

KOCAELİ ÜNİVERSİTESİ * FEN BİLİMLERİ ENSTİTÜSÜ

**IEEE 802.11B KLAN KULLANARAK CAN SEGMENTLERİ
GENİŞLETEN ARABAĞLAŞIM BİRİMİ TASARIMI**

DOKTORA TEZİ

Elektronik Tek. Öğrt. Cüneyt BAYILMIŞ

**Anabilim Dalı: Elektrik Eğitimi
Danışman: Doç.Dr. İsmail ERTÜRK**

KOCAELİ, 2006

KOCAELİ ÜNİVERSİTESİ * FEN BİLİMLERİ ENSTİTÜSÜ

**IEEE 802.11B KLAN KULLANARAK CAN SEGMENTLERİ
GENİŞLETEN ARABAĞLAŞIM BİRİMİ TASARIMI**

DOKTORA TEZİ

Cüneyt BAYILMIŞ

Tezin Enstitüye Verildiği Tarih: 25 Ağustos 2006

Tezin Savunulduğu Tarih: 13 Eylül 2006

Tez Danışmanı

Doç.Dr. İsmail ERTÜRK



Üye

Prof.Dr. Hüseyin EKİZ



Üye

Doç.Dr. Yunus Emre ERDEMLİ



Üye

Yrd.Doç.Dr. Celal ÇEKEN



Üye

Yrd.Doç.Dr. İbrahim ÖZÇELİK



KOCAELİ, 2006

ÖNSÖZ VE TEŞEKKÜR

Endüstriyel kontrol uygulamalarında el ve dizüstü bilgisayarlar, taşınabilir cihazlar gibi hareketli sistemlerin kullanımının artması, kablosuz teknolojiler alanında yaşanan hızlı gelişmeler, kablolamayı azaltmayı amaçlayan sahayolu ağlarının kablosuz teknolojilerden yararlanmasının kaçınılmaz olduğunu göstermektedir. Bu tezde sunulan çalışmaların, bu alanda karşılaşılan problemlere hem teorik hem de pratik çözümler ve katkılar sağlaması hedeflenmiştir.

Tez çalışmasındaki prototip ve endüstriyel kontrol uygulamaları, Türkiye Bilimsel ve Teknolojik Araştırma Kurumu'nun (TÜBİTAK) EEEAG/105E059 numaralı proje desteği ile gerçekleştirilmiştir.

Tez çalışmam süresince değerli zamanlarını ayıran, bilgi ve deneyimlerini paylaşan, çalışmalarımı yönlendiren ve her zaman destek olan tez danışmanım sayın Doç.Dr. İsmail ERTÜRK'e (KO.Ü), tezin yazımından, modelleme ve donanımın gerçekleştirilmesine kadar her türlü ilgi ve desteği gösteren ve aynı zamanda tez izleme jürisi üyeleri sayın Yrd.Doç.Dr. Celal ÇEKEN (KO.Ü) ve sayın Yrd.Doç.Dr. İbrahim ÖZÇELİK'e (SA.Ü) teşekkürlerimi sunarım. Başta Necla BANDIRMALI, Ahmet KARACA, Ersoy KELEBEKLER, Adnan SONDAŞ ve Sedat ATMACA olmak üzere çalışmalarım boyunca desteklerini gördüğüm tüm dostlarıma da yardımları nedeniyle teşekkür ederim.

Beni bugünlerime getiren, her konuda destek veren ve yanımda olan çok değerli annem Sevim ve ağabeyim İsmail BAYILMIŞ'a yaptıkları her şey için çok teşekkür ediyorum.

Ve son olarak, en zor sağlık şartları altında bile okumam için çaba gösteren babam merhum Fevzi BAYILMIŞ'ın bugünleri görmesini dilerdim.

İÇİNDEKİLER

| | |
|--|------|
| ÖNSÖZ VE TEŞEKKÜR | i |
| İÇİNDEKİLER | ii |
| ŞEKİLLER DİZİNİ | v |
| TABLolar DİZİNİ | vii |
| SİMGELER | viii |
| ÖZET | xi |
| İNGİLİZCE ÖZET | xii |
| 1. GİRİŞ | 1 |
| 1.1. Literatürde Yapılan Çalışmaların Özetleri | 3 |
| 1.2. Tez Çalışmasının Amacı ve Önerilen Çözüm Yöntemi | 6 |
| 1.3. Tez Çalışmasının Katkıları | 8 |
| 1.4. Tez Organizasyonu | 9 |
| 2. ENDÜSTRİYEL İLETİŞİM SİSTEMLERİ VE DENETLEYİCİ ALAN AĞI ... | 11 |
| 2.1. Giriş | 11 |
| 2.2. Sahayolu Sistemleri | 12 |
| 2.2.1. Otomasyon hiyerarşisi | 13 |
| 2.2.2. Sahayolu sistemlerinin avantajları | 15 |
| 2.2.3. OSI referans modeli ve sahayolu sistemlerinin protokol mimarisi | 15 |
| 2.3. Denetleyici Alan Ağı (Controller Area Network, CAN) | 17 |
| 2.3.1. CAN genel karakteristikleri | 18 |
| 2.3.2. CAN protokol mimarisi | 18 |
| 2.3.3. CAN denetim mekanizması (arbitration mechanism) | 20 |
| 2.3.4. CAN mesaj çerçeve tipleri | 22 |
| 2.3.4.1. Veri çerçevesi | 22 |
| 2.3.4.2. Uzak çerçeve | 24 |
| 2.3.4.3. Hata çerçeveleri | 25 |
| 2.3.4.4. Aşırıyük çerçevesi | 26 |
| 2.3.5. CAN hata mekanizması | 26 |
| 2.3.6. CAN bit zamanlaması ve senkronizasyon | 30 |
| 2.3.7. CAN mesaj yanıt zaman analizi | 31 |
| 2.3.8. CAN entegrelerinin sınıflandırılması | 33 |
| 2.3.9. CAN denetleyici bağlantı şeması | 35 |
| 2.3.10. CAN uygulama alanları | 35 |
| 2.4. Sonuç | 36 |
| 3. KABLOSUZ YEREL ALAN AĞLARI | 38 |
| 3.1. Giriş | 38 |
| 3.2. Kablosuz Yerel Alan Ağlarının Avantajları | 39 |
| 3.3. Kablosuz Yerel Alan Ağlarının Kullanım Alanları | 40 |
| 3.4. Kablosuz LAN Ağ Topolojisi | 40 |
| 3.5. IEEE 802.11 Standardı | 41 |
| 3.5.1. IEEE 802.11 protokol mimarisi | 42 |
| 3.5.2. Çerçeve formatları | 43 |

| | |
|--|-----|
| 3.5.3. Çerçeveler arası boşluk (inter frame space, IFS) | 45 |
| 3.5.4. Ortam erişim mekanizması | 46 |
| 3.5.4.1. Dağıtık eşgüdüm fonksiyonu (DCF)..... | 47 |
| 3.5.4.1.1. Taşıyıcı duyarlı çoklu erişim (CSMA)..... | 47 |
| 3.5.4.1.2. Çarpışmadan kaçınma (CA) ve backoff prosedürü | 48 |
| 3.5.4.1.3. Hata sezme | 51 |
| 3.5.4.1.4. Sanal taşıyıcı algılama (virtual carrier sense) | 52 |
| 3.5.4.2. Nokta eşgüdüm fonksiyonu (PCF)..... | 54 |
| 3.5.5. IEEE 802.11 alt standartları | 55 |
| 3.5.5.1. IEEE 802.11a standardı..... | 56 |
| 3.5.5.2. IEEE 802.11b standardı | 56 |
| 3.5.5.3. IEEE 802.11g standardı | 57 |
| 3.6. Sonuç..... | 57 |
| 4. ENDÜSTRİYEL AĞLARDA ARABAĞLAŞIM VE CAN/IEEE 802.11b KLAN KABLOSUZ ARABAĞLAŞIM BİRİMİ TASARIMI..... | 59 |
| 4.1. Giriş..... | 59 |
| 4.2. Arabağlaşım Cihazları..... | 60 |
| 4.3. Endüstriyel Ağlarda Arabağlaşım | 63 |
| 4.3.1. Endüstriyel ağlarda kablolu omurga kullanımı | 64 |
| 4.3.2. Endüstriyel ağlarda kablosuz iletişim kullanımı | 67 |
| 4.4. CAN/IEEE 802.11b KLAN Kablosuz Arabağlaşım Biriminin Tasarımı | 72 |
| 4.4.1. KAB'ın görevleri | 74 |
| 4.4.2. KAB'ın kullanıldığı ağ mimarisi | 75 |
| 4.4.3. KAB'ın OSI katmanlı mimarisi | 76 |
| 4.4.4. KAB'da CAN mesajların kapsüllemesi | 77 |
| 4.4.5. CAN/IEEE 802.11b KAB modeli | 78 |
| 4.4.5.1. KAB'ı oluşturan birimlerin işlevleri | 79 |
| 4.4.5.2. KAB'ın çalışma prosedürü..... | 84 |
| 4.5. Sonuç..... | 88 |
| 5. CAN/IEEE 802.11B KLAN KABLOSUZ ARABAĞLAŞIM BİRİMİNİN MODELLENMESİ VE BAŞARIM ANALİZİ | 90 |
| 5.1. Giriş..... | 90 |
| 5.2. Basitleştirilmiş Analitik Model..... | 92 |
| 5.3. Modelleme, Benzetim ve OPNET Yazılımı..... | 93 |
| 5.3.1. OPNET benzetim yazılımı | 94 |
| 5.4. CAN/IEEE 802.11b KLAN Kablosuz Arabağlaşım Biriminin OPNET Modeli | 98 |
| 5.4.1. CAN modeli | 98 |
| 5.4.1.1. CAN proses modeli ve algoritması | 99 |
| 5.4.2. CAN/IEEE 802.11b KAB modeli | 102 |
| 5.4.2.1. KAB CAN proses modeli ve algoritması..... | 103 |
| 5.4.2.2. KAB KLAN proses modeli ve algoritması | 107 |
| 5.5. CAN/IEEE 802.11b KLAN KAB'ın Kullanıldığı Endüstriyel Ağ Uygulaması ve Başarım Analizi | 110 |
| 5.5.1. Benzetim sonuçları ve başarım analizi..... | 113 |
| 5.6. Sonuç..... | 119 |
| 6. CAN/IEEE 802.11B KLAN KABLOSUZ ARABAĞLAŞIM BİRİMİ PROTOTİPİNİN GERÇEKLEŞTİRİLMESİ ve UYGULAMALARI..... | 120 |
| 6.1. Giriş..... | 120 |
| 6.2. CAN/IEEE 802.11b KLAN KAB'ın Prototipi | 121 |

| | |
|--|-----|
| 6.2.1. 8051 tabanlı CAN denetleyici geliştirme kartı..... | 121 |
| 6.2.2. IEEE 802.11b KLAN birimi | 122 |
| 6.2.3. CAN/IEEE 802.11b KLAN KAB | 123 |
| 6.3. Tasarlanan KAB Prototipinin Kullanıldığı Endüstriyel Ağ Uygulaması..... | 124 |
| 6.4. CAN Kullanan Dağıtık Kontrol Sisteminin Tasarlanan KAB Aracılığıyla Kablosuz Olarak Sesli Kontrolü | 129 |
| 6.5. Sonuç..... | 131 |
| 7. SONUÇLAR VE ÖNERİLER | 133 |
| KAYNAKLAR | 138 |
| EKLER..... | 143 |
| KİŞİSEL YAYINLAR VE ESERLER | 158 |
| ÖZGEÇMİŞ | 162 |

ŞEKİLLER DİZİNİ

| | |
|---|----|
| Şekil 1.1: Sahayolu sistemlerinde kablosuz iletişim sistemlerinin kullanıldığı örnek bir ağ | 3 |
| Şekil 2.1: Bilgisayar kontrollü dağıtık sistemler | 11 |
| Şekil 2.2: Bir sahayolu sisteminin genel görünüşü | 12 |
| Şekil 2.3: BBİ mimarisi | 14 |
| Şekil 2.4: OSI referans ve alan ağı modeli | 16 |
| Şekil 2.5: CAN protokol mimarisi ve protokollerin görevleri | 19 |
| Şekil 2.6: Bit düzeyinde denetim mekanizmasının çalışması | 21 |
| Şekil 2.7: Standart ve Genişletilmiş CAN veri çerçeveleri | 22 |
| Şekil 2.8: CAN hata çerçeve biçimleri | 25 |
| Şekil 2.9: CAN aşırıyük çerçevesi | 26 |
| Şekil 2.10: Hata durumları | 29 |
| Şekil 2.11: CAN'de bit zamanlaması | 30 |
| Şekil 2.12: CAN denetleyicilerinin sınıflandırılması | 33 |
| Şekil 2.13: CAN genel bağlantı şekli | 35 |
| Şekil 3.1: Kablosuz ağlar ve uygulamadaki yerleri | 38 |
| Şekil 3.2: KLAN topolojileri: a) eşe-eş ağ b) erişim noktalı ağ | 41 |
| Şekil 3.3: IEEE 802.11 temel referans modeli (Bing, 2000) | 42 |
| Şekil 3.4: MAC (MPDU) genel çerçeve biçimi (ANSI/IEEE Std 802.11 1999) | 43 |
| Şekil 3.5: RTS çerçeve biçimi (ANSI/IEEE Std 802.11 1999) | 44 |
| Şekil 3.6: CTS çerçeve biçimi (ANSI/IEEE Std 802.11 1999) | 44 |
| Şekil 3.7: IEEE 802.11b DSSS PLCP çerçeve biçimi (ANSI/IEEE Std 802.11 1999) | 45 |
| Şekil 3.8: Çerçeveler arası boşluk tanımlamaları (Bing 2000) | 45 |
| Şekil 3.9: IEEE 802.11 ortam erişim mekanizmasının genel çalışması (ANSI/IEEE Std 802.11, 1999) | 47 |
| Şekil 3.10: CW'nin üssel artımına bir örnek (ANSI/IEEE Std 802.11, 1999) | 49 |
| Şekil 3.11: CSMA/CA kullanarak çoklu paket gönderimi | 50 |
| Şekil 3.12: DCF ortam erişim mekanizmasının genel çalışma akış diyagramı | 51 |
| Şekil 3.13: Sanal algılama kullanılarak paket iletimi | 53 |
| Şekil 3.14: Tek bir RTS/CTS kullanılarak ard arda paket gönderimi | 54 |
| Şekil 3.15: PCF erişimi | 55 |
| Şekil 3.16: PCF erişim mekanizmasında çerçeve transferi | 55 |
| Şekil 4.1: Arabağlaşım cihazları ile birbirine bağlanmış bir ağ örneği | 59 |
| Şekil 4.2: Arabağlaşım cihazlarının OSI modelindeki yeri | 61 |
| Şekil 4.3: BBİ mimarisine göre endüstriyel ağlarda arabağlaşım | 64 |
| Şekil 4.4: Endüstriyel ağların arabağlaşımında kablolu omurga kullanımı | 64 |
| Şekil 4.5: Endüstriyel ağlarda kablosuz iletişimin kullanıldığı bir ağ | 68 |
| Şekil 4.6: Saydam-çevrimli yerel köprü | 74 |
| Şekil 4.7: KAB'ın kullanıldığı örnek bir ağ | 76 |
| Şekil 4.8: CAN/IEEE 802.11b KAB OSI katmanlı mimarisi | 77 |

| | |
|--|-----|
| Şekil 4.9: IEEE 802.11b DSSS çerçevesi içerisine CAN 2.0A çerçevelerinin kapsüllemesi | 77 |
| Şekil 4.10: CAN/IEEE 802.11b KLAN KAB işlevsel şeması..... | 78 |
| Şekil 4.11: CAB birimi ve tamponlara bağlantısı | 79 |
| Şekil 4.12: CÖFİB biriminin genel çalışması | 80 |
| Şekil 4.13: CÖFİB biriminin bakış tablosu ve tamponlarla bağlantısı | 81 |
| Şekil 4.14: CAB biriminin işlevsel yapısı ve tamponlar ile bağlantısı | 82 |
| Şekil 4.15: KLÖFİB biriminin genel çalışması..... | 83 |
| Şekil 4.16: KLÖFİB biriminin bakış tablosu ve tamponlara bağlantısı..... | 83 |
| Şekil 4.17: CAN–KLAN / KLAN–CAN tarafına veri transfer süreci akış diyagramı | 86 |
| Şekil 4.18: KLAN–CAN / CAN–KLAN tarafına veri transfer süreci akış diyagramı | 87 |
| Şekil 5.1: İletişim sistemleri geliştirmede kullanılan yöntemler..... | 90 |
| Şekil 5.2: CAN/IEEE 802.11b/CAN ağdaki gecikmeler için örnek bir model..... | 92 |
| Şekil 5.3: OPNET sistem modellerinin tasarımında kullanılan editörlerin hiyerarşik ilişkisi | 97 |
| Şekil 5.4: KAB’ın kullanıldığı örnek bir endüstriyel ağ..... | 98 |
| Şekil 5.5: CAN düğüm (node) modeli | 99 |
| Şekil 5.6: CAN_MAC modülü proses modeli | 100 |
| Şekil 5.7: CAN_MAC modülü proses modeli algoritması | 101 |
| Şekil 5.8: KAB düğüm (node) modeli | 103 |
| Şekil 5.9: KAB CAN_MAC modülü proses modeli..... | 105 |
| Şekil 5.10: KAB CAN_MAC modülü proses modeli algoritması..... | 106 |
| Şekil 5.11: KAB KLAN_MAC modülü proses modeli | 108 |
| Şekil 5.12: KAB KLAN_MAC modülü proses modeli algoritması | 109 |
| Şekil 5.13: KAB’ın kullanıldığı endüstriyel ağ uygulaması | 111 |
| Şekil 5.14: CAN veriyolu A’nın kullanım yüzdesi..... | 114 |
| Şekil 5.15: P ₁₃₋₁₄ yerel CAN mesajının uçtan uca gecikmesi..... | 114 |
| Şekil 5.16: En yüksek öncelikli P ₁ uzak CAN mesajının uçtan uca gecikmesi | 115 |
| Şekil 5.17: En düşük öncelikli P ₁₂ uzak CAN mesajının uçtan uca gecikmesi..... | 116 |
| Şekil 5.18: KAB _A CAN–KLAN işlem zamanı | 117 |
| Şekil 5.19: KAB _A KLAN-CAN işlem zamanı..... | 117 |
| Şekil 5.20: KAB’ın işlem zamanının P ₁ uzak CAN mesajın uçtan uca toplam gecikmesi içerisindeki oranı | 118 |
| Şekil 5.21: Farklı CAN veriyolu iletim hızlarında paket kayıp oranı | 118 |
| Şekil 6.1: 8051 tabanlı CAN denetleyici geliştirme kartı ve blok diyagramı | 122 |
| Şekil 6.2: IEEE 802.11b KLAN birimi ve blok diyagramı..... | 123 |
| Şekil 6.3: CAN/IEEE 802.11b KLAN KAB ve blok diyagramı..... | 124 |
| Şekil 6.4: KAB’ın kullanıldığı endüstriyel ağ uygulamasının blok diyagramı..... | 125 |
| Şekil 6.5: KAB’ın kullanıldığı endüstriyel ağ uygulaması | 125 |
| Şekil 6.6: CAN geliştirme kartının çalışma akış diyagramı..... | 127 |
| Şekil 6.7: KAB çalışma akış diyagramı | 128 |
| Şekil 6.8: CAN ile kontrol edilen model arabanın KAB vasıtasıyla kablosuz olarak ses ile kontrolü uygulaması blok diyagramı | 130 |
| Şekil 6.9: CAN ile kontrol edilen model arabanın KAB vasıtasıyla kablosuz olarak ses ile kontrolü uygulaması | 130 |
| Şekil 6.10: CAN ile kontrol edilen model arabanın KAB vasıtasıyla kablosuz olarak ses ile kontrolü uygulamasının çalışma akış diyagramı..... | 131 |

TABLolar DİZİNİ

| | |
|---|-----|
| Tablo 2.1: Sahayolu sistemlerinin genel karakteristikleri..... | 13 |
| Tablo 2.2: CAN genel karakteristikleri..... | 18 |
| Tablo 3.1: IEEE 802.11 standart ailesi..... | 57 |
| Tablo 5.1: İletişim sistemlerini modellemek için yaygın olarak kullanılan benzetim yazılımları..... | 94 |
| Tablo 5.2: OPNET kullanıcı tipleri ve kullanım amaçları..... | 95 |
| Tablo 5.3: CAN segmentlerin Yerel (Y) ve Uzak (U) mesaj bilgileri..... | 112 |
| Tablo 6.1: Endüstriyel ağ uygulamasında kullanılan CAN mesaj bilgileri | 125 |
| Tablo 6.2: Model arabanın kontrolü için kullanılan CAN mesajların bilgileri..... | 129 |

SİMGELER

| | |
|--------|--|
| C | : Mesaj iletim süresi (s) |
| D | : Mesajın varma sınır değeri (s) |
| d | : Mesaj yük büyüklüğü (bayt) |
| P | : CAN düğümün öncelik değeri |
| R | : En kötü durum gecikme süresi (s) |
| S | : Tek bir tanıtıcı değere sahip mesaj akışı |
| T | : Mesajın üretim aralık zamanı (s) |
| t | : Mesajın kuyruğa atılmasından veriyolu erişimini kazanmasına kadar geçen süre (s) |
| τ | : Veriyolunun bir bit iletim zamanı (s) |

Alt indisler

| | |
|------|--|
| B1 | : Kaynak düğümün bulunduğu veriyolundaki yayılım gecikmesi |
| B2 | : Hedef düğümün bulunduğu veriyolundaki yayılım gecikmesi |
| bit | : Bir bit |
| C1 | : Kaynak CAN düğümünde oluşan gecikme |
| C2 | : Hedef CAN düğümünde oluşan gecikme |
| KAB1 | : Kaynak KAB'da meydana gelen gecikme |
| KAB2 | : Hedef KAB'da meydana gelen gecikme |
| Ky | : Kablosuz iletim ortamından kaynaklanan yayılım gecikmesi |
| m | : İlgili mesaj |
| r | : CAN düğümüne hedeflenen mesajın önceliği |
| s | : CAN düğüm tarafından üretilen mesajın önceliği |
| T | : Toplam gecikme |

Kısaltmalar

| | |
|-----|---|
| ACK | : ACKnowledgement |
| AP | : Access Point |
| ATM | : Asynchronous Transfer Mode (Asenkron Transfer Modu) |
| BBİ | : Bilgisayarla Bütünleşik İmalat |
| BRP | : Baud Rate Prescaler |
| BSS | : Basic Service Set |
| BT | : Bakış Tablosu |
| BTL | : Bit Timing Logic |
| CAB | : CAN Arayüz Birimi |
| CAL | : CAN Application Layer |
| CAN | : Controller Area Network (Denetleyici Alan Ağı) |
| CCK | : Complementary Code Keying |
| CD | : Collision Detection |
| CDA | : Common Digital Architecture |

| | |
|-------------|---|
| CERN | : European Organization for Nuclear Research |
| CFP | : Contention Free Period |
| CIM | : Computer Integrated Manufacturing |
| CiA | : CAN in Automation |
| CNC | : Computer Numeric Control |
| CÖFİB | : CAN Öğrenme, Filtreleme ve İletme Birimi |
| CRC | : Cyclic Redundancy Check |
| CSMA/CA | : Carrier Sense Multiple Access with Collision Avoidance |
| CSMA/CD+AMP | : Carrier Sense Multiple Access with Collision Detection with Arbitration on Message Priority |
| CSMA/CD | : Carrier Sense Multiple Access with Collision Detection |
| CTS | : Clear To Send |
| CW | : Collision Window |
| DBPSK | : Differential Binary Phase Shift Keying |
| DCF | : Distributed Coordination Function |
| DIFS | : Distributed Coordination Function IFS |
| DLC | : Data Length Code |
| DLL | : Data Link Layer |
| DQPSK | : Differential Quadrature Phase Shift Keying |
| DSP | : Digital Signal Processing |
| DSSS | : Direct Sequence Spread Spectrum |
| EIFS | : Extended IFS |
| EOF | : End of Frame |
| FDDI | : Fiber Distributed Data Interface |
| FHSS | : Frequency Hopping Spread Spectrum |
| FIFO | : First Input First Output |
| FPGA | : Field Programmable Gate Array |
| GPRS | : General Packet Radio Service |
| GSM | : Global System for Mobile Communications |
| HD | : Hedef Düğüm |
| HiperLAN | : High Performance Radio Local Area Network |
| IDE | : Identifier Extention |
| IEC | : International Electrotechnical Commision |
| IEEE | : Institute of Electrical and Electronics Engineers |
| IFS | : Inter Frame Space (Çerçeveler Arası Boşluk) |
| ISA | : Instrumentation Society of America |
| ISM | : Industries, Scientific, Medical |
| ISO | : International Standards Organization |
| KAB | : Kablosuz Arabağlaşım Birimi |
| KATM | : Kablosuz ATM |
| KD | : Kaynak Düğüm |
| KLAB | : KLAN Arayüz Birimi |
| KLAN | : Kablosuz LAN |
| KLÖFİB | : KLAN Öğrenme, Filtreleme ve İletme Birimi |
| LAN | : Local Area Network (Yerel Alan Ağı) |
| LLC | : Logical Link Control |
| LMDS | : Local Multipoint Distribution System |
| MAC | : Medium Access Control (Ortam Erişim Kontrolü) |
| MB | : Mesaj Boyutu |

| | |
|-------------|--|
| MPDU | : MAC Protocol Data Unit |
| MPLS | : Multiprotocol Label Switching |
| MT | : Mesaj Tipi |
| NAV | : Network Allocation Vector |
| OFDM | : Orthogonal Frequency Division Multiplexing |
| OPNET | : OPTimized Network Engineering Tool |
| OSI | : Open Systems Interconnections |
| PCF | : Point Coordination Function |
| PDF | : Probability Density Function (Olasılık Sıklık Fonksiyonu) |
| PIFS | : Point Coordination Function IFS |
| PLC | : Programmable Logic Controller (Programlanabilir Lojik Denetleyici) |
| PLCP | : Physical Layer Convergence Procedure |
| Profibus | : Process Field Bus |
| Profibus-DP | : Process Field Bus Decentralized Periphery |
| PMD | : Physical Medium Dependent |
| P-NET | : Process Network |
| REC | : Receive Error Counter |
| RF | : Radio Frequency |
| RFMAC | : Remote Frame Medium Access Control |
| RTR | : Remote Transmission Request |
| RTS | : Request To Send |
| SAE | : Society of Automotive Engineers |
| SDS | : Smart Distributed Systems |
| SFR | : Special Function Register |
| SIFS | : Short Inter Frame Space |
| SJW | : Synchronization Jump Width |
| SLIO | : Serial Linked Input Output |
| SOF | : Start of Frame (Çerçeve Başlangıcı) |
| SRAM | : Static Random Access Memory |
| SRR | : Substitute Remote Request |
| STS | : Standardized Teleoperation System |
| TEC | : Transmit Error Counter |
| TCP/IP | : Transmission Control Protocol/Internet Protocol |
| TCP | : Transmission Control Protocol |
| UDP | : User Datagram Protocol |
| UHF | : Ultra High Frequency |
| U-NII | : Unlicensed National Information Infrastructure |
| WAN | : Wide Area Network (Geniş Alan Ağı) |
| WÇA | : WLAN Çerçeve Adresi |
| WECA | : Wireless Ethernet Company Alliance |
| WLAN | : Wireless Local Area Network |
| WMAC | : Wireless Medium Access |
| WorldFIP | : Factory Instrumentation Protocol |

IEEE 802.11B KLAN KULLANARAK CAN SEGMENTLERİ GENİŞLETEN ARABAĞLAŞIM BİRİMİ TASARIMI

Cüneyt BAYILMIŞ

Anahtar Kelimeler: Denetleyici Alan Ağı (CAN), Kablosuz İletişim, IEEE 802.11b KLAN, Kablosuz Arabağlaşım

Özet: Fabrika otomasyonu ve endüstriyel üretim ortamının düşük seviyelerindeki algılayıcı, eyleyici, PLC, robot, mikrodenetleyici, vb. dağıtık kontrol cihazların arabağlantısı sahayolu ağları (FieldBus) ile sağlanır. Sahayolu ağları, kablolanmanın azalmasından kaynaklanan kurulum ve bakım kolaylığı, düşük maliyet, güvenilirlik, yeni denetleyici ekleme/çıkarma esnekliği, yüksek performans, kritik zamanlı haberleşme gibi birçok avantaj sunar.

Sahayolu ağlarının, dağıtık alanın büyüklüğü ve veriyolunun uzunluğundan kaynaklanan sınırlamaları sebebiyle birbirleri ile haberleşmesi ve hücre/yönetim seviyesindeki diğer yerel alan ağları ile haberleşmesi olmak üzere iki temel problemi bulunmaktadır. Bu problemleri gidermek ve sahayolu ağlarının genişletilmesini sağlamak için, endüstriyel kontrol ortamı birkaç sahayolu ağından oluşacak şekilde bölünerek her bir sahayolu ağı arabağlaşım cihazı olarak adlandırılan elemanlar yardımıyla birbirlerine ve diğer yerel alan ağlarına bağlanırlar.

Kablosuz iletişim teknolojilerinin ofis ortamlarındaki başarısı ve yaygın kullanımı, endüstriyel kontrol uygulamalarında el ve dizüstü bilgisayarlar, taşınabilir cihazlar, vb. hareketli sistemlerin kullanımının artması, kablolanmayı azaltmayı amaçlayan sahayolu ağlarının da kablosuz teknolojilerden yararlanması gerekliliğini ortaya koymaktadır.

Bu tez çalışmasında, en başarılı sahayolu sistemlerinden biri olan CAN temelli dağıtık kontrol sistemlerinin IEEE 802.11b KLAN vasıtasıyla kablosuz ortam üzerinden birbirleri ile ve diğer yerel alan ağları ile haberleşmesini sağlayacak CAN/IEEE 802.11b Kablosuz Arabağlaşım Birimi (KAB) tasarımı gerçekleştirilmiştir. KAB, OPNET Modeler yazılımı kullanılarak modellenmiş ve KAB'ın kullanıldığı örnek bir endüstriyel ağ uygulaması benzetime tabi tutularak SAE Benchmark'a göre başarımlar değerlendirilmiştir. Elde edilen benzetim sonuçları 100 Kbit/s CAN veriyolu iletim hızında uygulamadaki tüm mesajların SAE Benchmark'da belirtilen gerçek zamanlı iletişim gereksinimini karşıladığını göstermektedir. Ayrıca tez çalışmasında önerilen KAB'ın temel bir prototipi gerçekleştirilerek farklı endüstriyel kontrol uygulamalarında kullanılabilirliği gösterilmiştir.

THE DESIGN AND IMPLEMENTATION OF AN INTERWORKING UNIT EXTENDING CAN SEGMENTS USING IEEE 802.11B WLAN

Cüneyt BAYILMIŞ

Keywords: Controller Area Network (CAN), Wireless Networking, IEEE 802.11b WLAN, Wireless Interworking

Abstract: FieldBus networks provide interconnection of distributed control devices such as sensors, actuators, PLCs, robots, microcontrollers at low levels of factory automation and industrial manufacturing environments. They offer several advantages resulted from their less need for cable installation, easy installation and maintenance, low cost, high reliability, new controller join/remove flexibility, high performance, appropriateness for critical real-time communication.

Due to their applied size of distributed area and length of bus, FieldBus networks introduce two main problems that are needs for communication with independent FieldBus segments and with other types of LANs at cell/management level. In order to overcome these problems and to extend FieldBus length, industrial control environment is divided into several FieldBus networks and each is connected to others and is also connected with other LANs using suitable devices called as Interworking Units.

Widespread use and success of wireless communication technologies in office environment and increasing use of mobile systems such as handheld computers, laptops, transportation devices in industrial control applications have introduced the necessity of FieldBus networks benefiting from these wireless technologies.

In this thesis research, a CAN/IEEE 802.11b Wireless Interworking Unit (WIU) has been designed and implemented. The WIU provides communication with other LANs and each others over wireless medium through IEEE 802.11b WLAN of CAN based distributed control systems. The designed WIU has been modeled using OPNET Modeler software. Performance evaluation of an example industrial network application including the WIUs is presented with comparisons to the SAE Benchmark. The obtained simulation results have shown the successful real-time transferring of all CAN messages in the application which uses CAN bus rates of 100 Kbit/s according to SAE Benchmark. In addition, a prototype of the WIU has been implemented and its application in different industrial control systems has been realized.

1. GİRİŞ

Bir fabrika ya da endüstriyel otomasyon ortamı düşünülduğünde; veri iletim hızı, veri paketi boyutu ve kullanılan protokol açısından birçok farklılığa sahip, algılayıcı (sensör), eyleyici (actuator), programlanabilir mantıksal denetleyici (Programmable Logic Controller, PLC) ve mikrodenetleyicili cihazların kullanıldığı üretim sistemlerinden bürolardaki PC'lere kadar birçok cihaz akla gelmektedir. Endüstriyel üretim ortamı içerisinde tüm bu cihazların birbirleri ile doğrudan ya da dolaylı olarak iletişimi Bilgisayarla Bütünleşik İmalat (BBİ) mimarisi (Computer Integrated Manufacturing, CIM) ile tanımlanır.

Sahayolu sistemleri, BBİ mimarisinin düşük seviyelerindeki algılayıcı, eyleyici, PLC, mikrodenetleyici gibi kontrol cihazlarının bağlantısını sağlar. Bu sistemler, başta kablolanmanın azalmasından kaynaklanan kolay kurulum ve bakım olmak üzere, maliyet kazancı, güvenilirlik, yeni denetleyici ekleme/çıkarma esnekliği gibi avantajlar sunmaktadır (Özçelik, 2002, Thomesse, 2005, Tovar, 1999).

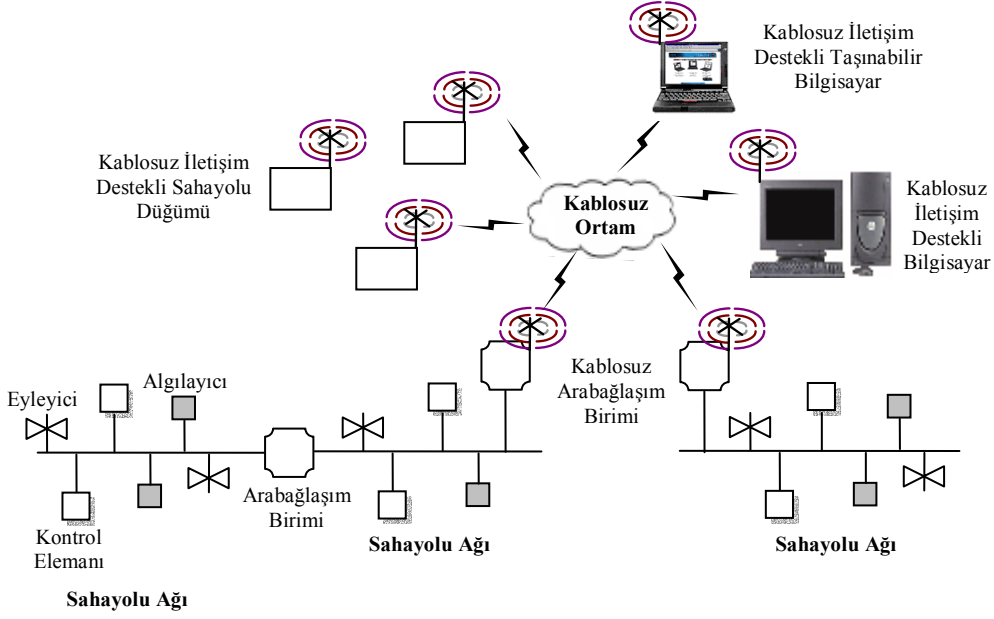
Sahayolu sistemleri fabrika otomasyonu, süreç (process) kontrol, güvenlik sistemleri, robotik uygulamalar, otomotiv sistemleri, bina otomasyonu gibi birçok endüstriyel kontrol uygulamalarında yaygın olarak kullanılmaktadır. Günümüzde farklı organizasyonlar tarafından ve farklı amaçlar için geliştirilmiş CAN (Controller Area Network), PROFIBUS (Process Field Bus), WorldFIP (Factory Instrumentation Protocol), InterBus ve P-NET (Process Network), vb. birçok sahayolu sistemi bulunmaktadır.

Kablosuz iletişim standartlarında yaşanan hızlı gelişmeler, sahayolu sistemlerinin kullanıldığı uygulamalarda hareketli (mobile) cihazların/modüllerin kullanımının artması, endüstriyel üretim ortamlarında kablolanmayı azaltmayı amaçlayan sahayolu ağlarının kablosuz teknolojilerle entegrasyonunu kaçınılmaz hale getirmiştir.

Sahayolu sistemlerinin zamanlama, güvenilirlik, performans gibi uyulması zorunlu kısıtlamaları bulunmaktadır. Muhtemel bir entegrasyon ortamının önemli bir parçası olarak kablosuz sistemlerin de bu ihtiyaçları karşılaması beklenmektedir. Ancak yüksek bit hata oranı, sınırlı bant genişliği, çok yönlü sönümlenme (multipath fading), hava koşulları ve çevresel özelliklerden kaynaklanan birçok sınırlamalara sahip kablosuz iletim ortamı göz önüne alındığında sahayolu sistemlerinin tüm gereksinimlerinin karşılanması oldukça zordur. Bu kısıtlamalarına rağmen, kablosuz iletişim teknolojilerinde güvenilirliğin geliştirilmesi ve yüksek veri iletim hızlarının sağlanması gibi performans artırıcı gelişmeler yakın zamanda kablosuz sistemlerin, sahayolu ağlarının arabağlaşımında önemli bir çözüm olacağını göstermektedir (De Pellegrini ve diğ., 2006, Thomesse, 2005).

Kablosuz ağlar, kurulum kolaylığı, hareketlilik, mevcut ağ yapılarını genişletme gibi avantajları sayesinde geleneksel kablolu sahayolu sistemlerinin hareketli sistemler ile bağlantısının sağlanması ve kablolanın pahalı ya da zor/tehlikeli olduğu endüstriyel üretim ortamlarından kaynaklanan sınırlamalarını ortadan kaldıracak potansiyel bir çözüm yaklaşımı sunmaktadır. Özellikle kablosuz sistemlerin doğrudan eşe-eş (Ad-hoc) ağ oluşturma yeteneği endüstriyel üretim ortamlarında büyük bir esneklik sağlar.

Şekil 1.1'de sahayolu sistemlerinin kablosuz iletişim teknolojileri üzerinden haberleşmesini gösteren melez (hybrid) kablolu/kablosuz endüstriyel ağ yapısı görülmektedir. Şekilden anlaşılacağı üzere saha seviyesindeki cihazlar kablosuz olarak iki farklı şekilde haberleşebilir. Birincisinde; saha seviyesindeki tek bir cihaz doğrudan kablosuz iletişim desteği ile kablosuz ortam üzerinden endüstriyel ağa bağlanır. Tüm düğümlerin kablosuz iletişim yeteneğine sahip olması bir endüstriyel ağ içerisindeki düğüm sayısı düşünüldüğünde önemli bir maliyet yükü getirmektedir. Bundan dolayı bu yöntem genellikle tercih edilmemektedir. İkincisinde ise bir sahayolu ağını oluşturan cihazlar, bu ağa bağlı kablosuz iletişim desteğine sahip Kablosuz Arabağlaşım Birimi (KAB) vasıtasıyla kablosuz olarak haberleşir. Bu yöntem maliyet kazancının yanı sıra mevcut sahayolu sistemlerinin kablosuz ortam üzerinden kolaylıkla genişletilmesini de sağlamaktadır.



Şekil 1.1: Sahayolu sistemlerinde kablosuz iletişim sistemlerinin kullanıldığı örnek bir ağ

Literatürde farklı sahayolu sistemlerinin farklı kablosuz iletişim teknolojilerini kullanarak kablosuz ortam üzerinden haberleşmesi üzerine birçok çalışma/araştırma bulunmaktadır. Aşağıdaki alt bölümde bu çalışmalardan birkaçı kısaca özetlenmektedir. Literatür özetinde verilen çalışmalara Bölüm 4’de detaylı bir şekilde değinilmektedir.

1.1. Literatürde Yapılan Çalışmaların Özetleri

Kutlu ve diğ. (1996a) ve (1996b), çalışmalarında dağıtık ve merkezileştirilmiş CAN sahayolu sistemlerinin kablosuz haberleşmesi için önerilen WMAC (Wireless Medium Access) ve RMAC (Remote Frame Medium Access Control) protokollerinin SAE (Society of Automotive Engineers) Benchmark kullanılarak gerçekleştirilen başarımlarını sunmuştur.

Cavalieri ve Pano (1998), BBİ mimarisine göre bir endüstriyel üretim ortamında IEC/ISA sahayolu ağının IEEE 802.11 kablosuz LAN (KLAN) üzerinden diğer IEC/ISA sahayolu ağları ve yönetim/hücre seviyesindeki cihazlarla haberleşmesini sağlayacak bir yapı sunmuştur. IEC/ISA sahayolu ağının kritik zamanlı iletişim

gereksinimini karřılamak için IEEE 802.11 KLAN ortam erişim mekanizmasında (Medium Access Control, MAC) bir takım deęişiklikler önermektedir.

Alves (2003) çalışmasında PROFIBUS sahayolu aęının IEEE 802.11b KLAN üzerinden haberleşmesini saęlayacak bir yapı sunmuştur. Melez kablolu/kablosuz haberleşme mimarisi üzerinde gerçek zamanlı iletişimin garanti edilmesi ve desteklenmesi için deęişik yaklaşımlar ve uygun mekanizmalar sunmuştur. Ayrıca kablosuz iletişim standardı olarak IEEE 802.11b KLAN'ını seçme sebeplerini anlatmıştır.

Willig (2003a) çalışmasında kablolu ve kablosuz PROFIBUS düğümlerin sanal bir halka oluşturacak şekilde haberleşmesini saęlayan bir yapı sunmuştur. Amacı mevcut kablolu sahayolu sistemleri ile kablosuz segmentleri birleştirmektir. Willig (2003b) çalışmasında ise gerçek zamanlı iletişim gereksinimlerine göre kablosuz PROFIBUS'ın ortam erişim ve veri iletim katmanı için önerilen iki farklı yaklaşımı karşılaştırmıştır. Kablosuz PROFIBUS'lar için polling-based protokollerin uygun olabileceğini ortaya koymuştur.

Ertürk (2003) ve (2005) çalışmalarında sabit uzak CAN segmentlerin Kablosuz ATM (KATM) üzerinden haberleşmesini saęlayan kapsülleme tekniğine dayalı bir yaklaşım önermiştir. OPNET yazılımı kullanılarak gerçekleştirilen benzetimlerin başarımlarını SAE Benchmark'a göre gerçekleştirmiş ve CAN uygulama trafıklarının KATM ile taşınabileceğinin yanı sıra ihtiyaç duyulan servis kalitesinin de saęlanabileceğini göstermiştir.

Koulamas ve dię. (2004), melez kablolu/kablosuz PROFIBUS sahayolu sistemlerinde kullanılacak doğrudan geçişli (Cut-Through) bir iletim elemanı tanımlamış ve mimarisi ile çalışmasını vermiştir. Doğrudan geçişli elemanın iletim gecikmesinin ayrıntılı analitik modelini sunmuştur. Kablosuz iletişim teknolojisi olarak IEEE 802.11 KLAN'ı kullanmıştır.

Miorandi ve Vitturi (2004), Ethernet ve Bluetooth temelli PROFIBUS–DP sahayolu sisteminin melez kablolu/kablosuz aęlara uygulanabilirliğinin analizini sunmaktadır.

Teorik çalışılan bu araştırmada belirtilen sistemin katmanlı protokol yapısı verilmektedir. Aynı yazarlar (2005) diğer bir çalışmalarında ise Bluetooth kablosuz teknolojisine dayalı olarak PROFIBUS–DP sahayolu sistemlerinin kablosuz iletişimi için bir çözüm sunmaktadırlar. Bu çözüm kapsamında, BT–PROFI olarak adlandırdıkları yeni bir protokol önermişlerdir.

Willig ve diğ. (2005), endüstriyel ağların arabağlaşımı için kullanılan kablosuz teknolojileri ve uygulamaları sunmaktadır. Aynı zamanda sahayolu sistemlerinin güvenilirlik ve kritik zamanlı haberleşme gereksinimlerinin kablosuz teknolojiler ile sağlanabileceğini tartışmıştır.

De Pellegrini ve diğ. (2006), çalışmalarında saha seviyesinde kablosuz ağların kullanımını ele almakta ve haberleşme gereksinimlerini incelemektedir. 3 katmanlı genel bir kablosuz sahayolu protokol mimarisi önermektedir. Kablosuz sahayolu sistemlerinde kablosuz iletişim standardı olarak IEEE 802.11 ve Bluetooth'un kullanımını önermişlerdir. Diğer çalışmalardan farklı olarak veri güvenliği ve güç tüketiminden de bahsetmektedirler.

Endüstriyel ağların arabağlaşımı için birçok farklı kablosuz iletişim standardı kullanılmasına rağmen, uygulanabilirlik, düşük maliyet, yaygın kullanım, vb. özelliklerden dolayı kablosuz Ethernet olarak da adlandırılan IEEE 802.11 KLAN üzerine daha çok çalışma yapılmaktadır. Bunun en önemli sebebi günümüzde oldukça yaygın bir kullanım oranına sahip standart kablolu Ethernet yapısı ile sağladığı kolay entegrasyondur.

Yukarıda verilen araştırma çalışmalarının yanı sıra bazı firmalar tarafından değişik ürünler de geliştirilmektedir. Örneğin RMCAN firmasının RF, UHF, Bluetooth, GSM, GPRS ve KLAN (WLAN) kablosuz arayüzlerin kullanıldığı ürünleri bulunmaktadır (<http://www.rmcan.com>). Geliştirilen bu ürünler köprü temelli arabağlaşım elemanı değil, sadece CAN temelli kablosuz arabirimlerdir. Bunun yanı sıra eğitim ve öğretim amaçlı geliştirme setleri de (evaluation board) bulunmaktadır. Tüm bu gelişmelere rağmen ideal bir çözüm henüz bulunamayıp, bu yönde çalışmalar ve öneriler devam etmektedir.

1.2. Tez Çalışmasının Amacı ve Önerilen Çözüm Yöntemi

Günümüzde saha seviyesindeki iletişim için birçok sahayolu sistemi kullanılmaktadır. Tüm sahayolu sistemlerinin olduğu gibi CAN temelli dağıtık kontrol sistemlerinin de dağıtık alanın büyüklüğü ve uzak CAN segmentlerin birbirleri ile ve diğer Yerel Alan Ağları (Local Area Network, LAN) ile haberleşme ihtiyacı olmak üzere iki temel problemi bulunmaktadır.

Endüstriyel kontrol uygulamalarında hareketli cihazların/modüllerin kullanımının artması ve kablosuz ürünlerin hızla yaygınlaşması dikkate alındığında yukarıda söz edilen CAN temelli dağıtık kontrol sistemlerinin problemlerini gidermek için en iyi çözümün kablosuz iletişim teknolojilerini saha seviyesinde kullanmak olduğu ortaya çıkmaktadır.

Bu tez çalışmasının amacı, CAN temelli dağıtık kontrol sistemlerinin IEEE 802.11b KLAN vasıtasıyla kablosuz ortam üzerinden birbirleri ve diğer yerel alan ağları ile haberleşmesini sağlayacak CAN/IEEE 802.11b Kablosuz Arabağlaşım Birimi tasarlamak, benzetim yöntemiyle gerçeklemek, başarıml analizini çıkarmak ve temel bir prototipini gerçekleştirerek kullanılabilirliğini değişik kontrol uygulamaları üzerinde göstermektir.

Bu tez çalışmasında önerilen çözüm yöntemi ve bu çalışmayı literatür özetinde verilen çalışmalardan ayıran önemli bazı noktalar özetle şunlardır:

- a. Sahayolu sistemlerinden CAN ve kablosuz iletişim teknolojilerinden IEEE 802.11b KLAN kullanılmıştır.
- i. CAN, başlangıçta yalnızca otomotiv uygulamaları için tasarlanmış olsa da yüksek hız, düşük maliyet, yüksek başarıml ve üstün karakteristik özelliklerinden dolayı birçok dağıtık endüstriyel kontrol uygulamalarında yaygın olarak kullanılmaktadır.
- ii. IEEE 802.11b KLAN kablosuz erişim teknolojisi, düşük maliyeti, uygulanabilirliği, yaygın kullanımı, mevcut kablolu Ethernet yapılarının kablosuz uzantısı olması gibi özellikleri sebebiyle tercih edilmiştir.

- b. Bu tez çalışmasında CAN ve IEEE 802.11b KLAN benzer olmayan iki ağın arabağlaşımını sağlayan CAN/IEEE 802.11b KLAN Kablosuz Arabağlaşım Birimi'nin;
- i. Gerçekleme aşaması için, daha önce yapılan CAN/Ethernet (Ekiz, 1997, Ekiz ve diğ., 1996, 1997), CAN/ATM ve PROFIBUS/ATM (Özçelik, 2002) yerel köprü tasarım yöntemleri esas alınarak gerçekleştirilen CAN/IEEE 802.11b KAB tasarım mimarisi ayrıntılı olarak sunulmaktadır.
- ii. OPNET Modeler™ yazılımı kullanılarak bilgisayar benzetim modeli gerçekleştirilmektedir ve SAE Benchmark'a göre KAB'ın kullanıldığı örnek bir endüstriyel ağın başarımları analizi verilmektedir.
- iii. Temel prototipi gerçekleştirilmektedir ve farklı kontrol uygulamalarında kullanımı verilmektedir.
- c. Literatür özetinde verilen tüm çalışmalarda özellikle önerilen arabağlaşım biriminin sebep olduğu gecikme üzerine odaklanılmaktadır. Gecikme analizine göre önerilen arabağlaşım elemanlarının sahayolu sistemlerinin gereksinimlerini karşılayıp/karşılayamayacakları ortaya konmaktadır. Ancak ayrıntılı bir tasarım mimarisi verilmemektedir. Bu tez çalışmasında ise bağımsız CAN segmentlerin IEEE 802.11b KLAN üzerinden haberleşmesini sağlayan KAB'ın hem ayrıntılı tasarımı hem de bilgisayarlı benzetim yöntemi ile başarımları analizi sunulmaktadır.
- d. CAN veri iletim hızı maksimum 1 Mbit/s'dir. IEEE 802.11b KLAN'ın veri iletim hızı ise maksimum 11 Mbit/s'dir. Böylelikle CAN ve KLAN ağların arabağlaşımını sağlayan KAB, CAN veri iletim hızını 11 kat arttırmaktadır. Bu hız endüstriyel kontrol uygulamalarının gereksinimlerini karşılayabilecek sınırlar içerisinde. Bu çalışma veri iletim hızından ziyade, CAN ve IEEE 802.11 KLAN ortam erişim mekanizmalarının birlikte çalışabilirliği üzerine odaklanılmaktadır. IEEE 802.11 KLAN standart ailesini oluşturan IEEE 802.11a/b/g gibi tüm alt standartlar aynı ortam erişim mekanizmasını (Carrier Sense Multiple Access with Collision Avoidance, CSMA/CA) kullanmaktadır. Aralarındaki temel fark, farklı veri iletim hızı sağlayan DSSS (Direct Sequence Spread Spectrum), FHSS (Frequency Hopping Spread Spectrum) ve OFDM (Orthogonal Frequency Division Multiplexing) gibi farklı fiziksel katman seçenekleri sunmalarıdır. Fiziksel katman seçeneğinin değiştirilmesi

gerçekleştirilen KAB'ın mekanizmasında büyük bir değişiklik gerektirmeyecektir. Tez çalışmasında fiziksel katman olarak DSSS kullanılmıştır.

1.3. Tez Çalışmasının Katkıları

Yapılan tez çalışmasının bilime ve endüstriye kazandırdığı yenilikler ve katkı aşağıda maddeler halinde sunulmaktadır:

- CAN temelli dağıtık kontrol sistemlerinin birbirleri ile ve diğer yerel alan ağları ile haberleşme gereksinimini karşılamak ve kablosuz ortam üzerinden haberleşmelerini sağlamak üzere CAN sahayolu sistemlerinde IEEE 802.11 KLAN standardının kullanımı önerilmiştir.
- Daha önce yapılan CAN/Ethernet (Ekiz, 1997, Ekiz ve diğ., 1996, 1997), CAN/ATM ve PROFIBUS/ATM (Özçelik, 2002) yerel köprü tasarım yöntemleri esas alınarak CAN sahayolu ile IEEE 802.11b KLAN iletişimini sağlamak üzere CAN/IEEE 802.11b KLAN Kablosuz Arabağlaşım Birimi tasarımı gerçekleştirilmiştir.
- Tasarımı gerçekleştirilen KAB, OPNET Modeler yazılımı kullanılarak bilgisayarlı benzetim modeli oluşturulmuş ve KAB'ın kullanıldığı endüstriyel bir ağ uygulamasının SAE Benchmark'a göre başarımlar analizi sunulmuştur.
- KAB'ın fiziksel gerçeklemelerini yapacak kişi, kurum ya da firmalar için örnek bir prototip gerçekleştirilmiş ve prototip farklı endüstriyel kontrol uygulamalarında kullanılmıştır.
- CAN ve IEEE 802.11b KLAN ağların birbirleri ile iletişimde çerçeve yapıları ve büyüklükleri, ortam erişim teknikleri, öncelik mekanizmaları, yönlendirme kararları gibi protokol farklılıklarından ortaya çıkan problem ve çözümleri bu çalışma içerisinde verilmiştir.
- OSI (Open Systems Interconnections) referans modelinin bir ve ikinci katmanlarını destekleyen herhangi bir sahayolu sistemi ile IEEE 802.11b KLAN kablosuz arabağlaşım birimi tasarımı için izlenebilecek bir yordam sunulmuştur.

1.4. Tez Organizasyonu

Tez organizasyonu ařađıda 6zetlenen 7 b6l6mden oluřmaktadır:

B6l6m 1: Giriř: Bu b6l6mde tez 6alıřmasına konu olan problemin tanımı, 6alıřmanın amacı, literat6rde bu problemin 66z6m6 6zerine yapılan 6alıřmaların 6zeti, tez 6alıřmasını literat6rde yapılan 6alıřmalardan ayıran temel 6zellikler ve tez 6alıřmasında izlenen y6ntem ile tez organizasyonu hakkında bilgi sunulmaktadır.

B6l6m 2: End6striyel İletiřim Sistemleri ve Denetleyici Alan Ađı (CAN): Bu b6l6mde end6striyel otomasyon uygulamalarında sahayolu sistemlerinin yeri, tanımı, kullanım ama6ları ve end6striyel 6retim ortamındaki iletiřimi tanımlayan otomasyon hiyerarřisi ve tez 6alıřmasına konu olan CAN sahayolu sistemi hakkında ayrıntılı bilgi verilmektedir.

B6l6m 3: Kablosuz Yerel Alan Ađları: Kablosuz ađların avantajlarından, kullanım alanlarından bahsedilmekte ve tez 6alıřmasının bir par6ası olan IEEE 802.11 kablosuz yerel alan ađlarının protokol mimarisi, standartları, 6er6eve bi6imleri ve 6zellikle ortam eriřim mekanizmasına ayrıntılı olarak deđinilmektedir.

B6l6m 4: End6striyel Ađlarda Arabađlařım ve CAN/IEEE 802.11b KLAN Kablosuz Arabađlařım Birimi Tasarımı: Bu b6l6mde ilk olarak ađların arabađlařımında kullanılan cihazlar kısaca tanıtılmaktadır. Ardından end6striyel sistemlerde arabađlařım gereksiniminin nedenleri, arabađlařım i6in sunulan kablolu ve kablosuz yapılar verilmekte ve literat6rde yapılmıř bazı 6alıřmalara deđinilmektedir. Aynı zamanda, ger6ekleřtirilen bu tez 6alıřmasını literat6rdeki benzerlerinden ayıran temel noktalardan bahsedilmektedir. Son olarak tez 6alıřmasını oluřturan ve bađımsız CAN segmentlerin kablosuz ortam 6zerinden birbirine bađlanmasını sađlayacak CAN/IEEE 802.11b KLAN Kablosuz Arabađlařım Biriminin tasarımı ayrıntılı olarak sunulmaktadır.

B6l6m 5: CAN/IEEE 802.11b KLAN Kablosuz Arabađlařım Biriminin Modellenmesi ve Bařarım Analizi: Ađ tabanlı sistemlerin modellenmesi i6in

kullanılan yaklaşımlar, bilgisayar benzetim yönteminin özellikleri ve tez çalışmasının gerçekleştirildiği OPNET Modeler programı hakkında bilgi verilmektedir. Aynı zamanda önerilen sistemin basit analitik modeli sunulmaktadır. Bölüm 4’de ayrıntılı tasarım mimarisi verilen CAN/IEEE 802.11 KLAN KAB’ın OPNET Modeler yazılımı kullanılarak gerçekleştirilen modelin tasarım aşamaları ve akış diyagramları ayrıntılı olarak verilmektedir. Ayrıca OPNET Modeler yazılımında standart CAN düğüm modeli olmadığından gerçekleştirilen CAN düğüm modelinin de tasarım aşamaları ve akış diyagramları da sunulmaktadır. Bununla birlikte OPNET yazılımında gerçekleştirilen CAN ve KAB düğüm modellerinin proses modellerine ait program kodları Ek–A ve Ek–B’de verilmektedir. Tasarlanan KAB’ın ve CAN düğümlerin kullanıldığı SAE Benchmark’a göre geliştirilen örnek bir endüstriyel ağ uygulaması benzetime tabi tutulmakta ve başarımları analizi sunulmaktadır.

Bölüm 6: CAN/IEEE 802.11b KLAN Kablosuz Arabağlaşım Birimi Prototipinin Gerçekleştirilmesi ve Uygulamaları: Bu bölümde, Bölüm 4 ve 5’de tasarımı ve benzetim modeli sunulan KAB’ın gerçekleştirilen prototipi verilmektedir. Bu prototipin kullanıldığı iki farklı endüstriyel kontrol uygulaması sunulmaktadır.

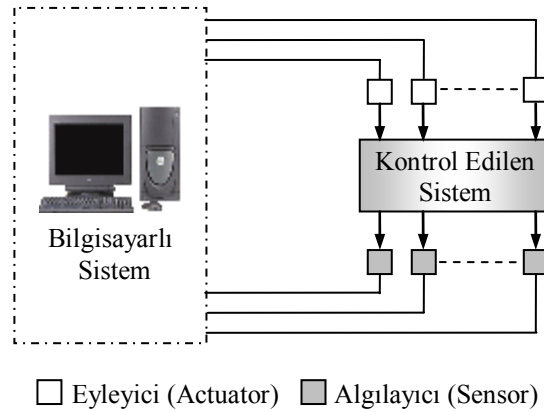
Sonuçlar ve Öneriler bölümünde, yapılan çalışmalardan elde edilen sonuçlar genel hatlarıyla değerlendirilerek çalışmanın bilime ve endüstriye sağlayabileceği katkılar tartışılmıştır. Daha sonra yapılabilecek çalışmalar için önerilerde bulunmaktadır.

2. ENDÜSTRİYEL İLETİŞİM SİSTEMLERİ VE DENETLEYİCİ ALAN AĞI

2.1. Giriş

Endüstriyel bir ortamda, üretime ve planlamaya katkıda bulunan tüm cihazların doğal olarak birbirleri ile haberleşmesi istenir. Bu cihazlar; sahadaki PLC, CNC, robot, kontrol cihazları, vb. olabildiği gibi, tesisin bürolarında kullanılan ve değişik firmalara ait olan bilgisayar sistemleri, modelleme bilgisayarları gibi çok çeşitli donanımlar olabilir (Özçelik, 2002).

Gelişen teknoloji ile birlikte günümüze kadar endüstriyel sistemlerin iletişimde büyük ilerlemeler yaşanmıştır. Endüstriyel ortamlardaki dağıtık sistemlerin kontrolü için izlenen geleneksel yöntemde, kontrol edilecek sistem üzerindeki her bir eyleyici/algılayıcı ayrı hatlar üzerinden merkezi kontrol noktasına bağlanmaktaydı (Şekil 2.1). Ayrı ayrı kablolama gereksinimi, endüstriyel sistemlere başta maliyet olmak üzere kurulum ve bakım gibi dezavantajlar getirmekteydi.



Şekil 2.1: Bilgisayar kontrollü dağıtık sistemler

Günümüzde bu dezavantajları ortadan kaldırmak için endüstriyel sistemlerin iletişimde sahayolu (Fieldbus) olarak adlandırılan ve süreç kontrol

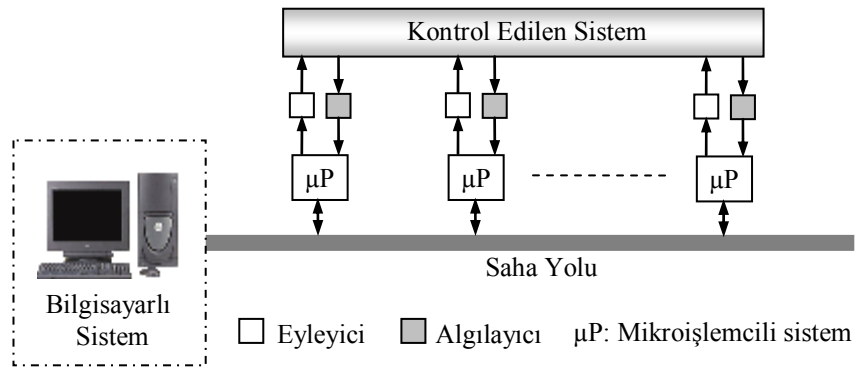
uygulamalarında kontrol sistemleri ile saha (field) elemanları arabağlantısını sağlayan sayısal haberleşme hatları kullanılmaktadır.

Bu bölümde endüstriyel sistemlerin iletişiminde kullanılan sahayolu sistemlerinin temel karakteristikleri, genel yapısı ve protokol mimarisi verilecektir. Ayrıca tez çalışmasında kullanılan Denetleyici Alan Ağı (CAN) ayrıntılı olarak incelenecektir.

2.2. Sahayolu Sistemleri

Endüstriyel iletişim sistemleri içerisinde geniş bir uygulama alanı bulan sahayolu ağları, özellikle Bilgisayarla Bütünleşik İmalat (BBI) zincirinin daha düşük seviyelerinde, algılayıcılar, eyleyiciler ve süreç denetleyicilerin arabağlantısı için tasarlanılmıştır (Lawrenz, 1997, Özçelik, 2002, Thomesse, 2005).

Şekil 2.2’de bir sahayolu sisteminin genel yapısı görülmektedir. Sahayolu sistemlerinin temel amacı endüstriyel kontrol uygulamalarındaki haberleşme için sistemdeki kablolamayı azaltmaktır. Tek kablo kullanımı başta maliyet kazancı olmak üzere, sisteme yeni denetleyici ekleme/çıkarma esnekliği, daha kolay kurulum ve bakım, güvenilirliğin artması ve daha kolay hata tespiti gibi avantajlar sağlar (Tovar, 1999).



Günümüzde endüstriyel iletişim sistemlerinde kullanılan çok sayıda sahayolu sistemi mevcuttur. Bunların en bilinenleri CAN, Profibus, WorldFIP, InterBus ve P-NET standartlarıdır. Bu sahayolu sistemlerinin her biri farklı zamanda, farklı

organizasyonlar tarafından ve farklı amaçlar için geliştirilmiştir. Uygulama alanlarına göre yapıları az ya da çok değişiklikler gösterebilmektedir. Bunlar arasındaki en temel fark fiziksel ortama erişim mekanizmalarından kaynaklanmaktadır (Lawrenz, 1997, Özçelik, 2002, Thomesse, 1999 ve 2005).

Tablo 2.1’de sahayolu sistemlerinin genel karakteristikleri verilmektedir (Patzke, 1998, Aydoğan, 2005).

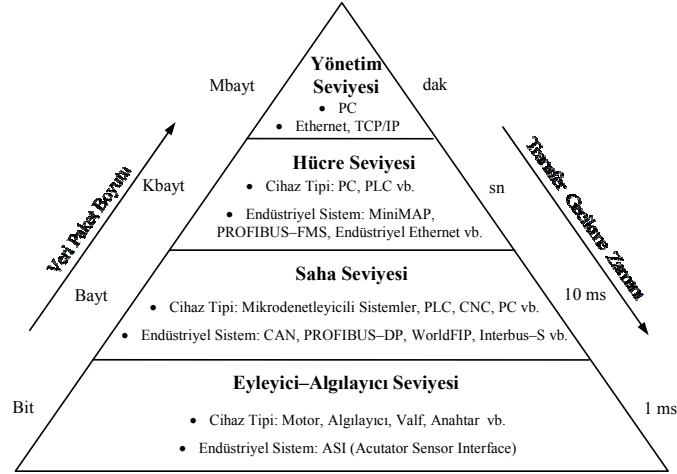
Tablo 2.1: Sahayolu sistemlerinin genel karakteristikleri

| | |
|----------------------------|--------------------------------------|
| Uygulama çeşitleri | Kontrol, otomasyon |
| Kullanılan veri tipi | 100 bayt’tan küçük katarlar |
| Etkileşim zamanı | $\ll 1$ sn |
| İstasyon tipi | Algılayıcılar, eyleyiciler, cihazlar |
| Gerçek zamanlı çalışabilme | Güçlü |

2.2.1. Otomasyon hiyerarşisi

Endüstriyel üretim ortamlarındaki haberleşme hiyerarşisi, bilgisayarla bütünleşik imalat (BBİ) mimarisi ile tanımlanır. Bu mimari, endüstriyel üretim ortamının hangi seviyelerinde ne tür haberleşme ve kontrol sistemleri olduğu hakkında bilgi verir.

Şekil 2.3’de birbirleriyle doğrudan ya da dolaylı olarak iletişimde bulunan Eyleyici-Algılayıcı, Saha, Hücre (Fabrika) ve Yönetim seviyelerinden oluşan BBİ modeli görülmektedir (Lawrenz, 1997, Özçelik, 2002). Hiyerarşideki seviyeler, bağlanan düğüm sayısı, istenen tepki zamanı, transfer edilecek bilgi miktarı, transfer sıklığı, istenen güvenilirlik, vb. faktörlerden dolayı benzer olmayan veri akışlarına sahip olurlar. Bu faktörlerden veri miktarı ile tepki zamanı arasındaki ilişki, BBİ modelindeki seviyeleri ortaya çıkartır.



Şekil 2.3: BBİ mimarisi

BBİ mimarisinin en alt seviyesini oluşturan Eyleyici-Algılayıcı seviyesinde endüstriyel üretim ortamındaki motor, anahtar, algılayıcı vb. cihazlar yer alır. Bu seviyedeki cihazlar kendi aralarında bilgi transferi gerektirmezler ve bir üst seviyedeki cihazların isteklerine göre çevrimli olarak veri transferi gerçekleştirirler.

Endüstriyel üretim ortamı içerisindeki dağıtılmış kontrol cihazları otomasyon hiyerarşisinin ikinci seviyesini oluşturan Saha seviyesinde bulunur. Bu seviyedeki cihazlar, üst seviyelerden herhangi bir bilgi almadan da kendi görevlerini yerine getirebilirler. Bu seviyedeki iletişim sahayolu sistemleri ile sağlanmaktadır. Günümüzde yaygın olarak kullanılan birçok sahayolu sistemi bulunmaktadır (CAN, Profibus-DP/PA, WorldFIP, Interbus-S, vb.). Sahayolu sistemleri bu seviyedeki cihazların birbirleri ile iletişiminin yanı sıra üst seviye ile iletişimlerini de sağlar.

Fabrika seviyesi olarak da adlandırılan Hücre seviyesi otomasyon hiyerarşisinin üçüncü seviyesini oluşturur. PLC, CNC, PC, vb. her türlü kontrol cihazının bulunduğu bu seviyede genellikle yüksek oranda veri transferi yapılmaktadır. Bu seviyedeki iletişim için çoğunlukla Endüstriyel Ethernet, Mini-MAP ve PROFIBUS-FMS iletişim sistemleri kullanılmaktadır.

Yönetim seviyesi BBİ modelinin en üst seviyesidir ve planlama seviyesi olarak da isimlendirilir. Ethernet ve TCP/IP (Transmission Control Protocol/Internet Protocol), bu seviyedeki ihtiyaçları karşılayan protokollerdir. Bu seviyede üretimin planlanması ile ilgili işlemler gerçekleştirilir (Özçelik, 2002).

2.2.2. Sahayolu sistemlerinin avantajları

Sahayolu sistemlerinin endüstriyel kontrol uygulamalarında kullanımının sağladığı başlıca avantajlar şunlardır:

- Yüksek performans ve düşük maliyet: Sahayolu sistemleri endüstriyel otomasyon uygulamalarında başta kablolamayı azaltma olmak üzere yeni eleman ekleme/çıkarma esnekliği, gerçek zamanlı iletişim, servis kalitesi desteği vb. özelliklerden dolayı yüksek performans ve düşük maliyet sağlamaktadır.
- Modülerlik ve geliştirilebilirlik: Kablolanmanın azalması ve mevcut sisteme kolaylıkla yeni eleman ekleme/çıkarma sistemin geliştirilebilmesini kolaylaştırmaktadır.
- Sürdürülebilirlik: Tek bir kablo kullanımı, kurulum ve bakım kolaylığı sağlar.
- Birlikte çalışabilirlik (interoperability) ve yer değiştirilebilirlik: Bir endüstriyel otomasyon sistemi farklı üreticilerin farklı ürünlerinden meydana gelmektedir. Sahayolu sistemi bu farklı cihazların aynı mesaj özellikleri ve aynı protokoller üzerinden birbirleri ile haberleşmesini sağlar. Yine arıza durumunda bir eleman aynı özelliklere sahip diğer bir eleman ile değiştirilebilir.
- Güvenilirlik ve kullanılabilirlik: Sahayolu sistemlerinde başarısız durumlarda sistem kontrolünün kaybedilmemesi amaçlanmaktadır. Kablo ve bağlantı elemanlarının sayısındaki azalma, sistemi basitleştirerek güvenilirliği arttırmaktadır (Cauffriez ve diğ., 2004, Lawrenz, 1997, Özçelik, 2002).

2.2.3. OSI referans modeli ve sahayolu sistemlerinin protokol mimarisi

Ağlar arası veri haberleşmesi, ISO (International Standards Organization) tarafından geliştirilen OSI referans modeline göre yapılmaktadır. OSI ağ veri haberleşmesi sırasında gerekli hizmet, özellik ve haberleşme kuralları ile mekanizmalarını katmanlar düzeyinde tanımlayan örnek bir modeldir.

Endüstriyel alan ağlarındaki standartlaştırma çalışmaları sonucunda OSI referans modeli esas alınarak Avrupa'da IEC (International Electrotechnical Commission) ve ABD'de ISA (Instrumentation Society of America) kurumları önderliğinde 3

katmanlı alan ağı modeli geliştirilmiştir. Şekil 2.4’de görüldüğü gibi sahayolu modeli, fiziksel, veri iletim ve uygulama katmanlarından meydana gelmektedir.

| <i>Katmanların İşlevleri</i> | OSI Modeli | | Sahayolu Modeli | |
|-----------------------------------|---------------------|---|------------------------|---------------------|
| <i>Kullanıcı Uygulaması</i> | Uygulama Katmanı | 7 | 3 | Uygulama Katmanı |
| <i>Veri Dönüşümü</i> | Sunum Katmanı | 6 | | |
| <i>Bağlantı Kontrolü</i> | Oturum Katmanı | 5 | | |
| <i>Mesaj Bütünlük Doğrulaması</i> | Ulaşım Katmanı | 4 | | |
| <i>Yönlendirme</i> | Ağ Katmanı | 3 | | |
| <i>Hata Tarama</i> | Veri İletim Katmanı | 2 | 2 | Veri İletim Katmanı |
| <i>Fiziksel Bağlantı</i> | Fiziksel Katman | 1 | 1 | Fiziksel Katman |

Şekil 2.4: OSI referans ve alan ağı modeli

Saha seviyesindeki cihazlar yalnızca birbirleri ile değil yönetim seviyesindeki PC’ler ile de iletişim içerisinde bulduklarından OSI referans modelinin tüm katmanlarının işlevlerini belirtmekte yarar vardır. Bu nedenle OSI katmanlarının tanımlanan görevleri aşağıda sıra ile kısaca açıklanmaktadır (Bayılmış, 2003, Çeken, 2004, Örencik ve Çölkesen, 2002).

7) Uygulama Katmanı (Application Layer): Uygulama programlarının ağa erişimi için gerekli işlevleri kapsar. Kullanıcı tarafından çalıştırılan ofis programları, paket yazılımlar, e-posta vb. uygulamalar bu katmanda tanımlıdır.

6) Sunum Katmanı (Presentation Layer): Bilginin iletimde kullanılacak biçiminin düzenlenmesini sağlar. Sıkıştırma/açma, şifreleme/şifre çözme, EBCDIC/ASCII dönüşümü ve ters dönüşümü gibi işlevlerin yerine getirilmesini sağlar.

5) Oturum Katmanı (Session Layer): İletişimde bulunacak iki nokta arasındaki oturumun kurulması, yönetilmesi ve sonlandırılmasını sağlar.

4) Ulaşım Katmanı (Transport Layer): Bu katman iki düğüm arasında mantıksal bir bağlantının kurulmasını sağlar. Ayrıca üst katmandan aldığı verileri parçalara (segment) ayırarak bir alt katmana iletir ve bir üst katmana bu parçaları birleştirerek sunar. Aynı zamanda akış kontrolü metotlarını kullanarak karşı tarafa gönderilen verinin yerine ulaşıp ulaşmadığını kontrol eder.

3) Ağ Katmanı (Network Layer): Veri paketlerinin yönlendirilerek alıcısına ulaşmasını sağlayan işlemlere sahiptir. Veri paketlerinin alıcısına giderken ağ koşullarına, önceliklere ve diğer parametrelere göre hangi yolun kullanılacağı bu katmanda değerlendirilir. Yönlendirme protokolleri bu katmanda çalışır.

2) Veri İletim Katmanı (Data Link Layer): Fiziksel adresleme, ağ topolojisi, hata ve akış denetimi gibi işlemler bu katmanda gerçekleştirilir. Veriler gönderilirken hata denetim bitlerinin eklenmesi, alıcı tarafta hatanın sezilerek düzeltilmesi, hatanın düzeltilmediği durumlarda verinin tekrar gönderilmesi vb. işlemler bu katmanda gerçekleştirilir. Anahtar ve köprü cihazları bu katmanda çalışır.

1) Fiziksel Katman (Physical Layer): Verinin fiziksel olarak bir haberleşme kanalı üzerinden aktarılması için gerekli işlemleri kapsar. Modülasyon teknikleri, çalışma gerilimleri, çalışma frekansları vb. ağın elektriksel ve fiziksel karakteristikleri bu katmanda belirlenir.

2.3. Denetleyici Alan Ağı (Controller Area Network, CAN)

Denetleyici Alan Ağı (CAN), endüstriyel ortamlardaki birçok dağıtık kontrol uygulamalarında kullanılan gerçek zamanlı haberleşme sistemidir. CAN, dağıtık gerçek zamanlı kontrolü, çok yüksek güvenlik düzeyinde ve etkin olarak destekleyen çekişme esaslı bir seri iletişim protokolü kullanır (Bosch, 1991, Lawrenz, 1997).

CAN, bir araç içerisinde bulunan elektronik kontrol birimleri arasındaki haberleşmeyi sağlamak üzere 80'lerin başlarında Robert Bosch GmbH firması tarafından geliştirilmiştir. Standart üretimlerine 1989'da Intel firması tarafından başlanmıştır. Aynı yıl Mercedes firması tarafından kullanımı ile otomotiv endüstrisinde zamanla tercih edilen bir teknoloji haline gelmiştir. Her ne kadar başlangıçta yalnızca otomotiv uygulamaları için tasarlanmış olsa da yüksek hız, düşük maliyet, yüksek başarımlar gibi nitelikleri sebebiyle çok kısa zamanda endüstriyel ortamlarda kullanılan veri yolları arasındaki yerini almıştır (Aydoğan, 2005, Ertürk, 2005, Özçelik, 2002).

CAN'in birçok endüstriyel uygulamada hızla yayılan kullanımı, kısa zaman içerisinde DeviceNET, SDS (Smart Distributed Systems), CAN Kingdom, CAN Open ve CAL (CAN Application Layer) gibi özel CAN tabanlı ağ çözümlerini ortaya çıkartmıştır (Ekiz, 1997).

Bu bölümde tez çalışmasının temelini oluşturan CAN ve IEEE 802.11 KLAN ağlar arasında arabağlaşımı sağlayan Kablosuz Arabağlaşım Birimi (KAB)'nin CAN kısmının çalışmasını anlayabilmek için CAN'in genel özellikleri ve çalışması incelenecektir.

2.3.1. CAN genel karakteristikleri

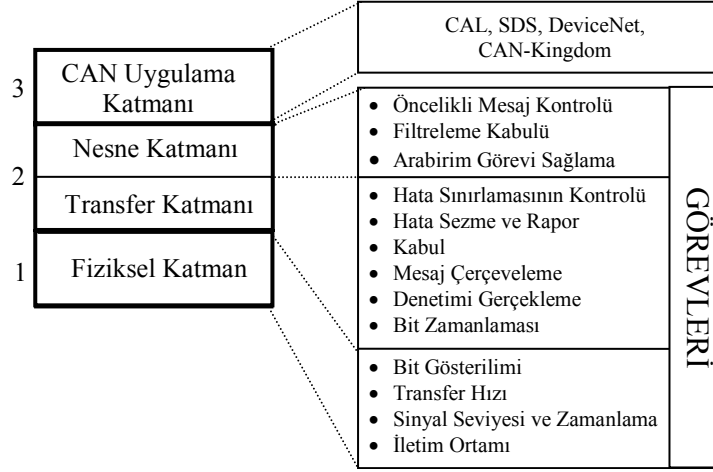
Tablo 2.2'de denetleyici alan ağının genel karakteristikleri özetlenmektedir.

Tablo 2.2: CAN genel karakteristikleri

| | |
|-----------------------------|---|
| Coğrafi Alan | LAN |
| Topoloji | Bus Topoloji |
| İletim Ortamı | Burulmuş-Çift (<i>Twisted-Pair</i>), Koaksiyel, Fiber |
| İletim Metodu | TemelBand (<i>Baseband</i>) |
| Kontrol Tipi | Dağıtık Kontrol |
| Ortam Erişim Kontrol Metodu | Taşıyıcı Duyarlı Çoklu Erişim (CSMA) |
| İletişim Tekniği | Yayın (<i>Broadcasting</i>) |
| Standartlar | ISO 11898 ve ISO 11519 |
| İletişim Protokolü | Seri İletişim |
| Maksimum Veri İletim Hızı | 1 Mbit/s |

2.3.2. CAN protokol mimarisi

CAN, tasarım saydamlığı ve gerçekleştirme esnekliğini sağlamak için yapısal olarak katmanlı halde geliştirilmiştir (Bosch, 1991). Şekil 2.5'de OSI referans modelinin Fiziksel, Veri İletim ve Uygulama katmanlarına karşılık gelen 3 katmanlı CAN protokol yapısı görülmektedir.



Şekil 2.5: CAN protokol mimarisi ve protokollerin görevleri

- 1) Fiziksel Katman: Elektriksel özelliklere dayalı olarak ağdaki farklı düğümler arasında mesaj bitlerinin nasıl iletileceğini tanımlar. Bu tanım içerisinde sinyal seviyesi, bit temsili ve iletim ortamı konuları bulunmaktadır. CAN, fiziksel katman üzerinde mükemmel bir hata sezme mekanizması sağlar (Özçelik, 2002).
- 2) Veri İletim Katmanı: Bu katman transfer ve nesne alt katmanlarından oluşmaktadır.
 - Transfer Katmanı: CAN protokolünün çekirdeğini (kernel) temsil eder. Bu katman, alınan mesajların nesne katmanına gönderilmesi ve nesne katmanından gönderilen mesajların kabul edilmesi ile ilgilidir. Aynı zamanda, bit zamanlaması, senkronizasyon, mesaj çerçeveleme, denetim mekanizması, kabul mekanizması, hata sezimi/sinyalleşme ve hata sınırlamasından (error confinement) da sorumludur.
 - Nesne Katmanı: Bu katman transfer katmanı tarafından alınan mesajların gerçekte kullanılıp kullanılmadığına karar verme, gönderilecek mesajların belirlenmesi (öncelik mekanizması kontrolüne göre) gibi görevleri gerçekleştirir (Özçelik, 2002).
- 3) CAN Uygulama Katmanı: Bu katmanda farklı uygulamaların özel ihtiyaçlarını karşılamak üzere CAN yapısına dayalı olarak geliştirilen bazı protokoller tanımlanmıştır. Her birisi, farklı bakış açıları ile farklı bir üstünlüğe sahip olan bu protokoller, CAN şartnamesine uygun olarak CAN iletişimine ve devrelerine

bağlıdır. Peraldi ve Decotignie (1995) çalışmalarında, CAN uygulama katmanı protokollerini karşılaştırılmalı olarak incelemektedir. Bunlar:

- SDS (Smart Distributed Systems) - Honeywell
- DeviceNet - AllenBradley
- CAN - Kingdom - Kvaser
- CAL (CAN Application Layer) – CiA (CAN in Automation)
- CAN Open - CiA (CAN in Automation)

2.3.3. CAN denetim mekanizması (arbitration mechanism)

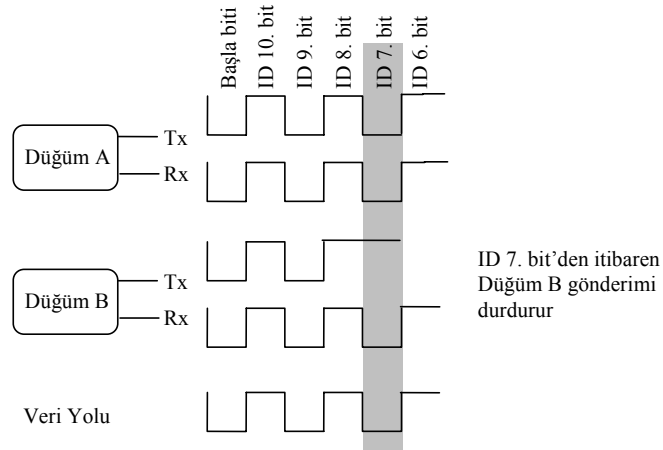
Ağ sistemlerinde veriyoluna erişim mekanizmaları, sistemlerin gerçek zamanlı uygulamaları destekleme yetenekleri ve erişim mekanizmasından kaynaklanan mesaj gecikmelerinin sistem performansına etkilerinden dolayı çok önemlidir. CAN çarpışmayı çözme ve gerçek zamanlı uygulamaları gerçekleştirme amacıyla tanıtıcı alan üzerinde öncelik esasına dayalı olan Taşıyıcı Duyarlı Çoklu Erişim/Mesaj Öncelik Denetimli Çarpışma Sezme (Carrier Sense Multiple Access/Collision Detection with Arbitration on Message Priority, CSMA/CD+AMP) protokolünü ortam erişim metodu olarak kullanır (Lawrenz, 1997, Ekiz, 1997, Farsi ve diğ., 1999, Aydoğan, 2005).

CAN yıkıcı olmayan çarpışma çözümü (non-destructive collision resolution) ve öncelik temelli ortam erişim yöntemi vasıtasıyla düğümlerin birbirine ortak yol (broadcast bus) topolojisi ile bağlandığı bir sistemdir (Navet ve Song 2001). Diğer sistemlerden farklı olarak denetleyici alan ağına bağlı düğümler herhangi bir adres bilgisine sahip değildir. Bunun yerine CAN düğümlerin ürettikleri her bir mesaj, tüm ağ içerisinde tek olan bir tanıtıcı (identifier) bilgisine sahiptir. Bu tanıtıcı bilgisi üretilen mesajların iletim önceliğini ve mesajların kabul/reddedilmesini belirler. Düşük değerli tanıtıcı bilgisine sahip mesaj ağ içerisinde daha yüksek önceliğe sahiptir (Bayılmış ve diğ., 2005b, Lawrenz, 1997, Navet ve Song, 2001).

Haberleşme ortamına erişmeye çalışan herhangi bir düğüm veriyolu (bus) boş olana kadar bekler, daha sonra tüm düğümlerin saatini senkronize etmek için ilk olarak

senkronizasyon bitlerinden başlayarak (mesajın ilk biti) mesajını bit bit gönderir. Eş zamanlı olarak birden fazla düğüm veriyoluna erişmeye çalışırsa çarpışma olur. Ancak CSMA/CD+AMP sayesinde her bir düğüm veriyoluna bit bit veri göndermesinin yanında eş zamanlı olarak veriyolunu da bit bit dinler. Böylelikle sürekli olarak gönderdiği veri ile veriyolu (bus) üzerindeki veriyi karşılaştırır. Bu işlemi bir VE (AND) kapısı kullanarak kolaylıkla yerine getirir. Eğer gönderdiği veri, veriyolundaki veriden farklı ise (tanıtıcı alan büyükse) derhal gönderme isteğini durdurur ve veriyolunu dinlemeye devam eder. Veriyolu üzerinde mesaj gönderen tüm düğümler veriyolunu dinleyerek en düşük değerlikli olan mesajı bulana kadar tanıtıcı alanlarını bit bit karşılaştırırlar. En yüksek öncelikli mesaj ek bir gecikme olmaksızın ilk olarak gönderilir. Mesajın gönderimi bittikten sonra aynı işlemler tekrar edilir (Lawrenz, 1997).

Şekil 2.6 CAN mesajın tanıtıcı alanı üzerindeki öncelik esasına dayalı çarpışmayı çözme yapısını özetlemektedir. Düğüm A ve B eş zamanlı olarak veriyoluna erişmişler ve senkronizasyon bitlerinden başlamak üzere veri iletimini bit bit gerçekleştirmektedirler. Her bir düğüm Tx ucundan veriyoluna iletilmek üzere verisini koyarken aynı zamanda da Rx ucundan veriyolunu dinlemekte ve her iki ucundaki bilgiyi de bit bit karşılaştırmaktadır. Tanıtıcı alanlarının ID7 numaralı bitine kadar her iki mesajın öncelik değeri aynıdır. Ancak Düğüm A'nın ID7 biti Düğüm B'nin ID7 bitinden küçük olduğunu Düğüm B fark ettiğinde veriyoluna iletim isteğini hemen durdurur. Böylelikle yüksek önceliğe sahip düğüm veri iletimine devam etmekte ve bu mesajın iletilmesinde herhangi bir gecikme yaşanmamaktadır.



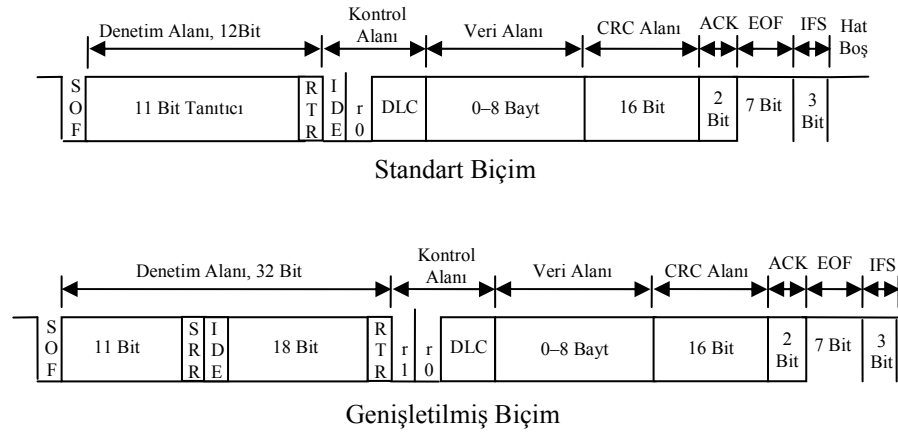
Şekil 2.6: Bit düzeyinde denetim mekanizmasının çalışması

2.3.4. CAN mesaj çerçeve tipleri

Denetleyici alan ağında Veri, Uzak (Remote), Hata ve Aşırıyük (Overload) çerçeveleri olmak üzere 4 farklı mesaj çerçevesi kullanılmaktadır.

2.3.4.1. Veri çerçevesi

Denetleyici alan ağında Standart ve Genişletilmiş biçim olmak üzere iki farklı veri çerçeve biçimi bulunmaktadır. Veri çerçeveleri arasındaki fark, denetim alanlarının uzunluğudur. Veri çerçevesi, kullanıcı verisinden başka veri akışını senkronize etmek, tanımlamak ve kontrol etmek için bilgi içerir.



SOF: Çerçeve Başlangıcı
DLC: Veri Uzunluk Kodu
CRC: Çevrimli Fazlalık Sınaması
SRR: Yedek Uzak İstek Biti
RTR : Uzak İletim İstek
ACK: Alındı Bilgisi
EOF : Çerçeve Sonu
IFS : Çerçeveler Arası Boşluk
IDE : Tanıtıcı Uzantı
r0 : Ayrılmış bittir, kullanılmamaktadır
r1 : Ayrılmış bittir, kullanılmamaktadır

Şekil 2.7: Standart ve Genişletilmiş CAN veri çerçeveleri

Veri çerçevesi, işlevleri kısaca aşağıda açıklanan her biri farklı uzunluklarda 7 alandan oluşur.

Çerçeve Başlangıcı (Start of Frame, SOF): 1 bit büyüklüğündedir ve CAN mesajın başlangıcını belirtir.

Denetim Alanı (Arbitration Field): Standart veri çerçevesinde 12 bit, Genişletilmiş veri çerçevesinde 32 bittir. Standart biçimde Tanıtıcı alanı (11 bit) ve Uzak İletim

İstek (Remote Transmission Request, RTR) (1 bit) alanlarından oluşur. Tanıtıcı alanı, mesajların iletim önceliğini ve kabul edilip/edilmemesini belirlemek amacıyla kullanılır. Daha düşük değerlikli tanıtıcı alana sahip veri çerçevesi daha yüksek iletim önceliğine sahiptir. 11 bit tanıtıcı alana sahip Standart çerçeve biçimi ile teorik olarak 2048 (2^{11}) farklı öncelik değerine sahip mesaj üretilebilirken bu değer 29 bit tanıtıcı alana sahip Genişletilmiş çerçeve biçimi ile 536.870.912 (2^{29})'dir. 1 bit büyüklüğündeki uzak iletim istek alanı bir düğümün diğer bir düğümden bilgi istemesi durumunda haberleşmeyi başlatmak için kullanılır. RTR biti gönderilen CAN çerçevesinin veri ya da uzak çerçeve olup olmadığını gösterir. RTR lojik '0' ise veri çerçevesi, lojik '1' ise uzak çerçevedir.

Kontrol Alanı (Control Field): Bu alan 1 bit tanıtıcı uzantı alanı (Identifier Extension, IDE), ileride kullanım için ayrılmış 1 bitlik r0 alanı ve gönderilen verinin boyutunu gösteren 4 bitlik veri uzunluk kodu (Data Length Code, DLC) olmak üzere 6 bitten oluşur. IDE lojik '0' ise gönderilecek herhangi bir tanıtıcı bilgisi olmadığını gösterir.

Veri Alanı (Data Field): Veri alanı DLC değerine bağlı olarak sıfır ile sekiz bayt arasında değişen uzunluğa sahiptir. CAN protokolünde veri bölümünün en düşük adresinde olan bayt, ilk gönderilir.

Çevrimli Fazlalık Sınama Alanı (Cyclic Redundancy Check, CRC): 15 bit CRC dizisi ve 1 bitlik yüksek seviyeli CRC belirticiden (CRC delimiter) oluşur. Bu alana mümkün olan iletim girişimlerini sezmek için başlangıç biti, denetim alanı, kontrol alanı, veri alanı ve CRC alanlarını kapsayarak hesaplanan bir kontrol kod yazılır. Bir mesajın gönderimi tamamlandıktan sonra CRC alanı kontrol edilir. Eğer çerçevede herhangi bir hata sezilmişse, tüm çerçeve yeniden gönderilir.

ACK Alanı (ACKnowledgment Field): Birer bitlik ACK slot ve ACK belirtici alanlarından oluşur. Bu alan mesajın alınıp alınmadığını ve herhangi bir hatanın sezilip sezilmediği hakkında gönderici düğümü bilgilendirir. Kaynak düğüm veri çerçevesinin ACK alanının her iki bitini lojik '1' seviyesinde gönderir. Hedef düğüm mesajı doğru olarak aldığı anda ACK slot alanı esnasında gönderici düğüme lojik '0'

seviyeli bir bit gönderir. Kaynak düğüm veriyolunu dinlediğinden dolayı bu değişikliği algılar ve en az bir düğümün mesajı tam ve doğru olarak aldığını anlar.

Çerçeve Sonu (End of Frame): Veri ve uzak çerçevelerinin tamamlandığını, ACK alanından sonra mesajın sonuna eklenen yedi adet lojik '1' seviyeli bit dizisi belirtir.

Çerçeveler Arası Boşluk (Inter Frame Space, IFS): İletimi senkronize ve kontrol etmek için veri çerçeveleri arasında minimum 3 bitlik bir çerçeveler arası boşluk gereklidir. Aksi takdirde hata çerçeveleri veya aşırıyük çerçeveleri, çerçeve sonu belirtecinden hemen sonra başlayabilir. IFS'den sonra veriyolu yeni bir iletme kadar boş durumdadır.

Genişletilmiş biçimde ise Standart biçimdeki alanlara ek olarak RTR bitinin işlevini yerine getiren Yedek Uzak İstek Biti (Substitute Remote Request, SRR) ve ileride kullanım için ayrılmış r1 biti eklenmiştir. Genişletilmiş biçimde IDE alanı, diğer tanıtıcı bitlerin kullanılacağını gösterir ve lojik '1' seviyesinde tutulur (Lawrenz, 1997, Özçelik, 2002).

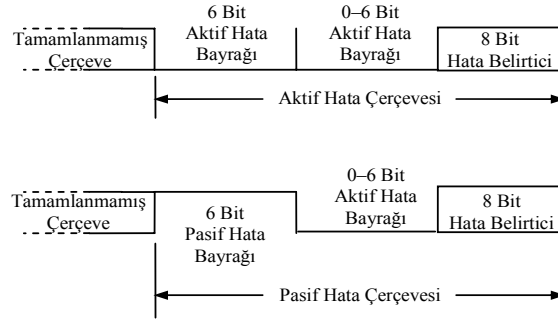
2.3.4.2. Uzak çerçeve

Belirli bir veri için alıcı olarak davranan bir istasyon, kaynak düğüme bir uzak çerçeve göndererek kendi verisinin iletimini başlatabilir (Bosch1991). Örnek olarak otomobil gösterge (dashboard) denetleyicisinin, motor yönetim sisteminden motor sıcaklığı hakkında bilgi istemesi verilebilir. Bu durumda otomobil gösterge denetleyicisi, bir veri çerçevesi ile cevap verecek olan motor denetleyicisine bir uzak çerçeve gönderir. Motor yönetim sistemi de gösterge denetleyicinin istediği veriyi aynı denetim alanı ile veri çerçevesi olarak gönderir (Ekiz, 1997).

Veri çerçevesindeki uzak iletim istek (RTR) bitinin lojik '1' yapılması ile uzak çerçeve oluşturulur. İletim isteğinde bulunulan uzak çerçevelerde herhangi bir veri gönderilmez. Çerçevde DLC alanında herhangi bir değer tanımlanmış olsa bile çerçeve veri bilgisi içermez (Özçelik, 2002).

2.3.4.3. Hata çerçeveleri

CAN, veriyolu boyunca veri tutarlığını garantilemek için örnekleme noktaları, yayılım esnasındaki sinyal bozulmaları, farklı anahtarlama eşikleri gibi sebeplerden meydana gelen veriyolu hatalarını hata çerçeveleri ile temsil eder (Özçelik, 2002, Pinho ve Vasques, 2003). Şekil 2.8’de denetleyici alan ağında kullanılan hata çerçeveleri görülmektedir. Hata çerçeveleri hata bayrağı ve hata belirtici (error delimiter) alanlarından oluşur. Bu çerçeveler aynı seviyeye sahip altı ardışık bitli bir bayrak ile gösterilir. Bu yapı dolgu biti (stuff bit) hatası oluşturduğundan ağdaki tüm düğümler veriyolu üzerinde bir hatanın meydana geldiğini anlarlar. Hata bayrağının bitleri, bozulan veri çerçevesinin üzerine yazılır. Bir düğümün hatayı sezmesi sonucunda diğer düğümler de ilgili hata durumunu sezerler ve kendi hata bayraklarını gönderirler. Bu dizinin toplam uzunluğu minimum altı bit ve maksimum on iki bit arasında değişir. Hata belirtici sekiz adet ‘1’ seviyeli bit içerir. Hata bayrağının iletiminden sonra ‘1’ seviyeli bitlerin gönderilmesi ile hata çerçevesi tamamlanır. Hata belirtici ve aradan (intermission) sonra verici, bozulan mesajı yeniden iletmek amacıyla tekrar veriyoluna erişmeyi dener (Lawrenz, 1997).

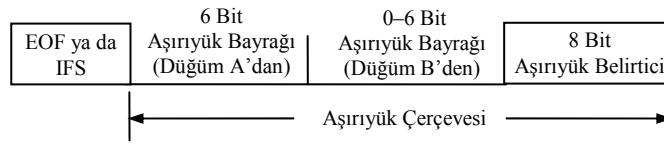


Şekil 2.8: CAN hata çerçeve biçimleri

CAN’de hata yönetim mekanizması sayesinde hatalı ya da tamamen bozulmuş bir düğümden gönderilen hata çerçevelerinin veriyolunu tamamen işgal etmesi önlenir. Bu mekanizma sayesinde düğüm, bulunduğu hata durumuna göre hata çerçeveleri üretir. Veri çerçevesi veya uzak çerçevenin iletimi esnasında sezilen hatalar, mevcut çerçevenin iletim zamanı içerisinde işlenir. Bu yordam veri veya uzak çerçeve ile bir hata çerçevesini birleştirir (Ekiz, 1997).

2.3.4.4. Aşırıyük çerçevesi

Aşırıyük çerçevesi, bir düğümün bir sonraki çerçeveyi almadan önce mevcut veriyi işlemek için yeterli zamana sahip olmadığını belirtmek için kullanılır. Bu çerçeve ile alıcı düğüm bir sonraki iletimin başlamasını geciktirmek ister. Aşırıyük çerçevesi ile aktif hata çerçevesi hemen hemen aynıdır. Aşırıyük çerçevesini aktif hata çerçevesinden ayıran fark, bu çerçevenin çerçeve sonu ya da IFS alanından sonra başlamasıdır.



Şekil 2.9: CAN aşırıyük çerçevesi

2.3.5. CAN hata mekanizması

CAN, temel ve en önemli özelliği olan çok yüksek güvenlik seviyesini içerisindeki “Hata Yönetimi Birimi” ile yerine getirir. Bu birimin işlevleri CAN protokol yapısındaki veri iletim katmanında (Data Link Layer, DLL) tanımlanmaktadır.

Denetleyici alan ağındaki bir düğüm hata tespit ettiğinde, ağıdaki diğer tüm düğümleri ivedi olarak hatanın varlığından bilgilendirir. Bu hata mesajından sonra ağıdaki düğümlerin hepsi aldıkları bitleri elimine ederler (discard). Hatalı mesajı gönderen düğüm, CAN’ın gönderilen bitlerin geri okunması özelliği vasıtasıyla bu hata hakkında uyarılır. Bu durumda gönderdiği mesajın hatalı olduğunu öğrenen düğüm, veriyolu boşaldığında mesajını tekrar gönderecektir. Bu işlem CAN denetleyici tarafından otomatik olarak yapılır. Tespit edilen hatanın bildirimi Şekil 2.8’de görülen “Hata Çerçeveleri” gönderimi ile yapılır (Lawrenz, 1997).

Düğüm hatayı tespit ettiğinde veriyolu üzerindeki tamamlanmamış mesajın hatalı kısmından itibaren 6 ardışık biti Aktif hata için baskın (lojik 0), Pasif hata için baskın olmayan (lojik 1) yapar. Böylelikle ağ üzerindeki herhangi bir düğüm bu mesajı gördüğünde Bit Dolgu hatasını tespit eder.

Eğer denetleyici alan ağı içerisinde bir düğüm sürekli hata çerçeveleri gönderirse, böyle bir arıza durumundan ağa bağlı diğer düğümleri korumak için hata çerçevesi üreten düğümün ağ ile olan bağlantısı adım adım koparılır.

Hata yönetim birimi tarafından yürütülen hata kontrol süreci hata tespiti, hata mekanizmasının işleyişi ve hata sınırlaması olmak üzere üç bölüme ayrılmıştır (Lawrenz, 1997, Özçelik, 2002, Navet ve Song, 2001, Gaujal ve Navet, 2005).

1) Hata Tespiti (Error Detection): Hata yönetim birimi beş farklı hata türünü sezme kabiliyetine sahiptir. Bunlar;

- Bit Hatası: Veriyoluna bit gönderen bir düğüm aynı zamanda veriyolunu da izler. İzleme bit bit yapılır. İzlenen bit seviyesi, gönderilen bit seviyesinden farklı olduğunda bir bit hatası tespit edilir.
- Bit Dolgu Hatası: Gönderimde bulunan bir düğüm normalde beş ardışık '1' seviyeli bittten sonra '0' seviyeli bir bit veya beş ardışık '0' seviyeli bittten sonra '1' seviyeli bir bit ekler. Bu işleme bit dolgulama denir. Eğer alıcı düğüm altı ardışık eşit bit seviyesi algılasa, bit dolgu hatası tespit etmiş olur.
- Çevrimli Fazlalık Sınaması (CRC) Hatası: Alınan CRC dizisi ile hesaplanan CRC değerinin birbirine eşit olmama durumunda ortaya çıkar.
- Çerçeve (Biçim) Hatası: CAN mesaj çerçevesi içerisinde belirli pozisyonlarda gönderilmesi gereken (CRC sınır belirteci, ACK sınır belirteci ve EOF) önceden tanımlı bit değerleri vardır. Alıcının bu pozisyonlarda geçersiz bir bit algılama durumu ile çerçeve hatası tespit edilir.
- Kabul Hatası: Gönderimde bulunan bir düğüm gönderdiği mesajın kabul edilmediğini tespit ettiğinde meydana gelen hata türüdür (Kaynak düğümün ACK slot boyunca bir baskın '0' seviye izleyememe durumudur).

2) Hata Mekanizmasının İşleyişi (Error Handling): Yukarıda bahsedilen hatalardan biri tespit edildiğinde derhal bir hata çerçevesi üretilecektir. Hata mekanizmasının işleyişi aşağıdaki sırayla gerçekleşmektedir.

- Hata tespit edilir.
- Derhal bir hata çerçevesi gönderilir.
- Ağdaki tüm düğümlerde hatalı mesaj yok edilir.
- Ağdaki tüm düğümlerin hata sayıcıları 1 arttırılır.
- Mesaj iletimi tekrar edilir.

3) Hata Sınırlaması (Error Limitation): Gönderilen hata çerçevelerinin birçok kez tekrar edilmesi gibi olaylar ile bir düğümün tüm sistemi bozmasını engellemek için CAN hata sınırlama mekanizmaları içerir. Hata sınırlamasının amacı, hatalı düğümün tespit edilip veriyolu ile olan bağlantısının koparılmasıdır. CAN bu işlemi gerçekleştirmek için iki farklı hata sayıcıya sahiptir. Bunlar:

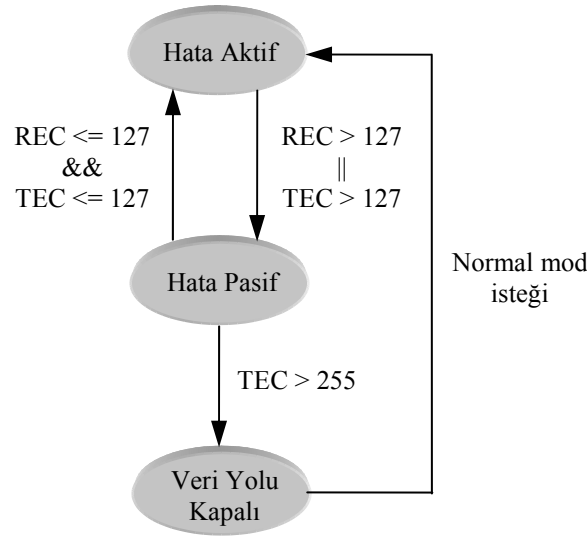
- Gönderim Hata Sayıcı (Transmit Error Counter, TEC): Düğümün gönderdiği çerçeveler üzerinde tespit edilen iletim hatalarını sayar.
- Alım Hata Sayıcı (Receive Error Counter, REC): Düğümün aldığı çerçeveler üzerinde tespit edilen hataları sayar.

Bir çerçevenin doğru alındığı veya gönderildiği durumda ilgili sayıcı azaltılır. İletim hatasının olduğu her zaman ise ilgili sayıcı arttırılır. Her bir CAN denetleyici Hata Aktif, Hata Pasif ve Veriyolu Kapalı olmak üzere üç farklı hata durumuna sahiptir. Bir düğüm her iki sayıcının durumuna bağlı olarak bu üç hata durumundan birinde olacaktır.

- Hata Aktif (Error Active): CAN denetleyicinin normal çalışma durumudur ve denetleyici başlangıç durumunda bu konumdadır. Bu durumda düğüm mesaj alabilir ve gönderebilir. Hata tespit edildiğinde Aktif hata bayrağı gönderilir. REC ve TEC sayıcılarının değeri 127'ye eşit veya küçükse düğüm bu durumdadır.
- Hata Pasif (Error Passive): Bu durumdaki bir düğüm, mesaj alabilir ve gönderebilir. Bir hata tespit edildiğinde, Pasif hata bayrağı gönderilir. Ancak hata pasif konumundaki bir düğüm, mesaj iletiminden sonra bir sonraki mesajı iletmeden önce belirli bir süre bekleyecektir. Bu bekleme son iletimin bitimini gösteren bitten sonraki 8 ilave bit kadarki süredir. REC ve TEC hata

sayıcılarından en az birinin değeri 127'den büyük ve TEC'in değeri 255'e eşit veya küçükse CAN denetleyici hata pasif durumundadır.

- Veriyolu Kapalı (Bus-Off): Hata yönetimi biriminin isteği ile denetleyicinin veriyolu ile olan bağlantısı kapatılır. Bu durumda CAN denetleyici mesaj alamaz ve gönderemez. Denetleyici mesajların iletimi ile ilgili ciddi problemlere sahip olduğunda bu duruma girer. TEC hata sayıcısının değeri 255'den büyük olduğunda düğüm, veriyolu kapalı durumuna geçer. CAN denetleyicinin hata aktif durumundan hata pasif durumuna geçişi otomatik olarak meydana gelmekteyken denetleyicinin veriyolu kapalı durumundan ayrılması yazılım ya da donanımsal olarak sıfırlanması (reset) ile gerçekleşir (Lawrenz, 1997, Navet ve Song, 2001).



Şekil 2.10: Hata durumları

TEC ve REC hata sayıcılarının değişimi aşağıdaki şekilde özetlenebilir (Gaujal ve Navet, 2005, Navet ve Song, 2001, Lawrenz, 1997).

1. Çerçeve iletimi başarılı ise

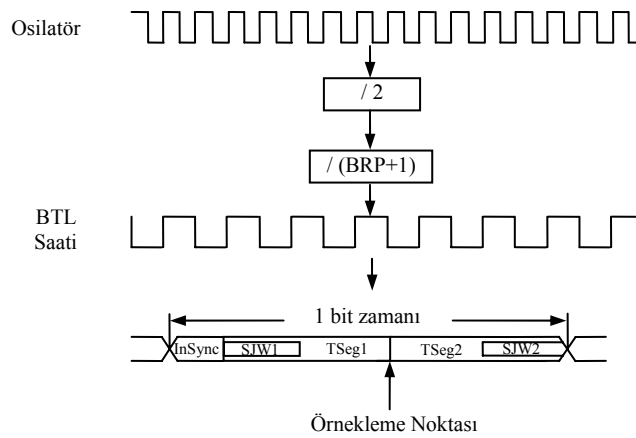
- Düğüm gönderen düğüm değilse: Eğer REC 1 ile 127 arasında ise 1 azaltılır, 0 ise değiştirilmez. Eğer REC 127'den büyükse hata sayıcı 119 ile 127 arasında rasgele bir değer alır.
- Düğüm gönderen düğüm ise: TEC 0 değilse 1 azaltılır. Aksi durumda değişmez.

2. İletim başarısız ise (iletim hatası varsa)
 - Düşüm gönderen düğüm değilse: REC 1 arttırılır.
 - Düşüm gönderen düğüm ise: TEC 8 arttırılır.

2.3.6. CAN bit zamanlaması ve senkronizasyon

Denetleyici alan ağındaki her düğüm dahili osilatörü ile bireysel olarak saatlenir (clock). Çerçeve alımındaki faz kaymalarına sebep olan bu durumu engellemek için her CAN denetleyici özel bir senkronizasyon algoritmasını destekler. Bu algoritma bit zamanlama mantık birimi (Bit Timing Logic, BTL) tarafından gerçekleştirilir.

CAN ağı sağlıklı çalışabilmesi için bir bitlik örnekleme zamanı (bit time) önemlidir. Kullanılan CAN türüne bağlı olarak bir bit zamanı, InSync Segment, Time Segment 1 (TSeg1) ve Time Segment 2 (TSeg2) olmak üzere 3 alt kısımdan meydana gelmektedir. Şekil 2.11’de görüldüğü üzere bu segmentler BTL periyodunun belirli bir miktarından oluşmaktadır. BTL, osilatör ve Baud Rate Prescaler (BRP) yardımıyla tanımlanır. Bir CAN çerçevesinin alımı sırasında meydana gelen faz kaymalarını telafi etmek için TSeg1 ve TSeg2 içerisinde yer alan senkronizasyon atlama genişliği (Synchronization Jump Width, SJW) kullanılır. SJW’nin değeri ayarlanarak bir bit zamanı uzatılabilir veya kısaltılabilir. Örnekleme zamanı TSeg1 ve Tseg2’nin orta noktasıdır.



Şekil 2.11: CAN’de bit zamanlaması

Veri bitlerinin mesajın toplam bit sayısına oranı olan veri iletim hızı, mesajın tipine, veri büyüklüğüne ve iletim hızına bağlıdır. Veri iletim hızı anlamına gelen bit zamanı, quartz osilatör, BRP ve BTL periyotların miktarı (T_{Seg1} ve T_{Seg2}'nin toplamı) ile belirlenir.

$$\text{Bit Zamanı} = t_{\text{InSync}} + T_{\text{TSeg1}} + T_{\text{TSeg2}} \quad (2.1)$$

$$= [1 * t_{\text{BTL}}] + [(T_{\text{TSeg1}} + 1) * t_{\text{BTL}}] + [(T_{\text{TSeg2}} + 1) * t_{\text{BTL}}] \quad (2.1a)$$

$$t_{\text{BTL}} = (\text{BRP}+1) * 2 * t_{\text{XOSC}} \quad (2.1b)$$

T_{Seg1}, T_{Seg2} ve BRP bit zamanlama kaydedicilerinin ilgili alanları vasıtasıyla programlanan sayısal değerlerdir. CAN programlanan bu değerlere göre mesajlarını göndermektedir (Lawrenz, 1997).

En yüksek önceliğe sahip bir mesaj, CAN için en yüksek veri iletim hızı olan 1 Mbit/s'de (1 Mbit/s veri iletim hızında bir bitin iletim zamanı 1 μ s'dir) kullanılan CAN tipine göre mesajın toplam bit sayısı miktarınca bit μ s'dir. Yani CAN 2.0A mesaj tipini kullanan bir sistemde en yüksek önceliğe sahip bir mesajın veriyoluna erişimi için geçen maksimum gecikme süresi 130 bit zamanıdır (130 μ s). Bu süre 29 bit tanıtıcı alanın kullanıldığı CAN 2.0B için 154 μ s'dir (Lawrenz, 1997).

2.3.7. CAN mesaj yanıt zaman analizi

Sahayolu düzeyindeki haberleşmede, veri büyüklükleri bit, bayt ve iletişim zamanları ms (milisaniye) seviyesindedir. Bu özelliklerdeki haberleşme türüne gerçek zamanlı (real-time) haberleşme adı verilir (Patzke, 1998, Thomesse, 1999).

Gerçek zamanlı sistemler rahatsız etmeyen (soft) ve kesin (hard) gerçek zamanlı olmak üzere genellikle ikiye ayrılır. Rahatsız etmeyen gerçek zamanlı sistemde bazen mesajlar geç varabilir. Fakat bu durum sistemin çalışmasını genellikle engellemez. Kesin gerçek zamanlı sistemde ise mesajlar kesinlikle belirtilen zaman içerisinde ulaştırılmalıdır (Lawrenz, 1997).

CAN kesin gerçek zamanlı (hard real-time) bir sistemdir. CAN tarafından transfer edilen mesajların gerçek zaman gereksinimlerini garanti etmek için mesajların en kötü durum (worst-case) yanıt zamanlarını değerlendirmek gereklidir (Pinho ve Vasques, 2003). Tindell ve diğ. (1995)'de denetleyici alan ağlarının mesaj yanıt zaman analizi ayrıntılı olarak ele alınmaktadır.

CAN'de ortam erişimi tanıtıcı alan üzerindeki öncelik esasına göre gerçekleştiğinden mevcut sabit öncelik zaman çizelgelemesi (fixed priority schedulability) bir CAN ağı uygulanabilir (Pinho ve Vasques, 2003, Lawrenz, 1997).

N tane mesaj akışı ile bir CAN ağ analizi aşağıdaki gibi tanımlanabilir.

$$S_m = (C_m, T_m, D_m) \quad (2.2)$$

S_m = m tanıtıcı değere sahip mesaj akışı (stream m) (m ağ boyunca tek bir değerdir).

C_m = mesaj m'nin (S_m) iletim süresidir.

T_m = mesaj m'nin üretim aralık zamanıdır.

D_m = S_m 'nin varma sınır değeri (deadline time).

Genellikle bir CAN ağda her mesaj için bir varma sınır değeri (D_m) tahsis edilir ve mesajın varma gecikmesi, D_m ile karşılaştırılır. Bundan dolayı ağdaki tüm mesajlar için tanıtıcı alan tahsisi şu şekilde yapılabilir. En kısa D_m 'ye sahip mesaja en yüksek öncelik değeri tahsis edilir.

Bir CAN mesajın en kötü durum gecikmesi (R_m) aşağıdaki gibi tanımlanır. R_m kuyruklama ve iletim gecikmelerinin toplamıdır.

$$R_m = t_m + C_m \quad (2.3)$$

R_m = mesaj m'nin gecikme süresi.

t_m = mesaj m'nin kuyruğa atılmasından veriyolu erişimi kazanmasına (arbitration mechanism) kadar geçen süre.

C_m = mesaj m'nin veriyolu üzerindeki iletim zamanı.

Mesaj iletim zamanı (C_m), mesajın maksimum yük büyüklüğünün (data length), mesaj içerisindeki dolu bitlerinin maksimum sayısının ve veriyolunun bit iletim zamanını bilinmesi ile hesaplanabilir.

$$C_m = \left(\left[\frac{34 + 8d}{4} \right] + 47 + 8d \right) \tau_{\text{bit}} \quad (2.4)$$

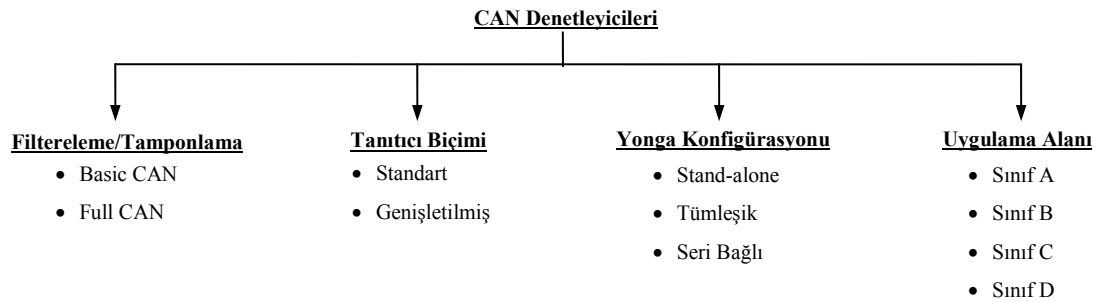
d = mesaj yükünün (data) bayt olarak büyüklüğü.

τ_{bit} = veriyolunun bir bit iletim zamanı (1 Mbit/s veri iletim hızı için 1 μ s'dir).

2.3.8. CAN entegrelerinin sınıflandırılması

Farklı uygulamaların farklı gereksinimlerini karşılamak için uygun CAN denetleyici seçilmelidir. CAN denetleyici yonga üreticilerinin neredeyse tamamı CAN şartnamesine uygun olduklarını ve birlikte çalışabildiklerini garanti ederler (Özçelik, 2002, Waizmann ve Sternert, 1994). Entegre seçiminde gerçekleştirme maliyeti, tasarım esnekliği, sistem güvenliği vb. kriterler göz önüne alınmalıdır.

Genel olarak CAN denetleyicileri, filtreleme/tamponlama yapısına, tanıtıcı biçimine, yonga konfigürasyonuna ve uygulama alanları olmak üzere dört farklı kritere göre sınıflandırılabilir (Şekil 2.12).



Şekil 2.12: CAN denetleyicilerinin sınıflandırılması

Filtreleme/Tamponlama (mailbox) yapısına göre CAN denetleyicileri Basic CAN ve Full CAN olmak üzere ikiye ayrılır. Basic CAN daha ucuz, Full CAN ise daha az iş yüküne sahiptir. Full CAN bir denetleyici de mesajlar CPU tarafından erişilen çift portlu bir RAM'de tutulurken, Basic CAN bir denetleyici de ise mesajlar CPU'nun

belleğinde tutulur. CPU, mesajların izlenmesi ile ilgili tüm işlemleri yapmalıdır. Kısaca mesaj filtreleme/tamponlama Full CAN denetleyicide donanımsal olarak gerçekleştirilirken, Basic CAN denetleyicide ise uygulama tarafından gerçekleştirilmektedir.

Tanıttıcı biçimine göre CAN denetleyici, 11 bit tanıttıcı alana sahip Standart biçim (CAN 2.0A) ve 29 bit tanıttıcı alana sahip Genişletilmiş biçim (CAN 2.0B) olarak ikiye ayrılır. Genişletilmiş çerçeve biçimi, Amerikan J1850 çoğullama standardı ile uyuşan tanıttıcılara izin vermek için geliştirilmiştir (Ekiz, 1997).

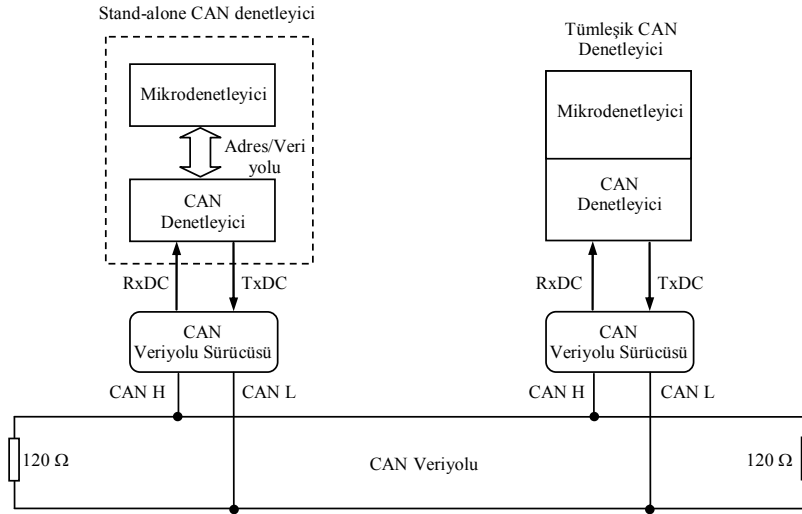
Yonga konfigürasyonuna göre CAN denetleyiciler, Stand-alone, Tümlüşik ve Seri Bağlı olmak üzere üç farklı tiptedir. Stand-alone CAN denetleyici adres ve veriyolları ile harici olarak birbirine bağlanan CAN yongası ve bir host mikrodenetleyiciden oluşur. Bu yapı tasarım esnekliği sağlamaktadır. Tümlüşik CAN denetleyicide ise CAN denetleyici ve mikrodenetleyici tek bir parça halindedir. Aralarındaki iletişim dahili olarak gerçekleşir. Tek bir yonga halindeki Tümlüşik CAN denetleyici gerçekleştirme maliyeti, sistem güvenilirliği, donanım tasarımını basitleştirme ve alan kazanma v.b. gibi birçok avantaj sağlar. Seri Bağlı CAN denetleyici (Serial Linked Input Output, SLIO) CAN veriyolu vasıtasıyla merkezi bir denetleyici tarafından programlanabilen düşük kabiliyetli zeki bir denetleyicidir (Lawrenz, 1997, Özçelik, 2002).

Uygulama alanına göre CAN denetleyiciler Sınıf A, B, C ve D olmak üzere dörde ayrılır. Sınıf A gövde elektronik kontrol uygulamalarını, Sınıf B paylaşılan bilgi tipinin durumunu sorgulayan uygulamaları, Sınıf C gerçek zamanlı kontrol uygulamalarını ve Sınıf D mobil iletişim uygulamalarını destekler (Ekiz, 1997). Bir araç içerisinde tüm uygulama sınıfları ile karşılaşmak mümkündür (BoschAH):

- Elektronik kontrol birimleri arasındaki iletişim (Sınıf C),
- Gövde ve konfor ile ilgili elektronik kontrol birimleri (Sınıf A ve B),
- Mobil iletişim için (Sınıf D).

2.3.9. CAN denetleyici bağlantı şeması

Şekil 2.13’de stand-alone ve tümleşik CAN denetleyici yongaları kullanılarak oluşturulan bir denetleyici alan ağının genel bağlantı şeması görülmektedir. CAN Newsletter (2005)’de farklı üreticilerin piyasaya sundukları stand-alone, tümleşik ve SLIO CAN denetleyici yongaları ve CAN veriyolu sürücü entegreleri karşılaştırmalı olarak verilmektedir.



Şekil 2.13: CAN genel bağlantı şekli

2.3.10. CAN uygulama alanları

CAN, otomotiv endüstrisindeki en bilinen haberleşme sistemidir. Her ne kadar başlangıçta yalnızca otomotiv uygulamaları için tasarlanmış olsa da yüksek performansı ve üstün karakteristik özelliklerinden dolayı birçok dağıtık endüstriyel kontrol uygulamalarında kullanılmaktadır. Başlıca kullanım alanları şunlardır:

- Otomotiv: Bir araç içerisindeki fren, ateşleme, vites kutusu, süspansiyon, havalandırma, ışıklar, merkezi kilit, alarm v.b. aksamaları kontrol eden elektronik devre elemanları denetleyici alan ağı üzerinden haberleşir. Otomotiv endüstrisinin önde gelen bir çok firması (Mercedes, Audi, BMW, Renault) üretimlerinde CAN kullanmaktadır (CANApp, 2006).
- Fabrika otomasyonuna yönelik endüstriyel uygulamalar: CAN kullanılan otomasyon uygulamalarına örnek olarak; PLC kontrollü üretim sistemleri, robot

kontrol sistemleri, tekstil makineleri, paketleme kontrol sistemleri, tarım makineleri, asansör sistemleri, akıllı algılayıcı/eyleyici kontrolü vb. birçok uygulama verilebilir (Ekiz, 1997).

- Tıbbi cihazlarda: X-ray cihazlarında, hasta odalarının izlenmesinde v.b. uygulamalarda kullanılmaktadır.
- Askeri uygulamalarda: Amerika Birleşik Devletleri (A.B.D) “Target Reliance Office”, savaş uçaklarının ve hızlı hücumbotların elektronik modüllerinin CAN temelli olarak haberleşmesini sağlayan ortak sayısal mimari (Common Digital Architecture, CDA) tasarlamıştır (Purdy, 1998). Yine A.B.D.’de Omnitech Robotics firması tarafından gerçekleştirilen CDA benzeri bir çalışmada CAN temelli standartlaştırılmış uzaktan kumanda sistemi (Standardized Teleoperation System, STS) yardımıyla mayın döşenmiş alanlar gibi yüksek risk bulunan bölgelerde insansız araç kullanımı sağlanarak mayın hissetme, nötralizasyon, patlatma, temizleme ve yön tayini işlemleri gerçekleştirilmiştir (Kutlu, 1999).
- Bina otomasyonu: Bina güvenlik sistemleri, çağrı sistemleri, yangın alarm ve söndürme sistemi, ısıtma ve havalandırma sistemleri CAN kullanılarak gerçekleştirilebilir (Kutlu, 1999).

Düşen yonga fiyatları CAN’i oyuncaklar, ses/video cihazlarının kontrolü vb. maliyet duyarlı uygulamalar için elverişli kılmıştır (Ekiz, 1997). CAN seri iletişim protokolü kullanan bir ağ sistemidir. Haberleşme ihtiyacı olan cihazların ve mikrodenetleyicilerin bulunduğu tüm uygulamalarda kullanılmaktadır. Uygulama alanları ve ayrıntıları ile ilgili geniş bilgi CANApp (2006)’da bulunabilir.

2.4. Sonuç

Endüstriyel üretim ortamlarındaki iletişim, üretimin hangi seviyesinde ne tür haberleşme ve kontrol sistemlerinin olacağını belirleyen BBİ mimarisine göre hiyerarşik bir yapıda gerçekleştirilir. BBİ mimarisi birbirleriyle doğrudan ya da dolaylı olarak iletişimde bulunan Algılayıcı–Eyleyici, Saha, Hücre ve Yönetim (en üst seviye) seviyelerinden oluşur.

Saha seviyesindeki iletişim ihtiyalarının karřılanması iin birok sahayolu sistemi kullanılmaktadır. Bu blmde en bařarılı sahayolu sistemlerinden olan ve CERN (European Organization for Nuclear Research) tarafından da nerilen CAN incelenmiřtir. CAN bařta otomotiv olmak zere endstriyel ortamlardaki birok dađıtık kontrol uygulamalarında yaygın olarak kullanılan gerek zamanlı haberleřme sistemidir. Denetleyici alan ađını diđer sahayolu sistemlerinden stn kılan nemli zellikleri yksek hızlı seri arabirim, dřk maliyetli fiziksel ortam, kısa yk byklđ, hızlı tepki zamanları ve yksek hata bulma ve dzeltme seviyesi olarak zetlenebilir.

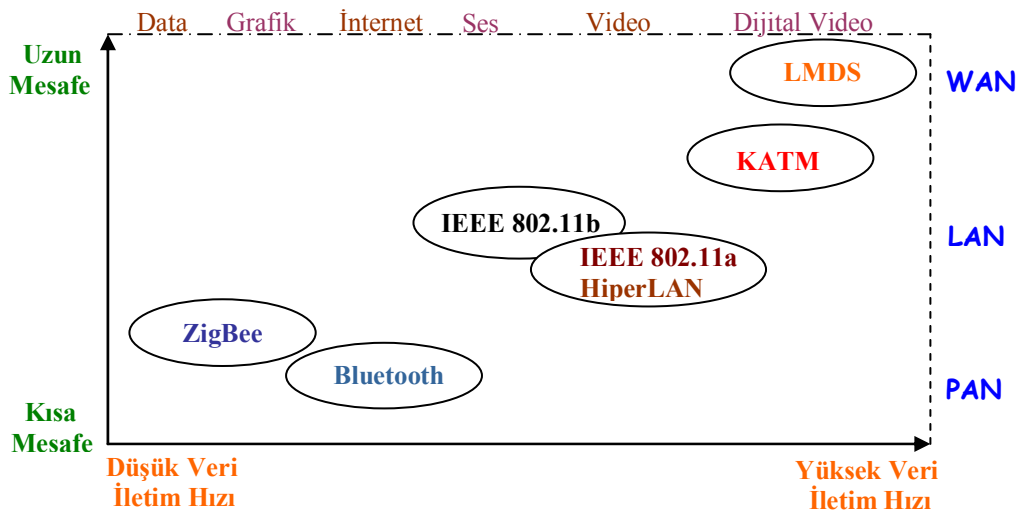
CAN temelli dađıtık sistemlerin, dađıtık alanın byklđ ve uzak CAN segmentlerin birbirleri ile ve diđer yerel alan ađları ile haberleřme ihtiyacı olmak zere iki temel problemi vardır. Tez alıřmasında bu probleme zm olacak Kablosuz Arabađlaşım Birimi (KAB) tasarlanmıřtır. KAB, kablosuz ortam zerinden CAN segmentlerin birbirleri ile ve diđer yerel alan ađları ile haberleřmesini sađlar. CAN ve KLAN ađlar arasında arabađlaşımı sađlayan KAB, her iki ađ eriřim mekanizmasına ve portlarına sahiptir. Bu blmde KAB'ın CAN kısmının tasarımının daha iyi anlaşılabilmesi iin denetleyici alan ađının genel zellikleri ve alıřması zetlenmiřtir.

3. KABLOSUZ YEREL ALAN AĞLARI

3.1. Giriş

Kablosuz ağlar, haberleşmek için radyo frekans (RF) teknolojilerini kullanan kablosuz terminallerin oluşturduğu sistemlerdir. Bunlar kablo kullanan eşleniklerinden farklı olarak kurulum kolaylığı ve ölçeklenebilirlik, hareketlilik, üretkenlik, ileriye yönelik maliyet kazancı, mevcut yerel ağ yapısını genişletme gibi birçok avantajlar sunmaktadır. Bu avantajlarına karşın kablosuz ağlar, kablosuz iletim ortamının doğasından kaynaklanan yüksek bit hata oranı ve sınırlı bant genişliği gibi dezavantajlara da sahiptir (Bayılmış, 2003, Çeken, 2004).

Farklı uygulama ve ihtiyaçları karşılamak üzere birçok kablosuz ağ teknolojisi geliştirilmiş ve geliştirilmektedir. Şekil 3.1’de günümüzde mevcut ve geliştirilmekte olan kablosuz ağ standartlarının, destekledikleri uygulama türlerine, veri iletim hızlarına, kapsama alanlarının büyüklüğüne ve coğrafik ağ yapılarına göre yapılan sınıflandırmaları özetlenmektedir (Bayılmış ve diğ., 2004a).



Şekil 3.1: Kablosuz ağlar ve uygulamadaki yerleri

IEEE 802.11 Kablosuz Yerel Alan Ağları (KLAN), kablolama sınırlamaları olmaksızın Ethernet ve Token Ring gibi geleneksel LAN teknolojilerinin tüm özelliklerini ve yararlarını sağlar. Bu nedenle mevcut yerel alan ağlarının kablosuz ortam üzerinden haberleşen şekli olan kablosuz yerel alan ağları hava üzerinden Ethernet (Ethernet on air) olarak da adlandırılır (Levillain, 2002).

Bu bölümde tez çalışmasında da faydalanılan ve günümüzde birçok uygulamada yaygın olarak tercih edilen IEEE 802.11 Kablosuz Yerel Alan Ağları (Wireless Local Area Network) incelenecektir.

Tez çalışmasının amacı, uzak CAN segmentlerin IEEE 802.11b KLAN üzerinden kablosuz olarak haberleşmesini sağlayacak Kablosuz Arabağlaşım Birimi (KAB) tasarımıdır. KAB'ın KLAN tarafının çalışmasını anlayabilmek için bu bölümde KLAN'ın temel karakteristikleri ve özellikle de ortam erişim mekanizması ayrıntılı olarak incelenecektir.

3.2. Kablosuz Yerel Alan Ağlarının Avantajları

Kablosuz ağların beş önemli avantajı şunlardır:

- Hareketlilik: Kablosuz ağlar, ağ kullanıcılarına kapsama alanı içerisinde hangi noktada olursa olsunlar, hareket halinde dahi gerçek zamanlı bilgi erişimi sağlar.
- Kurulum hızı ve basitliği: Kablosuz ağ sistemlerinin kurulumu hızlı ve kolaydır. İletişim hava üzerinden radyo dalgaları ile sağlandığından LAN'lardaki gibi duvar ve tavanlardan kablo çekme zorunluluğu yoktur.
- Kurulum esnekliği: Kablosuz ağ teknolojisi kablolu LAN'ların erişemeyeceği yerlere ulaşımı sağlar.
- İleriye yönelik maliyet kazancı: Kablosuz ağların ilk kurulum maliyetleri nispeten kablolu bir ağdan daha fazla olmakla birlikte hayat evresi sarfiyatı çok azdır. Uzun vadeli kazançları, çok yer değiştirme gerektiren dinamik ortamlarda kendini belli eder.
- Genişletilebilirlik: Kablosuz iletişim ortamı sayesinde dinamik bir yapıya sahip kablosuz ağlar ile kurulan yapılar kolaylıkla tekrar düzenlenebilir. Aynı zamanda

kurulu kablolu yapıların genişlemesini sağlarlar. En az iki istasyonun bir araya gelmesiyle oluşabileceği gibi erişim noktası kullanarak iletişim kuran istasyon sayısı yüzler hatta binleri bulabilir (Gast, 2002, Nicopolitidis ve diğ., 2003).

3.3. Kablosuz Yerel Alan Ağlarının Kullanım Alanları

Kablosuz yerel alan ağları kablolu ağların kullanıldığı tüm yerlerde kullanılabilir. Aşağıda WLAN'ların kullanım alanlarına birkaç örnek verilmektedir.

- Endüstri: Gerçek zamanlı kontrol, dağıtık kontrol sistemleri, otomasyon sistemleri, veritabanı bağlantısı.
- Ofis ortamı: Video konferans, bilgisayar çevre birimlerinin haberleşmesi.
- Hastane: Uzaktan görüntüleme, medikal görüntüler, hasta takibi.
- Eğitim merkezleri: Bilgi sistemlerine erişim, uzaktan eğitim.
- Taşıtlar: Araç tanıma sistemleri, araç içi kontrol uygulamaları.

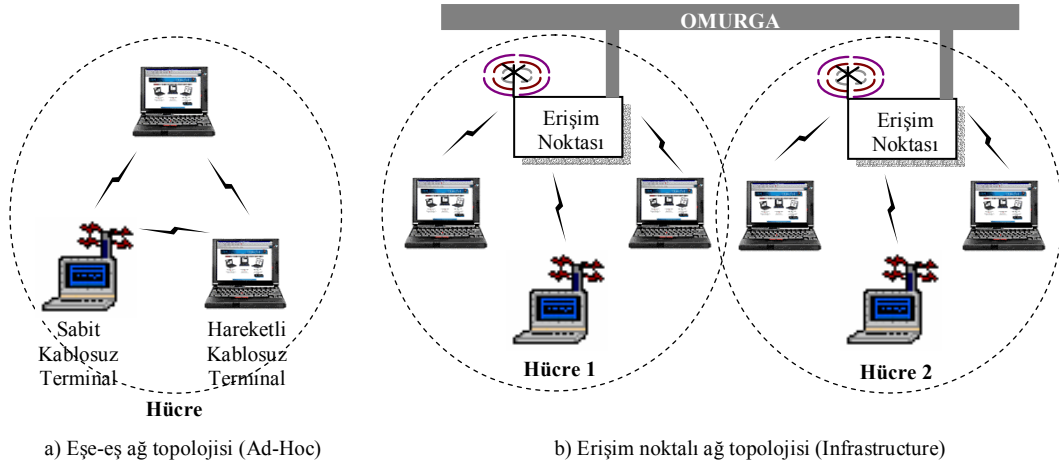
3.4. Kablosuz LAN Ağ Topolojisi

IEEE 802.11 WLAN hücresel mimariye dayalı ağ topolojisini destekler. İletim ortamı hücre olarak adlandırılan küçük alanlara bölünür. Her bir hücre Basic Service Set (BSS) olarak adlandırılır. Kablosuz ağlar, eş-eş ya da birebir ağ topolojisi (Ad-Hoc veya Independent BSS) ve erişim noktalı ağ topolojisi (Infrastructure BSS) olmak üzere iki tür ağ topolojisini desteklemektedir.

Eş-eş ağ, kurulumu hızlı ve kolay olan, geçici bağlantılar sağlamak üzere kurulan bir ağ yapısıdır. Aynı protokolü kullanan en az iki kablosuz terminalin bir araya gelmesi ile oluşur. Herhangi bir erişim noktası olmaksızın tüm kullanıcılar birbirleri ile iletişim kurarlar (Bayılmış ve diğ., 2003, 2004b, Nicopolitidis, 2003).

Erişim noktalı ağlar, yalnızca kablosuz terminallerin kendi aralarında haberleşmesine imkan veren eş-eş ağların genişletilmesini, kurulu kablolu yerel alan ağları ile bütünleşilmesini ve kablolu yerel alan ağları üzerinden sunuculara ulaşılabilmesini

sağlar. Bu yapıda hücre içerisinde iletişim koordinasyonunu sağlayan erişim noktaları (Access Point, AP) kullanılır. AP'ler kablolu ve kablosuz yerel alan ağları arasında köprü görevi gördüğü gibi kablosuz yerel alan ağlarının bant genişliklerini arttırarak daha çok sayıdaki kablosuz terminalin daha uzun mesafeli haberleşmelerini de sağlar (Çeken ve diğ., 2004).



Şekil 3.2: KLAN topolojileri: a) eşe-eş ağ b) erişim noktalı ağ

3.5. IEEE 802.11 Standardı

IEEE 802.11 kablosuz yerel alan ağı standardı Amerikan IEEE Elektrik ve Elektronik Mühendisleri Enstitüsü (The Institute of Electrical and Electronic Engineers) tarafından 1997 yılında geliştirilmiştir. IEEE KLAN standartlarını IEEE 802.11x şeklinde tanımlamış ve bu alanda standartlar geliştirmek üzere bir grup oluşturmuştur.

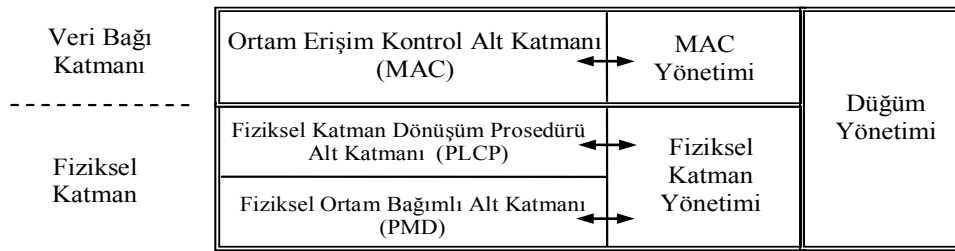
Kablolu yerel alan ağlarındaki Ethernet bağlantılarını kablosuz ortam üzerinden sağlayan IEEE 802.11 standardı, kablosuz yerel alan ağı standartları ailesinin temelini oluşturmaktadır. Zamanla farklı ihtiyaçlar ve farklı veri iletim hızlarını karşılamak için birçok alt standart geliştirilmiştir.

IEEE 802.11 standardı 2,4 GHz lisansız ISM (Industries, Scientific, Medical) bandında FHSS, DSSS ve infrared fiziksel bağlantı seçenekleri ile 2 Mbit/s'e kadar veri iletim hızlarına ulaşabilmektedir (ANSI/IEEE Std 802.11, 1999, Bing, 2000).

Alt bölümlerde IEEE 802.11 standardının protokol mimarisi, ortam erişim mekanizması, çerçeve biçimleri ve alt standartları verilmektedir.

3.5.1. IEEE 802.11 protokol mimarisi

IEEE 802.11 standardı protokol mimarisi OSI referans modelinin Fiziksel ve Veri Bağı katmanlarını kapsar. Şekil 3.3’de IEEE 802.11 standardı temel referans modeli görülmektedir.



Şekil 3.3: IEEE 802.11 temel referans modeli (Bing, 2000)

Fiziksel katman, kablosuz iletişim ortamı ile Ortam Erişim Kontrol (MAC) alt katmanını birbirine bağlayan arayüzdür. Fiziksel katman, Fiziksel Katman Dönüşüm Prosedürü (Physical Layer Convergence Procedure, PLCP) ve Fiziksel Ortam Bağımlı (Physical Medium Dependent, PMD) olmak üzere iki alt katmandan meydana gelmektedir.

PMD alt katmanı, kablosuz ortam karakteristiklerini (DSSS, FHSS veya DFIR) ve kablosuz ortam yoluyla veri iletimi için gerekli metotları (modülasyon, kodlama vb.) tanımlar. PLCP katmanı ise, MAC katmanından gelen paketleri PMD alt katmanı için düzenler. Aynı zamanda MAC katmanı için taşıyıcı sezme ve kanal tahsis (carrier sensing and channel assessment) işlemini gerçekleştirir (ANSI/IEEE Std 802.11, 1999, Bing, 2000).

MAC katmanı, kablosuz ortamın kullanıcılar arasında etkin olarak paylaşılmasını yani kullanıcıların ortama erişim mekanizmasını tanımlar. Bunun yanı sıra veri paketlerinin parçalanması (fragmentation), hata iyileştirme, hareketlilik yönetimi, güç koruma ve şifreleme gibi işlemleri de gerçekleştirir. MAC tüm fiziksel katman

türleri (DSSS, FHSS, DFIR) için ortak olmakla birlikte veri iletim hızları farklılık gösterir.

Fiziksel katman yönetimi, farklı bağlantı şartlarının uyarlanması ile ilgilenirken, MAC yönetimi ise senkronizasyon, güç yönetimi, birliktelik (association) ve tekrar birliktelik konuları ile ilgilenir. İstasyon yönetimi de fiziksel ve MAC yönetim katmanlarının etkileşiminden sorumludur (Bing, 2000).

3.5.2. Çerçeve formatları

KLAN ortam erişim katmanında farklı amaçlar için kullanılan üç temel MAC çerçeve biçimi (MAC Protocol Data Unit, MPDU) vardır. Bunlar:

- Veri çerçeveleri,
- Kontrol çerçeveleri (RTS, CTS, ACK),
- Yönetim çerçeveleri (işaret).

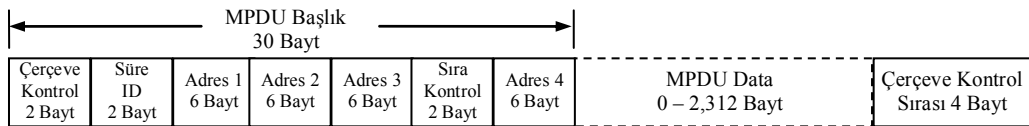
MAC veri çerçeve biçimi: Veri ve yönetim çerçeveleri için kullanılır. Şekil 3.4'de genel MAC çerçeve biçimi görülmektedir. MPDU başlık kısmındaki bölümler ve işlevleri şunlardır:

Çerçeve Kontrol: Dağıtık sisteme gönderilen/alınan paketlerin kontrolü, güç yönetimi, paket ayırma, şifreleme, kimlik belirleme (authentication).

Süre ID: Tahsis edilen vektörün süresi, güç koruma modunda çalışan düğümün tanımlanması.

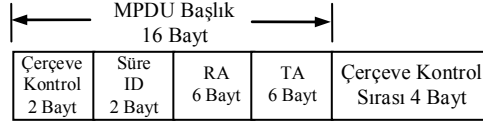
Adres 1-4: BSS ID, hedef, kaynak, verici/alıcı için adresler

Sıra Kontrol: Paket ve paket parçaları için sıra numarası



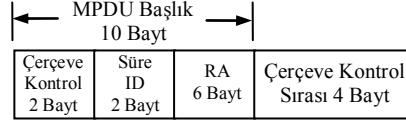
Şekil 3.4: MAC (MPDU) genel çerçeve biçimi (ANSI/IEEE Std 802.11 1999)

RTS (Request To Send) kontrol çerçeve biçimi: Süre alanında bir sonraki veri/yönetim çerçevesinin iletimi için gerekli süre tanımlanmaktadır (Şekil 3.5). RA, bir sonraki veri/yönetim çerçevesini alacak istasyonun adresini içerirken, TA, RTS çerçevesini gönderen istasyonun adresini içermektedir.



Şekil 3.5: RTS çerçeve biçimi (ANSI/IEEE Std 802.11 1999)

CTS (Clear To Send) kontrol çerçeve biçimi: CTS, RTS çerçevesine yanıt olarak gönderilir (Şekil 3.6). RA, alanına alınan RTS çerçevesindeki TA alanı adres bilgisi yüklenir. Süre alanına ise RTS çerçevesindeki süre alanındaki değerden CTS göndermek için gerekli zaman ve SIFS (Short Inter Frame Space) değerlerinin çıkarılması sonucu kalan değer yüklenir.

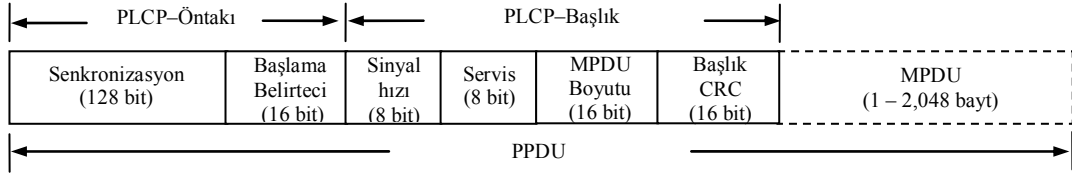


Şekil 3.6: CTS çerçeve biçimi (ANSI/IEEE Std 802.11 1999)

ACK kontrol çerçeve biçimi: ACK çerçeve CTS çerçevesi ile aynı biçimdedir. RA alanına hedef istasyonun adresi, Süre alanına da alınan çerçevenin süre alanındaki değerden ACK çerçeve göndermek için gerekli zaman ve onun çerçeve iletim boşluğu (SIFS) süresi çıkartılarak kalan değer yüklenir.

MAC katmanında oluşturulan çerçeveler iletilmek üzere fiziksel katmana gönderilir. Kullanılan fiziksel katmana (DSSS, FHSS, kızılötesi vb.) göre MAC çerçevesine bazı ilaveler yapılır. Şekil 3.7’de görülen IEEE 802.11b DSSS Fiziksel Katman Dönüşüm Prosedürü (PLCP) çerçeve biçimi; PLCP öntakısı (Preamble), PLCP başlık (header) ve MAC çerçevesinden oluşmaktadır. Öntakı alanı senkronizasyon, kanal tahsisi ve çerçeve zamanlaması için gerekli başlangıç alanını içerir. Başlık alanı ise kullanılan modülasyon tekniğini (DBPSK, DQPSK vb.), veri iletim hızı bilgisini,

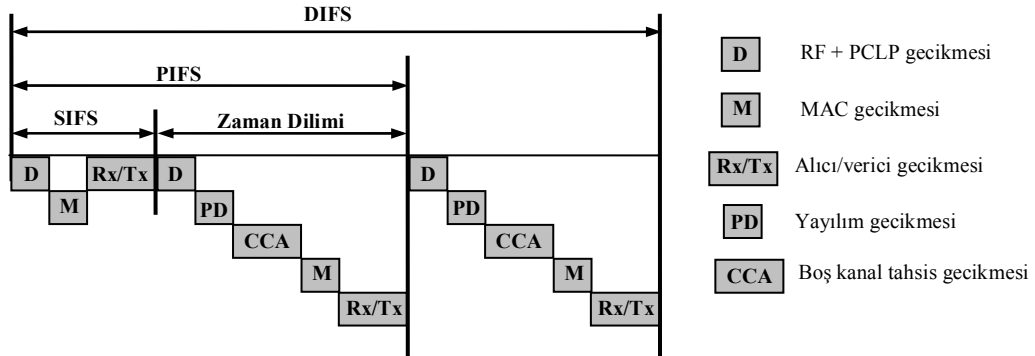
gönderilen MAC çerçevesinin boyutu ve başlık alanındaki hata kontrolü (CRC) için gerekli alanları içerir.



Şekil 3.7: IEEE 802.11b DSSS PLCP çerçeve biçimi (ANSI/IEEE Std 802.11 1999)

3.5.3. Çerçeveler arası boşluk (inter frame space, IFS)

Çerçeveler arasındaki zaman aralıkları, çerçeveler arası boşluk olarak adlandırılır ve kablosuz veri iletim hızlarından bağımsızdır. IFS her bir fiziksel katman için sabittir. IEEE 802.11 standardında MAC protokolünün ortama erişimi belirlemede çerçeveler arasındaki boşluk çok önemlidir. Çünkü çerçeveler arası boşluklar, ortama erişimi belirleyen Backoff algoritmasının çalışma süresini etkilemektedir (ANSI/IEEE Std 802.11, 1999, Bing, 2000). IEEE 802.11 ortama erişim için farklı öncelikler sağlamak amacıyla dört farklı çerçeveler arası boşluk tanımlar (Şekil 3.8).



Şekil 3.8: Çerçeveler arası boşluk tanımlamaları (Bing 2000)

- Zaman dilimi (Slot time): Backoff algoritmasında her zaman diliminde ortamın meşgul olup olmadığı kontrol edilir.
- En kısa çerçeveler arası boşluk (Short IFS, SIFS): Acil yanıt gönderiminde (ACK, RTS, CTS çerçevelerinin) kullanılır. SIFS, kullanılan fiziksel katmana bağlı olarak sabit bir değerdir. Ortam erişimini kazanmış bir istasyon SIFS aralıklara yüksek öncelikli olarak iletimini gerçekleştirir.

- Nokta eşgüdüm fonksiyon çerçeveler arası boşluk (Point Coordination Function IFS, PIFS): PCF erişim mekanizmasında ortam erişimini kazanmak için kullanılır. PIFS, SIFS ile zaman dilimi sürelerinin toplamına eşittir.
- Dağıtık eşgüdüm fonksiyon çerçeveler arası boşluk (Distributed Coordination Function IFS, DIFS): Ardışık veri paketleri arasındaki minimum gecikmedir. Ortamın boş olduğundan kesinlikle emin olmak için istasyonlar DIFS süresi boyunca erişimlerini ertelerler. DIFS, PIFS ile zaman dilimi sürelerinin toplamına eşittir.
- Genişletilmiş IFS (Extended IFS, EIFS): En uzun çerçeveler arası boşluktur. Hatalı paket alan istasyon tarafından kullanılır.

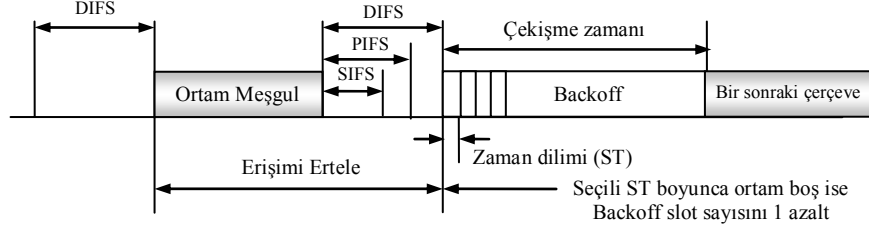
Örneğin DSSS kullanılan bir sistemde SIFS = 10 μ s, zaman dilimi ise 20 μ s'dir. FHSS kullanılan bir sistemde ise SIFS = 28 μ s, zaman dilimi ise 50 μ s'dir (ANSI/IEEE Std 802.11, 1999).

3.5.4. Ortam erişim mekanizması

Ortam erişim mekanizmaları, sınırlı bant genişliğine sahip kablosuz iletim ortamını kullanıcılar arasında etkin olarak paylaşdırmayı sađlayan kurallar bütünüdür. IEEE 802.11 MAC katmanı, ortam erişimini sađlamak üzere çekişme esaslı dağıtık eşgüdüm fonksiyonu (Distributed Coordination Function, DCF) ve çekişmeden bağımsız nokta eşgüdüm fonksiyonu (Point Coordination Function, PCF) olmak üzere iki farklı erişim mekanizması kullanır. Her iki mekanizma ayrı ayrı kullanılabilirdiđi gibi birlikte de kullanılabilir (Aad, 2003, ANSI/IEEE Std 802.11, 1999, Bing, 2000).

Şekil 3.9'da IEEE 802.11 ortam erişim mekanizmasının genel çalışma prensibi görölmektedir. PIFS yalnızca PCF erişim yöntemi ile çalışan düğümlerde kullanılır. PCF erişim noktası kullanılan sistemlerde geçerlidir. Bu yöntemde çekişmeden bağımsız olarak ortam boş olduđu sürece düğüm PIFS aralıkları ile çerçeve iletimini gerçekleştirir. DIFS ise DCF erişim mekanizmasını kullanan düğümlerde çerçeve iletimi arasındaki minimum süredir (Aad, 2003, ANSI/IEEE Std 802.11, 1999, Bing, 2000).

Ortam DIFS'ten daha büyük bir süre boş ise erişim başlar



Şekil 3.9: IEEE 802.11 ortam erişim mekanizmasının genel çalışması (ANSI/IEEE Std 802.11, 1999)

3.5.4.1. Dağıtık eşgüdüm fonksiyonu (DCF)

802.11 standardında temel ortam erişim yöntemi daha çok CSMA/CA olarak bilinen dağıtık eş güdüm fonksiyonudur (DCF). CSMA/CA Ethernette kullanılan CSMA/CD (Carrier Sense Multiple Access with Collision Detection) ile çok benzerdir. Her iki protokolda de, iletişim ortamının kullanımını belirlemek için taşıyıcı sezme (carrier sensing) ve ortamdaki çekişmeyi çözmek için üssel geri çekilme (Exponential Backoff) algoritması kullanılır (Aad, 2003, ANSI/IEEE Std 802.11, 1999, Bing, 2000).

CSMA/CA protokolü dağıtık kontrolü kullandığından KLAN içerisindeki erişimi düzenleyecek merkezi bir denetleyici yoktur. Bundan dolayı düğümler protokol kurallarına uydukları sürece istedikleri zamanda veri iletiminde bulunabilirler.

DCF'nin çalışmasını daha iyi anlayabilmek için bu bölümde, DCF'yi oluşturan CSMA, çarpışmadan kaçınma ve Backoff algoritmasının işleyişi, hata sezme ve sanal taşıyıcı sezme işlemleri alt bölümler olarak verilmektedir.

3.5.4.1.1. Taşıyıcı duyarlı çoklu erişim (CSMA)

CSMA birçok istasyonun eş zamanlı iletim isteğinden doğan çakışmaları en aza indirgeyen (fakat yok edemeyen) bir "konuşmadan önce dinle" (listen before you talk) metodudur. Bir düğüm veri iletiminde bulunmak için önce, ortamda devam eden bir iletim olup olmadığını anlamak için kablosuz ortamı dinler. Eğer ortam meşgul ise (devam eden bir iletim var ise) düğüm iletimini daha sonraki bir zamana

erteler. Eđer ortamın DIFS'den daha büyük bir süre için boş olduđu algılanırsa ivedi olarak paket gönderilir.

MAC katmanı ortamın durumuna (boş olup olmadığına) karar vermek için fiziksel katmanla birlikte çalışır. Alıcı vasıtasıyla alınan sinyalin gücü belirli bir eşik değerin altında ise ortamın boş olduđu belirlenir.

CSMA düşük trafik yoğunluđu olan ortamlarda oldukça etkilidir. Protokol minimum gecikme ile düğümlere iletim için izin verir. İletim ortamı boyunca sonlu yayılım gecikmesinden dolayı muhtemelen iki ya da daha fazla düğüm, eşzamanlı ortamı boş olarak algılayacak ve bu sebeple bir çarpışma meydana gelecektir. Çarpışmalar yüksek trafik yoğunluklu ortamlarda daha sık olacaktır. Paket iletim zamanı üzerinden zaman diliminin (slot time) oranı CSMA'in performansını etkiler.

Kablolu LAN'da çarpışmayı çözmek için çarpışma sezme (Collision Detection, CD) iyi bir yöntemdir. Ancak kablosuz LAN'da çarpışma sezme iki sebepten dolayı kullanılamaz. Birincisi çarpışma sezme eşzamanlı çift yönlü radyo (full duplex radio) gerektirir. Bu çok büyük bir maliyet getirir ve bunu yakın frekans bantlarında uygulamak oldukça zordur. İkincisi, kablosuz ortamın sınırlı kapsama alanı probleminde dolayı, tüm kablosuz terminaller birbirlerini dinleyemezler. Bu durumda aynı anda iki istasyon veri göndermek isteyebilir. Bunun yerine IEEE 802.11 çarpışmayı engellemek için çarpışmadan kaçınma (collision avoidance) yöntemini kullanır.

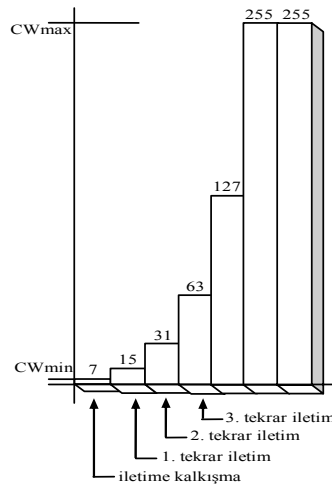
3.5.4.1.2. Çarpışmadan kaçınma (CA) ve backoff prosedürü

IEEE 802.11 standardında CSMA protokolü, ardışık paket iletimi arasında rasgele çerçeveler arası boşluk (Backoff) olarak tanımlanan çarpışmadan kaçınma şeması ile birlikte kullanılır. Çarpışmadan kaçınma bir paketin başarılı iletiminden sonra ivedi olarak yüksek çarpışma olasılığını azaltmak için çalışır. Çarpışmadan kaçınma, iletim gerçekleştiren toplam düğüm miktarını her biri farklı zaman dilimi (backoff time slot) kullanan daha küçük gruplara ayırmaya çaba gösterir. Backoff zamanı, çerçeveler arası boşlukta tanımlanan zaman diliminin (slot time) rasgele katıdır.

$$\text{Backoff zamanı} = \text{Rasgele bir deęer } ([0, CW]) \times \text{Zaman dilimi} \quad (3.1)$$

Rasgele deęer aslında sözde rasgeledir. Çünkü bu deęer kullanılan fiziksel katmanın karakteristiklerine baęlı olarak 0 ila CW (Collision Window) (CW_{min} – CW_{max}) deęeri arasındadır. Bununla birlikte zaman dilimi de fiziksel katman karakteristiklerine baęlıdır.

CW deęeri boyunca aęda paketler arasında çarpışma meydana gelebilir. Şekil 3.10'da çarpışmadan sonra her iletimize kalkışmada CW'nin alabileceęi deęer artışına bir örnek verilmektedir. Her zaman dilimi boyunca ortam boş ise backoff aralık deęeri (zaman dilimi sayısı) 1 azaltılır. Başarılı iletimden sonra ise CW, CW_{min} deęerine kurulur.

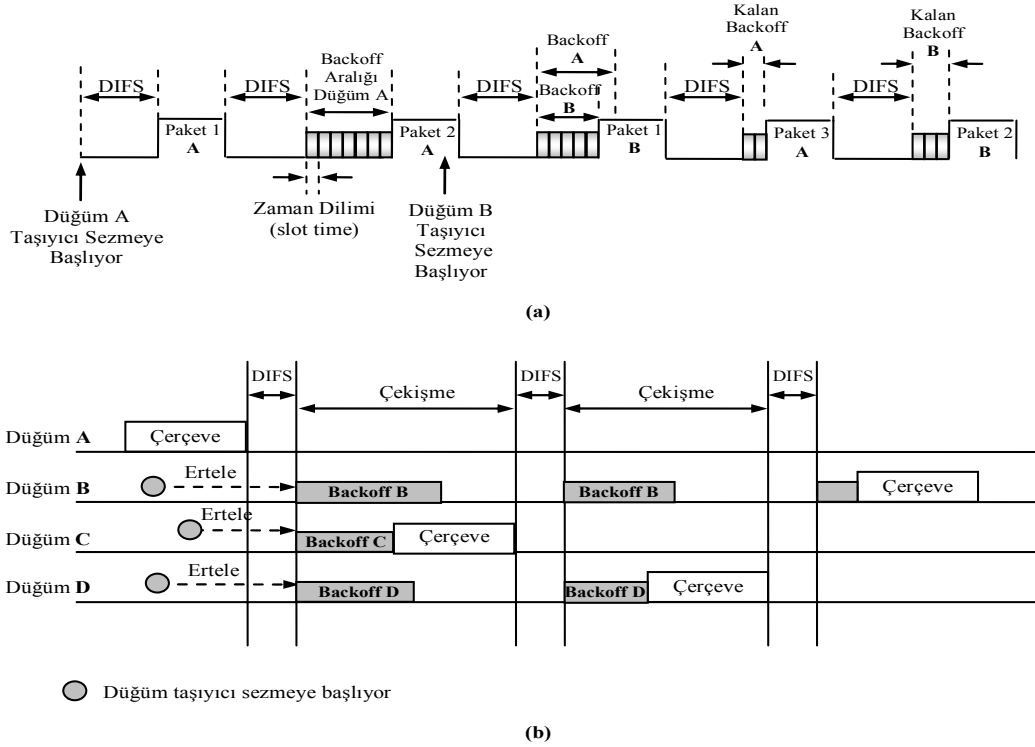


Şekil 3.10: CW'nin üssel artımına bir örnek (ANSI/IEEE Std 802.11, 1999)

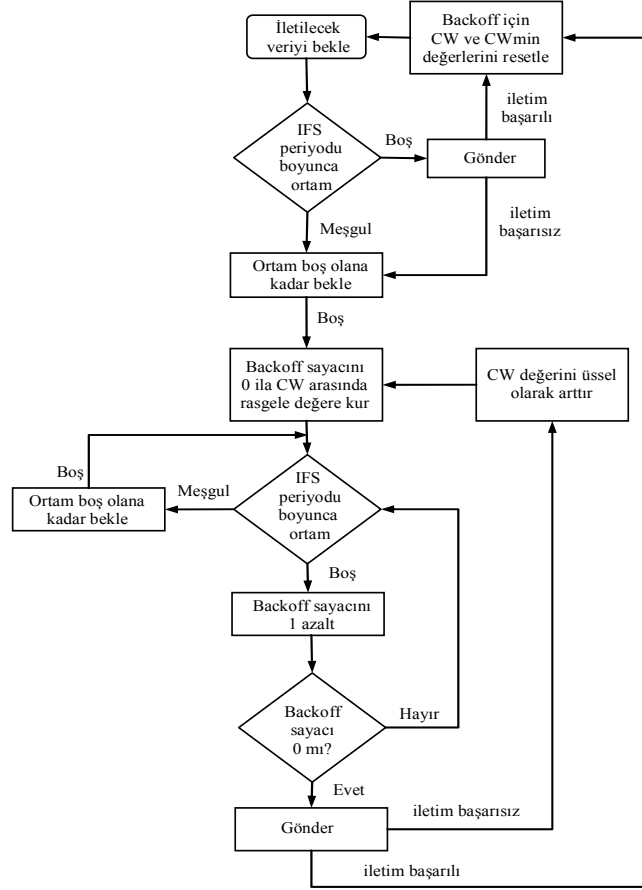
Eęer ortam meşgul olarak algılanırsa, düęüm ilk olarak DIFS'in sonuna kadar, ardından iletimize kalkışmadan önce zaman dilimlerinin rasgele miktarı (Backoff aralığı) kadar beklemelidir. Tekrar iletimize gerektiğinde belirli bir eşik deęerine kadar Backoff aralığı üssel olarak arttırılır. Paketler başarılı olarak iletildiğinde ise Backoff aralığı minimum (CW_{min}) bir deęere azaltılır.

Backoff aralığındaki her bir zaman diliminde taşıyıcı algılama gerçekleştirilir. Ortam zaman dilimi boyunca boş ise Backoff aralığı (sayacı) 1 zaman dilimi azaltılır. Eęer ortam meşgul olarak algılanırsa Backoff prosedürü askıya alınır (belli bir süre için

durdurulur) ve Backoff aralığı zaman dilimi sayısı olarak azaltılmaz (Backoff sayacı değeri değişmez). Bu durumda ortam DIFS'den daha büyük bir süre tekrar boş olursa Backoff prosedürü daha önce bırakılan zaman dilimi sayısından (kaldığı değerden) azaltılmaya devam eder. Diğer bir ifadeyle seçili Backoff aralığı öncekinden daha azdır. Bu süreç Backoff aralığı sıfır ('0') olana ve paket gönderilene kadar tekrar eder. Şekil 3.11'de Backoff prosedürünün çalışması görülmektedir. Ayrıca DCF erişim mekanizmasının genel çalışma süreci Şekil 3.12'de sunulan akış diyagramında özetlenmektedir.



Şekil 3.11: CSMA/CA kullanarak çoklu paket gönderimi



Şekil 3.12: DCF ortam erişim mekanizmasının genel çalışma akış diyagramı

3.5.4.1.3. Hata sezme

802.11 standardında bir paketin doğru olarak iletilip iletilmediği, ACK alındı paketlerinin gönderimi ile belirlenir. Bir paket doğru olarak alındığında vericiye bir ACK gönderilir. ACK SIFS'den sonra gönderilir. SIFS, DIFS'den küçük olduğundan herhangi yeni bir paketin gönderim zamanından önce alındı bilgisi gönderilmiş olur. ACK gelmez ise gönderici, paketin bozulduğunu (hata oluştuğunu) varsayar ve tekrar gönderir. Tekrar gönderme işleminin daha üst katmanlar yerine MAC katmanı tarafından gerçekleştirilmesi, kaybedilen mesajların hızla iyileştirilmesine olanak sağlar (Bing, 2000).

ACK her ne kadar güvenli paket iletimi için kullanılsa da yayın (broadcast) modunda yahut çoklu gönderim durumunda çok sayıda ACK gönderimi, çarpışma meydana getireceğinden pratik bir yöntem değildir.

3.5.4.1.4. Sanal taşıyıcı algılama (virtual carrier sense)

CSMA/CA'de taşıyıcı algılama mekanizması, MAC katmanı tarafından gerçekleştirilen sanal algılamaya ve fiziksel katman tarafından gerçekleştirilen düğümün vericisinin durumuna göre ortamın boş ya da meşgul olduğuna karar verir.

Sanal taşıyıcı algılama, kablosuz ortamın kullanılmaya yakın olduğunu bildiren rezervasyon bilgisinin kullanıcılar arasında dağıtılmasına dayanır. Bunun için gönderme istek paketi (RTS) ve göndermek için boş paketi (CTS) olmak üzere iki küçük boyutlu paket kullanılır (Aad, 2003, ANSI/IEEE Std 802.11, 1999, Bing, 2000).

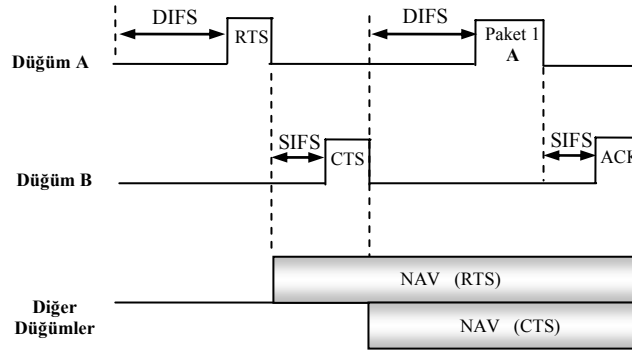
RTS, gönderen (kaynak) düğüm tarafından dağıtılırken, CTS, iletim izni isteyen düğümüne onay vermek için hedef düğüm tarafından dağıtılır. RTS ve CTS paketleri içerisinde bir süre tanımlama (Duration ID) alanı vardır. RTS/CTS çerçeveleri içindeki süre tanımlama alanı, bir sonraki veri çerçevesi ve ACK'nın toplam iletim süresini belirtir. Bu süre boyunca ortam, karşılıklı olarak RTS/CTS paketi gönderen düğümlerin veri paketlerinin veya ACK paketlerinin iletimi için ayrılır.

Kapsama alanı içerisinde veri iletiminde bulunacak olan (RTS göndermiş) ve iletimi kabul eden (CTS göndermiş) düğümlerin dışındaki diğer tüm düğümler, bu paketler içerisindeki (RTS ve CTS) süre tanımlama bilgisini çözerek ağ tahsis vektöründeki (network allocation vector, NAV) ortam rezervasyon bilgisini öğrenirler. NAV bilgisi sıfır ('0') değilse veya ortam meşgul ise düğümler iletim isteklerini erteleyecektir. Böylelikle çarpışmadan kaçınılmış olunur.

Bir başka deyişle, RTS çerçevesini alan tüm düğümler ilk olarak hedef düğüm bilgisini kontrol ederler. RTS ile adreslenen düğüm, RTS çerçevesi içerisindeki NAV bilgisi ortamın boş olduğunu gösterirse SIFS periyodundan sonra bir CTS çerçevesi gönderir. Eğer NAV bilgisi ortamın meşgul olduğunu gösterirse, düğüm RTS'ye cevap vermez. CTS'nin kaynak düğümüne gönderilme süresi vardır. Eğer bu süre içerisinde RTS'yi gönderen düğümüne bir yanıt (CTS) verilmez ise, RTS'nin başarısız olduğu kabul edilir. Bu durumda süreç tekrarlanır.

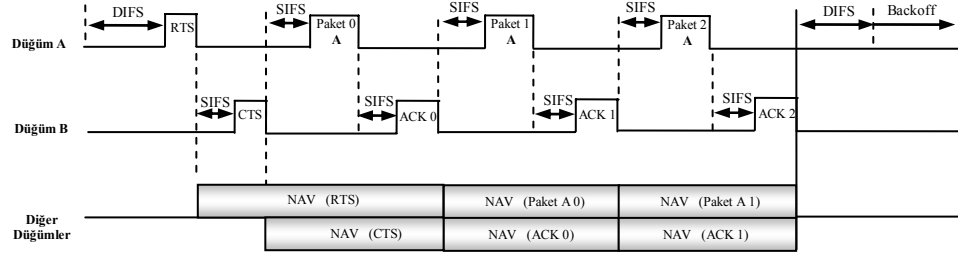
Yukarıda anlatılan sanal taşıyıcı algılama mekanizması MAC tarafından sağlanabilecektir. Aslında bu mekanizma NAV olarak da ifade edilebilir. NAV, RTS/CTS paketleri içerisinde süre bilgisine bağlı olarak ortamın gelecekteki trafik durumunu tahmin etmektir. Süre bilgisi (Süre ID) MPDU paketinin başlık bilgisi içerisinde tanımlanmaktadır. NAV aslında belli bir değerden (süre bilgisi) sıfıra ('0') doğru sayan bir sayıcı olarak düşünülebilir. NAV sıfır ise ortam boş, değil ise meşguldür. NAV süresince düğüm iletim girişimini erteleyecektir.

RTS/CTS mekanizmasının kullanılıp kullanılmayacağı, RTS eşik değeri özelliği ile belirlenir. RTS eşik değerinden daha büyük çerçeve boyutlarında, RTS/CTS mekanizması kullanılır. RTS eşik değeri her düğüm için ayrıdır.



Şekil 3.13: Sanal algılama kullanılarak paket iletimi

Eğer iki düğüm arasında ard arda veri paketleri gönderilecek ise tek bir defa hedef ve kaynak arasında RTS/CTS mekanizmasını çalıştırılması yeterlidir. RTS ve CTS'deki süre bilgisi ACK0'nun sonuna kadar ortamın meşgul olduğunu belirtmek için NAV'ı günceller. İlk paketin iletiminden sonra kaynak ve hedef düğümler arasında iletişimin devam edip etmeyeceğini RTS ve CTS'den sonra gönderilen paket (Paket A0) ve hedef tarafından paketin alındığını belirten ACK çerçeveleri içerisindeki süre tanımlama alanlarındaki değer belirler. Bir başka ifade ile veri paketi ve ACK bilgisi sanal RTS/CTS gibi davranır. Veri paketi ve ACK içerisindeki süre bilgisi '0' oluncaya kadar hedef ve kaynak arasındaki iletişim devam eder. Eğer süre bilgisi '0' ise artık son ACK bilgisinden sonra ortam erişimini kazanmak için tüm düğümler çekişme sürecine girerler (Aad, 2003, ANSI/IEEE Std 802.11, 1999, Bing, 2000).



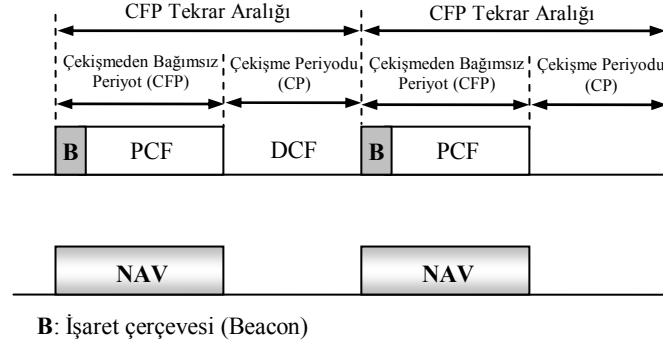
Şekil 3.14: Tek bir RTS/CTS kullanılarak ard arda paket gönderimi

3.5.4.2. Nokta eşgüdüm fonksiyonu (PCF)

Gerçek zamanlı uygulamalar, verinin iletim gecikmesine duyarlı trafiklerdir. Ancak CSMA/CA gerçek zamanlı trafik için uygun değildir. Bunun nedeni ise CSMA/CA ses, veri ve video gibi farklı trafik karakteristiklerine sahip uygulamaların servis kalitesi ihtiyaçlarını dikkate almaz (Aad, 2003, Bing, 2000).

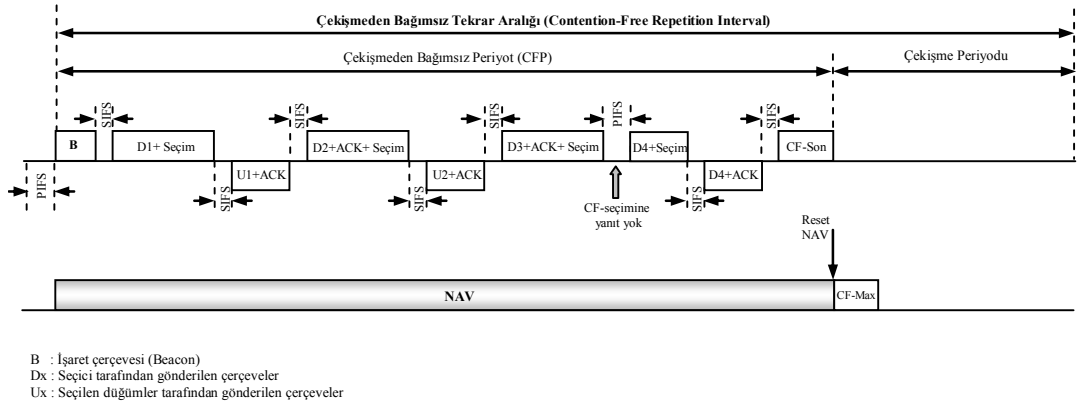
Çarpışma olasılığı, rasgele Backoff aralığı kullanımı ve uzun paketlerin iletimi aşırı gecikme değişimine sebep olabilir. Bununla birlikte, hata bulma için kullanılan ACK bilgisi ve kayıpları iyileştirmek için kullanılan tekrar iletim isteği, iletim gecikmesini arttıran diğer nedenlerdir.

PCF, gecikmeye duyarlı trafiklerin iletimi için kullanılmaktadır. IEEE 802.11 MAC, PCF yöntemini yalnızca erişim noktası kullanılan altyapılı topolojilerde (infrastructure network) kullanabilir. PCF genel olarak merkezileştirilmiş, çekişmeden bağımsız (contention-free) çoklu erişim şemasıdır. PCF merkezi düğüm olan erişim noktası üzerinde çalıştırılır ve merkezi erişim noktası kapsama alanındaki hangi düğümün iletimde bulunabileceğine karar verir. Seçilen düğümler çekişme olmaksızın ortama erişirler ve verilerini iletirler. Çekişmeden bağımsız periyot (Contention-Free Period, CFP), bir işaret (Beacon) çerçevesinin gönderimi ile başlar. İşaret çerçevesi, CFP'nin süresini bildirir. Buna göre tüm düğümler NAV bilgilerini güncellerler. CFP periyodunun sonlandırılması CFP-End çerçevesinin gönderimi ile gerçekleşir. CFP periyodu, PCF sanal taşıyıcı algılama mekanizmasını kullanır. PCF, Şekil 3.15'de görüldüğü gibi çekişmeden bağımsız ve çekişme periyotlarının değişiminden oluşur.



Şekil 3.15: PCF erişimi

Şekil 3.16’da PCF ortam erişim mekanizmasında çerçeve transfer işlemi görülmektedir.



Şekil 3.16: PCF erişim mekanizmasında çerçeve transferi

3.5.5. IEEE 802.11 alt standartları

IEEE 802.11x ailesinin temelini IEEE 802.11 standardı oluşturmaktadır. Bu standart 2,4 GHz lisansız ISM bandında FHSS, DSSS ve kızıllötesi uygulama seçenekleri ile 2 Mbit/s’e kadar veri iletim hızlarını destekleyebilmektedir. Gelişen teknoloji ile birlikte farklı ihtiyaçları karşılamak üzere farklı iletim hızları ve farklı fiziksel katman seçenekleri ile IEEE 802.11 standardını temel alan alt standartlar geliştirilmiştir. IEEE tarafından geliştirilen bu standartlar ANSI ve ISO tarafından da kabul edilmiştir. Bu alt standartların en yaygınları IEEE 802.11a, IEEE 802.11b ve IEEE 802.11g’dir.

3.5.5.1. IEEE 802.11a standardı

IEEE 802.11 ailesi içerisinde yeni nesil kablosuz LAN standardıdır denilebilir. 2,4 GHz'deki band genişliğini kullanan değişik uygulamalara, 5 GHz'lik frekans bandını tanımlayarak alternatif oluşturmuştur. 5,15-5,25 GHz, 5,25-5,35 GHz, 5,725-5,825 GHz frekansları arasında 300 MHz'lik bir frekans bandında çalışır. IEEE 802.11a standardı 5 GHz lisanssız U-NII (Unlicensed National Information Infrastructure) bandda OFDM modülasyonunu kullanarak veri iletim hızını kanal (üst üste binmeyen 8 kanal kullanır) başına 54 Mbit/s'e kadar çıkarmıştır. 6 Mbit/s, 9 Mbit/s, 12 Mbit/s, 18 Mbit/s, 24 Mbit/s, 36 Mbit/s, 48 Mbit/s ve 54 Mbit/s veri iletim hızlarını destekleyen bu standart, çoklu ortam uygulamaları ile veri aktarımının yoğun olduğu uygulamalar için daha uygun olacaktır (Bayılmış ve diğ. 2003, Gast, 2005).

DSSS yerine OFDM tekniğinin kullanılması daha iyi performans ve daha geniş kapsama alanı sunmakla birlikte daha fazla güç harcaması gerektirir. IEEE 802.11a HiperLAN2 standardına rakip olarak geliştirilmiştir.

3.5.5.2. IEEE 802.11b standardı

En yaygın ve kabul gören standarttır. 802.11b standardı 2,4 GHz ISM bandında çalışır ve modülasyon tekniği olarak yalnızca DSSS kullanır. 1 Mbit/s, 2 Mbit/s, 5,5 Mbit/s ve 11 Mbit/s veri iletim hızlarını destekler. Kablosuz yerel alan ağlarının 2,4 GHz ISM bandını mikrodalga fırın ve Bluetooth gibi ürünler ile paylaşması, olası parazitlerden dolayı kayıplara ve veri iletim hızlarının düşmesine neden olabilmektedir.

Farklı firmaların 802.11b ürünleri arasındaki birlikte-çalışabilirliğin bugün WiFi Alliance olarak bilinen WECA (Wireless Ethernet Company Alliance) tarafından onaylanması ile IEEE 802.11b bir endüstri standardı haline gelmiştir. Bu kurumun amacı WiFi ürünlerinin işlevliliğini sertifikalandırmak ve IEEE 802.11b'yi global bir standart yapmaktır (Levillain, 2002).

3.5.5.3. IEEE 802.11g standardı

Bu standardın kullanımındaki amaç, geçerli IEEE 802.11b standardı üzerinden veri iletim hız artırımını sağlamaktır. 802.11b’de olduğu gibi 2,4 GHz bandı kullanılarak 54 Mbit/s’lik veri iletim hızı sağlar. OFDM ve CCK (Complementary Code Keying) modülasyon tekniklerinin her ikisini de destekler. Günümüzde 802.11b’nin yerini almak üzeredir.

Anılan IEEE 802.11 standartları ile bu standart ailesi üzerine yapılan diğer çalışmalar Tablo 3.1’de özetlenmektedir.

Tablo 3.1: IEEE 802.11 standart ailesi

| Standart | Özellikleri |
|---|---|
| IEEE 802.11 | Orijinal WLAN standardı. 1-2 Mbit/s veri iletim hızlarını destekler. |
| IEEE 802.11a | 5 GHz U-NII bandında çalışan yüksek hızlı WLAN standardı. Kanal başına 54 Mbit/s veri iletim hızını desteklemektedir. |
| IEEE 802.11b | 2,4 GHz ISM bandında 11 Mbit/s veri iletim hızını desteklemektedir. |
| IEEE 802.11e | IEEE WLAN yapıları için QoS’ini arttırmak ve yönetmek |
| IEEE 802.11f | AP’ler arasında haberleşme protokolüdür (Inter Access Point Protocol, IAPP). |
| IEEE 802.11g | 802.11b standardı üzerine kurulan bu standart 2,4 GHz’de 54 Mbit/s veri iletim hızına ulaşabilmektedir. |
| IEEE 802.11h | IEEE 802.11a için dinamik kanal seçimi ve iletim gücü kontrolü sağlar. |
| IEEE 802.11i | IEEE 802.1X ile kombine güvenlik özellikleri sunmaktadır. |
| IEEE 802.11n | 2007’nin ortalarında standartlaştırma çalışmalarının tamamlanması beklenmektedir. Kablosuz yerel alan ağları içerisinde en yüksek veri iletim hızını (540 Mbit/s) ve çalışma mesafesini (kapalı ortamda 50m) desteklemesi planlanıyor. 802.11n, diğer 802.11 standartlarına MIMO (multiple-input multiple-output) eklenilerek geliştirilmektedir. |
| IEEE 802.1X | IEEE ağları için güvenlik çerçeve standardı. |
| WISPR (Wireless ISP Roaming) | Kablosuz Ethernet Uyumluluğu Topluluğu tarafından geliştirilen, kablosuz kamusal ağlar arasında dolaşım için tavsiyeler bütünüdür. |

3.6. Sonuç

Kablosuz iletişim ortamının sınırlamalarına rağmen, kablosuz yerel alan ağlarının kullanımı; kurulum kolaylığı ve basitliği, esnekliği, ileriye yönelik maliyet kazancı, hareketlilik ve mevcut yerel alan ağ yapısını genişletme gibi avantajlarından dolayı gün geçtikçe artmaktadır.

Günümüzde mevcut ve geliştirilmekte olan birçok kablosuz iletişim teknolojisi olmasına rağmen kablosuz Ethernet olarak adlandırılan IEEE 802.11 standardının en büyük avantajı, oldukça yaygın bir kullanım oranına sahip (%95 civarında) standart kablolu Ethernet yapısı ile sağladığı kolay entegrasyondur.

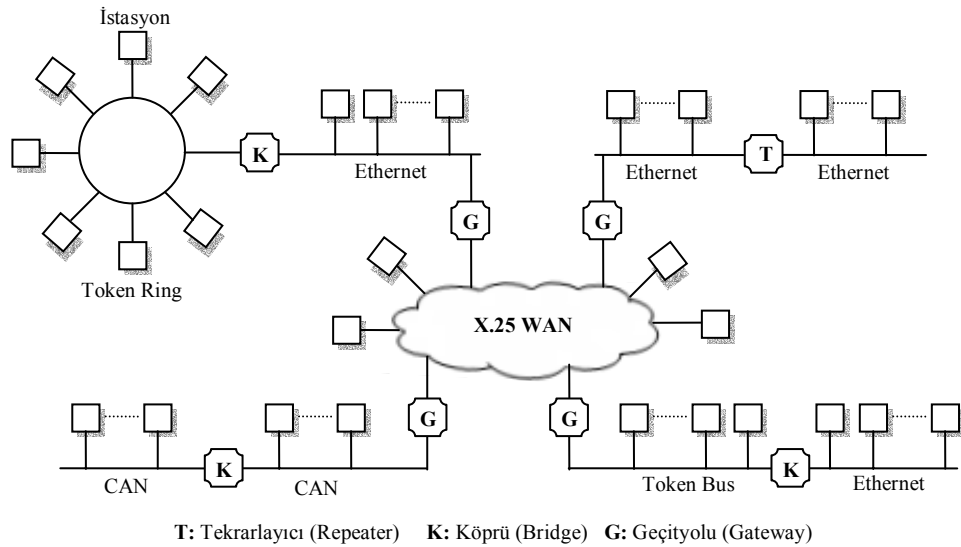
IEEE 802.11 standart ailesi farklı ihtiyaçlara cevap veren ve farklı veri iletim hızlarına sahip alt standartlardan oluşmaktadır. Tüm bu standartlar, CSMA/CA ortam erişim yöntemine dayalı olarak çalıştırılmakta ve geliştirilmektedir.

4. ENDÜSTRİYEL AĞLARDA ARABAĞLAŞIM VE CAN/IEEE 802.11B KLAN KABLOSUZ ARABAĞLAŞIM BİRİMİ TASARIMI

4.1. Giriş

Ağlar, bilgi ve servislerin bir iletişim ortamı üzerinden belli kurallar çerçevesinde paylaşımını sağlar. İki ya da daha fazla ağ bir uygulamada beraber kullanılırsa sistemler arasındaki çalışma modu “arabağlaşım” (internetworking) olarak adlandırılır. Arabağlaşım sözlük karşılığı olarak, “iki veya daha fazla ağ arasındaki iletişim” şeklinde tanımlanır (Özçelik, 2002).

Çok farklı alanlarda, farklı ihtiyaçları gidermek üzere kurulan benzer ya da benzer olmayan sistemlerin birbirleri ile haberleşme gereksinimi vardır. Arabağlaşım benzer ağlar arasında olduğu gibi benzer olmayan ağlar arasında da olur. Şekil 4.1’de arabağlaşım cihazları ile birbirine bağlanmış bir ağ örneği görülmektedir.



Şekil 4.1: Arabağlaşım cihazları ile birbirine bağlanmış bir ağ örneği

Arabağlaşımın sağladığı avantajlar şunlardır:

- Farklı ihtiyaçlara cevap veren alt ağların oluşmasını sağlar,
- Farklı şirketler veya gruplar tarafından kullanılan ağların birbirleri ile haberleşmesini sağlar,
- Sunulan hizmetlere kolay erişim sağlar,
- Mevcut ağ yapılarının arabağlaşım cihazları ile alt ağlara ayrılması, ağların bakımını kolaylaştırır.

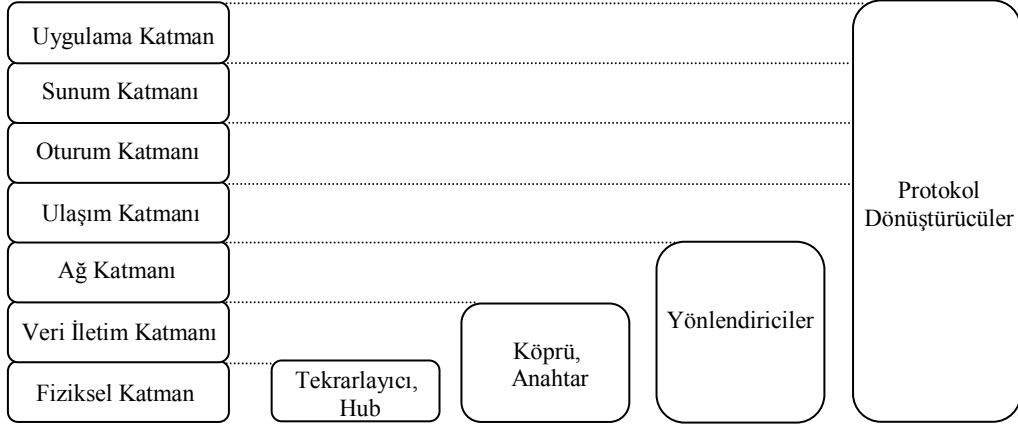
Bu tez çalışmasında, bağımsız CAN segmentlerin IEEE 802.11 KLAN ağlar üzerinden haberleşmesini sağlayan Kablosuz Arabağlaşım Birimi (KAB) tasarımı gerçekleştirilmektedir. Benzer olmayan iki ağın (CAN ve IEEE 802.11 KLAN) arabağlaşımını sağlayan KAB, arabağlantı cihazlarından köprü sınıfına girmektedir.

Bu bölümde; arabağlantı cihazları, endüstriyel ağların arabağlaşımını için sunulan kablolu ve kablosuz çözümler, yapıları ile birlikte incelenecektir. Son olarak KAB'ın ayrıntılı tasarım aşamaları verilmektedir.

4.2. Arabağlaşım Cihazları

Arabağlaşım cihazları, benzer veya benzer olmayan ağların haberleşmesi için kullanılan yüksek performanslı elemanlardır. Bir arabağlaşım cihazı, kaynak ya da hedef olarak çalışmaz. Veri akışının geliş ve gidişinin tutarlılığını korur. Tekrarlayıcı (repeater), hub, köprü (bridge), anahtar (switch), yönlendirici (router) ve geçityolu (gateway) gibi cihazlar Arabağlaşım tanımı içerisinde yer alırlar.

Şekil 4.2'de arabağlaşım cihazlarının OSI referans modelindeki yeri görülmektedir.



Şekil 4.2: Arabağlaşım cihazlarının OSI modelindeki yeri

Tekrarlayıcı:

En basit ağ arabağlantı elemanıdır. Benzer özelliklere sahip ağları birbirine bağlamak için kullanılır. Görevi iletişim hattının fiziksel uzunluğunu arttırmaktır. Yani bireysel bir bağlantıyı daha uzağa götürmek için kullanılır. Tekrarlayıcılar, iletim ortamında zayıflamış, üzerine gürültü eklenmiş olan elektriksel işareti tekrar üreterek gönderir. Tekrarlayıcılar tek giriş ve tek çıkışa sahip arabağlantı cihazlarıdır.

Hub:

OSI referans modelinin ilk katmanında çalışır ve tekrarlayıcı ile benzer bir işleve sahiptir. Tekrarlayıcıdan tek farkı çok sayıda bağlantı noktasının olmasıdır.

Köprü:

Köprüler LAN'ların birbirlerine bağlanması için kullanılırlar ve OSI referans modelinin ilk iki katmanında çalışırlar. Fiziksel olarak ayrı ağları bağlayabilirler. Köprü gerçekte veri iletim katmanı seviyesinde çalışır. Köprüler, sistemlerin iletim uzunluğunu arttırmanın yanı sıra ağların yük paylaşımını ve güvenliğini de sağlarlar. Köprü, aktarılan veriyi tutar ve iletir. Bundan dolayı bir ağ gecikme elemanı olarak da ele alınabilir. Genel olarak köprülerin işlevi, ağlar arası trafik süzme, izole etme ve benzer ya da benzer olmayan ağları, gerekli dönüştürme işlemlerini yaparak birbirlerine bağlamaktır. Köprü kullanılan ağlarda veri iletim işlemleri, köprü

içerisindeki MAC tablosuna dayalı olarak yapılır. Endüstriyel ağların arabağlaşımında genellikle köprü cihazları kullanılır (Öner, 2003, Özçelik, 2002).

Anahtar:

Köprü gibi OSI referans modelinin ilk iki katmanında çalışır. Hub'a benzer bir yapısı olmakla birlikte farklı olarak trafiği tüm düğümler yerine ilgili düğüme iletir (Çeken, 2004).

Yönlendiriciler:

OSI referans modelinin ilk 3 katmanını içeren yönlendiriciler, ağlar arasında bağlantı amacıyla kullanılırlar. Yönlendiriciler de köprüler gibi; süzme, trafik izole etme ve ağları birbirine bağlama işlevlerini yürütürler. Ancak yönlendiriciler köprülere göre daha çok bilgiye (ağ adresleri) ulaştıkları için bu bilgiyi kullanarak daha iyi paket dağıtımını yaparlar.

Yönlendiricileri, köprülerden ayıran temel farklar şunlardır. Yönlendiriciler OSI modelinin ilk 3, köprüler ise ilk 2 katmanında çalışırlar. Yönlendiriciler, tablolarında ağ adreslerini, köprüler ise MAC adreslerini tutarlar. Yönlendiriciler sadece kendilerine adreslenmiş paketleri incelerler. Köprüler ise LAN'lardaki bütün çerçevelerin varış adreslerini incelerler. Yönlendiriciler aldıkları paketi, yol kontrolü yaparak yönlendirirken, köprü herhangi bir yol kontrolü yapmadan paketi ağın diğer tarafına iletir (Öner, 2003).

Geçityolu (Protokol Dönüştürücü):

En karmaşık ağ arabağlantı elemanı olan geçityolları; mimarileri, protokolleri ve veri biçimleri farklı olan ağların bağlanması için kullanılır. Ağ geçidi olarak da adlandırılan bu cihazlar, ağlar arası mesaj ve protokol dönüşümlerini yapar. OSI referans modelindeki tüm katmanların fonksiyonlarını icra edebilen bir cihazdır.

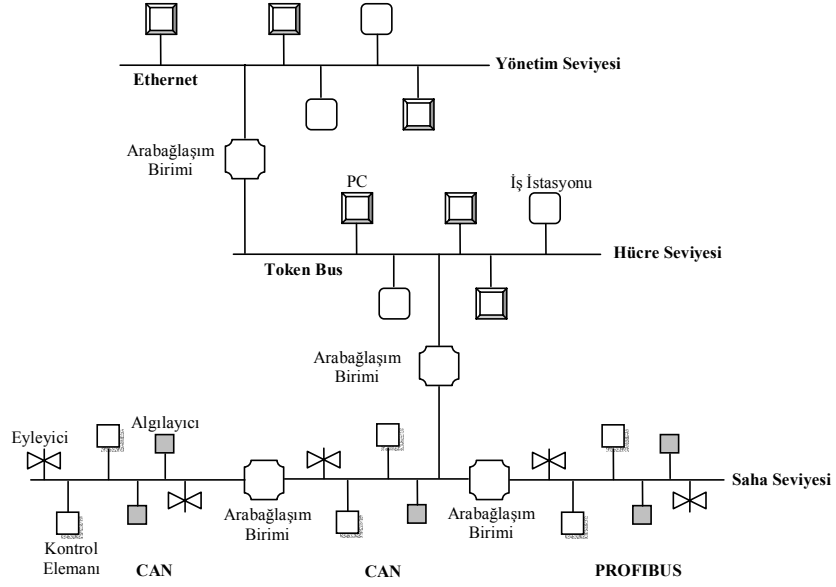
4.3. Endüstriyel Ağlarda Arabağlaşım

Haberleşme ağlarında olduğu gibi Sahayolu (FieldBus) sistemlerinde de arabağlaşım gereksinimi vardır. Sahayolu sistemlerinde arabağlaşım gereksiniminin nedenleri şunlardır:

- Sahayolu sistemlerinin genişletilmesinde ardışık olarak arabağlantısı yapılan kontrol elemanlarının sınırlı miktarı,
- Dağıtık alanın büyüklüğü ve veriyolu uzunluğundan kaynaklanan sınırlamalar,
- Bağımsız sahayolu sistemlerinin birbirleri ile haberleşme ihtiyacı,
- Sahayolu sistemlerinin diğer ağlar ile (LAN vb.) haberleşme ihtiyacı.

Sahayolu sistemlerinin bu sınırlamalarını ve ihtiyaçlarını gidermek amacıyla genellikle endüstriyel kontrol ortamı birkaç sahayolu (FieldBus) ağı oluşacak şekilde bölünerek her bir sahayolu ağı, Arabağlaşım Birimi (AB, Interworking Unit) olarak adlandırılan elemanlar yardımıyla birbirlerine bağlanır. Endüstriyel kontrol ve fabrika otomasyonu uygulamalarında BBİ modeline uygun olarak genellikle hiyerarşik bir yapı söz konusudur ve farklı seviyelerden oluşan katmanlar birbirine bağlanır. Üst katmanlarda denetim işlerini kotaran bilgisayarlar yer alırken, alt katmanlarda üst katmandan gelen komutları/görevleri yerine getiren algılayıcı, eyleyici, alan ve kontrol elemanları bulunmaktadır. Bu seviyeler arasındaki haberleşme de yine Arabağlaşım Birimleri (AB) tarafından sağlanmaktadır. Bu nedenle, karmaşık sistemlerin bütün birimlerinin kablolu bir altyapı ile bağlanamayacağı aşikardır.

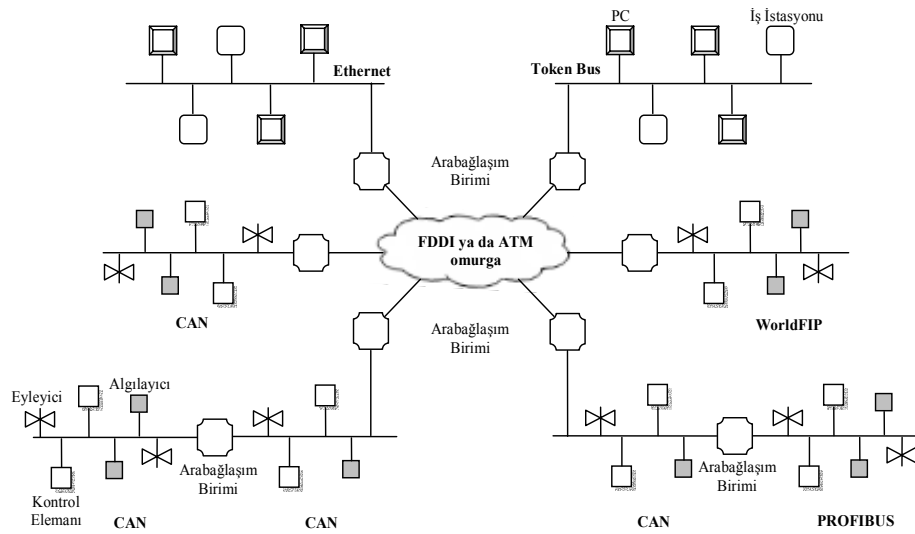
Şekil 4.3'de bir endüstriyel üretim ortamında BBİ mimarisine göre arabağlaşım yapısı görülmektedir. Çoğunlukla tercih edilen bu çözüm, kritik zaman uygulamalarında önemli sorunlara neden olmaktadır. Zira, iki sahayolu (FieldBus) sistemin birbirine doğrudan bağlanamaması, ister istemez ek bir gecikmeye neden olmaktadır. Bu gecikme süresini azaltmak için sahayolu ağları çoğunlukla bir omurgaya bağlanır. Günümüzde omurga olarak kablolu (FDDI, ATM vb.) ve kablosuz yapıların kullanımı üzerine çalışmalar yapılmaktadır (Bayılmış ve diğ., 2004c, 2005a, 2005c, Cavalieri ve Pano, 1998, Willig ve diğ., 2005).



Şekil 4.3: BBİ mimarisine göre endüstriyel ağlarda arabağlaşım

4.3.1. Endüstriyel ağlarda kablolu omurga kullanımı

Endüstriyel ağların arabağlaşımında gecikmenin azaltılması önemli ölçüde kullanılan kablolu omurganın bant genişliğine bağlıdır. Bu nedenle omurga ağlarda çoğunlukla FDDI ya da ATM teknolojileri tercih edilmektedir. Şekil 4.4’de bir endüstriyel kontrol ortamında BBİ mimarisine göre farklı seviyelerin kablolu omurga üzerinden bağlantısı görülmektedir.



Şekil 4.4: Endüstriyel ağların arabağlaşımında kablolu omurga kullanımı

Literatürde endüstriyel ağların arabağlaşımı için kablolu omurga kullanımı üzerine yapılan birçok çalışma bulunmaktadır. Bunlardan birkaçı aşağıda özetlenmektedir.

Ekiz ve diğ. (1996, 1997), Ekiz (1997), CAN sahayolu sisteminin, günümüzde kullanımı yaygın olan ve düşük maliyete sahip Ethernet teknolojisi ile iletişimini sağlayan CAN/Ethernet köprü tasarımını gerçeklemiştir. Ethernet, bir omurga ağ olmamasından, sınırlı bant genişliğine sahip olmasından ve paylaşımlı bir ortam kullanmasından dolayı gerçek zamanlı iletişimlere destekleyememe gibi önemli dezavantajlara sahiptir. Bu dezavantajlarına rağmen düşük maliyeti, yaygın kullanımı ve evrensel bir iletişim arayüzü olması sebebiyle sahayolu iletişim sistemlerinin genişletilmesinde ve diğer ağlar ile entegrasyonunda tercih edilen bir çözüm yöntemidir.

Kunert (1997), Kunert ve Zitterbart (1998) çalışmalarında tek bir PROFIBUS ağın ve bağımsız PROFIBUS segmentlerin ATM üzerinden haberleşmesini sağlayan arabağlaşım birimi için bir model tanımlamışlardır.

Vitturi (2001), çalışmasında fabrika otomasyonunda sahayolu seviyesinde Ethernet teknolojisinin kullanımı için Ethernet temelli haberleşme profili önermiş ve önerdiği yapıyı sahayolu sistemlerinden PROFIBUS–DP ve WorldFIP ile karşılaştırmıştır. Ethernet'in orijinal versiyonu sahayolu seviyesindeki gerçek zamanlı iletişim gereksinimini karşılamamasına rağmen, günümüzdeki uyarlamaları (Gigabit Ethernet vb.) bu yönde önemli değişikliklere sahiptir. Yine Vitturi (2003) çalışmasında sahayolu seviyesinde gerçek zamanlı iletişim için Ethernet üzerinde PROFIBUS–DP fonksiyonlarının gerçekleştirildiği ve mevcut PROFIBUS–DP uygulamaları ile uyulaşabilen DP–Ethernet olarak adlandırdığı yeni bir yapı önermiştir.

Özçelik (2002), sahayolu iletişim sistemlerinin, gerçek zamanlı haberleşme gereksiniminin karşılanması için, ATM protokolü ile iletişimi üzerine çalışma gerçekleştirmiştir. Bu çalışmasında, CAN/ATM ve PROFIBUS/ATM yerel köprü tasarımlarını Network II.5 benzetim programını kullanarak modellemiş ve aynı zamanda her iki yerel köprünün tasarım ve gerçekleştirme aşamalarını ayrıntılı olarak sunmuştur. Özçelik (2002), Özçelik ve Ekiz (2004), Özçelik ve Ekiz (2006)

çalışmalarında sunduğu başarımları sonuçları ile gerçekleştirmiş olduğu yerel köprülerin sahayolu sistemlerinin gereksinimlerini karşıladığını ortaya koymuştur.

Aydoğan (2005), yapmış olduğu çalışmasında CAN, PROFIBUS ve WorldFIP olmak üzere farklı sahayolu sistemlerinin ATM omurga üzerinden haberleşmesini sağlayan bir ağ mimarisi tasarlamıştır. Çalışmasını Network II.5 benzetim programını kullanarak gerçekleştirmiştir.

Tenruh ve Tarimer (2005) çalışmalarında CAN sahayolu sistemlerinin ATM ağı üzerinden uzaktan köprüleme yöntemini kullanarak genişletecek bir yapı önermektedir.

Yukarıda kısaca değinilen çalışmalardan da anlaşılacağı gibi, endüstriyel ağların dağıtık alanın büyüklüğü ve birbirleri ile ve diğer ağlar ile haberleşme gereksinimlerinin karşılanması için kablolu omurga kullanımı üzerine birçok çalışma yapılmaktadır. Fakat standartların ve endüstriyel protokollerin çeşitliliği nedeniyle istenen ideal çözüm henüz sağlanamamıştır. Bu durum endüstriyel ağlar için hala bir sorun teşkil etmektedir. Çalışmalarda kablolu iletişim tercihi olarak Ethernet ve ATM teknolojileri üzerinde yoğunlaşmaktadır. ATM gerçek zamanlı iletişim gereksinimini omurga yapıda ve uzak mesafelerde karşılaması sebebiyle tercih edilmesine rağmen, düşük maliyeti ve yaygın kullanımından dolayı Ethernet uygulamalarına daha sık rastlanılmaktadır.

Günümüzde birçok firma tarafından geliştirilen CAN/Ethernet arayüzlerine sahip ürünler bulunmaktadır. Yine CAN ve Ethernet portlarına sahip mikrodenetleyicilerin (DS80C410, DS80C411 vb.) kullanımı da yaygınlaşmaktadır.

Endüstriyel ağların arabağlaşımında kablolu çözümler aşağıda belirtilen özelliklerden dolayı tercih edilmemektedir:

- Endüstriyel üretim ortamlarında en önemli kriter, mevcut sistemlerin kolaylıkla genişletilebilmeleridir. Mevcut kurulu yapılar içerisinde sonradan kablolama işlemleri büyük bir sorundur.

- Kablolama, hareketli cihaz/robot ve taşınabilir bilgisayar gibi elemanların kullanıldığı endüstriyel otomasyon uygulamalarında büyük bir engel teşkil etmektedir.
- Kimyasal üretim gibi çevresel etmenlerin göz önünde bulundurulması gereken endüstriyel ortamlarda kablolama büyük bir tehlike arz etmektedir. Aynı zamanda bu tip ortamlarda elde edilen bilgilerin kablolu ortamdan aktarımı, ortamdan kaynaklanan parazitlerden (kablolunun ısınması vb.) dolayı sonuçları da etkilemekte ya da harici donanım kullanımına sebep olmaktadır.

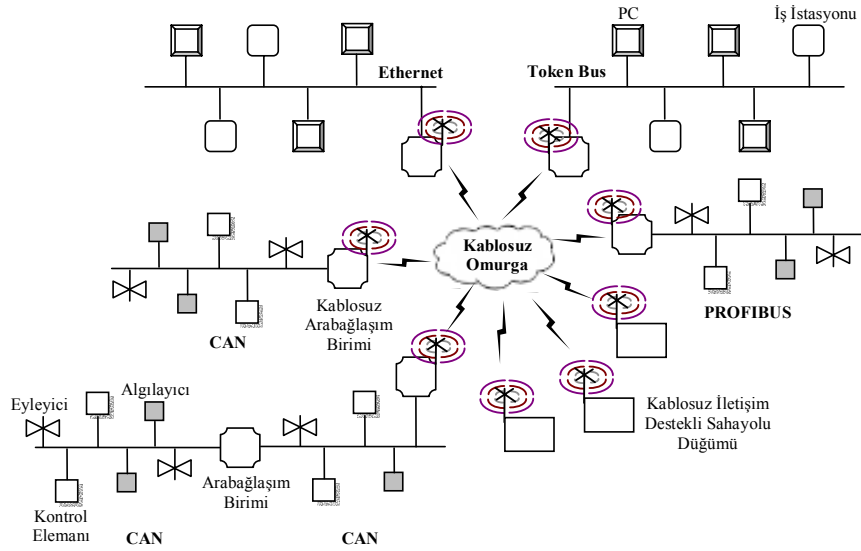
Endüstriyel ortamlarda kablolu omurga kullanımı, kablolama zorlukları, endüstriyel kontrol ortamının mevcut alt yapısı gibi nedenlerden dolayı son derece zordur. Dolayısıyla, günümüzde sahayolu sistemlerinin arabağlantısında, sağlayacağı kolaylıklar göz önüne alınarak kablosuz iletişimin kullanımı artık önemli bir alternatif teşkil etmektedir (Bayılmış ve diğ., 2005a, 2005c, Cavalieri ve Pano, 1998).

4.3.2. Endüstriyel ağlarda kablosuz iletişim kullanımı

Kablosuz ağlar, kablo kullanan eşleniklerinden farklı olarak kurulum kolaylığı ve ölçeklenebilirlik, hareketlilik, üretkenlik, ileriye yönelik maliyet kazancı, mevcut yerel ağ yapısını genişletme gibi sunduğu birçok avantajdan dolayı günümüzde pek çok uygulamanın yanı sıra endüstriyel ağların haberleşmesinde de kullanılmaktadır. Günümüzde farklı veri iletim hızlarına sahip ve farklı ihtiyaçlara cevap vermek üzere IEEE 802.11 KLAN, HiperLAN, Bluetooth, ZigBee, Kablosuz ATM gibi mevcut ve geliştirilmekte olan birçok kablosuz iletişim teknolojisi bulunmaktadır. Bu kablosuz iletişim teknolojileri kullanılarak sahayolu sistemlerinin arabağlaşımı için birçok çalışma yapılmaktadır.

Kablosuz sistemlerde genellikle hiyerarşik bir yapı söz konusu değildir. Bu özellik kritik zaman gecikmeli uygulamaların kablosuz iletişim teknolojileri üzerinden rahatlıkla gerçekleştirilebileceğinin göstergesidir. Şekil 4.5’de kablosuz iletişim yöntemi kullanılarak endüstriyel ağların arabağlaşımı görülmektedir. Burada kablosuz iletişim desteğinin sahayolu sistemlerinde iki farklı kullanımı verilmektedir.

Birincisinde CAN, PROFIBUS gibi sahayolu sistemi köprü görevi gören bir Kablosuz Arabağlaşım Birimi üzerinden kablosuz omurgaya bağlanırken, diğerinde ise tek bir sahayolu düğümü doğrudan kablosuz iletişim desteği ile kablosuz omurgaya bağlanmaktadır. Şekil 4.5’de kablosuz omurgadan kasıt, farklı kablolu (Ethernet, Token Bus CAN vb.) yapıların aynı kablosuz ortam üzerinden iletimidir.



Şekil 4.5: Endüstriyel ağlarda kablosuz iletişimin kullanıldığı bir ağ

Literatürde endüstriyel ağların arabağlaşımı için kablosuz teknolojilerin kullanımı üzerine yapılan birçok çalışma bulunmaktadır. Bunlardan birkaçı aşağıda özetlenmektedir.

Kutlu ve diğ. (1996a) ve (1996b), çalışmalarında dağıtık ve merkezileştirilmiş CAN sahayolu sistemlerinin kablosuz haberleşmesi için önerilen WMAC ve RMAC protokollerinin SAE Benchmark kullanılarak gerçekleştirilen başarımlarını sunmuştur. 125 Kbit/s ve daha yüksek veri iletim hızlarında CAN segmentlerin kablosuz haberleşmesinin gerçek zamanlı iletişim gereksinimini karşıladığını göstermiştir.

Cavalieri ve Pano (1998), IEC/ISA sahayolu sisteminin arabağlaşım problemini gidermek için kablosuz omurga olarak IEEE 802.11 kablosuz LAN’ın kullanıldığı bir çözüm önermiştir. Aynı zamanda IEC/ISA sahayolu sisteminin kritik zamanlı

iletişim gereksinimini karşılamak için IEEE 802.11 KLAN'ın ortam erişim mekanizmasında (MAC) küçük değişiklikler içermektedir.

Alves (2003), çalışmasında PROFIBUS temelli melez (hybrid) kablolu/kablosuz haberleşme mimarisi önermiştir. Önerdiği yapı ile gerçek zamanlı iletişimin garanti edilmesi ve desteklenmesi için yaklaşımlar ve uygun mekanizmalar sunmuştur. Kablosuz iletişim standardı olarak IEEE 802.11b'yi kullanmıştır.

Willig (2003a) çalışmasında kablolu ve kablosuz PROFIBUS düğümlerin sanal bir halka oluşturacak şekilde haberleşmesini sağlayan bir yapı sunmuştur. Amacı mevcut kablolu sahayolu sistemleri ile kablosuz segmentleri birleştirmektir. Willig (2003b) çalışmasında ise gerçek zamanlı iletişim gereksinimlerine göre kablosuz PROFIBUS'ın ortam erişim ve veri iletim katmanı için önerilen iki farklı yaklaşımı karşılaştırmıştır. Kablosuz PROFIBUS'lar için polling-based protokollerin uygun olabileceğini ortaya koymuştur.

Ertürk (2003) ve (2005) çalışmalarında sabit uzak CAN segmentlerin Kablosuz ATM (KATM) üzerinden haberleşmesini sağlayan kapsülleme tekniğine dayalı bir yapı sunmuştur. KATM kullanımı, CAN temelli dağıtık kontrol sistemlerin global sistemlerle (kablolu ATM omurga üzerinden) kolaylıkla entegrasyonunu sağlayacaktır. Aynı zamanda ATM'in kablolu ortamda sağladığı yüksek veri iletim hızı ve servis kalitesi özelliklerini kablosuz ortamda da gerçekleştirmeyi amaçlayan KATM, endüstriyel kontrol ortamlarında önemli bir kriter olan gerçek zamanlı iletişimi desteklemektedir. Çalışmanın başarımların analizi SAE Benchmark'a göre gerçekleştirilmiştir.

Koulamas ve diğ. (2004), melez (hybrid) kablolu/kablosuz PROFIBUS sahayolu sistemlerinde kullanılacak doğrudan geçişli (Cut-Through) iletim elemanı tanımlamış ve mimarisi ile çalışmasını vermiştir. Doğrudan geçişli elemanın iletim gecikmesinin ayrıntılı analitik modelini sunmuştur. Tek bir MAC kullanılan farklı fiziksel yapılarda (melez kablolu/kablosuz), doğrudan geçişli elemanın kullanılabilmesini ortaya koymuştur. Kablosuz iletişim teknolojisi olarak IEEE 802.11 KLAN'ı kullanmıştır.

Miorandi ve Vitturi (2004), Ethernet ve Bluetooth temelli PROFIBUS–DP sahayolu sisteminin melez kablolu/kablosuz ađlara uygulanabilirliđinin analizini sunmaktadır. Teorik alıřılan bu arařtırmada belirtilen sistemin katmanlı protokol yapısı verilmektedir. Aynı yazarlar (2005) diđer bir alıřmalarında ise Bluetooth kablosuz teknolojisine dayalı olarak PROFIBUS–DP sahayolu sistemlerinin kablosuz iletiřimi iin bir özüm sunmaktadırlar. Bu özüm kapsamında, BT–PROFI olarak adlandırdıkları yeni bir protokol önermiřlerdir.

Willig ve diđer. (2005), endüstriyel ađların arabađlařımı iin kullanılan kablosuz teknolojileri ve uygulamaları sunmaktadır. Aynı zamanda sahayolu sistemlerinin güvenilirlik ve kritik zamanlı haberleřme gereksinimlerinin kablosuz teknolojiler ile sađlanabileceđini tartıřmıřtır.

De Pellegrini ve diđer. (2006), alıřmalarında saha seviyesinde kablosuz ađların kullanımını ele almakta ve haberleřme gereksinimlerini incelemektedir. 3 katmanlı genel bir kablosuz sahayolu protokol mimarisi önermektedir. Kablosuz sahayolu sistemlerinde kablosuz iletiřim standardı olarak IEEE 802.11 ve Bluetooth'un kullanımını önermiřlerdir. Diđer alıřmalardan farklı olarak veri güvenliđi ve gü tüketiminden de bahsetmektedirler.

Yukarıdaki alıřmalardan da anlařılacađı gibi, endüstriyel ađların kablosuz iletiřim ortamı üzerinden arabađlařımı üzerine birok alıřma yapılmıřtır. Yapılan alıřmalarda kablosuz ortamın dođasından kaynaklanan hata sınırlamalarının özellikle sahayolu sistemlerin güvenilirlik ve gerek zamanlı iletiřim gereksinimlerini karřılamada problem olacađı görölmektedir. Ancak yüksek veri iletim hızlarına sahip uygulamalarda bunun sorun olmayacađı ortaya konmuřtur.

Endüstriyel ađların arabađlařımı iin birok farklı kablosuz iletiřim standardı kullanılmasına rađmen, uygulanabilirlik, düřük maliyet, yaygın kullanım gibi özelliklerden dolayı kablosuz Ethernet olarak da adlandırılan IEEE 802.11 KLAN üzerine daha ok alıřma yapılmaktadır. Bunun en önemli sebebi günümüzde oldukça yaygın bir kullanım oranına sahip standart kablolu Ethernet yapısı ile sađladıđı kolay entegrasyondur.

Yukarıda verilen araştırma çalışmalarının yanı sıra bazı firmalar tarafından değişik ürünler de geliştirilmektedir. Örneğin RMCAN firmasının RF, UHF, Bluetooth, GSM, GPRS ve WLAN kablosuz arayüzlerin kullanıldığı ürünleri bulunmaktadır (<http://www.rmcan.com>). Geliştirilen bu ürünler köprü temelli arabağlaşım elemanı değil, sadece CAN temelli kablosuz arabirimlerdir. Bunun yanı sıra eğitim ve öğretim amaçlı geliştirme setleri de (evaluation board) bulunmaktadır. Tüm bu gelişmelere rağmen ideal bir çözüm henüz elde edilememiştir. Bu yönde çalışmalar devam etmektedir.

Bu tez çalışmasında gerçekleştirilen CAN/IEEE 802.11b KLAN KAB'ı yukarıda verilen çalışmalardan ayıran temel özellikler şunlardır:

- a. Kablosuz iletim teknolojisi olarak aşağıdaki sebeplerden dolayı IEEE 802.11b KLAN tercih edilmiştir.
 - i. Dağıtık endüstriyel kontrol ortamı ve saha seviyesi dikkate alındığında düşük maliyeti, uygulanabilirliği, mevcut kablolu Ethernet yapıları ile sağladığı kolay entegrasyon gibi özellikleri sebebiyle IEEE 802.11 KLAN kablosuz erişim teknolojisi tercih edilmiştir.
 - ii. CAN veri iletim hızı maksimum 1 Mbit/s'dir. IEEE 802.11b KLAN'ın veri iletim hızı ise maksimum 11 Mbit/s'dir. Böylelikle CAN ve KLAN ağlarının arabağlaşımını sağlayan KAB, CAN veri iletim hızını 11 kat arttırmaktadır. Bu hız endüstriyel kontrol uygulamalarının gereksinimlerini karşılayabilecek sınırlar içerisindedir.
 - iii. Bu çalışma veri iletim hızından ziyade, CAN ve IEEE 802.11 KLAN ortam erişim mekanizmalarının birlikte çalışabilirliği üzerine odaklanmaktadır. IEEE 802.11 KLAN standart ailesini oluşturan IEEE 802.11a/b/g gibi tüm alt standartlar aynı ortam erişim mekanizmasını (CSMA/CA) kullanmaktadır. Aralarındaki temel fark, farklı veri iletim hızı sağlayan DSSS, FHSS, OFDM gibi farklı fiziksel katman seçenekleri sunmalarıdır. Fiziksel katman seçeneğinin değiştirilmesi gerçekleştirilen KAB'ın mekanizmasında büyük bir değişiklik gerektirmeyecektir.
- b. Literatür özetinde verilen tüm çalışmalarda özellikle önerilen arabağlaşım biriminin sebep olduğu gecikme üzerine odaklanılmaktadır. Gecikme analizine

göre önerilen arabağlaşım elemanlarının sahayolu sistemlerinin gereksinimlerini karşılayıp/karşılayamayacakları ortaya konmaktadır. Ancak ayrıntılı bir tasarım mimarisi verilmemektedir. Bu tez çalışmasında ise bağımsız CAN segmentlerin IEEE 802.11b KLAN üzerinden haberleşmesini sağlayan KAB'ın;

- i. Gerçekleme aşaması için ayrıntılı tasarım mimarisi verilmektedir.
- ii. OPNET Modeler yazılımı kullanılarak SAE Benchmark'a göre başarımlar analizi sunulmaktadır.
- iii. Temel prototip gerçekleştirilerek kullanılabilmesi için örnek uygulamalar verilmektedir.

4.4. CAN/IEEE 802.11b KLAN Kablosuz Arabağlaşım Biriminin Tasarımı

Endüstriyel ortamda üretim seviyesinde bulunan cihazların, hem birbirleri arasında (CAN, PROFIBUS vb. segmentlerin) hem de planlama seviyesindeki bilgisayarlar (Ethernet, Token Bus/Ring, ATM vb) ile haberleşme ihtiyacı vardır.

CAN başta otomotiv olmak üzere endüstriyel ortamlardaki birçok dağıtık kontrol uygulamalarında yaygın olarak kullanılan gerçek zamanlı bir iletişim sistemidir. Endüstriyel uygulamalardaki yaygın kullanımı DeviceNet, SDS, CANOpen, CANKingdom gibi CAN tabanlı ağ çözümlerinin ortaya çıkmasına sebep olmuştur (Ekiz, 1997).

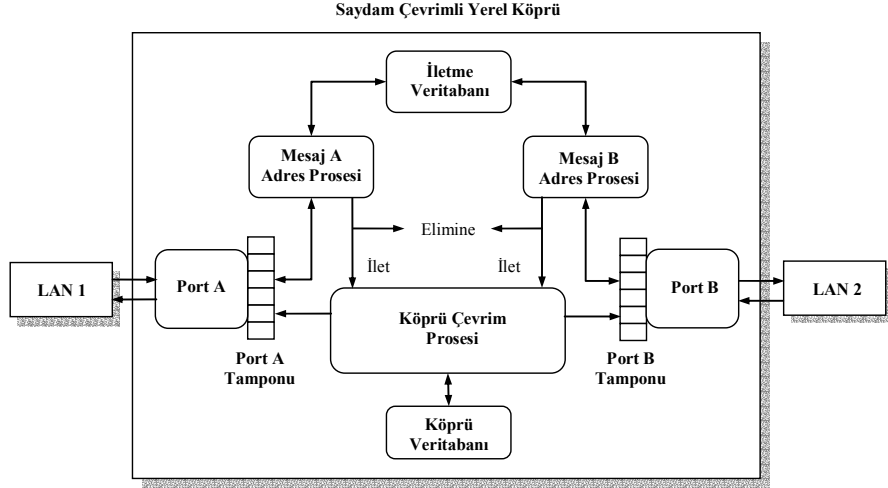
CAN temelli dağıtık kontrol sistemlerinin, dağıtık alanın büyüklüğü ve uzak CAN segmentlerin birbirleri ve diğer yerel alan ağları (LAN) ile haberleşme ihtiyacı olmak üzere iki temel problemi vardır. Literatürde bunların çözümü üzerine birçok çalışma yapılmıştır (Ekiz, 1997, Ekiz ve diğ., 1996, 1997, Özçelik, 2002, Tenruh, 2005).

Bu bölümde, yukarıda bahsedilen CAN segmentlerin problemlerini çözmek için önerilen Kablosuz Arabağlaşım Birimi (KAB)'ın tasarımı ayrıntılı olarak verilmektedir. KAB, kapsülleme metodunu kullanarak IEEE 802.11b KLAN üzerinden uzak CAN segmentlerin hem birbirleri ile hem de hücre/yönetim seviyelerindeki Ethernet tabanlı sistemler ile haberleşmesini sağlar.

Bu bölümde sunulan KAB tasarımında, kablosuz iletim teknolojisi olarak IEEE 802.11b KLAN tercih edilmiştir. Böylelikle CAN sistemlerin KAB vasıtasıyla hem kablosuz sistemler hem de Ethernet tabanlı kablolu sistemler ve internet ile bütünleşmesi sağlanmaktadır.

Benzer olmayan iki ağın (CAN ve IEEE 802.11 KLAN) arabağlaşımını sağlayan KAB, arabağlantı cihazlarından köprü sınıfına girmektedir. Köprüler OSI referans modelinin mantıksal bağ kontrolü (LLC) ya da ortam erişim kontrol (MAC) seviyesinde çalışan yüksek başarılı cihazlardır. LLC seviye köprüleme, benzer olmayan çerçeve biçimleri ve farklı veri iletim katman protokolleri kullanan LAN'ların birbirine bağlantısını sağlar. LAN'lar farklı çerçeve biçimlerine veya veri iletim protokollerine sahip olduklarında, aralarındaki iletişimin sağlanabilmesi için öğrenme, filtreleme ve uyumlaştırma (dönüştürme) gibi işlemlerin gerekli olduğu ortaya çıkar (Özçelik, 2002). Benzer olmayan ağların arabağlaşımı için gerekli tüm bu işlevler, Saydam-Çevrimli yerel bir köprü (Transparent-Translation Local Bridge) ile sağlanabilir.

Şekil 4.6'da saydam-çevrimli yerel bir köprünün iç şeması görülmektedir. Saydam-çevrimli yerel köprü, bir portu vasıtasıyla aldığı mesajı, filtreleme işlemlerinden geçirdikten sonra, (eğer mesaj köprünün diğer portunun bağlı olduğu LAN'ı hedefliyorsa) diğer portunun bağlı olduğu LAN'ın biçimine dönüştürerek gönderir. Bu tip köprüler, dönüştürme işlemlerinde seri iletişim protokollerinde yaygın olarak kullanılan kaynak/hedef adres veya LAN numarası gibi bilgileri kullanırlar. CAN, kaynak/hedef adres ve LAN numarası yerine mesajın içerisinde mesajın önceliğini belirten bir tanımlayıcı alan içerirken, KLAN ise kaynak/hedef adres bilgileri içerir. CAN'nin bu özelliğinden dolayı CAN ve KLAN ağların arabağlaşımı için geleneksel adres tabanlı köprüler kullanılamaz.



Şekil 4.6: Saydam-çevrimli yerel köprü

Bu bölümde, CAN ve KLAN ağların arabağlaşımı için gerekli olan işlevler, CAN/IEEE 802.11b KLAN KAB'ın önerilen ağ mimarisi, OSI referans modeline uygun katmanlı yapısı, kapsülleme işlemi ve KAB'ın işlevsel yapısı ayrıntılı olarak verilmektedir.

4.4.1. KAB'ın görevleri

Bir köprü, benzer olmayan ağların birbirine bağlanmasının yanı sıra, iletişim hattının fiziksel uzunluğunun arttırılması, trafik yoğunluğunun ayrıştırılması ve farklı fiziksel ortamların birlikte kullanılması gibi hizmetleri de sağlar. Bu tez çalışmasında gerçekleştirilen CAN/IEEE 802.11b KLAN Kablosuz Arabağlaşım Birimi, belirtilen tüm bu hizmetleri sunmaktadır.

KAB'ın IEEE 802.11b KLAN ağlar üzerinden CAN segmentlerin iletişimini sağlayabilmesi için öğrenme, filtreleme ve uyumlaştırma (dönüştürme) gibi işlemleri yerine getirmesi gerekmektedir. KAB, benzer olmayan iki ağ arasında arabağlaşımı sağlarken ağlar arasında uyumlaştırma işlemi için aşağıdaki nedenlerden dolayı en kötü durum (worst-case) dönüşümünü içermektedir.

- Her iki ağın çerçeve biçimlerinin ve boyutlarının farklı olması: IEEE 802.11b KLAN çerçevesi, CAN çerçevesinden daha uzun bir boyuta sahiptir (Şekil 4.9). Bundan dolayı dönüşüm için dolgu (padding) alanların eklenmesi/kaldırılması

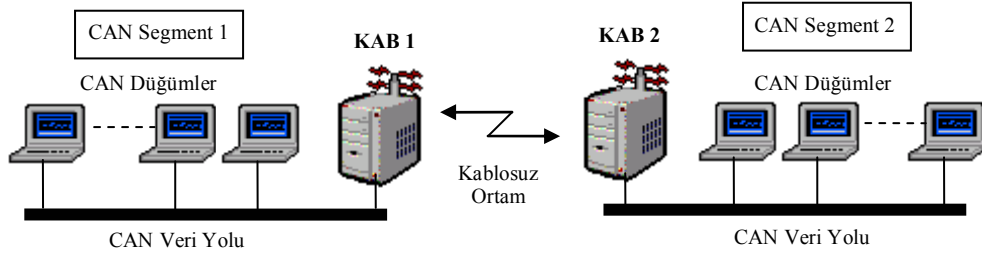
gerekir. Her iki çerçeve biçiminin birbirinden tamamen farklı olması, dönüşüm işlemi sırasında bazı alanların çıkarılması ya da yeni alanların oluşturulmasını gerektirir.

- Farklı öncelik mekanizması kullanmaları: CAN’de mesaj önceliğine dayalı bir mekanizma kullanılırken, IEEE 802.11b’de ise genel olarak bir öncelik yapısı yoktur. Ancak PCF erişim mekanizmasında ya da DCF erişim mekanizmasında RTS–CTS yapısı kullanılarak öncelik temelli bir yapı oluşturulabilir.
- Farklı ortam erişim kontrol mekanizmalarına ve topolojilerine sahip olmaları: Temelde her iki ağın kullandığı ortam erişim mekanizması birbirine benzemektedir. CAN yıkıcı olmayan çarpışma çözümü (non-destructive collision resolution) ve öncelik temelli ortam erişim yöntemi ile düğümlerin birbirine ortak yol (broadcast bus) topolojisi ile bağlandığı bir sistemdir. CAN tanıtıcı alan üzerinde öncelik esasına dayalı olan Taşıyıcı Duyarlı Çoklu Erişim/Mesaj Öncelik Denetimli Çarpışma Sezme (CSMA/CD+AMP) protokolünü ortam erişim metodu olarak kullanır. IEEE 802.11b KLAN ise hücrel mimariye dayalı ağ topolojisini (eşe–eş ve erişim noktalı) destekler ve ortam erişim mekanizması olarak Çarpışmadan Kaçınma ile Taşıyıcı Duyarlı Çoklu Erişim (CSMA/CA) ve kısmen gerçek zamanlı iletişimi desteklemek için PCF’yi kullanır.
- CAN mesajlarının kaynak ve hedef adreslerine sahip olmaması: CAN’daki düğümler, IEEE 802.11b KLAN’dan farklı olarak herhangi bir kaynak/hedef adres bilgisine sahip değildir. Bunun yerine CAN düğümlerin ürettikleri her bir mesaj, tüm ağ içerisinde tek olan bir tanıtıcı (identifier) bilgisine sahiptir. Bu tanıtıcı bilgisi, üretilen mesajların iletim önceliğini ve mesajların kabul/reddedilmesini belirler.

4.4.2. KAB’ın kullanıldığı ağ mimarisi

Şekil 4.7’de CAN segmentlerin IEEE 802.11b KLAN üzerinden arabağlaşımını sağlayan KAB’ın kullanıldığı örnek bir ağ yapısı görülmektedir. KAB, CAN ve KLAN ağlar ile haberleşmek üzere iki ayrı porta sahiptir. KAB’ın kullanıldığı böyle bir ağın genel çalışması şu şekildedir:

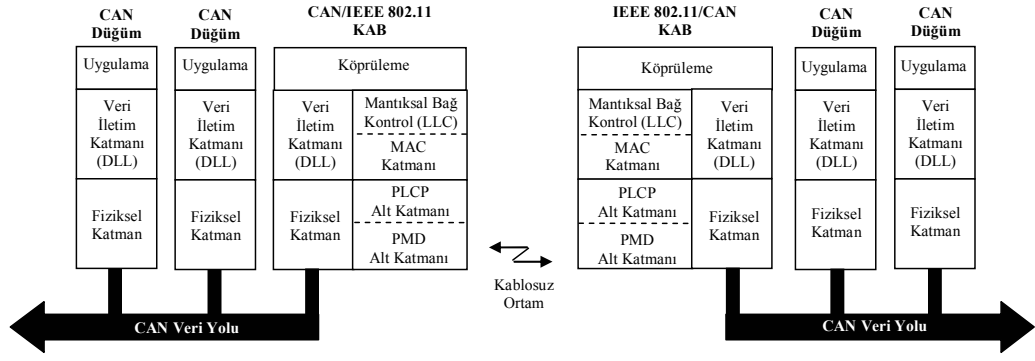
- CAN düğümler yerel (local) ve uzak (remote) mesajlar üreterek veriyoluna iletirler.
- KAB, CAN portu vasıtası ile CAN veriyolu üzerindeki mesajları (broadcast) alır, filtreler, eğer CAN mesaj uzak mesaj ise (diğer CAN segmenti hedefliyorsa) gerekli dönüşüm işlemlerini yapar ve KLAN portu vasıtasıyla hedef segmente göndermek üzere kablosuz ortama iletir.
- KAB, KLAN portu vasıtası ile kablosuz ortamdan KLAN çerçeveyi alır, filtreler (kendini adresliyorsa) ve gerekli dönüşüm işlemlerini gerçekleştirerek, CAN portu ile bağılı bulunduğu CAN veriyoluna uzak CAN mesajını iletir.
- CAN düğümü, veriyolu üzerindeki CAN mesajı tanıtıcı alanına göre kabul eder.



Şekil 4.7: KAB'ın kullanıldığı örnek bir ağ

4.4.3. KAB'ın OSI katmanlı mimarisi

IEEE 802.11b KLAN üzerinden uzak CAN segmentlerin arabağlantısını sağlayan KAB'ın farklı çalışma mekanizmalarına sahip iki ayrı portu bulunmaktadır. KAB, köprü temelli bir eleman olduğundan OSI referans modelinin ilk iki katmanında çalışır. Yukarıda verilen ağ mimarisine göre KAB'ın kullanıldığı sistemin OSI referans modeline dayalı katmanlı mimarisi Şekil 4.8'de görülmektedir (Bayılmış ve diğ., 2004c, 2005a, 2005c).

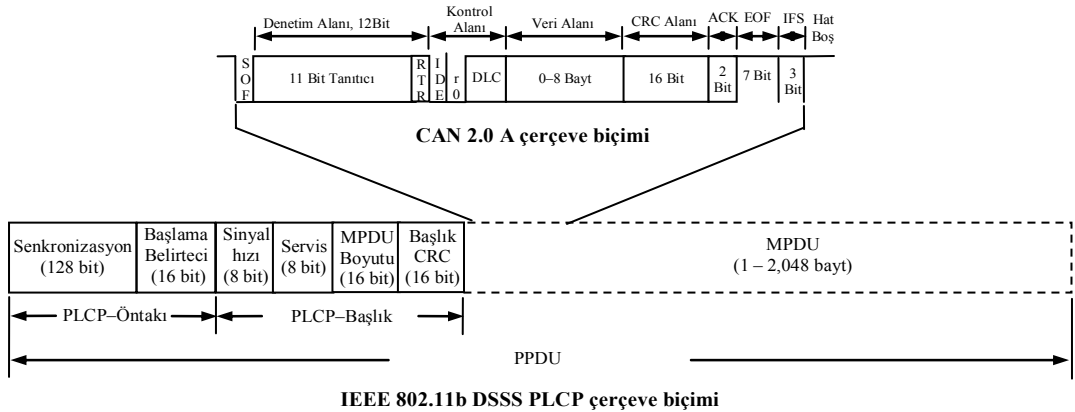


Şekil 4.8: CAN/IEEE 802.11b KAB OSI katmanlı mimarisi

4.4.4. KAB'da CAN mesajların kapsüllemesi

Kablosuz Arabağlaşım Biriminin temel fonksiyonu, CAN veriyolundan alınan uzak CAN 2.0A mesajlarını, hedef CAN segmente gönderilmek üzere kablosuz kanallar üzerinden taşınan IEEE 802.11b DSSS çerçeveleri içerisine kapsüllemektir (encapsulation). Hedef KAB'da ise kablosuz ortamdan alınan IEEE 802.11b DSSS çerçevesi (eğer o KAB'ı adresliyorsa) içerisinden uzak CAN mesajını çıkartıp (decapsulation) CAN veriyoluna iletmektir.

Şekil 4.9'da görüldüğü gibi CAN 2.0A mesajının büyüklüğünün 108 bit olmasından dolayı kolaylıkla bir IEEE 802.11b DSSS çerçevesinin MPDU (1–2,048 bayt) alanına yerleştirilebilir. Bu nedenle bir CAN 2.0A mesajının, IEEE 802.11b DSSS çerçevesi içerisinde taşınabilmesi için parçalama/tekrar birleştirme ya da sıkıştırma/açma gibi işlemlere gerek kalmaz.



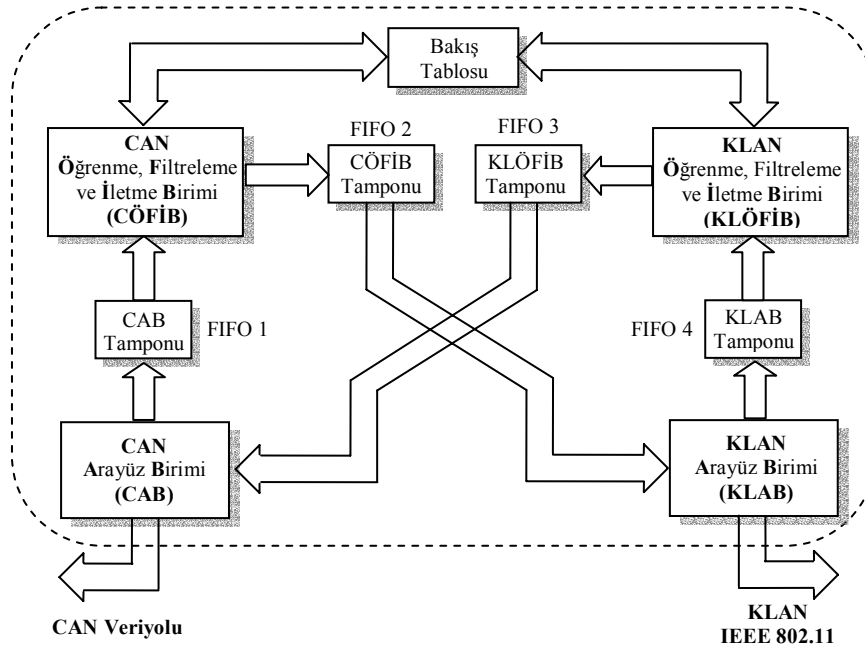
Şekil 4.9: IEEE 802.11b DSSS çerçevesi içerisine CAN 2.0A çerçevelerinin kapsüllemesi

4.4.5. CAN/IEEE 802.11b KAB modeli

Tasarlanan KAB modeli, gerçekleştirme aşamasında farklılıklar gösterebilir. Ancak çalışma mekanizması ve yerine getireceği işlevler temelde aynı olacaktır. KAB'ın başarımının yüksek olabilmesi için kullanılacak birimlerin, bu birimlerin yürüteceği görevlerin ve birbirleri ile olan ilişkilerinin doğru ve tam olarak belirlenmesi gerekir.

CAN/IEEE 802.11b KLAN KAB'ın işlevsel modeli Şekil 4.10'da görülmektedir (Bayılmış ve diğ., 2004c, 2005a, 2005c). Tasarlanan model birbirinden bağımsız dokuz ayrı birimden oluşmaktadır. CAN/IEEE 802.11b KLAN KAB'ın mimarisi iki aşamada incelenebilir:

- KAB'ın her bir biriminin yapısı ve işlevi,
- KAB'ın çalışma prosedürü (CAN'den KLAN'a ve KLAN'dan CAN'e iletim).



Şekil 4.10: CAN/IEEE 802.11b KLAN KAB işlevsel şeması

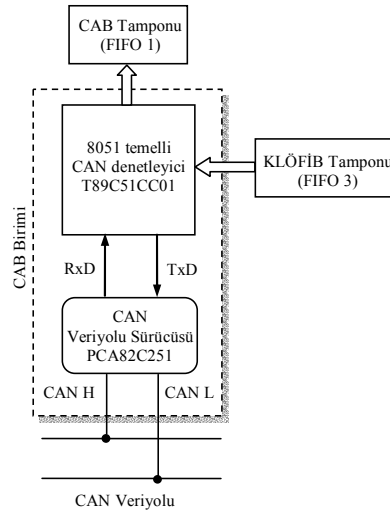
4.4.5.1. KAB'ı oluşturan birimlerin işlevleri

Şekil 4.10'da işlevsel modeli verilen KAB, dokuz ayrı birimden meydana gelmektedir. Bu birimlerin çalışması aşağıda kısaca açıklanmaktadır.

CAN Arayüz Birimi (CAB):

KAB'ın CAN veriyolu ile haberleşmesini sağlayan CAB, standart CAN alıcı/vericisine sahiptir. Bu birim CAN veriyolu üzerinden CAN çerçeve kabulü/iletimi, CRC işlemi, öncelik mekanizmasının çalışması gibi CAN standardının tüm fonksiyonlarını yerine getirir. CAB birimi, CAN veriyolu üzerindeki yerel/uzak CAN mesajlarını alır ve CAB tamponuna yazar. Aynı zamanda KAB'ın KLAN portu vasıtasıyla alınan ve KLÖFİB (KLAN Öğrenme, Filtreleme ve İletme Birimi) tamponunda kayıtlı CAN mesajlarını alarak veriyoluna iletir.

CAB birimi, 8051 temelli bir CAN denetleyici yongası (T89C51CC01) ve bu yonganın CAN veriyolu ile haberleşmesini sağlayan CAN alıcı/verici (PCA 82C251 transceiver) yongasından meydana gelir. Şekil 4.11'de CAB birimi ve CAB/KLÖFİB tamponları ile bağlantısı görülmektedir (Özçelik, 2002).



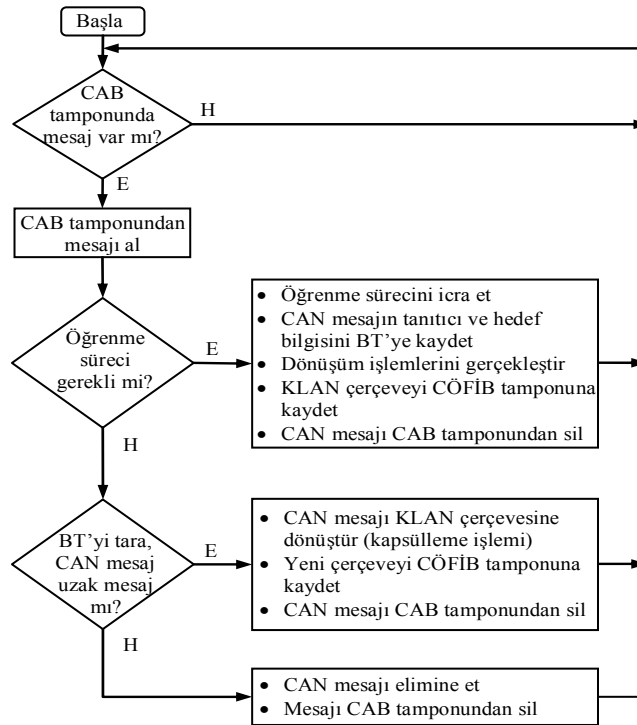
Şekil 4.11: CAB birimi ve tamponlara bağlantısı

CAB Tamponu (FIFO1):

CAB birimi vasıtasıyla CAN veriyolundan alınan CAN mesajların kaydedildiği tampondur. KAB içerisindeki tüm tamponlar, yüksek başarımlı için ilk giren ilk çıkar (First Input First Output, FIFO) tampon yapısındadır.

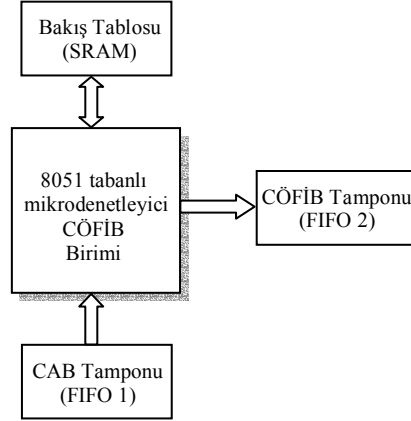
CAN Öğrenme, Filtreleme ve İletme Birimi (CÖFİB):

KAB'ın CAN tarafından KLAN tarafına CAN mesajının öğrenilmesi, filtrelenmesi ve iletilmesi işlemlerini kapsar. En önemli fonksiyonu, filtreleme işlemi için gerekli Bakış Tablosunu (BT, Look-up Table) öğrenme süreçlerinde oluşturması ve güncellemesidir. CÖFİB, CAB birimi vasıtasıyla CAN veriyolundan alınarak CAB tamponuna kaydedilmiş CAN mesajların tanıtıcı alanlarını, bakış tablosu ile karşılaştırarak uzak CAN mesajlarını belirler. Eğer alınmış CAN mesajı, uzak bir CAN mesajı ise gerekli çerçeve dönüştürme işlemlerini gerçekleştirerek (KLAN çerçeve içerisinde kapsülleme işlemi) kablosuz ortama iletmek üzere CÖFİB Tamponu'na gönderir. Aksi takdirde, CÖFİB CAN mesajını elimine eder (Şekil 4.12).



Şekil 4.12: CÖFİB biriminin genel çalışması

CÖFİB, tüm bu belirtilen işlemleri gerçekleştirebilmek için CAB tamponu, CÖFİB tamponu ve Bakış Tablosu ile haberleşmektedir. Bu yapı, CÖFİB olarak 8051 tabanlı bir mikrodenetleyici ve Bakış Tablosu olarak da bir SRAM kullanılarak gerçekleştirilebilir. Şekil 4.13’de böyle bir yapının blok modeli verilmektedir.



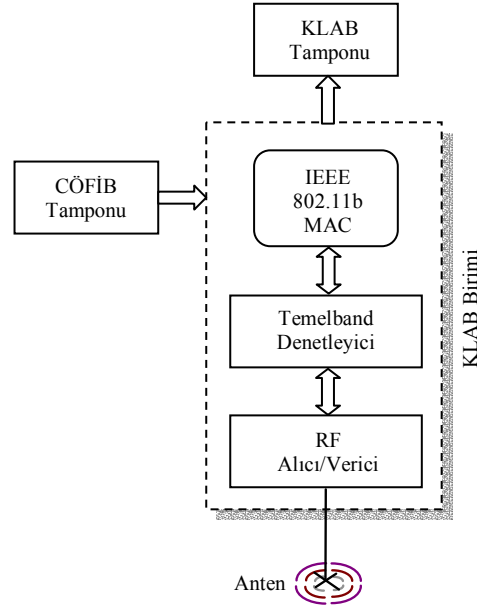
Şekil 4.13: CÖFİB biriminin bakış tablosu ve tamponlarla bağlantısı

CÖFİB Tamponu (FIFO 2):

Uzak CAN mesajların, kablosuz ortama iletilmek üzere CÖFİB birimi tarafından gerekli dönüşüm işlemleri gerçekleştirildikten sonra KLAN çerçeve biçiminde tutuldukları tampondur.

KLAN Arayüz Birimi (KLAB):

Kablosuz alıcı/vericiye sahip olan bu birim, kablosuz ortam üzerinden haberleşme için gerekli fonksiyonları sağlar. Kablosuz ortam üzerinden KLAN çerçevelerini alır ve KLAB tamponuna kaydeder. Aynı zamanda kablosuz ortama gönderilmek üzere KLAN çerçeve biçimine dönüştürülmüş (encapsulation) CÖFİB tamponunda tutulan uzak CAN mesajlarını kablosuz ortama gönderir. KLAB birimi, IEEE 802.11b standardının tüm işlevlerini yerine getirir. Şekil 4.14’de KLAB biriminin işlevsel yapısı ve tamponlarla olan bağlantısı görülmektedir. Gerçekleme aşamasında 2,4 GHz’de çalışan LANTRONIX firmasının IEEE 802.11b/g fiziksel katmanlı Wiport ürünü kullanılmıştır (Wiport Guide, 2006).



Şekil 4.14: CAB biriminin işlevsel yapısı ve tamponlar ile bağlantısı

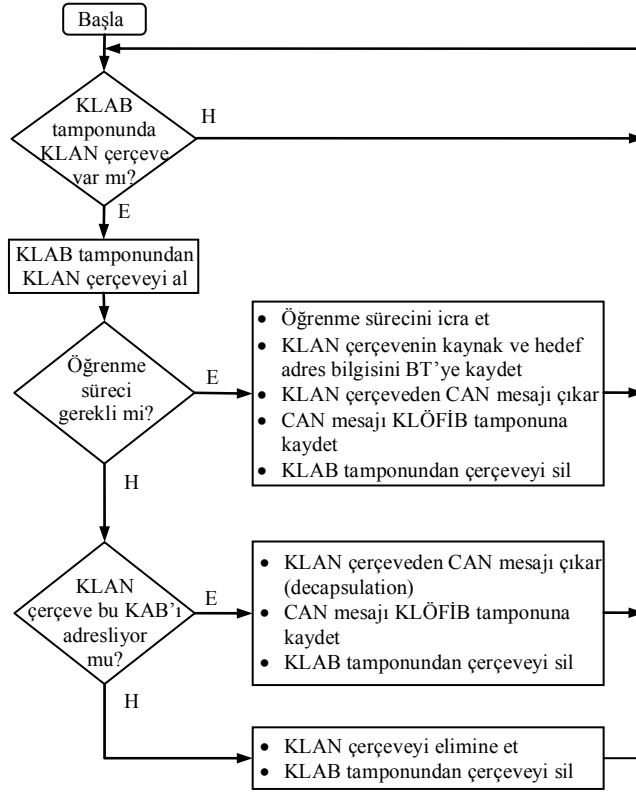
KLAB Tamponu (FIFO4):

KLAB birimi üzerinden alınan IEEE 802.11b KLAN çerçevelerin kaydedildiği tampondur.

KLAN Öğrenme, Filtreleme ve İletme Birimi (KLÖFİB):

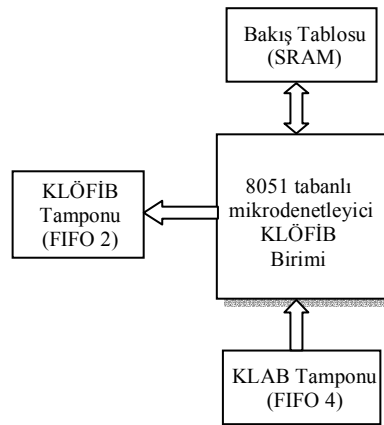
KAB'ın KLAN tarafından CAN tarafına KLAN çerçevesinin öğrenilmesi, filtrelenmesi ve iletilmesi işlemlerini kotarır. KLAB tamponunda tutulan bu KAB'ı hedefleyen KLAN çerçevelerin içerisine kapsüllenmiş olan CAN mesajlarını çıkartır ve KAB'ın CAN kısmına iletmek üzere KLÖFİB tamponuna kaydeder.

Öğrenme sürecinde, CÖFİB birimi ile birlikte Bakış Tablosunun oluşturulmasını ve güncellenmesini sağlar. Kapsama alanı içerisindeki KAB'ların adreslerini öğrenerek, BT içerisinde uzak CAN mesajların bu adresler ile eşleştirilmesine yardım eder. Şekil 4.15'de KLÖFİB biriminin genel çalışması özetlenmektedir.



Şekil 4.15: KLÖFİB biriminin genel çalışması

CÖFİB birimine benzer olarak KLÖFİB birimi de 8051 tabanlı bir mikrodenetleyici ile gerçekleştirilebilir. Şekil 4.16’da KLÖFİB biriminin bakış tablosu ve KLAB/KLÖFİB tamponları ile bağlantısı görülmektedir.



Şekil 4.16: KLÖFİB biriminin bakış tablosu ve tamponlara bağlantısı

KLÖFİB Tamponu (FIFO 3):

KLAN çerçeveleri içerisinde çıkarılan uzak CAN mesajların CAN veri yoluna iletilmeden önce kaydedildiği tampondur.

Bakış Tablosu (BT) (Look-up Table):

KAB'ın en önemli işlevsel birimidir. CÖFİB ve KLÖFİB birimlerinin filtreleme işlemlerinde kullandığı yerel/uzak CAN mesaj tanıtıcı bilgileri ile kapsama alanı içerisindeki kablosuz cihazların ya da diğer KAB'ların KLAN adreslerinin tutulduğu bellektir. Bakış tablosu olarak gerçekleştirme aşamasında SRAM kullanılabilir.

4.4.5.2. KAB'ın çalışma prosedürü

KAB'ın her bir portu farklı protokol, çerçeve/mesaj biçimi ve çerçeve/mesaj alma/iletme mekanizmasına sahiptir. Bundan dolayı her bir portunda gerçekleştirilen süreçler farklıdır. KAB, CAN tarafından KLAN tarafına ve KLAN tarafından CAN tarafına olmak üzere iki veri transfer sürecine sahiptir. Her iki yöndeki veri transferi öğrenme ve normal proses zamanı işlemlerinden oluşur.

Öğrenme proses zamanları sistem devreye alındığında ya da yeni bir eleman eklenip/çıkarıldığında gerçekleştirilir. Öğrenme prosesinde KAB'ın bir portundan diğer portuna aktarılacak uzak mesajların belirlenmesi, filtreleme ve çerçeve dönüşümü gibi işlemler için kullanılan Bakış Tablosu oluşturulur. BT oluşturmak için statik veya dinamik yaklaşımlar kullanılır. Statik yaklaşımda BT önceden belirlenir (hangi düğümün hangi mesajı üreteceği, hangi uzak/yerel düğümü hedeflediği gibi bilgiler) ve kayıtlar statik bir bellekte tutulmaktadır. Bu yaklaşım, yeni düğüm ekleme/çıkarma gibi durumları kontrol edememe ve tanıyamama gibi dezavantajlara sahip olsa da genellikle endüstriyel üretim ortamında hangi cihazın ne iş yapacağı, kiminle nasıl haberleşeceği bilindiğinden çoğunlukla tercih edilmektedir. Dinamik yaklaşım ise, sistem çalışırken yeni eklenen düğümlerin veya mesajların basit bir öğrenme prosesi sonucunda, BT'yi dinamik olarak oluşturması ve güncellemesidir.

KAB'ın tasarımında ve prototipinin kullanıldığı uygulamalarda BT, statik yaklaşımla oluşturulmuştur. Fakat BT'nin oluşturulmasında dinamik yaklaşım da kullanılabilir. Bu durumda BT, KAB'ın CÖFİB birimi vasıtasıyla CAN segmentteki yerel/uzak mesajlar tespit edilerek oluşturulan CAN_BT, KAB'ın KLÖFİB birimi vasıtasıyla kapsama alanları içerisindeki kablosuz cihazların adreslerinin tutulduğu KLAN_BT alt bakış tabloların birleşmesinden oluşur.

Bununla birlikte KAB içerisinde gerçekleştirilen işlemler paralel çalıştırılabilecek şekilde tasarlanmıştır. Örneğin CAN tarafından KLAN tarafına bir mesaj transferi yapılırken, eşzamanlı olarak KLAN tarafından CAN tarafına da bir çerçeve transferi gerçekleştirilebilir.

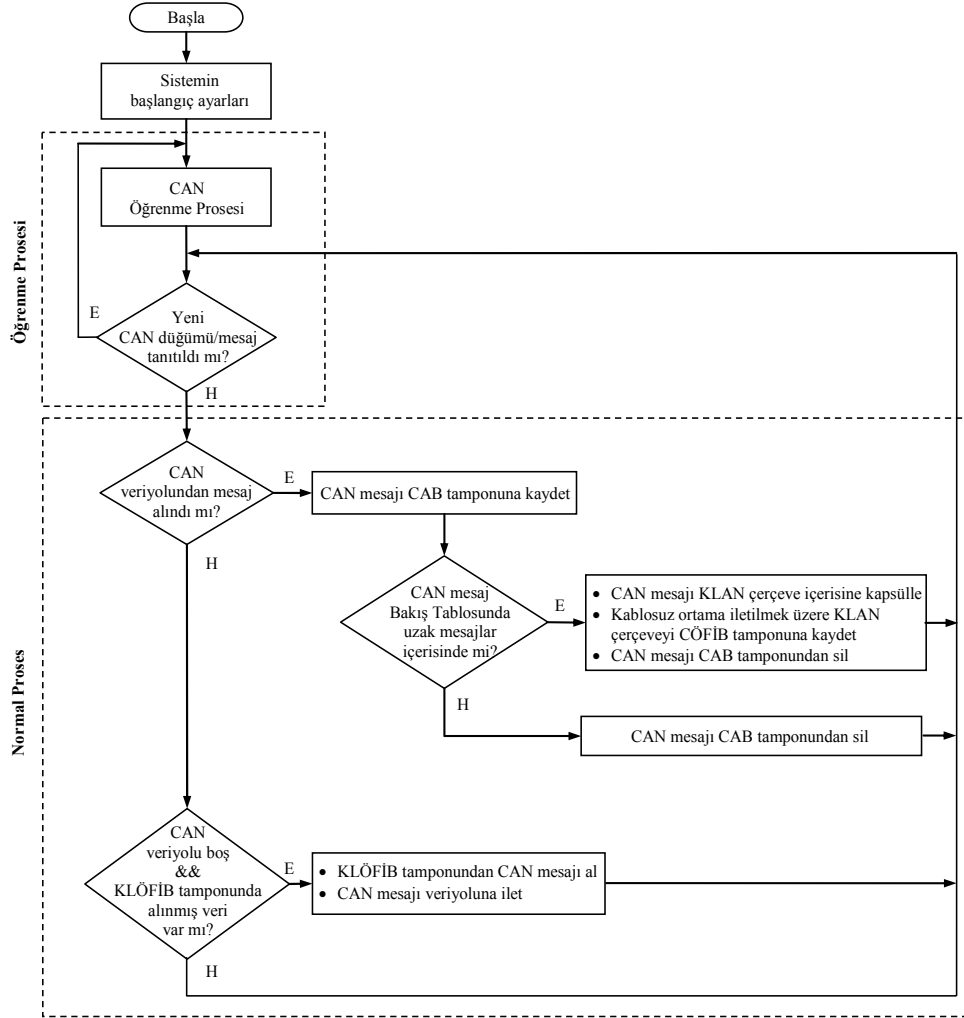
CAN–KLAN / KLAN–CAN tarafına veri transfer süreci:

CAN öğrenme prosesi, CAN tarafından KLAN tarafına gönderilen uzak CAN mesajların tespit edilerek Bakış Tablosunun oluşturulmasını sağlar. Bakış Tablosu, KAB'ın CAN mesajlarını filtrelemesinde ve uzak CAN segmentlere gönderilirken gerekli dönüşüm işlemlerinde kullanılır. Sisteme yeni bir düğüm ya da yeni bir CAN mesaj tanıtıldığında öğrenme süreci tekrar edilerek Bakış Tablosu güncellenir.

CAN normal veri transfer prosesinde, CAB birimi vasıtasıyla CAN veriyolundan alınan mesajlar önce CAB tamponuna kaydedilir. Daha sonra CÖFİB birimi CAB tamponundaki mesajın tanıtıcı bilgisini Bakış Tablosundaki tanıtıcı bilgileri ile karşılaştırarak mesajın bir uzak mesaj olup olmadığını kontrol eder. Eğer CAB tamponundaki mesaj, uzak CAN mesajı ise kapsülleme metodu ile KLAN çerçeve biçimine dönüştürülerek kablosuz ortama iletmek üzere CÖFİB tamponuna gönderilir. Eğer veriyolundan alınan CAN mesajın tanıtıcı bilgisi BT içerisinde yoksa CAN mesajı yerel bir mesajdır. Bu durumda mesaj, CAB tamponundan silinir.

Aynı zamanda KLÖFİB tamponunda kablosuz ortamdan alınan KLAN çerçevelerin içerisinde çıkarılmış CAN mesajları var ve CAN veriyolu boşsa, CAB birimi KLÖFİB tamponundan CAN mesajı olarak CAN veriyoluna iletir. İletimi tamamladıktan sonra KLÖFİB tamponundan CAN mesajı siler.

Yukarıda anlatılan tüm işlemler, Şekil 4.17’deki CAN tarafından KLAN tarafına veri transfer süreci akış diyagramında özetlenmektedir.



Şekil 4.17: CAN–KLAN / KLAN–CAN tarafına veri transfer süreci akış diyagramı

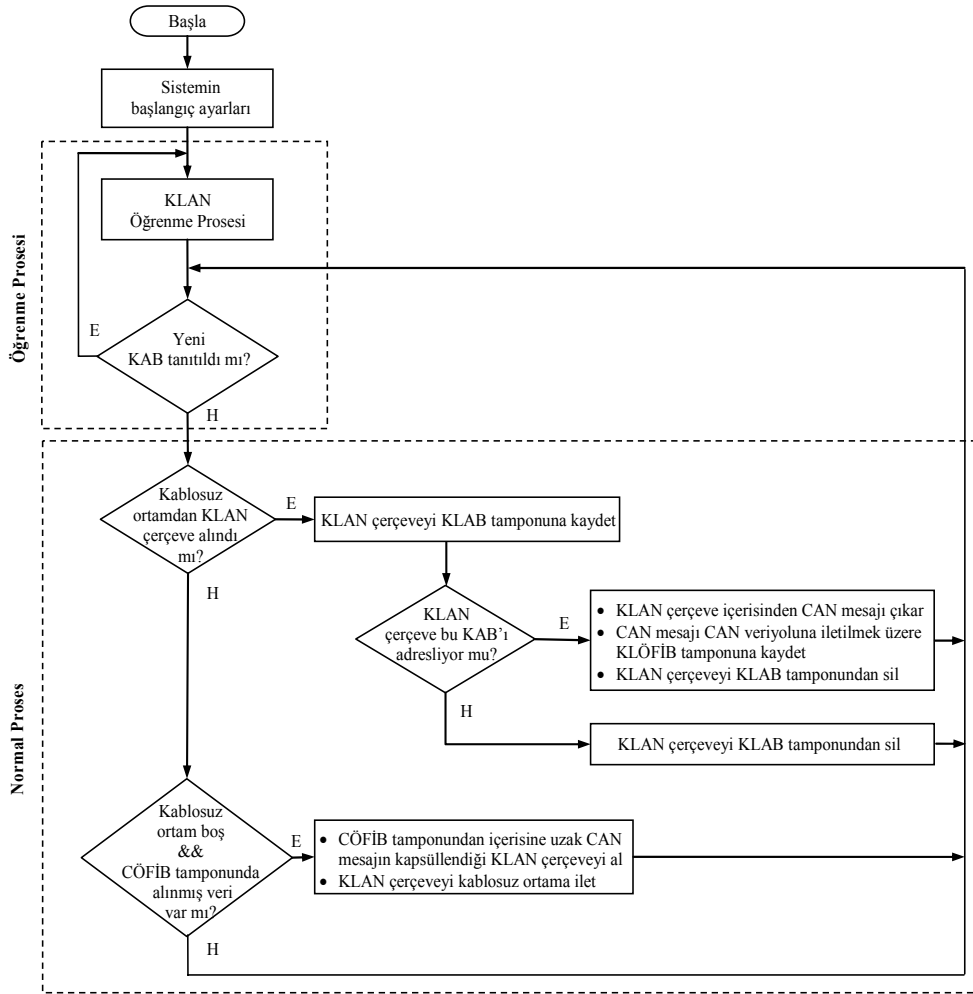
KLAN–CAN / CAN–KLAN tarafına veri transfer süreci:

KLAN öğrenme prosesinde, kapsama alanındaki tüm KAB’ların KLAN adresleri öğrenilerek uzak CAN mesajlarla eşleştirilmek üzere Bakış Tablosuna kaydedilir. Böylelikle KAB’ın genel bakış tablosunun oluşumu tamamlanır. Sisteme yeni bir KAB eklendiğinde öğrenme prosesi tekrarlanır.

KLAN normal veri transfer prosesinde, KLAB birimi vasıtasıyla kablosuz ortamdan alınan mesajlar KLAB tamponuna kaydedilir. Ardından bu KAB’ı adresleyen KLAN

çerçeve içerisinde kapsüllenmiş olan CAN mesajı çıkartılarak CAN veriyoluna iletmek üzere KLÖFİB tamponuna kaydedilir. İçerisinden CAN mesajı çıkartılan KLAN çerçeve, KLAB tamponundan silinir. Eğer KLAN çerçeve bu KAB'ı adreslemiyorsa KLAB tamponundan silinir.

Kablosuz ortama uzak CAN mesajı gönderme işlemi ise, CÖFİB tamponunda alınmış veri var ve kablosuz ortam boş olduğunda, KLAB birimi CÖFİB tamponundan içerisine uzak CAN mesajların kapsüllendiği KLAN çerçeveleri alarak, kablosuz ortama iletmesi ile gerçekleşir. Kablosuz ortama iletim yayın (broadcast) modunda yapılır. Yukarıda anlatılan tüm işlemler, Şekil 4.18'deki KLAN tarafından CAN tarafına veri transfer süreci akış diyagramında özetlenmektedir.



Şekil 4.18: KLAN-CAN / CAN-KLAN tarafına veri transfer süreci akış diyagramı

4.5. Sonuç

BBİ modeline uygun olarak endüstriyel bir tesis, en alt seviyedeki üretim ortamından en üst seviyedeki yönetim ortamına hiyerarşik bir yapıya sahiptir. Doğal olarak bu mimari içerisinde, üretim seviyesindeki cihazların birbirleri arasında (CAN, PROFIBUS vb. segmentlerin) ve yönetim seviyesindeki bilgisayarlar (Ethernet, Token Bus/Ring, ATM vb) ile haberleşme ihtiyacı vardır.

BBİ modelindeki farklı seviyeler, arabağlaşım birimleri vasıtasıyla birbirlerine bağlanabilirler. Ancak hiyerarşik yapıdaki bir bağlantı, kritik zaman uygulamalarında yeterli değildir. Yine iki sahayolu sisteminin birbirine doğrudan bağlanamaması ister istemez ek bir gecikmeye sebebiyet vermektedir. Bu gecikme sürelerini azaltmak ve sahayolu sistemlerinin birbirleri ile ve diğer yerel alan ağları ile haberleşmesini sağlamak için, sahayolu sistemleri kablolu ya da kablosuz omurgaya bağlanırlar.

Endüstriyel ortamlarda kablolu omurga kullanımı, kablolama zorlukları, endüstriyel kontrol ortamının mevcut alt yapısı, kurulu yapıların genişletilmesi gibi nedenlerden dolayı son derece zordur. Kablosuz teknolojiler alanında yaşanan hızlı gelişmeler ile birlikte günümüzde sahayolu sistemlerinin arabağlantısında, sağlayacağı kolaylıklar göz önüne alınarak kablosuz omurga kullanımı önemli bir alternatif teşkil etmektedir.

Endüstriyel ağların arabağlaşımı için birçok farklı kablosuz iletişim standardı kullanımının önerilmesine rağmen, uygulanabilirlik, düşük maliyet, yaygın kullanım ve özellikle kablolu Ethernet yapısının kablosuz ortamdaki uzantısı olması gibi sebeplerden dolayı IEEE 802.11b/g KLAN çoğunlukla tercih edilmektedir.

Bu tez çalışmasında, kapsülleme metodunu kullanarak IEEE 802.11b KLAN üzerinden uzak CAN 2.0A düğümlerini haberleştirme yeteneğine sahip bir CAN/IEEE 802.11b KLAN Kablosuz Arabağlaşım Birimi tasarımı ayrıntılı olarak sunulmaktadır.

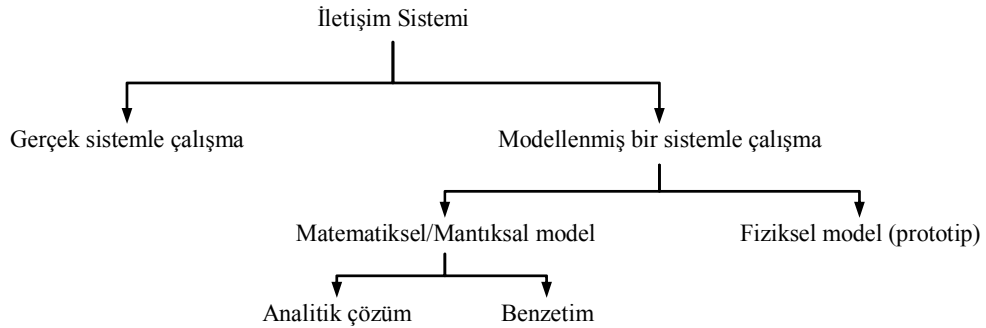
Tasarlanan CAN/IEEE 802.11b KLAN KAB modeli dört ana işlevi yerine getirmektedir. Bunlardan birincisi portlardan CAN mesajlarının ve KLAN

çerçevelerinin alınması, ikincisi, alınan bu mesaj ve çerçevelerin iletilip ileilmeyeceğine karar verilmesi, üçüncüsü, mesaj ve çerçevelerde gerekli olan değişikliklerin yapılması ve son olarak mesajın/çerçevenin diğer ortama iletilmesidir.

5. CAN/IEEE 802.11B KLAN KABLOSUZ ARABAĞLAŞIM BİRİMİNİN MODELLENMESİ VE BAŞARIM ANALİZİ

5.1. Giriş

Tasarımcılar ve araştırmacılar, bir sistemi gerçekleştirmeden önce, yeni sistemin daha ucuz yöntemlerle çalışırılığını ve yeterliliğini ortaya koyarlar. Karmaşık yapıya sahip iletişim sistemlerinde, başta kullanıcı davranışları, iletişim ortamının doğası ve toleranslar olmak üzere başarıyı etkileyen pek çok sebep vardır. İletişim sistemlerinin başarı analiz, gerçek bir sistem ya da modellenmiş eşleniğinin kullanılması ile elde edilebilir (Şekil 5.1). Bir sistemi modellemek için üç farklı yöntem kullanılır. Bunlar; fiziksel model ya da prototip, analitik çözüm ve benzetim (simülasyon) yöntemleridir (Ertürk, 2000, Çeken, 2004, Özçelik, 2002).



Şekil 5.1: İletişim sistemleri geliştirmede kullanılan yöntemler

Fiziksel model yönteminde başarı, mevcut sistem ya da sistemin prototipi değişik koşullar altında incelenerek elde edilebilir. Fiziksel model, en güvenilir ve en doğru yöntem olmasına rağmen, özellikle karmaşık iletişim sistemleri için planlama ve tasarım aşamaları gibi çeşitli konfigürasyonların denenmesinin zorunlu olduğu birçok durumda gerçekleştirilmesi oldukça zordur. Prototip ya da gerçek bir sistemle çalışma pratik olmayan yüksek maliyet ve uzun zaman gerektiren bir yöntemdir (Ertürk, 2000, Özçelik, 2002).

Analitik model, diğer yöntemler arasında basitlik avantajına sahiptir ve genellikle basitleştirilmiş varsayımlar ve ideal kabuller üzerine kurulur. Bu yüzden kesin sonuçlar istendiğinde analitik modeli oluşturmak karmaşıklık ve zaman tüketimi açısından sistemin prototipini oluşturmak kadar zordur. Analitik çözümlerde kullanılan en popüler yaklaşımlardan biri kuyruklama teorisidir (Ertürk, 2000).

Sistem modellemek için kullanılan bir diğer yöntem de benzetimdir. Benzetim somut anlamda belirli bir nesnenin modeli ya da temsilidir. Bir diğer deyişle benzetim, gerçek sistemin modelinin tasarımı ve bu model ile amacına yönelik olarak sistemin işletilmesi, sistemin davranışını anlayabilmek veya değişik stratejileri değerlendirebilmek için deneyler yürütülmesi sürecidir (Bayılmış, 2003). Benzetimin sayısal ortamda bilgisayarla gerçekleştirilmesi ise bilgisayar benzetimi olarak adlandırılır.

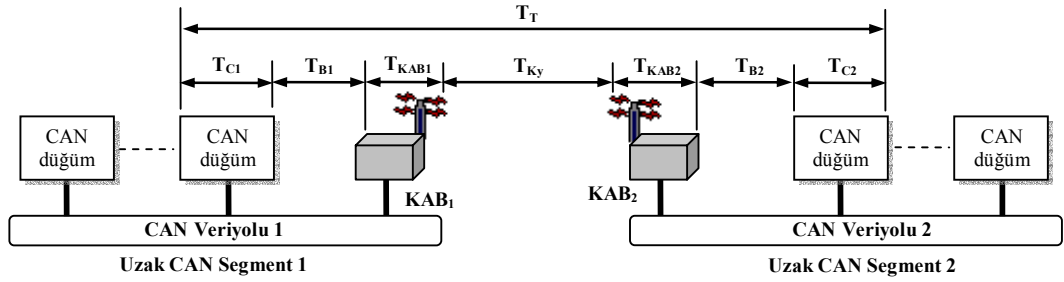
Bir sistemi modellemek için kullanılan yöntemler karşılaştırıldığında sayısal veri haberleşme ağlarının başarımlarında olaya dayalı (event-driven) bilgisayar benzetimi en iyi çözüm olarak görülmektedir (Ertürk, 2000).

Bu tez çalışmasında tasarlanan CAN/IEEE 802.11b KLAN Kablosuz Arabağlaşım Biriminin başarımların analizi için kısmen de olsa yukarıda bahsedilen üç modelleme yöntemi de kullanılmaktadır. Tasarlanan KAB'ın kullanıldığı örnek iki uygulama Bölüm 6'da ayrıntılı olarak sunulmaktadır.

Bu bölümde öncelikle tasarlanan KAB'ın kullanımının önerildiği ağ mimarisinin başarımların analizi için basitleştirilmiş analitik modeli verilmektedir. Daha sonra Bölüm 4'de ayrıntılı tasarımı verilen KAB'ın, OPNET Modeller yazılımı kullanılarak gerçekleştirilen bilgisayar benzetim modeli sunulmaktadır. Son olarak OPNET yazılımında modellenen KAB'ın kullanıldığı bir endüstriyel ağ örneğinin SAE Benchmark'a göre başarımların analizi verilmektedir.

5.2. Basitleştirilmiş Analitik Model

Bu bölümde KAB'ın kullanımının önerildiği ağ mimarisinin indirgenmiş analitik modeli verilmektedir. Şekil 5.2'de görüldüğü gibi analitik model, CAN düğümlerden oluşan iki bağımsız CAN segmenti, bu CAN segmentleri IEEE 802.11 KLAN üzerinden haberleşiren iki KAB'ı ve fiziksel iletim ortamını kapsamaktadır. Analitik model içerisinde tüm bu birimlerin zamanlama karakteristikleri verilmektedir.



Şekil 5.2: CAN/IEEE 802.11b/CAN ağdaki gecikmeler için örnek bir model

Bir segmentten diğer segmente hedeflenen uzak bir CAN mesajın uçtan uca gecikmesi aşağıdaki alt gecikme bileşenlerinden oluşmaktadır.

- Uzak CAN mesajın kaynak CAN düğümünde oluşturulması, kuyruklama ve CAN veriyoluna erişimi için geçen gecikme süresi (T_{C1}),
- Kaynak CAN veriyolundaki yayılım gecikmesi (T_{B1}),
- KAB'da mesajın alımı, filtrelenmesi, kapsülleme işlemi (encapsulation), kuyruklanması ve kablosuz ortama erişim için geçen gecikme (KAB'ın CAN tarafından KLAN tarafına işlem süresi, T_{KAB1}),
- Kablosuz iletim ortamından kaynaklanan yayılım gecikmesi (T_{Ky}),
- Hedef segmentteki KAB'ın kablosuz ortamdan KLAN çerçevesi alması, filtrelenmesi, KLAN çerçeve içerisinden uzak CAN mesajı çıkartması (decapsulation), kuyruklama ve CAN veriyoluna erişim için geçen gecikme süresi (KAB'ın KLAN tarafından CAN tarafına işlem süresi, T_{KAB2}),
- Hedef CAN veriyolundaki yayılım gecikmesi (T_{B2}),
- Hedef CAN düğümünde uzak CAN mesajın alınması (T_{C2}).

Yukarıda ifade edilen gecikme bileşenlerine göre uzak bir CAN mesajın uçtan uca gecikmesi şu şekilde formülize edilebilir:

$$T_T = T_{C1} + T_{B1} + T_{KAB1} + T_{Ky} + T_{KAB2} + T_{B2} + T_{C2} \quad (5.1)$$

Bölüm 2.3.7’de CAN veriyolu üzerinde CAN mesajların gecikme zaman analizi ayrıntılı olarak verilmektedir.

5.3. Modelleme, Benzetim ve OPNET Yazılımı

Modelleme ve benzetim tasarımcılara, bir sistemi gerçekleştirmeden önce sistemin çalışırılığının görülmesini, kullanıcı sayısı, ağ teknolojileri ve ağ konfigürasyonu gibi parametrelerin değişiminde sistemin göstereceği davranışın tespit edilmesini ve karşılaştırmalı değerlendirme imkanı sağlar.

Bir sistemin genel karakteristiklerinin gösterilimi, sistemin modeli olarak adlandırılır. Bir sistemin ya da prosesin matematiksel, algoritmik veya davranışsal karakteristiklerinin açıklanmasına ise modelleme denir (Ertürk, 2000, Özçelik, 2002). Benzetim, teorik olarak ya da gerçek bir sistemin modelinin tasarımı, bu model kullanılarak sistemin işletilmesi ve sistemin davranışının izlenmesi sürecidir. Bilgisayar benzetimi ise benzetim işleminin sayısal ortamda bilgisayarlarla gerçekleştirilmesidir.

Benzetim sonuçlarının doğruluğu, modelin gerçek sisteme yakınlığı ile doğru orantılıdır. Gerçeğine daha yakın model daha fazla ayrıntı gerektirir. Buna paralel olarak modelin tasarlanması ve benzetiminin gerçekleştirilmesi daha uzun sürer (Çeken, 2004).

Bilgisayar benzetiminde, sistemin doğasına bağlı olarak sürekli olay (continuous-event) ve ayrık olay (discrete-event) olmak üzere iki farklı benzetim tekniği kullanılabilir. Sürekli olay benzetim tekniğinde, olaylar zamana bağlı olarak bir süreklilik arz eder. Zaman doğrusal olarak arttıkça, prosesler de zaman içerisinde doğrudan değişirler. Sayısal veri iletişim sistemleri ve bilgisayar ağları, mesajların

üretimi ve dağıtımı gibi durumlar gerçekleştiğinde modüllerin çalıştırıldığı ve benzetim saatinin ilerlediği ayrık olay benzetim yöntemi ile modellenir.

Tablo 5.1’de iletişim sistemlerinin ve haberleşme ağlarının modellenmesinde yaygın olarak kullanılan benzetim yazılımları verilmektedir. Bu tez çalışmasında önerilen KAB’ın modelinin tasarlanmasında ve başarımlarının analizinin gerçekleştirilmesinde OPNET Modeler benzetim yazılımı kullanılmıştır.

Tablo 5.1: İletişim sistemlerini modellemek için yaygın olarak kullanılan benzetim yazılımları

| Ağ Benzetim Yazılımı | Üretici |
|--|---------------------------|
| OPNET Modeler | OPNET Inc. |
| OMNET | Açık kaynak kodlu yazılım |
| Network Simulator | Berkley University |
| Planning Workbench Suite | Bellcore |
| BONES | Comdisco |
| Network II.5, COMNET III, L_NET ve Simscript 2.5 | CACI Products Co. |
| Net Maker | Make Systems Inc. |
| Object Time | BNR |

5.3.1. OPNET benzetim yazılımı

OPNET Modeler, iletişim sistemleri ve haberleşme ağlarının modellenmesi için görsel bir benzetim ortamı sağlayan nesneye yönelik bir programdır. Modellenen sistemlerin davranış ve başarımlarının analizleri ayrık olay benzetim yöntemi ile gerçekleştirilir. OPNET hiyerarşik modelleme katmanlarından oluşur. Windows NT/2000/XP ya da UNIX altında çalışabilir.

OPNET, sistemi modelleme, veri toplama, benzetimi gerçekleştirme ve sonuçları analiz etme işlemlerini yerine getirir. ATM, TCP/IP, MPLS gibi ağ protokollerinin, 3Com, Cisco, Bays Network gibi firmaların yönlendirici, anahtar gibi ürünlerinin modellerini içeren geniş bir kütüphaneye sahiptir. OPNET’in en büyük avantajı editörlerinin yardımı ile yeni protokol ve ürünlerin modellerini oluşturabilmesi ve bunların model kütüphanesine eklenmesine olanak sağlamasıdır.

OPNET yazılımının genel kullanım uygulamaları arasında, standart LAN ve WAN'ların modellenmesi, ağlararası iletişim (internetworking), haberleşme mimarilerinde ve protokollerinde araştırma ve geliştirme, dağıtık algılayıcı ve kontrol ağlarının geliştirilmesi, uydu ve hareketli kablosuz ağlar sayılabilir. OPNET'in kullanıcı profili ve kullanım amacı Tablo 5.2'de verilmektedir (Opnet 2006).

Tablo 5.2: OPNET kullanıcı tipleri ve kullanım amaçları

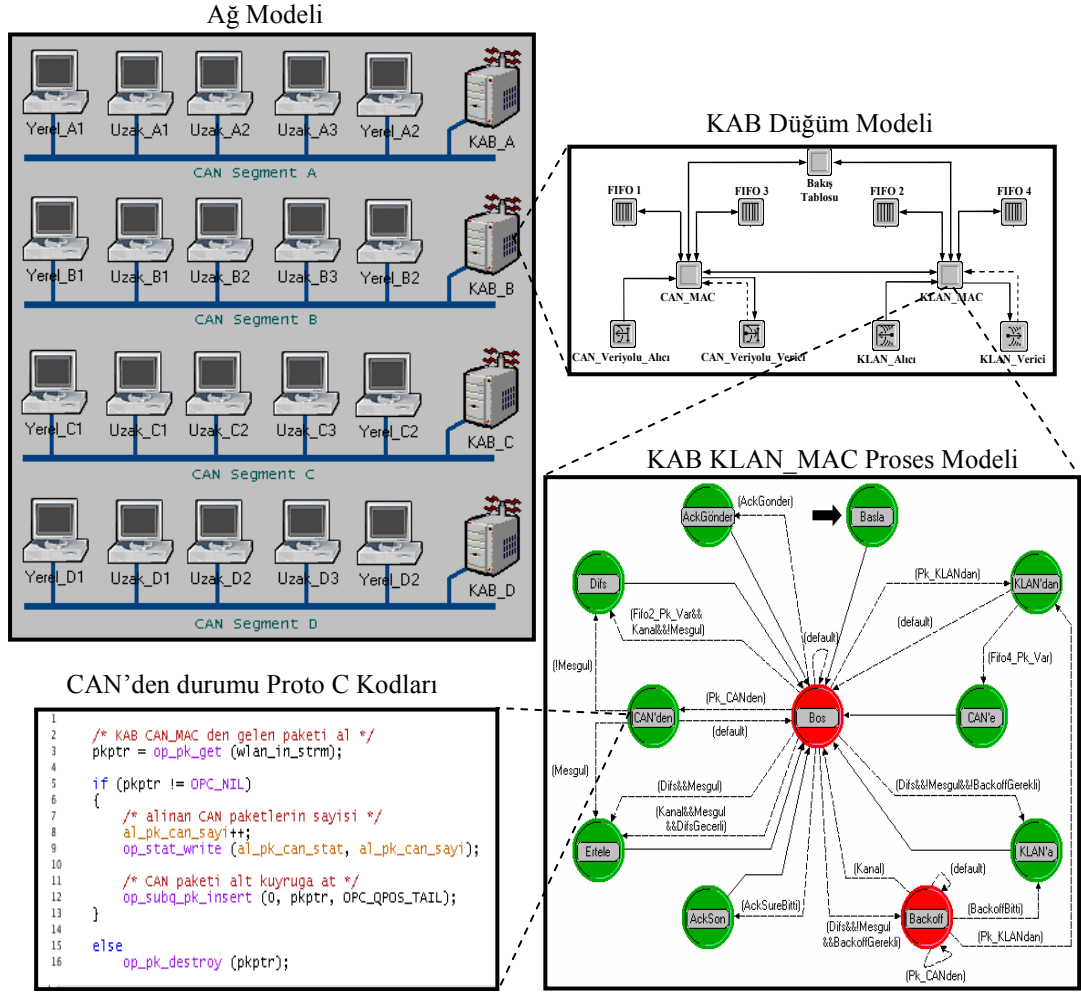
| Kullanıcı Tipi | Kullanım Amacı |
|------------------------|--|
| Servis Sağlayıcılar | Başarım ölçümü, trafik planlaması, ağ yönetimi |
| Ağ Ekipman Üreticileri | Ağ tasarımı, ürünlerin ve protokollerin testi |
| Araştırma Kuruluşları | Ağ ve protokol tasarımı, başarım değerlendirmesi |
| Yatırımcılar | Başarım optimizasyonu, ağ yönetimi |

OPNET Modeller yazılımıyla bir sistemin modellenmesi, birçok editörde yapılan işlemlerin birleştirilmesi ile gerçekleştirilir. Bir ağ modellemesinde, ağ içerisinde kullanılacak bağlantı hatları, veri ve kontrol paketleri, kullanılacak ağ elemanları, katmanlar ve protokoller ayrı editörlerde hazırlanır. OPNET Modeller yazılımında yaygın olarak kullanılan editörler ve görevleri şunlardır:

- Proje editörü (Project editor): Ağ modellerinin geliştirildiği, alt ağların oluşturulduğu, bağlantı hatları, düğümler ve coğrafik içeriğin tanımlandığı editördür. Bu editör temel benzetim ve analiz yeteneklerini içerir.
- Düğüm model editörü (Node model editor): Proje editöründe kullanılan ağ modellerindeki nesnelere (düğümlerin) geliştirildiği editördür. Bu editörde her bir nesne içerisinde proses modellerin olduğu modüller ve modülleri birbirine bağlayan iletim hatlarından oluşur. Düğüm modelindeki modüller, alıcı, verici, kuyruk, işlemci ve üreteçtir.
- Proses model editörü (Process model editor): Düğüm editöründeki nesnelere yapısının, işlevinin, parametrelerinin ve davranışlarının tanımlandığı, kontrol edildiği ve değiştirildiği editördür. Durum geçiş diyagramları ve bunların işlevlerini belirleyen Proto-C kodlarını içerir.

- Bağlantı model editörü (Link model editor): Ağ modellerinin/cihazlarının iletişimini sağlayan bağlantı (veriyolu, ara bağlantı vb.) modellerinin oluşturulduğu ve düzenlendiği editördür.
- Paket biçim editörü (Packet format editor): Veri, kontrol gibi bilgi paketlerinin tanımlandığı ve yapılarının geliştirildiği editördür.
- Arayüz kontrol bilgisi editörü (Interface control information editor, ICI): Prosesler arasında haberleşme kontrol bilgisi tanımlamak için kullanılır.
- Anten model editörü (Antenna pattern editor): Kablosuz modülündeki alıcı/vericiler için anten örnekleri oluşturmak ve düzenlemek için kullanılır.
- Modülasyon editörü (Modulation curve editor): Yalnızca kablosuz modül desteği eklenmiş OPNET Modeler yazılımlarında bulunur. Alıcı/vericiler için modülasyon işlemlerini düzenler.
- PDF editörü (Probability Density Function editor): Olasılık sıklık fonksiyonlarının düzenlendiği editördür. OPNET Modeler yazılımı üssel (exponential), poisson, gamma, uniform, laplace gibi birçok analitik dağılım fonksiyonunu sağlamaktadır.

OPNET Modeler yazılımında sistem modellenmesinde, yukarıda bahsedilen editörlerden genellikle hiyerarşik bir yapı halindeki proses, düğüm ve proje editörleri kullanılır. Modellenmiş bir ağ sistemi ve bu sistemi oluşturan hiyerarşik yapı Şekil 5.3'de görülmektedir. Bu yapıya göre proje editöründe oluşturulan ağ modelindeki her bir nesne, düğüm editöründe tanımlanan modüllerden meydana gelmektedir. Düğüm editöründeki her bir modül ağ elemanının davranışını, kullandığı protokolün çalışmasını ve özelliklerini belirlemektedir. Modülün bu işlevi, proses editörde gerçekleştirilen Proto-C kodu içeren durum geçiş diyagramları ile sağlanır. Proje editör en üst modelleme katmanı iken proses editör en alt modelleme katmanıdır.



Şekil 5.3: OPNET sistem modellerinin tasarımında kullanılan editörlerin hiyerarşik ilişkisi

Uzak CAN segmentlerin IEEE 802.11b KLAN ağlar üzerinden haberleşmesini sağlayan Kablosuz Arabağlaşım Biriminin tasarımında kablosuz haberleşme desteği gerektiğinden OPNET Modeller yazılımına ilave olarak OPNET Kablosuz Modülü de kullanılmıştır. Bu modül kablosuz ağların modellenmesine imkan sağlamaktadır ve kablosuz bağlantıların tüm özelliklerinin (modülasyon, frekans, iletim gecikmesi vb.) düzenlenebileceği açık bir yapı sunmaktadır.

5.4. CAN/IEEE 802.11b KLAN Kablosuz Arabağlaşım Biriminin OPNET Modeli

Kablosuz Arabağlaşım Birimi, uzak CAN segmentleri IEEE 802.11b KLAN ağlar üzerinden haberleştirmeyi amaçlamaktadır. Şekil 5.4’de KAB’ın kullanıldığı örnek bir endüstriyel ağ uygulaması görülmektedir. Böyle bir ağın gerçekleştirilmesi için ağdaki tüm elemanların modellenmesi gerekmektedir. Kullanılan ağ elemanları:

- CAN düğümler (OPNET Modeler yazılımında standart CAN düğüm modeli yoktur.),
- CAN veriyolu,
- Kablosuz Arabağlaşım Birimi.



Şekil 5.4: KAB’ın kullanıldığı örnek bir endüstriyel ağ

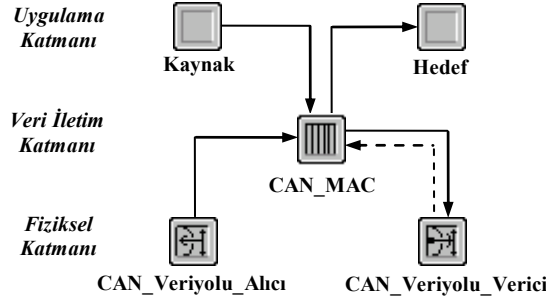
Bu alt bölümde CAN ve KAB elemanlarının OPNET modelleri verilmekte ve bu modeller kullanılarak gerçekleştirilen bir endüstriyel ağın başarımları analizi sunulmaktadır.

5.4.1. CAN modeli

OPNET Modeler yazılımında gerçekleştirilen CAN modeli, CAN mesajların üretilmesi, veriyoluna erişim, veriyolundaki mesajların alınması gibi CAN denetleyicinin tüm fonksiyonlarını yerine getirmektedir. Şekil 5.5’de OPNET yazılımıyla tasarlanan CAN düğüm modeli görülmektedir. Şekilden görüldüğü gibi CAN düğüm modeli, fiziksel, veri iletim ve uygulama katmanlarından meydana gelmektedir. Fiziksel katman, düğümün veriyolundan mesaj almasını/iletmesini sağlayan standart OPNET veriyolu (bus) alıcı ve vericisinden oluşmaktadır. Veri iletim katmanı ise CAN düğümünde kullanılan CSMA/CD+AMP protokolünün

çalışmasını sağlayan tüm işlevleri yerine getirir. Uygulama katmanı, Kaynak ve Hedef modüllerinden oluşmaktadır. Kaynak modülü, paket boyutu ve paket varış süresine bağlı olarak trafik üretilmesinden sorumludur. Hedef modülü ise düğüme gelen paketlerin istatistiksel bilgileri hesaplandıktan sonra yok edilmesini sağlar.

Düğüm modelindeki düz çizgiler, modüllerin birbirleriyle bağlantısını ve haberleşmesini sağlar. CAN_MAC ve CAN_Veriyolu_Verici arasındaki kesik çizgi ise CAN veriyolunun kullanım durumu (meşgul ya da boş) ile ilgili istatistiksel bilgi verir (Bayılmış ve diğ. 2005b).



Şekil 5.5: CAN düğüm (node) modeli

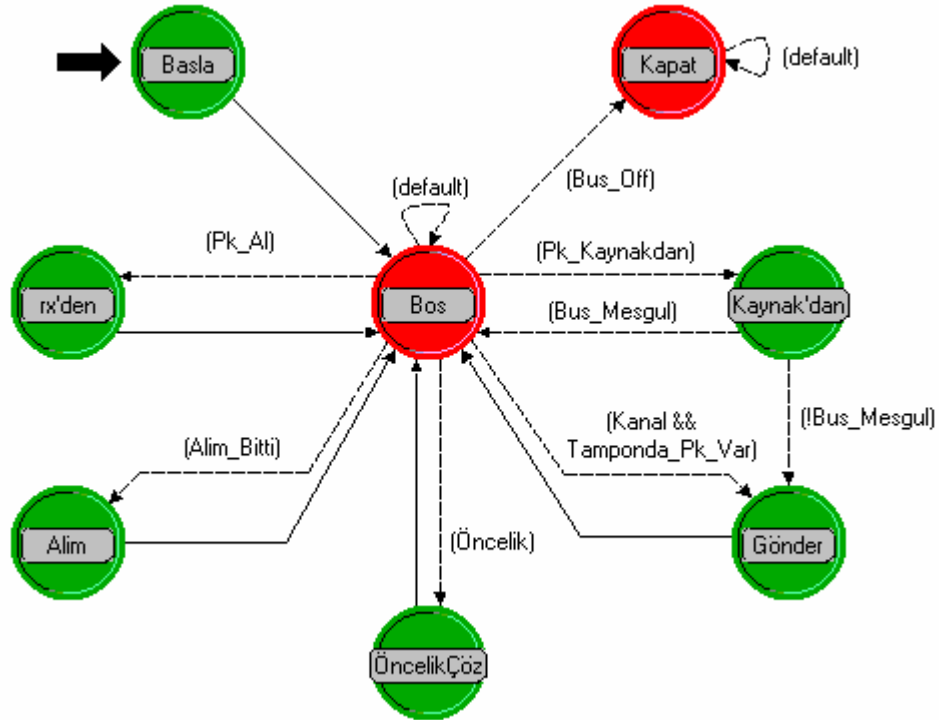
5.4.1.1. CAN proses modeli ve algoritması

Şekil 5.6’da CAN düğüm modelinin içerisinde kullanılan CAN_MAC modülünün (veri iletim katmanı) proses modeli görülmektedir (Bayılmış ve diğ. 2005b). CAN düğümde kullanılan CSMA/CD+AMP protokolünün işlevini yerine getiren bu model 8 durum makinesinden oluşmaktadır. Bunlar:

- “Basla” durum makinesi, üretilen mesajın öncelik bilgisi, mesaj alımında kullanılacak filtreleme bilgisi gibi CAN düğümle ilgili başlangıç değerlerini elde eder. Aynı zamanda istatistiksel bilgiler bu durum makinesinde tanımlanmaktadır.
- “rx’den” durum makinesi, veriyolu üzerinden mesaj alımı, CAN mesajın tanıtıcı alan bilgisi ile öncelik (priority) kontrolünü gerçekleştirir. Tanıtıcı bitleri süresince (arbitration) veriyolu üzerindeki tüm mesajlar alınır.

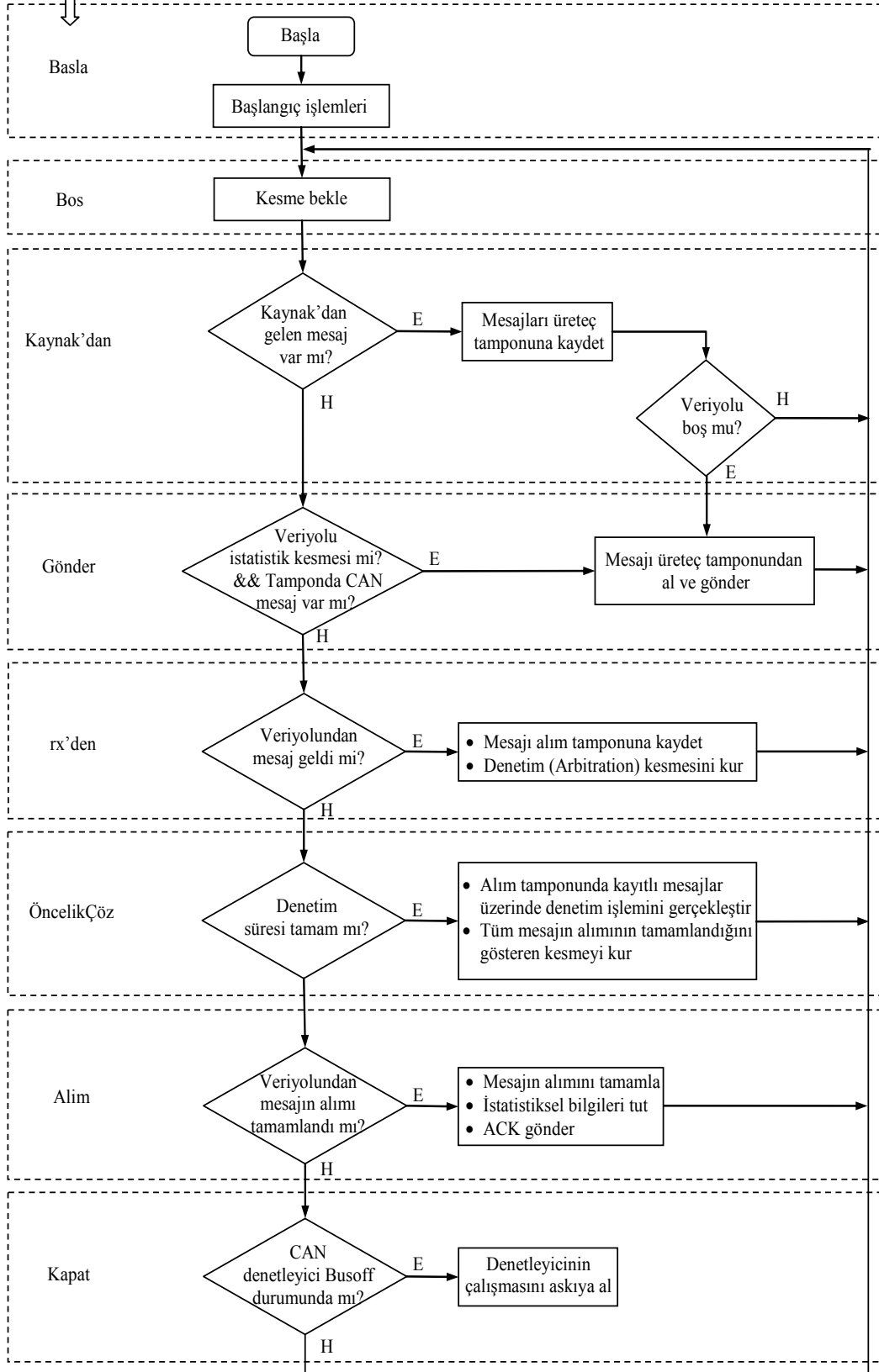
- “Bos” durum makinesi ise model içerisindeki tüm kesme isteklerinin değerlendirilmesini sağlar.
- “ÖncelikÇöz” durum makinesi, rx’den durum makinesi ile alımı başlatılan mesajın başlık alanındaki yük büyüklüğü bilgisine göre mesajın alımının tamamlanması için gerekli gecikmeyi belirten kesmeyi kurar.
- “Alim” durum makinesi ise CAN düğümüne gönderilen en yüksek öncelikli mesajın alımını gerçekleştirir. Veriyolu erişimini kazanan en yüksek öncelikli mesaj bu düğümü adresliyorsa mesaj Hedef modüle gönderilerek istatistiksel bilgileri tutulur, aksi takdirde mesaj elimine edilir.
- “Kaynak’dan” ve “Gönder” durum makineleri, Kaynak modülünden gelen mesajları, gerekli işlemler (mesajın tanıtıcı alanına öncelik bilgisinin yüklenmesi gibi) yapıldıktan sonra veriyolu boş ise iletiminden sorumludur.
- “Kapat” durum makinesi ise CAN hata sayıcılarının durumuna göre CAN düğümünün çalışmasını askıya alır.

CAN_MAC proses modelinde gerçekleştirilen tüm işlemlerin algoritması genel hatları ile Şekil 5.7’de görülmektedir. Ayrıca bu proses modele ait C kodları Ek-A’da verilmektedir.



Şekil 5.6: CAN_MAC modülü proses modeli

İşlemin Gerçekleştiği
Durum Makinesi
(Şekil 5.6)



Şekil 5.7: CAN_MAC modülü proses modeli algoritması

5.4.2. CAN/IEEE 802.11b KAB modeli

Benzer olmayan CAN ve IEEE 802.11b KLAN ağlar arasında arabağlaşımı sağlayan KAB'ın her bir portu farklı protokol, çerçeve/mesaj biçimi ve çerçeve/mesaj alma/iletme mekanizmasına sahiptir. Bundan dolayı tasarlanılan KAB, hem CAN hem de KLAN ağların tüm işlevlerini yerine getirmektedir.

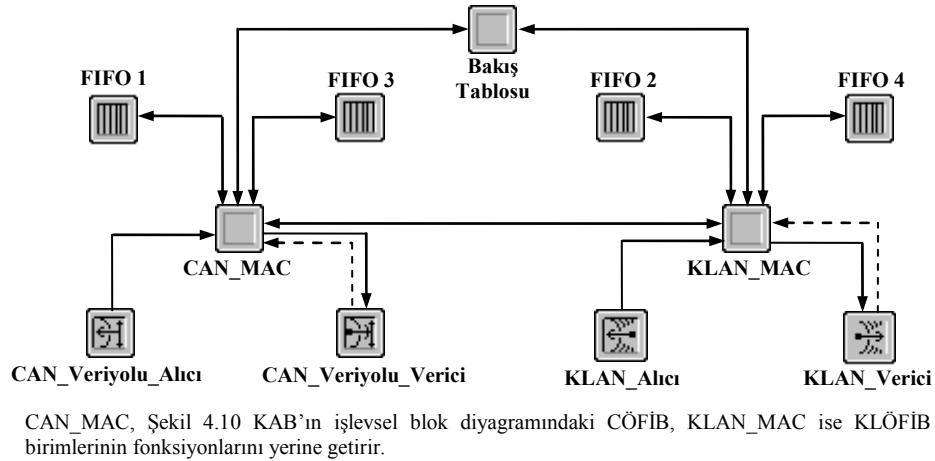
Şekil 5.8'de OPNET yazılımı ile tasarlanan KAB düğüm modeli görülmektedir. KAB'ın düğüm modeli, Bölüm 4'de verilen işlevsel blok diyagramı (Şekil 4.10) esas alınarak oluşturulmuştur. CAN_Veriyolu_Alıcı ve CAN_Veriyolu_Verici KAB'ın CAN veriyolu ile haberleşmesini sağlamaktadır ve işlevsel blok diyagramındaki CAN Arayüz Birimi'ne (CAB) karşılık gelmektedir. Aynı şekilde KLAN_Alıcı ve KLAN_Verici ise KAB'ın IEEE 802.11b KLAN ile haberleşmesini sağlamaktadır ve KLAN Arayüz Birimi'ne (KLAB) karşılık gelmektedir.

CAN_MAC modülü, işlevsel blok diyagramındaki CAN Öğrenme, Filtreleme ve İletme Birimi'ne (CÖFİB) karşılık gelmektedir ve CAN'de kullanılan CSMA/CD+AMP protokolünün çalışması, CAN veriyolundan alınan mesajların filtrelenmesi, öğrenme sürecinde uzak mesajların tespit edilerek Bakış Tablosu'nun oluşturulması, uzak CAN mesajın KLAN çerçeve içerisine kapsüllenmesi (encapsulation), kablosuz ortama iletilmek üzere ilgili tampona iletilmesi, alınan uzak CAN mesajların CAN veriyoluna iletilmesi gibi işlemleri gerçekleştirmektedir.

KLAN_MAC modülü ise IEEE 802.11b KLAN ağda kullanılan CSMA/CA protokolünün çalışması, kablosuz ortamdan alınan KLAN çerçevelerin filtrelenmesi, kendine ait olan çerçeve içerisinden uzak CAN mesajların çıkarılması (decapsulation) ve CAN veriyoluna iletilmek üzere ilgili tampona iletilmesi, öğrenme sürecinde uzak CAN mesajları ile ilgili KAB'ların adreslerinin eşleştirilerek genel Bakış Tablosu'nun oluşturulması, içerisine CAN mesaj kapsüllenen KLAN çerçevelerin kablosuz ortama iletilmesi gibi işlemleri yerine getirmektedir. Bu modül, işlevsel blok diyagramdaki KLAN Öğrenme, Filtreleme ve İletme Birimi'ne (KLÖFİB) karşılık gelmektedir.

KAB düğüm modelinde görülen tamponlar sırasıyla işlevsel blok diyagramındaki şu tamponlara karşılık gelmektedir. FIFO1, CAN veriyolundan alınan CAN mesajların tutulduğu CAB tamponu, FIFO2, uzak CAN mesajların KLAN çerçeve içerisine kapsülendiği ve kablosuz ortama iletmek üzere KLAN çerçevelerinin tutulduğu CÖFİB tamponu, FIFO3, CAN veriyoluna iletmek üzere KLAN çerçeveler içerisinden çıkarılan uzak CAN mesajların tutulduğu KLÖFİB tamponu ve FIFO4 ise kablosuz ortamdan alınan KLAN çerçevelerinin tutulduğu tampondur.

Düğüm modelindeki düz çizgiler, modüllerin birbirleriyle bağlantısını ve haberleşmesini sağlarken, kesik çizgiler ise ilgili ortamın (CAN veriyolu ve kablosuz iletim ortamı) kullanım durumu (meşgul ya da boş) ile ilgili istatistiksel bilgi verir.



Şekil 5.8: KAB düğüm (node) modeli

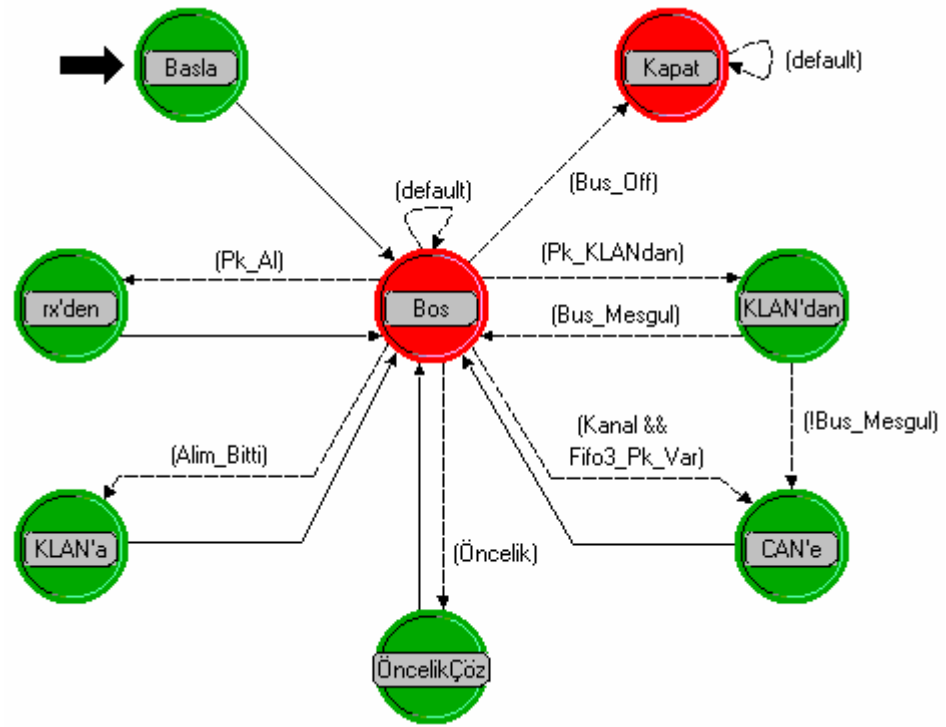
5.4.2.1. KAB CAN proses modeli ve algoritması

Şekil 5.9'da KAB düğüm modeli içerisindeki CAN_MAC modülünün proses modeli görülmektedir. CAN_MAC proses modeli aşağıda kısaca açıklanan 8 durum makinesinden oluşmaktadır.

- “Basla” durum makinesi, CAN veriyolundan mesaj alımında kullanılacak filtreleme bilgisi, CAN düğümle ilgili başlangıç değerleri ve Bakış Tablosu'nun oluşturulması gibi işlemlerden sorumludur. Ayrıca istatistiksel bilgiler bu durum makinesinde tanımlanmaktadır.

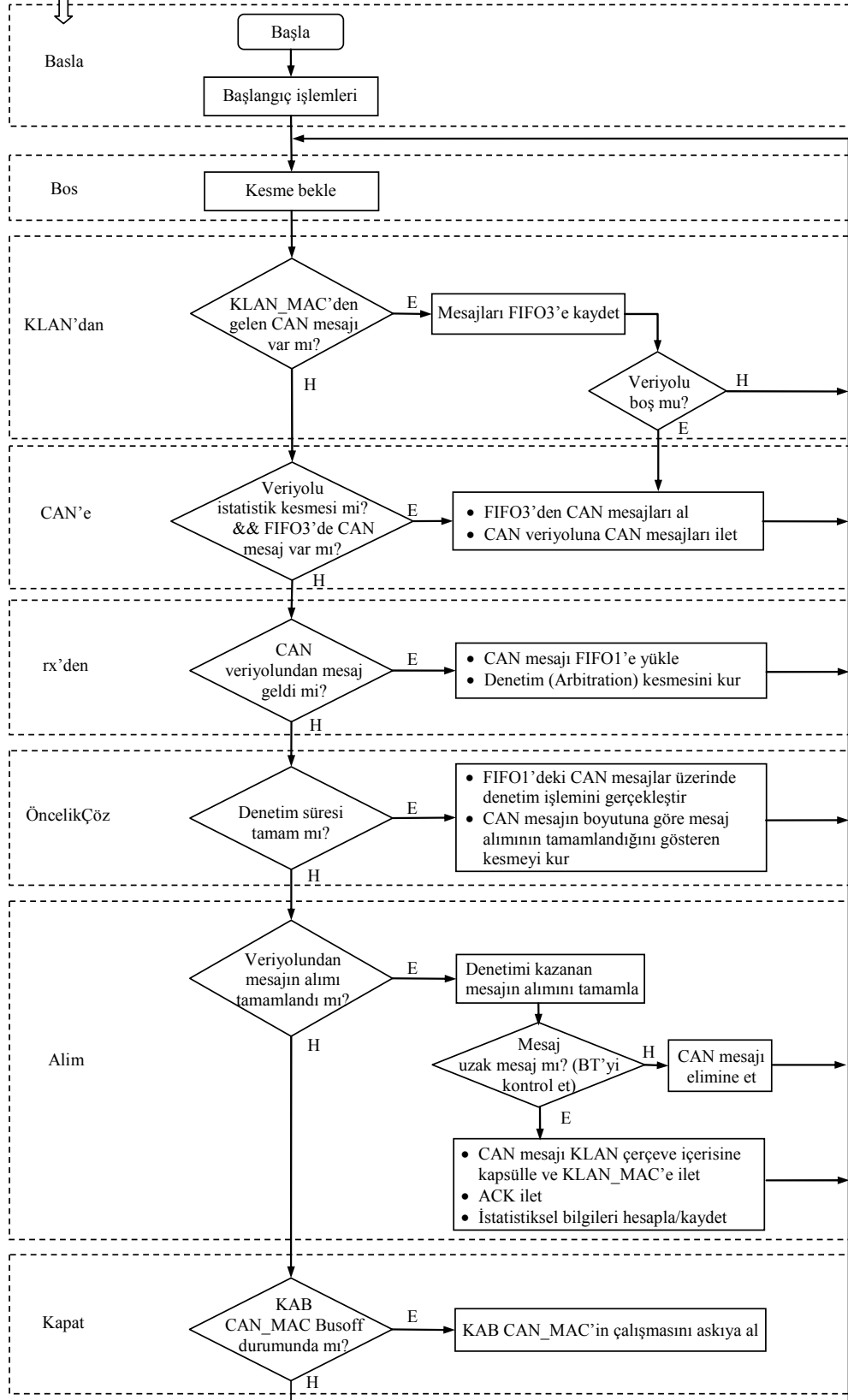
- “Bos” durum makinesi ise model içerisindeki tüm kesme isteklerinin değerlendirilmesini sağlar.
- “rx’den” durum makinesi, veriyolu üzerinden mesaj alımı, CAN mesajın tanıtıcı alan bilgisi ile öncelik (priority) kontrolünü gerçekleştirir. Tanıtıcı bitleri süresince (arbitration) veriyolu üzerindeki tüm mesajlar alınır.
- “ÖncelikÇöz” durum makinesi, rx’den durum makinesi ile alımı başlatılan mesajın başlık alanındaki yük büyüklüğü bilgisine göre mesajın alımının tamamlanması için gerekli gecikmeyi belirten kesmeyi kurar.
- “Alım” durum makinesi ise KAB’ın CAN kısmında en yüksek öncelikli mesajın alımını gerçekleştirir. Veriyolu erişimini kazanan en yüksek öncelikli mesaj, KAB Bakış Tablosu içerisindeki uzak mesajlar arasında ise CAN mesaj KLAN çerçeve içerisine kapsülendir ve kablosuz ortama iletmek üzere KAB düğüm modelindeki KLAN_MAC modülüne iletilir, aksi takdirde ise mesaj elimine edilir.
- “KLAN’dan” durum makinesi, KAB düğüm modeli içerisindeki KLAN_MAC modülünden gelen uzak CAN mesajları alır ve FIFO3 tamponuna kaydeder.
- “CAN’e” durum makinesi, KLAN’dan durum makinesi vasıtasıyla alınarak FIFO3’e kaydedilmiş uzak CAN mesajları tampondan alır ve veriyolu boş ise iletimini gerçekleştirir.
- “Kapat” durum makinesi ise CAN hata sayıcılarının durumuna göre KAB’ın CAN veriyolu ile olan bağlantısını askıya alır.

Şekil 5.10’da KAB CAN_MAC proses modelinin çalışma mekanizmasının özetlendiği algoritması görülmektedir. Bu proses modele ait C program kodları Ek-B’de verilmektedir.



Şekil 5.9: KAB CAN_MAC modülü proses modeli

İşlemin Gerçekleştiği
Durum Makinesi
(Şekil 5.9)



Şekil 5.10: KAB CAN_MAC modülü proses modeli algoritması

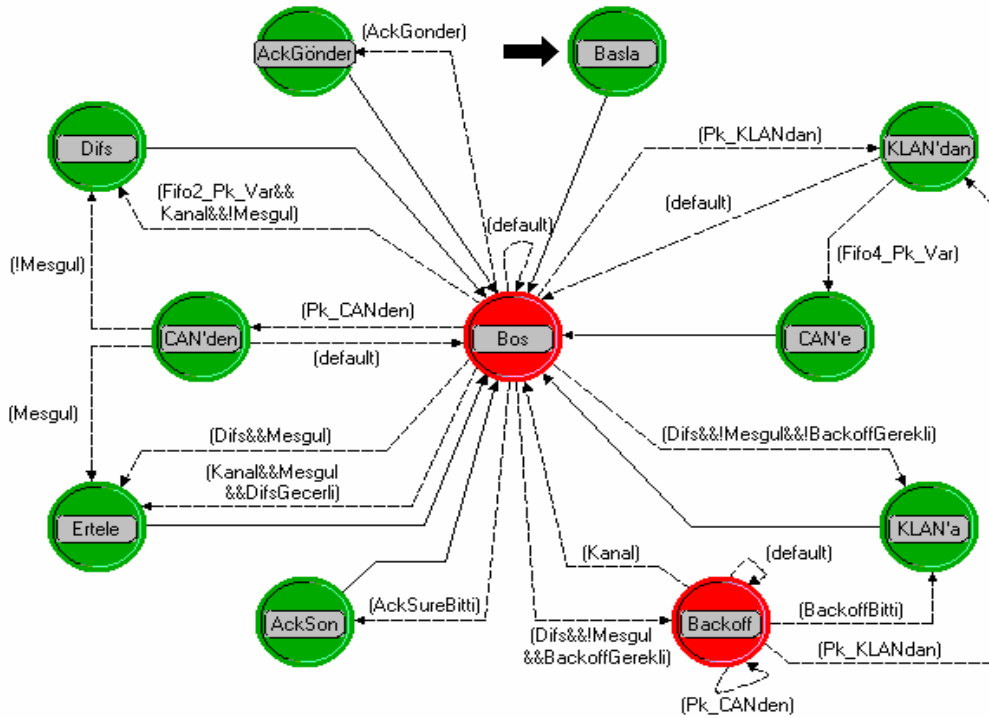
5.4.2.2. KAB KLAN proses modeli ve algoritması

KAB düğüm modeli içerisindeki KLAN_MAC modülünün proses modeli Şekil 5.11’de görülmektedir. Şekilden görüldüğü gibi proses model aşağıda kısaca açıklanan 11 durum makinesinden oluşmaktadır.

- “Basla” durum makinesi, KAB’ın KLAN adresleri, seçilen fiziksel katmana bağlı olarak çerçeveler arası boşluk süresi, uzak CAN mesajlar ile eşleştirilen KAB’ların KLAN adresleri gibi KAB’ın KLAN tarafına ait başlangıç bilgilerinin elde edilmesinden sorumludur. Modeldeki istatistikler burada tanımlanmaktadır.
- “Bos” durum makinesi ise model içerisindeki tüm kesme isteklerinin değerlendirilmesini sağlar.
- “CAN’den” durum makinesi, KAB CAN_MAC modülü tarafından iletilen içerisine uzak CAN mesajların kapsüllendiği KLAN çerçeveleri olarak FIFO2’ye kaydeder. Kablosuz ortamın durumunu (boş ya da meşgul) kontrol eder. Boş ise “Difs” değilse “Ertele” durumuna geçer.
- “Difs” durum makinesi, kurulu ‘Difs’ kesmesi yoksa CSMA/CA ortam erişim mekanizmasının çalışması için gerekli ‘Difs’ kesmesini kurar.
- “Ertele” durum makinesi ise, Backoff algoritmasının çalışması için “Backoff gerekli” bayrağını kurar. Ayrıca, geçerli ‘Difs’ kesmesini iptal eder.
- “Backoff” durum makinesinde, Backoff algoritmasının temelini oluşturan Backoff slotlarının sayısı belirlenir. Backoff slotlarının değeri, Backoff slotlarının sayısı ‘0’ ve çarpışma meydana gelmemişse $[0-CW_{min}]$ veya çarpışma meydana gelmiş ise CW değeri üssel olarak artırılarak $[0-CW_{yeni}]$ aralığında rasgele üretilir. CWyeni değeri CWmax değerinden büyük olduğunda CWmax değerine eşitlenir. Backoff slotlarının sayısı belirlendikten sonra Backoff algoritması çalıştırılır ve Backoff süresinin sonuna ‘BackoffBitti’ kesmesi kurulur. Backoff durumu içerisinde Kanal istatistik kesmesi veya kablosuz ortamdan KLAN çerçeve alım kesmesi gelirse Backoff slotlarının sayısı kaydedilir ve ‘BackoffBitti’ kesmesi iptal edilerek ilgili duruma gidilir. Ayrıca CAN_MAC’den paket kesmesi gelirse, paket FIFO2’ye iletilir ve Backoff algoritması çalışmaya devam eder. ‘BackoffBitti’ kesmesi oluşursa KLAN çerçeveyi iletmek üzere “KLAN’a” durum makinesine geçilir.

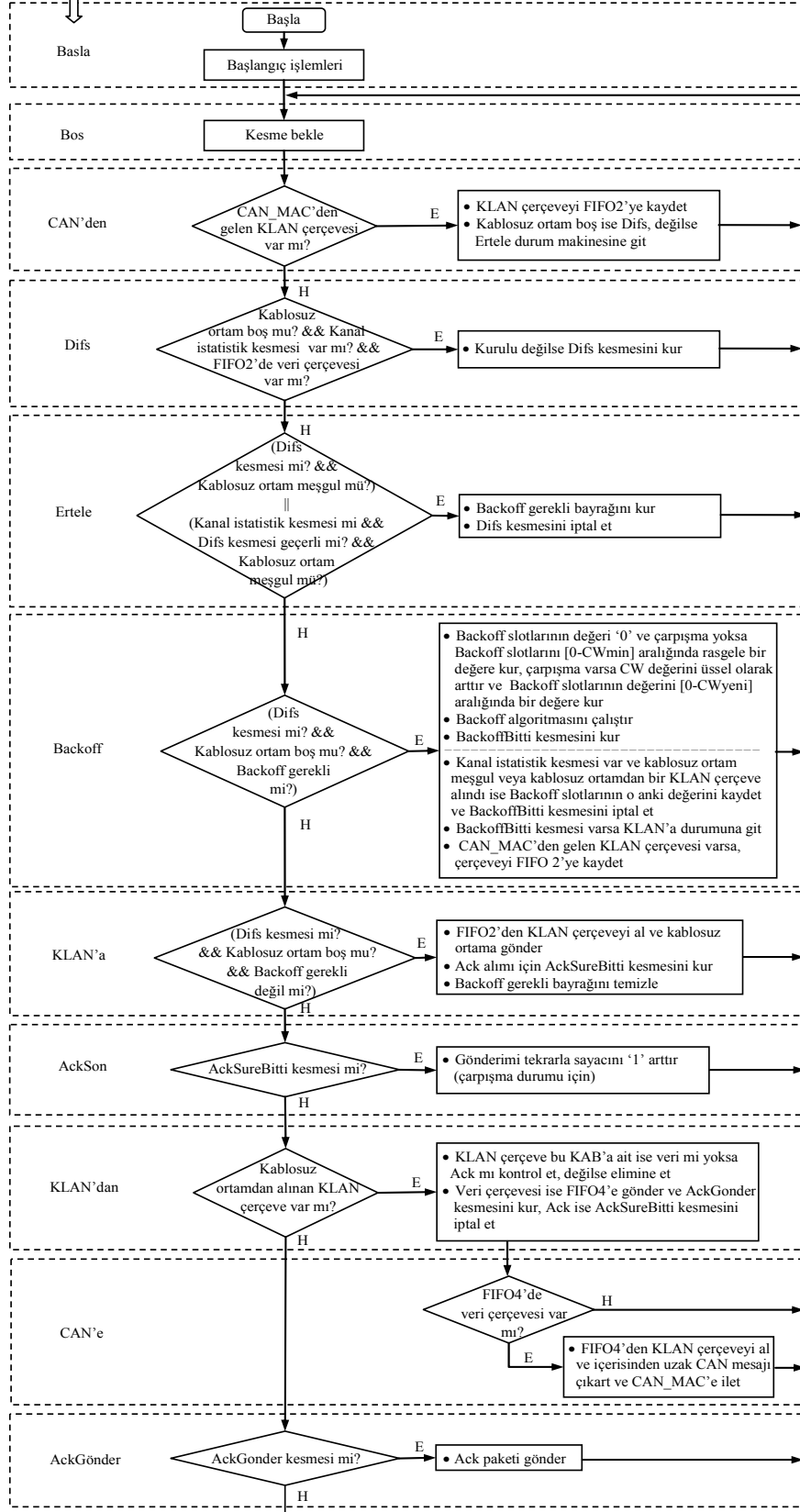
- “KLAN’a” durum makinesi ise kablosuz ortama erişim hakkı kazanıldığında FIFO2’den içerisine uzak CAN mesajların kapsüllendiği KLAN çerçeveleri olarak kablosuz ortama yayın modunda iletir. Backoff gerekli bayrağını temizler.
- “KLAN’dan” durum makinesi, kablosuz ortamdan gelen KLAN çerçeveleri bu KAB’a ait ise alır ve veri mi yoksa Ack mı kontrol eder, Veri ise FIFO4’e kaydeder ve ‘AckGonder’ kesmesini kurar. Ack ise ‘AckSureBitti’ kesmesini iptal eder. Aksi takdirde KLAN çerçeveleri elimine eder.
- “CAN’e” durum makinesi ise FIFO4’den KLAN çerçeveyi alır ve içerisinden uzak CAN mesajları çıkartır (decapsulation). CAN veriyoluna iletilmek üzere KAB’ın CAN_MAC modülüne iletir.
- “AckGönder” durum makinesi, Ack paketi gönderir.
- “AckSon” durum makinesi ise Ack süresi içerisinde bir Ack paketi alınmazsa çarpışma durumunu kontrol eden yeniden gönder sayacını ‘1’ arttırır.

KAB KLAN_MAC proses modelinde gerçekleştirilen tüm işlemlerin algoritması genel hatları ile Şekil 5.12’de görülmektedir. Ayrıca bu proses modele ait C program kodları Ek-B’de sunulmaktadır.



Şekil 5.11: KAB KLAN_MAC modülü proses modeli

İşlemin Gerçekleştiği
Durum Makinesi
(Şekil 5.11)



Şekil 5.12: KAB KLAN_MAC modülü proses modeli algoritması

5.5. CAN/IEEE 802.11b KLAN KAB'ın Kullanıldığı Endüstriyel Ağ Uygulaması ve Başarım Analizi

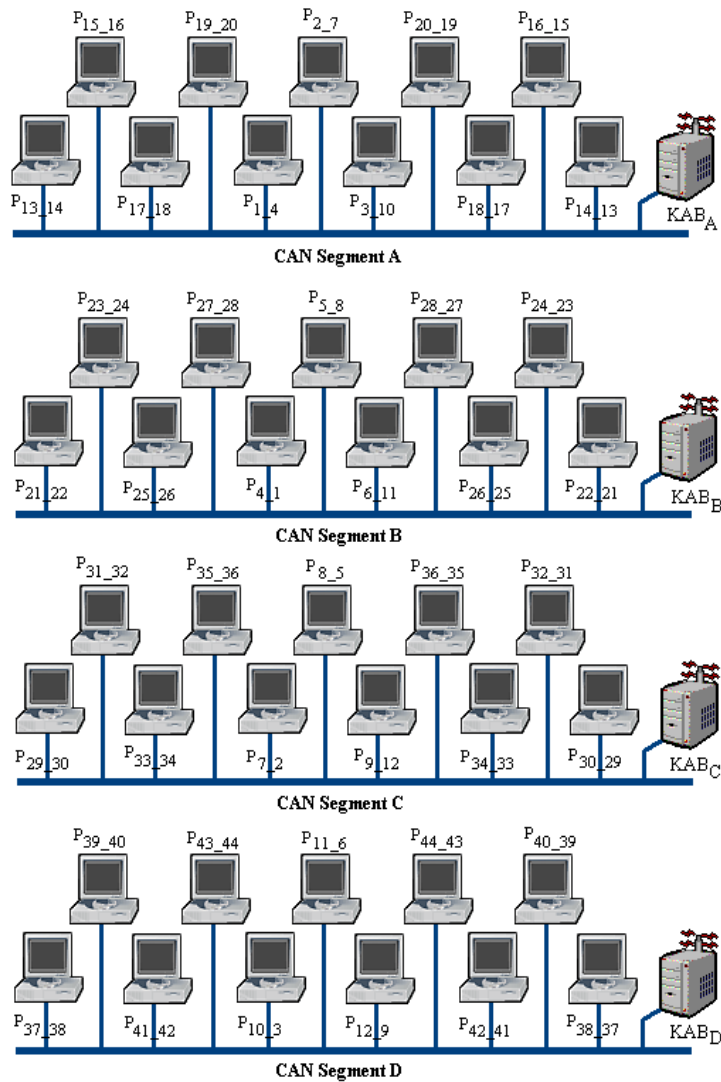
Bu alt bölümde yukarıda ayrıntılı tasarım modelleri sunulan KAB ve CAN düğümleri kullanılarak gerçekleştirilen bir endüstriyel ağ modeli ve başarım analizi sunulmaktadır.

CAN gerçeklemeleri o kadar çok yönlüdür ki, bir CAN uygulamasının başarım analizi için genel bir model yoktur. Buna rağmen SAE Benchmark CAN gerçeklemelerin değerlendirilmesinde bir ölçüt sistemi olarak kullanılmaktadır. SAE Benchmark bir araba prototipinde birkaç farklı altsistem arasında gönderilen sinyallerden oluşmakta ve bu sinyallerin CAN veriyolu üzerindeki zamanlamasını incelemektedir (Lawrenz, 1997). Gerçek zamanlı sistemlerin tasarımı ve başarım değerlendirmesi için 53 adet veri tipini 17 adet mesaj tipinde toplayan ve bunları 6 öncelik sınıfında tanımlayan bir ölçüt sistemidir.

Şekil 5.13'de KAB'ın kullanıldığı SAE Benchmark'a göre gerçekleştirilen endüstriyel ağ uygulaması görülmektedir. Şekilden anlaşılacağı üzere endüstriyel ağ her biri 11 CAN düğümü içeren 4 ayrı segmentten oluşmaktadır. Endüstriyel ağ benzetim modeli oluşturulurken göz önüne alınan ayrıntılar aşağıda maddeler halinde özetlenmektedir.

- CAN sistemi, atanmış (dedicated) bir sistem olduğu için ağ içerisindeki her bir CAN düğümün hangi düğümlerle hangi şartlar altında haberleşeceği başlangıçta belirlenmiştir. SAE Benchmark'a göre gerçekleştirilen CAN mesaj özellikleri Tablo 5.3'de verilmektedir. Tablodan görüldüğü gibi genel olarak üretilen her bir mesajın sabit bir üretim aralığı ve hedefe varması için tanınan bir üst gecikme değeri vardır.
- Tablo 5.3'den de görüldüğü gibi ağ içerisindeki her bir CAN düğümü tüm ağ içerisinde tek olan bir öncelik bilgisine sahip tek bir CAN mesajı üretmektedir.
- CAN mesajlar üssel dağılım fonksiyonu kullanılarak üretilmektedir.
- CAN veriyolu iletim hızı, endüstriyel uygulamalarda sıklıkla karşılaşılan değer olan 100 Kbit/s seçilmiştir.

- Kablosuz yerel alan ağı serbest uzay yayılım kanal modeli ile 11 Mbit/s veri iletim hızında çalıştırıldı.
- CAN veriyolu ve kablosuz ortamda oluşabilecek iletim hataları ihmal edilmiştir.
- Köprü tamponlarının boyutu, benzetimde sonsuz olarak kabul edilmiştir. Pratikte ise olması gereken minimum değer tespit edilerek uygun kapasiteli bellek kullanılabilir.
- KAB'ın çerçeveyi alması, tampona yazması, işleme, iletme ya da elimine etmesi gibi işlemlerin proses zamanları gecikme olarak tanımlanmıştır.



KAB : Kablosuz Arabağlaşım Birimi

P_{s-r} : P CAN Düğümü

s: CAN düğüm tarafından üretilen mesajın öncelik bilgisi

(Aynı zamanda CAN düğümün numarasını gösterir)

r: CAN düğümüne hedeflenen mesajın önceliği

Şekil 5.13: KAB'ın kullanıldığı endüstriyel ağ uygulaması

Tablo 5.3: CAN segmentlerin Yerel (Y) ve Uzak (U) mesaj bilgileri

| | KD | MT | Yük | MB | SAE | | HD | WÇA | |
|-----------------|-----------------|----|-----|----|--------|-----------------|-----------------|-----|----|
| | | | | | T (ms) | D (ms) | | KS | HS |
| Segment A | P ₁ | U | 1 | 7 | 5 | 5 | P ₄ | A | B |
| | P ₂ | U | 1 | 7 | 10 | 10 | P ₇ | A | C |
| | P ₃ | U | 2 | 8 | 5 | 5 | P ₁₀ | A | D |
| | P ₁₃ | Y | 1 | 7 | 5 | 5 | P ₁₄ | - | - |
| | P ₁₄ | Y | 1 | 7 | 50 | 5 | P ₁₃ | - | - |
| | P ₁₅ | Y | 2 | 8 | 5 | 5 | P ₁₆ | - | - |
| | P ₁₆ | Y | 3 | 9 | 10 | 10 | P ₁₅ | - | - |
| | P ₁₇ | Y | 1 | 7 | 5 | 5 | P ₁₈ | - | - |
| | P ₁₈ | Y | 6 | 12 | 10 | 10 | P ₁₇ | - | - |
| | P ₁₉ | Y | 1 | 7 | 5 | 5 | P ₂₀ | - | - |
| P ₂₀ | Y | 2 | 7 | 5 | 5 | P ₁₉ | - | - | |
| Segment B | P ₄ | U | 1 | 7 | 5 | 5 | P ₁ | B | A |
| | P ₅ | U | 1 | 7 | 10 | 10 | P ₈ | B | C |
| | P ₆ | U | 2 | 8 | 5 | 5 | P ₁₁ | B | D |
| | P ₂₁ | Y | 1 | 7 | 5 | 5 | P ₂₂ | - | - |
| | P ₂₂ | Y | 1 | 7 | 50 | 5 | P ₂₁ | - | - |
| | P ₂₃ | Y | 2 | 8 | 5 | 5 | P ₂₄ | - | - |
| | P ₂₄ | Y | 3 | 9 | 10 | 10 | P ₂₃ | - | - |
| | P ₂₅ | Y | 1 | 7 | 5 | 5 | P ₂₆ | - | - |
| | P ₂₆ | Y | 6 | 12 | 10 | 10 | P ₂₅ | - | - |
| | P ₂₇ | Y | 1 | 7 | 5 | 5 | P ₂₈ | - | - |
| P ₂₈ | Y | 2 | 7 | 5 | 5 | P ₂₇ | - | - | |
| Segment C | P ₇ | U | 1 | 7 | 5 | 5 | P ₂ | C | A |
| | P ₈ | U | 1 | 7 | 10 | 10 | P ₅ | C | B |
| | P ₉ | U | 2 | 8 | 5 | 5 | P ₁₂ | C | D |
| | P ₂₉ | Y | 1 | 7 | 5 | 5 | P ₃₀ | - | - |
| | P ₃₀ | Y | 1 | 7 | 50 | 5 | P ₂₉ | - | - |
| | P ₃₁ | Y | 2 | 8 | 5 | 5 | P ₃₂ | - | - |
| | P ₃₂ | Y | 3 | 9 | 10 | 10 | P ₃₁ | - | - |
| | P ₃₃ | Y | 1 | 7 | 5 | 5 | P ₃₄ | - | - |
| | P ₃₄ | Y | 6 | 12 | 10 | 10 | P ₃₃ | - | - |
| | P ₃₅ | Y | 1 | 7 | 5 | 5 | P ₃₆ | - | - |
| P ₃₆ | Y | 2 | 7 | 5 | 5 | P ₃₅ | - | - | |
| Segment D | P ₁₀ | U | 1 | 7 | 5 | 5 | P ₃ | D | A |
| | P ₁₁ | U | 1 | 7 | 10 | 10 | P ₆ | D | B |
| | P ₁₂ | U | 2 | 8 | 5 | 5 | P ₉ | D | C |
| | P ₃₇ | Y | 1 | 7 | 5 | 5 | P ₃₈ | - | - |
| | P ₃₈ | Y | 1 | 7 | 50 | 5 | P ₃₇ | - | - |
| | P ₃₉ | Y | 2 | 8 | 5 | 5 | P ₄₀ | - | - |
| | P ₄₀ | Y | 3 | 9 | 10 | 10 | P ₃₉ | - | - |
| | P ₄₁ | Y | 1 | 7 | 5 | 5 | P ₄₂ | - | - |
| | P ₄₂ | Y | 6 | 12 | 10 | 10 | P ₄₁ | - | - |
| | P ₄₃ | Y | 1 | 7 | 5 | 5 | P ₄₄ | - | - |
| P ₄₄ | Y | 2 | 7 | 5 | 5 | P ₄₃ | - | - | |

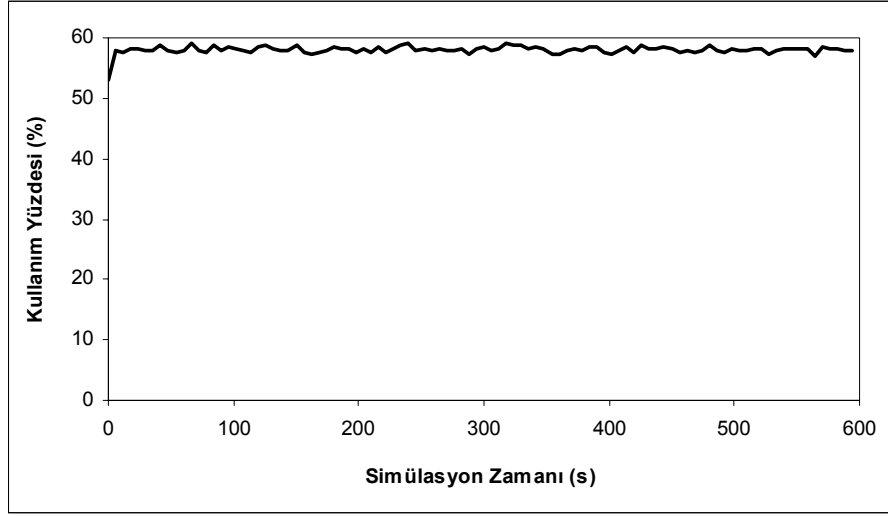
KD: Kaynak Düğüm
MT: Mesaj Tipi
Y: Yerel
U: Uzak
Yük*: Veri Boyutu (bayt)
MB*: Mesaj Boyutu (bayt)
SAE: Society of Automotive Engineers Benchmark
T: Üretim aralığı (ms)
D: Varma sınır değeri (ms)
HD: Hedef Düğüm
WÇA: WLAN Çerçeve Adresi
KS: Kaynak Segment
HS: Hedef Segment

(*Tüm CAN mesajlar Üssel Dağılım Fonksiyonu kullanılarak üretilmiştir.)

5.5.1. Benzetim sonuçları ve başarımların analizi

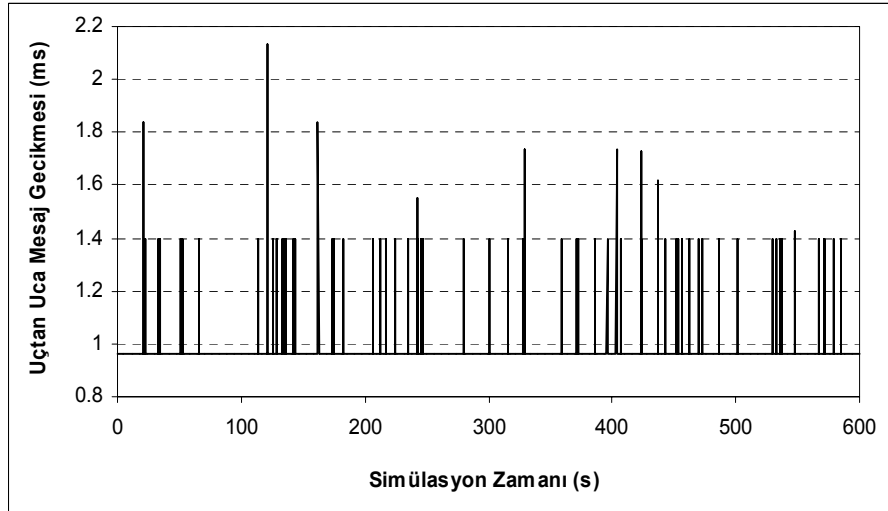
Endüstriyel ağın başarımların analizi, farklı yükler altında simülasyon süresine göre CAN veriyolunun kullanım yüzdesi (utilization), Yerel/Uzak CAN mesajların uçtan uca gecikmesi, Kablosuz Arabağlaşım Birimi'nin CAN-KLAN ve KLAN-CAN işlem zamanı, KAB'ın toplam işlem süresinin Uzak CAN mesaj gecikmesine etkisi ve değişik veri iletim hızlarındaki paket kaybı parametreleri değerlendirilerek gerçekleştirilmiştir.

CAN/IEEE 802.11b KLAN KAB'ın sistem karakteristiğini belirleyen en önemli etken KAB'ın bağılı olduğu veriyolunun kullanım yüzdesidir. CAN veriyolu kullanım yüzdesi, o veriyolu üzerinde saniyede transfer edilen mesaj yoğunluğunu gösterir. Şekil 5.14'de tasarlanan modeldeki CAN veriyolu A'nın simülasyon süresince elde edilen kullanım yüzdesi görülmektedir. Benzetim modelinde tüm segmentlerdeki trafik yoğunluğu eşdeğer olarak belirlenmiştir. Bu nedenle modeldeki tüm veriyollarının kullanım yüzdesi yaklaşık olarak aynıdır. Şekilden görüldüğü gibi benzetim süresince CAN veriyolunun kullanım yüzdesi yaklaşık olarak %58 seviyelerindedir. Bu oran, Tablo 5.3'de CAN düğümlerin mesaj üretim aralıkları dikkate alındığında veriyolu üzerinde 1720 mesaj/s değerine karşılık gelmektedir. Bu bilgi, KAB'ın kullanıldığı endüstriyel ağın istenen başarımlarını gösterebilmesi için CAN segmentte üretilmesi gereken mesaj yoğunluğunu göstermektedir. Sonuç olarak CAN veriyolunun kullanım yüzdesi veriyolu üzerinde transfer edilen mesaj sayısı ile orantılıdır.



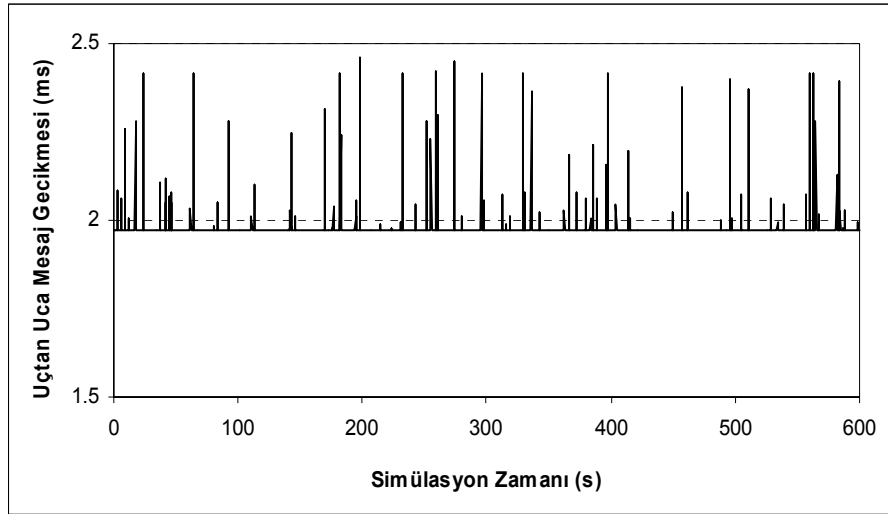
Şekil 5.14: CAN veriyolu A'nın kullanım yüzdesi

Şekil 5.15'de simülasyon süresi boyunca en yüksek öncelikli yerel P₁₃₋₁₄ CAN mesajının uçtan uca gecikmesi görülmektedir. Tablo 5.3'de verildiği gibi P₁₃ CAN mesajı aynı segmentteki P₁₄ CAN düğümünü hedeflemektedir ve üretilme aralığı ile varma sınır değeri 5 ms'dir. Şekilden görüldüğü gibi yerel P₁₃₋₁₄ CAN mesajı simülasyon süresi boyunca SAE Benchmark tarafından belirlenen varma sınır değerini aşmamaktadır. Maksimum gecikme değeri 2.1 ms ve ortalama gecikme değeri de 1.1 ms'dir.

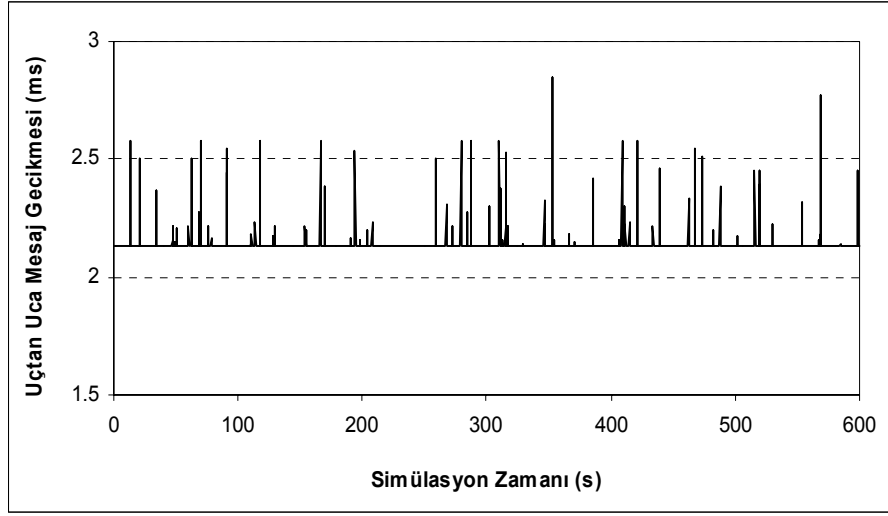


Şekil 5.15: P₁₃₋₁₄ yerel CAN mesajının uçtan uca gecikmesi

Şekil 5.16 ve Şekil 5.17 sırasıyla benzetim süresi boyunca en yüksek öncelikli uzak (P_1) CAN mesajın ve en düşük öncelikli uzak (P_{12}) CAN mesajın uçtan uca gecikmesini göstermektedir. Uzak mesajların uçtan uca gecikmesi KAB'ın işlem zamanını da içermektedir. Bu yüzden uzak mesajlar yalnızca yerel mesajlar ile değil farklı öncelik bilgisine sahip uzak mesajlar ile de karşılaştırılabilir. Tablo 5.3'de görüldüğü gibi her iki uzak mesajın SAE Benchmark tarafından belirlenen varma sınır değeri 5 ms'dir ve simülasyon süresi boyunca asla bu değer aşılmamaktadır. Bu, gerçekleştirilen KAB'ın uzak mesaj gecikmesine SAE Benchmark tarafından belirlenen varma sınır değerini geçmeyecek kadar bir etkisi olduğunu göstermektedir. Diğer taraftan en yüksek öncelikli uzak (P_1) CAN mesajın ortalama mesaj gecikmesi 1.98 ms iken en düşük öncelikli uzak (P_{12}) CAN mesajın ortalama mesaj gecikmesi 2.14 ms'dir. Aynı zamanda en düşük öncelikli uzak (P_{12}) CAN mesajın ortalama gecikmesi en yüksek öncelikli yerel (P_{13}) CAN mesajının uçtan uca ortalama gecikmesinin yaklaşık iki katıdır.



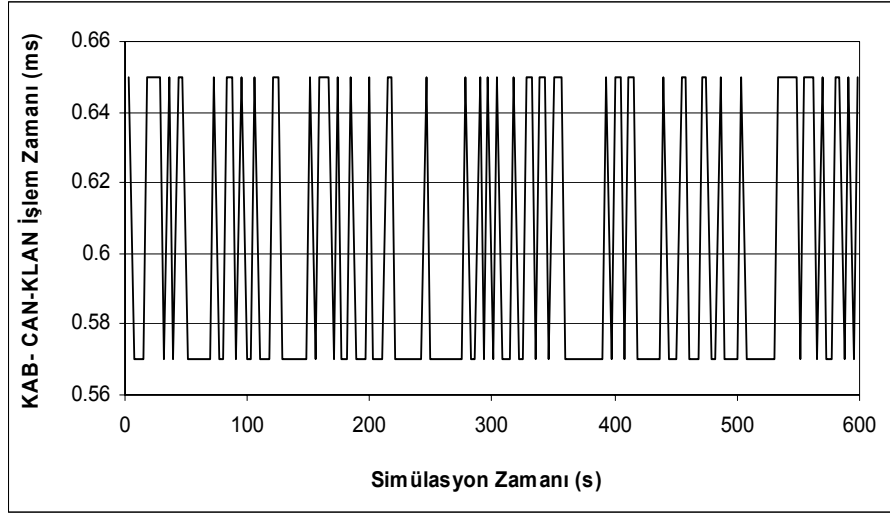
Şekil 5.16: En yüksek öncelikli P_1 uzak CAN mesajının uçtan uca gecikmesi



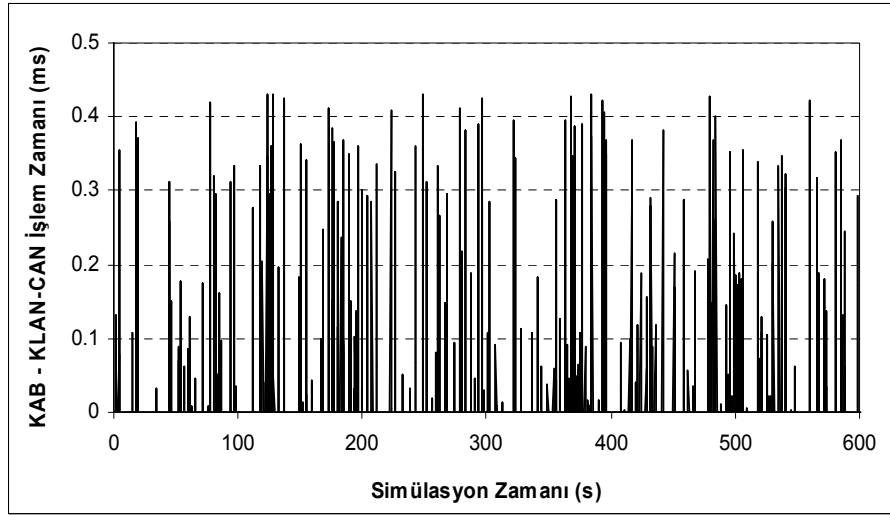
Şekil 5.17: En düşük öncelikli P₁₂ uzak CAN mesajının uçtan uca gecikmesi

Genellikle arabağlaşım elemanlarının performans değerlendirmesi filtreleme hızı ve gönderim hızı parametrelerine göre yapılmaktadır. Bununla birlikte KAB'ın bir portundan diğer portuna veri iletilinceye kadar geçen zaman da “toplam işlem zamanı” KAB içerisinde akan yüksek mesaj oranı ile ilgilidir. KAB işlem zamanı, hem arabağlaşım biriminin ağır yüklenmesinden hem de veriyolu üzerinde transfer edilen yerel/uzak mesaj oranlarından etkilenmektedir.

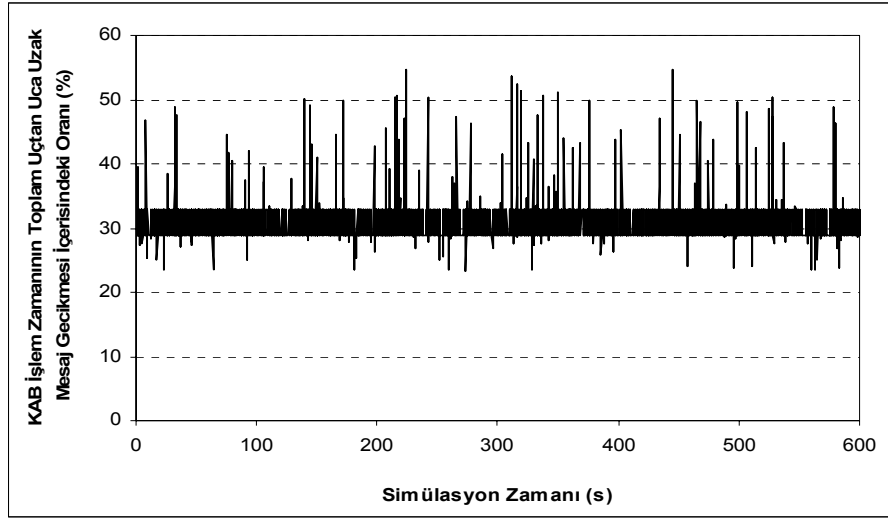
Şekil 5.18 ve Şekil 5.19 sırasıyla, benzetim süresi boyunca Kablosuz Arabağlaşım Birimi'nin CAN-KLAN ve KLAN-CAN işlem zamanını göstermektedir. CAN-KLAN işlem zamanı 0.57 ms ile 0.65 ms arasında iken KLAN-CAN işlem zamanı 0 ile 0.43 ms arasında değişmektedir. Sonuç olarak KAB içerisindeki toplam işlem zamanı en kötü durumda dahi 1.08 ms (0.65 ms + 0.43ms)'dir. Buradan görüldüğü üzere bir uzak mesajın transferinde KAB en az 0.57 ms'lik bir gecikme sağlamaktadır. KAB'ın toplam işlem zamanı uzak CAN mesajının uçtan uca gecikmesi içerisinde kısmen etkilidir. Şekil 5.20'de bu etki görülmektedir ve 100 Kbit/s CAN veri iletim hızında yaklaşık % 31'dir. Bu oranın daha yüksek CAN veri iletim hızlarında azalması beklenir.



Şekil 5.18: KABA CAN-KLAN işlem zamanı

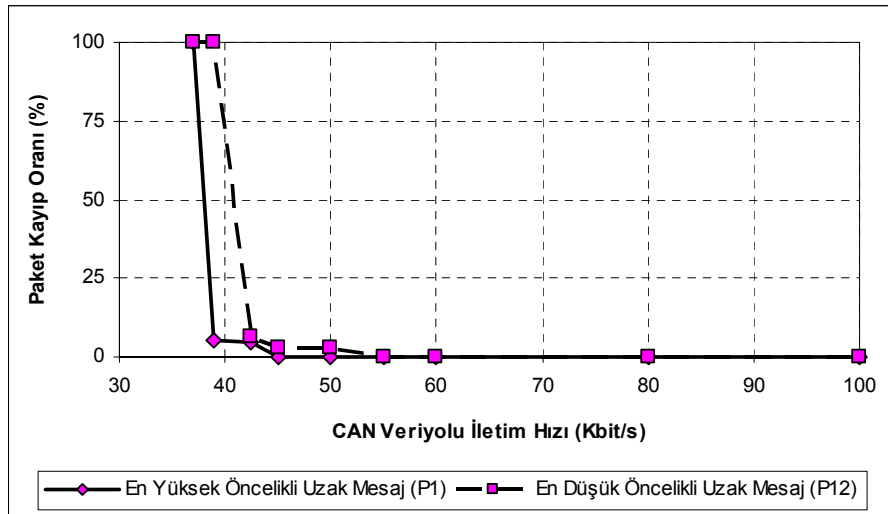


Şekil 5.19: KABA KLAN-CAN işlem zamanı



Şekil 5.20: KAB'ın işlem zamanının P_1 uzak CAN mesajın uçtan uca toplam gecikmesi içerisindeki oranı

Şekil 5.21'de farklı CAN veriyolu veri iletim hızlarında KAB üzerinden aktarılan en yüksek (P_{1-4}) ve en düşük (P_{12-9}) öncelikli uzak CAN mesajlarında meydana gelen paket kayıp oranı verilmektedir. Tablo 5.3'de verilen SAE Benchmark mesaj varma sınırı değeri içerisinde hedefine ulaşamayan mesajlar kayıp olarak kabul edilmektedir ve bu oran çoğunlukla benzetim modelindeki CAN veriyolu iletim hızı ile ilgilidir. CAN veriyolu veri iletim hızı 100 Kbit/s'den 37 Kbit/s'ye azaltıldığında, 50 Kbit/s'e kadar önerilen modelde paket kaybı görülmemektedir. Ancak 50–37 Kbit/s veri iletim hızı aralığında her iki uzak mesaj transferinde de artan bir paket kaybı oranı görülmektedir.



Şekil 5.21: Farklı CAN veriyolu iletim hızlarında paket kayıp oranı

Benzetim sonuçları CAN mesajların varma gereksinimleri göz önüne alındığında, gerçekleştirilen Kablosuz Arabağlaşım Birimi'nin performansı kabul edilebilirdir. Benzetim modelinde kullanılan SAE Benchmark'a göre 100 Kbit/s ve daha üzerinde veri CAN mesajları varma sınır değerini aşmadığından KAB uzak CAN segmentlerin haberleşmesi için gerekli servisleri sağlamaktadır.

5.6. Sonuç

İletişim sistemleri ve haberleşme ağlarını modellemek ve başarımlarını analizlerini elde etmek için günümüzde birçok yöntem kullanılmaktadır. Maliyet, kolaylık, tasarım ve değerlendirme süreci gibi avantajları dikkate alındığında sistem analizi için ayrık olaya dayalı bilgisayar benzetim yöntemi en çok tercih edilen yöntemdir. Birçok bilgisayar benzetim yazılımı vardır. Kablosuz iletişim desteği, nesneye yönelik ve görsel programlama imkanı sağlaması ve çok geniş bir kullanıcı profili olması gibi sebeplerden dolayı OPNET yazılımı haberleşme ağlarının modellenmesinde tercih edilen bir yazılımdır.

Bir sistemin benzetimini yapabilmek için öncelikle sistemi modellemek gerekir. Bu bölümde tez çalışmasının temelini oluşturan CAN/IEEE 802.11b KLAN KAB'ın OPNET Modeller yazılımında modellenmesi ve tasarım aşamaları ayrıntılı olarak sunulmuş ve tasarlanan KAB'ın kullanıldığı bir endüstriyel ağ uygulamasının değişik şartlar altında benzetimi gerçekleştirilmiştir.

Endüstriyel ağ uygulamasının başarımlarını analiz CAN sistemlerinin değerlendirilmesinde bir ölçüt olarak kullanılan SAE Benchmark'a göre yapılmıştır. Elde edilen sonuçlar tasarlanan KAB'ın CAN tabanlı endüstriyel sistemlerin gerçek zamanlı iletişim gereksinimini karşıladığını göstermektedir.

6. CAN/IEEE 802.11B KLAN KABLOSUZ ARABAĞLAŞIM BİRİMİ PROTOTİPİNİN GERÇEKLEŞTİRİLMESİ VE UYGULAMALARI

6.1. Giriş

Sayısal veri haberleşme mühendisliğinde sistem başarımlarını değerlendirmesi için fiziksel model (prototip) kullanımı en güvenilir ve en doğru yöntemdir. Ancak çok geniş ve kompleks uygulama alanına sahip sayısal veri haberleşmesinde, bu yöntemin uygulanması oldukça zordur. Fiziksel model yönteminde başarımlar, mevcut sistem ya da sistemin prototipi değişik koşullar altında incelenerek elde edilebilir (Ertürk 2000, Özçelik 2002).

Bir sistemin prototipini gerçekleştirme, çoğunlukla yüksek maliyet ve zaman gerektiren bir yöntemdir. Bundan dolayı diğer ağ başarımlar yöntemlerine göre pratik değildir. Ancak sistemin çalışırılığını hakkında kesin bir bilgi verir.

CAN gerçekleştirmeleri o kadar çok yönlüdür ki, bir CAN uygulamasının başarımlar analizi için genel bir model yoktur. Bilgisayarlı benzetim yönteminde olduğu gibi fiziksel model yönteminde de herhangi bir CAN ağ uygulaması, bize sistem hakkında bilgi vermektedir.

Bu bölümde CAN/IEEE 802.11b KLAN Kablosuz Arabağlaşım Birimi'nin prototipi ve bu prototipin kullanıldığı iki uygulama sunulmaktadır. Birinci uygulamada, KAB üzerinden iki bağımsız CAN segmentinin kablosuz haberleştiği bir endüstriyel ağ gerçekleştirilmiştir. Burada amaç, KAB'ın genel olarak çalışırılığını göstermektir. İkinci uygulamada ise CAN kullanılarak kontrol edilen bir arabanın KAB üzerinden kablosuz ve ses yönlendirilmesi gerçekleştirilmiştir. Böylece KAB'ın değişik kontrol uygulamalarında kullanabileceğine bir örnek verilmektedir.

6.2. CAN/IEEE 802.11b KLAN KAB'ın Prototipi

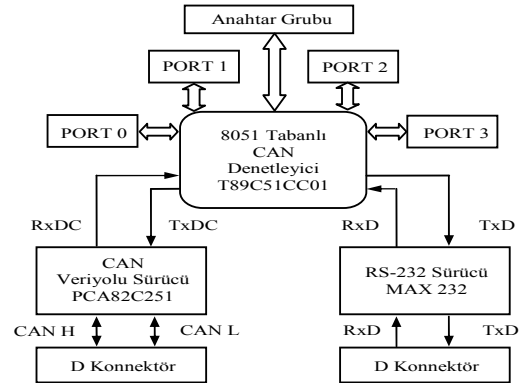
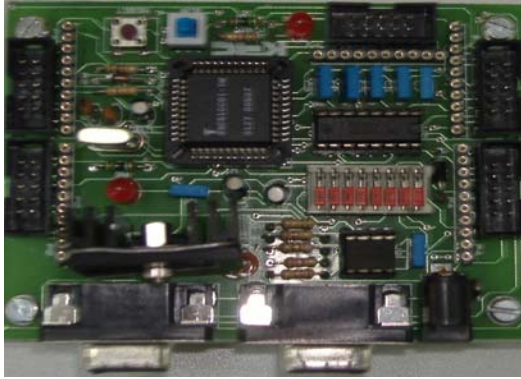
CAN/IEEE 802.11b KLAN Kablosuz Arabağlaşım Birimi, CAN ve IEEE 802.11b KLAN ağların arabağlaşımını sağladığından her iki ağ ile iletişim kurabilecek iki farklı yapıya ve iki farklı porta sahiptir. Her bir portta gerçekleştirilen süreçler farklıdır. KAB, CAN ağ ve KLAN ağ ile iletişimi sağlayacak iki ayrı birimden oluşmaktadır. KAB'ın CAN ağ ile iletişimi Karaca (2003)'de sunulan CAN geliştirme kartı, KLAN ağ ile iletişimi ise IEEE 802.11b standardını destekleyen kablosuz birim ile sağlanmaktadır.

6.2.1. 8051 tabanlı CAN denetleyici geliştirme kartı

Şekil 6.1'de CAN geliştirme kartı ve blok diyagramı görülmektedir. Bu kart, CAN standardının tüm özelliklerini sağlamaktadır. CAN geliştirme kartı, KAB'ın prototipinin CAN kısmı olarak kullanılmasının yanı sıra bağımsız bir CAN düğüm olarak da kullanılabilir. Geliştirme kartı genel olarak şu elemanlardan oluşmaktadır:

- T89C51CC01 mikrodenetleyicisi: CAN ağ uygulamalarında kullanılmak amacıyla geliştirilmiş, Bosch GmbH tarafından tanımlanan CAN seri haberleşme protokolünü gerçekleştirmek için istenen tüm özellikleri sağlayan 8 bit'lik bir mikrodenetleyicidir. İçerisindeki CAN denetleyici, CAN 2.0A standart ve CAN 2.0B genişletilmiş biçimlerini desteklemektedir. CAN denetleyici, CAN veriyolunun maksimum transfer hızı olan 1 Mbit/s veri iletim hızını sağlamaktadır. CAN denetleyiciye erişim özel fonksiyon saklayıcıları (Special Function Registers, SFR) aracılığı ile yapılır. SFR'ler yardımıyla erişilebilen birbirinden bağımsız 15 adet programlanabilir mesaj nesnesine (mailbox) sahiptir. Her bir mesaj nesnesi, mesaj alma, gönderme ve alım tamponu olarak programlanabilir.
- PCA82C251 entegresi (CAN veriyolu sürücüsü) ve D konnektör: Geliştirme kartının CAN veriyoluna bağlantısını sağlamaktadır. CAN veriyoluna fiziksel bağlantı RS485 soketi ile yapılmaktadır.

- MAX 232 entegresi (RS-232 sürücüsü) ve D konnektör: Sayısal elemanlar TTL ya da CMOS lojik seviyesi (0 ve 5V) kullanırlarken, bilgisayarların desteklediği RS-232 seri arabirimi negatif lojik kullanır. MAX 232 entegresi RS-232 seri arabirimi ile sayısal elemanlar arasında lojik seviye dönüşümünü gerçekleştirir. Geliştirme kartında T89C51CC01 mikrodenetleyicisine bilgisayardan program yüklenmesi ve RS-232 seri arabirimi kullanan diğer cihazlar ile haberleşmesi amacıyla MAX 232 entegresi ve RS-232 soketi kullanılmaktadır.
- Portlar: T89C51CC01 mikrodenetleyicisi Port 0, 1, 2 ve 3 olmak üzere 8 bit'lik 4 porta sahiptir. Her bir port uygulamalarda kullanılmak amacıyla ayrı soketlere bağlanmıştır. Aynı zamanda mikrodenetleyicinin CAN uçları (TxDC ve RxDC) da ayrı bir port şeklinde kullanılabilir.
- Anahtar grubu: Geliştirme kartının programlanma, çalışma, CAN veriyolu ile haberleşmenin açılması, sonlandırıcı dirençlerin kullanımı gibi ayarlar kart üzerindeki anahtar grubu yardımıyla yapılmaktadır.

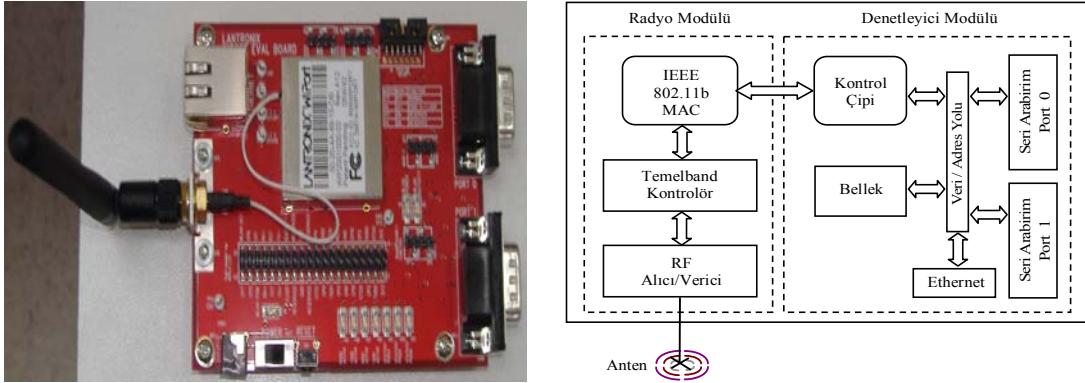


Şekil 6.1: 8051 tabanlı CAN denetleyici geliştirme kartı ve blok diyagramı

6.2.2. IEEE 802.11b KLAN birimi

Şekil 6.2'de IEEE 802.11b KLAN birimi ve blok diyagramı görülmektedir. IEEE 802.11b KLAN birimi, IEEE 802.11b standardının tüm özelliklerini sağlamaktadır. Bu birim IP adresine sahip bir web sunucu içermektedir. Böylelikle hem internet hem de yerel bir ağ üzerinden kontrol edilebilmektedir. IEEE 802.11b KLAN birimi uygulamaya bağlı olarak TCP (Transmission Control Protocol) ve UDP (User Datagram Protocol) protokollerini desteklemektedir (Wiport Guide 2006). Seri

arabirimlerinden ya da Ethernet girişinden gelen bilgiyi IEEE 802.11b KLAN biçimine çevirerek kablosuz ortama iletir. Aynı şekilde kablosuz ortamdan aldığı bilgiyi üzerinde gerekli dönüşüm işlemlerini yaparak seri arabirimlerine ya da Ethernet girişine iletir. IEEE 802.11b KLAN birimindeki programlanabilir entegrenin yazılımı güncellenerek (update) IEEE 802.11g standardını desteklemesi de sağlanabilir.



Şekil 6.2: IEEE 802.11b KLAN birimi ve blok diyagramı

6.2.3. CAN/IEEE 802.11b KLAN KAB

CAN geliştirme kartı ve IEEE 802.11b KLAN biriminden oluşan CAN/IEEE 802.11b KLAN Kablosuz Arabağlaşım Birimi prototipi ve blok diyagramı Şekil 6.3’de görülmektedir. CAN geliştirme kartı ve IEEE 802.11b KLAN birimine ilave olarak KAB’da ne gibi bir işlem yapıldığını (alınan ya da gönderilen mesaj içeriği vb.) göstermek üzere LCD gösterge de kullanılmaktadır. KAB’da meydana gelen işlemler, CAN tarafından IEEE 802.11b KLAN tarafına ve IEEE 802.11b KLAN tarafından CAN tarafına olmak üzere iki ayrı veri transfer sürecinden oluşmaktadır.

CAN tarafından IEEE 802.11b KLAN tarafına KAB’ın çalışması;

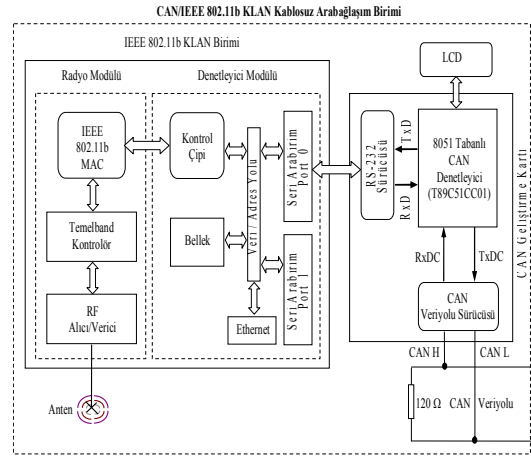
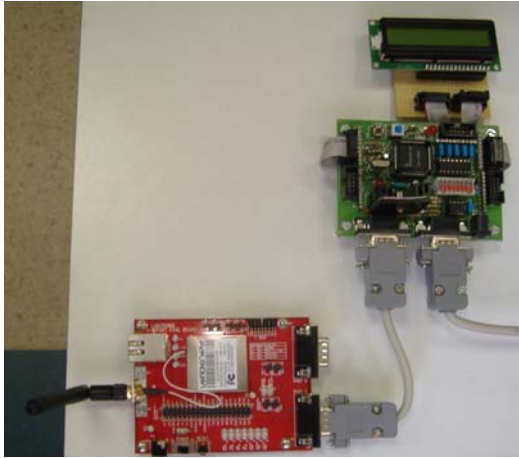
- KAB, CAN geliştirme kartının CAN portu vasıtasıyla CAN veriyolundan mesajları alır.
- Alınan CAN mesajın uzak ya da yerel bir mesaj olduğunu tanıttıcı alanındaki bilgiye bakarak kontrol eder. Eğer uzak bir mesaj ise mesajın tanıttıcı alan bilgisini ve mesaj içerisindeki veriyi CAN geliştirme kartının seri portundan

KAB'ın IEEE 802.11b KLAN biriminin seri portuna iletir. Aynı zamanda IEEE 802.11b KLAN birimine iletilen uzak mesajın tanıtıcı alan bilgisini (priority) ve içeriğini LCD'de gösterir. Aksi takdirde CAN mesajı elimine eder.

- KAB'ın IEEE 802.11b KLAN birimi, seri arabirimi vasıtasıyla aldığı her bilgiyi kablosuz ortama iletir.

IEEE 802.11b KLAN tarafından CAN tarafına KAB'ın çalışması;

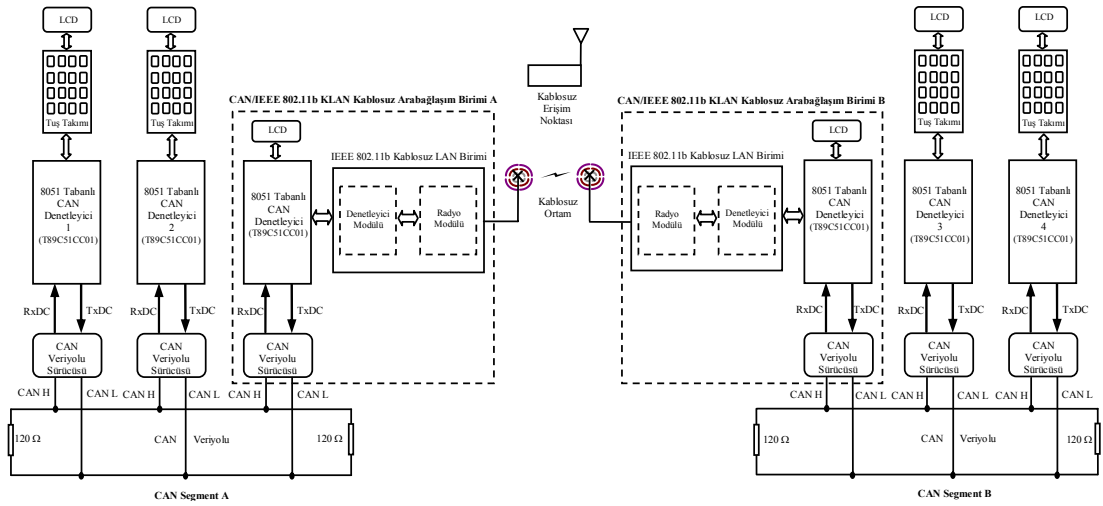
- IEEE 802.11b KLAN birimi, kablosuz ortamdan gelen KLAN çerçeveleri alır.
- Bu çerçevelerin kendini adresleyip adreslemediğini kontrol eder. Eğer kendini adresliyorsa içerisindeki bilgiyi çıkartıp ilgili seri arabirimine iletir. Aksi takdirde KLAN çerçeveyi elimine eder.
- CAN geliştirme kartı seri portundan gelen bilgiyi alır ve CAN mesajı oluşturarak, CAN veriyoluna iletir. CAN veriyoluna iletilen uzak CAN mesajın tanıtıcı alan bilgisini ve içeriğini LCD'de gösterir.



Şekil 6.3: CAN/IEEE 802.11b KLAN KAB ve blok diyagramı

6.3. Tasarlanan KAB Prototipinin Kullanıldığı Endüstriyel Ağ Uygulaması

Şekil 6.4 ve Şekil 6.5'de KAB'ın prototipinin kullanıldığı endüstriyel ağ uygulamasının blok diyagramı ve gerçekleştirilen uygulamanın resmi görülmektedir. Her iki şekilden de görüldüğü gibi endüstriyel ağ, her biri 2 CAN düğümü ve 1 KAB içeren 2 bağımsız CAN segmentten oluşmaktadır.



Şekil 6.4: KAB'ın kullanıldığı endüstriyel ağ uygulamasının blok diyagramı



Şekil 6.5: KAB'ın kullanıldığı endüstriyel ağ uygulaması

Tablo 6.1: Endüstriyel ağ uygulamasında kullanılan CAN mesaj bilgileri

| | Kaynak Dugum | Mesaj Tipi | Öncelik | Yük (bayt) | Hedef Dugum |
|-----------|--------------|------------|---------|------------|-------------|
| Segment A | 1 | Uzak | 1 | 8 | 3 |
| | | Yerel | 2 | 8 | 2 |
| | 2 | Uzak | 3 | 8 | 4 |
| | | Yerel | 4 | 8 | 1 |
| Segment B | 3 | Uzak | 5 | 8 | 1 |
| | | Yerel | 6 | 8 | 4 |
| | 4 | Uzak | 7 | 8 | 2 |
| | | Yerel | 8 | 8 | 3 |

CAN Veriyolu İletim Hızı: 500 Kbit/s
IEEE 802.11b Veri İletim Hızı: 11 Mbit/s

CAN sistemi, atanmış bir sistem olduğundan endüstriyel ağ içerisindeki her bir CAN düğümün hangi düğümle hangi şartlar altında haberleşeceği önceden belirlenmektedir. Tablo 6.1’de endüstriyel ağın ve ağda kullanılan CAN mesajların özellikleri verilmektedir. Tabloda belirtilen CAN mesajlar kullanıcı tarafından sistemde bulunan tuş takımları aracılığı ile rasgele bir anda üretilir.

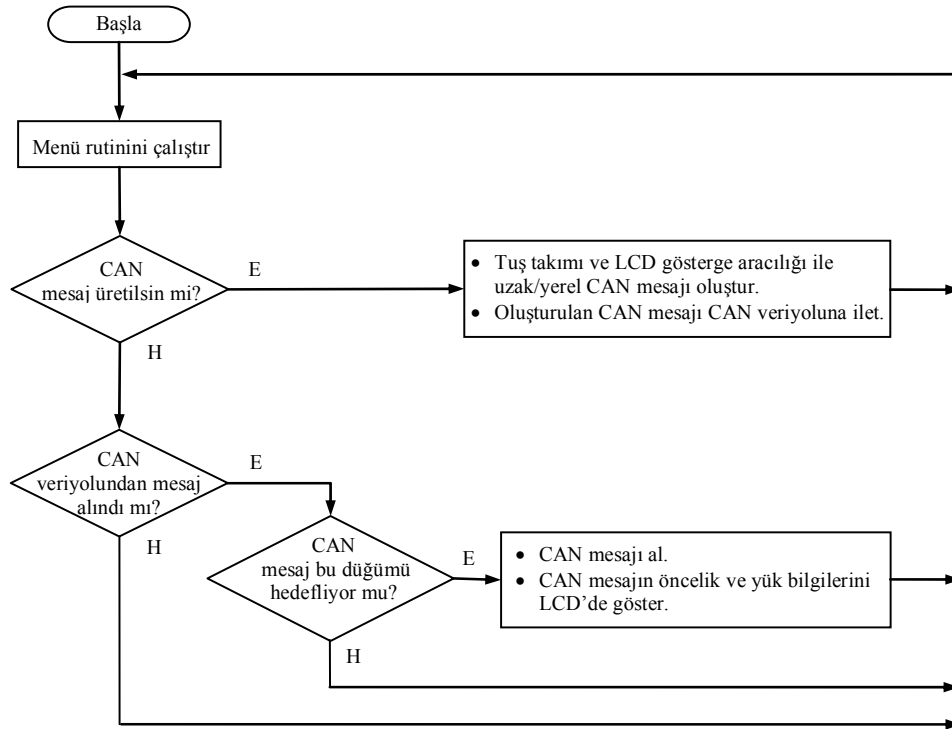
Endüstriyel ağ uygulamasında CAN düğümler tarafından üretilen uzak bir mesajın (örneğin CAN segment A’daki 1. düğümün 1 öncelik değerine sahip uzak mesajı) hedefine varmasında gerçekleşen işlemler kısaca şöyledir.

- Kaynak CAN segmentteki CAN denetleyicilere bağlı olan tuş takımı ve LCD gösterge aracılığı ile o denetleyicinin üreteceği mesaj tipi (uzak/yerel) ve mesaj içeriği (kullanıcı tarafından belirlenen 8 bayt’lık bilgi) girilerek oluşturulan CAN mesaj veriyoluna iletilir.
- KAB, CAN kartının CAN portu vasıtasıyla CAN veriyolundaki bu mesajı alır ve uzak mesaj olup olmadığını kontrol eder. Eğer uzak mesaj ise o mesajın öncelik bilgisini ve mesajın içeriğini kablosuz ortama iletmek üzere KAB’ın IEEE 802.11b KLAN birimine seri arabirimi vasıtasıyla iletir. KAB, IEEE 802.11b KLAN birimi kendisine gelen her bir bayt bilgiyi doğrudan kablosuz ortama iletir. Tabloda tanımlandığı gibi kablosuz ortamdan 1 CAN mesajı, 1 bayt öncelik bilgisi ve 8 bayt yük bilgisi olmak üzere ardışık olarak 9 bayt bilgi olarak iletilir. KAB’da bulunan LCD göstergede KAB’ın aldığı ve ilettiği CAN mesajın bilgileri gösterilir. Eğer CAN veriyolundan alınan mesaj yerel bir CAN mesajı ise elimine edilir.
- Hedef CAN segmentin KAB’ı, IEEE 802.11b KLAN birimi vasıtasıyla kablosuz ortamdan kendini adresleyen KLAN çerçeveleri alır ve içerisinden CAN mesajın bilgilerini çıkarır. KLAN birimi kablosuz ortamdan aldığı bilgiyi seri arabirimi vasıtasıyla KAB’ın CAN denetleyicisine seri port vasıtasıyla iletir. CAN denetleyici ardışık olarak aldığı 9 bayt bilgiyi (1 bayt öncelik ve 8 bayt yük) kullanarak ilgili CAN mesajı oluşturur ve CAN portu vasıtasıyla CAN veriyoluna iletir. Hedef KAB, kablosuz ortamdan aldığı ve CAN veriyoluna ilettiği CAN mesajın öncelik bilgisi ve içeriğini LCD göstergede gösterir.

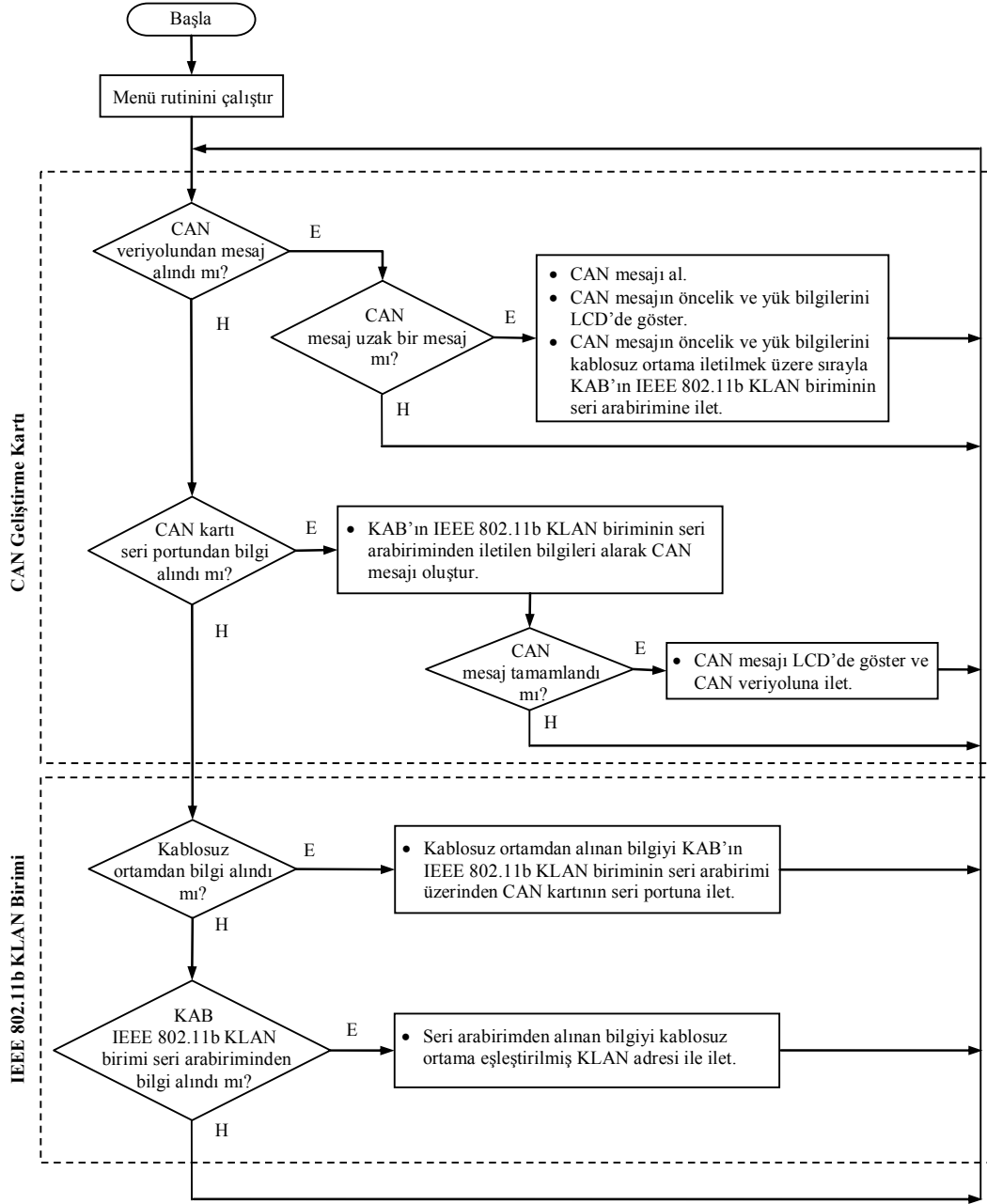
- Hedef CAN düğüm, CAN veriyolundaki kendisini adresleyen uzak CAN mesajı alır ve bu CAN mesajı LCD göstergede gösterir (Bayılmış ve diğ. 2006).

CAN/IEEE 802.11b KLAN KAB prototipi kullanılan endüstriyel ağ uygulamasındaki CAN düğümlerin ve Kablosuz Arabağlaşım Biriminin çalışması sırasıyla Şekil 6.6 ve Şekil 6.7’deki akış diyagramlarında özetlenmektedir.

Endüstriyel ağ uygulamasının gerçekleştirilmesinde ve 8051 tabanlı CAN geliştirme kartının programlanmasında Keil μ Vision2 geliştirme yazılımı kullanılmıştır. 8051 tabanlı mikrodenetleyiciler de C programla ve Keil μ Vision2 geliştirme yazılımının kullanımı için Özcerit ve diğ. (2005)’e bakılabilir.



Şekil 6.6: CAN geliştirme kartının çalışma akış diyagramı



Şekil 6.7: KAB çalışma akış diyagramı

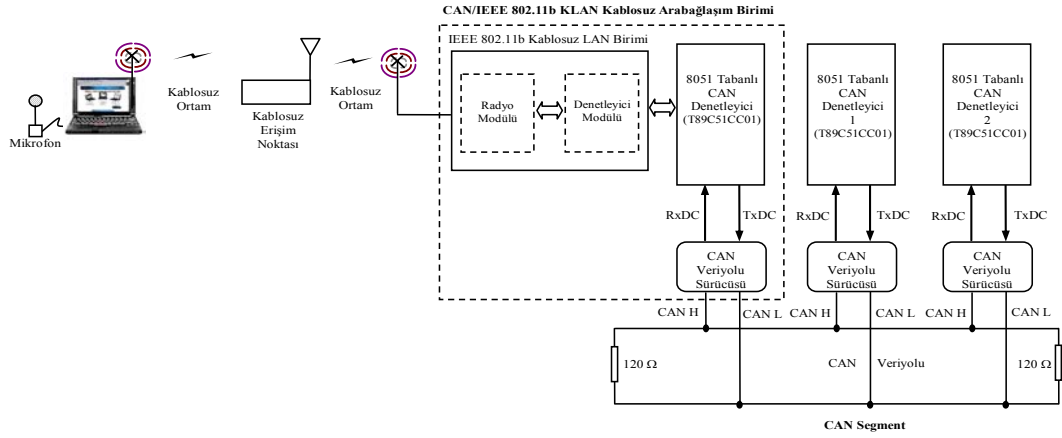
6.4. CAN Kullanan Dağıtık Kontrol Sisteminin Tasarlanan KAB Aracılığıyla Kablosuz Olarak Sesli Kontrolü

Şekil 6.8’de CAN kullanılarak kontrol edilen bir model arabanın KAB vasıtasıyla kablosuz olarak ses ile kontrolü uygulamasının blok diyagramı ve Şekil 6.9’da gerçekleştirilen bu uygulamanın resmi görülmektedir. Uygulama beş temel kısımdan oluşmaktadır.

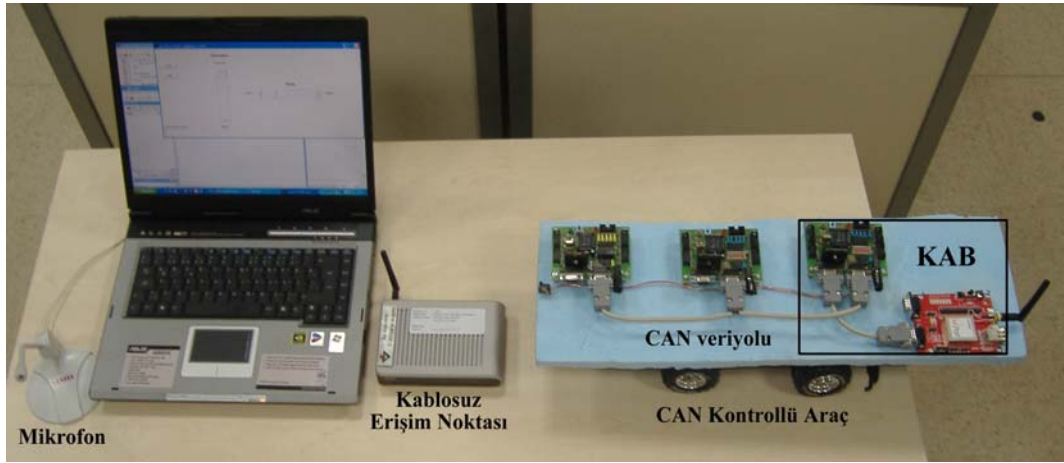
- CAN kullanılarak kontrol edilen model araba: Model arabanın kontrolü, iki CAN düğümünden oluşan bir CAN ağ ile gerçekleştirilmiştir. CAN sisteminde arabanın ileri, geri, sağ ve sol yönde hareketlerini ve durmasını sağlayan 5 CAN mesajı tanımlanmıştır (Tablo 6.2).
- KAB: Model arabanın kontrolünü sağlayan CAN ağa kablosuz erişimi sağlamaktadır.
- Kablosuz erişim noktası: Kablosuz Ethernet kartına sahip bilgisayar ve KAB’ın iletişimini sağlamaktadır.
- Kablosuz Ethernet kartına sahip bilgisayar: Bilgisayar ses tanıma yazılımının çalıştırılması ve KAB’a kablosuz olarak erişilmesi için kullanılmaktadır.
- Ses tanıma yazılımı: Ses tanıma yazılımı olarak Kelebekler (2006) tarafından MATLAB’da gerçekleştirilen çalışma kullanılmıştır. Bu yazılımda önce kullanıcının ses bilgileri ve komut (ileri, geri vb.) tanımlamaları sisteme eklenir. Girilen bir ses bilgisinin komuta karşılık gelip gelmediği kontrol edilir. Eğer gelirse o komut ile eşleşen değer kablosuz ortama iletilir.

Tablo 6.2: Model arabanın kontrolü için kullanılan CAN mesajların bilgileri

| Hedef CAN Düğüm | Ses Karşılığı | Öncelik | Yük (bayt) |
|-----------------|---------------|---------|------------|
| 1 ve 2 | Dur | 1 | 1 |
| 1 | İleri | 2 | 1 |
| 1 | Geri | 3 | 1 |
| 2 | Sağ | 4 | 1 |
| 2 | Sol | 5 | 1 |



Şekil 6.8: CAN ile kontrol edilen model arabanın KAB vasıtasıyla kablosuz olarak ses ile kontrolü uygulaması blok diyagramı

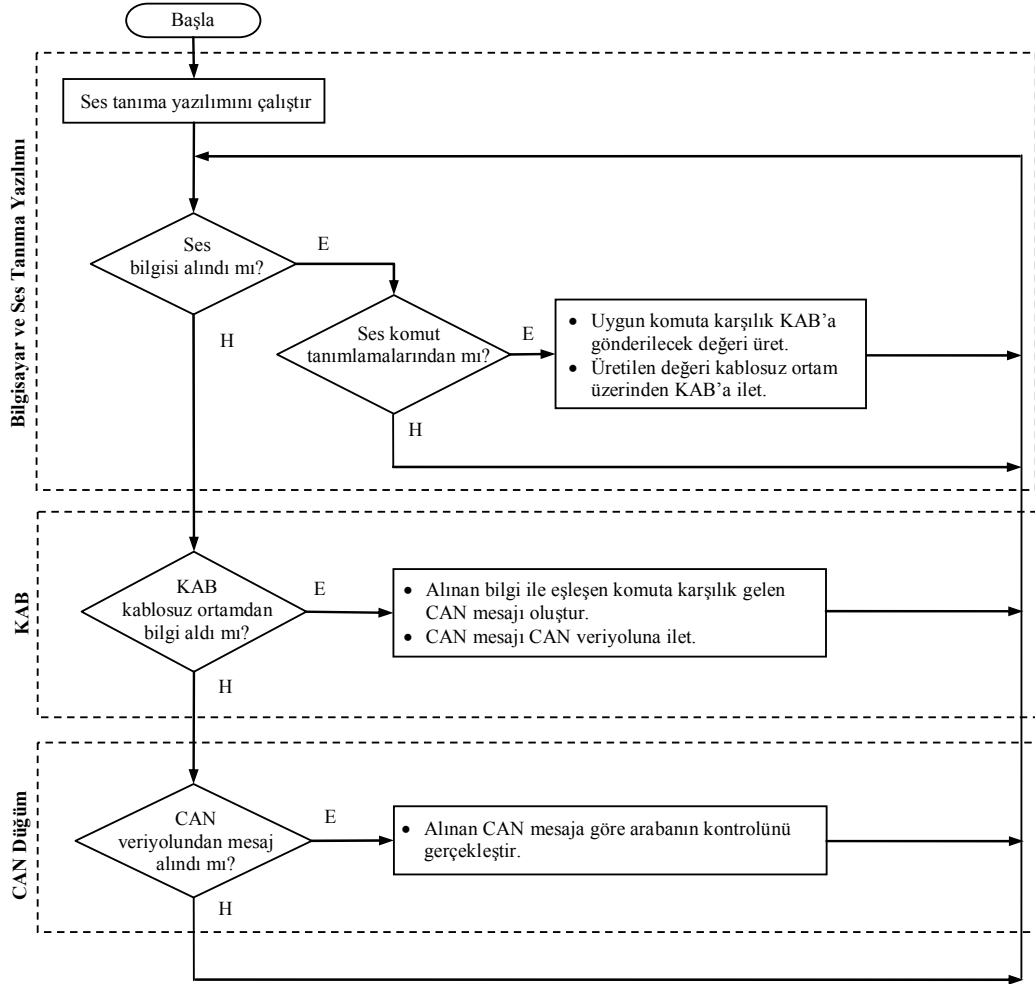


Şekil 6.9: CAN ile kontrol edilen model arabanın KAB vasıtasıyla kablosuz olarak ses ile kontrolü uygulaması

CAN ile kontrol edilen model arabanın KAB vasıtasıyla kablosuz olarak ses ile kontrolü uygulamasının çalışması Şekil 6.10'daki akış diyagramında özetlenmektedir. Uygulamanın çalışması kısaca şöyledir:

- Kablosuz Ethernet'e sahip bilgisayarda MATLAB'da gerçekleştirilen ses tanıma yazılımı çalıştırılır.
- İlk kullanım için ses tanıma yazılımında kullanıcının ses bilgileri ve model arabanın kontrolü için gerekli komutlar tanımlanır.
- Mikrofon aracılığı ile alınan sesin kullanıcıya ait ve tanımlanan komutlardan biri olduğu belirlenirse yazılım ilgili komutla eşleştirilen bir değer üretir ve bu değeri kablosuz Ethernet üzerinden kablosuz ortama iletir.

- KAB, kablosuz ortamdan gelen bilgiyi alır, bu bilgiye karşılık gelen CAN mesajı oluşturur ve CAN veriyoluna iletir.
- Uygun CAN denetleyici tarafından CAN veriyolundaki CAN mesajı alınır ve model arabanın kontrolü gerçekleştirilir.



Şekil 6.10: CAN ile kontrol edilen model arabanın KAB vasıtasıyla kablosuz olarak ses ile kontrolü uygulamasının çalışma akış diyagramı

6.5. Sonuç

Önceki bölümde ayırık olaya dayalı bilgisayar benzetim yöntemi kullanılarak modellenmesi ayrıntılı bir şekilde sunulan KAB'ın, bu bölümde gerçekleştirilen, basit ve genel bir prototipi üzerinde durulmuştur.

KAB'ın prototipi, CAN geliştirme kartı, IEEE 802.11b KLAN birimi ve LCD göstergeden oluşmaktadır. CAN geliştirme kartı, KAB'ın CAN veriyolu ile iletişimini ve CAN mesajların kablosuz ortama iletilmesi ya da kablosuz ortamdan alınan bilginin CAN veriyoluna iletilmesi için gerekli işlemleri yerine getirmektedir. IEEE 802.11b KLAN birimi, seri arabirimlerinden aldığı bilgiyi kablosuz ortama ve kablosuz ortamdan aldığı bilgiyi ilgili seri arabirimine iletmektedir. LCD gösterge ise KAB üzerinden iletilen uzak CAN mesajların bilgilerini gösterir.

KAB'ın farklı uygulamalarda farklı amaçlarla kullanılabilceğini göstermek için iki uygulama gerçekleştirilmiştir. Birinci uygulamada iki bağımsız CAN segmentten oluşan ve KAB vasıtasıyla haberleşen küçük bir endüstriyel ağ gerçekleşmiştir. İkinci uygulama ise KAB vasıtasıyla mevcut CAN temelli kontrol sistemlerinin kablosuz ve ses tanıma sistemi ile kontrol edilmesini içermektedir.

İki ayrı birimden oluşan KAB, bir ileri çalışma olarak DSP ve FPGA gibi yapılar yardımıyla tek bir ürün (çip) şeklinde de üretilebilir.

7. SONUÇLAR VE ÖNERİLER

Sahayolu sistemleri, BBİ zincirinin daha düşük seviyelerindeki algılayıcı, eyleyici, PLC, mikrodenetleyici gibi dağıtık kontrol cihazların arabağlantısı için tasarlanılmıştır. Bu sistemler, yalnızca bu seviyedeki cihazlar arasındaki kablolamayı azaltmakla kalmaz, cihazlar arasında gerçek zamanlı iletişimi sağlayan bir ağ yapısı da sunar.

Sahayolu ağları, kurulum ve bakım kolaylığı, maliyet, güvenilirlik, yeni denetleyici ekleme/çıkarma esnekliği, yüksek performans, kritik zamanlı haberleşme gereksinimlerini karşılaması gibi avantajları sayesinde fabrika otomasyonu, robotik uygulamalar, güvenlik sistemleri, otomotiv sistemleri, bina otomasyonu gibi birçok endüstriyel kontrol uygulamalarında yaygın olarak kullanılmaktadır.

Günümüzde saha seviyesindeki iletişim için geliştirilmiş çok sayıda sahayolu ağı bulunmaktadır. Sahayolu sistemlerinin en başarılılarından olan CAN, başlangıçta yalnızca otomotiv uygulamaları için tasarlanmış olsa da yüksek hız, düşük maliyet, yüksek başarımlı ve üstün karakteristik özelliklerinden dolayı birçok dağıtık endüstriyel kontrol uygulamalarında yaygın olarak kullanılmaktadır.

Kablosuz teknolojilerin ofis ve bilgisayar ağ haberleşmesindeki başarısı, sahayolu sistemlerinin kullanıldığı endüstriyel uygulamalarda hareketli cihazların kullanımının artması, kablolu ve kablosuz sistemlerin birlikte çalışabilirlik gereksinimi gibi nedenler endüstriyel sistemlerde kablosuz haberleşme desteğinin gerekliliğini ortaya koymaktadır.

Fabrika otomasyonu ve endüstriyel kontrol uygulamalarında kablosuz iletişimin sağlayacağı birçok yarar vardır. Bunlardan bazı önemlileri, kablo gereksiniminin olmamasından kaynaklanan sistem kurulum ve bakımında sağlanacak önemli maliyet ve zaman kazanımı, sistemlerde kolay konfigürasyon değişikliği ve yeniden

yapılandırma, tehlikeli alanlarda kolay kurulum, mevcut sabit sistemlerin hareketli ve diğer sabit sistemler ile kablosuz ortam üzerinden haberleşmesi şeklinde sıralanabilir.

Günümüzde IEEE 802.11 KLAN, Bluetooth, ZigBee, KATM gibi mevcut ve geliştirilmekte olan kablosuz teknolojilerin birçok farklı sahayolu sisteminde kullanımı üzerine çalışmalar/araştırmalar yapılmaktadır. Ancak çalışmaların birçoğunda, özellikle endüstriyel üretim ortamlarında Ethernet'in yaygın kullanımı dikkate alındığında, düşük maliyet, yaygın kullanım, yüksek veri iletim hızı ve Ethernet ile sağladığı kolay entegrasyon gibi özellikler sebebiyle IEEE 802.11 KLAN kablosuz iletişim teknolojisi tercih edilmektedir.

Bu tez çalışmasında, CAN temelli dağıtık kontrol sistemlerinin IEEE 802.11b KLAN vasıtasıyla kablosuz ortam üzerinden haberleşmesini sağlayan;

- a. CAN/IEEE 802.11b Kablosuz Arabağlaşım Birimi tasarlanmıştır (Bölüm 4),
 - i. Benzer olmayan iki ağın arabağlaşımını sağlayan KAB'ın tasarımında, her iki ağın farklı çerçeve biçimlerine ve boyutlarına sahip olmaları, farklı öncelik mekanizmaları kullanmaları, farklı yönlendirme algoritması koşturmaları, farklı topoloji ve ortam erişim tekniği kullanmalarından kaynaklanan problemlere çözümler önerilmiştir.
 - ii. Fiziksel gerçekleştirme aşaması için, daha önce yapılan CAN/Ethernet (Ekiz, 1997, Ekiz ve diğ., 1996, 1997), CAN/ATM ve PROFIBUS/ATM (Özçelik, 2002) yerel köprü tasarım yöntemleri esas alınarak gerçekleştirilen CAN/IEEE 802.11b KAB tasarım mimarisi ayrıntılı olarak sunulmuştur.
- b. Bilgisayarlı benzetim yöntemiyle modellenmiş ve SAE Benchmark'a göre başarımlı analizi sunulmuştur (Bölüm 5),
 - i. Tasarım mimarisi esas alınarak OPNET Modeller yazılımında bilgisayarlı benzetim modeli çıkartılmıştır. KAB, CAN ve IEEE 802.11b KLAN kısımlarından oluştuğundan, her iki kısmın proses modelleri ve algoritmaları ayrıntılı olarak verilmiştir.
 - ii. Modellenen KAB'ın kullanıldığı örnek bir endüstriyel ağ uygulaması benzetime tabi tutulmuş ve elde edilen sonuçlar KAB'ın kullanılabilirliği açısından karşılaştırmalı olarak değerlendirilmiştir. Endüstriyel kontrol uygulamalarında

yaygın olarak tercih edilen 100 Kbit/s CAN veri iletim hızında, CAN veriyolunun yaklaşık %58 kullanım oranında, saniyede 1720 mesajın transfer edildiği uygulamada tüm CAN mesajların SAE Benchmark'da belirtilen varma sınır değerini aşmadığı görülmüştür. Aynı zamanda KAB içerisinde meydana gelen gecikmenin, uzak CAN mesajların uçtan uca toplam gecikmesi içerisinde ortalama %31 katkısı olduğu ve paket kaybının 50 Kbit/s CAN veri iletim hızının altına inildiğinde ortaya çıktığı sonuç olarak elde edilmiştir. Bu sonuçlar, KAB'ın CAN ve IEEE 802.11b KLAN sistemler arasındaki iletişim için yeterli bir hizmet sağladığını göstermektedir.

- c. Temel bir prototipi gerçekleştirilerek kullanılabilirliği değişik kontrol uygulamaları üzerinde gösterilmiştir (Bölüm 6).
 - i. Karaca (2003)'de sunulan CAN geliştirme kartı ve IEEE 802.11b standardını destekleyen kablosuz birim (Wiport) kullanılarak temel bir prototip gerçekleştirilmiştir.
 - ii. Her biri 2 CAN düğümü ve 1 KAB içeren 2 bağımsız CAN segmentin kablosuz ortam üzerinden haberleştiği endüstriyel ağ uygulaması gerçekleştirilmiştir.
 - iii. CAN kullanılarak kontrol edilen bir model arabanın KAB vasıtasıyla kablosuz olarak ses ile kontrolü uygulaması gerçekleştirilmiştir.

Sunulan tez çalışmaları, CAN sahayolu sistemleri için özetle şu önemli çözümleri sağlamaktadır:

- CAN sahayolu sistemlerinin mesafe problemi giderilmiştir,
- CAN temelli dağıtık kontrol sistemlerinin IEEE 802.11b KLAN üzerinden birbirleri ile haberleşmesi sağlanmıştır,
- CAN düğümlerin Ethernet ya da IEEE 802.11b KLAN ağlar ile haberleşmesi sağlanmıştır.

TARTIŞMA VE ÖNERİLER

Bu tez çalışmasında CAN/IEEE 802.11b KLAN Kablosuz Arabağlaşım Birimi'nin tasarım, modelleme ve prototipinin geliştirilmesinden elde edilen sonuçlar ve katkılar doğrultusunda gelecekte yapılabilecek çalışmaları tasarım ve modelleme ile fiziksel gerçekleştirme olarak iki grup altında toplayabiliriz.

Tasarım ve modelleme olarak;

CAN sahayolu sistemi dışında endüstride kullanım halinde olan PROFIBUS, WorldFIP, P-NET gibi sahayolu sistemlerinin IEEE 802.11b kablosuz standardı üzerinden haberleşmesini sağlayacak Kablosuz Arabağlaşım Birimleri tasarlanabilir. Bu tasarımlar gerçekleştirilirken tez çalışmasındaki tasarım mimarisi ve OPNET yazılımında gerçekleştirilen KAB'ın KLAN tarafının modelinden yararlanılabilir.

CAN, PROFIBUS gibi farklı sahayolu sistemlerinin kablosuz ortam üzerinden haberleştiği heterojen ağ uygulamaları gerçekleştirilebilir.

Bulanık mantık, genetik algoritma gibi yapay zeka algoritmaları kullanılarak CAN/IEEE 802.11b KLAN KAB'ın başarımı en iyileştirilebilir.

Sahayolu sistemleri için güvenilirlik önemli bir kıstastır; bu doğrultuda, endüstriyel ağların kablosuz haberleşmesinin güvenilirliğini arttıracak çalışmalar yapılabilir.

CAN temelli dağıtık kontrol sistemlerinin IEEE 802.11b KLAN yerine TDMA/FDD gibi servis kalitesi desteği sağlayan ya da Bluetooth, ZigBee gibi diğer kablosuz teknolojiler üzerinden haberleşmesini sağlayan Kablosuz Arabağlaşım Birimleri gerçekleştirilerek bu tez çalışmasının sonuçları ile karşılaştırılabilir.

Fiziksel gerçekleştirme olarak;

Tasarlanan ve benzetim yöntemi ile modellenen CAN/IEEE 802.11b KLAN KAB'ın, FPGA gibi teknolojiler kullanılarak tek bir ürün olarak fiziksel gerçekleştirilmesi yapılabilir.

Tez çalışmasında sunulan prototip kullanılarak robot kontrolü, bina otomasyonu gibi kablosuz CAN temelli dağıtık kontrol uygulamaları gerçekleştirilebilir.

Yine prototipin kullanıldığı internet üzerinden kontrol edilebilen uygulamalar geliştirilebilir.

KAYNAKLAR

Aad, I., Castelluccia, C., “Priorities in WLANs”, *Computer Networks*, Vol. 41, 505–526, (2003).

Alves, F., “Real–Time Communications over Hybrid Wired/Wireless PROFIBUS–Based Networks”, Doktora Tezi, *Porto University*, Portekiz, (2003).

ANSI/IEEE Std 802.11, “Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications”, *IEEE Standards 802.11*, 70–90, 1999.

Aydoğan, T., “WorldFIP/ATM Yerel Köprü Tasarımı ve CAN/PROFIBUS/WorldFIP Ağlarının ATM Omurga Üzerinden Arabağlaşımı”, Doktora Tezi, *Sakarya Üniversitesi Fen Bilimleri Enstitüsü*, Sakarya, (2005).

Bayılmış, C., “Kablosuz Bilgisayar Ağlarının Performans Analizi”, Yüksek Lisans Tezi, *Sakarya Üniversitesi Fen Bilimleri Enstitüsü*, (2003a).

Bayılmış, C., Ertürk, I., Çeken, C., “Wireless Local Area Network Application For Non-Real Time Data Transfer”, *International Journal of Computational Intelligence*, Vol.1, No.1, 131–133 (2003b).

Bayılmış, C., Ertürk, I., Çeken, C., “Kablosuz Bilgisayar Ağlarının Karşılaştırılmalı İncelemesi”, *Gazi Üniversitesi Politeknik Dergisi*, Cilt 7, Sayı 3, 201–210, (2004a).

Bayılmış, C., Ertürk, I., Çeken, C., “A Comparative Performance Evaluation Study of IEEE 802.3 Wired and IEEE 802.11 Wireless LANs for Multimedia Data Traffic”, *Journal of Naval Science and Engineering*, 2, 1–12 (2004b).

Bayılmış, C., Ertürk, I., Çeken, C., “Wireless Interworking Independent CAN Segments”, *Lecture Notes in Computer Science*, Vol. 3280, 299–310, (2004c).

Bayılmış, C., Ertürk, I., Çeken, C., “Extending CAN Segments With IEEE 802.11 WLAN”, *The 3rd ACS/IEEE International Conference on Computer Systems and Applications AICCSA–05*, Egypt, (2005a).

Bayılmış, C., Ertürk, I., Çeken, C. and Özçelik, I., “Modelling Controller Area Networks Using Event Discrete Simulation Technique”, *Complex Computing Networks CCN2005*, İstanbul, 353–358, (2005b).

Bayılmış, C., Ertürk, I., Çeken, C., “A New Solution for Extending CAN Segments Using IEEE 802.11 WLAN”, *Journal of The Faculty of Engineering and Architecture of Gazi University*, Vol 20, No 2, 197–204, (2005c).

Bayılmış, C., Ertürk, I., Çeken, C., Özçelik, I., Karaca, A., “Implementation of a CAN/IEEE 802.11b WLAN/CAN Internetworking System Using A New Designed WIU”, *11th International CAN Conference*, Sweden, (2006).

Bing, B., “High-Speed Wireless ATM and LANs”, *Artech House Mobile Communications Library*, 1–102, (2000).

BoschAH, “Automotive Handbook”, 3. Edition, *Robert Bosch GmbH*.

CANApp, CAN Application Fields [online], CAN in Automation, <http://www.datamicro.ru/can/standards/pdf/CANappl.pdf> (**Ziyaret tarihi: 10 Mayıs 2006**).

“CAN Specification V 2.0”, *Robert Bosch GmbH*, (1991).

“CAN Newsletter”, *CAN in Automation*, 34–38, (September 2005).

Cauffriez, L., Ciccotelli, J., Conrard, B., Bayart, M., “Design of intelligent distributed control systems: a dependability point of view”, *Reliability Engineering and System Safety*, 84, 19–32, (2004).

Cavalieri, S., Pano, D., “A novel solution to interconnect FieldBus systems using IEEE wireless LAN technology”, *Computer Standards & Interfaces*, 9–23, (1998).

Çeken, C., “Kablosuz ATM Kullanarak Servis Kalitesi Desteği Sağlanmış Gerçek Zamanlı Veri Transferi”, Doktora Tezi, *Kocaeli Üniversitesi Fen Bilimleri Enstitüsü*, (2004).

Çeken, C., Ertürk, I., Bayılmış, C., “Wireless Networks for Real-Time Multimedia Communications”, *Broadband Wireless and WiMAX, Comprehensive Report by International Engineering Consortium (IEC)*, (2004).

De Pellegrini, F., Miorandi, D., Vitturi, S., Zanella, A., “On the Use of Wireless Networks at Low Level of Factory Automation Systems”, *IEEE Transactions On Industrial Informatics*, Vol. 2, No. 2, 129–143, (2006).

Ekiz, H., Kutlu, A., Powner, E., “Design and Implementation of a CAN/Ethernet Bridge”, *Proc. Of 3.International CAN Conference*, Paris, France, 11–20, 1-2 October (1996).

Ekiz, H., “Design, Implementation, and Performance Analysis of CAN/CAN and CAN/Ethernet Bridges”, Doktora Tezi, *University of Sussex*, Brighton, England (1997).

Ekiz, H., Powner, E., Kutlu, A., “Performance Analysis of a Ethernet/CAN Bridge”, *Proc. Of IEEE SICON’97 Conference*, Singapore, 71–85, 14–17 April (1997).

Ertürk, İ., “Internetworking Between ATM LANs and Legacy LANs Over ATM Networks”, DPhil. Thesis, *Sussex University, The School of Engineering and Information Technology*, England, (2000).

Ertürk, I., “Transporting CAN Messages over WATM”, *Lecture Notes in Artificial Intelligence*, Vol. 2639, 724–729, (2003).

Ertürk I., “A New Method for Transferring CAN Messages Using Wireless ATM”, *Journal of Network & Computer Applications*, 28 (1), 45–56, (2005).

Farsi, M., Ratcliff, K., Barbosa, M., “An overview of Controller Area Network”, *Computing & Control Engineering Journal*, 10, 113–120, (1999).

Gast, M., “802.11 Wireless Networks: The Definitive Guide”, Second Edition, *Q'Reilly*, (2005).

Karaca, A., “Denetleyici Alan Ağları Kullanılarak (CAN) Bina Güvenlik Uygulaması”, Yüksek Lisans Tezi, *Sakarya Üniversitesi Fen Bilimleri Enstitüsü*, Sakarya, (2003).

Kelebekler, E., “Otomobil İçerisindeki Cihazların Kontrolüne Yönelik Konuşmacıdan Bağımsız Konuşmacı Tanıma Sisteminin Gerçek Zamanlı Uygulaması”, Yüksek Lisans Tezi, *Kocaeli Üniversitesi Fen Bilimleri Enstitüsü*, Kocaeli, (2006).

Koulamas, C., Koubias, S., Papadopoulos, G., “Using Cut-Through Forwarding to Retain the Real-Time Properties of Profibus Over Hybrid Wired/Wireless Architectures”, *IEEE Transactions on Industrial Electronics*, Vol. 51, No. 6, 1208–1217, (2004).

Kunert, O., “Interconnecting Field-buses Through ATM”, *Proceedings of 2nd IEEE Workshop on Factory Communication System*, Barcelona, Spain, 1–7, (1997).

Kunert, O., Zitterbart, M., “Performance Aspects of PROFIBUS Segments Interconnected Through ATM”, *Symposium on Broadband European Networks'98*, Zurich, 1–8, 18–20 May (1998).

Kutlu, A., Ekiz, H., Powner, E., T., “Performance analysis of MAC protocols for wireless control area network”, *Proc. Int. Symp. Parallel Architectures, Algorithms, and Networks*, 494–499, (1996a).

Kutlu, A., Ekiz, H., Powner, E., T., “Wireless control area network”, *Networking Aspects of Radio Communication System*, 1–4, (1996b).

Kutlu, A., “CAN Veri Haberleşme Sistemi ve Askeri Uygulama Alanları”, *Silahlı Kuvvetler Dergisi*, Genel Kurmay Askeri Tarih ve Stratejik Etüt Başkanlığı Yayınları, Sayı 362, 55–57, (1999).

Lawrenz, W., “CAN System Engineering:from Theory to Practical Applications”, *Springer-Verlag*, 1–289, (1997).

Levillain, P., “Wireless LAN for Enterprises”, *Alcatel Telecommunications Review*, Vol. 4, 287–291, (2002).

Miorandi, D., Vitturi, S., “Hybrid wired/wireless implementations of Profibus DP: a feasibility study based on Ethernet and Bluetooth”, *Computer Communications*, 27, 946–960, (2004).

Miorandi, D., Vitturi, S., “A wireless extension of Profibus DP based on the Bluetooth radio system”, *Ad Hoc Networks*, 3, 479–494, (2005).

Navet, N., Gaujal, B., “Fault Confinement Mechanisms on CAN: Analysis and Improvements”, *IEEE Transaction on Vehicular Technology*, Vol. 54, Issue 3, 1103–1113, (2005).

Navet, N., Song, Y., “Validation of in-vehicle real-time applications”, *Computers in Industry*, 46, 107–122, (2001).

Nicopolitidis, P., Obaidat, M., S., Papadimitriou, G., I., Pomportsis, A., S., “Wireless Networks”, *WILEY*, 239–269, (2003).

OPNET, “OPNET Modeler 11.5 Documentation”, *OPNET Technologies*, Release 11.5, (2006).

Öner, D., “Bilgisayar Ağları”, *Papatya Yayıncılık*, 255–295, (2003).

Örencik, B., Çölkesen R., “Bilgisayar Haberleşmesi ve Ağ Teknolojileri”, *Papatya Yayıncılık*, 51–62, (2002).

Özcerit, A., Çakıroğlu, M, Bayılmış, C., “8051 Mikrodenetleyici Uygulamaları”, *Papatya Yayıncılık*, (2005).

Özçelik, İ., “CAN/ATM ve PROFIBUS/ATM Yerel Köprülerinin Tasarımı”, Doktora Tezi, *Sakarya Üniversitesi Fen Bilimleri Enstitüsü*, Sakarya, (2002).

Özçelik, İ., Ekiz, H., “Design, Implementation and Performance Analysis of The PROFIBUS/ATM Local Bridge”, *Computer Standart & Interface*, 26, 329–342, (2004).

Özçelik, İ., Ekiz, H., “Building an interconnection between PROFIBUS and ATM networks”, *Journal of Network and Computer Application*, baskı aşamasında (2006).

Patzke, R., “Fieldbus basics”, *Computer Standard & Interfaces*, 19, 275–293, (1998).

- Peraldi, M., Decotignie, J., “Combining Real Time Features of LANs FIP and CAN”, *2nd International CAN Conference Proceeding*, London, (1995).
- Pinho, L., M., Vasques, F., “Reliable Real-Time Communication in CAN Networks”, *IEEE Transactions on Computers*, Vol. 52., No. 12, 1594–1607, (2003).
- Purdy, D., “Target’s Common Digital Architecture”, *5th International CAN Conference*, (1998).
- Tenruh, M., Tarimer, İ., “A New Solution to Extend Controller Area Networks by Bridging Over ATM”, *Journal of Polytechnic, University of Gazi*, Vol. 8, No. 1, 37–41, (2005).
- Thomesse, J., P., “Fieldbuses and interoperability”, *Control Engineering Practice*, 7, 81–94, (1999).
- Thomesse, J., P., “Fieldbus Technology in Industrial Automation”, *Proceedings of The IEEE*, Vol. 93, No. 6, 1073–1101, (2005).
- Tindell, K., Burns, A., Wellings, J., A., “Calculating Controller Area Network (CAN) Message Response Times”, *Control Engineering Practice*, Vol. 3, No.8, 1163–1169, (1995).
- Tovar, E., “Supporting Real-Time Communications with Standard Fieldbus Networks”, Doktora Tezi, *Porto University*, Portekiz, (1999).
- Vitturi, S., “On The Use Of Ethernet At Low Level Of Factory Communication Systems”, *Computer Standards&Interfaces*, 23, 267–277, (2001).
- Vitturi, S., “DP–Ethernet: the Profibus DP protocol implemented on Ethernet”, *Computer Communications*, 26, 1095–1104, (2003).
- Waizmann, G., Sternert, F., “CAN–base for new applications in vehicle”, *1. International CAN Conference Proceeding*, Germany, (1994).
- Willig, A., “An architecture for wireless extension of PROFIBUS”, *Industrial Electronics Society 2003, IECON’03*, 2369–2377, (2003a).
- Willig, A., “Polling–Based MAC Protocols for Improving Real–Time Performance in a Wireless PROFIBUS”, *IEEE Transactions on Industrial Electronics*, Vol. 50, No. 4, 806–817, (2003b).
- Willig, A., Matheus, K., Wolisz, A., “Wireless Technology in Industrial Networks”, *Proceedings of The IEEE*, Vol. 93, No. 6, 1130–1151, (2005).
- Wiport Guide, 2006, Documentation — Embedded Device Servers [online], <http://www.lantronix.com/support/documentation.html> (**Ziyaret tarihi: 20 Haziran 2006**).

EK-A: CAN_MAC MODÜLÜ PROSES MODELİNE AİT PROGRAM KODLARI

Bu ekte, CAN düğümün CAN_MAC modülüne ait proses modelinin C programlama dilinde yazılan kodları verilmektedir.

“Basla” Durum Makinesi:

```
/* node'un ve nesnenin id'sini öğren */
my_id=op_id_self();
my_node_id=op_topo_parent(my_id);

/* alt kuyruk l'i temizle */
op_subq_flush(1);

/* nesnenin özelliklerini al */
op_ima_obj_attr_get(my_id, "Hedef_ID",&H_ID);
op_ima_obj_attr_get(my_id, "Priority", &priority);
op_ima_obj_attr_get(my_id, "Deadline_time", &dead_time);
j=OPC_FALSE;

//op_ima_obj_attr_get(my_node_id, "Source.Start Time", &baslama);
//op_ima_obj_attr_get(my_node_id, "Source.Stop Time", &bitis);

/* bitis zamani ile baslama zamanini karsilastir */
//if (bitis <= baslama)
//    hata_mesaj ("bitis zamani baslama zamanindan büyük olmalı",OPC_NIL, OPC_NIL);

/* Transmitter veri iletim hizini al */
tx_id = op_topo_assoc(my_id, OPC_TOPO_ASSOC_OUT, OPC_OBJTYPE_BUTX, 0);
comp_id = op_topo_child(tx_id, OPC_OBJTYPE_COMP,0);
tx_child_id = op_topo_child(comp_id, OPC_OBJTYPE_BUTXCH, 0);
op_ima_obj_attr_get(tx_child_id, "data rate", &tx_data_rate);

/* bit iletim zamanini hesapla */
tbit = 1/tx_data_rate;

/* hata sayicilerini kur */
tec=120;
rec=120;

/* local istatistikleri belirle */
gonderilen_pk_stat = op_stat_reg ("Gonderilen_Paket", OPC_STAT_INDEX_NONE,
OPC_STAT_LOCAL);
alinan_pk_stat = op_stat_reg ("Alinan_Paket", OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);
gercek_al_stat = op_stat_reg ("Gercek Alinan", OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);
pk_id_stat = op_stat_reg ("id", OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);
gon_ack_stat = op_stat_reg ("Gonderilen_ACK_Sayisi", OPC_STAT_INDEX_NONE,
OPC_STAT_LOCAL);
```

```

tec_stat = op_stat_reg ("Hata Sayici.Tec", OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);
rec_stat = op_stat_reg ("Hata Sayici.Rec", OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);
hata_mod_stat = op_stat_reg ("Hata Modu", OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);
ete_stat = op_stat_reg ("ETE Delay", OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);
packet_loss_stat = op_stat_reg ("Packet Loss", OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);

```

```

/* istatistikleri yaz */
op_stat_write (tec_stat, tec);
op_stat_write (rec_stat, rec);

```

“rx’den” Durum Makinesi:

```

/* Bus_Receiver'dan gelen paketi al */
if (op_intrpt_strm() == Rx_In_Strm)
{
    pkptr = op_pk_get (Rx_In_Strm);
    if (pkptr != OPC_NIL)
    {
        op_pk_format (pkptr, pk_format);
        if (strcmp(pk_format, "can_pk_data_20A")==0)
        {
            /* paketin priority'sini al */
            op_pk_nfd_get (pkptr, "Priority", &pk_priority);
            /* pakete bufferda siralamak icin deger ata */
            op_pk_priority_set (pkptr, pk_priority);
            /* paketi buffera gönder */
            op_subq_pk_insert(1, pkptr, OPC_QPOS_TAIL);
            /* arbitration islemi icin altkuyruğa atilan paketlerin priority istatistigini tut */
            op_stat_write(pk_id_stat, pk_priority);
            /* Arbitration süresi sonunda interrupt kur */
            if (j==OPC_FALSE )
            {
                //paketin alinma zamani
                op_intrpt_schedule_self(op_sim_time() + 11*tbit, Intrpt_Arbitration);
                j=OPC_TRUE;
            }
            alinan_pk_sayisi++;
            op_stat_write(alinan_pk_stat, alinan_pk_sayisi);
        }
        if (strcmp(pk_format, "can_pk_error")==0)
        {
            /* hata tipini al (aktif hata / pasif hata bayragi) */
            op_pk_nfd_get(pkptr, "Hata Turu", &hata_turu);
            op_pk_nfd_get(pkptr, "ID", &hata_id);
            if (hata_id == priority) // hatali paketi gonderen düğüm ise
            {
                tec=tec+8;
                op_stat_write(tec_stat, tec);
            }
            rec++; // hatali paketi gonderen düğüm degilse
            op_stat_write(rec_stat, rec);
            op_pk_destroy(pkptr);
        }
        if (strcmp(pk_format, "can_pk_ack")==0)
        {
            /* Alinan ack bu düğüme ait ise */
            op_pk_nfd_get(pkptr, "ACK_ID", &ackid);
            if (ackid==priority)

```

```

        if (!tec)
        {
            tec--; // basarili iletim tec 1 azaltilir
            op_stat_write(tec_stat, tec);
        }
        op_pk_destroy(pkptr);
    }
}
}

```

“ÖncelikÇöz” Durum Makinesi:

```

/* 1 indeksli alt kuyruktaki paketleri sirala */
op_subq_sort(1);

/* kuyrun en üstündeki paketi öğren */
if (op_subq_pk_access(1, OPC_QPOS_HEAD) != OPC_NIL)
    pksirali = op_subq_pk_access(1, OPC_QPOS_HEAD);

/* Arbitration'i kazanan paketin boyutunu öğren */
op_pk_nfd_get(pksirali, "pk_bit_sayisi", &pk_bit_sayisi);
// CAN mesajın iletim zamanını hesapla
alim_bitis_zamani = op_sim_time() + pk_bit_sayisi * tbit;
/* alimin tamamlandığını gösteren interuptu kur */
op_intrpt_schedule_self(alim_bitis_zamani - 11 * tbit, Intrpt_Comm);

```

“Alım” Durum Makinesi:

```

j = OPC_FALSE; // alimin tamamlandığını gösterir
alınan_pk_sayisi = 0;
pkptr = op_subq_pk_remove(1, OPC_QPOS_HEAD);
/* Arbitration işlemi için kullanılan alt kuyruğu temizle */
op_subq_flush(1);
/* Bu düğüme gelen en yüksek öncelikli CAN mesaj kendine mi ait? */
op_pk_nfd_get(pkptr, "Priority", &alınan_pk_id);
if (alınan_pk_id == H_ID)
{
    /* Alınan mesajın uçtan uca gecikmesini hesapla */
    op_pk_nfd_get(pkptr, "creation_time", &creation_time);
    ete = op_sim_time() - creation_time;
    op_stat_write(ete_stat, ete);
    if (ete > dead_time)
    {
        packet_loss++;
        op_stat_write(packet_loss_stat, packet_loss);
    }
    else
    {
        /* Bu düğümü adresleyen CAN mesajların istatistikini tut */
        gercek_al_sayisi++;
        op_stat_write(gercek_al_stat, gercek_al_sayisi);
    }
    /* paketi Sink nesnesine gönder */
    op_pk_send(pkptr, Sink_Out_Strm);
    /* Başarılı alım */
    if (rec > 1 && rec < 127)
        rec--;
    else if (rec > 127)

```

```

    rec=121;
    op_stat_write(rec_stat, rec);
    /* ACK gönder */
    if (!Bus_Busy)
    {
        ack_pkptr = op_pk_create_fmt("can_pk_ack");
        op_pk_nfd_set(ack_pkptr, "ACK_ID", alinan_pk_id);
        op_pk_send(ack_pkptr, Tx_Out_Strm);
        gon_ack_sayi++;
        op_stat_write(gon_ack_stat, gon_ack_sayi);
    }
}
else
    op_pk_destroy(pkptr);

```

“Kaynak’dan” Durum Makinesi:

```

/* Uretecten gelen CAN veri frameleri al */
if (op_intrpt_strm () == Src_In_Strm)
{
    pkptr = op_pk_get (Src_In_Strm);
    if (pkptr != OPC_NIL)
    {
        can_veri_pk_sayisi = (int) floor (op_pk_total_size_get (pkptr) / 100);
        if (can_veri_pk_sayisi < 1)
            can_veri_pk_sayisi = 1;
        for (i=0;i<can_veri_pk_sayisi;i++)
        {
            /* CAN veri paketi olustur */
            pkgonder = op_pk_create_fmt ("can_pk_data_20A");
            /* paketin priority sini yukle */
            op_pk_nfd_set (pkgonder, "Priority", priority);
            /* paketin veri ile birlikte boyutu */
            op_ima_obj_attr_get (my_id, "RTR", &RTR);
            if (RTR)
                DLC=0;
            else
                op_ima_obj_attr_get (my_id,"DLC",&DLC);
            /* paketin veri büyüklüğünü gonder */
            op_pk_nfd_set (pkgonder, "DLC", DLC);
            /* bit dolgu islemi ile birlikte paketin toplam bit uzunlugu */
            op_pk_nfd_set(pkgonder, "pk_bit_sayisi", (44 + (DLC*8)));
            /* paketin olusturulma zamanini kur */
            //op_pk_nfd_set (pkgonder, "creation_time", op_sim_time());
            op_subq_pk_insert (0, pkgonder, OPC_QPOS_TAIL);
        }
    }
}

```

“Gönder” Durum Makinesi:

```

/* Gönderme islemi hata durumlarına göre yapilir */
hata_yonetimi();
/* CAN node Aktif hata durumunda normal çalisir */
if (hata_durum == Bus_Aktif)
{
    /* Bufferdan paketi al */
    pkptr = op_subq_pk_remove (0, OPC_QPOS_HEAD);

```



```

    /* paketin olusturulma zamanini kur */
    op_pk_nfd_set (pkptr, "creation_time", op_sim_time());
    /* paketi bus a gönder */
    op_pk_send (pkptr,Tx_Out_Strm);
    // ack_ok=OPC_FALSE;
}

/* CAN node Pasif hata durumunda normal çalışır,
ancak her iletimden önce 8 bit zamani kadar bekler*/
if (hata_durum == Bus_Pasif)
{
    suanki_zaman = op_sim_time();
    /* 8 bit süresi kadar bekle */
    while ((suanki_zaman + 8 * tbit) > op_sim_time());
    /* Bus i kontrol et, bos ise gonder */
    if (!Bus_Busy)
    {
        /* Bufferdan paketi al */
        pkptr = op_subq_pk_remove (0, OPC_QPOS_HEAD);
        /* paketin olusturulma zamanini kur */
        op_pk_nfd_set (pkptr, "creation_time", op_sim_time());
        /* paketi bus a gönder */
        op_pk_send (pkptr,Tx_Out_Strm);
        // ack_ok=OPC_FALSE;
    }
}
gonderilen_pk_sayisi++;
op_stat_write(gonderilen_pk_stat, gonderilen_pk_sayisi);

```

Fonksiyonlar:

```

void hata_yonetimi ()
{
    FIN (hata_yonetimi ())
    if (tec<0)
        tec = 0;
    if (rec<0)
        rec = 0;
    if (tec>255)
    {
        hata_durum = Bus_Kapali;
        op_intrpt_schedule_self(op_sim_time(), Intrpt_Bus_Off);
    }
    else if ((rec>127) || (tec>127))
        hata_durum = Bus_Pasif;
    else
        hata_durum = Bus_Aktif;
    op_stat_write (hata_mod_stat, hata_durum);
    FOUT
}

```

Durum Makineleri Arası Geçiş Şartları:

```

#define INFINITE_TIME -1.0 // baslama ve bitis zamanlari icin sonsuz degeri
#define EPSILON 1e-10 // yuvarlama hatasi icin
#define Paket_Sonu      2
/* Constant Definitions (paket akis yonleri) */
#define Src_In_Strm      0

```

```

#define Sink_Out_Strm 0
#define Tx_Out_Strm 1
#define Rx_In_Strm 1

/* Self interrupts codes (kendi interrupt kodlarim) */
#define Intrpt_Arbitration 10
#define Intrpt_Comm 20
#define Intrpt_Data 30
#define Intrpt_Pk_Err 40
#define Intrpt_Bus_Off 50

/* Hata Durumlari */
#define Bus_Aktif 0
#define Bus_Pasif 1
#define Bus_Kapali 2

/* Transition Condition Macros */
#define Pk_Kaynakdan (op_intrpt_type () == OPC_INTRPT_STRM) && (op_intrpt_strm () ==
Src_In_Strm)
#define Bus_Mesgul (op_stat_local_read (0) == 1.0)
#define Kanal (op_intrpt_type () == OPC_INTRPT_STAT)
#define Tamponda_Pk_Var (!(op_subq_empty (0)))
#define Pk_Al (op_intrpt_type () == OPC_INTRPT_STRM) && (op_intrpt_strm () ==
Rx_In_Strm)
#define Öncelik (op_intrpt_type () == OPC_INTRPT_SELF) && (op_intrpt_code () ==
Intrpt_Arbitration)
#define Alim_Bitti (op_intrpt_type () == OPC_INTRPT_SELF) && (op_intrpt_code () ==
Intrpt_Comm)
#define Bus_Off (op_intrpt_type () == OPC_INTRPT_SELF) && (op_intrpt_code () ==
Intrpt_Bus_Off)

```

EK-B: KAB MODÜLÜNE AİT PROGRAM KODLARI

KAB, CAN_MAC ve KLAN_MAC proses modellerinden oluşmaktadır. Bu nedenle, bu ekte, her iki proses modelinin C programlama dilinde yazılan kodları verilmektedir.

KAB CAN_MAC Modülü Proses Modeline Ait Program Kodları

“Basla” Durum Makinesi:

```
/* node'un ve nesnenin id'sini öğren */
my_id=op_id_self();
my_node_id=op_topo_parent(my_id);
z=0;
/* alt kuyruk l'i temizle */
op_subq_flush(1);
/* nesnenin özelliklerini al */
op_ima_obj_attr_get(my_node_id, "CAN_Hedef_ID",&H_ID);
op_ima_obj_attr_get(my_node_id, "CAN_Hedef_ID1",&H_ID1);
op_ima_obj_attr_get(my_node_id, "CAN_Hedef_ID2",&H_ID2);
j=OPC_FALSE;
//op_ima_obj_attr_get(my_node_id, "Source.Start Time", &baslama);
//op_ima_obj_attr_get(my_node_id, "Source.Stop Time", &bitis);
/* bitis zamanı ile baslama zamanını karşılaştır */
//if (bitis <= baslama)
//  hata_mesaj ("bitis zamanı baslama zamanından büyük olmalı",OPC_NIL, OPC_NIL);
/* Transmitter veri iletim hızını al */
tx_id = op_topo_assoc(my_id, OPC_TOPO_ASSOC_OUT, OPC_OBJTYPE_BUTX, 0);
comp_id = op_topo_child(tx_id, OPC_OBJTYPE_COMP,0);
tx_child_id = op_topo_child(comp_id, OPC_OBJTYPE_BUTXCH, 0);
op_ima_obj_attr_get(tx_child_id, "data rate", &tx_data_rate);
/* bit iletim zamanını hesapla */
tbit = 1/tx_data_rate;
/* hata sayıcılarını kur */
tec=120;
rec=120;
/* local istatistikleri belirle */
gonderilen_pk_stat = op_stat_reg ("Gonderilen_Paket", OPC_STAT_INDEX_NONE,
OPC_STAT_LOCAL);
alınan_pk_stat = op_stat_reg ("Alınan_Paket", OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);
gercek_al_stat = op_stat_reg ("Gerçek Alınan", OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);
pk_id_stat = op_stat_reg ("id", OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);
gon_ack_stat = op_stat_reg ("Gonderilen_ACK_Sayisi", OPC_STAT_INDEX_NONE,
OPC_STAT_LOCAL);
tec_stat = op_stat_reg ("Hata Sayıcı.Tec", OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);
rec_stat = op_stat_reg ("Hata Sayıcı.Rec", OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);
hata_mod_stat = op_stat_reg ("Hata Modu", OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);
ete_stat = op_stat_reg ("ETE Delay", OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);
```

```

/* wiu istatistikleri */
can_filtering_rate_stat = op_stat_reg ("CAN_filtering_rate", OPC_STAT_INDEX_NONE,
OPC_STAT_LOCAL);
wlan_forwarding_rate_stat = op_stat_reg ("WLAN_forwarding_rate", OPC_STAT_INDEX_NONE,
OPC_STAT_LOCAL);
wlu_wlan_islem_zamani_stat = op_stat_reg ("wlu_wlan_islem_zamani",
OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);
/* istatistikleri yaz */
op_stat_write (tec_stat, tec);
op_stat_write (rec_stat, rec);

```

“rx'den” Durum Makinesi:

```

/* Bus_Receiver'dan gelen paketi al */
if (op_intrpt_strm() == Rx_In_Strm)
{
    pkptr = op_pk_get (Rx_In_Strm);
    if (pkptr != OPC_NIL)
    {
        op_pk_format (pkptr, pk_format);
        if (strcmp(pk_format, "can_pk_data_20A")==0)
        {
            /* paketin priority'sini al */
            op_pk_nfd_get (pkptr, "Priority", &pk_priority);
            /* pakete bufferda siralamak icin deger ata */
            op_pk_priority_set (pkptr, pk_priority);
            /* paketin filtering rate (süzme hizi) islemi */
            op_pk_nfd_set (pkptr, "suzme_zamani", op_sim_time() );
            /* wiu CAN_WLAN_processing time */
            op_pk_nfd_set (pkptr, "wlu_can_wlan_processing_time", op_sim_time() );
            /* paketi buffera gönder */
            op_subq_pk_insert (1, pkptr, OPC_QPOS_TAIL);
            /* arbitration islemi icin altkuyruğa atilan paketlerin priority istatistigini tut */
            op_stat_write(pk_id_stat, pk_priority);
            /* Arbitration süresi sonunda interrupt kur */
            if (j==OPC_FALSE )
            {
                //paketin alinma zamani
                op_intrpt_schedule_self(op_sim_time() + 11*tbit, Intrpt_Arbitration);
                j=OPC_TRUE;
            }
            alinan_pk_sayisi++;
            op_stat_write(alinan_pk_stat, alinan_pk_sayisi);
        }
        if (strcmp(pk_format, "can_pk_error")==0)
        {
            /* hata tipini al (aktif hata / pasif hata bayragi) */
            op_pk_nfd_get(pkptr, "Hata_Turu", &hata_turu);
            op_pk_nfd_get(pkptr, "ID", &hata_id);
            if (hata_id == priority) // hatali paketi gonderen düğüm ise
            {
                tec=tec+8;
                op_stat_write(tec_stat, tec);
            }
            rec++; // hatali paketi gonderen düğüm degilse
            op_stat_write(rec_stat, rec);
            op_pk_destroy(pkptr);
        }
    }
}

```

```

if (strcmp(pk_format, "can_pk_ack")==0)
{
    /* Alinan ack bu düğüme ait ise */
    op_pk_nfd_get(pkptr, "ACK_ID", &ackid);
    if (ackid==priority)
        if (!tec)
        {
            tec--;          // basarili iletim tec 1 azaltilir
            op_stat_write(tec_stat, tec);
        }
    op_pk_destroy(pkptr);
}
}
else
    op_pk_destroy(pkptr);
}

```

“ÖncelikÇöz” Durum Makinesi:

```

/* 1 indeksli alt kuyruktaki paketleri sirala */
op_subq_sort(1);
/* kuyrun en üstündeki paketi öğren */
if (op_subq_pk_access(1, OPC_QPOS_HEAD) != OPC_NIL)
    pksirali=op_subq_pk_access(1, OPC_QPOS_HEAD);
/* Arbitration'i kazanan paketin boyutunu öğren */
op_pk_nfd_get(pksirali, "pk_bit_sayisi", &pk_bit_sayisi);
// CAN mesajin iletim zamanini hesapla
alim_bitis_zamani=op_sim_time() + pk_bit_sayisi * tbit;
/* alimin tamamlandigini gosteren interuptu kur */
op_intrpt_schedule_self (alim_bitis_zamani-11*tbit, Intrpt_Comm);

```

“KLAN'a” Durum Makinesi:

```

j=OPC_FALSE; // alimin tamamlandigini gösterir
alinan_pk_sayisi=0;
pkptr=op_subq_pk_remove(1, OPC_QPOS_HEAD);
/* Arbitration islemi icin kullanılan alt kuyruğu temizle */
op_subq_flush(1);
/* Bu düğüme gelen en yüksek öncelikli CAN mesaj kendine mi ait? */
op_pk_nfd_get(pkptr, "Priority", &alinan_pk_id);
if (alinan_pk_id == H_ID || alinan_pk_id == H_ID1 || alinan_pk_id == H_ID2)
{
    /* Bu düğüümü adresleyen CAN mesajlarin istatistigini tut */
    gercek_al_sayisi++;
    op_stat_write(gercek_al_stat, gercek_al_sayisi);
    /* Alinan mesajin uçtan uca gecikmesini hesapla */
    op_pk_nfd_get(pkptr, "creation_time", &creation_time);
    ete=op_sim_time() - creation_time;
    op_stat_write(ete_stat, ete);
    /* CAN filtering rate (süzme hizi) hesapla */
    // CAN paketin pkbitsayisi * tbit - 11 tbit karsilik düşen bir gecikme elde edilir
    op_pk_nfd_get(pkptr, "suzme_zamani", &pk_filtering_time);
    can_filtering_rate = op_sim_time() - pk_filtering_time;
    op_stat_write (can_filtering_rate_stat, can_filtering_rate);
    /* CAN forwarding rate (gönderim hizi) hesapla */
    op_pk_nfd_set(pkptr, "bridge_forwarding_time", op_sim_time());
    /* paketi wiu_wlan_mac nesnesine gönder */
    op_pk_send(pkptr, wiu_can_out_strm);
}

```

```

/* Basarili alim */
if (rec>1 && rec<127)
    rec--;
else if (rec>127)
    rec=121;
op_stat_write(rec_stat, rec);
/* ACK gönder */
if (!Bus_Busy)
{
    ack_pkptr = op_pk_create_fmt("can_pk_ack");
    op_pk_nfd_set(ack_pkptr, "ACK_ID", alinan_pk_id);
    op_pk_send(ack_pkptr, Tx_Out_Strm);
    gon_ack_sayi++;
    op_stat_write(gon_ack_stat, gon_ack_sayi);
}
}
else
    op_pk_destroy(pkptr);

```

“KLAN’dan” Durum Makinesi:

```

/* Uretecten gelen CAN veri frameleri al */
if (op_intrpt_strm () == wiu_can_in_strm)
{
    pkptr = op_pk_get (wiu_can_in_strm);
    if (pkptr != OPC_NIL)
    {
        op_subq_pk_insert (0, pkptr, OPC_QPOS_TAIL);
    }
    else
        op_pk_destroy (pkptr);
}

```

“CAN’e” Durum Makinesi:

```

/* Bufferdan paketi al */
pkptr = op_subq_pk_remove (0, OPC_QPOS_HEAD);
/* WLAN gönderim hizinin hesabi */
op_pk_nfd_get (pkptr, "bridge_forwarding_time", &pk_forwarding_time);
wlan_forwarding_rate = op_sim_time() - pk_forwarding_time;
op_stat_write (wlan_forwarding_rate_stat, wlan_forwarding_rate);
z++;
if (z==10)
{
    wiu_wlan_islem_zamani = op_sim_time() - pk_forwarding_time;
    op_stat_write (wiu_wlan_islem_zamani_stat, wiu_wlan_islem_zamani);
    z=0;
}
/* paketi bus a gönder */
op_pk_send (pkptr, Tx_Out_Strm);
gonderilen_pk_sayisi++;
op_stat_write(gonderilen_pk_stat, gonderilen_pk_sayisi);

```

Durum Makineleri Arası Geçiş Şartları:

```
/* Constant Definitions (paket akis yonleri) */
#define wiu_can_in_strm      1
#define wiu_can_out_strm    1
#define Tx_Out_Strm        0
#define Rx_In_Strm         0

/* Self interrupts codes (kendi interrupt kodlarim) */
#define Intrpt_Arbitration 10
#define Intrpt_Comm       20
#define Intrpt_Data       30
#define Intrpt_Pk_Err     40
#define Intrpt_Bus_Off    50

/* Transition Condition Macros */
#define Pk_KLANdan (op_intrpt_type () == OPC_INTRPT_STRM) &&
                (op_intrpt_strm () == wiu_can_in_strm)
#define Bus_Mesgul (op_stat_local_read (0) == 1.0)
#define Kanal (op_intrpt_type () == OPC_INTRPT_STAT)
#define Fifo3_Pk_Var (!(op_subq_empty (0)))
#define Pk_AI (op_intrpt_type () == OPC_INTRPT_STRM) && (op_intrpt_strm () ==
                Rx_In_Strm)
#define Öncelik (op_intrpt_type () == OPC_INTRPT_SELF) && (op_intrpt_code () ==
                Intrpt_Arbitration)
#define Alim_Bitti (op_intrpt_type () == OPC_INTRPT_SELF) && (op_intrpt_code () ==
                Intrpt_Comm)
#define Bus_Off (op_intrpt_type () == OPC_INTRPT_SELF) && (op_intrpt_code () ==
                Intrpt_Bus_Off)
```

KAB KLAN_MAC Modülü Proses Modeline Ait Program Kodları

“Basla” Durum Makinesi:

```
/* prosesin id sini al */
process_id = op_id_self(); node_id = op_topo_parent(process_id); f=0;
Fifo4_data = OPC_FALSE; // kablosuz ortamdan KLAN paket alim kontrol
/* WLAN kaynak ve hedef nolarini ogren */
op_ima_obj_attr_get(node_id, "WLAN_Kaynak_MAC_No",&kaynak_id);
op_ima_obj_attr_get(node_id, "WLAN_Hedef_MAC_No", &hedef_id);
op_ima_obj_attr_get(node_id, "WLAN_Hedef_MAC_No1", &hedef_id1);
op_ima_obj_attr_get(node_id, "WLAN_Hedef_MAC_No2", &hedef_id2);
op_ima_obj_attr_get(node_id, "CAN_Hedef_ID", &can_id);
op_ima_obj_attr_get(node_id, "CAN_Hedef_ID1", &can_id1);
op_ima_obj_attr_get(node_id, "CAN_Hedef_ID2", &can_id2);
/* Transmitter veri iletim hizini al */
tx_id = op_topo_assoc (process_id, OPC_TOPO_ASSOC_OUT, OPC_OBJTYPE_RATX, 0);
comp_id = op_topo_child (tx_id, OPC_OBJTYPE_COMP,0);
tx_child_id = op_topo_child (comp_id, OPC_OBJTYPE_RATXCH, 0);
op_ima_obj_attr_get (tx_child_id, "data rate", &tx_data_rate);
tbit = 1/tx_data_rate; /* bit iletim zamanini hesapla */
slot_time = 20E-06; sifs_time = 10E-06;plcp_pk_header = 192E-06;
difs_time = sifs_time + 2*slot_time;
cw_min = 31; cw_max = 1023; //DSSS için
backoff_slots = 0.0; defer_ctrl = OPC_FALSE; retry_count = 0; // çarpışma için
backoffneed = OPC_FALSE;
```

```

/* Istatistiklerin kaydedilmesi */
al_pk_can_stat = op_stat_reg("Alinan_Paket_CAN", OPC_STAT_INDEX_NONE,
OPC_STAT_LOCAL);
gon_pk_stat = op_stat_reg("Gonderilen_Paket", OPC_STAT_INDEX_NONE,
OPC_STAT_LOCAL);
pk_islem_time_stat = op_stat_reg("Islem_Zamani", OPC_STAT_INDEX_NONE,
OPC_STAT_LOCAL);
backoff_stat = op_stat_reg ("Backoff_Timer", OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);
al_pk_wlan_stat = op_stat_reg ("Alinan_Paket_WLAN", OPC_STAT_INDEX_NONE,
OPC_STAT_LOCAL);
wlan_filtering_rate_stat = op_stat_reg ("WLAN_Suzme_Hizi", OPC_STAT_INDEX_NONE,
OPC_STAT_LOCAL);
can_forwarding_rate_stat = op_stat_reg ("CAN Gonderim Hizi", OPC_STAT_INDEX_NONE,
OPC_STAT_LOCAL);
wiu_can_islem_zamani_stat = op_stat_reg ("wiu_can_islem_zamani", OPC_STAT_INDEX_NONE,
OPC_STAT_LOCAL);
defer_stat = op_stat_reg ("erteleme", OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);
AlinanACKStat = op_stat_reg ("Alinan ACK Paket", OPC_STAT_INDEX_NONE,
OPC_STAT_LOCAL);
AckSonStat = op_stat_reg ("AckSureBitti Sayisi", OPC_STAT_INDEX_NONE,
OPC_STAT_LOCAL);
ack_send_stat = op_stat_reg("ack_send", OPC_STAT_INDEX_NONE, OPC_STAT_LOCAL);
op_stat_write(backoff_stat, backoff_slots);

```

“CAN’den” Durum Makinesi:

```

/* wiu CAN mac den gelen paketi al */
pkptr = op_pk_get (wlan_in_strm);
if (pkptr != OPC_NIL) {
    /* alinan CAN paketlerin sayisi */
    al_pk_can_sayi++; op_stat_write (al_pk_can_stat, al_pk_can_sayi);
    op_subq_pk_insert (0, pkptr, OPC_QPOS_TAIL); /* CAN paketi alt kuyruğa at */ }
else
    op_pk_destroy (pkptr);

```

“Difs” Durum Makinesi:

```

// difs interrupti gecerli degilse kur
if (op_ev_valid (difs_ev) == OPC_FALSE)
    difs_ev = op_intrpt_schedule_self((op_sim_time() + difs_time), ifs_intrpt);

```

“Ertele” Durum Makinesi:

```

if (Kanal&&Mesgul&&DifsGecerli){
    defer_ctrl = OPC_FALSE; // difs süresi içerisinde ortam mesgul ise erteleme aktif
    op_ev_cancel(difs_ev); // difs kesmesi iptal}

//if (defer_ctrl== OPC_FALSE){
backoffneed= OPC_TRUE; //defer_ctrl = OPC_TRUE;
defer_count++; op_stat_write(defer_stat,defer_count);//}

```

“Backoff” Durum Makinesi:

```

Enter Executives
if (backoff_slots== 0){
    if (retry_count==0)

```



```

        max_backoff=cw_min;
    else
        max_backoff= max_backoff*2 +1;
    if (max_backoff>cw_max)
        max_backoff = cw_max;
backoff_slots = floor (op_dist_uniform(max_backoff+1));}

backoff_start_time=op_sim_time();
/* BackoffBitti kesmesini kur */
backoff_self_ev = op_intrpt_schedule_self (op_sim_time()+backoff_slots*slot_time,
        backoff_end_intrpt);
op_stat_write (backoff_stat, backoff_slots); defer_ctrl=OPC_FALSE;

Exit Executives
if (Kanal || Pk_KLANdan) {
    backoff_slots = (op_sim_time()-backoff_start_time)/slot_time;
    op_ev_cancel(backoff_self_ev); op_stat_write (backoff_stat, backoff_slots);}

if (Pk_CANden){
pkptr = op_pk_get (wlan_in_strm); /* wiu CAN mac den gelen paketi al */
if (pkptr != OPC_NIL){
    /* alinan CAN paketlerin sayisi */
    al_pk_can_sayi++; op_stat_write (al_pk_can_stat, al_pk_can_sayi);

    op_subq_pk_insert (0, pkptr, OPC_QPOS_TAIL); /* CAN paketi alt kuyruğa at */}}

```

“KLAN’a” Durum Makinesi:

```

pkptr = op_subq_pk_remove(0, OPC_QPOS_HEAD); /* CAN paketi alt kuyruktan al */
/*----- CAN paketin özelliklerini al ve istatistikleri hesapla -----*/
op_pk_nfd_get (pkptr, "Priority", &can_pk_priority); /* CAN paketin önceliğini al */
op_pk_nfd_get (pkptr, "creation_time", &can_pk_creation_time); // oluşturulma zamanı
op_pk_nfd_get (pkptr, "pk_bit_sayisi", &can_pk_bit_sayi); /* CAN paketin bit sayısını al */
f++;
if (f==300){
    /* wiu CAN_WLAN işlem zamanı hesabi */
    op_pk_nfd_get (pkptr, "wiu_can_wlan_processing_time", &wiu_can_pk_islem_zamani);
    wiu_can_islem_zamani = op_sim_time() - wiu_can_pk_islem_zamani;
    op_stat_write (wiu_can_islem_zamani_stat, wiu_can_islem_zamani); f=0;}
/* CAN paket forwarding rate (gönderim hızı) */
op_pk_nfd_get (pkptr, "bridge_forwarding_time", &pk_forwarding_rate);
can_forwarding_rate = op_sim_time() - pk_forwarding_rate;
op_stat_write (can_forwarding_rate_stat, can_forwarding_rate);
/*-----CAN to WLAN Encapsulation-----*/
op_pk_destroy(pkptr); /* CAN paketi yok et */
pkptr = op_pk_create_fmt("wiu_wlan_mac_pk");/* WLAN çerçeve oluştur*/
/* CAN paketin priority, bit sayisi, oluşturulma zamanı bilgilerini wlan çerçeve içerisine göm */
op_pk_nfd_set (pkptr,"tanımlayıcı", can_pk_priority);
op_pk_nfd_set (pkptr, "can_bit_sayisi", can_pk_bit_sayi);
op_pk_nfd_set (pkptr, "can_creation_time", can_pk_creation_time);
/*-----*/
/* wiu nesnesinin WLAN adresini kur */
if (can_pk_priority == can_id)
    op_pk_nfd_set(pkptr, "hedef_adr", hedef_id);
else if (can_pk_priority == can_id1)
    op_pk_nfd_set(pkptr, "hedef_adr", hedef_id1);
else //(can_pk_priority == can_id)
    op_pk_nfd_set(pkptr, "hedef_adr", hedef_id2);

```

```

op_pk_send(pkptr, Tx_Strm); /* wiu WLAN çerçeveyi gönder */
gon_pk_sayi++; op_stat_write(gon_pk_stat, gon_pk_sayi);
backoff_slots=0.0; op_stat_write (backoff_stat, backoff_slots);
defer_ctrl=OPC_FALSE; max_backoff = cw_min; backoffneed = OPC_FALSE;

```

“KLAN’dan” Durum Makinesi:

```

pkptr = op_pk_get(Rx_Strm); /* WLAN rx'den gelen çerçeveyi al */
//op_pk_nfd_set(pkptr, "suzme_hizi", op_sim_time()); /* WLAN süzme hızı hesabi için */
if (pkptr != OPC_NIL){
    op_pk_nfd_get (pkptr, "hedef_adr", &pk_hedef_adr); /* wlan çerçevenin wlan adresini al */
    if (pk_hedef_adr == kaynak_id)
    {
        op_pk_format(pkptr, pk_format);
        if (strcmp("wiu_wlan_mac_pk", pk_format) == 0)
        {
            /* alınan wiu wlan çerçevelerin istatistiği */
            al_pk_wlan_sayi++;
            op_stat_write (al_pk_wlan_stat, al_pk_wlan_sayi);
            /* wiu nesnesinin wlan çerçeveleri alma (yayılım) gecikmesi */
            op_pk_nfd_get (pkptr, "creation_time", &olus_zamani);
            pk_islem_time = op_sim_time() - olus_zamani;
            op_stat_write (pk_islem_time_stat, pk_islem_time);
            op_pk_nfd_get (pkptr, "hedef_adr", &rec_pk_dest_adr);
            /*----- CAN - WLAN Decapsulation -----*/
            /* CAN priority, bit sayısı ve oluşturulma zamanı değerini al */
            op_pk_nfd_get (pkptr, "tanımlayici", &can_pk_id);
            op_pk_nfd_get (pkptr, "can_bit_sayisi", &can_bit_sayi);
            op_pk_nfd_get (pkptr, "can_creation_time", &uzak_pk_delay);
            op_pk_destroy (pkptr); /* CAN paketi yok et */
            op_intrpt_schedule_self(op_sim_time()+sifs_time, SIFS_END); // ack gönderim kesmesi
            pkptr = op_pk_create_fmt ("can_pk_data_20A");
            if (pkptr != OPC_NIL){
                /* CAN paketin priority, bit sayısı ve oluşturulma zamanını yükle */
                op_pk_nfd_set (pkptr, "Priority", can_pk_id);
                op_pk_nfd_set (pkptr, "pk_bit_sayisi", can_bit_sayi);
                op_pk_nfd_set (pkptr, "creation_time", uzak_pk_delay);
                /* wiu WLAN forwarding rate (gönderim hızı) kur */
                op_pk_nfd_set (pkptr, "bridge_forwarding_time", op_sim_time());
                //op_pk_nfd_set (pkptr, "bridge_forwarding_time", wlan_pk_filtering_time);
                /* CAN paketi alt kuyruğa at */
                op_subq_pk_insert (1, pkptr, OPC_QPOS_TAIL);
                Fifo4_data = OPC_TRUE; // CAN'e geçiş şartı }
            }
        }
        else if (strcmp("wiu_wlan_ack_pk", pk_format) == 0) {
            op_pk_destroy(pkptr);
            if (op_ev_valid (ackEndEvh) && op_ev_pending(ackEndEvh))
                op_ev_cancel (ackEndEvh);

            AlinanAck++;
            op_stat_write (AlinanACKStat, AlinanAck);
        }
        else
            op_pk_destroy(pkptr);
    }
}
else
    op_pk_destroy (pkptr);

```

“CAN’e” Durum Makinesi:

```
Fifo4_data = OPC_FALSE;
pkptr = op_subq_pk_remove(1, OPC_QPOS_HEAD); /* CAN paketi Fifo4 'den al */
op_pk_send (pkptr, wlan_out_strm); /* CAN paketi wiu CAN MAC e gönder */
```

“AckGönder” Durum Makinesi:

```
pkptr=op_pk_create_fmt("wiu_wlan_ack_pk");
op_pk_nfd_set (pkptr, "hedef_adr",rec_pk_dest_adr);
op_pk_send(pkptr,Tx_Strm); ack_send++; op_stat_write(ack_send_stat, ack_send);
```

“AckSon” Durum Makinesi:

```
AckSon++; op_stat_write(AckSonStat, AckSon);
backoffneed=OPC_FALSE; backoff_slots=0.0;retrycount++;
```

Durum Makineleri Arası Geçiş Şartları:

```
/* Akis Tanımlamaları */
#define wlan_in_strm 1
#define wlan_out_strm 1
#define Rx_Strm 0
#define Tx_Strm 0

/*Self Interrupt */
#define ifs_intrpt 10
#define backoff_slot_intrpt 20
#define ACK_END 30
#define SIFS_END 40
#define PRECISION_RECOVERY 0.000000000001
#define ACK_TIME 20E-06
/* Gecis Sartlari (Transation Conditions) */
#define Pk_KLANDan (op_intrpt_type()==OPC_INTRPT_STRM)&&(op_intrpt_strm()== Rx_Strm)
#define Pk_CANDen (op_intrpt_type () == OPC_INTRPT_STRM) && (op_intrpt_strm () ==
wlan_in_strm)
#define Mesgul (op_stat_local_read (0) == 1.0)
#define Fifo2_Pk_Var (!(op_subq_empty (0)))
#define Kanal (op_intrpt_type () == OPC_INTRPT_STAT)
#define Difs (op_intrpt_type () == OPC_INTRPT_SELF) && (op_intrpt_code () == ifs_intrpt)
#define BackoffBitti (op_intrpt_type() == OPC_INTRPT_SELF) && (op_intrpt_code () ==
backoff_end_intrpt)
#define BackoffGerekli (backoffneed == OPC_TRUE)
#define DifsGecerli (op_ev_valid (difs_ev)== OPC_TRUE)
#define Fifo4_Pk_Var (Fifo4_data == OPC_TRUE)
#define AckSureBitti (op_intrpt_type() == OPC_INTRPT_SELF) && (op_intrpt_code
() == ACK_END)
#define AckGonder (op_intrpt_type() == OPC_INTRPT_SELF) && (op_intrpt_code
() == SIFS_END)
```

KİŞİSEL YAYINLAR VE ESERLER

A. Uluslararası hakemli dergilerde yayımlanan makaleler

i. SCI tarafından taranan

1. Çeken, C., Ertürk, İ., Bayılmış, C., “A new MAC protocol designed on TDMA/FDD for QoS support in WATM networks”, *Computer Standards & Interfaces*, Vol. 28, No. 4, 451–466, April 2006.
2. Bayılmış, C., Ertürk, İ., Çeken, C., “Wireless Interworking Independent CAN Segments”, *Lecture Notes in Computer Science*, LNCS 3280, 299–310, 2004.
3. Çeken, C., Ertürk, İ., Bayılmış, C., “A New MAC Protocol Design for WATM Networks”, *Lecture Notes in Computer Science*, LNCS 3261, 564–574, 2004.

ii. Engineering index tarafından taranan

1. Bayılmış, C., Ertürk, İ., Çeken, C., “A New Solution for Extending CAN Segments Using IEEE 802.11 WLAN”, *Journal of The Faculty of Engineering and Architecture of Gazi University*, Vol. 20., No. 2, 197–204, 2005.

B. Uluslararası bilimsel toplantılarda sunulan ve bildiri kitabında basılan bildiriler

1. Bayılmış, C., Ertürk, İ., Çeken, C., Özçelik, İ., Karaca, A., “Implementation of a CAN/IEEE 802.11b WLAN/CAN Internetworking System Using A New Designed WIU”, 11th International CAN Conference, September, 2006.
2. Bayılmış, C., Özçelik, İ., Ertürk, İ., Çeken, C., Ekiz, H., “Performance Aspects of a Hybrid PROFIBUS Network using OPNET”, *Proceedings of the OPNETWORK 2006*, Washington D.C., USA, 2006.
3. Bayılmış, C., Ertürk, İ., Çeken, C., Özçelik, İ., “Modelling Controller Area Networks Using Discrete Event Simulation Technique”, *Complex Computing Networks*, Springer in Physics Series, Vol. 104, 353–358, Jan. 2006.
4. Bandırmalı, N., Çeken, C., Bayılmış, C., Ertürk, İ., “Effects of Transferring Data Traffic on Video Traffic in Wireless ATM Channels”, *International Conference on Electrical and Electronics Engineering, ELECO'2005*, 402–406, Bursa, Türkiye, 2005.

5. Bandırmalı, N., Çeken, C., Bayılmış, C., Ertürk, İ., “Kablosuz ATM MAC Protokollerinden PRMA/DA ve MAC-GB’nin Karşılaştırmalı İncelemesi”, 4. Uluslararası İleri Teknolojiler Sempozyumu, 372–375, Konya, Türkiye, 2005.
6. Bayılmış, C., Ertürk, İ., Çeken, C., “Extending CAN Segments With IEEE 802.11 WLAN”, The 3rd ACS/IEEE International Conference on Computer Systems and Applications, AICCSA–05, Egypt, January 2005.
7. Çeken, C., Ertürk, İ., Bayılmış, C., “A New MAC Protocol Design for Multimedia Applications in WATM Networks”, Proceedings of the OPNETWORK 2004, Washington D.C., USA, 2004.
8. Bayılmış, C., Ertürk, İ., Çeken, C., “Bluetooth Kablosuz Haberleşme Tekniği Kullanarak Veri Transferi Uygulaması”, 3rd International Advanced Technologies Symposium, 305–313, Ankara, Türkiye, 2003.
9. Bayılmış, C., Ertürk, İ., Çeken, C., “Kablosuz Yerel Alan Ağı Uygulaması”, 12th International Turkish Symposium on Artificial Intelligence and Neural Networks, 112–114, Çanakkale, Türkiye, 2003.
10. Özcerit, A., T., Bayılmış, C., Çakıroğlu, M., Çetin, O., “Mikrodenetleyici Tabanlı Eğitim Seti Geliştirilmesi ve Uygulanması”, EGİTEK 2003 Sempozyumu, 676–680, Kıbrıs, 2003.

C. Ulusal hakemli dergilerde yayımlanan makaleler

1. Bandırmalı, N., Çeken, C., Bayılmış, C., Ertürk, İ., “Video Trafikinin Kablosuz ATM Ortam Erişim Kontrol Mekanizmalarına Etkisinin Karşılaştırmalı İncelemesi”, Politeknik Dergisi, Gazi Üniversitesi, 9 (3), 175–180, 2006.
2. Bayılmış, C., Ertürk, İ., Çeken, C., “Kablosuz Bilgisayar Ağlarının Karşılaştırmalı İncelemesi”, Politeknik Dergisi, Gazi Üniversitesi, 7 (3), 201–210 2004.
3. Bayılmış, C., Ertürk, İ., Çeken, C., “A Comparative Performance Evaluation Study of IEEE 802.3 Wired and IEEE 802.11 Wireless LANs for Multimedia Data Traffic”, Journal of Naval Science and Engineering, 2, 1–12, July 2004.
4. Çeken, C., Ertürk, İ., Bayılmış, C., “Kablosuz ATM İle Gerçek-Zamanlı Olmayan Data Transferi”, Süleyman Demirel Üniversitesi, Fen Bilimleri Enstitüsü Dergisi 8 (1), 102–108, 2004.

5. Çeken, C., Ertürk, İ., Bayılmış, C., “Kablosuz ATM Teknolojisi ile Çoklu Ortam Trafiklerinin Transferi”, Otomasyon Dergisi, 8, 122–127, Ağustos 2003.
6. Bayılmış, C., Ertürk, İ., Çeken, C., “Endüstriyel Kablosuz Haberleşme Teknolojileri: Bluetooth ve ZigBee”, Otomasyon Dergisi, 8, 38–40, Ağustos 2003.
7. Bayılmış, C., Ertürk, İ., Çeken, C., “A Wireless Local Area Network Application For Non-Real Time Data Transfer”, International Journal of Computational Intelligence, Vol.1, No .1, 131–133, July 2003.

D. Ulusal bilimsel toplantılarda sunulan ve bildiri kitaplarında basılan bildiriler

1. Bayılmış, C., Ertürk, İ., Çeken, C., Bandırmalı, N., “DSR ve AODV MANET Yönlendirme Protokollerinin Başarım Değerlendirmesi”, Elektrik, Elektronik, Bilgisayar Mühendisliği 11. Ulusal Kongresi, EMO, 280–283, İstanbul, Türkiye, 22–25 Eylül, 2005.
2. Bandırmalı, N., Çeken, C., Bayılmış, C., Ertürk, İ., Kablosuz Erişim Yöntemlerinin Karşılaştırılmalı İncelemesi”, Elektrik, Elektronik, Bilgisayar Mühendisliği 11. Ulusal Kongresi, EMO, 95–98, İstanbul, Türkiye, 22–25 Eylül, 2005.
3. Çeken, C., Ertürk, İ., Bayılmış, C., “Kablosuz ATM Kullanarak Veri ve Ses Transferi Uygulaması”, ELECO 2002 Sempozyumu, 273–277, Bursa, Türkiye, 18–22 Aralık 2002.

E. Diğer yayınlar

1. Özcerit, A., T., Çakıroğlu, M., Bayılmış, C., “C ile Mikrodenetleyici Uygulamaları”, Papatya Yayıncılık, 2005.
2. Çeken, C., Ertürk, İ., Bayılmış, C., “Wireless Networks for Real-Time Multimedia Communications”, Broadband Wireless and WiMAX, published by International Engineering Consortium (IEC), Comprehensive Report, ISBN: 978-1-931695-30-x, 2004.

F. Görev aldığı projeler

1. “Kablosuz Yerel Alan Ağları Kullanarak CAN Segmentlerin Genişletilmesi İçin Arabirim Tasarımı”, TÜBİTAK, EEEAG-105E059, 2005.
2. “Gerçek Zamanlı Çoklu Ortam Uygulamalarının Kablosuz Transferi”, KOÜ, Bilimsel Araştırmalar Birimi, 104, 2003.

ÖZGEÇMİŞ

1979 yılında Edirne'nin Keşan ilçesinde doğdu. İlk, orta ve lise öğrenimini Keşan'da tamamladı. 1997 yılında girdiği Sakarya Üniversitesi Teknik Eğitim Fakültesi Elektronik ve Bilgisayar Eğitimi Bölümü Elektronik Öğretmenliği Programından 2001 yılında Elektronik Teknik Öğretmeni olarak mezun oldu. 2001-2003 yılları arasında Sakarya Üniversitesi Fen Bilimleri Enstitüsü, Elektronik ve Bilgisayar Eğitimi Anabilim Dalı'nda Yüksek Lisans öğrenimini tamamladı. 2003 yılında başladığı Kocaeli Üniversitesi Fen Bilimleri Enstitüsü, Elektrik Eğitimi Anabilim Dalı'ndaki doktora eğitimine halen devam etmektedir. 2002 yılından beri Kocaeli Üniversitesi Teknik Eğitim Fakültesi Elektronik ve Bilgisayar Eğitimi Bölümünde Araştırma Görevlisi olarak çalışmaktadır.