

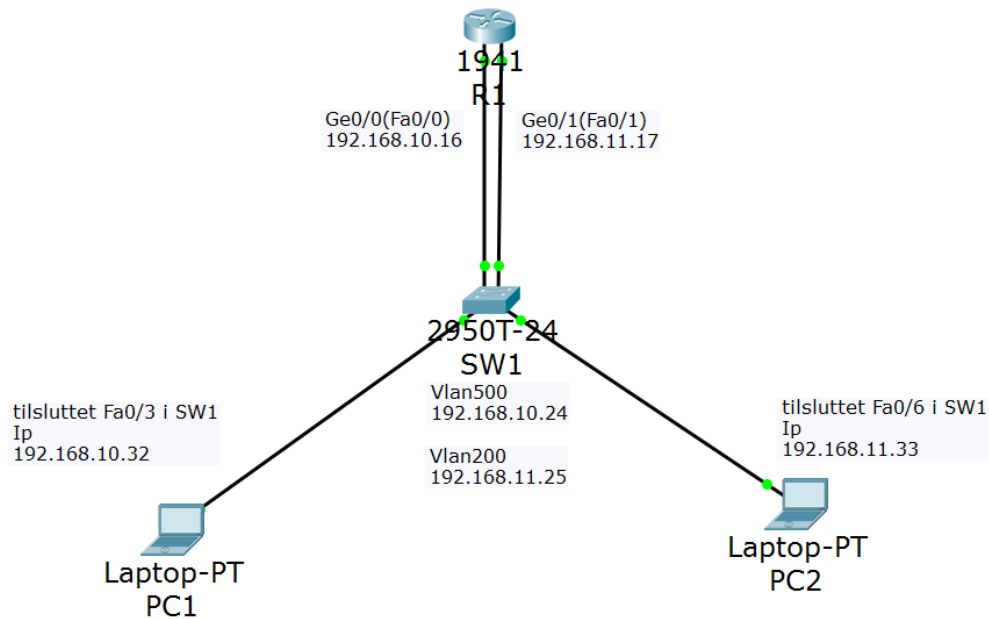
# Cisco C-sharp telnet project

## Problemformulering:

Vi har fået til opgave at lavet C# program der skal lette opsætning og vedligeholdelse af netværksudstyr i store netværk, er det en stor fordel at automatisere opsætning af enheder.

## Teknisk Indhold:

Her et billede af vores netværk lavet i pakket tracer:



Her på billedet kan vi se hvilke ip'er de forskellige devices har og hvilket vlan som vi har brugt.

Lavet af: Lavet af: Nicklas Pihl Mathiesen og Morten Peter Holmquist Larsen

Klasse: H1 Datateknikker Programmør

```
public static void R1Config()
{
    TelnetConnection T1;
    List<String> Result = new List<String>();
    int lineCounter = 1;

    T1 = new TelnetConnection("192.168.10.16", 23);

    T1.CiscoLogin("cisco");
    T1.CiscoEnable("cisco");
    T1.CiscoCommand("conf t");
    T1.CiscoCommand("inter fa0/1");
    T1.CiscoCommand("ip add 192.168.11.17 255.255.255.0");
    T1.CiscoCommand("no shut");
    T1.CiscoCommand("exit");
    T1.CiscoCommand("exit");
    T1.CiscoCommand("write");
    Result = T1.CiscoCommand("Show start");

    Result.ForEach(delegate (String line)
    {
        Console.WriteLine("{0}: {1}", lineCounter, line);
        lineCounter++;
    });
}
```

På billedet her kan vi se de funktioner vi valgt at bruge i vores projekt.

Først kalder vi vores class og giver den et navn, derefter laver vi en list som skal indeholde en string(tekst), så laver vi en int(tal), kalder vores class og giver den nogen paramenter, give vores kode til at logge ind og til at komme ind i enable, så lave vi nogen commands, til sidst lave vi et loop som viser alle de linjer som er vi vores startup config.

```
try
{
    int input = int.Parse(Console.ReadLine());
}
catch
{
    Console.WriteLine("that's not an option");
}
```

På billedet her kan vi se de try catch vi har brugt.

Tjekker om det du skrive er i den rigtige format.

```
if (input == r1)
{
    R1Config();
    Console.WriteLine("Press any key too exit");
    Console.ReadKey();
    runCommand = false;
}
else if (input == sw1)
{
    runCommand = true;
    Sw1Config();
    Console.WriteLine("Press any key too exit");
    Console.ReadKey();
    runCommand = false;
}
else if (input == both)
{
    runCommand = true;
    R1Config();
    Console.WriteLine("så går vi videre til SW1");
    Console.WriteLine("press anny key too continue");
    Console.ReadKey();
    Sw1Config();
    Console.WriteLine("Press any key too exit");
    Console.ReadKey();
    runCommand = false;
}
```

På billedet her er vores if statements.

Her ser du hvad der sker ved de forskellige funktioner. F.eks. hvis du trykker 1(som er lig med r1) vil den gå i gang med vores Router konfiguration, og der efter vil der komme en linje med en tekst hvor du kan trykke en knap af eget valg for at exit router funktionen eller hvis du trykker 2(som lig med sw1) så vil den gå videre med vores switch konfiguration hvor du så efter den er færdig igen kan trykke på en knap efter eget valg for at exit. Hvis du så trykker 3(som er lig med begge funktioner) vil den først gå i gang med vores router funktion, derefter skriver den så "så går vi videre med SW1" hvor du så skal trykke på en knap efter eget valg for at gå videre med vores switch konfiguration, hvor du så igen skal trykke på en knap efter eget valg for at exit.

### Konklusion:

Vi har fundet ud af det er god ting at lave hvis man er i en virksomhed hvor der kan være mange medarbejdere og dermed mange pc'er med til tilhørerne switches og routere der skal konfigureres. Til det syntes vi at det kan være et godt "redskab" at have med til det i stedet for at skulle gøre det flere gange. Men vi syntes så at det kunne være bedre at gøre det med SSH i stedet for da det er en mere sikkert end TELNET.