

T.C.
GALATASARAY ÜNİVERSİTESİ
MÜHENDİSLİK VE TEKNOLOJİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ



DAĞITIK SİSTEMLER ve UYGULAMALAR DERSİ
PROJE RAPORU

GRUP 5

ÖĞRENCİLER: Egemen İşgüder

Enes Dağ

İbrahim Fil

Timoteos Onur ÖZÇELİK

13 Ocak 2019

Peer-to-Peer Sistem Nedir? Neden Vardır?

P2P, istemci-sunucu mantığıyla çalışan, kullanıcıların istemci olarak bütün hizmetleri merkezi bir sunucudan aldıkları sistemlerin aksine eşlerin kendilerine düşen sorumluluklarını gerçekleştirmesi üzerine merkezilikten uzak kurulan bir mimaridir. Yani ağdaki trafik tek bir merkez sunucu üzerinden dönmemektedir. Bu sistemlerde veri akışı uçtan uca gerçekleşmektedir. Ağ trafiği tasarlanma şekline bağlı olarak aylak aracı-eş ve haberleşmek isteyen eş-eş çiftleri arasında ya da direkt olarak sadece eş-eş arasında gerçekleşmektedir. Veri akışı ve bağlantı işlemleri merkezi bir server üzerinden gerçekleşmediği için ağın herhangi bir noktasında yaşanan tıkanıklık sadece o haberleşmeyi etkiler fakat merkezi sistemlerde merkezde bulunan serverde yoğunluk olursa bunun etkisi bütün sistem üzerinde görülür (SPoF).

Proje kapsamında da her eşte sistemdeki arada kullanılan aylak eşler tarafından öğrenilen diğer eşlerin adreslerinin ve diğer bilgilerinin bulunduğu bir liste bulunmaktadır ve istediği hizmet varsa bunu elindeki adreslerle (eşlerle) anlık mantıksal bir bağlantı kurarak isteğini bildirir ve geri dönüşünü alır.

Projenin Tanımı Ve Amacı

Dağıtık Sistemler ve Uygulamaları dersi altında yapılan projenin amacı peer-to-peer sistem düşüncesiyle çalışan bir twitter sistemi yapmaktır. Amaç olarak merkezi olmayan bir sistem üzerinden yayıncıların kendi takip ettiği ve kendisini takip eden kişilerle twitterda olduğu gibi karşılıklı paylaşım yapabilmektir. Bu sistemde paylaşım yapan kişilere yayıncı adı verilmiştir. Ayrıca dışarıdan sisteme dahil olmak isteyen kişiler için üzerinde adres kayıtları bulunan aracı adında yapılar da vardır. İstemci-Sunucu mimarisinden farklı olarak bu yapıda bulunan her bir eş hem istemci hemde sunucudur. Bu şekilde eşten eşe haberleşme hedeflenmektedir.

Proje kapsamında, sistemin içerisinde iki tip eş bulunmaktadır:

1. Aracı
2. Yayıncı

Protokol Tasarımı

1.1 Eş (İ) → Aracı (S)

Table 1: Sisteme dahil olmak isteyen Eşin Aracıya isteklerinin protokol özeti

İstek	Parametre	Cevap	Parametre	Açıklama
INF	UUID;NICKNAME;IP;PORT;TYPE	HEL TOKEN		Login isteği ve Kabulü
LOG	UUID;NICKNAME;IP;PORT;TOKEN	HEL TOKEN		Login isteği ve Kabulü
		REJ		Hatalı bilgi geldi
LSQ		LSA	kullanici1;...;kullaniciN	Liste Sorgusu ve Yanıtı
QUI		BYE		Bağlantıyı kapatıyorum
		ERL		Daha kayıt Yapılmadı
Komut		ERR		Hatalı Komut

1.2 Aracı (İ) → Eş (S)

Table 2: Sisteme dahil olmak isteyen Eşe karşılık Aracının sorgularının protokol özeti

İstek	Parametre	Cevap	Parametre	Açıklama
WHO		MID	UUID	UUID Teyit İsteği
QUI		BYE		Bağlantıyı kapatıyorum
		Cevap yok		Cevap yoksa kayıt silinir

1.3 Yayıncı 1 (İ) → Yayıncı 2(S)

Table 3: Yayıncı 1'in Yayıncı 2'ye sorgularının protokol özeti

İstek	Parametre	Yanıt	Parametre	Tanım
PUB	açıkanahıtar	PUB	açıkanahıtar	Açık Anahtar İsteği
SMS	İmzalanmışhash;text	SMS	İmzalanmışhash;text	Yayıncı anahtar onay için imza isteği
POK		POK		Karşılıklı Açık Anahtar Onayı
POK		PER		Karşı Taraf Onaylamadı.
PER		PER		Açık Anahtar karşılıklı olarak onaylanmadı.
SUB		SOK		Subscribe İsteği
USB		UOK		Unsubscribe İsteği
DMB	mikroblogsayısı	MBM	text;timestamp	Mikroblog İsteği - Cevabı Döngü İçerisinde
BLG	text;timestamp	TKN		Blog Yayımlı yapıldı ve dağıtıldı
MSG	mesaj	MOK		Özel Mesaj İsteği
SBM		BOK		Bloklandın Bildirisi
SUM		ROK		Bloklandı Kaldırıldı Bildirisi
QUI		BYE		Bağlantıyı kapatıyorum
Komut		ERR		Hatalı Komut
Komut		ERK		İmza yani açık anahtar onaylanmadı
Komut		ERL		Kayıt Yapılmadı
Komut		BLK		Blok Listesinde
Komut		ERS		Subscribe Olunmadı

Aracı Nedir ve Neden Vardır?

Kelimenin tanımına bakarsak aracı bir konuda araya giren, anlaşma sağlayan kimse, arabulucudur, bizim projemizde de aracının yaptığı işlev de tam olarak budur. Tanımda belirtildiği üzere, aracı her zaman ayakta kalarak bağlantı beklemekte ve bağlantı geldiğinde bağlantıyı yapan eşin – özellikle yayıncıların – öncelikle kendisine kayıt olunması için bilgilerin gönderilmesini bekler ve eşlerin güvenilirliği onaylandığında liste sorgulaması yapan eşlere kendi bildiği diğer bütün eşlerin bilgilerini yollamaktadır ve bu nedenlerden ötürü arayüzü yoktur..

Aracının İşleyişi

Aracı tanımda belirttiğimiz gibi sürekli bağlantı beklemektedir. Bağlanmak ve diğer eşlere dair bilgileri öğrenmek isteyen bir eş ilk önce aracıya kayıt işlemini gerçekleştirmesi gerektir. Kayıt işleminin güvenliği için kayıt isteği geldiğinde kendi tarafında bir istemci oluşturarak gönderilen IP ve PORT'a bağlantı açar ve UUID bilgisini sorgular. Gelen bilgi sunucuya geri dönüş yaptırılarak sunucu tarafında UUID eşleşmesi sağlandığında olumlu geri dönüş mesajı bu ara işlemler tekrar ve tekrar gerçekleştirilmesin diye üretilmiş tekil bir TOKEN değeri geri döndürülerek kayıt işlemi tamamlanmış olur. Bundan sonra liste sorgulamalarına da döngü içerisinde elindeki liste bitene kadar eşler hakkındaki bilgiler geri döndürülür. Liste iletiminin bittiği de END protokolüyle belirtilir.

Yayıncı Nedir?

Yayıncı (Blogger), peer-to-peer twitter paylaşım uygulamasında mikroblog yapıda içerik gönderip alabilen yapıda, uygulamanın iki eş türünden biridir. Son kullanıcı tarafından bir arabirim aracılığıyla kontrol edilebilen yayıncı, hem yayıncılardan gelen isteklere cevap verebilmekte hem de arayüz üzerinden gelen komutlarla diğer yayıncılara istekte bulunabilmektedir. Yayıncı eş (peer), diğer eş tipi olan aracılardan yapabildiği her işlemi yapabilmektedir.

Bunlara ek olarak yayıncı kendisine has olarak gerçekleştirebildiği fonksiyonlar aşağıdaki gibidir:

- Açık anahtar (public key) paylaşımı ve hash gönderimi ile onaylanması
- Bağlanılan eşin belirli sayıdaki son mikroblog yayınlarını isteme
- Karşı eşin mikrobloglarına üyelik isteği gönderme (subscribe)
- Üyelikten çıkma (unsubscribe)
- Yayıncıya özel mesaj gönderme ve yayıncıdan özel mesaj alma
- Kendisini takip eden yayıncılara mikroblog yayını yapma
- Yayıncıya engelleme ve bunu iletme
- Yayıncının engelini kaldırma ve bunu bildirme

Yayıncı, gizlilik gereği bu faaliyetlerin büyük çoğunluğunu RSA ile şifrelenmiş şekilde gerçekleştirmektedir.

Yayıncı, temel olarak dört ana kısımdan oluşmaktadır:

1. *Listen* sınıfı
2. *Server* sınıfı
3. *Client* sınıfı
4. *UI* sınıfları

Yayıncı Listen Sınıfı

Yayıncı içerisinde bulunan bu sınıf, yayıncıya gelen bağlantı isteklerini karşılayarak, bir çeşit connection handler görevi görmektedir. Sınıf içerisinde bir accept () metodu kullanarak sonsuz döngü içerisinde gelecek olan bağlantılar beklenmektedir. Her yeni gelen bağlantı için yeni bir “server” sınıfı nesnesi oluşturulmaktadır.

Yayıncı Server Sınıfı

Yayıncı server sınıfında ise, bağlantı kurulduktan sonra diğer yayıncılar tarafından gelen olan istekler karşılanır. Bağlantı talepleri ve protokole dahil olan bütün istekler bu sınıf içerisinde yanıtlanır.

Sınıf içerisinde yer alan run metodunda bir sonsuz döngü içerisinde, şifrelenmiş ya da şifrelenmemiş tüm gelen (incoming) protokol mesajları alınıp protokol ve geriye kalan protokol mesajı olarak ayrıştırılarak parser metoduna aktarılmaktadır.

Parser metodunda ise en önemli kısımlardan biri gelen bağlantıların kayıt (login) işleminin gerçekleştirilmesidir. Gelen protokol mesajı bir kayıt talebi ise server yapısı bunu bir Client nesnesi oluşturarak karşı tarafla olan bağlantısını teyit eder. Bağlantı teyidi gerçekleştiyse eğer karşı tarafın bilgilerini arayüz, Server ve Client nesnelerinin ortak olarak paylaştığı bir sözlük yapısına kaydeder, bu sözlük program başında bir metin dosyasından okunur ve program sonunda ve sistemin beklenmedik olarak kapanmasına karşılık güncellenme yapılan her noktada dosya üzerine tekrar ve tekrar kaydedilir. Kayıt işlemi gerçekleştikten sonra ilgili cevap protokol mesajı karşı tarafa gönderilir. Server cevap protokol mesajlarını direk olarak send() methoduyla iletilir, başka bir arabirime ya da sınıfa ihtiyaç duymaz. Burada dikkat edilen nokta karşılıklı bağlantılar sadece istek anında oluşturulan Client nesneleriyle gerçekleştirildiğinden kayıt işlemi her seferinde gerçekleştirilmelidir. Bu işlemin her seferinde sorgulanmaması içerisinde için isteği alan eş tarafından üretilen ve eşler arasında onaylanan bir TOKEN değeri kullanılmaktadır. Bu token değeri onaylandığı sürece kayıt direkt olarak gerçekleştirilir.

Yalnız parser metodunda ayriyeten öncelik verilen bazı protokol mesajları bulunmaktadır. Bunlar diğer yayıncıdan yayıncıya yapmış olunan bağlantılara bağlı olmaksızın gerçekleştirebileceği protokollerdir. Mesela, karşı tarafın kendisine kayıt isteği geldiğinde bağlantı sorgulaması (WHO) veya bloklanma (SBM) ve blok kaldırma (SUM) işlemleri daha önce kayıt olunup olunmadığına veya takip edilip edilmediğine veya karşı tarafın blok edilip edilmediğine bağlı olmaksızın gerçekleştirilebilmelidirler.

Parser metodundaki diğer protokol mesajlarının incelenip cevaplarının hazırlanması ise kayıt aşamasına göre daha basit yapıda olup, uygulama protokolünde belirlenen şekilde mesajın incelenip, cevabın hazırlanıp, gönderilmesi esasına dayanır. Bununla ilgili olarak tek belirtilmeye değer ek bilgi, yayıncılar her istek yapıldığı anda birbirlerine kayıt işlemi gerçekleştireceklerinden kayıt işlemleri esnasında bilgilerin tutulduğu sözlük yapısından karşılıklı geçmiş bağlantı bilgileri kontrol edilir ve buna göre gelen istekler cevaplanır.

Sunucunun gönderdiği cevaplar bir eş liste içeriyorsa veya bir mikroblog serisi gönderecekse bunu bir döngü içerisinde gerçekleştirir ve döngü sonunda “END” mesajıyla döngü sonunu karşı tarafa belirtir.

Yayıncı Client Sınıfı

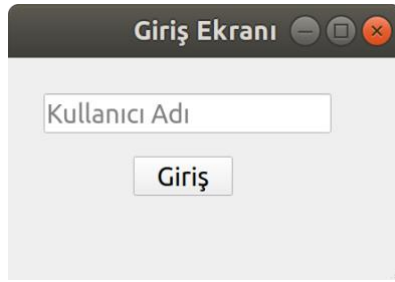
Client sınıfı temelde iki amaçla kullanılmaktadır. Birinci olarak, kayıt işlemi gerçekleştirmek isteyen yayıncıların kimlik kontrolünü gerçekleştirmek üzere IP ve PORT bilgisi üzerinden UUID sorgusu yapmak üzere; ikinci olarak, bilinen yayıncıların aktiflik testi ve arayüz üzerinden yapılan isteklerin diğer yayıncılara bağlantı kurularak protokol mesajlarının iletilmesi amacıyla kullanılmaktadır.

Yayıncı UI Sınıfları

Tasarım olarak üç ayrı UI sınıfı kullanılmıştır:

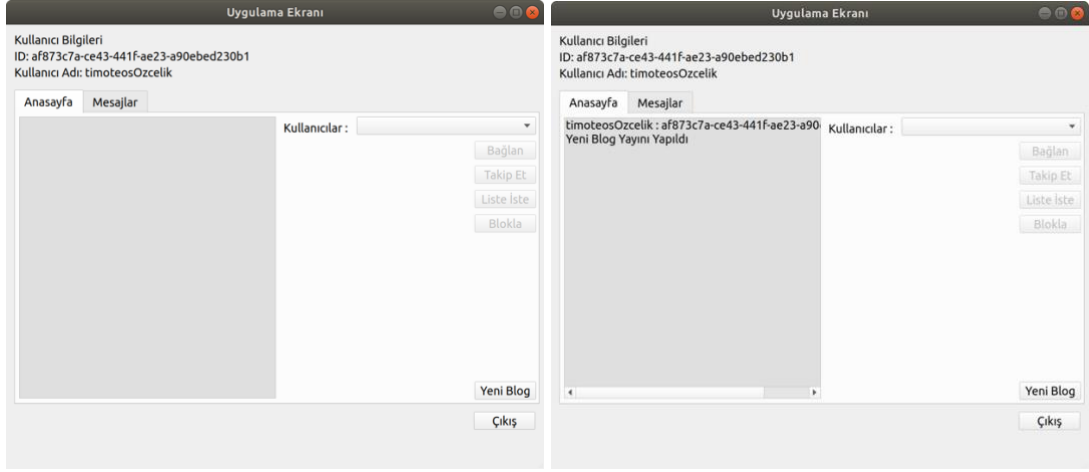
1. OpeningScreen_UI

Giriş Ekranı kullanıcının sistem içerisine girerken kullanacağı kullanıcı adını girdiği ekrandır. Giriş butonuna basıldığında, kullanıcı adı hanesi doluysa bu kullanıcı adı kabul edilir ve Anasayfa ekranına geçiş yapılır.



2. Homepage_UI

Anasayfa ekranı, yayıncıların temel isteklerini gerçekleştirdiği ekrandır. Bu ekran üzerinden kullanıcı bilgileri, yayıncıların listeleri ve yapılan bağlantılara göre, protokollerin uygulanış düzenine göre basılabilir ve basılamaz olarak ayarlanan Bağlan, Takip Et, Liste İste, Blokla butonları, paylaşılan ve takip edilen yayıncılardan alınan mikrobloglar için Anasayfa ve mesajlaşma için Mesajlar sekmeleri, yeni blog paylaşımı için Yeni Blog ve sistemden ayrılış için Çıkış butonları bulunmaktadır. Burada belirtmeye değer olarak Kullanıcılar kısmı, belli aralıklarla arkaplan üzerinde araçlardan liste sorgusu yapıp, daha sonrasında güncellenen bilinen yayıncılar listesindeki yayıncılarla bağlantı kurulmaya çalışılarak aktiflik testi yapılır ve aktif olan yayıncılar burada gösterilir.



3. NewMicroBlog_UI

Yeni Blog Yayınla ekranı, mikroblog yayını için içeriğin yazıldığı ve Yayınla butonuna basıldığında, ebeveyni olan Anasayfa ekranının add_blog metodunun çağrıldığı arayüz ekranıdır.

