



Geleceğin otomotivleri için hangi sayısal veri yolu? Which digital data bus for the automotive of the future?

Yasemin Yılmazoğlu^{1,*} , Ufuk Sakarya² 

¹ Yıldız Teknik Üniversitesi Elektronik ve Haberleşme Mühendisliği Bölümü, 34220, İstanbul Türkiye

Öz

Günümüzde yükselen yapay zekâ eğilimi ile otonom sürüşlü otomobillerin önemi giderek artmaktadır. Otonom sürüş yeteneğinin kazanılmasında önemli alt parçalar bulunmaktadır. Bu alt parçalardan birisi otomobilde yer alan gelişmiş sensör verilerinin yapay zekâ işlemcilerine gerçek zamanlı ve güvenilir bir şekilde aktarılmasıdır. Bu tip bir çözüme gitmeden önce yapılması gereken ilk iş ise mevcut durumun bu amaç doğrultusunda analiz edilmesidir. Bu derleme makalesi bu amaç ile oluşturulmuştur. Günümüzde otomotivde kullanılan belli başlı haberleşme veri yolları karşılaştırmalı olarak analiz edilerek sunulmuştur.

Anahtar kelimeler: Sayısal veri yolu, Araç içi haberleşme, Derleme

1. Giriş

İnsanların en önemli ihtiyaçlarından birisi hiç şüphesiz ulaşımıdır. Geçmişten günümüze ulaşım için çözümler birçok bilim adamının ana uğraşısı olmuştur. Günümüzde yükselen yapay zekâ eğilimi ile otonom sürüşlü otomobillerin önemi giderek artmaktadır. Otonom sürüş yeteneğinin kazanılmasında önemli alt bileşenler bulunmaktadır. Bu alt parçalardan birisi otomobilde yer alan gelişmiş sensör verilerinin yapay zekâ işlemcilerine gerçek zamanlı ve güvenilir bir şekilde aktarılmasıdır. Başka bir deyiş ile bu iş için en uygun veri yolunun tasarlanmasıdır. Bu tip bir çözüme gitmeden önce yapılması gereken ilk iş ise mevcut durumun bu amaç doğrultusunda analiz edilmesidir. Bu derleme makalesi bu amaç ile oluşturulmuştur. Günümüzde otomotivde kullanılan belli başlı haberleşme veri yolları karşılaştırmalı olarak analiz edilerek sunulmuştur. Bu çalışmaların sonucundan her bir yöntemin artı ve eksilerine göre sistem isterlerinin tasarlanması bu makalenin kapsamı dışındadır. Bununla birlikte geleceğin akıllı otomobillerinin haberleşme veri yollarının tasarımında bir adımın atılması ve ülkemizde bu alanın öneminin vurgulanarak farkındalığın artırılması amaçlanmıştır.

1.1 Otomotiv veri yollarının tarihsel gelişimi

Otomotiv endüstrisinde elektronik sistemlerin artan önemiyle birlikte, araç içi iletişim protokollerinin gelişimi büyük bir ivme kazanmıştır. Bu süreçte, farklı veri yolu teknolojileri, araçlardaki elektronik kontrol üniteleri arasında veri alışverişini sağlamak için geliştirilmiştir.

Abstract

Today, with the rising artificial intelligence trend, the importance of autonomous driving cars is increasing. There are important sub-components in gaining autonomous driving ability. One of these sub-components is the real-time and reliable transfer of advanced sensor data in the car to artificial intelligence processors. The first thing to do before going to such a solution is to analyze the current situation in line with this purpose. This review article was created for this purpose. Today, certain communication data paths used in the automotive industry are comparatively analyzed and presented.

Keywords: Digital data bus, Intra-communication in car, Survey

Otomotiv elektroniği 1950'li yıllarda, marş motorları, ateşleme sistemleri ve temel elektrik aksamaları ile sınırlıydı. Elektrikli camlar, merkezi kilit ve klima gibi sistemlerin eklenmesiyle, 1970'li yıllarda araç içi kablolama giderek karmaşık hale geldi. Bu dönemde her elektronik bileşen, doğrudan bir elektrik bağlantısı ile yönetiliyordu. Ancak bu yaklaşım, ağırlık artışı ve maliyet nedeniyle sürdürülemez hale gelmiştir. 1980'lere gelindiğinde, araçlarda kullanılan elektronik sistemlerin sayısındaki artış, daha verimli veri iletim yöntemlerine olan ihtiyacı doğurdu. Bu dönemde, seri veri yolu sistemleri geliştirilmeye başlandı.

Bosch tarafından 1986 yılında geliştirilen CAN protokolü, otomotiv endüstrisinde devrim niteliğinde bir yenilik sundu. ECU'lar arasında doğrudan kablo bağlantısı yerine ortak bir veri yolu üzerinden haberleşme sağlayarak, kablolama karmaşıklığını ve maliyetleri düşürdü. CAN, 1991 yılında Mercedes-Benz tarafından ilk defa seri üretim bir araçta (S-Class W140) kullanıldı ve kısa sürede otomotiv endüstrisinin standardı haline geldi. CAN protokolünün yaygınlaşmasının ardından, belirli ihtiyaçlara yönelik farklı veri yolları geliştirilmeye başlandı [1].

1999 yılında, otomotiv üreticileri düşük maliyetli sistemlerde kullanılmak üzere LIN protokolünü geliştirdi. LIN, özellikle düşük veri hızları gerektiren uygulamalarda kullanıldı [2]. 1998 yılında ise, otomotiv multimedya sistemleri için MOST veri yolu geliştirildi. MOST, yüksek hızlı veri iletimi ve ses/görüntü senkronizasyonu sağlamak amacıyla kullanıldı [3].

Gelişen otomotiv elektroniği ve otonom sürüş teknolojileri, daha yüksek hız ve bant genişliği

* Sorumlu yazar / Corresponding author, e-posta / e-mail: yasemin.yilmazoglu@std.yildiz.edu.tr (Y. Yılmazoğlu)

Geliş / Received: 19.04.2024 Kabul / Accepted: 25.02.2025 Yayınlanma / Published: xx.xx.20xx

doi: 10.28948/ngumuh.1588278

gereksinimlerini doğurdu ve 2006 yılında BMW, FlexRay protokolünü ilk kez seri üretim bir araçta (BMW X5 E70) kullandı. FlexRay, zaman tetiklemeli (time-triggered) iletişim sağlayarak, özellikle otonom sürüş ve güvenlik sistemleri için kritik bir çözüm sundu [4]. Sonrasında Ethernet teknolojisinin otomotiv sektörüne adapte edilmesiyle, BroadR-Reach Automotive Ethernet 2008 yılında tanıtıldı. Bu protokol, özellikle ADAS ve otonom sürüş sistemleri için geliştirildi [5].

Günümüzde ise otomotiv veri yolları, V2X (Vehicle-to-Everything) iletişimi, otonom sürüş sistemleri ve gelişmiş güvenlik çözümleri için sürekli gelişmektedir. V2X, araçların altyapı, diğer araçlar ve yayalarla haberleşmesini sağlamak için LTE ve 5G teknolojilerini kullanan bir teknolojidir [6]. Makalede araç içi haberleşme sistemleri tanıtıldığından V2X ile ilgili detay bilgiye yer verilmemiştir.

2. Otomotivde kullanılan haberleşme veri yolları

2.1. CAN veri yolu

CAN Bus (Controller Area Network), ilk olarak Bosch tarafından 1980'lerde geliştirilen, araç içi elektronik kontrol üniteleri (ECU'lar) arasında iletişimi sağlayan bir seri iletişim protokolüdür. CAN, düşük maliyetli ve güvenilir bir veri iletim çözümü sunduğu için özellikle otomotiv endüstrisinde yaygın bir şekilde kullanılmaktadır [7]. Bu protokol, araç içi sistemdeki çok sayıda elektrikli bileşenin, örneğin mikrodenetleyiciler, elektronik kontrol üniteleri, sensörler, cihazlar ve aktüatörlerin, tek/çift telli bir veri yolu üzerinden birbirleriyle iletişim kurmasını sağlamak amacıyla tasarlanmıştır. CAN, doğası gereği paketlerini yayımlayarak çalışır, yani CAN veri yoluna bağlı tüm düğümler, tüm paket iletimlerini alabilir. Ayrıca, CAN paketinde bir çerçeve, bir yapı olarak tanımlanır ve ağda bir dizi CAN verisi taşır. Her iletilen CAN çerçevesi için çekişme tanımlayıcı (ID) alanı, paketlerin önceliğini gösterir. ID bit değeri ne kadar düşükse, paketin önceliği o kadar yüksektir. Bu protokol, CAN veri yolu trafiği içindeki çarpışmaları önlemek için tasarlanmıştır [8].

2.1.1 Genel bilgiler

CAN veri yolu, özellikle araçlarda bulunan elektronik sistemlerin birbiriyle sorunsuz bir şekilde iletişim kurmasını sağlar. Bu sistem sayesinde farklı kontrol üniteleri, merkezi bir bilgisayara ihtiyaç duymadan doğrudan veri alışverişi yapabilir. Örneğin, motor kontrol ünitesi, fren sistemi veya hava yastıkları gibi hayati sistemler CAN veri yolu üzerinden birbiriyle iletişim kurarak koordineli bir çalışma sergiler [9]. CAN protokolü, hata tespiti ve veri güvenliği gibi özelliklere sahip olup, otomotiv sektöründe yaygın olarak kullanılır. Ayrıca, protokol çoklu sahip-köle (master-slave) mimarisini destekler ve birden fazla cihazın aynı anda veri iletebilmesine olanak tanır.

2.1.2 Mesaj yapısı

CAN protokolünde her mesajın belirli bir formatı vardır. Bu formatta bir mesaj kimliği (ID), veri uzunluğu kodu (DLC), veri alanı ve döngüsel artıklık denetimi (CRC) yer alır. Her mesaj belirli bir önceliğe sahiptir ve bu öncelik, mesajın ID'si ile belirlenir. Yüksek öncelikli mesajlar, ağda

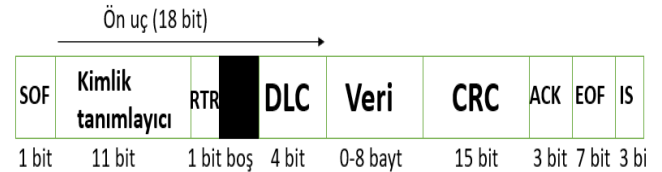
diğer mesajlara göre öncelikli olarak iletilir. CAN'de iki ana mesaj formatı vardır: Standart Çerçeve (11-bit ID) ve Genişletilmiş Çerçeve (29-bit ID)

CAN mesaj yapısındaki alanların açıklamaları Tablo 1'de verilmiştir.

Tablo 1. CAN veri yolu alan adları ve açıklamaları

Alan Adı	Açıklama
ID	Mesajın önceliğini belirleyen kimlik
DLC	Veri uzunluğu kodu
Veri Alanı	İletilen asıl veri
CRC	Hata tespit mekanizması
ACK	Mesajın başarıyla alındığını belirten bayrak

Şekil 1'de gösterildiği gibi, standart bir CAN 2.0A veri çerçevesi yedi alandan oluşur: Çerçeve başlama (SOF) biti, 18 bit başlık, 0-8 bayt veri, 15 bitlik döngüsel artıklık denetimi alanı, 3 bitlik onay yuvası (ACK), 7 bitlik çerçeve bitiş alanı (EOF) ve son olarak 3 bitlik ara çerçeve boşluğu. Ayrıca, bir çerçevenin başlığı üç küçük alana bölünebilir: 11 bit tanımlayıcı alanı (CAN 2.0B için 29 bit, genişletilmiş format), uzaktan iletim isteği (RTR) biti ve 4 bitlik veri uzunluk kodu. Bir CAN çerçevesi en fazla 8 bayt veri içerebilir. Tanımlayıcı kısmı (genişletilmiş CAN çerçeve formatı için 11 bit veya 29 bit), veri yoluna erişim için yapılan çekişme sırasında mesaj önceliklerini belirler. CAN çekişme kontrolü, veri yolu üzerindeki iletim sırasında çerçeve çarpışmalarını önlemek için çarpışma çözünürlüğü ile taşıyıcı algılama çoklu erişimi (CSMA/CR) mekanizmasına dayanır. Bu noktada, her mesaj için benzersiz olan ve bir CAN çerçevesinin başlığına ait olan tanımlayıcı alan, mesajın önceliğini belirler [10].



Şekil 1. CAN 2.0A mesaj formatı (8 baytlık veri işlemesi)

2.1.3. Daha hızlı CAN: CAN FD

CAN FD (Esnek Veri Hızlı) veri yolu, Robert Bosch GmbH tarafından geliştirilen ve ISO standart spesifikasyonuna dahil edilen, daha fazla veri taşımak ve daha yüksek hızlarda iletim yapmak için klasik CAN protokolünün bir uzantısıdır.

Klasik CAN çerçeveleri ile karşılaştırıldığında başlıca farklar şunlardır:

- Çerçeve başlığından bir yedek bit, klasik CAN ile CAN FD çerçevelerini ayırt etmek için kullanılır.
- Standart veri birimi (SDU), başlık ve fragmanın veri hızına göre daha yüksek bir hızda iletilir (15 Mbps'ye kadar) ve 0-64 bayt arasında boyutları destekler, klasik CAN'e kıyasla 8 kat daha fazla veri taşır.

- SDU uzunluğu yalnızca 4 bit ile belirtilir, bu nedenle izin verilen uzunluklar 0, 1, ..., 8 bayt ya da daha uzun yükler için 12, 16, 20, 24, 32, 48, 64 bayttır [9].

Şekil 2’de CAN FD’nin klasik mesaj çerçeve yapısı verilmiştir.

CAN Kimlik tanımlayıcı (11 ya da 29 bit)	SDU uzunluğu (4 bit)	SDU (0-64 bayt)	CRC (17 ya da 21 bit)
--	-------------------------	--------------------	--------------------------

Şekil 2. CAN FD mesaj çerçeve yapısı

CAN ve CAN FD arasındaki farklar Tablo 2’de verilmiştir.

Tablo 2. CAN ve CAN FD arasındaki teknik farklar

Özellik	CAN	CAN FD
Maksimum Veri Hızı	1 Mbps	8 Mbps
Maksimum Veri Alanı	8 bayt	64 bayt
Esneklik (Flexible Data-Rate)	Sabit bit hızı	Veri ve ACK bölgesinde değişken hız
Hata Algılama ve Düzeltme	CRC-15 ve CRC-16	Güçlü CRC-17 ve CRC-21
Geriye Dönük Uyumluluk	Tüm CAN ağlarıyla uyumlu	Klasik CAN ile kısmen uyumlu
Güvenlik ve Güvenilirlik	Daha düşük hata toleransı	Daha yüksek güvenlik mekanizmaları
Tipik Kullanım Alanı	Motor kontrolü, güvenlik sistemleri, klima kontrolü	ADAS, otonom sürüş, yüksek bant genişlikli sistemler

Düşük hız gerektiren ve maliyetin önemli olduğu uygulamalarda klasik CAN hala kullanılmakta olsa da elektrikli araçlar, otonom sürüş ve ADAS gibi ileri seviye sistemlerde CAN FD’nin kullanımı daha yaygındır

2.1.4. CAN için donanım gereksinimleri

CAN veri yolu sistemlerinin çalışması için özel donanımlar gerekmektedir. Bu donanımlar arasında CAN kontrolörü, alıcı-verici ve fiziksel katman bileşenleri yer alır.

CAN kontrolörü, veri trafiğini yöneten merkezi bileşen olup, mesajları göndermek ve almakla görevlidir. Alıcı-verici ise, verilerin analog sinyaller şeklinde fiziksel kablo üzerinden iletilmesini sağlar. Fiziksel katman, genellikle iki telli bükülmüş çift kablo (twisted pair) kullanılarak oluşturulur ve elektromanyetik girişimlere karşı dirençli bir yapı sağlar [10]. Tablo 3’de CAN için sağlanması gereken donanım bileşenleri verilmiştir.

Tablo 3. CAN için donanım gereksinimleri

Bileşen	İşlev
CAN Kontrolörü	Mesajları gönderir ve alır
Alıcı-Verici (Transceiver)	Verileri fiziksel katman üzerinden iletir
Fiziksel Katman	Bükülmüş çift kablo üzerinden veri iletimi

2.1.5. CAN mesajları nasıl okunur?

Araç iletişimini daha ayrıntılı anlamak için tüm ECU’lar ve bunlara bağlı CAN mesaj kimliklerinin belirlenmesi gerekmektedir. Bu amaçla CANoe ya da CANalyzer uygulamaları kullanılabilir. CANalyzer ve CANoe araçları, kapsamlı bir ölçüm ve simülasyon yetenekleri setini birleştirerek CAN tabanlı modül veya sistem geliştirmenin temel ihtiyaçlarını karşılamak için geliştirilmiştir. CANalyzer ve CANoe birden fazla CAN ağına arayüz oluşturabilir ve tüm iletişim aktarımları için doğru zaman damgalı ölçümler sağlayabilir [11]. İlk olarak, CANalyzer kullanarak CAN veri yoluna erişim sağlayacak fiziksel bir bağlantı, tercihen referans sistem ECU’suna yakın bir yerde araca monte edilmelidir. CANalyzer’ın "izleme fonksiyonu" ile veri yolu iletişimi ve tüm ECU’ların CAN mesajları kolayca görüntülenebilir (Şekil 3).

CAN ID’lerinin yanı sıra her bir mesajın döngü süresi ve uzunluğu da analiz edilebilir. Bu bilgiler, referans sisteminin doğru çalışmasını sağlamak için tüm katılımcı ECU’ların bir veri yolu simülasyonunda sonradan kullanılmak üzere önemlidir.

ID	Ad	Veri	CAN Frame
62404.932600	CAN 1	18FF012Ex	CAN Frame
62404.932600	CAN 2	18FE4A03x	CAN Frame
62404.933100	CAN 1	18FFC8D1x	CAN Frame
62404.933200	CAN 3	18FF022Ex	CAN Frame
62404.943000	CAN 1	DC1700x	CAN Frame
62404.943300	CAN 1	C00D100x	CAN Frame
62404.933700	CAN 1	18FF022Ex	CAN Frame
62404.943600	CAN 1	18FFC500x	CAN Frame
62404.936600	CAN 1	18FF3821x	CAN Frame
62404.934900	CAN 2	CF02F2Ax	CAN Frame
62404.936900	CAN 2	18FF3821x	CAN Frame
62404.935800	CAN 1	CF02F2Ax	CAN Frame
62404.943500	CAN 2	18F0090Bx	CAN Frame
62404.944100	CAN 1	18F0090Bx	CAN Frame
62404.943800	CAN 2	FF0C0Bx	CAN Frame
62404.936100	CAN 1	CFE6CEEEx	CAN Frame
62404.944100	CAN 2	8FE6E0Bx	CAN Frame
62404.936400	CAN 2	CFE6CEEEx	CAN Frame
62404.925400	CAN 2	18F0010Bx	CAN Frame
62404.926000	CAN 1	18F0010Bx	CAN Frame
62404.937800	CAN 1	C010071x	CAN Frame

Şekil 3. CANalyzer ile CAN mesajlarının okunması ve izlenmesi

```
primary_status_4_S0 - Notepad
File Edit Format View Help
***BUSMASTER Ver 3.2.2***
***PROTOCOL CAN***
***NOTE: PLEASE DO NOT EDIT THIS DOCUMENT***
***[START LOGGING SESSION]***
***START DATE AND TIME 10:10:2024 11:2:34:413***
***HEX***
***SYSTEM MODE***
***START CHANNEL BAUD RATE***
***CHANNEL 1 - Kvaser - Kvaser Leaf Light v2 #0 (Channel 0), Serial Number- 0, Firmware- 0x000003a5 0x00040013 - 500k
***END CHANNEL BAUD RATE***
***START DATABASE FILES***
***C:\Users\yyilmazo\Desktop\dbc_files\RigidBev_VCU1.dbf***
***END DATABASE FILES***
***<Time><Tx/Rx><Channel><CAN ID><Type><DLC><DataBytes>***
11:02:34:4070 Rx 1 0xCF090F3 x 8 50 1F 50 1F 30 34 00 7D
11:02:34:4080 Rx 1 0xCF091F3 x 8 A0 76 82 0E 7A 0E FF FF
11:02:34:4080 Rx 1 0xCF180F3 x 8 C0 76 FF FF 7C 0E FF FF
11:02:34:4080 Rx 1 0x18037C00 x 8 00 00 00 00 00 00 00 00
11:02:34:4080 Rx 1 0xC1BF35B x 8 00 00 00 00 00 00 D0 6F
11:02:34:4090 Rx 1 0xCF092F3 x 8 3D 24 16 24 29 24 FF FF
11:02:34:4090 Rx 1 0xCF1B1F3 x 8 2D 24 16 24 23 24 FF FF
11:02:34:4090 Rx 1 0xCFF01CB x 8 94 03 00 7D FF 7C 00 00
11:02:34:4100 Rx 1 0xCF093F3 x 8 B0 6D FF FF FF FF FF FF
11:02:34:4100 Rx 1 0xCF094F3 x 8 E0 A8 E0 A8 FF FF FF FF
11:02:34:4100 Rx 1 0xCFF02CB x 8 BF 6A 4A A6 34 01 FF FF
11:02:34:4100 Rx 1 0xCFF03CB x 8 B8 6A 04 00 26 62 22 FF
11:02:34:4110 Rx 1 0xCF096F3 x 8 F0 FF CF FF 5F FF FF F3
11:02:34:4110 Rx 1 0x18077C00 x 8 00 00 0A 0F A0 5E 40 55
11:02:34:4110 Rx 1 0x18087C00 x 8 E0 FF 80 20 00 00 00 00
11:02:34:4110 Rx 1 0x18EFF700 x 8 00 00 62 02 B4 91 00 00
11:02:34:4120 Rx 1 0xCF0A0F3 x 8 B0 D9 02 FF FF FF FF FF
11:02:34:4120 Rx 1 0xCF18EF3 x 8 D4 07 D4 07 FF FF 00 7D
```

Şekil 4. CAN mesaj logu

İlgili CAN mesajları belirlendikten sonra, mesaj veri baytlarını ayrıntılı olarak incelemek ve fiziksel sinyalleri tespit etmek kaçınılmazdır. Bu, fiziksel girişler manuel olarak üretilerek (örneğin gaz pedalını açmak, sürüş yapmak, fren yapmak vb.) ve belirli CAN mesajları ile bunların baytlarını paralel olarak gözlemleyerek başarılabılır. Ardından, bir CAN mesajı, CAN verisi ve bir fiziksel giriş değeri arasında bir ilişki kurulabilir [12].

CAN veri yolu kayıtları CANoe dışındaki aplikasyonlarla da sağlanıp izlenebilir. Şekil 4'te farklı bir CAN veri yolu izleme aracı ile alınmış mesaj kayıtlarını içeren bir dosya görüntüsü verilmiştir.

2.1.6. Kullanım alanları

İlk CAN gelişimi, büyük ölçüde otomotiv endüstrisi tarafından desteklenmiştir: CAN, çeşitli binek otomobillerde, kamyonlarda, teknelerde, uzay araçlarında ve diğer araç türlerinde kullanılmaktadır. Protokol günümüzde de endüstriyel otomasyon ve ağ tabanlı gömülü kontrol sistemleri gibi alanlarda yaygın olarak kullanılmaktadır ve üretim makineleri, tıbbi ekipmanlar, bina otomasyonu, dokuma makineleri ve tekerlekli sandalyeler gibi çeşitli ürünlerde uygulamaları bulunmaktadır [7]. Modern araçlarda, motor kontrolü, şanzıman yönetimi, ABS, hava yastıkları, klima sistemleri gibi farklı sistemler CAN ağı üzerinden haberleşir. Bu sistemlerin yanı sıra, CAN veri yolu, endüstriyel otomasyon, medikal cihazlar, demiryolu sistemleri ve havacılık gibi farklı sektörlerde de kullanılmaktadır. Örneğin, elektrikli araçların batarya yönetim sistemlerinde CAN protokolü yaygın olarak kullanılmaktadır.

2.1.7. Otomotivdeki yeri

CAN veri yolu, modern araçların elektronik altyapısının bel kemiğini oluşturur. Araç içerisindeki tüm elektronik

kontrol üniteleri CAN ağı üzerinden birbirleriyle iletişim kurar. Bu da hem maliyetleri düşürür hem de araç içi kablolama karmaşıklığını azaltır. Özellikle hibrit ve elektrikli araçlarda, motor, batarya ve enerji yönetim sistemlerinin entegrasyonu CAN veri yolu ile sağlanmaktadır [8]. Bu nedenle, CAN veri yolu otomotiv endüstrisinde vazgeçilmez bir iletişim protokolü haline gelmiştir.

2.1.8. Avantajlar ve dezavantajlar

CAN veri yolunun en büyük avantajlarından biri, hızlı ve güvenilir bir iletişim sunmasıdır. Ayrıca, düşük maliyetli olması ve hata tespit mekanizmalarının etkinliği, sistemin güvenliğini artırır. Ancak, CAN veri yolu sınırlı bant genişliğine sahiptir ve bu, özellikle yüksek veri gereksinimi olan uygulamalarda dezavantaj yaratabilir. Bunun yanı sıra, güvenlik açıkları, özellikle günümüzde siber saldırılara karşı hassasiyet oluşturabilir. Son yıllarda, bu güvenlik açıklarını gidermek için CAN protokolüne ek güvenlik katmanları eklenmiştir. Tablo 4'te CAN veri yolunun avantaj ve dezavantajları kısaca maddelendirilmiştir.

Tablo 4. CAN veri yolu avantaj ve dezavantajları

Avantajlar	Dezavantajlar
Düşük maliyetli	Sınırlı bant genişliği
Hızlı ve güvenilir iletişim	Güvenlik açıklarına karşı hassasiyet
Hata tespit mekanizmaları etkin	Yüksek veri hızları için yetersiz olabilir

2.1.9. CAN güvenlik sorunları

CAN veri yolu ilk olarak 1980'lerin başında tasarlandığında, güvenlik ana odak noktası değildi. Başlangıçta, sadece birkaç ECU arasında iletişim kurmak

için kullanılan bu sistem, son kullanıcı tarafından erişime açık değildi. Ancak zamanla otomotiv endüstrisi büyük bir değişim geçirdi ve bugün onlarca ECU birbiriyle bağlantılı hale geldi. Yasal düzenlemeler de teşhis amaçlı veri yoluna erişim sağlanmasını gerektirmektedir [10].

CAN protokolü çeşitli güvenlik özelliklerine sahip olmasına rağmen, hala saldırılara açıktır. En büyük sorunlardan biri şifreleme ve kimlik doğrulamanın bulunmamasıdır. Bu eksiklik, yetkisiz düğümlerin ağı katılmasına ve iletişim kurmasına olanak tanır. Verilerin şifrelenmemesi, saldırganların bu verileri kolayca dinleyip anlamasına imkân tanır. Özellikle modern araçların sürücülere ait konum bilgileri ve adres defteri gibi verileri topladığı göz önüne alındığında, bu durum önemli gizlilik sorunlarına yol açabilir. Ayrıca saldırganlar, sisteme hatalı veri enjekte edebilir [13].

Ayrıca CAN protokolü, Hizmet Engelleme (DoS) saldırılarına karşı da savunmasızdır. Arabuluculuk mekanizması, önceliği olan düğümlerin önce veri iletmesine izin verir. Eğer kötü niyetli bir düğüm sürekli olarak en yüksek önceliğe sahip olursa, diğer düğümler veri iletemez. Bu durumda, CAN veri yolundaki iletişim kesintiye uğrar. Bir saldırgan iletişim sırasında kasıtlı olarak bir hata oluşturduğunda, hata sayaçlarının artmasına neden olur ve bu sayede düğüm veri yolundan çıkarılabilir [13].

Ayrıca CAN veri yolu, her bir mesajın tüm ECU'lara iletilmesini sağlayan bir yayın mekanizmasına dayanır. Bu durum, saldırganın bir ECU'ye erişim sağladığında bütün ağı dinleyebilmesine veya bozabilmesine neden olabilmektedir. CAN protokolü, güvenlik açısından değerlendirildiğinde CIA (Gizlilik, Bütünlük ve Erişilebilirlik) üçlemesi temel alınarak analiz edilmektedir. Bu üçleme, herhangi bir sistemin güvenlik açıklarını değerlendirmek için kullanılan temel bir güvenlik modelidir.

- **Gizlilik:** Verilerin yalnızca yetkili kişiler tarafından erişilebilir olmasını sağlamayı hedefler. Normalde, kriptografik yöntemler kullanılarak gizlilik sağlanır. Ancak, CAN protokolünde bu tür yerleşik güvenlik önlemleri bulunmamaktadır.
- **Bütünlük:** Verilerin iletim sırasında değiştirilmediğini garanti eder. CAN, CRC ile veri bütünlüğünü sağlamak için temel bir mekanizmaya sahiptir. Ancak, CRC sadece rastgele hataları algılayabilir, kötü niyetli bir düğüm tarafından yapılan değişiklikleri tespit edemez.
- **Erişilebilirlik:** Ağın her zaman yetkili kullanıcılar tarafından kullanılabilir olmasını ifade eder. CAN protokolünde, düğümler arasında bir önceliklendirme kuralı bulunur. Bu kural, daha yüksek öncelikli düğümün veri iletimine öncelik verir.

Sonuç olarak, CAN protokolü güvenlik üçlemesi kapsamında tam anlamıyla güvenlik sağlamaktan uzaktır. Gizlilik, bütünlük ve erişilebilirlik konularında önemli açıklar taşımaktadır.

2.1.10 CAN güvenlik sorunlarına alınabilecek önlemler

CAN veri yolunun siber güvenlik açıklarına karşılık alınabilecek birtakım önlemler mevcuttur. Aşağıda bu önlemler maddelendirilmiştir.

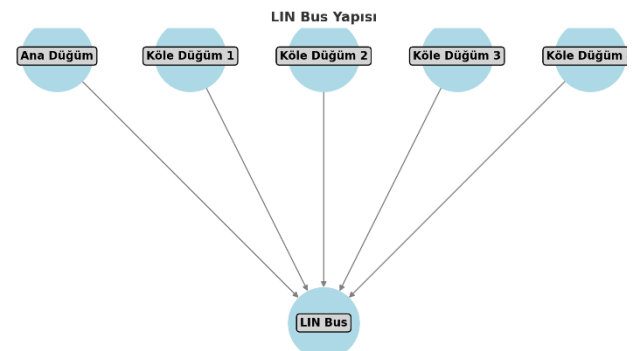
- **Mesaj Kimlik Doğrulama Mekanizmaları:** CAN veri yolu, saldırılara karşı kriptografik doğrulama yöntemleri kullanılmalıdır. Örneğin: Message Authentication Code (MAC) kullanılarak mesajların sahte olup olmadığı kontrol edilebilir. HMAC-SHA256 gibi hash tabanlı doğrulama yöntemleri ile mesajların kaynağı güvence altına alınabilir [14].
- **Anomali Tespiti:** Araç içinde davranış tabanlı anomali tespit sistemleri kullanılabilir. Bu sistemler, olağandışı mesaj yoğunluğu veya kimlik sahtekârlığı gibi saldırıları tespit edebilir [15]. Örneğin CAN saldırı tespit sistemleri (IDS), araç içi iletişimi analiz ederek anormal mesajları engelleyebilir [16].
- **Fiziksel Erişim Kısıtlamaları:** Saldırganların CAN veri yoluna fiziksel erişim sağlamasını önlemek için OBD-II portuna yetkisiz erişimi engelleyen güvenlik çözümleri uygulanmalıdır ve güvenli donanım bileşenleri kullanılarak saldırılara karşı fiziksel koruma sağlanmalıdır [17].

2.2. LIN veri yolu

LIN (Local Interconnect Network), otomotiv sektöründe düşük maliyetli ve düşük hızda iletişim sağlayan bir veri yolu protokolüdür. 1999 yılında tanıtılan bu protokol, CAN veri yolunun daha karmaşık olduğu ve yüksek hız gerektirmeyen uygulamalarda kullanılmak üzere geliştirilmiştir.

2.2.1. Mesaj ve çerçeve yapısı

LIN mesaj yapısı, ana düğüm tarafından başlatılan bir başlık ve ardından gelen bir yanıt mesajından oluşur. Başlık, senkronizasyon ve kimlik bilgilerini içerirken, yanıt mesajı veri yükü ve kontrol bilgilerini içerir. Mesaj yapısında basitlik ön plandadır ve çoğu zaman küçük boyutlu verilerin iletimi için uygundur. Şekil 4'te LIN veri yoluna ait mimari gösterilmiştir.



Şekil 4. LIN veri yolu mimarisi – ana düğüm ve köle düğümler

LIN sisteminde tek bir ana cihaz (master) ve ona bağlı köle cihazlar (slave) bulunur. Ana cihaz hem ana görevleri hem de köle görevlerini yönetirken, köle cihazlar yalnızca köle görevlerini yerine getirir. Veri yolu, fiziksel olarak bir ana ve birden fazla köle düğümden oluşur. LIN protokolündeki tüm iletişim, ana görev tarafından analiz edilir ve bu görevi yalnızca ana düğüm yerine getirir.

LIN veri yolu çerçevesi ise iki bölümden oluşur: başlık (header) ve yanıt (response). Başlık, ana düğüm tarafından gönderilir ve aşağıdaki bileşenlerden oluşur:

- Break: LIN çerçevesinin başlangıcı, CAN protokolündeki çerçeve başlangıcı (SOF) ile benzer şekilde bir "break" sinyali ile yapılır. Break sinyali, 13 dominant bit ve bir resesif bitten (break delimiter) oluşur. Veri yolundaki tüm düğümler, break sinyali ile başlatılır.
- Sync: Senkronizasyon (Sync) alanı, mesaj başlığının ikinci bölümüdür ve ana düğüm tarafından iletilir. Hexadecimal olarak x55 karakteri, köle düğümlerin otomatik aktarım hızını hesaplamasını sağlar. Bu hesaplama köle düğüm tarafından yapılır ve veri yolu ile transfer hızının senkronize edilmesine yardımcı olur [18].
- Identifier: Kimlik (Identifier), başlık bölümünün son alanıdır ve ana düğüm tarafından başlığa iletilir. Kimlik, veriyi alacak olan cihazı tanımlar ve her köle düğüm gelen kimlikleri bekler. LIN ağında toplamda 64 farklı kimlik bulunur. 0-59 arası veri çerçevelerinin iletimi için, 60-61 ise tanımlama verilerinin iletimi için kullanılır. 62 numaralı kimlik genişleme için ayrılmış olup, 63 ise gelecekteki optimizasyonlar için ayrılmıştır.
- Data: Veri alanı, mesajın gerçek verisini içerir ve bu veriler 0 ile 8 bayt arasında olabilir. Yanıt olarak köle düğüm tarafından iletilir.
- Checksum: Kontrol toplamı (Checksum), yanıt bölümünde yer alır ve köle düğüm tarafından iletilir. LIN protokolünde, 8 bitlik kontrol toplamını hesaplayan iki farklı algoritma bulunur [19].

Her LIN mesaj çerçevesi, ana düğüm tarafından gönderilen bir başlık ile başlar. Bu başlık, bir kopma bayt alanı, senkronizasyon bayt alanı ve korumalı kimlik bilgisini içerir. Köleler istenen yanıtı göndermeden önce kısa bir duraklama, yanıt alanı olarak bilinen boşluk oluşur. Her çerçeve sonunda, bir sonraki çerçeve başlatılana kadar LIN veri yolu yüksek seviyede tutulur ve kopma baytı, veri yolunu tekrar düşük seviyeye çeker.

Bu yanıt, sekiz veri bayt alanına ek olarak bir kontrol toplamı bayt alanı içerebilir. LIN 2.0 spesifikasyonunun yayınlanmasıyla birlikte, yeni bir kontrol toplamı hesaplama yöntemi tanıtıldı. Halen kullanılan LIN 1.3 spesifikasyonuna uyum sağlamak amacıyla, önceki kontrol toplamı hesaplama yöntemi de LIN 2.0 tarafından desteklenmektedir. Yeni kontrol toplamı, "gelişmiş kontrol toplamı" olarak adlandırılır ve korumalı kimlik bilgisi dahil tüm veri baytlarının ters çevrilmiş sekiz bitlik toplamı ile hesaplanır. LIN 1.3 kontrol toplamı ise klasik kontrol toplamı olarak adlandırılır ve korumalı kimlik bilgisi olmadan hesaplanır.

Her bir köle, kopma bayt alanını algıladığı anda aktif duruma geçer. Ardından gelen Senkronizasyon bayt alanı, kölelerin bir sonraki iletim için ana düğümle senkronize olmasını sağlar. Senkronizasyon dizisi, her zaman Ox55 veri değeri ile gönderilen bir bayt alanıdır, böylece ağdaki tüm köleler, bu sinyalin kenarlarını algılayarak ana düğümün saatine senkronize olabilir [20].

Kopma bayt alanı hariç, LIN mesajındaki tüm bayt alanları, seri haberleşme tabanlı çoğu iletişimde bilinen düşük seviye başlangıç biti ile başlar, ardından sekiz veri biti gelir ve bayt alanı bir durdurma biti ile tamamlanır. Korumalı kimlik alanı altı kimlik bitinden ve iki parite bitinden oluşur, bu da bir LIN ağında 64 farklı kimlik oluşturulmasını sağlar.

Korumalı kimlik bilgisine dayanarak, köleler yanıt alanından sonra istenen yanıtı göndermeye başlar veya daha fazla veri bekler [18].

3. Kullanım alanları

LIN veri yolu, kullanımı, gecikmenin herhangi bir soruna neden olmadığı kritik olmayan işlemler ve görevlerle sınırlıdır [21]. Uygulama örnekleri arasında ayna ayarı, klima kontrolü, iç ışık gibi basit kontrol görevleri bulunmaktadır. Aşağıda genel kullanım yerleri verilmiştir.

- Otomotiv Sektörü: LIN ağı, araç içi elektronik sistemlerin birbiriyle haberleşmesini sağlar. Düşük hız ve düşük maliyet gereksinimleri nedeniyle araçların kapı kilitleri, aynalar, cam açma/kapama sistemleri, koltuk ayarlama motorları, klima sistemleri gibi alt sistemlerinde kullanılır.
- Araç Aydınlatma Sistemleri: İç ve dış aydınlatma sistemleri, özellikle de far ve iç ışıklandırma gibi donanımlar için uygun maliyetli bir iletişim protokolü sunar.
- Sensör Ağları: Araç içindeki çeşitli sensörler (yağ basıncı, sıcaklık, hız, park sensörleri) için LIN kullanılabilir. Düşük veri aktarım hızına ihtiyaç duyulan ve sürekli iletişim gerektirmeyen yerlerde tercih edilir.
- Otomatik Kapı Sistemleri: Elektrikli kayar kapı, bagaj ve sunroof gibi sistemlerde LIN protokolü kullanılarak maliyet tasarrufu sağlanabilir.
- Ev Otomasyonu ve Beyaz Eşya: Düşük maliyet ve hız avantajı nedeniyle LIN protokolü, ev otomasyon sistemlerinde ve bazı beyaz eşya uygulamalarında kullanılabilir.

LIN ile oluşturulan sistemlerde, LIN veri yolunun kullanımı sadece maliyeti düşürmekle kalmamakta, aynı zamanda temel teknik cihazların operasyonel ve teşhis bilgilerini tam olarak kullanmak için hiyerarşik bir ağ yapısı sağlandığı görülmüştür [22].

4. Avantaj ve dezavantajları

LIN, donanım ve yazılım gereksinimleri açısından CAN gibi daha karmaşık protokollere kıyasla çok daha ucuzdur. Bu nedenle, düşük maliyetli araç alt sistemlerinde tercih edilir. Ayrıca oldukça basit bir yapıya sahiptir. Sahip-köle yapısı sayesinde veri transferi kolay bir şekilde gerçekleştirilir ve karmaşık zamanlama gereksinimleri yoktur.

Bunun yanında LIN, düşük hızda çalıştığı için enerji tasarrufu sağlar ve gerçek zamanlı veri aktarımı gerektirmeyen uygulamalarda ideal bir protokoldür. Örneğin, cam açma/kapama sistemleri gibi düşük veri hızlarına ihtiyaç duyan sistemlerde kullanılır.

LIN, yeni cihazların kolayca eklenebildiği esnek bir yapıya sahiptir, birden fazla alt sistemde kolayca entegre edilebilir. Son olarak diğer protokollere göre daha az kablo gerektirir, bu da montaj süresini ve maliyetini azaltır.

Dezavantajlarına bakılacak olursa maksimum 20 kbps hızında çalıştığı bu sebepten de yüksek hız ve büyük veri transferi gerektiren uygulamalar için yetersiz kaldığı söylenebilir.

LIN'in sahip-köle yapısı, her ağda bir ana sahip cihaz gerektirir. Bu da bazı durumlarda ağın esnekliğini sınırlayabilir ve zamanlama ile ilgili kısıtlamalar doğurabilir. Ayrıca LIN, zaman kritik uygulamalar için uygun değildir. Bu nedenle, gerçek zamanlı veri aktarımının önemli olduğu uygulamalarda kullanılmaz.

LIN protokolü, CAN gibi daha gelişmiş protokoller kadar güçlü bir hata tespit ve düzeltme mekanizmasına sahip değildir. Bu da güvenilirlik gerektiren uygulamalarda bir dezavantaj oluşturabilir.

Son olarak ise LIN, sınırlı bir iletişim mesafesine sahiptir. Bu yüzden geniş kapsamlı veya uzun mesafeli iletişim gerektiren sistemlerde kullanımı zordur.

LIN protokolü, özellikle düşük maliyetli, basit yapı ve düşük hız gerektiren uygulamalar için ideal olsa da, yüksek veri hızı, güvenilirlik ve gerçek zamanlılık gerektiren durumlar için yetersiz kalır. LIN veri yolunun avantaj ve dezavantajları **Tablo 5**'te kısaca maddelendirilmiştir.

Tablo 5. LIN veri yolu avantaj ve dezavantajları

Avantajlar	Dezavantajlar
Düşük Maliyet	Düşük Veri Hızı
Basitlik	Sahip-köle (Master-Slave) Yapısı
Düşük Güç Tüketimi	Gerçek Zamanlı Uygulamalarda Yetersiz
Gerçek Zamanlı Olmayan Uygulamalara Uygun	Hata Tespit ve Düzeltme Kapasitesi Sınırlı
Esneklik ve Genişletilebilirlik	Sınırlı Mesafe
Az Kablo Gereksinimi	

2.3. FlexRay veri yolu

FlexRay, yüksek hız ve güvenilirliğe sahip, özellikle otomotiv endüstrisinde güvenlik açısından kritik uygulamalar için geliştirilmiş bir iletişim protokolüdür. DaimlerChrysler ve BMW tarafından gerçekleştirilen incelemeler ve edinilen tecrübeler, mevcut veri yolu sistemlerinin hiçbirinin gelecekteki otomotiv sistemleri için gerekli tüm gereksinimleri karşılayamadığını göstermiştir. 1999 yılının ortalarında BMW ve DaimlerChrysler, gelecekteki bir veri yolu sisteminin geliştirilmesi ve spesifikasyonlarının belirlenmesi için birlikte çalışma kararı almıştır. Bu iş birliği, gelecekteki FlexRay veri yolu sistemi için gereksinimlerin belirlenmesi ile sonuçlanmıştır [23]. Bu konsorsiyum zamanla Bosch, Philips gibi diğer büyük firmaların da katılımıyla büyümüştür.

Araçlarda yüksek veri iletim hızları gereksinimi, olay tetiklemeli ve zaman tetiklemeli sistemlerde deterministik davranış beklentisi, hata toleransı, dağıtık kontrollere daha iyi desteklenebilmesi ve veri yolu sistemlerinin

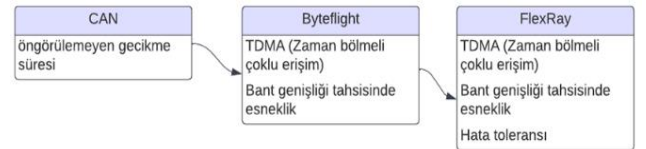
standardizasyonu gibi amaçlarla protokolün geliştirilmesi sağlanmıştır.

FlexRay, CAN ve LIN protokollerini tamamlayan ya da onların yerini alabilecek yüksek performanslı bir veri yolu çözümüdür. Ayrıca FlexRay, zaman tetiklemeli çoklu erişim (TDMA) yaklaşımını benimseyerek, güvenlik ve deterministik iletişim gereksinimlerini karşılamayı hedefler [24].

2.3.1. Byteflight/Fieldbus/FlexRay ilişkisi

Fieldbus, uzun mesafelerde ve yüksek dış yük altında farklı sistem bileşenleri arasında veri alışverişi yapmak için kullanılan seri bir veri yolu sistemidir. Fieldbus teknolojilerinin iletişim protokolü de FlexRay'e benzer şekilde çalışır. Ancak Fieldbus'ta, veri iletiminin zamanlamasını kontrol eden bir ana düğüm vardır. Buna karşın, FlexRay'de her düğüm, kendine ayrılan yuvalarda veri iletir; bu, ağın iletişim zamanlamasını sürdürmesi için tek bir düğüme bağımlı olmadığı anlamına gelir. Fieldbus konfigürasyonunda ise ana düğümün arızalanması halinde ağ başarısız olabilir [25]. Ancak Fieldbus teknolojilerinde, cihazlarda yerleşik yedeklilik bulunur ve bu, tüm ağın arızalanma olasılığını azaltır. Bunun yanında Fieldbus genellikle fabrika otomasyonu, proses kontrolü ve endüstriyel ortamlarda kullanılır. Örneğin, üretim veya endüstriyel bir ortamdaki sensörler, aktüatörler, kontrolörler ve aletler Fieldbus protokollerini kullanarak birbirleriyle iletişim kurabilir. Fieldbus ayrıca bu sistemlerin raporlama veya uzaktan izleme için datacom ağlarıyla iletişim kurmasını sağlar ve sistem ile Ethernet, TCP/IP ve çeşitli kablosuz teknolojiler gibi standart veri iletişim protokolleri arasında ağ geçidi cihazları veya çeviriciler olarak işlev görür.

Byteflight ise FlexRay'in gelişmesine katkı sağlayan bir protokoldür. Byteflight, TDMA kullanarak belirli bir bant genişliği tahsisi sağlar [26]. Bu, CAN protokolüne kıyasla daha tutarlı ve önceden belirlenmiş veri iletimi sağlar. Byteflight'ın hata toleransı konsepti ile geliştirilmesi ise FlexRay'in oluşturulmasını sağlamıştır. İki kanallı veri yolu yapısı sayesinde, veri iletiminde bir kanal arızalanırsa diğer kanal üzerinden iletim devam edebilir. **Şekil 5**'te Byteflight, Fieldbus ve FlexRay arasındaki ilişki gösterilmiştir.



Şekil 5. CAN, byteflight, flexray ilişkisini içeren diyagram

2.3.2. FlexRay çalışma mekanizması

FlexRay protokolü, zaman tetiklemeli veri iletimini destekler. Bu, ağdaki her düğümün belirli zaman dilimlerinde veri gönderebilmesi anlamına gelir. FlexRay'in temel çalışma prensibi, TDMA yöntemi ile ağın senkronize bir şekilde çalışmasını sağlamaktır. FlexRay döngüsü, statik ve dinamik segmentlerden oluşur.

Statik segment, zaman tetiklemeli veri iletimine olanak sağlar ve her bir düğümün belirlenen zaman diliminde veri göndermesini mümkün kılar. Dinamik segment, olay tetiklemeli veri iletimine olanak tanır ve ihtiyaç duyulan veri trafiği için esneklik sağlar. FlexRay ayrıca, sembol penceresi ve boşta kalma süresi gibi iletişim döngüsünü kontrol eden zaman dilimlerine sahiptir. Sembol penceresi, düğümler arasında senkronizasyon sağlarken, boşta kalma süresi ağın geri kalanı için kullanılmayan zaman dilimini ifade eder [23].

FlexRay'ın iletişim statik ve dinamik kısım arasındaki sınır, sistemin çalışma zamanı öncesinde serbestçe tanımlanabilir ve her iki kısım da boş olabilir. İletişim döngüsü bir SENKRON (SYNC) sembolüyle başlar. Zaman dilimleri, hem statik hem de dinamik kısım için kullanılan kimlik numaralarıyla (ID) tanımlanır. Her iki kanala da bağlı olan ağ düğümleri, statik kısımda mesajlarını her iki kanalda da eşzamanlı olarak gönderirler. Dinamik kısımda ise kanallardaki zaman dilimlerinde farklılıklar olabilir. Saat senkronizasyonunu sağlamak için yalnızca her iki kanala bağlı ağ düğümlerinden gelen mesajlar kullanılır. Tüm ağ düğümleri saat senkronizasyonu gerçekleştirir, ancak yalnızca statik kısımdan seçilen mesajlar bu senkronizasyon için kullanılır.

- Başlatma: İletişim sistemi, en az iki düğümün iletişime katılmasıyla başlatılır. İki kanallı bir sistemde, yalnızca her iki kanala bağlı düğümler başlatmaya katılabilir. Bir düğüm, ilk mesajını göndermeden önce, başlatma zaman aşımının sona ermesini beklemelidir. Bu süre, döngü uzunluğuna bağlıdır.
- Saat Senkronizasyonu: FlexRay'de saat senkronizasyonu, sistemdeki tüm düğümlerin küresel zamanı tutarlı ve standart bir şekilde görmesini sağlar. Senkronizasyon, sistemde en az iki aktif düğüm olduğu sürece gerçekleşir. Statik bir bölüme sahip bir sistemde, başlatma işleminden sonra dağıtık, hata toleranslı bir saat senkronizasyon algoritması kullanılır.

2.3.3. FlexRay sistem mimarisi

FlexRay, çift kanallı bir ağ mimarisi ile çalışır. Her FlexRay düğümü, bir ana birim (host), bir iletişim kontrolcüsü (CC), bir veri yolu sürücüsü (BD) ve isteğe bağlı bir veri yolu koruyucusu (BG) içerir.

- Ana Birim (Host): İletişim sürecini kontrol eden kullanıcı yazılımıdır.
- İletişim Kontrolcüsü (CC): FlexRay protokolünün tüm yönlerini uygulayan ve protokolün çekirdeği olan birimdir.
- Veri Yolu Koruyucusu (BG): Hatalı medya erişimine karşı slotları koruyan bir elektronik bileşendir.
- Veri Yolu Sürücüsü (BD): İletişim kontrolcüsünü bir iletişim kanalına bağlayan verici ve alıcıdan oluşur [27].

2.3.4. FlexRay mesaj çerçeve yapısı

FlexRay mesaj çerçevesi üç ana bölümden oluşur: başlık (header), yük (payload) ve fragman (trailer). Başlık bölümü, mesajın kimliği, uzunluğu ve hata kontrolü bilgilerini içerir. Yük bölümü ise iletilecek veriyi taşır ve fragman bölümü,

mesajın doğru bir şekilde alınıp alınmadığını kontrol eden CRC içerir.

FlexRay'de tüm mesajlar için mesaj formatı aynıdır. Bir SYNC biti ayarlandığında, ilk veri baytında döngü sayacı kodlanır. Döngü sayacı, iletişim döngülerini sayan 8 bitlik bir sayaçtır ve tüm düğümlerde sistem genelinde aynıdır.

1. ID: 10 bitlik bir tanımlayıcı, değer aralığı (1,0 ... 1023). Statik kısımda slot pozisyonunu ve dinamik kısımda önceliği tanımlar. Tanımlayıcı ne kadar küçükse, öncelik o kadar yüksektir. ID=0 SYNC sembolü için ayrılmıştır. Bir tanımlayıcı bir ağda yalnızca bir kez kullanılabilir. Her düğüm hem statik hem de dinamik kısımda bir veya daha fazla tanımlayıcı kullanabilir.
2. MUX: 1 bitlik çoklama alanı. Bu bit, aynı tanımlayıcı ile farklı verilerin iletilmesini sağlar.
3. SYNC: 1 bitlik senkronizasyon alanı. Bu bit, mesajın saat senkronizasyonu için kullanılıp kullanılmadığını ve ilk veri baytının döngü sayacını içerip içermediğini belirtir (SYNC="1": döngü sayacı ve saat senkronizasyonu ile mesaj, SYNC="0": döngü sayacı olmadan mesaj).
4. LEN: 4 bitlik uzunluk alanı, veri baytlarının sayısını belirtir (0-12,0). Döngü sayacı kullanılıyorsa (SYNC=1), LEN=11 olarak ayarlanır.
5. CYCLE: Döngü alanı, döngü sayacı olarak kullanılabilir veya ilk veri baytı olarak işlev görür. Döngü sayacı, tüm iletişim denetleyicilerinde iletişim döngüsünün başında senkronize olarak artırılır.
6. DATA: 0-12 baytlık veri baytları.
7. CRC: 15 bitlik döngüsel artıklık denetimi.

Döngü sayacına sahip bir mesajın çerçeve formatı Şekil 6'da verilmiştir.

ID	MUX	SYNCH	LEN	CYCLE	DATA	CRC	0
10 bit	1 bit	1 bit	4 bit	8 bit	0...11 bayt	15 bit	1 bit

Şekil 6. Döngü sayacı kullanılan mesaj çerçeve formatı

Döngü sayacına sahip olmayan bir mesajın çerçeve formatı ise Şekil 7'de verilmiştir.

ID	MUX	SYNCH	LEN	DATA	CRC	0
10 bit	1 bit	1 bit	4 bit	0...12 bayt	15 bit	1 bit

Şekil 7. Döngü sayacı olmadan kullanılan mesaj çerçeve formatı

2.3.5 FlexRay ağ topolojisi

FlexRay, esnek bir ağ topolojisi sunar. Yıldız topolojisi, veri yolu topolojisi ve halka topolojisi gibi çeşitli topolojilerle çalışabilir. FlexRay ayrıca aktif ve pasif yıldız bağlayıcıları da destekler. Aktif yıldız bağlayıcılar, sinyalin güçlendirilmesini sağlayarak veri iletim mesafesini artırır.

FlexRay çift kanallı bir sistem olarak geliştirilmiştir, böylece hem tek kanallı hem de çift kanallı sistemler gerçekleştirilebilir.

Yıldız topolojileri aktif veya pasif yıldız bağlayıcılar kullanılarak gerçekleştirilebilir. Aktif bir yıldız bağlayıcı, elektrik sinyallerini yükselten ve dağıtan elektronik bir cihazdır. Buna karşılık, pasif bir yıldızın aktif elemanları yoktur, bu da elektrik sinyallerinin yükseltilmediği anlamına gelir. Pasif yıldız esasen başka hiçbir bileşeni olmayan bir elektrik bağlantısıdır.

FlexRay, yalnızca "saf" topolojilerle sınırlı değildir; daha önce açıklanan topolojiler de birleştirilebilir. Veri yolu topolojisi, düğümlerin bir hat boyunca sıralandığı ve her bir düğümün sırasıyla veri aldığı sistemlerde kullanılır. Yıldız topolojisi ise merkezi bir düğümden gelen verilerin diğer düğümlere iletiildiği sistemlerde tercih edilir. FlexRay, bu topolojileri birleştirerek hibrit ağlar oluşturma imkânı sunar [28].

2.3.6 Kullanım alanları

FlexRay, özellikle otomotiv endüstrisinde, güvenlik açısından kritik uygulamalarda yaygın olarak kullanılır. Direksiyon, fren ve süspansiyon gibi x-by-wire sistemlerinde kullanılarak, araçlarda mekanik bağlantıların yerini alabilir. Bunun yanı sıra, FlexRay, sanayi otomasyonu, uçak ve tren sistemleri gibi diğer alanlarda da yaygın bir şekilde kullanılmaktadır.

2.3.7 Avantaj ve dezavantajları

FlexRay'ın en büyük avantajlarından biri yüksek veri hızıdır. FlexRay, her iki kanal üzerinden 10 Mbps hızında veri iletebilir. Bu, CAN protokolüne göre 20 kat daha hızlı veri iletimine olanak tanır. Ayrıca, FlexRay'ın deterministik yapısı, veri iletim süresinin önceden belirlenmesini sağlar ve bu da özellikle güvenlik açısından kritik uygulamalarda büyük bir avantajdır.

FlexRay'ın dezavantajları arasında maliyetlerin yüksek olması ve sistemin karmaşıklığı bulunmaktadır. FlexRay, CAN gibi daha düşük maliyetli sistemlere kıyasla daha pahalıdır. Ayrıca, FlexRay'ın uygulanması ve yönetimi, daha karmaşık bir yapı gerektirir, bu da sistemin kurulum ve bakım maliyetlerini artırabilir.

Tablo 6'da CAN, FieldBus, Byteflight ve FlexRay protokollerinin bazı özellikler bakımından kıyası verilmiştir.

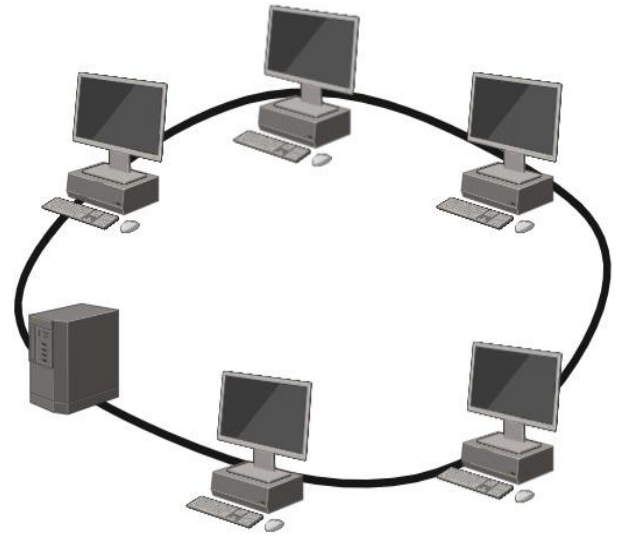
2.4 MOST protokolü

Medya Odaklı Sistem Taşıma (MOST) ağı, MOST iş birliği organizasyonu protokolü tabanlı otomotiv multimedia bilgi iletim teknolojisi tarafından geliştirilen bir iletim ağı sistemidir [29]. Yüksek iletim hızı, düşük maliyet, yüksek anti-parazit yeteneği ve esneklik gibi avantajlara sahiptir [30]. MOST, ses, video ve diğer multimedia verilerinin araç içi eğlence sistemlerinde iletimini sağlamak amacıyla geliştirilmiştir. MOST, araçlardaki yaygın veri iletim yollarından farklı olarak halkalı topolojiye dayanmaktadır. Fiber optik kablo tabanlı bir veri iletim protokolü olan MOST, ilk olarak 2001 yılında kullanıma sunulmuştur. 2013 yılı itibarıyla 140 araç modelinde kullanılmaya başlanmış ve Audi A3, Mercedes S serisi gibi

araçlarda yer almıştır. Günümüzde birçok otomobil markası tarafından tercih edilmektedir [31].

Geleneksel veri yolu ağı sistemi düşük güvenilirliğe ve hata toleransına sahiptir. Bu nedenle, ses verilerinin geleneksel araç veri yolu üzerinden yüksek bant genişliğinde iletimi yeni bir sorun haline gelmiştir. MOST yeni ve esnek bir mimariye sahip bir iletişim sistemidir. Ses sinyallerini eşzamanlı olarak iletebilir ve günümüzde otomobillerde en yaygın kullanılan multimedia sistemidir [32].

MOST protokolünde, alıcı-vericiler değiştirildikten sonra kablo bağlantıları üzerinden iletişim mümkündür. Cihazlardan biri (master) halkadaki cihazlar için tüm veri tabanını içerir. Şekil 8'de aygıtlar üzerinde halka topolojisinin gösterimi verilmiştir. Bunun yanı sıra bu protokolün en büyük avantajı, yüksek bant genişliği sunması ve veri iletiminde yüksek güvenilirlik sağlamasıdır.



Şekil 8. Aygıtlar üzerinde gösterilmiş temsili halka topolojisi

2.4.1 Mesaj çerçeve yapısı

MOST protokolü, veri iletiminde belirli bir çerçeve yapısı kullanır. MOST25, MOST50 ve MOST150 olmak üzere üç farklı versiyonu bulunur ve bu versiyonlar veri iletim hızlarına göre adlandırılmıştır. Her bir versiyonun çerçeve yapısı farklıdır. MOST25'teki veri alışverişi protokolü diğer protokollerden (örneğin CAN, LIN) çok daha karmaşıktır [33]. MOST25'te veri çerçevesi 16 alt çerçeveden oluşurken, MOST150'de bu sayı 384 byte'a kadar çıkabilir [31]. Çerçevede kontrol verileri, senkron veri ve asenkron veri olmak üzere üç farklı veri türü bulunur.

En tipik veri türü, multimedia verilerini temsil eden ve çerçevenin yapılandırılabilir en büyük bölümünü kaplayan senkron veridir. Asenkron veriler, GPS sistemleri, erişilen dosyalar hakkında bilgiler ve MOST sistemi içindeki kapsüllenmiş protokollerin verileri gibi çoklu ortam bilgilerini desteklemek için kullanılır. Kontrol verileri ağ bağlantı noktaları arasındaki iletişimin yönetilmesinden sorumludur. Bu nedenle, veri çerçevesi 32 bayt uzunluğundadır ve çerçevenin her bir iletişim birimi tarafından 16 alt çerçeveye bölünmesi gerekir (Şekil 9).

Tablo 6. CAN,fieldbus, byteflight, flexray kıyası

Özellik	CAN	FieldBus	Byteflight	FlexRay
Mesaj iletimi	asen kron	sen kron	sen kron ve asen kron	sen kron ve asen kron
İletişim Yöntemi	Dinamik arbitraj ile zaman tetiklemeli iletişim (üst protokol kullanılarak)	Master-Slave konfigürasyonu, veriyi ileten düğümü master belirler	TDMA	TDMA, ancak dinamik iletişime izin verir
Mesaj tanımlama	mesaj tanımlayıcı	zaman dilimi	mesaj tanımlayıcı	zaman dilimi
Veri hızı	1 Mbps brüt	12 Mbps brüt	10 Mbps brüt	10 Mbps brüt
Bit kodlama	Bit doldurmalı NRZ	Modifiye frekans modülasyonu (MFM)	Başlangıç/bitiş bitli NRZ	Başlangıç/bitiş bitli NRZ
İletim Bayt Sayısı	0-8	1-244 veri baytı ve 9-11 kontrol baytı (PROFIBUS-PA)	0-12	0-254
İletişim Kanalı Sayısı	1	1	1-2 – ikinci kanal yedek olarak kullanılabilir	2 – ikinci kanal yedek olarak kullanılabilir
Fiziksel katman	1 Mbps'ye kadar verici/alıcı	Master tarafından, 100 ns aralığında saat senkronizasyonu ile desteklenir.	10 Mbps'ye kadar optik verici/alıcı	Diferansiyel sinyalle 10 Mbps
Saat senkronizasyonu	sağlanmamış	master tarafından, 100 ns aralığında	master tarafından, 100 ns aralığında	dağıtık, µs aralığında
Zaman uyumluluğu	desteklenmiyor	destekleniyor	yüksek öncelikli mesajlar için desteklenir	destekleniyor
Gecikme titreşimi	veri yolu yüküne bağlı	tüm mesajlar için sabit.	t_cyc'ye göre yüksek öncelikli mesajlar için sabit	tüm mesajlar için sabit
Hata Denetimi	15 bit CRC	1 baytlık çerçeve kontrol sırası (PROFIBUS-PA)	Optik fiber ve transceiver ile sağlanır, başlangıç/bitiş bitleri ile NRZ kodlama kullanılır.	11 bit başlık CRC & 24 bit çerçeve CRC
Genişletilebilirlik	zaman-kritik olmayan uygulamalar için mükemmel	yalnızca orijinal tasarımda genişletme planlanmışsa mümkün	yüksek öncelikli mesajlar için genişletme ve bant genişliği üzerinde etkisi	fonksiyonel ve yapısal alanın ayrılması
Esneklik	her düğüm için esnek bant genişliği	her düğüm için yalnızca bir mesaj ve TDMA döngüsü.	her düğüm için esnek bant genişliği	her düğüm için çoklu yuva, dinamik
Karmaşıklık	Düşük	Yüksek	Yüksek	Yüksek
Kullanım Alanları	CAN otomasyon ve otomotiv uygulamalarında yaygın olarak kullanılır	Otomasyon amaçları için yaygın olarak kullanılır	Otomotiv uygulamalarında, özellikle güvenlik sistemlerinde yaygın olarak kullanılır.	Yeni protokol, geniş çapta kullanılmıyor ancak gelecekteki uygulamalar için yüksek seviyede araştırmalar yapılıyor

2.4.2 MOST protokolü versiyonları

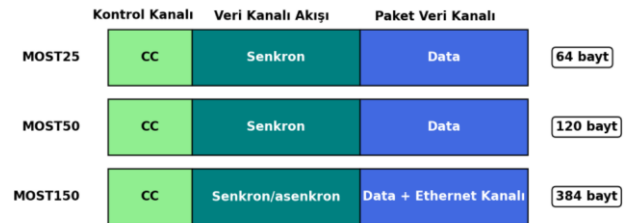
MOST protokolünde versiyon isimleri Mbps cinsinden veri hızlarına göre isimlendirilmiştir. Tablo 7'de versiyonların çerçeve boyutları byte cinsinden görülebilir.

Tablo 7. MOST protokolü çerçeve boyutları

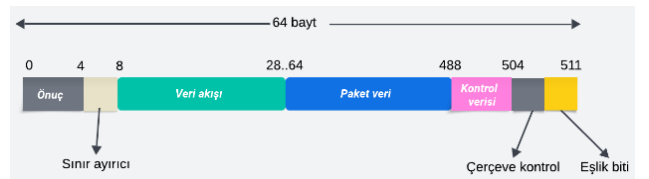
Versiyon	Veri Hızı (Mbps)	Çerçeve Boyutu (Byte)
MOST25	25	16 Alt Çerçeve
MOST50	50	128 Byte
MOST150	150	384 Byte

MOST25: MOST25 protokolünde ikinci katman, iletişimde çerçeve yapısına dayanır ve bu çerçeveler 16 çerçevelik bloklar halinde gönderilir. Her bir çerçeve, senkronize veri (örneğin, ses verisi) ve paket veri (örneğin, GPS verisi) içerir. Çerçevadaki senkron ve paket veri bölümlerinin boyutları sınır işaretleyicileri ile önceden belirlenmiştir. Paket veri bölümü en fazla 9 dört bitlik kelime içerir ya da hiç veri taşımayabilir. Ağın kontrolü için kullanılan kontrol mesajları, büyük bir mesajın parçaları ayrılmasıyla bloklar halinde iletilir [31].

MOST25 protokolünde senkron veri bölümü, sürekli veri akışı gerektiren ses gibi veriler için kullanılırken, paket veri bölümü GPS gibi sistemlerden gelen asenkronize verilerin iletimine olanak tanır.



Şekil 9. MOST protokolü mesaj çerçeve yapısı



Şekil 10. MOST25 protokolü çerçeve yapısı

MOST50: MOST50, veri iletim hızını MOST25'e kıyasla iki kat artırır ve fiziksel katmanda çerçeve yapısında değişiklik yapılmıştır. MOST50 çerçevesi, 128 bayt büyüklüğünde olup daha basit ve esnek bir yapı sunar. Çerçevenin başlangıcında 11 baytlık bir kontrol alanı bulunur. Stream veya paket veri kullanılmadığında, bu bölümler kaldırılabilir, böylece çerçeve daha verimli bir şekilde kullanılabilir.

MOST50 protokolü, artan hız ve sadeleştirilmiş çerçeve yapısıyla veri iletimini optimize eder. Kontrol verilerinin tek bir başlık alanında toplanması, veri yönetimini ve hata ayıklamayı daha kolay hale getirir. Ayrıca, çerçeve boyutunun iki katına çıkarılması, daha fazla verinin taşınmasına olanak tanır [31].



Şekil 11. MOST50 protokolü çerçeve yapısı

MOST150: MOST150, MOST50'ye kıyasla üç kat daha yüksek bir veri iletim hızına sahiptir ve 384 baytlık çerçeveler kullanır. Çerçeve yapısı değişmeden kalmış olsa da, veri alanı genişletilmiştir. MOST150, senkron veri bölümüne ek olarak izokron veri türünü destekler. İzokron veri, senkron iletişim hızından farklı bir örnekleme hızına sahip veri türüdür ve bu özellik, özellikle ses ve video akışı gibi uygulamalarda senkron iletişim ortamının simüle edilmesi için kullanılır.

MOST150 protokolü, yüksek hızda veri iletimi gereken ses ve görüntü gibi uygulamalarda tercih edilir. Hizmet Kalitesi (QoS) desteği ile belirli uygulamalar için gerekli bant genişliği ayrılabilir ve Ethernet tabanlı veri iletimine olanak tanır. Ayrıca, senkron veri iletişiminin yanı sıra asenkron veri kanalı da bulunur ve bu kanal üzerinden TCP/IP protokolü kullanılarak veri iletimi gerçekleştirilebilir [31].

Çarpışma uyarısı, trafik işareti izleme, şeritten ayrılma uyarısı, şerit yönlendirme, yaya uyarısı, gece görüşü, uyarılar hız sabitleyici veya çarpışma öncesi uyarı gibi gelişmiş sürücü destek sistemleri çok çeşitli araç alt sistemleriyle entegrasyon gerektirdiğinden MOST150 bu sistemler için rahatça kullanılabilir [34].

MOST150 tarafından sunulan 150 Mbps aralığındaki verim, araç iletişim ağları içinde en yüksek olanıdır. Buna rağmen, optik fiberlerin fiziksel katmanından yararlanmaya çalışan tasarımcılarla daha da hızlı çözümler aranmaktadır.



Şekil 12. MOST150 protokolü çerçeve yapısı

Sonuç olarak bu versiyonlar karşılaştırıldığında MOST150'nin yüksek veri hızları ve geniş bant genişliği gerektiren uygulamalar için en uygun protokol olduğu; MOST50'nin orta seviyedeki veri iletim gereksinimlerini karşılayacak şekilde optimize edildiği; MOST25'in ise düşük veri ihtiyacı olan uygulamalar için ideal olduğu görülmektedir.

Tablo 8'de bu versiyonların bant genişliği bilgileri dolayısıyla iletişim hızları karşılaştırılmıştır.

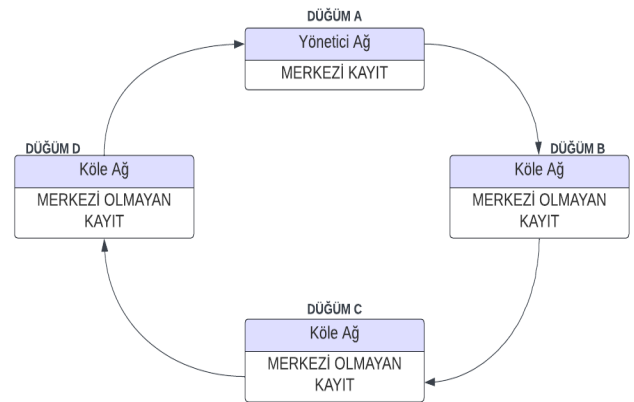
2.4.3 MOST protokolü versiyonları

MOST protokolü bir halka topolojisi kullanarak çalışır. Her bir cihaz (düğüm) gelen veri çerçevesini işler ve bir sonraki cihaza iletir. Bu halka yapısı, veri iletiminde yüksek hız ve güvenilirlik sağlar. Ağda bir ana düğüm (master) bulunur ve diğer cihazlar bu ana düğüme senkronize olur. Ana düğüm, ağdaki cihazların durumunu yönetir ve sistemin düzgün çalışmasını sağlar.

Önceki düğümden alınan veri bloğu bilgi ve komut kaynağı olarak kullanılır. Önceki düğümden alınan blok yeniden oluşturulur ve iletir. Kapalı cihazlar optik sinyali analizi olmadan iletir. Veri aktarımı, blok gönderici tarafından alındığında tamamlanır (Şekil 13).

Tablo 8. MOST protokol versiyonlarının kıyası

Parametre	MOST25	MOST50	MOST150
Akış verisi			
Minimum bant genişliği [Mbps]	8.48	0.38	0
Maksimum bant genişliği [Mbps]	21.17	44.93	142.85
Paket verisi			
Minimum bant genişliği [Mbps]	0	0	0
Maksimum bant genişliği [Mbps]	10.84	44.54	142.85
Kontrol verisi			
Minimum bant genişliği [kbps]	405.84	448.0	512.0
Maksimum bant genişliği [kbps]	405.84	810.62	1130.0



Şekil 13. MOST haberleşme diyagramı – yönetici ve köle ağlar

2.4.4 Kullanım alanları

MOST protokolü, özellikle otomotiv endüstrisinde araç içi multimedya sistemlerinin yönetiminde yaygın olarak kullanılmaktadır. Araç içi eğlence sistemleri, seyrüsefer sistemleri, kamera görüntüleme sistemleri gibi birçok multimedya bileşeni, MOST ağı üzerinden birbirine bağlanır ve senkronize şekilde çalışır.

2.4.5 Avantaj ve dezavantajları

MOST protokolünün en büyük avantajı, yüksek veri iletim hızına sahip olması ve veri iletiminde yüksek güvenilirlik sağlamasıdır. Fiber optik kablo kullanımı, elektromanyetik parazitlere karşı dayanıklılık sağlar. Ancak MOST sisteminin kurulum maliyetleri, fiber optik kablo altyapısı ve özel donanım gerektirdiği için yüksektir. Bu sebepten maliyet açısından, fiber optik kablolar ve donanımlar diğer ağ çözümlerine göre daha pahalı olabilir.

2.5 Ethernet

Ethernet, bilgisayar ağlarında veri iletimi için kullanılan bir teknoloji olup, ilk olarak yerel alan ağlarında (LAN) yaygınlaşmıştır. Günümüzde Ethernet, hem tüketici elektroniği hem de endüstriyel ve otomotiv uygulamalarda geniş çapta kullanılmaktadır. Ethernet'in avantajları arasında yüksek bant genişliği, düşük gecikme süreleri ve esnek yapı bulunur. Özellikle otomotiv endüstrisinde Ethernet, CAN ve FlexRay gibi geleneksel iletişim protokollerine kıyasla daha yüksek veri hızları sunması nedeniyle tercih edilmektedir [25, 26].

2.5.1 Donanım gereksinimi

Ethernet ağı için gereken donanım bileşenleri arasında ağ kartları, anahtarlar (switch), yönlendiriciler (router) ve kablolar bulunur. Otomotiv uygulamaları için Ethernet, özellikle sınırlı alan ve elektromanyetik uyumluluk (EMC) gereksinimlerine uygun olacak şekilde optimize edilmiştir. Standart Ethernet kabloları genellikle twisted pair kablolar üzerine kuruludur, ancak otomotiv uygulamalarında daha ince ve hafif kablolar tercih edilir [27, 28].

2.5.2 Mesaj yapısı

Ethernet'in temel mesaj yapısı, bir çerçeve formatına dayanır. Her bir çerçeve, başlık (header), veri alanı (payload) ve sonlandırma (trailer) bölümlerinden oluşur. Başlık kısmı, kaynaktan hedefe verilerin nasıl iletileceği hakkında bilgi sağlar. Veri alanı, iletilen bilginin kendisini içerirken, sonlandırma kısmı veri bütünlüğünü sağlamak için hata denetimi gibi bilgileri içerir. Otomotiv uygulamalarında bu yapı, zaman hassasiyeti olan veri akışları için optimize edilebilir [35]. Şekil 14'te Automotive Ethernet (AE) mesaj yapısı verilmiştir.

Önuc	SOF	Hedef Adres	Kaynak Adres	VLAN	Tür	DATA	FCS
7 bit	1 bit	6 bit	6 bit	4 bit	2 bit	46-1500	4 bit

Şekil 14. Automotive ethernet mesaj çerçeve yapısı

2.5.3 Automotive Ethernet çeşitleri

- Zaman Tetiklemeli Ethernet (TTE): SAE tarafından SAE6802 standardıyla tanımlanmıştır. Genellikle güvenliğin ön planda olduğu uygulamalarda, özellikle ulaşım sektöründe ve endüstriyel otomasyon sistemlerinde kullanılır. IEEE 802.3 Ethernet ile uyumlu olup, zaman tetiklemeli teknoloji sayesinde standart Ethernet'in sağlam hata toleransı ve gerçek zamanlı işlevleriyle entegre edilmiş belirleyici gerçek zamanlı iletişimi mümkün kılar.
- Esnek Zaman Tetiklemeli Ethernet (FTTE): Katı gerçek zamanlı işlemleri paylaşımlı veya anahtarlamalı Ethernet üzerinden sorunsuz ve esnek bir şekilde destekler. CAN'de olduğu gibi, gerçek zamanlı trafiğin kaynağı adresleme şeması aracılığıyla belirlenir. FTT Ethernet protokolü, etkili bir ana/multi-köle iletim kontrol tekniği kullanarak çevrimiçi zamanlama ve kabul kontrolünü birleştirir, böylece dinamik iletişim gereksinimlerinde dahi gerçek zamanlı iletişimi garanti eder.
- Zaman Duyarlı Ağ (TSN): IEEE'nin 802.1 çalışma grubu tarafından geliştirilen Ethernet standartları kümesini ifade eder. TSN, veri iletiminde zaman garantileri sağlamayı amaçlayan bir standart Ethernet genişlemesidir. TSN teknolojisinin temel yetenekleri şunlardır:
 - Ağ elemanları arasında zaman senkronizasyonu sağlama,
 - Kontrollü ve öngörülebilir gecikmeler,
 - İletişim yollarının seçimi ve hata toleransı,
 - Yedeklilik.
- Kablosuz Zaman Duyarlı Ağ (WTSN): TSN kavramını kablosuz iletişime uyarlayarak garantili kablosuz hizmetler sunar. Esnek kurulum imkânı, daha düşük bakım maliyetleri, yeniden yapılandırılabilirlik ve hareket kabiliyeti gibi avantajlar sağlar. WTSN, Endüstri 4.0 için esnek ve verimli bir teknoloji olarak öne çıkacaktır [36].

2.5.4 Ethernet'in otomotivde kullanımı

Ethernet, otomotiv sektöründe yüksek veri hızı ve esnek yapı sunması nedeniyle yaygın bir şekilde kullanılmaktadır. Özellikle Gelişmiş Sürücü Destek Sistemleri (ADAS), otonom sürüş ve eğlence sistemlerinde veri transferi kritik önemdedir. Tablo 9, Ethernet'in otomotivde kullanım alanlarını ve sağladığı avantajları gösteriyor.

2.5.5 Avantajlar ve dezavantajlar

AE'nin avantajları şunlardır:

- Standartlaştırılmıştır: ISO, AE protokolünü ISO 21111 ile standartlaştırmıştır.
- Daha yüksek bant genişliği sağlar: LIN (20 kb/s), CAN (1 Mb/s), FlexRay (10 Mb/s) ve MOST (25, 50 veya 150 Mb/s) ile karşılaştırıldığında daha yüksek veri iletim hızlarına sahiptir ve yanıt süresi kısadır.
- Dünyanın en yaygın ağ standardı: TCP/IP, genellikle Ethernet üzerinden uygulanır. Ethernet, en yaygın ağ arabirim protokolü olduğu için, iyi test edilmiş ve üst katman yazılım arayüzleri standart Ethernet ile aynıdır.

- Yüksek esneklik sağlar: Tak ve çalıştır özelliklerine sahiptir, bu da bileşenlerin gerektiğinde otomatik algılama ve yapılandırma ile bağlanıp çıkartılmasına olanak tanır.
- EMC, EMI (Elektromanyetik Girişim) ve otomotiv uygulamaları için gerekli sıcaklık sınıfı gereksinimlerini karşılar.
- Her çerçevenin bir kaynağı ve bir veya birden fazla hedefi vardır: Diğer otomotiv ağ türlerinde, çerçevenin vericisi tanımlanmaz ve tüm düğümler her zaman tüm çerçeveleri alır, ardından çerçeve içeriğinin kullanılıp kullanılmayacağına karar verir.
- Ethernet teknolojisi ölçeklenebilir: Günümüzün otomotiv sistemlerinin getirdiği ölçeklenebilirlik gereksinimlerini karşılar (yani birbirine bağlanacak düğüm sayısı sürekli artmaktadır).

AE'nin dezavantajları şunlardır:

- Düşük hata toleransı vardır.
- Ethernet'in yıldız topolojisi nedeniyle, her düğümün tek bir bağlantı noktasına (anahtara) geri bağlanması gerekir: Bu, kablolama açısından daha sınırlayıcı gereksinimler getirir.
- Ağa daha fazla anahtar eklemek, ağaç topolojisi yaratır: Bu, daha fazla esneklik sağlar ancak maliyeti artırır.
- Ethernet, "en iyi gayret" ile çalışan bir iletişim protokolüdür: Bu, güvenilirlik ve zamanında teslimatlar konusunda güvenilir olmadığı anlamına gelir.
- Ethernet, Radyo Frekansı Girişimi'nden (RFI) etkilenir: Bu nedenle, güvenli ve gerçek zamanlı sistemlerde kullanılamaz.

Bu son dezavantajın bir sonucu olarak, Ethernet standardına son on yılda birçok geliştirme eklenmiştir. Bu geliştirmeler, gerçek zamanlı özellikleri destekleyebilmek amacıyla tasarlanmıştır. Zaman Tetiklemeli Ethernet (TTE), Esnek Zaman Tetiklemeli Ethernet (FTTE), Zaman Duyarlı Ağ (TSN) ve Kablosuz Zaman Duyarlı Ağ (WTSN) gibi çözümler, belirli gecikmeleri garanti edebilecek gelecek vaat eden çözümler arasındadır [36].

Tablo 10. Araç içi haberleşme sistemlerinin teknik açıdan karşılaştırılması

Özellik	CAN	LIN	Flex Ray	MOST	AE
Maksimum Veri Hızı	1 Mbps	20 kbps	10 Mbps	150 Mbps	1 Gbps (ve üzeri)
Maksimum Veri Alanı	8 bayt (CAN) / 64 bayt (CAN FD)	8 bayt	254 bayt	64 bayt (çerçeve başına)	Maksimum MTU (1500 bayt)
İletişim Türü	Çoklu erişim (Broadcast)	Master-Slave	Deterministik (Zaman tetiklemeli)	Ring Topolojisi	Tam çift yönlü (Full Duplex)
Hata Toleransı	Orta	Düşük	Yüksek	Orta	Yüksek
Kullanım Alanı	Motor kontrolü, fren sistemleri	Sensörler, klima kontrolü	Güvenlik kritik sistemler, otonom sürüş	Multimedya sistemleri	ADAS, otonom sürüş
Maliyet	Düşük	Çok düşük	Yüksek	Orta-Yüksek	Orta-Yüksek
Uyumluluk	Otomotivde yaygın, eski sistemlerle uyumlu	Basit uygulamalar için uygun	Lüks ve otonom araçlarda kullanılıyor	Multimedya odaklı, diğer sistemlerle sınırlı uyumluluk	Yeni nesil araçlarda yaygınlaşıyor

Tablo 9. Automotive ethernet kullanım alanları

Kullanım Alanı	Açıklama	Avantajları
ADAS (Gelişmiş Sürücü Destek Sistemleri)	Yüksek bant genişliği ile radar, kamera ve sensör verileri işlenir.	Düşük gecikme süresi, yüksek veri aktarım hızı
Eğlence Sistemleri	Araç içi multimedya ve internet erişimi sağlar.	Güvenilir bağlantı, yüksek hız
Otonom Sürüş	Araç sensörleri ve kontrol üniteleri arasındaki veri akışını sağlar.	Zaman duyarlı ağ (TSN) ile hassas veri yönetimi

5. Araç içi haberleşme sistemlerinin karşılaştırılması

CAN, LIN, FlexRay, MOST ve AE gibi protokoller farklı veri hızlarına, topolojilere ve iletişim mimarilerine sahiptir. CAN ve LIN gibi olay tetiklemeli protokoller, daha düşük hızlar ve daha az karmaşık uygulamalar için uygundur. Öte yandan, FlexRay ve MOST gibi zaman tetiklemeli protokoller, özellikle x-by-wire uygulamaları ve multimedya sistemleri gibi yüksek hız ve gerçek zamanlı veri iletimini gerektiren uygulamalarda kullanılmaktadır. AE ise, Ethernet'in esnekliğini ve yüksek hızını sunduğundan, gelecekte araç içi iletişim sistemlerinin temel protokollerinden biri olma potansiyeline sahiptir.

Her protokolün kendine özgü avantajları ve dezavantajları vardır. CAN, yaygın kullanımı, düşük maliyeti ve sağlamlığı ile öne çıkarken, LIN düşük hız gerektiren basit uygulamalar için idealdir. FlexRay, yüksek hız ve güvenilirliği ile kritik uygulamalar için tasarlanmıştır, ancak maliyeti ve karmaşıklığı yüksektir. MOST, yüksek hız ve bant genişliği gerektiren multimedya uygulamaları için uygundur, ancak yüksek maliyetli bir çözümdür. AE ise veri iletim hızının yanı sıra esnekliği ile ön plana çıkmaktadır, ancak güvenilirlik konusunda bazı sınırlamaları vardır. [36]. Tablo 10'da CAN, LIN, FlexRay, MOST ve AE için teknik karşılaştırmalar verilmiştir.

6. Araç içi haberleşme sistemlerinin avantaj ve dezavantajları

Aşağıdaki tabloda araç içi haberleşme yollarının avantaj ve dezavantajları verilmiştir.

Tablo 11. Araç içi haberleşme sistemlerinin avantaj ve dezavantajları

Protokol	Avantajlar	Dezavantajlar
CAN	Standartlaştırılmış, düşük maliyet, uygulama kolaylığı, çoklu master ve çoklu yayın, esneklik, yüksek verimlilik, EMI toleransı, verimlilik	Sınırlı uzunluk (40 metre), düşük performans, yüksek yazılım ve bakım maliyetleri, veri yolunun yüksek dirençleri ile sonlandırılması gerekliliği
LIN	Standartlaştırılmış, uygulaması kolay, düşük maliyetli, esnek, belirleyici, kendi kendini senkronize eder, EMI azaltma, garantili gecikme süreleri	Düşük veri hızı, ana-köle yapılandırması, düşük hata toleransı
FlexRay	Standartlaştırılmış, yüksek veri hızı, farklı topolojileri destekler, yüksek güvenilirlik, hem senkron hem de asenkron veri aktarımı, iki bağımsız kanal	Yüksek maliyet, karmaşık uygulama, düşük çalışma voltaj seviyeleri
MOST	Standartlaştırılmış, esnek, yüksek veri hızı, değişken baud hızlarını destekler, optik fiber kullanımı	Çok yüksek maliyet, düşük hata toleransı
AE	Standartlaştırılmış, yüksek veri iletim hızı, yüksek esneklik, düşük yanıt süresi, EMC ve EMI gereksinimlerini karşılar, ölçeklenebilir	Düşük hata toleransı, yüksek maliyet, yıldız topolojisi nedeniyle daha fazla kablolama gereksinimi, RFI'dan etkilenir, güvenilir olmayan bir protokoldür

7. Sonuçlar

Bu çalışmada otomotiv sektöründe otonom sürüş teknolojilerinin gelişimi ve bu süreçte veri yollarının kritik rolü ele alınmıştır. Günümüzde otomobillerde kullanılan çeşitli veri yolu protokollerinin (CAN, LIN, FlexRay, MOST ve Ethernet) her birinin kendine özgü avantaj ve sınırlamaları bulunmaktadır. CAN, düşük maliyetli ve güvenilir bir yapı sunarken; LIN düşük veri hızı gerektiren basit uygulamalarda tercih edilmektedir. FlexRay, yüksek hız ve güvenilirlik gerektiren kritik uygulamalarda öne çıkarken; MOST, yüksek bant genişliği gerektiren multimedya sistemleri için uygun bir çözümdür. Ethernet ise yüksek veri iletim hızı ve esnek yapısıyla geleceğin akıllı araç teknolojileri için potansiyel bir aday olarak görülmektedir.

Ayrıca havacılıkta kullanılmakta olan veri yollarının da otomotivde kullanılabilirliği üzerine birtakım araştırmalar yapılmış ve ARINC 818 veri yolunun 28.05 Gbps'ye kadar veri iletim hızını desteklemesinin yanında bu veri yolunun güvenilir, esnek, kararlı olması gibi çok fazla avantajları bulunması sebebi ile otomotiv sektörünün ihtiyaçları özelinde incelenmesinin geleceğin akıllı taşıtlarının veri yolları tasarımıyla önemli olabileceği öngörülmektedir [37]. Bununla birlikte, otonom sürüş ve yapay zekâ gibi yeni teknolojilerin yükselişiyle, veri yolu güvenliği, hız ve genişletilebilirlik gereksinimleri daha da ön plana çıkmaktadır.

Günümüzde otomotiv veri yolu teknolojileri, bağlantılı ve otonom araçların yaygınlaşması, elektrikli araçların artışı ve V2X iletişimi gibi faktörlerle dönüşüm geçirmektedir. CAN veri yolu, sınırlı hız kapasitesi nedeniyle CAN FD ile

değiştirilmeye başlanmıştır [38]. LIN, düşük maliyetli sistemlerde varlığını sürdürebilir ancak uzun vadede kablosuz haberleşme teknolojileri ile değişebilme olasılığı mevcuttur.

FlexRay, deterministik iletişim avantajına rağmen Automotive Ethernet'in sunduğu yüksek hız ve esneklik nedeniyle otomotiv endüstrisinde giderek daha az tercih edilmektedir. MOST ise, multimedya sistemleri için geliştirilmiş olmasına rağmen karmaşık yapısı ve yüksek maliyeti nedeniyle Ethernet tabanlı çözümlerle değiştirilmektedir [39]. Automotive Ethernet, 1 Gbps ve üzeri hızları, tam çift yönlü iletişimi ve yüksek hata toleransı sayesinde özellikle ADAS, otonom sürüş ve yüksek bant genişliği gerektiren sistemlerde standart hale gelmektedir. Siber güvenlik önlemlerinin artırılması amacıyla Secure CAN ve kriptografik doğrulama yöntemleri yaygınlaşması ile veri yollarının güvenilirliğinin artması beklenmektedir. Bu bağlamda, gelecekte daha yüksek hız ve güvenlik sağlayan yeni protokol ve sistemlerin geliştirilmesi gerekli olduğu ve bu alanda gelişimlerin sağlanacağı açıkça ortaya çıkmaktadır.

Çıkar çatışması

Yazarlar çıkar çatışması olmadığını beyan etmektedir.

Benzerlik oranı (iThenticate): %11

Kaynaklar

- [1] Bosch. CAN Specification (Version 2.0). Robert Bosch GmbH, 1991.
- [2] LIN Consortium. LIN Specification (Version 2.2A). LIN Consortium, 2010.
- [3] MOST Cooperation. MOST Specification (Version 3.0). MOST Cooperation, 2014.
- [4] FlexRay Consortium. FlexRay Communications System Specification (Version 3.0). FlexRay Consortium, 2010.
- [5] OPEN Alliance. Automotive Ethernet Specification. OPEN Alliance, 2014.
- [6] 3GPP. Release 14: Cellular V2X. 3rd Generation Partnership Project, 2017.
- [7] Li, R., Liu, C., & Luo, F. A design for automotive CAN bus monitoring system. In 2008 IEEE vehicle power and propulsion conference pp. 1-5. IEEE, September, 2008. <https://doi.org/10.1109/VPPC.2008.4677544>
- [8] Lokman, S. F., Othman, A. T., & Abu-Bakar, M. H. Intrusion detection system for automotive Controller Area Network (CAN) bus system: a review. EURASIP Journal on Wireless Communications and Networking, 2019 (1), 2019. <https://doi.org/10.1186/s13638-019-1484-3>
- [9] Guştin, M. CAN Bus Security Protocol: lightweight message confidentiality, authentication, and freshness on an automotive bus (Doctoral dissertation, Graz University of Technology), 2022.
- [10] Keskin, U. In-vehicle communication networks: a literature survey, 2009.
- [11] Gowda, M. B., Biradar, R. C., & Kotgire, M. K. CAN controlled complex real time stimulus for verification

- and validation of powertrain systems. In 2018 Second International Conference on Advances in Electronics, Computers and Communications (ICAEECC) pp. 1-4. IEEE, February, 2018. <https://doi.org/10.1109/ICAEECC.2018.8479476>
- [12] Freiburger, S., Albrecht, M., & Käufl, J. Reverse engineering technologies for remanufacturing of automotive systems communicating via CAN bus. *Journal of remanufacturing*, 1, 1-14, 2011. <https://doi.org/10.1186/2210-4690-1-6>
- [13] Bozdal, M., Samie, M., & Jennions, I. A survey on can bus protocol: Attacks, challenges, and potential solutions. In 2018 International Conference on Computing, Electronics & Communications Engineering (iCCECE) pp. 201-205, IEEE, August 2018. <https://doi.org/10.1109/iCCECOME.2018.8658720>
- [14] Wolf, M., Weimerskirch, A., & Paar, C., "Secure automotive on-board networks," *Proceedings of the IEEE*, 2006.
- [15] Cho, K., & Shin, K. G., "Fingerprinting electronic control units for vehicle intrusion detection," *USENIX Security Symposium*, 2016.
- [16] Larson, U. E., & Nilsson, D. K., "Securing Vehicles Against Cyber Attacks," *Proceedings of the 16th International Conference on Embedded Security*, 2008.
- [17] Marchetti, M., & Stabili, D., "Anomaly Detection of CAN Bus Messages Through Analysis of ID Sequences," *IEEE Transactions on Vehicular Technology*, 2017. <https://doi.org/10.1109/IVS.2017.7995934>
- [18] AREA, B. S. S. A. (n.d.). *Automotive Bus Systems*. Retrieved from Atmel website.
- [19] Hardt, W., & Schmidt, D. I. R. (n.d.). *Automotive Communication Buses*.
- [20] Piao, C., Liu, Q., Huang, Z., Cho, C., & Shu, X. VRLA battery management system based on LIN bus for electric vehicle. In *Advanced Technology in Teaching Selected papers from the 2012 International Conference*, 2013.
- [21] Artal, J. S., Caraballo, J., & Dufo, R. CAN-LIN-Bus protocol. Implementation of a low-cost serial communication network, June 2014. <https://doi.org/10.1109/TAEE.2014.6900168>
- [22] Xu, Y., Wang, J., Chen, W., Tao, J., & Liu, Q. Application of LIN bus in vehicle network. In 2006 IEEE International Conference on Vehicular Electronics and Safety pp. 119-123. IEEE, December 2006. <https://doi.org/10.1109/ICVES.2006.371566>
- [23] Berwanger, J., Ebner, C., Schedl, A., Belschner, R., Fluhrer, S., Lohrmann, P. & Mores, R. FlexRay—the communication system for advanced automotive control systems. *SAE transactions*, 2001. <https://doi.org/10.4271/2001-01-0676>
- [24] Shaw, R. & Jackman, B. An introduction to FlexRay as an industrial network. *IEEE International Symposium on Industrial Electronics*, 2008. <https://doi.org/10.1109/ISIE.2008.4676987>
- [25] PROFIBUS, PROFIBUS PA technology and application, System Description, Karlsruhe: PROFIBUS Nutzerorganisation e.V, 2007.
- [26] Makowitz, R., & Temple, C. Flexray-a communication network for automotive control systems. In 2006 IEEE International Workshop on Factory Communication Systems pp. 207-12) IEEE, June 2006. <https://doi.org/10.1109/WFCS.2006.1704153>
- [27] Xu, Y. N., Jang, I. G., Kim, Y. E., Chung, J. G. & Lee, S. C. Implementation of FlexRay protocol with an automotive application. *International SoC Design Conference*, 2008. <https://doi.org/10.1109/SOCD.2008.4815675>
- [28] RAUSCH, M. FlexRay. CARL HANSER VERLAG GMBH & 2007
- [29] MOST Cooperation, *MOST Specification Rev. 3.0 E2*, MOST Cooperation, ch.2, 2010
- [30] Lu, S., Fang, Z., Qu, G., & Gao, S. ETBAC-Based model in media oriented system transport network. In 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications, 2015. <https://doi.org/10.1109/CIT/IUCC/DASC/PICOM.2015.72>
- [31] Sumorek, A., & Buczaj, M. The evolution of 'media oriented systems transport' protocol. *Teka Komisji Motoryzacji i Energetyki Rolnictwa*, 14 (3), 115-120, 2014.
- [32] MOST Cooperation, *MOST Book*, 2st ed. MOST Cooperation, ch.4. 2011.
- [33] Styła S., Walusiak S., Pietrzyk W. Computer simulation possibilities in modelling of ignition advance angle control in motor and agricultural vehicles. *Teka Komisji Motoryzacji Energetyki i Rolnictwa*, 8, pp. 231-240, 2008.
- [34] Sumorek, A., & Buczaj, M. New elements in vehicle communication 'media oriented systems transport' protocol. *Teka Komisji Motoryzacji i Energetyki Rolnictwa*, 12 (1), 275-279, 2012.
- [35] Zhou, Z., Lee, J., Berger, M. S., Park, S., & Yan, Y. Simulating TSN traffic scheduling and shaping for future automotive Ethernet. *Journal of Communications and Networks*, 23 (1), 53-62, 2021.
- [36] Douss, A. B. C., Abassi, R., & Sauveron, D. State-of-the-art survey of in-vehicle protocols and automotive Ethernet security and vulnerabilities. *Mathematical Biosciences and Engineering*, 20 (9), 17057-1709, 2023.
- [37] Pamuk A., Sakarya U. Havacılıktaki Video Veri Yolu Standardının Akıllı Taşıtlar İçin Kullanım Olasılığı Üzerine Bir İnceleme, *KONJES*, 2023.
- [38] Kvaser. Comparing CAN FD with Classical CAN. Kvaser Technical Documents. 2016.
- [39] IEEE Standards Association. Transitioning to automotive Ethernet. *IEEE Standards*. 2019.

