

Bil 452 Proje

Fatih Furkan Has
141101024

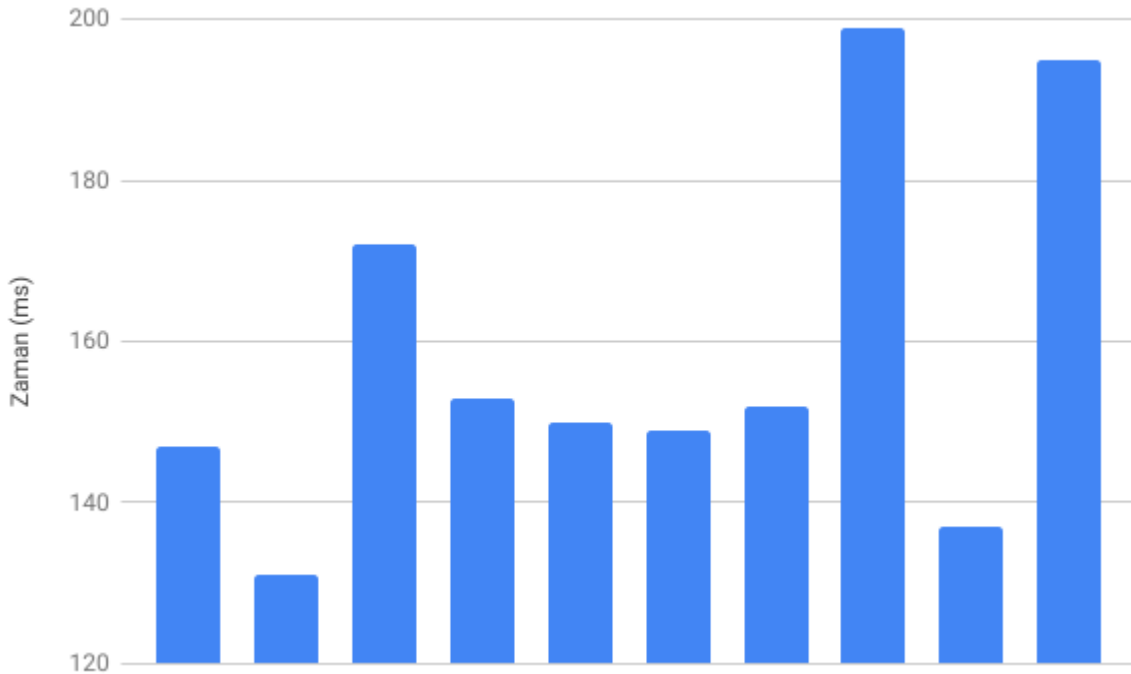
1-)

TCP

Bu projede TCP ile relay server üzerinden dosya transferi için Java kullandım, server tarafında client'lar için multithread bir yapı kullandım ve dosya gönderiminin 1024 byte'lık paketler ile yaptım.

Implementasyon aşamasında ise bazı önemli kararlar verdim. Tüm clientlar yani gönderici ve alıcı server'a bağlanmadığı zaman ikisini de beklettim, ikisi de bağlantıktan sonra gönderici server'a dosyayı göndermeye ve server da gelen bilgiyi alıcıya iletmeye başladı, dosya aktarımının tamamlandığını anlamak için gönderici tarafında alıcıya ilk başta dosya boyutunu ilettim ve son olarak dosya aktarımı tamamlandığında gönderici tarafından oluşturulan dosya hash'i alıcıya iletili bu sayede dosyanın doğru alınıp alınmadığı da kontrol edilmiş oldu

Performans testleri için projenin ana klasöründe bulunan 3.5Mb'lık "deneme.txt" dosyasını kullandım. 10 farklı deneme ile ilgili sonuçları aşağıdaki grafikte görebiliriz. Performans'ı ölçmek için server, receiver ve sender classlarını çalıştırdıktan sonra receiver class'ının çıktısında en altta "Total download time" satırında süre yazmaktadır. Alıcının dosyayı indirme süresini performans ölçütü olarak aldım ve dosya 10 deneme de karşı tarafa %100 doğruluk ile iletili

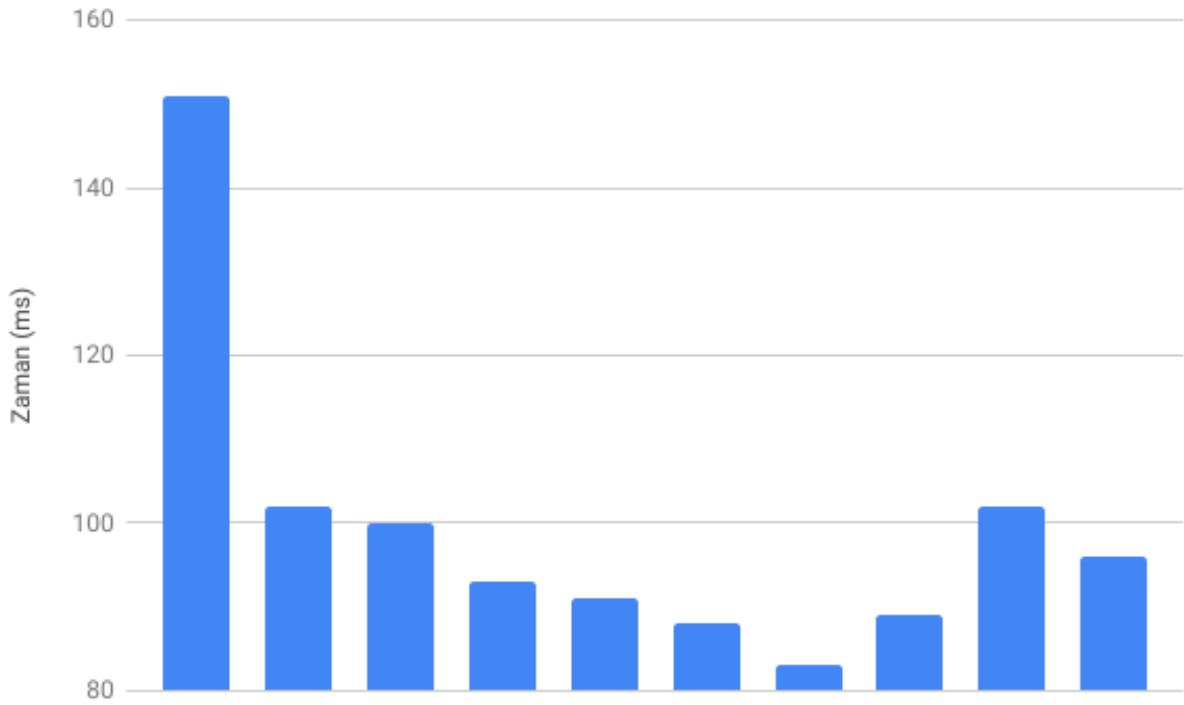


UDP

Bu projede UDP ile relay server üzerinden dosya transferi için Java kullandım, server tarafında client'lar için iki farklı port açtım, clientlar'ın bağlandığını ve dosya transferini başlatmaları için küçük bir handshake süreci tasarladım ve dosya gönderiminin 1024 byte'lık paketler ile yaptım.

Implementasyon aşamasında ise tüm clientlar yani gönderici ve alıcı server'a bağlanmadığı zaman ikisini de beklettim, ikisi de bağlantıktan sonra gönderici server'a dosyayı göndermeye ve server da gelen bilgiyi alıcıya iletmeye başladı, gönderici dosyayı gönderdikten sonra dosyanın bittiğine dair server'a bilgi verdi bu sayede server dosya için veri almayı ve alıcıya veri aktarmayı bıraktı ve dosya aktarımı başlamadan önce gönderici tarafından oluşturulan dosya hash'i alıcıya iletildi bu sayede dosyanın doğru alınıp alınmadığı da kontrol edilmiş oldu

Performans testleri için projenin ana klasöründe bulunan 3.5Mb'lık "deneme.txt" dosyasını kullandım. 10 farklı deneme ile ilgili sonuçları aşağıdaki grafikte görebiliriz. Performans'ı ölçmek için server, receiver ve sender classlarını çalıştırdıktan sonra receiver class'ının çıktısında en altta "Total download time" satırında süre yazmaktadır. Alıcının dosyayı indirme süresini performans ölçütü olarak aldım. Dosya 10 deneme de karşı tarafa dosya tam olarak iletilemedi. Grafikte görüldüğü gibi UDP, TCP ile karşılaştırıldığında dosya transferini daha kısa sürede yapmıştır ancak güvenilirlik olarak TCP ile karşılaştırıldığında çok geride kalmıştır çünkü hiçbir zaman dosya tam olarak aktarılamamıştır.



2 ve 3)

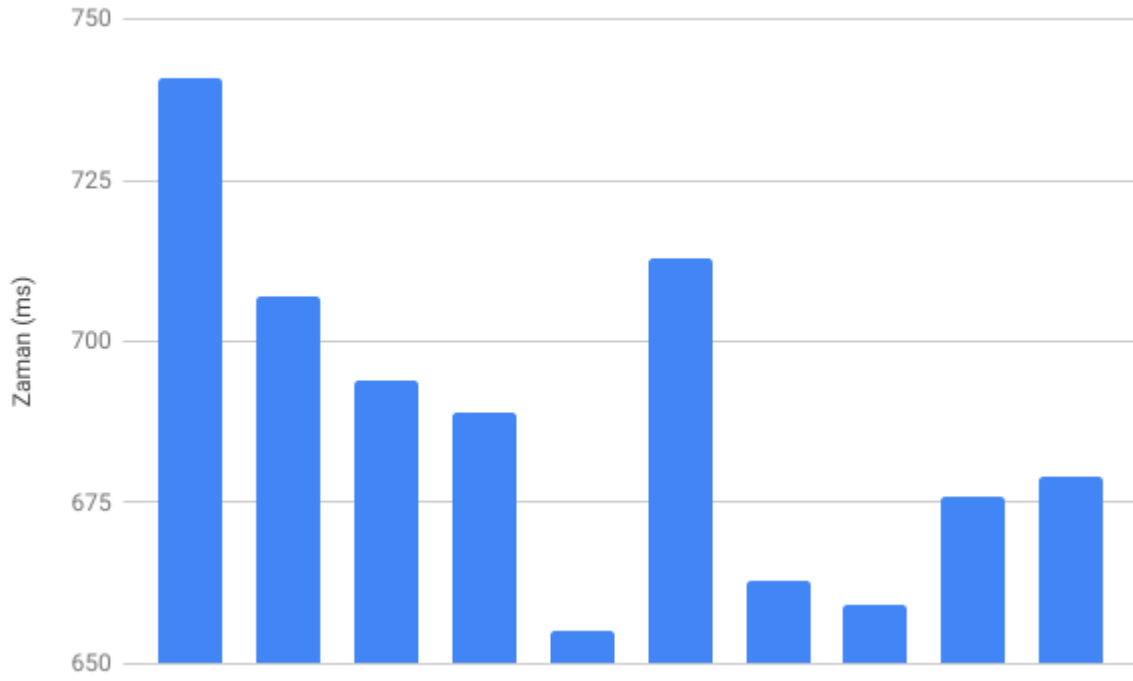
UDP STOP & WAIT

Bu projede UDP stop & wait ile relay server üzerinden dosya transferi için Java kullandım. Server tarafında client'lar için iki farklı port açtım, clientlar'ın bağlandığını ve dosya transferini başlatmaları için küçük bir handshake süreci tasarladım. Dosya gönderiminin 1024 byte'lık paketler ile yaptım. Gönderdiğim veride 962 byte dosyaya ait 32 byte checksum için 7 byte splitter (verileri ayırmak için özel bir string) 16 byte sequence number yine 7 byte splitter. Checksum oluştururken md5 kullandım.

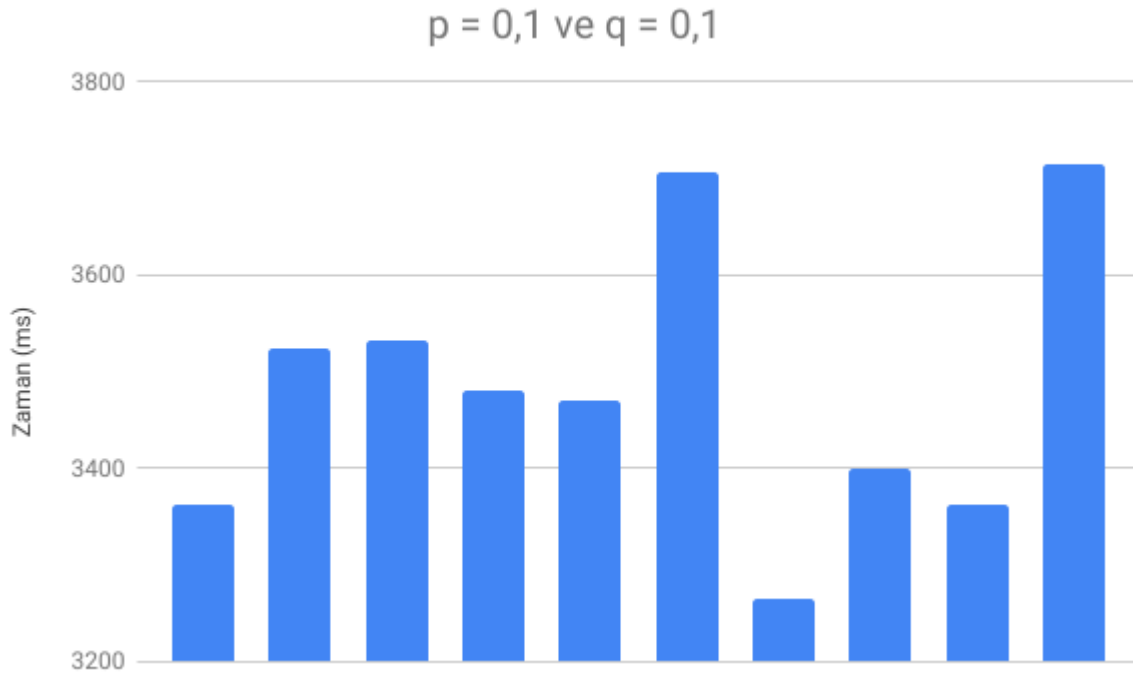
Burada aynı zamanda 3. soruda istenen davranışı simule eden programı da yazdım. UDPStopAndWaitServer.java dosyasının 14. satırında yer alan errorRateFlag eğer true yapılırsa program "pAndQFile.txt" dosyasından p ve q değerlerini okuyup buna göre error davranışını simule etmektedir. Eğer errorRateFlag true olarak set edilirse paket aktarımı normal şekilde yapılmaktadır.

Implementasyon aşamasında ise Tüm clientlar yani gönderici ve alıcı server'a bağlanmadığı zaman ikisini de beklettim, ikisi de bağlantıktan sonra gönderici server'a dosyayı göndermeye ve server da gelen bilgiyi alıcıya iletmeye başladı. Gönderici dosyayı gönderdikten sonra son sequence number olarak bir MAGICNUMBER göndermekte alıcı da bu MAGICNUMBER'ı gördüğü anda dosyanın bittiğini anlamakta. Eğer gönderilen checksum ve aldığım datanın hash'i eşit değilse alıcı tarafında o sequence'in tekrar gönderilmesi istenmekte, aynı zamanda eğer 1 saniye içinde bir veri alamazsak yani bir drop olursa da alıcı o paketin tekrar gönderilmesini istemektedir. Gönderici ise 10 saniye içinde karşı taraftan bir ACK alamazsa eğer aynı paketi tekrar göndermeye devam etmektedir.

Performans testleri için projenin ana klasöründe bulunan 3.5Mb'lık "deneme.txt" dosyasını kullandım. 10 farklı deneme ile ilgili sonuçları aşağıdaki grafikte görebiliriz. Performans'ı ölçmek için server, receiver ve sender classlarını çalıştırdıktan sonra receiver class'ının çıktısında en altta "Total download time" satırında süre yazmaktadır. Alıcının dosyayı indirme süresini performans ölçütü olarak aldım. Dosya 10 deneme de karşı tarafa %100 doğruluk ile iletildi



Görüldüğü gibi süre olarak TCP ve UDP'nin çok gerisinde kalmıştır.



Burada da errorRateFlag true yapılarak dosyadan p ve q değerleri okunmuştur. Bu şekilde de görüldüğü gibi dosyanın aktarım performansı hepsinden daha düşük seviyelerde olmuştur. TCP'nin ve UDP'nin ms'ler içerisinde aktardığı dosya bu şekilde ancak saniyeler içerisinde aktarılabilmektedir.