הגדרות ב-Packet Tracer

VLAN:

נזכור שתחילה יש להגדיר לכל מחשב ומחשב כתובת IP ו-subnet-mask.
מחשבים הנמצאים תחת אותו ה-VLAN צריכים להיות ברשת זהה.
כל VLAN ו-VLAN צריך להיות ברשת נפרדת.

1. **הגדרת Access point.**
את ה-Access יש להגדיר אך ורק בסוויצ'ים, **ואך ורק ב-interfaces המחוברים ל-end-devices (מחשבים). כלומר, לא בין סוויץ' לסוויץ'!** לא צריך להגדיר דבר ב-interface מהכיוון של המחשב!
יש להגדיר את ה-Access עבור כל interface ו-interface המחוברים למחשב בנפרד.

Access configuration:

```
Switch>enable
Switch# conf t
Switch(config)#interface <interface_name>
Switch(config-if)#switchport access vlan <number_of_vlan>
```

2. **הגדרת Trunk**
את ה-Trunk יש להגדיר אך ורק ב-interfaces המחוברים בין סוויץ' לסוויץ', או בין סוויץ' לראוטר.

```
Switch>enable
Switch# conf t
Switch(config)#interface <interface_name>
Switch(config-if)# switchport mode trunk
```

Router on a stick

הגדרת תקשורת בין VLANים דרך ראוטר. הראוטר היחיד יהיה מחובר לסוויצ'ים הנמצאים בתוך ה-LAN.

עבור הגדרה זו יש תחילה להקים את ה-VLANים הרצויים בתוך ה-LAN.
כמו כן, יש להגדיר עבור כל end-device השייך ל-VLAN את ה-Default gateway שלו (משום שכעת אנחנו מערבים ראוטר, שישמש כ-default gateway). לרוב, ה-Default gateway יהיה כתובת ה-host האחרונה ברשת, כלומר, כתובת אחת לפני כתובת ה-Broadcast.

לדוגמא:
אם הרשת שלנו היא 10.0.0.0, עם subnet mask של 255.255.255.0, כתובת ה-default gateway תהיה 10.0.0.254.

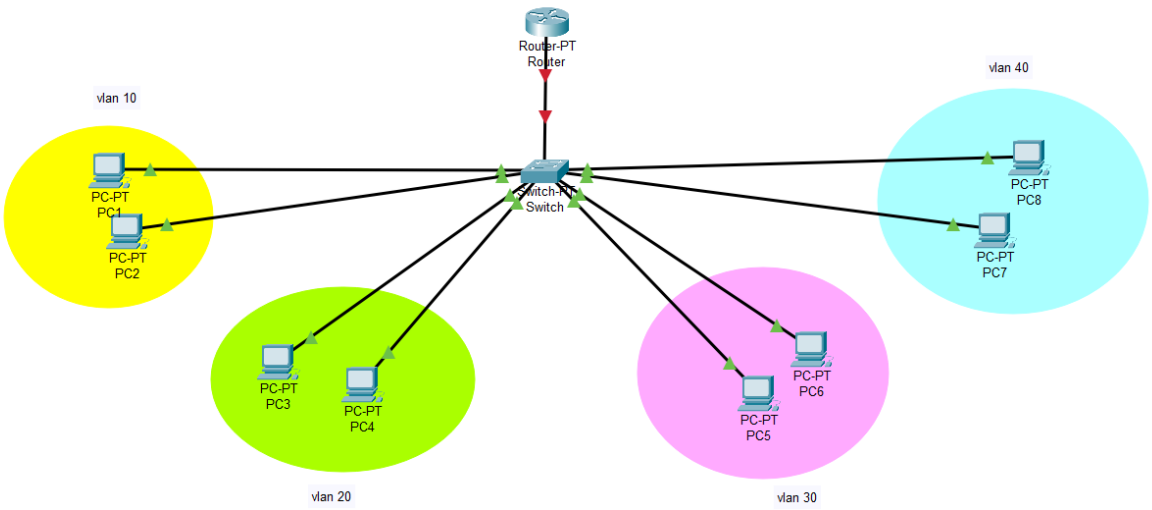
יש להגדיר את עבור כל אחד ואחד מהVLANים בנפרד.

1. יצירת Router on a stick

```
Router>enable
Router# conf t
Router(config)#int <interface_name>
Router(config-if)#no shutdown
Router(config)#int <interface_name>.<number_of_vlan>
Router(config-subif)#encapsulation dot1q <number_of_vlan>
Router(config-subif)#ip address <default_gateway_IP> <subnet_mask>
```

לדוגמא:
הגדרה בראוטר עבור:

- VLAN 10, כתובת רשת 10.0.0.0, subnet mask 255.255.255.0, DG: 10.0.0.254
- VLAN 20, כתובת רשת 10.1.0.0, subnet mask 255.255.255.0, DG: 10.1.0.254
- VLAN 30, כתובת רשת 10.2.0.0, subnet mask 255.255.255.0, DG: 10.2.0.254
- VLAN 40, כתובת רשת 10.3.0.0, subnet mask 255.255.255.0, DG: 10.3.0.254



*הטופולוגיה נוצרה רק לשם ההמחשה. לא הוגדרו בה ההגדרות הרצויות, לכן החיבורים לא בהכרח ירוקים.

```
Router>enable
Router# conf t
Router(config)#int g0/0 (the interface name can differ)
Router(config-if)#no shutdown
Router(config)#int g0/0.10 (for VLAN 10)
Router(config-subif)#encapsulation dot1q 10 (VLAN ID)
Router(config-subif)#ip address 10.0.0.254 255.255.255.0 (will be the
default gateway of VLAN 10)
Router(config)#int g0/0.20 (for VLAN 20)
Router(config-subif)#encapsulation dot1q 20 (VLAN ID)
Router(config-subif)#10.1.0.254 255.255.255.0 (will be the default gateway of
VLAN 20)
Router(config)#int g0/0.30 (for VLAN 30)
Router(config-subif)#encapsulation dot1q 30 (VLAN ID)
Router(config-subif)#10.2.0.254 255.255.255.0 (will be the default gateway of
VLAN 30)
Router(config)#int g0/0.40 (for VLAN 40)
Router(config-subif)#encapsulation dot1q 40 (VLAN ID)
Router(config-subif)#10.3.0.254 255.255.255.0 (will be the default gateway of
VLAN 40)
```

יצירת כתובות IP עבור ראוטרים ויצירת רשת בין שני ראוטרים

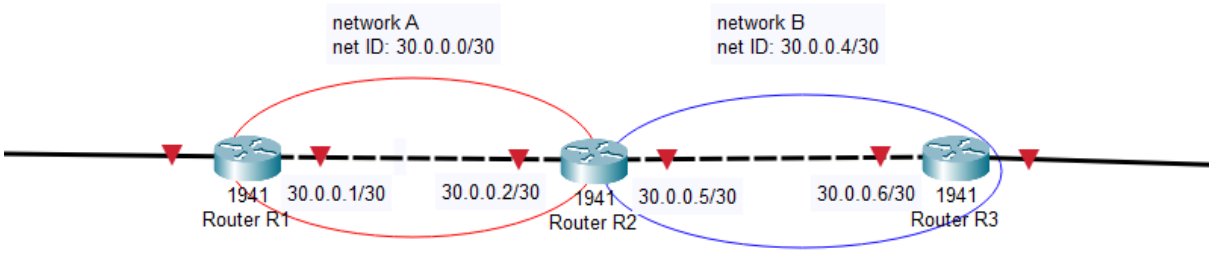
לעיתים נצטרך להגדיר "חיבור", כלומר רשת, בין שני ראוטרים. כל interface בראוטר המחובר ל-interface בראוטר אחר, יחשב לרשת נפרדת.

כל רשת צריכה לכלול שני hosts, אחד עבור כל ראוטר. מסיבה זו, נגדיר כל רשת ורשת עם subnet mask של 255.255.255.252. רשת כזו כוללת 4 כתובות כך שהראשונה שמורה עבור כתובת הרשת, האחרונה עבור broadcast וביניהן נשארו שתי כתובות, אחת עבור ה-interface בראוטר הראשון והשנייה עבור ה-interface בראוטר השני.

1. הגדרת IP עבור ראוטר

```
Router>enable
Router# conf t
Router(config)#int <interface_name>
Router(config-if)#no shutdown
Router(config-if)#ip address <ip_adress> <subnet_mask>
Router(config-if)#exit
```

לדוגמא:
נרצה להגדיר IP עבור רשתות שיתקיימו בהמשך בין R1 ל-R2 (נסמן – A), ובין R2 ל-R3 (נסמן – B)



*הטופולוגיה נוצרה רק לשם ההמחשה. לא הוגדרו בה ההגדרות הרצויות, לכן החיבורים לא בהכרח ירוקים.

הגדרות ב-R1:
הגדרות עבור רשת A:

```
R1>enable
R1#conf t
R1(config)#int g0/0/0 (the interface name can differ)
R1(config-if)#no shutdown
R1(config-if)#ip address 30.0.0.1 255.255.255.252
R1(config-if)#exit
```

הגדרות ב-R2:
הגדרות עבור רשת A:

```
R2>enable
R2#conf t
R2(config)#int g0/0/0 (the interface name can differ)
R2(config-if)#no shutdown
R2(config-if)#ip address 30.0.0.2 255.255.255.252
R2(config-if)#exit
```

הגדרות עבור רשת B:

```
R2(config)#int g0/1/0 (the interface name can differ)
R2(config-if)#no shutdown
R2(config-if)#ip address 30.0.0.5 255.255.255.252
R2(config-if)#exit
```

הגדרות ב-R3:
הגדרות עבור רשת B:

```
R3>enable
R3#conf t
R3(config)#int g0/0/0 (the interface name can differ)
R3(config-if)#no shutdown
R3(config-if)#ip address 30.0.0.6 255.255.255.252
R3(config-if)#exit
```

הגדרת פרוטוקול Static routing

פרוטוקול ניווט בין ראוטרים.
פרוטוקול בו אנו מגדירים ידנית את הדרך בין ראוטר לראוטר במטרה להגיע ליעד נבחר ספציפי.

אנו מגדירים עבור כל ראוטר:

- הרשת אליה אנחנו רוצים להגיע **בסופו של דבר** (היעד הסופי)
- ה-subnet mask של הרשת הסופית
- ה"קפיצה הבאה" = הראוטר אליו הראוטר הנוכחי מעבר את החבילה בדרכה ליעד הסופי (כמובן שהם חייבים להיות מחוברים!)

כמובן שאם אנו רוצים להעביר פקטה בדרך מסויימת בין שתי רשתות, עלינו להגדיר את 2 הכיוונים! מעבר מרשת A ל-B בדרך הרצויה **ובנוסף** מעבר מהמרשת B ל-A.

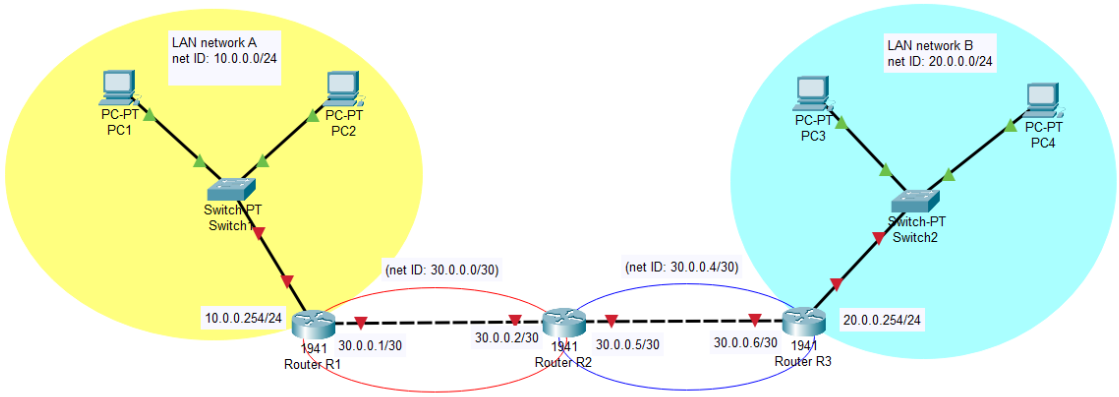
* כאשר הגענו במעבר מרשת A לרשת B (למשל) לראוטר המחובר ישירות לרשת A (ה- default gateway של רשת A), איננו צריכים להגדיר עבורו static route בכיוון של רשת A! הראוטר עצמו כבר יודע "להיכנס" אל הרשת הפנימית שלו.

* פרוטוקול זה "חזק" יותר מפרוטוקול RIP!, כלומר, אם יוגדרו על אותו הראוטר שני הפרוטוקולים, הפרוטוקול שיבחר בסופו של דבר יהיה static routing.

*ה-administrative distance של הנתיב אותו אנו מזינים, כל עוד לא נכתב אחרת, יהיה באופן אוטומטי ב"שווי" של 1.

```
Router>enable
Router# conf t
Router(config)#ip route <destination_ip> <destination_subnet mask> <next_hop_ip>
```

לדוגמא:
קיימות:
- רשת A – כתובת רשת 10.0.0.0, subnet mask: 255.255.255.0
-ראוטר R1 – ה-default gateway של רשת A. נמצא ברשת עם ראוטר R2. כתובת IP: 30.0.0.1
-ראוטר R2 – נמצא ברשת עם ראוטר R1, כתובת IP: 30.0.0.2. נמצא ברשת עם ראוטר R3, כתובת IP: 30.0.0.5.
-ראוטר R3 – נמצא ברשת עם ראוטר R2, כתובת IP: 30.0.0.6. ה-default gateway של רשת B.
- רשת B – כתובת רשת 20.0.0.0, subnet mask: 255.255.255.0



*הטופולוגיה נוצרה רק לשם ההמחשה. לא הוגדרו בה ההגדרות הרצויות, לכן החיבורים לא בהכרח ירוקים.

הגדרה של static routing בין רשת A לרשת B (ובחזרה):

```
הגדרות ב-R1:
הגדרות עבור תקשורת עם רשת B (מ-A ל-B):

R1>enable
R1# conf t
R1(config)#ip route 20.0.0.0 255.255.255.0 30.0.0.2
הגדרות ב-R2:
הגדרות עבור תקשורת עם רשת B (מ-A ל-B):

R2>enable
R2# conf t
R2(config)#ip route 20.0.0.0 255.255.255.0 30.0.0.6

הגדרות עבור תקשורת עם רשת A (מ-B ל-A):

R2(config)#ip route 10.0.0.0 255.255.255.0 30.0.0.2
הגדרות ב-R3:
הגדרות עבור תקשורת עם רשת A (מ-B ל-A):

R3>enable
R3# conf t
R3(config)#ip route 10.0.0.0 255.255.255.0 30.0.0.5
```

הגדרת floating route

כדי ליצור נתיב "Plan B" עבור המסלול אותו יצרנו (כדי למנוע שבמצב של תקלה במסלול הרגיל הפקטה לא תוכל להגיע ליעדה) עלינו לקבוע administrative distance גבוהה יותר מהדיפולטיבי (1) עבור נתיב חדש. נוכל לעשות זאת בצורה הבאה:

```
Router>enable
Router# conf t
Router(config)#ip route <dest_ip> <dest_sub_mask> <next_hop_ip> <administrative distance>
                                                                                (>1)
```

ככל שהמספר גבוה יותר – העדיפות של הנתיב נמוכה יותר.

הגדרת default route

כדי ליצור נתיב דיפולטיבי עבור כתובות שאינן נמצאות בטבלאת הניווט של הראוטר, נוכל להשתמש ב- default route. נעשה זאת באמצעות שימוש בכתובת ה-IP 0.0.0.0 וה-subnet mask 0.0.0.0 בצורה הבאה:

```
Router>enable
Router# conf t
Router(config)#ip route 0.0.0.0 0.0.0.0 <next_hop_ip>
```

*כמובן שגם ל-default route אפשר ליצור floating route.

הגדרת פרוטוקול OSPF (Dynamic routing)

פרוטוקול ניווט דינאמי בין ראוטרים.

אנו מגדירים עבור כל ראוטר שאנו רוצים שישתמש בפרוטוקול:

1. את עצם השימוש בפרוטוקול, כולל מספר התהליך (נשתמש בלימודים אך ורק בתהליך מספר 1)
2. כל הרשתות המעורבות בניווט המחוברות לראוטר, כולל wild card של כל רשת כזו. כמו כן איזור הניווט (בלימודים נשתמש אך ורק באזור 0).

לא למדנו, ניתן להתעלם נכון לעכשיו

בנוסף, עבור ראוטרים אשר באחד או יותר מקצותיהם מסתיים תהליך הניווט, כלומר, אין ראוטר אלא נמצא end device (למשל, מחשב), או סוויץ' שמאחוריו עומדת LAN:

3. יש לציין את החיבור של הראוטר למכשירים הללו כ-passive-interface. **חשוב לציין! הגדרה זו לא באה במקום הכנסת הרשת של אותו המחשב כמעורבת בתהליך הניווט! יש לקיים גם את הכנסת הרשת וגם את הגדרת ה-passive interface!**

--

חישוב wild card:

כדי למצוא את ה-wild card של רשת נבצע את השלבים הבאים:

1. נבדוק מהו ה-subnet של הרשת הרצויה
2. נחסיר כל אוקטטה ואוקטטה מה-subnet המלא, כלומר מ-255.255.255.255 התוצאה שתתקבל היא ה-wild-card של הרשת.

<u>דוגמא 1:</u>

1. Subnet = 255.255.255.0

2.

```
  255.255.255.255
-  255.255.255.  0
wild card  =  0 . 0 . 0 . 255
```

<u>דוגמא 2:</u>

1. Subnet = 255.255.255.252

2.

```
  255.255.255.255
-  255.255.255.252
wild card =  0 . 0 . 0 . 3
```

--

```
Router>enable
Router# conf t
Router(config)#router ospf <number_of_process (1)>
Router(config-router)#network <net_ID> <wild_card> area <number_of_area (0)>
Router(config-router)#network <another_net_ID> <wild_card> area <number_of_area (0)>
.
.
.
```

במידה וקיימים חיבורי-קצה שאינם לראוטרים – להתעלם נכון לעכשיו!!!:

```
Router(config-router)#passive-interface <name_of_interface>
Router(config-router)#passive-interface <another_name_of_interface>
.
.
.
```

Network Statement Shortcut in OSPF configuration

Below is a simple shortcut in OSPF to advertise all interfaces in **OSPF** routing protocol –

Advertisements

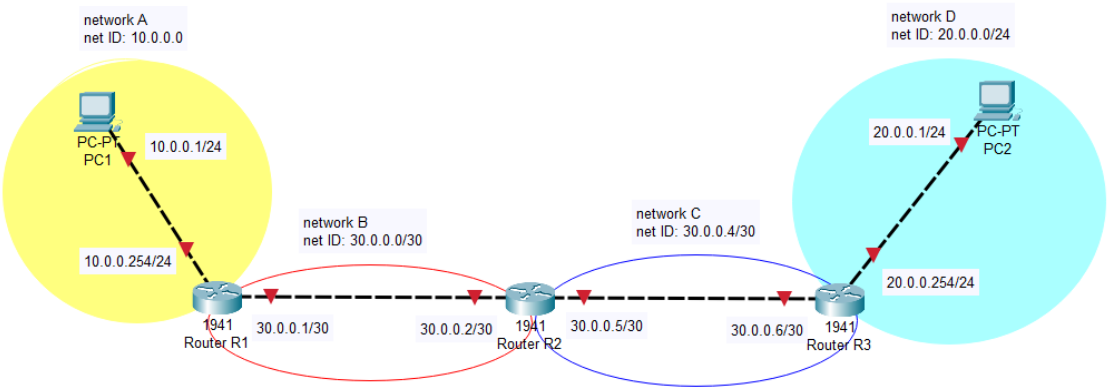
```
R1(config)#router ospf 1
R1(config-router)#network 0.0.0.0 0.0.0.0 area 0
```

Or, other one can be like this –

```
R1(config)#router ospf 1
R1(config-router)#network 0.0.0.0 255.255.255.255 area 0
```

לדוגמא:

- קיים מחשב 1 אשר כתובת ה-IP שלו היא 10.0.0.1/24 (כלומר, כתובת רשת 10.0.0.0). נסמן: רשת A.
- הראוטר R1 מחובר ברשת A למחשב 1 כ-default gateway. מצידו השני, R1 מחובר לראוטר R2 ברשת B, תחת הכתובת 30.0.0.1/30 (כלומר, כתובת רשת 30.0.0.0)
- רטואר R2 נמצא ברשת B תחת הכתובת 30.0.0.2/30. מצידו השני, R2 מחובר לראוטר R3 ברשת C, תחת הכתובת 30.0.0.5/30 (כלומר, כתובת רשת 30.0.0.4).
- רטואר R3 נמצא ברשת C תחת הכתובת 30.0.0.6/30. מצידו השני, הראוטר מחובר למחשב 2 ברשת D כ-default gateway. כתובת רשת: 20.0.0.0/24.
- מחשב 2 מחובר ברשת D לראוטר R3 תחת הכתובת 20.0.0.1/24.



*הטופולוגיה נוצרה רק לשם ההמחשה. לא הוגדרו בה ההגדרות הרצויות, לכן החיבורים לא בהכרח ירוקים.

הגדרה של OSPF בין רשת A לרשת D:

הגדרות ב-R1:

נגדיר את השימוש בפרוטוקול ואת הרשתות המשתתפות בתהליך הניווט:

```
R1>enable
R1# conf t
R1(config)#router ospf 1
R1(config-router)#network 10.0.0.0 0.0.0.255 area 0
R1(config-router)#network 30.0.0.0 0.0.0.3 area 0
```

"נחסום" את PC1- להתעלם!!:

```
R1(config-router)#passive-interface f0/0 (the interface name can differ)
```

הגדרות ב-R2:

נגדיר את השימוש בפרוטוקול ואת הרשתות המשתתפות בתהליך הניווט:

```
R2>enable
R2# conf t
R2(config)#router ospf 1
R2(config-router)#network 30.0.0.0 0.0.0.3 area 0
R2(config-router)#network 30.0.0.4 0.0.0.3 area 0
```

הגדרות ב-R3:

נגדיר את השימוש בפרוטוקול ואת הרשתות המשתתפות בתהליך הניווט:

```
R3>enable
R3# conf t
R3(config)#router ospf 1
R3(config-router)#network 30.0.0.4 0.0.0.3 area 0
R3(config-router)#network 20.0.0.0 0.0.0.255 area 0
```

"נחסום" את PC2 – להתעלם!!:

```
R3(config-router)#passive-interface f0/0 (the interface name can differ)
```

פרוטוקול DHCP

פרוטוקול חלוקת כתובות דינאמי.
לעיתים לא נרצה להגדיר ידנית כתובת IP עבור כל מחשב ומחשב ברשת שלנו, אלא שהכתובות יחולקו באופן דינאמי באמצעות פרוטוקול DHCP. את הפרוטוקול נגדיר בשלב זה על ראוטר, כך שזה יהיה אחראי על חלוקת הכתובות (ניתן להגדיר במקום גם שרת DHCP. לא נעשה זאת בקורס).
לראוטר ישנה אפשרות לחלק כתובות לרשתות הלוקאליות שלו (ה"מחוברות" אליו), אך כל עוד קיים פרוטוקול ניווט, גם עבור רשתות אחרות הרחוקות ממנו (באמצעות הגדרה נוספת).

שלבי הגדרת הפרוטוקול:

1. נגדיר "pool" (כלומר, מאגר כתובות) וניתן לו שם בראוטר המבוקש. לרוב – שם הוילאן של הרשת.
2. נגדיר את הטווח של מאגר הכתובות (כתובת רשת, subnet mask)
3. נציין את הכתובת של ה-default gateway של הרשת הרצוייה
4. נגדיר שרת DNS עבור הרשת הרצוייה (בקורס לא נציין שרת אמיתי. ניתן להשתמש בכתובת של גוגל: 8.8.8.8)
5. נגדיר את הדומיין עבור ה-DHCP (בקורס לא נציין דומיין אמיתי. ניתן להשתמש בכתובת של גוגל: google.com)
6. נחריג מהמאגר את הכתובות בטווח אותן לא נרצה שהראוטר יחלק. כתובת חובה בפקודה זו היא כתובת ה-default gateway של הרשת. בין היתר נרצה להחריג גם כתובות של שרתים, מצלמות וכד'.

במידה והרשת לוקאלית לראוטר, סיימנו את ההגדרה.
במידה והרשת מרוחקת, נבצע את הפקודות הבאות בראוטר הצמוד לרשת המבוקשת:

7. נגרום לראוטר המרוחק להעביר הודעות ברודקאסט מסוג discover (בחלק מתהליך ה-DORA של ה-DHCP):
   -נכנס אל הראוטר המרוחק, ונבחר באינטרפייס של הדיפולט גייטווי של הרשת הרצוייה.
   -נגדיר את אחת מכתובות הכניסה אל הראוטר הכולל את פרוטוקול ה-DHCP.

```
Router>enable
Router# conf t
Router(config)# ip dhcp pool <name_of_pool>
Router(dhcp-config)# network <net_ID> <subnet_mask>
Router(dhcp-config)# default-router <default_gateway_ip>
Router(dhcp-config)# dns-server <dns_server_ip>
Router(dhcp-config)# domain-name <domain_name>
Router(dhcp-config)# exit
Router(config)# ip dhcp excluded address <ip_address>
```

המשך עבור רשת מרוחקת:

```
Router2>enable
Router2# conf t
Router2(config)# int <name_of_deafault_gateway_interface>
Router2(config-if)# ip helper-address <ip_address>
```

לדוגמא:

- ראוטר1 אחראי לפרוטוקול DHCP.
- בין ראוטר1 לראוטר2 קיימת רשת שכתובתה 10.0.0.0/30, כך שראוטר1 הוא בעל הכתובת 10.0.0.1 וראוטר2 הוא בעל הכתובת 10.0.0.2
- ראוטר2 משתמש באינטרפייס g0/0.10 כ-default gateway של הרשת 192.168.1.0/24 המוגדרת כ-VLAN 10.

נגדיר ב-router1:

```
Router1>enable
Router1# conf t
Router1(config)# ip dhcp pool vlan10
Router1(dhcp-config)# network 192.168.1.0 255.255.255.0
Router1(dhcp-config)# default-router 192.168.1.254
Router1(dhcp-config)# dns-server 8.8.8.8
Router1(dhcp-config)# domain-name google.com
Router1(dhcp-config)# exit
Router1(config)# ip dhcp excluded address 192.168.1.254
```

המשך עבור רשת מרוחקת:

נגדיר ב-router2:

```
Router2>enable
Router2# conf t
Router2(config)# int g0/0.10
Router2(config-if)# ip helper-address 10.0.0.1
```

פרוטוקול NAT

פרוטוקול המאפשר יציאה לרשת של מספר רשתות ומחשבים תחת כתובת יחידה – כתובת יציאה מראוטר נבחר.
הפרוטוקול מאפשר את המשך השימוש בכתובות IPv4, בכך שהוא "חוסך" כתובת IP בעת היציאה לרשת.

*כיצד הפקטות יודעות את דרכן חזרה אל המחשב השולח אם כולן יוצאות אל הרשת תחת אותה הכתובת?
לכל מחשב ומחשב כתובות MAC ייחודית. בעת ההגעה חזרה אל הראוטר, הניווט אל המחשב השולח יתבצע באמצעות כותבת זו.

שלבי הגדרת הפרוטוקול:

בראוטר הקיצון שמוציא אותנו מהרשת נגדיר:

1. "אישור" עבור כל אינטרפייס אשר מאחוריו כתובות אותן נרצה לתרגם לכתובות חיצונית.
2. "אישור" עבור האינטרפייס אשר בכתובתו נרצה להשתמש כדי להוציא את הפקטות מהרשתות הנ"ל.
3. רשימת ACL שתאשר קבלה של התרגום עבור הרשתות שהזנו.
4. "אישור" העמסה של כל הכתובות אותן נרצה לתרגם לכתובת חיצונית על האינרפייס הנבחר.

```
Router>enable
Router# conf t

Router(config)# int <name_of_interface>
Router(config-if)# ip nat inside (הכתובות מאחורי אינטרפייס זה יתורגמו לכתובת חיצונית)
Router(config-if)# int <another_name_of_interface>
Router(config-if)# ip nat inside (הכתובות מאחורי אינטרפייס זה יתורגמו לכתובת חיצונית)
.
.
.

Router(config-if)# int <name_of_interface>
Router(config-if)# ip nat outside (הכתובת של אינטרפייס זה תשמש ככתובת החיצונית)

Router(config-if)# exit
Router(config)# access-list <num_of_ACL> permit <allowed_adresses> <wild_mask>
Router(config)# ip nat inside source list <num_of_acl> interface <name_of_int> overload
```

לדוגמא:

לראוטר1 מחוברות רשתות באינטרפייסים הבאים:

- g0/0
- g0/1.10
- g0/1.20

כל שסך הכל כולן כלולות תחת הכתובת 192.168.1.0/24

כמו כן, מחובר אל הראוטר באינטרפייס g0/2 ראוטר נוסף. דרך הכתובת של כניסה זו נרצה להוציא את הפרטות מהכתובות של הרשתות המצויינות למעלה.

```
Router1>enable
Router1# conf t

האינטפייסים שמאחוריהם הכתובות שנרצה שיתורגמו לכתובת חיצונית:
Router1(config)# int g0/0
Router1(config-if)# ip nat inside
Router1(config-if)# int g0/1.10
Router1(config-if)# ip nat inside
Router1(config-if)# int g0/1.20
Router1(config-if)# ip nat inside

האינטרפייס שבכתובתו נשתמש ככתובת החיצונית:

Router1(config-if)# int g0/2
Router1(config-if)# ip nat outside
Router1(config-if)# exit
יצירת ACL:

Router1(config)# access-list 1 permit 192.168.1.0 0.0.0.255

העמסת הכתובות:

Router1(config)# ip nat inside source list 1 interface g0/2 overload
```

פרוטוקול PVST

פרוטוקול זה מאפשר מעין "רוטציה" בין הסוויצ'ים ברשת על תפקיד ה-root bridge בעת פרוטוקול STP, במטרה ליצור חלוקת עומסים ברשת. עבור כל וילאן, נגדיר מיהו הסוויץ' שישמש כ-root bridge בעת פעילותו, וכך פורטים שונים ייסגרו עבור כל וילאן (כלומר, בכל פעם נתיב אחר יהיה חסום לתעבורה).

שלבי הגדרת הפרוטוקול:

1. הגדרת פרוטוקול VTP
פרוטוקול זה מפיץ וילאנים הנמצאים על סוויץ' ראשי המוגדר כסרבר, אל הסוויצ'ים המחוברים אליו המוגדרים כקליינטים.
   a. תחילה נגדיר את הפרוטוקול בסוויץ' (נציין דומיין [בקורס: google.com] וסיסמא) אותו נבחר כסרבר. לאן מכן נגדיר את שאר הסוויצ'ים בקליינטים (בדרת המחדל של כל סוויץ' היא להיות סרבר, לכן יש להסב את שאר הסוויצ'ים), ונגדיר עליהם את הפרוטוקול (נגדיר דומיין וסיסמא זהים!).
   b. נגדיר על הסוויץ' אותו בחרנו כסרבר את כל הוילאנים הקיימים ברשת.
   c. נגדיר טראנק בין החיבורים בין סוויץ' לסוויץ'. הגדרה מצד מאינטרפייס בצידו האחד של החיבור יוצרת את הטראנק גם בין צידו השני! אין צורך להגדיר בשני הצדדים.

2. נגדיר את פרוטוקול ה-PVST.
נגדיר עבור כל סוויץ' שישמש כ-root bridge כאשר ישנה תעבורה בין וילאנים המחוברים אליו ישירות.

נגדיר את בסוויץ' הסרבר את פרוטוקול VTP:

```
SwitchServer>enable
SwitchServer# conf t
SwitchServer(config)# vtp domain <domain_name>
SwitchServer(config)# vtp password <password>
```

נגדיר את את שאר הסוויצ'ים בקליינטים ונגדיר עליהם את פרוטוקול ה-VTP:

```
SwitchClient1>enable
SwitchClient1# conf t
SwitchClient1# vtp mode client
SwitchClient1(config)# vtp domain <same_domain_name>
SwitchClient1(config)# vtp password <same_password>

SwitchClient2>enable
SwitchClient2# conf t
SwitchClient2# vtp mode client
SwitchClient2(config)# vtp domain <same_domain_name>
SwitchClient2(config)# vtp password <same_password>

.
.
.
```

נגדיר על הסוויץ' הסרבר את כל הוולנים הקיימים:

```
SwitchServer>enable
SwitchServer# conf t
SwitchServer(config)# vlan <vlan number>
SwitchServer(config-vlan)# exit

SwitchServer(config)# vlan <another vlan number>
SwitchServer(config-vlan)# exit

.
.
.
```

נגדיר טרנקים בין הסוויצ'ים:

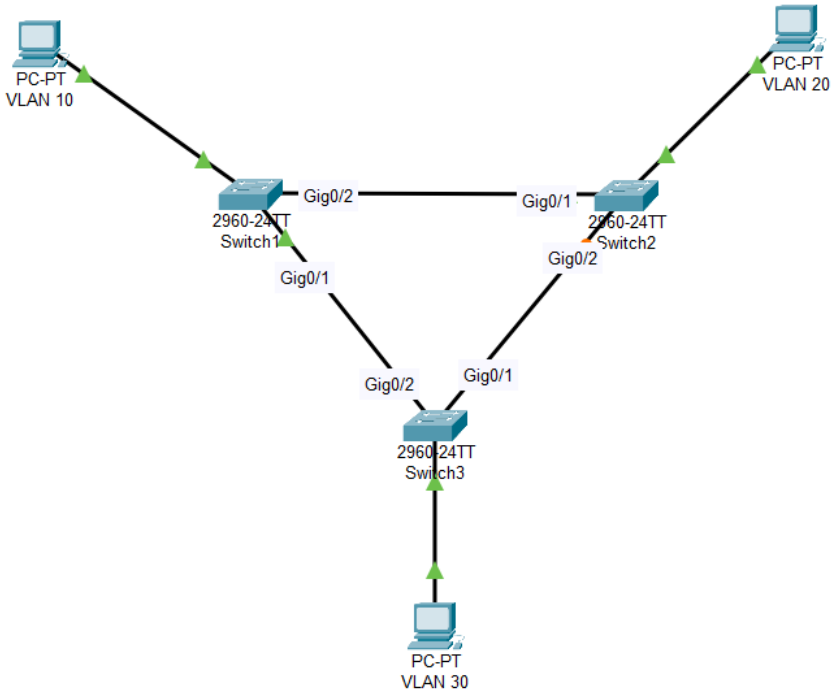
```
Switch>enable
Switch# conf t
Switch(config)# int <name_of_interface>
Switch(config-if)# switchport mode trunk

AnotherSwitch>enable
AnotherSwitch# conf t
AnotherSwitch(config)# int <name_of_interface>
AnotherSwitch(config-if)# switchport mode trunk

.
.
.
```

נגדיר את פרוטוקול PVST על כל סוויץ':

```
Switch>enable
Switch# conf t
Switch(config)# spanning-tree vlan <vlan_number> root primary
Switch(config)# spanning-tree vlan <another_vlan_number> root primary
.
.
.


AnotherSwitch>enable
AnotherSwitch# conf t
AnotherSwitch(config)# spanning-tree vlan <vlan_number> root primary
AnotherSwitch(config)# spanning-tree vlan <another_vlan_number> root primary
.
.
.



.
.
.
```

לדוגמא:

נתונה הטופולוגיה הבאה:



*הטופולוגיה נוצרה רק לשם ההמחשה. לא הוגדרו בה ההגדרות הרצויות, לכן החיבורים לא בהכרח ירוקים.

נבחר בסוויץ'1 כסרבר ונגדיר פרוטוקול VTP:

```
Switch1>enable
Switch1# conf t
Switch1(config)# vtp domain google.com
Switch1(config)# vtp password 123456
```

נגדיר את את שאר הסוויצ'ים בקליינטים ונגדיר עליהם את פרוטוקול ה-VTP:

בסוויץ'2:

```
Switch2>enable
Switch2# conf t
Switch2# vtp mode client
Switch2(config)# vtp domain google.com
Switch2(config)# vtp password 123456
```

בסוויץ'3:

```
Switch3>enable
Switch3# conf t
Switch3# vtp mode client
Switch3(config)# vtp domain google.com
Switch3(config)# vtp password 123456
```

נגדיר על הסוויץ' הסרבר, כלומר על סוויץ' 1 את כל הוילנים ברשת

```
Switc1ו>enable
Switch1# conf t
Switch1(config)# vlan 10
Switch1(config-vlan)# exit
Switch1(config)# vlan 20
Switch1(config-vlan)# exit
Switch1(config)# vlan 30
Switch1(config-vlan)# exit
```

נגדיר טרנקים בין הסוויצ'ים:

```
Switch1>enable
Switch1# conf t
Switch1(config)# int g0/1
Switch1(config-if)# switchport mode trunk
Switch1(config)# int g0/2
Switch1(config-if)# switchport mode trunk
```

נגדיר את פרוטוקול PVST על כל סוויץ':

בסוויץ1:

```
Switch1>enable
Switch1# conf t
Switch1(config)# spanning-tree vlan 10 root primary
```

בסוויץ'2:

```
Switch2>enable
Switch2# conf t
Switch2(config)# spanning-tree vlan 20 root primary
```

בסוויץ'3:

```
Switch3>enable
Switch3# conf t
Switch3(config)# spanning-tree vlan 30 root primary
```

הגדרת portfast ו-BPDU

1. **Portfast**

הגדרה שניתן לשים על פורטים בסוויץ' המחוברים למכשירי קצה, במטרה לבטל את שלב ה-listening של הפורט בעת העברת פקטות BPDU. הסיבה לכך היא שהמכשיר המחובר לפורט אינו סוויץ' נוסף, ולכן אין צורך בשלב זה, וניתן לחסוך את הזמן שהתהליך לוקח.

שלבי הגדרת ה-portfast:

1. כניסה לפורט הרצויי
2. כתיבת ההגדרה

```
Switch> enable
Switch# conf t
Switch(config)#int <interface_name>
Switch(config-if)# spanning-tree portfast
```

2. **BPDU guard**

הגדרה זו מונעת מהודעות BPDU להגיע למכשיר קצה, במטרה למנוע שיתוף מידע על מצב וסטטוס הסוויצ'ים בטופולוגיה. חשוב, למשל, במצב בו פורץ נמצא באחד מהמחשבים בטופולוגיה ועשוי לקבל מידע על הסוויצ'ים באמצעות הודעות BPDU. אם נחסום הגישה – המידע לא יוכל להגיע לידיו.

** בכדי להפעיל הגדרה זו, על הפורטים הרצויים להיום במצב Portfast!

שלבי הגדרת BPDU guard:

```
Switch> enable
Switch# conf t
Switch(config)#spanning-tree portfast bpduguard default
```

פרוטוקול SSH

פרוטוקול המאפשר השתלטות מאובטחת מרחוק על מכשיר (סוויץ', ראוטר, מחשב וכו').
המידע העובר באמצעות פרוטוקול זה הינו מוצפן.

שלבי הגדרת ה-SSH על סוויץ':

1. נגדיר host name עבור הסוויץ'
2. נפעיל את הפורטים הוירטואליים של ה-VTY. ישנם 16 פורטים במספר, אשר כל אחד מהם מאפשר חיבור SSH יחיד ל-user. (בקורס – נגדיר את כל הפורטים, כלומר 0-15)
3. בתוך הגדרת ה-VTY נאשר את הפעלת ה-SSH.
4. נצא, ונגדיר domain עבור הסוויץ' (בקורס: google.com)
5. נגדיר את הצפנת ה-RSA של ההודעות. (בקורס נבצע הצפנה באורך של 2048 ביטים, ההצפנה הסטנדרטית).
6. נכנס אל הפורט vlan 1
7. נגדיר לו כתובת IP (בקורס: 100.0.0.1) ו-subnet mask (בקורס: 255.255.255.252).
8. נדליק את הפורט
9. נגדיר את כתובת ה-default gateway
10. נבחר שם משתמש וסיסמא עבור ה-SSH

בכדי שנוכל להעביר תקשורת מ-vlan1, עלינו לקנפג לו router on a stick ולאחר מכן גם לפרסם אותו בראוטר המתאים כחלק מפרוטוקול ה-ospf.

בסוויץ':

```
Switch> enable
Switch# conf t
Switch(config)# hostname <name_of_host>
Switch(config)# line vty 0 15
Switch(config-line)# transport input ssh
Switch(config-line)# login local
Switch(config-line)#exit
Switch(config)# ip domain name <name_of_domain>
Switch(config)# crypto key generate rsa general-keys modulus 2048
Switch(config)# int vlan 1
Switch(config-if)# ip address <ip_address_of_vlan1> <Subnet_mask_of_vlan1>
Switch(config-if)# no shutdown
Switch(config-if)# exit
Switch(config)# Ip default-gateway <default_gateway_of_vlan1>
Switch(config)# user <user_name> secret <password>
```

```
((שלבי הגדרה על הראוטר:))
((ROAS:))

Router>enable
Router# conf t
Router(config)# int <interface_name>.1
Router(config-subif)# encapsulation dot1q 1
Router(config-subif)# ip address <default_gateway_vlan1> <subnet_mask_vlan1>


((OSPF:))
Router(config)#router ospf <number_of_process (1)>
Router(config-router)#network <vlan1_net_ID> <vlan1_wild_card> area <number_of_area (0)>
```

כניסה ל-SSH ממחשב מרוחק:

```
במחשב:

C:\> ssh -l <user_name> "ssh_in_vlan1_ip”
Password: <password>
```

לדוגמא:

נגדיר ssh על סוויץ'1, המחובר לראוטר1 בפורט g0/0:

```
בסוויץ'1:
Switch1> enable
Switch1# conf t
Switch1(config)# hostname S1
S1(config)# line vty 0 15
S1(config-line)# transport input ssh
S1(config-line)# login local
S1(config-line)#exit
S1(config)# ip domain name google.com
S1(config)# crypto key generate rsa general-keys modulus 2048
S1(config)# int vlan 1
S1(config-if)# ip address 100.0.0.1 255.255.255.252
S1(config-if)# no shutdown
S1(config-if)# exit
S1(config)# Ip default-gateway 100.0.0.2
Switch(config)# user google secret 123456
```

בראוטר1:
ROAS:

```
Router1>enable
Router1# conf t
Router1(config)# int g0/0.1
Router1(config-subif)# encapsulation dot1q 1
Router1(config-subif)# ip address 100.0.0.2 255.255.255.252>
```

OSPF:

```
Router(config)#router ospf 1
Router(config-router)#network 100.0.0.0 0.0.0.3 area 0
```

במחשב:

```
C:\> ssh -l google 100.0.0.1
Password: 123456
```

Port security

הליך אבטחתי המאפשר הגדרה של מכשירי קצה ספציפיים המאושרים לחיבור לפורט מסוים. במידה ומתחבר לפורט מכשיר קצה שאינו מאושר – הפורט יכבה.

שלבי הגדרת port security על סוויץ' עבור פורט מסוים:

1. נכנס אל הפורט הרצוי בסוויץ'
2. נגדיר את המצב של הפורט ל-access
3. נכנס להגדרות port-security
4. נגדיר את מספר המחשבים אשר להם חיבור מאושר לפורט (בקורס- 1)
5. נבדוק מה כתובת ה-mac של המחשב הרצוי
6. נכניס בהגדרות ה-port security את כתובת המק הרצוייה
7. נגדיר מה יקרה בעת חיבור לא רצוי – כיבוי האינטרפייס.
8. נכבה ונדליק מחדש את הפורט

בסוויץ':

```
Switch> enable
Switch# conf t
Switch(config)# int <interface_name>
Switch(config-if)# switchport mode access
Switch(config-if)# switchport port-security
Switch(config-if)# switchport port-security maximum <number_of_allowed_mac_addresses>
Switch(config-if)# switchport port-security mac-address <mac_address>
Switch(config-if)# switchport port-security violation shutdown
Switch(config-if)# shutdown
Switch(config-if)# no shutdown
```

פקודה למציאת כתובת mac של מחשב:

במחשב:

```
C:\> ipconfig /all
```

הגדרת סיסמאות

לעיתים נרצה להגדיר סיסמאות עבור ה-privilege levels השונים במכשירים המקשרים (סוויצ'ים, ראוטרים).

שלבי הגדרת הסיסמאות:

1. סיסמא ל-User EXEC mode :
   a. דרך ה-Global Configuration mode נכנס ל-line console 0
   b. נגדיר סיסמא
   c. נבצע אישור לסיסמא

2. סיסמא ל-Privileged EXEC mode :
   a. דרך ה-Global Configuration mode נגדיר סיסמא.

סיסמא ל-User EXEC mode :

```
Switch> enable
Switch# conf t
Switch(config)# line console 0
Switch(config-line)# password <password>
Switch(config-line)# login
Switch(config-line)# exit
```

סיסמא ל-Privileged EXEC mod :

```
Switch(config)# enable secret <password>
```

MOTD

*****שלב לא הכרחי!!!!*****

כתיבת הודעה שתוצג בעת פתיחת ה-CLI של מfשיר מקשר נבחר

סיסמא ל-User EXEC mode :

```
Switch> enable
Switch# conf t
Switch(config)# banner motd #<message>#
```

Standard Named ACL

ACL משמש כ"מסנן" פקטות עבור ראוטר. הוא יכול לאפשר או לחסום גישה של פקטות מסוגים שונים וממקורות שונים ליעדים שונים.

באמצעות standard ACL ניתן לאפשר או לחסום גישה של פקטות אך ורק ע"פ כתובת המקור (source), ורק באמצעות כתובת מסוג IPv4. כלומר, אנו יכולים לקבוע, למשל, "פקטות ממחשב שכתובתו x.x.x.x לא יכולות לעבור" אך לא "פקטות ממחשב שכתובתו x.x.x.x לא יכולות לעבור בראוטר אל מחשב שכתובתו y.y.y.y" (זה אפשרי ב-extended ACL).

<u>תהליך כללי של כתיבת standard Named ACL:</u>
תחילה, נקים את הACL בראוטר המחובר ליעד ונעניק לו שם. נכתוב את ההגדרות הרצויות (שורה לאחר שורה) ולאחר מכן נצמיד את הרשימה ל-interface ספציפי, הקרוב ביותר אל היעד. לבסוף, נקבע כי היציאה של הפקטות מן הראוטר החוצה לכיוון המטרה אסורה (או מותרת. כתלות ב-ACL).

* כאשר אנו רוצים לחסום כתובת מקור של מחשב ספציפי (ולא רשת) עלינו להכניס wild card של 0.0.0.0 או לכתוב host ולאחריו את כתובת ה-IP הרצויה.

* הרשימה נקראת מלמעלה למטה. **יש חשיבות לסדר!** לאחר שהזנו האם לדחות או לאפשר מעבר לפקטות מרשתות/מחשבים ספציפיים, **עלינו לכתוב ACE המתייחס לפקטות משאר המקורות שנותרו ("כל השאר")**. בתור ברירת מחדל, בסוף כל רשימה מופיע "deny any", כלומר, במקרה של white list – עלינו לא לעשות דבר. מנגד, במקרה של "black list", עלינו להוסיף "permit any".

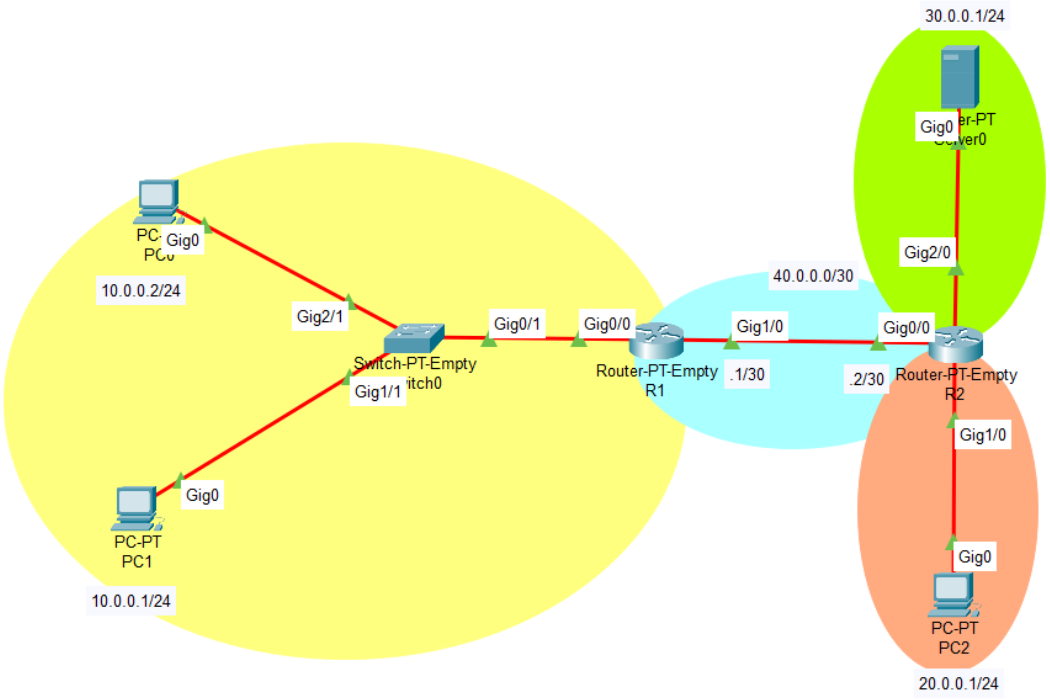
הגדרת Access list:

```
Router> enable
Router # conf t
Router(config)#ip access-list standard <name_of_ACL>


Router(config-std-nacl)# <permit\deny> <ip_address> <wild_card>
                                                    any

Router(config-std-nacl)# <permit\deny> <another ip_address> <wild_card>
                                                    any
.
.
.
Router(config-std-nacl)# exit
```

שיוך ל-interface ספציפי:

```
Router(config)# int <interface_name>
Router(config-if)# ip access-group <name_of_ACL> out
```

לדוגמא:
קיימת הרשת הבאה:



אנו רוצים לאפשר את הגישה לשרת (צד ימין למעלה) אך ורק לרשת הכתומה (רשת 20.0.0.0, כרגע נמצא בה רק מחשב בעל כתובת 20.0.0.1) ולמנוע גישה אליה מכל השאר. כלומר, נייצר white-list ב-R2, שיאפשר העברת פקטות לשרת אך ורק מרשת 20.0.0.0 .

```
                                                    הגדרת Access list ב-R2:

R2> enable
R2# conf t
R2(config)#ip access-list standard WHITE20


R2(config-std-nacl)# permit 20.0.0.0 0.0.0.255
R2(config-std-nacl)# exit

                                                    נשייך את ה-Access-list ל-interface:
R2(config)# int g2/0
R2(config-if)# ip access-group WHITE20 out
```

Extended Named ACL

ACL משמש כ"מסנן" פקטות עבור ראוטר. הוא יכול לאפשר או לחסום גישה של פקטות מסוגים שונים וממקורות שונים ליעדים שונים.

באמצעות extended ACL אנו יכולים למנוע מעבר של פקטות באמצעות הראוטר ע"פ מקור (מאיפה הגיע הפקטה) ויעד (לאן היא נשלחת). כמו כן, אנו יכולים לחסום מעבר של פקטות לא רק ע"פ IP (כמו שקורה ב-standard ACL) אלא גם ע"פ סוגי פרוטוקולים, פורטים וכו'.

בשונה מ-standard ACL, ב-extended ACL יש ליצור את הACL כמה שיותר קרוב אל המקור (כלומר, בראוטר המחובר אליו), ולמנוע *כניסה* של הפקטות מהראוטר (ולא יציאה מראוטר כמו שעשינו ב-standard ACL).

<u>חשוב</u>
ב-extended ACL ישנן המון אפשרויות לביצוע. לא אוכל להכניס את כולן כאן, ולכן אני ממליצה להשתמש ב"?" תוך כדי כתיבת הפקודה כדי להבין מהן האפשרויות הבאות, ולבחור את הרצויה עבורכם.

הגדרה כללית של extended ACL עבור כתובות IP:

הגדרת Access list:

```
Router> enable
Router # conf t
Router(config)#ip access-list extended <name_of_ACL>
Router(config-ext-nacl)# permit\deny> ip <source_ip> <wild_card>  <dest_ip> <wildcard>
                                                any                     any

.
.
```

שיוך ל-interface ספציפי:

```
Router(config)# int <interface_name>
Router(config-if)# ip access-group <ACL_name> in
```

הגדרה כללית של extended ACL עבור פרוטוקול ספציפי בפורט ספציפי:

הגדרת Access list:

```
Router> enable
Router # conf t
Router(config)#ip access-list extended <name_of_ACL>
Router(config-ext-nacl)# <permit\deny> tcp <source_ip> <wild_card> eq <source_port_num>
                                                  any
```

המשך פקודה!!:

```
<dest_ip> <wildcard> eq <dest_port_num>

          Any

.
.
.
```

שיוך ל-interface ספציפי:

```
Router(config)# int <interface_name>
Router(config-if)# ip access-group <name_of_ACL> in
```

* במקום eq (=equal) ניתן להתייחס גם לטווחים של פורטים, לפי העיקרון הבא:

- **eq 80** = equal to port 80
- **gt 80** = greater than 80 (81 and greater)
- **lt 80** = less than 80 (79 and less)
- **neq 80** = NOT 80
- **range 80 100** = from port 80 to port 100

* מספרי פורטים מוכרים:

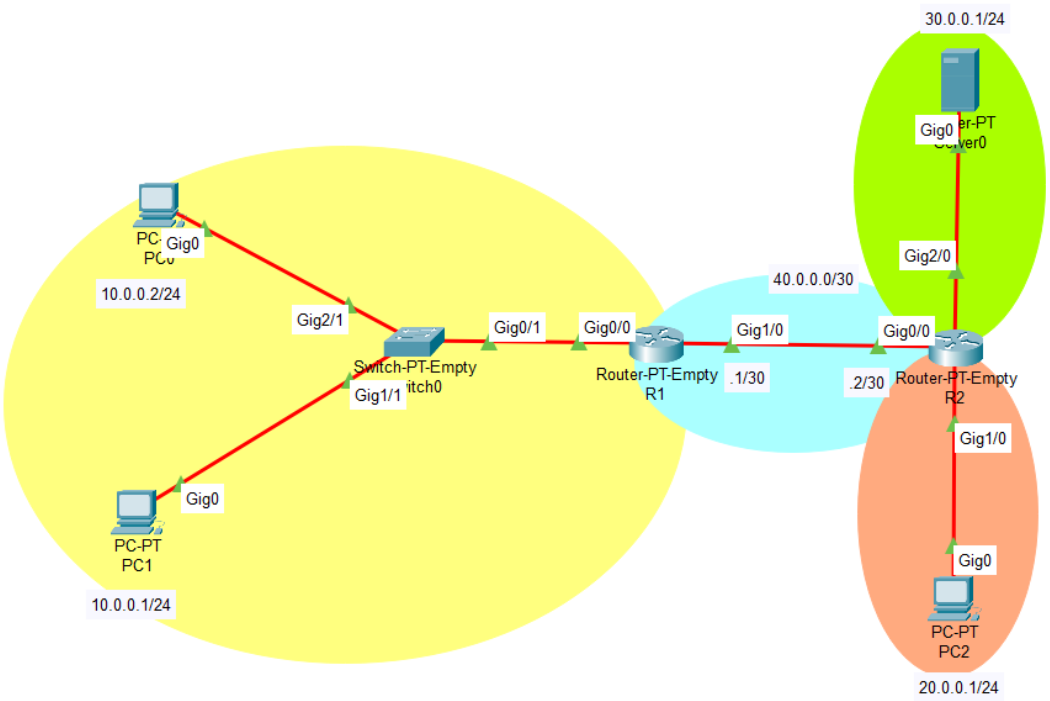
TCP
- FTP data (20)
- FTP control (21)
- SSH (22)
- Telnet (23)
- SMTP (25)
- HTTP (80)
- POP3 (110)
- HTTPS (443)

UDP
- DHCP server (67)
- DHCP client (68)
- TFTP (69)
- SNMP agent (161)
- SNMP manager (162)
- Syslog (514)

TCP & UDP
- DNS (53)

דוגמא:

קיימת הרשת הבאה:



1. חסימת מעבר ע"פ כתובות IP:

נחסום את הגישה של הרשת 20.0.0.0 אל המחשב הספציפי 10.0.0.1 (ברשת 10.0.0.0). כלומר, נייצר black-list ב-R2 שתמנע את הגישה מהרשת הספציפית (20.0.0.0) בלבד ותאפשר גישה לכל השאר.

```
                                                    הגדרת Access list ב-R1:

R1> enable
Router # conf t
Router(config)#ip access-list extended BLACK20
Router(config-ext-nacl)# deny ip 20.0.0.0 0.0.0.255 10.0.0.1 0.0.0.0
Router(config-ext-nacl)# permit ip any any

                                                    נשייך את ה-Access-list ל-interface:
R1# int g1/0
R1(config-if)# ip access-group BLACK20 in
```

2. חסימת מעבר מפורט 80 (http) משרת HTTP אל מחשב (=חסימת גישה לאינטרנט עבור מחשב ספציפי)

נחסום את המעבר של פקטות מפורט 80 של שרת ה-HTTP (השרת העליון מימין) לפורט 80 של המחשב בעל כתובת ה-IP 20.0.0.1 (רשת 20.0.0.0). נבצע את החסימה ב-R2.

```
                                                    הגדרת Access list ב-R2:
R2> enable
R2 # conf t
R2(config)#ip access-list extended noINTERNET
R2(config-ext-nacl)# deny tcp 30.0.0.1 0.0.0.0 eq 80 host 20.0.0.1 eq 80
R2(config)# permit ip any any

                                                    נשייך את ה-Access-list ל-interface:
R2# int g2/0
R2(config-if)# ip access-group noINTERNET in
```