

**RPL'NİN GÜVENİLİR VE DÜŞÜK GÜCE SAHİP HAREKETLİLİK
DESTEĞİ İÇİN YENİDEN TASARLANMASI**

**REDESIGNING RPL FOR RELIABLE AND LOW POWER
MOBILITY SUPPORT**

BARIŞ TOĞRUL

DOÇ. DR. ALİ ZİYA ALKAR

Tez Danışmanı

Hacettepe Üniversitesi
Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin
Elektrik ve Elektronik Mühendisliği Anabilim Dalı için Öngördüğü
YÜKSEK LİSANS TEZİ olarak hazırlanmıştır.

2017

BARIŞ TOĞRUL tarafından hazırlanan “RPL’nin güvenilir ve düşük güce sahip hareketlilik desteği için yeniden tasarlanması” adlı bu çalışma, aşağıdaki jüri tarafından **Elektrik ve Elektronik Mühendisliği ANA BİLİM DALI’nda Yüksek Lisans** olarak kabul edilmiştir.

Prof. Dr. Hüseyin Selçuk GEÇİM
Başkan

Doç. Dr. Ali Ziya ALKAR
Danışman

Doç. Dr. Cenk TOKER
Üye

Doç. Dr. Umut SEZEN
Üye

Yrd. Doç. Dr. Dinçer GÖKCEN
Üye

Bu tez Hacettepe Üniversitesi Fen Bilimleri Enstitüsü tarafından **YÜKSEK LİSANS TEZİ** olarak onaylanmıştır.

Prof. Dr. Menemşe GÜMÜŞDERELİOĞLU
Fen Bilimleri Enstitüsü Müdürü

YAYINLAMA VE FİKRİ MÜLKİYET HAKLARI BEYANI

Enstitü tarafından onaylanan lisansüstü tezimin/raporumun tamamını veya herhangi bir kısmını, basılı (kağıt) ve elektronik formatta arşivleme ve aşağıda verilen koşullarla kullanıma açma iznini Hacettepe Üniversitesine verdiğimi bildiririm. Bu izinle Üniversiteye verilen kullanım hakları dışındaki tüm fikri mülkiyet haklarım bende kalacak, tezimin tamamının ya da bir bölümünün gelecekteki çalışmalarda (makale, kitap, lisans ve patent vb.) kullanım hakları bana ait olacaktır.

Tezin kendi orijinal çalışmam olduğunu, başkalarının haklarını ihlal etmediğimi ve tezimin tek yetkili sahibi olduğumu beyan ve taahhüt ederim. Tezimde yer alan telif hakkı bulunan ve sahiplerinden yazılı izin alınarak kullanması zorunlu metinlerin yazılı izin alarak kullandığımı ve istenildiğinde suretlerini Üniversiteye teslim etmeyi taahhüt ederim.

- Tezimin/Raporumun tamamı dünya çapında erişime açılabilir ve bir kısmı veya tamamının fotokopisi alınabilir.

(Bu seçenekle teziniz arama motorlarında indekslenebilecek, daha sonra tezinizin erişim statüsünün değiştirilmesini talep etmeniz ve kütüphane bu talebinizi yerine getirirse bile, tezinin arama motorlarının önbelleklerinde kalmaya devam edebilecektir.)

- Tezimin/Raporumun 01.01.2020 tarihine kadar erişime açılmasını ve fotokopi alınmasını (İç Kapak, Özet, İçindekiler ve Kaynakça hariç) istemiyorum.

(Bu sürenin sonunda uzatma için başvuruda bulunmadığım takdirde, tezimin/raporumun tamamı her yerden erişime açılabilir, kaynak gösterilmek şartıyla bir kısmı ve ya tamamının fotokopisi alınabilir)

- Tezimin/Raporumun tarihine kadar erişime açılmasını istemiyorum, ancak kaynak gösterilmek şartıyla bir kısmı veya tamamının fotokopisinin alınmasını onaylıyorum.

- Serbest Seçenek/Yazarın Seçimi

05. / 10. / 2017..


(İmza)

Öğrencinin Adı Soyadı
Barış TOĞRUL

Sevgili aileme,

ETİK

Hacettepe Üniversitesi Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmada,

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- ve bu tezin herhangi bir bölümünü bu üniversitede veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.


BARIŞ TOĞRUL

ÖZET

RPL'İN GÜVENİLİR VE DÜŞÜK GÜCE SAHİP HAREKETLİLİK DESTEĞİ İÇİN YENİDEN TASARLANMASI

Barış TOĞRUL

Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Danışmanı : Doç. Dr. Ali Ziya ALKAR

Eylül 2017 , 106 sayfa

Kablosuz sensör ağları teknolojilerinin gelişmesiyle birlikte uygulama alanları da hızlı bir şekilde artış göstermiştir. Ancak sensör ağlarının gerçek potansiyeli kullanılan protokollerin standartlaştırılmamasından dolayı ortaya çıkmamaktadır. Bu soruna çözüm üretebilmek için IETF (İnternet Mühendisliği Görev Grubu) tarafından geliştirilen 6LoWPAN (Düşük Güçlü Kişisel Alan Ağları Üzerinde IPv6) uyumlandırma katmanı, kablosuz sensör ağlarının IPv6 altyapısında çalışabilmesini sağlamıştır. Aynı çalışma grubu IPv6 uyumlu bir yönlendirme protokolü olan RPL (Düşük Güçlü ve Kayıplı Ağlar için Yönlendirme Protokolü) protokolünü geliştirmiştir. Tez kapsamında yapılan çalışma RPL protokolünün temel eksiklerinden birisi olan hareketliliği ele almaktadır. Hareketlilik kablosuz sensör ağlarında temel ancak kapsamlı bir problemdir. Ev ve endüstri otomasyonu, hasta takibi gibi yüksek paket gönderim oranına ihtiyaç duyan uygulamalarda hareketlilik, paket gönderim oranını düşürmekte ve uygulama gereksinimlerinin karşılanmasını zorlaştırmaktadır. Tez kapsamında bu sorunun önüne geçebilmek için RPL protokolüne hareketlilik yeteneği kazandırılmıştır. Hareketliliği yönetebilmek için Phoenix adı altında bir hızlı geçiş algoritması tasarlanmıştır.

Tasarımın amacı uygulama katmanından bağımsız şekilde hızlı bir geçiş yaparak paket gönderim oranını yükseltip, enerji tüketimini düşürmek olmuştur.

Tasarlanan algoritma kablosuz sensör ağı işletim sistemlerinden birisi olan Contiki üzerinde gerçekleştirilmiştir. Algoritmanın performansını incelemek için çeşitli senaryolarda karşılaştırmalı benzetim ve saha testleri yapılmıştır. Karşılaştırmalar standart Contiki RPL protokolü ve kontrol mesajı arttırılmış RPL protokolü ile yapılmıştır. Benzetim ve saha testleri, geliştirilen Phoenix algoritmasının geçiş gecikmesini azaltarak paket gönderim oranını ve enerji tasarrufunu büyük bir oranda arttırdığını göstermiştir.

Anahtar Kelimeler: RPL, Hızlı Geçiş Algoritması, Kablosuz Sensör Ağı, 6LowPAN, Paket Gönderim Oranı, Düşük Güç

ABSTRACT

REDESIGNING RPL FOR RELIABLE AND LOW POWER MOBILITY SUPPORT

Barış TOĞRUL

M.S., Department of Electrical and Electronics Engineering

Supervisor : Assoc. Prof. Dr. Ali Ziya ALKAR

September 2017, 106 pages

With the development of wireless sensor networking technologies, its application areas have also increased rapidly. However, this increase has not been able to use its true potential since the used protocols have not been standardized. To address this problem, the 6LoWPAN (IPv6 over Lowpower Wireless Personal Area Network) adaptation layer developed by the IETF (Internet Engineering Task Force) has enabled wireless sensor networks to work in the IPv6 infrastructure. The same workgroup has developed the RPL (IPv6 Routing Protocol for Low Power and Lossy Networks), an IPv6 compatible routing protocol. The study carried out within the scope of the thesis deals with the mobility which is one of the fundamental shortcomings of the RPL protocol.

Mobility is a basic but comprehensive problem in wireless sensor networks. In applications that require high packet transmission rates, such as home and industrial automation or patient tracking, mobility reduces packet transmission and makes it difficult to meet application requirements. In order to avoid this problem, the RPL protocol has been given mobility

ability within the scope of the thesis. To manage the mobility, a fast hand off algorithm under the name of Phoenix has been designed. The goal of the design has been to reduce the energy consumption while increasing the packet transmission rate by making a rapid transition between parent nodes independently from the application layer.

The designed algorithm is implemented on Contiki, one of the wireless sensor network operating systems. In order to examine the performance of the algorithm, comparative simulations and field tests were performed in various scenarios. The comparisons were made by using the standard Contiki RPL protocol and the RPL protocol which uses increased number of control messages. Simulation and field tests have shown that the Phoenix algorithm improves the packet transmission rate and energy conservation by reducing the hand off delay.

Keywords: RPL, Fast Hand off Algorithm, Wireless Sensor Networks, 6LowPAN, Packet Delivery Rate, Low Power

TEŐEKKÖR

BaŐta hiç bir desteęini esirgemeyen tez danıŐmanım Doç. Dr. Ali Ziya ALKAR'a, tüm zorluklara karŐı beraber göęüs gerdięim hayat arkadaŐım Zeynep iędem TOęRUL'a, bizleri okutup bu günlere getiren annelerim Emel TOęRUL ve Ayten YAKIŐIR'a, rahmetli babam Levent TOęRUL ve babam Yılmaz YAKIŐIR'a, abim Burak TOęRUL'a, yengem Zeynep KIRÇUVAL TOęRUL'a ve ablam Buket YAKIŐIR'a, ailem gibi sevdięim tüm dostlarıma ve alıŐma arkadaşlarıma, son olarak iŐverenim ASELSAN A.Ő.'ye teŐekkörü bir borç bilirim.

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	i
ABSTRACT	iii
TEŞEKKÜR	v
İÇİNDEKİLER	vi
ŞEKİLLER LİSTESİ	xi
SİMGELER VE KISALTMALAR	xiv
1 GİRİŞ	1
1.1 Tezin Akışı	2
2 TEKNİK ARAŞTIRMALAR VE GELİŞTİRME ORTAMI	4
2.1 Düşük Güçlü Kablosuz Sensör Ağları	4
2.1.1 Kullanım Alanları	4
2.1.1.1 Çevre Ölçümü	4
2.1.1.2 Sağlık Takibi	5
2.1.1.3 Ev - Endüstri Otomasyonu	6
2.1.1.4 Askeri Uygulamalar	7
2.1.2 Ağın Yapısı	7
2.1.2.1 Trafik Modelleri	9
2.1.2.2 Kablosuz Ağ Oluşturma	11

2.1.2.3	Kablosuz Ağlarda Yönlendirme	12
2.1.3	Kablosuz Sensör Ağları Yönlendirme Protokolleri	13
2.1.3.1	Yönlendirme Protokollerinin Standartlaşması	15
2.1.3.2	RPL	15
2.1.3.3	6LowPAN	15
2.2	Kablosuz Sensör Ağları Protokol Katmanları	16
2.2.1	Fiziksel Katman	16
2.2.2	Radyo Çevrim Katmanı	17
2.2.3	Ortam Erişimi Kontrol Katmanı	18
2.2.4	Uyumlandırma Katmanı	19
2.2.5	Ağ Katmanı	20
2.2.6	Ulaşım Katmanı	20
2.2.7	Uygulama Katmanı	21
2.3	Geliştirme Ortamı	22
2.3.1	Contiki İşletim Sistemi	22
2.3.2	Cooja Ağ Simülatörü	23
2.3.3	Tasarlanan Geliştirme Kartı	24
2.3.3.1	Kablosuz Mikrokontrolcü	25
2.3.3.2	Anten Tasarımı	26
2.3.3.3	Baskı Devre Kartı	27
3	RPL YÖNLENDİRME PROTOKOLÜ	28
3.1	Ağ Topolojisi	29

3.1.1	Yukarı Yönlü Yönlendirme	29
3.1.2	Aşağı Yönlü Yönlendirme	30
3.2	Yönlendirme Kıstasları	30
3.2.1	En Az Hop Sayısı	31
3.2.2	Enerji	31
3.2.3	Beklenen Gönderim Sayısı	31
3.3	RPL Kontrol Mesajları	31
3.3.1	DIS	32
3.3.2	DIO	32
3.3.3	DAO	34
3.3.4	DAO-ACK	35
3.4	Görev Fonksiyonu	35
3.5	DODAG Oluşumu	36
3.6	Trickle Zamanlayıcısı	38
4	KABLOSUZ SENSÖR AĞLARINDA HAREKETLİLİK VE GELİŞTİRİLMİŞ YÖNTEMLER	39
4.1	Kablosuz Sensör Ağlarında Hareketlilik	39
4.1.1	Hareketliliğin Tanımı	39
4.1.2	Hareketlilik Modelleri	40
4.1.3	Birimler Arası Geçiş Mekanizması	41
4.2	Kablosuz Sensör Ağları İçin Geliştirilmiş Geçiş Algoritmaları	42
4.2.1	MoMoRo Algoritması	44

4.2.1.1	Durağan Ortam Testi	45
4.2.1.2	Hareketli Ortam Testi	46
4.2.1.3	MoMoRo Algoritması Performans Sonuçları	47
4.2.2	Smart-Hop Algoritması	47
4.2.2.1	Hareketliliğin Hızı ve Veri Gönderme Periyodu Testi	50
4.2.2.2	Radyo Çevrim Katmanı ve Veri Gönderme Periyodu Testi	51
4.2.2.3	Smart-Hop Algoritması Performans Sonuçları	52
5	GELİŞTİRİLEN PHOENIX GEÇİŞ ALGORİTMASI	53
5.1	İletişim Kontrolü Mekanizması	53
5.1.1	Phoenix DIS Mesajı	54
5.1.2	Phoenix DAO Mesajı	55
5.1.3	RSSI Ölçümü	55
5.1.4	Phoenix Zamanlayıcıları	57
5.1.4.1	Bağlanma İsteği Zamanlayıcısı	57
5.1.4.2	Kontrol Mesajlaşması Zamanlayıcısı	57
5.1.4.3	Cevap Bekleme Zamanlayıcısı	58
5.1.4.4	Hareketli Birim Güncelleme Zamanlayıcısı	59
5.2	Geçiş Kontrolü Mekanizması	59
5.2.1	Ağa Katılma Evresi	60
5.2.2	İletişim Evresi	60
5.2.2.1	Durağanlık Algılayıcısı	63
5.2.3	Geçiş Evresi	65

5.3	Güç Kontrolü Mekanizması	66
6	BENZETİM VE SAHA TESTLERİ	69
6.1	Test Altyapısı	69
6.1.1	Phoenix Algortima Parametreleri	69
6.1.2	Karşılaştırılacak RPL Modelleri	71
6.2	Benzetimler	72
6.2.1	Benzetim Senaryosu	72
6.2.2	Benzetim Sonuçları	74
6.2.2.1	Birim Hızı Karşılaştırmaları	74
6.2.2.2	Mesajlaşma Hızı Karşılaştırmaları	77
6.2.2.3	Çalışma Seçenekleri Karşılaştırmaları	81
6.3	Saha Testleri	83
6.3.1	Saha Testleri Senaryosu	83
6.3.2	Saha Testleri Sonuçları	87
6.3.2.1	Yukarı Yönlü Mesajlaşmada Mesajlaşma Hızı Karşılaştırması	87
6.3.2.2	Aşağı Yönlü Mesajlaşmada Mesajlaşma Hızı Karşılaştırması	89
7	SONUÇLAR	91
7.1	Test Sonuçları	92
7.2	Yapılacak Çalışmalar	92
	KAYNAKLAR	93
	EKLER	99
	ÖZGEÇMİŞ	105

ŞEKİLLER LİSTESİ

ŞEKİLLER

Şekil 2.1	Sıcaklık ve nem değerleri gerçek zamanlı ölçülen bir dağ [16]	5
Şekil 2.2	Hastane içerisinde doktorun takip edilmesi [22]	6
Şekil 2.3	Bir noktadan bir noktaya iletişim	9
Şekil 2.4	Çok noktadan bir noktaya iletişim	10
Şekil 2.5	Bir noktadan çok noktaya iletişim	10
Şekil 2.6	Ethernet ağının yapısı	11
Şekil 2.7	Geçirgen olmayan kablosuz ağ yapısı	11
Şekil 2.8	Asimetrik yapıdaki kablosuz ağ	12
Şekil 2.9	Kablosuz sensör ağları için OSI katmanları [48]	17
Şekil 2.10	802.15.4 fiziksel katmanının çeşitleri [49]	17
Şekil 2.11	802.15.4 fiziksel katmanında kullanılan kanallar [49]	18
Şekil 2.12	Contiki tarafından desteklenen bazı mikrokontrolcüler [51]	22
Şekil 2.13	Cooja üzerinde çalışan veri toplama simülasyonu	23
Şekil 2.14	Cooja üzerinde incelenen kablosuz iletişimler	24
Şekil 3.1	Ağaç yapısı	29
Şekil 3.2	DIS mesajının yapısı [8]	32
Şekil 3.3	DIO mesajının yapısı [8]	33

Şekil 3.4	DAO mesajının yapısı [8]	34
Şekil 3.5	DAO-ACK mesajının yapısı [8]	35
Şekil 3.6	DODAG oluşum süreci. Çizikli oklar DIO mesajlarını gösterirken, tam oklar tercih edilen ebeveyn seçimlerini göstermektedir.	37
Şekil 4.1	Durağan ağ testinde karşılaştırmalar	45
Şekil 4.2	Hareketli ağ testinde karşılaştırmalar	46
Şekil 4.3	Smart-Hop algoritması evreleri [68]	48
Şekil 4.4	Smart-Hop performans testi 1	50
Şekil 4.5	Smart-Hop performans testi 2	51
Şekil 5.1	Phoenix DIS mesajı	54
Şekil 5.2	Phoenix DAO mesajı	56
Şekil 5.3	Phoenix algoritması çalışma şeması	61
Şekil 6.1	Test Senaryosu	73
Şekil 6.2	Yukarı yönlü mesajlaşmada birim hızı karşılaştırmaları	76
Şekil 6.3	Aşağı yönlü mesajlaşmada birim hızı karşılaştırmaları	78
Şekil 6.4	64 Hz kanal kontrol hızı altında mesajlaşma hızı karşılaştırmaları	80
Şekil 6.5	8 Hz kanal kontrol hızı altında mesajlaşma hızı karşılaştırmaları	82
Şekil 6.6	Farklı mesajlaşma hızlarında Phoenix çalışma seçenekleri karşılaştırması	84
Şekil 6.7	Farklı birim hızlarında Phoenix çalışma seçenekleri karşılaştırması	85
Şekil 6.8	Saha Testi Senaryosu	86
Şekil 6.9	Yukarı yönlü mesajlaşmada mesajlaşma hızı karşılaştırmaları	88
Şekil 6.10	Aşağı yönlü mesajlaşmada mesajlaşma hızı karşılaştırmaları	90

EK E.1	CC2538 Çipinin İç Yapısı [73]	99
EK E.2	Kullanılan Antenin Boyutları [74]	100
EK E.3	Kullanılan Antenin Frekans cevabı [75]	100
EK E.4	Tasarlanan Kartın Şema 1.Sayfası	101
EK E.5	Tasarlanan Kartın Şema 2.Sayfası	102
EK E.6	Tasarlanan Kartın Ön Yüzünün 3 Boyutlu Görünümü	103
EK E.7	Tasarlanan Kartın Arka yüzünün 3 Boyutlu Görünümü	103
EK E.8	Tasarlanan Kartın Tamamlanmış Hali	104

SİMGELER VE KISALTMALAR

Kısaltmalar

6LowPAN	IPv6 over Lowpower Wireless Personal Area Network (Düşük Güçlü Kişisel Alan Ağları Üzerinde IPv6)
ADC	Analog to Digital Converter (Analog Dijital Çevirici)
AM	Amplitude Modulation (Genlik Kiplenimi)
AODV	Ad-hoc On-Demand Distance Vector (İstek Üzerine Geçici Mesafe Vektörü)
ASK	Amplitude Shift Keying (Sayısal Genlik Kiplenimi)
CDMA	Code Division Multiple Access (Kod Bölmeli Çoklu Erişim)
CRC	Cyclic Redundancy Check (Döngüsel Artıklık Denetimi)
DAG	Directed Acyclic Graph (Yönlü Düz Ağaçlar)
DAO	DODAG Destination Advertisement Object (DODAG Hedef Duyuru Objesi)
DAO-ACK	DODAG Destination Advertisement Object Acknowledge (DODAG Hedef Duyuru Objesi Onayı)
DIO	DODAG Information Object (DODAG Bilgi Objesi)
DIS	DODAG Information Solicitation (DODAG Bilgi Talebi)
DODAG	Destination Oriented Directed Acyclic Graph (Hedefe Yönelik Yönlü Düz Ağaçlar)
DTSN	Destination Advertisement Trigger Sequence Number (Hedef Duyuru Tetikleme Sıra Numarası)
ETX	Expected Transmission Count (Beklenen Gönderim Sayısı)
FM	Frequency Modulation (Sıklık Kiplenimi)
FSK	Frequency Shift Keying (Sayısal Sıklık Kiplenimi)
FTP	File Transfer Protocol (Dosya Aktarım Protokolü)
HTTP	Hypertext Transfer Protocol (Bağlantılı Metin Aktarım Protokolü)
I2C	Inter-Integrated Circuit (Entegre Devreler Arası İletişim)
ICMPv6	Internet Control Message Protocol Version 6 (İnternet Kontrol Mesajlaşma Protokolü Versiyon 6)
IETF	Internet Engineering Task Force (İnternet Mühendisliği Görev Grubu)

IP	Internet Protocol (İnternet Protokolü)
IPSO	IP for Smart Objects (Akıllı Objeler İçin IP)
ISO	International Organization for Standardization (Uluslararası Standartlar Teşkilâtı)
JTAG	Joint Test Action Group (Ortak Test Faaliyet Grubu)
KSA	Kablosuz Sensör Ağları
LEACH	Low-energy Adaptive Clustering Hierarchy (Düşük Enerjili Uyarlanabilir Kümeleme Hiyerarşisi)
MAC	Media Access Control (Ortam Erişim Yönetimi)
MOP	Mode of Operation (Çalışma Şekli)
MQTT	Message Queuing Telemetry Transport (Telemetri Mesajları Sıralama Protokolü)
OF0	Objective Function 0(Görev Fonksiyonu 0)
OLSR	Optimized Link State Routing (Optimize Bağlantı Durumu Yönlendirmesi)
OSI	Open Systems Interconnection (Açık Sistemler Arabağlaşımı)
P-RPL	Phoenix RPL
PSK	Phase Shift Keying (Evre Kaydırma Kiplenimi)
RFC	Request For Comments (Resmi İnternet Protokol Standartları)
ROLL	Routing Over Low-power and Lossy Networks (Düşük Güçlü ve Kayıplı Ağlar Üzerinde Yönlendirme)
RPL	Routing Protocol for Low Power and Lossy Networks (Düşük Güçlü ve Kayıplı Ağlar için Yönlendirme Protokolü)
RSSI	Received Signal Strength Indicator (Alınan İşaretin Şiddet Göstergesi)
SPI	Serial Peripheral Interface (Çevre Birimler Arası Seri Arayüz)
SPIN	Sensor Protocols for Information Negotiation (Bilgi Anlaşması için Algılayıcı Protokolleri)
TCP	Transmission Control Protocol (Geçiş Kontrol protokolü)
UART	Universal Asynchronous Receiver/Transmitter (Evrensel Asenkron Alıcı/Verici)
UDP	User Datagram Protocol (Kullanıcı Veri Birimi Protokolü)
USB	Universal Serial Bus (Evrensel Seri Veriyolu)
WIFI	Wireless Fidelity (Kablosuz Bağlantı)

1. GİRİŞ

Kablosuz sensör ağları çok sayıda birimden oluşan kablosuz iletişim yapılarıdır. Gelişen teknoloji ile birlikte kablosuz sensör ağlarının uygulama alanları çok çeşitlenmiştir. Bunların başında güvenlik veya veri toplama amaçlı ortam izleme, ev ve endüstri otomasyonu, askeri uygulamalar ya da hastanelerde hayati verilerin takibi gibi uygulamalar gelmektedir. Ancak ağı oluşturan birimler çok limitli enerji, işlem gücü, iletim gücü ve bellek kaynaklarına sahiptir. Bu sebeple genellikle pil ile çalışan ve fiziksel olarak çok geniş alanlara yayıldığından, pil değişiminin mümkün olmadığı birimlerin ömürleri pil ile birlikte sona ermektedir. Kablosuz sensör ağının ömrünü belirleyen en önemli parametrelerden bir tanesi mesajların ne kadar verimli gönderildiğidir. Bu verimliliği sağlamak için gönderilen mesajın en az sayıda ara birim ile iletilmesi ve kullanılan yolun en kaliteli yol olması gerekmektedir. Kablosuz sensör ağlarında mesaj iletiminde kullanılacak yolun belirlenmesi için birçok yönlendirme protokolü geliştirilmiştir. Bunlardan bazıları AODV [1], OLSR [2], SPIN [3], LEACH [4] gibi protokollerdir. Ancak bahsi geçen yönlendirme protokollerinin firmalara tescilli olması ve protokollerin belirli uygulamalara göre tasarlanmış olması kablosuz sensör ağlarının tam potansiyelinin kullanılmasını engellemiştir. Standartlaştırılma bu durum altında zorunlu hale geldiğinden *IETF (İnternet Mühendisliği Görev Grubu)* [5] *ROLL (Düşük Güçlü ve Kayıplı Ağlar Üzerinde Yönlendirme)* [6] ve *6LowPAN (Düşük Güçlü Kişisel Alan Ağları Üzerinde IPv6)* [7] adı altında 2 çalışma grubu kurarak kablosuz sensör ağlarının kullanabileceği bir yönlendirme ve IPv6 adaptasyon yapısı oluşturmuştur. IP tabanlı yönlendirme protokolü olan *RPL (Düşük Güçlü ve Kayıplı Ağlar için Yönlendirme Protokolü)* [8] bu çalışmanın sonucu olarak geliştirilmiş olup RFC6550 numarası ile internet standartları arasına girmiştir. RPL protokolü ağ içerisindeki aşağı ve yukarı yönlü yönlendirme tablolarını verimli bir şekilde oluşturup idamesini sağlamaktadır. Ancak RPL protokolü durağan veya çok yavaş değişen ağ yapıları için tasarlanmıştır. Bu yüzden ağ içerisinde hareketli bir birim olursa, RPL değişen ağ yapısına ayak uyduramayıp mesajları yönlendirirken hareketli birimin o anki konumuna değil, eski konumuna yönlendirir. Sonuç olarak ağ kendisini onarana kadar geçen sürede paket kaybı yaşanır. Hareketlilik problemini çözmek ve yönlendirme tablolarının hızlı bir biçimde onarılmasını sağlamak için tez kapsamında *Phoenix* adı altında bir geçiş algoritması geliştirilmiştir.

Bu tez kapsamında geliştirilen algoritma katmanlı yapıya uygun şekilde tasarlanmıştır. Bu sayede farklı üst veya alt katman protokolleri ile çalışabilmekte ve performansı bu katmanların çalışma şekillerinden en az düzeyde etkilenmektedir. Phoenix algoritmasında RPL kontrol mesajlarına standartlara uygun eklentiler yaparak yeni kontrol mesajları oluşturulmuştur. Bu mesajlar sayesinde birimler arasında çift yönlü bağlantı kalitesi kestirimi yapılabilmektedir. Bu bilgiler eşliğinde hareketli birim ebeveyn değiştirmesi gerektiğini hızlı bir biçimde anlayıp geçişi gerçekleştirebilmektedir. Phoenix algoritmasında hızlı geçiş mekanizması yanında bir de kablosuz sensör ağlarının en büyük problemi olan enerjinin tasarruflu kullanılmasını sağlayan mekanizmalar geliştirilmiştir.

Tasarlanan algoritma sensör ağlarında ilk IP uygulama katmanını kullanan Contiki [9] işletim sistemi üzerinde gerçekleştirilmiştir. Kablosuz sensör ağlarının bellek kısıtlamaları göz önüne alınarak geliştirilen yazılım paketi var olan yazılım mimarisine entegre edilmiştir. Contiki işletim sistemi içerisinde bulunan Cooja [10] sensör ağı benzetim ortamı sayesinde geliştirme ve performans testleri yapılmıştır. Benzetim ortamında yapılan testleri gerçek ortamda deneyebilmek için tez kapsamında tasarlanan ve üretilen bir geliştirme kartı kullanılmıştır. Gerçek ortam ve benzetim testleri sonunda algoritmanın ağ performansını ciddi ölçüde iyileştirdiği görülmüştür.

1.1 Tezin Akışı

Bölüm 2’de tezin kapsamının kavranabilmesi için gerekli temel bilgiler anlatılacaktır. Bu temel bilgilere kablosuz sensör ağlarının tanımı ve kullanım alanları ile başlanacaktır. Devamında kablosuz sensör ağlarında kullanılan trafik modelleri ve ağ yapısının nasıl oluşturulduğuna değinilecektir. Tez dahilinde kullanılan standart protokollere kısaca değinilecek, ardından sensör ağlarında kullanılan protokol katmanları anlatılacaktır. Bu bölümde son olarak tezde kullanılan geliştirme ve test ortamları gösterilecektir.

Bölüm 3’te ise hareketlilik desteği eklenecek olan protokolün yapısı anlatılacaktır. Bu yapı içerisinde protokolün hangi ağ yapısını kullandığı, ağ oluşturmak için gerekli olan kontrol mesajları ve mesajların ağ içerisinde iletebilmesi için gerekli yönlendirmelerin hangi kısıtlara göre yapıldığı detaylı bir şekilde incelenecektir.

Bir sonraki bölüm olan Bölüm 4'te tezin çözümünü üretmesi beklenen sorun olan kablosuz sensör ağlarında hareketliliğin tanımı yapılacaktır. Tanımların ve hareketlilik modellerinin devamında literatürde geçen yaygın hareketlilik yönetim algoritmaları ve bu algoritmaların performans kriterleri anlatılacaktır. Tez kapsamında geliştirilen algoritmanın karşılaştırılması için gerekli kriterler bu bölümde incelenecektir.

Takip eden Bölüm 5'te tezin şimdiye kadar değindiği ağ yapıları, sorunları ve kıstasları temel alınarak geliştirilen hareketlilik ve enerji yönetim algoritması ortaya konacaktır. Bu bölümde geliştirilen algoritmanın kullandığı kontrol mesajlaşmaları, ağ içerisinde birimin hareketi, enerji tasarrufunun nasıl sağlandığı ve algoritmanın kullandığı zamanlayıcılar detaylı olarak incelenecektir.

Geliştirilen algoritmanın isterleri ne düzeyde sağladığını test etmek için Bölüm 6'da yer alan benzetim ve saha testleri uygulanacaktır. Karşılaştırmalı testler yapabilmek için gerekli test senaryo ve düzenekleri bu bölümde anlatılacaktır. Algoritmanın farklı senaryolar altında yapılan benzetim ve saha testlerinin sonuçları bölüm içerisinde ortaya konacaktır.

Son olarak sonuç bölümü olan Bölüm 7'de geliştirilen algoritmanın standart protokole ve literatürde var olan algoritmalara karşı başarı oranı gösterilecektir. Bu bölümde ek olarak geliştirilen algoritmanın var olan eksik yönlerine değinilecek ve bunların nasıl giderilebileceğine dair öneriler verilecektir.

2. TEKNİK ARAŞTIRMALAR VE GELİŞTİRME ORTAMI

2.1 Düşük Güçlü Kablosuz Sensör Ağları

KSA (Kablosuz Sensör Ağları) araştırma ve uygulama alanlarında yeni ufuklar açmıştır [11]. Her geçen gün farklı senaryolarda çalışabilecek algoritmalar üretilirken, ilerleyen teknoloji ile birlikte yeni kullanım alanları doğmaktadır. Doğaları gereği karakteristik özellikleri uygulamadan uygulamaya oldukça değişkenlik göstermektedir. Ancak bütün bu değişkenliğin içerisinde ortak olan bazı özellikler vardır. KSA coğrafyada rastgele dağılmış, fazla sayıda küçük sensör modüllerinden oluşur. Üzerlerinde fiziksel değerleri ölçebilen, kontrol edebilen bu bilgi ve komutları kablosuz arayüzü ile gönderip alabilen ekipmanlar barındırırlar. Genellikle batarya ile çalışan, enerjileri çok kısıtlı olan bu modüller birbirleri ile düşük enerji ve düşük bant genişliği kullanarak iletişim kurarlar. Modüllerin işlemci gücü ve hafızaları oldukça kısıtlı olmasına karşın, geniş alanda kapsama alanı oluşturmak için birbirleri ile haberleşip çoklu sıçrama metotları kullanmak zorundadırlar. Bu metotlar kısıtlı kaynaklardan dolayı ağı çok verimli bir şekilde yapılandırmalı ve durumu işlevsel olacak şekilde korumalıdır.

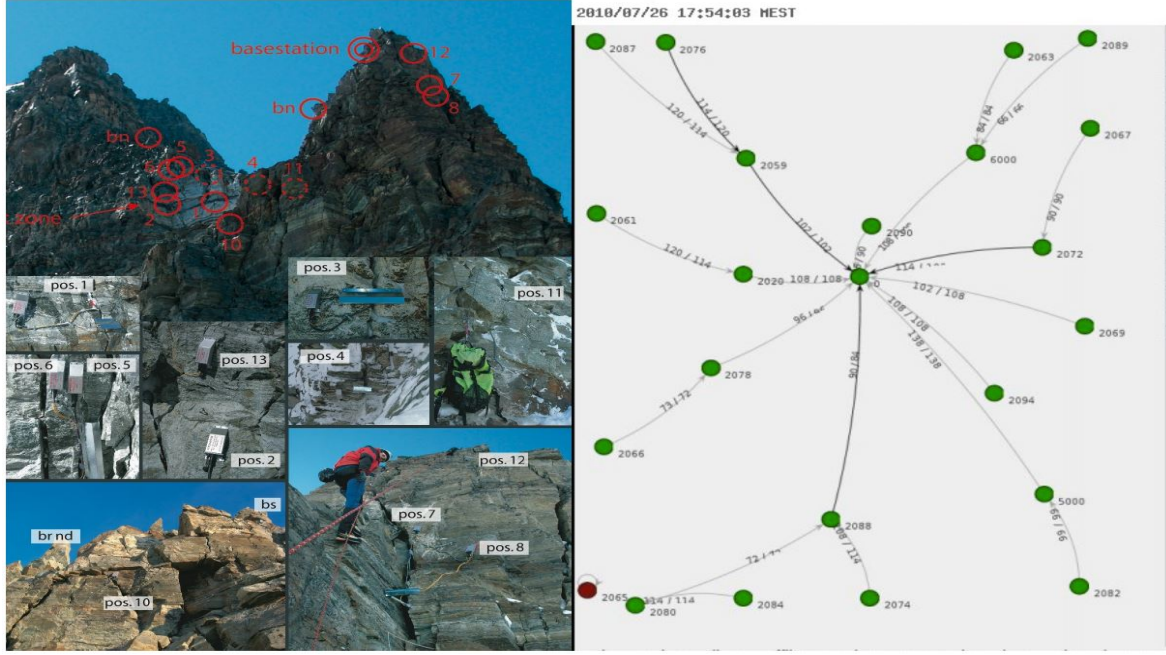
2.1.1 Kullanım Alanları

KSA çok geniş çeşitliliğe sahip olan ölçüm ve kontrol sistemlerinin temelini oluşturur [12]. Kullanım alanları gün geçtikçe artan, günümüzde yoğunlukla çevre ölçümü, ev ve endüstri otomasyonu, güvenlik sistemleri, anlık sağlık takibi ve araç takibinde kullanılmaktadır.

2.1.1.1 Çevre Ölçümü

KSA çevre ölçümünde kilit rol oynar. İnsan gücü ile ölçülemeyecek büyüklükte veya zorlu doğa koşullarına sahip geniş alanlarda uzun zaman boyunca bölgesel değişimleri gözlemlemek için veri toplanabilir. Toplanan veriler tarımın sürdürülebilirliği ve risk yönetimi için kritik olan toprak kalitesi, toz, nem sıcaklık gibi veriler olabileceği gibi, uzun vadeli ve

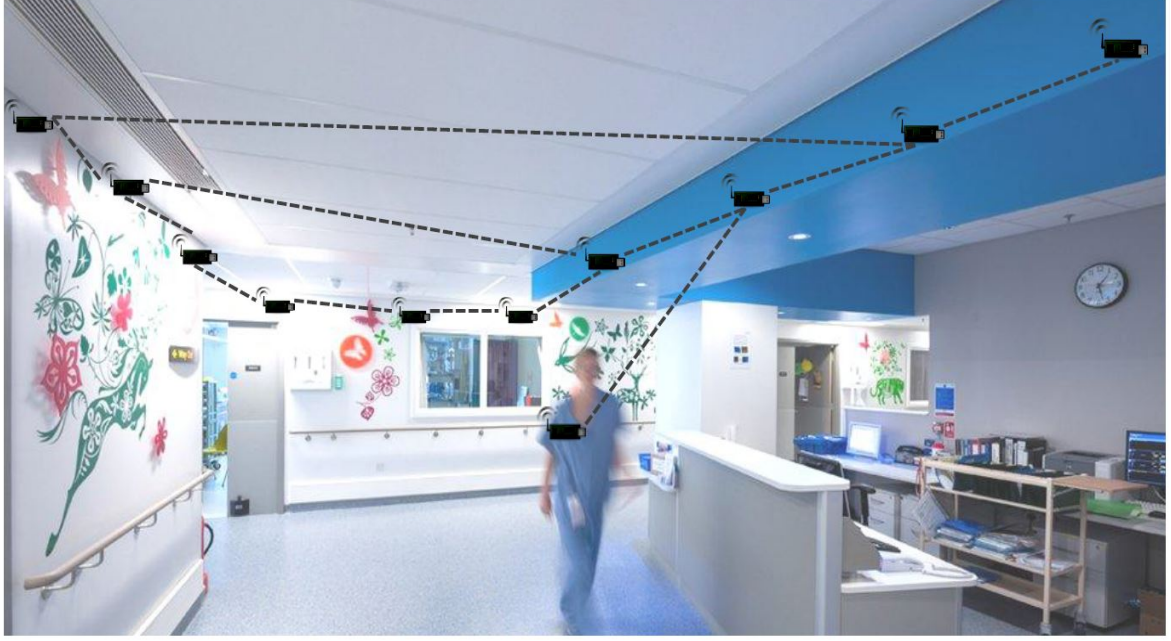
gerçek zamanlı hava kalitesi ölçümü ile bölgesel hava kirlilik oranı gibi uzun vadeli eğilim ölçülebilir [13][14][15]. Şekilde 2.1’de İsviçre Matterhorn kentinde bulunan bir dağ üzerine yerleştirilen 16 adet kablosuz birim gösterilmektedir [16]. Kablosuz birimler dağın farklı bölgelerinden ölçtükları sıcaklık ve nem değerlerini birbirleri aracılığıyla dağın tepesinde bulunan, kablosuz ağı bir başka ağı yapısına bağlayan kök birim adı verilen birime göndermektedirler. Bu sayede dağın durumu hakkında gerçek zamanlı verilere ulaşılabilir.



Şekil 2.1: Sıcaklık ve nem değerleri gerçek zamanlı ölçülen bir dağ [16]

2.1.1.2 Sağlık Takibi

Sağlık hizmetlerinde hastalara yardımcı olabilecek birçok sistem kablosuz sensör ağları ile gerçekleştirilebilir. Bu uygulamaların başında hayati değerlerin anlık takibi, acil yardım butonları, ilaç yönetimi, Şekil 2.2 ile görülebileceği gibi hasta ve doktorun hastane içerisinde takibi gibi kullanımlar gelebilir [17][18][19][20][21]. Uygulamalar sadece hastane içinde kalmayıp evde tedavi gören hastalar için uzaktan 7/24 izlenebilme imkanı sağlar. Böylelikle hastanın hastaneye gitme sıklığı düşürülebilir ve daha zahmetsiz tedavi süreçleri oluşturulabilir.



Şekil 2.2: Hastane içerisinde doktorun takip edilmesi [22]

2.1.1.3 Ev - Endüstri Otomasyonu

KSA günümüzde ev ve endüstri otomasyonu için standart yöntem olarak kabul görmüştür. Eski ev otomasyon sistemleri kablolu ağlar veya güç şebekesi üzerinden veri iletişimi gerçekleştirmektedir. Bu da çok yüksek kurulum ve bakım maliyetlerine, kullanımın sınırlandırılmasına veya hiç uygulanamamasına sebebiyet vermektedir. Kablosuz sensör ağı ile düşük kurulum maliyeti, kullanım kolaylığı, uygulanabilirliğin çok yüksek olması gibi özelliklerle tam potansiyeline ulaşan ev otomasyonu sistemleri günümüzde aydınlatma, ısıtma, enerji kullanımı, güvenlik, zararlı gaz algılama ve eğlence sistemleri gibi hemen hemen her alanda aktif olarak kullanılmaktadır. Bunun sonucunda evlerde ciddi miktarlarda kaynak tasarrufu yapılabilmekte, uzaktan evin durumu ve güvenliği ile ilgili bilgi alınabilmektedir [23][24][25].

Endüstriyel uygulamalarda ise KSA'lar üretim süreçlerinin en önemli bileşenlerindedir. Günümüzde standartlaşmasını neredeyse tamamlayan Endüstri 4.0 ve kablosuz sensör ağlarını envanter takibi, malzeme kalitesinin tüm süreç boyunca izlenmesi ve üretim aşamalarında otomasyon gibi birçok alanda kullanmayı hedeflemiştir [26]. Örneğin kullanılan makinelerin durumlarının anlık görüntülenebilmesi, otomasyon sisteminin bir hata durumunda operatörleri uyarabilmesini ve daha büyük zararları önleyebilmesini sağlar [27][28][29].

2.1.1.4 Askeri Uygulamalar

KSA askeri uygulamalarda çok önemli bir konumdadır. İletişim, uzaktan kontrol, izleme ve alarm sistemleri, hedef takip sistemleri gibi birçok alanda kullanılabilir [30]. Ağ oluşturulan modüller boyut ve ağırlık olarak çok küçük olabildiklerinden savaş alanına havadan atılabilir. Ağın hiç bir operatör yardımı olmaksızın kendi kendini oluşturması ve idame edebilmesi sayesinde hayati risk olmadan gözlenmek istenen alandan ses, görüntü, sismik izler ve ısı izleri gibi bilgiler uzun süre boyunca taktik üstünlük sağlamak için toplanabilir [31][32][33].

2.1.2 Ağın Yapısı

KSA'ların kullanım alanları çok fazla çeşitlilik göstermesine rağmen bu yapılarda kullanılan yazılım ve donanım mimarileri çok benzerdir. Kablosuz sensör ağları uygulamalarında genellikle yüksek miktarda modülün, yüksek yoğunlukla konumlandırılması öngörülmüştür. Ağlarda bir veya birden fazla veri toplayıcı nokta bulunabilir. Bu noktalar ağın geri kalanından veri toplarlarken aynı zamanda ağ başka ağ yapılarına bağlamak için köprü görevi görür. Modüllerde kullanılan donanımlar genellikle bir mikroişlemciden, kilobaytlar mertebesinde geçici ve kalıcı belleklerden, düşük bant genişliği ve düşük çıkış gücü seviyeli radyo almalı-göndermeciden ve son olarak bu bileşenleri en düşük enerji harcayacak şekilde birbirine bağlayan elektronik devrelerden oluşmaktadır [34][35]. Kablosuz sensör ağları genellikle gözetimsiz çalışmak üzere tasarlandıklarından şu sorunları çözmek zorundadırlar:

- **Sınırlı Enerji Kaynağı**

Kablosuz sensör ağlarında kullanılan modüller genellikle bataryalıdır. Özellikle yüksek miktarda kullanıldıklarından ve kolaylıkla ulaşılamayan yerlere bırakılabildiklerinden, batarya değişimi bu modüller için mümkün olmayabilmektedir. Ayrıca bataryaya alternatif olarak çevreden kendi enerjisini üretebilen modüller kullanılabilir. Ancak bunlar da güneş enerjisi, kinetik enerji, radyo dalgaları gibi limitli enerji üretebilen kaynaklardır. Öyle ki kaynak ne olursa olsun enerji kablosuz sensör ağlarında çok değerlidir. Modülün ömrü direkt olarak enerji miktarı ile orantılı olduğundan en küçük enerji bile boşa harcanmamalıdır. Buradaki en büyük zorluk kullanım per-

formansından ödün vermeden enerjiyi koruyabilmektir. Enerji kayıplarının en önemli sebeplerinden birisi veri çakışmasıdır. Birden çok modül aynı anda veri göndermeye çalıştıklarında veriler bozulacak ve aynı veriyi tekrar göndermek zorunda kalacaklardır. Bir başka sebep ise ağda veri gönderen kimse olmasa bile o kanalın dinlenmesi esnasında harcanan enerjidir.

- **Sınırlı Bellek ve İşlem Gücü**

KSA'lar çok yüksek miktarda modül içerebildiğinden modüllerin olabildiğince ucuz olması gerekmektedir. Hem enerji tasarrufu yapılabilmesi için, hem de maliyetin artmaması için modüllerin bellekleri ve işlem güçleri çok kısıtlı tutulmuştur. Kullanılan modüller genellikle 8-32kB geçici hafızaya, 64-512kB mertebesinde kalıcı hafızaya ve 10MIPS den düşük işlem güçlerine sahip işlemcilerle sahiptir. Bundan dolayı kullanılan algoritmaların hem bellek bakımından, hem de işlem gücü bakımından verimli olması gerekmektedir.

- **Otonom Ağ Oluşumu Yapabilme ve Ağ İdame Ettirebilme**

KSA'lar genellikle insan erişimi olmayan yerlere yüksek miktarlarda dağıtılır. Bu yüzden dağıtım öncesinde yüzlerce modülün ayarlanması veya saha üzerinde herhangi bir ayar ve tamirat yapmak mümkün olmayabilir. Bu modüller dağıtım yapıldıktan sonra otonom bir şekilde adresleme yapabilmeli, birbirleri ile bağlantı kurabilmeli ve yönlendirme işlemlerini tamamlayabilmelilerdir. Herhangi bir çevresel faktörden dolayı ağ yapısında değişim olması halinde bu değişime otomatik bir şekilde adapte olabilmelilerdir.

- **Kayıplı Kablosuz İletişim Ortamında Çalışabilme**

Kablosuz iletişim ortam müdahalelerinden kolaylıkla etkilenebildiğinden ve iletişim düşük güçlü radyolar ile sağlanmaya çalışıldığından çoğu zaman iletişim kayıpları yaşanabilmektedir. Modüllerin bu kayıpları tolere edebilir bir şekilde çalışması beklenmektedir. Kullanılan algoritmaların enerji kısıtlarını dikkate alırken veri kayıplarını da minimize edecek şekilde kurgulanması gerekmektedir.

- **Bilgi Güvenliğini Sağlayabilme**

Bilgi güvenliği KSA'ların kritik bilgiler taşıdığı uygulamalarda en önemli parametrelere aittir. Kablosuz yayının doğası gereği gönderilen mesaj dinleyen herkes tarafından duyulabilmektedir. Kısıtlı kaynaklar ve değişen ağ topolojileri, veri bütünlüğü, veri

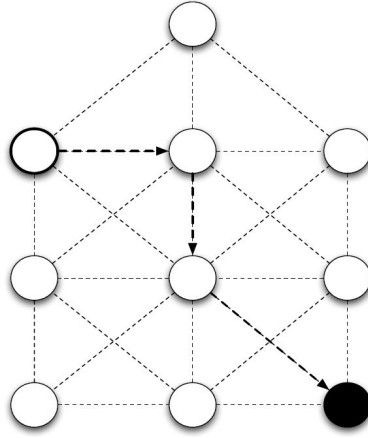
güvenliği ve sayısal imza mekanizmaları gibi bilinen yöntemleri kullanılmaz kılmaktadır. Bu sebeplerden ötürü daha az kaynak harcayan ancak hala tatmin edici düzeyde güvenlik sağlayan yöntemler geliştirilmelidir.

2.1.2.1 Trafik Modelleri

KSA'lardaki ana amaç elde ettikleri ölçüm bilgilerini ağ içerisinde ve dışarısında bir yere göndermek veya başka birimlerden bilgi talep etmektir. Bunu sağlayabilmek ve daha karmaşık görevleri tamamlayabilmek için gerekli trafik modelleri geliştirilmiştir.

- **Bir Noktadan Bir Noktaya İletişim**

Trafik önceden belirlenmiş gönderici ve alıcı arasında gerçekleşir. Şekil 2.3 bu duruma örnektir. Bu model bir birimin başka bir birimden bilgi talep ettiği veya ona bilgi gönderdiği trafikler için kullanılır. Ağın büyüklüğünden ötürü alıcı ve verici birbirlerinin kapsama alanı içinde olmazlar ise bu bilgi talep edilip, gönderilirken bilgi trafiği ara modüller üzerinden gerçekleştirilir [36].

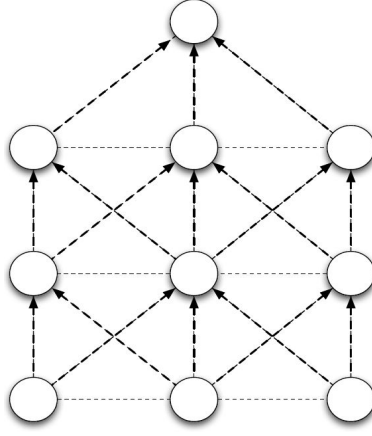


Şekil 2.3: Bir noktadan bir noktaya iletişim

- **Çok Noktadan Bir Noktaya İletişim**

Bu modelde çok sayıda modül ürettikleri bilgileri Şekil 2.4 gibi tek bir noktaya iletir. Veri akışı bu yapıda güvenilir olmak zorunda değildir [37]. Bu tür veri toplama uygulamalarında toplanan veriler birleştirilip kullanılabilirdiğinden anlık bir verinin kaybolması sonucu büyük bir oranda etkilememektedir. Örneğin orman yangını uyarı

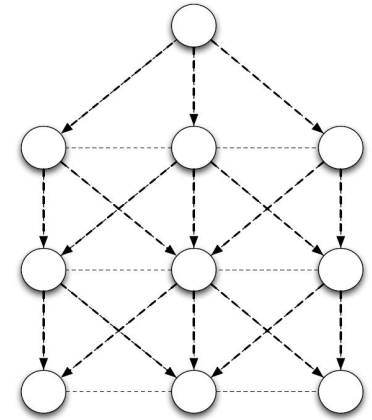
sistemlerinde gönderilen 1 adet sıcaklık bilgisinin kaybolması büyük bir sorun teşkil etmemektedir.



Şekil 2.4: Çok noktadan bir noktaya iletişim

- **Bir Noktadan Çok Noktaya İletişim**

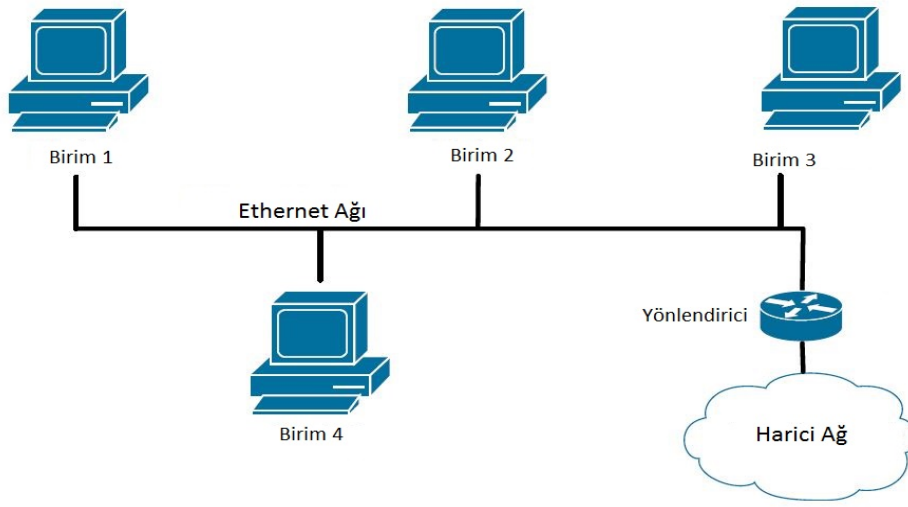
Noktadan çok noktaya veri iletimi uygulamalarında veri bir noktadan (*genellikle ağ yöneticisi*) birçok noktaya Şekil 2.5 ile gösterildiği gibi iletilebilir . Çok noktadan bir noktaya trafik modelinin aksine burada veri akışı güvenli olmalıdır. Örneğin ağ yöneticisinin diğer sensör modüllerine gönderilen örnekleme hızı, gönderim hızı gibi parametreler sistemin düzgün çalışması için bütün birimler tarafından alınmalıdır. Bu tür yapılarda çeşitli güvenli veri akışı algoritmaları kullanılmaktadır [38].



Şekil 2.5: Bir noktadan çok noktaya iletişim

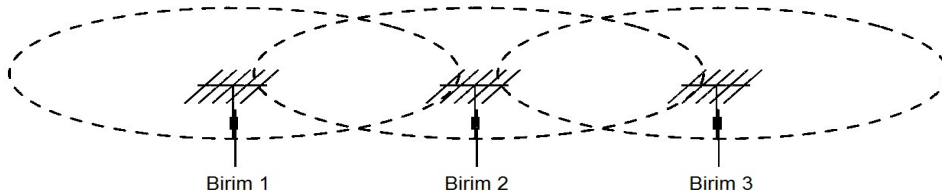
2.1.2.2 Kablosuz Ağ Oluşturma

Kablosuz ağ oluşturma konusunda komşuluk kavramı şu soruya cevap verdiği için önemlidir: Hangi modül bir başka modüle direkt bağlıdır? Kablolu bir ağ olan Ethernet altyapısında bütün birimler bağlandıkları lokal ağ hakkında aynı komşuluk listesine sahiptir [39]. Yani ethernet ağları simetrik ve geçirgendir. Şekil 2.6 ile görülebileceği gibi 1 numaralı birimin 2 numaralı birime gönderdiği mesajı, 3. ve 4. birimler de duyar. Gönderilen bütün mesajlar lokal ağdaki herkes tarafından duyulur ve herkes birine mesaj gönderip birbirinden mesaj alabilir.



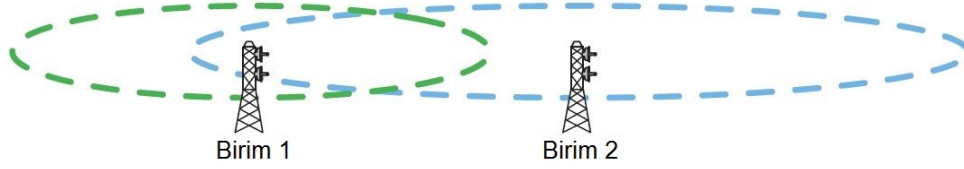
Şekil 2.6: Ethernet ağının yapısı

KSA'larda ise durum farklıdır. Kablosuz Ağlar geçirgen değildir. Şekil 2.7 ile gösterilebileceği gibi 2 numaralı birim 1 ve 3 numaralı birim ile haberleşebiliyorken, 1 ve 3 numaralı birimler kendi aralarında haberleşemezler. Bundan dolayı her birimin komşuluk listesi kendine özgüdür.



Şekil 2.7: Geçirgen olmayan kablosuz ağ yapısı

Kablosuz ađlar aynı zamanda simetrik olmayabilirler. Őekil 2.8 ile gsterilebileceđi gibi 1 numaralı birim 2 numaralı birime mesaj gnderebilirken, 2 numaralı birimin gnderdiđi mesajlar 1 numaralı birime iletilemeyecektir. Bu tr iletiŐime asimetrik iletiŐim denir.



Őekil 2.8: Asimetrik yapıdaki kablosuz ađ

Kablolu ađlarda ađı oluŐturan elemanların durumu genellikle durađandır. Cihazlar ađa ya bađlıdır, ya da bađlı deđildir. Dahası ađa bađlı kaldıkları srede aralarında veri iletiŐimi aısından kaliteli bir bađlantı vardır. Gnderilen paketler ok yksek oranda hedefine iletilir. Kablosuz ađlarda ise durum ok farklıdır. Sinyal kalitesi srekli deđiŐtiđinden birimler sıklıkla ađdan kopup yeniden ađa bađlanabilirler. Ayrıca gnderilen mesajların bir kısmı hedefine iletilirken, bir kısmı kaybolabilir. Bundan dolayı ađ yapısı zaman iinde srekli deđiŐkenlik gsterebilir.

2.1.2.3 Kablosuz Ađlarda Ynlendirme

KSA'larda ynlendirme, verinin kaynađından hedefe ulaŐması iin izlemesi gereken yolu belirler. Bu iŐlemin iki ana basamađı vardır: OluŐturma ve bakım.

OluŐturma aŐamasında verinin birbirini grebilen ve gremeyen birimler arasında nasıl en iyi Őekilde aktarılacađını belirten bir tablo oluŐturulur. Bu tabloda birbirini gremeyen birimler arasında veri iletiŐimi gerekleŐtirilecek ise veri aktarımının hangi ara birimler zerinden yapılacađı bulunmaktadır. En iyi diye bahsedilen yol uygulamadan uygulamaya deđiŐiklik gsterebilir. Uygulamalar iin en nemli parametreler genellikle utan-uca en gecikmesiz yol, en yksek bant geniŐliđi, en dŐk enerji tketimi ya da en dŐk hoplama kısıtlarından biri veya birden fazlası olabilir.

İkinci basamak ise bakımdır. KSA'ların dođası geređi ađ yapısı srekli deđiŐkenlik gsterebilir. Modllerinin enerjilerinin tkenmesi, hareketliliđi, bađlantı kalitesi gibi sebepler nceden hesaplanmış ynlendirme haritasının geersiz olmasına sebep olabilir. Ynlendirme algoritması veri iletiminin srekliliđini sađlayabilmek iin yeni yollar bulmak zorundadır.

Bu hesaplamalar periyodik olabileceği gibi sadece veri iletileceği zamanda da yapılabilmektedir. Bu düşünce yapısı yönlendirme algoritmalarını ikiye ayırır.

- **Reaktif Yönlendirme**

Bu tarz yönlendirmelerde yönlendirme haritası, verinin gönderilmesi gerektiğinde ve aynı zamanda bu gönderim için geçerli bir yol bulunamadığında hesaplanır. Hesaplamaya yeni bir yol oluşturmak için gerekli olan istek paketlerinin ağa dağıtımı ile başlanır. Hedef birimden cevap gelene kadar beklenir. Gelen cevap üzerine yeni yol tayin edilmiş olur ve veri ilgili hedefe yeni yol üzerinden aktarım yapılır. Ancak bu işlem zaman aldığından verinin gönderimini gecikmeye uğrattır. Ağ üzerinde yönlendirmelerin sürekli değişmesi gerekiyor ise yüksek oranda kontrol mesajlaşmaları yapılacak, bu da hem zaman hem de enerji kaybı yaratacaktır. *AODV (İstek Üzerine Geçici Mesafe Vektörü)* [1] bu tarz yönlendirme yapısına uygun bir örnektir.

- **Proaktif Yönlendirme**

Proaktif yönlendirmelerde ise yönlendirme haritası periyodik olarak güncellenerek veri daha gönderilmeden hangi yol üzerinden gönderileceği bilinmektedir. Bu sayede veri, yönlendirme tabanlı bir gecikmeye uğramadan hedefine gönderilebilir. Proaktif yönlendirmelerde yol haritasını güncel tutabilmek için veriden bağımsız sürekli kontrol mesajlaşmaları yapılır. Bu mesajlaşmalarda veri gönderimi yapılmaya bile birimin sürekli enerji harcamasına sebep olmaktadır. *OLSR (Optimize Bağlantı Durumu Yönlendirmesi)* [2] bu tarz yönlendirme çeşitlerinden günümüzde aktif olarak kullanılan birisidir.

2.1.3 Kablosuz Sensör Ağları Yönlendirme Protokolleri

Günümüzde halihazırda birçok yönlendirme protokolü geliştirilmiştir [40]. Her bir protokol daha önce bahsedilen (Bölüm 2.1.2) kısıtlamaların bir ya da birden fazlasına çözüm olmak üzere geliştirilmiştir. Bu protokollerden en basit olanı bilgiyi ağ içerisinde çığ etkisi ile yaymaktır [40]. Bu protokolda her bir birim aldığı mesajı mesaj hedefine ulaşıncaya kadar komşu birimlerine iletir. İletim esnasında mesaj fırtınası yaratmamak için birçok yöntem geliştirilmeye çalışılmışsa da mesajların defalarca gereksiz yere tekrarlanmasından dolayı büyük oranda enerji kaybı yaşanır. Bu yöntem örneğin ormanlarda yangın detektörü olarak çalışan, ömürlerinde sadece birkaç kez veri iletimi yapan sistemlerde kullanılmaktadır.

Bir önceki yöntemdeki gereksiz trafiği önlemek için *SPIN (Bilgi Anlaşması için Algılayıcı Protokolleri)* [3] isimli protokol ortaya çıkmıştır. Bu protokolde veri gönderilmeden önce veri ile ilgili bir bilgi paketi komşulara gönderilir. İlgili veriyi almak isteyen komşular gönderen birime veriyi almak istediklerine dair bir cevap mesajı yollarlar. Bunun üzerine veri göndericisi mesajı herkese göndermek yerine sadece isteyenlere gönderir. Ancak bu sistemde veriyi özetleyen bilgi paketçığının belli bir standartı yoktur ve paketin yapısı uygulamadan uygulamaya değiştiği için kullanım alanı sınırlı kalmıştır. Bir diğer eksikliği ise verinin hedefe ulaşıp ulaşmadığını kontrol eden bir mekanizmanın bulunmamasıdır. Mesaj iletilirken arada mesajı almak istemeyen bir birim var ise mesaj gönderimi sekteye uğrayacaktır [3].

KSA'larda ölçeklenebilirlik ve enerji tasarrufunu artırıp ağ ömrünü uzatmayı hedefleyen bir diğer protokol ise *LEACH (Düşük Enerjili Uyarlanabilir Kümeleme Hiyerarşisi)* [4]'dir. Enerji tasarrufu sağlamak için modülleri gruplar. Gruplanan modüllerin bir grup başkanı vardır. Buradaki amaç gönderilecek verinin kaynaktan direkt hedefe gitmesi yerine ortak bir noktada toplanıp, bir kerede gönderilmesidir. Burada kazanç hem grup içerisindeki modüllerin birbirine geometrik olarak yakın olmasından dolayı veri gönderiminin daha düşük bir enerji ile yapılabilmesiyle, hem de mesajların toplanıp bir kerede gönderilmesiyle ağ içerisindeki trafiğin azaltılması sayesinde sağlanır.

KSA'larda enerji tasarrufu sağlayan bir diğer yöntem ise veri birleştirmedir. Bu protokolde veri iletimini sağlayan ara birimler kendine gelen veriler üzerinde bir takım işlemler yaparak trafiği azaltmayı hedefler. Örneğin birimler kendilerine gelen verilerin ortalamasını alma, en azı veya en çoğunu bulma gibi fonksiyonlar kullanarak, sadece uygulama için önemli olan verileri iletirler. Ancak bu tarz protokollerde de belli bir standart olmadığından kullanım alanları sınırlı kalmıştır.

KSA'larının kullanım yerleri arttıkça, protokollerin uygulamalar ve donanım birimleri arasında uyumluluğu, diğer ağ yapıları ile birlikte çalışabilmeleri, aynı zamanda bunu en az enerji ve en fazla esneklik çerçevesinde gerçekleyebilmeleri bir ihtiyaç haline gelmiştir. Böyle standart bir protokolün bulunmaması ve bu standartlaşmayı yapabilecek bir organizasyonun olmaması KSA'ları standartlaşmaya zorlamıştır.

2.1.3.1 Yönlendirme Protokollerinin Standartlaşması

KSA'ların uygulama potansiyelleri ortaya çıktıkça endüstrinin ilgisi de bu yöne kaymıştır. Ancak daha önceden geliştirilen algoritmaların bir takım firmalara tescilli olması ve bundan dolayı uygulamadan uygulamaya olan uyumluluğun sağlanamaması endüstride büyük bir açık oluşturmuştur. Bu açıklık IPv6'nın yaygınlaşması ve bütün cihazların aynı ağa bağlanabilme düşüncesi ile daha da büyümüştür. Başlarda *IPSO (Akıllı Objeler İçin IP)* [41] adı altında bir grup uygulama mühendisinin çalışması ile, endüstri ihtiyaçlarının IP tabanlı KSA'lar ile nasıl karşılanabileceği araştırılmaya başlanmıştır. İlerleyen zamanlarda bu standartlaştırma işlemine *IETF (İnternet Mühendisliği Görev Grubu)* [5] da katılmıştır. *IETF ROLL (Düşük Güçlü ve Kayıplı Ağlar Üzerinde Yönlendirme)* [6] ve *6LowPAN (Düşük Güçlü Kişisel Alan Ağları Üzerinde IPv6)* [7] adı altında 2 çalışma grubu kurmuş ve standartların ana taslaklarını oluşturmuşlardır.

2.1.3.2 RPL

IETF tarafından oluşturulan ROLL çalışma grubunun amacı KSA'larda karşılaşılan düşük güç çıkışı, düşük bant genişliği, limitli enerji ve işlem kaynakları, kayıplı ağlar gibi kısıtlamalar altında çalışabilen bir yönlendirme protokolü geliştirmektir. Geliştirilen protokol bilgi güvenliğini sağlamalı, aynı zamanda kolay yönetilebilir olmalı ve IPv6 ile uyumlu olmalıdır. ROLL çalışma grubu, endüstri istekleri doğrultusunda ev otomasyonu [42][43], kırsal alan uygulamaları [45] ve endüstriyel uygulamalar [44] için gereksinimleri belirlemiştir. Çalışmalarının sonucu olarak ROLL grubu ortaya bütün bu gereksinimleri karşılamayı hedefleyen *RPL (Düşük Güçlü ve Kayıplı Ağlar için Yönlendirme Protokolü)* isimli yönlendirme protokolünü ortaya çıkarmıştır. Bu yönlendirme protokolü *RFC (Resmi İnternet Protokol Standartları)* tarafından tanınarak "*RFC 6550*" [8] numarası ile standartlara girmiştir. RPL protokolü ile detaylı bilgi sonraki bölümlerde verilecektir.

2.1.3.3 6LowPAN

6LowPAN çalışma grubu ise KSA'ların IPv6 ağ yapısı ile uyumlu çalışabilmesini sağlamak amacı ile IETF tarafından kurulmuştur. Buradaki zorluk veri kaybı yaratmadan KSA'lar gibi

ciddi kısıtlamaları olan bir sistem ile, kaynak bakımından zengin ve gelişmiş cihazların olduğu bir sistemi aynı ağa bağlayabilmektir. Çalışmalar sonucunda IPv6 paketlerini kayıpsız sıkıştırarak, bu paketleri KSA'lar için kullanıma uygun hale getiren ve hiç bir ara basamak kullanmadan KSA'ları IPv6 internet yapısına bağlayabilen bir katman geliştirmişlerdir. RPL gibi 6LowPAN uyumlandırma katmanı da RFC tarafından tanınarak "RFC6282" [46] numarası ile standartlara girmiştir.

2.2 Kablosuz Sensör Ağları Protokol Katmanları

Günümüzde kullandığımız ağ modelleri veri taşımının en basit hali olan veriyi bir ortamdan diğer ortama kopyalama mekanizmasının üzerine kurulmuştur. Akla gelebilecek ilk mekanizma disket ile veri taşımaktır. Gelişen teknoloji ile birlikte hem kullanılabilir ortamların hem de veri taşıma yöntemlerinin sayısı artmıştır. Ancak herhangi bir standartlaşma olmadığı için cihazlar birbiri ile uyum içinde haberleşememekteydiler. Örneğin bir kişinin IBM cihazı varsa, sadece IBM cihazları ile haberleşebilmekteydi. Bu uyumsuzluğun üzerine 1984 yılında ISO (*Uluslararası Standartlar Teşkilâtı*) [47] adında bir grup OSI (*Açık Sistemler Arabağlaşımı*) [48] modelini yaratmıştır. Bu model farklı donanım ve yazılım firmaları arasındaki birlikte çalışabilirliği sağlayan yöntemleri tanımlar. Bu standartlaşmanın sonucu olarak bir firmaya bağımlı kalınmaksızın farklı cihazlar aynı ağda iletişim kullanılabilir hale gelmiştir. Ortaya çıkan rekabet firmaların daha gelişmiş ürünler çıkarmasını da kamçılamıştır.

OSI katmanlı bir yapıdır. Katmanlarına ayrılmış bir model kullanmanın avantajı yapılacak işlemleri parçalara bölmekle başlar. Her parça birbirinden bağımsız olarak geliştirilip test edilebilir. Farklı ihtiyaçlar ortaya çıktığında ise sadece ilgili parça veya parçalar değiştirilerek daha hızlı bir şekilde soruna çözüm bulunabilir. Bu sayede genel uygulama yapısı daha kolay yönetilebilir hale getirilir. OSI katmanları kablosuz ağlar için Şekil 2.9 ile gösterilen bileşenlere ayrılabilir.

2.2.1 Fiziksel Katman

OSI katmanlarından en düşük seviyeli olan fiziksel katmanın 2 ana görevi vardır: Veriyi göndermek ve almak. Bu katman gönderimin hangi ortamda hangi modülasyon ile transfer



Şekil 2.9: Kablosuz sensör ağları için OSI katmanları [48]

edileceğini belirler. Gönderim ortamı elektrik, mikrodalga, ışık ve ses gibi çeşitli seçeneklerden birisi olabilirken, modülasyon çeşitleri de AM, FM, ASK, FSK ve PSK gibi seçeneklere sahip olabilir. KSA'ların kısıtlamaları düşünüldüğünde bu kısıtlamaları sağlamak için bir fiziksel katman geliştirilmiştir. "IEEE 802.15.4" numarası ile 2003 yılında standartlara giren bu katmanın gelişimi halen devam etmektedir. Günümüzde en son versiyonu olan "802.15.4g" versiyonu bulunan standartın fiziksel katman özelliklerinden bazıları Şekil 2.10 ve Şekil 2.11 ile gösterilmiştir.

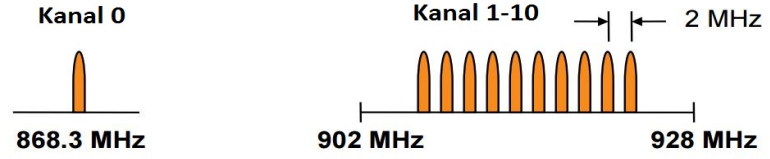
Bant Genişliği (MHz)	Kanal Aralığı	Kanal Sayısı	Yonga Hızı Kyonca/saniye	Modülasyon	Bit Hızı Kb/saniye	Sembol Hızı Ksembol/saniye	Sembol Tipi
868-868,6	-	1	300	BPSK	20	20	İkili
902-928	2 MHz	10	600	BPSK	40	40	İkili
2400-2483,5	5 MHz	16	2000	O-QPSK	250	62,5	16'lı Ortogonal

Şekil 2.10: 802.15.4 fiziksel katmanının çeşitleri [49]

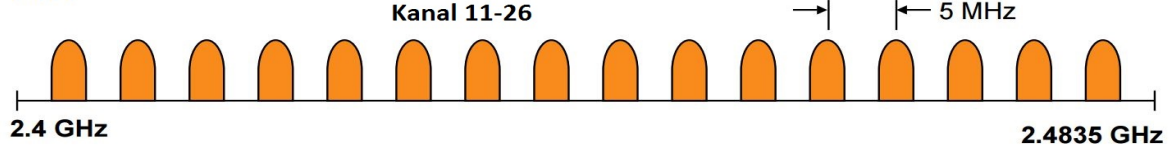
2.2.2 Radyo Çevrim Katmanı

Standart OSI modellemesinde olmayan, ancak enerji kısıtlamaları olan kablosuz sensör ağları için vazgeçilmez olan bir katmandır. Cihazların haberleşmesi sırasında gönderen cihazın verici tarafının, alan cihazın ise alıcı tarafının açık olması gerekmektedir. Ancak cihazın ne zaman alım yapacağı belli değilse veri kaybetmemek için alıcı tarafı hep açık tutmalıdır. Alıcı tarafı hep açık tutulduğunda ise yüksek miktarda enerji tüketilecek, yıllarca çalışması gereken cihazın enerjisi günler içinde bitecektir. Bu sorunun önüne geçmek için fiziksel kat-

868MHz/915MHz PHY



2.4 GHz PHY



Şekil 2.11: 802.15.4 fiziksel katmanında kullanılan kanallar [49]

manın almaç tarafı periyodik olarak kapatılıp açılır. Genellikle kapalı duran almaç devreleri çok kısa süreliğine açılıp ortam dinlemesi yapıldıktan sonra tekrar kapatılır. Eğer herhangi bir taşıyıcı sinyal tespit edilirse bir miktar daha dinlemede kalınır. Ancak gönderilen verinin başı tespit edilemez ise veri tamamıyla alınamayacağından göndericinin bir kez daha gönderim yapması gerekir. Bu yapıda ortam dinleme sıklığı ne kadar artarsa veri kaybı miktarı o kadar azalacak ancak harcanan enerjide bir o kadar artacaktır. Bu parametre uygulamanın gereksinimine göre belirlenmektedir.

2.2.3 Ortam Erişimi Kontrol Katmanı

OSI katmanlarında 3. katman olan ortam erişim kontrolü katmanı iki birim arasındaki veri transferini kontrol eder. 802.15.4 standardı bu katmanın da özelliklerini tanımlamaktadır. Bu katmanın görevleri;

Gönderici tarafında:

- Bir üst katman olan ağ katmanından gelen verileri fiziksel katmanın kullanabileceği şekilde paketler.
- Veri bütünlüğünü sağlamak için CRC (Döngüsel Artıklık Denetimi) [50] hesaplarını yapar ve pakete ekler.
- Veri güvenliği isteniyorsa şifreleme yapılır.

- Hedefin fiziksel adresi belirlenir. Adresler özel bir adrese, grup adresine ya da dinleyen herkese olmak üzere seçilir.
- Zaman senkronizasyonu isteniyorsa periyodik mesaj gönderimi yapılır.

Alıcı tarafında;

- Bir alt katman olan fiziksel katmandan gelen ham bilgi ağ katmanının anlayabileceği biçimde paketlenir.
- Veri bütünlüğünü doğrulamak için CRC hesaplaması yapılır.
- Veri güvenliği kullanılmış ise veri çözülmesi yapılır.
- Adres, link kalitesi ve RSSI (Alınan İşaretin Şiddet Göstergesi) gibi bilgiler hesaplanır.
- Gönderen tarafından mesaj alındı teyidi istendi ise otomatik olarak teyit mesajı gönderir.
- Zaman senkronizasyon bilgisi mevcut ise göndericiye senkron olunur.

2.2.4 Uyumlandırma Katmanı

KSA'ların standartlaştırılmasındaki önemli motivasyonlardan birisi de birimlerin herhangi bir ara basamağa ihtiyaç duymadan var olan IPv6 ağına bağlanabilmesidir. Ancak bu iki ağ yapısı tasarım olarak birbirinden çok farklıdır. KSA'lardaki kısıtlamaların başında gelen düşük bant genişliği ve limitli enerji gibi sorunlar yüksek kapasiteli bilgisayarlar, cep telefonları, internet paylaştırıcıları için geçerli değildir. Onlarda aksine neredeyse sınırsız enerji, Gbit/s seviyelerinde bant genişliği, karşılaştırılamayacak kadar farklı bellek ve işlem gücü fazlalığı vardır. Bu farklardan ötürü bu iki ağı birleştirmek için ikisi arasına 6LowPAN adında bir uyumlandırma katmanı geliştirilmiştir. Uyumlandırma katmanının görevleri:

- IPv6 yapısında en küçük veri paketi 1280 bayttır. Buna karşın 802.15.4 standardında en büyük paket boyutu ise 127 bayttır. 127 baytın başlık ve güvenlik bilgileri de düşünülürse zaman zaman veri olarak geriye sadece 81 bayt kalır. Geriye adresleri ve taşıma

katmanı başlıklarını da çıkardığımızda ham veri için sadece birkaç bayt kalır. Bundan dolayı IPv6 paketleri doğrudan kullanılmak yerine 6LowPAN tanımlamalarına göre sıkıştırılır. 40 bayt olan IPv6 başlığı sıkıştırılma sonrasında mesaj tipine göre 2, 12 veya 20 bayt olacak şekilde küçültülür.

- IPv6 128 bit adrese sahip iken 802.15.4 standardında birimler 64bit veya 16bit adres uzunluğuna sahiptir. Bundan dolayı KSA birimleri kendi IP adreslerini üretebilme özelliğine sahiptirler. Ortam erişim katmanında veri iletişimi yaparken 64bit veya 16bit uzunluğunda adres kullanırken, ağ katmanında ise birimler lokal adreslerine bütün ağda ortak olan ön bilgiler ekleyerek, üst katman uygulamaları için standart IPv6 adreslerini kullanırlar.
- IPv6 cihazları mesaj hoplamasını dağıtıcılar üzerinden yaparken, KSA'larda mesaj hoplamasını destekleyebilmek için bütün birimlerin dağıtıcı gibi çalışması gerekir. Bu yüzden 6LowPAN bir üst katman olan ağ katmanının hoplama yapabilmesi için gerekli mesaj yapılarını tanımlar.

2.2.5 Ağ Katmanı

Standart bir OSI katmanı olan ağ katmanı değişken uzunluklu veri dizilerinin bir ağ biriminden aynı veya farklı bir ağa bağlı başka bir ağ birimine aktarımını sağlar. Bu aktarımı sağlarken mesaja yönlendirme tablosu da eklenir. Eğer mesaj lokal ağın dışına gönderilecek ise yönlendirme tablosu yardımı ile dağıtıcılar ve sunucular üzerinden hedefe erişir. Kablosuz sensör ağlarındaki anlamı ise ortam erişim katmanı mesajı sadece komşu modüllere iletmekten sorumlu iken, ağ katmanı bu mesajın gerekirse hoplayarak kaynağından hedefine iletilmesinden sorumludur. Bunu başarabilmek için daha önceden Bölüm 2.1.3'te anlatılan yönlendirme protokollerini kullanır. KSA'lar için özel geliştirilmiş RPL bunun en güzel örneğidir.

2.2.6 Ulaşım Katmanı

Standart bir OSI katmanı olan ulaşım katmanı üst katmanlardan gelen veriyi kontrollü bir şekilde hedefine göndermek üzere ağ katmanına iletir. Ulaşım katmanının en bilinen örnekleri TCP (Geçiş Kontrol protokolü) ve UDP (Kullanıcı Veri Birimi Protokolü) protokolleridir.

Ulaşım katmanının en önemli görevlerinden birisi gönderilecek verinin ağ katmanının bir kerede gönderebileceğinden büyük olması halinde verinin küçük parçalara bölünmesidir. Bunun dışında kullanılan protokole göre ulaşım katmanında yapılabilecek işlemler;

- Mesajın güvenli bir şekilde hedefine erişmesini sağlar. Eğer mesaj ağ üzerinde kaybolduysa mesajın tekrar gönderiminden sorumludur.
- Mesajın gönderildiği sıra ile alınmasını sağlar. Ağ üzerinde bütün paketler aynı yönlendirme ile gitmeyebilir. Bu durumda sonradan gönderilen paketler hedefine daha önce erişmiş olabilir. Bölünmüş paketlerin aynı sıra ile birleştirilmesi bu katmanda yapılır.
- Mesajın gönderimi esnasında gönderim hızını kontrol eder. Örneğin hattın bant genişliği gerekenden daha az ise gönderim hızını bant genişliğinin kaldırabileceği seviyede düşürür.
- Port çoklama ile ağ biriminin birden çok noktaya bağlantı yapmasını sağlar. Bu sayede ağ birimi aynı anda birden çok ağ birimi ile iletişime geçebilir. Ulaşım katmanında çeşitli protokoller kullanma imkanı olsa da KSA'ların kısıtlamaları düşünüldüğünde basitliğinden ötürü UDP tercih edilmektedir.

2.2.7 Uygulama Katmanı

Kullanıcıya en yakın katman uygulama katmanıdır. Uygulama tasarımcısının alt katmanları bilmesine gerek kalmadan ağ üzerinde çeşitli uygulamalar koşturmasını sağlar. Bunlardan en bilindikleri FTP, HTTP ve IMAP gibi protokollerdir. Ancak KSA'ların düşük bellek ve işlem gücü kapasitelerinden ötürü bu tarz ağır uygulamaları kaldırmaları pek mümkün olmamaktadır. Onların yerine MQTT (Telemetri Mesajları Sıralama Protokolü) gibi düşük kaynak gereksinimi olan uygulamalar kullanılmaktadır.

2.3 Geliştirme Ortamı

2.3.1 Contiki İşletim Sistemi

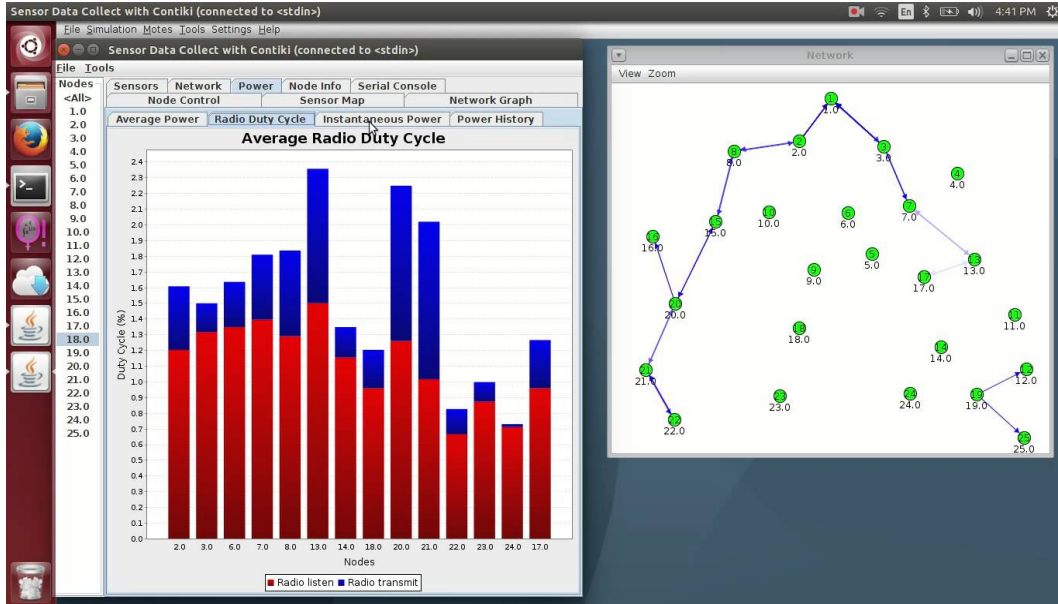
Contiki [9] KSA'larında mevcut olan kısıtlamalar temel alınarak hazırlanmış, içerisinde daha önceden anlatılan katmanları barındıran bir işletim sistemidir. Katmanların içerisindeki algoritmaların kolay bir şekilde uygulanabilmesi için gerekli zamanlama fonksiyonları, bellek yönetimleri ve çoklu görev çalıştırabilme özellikleri gibi çeşitli altyapıları sağlar. Medya ortam katmanı ile üst katmanları birbirine bağlayan sürücü programlarını içerir. Bu sayede farklı firmaların mikroişlemcileri uyum içinde kullanılırken üst katmanlar için etkilenmeden uygulamalar yazılabilir [51]. C programlama dili ile geliştirildiğinden birçok mikroişlemci mimarisi ile çalışabilir. Şekil 2.12 ile Contiki işletim sisteminin desteklediği bazı donanımlar gösterilmektedir. Contiki diğer KSA işletim sistemleri arasında içerisinde IP iletişim katmanlarını barındıran ilk işletim sistemi olma özelliğini taşır. Contiki çok az kaynak gereksinimine ihtiyaç duyar. Örneğin TCP/IP protokol katmanını kullanan bir uygulama yaklaşık 50kB kalıcı belleğe, 6kB geçici belleğe ihtiyaç duyar ki, bu değerlere günümüz mikrokontrolcüler kolaylıkla ulaşabilmektedir.

Mikroişlemci/Çip Üzerinde Sistem	Radio Çipi	Deneme Kartı	Cooja Simülasyon Desteği
TI CC2538	Dahili / CC1200	RE-Mote	-
nRF52832	Dahili	nRF52 DKEKE	-
RL78	ADF7023	EVAL-ADF7023DB1	-
TI CC2538	Dahili	cc2538dk	-
TI MSP430x	TI CC2420	exp5438, z1	+
TI MSP430x	TI CC2520	wismote	+
Atmel AVR	Atmel RF230	avr-raven, avr-rcb, avr-zigbit, iris	-
Atmel AVR	TI CC2420	micaz	+
Freescale MC1322x	Dahili	redbee-dev, redbee-econotag	-
TI MSP430	TI CC2420	sky	+
TI MSP430	TI CC1020	msb430	-
TI MSP430	RFM TR1001	esb	+
Atmel Atmega128 RFA1	Dahili	avr-atmega128rfa	-
Microchip pic32mx795f512l	Microchip mrf24j40	seed-eye	-
TI CC2530	Dahili	cc2530dk	-
6502	-	apple2enh, atari, c128, c64	-

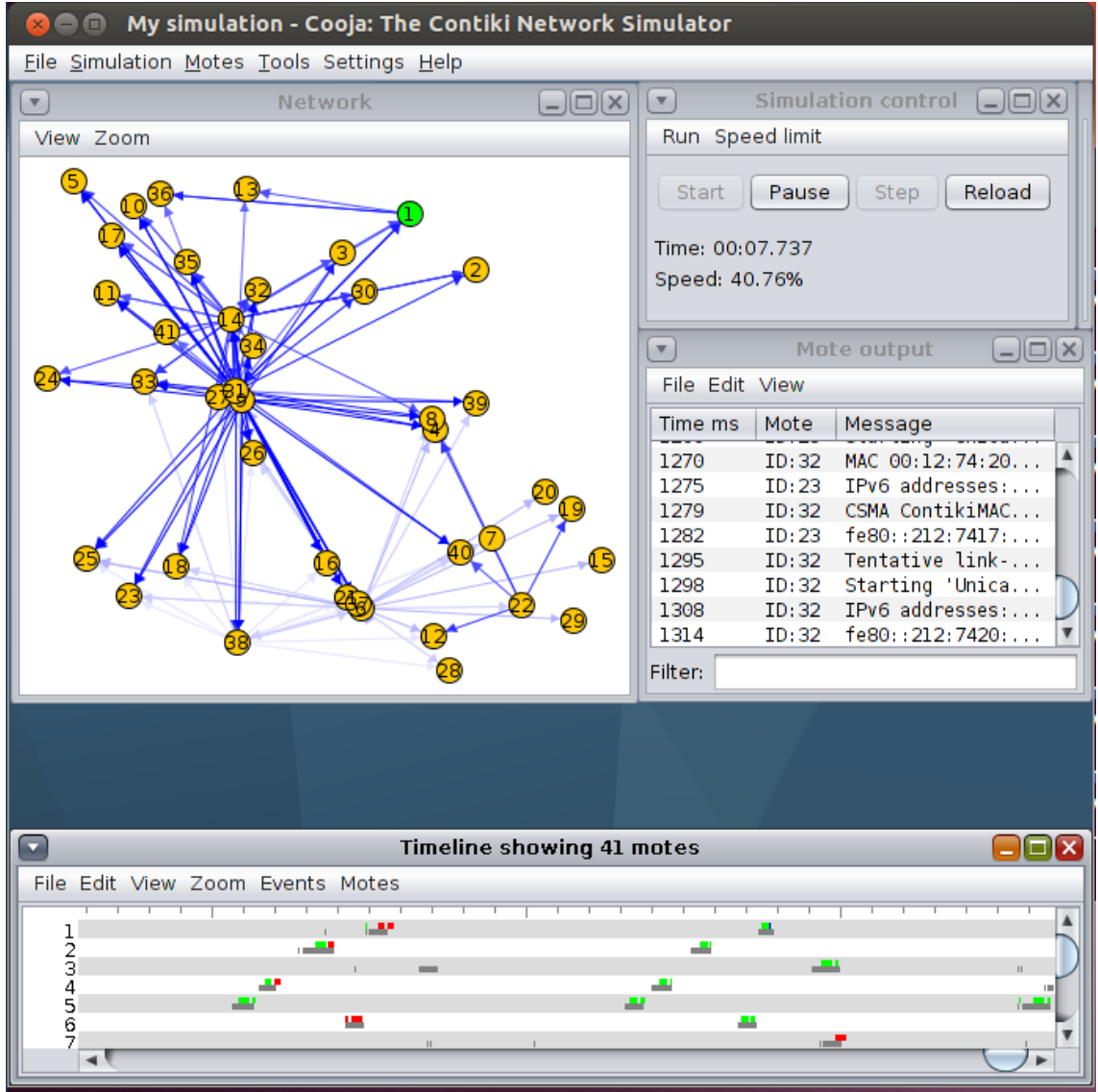
Şekil 2.12: Contiki tarafından desteklenen bazı mikrokontrolcüler [51]

2.3.2 Cooja Ağ Simülatörü

KSA'lar çok sayıda birimden oluştuğundan, geliştirme ve test aşamasında geliştirme kartları kullanmak çok zaman kaybettiricidir. Her kartın tek tek programlanması, kartlardan aynı anda ölçümler alınması, kartların coğrafik olarak geniş alanlara dağıtılması ihtiyacı gibi sorunlar sebebiyle geliştiriciler çoğu zaman simülatör kullanmayı tercih ederler. Bu ihtiyacı karşılamak için Cooja [10] ağ simülatörü geliştirilmiştir. Cooja JAVA tabanlı bir simülatör olup, Contiki tarafından üretilen uygulamaları çalıştırabilmektedir. Cooja çeşitli eklenebilir özellikler ile benzetim yapılacak birim veya birimleri detaylı şekilde inceleme imkanı sunar. Birimlerin giriş noktalarına değerler verilebilir, çıkış noktaları gözlemlenebilir, cihaz ile terminal bağlantısı kurulabilir, zaman ekseni üzerinde ne zaman alma gönderme yapıldığı, hangi aralıklar ile ortam dinlemesi yapıldığı incelenebilir, işlemci içerisinde tanımlanmış değişkenlerin değerleri gözlemlenebilir ve daha birçok ek özellik eklenebilir. Cooja simülatörü KSA'ları benzetim yaparken cihazların arasındaki ilişkileri de inceleme imkanı verir. Bağlantı kalitesini kontrol ettirebilir ve mesafe ile değişebilen bağlantı kalitesi tanımlatabilir. Cihazların belirlenen zamanlarda önceden tanımlanmış koordinatlara gitmesini sağlayarak hareketlilik etkileri incelenebilir. Şekil 2.13 ve Şekil 2.14 ile Cooja simülatörünün kullanıcı arayüzü gösterilmiştir.



Şekil 2.13: Cooja üzerinde çalışan veri toplama simülasyonu



Şekil 2.14: Cooja üzerinde incelenen kablosuz iletişimler

2.3.3 Tasarlanan Geliştirme Kartı

Simülasyon üzerinde alınan sonuçların gerçek ortamda da denenip doğrulanması her zaman bir gerekliliktir. Gerçek dünyada yapılan testler benzetimlerde gözlemlenemeyecek sorunları ortaya çıkarmaktadır. Ayrıca benzetim ortamında doğrulukla test edilemeyen kablosuz ağın bağlantı kalitesi, kapsama alanı, ortamda bulunan radyo gürültüsünün bağlantı kalitesine etkisi, anten tasarımından kaynaklanan performans özellikleri ve harcanan enerji gibi daha birçok testin gerçek donanımlar üzerinde yapılması gereklidir. Bu sebeple tez kapsamında yapılan çalışmaların gerçek dünyada da test edilebilmesi için bir takım kıstasları

sağlayan bir geliştirme kartı tasarlanmıştır. Tasarlanan kart kullanıcıya kolay program yükleme altyapısı sağlamakta ve kullanıcı veya diğer çevre birimleri ile haberleşmek için birçok seçenek içermektedir. KSA'ların kısıtlamalarından olan küçük boyutlar, çeşitli bağlantı arayüzleri ve düşük maliyet kartın tasarlanmasına öncülük etmiştir. Geliştirme kartının birim fiyatı 5\$ civarında olup seri üretim halinde kolaylıkla 2\$-3\$ aralığına inebilmektedir. Geliştirme kartı mikrokontrolcü, baskı devre kartı, anten ve çevre birim malzemelerinden oluşmaktadır.

2.3.3.1 Kablosuz Mikrokontrolcü

Kart tasarımı yapılırken en büyük problemlerden birisi kullanılacak işlemcinin seçimi olmuştur. Kullanılacak işlemcinin seçilmesinde KSA'lar için gerekli şartları sağlaması, adet bazında tedarik edilebilmesi, kolay ve standart arayüzler ile programlanabilmesi, Contiki işletim sistemi ile uyumlu olması, ucuz olması, besleme devrelerinin kompleks yapıda olmaması gibi birçok parametre önem arz etmektedir. Bahsi geçen filtreler aracılığı ile piyasada bulunan onlarca işlemci arasından seçim yapıldığında Texas Instruments firmasından CC2538SF53 tip numaralı malzemenin kullanılmasına karar verilmiştir. Ek E.1 ile çipin iç yapısı gösterilmiştir.

Genel Özellikleri

- Maksimum 32MHz saat hızında düşük güçlü ARM Cortex M3 mimarili çekirdek
- Flash tabanlı 512KB kalıcı bellek ve SD tabanlı 32KB geçici bellek
- cJTAG ve JTAG programlama ve hata ayıklama arayüzü
- UART üzerinden programlanabilme
- Akım tüketimi 400nA seviyelerine inebilen çeşitli düşük güç modları
- 2V-3.6V aralığında geniş besleme gerilimi
- -40°C +125°C endüstriyel çalışma sıcaklık aralığı
- 56 bacaklı 8mm-8mm genişliğinde QFN56 paket
- Contiki işletim sistemi ile uyumlu çalışabilme

Çevrebirim Özellikleri

- 4 adet 32bit çözünürlüklü sayaç
- 8 kanallı 12Bit çözünürlüklü ADC
- USB 2.0 12Mbps arayüz
- 2 Adet SPI
- 2 Adet UART
- 1 adet I2C arayüzü
- 32 adet genel amaçlı giriş çıkış noktası

Radyo Almaç-Göndermeç Özellikleri

- 2.4GHz IEEE 802.15.4 uyumlu fiziksel ve ortam erişim katmanı
- -97dB alıcı hassasiyeti
- En fazla 7dB olacak şekilde, 30dB ayarlanabilir güç çıkışı
- 100dB çözünürlüklü alınan sinyal gücü ölçümü

2.3.3.2 Anten Tasarımı

KSA'larda kablosuz bağlantı kalitesini ciddi şekilde etkileyen faktörlerden birisi de antendir. KSA'larda antenin birimin maliyetini arttırmaması için olabilecek en ucuz şekilde yapılması gerekmektedir. Ayrıca performans özelliklerinden ödün verilmeden birimin toplam boyutunu arttırmaması için antenin olabildiğince küçük olması gerekmektedir. Geliştirme kartında da bu kıstaslar göz önüne alınarak "Ters F" [52] modelinde bir baskı devre anten kullanılmıştır. Kullanılan anten Texas Instruments tarafından tasarlanmış, 2.4GHz bandında, 370MHz bant genişliğine sahip olan ve direkt olarak 50ohm empedans gösteren bir antendir. Ek E.2 ile anten boyutları gösterilirken, Ek E.3 ile antenin frekans cevabı gösterilmektedir. Ayrıca karta baskı devre anteni dışında harici bir anten takabilmek için bir adet UFL konektör bulunmaktadır.

2.3.3.3 Baskı Devre Kartı

Baskı devre kartı tasarlanırken düşük maliyet için en önemli kıstaslar kartın boyutu ve tabaka sayısıdır. Kartın boyutu üzerinde barındırdığı bileşenler için yeteri kadar büyük olacak şekilde seçilirken, standart geliştirme ortamlarında kullanılacak kadar küçük seçilmiştir. Kart üzerinde kolaylıkla geliştirme yapmak için, giriş çıkış noktaları arası mesafe tasarlanan kartın "breadboard" isimli deney tablalarına takılabileceği şekilde belirlenmiştir. Kart 2 tabaka olarak tasarlanmış olup, kart kalınlığı 800 μm , bakır kalınlığı ise 35 μm olacak şekilde üretilmiştir. Kart uzunluğu 45 mm ve genişliği ise 28 mm olacak şekilde üretilmiştir. Kartın enerji beslemesi 3 ayrı yerden olabilir.

- **Kartı USB konektörü üzerinden besleme:** Karta "mini-USB" kablosu takılarak karta güç verilebilir. USB üzerinden kart ile iletişimde kurulabildiğinden kartı çalıştırmak ve kart ile haberleşmek için gerekli en az donanım bileşeni budur.
- **Kartı "Güç Giriş Noktası" üzerinden besleme:** Kart üzerinde çip için gerekli çalışma gerilimi üreten bir voltaj regülatörü bulunmaktadır. 200 mA 3,3 V çıkış gerilimi üreten bu entegre, radyo çipini ve diğer çevre bileşenlerini beslemektedir. Bu girişten kart beslenirken regüle edilmemiş şekilde 6 V-3,4 V aralığında voltaj çıkışı veren herhangi bir kaynak bağlanabilir.
- **Kartı "Çekirdek Beslemesi Noktası" üzerinden besleme:** Bu yöntemde ise çipi bu nokta aracılığı ile direkt beslemek mümkün olmaktadır. Bu sayede çip gerilimi 2 V seviyelerine kadar indirilerek en düşük enerji tüketimi sağlanabilir. Ancak bu nokta hiç bir koruma olmadan çipe direkt bağlandığından uygulanan gerilimin kontrollü ve gürültüsüz olması gerekmektedir.

Tasarlanan kartın üzerinde 1 adet kullanıcı düğmesi, 1 adet çipi baştan başlatma düğmesi, karta güç uygulandığında yanan 1 adet kırmızı LED ve kırmızı-yeşil-mavi renklerinde 3 adet LED ışık bulunmaktadır. Ayrıca kart üzerinde çipin çalışması için gerekli 32 MHz hızında saat işareti ve 32.768 KHz hızında yavaş saat işareti üreticileri bulunmaktadır. Kartın şeması Ek E.4 ve Ek E.5 ile gösterilirken, Ek E.6 ve Ek E.7 ile kartın tasarım ortamındaki 3 boyutlu görüntüsü görülebilir. Ek E.8 ile gösterilen şekilde ise kartın üretilmiş hali bulunmaktadır.

3. RPL YÖNLENDİRME PROTOKOLÜ

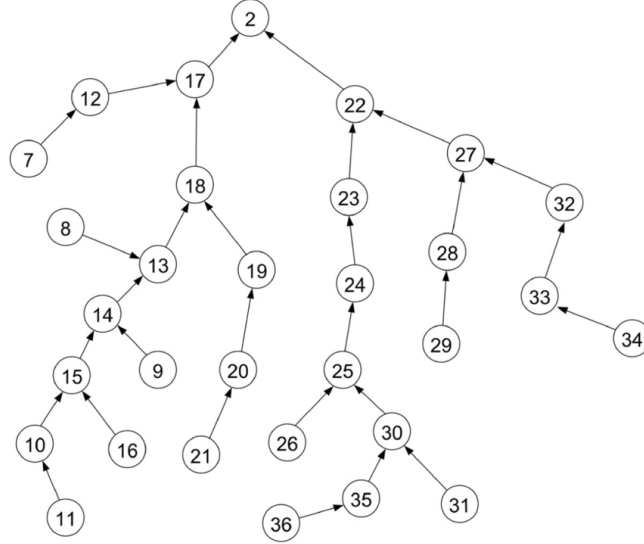
RPL (Düşük Güçlü ve Kayıplı Ağlar için Yönlendirme Protokolü) düşük güçlü ve kayıplı bağlantılara sahip KSA'lar için geliştirilmiş IPv6 tabanlı bir yönlendirme protokolüdür [8]. Bağlantı durumu yönlendirmesi kullanan protokollere göre, mesafe vektörü yönlendirmesini kullanan RPL çok az bir bellek tüketimine gereksinim duyar. RPL protokolü proaktif yönlendirmeye sahiptir. Bundan dolayı ağ kurulduğu andan itibaren yönlendirme protokolü çalışmaya başlar ve periyodik olarak ağın durumu hakkında bilgi toplar. RPL protokolünde yönlendirme topolojisi ağaç şeklindedir ve buna Yönlü Düz Ağaçlar *DAG (Yönlü Düz Ağaçlar)* denir. DAG yapısı ağaç yapısının yön almış halidir. RPL *DODAG (Hedefe Yönelik Yönlü Düz Ağaçlar)* şeklinde kurulur. Yani ağın internet gibi harici ağlara çıkış noktası olan birim, o DAG'ın kök birimi olarak davranır. Bir RPL ağında birden çok DODAG bulunabilir ve bütün DODAG'ların birleşimine RPL olayı denir. Her RPL olayının özel bir kimlik numarası vardır. Ağda aynı anda birden çok RPL olayı bulunabilir ve her bir birim birden fazla sayıda olaya bağlı olabilir. Ancak her bir birim tüm RPL olayları için bir DODAG yapısına aittir [53].

RPL'de kök birim dışındaki bütün birimlerin bağlanmayı tercih ettiği bir ebeveyn birim bulunur. Ebeveyn birim, çocuk birimi ağın kök birime daha yakın olan kısımlarına bağlar. Örneğin çocuk birime gelen bir mesajın hedefi çocuk birimin yönlendirme tablosunda kayıtlı değilse, mesajı hedefine iletmesi için ebeveyn birime verir. Bu döngü paket hedefine erişinceye kadar devam eder.

Tercih edilen ebeveynin seçimi ağ kalitesi açısından çok önemlidir. Yanlış seçilen yönlendirme tercihleri ağ üzerinde mesajların kaybolmasına ve dolayısıyla enerjinin boşuna kullanılmasına yol açar. Bu sebeple RPL'de yönlendirmenin efektif bir şekilde yapılmasını sağlayan birçok özellik vardır.

3.1 Ağ Topolojisi

KSA'larda genellikle kablolu ağlardaki gibi önceden belirlenen bir ağ bağlantı şeması olmadığından, RPL protokolünün önce yönlendirme tablosunu kurması gerekmektedir.



Şekil 3.1: Ağ yapısı

RPL protokolü ağ yapısındaki şemasını kurmaya ağacın en tepesinde bulunan kök birimden başlar ve Şekil 3.1 ile görüldüğü gibi yapraklara ulaşmaya kadar bu kurulum devam eder. RPL'de tercih edilen bir ebeveyn olsa da o birimin yokluğunda bağlanılabilecek alternatif ebeveynlerde yönlendirme listeye eklenmektedir. Bu sayede KSA'ların gereksinimi olan ihtiyaç fazlası yönlendirmeler de [54] hazır bekletilir. RPL mesajları akış yönüne göre “yukarı yönlü” ve “aşağı yönlü” olarak iki kategoriye ayırır. Yukarı mesajlar yaprak birimlerden kök birime doğru iletilirken, aşağı mesajlar kök birimden yaprak birimlere doğru iletilir.

3.1.1 Yukarı Yönlü Yönlendirme

Yukarı yönlü bir mesaj trafiği oluşturmak için gerekli bilgi DODAG bilgisi içerisinde bulunur. DODAG bilgisinde birimlerin tercih ettiği ebeveyn birimler bulunur. Bu bilgi vasıtası ile kök birime mesaj gönderilirken mesaj önce tercih edilen ebeveyn birime gönderilir. O birim de mesajı kendi ebeveyn birimine gönderir, bu döngü mesaj hedefine ulaşmaya ka-

dar devam eder. DODAG içerisinde kimin ebeveyn, kimin çocuk birim olacağını, yani ağaç topolojisi içerisinde kimin köke yakın, kimin yapraklara daha yakın olacağını, rütbe faktörü belirler. Rütbe faktörü görev fonksiyonu sonucunda hesaplanır ve kök her zaman 0 rütbe faktörüne sahip olurken, kökten uzaklaştıkça bu sayı büyür. Ebeveyn birimin rütbe faktörü her zaman ona bağlı olan çocuk birimden küçüktür. Eğer tersi olursa döngü oluşur ve hızlı bir şekilde düzeltilmesi gerekir.

RPL protokolünde DODAG bilgisi ebeveynlerin gönderdiği, DODAG bilgisini yayan bilgi olan *DIO (DODAG Bilgi Objesi)* ile dağıtılır. Bu mesajı alan çocuk birimler ağ ve ağ parametreleri hakkında bilgi sahibi olur. Eğer uygunluk sağlanırsa, DIO mesajının göndericisi olan birimi tercih edilen ebeveyn birim olarak listelerine ekler. Birimler ortamdan DIO mesajı almak isterlerse, *DIS (DODAG Bilgi Talebi)* mesajı gönderir. Mesajlar ile detaylı bilgiye daha sonraki kısımlarda değinilecektir.

3.1.2 Aşağı Yönlü Yönlendirme

Mesajın kök birimden ağın herhangi bir yerindeki bir birime gönderilmesi için gerekli yönlendirmeyi içerir. Bu yönlendirmenin oluşturulması için *DAO (DODAG Hedef Duyuru Objesi)* mesajları kullanılır. Birim DIO mesajını alıp tercih ettiği ebeveyn birimi seçtikten sonra ona DAO mesajı yollar. Bu mesaj kök birime kadar iletilerek, kök birim dahil olmak üzere yol üzerindeki bütün ebeveyn birimlerin DAO mesajını yollayan çocuk birime nasıl erişebileceklerinin tanıtımı yapılır.

3.2 Yönlendirme Kıstasları

KSA'larda yönlendirmenin verimli bir şekilde yapılabilmesi karmaşık stratejiler ile mümkün olur. Bundan dolayı DODAG yapısının oluşması ve devamlılığının sağlanması için birtakım kıstaslar oluşturulmuştur. Bu kıstaslar sayesinde yönlendirme için kullanılacak yolun kalitesi belirlenip, yolun kullanılıp kullanılmayacağına karar verilir. Bu kıstaslardan en çok kullanılanları en az hop sayısı, enerji ve beklenen gönderim sayısıdır.

3.2.1 En Az Hop Sayısı

Yönlendirme protokollerinde çoğunlukla kullanılan bu kıstas alıcı ve gönderici arasında, mesajın en az hop sayısı ile iletilmesini sağlamaya yarar. Bu sayede mesaj aktarımı sırasında en az sayıda birim mesajın iletimine dahil olup, toplamda daha az enerji harcanması öngörülür. Ancak bu kıstas seçilen yolun en uygun yol olup olmadığını belirlemez. Örneğin seçilen yolda bütün mesaj trafiği bir ara birim üzerinden aktarılıyor olabilir. Bu aktarım ilgili birimin enerjisini hızla tüketecektir.

3.2.2 Enerji

Enerji KSA'larda en önemli parametre olduğu için enerjinin dengeli kullanılması kritiktir. Yönlendirme tablosu kurulurken birimlerin kalan enerjileri gözetilmeli, enerjisi az kalan birimin yönlendirici birim olarak seçilmemesine özen gösterilmelidir. Bununla beraber gücünü şebekeden alan birimlerin enerji sorunu olmadığından yönlendirmelerin büyük bir kısmını bu birimlerin yapması daha uygun olacaktır. Ancak sonuç olarak olması gerekenden daha uzun hatlar oluşacak ve potansiyel veri kayıpları artacaktır.

3.2.3 Beklenen Gönderim Sayısı

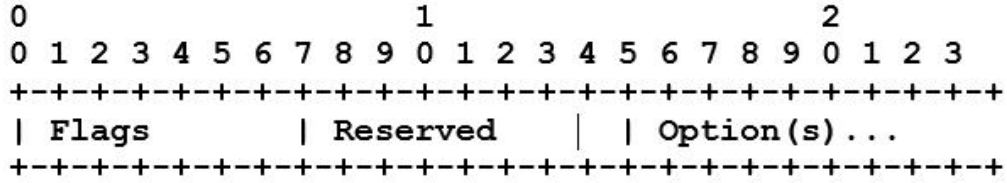
Beklenen gönderim sayısı bir mesajın iki komşu birim arasında güvenli bir şekilde iletilebilmesi için MAC katmanında gerçekleşen gönderim sayısıdır. İki birim arası bağlantı kalitesi çok iyi ise beklenen gönderim sayısı 1 olacaktır. Link kalitesi düştükçe beklenen sayı artacaktır. Bu değer birimlerin birbirleri ile hangi kalitede haberleştiklerini ortaya çıkarmaktadır. Ancak bu değerlerin ortaya çıkması için birimler arasında bir miktar haberleşme yapılması gerekmektedir. Bu da hızlı kurulması gereken ağların kurulumunu yavaşlatacaktır.

3.3 RPL Kontrol Mesajları

RPL yönlendirme katmanını oluşturmak ve devamlılığını sağlamak için ICMPv6 tabanlı 4 adet kontrol mesajı kullanılır. Bu mesajlar DIS, DIO, DAO ve DAO-ACK sırası ile açıklanmaktadır.

3.3.1 DIS

DIS mesajı kontrol mesajları arasında en basit yapıya sahip olan mesajdır. Amacı birimin etrafında bulunan diğer birimlerden bilgi almasını sağlamaktır. Bu mesajı alan birimler DIO mesajı göndererek kendi durumları ve bağlı oldukları DODAG hakkında bilgi verirler. DIS mesajı birimin bir ağa katılmak istemesi halinde ve aynı zamanda çevredeki birimlerin periyodik DIO mesajlarını beklemek istememesi halinde gönderilir. Mesajın yapısı Şekil 3.2 ile görülebileceği gibi şu an kullanılmayan “flags” ve “options” sekmelerinden oluşur. Ayrıca standartlarda iç yapısı tanımlanmamakla birlikte kullanıcının isteği doğrultusunda doldurulan “options” sekmesi bulunmaktadır.

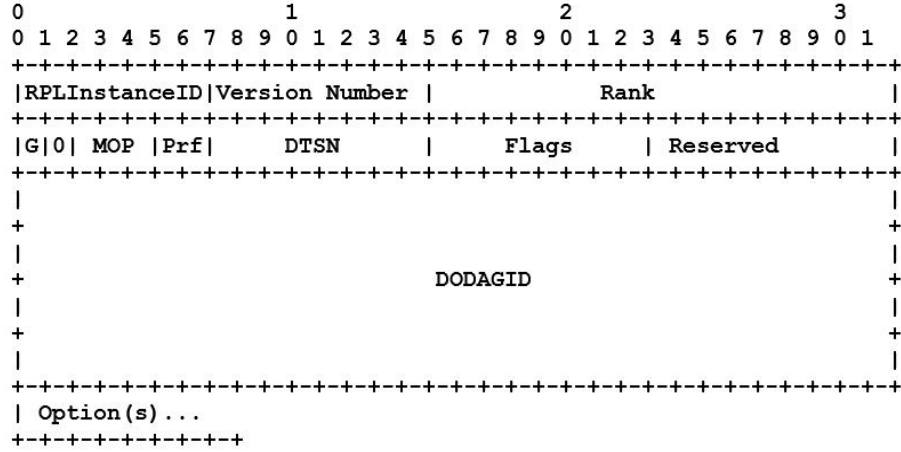


Şekil 3.2: DIS mesajının yapısı [8]

3.3.2 DIO

DIO mesajı komşu birimlere periyodik olarak gönderilir. Amacı gönderen birimin bağlı olduğu DODAG hakkında bilgi yaymaktır. DIO mesajı vasıtası ile yukarı yönlü yönlendirme sağlanır. Mesajı alan ve rütbe faktörü daha büyük olan birim (o birime göre çocuk birim) DIO mesajındaki bilgileri kullanarak ağa katılır ve DIO gönderimi yapan birimi tercih edilen ebeveyn birim olarak seçer. DIO mesajı ağ hakkında bir çok bilgi içermektedir. Şekil 3.3 ile ilgili bilgiler gösterilmiştir.

- RPLInstanceID: Bu bilgide DODAG’ın içinde bulunduğu RPL olay kimlik numarası bulunur. Bir olay içinde 255 ayrı DODAG bulunabilir.
- Version number: DODAG’ın o anki versiyon numarasıdır. Versiyon numarası ağ parametreleri ile ilgili bir değişiklik olduğu zaman kök birim tarafından 1 arttırılır. DODAG içerisindeki bütün birimlerin aynı versiyon numarasına sahip olması gerekmektedir.



Şekil 3.3: DIO mesajının yapısı [8]

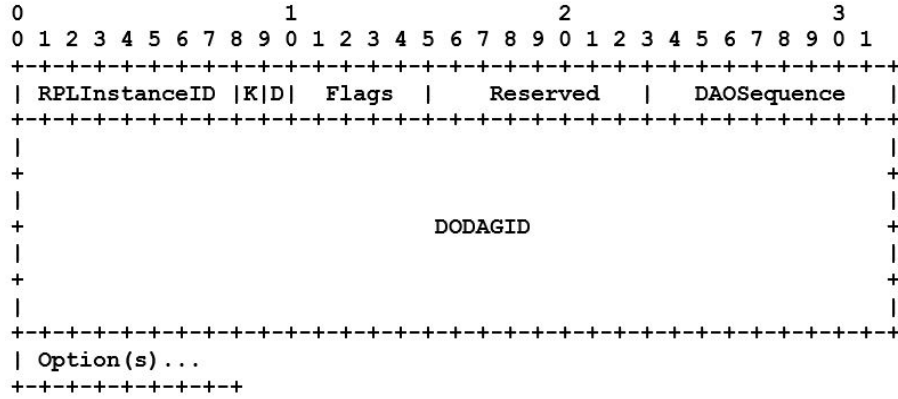
tedir. Versiyon numarası düşük olan bir birim aldığı DIO mesajından en güncel parametreleri öğrenebilir. Bu sayede gerekli güncellemeler bütün DODAG içerisinde dağıtılmış olur.

- Rank: DIO mesajı yollayan birimin DODAG içerisinde kök birime olan uzaklığını belirtir. Sayı ne kadar büyük ise birim kök birimden o kadar uzaktır. DIO mesajını alan bir birim mesajı gönderen birimin rütbesi ile kendi rütbesini karşılaştırır. Gönderici daha düşük rütbeyle sahip ise o birimi alternatif ebeveyn listesine ekler.
- DTSN (Hedef Duyuru Tetikleme Sıra Numarası): Aşağı yönlü yönlendirmenin oluşumu ve devamlılığını sağlayan kısımdır. Ebeveyn birimin kendi çocuklarından DAO mesajı talep etmesi için kullanılır. Bu sayı ebeveyn birim tarafından artırıldığında, ebeveyn birim tarafından gönderilen DIO mesajını alan çocuk birim DAO mesajı gönderir. Böylelikle aşağı yönlü hattın geçerliliği kontrol edilmiş olur.
- Grounded (G): Kök birimin KSA dışında başka bir ağa bağlı olup olmadığını belirtir.
- MOP (Çalışma Şekli): RPL'in çalışma koşulunu belirler. Koşullar içerisinde aşağı yönlü yönlendirmenin olup olmayacağı, yönlendirme tablosunun bütün birimlerde mi ya da sadece kök birimde mi tutulacağı belirtilir.
- DODAGPreference: DODAG kök birimin aynı RPL olayı içinde bulunan diğer DODAG kök birimleri arasındaki tercih sırasını belirler. Sayı büyüdükçe kök birim daha fazla tercih edilir olur.

- DODAGID: DODAG'ı oluşturan kök birimin IPv6 adresidir.
- Options: DIS mesajında olduğu gibi DIO mesajlarında da kullanıma açık bir kısım yaratılmıştır.

3.3.3 DAO

DAO mesajının amacı aşağı yönlü yönlendirmeyi sağlamak ve güncel tutmaktır. Kök birim dışında bütün birimler DAO mesajı gönderirler. DIO mesajının alınmasından sonra çocuk birim tercih edilen ebeveyn birime DAO mesajı yollar. Ebeveyn bu mesajı kök birime ulaşıncaya kadar kendi ebeveynlerine yollar. Mesaj kök birime ulaştığında kök birim ve mesajı alan bütün ebeveyn birimler DAO mesajını gönderen çocuk birime nasıl ulaşacaklarını öğrenirler. DAO mesajının yapısı Şekil 3.4 ile gösterilmektedir.

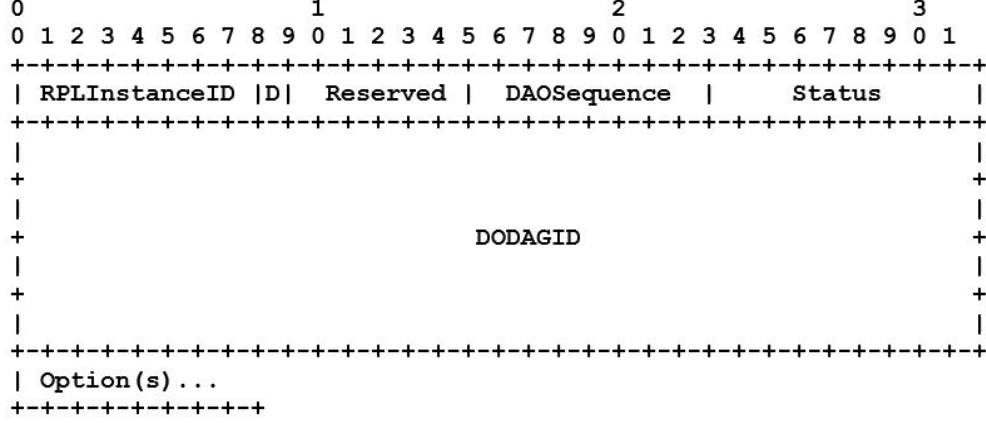


Şekil 3.4: DAO mesajının yapısı [8]

- RPLInstanceID: DIO mesajından öğrenilen RPL olay numarasıdır.
- K: DAO mesajını alan birimin, gönderen birime alındı onayı gönderip göndermeyeceğini belirler.
- D: DODAGID'nin DAO mesajıyla gönderilip gönderilemeyeceğini belirler.
- DAOSequence: Gönderilen her DAO mesajında 1 arttırılır.
- DODAGID: DIO mesajından öğrenilen DODAG kimliğidir.
- Options: DIS mesajında olduğu gibi DAO mesajlarında da kullanıma açık bir kısım yaratılmıştır.

3.3.4 DAO-ACK

DAO mesajını alan ebeveyn tarafından DAO göndericisine gönderilir. Amacı gönderilen DAO mesajının alınıp alınmadığını, alınmış olsa bile mesajın geçerli olup olmadığını bildirir. Mesaj yapısı Şekil 3.5 ile gösterilmiştir.



Şekil 3.5: DAO-ACK mesajının yapısı [8]

- RPLInstanceID: DIO mesajından öğrenilen RPL olay numarasıdır.
- D: DODAGID'nin DAO mesajıyla gönderilip gönderilemeyeceğini belirler.
- DAOSequence: Alınan DAO mesajındaki sıra numarasını DAO göndericisine geri gönderir. Bu sayede çok sayıda gönderim olmuş ise hangi mesajın cevabının hangisi olduğu anlaşılır.
- Status: Standartlarda tam olarak tanısı koyulmamış olsa da var olan RFC6550 versiyonundaki tanıma göre, bu değer 0 olursa alınan DAO kabul edilmiştir. 1-127 arasında ise DAO alıcısı, DAO göndericisinin ebeveyni olarak davranmak ister ancak uzun vadede daha uygun bir ebeveyn bulmasını tavsiye eder. Status değeri 127-255 arasında ise gönderilen DAO mesajının kabul edilmediğini bildirir [8].

3.4 Görev Fonksiyonu

Görev Fonksiyonu rütbe faktörünün hesaplanması ve tercih edilen ebeveyn seçimi için gerekli yönlendirme kıstaslarının ve fonksiyonlarının bütünüdür. Her RPL olayı kendine özel

bir görev fonksiyonu içerebilir.

Görev fonksiyonu uygulamanın gerektirdiği şekilde biçimlendirilebilir. Örneğin yüksek veri hızı gereken uygulamalarda görev fonksiyonu için en uygun yol en az gecikmeli yol iken, düşük trafiğin yeterli olduğu, bataryalı birimlerin oluşturduğu KSA'lar için en uygun yol gecikmenin önemsiz olduğu ancak en düşük enerjinin harcanacağı yoldur. Görev fonksiyonu rütbe faktörünü hesaplamak için birçok parametreyi içerebilir. Bataryalı birimlerin prize bağlı birimlerle ortak kullanıldığı uygulamalarda trafiğin büyük bir kısmını prize bağlı cihazlardan geçirmesi ağ ömrünü uzatacaktır. Bunu yapabilmek için prize bağlı cihazların rütbe faktörünü düşük tutan bir görev fonksiyonu gereklidir.

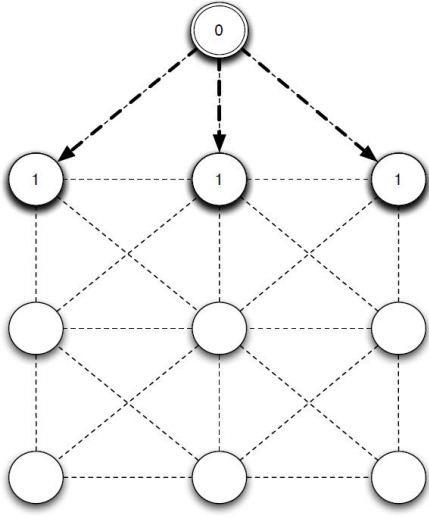
RPL protokolü "OF0 (Görev Fonksiyonu 0)" ve "ETX (Beklenen Gönderim Sayısı)" isimli iki ayrı görev fonksiyonu içerir. OF0 en az hop sayısını yönlendirme kıstası olarak alınırken, ETX mesaj gönderim oranının iyiliğini kıstas olarak kullanır. RPL protokolünde görev fonksiyonu ayrı bir katman olarak geliştirildiğinden istenilen uygulamaya göre özel bir görev fonksiyonu geliştirilebilir. Şu an "OF0" RPL protokolünde varsayılan görev fonksiyonu olarak kullanılmaktadır [8].

3.5 DODAG Oluşumu

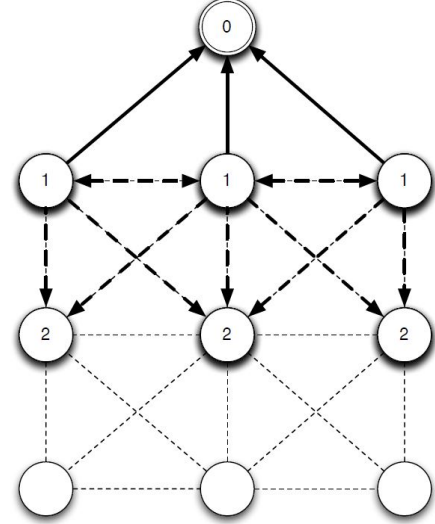
RPL protokolünde DODAG oluşumu kök birim ile başlatılır. Kök birim ağ parametrelerini ve çalışma modlarını belirleyerek DIO mesajı yayımlar. Yayımlanan bu mesaj Şekil 3.6a ile gösterildiği gibi dinleyen herkese gönderilir.

Ağ parametreleri RPLInstanceID ve DODAGID bilgileri değişmeden DIO mesajları ile yayılırken, mesajı alan her birim kendi rütbe faktörünü RPL olayında belirtilen görev fonksiyonuna göre hesaplar, uygunsa ebeveyn seçer ve kendi DIO mesajını dinleyen herkesin duyabileceği şekilde yayımlar. Şekil 3.6b bu durumu göstermektedir. Aynı döngü bütün birimler ağa katılincaya kadar Şekil 3.6c ve 3.6d deki gibi devam eder.

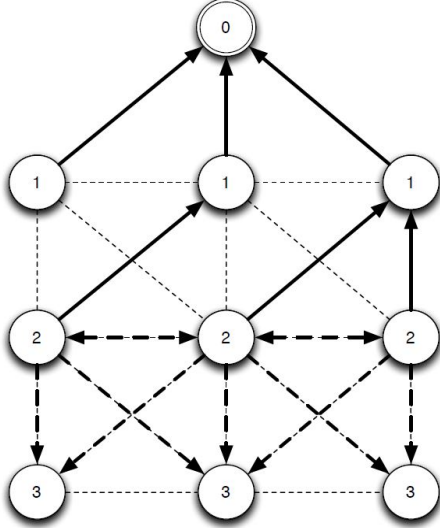
Eğer çalışma modlarından aşağı yönlü yönlendirme seçilmiş ise DAO mesajları gönderilir. DIO mesajını alıp rütbe faktörüne göre olası ebeveynler arasından en uygun olanı seçildikten sonra, ağa katılan bu birim ebeveyn birimine DAO mesajı yollar. Gönderilen DAO mesajı kök birime ulaşıncaya kadar Şekil 3.6d ile gösterilen yollar üzerinden ağ içinde yukarıya



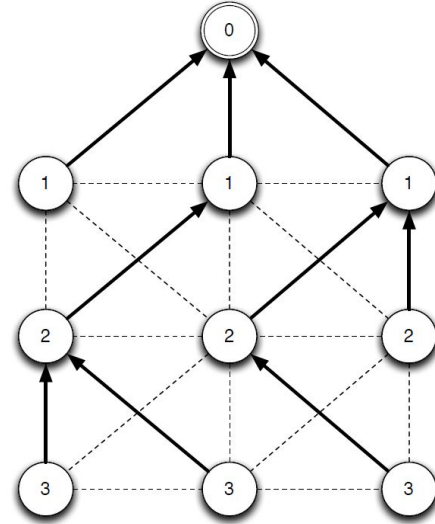
(a) Kök birim DIO mesajını dinleyen herkesin duya-
cağı şekilde yayar



(b) DIO mesajını alan birimler ebeveyn seçtikten sonra
kendi DIO mesajlarını gönderir



(c) Bütün birimler DODAG'a katılıncaya kadar DIO
mesajları gönderilmeye devam eder



(d) Bütün birimler DODAG'a katılmıştır

Şekil 3.6: DODAG oluşum süreci. Çizikli oklar DIO mesajlarını gösterirken, tam oklar tercih edilen ebeveyn seçimlerini göstermektedir.

doğru ilerler. Ağa katılım tamamlanınca kontrol mesajları *Trickle* [55] algoritmasından belirlenen sürelerle tekrarlanır. Bu sayede ağ içerisinde bir hareketlilik durumu, bir birimin çalışmaması veya bağlantı kalitesinin değişimi gibi bir durum olmuş ise tercih edilen ebeveyn birim alternatif ebeveynler ile değiştirilerek yönlendirmenin aksamaması sağlanır.

3.6 Trickle Zamanlayıcısı

Trickle KSA'lar içerisinde birimlerin kayıplı ağlarda düşük enerji kullanarak kolay uygulanabilir ve ayarlanabilir şekilde veri transferi gerçekleştirmesini sağlayan bir mesaj gönderim zamanlayıcı algoritmasıdır. *Trickle* algoritması ağda yayılması gereken yeni bir mesaj varsa gönderim zamanını dinamik olarak ayarlayarak ilgili mesajı yaymaya başlar. Ağda gönderilecek yeni bir mesaj yoksa mesajların gönderim periyodu azaltılarak enerji tasarrufu yapılır [55]. Bu sayede veri, ağ içerisinde az enerji harcayarak güvenli bir şekilde dağıtılmış olur. RPL protokolünde *Trickle* zamanlayıcısı DIO ve DAO kontrol mesajlarının gönderimi için kullanılır. Ağ içerisinde yayılması gereken yeni bir bilgi varsa (örneğin DODAG versiyonu değişimi) DIO gönderim sıklığı en sık haline gelir. Bundan sonra gönderilecek her DIO mesajı 2 kat uzun süre sonra gönderilir. Gönderim süresi sıklığı limit değere gelinceye kadar azaltılır. Böylelikle yeni mesaj ağ içerisinde hızlı bir şekilde yayılırken, uzun vadede enerji tasarrufu yapılabilir. *Trickle* algoritması için ayarlanabilir 3 parametre vardır:

- **Imin:** Gönderilecek 2 DIO mesajı arasındaki minimum süredir. Bu süre her gönderimden sonra ikiye katlanarak gönderim sıklığı azaltılır.
- **Imax:** Gönderilecek 2 DIO mesajı arasındaki maksimum süredir. İkiye katlanarak büyüyen gönderim sıklığının alabileceği en büyük değerdir.
- **Artıklık Sabiti:** Trickle algoritması mesaj gönderim zamanlarını ayarladığı gibi gereksiz mesajların da iletimine engel olur. Yeni kurulan ağlarda zamanlayıcılar düşük değerlere kurulduğu için aynı bilgiye sahip bir çok mesaj aynı anda gönderilir. Bundan dolayı DIO gönderecek birim DIO gönderim süresi içerisinde göndereceği DIO mesajı ile aynı bilgiye sahip başka DIO mesajları duyarsa DIO gönderimi iptal edebilir. Artıklık sabiti bu mesajdan kaç tane duyması halinde gönderimin iptal edileceğini belirler.

Trickle zamanlayıcısının RPL kontrol mesajı hızını ayarlamak için Eşitlik 3.1 kullanılır.

$$I_{min} = 2^{MINIMUM\ DIO\ ARALIGI} \quad (3.1)$$

$$I_{max} = I_{min} \times 2^{DIO\ ARALIK\ KATLAMASI} \quad (3.2)$$

4. KABLOSUZ SENSÖR AĞLARINDA HAREKETLİLİK VE GELİŞTİRİLMİŞ YÖNTEMLER

4.1 Kablosuz Sensör Ağlarında Hareketlilik

KSA'lar için kullanılan yönlendirme protokollerine hareketlilik desteği katılacaksa hareketliliğin tanımı ve bu konu ile ilgili daha önceden yapılan çalışmaların detaylı bir şekilde incelenmesi gerekmektedir.

4.1.1 Hareketliliğin Tanımı

KSA'lar gelecek için umut veren teknolojilerin başında gelir. Kablolu sistemlere göre kurulumları oldukça kolay ve sistem maliyeti oldukça azdır. Ancak KSA'larda ağ yapısı hareketlilikten dolayı zaman içinde değişir ve ağın bu değişime hızlı bir şekilde adapte olması gerekir. Aksi takdirde veri kaybı olacaktır. Örneğin hareket halindeki birim bir süreliğine bağlı olduğu ebeveyn birimin kapsama alanından çıkıp, bir başka alternatif ebeveyn birimin kapsama alanına giriyor olabilir. Bu süre zarfında yönlendirme tablosu hızlı bir biçimde güncellenmezse mesaj iletilirken yönlendirmelerde ara birim olarak hareketli birimin kapsama alanında olmadığı ebeveyn birim kullanılacaktır. Bu durumda hem yukarı yönlü mesajlar, hem de aşağı yönlü mesajlar hedefine ulaşamayacaktır. Bu durum hareketli birimin tercih ettiği ebeveyn birimi yenisiyle değiştirmesine kadar devam edecektir. Hareketlilik KSA'larda ciddi bir biçimde üzerine düşülmesi gereken bir konudur. KSA'larda hareketlilik 2 ana sınıfta incelenebilir.

- **Fiziksel Hareketlilik (Güçlü Hareketlilik)**

KSA içerisindeki bir birimin fiziksel olarak başka bir noktaya taşınmasıdır. Taşınma sensörün üzerine takıldığı objenin (insan, araç vb..) istemli hareketliyle olabileceği gibi rüzgar, doğal hareketler gibi istemsiz nedenlerden dolayı da olabilmektedir. Fiziksel hareketlilik, KSA'ların kullanıldığı bazı uygulamalarda kilit rol oynar. Örneğin hastanelerde hareketli bir hastanın hayati verilerini takip eden bir uygulamada [56]

veya afet durumlarındaki arama kurtarma çalışmalarında [57] kablosuz birimin üzerine takıldığı kişinin hareketliliğinden dolayı ağ içerisinde sıklıkla topoloji değişimlerine sebebiyet verecektir. Uygulamanın gerekliliklerine göre bu topoloji değişimleri sırasında en az gecikme ve en az paket kaybı olması gerekmektedir. Standart yönlendirme protokolleri genellikle durağan ağ yapıları için geliştirildiklerinden fiziksel hareketlilikte yüksek gecikme ve yüksek veri kaybı yaşanmaktadır.

- **Sanal Hareketlilik (Zayıf Hareketlilik)**

Birimin bir noktadan bir noktaya taşınması ile değil, ağ içerisine bir birimin eklenmesi, bir birimin çalışmaması veya gün içerisinde ortam gürültüsü ile birimler arası bağlantı kalitesinin bozulması gibi nedenlerle tetiklenen ağ değişimleridir. Bu tarz değişimler sıklıkla olmamakla birlikte, ağın sınırlı bir kısmında değişikliğe sebebiyet verirler. Bu tarz yavaş değişiklikleri özel bir geliştirme yapılmaksızın yönlendirme protokolleri yönetebilmektedir.

4.1.2 Hareketlilik Modelleri

KSA'lar birçok uygulamada kullanılabildiğinden karşı koymaları gereken hareketlilik modelleri de gün geçtikçe çeşitlenmektedir. Hareketlilik karşısında ağ performansını güçlendirecek bir protokol geliştirilmesi isteniyorsa önce hareketliliğin modelleri sınıflandırılmalıdır. Bundan dolayı hareketlilik 3 ana sınıfta modelleyebiliriz [58].

- **Yaya Hareketlilik Modeli**

Bu modelde üzerinde kablosuz sensör birimi olan yayaların veya hayvanların hareket özellikleri tanımlanır. Özelliklerin başında hızların 0-10 km/h mertebelerinde olması, çoğunlukla 2 eksenle hareket etmesi ve iç-dış mekanlarda kullanımdan dolayı yüksek bağlantı kalitesi değişimi gibi parametreler gelir. Bu modelde hareket vektörü oldukça değişkendir. Hareketlilik bir anda ortaya çıkıp bir anda sonlanabilir [59].

- **Araç Hareketlilik Modeli**

Araç hareketlilik modeli, yayalardan hızlı giden, üzerinde kablosuz iletişim birimi olan araçların tanımlamasını yapar. Arabalar, trenler, bisikletler bu modelin kapsamı içindedir. Araç hareketliliğinde hızlar genellikle 20-100 km/h aralığındadır. Araçların genellikle yüksek enerjili elektrik kaynakları olduğundan, enerji problem olmasa da,

metalik yüzeylerin çokluğu yansıma miktarını arttırarak ani değişen bağlantı kalitelere sebebiyet verir. Hareket vektörü genellikle tek eksen ve tek yöndedir. Hareketlilik yavaş yavaş oluşup, yavaş yavaş sonlanır [59].

- **Dinamik Ortam Hareketlilik Modeli**

Bu modelde rüzgar, su hareketliliği, toprak hareketliliği gibi doğal hareketlerden meydana gelen değişimler tanımlanmaktadır. Çok sık gerçekleşmese de gerçekleştiği zaman ağın büyük bir bölümünü etkileyebilir, kullanılmaz hale getirebilir.

4.1.3 Birimler Arası Geçiş Mekanizması

Hareketlilikten dolayı kablosuz ağlarda ağ içinde geçiş yapmak günümüzde kullanılan hücrel ağlar ve kablosuz lokal ağlar gibi teknolojilerde çokça araştırılmış ve uygun çözümler bulunmuştur. Günümüzde 2 adet geçiş tipi bulunmaktadır:

- Yatay Geçiş Kablosuz birimin hareketliliğinden veya sinyal kalitesinden dolayı bağlı olduğu erişim noktasından aynı ağ yapısı içinde bulunan bir başka erişim noktasına geçmesine denir. Örneğin hücrel ağlar veya Wi-Fi bu tarz yapıya örnektir.
- Dikey Geçiş Kablosuz birimin hareketliliğinden veya sinyal kalitesinden dolayı bağlı olduğu erişim noktasından farklı ağ yapısı içinde bulunan bir başka erişim noktasına geçmesine denir. Bu tarz yapılar günümüzde yavaş yavaş kullanıma geçen heterojen yapılara örnektir. Örneğin veri trafiğinin hücrel ağ ile başlayıp Wi-Fi ile devam ettiği uygulamalarda dikey geçiş kullanılır.

Geçiş mekanizmaları hücrel yapılarda [60]ve lokal kablosuz ağlarda [61] detaylı bir şekilde incelenmiş olsa da, düşük güçlü KSA'lar aynı ilgiyi daha görememiştir. KSA'lar için geçiş algoritması geliştirmeden önce gelişimi belli bir olgunluk seviyesine gelmiş hazırları kısaca incelemek gereklidir.

- **Hücrel Ağlarda Geçiş**

Hücrel yapılarda geçiş kararı çok güçlü bir merkezi sistem olan baz istasyonu tarafından yönetilir. Hücrel ağlarda en büyük problem devam eden bir çağrı sırasında mobil cihazın baz istasyonu değiştirmesidir. Geçiş yapılan baz istasyonunda boş bir

kanal yok ise çağrı düşecektir. Bundan dolayı baz istasyonları önceden rezerve edilmiş boş kanallar bulundurmaktadır. Hücresel yapılarda geçiş çok kanallı yapıları ile, aynı anda iki yönlü iletişim kurabilme yeteneği ve CDMA erişimi ile KSA'ların kurabileceği ağlardan çok daha sofistike yapıdadır. Bu yüzden hücresel yapılardaki algoritmalar KSA'lara tam anlamıyla uygun değildir.

- **Lokal Kablosuz Ağlarda Geçiş**

Hücresel yapıların aksine lokal kablosuz ağlarda geçiş yapacak birim ağ hakkında detaylı bilgiye sahip değildir. Ayrıca hücresel sistemlerde sinyal kalitesi sürekli izlenirken, kablosuz ağlarda sinyal kalitesi belli bir seviyenin altına düştükten sonra ortam dinlemesi yapmaktadır. Kablosuz ağlarda geçiş 2 evreden oluşur. Birinci evre büyük bir gecikmeye sebebiyet veren ortamdaki diğer erişim noktalarının bulunduğu "keşif" evresidir. İkinci evre ise bulunan erişim noktası ile bağlantının kurulmasını sağlayan "kimlik doğrulama" evresidir. Kablosuz lokal ağları KSA'lardan ayıran en büyük özellikler geniş bant genişliği, enerjinin büyük bir problem olmaması ve güçlü ağ işlemcileridir. Bundan dolayı burada kullanılan yapılar ve uzun geçiş süreleri KSA'lar için tam anlamıyla uygun olmamaktadır.

4.2 Kablosuz Sensör Ağları İçin Geliştirilmiş Geçiş Algoritmaları

RPL protokolü genel olarak durağan ağ yapıları için geliştirilmiştir [62]. RPL protokolü hareketli birimlerin DODAG'a katılması konusunda herhangi bir kısıtlamaya sahip değildir. Ancak birim ağ içerisinde hareket ederken büyük olasılıkla o an bağlı olduğu ebeveyn birimden kopacaktır. Birimin kendini ağa tekrardan bağlayabilmesi için ortamdaki potansiyel ebeveyn birimleri öğrenmesi gerekmektedir. Daha önceden de bahsedildiği gibi birim DIS mesajı yollayarak çevresindeki birimlerden bilgi toplayabilmektedir. Ancak RPL protokolünde DIS mesajının ebeveyn değiştirmek için nerede ve ne zaman kullanılacağı belirtilmemiştir. Bu kısımlar açık kaynak kodlu olan RPL protokolü için geliştirilmeyi bekleyen noktalardır. Bu eksiklik üzerine halihazırda bir kaç çalışma yapılmış [63][64] ancak bunların arasından iki tanesi [65][66] hatırı sayılır bir gelişme yaratmıştır.

Geçiş algoritmaları genellikle 3 evreden oluşur. Bunlar birimin içinde bulunduğu durumu anlamak için ölçümler aldığı ölçüm evresi, geçiş algoritmasının alınan ölçümler üzerindeki kararı verdiği geçiş kararı evresi ve son olarak geçişin kendisidir. Kablosuz ağlarda birimler

arası geçiş ağıdaki hizmet kalitesini ciddi oranda etkileyebilir. Dolayısıyla, geçiş algoritması tasarlanırken göz önüne alınması gereken bir takım özellikler ve gereksinimler vardır.

- **Geçiş Gecikmesi:** Geçiş algoritmasının en önemli parametrelerindedir. Geçiş sırasında geçiş gecikmesi olabildiğince az olmalıdır. Geçiş yapan birimin ağa bağlı olduğu süre en aza indirilebilirse başarılı paket gönderim oranı artacaktır.
- **Sekme Etkisi:** Geçiş yapan birim geçişini yaptıktan sonra durağan bir yapıda olmalıdır. Geçiş gerçekleşirdikten çok kısa bir süre sonra gereksiz başka bir geçiş yapmamalıdır. Yapılan her değişim fazladan kontrol paketleri gönderilmesine, gereksiz zaman ve enerji kaybı oluşuma sebebiyet verecektir.
- **Başarılı Paket Gönderim Oranı:** Geçiş algoritması uygulamanın gerektirdiği oranda başarılı paket gönderim oranına sahip olmalıdır. İstenilen değerin altında kalınırsa aynı mesajın tekrar tekrar gönderilmesi gerekmektedir. Bu gönderimler de fazladan enerji ve zaman kaybı yaratacaktır.
- **Ağ Kontrol Yüğü:** Başarılı bir geçiş algoritması geçiş için gerekli olan kontrol mesajlaşmasını en aza indirmelidir. Gönderilen her paket fazladan enerji harcayacaktır.
- **Katmanlı Yapı:** Tasarlanan geçiş algoritması katmanlı yapıya uymalıdır. Algoritma mümkünse bir katmanda bitirilmelidir ve diğer katmanlardan bağımsız çalışabilmelidir. Katmanlı yapının bozulması hem algoritmanın kontrol edilebilirliğini düşürecek, hem de katmanlar arasındaki izolasyonu bozacaktır. Geçiş algoritmasının performansı, algoritmanın yerleştirildiği katmanın dışındaki katmanlar tarafından belirlenmemelidir.
- **Uygulanabilirlik:** RPL protokolü genelde 802.15.4 fiziksel katmanında kullanılsa da, sadece o fiziksel katmana özgü bir biçimde tasarlanmamıştır. RPL 802.15.4 fiziksel katmanının en temel mesaj alma ve mesaj yollama fonksiyonlarını kullanmaktadır. Bu sayede istenildiği takdirde başka fiziksel katman ile de kullanılabilir. RPL protokolüne uyumlu bir geçiş algoritması geliştirilirken, algoritmanın RPL ile aynı uyumluluk seviyesine sahip olması gerekmektedir.
- **Uyumluluk:** Tasarlanan algoritmayı taşıyan birim veya birimler, üzerlerinde herhangi bir değişiklik yapılmamış birimler ile aynı ortamda çalışabilmelilerdir.

4.2.1 MoMoRo Algoritması

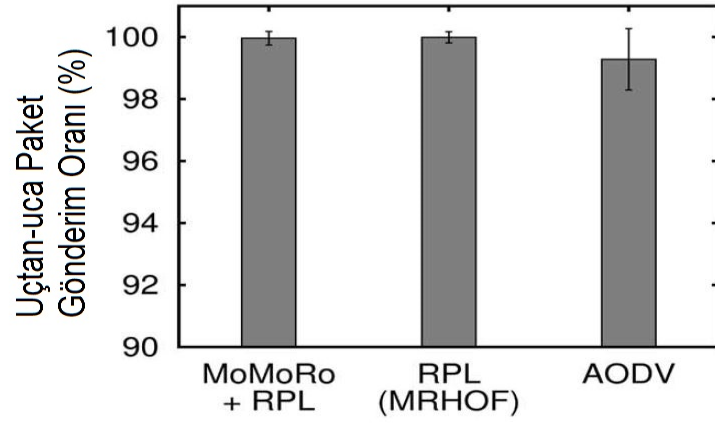
MoMoRo [65] algoritması kurulmuş bir RPL ağı içerisindeki birimlerin hareketliliği sağlamak için 2 ana kısımdan oluşur. Birinci kısım ebeveyn birim ile bağlantının koptuğunu anlayan algoritmadır. Bunu sağlamak için algoritma yönlendirme katmanında gözlemci olarak çalışmaktadır. Ebeveyn birime gönderilen uygulama katmanı mesajları ve bunun karşılığında alınan tasdik mesajlarının oranına bakılır. Oran belli bir değerin altına düşer ise algoritma bağlantıda bir problem olduğuna karar verir. Bu kararın üzerine MoMoro algoritması belirli bir süre bekleyerek ebeveyn birimine son bir defa tekrardan ulaşmaya çalışır. Bu beklemenin amacı bağlantıda anlık bir bağlantı kalitesi değişimi olup olmadığına kanaat getirmektir. Son denemeye de belirli bir süre içerisinde cevap alınmaz ise yeni bir ebeveyn birim aranmaya başlanır. İkinci kısım ise yeni bir bağlantı kurulumudur. MoMoRo algoritması var olan bağlantının koptuğunu belirler ise yeni bağlantı noktası aramak için ortama herkesin duyabileceği şekilde bağlanma isteği mesajları yayar. Mesajı duyan olası ebeveynler hareketli birime cevap mesajı yollarlar. Hareketli birim yollanan bu mesajlardan bağlantı kalitesini tayin etmek için 3 parametreyi inceler. Bunlar;

- Ortama yayılan mesaj sayısı ile olası ebeveynden gelen cevap sayısının oranına bakılır.
- Olası ebeveynden gelen cevap mesajlarının ortalama sinyal kuvvetine bakılır.
- Olası ebeveynden gelen cevap mesajlarının sinyal kuvvetlerinin varyansına bakılır.

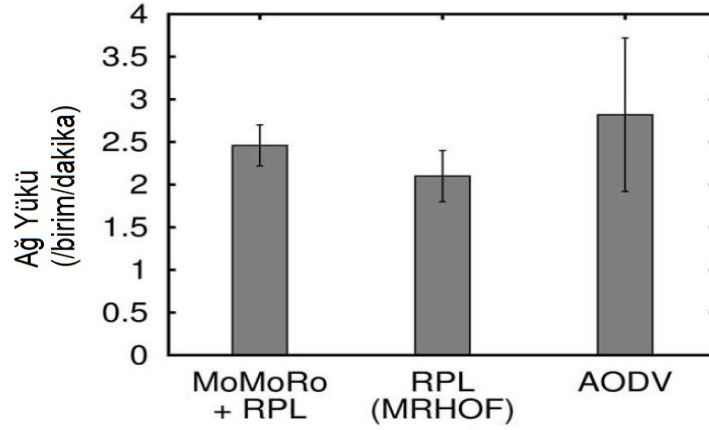
İncelenen bu parametreler doğrultusunda MoMoRo algoritması ortamdaki bağlantı kalitesi en iyi ebeveyn birimi tayin ederek ona bağlanır. Ancak bu parametrelerin ne olacağının bulunması karmaşık, kurulacak ağ hakkında fazlaca bilgi ve tecrübe gerektirmektedir [65]. MoMoRo geliştiricisi tarafından uygun görülen parametreler eşliğinde bazı performans testleri yapılmıştır. Yapılan testler karşılaştırma için üzerinde herhangi bir değişiklik yapılmamış RPL protokolü, AODV yönlendirme protokolü [67] ve geliştirilen MoMoRo için tekrarlanmıştır. Testler yapılırken veri toplama topolojisi seçilmiş, her bir birim saniyede 1 kere kök birime mesaj yollayacak şekilde programlanmıştır. Test sonuçları olarak ağ yükü ve mesajların iletim oranı incelenmiştir.

4.2.1.1 Durağan Ortam Testi

Bu testte karşılaştırılan 3 protokolün hiç bir hareketliliğin olmadığı, radyo çevrim katmanının kullanılmadığı durağan bir ağ yapısındaki performansları incelenmiştir. Şekil 4.1a ile ağ içerisindeki bir birim ile kök birim arasındaki veri iletişimde durağan ağlarda 3 protokolünde başarılı sonuç verdiği görülmektedir. Şekil 4.1b ile ise MoMoRo algoritmasının standart RPL protokolünden biraz daha fazla ağ yüküne sahip olduğu, AODV'nin ise çok değişken ve en fazla ağ yüküne sahip olduğu gösterilmiştir.



(a) Protokoller arası başarılı paket gönderim oranı karşılaştırması [65]

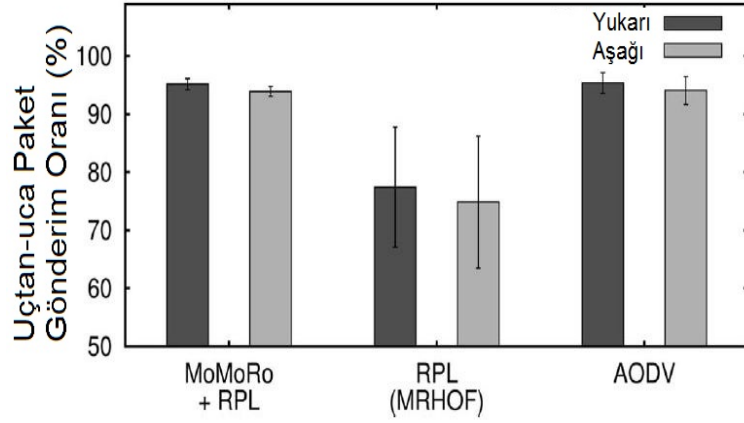


(b) Protokoller arası hareketli birim için ağ yükü karşılaştırması [65]

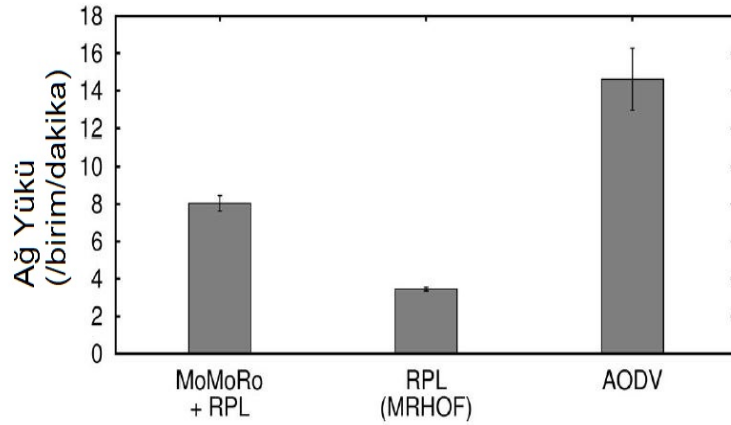
Şekil 4.1: Durağan ağ testinde karşılaştırmalar

4.2.1.2 Hareketli Ortam Testi

Bu testte ise ağ içerisinde 1 adet hareketli birim bulunmaktadır ancak hareketli birimin hızından bahsedilmemiştir. Hareketli birim ile kök birim arasındaki veri hem aşağı yönlü hem de yukarı yönlü olarak incelenmiştir. Şekil 4.2a ile ağ içerisindeki hareketliliğin standart RPL protokolünün paket gönderim oranını ciddi biçimde düşürdüğü, ancak MoMoRo ve AODV'nin hareketliliği başarılı bir şekilde yönetebildiği gösterilmiştir. Şekil 4.2b ile ise RPL'nin hareketliliği algılayamamasından dolayı kontrol mesajlaşmalarını arttırmadığı, MoMoRo'nun AODV'den daha az kontrol mesajlaşması ile hareketliliği sağlayabildiği görülmektedir.



(a) Protokoller arası başarılı paket gönderim oranı karşılaştırması [65]



(b) Protokoller arası hareketli birim için ağ yükü karşılaştırması [65]

Şekil 4.2: Hareketli ağ testinde karşılaştırmalar

MoMoRo algoritması aktif ebeveyn ile arasındaki bağlantının kopmasını, hedefine ulaşama-

yan uygulama katmanı mesajları ile anlamaktadır. Bundan dolayı artan hareketlilik akabinde gittikçe azalan paket gönderim oranı yaratacaktır. Ayrıca bir diğer sorun geçiş algoritmasının performansı, uygulama katmanındaki mesaj gönderim oranı ile belirlenmektedir. Şöyle ki, uygulama katmanındaki mesaj gönderim sıklığı arttıkça, MoMoRo bağlantı kopmasını daha hızlı anlayıp geçiş yapabilmektedir. Ancak uygulama katmanı çok az sıklıkla ya da herhangi bir mesaj yollamadığı takdirde MoMoRo hareketliliğe karşı tepki süresini uzatacak veya hiç bir önlem alamayacaktır. Hareketli birim hareket edip etmediğini anlamayacaktır. Hareketlilik ancak yukarı yönlü bir mesaj gönderilip mesaja cevap alınamadığı takdirde anlaşılacak ve yeni bir ebeveyn seçilecektir. Ancak hareketli birimin uygulama katmanı yukarı yönlü mesaj göndermez iken kök birim o birime aşağı yönlü bir mesaj gönderirse mesaj hedefe ulaşmayacaktır.

4.2.1.3 MoMoRo Algoritması Performans Sonuçları

MoMoRo periyodik mesajlaşmanın olduğu uygulamalarda mesajlaşmanın sıklığına göre duran veya az hareketli ağlar için tatmin edici bir performans gösterirken, hareketlilik arttıkça mesaj gönderim oranları azalarak uygulamanın gerekliliklerini karşılamaz hale gelmektedir. Geçiş hızı belirtilmemiş olsa da algoritmanın yapısına göre bağlantının koptuğunun anlaşılması için en az 2 uygulama mesajı kaybı, yeni bir ebeveyn bağlanmak için ise en az 1 adet bağlantı isteği mesajı yollanması ve uzunluğu belirtilmemiş bir bekleme süresi geçmesi gerekmektedir.

4.2.2 Smart-Hop Algoritması

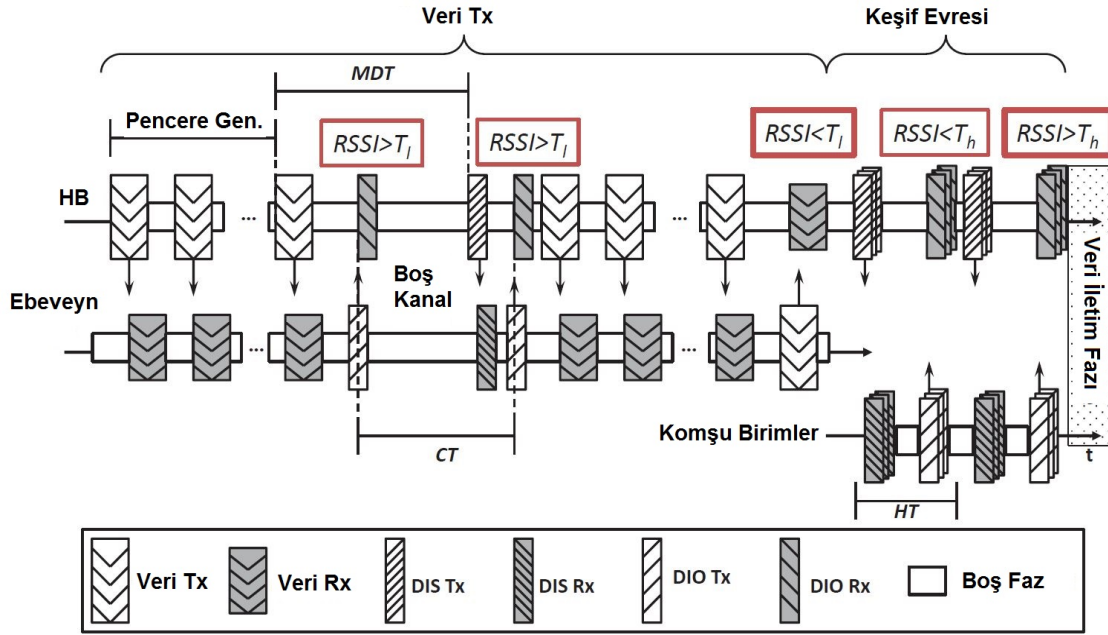
Smart-Hop [66] algoritması kablosuz sensör ağlarında hareketliliği yönetmek için geliştirilmiş yöntemlerden birisidir. Algoritma RPL yönlendirme protokolü için geliştirilmediğinden mRPL [68] adı altında RPL için uygun hale getirilmiştir. Smart-Hop algoritması 2 ana evreden oluşmaktadır. Bunlar *veri iletim* evresi ve *keşif* evreleridir.

- **Veri İletimi Evresi:** Bu evrede hareketli birim ile bağlı olduğu ebeveyn birim arasında kurulu bir bağlantı mevcuttur ve hareketli birim ebeveyn birime periyodik olarak uygulama katmanı mesajı yollamaktadır. Ebeveyn birim belirli sayıda mesaj aldıktan sonra, alınan mesajların ortalama RSSI bilgisini hareketli birime DIO kontrol mesajı

içerisinde gömülü olarak yollar. Hareketli birim bağlantı kalitesini bu sayede tayin edebilir. Ancak ebeveyn birim herhangi bir uygulama katmanı mesajı almaz ise cevap mesajı da yollamayacağı için bağlantı kalitesi kestirilemez. Hareketli birim, aldığı cevap mesajlarındaki ortalama RSSI bilgisini belirlenmiş bir değer ile karşılaştırır. Eğer RSSI değeri belirli bir değerden düşük ise 2. evre olan *keşif* evresine geçilir.

- **Keşif Evresi:** Bu evrede hareketli birim yeni ebeveynler bulmak için ortama belirli sayıda DIS mesajı yayar ve ebeveyn birimlerden cevap bekler. DIS mesajını alan ebeveyn birimler ortalama RSSI bilgisini DIO mesajları içerisine gömüp hareketli birime gönderirler. Ortalama RSSI değeri belirlenen bir değerden yüksek olan ebeveyn birim yeni ebeveyn olarak seçilerek yeniden *veri iletimi* evresine geçilir.

Evreler ve ilgili mesajlaşmalar Şekil 4.3 ile gösterilmiştir.



Şekil 4.3: Smart-Hop algoritması evreleri [68]

Smart-Hop algoritmasının geçiş özelliklerini belirleyen bazı parametreler vardır. Parametrelerin değerlerine göre geçiş hızları ve ağ yükü ayarlanabilmektedir.

- **RSSI Limiti:** Veri iletimi evresindeki hareketli birim, bağlı olduğu ebeveyn birimden periyodik olarak RSSI bilgisi almaktadır. Aldığı RSSI bilgisini RSSI limiti ile karşılaştırır. Alınan RSSI değeri limit değerinin altında kalır ise keşif evresine geçilir. Bu

değer kritik rol oynamaktadır. Limit değer çok aşağıda seçilir ise güvenli sayılabilecek bir bağlantı güvensiz diye nitelendirilip geçiş yapılmasını tetikleyecektir. Limitin çok yukarıda seçilmesi ise bağlantının çok güvensizleşmesinde bile bir geçiş tetiklememesine neden olur. Bundan dolayı uygulamadan uygulamaya değişen, uygun bir değer bulunması gerekmektedir.

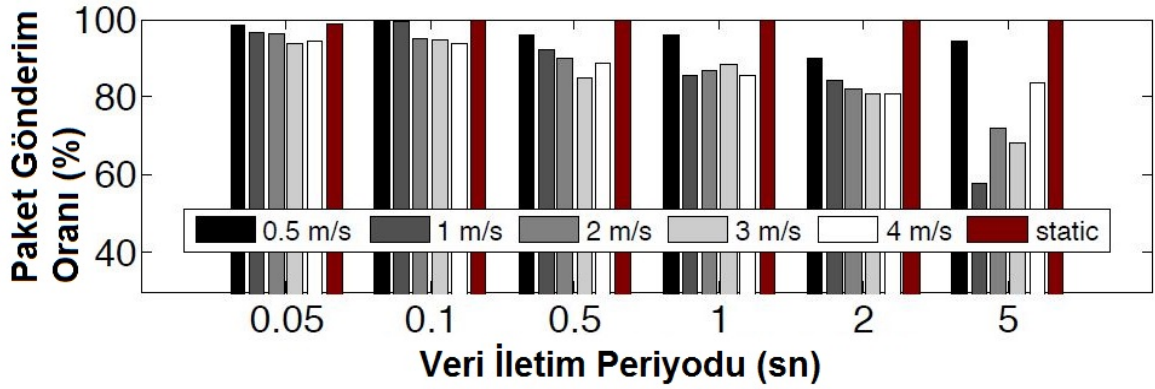
- **Pencere Genişliği:** Ortalama RSSI hesaplanması için belirlenen zaman aralığı içerisinde alınması gereken uygulama katmanı mesajıdır. Pencere genişliği dar tutulursa bağlantı kalitesi hakkında daha çok cevap mesajı oluşturulacağından ağ yükü de artmaktadır. Ayrıca anlık değişimlerde ebeveyn değiştirmeyi tetikleyeceğinden gereksiz enerji harcanacaktır. Geniş pencere genişliği ise bağlantı kalitesi hakkında ortalama bir bilgi vererek sistemin hareketliliğe karşı hassasiyetini düşürecektir.
- **Histerezis Marjı:** *Keşif* evresindeki hareketli birim çevresindeki ebeveynlerden ortalama RSSI bilgisi talep etmektedir. Alınan değerler RSSI limitinden histerezis payı kadar yüksek ise işleme alınır. Histerezis payı yüksek seçilir ise hareketli birim çok seçici olacağından güvenli olan bağlantıları dahi kabul etmeyecektir. Eğer bu değer çok küçük seçilir ise hareketli birim çok sık ebeveyn değiştirerek pinpon etkisi yaratacaktır. Sonuç olarak fazla enerji kaybedecek ve değişimler sırasında başarılı mesaj gönderme oranı azalacaktır.
- **Bağlantı Sayacı:** Bu sayaç en uzun *Trickle* zamanlayıcısı periyoduna kurularak çalışmaya başlatılır. Bu süre zarfında aktif iletişimde olunan ebeveyn birimden cevap mesajı gelirse sayaç sıfırlanır. Aksi takdirde ebeveyn ile bağlantının koptuğuna kanaat getirilip *keşif* evresine geçilir.
- **Hareketlilik Algulama Zamanlayıcısı:** Smart-Hop algoritması bağlantı kalitesini hareketli birimin ebeveyn birime gönderdiği uygulama katmanı mesajlarına karşılık ebeveyn birimin gönderdiği DIO mesajları aracılığı ile öğrenir. Ancak gönderilecek bir uygulama katmanı mesajı yok ise bağlantı kalitesi hakkında bilgi edinilemez. Bundan dolayı belirli bir periyot ile hareketli birim ebeveyn birimine DIS mesajları yollayarak ondan içerisinde ortalama RSSI bilgisini içeren DIO mesajı talep eder. Ancak bu sistemde periyot çok kısa seçilir ise ağ yükü artacak, az seçilir ise de hareketliliğe karşı pasif kalınacaktır.

Smart-Hop algoritmasının performansını değerlendirmek için çeşitli senaryolar ile benzetim

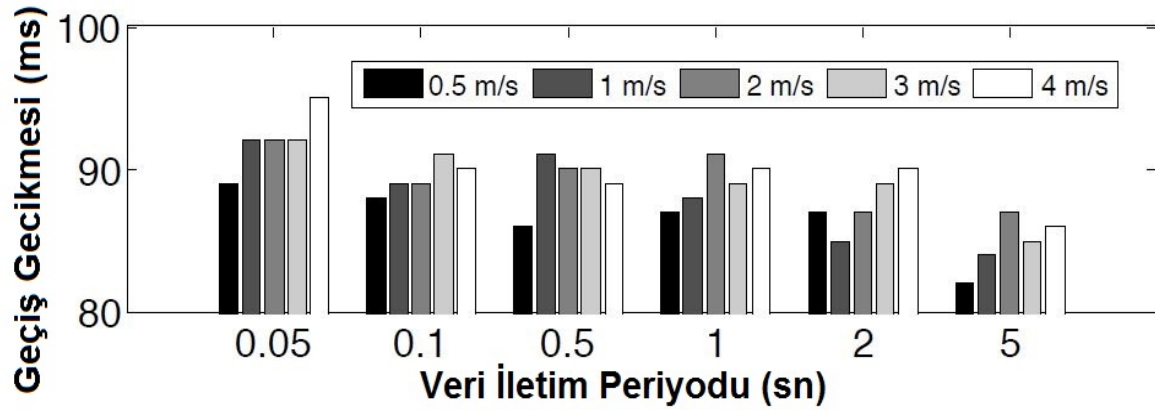
ve saha testleri yapılmıştır.

4.2.2.1 Hareketliliğin Hızı ve Veri Gönderme Periyodu Testi

Yapılan bu testte uygulama katmanının gönderdiği mesaj sıklığı ile hareketliliğin hızının, performansa etkisi incelenmiştir. Test süresince radyo çevrim katmanı kapalı tutulmuştur. Yani radyo katmanı sürekli olarak dinleme konumundadır. Performans kriterleri doğru paket gönderim oranı ve geçiş gecikmesi olarak seçilmiştir. Şekil 4.4a ile değişen hızlar ve paket gönderim oranının ilişkisi incelenmiştir. Şekil 4.4b ile aynı test ortamında geçiş gecikmesi gösterilmiştir.



(a) Smart-Hop başarılı paket gönderim oranı karşılaştırması [68]



(b) Smart-Hop hareketli birim için geçiş gecikmesi karşılaştırması [68]

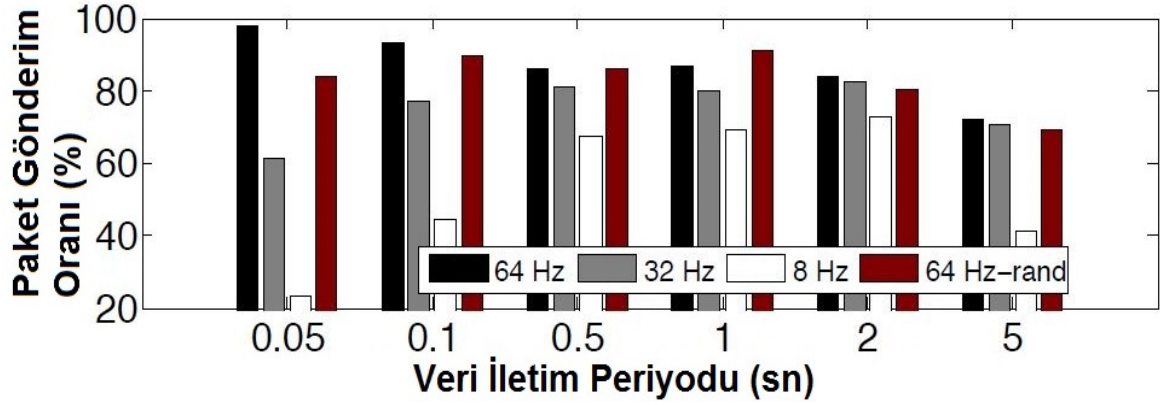
Şekil 4.4: Smart-Hop performans testi 1

Bu testin sonucu olarak sıklığı azalan uygulama katmanı mesajları geçiş performansını kötü yönde etkilemektedir. Hız ekseninde test sonuçları değerlendirildiğinde hızlı uygulama kat-

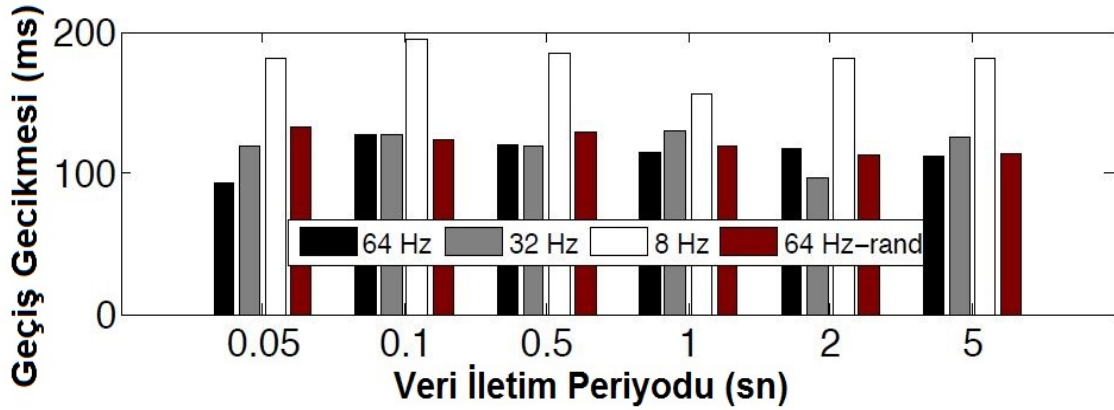
manı mesajlarında hızın büyük bir değişiklik getirmediği, uygulama katmanı mesajları azaldıkça hızın paket gönderim oranını önemli ölçüde etkilediği gözlemlenmektedir. Geçiş gecikmesi ise test süresi boyunca neredeyse sabit kalmıştır.

4.2.2.2 Radyo Çevrim Katmanı ve Veri Gönderme Periyodu Testi

Bu testte ise hareketli birimin hızı 2 m/s olacak şekilde sabit tutulup, radyo çevrim katmanının periyodu ve uygulama katmanının gönderdiği veri sıklığı değiştirilerek Smart-Hop algoritmasının performansı incelenmiştir. Şekil 4.5a ile değişen radyo çevrim periyodu ve paket gönderim oranlarının ilişkisi incelenmiştir. Şekil 4.5b ile aynı test ortamında geçiş gecikmesi gösterilmiştir.



(a) Smart-Hop başarılı paket gönderim oranı karşılaştırması [68]



(b) Smart-Hop hareketli birim için geçiş gecikmesi karşılaştırması [68]

Şekil 4.5: Smart-Hop performans testi 2

İkinci testin sonucunda düşük sıklıkta uygulama katmanı mesajı ve düşük radyo çevrim

periyodunda paket gönderim oranı %25 civarlarına kadar düşebilmektedir. Radyo çevrimi sıklaştıkça ve gönderilen mesaj sayısı arttıkça paket gönderim oranı tatminkar seviyelere çıkabilmektedir. Geçiş gecikmesi 8 Hz radyo çevrim sıklığında 200 ms civarlarına kadar çıksa da bu değer bile birçok uygulamanın isterlerini karşılayabilmektedir.

4.2.2.3 Smart-Hop Algoritması Performans Sonuçları

Smart-Hop algoritması bağlantı kalitesi hakkındaki bilgileri, gönderilen uygulama katmanı mesajları ile öğrenmektedir. Bu yüzden geçiş algoritmasının performansı, direkt olarak uygulama katmanına bağlıdır. Bu da Smart-Hop algoritmasının MoMoRo algoritması ile aynı sorunlara sahip olacağı, azalan uygulama katmanı mesajlarının geçiş algoritmasını hareketliliğe karşı pasif hale getireceği anlamına gelmektedir. Ancak ebeveyn geçişini paket kaybı ile anlamadığı için aynı koşullarda Smart-Hop algoritması MoMoRo algoritmasından daha yüksek paket gönderim oranına sahip olacaktır. Smart-Hop algoritmasının performansı azalan radyo çevrim sıklığı ile ciddi biçimde düşmektedir. Bu yüzden 1 Hz gibi çevrim sıklığına sahip uygulamalar için neredeyse kullanılamaz hale gelmektedir.

Smart-Hop algoritmasının en büyük problemlerinden birisi de bağlantının koptuğuna belirlenmiş bir RSSI değeri ile karar vermesidir. Ancak belirlenen değer altında da ortam gürültüsüne bağlı olarak iletişim kurmak mümkün olduğundan zaman zaman gereksiz ebeveyn değişimi yapılmaktadır. Smart-Hop algoritması uygulama katmanındaki veri iletişiminin yüksek olduğu, radyo çevriminin kapalı ya da yüksek sıklıkla çalıştığı ve hareketliliğin yüksek olduğu uygulamalar için uygun bir algoritmadır.

5. GELİŞTİRİLEN PHOENIX GEÇİŞ ALGORİTMASI

KSA'lar için çeşitli hareketlilik çalışmaları yapılmış ve Smart-Hop gibi hızlı geçiş algoritmaları yapılmış olsa da hala geliştirilmeye açık birçok nokta vardır. Katmanlı yapıya uygunluk, geçiş performansının diğer katmanlar tarafından etkilenmemesi, farklı uygulamalara uyumluluk, farklı radyo yongalarının bir arada çalışabilmesini sağlamak, her durumda yüksek paket gönderim oranına sahip olmak, düşük işlem gücü ve yazılım boyutuna sahip olmak, düşük enerji harcamak, düşük ağ yüküne sahip olmak ve birçok fiziksel katman ile çalışabilmek gibi birçok isteri kapsayan bir geçiş algoritması geliştirilmelidir. Bu eksikliği olabildiğince kapatmak için tez kapsamında *Phoenix* adı altında yönlendirme katmanında çalışan geliştirilmeye çok açık bir geçiş algoritması geliştirilmiştir. Algoritmanın tasarımı yukarıdaki kıstasları kapsayacak yapıda şekillenmiştir. Algoritma iki ana mekanizmadan ve bir yardımcı mekanizmadan oluşmaktadır.

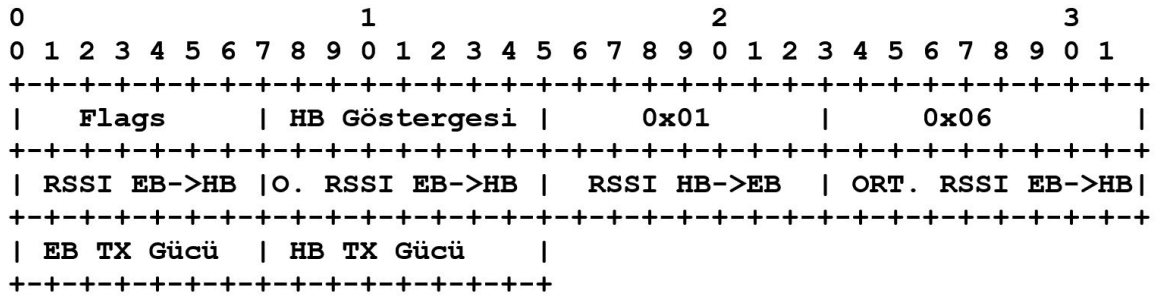
- Bağlantı kalitesinin ölçüldüğü ve hareketli birim ile ebeveyn birimin haberleştiği *İletişim Kontrolü* mekanizması
- Kontrol mekanizmasından gelen bilgilerin kullanıldığı geçiş kararının verilip uygulandığı *Geçiş Kontrolü* mekanizması
- Kontrol mekanizmasından gelen bilgilerin değerlendirilip güç tüketimini düşürmeyi hedefleyen yardımcı *Güç Kontrolü* mekanizması

5.1 İletişim Kontrolü Mekanizması

İletişim kontrolü mekanizması Phoenix geçiş algoritmasının temelini oluşturur. Bu mekanizma sayesinde hareketli birim ve ebeveyn birim birbirleri ile haberleşebilir, bağlantı parametrelerini paylaşabilir ve birbirlerine komutlar gönderebilirler. Phoenix algoritması geliştirilirken birimleri haberleştirmek için yeni bir kontrol mesajı yaratılmamış, var olan DIS ve DAO mesajları RPL protokolünün izin verdiği ölçüde [8] değiştirilmiştir. Böylelikle bir mesaj ile hem RPL için gerekli kontrol mesajları iletilecek, hem de geçiş için gerekli olan kontrol mesajlaşmaları gerçekleştirilebilecektir.

5.1.1 Phoenix DIS Mesajı

DIS mesajı RPL yönlendirme protokolünde iletilebilecek en kısa kontrol mesajıdır ve gönderimi sırasında en az enerji harcanmaktadır. Bu yüzden hareketli birim ve ebeveyn birimin veri alışverişi yapabilmesi için DIS mesajında bazı değişiklikler yapılmıştır. Standart RPL DIS mesajında bulunan ve gelecek uygulamalar için rezerve edilmiş bölge hareketli birim göstergesi olarak kullanılmıştır. Buna ek olarak RPL standardının izin verdiği şekilde opsiyon bölgelerine ebeveyn birim ile hareketli birimin haberleşmesini sağlayan ek veriler eklenmiştir. Şekil 5.1 ile geliştirilen yeni DIS mesajı görülebilir.



Şekil 5.1: Phoenix DIS mesajı

DIS mesajının taşıdığı bilgiler şunlardır;

- **Flags:** Şu an RPL standardı tarafından kullanılmayan sekmedir.
- **HB Göstergesi:** DIS mesajının Phoenix algoritması kullanan bir birim tarafından gönderildiğini gösteren bölgedir. Bu bölge standart RPL protokolünde 0, Phoenix algoritmasında kontrol mesajı amacı ile kullanılıyor ise (**Mod 1**) 1, alternatif ebeveyn den hızlı bir şekilde DIO istemek (**Mod 2**) için 2 değerine sahiptir. Standart RPL protokolü kullanan bir birim bu bölgeye bakmadığı için [8] değiştirilmiş DIS mesajı aldığı zaman fazladan eklenmiş verileri işleme almamaktadır.
- **0x01:** RPL standardına göre opsiyon bölgelerinin kullanıldığını gösteren göstergedir.
- **0x06:** Kendinden sonra 6 adet opsiyon baytının geleceğini gösteren değerdir.
- **RSSI EB->HB:** Hareketli birime ebeveyn birim tarafından gönderilen Phoenix DIS mesajının RSSI değerini gösteren bölgedir.

- **O. RSSI EB->HB:** Hareketli birime ebeveyn birim tarafından gönderilen Phoenix DIS mesajlarının ortalama RSSI değerini gösteren bölgedir.
- **RSSI HB->EB:** Ebeveyn birime hareketli birim tarafından gönderilen Phoenix DIS mesajının RSSI değerini gösteren bölgedir.
- **O. RSSI HB->EB:** Ebeveyn birime hareketli birim tarafından gönderilen Phoenix DIS mesajlarının ortalama RSSI değerini gösteren bölgedir.
- **EB TX Gücü:** Ebeveyn birimin radyo çıkış gücünü gösteren bölgedir.
- **HB TX Gücü:** Hareketli birimin radyo çıkış gücünü gösteren bölgedir.

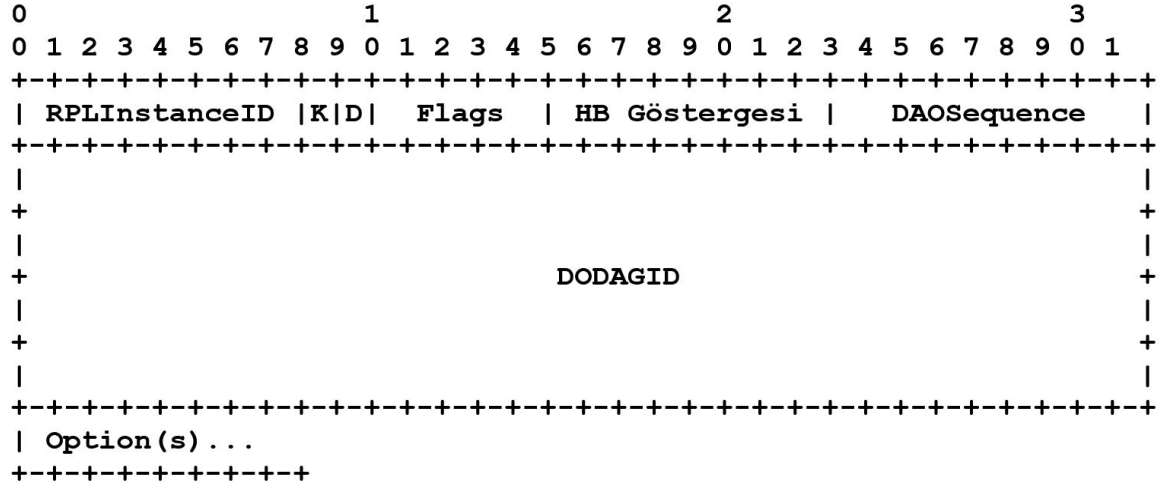
Phoenix DIS mesajının bir özelliği de alıcı tarafında *Trickle Zamanlayıcısını* sıfırlamamasıdır. Standart RPL protokolünde DIS mesajı alan birim *Trickle Zamanlayıcısını* sıfırlayarak ortama yüksek sayıda DIO mesajı gönderir. Bu eklenti yapılmıyaydı Phoenix algoritmasında kontrol mesajlaşması olarak DIS mesajını kullanıldığından ve DIS mesajını duyan her bir birim de *Trickle Zamanlayıcısını* sıfırlayacağından ağ bazında yüksek oranda enerji tüketilmiş olacaktı.

5.1.2 Phoenix DAO Mesajı

Phoenix algoritmasında DAO mesajı içeriğinde büyük bir değişiklik yapılmamıştır. Sadece, Şekil 5.2 ile görülebileceği gibi rezerve bölge yerine hareketli birim göstergesi eklenmiştir. Böylelikle hareketli birim bağlandığı ebeveyn birime DAO mesajı yolladığında, ebeveyn birim kendisine hareketli bir birimin bağlandığını anlayabilmektedir. Orijinal RPL protokolüne sahip bir birime değiştirilmiş DAO mesajı yollandığında ise DIS mesajında olduğu gibi bu bölgeye bakılmayacağından hareketlilik ile ilgili herhangi bir işlem yapılmayacaktır.

5.1.3 RSSI Ölçümü

Geçiş algoritması tasarlanırken en önemli sorunlardan birisi bağlantı kalitesini doğru bir biçimde ölçebilmektir. Phoenix algoritması bağlantı kalitesini hızlı ve doğru bir biçimde ölçmek için RSSI değerini kullanılmaktadır. RSSI değeri radyo yongası *Ortam Erişim Kontrolü* katmanından, alınan her bir mesaj için ölçülüp yönlendirme katmanına iletilmektedir.



Şekil 5.2: Phoenix DAO mesajı

RSSI değerinin hassaslığı günümüz radyo yongalarında ortalama ± 4 dB olup dinamik alanı ise almaç hassaslığını kapsayacak şekilde tasarlanmıştır. Örneğin CC2538 radyo yongası -88 dB alıcı hassasiyetine sahip iken -100 dB RSSI ölçümü yapabilmektedir. Ölçülen RSSI değeri, kullanılan radyo işlemcisinin almaç hassasiyeti ile karşılaştırıldığında bağlantı kalitesi hızlı bir biçimde kestirilebilmektedir.

RSSI ölçümü, doğası gereği değerler mesajı gönderen tarafta değil, mesajı alan tarafta ölçülmektedir. Bu yüzden mesaj göndericisi kendi gönderdiği mesajın bağlantı kalitesini tayin edememektedir. Phoenix algoritması bu sorunun üstesinden gelmek için RSSI ölçümünü mesaj göndericisine geri gönderen, birimlerin bağlantı kalitesini anlık ve ortalama cinsinden ve iki yönlü izleyebileceği bir çalışma şeması kullanmaktadır. Bu sayede birimler bağlantının sadece asimetric olup olmadığını anlamakla kalmaz, *asimetrikliğin derecesini* de tespit etmiş olurlar.

KSA birimlerinin işlem gücü kısıtlamaları düşünüldüğünde algoritmada kullanılan bütün işlemlerin olabildiğince az kaynak kullanması gerekmektedir. Ortalama RSSI değeri hesaplanırken de bu düşünce göz önüne alınmıştır. Ortalama RSSI değeri hesaplanırken, birimlerin haberleşmeye başladıkları andan son ana kadar olan ortalama almayı almak, ortalama değeri zaman içerisinde değişen KSA uygulamaları için uygun bir yöntem değildir. Onun yerine hareketli ortalama metodlarını kullanmak gerekmektedir. Bu sayede bağlantı kalitesinin zaman içerisindeki davranışı daha detaylı incelenebilir. Phoenix algoritmasında hareketli ortalama yöntemi yüksek geçici bellek kullanımına ihtiyaç duyduğundan onun yerine ağırlıklı top-

lama yöntemi kullanılmıştır. Eşitlik 5.1 ile yaklaşık son 5 değerın ortalaması alınmaktadır. Son 5 değerin ortalamasını bulmak için sadece 1 eleman geçici bellekte tutularak iki adet bir bit sağa kaydırma ve bir adet toplama işlemi yeterli olmaktadır. 5 değerin özel bir anlamı olmayıp Eşitlik 5.1 sonucunda yaklaşık olarak ortaya çıkmaktadır.

$$\text{Ortalama } RSSI_n = 0,5 \times \text{Ortalama } RSSI_{n-1} + 0,5 \times RSSI_n \quad (5.1)$$

Sonuç olarak RSSI ölçüm metodu vasıtasıyla haberleşen iki birim birbirleri ile hangi kalitede haberleştiklerini belirleyebilmektedirler.

5.1.4 Phoenix Zamanlayıcıları

Phoenix algoritmasında bir takım zamanlayıcılar kullanılmaktadır. Bu zamanlayıcılar sayesinde hareketliliğe karşı hızlı bir tepki mekanizması elde edilirken aynı zamanda düşük enerji harcamak için hareketlilik performansından ödün verilmeyecek şekilde kontrol mesajlaşma sıklığı kontrol edilmektedir. Phoenix algoritmasında hareketlilik kontrolünün büyük bir çoğunluğu hareketli birimde bulunmaktadır. Bu kontrollerin yapılabilmesi için hareketli birimde üç ana zamanlayıcı bulunmaktadır.

5.1.4.1 Bağlanma İsteği Zamanlayıcısı

Hareketli birimin hiç bir aktif ebeveyne bağlı olmadığı zaman aktif olan sabit bir zamanlayıcıdır. Amacı hareketli birimin periyodik olarak DIS mesajı yollamasını sağlayarak etrafında bulunan ebeveyn birimleri haberdar etmektir. Bu zamanlayıcının değeri çok yüksek tutulursa hareketli birimin ağa ilk bağlanma hızı düşecek, düşük tutulursa ise etrafında hiç bir ebeveynin bulunmaması halinde yüksek miktarda mesaj göndererek enerji harcayacaktır.

5.1.4.2 Kontrol Mesajlaşması Zamanlayıcısı

Hareketli birimin hareketliliği destekleyen bir ebeveyne bağlı olması durumunda aktif olan değişken bir zamanlayıcıdır. Kontrol mesajlaşması zamanlayıcısı geçiş gecikmesinin azaltılmasından, aynı zamanda mümkün olan zamanlarda ise mesajlaşmayı azaltarak enerji tasarrufu yapılmasından sorumludur. Bu zamanlayıcı her bir periyodunu doldurduğunda ha-

reketli birimin ebeveynine Phoenix DIS (Bölüm 5.1.1) mesajı yollar. Gönderilen bu mesaj içerisinde ebeveyn birimin hareketli birime son yolladığı mesajın RSSI değeri ve son 5 mesajın ortalama RSSI değeri bulunur. Mesajı alan ebeveyn birim aldığı mesajın RSSI değerini ve son 5 mesajın ortalama RSSI değerini hesaplayarak hiç vakit kaybetmeden hareketli birime geri gönderir. Bu noktada hem hareketli birimde hem de ebeveyn birimde iki yönlü anlık ve ortalama RSSI değerleri bulunmaktadır. Hareketli birim kendi gönderdiği yukarı yönlü mesajın RSSI değerine bakarak bir sonraki *Kontrol Mesajlaşması Zamanlacısını* kurar. Phoenix algoritmasındaki hakim düşünce bağlantı kalitesinin yükselmesi ile gönderilecek kontrol mesajlarının azaltılmasıdır. Çünkü hareketli birim ile ebeveyn birim arasında kaliteli bir bağlantı varsa yüksek ihtimalle kısa bir süre sonra da hareketli birim hala ebeveyn birimin kapsama alanı içerisinde kalacaktır. Bundan dolayı kısa bir süre içinde ebeveyn geçişi beklenmediğinden mesajlaşma sıklığının düşürülmesi enerji tasarrufu sağlayacaktır. Ters düşünülduğünde ise azalan bağlantı kalitesi kısa bir süre içinde ebeveyn geçişi olabileceğinin işareti olduğundan, kontrol mesajlaşmalarını arttırmak bağlantı kalitesi hakkında daha kesin bir bilgi verip, gerektiğinde hızlı bir şekilde ebeveyn geçişini sağlamaktadır.

5.1.4.3 Cevap Bekleme Zamanlayıcısı

Hareketli birimin hareketliliği destekleyen bir ebeveyne bağlı olması durumunda aktif olan sabit bir zamanlayıcıdır. *Cevap Bekleme Zamanlayıcısı* bağlantının ebeveyn birim ile devam edip etmediğini kontrol etmekten sorumludur. *Kontrol Mesajlaşması Zamanlayıcısı* tarafından gönderilen Phoenix DIS mesajına, bağlantı olması durumunda ebeveyn birimden hızlı bir şekilde cevap gelecektir. *Cevap Bekleme Zamanlayıcısı* hareketli birim tarafından Phoenix DIS mesajı gönderildiği anda çalıştırılır. Belirlenen bir süre içerisinde cevap gelmez ise bağlantıda bir sorun olduğu algoritma tarafından anlaşılır. Hareketli birime cevap gelmemesi durumunda, en kısa aralıklar ile ebeveyn birime tekrardan kontrol mesajları gönderilir. Gönderilen belirli bir sayıdaki mesaja cevap gelmediği takdirde algoritma o bağlantının geçersiz olduğuna kanaat getirip yeni ebeveynler aramak amacıyla DIS mesajı gönderimini sağlayan *Bağlanma İsteği Zamanlayıcısını* çalıştırır.

5.1.4.4 Hareketli Birim Güncelleme Zamanlayıcısı

Ebeveyn birimde bulunan bu zamanlayıcı bulunduğu birime bağlı hareketli birimlerin bağlantı durumlarını güncellemekten sorumludur. Hareketli birim seçtiği ebeveyn birime Phoenix algoritması aracılığı ile bağlandığı takdirde ebeveyn birimde parametrelerin tutulması için bir miktar geçici bellek kullanılır. Tutulan bu bellek alanı her bir hareketli birim için artacağından, ortamda birçok hareketli birim bulunması halinde ebeveyn biriminin belleği dolacaktır. Bu yüzden aktif olarak bağlı olmaması durumunda ilgili hareketli birimlerin kapladığı bellek alanları temizlenmelidir. Hareketli birimler genellikle habersiz bir şekilde, bağlantıları koparak ayrıldıkları için ebeveyn birim koştukları anı yakalayamaz. Onun yerine hareketli birim, bu zamanlayıcının periyodu içerisinde kontrol mesajı yolladığında aktif sayılır. Belirlenen periyot sayısı içerisinde hiç bir kontrol mesajı yollamayan hareketli birimler bağlı oldukları ebeveyn birimden ayrılmış sayılarak bilgileri silinir.

Sonuç olarak *İletişim Kontrol* mekanizması ile *Geçiş Kontrolü* ve *Güç Kontrolü* mekanizması için gerekli olan iki yönlü bağlantı kalitesi kestirimi, birimlerin birbirleri ile haberleşmesini sağlayan, birbirlerine emirler verebileceği iletişim kanalları altyapıları hazırlanmıştır.

5.2 Geçiş Kontrolü Mekanizması

Önceden bahsedilen MoMoRo algoritmasında (Bölüm 4.2.1) bağlantının koptuğu uygulama katmanı mesajlarının iletilememesi ile anlaşılmaktaydı. Bu yöntem sonuç olarak paket gönderim oranının hareketliliğin fazla olduğu uygulamalarda oldukça düşük olmasına neden olmaktadır. Belli bir RSSI seviyesinin altında bulunma durumunu, bağlantı kalitesinin yetersiz olması olarak yorumlayan Smart-Hop (Bölüm 4.2.2) algoritması ise bağlantı daha kullanılabilir halde iken bağlantıyı değiştirmektedir. Ayrıca iki algoritmanın da geçiş yapabilmek için uygulama katmanına bağlı olması katmanlı yapıyı ihlal etmektedir. Aşağı yönlü mesajlaşmanın da hızlı bir şekilde sağlanabilmesi için gerekli yapılar bu algoritmelerde kurulanmamıştır.

Phoenix algoritması katmanlı yapıya uygun, ebeveyn geçişlerinde RPL *Görev Fonksiyonunu* temel alan, bu sayede farklı görev fonksiyonları ile beraber çalışabilen, performansı uygulama katmanına bağlı olmadan hem aşağı yönlü hem de yukarı yönlü geçişi hızlı bir şe-

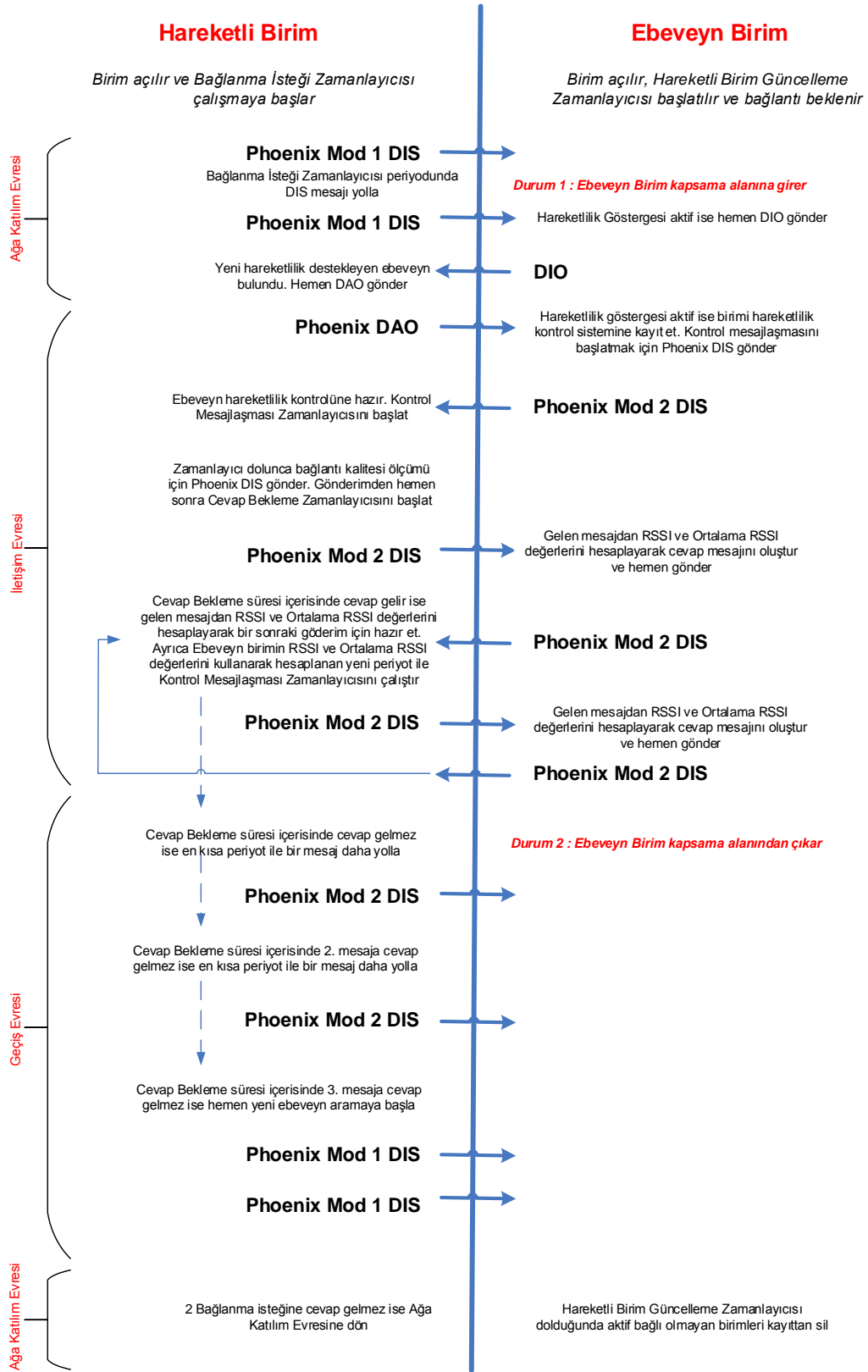
kilde gerçekleştirebilen bir şekilde tasarlanmıştır. Bunu yaparken belirli bir RSSI değerinin altında kalmayı bağlantı kalitesizliği olarak görmeyip, ağ bazında yüksek enerji harcayan ebeveyn geçiş sayısını en aza indirmek için kurulu olan bağlantıyı son noktasına kadar kullanmayı hedefler. RSSI değerini bağlantı kalitesi çıkarımı yapmak için kullanılır ve bu değer ile kontrol mesajlaşma sıklığını ayarlar. Bağlantı kopmasını ise uygulama katmanı mesajlarının kaybından değil, algoritmanın kendi gönderip aldığı kontrol mesajlarının kaybından anlamaktadır. Bu sayede uygulama mesajları kaybedilmeden geçiş sağlandığından geniş bir hareketlilik hızı aralığında paket gönderim oranı iki yönde de yüksek tutulabilmektedir. Phoenix algoritmasının temel çalışma yapısı Şekil 5.3 ile gösterilebilir ve 3 evreye ayrılır.

5.2.1 Ağa Katılma Evresi

Bu evrede, hareketli birimin herhangi bir ebeveyne bağlı olmadığı an ile yukarı yönlü iletişimin kurulduğu ana kadar geçen süre kapsanmaktadır. Hareketli birim etrafındaki ebeveynleri keşfetmek için *Bağlanma İsteği Zamanlayıcısı* tarafından belirlenen periyotlar ile herkesin duyabileceği şekilde Phoenix DIS mod 1 mesajı yayınlar. Standart RPL protokolünde DIS mesajı alan bir ebeveyn *Trickle* zamanlayıcısından dolayı belli bir süre bekledikten sonra DIO mesajı göndermektedir. Bu bekleme süreleri hızlı şekilde ebeveynine bağlanmak isteyen hareketli bir birim için geçiş süresini uzatmaktadır. Gönderilen DIS mesajı içerisinde Phoenix algoritması tarafından belirlenen bir gösterge bulunmaktadır (Bölüm 5.1.1). Gösterge sayesinde hareketli birimin kapsama alanında bulunan, Phoenix algoritmasını destekleyen bir ebeveyn birim DIS mesajını aldığı anda beklemeden DIO mesajı gönderir. Bu sayede hareketli birime en az gecikme ile ağ ile ilgili bilgiler aktarılmış olur. Hareketli birim olası ebeveyninden aldığı DIO mesajını standart RPL protokolü ve belirlenen *Görev Fonksiyonu* doğrultusunda işleyerek ebeveyni tercih edilen ebeveyn olarak seçer. Bu noktadan itibaren hareketli birimin yukarı yönlü iletişimi kullanılabilir hale gelir.

5.2.2 İletişim Evresi

İletişim evresi hareketli birimin aşağı yönlü iletişiminin oluşturulması, bağlantı kalitesinin ölçümü ve gerekirse ebeveyn değişimine kadar geçen süreyi kapsar. *Ağa Katılım* evresini tamamlayan hareketli birim hızlı bir şekilde aşağı yönlü iletişimi kurmak ve *Hareketlilik Kontrol* mekanizmasını başlatmak için ebeveyn birime gecikme olmadan Phoenix DAO me-



Şekil 5.3: Phoenix algoritması çalışma şeması

sajı yollar.

Phoenix DAO mesajını alan ebeveyn birim RPL standardı gereği öncelikle DAO mesajını kök birime kadar ileterek aşağı yönlü iletişimi hazır hale getirir. Daha sonra kendisine bir hareketli birimin bağlandığını anlayarak hareketlilik mekanizmasını çalıştırır. Öncelikle hareketli birimin kendisine daha önceden kayıtlı olup olmadığını kontrol eder. Kayıtlı değil ve kaydedecek yeterli bellek alanı mevcut ise hareketli birime kullanılmayan bir *Hareketli Birim Numarası* vererek kaydeder. Ebeveyn birim hareketlilik kontrolüne hazır olduğunu hareketli birime Phoenix DIS mesajı yollayarak haber verir.

Bu DIS mesajını alan hareketli birim ebeveyn birimin hazır olduğunu öğrenerek *Kontrol Mesajlaşması Zamanlayıcısını* çalıştırır. Zamanlayıcı dolduğunda hareketli birim ebeveyn birime iki yönlü RSSI ve ortalama RSSI değerlerini ölçmek için Phoenix DIS mesajı yollar. Mesajı yolladığını anda *Cevap Bekleme Zamanlayıcısını* çalıştırarak ebeveyn birimden cevap beklemeye başlar. Ebeveyn birim bu süre zarfında aldığı Phoenix DIS mesajını işlemeye başlar. Öncelikle DIS mesajının kendisine kayıtlı olan bir hareketli birimden geldiğini doğrular. Daha sonra alınan mesajın RSSI ve ortalama RSSI değerini hesaplayarak DIS mesajının ebeveyn tarafından doldurulacak kısımlarını doldurur. Bütün işlemler tamamlanırsa hareketli birime DIS mesajı cevabı yollanır.

Hareketli birim *Cevap Bekleme Zamanlayıcısı* süresi içerisinde ebeveyn birimden cevap mesajı alır ise zamanlayıcıyı iptal ederek mesajı işlemeye başlar. Hareketli birim de gelen cevap mesajından RSSI ve ortalama RSSI değerlerini hesaplayarak bilgileri Phoenix DIS mesajı içerisinde kendine ayrılan bölgeye yazar. Ebeveyn birimin ölçüp DIS mesajının içine yazdığı RSSI değeri (hareketli birimin ebeveyn birime gönderdiği mesajın ebeveyn tarafında ölçülen RSSI değeri) kullanılarak *Kontrol Mesajlaşması Zamanlayıcısının* yeni değeri hesaplanır. Hesaplama yapılırken kullanılan yonganın almaç hassasiyeti ve RSSI değeri Eşitlik 5.2 ile gösterildiği gibi 0 ile 9 arasına çevrilir. Bu değer daha önceden belirlenmiş 10 elemanlı bir *DIS Gönderme Aralığı* dizisinden karşılık gelen periyodu seçer. Burada 9 en uzun DIS gönderme periyoduna karşılık gelirken 0 değeri en sık DIS mesajı gönderimini belirtmektedir. Örneğin *DIS Gönderme Aralığı* dizisi saniye cinsinden ilk elemanından son elemanına sırası ile 1,2,3,4,5,6,7,8,9 ve 10 değerlerine sahip ise ve çevrim sonucunda 3. elemanın seçilmesi hesaplanmışsa bir sonraki kontrol mesajlaşması 3 saniye sonra yapılacaktır.

$$X = \frac{(RSSI - Alma Hassasiyeti_{min}) \times 9}{(Alma Hassasiyeti_{max} - Alma Hassasiyeti_{min})} \quad (5.2)$$

Bu sayede hareketli birim bağlantı kalitesini tahmin edip yakın bir zamanda ebeveyn değiştirip değiştirmeyeceğine karar verir. Bağlantı kalitesi arttıkça değişim olma olasılığı da azalacağından enerji tasarrufu yapmak için kontrol mesajlaşma sıklığını azaltır. Aksi bir durumda bağlantı kalitesi almaç hassasiyeti ile yakın seviyelere geldiğinde, kısa bir süre sonra geçiş gerçekleşme olasılığı fazla olduğundan kontrol mesajlaşmasını sıklaştırır. Sonuç olarak seçilen değer ile *Kontrol Mesajlaşması Zamanlayıcısı* yeniden çalıştırılarak kontrol mesajlaşmaları gönderilip alınır.

Phoenix algoritmasında *Kontrol Mesajlaşması Zamanlayıcısı* için yardımcı olarak kullanılacak bir tasarruf mekanizması daha geliştirilmiştir.

5.2.2.1 Durağanlık Algılayıcısı

Bu mekanizmada "hareketli birimin hareketlerinin azalması durumunda ebeveyn değişimi gerçekleştirme olasılığı da azalma eğilimi gösterecektir" düşüncesi hakimdir. Bundan dolayı hareketli birim durağan bir hale gelirse durağan kaldığı süre boyunca *Kontrol Mesajlaşması Zamanlayıcısı* bir miktar kısılarak, gönderilecek kontrol mesajı sayısı azaltılıp enerji tasarrufu sağlanabilecektir. *Durağanlık Algılayıcısı* aktif olmak için RSSI değerinin değişimini inceler. Algoritma 1 ile gösterilen algoritma çalıştığı taktirde her mesajda RSSI değeri sisteme olduğundan daha büyük gösterilecektir. Bu sayede *DIS Gönderme Aralığı* dizisinden normalden daha az sıklıkta kontrol mesajı gönderimini sağlayacak değerler seçilecektir. Seçim ebeveyn geçiş hızını çok fazla yavaşlatmamak için çevrim değerini en fazla 4 birim yavaşlatacak şekilde sınırlandırılmıştır.

Sonuç olarak hareketli birimin hareketi azalıp durağan bir yapıya döndüğü andan itibaren kontrol mesaj sıklığı azaltılarak enerji tasarrufu sağlanabilecektir. Durağan birimde tekrardan bir hareketlenme olduğunda bu yapı iptal edilerek normal çalışmaya devam edilecektir.

Algoritma 1 Durağanlık Algılama Algoritması

Girdi:

$RSSI$: Anlık ölçülen RSSI değeri

$RSSI_{etkin}$: Durağanlık sonucunda etkin RSSI değeri

$RSSI_{etkin(n-1)}$: Geçici Değer

$ORSSI$: Ortalama RSSI değeri

$RSSINORM$: RSSI değeri ile yapılan normalizasyon değeri

$RSSINORM_{etkin}$: Etkin RSSI değeri ile yapılan normalizasyon

$RSSI_{etkin} = RSSI$

while ($ORSSI - RSSI \leq |2dB|$) **do**

Durağanlık Aktif

if ($RSSINORM_{etkin} - RSSINORM \leq 4$) **then**

$RSSI_{etkin} = 1.1 \times RSSI_{etkin(n-1)}$ (*Her kontrol mesajında RSSI etkin değeri artar*)

$RSSI_{etkin}$ ile normalizasyon hesapla

else

$RSSI_{etkin}$ ile normalizasyon hesapla

end if

$RSSI_{etkin(n-1)} = RSSI_{etkin}$ Kontrol Mesajlaşması Zamanlayıcısını kur

end while

return

5.2.3 Geçiř Evresi

Geçiř evresi, hareketli birimin ebeveyn birime gönderdiđi DIS mesajına cevap alamamasından itibaren yeni bir ebeveyn birime bağlanmasına kadar olan süreci içerir. Hareketli birim ebeveyn birime mesaj gönderdiđi anda cevap mesajını beklemeye başlar. Bu süre zarfında herhangi bir cevap alınmadı ise birim *Geçiř Evresine* girer. Geçiř evresinde amaç bağlantının koptuđundan emin olunup ebeveyn deđişimini gerçekleřtirmektir. Gereksiz yere ebeveyn deđişimi yapmak kök birime kadar gönderilen DAO mesajlarından dolayı ađ bazında büyük miktarda enerji kaybı yaratacaktır. Hareketli birim, gönderdiđi DIS mesajına karşılık alamadıđında bağlantının ortalama RSSI deđerini kontrol eder. Ortalama deđer belirlenen bir "Güvenilir RSSI" deđerinin üzerinde ise bağlantıda anlık bir sorun olduđu düşünülüp *Kontrol Mesajlaşması Zamanlayıcısı* en son deđerine kurularak *İletişim Evresine* geri dönülür. Gönderilen mesaja tekrardan cevap gelmez ise veya ilk mesaj kaybında ortalama deđer "Güvenilir RSSI" deđerinin üzerinde deđil ise *Kontrol Mesajlaşması Zamanlayıcısı* en kısa deđerine kurularak 2 sefer daha mesaj gönderilir. Mesajlardan herhangi birisine cevap gelir ise yeniden *İletişim Evresine* geçilir. Ancak gönderilen 2 mesaja da cevap alınmaz ise ebeveyn ile bağlantının koptuđu düşünülerek ebeveyn birim tercih edilen ebeveyn birimlikten çıkarılır. Bu durum Phoenix algoritması tarafından üst katmanlara bildirim olarak iletilir. Uygulama katmanı isterse bu noktada göndereceđi mesajları yeni bir ebeveyn birime bağlanana kadar bekletebilir. Böylelikle paket gönderim oranı bir kat daha arttırılabilir. Hareketli birim hızlı bir şekilde yeni bir ebeveyn bulmak için ortama en kısa *Kontrol Mesajlaşması Zamanlayıcı* periyodu ile DIS mesajı yayar. Ortamda bulunan alternatif bir ebeveyn birimden DIO mesajı gelir ise hızlı bir şekilde ebeveyn deđiřtirilerek *İletişim Evresine* geçilir. Belirlenen sayıda gönderilen DIS mesajına cevap alınmamıř ise ortamda alternatif bir ebeveyn bulunamadıđı düşünülür. Hareketli birim enerji tasarrufu yapmak için *Kontrol Mesajlaşması Zamanlayıcısını* kapatarak daha yavař DIS mesajı gönderimini sađlayan *Bađlanma İsteđi Zamanlayıcısını* aktif eder ve tekrardan *Ađa Katılım* evresine geçer.

Sonuç olarak *İletişim Kontrolü* altyapısı kullanılarak *Geçiř Kontrolü* mekanizması kullanılabilir hale getirilmiřtir. Bu sayede hareketli birim enerji tasarrufunu ön planda tutarak ađ içerisinde hızlı bir şekilde ebeveyn deđiřtirebilirken yüksek paket gönderim oranını da sađlayabilmektedir.

5.3 Güç Kontrolü Mekanizması

Phoenix Algoritmasının temelini oluşturan *İletişim Kontrolü Mekanizması* sayesinde birimler, aralarındaki bağlantının kalitesini belirlerken, aynı zamanda birbirleri ile haberleşebilir ve birbirlerine emirler verebilirler. Bu altyapı geçiş mekanizmasında kullanılabilmesi gibi daha birçok amaçla da kullanılabilir. *Güç Kontrolü* yardımcı mekanizması da bu uygulama alanlarından birisidir.

KSA'larda daha önce güç kontrolü üzerine birçok çalışma yapılmıştır [69][70][71]. Ancak yapılan bu çalışmalar güç kontrolünü, standart protokollerden bağımsız bir şekilde geliştirmişler ve gelişimleri deneysel çalışmalar ile doğrulamamışlardır. RPL protokolü ile uyumlu çalışabilen, deneysel sonuçlarla doğrulanmış bir güç kontrolü şeması yakın bir süre önce geliştirilmiş olup [72], ağ bazında sabit tutulan bir çıkış gücü değerinin ağ performansını düşürdüğünü göstermiştir. Hem güç tasarrufu sağlamak, hem de paket gönderim oranını arttırmak için ağ bazında sabit olmayan bir güç çıkış değerlerine sahip olunması gerektiği incelenmiştir. Ancak bu yöntem çoğunlukla durağan ağ yapıları için kullanılacak ve hareketliliğe uyum sağlamayan bir yapıda tasarlanmıştır.

Katmanlı yapının avantajlarında anlatıldığı gibi (Bölüm 2.2) en önemli özelliklerden bir tanesi katmanların birbirlerinden bağımsız çalışabilmeleridir. Phoenix algoritmasının üzerinde çalıştığı Contiki işletim sistemi de (Bölüm 2.3.1) katmanlı yapıyı destekleyecek şekilde kurgulanmıştır. Bu sayede örneğin yönlendirme katmanı ortam erişim katmanında bulunan bir parametrenin değişimini isterse bunu önceden tanımlanmış fonksiyonlar aracılığı ile bu isteğini yapabilmektedir. Kullanılan fonksiyonlar radyo yongaları değişse bile sabit kalmaktadır. Çünkü bu fonksiyonların tanımlamaları desteklenen her yonga için ortam erişim katmanında yapılmış olduğundan, üst katmanlardan bakıldığında bütün yongalar eşit görünmektedir. Uygulama katmanı ortam erişim katmanının kullanılan radyo frekans kanalını değiştirmesini isterse bunu o işletim sistemi tarafından desteklenen bütün radyo yongaları için aynı komut ile yapar. Bu sayede alttaki katmandaki değişiklikler gözetilmeksizin uygulamalar geliştirilebilir.

Phoenix *Güç Kontrolü* mekanizması anlatılan katmanlı yapının avantajlarını kullanarak çalışır. Bu mekanizmanın amacı mümkün oldukça radyo gücünün kısılması ile ağ bazında enerji tasarrufu yapılmasıdır. Güç kontrolü ile yapılan enerji tasarrufu sadece paket göndermek

için gerekli olan enerjinin azalmasıyla kısıtlı değildir. Gönderilen her paket, paket hedefi kendileri olmasa bile, paketi alan birimler tarafından işlenerek enerji tüketimi yaratmaktadır. Örneğin yüksek güç ile gönderilen her DIS mesajı alıcıların *Trickle Zamanlayıcısını* sıfırlayarak belki yüzlerce gereksiz kontrol mesajının ağ içerisinde gönderilmesine sebebiyet vermektedir. Phoenix algoritması kapsamında bu soruna çözüm üretebilmek için hem uygulama katmanı mesajlarının, hem de kontrol mesajlarının ideal güç seviyesi ile gönderilmesi hedeflenmiştir. Tez kapsamında şu an için hareketli birimlerin gönderdiği kontrol ve uygulama katmanı mesajları gönderiminde güç kontrolü yapılabilirken, ebeveyn birimlerin sadece hareketli birimlerine gönderdiği kontrol mesajları gönderiminde güç kontrolü yapılabilmektedir.

Phoenix *Güç Kontrolü* mekanizmasının temel çalışma prensibi bağlantı gücünü, başarılı paket gönderim oranını etkilemeyecek kadar düşük, fakat ani değişimlere ise ayak uydurabilecek kadar yüksek tutmaktır. Bu amaçla paket gönderim oranının neredeyse %100 olduğu, bağlantının anlık değişimlere karşı dayanım göstereceği ve radyo yongasının alıcı hassasiyetinden 25 dB az olduğu güç noktası hedef RSSI seviyesi olarak belirlenmiştir. Güç seviyesinin çok sık değişmemesi için hedef noktanın ± 7 dB içerisinde olmak yeterli sayılmıştır. *Güç Kontrol* mekanizması hareketli birim ile ebeveyn birim arasındaki bağlantı kalitesinin ortalama RSSI değerini kullanarak çalışır. Ortalama RSSI değerini hedef RSSI değerine eşitlemek için her kontrol mesajında adım adım gücü kısar ya da artırır. Ancak kısılan iletim gücü anlık bağlantı kalitesi değişimlerinde paket kaybı yaratma riskini de beraberinde getirir. Bu yüzden iletişim esnasında paket kaybı olur ve *İletişim Evresinden Geçiş Evresine* geçilir ise hemen iki yönlü güç seviyesi artırılarak yeniden sağlıklı bir iletişim kurulmaya çalışılır. Algoritma 2 ile *Güç Kontrolü* mekanizmasının çalışma prensibi detaylı olarak anlatılabilir.

Sonuç olarak *Güç Kontrolü* mekanizması ile RPL protokolü ile uyumlu bir güç kontrol şeması geliştirilmiştir. Geliştirilen şemanın amacı gereksiz yüksek çıkış gücü kullanımı önlenmek, bu sayede hem uygulama katmanı mesajlarında hem de kontrol mesajlarında ağ bazında bir enerji tasarrufu sağlamaktır. Güç azaltışı yapılırken uygulama paketi kaybını önlemek için ister asimetrik, isterse simetrik bağlantı kalitesi sorunu olduğunda iki birimde çıkış güçleri hızlıca yükseltilerek bağlantının tekrardan kullanılabilir hale getirilmesi planlanmıştır.

Algoritma 2 Hareketli Birim İçin Güç Kontrolü Algoritması

Girdi:

RSSI: Anlık ölçülen RSSI değeri

ORSSI: Hareketli Birimin Ebeveyn birime gönderdiği mesajın Ortalama RSSI değeri

HEDEFRSSI: Hedef RSSI değeri

if (*Cevap Bekleme Zamanlayıcısı* içerisinde ebeveyn birimden cevap alındı) **then**

Gelen Phoenix DIS Mod 1 mesajını aç

ORSSI değerini öğren

if ($ORSSI \leq (HEDEFRSSI - 7)$) **then**

Çıkış gücü yetersiz

Çıkış gücünü 1 kademe arttır

else if ($ORSSI \geq (HEDEFRSSI + 7)$) **then**

Çıkış gücü fazla

Çıkış gücünü 1 kademe azalt

else

Çıkış gücü yeterli

Çıkış gücünü değiştirme

end if

else if (*Cevap Bekleme Zamanlayıcısı* içerisinde 1. mesaja cevap alınamadı) **then**

Hareketli birimin çıkış gücü yetersiz olabilir

Çıkış gücünü 1 kademe arttır

Ebeveyn birimin çıkış gücü yetersiz olabilir

Ebeveyn birime çıkış gücünü en yüksek seviyeye getirme emri ver

else if (*Cevap Bekleme Zamanlayıcısı* içerisinde 2. mesaja cevap alınamadı) **then**

Hareketli birimin çıkış gücü yetersiz olabilir

Çıkış gücünü en yüksek seviyeye getir

Ebeveyn birimin çıkış gücü yetersiz olabilir

Ebeveyn birime çıkış gücünü en yüksek seviyeye getirme emri ver

else

Geçiş Kontrolü Mekanizması yönetimi devralır

end if

return

6. BENZETİM VE SAHA TESTLERİ

Tez kapsamında tasarlanan Phoenix Hızlı Geçiş algoritması, Contiki işletim sistemi üzerinde geliştirilmiştir. Contiki işletim sistemi, katmanlı tasarım yapısını temel aldığı için geliştirilen algoritma sadece yönlendirme katmanını değiştirecek şekilde Contiki içerisine yerleştirilmiştir. Algoritma RPL protokolünün yönetildiği `contiki/core/net/rpl/` dizini altında 2 dosya şeklinde geliştirilmiştir. Bu dosyalar;

- `mobile_icmp.h` : Phoenix algoritmasının fonksiyon ve değişken tanımlarının yapıldığı dosya
- `mobile_icmp.c` : Phoenix algoritmasının fonksiyonlarının işletildiği dosya

Geliştirilen algoritma KSA kısıtlamalarından olan sınırlı bellek kaynağı göz önüne alınarak kurgulanmıştır. Örneğin Contiki örnek uygulamalarından olan periyodik UDP mesaj gönderme uygulaması 126kB kalıcı ve 16kB geçici belleğe ihtiyaç duyarken, aynı uygulama Phoenix algoritmasıyla derlenmesi halinde 9kB kalıcı ve 600B ek geçici belleğe ihtiyaç duymaktadır. Bu boyutlar daha azaltılabilecek olsa da, günümüz işlemcileri için bellek konusunda bir sorun yaratmayacağı düşünülmüştür.

6.1 Test Altyapısı

6.1.1 Phoenix Algoritma Parametreleri

Phoenix algoritması hareketliliğin yönetilmesi için bir çok parametreye sahiptir. Bu parametreler aracılığı ile hareketlilik performansı ve enerji tasarrufu arasında değişkenlik sağlanabilir. Uygulama katmanındaki isterler bu dengenin nerede kurulacağını belirler. Tez kapsamında yapılacak performans testlerinde uygulama olarak düzensiz hareket eden insan modeli (Bölüm 4.1.2) göz önüne alınmıştır. Bu modelde hızlar 0-4 m/s civarında olup hareketliliğin bir anda ortaya çıkabildiği ve bir anda durağanlaştığı bir hareketlilik şeması ele alınmıştır.

Bahsedilen hareketlilik modeli için Phoenix algoritmasına girilen parametreler şunlardır:

- Hareketli Birim Sayısı: Bir ebeveyn birimin aynı anda kontrol edebileceği hareketli birim sayısıdır. Bu modelde 8 olarak seçilmiştir. Bu sayı kullanılan mikrodenetleyicinin geçici hafızası ile doğru orantılıdır.
- Bağlanma İsteği Zamanlayıcısı: Hareketli birim hiç bir ebeveyn birime bağlı olmadığı anda bağlantı isteği göndermek için bu zamanlayıcıyı kullanmaktadır. 5 saniye olarak seçilen bu değer bağlanma hızı ve enerji tüketiminin arasında seçilmiştir.
- Kontrol Mesajlaşması Zamanlayıcısı DIS Gönderme Aralığı dizisi (saniye): [0.6 , 1 , 1.8 , 3 , 5 , 9 , 13 , 20 , 35 , 60]
- Cevap Bekleme Zamanlayıcısı: Hareketli birimin gönderdiği Phoenix DIS mesajına cevap beklediği süredir. 0.3 saniye süresi içerisinde geçerli bir mesaj alınmış olması gerekmektedir.
- Hareketli Birim Güncelleme Zamanlayıcısı: Ebeveyn birimin kendisine bağlı olduğu hareketli birimlerin geçerliliğinin kontrol edildiği süredir. 120 saniye içerisinde hareketli birimden cevap alınamamış ise ilgili hareketli birim ile ilişki kesilir.
- Güvenilir RSSI değeri : Bağlantının güvenilir olduğunu gösteren değerdir. Bu modelde CC2538 yongasının hata değerinin %1 civarında olduğu değer olan 60 dB olarak seçilmiştir.
- Hedef RSSI değeri : Güç çıkışı ayarı ile hedeflenen RSSI değeridir. Bu modelde 60 dB olarak seçilmiştir.

Tasarlanan Phoenix geçiş algoritmasının performansını değerlendirmek için bir takım ölçütlere ihtiyaç vardır. Bu ölçütler algoritmanın hangi faydaları sağlarken hangi değerleri kabul edilebilir seviyelerde tuttuğunu gösterir.

- **Geçiş Gecikmesi:** Geçiş gecikmesi geçiş algoritması için en önemli parametrelerdendir. Ancak geçiş gecikmesinin tanımı uygulamadan uygulamaya fark edebilmektedir. Geçiş gecikmesinin tanımı olarak hareketli birimin bağlı olduğu ebeveyn birimin kapsama alanından çıktığı anda yeni kapsama alanında bulunan ebeveyne bağlanması için

gerekli süre olarak belirlenmiştir. Tez kapsamında geçiş gecikmesi aşağı yönlü mesajlar için ve yukarı yönlü mesajlar için ayrı ayrı incelenmiştir. Yukarı yönlü mesajlar için yeni ebeveyn birimin seçildiği an, aşağı yönlü mesajlar için ise seçili ebeveyn birime DAO mesajının gönderildiği an geçişin tamamlanması olarak tanımlanmıştır.

- **Ağ Yüğü:** Ağ yüğü, kablosuz sensör ağlarında uygulama katmanı mesajı dışındaki diğer bütün mesajların toplamıdır. Bu değeri ağ bazında ölçmek için Cooja ağ benzetim programında bulunan radyo görev çevrim aracı kullanılabilir. Kullanılan araçta her bir birimin radyo devrelerinin hangi yüzde ile açık kaldığı, alma ve gönderme yaptığı incelenebilir. Böylelikle direkt olarak enerji tüketimi karşılaştırılabilir.
- **Başarılı Paket Gönderim Oranı:** Gönderilen uygulama katmanı mesajlarının hangi yüzde ile hedefine teslim edildiğini gösteren değerdir. Yukarı ve aşağı yönlü mesajlar ayrı ayrı incelenecektir.
- **Güç Tüketimi:** Tez kapsamında yapılan çalışmalarda Cooja ağ benzetim programının gerçek donanımla yapılan testler ile neredeyse aynı sonuçlar verdiği görülmüştür. Ancak benzetim ortamında kablosuz sensör ağları için en önemli değer olan güç tüketimi ölçülememektedir. Bundan dolayı saha testleri yapılırken tasarlanan geliştirme kartı üzerinde karşılaştırmalı güç tüketimi testi yapılacaktır.

6.1.2 Karşılaştırılacak RPL Modelleri

Tasarlanan Phoenix algoritmasının hangi parametreleri iyileştirip hangilerini uygunsuz hale getirdiğini görmek için karşılaştırılmalı testler yapılması gerekmektedir. Bundan dolayı tez kapsamında yukarıda verilen parametreler ile çalışan Phoenix algoritması ile karşılaştırmak üzere, *Trickle* (Bölüm 3.6) zamanlayıcısı farklı hızlarda çalışan ve standart RPL protokolü kullanan iki farklı birim kullanılmıştır.

- **RPL_12_8:**

RPL_12_8 isimli birimde *Trickle* zamanlayıcısı parametreleri Eşitlik 3.1 ile hesaplanabilir;

MINIMUM DIO ARALIGI = 12

DIO ARALIK KATLAMASI = 8

değerleri eşitliğe girilirse 2 DIO mesajı arası en kısa periyot 4 saniye, en uzun periyot ise 17 dakika olarak bulunur. Contiki işletim sisteminde kullanılan standart *Trickle* zamanlayıcısı değerleri bunlardır:

- **RPL_8_4;**

RPL_8_4 isimli birimde ise *Trickle* zamanlayıcısı parametreleri;

MINIMUM DIO ARALIGI = 8

DIO ARALIK KATLAMASI = 4

olarak seçilmiştir. Bu sayede 2 DIO mesajı arası en kısa periyot 256 milisaniye, en uzun periyot ise 4 saniye olarak bulunur.

- **P-RPL;**

Tez kapsamında geliştirilen birimdir. Bu yapıda hareketli birim ile ebeveyn haberleşirken geliştirilen algoritma kullanılırken, hareketli birime bağlı olmayan birimler aralarında standart 12_8 *Trickle* zamanlayıcısı parametrelerini kullanırlar.

6.2 Benzetimler

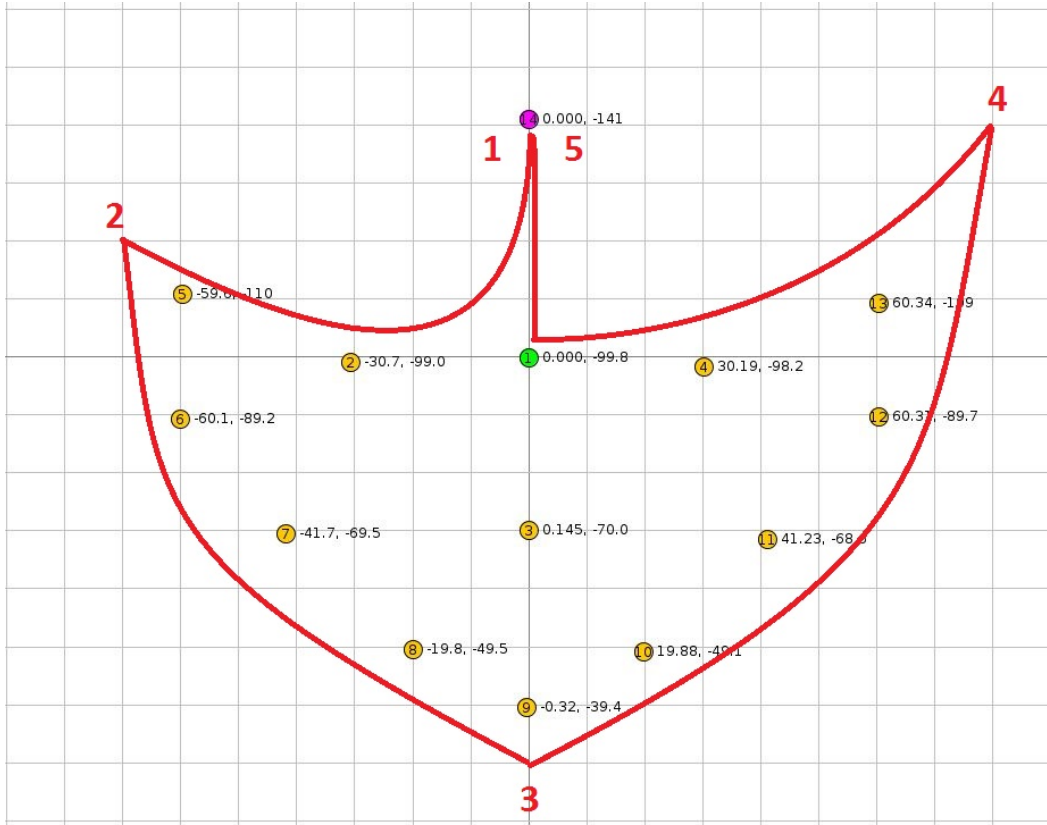
Gerçek testlere çok yakın sonuçlar veren benzetimler, gerçek birimlerle yapılması zor ve zaman alıcı olan deneyleri yapmak için kullanılabilir. Dahası karşılaştırmalı testler yapılırken, test edilen birimlerin tamamen aynı senaryo ile çalışması sağlanabilmektedir. Bu sayede karşılaştırmalar daha doğru sonuç verecektir.

6.2.1 Benzetim Senaryosu

Tez kapsamında bahsi geçen birimlerin karşılaştırılması için bir test senaryosu geliştirilmiştir. Bu senaryoda amaç düzensiz hareket eden insan modelini (Bölüm 4.1.2) test etmektir. Şekil 6.1 ile geliştirilen senaryo görülebilmektedir. Senaryo, 1 kök (yeşil), 12 ebeveyn (sarı) ve 1 hareketli (mor) birimden oluşmaktadır. Test, durağan ve farklı hızlarda hareketin olduğu anları içermektedir.

Senaryo adımları şunlardır;

1. Test çalışmaya başladığında test edilen birim ağa katılmadan önce, ağın yönlendirme



Şekil 6.1: Test Senaryosu

tablolarını oluşturması için 15 dakika süre ile beklenildiği andır. 15 dakika sonunda hareketli birim ağa katılarak kök birim ile aşağı ve yukarı yönlü, değişken periyotlar ile haberleşirken ağ içerisinde değişken hızlar ile kırmızı çizgiyi izleyerek 2 numaralı konuma doğru hareket etmeye başlar.

2. Hareketli birim kırmızı çizgiyi izleyerek bu noktaya geldiğinde 1 dakika bekleyerek 3 numaralı konuma doğru yoluna devam eder.
3. Hareketli birim kırmızı çizgiyi izleyerek bu noktaya geldiğinde 1 dakika bekleyerek 4. numaralı konuma doğru yoluna devam eder.
4. Hareketli birim kırmızı çizgiyi izleyerek bu noktaya geldiğinde 1 dakika bekleyerek 5. numaralı konuma doğru yoluna devam eder
5. Bu noktaya geldiği an test sonlanır ve değerler kayıt edilerek bir sonraki teste geçilir.

6.2.2 Benzetim Sonuçları

Tez kapsamında farklı hızlar, uygulama katmanı mesajlaşma periyotları, mesajlaşma yönleri, çalışma seçenekleri ile toplam 66 adet benzetim testi yapılmış olup sonuçları grafikler ile anlatılmıştır. Her bir testin sonucu 3 adet grafikte anlatılmaktadır. İlk grafik değişken parametreye göre paket gönderim oranını göstermektedir. İkinci grafik ise değişken parametreye karşı hareketli birimin radyo görev çevirimini göstermektedir. Her bir birimin 3 ayrı radyo görev çevrimi değeri vardır. Bunlar sırasıyla radyonun toplamda ne kadar açık kaldığı, radyonun ne kadar gönderme yaptığı ve son olarak radyonun ne kadar alma yaptığıdır. Bu değerler radyodan kaynaklı enerji tüketimi çıkarımını yapmayı sağlamaktadır. Sonuncu grafik ise radyo görev çevrimi değerinin ağ bazındaki ortalamasıdır. Bu sonuçlar sayesinde paket gönderim oranı incelenirken hem birim bazında hem de ağ bazında enerji tüketimi ve verimliliği incelenebilmektedir.

6.2.2.1 Birim Hızı Karşılaştırmaları

Bu aşamada radyo çevrim katmanı 64 Hz, mesajlaşma hızı saniyede 3 mesaj olacak şekilde sabit tutulup, birim hızının başarılı paket gönderim oranına ve enerji tüketimine etkisi karşılaştırılmıştır. Test süresince aksi belirtilmedikçe *Güç Kontrolü* mekanizması ve *Durağanlık Algılama* mekanizması kapalı tutulmuştur.

Yukarı Yönlü Mesajlaşmada Birim Hızı Karşılaştırması

Testin bu aşamasında mesajlar hareketli birimden kök birime olacak şekilde yönlendirilmiştir. Şekil6.2a ile görülebileceği gibi yavaş hızlarda test edilen tüm birimler tatminkar paket gönderim oranına sahipken, hız arttıkça RPL12_8 kullanılmaz hale gelmektedir. P-RPL test boyunca en yüksek gönderim oranına sahiptir.

Enerji tüketimi açısından Şekil 6.2b incelendiğinde P-RPL yüksek hareketlilik altında en az radyo kullanım oranına sahiptir. Bunun en önemli sebeplerinden bir tanesi hızlı geçiş algoritması sayesinde sürekli aktif bir ebeveyn birime bağlı kalabilmektir. Bu sayede mesaj gönderimi yapılırken *Ortam Erişim Katmanı* mesaj gönderim onayını alabildiği için tekrar mesaj gönderimi yapmaz. Gönderilen mesaja cevap alınmadığı takdirde ise mesaj gönderimi belirlenen bir sayı kadar tekrar edilmektedir. Bu sayı test boyunca Contiki işletim sis-

teminin baz değeri olan 4 olarak kullanılmıştır. Ağ bazındaki enerji tüketimini görebilmek için Şekil 6.2c incelendiğinde ise P-RPL'nin standart RPL12_8 biriminden oluşan ağdan yaklaşık %8 daha yüksek olduğu görülmektedir. Bunun sebebi Phoenix algoritmasının ek olarak kullandığı kontrol mesajlarıdır. RPL8_4 biriminden oluşan ağın ise fazla miktarda kontrol mesajı gönderiminden dolayı P-RPL biriminden 1,5 kat daha yüksek enerji tükettiği görülebilmektedir.

Sonuç olarak geliştirilen Phoenix algoritması ağ bazında neredeyse standart RPL kadar az enerji harcayıp, birim bazında en az enerjiyi harcarken geniş bir hareketlilik aralığında %91 ortalama ile en yüksek paket gönderim oranına sahiptir. Hareketli birimin mesaj gönderme verimliliği düşünüldüğünde (paket gönderim oranı)/(radyo kullanımı oranı) ile kabaca bir değer elde etmek mümkündür. Bu değer sadece kullanılan birim enerji için gönderilen paket oranıdır. Başarılı paket gönderim oranı sayısı ayrıca incelenmelidir. Verimliliğin bu test için test ortalaması alındığında:

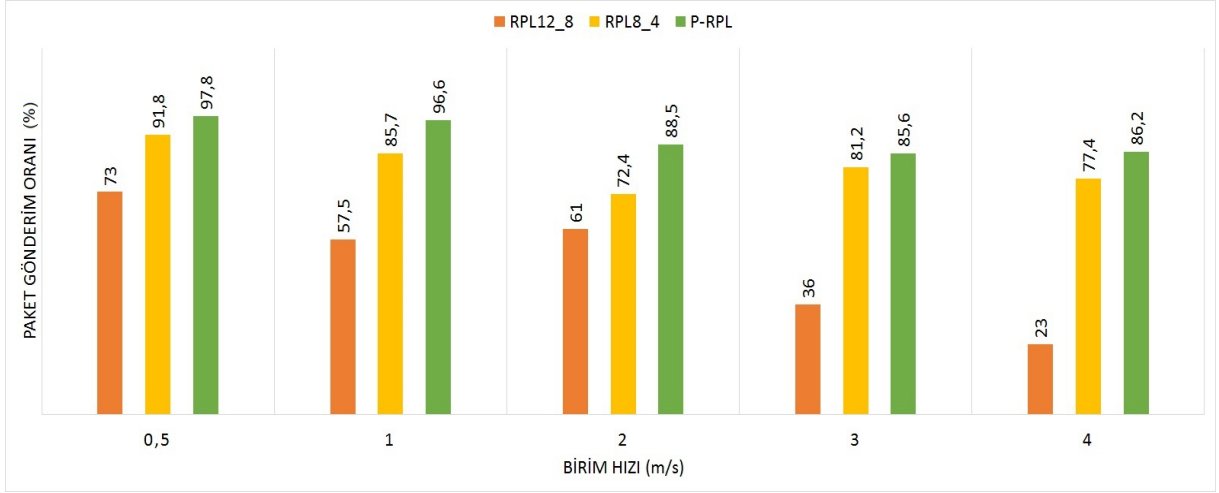
- RPL12_8 : 0,75
- RPL8_4 : 1,1
- P-RPL : 1,35

değerleri bulunmaktadır.

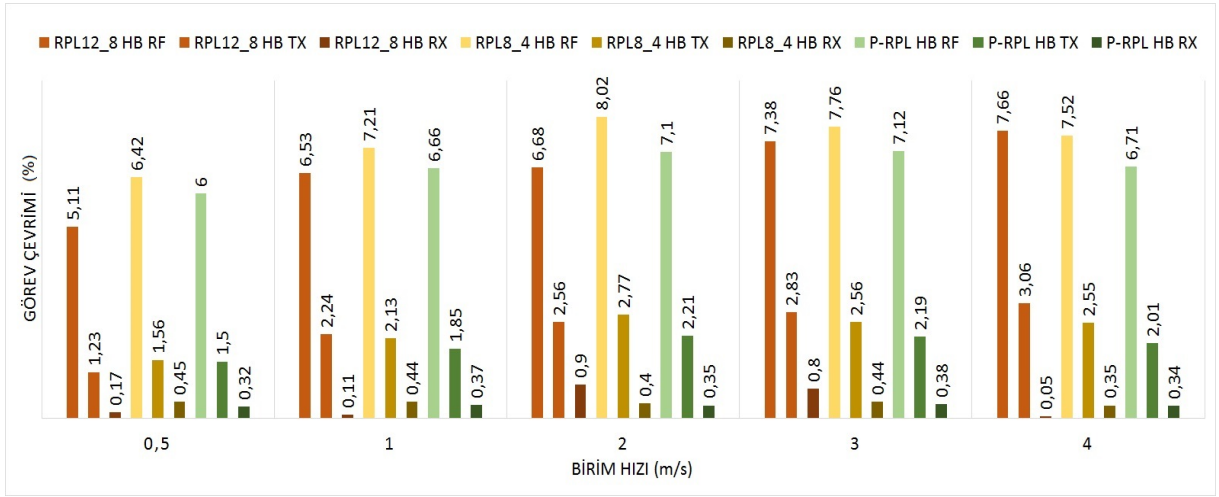
Aşağı Yönlü Mesajlaşmada Birim Hızı Karşılaştırması

Testin bu aşamasında mesajlar kök birimden hareketli birime olacak şekilde yönlendirilmiştir. Şekil 6.3a ile görülebileceği gibi yavaş hızlarda bile RPL12_8 ve RPL8_4 birimleri kabul edilemez paket gönderim oranına sahipken, P-RPL test boyunca açık ara en yüksek gönderim oranına sahip olmuştur. Bunun sebebi DAO mesajı gönderiminin bekletilmeden yapılması sayesinde aşağı yönlü yönlendirme tablosunun hızlı bir şekilde güncellenmesini sağlamaktır. Bu sayede *Phoenix Algoritmasının* diğer geçiş algoritmalarından farklı olarak aşağı yönde de yüksek paket gönderim oranına sahip olma özelliği kanıtlanmaktadır.

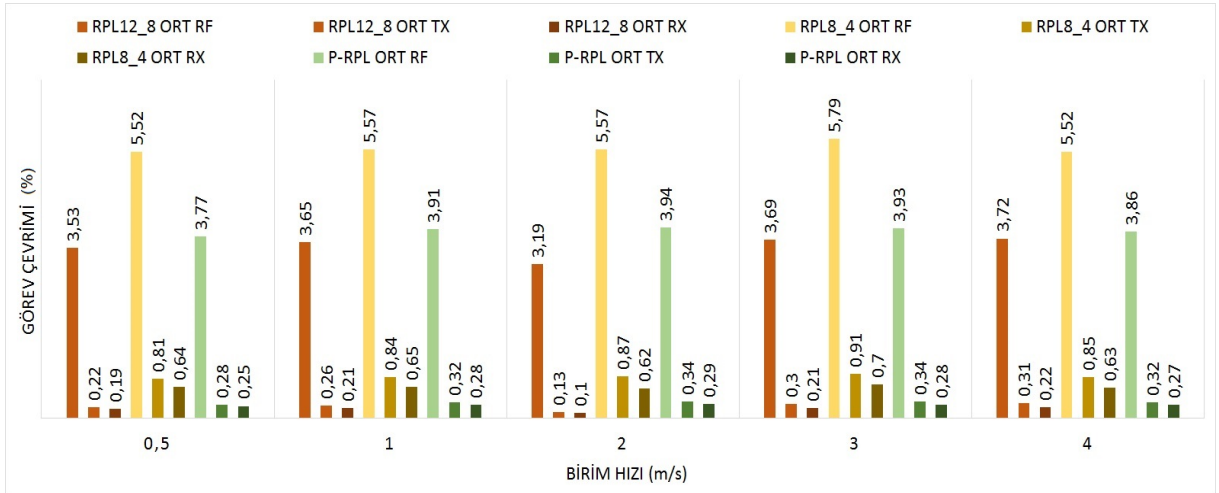
Enerji tüketimi açısından Şekil 6.3b incelendiğinde P-RPL hareketli biriminin test boyunca en yüksek radyo kullanıma sahip olduğu görülmektedir. P-RPL en yüksek enerji tüketimine sahip olsa da diğer birimler hareketlilikten dolayı kendilerine mesaj ulaştırılmadığından



(a) Yukarı yönlü mesajlaşmada birim hızına karşılık paket gönderim oranı grafiği



(b) Yukarı yönlü mesajlaşmada birim hızına karşılık hareketli birim radyo görev çevrimi oranı grafiği



(c) Yukarı yönlü mesajlaşmada birim hızına karşılık ortalama radyo görev çevrimi oranı grafiği

Şekil 6.2: Yukarı yönlü mesajlaşmada birim hızı karşılaştırmaları

radio kullanımları azalmaktadır. Bu veri paket gönderim oranı ile beraber incelendiğinde anlatılmak istenen ortaya çıkacaktır. P-RPL'nin paket gönderim oranı RPL12_8 den 7 kat yüksek iken, radio kullanımı sadece 1,7 kat yüksektir. Aynı şekilde 4,5 kat ile 1,2 kat farkı P-RPL ile RPL8_4 arasında görülebilmektedir. Ağ bazındaki enerji tüketimini görebilmek için Şekil 6.3c incelendiğinde ise R-RPL'nin kontrol mesajlarından dolayı RPL12_8 den yaklaşık %10 daha fazla enerji tükettiği, RPL8_4'ün ise sık kontrol mesajlaşmasından ötürü en yüksek enerji tüketimine sahip olduğu görülmektedir.

Sonuç olarak geliştirilen *Phoenix Algoritması* ağ bazında neredeyse standart RPL kadar az enerji harcayıp, birim bazında ise en yüksek (paket gönderim oranı)/(radio kullanımı oranı) değerine sahiptir. Diğer birimlerin geniş bir hareketlilik aralığında kabul edilemez derecede başarısız olduğu bu testte P-RPL ortalama %87 oranında paket gönderimine sahiptir. Mesaj gönderim verimlilikleri;

- RPL12_8 : 0,39
- RPL8_4 : 0,62
- P-RPL : 2,0

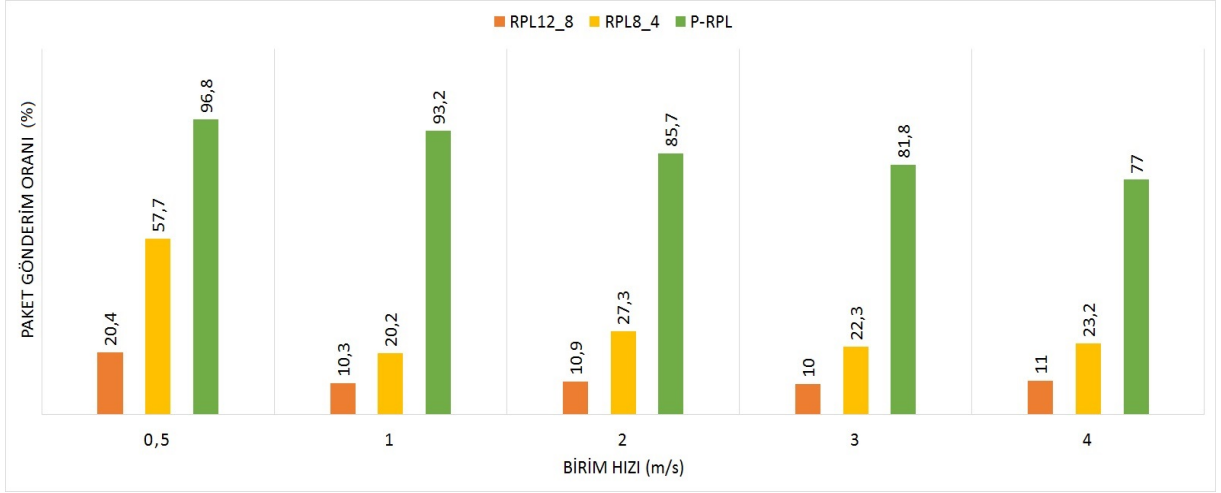
olarak ölçülmektedir.

6.2.2.2 Mesajlaşma Hızı Karşılaştırmaları

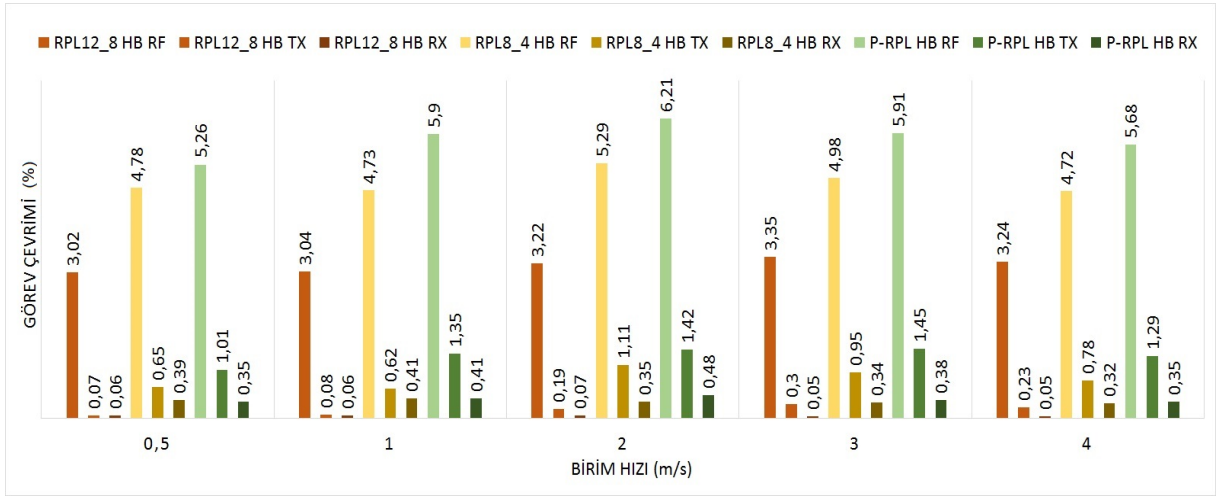
Bu karşılaştırmada birim hızı ortalama insan yürüyüş hızı olan saatte 5 kilometre olacak şekilde sabit tutulup, radio çevrim katmanı 64 Hz ve 8 Hz olacak şekilde değiştirilip, farklı yukarı yönlü mesajlaşma hızlarının performans etkisi test edilmiştir.

64 Hz Radio Çevriminin Karşılaştırması

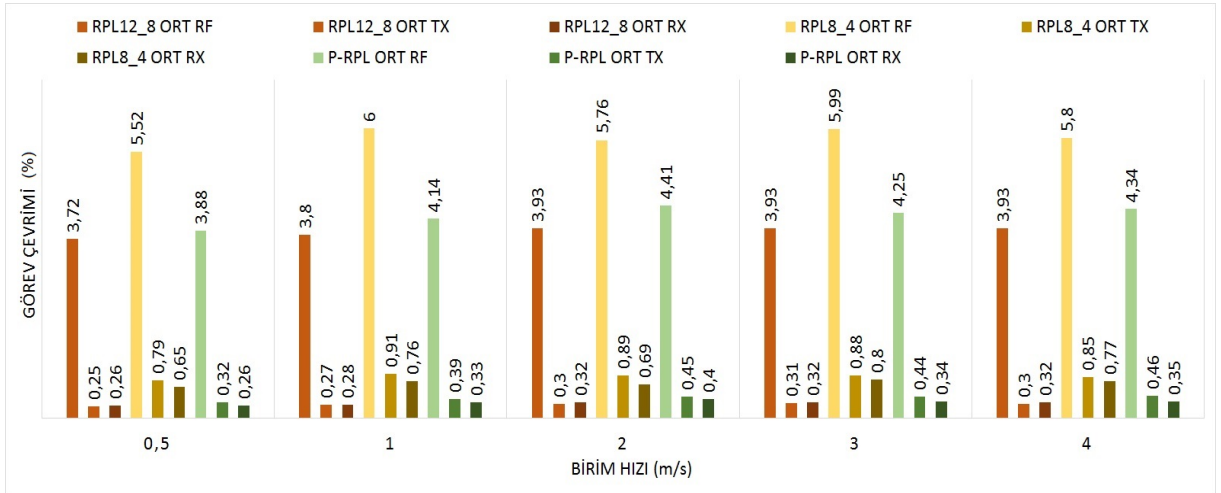
Testin bu aşamasında mesajlar hareketli birimden kök birime olacak şekilde yönlendirilmiştir. Şekil 6.4a ile görülebileceği gibi yürüme hızındaki hareketlilik altında RPL12_8 yaklaşık %50 paket gönderim oranına sahipken, yüksek mesajlaşma hızında RPL8_4 tatminkar sonuçlar vermektedir. Ancak mesajlaşma sıklığı azaldıkça RPL8_4 hareketliliğe karşı hızlı tepki gösterememiştir. Bunların yanında P-RPL test boyunca aynı paket gönderim oranını korumuştur. Bu gözlem *Phoenix Algoritmasının* tasarım amaçlarından biri olan uygulama



(a) Aşağı yönlü mesajlaşmada birim hızına karşılık paket gönderim oranı grafiği



(b) Aşağı yönlü mesajlaşmada birim hızına karşılık ortalama radyo görev çevrimi oranı grafiği



(c) Aşağı yönlü mesajlaşmada birim hızına karşılık ortalama radyo görev çevrimi oranı grafiği

Şekil 6.3: Aşağı yönlü mesajlaşmada birim hızı karşılaştırmaları

katmanından bağımsız geçiş gecikmesi ve paket gönderim oranına sahip olma özelliğini kanıtlamaktadır.

Enerji tüketimi açısından Şekil 6.4b incelendiğinde beklenildiği gibi P-RPL'nin en yüksek mesaj verimliliğine sahip olduğu görülebilmektedir. Ağ bazındaki enerji tüketimini görebilmek için Şekil 6.4c incelendiğinde ise P-RPL'nin yüksek paket gönderim oranına, test ortalaması bazında RPL12_8'e karşı aynı radyo kullanımı ile ulaştığı görülmektedir. Hareketli birimlerin mesaj gönderme verimlilikleri;

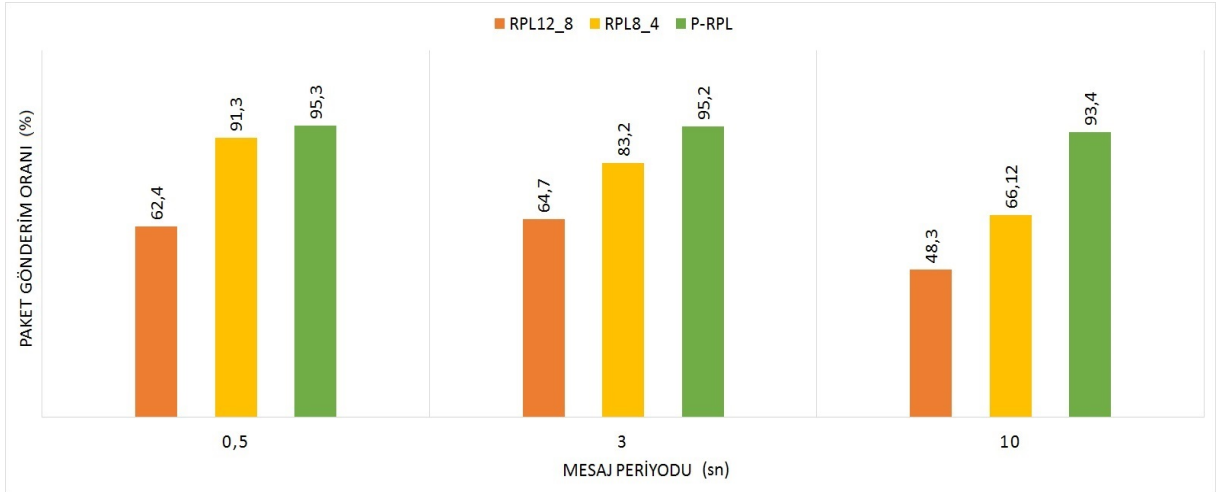
- RPL12_8 : 0,54
- RPL8_4 : 0,84
- P-RPL : 1,1

olarak hesaplanmaktadır.

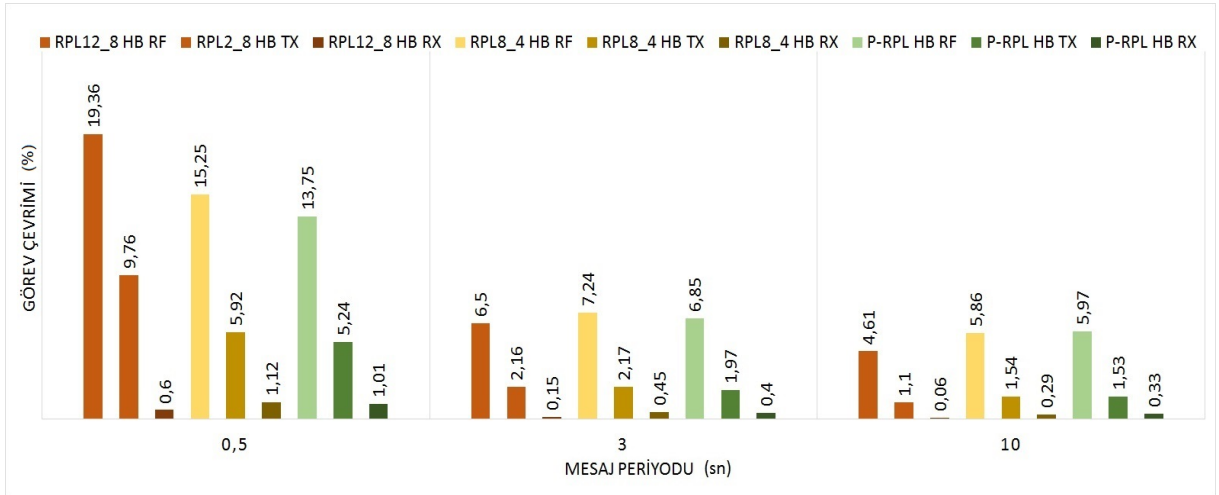
8 Hz Radyo Çevriminin Karşılaştırması

Testin bu aşamasında mesajlar hareketli birimden kök birime olacak şekilde yönlendirilmiştir. Bu testin diğer testlerden farkı radyo çevrim katmanının 64 Hz değil, 8 Hz olarak kanal kontrolü yapmasıdır. Bu düşük kanal kontrol hızı radyo kullanımını azaltırken, mesajların zaman zaman iletilmemesinden dolayı paket gönderim oranını azaltmaktadır. Bu özelliğinden ötürü mesajlaşmanın az olduğu, paket gönderiminin makul kabul edildiği, ağ bazında enerji tüketiminin azaltılmak istendiği uygulamalar için uygun bir kontrol hızıdır. Şekil 6.5a ile görülebileceği gibi yüksek mesaj gönderim hızlarında hareketsiz birimin bile paket gönderim oranı %30 civarlarındadır. Ancak mesaj gönderimi yavaşladığı zaman hareketsiz birim tarafından gönderilen bütün mesajlar yerine ulaşmaktadır. Bu şartlar altında bile P-RPL paket gönderim oranını hala uygulama katmanı hızından bağımsız tutabilmektedir. RPL8_4 yüksek kontrol mesajlaşması yaptığı için kullanımı kısıtlı olan radyo kanalını doldurmuş ve bundan dolayı paket gönderim oranını düşürmüştür.

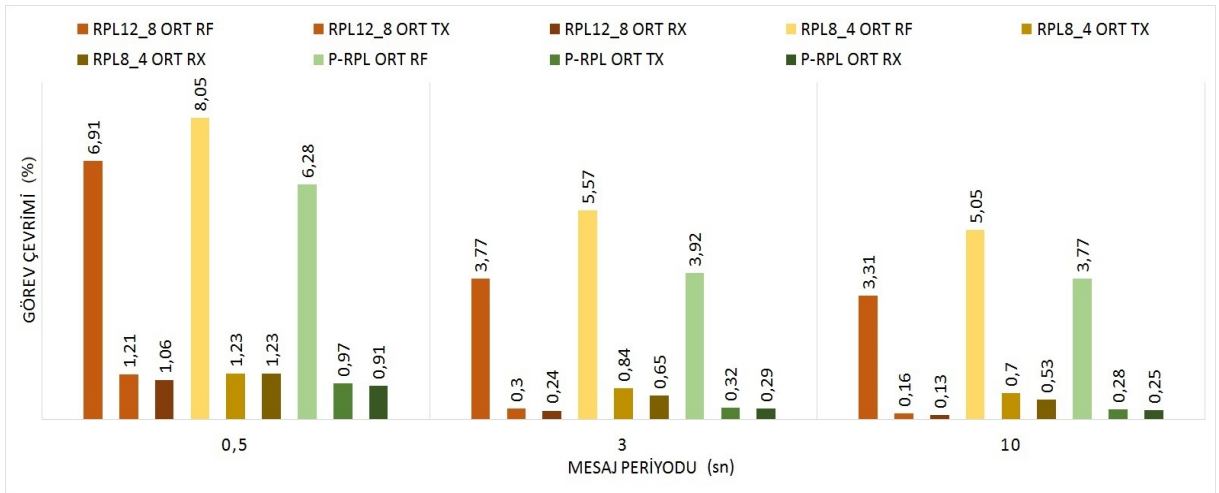
Enerji tüketimi açısından Şekil 6.5b incelendiğinde yüksek mesaj gönderim hızı altında paket kaybı yüksekliğinden dolayı *Phoenix Algoritması* gereksiz ebeveyn değiştirdiği için yüksek enerji tüketimi yapmıştır. Mesaj hızı azaldığı anda kanal, kontrol ve uygulama katmanı mesajlarının kullanılabilmesi serbestliğe ulaştığından P-RPL yeniden en yüksek performansa sahip birim haline gelmiştir. Ağ bazındaki ortalama enerji tüketimi Şekil 6.5c ile



(a) Yukarı yönlü mesajlaşmada mesajlaşma hızına karşılık paket gönderim oranı grafiği



(b) Yukarı yönlü mesajlaşmada mesajlaşma hızına karşılık hareketli birim radyo görev çevrimi oranı grafiği



(c) Yukarı yönlü mesajlaşmada mesajlaşma hızına karşılık ortalama radyo görev çevrimi oranı grafiği

Şekil 6.4: 64 Hz kanal kontrol hızı altında mesajlaşma kızı karşılaştırmaları

görülmektedir. Hareketli birim ağ içerisinde gezerken ebeveynleri ile yaptığı kontrol mesajları ortalama radyo kullanımını bir miktar arttırmıştır.

Sonuç olarak 8 Hz kanal kontrolü hızı altında P-RPL şuan için paket gönderim oranından taviz vermese de, enerji tüketimi yönünden geliştirilebilir konumdadır. Kontrol mesajı kaybı altında daha az sıklıkta ebeveyn değiştirmelidir. Bu testte ortalama mesaj verimlilikleri;

- RPL12_8 : 0,37
- RPL8_4 : 0,1
- P-RPL : 0,42

olarak bulunmuştur.

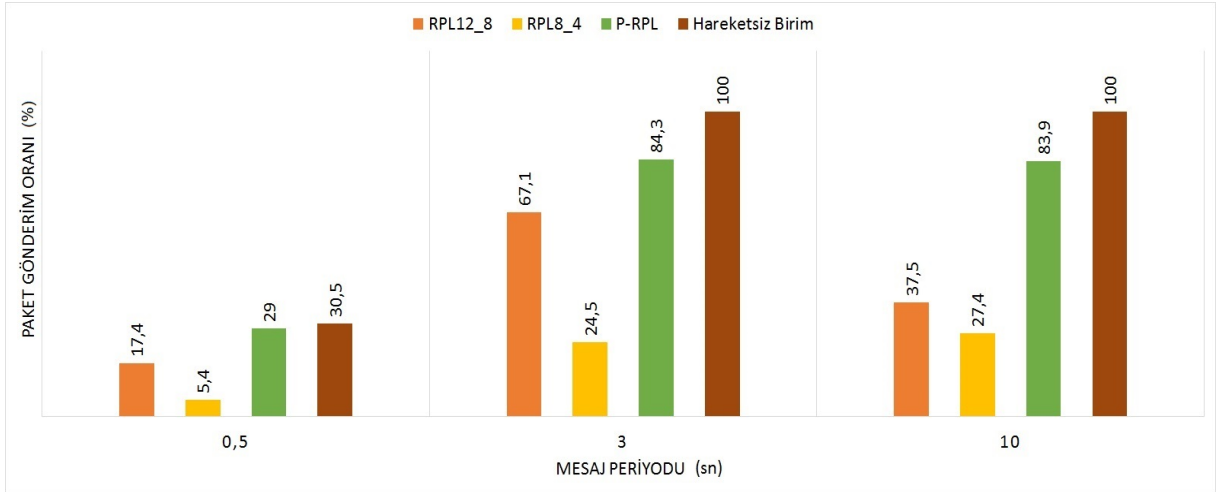
6.2.2.3 Çalışma Seçenekleri Karşılaştırmaları

Bu karşılaştırma Phoenix Algoritmasının *Güç Kontrolü* mekanizması ve *Durağanlık Algılama* mekanizmalarının karşılaştırmasıdır. Test boyunca radyo çevrim katmanı 64 Hz tutularak farklı birim hızı ve yukarı yönlü mesajlaşma hızı altında çalışma seçeneklerinin performansı karşılaştırılmıştır.

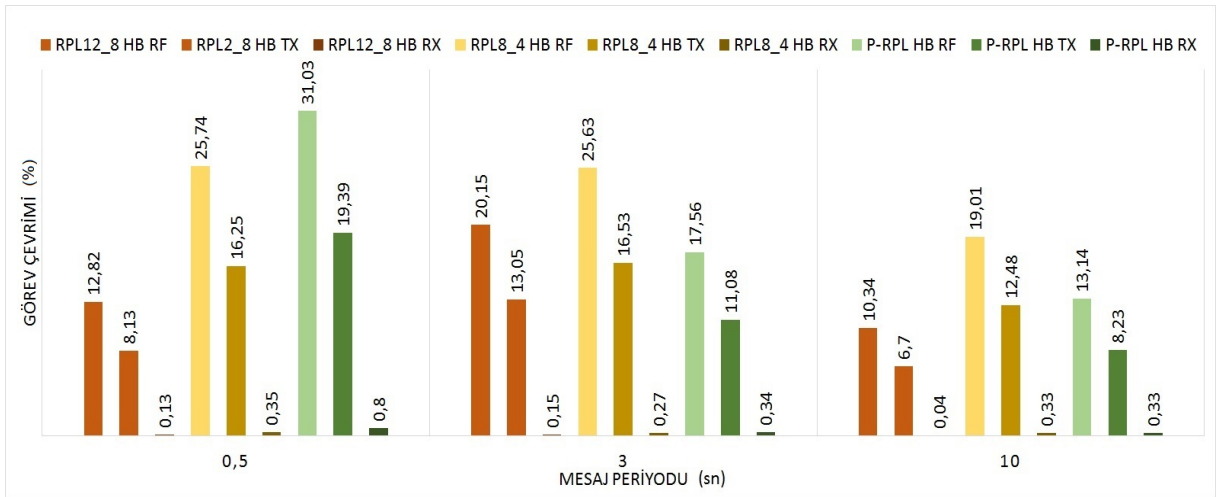
Mesaj Hızı Karşılaştırması

Testin bu aşamasında *Phoenix Algoritması* içerisinde yer alan *Güç Kontrolü* ve *Durağanlık Algılama* mekanizmalarının etkisi incelenecektir. Şekil 6.6a ile gösterilen paket gönderim oranı karşılaştırmasında beklenildiği gibi en yüksek oranın hiç bir mekanizmanın kullanılmadığı birim olduğu görülmektedir. *Güç Kontrolü* mekanizması ile kısılan güç, hareketli birimin anlık hareketlerinde ebeveyn birimin kapsama alanından çıkmasına sebebiyet vermektedir. *Durağanlık Algılama* mekanizması ise hareketli birimin durağan anlarında kontrol mesajlaşma sayısını azaltarak enerji tasarrufu yapmaktadır. Ancak azalan kontrol mesajı, hareketli birimin durağanlığının devamında başlayan anlık hareketlerinde ebeveyn geçişini geciktirmektedir. Geciken geçişlerde paket gönderim oranını düşürmektedir.

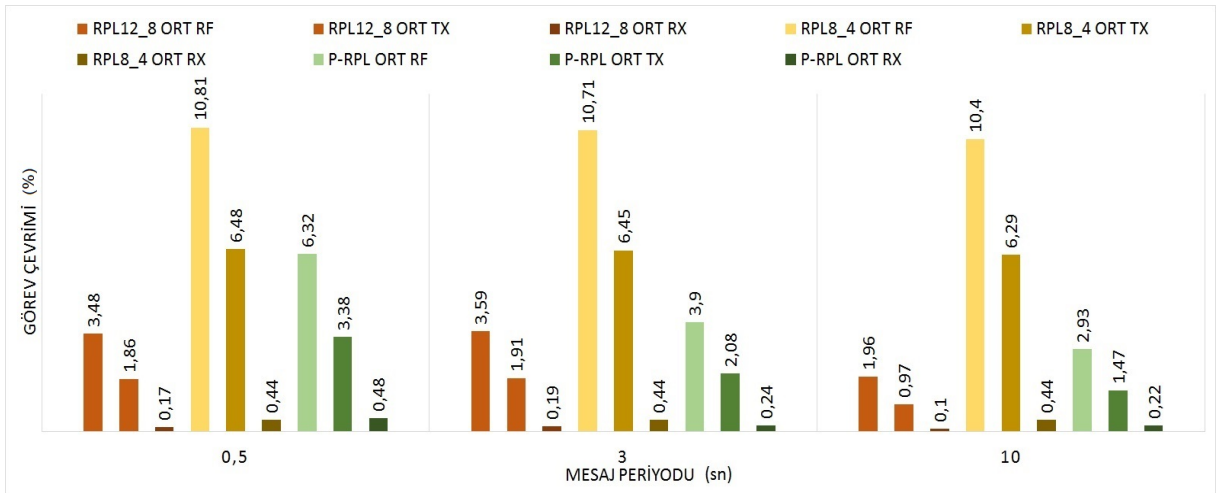
Enerji tüketimi açısından Şekil 6.6b incelendiğinde en yüksek radyo kullanımının *Güç Kontrol* mekanizması tarafından yapıldığı görülmektedir. Bunun sebebi kısılan gücün RSSI seviyesini düşürmesi ve buna bağlı olarak kontrol mesajı sıklığının artmasıdır. Ancak benzetim



(a) Yukarı yönlü mesajlaşmada mesajlaşma hızına karşılık paket gönderim oranı grafiği



(b) Yukarı yönlü mesajlaşmada mesajlaşma hızına karşılık hareketli birim radyo görev çevrimi oranı grafiği



(c) Yukarı yönlü mesajlaşmada mesajlaşma hızına karşılık ortalama radyo görev çevrimi oranı grafiği

Şekil 6.5: 8 Hz kanal kontrol hızı altında mesajlaşma hızı karşılaştırmaları

ortamında çıkış gücü ile radyo kullanımı arasında herhangi bir bağlantı bulunmadığı için verimlilik açısından doğru bir bilgi elde edilememektedir. *Durağanlık Algılama* mekanizması durağan durumlarda kontrol mesajlaşmasını azalttığı için test boyunca en düşük radyo kullanımına sahiptir. Bu da tasarım amacını doğrular niteliktedir. Şekil 6.6c ile ağ bazında ortalama enerji tüketimi görülmektedir. Burada mesajlaşmanın sık olduğu durumda *Güç Kontrolü* tarafından kısılan gücün ağ içerisindeki radyo kullanımını azalttığı görülmektedir. Bunun sebebi gönderilen mesajların daha az sayıda birim tarafından alınarak radyo kullanımını azaltmaktır.

Sonuç olarak benzetim testinde *Güç Kontrolü* mekanizmasının enerji tüketimine etkisi görülemez de tasarlanan mekanizmalar beklenildiği gibi çalışmaktadır.

Birim Hızı Karşılaştırması

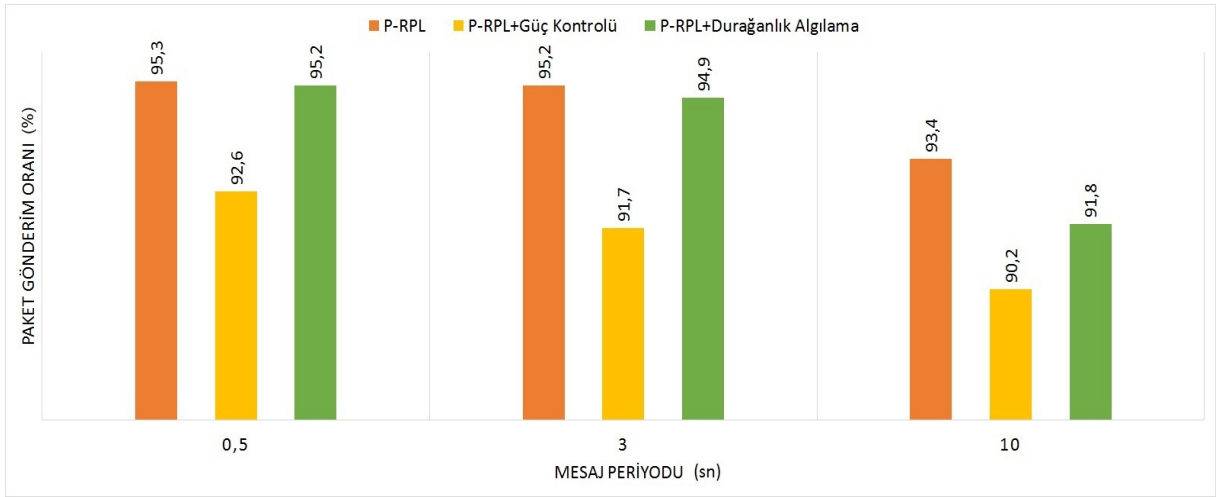
Bu aşamada da tasarlanan mekanizmaların değişik birim hızları altında çalışırılığının testi yapılmıştır. Çıkan sonuçlar bir önceki test ile aynı nedenlerle benzer değerlere sahiptir. Şekil 6.7a, Şekil 6.7b, Şekil 6.7c ile test sonuçları incelenebilmektedir.

6.3 Saha Testleri

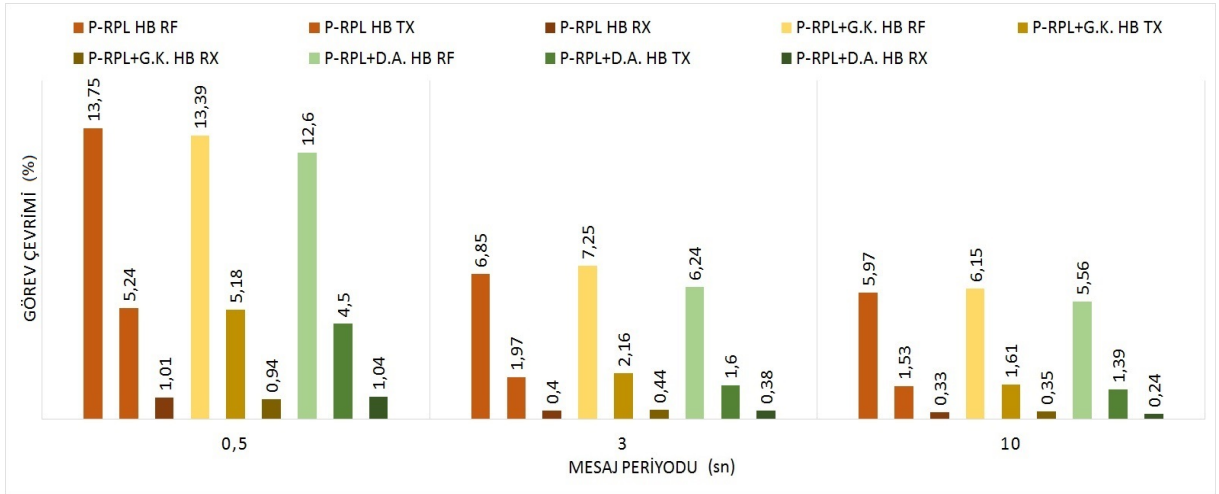
Tez kapsamında farklı mesajlaşma hızları ile toplam 18 adet test yapılmış olup sonuçları grafikler ile gösterilmiştir. Test sonuçlarını gösterebilmek için 2 ayrı grafik kullanılmıştır. İlk grafik başarılı paket gönderim oranını, ikinci grafik ise birimlerin harcadığı enerjiyi göstermektedir. Saha testlerinin amacı hem benzetim testlerinin doğruluğunu ölçmek hem de benzetim testlerinde ölçülemeyen enerji tüketimini ölçebilmektir. Gerçek testlerin benzetim testlerine göre uzun zaman almasından dolayı sadece en önemli görülen senaryolar test edilmiştir.

6.3.1 Saha Testleri Senaryosu

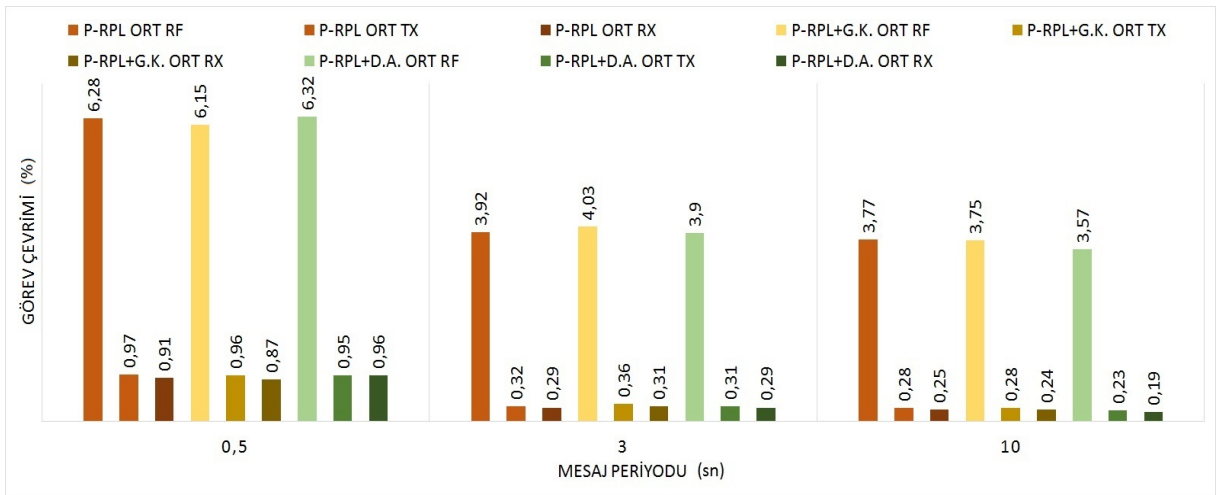
Tez kapsamında bahsi geçen birimlerin karşılaştırılması için bir saha testi senaryosu geliştirilmiştir. Bu senaryoda da amaç benzetim testlerinde olduğu gibi düzensiz hareket eden insan modelini test etmektir. Şekil 6.8 ile kullanılan senaryo görülebilmektedir. Senaryo 1



(a) Yukarı yönlü mesajlaşmada Phoenix çalışma seçeneklerine karşılık farklı mesajlaşma hızlarında paket gönderim oranı grafiği

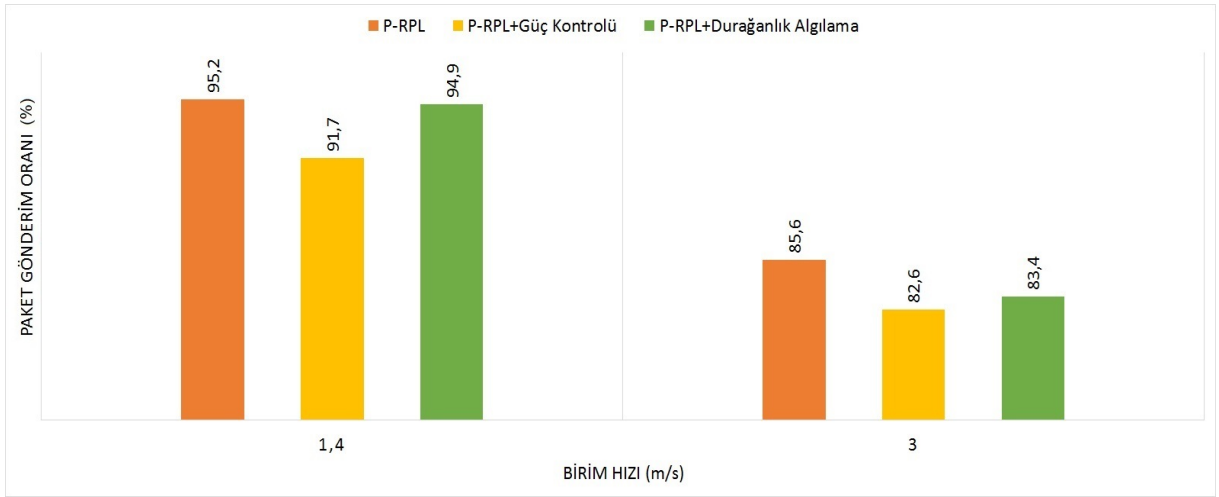


(b) Yukarı yönlü mesajlaşmada Phoenix çalışma seçeneklerine karşılık farklı mesajlaşma hızlarında hareketli birim radyo görev çevrimi oranı grafiği

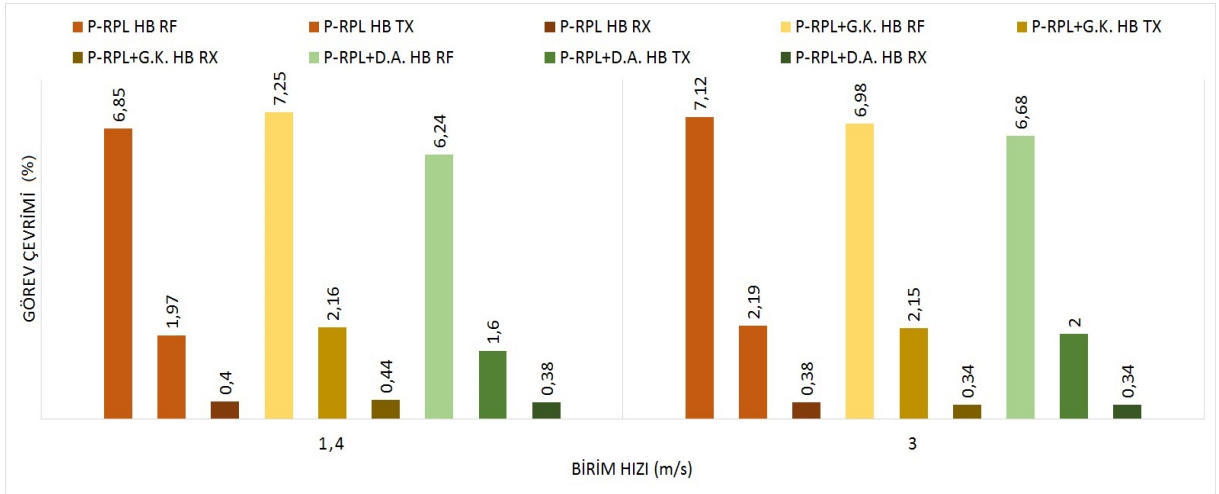


(c) Yukarı yönlü mesajlaşmada Phoenix çalışma seçeneklerine karşılık farklı mesajlaşma hızlarında ortalama radyo görev çevrimi oranı grafiği

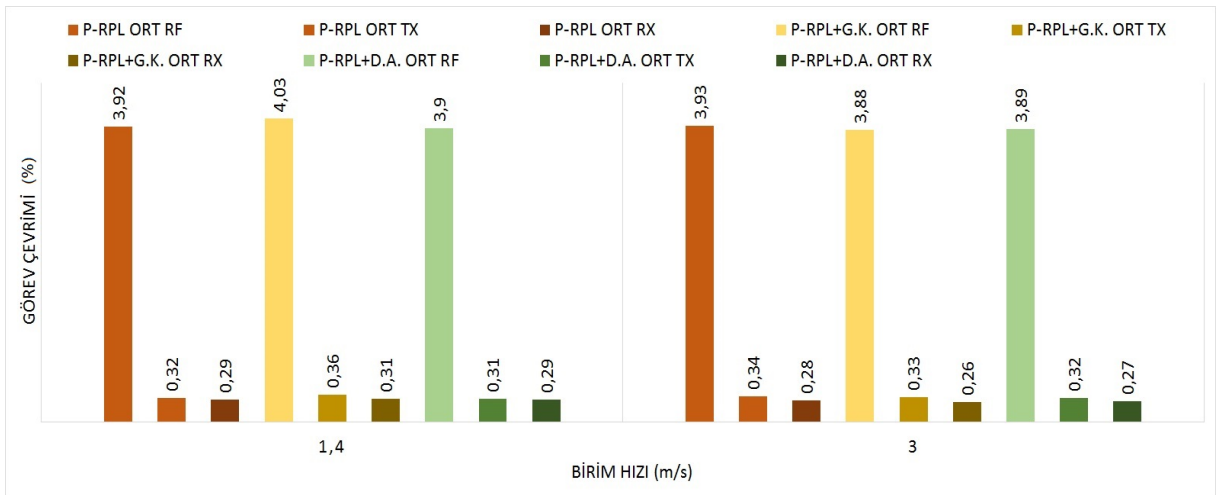
Şekil 6.6: Farklı mesajlaşma hızlarında Phoenix çalışma seçenekleri karşılaştırması



(a) Yukarı yönlü mesajlaşmada Phoenix çalışma seçeneklerine karşılık farklı birim hızlarında paket gönderim oranı grafiği



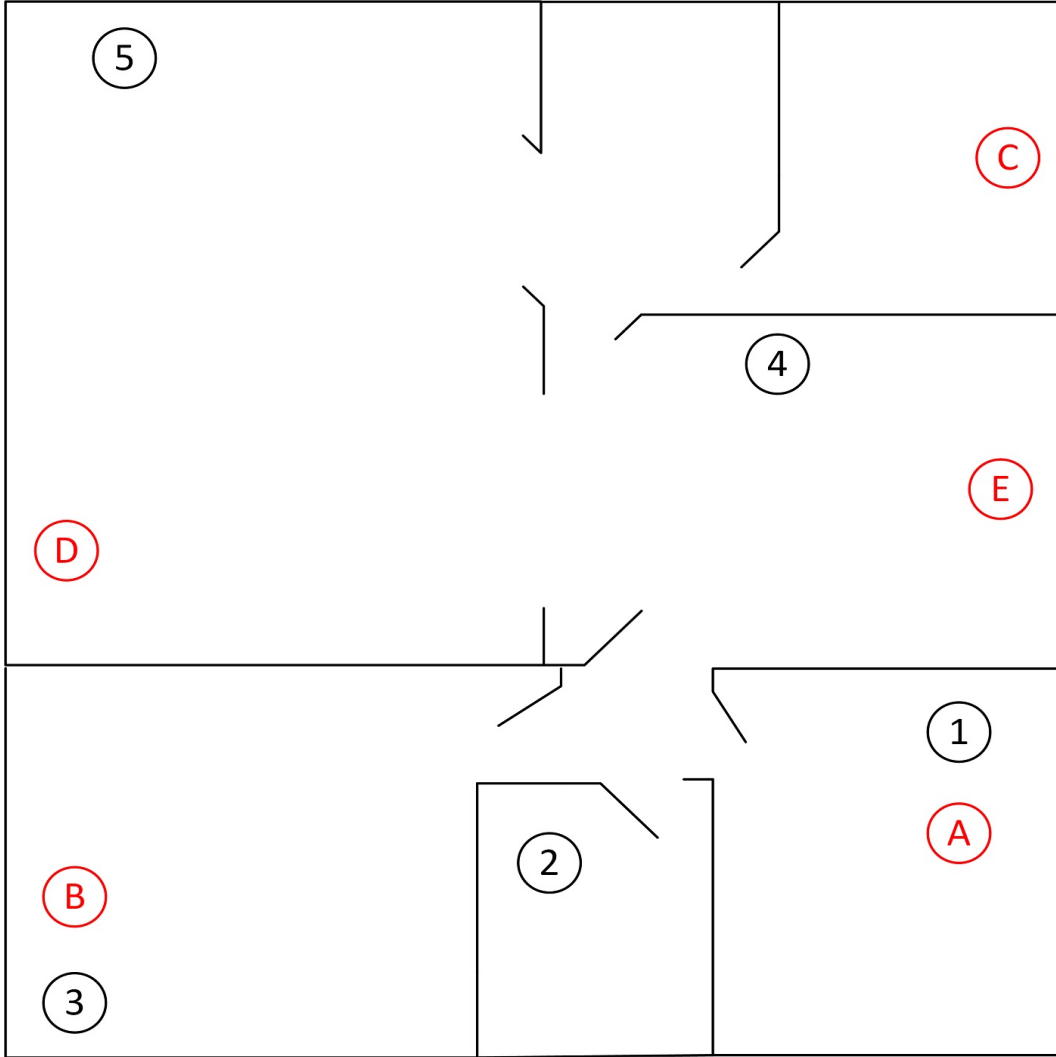
(b) Yukarı yönlü mesajlaşmada Phoenix çalışma seçeneklerine karşılık farklı birim hızlarında hareketli birim radyo görev çevrimi oranı grafiği



(c) Yukarı yönlü mesajlaşmada Phoenix çalışma seçeneklerine karşılık farklı birim hızlarında ortalama radyo görev çevrimi oranı grafiği

Şekil 6.7: Farklı birim hızlarında Phoenix çalışma seçenekleri karşılaştırması

kök (1 numaralı birim), 4 ebeveyn ve 1 hareketli birimden oluşmaktadır. Test durağan ve hareketliliğin olduğu anları içermektedir.



Şekil 6.8: Saha Testi Senaryosu

Senaryo adımları şunlardır:

1. Test çalışmaya başladığında test edilen birim ağa katılmadan önce, ağın yönlendirme tablolarını oluşturması için 15 dakika süre ile beklenildiği anıdır. 15 dakika sonunda hareketli birim ağa A noktasından katılarak kök birim ile aşağı ve yukarı yönlü haberleşmeye başlar. Gönderilen ilk 5 mesajın ardından hareketli birim B noktasına gider.
2. Hareketli birim B noktasına geldiğinde burada 1 dakika bekleyerek C noktasına doğru yoluna devam eder.

3. Hareketli birim **C** noktasına geldiğinde burada 1 dakika bekleyerek **D** noktasına doğru yoluna devam eder.
4. Hareketli birim **D** noktasına geldiğinde burada 1 dakika bekleyerek **E** noktasına doğru yoluna devam eder.
5. Hareketli birim **D** noktasına geldiğinde burada beklemeden **A** noktasına geri döner.

6.3.2 Saha Testleri Sonuçları

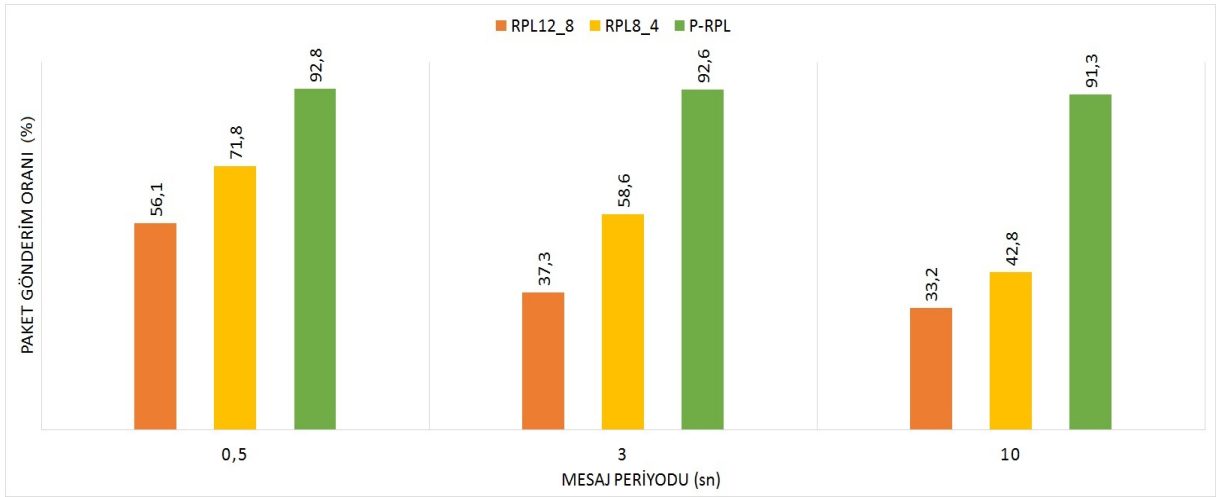
Bu karşılaştırmalarda birim hızı yavaş insan yürüyüş hızı olan saatte 2 kilometre olacak şekilde sabit tutulup, radyo çevrim katmanı 64 Hz olacak şekilde seçilmiştir. Testin amacı farklı mesajlaşma hızlarının performansa etkisini incelemektir.

6.3.2.1 Yukarı Yönlü Mesajlaşmada Mesajlaşma Hızı Karşılaştırması

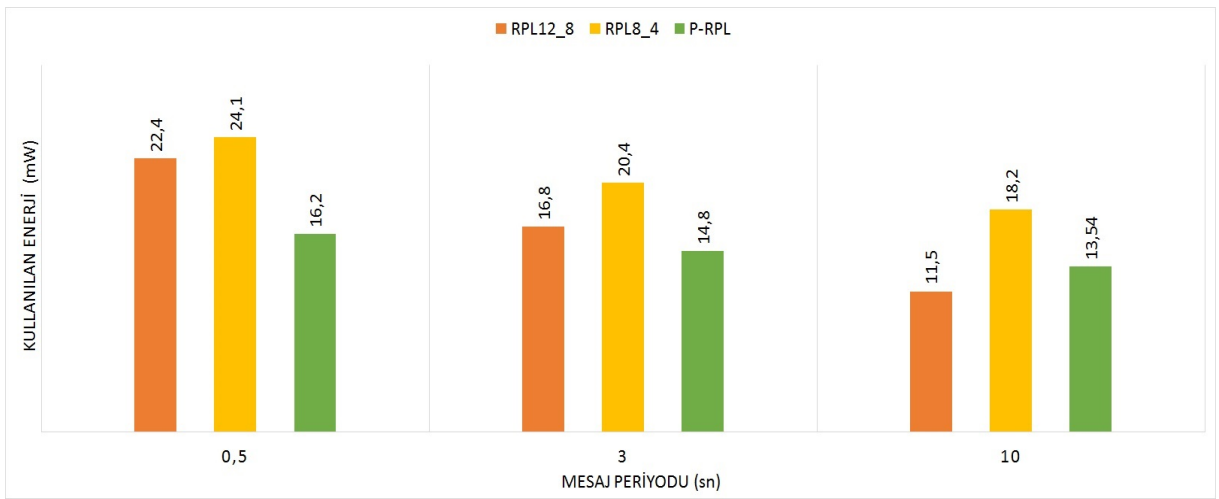
Testin bu aşamasında mesajlar hareketli birimden kök birime olacak şekilde yönlendirilmiştir. Şekil 6.9a ile görülebileceği gibi P-RPL uygulama katmanından bağımsız olarak hızlı ebeveyn geçişlerini gerçekleştirmiş ve buna bağlı olarak yüksek paket gönderim oranı değerlerini yakalamıştır. RPL8_4 yüksek kontrol mesajlaşması sayesinde RPL12_8'den daha iyi paket gönderim oranına sahip olsa da ikisinin değeri de azalan mesajlaşma ile düşmüş, test boyunca P-RPL'in altında kalmıştır.

Enerji tüketimi açısından Şekil 6.9b incelendiğinde birimlerin test boyunca kullandıkları ortalama güç tüketimi görülebilir. P-RPL testin büyük bir bölümünde en az güç tüketimine sahiptir. Bunun en önemli sebeplerinden bir tanesi benzetim testlerinde anlatıldığı gibi hızlı geçiş algoritması sayesinde sürekli aktif bir ebeveyn birime bağlı kalabilmektir. Gönderilen mesajlar tekrar edilmeden hedefine ulaşabilmektedir. Sonuç olarak toplamda daha az paket gönderildiğinden güç tüketimi de azalmaktadır. Bu etki mesajlaşmanın sık olduğu testlerde daha rahat görülebilmektedir.

Sonuç olarak geliştirilen Phoenix algoritması daha az enerji ile daha yüksek paket gönderim oranına sahiptir. Test boyunca yapılan gözlemler RPL12_8 ve RPL8_4'ün çok az ebeveyn değiştirdiğini, başarılı gönderilen paketlerin ise hareketli birimlerin eski ebeveynlerin kapsama alanına tekrar girildiği anda oluştuğunu, bundan dolayı farklı senaryolarda bu oranla-



(a) Yukarı yönlü mesajlaşmada mesajlaşma hızına karşılık paket gönderim oranı grafiği



(b) Yukarı yönlü mesajlaşmada mesajlaşma hızına karşılık ortalama güç tüketimi grafiği

Şekil 6.9: Yukarı yönlü mesajlaşmada mesajlaşma hızı karşılaştırmaları

rın çok daha fazla azalabileceğini göstermektedir. Hareketli birimlerin mesaj verimlilikleri kabaca paket gönderim oranı/tüketilen ortalama güç ile hesaplanabilir. Bu testin sonuçları aşağıdaki gibidir:

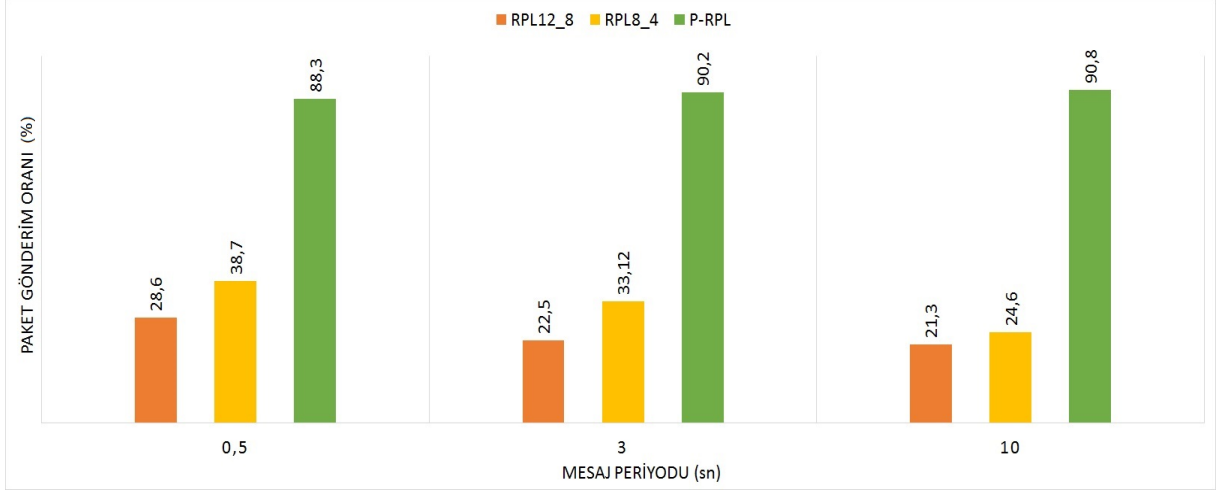
- RPL12_8 : 0,25
- RPL8_4 : 0,27
- P-RPL : 0,62

6.3.2.2 Aşağı Yönlü Mesajlaşmada Mesajlaşma Hızı Karşılaştırması

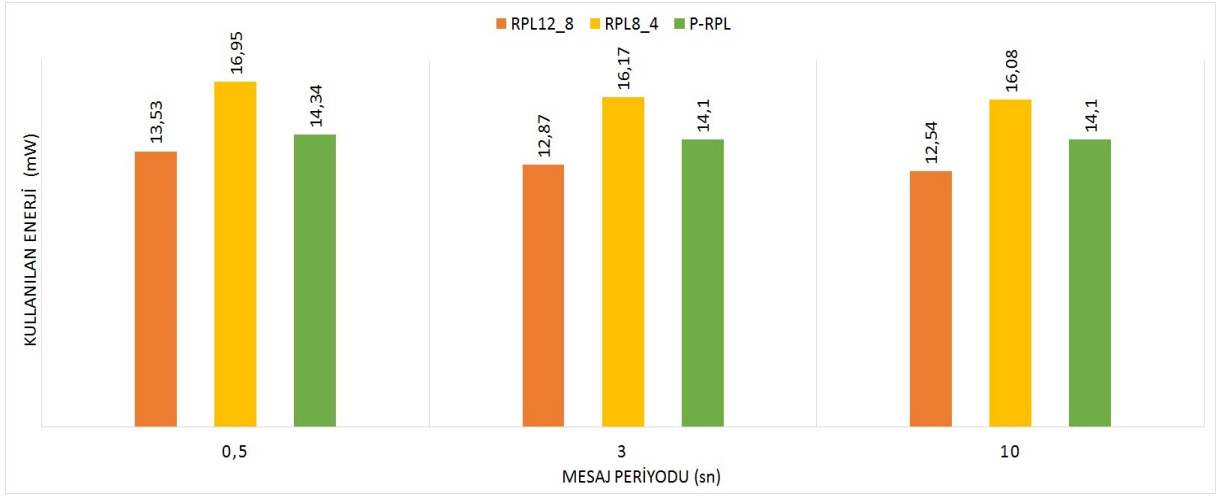
Testin bu aşamasında mesajlar kök birimden hareketli birime olacak şekilde yönlendirilmiştir. Şekil 6.10a ile görülebileceği gibi RPL12_8 ve RPL8_4 birimlerinin paket gönderim oranı kabul edilemeyecek seviyelerde iken, P-RPL %90 ortalama ile mesajlaşma hızından bağımsız şekilde yüksek bir gönderim oranına sahiptir.

Enerji tüketimi açısından Şekil 6.10b incelendiğinde birimlerin test boyunca kullandıkları ortalama güç tüketimi görülmektedir. P-RPL test boyunca RPL12_8'den %9 daha fazla güç tüketmiştir. Bunun sebebi *Phoenix Algoritması* kontrol mesajları ve kök birimden gelen mesajların alınmasıdır. Sonuçlar ve sebepler bir önceki test sonuçları ile paralellik gösterip mesaj verimlilikleri aşağıdaki gibidir:

- RPL12_8 : 0,18
- RPL8_4 : 0,19
- P-RPL : 0,63



(a) Aşağı yönlü mesajlaşmada mesajlaşma hızına karşılık paket gönderim oranı grafiği



(b) Aşağı yönlü mesajlaşmada mesajlaşma hızına karşılık ortalama güç tüketimi grafiği

Şekil 6.10: Aşağı yönlü mesajlaşmada mesajlaşma hızı karşılaştırmaları

7. SONUÇLAR

KSA'larda standartlaşma RPL protokolü ile büyük ivme kazanmıştır. RPL IP desteği sayesinde sensör birimlerinin var olan IP ağına kolaylıkla katılabilmesini sağlayarak yüzlerce uygulama alanının önünü açmıştır. Ancak uygulama alanları arttıkça RPL protokolünün de eksikleri ortaya çıkmaktadır. Bu sorunlardan en önemlilerinden birisi hareketliliktir. Hareketlilik, ağın kurulu yapısını bozarak yönlendirme tablolarını geçersiz kılmaktadır. Bu sorun gönderilen başarılı paket oranını düşürmekle beraber gönderilemeyen her paket için fazlaca gereksiz enerji tüketimine sebebiyet vermektedir. Fakat endüstri standardı olma yolundaki en büyük aday olan RPL yönlendirme protokolü içerisinde hareketliliği yönetecek mekanizmalar barındırmamaktadır. Bu eksikliği gidermek için birkaç çalışma yapılmış olup, yapılan bu çalışmalardan en yüksek başarıya ulaşanları MoMoRo [65] ve Smart-Hop [66] algoritmalarıdır. Ancak geliştirilen bu algoritmaların performansları algoritmanın geliştirildiği ağ katmanına bağlı olmayıp, uygulama katmanı tarafından gönderilen mesajlara bağlıdır. Bu sebeple tasarlanan bu geçiş algoritmaları farklı uygulama alanları için farklı ve yetersiz sonuçlar vermektedir. Örneğin azalan uygulama katmanı mesajlarında bu algoritmaların geçiş hızları ciddi biçimde düşerek yüksek paket kaybı yaratmaktadırlar.

Bu eksikliğin üzerine tez kapsamında kablosuz sensör ağlarının kısıtlamaları dikkate alınarak performansı uygulama katmanından bağımsız, hem yukarı yönlü, hem de aşağı yönlü mesajlar için, *Phoenix* adı altında hızlı geçiş algoritması tasarlanmıştır. Algoritma katmanlı yapıya uygun, kolay ayarlanabilir, RPL protokolü gibi herhangi bir fiziksel katmanda çalışabilecek şekilde tasarlanmıştır. Tasarımın amacı ise en az enerji kullanarak en yüksek başarılı paket gönderim oranına sahip olmaktır. Tasarlanan algoritma performansı incelenmek üzere farklı test senaryolarında çalıştırılmıştır. Testlerde karşılaştırma olarak RPL12_8 isimli standart *Trickle Zamanlayıcısı* değerlerine sahip birim ile çok daha fazla sayıda RPL kontrol mesajı kullanan RPL8_4 birimi kullanılmıştır. Karşılaştırmalar hem benzetim ortamında, hem de tez kapsamında üretilen geliştirme kartı aracılığı ile gerçek düzenekler üzerinde tekrarlanmıştır.

7.1 Test Sonuçları

Phoenix Algoritmasının performans testleri öncelikle hızlı ve detaylı testlerin yapılmasına imkan veren benzetim ortamında yapılmıştır. Algoritmanın farklı uygulamalardaki kullanılabilirliğini ölçmek için çeşitli senaryolar oluşturulmuştur. Farklı birim hızları, mesajlaşma hızları ve yönleri, radyo kanal kontrol hızları ile karşılaştırmalı testler yapılmıştır. Ayrıca *Phoenix Algoritmasının* içerdiği özelliklerden olan *Durağanlık Algılama* ve *Güç Kontrol* mekanizmaları incelenmiş, algoritmanın bu mekanizmalar vasıtası ile ek enerji tasarrufu yapabileceği gösterilmiştir. Yapılan benzetim testleri sonucunda algoritmanın tasarlanma amaçları doğrulanmıştır. Phoenix Algoritması hareketlilik altında karşılaştırılan diğer birimlere göre çok daha yüksek başarılı paket gönderim oranına sahiptir. Gerçekleştirilen testlerde yukarı yönlü mesajlarda Phoenix algoritması 4 kata kadar, aşağı yönlü mesajlarda ise 8 kata kadar daha yüksek paket gönderim oranına sahip olmuştur. Dahası bu orana enerji tüketimini bazı senaryolarda çok az arttırarak, bazılarında ise standart RPL protokolünden bile daha az enerji harcayarak ulaşmıştır.

Sonuç olarak gönderilen mesajların hem gönderim oranı arttırılmış, hem de gönderilen mesaj başına harcanan enerji azaltılmıştır. Benzetim testlerinin devamında yapılan gerçek testler de benzetim testleri ile paralel sonuçlar vermiştir. Bu paralellikten yola çıkarak saha testlerinde gerçekleştirilemeyen senaryoların, saha testi sonuçları da yüksek doğrulukla tahmin edilebilir olmuştur. Geliştirilen algoritmanın sadece benzetim ortamında kalmayıp, gerçek uygulamalarda da başarılı bir şekilde çalışabileceğini kanıtlamıştır.

7.2 Yapılacak Çalışmalar

Phoenix algoritmasının bir takım eksiklikleri vardır. Şu an hareketli birimler başka hareketli birimlerin ebeveyn birimi olamamaktadır. Bunun sebebi hareketli birimlerin uygulamada en uç birim olarak kullanılmasıdır. Ek olarak bir diğer eksik ise hareketli birimlerin *Geçiş Evresinde* ebeveyn değiştirirken ortamdaki ebeveynlerin en iyisine bağlanmamasıdır. Sebebi geçiş süresini uzatmamak için ilk cevap gönderen ebeveyn birime bağlanmasıdır. Bu soruna çözüm olarak *Geçiş Evresine* girilip, daha geçiş gerçekleşmeden kapsama alanında bulunan ebeveyn birimlerden bilgi toplanabilir. Toplanan bu bilgi geçiş karar verildiği an kullanılıp belirlenen bir kıstasa göre en iyi ebeveyn birime bağlanmasını sağlayabilir.

KAYNAKLAR

- [1] C.E. Perkins and E.M. Royer. Ad-hoc on-demand distance vector routing. In *wmcsa*, page 90. Published by the IEEE Computer Society, 1999.
- [2] T. Clausen, P. Jacquet, C. Adjih, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L. Viennot. Optimized link state routing protocol (OLSR), RFC 3626. 2003.
- [3] W.R. Heinzelman, J. Kulik, and H. Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 174-185. ACM, 1999.
- [4] MJ. Handy, M. Haase, and D. Timmermann. Low energy adaptive clustering hierarchy with deterministic cluster-head selection. In *Mobile and Wireless Communications Network, 2002. 4th International Workshop on*, pages 368-372. IEEE, 2002.
- [5] IETF. The Internet Engineering Task Force IETF. IETF, <http://www.ietf.org/>, Eriřim: 14.02.2017
- [6] The ROLL Design Team. Charter of the ROLL working group. IETF, Description of Working Group, <http://datatracker.ietf.org/wg/roll/charter/>, Eriřim: 15.02.2017
- [7] The 6lowpan Design Team. Charter of the 6lowpan working group. IETF, Description of Working Group, <http://datatracker.ietf.org/wg/6lowpan/charter/>, Eriřim: 15.02.2017
- [8] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," RFC 6550 (Proposed Standard), 2012.
- [9] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki - a lightweight and flexible operating system for tiny networked sensors," in *Local Computer Networks, 2004. 29th Annual IEEE International Conference*, pp. 455–462, 2004.
- [10] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-Level Sensor Network Simulation with COOJA," pp. 641–648, 2006.
- [11] K. Romer and F. Mattern. The design space of wireless sensor networks. *Wireless Communications, IEEE*, 11(6):54-61, 2004.
- [12] J. Yick, B. Mukherjee, and D. Ghosal. Wireless sensor network survey. *Journal of Computer Networks*, Volume 52(Number 12):292-2330, 2008.
- [13] G. Wener-Allen, K. Lorincz, M. Ruiz, O. Marcillo, J. Johnson, J. Lees, and M. Walsh, "Deploying a wireless sensor network on an active volcano. data-driven applications in sensor networks," *IEEE Internet Computing*, vol. 2, pp. 18–25, 2006.

- [14] G. Liu, R. Tan, R. Zhou, G. Xing, W.-Z. Song, and J. M. Lees, "Volcanic earthquake timing using wireless sensor networks," in Proceedings of the 12th international conference on Information processing in sensor networks, pp. 91–102, ACM, 2013.
- [15] W.-Z. Song, R. Huang, M. Xu, A. Ma, B. Shirazi, and R. LaHusen, "Air-dropped sensor network for real-time high-fidelity volcano monitoring," in Proceedings of the 7th international conference on Mobile systems, applications, and services, pp. 305–318, ACM, 2009.
- [16] A. Hasler, I. Talzi, J. Beutel, C. Tschudin, and S. Gruber. Wireless sensor networks in permafrost research-concept, requirements, implementation and challenges. In Proceedings of the 9th International Conference on Permafrost (NICOP 2008), volume 1, pages 669-674, 2008.
- [17] H. Alemdar and C. Ersoy, "Wireless sensor networks for healthcare: A survey," *Computer Networks* 54, 2688–2710, 2010.
- [18] T. Brennan, L. Leape, N. Laird, L. Hebert, A. Localio, A. Lawthers, J. Newhouse, P. Weiler, and H. Hiatt, "Incidence of adverse events and negligence in hospitalized patients," *New England journal of medicine* 324, 370–376, 1991.
- [19] A. Initiative, "National Patient Safety Goals," *Hospital Pharmacy* 38, 490–496, 2008.
- [20] R. Wilson et al., "The quality in Australian health care study," *Medical Journal of Australia* 163, 458–471, 1995.
- [21] T. Gao, D. Greenspan, M. Welsh, R. Juang, and A. Alm, "Vital signs monitoring and patient tracking over a wireless network," in Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the, pp. 102–105, IEEE, 2006.
- [22] Darwish, A., Hassanien, A. E. *Wearable and Implantable Wireless Sensor Network Solutions for Healthcare Monitoring and Sensors*. MDPI AG. 2011.
- [23] W. Huiyong, W. Jingyang, and H. Min, "Building a smart home system with WSN and service robot," in Measuring Technology and Mechatronics Automation (ICMTMA), 2013 Fifth International Conference on, pp. 353–356, IEEE, 2013.
- [24] P. Waide, J. Ure, G. Smith, and B. Bordass, "The scope for energy and CO2 savings in the EU through the use of building automation technology" final report, Waide Strategic Efficiency, 2013.
- [25] K. Jaafar and M. K. Watfa, "Sensor networks in future smart rotating buildings," in Consumer Communications and Networking Conference (CCNC), 2013 IEEE, pp. 962–967, IEEE, 2013.
- [26] Tao F., Zuo Y., Xu L. D., Zhang L. "IoT-Based intelligent perception and access of manufacturing resource toward cloud manufacturing", *IEEE Transactions on Industrial Informatics*, pp. 1547-1557, IEEE, 2014

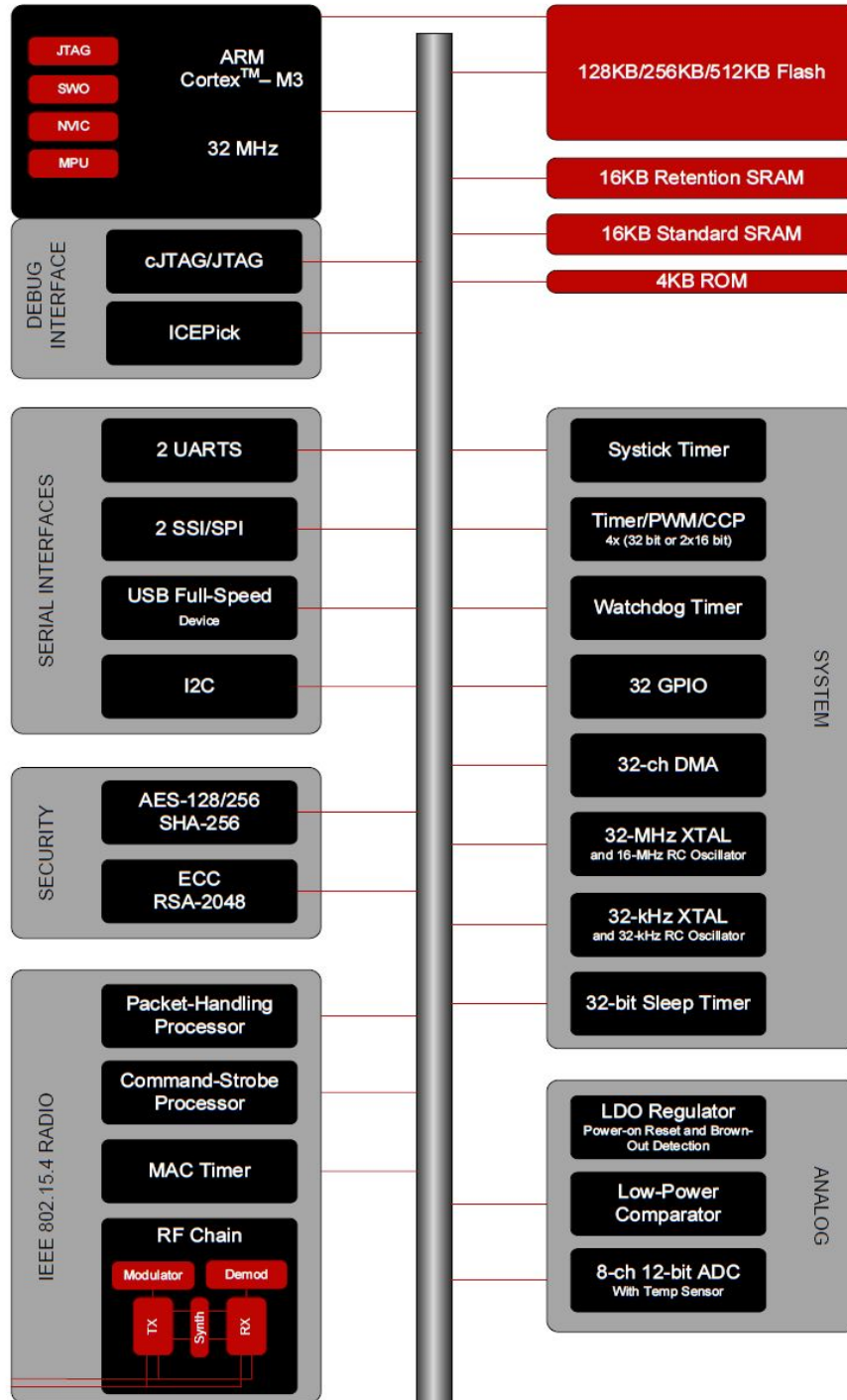
- [27] D. Bijwaard, W. Kleunen, P. Havinga, L. Kleiboer, and M. Bijl. Industry: using dynamic wsns in smart logistics for fruits and pharmacy. In *SenSys*, pages 218–231, 2011.
- [28] H.-J. Korber, H. Wattar, and G. Scholl. Modular wireless real-time sensor/actuator network for factory automation applications. *Industrial Informatics, IEEE Transactions on*, 3(2):111 –119, 2007.
- [29] M. Li and Y. Liu. Underground coal mine monitoring with wireless sensor networks. *ACM Trans. Sen. Netw.*, 5(2):10:1–10:29, 2009.
- [30] Ishfaq Ahmad, Khalil Shah. "Military Applications using Wireless Sensor Networks: A survey." *International Journal of Engineering Science and Computing* 7039-7043 ISSN: 23213361, 2016.
- [31] P. S. Cannon and C. R. Harding, "Future military wireless solutions," Ch. 8 in *Wireless Communications: The Future*, Editor William Webb, John Wiley and Sons, 2007.
- [32] M. Winkler, K.-D. Tuchs, K. Hughes, and G. Barclay, "Theoretical and practical aspects of military wireless sensor networks" In *Journal of Telecommunications and Information Technology*, pp. 37-45, 2008.
- [33] A. Grilo, R. Silva, P. Nunes, J. Martins, and M. Nunes, "Wireless sensor networking support to military operations on urban terrain," 12th International Command and Control Technology Symposium (ICCRTS'07), Command and Control Research Program (CCRP), 2007.
- [34] Chipcon Systems GmbH. CC1020 Single Chip Low Power RF Transceiver for Narrowband Systems Datasheet. Technical report. http://cst.mi.fu-berlin.de/projects/ScatterWeb/moduleComponents/CC1020_Data_Sheet_1_7.pdf, Erişim: 03.03.2017
- [35] Germany ScatterWeb GmbH Berlin. MSB: modular sensor board fact sheet v1.0. Technical report, ScatterWeb GmbH. <http://www.scatterweb.com/content/downloads/datasheets/fact-sheet-msb430-v1.0-en.pdf>, Erişim: 02.03.2017.
- [36] The ROLL Design Team. Reactive Discovery of Point-to-Point Routes in Low Power and Lossy Networks. <https://tools.ietf.org/html/draft-ietf-roll-p2p-rpl-01>, Erişim: 12.03.2017.
- [37] P. Levis, E. Brewer, D. Culler, D. Gay, S. Madden, N. Patel, J. Polastre, S. Shenker, R. Szewczyk, and A. Woo. "The emergence of a networking primitive in wireless sensor networks", *Communications of the ACM*, Volume 51(Number 7), pp. 99-106, 2008.
- [38] Internet Engineering Task Force (IETF) Request for Comments: 6206, The Trickle Algorithm ISSN: 2070-1721
- [39] H. Frazier. The 802.3 z gigabit Ethernet standard. *Network, IEEE*, 12(3), pp. 6-7, 2002.

- [40] K. Akkaya and M. Younis. A survey on routing protocols for wireless sensor networks. *Journal of Ad Hoc Networks*, Volume 3(Number 3), pp. 325-349, 2005.
- [41] IPSO Alliance. IP for Smart Objects (IPSO) Alliance Charter. <http://www.ipsoalliance.org/Pages/Charter.php>, Erişim: 10.03.2017.
- [42] The ROLL Design Team. Building Automation Routing Requirements in Low-Power and Lossy Networks, RFC 5867. <https://datatracker.ietf.org/doc/draft-ietf-roll-building-routing-reqs/> Erişim: 10.03.2017.
- [43] The ROLL Design Team. Home Automation Routing Requirements in Low-Power and Lossy Networks, RFC 5826. <https://datatracker.ietf.org/doc/draft-ietf-roll-home-routing-reqs/> Erişim: 10.03.2017.
- [44] The ROLL Design Team. Industrial Routing Requirements in Low-Power and Lossy Networks, RFC 5673. <https://datatracker.ietf.org/doc/draft-ietf-roll-indus-routingreqs/> Erişim: 10.03.2017.
- [45] The ROLL Design Team. Routing Requirements for Urban Low-Power and Lossy Networks, RFC 5548. <https://datatracker.ietf.org/doc/draft-ietf-roll-urban-routingreqs/> Erişim: 10.03.2017.
- [46] J. Hui, Ed., Arch Rock Corporation, P. Thubert, Cisco, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks", RFC 6282, 2011.
- [47] <https://www.iso.org/home.html>, Erişim: 03.04.2017
- [48] H. Zimmermann, N. Naffah, "On open systems architecture", Proc. ICC, pp. 669-674, 1978-Sept.
- [49] Wireless Medium Access Control (MAC) and physical layer (PHY) specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)." IEEE 802.15.4 Standard. Part 15.4 (2006).
- [50] T. Henriksson, Liu Dake, "Implementation of fast CRC calculation", Proceedings of the Design Automation Conference (ASP-DAC 2003), pp. 563-564, Jan.2003.
- [51] <http://www.contiki-os.org/hardware.html>, Erişim: 03.04.2017
- [52] Kin-Lu Wong, "Planar Antennas for Wireless Communication," Published by John Wiley and Sons, Inc., Chapter: 2, pp. 26-65, 2003.
- [53] O. Gaddour and A. Koubi, "Survey RPL in a nutshell: A survey" *Computer Networks* 56, pp. 3163–3178 (2012).
- [54] J. P. Vasseur, R. Kelsey, R. Struik, P. Levis, T. Winter, A. Brandt, J. Hui, K. Pister, T. Clausen, and P. Thubert, "RPL: IPv6 Routing Protocol for Low power and Lossy Networks." [Online]. Available: <http://tools.ietf.org/html/draft-ietf-roll-rpl-19>, Erişim: 07.04.2017.

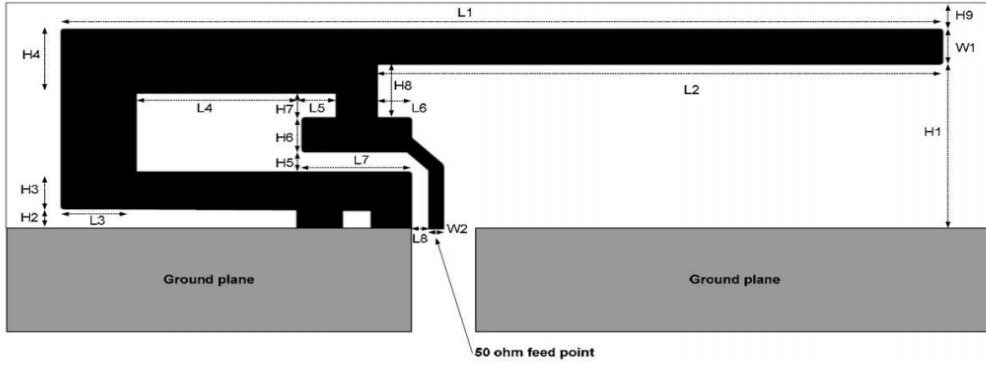
- [55] P. Levis, T. Clausen, J. Hui, O. Gnawali, and J. Ko, "The Trickle Algorithm," RFC 6206, 2011.
- [56] Serhan Dagtas, Yuri Natchetoi, and Huaigu Wu. "An integrated wireless sensing and mobile processing architecture for assisted living and healthcare applications.", In 1st ACM SIGMOBILE international workshop on Systems and networking support for healthcare and assisted living environments, pp. 70–72. ACM, 2007.
- [57] Konrad Lorincz, David J Malan, Thaddeus RF Fulford-Jones, Alan Nawoj, Antony Clavel, Victor Shnayder, Geoffrey Mainland, Matt Welsh, and Steve Moulton. "Sensor networks for emergency response: challenges and opportunities." *Pervasive Computing*, IEEE, 3(4), pp. 16–23, 2004.
- [58] Jerome Harri, Fethi Filali, and Christian Bonnet. "Mobility models for vehicular ad hoc networks: a survey and taxonomy. *Communications Surveys in Tutorials*", IEEE, 11(4), pp. 19–41, 2009.
- [59] Christian Schindelhauer. "Mobility in wireless networks." In *Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM)*, pp. 100–116. Springer, 2006.
- [60] Sunghyun Cho, E.W. Jang, and J.M. Cioffi. Handover in multihop cellular networks. *IEEE Communications Magazine*, 47(7), 2009.
- [61] I. Ramani and S. Savage. "Syncscan: practical fast handoff for 802.11 infrastructure networks." In *IEEE Conference on Computer Communications (INFOCOM)*, 2005.
- [62] J Tripathi et al., "A Performance Evaluation Study of RPL: Routing Protocol for Low power and Lossy Networks" in *proc. of the 44th IEEE Annual Conference on Information Sciences and Systems (CISS)*, 2010.
- [63] K. C. Lee et al., "RPL Under Mobility" in *proc. of the IEEE Consumer Communications and Networking Conference (CCNC)*, 2012.
- [64] I.E. Korbi et al., "Mobility Enhanced RPL for Wireless Sensor Networks" in *proc. of the 3rd International Conference on the Network of the Future (NOF)*, 2012.
- [65] J. Ko, M. Chang, "Momoro: Providing mobility support for low-power wireless applications" *Systems Journal*, IEEE 99, pp. 1–10, 2014.
- [66] H. Fotouhi, M. Zuniga, M. Alves, A. Koubaa, P. Marron, "Smart-hop: A reliable handoff mechanism for mobile wireless sensor networks" *EWSN*, 2012.
- [67] C. Gomez, P. Salvatella, O. Alonso, and J. Paradells, "Adapting AODV for IEEE 802.15.4 mesh sensor networks: Theoretical discussion and performance evaluation in a real environment" *IEEE WoWMoM*, p. 170, 2006.
- [68] H. Fotouhi, D. Moreira, M. Alves "Mobile IoT: smart-HOP over RPL" *Technical Report in CISTER-TR-140709*, 2014.

- [69] J. Ko and A. Terzis, "Power control for mobile sensor networks: An experimental approach," in *Proceeding IEEE SECON*, pp. 1–9, 2010.
- [70] J. Sheu, K. Hsieh, Y. Cheng, "Distributed Transmission Power Control Algorithm for Wireless Sensor Networks", *Journal of Information Science and Eengineering* 25, pp. 1447-1463, 2009.
- [71] S. Lin, J. Zhang, G. Zhou, L. Gu, T. He, and J. A. Stankovic, "ATPC: Adaptive transmission power control for wireless sensor networks," in *Proceedings of ACM Conference on Embedded Networked Sensor Systems*, pp. 223-236, 2006.
- [72] H. Kin, J. Paek, "Transmission Power Control in IPv6 Routing Protocol for Low-Power Wireless Network", *Sensys'16*, pp. 14–16, 2016
- [73] CC2538 Çip Ailesi Dokümanı, "<http://www.ti.com/lit/ds/symlink/cc2538.pdf>" Erişim: 23.09.2017.
- [74] Texas Instruments DN007 Baskı Devre Anten Tasarım Ölçüleri, "<http://www.ti.com/lit/an/swru120c/swru120c.pdf>" Erişim: 23.09.2017.
- [75] Texas Instruments DN007 Baskı Devre Anten Frekans Cevabı, "<http://www.ti.com/lit/an/swra328/swra328.pdf>" Erişim: 23.09.2017.

EKLER

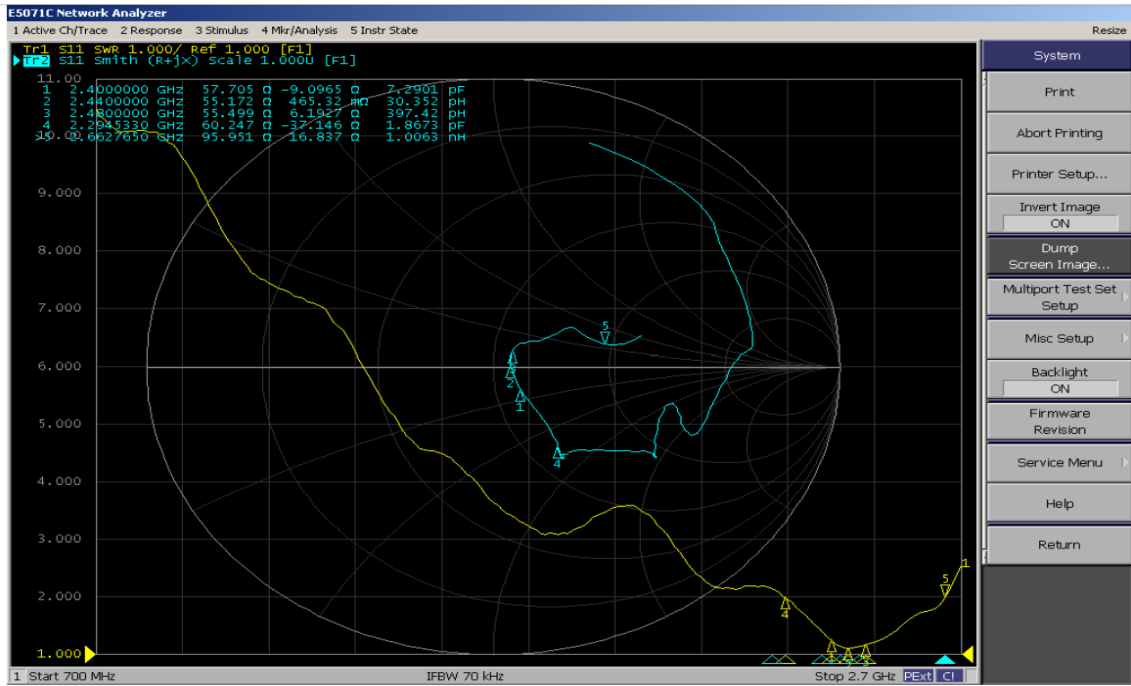


Ek E.1: CC2538 Çipinin İç Yapısı [73]

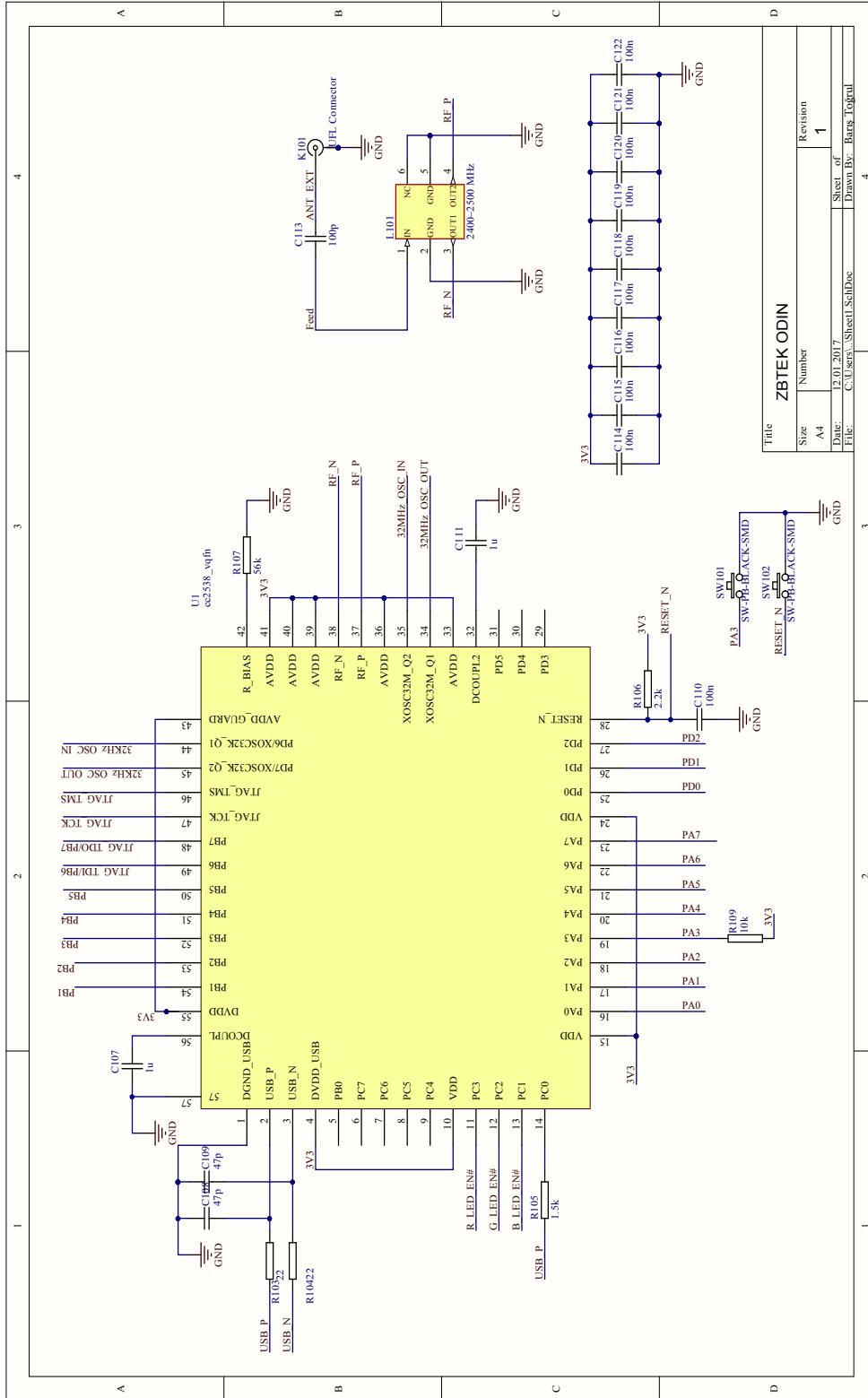


H1	5.70 mm	W2	0.46 mm
H2	0.74 mm	L1	25.58 mm
H3	1.29 mm	L2	16.40 mm
H4	2.21 mm	L3	2.18 mm
H5	0.66 mm	L4	4.80 mm
H6	1.21 mm	L5	1.00 mm
H7	0.80 mm	L6	1.00 mm
H8	1.80 mm	L7	3.20 mm
H9	0.61 mm	L8	0.45 mm
W1	1.21 mm		

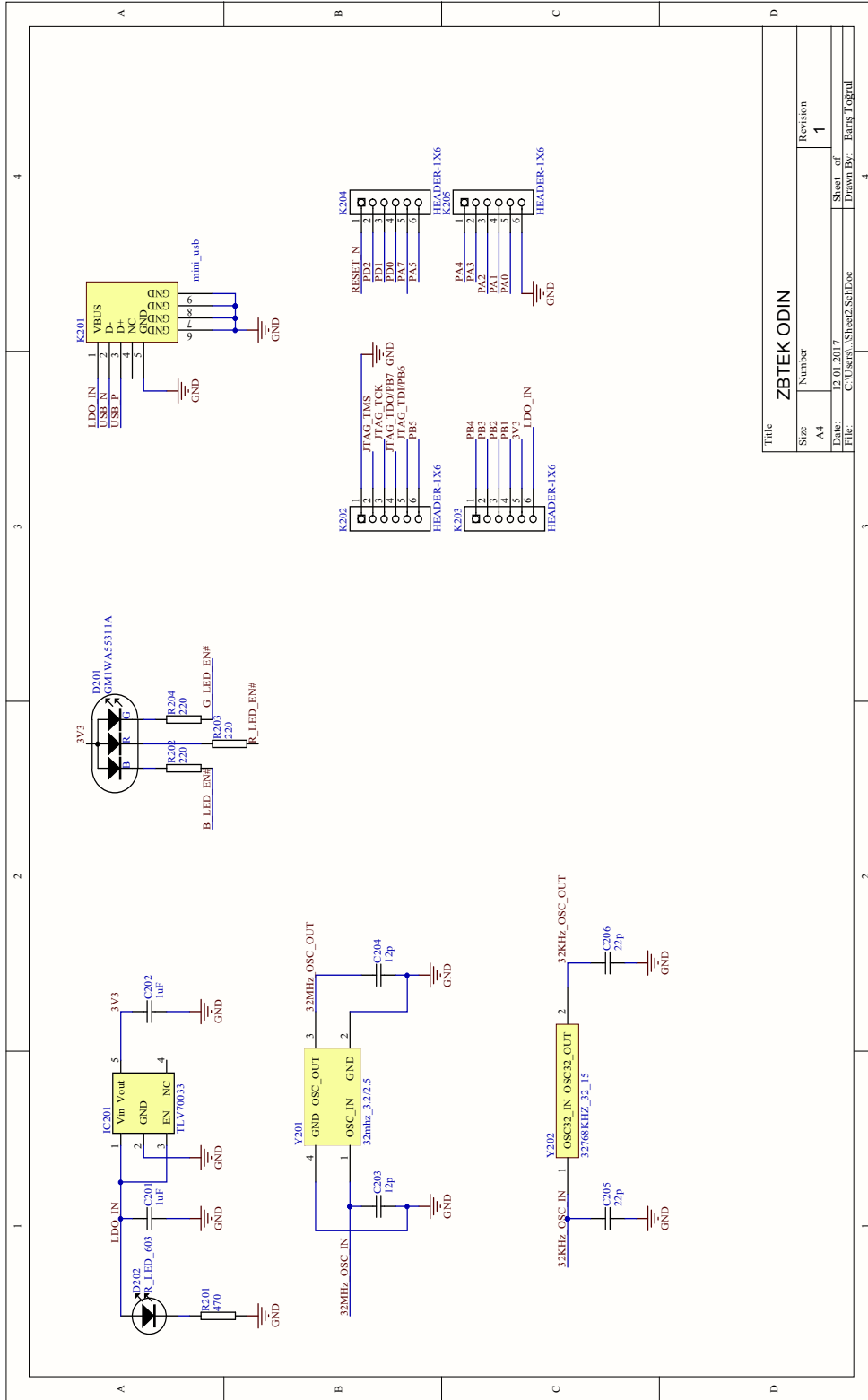
Ek E.2: Kullanılan Antenin Boyutları [74]



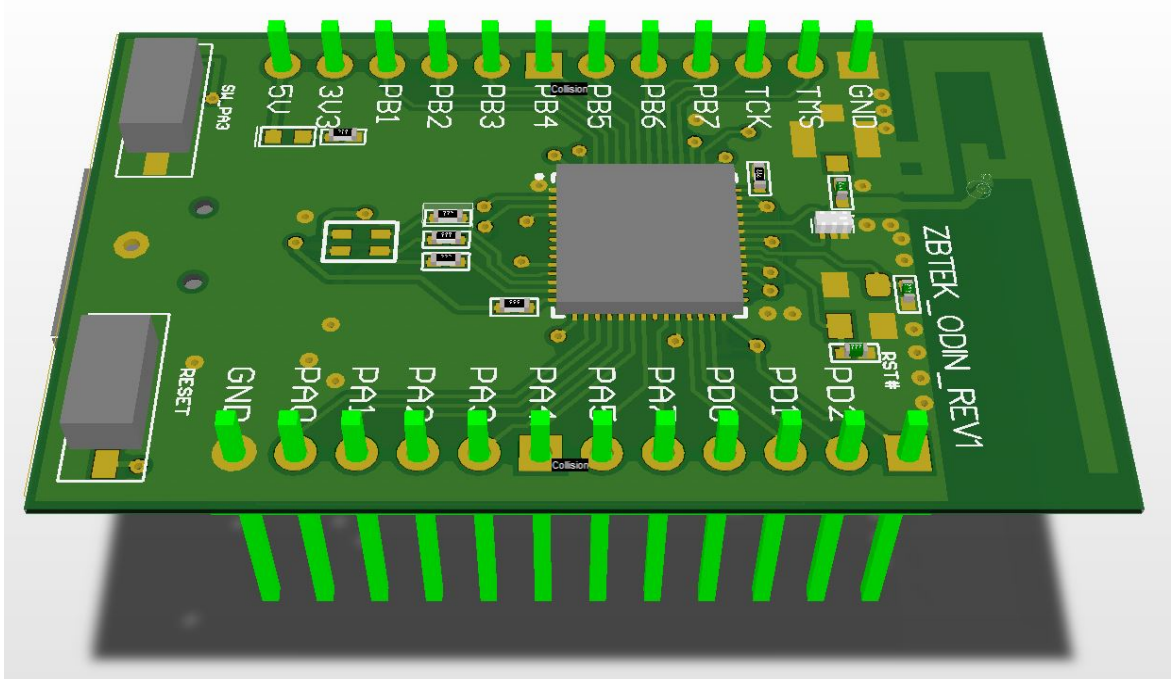
Ek E.3: Kullanılan Antenin Frekans cevabı [75]



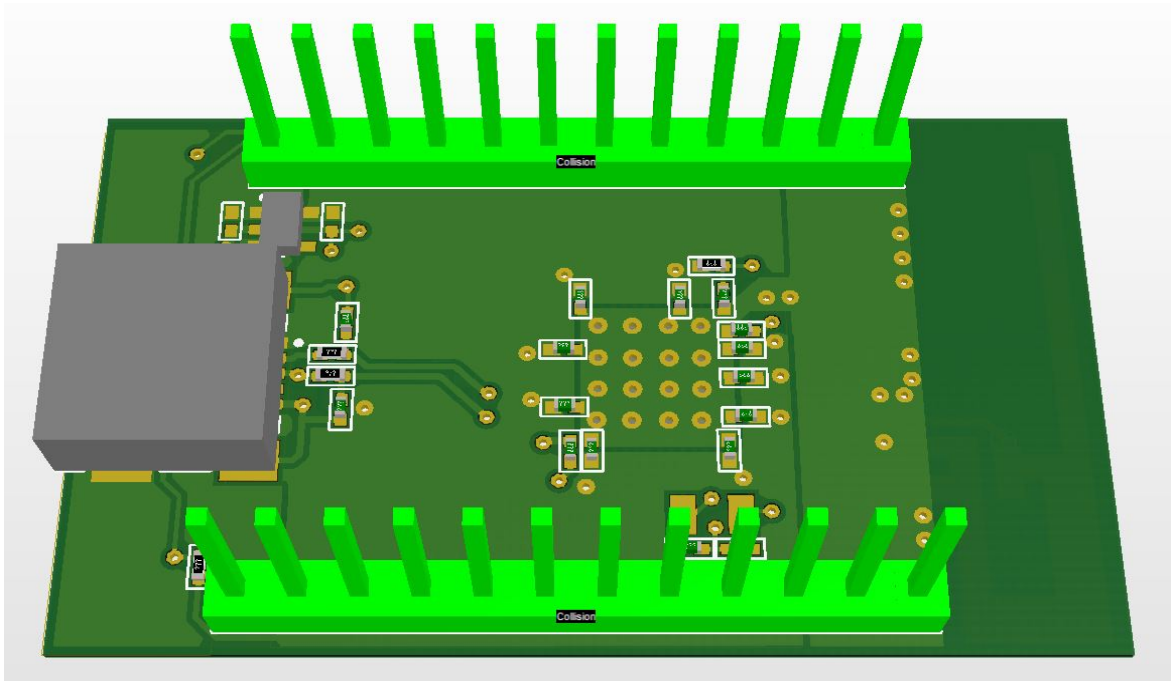
Ek E.4: Tasarlanan Kartın Şema 1.Sayfası



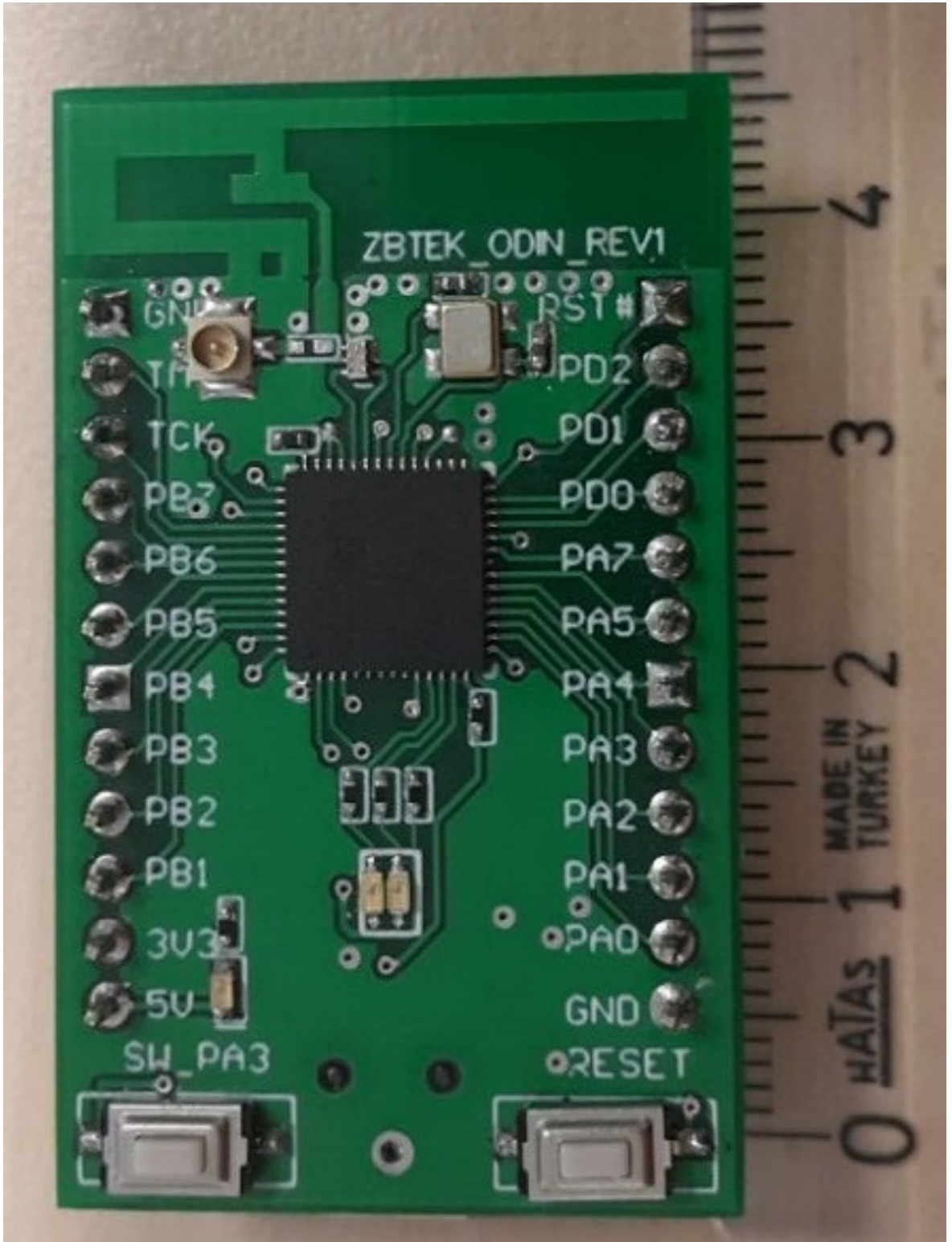
Ek E.5: Tasarlanan Kartın Şema 2.Sayfası



Ek E.6: Tasarlanan Kartın Ön Yüzünün 3 Boyutlu Görünümü



Ek E.7: Tasarlanan Kartın Arka yüzünün 3 Boyutlu Görünümü



Ek E.8: Tasarlanan Kartın Tamamlanmış Hali

ÖZGEÇMİŞ

Kimlik Bilgileri

Adı Soyadı: Barış TOĞRUL

Doğum Yeri: İstanbul, Türkiye

Medeni Hali: Evli

E-Posta: baris_togrul@hotmail.com

Adresi: Aselsan A.Ş. Teknokent Yerleşkesi, ODTÜ Teknokent - ANKARA

Eğitim

Lise: Adana Erkek Lisesi - 2008

Lisans: Hacettepe Üniversitesi - 2013

Y. Lisans: Hacettepe Üniversitesi - 2017

Yabancı Dil ve Düzeyi

İngilizce: İleri

İş Deneyimi

2013- ASELSAN A.Ş. Donanım Tasarım Mühendisi

Deneyim Alanları

Yüksek hızlı baskı devre kartı tasarımı

Düşük güçlü sistem tasarımı

ARM-Cortex M yazılım tasarımı

Tezden Üretilmiş Projeler ve Bütçesi

-

Tezden Üretilmiş Yayınlar

-

Tezden Üretilmiş Tebliğ ve Poster Sunumu ile Katıldığı Toplantılar

-



HACETTEPE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
YÜKSEK LİSANS/DOKTORA TEZ ÇALIŞMASI ORJİNALLİK RAPORU

HACETTEPE ÜNİVERSİTESİ
FEN BİLİMLER ENSTİTÜSÜ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI BAŞKANLIĞINA

Tarih: 26/09/2017

Tez Başlığı / Konusu: RPL'İN GÜVENİLİR VE DÜŞÜK GÜCE SAHİP HAREKETLİLİK DESTEĞİ İÇİN YENİDEN TASARLANMASI

Yukarıda başlığı/konusu gösterilen tez çalışmamın a) Kapak sayfası, b) Giriş, c) Ana bölümler d) Sonuç kısımlarından oluşan toplam 116 sayfalık kısmına ilişkin, 26/09/2017 tarihinde şahsım/tez danışmanım tarafından Turnitin adlı intihal tespit programından aşağıda belirtilen filtrelemeler uygulanarak alınmış olan orijinallik raporuna göre, tezimin benzerlik oranı % 2'tür.

Uygulanan filtrelemeler:

- 1- Kaynakça hariç
- 2- Alıntılar hariç/dâhil
- 3- 5 kelimedenden daha az örtüşme içeren metin kısımları hariç

Hacettepe Üniversitesi Fen Bilimleri Enstitüsü Tez Çalışması Orijinallik Raporu Alınması ve Kullanılması Uygulama Esasları'nı inceledim ve bu Uygulama Esasları'nda belirtilen azami benzerlik oranlarına göre tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Gereğini saygılarımla arz ederim.

Tarih ve İmza

Adı Soyadı: Barış TOĞRUL

Öğrenci No: N13222737

Anabilim Dalı: Elektrik-Elektronik Mühendisliği

Programı:

Statüsü: Y.Lisans Doktora Bütünleşik Dr.

26.09.2017

DANIŞMAN ONAYI

UYGUNDUR.

DOÇ. DR. ALI ZIYA ALKAR
(Unvan, Ad Soyad, İmza)