**Python ile Ağ Topolojisi Çizimi**

**Ağ Topolojisi Geliştirme Uygulaması**

**Adı Soyadı:** Süleyman Şahin

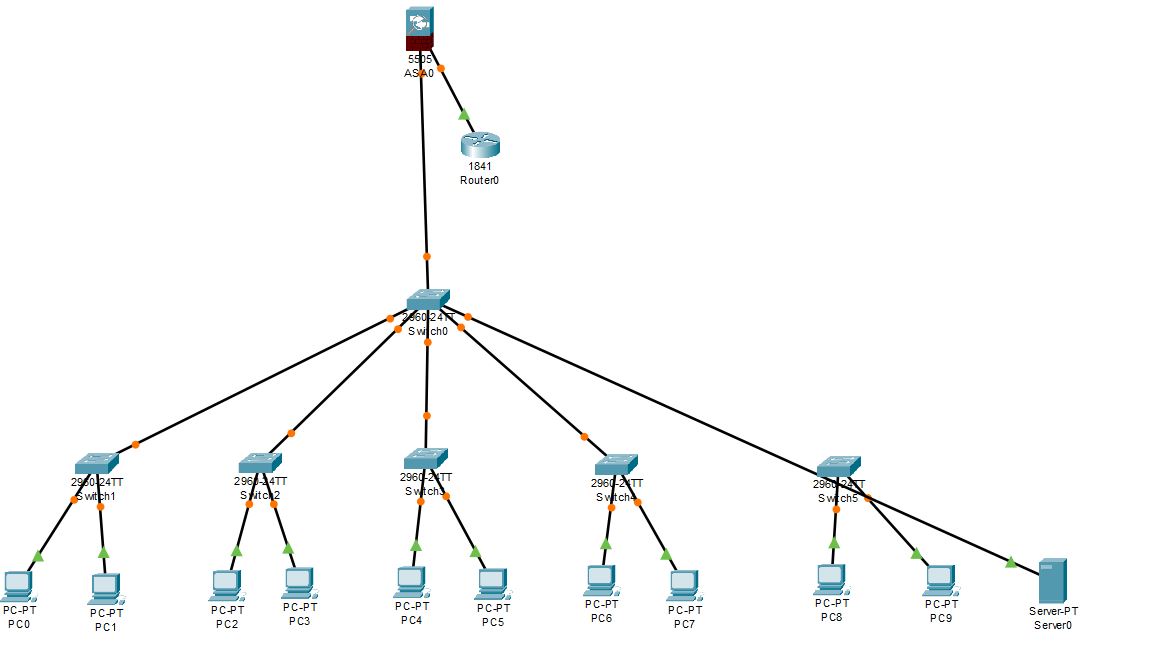
**Numarası:** 21370031057

**Program:** Bilgisayar Mühendisliği

1. ***Giriş***

Bu raporda, Cisco Packet Tracer ile oluşturulmuş ağ topolojisini çizen Python kodu anlatılmaktadır. Ağ topolojisinin görselleştirilmesi için **matplotlib** ve **networkx** kütüphaneleri kullanılmıştır. Bu dosyalar, düğümler ve parçalarının temsil edildiği bir grafik oluşturmak ve bu grafiği görselleştirmek için kullanılır. Bu raporun amacı, verilen kodun nasıl değiştiğini ve bu sürecin nasıl ilerlediğini detaylandırmaktır.

**2.Çizimi Yapılan Ağ Topolojisi**



**3.Kullanılan Kütüphaneler**

Bu Python programı iki ana kütüphanede kullanılmaktadır:

**.** **NetworkX** : Graf teorisini kullanarak düğüm ve kenarlarla grafik yapıları oluşturmak ve bu grafikleri oluşturmak için kullanılan bir kütüphanedir.

**. Matplotlib** : Python'da görseller çizmek için kullanılan popüler bir veri görselleştirme kütüphanesidir.



Yukarıdaki kod, bu iki kütüphaneyi içeri aktararak, başında kullanıma hazır hale getirir.

**4. Ağ Topolojisi Modeli**

4.1. Ağ Cihazları ve Bağlantıları

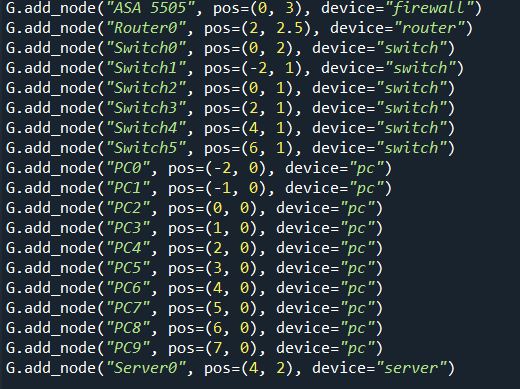
Ağ topolojisinde, çeşitli cihazlar (PC, router, switch, sunucu, güvenlik duvarı) ve bu cihazlar arasında yapılar bulunur. NetworkX'te her bir cihaz bir düğüm olarak, cihazlar arasındaki yapılar ise kenarlar olarak temsil edilir.



Bu satır ile bir grafik nesnesi oluşturulur. Bu grafiklerin, cihazların ve bağlantıların yer alacağı yapıdır.

4.2. Düğümlerin (Cihazların) Eklenmesi

Her bir ağ cihazında, ağ grafiğinde bir düğüm olarak eklenir. Aşağıdaki kodda, farklı cihazlar farklı koordinatlara yerleştirilmiştir.



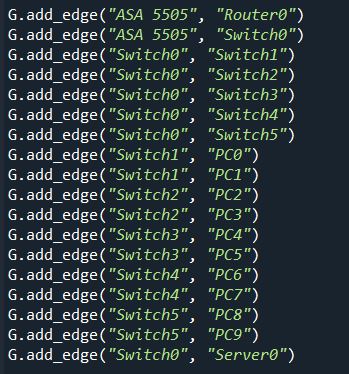
**. add\_node** fonksiyonu ile her bir ağ cihazı (düğüm) eklenir.

**. pos** Parametreler ile ölçümlerdeki konum (x, y) koordinatları ile belirlendi.

**. device** Parametreler ise cihazdaki bileşenlerini eklemek için eklenmiştir. (Güvenlik duvarı, yönlendirici, anahtar, PC, sunucu)

4.3. Bağlantıların Eklenmesi

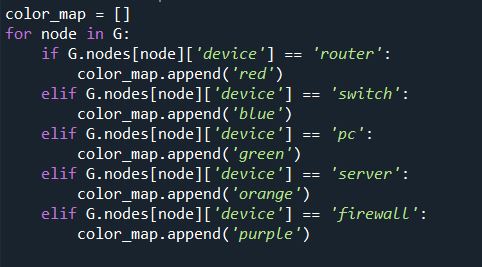
Cihazlar arasındaki kopuklukları, iki düğüm arasında kenar oluşturmak için:



Yukarıdaki kodlar, cihazlar arasındaki fiziksel verileri içerir.

**5.Görselleştirme**

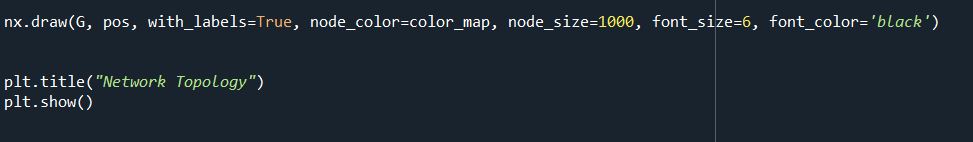
Grafik yapısındaki düğümleri ve renkleri belirli renklerle görselleştirmek için düğüm ve kenar renkleri belirlenir. Her cihaz türü için farklı bir renk atanmıştır:



Bu ayarlanabilir, cihaz türü için uygun bir renk atanarak görsel ayrımlar korunur. Yönlendirici için kırmızı, değiştirici için mavi, PC'ler için yeşil, sunucu için turuncu, güvenlik duvarı için ise mor renk kullanılmıştır.

5.1. Grafiğin Çizimleri

Son olarak ağ topolojisi aşağıdaki kodla görselleştirilmiş:



**. nx.draw** : Ağı ve düğümleri çizer.

**.** **pos**: Düğüm isimlerini belirler.

**. node\_color**: Düğüm renklerini belirler.

**. node\_size**: Düğümün büyüklüğünü belirler.

**. font\_size** ve **font\_color**: Düğüm etiketlerinin (cihaz isimleri) görünüm ayarları.

**. plt.title** : Grafiğin isimlerini belirler.

**. plt.show** : Grafiği ekranda görüntüler.

**6. Sonuç**

Bu raporda, bir ağ topolojisinin Python aracılığıyla nasıl görselleştirildiği anlatılmıştır. **NetworkX** kütüphanesi ile düğümler ve kenarlar aracılığıyla ağ bileşenleri ve bağlantıları tanımlanmış, **Matplotlib** ile bu yapı görsel olarak çizilmiştir. Cihazların yapılan ayarları ve cihaz türü için farklı renk kullanımı, daha anlaşılır olması sağlandı. Bu tür görselleştirmeler, özellikle ağ tasarımı ve yönetimde büyük rahatlık sağlar.

**7. Eklemeler ve Geliştirmeler**

Bu Python programı, temel bir ağ topolojisini çizmektedir. Program daha da geliştirilebilir ve şu eklemeler yapılabilir:

**.** Cihazın daha ayrıntılı parametreleri (IP adresleri, MAC adresleri vb.).

**.** Dinamik olarak cihaz ve bağlantı ekleme seçenekleri.

**.** Cihazlar arası verileri dağıtmayı değiştirenler.