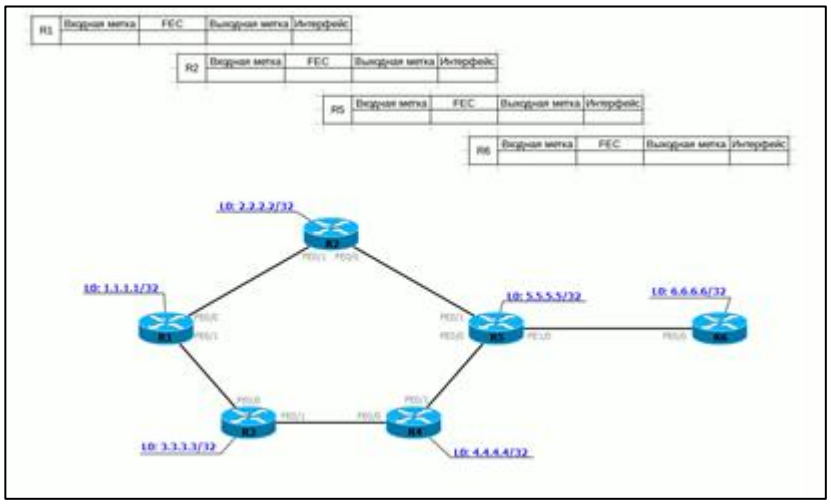
מצב ה-Initialization - השכנים מתגלים בעזרת הודעות LDP discovery-message ומבססים TCP Connection ביניהם תחת אותו פורט לוגי. הודעות נוספות (מלבד הראשונה שהוזכרה) מועברות בין השכנים עם TTL=255.

הודעות KeepAlive מועברות באופן רציף בין נתבי ה LSR-דרך חיבור ה TCP-כדי לוודא שהחיבור אכן פעיל.

לאחר שה-LSR מזהה את ה-FECים הרלוונטיים ליעדים הסופיים באמצעות פרוטוקולי ניתוב (למשל OSPF או IS-IS), הוא מתחיל להחליף עם השכנים הודעות המכונות Mapping Label Messages, שמטרתן להקצות Label עבור כל FEC רלוונטי. חשוב להדגיש, כי גם אם הנתב אינו מחובר ישירות ל-LSR Egress, הוא לומד את ה-Label המתאים עבור ה-FEC דרך שרשרת השכנים בדרך ל-Egress, כך שההקצאה מתבצעת לאורך כל הנתיב. אופן ההעברה תלוי ב-Label Distribution Mode (שהוזכרו קודם לכן) שהוגדר עבור כל FEC.

כך הנתבים ימלאו את טבלאות התוויות שלהם ומעבירים בהתאם מידע רלוונטי לנתבים "הגבוהים" יותר בשרשרת, כדי שאלה גם יוכלו "להצמיד" תווית רלוונטית ל-FEC.

בעת ובעונה אחת שבה נבנה LSP מ-R6 ל-R1 דרך R5 ו-R2 , נבנה גם LSP מ-R6 ל-R1 דרך R4 ו-R3.



[התמונה לקוחה מתוך אתר linkmeup]

כפי שניתן להבין, R6 הוא לא הנתב היחיד ששולח FEC על 6.6.6.6 (לצורך הדוגמה), אלא ששאר הנתבים עושים גם הם את הפעולה הזו. מכאן, ניתן לגזור שנוצרים כמויות של LSPs במהירות במרחב ה-MPLS שלנו. כתוצאה מכך, כל נתב LSR יידע על כל FEC ובהתאם על כל LSP רלוונטי כדי להגיע אליו.

LDP תומך בכל המצבים האפשריים שהוזכרו קודם לכן (כגון: DoD, DU וכו'). למעשה, השימוש ב-LDP ישתנה תחת חסותם של יצרנים שונים. לדוגמה, כולם תומכים במצב ה-DU עבור LDP, בעוד שב-Juniper ההפצה היא Order, ב-Cisco היא Independent.

אם כי, הדבר הכי חשוב לדעת על LDP זה שהוא לא תומך בפרוטוקולי ניתוב דינאמי. בהקשר פעולתו, ניתן לדמותו אנלוגית ל-(PIM Dense Mode) PIM DM – מציף את כל הרשת (תחת מרחב ה-MPLS) בתוויות. הפצה שמתבססת על מידע שמתקבל מטבלת הניתוב של ה-LSR. אם קיימות 2 תוויות עבור אותו FEC, אז הבחירה תהיה ב-LSP שהתקבל דרך הממשק הטוב ביותר. מכאן ניתן להסיק מספר דברים: האחד, זה שניתן להשתמש בכל פרוטוקול IGP שעולה על רוחנו. השני, זה ש-LDP יכול ויבנה אך ורק LSP אחד, שיהיה גם הכי טוב מבין הקיימים, ומכאן שגם לא יהיה LSP שישמש כ-Backup. השלישי והאחרון – בבואנו לערוך שינויים בטופולוגית הרשת, חשוב לזכור ש-LSP ייבנה מחדש בהתאם לטבלת הניתוב. מה שאומר, שפרוטוקול ה-IGP חייב להתכנס, ורק אז ה-LSP יקום. באופן כללי, לאחר הפעלת השימוש ב-LDP, התעבורה תזרום בדיוק כפי שהייתה זורמת קודם לכן, רק עם ההבדל המרכזי שנוסף והוא הופעת התוויות. העניין ב-LDP זה שהוא יבצע Load Balance פר Flow של מידע.

RSVP

ניהול תזרים תעבורת המידע אומר שניתן להפנות תעבורה בין נתבי LSR באופן שנכתיב, בהתאם למגבלות. במקרה של הדוגמה שלנו, לקוח שירות VPN מסוים עם רוחב פס מובטח של 100MB/S. יחד עם זאת, באותו הזמן ברשת מועבר סרטון לעשרות לקוחות שמשכירים את השימוש של אותו ה-VPN. במקרה כזה, רוחב הפס לא יישמר, וכנראה שאם לא נתערב בסיטואציה כזו, איפשהו ב-R2 נחווה עומס יתר וההתחייבות ל-100 ה-MB/S לא תהיה מתורגמת למעשים. לקוחות יכולים להיות כמונו, צרכנים פשוטים בבית, אך קיימים מקרים שבלקוחות מדובר בעצם בחברות, שמשלמות מחיר גבוה עבור שירותים אלה.

MPLS TE מאפשר לך לעבור על כל נתב LSR מהשולח ועד לנמען ולשמר בכל אחד מהם משאבים. זאת, מתאפשר בעזרת ארגון נכון של ערכי QoS לאורך כל הנתיב, במקום לאפשר לכל נתב להחליט באשר לפקטה המועברת. כמו LDP, שמו של הפרוטוקול שוב עושה שכל בהבנת תפקידו – **Resource ReSerVation Protocol**. בזכות העברת מידע תחת QoS מסוים כל נתב משמר את המשאבים הרלוונטיים.

בהערכה ראשונית מהלך העבודה של הפרוטוקול פשוט:

1. נתב המקור רוצה לשלוח תעבורת מידע של 5MB/S. לפני שהוא עושה זאת, הוא שולח בקשת RSVP כדי לשמר את רוחב הפס לנמען, הודעה שנקראת Path Message. ההודעה כוללת מספר מזהי זרם כשכל Node יכול לזהות אח"כ באמצעותם את השתייכותו לפקטות ה-IP המתקבלות, ובהתאם את רוחב הפס הנדרש שנדרש להקצות.
2. הודעת ה-Path Message עוברת מאיבר לאיבר עד לנמען. יעד ההודעה בכל איבר נקבע בהתאם לטבלת הניתוב.
3. בכל נתב שקיבל את ההודעה מתבצעת בחינה של המשאבים. במידה שיש לו מספיק רוחב פס. הוא מסגל את האלגוריתמים הפנימיים שלו כך שהוא יוכל לעבד את זרם המידע כראוי ושתמיד יהיה מספיק רוחב פס.
4. אם אין לו מספיק רוחב פס (5MB/S) (בעקבות עיסוק זרמים אחרים) הוא מסרב להקצות משאבים ומחזיר הודעה מתאימה לשולח.
5. ברגע שהודעת ה-Path Message מגיעה לנתב הנמען, האחרון משיב הודעת Resv, שמאשרת דה-פקטו שהוקצו משאבים לאורך כל המסע.
6. נתב המקור, מקבל את הודעת ה-Resv ומבחינתו הוא מבין שהכל מוכן לשליחת מידע.

למעשה, התהליכים הרבה יותר מורכבים ממה שתיארתי כאן, אך לא ארחיב אודותיהם בשלב זה.

אממה, מה שיותר מעניין אותנו הוא ההרחבה של RSVP, הלא היא RSVP TE, שפותחה במיוחד בעבור MPLS TE. המטרה של הפרוטוקול זהה כשל LDP, בסופו של דבר בניית LSP מהנמען למוען. אבל, עם ייחודיות - הדגש על הניואנסים ב-LSPs שחייבים לענות על תנאים מסוימים.

עניין דו-הכיווניות לא משתנה, כך שהמשאבים יישמרו רק בכיוון אחד (כשם ש-LSP בנוי רק לכיוון אחד). במקרה שמעוניינים לכיוון השני, חייב ליצור אותו.

תחילה, לא נעסוק בפונקציונליות שימור המשאבים, אלא נתמקד בתהליך יצירת ה-LSP.

אובייקט חדש של הודעות נוצר בעת מעבר ההודעה של Path Message מאיבר לאיבר בדרך, והוא הודעת ה-Label Request. למה הכוונה? הודעת ה-Path Message מעוררת את הנתב לבחור Label עבור ה-FEC הרלוונטי. ה-Provoke הזה מכונה Label Request.

מעתה שההודעה מגיעה אל נתב ה-Egress LSR, הנתב יצרף Label לכל הודעה שיעדה עכשיו הוא המוען, כך שהודעת ה-Resv מגיעה עד למוען כשתוויות הופצו לאורך כל ה-LSP החדש שנוצר.

כלומר הבקשה של התוויות נעשית ב-Downstream (ב-Path Message מהשולח לנמען) וה-Labels מועברים ב-Upstream (ב-Resv מנמען לשולח).

מכאן, חשוב שנתייחס להבדל מרכזי בין LDP ל-RSVP-TE – בניגוד ל-LDP, ה-RSVP-TE עובד בהסתמך על תוצאות עבודתם של פרוטוקולי ניתוב דינאמיים ומסגל אותן לעצמו.

ראשית, נדרשת עבודה רק עם פרוטוקולי Link-State. דרישה המותירה אותנו עם פרוטוקולי OSPF ו-ISIS. שנית, עבודתם יחד עם RSVP TE מציגה אלמנטים חדשים בפרוטוקולים – באשר ל-OSPF LSA זה Opaque LSA, ובאשר ל-ISIS מתווספים TLV IS Neighbor ו-IP reachability. לא ארחיב באריכות על אלמנטים אלה במסמך זה. שלישית, כדי לאפשר חישוב של הנתיב בין ה-Ingress LSR ל-Egress LSR נערך שינוי מיוחד ב-SPF, כך שבמקומו אלגוריתם CSPF נכנס לשימוש.

הודעת ה-Path Message מועברת בעזרת תשדורת Unicast לכיוון ה-FEC. היינו יכולים להשתמש בטבלת הניתוב הרגילה כדי לממש את המעבר המתואר.

למען האמת, הנתיב עובר ברשת לא בעזרת FIB בכל איבר בדרך, משום שאז לא תהיה לו יכולת לספק שימור משאבים או חיפוש אחר ניתובים חלופיים. הנתיב של ה-LSP נקבע מראש על ידי ה-Ingress LSR עבור כל המסלול ל-Egress LSR. החישוב נעשה באמצעות אלגוריתם CSPF, אשר לוקח בחשבון לא רק את טבלאות הניתוב הרגילות אלא גם את מגבלות המשאבים והדרישות הספציפיות של ה-LSP, כגון רוחב פס ודרישות QoS. לאחר שהנתיב מחושב, הודעות ה-Path Message מועברות לאורך הנתיב שנבחר, כאשר כל נתב בדרך מקצה את המשאבים הדרושים ומקבל את ה-Label המתאים. כך, למרות שהמסלול עצמו נקבע מראש, ההודעות מיישמות אותו בפועל ומשמרות את המשאבים לאורך כל הנתיב.

כדי לדעת לבנות את הנתיב הזה, RSVP TE צריך להכיר את טופולוגיית הרשת שהוא משיג בעזרת עבודתם של הפרוטוקולים עליהם הוא מסתמך – OSPF/ISIS.

עם זאת, הפרוטוקולים הללו היו צריכים לעבור מעט התאמות בעבודתם עם RSVP TE. אותן הגבלות או Constraints עשויים לכלול דרישות לדוגמה כמו מינימום רוחב הפס האפשרי, סוג הקו, מספר האיברים שה-LSP חייב לעבור דרכם בדרך וכו'.

לצורך מטרה זו, ההודעות העוברות בין הנתבים של הפרוטוקולים (OSPF/ISIS) כוללות בנוסף למידע הבסיסי מידע נוסף אודות מאפייני הקווים והממשקים.

לדוגמה, OSPF הציג 3 סוגים נוספים של LSA לצורך כך:

- Type 9 – Link-local scope
- Type 10 – Area-local scope
- Type 11 – AS scope

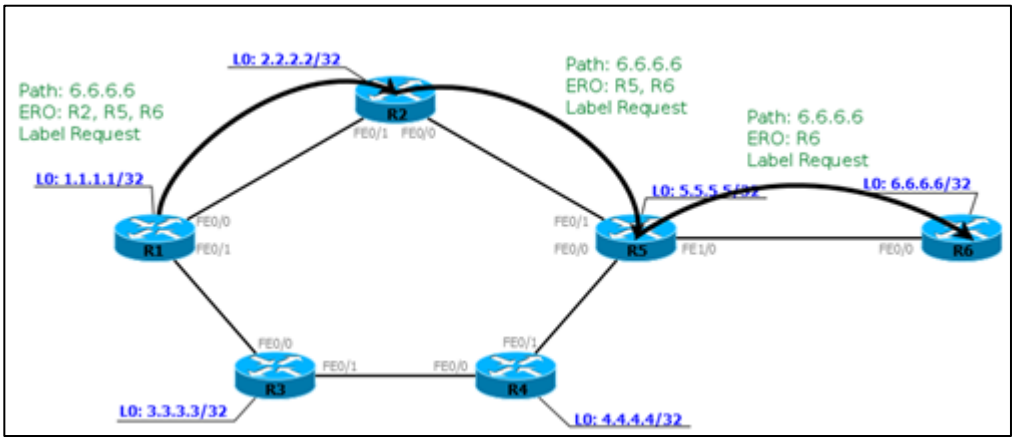
ב-Opaque מתכוונים לסוגים המיוחדים של LSA שהוזכרו לעיל, שלא נלקחים ב-OSPF כשהוא מבצע חישובים בטבלת הניתוב. הם יכולים להיות משומשים על ידי שאר פרוטוקולים בהתאם לצרכיהם. במקרה של TE, הוא עושה בהם שימוש כדי לבנות את הטופולוגיה שלו, שנקראת Traffic Engineering Database – **TED**.

ISIS עובד באותו אופן - פרסומים חדשים שלו יהיו:

- IS-IS TLV 22 – Extended IS Reachability
- IS-IS TLV 134 – Traffic Engineering router ID
- IS-IS TLV 135 – Extended IP Reachability

נעיף מבט מעט קרוב יותר על כל התהליך:

1. בנתב R1 אנו מפעילים MPLS TE ומגדירים ISIS/OSPF כדי להחליף מידע בין הנתבים שיתמוך ב-TE, כמו מידע על משאבים זמינים. בשלב הזה, TED מתבסס. RSVP עדיין מאחורי הקלעים.
2. יצרנו ממשק Tunnel שבו ציינו את סוגו (Traffic Engineering), כתובת יעד (6.6.6.6), ודרישות משאבים הכרחיים. LSR טוען את אלגוריתם ה-CSPF: צריך לחשב את הנתיב הקצר ביותר מ-R1 ל-6.6.6.6, תוך לקיחה בחשבון בהגבלות שמוצבות. בשלב הזה מתקבל הנתיב האופטימלי (רשימת איברים מהשולח לנמען).
3. התוצאה של השלב הקודם מוזנת על ידי RSVP, והופכת לאובייקט בשם ERO. R1 מרכיב את ה- RSVP Path, שם הוא מוסיף את ה-ERO. לאובייקט הזה נוסף גם אובייקט ה-Label Request, שאומר שמתי שקיבלת את הפקטה, אתה צריך לבחור Label עבור אותו FEC.
   ERO (Explicit Route Object) – הודעת RSVP Path שכוללת רשימה של האיברים שההודעה המצוינת מיועדת לעבור דרכם.
4. הודעת ה-Path Message אם כך, מועברת איבר-איבר בהתאם ל-ERO (ולא לפי ה-Routing Table, כולל מקרים שבהם ה-ERO וה-IGP יציעו דרך חופפת).
5. כש-R2 למשל, מקבל את ה-RSVP Path, הוא בודק את זמינות המשאבים הדרושים, ובמידה שקיימת – מקצה בהתאם. הודעת RSVP Path שקיבל נמחקת, הנתב יוצר אחת חדשה, ומעדכן בהתאם את ה-ERO – מסיר את עצמו מהרשימה, כדי כשההודעה תגיע לנתב הבא, שהוא לא יחזור חזרה אחורה לכיוון R2 . כך מועברת ההודעה המעודכנת לעבר האיבר הבא ברשימה.
6. R5 מבצע את אותה פעולה שביצע R2 (שלב מספר 5).



[התמונה לקוחה מתוך אתר linkmeup]

7. R6, שמבין בהתאם להודעה, שהוא "האחראי לכל האנדרלמוסיה" מוחק את הודעת ה-Path, יוצר הודעת תגובה של Resv ומצרף אליה אובייקט של Label.
   - חשוב להבין שעד כה התוויות רק הודגשו, אך לא הופצו. כעת, הן מתחילות להיות מוכרזות באמצעות נתב ה-LSR שמבקש אותן.
8. הודעת ה-Resv מתחילה לעבור איבר-איבר לאחור עד ל-Ingress LSR. ההודעה מחויבת לעבור באותם איברים שעברה הודעת ה-Path (רק בסדר הפוך כמובן).
   - שאלה עבורכם: ה-LSP שנבנה הוא מה-Ingress ל-Egress? מ-R1 ל-R6 או שמא הפוך?
   - בסיום, LSP נוסד מ-R1 ועד ל-FEC (6.6.6.6/32). המידע יכול לעבור ב-LSP הזה מ-R1 ל-R6 בלבד. בכדי לאפשר מעבר מידע בכיוון ההפוך, צריך לייסד LSP בכיוון ההפוך, הרי הוא חד-כיווני. כל הפעולות ליווסדו יהיו זהות כשם שנוסד הנוכחי.

השאלה למה יש צורך בציון כתובת ה-FEC בפקטה שמועברת בין איבר לאיבר, אם כל איבר כבר מופיע ב-ERO והדרך לכאורה כבר סלולה לחלוטין?

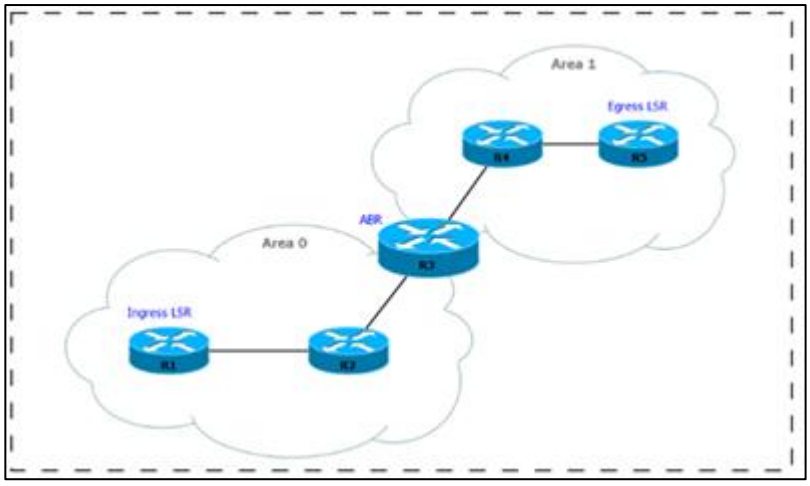
התשובה לכך טמונה בעובדה שלעיתים אובייקט ה-ERO לא יכיל את האיברים שצריך לעבור דרכם בדרך, מה-Ingress ועד ה-Egress LSR. מצב של השמטה כזה עלול להיגרם בשל "עצלנות" של נתב ה-Ingress LSR בחישוב הנתיב כולו.

בעיה מתרחשת כאשר לוקחים בחשבון את ה-IGP Zones. לשני פרוטוקולי ה-IGP, הן ISIS והן OSPF יש את אופציית החלוקה לאזורים כדי לפשט את הניתובים. ברשתות גדולות (שמונות מאות, אם לא אלפי, איברים) יש בעיה בחישוב Best Path בעזרת האלגוריתם (דבר שייקח זמן ומשאבים). לכן, האזור הגלובאלי מחולק למספר אזורי ניתוב.

כל ה"קרע" (snag) הזה מתרחש כאשר מדובר בפרוטוקול שעוסק בטופולוגיה, אם כי כשמדובר ב-Zone אז ההתייחסות לטופולוגיה היא בהתאם רק למה שנמצא בתוך אותו אזור. הנתבים הפנימיים לא יודעים בנוגע לאופן שבו מאורגנים שאר האזורים, אלא הם רק יודעים שכדי להגיע לאזור מסוים, הם צריכים להעביר את הפקטות שלהם אל נתב ה-ABR (Area Border Router). במילים אחרות, אם הרשת הגדולה מפוצלת למספר אזורים, אז בשילוב של MPLS TE ה-CSPF יחווה קשיים בחישוב כל הנתיב, משום שמבחינתו הכתובת שמאזור אחר היא ענן כלשהו, ולא איבר ספציפי.

במקרה הזה נכנס לתמונה ה-**Explicit Path**, שמהווה דרך ישירה לשליטה על הנתיב לפיו ייבנה ה-LSP. הארכיטקט יכול לקבוע בצורה עצמאית את האיברים שעליהם יושתת ה-LSP. נתב ה-Ingress LSR יהיה חייב לעבוד לפי הקווים המנחים שנקבעו. עובדה הזו מעניקה לאלגוריתם ה-CSPF מעט יכולת גיוון.

איך ה-Explicit Path פותר את הבעיה? הבה נראה דוגמה:



[התמונה לקוחה מתוך אתר linkmeup]

חייבות להיות נקודות אמצע/תיווך, שלפי הדוגמה הן: R1,R3,R5. זה גם ה-Explicit Path.

כשאנו מזינים את אלגוריתם ה-CSPF עם ה-Explicit Path הזה, אז הוא בונה את החתיכה כדי להגיע מ-R1 ל-R3 (ב-Area 0). את הנתיב שהוא בונה הוא מתעד ב-ERO, בתוספת איבר אחד מתוך ה-Explicit Path, שהוא R5 (היעד הסופי של החבילה). מכאן שה-ERO יהיה R3, R1,R2. כעת הפקטה מועברת ברשת בהתאם ל-ERO ומגיעה ל- R3. R3 מזהה לפי הכתובת המועברת בפקטה שהוא לא היעד, אלא רק נקודת מעבר, ולכן לוקח את התנאים המצוינים עבור המשאבים הנדרשים בשכלול של כתובת היעד מתוך ה-Explicit Path ומשתמש ב-CSPF כדי לבנות Path. האחרון מנפיק רשימה שלמה של איברים כדי להגיע אל היעד (R3,R4,R5), שמומרת ל-ERO. ועכשיו, הכל שוב עובד בהתאם לעבודה הרגילה.

הווה אומר, שבבחינת ההשוואה שבין ה-Explicit Path ל-ERO, אם ה-Explicit Path מצוין בעבור ה-Tunnel, אז RSVP רוקם ERO, שלוקח בחשבון את הדרישות שנובעות מה-Explicit Path. גם אם ה-Explicit Path מכיל את כל האיברים הנדרשים כדי להגיע אל ה-Egress LSR, ה-RSVP עדיין ישנע מידע אל ה-CSPF.

מסקנה: אם ה-Ingress ו-Egress LSR ממוקמים באזורי IGP שונים, החישוב של הנתיב מבוצע בנפרד בכל אזור, כאשר נקודת השליטה המרכזית היא הנתב ה-ABR.

כדאי להבין ש-Explicit Path שימושי לא רק בעבור המקרה הנדון, אלא באופן כללי מדובר בכלי נוח לשימוש כאשר מדובר ב-LSP.

מושג נוסף שכדאי להכיר בהקשר זה הוא **RRO**. מדובר באובייקט שמאפשר ל-RSVP-TE לתעד את הנתיב שעבר ה-LSP לאורך הרשת. כל נתב בדרך מוסיף את המידע שלו ל-RRO, וכך ניתן לראות את סדר הנתבים והמסלול שה-LSP עבר בפועל. שימוש ב-RRO מסייע במעקב, ניתוח ותכנון של המסלולים, מבלי להשפיע על אופן פעולתו של ה-LSP. כדי למנוע בלבול, נבחין בהבדל חשוב - בעוד ש-RRO מתעד את הנתיב בפועל שעבר ה-LSP, ה-ERO מאפשר ל-Ingress LSR לקבוע מראש את המסלול שה-LSP אמור לעבור. כלומר, ERO מציין את הנתיב הרצוי מראש, בעוד ש-RRO משקף את המסלול שבפועל נעשה בו שימוש ומאפשר בדיקה ומעקב אחר הנתיב.

דגשים בנוגע לפרוטוקולים והשימוש בהם

בסיום המעבר על הפרוטוקולים, הגיוני שיעלו אצלכם תהיות הנוגעות לנקודות הבאות:

- הבדל שבשימוש **RSVP TE LSP** לבין **LDP LSP** - מנקודת מבט גבוהה על שני הפרוטוקולים, אין באמת קונספט של LSP. סך הכל מדובר בנתיב של תווית מתחלפת (Tag Switching Path). עם זאת, ניתן להבחין בין המושג CR-LSP (ConstRaint-based LSP), שנבנה על ידי RSVP TE. בהתייחסות כזו CR-LSP חייב לענות על התנאים המצוינים ב-Tunnel Interface.
- נניח שאחד הנתבים המקשרים (**Intermediate Node**) לא תומך ב-**RSVP-TE** או **LDP** או שהפרוטוקול לא-פעיל על הממשק. האם יבנה LSP? או שמא באיבר הספציפי הזה הפקטה תעבור ל-IP ולאחריו תשוב שוב ל-MPLS?

  במקרה שמדובר ב-RSVP TE Ingress, הנתב "הבעייתי" לא ייכלל ב-TED, ומכאן שלא תהיה אפשרות לבנות LSP לנתב ה-Egress. בהתאם, לא יישלחו הודעת Path, לא תוויות ולא LSP. מידע לא יוכל לעבור על גבי ה-Tunnel.

אם עסקינן ב-LDP – הסיטואציה נעשית מעניינת יותר – אם למשל LDP לא יהיה פעיל על הפורט שהולך לכיוון R5 מ-R2 אז:

1. ב-R1 יהיה Label בעבור ה-FEC, ולמעשה שני Label-ים: אחד שמיועד דרך R2 והשני דרך R3. מכיוון שלפי טבלת הניתוב הדרך הטובה ביותר היא דרך R2 , אז ה-LSP יושתת בכיוון R2.
2. ב-R2 יהיה סימון/תווית, אך לכיוון R1 (1.1.1.1). מכאן, שזו לא הדרך הטובה ביותר, ועל כן היא לא תהיה בשימוש. מכאן שה-LSP שתוכנן חדל מלהתקיים.
3. ב-R5 כן נמצא תווית בעבור ה-FEC הרלוונטי (בניגוד ל-R2). אולם, כפי שמתקבל קיבלנו LSP קרוע/מפוצל (מ-R1 ל-R2 ומ-R5 ל-R6), שהוא בעצם לא עונה על ההגדרה של LSP לפיו תוויות חייבות להתחלף לאורך כל המסלול ומעבר מ-MPLS ל-IP (בין R2 ל-R5) ואז שוב ל-MPLS לא מקובל. כך ש-LSP בעבור FEC 6.6.6.6/32 לא יתקיים כאן.

תעבורה רגילה תעבור דרך הנתיב הזה, אך שירותים של MPLS, כמו VPN, **לא** יעבדו.

- למה שבכלל נשתמש ב-**MPLS TE** כשאפשר להשתמש ב-**IGP Metrics** לצורך ניווט/הפניית המידע? באופן כללי, קביעת הנתיב שבו תעבור תעבורת המידע יכולה להתבצע ע"י קביעת ה-Cost בהתייחס לקישורים, לממשקים וכו'. עם זאת, תחזוקה של שיטה/מערכת שכזו עלולה להוסיף בעיות. מה גם, שלא תהיה אופציה בדרך שכזו להפריד שני זרמים של תעבורת מידע כך שיעברו בשתי דרכים שונות. למעשה, שימוש במטריקות יעביר את בעיית הפצת המידע ברשת מ"כתף אחת לאחרת".

יחד עם זאת, אם ברשותך מספר LSPs שונים אתה יכול לנווט דרכם מידע כרצונך. נכון, גם בתמיכה ב-TE עולים קשיים. ובכל זאת – אם ניקח לדוגמה שני צרכנים שלאחד אנו מחויבים להעביר 40 MB/S ולשני 50 MB/S. נניח וכל ההגדרות שנוגעות למטריקות, ולחישובי QoS סבוכים התבצעו בדרך מייגעת כדי לענות על הצורך – מה יקרה אם משהו ישתבש מבחינה אופטית בכבילה, שייקח שבוע עד שיתוקן? במקרה הזה אנו חייבים לייצר לעצמנו גיבויים שמתאפשרים הודות ל-BackUp LSPs ב-TE. אחרת, לא ניתן יהיה לעמוד בהתחייבויות הרשומות בחוזי ה-SLA (Service-Level-Agreement), הסכם בין הספק ללקוח.

אסכם בנוגע לפרוטוקולים - חשוב להבין שטכנולוגיית ה-MPLS לא מסדירה בהכרח פרוטוקול מסוים להצפת תוויות, ולכן התוצאות הסופיות יכולות להיות שונות מרשת לרשת, בהתאם לפרוטוקולים שנבחרו. לעיתים אף קיים תרחיש של LDP over TE. במקרה כזה RSVP-TE מיועד לארגון התעבורה ויישום של Traffic Engineering בעוד ש-LDP משמש להפצת תוויות שירות (כגון VPN). התוצאה היא שילוב יכולות של שני הפרוטוקולים, במצבים שבהם נמצא בכך צורך.

וקטורי תקיפה והגנה

המנגנון איתו עובדת MPLS מביא איתו פגיעויות לא שגרתיות. בין השאר כי התוויות עצמן הופכות לקטע קריטי בנתיב התעבורה ולא רק "מטא-מידע". MPLS נועדה בראש ובראשונה לשפר ביצועים, לייעל ניתוב ולתמוך בשירותי VPN, אך היא חסרה מנגנוני הצפנה או אימות זהות מובנים. עובדה זו הופכת אותה לפגיעה למספר מתקפות משמעותיות, שחלקן מנצלות חולשות ברמת ניהול התוויות (Labels) וחלקן ברמת מישור הנתונים (Data Plane). הסקירה הבאה תנתח חלק מהאיומים הפוטנציאליים והפתרונות האפשריים כנגדם.

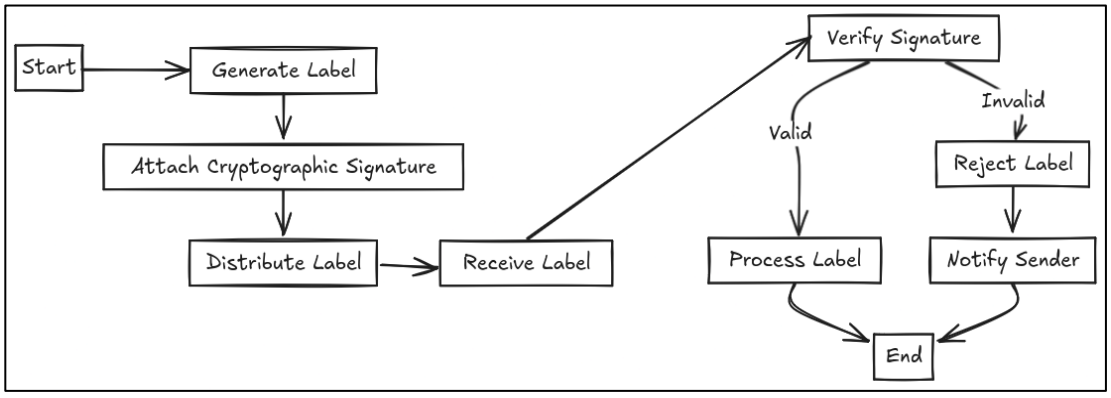
אך רגע לפני שנצלול לכל זה - חשוב שנכיר את המושג VPN.

VPN או Virtual Private Network הוא מנגנון שמאפשר ללקוח "רשת פרטית" גם כשהוא משתמש בתשתית משותפת (כמו האינטרנט או MPLS). ברשת MPLS, כל לקוח מקבל VPN משלו באמצעות תוויות שמבדילות (Service Labels) את התעבורה שלו משל אחרים. בהיבט האבטחתי, זה לא אומר שהתעבורה מוצפנת. מדובר בהפרדה לוגית.

איומים פוטנציאליים:

1. זיוף (Spoofing) ושיחזור (Replay) של תוויות - תוקפים יכולים לזייף חבילות תעבורה תוך שימוש בתוויות פנימיות של VPN קיים. כך נוצרת אשליה שמדובר במידע לגיטימי, למרות שמדובר בזיוף. במתקפות Replay, חבילות ישנות שנלכדו ברשת נשלחות מחדש במטרה ליצור עומס או לשבש את פעילות השירותים. איום זה מנצל את האמינות של המנגנון, ומאפשר לתוקף להחדיר תעבורה זדונית שתנותב בתוך הרשת כאילו היא לגיטימית.

המחשת תהליך אימות התוויות וכיצד חוסר בו מאפשר את התקיפה:

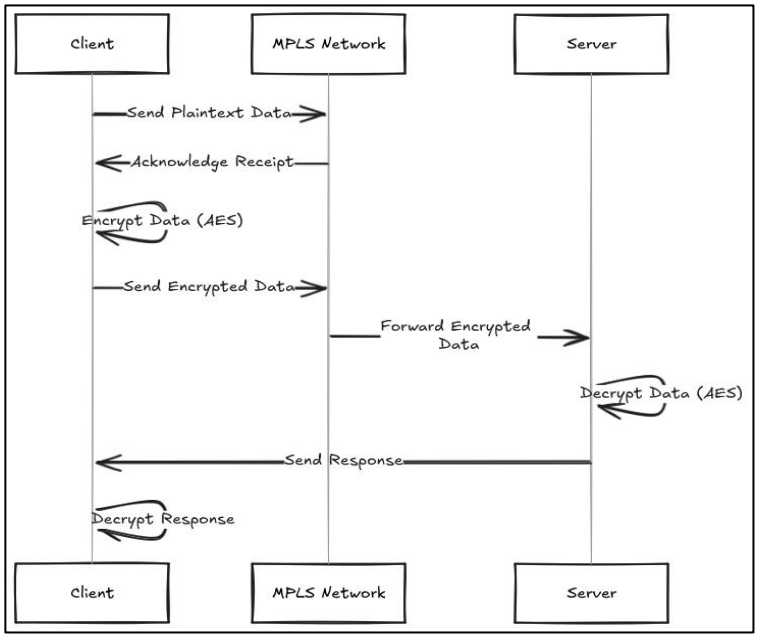


[התמונה לקוחה מתוך אתר arxiv]

2. הזרקה או שינוי תוויות (Label Injection/Modification) - תוקף עלול לשנות את מסלול התעבורה על ידי הזרקה של תוויות זדוניות, ולהסיט מידע ליעד פוגעני. התקפה כזו מתאפשרת לרוב על ידי השגת שליטה על פרוטוקולי ניתוב כמו LDP או BGP בנתבי MPLS, ודרכה אף ניתן לקבל גישה לרשתות VPN של לקוחות. הרי כל המנגנון של MPLS נשען על אמינות התווית, ולכן מניפולציה עליה היא וקטור תקיפה מרכזי.

3. חיבור לא מורשה של נתב לרשת ה-MPLS - בשל העובדה שנתבי PE משמשים כנקודת חיבור קריטית ללקוחות, תוקף שמצליח "להוסיף" נתב משלו לתשתית עשוי להפוך לשחקן לגיטימי ברשת. כך הוא יכול לעקוף בידול לוגי של VPN ולצפות במידע של לקוחות אחרים. איום זה מנצל את האמון הטבעי שקיים בתוך תשתית MPLS פנימית, ומאפשר לתוקף לחדור לרשת מבלי לעמוד בפני בקרות אבטחה אשר עשויות להיות מופעלות בנקודות חיבור חיצוניות.

4. יירוט תעבורה (Eavesdropping) - היעדר הצפנה טבעית בתעבורת MPLS מאפשר יירוט של נתונים לאורך הדרך. גם אם קיים בידול לוגי בין לקוחות שונים (VPN), המידע עצמו עובר כטקסט גלוי. תוקף שמצליח ליירט אותו יכול לחשוף פרטי מידע רגישים. MPLS לא תוכנן לטפל בנושאי חשאיות. אמנם התמונה הבאה מתארת הצפנה, אולם נועדה לחזק את הרעיון ש-MPLS במקור חסר מנגנון כזה, וזה מה שצריך לתקן:



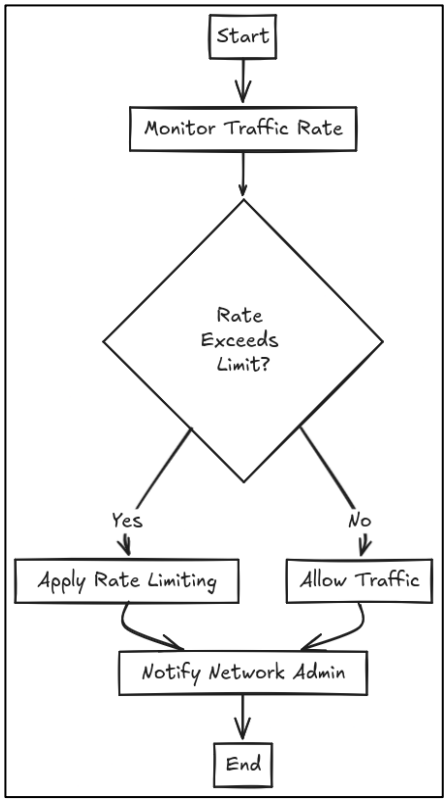
[התמונה לקוחה מתוך אתר arxiv]

5. תקיפות על מישור הנתונים (Data Plane Attacks) - באמצעות מניפולציה של LSPs (Label Switched Paths), ניתן לשבש את זרימת התעבורה, להסיט נתיבים קיימים או לגרום לניתוק שירותים קריטיים. זו פגיעה ישירה בשרידות הרשת ובזמינות השירות.

אסטרטגיות מיתון והגנה:

1. Label-Based Access Control - טבלאות "הרשאה" שמגבילות אילו תוויות מותר לקבל והיכן, מנגנון חיוני למניעת spoofing והזרקה.
2. End-to-End Encryption - הצפנה בין נקודת מקור ליעד, שמנטרלת יירוט של המידע על גבי MPLS.
3. Traffic Encryption Protocols - הוספת שכבת הצפנה לפקטות במהלך המעבר ברשת, כפי שמתואר בתמונה במעלה העמוד וביחד מאפשר שמירה על חשאיות.
4. הגנה מפני DoS: מנגנונים כמו Traffic Shaping ו-Rate Limiting לשליטה בצורה הדוקה על עומס הודעות לא מוסדרות ומניעת הצפה.
5. ניהול אוטומטי ואמינות הרשת (Automated Configuration Management & Redundancy) - שימוש בכלים לאימות תצורה ושכפול נתיבים חלופיים (Backup LSPs) לשמירת זמינות ושחזור מהיר במקרה של תקלות.

תמונה להמחשת תרשים זרימת התגובה למתקפת DoS ואופן טיפול:



[התמונה לקוחה מתוך אתר arxiv]

סיכום

MPLS היא טכנולוגיה שמטרתה לייעל את הניתוב ברשתות גדולות, תוך שימוש בתוויות (Labels) במקום בחיפוש ארוך בטבלת ניתוב. הרעיון המרכזי פשוט: החבילה מתויגת כבר בכניסה ל-MPLS Domain ע"י ה-Ingress Router, וכל נתב לאורך הדרך (LSR) מחליף את התווית בהתאם לטבלת התוויות שלו, עד שהחבילה מגיעה ל-Egress Router, שבו מוסרת התווית. כך נחסך עיבוד מיותר והנתב יכול לעבוד מהר יותר.

רכיבים עיקריים

- Labels ו-Stack Labels- כל פקטה יכולה לשאת תווית אחת או יותר, כשהתווית העליונה היא זו שלפיה הנתב מחליט.
- אופרציות בסיסיות - Push - הוספת תווית, Pop - הסרה, Swap - החלפה.
- סוגי נתבים - Ingress LSR - מוסיף תווית, Transit LSR מחליף, Egress LSR מסיר.
- LSP – Label Switched Path - המסלול החד-כיווני שבו עוברת הפקטה בין ה-Ingress ל-Egress.

קונספטים מרכזיים

- **FEC – Forwarding Equivalence Class** - קיבוץ של חבילות עם מאפיינים משותפים (כתובת יעד, QoS, מזהה VPN וכו') כדי שיעברו באותו LSP.
- **LIB/LFIB -** טבלאות המידע שמנהלות את הקשרים בין Prefixes ל-Labels ב-Control Plane ואת פעולות ההעברה בפועל ב-Data Plane.
- **MPLS Header** - כולל שדה Label, שדה TC לאכיפת QoS, שדה S לסימון תחתית המחסנית, ו-TTL למניעת לולאות.

פרוטוקולים להפצת תוויות

- **LDP (Label Distribution Protocol) -** הפשוט ביותר, מבוסס על טבלת ניתוב קיימת (IGP) ומפיץ תוויות אוטומטית.
- **RSVP-TE** הרחבה של RSVP, מאפשרת Traffic Engineering והקצאת משאבים (QoS) לאורך נתיב מוגדר מראש.
- **MP-BGP** - משמש ליישום VPN ולשילוב MPLS עם שירותי ריבוי לקוחות.

יתרונות MPLS

- ביצועים גבוהים – בזכות עיבוד מהיר ופחות בדיקות Routing.
- גמישות – מאפשרת להעביר לא רק IPv4 אלא גם Frame Relay, Ipv6 ועוד.
- QoS - שליטה בתעדוף ובמשאבים לאורך הנתיב.
- VPN - מאפשרת לספקי שירות לייצר רשתות וירטואליות מאובטחות ומבודדות ללקוחות שונים.
- Traffic Engineering - ניהול עומסים, עקיפת צווארי בקבוק, ושימוש אופטימלי במשאבים.

עקרונות חשובים

- נתיבי LSP הם חד-כיווניים – לכל כיוון יש לבנות נתיב נפרד.
- FEC הוא הבסיס ליעילות – הוא מגדיר מראש אילו חבילות יקבלו טיפול זהה.
- שילוב עם BGP - מאפשר לצמצם את הצורך לשאת טבלאות BGP ענקיות בכל הנתבים, ולהסתפק בניהולן בקצוות בלבד.
- תצורות שמרניות מול ליברליות – קובעות כיצד הנתב ישמור או יזרוק תוויות שאינן בשימוש.

לאור האמור MPLS איננה "עוד פרוטוקול ניתוב", אלא שכבת תוויות שנבנית מעל פרוטוקולים קיימים. היא נועדה לייעל את העברת התעבורה, לאפשר ניהול גמיש של משאבים ולספק שירותים מתקדמים כמו VPN ו-QoS. בעולם ספקי השירות MPLS היא טכנולוגיה קריטית שממשיכה להיות בסיס לפתרונות מודרניים גם היום. מומלץ שעם השימוש ב-MPLS יהיה גם שימוש באסטרטגיות הגנה, שיבטיחו את שרידות, אמינות וחשאיות המידע ברשת.

על המחבר

עמית גבאי שירת בקבע בתפקיד בתחום הנדסת מערכות תקשורת ואבטחת מידע בצה"ל,עם ניסיון מעשי בניהול, תכנון, ותפעול שוטף של רשתות תקשורת מרובות. בעל הסמכת – Cisco Certified Specialist Enterprise Core, בוגר תוכנית "מגשימים", וכן בוגר י"ג בהנדסת תוכנה. עמית משלב ניסיון מעשי עם זיקה עמוקה לטכנולוגיה, סקרנות מחקרית ואמביציה לקדם חדשנות בעולם ה-Networking וה-Cybersecurity. לצד עבודתו המקצועית, הוא פועל להנגשת ידע מורכב לקהל רחב, מתוך אמונה ששיתוף ידע הוא מנוע מרכזי בהתפתחות הקהילה הטכנולוגית בעולם ובישראל בפרט.

עמית משמש כחבר בקהילת 'מגשימים נקסט' – ארגון הבוגרים של "מגשימים", תוכנית הסייבר הלאומית, שמטרתה לקדם מצוינות, מקצועיות וציונות באמצעות הנגשת לימודי מדעי המחשב והסייבר לתלמידי הפריפריה. הקהילה מהווה עבור הבוגרים בית חם, המאגד סיניורים, יזמים והצלחות משמעותיות.

להצטרפות ומעקב אחר הקהילה בסושיאל:

https://www.linkedin.com/company/magshimim-next/

https://www.instagram.com/magshimim.next/

מקורות מידע

- linkmeup - https://linkmeup.ru/blog/1207/
- Wikipedia - https://he.wikipedia.org/wiki/MPLS
- Cisco – "Relationship between LIB,FIB and LFIB." https://community.cisco.com/t5/mpls/relationship-between-lib-fib-and-lfib/td-p/782307
- Quora – "Why is MPLS faster than IP routing?" https://www.quora.com/Why-is-MPLS-faster-than-IP-routing
- arxiv - Security Implications and Mitigation Strategies in MPLS Networks https://arxiv.org/html/2409.03795v1
- What is MPLS? Multiprotocol Label Switching Defined - Fortinet https://www.fortinet.com/resources/cyberglossary/mpls#:~:text=Protects%20your%20network%20from%20threats%20that%20MPLS,network%20easier%20to%20manage%20and%20keep%20secure.
- Cato Networks - What Is MPLS? Definition, Pros/Cons, Alternatives https://www.catonetworks.com/what-is-mpls/
- Mitigating Some Security Attacks in MPLS-VPN Model “C” - UPV https://personales.upv.es/thinkmind/dl/journals/netser/netser_v5_n34_2012/netser_v5_n34_2012_12.pdf
- MPLS and VPLS Security - Black Hat https://www.blackhat.com/presentations/bh-europe-06/bh-eu-06-Rey-up.pdf
- MPLS circuits: A guide to multiprotocol label switching - Meter https://www.meter.com/resources/mpls-circuit