

Dynamic Routing

The Best Of Routing

Introduction

- ▶ פרוטוקולי ניתוב דינמיים נמצאים בשימוש כבר משנות ה-80'. הפרוטוקול הראשון שסלל את הדרך היה RIP גרסה 1, פרוטוקול בעל אלגוריתם פשוט למדי, שמילא את כל צורכי הרשתות של אותם ימים. במשך השנים הרשתות גדלו, התפתחו והפכו יותר מורכבות, דבר שהצריך פרוטוקולים חכמים יותר ומתקדמים יותר כמו OSPF ו-EIGRP. התפקיד של פרוטוקולי הניתוב הוא פשוט וקריטי ביותר לכל רשת. הם דואגים ללמוד באופן עצמאי את טופולוגית הרשת ולנתב מידע בצורה המהירה והיעילה ביותר מכל מקור לכל יעד ברשת, מאפיין נוסף הוא הסתגלות לשינויים, הפרוטוקולים מסוגלים לגלות רשתות חדשות ולהסיר רשתות שנותקו/כשלו מטבלאות הניתוב ושם זה לא נגמר, הפרוטוקול יחפש מסלול חלופי לכל רשת שאבדה עקב כשל כזה או אחר.
- ▶ כל פרוטוקול ניתוב כולל כמה גרסאות, כל גרסה מציעה מאפיינים מתקדמים כמו תמיכה ברשתות שעברו סיבנוט (CIDR), אבטחת עדכוני הפרוטוקול ותמיכה בפרוטוקול IPv6.
- ▶ בטופולוגיות רשת גדולות, נעדיף תמיד להשתמש בניתוב דינמי ולא סטטי.



Advantages Vs. Disadvantages

▶ היתרונות:

- **ניתוב דינמי מתאים לכל טופולוגיה** - בדר"כ נשתמש בטופולוגיות בנוניות-גדולות.
- **ניתוב דינמי הוא עצמאי לחלוטין** - הפרוטוקול אוטומטית יכיר וילמד רשתות מרוחקות וימצא מסלול חלופי (אם קיים) לרשתות שכבר לא ניתן לגשת אליהן, עקב נתב או לינק שכשלו.
- **ניתוב דינמי הוא מהיר ויעיל** - אלגוריתם מתוחכם ומידע שנאסף מנתבים אחרים מסייע לפרוטוקול למצוא את המסלול (Route) הכי מהיר לכל רשת!

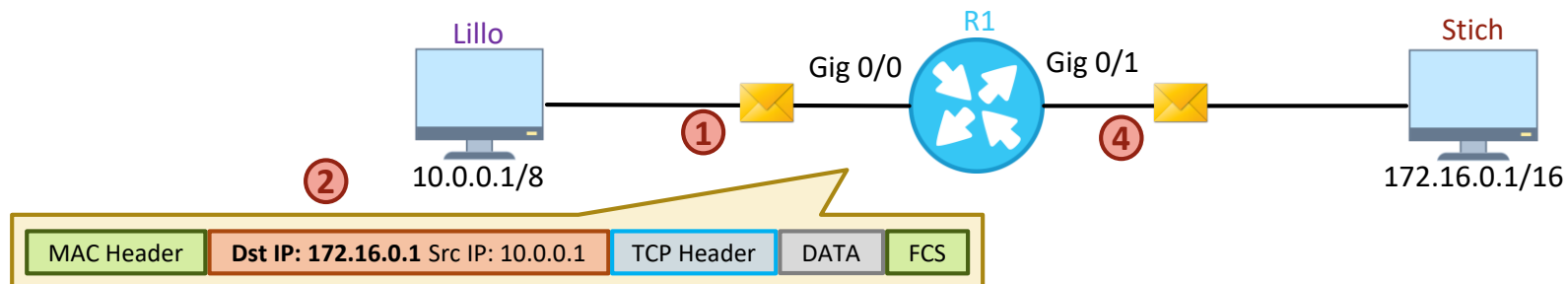
▶ חסרונות:

- **הגדרה מורכבת** - לניתוב דינמי מאפיינים רבים, כל מאפיין דורש רמה מסויימת של תכנון והקפדה. לכן בטופולוגיות מורכבות ניתוב דינמי דורש הגדרה ודיוק רב.
- **פחות מאובטח** - בשונה מניתוב סטטי, ניתוב דינמי שולח עדכונים באופן קבוע ברחבי הרשת, עדכונים אלו מהווים סיכון בידיים הלא נכונות. לכן חושב ליישם את הגדרות האבטחה של כל פרוטוקול.
- **צריכת משאבים רבים** - פעולת חישוב האלגוריתם של כל פרוטוקול צורכת משאבי חומרה רבים. כמו: כוח עיבוד (CPU), זיכרון (RAM) ואחזים מעטים מרוחב הפס (Bandwidth).

Routing Process

תהליך הניתוב שמבצע הנתב מתבסס על שני פרמטרים חשובים: יעד החבילה וטבלת ניתוב מעודכנת! פרוטוקולי הניתוב הדינמיים דואגים תמיד לעדכן את טבלאות הניתוב של הנתבים בטופולוגיה במידע הכי עדכני ומדויק! מנהל הרשת יכול לישון בראש שקט כל עוד פרוטוקול כמו OSPF דואג שמידע מכל סוג שהוא, נשלח בצורה הכי מהירה ומדויקת מכל מקום לכל מקום בארגון. לדוג': שליטה מרחוק במחשבי הארגון או שליחת דואר אלקטרוני.

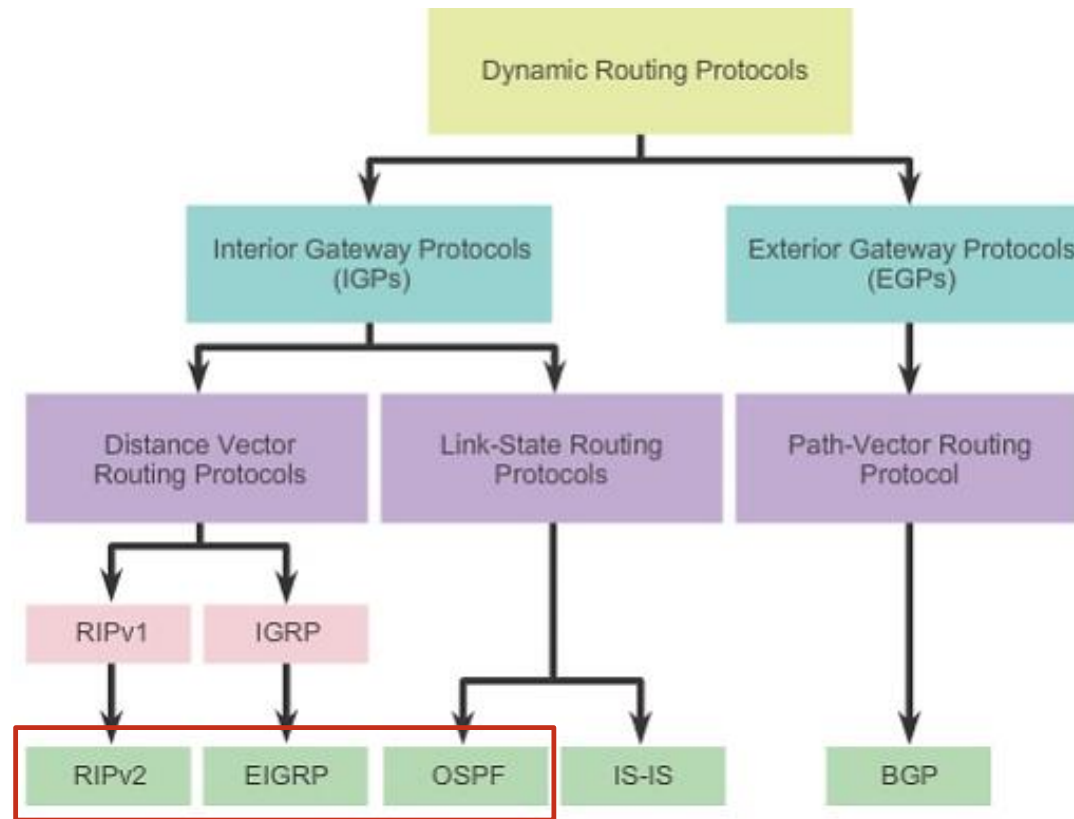
תהליך הניתוב במבט מהיר:



R1#show ip route
10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C 10.0.0.0/8 is directly connected, GigabitEthernet0/0
L 10.0.0.254/32 is directly connected, GigabitEthernet0/0
172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C 172.16.0.0/16 is directly connected, GigabitEthernet0/1
L 172.16.0.254/32 is directly connected, GigabitEthernet0/1

Interior & Exterior

תרשים שמסכם את סוגי הפרוטוקולים, קיימים שני סוגי עיקריים: פרוטוקול ניתוב פנימי וחיצוני.
רוב פרוטוקולי הניתוב מתאימים לרשתות פנימיות (Autonomous System).



Protocol Types

▶ מטרת פרוטוקולי הניתוב היא אחת: ניתוב יעיל ומהיר, אבל כל פרוטוקול עושה זאת בצורה מעט שונה. ניתן לסווג את פרוטוקולי הניתוב הפנימיים לשני סוגים:

▶ **Distance Vector**-סוג זה של פרוטוקול נקרא כך כי הוא כולל שני פרמטרים: מרחק (Distance) וכיוון (Vector). נתב שמפעיל פרוטוקול מסוג זה יחליף מידע ניתוב אך ורק עם הנתבים שכנים, אליהם הוא מחובר ישירות. זאת אומרת שהנתב המקומי יודע מאיזה נתב הוא הכיר רשת מסויימת, אבל לא ברור לו כיצד אותו נתב שכן הכיר את אותה הרשת. הנתב המקומי אינו רואה מה מתרחש מעבר לנתבים השכנים שלו. בגלל אופי הפרוטוקול יכולים להיווצר Routing loops לכן פיתחו פיצ'רים למנוע זאת כמו: split horizon. לדוג' ניווט רק בעזרת שלטי דרכים.

○ RIP ו-EIGRP הם פרוטוקולים מסוג **Distance Vector**

▶ **Link-State**-סוג פרוטוקול זה בשונה מהפרוטוקול הקודם, מצריך שכל הנתבים ברשת יחליפו מידע ניתוב אחד עם השני. זאת אומרת שכל נתב בטופולוגיה מכיר כל נתב אחר (לא רק שכנים) בתור התחלה ולאחר מכן לומד את כל הרשתות וכל המסלולים האפשריים לכל רשת. לדוג' ניווט בעזרת מפה.

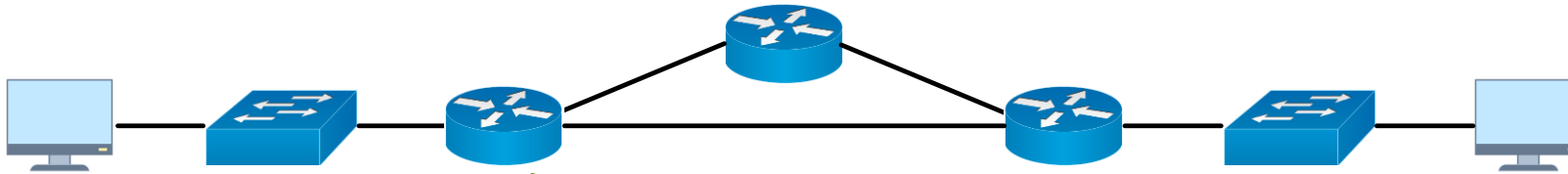
○ OSPF הוא פרוטוקול מסוג **Link-State**

Routing Table

- ▶ הנתב מסתמך לחלוטין על ניתובים (Routes) בטבלת הניתוב, אך איך הנתב בכלל לומד את הניתובים הללו!
- ▶ הנתב לומד או יותר נכון מוסיף מסלולים (Routes) ורשתות בשלוש דרכים עיקריות:
 1. **הרשת מחוברת ישירות** לנתב או Directly connected - במצב כזה נוכל לראות את האות **C** או **L** ליד הנתיב.
 2. **ניתוב סטטי** - מנהל הרשת מגדיר ומוסיף את הנתיב בצורה ידנית לטבלת הניתוב, במצב כזה נוכל לראות את האות **S** ליד הנתיב.
 3. **ניתוב דינמי** - פרוטוקול ניתוב דינמי אוסף מידע על רשתות מרוחקות אוטומטית ומעדכן את טבלת הניתוב, במצב כזה נוכל לראות את הסימון של הפרוטוקול דרכו נלמד הנתיב:
 - האות **R** מייצגת את פרוטוקול הניתוב RIP.
 - האות **O** מייצגת את פרוטוקול הניתוב OSPF.
 - האות **D** מייצגת את פרוטוקול הניתוב EIGRP.

Routing Table

דוגמה לטבלת ניתוב של נתב בטופולוגיה בה מיושם ניתוב סטטי וניתוב דינמי עם OSPF: ►



רשת מקומית, מחוברת ישירות

רשת מרוחקת, נלמדה בעזרת ניתוב דינמי OSPF

רשת מרוחקת, נלמדה בעזרת ניתוב סטטי

R1#show ip route

172.16.0.0/16 is variably subnetted, 5 subnets, 2 masks

C 172.16.0.0/30 is directly connected, Serial0/0/0

L 172.16.0.1/32 is directly connected, Serial0/0/0

O 172.16.0.4/30 [110/128] via 172.16.0.2, 00:01:54, Serial0/0/0

[110/128] via 172.16.0.10, 00:01:54, Serial0/0/1

C 172.16.0.8/30 is directly connected, Serial0/0/1

L 172.16.0.9/32 is directly connected, Serial0/0/1

192.168.10.0/24 is variably subnetted, 2 subnets, 2 masks

C 192.168.10.0/24 is directly connected, GigabitEthernet0/0

L 192.168.10.254/32 is directly connected, GigabitEthernet0/0

O 192.168.20.0/24 [110/65] via 172.16.0.2, 00:04:06, Serial0/0/0

S* 0.0.0.0/0 is directly connected, Serial0/0/0

Administrative Distance

- ▶ Administrative Distance הוא ערך קבוע אשר מוגדר מראש לכל שיטת ניתוב, ערך זה נועד לעזור לנתבים להחליט באיזו שיטה עדיף להשתמש. זאת אומרת נתב יעדיף תמיד להשתמש בשיטת הניתוב בעלת הערך המספרי הנמוך ביותר.
- ▶ לדוגמה: נתב אשר מוגדר עם נתיב סטטי לרשת מסויימת וגם פרוטוקול ניתוב דינמי OSPF לאותה הרשת, יעדיף הנתב להשתמש בנתיב הסטטי, כי הוא בעל ערך AD נמוך יותר!
- ▶ טבלת ערכי ה-AD:

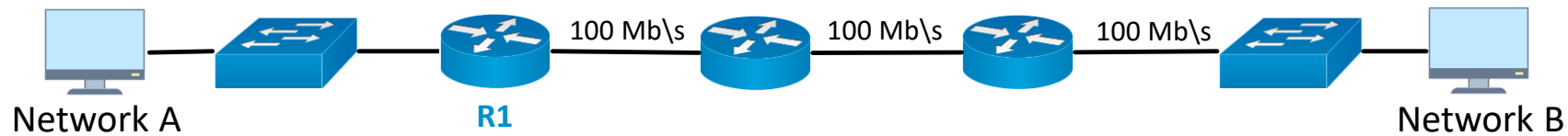
Administrative Distance	Default Distance
Connected Interface	0
Static Route	1
Internal EIGRP	90
OSPF	110
IS-IS	115
RIP	120

Metric

- ▶ Metric הוא ערך בו נתבים נעזרים בכדי לבחור את המסלול (Route) הכי מהיר לרשת יעד מרחוקת מבין המסלולים האחרים (אם קיימים). מספר פרמטרים עוזרים לפרוטוקול ניתוב דינמי לקבוע את ה-Metric של כל מסלול. המסלול בעל ה-Metric הכי נמוך לרשת יעד מסוימת נבחר להיות הנתיב שיכנס לטבלת הניתוב.
- ▶ הפרמטרים בהם משתמש כל פרוטוקול נקרא Metric, הפרמטרים בהם כל פרוטוקול משתמש לחישוב המסלול:
 - פרוטוקול RIP משתמש ב-Hop Count. הפרוטוקול מודד את כמות הנתבים בכל מסלול ובוחר את המסלול בעל כמות הנתבים הקטנה ביותר.
 - פרוטוקול OSPF משתמש ב-Cost. הפרוטוקול בודק את רוחב הפס של כל לינק במסלול אל רשת היעד ומחבר את ערכי המהירות ל-Cost סופי, שמייצג את כל המסלול.
 - פרוטוקול EIGRP משתמש בשני פרמטרים: רוחב פס (Bandwidth) ועיכוב (Delay). הפרוטוקול מחשב את ה-Delay של כל לינק במסלול אל רשת היעד וכולל גם חישוב של רוחב הפס. ניתן להוסיף פרמטרים נוספים לחישוב בפרוטוקול זה.

Metric Calculation

דוגמה לחישוב ה-Metric של מסלול (Route) בין רשת A ל-B לפי כל פרוטוקול: ►



► RIP: ה-Metric של המסלול הוא: 2

○ רשת B נמצאת במרחק 2 נתבים מ-R1

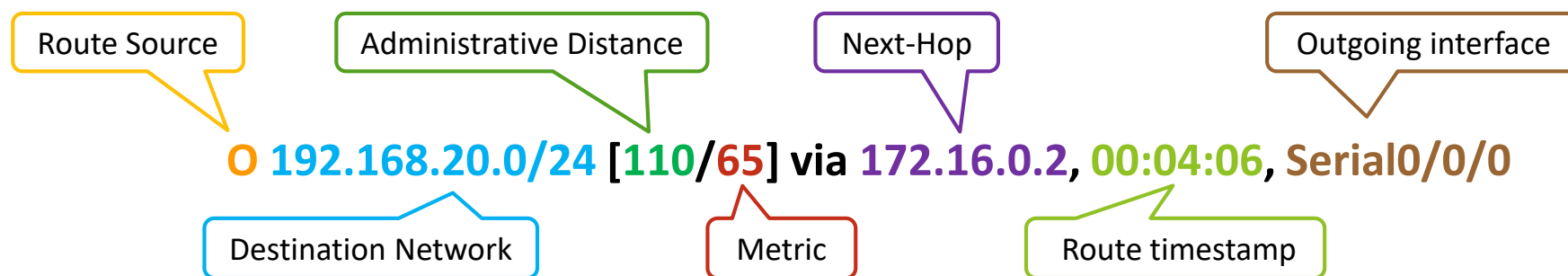
► OSPF: ה-Metric של המסלול הוא: 3

○ ה-Cost של כל לינק שווה 1 לפי רוחב הפס 100 מגה.

○ 3 לינקים (ממשקים) מפרידים בין R1 לרשת B.

Remote Network Entire

ניתוח והסבר לנתיב של פרוטוקול OSPF בטבלת הניתוב (רלוונטי גם עבור פרוטוקולים אחרים): ►















- Route Source – סימון שמייצג את הפרוטוקול דרכו נלמד הנתיב.
- Destination Network – רשת היעד המרוחקת.
- Administrative Distance – ערך שמסייע לנתב לבחור את פרוטוקול הניתוב המועדף.
- Metric – ערך שהקצה פרוטוקול הניתוב לנתיב המדובר (ערך נמוך מייצג נתיב מועדף).
- Next-Hop – כתובת ה-IP של הנתב השכן, דרכו הנתב המקומי ישלח את המידע אל הרשת המרוחקת.
- Route timestamp – חותמת זמן, כמה זמן הנתב מכיר את הנתיב המדובר.
- Outgoing Interface – הממשק דרכו ישלח הנתב המקומי את המידע אל הרשת המרוחקת.

Topology Changes

- ▶ תכונה אחת בולטת שהופכת ניתוב דינמי ליעיל ביותר לעומת ניתוב סטטי, היא היכולת להסתגל אוטומטית לשינויים בטופולוגית הרשת. כל פרוטוקולי הניתוב חולקים את התכונה הזו.
- ▶ הפרוטוקול יכול כמעט מיד לגלות נתב או רשת חדשה שהצטרפו לטופולוגיה, הפרוטוקול גם יכול לזהות נפילה של נתב או אובדן תקשורת עם רשת מרוחקת. בכל מקרה בו המסלול (Route) נקטע, ינסה הפרוטוקול למצוא מסלול חלופי במידה וקיימות אפשרויות נוספות.
- ▶ פרוטוקולי ניתוב מזהים שינויים בטופולוגיה בעזרת מספר כלים:
 - חבילות Hello - מגלות נתבים אחרים ויוצרות יחסי שכנות איתם.
 - Hello Interval - ערך שקובע כל כמה זמן הנתב ישלח הודעת Hello לשכניו, מטרת הודעה זו לעדכן את שאר הנתבים שהכל כשורה. לדוג' חבילת Hello כל 10 שניות.
 - Dead Interval - ערך שקובע כמה זמן ימתין הנתב המקומי, לפני שיסיר את הנתב השכן מטבלאות הניתוב והשכנים. מטרת הטיימר היא להסיר נתבים שכבר לא מגיבים או נגישים כלל מטופולוגית הרשת של הפרוטוקול.

[רשימת הערכים המלאה](#)

Dynamic Protocols Comparison

Protocol	Type	AD	Mark	Metric	CISDR\VLSM	IPv6 תומך	Secure
RIPv1	Distance Vector	120	R	Hop Count			
RIPv2	Distance Vector	120	R	Hop Count			
RIPv3	Distance Vector	120	R	Hop Count			
OPSFv2	Link State	110	O	Cost			
OSPFv3	Link State	110	O	Cost			
EIGRPv4	Distance Vector	90	D	Bandwidth Delay			
EIGRPv6	Distance Vector	90	D	Bandwidth Delay			

Administrative Distance - AD*

Protocols Interval comparison

השוואה בין ערכי הטיימרים (intervals) של הפרוטוקולים. לפי ערכים אלו ניתן לחזות את זמן התגובה של כל פרוטוקול לשינויים בטופולוגיה. ככל שהפרוטוקול יותר מתקדם זמן התגובה של יותר מהיר, מאפיין חיוני לרשתות מודרניות.

Protocol	Hello Interval	Hold\Dead Interval	Routing Table mark
RIP	30 Seconds	180 Seconds	R
OSPF	10 Seconds	40 Seconds	O
EIGRP	5 Seconds	15 Seconds	D

בטבלה לעיל מוצגים ערכים שהם **ברירת המחדל** של כל פרוטוקול. שימו לב! ניתן לשנות את ערכי הטיימרים לערכים שעל דעת מנהל הרשת יתאימו יותר לטופולוגיה ולצרכי הרשת.
לדוגמה: לשנות את הערכים של פרוטוקול OSPF ל- hello כל 3 שניות ו- hold ל-12 שניות.

