T.C.
MUĞLA SITKI KOÇMAN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRONİK VE BİLGİSAYAR EĞİTİMİ
ANABİLİM DALI

ZAMAN TETİKLEMELİ DENETLEYİCİ ALAN AĞI İÇİN MATRİS ÇEVRİMİ TASARIMI VE SİMÜLASYON MODELİ GELİŞTİRİLMESİ

YÜKSEK LİSANS TEZİ

SÜLEYMAN HALİL TEMEL

AĞUSTOS 2014
MUĞLA

S. H. TEMEL ELEKTRONİK VE BİLGİSAYAR EĞİTİMİ ANABİLİM DALI
YÜKSEK LİSANS TEZİ MUĞLA 2011

T.C.
MUĞLA SITKI KOÇMAN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTRONİK VE BİLGİSAYAR EĞİTİMİ
ANABİLİM DALI

ZAMAN TETİKLEMELİ DENETLEYİCİ ALAN AĞI İÇİN MATRİS ÇEVRİMİ TASARIMI VE SİMÜLASYON MODELİ GELİŞTİRİLMESİ

YÜKSEK LİSANS TEZİ

SÜLEYMAN HALİL TEMEL

AĞUSTOS 2014
MUĞLA

MUGLA SITKI KOÇMAN ÜNİVERSİTESİ

Fen Bilimleri Enstitüsü

TEZ ONAYI

SÜLEYMAN HALİL TEMEL tarafından hazırlanan **ZAMAN TETİKLEMELİ DENETLEYİCİ ALAN AĞI İÇİN MATRİS ÇEVRİMİ TASARIMI VE SİMÜLASYON MODELİ GELİŞTİRİLMESİ** başlıklı tezin, 11/08/2014 tarihinde aşağıdaki jüri tarafından Elektronik ve Bilgisayar Eğitimi Anabilim Dalı'nda yüksek lisans derecesi için gerekli şartları sağladığı oybirliği/oyçokluğu ile kabul edilmiştir.

TEZ SINAV JURİSİ

Yrd. Doç. Dr. Osman ÖZKARACA **(Jüri Başkanı)** İmza:

Bilişim Sistemleri Mühendisliği Anabilim Dalı,
Muğla Sıtkı Koçman Üniversitesi, Muğla

Doç. Dr. Mahmut TENRUH **(Danışman)** İmza:

Elektronik ve Bilgisayar Eğitimi Anabilim Dalı,
Muğla Sıtkı Koçman Üniversitesi, Muğla

Yrd. Doç. Dr. Gürcan ÇETİN **(Üye)** İmza:

Bilişim Sistemleri Mühendisliği Anabilim Dalı,
Muğla Sıtkı Koçman Üniversitesi, Muğla

ANA BİLİM DALI BAŞKANLIĞI ONAYI

Yrd. Doç. Dr. Serkan BALLI İmza:

Elektronik ve Bilgisayar Eğitimi Anabilim Dalı Başkan Vekili,
Muğla Sıtkı Koçman Üniversitesi, Muğla

Doç. Dr. Mahmut TENRUH İmza:

Danışman , Elektronik ve Bilgisayar Eğitimi Anabilim Dalı,
Muğla Sıtkı Koçman Üniversitesi, Muğla

Savunma Tarihi: 11/08/2014

Tez çalışmalarım sırasında elde ettiğim ve sunduğum tüm sonuç, doküman, bilgi ve belgelerin tarafımdan bizzat ve bu tez çalışması kapsamında elde edildiğini; akademik ve bilimsel etik kurallarına uygun olduğunu beyan ederim. Ayrıca, akademik ve bilimsel etik kuralları gereği bu tez çalışması sırasında elde edilmemiş başkalarına ait tüm orijinal bilgi ve sonuçlara atıf yapıldığını da beyan ederim.

Süleyman Halil TEMEL

11/08/2014

ÖZET

ZAMAN TETİKLEMELİ DENETLEYİCİ ALAN AĞI İÇİN MATRİS ÇEVRİMİ TASARIMI VE SİMÜLASYON MODELİ GELİŞTİRİLMESİ

Süleyman Halil TEMEL

Yüksek Lisans Tezi

Fen Bilimleri Enstitüsü

Elektronik ve Bilgisayar Eğitimi Anabilim Dalı

Danışman: Doç. Dr. Mahmut TENRUH

Ağustos 2014, 89 sayfa

Denetleyici Alan Ağı (CAN: Controller Area Network) ilk olarak araç içi, Elektronik Kontrol Üniteleri (ECU) arası haberleşme ihtiyacını gidermek ve aynı zamanda kablolama karmaşıklığını ortadan kaldırmak amacıyla geliştirilmiştir. CAN protokolü, düşük maliyeti ve yüksek performansı sayesinde geniş uygulama alanları bulmuştur. CAN protokolünde iletişim genellikle olay tetiklemelidir. Mutlak gerçek zamanlı bir işlemin gerçekleştirilmesinde verinin gerekli zaman sınırı içerisinde iletilmesi şarttır. Bu zaman şartının karşılanamaması halinde sistemde ciddi problemler ortaya çıkabileceği için, TTCAN (TTCAN: Time-triggered CAN) protokolü kullanılmaya başlanmıştır. TTCAN protokolünde, iletişim zaman tetiklemeli olarak gerçekleştirilmektedir. TTCAN periyodik olarak üretilen mesajların iletilmesinde, Matris Çevrimi (Matrix Cycle) kullanır. Matris Çevrimi tasarlanırken, kullanılan yöntemlerden birisi Azaltılmış Matris Çevrimi yöntemidir.

Bu çalışmada Azaltılmış Matris Çevrimi yöntemiyle, PSA Benchmark mesaj seti için Matris Çevrimi geliştirilmesi amaçlanmıştır. Bu yöntem kullanılarak geliştirilen Matris Çevriminde, mesajlar orijinal periyot değerleriyle matris içerisine sorunsuz bir şekilde yerleştirilebilmiştir. Bu sayede mesajlar, gecikme olmadan iletilebilmiştir. Geliştirilen sistemle elde edilen performans gelişmesini incelemek için, simülasyon modelleri oluşturulmuştur. Önerilen yöntemin uygulandığı TTCAN modelinin yanı sıra, elde edilen performansın karşılaştırılması amacıyla, standart CAN modeli geliştirilmiştir. Elde edilen simülasyon sonuçları, geliştirilen TTCAN sisteminin, maksimum iletim gecikmesi değerlerinde, en düşük öncelikli mesaj için % 93, en yüksek öncelikli mesaj için ise % 25 oranında performans gelişmesi sağladığını göstermiştir.

Anahtar Sözcükler: Denetleyici Alan Ağı (CAN), Zaman Tetiklemeli CAN, PSA Mesaj Seti, Azaltılmış Matris Çevrimi, Mesaj Planlama

ABSTRACT

DESIGNING MATRIX CYCLE AND DEVELOPING SIMALATION MODEL FOR TIME TRIGGERED CONTROLLER AREA NETWORK

Süleyman Halil TEMEL

Master of Science (M.Sc.)

Graduate School of Natural and Applied Sciences

Department of Electronics and Computer Education

Supervisor: Assoc. Prof. Dr. Mahmut TENRUH

August 2014, 89 pages

Controller Area Network (CAN) was initially developed to provide in-vehicle communication among Electronic Control Units (ECUs) and also to remove complexity of wiring. CAN protocol has a wide application area due to its low-cost and high performance. In CAN protocol, communication is generally event-triggered. In a hard real-time task, the time requirement of a message delivery is critical since the messages have to be delivered in their deadlines. In such tasks, missing a deadline may cause a major failure in the system. Because of this, TTCAN (Time-triggered CAN) has been developed. In TTCAN, communication is realized in a time-triggered manner. TTCAN uses a Matrix Cycle in communication of periodical messages. One way of designing a Matrix Cycle is the use of Reduced Matrix Cycle method.

In this study, it is aimed to develop a Matrix Cycle for PSA Benchmark with Reduced Matrix Cycle method. In the Matrix Cycle developed by this method, messages can be placed with their original period values. Thus, messages can be transmitted without delays. In order to investigate the performance improvement obtained with the developed system, some simulation models have been designed. In addition to TTCAN model, another model with standard CAN protocol has also been developed for performance comparison. The simulation results have shown that the developed TTCAN system provides % 93 performance improvement for the lowest priority message and % 25 for the highest priority message for maximum delay values.

Keywords : CAN, TTCAN, PSA Message Set, Reduced Matrix Cycle, Message Scheduling

ÖNSÖZ

Bu tezin hazırlanmasında emeği ve katkılarından dolayı danışman hocam sayın Doç.Dr.Mahmut TENRUH'a, tezin bitirilmesi aşamasında sağladığı kolaylıklar ve anlayış dolayısıyla sevgili eşim Eylem BALCI TEMEL'e, kızlarım Deniz ve Emek'e teşekkür ederim.

İÇİNDEKİLER

ÖNSÖZ ..... **vi**
İÇİNDEKİLER ..... **vii**
ÇİZELGELER DİZİNİ ..... **x**
ŞEKİLLER DİZİNİ ..... **xi**
SEMBOLLER VE KISALTMALAR DİZİNİ ..... **xiii**
1. GİRİŞ **1**
1.1. Amaç ve Kapsam 1
1.2. Kaynak Özetleri 3
2. MALZEME VE YÖNTEM **7**
2.1. Gerçek Zamanlı Sistemler 7
2.2. Denetleyici Alan Ağı (CAN) 8
2.2.1. CAN sistem özellikleri 8
2.2.2. CAN ve OSI/ISO modeli 9
2.2.2.1. Fiziksel katman *10*
2.2.2.2. Ortam erişim denetim alt katmanı *11*
2.2.2.3. Mantıksal bağlantı denetimi ve mesaj filtreleme *13*
2.2.2.4. CAN uygulama katmanı *13*
2.2.3. Paket biçimleri 14
2.2.3.1. Veri paketi *14*
2.2.3.2. Uzak paket *17*
2.2.3.3. Hata paketi *17*
2.2.3.4. Aşırı yük paketi *18*
2.2.4. Hata denetimi 19
2.2.5. Hata denetim mekanizmaları 19
2.2.5.1. Dönüşsel artıklık denetimi *19*
2.2.5.2. Paket biçimi denetimi *19*
2.2.5.3. Onay hata denetimi *20*
2.2.5.4. Bit izleme *20*

2.2.5.5. Bit doldurma *20*
2.2.6. CAN denetleyicilerinin hata durumları 20
2.2.6.1. Hata aktif *21*
2.2.6.2. Hata pasif *21*
2.2.6.3. Hat kapalı *22*
2.2.7. CAN hat uzunluğundaki sınırlamalar ve zamanlama 22
2.2.8. CAN bit zamanlaması 23
2.2.9. Yayılım gecikmesi 24
2.2.10. Eş zamanlama (Senkronizasyon) 27
2.2.11. CAN uygulamalarının sınıflandırılması 28
2.2.12. CAN denetleyicilerinin sınıflandırılması 28
2.2.12.1. Süzme ve tamponlamaya göre sınıflandırma *28*
2.2.12.2. Tanıtıcı biçimine göre sınıflandırma *29*
2.2.12.3. Çip konfigürasyonuna göre sınıflandırma *30*
2.3. Zaman Tetiklemeli Denetleyici Alan Ağı (TTCAN) 30
2.3.1. Matris çevrimi 33
2.3.1.1. Özel zaman pencereleri *34*
2.3.1.2. Kararlaştırma zaman pencereleri *35*
2.3.1.3. Serbest zaman pencereleri *35*
2.3.2. Global zaman 35
2.3.3. Referans mesajı 36
2.3.4. Çevrim süresi 37
2.3.5. Mesajların gönderilmesi ve alınması 37
2.3.6. Gönderilme penceresi 37
2.3.7. Zaman yöneticisinin hata toleransı 38
2.3.8. CAN ve TTCAN karışık istasyonlar 38
2.3.9. TTCAN uygulamaları 39
2.3.1.1. Otomobillerde kullanımı *39*
2.3.1.1. Endüstriyel uygulamalar *39*
2.4. Esnek Veri Oranlı Denetleyici Alan Ağı (CAN-FD) 39
2.4.1. Temel kavramlar 40
2.4.2. Paket biçimi 42

2.4.2.1. Veri uzunluğu kodu *43*
2.4.2.2. CRC alanı *44*
2.4.2.3. CRC dizisi *44*
2.4.2.4. CRC sınırlayıcı *45*
2.4.2.5. Onay alanı *46*
2.4.2.6. Onay yeri *46*
2.4.2.7. Onay sınırlayıcı *46*
2.4.2.8. Paket sonu *46*
3. TASARIM VE UYGULAMA **47**
3.1. PSA Benchmark 47
3.2. PSA Benchmark Mesaj Seti Kullanılarak Simülasyon Modellerinin Tasarlanması 48
3.2.1. Denetleyici alan ağı için model tasarlanması 49
3.2.2. Zaman tetiklemeli denetleyici alan ağı için model tasarlanması 49
3.2.3. Matris çevrimi tasarımı 52
4. BULGULAR VE İRDELEME **56**
4.1. Veri yolu Sonuçlarının Değerlendirilmesi 56
4.2. TTCAN Modelinin Simülasyonu İle Elde Edilen Sonuçlar 59
4.2.1. 1 Mbps hızında elde edilen sonuçlar 59
4.2.2. 500 kbps hızında elde edilen sonuçlar 63
4.2.3. 250 kbps hızında elde edilen sonuçlar 66
4.2.3. CAN ve TTCAN modellerindeki mesaj iletim gecikmelerinin hat iletim hızlarına göre incelenmesi 69
5. SONUÇLAR VE ÖNERİLER **76**
KAYNAKLAR **78**
EKLER **82**
EK A. Network II.5 Programı Ara Yüzü 82
EK B. Modelleme Kodları Örneği 83
EK C. Simülasyon Sonuçları Örneği 87
ÖZGEÇMİŞ **89**

ÇİZELGELER DİZİNİ

Çizelge 2.1 CAN için önerilen bit hızları…….. 11

Çizelge 2.2. Standart ve genişletilmiş paketlerde bit sayıları……15

Çizelge 2.3. CAN sisteminde yayılım gecikmesi örneği……26

Çizelge 2.4. Bit hızlarına göre kablo uzunlukları……... 27

Çizelge 2.5. Veri kodu uzunluğu ile tanımlanan bayt sayıları……43

Çizelge 3.1. PSA mesaj seti ve gönderen birimler…… 48

Çizelge 3.2. Matris çevrimi değerleri……..… 53

Çizelge 4.1. CAN-TTCAN hat kullanım yüzdeleri……56

Çizelge 4.2. 1 Mbps hat iletim hızında TTCAN ve CAN sistemleri simülasyon değerleri……..59

Çizelge 4.3. 500 kbps hat iletim hızında TTCAN ve CAN sistemleri simülasyon değerleri……..63

Çizelge 4.4. 250 kbps hat iletim hızında TTCAN ve CAN sistemleri simülasyon değerleri…….…66

Çizelge 4.5. Tüm hat iletim hızlarında CAN-TTCAN modelleri gecikme oranları……69

Çizelge 4.6. TTCAN modelinde tüm mesajların, tüm hat iletim hızlarındaki mesaj gecikmeleri……. 70

Çizelge 4.7. CAN modelinde tüm mesajların, tüm hat iletim hızlarındaki maksimum gecikmeleri……..……72

Çizelge 4.8. CAN modelinde tüm mesajların, tüm hat iletim hızlarındaki minimum gecikmeleri……. 73

Çizelge 4.9. CAN modelinde tüm mesajların, tüm hat iletim hızlarındaki ortalama gecikmeleri……. 74

Çizelge 4.10. CAN modelinde mesajların ortalama iletim gecikmelerinin hat iletim hızlarına göre oranları……..…….. 75

ŞEKİLLER DİZİNİ

Şekil 2.1. ISO/OSI modeli ve CAN katmanları……...……...9
Şekil 2.2. CAN yapılandırması ve fiziksel hat bağlantısı……... 10
Şekil 2.3. CAN istasyonları arasında bit esaslı karşılaştırma……. 12
Şekil 2.4. Standart ve genişletilmiş veri paketleri…….. 15
Şekil 2.5. Paketler arası boşluk…….. 17
Şekil 2.6. Hata paketi……..… 17
Şekil 2.7. Aşırı yük paketi…….. 18
Şekil 2.8. İstasyon durum geçişleri……21
Şekil 2.9. Zaman dilimi ve bit zamanı……23
Şekil 2.10. Bit zamanı bölümleri ve örnekleme noktası……... 24
Şekil 2.11. CAN sisteminde sinyal gecikmesi……..25
Şekil 2.12. İstasyonlar arası yayılım gecikmesi……26
Şekil 2.13. CAN denetleyicilerin sınıflandırılması……...…28
Şekil 2.14. Zaman tetiklemeli ağ yapısı……....31
Şekil 2.15. CAN-TTCAN ağ modelleri……32
Şekil 2.16. TTCAN matris çevrimi……...…33
Şekil 2.17. Temel çevrim……..34
Şekil 2.18. CAN-FD paket biçimi…… 41
Şekil 2.19. CAN-FD paketi kontrol alanı……..… 42
Şekil 2.20. CAN-FD standart ve genişletilmiş veri paketleri……... 42
Şekil 2.21. CRC alanı……..…. 44
Şekil 2.22. CAN-FD paketi onay alanı……..… 46
Şekil 3.1. PSA ağı……...47
Şekil 3.2. Simülasyon programında tasarlanan CAN modeli……. 50
Şekil 3.3. Simülasyon programında tasarlanan TTCAN modeli…… 51
Şekil 3.4. Tasarlanan matris çevrimi…….. 54
Şekil 3.5. Mesaj pencerelerinin zamanlama değerleri……...…. 54
Şekil 4.1. CAN-TTCAN hat kullanım oranları…….. 57
Şekil 4.2. CAN ve TTCAN sistemlerinde hat erişim gecikmeleri……..57

Şekil 4.3. CAN sisteminde ortalama ve maksimum erişim gecikmelerinin logaritmik ölçekte gösterimi……...…58
Şekil 4.4. 1 Mbps veri iletim hızında TTCAN, CAN maksimum ve ortalama mesaj iletim gecikmeleri değerleri……61
Şekil 4.5. 1 Mbps veri iletim hızında TTCAN ve CAN minimum mesaj iletim gecikmesi değerleri……..…62
Şekil 4.6. 500 kbps veri iletim hızında TTCAN, CAN maksimum ve ortalama mesaj iletim gecikmeleri değerleri……. 64
Şekil 4.7. 500 kbps veri iletim hızında TTCAN ve CAN minimum mesaj iletim gecikmesi değerleri…….....65
Şekil 4.8. 250 kbps veri iletim hızında TTCAN, CAN maksimum ve ortalama mesaj iletim gecikmeleri değerleri……. 67
Şekil 4.9. 250 kbps veri iletim hızında TTCAN ve CAN minimum mesaj iletim gecikmesi değerleri…….68
Şekil 4.10. TTCAN modelinde tüm mesajların, tüm hat iletim hızlarındaki gecikmeleri……. 70
Şekil 4.11. CAN modelinde tüm mesajların, tüm hat iletim hızlarındaki maksimum gecikmeleri……. 72
Şekil 4.12. CAN modelinde tüm mesajların, tüm hat iletim hızlarındaki minimum gecikmeleri……..73
Şekil 4.13. CAN modelinde tüm mesajların, tüm hat iletim hızlarındaki ortalama gecikmeleri……. 74

SEMBOLLER VE KISALTMALAR DİZİNİ

Δ	Hat Gecikmesi
C_m	Maksimum Mesaj Boyutu
$t_{denetleyici}$	CAN Denetleyicisinin Yayılım Gecikmesi
$t_{gönderici}$	Göndericinin Yayılım Gecikmesi
Rx	Mesajın alınması (Receiving)
Tx	Mesajın iletimi (Transmitting)
ACK	Onaylama (Acknowledge)
BC	Temel Çevrim (Basic Cycle)
CAL	CAN Uygulama Katmanı (CAN Application Layer)
CAN	Denetleyici Alan Ağı (Controller Area Network)
CiA	Otomasyonda CAN (CAN in Automation)
CRC	Dönüşsel Artıklık Denetimi (Cyclic Redundancy Check)
CSMA/CD	Çarpışma Denetimli Taşıyıcı Algılamalı Çoklu Erişim (Carrier Sense Multiple Access with Collision Detection)
CSMA/CD-CR	Çarpışma Çözümlemesi ile Çarpışma Denetimli Taşıyıcı Algılamalı Çoklu Erişim (Carrier Sense Multiple Access with Collision Detection)
DLC	Veri Uzunluğu Kodu (Data Length Code)
ECU	Elektronik Denetleme Birimi (Electronic Control Units)
EOF	Paket Sonu (End Of Frame)
EZB	Eş Zamanlama Bölümü
FB	Faz Bölümü
HD	Hamming Uzaklığı (Hamming Distance)
ID	Tanıtıcı (Identifier)
IDE	Tanıtıcı Uzantısı (Identifier Extension)
ISO	Uluslararası Standardizasyon Kuruluşu (International Organization for Standardization)
INT	Ara Alan (Intermission Field)
J	Mesajlardaki Sapma (Jitter)
L	Birimler Arası Hat Uzunluğu
LLC	Mantıksal Bağlantı Denetimi (Logical Link Control)

MAC	Ortam Erişim Denetimi (Medium Access Control)
MC	Matris Çevrim (Matrix Cycle)
NRZ	Sıfıra Dönüşü Olmayan (Non-Return to Zero)
NTU	Ağ Zaman Birimi (Network Time Unit)
OSI	Açık Sistem Bağlantısı (Open Systems Interconnection)
PCAL	Taşınabilir CAN Uygulama Katmanı (Portable CAN Application Layer)
PLC	Programlanabilir Mantık Denetleyiciler (Programmable Logic Control)
R	Mesaj bit hızı (Rate)
RTR	Uzak İletim İsteği (Remote Transmit Request)
SAE	Otomotiv Mühendisleri Derneği (Society of Automotive Engineers)
SDS	Akıllı Dağıtılmış Sistem (Smart Distiributed System)
SOF	Paket Başlangıcı (Start Of Frame)
SRR	Uzak İsteği Değiştirme (Substitude Remote Request)
TTCAN	Zaman Tetiklemeli Denetleyici Alan Ağı (Time Triggered Controller Area Network)
TDMA	Zaman Bölmeli Çoklu Erişim (TDMA)
TTP	Zaman Tetiklemeli Protokol (TTP)
Tq	Zaman Dilimi (Time Quantum)
VAN	Araç İçi Ağ (Vehicle Area Network)
Y.B.	Yayılım Bölümü

1. GİRİŞ

1.1 Amaç ve Kapsam

Otomobil endüstrisinde araç içi haberleşmeye artan talep CAN (CAN: Controller Area Network) ve diğer araç içi haberleşme ağlarının gelişmesine sebep olmuştur. Burada esas amaç, araç içerisindeki elektrik tesisatındaki karmaşıklığın azaltılmasıdır. Araç içerisinde kullanılan kablo uzunluğu toplam 2 km'ye ulaşmakta, ağırlığı ise 100 kg'ı aşmaktadır (Lawrenz, 1997). Bu karmaşık yapı, tesisatın döşenmesini zorlaştırmakta, bakım ve onarım işlemlerini güçleştirerek maliyetin artmasına sebep olmaktadır. Otomobil endüstrisiyle ilgili şirketler, araç içi haberleşme protokolleri geliştirmek için 1980'lerde araştırma çalışmalarına başlamışlardır. Değişik ihtiyaçlar göz önünde tutularak yapılan çalışmalar sonunda değişik üreticiler tarafından değişik haberleşme protokolleri geliştirilmiştir. Bunlardan bazıları CAN, VAN, J1850 ve CCD protokolleridir.

Bu protokollerden en önde gelenleri arasında Robert Bosch GmbH tarafından geliştirilen Denetleyici Alan Ağı (CAN) protokolü görülmektedir. Bosch, bu protokolün geliştirilmesi ve silikon entegre üretimi için üretici firma Intel ile işbirliği yapmış ve Autobus protokolüne dayalı ilk standart silikon CAN ürünü 1989 yılında Intel tarafından üretilmiştir. Diğer oto içi haberleşme ağları arasında CAN özellikle Avrupa'da en yaygın kullanılan haberleşme ağı olmuştur ve ilk kez 1990'ların başında otomobillerde kullanılmaya başlanmıştır (Anonim1, 1998). CAN kullanımı araç içinde bulunan bütün elektrikli ve elektronik cihazların bir tek haberleşme hattı üzerinden kontrol edilmesini ve elektrik tesisatındaki karmaşık yapının ortadan kaldırılmasını sağlamıştır. CAN kullanımı, geniş uygulama alanı, yüksek performansı, düşük maliyeti ve çok sayıda yarı-iletken üreticisi tarafından üretilmesi nedeniyle sadece otomotiv endüstrisiyle sınırlı kalmamış, diğer endüstri alanlarında da yaygın olarak kullanılmaya başlanmıştır (Anonim1, 1999).

Endüstriyel uygulamalarda iletişim protokolleri üst seviye ve alt seviye iletişim ağlarından meydana gelir. Üst seviye, fabrika hattı *(factory-bus)* olarak bilinir ve yönetim amaçlı kullanılır, alt seviye ise üretim hattı *(field-bus)* olarak bilinir ve işlemciler arası, algılayıcı – aktüatör arası iletişim için kullanılır.

Silikon teknolojisinin hızlı gelişimi daha akıllı algılayıcı ve aktüatörlerin dağıtılmış kontrol sistemlerinde kullanılmasını sağlamıştır. Bu sistemlerin kullanımının artması, sistemler arası iletişimin sağlanması ve iletişim protokollerinin standartlaştırılması ihtiyacını ortaya çıkarmış ve LON, Interbus-S, FIP, Profibus, P-NET, Bitbus, EIB, SDLC gibi çeşitli üretim hattı protokolleri geliştirilmiştir (Lawrenz,1997).

Bu çalışmada esas olan CAN protokolünde iletişim genellikle olay tetiklemelidir. Mutlak gerçek zamanlı bir işlemin gerçekleştirilmesinde verinin gerekli zaman sınırı içerisinde iletilmesi şarttır. Bu zaman şartının karşılanamaması halinde sistemde ciddi problemler ortaya çıkabilir. CAN sistemlerinde özellikle düşük öncelikli mesajlar için, mesaj gecikmesinde bir üst sınır önerilemeyebilir. Bu nedenle TTCAN (TTCAN: Time-triggered CAN) protokolü geliştirilmiştir. Bu standart OSI referans modelinin oturum katmanına karşılık gelir ve mevcut CAN protokolüne Zaman Bölmeli Çoklu Erişim *(TDMA-Time Division Multiple Access)* mimarisi ekler. Bu mimaride senkronize zamanlama, her istasyon tarafından yürütülür. Periyodik olarak iletilen referans mesajı ile ortak zaman kullanılır. Bu sayede bütün istasyonlar senkronize çalışır. Her istasyon global zamanın bir kopyasını bir sayaçta tutar ve ağ zaman biriminde *(NTU-Network Time Unit)* değeri bir arttırılır (Leen ve Heffernan, 2002). TTCAN protokolünde, iletişim zaman tetiklemeli olarak gerçekleştirilmektedir ve mesajlar belirli zaman pencerelerinde iletilir. Tüm iletim dizisi Matris Çevrimi (MC-Matrix Cycle) olarak ifade edilir ve periyodik olarak tekrar eder. Matris çevrimi tasarlanırken, kullanılan yöntemlerden birisi Azaltılmış Matris Çevrimi yöntemidir (Schmidt ve Schmidt, 2007; Tenruh, 2011)

Bu çalışmada Azaltılmış Matris Çevrimi yöntemiyle, PSA Benchmark mesaj seti için Matris Çevrimi geliştirilmesi ve TTCAN modeline uygulanması, TTCAN ve CAN modelleri arasında performans karşılaştırması yapılması amaçlanmıştır. Bu yöntem kullanılarak geliştirilen Matris Çevriminde, mesajlar orijinal periyot değerleriyle matris içerisine sorunsuz bir şekilde yerleştirilebilmiştir.

Bu sayede mesajlar, gecikme olmadan iletilebilmiştir. Geliştirilen matris çevrimi, simülasyon modeli üzerinde uygulanmış ve elde edilen simülasyon sonuçlarıyla sistem performansı incelenmiştir. Geliştirilen TTCAN modelinde elde edilen performans artışını incelemek için bir standart CAN modeli kullanılmış ve sonuçlar karşılaştırılmıştır. Elde edilen sonuçlara göre geliştirilen TTCAN modelinde mesaj gecikmelerinde önemli derecede performans gelişmesi sağlandığı görülmüştür.

Bu tezde birinci bölümde tez konusunun genel tanıtımı yapılmıştır ve Zaman Tetiklemeli Denetleyici Alan Ağları ile ilgili daha önce yapılmış olan ilgili çalışmalara yer verilmiştir. İkinci bölümde Denetleyici Alan Ağı, Zaman Tetiklemeli Denetleyici Alan Ağı ve Esnek Veri Oranlı Denetleyici Alan Ağına ilişkin teknik bilgiler verilmiştir. Üçüncü bölümde, PSA benchmark mesaj seti ve bu benchmark kullanılarak modellerin geliştirilmesi için yapılan çalışmalar, geliştirilen modeller hakkında bilgi verilmiştir. Dördüncü bölümde modelleme ve simülasyon çalışmaları sonucunda elde edilen veriler çizelgeler ve şekiller yardımıyla karşılaştırılarak sistemde elde edilen performans gelişmeleri incelenmiştir. Beşinci bölümde, hazırlanan tezde elde edilen sonuçların genel bir değerlendirmesi yapılmıştır.

1.2. Kaynak Özetleri

Bu bölümde Denetleyici Alan Ağı, Zaman Tetiklemeli Denetleyici Alan Ağı, Esnek Veri Oranlı Denetleyici Alan Ağı, Mesaj Planlama ve PSA Benchmark konularının ele alındığı ve bu tez çalışmasında yararlanılan çalışmalar sunulmuştur.

Seri haberleşmenin daha fazla uygulama alanında kullanılmasıyla birlikte, mesaj tanıtıcılarının standartlaştırılması ihtiyacı ortaya çıkmış, bu ihtiyacı karşılamak amacıyla CAN protokolü geliştirilmiştir. Bosch (Anonim, 1991) tarafından CAN protokolünün CAN 2.0A ve CAN2.0B olmak üzere standartları belirlenmiştir.

1992 yılında kurulan Otomasyonda CAN (CiA) grubu tarafından CAN protokolünün havacılık (Stock, 1999), trenler (Anonim2, 1998), denizcilik (Anonim2, 1999) uygulama alanları ile ilgili çalışmalar yapılmıştır. Bunun yanında zirai makinelerde (Hofstee ve Goense, 1997), robot kontrol sistemlerinde (Fredriksson, 1997) kullanımına yönelik çalışmalar da yapılmıştır.

CAN protokolünün olay tetiklemeli yapısından kaynaklanan, periyodik mesajların iletilmesindeki problemleri gidermek amacıyla zaman tetiklemeli CAN (TTCAN) geliştirilmiştir. TTCAN, mevcut CAN yapısının üzerine oturum katmanı eklenerek geliştirilmiştir. TTCAN protokolünün tanıtılması ve mesaj boyutunun hesaplanması ile ilgili Leen ve Heffernan (2002) tarafından çalışmalar yapılmıştır.

Zaman tetiklemeli ve olay tetiklemeli sistemleri kontrol teorisi açısından karşılaştırmak amacıyla Albert (2004) tarafından bir çalışma yapılmıştır. Bu çalışmada dağıtılmış kontrol sistemlerinin gereksinimleri özetlenmiş, bu gereksinimlere göre her iki sistemin avantajları ve dezavantajları belirlenmiştir. Belirlenen ölçütler CAN ve TTCAN sistemlerine uygulanarak sonuçlar izlenmiş ve dağıtılmış kontrol uygulamalarında periyodik işlemler için TTCAN modelinin uygun olduğu değerlendirilmiştir.

TTCAN ağında mesaj planlama, mesajlara karşılık gelen zaman aralıklarının belirlenmesi prosedürü olarak tanımlanabilir. Etkin bir planlama tablosu tasarlamak, sistem performansını belirlemede en önemli konudur.

Matris çevrimi tasarımında, ağ bant genişliğinin etkin kullanımı *(utilization)* ve periyodik mesajın periyodunun sapması *(jitter)* gibi bazı performans ölçütleri bulunur. Ayrıca gerçek zamanlı mesajların tam zamanında iletilmesi gereksinimi gibi sınırlayıcı durumlar da bulunur. TTCAN denetleyicideki kaydedicilerin sayısı da donanım sınırları olarak göz önünde bulundurulur.(Schmidt ve Schmidt, 2005; Albert ve Hugel, 2005)

Mesaj planlaması, esas olarak en iyi çözüme, performans ölçütünün seçimine ve üzerinde durulan şartlara bağlıdır.(Albert ve Hugel, 2005)

Fonseca vd. (2001) stokastik optimizasyon algoritması kullanarak doğru planlama tablosu üretmek için bir araç tanımlamıştır. Bu yaklaşımda planlama iki evreden oluşur. Birincisi çizelgeleme evresi ve ikincisi iyileştirme evresidir. Çalışmanın temel amacı sapma *(jitter)* süresini en aza indirmektir. İlk olarak mesajların ortalama periyotlarına göre matris seti üretilir. İyileştirme evresinde, matrisler küçük sapmalarla üretilir. Zaman tetiklemeli mesajlar, özel zaman pencerelerine yerleştirilir. Bununla beraber, olay tetiklemeli mesajların gerçek zaman performansı dikkate alınmaz.

Albert ve Hugel (2005) sezgisel *(heuristic)* planlama metodu geliştirmişlerdir. Bu yaklaşımda olay tetiklemeli mesajların performansı artırılmıştır. Tepki farklılığı *(Distinctness of Reaction: DoR)* olarak isimlendirilen bir ölçü ile bir haberleşme sisteminin, asenkron harici olaylara karşı tepkisi değerlendirilmiştir. Bu çalışma, iletişim matrisi içinde iyi dağıtılmış *(well-distributed)* kararlaştırma pencereleri ile önemli performans artışı sağlandığını göstermektedir.

Bir diğer mesaj planlama aracı Naugton ve Heffernan (2005) tarafından tanıtılan akıllı plan *(Smart-Plan)* dır. Bu algoritmada mesajlar ilk temel çevrime (BC) en düşük boş zaman *(SlackTime)* değerlerine göre yerleştirilir. Bu değer, en son iletim zamanına kalan zamanı tanımlar. Devam eden temel çevrimlerde geri kalan mesajlar da pencerelere kendi boş zaman değerlerine göre yerleştirilir.

Bir diğer sezgisel planlama yaklaşımı Johansson (2004) tarafından sunulmuştur. Bu çalışmada matris çevrimi mesaj periyotlarının en küçük ortak katına (OKEK) göre oluşturulmuştur. Matris çevriminin planlanabilirliği gösterilmiş ve metot bir mekanik sistemin mesaj setine uygulanmıştır.

Sistematik mesaj planlama yaklaşımını amaçlayan bir diğer çalışmada da farklı sınırlamalar ve performans ölçütleri araştırılmış ve buna göre sistematik planlama tablosu geliştirilmiştir. (Schmidt ve Schmidt, 2007) SAE benchmark için bir örnek sunulmuştur. Çalışmada esnek planlama imkanı sağlayan, matris çevrim azaltma metodu tanıtılmıştır.

Qiao vd. (2007) planlama tablosunu iyileştirmek üzere bir genetik algoritma oluşturmayı amaçlamıştır. Bu sayede özel zaman pencerelerini küçülterek, kararlaştırma zaman pencerelerindeki iletim zamanlarını artırmıştır. Bu sayede olay tetiklemeli mesajların iletim süreleri artırılabilmiştir.

Ding (2008) genetik algoritma tabanlı metodu SAE ve PSA mesaj setlerine uygulamıştır.

Ryu (2009) tarafından yapılan bir çalışma olay tetiklemeli mesajların performansları konusuna odaklanmıştır. Çalışmada $O(n^2)$ planlama algoritması sunulmuştur. Bu algoritmada özel zaman pencerelerinin maksimum sürekliliğini en aza indirme amaçlanmıştır. Bu nedenle en kötü bekleme olay tetiklemeli mesajların azaltılmasıyla gerçekleşmiştir.

Zhu (2010) matris çevrimi oluşturmak için dört tip ikili paketleme *(bin-packing)* algoritması tanımlamıştır. Çalışmada, matris çevrimindeki temel çevrimleri belirlemek için iki strateji tanıtılır. Çalışmada ayrıca bu algoritmaların performans analizleri de verilmektedir.

Xiao (2010) tarafından yapılan bir diğer çalışmada, TTCAN ağındaki mesajların dinamik olarak kararlaştırma pencerelerinde planlanabildiği, gerçek zaman dinamik öncelik algoritması sunulur. Bu algoritma ağ gecikmesini azaltarak, haberleşmenin gerçek zaman performansını arttırmıştır.

CAN ağı en fazla 1 Mbps hat iletim hızına erişebilmektedir. Bu sınırlamanın birinci sebebi, kararlaştırma mekanizması için gerekli olan minimum bit uzunluğu ve ikincisi de en fazla 8 bayt mesaj iletilmesidir. CAN veri hızını yükseltebilmek amacıyla Esnek Veri Oranlı CAN (CAN-FD) protokolü, Bosch (Anonim, 2011) tarafından geliştirilmiş ve özellikleri tanıtılmıştır.

CAN-FD protokolünün tanıtılması (Hartwich, 2012) ve CANopen sistemler için sağladığı yeni fırsatları incelemek amacıyla (Oertel, 2012) tarafından çalışmalar yapılmıştır.

Bu tez çalışmasında kullanılan PSA Benchmark mesaj seti kullanılarak CAN ve VAN çoklu ağ yapısını içeren araç içi modelleme uygulaması (Castelpietra vd., 2000) tarafından yapılmıştır. Ayrıca SAE ve PSA Benchmarkların incelenerek yeni bir benchmark geliştirme çalışması (Mohammed ve Holou, 2010) tarafından yapılmıştır.

2.MALZEME VE YÖNTEM

2.1. Gerçek Zamanlı Sistemler (Real Time Systems)

Gerçek zamanlı sistemler, gerçek zamanlı işlemlerin yerine getirilmesini destekleyen ve zamanlama şartlarının karşılanmasını garanti eden sistemler olarak tanımlanabilir. Gerçek zamanlı sistemler, mutlak gerçek zamanlı sistemler (HRTS: Hard Real-time Systems) ve toleranslı gerçek zamanlı sistemler (SRTS:Soft Real-time Systems) olarak sınıflandırılabilir (Baba, 1996). Bir gerçek zamanlı sistemde her iki guruptaki işlemlerin karışımı bulunabilir.

Mutlak gerçek zamanlı bir işlemin gerçekleştirilmesinde verinin gerekli zaman sınırı içerisinde iletilmesi şarttır. Bu zaman şartının karşılanamaması halinde sistemde ciddi problemler ortaya çıkabilir. Diğer taraftan, toleranslı gerçek zamanlı işlemlerin yerine getirilmesinde zaman şartının yerine getirilmesi tercih edilir, ancak bunun yerine getirilememesi halinde kritik sonuçlar ortaya çıkmaz.

Gerçek zamanlı işlemler ortaya çıkış süreleri açısından periyodik ve düzensiz zaman aralıklı işlemler olarak sınıflandırılabilir. Periyodik işlemler belirli zaman aralıklarıyla yerine getirilen işlemlerdir. Bu işlemler zaman tetiklemeli işlemler *(Time-triggered)* olarak da isimlendirilebilir. Bir motorun hızını ölçen sistemin sabit zaman aralıklarıyla veri üretmesi buna örnek gösterilebilir. Düzensiz zaman aralıklı işlemler belirli bir durumun ortaya çıkması halinde yerine getirilen işlemlerdir. Bu işlemler olay tetiklemeli *(event-driven)* olarak da isimlendirilebilir. Buna örnek olarak acil durum anında ortaya çıkan bir işlem gösterilebilir. Buradan gerçek zamanlı işlemleri belirleyen iki temel özelliğin işlemlerin ortaya çıkışındaki zaman aralığı ve işlemle ilgili veri iletimindeki zaman sınırı olduğu sonucu çıkartılabilir (Tenruh, 2001).

2.2. Denetleyici Alan Ağı (CAN- Controller Area Network):

CAN ilk olarak araç içinde elektronik denetleme birimleri (ECU-Electronic Control Unit) arası gerçek zamanlı haberleşme ihtiyacını karşılamak üzere geliştirilmiştir. Modern araçlarda araç içi iletişimde, 70 adet civarında elektronik denetleme birimi ve 2500 sinyal çeşidi kullanılabilmektedir (Albert, 2004). Bu durum tesisatın kurulmasını zorlaştırmakta ve maliyeti artırmaktadır. 1980'lerden bu yana yapılan araştırmalarla, değişik üreticiler tarafından çok sayıda haberleşme protokolü geliştirilmiştir. CAN, düşük maliyeti ve yüksek performansıyla endüstriyel otomasyon uygulamalarında kabul gören bir standart (ISO-11898) haline gelmiştir.

CAN, mikrodenetleyicilerden, sensör ve aktuatörlerden oluşan güçlü haberleşme ortamı sayesinde, araç içinde bulunan bütün elektrikli ve elektronik cihazların bir tek haberleşme hattı üzerinden kontrol edilmesini ve elektrik tesisatındaki karmaşık yapının ortadan kaldırılmasını sağlamıştır. CAN otomotiv endüstrisi dışında yolcu arabaları (Wenkebach ve Reckels, 1989), kamyonlar, trenler, uçaklar (Stock, 1999), gemiler (Anonim2, 1999), zirai makineler (Hofstee ve Goense, 1997), programlanabilir mantıksal kontrol (PLC) uygulamaları, robot kontrolü (Fredriksson, 1997), akıllı motor kontrolü, akıllı algılayıcı ve uygulayıcılar, laboratuvar otomasyonu, tekstil makineleri, bina otomasyonu, asansörler, alış-veriş makineleri, ve endüstriyel otomasyon alanları gibi çok çeşitli alanlarda da kullanılmaktadır.

2.2.1. CAN Sistem Özellikleri

CAN, taşıyıcı algılamalı çoklu ortam erişim yöntemiyle birlikte çarpışma çözümlemeli bir iletişim yöntemi uygular *(CSMA/CD+CR: Carrier Sense, Multiple Access/Collision Detection with Collision Resolution).*

Veri iletişimi bütün ünitelerin bağlı bulunduğu iletişim hattı üzerinden verinin gönderilmesi ile gerçekleşir. Bütün üniteler gönderilen veriyi kontrol eder, sadece ilgili üniteler veriyi kaydeder ve kullanır.

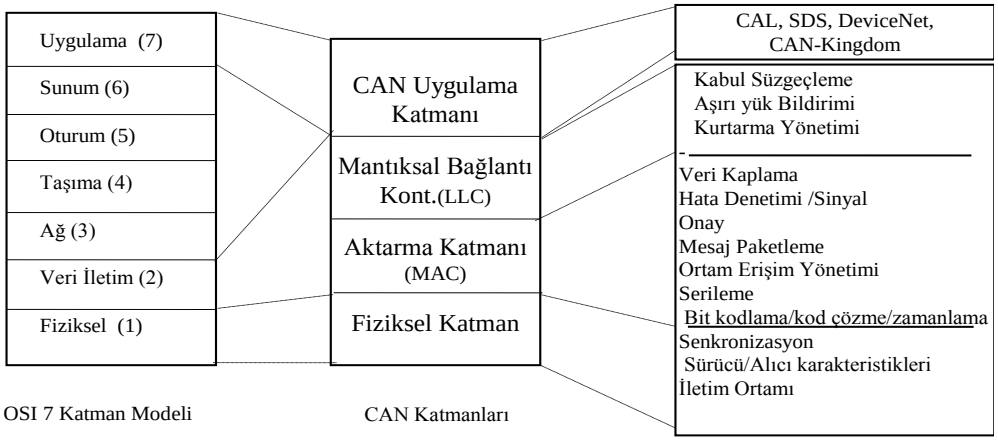
Bir kontrol ünitesi, eğer yeni bir veri ortaya çıkmışsa, bunu haberleşme hattından göndermeye başlar. İletişim genellikle olay tetiklemelidir *(event-driven).* Eğer

gerekiyorsa kontrol ünitesi içerisinde bulunan bir zamanlayıcı sayesinde, zaman tetiklemeli olarak, periyodik aralıklarla veri gönderecek şekilde de ayarlanabilir.

Olay tetiklemeli iletişimin yanında CAN aynı zamanda herhangi bir kontrol ünitesinin bir başka kontrol ünitesinden veri istemesi *(on-demand)* esasına göre de veri iletişimi sağlar.

CAN bütün iletişim sistemi üzerinde tutarlılık olması esasına göre çalışır. Bir CAN sistemi içerisinde bir veri gönderildiği zaman bütün üniteler tarafından kabul edilir veya reddedilir. Bunun sebebi sistemin sahip olduğu hata önleme mekanizmasıdır. Veri iletişimi sırasında her hangi bir kontrol ünitesi bir hata tespit ederse derhal bir hata mesajı gönderir ve bütün üniteler iletilen veriyi iptal eder. Veri, gönderici tarafından yeniden gönderilir. CAN toplam beş çeşit hata önleme mekanizmasına sahiptir. Bu nedenle hatalı bir verinin tespit edilememesi ihtimali $p < 4.7 * 10^{-11}$ olarak ifade edilebilir ve bu da 1 milyon saatlik çalışma sırasında 1 hatanın fark edilememesinden daha düşük bir ihtimaldir.

2.2.2. CAN ve ISO/OSI Modeli



Şekil 2.1. ISO/OSI modeli ve CAN katmanları

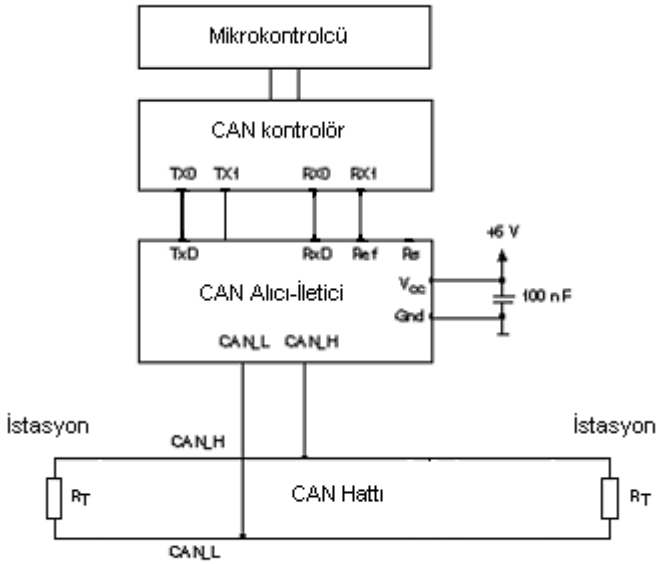
CAN modelinin en alt iki katmanı OSI modelinin en alt iki katmanıyla eşleşir. Veri bağlantı katmanı, Mantıksal Bağlantı Denetimi (LLC) ve Ortam Erişim Denetimi (MAC) olmak üzere iki alt-katmana ayrılmıştır. CAN ISO-11898 standardında belirtilmiştir. Şekil 2.1.'de (Anonim, 1991) OSI 7 katman modeliyle CAN katmanları arasındaki ilişki gösterilmiştir.

CAN mimarisi Fiziksel *(Physical)*, Mantıksal Bağlantı Denetimi (LLC) ve Ortam Erişim Denetimi (MAC) olarak üç bölüme ayrılabilir. Fiziksel katman diğer ağ teknolojilerinde olduğu gibi, mesaj kaynağındaki bitleri hedefe aktarır.

Ortam Erişim Kontrol alt-katmanı veri paketlerini kontrol eder, doğrular ve hata meydana gelmesi durumunda sinyal gönderir, hat erişimi için gerekli işlemleri gerçekleştirir. Mantıksal Bağlantı Denetimi alt-katmanı ise gelen mesajları süzer ve veri aktarımı, veri istemi, veri kurtarma, aşırı yük ikazı gibi servisler sağlar.

2.2.2.1 Fiziksel katman

CAN Ethernet'e benzer olarak yol *(bus)* topolojisine sahiptir ve mikroişlemcisi olan elektronik aygıtları birbirine bağlamak için tasarlanmıştır. Mikroişlemci ağ iletişimini kontrol eden CAN denetleyicisine bağlıdır.



Şekil 2.2. CAN yapılandırması ve fiziksel hat bağlantısı

Fiziksel bağlantı için çift burgulu kablo veya fiber optik kablo gibi farklı ortamlar kullanılabilir. Şekil 2.2.'de bir CAN ünitesinin fiziksel bağlantı örneği görülmektedir. Ortamda CAN_H ve CAN_L olarak isimlendirilen iki sinyal hattı bulunur ve mantıksal 0 (sıfır) bilgisi içeren bir bit, mantıksal 1 (bir) bilgisi içeren bir bite oranla önceliğe sahiptir. Diğer bir deyişle 0 biti 1 bitine baskındır.

Bu mekanizma, iletimin zor olduğu elektriksel ortamlarda bile güvenli bir veri aktarımı sağlar (Tenruh, 2001). Çizelge 2.1.'de CiA DS-102 tarafından önerilen fiziksel bağlantı uzaklıkları ve bit hızları gösterilmektedir.

Çizelge 2.1. CAN için önerilen bit hızları

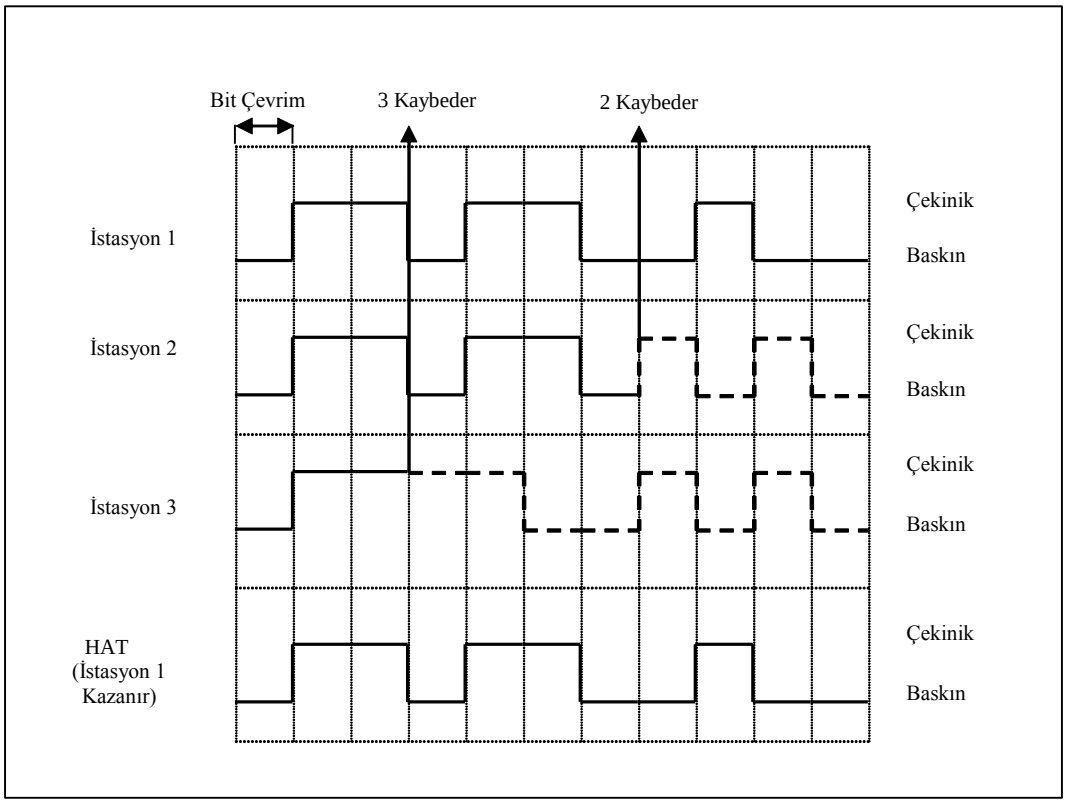
Bit Hızı	Nominal Bit Zamanı (µs)	Açıklama
1 Mbit/s	1	25 m (en fazla 40 m)
800 kbit/s	1.25	50 m
500 kbit/s	2	100 m
250 kbit/s	4	250 m
125 kbit/s	8	500 m
50 kbit/s	20	1 km
20 kbit/s	50	2.5 km
10 kbit/s	100	5 km (En düşük bit hızı)

2.2.2.2 Ortam erişim denetim (Media access control - MAC) alt katmanı

Ortam erişim denetim *(Media Access Control-MAC)* alt-katmanının en önemli fonksiyonu hat yerleşimini düzenlemektir. Hat üzerinde bulunan farklı istasyonlar aynı anda veri iletimine başlayabilir, bu durumda çarpışma meydana gelir ve veri kaybı oluşabilir. Aynı problem Ethernet'te de vardır ve çarpışma meydana geldiğinde her istasyon rastgele bir zaman aralığında tekrar veri iletimine başlar ancak bu durumda da hattın boş olduğu ve çarpışma olmayacağı kesin değildir. Teorik olarak hat erişimindeki gecikmenin bir üst sınırı yoktur ve her çarpışmadan sonra hat belirli bir süre boş kalır. Bu mekanizma CSMA-CD (Taşıyıcı algılamalı çoklu erişim yöntemiyle birlikte çarpışma denetimi) olarak adlandırılır. Çarpışmadan sonra hiçbir istasyon veri iletimine devam etmez, dolayısıyla bu mekanizmada hat erişimi yıkıcı olarak değerlendirilir.

CAN Ethernet'te kullanılan mekanizmayla benzerlikleri olan CSMA/CD ile birlikte yıkıcı olmayan Bit esaslı karşılaştırma mekanizmasını kullanır. Yüksek öncelikli mesajlar için daha az gecikme sağladığından gerçek-zamanlı sistemler için uygundur. Şekil 2.3.'te bit esaslı karşılaştırma mekanizması gösterilmiştir (Tenruh, 2001). Ethernet'ten farklı olarak CAN protokolünde iletişim hedefe-yönelik değil, veriye-

yöneliktir. Bunun anlamı mesajın belirli bir hedefe adreslenmesi değil, taşıdığı bilgiye göre tanıtılmasıdır.



Şekil 2.3. CAN istasyonları arasında bit esaslı karşılaştırma

Mesajın tanıtıcı kısmı aynı zamanda verinin önceliğini de belirler. Veri paketinin ilk kısmında bulunur ve çarpışmanın çözümlenmesinde görev alır. CAN istasyonları veri iletirken her zaman hattı görüntüler. Eğer iki istasyon aynı anda iletime başlamışsa, istasyonlardan biri baskın olan 0 biti, diğeri çekinik olan 1 biti gönderene kadar çarpışma algılanmaz. Çarpışma meydana geldiğinde, hat üzerinde 0 olan bit iletilir, 1 göndermeye çalışan istasyon 0 bilgisini görür ve bir çarpışma algılar. Çarpışmadan sonra, iletimi hemen durdurur ve alma *(receiving)* moduna geçer. 0 gönderen istasyon çarpışma algılamaz ve iletime devam eder. Bu yolla, düşük önceliğe sahip olan veriler iletilmez, en yüksek önceliğe sahip olan veri iletilir. Veri paketinin tanıtıcı kısmı eşsiz bilgidir ve iki veri paketinin aynı tanıtıcıya sahip olması imkansızdır.

Bu, hat erişim mekanizması en yüksek önceliğe sahip olan verinin iletilmesini garanti altına alır. İletime izin verilmeyen tek zaman ardışık iki veri paketi arasındaki üç bit bekleme zamanıdır.

2.2.2.3 Mantıksal bağlantı denetimi (LLC-Logical link control) ve mesaj filtreleme

Mantıksal bağlantı denetimi alt-katmanının en önemli fonksiyonlarından biri, mesajın filtrelenmesidir. Bir istasyon veriyi tanıtıcıyla birlikte iletir ve diğer istasyonlar bu veriyi alır. Mantıksal bağlantı denetimi katmanı tanıtıcıyı inceler, verinin istasyon tarafından kullanılıp kullanılmayacağına karar verir. Eğer veri kabul edilirse, bir üst katmana geçirilir. CAN denetleyicilerinin veri tanıtıcısını tutan belirli kaydedicileri vardır. Denetleyici, veriyi kabul etmeye veya etmemeye karar verdiğinde, kaydedici bir maske işlevi görür ve kabul edilen veriler geçirilir, diğerleri dışarıda bırakılır.

2.2.2.4 CAN Uygulama (Application) katmanı

8 bayttan uzun verilerin aktarılması ve alınması gibi, veri iletim katmanının temel fonksiyonlarının ötesinde servislere ihtiyaç duyan bazı CAN uygulamaları bulunmaktadır. Bu tip bir uygulama, uygulama katmanında gerçekleştirilir. Bu ihtiyaçları karşılamak için bazı organizasyonlar bir kaç uygulama katmanı geliştirmiştir (Etschberger, 1997; Schofield, 1996; Lenartsson, 1995).

Bunlardan bazıları :

CAL, CANopen, PCAL, DeviceNet, SDS ve CAN Kingdom olarak sıralanabilir.

CAL (CAN Uygulama Katmanı - CAN Application Layer)

Philips Medical System tarafından geliştirilen bir protokolü temel alır. Uygulamadan bağımsız bir katmandır ve şu anda CiA kullanıcı grubu tarafından sürdürülmektedir.

CANopen : CiA DS-301 iletişim profilinde tanımlanmış bir CAL uygulamasıdır.

PCAL (Taşınabilir CAN Uygulama Katmanı -Portable CAN Application Layer)

CAL uygulamasının başka bir örneğidir. Almanya Federal Silahlı Kuvvetleri Üniversitesi tarafından geliştirilmiştir.

DeviceNet: Endüstriyel otomasyon uygulamalarında geniş bir kullanımı vardır. CiA tarafından kabul edilmiş ve Rockwell/Allen-Bradley tarafından geliştirilmiş bir uygulama katmanıdır.

SDS (Akıllı Dağıtılmış Sistemler- Smart Distributed Systems) : Esas olarak makine kontrol uygulamalarında kullanılır, Honeywell tarafından geliştirilmiştir.

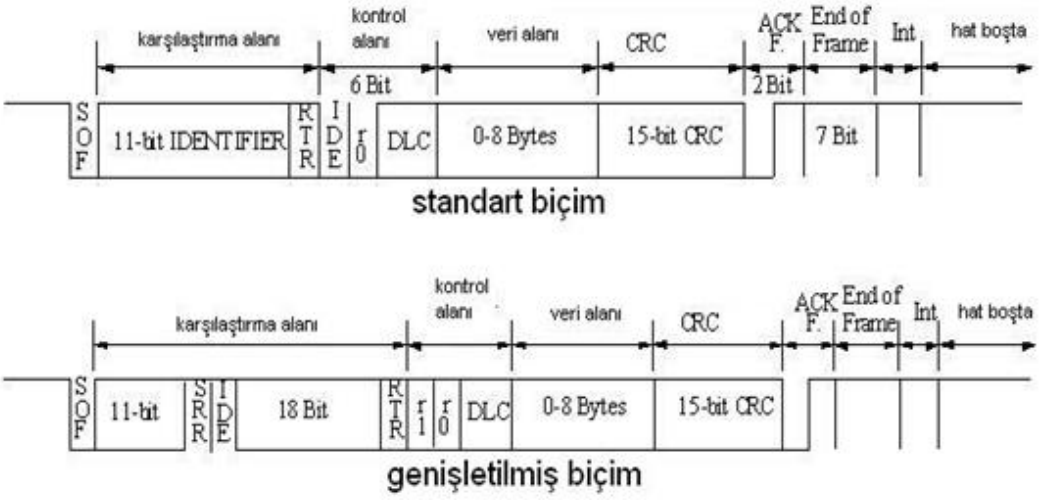
CAN Kingdom : İsveç şirketi Kvaser AB tarafından geliştirilen CiA tarafından kabul edilen bir uygulama katmanıdır.

2.2.3. Paket Biçimleri

CAN, Veri Paketi (Data Frame), Uzak Paket (Remote Frame), Hata Paketi (Error Frame) ve Aşırı yük Paketi (Overload Frame) olmak üzere 4 farklı pakete sahiptir (Anonim, 1991). CAN protokollerinin iki farklı versiyonu vardır. İlk versiyon olan CAN 1.0 da veri paketinde ve uzak pakette tanıtıcı kısmı 11 bittir. Diğer versiyon CAN 2.0 ise CAN 2.0A ve CAN 2.0B olmak üzere ikiye ayrılmıştır. CAN 2.0A versiyonu CAN 1.0 ile tam uyumludur ve aynı paket biçimlerine sahiptir. CAN 2.0B versiyonunda paket tanıtıcı kısmı 29 bit olarak belirlenmiştir. CAN 2.0B versiyonu CAN 1.0 ve CAN 2.0A biçimlerini destekler. Tanıtıcı uzunluğuna göre veri ve uzak paketler Standart ve Genişletilmiş olmak üzere ikiye ayrılır.

2.2.3.1. Veri paketi (Data frame)

Veri paketleri 7 temel alandan oluşur. Bunlar: Başlangıç Biti *(Start bit)*, Kararlaştırma Alanı *(Arbitration field)*, Kontrol Alanı *(Control field)*, Veri Alanı *(Data field)*, Dönüşsel Artıklık Denetimi Alanı *(Cyclic Redundancy Check - CRC field)*, Onay Alanı *(Acknowledge field)* ve Paket Sonu Alanı *(End of Frame field)* olarak sıralanır. Şekil 2.4.'te standart ve genişletilmiş veri paketleri görülmektedir (Anonim3, 1999). Çizelge 2.2.'de ise CAN paketlerinde bulunan bit sayıları gösterilmiştir.



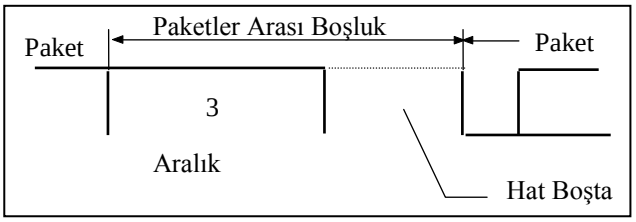
Şekil 2.4 Standart ve genişletilmiş veri paketleri

Çizelge 2.2. Standart ve genişletilmiş paketlerde bit sayıları

STANDART		GENİŞLETİLMİŞ	
Mesaj Alanı	Bit Sayısı	Mesaj Alanı	Bit Sayısı
SOF	1	SOF	1
Tanıtıcı(ID)	11	Tanıtıcı (ID)	11
RTR	1	SRR	1
IDE	1	IDE	1
r0	1	Tanıtıcı (ID)	18
DLC	4	RTR	1
Veri Alanı	0-64	r1	1
CRC Alanı	16	r0	1
ACK Alanı	2	DLC	4
Paket Sonu	7	Veri Alanı	0-64
		CRC Alanı	16
		ACK Alanı	2
		Paket Sonu	7
Toplam	44-108	Toplam	64-128
En fazla Dolgu biti	19	En fazla Dolgu biti	23
En fazla Doldurulan	127	En fazla Doldurulan	151

- **Paket Başlangıcı *(SOF)* :** Veri paketinin veya Uzak paketin başlangıcını gösterir, baskın (0) bir bit içerir.
- **Tanıtıcı *(Identifier)* :** Standart biçimde 11 bit uzunluğundadır, genişletilmiş biçimde 11 bit temel tanıtıcı ve 18 bit genişletilmiş tanıtıcı olarak iki kısımdan oluşur. İki versiyonda da tanıtıcılar en önemli bitten (MSB) başlayarak iletilir.

- **Uzak İletim İsteği *(Remote Transmission Request-RTR)*:** Tanıtıcıdan sonra gelir ve veri paketlerinde bu bit her zaman baskın (0) dır.
- **Uzak İsteği Değiştirme *(Substitute Remote Request – SRR)*:** Sadece genişletilmiş biçimde bulunur ve standart biçimdeki RTR bitinin yerine geçer. Standart veya genişletilmiş biçimi ayırt etmek için her zaman çekiniktir, tanıtıcı alanları eşit olan biçimler, hat erişmi için yarıştığında standart biçim erişim elde eder.
- **Tanıtıcı Uzantısı *(Identifier Extension-IDE)*:** Paketin hangi biçimde olduğunu gösterir. Standart biçimde her zaman baskın, genişletilmiş biçimde her zaman çekiniktir. Standart biçimde kontrol alanına, genişletilmiş biçimde kararlaştırma alanına dahildir.
- **r0, r1:** Gelecekte kullanım için ayrılmış her zaman baskın olan bitlerdir.
- **Veri Uzunluğu Kodu *(Data Length Code-DLC)*:** Kendisinden sonra gelen verinin uzunluğunu gösteren 4 bitlik alandır. Veri alanı 0 ile 8 bayt arasında değişir.
- **Veri *(Data)*:** İstasyonların ihtiyacı olan bilgiyi içerir ve en fazla 8 bayt uzunluğunda olabilir. Diğer ağ teknolojileri ile karşılaştırıldığında oldukça sınırlıdır, fakat CAN'in özel kullanım amacı olmasından dolayı, istasyonların çok büyük veri paketleri kullanması istenilen bir durum değildir, daha kısa veri paketleri tercih edilir.
- **Dönüşsel Artıklık Denetimi *(CRC)*:** Her mesaj paketinin hat üzerinden gönderdiği 15 bitlik bir bilgidir.
- **CRC Sınırlayıcı:** Verinin karşıdan alındığını bildiren bittir. Gönderici çekinik (1) seviye olarak iletir, alıcı baskın(0) seviyesine dönüştürür ve mesajın ulaştığı anlaşılır.
- **Onay *(ACK)* Sınırlayıcı:** CRC sınırlayıcı gibi önceden tanımlı, çekinik bir bittir.
- **Paket Sonu (EOF):** 7 tane çekinik bit içeren, paket sonunu belirten bilgidir.
- **Aralık (INT):** Ardışık iki veri paketini veya uzak paketi ayıran, 3 biti de çekinik olan alandır. Bu alanda istasyonun herhangi bir veri paketi veya uzak paket iletimine başlamasına izin verilmez, ancak hata veya aşırı yük paketleri iletilebilir. Aralık zamanının ve hattın boşta kaldığı zamanın toplamına paketler arası boşluk adı verilir. Aralık zamanından sonra paketler arası boşluk esnasında bir baskın bit bulunur ve diğer CAN denetleyicileri bunu paket başlangıcı olarak yorumlar. Şekil 2.5.'te paketler arası boşluk gösterilmektedir.



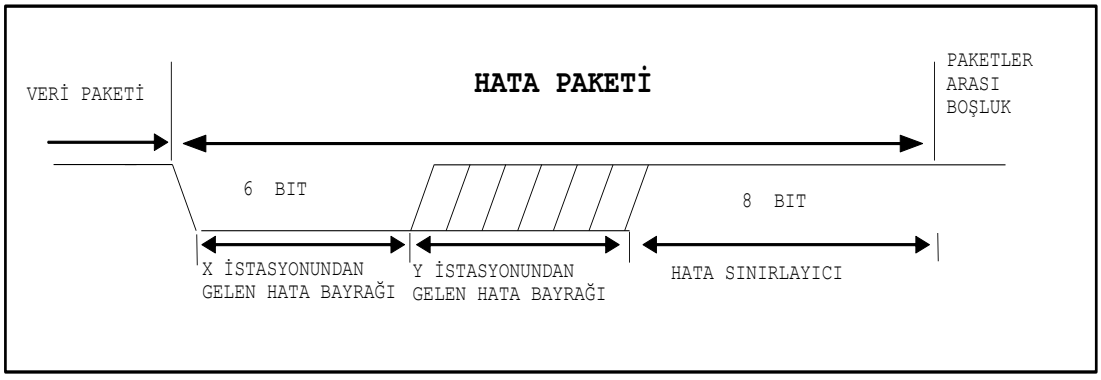
Şekil 2.5. Paketler arası boşluk

2.2.3.2. Uzak paket (Remote frame)

Uzak paketlerin veri paketlerine oldukça benzer yapıları vardır, bu yüzden iki paket birlikte değerlendirilir ve ikisine birden mesaj paketleri denir. Bir istasyon bilgi istediğinde bir uzak paket iletir ve bu paketin veri alanı yoktur yani DLC biti her zaman sıfırdır (0). Uzak pakette RTR biti her zaman çekiniktir. RTR biti kararlaştırma alanına dahil olduğu için, bir veri paketiyle bir uzak paketin hat erişimi için yarışması durumunda veri paketi RTR biti baskın olduğu için, hat erişimi kazanır.

2.2.3.3. Hata paketi (Error frame)

Hata paketi iki alandan oluşur. İlk alan farklı istasyonlardan iletilen hata bayraklarının pozisyonudur, ikinci alan ise hata sınırlayıcıdır. Şekil 2.6.'da hata paketi görülmektedir.



Şekil 2.6. Hata paketi

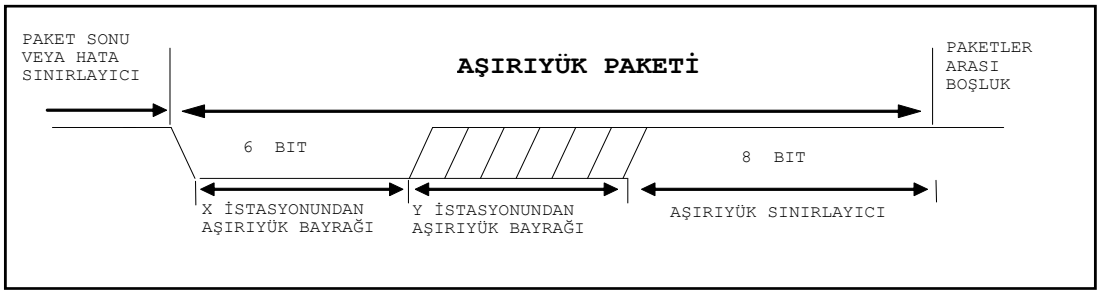
Aktif ve Pasif olmak üzere iki hata bayrağı vardır. Aktif bayrağı 6 tane baskın, pasif bayrağı 6 tane çekinik bit içerir. İkisi de bit doldurma *(bit-stuffing)* kuralını ihlal eder

ve alıcının hata algılayıp, hata sayacını artırmasına neden olurlar. İstasyon hata bulduğunda hemen bir hata bayrağı iletir, yani hatalı mesaj bitmese bile hata bayrağı başlayabilir. Bu yüzden aktif hata bayrağı mesajın geri kalan 6 bitinin üstüne yazılır ve bu durumda diğer istasyonlar dolgu hatası veya biçim hatası algılarlar ve hata bayrağı gönderirler. Bu yöntemle hata bayraklarının süper pozisyonu olarak adlandırılan sıralı 12 bit hat üzerinde görünür.

Hata sınırlayıcı alanı, hata bayrağını takip eden 8 tane ardışık bitten oluşur. Bir istasyonun hata sınırlayıcısı, devam eden diğer istasyonların hata bayrakları üzerine yazılabilir. İstasyon hat üzerinde çekinik bit görüntüleyene kadar çekinik bit gönderir. Sonra 7 tane çekinik bit daha gönderir.

2.2.3.4. Aşırı yük paketi (Overload frame)

Aşırı yük paketleri hata paketlerine yapı olarak çok benzerdir ancak biraz farklı amaçları vardır. Aşırı yük paketleri genellikle bir sonraki mesaj paketinin iletimini beklemek için gönderilirler ve hata sayacını artırmazlar.
Bir hata paketinden sonra veya aşırı yük paketinden sonra aralık zamanında aşırı yük paketi gönderilebilir. İstasyon böyle bir paketi zamana bakarak hata paketinden ayırır. Aşırı yük paketi 6 tane baskın bit içeren "aşırı yük bayrağı" ve 8 tane çekinik bit içeren "aşırı yük sınırlayıcı" alanlarından meydana gelir. Şekil 2.7.'de aşırı yük paketi görülmektedir.



Şekil 2.7. Aşırı yük paketi

Aşırı yük paketi iletilmesi üç durumda gerçekleşir:

1. Alıcının bir sonraki mesaj paketine geçmeden, güncel veriyi işlemek için biraz daha zamana ihtiyacı varsa.
2. Aralık alanının birinci veya ikinci biti baskın ise.
3. Hata sınırlayıcının veya aşırı yük sınırlayıcının sekizinci biti baskın ise.

İlk durumda aşırı yük paketi sadece bir sonraki aralık zamanının ilk bitinde başlayabilir. Son iki durumda ise, baskın bit algılandıktan sonra bir sonraki bit çevriminde başlayabilir. Bir istasyon, aşırı yük paketi alırsa, kendi aşırı yük paketini iletir ve bu şekilde hata paketinde olduğu gibi aşırı yük paketi, aşırı yük bayrakları ve sınırlayıcılarını içerir. Mesaj paketlerini bekletmek için en fazla iki aşırı yük paketi iletilebilir.

2.2.4. Hata denetimi (Error handling)

CAN güvenli bir iletişim sağlayan 5 hata denetim mekanizmasına sahiptir. Tespit edilemeyen hata oranı yaklaşık bir milyon çalışma saatinde birdir (Anonim3, 1999). Hata tespit eden herhangi bir istasyon, iletimi, paket üzerine baskın bitler yazarak durdurur. İletimi gerçekleştiren istasyon mesajı tekrar eder.

2.2.5. Hata denetim mekanizmaları

2.2.5.1. Dönüşsel artıklık denetimi (Cyclic redundancy check)

Her iletilen mesaj paketinin CRC alanına, 16-bit (CRC) kodu eklenir. Mesaj iletildiğinde CRC kodu hesaplanır ve sonuç alınan ile karşılaştırılır. Sonuçlar farklı ise alıcı hata paketi gönderir.

2.2.5.2. Paket biçimi denetimi (Frame format check)

Paketteki belirli alanların (CRC sınırlayıcı, paket sonu ve paketler arası boşluk) değerleri önceden tanımlı olmalıdır. Bu alanlardan herhangi birisi alınan pakette önceden tanımlı bir değer içermiyorsa biçim hatası meydana gelir ve hata bayrağı iletilir.

2.2.5.3. Onay hata denetimi (Acknowledgement error check)

Yayınlanan pakette çekinik bit her zaman ACK alanında iletilir. Mesaj alındığında baskın bit çekinik bit üzerine yazılır. Yani, gönderici ACK alanını ilettiği zaman bir veya daha fazla alıcı, gönderici tarafından görüntülenen ve onay olarak yorumlanan tek bir baskın bit görüntüler. Eğer gönderici bu biti görüntülemezse ACK hatası meydana gelir.

2.2.5.4. Bit izleme (Bit monitoring)

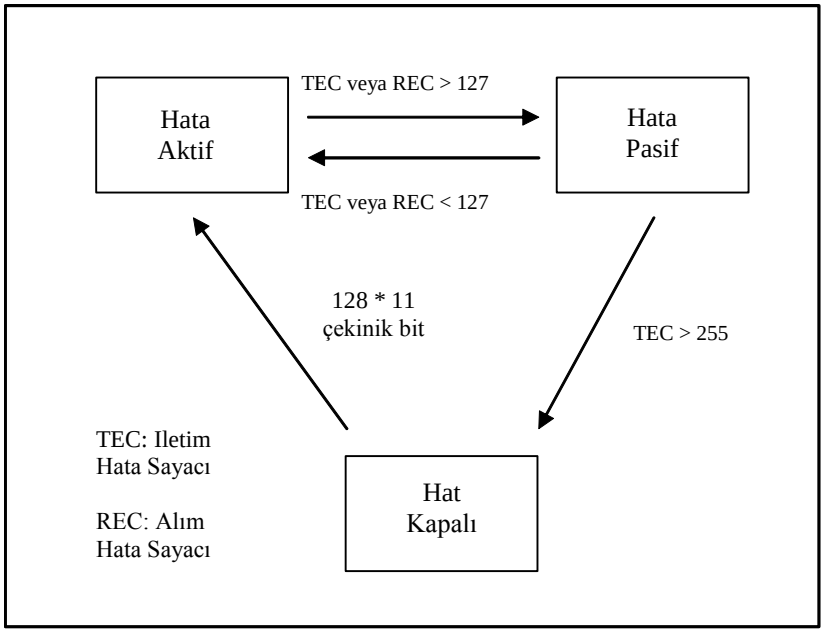
Bir mesaj iletildiğinde gönderici hattı izler, görüntülenen değer ile iletilen değeri karşılaştırır. İki değer aynı değilse, Bit hatası meydana gelir.

2.2.5.5. Bit doldurma (Bit stuffing)

CAN modelinin hata denetim mekanizması içinde bit doldurma önemli bir yer tutar. Paket Başlangıcı, kararlaştırma alanı, kontrol alanı, veri alanı ve CRC dizisi alanları sıfıra geri dönmeme *(NRZ-NonReturn to zero)* metoduna göre doldurulur. CAN sisteminde istasyonlar arası zaman senkronizasyonu bit değişimleri sırasında sağlanır. Bu nedenle, aynı seviyede en fazla beş ardışık bitin gönderilmesine izin verilir. Eğer, önceki alanda ardışık beş özdeş bit görüntülenirse, gönderici beş bitten sonra karşıt değerde bir bit ekler. Alıcı da ardışık beş özdeş bit algılarsa altıncı biti kaldırır. Eğer altıncı bit de aynı değerde ise, dolgu hatası meydana gelir.

2.2.6. CAN denetleyicilerin hata durumları

Bir denetleyici veriyi hatalı olarak iletmeye veya almaya başlarsa, bu istasyonda ve diğer istasyonlarda hata denetim mekanizması devreye girer. Hatalı veri iletimi durumunda hatalı istasyon hattı yararsız şekilde meşgul eder. Bunu engellemek için, her istasyonda, biri iletim hataları için ikincisi alım hataları için iki tane hata sayacı bulunur. Hata tespit edildiğinde istasyon uygun sayacı artırır. Şekil 2.8.'de istasyonun hata denetimi için durum geçişleri görülmektedir.



Şekil 2.8. İstasyon durum geçişleri

Farklı hata durumlarında sayacın hangi değeri alacağını belirleyen bir tablo kullanılır. İstasyon iletim veya alım işlemini başarılı bir şekilde gerçekleştirdiğinde, uygun sayaç 0 değilse azaltılır. Hata sayaçlarına göre istasyon 3 durumdan birinde olabilir: Hata aktif, hata pasif ve hat kapalı.

2.2.6.1. Hata aktif (Error active):

Bu durumda istasyon ağda tüm aktivitelere katılır ve iki hata sayacı değeri de 128'in altındadır. Ağda herhangi bir hata yoksa, tüm istasyonlar hata aktif durumundadır ve hata sayaçları 0'dır.

2.2.6.2. Hata pasif (Error passive):

Bir istasyonun hata sayaçlarından biri 127 değerine ulaşırsa, hata aktif durumundan hata pasif durumuna geçer. Bu durum istasyonun ağda sürekli olarak hata algıladığını gösterir, istasyon iletim ve alım işlemlerine olağan şekilde devam eder ancak bu durumda aktif hata bayrağı iletmesine izin verilemez, pasif hata bayrağı iletmesine izin verilir.

Pasif hata bayrağı, altı tane çekinik bitten oluşur ve Aktif hata bayrağı kadar etkili değildir. Bu tedbir pasif hata bayrağı ileten istasyonun kendisinde sorun olabileceği ihtimali dolayısıyla alınır. Problem çözüldüğünde hata pasif durumundaki istasyon sayaçlarını azaltır ve hata aktif durumuna geri döner.

2.2.6.3. Hat kapalı (Bus off):

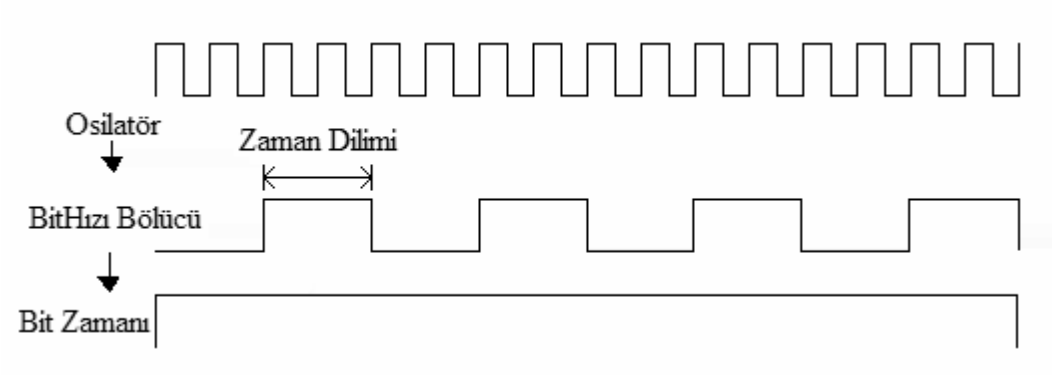
Bir istasyonun iletim sayacı değeri 255'i geçerse, istasyon hat kapalı durumuna geçer, iletimi durdurur ve hattı etkilemez. Hat kapalı durumundaki bir istasyonun düzgün çalışmadığı ve işlemlerine devam edemeyeceği varsayılır. İstasyon hat kapalı durumuna geçtiğinde, tam olarak fonksiyonlarını yerine getiremez, sadece ağda kalır. Böylece diğer istasyonların veri değişimini etkilemez. Hat kapalı durumundaki bir istasyon, 128*11 tane çekinik bit veya 128 doğru mesaj algılarsa hata aktif durumuna geçebilir (Tenruh, 2001).

2.2.7 CAN hat uzunluğundaki sınırlamalar ve zamanlama

CAN hattının maksimum uzunluğunu esas olarak aşağıdaki üç fiziksel etken belirler.
1. Hat üzerindeki CAN denetleyici, Alıcı-gönderici gibi istasyonların iletim hattı ve döngü gecikmeleri,
2. İstasyonlar arasında göreceli osilatör toleransından kaynaklanan, bit zaman dilimi *(bit time quantum)* farkları,
3- Sinyal genliğinin kablo direncine ve istasyonların giriş direncine bağlı olarak düşmesi.
İlk iki etken CAN protokol yapısı ve bit zamanlama özellikleri ile ilgilidir. Bu etkenlerle ilgi teknik bilgi aşağıda verilmiştir. Üçüncü etken ise, sistemin doğru akım karakteristikleriyle ilgilidir ve sinyalin köprü aracılığıyla hedef istasyonlarda tekrar edilmesi ile giderilebilir.

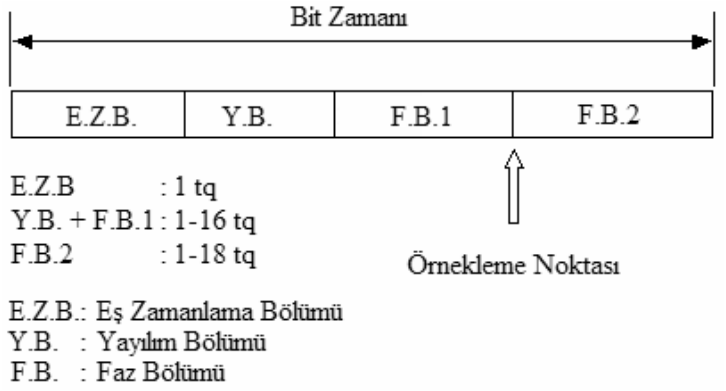
2.2.8. CAN bit zamanlaması

CAN protokolünün önemli bir özelliği, bit oranı (bit-rate) ve bit örnekleme noktasının kullanıcı tarafından tanımlanmasıdır. Bu durum, kullanıcıya farklı uygulamaların ihtiyaçlarına göre ağ performansını iyileştirme imkanı sağlar. İyileştirme işlemi sırasında, referans osilatör toleransı ve sistemdeki çeşitli sinyal yayılım gecikmeleri gibi zamanlama parametreleri arasındaki ilişkiyi hesaba katmak gerekir. Örneğin, bit sürelerinde örnekleme noktasının geç seçilmesi, daha toleranslı bir yayılım gecikmesi ve çok daha uzun bir hat sonucunu doğurur. Bununla birlikte, örnekleme noktasının bit süresinin ortalarına yakın seçilmesi, sistemdeki her birim için çok daha büyük osilatör toleransı sağlar. Bu örneklerden hareketle, daha geniş osilatör toleransı ile daha uzun hattın birbiriyle çelişen amaçlar olduğu söylenebilir (Tenruh, 2001).



Şekil 2.9. Zaman dilimi ve bit zamanı

Bit zamanlama, bir Denetleyici Alan Ağı sisteminde, hat uzunluğunu belirleyen temel konudur. Şekil 2.9'da bit zamanının elde edilişi gösterilmiştir. Bit zamanlamasında temel birim Zaman Dilimi *(time quantum-tq)* olarak adlandırılır ve her bölüm zaman diliminin tam sayı katıdır. Zaman dilimi bir CAN istasyonu tarafından kullanılan en küçük zamanlama çözünürlüğüdür ve uzunluğu CAN istasyonu osilatör frekansının tam olarak bölünmesi ile belirlenir. Bir bit en az 8 ve en çok 25 zaman diliminden oluşur. Bit zamanı, CAN denetleyicisinde zaman dilimi genişliğinin ve bir bitin değişik bölümlerdeki zaman dilimi sayısının hesaplanmasıyla seçilir. Şekil 2.10.'da zaman diliminin osilatör sinyalinden üretilmesi ile birlikte bit zamanlama işlemi görülmektedir.



Şekil 2.10. Bit zamanı bölümleri ve örnekleme noktası

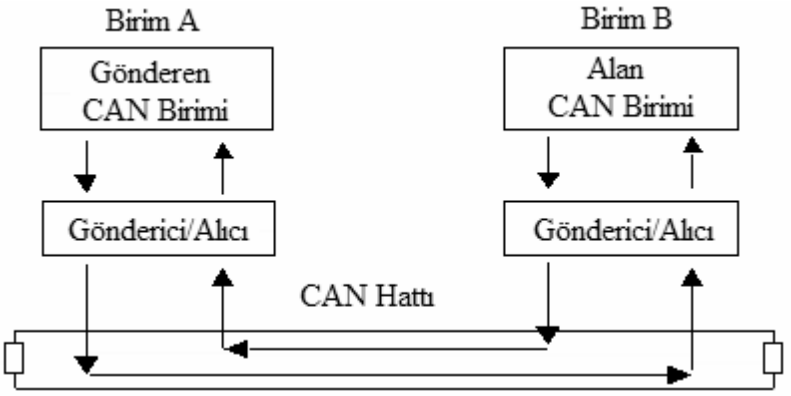
Şekil 2.10'da görüldüğü gibi CAN bit zamanı, birbirine eşit olmayan birkaç zaman bölümünden oluşur. E.Z.B. alanı 1 sabit zaman diliminden *(tq)* oluşur. Bu alan, ağdaki birimlerinin eş zamanlamasında kullanılır. Bu alanda bir düşen kenarın bulunması beklenir. Y.B. alanı 1 ile 8 zaman dilimi arasında programlanabilen bir alandır ve ağdaki sinyal yayılım gecikmelerini karşılamak amacıyla kullanılır. F.B.1 alanı 1 ile 8 zaman dilimi uzunluğunda olabilecek şekilde programlanabilir. Bu alan kenar faz hatalarını gidermekte kullanılır ve tekrar eş zamanlama *(resynchronisation)* sırasında uzatılmış olabilir. F.B.2 alanı 1 ile 18 zaman dilimi uzunluğunda olabilir ve bilgi işleme zamanını içerir. Bu alan aynı zamanda kenar faz hatalarını gidermede kullanılır ve tekrar eş zamanlama sırasında kısaltılmış olabilir. Bilgi işleme zamanı 2 zaman dilimi uzunluğundan küçük ya da eşittir. Örnekleme noktasıyla birlikte başlar ve bit seviyesinin hesaplanmasına ayrılmıştır. Örnekleme noktası, hat seviyesinin okunduğu ve bit değerinin algılandığı andır.

Örnekleme noktasının programlanması ile bit zamanlama iyi bir şekilde gerçekleştirilebilir. Ancak burada dikkat edilmesi gereken nokta, geç örnekleme noktası seçimi daha fazla hat uzunluğuna izin verirken, erken örnekleme noktası seçiminin ise daha fazla osilatör toleransı sağlamasıdır (Tenruh, 2001).

2.2.9. Yayılım gecikmesi

Denetleyici Alan Ağı hat erişiminde, yıkıcı olmayan *(non-destructive)* bit esaslı kararlaştırma mekanizması kullanır. Bu durumun sonucu olarak bir CAN sisteminde yayılım gecikmesi önemli bir konu haline gelir (Anonim1, 1999).

Kararlaştırma esnasında, birden çok istasyon aynı anda tanıtıcılarını hat üzerinden iletir. Bütün istasyonların bit kenarlarında eş zamanlanmış olmalarından dolayı aşırı yayılım gecikmesi meydana gelir ve bu durum kararlaştırmanın geçersiz olmasına neden olur. Denetleyici Alan Ağı sisteminde hat uzunluğunu sınırlandıran çeşitli gecikmeler vardır.

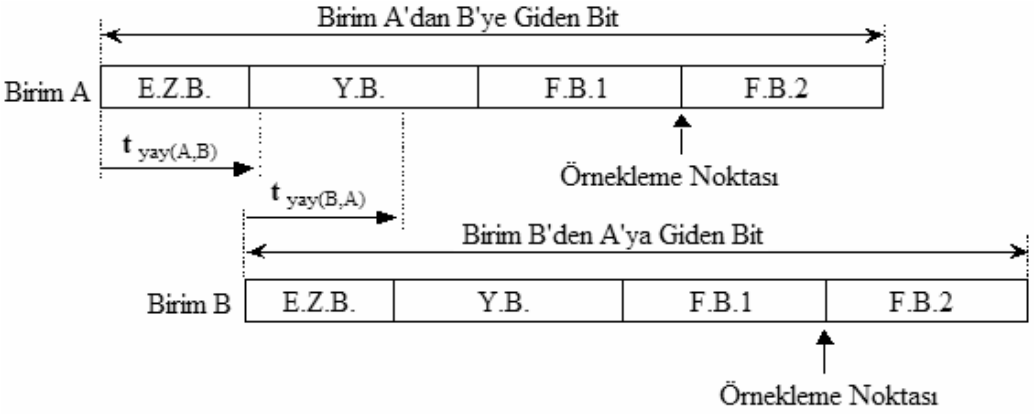


Şekil 2.11. CAN sisteminde sinyal gecikmesi

Şekil 2.11'de iki CAN istasyonu arasındaki sinyal yolu ve yayılım gecikmesi etkenleri görülmektedir. İstasyonlar arasındaki tek yönlü yayılım gecikmesi, örneğin A ve B arasındaki, $t_{yay}(A,B)$ şeklinde tanımlanabilir. Toplam gecikme, gönderici-alıcıda 120 ile 250 ns arası, CAN denetleyicisinde 50 ile 62 ns arası ve iletim kablosunda yaklaşık 5ns/m olan gecikmelerin toplamıdır. Eş zamanlamadan sonra en uzaktaki istasyon, yayılım gecikme zamanı ile birlikte anahtarlama kenarlarını bekler ve iletici istasyon, alıcı istasyonun onay alanının (ACK) veya tanıtıcı alanının geçerliliğini garanti altına almak için, dönüş yayılım zamanını beklemek zorundadır. Bu yüzden sistemdeki toplam yayılım gecikmesi iki istasyonun gecikmelerinin toplamıdır ve aşağıdaki gibi ifade edilebilir.

$$t_{(yayılım)} = 2\,(t_{kablo} + t_{denetleyici} + t_{gönderici}) \qquad (2.1)$$

Şekil 2.12.'de yayılım gecikmesinin iki istasyon arasındaki sinyal iletimine etkisi görülmektedir.



Şekil 2.12. İstasyonlar arası yayılım gecikmesi

Çizelge 2.3.'te bir CAN sistemindeki yayılım gecikmesine örnek değerler verilmiştir. Bu örnekte en çok gecikme 1630 ns'dir ve bu 250 Kbps hat hızındaki bir CAN bit zamanının % 41'ine denk gelmektedir. Bundan dolayı, işaret yayılması gecikmelerini karşılamak zorunludur ve toplam yayılım gecikmesi bir bit içindeki Yayılım Zamanı Bölümü uzunluğundan az olmalıdır. Bu durum CAN hat uzunluğunun, belli hızda belli bir sınırda olmasının nedenidir. Çizelge 2.4.'de belirli bit hızlarına göre kullanılabilecek en fazla kablo uzunlukları verilmiştir.

Çizelge 2.3. CAN sisteminde yayılım gecikmesi örneği

Sembol	Açıklama	En Az	En Çok
$t_{gönderici}$	Göndericinin Yayılım Gecikmesi	30 ns	157 ns
$t_{denetleyici}$	CAN Denetleyicisinin Yayılım Gecikmesi	15 ns	40 ns
Δ	Hat Gecikmesi	5 ns/m	6.5 ns/m
L	Birimler Arası Hat Uzunlugu	3 m	95 m
t_{kablo}	tkablo = L . Δ	15 ns	618 ns
$t_{yayılım}$	2 (t_{kablo}+ $t_{denetleyici}$ + $t_{gönderici}$)	120 ns	1630 ns

Çizelge 2.4. Bit hızlarına göre kablo uzunlukları

Bit Hızları	Hat Uzunluğu (m)	Bit Zamanı (µs)
1 Mbps	30	1
800 Kbps	50	1.25
500 Kbps	100	2
250 Kbps	250	4
125 Kbps	500	8
62.5 Kbps	1000	20
20 Kbps	2500	50
10 Kbps	5000	100

2.2.10. Eş zamanlama (Senkronizasyon)

Bit zamanlamada önemli olan bir diğer konu da eş zamanlamadır ve CAN hat uzunluğunun sınırlandırılması ile ilgilidir. Eş zamanlama, istasyonlar arasında oluşabilecek faz hatalarına karşın, mesajların doğru çözümlenmesini garantiler (Lawrenz, 1997). Faz hataları temel olarak, osilatör kaymaları, birbirine uzak istasyonlar arasında meydana gelen yayılım gecikmeleri ve gürültüden kaynaklanır.

Eş zamanlamanın sert *(hard)* ve yumuşak *(soft)* olmak üzere iki tipi vardır. Yumuşak eş zamanlama tekrar eş zamanlama *(resynchronisation)* olarak da bilinir.

Sert eş zamanlama mesaj paketinin başlangıcında gerçekleştirilir. Hat boştayken ağdaki her CAN denetleyicisinin, ilk çekinik (1) bitin, baskın bite (0) geçtiği eş zamanlama bölümünde (E.Z.B), kendi bit zamanını ayarlamasıyla gerçekleştirilir. Yumuşak eş zamanlama ise her çekinik bitten, baskın bite geçilen kenarda, mesajın geri kalanı boyunca devamlı olarak gerçekleştirilir.

Yumuşak eş zamanlama esnasında, eğer eş zamanlama bölümünden (E.Z.B) sonra, örnekleme noktasından önce bir düşen kenar (1'den 0'a geçiş) alınırsa, alıcı bu durumu sinyalin daha yavaş bir ileticiden geldiği şeklinde yorumlar. Bu durumda, alıcının faz bölümü (F.B.1) ileticinin zamanlamasıyla eşleştirmek için uzatılır. Eğer tersi olursa, yani alıcı, örnekleme noktasından sonra eş zamanlama bölümünden önce bir düşen kenar alırsa bu durum, sinyalin daha hızlı bir ileticiden geldiği şeklinde yorumlanır ve alıcı ikinci faz bölümünü (F.B.2) kısaltır.

2.2.11. CAN uygulamalarının sınıflandırılması

CAN uygulamaları ISO ve SAE gibi çeşitli organizasyonlar tarafından sınıflandırılmıştır.

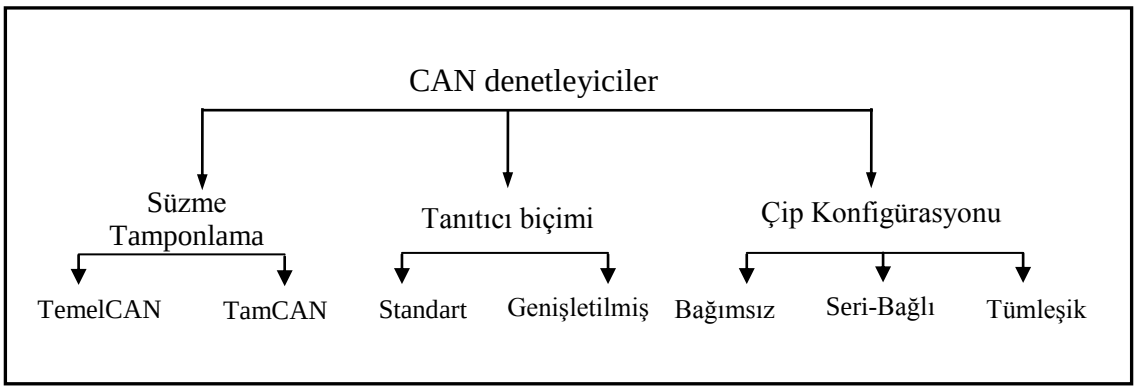
SAE tarafından yapılan sınıflandırmaya göre CAN uygulamaları, A sınıfı, B sınıfı, C sınıfı ve D sınıfı olmak üzere 4 kısımda incelenir.

ISO tarafından yapılan sınıflandırmaya göre CAN uygulamaları iki kısımda incelenir.

1. (ISO 11519-1) İletişim hızı 125 kbps altında olan düşük hızlı uygulamalar.
2. (ISO 11898) İletişim hızı 125 kbps üstünde olan yüksek hızlı uygulamalar.

2.2.12. CAN denetleyicilerinin sınıflandırılması

Denetleyicileri 1. Süzme-Tamponlama, 2. Tanıtıcı Biçimi, 3. Çip Konfigürasyon ölçütlerine göre olmak üzere üç grupta sınıflandırabiliriz. Gruplandırma Şekil 2.13.'te gösterilmiştir.



Şekil 2.13. CAN denetleyicilerin sınıflandırılması

2.2.12.1. Süzme ve tamponlamaya göre sınıflandırma

İlk sınıflandırma CAN nesnelerinin kabul süzmesine göre yapılabilir. Ara tamponu olan CAN denetleyicileri TemelCAN olarak bilinir. TemelCAN çipleri protokole göre bit dizileri oluşturmak ve doğrulamak için gerekli bir donanım olarak üretilmişlerdir. Gönderilen ve alınan verinin yönetimle ilgili sınırlı bir bölümü gönderilir ve alınır.

Onay süzmesi özellikle CAN denetleyici tarafından gerçekleştirilir. Bu denetleyicilerin tipik olarak iki kabul etme ve bir iletim tamponu vardır. Maske kaydedicisindeki 8 bit sınırlı bir onay süzmesine izin verir, dolayısıyla tanıtıcının sadece en önemli 8 biti kullanılabilir. Mesajları seçmek için 8 bitten fazlası gerekirse, CAN denetleyici ile aynı devrede bulunan mikrodenetleyici, onay süzmesini yazılımla tamamlamak zorundadır. Bu denetleyicinin avantajı az yer kaplaması ve düşük maliyetle üretilebilmesidir. Aynı zamanda mesaj işlemede esneklik sağlar.

Nesne belleği olan denetleyiciler TamCAN olarak bilinir. TemelCAN denetleyici gibi işlev görür ancak aynı zamanda belirli mesajları yönetebilir. Örneğin birden fazla istek olması durumunda hangi nesnenin önce iletileceğini belirleyebilir. Gelen mesajlar için onay süzmesi yapabilir. İletilen veri uygun RAM alanına yazılır, alınan veriler sırasıyla okunur. Mikrodenetleyici, iletim isteği gibi birkaç biti yönetmek durumundadır. TamCAN denetleyiciler mikrodenetleyici yükünü mümkün olduğunca hafifletmek için tasarlanmışlardır. Diğer taraftan bu teknik, donanım için çoklu tampon gerektirir. Esnekliği artırmak için bu tamponlar alıcı veya gönderici tamponlar olarak programlanabilir.

Günümüzde her iki çalışma türünü kapsayan denetleyiciler mevcuttur. Ara tamponu olan denetleyicilerin de nesne bellekleri vardır. Dolayısıyla, TemelCAN ve TamCAN arasında ayrım yapmaya gerek yoktur denilebilir.

2.2.12.2. Tanıtıcı biçimine göre sınıflandırma

Mesajda kullanılan tanıtıcı biçimine göre CAN2.0A ve CAN2.0B olarak sınıflandırılabilir. CAN 2.0A da tanıtıcı 11 bit uzunluğunda, genişletilmiş CAN paketi olarak da bilinen CAN2.0B de ise tanıtıcı 29 bit uzunluğunda olabilir. Bu tanımlamalarla ilgili olarak denetleyiciler arasında bazı farklar bulunur. Problemleri engellemek için CAN 2.0A ve CAN 2.0B arasında geçiş sağlayan anahtarlama seçeneği kullanılmaktadır.

2.2.12.3. Çip konfigürasyonuna göre sınıflandırma

Bu sınıflandırma CAN çipinin entegrasyon derecesine göre yapılır. Bağımsız denetleyicilerde, denetleyici mikrodenetleyiciden ve alıcı-ileticiden ayrıdır. Bu denetleyicilerin kullanılma sebeplerinden bazıları, herhangi bir mikrodenetleyici ile bağlantı kolaylığı, silikon boyutu, maliyeti, aynı çip içinde farklı gereksinimlerin ve çözüm güçlüğünün olması olarak sıralanabilir.

Mikrodenetleyici ile aynı çipte olan CAN denetleyiciler tümleşik olarak isimlendirilir ve denetleyici, mesaj tamponları arasında daha iyi erişim, düşük silikon boyutu, yüksek güvenlik gibi avantajlar sağlar. Tümleşik bir denetleyicide kaydediciler dahili adres/veri hattı kullanılarak adreslenir, dolayısıyla daha hızlı erişim sağlanır. Bağımsız denetleyicilerde harici adres/veri hattı kullanıldığı için erişim daha yavaştır.

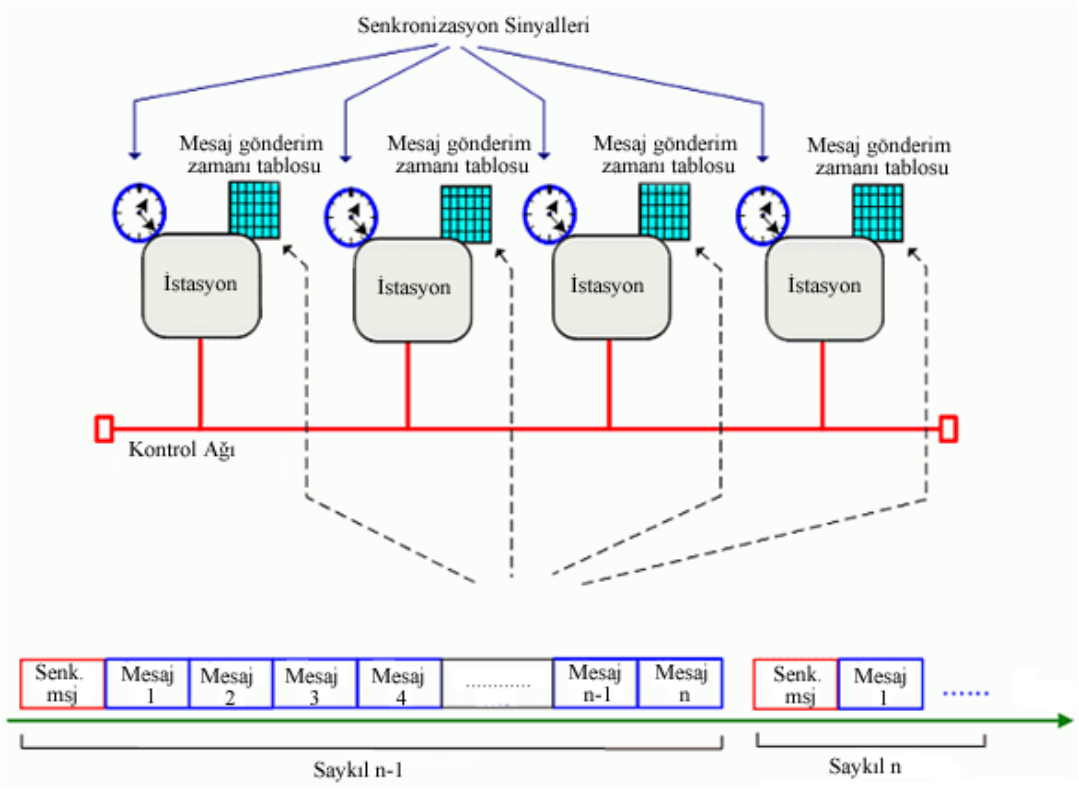
Diğer denetleyici tipi Seri Bağlantı G/Ç (Serial Link I/O-SLIO) olarak isimlendirilen ve programlama yeteneği olmayan CAN denetleyicilerdir. Bu denetleyiciler yönetim için programlanabilir CAN istasyonlarına ihtiyaç duyarlar. Daha çok araba gövdelerinde kullanılırlar, düşük silikon boyutu ve maliyet gibi avantajları vardır.

2.3. Zaman Tetiklemeli Denetleyici Alan Ağı (TTCAN)

TTCAN, CAN yapısının değişmeden üzerine eklenen bir katman olarak tanımlanabilir. Bu üst seviye protokol global bir sistem saati sağlayarak bütün istasyonların iletişim planlamasını senkronize eder. Bu protokol, motor yönetimi gibi yüksek güvenlikli uygulamalar için geliştirilmiştir. Fren, direksiyon gibi, otomobillerde güvenliğin önemli olduğu ve bu gibi mekanik, hidrolik sistemlerin yerini elektronik sistemlerin aldığı uygulamalarda kullanılmak üzere tasarlanmıştır. TTCAN mutlak gerçek zamanlı sistemlerde, yani mesajın doğru olarak iletilmesinin yanında tam zamanında iletilmesinin de önemli olduğu sistemlerde kullanılır. Klasik CAN ağlarında eğer mesaj iletilirken bir hata meydana gelmişse, mesaj tekrar iletilir. Bu durum TTCAN uygulamaları için geçerli değildir, böyle bir durumda kontrol uygulaması bilgilendirilir (Leen ve Haffernan, 2002).

Zaman tetiklemeli CAN sisteminde mesaj gönderme ve G/Ç servisleri gibi sistem olayları periyodik olarak gerçekleştirilir. Olay tetiklemeli bir sistemde ise sistem işlemlerinin önceden tanımlanmış bir zamanlaması yoktur. Olay tetiklemeli sistemden farklı olarak, zaman tetiklemeli sistemde mesaj iletimi bir zaman dizisiyle gerçekleştirilir ve bütün istasyonlar senkronize edilmiş, global saate ayarlanmıştır. Her mesajın kendine ait zamanı olduğu için herhangi bir çarpışma meydana gelmez. Bütün mesajların zamanlaması çalışma zamanından önce statik zaman planlaması olarak gerçekleştirilir. Bütün sistem mesajları zamanlarının planlanması sorunsuz bir sistem meydana getirir.

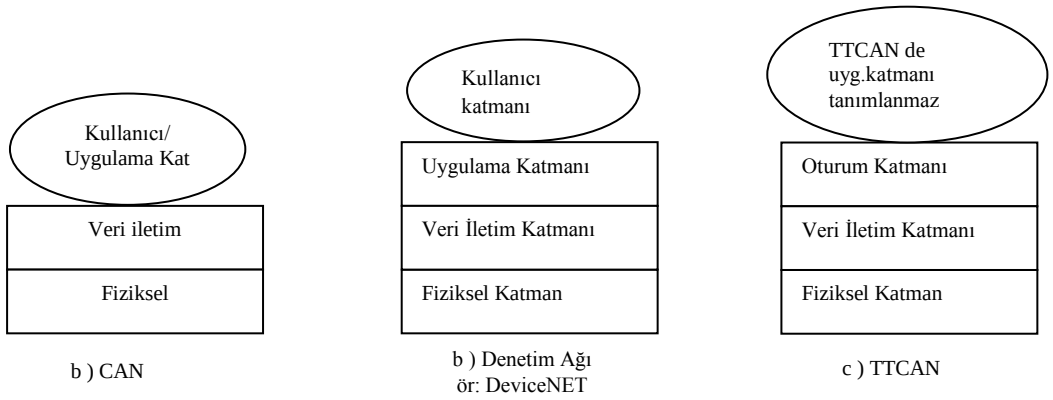
Ağdaki bütün istasyonların genel zamanı vardır ve her istasyon ağ zaman planlamasını bilir. Şekil 2.14.'te bu durum gösterilmektedir.



Şekil 2.14. Zaman tetiklemeli ağ yapısı

CAN ağ yapısı dünya çapında en popüler otomobil ağ yapılarından olmasına rağmen güvenliğin kritik olduğu uygulamalarda yani zaman tetikleme ihtiyacı olan durumlarda TTCAN tercih edilir.

CAN mimarisinde OSI katmanlarının ikisi bulunur, fiziksel katman ve veri iletim katmanı. DeviceNET gibi CAN tabanlı kontrol ağları, uygulama katmanını da eklemişlerdir. TTCAN klasik CAN katmanlarına zamanlamayı gerçekleştirmek için oturum (session OSI- katman 5) katmanı ekler fakat uygulama katmanı yoktur (Leen ve Heffernan, 2002). Şekil 2.15.'te bu durum karşılaştırmalı olarak görülmektedir.



Şekil 2.15. CAN-TTCAN ağ modelleri

Dağıtılmış kontrol uygulamalarında TTCAN, CAN ağına, oturum katmanının biçimsel zamanlama ortamını kullanarak erişir, böylece uygulama mesajlarının önceliğine göre yerleştirilmiş zaman pencereleri bulunur. Zaman tetiklemeli CAN sisteminde mesaj gönderme ve G/Ç servisleri gibi sistem olayları periyodik olarak gerçekleştirilir. Sistemin bütün durum geçişleri önceden tanımlı bir zamanla ilişkilendirilir. Olay tetiklemeli sistemde, sistem durum geçişlerinin önceden tanımlı zamanları yoktur, dolayısıyla bütün mesajlar en kötü ihtimalle yarışma durumunda kalabilir ve sonuç olarak uzun beklemeler olabilir.

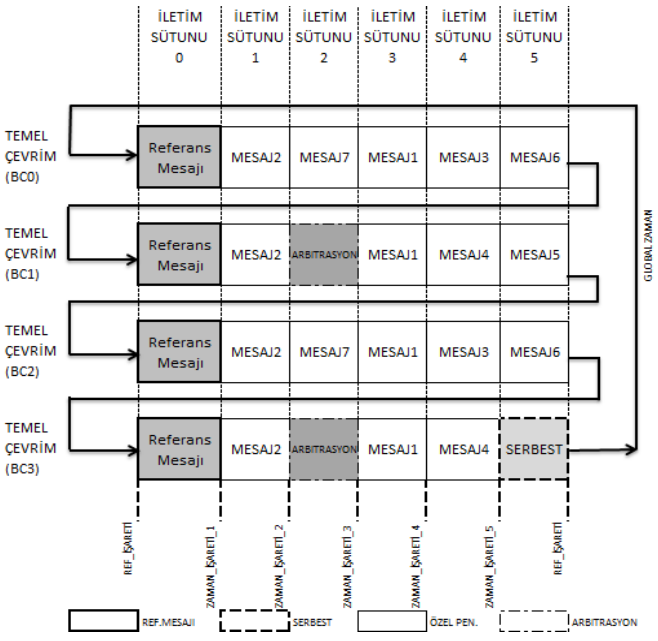
TTCAN sadece güvenilir mesaj zamanlaması sağlamaz, aynı zamanda mevcut bant genişliğinin etkili kullanımını da sağlar. Güncel olay tetiklemeli CAN uygulamalarında ağdan yararlanma göreceli olarak düşüktür. Kritik olmayan uygulamalar için ağdan yararlanma %50 seviyesindedir ve mutlak gerçek zamanlı sistemler için ağdan yararlanma daha düşük, %20- %30 civarındadır. Bu oran mutlak gerçek zamanlı sistemler için uygundur, çünkü hata olması durumunda mesajın

yeniden gönderilmesi gerekir. TTCAN ağ bant genişliğinden %90 seviyesinde yararlanabilir.

TTCAN hata kontrolünü iki ayrı basamakta gerçekleştirir. Birincisi CAN standardında olduğu gibi fiziksel ve veri iletim katmanında yapılır, ikincisi ise zaman planlama hatalarına odaklanan oturum katmanında yapılır.

2.3.1. TTCAN matris çevrimi

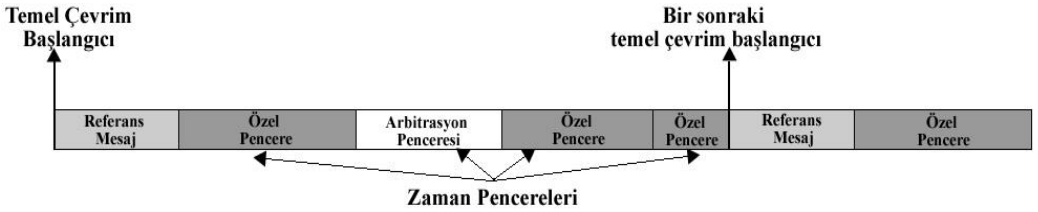
Zaman tetiklemeli çizelge belirli sırada zaman slotlarından veya pencerelerinden oluşur. Her mesaj penceresinde bir mesaj iletilebilir. Bütün olarak geçiş sırası matris çevrimi olarak bilinir ve mesaj iletimi periyodik olarak tekrar eder. Matris çevrimi, bir referans mesajıyla başlayan ve bir sonraki referans mesajıyla son bulan temel çevrimlerden meydana gelir (Leen ve Heffernan). Şekil 2.16'da 4 temel çevrimden oluşan örnek bir matris çevrimi görülmektedir. Aslında matris çevrimi, bütün ağ için geçerli olan bir zamanlama yapısını temsil eder. TTCAN protokol terminolojisinde zaman işareti (time-mark) zaman içerisinde belirli bir duruma işaret eder.



Şekil 2.16. TTCAN matris çevrimi

Veri hattında bulunan istasyonlar, bir pencere süresince tam iletim kapasitesine sahip olarak zaman pencerelerine atanırlar. Her istasyon mesaj paketini temel çevrim biçiminde gönderir. Bütün temel çevrimler zaman noktasında eşit büyüklüktedir, ancak değer noktasında gönderilen paketlerin içeriğine ve uzunluğuna bağlı olarak farklılık olabilir. Bir temel çevrim tamamlandığında, aynı erişim biçimine fakat farklı paket sırasına sahip bir sonraki temel çevrim başlar.

Temel çevrimlerin sayısı (0 ile 63 arasında) matris çevriminin uzunluğunu belirler. İki referans mesajı arasında kalan periyot temel çevrim (Basic Cycle – BC) olarak adlandırılır. Bir temel çevrim periyodik mesajların gönderimi için kullanılan özel mesaj pencerelerinden, sporadik mesajların gönderimi için kullanılan kararlaştırma (arbitrasyon) pencerelerinden ve sistemin geliştirilme ihtimaline karşı konulmuş olan boş zaman pencerelerinden oluşabilir. Bu pencereler farklı büyüklüklerde olabilir. Matris çevrimi tamamlandığında, iletim matris çevriminin tekrarı biçiminde devam eder. Şekil 2.17.'de örnek bir temel çevrim görülmektedir.



Şekil 2.17. Temel çevrim

Matris çevrimini oluşturan zaman pencereleri üç farklı tipte olabilir.

2.3.1.1 Özel zaman pencereleri (Exclusive time windows)

Temel çevrimde periyodik bir mesajın gönderilmesi için sadece o mesaja özel pencere aralığı tanımlanmıştır. Buna özel zaman penceresi *(Exclusive time window)* denir. Özel zaman penceresi boyunca sadece belirli bir istasyondan bir mesaj gönderilebilir. Ortama erişim için pencere süresi boyunca herhangi bir yarışma yoktur.

2.3.1.2 Kararlaştırma zaman pencereleri (Arbitration time windows):

TTCAN uygulamalarında temel çevrimlerde zaman tetiklemeli sinyallerin yanında olay tetiklemeli sinyaller için ayrılmış özel pencereler de bulunmaktadır. Bu pencerelere kararlaştırma penceresi *(arbitration window)* adı verilir. Bu zaman aralıklarında sistemde bulunan olay tetiklemeli sinyallerin gönderimi gerçekleştirilir. Bu tip pencereleri farklı zaman aralıklarında birden fazla istasyon kullanabilir. Bunun yanında aynı zaman aralığında iki veya daha fazla istasyon kararlaştırma penceresini kullanmak isteyebilir. Böyle bir durumda CAN modeline özel, yıkıcı olmayan bit öncelikli kararlaştırma özelliği *(non-destructive bitwise arbitration)* devreye girer ve ortaya çıkabilecek bir mesaj çarpışmasını engeller. İki ya da daha fazla kararlaştırma penceresi birleştirilebilir. Bunlara birleştirilmiş kararlaştırma penceresi *(merged arbitration window)* denir. Bu bize periyodik olmayan mesajların gönderilmesi için daha geniş bir zaman aralığı sağlar. Tüm kararlaştırma zaman pencerelerinde mesajlar iletilmeleri için yeterli zaman varsa kararlaştırmaya girerler.

2.3.1.3 Serbest zaman pencereleri (Free time windows):

Boş zaman pencereleri temel çevrimde istasyonlara ayrılan zaman dilimlerinden arta kalan zamanı belirtmektedir. Bu zaman boşluklarında veri hattında hiçbir mesaj hareketi gerçekleşmez. Bu zaman aralığının bırakılmasının nedeni ileride oluşabilecek bir sistem geliştirmesinin önünü açmaktır.

2.3.2. Global zaman

Dağıtılmış ağ uygulamalarında, senkronize planlama için ilk şart, bütün planlama işlemlerinin referansı olan global zamandır. TTCAN içinde, planlı olarak mesaj iletim işlemi gerçekleştiren her istasyon, global zamanın kopyasını bir sayaç içerisinde tutar ve her Ağ Zaman Biriminde *(Network Time Unit-NTU)* bu sayacı bir artırır. Mutlak referansı kurmak ve ağ içerisindeki zamanın hızını ayarlamak için yüksek frekanslı bir Referans mesajı yayınlanır. Bu mesaj, zamanın ağdaki görüntüsünü işaretlemek ve ağdaki senkronizasyon kalitesinde gerekli düzenlemeleri

yapmak için bir bekleme zamanı yaratır. TTCAN'de zaman senkronizasyonunun iki seviyesi vardır. Seviye1 ve Seviye2.

Seviye1, mesajın zaman tetiklemeli planlamasında gerekli olan özellikleri destekler. Seviye2, seviye1'in gelişmiş halidir. Seviye1'de, mesajların zaman tetikleme planı dahilinde gönderilebilmesi için gerekli olan özellikler desteklenir. Seviye2'de fiziksel saniye ile ilgili olan hassas global zaman sağlanır. Ağ zaman birimi (NTU) TTCAN uygulamasının seviyesine göre farklı olarak ölçülendirilebilir.
Seviye1'de ağ zaman birimi, ağın bit oranının nominal bit zamanı süresine eşittir. Seviye2'de ise ağ zaman birimi fiziksel saniyenin bir parçasıdır, çünkü bu durum farklı ağ protokolleri arasında, geniş, senkronize uygulamalar oluşturmayı kolaylaştırır. Böylece TTCAN Bluetooth, TTP/A, TTP/C, GPS gibi farklı ağlarla ara yüz oluşturabilir.

2.3.3. Referans mesajı

Veri hattında zaman yöneticisi *(Time Master)* olan istasyon tarafından üretilen referans mesajı, mesajlaşma işleminin önceden tanımlanmış olan zaman dizisini başlatır. O anda ağda zaman yöneticisi olan istasyon referans mesajını iletir. Bu istasyon potansiyel zaman yöneticilerinin üyesidir ve Temel Çevrim *(Basic Cycle-BC)* başladığında zamanı ayarlamak için yerel saatini kullanır. Zaman yöneticisinin zamanı ağın global zamanına yani ortak zamana eşittir. Diğer istasyonlar (zamanı alan) mesaj iletim işlerini referans mesajın oluşuna göre planlarlar. Her bir istasyon referans mesajıyla eş zamanlıdır. Referans mesajlar belirli bir bölgede düzenli aralıklarla gelen mesajlar için bir referans noktası oluşturur. Bu düzenli mesajlar zinciri zaman bölmeli erişim *(TDMA)* şemasına dayanır. Bu şema yalnızca düzenli zaman aralıklarından ve pencerelerinden oluşur. TTCAN ağında iletişim referans mesajlardan oluşan yapının üzerine inşa edilir. Bu durum referans mesajlarının sigorta görevini üstlendiği bir çeşit protokol olarak düşünülebilir. Referans mesajlarının, dolayısıyla iletişimin varlığını sağlamak amacıyla protokol, istasyon dizisi (en fazla 8) içindeki herhangi bir istasyonu zaman yöneticisi olarak yetkilendirir.

2.3.4. Çevrim süresi (Cycle time)

Referans mesaj herhangi bir istasyon tarafından başarılı bir şekilde alındığında (Zaman yöneticisi de dahil) istasyon bir zamanlayıcı başlatır. Bu zamanlayıcı referans mesajın ortaya çıkışını baz alarak temel çevrim içerisindeki zamanın ilerlemesini kaydeder. Bu zaman çevrim süresi olarak bilinir. Çevrim süresinin birimi bir ağ zaman birimi (NTU) kadardır. TTCAN uygulama seviyesine göre farklı şekillerde ölçeklendirilebilir. Seviye1'de ağ zaman birimi (NTU) süre olarak 1 bitlik zamana eşittir. Seviye2'de ağ zaman birimi saniyenin belirli bir kısmı kadardır. Burada ağ zaman birimi saniyenin belirli bir kısmı kadar olduğu için farklı ağ protokolleri arasında geniş senkronize dağıtık sistem uygulamalarının oluşturulması daha basit hale getirilmiştir.

2.3.5. Mesajların gönderilmesi ve alınması

Bir istasyon tarafından mesajların gönderilmesi Tx Trigger ile başlatılır. Tx_Trigger özel bir mesaj aktarımı olduğunda gerekli bilgileri tutan bir kaydedici setidir. Tx_Trigger'ın 4 temel bileşeni vardır. Bunlar ;

- Bir özel mesajı gösteren işaretçi
- Mesajın gönderildiği iletim sütunu
- Mesajın ilk gönderildiği temel çevrim (Çevrim numarası)
- Belirtilen iletim sütununda mesajın hangi konumda tekrarlanacağını gösteren bir tekrarlama faktörü. Mesaj tekrarlanmıyorsa bu faktör aktif değildir.

Mesajın alınması TTCAN'de Rx_Trigger'ın kullanımı yoluyla doğrulanır. Bu kaydedici setleri özel bir mesajın beklenilen varış zamanıyla alakalı bilgiyi depolayan Tx_Trigger'lara benzer.

2.3.6. Gönderilme penceresi (Tx Enable Window)

Özel bir pencerede bir mesajın gönderilmesi sadece gönderilme penceresi (Tx Enable Window) olarak bilinen bir zaman aralığında başlatılabilir. Bu süre 1 ile 16 ağ zaman birimi (NTU) uzunluğunda olabilir.

Eğer hat, zaman penceresinin bu ilk evresi boyunca boş değilse mesaj gönderimi başarısızlıkla sonuçlanacaktır. Belirlenen gönderilme zamanında istasyonun mesajını geç göndermesi durumunda bir sonraki mesaj için ayrılan zaman penceresinde taşma meydana gelebileceği için, gönderilecek mesajlara bir sınır getirilmesi zorunluluğu ortaya çıkmıştır. Bu durum zincirleme mesaj gecikmelerinin de önüne geçmiş olacaktır.

2.3.7. Zaman yöneticisinin (Time Master) hata toleransı

TTCAN ağındaki tüm zaman tetiklemeli haberleşme işlemleri geçerli zaman yöneticisinin oluşturduğu referans mesajın oluşumuyla gerçekleşmektedir. Böylece zaman tetiklemeli protokol, zaman yöneticisinin hata tolerans fonksiyonunu oluşturmuş olur. Zaman yöneticisinin başarısız olması durumunda potansiyel bir diğer zaman yöneticisi referans mesajı göndererek senkronizasyon işlemini sağlar. TTCAN ağındaki 8 istasyondan herhangi biri öncelik sırasına göre potansiyel bir zaman yöneticisi olarak çalışabilir.

2.3.8. CAN ve TTCAN karışık istasyonlar

TTCAN, zaman tetiklemeli istasyonlar ile geleneksel CAN istasyonlarının aynı ağ içerisinde kullanılmasına izin verir, ancak uygulama aşamasında dikkatli düşünmeyi gerektirir. CAN olan istasyonlar, TTCAN istasyonların ağ içerisinde uyduğu temel zaman tetikleme yapısını bozmamalıdır. Örnek olarak periyodik olmayan mesajlar, özel bir planlama mesajının sonrasında gönderilmek üzere tetiklenebilir şeklinde bir kurala bağlanabilir. Bu işlemler kararlaştırma *(arbitration)* veya serbest *(free)* zaman pencerelerinde yer alabilir. Planlanmamış *(non-scheduled)* mesaj aktarımları, planlanmış mesaj aktarımı başlamadan önce tamamlanmalıdır. Bazı uygulamalar bu tip bir tekniği, sistemi TTCAN tabanlı olarak tekrar tasarlamanın fazla pahalı olduğu durumlarda veya yazılım uygulamasının sistem kaynaklarının sınırlı olmasından dolayı mümkün olmadığı durumlarda kullanabilir.

2.3.9. TTCAN uygulamaları

2.3.9.1. Otomobillerde kullanımı

Güvenlik, rahatlık, yakıt tasarrufu gibi ihtiyaçları karşılamak için birçok elektronik sistem geliştirilmiştir. Otomotiv endüstrisinde fren sistemlerinde, motor sistemlerinde, çekiş gücü kontrolünde, iklimlendirmede, merkezi kilit sisteminde ve koltukların, aynaların otomatik kontrolünde kullanılmaktadır.

2.3.9.2. Endüstriyel Uygulamalar

CAN denetleyicilerin düşük maliyetli ve küçük olması, aynı zamanda gerçek zamanlı sistemlerde yüksek hızlarda çalışması, zor ortamlarda çalışması, otomotiv endüstrisinin dışında da kullanılmasını sağlamaktadır. Örneğin: Marina kontrol ve dolaşım sistemlerinde, kontrol sistemlerinde ve tekstil makinelerinde.

2.4. Esnek Veri Oranlı Denetleyici Alan Ağı (CAN-FD)

Denetleyici Alan Ağı çok yüksek güvenlik derecesine sahip, dağıtılmış gerçek zamanlı kontrolü, verimli bir şekilde destekleyen bir seri iletişim protokolüdür. Yüksek hızlı ağlardan, düşük maliyetli çoklu kablolamaya kadar uygulama alanına sahiptir. Otomotiv elektroniğinde, Motor Kontrol Birimleri, Algılayıcılar, Anti-Patinaj sistemleri gibi sistemler CAN protokolü ile 1Mbit/s bit oranına kadar iletişim kurabilirler. Aynı zamanda, kablolama olmadan, lambalar ve elektrikli camlar gibi gövde elektroniğini kurmak mümkün olabilir.

CAN ağlarında veri oranını sınırlayan iki etken vardır. Bunlardan birincisi, CAN kararlaştırma mekanizması için gerekli olan bit uzunluğu ve ikincisi de veri bitlerinin sayısıdır.

Esnek veri-oranlı denetleyici alan ağı *(CAN-FD: CAN with Flexible Data-Rate),* ISO 11898-1 standardında tanımlanan, CAN tabanlı yeni bir protokoldür. CAN-FD kararlaştırma mekanizmasını kullanır, bunun yanında kararlaştırma alanından sonra bit oranını artırır.

Etkili veri oranları, veri alanlarının artırılması ile sağlanır. Standart CAN protokolü 4 bit veri uzunluk kodu kullanır ve en fazla 16 farklı kod tanımlanabilir. Ancak bu kodların ilk dokuzu (0 ile 8 arası) kullanılır. 9 ile 15 arası kodlar 8 bayt veriyi göstermek için kullanılır. CAN-FD protokolünde bu kodlar daha uzun veri alanlarını göstermek için kullanılır (Anonim, 2011).

Esnek veri-oranlı denetleyici alan ağı, CAN protokolünün yüksek hızlı veri oranı gerektiren uygulamalarda kullanılmasını sağlar. CAN FD denetleyicileri aynı zamanda, standart CAN iletişimi içinde, CAN FD protokolünü desteklemeyen diğer denetleyicileri yedek konumuna alarak, bir bölümde, yazılım güncelleme gibi bazı özel uygulamaları gerçekleştirmeye de uygundur.

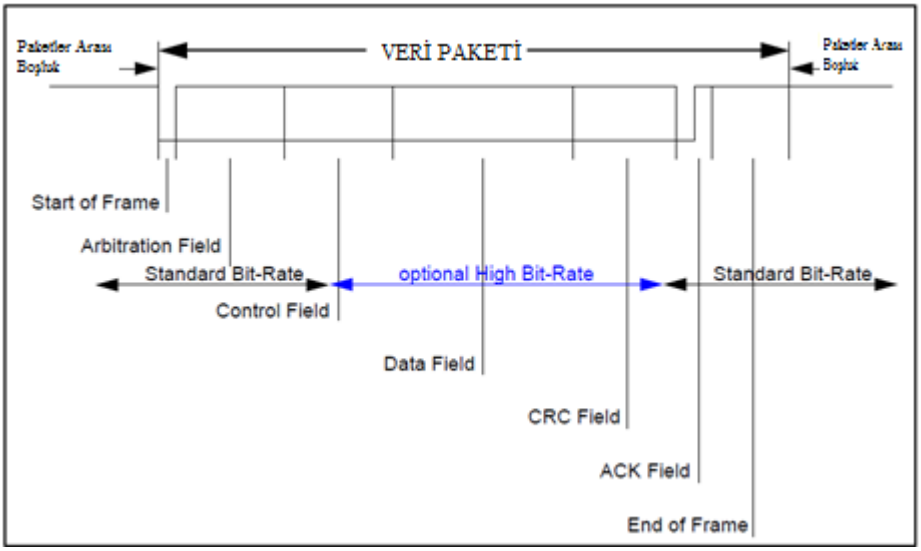
CAN FD, bir CAN paketinin ilk ayrılmış *(reserve)* bitini kullanır. CAN paketinde bu bit baskın (0) olarak iletilir. CAN FD paketinde bu bit, çekinik (1) olarak iletilir ve paketin CAN FD biçiminde iletildiğini gösterir. CAN FD denetleyicisi iki biçimdeki paketi de çözmeye uygundur. CAN FD ve CAN paketleri arasında temelde iki fark bulunur. Bunların ilki paketlerin 8 baytın üzerinde veri ile kullanılabilme seçeneği, ikincisi ise kararlaştırma (arbitration) bittikten sonra farklı bit oranına geçilebilmesidir. Kararlaştırma (Arbitration) evresindeki bit oranı standart CAN ağında olduğu gibi aynı sınırlamalara sahiptir, veri iletim evresindeki bit oranı ise seçilen alıcının performansı ile ve CAN FD ağının karakteristiği ile ilgilidir. İki bit oranı aynı da olabilir. Standart CAN alıcıları, CAN FD için kullanılabilir. CAN FD protokolü denetleyicileri, farklı bir işlem modunda, yüksek bit oranında, adanmış CAN FD alıcısına geçebilir. Bu alıcılar CAN protokolünün NRZ *(Non-return to zer*o) kodlama sistemi ile sınırlı değildir. Yüksek bit oranında, alternatif kodlama sistemi kullanabilir (Anonim, 2011).

2.4.1. Temel kavramlar

CAN FD paketi, standart CAN paketi ile benzerdir, aradaki fark, CAN FD paketinde Veri Alanı (Data Field) ve CRC (Cylic Redundancy Code) alanı daha uzun olabilir. Şekil 2.18.'de CAN-FD paket biçimi görülmektedir (Anonim, 2011). Mesajın geçerli olabilmesi için, CAN protokolünde olduğu gibi, baskın onay bitinin (ACK) en az bir alıcı tarafından gönderilmesi gerekir.

Hata kontrolü standart CAN protokolünde olduğu gibi gerçekleştirilir ve 5 tip hata bulunur. Bunlar:

Bit Hatası (Bit Error), Doldurma Hatası (Stuff Error), CRC Hatası, Biçim Hatası (Format Error) ve Onay Hatası (ACK Error) dır.



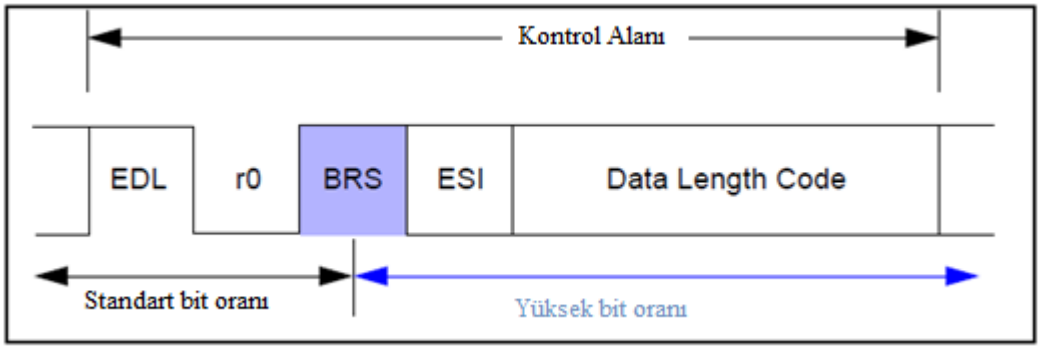
Şekil 2.18. CAN-FD paket biçimi

CAN FD paketinde, kararlaştırma alanının hemen ardından gelen ve her zaman baskın olan, ayrılmış bit, (r0) biti, çekinik yapılmış ve EDL (Extended Data Length- Genişletilmiş Veri Uzunluğu) olarak isimlendirilmiştir. EDL biti, 11 bit tanımlayıcılı paketlerde, IDE bitinden, 29 bit tanımlayıcılı paketlerde, RTR bitinden sonra gelir. EDL bitinin ardından, gelecekteki genişlemeler için ayrılan r0 biti gelir.

CAN FD paketinde ek olarak iki bit bulunur. Bunlar: BRS (Bit Rate Switch- Bit Oranı Anahtarı) ve ESI (Error State Indicator- Hata Durum İşaretçisi) bitleridir. Şekil 2.19.'da CAN-FD paketinde yeni eklenen alanlar gösterilmektedir (Anonim, 2011).

Eğer BRS biti çekinik olarak iletilir ise, bit oranı önceden belirlenen alternatif bit oranına dönüştürülür, baskın olarak iletilir ise bit oranı değişmez. CAN FD paketi BRS bitine kadar standart CAN paketi ile aynı bit oranında iletilir. Bit oranı değiştirildiğinde CRC sınırlayıcıya kadar, hata meydana gelmemesi halinde alternatif bit oranında iletim sağlanır. Bu aşamada bir hata meydana gelmesi durumunda CAN FD denetleyicisi hata paketi gönderir. Hata paketleri standart CAN hata paketleri ile aynı bit oranında iletilir.

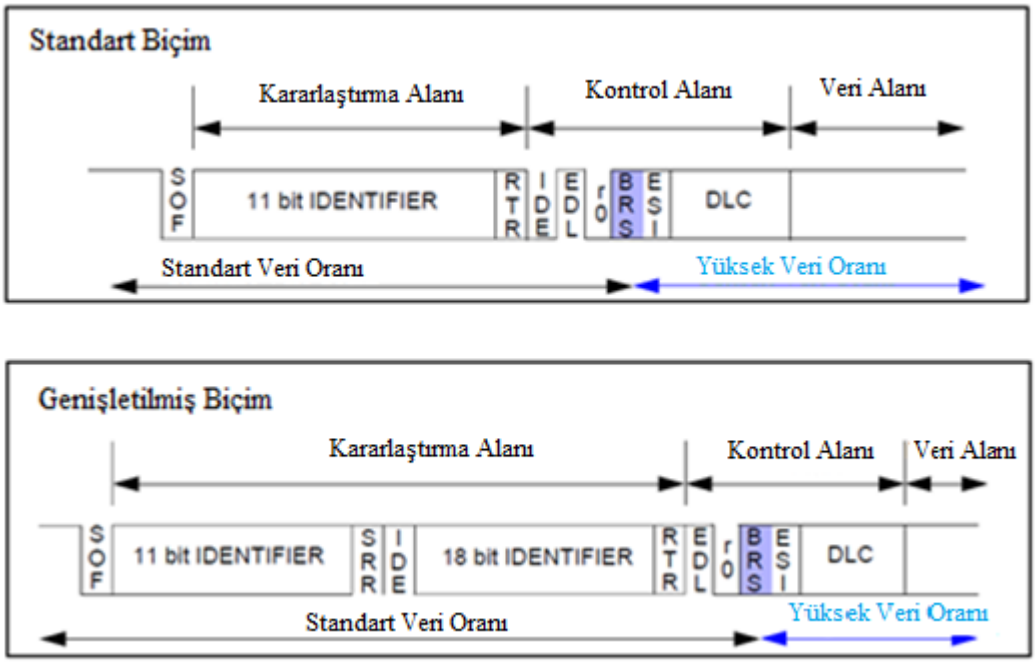
Eğer ESI biti çekinik olarak iletilir ise, hata pasif, baskın olarak iletilir ise, hata aktif durumu gerçekleşir.



Şekil 2.19. CAN-FD paketi kontrol alanı

2.4.2. Paket biçimi

CAN FD protokolü, standart CAN protokolündeki paketlerin ikisini de, yani hem 11 bit tanıtıcıya hem de 29 bit tanıtıcıya sahip paketleri destekler. İki paket biçiminde de bit oranı BRS biti ile değiştirilebilir. Şekil 2.20.'de CAN-FD protokolünün standart ve genişletilmiş paketleri görülmektedir (Anonim, 2011).



Şekil 2.20. CAN-FD standart ve genişletilmiş veri paketleri

2.4.2.1. Veri uzunluğu kodu (Data length code)

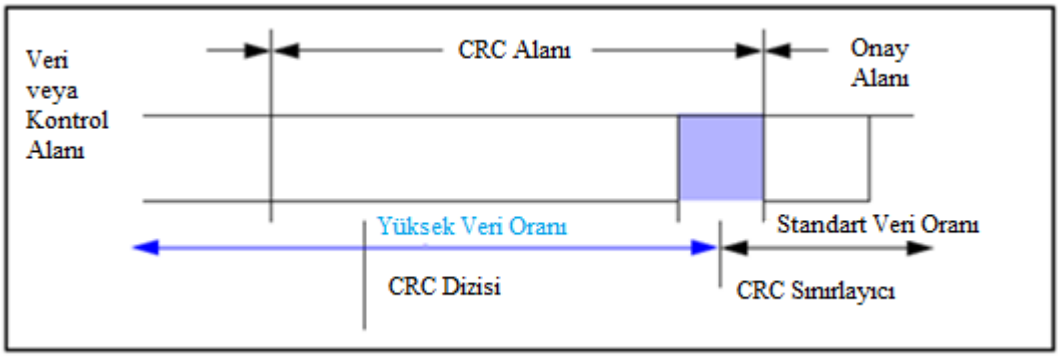
Veri alanının bayt olarak uzunluğu DLC içinde tutulur. DLC 4 bit kapasiteye sahiptir ve Kontrol Alanında (Control Field) iletilir. DLC bitlerinin kodlaması CAN FD protokolünde, standart CAN protokolündekine göre farklılık gösterir. İlk 9 kod aynı olmakla birlikte 8 bayta kadar olan veri alanını tanımlar. Ancak CAN FD protokolünün 8 baytın üzerinde veri iletimine imkan vermesi dolayısıyla, 8 baytın üzerindeki veri alanları da kodlanmıştır. Bu durum Çizelge 2.5.'te gösterilmiştir.

Çizelge 2.5. Veri kodu uzunluğu ile tanımlanan bayt sayıları

	Veri Bayt Sayıları	Veri Kodu Uzunluğu (DLC)			
		DLC3	DLC2	DLC1	DLC0
ISO 11898-1	0	0	0	0	0
	1	0	0	0	1
	2	0	0	1	0
	3	0	0	1	1
	4	0	1	0	0
	5	0	1	0	1
	6	0	1	1	0
	7	0	1	1	1
	8	1	0	0	0
CAN FD için önerilen ek kodlar.	12	1	0	0	1
	16	1	0	1	0
	20	1	0	1	1
	24	1	1	0	0
	32	1	1	0	1
	48	1	1	1	0
	64	1	1	1	1

2.4.2.2. CRC alanı

CRC alanı CRC Dizisi (CRC SEQUENCE) ve ardından gelen CRC sınırlayıcısından (CRC DELIMITER) oluşur. Şekil 2.21.'de (Anonim, 2011) CRC alanının paket içindeki yeri görülmektedir.



Şekil 2.21. CRC alanı

2.4.2.3. CRC dizisi (CRC sequence)

Paket denetim dizisi, BCH kodundan elde edilir. CRC hesaplamasını gerçekleştirmek için, bölünen polinom, ilgili bit akışının katsayıları ile tanımlanır. Polinom, üretici polinom *(Generator Polynomial)* tarafından bölünür.

CRC hesaplaması için ilgili bit akışı; doldurma bitlerini, kararlaştırma alanını, kontrol alanını, eğer varsa veri alanını ve n (üretici polinom tarafından oluşturulan) için en düşük katsayıları içerir.

CAN-FD daha uzun veri alanına izin verdiğinden dolayı, paketlerin HD (Hamming Uzaklığı, Hamming Distance) değerini 6 değerinde tutmak gerekir. Bunun için iki yeni polinom tanımlanır. İlk polinom g17, 16 bayta kadar veriye sahip paketler için kullanılır, ikinci polinom g21 ise 16 bayttan fazla veriye sahip paketler için kullanılır.

$g17 = X^{17} + X^{16} + X^{14} + X^{13} + X^{11} + X^{6} + X^{4} + X^{3} + X^{1} + 1$

$g21 = X^{21} + X^{20} + X^{13} + X^{11} + X^{7} + X^{4} + X^{3} + 1$

16 bayt üzerindeki veriye sahip paketler için farklı polinom kullanılmasından dolayı, CRC alanı uzunluğu artar. CAN-FD veri paketlerindeki CRC dizisi alanının uzunluğu veri uzunluk koduna (DLC) bağlıdır (Hartwich, 2012).

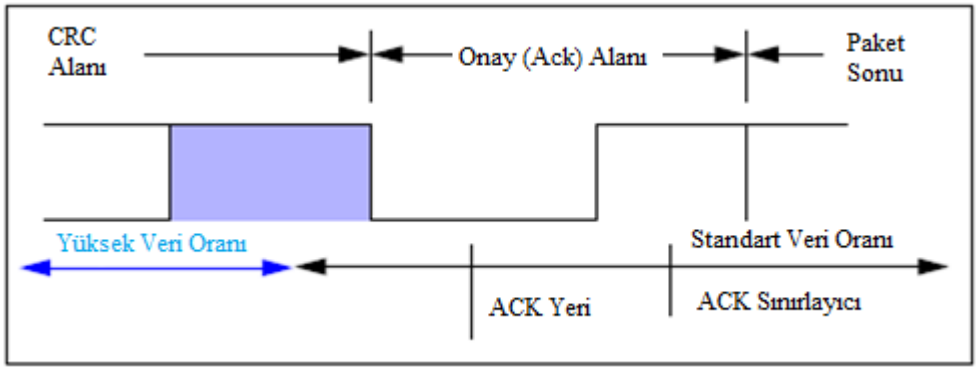
Her CRC Dizisi ayrı bir kaydırma kayıt bloğunda hesaplanır. Paketin başlangıcında, bütün düğümlerde, EDL biti ve DLC tarafından seçilen CRC dizilerinden birisi kararlaştırmaya girene kadar bütün CRC dizileri aynı zamanda hesaplanır. Sadece seçilen CRC dizisi, bir CRC hatasına neden olabilir. CAN FD paketlerinde, CRC dizisinden önce oluşan doldurma (stuff) bitleri, CRC tarafından korunabilir. CRC dizisi içinde oluşabilecek doldurma bitleri, CRC hesaplamasına veya CRC denetimine dahil edilmemelidir. CAN bit doldurma yöntemi CRC dizisi için değişir, doldurma bitleri sabit bir pozisyona eklenir. CRC dizisinin ilk bitinden önce sabit bir doldurma biti olmalıdır, bundan önceki alanın son bitleri bile olsa, CAN'in doldurma şartını yerine getirmez. Eklenen doldurma biti, CRC dizisinin her dört bitinden sonra eklenmelidir. Sabit doldurma bitinin değeri, bir önceki sabit doldurma bitinin tersi olmalıdır. Alıcı, sabit doldurma bitlerini, CRC denetimi için bit akışından çıkarabilir. Gerçek CRC polinomları, Veri Uzunluk Kodu alanının kodlamasının sonlanmasından sonra tanımlanması planlanmaktadır (Anonim, 2011).

2.4.2.4. CRC sınırlayıcı

CRC dizisinin ardından, bir ve ya iki çekinik bitten oluşan CRC sınırlayıcı gelir. Bir iletici, CRC sınırlayıcı olarak sadece bir çekinik bit gönderebilir, ancak bütün iletim istasyonları iki çekinik biti kabul edebilir. Bir alıcı Onay bitini ilk CRC Sınırlayıcı bitinden sonra gönderir. CAN FD denetleyicileri, CRC sınırlayıcı bitinden sonra düşük bit oranına geri döner. CAN ağındaki istasyonlar arasındaki faz geçişleri alıcılardaki bekleme zamanları ve CAN hattındaki yayılma zamanları aracılığıyla tanımlanır. CAN ve CAN-FD protokollerinde faz geçişleri aynıdır fakat düşük bit oranı zamanında orantılı olarak da uzundur. Ağdaki bütün alıcıların, ileticiye olan uzaklığına bağlı olarak farklı faz geçişleri olabilir. Çünkü iletilen sinyali alma zamanları değişebilir. Düşük bit oranına geri dönüldüğünde, faz geçişlerini karşılamak üzere, Onay (ACK) alanından önce ve sonra ek bir bit zamanına izin verilmiştir.

2.4.2.5.Onay (ACK) alanı

Onay alanı, Onay yeri (ACK Slot) ve Onay Sınırlayıcı (ACK Delimiter) bölümlerinden oluşan iki veya üç bitlik alandır. Şekil 2.22.'de onay alanının paket içindeki yeri gösterilmektedir (Anonim, 2011). İleten istasyon Onay alanında iki çekinik bit gönderir. Alıcı istasyon ise geçerli bir mesaj aldığını, onay yerinin başında bir adet baskın bit ile rapor eder.



Şekil 2.22. CAN-FD paketi onay alanı

2.4.2.6. Onay yeri (ACK slot)

CRC dizinini alan bütün istasyonlar, Onay yeri içinde yer alan ileticinin çekinik bitini baskın bir bit ile değiştirirler. Alıcılar arasındaki faz geçişlerini düzenlemek üzere, bütün istasyonlar onay alanında ardarda gelen iki baskın biti geçerli onay için kabul edebilirler.

2.4.2.7. Onay sınırlayıcı (Ack delimiter)

Onay sınırlayıcı biti, Onay yerinden sonra gelen ve her zaman çekinik olan bittir.

2.4.2.8. Paket sonu (End of frame)

Her veri paketi ve uzak paket yedi adet çekinik bit ile sonlandırılır.

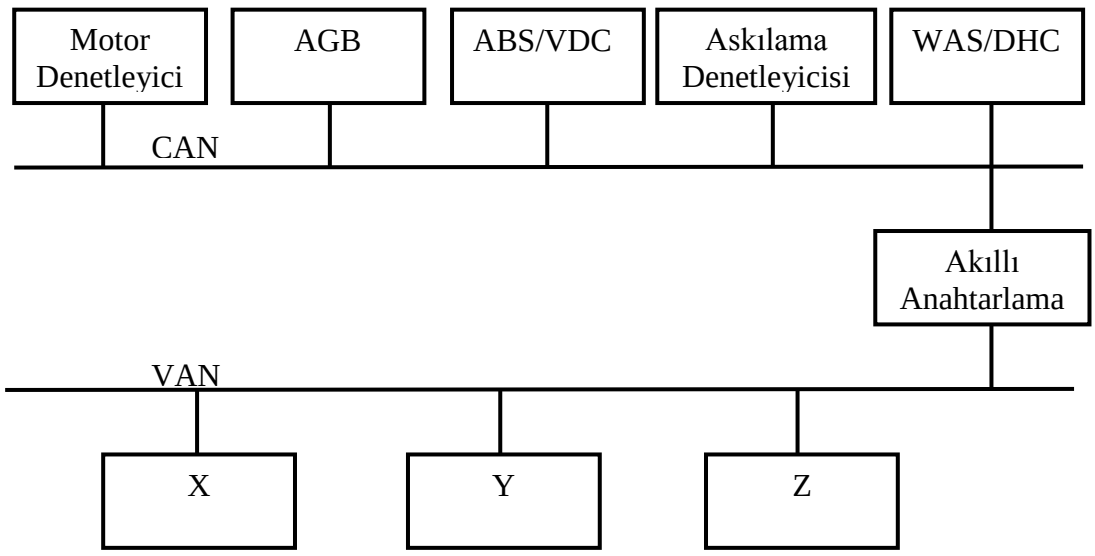
3. TASARIM VE UYGULAMA

3.1. PSA Benchmark

PSA benchmark 1997 yılında Peugeot-Citroen tarafından araç içi ağlar düşünülerek geliştirilmiştir. Bu benchmark, yüksek hızlı denetim iletişimi için kullanılan CAN ve düşük hızlı gövde iletişimi için kullanılan VAN protokollerinden oluşur. Şekil 3.1.'de PSA ağı yerleşimi görülmektedir. Akıllı Anahtarlama Birimi (ISU – Intelligent Swicthing Unit), protokoller arası geçişi sağlar ve akıllı anahtar olarak görev yapar (Mohammad ve Holou, 2010).

CAN ağının 5 birimi bulunur. Bunlar: Motor Denetleyici, Otomatik Vites Kutusu Denetleyici (AGB), Kilitlemesiz Fren Sistemi ve Araç Hareket Denetleyici (ABS/VDC), Askılama Denetleyici, Teker Açı Algılayıcı ve Hareketli Far Doğrulayıcı (WAS / DHC)

VAN ağının ise, güvenlik nedeniyle X,Y,Z olarak isimlendirilen, 3 birimi bulunur.



Şekil 3.1. PSA ağı

CAN hattında 12 adet mesaj, VAN hattında ise 7 adet mesaj bulunur. Bu mesajlar Çizelge 3.1.'de gösterilmiştir. Bu çalışmada CAN hattında bulunan 12 adet mesaj ile çalışılmıştır.

Çizelge 3.1. PSA mesaj seti ve gönderen birimler

Kaynak Birim	Mesaj Periyodu (ms)	Mesaj No (öncelik)	Veri Boyutu (Bit)	Ağ
Motor Denetleyici	10	M1	64	CAN
	20	M3	24	CAN
	100	M10	56	CAN
Otomatik Vites Kutusu (AGB)	15	M4	16	CAN
	50	M11	20	CAN
Kilitlemesiz Fren Sistemi (ABS)	15	M7	32	CAN
	20	M5	40	CAN
	40	M6	40	CAN
	100	M12	8	CAN
Teker Açı Algılayıcı (WAS)	14	M2	24	CAN
Askılama Denetleyicisi (SC)	20	M9	32	CAN
Gövde Uygulama Denetleyicisi (ISU)	50	M8	40	CAN
	50		64	VAN
	10		80	VAN
X	150		32	VAN
	200		32	VAN
Y	50		128	VAN
Z	100		160	VAN
	150		16	VAN

3.2. PSA Benchmark Mesaj Seti Kullanılarak Simülasyon Modellerinin Tasarlanması

Bu çalışmada PSA Benchmark mesaj seti kullanılarak Denetleyici Alan Ağları (CAN) ve Zaman Tetiklemeli Denetleyici alan Ağları (TTCAN) için simülasyon modelleri tasarlanmış ve geliştirilmiştir. Bu çalışmanın esas amacı PSA mesaj seti kullanarak TTCAN modelinin geliştirilmesi ve performans analizinin

gerçekleştirilmesidir. Ancak performans gelişiminin, sonuçların karşılaştırılarak analizinin yapılması için aynı mesaj seti kullanılarak, standart CAN erişim yönteminin kullanıldığı simülasyon modeli de geliştirilmiştir.

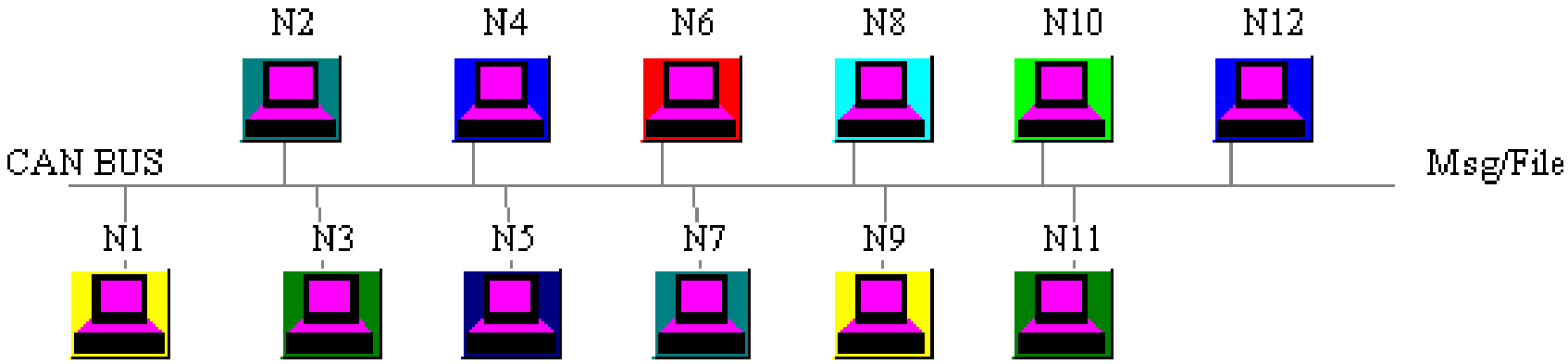
Simülasyon modellerinin gerçekleştirilmesinde ticari bir yazılım paketi olan Network II.5 simülasyon programı kullanılmıştır (Caci, 1997). Kullanılan simülasyon programına ait ara yüz görüntüsü EK A'da verilmiştir

3.2.1. Denetleyici alan ağı için model tasarlanması

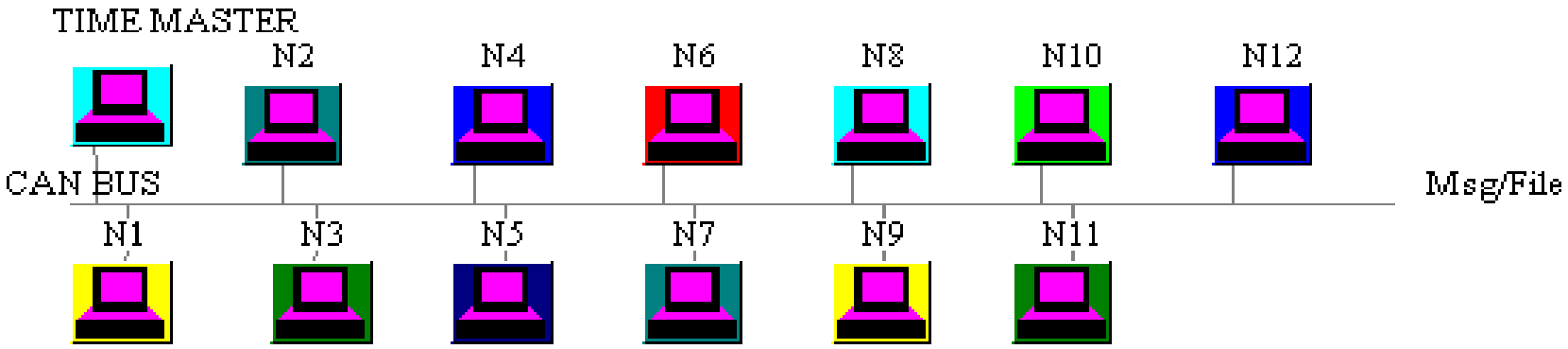
CAN modelinin geliştirilmesinde PSA benchmark mesaj seti tanımlamaları esas alınmıştır. Mesaj setinde 12 adet periyodik mesaj bulunmaktadır ve mesajlar önceliklerine göre sıralanmışlardır. CAN modelinde kullanılan ortam erişim yöntemi öncelik esaslı kararlaştırma yöntemi (CSMA / CD-CR) olduğu için mesajlar önceliklerine göre yerleştirilmiştir. Simülasyon programında her mesaj için bir ünite (node) olmak üzere toplam 12 adet ünite oluşturulmuştur. Mesajların periyot, mesaj boyutu gibi özellikleri, program aracılığıyla mesajları üreten ünitelerde tanımlanmıştır. Tasarlanan model 1 Mbps, 500 Kbps ve 250 Kbps hızlarında çalıştırılmış ve sonuçlar gözlemlenmiştir. Tasarlanan model Şekil 3.2.'de gösterilmiştir.

3.2.2. Zaman tetiklemeli denetleyici alan ağı için model tasarlanması

Zaman tetiklemeli sistemde kullanılan matris çevrimi ile tüm periyodik mesajlar kendilerine ayrılan özel mesaj pencereleri sayesinde eş zamanlı olarak gecikmeye uğramadan iletilirler. Bu sayede standart CAN ortam erişim yönteminde karşılaşılan ve özellikle düşük öncelikli mesajlarda görülen gecikmeler ortadan kaldırılmış olur. Simülasyon modelinde bir zaman yöneticisi *(time master)* ve her bir mesajın oluşturulması için tanımlanmış 12 adet ünite bulunmaktadır. Simülasyon programında oluşturulan tasarım Şekil 3.3.'te görülmektedir. Modele ait örnek kodlar EK B.'de verilmiştir.



Şekil 3.2. Simülasyon programında tasarlanan CAN modeli



Şekil 3.3. Simülasyon programında tasarlanan TTCAN modeli

Zaman yöneticisi her temel çevrimin (basic cycle-BC) başlangıcında, bir referans mesajı göndererek, diğer ünitelerin eş zamanlı bir şekilde kendi zaman dilimleri (Mesaj Penceresi) içerisinde mesajlarını iletmelerini sağlar. Periyodik mesajların gecikmesiz bir şekilde gönderilebilmesi için, farklı periyotlarda üretilen mesajlar için ayrılan zaman dilimlerinin matris çevrimi içerisinde çakışmadan yerleştirilebilmesi önemli bir tasarım konusudur. Tüm mesajların tekrarlama periyotlarının, en küçük mesaj periyodunun 2^j katı olması durumunda, mesaj pencerelerinin hiç çakışma olmadan yerleştirilmesi oldukça kolay bir işlemdir (Schmidth ve Schmidth, 2007). Ancak bu şart gerçekte karşılaşılan mesaj setleri için her zaman sağlanamayabilir. Genellikle karşılaşılan mesaj periyotları en küçük tekrarlama periyodunun 2^j katı değildir. J sabitinin değeri 0 ile 6 arasında olabilir.(Leen ve Heffernan, 2002)

Bu tez çalışmasında kullanılan yöntemde periyotların sadece 2^j katı değil, herhangi bir katı olması mesajların matris çevrimine yerleştirilebilmesi için oldukça büyük kolaylık sağlar. Hatta bunun ötesinde en küçük tekrarlama periyodunun katı dahi olmayan mesajların da, bu çalışmada kullanılan yöntemle matris çevrimine yerleştirilmesi mümkün olabilmektedir. Bu özelliği ile kullanılan yöntem 2^j katı olma kuralı sınırının ötesinde bir matris çevrimi tasarımı olanağı sunmaktadır. Bu yöntemin uygulanabilmesinde azaltılmış matris esasının kullanılmasının önemli bir yeri vardır. Normalde matris çevrimi tasarımında matris boyutunun mesaj periyotlarının ortak katlarının en küçüğü (OKEK) olması esas alınır. Bu durum özellikle en küçük mesaj periyodunun katı olmayan mesaj periyotlarının bulunduğu mesaj setleri için çok büyük boyutlarda matris çevrimlerinin ortaya çıkmasına sebep olur. Bunun sonucunda kullanılan denetleyicinin ve TTCAN standardının getirdiği sınırlamaların ötesine geçilmiş olur. Bu çalışmada kullanılan azaltılmış matris yöntemine dayalı çözüm bu sorunu ortadan kaldırmaktadır.

3.2.3. Matris çevrimi tasarımı

Matris Çevrimi (MC) tasarımı yapılırken, 4 adet Temel Çevrimden (BC) oluşan Matris Çevriminin toplam süresi 40 ms olarak ve her Temel Çevrim süresi 10 ms olarak planlanmıştır.

Azaltılmış Matris Çevrimi yöntemi uygulanırken, Matris Çevriminin (MC) toplam süresi ile her mesajın periyodunun ayrı ayrı Ortak Katlarının En Küçüğü (OKEK) hesaplanır. Daha sonra her mesaj için tekrar sayısı (Ts), her mesaj için bulunan OKEK'in, mesaj periyoduna (Pm) bölünmesiyle bulunur (Tenruh,2011). Bu konuda farklı yöntemler kullanılarak tasarlanan matris çevrimleri de bulunmaktadır (Ding, 2008; Xiao, 2010; Zhu, 2010).

Mesajların matris çevrimindeki tekrar sayıları hesaplanırken, Matris çevriminin toplam süresi olan 40 ms ile mesajın periyodunun OKEK'i hesaplanmıştır. Bulunan değer (OKEK) mesajın periyoduna (Pm) bölünerek, tekrar sayısı (Ts) hesaplanmıştır. Buna göre hesaplanan değerler Çizelge 3.2.'de gösterilmektedir.

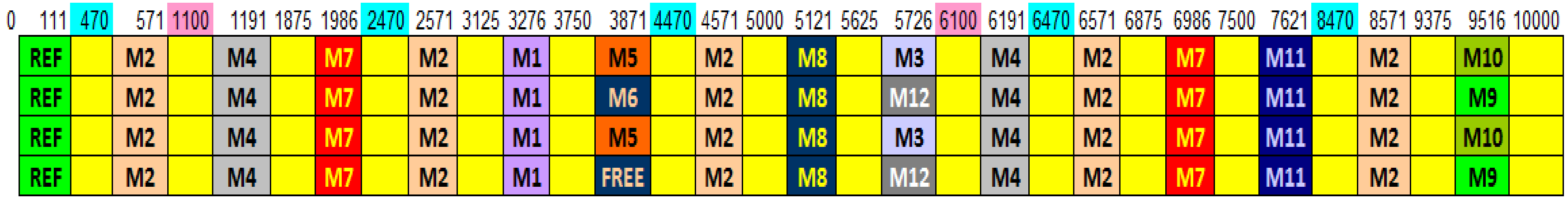
Çizelge 3.2. Matris çevrim değerleri

Mesaj No	Pm	OKEK (Pm ; 40)	Ts = (OKEK / Pm)
M1	10	40	4
M2	14	280	20
M3	20	40	2
M4	15	120	8
M5	20	40	2
M6	40	40	1
M7	15	120	8
M8	50	200	4
M9	20	40	2
M10	100	200	2
M11	50	200	4
M12	100	200	2

Mesajların tekrar sayıları (Ts) hesaplandıktan sonra, toplam tekrar sayısı 59 olarak bulunmuş, her temel çevrimin başında bulunması gereken 4 adet referans mesajı eklendiğinde toplam pencere sayısı 63 adet olmuştur ve Matris Çevriminin 64 pencereden oluşması planlanmıştır. Her temel çevrimin de 16 pencereden oluşması planlanmıştır. Altmış dördüncü (64.) pencerede tanımlı bir mesaj olmadığı için bir pencere Serbest *(Free)* olarak bırakılmıştır. Daha sonra mesajların bulunan sayıda, Matris Çevrimine yerleştirme planlaması yapılmıştır. Tasarımı yapılan Matris Çevrimi Şekil 3.4.'te görülmektedir. Zamanlama değerleri Şekil 3.5.'te görülmektedir.

Ref	M2	M4	M7	M2	M1	M5	M2	M8	M3	M4	M2	M7	M11	M2	M10
Ref	M2	M4	M7	M2	M1	M6	M2	M8	M12	M4	M2	M7	M11	M2	M9
Ref	M2	M4	M7	M2	M1	M5	M2	M8	M3	M4	M2	M7	M11	M2	M10
Ref	M2	M4	M7	M2	M1	Serbest	M2	M8	M12	M4	M2	M7	M11	M2	M9

Şekil 3.4. Tasarlanan matris çevrimi



Şekil 3.5. Mesaj pencerelerinin zamanlama değerleri

Mesaj pencerelerinin uzunluklarının hesaplanmasında oluşabilecek maksimum mesaj boyutu dikkate alınmıştır (Navet vd., 2000)

CAN paketinin iletilmesinde senkronizasyon, bit değerinin değişimleri sırasında sağlanır ve senkronizasyonun korunması için peş peşe en fazla 5 adet bitin aynı seviyede iletilmesine izin verilir. Aynı seviyede daha fazla bit olması durumunda, zıt yönde bir altıncı (6.) bit eklenir. Bu durum mesaj boyutunun uzamasına neden olur. Alıcı tarafta bu fazla bitler mesajdan atılır. Bu durumda maksimum mesaj boyutu (C_m) aşağıdaki şekilde hesaplanır (Navet vd., 2000 ; Tindell ve Burns, 1994).

$$C_m = \left(\left(\frac{34+8\text{sm}}{4}\right) + 47 + 8\text{sm}\right)\text{Tbit} \qquad (3.1)$$

Burada ;

34 bit: Senkronizasyon için bit ekleme yapılabilecek, sabit bit sayısını göstermektedir;

8sm: Byte olarak uzunluğu verilen veri boyutunun bit olarak hesaplanmasını sağlar.

4 bölen değeri: Senkronizasyon için eklenebilecek maksimum bit sayısının elde edilmesini sağlar.

47 sabiti: Standart CAN paketinde veri alanı olmadan bulunan sabit bit uzunluğunu ifade eder.

Tbit: Hattın bit zamanını ifade eder. Örneğin; 1 Mbps için 1µs, 500 Kbps için 2µs

Her mesaj penceresinin başlangıcında 16 bit uzunluğunda bir Tx aralığı, mesajın başlama aralığı olarak bulunur. Pencere boyutunun hesaplanmasında Maksimum mesaj uzunluğuna (C_m) Tx değeri eklenir.

Matris Çevriminde (Şekil 2.26.), mesajların dağılımı yapılırken pencereler arasında, eşit boşluklar olması planlanmıştır. Daha sonra, mesaj yerleşimleri yapılırken, pencere boyutlarındaki taşmaların önlenmesinde, bu boşluklar tasarım esnekliği sağlamaktadır.

4. BULGULAR VE İRDELEME

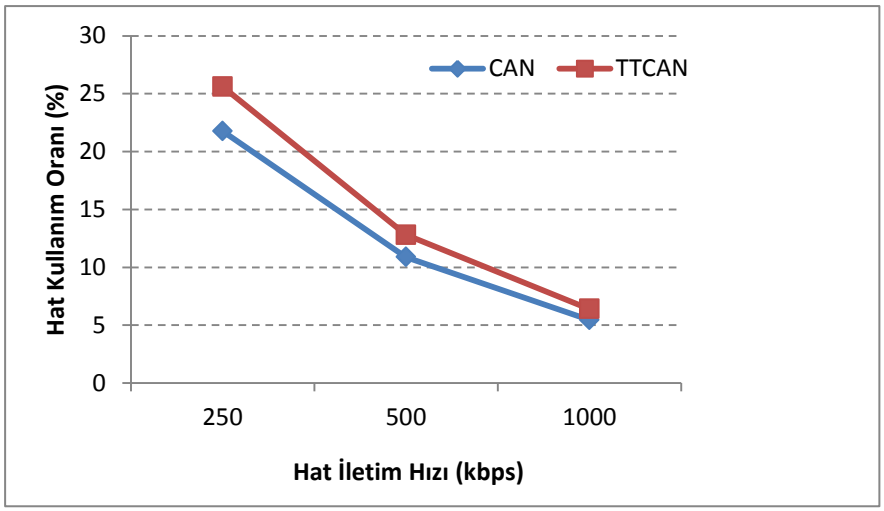
Bu bölümde tasarımı yapılan Denetleyici Alan Ağı ve Zaman Tetiklemeli Denetleyici Alan Ağı modellerinin simülasyon çalışmaları yapıldıktan sonra elde edilen sonuçlar çizelgeler ve grafikler yardımıyla incelenmekte ve veriler değerlendirilmektedir. Simülasyon sonuçları örneği EK C.'de verilmiştir.

4.1 Veriyolu Sonuçlarının Değerlendirilmesi

Çizelge 4.1.'de değişik veriyolu iletim hızlarında CAN ve TTCAN modellerinin simülasyonları sonucunda elde edilen veriyolu kullanım yüzdeleri ve hat erişim gecikmeleri görülmektedir. Elde edilen veriyolu kullanım yüzdelerinin daha iyi karşılaştırılması ve anlaşılabilmesi için sonuçlar görsel olarak Şekil 4.1.'de grafik üzerinde gösterilmiştir.

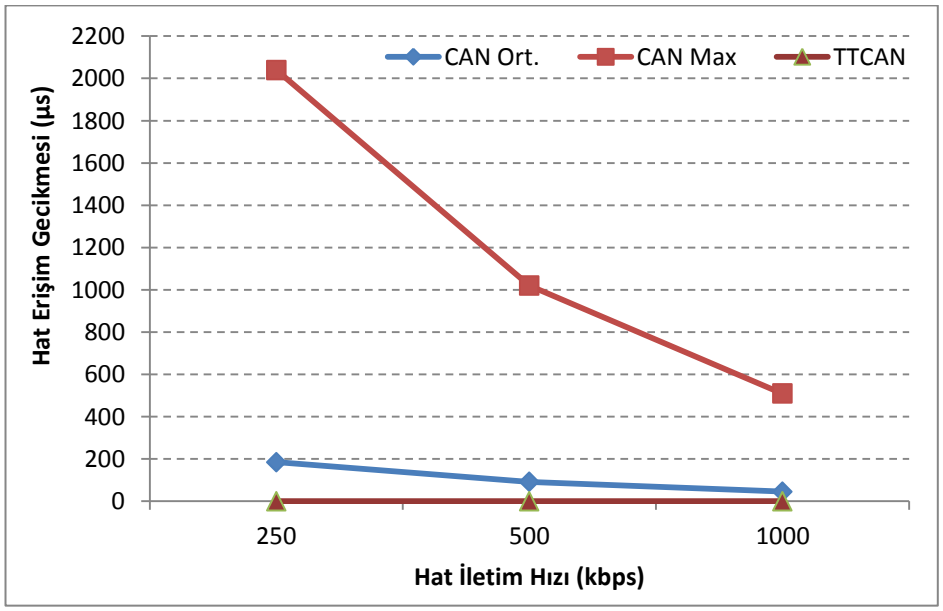
Çizelge 4.1. CAN-TTCAN hat kullanım yüzdeleri

TTCAN	**1 Mbps**	**500 Kbps**	**250 Kbps**
Ortalama Hat Erişim Gecikmesi (µs)	0	0	0
Maksimum Hat Erişim Gecikmesi (µs)	0	0	0
Standart Sapma	0	0	0
Hat Kullanım Yüzdesi	6,402%	12,804%	25,608%
CAN	**1 Mbps**	**500 Kbps**	**250 Kbps**
Ortalama Hat Erişim Gecikmesi (µs)	46,038	92,111	185,052
Maksimum Hat Erişim Gecikmesi (µs)	510	1020	2040
Standart Sapma	90	180,375	360,487
Hat Kullanım Yüzdesi	5,442%	10,884%	21,768%



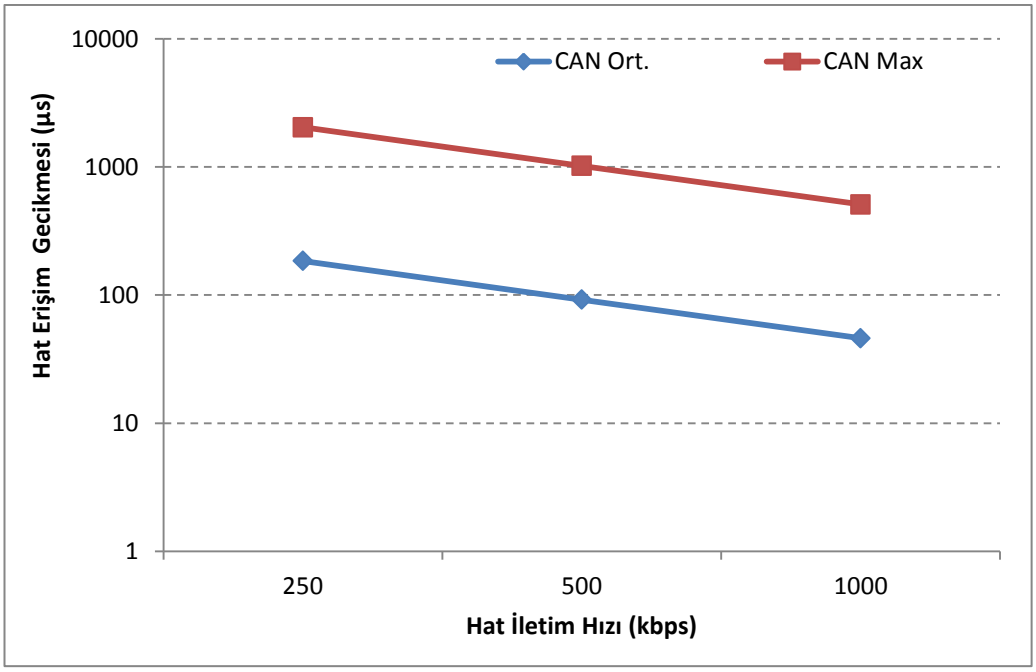
Şekil 4.1 CAN-TTCAN hat kullanım oranları

Hat iletim hızı azaldıkça hat kullanım yüzdeleri artmaktadır. Bunun sebebi düşük hat iletim hızında mesajların iletim sürelerinin artması dolayısıyla hattın meşgul olma süresinin artmasıdır. Tersi düşünüldüğünde de hat iletim hızı arttığında mesaj iletim süreleri azalacak ve dolayısıyla hat kullanım yüzdeleri daha düşük olacaktır. Çizelge 4.1.'de ve Şekil 4.1.'de görüleceği üzere TTCAN için hat kullanım yüzdeleri, CAN ağına göre bir miktar fazladır. Bunun sebebi, her temel çevrim (BC) başlangıcında, mesaj zamanlamalarındaki senkronizasyonu sağlamak amacıyla gönderilen referans mesajıdır. Buna karşılık TTCAN sisteminde, mesajlar özel mesaj pencerelerinde ve periyodik olarak gönderildiği için gecikmeler sıfırdır.



Şekil 4.2. CAN ve TTCAN sistemlerinde hat erişim gecikmeleri

Şekil 4.2.'de görüleceği üzere TTCAN sistemi için bütün hat iletim hızlarında hat erişim gecikmesi sıfırdır. CAN sisteminde ise ortalama gecikme ve maksimum gecikme değerleri, hat iletim hızı arttıkça azalmaktadır. Bunun sebebi Şekil 4.2.'de görüleceği gibi, hat iletim hızı arttıkça, hat kullanım yüzdesinin azalması ve bu sayede mesaj gönderime hazır olduğunda hattın boş bulunma olasılığının fazla olmasıdır. Ortalama erişim gecikmesi bir mesajın erişim için bekleyeceği ortalama süreyi ifade eder. Maksimum erişim gecikme süresi ise mesajın hat erişiminden önce karşılaştığı maksimum süreyi ifade eder. Maksimum erişim gecikme süresi mesajın iletim ömrü *(Deadline)* dikkate alındığında önemli bir yer tutar. Bu nedenle bu değerin dikkate alınması ve değerlendirilmesi gerekir. TTCAN sistemlerinde mesajların özel zaman pencerelerinde iletilmesi ve hat erişim gecikmesinin sıfır olması sayesinde bu sorunla karşılaşılmaz.



Şekil 4.3. CAN sisteminde ortalama ve maksimum erişim gecikmelerinin logaritmik ölçekte gösterimi

Şekil 4.3.'te görüleceği gibi ortalama ve maksimum erişim gecikme değerleri, hat iletim hızı arttıkça logaritmik olarak azalmaktadır. Gecikme değerleri logaritmik ölçekte gösterildiğinde karşılaştırma daha açık bir şekilde incelenebilmektedir.

4.2. TTCAN Modelinin Simülasyonu İle Elde Edilen Sonuçlar

TTCAN sistemi için tasarlanan matris çevrimi 3 ayrı hat iletim hızında test edilmiştir. Tasarlanan sistemle sağlanan performans gelişiminin analizini yapabilmek için CAN sistemi de aynı şekilde 3 ayrı hat iletim hızında test edilmiş ve sonuçlar karşılaştırılmıştır.

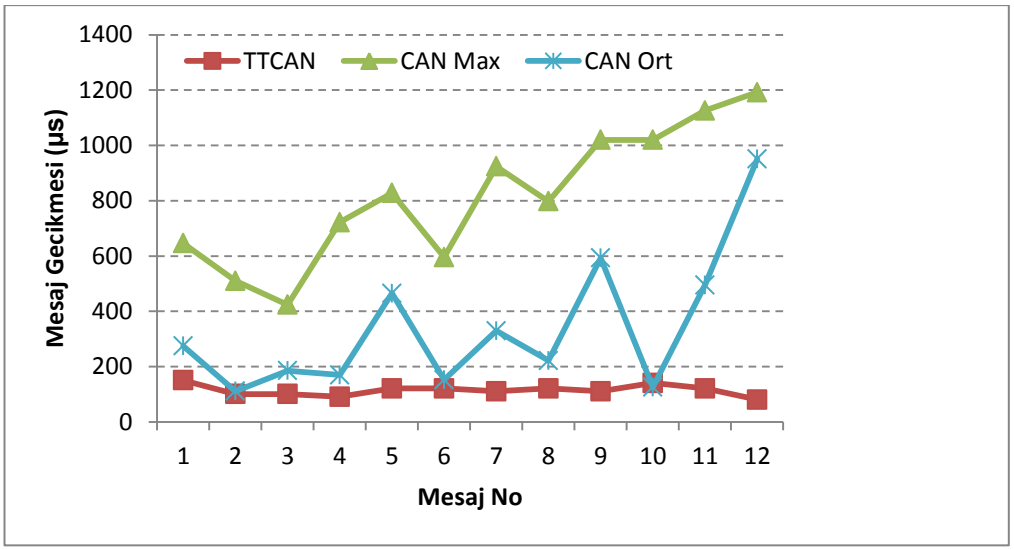
4.2.1. 1 Mbps hat hızında elde edilen sonuçlar

Geliştirilen TTCAN modelinin 1 Mbps hat iletim hızında simülasyonu gerçekleştirilmiş ve elde edilen sonuçlar Çizelge 4.2.'de gösterilmiştir.
Performans gelişmesinin karşılaştırmalı olarak incelenmesi amacıyla, CAN modeli ile ilgili simülasyonlar da gerçekleştirilmiş ve sonuçlar tabloya eklenmiştir.

Çizelge 4.2. 1Mbps veri iletim hızında TTCAN ve CAN sistemleri simülasyon değerleri

	TTCAN			CAN		
Hat Hızı	**1 Mbps**			**1 Mbps**		
Mesaj No	Maksimum İletim Süresi (µs)	Minimum İletim Süresi (µs)	Ortalama İletim Süresi (µs)	Maksimum İletim Süresi (µs)	Minimum İletim Süresi (µs)	Ortalama İletim Süresi (µs)
1	151	151	151	646	136	274,851
2	101	101	101	510	86	111,313
3	101	101	101	424	86	185,399
4	91	91	91	722	76	170
5	121	121	121	828	328	465,018
6	121	121	121	596	106	152,403
7	111	111	111	924	172	329,592
8	121	121	121	798	212	222,008
9	111	111	111	1020	424	593,019
10	141	141	141	1020	126	126,025
11	121	121	121	1126	106	495,31
12	81	81	81	1192	828	950,619

Çizelge 4.2.'de birinci sütunda PSA benchmark mesaj setinde bulunan mesajlar öncelik sırasına göre yukardan aşağıya doğru listelenmiştir. En yüksek önceliğe sahip mesaj ilk satırda yer almaktadır ve aşağıya doğru azalan öncelik sırasına göre mesajlar listelenmiştir. Çizelgede TTCAN ve CAN simülasyonlarından elde edilen minimum, maksimum ve ortalama mesaj iletim süreleri gösterilmektedir. TTCAN için maksimum, minimum ve ortalama mesaj iletim süreleri eşit olarak elde edilmiştir. Bunun sebebi, PSA benchmark mesaj setindeki tüm mesajların periyodik olması ve simülasyon modelinde her mesajın kendisine ayrılan özel mesaj penceresinde, hat erişim gecikmesi olmaksızın iletilmesidir. Her mesaj, matris çevrimi (MC) içerisinde, kendi mesaj penceresinde iletildiği için aynı anda hat üzerinde iletilen bir başka mesaj bulunmadığından herhangi bir hat erişim gecikmesi söz konusu değildir. Mesajlar öncelik elemesi veya herhangi bir sıra beklemesi olmaksızın gecikmesiz bir şekilde hat erişimi elde ederler. Bu nedenle burada mesaj iletim süresini belirleyen etken, mesajın boyutu (C_m) ve hat iletim hızıdır. CAN modelinde ise bu etkenlere ek olarak öncelik elemesinden ve mesajın gönderilmeye hazır olduğu anda hattın meşgul olma olasılığından kaynaklanan ortam erişim gecikmeleri de etkilidir. Bu nedenle CAN sistem modelinde elde edilen mesaj iletim gecikmeleri, TTCAN modeline göre çok daha yüksek değerlerdedir. CAN modelinde öncelik sırasının, mesaj gecikmelerinde önemli bir etken olduğu görülmektedir. Tabloda yukardan aşağıya doğru mesaj önceliği azaldıkça, CAN mesaj gecikmelerinin önemli oranda arttığı görülmektedir. TTCAN modelinde ise tüm mesajların öncelik değerlerinden bağımsız olarak, ortam erişim gecikmesi olmaksızın iletildiği görülmektedir. Çizelgedeki değerler daha iyi karşılaştırma yapabilmek için grafikler şeklinde incelenebilir.



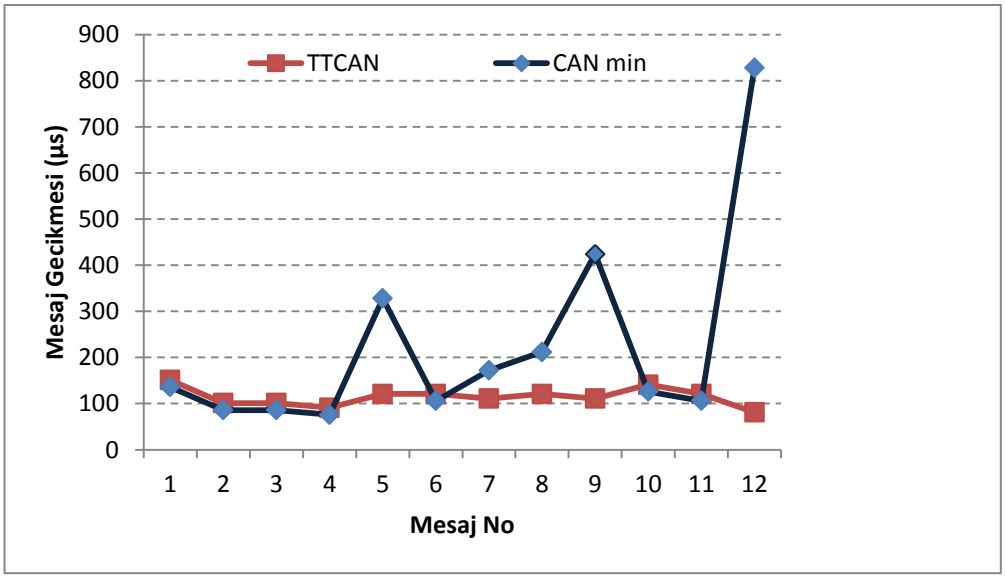
Şekil 4.4. 1 Mbps veri iletim hızında TTCAN, CAN maksimum ve ortalama mesaj iletim gecikmeleri değerleri

Şekil 4.4.'de 1 Mbps veri iletim hızında simülasyon modellerinde elde edilen mesaj iletim gecikme süreleri, mesaj öncelik sıra numaralarına göre verilmiştir. Grafikte yatay eksende mesajlar 1'den 12'ye kadar, en yüksek öncelikten en düşük önceliğe doğru sıralanmıştır. Dikey eksende ise mesajların iletim gecikme süreleri gösterilmiştir. Şekilde TTCAN modelinin, CAN modeliyle karşılaştırıldığında büyük bir performans gelişimi sağladığı görülmektedir. Şekilde "CAN Max" olarak görülen grafik eğrisi, CAN mesajlarının simülasyon sırasında karşılaştıkları maksimum iletim gecikmesi sürelerini vermektedir.

Bu grafik eğrisinden, mesajların önceliği azaldıkça iletim gecikme sürelerinin genellikle arttığı görülmektedir. Yüksek mesaj trafiğinin bulunduğu ortamlarda bu durum özellikle düşük öncelikli mesajların iletim ömrünü (Deadline) aşmasına sebep olabilir. Mutlak gerçek zamanlı sistemlerde bu sınırın aşılmaması büyük önem taşır. Bu maksimum değerlerle karşılaştırıldığında TTCAN modelinin büyük avantaj sağladığı görülmektedir. En yüksek öncelikli CAN mesajının (M1) maksimum iletim süresinin TTCAN modelindeki sürenin 4 katından fazla (4.278) olduğu, en düşük öncelikli CAN mesajının (M12) maksimum iletim süresinin ise 14 katından fazla (14.716) olduğu görülmektedir.

Şekilde "CAN Ort" olarak görülen grafik eğrisi, CAN mesajlarının simülasyon sırasında karşılaştıkları ortalama iletim gecikmesi sürelerini göstermektedir. Bu

değerler incelendiğinde CAN iletim ortalamalarının TTCAN modelindeki sürelerden fazla olduğu görülmektedir. Buradan TTCAN modelinin hemen tüm mesajlar için ortalama değerler açısından performans artışı sağladığı görülmektedir. M2, M6 ve M10 gibi bazı mesajlarda CAN ortalama değerlerinin TTCAN değerlerine yakın olduğu görülmektedir. Bunun sebebinin, tüm mesajların periyodik olarak tekrar etmesi nedeniyle bazı mesajların genellikle tekrarlanabilen, hattın erişime uygun olduğu boşluk anlarına denk gelmiş olabileceği düşünülebilir. Bu durumun, maksimum değerlerde, M2, M3, M6 mesajları üzerinde de etkili olduğu görülmektedir. Bu ortalama değerler karşılaştırıldığında, en yüksek öncelikli mesajın (M1) CAN modelindeki ortalama iletim süresinin, TTCAN modelindeki iletim süresinin (1.820) katı, en düşük öncelikli mesajın (M12) CAN modelindeki ortalama iletim süresinin ise, TTCAN modelindeki ortalama sürenin 11 katından fazla (11.736) olduğu görülmektedir.



Şekil 4.5. 1 Mbps veri iletim hızında TTCAN ve CAN minimum mesaj iletim gecikmesi değerleri

Şekil 4.5.'te TTCAN modelinin iletim süreleri ile CAN modelinin minimum iletim süreleri gösterilmiştir. CAN modelindeki minimum değerlerin bazılarının (M1, M2, M3, M4, M6, M10, M11) TTCAN modelindeki iletim sürelerinden daha düşük olduğu görülmüştür. Bunun sebebi, TTCAN Matris çevrimi tasarlanırken, mesajların boyutlarına (C_m), 15 bitlik Tx değerinin eklenmesidir. Tx değeri mesaj pencerelerinin başlangıcında bulunan ve mesajın iletilmeye başlaması için tanınan ve en fazla 16 bit uzunluğunda olabilen bir değerdir.

Burada en kötü ihtimalle 16.bit içerisinde iletilmeye başlandığı kabul edilmiş ve 15 bitlik bir gecikme süresi eklenmiştir. Bu nedenle TTCAN iletim sürelerinin, minimum CAN iletim süresinden 15 bit (15µs) daha fazla olduğu görülmektedir. Bu mesajlar için CAN modelinde minimum mesaj iletim süresinin, hesaplanan mesaj boyutu (C_m) için gereken gecikme süresine eşit olduğu görülmektedir. Burada minimum iletim süresinin, simülasyon sırasında bir kez bile gerçekleşen bir değer olabileceği düşünüldüğünde, performans karşılaştırması için belirleyici bir ölçüt değildir. Özellikle mutlak gerçek zamanlı sistemlerde performans analizi için esas belirleyici ölçüt en kötü *(Worst Case)* durum olan maksimum gecikme süreleridir. Hiçbir mesajın, o mesaj için tanınan iletim ömrü sınırını aşmaması gerekir.

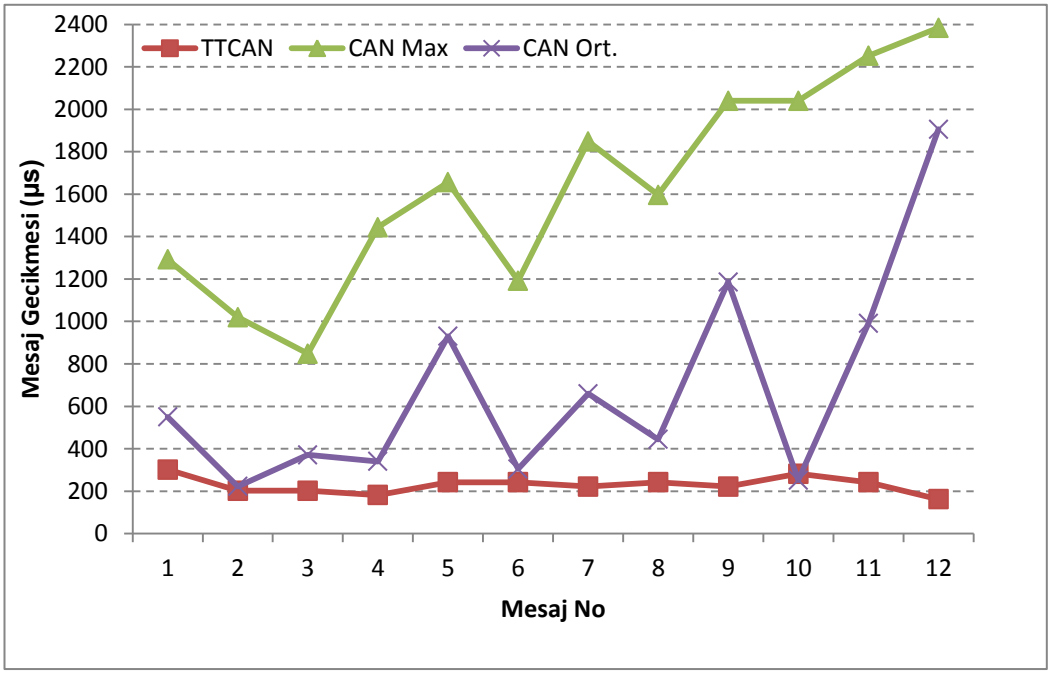
4.2.2. 500 kbps hat hızında elde edilen sonuçlar

500 kbps hat iletim hızında gerçekleştirilen simülasyon sonuçları Çizelge 4.3.'te gösterilmiştir. Aynı hat iletim hızında CAN modeli ile ilgili gerçekleştirilen simülasyon sonuçları da çizelgeye eklenmiştir.

Çizelge 4.3. 500 kbps hat iletim hızında TTCAN ve CAN sistemleri simülasyon değerleri

	TTCAN			CAN		
Hat Hızı	**500 Kbps**			**500 Kbps**		
Mesaj No	Maksimum İletim Süresi (µs)	Minimum İletim Süresi (µs)	Ortalama İletim Süresi (µs)	Maksimum İletim Süresi (µs)	Minimum İletim Süresi (µs)	Ortalama İletim Süresi (µs)
1	302	302	302	1292	272	549,702
2	202	202	202	1020	172	222,892
3	202	202	202	848	172	370,799
4	182	182	182	1444	152	340
5	242	242	242	1656	656	930,035
6	242	242	242	1192	212	304,806
7	222	222	222	1848	344	659,184
8	242	242	242	1596	424	444,016
19	222	222	222	2040	848	1186,037
10	282	282	282	2040	252	252,05
11	242	242	242	2252	212	990,619
12	162	162	162	2384	1656	1905,333

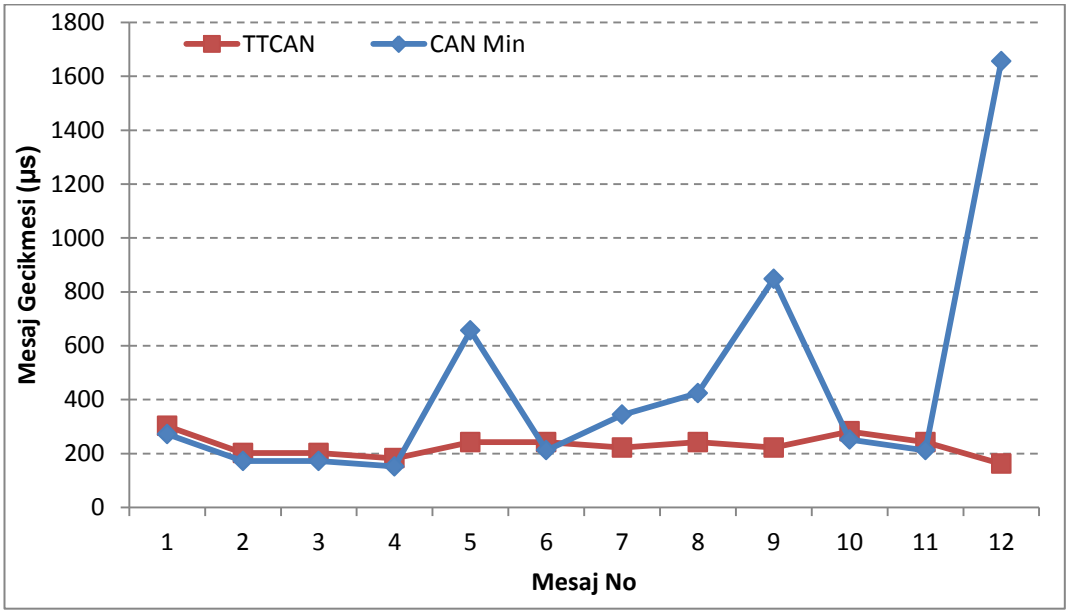
Çizelge 4.3.'ten 500 Kbps veri iletim hızında elde edilen sonuçların, 1 Mbps hızındaki sonuçlarla benzer olduğu görülmektedir. Ancak veri iletim hızının yarıya düşmesi sebebiyle bir bit için gerekli iletim süresi (C_m) iki katına çıkmaktadır.



Şekil 4.6. 500 kbps veri iletim hızında TTCAN, CAN maksimum ve ortalama mesaj iletim gecikmeleri değerleri

Şekil 4.6.'da 500 kbps veri iletim hızında simülasyon modellerinde elde edilen mesaj iletim gecikme süreleri, mesaj öncelik sıra numaralarına göre verilmiştir. Şekilde bu veri iletim hızında da TTCAN modelinin CAN modeline göre önemli derecede performans gelişmesi sağladığı görülmektedir. Şekilde TTCAN mesaj iletim gecikmeleri ve CAN Maksimum iletim gecikmeleri karşılaştırıldığında, 1 Mbps iletim hızında olduğu gibi 500 kbps iletim hızında da, en yüksek öncelikli CAN mesajının (M1) maksimum iletim süresinin TTCAN modelindeki sürenin 4 katından fazla (4.278) olduğu, en düşük öncelikli CAN mesajının (M12) maksimum iletim süresinin ise 14 katından fazla (14.716) olduğu görülmektedir. Bunun sebebi olarak, her iki iletim hızındaki modelin de aynı şartlar altında çalıştırılması ve tüm mesajların periyodik olmasından dolayı benzer tekrarlayan iletim paternlerinin ortaya çıkabilecek olması düşünülebilir. Ayrıca maksimum değer, simülasyon sırasında karşılaşılan bir tek durumun değerini gösterdiğinden, aynı durumun iki simülasyon modelinde de ortaya çıktığı düşünülebilir.

Ortalama değerler karşılaştırıldığında, 1 Mbps iletim hızında olduğu gibi 500 kbps hızında da, en yüksek öncelikli mesajın (M1) CAN modelindeki ortalama iletim süresinin, TTCAN modelindeki iletim süresinin (1.820) katı, en düşük öncelikli mesajın (M12), CAN modelindeki ortalama iletim süresinin ise, TTCAN modelindeki ortalama sürenin 11 katından fazla (11.761) olduğu görülmektedir.



Şekil 4.7. 500 kbps veri iletim hızında TTCAN ve CAN minimum mesaj iletim gecikmesi değerleri

Şekil 4.7.'de görüldüğü gibi 500 Kbps veri iletim hızında da 1 Mbps veri iletim hızında olduğu gibi, CAN modelindeki minimum değerlerin bazılarının (M1, M2, M3, M4, M6, M10, M11) TTCAN modelindeki iletim sürelerinden daha düşük olduğu görülmüştür. Bunun sebebi 1 Mbps hızında olduğu gibi, TTCAN Matris çevrimi tasarlanırken, mesajların boyutlarına (C_m), 15 bitlik Tx değerinin eklenmesidir. Ancak 1 Mbps veri iletim hızından farklı olarak, 500 Kbps veri iletim hızında aradaki farkın 30 µs (30 = 15 bit * 2µs) olduğu görülmektedir.

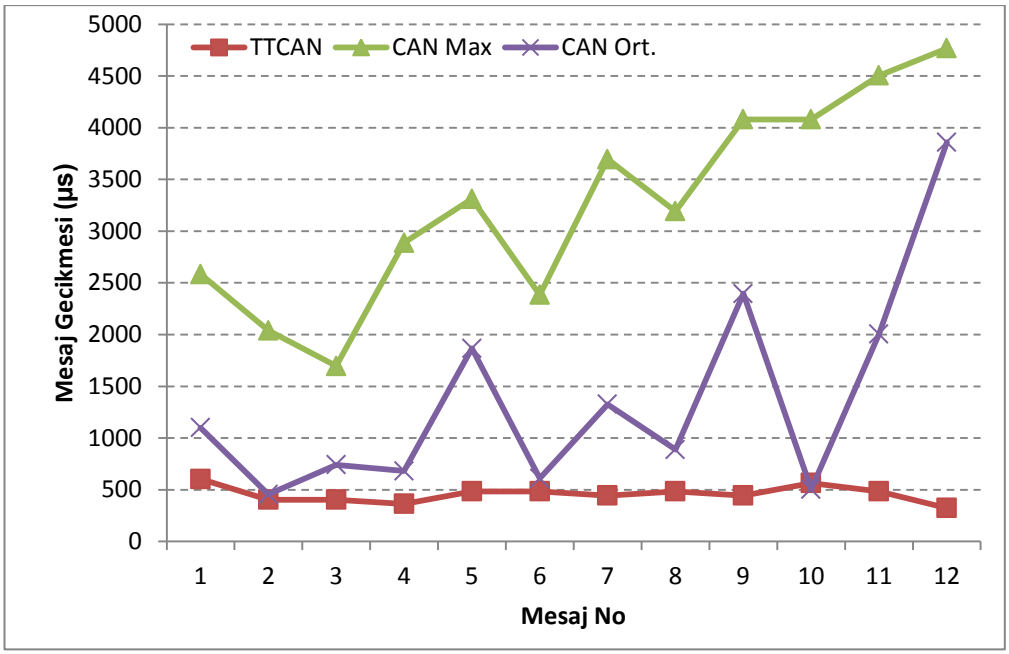
4.2.3. 250 Kbps hat hızında elde edilen sonuçlar

250 Kbps hat iletim hızında gerçekleştirilen simülasyon sonuçları Çizelge 4.4.'de gösterilmiştir. Aynı hat iletim hızında CAN modeli ile ilgili gerçekleştirilen simülasyon sonuçları da çizelgeye eklenmiştir.

Çizelge 4.4. 250 kbps hat iletim hızında TTCAN ve CAN sistemleri simülasyon değerleri

	TTCAN			CAN		
Hat Hızı	**250 Kbps**			**250 Kbps**		
Mesaj No	Maksimum İletim Süresi (µs)	Minimum İletim Süresi (µs)	Ortalama İletim Süresi (µs)	Maksimum İletim Süresi (µs)	Minimum İletim Süresi (µs)	Ortalama İletim Süresi (µs)
1	604	604	604	2584	544	1099,404
2	404	404	404	2040	344	454,451
3	404	404	404	1696	344	741,597
4	364	364	364	2888	304	681,309
5	484	484	484	3312	1312	1866,621
6	484	484	484	2384	424	609,611
7	444	444	444	3696	688	1325,738
8	484	484	484	3192	848	888,031
9	444	444	444	4080	1696	2395,006
10	564	564	564	4080	504	504,099
11	484	484	484	4504	424	2005,806
12	324	324	324	4768	3312	3859,801

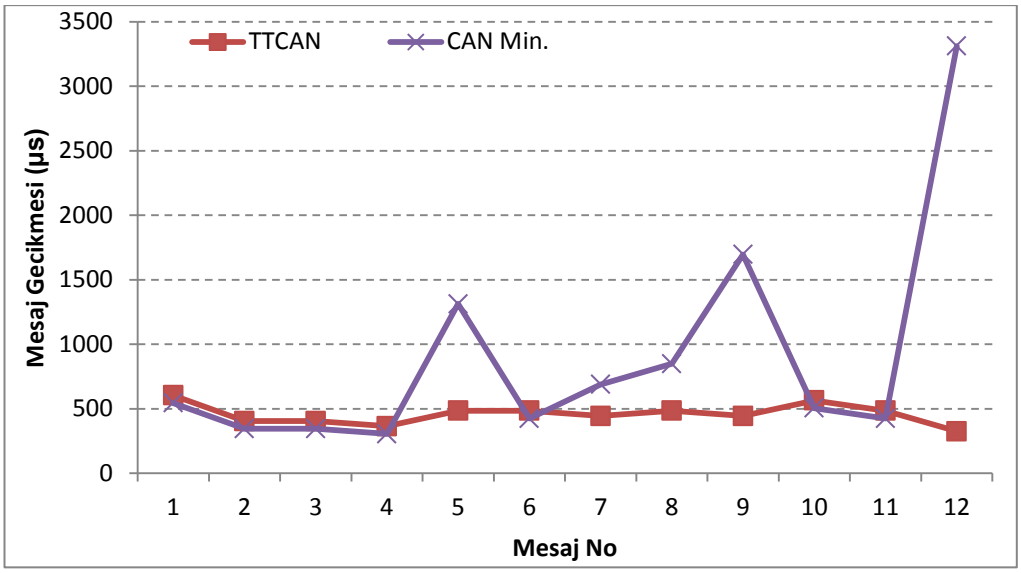
Çizelge 4.4.'ten 250 kbps veri iletim hızında elde edilen sonuçların, 1 Mbps ve 500 kbps hızlarındaki sonuçlara benzer özellikler gösterdiği görülmektedir. Ancak veri iletim hızının daha düşük olması sebebiyle bir bit için gerekli iletim süresi (C_m) 500 kbps hızına göre 2 kat, 1 Mbps hızına göre ise 4 kat daha yüksek değerler almaktadır.



Şekil 4.8. 250 kbps veri iletim hızında TTCAN, CAN maksimum ve ortalama mesaj iletim gecikmeleri değerleri

Şekil 4.8.'de 250 kbps veri iletim hızında CAN ve TTCAN modellerinin maksimum ve ortalama gecikme sonuçları, öncelik sırasına göre grafik şeklinde görülmektedir. 1 Mbps ve 500 kbps iletim hızlarında olduğu gibi, 250 kbps iletim hızında da TTCAN modelinin yüksek performans gelişmesinin sürdürüldüğü görülmektedir. TTCAN mesaj iletim gecikmeleri ile CAN maksimum mesaj iletim gecikmeleri karşılaştırıldığında, en yüksek önceliği sahip mesajın (M1), CAN modelindeki mesaj iletim gecikmesi, TTCAN modelindeki iletim gecikmesinin 4 katından fazla (4.278), en düşük önceliğe sahip mesajın (M12) iletim gecikmesinde ise bu oranın 14 katından fazla (14.716) olduğu görülmüştür.

TTCAN mesaj iletim gecikmeleri ile, CAN ortalama iletim gecikmesi değerleri karşılaştırıldığında, en yüksek önceliğe sahip mesajın (M1), CAN modelindeki ortalama mesaj iletim gecikmesi, TTCAN modelinin mesaj iletim gecikmesinin (1.820) katı, en düşük önceliğe sahip mesajın (M12) iletim gecikmesinde ise bu oranın 11 katından fazla (11.912) olduğu görülmüştür.



Şekil 4.9. 250 kbps veri iletim hızında TTCAN ve CAN minimum mesaj iletim gecikmesi değerleri

Şekil 4.9.'da 250 Kbps veri iletim hızında da 1 Mbps ve 500 kbps veri iletim hızlarında olduğu gibi, CAN modelindeki minimum değerlerin bazılarının (M1, M2, M3, M4, M6, M10, M11) TTCAN modelindeki iletim sürelerinden daha düşük olduğu görülmüştür. Bunun sebebi TTCAN Matris çevrimi tasarlanırken, mesajların boyutlarına (C_m), 15 bitlik Tx değerinin eklenmesidir. Ancak 250 kbps veri iletim hızında aradaki farkın 60 µs (15 bit x 4 µs) olduğu görülmektedir.

Çizelge 4.5.'te tüm hat iletim hızlarında, tüm mesajlar için CAN modelindeki maksimum ve ortalama iletim gecikme sürelerinin, TTCAN modelindeki iletim gecikme sürelerine oranları görülmektedir. Buradan, maksimum mesaj gecikmeleri dikkate alındığında tüm mesajlar için TTCAN modelinin çok daha iyi performans sağladığı görülmektedir.

Ortalama mesaj gecikmeleri dikkate alındığında ise, bir mesaj (M10) dışında tüm mesajlar için TTCAN modelinin yine daha iyi performans sağladığı görülmektedir. Burada M10 mesajının tekrarlayan mesaj iletim paterni içerisinde, çoğunlukla hattın boş olduğu bir anda iletildiği düşünülebilir. Tekrarlayan iletim paternleri tüm hızlarda benzer şekilde ortaya çıkmaktadır. Bu sebeple, maksimum gecikme süreleri dikkate alındığında, tüm iletim hızlarındaki oranlar aynı kalmaktadır.

Bunun yanında, ortalama gecikme süreleri dikkate alındığında ise, mesaj çakışmalarındaki farklılıkların çok az oranda olduğu düşünülebilir. Bundan dolayı ortalama değer oranlarında çok küçük farkların ortaya çıktığı görülmektedir.

Çizelge 4.5. Tüm hat iletim hızlarında CAN-TTCAN modelleri gecikme oranları

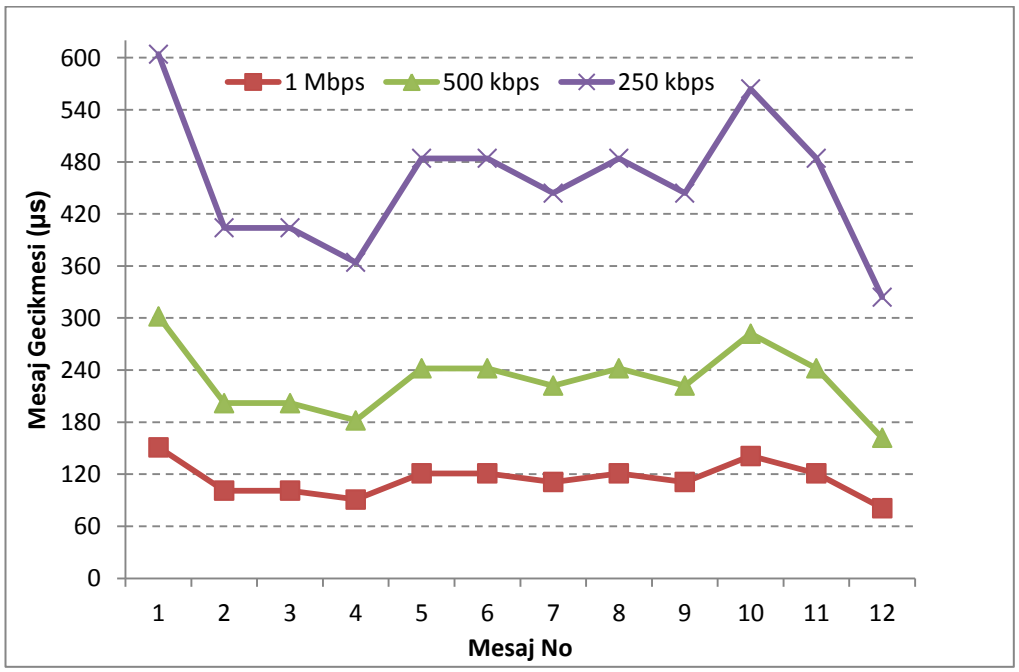
Hat İletim Hızı	1 mbps		500 kbps		250 kbps	
Mesaj No	Maksimum Gecikme Oranı	Ortalama Gecikme Oranı	Maksimum Gecikme Oranı	Ortalama Gecikme Oranı	Maksimum Gecikme Oranı	Ortalama Gecikme Oranı
1	4,278145695	1,820205298	4,278145695	1,820205298	4,278145695	1,820205298
2	5,04950495	1,102108911	5,04950495	1,103425743	5,04950495	1,124878713
3	4,198019802	1,835633663	4,198019802	1,835638614	4,198019802	1,835636139
4	7,934065934	1,868131868	7,934065934	1,868131868	7,934065934	1,871728022
5	6,842975207	3,843123967	6,842975207	3,843119835	6,842975207	3,856654959
6	4,925619835	1,259528926	4,925619835	1,259528926	4,925619835	1,25952686
7	8,324324324	2,969297297	8,324324324	2,969297297	8,324324324	2,985896396
8	6,595041322	1,83477686	6,595041322	1,83477686	6,595041322	1,834774793
9	9,189189189	5,342513514	9,189189189	5,342509009	9,189189189	5,394157658
10	7,234042553	0,893794326	7,234042553	0,893794326	7,234042553	0,893792553
11	9,305785124	4,093471074	9,305785124	4,093466942	9,305785124	4,144227273
12	14,71604938	11,73603704	14,71604938	11,76131481	14,71604938	11,91296605

4.2.4. CAN ve TTCAN modellerindeki mesaj iletim gecikmelerinin, hat iletim hızlarına göre incelenmesi

Mesajların tüm hat iletim hızlarındaki iletim gecikmeleri, öncelik sıralarına göre yerleştirilmiş olarak, TTCAN modelindeki iletim gecikmeleri Çizelge 4.6.'da gösterilmiştir. TTCAN modelinde, mesajların maksimum, minimum ve ortalama değerleri eşittir. Bunun sebebi mesajların kendileri için ayrılan zaman pencerelerinde, periyodik olarak iletilmeleridir. Ayrıca mesajların iletim gecikmeleri, hat iletim hızının yarıya düşmesiyle, bir önceki iletim hızındaki değere göre iki katına çıkmaktadır. Şekil 4.10.'da bu durum grafiksel olarak görülmektedir.

Çizelge 4.6. TTCAN modelinde tüm mesajların, tüm hat iletim hızlarındaki gecikmeleri

TTCAN	Hat İletim Hızı		
Mesaj NO	1 Mbps	500 kbps	250 kbps
1	151	302	604
2	101	202	404
3	101	202	404
4	91	182	364
5	121	242	484
6	121	242	484
7	111	222	444
8	121	242	484
9	111	222	444
10	141	282	564
11	121	242	484
12	81	162	324



Şekil 4.10. TTCAN modelinde tüm mesajların, tüm hat iletim hızlarındaki gecikmeleri

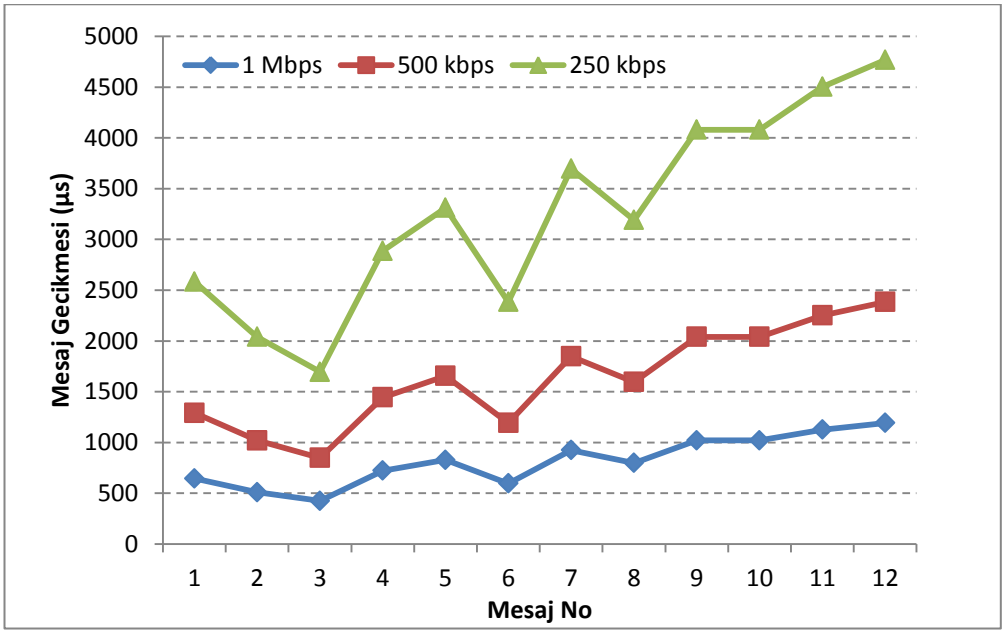
CAN modelinde mesajların iletim gecikmesi değerleri, TTCAN modelinden farklı olarak, tüm hat iletim hızlarında eşit değildir. Bu sebeple CAN modelinde, mesajların iletim gecikmeleri, maksimum, minimum ve ortalama değerler üzerinden incelenmiştir.

Çizelge 4.7.'de mesajların, CAN modelinde tüm hat iletim hızlarındaki maksimum iletim gecikmeleri gösterilmiştir. Mesajların iletim gecikmeleri değerleri TTCAN modelinde olduğu gibi, CAN modelinde de, hat iletim hızının yarıya düşmesiyle, bir önceki iletim hızındaki değere göre iki katına çıkmaktadır. Bunun sebebi, daha önce belirtildiği gibi, tüm mesaj setinin periyodik mesajlardan oluşması ve dolayısıyla sürekli tekrarlayan iletim paternlerinin ortaya çıkmasıdır. Periyodik olmayan mesajlar içeren bir mesaj seti için bu durum farklı olacaktır. Şekil 4.11.'de tüm hat iletim hızlarında, mesajların maksimum iletim gecikme değerleri grafik şeklinde görülmektedir.

Çizelge 4.8.'de CAN modelinde tüm mesajların, tüm hat iletim hızlarındaki minimum gecikme değerleri görülmektedir. Minimum gecikme değeri, tüm simülasyon sırasında ortaya çıkan en düşük değerdir ve (C_m) eşitliğiyle hesaplanabilir. Bu değer mesajın ortaya çıkmasıyla, hiç gecikmesiz hat erişiminin elde edilmesi durumundaki değerdir. C_m değerinin hesaplamasında, hat iletim hızının değişmesiyle sadece (Tbit) değeri iki katına çıkacağı için mesajın minimum gecikme süresi de önceki hat iletim hızına göre iki katına çıkmaktadır. Şekil 4.12.'de ise minimum mesaj gecikme süreleri, tüm iletim hızlarında grafiksel olarak görülmektedir.

Çizelge 4.7. CAN modelinde tüm mesajların, tüm hat iletim hızlarındaki maksimum gecikmeleri

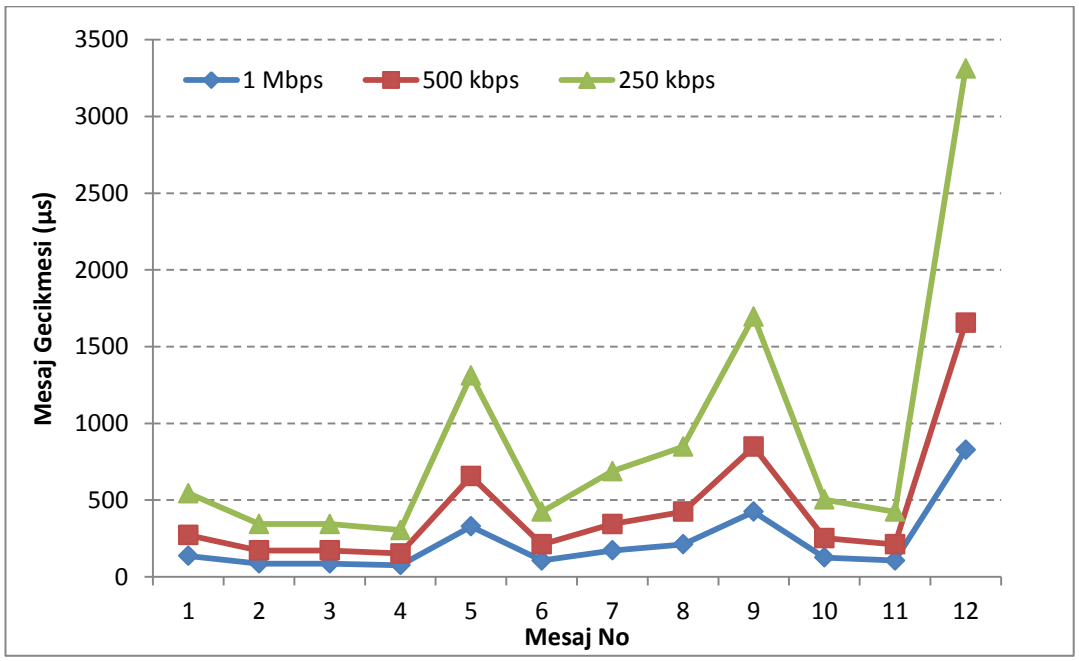
CAN Mak.	Hat İletim Hızı		
Mesaj No	**1 Mbps**	**500 kbps**	**250 kbps**
1	646	1292	2584
2	510	1020	2040
3	424	848	1696
4	722	1444	2888
5	828	1656	3312
6	596	1192	2384
7	924	1848	3696
8	798	1596	3192
9	1020	2040	4080
10	1020	2040	4080
11	1126	2252	4504
12	1192	2384	4768



Şekil 4.11. CAN modelinde tüm mesajların, tüm hat iletim hızlarındaki maksimum gecikmeleri

Çizelge 4.8. CAN modelinde tüm mesajların, tüm hat iletim hızlarındaki minimum gecikmeleri

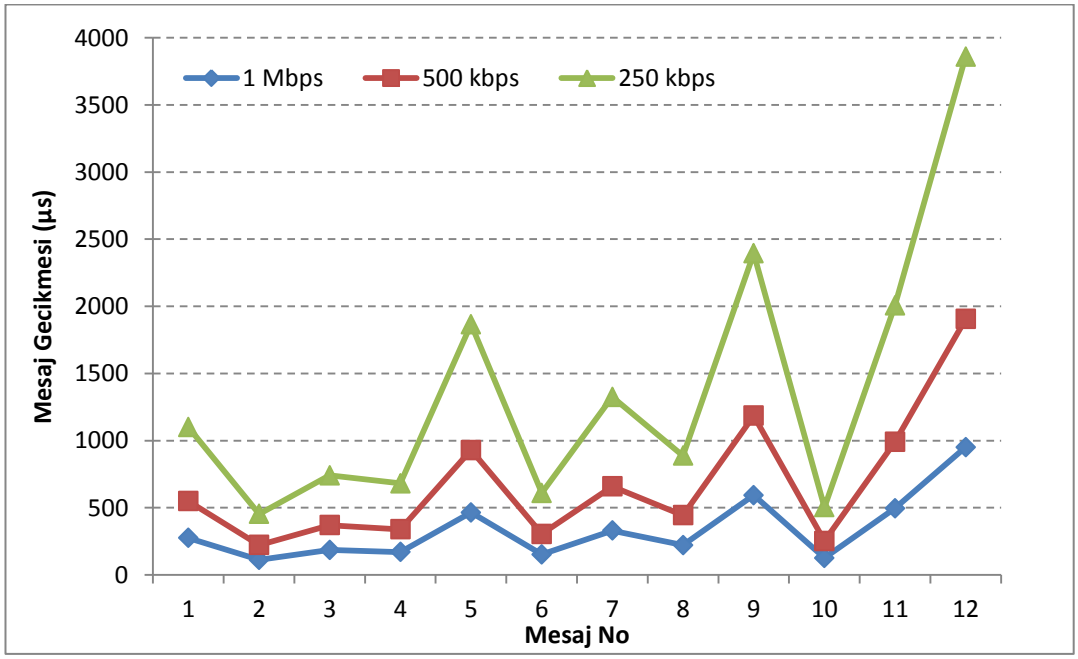
CAN Min.	Hat İletim Hızı		
Mesaj No	1 Mbps	500 kbps	250 kbps
1	136	272	544
2	86	172	344
3	86	172	344
4	76	152	304
5	328	656	1312
6	106	212	424
7	172	344	688
8	212	424	848
9	424	848	1696
10	126	252	504
11	106	212	424
12	828	1656	3312



Şekil 4.12. CAN modelinde tüm mesajların, tüm hat iletim hızlarındaki minimum gecikmeleri

Çizelge 4.9. CAN modelinde tüm mesajların, tüm hat iletim hızlarındaki ortalama gecikmeleri

CAN Ort.	Hat İletim Hızı		
Mesaj No	**1 Mbps**	**500 kbps**	**250 kbps**
1	274,851	549,702	1099,404
2	111,313	222,892	454,451
3	185,399	370,799	741,597
4	170	340	681,309
5	465,018	930,035	1866,621
6	152,403	304,806	609,611
7	329,592	659,184	1325,738
8	222,008	444,016	888,031
9	593,019	1186,037	2395,006
10	126,025	252,05	504,099
11	495,31	990,619	2005,806
12	950,619	1905,333	3859,801



Şekil 4.13. CAN modelinde tüm mesajların, tüm hat iletim hızlarındaki ortalama gecikmeleri

Çizelge 4.9.'da CAN modelinde tüm mesajların, tüm hat iletim hızlarındaki ortalama gecikme değerleri görülmektedir. Ortalama gecikme süresi değerleri, minimum ve maksimum değerlere benzer şekilde bir artış göstermektedir. Ancak bu artışın genellikle bir önceki hızdaki gecikmeye göre tam iki kat değil, yaklaşık iki kat olduğu görülmektedir. Bu oranlar Çizelge 4.10.'da görülmektedir. Minimum ve maksimum değerler tüm simülasyon sırasında bir kez ortaya çıksa bile alınan değerlerdir. Ortalama değerler ise, tüm mesaj gecikme değerlerinin ortalamasını vermektedir. Simülasyon sırasında farklı iletim hızlarında karşılaşılan değerlerin ortalamalarının oranı yakın fakat farklı değerler almaktadır. Şekil 4.13.'de CAN modelinde tüm mesajların, tüm hat iletim hızlarındaki ortalama gecikme süreleri görülmektedir.

Çizelge 4.10. CAN modelinde mesajların ortalama iletim gecikmelerinin hat iletim hızlarına göre oranları

Mesaj No	500 kbps-250 kbps oranı	1 Mbps-500 kbps oranı
1	2	2
2	2,038884303	2,002389658
3	1,999997303	2,000005394
4	2,00385	2
5	2,007043821	1,99999785
6	1,999996719	2
7	2,01118049	2
8	1,999997748	2
9	2,019334979	1,999998314
10	1,999996033	2
11	2,024800655	1,999997981
12	2,025788143	2,004307719

5. SONUÇLAR VE ÖNERİLER

Bu çalışmada, Zaman Tetiklemeli Denetleyici Alan Ağı için, azaltılmış matris yöntemine dayalı olarak geliştirilmiş bir matris çevriminin performans analizinin gerçekleştirilmesi amaçlanmıştır. Bu amaçla, PSA Benchmark mesaj seti kullanılarak bir matris çevrimi geliştirilmiş ve simülasyon modeline uygulanmıştır.

CAN protokolü öncelik eleme esaslı ortam erişim yöntemi kullandığı için özellikle düşük öncelikli mesajların iletiminde büyük gecikmelere sebep olabilir. Gerçek zamanlı sistemlerde mesajların verilen süre içerisinde iletilmesi gerekir. CAN protokolünde ortaya çıkan bu gecikmeleri önlemek için, her mesaj için özel iletim pencerelerinin bulunduğu, zaman tetiklemeli denetleyici alan ağı (TTCAN) modeli kullanılabilir. Özellikle tümüyle periyodik mesajlardan oluşan mesaj setlerinde bu yöntem büyük performans artışı sağlar.

Bu çalışmada, tümüyle periyodik mesajlardan oluşan ancak periyotların minimum mesaj periyodunun 2^n şeklinde olması şartını sağlamadığı PSA Benchmark mesaj seti kullanılmıştır. PSA benchmark mesaj setine ilk defa uygulanan azaltılmış matris çevrimi yöntemiyle, mesajlar matris çevrimine sorunsuz bir şekilde yerleştirilebilmiştir. Sağlanan performans gelişmesinin incelenebilmesi için bu mesaj setiyle, standart CAN protokolü ve erişim yönteminin kullanıldığı bir simülasyon modeli geliştirilmiştir. Standart CAN protokolü için elde edilen sonuçlar ile TTCAN modeli için elde edilen sonuçlar karşılaştırılmış ve performans gelişmesinin analizi yapılmıştır. Performans analizi, TTCAN modeli için matris çevrimi tasarımının yapılabildiği, 1 Mbps, 500 kbps ve 250 kbps hızlarında gerçekleştirilmiştir.

Gerçekleştirilen bu performans analizleri sonucunda, tasarlanan bu matris çevrimi ile mesajlar pencere sürelerinde herhangi bir çakışma veya taşma olmadan ve herhangi bir gecikme olmadan iletilebilmiştir.

Tasarlanan TTCAN modelinin, standart CAN modeline göre, maksimum mesaj iletim gecikmelerinde, en düşük öncelikli mesaj için % 93 oranında, en yüksek öncelikli mesaj için % 25 oranında performans artışı sağladığı görülmüştür.

Bu çalışmada kullanılan azaltılmış matris yöntemi kullanılarak farklı mesaj setleri için performans analizleri gerçekleştirilebilir. Ayrıca Esnek Veri Oranlı Denetleyici Alan Ağı (CAN-FD) standardının kullanıldığı bir matris tasarımı gerçekleştirilerek, performans analizleri incelenebilir.

KAYNAKLAR

Albert A. (2004) Comparison of event-triggered and time-triggered concepts with regard to distributed control system, *Proceedings of the embedded world*, p. 235–52.

Albert A. ve Hugel R. (2005) Heuristic scheduling concepts for TTCAN networks. *Proceedings of the international CAN conference*.

Anonim, *CAN Specification, Version 2.0*, Robert Bosch GmbH, Germany, 1991.

Anonim, *CAN with Flexible Data-Rate White Paper Version 1.1*, Robert Bosch GmbH, Germany, 2011.

Anonim1, *Introduction to In-Vehicle Networking*, MCS96 papers, Intel Corporation, 1998.

Anonim2, *CAN-based Computer System for Railways*, CAN Newsletter , CAN in Automation (CiA), Germany, 1998.

Anonim1, *CAN Sales Figures*, Press Releases, CAN in Automation (CiA), Germany, 1999.

Anonim2, *NMEA 2000: CAN in Marine Electronics*, CAN Newsletter , CAN in Automation (CiA), Germany, 1999, pp. 52-53.

Anonim3, *CAN Data Link Layer*, CiA CAN Presentations, www.can-cia.de, Germany, 1999.

Baba, M.D. (1996) *Fault Tolerance in Distributed Real-Time Computer Systems*, PhD Thesis, University of Sussex.

Castelpietra, P., Song, Q., Simonot, F. ve Cayrol, O. (2000) Performance Evaluation of a Multiple Networked in-Vehicle Embedded Architecture *WFSC2000*

Ding, S., Xie, Z. ve Yin, X. (2008) A GA-based systematic message scheduling method for time- triggered CAN, *IEEE/IFIP international conference on embedded and ubiquitous computing*.

Etschberger, K. (1997) CAN-based Higher Layer Protocols and Profiles, *4th International CAN Conference*, Germany.

Fonseca J., Cotinho F. ve Barreiros J. (2001) Scheduling for a TTCAN network with a stochastic optimization algorithm, *Proceedings of the international CAN conference.*

Fredriksson, L. B. (1997) , *"Distributed Embedded Control Systems in Robotics"*, KVASER AB. Sweden.

Hartwich, F.(2012) CAN with Flexible Data-Rate, *ICC2012, CiA*

Hofstee, J. W. ve Goense, D. (1997), Simulation of a CAN-based Tractor-Implement Field Bus According to DIN9684, *J. Agric. Engng. Res*., pp. 89-100.

Johansson R. (2004) Time and event triggered communication scheduling for automotive applications. *Technical report 17*. Goteborg, Sweden: Chalmers Lindholmen University College.

Lawrenz, W. (1997) *CAN System Engineering From Theory to Practical Applications*, Springer-Verlag Inc., New York,.

Leen G. ve Heffernan, D. (2002), TTCAN: a new time-triggered controller area network, *Microprocessors and Microsystems*, 26(2),77–94

Lennartsson, K. ve Fredriksson, L., B. (1995) "Fundamental Parts in SDS, DeviceNet and CAN Kingdom", *2nd International CAN Conference*, London, UK.

Mohammad, U., ve Holou, N. (2010) Development of An Automotive Communication Benchmark, *Canadian Journal on Electrical and Electronics Engineering Vol. 1, No. 5*, p 99-115

Navet, N., Song, Y.Q. ve Simonot, F. (2000) Worst-case deadline failure probability in real-time applications distributed over controller area network, *Journal of System Architecture*,46(7), 607–17

Naughton M. ve Heffernan D. (2005) SMART-Plan: a new message scheduler for real-time control networks, *IEE Irish signals and systems conference*.

Oertel, H., J. (2012) Using CAN with flexible data-rate in CANopen systems, *ICC2012, CiA.*

Qiao X., Wang K., Sun Y., Huang W. ve Wang F. (2007) A genetic algorithms based optimization for TTCAN. *International conference on vehicular electronics and safety (ICVES).*

Ryu M. (2009) A scheduling algorithm for minimizing exclusive window durations in time-triggered controller area network. *IEICE Transactions on Communications* E92.B(8),2739–42

Schmidt, K. ve Schmidt, E.,G. (2007) Systematic message schedule construction for time- triggered CAN, *IEEE Transaction on Vehicular Technology* ;56(6): 3431–41.

Schofield, M.,J. (1996) Controller Area Network, How CAN Works, Error Handling CAN Application Layers.

Stock, M. (1999), Higher-Layer Protocol for Avionics, *CAN Newsletter*, pp 28, 30, 32, 34

Tenruh, M. (2001), *Extending Controller Area Networks, CAN cut-trough bridging, CAN over ATM and CAN based ATM Fieldbus*, PhD thesis, The University of Sussex.

Tenruh, M. (2011) Message Scheduling with Reduced Matrix Cycle and Evenly Distributed Sparse Allocation for Time-Triggered CAN, *Journal of Network and Computer Applications*, Vol. 34, pp. 1240-1251.

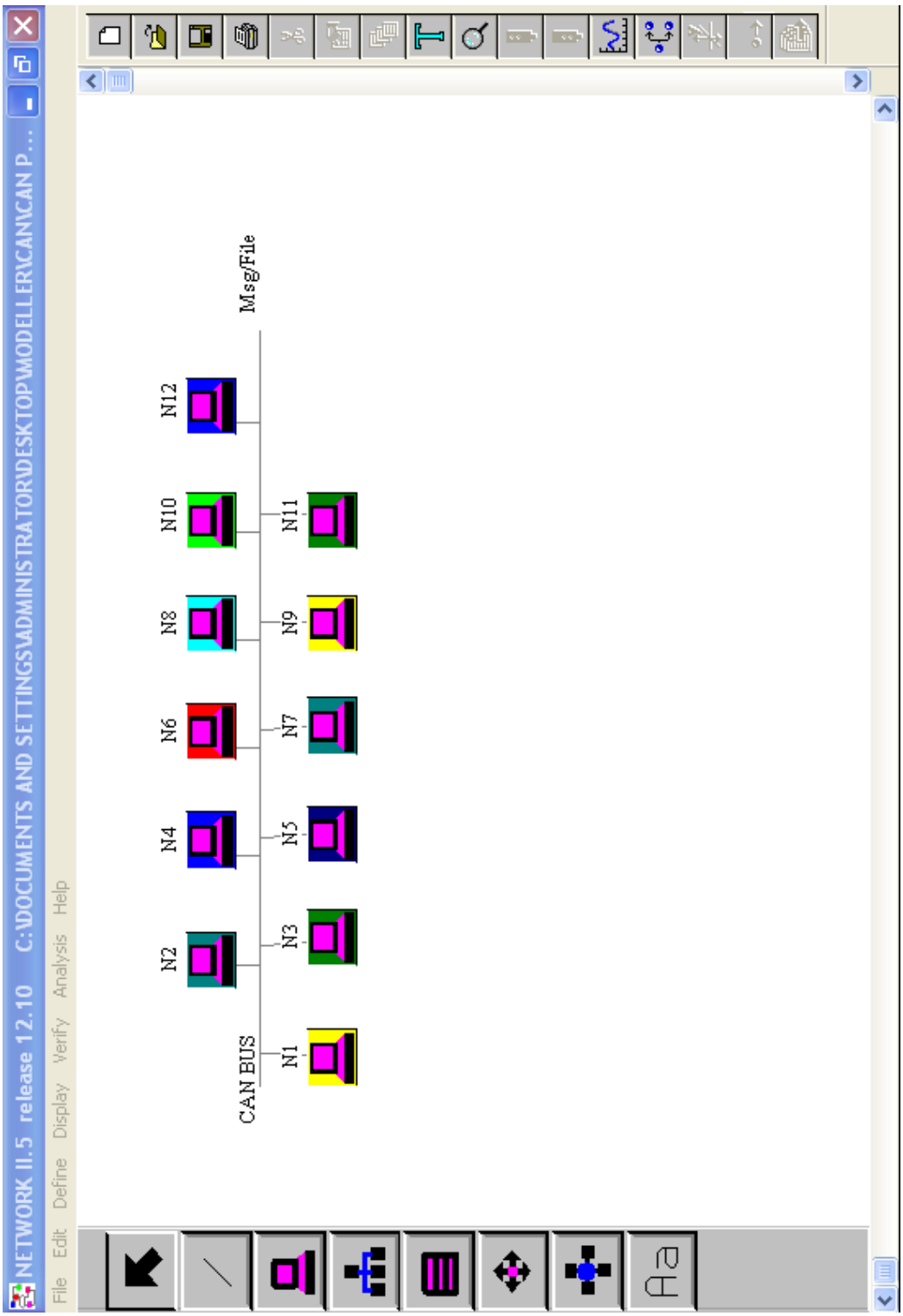
Tindell, K. ve Burns, A. (1994) Guaranteed message latencies for distributed safety-critical hard real-time control networks, *Technical report YCS229, Department of Computer Science, University of York.*

Wenkebach U. ve Reckels B. (1989) System Concepts for Serial Data Communication in Cars, *Proceeding of the Institution of Mechanical Engineer, Seventh International Conference Automotive Electronics*, pp. 203-212

Xiao, T., Li, X., Tan, X. ve Zhou, X. (2010) Real-time dynamic scheduling algorithm for TTCAN and it's realization, *3rd international conference on advanced computer theory and engineering (ICACTE)*

Zhu, Z., Sui, J. ve Yang, L. (2010) Bin-packing algorithms for periodic task scheduling. *2010 WASE international conference on information engineering (ICIE)*

EK A. Network II.5 Programı Arayüzü



```
***** PROCESSING ELEMENTS
HARDWARE TYPE = PROCESSING
  NAME = TIME MASTER
    LOCATION =       30.451         3.806
    STYLE/COLOR =   1   3  1.00    0.   1  -9.58   3.07   1.00
    ICON = pedef.icn
    BASIC CYCLE TIME = 1. MIC
    INPUT CONTROLLER = YES
    QUEUE FLAG = NO
    INSTRUCTION REPERTOIRE =
      INSTRUCTION TYPE = MESSAGE
        NAME ; SEND REF MSG
          MESSAGE ; REF MSG
          LENGTH ;  96 BITS
          DESTINATION PROCESSOR ; *
          QUEUE FLAG ; NO
          RESUME FLAG ; NO
          INHIBIT MESSAGE TO SELF ; NO
      INSTRUCTION TYPE = PROCESSING
        NAME ; MSG WINDOW TIME 66
          TIME ;  81. CYCLES
        NAME ; MSG WINDOW TIME 76
          TIME ;  91. CYCLES
        NAME ; MSG WINDOW TIME 86
          TIME ;  101. CYCLES
        NAME ; MSG WINDOW TIME 96
          TIME ;  111. CYCLES
        NAME ; MSG WINDOW TIME 106
          TIME ;  121. CYCLES
        NAME ; MSG WINDOW TIME 126
          TIME ;  141. CYCLES
        NAME ; MSG WINDOW TIME 136
          TIME ;  151. CYCLES
        NAME ; NOP
          TIME ;  0. CYCLES
        NAME ; TIME PRCS ARB1
          TIME ;  359. CYCLES
        NAME ; TIME PRCS ARB2
          TIME ;  529. CYCLES
        NAME ; TIME PRCS ARB3
          TIME ;  684. CYCLES
        NAME ; TIME PRCS ARB4
          TIME ;  484. CYCLES
        NAME ; TIME PRCS ARB5
          TIME ;  554. CYCLES
        NAME ; TIME PRCS ARB6
          TIME ;  474. CYCLES
        NAME ; TIME PRCS ARB7
          TIME ;  599. CYCLES
        NAME ; TIME PRCS ARB8
          TIME ;  429. CYCLES
        NAME ; TIME PRCS ARB9
```

```
    TIME ;  504. CYCLES
  NAME ; TIME PRCS ARB10
    TIME ;  374. CYCLES
  NAME ; TIME PRCS ARB11
    TIME ;  279. CYCLES
  NAME ; TIME PRCS ARB12
    TIME ;  304. CYCLES
  NAME ; TIME PRCS ARB13
    TIME ;  514. CYCLES
  NAME ; TIME PRCS ARB14
    TIME ;  849. CYCLES
  NAME ; TIME PRCS ARB15
    TIME ;  804. CYCLES
  NAME ; TIME PRCS ARB16
    TIME ;  484. CYCLES
  NAME ; TX WINDOW TIME PRCS
    TIME ;  15. CYCLES
INSTRUCTION TYPE = SEMAPHORE
  NAME ; SET MC TIME SEM
    SEMAPHORE ; MC TIME SEM
    SET/RESET FLAG ; SET
    INCREMENT BY ;    1
  NAME ; RESET MC TIME SEM
    SEMAPHORE ; MC TIME SEM
    SET/RESET FLAG ; RESET
    DECREMENT BY ;    1
  NAME ; SET BC COUNT SEM
    SEMAPHORE ; BC COUNT SEM
    SET/RESET FLAG ; SET
    INCREMENT BY ;    1
  NAME ; RESET BC COUNT SEM
    SEMAPHORE ; BC COUNT SEM
    SET/RESET FLAG ; RESET
    DECREMENT BY ;    1
  NAME ; SET BC SUB COUNT SEM
    SEMAPHORE ; BC SUB COUNT SEM
    SET/RESET FLAG ; SET
    INCREMENT BY ;    1
  NAME ; RESET BC SUB COUNT SEM
    SEMAPHORE ; BC SUB COUNT SEM
    SET/RESET FLAG ; RESET
    DECREMENT BY ;    1
  NAME ; SET M1 WINDOW SEM
    SEMAPHORE ; M1 WINDOW SEM
    SET/RESET FLAG ; SET
    INCREMENT BY ;    1
  NAME ; SET M2 WINDOW SEM
    SEMAPHORE ; M2 WINDOW SEM
    SET/RESET FLAG ; SET
    INCREMENT BY ;    1
  NAME ; SET M3 WINDOW SEM
    SEMAPHORE ; M3 WINDOW SEM
```

```
  SET/RESET FLAG ; SET
  INCREMENT BY ;    1
NAME ; SET M4 WINDOW SEM
  SEMAPHORE ; M4 WINDOW SEM
  SET/RESET FLAG ; SET
  MODIFY COUNT ; NO
NAME ; SET M5 WINDOW SEM
  SEMAPHORE ; M5 WINDOW SEM
  SET/RESET FLAG ; SET
  INCREMENT BY ;    1
NAME ; SET M6 WINDOW SEM
  SEMAPHORE ; M6 WINDOW SEM
  SET/RESET FLAG ; SET
  INCREMENT BY ;    1
NAME ; SET M7 WINDOW SEM
  SEMAPHORE ; M7 WINDOW SEM
  SET/RESET FLAG ; SET
  INCREMENT BY ;    1
NAME ; SET M8 WINDOW SEM
  SEMAPHORE ; M8 WINDOW SEM
  SET/RESET FLAG ; SET
  INCREMENT BY ;    1
NAME ; SET M9 WINDOW SEM
  SEMAPHORE ; M9 WINDOW SEM
  SET/RESET FLAG ; SET
  INCREMENT BY ;    1
NAME ; SET M10 WINDOW SEM
  SEMAPHORE ; M10 WINDOW SEM
  SET/RESET FLAG ; SET
  INCREMENT BY ;    1
NAME ; SET M11 WINDOW SEM
  SEMAPHORE ; M11 WINDOW SEM
  SET/RESET FLAG ; SET
  INCREMENT BY ;    1
NAME ; SET M12 WINDOW SEM
  SEMAPHORE ; M12 WINDOW SEM
  SET/RESET FLAG ; SET
  INCREMENT BY ;    1
NAME ; RESET M1 WINDOW SEM
  SEMAPHORE ; M1 WINDOW SEM
  SET/RESET FLAG ; RESET
  DECREMENT BY ;    1
NAME ; RESET M2 WINDOW SEM
  SEMAPHORE ; M2 WINDOW SEM
  SET/RESET FLAG ; RESET
  DECREMENT BY ;    1
NAME ; RESET M3 WINDOW SEM
  SEMAPHORE ; M3 WINDOW SEM
  SET/RESET FLAG ; RESET
  DECREMENT BY ;    1
NAME ; RESET M4 WINDOW SEM
  SEMAPHORE ; M4 WINDOW SEM
  SET/RESET FLAG ; RESET
```

```
  DECREMENT BY ;    1
NAME ; RESET M5 WINDOW SEM
  SEMAPHORE ; M5 WINDOW SEM
  SET/RESET FLAG ; RESET
  DECREMENT BY ;    1
NAME ; RESET M6 WINDOW SEM
  SEMAPHORE ; M6 WINDOW SEM
  SET/RESET FLAG ; RESET
  DECREMENT BY ;    1
NAME ; RESET M7 WINDOW SEM
  SEMAPHORE ; M7 WINDOW SEM
  SET/RESET FLAG ; RESET
  DECREMENT BY ;    1
NAME ; RESET M8 WINDOW SEM
  SEMAPHORE ; M8 WINDOW SEM
  SET/RESET FLAG ; RESET
  DECREMENT BY ;    1
NAME ; RESET M9 WINDOW SEM
  SEMAPHORE ; M9 WINDOW SEM
  SET/RESET FLAG ; RESET
  DECREMENT BY ;    1
NAME ; RESET M10 WINDOW SEM
  SEMAPHORE ; M10 WINDOW SEM
  SET/RESET FLAG ; RESET
  DECREMENT BY ;    1
NAME ; RESET M11 WINDOW SEM
  SEMAPHORE ; M11 WINDOW SEM
  SET/RESET FLAG ; RESET
  DECREMENT BY ;    1
NAME ; RESET M12 WINDOW SEM
  SEMAPHORE ; M12 WINDOW SEM
  SET/RESET FLAG ; RESET
  DECREMENT BY ;    1
NAME ; SET BC TIME SEM
  SEMAPHORE ; BC TIME SEM
  SET/RESET FLAG ; SET
  INCREMENT BY ;    1
NAME ; RESET BC TIME SEM
  SEMAPHORE ; BC TIME SEM
  SET/RESET FLAG ; RESET
  DECREMENT BY ;    1
```

EK C. Simülasyon Sonuçları Örneği

TTCAN 1 Mbps

Semaphore Statistics

from 0. to 60. seconds
(All times reported in microseconds)

Semaphore Name	M1MSG SEM	M1 WINDOW SEM	M2MSG SEM
num Times Set	6000	6000	4286
num Times Reset	6000	6000	4286
Percent Time Set	1.510	1.510	.721
number Responses	6000	6000	4286
avg Response Time	151.000	151.000	101.000
max Response Time	151.000	151.000	101.000
min Response Time	151.000	151.000	101.000
std dev Time	0.	0.	0.
maximum Count	1	1	1
avg Count	.015	.015	.007
min Count	0	0	0

Semaphore Name	M2 WINDOW SEM	M3MSG SEM	M3 WINDOW SEM
num Times Set	30000	3000	3000
num Times Reset	30000	3000	3000
Percent Time Set	5.050	.505	.505
number Responses	30000	3000	3000
avg Response Time	101.000	101.000	101.000
max Response Time	101.000	101.000	101.000
min Response Time	101.000	101.000	101.000
std dev Time	0.	0.	0.
maximum Count	1	1	1
avg Count	.050	.005	.005
min Count	0	0	0

EK C.(devam)

CAN 1Mbps

Semaphore Statistics

from 0. to 3600. seconds
(All times reported in microseconds)

Semaphore Name	M1MSG SEM	M2MSG SEM	M3MSG SEM
num Times Set	360000	257143	180000
num Times Reset	360000	257143	180000
Percent Time Set	2.749	.795	.927
number Responses	360000	257143	180000
avg Response Time	274.851	111.313	185.399
max Response Time	646.000	510.000	424.000
min Response Time	136.000	86.000	86.000
std dev Time	119.643	72.419	106.867
maximum Count	1	1	1
avg Count	.027	.008	.009
min Count	0	0	0

ÖZGEÇMİŞ

Kişisel Bilgiler

Ad Soyad :Süleyman Halil TEMEL

Uyruk :T.C.

Doğum Yeri ve Tarihi:14/06/1978

Medeni Hali :Evli

Telefon :0 505 677 82 25

E-posta :haliltemel@gmail.com

Eğitim

Alınan Derece	Aldığı Kurum/Üniversite	Mezuniyet Yılı
Lise	Gönen Anadolu Öğretmen Lisesi	1995
Lisans	Gazi Üniversitesi T.E.F	2000
Yüksek Lisans	Muğla Sıtkı Koçman Üniversitesi	2014

İş Tecrübesi

Yıl	Yer	Pozisyon/görev
2000-2011	Yatağan	Öğretmen
2011-	Muğla	Öğretmen

Yabancı Dil(ler)

Dil (İngilizce, İspanyolca)	Başlangıç (İsp.)	Orta (İng.)	İleri
Yazma	X	X	
Konuşma	X	X	
Anlama	X	X	
Okuma	X	X	

Bilimsel Faaliyetler

1. 16. Akademik Bilişim Konferansı, Mersin, 5-7 Şubat 2014, Bildiri Sunumu