

## Lucrarea de laborator N3

### Protocolul STP. Tehnologia Etherchannel

**Scopul principal** al acestei lucrări este de a ilustra modul de funcționare al protocolului STP și al tehnologiei Etherchannel.

#### Obiective:

- A explica conceptele de agregare, fiabilitate, balansare a traficului
- A arăta modul de funcționare a protocolului STP și în special a versiunii actuale PVST+ a acestuia
- A ilustra posibilitățile tehnologiei Etherchannel, care permite folosirea simultană a mai multor link-uri între două dispozitive pentru transmiterea simultană a traficului

#### Protocolul STP – Spanning Tree Protocol

Vom analiza cum se rezolvă problema formării ciclurilor în rețea, atunci când se construiesc topologii Ethernet fiabile, care să funcționeze la ieșirea din funcțiune a unor componente vitale ale rețelei, prin inserarea unor linii de comunicații suplimentare (redundante).

##### I. Pentru ce este necesar STP?

Cisco recomandă utilizarea următorului model ierarhic pe 3 nivele (a se vedea Figura 1) pentru construcția rețelelor de dimensiuni mari și medii (modelul este aplicat chiar dacă rețeaua este de ordinul a câteva sute de host-uri)

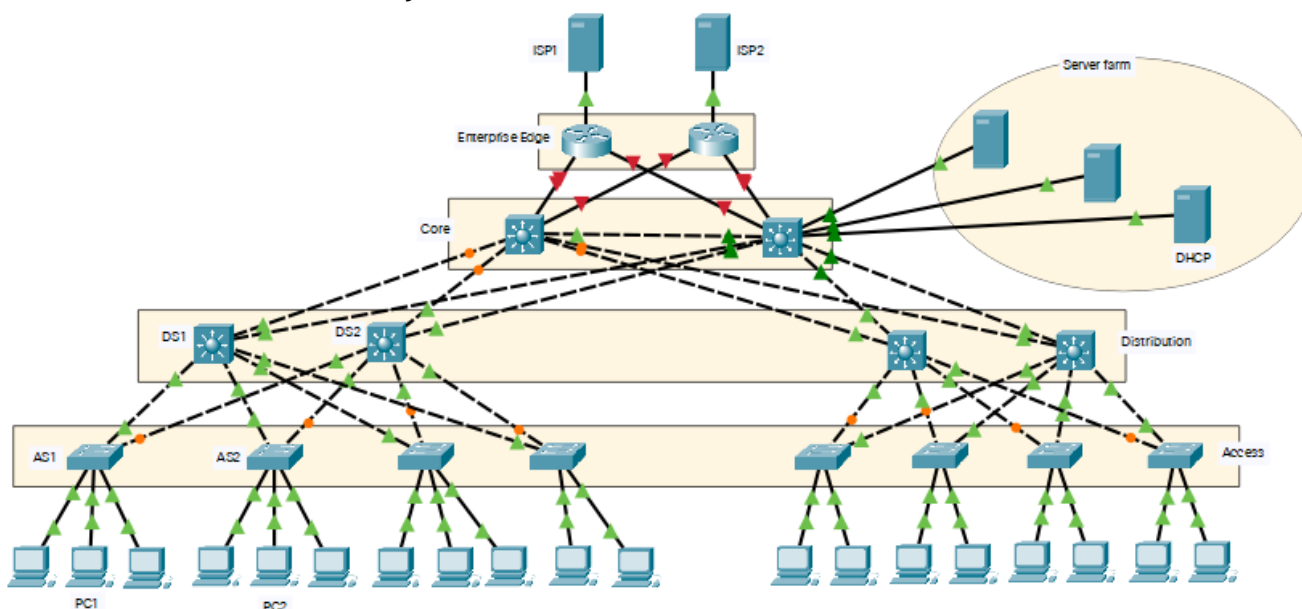


Figura 1

Din figură se vede că fiecare switch de la nivelul de Access este conectat la două switch-uri de la nivelul Distribution.

Dacă un switch de la nivelul Distribution iese din funcțiune => legătura nu se va pierde.

Dar pentru a asigura conexiune, în caz de avarie, au fost folosite linii suplimentare (redundante).

Cablurile încrucișate sunt suplimentare – dar fără de acestea nu este posibilă realizarea unei astfel de scheme.

Așadar, există problema: cablurile încrucișate formează un ciclu în rețea

Dacă PC1 vrea să afle adresa MAC a lui PC2 => se trimite o solicitare ARP spre AS1 => acest pachet broadcast ajunge la DS1 și DS2 => de la DS1 la AS2 => de la AS2 la DS2 => de la DS2 la AS1 => astfel frame-ul s-a întors la AS1 => procesul se repetă => se formează un ciclu (frame-ul broadcast parcurge până la infinit acest ciclu).

În rețele este absolut necesar de implementat o schemă ce asigură stabilitate în funcționare

Switch-ul de la nivelul Distribution poate deservi un număr mare de switch-uri de la nivelul Access => iar în fiecare switch de nivel Access avem câte 48 de porturi => astfel, switch-ul de nivel Distribution

deservește un număr mare de host-uri => dacă switch-ul de nivel Distribution iese din funcțiune, atunci se va crea un mare disconfort pentru utilizatori

Însă dacă este prezent încă un switch de nivel Distribution, conectat la switch-ul de nivel Access, atunci acesta preia traficul => problema este rezolvată la acest moment

În Figura 2 se vede că dacă dispăre legătura dintre switch-uri => dispăre legătura dintre PC1 și PC2



Figura 2

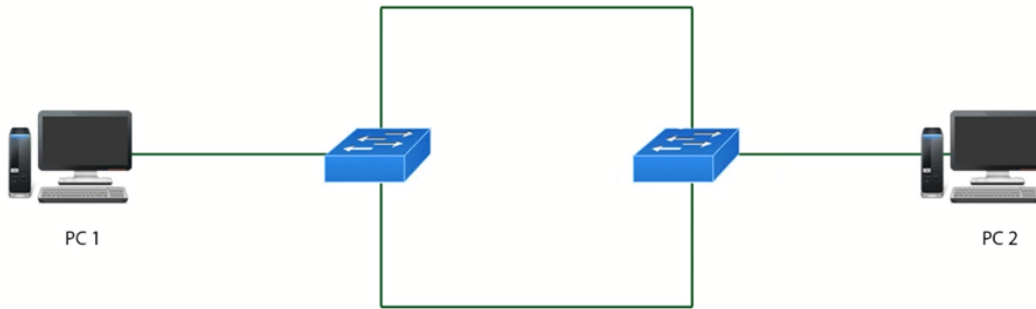


Figura 3

Pot fi utilizate două cabluri (a se vedea Figura 3) => apare o ciclare în comutarea pachetelor => frame-ul va parcurge până la infinit acest ciclu, până nu vom reîncărca switch-ul

Frame-ul broadcast de la PC1 => Sw1 => Sw1 trimite două copii ale frame-ului la Sw2 => PC2 primește două copii ale frame-ului

Două copii ale frame-ului vor trece permanent de la Sw1 la Sw2 și înapoi

PC1 și PC2 permanent formează frame-uri broadcast => acestea rămân în ciclul corespunzător pe care îl parcurg până la infinit => în rezultat, în ciclu se acumulează frame-uri => peste ceva timp canalul se va umple la 100 % => va avea loc furtuna broadcast

În canale se acumulează frame-uri și crește încărcarea asupra dispozitivelor de rețea => rețeaua nu va funcționa

Dacă pe switch toate indicatoarele clipească simultan => rezultă că a avut loc furtuna broadcast

O altă problemă – tabelul MAC al switch-ului este completat incorect

De exemplu, pentru switch-ul Sw1: mai întâi adresa MAC a lui PC1 se atribuie lui Fa 0/1, după care lui Fa 0/2, Fa 0/3 ș.a.m.d. => tabelul de adrese MAC permanent se modifică

La fel și cu Sw2

=>

1) refuz în deservirea rețelei din cauza încărcării acesteia => furtuna broadcast (canalele sunt înfundate cu pachete broadcast)

2) instabilitatea tabelului cu adrese MAC

Cum pot fi soluționate aceste probleme, dar astfel încât să rămână două cabluri, adică să fie asigurată funcționarea rețelei în caz de avarie? => Protocolul *Spanning Tree Protocol* (STP)

STP – mecanismul ce permite eliminarea ciclurilor în rețea

Dacă pe switch-urile din Figura 3 va rula protocolul STP, atunci acesta va elimina ciclul prin blocarea unui port de switch (blocarea canalului). Dacă o conexiune se va întrerupe, atunci protocolul STP va debloca a doua conexiune

La fel în Figura 1 link-ul dintre AS1 și DS2 va fi deconectat

Dacă DS1 va înceta să funcționeze => protocolul STP va debloca link-ul blocat anterior => AS1 va putea funcționa în continuare

Dacă vom conecta trei switch-uri ca în Figura 4, atunci rețeaua nu va funcționa imediat (avem cercuri de culoare orange), ci peste 15-30 de secunde => din cauza că rulează protocolul STP, care verifică dacă în rețea există cicluri => va vedea că nu există cicluri și va debloca porturile

Dacă vom conecta Sw1 cu Sw2 (a se vedea Figura 5) => rețeaua iarăși va înceta să funcționeze (culoare

orange) => protocolul STP verifică dacă există cicluri => pe Sw1 și Sw2 activează ambele porturi, iar pe Sw0 – activează doar unul din cele două porturi  
Se blochează nu tot link-ul, ci doar portul

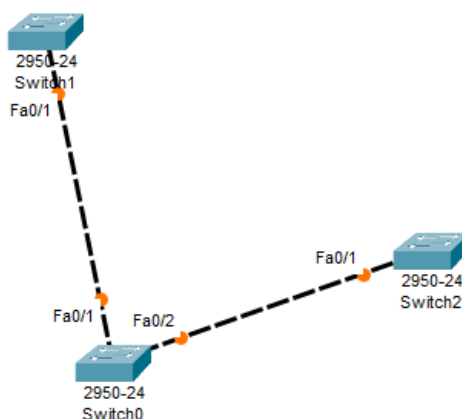


Figura 4

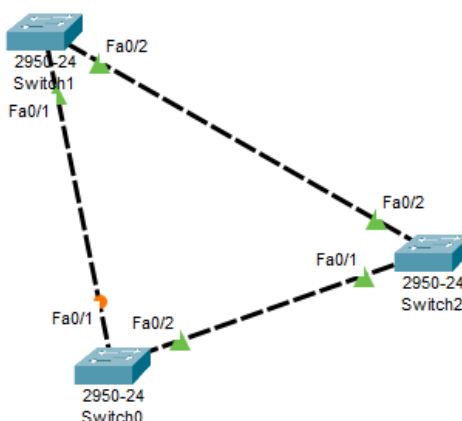


Figura 5

Întrebare: de ce a fost blocat anume acel port și nu altul?

Este necesar să se știe cum poate fi dirijat protocolul STP, astfel încât să fie blocat link-ul cel mai puțin eficient

II. Cum funcționează protocolul STP?

Prima etapă: Selectarea switch-ului rădăcină (numit root). Celelalte switch-uri stabilesc rute până la switch-ul rădăcină

Etapă a doua: Selectarea porturilor rădăcină (root) = porturi care „privesc direct” spre switch-ul rădăcină (a se vedea Figura 6)

Figura 6	Figura 7	Figura 8

Etapă trei:

a) Selectarea porturilor desemnate (designated) = porturi care nu „privesc” spre switch-ul rădăcină, dar prin care pot fi transmise pachete de date (a se vedea Figura 7)

b) Selectarea porturilor alternative (alternate) = porturile ce sunt blocate (a se vedea Figura 8)

O descriere mai detaliată:

1. Selectarea switch-ului rădăcină

Toate switch-urile construiesc arborele de legătură până la switch-ul rădăcină și formează și configurația inițială de rețea o topologie stea (care garantat nu include cicluri)

Obiectivul de bază al fiecărui switch este să construiască o rută până la switch-ul rădăcină

Selectarea switch-ului rădăcină este bazată pe comparația între unele valori, precum *prioritatea* (priority) și *adresa MAC a switch-ului* (a se vedea Figura 9)

Switch-urile ce mențin STP => au adrese MAC

Primul criteriu de selectare a switch-ului rădăcină – ca switch rădăcină este selectat acel switch care are cea mai mică valoare a *priorității*

Valoarea implicită a priorității este – 32768

Dacă prioritățile sunt egale => atunci se compară adresele MAC, octeții corespunzători de la stânga la dreapta => switch-ul ce are cea mai mică valoare a adresei MAC este numit ca switch rădăcină

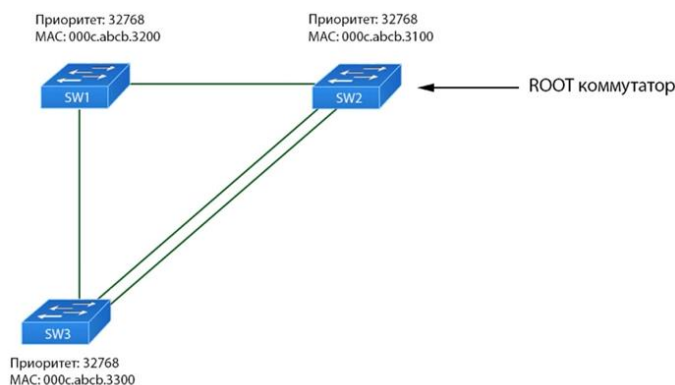


Figura 9

Valorile priorității și a adresei MAC se transmit împreună => acestea se conțin în câmpul *Bridge ID* (Figura 10)



Figura 10

Switch-ul cu valoarea cea mai mică pentru Bridge ID este switch-ul rădăcină

Dacă accesăm linia de comandă CLI a switch-ului și dăm comanda

*show spanning-tree* => vom vedea separat valorile pentru prioritate și pentru adresa MAC

Între switch-uri se transmit mesaje BPDU (Bridge Protocol Data Units)

În Figura 11 este arătată structura unui frame BPDU

Структура BPDU кадра											
2 байта	1 байт	1 байт	1 байт	8 байт	4 байта	8 байт	2 байта	2 байта	2 байта	2 байта	2 байта
Protocol ID	Version	Message Type	Flags	Root Bridge ID	Root Path Cost	Sender Bridge ID	Port ID	Message Age	Maximum Age	Hellow Time	Forward Delay

Figura 11

Într-un câmp pe 8 octeți, numit Root Bridge ID

Dacă în rețeaua prezentată prin Figura 5 trecem în modul Simulation, apăsăm butonul Show all/none, după care pe butonul Edit filters -> Misc și lăsăm bifate doar pachetele STP, vom vedea cum se propagă frame-urile STP (acolo se poate vedea că acest tip de pachet este de nivelul doi al stivei de protocoale TCP/IP) – frame-urile STP (în realitate BPDU) trec chiar și prin porturile blocate

La prima conectare fiecare switch crede că el este switch-ul rădăcină și transmite celorlalți vecini că este switch-ul rădăcină

De exemplu, switch-ul Sw3 a obținut două BPDU (de la Sw1 și Sw2) și compară valoarea proprie Bridge ID cu Bridge ID-urile lui Sw1 și Sw2. Dacă va vedea că valoarea Bridge ID a unui alt switch este mai mică (de exemplu, la Sw2) => Sw3 va considera de acum că Sw2 este switch-ul rădăcină

În continuare, la transmiterea BPDU către vecini, Sw3 va preciza nu valoarea sa Bridge ID, ci a noului switch rădăcină (Sw2) => Sw3 arată lui Sw1 că switch-ul rădăcină este Sw2

Procedura se repetă până atunci când toate switch-urile vor stabili același switch rădăcină

## II. Selectarea porturilor rădăcină

Se face în baza costului interfeței

Costul asociat switch-ului rădăcină este 0

Costul se transmite prin frame-ul BPDU în câmpul *Root Path Cost*

Accesăm linia de comandă CLI de pe Switch0 (a se vedea Figura 5):

În modul privilegiat scriem:

`Switch# show spanning-tree` (se poate scrie numai *span*)

Vom obține informație despre Root ID, Bridge ID, Interface (se vede rolul porturilor):

```
Switch>en
Switch#show span
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    32769
            Address    0006.2AB1.A3EA
            Cost        19
            Port        2(FastEthernet0/2)
            Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32769 (priority 32768 sys-id-ext 1)
            Address    00D0.97B3.6E78
            Hello Time  2 sec  Max Age 20 sec  Forward Delay 15 sec
            Aging Time  20

Interface Role Sts Cost Prio.Nbr Type
-----
Fa0/1     Altn BLK 19   128.1 P2p
Fa0/2     Root FWD 19   128.2 P2p
```

Cum putem impune un switch ca el să devină switch rădăcină (din careva considerente)?

Aceasta se poate face în modul următor (în linia de comandă CLI de pe Sw2, de exemplu):

```
Switch>en
Switch#conf ter
Switch(config)#spanning-tree vlan 1 root primary
```

Prioritatea lui Sw2 va deveni cu 8192 (iar la repetarea procedurii cu 4096) unități mai mică ca prioritatea lui Sw1 (care este switch-ul rădăcină) => are loc recalculul topologiei => aproximativ peste 30 de secunde legăturile se vor schimba

Pentru stabilirea manuală a priorității:

```
Switch(config)#spanning-tree vlan 1 priority 4096
```

PVST+ și Rapid PVST+ => versiuni actuale ale protocolului STP (dar reprezintă un soft proprietar Cisco)

Atunci când în rețea avem VLAN-uri => fizic există un ciclu, dar din punctul de vedere logic al VLAN-urilor – nu există ciclu

Uneori punctele ce corespund porturilor deconectate ale switch-ului, oricum rămân de culoare verde

Aceasta este deoarece pentru un VLAN se deconectează un port, iar pentru celălalt – uneori altul

Cu ajutorul comenzii `show spanning-tree` se poate vizualiza informația corespunzătoare fiecărui VLAN

Să examinăm configurația din Figura 12

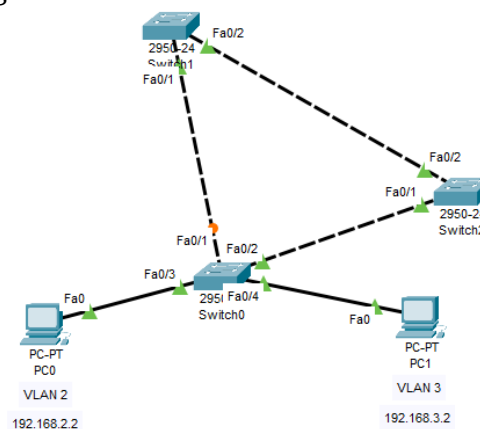


Figura 12

Protocolul PVST+ lucrează cu VLAN-urile și construiește un arbore separat în raport cu fiecare VLAN

Se poate de făcut astfel încât să avem un switch rădăcină pentru VLAN 2 și un alt switch rădăcină pentru

## VLAN3

Mai întâi creăm VLAN-urile 2 și 3 (conform schemei ilustrate anterior când am examinat tema dedicată VLAN-urilor)

```
Pe Switch0:
Switch>en
Switch#conf ter
Switch(config)#int fa 0/3
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 2
Switch(config-if)#exit
Switch(config)#int fa 0/4
Switch(config-if)#switchport mode access
Switch(config-if)#switchport access vlan 3
Switch(config-if)#exit
Switch(config)#int fa 0/2
Switch(config-if)#switchport mode trunk
Switch(config-if)#exit
Switch(config)#int fa 0/1
Switch(config-if)#switchport mode trunk
Switch(config-if)#exit
```

```
Pe Switch1:
Switch>en
Switch#conf ter
Switch(config)#int fa 0/2
Switch(config-if)#switchport mode trunk
Switch(config-if)#exit
Switch(config)#vlan 2
Switch(config-vlan)#exit
Switch(config)#vlan 3
Switch(config-vlan)#exit
```

```
Pe Switch2:
Switch>en
Switch#conf ter
Switch(config)#int range fa 0/1-2
Switch(config-if-range)#switchport mode trunk
Switch(config-if-range)#exit
Switch(config)#vlan 2
Switch(config-vlan)#exit
Switch(config)#vlan 3
Switch(config-vlan)#exit
```

## Accesăm linia de comandă CLI de pe Switch0:

```
Switch#show span
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID      Priority      24577
              Address      0006.2AB1.A3EA
              Cost        19
              Port        2(FastEthernet0/2)
              Hello Time  2 sec   Max Age 20 sec   Forward Delay 15 sec

  Bridge ID    Priority      32769 (priority 32768 sys-id-ext 1)
              Address      00D0.97B3.6E78
              Hello Time  2 sec   Max Age 20 sec   Forward Delay 15 sec
              Aging Time  20

Interface      Role Sts Cost      Prio.Nbr Type
-----
Fa0/1          Altn BLK 19        128.1    P2p
Fa0/2          Root FWD 19        128.2    P2p

VLAN0002
  Spanning tree enabled protocol ieee
  Root ID      Priority      32770
```

	Address	0006.2AB1.A3EA			
	Cost	19			
	Port	2(FastEthernet0/2)			
	Hello Time	2 sec	Max Age 20 sec	Forward Delay 15 sec	
Bridge ID	Priority	32770	(priority 32768 sys-id-ext 2)		
	Address	00D0.97B3.6E78			
	Hello Time	2 sec	Max Age 20 sec	Forward Delay 15 sec	
	Aging Time	20			
Interface	Role	Sts	Cost	Prio.Nbr	Type
-----	----	---	-----	-----	-----
Fa0/1	Altn	BLK	19	128.1	P2p
Fa0/2	Root	FWD	19	128.2	P2p
Fa0/3	Desg	FWD	19	128.3	P2p
VLAN0003					
	Spanning tree enabled protocol ieee				
Root ID	Priority	32771			
	Address	0006.2AB1.A3EA			
	Cost	19			
	Port	2(FastEthernet0/2)			
	Hello Time	2 sec	Max Age 20 sec	Forward Delay 15 sec	
Bridge ID	Priority	32771	(priority 32768 sys-id-ext 3)		
	Address	00D0.97B3.6E78			
	Hello Time	2 sec	Max Age 20 sec	Forward Delay 15 sec	
	Aging Time	20			
Interface	Role	Sts	Cost	Prio.Nbr	Type
-----	----	---	-----	-----	-----
Fa0/1	Altn	BLK	19	128.1	P2p
Fa0/2	Root	FWD	19	128.2	P2p
Fa0/4	Desg	FWD	19	128.4	P2p

Am obținut informația pentru VLAN1, VLAN2 și VLAN3; vedem care port-uri sunt blocate în raport cu fiecare VLAN

Accesăm linia de comandă CLI pe Switch2 și dăm comanda *show span*:

Switch#show span
VLAN0001
Spanning tree enabled protocol ieee
Root ID Priority 24577
Address 0006.2AB1.A3EA
This bridge is the root
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Bridge ID Priority 24577 (priority 24576 sys-id-ext 1)
Address 0006.2AB1.A3EA
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Aging Time 20
Interface Role Sts Cost Prio.Nbr Type
-----
Fa0/1 Desg FWD 19 128.1 P2p
Fa0/2 Desg FWD 19 128.2 P2p
VLAN0002
Spanning tree enabled protocol ieee
Root ID Priority 32770
Address 0006.2AB1.A3EA
This bridge is the root
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Bridge ID Priority 32770 (priority 32768 sys-id-ext 2)
Address 0006.2AB1.A3EA
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Aging Time 20

Interface	Role	Sts	Cost	Prio.	Nbr	Type
-----------	------	-----	------	-------	-----	------

Fa0/1	Desg	FWD	19	128.1	P2p	
Fa0/2	Desg	FWD	19	128.2	P2p	

VLAN0003

Spanning tree enabled protocol ieee

Root ID Priority 32771

Address 0006.2AB1.A3EA

This bridge is the root

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 32771 (priority 32768 sys-id-ext 3)

Address 0006.2AB1.A3EA

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Aging Time 20

Interface	Role	Sts	Cost	Prio.	Nbr	Type
-----------	------	-----	------	-------	-----	------

Fa0/1	Desg	FWD	19	128.1	P2p	
Fa0/2	Desg	FWD	19	128.2	P2p	

Vedem că Switch2 reprezintă switch-ul rădăcină pentru VLAN 2 și VLAN 3

Vom face ca switch rădăcină pentru VLAN-ul 3 să devină switch-ul Switch1:

Accesăm Switch1 și în linia de comandă executăm:

```
Switch#conf ter
```

```
Switch(config)#spanning-tree vlan 3 root primary
```

Dacă vom intra în linia de comandă a lui Switch0 și vom scrie comanda *show span*, atunci vom vedea că porturile au roluri diferite în raport cu VLAN 2 și VLAN 3:

VLAN0002

Spanning tree enabled protocol ieee

Root ID Priority 32770

Address 0006.2AB1.A3EA

Cost 19

Port 2(FastEthernet0/2)

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 32770 (priority 32768 sys-id-ext 2)

Address 00D0.97B3.6E78

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Aging Time 20

Interface	Role	Sts	Cost	Prio.	Nbr	Type
-----------	------	-----	------	-------	-----	------

Fa0/1	Altn	BLK	19	128.1	P2p	
Fa0/2	Root	FWD	19	128.2	P2p	
Fa0/3	Desg	FWD	19	128.3	P2p	

VLAN0003

Spanning tree enabled protocol ieee

Root ID Priority 24579

Address 00D0.9772.8013

Cost 19

Port 1(FastEthernet0/1)

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 32771 (priority 32768 sys-id-ext 3)

Address 00D0.97B3.6E78

Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Aging Time 20

Interface	Role	Sts	Cost	Prio.	Nbr	Type
-----------	------	-----	------	-------	-----	------



Fa0/1	Root	FWD	19	128.1	P2p
Fa0/2	Altn	BLK	19	128.2	P2p
Fa0/4	Desg	FWD	19	128.4	P2p

Pentru VLAN 3 Switch1 a devenit switch rădăcină; pentru VLAN 2 Switch2 a devenit switch rădăcină  
În continuare vom arăta cum se face ieșirea în Internet prin router (a se vedea Figura 13)

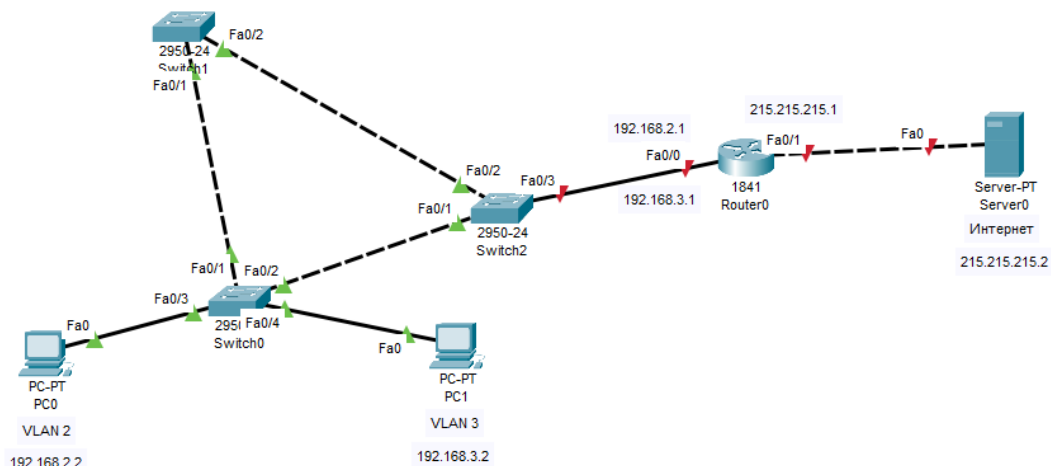


Figura 13

Pentru aceasta vom executa următoarele:

1. Pe host-urile PC0 și PC1 configurăm adresele IP ale routerului implicit 192.168.2.1 și, corespunzător 192.168.3.1
2. Facem portul Fa 0/3 al switch-ului Switch2 să fie de tip trunk
3. Configurăm două subinterfețe (Fa 0/0.2 și Fa 0/0.3) pe portul Fa 0/0 al routerului

```
Router>en
Router#conf ter
Router(config)#int fa 0/0.2
Router(config-subif)#encapsulation dot1Q 2
Router(config-subif)#ip add 192.168.2.1 255.255.255.0
Router(config-subif)#exit
Router(config)#int fa 0/0.3
Router(config-subif)#encapsulation dot1Q 3
Router(config-subif)#ip add 192.168.3.1 255.255.255.0
Router(config-subif)#exit
Router(config)#int fa 0/0
Router(config-if)#no shut    % включаем физический интерфейс
```

4. Atribuim IP-ul 215.215.215.1 pe interfața Fa 0/1

```
Router(config)#int fa 0/1
Router(config-if)#ip add 215.215.215.1 255.255.255.0
Router(config-if)#no shut
```

5. Pe server configurăm IP-ul 215.215.215.2 și adresa routerului implicit 215.215.215.1

Dacă dăm un ping către serverul de Internet 215.215.215.2 de pe host-urile PC0 și PC1 => vedem că este conexiune

În modul Simulation vom urmări traseul pachetelor de date (lăsăm să fie vizualizate doar pachetele ICMP):

Pe host-ul PC0:

În VLAN2 și în VLAN3 avem trasee diferite pe care se deplasează pachetele

Pe host-ul PC1:

În VLAN2 și în VLAN3 avem trasee diferite pe care se deplasează pachetele

### **Tehnologia Etherchannel**

EtherChannel este o tehnologie de agregare a legăturilor dezvoltată de Cisco Systems. Tehnologia vă permite să combinați mai multe canale Ethernet fizice într-unul logic pentru a crește lățimea de bandă și a îmbunătăți fiabilitatea conexiunii (stabilitate la defectiuni).

EtherChannel oferă posibilitatea de a combina de la două la opt porturi Ethernet de 100 Mbps, 1 Gbps

sau 10 Gbps (toate porturile din canal trebuie să aibă aceeași viteză, aceleași setări duplex și aceleași setări VLAN), ce funcționează pe bază de cablu cu perechi răsucite sau fibră optică, ceea ce permite să obținem o viteză de până până la 800 Mbps, 8 Gbps sau 80 Gbps. În plus, de la unul la șapte porturi pot fi inactive și pot intra în funcțiune atunci când conexiunea este întreruptă pe unul dintre porturile active. Dacă nu există porturi rezervă, traficul este distribuit automat prin toate conexiunile active.

Canalul logic se poate stabili între routere, switch-uri și adaptoare de rețea pe server. Toate adaptoarele de rețea care fac parte dintr-un link (canal) primesc o singură adresă MAC, făcând link-ul transparent pentru aplicațiile de rețea. Balansarea traficului între porturi se face în baza unei funcții hash care se aplică la adresa MAC sau la adresa IP sau la numerele de port TCP sau UDP ale sursei și destinatarului. În cazul cel mai nefavorabil se poate întâmpla că tot traficul poate fi transportat printr-o singură legătură fizică.

Atunci când utilizați STP împreună cu EtherChannel, toate conexiunile fizice sunt considerate ca o conexiune logică și pachetele BDPD sunt trimise fizic doar prin una din acele conexiuni fizice. Un algoritm special ce stă la baza protocolului EtherChannel permite să se identifice posibilele neconcordanțe, atunci când unul dintre switch-uri nu este configurat pentru a gestiona corect link-ul (canalul).

La configurarea EtherChannel, porturile de pe ambele părți ale legăturii sunt adăugate în canalul logic sau manual sau se folosește unul dintre protocolele de agregare automatizată a porturilor: protocolul proprietar Cisco PAgP sau cel descris în standardul IEEE 802.3ad LACP.

EtherChannel este o tehnologie de agregare a legăturilor dintre dispozitive, dezvoltată de Cisco Systems. Tehnologia permite combinarea mai multor canale Ethernet fizice într-unul logic pentru a crește lățimea de bandă și, cel mai important, pentru a crește fiabilitatea conexiunii. Adică, EtherChannel este un mecanism de asigurare a fiabilității rețelei la nivel de transfer de date.



Figura 23

Deoarece avem 2 cabluri între switch-uri, un port va fi blocat de protocolul Spanning Tree, astfel încât să nu se formeze un ciclu (bucă). Și dacă un link iese din funcțiune, atunci se activează celălalt.

Problemă: la un moment dat de timp se transferă date doar printr-un cablu. S-au procurat 2 interfețe, dar, de fapt, se folosește doar una.

O interfață de router Cisco costă mai mult de 8000 de lei, iar un port de switch costă mai puțin.

Pentru soluționarea problemei, a fost inventat mecanismul Etherchannel, prin care dispozitivele sunt conectate prin două cabluri, care funcționează ca unul singur, adică două interfețe fizice sunt combinate într-una logică.

=> creșterea performanței (100 megabiți + 100 megabiți = 200 megabiți)

=> fără buclă

=> este asigurată fiabilitatea; dacă un cablu iese din funcțiune => datele vor merge prin al doilea cablu

### **I. EtherChannel de nivel 2**

În Cisco Packet Tracer vom configura un EtherChannel de nivel 2:

pentru aceasta vom considera 2 switch-uri pe care le vom conecta prin două cabluri:

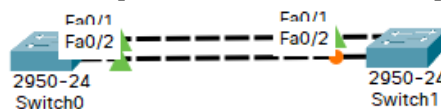


Figura 24

Definim următoarele comenzi:

```
Pe switch-ul Switch0:  
Switch(config)#int range fa 0/1, fa 0/2  
Switch(config-if-range)#channel-group 1 mode active  
% A fost creată interfața port-canal Port-channel 1
```

Pe switch-ul Switch1:  
Switch(config)#int range fa 0/1, fa 0/2  
Switch(config-if-range)#channel-group 1 mode active  
% A fost creată interfața port-canal Port-channel 1

Aceleași comenzi au fost setate pe ambele switch-uri => punctul galben a dispărut => protocolul Spanning Tree nu este necesar

Două porturi fizice sunt conectate într-un singur port logic, numit port-channel (port-canal).

În linia de comandă a dispozitivului puteți utiliza comanda end pentru a reveni la modul privilegiat #.

Pe Switch0 definim comanda

show etherchannel summary

```
Switch#show eth summary
Flags:  D - down          P - in port-channel
        I - stand-alone  s - suspended
        H - Hot-standby (LACP only)
        R - Layer3       S - Layer2
        U - in use       f - failed to allocate aggregator
        u - unsuitable for bundling
        w - waiting to be aggregated
        d - default port

Number of channel-groups in use: 1
Number of aggregators:          1

Group  Port-channel  Protocol    Ports
-----+-----+-----+-----
1      Pol(SU)        LACP        Fa0/1(P) Fa0/2(P)
```

Figura 25

În tabelul rezultat (vezi Figura 25) putem vedea:

Interfețele fizice care participă în Etherchannel: Fa 0/1 (P), Fa 0/2 (P)

(P înseamnă port-channel)

Acele două interfețe fizice au fuzionat într-o singură interfață logică Po1(SU)

(SU înseamnă (Layer 2 in use) că interfața funcționează la al doilea nivel al stivei TCP/IP, adică avem Etherchannel care lucrează la nivel de comutare)

Protocolul utilizat este LACP - un protocol pentru verificarea corectitudinii asamblării Etherchannel

Dacă Etherchannel a fost asamblat incorect => protocolul LACP va verifica și nu va permite să se lucreze mai departe cu Etherchannel => deoarece dacă Etherchannel nu este asamblat corect => se poate forma o buclă

Există, de asemenea, protocolul PAGP și, la fel, se poate verifica manual procedura de asamblare a Etherchannel (fără a utiliza vreun protocol). Protocolul nu permite formarea unei bucle la asamblarea Etherchannel.

Protocole de verificare a configurației Etherchannel:

LACP este un protocol deschis dezvoltat de IEEE (multivendor)

PAGP este un protocol proprietar Cisco (rulează numai pe echipamente Cisco)

Configurare manuală - fără protocol

Când am asamblat un Etherchannel în Cisco Packet Tracer, am scris comanda:

channel-group 1 mode active

adică rula protocolul LACP. Dacă am fi scris

channel-group 1 mode auto

atunci ar fi rulat protocolul PAGP.

Dacă am fi scris

channel-group 1 mode on

atunci ar fi trebuit să configurăm manual.

LACP	Active	Passive
Active	Da	Da
Passive	Da	Nu

Când se aplică protocolul LACP, pe un switch scriem mode active, iar pe celălalt putem scrie mode passive. Este interzis să se scrie la ambele switch-uri mode passive.

Vom lucra cu LACP, deoarece este acceptat și de alți furnizori.

Pentru protocolul PAgP, avem următorul tabel:

PAgP	Desirable	Auto
Desirable	Da	Da
Auto	Da	Nu

În cazul configurării manuale avem:

Configurare manuală	On
On	Da

Cerințe pentru construirea Etherchannel:

- Același tip de port (access sau trunk);
- Aceeași viteză și duplex;
- Aceleași VLAN-uri native și allowed pe trunk-uri;
- Același VLAN pe porturile de tip access.

Există, de asemenea, un tip de Etherchannel, care funcționează la nivelul 3 al stivei TCP / IP => avem un Etherchannel care rulează la nivel de rutare pe switch-urile de nivelul 3.

În modelul ierarhic de rețea pe 3 niveluri propus de Cisco, avem asigurat un nivel suplimentar de fiabilitate: fiecare două fire paralele sunt combinate într-un Etherchannel => în Figura 26, în loc de 6 cabluri (linii de comunicație) obținem 3 (logice).

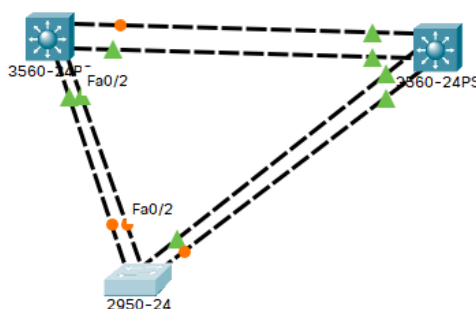


Figura 26

Combinăm două legături care conectează switch-ul de nivel doi cu switch-ul de nivel trei din stânga, într-un singur port channel 1:

```
Switch(config)#hostname SW0
SW0(config)#int range fa 0/1, fa 0/2
SW0(config-if-range)#channel-group 1 mode active

Switch(config)#hostname SW1
SW1(config)#int range fa 0/1, fa 0/2
SW1(config-if-range)#channel-group 1 mode active
```

În mod similar, combinăm celelalte două perechi de legături, respectiv, în port-channel 2 și 3.

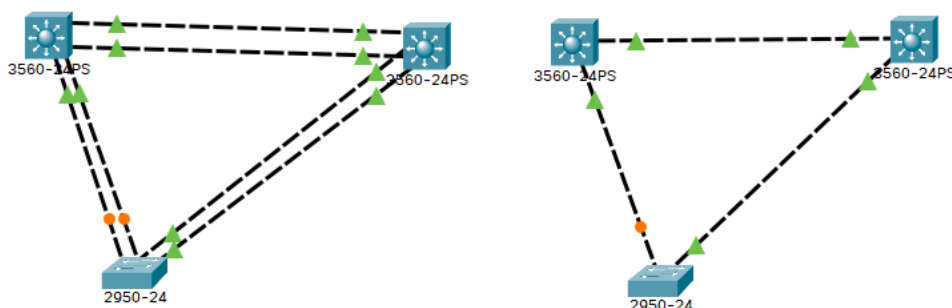


Figura 27

Rezultatul este configurația prezentată în Figura 27 (stânga), similară cu configurația din Figura 27 din dreapta. Deoarece avem o buclă => protocolul Spanning Tree blochează Port-channel 1 => blochează nu o interfață fizică, ci o interfață logică. Acest lucru poate fi vizualizat, utilizând comanda show span pe switch-ul L2 (vezi Figura 28):

```
Switch>en
Switch#show span
VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    32769
            Address     0003.E461.3192
            Cost        9
            Port        26(Port-channel2)
            Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32769 (priority 32768 sys-id-ext 1)
            Address     000D.BDD5.59D1
            Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
            Aging Time   20
```

Interface	Role	Sts	Cost	Prio.Nbr	Type
Po2	Root	FWD	9	128.26	Shr
Po1	Altn	BLK	9	128.25	Shr

Figura 28

Să configurăm interfața logică Po1 (SU) ca un trunk, ceea ce va face ca interfețele fizice corespunzătoare Fa 0/1 (P) și Fa 0/2 (P) la fel să devină de tip trunk:

```
Switch(config)#int port-channel 1
Switch(config-if)#switchport mode trunk
```

Am trecut port-channel 1 în modul trunk. Pe SW0 definim comanda

```
show int fa 0/1 sw
```

Obținem:

```
SW0#show int fa 0/1 sw

Name: Fa0/1
Switchport: Enabled
Administrative Mode: dynamic auto
Operational Mode: static access
Administrative Trunking Encapsulation: negotiated
Operational Trunking Encapsulation: native
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Voice VLAN: none
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk private VLANs: none
Operational private-vlan: none
Trunking VLANs Enabled: All
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
Capture VLANs Allowed: ALL
Protected: false
Unknown unicast blocked: disabled
Unknown multicast blocked: disabled
Appliance trust: none
```

Se vede că acest port a fost trecut în modul trunk, adică a fost configurat dinamic.

Este necesar de transformat în trunk și cealaltă pereche orientată spre switch-ul L3 din dreapta. Dar în setul de comenzi pentru perechea de switch-uri L3 vom adăuga comanda

```
switchport trunk encapsulation dot1q
```

```
SW0(config-if)#int port-channel 3
SW0(config-if)#sw tr enc dot1q
SW0(config-if)#sw mo tr
```

Pe switch-ul L2 se va face de tip trunk și perechea orientată spre switch-ul L3 din dreapta:

```
Switch(config)#int port-channel 2
Switch(config-if)#switchport mode trunk
```

Două interfețe trebuie combinate într-un singur Etherchannel atunci când acestea nu sunt încă configurate (ca să nu apară dificultăți). După care accesăm port-channel și setăm parametrii necesari. Deoarece în port-channel-ul examinat există 2 cabluri fizice => prin fiecare cablu poate trece o parte a traficului. Puteți combina până la 16 cabluri (și interfețe) într-un singur port-channel, dar de fapt, 8 porturi vor funcționa, iar restul 8 se vor afla în modul de așteptare. Traficul va fi balansat prin 8 porturi, iar celelalte 8 porturi se vor afla în stare de repaos și vor aștepta până atunci când o linie de comunicație se va bloca => iată atunci aceste porturi se vor activa.

Cum se face balansarea traficului (și se amplifică randamentul)?

```
Switch# show etherchannel load-balance
Результат:
EtherChannel Load-Balancing Operational State (src-mac):
Non-IP: Source MAC address
IPv4: Source MAC address
IPv6: Source MAC address
```

Aceasta înseamnă că balansarea va fi efectuată în baza adresei MAC a sursei.

Calculatorul formează un frame, care este transmis switch-ului. Mecanismul Etherchannel calculează valoarea hash de la adresa MAC a sursei, ia ultimii câțiva biți din valoarea hash și dacă (de exemplu) la sfârșitul acestei valori hash se obține bitul 0, atunci frame-ul este trimis prin prima interfață, iar dacă ultimul bit este 1 - prin a doua interfață.

Algoritmul de balansare aplicat în cadrul Etherchannel poate fi modificat:

```
Switch(config)#port-channel load-balance ?
dst-ip Dst IP Addr
dst-mac Dst Mac Addr
src-dst-ip Src XOR Dst IP Addr
src-dst-mac Src XOR Dst Mac Addr
src-ip Src IP Addr
src-mac Src Mac Addr
```

Selectăm o comandă, care va fi aplicată tuturor port-channel-urilor (1, 2 și 3).

*Metode de balansare în Etherchannel:*

Metoda de balansare	De la ce se va calcula valoarea hash?	Dispozitivul
Dst-ip	Ip adresa destinației	Toate switch-urile
Dst-mac	Mac adresa destinației	Toate switch-urile
Src-dst-ip	Ip adresele sursei și a destinației	Toate switch-urile
Src-dst-mac	Mac adresele sursei și a destinației	Toate switch-urile
Src-ip	Ip adresa sursei	Toate switch-urile
Src-mac	Mac adresa sursei	Toate switch-urile
Src-port	Numărul de port al sursei	Switch-urile din seria Cisco Catalyst 4500 и 6500
Dst-port	Numărul de port al destinației	Switch-urile din seria Cisco Catalyst 4500 и 6500
Src-dst-port	Numerele de port ale sursei și destinației	Switch-urile din seria Cisco Catalyst 4500 и 6500

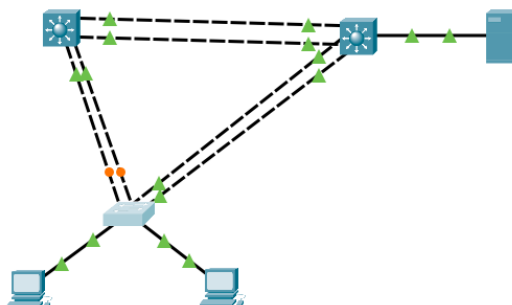


Figura 29

```
Switch#show etherchannel load-balance
EtherChannel Load-Balancing Operational State (src-mac):
Non-IP: Source MAC address
```

IPv4: Source MAC address  
IPv6: Source MAC address

Serverul va trimite pachete cu aceeași adresă Mac a sursei (adresa sa) => traficul va fi trimis prin același cablu => nu se va realiza o balansare a traficului

Performanța va fi mai mare dacă va fi activat src-mac, atunci când există mult trafic către diferite servere, de la utilizatori diferiți.

Algoritmul care pare a fi cel mai rațional este cel în care se activează src-dst-ip => se vor obține valori hash diferite (deoarece sau src, sau dst vor fi diferite) => canale de transmisie diferite => există balansare a traficului

Un alt algoritm ce prezintă interes este src-dst-port (dacă este menținut de către dispozitivul de rețea) => porturile sursă se vor modifica atunci când se utilizează protocolul TCP sau UDP.

## II. Etherchannel de nivelul 3 pe switch-uri de nivelul 3

În configurația din Figura 29 vom elimina serverul și vom adăuga încă două switch-uri L3, pe care le conectăm așa cum se arată în Figura 30:

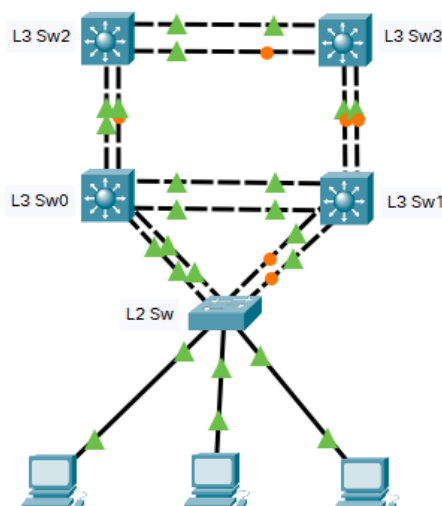


Figura 30

Între L3 Sw0 și L3 Sw2, L3 Sw1 și L3 Sw3, L3 Sw2 și L3 Sw3 se vor configura Etherchannel-uri de nivelul 3.

Avem două cabluri între L3 Sw0 și L3 Sw2, dar trebuie să facem ca la nivel logic să fie o linie de comunicație.

Configurarea L3 Etherchannel pe switch-ul L3 Sw2:

```
Switch(config)#int port-channel 4
Switch(config-if)#no switchport
Switch(config-if)#ip address 10.0.1.2
255.255.255.0
Switch(config-if)#exit
Switch(config)#int range fa 0/1, fa 0/2
Switch(config-if-range)#no switchport
Switch(config-if-range)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface
FastEthernet0/3, changed state to down
%LINEPROTO-5-UPDOWN: Line protocol on Interface
FastEthernet0/3, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface
FastEthernet0/4, changed state to down
%LINEPROTO-5-UPDOWN: Line protocol on Interface
FastEthernet0/4, changed state to up
Switch(config-if-range)#channel-group 4 mode
active
Switch(config-if-range)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface
FastEthernet0/3, changed state to down
%LINEPROTO-5-UPDOWN: Line protocol on Interface
FastEthernet0/3, changed state to up
```

În mod manual creăm interfața port-channel 4  
Trecem interfața în regim de router  
Definim o adresă IP

Trecem la configurarea grupului de porturi

Activăm Etherchannel cu protocolul LACP de  
verificare a corectitudinii de configurare



```
%LINEPROTO-5-UPDOWN: Line protocol on Interface
FastEthernet0/4, changed state to down
%LINEPROTO-5-UPDOWN: Line protocol on Interface
FastEthernet0/4, changed state to up
```

Vom proceda la fel și pe L3 Sw0, definind acolo adresa IP 10.0.1.1, masca 255.255.255.0 și range fa 0/1, fa 0/2. Pe L3 Sw0:

```
Switch(config)#do show etherchannel summary
```

```
Flags: D - down      P - in port-channel
       I - stand-alone s - suspended
       H - Hot-standby (LACP only)
       R - Layer3     S - Layer2
       U - in use     f - failed to allocate aggregator
       u - unsuitable for bundling
       w - waiting to be aggregated
       d - default port
```

```
Number of channel-groups in use: 1
```

```
Number of aggregators:      1
```

Group	Port-channel	Protocol	Ports
4	Po4(RU)	LACP	Fa0/3(P) Fa0/4(P)

În mod analog se configurează Etherchannel pe

L3 Sw3 și L3 Sw1,

L3 Sw2 și L3 Sw3.

Obținem că pe legăturile dintre L3 Sw0 și L3 Sw2, L3 Sw1 și L3 Sw3, L3 Sw2 și L3 Sw3 au apărut triunghiuri de culoare verde, iar pe legăturile dintre L3 Sw0 și L3 Sw1, L2 Sw și L3 Sw0, L2 Sw și L3 Sw1 avem o buclă și, prin urmare, porturile lui L2 Sw orientate spre L3 Sw0 sunt dezactivate (a rulat protocolul STP).

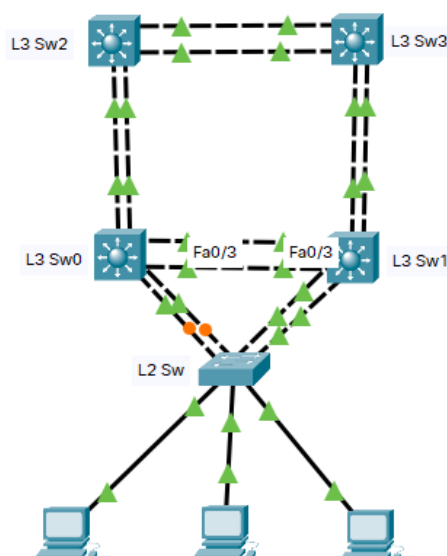


Figura 31

### III. Etherchannel de nivel 3 pe routere

Cu părere de rău, pe Cisco Packet Tracer nu a fost implementată tehnologia Etherchannel la nivel de routere. În GNS3 este parțial implementată. Pe routere nu există comenzi pentru rularea protocoalelor LACP sau PAgP, de aceea, pe acestea Etherchannel este configurat numai manual.

Pe echipamente reale, pentru a configura Etherchannel pe routere, pe interfețele conectate direct între ele vom da următoarele comenzi:

```
int port-channel 8
no shut
ip address 192.168.1.1 255.255.255.0
```



```

exit
int range fa 0/1, fa 0/0
no shut
channel-group 8
exit

```

Particularitățile configurării L3 Etherchannel pe routere:

- Este acceptată doar agregarea statică, fără utilizarea protocoalelor
- Pot fi create doar două interfețe agregate
- Numărul maxim de interfețe într-un Etherchannel este 4 (pe switch-uri - 16)
- Algoritmul de balansare utilizează adresele IP ale sursei și destinației (src-dst-ip). Acest algoritm este activat ca implicit și nu poate fi modificat
- Doar acele interfețe pot fi agregate care se află pe module de același tip

Etherchannel-ul clasic a fost creat pentru agregarea legăturilor, astfel încât două dispozitive de rețea să fie conectate prin două cabluri. Apoi au apărut mecanisme care i-au permis lui Etherchannel să elimine buclele, astfel nemaifiind necesar protocolul lent Spanning Tree. O astfel de tehnologie este VSL (Virtual Switch Link).

Din păcate, la această etapă nu vom putea arăta cum se stabilește conexiunea cu serverul de Internet prezentat în Figura 32, deoarece pentru aceasta trebuie să se configureze pe switch-urile L3 și pe router protocoale de rutare a pachetelor, iar aceste chestiuni le vom examina ulterior, în cadrul lucrării de laborator N5.

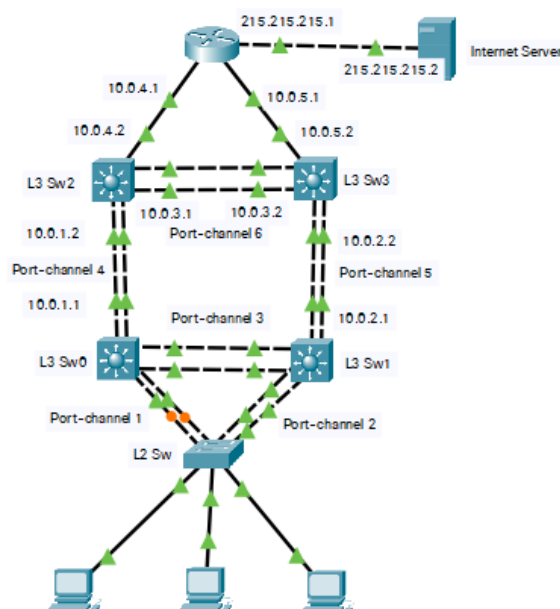


Figura 32

## Cerințe pentru realizarea lucrării de laborator N°3

În Cisco Packet Tracer, efectuați următoarele:

1. Construiți topologia logică de rețea prezentată în Figura 33. Folosind datele din Tabelul 1, configurați dispozitivele rețelei. Construiți și configurați cele trei VLAN-uri (cu ID-urile k+1, k+2 și k+3) indicate în Figura 33 (această etapă a fost efectuată și în lucrarea de laborator N2).

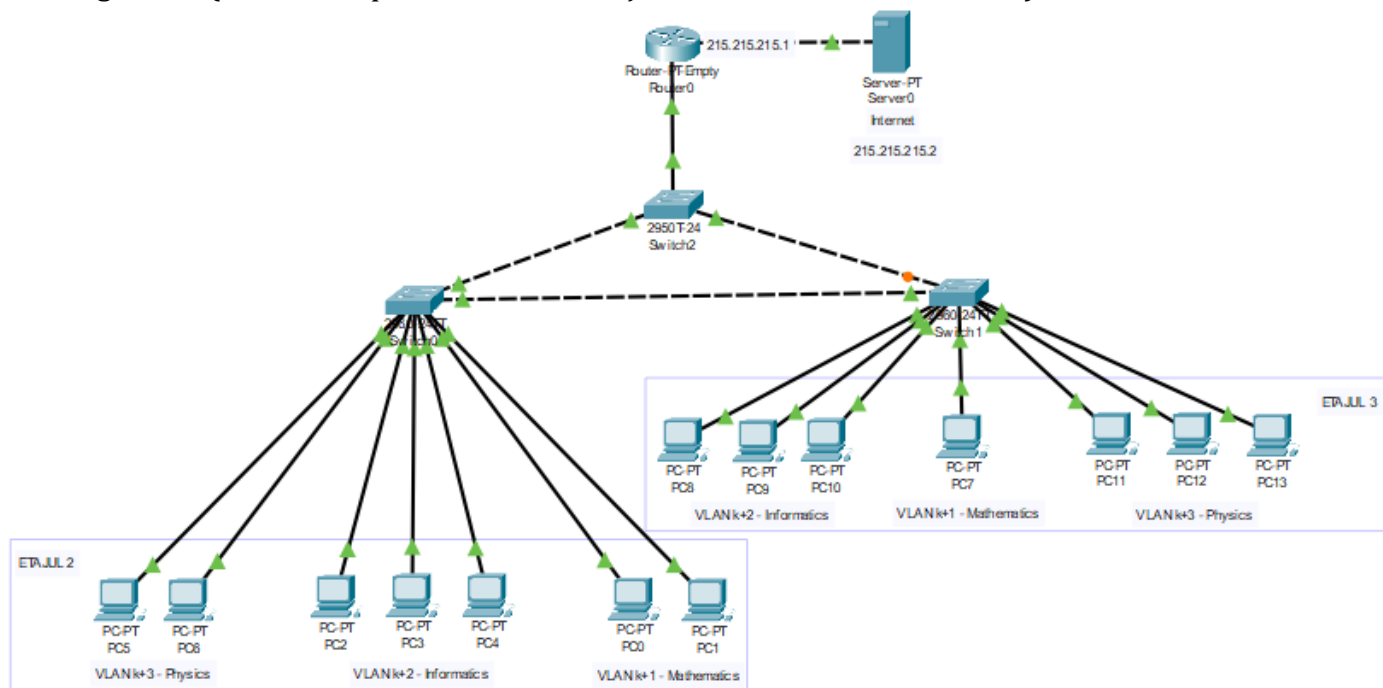


Figura 33

2. Pentru comutatoarele Switch0, Switch1 și Switch2 arătați și comentați cu privire la fiecare dintre VLAN-urile k+1, k+2 și k+3 informația ce se referă la switch-ul rădăcină (root), Bridge ID, tipurile de porturi (root=rădăcină, designated=desemnat, alternate=alternativ). Explicați de ce a fost selectat același switch în calitate de root pentru fiecare dintre VLAN-urile k+1, k+2 și k+3. Arătați cum se deplasează pachetul ICMP către serverul de Internet și înapoi, acesta fiind transmis de la host-ul a) PC6; b) PC10.

Salvați configurația creată cu numele **Nume\_Prenume\_Grupa\_Retea3a.pkt**.

3. Realizați operațiile necesare pentru ca Switch0 să fie root pentru host-urile din VLAN-urile k+1 și k+3, iar Switch1 să fie root pentru host-urile din VLAN-ul k+2. Pentru fiecare switch (Switch0, Switch1, Switch2) arătați informația ce se referă la switch-ul rădăcină (root), Root ID, Bridge ID, tipurile de porturi (root, designated, alternate). Arătați cum se deplasează pachetul ICMP către serverul de Internet și înapoi, acesta fiind transmis de la host-ul a) PC6; b) PC10 c) PC7.

Salvați configurația creată cu numele **Nume\_Prenume\_Grupa\_Retea3b.pkt**.

**Tabelul 1**

Subrețea VLAN	Nume/Tip dispozitiv	Port de switch sau router	IP adresa	Masca subrețelei	Default Gateway
VLAN k+1 192.168.k+1.0 255.255.255.0	PC0	FastEthernet0	192.168.k+1.k+1	255.255.255.0	192.168.k+1.1
	PC1	FastEthernet0	192.168.k+1.k+2	255.255.255.0	192.168.k+1.1
	PC7	FastEthernet0	192.168.k+1.k+3	255.255.255.0	192.168.k+1.1
VLAN k+2 192.168.k+2.0 255.255.255.0	PC2	FastEthernet0	192.168.k+2.k+1	255.255.255.0	192.168.k+2.1
	PC3	FastEthernet0	192.168.k+2.k+2	255.255.255.0	192.168.k+2.1
	PC4	FastEthernet0	192.168.k+2.k+3	255.255.255.0	192.168.k+2.1
	PC8	FastEthernet0	192.168.k+2.k+4	255.255.255.0	192.168.k+2.1
	PC9	FastEthernet0	192.168.k+2.k+5	255.255.255.0	192.168.k+2.1
	PC10	FastEthernet0	192.168.k+2.k+6	255.255.255.0	192.168.k+2.1
	PC5	FastEthernet0	192.168.k+3.k+1	255.255.255.0	192.168.k+3.1

VLAN k+3 192.168.k+3.0 255.255.255.0	PC6	FastEthernet0	192.168.k+3.k+2	255.255.255.0	192.168.k+3.1
	PC11	FastEthernet0	192.168.k+3.k+3	255.255.255.0	192.168.k+3.1
	PC12	FastEthernet0	192.168.k+3.k+4	255.255.255.0	192.168.k+3.1
	PC13	FastEthernet0	192.168.k+3.k+5	255.255.255.0	192.168.k+3.1
	Router PT-Empty	Subinterfețele primei interfețe:			
		FastEthernet	192.168.k+1.1	255.255.255.0	-
		FastEthernet	192.168.k+2.1	255.255.255.0	-
		FastEthernet	192.168.k+3.1	255.255.255.0	-
		A doua interfață:			
		FastEthernet	215.215.215.1	255.255.255.0	-
	Server-PT	FastEthernet	215.215.215.2	255.255.255.0	215.215.215.1
	Multilayer Switch 1	VLAN k+1	192.168.k+1.1	255.255.255.0	-
		VLAN k+2	192.168.k+2.1	255.255.255.0	-
		VLAN k+3	192.168.k+3.1	255.255.255.0	-

(*k* = numărul de ordine alfabetică a studentului în registrul grupei)

4. Construiți topologia logică de rețea prezentată în Figura 34. Configurați Etherchannel de nivelul 2 pe switch-urile din rețea, în acele locuri unde avem indicii port-channel 1, ..., port-channel 11. Pentru a configura Etherchannel, utilizați protocolul PAGP.

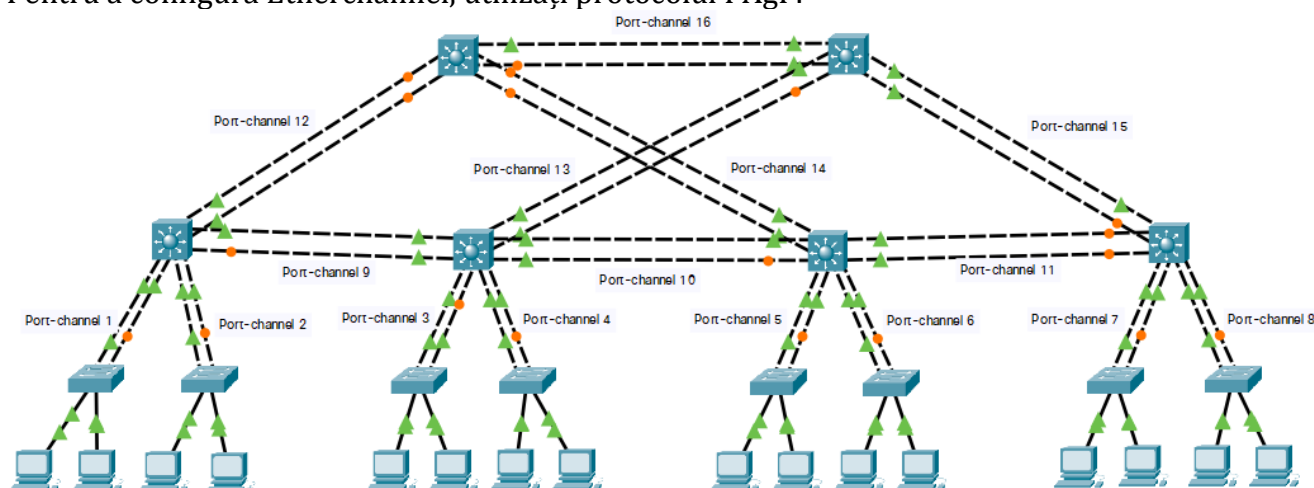


Figura 34

5. Configurați Etherchannel de nivelul 3 pe switch-urile din rețea, în acele locuri unde avem indicii port-channel 12, ..., port-channel 16. Pentru a configura Etherchannel, utilizați protocolul PAGP. Adresele IP necesare la configurare le găsiți în tabelul din Figura 35:

k- numărul de ordine al studentului	Adresa de rețea	Adresele interfețelor porturilor	Masca
port-channel 12	192.168.k+1.0	192.168.k+1.1	255.255.255.0
		192.168.k+1.2	255.255.255.0
port-channel 13	192.168.k+2.0	192.168.k+2.1	255.255.255.0
		192.168.k+2.2	255.255.255.0
port-channel 14	192.168.k+3.0	192.168.k+3.1	255.255.255.0
		192.168.k+3.2	255.255.255.0
port-channel 15	192.168.k+4.0	192.168.k+4.1	255.255.255.0
		192.168.k+4.2	255.255.255.0
port-channel 16	192.168.k+5.0	192.168.k+5.1	255.255.255.0
		192.168.k+5.2	255.255.255.0

Figura 35

Salvați configurația creată în fișierul **Nume\_Prenume\_Grupa\_Retea3c.pkt**.

Realizați o dare de seamă asupra lucrului efectuat, care să conțină răspunsuri explicite la fiecare punct formulat în cerințe.

Trimiteți darea de seamă împreună cu fișierele .pkt la adresa de e-mail [capceleam@gmail.com](mailto:capceleam@gmail.com) și încărcați fișierul cu darea de seamă și fișierele .pkt în mapa *Lucrarea de laborator N3* din pagina dedicată cursului de Rețele de Calculatoare a platformei educaționale moodle.usm.md.