~ Laborator 5 ~

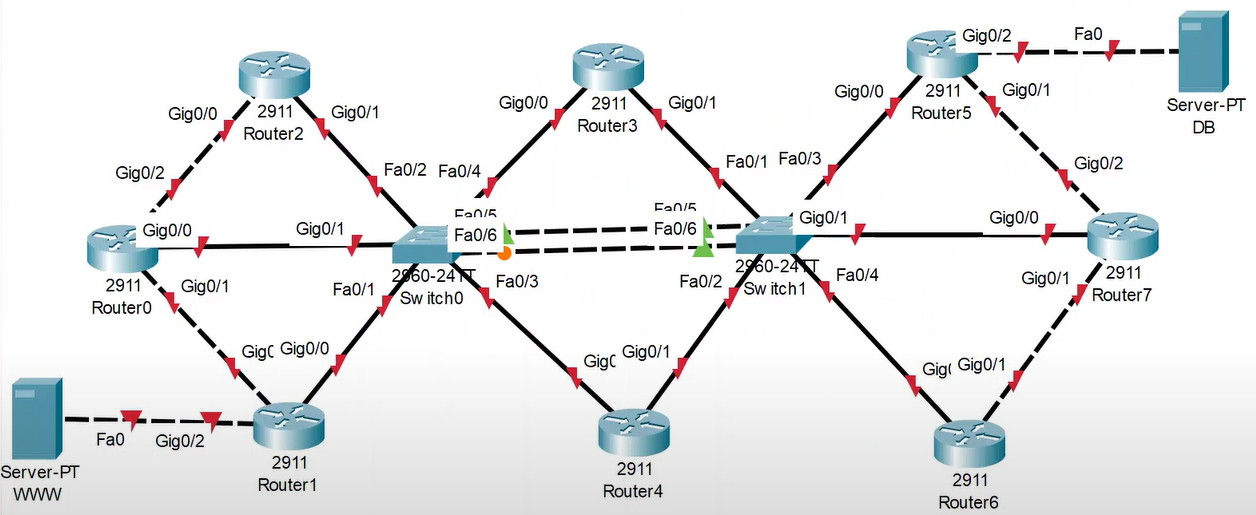
În acest laborator vorbim despre configurarea **GRE – Generic Routing Encapsulation**, care este un fel de VPN.

Link-uri de Referință: [aici1](https://www.nstec.com/what-is-the-difference-between-gre-and-vpn/) (diferența dintre GRE și VPN), [aici2](https://community.cisco.com/t5/networking-documents/how-to-configure-a-gre-tunnel/ta-p/3131970) (despre GRE).

Pentru configurarea tunelului GRE: ***L5.pdf*** (am scris în PDF mai multe detalii și rezolvări); ***L5\_Topologie*** → găsim topologia din PDF pe care putem lucra; ***L5 Comenzi.txt*** → găsim comenzile din PDF scrise în ordine

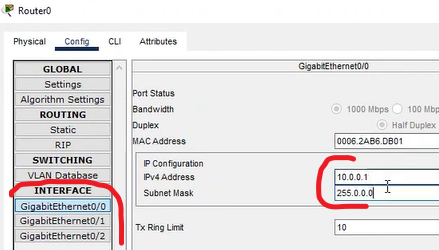
În folderul **1) VPN+OSPF (Bonus - Etherchannel)**

***L5\_VPN\_S1*** → este fișierul rezolvat. Încercăm să creăm un tunel (VPN) între frontend și backend. Mai jos, avem rezolvarea și modul de gândire:

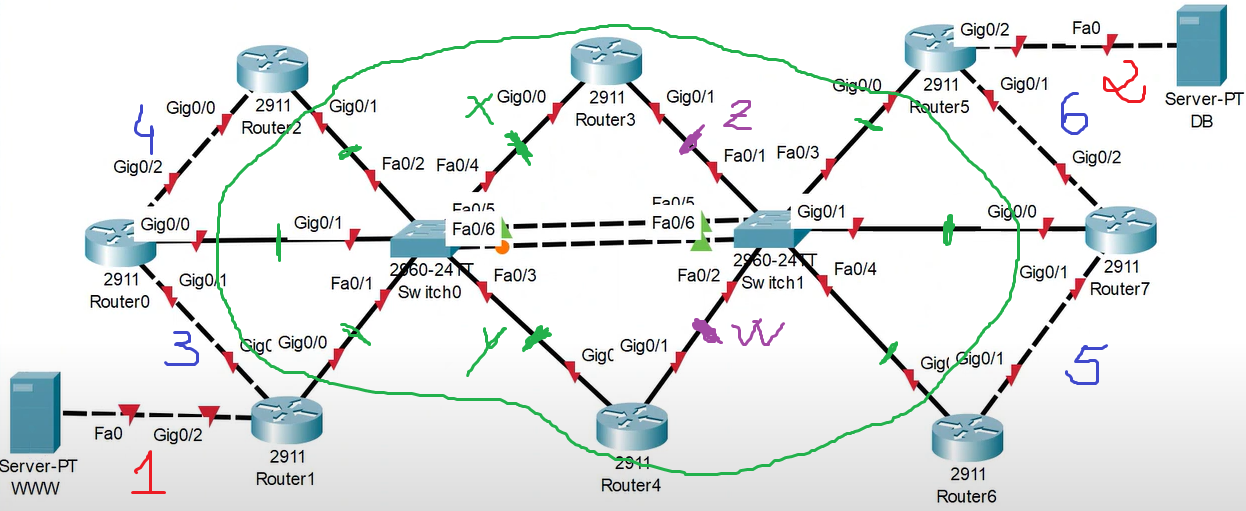


* Ne punem întrebarea ***câte rețele avem***. Dacă nu cunoaștem numărul, nu știm câte IP-uri trebuie alocate, care este masca asociată etc... Și vrem să alocăm *cât mai puține IP-uri*.

**!!!Un router are un modul, cu mai multe interfețe (ex: Giga0/0, Giga0/1...) și fiecare interfață trebuie să aibă o rețea diferită (când e legat de alte routere, nu vorbim de switch-uri).**



După această regulă, **R0R2** este o rețea diferită de **R0R1**; **R7R5** diferit de **R7R6**; **R2R0** diferit de **R2R-**; **R1R0** diferit de **R1R-**; **R5R7** diferit de **R5R-**; **R6R7** diferit de **R6R-**; **R3R-(rețea spre S0)-** diferit de **R3R-(rețea spre S1)-**; **R4R-(rețea spre S0)-** diferit de **R4R-(rețea spre S1)-**.



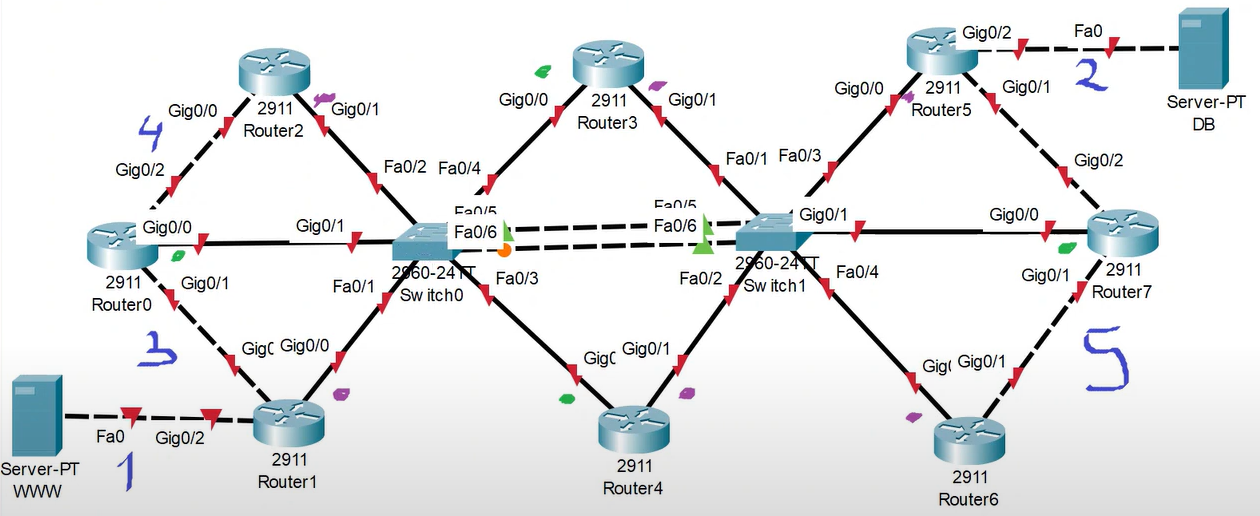
Obținem deci **6** rețele diferite.

De ce ce avem în interiorul cercului verde **nu** poate fi considerat o singură rețea (să avem deci **7 rețele**)? → Conform ce am spus mai sus, **R3** și **R4** au câte 2 interfețe și fiecare interfață trebuie să aibă o rețea diferită (deci, ***x*** nu poate fi în aceași rețea cu ***z*** și ***y*** nu poate fi în aceași rețea cu ***w***).

De aceea, în interiorul cercului trebuie să avem **2** rețele (cea cu *verde* și cea cu *mov*)

→ **TOTAL: 8 REȚELE**.

* Acum încercăm să împărțim ce este în interiorul cercului astfel încât să avem cât mai puține ***IP-uri redundante***. La momentul actual, conform desenului de mai sus avem 8 (*verde*) și 2 (*mov*). 8+2(NA+BA)=10 **<** 16=2^4 & 2+2(NA+BA)=4 **=** 4=2^2. IP-uri redundante = 6. Dacă facem o împărțire de genul 6 și 4 astfel încât ambele să fie mai mici decât 8, atunci obținem doar 2 IP-uri redundante. O posibilă împărțire:



* Acum trebuie să ***ordonăm*** rețelele ***descrescător*** ca ***număr de device-uri conectate***, pentru a le putea asigna IP-uri în mod corespunzător (în cerință ni se precizează să folosim 10.0.0.0/8).

Deci, avem următoarele rețele cu nr device-uri conectate:

R1 – 6 device-uri(+2 – *NA și BA mereu tb luate în calcul*) <= 8=2^**3**: 10.0.0.0/29=32-**3**

R2 – 4 device-uri < 8=2^**3**: 10.0.0.8/29=32-**3**

R3 – 2 device-uri <= 4=2^**2**: 10.0.0.16/30=32-**2**

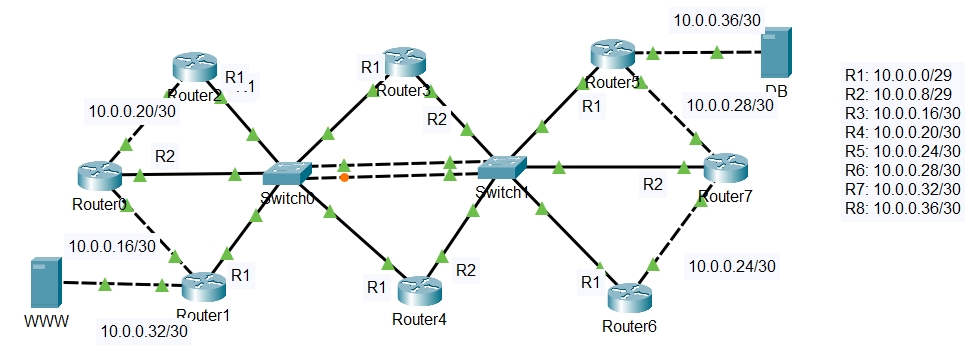
R4 – 2 device-uri: 10.0.0.20/30

R5 – 2 device-uri: 10.0.0.24/30

R6 – 2 device-uri: 10.0.0.28/30

R7 – 2 device-uri: 10.0.0.32/30 → folosim pentru www (frontend)

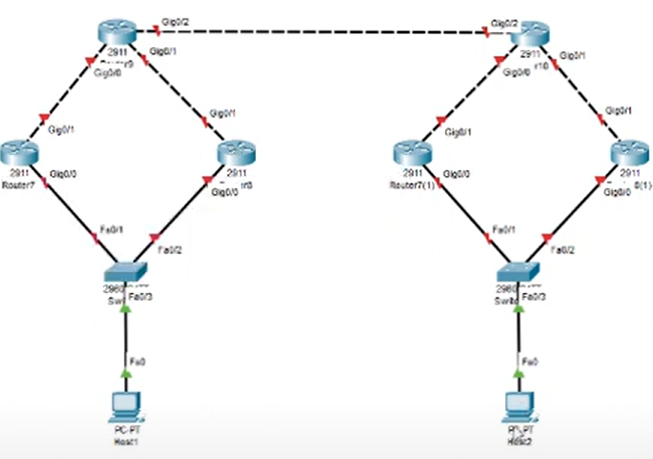
R8 – 2 device-uri: 10.0.0.36/30 → folosim pentru DB (backend)



Se configurează Router-ele cu comenzile simple și cu OSPF. Se configurează Serverele exact ca PC-urile.

* ***Cum alegem între ce routere definim VPN-ul?*** Am spus că vrem să facem VPN între WWW și DB. În cazul nostru, răspunsul este evident: alegem Router1 și Router5.

Dar ce facem dacă avem o topologie de genul acesta și vrem VPN între cele 2 PC-uri?

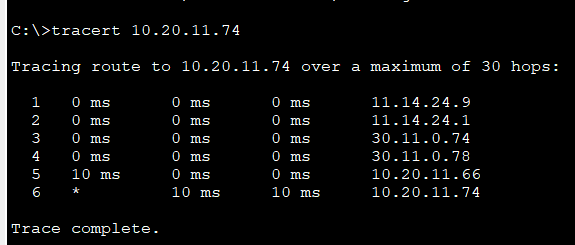


Nu am știi exact care din cele 2 routere de la fiecare PC să alegem. Avem 2 soluții (bine, noi putem să alegem să creăm tunel VPN între cele 2 routere de sus, dar nu e best-practice):

1. Fie, ne uităm la **default gateway-ul PC-urilor**, ca să vedem de care router se leagă ***(!!! Default gateway PC = IP router de care se leagă***)

SAU

1. Scriem în command prompt al unuia dintre PC-uri: ***tracert -IP celălalt PC-***



* Acum ne punem întrebarea ***de ce avem 2 legături între cele 2 switch-uri*** care par a produce un ciclu: una din ele nu este redundantă (mai ales că apare cu portocaliu, deci pare că nu merge)?

Pentru a înțelege asta, trebuie să știm că toate **switch-urile** au [**protocolul STP**](https://www.pluralsight.com/blog/it-ops/switching-and-stp), care identifică dacă avem ciclu și, după un număr de repetări, aruncă pachetul, ca să nu avem buclă la infinit (există un protocol similar și pentru routere – **TTL***\_default:16 bucle\_*).

De asemenea, într-un **switch** avem ***mai multe porturi de intrare,*** care primesc trafic de la diferite device-uri, și switch-ul transmite acest trafic pe ***o singură interfață de ieșire***. Astfel, există riscul ca toate device-urile combinate să producă un trafic de ieșire mult mai mare decât capacitatea interfeței de ieșire a switch-ului. Astfel, avem posibilitatea de a ne defini ***EtherChannel*** (ne defineam un singur port virtual din mai multe interfețe, în cazul nostru din cele 2 legături) și STP-ul spune că traficul va fi împărțit în mod egal între aceste interfețe.

Comenzile pentru Ethernet sunt în ***Listă Comenzi.txt***, la switch-uri, dar sunt facultative.