

T.C. GALATASARAY ÜNİVERSİTESİ
MÜHENDİSLİK VE TEKNOLOJİ FAKÜLTESİ

INF 443
DAĞITIK SİSTEMLER VE UYGULAMALARI
PROJE RAPORU
GRUP 4

Yasemin AKAN

Zeki GÜLER

Volga GÜNEL

Mustafa ÇÖREKÇİ

Orhan TOMBUL

Bölüm: Bilgisayar Mühendisliği

Atay ÖZGÖVDE

Serhan DANIŞ

1. Proje Genel Tanımı

Dağıtık Sistemler ve Uygulamalar projesi, merkezci olmayan mikroblog temelli bir sosyal ağ tasarımı ve gerçekleştirmesidir. Bu sistemde aracı ve yayıncı olmak üzere iki farklı peer tasarlanması istenmektedir. Aracı peer'ların görevi ağa katılan diğer aracı veya yayıncı peer'lar sayesinde ağ genişletmektir. Projenin aşamaları aşağıdaki gibidir:

- Protokol tasarımı
- Github repository
- Aracı peer (Negotiator) gerçekleştirilmesi
- Yayıncı peer (Blogger) gerçekleştirilmesi
- Arayüz tasarımı

2. Proje Gelişim Aşamaları

2.1. Protokol Tasarımı

Proje için ilk olarak protokol tasarımı yapılmıştır. Aracı ve yayıncı peer'lar için iki ayrı bölüm olarak düşünülmüştür. Aracı peer'lar sadece bağlantı bilgilerini aktarmak ve birbirlerini tanımayan düğümler için herkesin tanıdığı bir düğüm görevlerini yerine getirmektedirler. Bu yüzden bağlantı kurmak için peer bilgilerinin gönderildiği bir protokol mesajı, bu bilgilerin kontrol edilmesi için bir protokol mesajı, bir de bağlantı kurma gerçekleşmişse o peer'dan onun tanıdığı düğümlerin bilgilerini isteme için bir protokol mesajı yeterli olmuştur.

Yayıncı peer'lar için tweet atma ve özel mesaj işlemleri de dahil olduğu için bu haberleşmenin şifreli olması gerekmektedir. Bunun için yayıncı peer'lar arasında asimetrik şifreleme algoritması olan RSA kullanıldığı için açık anahtar paylaşımı gerekmektedir. Ayrıca imzalama ile karşı peer'ın kimliği kontrol edilmelidir. Protokolde bu iki işlem önce ayrı iki protokol mesajı olarak düşünülmüş, daha sonra tek protokol mesajına indirilmiştir. Protokol tasarımı github repository'ye bulunmaktadır. Yayıncı peer'ların birbirini takip etme, takipten çıkma, engelleme, engeli kaldırma, özel mesaj gönderme işlemlerini de gerçekleştirmeleri beklenmektedir. Bunlar için ayrı ayrı protokol mesajları yazılmıştır.

2.2. Github Repository

Bu projeyi geliştirirken projede yapılan her değişikliğin grup bireyleri tarafından takip edilebilmesi için bir Github repository oluşturulmuştur. Aşağıdaki linkten ulaşabilirsiniz.

https://github.com/orhantombulgsu/dagitik_home_group4

2.3. Aracı Peer (Negotiator) Gerçeklemesi

Protokol tasarımıından sonra ilk olarak aracı peer geliştirilmeye başlanmıştır. Aracı peer'lar tanıdığı düğümlerin bilgilerini UUID'ler key ve diğer bilgiler (host, port, name, type) values olmak üzere bir dictionary'de tutarlar ve bu bilgiler json uzantılı bir dosyaya kaydedilir. Program her başlatıldığında daha önceden kayıtlı olan bilgiler tekrar aracı peer'ın dictionary'sine yüklenir. Her peer'ın eşsiz bir UUID üretmek için uuid kütüphanesi import edilmiş ve bilgisayara özel bir UUID üreten uuid.uuid1() kullanılmıştır. Her aracı peer'ın Logger, Server, Client ve UserInput olmak üzere çalıştırdığı 4 thread'i vardır. UserInput thread konsoldan input alınmak için oluşturulmuş olup aracı herhangi bir anda protokol mesajını kullanıcıdan alabilmektedir. Server thread soketten gelen mesajlarını alarak bu mesajlara uygun cevabı readerParser metoduyla üretir ve soketten gönderir. Eğer gelen mesaj yeni bağlantı isteği içinse bağlantı kurmak isteyen karşı peer'ın bilgilerini kontrol etmek için Client thread açılır ve control() metodu sayesinde karşı peer'ın Server'ına bilgilerini isteyen protokol mesajı gönderir ve o soketten cevap bekler. Gelen cevaba göre karşı peer'ın client ve server'ından gelen bilgilerini karşılaştırarak onay veya hata mesajını server'a döndürür. Projede protokolda tasarlanan aracı peer'ların yapması gereken işlemlerin hepsi gerçekleşmiştir. 2'si yayıncı peer 1'i aracı peer olmak üzere 3 bilgisayar kullanılarak test edildiğinde 2 yayıncı peer da aracıya bağlantı isteği gönderebilmiş, aracının tanıdığı düğümlerin bilgilerini isteyerek cevap alabilinmiştir. Github repository'de aracı peer'ın son hali Araci_v5.py dosyasıdır. Diğer versiyonlar eski versiyonlar olup, hem yeri gelindiğinde kullanılmak(denenmek) hem de projenin takibi açısından versiyonlanmıştır.

2.4. Yayıncı Peer (Blogger) Gerçeklemesi

Yayıncı peer, aracı peer'ın yapabildiği her şeyi gerçeklemesi gerektiği için geliştirilen aracı peer'ın üzerine eklemeler yapılarak devam edilmiştir. Yayıncı peer için işlemlerin şifreli gerçekleşmesi gerektiği için her yayıncı peer için açık ve kapalı anahtar üretilmiş ve bunlar dosyaya yazılmıştır. Yayıncı tekrar çalıştırıldığında yeni bir anahtar çifti üretmemesi için eğer

daha önceden üretilmişse dosyadan okuyarak aynı anahtar çiftini kullanması sağlanmıştır. Tanındığı diğer düğümlerin bilgilerini tutmak için aracı peer ile aynı yöntem kullanılmıştır. Fakat şifreleme olacağı için tutulan bilgilere ek olarak açık anahtar bilgisi eklenmiştir. Yayıncı peer'ın Logger, Server, Client, UserInput ve Arayüz olmak üzere 5 thread'i vardır. Aracı peer'da yapılan işlemler yayıncı için de aynıdır. Fakat yayıncı peer için client thread'in control() metodu açık anahtar paylaşımı ve kontrolü için de kullanılmaktadır. Protokolde karşı peer'dan açık anahtar ve gönderilen mesajdaki herhangi bir metnin kapalı anahtar ile imzalanmış hali aynı protokol mesajı ile istenmektedir. Bu yüzden karşı peer'ın mesajı soketten string olarak gönderilmiştir. Bu noktada karşılaşılan sorun, anahtarların string'den tekrar bir RSA nesnesine dönüştürülmesi zorluğu olmuştur. Çözüm olarak ise karşı peer açık anahtarı exportKey() metodu ile byte'a çevrilmiş, gönderilen mesajın tamamı string olsa da bu mesaj alındığında split() metodu sayesinde açık anahtar ve imzalı metin ayrılış, açık anahtar encode() ile byte'a çevrilerek importKey() metodu ile de açık anahtar elde edilmiştir. Açık anahtar ile imzalı metin açılabilmiş ve onay veya hata mesajı döndürülmüştür.

Yayıncı peer ara yüzden atmak istediği tweeti yazıp ilgili butona basıldığında, o anda sisteme bağlı olan diğer peer tarafından atılan tweet görünmektedir. Projede yayıncı peer için bu işlemler gerçekleştirilebilmiştir. Diğer işlemler de koda eklenmiş fakat çalıştırılamamıştır. Gerçeklenemeyen kısımlar ile ilgili planlanan adımlar da şöyledir. Yayıncı peer'ların takip ettiği kişiler, takip edenler, daha önce atılan tweetler, engellenen kişiler bir liste halinde tutulmaktadır. Gelen protokol mesajına göre her yeni ekleme veya çıkarma yapıldığında bu listeler ve listelerin içeriklerinin yazıldığı json uzantılı dosyalar güncellenmektedir. Yayıncı çıkış yapıp tekrar bağlandığında ara yüzde bunları görebilecektir. Github repository'de gerçekleştirilemeyen kısımlar için yapılan denemeler yorum satırına alınmıştır. Yayıncı için son versiyon Yayıncı_v6.py dir.

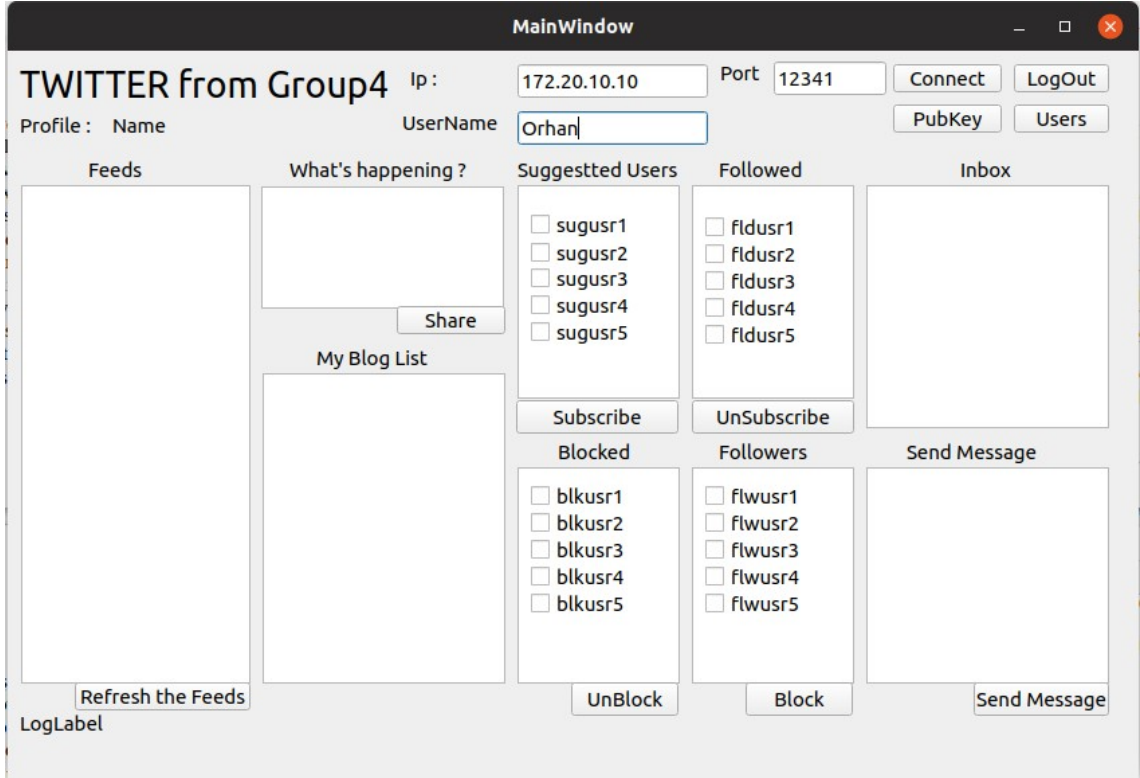
2.5. Arayüz (User Interface) Gerçeklemesi

Arayüz Yayıncının gerçekleştirilmesi için araç olarak kullanıldı. Son versiyonu Arayüz_v7.py olup Yayıncıda yazılan Threadlerin çağırılması ile bu python dosyası Yayıncı olarak çalışabilmektedir.

Arayüz tasarımında ilk olarak pyqt5 yüklenerek qt5 designer ile görsel olarak bir kullanıcı arayüzünde ne olacağı tartışıldı. Bunun üzerine bir arayüz oluşturuldu. Bu arayüzde ip ve port için textbox, kullanıcının kullanıcı adını girebildiği bir textbox, twit paylaşabildiği bir textbox, önerilen kullanıcılar bölümü için qlistwidget , takip edilenler için qlistwidget, takipçiler için qlistwidget, engellenenler için qlistwidget,

microblog için qlistwidget, twit akışı için qlistwidget, mesaj kutusu için qlistview ve bir çok button kullanıldı. Ayriyetten checkboxlar kullanıldı. Arka yüz için oluşturulan ui.py dosyasının içinde daha anlaşılabilir olması için buttonlara ve alanlara mantıklı isimler verildi. Örneğin connect butonu için connect_button adı verildi. Central Layout kullanılarak kullanıcı arayüzünün ekrana göre şekillenmesi sağlanmıştır.

Burada her butonun fonksiyonu oluşturuldu. Bazı butonlara birden fazla fonksiyon atandı. Örneğin connect butonu bir yandan ip ve port alıp bağlantı sağlarken bir yandan da profile adının karşısındaki alana kullanıcı adını girmektedir. Her aşamada test edilerek checkbox olsun, textbox olsun kontrolü sağlanarak githuba pushlanmıştır. Bu sayede kod anlamlı ve yazımı kolay olması sağlanmıştır. Arayüzümüz Şekil 1. de görülmektedir.



Şekil 1. Arayüz_Grup4