



**T.C.
İSTANBUL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**



DOKTORA TEZİ

**KABLOSUZ ALGILAYICI AĞLARDA YÖNLENDİRME
İÇİN GENEL UYGULAMA GELİŞTİRME ÇERÇEVESİNİN
TASARIMI**

Ali Nourouzi ALLAYLEH
Bilgisayar Mühendisliği Anabilim Dalı
Bilgisayar Mühendisliği Programı

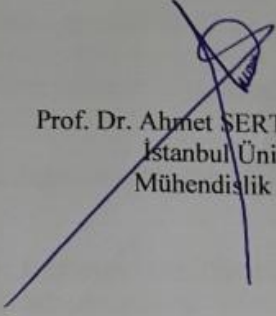
Danışman
Prof.Dr. Ahmet SERTBAŞ
II. Danışman
Prof.Dr. A.Halim ZAİM

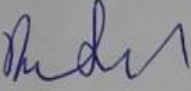
Haziran 2015

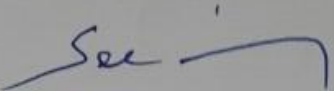
İSTANBUL

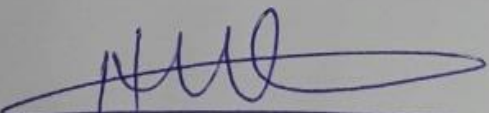
Bu çalışma, 10/06/2015 tarihinde aşağıdaki jüri tarafından Bilgisayar Mühendisliği Anabilim Dalı Bilgisayar Mühendisliği programında Doktora Tezi olarak kabul edilmiştir.


Tez Jürisi:


Prof. Dr. Ahmet SERTBAŞ (Danışman)
İstanbul Üniversitesi
Mühendislik Fakültesi


Prof. Dr. Rifat YAZICI
İstanbul Ticaret Üniversitesi
Mühendislik ve Tasarım Fakültesi


Prof. Dr. Selim AKYOKUŞ
Doğuş Üniversitesi
Mühendislik Fakültesi


Doç. Dr. Hacı Ali MANTAR
Gebze Teknik Üniversitesi
Mühendislik Fakültesi


Yrd. Doç. Dr. Muhammed Ali AYDIN
İstanbul Üniversitesi
Mühendislik Fakültesi

ÖNSÖZ

Doktora tezi olarak hazırladığımız bu çalışma, 10 Haziran 2015 tarihinde İstanbul Üniversitesi Fen Bilimler Enstitüsü'nde Prof. Dr. Ahmet Sertbaş, Prof. Dr. A. Halim Zaim, Prof. Dr. Rıfat Yazıcı ve Prof. Dr. Selim Akyokuş'ten oluşan jüri önünde savunulmuş ve başarılı bulunmuştur.

Tezin oluşumunda görüşlerinden ve yardımlarından istifade ettiğim pek çok kişi olmuştur. Fakat en önemli desteği veren danışmanlarım Prof. Dr. Ahmet Sertbaş ve Prof. Dr. A. Halim Zaim olmuştur. Kendilerine duyduğum minnettarlığı ifade etmem kolay değildir.

Son olarak başta Eşim olmak üzere tüm aileme ve biricik Annem'e derin teşekkürlerimi sunuyorum.

Bu çalışmanın araştırmacılara faydalı olması temennimdir. Nihayet çalışmadaki hataların sorumluluğu da şüphesiz ki bu satırların yazarına aittir.

Haziran 2015

Ali Nourouzi Allayleh

İÇİNDEKİLER

Sayfa No

ÖNSÖZ.....	i
İÇİNDEKİLER	ii
ŞEKİL LİSTESİ.....	vi
TABLO LİSTESİ	viii
SİMGE VE KISALTMA LİSTESİ	ix
ÖZET.....	x
SUMMARY	xii
1. GİRİŞ.	1
1.1. TEZİN YAPISI.....	3
2. GENEL KISIMLAR.....	4
2.1. ETKİN ENERJİ TÜKETİMİ PARAMETRELERİ.....	4
2.1.1 Komşu Döğümler ve Havuz Döğüm Arasındaki Mesafe	5
2.1.2 Komşu Döğümler ve Verici Döğüm Arasındaki Mesafe.....	5
2.1.3. Komşu Döğümün Enerjisi.....	6
2.1.4. Komşu Döğümün Tampon Kapasitesi	6
2.1.5. Teslimat Süresi.....	7
2.1.6. Yol Seçiminde Başarı/Başarısızlık.....	7
2.1.7. Atlama Sayısı	8
2.1.8. İletimde Gecikme	8
2.1.9. İletim Güvenliğinin Risk Ve Olasılığı	8
2.1.10. Kaynak Sayısı	8
2.2. YÖNLENDİRME PROTOKOLLERİ.....	9
2.2.1. LEACH Protokolü	9
2.2.1.1. LEACH'in Avantajları.....	10
2.2.1.2. LEACH'in Dezavantajları.....	11
2.2.1.3. LEACH'de İyileştirmeler.....	11
E-LEACH	11
2.2.2. Doğrudan Yayılma.....	13

2.2.2.1	<i>İlgi Yayılımı</i>	13
2.2.2.2	<i>Gradyan Ayarı</i>	13
2.2.2.3	<i>Güçlendirme</i>	13
2.2.2.4	<i>Veri Teslimatı</i>	14
2.2.2.5	<i>Doğrudan Yayılmanın Avantajları</i>	14
2.2.2.6	<i>Doğrudan Yayılmanın Dezavantajları</i>	15
2.2.3	<i>Gossiping Protokolü</i>	15
2.2.3.1	<i>Gossiping'in Avantajları</i>	16
2.2.3.2	<i>Gossiping'in Dezavantajları</i>	17
2.2.3.3	<i>İyileştirilmiş Gossiping Protokolleri</i>	17
2.2.4	<i>Enerji Verimli Algılayıcı Yönlendirme Protokolü</i>	19
2.2.4.1	<i>Algoritma</i>	20
2.2.4.2	<i>EESR'nin Avantajları</i>	22
2.2.4.3	<i>EESR'nin Dezavantajları</i>	22
2.2.4.4	<i>EESR'de İyileştirmeler</i>	23
3.	MALZEME VE YÖNTEM	25
3.1.	ENERJİ TÜKETİMİ İÇİN ETKİN PARAMETRELER VE DEĞERLENDİRİMİ	25
3.1.1.	İki Düğüm Arası Mesafenin Tahmini	25
3.1.1.1.	<i>İşaret Gücü</i>	26
3.1.1.2.	<i>İki Düğüm Arasındaki Ortalama Gecikme</i>	26
3.1.1.3	<i>GPS</i>	27
3.1.1.4.	<i>Yerleşim ve Dağılım</i>	27
3.1.1.5.	<i>Olasılık</i>	28
3.1.1.6.	<i>Küme Başı Sayısı</i>	28
3.1.1.7.	<i>Mesafe Tahmin Yöntemlerinin Karşılaştırılması</i>	29
3.1.2.	Düğümelerde Uygun Enerji Tüketim Modeli Uygulamak	30
3.1.2.1.	<i>Klasik Enerji Tüketimi Modeli</i>	31
3.1.2.2.	<i>μAMPS Özel Modeli</i>	32
3.1.2.3.	<i>Mica2 Özel Modeli</i>	33
3.1.2.4.	<i>Mica2 Özel Modeli Ve Gerçek Ölçümler</i>	34
3.1.2.5.	<i>Karşılaştırma Sonucu</i>	35
3.1.3.	Ağ Dayanıklılığı	36
3.1.4	Kaynak Sayısı	37
3.1.4.1.	<i>İki Fazlı Yöntem</i>	37

3.1.4.2. Çekme Diffüzyonu Yöntem	38
3.1.4.3 Tek Fazlı Yöntem	40
3.1.4.4 Kaynak ve Havuz Üzerindeki Filtrenin Seçilmesi	40
3.1.4.5. Kaynak Sayısına Göre Yöntemlerin Karşılaştırılması	41
3.1.5. Atlama Sayısı	43
3.1.5.1. Yöntemler	43
3.1.5.2. Atlama Sayısı açısından Yöntemlerin Karşılaştırılması.....	44
3.2 KABLOSUZ ALGILAYICI AĞI'NİN ÇALIŞMA AŞAMALARI.....	45
3.3. GENETİK ALGORİTMA	49
3.3.1. Başlatma.....	49
3.3.2. Uygunluk.....	50
3.3.3. Seçme	50
3.3.4. Çaprazlama (Crossover).....	50
3.3.5. Kablosuz Algılayıcı Ağlarında Uygunluk Parametreleri	51
3.4 AĞ SİMULATÖRÜ - 2	53
3.5. MOBİL ALGILAYICI AĞLAR İÇİN GENETİK YAKLAŞIM.....	54
3.5.1. Problem Tanımı	54
3.5.2. Başlatma.....	58
3.5.3. Veri Kümeleme	58
3.5.4. Seçme İşlemi	58
4. BULGULAR.....	60
4.1. KAA YÖNLENDİRME PROTOKOLLERİNİN KARŞILAŞTIRMASI.....	60
4.1.1. Simülasyon Sonuçları	64
4.2 GENETİK ALGORİTMA YAKLAŞIMI KULLANILARAK ELDE EDİLEN SONUÇLAR	67
4.2.1. Düğüm Yerleşim Optimizasyonu	67
4.2.1.1. GA Uygunluk Fonksiyonu Seçimi.....	67
4.2.1.2. Simülasyon Sonuçları.....	69
4.2.2. Kapsama Alanının Belirlenmesi	71
4.2.2.1. Seçilen Uygunluk Fonksiyonu	72
4.2.2.1.1. ECEP'lerde Kapsama Problemi İçin Formül	72
4.2.2.1.2. ECEP Bağlanılabilirliğinin Belirlenmesi.....	74
4.2.2.1.3. Şifreleme Yöntemi.....	74
4.2.2.2. Değerlendirme ve Simülasyon Sonuçları	75
4.2.3. KAA'de kümeleme	77

4.2.3.1. Seçilen Uygunluk Fonksiyonu	77
4.2.3.1. Değerlendirme ve Simülasyon Sonucu	78
4.2.4. Kablosuz Algılayıcı Ağlarda Verilerin Birleştirilmesine İlişkin Sonuçlar ...	80
4.2.4.1. Uygunluk Fonksiyonu Seçilmesi.....	80
4.2.4.2. Değerlendirme ve Simülasyon Sonuçları	81
4.2.5. MOBİL AĞLAR İÇİN GENETİK YAKLAŞIM SONUÇLARI	83
4.2.5.1. Uygunluk Fonksiyonu Seçimi	83
4.2.5.2. Değerlendirme ve Simülasyon Sonuçları	84
5. TARTIŞMA VE SONUÇ	91
KAYNAKLAR	94
EKLER.....	94
EK 1: OZEL SIMÜLATÖR YAZILIMI	98
Sınıflar.....	98
EK 2. TÜRKÇE - İNGİLİZCE TERİMLER SÖZLÜĞÜ	148
EK 3: TEZ DEN ÇIKAN YAYINLAR.....	150
ÖZGEÇMİŞ.....	151

ŞEKİL LİSTESİ

Sayfa No

Şekil 2.1: a- Kaynağı keşfetmek için yollanan ilgi bilgileri iletimi; b- Kaynaktan havuza keşif bilgisinin iletimi	5
Şekil 2.2: LEACH protokolü ve TDMA çizelgeleri	10
Şekil 2.3: Doğrudan Yayılma protokolünün çalışması	14
Şekil 2.4: Flooding ve Gossiping iletim mekanizmaları	16
Şekil 2.5: SGDF’de yönlendirme senaryosu	18
Şekil 2.6: LGOSSIPING’de veri yönlendirmenin şematiği	18
Şekil 2.7: ELGOSSIPING’de yönlendirme	19
Şekil 2.8: Boyutlu (x, y) koordinat bazlı düğümlerin konumları	21
Şekil 2.9: 16 Otopanlı ağ	23
Şekil 2.10: Her sektördeki röle düğümleri	24
Şekil 3.1: Değişik sinyal güçleri ile yayılımın şematik görüntüsü	26
Şekil 3.2: Kentsel alanlarda algılayıcıların dağılımı	27
Şekil 3.3: Bir kablosuz algılayıcı ağında ızgara yerleşimi	28
Şekil 3.4: Baş küme düğümleri tarafından geçilen adımların sayısı	29
Şekil 3.5: İki faz çekme yayma metodunun fonksiyonu	38
Şekil 3.6: Çıkış yayılım yöntemi	39
Şekil 3.7: Tek fazlı çekme metodunun fonksiyonu	40
Şekil 3.8: Havuz ve kaynak sayısına göre doğru yayılım metodunun seçilmesi	41
Şekil 3.9: İteleme diffüzyonu ve tek fazlı çekme metodu diyagramları	42
Şekil 3.10: Değişik sayıda kaynak ve havuz ile tek-faz çekme metodu	42
Şekil 3.11: İlgi pakedinin yayılması ve atlama sayısının güncellenmesi	44
Şekil 3.12: KAA’lerin temel çalışma aşamaları	45
Şekil 3.13: KAA için kılavuzlu yerleşim	46
Şekil 3.14: Kapsama alanı aşaması	46
Şekil 3.15: KAA’ların kümelenmesi	47
Şekil 3.16: KAA’lerin farklı aşamalarında görülen enerji tüketimi	48
Şekil 3.17: GA mekanizmasının genel şeması	49
Şekil 3.18: Mobil algılayıcı ağ için önerilen GA tabanlı algoritma.	57

Şekil 4.1: Toplam enerjinin iteratif değişimi	65
Şekil 4.2: Hayatta kalan toplam düğüm sayılarının değişimi	65
Şekil 4.3: Kaybolan paket sayısının değişimi	66
Şekil 4.4: Her bir işlemdeki atlama sayısını gösteren grafik	66
Şekil 4.5: Alınan paket sayısını gösteren grafik	66
Şekil 4.6: Mevcut Algılayıcı sayısı ile ağ içerisinde bulunan aktif ve alınan paket sayılarının karşılaştırımı	70
Şekil 4.7: Güç dayanğı ile ağı kullanım süresinin karşılaştırımı	70
Şekil 4.8: Kablosuz Algılayıcı ağlarında kapsama şeması	71
Şekil 4.9: Temsili şifreli ağ	75
Şekil 4.10: Verilen ölçekte ağ ömrü	77
Şekil 4.11: Ağ kullanım ömrü süresince enerji tüketim oranı	79
Şekil 4.12: İki yöntemle canlı devrelerin karşılaştırılması	80
Şekil 4.13: Sanal ortamların ömrü boyunca görülen enerji tüketim oranları	82
Şekil 4.14: Her iki yöntemle kapsama karşılaştırması	83
Şekil 4.15: Örnek sanal ortam	84
Şekil 4.16: Enerji tüketiminin sanal ortamın ömrüne oranı	85
Şekil 4.17: Önerilen ve LEACH algoritmaları arasında kapsam karşılaştırması	85
Şekil 4.18: Simulatörde kullanılan örnek parametreler	88
Şekil 4.19: Bulunan tüm rotaların hesaplanan tercih değerleri	89
Şekil 4.20: Rotaların uygulamalardaki tercih değerleri	89
Şekil 4.21: Her bir uygulama için önerilen rota tercih değerleri	90

TABLO LİSTESİ

Sayfa No

Tablo 2.1: Kuadran adları, sektör kimliği ve yönetici düğüm adları	21
Tablo 3.1: Mesafe tahmin yöntemleri	30
Tablo 3.2: Radyo özellikleri, klasik model	32
Tablo 3.3: μ AMPS modeli algılayıcı durumları.....	32
Tablo 3.4: Mica2 modelinin akım tüketimi.....	33
Tablo 3.5: Gerçek ölçümler ile akım tüketimi	34
Tablo 3.6: Enerji tüketimi modelleri	35
Tablo 3.7: Kaynak ve havuz sayısına göre yayılım yöntemleri	41
Tablo 3.8: Atlama sayısına göre servis kalitesi parametreleri	44
Tablo 4.1: Düz ve Hiyerarşik yönlendirme protokolleri arasındaki genel karşılaştırma	61
Tablo 4.2: Yönlendirme protokoller arasında genel karşılaştırma	62
Tablo 4.3: Simülasyon genel parametreleri	65
Tablo 4.4: Simülasyon parametreleri	69
Tablo 4.5: GA parametre değerleri	69
Tablo 4.6: Ağın ilk zamanlarında tespit edilen değerler	76
Tablo 4.7: Ağın son zamanlarında tespit edilen değerler.....	76
Tablo 4.8: Simülasyon parametreleri	79
Tablo 4.9: GA parametre değerleri	79
Tablo 4.10: Simülasyon parametreleri	82
Tablo 4.11: GA parametre değerleri	82
Tablo 4.12: Test edilen sanal ortam parametreleri.....	84

SİMGE VE KISALTMA LİSTESİ

Kisaltmalar	Açıklama
GPS	: Global Positioning System
WSN	: Wireless Sensor Network
QoS	: Quality of Service
FF	: Fitness Function
DV	: Direction Vector
CH	: Cluster Head
JPAC	: Java Genetic Algorithm Package
LEACH	: Low Energy Adaptive Clustering Hierarchy
PEGASIS	: Power-Efficient Gathering in Sensor Information Systems
BS	: Base Station
DDBS	: Direct distance to Base Station
CD	: Cluster based distance
CDS	: Cluster-based distance-standard
MST	: Minimum Spanning Tree
TDMA	: Time division multiple access
GA	: Genetic Algorithm
EESR	: Energy Efficient Sensor Routing

ÖZET

DOKTORA TEZİ

KABLOSUZ ALGILAYICI AĞLARDA YÖNLENDİRME İÇİN GENEL UYGULAMA GELİŞTİRME ÇERÇEVESİNİN TASARIMI

Ali Nourouzi ALLAYLEH

İstanbul Üniversitesi

Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

Danışman : Prof.Dr.Ahmet Sertbaş

II. Danışman : Prof.Dr.A.Halim Zaim

Kablosuz algılayıcı ağlarının (KAA), kısıtlı radyo iletişim kapasitesi ve pek çok amaç için kullanılan ve kullanılmaya devam edecek olan kısıtlı programlama ile donatılmış çok sayıda algılama aygıtından oluşur. Kablosuz algılayıcı ağlarında bilgilerin algılayıcı devrelerinden baz istasyonuna transferini sağlayan kümeleme ve yönlendirme özellikle önemlidir. Ağdaki düşük güç enerji devreleri nedeniyle ağ ömrünü belirlemek için kullanılan en önemli unsurlardan biri enerjidir.

Kablosuz algılayıcı ağlarının (KAA) gelişimi boyunca sürekli olarak zorlukla karşılaşmış olup çok sayıda araştırmacı bu zorlukların üstesinden gelmeye çalışmıştır. Temel zorluklar arasında ağların enerji tüketimi, dayanıklılığı ve performansı yer almakta olup, bunlar araştırmacılar tarafından büyük ilgi görmüştür. Sorunları derinlemesine araştırdıkça, bu zorluklara neden olan parametrelerle ilgili daha ayrıntılı bir bakış açısı kazandık. Diğer bir deyişle, bunlar önemsiz ancak aynı zamanda da iyileştirilmesi halinde ağın geliştirilmesini ve zorlukların üstesinden gelinmesini sağlayabilecek etkili parametrelerdir. Bu bölüm tüm parametrelerin teknik bir bakış açısıyla ele almayı ve araştırmayı amaçlamakta olup önceki çalışmalar hakkında da genel bir rapor sunmaktadır. Ayrıca, KAA'lerin geliştirilmesi konusundaki faktörlerin ve yaklaşımların avantajları ve zayıf yönleri tartışılmaktadır. Çalışmada parametreler sınıflandırılırken, ağların farklı katmanlarını etkileyen süreçler sunuluyor. Bunun yanı sıra genel yöntemlerin bir karşılaştırması yapılırken, ilave tavsiyelerde bulunuluyor. Her bir karşılaştırma sonunda zorluklarla mücadele etmek ve ağları iyileştirmek için izlenecek en iyi yöntem sunuluyor.

Bir ağda kullanılan düşük güç enerji devreleri nedeniyle, enerji kablosuz algılayıcı ağlarının (KAA'lerin) uzun ömürlü olmasında çok önemli bir rol oynuyor. Bu bölümde kablosuz algılayıcı ağlarında optimum enerji tüketimi sağlayan bir yönlendirme yöntemiyle ilgili ayrıntılara yer verilmiştir. Yüksek enerji verimliliğine sahip bazı routing protokolleri (LEACH, Direcred Diffusion, Gossiping, PEGASIS ve EESR) incelenmiştir. Yönlendirme protokolleri, uygulamaların ve genel olarak KAA 'lerin gereksinimlerini etkileyen çeşitli ölçütler bakımından karşılaştırıldı.

Kablosuz algılayıcı ağları (KAA) için çok sayıda uygulama olduğu biliniyor ve bu çeşitlilik mevcut protokollerin ve özel parametrelerin geliştirmesin ihtiyacını doğuruyor. Öne çıkan bazı parametreler arasında tüm uygulamalarda kilit rol oynayan ağın ömrü ve yönlendirme enerji tüketimi yer alıyor. Doğrusal olmayan optimizasyon yöntemlerinden biri olan genetik algoritma ise büyük ölçekli uygulamalardaki verimliliği sayesinde ve son formülün operatörler tarafından değiştirilebilmesi nedeniyle görece olarak daha iyi bir seçenek oluşturuyor. Bu bölümde bir KAA'nin devre yerleştirme, ağ kapsama alanı, kümeleme ve veri toplama dahil tüm işlemsel aşamalarında kapsamlı bir iyileşme ortaya koymaya çalışılmakta olup yönlendirme ve uygulama bazlı KAA için ideal bir parametre setine ulaşmaya çalışılmaktadır. Genetik algoritma kullanarak ve NS'deki simülasyonların sonuçlarından hareketle belli bir form işlevine ulaşılmış, optimize edilmiş ve KAA'lerin tüm işlemsel aşamaları için özelleştirilmiştir.

Son bölümlerde belirtildiği gibi algılayıcı ağlarıyla ilgili en önemli sorunlardan biri enerji tüketimini azaltmak ve ağ ömrünü arttırmaktır. Bu bölümde uygulama bazında parametrelerin önceliği konusunda bazı tanımlamalarda bulunuyoruz. Bu parametreler çok amaçlı sistemler için esnek bir şekilde düzenlenebilmektedir. En iyi performansa ulaşmak için GA ile geliştirdiğimiz matematiksel formülü kullanarak yeni bir yaklaşım sistemi sunuyoruz. Tüm olası yollar keşfedildikten sonra, sistem uygulama bazında öncelikleri belirtiyor. Son olarak umut vadeden sonuçlara ulaştığımız çalışmamızdan bazı sonuçlar sunuyoruz. Önceden var olan yönlendirme algoritmaları ve yardımcı modelimiz arasında bir karşılaştırma yapılmıştır. Simülasyon sonuçları sunulan yöntemlerin daha yüksek başarı oranı, verimlilik anlamında daha iyi performans gösterdiğini ve algılayıcı ağlarının ömrünü etkili bir şekilde arttırdığını göstermektedir.

Haziran 2015, 165 sayfa.

Anahtar kelimeler: Kablosuz algılayıcı ağlarının, Genetik algoritma, Yönlendirme protokolleri, Enerji verimliliği

SUMMARY

Ph.D. THESIS

DESIGN OF A GENERIC APPLICATION DEVELOPMENT FRAMEWORK FOR ROUTING IN WIRELESS SENSOR NETWORKS

Ali Nourouzi ALLAYLEH

İstanbul University

Institute of Graduate Studies in Science and Engineering

Department of Computer Engineering

Supervisor : Prof.Dr. Ahmet Sertbaş

Co-Supervisor : Prof.Dr. A. Halim Zaim

WSN consist of large number of sensing devices, which are equipped with limited radio communication capabilities and limited computing which is used for many purposes in the present and the future. Clustering and routing in wireless sensor networks for the transfer of information from sensor nodes to base station are especially important. Energy is one of the most important items to determine the network lifetime due to low power energy nodes included in the network.

The development of wireless sensor networks (WSNs) has never been free of challenges and many researchers have been interested in overcoming them. The energy consumption, longevity, and performance of the networks are among the main challenges and have drawn a great deal of attention from researchers. To put it another way, there are trivial but at the same time effective parameters whose improvement can enhance the network and overcome the challenges. This Thesis tries to introduce and survey all the parameters from a technical viewpoint and presents a general report on the previous works. In addition, advantages and weaknesses of the factors and approaches to improving WSNs are discussed. The study proposes a classification of the parameters and processes that affect different layers of the networks. Furthermore, a comparison of the common methods and further recommendations are represented. Each comparison is concluded by a proposal of the best method to deal with the challenges and improve the networks.

Because of the low-power energy nodes used in a network, energy plays a pivotal role in the lifetime of wireless sensor networks (WSNs). This thesis elaborates on a routing

method featured with optimum energy consumption in wireless sensor networks. Some of routing protocols with high energy efficiency (LEACH, Director Diffusion, Gossiping, and EESR) were examined. The routing protocols were compared regarding variety of metrics influencing requirements of the specific application and WSNs in general.

There are several applications known for wireless sensor networks (WSN), and such variety demands improvement of the currently available protocols and the specific parameters. Some notable parameters are lifetime of network and energy consumption for routing which play key role in every application. Genetic algorithm is one of the non-linear optimization methods and relatively better option thanks to its efficiency for large scale applications and that the final formula can be modified by operators. This Thesis tries to exert a comprehensive improvement in all operational stages of a WSN including node placement, network coverage, clustering, and data aggregation and achieve an ideal set of parameters of routing and application based WSN. Using genetic algorithm and based on the results of simulations in NS a specific fitness function was achieved, optimized, and customized for all the operational stages of WSNs.

As mentioned, one of the most important issues related to sensor networks is reducing energy consumption and increase network lifetime. In this thesis we assume some definition about priority of the parameters based on applications. These arrangements of parameters are flexible to multipurpose systems. We propose a new approach system by using mathematical formula that we improved it by GA to reach best performance. After discover all possible routes, the system specifies priorities based on application. At last, we present some results according to our experiment which were promising outcomes. A comparison has been made between pre-existing routing algorithms and our agent model. The Simulation results show that the proposed method gives better performance in terms of higher success rate, throughput and it effectively increases the lifetime of the sensor networks.

June 2015, 165 Pages

Keywords: Wireless Sensor Network, Genetic Algorithm, Routing Protocols, Energy efficiency.

1. GİRİŞ

Bir olayın saptanması ve raporlanması kablosuz ağların amaçlarından birisidir. ve bu kablosuz ağlardaki bütün düğümler tarafından yapılır. Yönlendirme konusunda ciddi bir konu, kaynak ile hedef arasında uygun bir rotanın çizilmesidir. Bu amaca genellikle bir çok algoritma ve protokol aracılığıyla ulaşılır. Doğrudan ve yönlendirme, dayanıklılık ve enerji tüketimi gibi faktörleri de etkiler. Bu konu ile ilgilenilirse ağ performansı fark edilebilir bir şekilde değiştirilebilir. İletim, yakın düğüm seçimi ve uygun düğüm seçimi konularını çalışmanın, yönlendirme ile enerji tüketimi, dayanıklılık, trafik, zaman ve diğer konularda daha iyi bir çözüm bulmaya yardımcı olabileceğinin gösterilmiş olması, bu çalışmanın önemini göstermektedir.

Algılayıcı teknolojilerinde, uygulamalardaki çeşitlilik kadar çok gelişme elde edilmiştir. ve daha bir çok gelişme de devam etmektedir. Düşük maliyetli ve düşük güç tüketen küçük boyutlu algılayıcı düğümlerinin tasarımı ve geliştirilmesi son zamanlarda üzerinde yoğunlaşılacak bir konum olmuştur. Bu çalışmada bazı enerji verimli yönlendirme protokollerini (LEACH, Doğrudan Yayılma, Gossiping ve EESR), bunların geliştirilmelerini ve ayrıca bunların kablosuz algılayıcı ağına özel veri toplama ve ağ içinde işleme, kümeleme, farklı düğüm rolü atama ve veri merkezli yöntemler gibi tekniklerini inceledik. Bunların ardından bu incelenen yönlendirme protokollerini genel olarak belirli uygulama gerekliliklerini ve KAA etkileyen farklı ölçütlere göre karşılaştırdık. İncelediğimiz bütün sonuçları NS2 ve Matlab simülasyon programlarıyla uyguladık. 4. Bölümde bu karşılaştırmaların uygulama sonuçlarını dikkate alarak enerji tüketme açısından en uygun protocol gösterilmiştir.

Kablosuz Algılayıcı Ağlarının gücü sınırlı kapasiteli piller tarafından beslenmektedir. Bu algılayıcı düğümlerinin güç yönetimi etkinlikleri nedeniyle, ağ topolojisi dinamik olarak değişir. Bu temel özellikler iletişim protokolleri için ek zorluklar teşkil etmektedir. Bu tezde yönlendirme protokollerinin güvenli enerji tüketimi ile çalıştırılması üzerinde çalıştık ve enerji optimizasyonu üzerindeki etki faktörlerini

tartıştık. Gossiping protokolünde az miktarda dikkat ile, sonraki sekmenin seçiminde bazı değişiklikler yapıldığında ağ ömrünün artırılabilceğini saptadık.

Kablosuz Algılayıcı Ağlar (KAA) algılama, hesaplama ve iletişim yetisine sahip çeşitli gözetimsiz aygıtları içerir. Bu enerjisi kısıtlı aygıtlar birçok türde uygulama için kullanılır [1]. Örneğin, KAA'lar çevre ve yaşam alanı izlemede, yollarda trafik ölçümlerinde, araç takibinde ve bina içinde personel izlemede kullanılabilir. KAA'ların çeşitli uygulamaları bulunmasına rağmen genellikle belirli sayıda düğüm için maksimum alanı kapsamak ve tekil düğümlerin işlemsel ömrünü uzatmak şeklinde iki ortak amaç için kullanılır [2].

Kapsama optimizasyon problemi hakkındaki ilk çalışmalardan biri Gage [8] tarafından 1992 tarihinde yapıldı. Mevcut çalışmada olduğu gibi, aynı ilgi alanının genişletilmiş şekilde kapsanmasını sağlayacak olan algılayıcı düğümlerin istatistikî düzenlemesini gerçekleştirmek için genel bir kapsama yöntemi önerdiler.

2002 yılında, Howard ve ark. [9] düğümleri kaynak içerisine yaymak için itici davranış kullanan GA tabanlı bir yaklaşım sundular. Bu çalışmada, birçok araştırmacının karşılaştığı yerel-optimum çözüm problemlerinden kaçınmak için düğüm yerine sanal bir "kuvvet" algoritmasından faydalanıldı [10-12]. Sadece uygulama için yalnızca çevrimdışı planlamanın gerekli olduğu bir tam model ağı göz önünde bulundurdular.

Kalpakis ve ark. [10], sabit düğüm lokasyonları ve baz istasyonu ile ağın ömrünü maksimuma çıkartabilen Maksimum Ömür Veri Toplama Algoritması (MLDA) önerdiler. Bu algoritma doğrusal bir program yürüterek maksimum aktarım işleyişini sağlayan kenar kapasitelerini buluyor.

Ardından Hussain ve ark. [12] enerji tüketimi, aktarım tablosu vb. unsurları içeren bir optimum çözüm elde etmek için GA yöntemi kullanan bir yaklaşım önerdiler.

Qu ve ark. [2], çok amaçlı bir GA tabanına sahip bir algılayıcı taşıma yöntemi tanıttılar. Bu çalışmada amaç, kapsama alanını maksimuma çıkartmak ve ağdaki enerji kullanımını asgariye indirmektir.

Azadeh ve ark. [1] WSN'lerdeki enerji sınırlamasını tanımladılar ve ardından enerji tasarruflu yönlendirme oluşturmak için bir GA kullandılar. Önerdikleri algoritma ise sistem ömrünü maksimuma çıkaran bir dizi yönlendirme yolu üretmekteydi.

1.1. TEZİN YAPISI

Tezin 2. kısmında, uygulamadan bağımsız olarak, KAA yönlendirme protokolleri dayanıklılığı arttırmalı ve toplam enerji tüketimini düşürmeli olmasını bahs ediyoruz. KAA tasarımında pek çok önemli konu dikkate alınmalıdır. Bir çok uygulamada algılayıcı leri değiştirmek veya şarj etmek mümkün değildir. Bu doğrultuda, enerji ve dayanım ömrü süresini arttırmak için bazı çalışmalar yapılmıştır. Dayanıklılık ve enerji tüketimi konularını engelleyerek, öneriler ve çözümler sunulmuştur (seçim parametrelerinin ayarlanması).

Tezin 3. kısmında, incelenen yönlendirme protokollerini genel olarak belirli uygulama gerekliliklerini ve KAA'yi etkileyen farklı ölçütlere göre karşılaştırdık. İncelediğimiz bütün sonuçları NS2 ve Matlab simülasyon programlarıyla uyguladık. Bu karşılaştırmaların uygulama sonuçlarını dikkate alarak enerji tüketme açısından en uygun protocol gösterilmiştir.

Tezin 4. kısmında, interaktif bir GA kullanan yöntemleri sunuyoruz. Ayrıca, kalan güç miktarını göz önünde bulundurarak uygun konumları belirleyen hareket algılayıcı düğümlerini yönlendirmek için online yaklaşımdan faydalanılmıştır. Son olarak, bulunan çözüm eş zamanlı olarak çevresel kapsamayı en üst seviyeye çıkartıyor ve güç tüketim ölçütlerini en aza indiriyor. Simülasyon sonuçları önerilen uygunluk fonksiyonunun bu amaçları başardığını doğrulamaktadır.

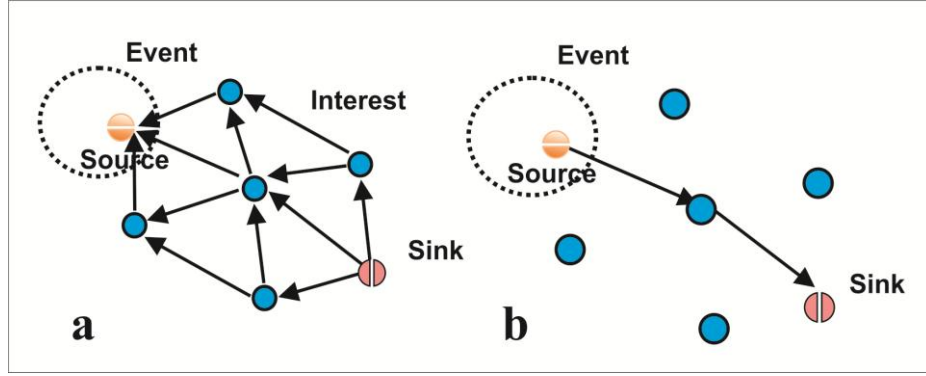
Son olarak, tez çalışmasından elde edilen bulgular ve sonuçlar tartışılacaktır.

2. GENEL KISIMLAR

Bir olayın saptanması ve raporlanması kablosuz ağların amaçlarından birisidir ve bu kablosuz ağlardaki bütün düğümler tarafından yapılır. Yönlendirme işleminde ciddi bir konu, kaynak ile hedef arasında uygun bir rotanın çizilmesidir. Bu amaca yönelik bir çok algoritma ve protokol geliştirilmiştir. Doğrudan ve etkin yönlendirme, dayanıklılık ve enerji tüketimi gibi faktörleri etkilediği gibi, ağ performansının iyileştirilmesini de sağlayabilir. Bu tezde; etkin iletim, yakın düğüm seçimi ve uygun düğüm seçiminin yönlendirme ile enerji tüketimi, dayanıklılık, trafik, zaman vb. konularda daha iyi bir çözüm bulmaya yardımcı olabileceği gösterilmiştir. Uygulamadan bağımsız olarak, KAA yönlendirme protokolleri dayanıklılığı arttırmalı ve toplam enerji tüketimini düşürmelidir. Bu tezde, enerji ve dayanım ömrü süresini arttırmaya dayalı bazı çalışmalar yapılmıştır. Bu bölümde, seçim parametrelerinin ayarlanmasına dayanan yönlendirme konusunda yeni öneriler ve çözümler sunulmuştur.

2.1. ETKİN ENERJİ TÜKETİMİ PARAMETRELERİ

Yönlendirme, Kablosuz Ağlarda ana fonksiyonlardan biridir. Yönlendirme için en uygun seçim, komşu düğümlere yol talepleri gönderilerek ve cevaplar alınarak yapılır. Yönlendirmede önemli bir nokta, rotaya göre değişik etki seviyelerinin ortaya çıkabilmesidir. Genel olarak, yönlendirme paketleri kaynaktan havuz düğümlere ve havuzdan komşu düğümlere olmak üzere iki şekilde uygulanabilir. Birinci türde, yönlendirme düğümü yönlendirme paketini havuza gönderdiği zaman gerçekleşir, ikinci türde ise, havuzdan komşu düğümlere bir cevap iletildiği ve kaynağa geri döndüğü zaman gerçekleşir. Bu iki farklı yönlendirmenin ağ üzerinde değişik etkileri bulunmaktadır [7]. Şekil (2.1.a)'da görüldüğü gibi, havuz düğüm bir verici olarak görev yapar ve ilgi mesajı komşu düğümlere yayar; Şekil (2.1.b)'de ise, kaynak düğümü de bir verici olarak görev yapar. Dolayısıyla, yönlendirmede en uygun komşu düğümün seçilmesine iki değişik açıdan yaklaşılabilir.



Şekil 2.1. a- Kaynağı keşfetmek için yollanan ilgi bilgileri iletimi;
b- Kaynaktan havuza keşif bilgisinin iletimi.

Şekil 2.1'de görüldüğü üzere, kaynakları bulmak için, düğümler komşularına ilgi paket grupları yollarlar, kaynak düğümün bulunmasından sonra da keşif bilgisi havuz düğümüne iletilir. Konsept olarak ideal adımın, araştırma için seçilen düğümün iletim anında en doğru tercih olması kabul edilmiştir. Bu bölümde, yönlendirme ve haritalama konusunda önemli rolü olan faktörler ve parametreler incelenmeye çalışılmıştır. Parametrelerin teknik olarak değerlendirilmesi KAA'ların dışında bazı konuları ortaya çıkartmıştır. Ve bu konuların çözülmesi daha başarılı bir yönlendirme elde edilmesini sağlamıştır. Bilinen yöntemler ve parametrelerin fonksiyonları ile ilgili genel bilgi ve etkileyen/etkilenen faktörler aşağıda verilmiştir.

2.1.1 Komşu Düğümler ve Havuz Düğüm Arasındaki Mesafe

KAA'deki havuz düğüm, verici düğümün frekans aralığında bulunan ve vericinin coğrafi pozisyonunun bir parçası olan bir düğümdür. Bu ağda, iletici düğümün yönlendirimi için komşu düğümün yerini bilmesi gerekir. Paket yönetimi ve rota haritasının oluşturulması konusunda komşu düğümün havuz rotasında olup olmadığı hususu son derece önemlidir. Uzak mesafeli havuz düğümlerde, ortalama enerji tüketimi ve rota haritasının kurulum gecikmesinin daha büyük oluşu örnek olarak verilebilir.

2.1.2 Komşu Düğümler ve Verici Düğüm Arasındaki Mesafe

Komşu düğümler arasındaki mesafe kısa ise, verici daha az enerjiye ihtiyaç duyar. Bu parametre kullanılarak daha uygun rota ve yön seçim yapılması mümkündür.

2.1.3. Komşu Düğümün Enerjisi

Algılayıcıların fiziksel yeterlilikleri hesaba katıldığı zaman (pil kapasitesi gibi), enerji kablosuz ağlarda önemli bir problem olarak ortaya çıkmaktadır.

Uyarlanabilir Kümeleme Hiyerarşisi (LEACH) modeli ile elde edilen (verici) ve alınan (alıcı) enerjileri değerlendirilmiştir. (2.1)'de verildiği gibi, alıcı ile verici arasındaki mesafe (d) referans mesafeden uzun ise (d_0) çoklu yol modeli (α yoldaki enerji kaybı katsayısı i ken); aksi durumda ise açık çevre modeli (enerji kaybı katsayısı = 2) uygulanır.

$$E_{Tx}(l, d) = E_{Tx-elec}(l) + E_{Tx-amp}(l, d)$$

$$= \begin{cases} lE_{elec} + l\varepsilon_{fs} d^2 & d < d_0 \\ lE_{elec} + l\varepsilon_{mp} d^4 & d \geq d_0 \end{cases} \quad (2.1)$$

Burada , E_{elect} elektronik devreleri etkinleştirmek için gerekli olan enerjiyi, ε_{fs} ve ε_{mp} ise sırasıyla açık alan ve çoklu-yol modları için güç artırıcının aktive edilme enerjilerini göstermektedir.

Verici tarafta (2.2 ve 2.3) tüketilen enerjinin "p" ve "q" katsayıları ile gösterilen genel formu şu şekildedir:

$$E_{Tx}(l, d) = p + qd^\alpha \quad (2.2)$$

Alıcı tarafında ise, "1" bit daha almak için tüketilen enerji:

$$E_{Rx}(l) = E_{Rx-elec}(l) = lE_{elec} = p \quad (2.3)$$

2.1.4. Komşu Düğümün Tampon Kapasitesi

Ekonomik enerji tüketimi ile ilgili parametreler ağ yapısında kritik elemanlardır. Bu nedenle, bu tezin ana konusu olmuştur. Bu parametreler esas olarak her bir düğümde trafik yükü ile ilgilidir ve tampon alanı bilgisi ile elde edilirler. Trafik yükü ile birlikte işlemcinin işleyebileceği girdiler de artar. Algılayıcılarda bellek pahalı bir bileşen olduğu için her bir algılayıcıdaki bellek miktarı sınırlıdır. Bu faktörler bu tampon türlerinin yönetimi için uygun bir algoritma kullanılmasını gerektirmektedir. Tampon geçici bir bellektir ve genellikle işlem yapan herhangi bir cihazda bulunabilir. Tampon, değişik işlemciler için dengeyi ve eş işlem hızını sağlar. Tampona bilgi yollamadan

önce yapılması gereken önemli bir şey tamponun durumunu belirlemektir (dolu veya boş). Eğer tampon dolu ise, tampona yollanan bilgiler düşürülür ve kaybedilir, bu durum daha ağır trafiğe ve ağ dayanıklılığı üzerinde olumsuz etkiye sebep olur. Herhangi bir oranda, yüksek tampon fiyatı işlem maliyetini artırır. Algılayıcıların tamponlarını yönetmek için önerilen çözümler sıralama teorisi, genetik algoritma ve yollanan ve alınan paketlerin başlık bilgilerinin azaltılmasıdır.

2.1.5. Teslimat Süresi

Bir paketin kaynaklar arasında dolaşması ve hedefe ulaşması için gerekli olan süre, teslimat süresidir. Parametrenin önemi, gecikme süresi arttıkça, ağ trafiğinin artması ve ağ dayanıklılığın değişmesi ile ilgilidir. Ayrıca, bu parametre uygun komşu düğümler bulmaya yardımcı olur. İletim süresi; çevre gürültüsü, işaret gücü, iletilen paketin sesi, çarpışma ve düğümler arasındaki mesafe gibi farklı parametrelerden de etkilenir. İletim süresini azaltmak amacıyla artırıcılar kullanmak ve yolun gürültü oranını düşürmek gibi yöntemler den yararlanır. Düğümler arasındaki yol aynı zamanda iletim süresini de etkiler, mesafe ile iletim süresi orantılıdır.

$$\text{İletim süresi} = \text{yayılm süresi} - \text{tam alım süresi}$$

Yani, iletim süresi servis kalitesini doğrudan etkiler ve daha iyi iletim süresi daha yüksek performans ve daha az trafik yükü anlamına gelir.

2.1.6. Yol Seçiminde Başarı/Başarısızlık

Paket başlığındaki değişkenler kullanılarak, yol seçiminde başarılar ve başarısızlıklar kaydedilebilir ve raporlanabilir. Bu sayede, her bir düğümün raporu (başarı/başarısızlık oranı) kontrol edilebilir. Seçim fonksiyonunda etkin olan başka bir parametre ise, daha yüksek oranda başarısız olacak düğümlerin daha düşük oranda seçilebilmesidir. Başlık bilgisinde tutulan bu parametre başarılı ve başarısız iletimleri kaydetmek ve raporlamak için çok pratiktir. Bir paket gönderilmeden önce, her bir düğüm kontrol edilerek daha yüksek başarısızlık oranına sahip düğümlerin daha az seçilmeleri sağlanır. Rota seçiminde de bu parametreden yararlanır. Bununla birlikte, bu parametre ses, trafik ve işlem yükü gibi engellerden bağımsız değildir. Ancak en iyi rotayı bulmaya yardımcı olur. Örneğin uçta biten bir düğümün ihtiyaç duyulan enerji ve başka

faktörler konusunda en iyi seçim olarak beklenilmesine rağmen, düğüm hala ulaşılabilir olmayabilir ve düğümüne yollanan paketler düşebilirler. Yani düğümün kayıp oranı artar ve gelecekte bu düğümün seçilme olasılığı azalır. Ayrıca, bu parametre ciddi coğrafi/çevresel değişiklikler olmayan durumlarda kullanılması önerilir.

2.1.7. Atlama Sayısı

Havuz düğüm atlama sayısını ve havuza ulaşmak için gidilmesi gereken yol uzunluğunu tahmin etmek için bu parametre kullanılabilir. Ağın atlama sayısı ve amacına bağlı olarak, bu parametre pozitif veya negatif olarak yorumlanabilir. Örneğin, daha fazla sayıda atlama enerjini dengelemek için tercih edilebilir (amaç); diğer bir deyişle, belirli uygulamalar için daha fazla atlama tercih edilirken acil durumlar ve medikal uygulamalar gibi bazı durumlarda daha az atlama tercih edilebilir.

2.1.8. İletimde Gecikme

Bu parametre, iletimi ve paket taşımalarını tamamlamak için uçtan-uca geçen gecikme süresini temsil eder. Çevre gürültüsü ve trafik yükü gibi faktörler bu parametreyi etkiler. Yukarıda değerlendirilen parametreler (boşluklar ve kaynak arasındaki mesafe, enerji seviyesi, tampon alanı, taşıma süresi, başarı/başarısızlık oranı gibi) bu iletim türünde etkilidirler, fakat önemli olan etken parametrelerin değişik etkileme oranlarına sahip olmasıdır.

2.1.9. İletim Güvenliğinin Risk Ve Olasılığı

Bu parametre, havuz düğümler ana verilerden yola çıkarak olay temelli sorgular gönderdiği ve daha düşük riskli ve daha güvenli sorgular yollamaya çalıştığı durumda kullanılır. Örneğin, bir gemi yangınına kontrol etmek için, soru/cevap mesajları genellikle makine odasına gönderilir. Yani başlangıç, olayın olma ihtimalinin en yüksek olduğu yerden yapılır.

2.1.10. Kaynak Sayısı

Bu parametre havuz düğümüne göre ayarlanır; havuz düğüm menziline daha çok kaynağı kapsayan düğümlere öncelik vermeye çalışır. Bu sayede ilgisiz düğümlere enerji harcanması engellenir.

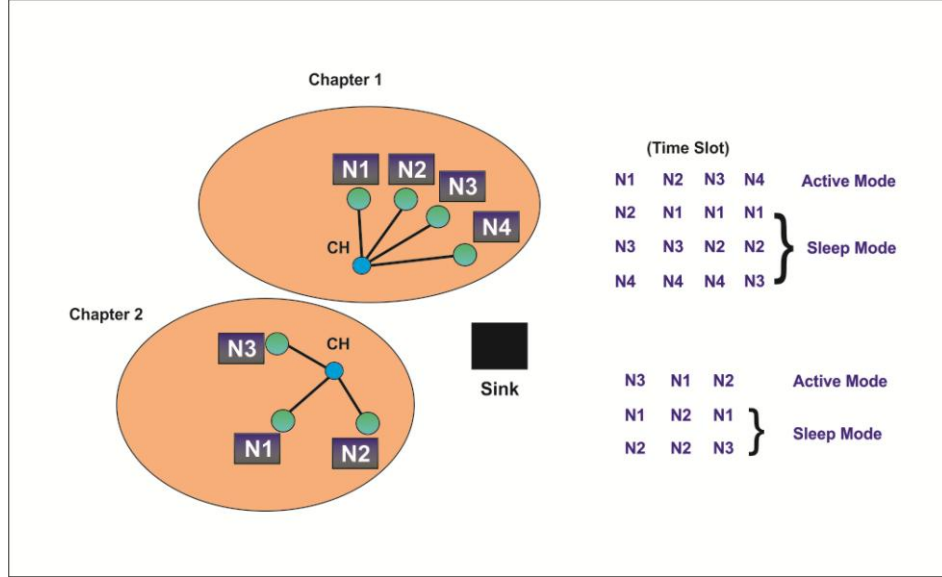
2.2. YÖNLENDİRME PROTOKOLLERİ

KAA küçük algılayıcı düğümlerinden, dolayısıyla algılayıcılar (sıcaklık, ışık, nem, radyasyon vb.), mikroişlemci, bellek, alıcı-verici ve güç kaynağından oluşur. Mevcut ve potansiyel KAA uygulamalarını gerçekleştirmek amacıyla, gelişmiş ve son derece verimli iletişim protokolleri gereklidir. KAA uygulamaya özel olduğundan KAA tasarım gereklilikleri uygulamaya göre değişir. Bu nedenle, yönlendirme protokollerinin gereklilikleri de her uygulama için farklıdır. Örneğin ortam uygulamaları için tasarlanan bir yönlendirme protokolünün gereklilikleri, bir çok açıdan ordu ya da sağlık uygulamaları için tasarlanan protokolden farklıdır. Sonuç olarak, yönlendirme protokollerinin gereklilikleri uygulama sayısı kadar çeşitlilik gösterir [11,17]. Yönlendirme Protokolü tasarımında kullanılan Ölçeklenebilirlik, Gecikme Süresi, Verimlilik, Kaynak Bilinirliği, Veri Toplama, Optimum Rota, Ek Yük vb. ölçütler bulunmaktadır. Bazı uygulamalarda bu ölçütlerden bazılarının seçilmesi ve diğer uygulamalarda ise başka ölçütlerin seçilmesi gerekir. Ancak, tüm Kablosuz Algılayıcı Ağlarının yönlendirme protokolleri tasarımında, uygulamadan bağımsız olarak ağ ömrünü en üst düzeye çıkarmaya ve genel ağ güç tüketimini en aza indirmeye çalışmak esastır. Bu nedenle, enerji tüketimi parametresi diğer faktörlerin yanında daha önceliklidir [10,18]. Bu farklılıklar, KAA 'daki yönlendirme sorunu için uygulama ve mimari gerekliliklerinin yanısıra KAA 'ın doğasından kaynaklanan teknik özelliklerini de hesaba katan yeni bir çok algoritma önerilmesine neden olmuştur.

2.2.1. LEACH Protokolü

Düşük Enerjili Uyarlamalı Kümeleme Hiyerarşisi (LEACH), yerel küme tabanlı istasyonlarının rastgele rotasyonuna dayalı bir protokoldür. LEACH'teki düğümler kümelere ayrılmıştır ve her küme, Küme Üyeleri olarak adlandırılan üyeler ve Küme Başlı (KB) olarak adlandırılan bir koordinatör düğümden oluşur. Küme başları statik şekilde seçilmez, zira bu yaklaşım ağdaki algılayıcı düğümlerinin hızlı ölmesine neden olur. Bununla birlikte, KB'nin rolünü ağdaki diğer düğümlere dağıtarak düğümler arasındaki enerji tüketimini dengelemek amacıyla kullanılmıştır. Ayrıca LEACH, bir kümedeki kanal erişimini düzenlemek amacıyla Zaman Bölünmeli Çoklu Giriş (TDMA) protokolünü kullanmaktadır[18]. TDMA yuvalarını küme üyelerine atamak KB'lerin sorumluluğudur. KB ile bir üye eşler arası iletişim, tam da o üyeye atanan zaman aralığında gerçekleştirilir ve diğer üyeler bu sırada uyku modunda olur. Bu sayede

enerji kaybı azalır; (Şekil 2.2). LEACH, TDMA iletişim protokolü ayrıca kümeler arasındaki girişimi azaltmak için de kullanır. LEACH, daha fazla enerji harcayan Düz Mimari Protokollerinin dezavantajlarını ortadan kaldırmak için üretilmiştir [10,11].



Şekil 2.2: LEACH protokolü ve TDMA çizelgeleri.

KB, düğümler tarafından toplanan veriyi alarak daha küçük boyutlu ve anlamlı veriye dönüştürür, ardından veriyi daha düşük enerji tüketen alıcıya gönderir. LEACH, veriyi kısa mesafeler üzerinden göndermeye ve iletim sayısını azaltmaya çalışır, zira enerji tüketimi mesafeye ve veri boyutuna bağlıdır. Sonuç olarak LEACH'in ana sorunu, özellikle bu KB'ler alıcıdan uzak yerleştirildiği zaman KB'nin alıcıya doğrudan gönderimidir. Bununla birlikte, diğer KB'ler üzerinden alıcıya çok sekmeli iletim sağlamak bu sorunu çözecektir; burada KB veriyi alıcıya ulaştırana kadar yalnızca sonraki KB'lere iletir ve diğer KB'lerden gelen veriyi yeniden toplaması gerekmez. Doğrudan iletim ve diğer minimum enerji tüketimli yönlendirme protokolleriyle karşılaştırıldığında, LEACH, enerji kaybında önemli bir azalma elde etmektedir. Aşağıda LEACH'in ana özellikleri (avantajları / dezavantajları) ayrıntılı incelenmiştir [18].

2.2.1.1. LEACH'in Avantajları

- İletişimin çoğunu kümeler içinde kısıtlayarak ağ içinde ölçeklendirilebilirlik sağlar.
- KB'lerin düğümler tarafından alınan veriyi toplaması sonucu ağda üretilen trafik kısıtlanır. Bu sayede büyük boyutlu bir ağ, trafikte aşırı yüklenme olmadan dağıtılabilir ve düz topolojiden daha iyi bir enerji verimliliği elde edilebilir.

- Düzümden küme başına tek sekmeli yönlendirme yapıldığından enerji tasarrufu sağlanır.
- KB'nin rolünü diğer düğümlere dağıttığı için dağılımlılık özelliği gösterir.
- Ağ ömrünü üç şekilde artırır. Birincisi KB'nin (normal düğümlerden daha fazla enerji tüketen) rolünü diğer düğümlere dağıtır. İkincisi veri, KB'ler tarafından toplanır. Son olarak KB tarafından kendi üyelerine atanan TDMA, algılayıcıların çoğunu, özellikle de olay tabanlı uygulamalarda uyku moduna alır. Bu sayede ağ ömrü artırılabilir ve doğrudan iletme göre enerji kaybında 7 kattan fazla azalma elde edilebilir [18].
- Kümeleri oluşturmak için düğümlerin yer bilgisi gerekmez. Bu yüzden de güçlü ve basittir.
- Son olarak dinamik bir kümeleme yaklaşımıdır ve sürekli izleme gerektiren ve veri toplama işleminin düzenli olarak merkezi bir konumda gerçekleştiği uygulamalar için son derece uygundur.

2.2.1.2. LEACH'in Dezavantajları

- Büyük oranda küme başlarına güvenir ve küme başlarının arızalanması gibi sağlamlık sorunları ile karşılaşılır.
- Büyük ağlardaki dinamik kümelemelerde enerjinin verimsiz olmasına neden olan küme başı değişimleri ve hesaplamalara bağlı ek maliyetler doğurur.
- KB'ler alıcı ile doğrudan iletişim kurar; Bu yüzden alıcı ile tek sekmeli iletişim gereken büyük ölçekli ağlarda iyi çalışmaz.
- KB seçimi rastgeledir ve bunda enerji tüketimi hesaba katılmaz.
- Son olarak çok sekmeli kümeler arası iletişim gerektiren büyük bir alanı kapsayan uygulamalarda iyi çalışmaz.

2.2.1.3. LEACH'de İyileştirmeler

LEACH'in bazı dezavantajları nedeniyle, bu protokolün daha iyi çalışması için çok sayıda araştırma yapılmıştır. Bu araştırmalardan bazıları şunlardır: E-LEACH, TL-LEACH, M-LEACH, LEACH-C ve V-LEACH [10].

E-LEACH

E-LEACH protokolü KB seçim prosedürünü iyileştirmektedir. LEACH gibi bu protokol de kümelere bölünmüştür, burada ilk kümedeki tüm düğümlerin KB olma olasılığı

aynıdır. Ancak ilk kümeden sonra her düğümün kalan enerjisi farklıdır ve düşük enerjiye sahip düğüm yerine yüksek kalan enerjiye sahip düğüm KB olarak seçilir [18].

TL-LEACH

LEACH'te KB veriyi bir sekmede baz istasyona gönderir. Ancak TL-LEACH'te KB veriyi küme üyelerinden toplar ve KB ile baz istasyon arasında yer alan bir KB üzerinden baz istasyona yeniden yönlendirir [19].

M-LEACH

Daha önce bahsedildiği gibi LEACH'te KB veriyi bir sekmede baz istasyona gönderir. M-LEACH protokolünde, KB veriyi alıcıya diğer KB'leri röle istasyonları şeklinde kullanarak gönderir [20]. Bu protokolda KB'lerin baz istasyonlarından uzak olması ve çok miktarda enerji tüketmesi sorunu çözülmüştür.

V-LEACH

Yeni LEACH protokolü sürümünde, küme içinde bir KB bulundurmaya ek olarak, KB ömrünü tamamladığında KB'nin görevini üstlenen bir yardımcı KB vardır [21]. Bir KB ömrünü tamamladığında, küme kullanılmaz hale gelir, çünkü düğüm üyeleri tarafından toplanan veir alıcıya ulaşmaz.

LEACH-C

LEACH'in KB'lerin yerleri konusunda bilgisi yoktur. LEACH-C protokolü, küme başlarını ağ boyunca dağıtarak daha iyi bir performans üretebilir. Kurulum aşamasında her düğüm alıcıya kendi kalan enerjisini ve konumunu gönderir. Alıcı daha sonra merkezi bir küme formasyonu algoritması çalıştırarak ilgili dairenin kümelerini belirler. Ancak bu protokol ağdaki tüm algılayıcıların konum bilgisini gerektirdiğinden (normalde GPS tarafından sunulur) verimli değildir [20].

2.2.2. Doğrudan Yayılma

Doğrudan Yayılma (Directed Diffusion), KAA'daki bilginin toplanması ve yayılmasına yönelik veri merkezli bir yönlendirme protokolüdür. Alıcı, algılayıcılardan özel bir bilgi talep ettiğinde alıcıdan algılayıcılara akan verinin gerekliliklerine yönelik olarak geliştirilmiştir. Ana amacı, temel enerji tasarrufunu gerçekleştirerek ağ ömrünü uzatmaktır. Bu amaca ulaşmak için düğümler arasındaki etkileşimi mesaj takası ile sınırlı bir ortamda tutması gerekir. Yerleştirilmiş etkileşim, bu protokolün eşsiz bir özelliği olan çok yönlü iletimi sağlar. Bu eşsiz özellik, düğümlerin alıcının sorgularını yanıtlayabilmesi ile birlikte dikkat çekici bir enerji tasarrufu sağlar [19]. Alıcı sorgusuyla ilgili algılayıcılar arasındaki rotayı oluşturmak için dört aşama vardır; (A) ilgi yayılımı, (B) gradyan ayarı, (C) güçlendirme ve (D) veri teslimatı aşamalarının ayrıntılı açıklaması aşağıda verilmiştir.

2.2.2.1 İlgi Yayılımı

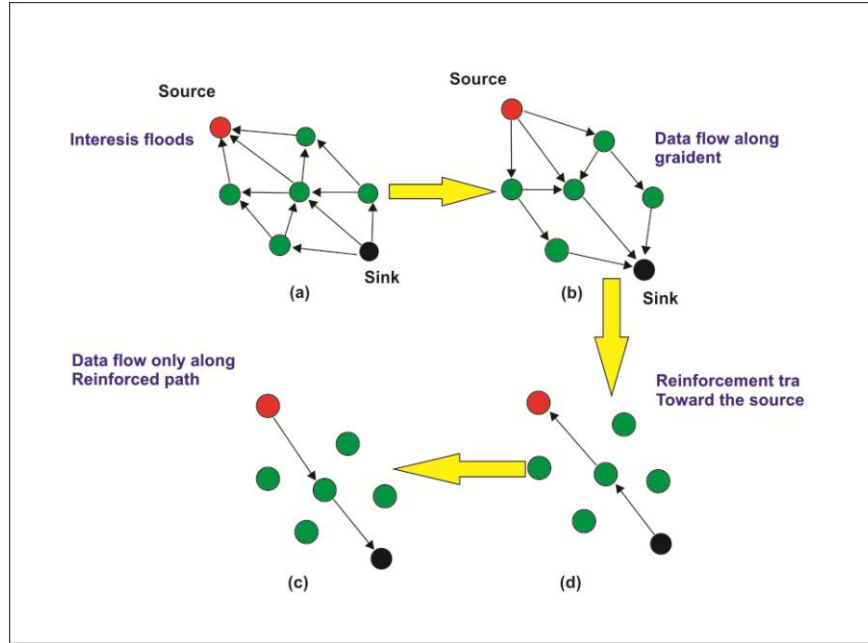
Bir alıcı bir olay saptadığında, ilgi mesajları başlatır ve bunları ağdaki tüm düğümlere iletir. Bu mesajlar açıklayıcı mesajlardır ve belirli görev için eşleşen veriye sahip düğümleri belirtir. Bu aşamada alıcı düzenli olarak ilgi mesajını yayınlar. İlgi mesajı alındığında, her algılayıcı düğümü bunu bir ilgi ön belleğine kaydeder. Bundan sonra düğümler bu mesajı, bu ilgi mesajı ile ilgili olan düğüme kadar diğer düğümlere iletir (Şekil 2.3a).

2.2.2.2 Gradyan Ayarı

Yerel kurallar temelinde, gradyan ayarında farklı teknikler kullanılır. Örneğin en yüksek kalan enerjiye sahip düğümler gradyan oluşumu için seçilebilir. Ağ boyunca ilgi yayılımı sırasında, kaynaktan alıcıya geri dönen gradyanlar oluşacaktır. Gözlemi ilgi mesajı ile eşleşen bir düğüm bir kaynak kod olur ve verisini gradyan yolu üzerinden alıcıya geri gönderir (Şekil 2.3b).

2.2.2.3 Güçlendirme

Gradyan oluşumu aşamasında, kaynaktan alıcıya birçok yol oluşur. Yani kaynak verisini birden fazla rota kullanarak alıcıya gönderebilir. Ancak Şekil 2.3'de de gösterildiği gibi, alıcı belirtilen yol üzerinden aynı ilgiyi yeniden göndererek belirli bir yolu güçlendirir; bu yolun seçiminde en iyi bağlantı kalitesi, bir komşudan alınan paketlerin sayısı, en düşük gecikme gibi belirli kurallar uygulanır. Bu yol boyunca her düğüm yalnızca güçlendirmeyi kendi sonraki sekmesine iletir [23]. Son olarak bu aşamada alıcı, birden çok yönlü teslimat sağlamak için birden fazla yol seçebilir.



Şekil 2.3: Doğrudan Yayılma protokolünün çalışması.

2.2.2.4 Veri Teslimatı

Güçlendirme aşamasından sonra, Şekil 2.3(d)'de gösterildiği gibi, kaynak ile alıcı arasındaki rota oluşturulmuştur ve veri iletim için hazırdır. Doğrudan Yayılmanın teknik özellikleri (avantaj/dezavantajlı yanları) aşağıda verilmiştir [24,25].

2.2.2.5. Doğrudan Yayılmanın Avantajları

- Tek bir düğümden veri toplamayı sağlamak için tasarlanmıştır.
- Çok amaçlı kablosuz algılayıcı ağlarında ve sorgu (“BELİRLİ BİR ALANDAKİ SICAKLIĞI VER” ya da “SİYAH İNEĞİ GÖREBİLEN VAR MI”) gönderen algılayıcı ağlarında iyi çalışır.
- Diğer yönlendirme algoritmalarının aksine, Doğrudan Yayılmada aynı anda birden fazla alıcı sorgu oluşturabilir ve veri alabilir; bu sayede tek bir ağ içinde eş zamanlı sorgular yönetilebilir.
- İlgiler/sorgular kaynaklar değil alıcı tarafından ve yalnızca bir istek olduğunda yayınlanır. Dahası, tüm iletişim komşular arasındadır, bu da adresleme gerekliliğini ortadan kaldırıp her düğümün veri toplamasına izin verir. Sonuç olarak her iki husus da enerji tüketimini azaltmaya katkıda bulunur.
- Kullanıcının ilgisi temelinde uygulamaya bağımlı rotalar sunar.

- Ne genel bir düğüm adresleme mekanizması ne de genel bir ağ topolojisi gerektirir. Ayrıca rotalar yalnızca bir ilgi olduğunda oluşur. Sonuç olarak enerji verimliliği elde edilir.
- Kullanıcının isteklerini karşılamak için ağ rotaları algılayıcı okuma değişimlerine göre değiştirilir.
- Yalnızca eşleşen bilgiye sahip düğümler, bilgi üretimine dahil olur.
- Çoğunlukla ilgi için belirli bir rota seçer. Bu yüzden ağdaki enerji tüketimini azaltır.

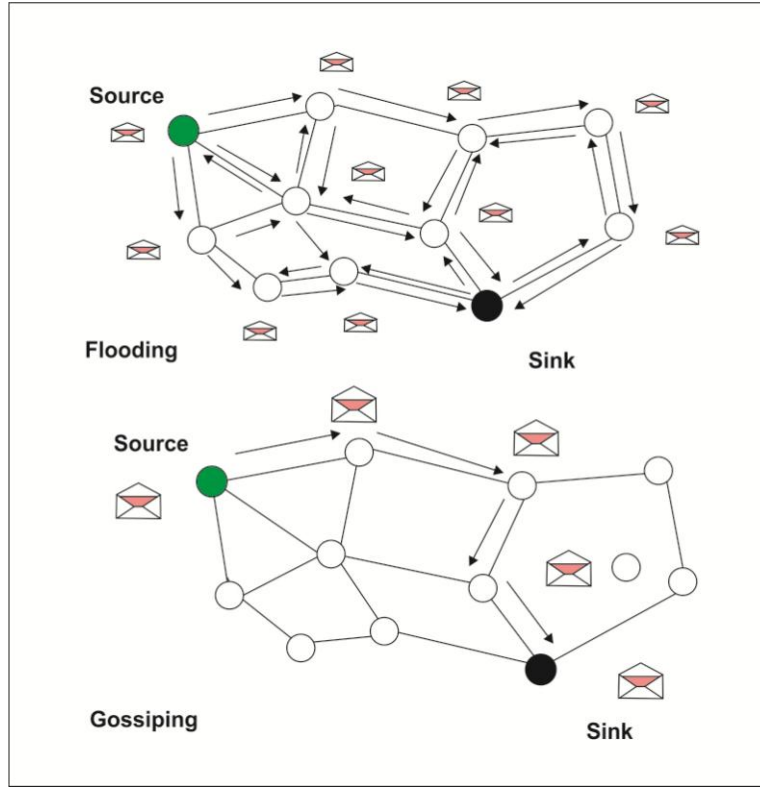
2.2.2.6. Doğrudan Yayılmanın Dezavantajları

- Genelde düz bir topolojiyi temel alır. Bu yüzden ölçeklenebilirlik ve tıkanma (özellikle alıcıya yakın düğümlerde) sorunları vardır.
- Veri ve sorgu için eşleştirme işlemi sırasında algılayıcıda bir aşırı yük sorunu oluşur.
- Doğrudan Yayılmada ilk ilgi düşük bir veri oranı içerir. Ancak ilgi yayılma aşamasının akış işlemi sırasında önemli bir yüklenmeye neden olur.
- İlginin her bir düğüme yayılması için gereken akış nedeniyle, enerji verimliliği açısından optimize değildir ve ilgi gradyanları ile alınan mesajların depolanması için yüksek miktarda bellek gerektirir.
- Çoğunlukla kaynak ile hedef arasındaki en kısa yolu seçer, bu da o yol üzerindeki düğümlerin ömrünün daha hızlı tükenmesine yolaçar.
- Doğrudan Yayılma sorgu tabanlı bir protokol olduğundan, ortam izleme uygulamaları gibi sürekli veri aktarımları gereken uygulamalarda (dinamik uygulamalar) iyi çalışmayabilir.

2.2.3. Gossiping Protokolü

Gossiping, bir veri röle protokolüdür. Bu nedenle, akış protokolü gibi yönlendirme tablolarına ve topoloji bakımına ihtiyaç duyulmaz. Akışın çökme gibi dezavantajlarını gidermek için üretilen bir akış geliştirme aracıdır. Akışta bir düğüm, alıcı düğüm başka bir düğümden aynı veriyi henüz almış olsa bile veriyi tüm komşularına yayınlar. Yayın, verinin hedefte alınmasına kadar devam eder [26]. Ancak Gossiping'te bir düğüm komşularından birini, paketi iletmek için rastgele seçer ve seçilen düğüm paketi aldığı anda başka bir düğümü rastgele seçerek paketi iletir. Bu işlem hedefe ya da sekme sayısı aşılanaya kadar devam eder. Sonuç olarak yalnızca seçilen düğümler/komşular

alınan paketi alıcıya iletir. Flooding'in tersine Gossiping, teke tek iletişim senaryolarında daha iyi çalışır ancak tekten çoğa iletişim senaryolarında iyi çalışmaz. Hem Flooding hem de Gossiping'teki paketi iletim mekanizmaları Şekil 2.4'te gösterilmektedir. Gossiping'in ana amacı güç tüketimini azaltmak ve yönlendirme sistemlerini olabildiğinde basit tutmaktır [22,27].



Şekil 2.4: Flooding ve Gossiping iletim mekanizmaları.

Ancak, Gossiping iletiminde, veri yayılmasından kaynaklanan veri gecikmesi sorunu yaşanır. [26];

2.2.3.1 Gossiping'in Avantajları

- Yönlendirme tablosu olmadığı için Gossiping iletiminde daha az enerji tüketilir.
- Flooding yöntemi ile yaşanabilecek çökme sorununu giderecek bir iyileştirme yöntemi olarak ortaya çıkmıştır.
- Gossiping'te yalnızca seçilen düğümler verinin alıcıya iletimine katkıda bulunur.
- Teke tek iletişimde iyi çalışır ancak tekten çoğa iletişimde iyi çalışmaz.

2.2.3.2. Gossiping'in Dezavantajları

- Sonraki sekme komşusu rastgele seçilir, yani bunlara kaynağın kendisi de dahil olabilir.
- Paket kaybı sorunu yaşanır.
- Gossiping'in önemli bir dezavantajı veri yayılmasına bağlı gecikmelerdir.

2.2.3.3. İyileştirilmiş Gossiping Protokolleri

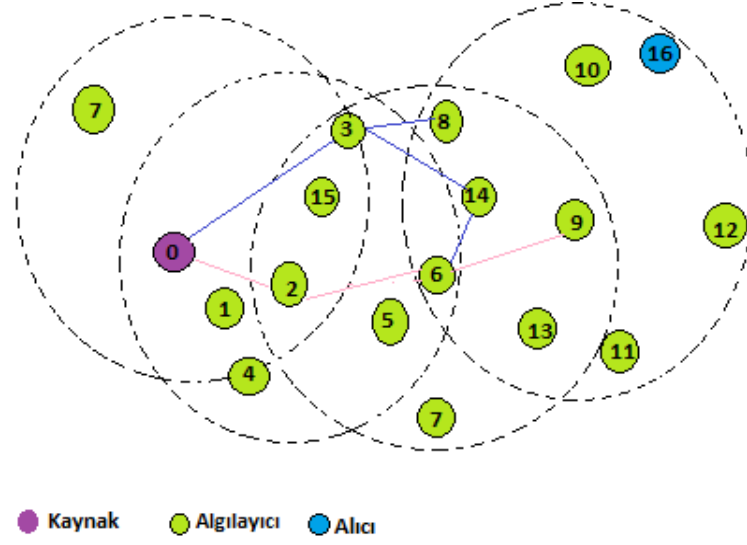
Gossiping protokolünü geliştirmek ve iyileştirmek amacıyla birçok protokol üretilmiştir. Bunların bazıları FLOSSIPING, SGDF, LGOSSIPING ve EL-GOSSIPING protokolleridir.

FLOSSIPING Protokolü

FLOSSIPING protokolü, Flooding ve Gossiping yönlendirme protokollerinin yaklaşımlarını birleştirerek iyileştirme sağlamıştır. Bir düğümde gönderilecek bir paket olduğunda, bir eşik belirler ve bunu paket üst bilgisine kaydeder, ardından paketi Gossiping modunda göndermek için rastgele bir komşu seçer, diğer taraftan diğer komşu düğümler bu paketi dinler ve rastgele bir sayı üretir. Rastgele sayıları üretilen eşikten daha düşüktür komşular paketi Flooding modunda yayınlar. Sonuç olarak Flossiping, Flooding'deki paket yükü sorununu ve Gossiping'deki gecikme sorununu iyileştirir [28].

SGDF Protokolü

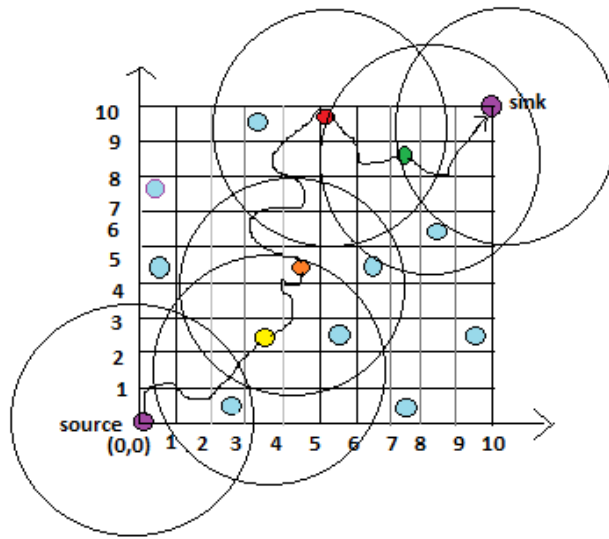
SGDF (Single Gossiping with Directional Flooding) yönlendirme protokolü Ağ Topolojisi Başlatma ve Yönlendirme olmak üzere iki aşamadan oluşur. İlk aşamada her düğüm bir gradyan üretir (alıcıya olan sekmelerin sayısını gösterir). İkinci aşamada ise paketi iletmek için, SGDF tekli Gossiping ve yönlü Flooding yönlendirme şemalarını kullanır. Böylelikle SGDF kullanılarak (Şekil 2.5) yüksek paket teslimat oranı, düşük mesaj karmaşıklığı ve kısa paket gecikmesi gibi iyileştirmeler elde edilir [22].



Şekil 2.5: SGDF'de yönlendirme senaryosu.

LGOSSIPING Protokolü [26]

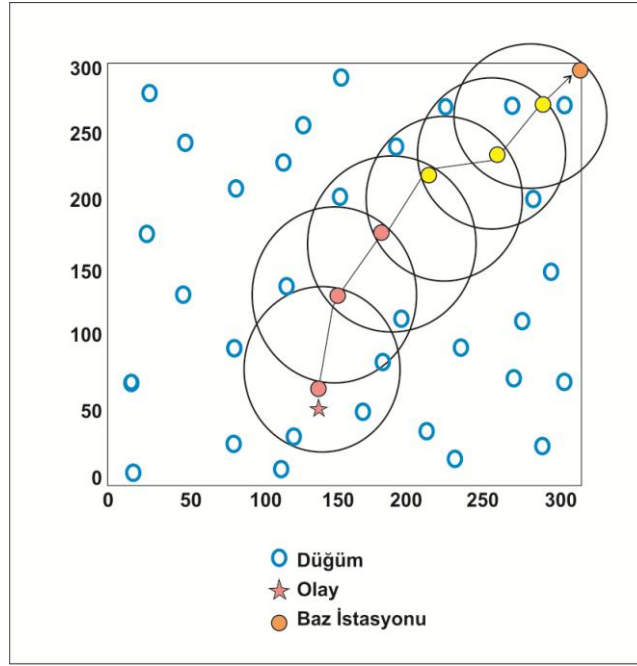
LGOSSIPING (Lokasyon tabanlı Gossiping) protokolünde, bir düğümde gönderilecek bir olay olduğunda, düğüm kendi iletim yarıçapındaki bir komşu düğümü rastgele seçer. Komşu düğüm bu olayı aldığı anda, kendi iletim yarıçapındaki başka bir düğümü rastgele seçer ve olayı gönderir. Bu işlem alıcıya kadar devam eder. Sonuç olarak gecikme sorunu bir dereceye kadar çözülür. Şekil 2.6'da LGOSSIPING'in iletim mekanizması gösterilmektedir.



Şekil 2.6: LGOSSIPING'de veri yönlendirmenin şematığı.

ELGOSSIPING Protokolü [29]

ELGOSSIPING protokolünde, bir düğüm bir olay algıladığında ve bunu göndermek istediğinde, kendi iletim yarıçapındaki ve baz istasyonuna / alıcıya en kısa mesafedeki bir komşu düğümü seçer. Komşu düğüm olayı algıladığında, kendi iletim yarıçapında ve yine alıcıya en kısa mesafedeki başka bir komşu düğümü seçer. Olay, alıcıya kadar aynı şekilde ilerler (Şekil 3.6.).



Şekil 2.7: ELGOSSIPING'de yönlendirme.

2.2.4 Enerji Verimli Algılayıcı Yönlendirme Protokolü

Enerji Etkin Algılayıcı Yönlendirme (EESR) klasik bir yönlendirme algoritmasıdır. [29] Ve özellikle enerji tüketimini ve veri gecikmesini azaltmak ve KAA 'da ölçeklenebilirlik sağlamak için önerilmiştir. Temelde Ağ Geçidi, Baz İstasyonu, Yönetici Düğümler ve Algılayıcı Düğümlerinden oluşur [23]. Bunların görevleri şunlardır:

- Ağ Geçidi: Yönetici düğümlerden mesajları iletir ya da Baz İstasyona başka ağlar oluşturur.

- Baz İstasyon: Normal algılayıcı düğümlerine göre fazladan teknik özellikleri vardır. Ağ Geçidine/Ağ Geçidinden mesajlar gönderir/alır. Dahası algılayıcı düğümlerine sorgular gönderir ve algılayıcı düğümlerinden veri alır.
- Yönetici Düğümler ve Algılayıcı Düğümleri: Ortamdan veri toplar ve bu verileri Baz İstasyonuna 1 sekme mesafeye kadar birbirine gönderir.

Uygulama alanı 2 boyutlu (x, y) koordinatlara göre dört kuadrana bölünmüştür; (+ +), (+ -), (- -) ve (- +). Baz İstasyon merkezde (koordinatta) yer alır. Dahası her kuadranda, ortadaki Bas İstasyonu gösteren sektörlere bölünmüştür; bunların sayısı, bir mesajın baz istasyondan kuadrandaki en uzak noktaya iletilmesi için gereken minimum sekme sayısıyla belirlenir.

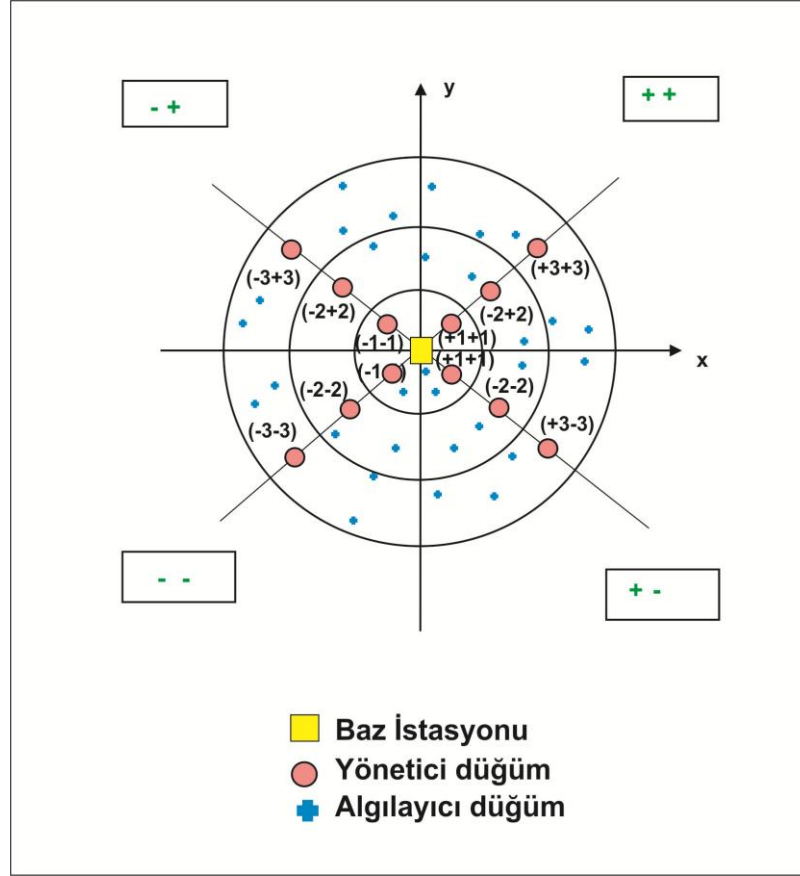
Yönetici Düğümler, kuadranın çapraz hattı üzerinde birbirinden bir sekme mesafeyle her sektörün ortasında yerleştirilir (önceden belirlenir), diğer düğümler ise uygulama alanında rastgele dağıtılır (Şekil 2.8) [29].

Şekil 2.8’de gösterildiği gibi, her kuadranın üç sektörü vardır. Buna göre, Baz İstasyon en uzak düğüm ile en az 3 sekmede iletişim kurabilir. Her sektörün, Baz İstasyondan aldığı, kuadrana adı ve baz istasyona olan mesafeyle belirlenen kendi kimliği vardır. Örneğin 1 sekme mesafedeki sektörlerin adları (+1 +1) sektörü, (+1 -1) sektörü, (-1 -1) sektörü ve (-1 +1) sektörleridir.

Her algılayıcı düğümü, 1 sekme mesafedeki komşusuna bir “HELLO” mesajı yayımlayarak Tablo 2.1’de gösterilen kendi EESR tablosunu oluşturur. Tablo baz istasyona olan mesafeyi, Kuadrana Adlarını, Sektör Kimliğini ve Yönetici Düğüm Adlarını içerir.

2.2.4.1. Algoritma

Düğümler dağıtıldıktan sonra, Baz İstasyon her düğümün bağlı yön bilgisini ve sektör kimliğini gönderir, ardından her düğüm kendi EESR tablosunu oluşturur. Bir düğüm bir olay saptadığında, olayın gönderileceği sonraki düğümü belirlemek için kendi EESR tablosunda yer alan 1 sekme mesafedeki tüm komşu düğümlerin sektör kimliğini araştırır. Düğüm, şu üç prosedürden birinde sonraki düğümünü seçer:



Şekil 2.8: Boyutlu (x, y) koordinat bazlı düğümlerin konumları.

Tablo 2.1: Kuadran adları, sektör kimliği ve yönetici düğüm adları.

Baz istasyona olan mesafe	Kuadran adı	Sektör Kimliği	Yönetici Düğüm adı	Baz istasyona olan mesafe
1 sekme	(++)	(+1+1)	Sektör	+1 +1M.N
	(+-)	(+1-1)	Sektör	+1 -1M.N
	(--)	(-1-1)	Sektör	-1 -1M.N
2 sekme	(-+)	(-1+1)	Sektör	-1 +1M.N
	(++)	(+2+2)	Sektör	+2 +2M.N
	(+-)	(+2-2)	Sektör	+2 -2M.N
3 sekme	(--)	(-2-2)	Sektör	-2 -2M.N
	(-+)	(-2+2)	Sektör	-2 +2M.N
	(++)	(+3+3)	Sektör	+3 +3M.N
	(+-)	(+3-3)	Sektör	+3 -3M.N
	(--)	(-3-3)	Sektör	-3 -3M.N
	(-+)	(-3+3)	Sektör	-3 +3M.N

- Bir Yönetici Düğüm 1 sekme mesafede varsa, sonraki sekme kendisi olacaktır.
- Bir Yönetici Düğüm 1 sekme mesafede yoksa, sonraki sekme olabilecek aynı sektörde yer alan 1 sekme mesafedeki normal bir düğümü kontrol eder.
- Bir Yönetici Düğüm 1 sekme mesafede yoksa, sonraki sekme olabilecek kendi sektörü dışında yer alan ama Baz İstasyonuna yakın başka bir düğümü arar.

Aramada, aynı kuadranda yer alan düğümler tercih edilir. Kendinden sonraki komşu düğüm seçildikten sonra, ilk düğüm olayı yalnızca bu seçilen düğüme gönderecektir. Seçilen düğüm olayı aldığı anda, aynı prosedürü tekrarlayarak kendi sonraki tek sekmesini seçer ve olayı gönderir. Bu işlem Baz İstasyon olayı alana kadar devam eder. Ancak bir Yönetici Düğüm olayı alırsa, olay Baz İstasyona kadar yöneticiden-yöneticiye gönderilir [29].

2.2.4.2. EESR'nin Avantajları

EESR protokolü, uygulama alanını sektörlere böler; bu sayede ölçeklenebilir. Enerji verimlidir ve bu özelliği üç yolla elde eder: Birinci yolda olayı yalnızca bir düğüme gönderir ve iletmez; ikincisinde Yönetici Düğümler veriyi ön tanımlı bir en kısa yol üzerinden yeniden yönlendirir; ve sonuncusunda ilk olay gönderildikten sonra normal düğümler, kendi EESR tablolarını kullanarak sonraki düğümü kolayca seçer. Bu yöntemde, daha az enerji tüketilir ve ağ ömrü uzundur. Protokol gereği veriyi aynı rotadan göndermemek ve bu düğümlerin enerjisini tüketmemek için verinin iletileceği başka rotalar seçilir. EESR iletişimi birebir kurarak birinci olay gönderildikten sonra, sonraki sekmenin kolayca bulunmasını sağlar. Bu nedenle, iletişim düşük gecikmelidir.

2.2.4.3. EESR'nin Dezavantajları

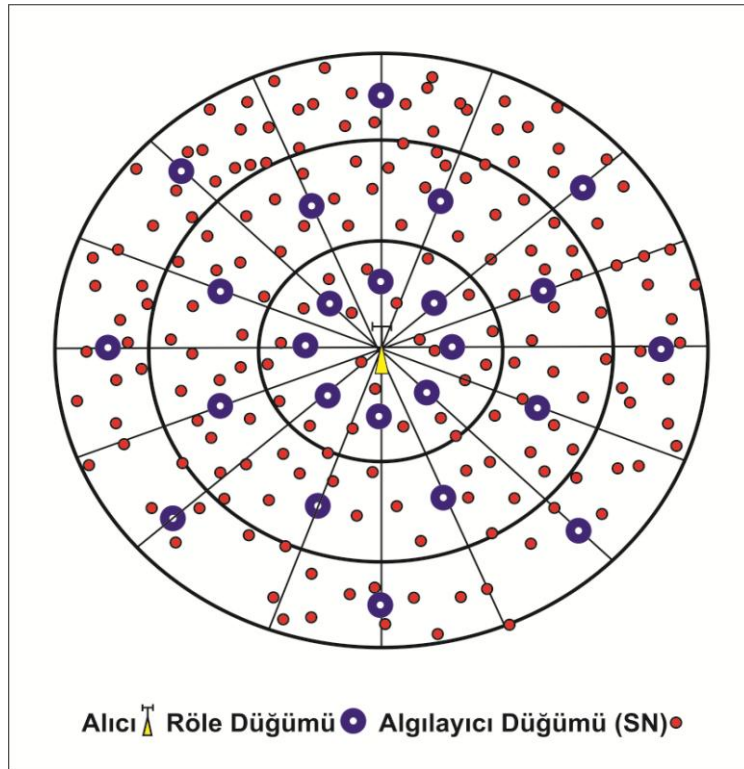
- Olayı saptayan düğümün 1 sekme mesafesindeki tüm düğümler iletim menziline dışında olabilir. Bu yüzden sonraki düğümü seçmek için belirli bir kriter yoktur.
- En uzak sektördeki bir düğüm bir olay saptarsa ve sonraki sekme daha düşük bir sektörde yer alıyorsa, düşük sektörün enerjisinin bitmesi halinde veri kaybolacaktır.
- En uzak sektöre yerleştirilen normal düğümler bir olay saptar ve bunların sonraki sekmesi her defasından rastgele bir Yönetici Düğüm olursa, bu Yönetici Düğümlerin enerjisi daha çabuk biter, çünkü bunlar olayı Baz İstasyona kadar yöneticiden - yöneticiye (röleden-röleye) gönderir.

- Enerji tüketiminde bir denge yoktur, yani bazı düğümler enerjilerini diğer düğümlerden daha önce tüketir.

2.2.4.4. EESR'de İyileştirmeler

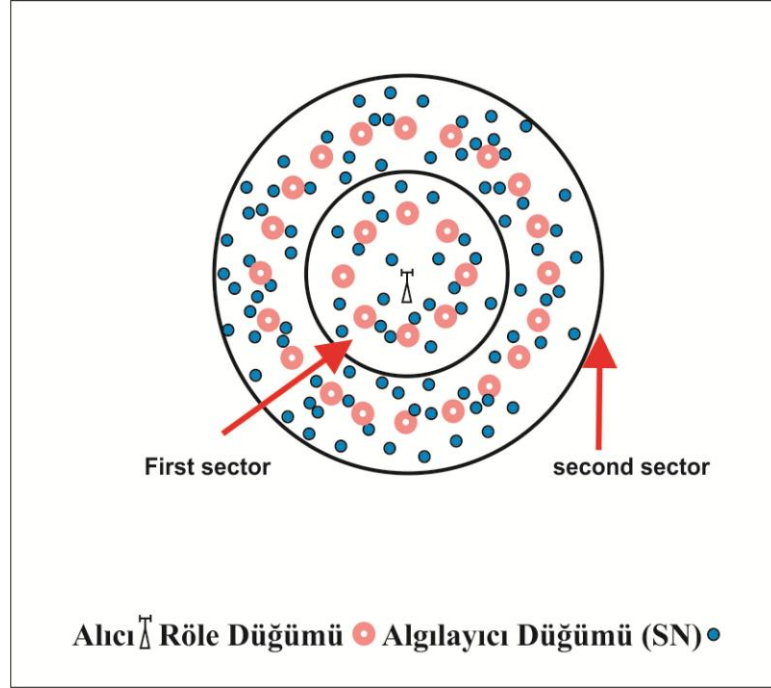
Bu bölümde, EESR'in dezavantajları nedeniyle eşmerkezli sektörler oluşturarak EESR'de yeni bir optimum yönlendirme algoritması önerilmiştir.

1. Çözüm, Her kuadrantdaki Otobanların (diyagonallerin) sayısını artırmak :
Bu çözümde ikinci ve üçüncü sorunlar (EESR'nin yukarıda bahsedilen dezavantajları) çözülmüş oluyor. Ancak birinci sorun, çözülmesi gereken en önemli sorundur. (Şekil 2.9)



Şekil 2.9: 16 Otobanlı ağ.

2. Çözüm: Şekil 2.9'da gösterilen otobanlardan bağımsız olarak adil sayıda röle düğümü (Yönetici Düğümler) belirlemektir. Bu çözümde birinci ve sonuncu dezavantajlar giderilmiştir [29]. Yeni geliştirilen iyileştirilmiş protokolün yönlendirme işlemi Şekil 2.10'daki akış şemasında gösterilmektedir.



Şekil 2.10: Her sektördeki röle düğümleri.

3. MALZEME VE YÖNTEM

KAA ile yapılan uygulamalar askeri, ticari ve tıbbi uygulamalar olmak üzere üç kategoriye ayrılır. Askeri alanda KAA uygulamaları olarak iletişim, komut ve istihbarat savunması; Tıbbi alanda KAA uygulamalarından bazıları uzak bölgelerdeki engelliler için tasarlanmış sağlık hizmetleri sistemi, yaşlı insanlar için tasarlanmış akıllı ortam, doktor ve sağlık çalışanı iletişim ağları ve hasta izleme sistemi; Ticari alandaki KAA uygulamalarından bazıları ise güvenlik sistemleri, yangın güvenliği sistemleri, çevre kirliliği izleme sistemleri (kimyasal, mikrobik ve nükleer kirlilik), araç izleme, denetleme ve kontrol sistemleri, trafik kontrol sistemi ve doğal afet sistemleri sayılabilmektedir (örneğin deprem ve sel) [30]. KAA ile gerçekleştirilen geniş uygulama çeşitliliği sayesinde çok sayıda esnek parametre içeren birçok protokol geliştirilmiştir. Ancak bazı parametreler, geniş kullanım alanı nedeniyle, birçok uygulama içerisinde (ortak parametre olarak) bulunabilir ve büyük önem arz eder. Bununla birlikte Kablosuz Algılayıcı Ağları, taşınabilir güç kaynakları ve tekrar şarj edilebilir piller kullanır. Ancak teknolojik kısıtlamalar nedeniyle bu piller yalnızca kısa süreli enerji tedariki sağlamaktadır. Bu nedenle, enerjinin böyle ağlarda en iyi şekilde kullanılabilmesi son derece önemlidir [31].

3.1. ENERJİ TÜKETİMİ İÇİN ETKİN PARAMETRELER VE DEĞERLENDİRİMİ

Yönlendirme konusunda ciddi bir problem, kaynak ile hedef arasında uygun bir rotanın çizilmesidir. Bu amaca genellikle bir çok algoritma ve protokol aracılığıyla ulaşılır. Doğrudan ve etkin yönlendirme, dayanıklılık ve enerji tüketimi gibi faktörleri de etkiler. Bu bölümde, tezin asıl amacı olan enerji tüketimini etkileyen parametreleri inceledik.

3.1.1. İki Düğüm Arası Mesafenin Tahmini

Kablosuz Algılayıcı Ağlarında performansa etki eden en önemli parametreler ve bu parametrelerin kablosuz algılayıcı ağları üzerlerindeki etkileri detaylı olarak bu kısımda

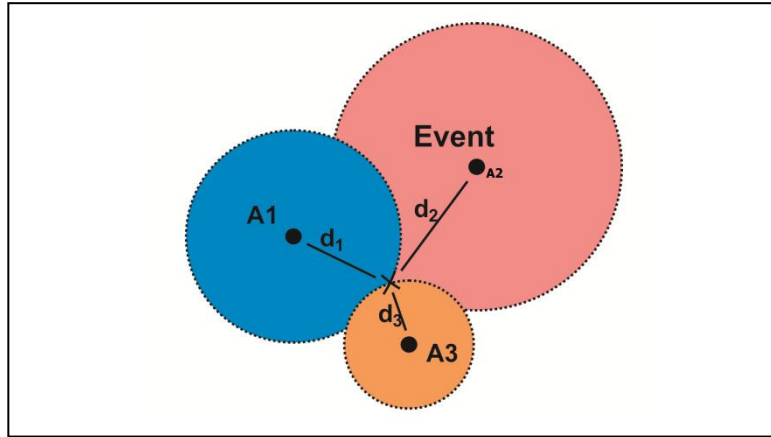
incelenmiştir. Burada incelenen parametreler, genelde uygulamaya bağlı olarak değişirler ve performans üzerinde farklı etkilere sahiptirler.

3.1.1.1. İşaret Gücü

Yayılan bir paket alındıktan sonra, ilk düğüm işaret gücünü kontrol eder ve iki düğüm arası mesafeyi elde edilen güç ile saptamaya çalışır [7,10]. Doğal olarak, uzak düğümlerden alınan işaret gücü yakın düğümlere göre daha azdır. [8, 15] çalışmalarında işaret gücü ile mesafe arasında ilişki (3.1)'de tanımlanmıştır.

(3.1)

Burada P_t ve P_r sırasıyla gönderilen ve alınan işaretlerin güçleri, G_r ve G_t sırasıyla alıcı ve vericinin işaret kazançları, λ dalga boyu ve L ağ kayıp oranını göstermektedir. Büyük P_r değeri, vericiye daha yakın olmak demektir.



Şekil 3.1: Değişik sinyal güçleri ile yayılımın şematik görüntüsü.

Şekil 3.1'de görüldüğü gibi, daha düşük işaret güçlü A_2 düğümü yakın iki düğümden (A_1 ve A_3) ve komşu x düğümünden daha uzaktadır.

3.1.1.2. İki Düğüm Arasındaki Ortalama Gecikme

Gecikme; bir uçtan diğer uca paketim iletimi ve alımı boyunca harcanan süre olarak tanımlanır. Ortam gürültüsü ve trafik yükü gibi faktörler gecikme süresini etkileyebilir. Diğer bir deyişle, uçtan uca gecikme süresi paket kaynaktan çıktığı anda başlar ve alıcı düğüm paketi aldığı zaman biter. Gecikme şu şekilde elde edilir:

$$AVG_Dend-end= [dtrans+dprop+dproc] \quad (3.2)$$

$AVG_D_{uçtan\ uca} = uçtan\ uca\ gecikme$

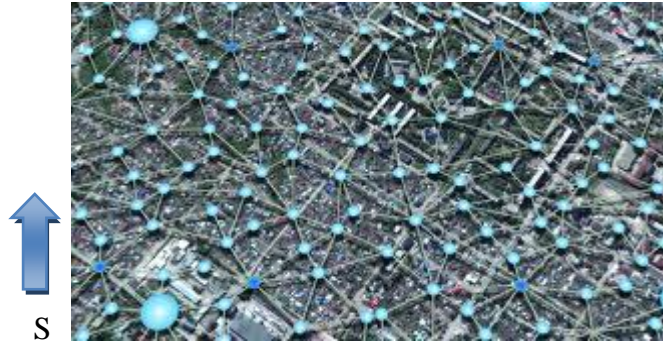
$AVG_D_{trans} = aktarım\ gecikmesi$

$AVG_D_{prop} = yayılım\ gecikmesi$

$AVG_D_{proc} = işlem\ gecikmesi$

3.1.1.3 GPS

Düğümmler arası mesafe tahmini, her bir algılayıcıya GPS donanımı eklenerek de yapılabilir. GPS teknolojisi uydulardan gelen iletim bilgilerini ve alım zamanını karşılaştırır ve ikisi arasındaki farktan yararlanılarak GPS alıcısının uyduya göre konumu hesaplanır. Birden fazla uydudan alınan mesafe bilgisi ölçülerek algılayıcının yeri bulunabilir ve bir elektronik harita üzerinde gösterilebilir. GPS uyduları her 24 saatte 2 kere Dünya etrafında tur atarlar ve bilgi gönderimi yaparlar.

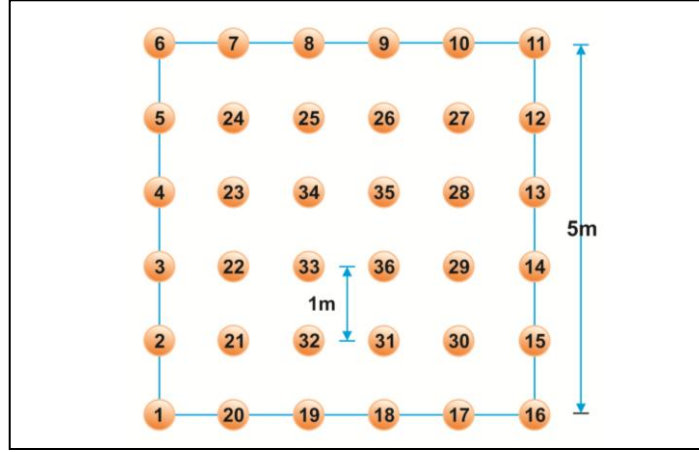


Şekil 3.2: Kentsel alanlarda algılayıcıların dağılımı.

Enlem, boylam ve rakım bilgilerinin doğru olarak elde edilebilmesi için en az dört uydudan bilgi alınması gerekir. Bir örnek olarak, Şekil 3.2’de kentsel bir alandaki algılayıcı dağılımı ve GPS kullanarak şehirdeki algılayıcıların enlem ve boylam bilgilerinin pozisyon haritasında nasıl elde edilebileceği gösterilmiştir.

3.1.1.4. Yerleşim ve Dağılım

Kablosuz Algılayıcı Ağlarının dağılım düzenleri temel olarak tekdüze, rastgele ve ızgara biçiminde olabilir. Izgara dağılım durumunda, her iki düğüm arasındaki mesafe tahmin edilebilir ve her bir düğümün pozisyonu başlık bilgisinden adım adım gidilerek hesaplanabilir. Şekil 3.3 düğümler arasındaki mesafelerin sabit olduğu bir ızgara yerleşim biçimini göstermektedir.



Şekil 3.3: Bir kablosuz algılayıcı ağında ızgara yerleşimi.

Yerleşim genellikle ağın uygulama amacına göre belirlenir. Örneğin askeri uygulamalarda, uçaklar bölgeye rastlantısal olarak dağılmış algılayıcılar taşırken, su altı algılayıcıları gibi uygulamalarda tekdüze, kentsel ağlarda ise ızgara yerleşimi (Şekil 3.3) tercih edilir. Rastgele dağıtım sistemi, genellikle Poisson dağılımını kullanır; ayrıca bir binom olasılığı varsayımı yapabilir. Poisson, rastgele dağılım için bir tahmin aracı olmaktan başka, zamanda ve konumda rastlantısal olarak gerçekleşen olaylar için güvenilir bir olasılık modelidir. Genellikle kablosuz algılayıcı ağlarını daha detaylı anlamak veya simüle etmek için kullanılan Poisson (3.3) deki gibi tanımlanabilir:

$$P(X=x) = e^{-\mu} \mu^x / x! \quad (3.3)$$

Burada $x=0, 1, \dots, n$ tamsayıyı, ($e=2.71828$) doğal logaritma tabanını göstermektedir.

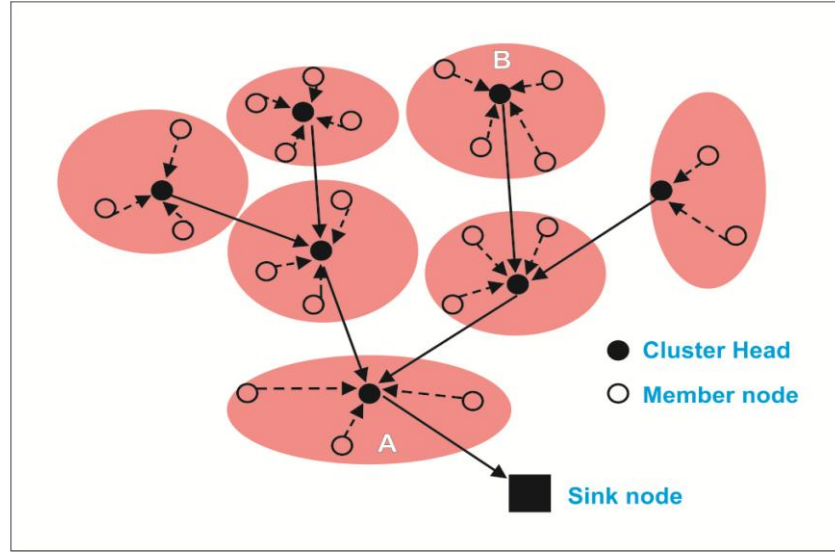
3.1.1.5. Olasılık

Bazı durumlarda, mesafe olay sıklığı üzerinden tahmin edilebilir. Örneğin bir yangın durumunda, yangının motor odasında olma ihtimali o kadar yüksektir ki, olasılıksal olarak olayın olduğu yere olan uzaklık tahmin edilebilir. Genel olarak, yüksek riskten dolayı, olasılık tahmini diğer tahminlerle birlikte yapılır.

3.1.1.6. Küme Başı Sayısı

Geçilen kümelerin bilgileri her bir paket başlığına kaydedilir. Bu sayede, komşu ve komşu olmayan kümeler kullanılarak ağın yoğunluğundan yola çıkarak mesafe hesaplanabilir. Bir düğümden bir paket alındıktan sonra her bir paketin başlık kayıtları

toplam geçilmiş olan baş kümesini gösterir. Bu sayede her bir küme başı arasındaki mesafe tahmin edilebilir.



Şekil 3.4: Baş küme düğümleri tarafından geçilen adımların sayısı.

Örneğin, düğüm A (Şekil. 3.4) baş-kümeden bir paket alır ve her iki küme arasındaki ortalama mesafe 10 metredir; dolayısıyla A ve B düğümleri arasındaki tahmini mesafe 20 metredir.

3.1.1.7. Mesafe Tahmin Yöntemlerinin Karşılaştırılması

Bir önceki bölümde mesafe tahmini yöntemleri incelenmiştir. Bu yöntemler belirli uygulamalar için uygundur; ancak hiçbirisi mükemmel değildir. Örneğin, mesafe tahmini için GPS metodu çok maliyetli (algılayıcı sayısı arttıkça maliyet daha da artmaktadır) olduğundan önerilmemektedir. Mars'ta su aramak gibi önemli ve kritik amaçlar için fiyat çok önemli değilken; vahşi hayvanları gözlemlemek söz konusu olduğu zaman fiyat konusu daha önemli olabilir. Mesafenin yerleşim ve dağılım ile bulunması ızgara yerleşimli KAA için kolay olmasına rağmen, çoğu uygulamada bu ağları kullanmak uygulanabilir değildir (algılayıcıların uzak yerlere koyulması ve bir olayın incelenmesi gibi). Bu yüzden parklar, bahçeler, yeşil alanlar, spor tesisleri ve çiftlikler gibi rastlantısal dağılımlar için bu hesaplamalar uygulanabilir değildir. Olasılık tahmini yöntemleri belirli özel uygulamalar için uygundur. Hatalı sonuç verebilme riski nedeniyle, olasılık yöntemi yanıltıcı olabilir; bu nedenle, birçok uygulama için tavsiye edilmez. Paket tarafından geçilen adımlardan yola çıkarak mesafenin hesaplanması yöntemi (Baş-küme adımı) kümeleme söz konusu iken oldukça işe yarar. Daha önceden

belirtildiği gibi, işaret gücü yöntemi mesafenin hesaplanması için önerilir. Kompleks hesaplamalar yapılmıyorken, elde edilen sonuçlar diğer yöntemlere göre daha mantıksal ve doğrudur.

Tablo 3.1: Mesafe tahmin yöntemleri.

YÖNTEM	Kısıtlama	Uygulama	İşlem hacmi	Hassasiyet
GPS	Maliyet	Yerleştirme	Düşük	Yüksek
Sinyal gücü	İşlem	Mesafenin belirlenmesi	Orta	İyi
Yerleşim ve Dağıtım	Uygulama	Kentsel algılayıcıların dağılımı, savaşı meydanları, su altı	Yerleşime göre, ızgara için düşük	Sadece ızgaralar için iyi
Olasılık	Güvenilirlik	Olaya kadar olan mesafenin tahmini	Yüksek	Düşük
Baş-küme adımı	İşlem ve iletim	Geçilen adımların tahmini	Orta	Orta
Uçtan uca gecikme	İşlem	Tahmini Gecikme	Orta	İyi

3.1.2. Düğümlerde Uygun Enerji Tüketim Modeli Uygulamak

Enerji, KAA'lerde dayanıklılık, güvenilirlik ve optimum rotanın tanınması probleminde doğrudan ve ciddi bir etkiye sahip olduğu için önemli bir konudur. Bu sebeple kablosuz algılayıcı ağlarında enerji yönetimi ön plana çıkmıştır. Enerji konusu tek bir düğüm ile değil, ağdaki ortalama enerji tüketimi (bütün düğümlerin toplam tüketimi) ile ilgilidir. Diğer bir deyişle, KAA'da enerji tek bir düğümün enerjisi ve ağın toplam enerjisi olmak üzere iki farklı şekilde incelenebilir.

Önemli olan Bu konuda, dayanıklılık ile ağın enerji hatırlatma arasında doğrudan bir ilişki olduğudur. Dayanıklılığı hesaplamak için kullanılan bir yöntem ise eğer bir

düğümün enerjisi %10'un altına düşerse onu yok saymaktır. Bu nedenle, düğüm enerjisi ve nasıl elde edildiği çok önemlidir.

Kablosuz iletişim teknolojisindeki son gelişmeler ağları daha küçük algılayıcılarla oluşturmayı mümkün hale getirmiştir. Bu ağlar; askeri, medikal, endüstriyel, meteorolojik, ticari takip ve ev güvenliği uygulamalarında kullanılmışlardır [7, 8]. Her ne kadar bu teknoloji büyük bir gelecek vaat ediyor olsa da, hala kablosuz algılayıcı ağlarını uygulanabilir hale getirmek için çözülmesi gereken problemler vardır. Bu problemlerden biri algılayıcıların bataryalarına bağlı olması, diğeri de algılayıcıların ulaşamayacak yerlere koyulması nedeniyle bakım yapmanın mümkün olmamasıdır. Minimum enerji tüketimi, sistem tasarımında en öncelikli konulardan birisidir. Heinzelman ve arkadaşları en iyi bilinen enerji tüketim modellerinden birisini ortaya koydular ve bu model düşük enerji tüketimli ağ protokollerinin tasarlanması konusunda geniş bir ilgi uyandırdı [10, 11]. Bazı farklı modeller anlatılmakla birlikte, algılayıcıların enerji tüketimini bulma konusunda literatürde henüz yeterince çalışma bulunmamaktadır. Anastasia ve arkadaşları bir algılayıcı düğüm tarafından tüketilen akım tüketimini ölçmek için bir voltmetre kullandılar [13]. Başka bir çalışmada [12], değişik durumlarda algılayıcıların güç tüketimi osiloskop yardımı ile ölçüldüğü halde, herhangi bir istatistiksel doğrulama yapılmamıştır.

Enerji tüketimi için değişik ölçümler kullanılarak (watt / joule / volt) farklı şekillerde enerji hesaplanabilir. Bu bölümde, kullanılan enerji tüketim modellerinin performansları karşılaştırmalı olarak Tablo (3.1) de yapılmıştır.

3.1.2.1. Klasik Enerji Tüketimi Modeli

Heinzelman ve arkadaşları, iletişim alt sistemlerinin enerji tüketiminde belirleyici rollere sahip olabileceği gözleminde yola çıkarak bir enerji tüketim modeli (Tablo 3.2) ortaya koydular [10, 11].

Tablo 3.2: Radyo özellikleri, klasik model.

Radyo modu	Enerji Tüketimi
İletim Elektroniği (E Tx-elec) Alıcı Elektroniği (E Rx-elec) E Tx-elec= E Rx-elec= E elec	50 nJ/bit
Aktarım Kuvvetlendirici (E amp)	100 pJ/bit/m ²
Boşta (E amp)	40 nJ/bit
Uyku	0

Bu model, radyoları daha düşük bir güç tüketimi ile tanıyabildiğinden bilinen bazı standart tanımlardan daha iyidir (örn. Bluetooth) [8, 15]. Olağan bir başlangıç model tarafından tanınır, ancak hala bir kablosuz algılayıcı ağının içindeki fiziksel radyo davranışının doğrulanması gerekir. Uygulamaya bağlı olarak, CPU ve algılayıcıya tamamen reddedilir ya da düğüm tarafından enerji tüketimini anlamak için kullanılır. Bu nedenle, bu elemanlar tarafından yapılan enerji tüketiminin bir kısmının radyo modeli ile birlikte çalıştırılması gerekir. Çünkü sadece radyonun değil, bütün sistemin bir güç kaynağının olması gerekir.

3.1.2.2. μ AMPS Özel Modeli

Özel platformlar için tasarlanmış olan (μ AMPS Kablosuz Algılayıcı Düğümü) bir model Shih ve arkadaşları tarafından önerilmiştir. Platformda güçlü bir mikroişlemci (ARM SA 1110, clock speed 59-206 MHz) bulunmaktadır. Mikrodenetleyici enerji tüketimi, radyo ve enerji kaçağı modelde dikkate alınmaktadır. [9] (Tablo 3.3)

Tablo 3.3: μ AMPS modeli algılayıcı durumları.

Durum	SA-1110	Algılayıcı , A/D	Radyo	Pk (mW)
Aktif	Aktif	duyu	tx/rx	1040
Hazır	Boşta	duyu	Rx	400
Monitör	Uyku	duyu	Rx	270
Gözlem	Uyku	duyu	Kapalı	200
Derin Uyku	Uyku	Kapalı	Kapalı	10

1-bit İletim/Alım için enerji tüketimi model tarafından elde edilememiştir. Herhangi bir oranda, platform tarafından kullanılan hız 1 Mbps olarak alınmıştır; yani metod ile 1-bit

için ihtiyaç duyulan enerjiyi bulmak mümkündür. Güç değeri ile, 1-bit için gerekli olan iletim/alım enerji tüketimi hesaplanabilir [11, 16].

$$\text{Enerji} = \text{Güç} \times \text{Zaman} \quad (3.4)$$

Burada, Güç watt cinsinden, zaman ise saniye cinsinden hesaplanır. 1-bit göndermek veya almak için gerekli zaman $1 / 1 \text{ Mbps} = 1 \mu\text{sec}$

$$\text{Enerji Tbirbit} = 1040 \times 1 \times 10^{-3} \text{ W} \times 1 \times 10^{-6} \text{ sn.}$$

$$\text{Enerji Tbirbit} = 1.04 \mu\text{J/bit}$$

$$\text{Enerji RbirbitHazırdurum} = 0.4 \mu\text{J/bit}$$

$$\text{Enerji Rbitbitİzlemedurumu} = 0.27 \mu\text{J/bit}$$

3.1.2.3. Mica2 Özel Modeli

Polastre ve arkadaşları tarafından ileri sürülen bu model iletim, alım, dinleme, veri örnekleme ve uyku enerjilerini toplayarak Mica2 için harcanan toplam enerji bulunur [8,14]. Bu değerler beklenen enerji tüketiminin CPU tarafından hesaplanması ve radyo yardımıyla bulunurlar. Akım tüketiminin bir özeti Tablo 3.4'de verilmiştir.

Tablo 3.4: Mica2 modelinin akım tüketimi.

İşlem	Zaman (s)		I (mA)	
Radyonun açılması (b)	350 E-6	trinit	6	Crinit
Radyoyu açma (c)	1.5E-3	tron	1	Cron
RX/TX (d) geçişi	250E-6	trx/tx	15	crx/tx
Radyo örnekleme süresi (e)	350E-6	tsr	15	Csr
Radyo örnek değerlendirme (f)	100E-6	tev	6	Cev
1 byte alınması	416E-6	trxb	15	crxb
1 byte gönderilmesi	416E-6	ttxb	20	ctxb
Örnek algılayıcı ler	1.1	tdata	20	cdata

Tablo 3.4'de verilen değerlerden hareketle, Mica2'nin 3V bir bataryaya sahip olduğunu [14] varsayılırsa, 1-bit için iletim ve alım enerjisi (3.5)'deki gibi hesaplanabilir.

$$\text{Enerji} = \text{Akım} \times \text{Voltaj} \times \text{Süre} \quad (3.5)$$

$$\text{Energy}_{\text{Tx}} = 20 * 10^{-3} \text{ A} * 3 \text{ Volts} * 416 * 10^{-6} \text{ sec} / 8 \text{ bits} = 3.12 \mu\text{J/bit}$$

$$\text{Energy}_{\text{Rx}} = 15 * 10^{-3} \text{ A} * 3 \text{ Volts} * 416 * 10^{-6} \text{ sec} / 8 \text{ bits} = 2.34 \mu\text{J/bit}$$

Burada akım amper, gerilim volt ve zaman saniye cinsinden belirtilmiştir.

3.1.2.4. Mica2 Özel Modeli Ve Gerçek Ölçümler

Shnayder ve arkadaşları Mica2 platformu üzerindeki ölçümlerden yola çıkarak bir akım tüketimi modeli (Tablo 3.5) sunmuşlardır [12].

Tablo 3.5: Gerçek ölçümler ile akım tüketimi.

Mod	Akım	Mod	Akım
(CPU)		(RADYO)	
Aktif	8.0 mA	Rx	7.0 mA
Boşta	3.2 mA	Tx (-20 dBm)	3.7 mA
ADC Gürültüsü	1.0 mA	Tx (-19 dBm)	5.2 mA
Düşürme			
Kapama	103 μ A	Tx (-15 dBm)	5.4 mA
Güç tasarrufu	110 μ A	Tx (-dBm)	6.5 mA
Bekleme	216 μ A	Tx (-dBm)	7.1 mA
Uzatılmış bekleme	223 μ A	Tx (dBm)	8.5 mA
Dahili Osilatörü	0.93 mA	Tx (dBm)	11.6 mA
LEDler	2.2 mA	Tx (dBm)	13.8 mA
Algılayıcı kartı	0.7 mA	Tx (dBm)	17.4 mA
EEPROM erişimi		Tx (+10 dBm)	21.5 mA
Okuma	6.2 mA		
Okuma Süresi	565 μ s		
Yazma	18.4 mA		
Yazma Süresi	12.9 ms		

Tablo 3.5'deki her bir aktif bileşen için enerji tüketimlerinin toplamı toplam akımı verir. Örneğin, sadece CPU aktif durumda iken ve +10dBm (en kötü senaryo) üzerinden yayın yapan bir radyonun 1-bit gönderebilmek ve alabilmek için tükettiği enerji hesaplanmak istenilirse, iletilen her bir bitin enerji maliyeti (3.6)'daki gibi hesaplanabilir:

$$\text{Energy}_{Tx} = (8+21.5) * 10^{-3} \text{ A} * 3 \text{ Volts} * 416 * 10^{-6} \text{ sec} / 8 \text{ bits} = 4.602 \mu\text{J/bit}$$

$$\text{Energy}_{Rx} = (8+7) * 10^{-3} \text{ A} * 3 \text{ Volts} * 416 * 10^{-6} \text{ sec} / 8 \text{ bits} = 2.34 \mu\text{J/bit}$$

(3.6)

Bu model için elde edilen enerji maliyet değerleri [15]'te önerilen model ile hesaplanan enerji maliyetlerine benzerlik göstermektedir. Bu farkın muhtemel sebebi, Polastre ve arkadaşlarının iletim sırasında ne kadar enerji kullanıldığını belirtmemiş olmaları ve bunun muhtemelen [8,12]'de kullanılan farklı olmasıdır.

3.1.2.5. Karşılaştırma Sonucu

Bu çalışmada incelenen enerji tüketim modellerini dikkate değer kılan, her ne kadar algılayıcıların enerji tüketimi değişik modellere dayalı olsa da değişik enerji tüketim şekilleridir. En son teknolojik ve elektronik gelişmeler enerji tüketimini hesaplamayı kolaylaştırmış, dolayısıyla her bir algılayıcı kalan enerji yüzdesini yazılım kullanarak rahatlıkla verebilmektedir. Tablo 3.6'da gösterildiği gibi klasik enerji modeli düşük operasyonel güce ihtiyaç duymakta, veri alma ve vermede daha az enerji kullanmaktadır. Ancak klasik enerji modelinin bekleme modunda hiç enerji kullanmadığı görülmektedir, bu diğer modellerde söz konusu bir durum değildir. Diğer taraftan, klasik enerji modelinde enerji joule cinsinden ölçülürken μAMPS ve Mica2 modellerinde sırasıyla voltaj ve watt kullanılmaktadır. Tablo 3.5 operasyonel gücün artması ile enerji maliyetinin arttığını; ve daha çok işlemin daha çok enerjiye ihtiyaç duyduğunu göstermektedir. Örneğin, gürültü kontrol işlemi daha az işlem yapmakta ve daha az enerji tüketmektedir.

Tablo 3.6: Enerji tüketimi modelleri.

Modeller	Enerji Modeli	İletim Maliyeti	Alım Maliyeti	Bekleme modu enerji tüketimi	Seçenek ve işlem
Klasik	joule	50 nJ/bit	50 nJ/bit	0	Düşük
μAMPS	Watt	1.04 $\mu\text{J/bit}$	0.4 $\mu\text{J/bit}$	10 mW	Orta
Mica2	Volt	3.12 $\mu\text{J/bit}$	2.34 $\mu\text{J/bit}$	0	Yüksek
Mica2 gerçek ile	Volt	4.602 $\mu\text{J/bit}$	2.34 $\mu\text{J/bit}$	216 μA	Çok yüksek

3.1.3. Ağ Dayanıklılığı

Algılayıcı ağının az çekmesi bu ağların dezavantajlarından bir tanesidir, özellikle sınırlı güç kaynağı olması durumunda gerçekleşir. Ayrıca, bir düğümün özel bir pozisyonu ağın dayanıklılığı konusunda risk teşkil ediyor olabilir. Örneğin havuza (Sink) yakın bir yerde bulunan bir düğüm, işlem yükü daha ağır olduğu ve daha hızlı enerji harcadığı için daha çabuk işlem dışı kalabilir. Böyle bir düğümün devre dışı kalması havuz ile iletişimi keseceği için bütün ağın devre dışı kalmasına sebep olabilir. Bazı enerji yönetimi çözümleri yapı ile bağımsızdır. Burada belirtilen durumda, otomatik yapı etkindir; karar vermenin büyük bir kısmı bölgesel olarak yapılmaktadır, bu sayede kritik düğümlerdeki trafik yükü rahatlar, düğümlerin ve ağın dayanıklılığı artar. Enerjinin erken bitmesi, daha az trafik de olan ve dağılmış durumdaki düğümlerde de sık rastlanan bir durumdur. Böyle durumlarda, her bir düğümde güç yönetiminin yapılması, kritik düğümlerin daha az işlem taşınması ve enerji duyarlı çözümler sunulması yardımcı olabilir. Bir ağın dayanıklılığı değişik şekillerde de değerlendirilebilir, yani başka hesaplama yöntemleri de vardır.

- İlk algılayıcı devre dışı kalması:

İlk defa bir algılayıcı devre dışı kaldığı zaman gerçekleşir; ilk algılayıcı devre dışı ise ağ da devre dışı kalır.

- Algılayıcının kalan enerjisi %10'un altına inmesi:

Bu metod diğerlerine göre daha uygulanabilir, algılayıcı %10 enerjisini acil durumlar için tutar, bu yedek enerji sadece acil durumlarda kullanılır.

- Son algılayıcının devre dışı kalması ile ağın devre dışı kalması:

Bu metod çok mantıklı değildir ve nadiren kullanılır.

Ağ işlevsel değilken, yani havuza (Sink) yakın bir düğüm devre dışı kalmışken havuz ağ ile iletişime geçemez ve ağ görevini yerine getiremez.

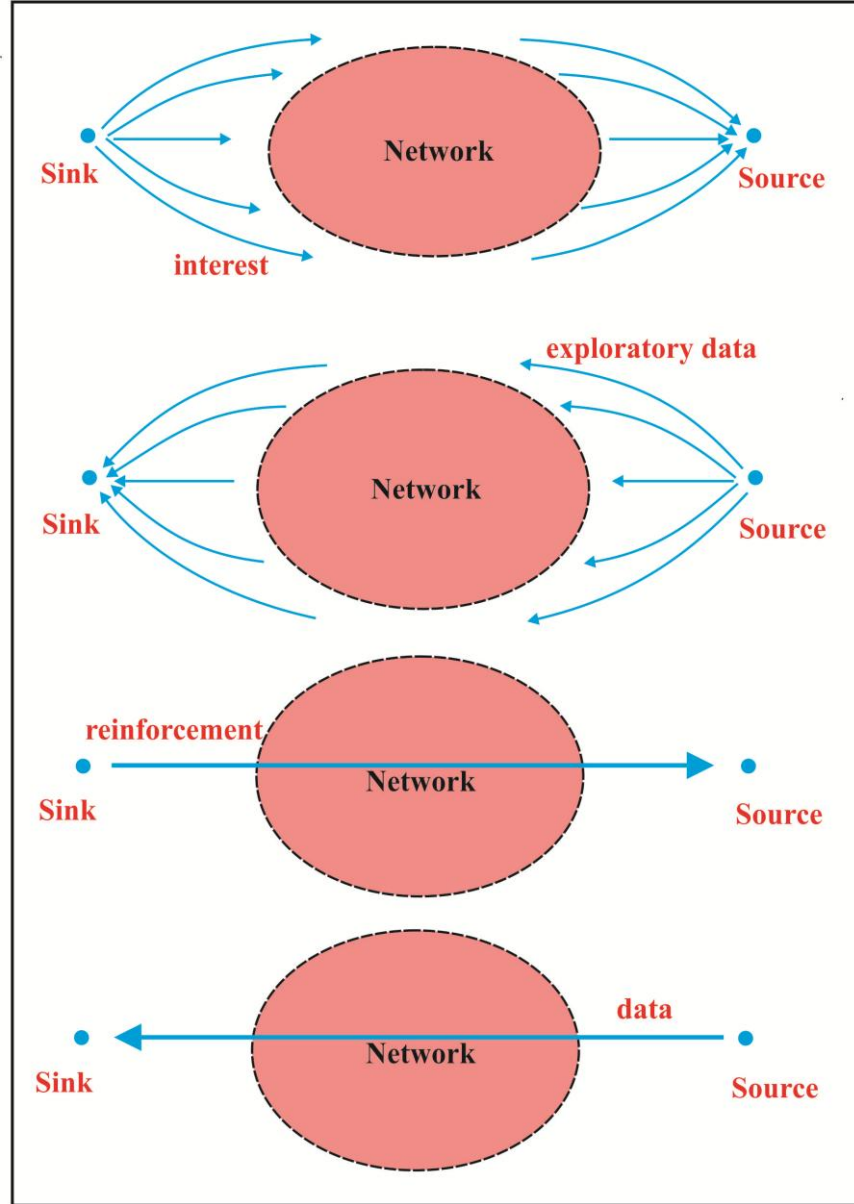
Dayanıklılığı arttırmak için yapılan en yaygın şeylerden birisi ağdaki düğümler arasında enerji tüketimini dengelemektir. Adil enerji tüketimi, daha önce belirtildiği gibi, genetik algoritma, bulanık mantık algoritması ve yapay sinir ağları aracılığıyla uygulanır.

3.1.4 Kaynak Sayısı

Kaynak seçimini etkileyen parametrelerden birisi de kaynak sayısıdır. Bu terim bir ağdaki kaynakları veya aynı anda meydana gelen olayları belirtmekte kullanılır. Örneğin, askeri bir alan düşman tarafından birkaç kere (aynı anda) istila edilmektedir ve bu durumu belirten düğümler birbirine bağlıdır. Başka durumlarda, bir düğüm birkaç olayı izleyebilir ve hepsini havuza raporlayabilir. Bu özelliğin avantajları ve zayıflıkları vardır. Bir düğümün birden fazla olayı aynı anda gözlemleyebilmesi diğer düğümlere beklemede olma ve daha çok enerji tasarrufu yapma şansı sağlamaktadır. Diğer taraftan raporlamanın bütün yükünü alan bir düğüm enerjisini daha çabuk harcayacaktır ve bu durum da anahtar düğümler daha erken devre dışı kalabilecektir. Bu iki durum arasındaki dengeyi sağlamak için çeşitli mekanizmalar ve yönetim araçları vardır.

3.1.4.1. İki Fazlı Yöntem

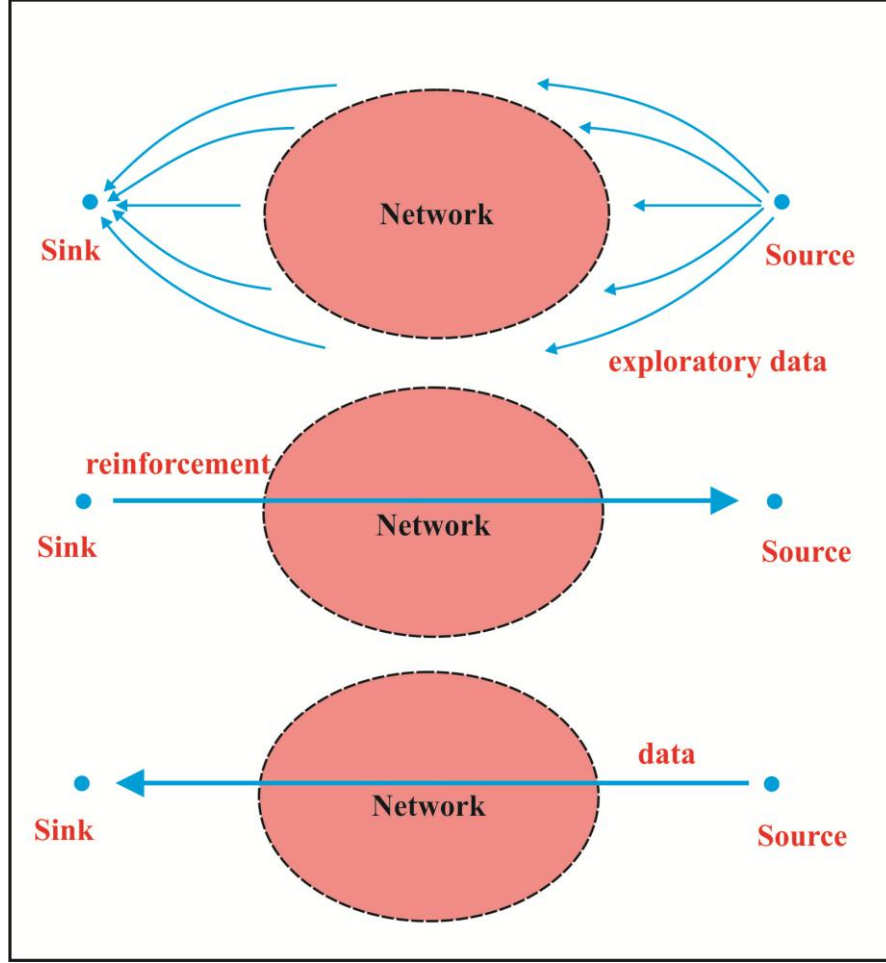
İki fazlı çekme protokolü (Şekil 3.5) havuz ve kaynaklar arasında, ilgi ve meyil paketleri ile rota kurulumlarını sağlar ve yüksek kalite rota amplifikasyon mekanizması ile tanınır. İlk aşamada, verici/hedef düğüm bir ilgi mesajını ağda iletir, ne zaman bir düğüm toplanan veri ile ilgili bir uyum bulursa, bir keşif mesajı kaynağa gönderilir. Oluşmuş olan meyillerden, veriler en kısa iletim süresi ile en kaliteli kanalı bulmayı dener. Kaynağa ulaşıldıktan sonra, kaynak düğüm toplanan veriyi kaynağa iletir.



Şekil 3.5: İki faz çekme yayma metodunun fonksiyonu.

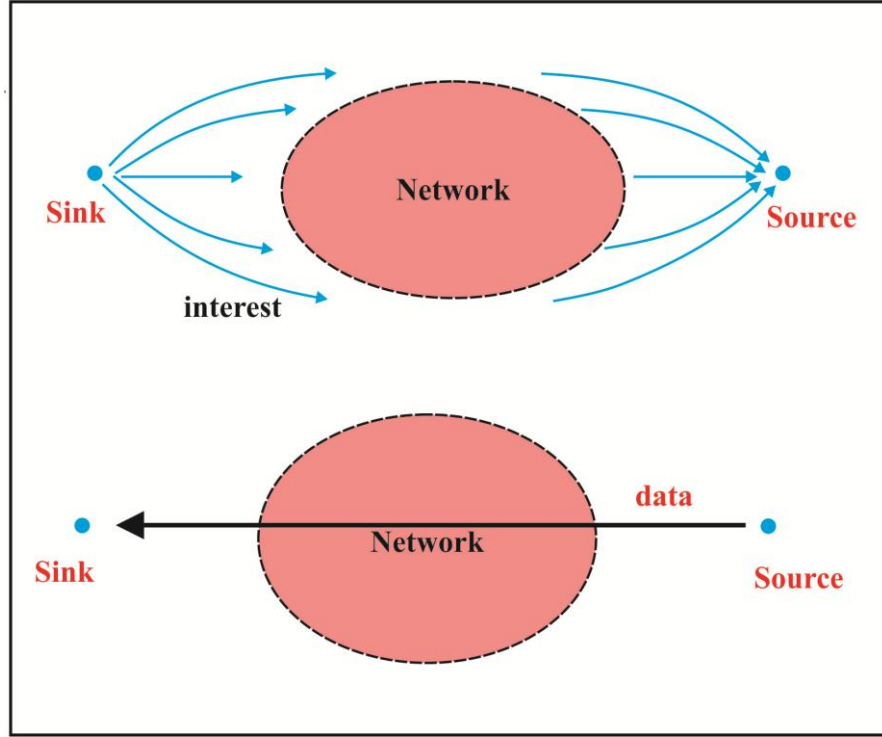
3.1.4.2. Çekme Diffüzyonu Yöntem

Çekme fazı çekme metodu çoğu konuda en iyi seçimdir, ancak bazı kullanımlarda düşük performans gösterir, özellikle çok kaynak ve alıcı varken ve alıcıların özel istekleri varken spesifik bağlantı veri trafiği hacmini ciddi bir şekilde etkiler. Bu problemle başa çıkmak için, çekme diffüzyonu yöntemi geliştirilmiş, bilgi yayıcı ve kullanıcı rolleri yer değiştirilmiştir. Bu sayede veri kaynakları etkin olarak kullanıcıları ararlar.



Şekil 3.6: Çıkış yayılım yöntemi.

Metod, paketleri ağ üzerinde tek mod halinde (keşif verisi) yollarken, iki-fazlı çekme metodunda paketler iki mod halinde yollanırlar (Şek. 3.6). İteleme diffüzyon metodu az sayıda kaynak ve az sayıda alıcı (sink) için uygundur.



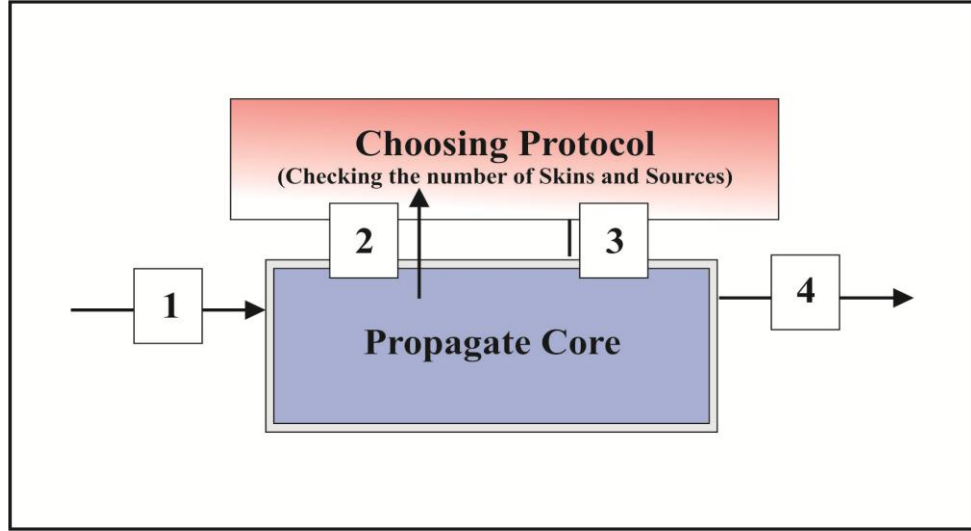
Şekil 3.7: Tek fazlı çekme metodunun fonksiyonu.

3.1.4.3 Tek Fazlı Yöntem

Şekil 3.7'de görüldüğü gibi, iki faz çekme metodununun fazlarından birisi talep üzerine çıkartılmıştır. İki-faz çekme yöntemindeki gibi, talep edenler ilgi mesajlarını ağ üzerinden gönderirler; ancak iki-faz metodunun eksikliğinde, bir ilgi kaynağına ulaştığı zaman, keşif verisi olarak almaz. Bunun yerine, önceden oluşturulmuş olan meyil üzerinden asıl veriyi hedefine gönderir. Bu prosedür kullanıldığı zaman mesajların amplifikasyona ihtiyacı kalmaz. Bu metod çok sayıda kaynak ve az alıcı için uygundur.

3.1.4.4 Kaynak ve Havuz Üzerindeki Filtrenin Seçilmesi

Kaynak sayısına bağlı olarak, algılayıcı veri yayılımı ağları uygun filtreleri, bir grup mekanizmayı ve yönetim metodlarını kullanır. Ağlar uygun yayılım filtresini operasyona göre seçerler.



Şekil 3.8: Havuz ve kaynak sayısına göre doğru yayılım metodunun seçilmesi.

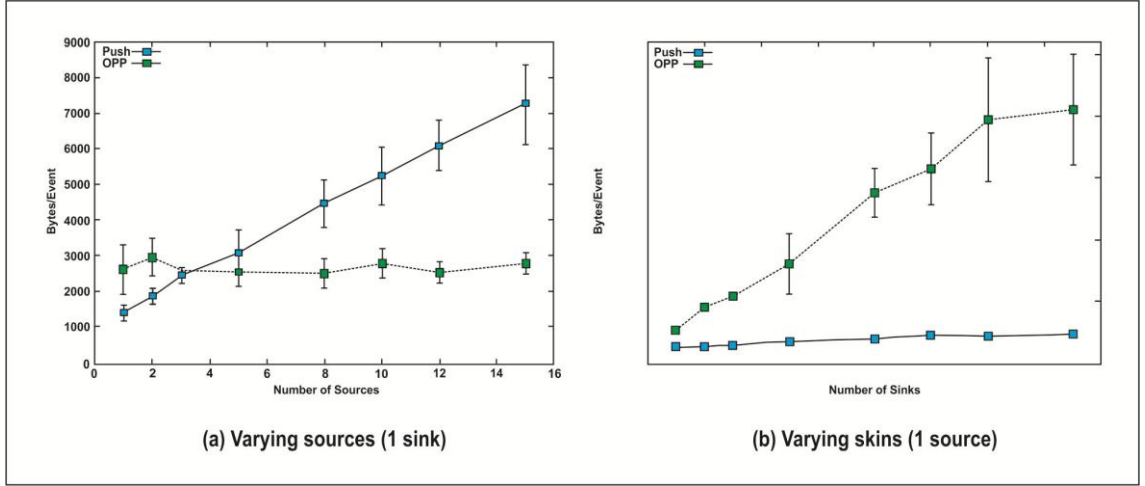
Şekil 3.8'da gösterildiği üzere, ağ üzerinde bir olay olduktan sonra, nakledici çekirdek toplam havuz ve kaynak sayısını kontrol eder ve en uygun filtreyi uygulayarak ilgi pakedini ağa gönderir.

3.1.4.5. Kaynak Sayısına Göre Yöntemlerin Karşılaştırılması

Tabloda 3.7'de listelendiği gibi, kaynak sayısı yüksek iken, tek fazlı çekme metodu iki-fazlı çekme metodundan daha iyi performans gösterir. İtme diffüzyon metodu olayları raporlamada diğer metodlar kadar iyi değildir; ancak toplam havuz sayısı çok yüksekken en iyi yöntemdir. Ayrıca tek fazlı çekme metodu ana istasyona olayları uygun bir şekilde raporlayamaz.

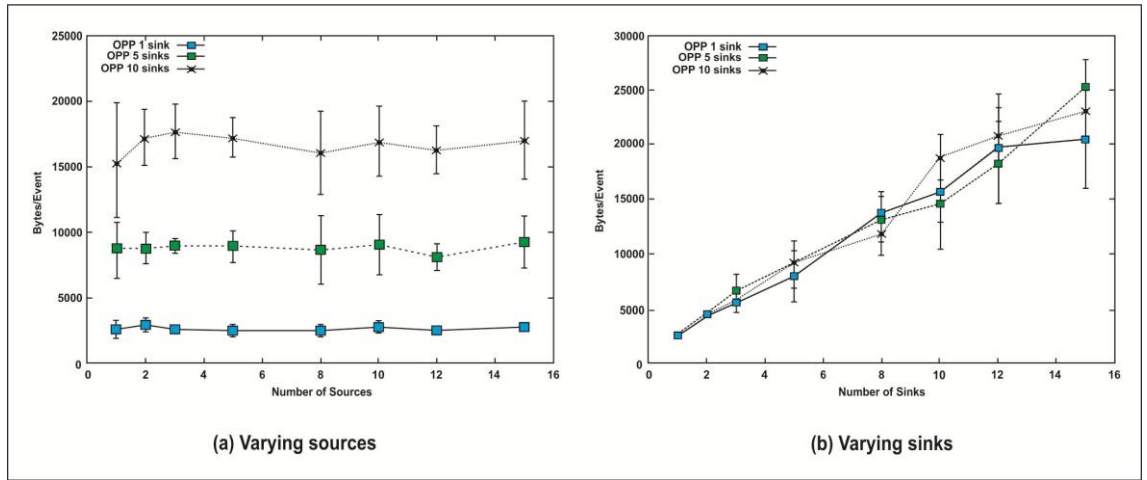
Tablo 3.7: Kaynak ve havuz sayısına göre yayılım yöntemleri.

	Kaynak sayısı	Alıcı sayısı (havuzlar)
Tek fazlı çekme	Yüksek	Düşük
İki-fazlı çekme	1	1
Çekme diffüzyonu	Düşük	Yüksek



Şekil 3.9: İteleme diffüzyonu ve tek fazlı çekme metodu diyagramları.

Şekil 3.9a ve 3.9b üzerinde gösterildiği gibi, çekme diffüzyon metodu ile kıyaslandığı zaman, tek fazlı çekme metodu çok sayıda havuz söz konusuysen daha iyi bir performansa veya daha iyi bir raporlamaya sahip değildir. Şekil 3.10b toplam kaynak sayısı çekme diffüzyon metodunun neredeyse doğrusal olduğunu ve tek-fazlı çekme metoduna göre düşük performansa sahip olduğunu ortaya koymaktadır. İlkinin raporlama performansı ikincisine göre kaynak sayısı arttıkça daha yüksek bir şekilde artmaktadır. Bununla birlikte, şekil 3.10'de belirtildiği gibi, kaynak sayısı arttıkça tek fazlı çekme metodu yüksek sayıda olay gerçekleşmesi durumunda raporlamada sorunlar yaşamaktadır.



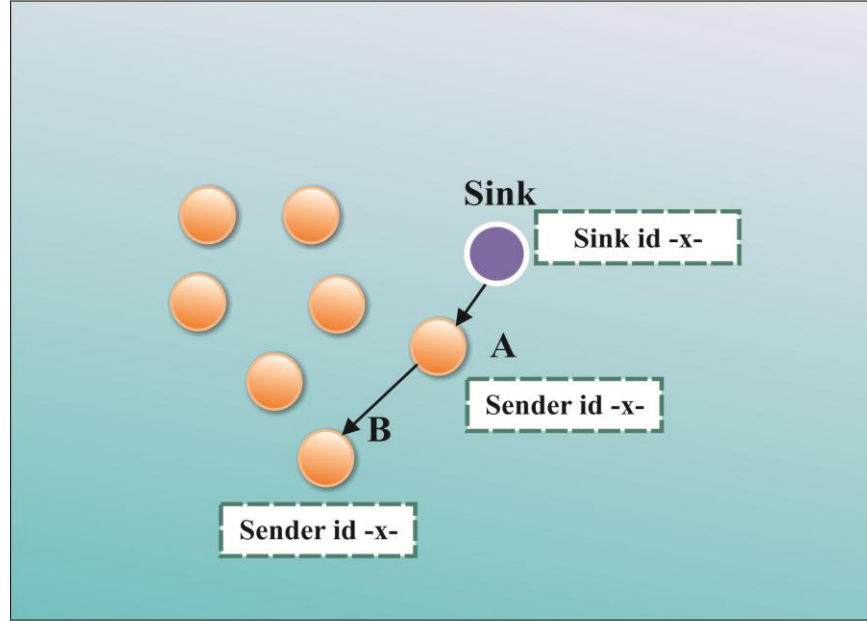
Şekil 3.10: Değişik sayıda kaynak ve havuz ile tek-faz çekme metodu.

3.1.5. Atlama Sayısı

Atlama sayısı, toplam geçilmiş olan atlama adedini göstermektedir. Herbir düğüm başlıktaki bilgileri değiştirerek paketin özelliklerini günceller ve böylelikle komşulara havuzdan veya kaynaktan itibaren toplam geçilen atlama sayısı iletilir. Tekdüze yerleşim durumunda, atlama sayısı parametresi uzaklığın ölçülmesi için kullanılır. Bu parametrenin temel kullanımı kaynak ile havuz arasındaki rotanın belirlenmesidir.

3.1.5.1. Yöntemler

Genel olarak, kaynak düğüm en az enerjiye ihtiyaç duyacak adımı seçer. Burada önemli bir nokta, en az geçiş ile gidilecek olan rotanın hem avantajlarının hem de dezavantajlarının olmasıdır. Avantajlardan birisi, paket hedefine daha kolay ulaşır. Bu daha az uçtan-uca gecikme anlamına gelir, bu zamanlamaya önem veren uygulamalar için değerlidir. Başka bir avantaj da ,daha az düğümün iletimde ve dolayısıyla daha çok düğümün bekleme durumunda kalması, daha çok enerjinin tasarruf edilmesidir. Dezavantajlı bir yanı, daha çok düğümün katılacağı bir rota seçilirse düğümler daha uzun süre işlevsel kalabilir, daha iyi bir enerji dengesi sağlanabilir. Böylelikle, bir düğümün çok yüklenmesi yerine daha çok düğüm işlevsel olabilir. Diğer taraftan, daha çok düğüm içeren rotanın uygulanması daha çok trafik yüküne ve daha uzun gecikmelere sebep olur. Bu durum daha iyi iletim oranı, enerji dengesi ve paket kaybında düşüş gibi avantajlar sağlasa da servis kalitesini ciddi bir şekilde düşürür. Son olarak, tüm bu avantaj ve dezavantajlı yönler uygulamaya ve ağ tipi yönetimine göre değerlendirilebilir. Şekil 3.11'de gösterildiği gibi, atlama sayısı paketin başlığında kaydedilir ve havuza ulaşmaya kadar her bir düğümde güncellenir; böylece havuz toplam atlama sayısını bilebilir.



Şekil 3.11: İlgili pakedinin yayılması ve atlama sayısının güncellenmesi.

3.1.5.2. Atlama Sayısı açısından Yöntemlerin Karşılaştırılması

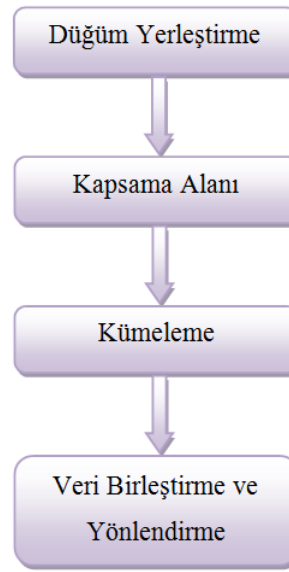
Bir kaynak, Tablo 3.8'de belirtilen parametrelere göre en uygun düğüm rotasını seçer. Genellikle, kaynak düğümü minimum enerji ihtiyacı olacak bir rotayı seçmeye çalışır, ancak avantajlı ve dezavantajlı yönler göz önünde bulundurulmalıdır. Daha kısa bir rota, daha az atlama içerir ve mesajın daha hızlı yollanmasını sağlar. Daha hızlı iletim, daha kısa uçtan-uca gecikme anlamına gelir, bu da hızın önemli olduğu uygulamalar için çok değerlidir. Kısa bir yol seçmenin başka bir avantajı ise, rota üzerinde daha az düğümün yer alması ve daha çok düğümün beklemede kalmasıdır.

Tablo 3.8: Atlama sayısına göre servis kalitesi parametreleri.

	Enerji	Gecikme	İletim	Trafik	Dayanıklılık
Atlama sayısında artış	Enerji dengesinin dengelenmesi	Yüksek yayılım sayısı nedeniyle artış	Düğümlere daha iyi erişim sebebiyle artış ve paket kaybında düşüş	Daha çok iletim nedeniyle trafikte artış	Tek düğüme aşırı yüklenmenin engellenmesi sebebiyle daha yüksek dayanıklılık
Atlama sayısında azalma	Bir düğümün çok kullanılması	Daha az iletim nedeniyle trafikte azalma	Uzun menzilli iletme bağlanma, paket kaybında artış/azalış	Trafik yükünün artmasının engellenmesi	Pil bitmesi durumlarında azalma ve dayanıklılığın artırılması

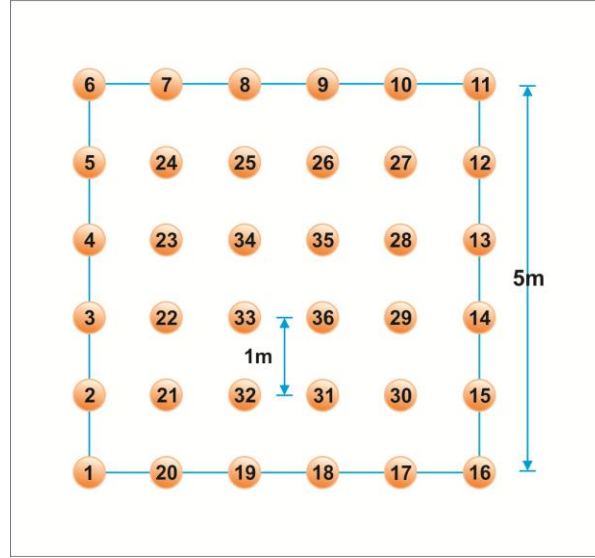
3.2 KABLOSUZ ALGILAYICI AĞI'NİN ÇALIŞMA AŞAMALARI

Kablosuz Algılayıcı Ağlarda, destek sürekliliği, algılayıcılar arasında sürekli iletişim ve veri bütünlüğünün sağlanması en önemli parametreler haline gelmiştir. Bu bölümde, KAA'ların farklı işlem aşamaları incelenmektedir. Genel anlamda, klasik KAA'ların işlem aşamaları Düğüm Yerleştirme (Node Placement), Kapsama Alanı (Network Coverage), Kümeleme (Clustering), Veri Birleştirme ve Yönlendirme (Data Aggregation and Routing) olarak sınıflandırılabilir (Şekil 3.12).



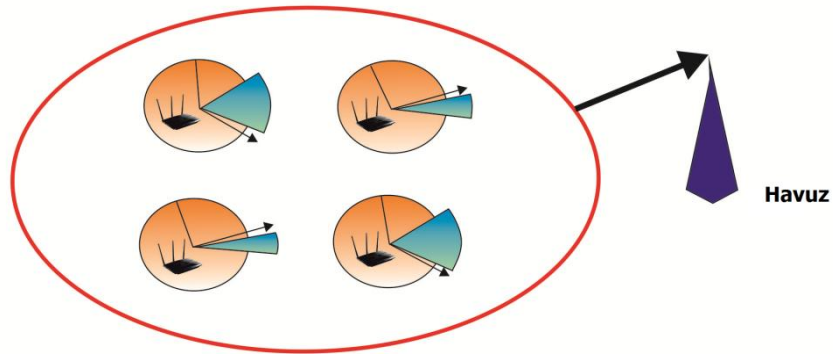
Şekil 3.12: KAA'lerin temel çalışma aşamaları.

KAA kurulumu öncesinde gerçekleştirilen “düğüm yerleştirme,” son derece önemli bir aşamadır. KAA'lar için genel olarak bilinen çok sayıda devre dağılım modeli mevcuttur; bunlardan bazıları düzenli, rastgele ve izgara dağılım modelleridir. Izgara yerleşim modelinde algılayıcılar arasındaki mesafe tahmin edilebilir. Şekil 3.13'deki izgara yerleşim örneğinde de görüldüğü gibi, algılayıcılar arasındaki boşluklar sabittir.



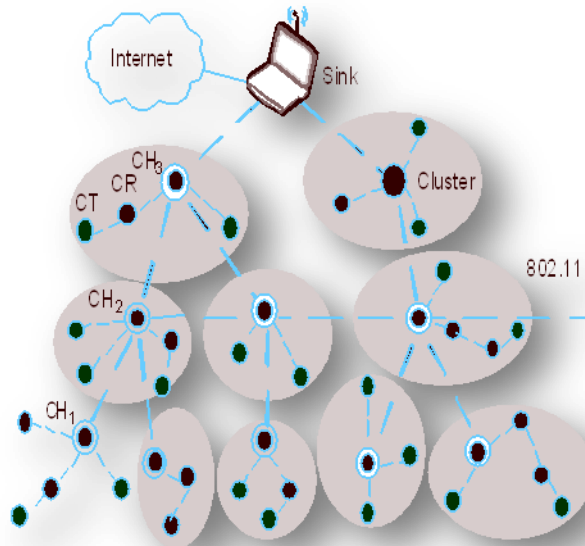
Şekil 3.13: KAA için kılavuzlu yerleşim.

Yerleşim şekli ile ilgili karar, istenen uygulamaya bağlı olarak seçilir (bkz. bölüm 3.1.1.4). Poisson dağılımı, iyi bir tahmin dağılımı sağlamasına ek olarak, zaman ya da mekânda rastgele gerçekleşen olaylar için de kullanılabilir önemli bir olasılık modelidir. Bu dağılım, aynı zamanda kablosuz ağlar üzerinde yapılan detaylı çalışmalar ve bunların simülasyonu için de yaygın olarak kullanılır. Bir sonraki aşama ise servis çeşitliliği göz önüne alınarak algılayıcıların bağlanmasıdır. Daha önce de belirtildiği gibi, algılayıcıların farklı özellikleri arasında radyo menzili ve hizmet alanı kilit faktörlerdir. Gerekli kapsama alanı dikkate alınarak, hizmetlerin en iyi şekil ve miktarda sağlanabilmesi için en iyi yerleşim modelinin kullanılması son derece önemlidir. Yanlış yerleşim modellerinin kullanımı ise enerji ve finansal kaynak israfına neden olacaktır. (Şekil 3.14)



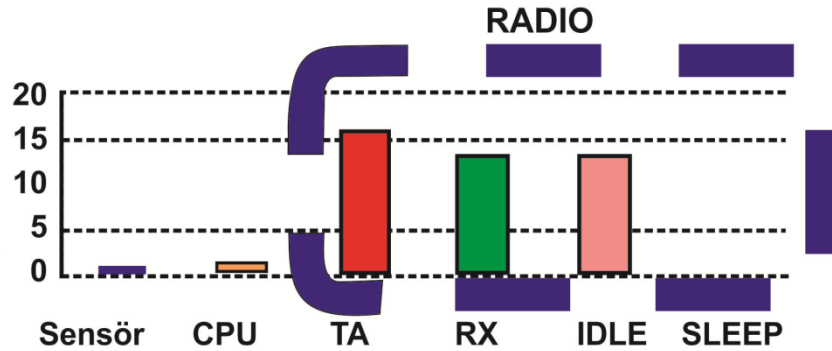
Şekil 3.14: Kapsama alanı aşaması.

KAA optimizasyonunda anahtar rol oynayan başka bir ana işlem ise kümelemedir. Kümeleme ile algılayıcı ağlar, bölme küme (division cluster) olarak da bilinen gruplara ayrılır. Her bir kümenin, küme içerisindeki düğümlerden gelen verileri toplayan ve daha sonra da bu verileri doğrudan ya da diğer kümelere ait başlıkları aşama aşama kullanarak alıcıya gönderen küme başları (Cluster Head) bulunmaktadır. Bu nedenle, verilerin alıcıya doğrudan iletiildiği durumlara kıyasla, devrelerin iletişim istikametinde azalma görülebilir. Yani, algılayıcı ağlar arasındaki yükün azaltılması açısından kümeleme işlemi etkili bir yaklaşım olacaktır. Şekil 3.15'te KAA'larda kümelenmiş devreler görülmektedir. Kümeleme işlemi ile daha fazla enerji tasarrufu sağlanabilir ve KAA'lerdeki algılayıcıların kullanım ömrü arttırılabilir. Bu tekniğin diğer avantajlarından bazıları ise yüksek güvenlik, daha fazla veri aktarımı ve ağı gelişmiş ölçeklenebilirliğidir. Daha iyi performans için, uygulamaya bağlı olarak farklı protokollerin kullanılması da mümkündür. Küme başları daha fazla enerji tüketimine sebep olsa da protokoller ile ağların kullanım ömrü güvenli bir şekilde arttırılabilir. Bu nedenle, bir ağı yaşam süresini uzatmak için, küme başlarının her bir küme yerleşimi periyodunda tekrar seçilmesi gerekir. Bu protokoller, etkili bir kümeleme algoritması kullanımlarını sağlasa da, en iyi düğümün küme başı olabilmesi opsiyonunu ortadan kaldırır. Ancak, optimizasyon ile algoritmaların enerji tüketimini önemli ölçüde azaltması ve ağların etkinlik ve kullanım sürelerinin arttırılması mümkündür.



Şekil 3.15: KAA'ların kümelenmesi.

Sonuç olarak, ana istasyonlar arasındaki veri transferi ve sorgular, bilgi alış noktaları ya da eylemler de KAA'larla ilgili önemli durumlardan bazılarıdır. Verinin devre ile baz istasyonu arasında doğrudan transferi basit bir veri transferi işlemidir. Tek adımlı, yönelimli işlem oldukça masraflıdır; devre ile baz istasyonu arasındaki mesafe arttıkça daha fazla enerjiye ihtiyaç duyulur ve bunun sonucu olarak da ağıın kullanım ömrü kısılır. Başka bir veri transferi işlemi ise belirli bir etki alanı için kullanılan çok basamaklı, yönelimli transferdir. Bu işlem önemli ölçüde enerji tasarrufu sağlar ve ağ içerisindeki çakışmaları büyük oranda azaltır; ancak bu işlemin de, özellikle yönlendirme mekanizmalarına bağlı bir takım kısıtlamaları mevcuttur. Araştırmacıların veri toplama ve yönlendirme aşamalarına olan ilgisinin temel sebebi de bu aşamada tüketilen çok miktarda enerjidir. KAA'nin her aşaması için gerekli olan enerji Şekil 3.16'da görülmektedir. Burada da açıkça görüldüğü üzere, en çok enerji tüketimi telsiz iletişim esnasında gerçekleşir. Buna bağlı olarak, bu aşamaya ilişkin yapılan detaylı çalışmalar sayesinde KAA'nin enerji tüketimi ve ağıın kullanım ömrü optimize edilebilir [31,32,33].

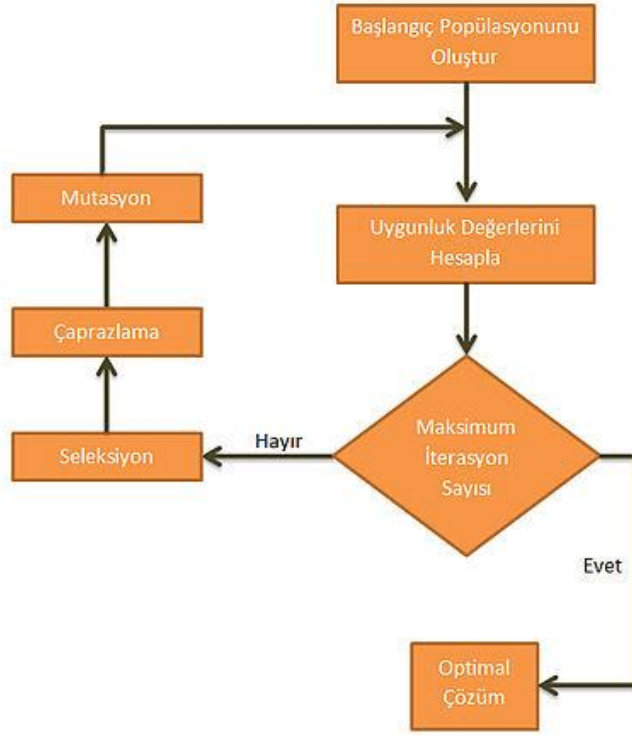


Şekil 3.16: KAA'lerin farklı aşamalarında görülen enerji tüketimi.

Kablosuz Algılayıcı Ağları optimize etmek için kullanılan bulanık mantık (fuzzy logic), yapay sinir ağları ve evrimsel algoritmalar gibi çeşitli yöntemler mevcuttur. Ancak, en yaygın yöntem, büyük ölçekli ağlarda daha iyi sonuçlar vermesi ve son aşamada nihai bir formül oluşturması bakımından insan kullanıcıları için çok daha kullanışlı ve faydalı bir yöntem olan genetik algoritmadır. Bu nedenle de, bu bölümde ağların optimizasyonu ve gereksinimlere uyarlanması için genetik algoritma kullanılmıştır [34].

3.3. GENETİK ALGORİTMA

Evrensel sezgisel algoritma (Universal Heuristic Algorithm) olarak da bilinen genetik algoritma, farklı bireyler yaratarak optimal bir çözüm tahmininde bulunur [35]. Algoritma temelde, bir uygunluk (fitness) fonksiyonu oluşturmaya ve bu fonksiyonun optimizasyonuna dayanır. Bir sonraki bölümde, genetik algoritmanın temel bölümleri verilmiştir (Şekil 3.17).



Şekil 3.17: GA mekanizmasının genel şeması.

3.3.1. Başlatma

Genetik algoritma, 0 veya 1 değerlerinden oluşan bir gen dizilimine sahip rastgele kromozomlar içeren bir başlangıç popülasyonu ile başlar. Daha sonra, çaprazlama (crossover) ve seçme operatörlerini de içeren ve tekrarlayan prosesler ile algoritma bireyleri optimum bir çözüm sağlamaya yönlendirir. Yeni bir popülasyon geliştirmenin kalıcı GA ve aşamalı GA olmak üzere iki yolu vardır. Birinci yol tercih edilmesi halinde, popülasyon içerisindeki bir ya da iki üyenin yerine yenileri konulurken, ikinci yolda ise belirli bir aşamaya ait tüm bireylerin yerine yenileri getirilir.

3.3.2. Uygunluk

Genetik algoritma altında, uygunluk fonksiyonu tanım olarak her bir kromozomun niteliklerine göre puanlanması süreci anlamına gelir. Verilen puanlar, daha fazla üremenin devamlılığını sağlayan birer unsurdur. Uygunluk fonksiyonunun probleme bağlı oluşu göz önünde bulundurulması gereken bir durumdur çünkü bazı problemler söz konusu olduğunda problemin tanımlanması mümkün olmayabilir. Doğal olarak, bireylerin uygunluk puanlarına bağlı yeni bir nesile geçiş yapmasına izin verilir. Bu durumda ise bireylerin kaderini belirleyen unsur bu puanlar olacaktır.

3.3.3. Seçme

Birbirini takip eden nesiller esnasında, mevcut nesile ait üyeler çiftleşmek üzere uygunlukları baz alınarak seçilir ve yeni bir nesil oluşturulur. Uygunluk puanı daha yüksek olan bireylerin seçilme şansı daha yüksektir; bu süreç de en iyi çözümün ayrıcalıklı olarak seçilebilmesi ile sonuçlanır. Popülasyon çeşitliliği sağlanabilmesi bakımından, fonksiyonların çoğu uygunluğu yüksek olmayan bireylerden daha az sayıda barındırabilmesi için stokastik olarak tasarlanmış bir öğeye sahiptir[28]. Kullanılan pek çok seçme yöntemi arasından Roulette-Wheel, Denklem 3.7 de verilen olasılık formülüne göre uygun bireyleri ayırt etmek amacıyla kullanılır.

$$P_i = \frac{F_i}{\sum_{j=1}^n F_j} \quad (3.7)$$

Bu denklemde F_i uygunluk kromozomunu ve ' n ' ise popülasyon büyüklüğünü temsil etmektedir. Roulette-Wheel'e göre, bireylerin her birine 0 ile 1 arasında bir değer verilir.

3.3.4. Çaprazlama (Crossover)

Çaprazlama ya da çoğaltma süreci üretime doğru atılan temel adımdır. Aslına bakılırsa, bu süreç, kalıtsal özelliklerin bir nesilden diğerine aktarıldığı cinsel üreme sürecinin bir simülasyonu niteliğindedir. Üreme esnasında gerçekleşen çaprazlama sürecinde, ıslah (breeding) seçimi süreci ile ebeveyn olacak bireylerden oluşan bir çift kullanılır. Yeni popülasyon istenen büyüklüğe ulaşana dek bu süreç devam eder. Genel olarak, her

birinin amacı diğerinden farklı olan çok sayıda çaprazlama işlemi gerçekleştirilir. Bunun en kolay yolu ise tek-nokta yöntemidir; bu yöntemde ebeveyn rollerinin bölüştürülmesi için rastgele bir nokta seçilir.

3.3.5. Kablosuz Algılayıcı Ağlarında Uygunluk Parametreleri

Bir kromozomun uygunluğu, enerji tüketiminin ne ölçüde minimize edilebileceğini ve kapsama alanının ne ölçüde maksimize edilebileceğini belirler. Aşağıda yer alan bölümde, KAA'larda kullanılan bir takım önemli parametreler açıklanmaktadır.

1. Baz İstasyonuna Direk Uzaklık (DDBS): Tüm algılayıcı devreleri ile d_i şeklinde belirtilen (Denklem 3.8) BS (Baz İstasyonu) arasındaki uzaklıkların toplamını belirtir.

$$DDBS = \sum_{i=1}^m d_i \quad (3.8)$$

Yukarıdaki denklemde kullanılan ' m ' düğüm (devre) sayısını belirtir. Açıkça görüldüğü gibi, enerji tüketimi ile devre sayısı arasında önemli bir ilişki vardır ve büyük KAA'lar için kullanılan enerji devre sayısı ile orantısal olarak oldukça fazladır. Buna ek olarak, DDBS kapalı devre sayısının önemli olmadığı daha küçük ağlar için uygun olacaktır.

2. Küme-tabanlı Uzaklık (CD): toplam KB ve BS uzaklıkları ve belirlenmiş üye devreler ile bunların küme başları arasındaki uzaklıkların toplamı (Denklem 3.9).

$$CD = \left(\sum_{i=1}^n \left(\sum_{j=1}^m d_{ij} \right) + D_{is} \right) \quad (3.9)$$

Bu denklemde, ' n ' küme sayısını ve ' m ' ise ilişkili üyeleri belirtir; ' d_{ij} ' bir devre ile bunun KB'si (küme başı) arasındaki uzaklığını belirtir; ve ' D_{is} ' KB ile BS arasındaki uzaklığı temsil eder. Bu çözüm, çok sayıda geniş aralıklı devrenin kullanıldığı ağlar için uygun olacaktır. Kümeler arası uzaklığın fazla olması aynı zamanda daha fazla enerji tüketimi anlamına gelir. Enerji tüketiminin minimize edilebilmesi için ise CD'nin çok büyük olmaması gerekmektedir [37]. Yoğunluk her bir küme içerisindeki devre sayısı anlamına gelir ve kümelerin yoğunluğu da bu önlem ile kontrol altına alınabilir.

3. Küme-tabanlı standart uzaklıklı CDS: ortalama küme uzaklığı yerine, standart türetme (derivation) ile kümenin uzaklıklarında görülen değişimler ölçülür. CDS, algılayıcı devrelerinin yerleşimi (rastgele ya da önceden belirlenmiş) ile ilgili bir fonksiyondur. Rastgele yerleşimde farklı boyutlara sahip kümeler mevcuttur. Böylece, küme uzaklığının verilen sapması içerisinde yer alan bir SD kabul edilebilir olacaktır. Bu durumda, küme uzaklığında görülen farklılıklar sıfır değildir ancak sapmanın bilgi dağıtımına uygun olarak kullanılması gerekir[38]. Her halükarda, devre konumlarının tek şekilli olarak dağıtıldığı, önceden belirlenmiş (deterministik) yerleşim koşulları altında sağlanan küme uzaklıklarının minimize edilmesi gerekmektedir. Genel olarak, tek şekilli küme-tabanlı uzaklıklarda görülen değişiklikler, ilgili ağın zayıf olduğuna işaret eder; ancak, devrelerin rastgele yerleştirilmesi halinde durum böyle olmayacaktır.

$$\mu = \frac{\sum_{i=1}^n d_c}{n} \quad (a) \quad SD = \sqrt{\sum_{i=1}^n (\mu - d_c)^2} \quad (b) \quad (3.10)$$

Denklem 3.10 (a) ve 3.10 (b)'de görülen ' μ ,' küme uzaklıklarının ortalamasını, SD ise küme uzaklığı sapmasını göstermektedir.

4. Transfer Enerjisi (E): toplanan tüm verilerin Baz İstasyonuna (BS'ye) transferi için ihtiyaç duyulan enerji tüketimi miktarını belirtir. ' m 'nin bir küme içerisinde bulunan bağlantılı devre sayısını temsil ettiği durumlarda, E değeri Denklem 3.11 kullanılarak bulunur:

$$E = \sum_{i=1}^n \left(\sum_{j=1}^m e_{jm} + m^* E_R + e_i \right) \quad (3.11)$$

Bu denklemde ' e_{jm} ' verilerin bir devreden ilgili KB'ye transferi için gerekli olan enerji miktarını belirtir. Böylece, i indisli toplamda yer alan ilk terim, toplanmış verilerin KB'lere transferi için gerekli olan toplam enerji tüketimini, bu toplama işlemindeki ikinci terim verilerin üyelerden toplanması için gerekli olan toplam enerjiyi, e_i ise küme başından BS'ye iletim için gerekli olan enerji miktarını temsil eder.

5. İletim Sayısı (T): Genel olarak, tüm izleme periyotlarında gerçekleşen iletim sayısı BS tarafından belirlenir. Bu belirleme işlemi ise koşullar ve ağın enerji seviyesi göz önüne alınarak yapılır. Bu nedenle, 'T' uzun süreli aşamayı temsil eder; bu aşamayı maksimize eden bir optimum çözüm ve minimize eden alt bir çözüm kabul edilebilir. En iyi çözüm ya da kromozom kalitesi de önceki GA-tabanlı çözümlerin performansını belirler.

Bir sonraki bölümde, KAA'lerin tüm ana çalışma özelliklerini (örn. devre yerleşimi, kapsama alanı, kümeleme ve veri birleştirme) geliştirmek için genetik algoritma kullanılarak hazırlanmış bir uygunluk fonksiyonu tanımlanmıştır. Başka bir ifadeyle, uygunluk fonksiyonları temel olarak enerji tüketimi ve kullanım ömrü parametrelerini geliştirmek için kullanılır. Simülasyon sonuçları da protokollerin gelişimini doğrulamaktadır.

3.4 AĞ SİMULATÖRÜ - 2

Ağ mühendisleri genelde herhangi bir ağ sistemini oluşturmadan önce yapmak istedikleri işleri, sisteminin gerçekleştirilmesi gerekenleri kontrol edebilecekleri, sisteminin performansını değerlendirebilecekleri ve ince ayar yapabilecekleri kısacası sistem ile ilgili herşeyi gerçekleştirebilecekleri yazılımları (simulator) kullanmaktadırlar.

Bu tez çalışmasında, NS2 Programı bize bir ağ ortamı oluşturmamız, oluşturulan ağ ortamının bağlantı analizlerini yapmamız ve bunu sanal ortamda görebilmemizi sağlayan bir simulator programı olarak seçilmiştir.

NS-2 UC Berkeley tarafından geliştirilmiş çeşitli IP tabanlı ağ yapılarını simüle edebilen olay tabanlı bir ağ simülatörüdür. NS-2 kullanılarak TCP, UDP gibi ağ protokollerini ve FTP, Telnet, WEB, CBR ve VBR gibi trafik kaynaklarını ve Drop Tail, RED, CBQ gibi ağ yönlendiricileri kuyruk yönetimini ve Dijkstra gibi yönlendirme algoritmalarını gerçekleştirilebilir. NS aynı zamanda LAN simülasyonları için bazı MAC katmanı protokollerini de gerçekleştirme imkanı sunar. NS-2 list, heap

ve hash tabanlı takvim şeklinde 3 tip ayrık olay dağıtımını simüle edebilen bir ayrık olay dağıtıcı simülatörüdür. Aynı zamanda NS2, yapısal olarak ağ düğümlerinin gerçekleştirimi, bağlanması, yönlendirilmelerini, bazı taşıyıcı seviye protokolleri (özellikle UDP ve TCP'nin çok farklı çeşitleri) ve bazı ağ trafiği üreticilerini sağlar. Bunun yanısıra simülatöre ekstra fonksiyonlar tanımlanarak genişletilebilmektedir. Ayrıca gerekli tcl derleyiciler, simülasyon senaryo üretici ve simülasyon topoloji üretici gibi kullanımlara imkan tanımaktadır.

3.5. MOBİL ALGILAYICI AĞLAR İÇİN GENETİK YAKLAŞIM

Gezgin Algılayıcı Ağlar ise algılama, hesaplama ve iletişim fonksiyonu barındıran düğümlerden oluşan ve hareket kabiliyetine sahip bir KAA'dır[49]. Bu hareketlilik ölçütü bilinmeyen ortamlarda bir tasarım sorunu teşkil eder. Kapsama ve kullanım ömrü sorunlarını eşzamanlı olarak çözmeye genetik algoritma (GA) yenilikçi bir yaklaşım olabilir. Ağ kullanım ömrü, ilk düğümün veya ağdaki bütün düğümlerin bir bölümünün enerjilerin tükenmesi için geçen süre olarak tanımlanabilir. Bahsi geçen GA yöntemi hem dinamik ortamda hem de dinamik ağ topolojisinde optimizasyonlar için uygulanabilir [34]. Bu yazıda, ağ kullanım ömrünü uzatan azami kapsama ve asgari enerji tüketimine sahip bir ağ tasarlamak üzere gerçek zamanlı bir GA geliştirdik.

Temel bir algılayıcı düğümü çoğu durumda 5 ana bileşenden oluşur: (a) tek enerji kaynağı olarak kabul edilen bir güç kaynağı, (b) belleğe sahip bir denetleyici, (c) algılama cihazı, (d) iletişim sistemi ve (e) mobil kablosuz algılayıcı düğümler için bir mobil platform. Bu mimari parçalarının hepsi, özellikle de alan kapsama ve mobil algılayıcı düğümlerin yayılması prosedüründe enerji tüketilir.

3.5.1. Problem Tanımı

Enerji tüketimi ağ konumlandırmasının verimini ölçmede en önemli unsurlardan biri olup bu konumlandırma üç bölüme ayrılabilir: 1-her düğüm için aktarma, küme başı tarafından paketlerin alımı, küme başından biriktiriciye veri aktarımı için enerji kullanımı. Bu bölüm de, her düğümün aktarımındaki enerji tüketiminin enerji kullanımına yansıtıldığını varsaydık. Kromozom büyüklüğünün ölçülmesinde ağ kapsamı da bir başka önemli parametredir. Burada, bir kromozomun büyüklüğü enerji tüketimine ve ağ kapsamına bağlı olduğu kabul edilmiştir.

Bu algoritmayı çalıştırmak için, baz istasyonuna kurulabilen güçlü bir işlemciye ihtiyaç duyulmaktadır. Bunun yanında, baz istasyonu her algılayıcının konumunu bilmesi gerekmektedir. Ardından baz istasyonu algılayıcıların hangi yöne gitmeleri gerektiğine karar vermek için optimizasyon algoritmasını çalıştırır. Ayrıca bu bölüm de, her düğüm yakında bulunan düğümlere ve engellere olan mesafesini görüntüleyebildiği tüm yönlü bir kameraya ve tarayıcı lazerli mesafe ölçer algılayıcıya sahip olduğu göz önünde bulundurulmuştur. Algılayıcı ağ varsayımları aşağıda listelenmiştir. Buna ek olarak algılama ve çok amaçlı GA modelleri de tarif edilmiştir.

A. Ağ Varsayımı

Ağ algılayıcıları için aşağıdaki varsayımlar yapılmıştır:

1. Bütün algılayıcılar doğrudan veya çoklu atlama yoluyla baz istasyonu ile iletişim kurabilirler.
2. Algılayıcıların kapsamı bu algılayıcının merkezde olduğu bir daire ile tanımlanır.
3. Bütün algılayıcılar GPS veya diğer konum cihazlarına sahiptir. Hareket mesafelerindeki herhangi bir konuma (bilinen koordinatlarla) geçebilirler.
4. Algılayıcılar sınırlar ve duvar olarak görülen engeller üzerinden algılama yapamaz veya ötesine geçemezler.

B. Algılama Modeli

Algoritmamızda göz önüne alınan ilk amaç kapsamadır. Varsayılan algılama modeli mümkün olduğu kadar çok kapsanması öngörülen bir ikili modeldir. Bu durumda algılayıcıdan belli mesafe uzaklıktaki alan %100 kapsanmış olarak görülebilir ve algılama mesafesinin dışındaki alan ise bu algılayıcı tarafından kapsanmadığı için %50'den az kapsanıyor olarak ele alınacaktır [49].

Algılama sahasının şebeke olması öngörülmektedir. Bütün alanın kapsaması, kapsanabilen şebeke noktaları ile orantılıdır. Şebeke noktalarını (x,y) ve (x_i,y_i) olarak ele aldığımız zaman, bir algılayıcı düğümü $As_i(x_i,y_i)$ tarafından algılanabilme olasılığı Formül (1)'deki gibi tanımlanabilir. $As_i(x_i,y_i)$ burada A düğümünün (x_i,y_i) 'de konumlandığı anlamına gelir. R_s ise ağda bulunan algılayıcı düğüm sayısıdır.

$$\begin{cases} 1 & A_s = \frac{\sqrt{(x-x_i)^2 + (y-y_i)^2}}{R_s} \times 100 \therefore R_s \neq 0 \\ 0 & \textit{Otherwise} \end{cases} \quad (3.12)$$

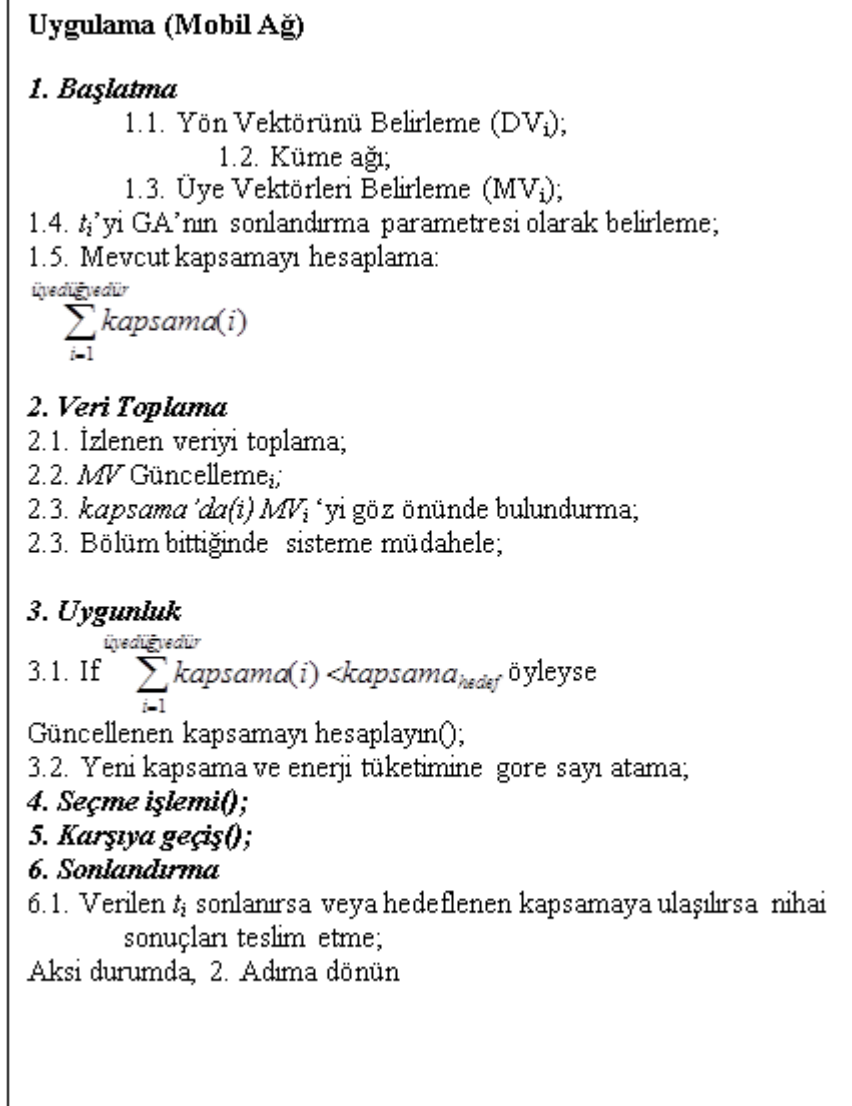
(3.12) İfadesi, A_s düğümünden algılama noktaları (x, y) ve (x_i, y_i) 'in olasılığını göstermektedir.

Fonksiyonun diğer amacı, tarafından kat edilen mesafeyi en aza indirerek algılayıcının enerjisini korumaktır. R_s algılayıcı düğümü bulunduran bir algılayıcı ağ için ortalama kat edilen mesafe (2) de verilmiştir:

$$S_i = \frac{\sum_{i=1}^N |G_i - G'_i|}{R_s} \Leftrightarrow G_i \neq G'_i \quad (3.13)$$

(3.13) Denklemi, Ortalama kat edilen mesafeyi göstermektedir.

$G(x, y)$ ve $G'_i(x'_i, y'_i)$ algılayıcı düğümünün ilk ve son konumlarına karşılık gelen noktalar. Dolayısıyla bu aynı zamanda (x, y) ve (x'_i, y'_i) arasındaki mesafedir. Bütün algılayıcı düğümleri yönetmek için ilişkili düğümlerden oluşan seçilmiş bir alt küme oluşturuyoruz [52]. İlk olarak bütün algılayıcı düğümleri küme olarak tabir edilen gruba alıyoruz ki bu da Küme Başı (KB) olarak isimlendirilen ilişkili düğümler tarafından yönetiliyor. Gelişen algoritma, Algoritma 1'de açıklandığı üzere beş adımdan oluşmaktadır. Bu algoritma, dinamik çevresel takibe yönelik GA tabanlı işlemi özetlemektedir (Şekil 3.18). Bütün düğümler toplanan bilgiyi karşılık gelen KB'leri aracılığıyla ortak Baz İstasyonuna (BS) aktarırlar. Baz istasyonu, algoritmayı yürüten ve sonuçları düğümlere ileten, sürekli güç desteği verilen bir kaynaktır.



Şekil 3.18: Mobil algılayıcı ağ için önerilen GA tabanlı algoritma.

Bu algoritma karşılıklı olarak (online durumda) t_i 'ye maksimum fayda sağlaması gereken MV_i 'yi hesaplamaktadır. T_i kullanıcı tarafından belirlenmekte olup (bu algorithmada ise 1.4 numaralı adım) bu durum istenen amaçlara bağlıdır; gelecek nesiller daha doğru bir çözüm tahmini yapmak için açıkça kullanılabilir. Ancak kullanıcı fikrine göre bu değerin saptaması yapılabilir.

Ancak ağ hareketi canlı olduğu için tekrarlama sınırlandırılmalıdır. Karşılık gelen üye düğümlerine MV_i göndermek, yeni MV 'yi aktarmak ve yüklemek için enerji gerektirir. Bu görev enerji tasarrufu sağlamak için uzun aralıklarla yapılabilir ve böylece ağın

ömrü uzatılabilir. Son olarak, atanan t değerine ulaşıldığında veya hedef kapsama alanının tümü elde edildiğinde algoritma sonlandırılır[59,60].

3.5.2. Başlatma

Başlangıç algoritmanın ilk adımı olup algoritmaya ilk değeri atamak için t_i seçilir. Düşümler kümelenir ve KB' ler belirlenir. Algılayıcı düşümlerin mobil ağdaki hareketliliğinden dolayı, önceki çalışmalara benzer şekilde [55,56] her düşüğümün durumunu belirlemek için Yön Vektöründen (DV) faydalanıyoruz. DV n -boyutlu bir vektör ($n \in \mathbb{N}$) olup $[0, 1]$ aralığına ölçeklendirilir. Bu değer, farklı yönlere hareket eden düşümlerin ihtimalini gösterir. Açıkça görüldüğü üzere bu boyutun toplamı 1,0'dır.

Aynı zamanda, DV ve KB' lerin konumlarından faydalanan GA, düşüm hareketlerini açığa çıkarır. Küme başları her düşüğümün durumundan kalan güç seviyesi ile birlikte haberdar olmalıdır. Dolayısıyla, üye-düşüm tanımlayıcı (ID) ve enerji seviyesinden oluşan Üye Vektör (MV) adındaki konsepti sunmaktayız. Çevreyi takip edip düşümlerden veri topladıktan sonra, her algılayıcı üye toplamış olduğu veriyi denk gelen KB' ye periyodik olarak aktarır ve ardından MV güncellenir. Her KB'nin bir MV_i ($i \in$ üye düşüm sayısı) dizini olup bunlar GA'nın optimum ağ kapsaması için gerekli olan enerjiyi hesaplamasına olanak sağlamaktadır.

3.5.3. Veri Kümeleme

1. Algoritmanın 2. adımını ele alalım. Bu bölüm algoritmamızın interaktif veya gerçek zamanlı özelliğini gösteriyor. Ayrıca, algoritma çalışırken uygunluk fonksiyonu için en güncel değişkenleri göz önünde bulundurmaya üzere her devirde düşümlerden ve ağdan yeni veri toplanır. Böylece kromozomlara atanan sayılar MV_i dâhil olmak üzere güncellenen değişkenlere dayanır.

3.5.4. Seçme İşlemi

Birbirini izleyen her nesil süresince, mevcut üyeler arasından uygunluk temel alınarak eşleşmeyi gerçekleştirmek üzere yeni bir topluluk seçilir. Daha uygun olan tekillere de benzer şekilde derecelendirilerek en iyi çözümün öncelikli tercih edilmesine yol açarlar. Daha küçük ve daha az uygun üyelerin topluluğun çeşitliliğini sürdürmesine

yardımcı olmaları için uygunluk fonksiyonlarının çoğu olasılıksal bir parça ile oluşturulur [52,57]. Mevcut seçim yöntemlerinin arasından, Rulet Çarkı uygun tekilleri seçilmiştir:

$$P_i = \frac{F_i}{\sum_{j=1}^n F_j} \quad (3.14)$$

(3.14)denklemini önerilen algoritmanın seçim yöntemini göstermektedir.

F_i ve n sırasıyla kromozomsal uygunluk ve topluluk boyutudur. Rulet Çarkına göre, devam eden sayısal aralıkta atanan her değer 0 ile 1 arasındadır.

4. BULGULAR

Kablosuz Algılayıcı Ağların günlük hayatta kullanımı ve uygulama alanlarının çeşitliliği hızlı bir şekilde arttığından, bu ağ teknolojisinin kısıtları üzerinde yoğunlaşmıştır. KAA'larla gerçekleştirilen uygulamaları güçleştiren en önemli kısıtlar ise enerji tüketimi ve ağ ömrüdür. Tez çalışmasında, KAA için mevcut yönlendirme protokollerini enerji kısıtı açısından değerlendirerek karşılaştırmalı performans analizleri elde edilmiştir.

4.1. KAA YÖNLENDİRME PROTOKOLLERİNİN KARŞILAŞTIRMASI

KAA yönlendirme protokolleri, genel olarak düz ve hiyerarşik yönlendirme olarak iki kategoriye ayrılabilir. Bu nedenle, Tablo 4.1'de düz ve hiyerarşik yönlendirme yaklaşımlarının genel bir karşılaştırması verilmiştir [20]. Bu çalışmada, düz yönlendirme protokolü kategorisinde bulunan Doğrudan Yayılım, Gossiping ve EESR protokolleri ile hiyerarşik yönlendirme protokolü olan LEACH yöntemlerinin performans analizleri elde edildiğinden Tablo 4.2'de bu protokollere ilişkin genel özellikler detaylı olarak verilmiştir.

Gossiping, diğer yöntemlere göre çok basittir ve herhangi bir yönlendirme tablosu ya da topoloji yönetimi gerektirmez. Gossiping protokolü Flooding protokolünü temel almaktadır. Bilindiği gibi, yüksek bağlantırlık özelliği, bir düğümün kendi komşularını algıladığı anda paketleri gönderebilmesi ve iletebilir hale gelmesi yeteneğidir. Gossiping protokolünde her bir paket tüm komşulara yayınlamak yerine, paket tek bir komşuya rastgele gönderilir, yani bir kerede bir paketin yalnızca tek bir kopyası aktarımdadır. Paketi alan komşu, göndermek için rastgele başka bir düğüm seçer. Ancak bu düğüm, paketi gönderen düğüm de olabilir. Bu işlem paket hedefine ulaşana ya da paketin maksimum sekme sayısı aşılanaya kadar devam eder.

Gossiping diğer (LEACH, Doğrudan Yayılma, EESR) yönlendirme protokolleri ile karşılaştırıldığında; orta miktarda güç tükettiği, verinin ağ boyunca yayılma (bire bir iletişim nedeniyle) gecikmesinin çok yüksek olduğu ve protokolün rastgele doğası nedeniyle sekme sayısının çok yüksek olduğu görülmektedir. Öte yandan, bir ağdaki düğüm sayısı arttıkça, paketin izleyebileceği yol sayısı ve ağı geçmek için gerçekleştirilen ortalama sekme sayısı artar. Bu yüzden, paketlerin sekme sayısı maksimum değere ulaştığında veri düşer. Büyük ağlarda bir paketin sekme sayısının bu değere ulaşması ve bu yüzden daha fazla paketin düşmesi daha olasıdır. Küçük ağlarda gönderilen paketlerin kabaca yarısı kaybolurken büyük ağlarda bu oran muazzam artış gösterir.

Tablo 4.1: Düz ve Hiyerarşik yönlendirme protokolleri arasındaki genel karşılaştırma.

Hiyerarşik Yönlendirme	Düz Yönlendirme
• Rezervasyon tabanlı planlama.	• Çekişme tabanlı planlama.
• Çatışma önlenir.	• Çatışma yükü vardır.
• Küme başları ile veri toplama.	• Düğümlerin uyku zamanının kontrolü yoluyla değişken görev döngüsü.
• Basit ancak optimum olmayan yönlendirme.	• Çok sekmeli yol üzerindeki düğüm, komşulardan gelen veriyi toplar, Yönlendirme optimize edilebilir ancak yapının karmaşıklığı artar.
• Genel ve yerel senkronizasyonu gerektirir.	• Bağlantılar, senkronizasyon olmadan anında oluşur.
• Ağ boyunca küme bilgisinin yükü.	• Rotalar yalnızca iletilecek verinin olduğu bölgelerde oluşur.
• .Küme başları tarafından oluşturulan çok sekmeli ağ her zaman kullanılabilir olduğundan daha az gecikme.	• Ara düğümlerin uyanması ve çoklu yolun oluşturulması nedeniyle daha fazla gecikme.
• Enerji kaybı eş düzeyli.	• Enerji kaybı trafik modellerine bağlı.
• Enerji kaybı kontrol edilemez.	• Enerji kaybı trafik modellerine göre uyarlanır.
• Adil kanal tahsisi.	• Adillik garanti edilmez.

Tablo 4.2: Yönlendirme protokoller arasında genel karşılaştırma.

	LEACH	Direct Diffusion	Gossiping	EESR
Sınıfı	Hiyerarşik	Düz	Düz	Düz
Ölçeklenebilirlik	Yüksek (düğümler kümelerle bölünür)	Sınırlı (düz topolojinin doğası nedeniyle)	Sınırlı (düz topolojinin doğası nedeniyle)	Yüksek (uygulama alanı sektörlere bölünür)
Ömrü	Çok İyi (TDMA nedeniyle, algılayıcıların çoğu uyku modundadır ve KB'nin rolü diğer düğümlere dağıtılır)	İyi (İstek bazlı ve komşudan-komşuya iletişim)	Orta (yüksek gecikme sorunu vardır)	Çok İyi (Ön tanımlı Yönetici Düğümü tahsisi ve sektörlerin kullanılması nedeniyle)
Enerji Verimliliği	Yüksek (düğümden küme başına tek sekmeli yönlendirme)	Yüksek (çoğunlukla ilgi için belirli bir rota seçer ve rotalar yalnızca bir ilgi olduğunda oluşur)	Orta (LEACH ya da EESR gibi değildir. Ancak yönlendirme tablosu ve topoloji bakımı gerekmez)	Yüksek (Yalnızca 1 sekme mesafedeki komşulara gönderir ve veriyi iletmek için farklı rotalar seçer)
Veri toplama	Var (veri KB'ler tarafından toplanır)	Var (her düğüm veriyi toplar ardından sonraki sekmeye yeniden yönlendirir)	Yok (her düğüm veriyi toplar ardından sonraki sekmeye yeniden yönlendirir)	Var
Görüşme tabanlı	Yok (sinyal gücüne göre)	Var (Görüşme, Gradyan oluşumu aşamasında yapılır)	Yok (bir düğüm gönderim için komşu bir düğümü rastgele seçer)	Var (optimum rotayı bulmak için EESR tablosu oluşturulur)
Kaynak Bilinirliği	Var	Var	Yok	Yok

Tablo 4.2 (devam): Yönlendirme protokoller arasında genel karşılaştırma.

Sekme İletişimi	Tekli Sekme (Üye-KB ve KB-Baz İstasyon)	Çoklu Sekme (diğer düğümler üzerinden kaynaktan baz istasyona)	Çoklu Sekme (diğer düğümler üzerinden kaynaktan baz istasyona)	Çoklu Sekme (baz istasyona kadar her defasında 1 sekme)
Optimum Rota	Yok (Üyenin KB'ye yeniden yönlendirme için bir şansı vardır)	Var (Güçlendirme aşaması)	Yok (rastgele seçim)	Var (her defasında kendi EESR tablosunu kontrol ederek en iyi Sektör Kimliğini bulur)
Gecikme	Az (düğüm kendi verisini doğrudan KB'ye gönderir)	Yüksek (ilgi yayılması sırasındaki akış nedeniyle)	Çok yüksek (veri yayılması nedeniyle)	Az (aşırı yük yoktur ve ilk olaydan sonra, sonraki sekme kolayca bulunabilir)
Verim	Çok yüksek (Gecikme yoktur ve bir kerede yalnızca bir düğüm kanala erişir)	Kabul edilebilir	Düşük (yüksek gecikme nedeniyle)	Yüksek (Optimum yol seçili, gecikme ve aşırı yük yoktur)
Aşırı yük	Yok (KB veriyi birden çok düğümden toplar)	Var (Veri ve sorguların eşleştirilmesi işlemi sırasında aşırı yük)	Yok (doğrudan komşu bir düğüme gönderir)	Yok (bire bir iletişim)
Senaryo tipi	Olay tabanlı (yalnızca bir olay olduğunda algılayıcı bunu algılar)	Sorgu tabanlı (Sorgular, yalnızca bir istek olduğunda kullanıcı tarafından yayınlanır)	Olay tabanlı (yalnızca bir olay olduğunda algılayıcı bunu algılar)	Olay tabanlı (yalnızca bir olay olduğunda algılayıcı bunu algılar)

Tablo 4.2 (devam): Yönlendirme protokoller arasında genel karşılaştırma.

Uygulamalar	İzleme uygulaması. (dinamik uygulama. -bir olay olduğunda bir düğüm bunu algılar)	Çok Amaçlı Uygulamalar. (birden fazla alıcı aynı anda sorgu üretip veri alabilir)	Uygulama ihtiyacı. (bire bir iletişim)	İzleme uygulaması . (dinamik uygulama)
Uygulama Türü	Sağlık izlemesi (yapay retina)	Ortam izleme (PODS Hawaii)	Geliştirme aşamasında ortam izleme ve ağ başlatma	Ortam izleme, örn, Tarım uygulaması

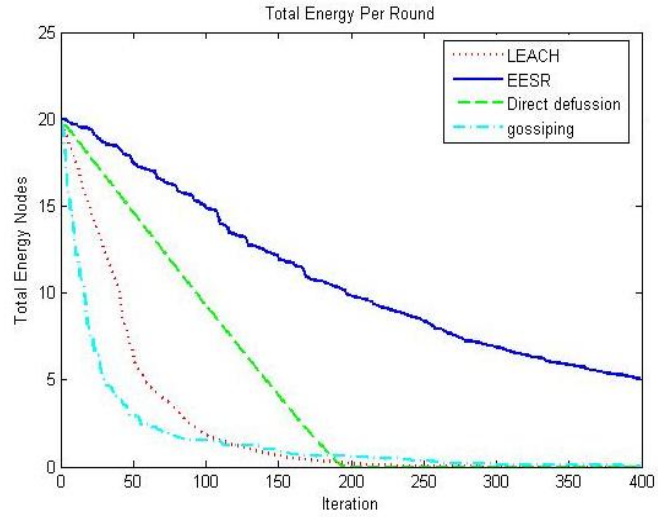
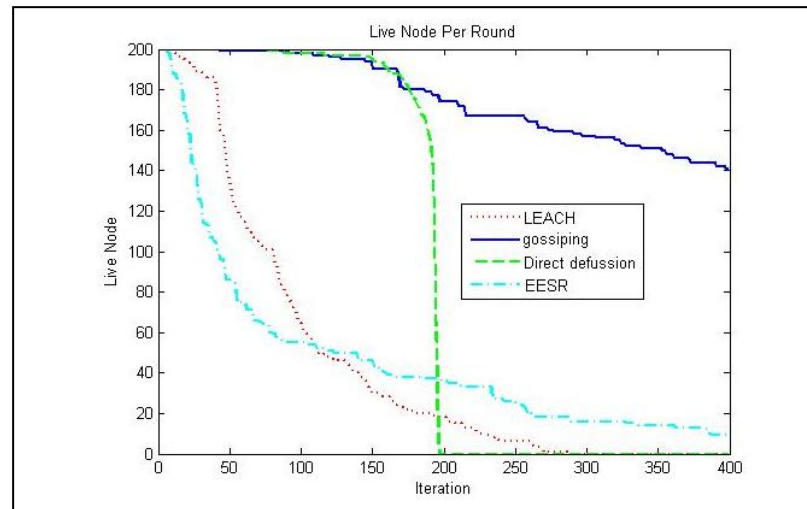
Gossiping Protokolünde bazı değişiklikler yaparak enerji tüketimini azaltabilir ve ağ ömrünü artırabiliriz. Bu nedenle Gossiping Protokolünün dezavantajlarını iyileştirmek amacıyla, Gossiping'e bir eklenti olarak Flossiping, SGDF, LGossiping ve ELGossiping gibi birçok yeni protokol önerilmiştir.

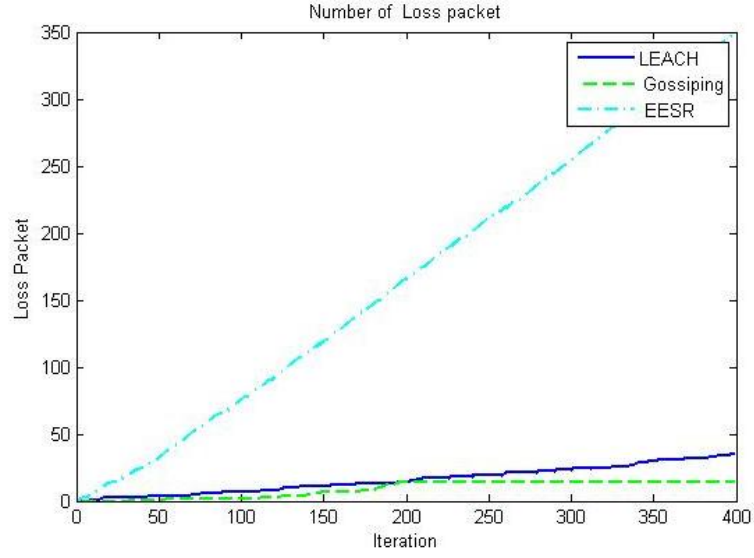
4.1.1. Simülasyon Sonuçları

Bölüm 4.1'de, KAA yönlendirme protokolleri özel parametrelere göre sınıflandırılarak (Tablo 4.1 ve Tablo 4.2) birbirleriyle karşılaştırılmıştır. Tablo 4.2'den de görülebileceği gibi, özellikle enerji konusunda en uygun Gossiping protokolüdür. Tablo 4.3'de NS2 ve Matlab programları kullanılarak yapılan benzetim sonuçları verilmiştir. Şekil 4.1 incelenen yönlendirme metodlarının toplam enerjilerinin iteratif değişimini, Şekil 4.2 hayatta kalan toplam düğüm sayılarının değişimini, Şekil 4.3 ise, yönlendirme esnasında kaybolan paket sayılarını, Şekil 4.4, kaynak düğümden varış noktasına kadar olan düğüm atlama sayısını, Şekil 4.5 ise alınan paketlerin sayısını göstermektedir.

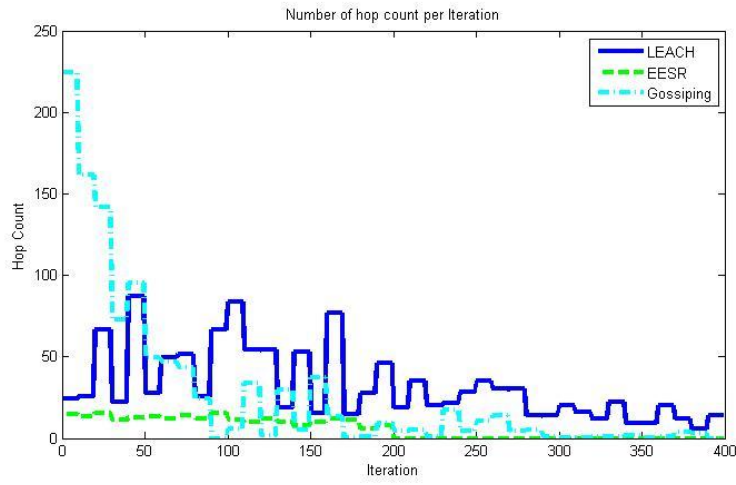
Tablo 4.3: Simülasyon genel parametreleri.

Parametre	Değer
Ağ ölçüsü	200*200 m2
algılayıcı sayısı	200
başlangıç gücü	0.1 J
Eelec	50 nJ/bit
ϵ_{fs}	10 pJ/bit/m2
ϵ_{mp}	0.0013 pJ/bit/m4
Data Paket boyutu	4000 bits
Kontrol Paket boyutu	32 bits

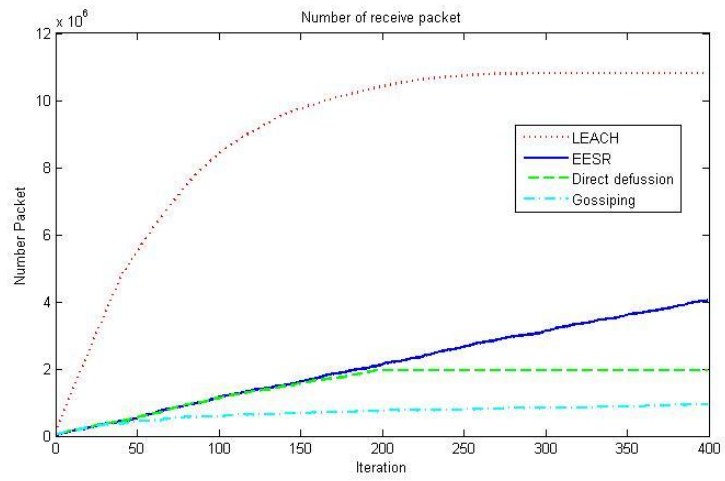
**Şekil 4.1:** Toplam enerjinin iteratif değişimi.**Şekil 4.2:** Hayatta kalan toplam düğüm sayılarının değişimi.



Şekil 4.3: Kaybolan paket sayısının değişimi.



Şekil 4.4: Her bir işlemdeki atlama sayısını gösteren grafik.



Şekil 4.5: Alınan paket sayısını gösteren grafik.

Bilindiği gibi, Kablosuz Algılayıcı Ağlarının gücü sınırlı kapasiteli piller tarafından sağlanmaktadır. Bu algılayıcı düğümlerinin güç yönetimi etkinlikleri nedeniyle, ağ topolojisi dinamik olarak değişir. Ayrıca, Gossiping protokolü üzerinde sonraki sekmenin seçiminde bazı değişiklikler yapılırsa, ağ ömrünün artırılabilceği tespit edilmiştir (Şekil 4.5).

4.2 GENETİK ALGORİTMA YAKLAŞIMI KULLANILARAK ELDE EDİLEN SONUÇLAR

Bu bölümde, genetik algoritma yardımıyla optimum uygunluk fonksiyonları elde edilerek tezde incelenen protokoller optimize edilmiştir. Protokollerin simülasyonları JPAC, MATLAB ve NS gibi araçlar kullanılarak yapılmıştır. Sonuçlara dayanarak enerji tüketimi ile ağ ömrü parametreleri açısından protokollerin optimizasyonları doğrulanmıştır. Daha ideal Kablosuz Algılayıcı Ağların geliştirilebilmesi, enerji tüketimi parametreleri ile ağ ömründe gösterilen gelişmelere bağlıdır.

4.2.1. Düğüm Yerleşim Optimizasyonu

Kablosuz Algılayıcı Ağlarının performanslarının daha iyi hale getirilmesinde rol oynayan temel faktörlerden bir tanesi devre düğümü yerleşimidir. Bu bölümde, Kablosuz Algılayıcı Ağlarının (KAA) düğüm yerleşim optimizasyonu ile ilgili bulgular verilmiştir.

Bu çalışmada, algılayıcıların iletişim alanının sabit olduğu kabul edilmiştir ve Akıllı Devre Yerleştirme Protokolü olarak Genetik Algoritma kullanılmıştır. Tezin temel hedefi olan Kablosuz Algılayıcı Ağların kapsama alanı ve ağın kullanım ömrü optimize edilmiştir. Genetik algoritma yardımıyla mesafe değerlendirmesi minimum sayıda düğüm (devre) kullanarak kapsama altına alınırken optimum enerji tüketimi sağlanmıştır.

4.2.1.1. GA Uygunluk Fonksiyonu Seçimi

Genetik Algoritma minimum düğüm sayısının hesaplanmasını sağlar. Bir sonraki aşama, ağ mimarisini iyileştirmek için ağ bağlantısının değerlendirilmesidir. Algoritma buna ek olarak, uygulanan devrelerin bağlantı etki alanını da göz önüne alır. Bu durum

uygunluk fonksiyonunun farklı ağ türlerinin ölçümü için de kullanılabilir esnekliğe sahip olduğunu gösterir. Burada, kapsama alanının genişletilmesini temel alan bir uygunluk fonksiyonu tanımlanmıştır. Bu fonksiyonun temel amacı, daha geniş kapsama alanı için optimum bir çözüm bulunması ve bu esnada enerji tüketimi verimliliğinin korunmasıdır. Burada göz önünde tutulan iki temel hedef, ağın kapsama alanı ve kullanım süresidir. Ağın kapsama alanı, her bir algılayıcının merkezinde bulunan etki alanı R ile elde edilen disk birim alanı ile belirlenir. Denklem 4.1'e göre, birleşim alanı toplam alan tarafından normalize edilir [41].

$$kapsama\ alanı = \begin{cases} \bigcup_{i=1}^n R_{x_i, y_i}^2 & \because |(x_i, y_i) - (x_{i-1}, y_{i-1})| = (0,0) \\ \bigcup_{i=1}^n R_{x_i, y_i}^2 - r_{x_i, y_i}^2 & \because o.w. \end{cases} \quad (4.1)$$

$$burada\ r_{x_i, y_i} = |(x_i, y_i) - (x_{i-1}, y_{i-1})|$$

Amaç fonksiyonu olan minimum enerjinin hesaplanması, Quint F.P ve ark. tarafından Denklem 4.2'deki gibi elde edilmiştir [40].

$$\min_{f} = \sum_{i \in S} (\varepsilon + d_i) * y_i + \sum_{j \in D} NC_j * h_j \quad (4.2)$$

Burada; ε sabiti bir devrenin oluşturulması için gerekli olan enerjiyi belirtir; d_i ise yukarı yön ile aşağı yön arasındaki yönlendirme maliyetini ifade eder. Bu dayanak, program başlatılmadan önce Dijkstra algoritması kullanılarak hesaplanır. Elde edilen değer, uzakta bulunan devreler için bir tür ceza niteliğindedir. NC ortaya çıkarma noktalarının fazla yüklemesidir (sürşarj); y_i i. düğümün aktivasyon durumunu, h_j ise j noktasının kapsama durumunu göstermektedir.

$$uygunluk\ fonksiyonu = \min_{f_{i,j}} / kapsama\ alanı_{i,j} \quad (4.3)$$

Düzenli bir ağ kurulabilmesi için gerekli olan enerji seviyesi payı ve enerji miktarı (4.3)'deki uygunluk fonksiyonu kullanılarak belirlenebilir.

4.2.1.2. Simülasyon Sonuçları

Deneyle 200 düğüm (N), 100*100 m² boyutlarında bir ağ (M) ve de ağdan 200m uzaklıkta bir Baz İstasyonu (BS) kullanılarak gerçekleştirilmiştir. Simülasyon parametreleri Tablo 4.4’de verilmektedir. Algılayıcılar arasındaki iletişim ortamı nedeniyle KAA bağlantılabirlik alanının optimizasyonu önemli ölçüde doğrusal değildir (nonlinear optimization). Genetik Algoritma tekniği doğrusal olmayan optimizasyon problemlerinin çözümünde kullanılan en efektif yöntemlerden biridir[41].

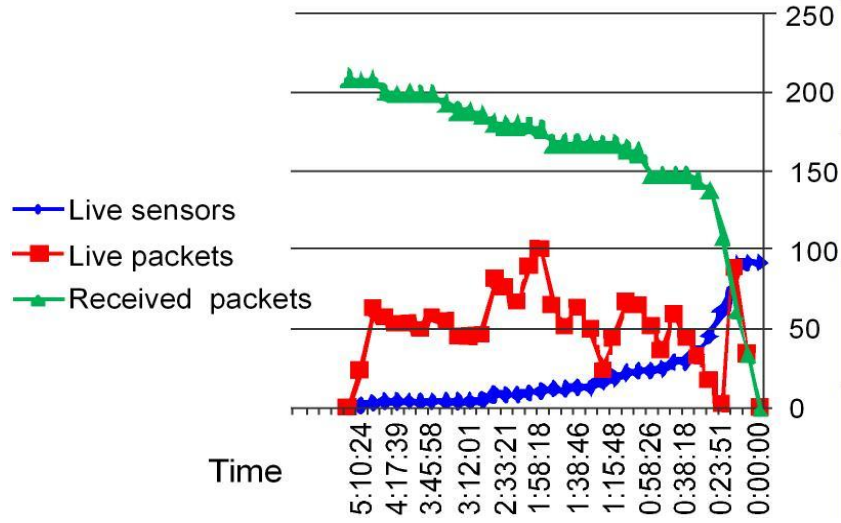
Tablo 4.4: Simülasyon parametreleri.

Ağ boyutu	100 m
Devre Sayısı	200
Başlangıç enerjisi	2 J
E_e	50 nJ/bit
ε_l	0.0013 pJ/bit/m ²
ε_s	10 pJ/bit/m ²
Ağ alanı	100*100 m ²
BS mesafesi	200 m
Paket boyutu	200 bit
$d_{co}=d_{crossover}$	85 m

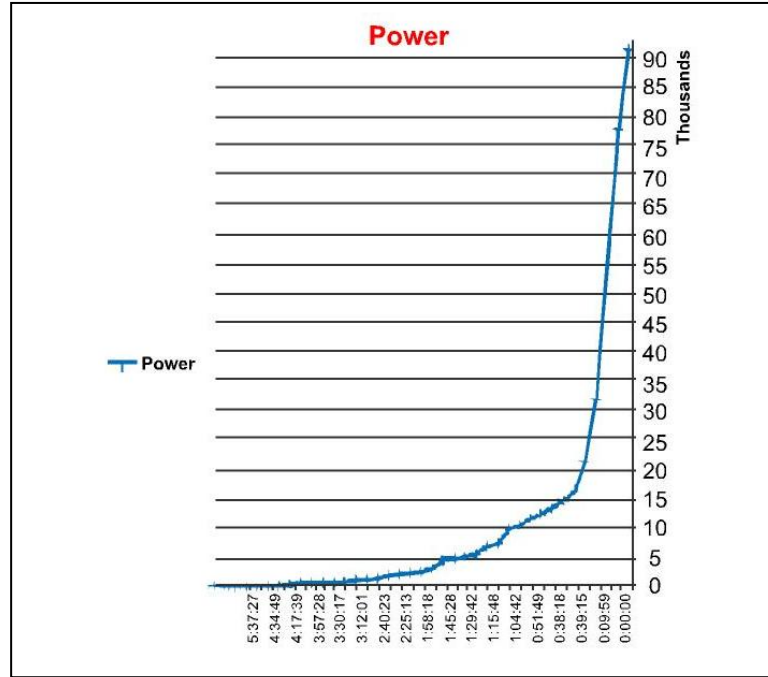
İletim Ortamının simülasyonu için kullanılan GA parametreleri Tablo 4.5’de verilmiştir. Nihai sonuçlar üzerindeki herhangi bir etkileri olmadığı için kromozomlar tesadüfi olarak seçilebilir. Bunun anlamı, kromozoma bağlı olmaksızın, optimum sonuç elde edilebilmesidir. Burada yineleme (iterasyon) sayısı sabit (100) alınmıştır.

Tablo 4.5: GA parametre değerleri.

Aday bireylerin sayısı	100
Kromozom uzunluğu	20
Çaprazlama (Crossover) Oranı	0.5
Değişim (Mutasyon) Oranı	0.2
Yineleme (İterasyon)	100



Şekil 4.6: Mevcut Algılayıcı sayısı ile ağ içerisinde bulunan aktif ve alınan paket sayılarının karşılaştırımı.



Şekil 4.7: Güç dayanağı ile ağın kullanım süresinin karşılaştırımı.

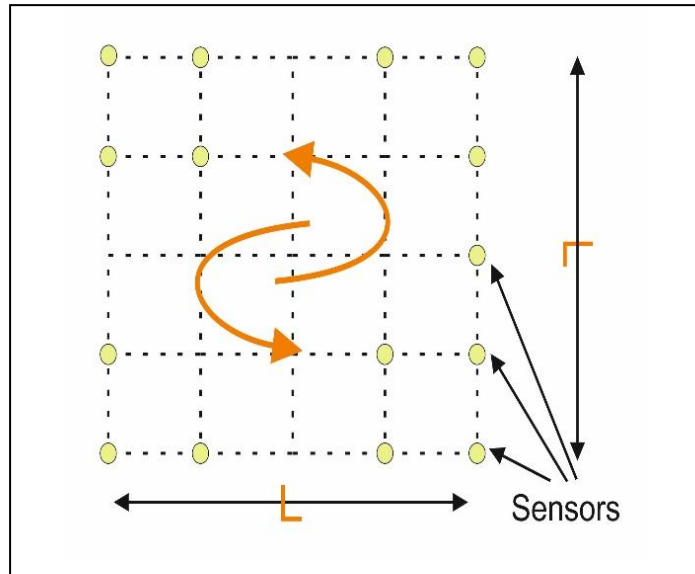
200 paket üzerinde gerçekleştirilen deneylere ait ortalamalar Şekil 4.6 ve Şekil 4.7’de verilmiştir. Şekil 4.7’de, ağır paketler nedeniyle, ağ enerjisi 0:39:15 sn. sürede büyük oranda azalırken, Şekil 4.6’de devam eden paket transfer oranı 5:10:24 sn. süre devam ettiği görülmektedir. Ağ düğümlerinin optimum yerleşimi nedeniyle, ağır paketlerin etkisi tolere edilerek ağ ömrü optimize edilmiştir. Burada, bir düğümün ölmesi durumunda, alıcıya veri gönderimi fonksiyonunu yitirmemiş olan komşu devre

tarafından gerçekleştirilmektedir. Sonuç olarak, önerilen algoritma optimaldir ve ağıın kullanım süresini uzatmaktadır.

4.2.2. Kapsama Alanının Belirlenmesi

KAA'ların kapsama alanı, fiziksel alanın gözlenmesinde algılayıcıların kullanım ömrü performansının bir ölçütü olarak tanımlanır. Kapsama alanı aynı zamanda algılayıcı ağının bağlanılabilirliği için de kritik bir faktördür. Bir terim olarak bağlanılabilirlik, Algılayıcı devrelerinin veri alıcısı ile iletişim kurabilme kapasitesini belirtir. Gerçek dünyada kullanılan KAA uygulamasını temel alarak, kapsama alanı ile ilgili durumun değerlendirilmesi için 2-D bir alan içerisinde kuramsal parametreler (A, B, C) belirlenmiştir.

Bu çalışmada, parametreler gerçek duruma mümkün olduğunca yakın olacak biçimde tanımlanmıştır[38]. Uzamsal A, B ve C değişkenleri, $\rho_A \ll \rho_B \ll \rho_C$ koşullu alanların her birindeki algılayıcı yoğunluklarını temsil etmektedir. Bu kavram, ağların genel özelliklerine ek olarak, özel amaçlı ağların kendine has özelliklerini de açıklamaktadır. Bu çalışmada, tüm kapsama alanının birim uzunluklu Öklit kare alanlarından oluştuğu kabul edilmiştir. Diğer çalışmalarda kullanılan konfigürasyon, grid-tabanlı kablosuz Algılayıcı ağ yerleşimidir [38]. Şekil 4.8'da KAA için belirlenen kapsama alanının genel şeması görülmektedir.



Şekil 4.8: Kablosuz Algılayıcı ağlarında kapsama şeması.

KAA’da kullanılan küçük algılayıcılar sınırlı güce, sınırlı iletim mesafesine, sınırlı kapasiteye ve duruma bağlı olarak sınırlı algılama modu opsiyonuna (üç çalışma modu bulunmaktadır) sahip olduğu kabul edilmiştir. A parametresinin düşük yoğunluğu algılayıcının en uzun iletim mesafesine sahip olmasına; C parametresinin düşük yoğunluğu ise algılayıcının en kısa iletim mesafesine sahip olmasına işaret etmektedir. Bu yaklaşımda, komşuluğunda belirli algılayıcılar bulunan ve de aynı çalışma moduna sahip olan kümelerde optimum enerji tüketimi sağlanabilmesi için, bir kümeleme çözümü oluşturulmuştur. Tüm kümeler baz istasyonu (BS) ya da alıcı ile iletişim kurmak için çok-sekmeli (multi-hop) iletim kullanılabilir. Normal çalışma esnasında, sorumlu küme ortam izleme ve belirli periyotlarda veri birleştirme görevini gerçekleştirir ve verileri BS’ye aktarır. Bu bölümde, istenen alanlarda izleme ve ölçüm yapma ile ilgili üç ana parametrenin (bağlanılabilirlik, enerji tüketimi ve kapsama alanı (ECEP)) optimizasyonu için kullanılacak çok amaçlı algoritma verilmiştir.

4.2.2.1. Seçilen Uygunluk Fonksiyonu

Mümkün olan en az sayıda kısıtlamaya (örn. çalışma enerjisi, bağlantısını kaybetmiş devre sayısı ve sorumlu kümedeki çakışma hataları) sahip olan bazı olası optimum ağ topolojilerinin oluşturulabilmesi için yeni bir algoritma türü kullanılmıştır. Tüm çalışma modlarını muhtemel durumları içinde ele alan uygunluk fonksiyonu göz önüne alındığında, bu teknik uygulanan tüm parametrelerin değerlendirmesini yapmaktadır.

4.2.2.1.1. ECEP’lerde Kapsama Problemi İçin Formül

Genetik Algoritmanın bir parçası olan doğru uygunluk fonksiyonlarını bulmak için Quintao ve ark. [42] tarafından geliştirilmiş olan bir formül kullanılmıştır; söz konusu formül Nakamura’ya ait olan formülün geliştirilmiş versiyonudur [43]. Formülde; ‘A’ verilen izleme alanını, ‘S’ algılayıcı devreleri setini, ‘D’ istenen noktalar setini, ‘A^d’ Algılayıcılar tarafından izlenmesi istenen alanlar setini, ‘NC’ istenen alandaki kapsama yetersizliği için uygulanacak ceza maliyetini ‘AE’ enerjinin aktif hale getirilmesini, ‘PC’ devrelerden BS’ye giden yol için ceza maliyetlerini (Dijkstra algoritması kullanılarak hesaplanan) göstermektedir. Bu modelin değişkenleri ise şöyledir:

$$\begin{aligned}
 x_{ij} &= 1, & \text{‘i’ devresi j talep noktasını kapsadığında ya da 0 olduğunda} \\
 y_i &= 1, & \text{‘i’ devreleri aktif olduğunda ya da 0 olduğunda} \\
 h_j &= 1, & \text{‘j’ talep noktası kapsanmadığında.}
 \end{aligned} \tag{4.4}$$

Kısıtlar:

$$\sum_{ij} (x_{ij} + h_j) \geq 1, \forall j \in D \ \& \ \forall ij \in A^d \quad (4.5)$$

$$x_{ij} \leq y_i, \forall i \in S \ \& \ \forall ij \in A^d \quad (4.6)$$

$$0 \leq x_{ij} \leq 1, \forall ij \in A^d, h_j \geq 0, \forall j \in D \quad y \in \{0,1\}, \{x, h\} \in \mathfrak{R}$$

Bir Algılayıcı tarafından izlenmesi talep edilen her nokta ya da kapsama dışında tutulması istenen noktalar (4.4), (4.5) ve (4.6) maddelerinde belirtilen kısıtlar yardımıyla gösterilebilir; buna göre, yalnızca aktif durumdaki devreler sezme kapasitesine sahiptir.

Çalışmada üst üste binen aktif küme hatalarının ceza maliyeti; sırasıyla OPCE ve EC damgalı enerji tüketimleri göz önüne alınarak uygunluk fonksiyonu (FF) aşağıdaki gibi geliştirilmiştir.

$$FF = \min(Usage_Cost + Penalty_Cost)$$

$$Usage_Cost = \sum_{i \in S} (AE_i + PC_i + EC) \times y_i$$

$$Penalty_Cost = \sum_{j \in M} \sum_{k \in D} (NC_{jk} + OPCE_{jk}) \times h_k \quad (4.7)$$

Kısıt:

$$M \in \{A, B, C\}$$

EC'nin sayısal değeri, algılayıcının ağ içerisinde bulunduğu moda bağlı olarak hesaplanmıştır. Görüldüğü gibi, geniş iletişim mesafesi için Algılayıcının 'A' modunda çalışması gerekmektedir. 'C' moduna kıyasla 'A' ve 'B' modları için sırasıyla 4 ve 2 kat enerji kullanımı söz konusuysa, EC sayısal değeri (4.8)'deki gibi hesaplanır.

$$EC = \frac{4n_A + 2n_B + n_C}{\sum_{i \in S} n_i} \quad (4.8)$$

Uygunluk Fonksiyonu (FF) içeriğinde bulunan OPCE değişkeni göz önüne alındığında, aktif kümede gerçekleşen ve üst üste binen hata nedeniyle israf edilen enerji miktarı bulunur.

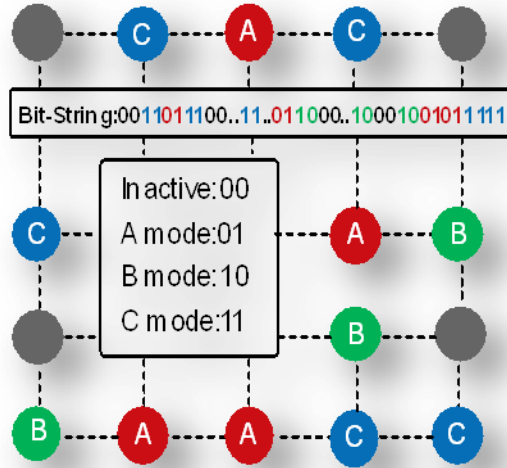
4.2.2.1.2. ECEP Bağlanılabilirliğinin Belirlenmesi

Bu tezde, Kruskal algoritması kullanılarak devre bağlanılabilirliği açısından optimum çözüm elde edilmiştir. Çalışmada önerilen çözüm sürecinin dört aşaması vardır:

- Ağ bir G grafiği olarak kabul edilir. 'x' ve 'y' arasındaki uzaklık 'x' ve 'y' devreleri arasındaki iletişim mesafesinden daha fazla olduğunda, söz konusu G grafiğindeki 'x' ve 'y' tepe noktaları arasında bir köşe oluşur. Kruskal algoritmasının kullanılmasıyla, minimum bir kapsama ağacı (MST – Minimum Spanning Tree) oluşturulur ve böylece kümelenmiş verilerin yönlendirilmesi için tepe noktaları arasında daha kısa bir yol oluşturulur.
- En kısa yollarda bağlanılabilirlik sağlanması için MST ağacı köşelerinin sayısı ile tepe noktalarının sayısı aynı, -1 olmalıdır; aksi takdirde, aktif olmayan devreler aktif hale geçecektir (bu durum, bazı devreler için mevcut iletişim mesafesinden daha fazla iletim mesafesi oluşunu açıklar). Kruskal tekniği yeni aktive edilmiş ve bağlantısı kesilmiş devreler için kullanılır. Bunun sonucunda ise yeni ve hafif bir ağaç oluşur.
- Bağlantısı kesilmiş devre ile BS arasında en kısa mesafe sağlanır ve bu yollarda bulunan dahili Algılayıcı devreleri E setine eklenir.
- E'de bulunmayan, yeni aktive edilmiş tüm devreler deaktive olur. Böylece, ağ enerjisi korunurken kalitenin aynı kalması sağlanır.
- Son olarak, devrelerin iletim mesafesi ve Algılayıcı devrelerinin konumları temel alınarak, bir ya da iki ağ topolojisi geliştirilmiştir. Maksimum kapsama alanına sahip ağların en önemli özelliği optimum kapsama alanı ve enerji kullanımüdür.

4.2.2.1.3. Şifreleme Yöntemi

Önerilen yaklaşım sanal olarak eşit alt-alanları bulunan bir kare alan ($L \times L$) içerisinde uygulanmıştır. Devreler alt-alanların kesişme noktalarına yerleştirilmiştir ve dört ayrı ifadeye sahiptir: 1- aktif (00), 2- mod A aktif (01), 3- mod B aktif (10), ve 4- mod C aktif (11). Şekil 4.9'da şifrelenmiş ağ gösterilmektedir.



Şekil 4.9: Temsili şifreli ağ.

Görüldüğü gibi, devrelerin yerleşimi bir kromozomu andırmaktadır. Yani, devrelerin her biri bir geni temsil etmekte ve belirli bir düzene sahip olan tüm gen dizileri de bir kromozom oluşturmaktadır. Böylece, her bir devre 2-bit ile gösterilmiştir ve ağ içerisinde L2 genleri bulunduğu için ,bulunan L devrelerinde 2L2 bulunmaktadır.

4.2.2.2. Değerlendirme ve Simülasyon Sonuçları

Çalışmada, WSN simülatörü kullanarak önerilen algoritma ile izlenen alanın neredeyse %100'ünde kapsama sağlanmıştır. Zaman içerisinde canlı kalan paketlerin yanı sıra, enerji tüketimi seviyesi ve aktif devrelerin sayısı Tablo 4.6'te verilmiştir. Başlangıçta BS'ye iletilen paketlerin sayısı yalnızca 18 iken daha sonra bu sayı 83 pakete çıkmıştır.

Tablo 4.6: Ağın ilk zamanlarında tespit edilen değerler.

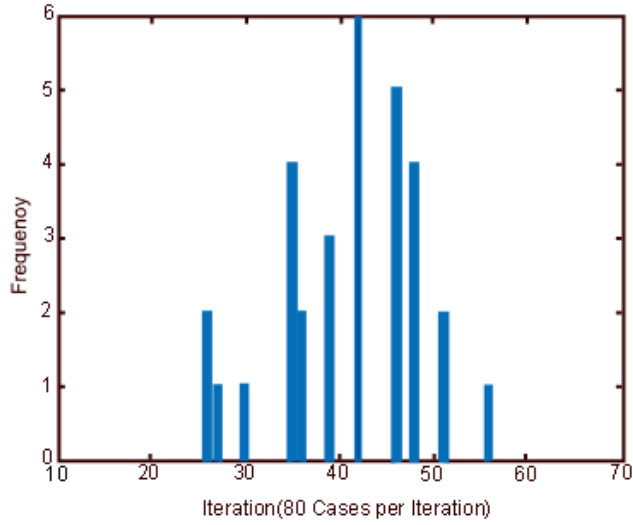
Zaman (Nanosaniye)	Güç	Aktif Algılayıcılar / 35	Canlı Paketler
00:35.090	20718	27	18
00:49.330	15332	27	60
01:08.107	8031	20	53
01:18.182	6739	18	83
01:36.719	5333	14	30
01:48.626	4368	12	36
02:02.956	3354	10	38
02:09.666	3123	8	38
02:18.048	2897	8	47
02:46.709	2032	7	54
02:55.352	1860	6	52

Tablo 4.7’de, ağın en son ölüm zamanı 5:13:781 olarak görülmektedir. Bunun yanı sıra, aktif algılayıcı sayısı 0’a yaklaştıkça canlı paketlerin sayısında yavaş bir düşüş söz konusudur.

Tablo 4.7: Ağın son zamanlarında tespit edilen değerler.

Zaman (Nanosaniye)	Güç	Aktif Algılayıcılar / 35	Canlı Paketler
03:22.821	1096	5	87
03:45.804	616	4	89
03:58.382	421	3	75
04:15.196	307	2	33
04:39.882	122	1	25
05:13.781	0	0	0

Genetik algoritma tekrarında (iterasyon) devre sayısının ağ ömrü üzerindeki rolü Şekil 4.10’da gösterilmektedir. 50 ve 57. tekrarların da gösterdiği gibi, birey sayısının fazla oluşu her zaman daha iyi sonuçlar vermemektedir.



Şekil 4.10: Verilen ölçekte ağ ömrü.

4.2.3. KAA'de kümeleme

Enerji tüketimi ile uzaklık parametreleri arasında negatif bir ilişkisi bulunmaktadır. Üye devreler ile bağlı buldukları Küme Başı arasındaki mesafeyi azaltma yollarından bir tanesi daha fazla küme kullanılmasıdır; her kümenin bir ya da birden fazla küme başı bulunabilir; bu yöntem, enerji tüketimi açısından ekonomiktir. Ancak, daha fazla küme kullanımı daha uzun mesafelerin oluşmasını engeller. Buna bağlı olarak, her devrede ortalama bir enerji tüketimi sağlayabilmek için toplam enerji kullanımının toplam devre uzaklığına oranı belirlenmiştir.

4.2.3.1. Seçilen Uygunluk Fonksiyonu

Genetik Yaklaşım kullanarak, optimal KAA enerji tüketimi ve kapsama alanı kısıtı altında, küme tabanlı sınıflandırma için önerdiğimiz uygunluk fonksiyonu Denklem 4.9'da gösterilmiştir.

Bu denklemde, $((e_i \cdot T) \times e_j \cdot T)$ toplam enerji tüketimini; $((D_a \times \text{nodes}) \times (D_b \times \text{CHs}))$ ise devreler ile kümeler arasındaki toplam mesafe ile küme başları arasındaki toplam mesafenin çarpımını göstermektedir [44].

$$F(i) = \left(\frac{e_i \times T}{D_a \times \# \text{Nodes}} \right) \times \left(\frac{e_j \times T}{D_b \times \# \text{CHs}} \right) \quad (4.9)$$

$$\begin{aligned}
& : D_a = \frac{Width \times g_i}{\sqrt{\#Clusters}} \\
& \forall g_i \in \{DDBS, CD, CDS\} \\
& [g_i = DDBS] = 1 \because [\#clusters = 1] = 1 \\
& \therefore (D_a = D_b = \#CHs = 1)
\end{aligned} \tag{4.10}$$

Küme-tabanlı ya da doğrudan metod kullanarak F(i) uygunluk fonksiyonu ile tüm kromozomları sınıflandırabilir. Optimum sonuç için gerekli olan optimum kromozom seçimi etkinliğini kaybetmekte olan jenerasyon temel alınarak yapılır.

4.2.3.1. Değerlendirme ve Simülasyon Sonucu

Burada önerilen GA-tabanlı yaklaşım ile diğer küme-tabanlı protokoller (örn. LEACH) arasında bir karşılaştırma yapılmıştır. Simülasyonda kullanılan parametreler Tablo 4.8’te verilmektedir. Her kümede yalnızca bir KB bulunmaktadır. KB sayısının belirlenmesi için genetik algoritma işlemi kullanılmıştır. Simülasyonun GA parametreleri Tablo 4.9’da verilmektedir. Kromozomların rastgele seçim yöntemi ile seçilmesi mümkündür. Yineleme (iterasyon) sayısı sabittir (100).

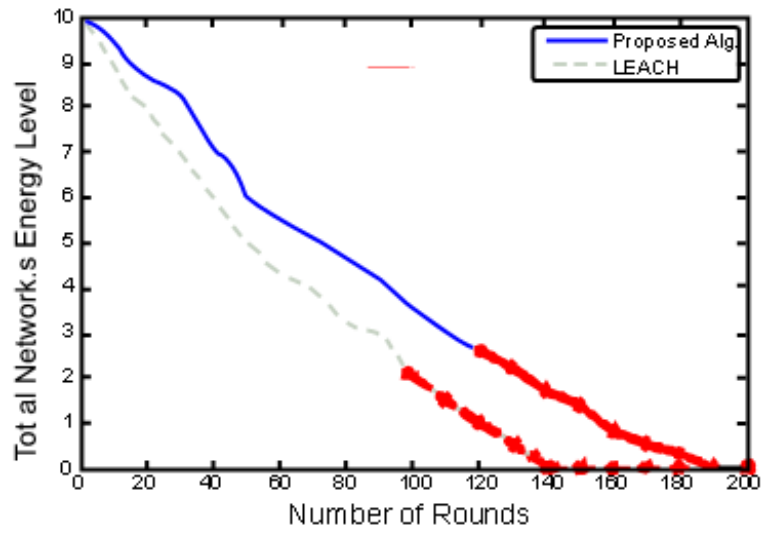
Şekil 4.11 ve Şekil 4.12’te, önerilen algoritma ve LEACH arasında ağ enerji kullanımı ve ağ ömrü açısından yapılan karşılaştırmalar gösterilmektedir. Karşılaştırma 200 zaman periyodu içerisinde yapılmıştır. Şekil 4.11, LEACH yönteminde KB’lerin birleştirilmiş enerji tüketimi sonucunda devrelerin ömrünün kısaldığını, Şekil 4.12 ise devreler enerji durumuna bağlı olarak sistemden çıkarıldığı için önerilen algoritmadaki birinci devrenin ölümü LEACH’e kıyasla daha geç gerçekleştiğini göstermektedir. Ancak buna rağmen, ağ çalışmaya minimum canlı devre sayısı ile devam etmektedir. Genel olarak, geride kalan bireyler tek tip enerji tüketimine sahip bir küme oluşturmaktadır. Bu oluşum ise devrelerin enerji durumunu ve KB/BS mesafelerini göz önüne alan algoritma uygunluk fonksiyonu sayesinde gerçekleşmektedir. Böylece, ağ ömrü önemli ölçüde uzamaktadır. [44]

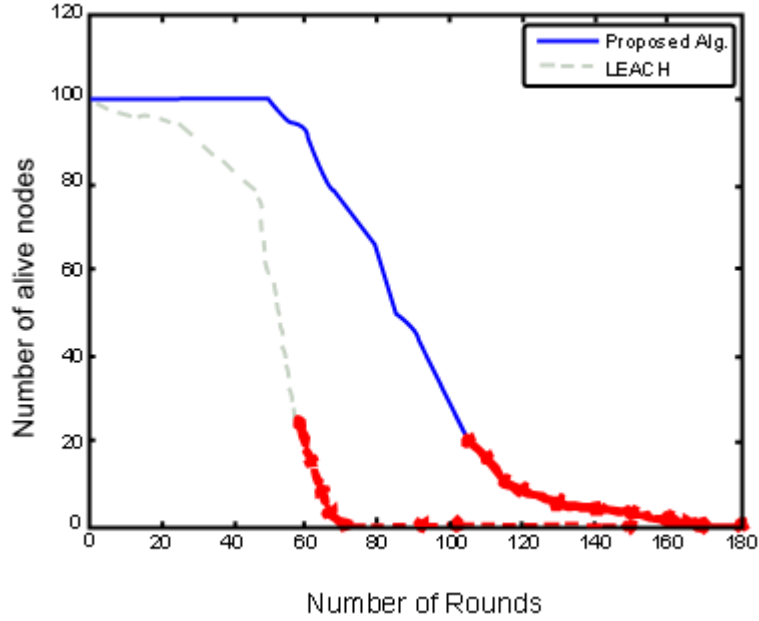
Tablo 4.8: Simülasyon parametreleri.

Ağ boyutu	100 m
Devre Sayısı	200
Başlangıç enerjisi	2 J
E_e	50 nJ/bit
ϵ_l	0.0013 pJ/bit/m ²
ϵ_s	10 pJ/bit/m ²
Ağ alanı	100*100 m ²
BS mesafesi	200 m
Paket boyutu	200 bit
$d_{co}=d_{crossover}$	85 m

Tablo 4.9: GA parametre değerleri.

Aday bireylerin sayısı	100
Kromozom uzunluğu	20
Çaprazlama (Crossover) Oranı	.5
Değişim (Mutasyon) Oranı	.2
Yineleme (İterasyon)	100

**Şekil 4.11:** Ağ kullanım ömrü süresince enerji tüketim oranı.



Şekil 4.12: İki yöntemle canlı devrelerin karşılaştırılması.

4.2.4. Kablosuz Algılayıcı Ağlarda Verilerin Birleştirilmesine İlişkin Sonuçlar

4.2.4.1. Uygunluk Fonksiyonu Seçilmesi

Bu aşamada önerilen uygunluk fonksiyonunun geliştirilmesi için daha sonra geliştirilmiş olan ve Nakamura formülü olarak da bilinen bir formül kullanılmıştır. Formülde (Denklem 4.11), ‘A’ verilen izleme alanını, ‘S’ algılayıcı devresi setini, ‘D’ talep edilen noktalar grubunu, ‘A^d’ ilgili alanın izlenmesinden sorumlu algılayıcıların setini, ‘AL’ düşük enerji modunda enerjinin açılmasını, ‘AH’ yüksek enerji modunda enerjinin açılmasını (paket yükünün ‘BS’ye gönderilmesi), ‘EC’ 3 farklı konumdaki (A, B, C) devrelerin maliyetini, ‘S’ ise eşik toplama setini göstermektedir. (bkz.4.2.2.1.1)

$$G(i) = \sum_{i \in S} (AE_i + EC_i) \quad \forall i \in D \& \forall ii \in A^d, AE \in \{AL, AH\} \quad (4.11)$$

Uygunluk fonksiyonu ile tüm kromozomların puanlanması ve karşılaştırma yapılması mümkündür. Denklem (4.11) kullanılarak tüm olası yollar ve devre konumları belirlenebilir. Bu işlem devrenin düşük/yüksek enerji moduna ayarlanabilmesi için

gereklidir. Denklem (4.12)'de önerilen uygunluk fonksiyonu, en az 3 konum kullanılarak (neredeyse kesin uygunluk için) geliştirilmiştir.

$$EC = \frac{4n_A + 2n_B + n_C}{\sum_{i \in S} n_i} \quad (4.12)$$

$$M \in \{A, B, C\}$$

Burada, EC'nin sayısal değerini algılayıcı ağının modu etkiler. Uzun iletişim mesafesi nedeniyle A modunda çalışan bir algılayıcı devresinin daha yüksek enerji tüketmesi beklenir. Diğer yandan, daha kısa iletişim mesafesine sahip olan B ve C modları enerji tüketimi sıralamasında bir sonraki sırada yer almaktadır. A modunda enerji tüketiminin C modundan dört, B modundan ise 2 kat daha fazla olduğu kabul edilir. [48]

KAA'da veri paketlerinin gönderimi için gerekli enerji ve enerjiye bağlı olarak seçilen uygunluk fonksiyonu Denklem (4.13)'de hesaplanmıştır.

$$E(i)_{Total} = E_{Monitor} + G(i) \quad (4.13)$$

$$F(i) = \frac{E(i)_{Total}}{N}$$

Burada, N devre sayısını ifade eder. F(i) fonksiyonu kullanılarak kurulum enerjisine bağlı olarak elektrik enerjisi hesaplanır. .

4.2.4.2. Değerlendirme ve Simülasyon Sonuçları

Algoritmanın uygulanması için bir ağ simülatörü kullanılmıştır. Simülasyonun iki aşaması bulunmaktadır. İlk aşamada, genetik algoritma tabanlı bölümü uygulanması için Java editor (Java Genetik Algoritma paketi (JPAC)) kullanılmıştır. İkinci aşamada ise, simüle edilen ortamdaki algılayıcı devresi ve BS arasındaki farklı yolların izlenmesi için OMNET++ kullanılmıştır. Elde edilen algılayıcı ağı ile ilgili parametreler Tablo 4.10'de listelenmektedir.

Tablo 4.10: Simülasyon parametreleri.

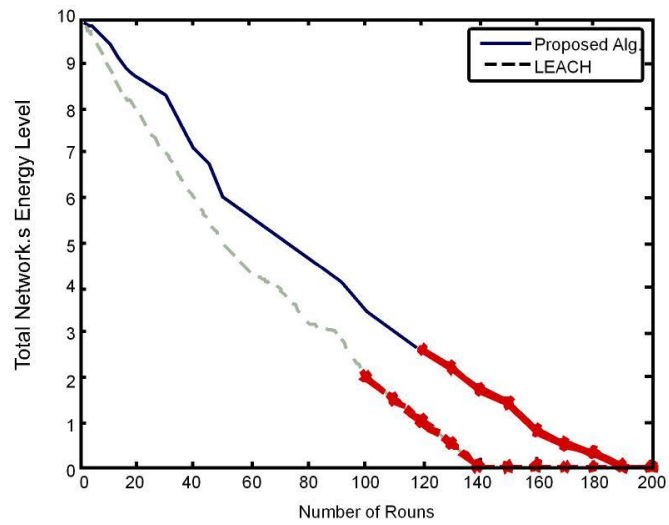
Ağ boyutu	100 m
Devre Sayısı	200
Başlangıç enerjisi	0.8 J
BS konumu	Kaynak merkezi
Ağ alanı	100*100 m ²
Simüle edilen senaryo	Ortalamanın 5 katı olduğu bildirilmiştir.
Ağaç Kullanım Süresi	15 periyot
Devrelerin yerleşimi	Rastgele

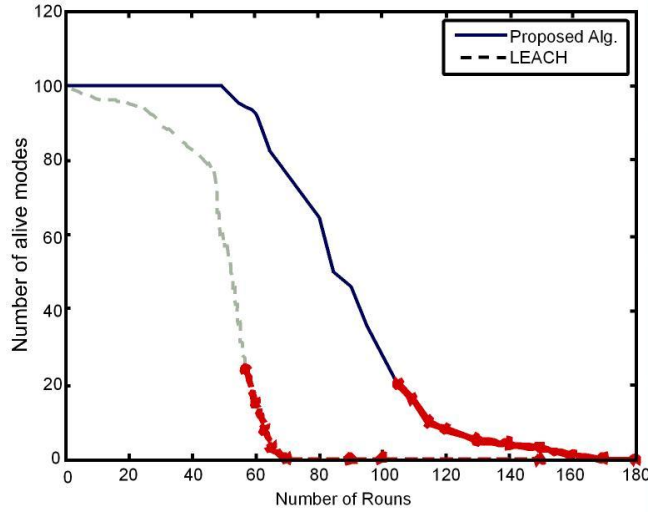
Tablo 4.10'da görüldüğü gibi, simülasyonda kullanılan ağaçların her biri 15 periyot boyunca kullanılmış ve simülasyon her senaryo için 5 defa tekrar edilmiştir.

Tablo 4.11: GA parametre değerleri.

Aday bireylerin sayısı	500
Kromozom uzunluğu	20
Çaprazlama (Crossover) Oranı	.7
Değişim (Mutasyon) Oranı	.7
Yineleme (İterasyon)	200

Tablo 4.11'de, ortam simülasyonu için kullanılan parametreler verilmiştir. Simülasyonda, kalan enerji ve paket boyutu gibi öncelikler dikkate alınarak bulunan tüm yollar içinde GA optimum olanını seçmiştir.

**Şekil 4.13:** Sanal ortamların ömrü boyunca görülen enerji tüketim oranları.



Şekil 4.14: Her iki yöntemle kapsama karşılaştırması.

Şekil 4.13 optimum BS yolunun belirlenmesi için gerekli olan birleştirilmiş paket yükü ve enerji yükünü göstermektedir. Şekil 4.14'da, ilk devrenin sistemden çıkarıldığı ilk zaman gösterilmektedir. Önerilen yöntem ile LEACH protokolü karşılaştırıldığında, önerilen yöntemde ilk devrenin ölüm zamanının önemli ölçüde geciktiği görülmüştür. Buna ek olarak, minimum aktif devre sayısı ile ağ çalışmaya devam etmektedir. Son bireyler, yolu neredeyse tek tip bir enerji tüketimi ile sağlamaktadır. Bu özellik ise ağı kullanım ömrüne son derece önemli bir katkıda bulunmaktadır. [48]

4.2.5. MOBİL AĞLAR İÇİN GENETİK YAKLAŞIM SONUÇLARI

4.2.5.1. Uygunluk Fonksiyonu Seçimi

Uygunluk fonksiyonu seçimi, kapsama alanına ve kapsama+izleme amaçlı harcanan enerji miktarına bağlıdır. Kapsama ve enerji kriterlerine dayalı olarak önerilen uygunluk fonksiyonu Denklem (4.14)'de verilmiştir.

$$Fitness(x) = \sum_{j=1}^n [(coverage_{x,j,e} - coverage_{x,j,e-1}) / coverage_{CH \times P} - (energy_{x,j,e} - energy_{x,j,e-1}) / energy_{CH \times P}] \quad (4.14)$$

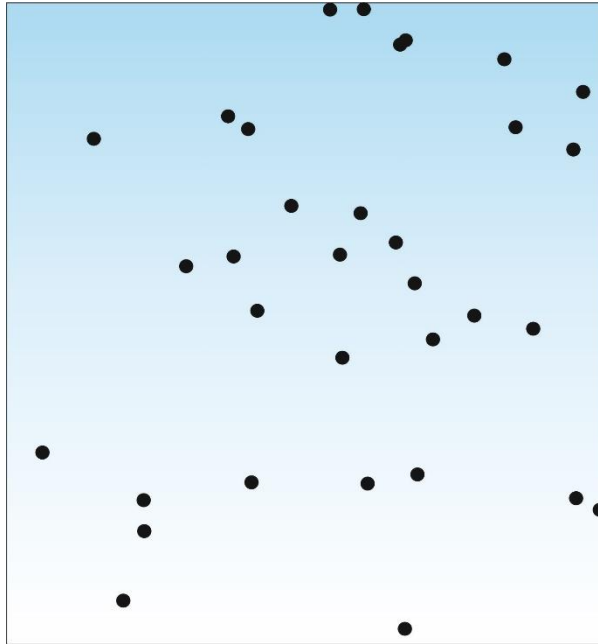
Burada Kapsama (coverage) kat edilen alanı ifade ederken, enerji (energy) düğümlerdeki kalan güç miktarını, p ise toplam veri paketi sayısını, n veri paketlerinin aktarımında yer alan aktif düğümlerin sayısını, e ise adım sayısını göstermektedir. Bu fonksiyon bütün koşulları göz önüne almak amacıyla formüle edilmiş olup kromozomlar arasında kapsamlı bir karşılaştırmaya olanak tanımaktadır.

4.2.5.2. Değerlendirme ve Simülasyon Sonuçları

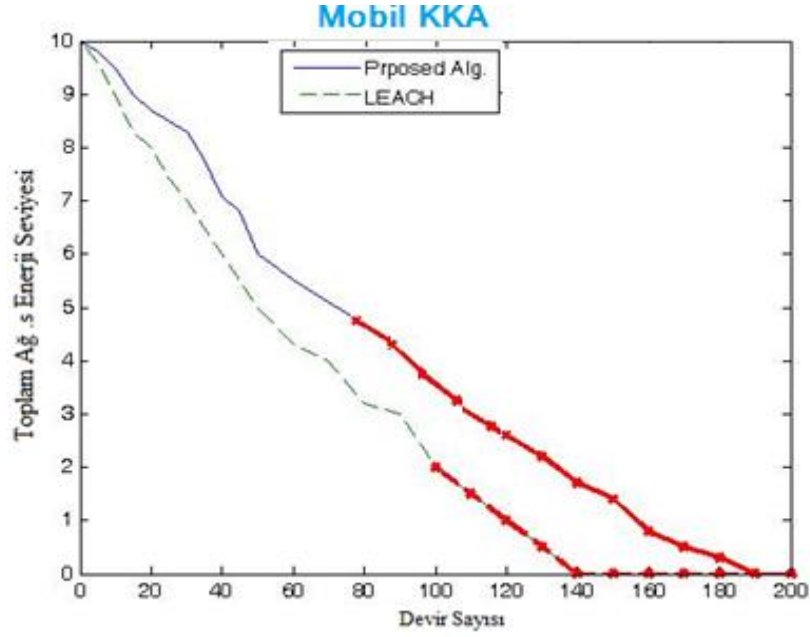
Hedeflenen algoritma ağ benzeştiricisi yardımıyla uygulanmıştır. Algoritmanın testi için Java Genetik Algoritma Paketi (JPAC) kullanılmıştır. Şekil 4.15’de gösterildiği gibi sanal ortamda düğümlerin hareketini takip etmek için OMNET++ kullanılmıştır. Test edilen Sanal Ortam Parametreleri Tablo 4.12’de verilmiştir.

Tablo 4.12: Test edilen sanal ortam parametreleri

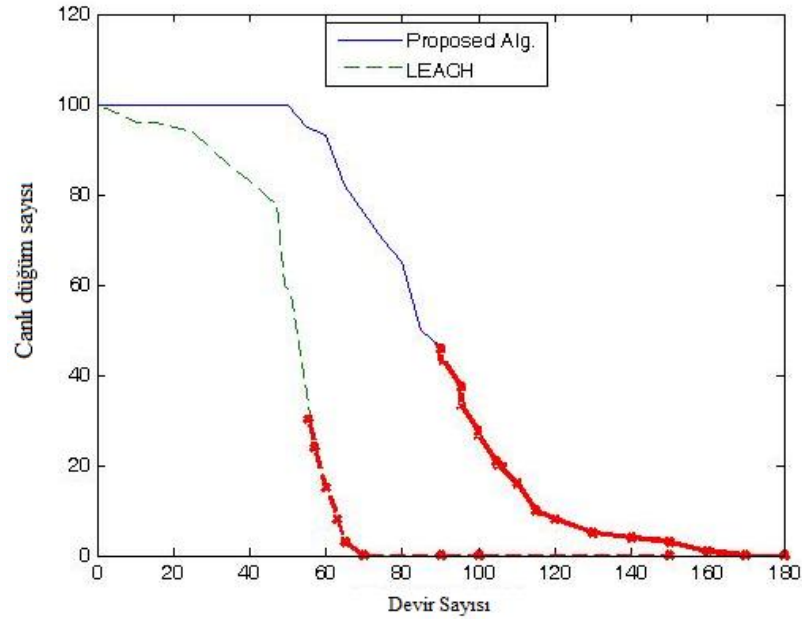
Ağ boyutu	100 m × 100 m
Düğüm numaraları = n	200
Başlangıç enerjisi	2 J
Düğüm enerjisi	50 nJ/bit
Kapsanmayan alan	$10 \times 10 \text{ m}^2$
BS mesafesi	200 m
Paket boyutu	200 bits



Şekil 4.15: Örnek sanal ortam.



Şekil 4.16: Toplam Enerji sanal ortamın ömrüne oranı.



Şekil 4.17: Önerilen ve LEACH algoritmaları arasında kapsam karşılaştırması.

Şekil 4.16'da önerilen algoritma ve LEACH protokolü kullanan algoritma arasında 200 zaman birimi (yılı) üzerinden ağ enerjisi ve ömrüne ilişkin yapılan karşılaştırmalar gösterilmiştir. LEACH protokolüne göre, Küme Başı düğümlerin birleştirilmiş enerji tüketiminin kısa ömürlü olduğu ve düşük enerji durumu sebebiyle ilk düğümün kaldırma süresinin (ilk düğümün devre dışı kalması yüzünden) daha gecikmeli olduğu tespit edilmiştir. Ayrıca, asgari sayıda düğüm çalışır olduğu sürece ağ da çalışır

vaziyette kalmaktadır. Genel olarak, düğümlerin enerji durumunu ve Küme Başı düğümler ile Baz İstasyonu arasındaki mesafeyi göz önünde bulunduran uygunluk fonksiyonu algoritmasının kullanımından dolayı, kalan tekiller ağın ömrünü önemli derecede artıran bir veri birikimi ile birleşik enerji sağlamaktadır. Şekil 4.17’de, LEACH ile önerilen algoritma arasındaki kapsama karşılaştırması geçen süre baz alınarak gösterilmektedir. Ayrıca algoritmamızın ortaya çıkardığı topoloji 60. devire kadar bütün alanı kapsarken, 180. devire kadar kapsama alanı yavaşça azaldığı görülmüştür.

Bu bölümde, mobil ağlar için interaktif GA tabanlı bir yaklaşım sunulmuştur. Ayrıca, kalan güç miktarını göz önünde bulundurarak uygun konumları belirleyen hareket algılayıcı düğümlerini yönlendirmek için online yaklaşımdan faydalanılmıştır. Son olarak, bulunan çözüm eş zamanlı olarak çevresel kapsamayı en üst seviyeye çıkarttığı ve güç tüketim ölçütlerini en aza indirdiği görülmüştür. Simülasyon sonuçları önerilen uygunluk fonksiyonunun bu amaçları başardığını doğrulamaktadır. Öte yandan, LEACH protokolü ile karşılaştırıldığında kapsamının geliştirildiği ve ağ kullanım ömrünün de uzatıldığı görülmüştür.

4.2.6. Uygulama Tabanlı Yönlendirme

Yaklaşımımız hedef düğümün "Ağ İşlevselliği", "Zaman Süresi", "Sekme Sayısı" ve "Kalan Enerji" kriterlerinin karşılanması bakımından akıllı bir şekilde yönlendirildiği optimal bir yol bulmak amaçlı önerilmiştir. Bu kriterleri karşılayan yol, yönlendirme çizelgesinde kaydedilir veya güncellenir.

Düşük "Sekme sayısı" kriterinde kısa yol veya süre belirtisi olmadığından, "Zaman süresi" kriteri ile birlikte değerlendirilmelidir. Düşük Sekme sayılarına karşın, düğümler arasında büyük mesafeyle karşılaşacağımız bir durum olabilir. Bu durum sonucu hedef düğümüne paket yönlendirmesi için ortalama sürede artış meydana gelebilir. Hedef düğümüne ilişkin yol bulma işleminde çeşitli kriterlerin değerlendirilmesinin sebebi, farklı koşullarda ve ağ işlevselliklerinde esneklikte artış sağlamaktır. Önerilen protokolle yönlendirme hesaplamasını minimize etme ve ayrı düğümlerle bağımsız hesaplama gibi ilave amaçları da gerçekleştirmeyi hedefliyoruz. Bu özellikler maliyet etkin enerji için ortalama mutabakatın da azalmasına neden olur.

4.2.6.1 Uygulama Tabanlı Sensör Ağı İşlevsellikleri

Genel olarak, sensör ağlarının Ticari, Askeriye ve Özel(Standart) olmak üzere üç uygulama alanı vardır. Bu uygulama alanları için, yukarıda bahsedilen kriterlerin önceliğinin ve etkinliğinin aşağıdaki gibi değerlendirildiği düşünelim:

Ticari: (1) Maliyet, (2) Enerji, (3) Sekme sayısı ve (4) Zaman süresi

Askeriye: (1) Enerji, (2) Zaman süresi (3) Sekme sayısı ve (4) Maliyet

Özel(Standart): (1) Sekme sayısı, (2) Enerji, (3) Maliyet ve (4) Zaman süresi

Bu çalışmada önerilen protokol, her bir uygulamada, akıllı bir şekilde öncelikli başlatma sağlar ve ardından paket yönlendirmesi için en iyi kriterlerin kombinasyonunu sunar. Genetik Algoritma kullanılarak her bir düğüm için komşu bir düğüm seçilir.

Denklem (4.15)'de sözkonusu kombinasyon verilmiştir.

$$\begin{aligned}
 P(\text{rout})= & (Custom) \left(\frac{\sum_{i=1}^n \exp(e_i)}{\text{pow}(h) * \sum_{j=1}^k c_j} + \frac{1}{\sum_{j=1}^k t_j} \right) + \\
 & (Enterprise) \left(\frac{\sum_{i=1}^n \exp(e_i)}{\sum_{j=1}^k \text{pow}(c_j) * h} + \frac{1}{\sum_{j=1}^n t_j} \right) + \\
 & (Military) \left(\frac{\sum_{i=1}^n \text{pow}(e_i)}{\sum_{j=1}^k \exp(t_j) * h} + \frac{1}{\sum_{j=1}^k c_j} \right)
 \end{aligned}$$

(4.15)

Burada; pow ve exp sırasıyla (4.16) ve (4.17)'de tanımlanmıştır.

$$\exp(x) = e^x = 2.71^x$$

(4.16)

$$\text{Pow}(x) = x^x$$

(4.17)

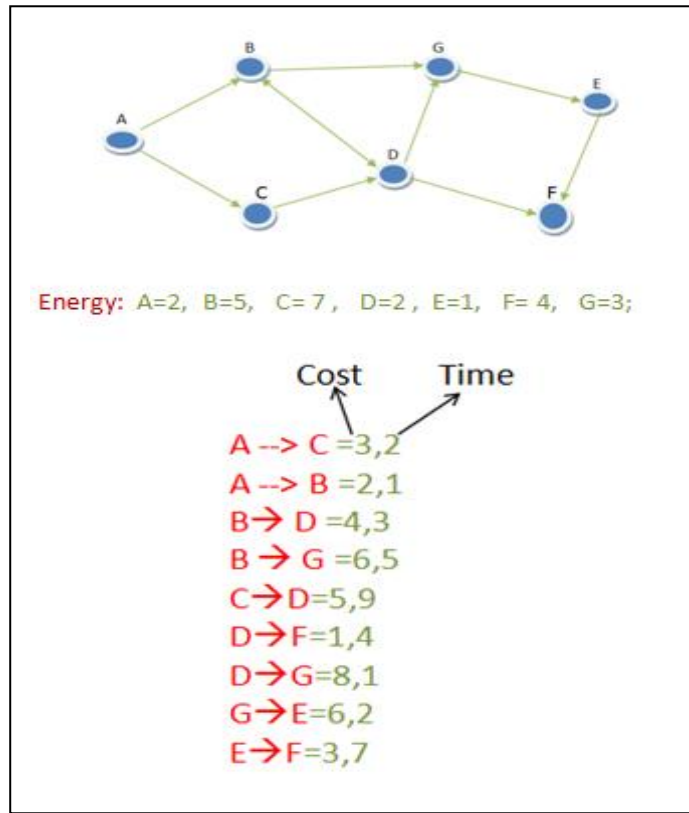
Denklem (4.15) yukarıda bahsedilen dört kriteri ve uygulamaları göz önünde bulundurarak ağ düğümlerinin yol seçim faktörünü belirleyen genel bir ilişki tanımlamaktadır. Herhangi bir düğüm vasıtasıyla herhangi bir paketi yönlendirmek istediğimizde Denklem (4.15)'deki fonksiyon çağrılır. Herhangi bir düğümdeki ilişki çağrıldığında düğümün komşu seçimini gösteren nümerik bir değer geri gelir. Askeri uygulamada örneğin ilk Özel ve Ticari değer 0, ilk Askeri değer ise 1 olarak ayarlanır ve ilişkinin Özel ve Ticari kısımları otomatik olarak devre dışı kalır. Burada k ve N sırasıyla her bir yolun kenarlarını ve düğümlerini göstermektedir. Bu nedenle Maliyet

ve Zaman süresi kriterleri kenardan kenara değişir; kalan enerji her bir düğüme göre değişir ve sekme sayısı da her bir yola göre farklılık gösterir.

Fonksiyonların önceliği aşağıdaki gibi düşünülmüştür:

(1) $pow()$, (2) $exp()$, (3) Σ .

Denklem (4.15-4.17) formülleri kullanılarak, genetik algoritma yaklaşımına dayalı özgün bir simülasyon oluşturulmuştur. Şekil 4.18 simülasyonda kullanılan örnek parametreleri, Şekil 4.19 ise bu parametreleri kullanarak bulunan tüm rotaların hesaplanan tercih değerlerini göstermektedir.



Şekil 4.18: Simülasyonda kullanılan örnek parametreler.

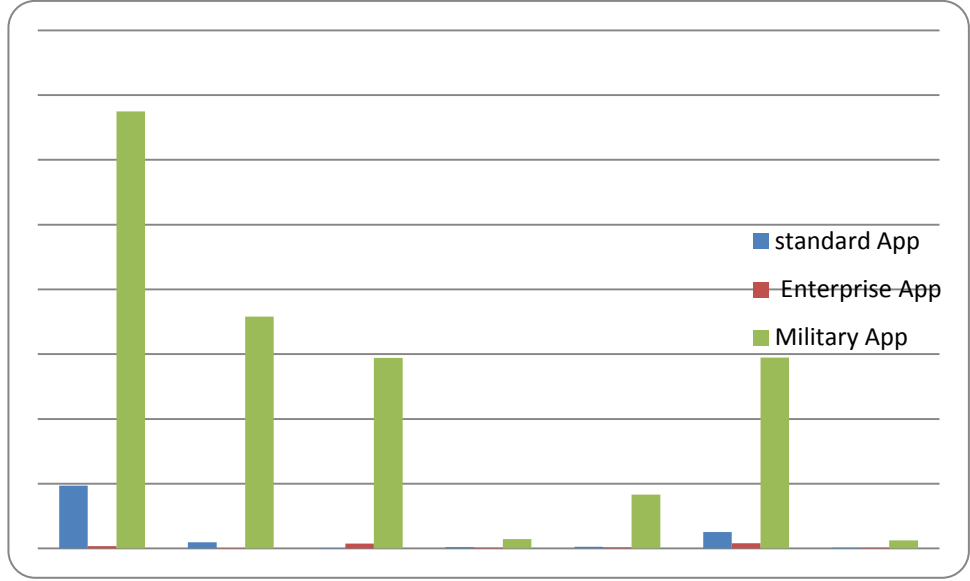
P1: A→C→D→F Custom=4.86506 Enterprise =0.189937 Military =33.7425	P3: A→C→D→B→G→E→F Custom =0.0367758 Enterprise =0.0380179 Military =14.726
P2: A→C→D→G→E→F Custom =0.0628359 Enterprise =0.0476332 Military =17.916	P4: A→B→G→E→F Custom =0.120252 Enterprise =0.0672913 Military =0.738621
P5: A→B→G→D→F Custom =0.145568 Enterprise =0.0909126 Military =4.15577	P6: A→B→D→F Custom =1.27732 Enterprise =0.403147 Military =14.7377
P7: A→B→D→G→E→F Custom =0.074776 Enterprise =0.0714314 Military =0.648501	

Şekil 4.19: Bulunan tüm rotaların hesaplanan tercih değerleri.

Şekil 4.20 rotaların uygulamalardaki tercih değerlerini ve Şekil 4.21 ise her bir uygulama için önerilen rota tercih değerlerini göstermektedir.

Path	standard App	Enterprise App	Military App	HopCount
1	✓ 4,86	⚠ 0,18	✓ 33,74	3
2	✗ 0,47	✗ 0,04	⚠ 17,9	5
3	✗ 0,03	✓ 0,38	⚠ 14,72	6
4	✗ 0,12	✗ 0,06	✗ 0,73	4
5	✗ 0,14	✗ 0,09	✗ 4,15	4
6	✗ 1,27	✓ 0,4	⚠ 14,73	3
7	✗ 0,07	✗ 0,07	✗ 0,64	5

Şekil 4.20: Rotaların uygulamalardaki tercih değerleri.



Şekil 4.21: Her bir uygulama için önerilen rota tercih değerleri.

5. TARTIŞMA VE SONUÇ

Kablosuz Algılayıcı Ağlar uygulamalara özeldir, yani tasarımları uygulamaya bağlı olarak değişir. Bu ağlarda; Ölçeklenebilirlik, Gecikme Süresi, Yayılma, Bilinçlendirme, Destek İhtiyacı, Veri Toplama, Optimum Yol, Ek Yük Maliyetleri gibi temel ölçülebilirler incelenmektedir. Bu ölçülebilirler olan ihtiyaç uygulamalara göre değişmekle birlikte, ağın dayanıklılığının tahminini de güçleştirmektedir. Ağ performansını etkileyebilecek parametreler komşu düğüm ve havuz ile olan mesafe, her bir düğüm ile verici düğüme olan mesafe, komşu düğümün enerjisi, komşu düğümün tampon kapasitesi, gecikme süresi, başarılı kaynak sayısı, iletimdeki gecikme, atlama sayısı, yol seçiminde başarı veya başarısızlık, iletim-güvenilirliğinin riski ve olasılıkları gibi 10 ana parametre incelenmiş ve optimize edilmiştir. Değişik mesafe tahmin etme metotları gözden geçirilmiştir, her birisi belirli durumlar için en iyi uygulama olurken hiçbirisi her durumda mükemmel olarak nitelendirilmemiştir. Mesafeyi anlamak için GPS metodu çok önerilmemektedir, çünkü bu çok maliyetlidir. Olasılık tahmini yöntemleri özel uygulamalar için uygundur. Hatalı sonuç verebilme riski nedeniyle, bu yöntem yanıltıcı olabilir; bu nedenle, birçok uygulama için tavsiye edilmez. Daha önceden belirtildiği gibi, sinyal gücü yöntemi mesafenin hesaplanması için önerilir.

Enerji tüketim modelleri etkileyen parametreler de önemli rol almaktadır. En son teknolojik ve elektronik gelişmeler enerji tüketimini hesaplamayı kolaylaştırmış, dolayısıyla her bir algılayıcı kalan enerji yüzdesini yazılım kullanarak rahatlıkla verebilmektedir. Geçen bölümde gösterildiği gibi klasik enerji modeli düşük operasyonel güce ihtiyaç duymakta, veri alma ve vermede daha az enerji kullanmaktadır. Ancak klasik enerji modelinin bekleme modunda hiç enerji kullanmadığı görülmektedir. Yine de, kaynak sayısının artması olay sayısı da artınca raporlamada başarısızlıklara yol açmaktadır. Daha az atlama olan yollar mesajları daha hızlı bir şekilde iletmektedir. Daha hızlı iletim, daha kısa uçtan-uca gecikme anlamına gelmektedir. Kısa bir yol seçmenin başka bir avantajı daha az düğümün yer almasıdır ve daha çok düğümün beklemede kalmasıdır. Tamponların yüksek fiyatları göz önünde bulundurulduğu zaman, tampon kapasite eklenmesi daha pahalı bir ağ anlamına

gelmektedir. Ancak büyük tamponlar paket kaybını azaltmakta ve eşit enerji tüketimini geliştirmektedir.

Bilindiği gibi, özel tasarım algılayıcılar kolaylıkla ağlara yerleştirilebilirler. Ancak, KAA sorunsuz değildir, güç, hafıza ve bant genişliği gibi limitleri vardır. Bu dezavantajlar verinin toplanmasını ve işlenmesini ciddi bir durum haline getirmektedir. Bu temel özellikler iletişim protokolleri için ek zorluklar teşkil etmektedir. Bu nedenle, , tezde incelenen yönlendirme protokollerinin güvenli enerji tüketimi ve enerji optimizasyonu üzerindeki etki faktörleri araştırılmıştır. Ayrıca, Gossiping protokolü üzerinde sonraki sekmenin seçiminde bazı değişiklikler yapılırsa, ağ ömrünün artırılacağı tespit edilmiştir. Ayrıca bu nedenle Gossiping Protokolünün dezavantajlarını iyileştirmek amacıyla, yeni protokol önerilmiştir (FELGOSSIPING).

Sonuç olarak, KAA'lerle ilgili en önemli kısıtlamalar enerji tüketimi ve ağ ömrü olduğu görülmüştür. Kısıtlamalar KAA uygulamaları için de birer sorun teşkil etmektedir. Genel olarak bakıldığında, KAA'lerin işlem aşamaları şunlardır: Devre yerleştirme, ağ kapsama, kümeleme, veri birleştirme ve yönlendirme. Bu işlem aşamaları üzerinde teknik bir araştırma yapılmıştır. Araştırmada mevcut sorunlar tespit edilerek iyileştirilmiştir ve böylece de ideal ağ parametrelerine ulaşılmıştır.

Kablosuz Algılayıcı Ağları optimize etmek için kullanılan bulanık mantık, yapay sinir ağları ve evrimsel algoritmalar gibi çeşitli yöntemler mevcuttur. Ancak, en yaygın yöntem, büyük ölçekli ağlarda daha iyi sonuçlar vermesi ve son aşamada nihai bir formül oluşturması bakımından çok daha kullanışlı ve faydalı bir yöntem olan genetik algoritmadır. GA'nın stokastik yapısı nedeniyle farklı algoritma çalışmaları ile farklı çözümler kullanılarak farklı performanslar elde edilmesi mümkündür. KAA simülatörü ile uygulanan önerilen algoritma da izlenen alanın neredeyse %100'ünde kapsama sağlanmıştır. Tezde önerilen yöntemin etkinliğini göstermek için, zaman içerisinde canlı kalan paketlerin yanı sıra, enerji tüketimi seviyesi ve aktif devrelerin sayısı verilmiştir.

Çalışmanın sonunda, Optimum Baz istasyonu yolunun belirlenmesi için gerekli olan birleştirilmiş paket yükü ve enerji yükü değişimleri gösterilmiştir. Açıkça görüldüğü gibi, LEACH protokolü ile yapılan kıyaslamada ilk devrenin ölüm zamanı önemli ölçüde gecikmiştir. Buna ek olarak, minimum aktif devre sayısı ile de ağ çalışmaya devam etmektedir. Genel olarak, yönlendirme algoritması kullanıldığında, uygunluk fonksiyonu ile üç enerji statüsü modu (düşük, orta, yüksek) ya da BS için iletişim mesafesi alınır. Bu yaklaşımda, komşuluğunda belirli algılayıcılar bulunan ve de aynı

alıřma moduna sahip olan kmelerde optimum enerji tketimi saėlanabilmesi iin, bir kmeleme zm oluřturulmuřtur. Tm kmeler baz istasyonu (BS) ya da alıcı ile iletiřim kurmak iin ok-sekmeli iletim kullanabilir. İstenen alanlarda izleme ve lm yapma ile ilgili kullanılabilir ok amalı algoritma verilmiřtir.

KAYNAKLAR

- [1] Akyildiz, I.F., Su, W., Sankarasubramaniam Y., E. Cayirci., 2002, A Survey on Sensor Networks, *IEEE Communications Magazine*, 40 (8), 20-28.
- [2] Polastre, J., 2003, *Sensor Network Media Access Design*, Technical Report, Computer Science Division, Regents of the University of California.
- [3] Shih, E., Cho, S., Ickes, N., Min, R., Sinha, A., Wang, A., Chandrakasan, A., 2001, Physical layer driven protocol and algorithm design for energy-efficient wireless sensor networks, *Proceedings of the 7th annual international conference on Mobile computing and networking*, 16-21 July 2001 Rome, Italy, ACM, 272-282.
- [4] Heinzelman, W.B., Chandrakasan, A.P., Balakrishnan, H., 2002, An Application-Specific Protocol Architecture for Wireless Microsensor Networks, *IEEE Transactions on Wireless Networking*, 1 (4), 660–670.
- [5] Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H., 2000, Energy-Efficient Communication Protocol for Wireless Microsensor Networks, *33rd Annual Hawaii International Conference on Systems Sciences*, 4-7 January 2000 Island of Maui, Hawaii, USA, IEEE, 256-270 .
- [6] Shnayder, V., Hempstead, M., Chen, B., Werner A.G., Welsh, M., 2004, Simulating the Power Consumption of Large Scale Sensor Network Applications, *2nd international conference on Embedded networked sensor systems*, 3-5 November 2004 Baltimore, Maryland, USA, ACM, 239-248.
- [7] Anastasi, G., Conti, M., Falchi, A., Gregori, E., Passarella, A., 2004, Performance Measurements of Mote Sensor Networks, *International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile System*, 4-6 October 2004 Venezia, Italy, ACM/IEEE, 78-89.
- [8] Crossbow, M., 2004, *XBOW Smart Dust Product Information Guide*, http://www.xbow.com/Support/Support_pdf_files/XBOW_Smart_Dust_ProductInfoGuide.pdf , [Ziyaret Tarihi: 08-07-2015].
- [9] Polastre, J., Hill, J., Culler, D., 2004, Versatile Low Power Media Access for Wireless Sensor Networks, *2nd international conference on Embedded networked sensor systems*, 3-5 November 2004 Baltimore, Maryland, USA, ACM, 59-64.
- [10] Rawat, U., Sharma, M., 2012, Directed Diffusion: Features, Current Developments, Issues and Analysis, *International Journal of Computer Applications*, 49 (12), 25-36
- [11] Zheng, J., Jamalipour, A., 2009, *Wireless Sensor Networks: A Networking Perspective*, John Wiley & Sons, Hoboken, 260-278.

- [12] Fan, X.N., Song, Y.L., 2007, Improvement on LEACH Protocol of Wireless Sensor Network, *International Conference on Sensor Technologies and Application*, 14-20 October 2007 Valencia, Spain, IEEE, 150-164.
- [13] Intanagonwiwat, C., Govindan, R., Estrin, D., 2000, Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks, *6th annual international conference on Mobile computing and networking*, 6-11 August 2000 Boston, Massachusetts, USA, ACM, 56-67.
- [14] Lee, Y.H., Lee, K.O., Lee, H.J., Kusdaryono, A., 2010, CBERP: Cluster Based Energy Efficient Routing Protocol for Wireless Sensor Network, *12th International Conference on Networking, VLSI and Signal Processing*, 20-22 February 2010 Cambridge, UK, WSEAS, 24-36.
- [15] Akyildiz, I., Canvuran, M., 2010, *Wireless Sensor Networks*, Wiley, London, UK, 0-516.
- [16] Yen, W., Chen, C.W., Yang, C.H., 2008, Single Gossiping with Directional Flooding Routing Protocol in Wireless Sensor Networks, *3rd Conference on Industrial Electronics and Applications*, 3-5 June 2008 Singapore, IEEE, 1604-1609.
- [17] Oh, H., Chae, K., 2007, An Energy-Efficient Sensor Routing with Low Latency, Scalability in Wireless Sensor Networks, *International Conference on Multimedia and Ubiquitous Engineering*, 26-28 April 2007 Seoul, South Korea, IEEE, 147-152.
- [18] Lindsey, S., Sivalingam, K.M., 2002, Data Gathering Algorithms in Sensor Networks Using Energy Metrics, *IEEE Transactions on Parallel and Distributed Systems*, 13 (9), 924-934.
- [19] Holger, K., Willig A., 2005, *Protocols and Architectures for Wireless Sensor Networks*, John Wiley & Sons, Ltd., Chichester, UK, ISBN: 0-470-09510-5, 360-401
- [20] Sohraby, K., Minoli, D., Znati, T., 2007, *Wireless Sensor Networks, Technology, Protocols, and Applications*, John Wiley & Sons, Inc., Hoboken, New Jersey, USA, 170-202.
- [21] Norouzi, A., Dabbaghian, M., Hatamizadeh, A., Üstündağ, B.B., 2010, An Improved ELGossiping Data Distribution Technique with Emphasis on Reliability and Resource Constraints in Wireless Sensor Network, *International Conference on Electronic Computer Technology*, 7-10 May 2010 Kuala Lumpur, Malesia, IEEE, 179-183.
- [22] Norouzi, A., Babamir, F., Zaim, A., 2011, A Novel Energy Efficient Routing Protocol in Wireless Sensor Networks, *Wireless Sensor Network*, 3 (10), 341-350.
- [23] Khodashahi, M.H., Norouzi, A., Amiri, F., Dabbaghian, M., 2010, A Novel Optimal Routing Algorithm by Creating Concentrically Sectors in Wireless Sensor Networks, *8th Annual Communication Networks and Services Research Conference*, 11-14 May 2010 Montreal, France, IEEE, 168-173.
- [24] Akyildiz, I.F., Vuran, M.C., 2010, *Wireless Sensor Networks*, John Wiley & Sons, Ltd, Chichester, UK.

- [25] Akkaya, K., Younis, M., 2005, A Survey of Routing Protocols in Wireless Sensor Networks, *Elsevier Ad Hoc Network Journal*, 3 (3), 315-339.
- [26] Romoozi, M., Vahidipour, M., Romoozi, M., Maghsoodi, S., 2010, Genetic Algorithm For Energy Efficient And Coverage-Preserved Positioning in Wireless Sensor Networks, *International Conference on Intelligent Computing and Cognitive*, 22-23 June 2010 Kuala Lumpur, Malesia, IEEE, 22-25.
- [27] Norouzi, A., Zaim, A.H., 2013, Genetic Algorithm Application in Optimization of Wireless Sensor Networks, *The Scientific World Journal*, Hindawi Publisher, 0 (0), 110-136.
- [28] Norouzi, A., Babamir, F.S., Zaim, A.H., 2013, An Interactive Genetic Algorithm for Mobile Sensor Networks, *Studies in Informatics and Control*, 22 (2), 213-218.
- [29] Goldberg, D.E., 1989, *Genetic Algorithm in a Search Optimization and Machine Learning*, Addison Wesley, USA, 248-290.
- [30] Kreinovich, V., Quintana, C., Fuentes, O., 1933, Genetic algorithms: what fitness scaling is optimal?, *Cybernetics and Systems: an International Journal*, 24, 9-26.
- [31] Hussain, S., Matin, A., Islam, O., 2007, Genetic Algorithm for Hierarchical Wireless Sensor Network, *Journal of Networks*, 2 (5), 87-97.
- [32] Bhondekar, A.P., Vig, R., Singla, M.L., Ghanshyam, C., Kapur, P., 2009, Genetic algorithm Based Node Placement Methodology For Wireless Sensor Networks, *International MultiConference of Engineering and Computer Science*, 18-20 March 2009 Hong Kong, China, IAENG, 106-112.
- [33] Trigoni, N., Yao, Y., Demers, A., Gehrke, J., Rajaramany, R., 2004, Wave Scheduling: Energy-Efficient Data Dissemination For Sensor Networks, *International Workshop on Data Management for Sensor Networks (DMSN), in Conjunction With The International Confernece on Very Large Data Bases (VLDB)*, 30 August 2004 Toronto, Canada, ACM, 70-95.
- [34] Quint, F. P., Nakamural, F., Mateusl, G.R., 2005, A Hybrid Approach To Solve The Coverage and Connectivity Problem in Wireless Sensor Networks, *Workshop on Meta-heuristics: Design and Evaluation of Advanced Hybrid Meta-heuristics*, 1 November 2004 Monticello, IL, USA, 60-71.
- [35] Orman, Z., Norouzi, A., Babamir, F.S., Intelligent Node Placement using GA (INPGA) Protocol in Wireless Sensor Networks, in *Proceedings of the International Conference on Wireless Networks (ICWN'11)* ISBN #: 1-60132-187-2 , 280-296.
- [36] Quintão, F.P., Nakamura, F.G., Mateus, G.R, 2004, A Hybrid Approach to Solve the Coverage and Connectivity Problem in Wireless Sensor Networks, *In 4th European Workshop on Meta-Heuristics: Design and Evaluation of Advanced Hybrid Meta-Heuristics*, Nottingham, United Kingdom, 45-56.
- [37] Nakamura, F.G., 2003, *Planejamento Dinamico para Controle de Cobertura e Conectividade em Redes de Sensores Sem Fio Planas*, Thesis (Ms), Universidade Federal de Minas Gerais.
- [38] Norouzi, A., Babamir, F., Zaim, A., 2011, A New Clustering Protocol for Wireless Sensor Networks Using Genetic Algorithm Approach, *Wireless Sensor Network*, 3 (11), 362-370.

- [39] Hussain, S., Islam, O., 2007, An Energy Efficient Spanning Tree Based Multi-Hop Routing in Wireless Sensor Network, *Wireless Communications and Networking Conference (WCNC)*, 11-15 March 2007 Honk Hong, China, IEEE, 310-324.
- [40] Tan, H.O., Körpeoğlu, I., 2003, Power Efficient Data Gathering And Aggregation in Wireless Sensor Networks, *ACM SIGMOD Record*, 32 (4), 66-71.
- [41] Islam, O., Hussain, S., 2007, Genetic algorithm for data aggregation trees in wireless sensor networks, *3rd International Conference on Intelligent Environments*, 24-25 September 2007 Germany, IET, 312-316.
- [42] Norouzi, A., Babamir, F., Orman, Z., 2012, A Tree Based Data Aggregation Scheme for Wireless Sensor Networks Using GA, *Wireless Sensor Network*, 4 (8), 191-196.
- [43] Pourkabirian, A. and Haghghat, A.T., 2007, *Energy-Aware, Delay-Constrained Routing In Wireless Sensor Networks Through Genetic Algorithm*, *Proceeding of SoftCOM*, Qazvin University, Iran, 1-5.
- [44] Suen, Y. ,2004, A Genetic Algorithm Based Mobile Sensor Network Deployment Algorithm, Technical report, Department of Electrical and Computer Engineering, The University of Texas at Austin, 300-316
- [45] Paz, E.C., 1998, A Survey Of Parallel Genetic Algorithm,,*Journal of Calculateurs Parallels, Resmeaux Et Systems Repartis*, 10, 20-26
- [46] Babamir, F.S., Hatamizade, A., Babamir, S. M., Dabbaghian, M., Norouzi, A., 2010, Application of Genetic Algorithm in Automatic Software Testing, *Second International Conference on Network Digital Technology*, 7-9 July 2010 Charles University, Prague, Czech Republic, DIRF, 545–552
- [47] Norouzi, A., Sertbas, A., 2011, An Integrated Survey In Efficient Energy Management for KAA using Architecture Approach, *International Journal of Advanced Networking and Applications*, 3 (1), 968–977
- [48] Qu, Y., Georgakopoulos, S.V., 2011, Relocation of Wireless Sensor Network Nodes Using A Genetic Algorithm, *12th Annual Wireless and Microwave Technology Conference*, 18-19 April 2011 Clearwater Beach, FL, USA, IEEE, 1-5.
- [49] Gage, D., 1992, Command control for many-robot systems, *Unmanned Systems Magazine*, 10 (4), 28–34
- [50] Howard, A., Mataric, M., Sukhatme, G., 2002, Mobile Sensor Network Deployment Using Potential Fields: A Distributed, Scalable Solution To The Area Coverage Problem, *International Symposium on Distributed Autonomous Robotics Systems*, 25-27 June 2002 Fukuoka, Japan, KROS, 287-295.
- [51] Kalpakis, K., Dasgupta, K., Namjoshi, P., 2003, Efficient algorithms for maximum lifetime data gathering and aggregation in wireless sensor networks, *Computer Networks*, 42 (6), 697–716.
- [52] Dasgupta, K ., Kalpakis, K., Namjoshi, P., 2003, An Efficient Clustering-Based Heuristic For Data Gathering And Aggregation in Sensor Networks, *IEEE Wireless Communications and Networking Conference*, 16-20 March 2003 Louisiana, USA, IEEE, 1948–1954.

EKLER

EK 1: OZEL SIMÜLATÖR YAZILIMI

Sınıflar

Bu kategori içerisinde üç sınıf yer almakta olup bunların her biri burada kısaca açıklanmıştır.

1. CNode

```
public CNode(String Line)
{
    NodeName = Line.Split('=')[0].Replace("\n", "").Replace("\r", "");
    String RightSide = Line.Substring(Line.IndexOf('=') + 1);
    String[] temp = RightSide.Split(new char[]{' ',''},StringSplitOptions
RemoveEmptyEntries );
    Energy = int.Parse(temp[0]);
    if (temp[1].Split('=')[0].ToUpper() == "X")
        X = int.Parse(temp[1].Split('=')[1]);
    else
        Y = int.Parse(temp[1].Split('=')[1]);
    if (temp[2].Split('=')[0].ToUpper() == "X")
        X = int.Parse(temp[2].Split('=')[1]);
    else
        Y = int.Parse(temp[2].Split('=')[1]);
}
```

Bu kategoride açık bir şekilde üç yöntem göze çarpmaktadır. CNode Sınıfı, Nod Adı, X ve Y (konum) adında üç değişkendir her bir nodun özellikleri ile her bir nod için giriş değerlerini baz alan Enerji (nod enerjisi) saklar. (Simülasyon Formu'nda kullanılan yöntemle açılan dosyadan her bir nod için bilgiler okunur ve bu nodlara ayrı ayrı değişkenler atanır).

Yöntem kurucu yöntemi olup, ilgili değişkenlerde her bir noda ait bilgileri kaydeder. (Simülasyon Formu'nda kullanılan yöntemle açılan dosyadan her bir nod için bilgiler okunur ve bu nodlara ayrı ayrı değişkenler atanır).

```
public CNode(String Name, int X, int Y, int Energy)
{
    NodeName = Name;
    this.X = X;
    this.Y = Y;
    this.Energy = Energy;
}
```

Yöntem kurucu yöntemi olup her bir noda ait giriş parametrelerini kaydeder. (Simülasyon Formu'nda kullanılan yöntem geçerli değişkenlerde her bir noda ait bilgileri yazar).

```
public string ToCode()
{
    return String.Format("{0}={1},X={2},Y={3}", NodeName,Energy, X, Y);
}
```

Her bir nod için yöntem tarafından bir dizi oluşturularak Simülasyon Formu dosyasında yeniden kodlanır.

2. C Connection

```
CConnection:
public class CConnection
{
    public string StartNode = "";
    public string EndNode = "";
    public int Cost = 0;
    public int TimeDuration = 0;
    public CConnection(String Line)
    {
        String Left = Line.Split('=')[0];
        String Right = Line.Split('=')[1];
        StartNode=Left.Split(',')[0].Replace("\n", "").Replace("\r", "");
        EndNode=Left.Split(',')[1].Replace("\n", "").Replace("\r", "");
        Cost = int.Parse(Right.Split(',')[0]);
        TimeDuration = int.Parse(Right.Split(',')[1]);
    }
}
```

```

public CConnection() { }
public CConnection(String Start, String End, int Cost, int
    TimeDuration)
{
    StartNode = Start;
    EndNode = End;
    this.Cost = Cost;
    this.TimeDuration = TimeDuration;
}

public string ToCode()
{
    Return String.Format("{0},{1}={2},{3}", StartNode, EndNode, Cost,
        TimeDuration);
}
}

```

Sınıfta iki nod arasındaki her bir bağlantıya ait bilgileri kaydetmek için StartNode (kaynak), EndNode (hedef), Maliyet ve TimeDuration (zaman süresi) isimli dört değişken kullanılır.

```

public CConnection(String Line)
{
    String Left = Line.Split('=')[0];
    String Right = Line.Split('=')[1];
    StartNode=Left.Split(',')[0].Replace("\n", "").Replace("\r", "");
    EndNode = Left.Split(',')[1].Replace("\n", "").Replace("\r", "");
    Cost = int.Parse(Right.Split(',')[0]);
    TimeDuration = int.Parse(Right.Split(',')[1]);
}

```

Yöntem bir kurucu yöntemi olup Simülasyon Formu'nda iki nod arasındaki her bir bağlantıya ait bilgileri getirir.

```

public CConnection(String Start, String End, int Cost, int
    TimeDuration)
{
    StartNode = Start;
    EndNode = End;
    this.Cost = Cost;
    this.TimeDuration = TimeDuration;
}

```

Yöntem bir kurucu yöntemi olup Simülasyon Formu'nda iki nod arasındaki her bir bağlantıya ait bilgileri yazar.

```

public string ToCode()
{
    returnString.Format("{0},{1}={2},{3}", StartNode, EndNode, Cost, TimeDur;
}

```

Simülasyon Formu'nda kullanılan yöntem bağlantıları dosyaya kaydetmek için özel bir format belirler.

3. NetworkStimulator

Sınıf açılan dosyayı okur ve kaynak ve hedef nodlar arasındaki rotayı keşfeder. Bu sınıf tarafından belirlenen tüm değişkenlerin rotayı ve maliyeti alması gerekir (1. formül)

```

bool haveResult = false;
List<CNode> Nodes;
List<CConnection> Connections;
String[] Result = new String[1000];
string From = "";
string To = "";
double Path_t;
double N_energy_exp;
double N_energy_pow;
double P_cost ;
double P_time_exp ;
double result = 0;
double MinResult; String MinPath; bool IsMinResSetted;
double MaxResult; String MaxPath; bool IsMaxResSetted;
public int Custom;
public int Enterprise;
public int Military;
PictureBox picture = null;
ListBox lst = null;
Graphics grp;
Image resPic;
String InputPath = "";
String OutputPath = "";
readonly int NodeRaduis = 5;

```

Belirlenen yöntem () rotayı bulmakta kullanılan ana değişkenleri varsayılan moda çevirir. Formadaki Başlat tuşuna her basıldığında yöntem çağrılır.

```

public void set()
{
    MinResult = 0;
    MinPath = "";
    IsMinResSetted = false;
    MaxResult = 0;
    MaxPath = "";
    IsMaxResSetted = false;
    Custom = 0;
    Military = 0;
    Enterprise = 0;
}

```

Aşağıdaki yöntemle grafik arayüzü ve simüle edilen dosya varsayılan moda çevrilir ve simülasyon dosyası silinir.

```

public NetworkSimulator(String DataPath, String OutPath, PictureBox
Target, ListBox List)
{
    InputPath = DataPath;

    OutputPath = OutPath;
    File.Delete(OutPath);
    picture = Target;
    lst = List;
    resPic = new Bitmap(picture.Width - 1, picture.Height - 1);
    grp = Graphics.FromImage(resPic);
}

```

ReadFile() yöntemi açılan dosyayı okur ve kaynak ve hedef nodlar ile nodlar arasındaki bağlantılar dahil her bir nodla ilgili bilgileri toplar.

```

public void ReadFile()
{
    StreamReader r = new StreamReader(InputPath);
    String Line = "";
    Nodes = new List<CNode>();
    Connections = new List<CConnection>();
    Line = r.ReadLine();

    while (!r.EndOfStream)
    {
        Line = r.ReadLine();
        if (Line.Contains("->"))
            break;
        CNode temp = new CNode(Line);
        Nodes.Add(temp);
    }
    From=Line.Split(new String[]{"->"},
StringSplitOptions.RemoveEmptyEntries)[0];
    To = Line.Split(new String[] { "->" },
StringSplitOptions.RemoveEmptyEntries)[1];
    while (!r.EndOfStream)
    {
        Line = r.ReadLine();
        CConnection temp = new CConnection(Line);
        Connections.Add(temp);
    }
}

```

DrawNodes() yöntemi açılan dosyadaki verileri temel alarak nodları çeker.

```

public void DrawNodes()
{
    grp.Clear(Color.DarkGreen);
    foreach (CNode node in Nodes)
    {
        grp.FillEllipse(Brushes.LightBlue,node.X,node.Y,
NodeRaduis,NodeRaduis);
    }
    picture.Image = resPic;
}

```

SearchInResualts(dizi a, int b) yöntemi nod rotasının kaynak nodu hedef noda bağlayamaması ve yeniden bir rota aramaya başlaması halinde rotanın hedefe ulaşarak -1'e geri dönüp dönmeyeceğini saptamak için kullanılır.

```

int SearchInResults(string a, int b)
{
    for (int i = 0; i < b; i++)
        if (String.Compare(Result[i], a) == 0)
            return i;
    return -1;
}

```

SearchInConnections(dizi a, dizi b) iki nod arasındaki rotayı bulduğunda rotanın maliyetini ve noktanın geri dönüş endeksini hesaplar ve çağırılan yöntemdeki rota maliyetini günceller.

```

int SearchInConnections(string a, string b)
{
    for (int i = 0; i < Connections.Count; i++)
        if (((string.Compare(a, Connections[i].StartNode) == 0) &&
            (string.Compare(b, Connections[i].EndNode)
            ==0)) || ((string.Compare(a,Connections[i].EndNode) ==
            0)&&(string.Compare(b,Connections[i].StartNode) == 0)))
            return i;
    return -1;
}

```

Search In Nodes(dizi a) nodun nod dizisindeki konumunu tespit eder ve çağırılan yöntem kaynak ve hedef arasındaki rotadaki nodların enerjisin toplayabilir.

```

int SearchInNodes(string a)
{
    int i;
    for (i = 0; i < Nodes.Count; i++)
        if (string.Compare(Nodes[i].NodeName, a) == 0)
            return i;
    return -1;
}

```

Traverse(String From, String To, int hop) simülasyondaki en önemli yöntemdir. Yöntem kaynaktan hedef noda giden rotayı belirleyerek rotanın maliyetini hesaplar ve en kısa ve en uzun rotaları belirler. Yöntem ayrıca Simülasyon Dosyası'nı oluşturur ve dosyayı belirtilen hedef dizine kaydeder.

```

void Traverse(String From, String To, int hop)
{
    int i, j;
    string str = "";
    string Path = "";
    Result[++hop] = From;
    if (string.Compare(From, To) == 0)
    {
        int cost = 0;
        float l = 1;
        StreamWriter outfile = new StreamWriter(OutputPath, true);
        Path = Result[0];
        outfile.Write(Result[0]);
        P_cost = 0;
        Path_t = 0;
        P_time_exp = 0;
    }
}

```

```

N_energy_exp = 0;
N_energy_pow = 0;
for (i = 1; i <= hop; i++)
{
    outfile.Write('-');
    Path += "-" + Result[i];
    j = SearchInConnections(Result[i - 1], Result[i]);
    cost += Connections[j].Cost;
    P_cost+=Math.Pow(Connections[j].Cost,
        Connections[j].Cost);
    P_time_exp += Math.Exp(Connections[j].TimeDuration);
    Path_t += Connections[j].TimeDuration;
    outfile.Write(Result[i]);
}
for (i = 0; i <= hop; i++)
{
    j = SearchInNodes(Result[i]);
    N_energy_exp += Math.Exp(Nodes[j].Energy);
    N_energy_pow+=Math.Pow(Nodes[j].Energy,
        Nodes[j].Energy);
}
result = (Custom * ((N_energy_exp / (Math.Pow(hop, hop) *
    cost)) + (1 /Path_t)));
result += (Enterprise * ((N_energy_exp / (P_cost * hop)) +
    (1 /Path_t)));
result += (Military * ((N_energy_pow / (P_time_exp * hop))
    + (1 /cost)));
Path += "=" + result;
outfile.Write('=');
if (!IsMaxResSetted)
{
    MaxPath = Path;
    MaxResult = result;
    IsMaxResSetted = true;
}
else
{
    MaxPath = (MaxResult > result ? MaxPath : Path);
    MaxResult = (MaxResult > result? MaxResult: result);
}
if (!IsMinResSetted)
{
    MinPath = Path;
    MinResult = result;
    IsMinResSetted = true;
}
else
{
    MinPath = (MinResult < result ? MinPath : Path);
    MinResult = (MinResult < result ?MinResult: result);
}
lst.Items.Add(Path);
str = result.ToString();
outfile.WriteLine(str);
outfile.Close();
}
else
{
    for (i = 0; i < Connections.Count; i++)
    {
        if ((String.Compare(From, Connections[i].StartNode)
            ==0)&&(SearchInResults(Connections[i].EndNode,
            hop) == -1))
        {
            Traverse(Connections[i].EndNode, To, hop);
        }
        Else if ((String.Compare(From, Connections[i]
            .EndNode)==0)&&(SearchInResults(Connections[i]
            .StartNode, hop) == -1))
            Traverse(Connections[i].StartNode, To, hop);
        }
    }
    haveResult = true;
}
}

```

DrawPath(Dizi Yolu) yöntemi simülatörde listelenen rotaları grafik olarak çeker.

```

public string DrawPath(String Path)
{
    DrawNodes();
    String Right = Path.Split('=')[1];
    String Left = Path.Split('=')[0];
    String[] nodes = Left.Split('-');
    bool LineFlag = false;
    PointF TempP = new PointF();
    PointF LastP = new PointF(-1, -1);
    foreach (String node in nodes)
    {
        CNode nd = FindByName(node);
        grp.FillEllipse(Brushes.LightBlue, nd.X, nd.Y, NodeRaduis,
            NodeRaduis);
        if (!LineFlag)
        {
            TempP = new PointF(nd.X + NodeRaduis / 2, nd.Y +
                NodeRaduis / 2);
            if (LastP.X != -1)
                grp.DrawLine(Pens.Yellow, TempP, LastP);
        }
        else
            grp.DrawLine(Pens.Yellow, TempP, new PointF(nd.X +
                NodeRaduis / 2, nd.Y + NodeRaduis / 2));
        LineFlag = !LineFlag;
        LastP = new PointF(nd.X + NodeRaduis / 2, nd.Y+NodeRaduis/2);
    }
    CNode StartNode = FindByName(From);
    CNode EndNode = FindByName(To);
    SizeF sz=grp.MeasureString("Start", newFont("Tahoma", 8, FontStyle.Bold));
    grp.DrawString("Start", new Font("Tahoma", 8, FontStyle.Bold),
        Brushes.White, StartNode.X + sz.Width / 2, StartNode.Y);
    grp.FillEllipse(Brushes.DarkMagenta, StartNode.X, StartNode.Y,
        NodeRaduis, NodeRaduis);
    sz = grp.MeasureString("End", new Font("Tahoma", 8, FontStyle.Bold));
    grp.DrawString("End", new Font("Tahoma", 8, FontStyle.Bold),
        Brushes.White, EndNode.X + sz.Width / 2, EndNode.Y);
    grp.FillEllipse(Brushes.Red, EndNode.X, EndNode.Y, NodeRaduis,
        NodeRaduis);
    sz = grp.MeasureString("End", new Font("Tahoma", 8, FontStyle.Bold));
    grp.DrawString("End", new Font("Tahoma", 8, FontStyle.Bold),
        Brushes.White, EndNode.X + sz.Width / 2, EndNode.Y);
    grp.FillEllipse(Brushes.Red, EndNode.X, EndNode.Y, NodeRaduis,
        NodeRaduis);
    return Right;
}

```

FindByName(Dizi Adı) yöntemi kaynak ve hedef nodları bularak başlangıç ve bitiş nodlarının grafik arayüzünde vurgulanmasını sağlar **!**

```

private CNode FindByName(String Name)
{
    for (int i = 0; i < Nodes.Count; i++)
    {
        if (Nodes[i].NodeName == Name)
            return Nodes[i];
    }
    throw new Exception();
}

```

DrawMaxPath() ve DrawMinPath() yöntemleri en uzun ve en kısa rotaları belirler.


```

public double DrawMaxPath()
{
    DrawPath(MaxPath);
    return MaxResult;
}

public double DrawMinPath()
{
    DrawPath(MinPath);
    return MinResult;
}

```

4. Ağ Yapılandırma Kodları

Aşağıda nodları ve nodlar arasındaki rotaları çizmek için kullanılan yöntemler ele alınmıştır.

```

private void _PointOnNode(MouseEventArgs e)
{
    CurrentIndex = "";
    s = 0;
    foreach (string key in nodeRegions.Keys)
    {
        s++;
        if (nodeRegions[key].Contains(e.Location))
        {
            CurrentIndex = key;
            break;
        }
    }
    if (CurrentIndex != "")
    {
        _DrawPath();
        _StartEnd();
        grp.FillEllipse(Brushes.Fuchsia, nodeRegions[CurrentIndex]);
        picMap.Image = res;
        numX.Value = nodes[CurrentIndex].X;
        numY.Value = nodes[CurrentIndex].Y;
        lblEnergy.Text = nodes[CurrentIndex].Energy.ToString();

    }
    else
    {
        lblEnergy.Text = "0";
    }
    pnlNodeModify.Enabled = CurrentIndex != "";
}

```

Nodların ortama yerleştirilmesi sona erdiğinde ve operatörler bir nodla ilgili bilgi (konum, enerji) çekmek istediğinde yöntem çağrılır.

```

private void _DrawNodes()
{
    foreach (string key in nodeRegions.Keys)
    {
        grp.FillEllipse (Brushes.White, nodeRegions[key]);
    }
}

```

Yöntem ağ yapılandırma formundaki nodları çeker.

```

private void _DrawPath()
{
    res = new Bitmap(picMap.Width - 1, picMap.Height - 1);
    grp = Graphics.FromImage(res);
    for (int k = 0; k < i - 1; k++)
    {
        for (int j = 1; j < i; j++)
        {
            int dx = n[0, k] - n[0, j];
            int dy = n[1, k] - n[1, j];
            dx *= dx;
            dy *= dy;
            if ((int)Math.Floor(Math.Sqrt(dx + dy)) < (int)numRCR.Value)
            {
                grp.DrawLine(Pens.Yellow, n[0, k], n[1, k], n[0, j], n[1, j]);
            }
        }
    }
}

```

Yöntem iki nod arasındaki bağlantıları çeker.

```

private void _AddNode(MouseEventArgs e)
{
    DateTime now = DateTime.Now;
    String key=String.Format("{0}{1}{2}{3}", now.Hour, now.Minute,
        now.Second, now.Millisecond);
    nodeRegions.Add(key, newRectangle(e.X-Radius/2, e.Y-
        Radius/2, Radius, Radius));
    nodes.Add(key, new CNode("", e.X, e.Y, (rdInfKind.Checked ? (new
        Random()).Next(0, (int)numMaxEnergy.Value) :
        (int)numEnergy.Value));
    n[0, i] = e.X;
    n[1, i++] = e.Y;
    _DrawPath();
}

```

```

        _DrawNodes();
        picMap.Image = res;
        lblNodeCount.Text = nodeRegions.Count.ToString();
        pnlFile.Enabled = nodes.Count == (int)numNodeCount.Value;
        cmbTo.Items.Clear();
        cmbFrom.Items.Clear();
        cmbTo.Items.Clear();
        cmbFrom.Items.Clear();
        NameNodes();
        foreach (string Key in nodes.Keys)
        {
            cmbFrom.Items.Add(nodes[Key].NodeName);
            cmbTo.Items.Add(nodes[Key].NodeName);
        }
        cmbFrom.SelectedIndex = cmbTo.SelectedIndex = 0;
    }
}

```

Yöntem yeni nodları ortama ekler.

```

private void btnRemoveNode_Click(object sender, EventArgs e)
{
    for (int j = 0; j < i; j++)
    {
        if ((n[0, j] == nodes[CurrentIndex].X) && (n[1, j] ==
nodes[CurrentIndex].Y))
        {
            n[0, j] = n[0, i-1];
            n[1, j] = n[1, i-1];
            i--;
            btnRandomNode.Enabled = true;
        }
    }
    nodeRegions.Remove(CurrentIndex);
    nodes.Remove(CurrentIndex);
    CurrentIndex = "";
    _DrawPath();
    _DrawNodes();
    picMap.Image = res;
    pnlNodeModify.Enabled = false;
    lblNodeCount.Text = nodeRegions.Count.ToString();
    cmbTo.Items.Clear();
    cmbFrom.Items.Clear();
    cmbFrom.SelectedIndex = cmbTo.SelectedIndex = -1;
    pnlFile.Enabled = btnSave.Enabled = false;
}
}

```

Yöntem seçili nodu siler.

```

private void btnMove_Click(object sender, EventArgs e)
{
    if (((int)numWid.Value > (int)numX.Value) && ((int)numHei.Value >
        (int)numY.Value))
    {
        nodeRegions[CurrentIndex] = new Rectangle((int)numX.Value - Radius /
            2, (int)numY.Value - Radius / 2, Radius, Radius);
        for (int j = 0; j < i; j++)
        {
            if ((n[0, j] == nodes[CurrentIndex].X) && (n[1, j] ==
                nodes[CurrentIndex].Y))
            {
                n[0, j] = (int)numX.Value;
                n[1, j] = (int)numY.Value;
                nodes[CurrentIndex].X = (int)numX.Value;
                nodes[CurrentIndex].Y = (int)numY.Value;
            }
        }
        _rest();
    }
}

```

Yöntem seçili nodun konumunu değiştirir.

```

private void btnRandomNode_Click(object sender, EventArgs e)
{
    if (nodeRegions.Count < (int)numNodeCount.Value)
    {
        Random rnd = new Random();
        _AddNode(new MouseEventArgs(MouseButtons.Left, 1, rnd.Next
            ((int)numWid.Value), rnd.Next((int)numHei.Value), 0));
    }
    else
    {
        rdAddNode_CheckedChanged(sender, e);
        _DrawPath();
        _DrawNodes();
        pnlNodeModify.Enabled = true;
        btnRandomNode.Enabled = false;
        btnSave.Enabled = true;
        picMap.Image = res;
        cmbFrom_SelectionChangeCommitted(sender, e);
    }
}

```

Yöntem yeni nodları rastgele ortama ekler.

```

private void NameNodes()
{
    int i = 0;
    foreach (string key in nodes.Keys)
        nodes[key].NodeName = "n" + (i++);
}

```

Yöntem operatörün ortamdaki seçili nodu adlandırmasını sağlar.

```

private void CreateConnections()
{
    Dictionary<string, CNode>.ValueCollection nds = nodes.Values;
    CNode[] arrOfNodes = nds.ToArray<CNode>();
    Random rnd = new Random();
    for (int i = 0; i < arrOfNodes.Length; i++)
    {
        for(int j = i + 1 ; j < nds.Count ; j++)
        {
            CConnection con = new CConnection();
            int dx = arrOfNodes[i].X - arrOfNodes[j].X;
            int dy = arrOfNodes[i].Y - arrOfNodes[j].Y;
            dx *= dx;
            dy *= dy;
            int cost = (int)Math.Floor(Math.Sqrt(dx + dy));
            if (cost < (int)numRCR.Value)
            {
                con.StartNode = arrOfNodes[i].NodeName;
                con.EndNode = arrOfNodes[j].NodeName;
                con.Cost = cost;
                con.TimeDuration = rnd.Next((int)numMaxTime.Value);
                Connections.Add(con);
            }
        }
    }
}

```

Yöntem nodlar arasındaki bağlantıları tespit eder.

```

private void _StartEnd()
{
    _DrawNodes();
    int i = int.Parse(cmbFrom.Text.Substring(1));
    int j = int.Parse(cmbTo.Text.Substring(1));
    if (i != j)
        grp.FillEllipse(Brushes.Blue, n[0, j] - 2, n[1, j] - 2, Radius, Radius);
    grp.FillEllipse(Brushes.Red, n[0, i] - 2, n[1, i] - 2, Radius, Radius);
    picMap.Image = res;
}

```

Yöntem kaynak ve hedef nodları çeker.

5. Simülâtör kodları

```

private void btnLoad_Click(object sender, EventArgs e)
{
    if (dlgOpen.ShowDialog() == DialogResult.OK)
    {
        String Output = dlgOpen.FileName.Substring(0,
            dlgOpen.FileName.LastIndexOf('.'));
        Output += "-Out.txt";
        String Line = "";
        StreamReader r = new StreamReader(dlgOpen.FileName);
        Line = r.ReadLine();
        picResult.Width = int.Parse(Line.Substring(0,
            Line.IndexOf(',') .Substring(Line.IndexOf('=') + 1));
        picResult.Height = int.Parse(Line.Substring(Line.IndexOf(',') +
            1).Substring(Line.IndexOf('=') + 1));
        sim = new NetworkSimulator(dlgOpen.FileName, Output,
            picResult, lstPaths);
        label1.Text = "طول : " + picResult.Width;
        label4.Text = "عرض : " + picResult.Height;
        sim.ReadFile();
        sim.DrawNodes();

        btnStart.Enabled = FormulaEleman.Enabled = true;
    }
}

```

Yöntem Simülâtör Formu'ndaki ağ yapılandırma dosyasını açar.

```

private void btnStart_Click(object sender, EventArgs e)
{
    lstPaths.Items.Clear();
    StringOutput=dlgOpen.FileName.Substring(0,dlgOpen.FileName.LastIndexOf('.'));
    Output += "-Out.txt";
    String Line = "";
    StreamReader r = new StreamReader(dlgOpen.FileName);
    Line = r.ReadLine();
    picResult.Width=int.Parse(Line.Substring(0,
    Line.IndexOf(',') .Substring(Line.IndexOf('=') + 1));
    picResult.Height=int.Parse(Line.Substring(Line.IndexOf(',')
    +1).Substring(Line.IndexOf('=') + 1));
    sim = new NetworkSimulator(dlgOpen.FileName, Output, picResult, lstPaths);
    sim.ReadFile();
    sim.DrawNodes();
    sim.set();
    if (rdCustom.Checked)
        sim.Custom = 1;
    else if (rdMilitary.Checked)
        sim.Military = 1;
    else if (rdEnt.Checked)
        sim.Enterprise = 1;
        btnMaxPath.Enabled = btnMinPath.Enabled =label13.Enabled
        sim.StartSimulate();
}

```

Yöntem Başlat tuşuna tıklamayı yönetir ve kaynak ve hedef nodlar arasındaki rotaları belirler ve verileri simülâtör dosyasına kaydeder.

GOSSIPING PROTOKOLÜ YAZILIM

```
function
[lives, energy, loss_event, delay, Numpacket]=gossiping(node_, Event_, iteration_, Event_per_round_, Loc_, sinkx_, sinky_, networkx_, networky_, initial_Energy_, R_send_, R_sense_, avg_delay_, d, deadline)
global node
global S
global iteration
global Event_per_round
global Loc
global sinkx
global sinky
global networkx
global networky
global initial_Energy
global R_send
global avg_delay
global Data_packet
global Control_packet
global Eelec
global Efs
global Emp
global EDA
global d0
global draw
global Event
global R_sense
Event=Event_;
R_sense=R_sense_;
```

```

node=node_;
iteration=iteration_;
Event_per_round=Event_per_round_;
Loc=Loc_;
sinkx=sinkx_;
sinky=sinky_;
networkx=networkx_;
networky=networky_;
initial_Energy=initial_Energy_;
R_send=R_send_;
avg_delay=avg_delay_;
Data_packet=4000;
Control_packet=32;
Eelec=50*0.000000001;
Efs=10*0.000000000001;
Emp=0.0013*0.000000000001;
EDA=5*0.000000001;
d0=sqrt(Efs/Emp);
draw=d;
figure(10);
cla reset;
for i=1:1:node
    S(i).E=initial_Energy;
    S(i).xd=Loc(1,i);
    S(i).yd=Loc(2,i);
    S(i).type='normal';
    S(i).head=0;
    S(i).dist_to_BS=distance(S(i).xd,S(i).yd,sinkx,sinky);
    % plot(S(i).xd,S(i).yd,'o','Color','b');

```



```

    % hold on;
end
num =1;
delay(1:iteration*Event_per_round)=0;
loss_event(1:iteration*Event_per_round)=0;
Numpacket(1:iteration*Event_per_round)=0;
for i=0:1:(iteration*Event_per_round)-1
    display(i);
    energy(i+1)=energy_of_network;
    lives(i+1)=live_sensor;
    if lives(i+1)~=0
        [delay(i+1), loss_event(i+1), Numpacket(i+1)]=setup_phase(i+1);
    end
end

end
end
%-----
energy_of_network
function sum=energy_of_network()
    global S
    sum=0;
    for i=1:1:length(S)
        if S(i).E>0
            sum=sum+S(i).E;
        end
    end
end
end
%-----
live_sensor

```

```

function live=live_sensor()
    global S
    live=0;
    for i=1:1:length(S)
        if S(i).E>0
            live=live+1;
        else
            S(i).type=' dead' ;
        end
    end
end

end
%-----
node_detect_Event
function [delay, Loss, Numpacket]=setup_phase(r)
    global S
    global Data_packet
    global Control_packet
    global Event
    global R_sense
    global sinkx
    global sinky
    figure(10);
    Numpacket=0;
    Loss=0; delay=0;
    EX=Event(1, r);
    EY=Event(2, r);
    plot(EX, EY, ' p', ' Color', ' r');
    hold on;
    TH_candid=0;

```

```

for i=1:length(S)
    if distance(S(i).xd,S(i).yd,EX,EY)<R_sense    &&
~strcmp(S(i).type,'dead')
        S(i).cluster_member=1;
        TH_candid=i;
        Numpacket=Numpacket+Data_packet;
        plot(S(TH_candid).xd,S(TH_candid).yd,'o','Color','b');
        hold on;
    end
end
if TH_candid==0
    Loss=1;
    return ;
end
Loss=1;
S(TH_candid).head=1;
for h=1:length(S)
    k=0; X=0;
    for i=1:length(S)
        if S(i).head==1
            for j=1:length(S)
                if ~strcmp(S(j).type,'dead') && i~=j
                    k=k+1;
                    X(k)=j;
                end
            end
        end
    end
    if k~=0
        CH=round(rand*length(X));
        if CH==0

```

```

        delay=delay+1;
        Loss=0;
        S(i).E=S(i).E-
consume_transmit(S(i).dist_to_BS,Data_packet);
        S(i).head=0;
        line([S(i).xd                               sinkx], [S(i).yd
sinky], 'color', 'r');
        return;
    end
    CH=X(CH);
    delay=delay+1;
    S(CH).head=1;
    S(i).head=0;
    d=distance(S(i).xd,S(i).yd,S(CH).xd,S(CH).yd);
    S(i).E=S(i).E-consume_transmit(d,Data_packet);
    S(CH).E=S(CH).E-consume_receive(Control_packet);
    line([S(i).xd                               S(CH).xd], [S(i).yd
S(CH).yd], 'color', 'y');
    if S(CH).E<0
        S(CH).type='dead';
    end
    if S(i).E<0
        S(i).type='dead';
    end
end
end
end
end
for i=1:length(S)

```

```

        S(i).head=0;
    end
end
%-----
consume_fusion
function con=consume_fusion(data)
    global EDA;
    con=data*EDA;
end
%-----
consume_transmit
function con=consume_transmit(LEACH_AD_DISTANCE, Control_packet)
    global Eelec
    global Efs
    global Emp
    global d0
    if LEACH_AD_DISTANCE<d0

con=2*(Control_packet*Eelec+Control_packet*Efs*LEACH_AD_DISTANCE^2);
        else

con=2*(Control_packet*Eelec+Control_packet*Emp*LEACH_AD_DISTANCE^4);
        end
    end
%-----
consume_receive
function con=consume_receive(Control_packet)
    global Eelec
    con=Control_packet*Eelec;

```

```
end
```

```
%-----
```

```
distance
```

```
function d=distance(x1, y1, x2, y2)
```

```
    d=sqrt((x1-x2)^2+(y1-y2)^2);
```

```
end
```

LEACH PROTOKOLÜ YAZILIMI

```

function
[variance,lives,energy,num_packet,Energy_CH,num_CH]=LEACH_PR(location,it,Ene,
nodes_)
format long
global xm
global ym
global sinkx
global sinky
global Data_packet
global Control_packet
global Eelec
global Efs
global Emp
global EDA
global d0
global alpha
global S
global non_CH
global init_energy
%%
%-----Main Parameters
nodes=nodes_;
iteration=it;
init_energy=Ene;
number_of_clusterhead_for_leach=5;
xm=100;
ym=100;
sinkx=0;
sinky=0;
%%
Data_packet=4000;

```

```

Control_packet=32;
Eelec=50*0.000000001;
Efs=10*0.000000000001;
Emp=0.0013*0.000000000001;
EDA=5*0.000000001;
d0=sqrt(Efs/Emp);
alpha=0.15;
non_CH=0;
KRound=iteration;
p=number_of_clusterhead_for_leach/nodes;
figure(1);
for i=1:1:nodes
    S(i).E=init_energy;
    S(i).xd=location(1,i);
    S(i).yd=location(2,i);
    S(i).type='normal';
    S(i).my_clusterhead=-1;
    S(i).dist_to_myClusterhead=100000;
    S(i).dist_to_BS=0;
    S(i).Head_in_round=-100;
    S(i).cluster_member=-1;
    S(i).packets=0;
end
Energ=0;
num=1;
for i=0:1:iteration-1
    display(i);
    energy(i+1)=energy_of_network();
    lives(i+1)=live_sensor();
    Energy_CH(i+1)=0;
    num_packet(i+1)=0;
    num_CH(i+1)=0;
    %-----

```



```

%Calculate variance of energy
%-----
for k=1:nodes
    Energ(k)=S(k).E;
end
variance(i+1)=var(Energ);
if lives(i+1)~=0
    nCluster=head_election_pahse(i,p);
    num_CH(i+1)=nCluster;
    num_packet(i+1)=lives(i+1)*Data_packet*alpha;
    if nCluster~=0
        Energy_CH(i+1)=Energy_consumption_CH;
        %advertisement phase
        advertisement_phase(nCluster);
        %cluster setup phase
        cluster_setup_phase();
        %schedule creation phase
        schedule_creation_phase();
        %data transmission phase
        num_packet(i+1)=data_transmission_phase();
        % Calculate place Sink

    end
    prepare_to_next_round();
end
end
end%function leach
%-----energy_of_network
function sum=energy_of_network()
    global S
    sum=0;
    for i=1:1:length(S)
        if S(i).E>0

```

```

        sum=sum+S(i).E;
    end
end
end
end
%-----live_sensor
function live=live_sensor()
    global S
    live=0;
    for i=1:1:length(S)
        if S(i).E>0
            live=live+1;
        else
            S(i).type='dead';
        end
    end
end
end
%-----head_election_pahse
function [number_of_cluster]=head_election_pahse(r,p)
    global S
    global sinkx
    global sinky
    global Data_packet
    global non_CH
    threshold=p/(1-p*(mod(r,1/p)));
    number_of_cluster=0;
%   mysum=0;
%   for i=1:1:length(S)
%       if S(i).E>0
%           mysum=mysum+S(i).E;
%       end
%   end
%
%   for i=1:1:length(S)

```

```

    random=rand(1,1);
    if      random<threshold    &&    S(i).Head_in_round    <=    r-1/p    &&
~strcmp(S(i).type,'dead') %random<((S(i).E*5)/mysum) & ~strcmp(S(i).type,'dead')
        S(i).type='cluster';
        S(i).dist_to_BS=distance(S(i).xd,S(i).yd,sinkx,sinky);
        S(i).Head_in_round=r;
        number_of_cluster=number_of_cluster+1;
    end
end
%-----Direct Transmit
if number_of_cluster==0
    non_CH=non_CH+1;
    for i=1:1:length(S)
        if ~strcmp(S(i).type,'dead')
            S(i).dist_to_BS=distance(S(i).xd,S(i).yd,sinkx,sinky);
            S(i).E=S(i).E-consume_transmit(S(i).dist_to_BS,Data_packet);
            if S(i).E<=0
                S(i).type='dead';
            end
        end
    end
end
end
end

end
%-----advertisement_phase
function advertisement_phase(nCluster)
    global S
    global ym
    global xm
    global Control_packet
    LEACH_AD_DISTANCE=sqrt(ym^2+xm^2)/2;
    for i=1:1:length(S)
        if strcmp(S(i).type,'cluster')

```

```

        S(i).E=S(i).E-consume_transmit(LEACH_AD_DISTANCE,Control_packet);
    elseif ~strcmp(S(i).type,'dead')
        S(i).E=S(i).E-(nCluster*consume_receive(Control_packet));
    end
end
end
%-----cluster_setup_phase
function cluster_setup_phase()
    global S
    global Control_packet
    for i=1:1:length(S)
        if strcmp(S(i).type,'normal') && ~strcmp(S(i).type,'dead')
            for j=1:1:length(S)
                if strcmp(S(j).type,'cluster')
                    if distance(S(i).xd,S(i).yd,S(j).xd,S(j).yd)<S(i).dist_to_myClusterhead
                        S(i).dist_to_myClusterhead=distance(S(i).xd,S(i).yd,S(j).xd,S(j).yd);
                        S(i).my_clusterhead=j;
                    end
                end
            end
        end
        %S(i).E=S(i).E-consume_transmit(SETUP_DISTANCE,Control_packet);
        S(i).E=S(i).E-consume_transmit(S(i).dist_to_myClusterhead,Control_packet);

        S(S(i).my_clusterhead).cluster_member=S(S(i).my_clusterhead).cluster_member+1;
        S(S(i).my_clusterhead).E=S(S(i).my_clusterhead).E-
        consume_receive(Control_packet);
        line([S(i).xd S(S(i).my_clusterhead).xd],[S(i).yd S(S(i).my_clusterhead).yd])
    end
end
end
%-----schedule_creation_phase
function schedule_creation_phase()
    global S

```

```

global Control_packet
global ym
global xm
SCHEDULE_DISTANCE=sqrt(ym^2+xm^2)/2;
for i=1:1:length(S)
    if strcmp(S(i).type,'cluster')
        S(i).E=S(i).E-
consume_transmit(SCHEDULE_DISTANCE,(S(i).cluster_member+1)*Control_packet
);
        if S(i).E<=0
            S(i).type='dead';
        end
    else
        S(i).E=S(i).E-consume_receive(Control_packet);
    end
end
end
%-----data_transmission_phase
function numpacket=data_transmission_phase()
global S
global Data_packet
global alpha
numpacket=0;
for i=1:1:length(S)
    if strcmp(S(i).type,'normal') && ~strcmp(S(i).type,'dead')
        S(i).E=S(i).E-consume_transmit(S(i).dist_to_myClusterhead,Data_packet);
        S(S(i).my_clusterhead).E=S(S(i).my_clusterhead).E-
consume_receive(Data_packet);
        if S(i).E<=0
            S(i).type='dead';
        end
        if S(S(i).my_clusterhead).E<=0
            S(S(i).my_clusterhead).type='dead';

```

```

        end
    elseif strcmp(S(i).type,'cluster')
        S(i).E=S(i).E-consume_fusion(Data_packet*(S(i).cluster_member+1));
        S(i).packets=Data_packet*(S(i).cluster_member+1)*alpha;
        S(i).E=S(i).E-consume_transmit(S(i).dist_to_BS,S(i).packets);
        numpacket=numpacket+S(i).packets;
    end
end
end
end
%----- Energy_consumption_CH
function ECH=Energy_consumption_CH()
    global S
    ECH=0;
    for i=1:1:length(S)
        if strcmp(S(i).type,'cluster') && S(i).E>0
            ECH=ECH+S(i).E;
        end
    end
end
end
%-----prepare_to_next_round
function prepare_to_next_round()
    global S
    global sinkx
    global sinky
    for i=1:1:length(S)
        if ~strcmp(S(i).type,'dead')
            S(i).type='normal';
            S(i).my_clusterhead=-1;
            S(i).dist_to_myClusterhead=100000;
            S(i).dist_to_BS=distance(S(i).xd,S(i).yd,sinkx,sinky);
            S(i).cluster_member=0;
            S(i).packets=0;
        end
    end
end

```

```

    end
end
%-----consume_fusion
function con=consume_fusion(data)
    global EDA;
    con=data*EDA;
end
%-----consume_receive
function con=consume_receive(Control_packet)
    global Eelec
    con=Control_packet*Eelec;
end
%-----distance
function d=distance(x1,y1,x2,y2)
    d=sqrt((x1-x2)^2+(y1-y2)^2);
end
%-----
function con=consume_transmit(LEACH_AD_DISTANCE,Control_packet)
    global Eelec
    global Efs
    global Emp
    global d0
    if LEACH_AD_DISTANCE<d0
        con=Control_packet*Eelec+Control_packet*Efs*LEACH_AD_DISTANCE^2;
    else
        con=Control_packet*Eelec+Control_packet*Emp*LEACH_AD_DISTANCE^4;
    end
end
end

```

LOKASYÖN VE BÖLÜM SAYIM UYGULAMASI

```

function Loc=Location(node,SX,SY,NX,NY,d)
Loc(1,1)=SX; Loc(2,1)=SY;
for i=1:node
    Conflict=1;
    while Conflict
        x=round(rand*NX);
        y=round(rand*NY);
        Conflict=0;
        %s=size(Loc);
        %for j=1:s(2)
        % if distance(Loc(1,j),Loc(2,j),x,y)<10
        %     Conflict=1;
        % end
        %end
    end
    Loc(1,i)=x;
    Loc(2,i)=y;
    %if d
        % plot(x,y,'.','Color','k');
        %hold on;
    %end
end
end
function d=distance(x1,y1,x2,y2)
d=sqrt((x1-x2)^2+(y1-y2)^2);
end
%-- Lokasiyon hazirlama uygulaması
function point = LocationPre(node_,sinkx_,sinky_,networkx_,networky_)
global node
global networkx
global sinkx

```



```

global sinky
networkx=networkx_;
node=node_;
sinkx=sinkx_;
sinky=sinky_;
b=networkx/100;
j=b*4;
plot(sinkx,sinky,'^','Color','m');
hold on;
j1=j; w=1;
    for k=0:1:b-1
        q=45;
        x1=k*100+50;
        for z=1:1:4
            x=x1*cos((pi*q)/180);
            y=x1*sin((pi*q)/180);
            q=q+90;
            j1=j1-1;
            point(1,w)=x;
            point(2,w)=y;
            w=w+1;
            plot(x,y,'*','Color','g');
            hold on;
        end
    end
end
for i=j+1 :node
    x=(round(rand*(networkx+networkx))-networkx);
    y=(round(rand*(networkx+networkx))-networkx);
    point(1,i)=x;
    point(2,i)=y;
    plot(x,y,'.','Color','k');
    hold on;
end

```

```

end
end
%-- Bölüm sayim uygulaması
function Nsector = numsector(node,point_,sinkx_,sinky_)
global point
global networkx
global sinkx
global sinky
point=point_;
z=networkx/50;
for i=1:node
    for j=0:z
        l=j*50;
        h=(j+1)*50;
        d=distanc(point(1,i),point(2,i),sinkx,sinky);
        if (l<d && d<h)
            Nsector(i)=j+1;
            if Nsector(i)==1
                plot(point(1,i),point(2,i),'.','Color','g');
                hold on;
            else
                plot(point(1,i),point(2,i),'.','Color','b');
                hold on;
            end
        end
    end
end

end

end
end
%-----distance
function d=distanc(x1,y1,x2,y2)
d=sqrt((x1-x2)^2+(y1-y2)^2);
end

```

EESR PROTOKOLÜ UYGULAMASI

function

```
[lives,energy,loss_event,delay,Numpacket]=PreEESR(node_,Event_,iteration_,Event_per_round_,point_,sector_,sinkx_,sinky_,networkx_,networky_,initial_Energy_,R_sense_,R_send_,avg_delay_,d)
```

global node

global Event_per_round

global point

global sector

global Data_packet

global initial_Energy

global R_sense

global R_send

global Event

global Control_packet

global draw

global sinkx

global sinky

global networkx

global networky

global s

global iteration

global avg_delay

global Eelec

global Efs

global Emp

global EDA

global d0

sector=sector_;

avg_delay=avg_delay_;

sink.x=sinkx_;

```

sink.y=sinky_;
network.x=networkx_;
network.y=networky_;
node=node_;
Event_per_round=Event_per_round_;
point=point_;
iteration=iteration_;
initial_Energy=initial_Energy_;
R_sense=R_sense_;
R_send=R_send_;
Event=Event_;
Data_packet=4000;
Control_packet=32;
draw=1;
Eelec=50*0.0000000001;
Efs=10*0.0000000000001;
Emp=0.0013*0.0000000000001;
EDA=5*0.0000000001;
d0=sqrt(Efs/Emp);
b=networkx/100;
j=b*4;
figure(10);
cla reset;
for i=1:j
    s(i).xd=point(1,i);
    s(i).yd=point(2,i);
    s(i).E=initial_Energy;
    s(i).type='manage';
    s(i).dist_to_BS=distanc(s(i).xd,s(i).yd,sinkx,sinky);
    s(i).number_of_sector=sector(i);
    s(i).next=0;
    plot(point(1,i),point(2,i),'*','Color','g');
end

```

```

for i=j+1:node
    s(i).xd=point(1,i);
    s(i).yd=point(2,i);
    s(i).E=initial_Energy;
    s(i).type='normal';
    s(i).dist_to_BS=distanc(s(i).xd,s(i).yd,sinkx,sinky);
    s(i).number_of_sector=sector(i);
    s(i).next=0;
    plot(point(1,i),point(2,i),'.','Color','b');
end
delay(1:iteration*Event_per_round)=0;
loss_event(1:iteration*Event_per_round)=0;
Numpacket(1:iteration*Event_per_round)=0;
for i=0:1:(iteration*Event_per_round)-1
    display(i);
    energy(i+1)=energy_of_network;
    lives(i+1)=live_sensor;
    if lives(i+1)~=0
        [loss_event(i+1),Numpacket(i+1)]=setup_phase(i+1);
        if loss_event(i+1)~=1
            delay(i+1)=data_transmission;
        end
    end
end
end
end
%-----distance to bs
function d=distanc(x1,y1,x2,y2)
d=sqrt((x1-x2)^2+(y1-y2)^2);
end
%-----energy_of_network
function sum=energy_of_network()
    global s
    sum=0;

```

```

for i=1:1:length(s)
    if s(i).E > 0
        sum=sum+s(i).E;
    end
end
end
%-----live_sensor
function live=live_sensor(r)
    global s
    global R_send
    global Control_packet
    for i=1:length(s)
        if s(i).E >=consume_transmit(R_send,Control_packet)
            s(i).E=s(i).E-consume_transmit(R_send,Control_packet);
        end
    end
    live=0;
    for i=1:1:length(s)
        if s(i).E>consume_transmit(R_send,Control_packet)
            live=live+1;
        else
            s(i).type='dead';
        end
    end
end
end
%-----consume_transmit
function con=consume_transmit(LEACH_AD_DISTANCE,Control_packet)
    global Eelec
    global Efs
    global Emp
    global d0
    if LEACH_AD_DISTANCE<d0

```

```

con=4*(Control_packet*Eelec+Control_packet*Efs*LEACH_AD_DISTANCE^2);
    else

con=4*(Control_packet*Eelec+Control_packet*Emp*LEACH_AD_DISTANCE^4);
    end
end
%-----node_detect_Event
function [Loss,Numpacket]=setup_phase(r)
    global s
    global Event
    global R_sense
    global R_send
    global Data_packet
    global index
    Numpacket=0;
    EX=Event(1,r);
    EY=Event(2,r);
    plot(EX,EY,'p','Color','r');
    hold on;
    index=1;
    Max_E=0;
    TH_candid=0;
    % nodes that detect an event
    for i=1:length(s)
        if distanc(s(i).xd,s(i).yd,EX,EY)<R_sense && ~strcmp(s(i).type,'dead')
            s(i).next=1;
            plot(s(i).xd,s(i).yd,'o','Color','b');
            hold on
            Numpacket=Numpacket+Data_packet;
            s(i).E=s(i).E-consume_receive(Data_packet);
            s(i).E=s(i).E-consume_transmit(R_send,Data_packet);
            if s(i).E<=0

```

```

        s(i).type='dead';
    end
    TH_candid(index)=i;
    index=index+1;
end
end
if TH_candid==0
    Loss=1;
    return;
end
if TH_candid
    Loss=0;

    return;
end
end
%-----consume_receive
function con=consume_receive(Control_packet)
    global Eelec
    con=Control_packet*Eelec;
end
%-----data-transmission
function delay=data_transmission()
global s
global draw
global R_send
global sinkx
global sinky
global Data_packet
global networkx
minimum=networkx;
sum_delay=0;
sucess=0;

```



```

M=0;
for p=0:1:length(s)
    for j=length(s):-1:1
        if s(j).next==1 && distanc(sinkx,sinky,s(j).xd,s(j).yd)<=R_send &&
~strcmp(s(j).type,'dead')
            s(j).E=s(j).E-
consume_transmit(distance(sinkx,sinky,s(j).xd,s(j).yd),Data_packet);
            if s(j).E<=0
                s(j).type='dead';
            end
            sum_delay=sum_delay+1;
            if draw
                line([sinkx s(j).xd],[sinky s(j).yd],'Color','k');
                hold on
            end
            s(j).next=0;
        elseif s(j).next==1 && strcmp(s(j).type,'manage')
            j1=s(j).number_of_sector;
            dim=R_send;
            for i=1:length(s)
                W=s(i).number_of_sector;
                if j1 > W && strcmp(s(i).type,'manage') && j~=i
                    if distanc(s(i).xd,s(i).yd,s(j).xd,s(j).yd)<= dim
                        dim=distanc(s(i).xd,s(i).yd,s(j).xd,s(j).yd);
                        sucess=i;
                    end
                end
            end
            if sucess~=0
                i=sucess;
                s(j).E=s(j).E-
consume_transmit(distance(s(i).xd,s(i).yd,s(j).xd,s(j).yd),Data_packet);
                if s(j).E<=0

```

```

    s(j).type='dead';
end
s(i).next=1;
s(j).next=0;
s(i).E=s(i).E-consume_receive(Data_packet);
if s(i).E<=0
    s(i).type='dead';
end
sum_delay=sum_delay+1;
if draw
    line([s(i).xd s(j).xd],[s(i).yd s(j).yd],'Color','c');
    hold on;
end
end
elseif s(j).next==1 && strcmp(s(j).type,'normal')
    MINI=R_send;
    j1=s(j).number_of_sector;
    suces=0;
    for i=1:length(s)
        W=s(i).number_of_sector;
        if j1<=W && strcmp(s(i).type,'manage') && j~=i
            if distanc(s(i).xd,s(i).yd,s(j).xd,s(j).yd)<= R_send
                suces=i;
            end
        end
    end
end
if suces ~=0
    i=suces;
    s(j).E=s(j).E-
consume_transmit(distance(s(i).xd,s(i).yd,s(j).xd,s(j).yd),Data_packet);
    if s(j).E<=0
        s(j).type='dead';
    end
end

```

```

s(i).next=1;
s(j).next=0;
s(i).E=s(i).E-consume_receive(Data_packet);
if s(i).E<=0
    s(i).type='dead';
end
sum_delay=sum_delay+1;
if draw
    line([s(i).xd s(j).xd],[s(i).yd s(j).yd],'Color','g');
    hold on;
end
end
x3=1;
for i=1:length(s)
    if distanc(s(i).xd,s(i).yd,s(j).xd,s(j).yd)<= R_send &&
strcmp(s(i).type,'normal') && suces==0 && j~=i
        M(x3)=i;
        x3=x3+1;
    end
end
if M~=0
    x2=round(rand*x3);
    if x2==0 || x2==x3
        x2=1;
    end
    M1=M(x2);
    s(j).E=s(j).E-
consume_transmit(distanc(s(M1).xd,s(M1).yd,s(j).xd,s(j).yd),Data_packet);
    if s(j).E<=0
        s(j).type='dead';
    end
    s(M1).next=1;
    suces=0;

```

```
s(j).next=0;
s(M1).E=s(M1).E-consume_receive(Data_packet);
if s(i).E<=0
    s(i).type='dead';
end
sum_delay=sum_delay+1;
if draw
    line([s(M1).xd s(j).xd],[s(M1).yd s(j).yd],'Color','m');
    hold on;
end
M=0;
end
end
end
delay=sum_delay;
end
```

PROTOKOLLARI KARŞILAŞTIRMA UYGULAMASI

```

function simulation
node=200;
iteration=400;
network.x=200;
network.y=200;
sink.x=0;
sink.y=0;
initial_Energy=.1;
R_send=25;
R_sense=R_send/2;
Event_per_round=1;
deadline=.17;
alpha=.001;
avg_delay=sqrt(network.x*network.y/node)*alpha;
draw=0;
%%
% initialize Variable
Loc=Location(node,sink.x,sink.y,network.x,network.y,draw);
% Location for EESR
LocEESR=LocationPre(node,sink.x,sink.y,network.x/2,network.y/2);
sector=numsector(node,LocEESR,sink.x,sink.y);%Number of sector in EESR
%%-----DLGossiping
% [alive,energy,loss_event,delay]=DlGossiping(node,Event,iteration,Event_per_round,L
oc,sink.x,sink.y,network.x,network.y,initial_Energy,R_send,R_sense,avg_delay,draw,d
eadline);
[variance,alive,energy,Numpacket,Energy_CH,num_CH]=LEACH_PR(Loc,iteration,ini
tial_Energy,node);
% subplot(3,2,1);
figure(1);
cla reset ;
xlabel('Iteration');

```

```

ylabel('Live Node');
title('Live Node Per Round');
x=0:1:iteration*Event_per_round-1;
line(x,alive,'Color','r');
%subplot(3,2,2);
figure(2);
cla reset;
xlabel('Iteration');
ylabel('Total Energy Nodes');
title('Total Energy Per Round');
line(x,energy,'Color','r');
%subplot(3,2,3);
figure(5);
cla reset;
xlabel('Iteration');
ylabel('Number Packet');
title('Number Packet');
for i=1:length(Numpacket)-1
    Numpacket(i+1)=Numpacket(i)+Numpacket(i+1);
end
line(x,Numpacket,'Color','r');
%-----EESR
Eve_x(1:Event_per_round*iteration)=round(((network.x/2)*rand(1,Event_per_round*it
eration))-((network.x/2)*rand(1,Event_per_round*iteration)));
Eve_y(1:Event_per_round*iteration)=round(((network.y/2)*rand(1,Event_per_round*it
eration))-((network.y/2)*rand(1,Event_per_round*iteration)));
Event=[Eve_x;Eve_y];
[alive,energy,loss_event,delay,Numpacket]=PreEESR(node,Event,iteration,Event_per_r
ound,LocEESR,sector,sink.x,sink.y,network.x,network.y,initial_Energy,R_sense,R_sen
d,avg_delay,draw);
%subplot(3,2,1);
figure(1);
x=0:1:iteration*Event_per_round-1;

```

```

line(x,alive,'Color','blue');
%subplot(3,2,2);
figure(2);
line(x,energy,'Color','blue');
%subplot(3,2,3);
figure(3);
for i=1:length(loss_event)-1
    loss_event(i+1)=loss_event(i)+loss_event(i+1);
end
line(x,loss_event,'Color','blue');
%subplot(3,2,4);
figure(4);
for i=1:10:length(delay)-1
    dsum=0;
    for j=i:i+9
        dsum=dsum+delay(j);
    end
    for j=i:i+9
        delay(j)=dsum/10;
    end
end
end
line(x,delay,'Color','blue');
figure(5);
for i=1:length(Numpacket)-1
    Numpacket(i+1)=Numpacket(i)+Numpacket(i+1);
end
line(x,Numpacket,'Color','blue');
%-----Direct ...
Eve_x(1:Event_per_round*iteration)=round(network.x*rand(1,Event_per_round*iteration));
Eve_y(1:Event_per_round*iteration)=round(network.y*rand(1,Event_per_round*iteration));
Event=[Eve_x,Eve_y];

```

```

[alive,energy,loss_event,delay,Numpacket]=Directdefussion(node,Event,iteration,Event
_per_round,Loc,sink.x,sink.y,network.x,network.y,initial_Energy,R_send,R_sense,avg_
delay,draw,deadline);
%subplot(3,2,1);
figure(1);
x=0:1:iteration*Event_per_round-1;
line(x,alive,'Color','g');
figure(2);
line(x,energy,'Color','g');
figure(3);
for i=1:length(loss_event)-1
    loss_event(i+1)=loss_event(i)+loss_event(i+1);
end
line(x,loss_event,'Color','g');
%subplot(3,2,4);
figure(4);

for i=1:10:length(delay)-1
    dsum=0;
    for j=i:i+9
        dsum=dsum+delay(j);
    end
    for j=i:i+9
        delay(j)=dsum/10;
    end
end
line(x,delay,'Color','g');
figure(5);
for i=1:length(Numpacket)-1
    Numpacket(i+1)=Numpacket(i)+Numpacket(i+1);
end
line(x,Numpacket,'Color','g');
%-----Gossiping

```



```

[alive,energy,loss_event,delay,Numpacket]=gossiping(node,Event,iteration,Event_per_r
ound,Loc,sink.x,sink.y,network.x,network.y,initial_Energy,R_send,R_sense,avg_delay,
draw,deadline);
%subplot(3,2,1);
figure(1);
x=0:1:iteration*Event_per_round-1;
line(x,alive,'Color','c');
legend('LEACH','EESR','Direct defussion' ,'gossiping');
figure(2);
%subplot(3,2,2);
legend('LEACH','EESR','Direct defussion' ,'gossiping');
line(x,energy,'Color','c');
figure(3);
legend('LEACH','EESR','Direct defussion' ,'gossiping');
for i=1:length(loss_event)-1
    loss_event(i+1)=loss_event(i)+loss_event(i+1);
end
line(x,loss_event,'Color','c');
%subplot(3,2,4);
figure(4);
legend('LEACH','EESR','Direct defussion' ,'gossiping');
for i=1:10:length(delay)-1
    dsum=0;
    for j=i:i+9
        dsum=dsum+delay(j);
    end
    for j=i:i+9
        delay(j)=dsum/10;
    end
end
line(x,delay,'Color','c');
figure(5);
legend('LEACH','EESR','Direct defussion' ,'gossiping');

```

```
for i=1:length(Numpacket)-1
    Numpacket(i+1)=Numpacket(i)+Numpacket(i+1);
end
line(x,Numpacket,'Color','c');
end
```

EK 2. TÜRKÇE - İNGİLİZCE TERİMLER SÖZLÜĞÜ

Kablosuz Algılayıcı Ağlar : Wireless Sensor Network

Yönlendirme Protokolü: Routing Protocol

Enerji Verimliliği: Energy Efficiency

Güç Tasarrufu: Power Saving

Genetik Algoritma: Genetic Algorithm

Ağ Optimizasyonu: Network Optimization

Ağ Uygulama: Network Application

Dağıtım: Packet Distribution

Ömür: Lifetime

Gecikme: Delay

Süre: Time Duration

Maliyet: Cost

Hizmet Kalitesi: Quality Of Service

Kümelene: Clustering

Uygunluk Fonksiyon: Fitness Function

Çaprazlama: Crossover

İlişki: Relation

Düğüm Yerleştirme: Node Placement

Ağ Kapsama: Network Coverage

Veri Toplama: Data Aggregation

Mobil Sensör Ağ: Mobile Sensor Network

Kromozom: Chromosome

Baz İstasyonu: Base Station

Küme Baş: Cluster Head

Enerji Tüketimi: Energy Consumption

Değerlendirme: Evaluation

Etkili Parametreleri: Effective Parameters

Uygun Yönlendirme: Optimal Routing

Komşu Düğüm: Neighbor Node

İstek Mesajı: Interest Message

Yakın Dügümler: Closer Nodes

Teslim Süresi: Delivery Time

Yol Seçimi: Path Selection

Uygun Seçim: Optimum Choice

Başarı Ve Başarısızlık Oranı: Success And Failure Rate

Özel Uygulama: Application-Specific

İletim Güvenilirlik Riski: Risk Of Transmission-Reliability

Verimli İletişim: Efficient Communication

Eşler Arası İletişim: Peer-Peer Communication

Tek Hop: Single-Hop

Takviye: Reinforcements

EK 3: TEZ DEN ÇIKAN YAYINLAR

Norouzi, A; Zaim, A.H; Sertbas, A, “A comparative study based on power usage performance for routing protocols in wireless sensor network”, In proceeding of 2013 IEEE TAECE, ISBN 978-1-4673-5612-1, Pages 10-15 (**IEEE International Conference**)

Ali Norouzi, Faezeh Sadat Babamir, A Halim Zaim, An Interactive Genetic Algorithm for Mobile Sensor Networks, Studies in Informatics and Control, ISSN 1220-1766, vol. 22 (2), pp. 213-218, 2013.(**IF 0.554**)

Ali Norouzi and A. Halim Zaim, “Genetic Algorithm Application in Optimization of Wireless Sensor Networks,” The Scientific World Journal, vol. 2014, Article ID 286575, 15 pages, 2014. doi:10.1155/2014/286575 (**IF 1.73**)

Norouzi, A.; Sertbas, A; Zaim, A,H., “Evaluation of Effective Parameters of Energy Consumption in Wireless Sensor Networks” Book Chapter in Horizons in Computer Science Research, Nova Science Publisher, USA 2014, Under publication (**BOOK CHAPTER**)

Norouzi, A; Zaim, A.H; Sertbas, A. “Energy Efficient Coverage Optimization in Wireless Sensor Networks based on Genetic Algorithm” Advances in Electrical and Computer Engineering journal, ISSN: 1582-7445, Submitted Dec 2013, Under Evaluation. (**IF 0.552**)

ÖZGEÇMİŞ



Kişisel Bilgiler

Adı Soyadı	Ali Nourouzi Allayleh
Uyruğu	Iran
Doğum tarihi, Yeri	22.06.1980,MOGHAN
Telefon	05303337830
E-mail	norouzi@ieee.org
Web adres	www.istanbul.edu.tr

Eğitim

Derece	Kurum/Anabilim Dalı/Programı	Yılı
Doktora	İ.Ü. Fen Bilimleri Enstitüsü/ ,Bilgisayar Müh. /	2014
Yüksek Lisans	Khazar Univ,Bilgisayar müh /	2008
Lisans	Bilgisayar Mühendisliği ,Tebriz AZAD Univ	2002
Lise	Dr Hesabi	1998

Makaleler / Bildiriler

Norouzi, A., Dabbaghian, M., Hatamizadeh, A. and Üstündağ, B. B., 2010, "An Improved ELGossiping Data Distribution Technique with Emphasis on Reliability and Resource Constraints in Wireless Sensor Network," *IEEE International Conference on Electronic Computer Technology*, Kuala Lumpur, 7-10 May, pp. 179-183.

Norouzi, A., Babamir, F. and Zaim, A., 2011, "A Novel Energy Efficient Routing Protocol in Wireless Sensor Networks," *Wireless Sensor Network*, Vol. 3 No. 10, pp. 341-350. doi: 10.4236/wsn.2011.310038.

Khodashahi, M. H., Norouzi, A., Amiri, F. and Dabbaghian, M., 2010, "A Novel Optimal Routing Algorithm by Creating Concentrically Sectors in Wireless Sensor Networks," *8th IEEE Annual Communication Networks and Services Research Conference*, Montreal, 11-14 May, pp. 168- 173

Norouzi, A., Zaim, A.H., 2013, Genetic Algorithm Application in Optimization of wireless sensor networks, The Scientific World Journal, Hindawi Publisher

Norouzi, A., Babamir, F.S., Zaim, A.H., 2013, An Interactive Genetic Algorithm for Mobile Sensor Networks, *Studies in Informatics and Control*, ISSN 1220-1766, vol. 22 (2), pp. 213-218

Orman, Z., Norouzi, A., Babamir, F.S. Intelligent Node Placement using GA (INPGA) Protocol in Wireless Sensor Networks, in *Proceedings of the International Conference on Wireless Networks (ICWN'11)* ISBN #: 1-60132-187-2

Norouzi, A., Babamir, F. and Zaim, A., 2011, "A New Clustering Protocol for Wireless Sensor Networks Using Genetic Algorithm Approach," *Wireless Sensor Network*, Vol. 3 No. 11, pp. 362-370. doi: 10.4236/wsn.2011.311042.

Norouzi, A., Babamir, F. and Orman, Z., 2012, "A Tree Based Data Aggregation Scheme for Wireless Sensor Networks Using GA," *Wireless Sensor Network*, Vol. 4 No. 8, pp. 191-196. doi: 10.4236/wsn.2012.48028.

Babamir, F. S., Hatamizade, A., Babamir, S. M., Dabbaghian, M. and Norouzi, A., 2010, "Application of genetic algorithm in automatic software testing," in *Second International Conference on Network Digital Technology*, Charles University, Prague, Czech Republic, vol. 2, pp. 545–552, July

Norouzi, A. and Sertbas, A., 2011, "An integrated survey in efficient energy management for KAA using architecture approach," *International Journal of Advanced Networking and Applications*, vol. 3, no. 1, pp. 968–977