Ex1

Am facut reprezentarea, dar nam atras atentie ca trebie deodata de salvat fara setari. Scuzati. Eu am salvat doar atunci cand cineva a atras atentie la acest moment in clasa noastra, si au spus ca trebuie sa fac in asa mod. De aceea eu am setat doar la momentul cand deja erau setate vlanurile. Oricum este cea mai usoara parte din test de aceea sper ca nu este o problema mare.

Ex2

10.100.200.192/26 – initial

Vlan2 – 27 hosturi

10.100.200.192 – de retea

255.255.255.192 – masca in zecimal

11111111. 11111111. 11111111.11000000 – masca in binar

Avem nevoie de 27 hosturi, de aceea extindem pana la 32(30 accesibile)

11111111. 11111111. 11111111.11100000 – masca in binar extinsa

10.100.200.192/27

Diapazon: 10.100.200.192/27 - 10.100.200.223/27

Vlan3 – 12 hosturi

10.100.200.224/27– de retea

255.255.255.224 – masca in zecimal

11111111. 11111111. 11111111.11100000 – masca in binar

Avem nevoie de 12 hosturi, de aceea extindem pana la 16(14 accesibile)

11111111. 11111111. 11111111.11110000 – masca in binar extinsa

10.100.200.224/28

Diapazon: 10.100.200.224/28 - 10.100.200.239/28

Vlan4 – 4 hosturi

10.100.200.240/28– de retea

255.255.255.240 – masca in zecimal

11111111. 11111111. 11111111.11110000 – masca in binar

Avem nevoie de4 hosturi, de aceea extindem pana la 8(6 accesibile)

11111111. 11111111. 11111111.11111000 – masca in binar extinsa

10.100.200.240/29

Diapazon: 10.100.200.240/29 - 10.100.200.248/29

Vlan5 – 4 hosturi

10.100.200.240/29– de retea

255.255.255.248 – masca in zecimal

11111111. 11111111. 11111111.11110000 – masca in binar

Avem nevoie de 4 hosturi

11111111. 11111111. 11111111.11110000 – masca in binar (nu extindem masca in binar deoarece ne ajung hosturi in aceasta masca)

10.100.200.248/29

Diapazon: 10.100.200.248/29 - 10.100.200.255/29

Ex2

1. Tinand cont de conditiile impuse noua ne ajung adrese de host ca sa acoperim toate dispozitivele ale vlanurilor.
2. Este prezent mai sus.

Ex3

Intai am configurat switchurile prin interfata data base vlan, si am adaugat toate vlanurile existente

Apoi am intrat in fiecare calculator sa pun access si sa ii atribui la ce vlan apartine.

Apoi am setat ruterele cu porturi virtuale, si am folosit encapsulation dot1q pentru ca ruterul sa poata lucra cu ele. Si prin ip adresss sa aiba adresa acestor vlanuri.

Ex4

Pentru fiecare interfata de port virtual am scris ip dhcp pool, apoi prin network am scris adresa vlanurilor de start, si masca lor inversa. Si prin default-router am scris adresa de gateway.

Ex 5

Prin access-list vlan\_numar permit (adresa\_vlanului) masca\_opusa\_vlanului data acces sa lucreze ruterul cu vlaurile date.

ip nat pool TRANS 219.219.217.1 219.219.219.1 netmask 255.255.255.192 adaugam access list-urile pt fiecare vlan la acel pool.

Am scris de doua ori fiindca este un diapazon de ipuri care il scriem. Si masca retelei.

Scriu ip nat inside source list nr\_vlan pool TRANS overload pentru toate vlanurile

si dupa fac ip nat inside/outside pe interfetele virtuale a router ului pt vlanuri.

ip nat inside - interfetele interioare ip nat outside - interfata exterioara

ex6

toate adresele de la dispozitive care au nevoie de acces la un server din afara retelei au nevoie sa iasa din ea. Dat fiind ca numarul de ipuri este limiitat, toate dispozitivele din reteaua data se schimba la iesire in ip public, ca sa se economiseasca aceste ipuri. Acele din interior sunt private. Si de obicei avem doar un singur ruterr de iesire din reteaua data, si cand pachetul ajunge la acest ruter pentru a iesi, el este reimpachetat cu ip ruterului, adica acel public. Apoi merge spre destinatie si se intoarce inapoi, fiindca ipul public este unic. Si cand ajunge la ruterul din care a iesit, acesta tine minte in tabel de la ce dispozitive el a primit pachete de iesire. Si in momentul cand pachetul se intoarce, ruterul deja stie cui trebuie sa il intoarca, si face asta automat, schimband ipul pachetului, si adresa finala la care treebuie sa ajunga. Asa face si la iesire.

Ex7

In transferul de date intre calculator si server, este implicat nivelul 3 de protocoale TCP/IP pe Routerul NAT. Fiindca nivelul 2 este folosit doar la switchuri de gradul 2, care nu le intereseaza ce informatie au de transmis, doar stiu unde trebuie sa o duca. Iar routerele si switchuriile de gradul 3(care indeplinesc aceleasi functii ca si ruterul) au nevoie de mai multe date pentru rutarea si lucrul cu frameurile. Ele cauta directia optima singuri, au mai multe tabele de informatie care au nevoie so retina la ei, dar nu doar tabelul ARP.

Ex8

Eu am reconstruit topologia in asa mod ca toate dispozitivele din vlan 2 si 3 sa fie la un switch2, si de aceea devine radacina in raport cu aceste vlanuri. Si la switch1 am pus dispozitivele cu vlan 4 5 si el a devenit radacina in raport cu vlanurile date.

Folosind comanda show spanning-tree vlan la fiecare switch am determinat ca:

Switch 0: Vlan2

Fa0/1 radacina

Fa0/2 alternativa

Fa0/3 desemnata

Fa0/4 desemnata

Vlan3:

Fa0/1 radacina

Fa0/2 alternativa

Fa0/3 desemnata

Fa0/4 desemnata

Vlan4

Fa0/1 radacina

Fa0/2 alternativa

Fa0/3 desemnata

Fa0/4 desemnata

Vlan5

Fa0/1 radacina

Fa0/2 alternativa

Fa0/3 desemnata

Fa0/4 desemnata

Switch1: vlan 2

Fa0/6 desemnata

Fa0/5 radacina

Fa0/3 desemnata

Vlan3

Fa0/4 desemnata

Fa0/5 radacina

Fa0/6 desemnata

Vlan4

Fa0/6 desemnata

Fa0/5 radacina

Fa0/2 desemnata

Vlan5

Fa0/6 desemnata

Fa0/5 radacina

Fa0/1 desemnata

Switch2: vlan2

Fa0/5 desemnata

Fa0/2 desemnata

Fa0/6 desemnata

Vlan3

Fa0/4 desemnata

Fa0/5 desemnata

Fa0/6 desemnata

Vlan4

Fa0/3 desemnata

Fa0/5 desemnata

Fa0/6 desemnata

Vlan5

Fa0/1 desemnata

Fa0/5 desemnata

Fa0/6 desemnata