

**Yıldız Teknik Üniversitesi**

**Bilgisayar Mühendisliği Bölümü – BLM3051 Veri İletişimi ve Bilgisayar Ağları Projesi**



**Dersin Yürütücüsü:** Dr. Öğretim Üyesi Furkan Çakmak

**Proje Üyesi Öğrenciler:**

**GRUP 24**

1-Salih Emre Karakaş 23011624

2-Anılcan Muşmul 23011622

3-Ahmet Furkan Çınarka 23011053

4-Onur Ergüt 22011018

**Data Link Layer Simülasyonu**

## **İçindekiler**

<b>Bölüm</b>	<b>Başlık</b>
--------------	---------------

<b>*</b>	<b>Proje Başlığı</b>
----------	----------------------

<b>*</b>	<b>İçindekiler</b>
----------	--------------------

<b>1</b>	<b>Giriş</b>
----------	--------------

<b>2</b>	<b>Sisteme Genel Bakış</b>
----------	----------------------------

<b>3</b>	<b>Uygulama Detayları</b>
----------	---------------------------

<b>4</b>	<b>İş Bölümü</b>
----------	------------------

<b>5</b>	<b>Test Süreci</b>
----------	--------------------

<b>7</b>	<b>Programdan Ekran Görüntüleri &amp; Teknik Detaylar</b>
----------	---

<b>8</b>	<b>Sonuç</b>
----------	--------------

<b>9</b>	<b>Kaynakça</b>
----------	-----------------

## 1-Giriş

### Projenin Amacı

Bu proje, OSI Referans Modeli'nin ikinci katmanı olan **Veri Bağlantı Katmanı** (Data Link Layer) işleyişini, temel görevlerinin basitleştirilmiş bir simülasyon ortamında kullanarak simüle etmek amacıyla tasarlanmıştır.

Yazılım, kullanıcıların ham veri içeren bir .dat dosyası seçmesine olanak tanır, içeriği 100-bit'lik çerçevelere böler, **CRC-16 hata denetimi** uygular ve bir **Gönderici–Alıcı** modeli üzerinden basit bir **Stop-and-Wait iletişim protokolünü** simüle eder.

Buna ek olarak, program her bir çerçevenin (ulaştı, bozuldu, kayboldu vb.) sonucunu görsel olarak **GUI** üzerinde gösterir ve tüm veriler gönderildikten sonra bir **checksum** hesaplayıp iletir.

Simülasyonda aşağıda verilen hata senaryoları da yer alır:

%10 ihtimalle bir çerçeve gönderim sırasında kaybolur

%20 ihtimalle bir çerçeve bozulur

%15 ihtimalle gönderilen onay mesajı (ACK) düşer

%5 ihtimalle son checksum çerçevesi hatalı olur

Bu proje, öğrencilerin Veri Bağlantı Katmanı'nda verinin nasıl çerçevelendiğini, kontrol edildiğini, gönderilip onaylandığını ve iletişim sırasında oluşabilecek hataların nasıl fark edilip yönetildiğini kavramasını sağlar.

### 2. Sisteme Genel Bakış

Bu proje, Veri Bağlantı Katmanı'nın görevlerini yansıtan modüler parçalara bölünmüştür. Hem arka plan işlemleri hem de grafiksel kullanıcı arayüzü (GUI) sağlamak amacıyla **C++** dili ve **Qt framework** kullanılarak geliştirilmiştir.

Sistemin temel modülleri şunlardır:

**Çerçeve (Frame) Modülü:** 100-bit veri çerçevelerinin oluşturulmasını ve tamponlanmasını kontrol eder; her çerçeveye 16-bit'lik CRC ekler.

**CRC Modülü:** Belirli bir polinom temelli olarak CRC-16 hata denetimini uygular. Tüm çerçeveler bu modülden geçerek CRC hesaplaması ya da doğrulaması yapar.

**Veri Bağlantı Katmanı Modülü:** Stop-and-Wait protokolü ile iletim sürecini yönetilir. Belirtilen olasılıklara göre veri kaybı, bozulma, ACK kaybı ve checksum hatalarını simüle eder.

**Checksum Modülü:** Tüm çerçevelerin CRC'lerini birleştirerek toplam checksum'u hesaplar ve bu değeri, akışın sonunu doğrulamak için ayrı bir çerçeve olarak gönderir.

**GUI Modülü:** Gönderici ve alıcıyı grafiksel olarak temsil eder, her bir çerçevenin durumunu (gönderildi, kayboldu, bozuldu, onaylandı) izler ve kullanıcıya dosya seçimi, gönderim başlatma/durdurma gibi etkileşimler sunar.

Tasarım; modülerlik, sadelik ve simülasyon doğruluğu öncelenerek, Veri Bağlantı Katmanı'nın temel işlevlerinin gerçekçi bir şekilde canlandırılmasını amaçlamaktadır.

### 3-Uygulama Detayları

#### Bit Ayrıştırma ve Çerçeveleme

Hammaddesi byte tabanlı olan .dat dosyası bit dizisine dönüştürülür ve 100-bit'lik çerçevelere bölünür. Sonda kalan segment 100 bit olmasa dahi, geçerli bir çerçeve olarak kabul edilir.

#### CRC-16 Hata Denetimi

Her çerçeve, şu polinom ile çalışan CRC-16 algoritmasından geçirilir:

$x^{16} + x^{15} + x^2 + 1$  (ikili: 1100000000000101, onaltılık: 0x18005).

Bu işlem sonucunda elde edilen 16-bit'lik CRC değeri çerçevenin sonuna eklenir. Alıcı taraf aynı algoritmayı kullanarak çerçevenin bütünlüğünü doğrular.

#### Stop-and-Wait Protokolü

Sistem, veri akışını yönetmek için Stop-and-Wait protokolünü kullanır. Gönderici her seferinde yalnızca bir çerçeve gönderir ve onay (ACK) alana kadar bekler.

Gerçekçi davranışı taklit etmek amacıyla şu hatalar rastgele şekilde simüle edilir:

%10 çerçeve kaybı

%20 çerçeve bozulması

%15 ACK kaybı

%5 son checksum çerçevesinde bozulma

Bu oranlar, rastgele sayı üretimi yoluyla sağlanır.

#### Checksum Üretimi

Tüm çerçeveler gönderildikten sonra, bu çerçevelere ait CRC değerleri birleştirilerek nihai bir checksum oluşturulur. Bu checksum, iletimin sonunda ayrı bir doğrulama çerçevesi olarak gönderilir.

#### Grafiksel Kullanıcı Arayüzü (GUI)

Qt kullanılarak geliştirilen GUI, kullanıcıya şu olanakları sunar:

Girdi dosyasını seçmek, gönderim başlatmak ve durdurmak çerçeve durumlarını (gönderildi, kayboldu, bozuldu vb.) anlık olarak izlemek, iletim günlüklerini takip etmek. GUI, çerçeve ayrımını kolaylaştırmak için renk kodlaması ve metin tabanlı loglama tekniklerini kullanır.

#### 4- İş Bölümü

Öğrenci İsmi	Öğrenci Numarası	Görevler
Salih Emre Karakaş	23011624	GUI arayüzünü tasarladı, kullanıcı etkileşimlerini arka plan işlemleriyle bağdaştırdı ve son test ile raporlama sürecinde yer aldı.
Ahmet Furkan Çınarka	23011053	Temel iletim mantığını geliştirdi; Stop-and-Wait protokolü, hata kontrolü ve sistem entegrasyonunu gerçekleştirdi.
Onur Ergüt	22011018	Genel yapıyı tasarladı, modül görevlerini tanımladı ve Frame ile CRC mantığının uygulanmasına katkı sağladı.
Anılcan Muşmul	23011622	Dosya giriş analizini yönetti, bit düzeyinde çerçeve oluşturmayı gerçekleştirdi ve checksum işlemleri ile sınır durumlarının doğrulanmasına destek oldu.

#### 5- Test Süreci

Sistem, farklı boyut ve içeriklerdeki çeşitli .dat dosyaları kullanılarak test edilmiştir. Her testte programın aşağıdaki durumları nasıl yönettiği gözlemlenmiştir:

Çerçeve oluşturma ve CRC üretimi, belirlenen olasılıklar doğrultusunda hata emülasyonu, Stop-and-Wait protokolü ile yeniden gönderim mantığı, nihai checksum hesaplama ve doğrulama, arayüz güncellemeleri ve gerçek zamanlı görselleştirme.

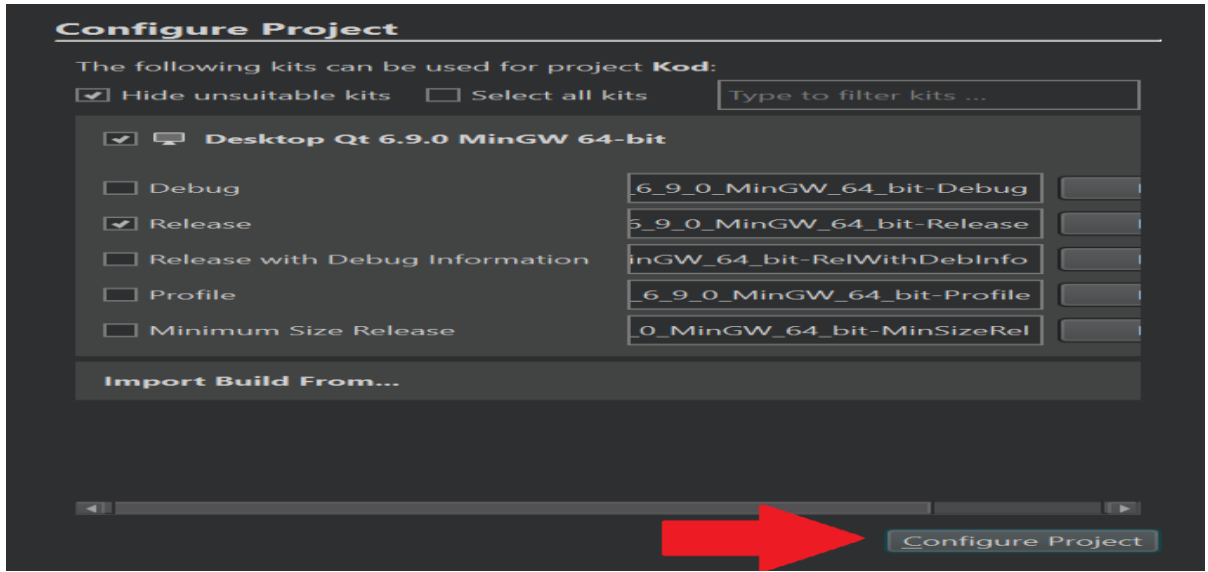
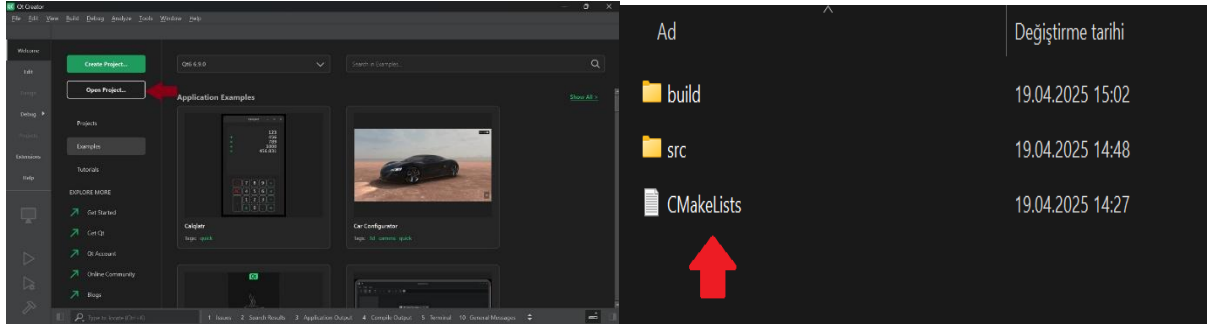
Özellikle dikkat edilen durumlar şunlardır:

Kenar durumlar (örneğin, 100 bit'e tam bölünmeyen dosyalar), bozulmuş çerçevelerin tespiti, geciken veya kaybolan onay (ACK) mesajları, kayıtların ve iletim durumlarının doğru şekilde gösterilmesi.

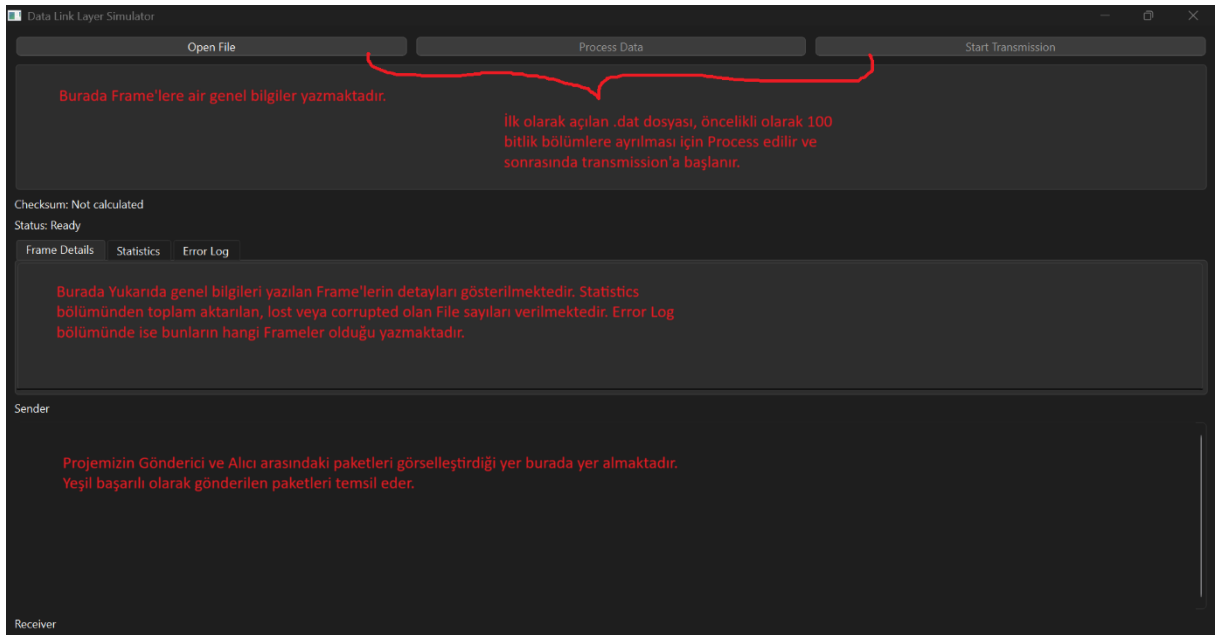
Simülasyon sırasında ekran görüntüleri de alınarak, başarılı şekilde iletilen, kaybolan ve bozulmuş çerçevelerin görsel çıktıları test edilmiştir.

## 6-Programdan Ekran Görüntüleri & Teknik Detaylar

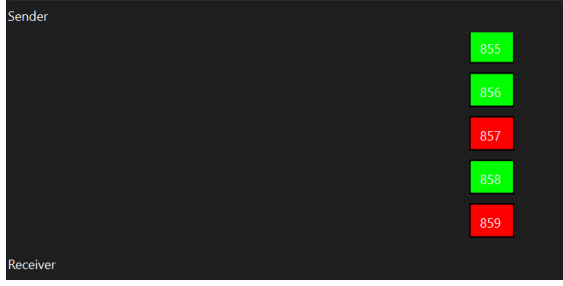
Programımız Qt Creator Community 6.9.0 ile compile edilmiştir.



Daha sonrasında Build edilen projemiz şekildeki gibi bir arayüze sahiptir:



Görselleştirme örneği:



Frame detay örneği:

```
Frame 0: Data: 100 bits, CRC: 10110100110011, Valid: Yes
Frame 1: Data: 100 bits, CRC: 0000011101001110, Valid: Yes
Frame 2: Data: 100 bits, CRC: 1000010010000101, Valid: Yes
Frame 3: Data: 100 bits, CRC: 1011011111000100, Valid: Yes
Frame 4: Data: 100 bits, CRC: 1101111010101001, Valid: Yes
Frame 5: Data: 100 bits, CRC: 0110101010010110, Valid: No, Errors: 1
Frame 6: Data: 100 bits, CRC: 1001100110001111, Valid: Yes

Checksum: 0x09F5
Transmission complete

Frame Details | Statistics | Error Log
Frame Number: 3
Bit Count: 100
CRC: B7C4
Status: Valid
Type: Full Frame

Data:
Hex: A5 07 26 56 16 D6 26 C6 50 A5 76 86 50
Binary: 10100101.00000111.00100110.01010110.00010110.11010110.00100110.11000110.01010000.10100101.01110110.10000110.01010000
```

Projemize ayrıca aşağıdaki gibi de erişilebilmektedir:

Ad	Değiştirme tarihi	Tür	Boyut
platforms	19.04.2025 15:02	Dosya klasörü	
DataLinkLayer	19.04.2025 12:53	Uygulama	238 KB
libgcc_s_seh-1.dll	25.05.2023 00:06	Uygulama uzantısı	107 KB
libstdc++-6.dll	25.05.2023 00:06	Uygulama uzantısı	2.191 KB
libwinpthread-1.dll	25.05.2023 00:06	Uygulama uzantısı	52 KB
Qt6Core.dll	28.03.2025 15:13	Uygulama uzantısı	10.502 KB
Qt6Gui.dll	28.03.2025 15:13	Uygulama uzantısı	10.948 KB
Qt6Widgets.dll	28.03.2025 15:13	Uygulama uzantısı	6.889 KB

## 7-Sonuç

Bu proje, Veri Bağlantı Katmanı'nın temel ilkelerini, çerçeve tabanlı iletim, hata tespiti ve akış kontrolünü içeren basit ama işlevsel bir uygulamayla başarıyla ortaya koymuştur.

CRC-16 kullanımı, Stop-and-Wait protokolü ve iletim hatalarının simülasyonu sayesinde sistem, iki bağlı düğüm arasında gerçekçi veri aktarım senaryolarını canlandırabilmiştir. Grafiksel arayüz, her bir çerçevenin durumunu anlık ve sezgisel biçimde göstererek,

bağlantı katmanı protokollerinin güvenilir olmayan ortamlarda nasıl çalıştığını anlamaya katkı sağlamıştır.

Geliştirme süreci boyunca modülerlik, anlaşılabilirlik ve hata toleransı ön planda tutulmuştur. Mevcut sistem eğitim amaçlı simülasyonlar için etkili olsa da, gelecekte şu iyileştirmeler yapılabilir:

Daha büyük dosyalar için pencere tabanlı protokollerin (örneğin Sliding Window) eklenmesi, hata oranlarının kullanıcı tarafından ayarlanabilir hâle getirilmesi, ayrıntılı iletim raporlarının kaydedilmesi, arayüzün görsel animasyonlar veya diyagramlarla zenginleştirilmesi.

Genel olarak bu proje, iletim hatalarının var olduğu ortamlarda bile veri bağlantı düzeyinde güvenilir iletişimin nasıl sağlanabileceğine dair değerli bir kavrayış sunmuştur.

## 8-Kaynakça

No	Kaynak	Kullanım Amacı
1	Tanenbaum, A. S., & Wetherall, D. J. (2010). <i>Computer Networks</i> (5. Baskı). Pearson.	Veri Bağlantı Katmanı'nın görevlerini ve genel ağ yapısını anlamak için kullanıldı.
2	Stallings, W. (2007). <i>Data and Computer Communications</i> (8. Baskı). Prentice Hall.	Çerçeveleme, hata denetimi ve Stop-and-Wait protokolü gibi kavramlar için başvuruldu.
3	ISO/IEC 7498-1:1994 – <i>OSI Referans Modeli</i> .	OSI katmanlarının, özellikle Katman 2'nin tanımı için kullanıldı.
4	Qt Documentation. <i>Signals and Slots in Qt</i> . <a href="https://doc.qt.io">https://doc.qt.io</a>	Qt GUI tasarımında olay tabanlı yapı ve sinyal-slot mekanizmasının uygulanması için kullanıldı.
5	TutorialsPoint. <i>Stop-and-Wait ARQ</i> . <a href="https://www.tutorialspoint.com">https://www.tutorialspoint.com</a>	Stop-and-Wait protokolünün temel mantığını kavramak ve uygulamak için referans alındı.
6	GeeksForGeeks. <i>Cyclic Redundancy Check (CRC)</i> . <a href="https://www.geeksforgeeks.org">https://www.geeksforgeeks.org</a>	CRC-16 hata denetimi algoritmasının temelini uygulamak için kullanıldı.