

**GEZGİN TASARSIZ AĞLARDA AĞIRLIK TABANLI OPTİMİZE
EDİLMİŞ KÜME BAŞI SEÇİMİ VE HİBRİT ADAPTİF
KÜMELEME YAKLAŞIMI**

KUDRET YILMAZ

**DOKTORA TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**DANIŞMAN
PROF. DR. RESUL KARA**

DÜZCE, 2025

T.C.
DÜZCE ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

GEZGİN TASARSIZ AĞLARDA AĞIRLIK TABANLI OPTİMİZE
EDİLMİŞ KÜME BAŞI SEÇİMİ VE HİBRİT ADAPTİF
KÜMELEME YAKLAŞIMI

Kudret YILMAZ tarafından hazırlanan tez çalışması aşağıdaki jüri tarafından Düzce Üniversitesi Lisansüstü Eğitim Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda **DOKTORA TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Prof. Dr. Resul KARA

Düzce Üniversitesi

Jüri Üyeleri

Prof. Dr. Resul KARA

Düzce Üniversitesi

Prof. Dr. Ali ÇALHAN

Düzce Üniversitesi

Prof. Dr. Mehmet ŞİMŞEK

Milli Savunma Üniversitesi

Doç. Dr. Murat İSKEFİYELİ

Sakarya Üniversitesi

Doç. Dr. Ferzan KATIRCIOĞLU

Düzce Üniversitesi

Tez Savunma Tarihi: 16/05/2025

BEYAN

Bu tez çalışmasının kendi çalışmam olduğunu, tezin planlanmasından yazımına kadar bütün aşamalarda etik dışı davranışımın olmadığını, bu tezdeki bütün bilgileri akademik ve etik kurallar içinde elde ettiğimi, bu tez çalışmasıyla elde edilmeyen bütün bilgi ve yorumlara kaynak gösterdiğimi ve bu kaynakları da kaynaklar listesine aldığımı, yine bu tezin çalışılması ve yazımı sırasında patent ve telif haklarını ihlal edici bir davranışımın olmadığını beyan ederim.

16 Mayıs 2025

Kudret YILMAZ

TEŐEKKÜR

Doktora öğrenimimde ve bu tezin hazırlanmasında gösterdiği her türlü destek ve yönlendirici fikirleriyle tecrübelerini hiçbir zaman esirgmeden benimle paylaşan çok değerli danışman hocam **Prof. Dr. Resul KARA**'ya en içten dileklerle teşekkür ederim.

Tezimin uygulama ve analiz süreçlerinde yardımlarını esirgemeyen kıymetli hocam **Doç. Dr. Ferzan KATIRCIOĞLU**'na gönülden teşekkür ederim.

Hayatım boyunca arkamda sessiz ama sarsılmaz bir destekle dimdik duran, her zaman dualarını ve inançlarını yanımda hissettiğim canım annem **Hamiyet YILMAZ** ve babam **İsmail YILMAZ**'a sonsuz teşekkür ederim.

Doktora sürecinde zaman zaman benden çok daha sabırlı davranan, anlayışıyla bana nefes aldırarak sevgili eşim **Feride YILMAZ**'a, varlıklarıyla hayatıma neşe ve anlam katan canım oğullarım **Yusuf Hamza YILMAZ** ve **Tuna İsmail YILMAZ**'a en içten sevgilerimle teşekkür ederim.

16 Mayıs 2025

Kudret YILMAZ

İÇİNDEKİLER

Sayfa No

ŞEKİL LİSTESİ.....	vii
ÇİZELGE LİSTESİ.....	viii
KISALTMALAR.....	ix
ÖZET	xi
ABSTRACT	xii
EXTENDED ABSTRACT.....	xiii
1. GİRİŞ.....	1
1.1. PROBLEMİN TANIMI.....	3
1.2. DİNAMİK MANET'LER İÇİN KÜMELEME ALGORİTMALARI VE LİTERATÜRDE YER ALAN GELİŞTİRME ÇALIŞMALARI	4
1.3. TEZİN AMACI.....	12
1.4. TEZİN KAPSAMI	12
1.5. BİLİME KATKISI.....	13
1.6. TEZ ORGANİZASYONU.....	14
2. MANET'LERDE YÖNLENDİRME VE KÜMELEME YAKLAŞIMLARI.....	16
2.1. MANET'LERDE YÖNLENDİRME PROTOKOLLERİ.....	16
2.1.1. Düz Yönlendirme Protokolleri.....	17
2.1.1.1. Proaktif (Tabloya Dayalı) Protokoller.....	17
2.1.1.1.1. DSDV.....	17
2.1.1.1.2. WRP.....	19
2.1.1.1.3. OLSR.....	20
2.1.1.1.4. STAR.....	21
2.1.1.2. Reaktif (İsteğe Bağlı) Protokoller.....	22
2.1.1.2.1. AODV.....	23
2.1.1.2.2. DSR.....	24
2.1.1.2.3. TORA.....	25
2.1.2. Hibrit Yönlendirme Protokolleri.....	28
2.1.2.1. Hiyerarşik Yönlendirme Protokolleri.....	28
2.1.2.1.1. HSR.....	28
2.1.2.1.2. CEDAR.....	30
2.1.2.1.3. CGSR.....	32
2.1.2.2. Konum Tabanlı Yönlendirme Protokolleri.....	34
2.1.2.2.1. LAR.....	34
2.1.2.2.2. DREAM.....	36
2.1.2.2.3. GPSR.....	37
2.2. MANETLERDE KÜMELEME.....	43
2.2.1. Tek Metrik Kümeleme Algoritmaları	46
2.2.1.1. LID.....	47
2.2.1.2. HCC.....	48
2.2.1.3. k-KONID.....	48
2.2.1.4. LEACH.....	49
2.2.2. Ağırlığa Dayalı Kümeleme Algoritmaları.....	50
2.2.2.1. WCA.....	51

2.2.2.2. WBACA	51
2.2.2.3. EE-WCA	53
2.2.2.4. OSCA	56
2.2.3. Optimizasyon Tabanlı Kümeleme Algoritmaları.....	57
2.2.3.1. MOPSO	58
2.2.3.2. CACONET	59
2.2.3.3. HFAPSO	61
2.2.3.4. F-CAPSO	63
2.2.4. Makine Öğrenmesi Tabanlı Kümeleme Algoritmaları.....	65
2.2.4.1. GAHC.....	66
2.2.4.2. E-MAVMMF	68
2.2.4.3. MEAFN-DBA	69
2.2.4.4. MKMPE	71
3. DİNAMİK MANETLER İÇİN HİBRİT ADAPTİF KÜMELEME ALGORİTMASI	77
3.1. EN İYİ KB'Yİ SEÇMEK İÇİN GSA- WCA ENTEGRASYONU	77
3.1.1. WCA'ya Genel Bakış	78
3.1.2. GSA'ya Genel Bakış.....	80
3.2. KÜMELEME VE DÜĞÜM ROLLERİNİN BELİRLENMESİ İÇİN GELİŞTİRİLMİŞ DBSCAN ALGORİTMASI.....	86
4. BULGULAR VE TARTIŞMA	94
4.1. SİMÜLASYON ORTAMI.....	94
4.2. DÜĞÜMLERİN SAHİP OLMASI GEREKEN ÖZELLİKLER VE SİMÜLASYON KISITLARI	96
4.3. DMHAKA -WCA KARŞILAŞTIRMASININ PERFORMANS KRİTERLERİ.....	97
4.3.1. DMHAKA Yönteminin WCA ile Karşılaştırmalı Performans Analizi... 98	
4.3.1.1. 20 düğümlü simülasyon çalışmasının performans karşılaştırması	99
4.3.1.2. Farklı düğüm sayılarına göre performans karşılaştırması	103
4.3.2. DMHAKA Yönteminin Güncel Kümeleme Algoritmaları ile Karşılaştırmalı Performans Analizi	109
4.3.2.1. Veri aktarım kapasitesine (Throughput) göre performans analizi.....	109
4.3.2.2. Ortalama kalan enerjiye göre performans analizi.....	111
4.3.2.3. Paket teslim oranına göre performans analizi	112
4.3.2.4. Ortalama uçtan uca gecikmeye göre performans analizi	114
5. SONUÇLAR VE ÖNERİLER.....	118
5.1. SONUÇLAR.....	118
5.2. ÇALIŞMANIN GETİRDİĞİ KATKILAR.....	119
5.3. TARTIŞMALAR VE ÖNERİLER.....	120
6. KAYNAKLAR	123
ÖZGEÇMİŞ.....	135

ŞEKİL LİSTESİ

Sayfa No

Şekil 1.1. Altyapılı kablosuz ağlar gösterimi.....	1
Şekil 1.2. MANET gösterimi.....	2
Şekil 2.1. Yönlendirme stratejisine göre bazı MANET yönlendirme protokolleri.....	16
Şekil 2.2. DSDV protokolü ve yönlendirme tablosu [45].....	18
Şekil 2.3. Düğüm 3 ile düğüm 6 arasındaki yolun kopması durumu ve yönlendirme tablosu [45].....	18
Şekil 2.4. WRP deki Yönlendirme tablosu [45].	19
Şekil 2.5. OLSR’de MPR seçiminin bir örneği [48]. (a) ağın klasik flooding mekanizması ile düğüm sayısı kadar iletim alması. (b) MPR şemasını kullanarak tüm ağın altı iletimle doldurulması.....	20
Şekil 2.6 STAR protokolü genel yapısı [39].....	21
Şekil 2.7. AODV protokolünün genel yapısı [40].	23
Şekil 2.8. DSR protokolünün genel yapısı [45].	25
Şekil 2.9. TORA’ da rota oluşturma [57].	27
Şekil 2.10. HSR protokol yapısı [45].....	29
Şekil 2.11. HSR paket gönderme [61].	29
Şekil 2.12. CEDAR’da rota kurulumu [45].....	31
Şekil 2.13. CEDAR’ da rota bakımı [45].....	32
Şekil 2.14. CGSR protokolü paket iletimi [45].	33
Şekil 2.15. (a) LAR protokolünün bölgeleri ve LAR 1 yönlendirme durumu.....	35
Şekil 2.16. DREAM algoritmasının mesafe etkisi, [74].	37
Şekil 2.17. GPSR Paket iletim sistemi [69].	38
Şekil 2.18. MANET’te düğüm rolleri gösterimi.....	43
Şekil 2.19. MANET’te kümeleme yöntemleri [7].	45
Şekil 2.20. MANET’te kümeleme algoritmalarının sınıflandırılması.	46
Şekil 3.1. v düğümünün belirli zamanlardaki gösterimi [16].....	79
Şekil 3.2. GSA’nın akış diyagramı.	84
Şekil 3.3. GSA’nın yakınsama grafiği.	85
Şekil 3.4. DBSCAN kümeleme örneği [100].	88
Şekil 3.5. DBSCAN algoritmasında örtüşen kümeler.	89
Şekil 3.6. KB iletim aralığı ve eps mesafesi.	90
Şekil 3.7. Önerilen yöntemin akış diyagramı.	91
Şekil 3.8. Önerilen yönteme göre kümeler ve düğüm rolleri örneği.	93
Şekil 4.1. 20 düğümlü simülasyon gösterimi.....	99
Şekil 4.2. 20 düğümlü senaryoda KB değişim oranı.	100
Şekil 4.3. 20 düğümlü senaryo için ortalama KB ömrü.....	101
Şekil 4.4. 20 düğümlü senaryoda ortalama kalan enerjinin zamana göre değişimi.....	102
Şekil 4.5. Farklı düğüm sayılarına göre ortalama KB değişim oranı.	103
Şekil 4.6. Farklı düğüm sayılarına göre ortalama KB ömrü.	105
Şekil 4.7. Farklı düğüm sayılarına göre ortalama küme ömrü.....	106
Şekil 4.8. Farklı düğüm sayılarına göre ortalama kalan enerji.	108
Şekil 4.9. Düğüm sayısına göre veri aktarım kapasitesi karşılaştırma grafiği.....	110
Şekil 4.10. Düğüm sayısına göre kalan enerji miktarlarının karşılaştırılma grafiği.	112
Şekil 4.11. Düğüm sayısına göre paket teslim oranı grafiği.	113
Şekil 4.12. Düğüm sayısına göre uçtan uca gecikme grafiği.....	115

ÇİZELGE LİSTESİ

	<u>Sayfa No</u>
Çizelge 2.1. Yönlendirme protokollerinin özeti, avantaj ve dezavantajları.....	39
Çizelge 2.2. Kümeleme algoritmalarının özeti, avantaj ve dezavantajları.	73
Çizelge 4.1. Kullanılan bilgisayarın sistem özellikleri.	95
Çizelge 4.2. Simülasyonlarda kullanılan parametreler.	96
Çizelge 4.3. Farklı düğüm sayılarında DMHAKA yönteminin diğer yöntemlere göre iyileştirme oranları.	116



KISALTMALAR

ACO	Karınca Kolonisi Optimizasyonu
AD	Ayrık Düğüm
ANN	Yapay Sinir Ağı
AODV	Tasarsız İsteğe Bağlı Uzaklık Vektörü
AOVA	Afrika Akbaba Optimizasyon Algoritması
BD	Bireysel Düğüm
BS	Baz İstasyonu
CACONET	Karınca Kolonisi Optimizasyonuna Dayalı Kümeleme Algoritması
CBR	Sabit Bit Hızı
CEDAR	Çekirdek Tabanlı Dağıtık Ad Hoc Yönlendirme
CDMA	Kod Bölmeli Çoklu Erişim
CGSR	Küme Ağ Geçidi Anahtar Yönlendirme Protokolü
CS	Kritik Bölge
DAG	Yönlendirilmiş Döngüsüz Grafik
DBSCAN	Gürültülü Veriler İçin Yoğunluk Tabanlı Uzamsal Kümeleme
DSDV	Hedef Sıralamalı Uzaklık Vektörü
DS	Dominator Küme
DSR	Dinamik Kaynak Yönlendirme
DT	Mesafe Tablosu
DMHAKA	Dinamik MANET'ler için Hibrit Adaptif Kümeleme Algoritması
DREAM	Hareketlilik Etkili Uzaklık Yönlendirme Algoritması
EE-WCA	Enerji Verimli Ağırlıklı Kümeleme Algoritması
EPS	Epsilon Değeri
EZ	Beklene Bölge
E-MAVMMF	Enerji Farkındalıklı Çok Özellikli Oylama Tabanlı Çok Ölçütlü Fonksiyon
FA	Ateşböceği Algoritması
FANET	Uçan Ad-Hoc Ağı
FCM	Bulanık C-Ortalamalar
FTP	Dosya Transfer Protokolü
F-CAPSO	Bulanık Kaotik Uyarlanabilir Parçacık Sürü Optimizasyonu
GA	Genetik Algoritma
GAHC	Tepe Tırmanışı ile Genetik Algoritma
GD	Geçityolu Düğümü
GİD	Geçityolu İstek Düğümü
GPS	Küresel Konumlandırma Sistemi
GPSR	Kablosuz Ağlar için Açgözlü Çevre Durumsuz Yönlendirme
GSA	Yerçekimi Arama Algoritması
HCC	En Yüksek Bağlantılılık Kümeleme Algoritması
HFAPSO	Hibrit Ateşböceği ve Parçacık Sürü Optimizasyon Algoritması
HSR	Hiyerarşik Durum Yönlendirme
IDT	Dolaylı Güven
İHA	İnsansız Hava Aracı
KB	Küme Başı
KÜ	Küme Üyesi
K-KONID	K-Komşuluk Kümeleme Algoritması

LAR	Konum Destekli Yönlendirme
LEACH	Düşük Enerjili Uyumlu Kümeleme Hiyerarşisi
LID	En Düşük Kimlik Kümeleme
LLC	Mantıksal Bağlantı Kontrolü
MAC	Ortam Erişim Kontrolü
MANET	Mobil Ad-Hoc Ağı
MEAFN-DBA	Uçan Ağlar için Yoğunluk Tabanlı Karşılıklı Dışlama Algoritması
MinPts	Minimum Nokta Sayısı
MKM	Modifiye Edilmiş K-Means Algoritması
MKMPE	Modifiye Edilmiş K-Ortalamaları Filipin Kartalı Algoritması
MMA	Modifiye Edilmiş Mayıs Sineği Algoritması
MOPSO	Çok Amaçlı Parçacık Sürü Optimizasyonu
MORS-ASO	Çok Amaçlı Güvenli ve Güven Farkındalıklı Yedekli Dizi Kaydırmalı Şifreleme için Uyarlanabilir Serçe Arama Optimizasyonu
MPR	Çoklu Yayın Yönlendirici
MRL	Mesaj Yeniden İletim Listesi
NS-2	Ağ Simülatörü-2
OLSR	Optimize Bağlantı Durum Yönlendirme
OSCA	Optimize Edilmiş Stabil Kümeleme Algoritması
PEO	Filipin Kartalı Optimizasyonu
PSO	Parçacık Sürü Optimizasyonu
QoS	Hizmet Kalitesi
RT	Yönlendirme Tablosu
RERR	Rota Hatası
RREP	Rota Yanıtı
RREQ	Rota Talebi
SKA	Sinüs Kosinüs Algoritması
SNR	Sinyal-Gürültü Oranı
SSA	Salp Sürüsü Algoritması
STAR	Kaynak Ağaç Adaptif Yönlendirme
SVM	Destek Vektör Makineleri
TCP	İletim Kontrol Protokolü
TELNET	Telekomünikasyon Ağı Üzerinden Terminal Erişim Protokolü
TORA	Geçici Sıralı Yönlendirme Algoritması
TR	İletim Aralığı
TSDR	Güvenli Dinamik Kaynak Yönlendirme
UDP	Kullanıcı Datagram Protokolü
ÜD	Üye Düğüm
VANET	Araç Ad-Hoc Ağı
WBACA	Ağırlık Tabanlı Adaptif Kümeleme Algoritması
WCA	Ağırlıklı Kümeleme Algoritması
WRP	Kablosuz Yönlendirme Protokolü
WSN	Kablosuz Sensör Ağı

ÖZET

GEZGİN TASARSIZ AĞLARDA AĞIRLIK TABANLI OPTİMİZE EDİLMİŞ KÜME BAŞI SEÇİMİ VE HİBRİT ADAPTİF KÜMELEME YAKLAŞIMI

Kudret YILMAZ

Düzce Üniversitesi

Lisansüstü Eğitim Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalı

Doktora Tezi

Danışman: Prof. Dr. Resul KARA

Mayıs 2025, 134 sayfa

Mobil ad-hoc ağlar (MANET), düğümlerin merkezi bir yönlendirme altyapısı olmaksızın rastgele hareket ettiği, dinamik ve kendini yöneten kablosuz ağlardır. Bu yapının en önemli sorunlarından biri, düğümlerin enerji tüketiminin artmasıyla ağ ömrünün kısılması ve veri iletiminde güvenilirliğin azalmasıdır. Bu nedenle MANET ortamlarında etkili bir kümeleme stratejisi, ağ performansının sürdürülebilirliği açısından kritik öneme sahiptir. Bu tez çalışmasında, MANET ortamlarında ağ performansını artırmak amacıyla Dinamik MANET'ler için Hibrit Adaptif Kümeleme Algoritması (DMHAKA) önerilmiştir. İki aşamalı yapıda tasarlanan algoritmanın ilk aşamasında, Küme Başı (KB) seçiminde düğüm derecesi, komşuluk mesafesi, kalan enerji ve hareketlilik gibi kriterler Ağırlıklı Kümeleme Algoritması (Weighted Clustering Algorithm -WCA) temel alınarak belirlenmiş ve Yerçekimi Arama Algoritması (Gravitational Search Algorithm -GSA) ile optimize edilmiştir. İkinci aşamada ise seçilen KB'ler etrafında düğüm rollerinin belirlenmesinde, parametre bağımlılığını azaltan ve daha dengeli kümeler oluşturan Geliştirilmiş Yoğunluk Tabanlı Kümeleme Algoritması (Geliştirilmiş-DBSCAN) kullanılmıştır. Simülasyon sonuçlarında DMHAKA'nın ağ ömrünü uzattığı ve paket teslim oranını artırdığı gözlemlenmiştir. WCA ile yapılan karşılaştırmalı analizlerde, DMHAKA'da KB değişim oranı %40 daha düşük, ortalama KB ömrü ise yaklaşık %60 daha uzun hesaplanmıştır. Ayrıca, ortalama küme ömrü değerleri DMHAKA'nın daha uzun süre stabil kümeler oluşturduğunu ve bu yapının enerji tasarrufu ile ağ ömrüne katkı sağladığını göstermiştir. Bu sonuçlar, DMHAKA'nın daha az yeniden yapılandırma gerektiren, kararlı bir küme yapısı sunduğunu ortaya koymaktadır. Ayrıca, EE-WCA, E-MAVMMF, TSDR ve MORS-ASO algoritmalarıyla yapılan karşılaştırmalı analizlerde; ortalama kalan enerji, uçtan uca gecikme, paket teslim oranı ve veri aktarım kapasitesi gibi temel metriklerde DMHAKA'nın üstün performans sergilediği tespit edilmiştir. Elde edilen bulgular, DMHAKA'nın çeşitli kümeleme algoritmalarıyla karşılaştırıldığında, ölçeklenebilir, kararlı ve enerji farkındalığı yüksek bir kümeleme yaklaşımı sunduğunu göstermektedir. Bu yönüyle DMHAKA, dinamik MANET koşullarında etkin veri iletimi sağlamak ve ağ performansını uzun vadede sürdürülebilir kılmak açısından güçlü bir alternatif olarak değerlendirilebilir.

Anahtar Sözcükler: Adaptif Kümeleme, Enerji Verimliliği, Makine Öğrenmesi, MANET, Yerçekimi Arama Algoritması.

ABSTRACT

WEIGHTED-BASED OPTIMIZED CLUSTER HEAD SELECTION AND HYBRID ADAPTIVE CLUSTERING APPROACH IN MOBILE AD-HOC NETWORKS

Kudret YILMAZ

Düzce University

Graduate School, Department of Computer Engineering

Doctoral Thesis

Supervisor: Prof. Dr. Resul KARA

May 2025, 134 pages

Mobile ad-hoc networks (MANETs) are dynamic and self-organizing wireless networks in which nodes move randomly without relying on centralized routing infrastructure. One of the major challenges in such networks is the increased energy consumption of participating nodes, which shortens the overall network lifetime and leads to instability in packet delivery. Therefore, an effective clustering strategy in MANETs is critically important for sustaining network performance. In this thesis, a Hybrid Adaptive Clustering Algorithm for Dynamic MANETs (DMHAKA) is proposed to enhance performance in dynamic MANET environments. The algorithm is designed in two stages. In the first stage, cluster head (CH) selection is performed based on criteria such as node degree, neighborhood distance, remaining energy, and mobility, using the Weighted Clustering Algorithm (WCA) as a basis. These criteria are then optimized using the Gravitational Search Algorithm (GSA). In the second stage, node roles around the selected CHs are determined using the Enhanced Density-Based Spatial Clustering of Applications with Noise (Enhanced-DBSCAN) algorithm. This approach reduces parameter dependency and allows the formation of more flexible and balanced clusters. Simulation results show that DMHAKA significantly extends network lifetime and improves the packet delivery ratio. In comparative analyses with the WCA algorithm, the CH change rate in DMHAKA was found to be 40% lower, and the average CH lifetime was approximately 60% longer. Moreover, the average cluster lifetime results indicate that DMHAKA forms more stable clusters, directly contributing to energy savings and longer network life. These results reveal that DMHAKA provides a stable clustering structure that requires fewer reconfigurations. In addition, comparative analyses with the EE-WCA, E-MAVMMF, TSDR, and MORS-ASO algorithms demonstrated that DMHAKA exhibits superior performance in key metrics such as average remaining energy, end-to-end delay, packet delivery ratio, and throughput. The findings indicate that DMHAKA offers a scalable, stable, and energy-aware clustering approach when compared with various clustering algorithms. In this regard, DMHAKA can be considered a strong alternative for enabling efficient data transmission and achieving long-term sustainable network performance under dynamic MANET conditions.

Keywords: Adaptive Clustering, Energy Efficiency, Machine Learning, MANET, Gravitational Search Algorithm.

EXTENDED ABSTRACT

WEIGHTED-BASED OPTIMIZED CLUSTER HEAD SELECTION AND HYBRID ADAPTIVE CLUSTERING APPROACH IN MOBILE AD-HOC NETWORKS

Kudret YILMAZ

Düzce University

Graduate School, Department of Computer Engineering

Doctoral Thesis

Supervisor: Prof. Dr. Resul KARA

May 2025, 134 pages

1. INTRODUCTION

Mobile Ad-hoc Networks (MANETs) are decentralized, self-organizing wireless systems formed by mobile nodes that communicate without the need for fixed infrastructure. These networks have become increasingly prominent in various domains such as disaster recovery, military operations, smart transportation systems, and mobile device networks. Despite their advantages, MANETs face critical challenges such as dynamic topologies, energy limitations, routing instability, and low packet delivery rates.

To address these challenges, effective clustering strategies are essential for improving network performance and ensuring reliable data transmission. Clustering involves grouping nodes into virtual clusters where certain nodes serve as Cluster Heads (CHs) to manage communication and coordination. However, the effectiveness of this process heavily depends on the accuracy of CH selection, which plays a key role in maintaining stability and optimizing energy usage.

This study proposes a two-phase hybrid clustering mechanism to enhance performance in dynamic MANET environments. In the first phase, cluster head selection is conducted by evaluating criteria such as node degree, neighborhood distance, residual energy, and mobility, based on the Weighted Clustering Algorithm (WCA), and then optimized using the Gravitational Search Algorithm (GSA). In the second phase, an Enhanced Density-Based Spatial Clustering of Applications with Noise (Enhanced-DBSCAN) algorithm an improved version of the classical DBSCAN is used to assign roles to nodes around the selected CHs. Unlike traditional DBSCAN, the enhanced version adaptively calculates the eps parameter based on local node density, enabling more stable, balanced, and flexible clusters suited to the dynamic topology of MANETs.

2. MATERIAL AND METHODS

This paper focuses on developing a two-stage hybrid clustering method to create more stable and energy-efficient clustering structures in dynamic MANET environments. In the first stage, the CH selection process is based on the WCA. Four key metrics node degree, neighborhood distance, remaining energy, and mobility are considered in the CH selection. These metrics are optimized using the Gravitational Search Algorithm (GSA), enabling a multi-criteria optimization approach to achieve a more balanced CH distribution.

In the second stage, the Enhanced-DBSCAN algorithm, which was specifically developed to overcome certain limitations of the classical DBSCAN, is used for assigning node roles. Unlike the traditional version, the enhanced model adaptively calculates the eps parameter for each CH based on the local node density. This removes the need for manual parameter tuning and allows for better adaptability in dynamic network topologies. Additionally, to reduce the communication load on CHs and improve inter-cluster communication, gateway nodes are designated further strengthening the structural stability of the network.

For routing, the Ad-hoc On-Demand Distance Vector (AODV) protocol was utilized. The standard routing packets such as RREQ, RREP, and RERR were modified to support the clustered network structure. In particular, new data fields were added to enhance routing via gateway nodes, enabling more efficient inter-cluster communication.

Simulations were carried out using the NS-2 simulator in scenarios where the number of nodes ranged from 20 to 100. The node mobility speed was randomly assigned between 2–10 m/s, and each simulation lasted 100 seconds. The simulation parameters are detailed in section 4.1. The primary performance metrics used in this study include average remaining energy, CH change rate, average CH lifetime, average cluster lifetime, throughput, packet delivery ratio, and average end-to-end delay.

3. RESULTS AND DISCUSSIONS

The simulation results have demonstrated that the proposed DMHAKA method significantly improves network performance across multiple metrics, especially when compared to both the classical WCA algorithm and more recent clustering approaches such as EE-WCA, E-MAVMMF, TSDR, and MORS-ASO. The evaluation was conducted based on throughput, packet delivery ratio, average residual energy, end-to-

end delay, CH lifetime, cluster stability, and CH change rate.

Compared to WCA, DMHAKA achieved a 40% reduction in CH change rate, which directly reflects enhanced cluster stability. The average CH lifetime increased by approximately 60%, and the average cluster lifetime also showed substantial improvement, particularly in high-density scenarios. These enhancements indicate a reduction in re-clustering operations and better resource conservation throughout the network.

Regarding energy performance, in simulations with 100 nodes, the average remaining energy with DMHAKA was 86 Joules, while WCA resulted in approximately 55 Joules, confirming that the DMHAKA method ensures more balanced energy consumption and extends network lifetime.

In comparative evaluations with modern algorithms, DMHAKA delivered consistent performance gains. For instance:

- Residual energy: 46.38% higher than EE-WCA, 18.35% over E-MAVMMF, 13.08% above TSDR, and 8.33% greater than MORS-ASO.
- Throughput improvements reached up to 46.32% compared to EE-WCA and averaged over 30% for most scenarios.
- Packet delivery ratio and end-to-end delay metrics also showed substantial enhancements, particularly when node density increased.

These results underscore DMHAKA's scalability and adaptability. Its stability and energy-aware design make it a reliable alternative for real-world MANET deployments. The method's ability to maintain performance under dynamic topologies and dense node distributions reinforces its robustness and efficiency as a clustering solution in mobile ad hoc networks.

4. CONCLUSION AND OUTLOOK

This study proposed a two-stage clustering algorithm named DMHAKA to address the challenges of energy efficiency, stability, and scalability in MANETs. The proposed method integrates a weighted CH selection process optimized via the GSA and an enhanced version of the DBSCAN algorithm for adaptive and flexible cluster formation. Simulation results demonstrated that the DMHAKA method significantly improves both structural stability and energy efficiency compared to classical and contemporary

clustering algorithms.

The comparative performance analyses revealed that DMHAKA reduces the CH change rate by approximately 30%, increases the average CH lifetime by up to 60%, and extends the average cluster duration considerably compared to the classical WCA algorithm. Furthermore, evaluations conducted against EE-WCA, E-MAVMMF, TSDR, and MORS-ASO algorithms showed that DMHAKA yields higher performance in terms of average residual energy, end-to-end delay, packet delivery ratio, and throughput, particularly in high-density dynamic network environments.

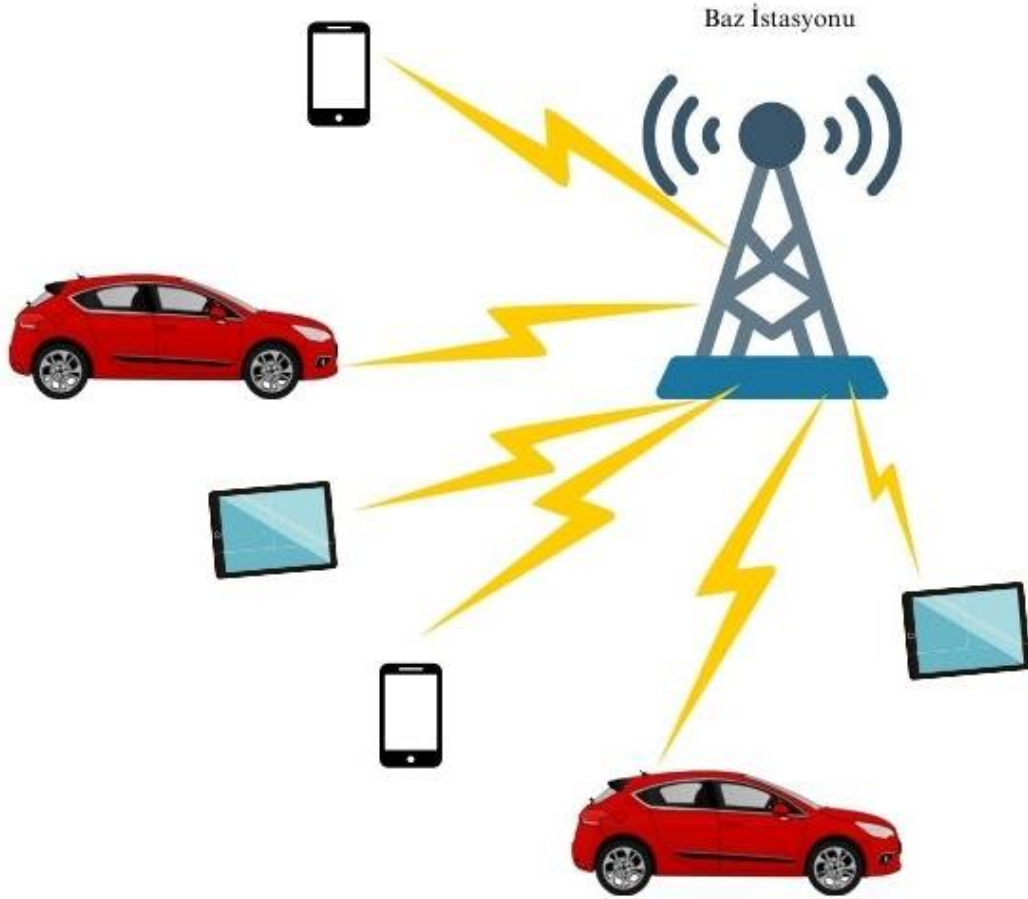
In addition to these improvements, the study also introduced a novel optimization framework that dynamically adapts the DBSCAN algorithm parameters based on local node density, eliminating the need for manual tuning. This contribution enhances the robustness of the clustering mechanism under varying network conditions.

Despite these strengths, the study has certain limitations and offers directions for future work. First, the proposed method was tested only in a simulation environment; its implementation on real devices or embedded systems will further validate its practical applicability. Second, only the random waypoint mobility model was used; future studies may consider different mobility scenarios. Third, new optimization targets such as load balancing, bandwidth utilization, and link lifetime could be included in future evaluations. Lastly, integration with deep learning-based routing protocols and advanced gateway selection techniques may enhance decision-making in dynamic scenarios.

Overall, this research provides an innovative and scalable solution for clustering in MANETs, offering a promising foundation for further studies in wireless network optimization.

1. GİRİŞ

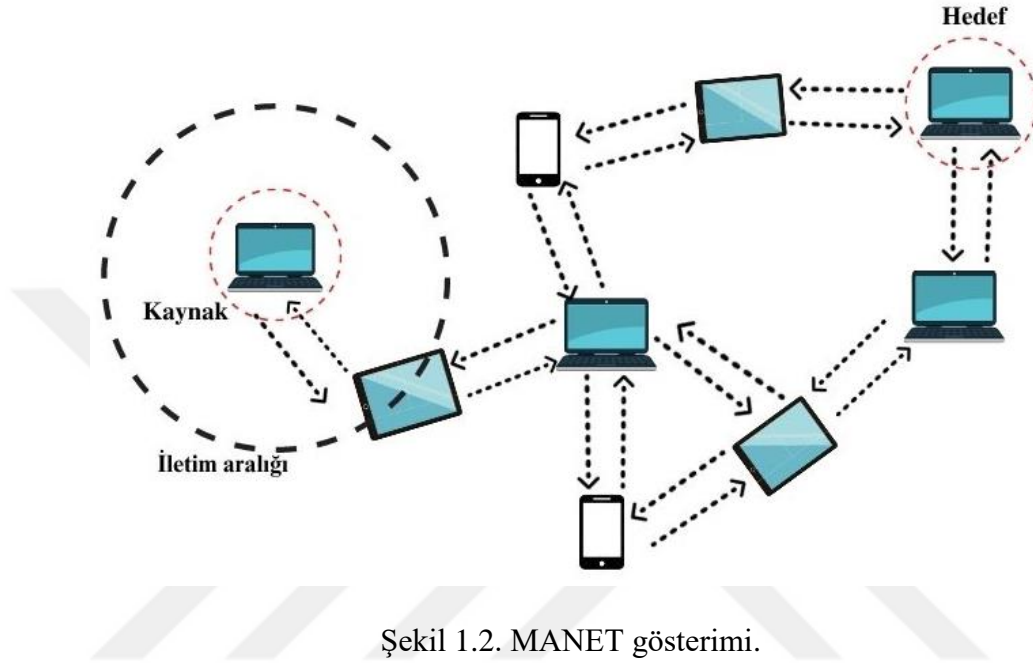
Kablosuz ađ teknolojisi, ilk olarak 1970'li yıllarda ortaya çıkmıř ve günümüze kadar sürekli geliřerek iletiřim alanında yaygın bir kullanım kazanmıřtır. Bu ađlar, altyapısına göre "altyapılı" ve "altyapısız" olmak üzere iki gruba ayrılmaktadır. Altyapılı kablosuz ađlarda iletiřim, Őekil 1.1'deki gibi sabit baz istasyonları ve eriřim noktaları aracılıđıyla sađlanır. Bu istasyonlar, ađdaki cihazlar arasında veri paketlerinin yönlendirilmesinden ve yönetilmesinden sorumludur [1].



Őekil 1.1. Altyapılı kablosuz ađlar gösterimi.

Ancak altyapısız kablosuz ađlarda sabit bir yönlendirici bulunmadıđından, her cihaz veri iletimi ve yönlendirme görevini kendisi üstlenir. Bu tür altyapısız ađların en yaygın örneklerinden biri Mobil Ad-Hoc Ađlar (Mobile Ad-Hoc Network- MANET)'dir.

MANET, Şekil 1.2’de gösterildiği gibi sabit bir altyapıya ihtiyaç duymadan, hareketli kablosuz cihazların bir araya gelerek oluşturduğu geçici ağ yapısını ifade etmektedir. Merkezi bir erişim noktası kullanılmadığı için, ağdaki her düğüm hem istemci hem de yönlendirici olarak işlev görür. Bu düğümler, diğer düğümlerden gelen verileri ileterek ağın sürekliliğini sağlar.



Şekil 1.2. MANET gösterimi.

Ad-Hoc ağlar ilk olarak askeri uygulamalarda kullanılmış olup, günümüzde ise afet yönetimi, akıllı ulaşım sistemleri ve taşınabilir cihazların iletişiminde yaygın olarak tercih edilmektedir. MANET’ler, herhangi bir sabit yapı gerektirmediği için hızlı kurulabilir ve geniş bir kapsama alanına yayılabilir. Bu ağların düğüm sayısı, birkaç cihazdan birkaç bine kadar çıkabilmektedir. MANET’lerde düğümlerin yüksek hareket kabiliyeti, ağ topolojisinin sürekli ve dinamik bir şekilde değişmesine yol açmaktadır. Bu değişim oranı, düğümlerin hareket hızına ve iletim gücüne bağlıdır. Ayrıca, bu ağlarda cihazlar genellikle küçük boyutlu olup sınırlı bir iletim gücüne sahiptir. Düşük iletim gücü, her bir düğümün yalnızca belirli bir mesafedeki komşu düğümlerle iletişim kurabilmesine neden olur ve bu durum, ağ topolojisinin düğüm hareketleriyle birlikte daha hızlı değişmesine neden olmaktadır [2], [3].

1.1. PROBLEMİN TANIMI

MANET sistemlerinde karşılaşılan zorluklar aşağıda listelenmiştir [4], [5].

- D ğ mlerin hareketli olması: MANET’lerdeki d ğ mler, ihtiyalarına baėlı olarak farklı hız ve y nlerde rastgele hareket edebilmektedir. Bu nedenle aė topolojisi kısa s reli ve  ng r lemez bir yapıya sahiptir. İletiřim iin kullanılan yollar, s rekli ve hızlı bir řekilde deėiřiklik g sterebilir.  zellikle, komřu d ğ mlerin sinyal menzili dıřına ıkması durumunda mevcut yollar kesintiye uėrayabilir. Bu durum, y nlendirme iřlemini karmařık hale getirerek d ř k paket teslim oranlarına yol amaktadır.
- Karmařıklık: Aėın daėıtık yapısına baėlı olarak birden fazla teknolojinin, y nlendirme yapısı ve protokol n n olması aėın karmařık bir řemada olmasına sebebiyet verebilmektedir.
-  leklenebilirlik: Aėda ok sayıda d ğ m bulunduėunda, dolařan paketlerin fazlalıėı nedeniyle bazı paketlerin genel kimlik numarası eksik olabilir. Bu durum, aė  zerinde yoėunluk kaynaklı tıkanıklıklara ve veri paketlerinin arpıřmasına neden olabilmektedir.
- Merkezi kontrol birimi olmayıřı: MANET’ler, herhangi bir merkezi kontrol mekanizmasına ihtiya duymadan, kendiliėinden ve plansız bir řekilde aė yapısı oluřturabilmektedir. Aėdaki d ğ mler istedikleri zaman aėa katılabilir veya aėdan ayrılabilirler. Bu dinamik yapı, merkezi bir kontrol birimi oluřturmayı olduka zor hale getirmektedir.
- Kaynakların sınırlı olması: Mobil d ğ mler, kablolu aėlarda kullanılan masa st  bilgisayarlarla kıyasla genellikle daha sınırlı iřlem g c ne, daha az bellek kapasitesine ve kısıtlı enerji kaynaklarına sahiptir. Bu durum d ğ mlerin iletiřimden kopmalarına, veri kayıplarına ve dinamik topoloji deėiřikliklerine yavař yanıt vermelerine sebep olmaktadır.

Bu zorlukların  stesinden gelmek iin MANET’lerde kullanılan y nlendirme algoritmalarının daha verimli hale getirilmesi gerekmektedir.  zellikle, y nlendirme iřlemleri sırasında kontrol ve baėlantı iin harcanan bant geniřliėinin ve buna baėlı enerji t ketiminin minimum seviyeye indirilmesi b y k  nem tařımaktadır.

1.2. DİNAMİK MANET'LER İÇİN KÜMELEME ALGORİTMALARI VE LİTERATÜRDE YER ALAN GELİŞTİRME ÇALIŞMALARI

Bu bölümde, MANET'ler için geliştirilen kümeleme algoritmalarına ilişkin literatürde yer alan temel yaklaşımlar ve bu yaklaşımlara yönelik geliştirme çalışmaları değerlendirilmektedir. Algoritmaların sınıflandırması ve teknik detayları, 2. bölümde tek metrik tabanlı, ağırlığa dayalı, optimizasyon temelli ve makine öğrenmesi temelli olmak üzere dört temel başlık altında sistematik biçimde sunulmuştur. Bu bölümde ise, özellikle literatürde yaygın olarak kullanılan bazı algoritmalara odaklanılarak, bu yaklaşımların güçlü ve zayıf yönleri eleştirel bir bakış açısıyla irdelenmektedir.

Literatürde, geliştirilen kümeleme algoritmaları genellikle karar verme yapılarındaki karmaşıklık düzeylerine ve kullandıkları parametre sayılarına göre sınıflandırılmaktadır. En temel ve erken dönem yaklaşımlar, tek bir ağ parametresine dayalı olarak KB seçimi yapmayı amaçlayan tek metrikli kümeleme algoritmalarıdır. Bu algoritmalar, basit yapıları sayesinde düşük işlem yükü sunsalar da MANET ortamlarının dinamik doğasına yeterince uyum sağlayamadıkları görülmektedir [6], [7].

Tek metrik tabanlı kümeleme algoritmaları, KB seçimini yalnızca tek bir ağ parametresine dayandırarak gerçekleştirmektedir. Bu parametre genellikle düğüm kimliği, kalan enerji miktarı, hareketlilik düzeyi, düğüm derecesi veya iletişim gücü gibi ağın yapısal ve işlevsel özelliklerini yansıtan ölçütlerden biridir. Örneğin, en düşük kimlik numarasına sahip düğümün KB olarak atanması, ağ yapısını basitleştirmek açısından hızlı ve düşük maliyetli bir çözüm sunar. Benzer şekilde, yüksek enerjiye sahip düğümlerin KB olarak seçilmesi, ağ ömrünü uzatmaya yönelik bir stratejidir. Düşük hareketliliğe sahip düğümlerin seçimi ise kümelerin daha kararlı olmasını hedeflemektedir. Bu yöntemlerin temel avantajı, hesaplama açısından basit ve hızlı olmalarıdır. Ancak, yalnızca tek bir metriğe dayalı karar mekanizması, ağın dinamik yapısını yeterince yansıtamamakta ve bu da ağ performansında dengesizliklere yol açabilmektedir [7], [8], [9].

Bu algoritma sınıfına dahil olan En Düşük Kimlik (LID) algoritması, merkezi olmayan yapısıyla, en düşük kimlik (ID) değerine sahip düğümün küme başı olarak seçilmesini esas alır. Düğümler belirli aralıklarla kendi kimliklerini yayınlar ve en düşük ID'ye sahip olan düğüm küme başı olarak atanır. Uygulama kolaylığı ve düşük kontrol yükü gibi avantajlarıyla öne çıkan bu yöntem, yalnızca kimlik bilgisine dayanması nedeniyle ağın

dinamik koşullarına uyum sağlamakta yetersiz kalabilmektedir. Özellikle aynı düğümlerin sürekli KB olarak seçilmesi, bu düğümlerin enerji kaynaklarının hızla tükenmesine ve dolayısıyla ağ performansının bozulmasına yol açar. Ayrıca, yalnızca belirli düğümler üzerinden gerçekleştirilen kümeler arası iletişim, bu düğümlerde yoğun trafik oluşmasına neden olmaktadır [10], [11].

Bu sınırlamaları gidermek amacıyla geliştirilen algoritmalarından biri, En Yüksek Bağlantılılık (HCC) algoritmasıdır. HCC, KB seçiminde yalnızca kimlik numarası yerine düğüm derecesini yani bağlantı sayısını dikkate alarak daha dengeli ve yapısal bir kümeleme yaklaşımı sunar. Her düğüm, belirli aralıklarla kendi kimliğini ve komşularının kimliklerini yayınlamak için lokal topolojiyi tanımlar. En fazla sayıda komşuya sahip olan düğüm KB olarak atanırken, eşit dereceli düğümler arasında en düşük ID'ye sahip olan tercih edilir. Bu yapı, özellikle daha az sayıda KB oluşmasını sağlayarak ağ üzerindeki kontrol yükünü azaltmakta ve ağın ölçeklenebilirliğini artırmaktadır [12]. HCC algoritması, LID'e kıyasla daha dengeli kümeler oluştursa da yüksek hareketlilik durumlarında KB değişimlerinin sıklığı artmakta; bu durum, güncelleme paketlerinin çoğalmasına ve kontrol mesajı trafiğinin yoğunlaşmasına neden olmaktadır. Sonuç olarak, ağın genel enerji verimliliği ve iletişim performansı bu durumdan olumsuz etkilenebilmektedir [9], [12].

LID ve HCC algoritmalarının sunduğu yapısal avantajları bir araya getirme amacıyla geliştirilen algoritmalarından biri de k-Komşuluk Kümeleme Algoritması (k-KONID)'dir. Bu yöntem, KB seçiminde öncelikli kriter olarak düğüm derecesini, ikincil kriter olarak ise kimlik numarasını kullanmaktadır. Böylece bağlantı yoğunluğu dikkate alınarak daha etkin KB seçimi yapılmakta, eşit dereceli durumlarda ise düşük ID'ye sahip düğümler tercih edilmektedir. k-KONID'in en önemli katkılarından biri, kümeleme yapısında k-hop mesafesi kavramını kullanmasıdır. Her KB, kendisine en fazla k-hop uzaklıktaki düğümleri içerecek şekilde kümeler oluşturur ve bu sayede küme sayısı ve küme içi düğüm sayısı optimize edilmeye çalışılmaktadır [13].

Literatürde yapılan karşılaştırmalı çalışmalar, bu yapının LID'e kıyasla daha düşük kontrol yükü oluşturduğunu, HCC'ye göre ise daha kararlı kümeler ürettiğini ortaya koymaktadır. Ancak, k-KONID algoritmasında KB düğümleri daha geniş bir alana hizmet verdiğinden enerji tüketimi hızla artmakta ve bu durum, özellikle enerji kısıtlı ağlarda KB ömrünü olumsuz etkilemektedir. Ayrıca, her düğümün k-hop uzaklıktaki tüm komşularına kümeleme kararlarını iletmesi gerektiğinden, algoritma zaman açısından

verimli bir yapı sunmamaktadır [7].

Geleneksel kümeleme algoritmalarında bağlantı yoğunluğu, kimlik numarası veya k-hop mesafesi gibi yapısal kriterler ön plandayken, enerji verimliliğine dayalı kümeleme ihtiyacı, literatürde farklı bir yönelimi beraberinde getirmiştir. Bu bağlamda geliştirilen en bilinen protokollerden biri olan Düşük Enerjili Uyarlanabilir Kümeleme Hiyerarşisi (LEACH), enerji verimliliğini artırmak amacıyla tasarlanmış bir yönlendirme protokolüdür. LEACH, ağı belirli aralıklarla rastgele seçilen KB'ler tarafından yönetilen kümelere ayırmakta ve her KB, veri toplama ve Baz İstasyonuna (BS) iletim görevlerini üstlenerek düğümler arası enerji tüketimini dengelemeyi hedeflemektedir. Ayrıca, KB'ler veri birleştirme işlemi yaparak hem iletişim maliyetini hem de gecikmesini azaltmakta, aynı zamanda düğümlerin uyku moduna geçmesini sağlayarak ek enerji tasarrufu sunmaktadır [14]. Bununla birlikte, KB'lerin her tur başında rastgele seçilmesi, ek kontrol yükü ve iletişim maliyeti doğurmakta, düşük enerjili düğümlerin KB olarak seçilmesi ise ağın genel performansını olumsuz etkileyebilmektedir. Ayrıca, sürekli veri aktarımı gerçekleştiren KB'lerin enerji kaynakları daha hızlı tükenmekte, bu da ağ ömrünü sınırlamaktadır [15].

Literatürdeki tek metrikli kümeleme algoritmaları, düşük hesaplama maliyeti ve uygulanabilirlik açısından önemli avantajlar sunmuş olsalar da mobil ad-hoc ağların dinamik yapısına uyum sağlama konusunda çeşitli sınırlılıklara sahiptirler. Bu yöntemlerin büyük çoğunluğu yalnızca bir parametreye dayandıkları için ağın genel performansını etkileyen diğer faktörleri göz ardı etmekte ve bu durum KB seçiminde dengesizliklere yol açmaktadır. Özellikle enerji tüketiminin adil dağılmaması, sık KB değişimi, iletişim yükü artışı gibi sorunlar bu algoritmaların sınırlı uygulanabilirliğini ortaya koymuştur. Bu eksiklikleri gidermek amacıyla, literatürde birden fazla parametreyi eş zamanlı değerlendiren ağırlığa dayalı kümeleme algoritmaları geliştirilmiştir. Bu sınıftaki algoritmalar, düğüm hareketliliği, iletim gücü, ideal komşu sayısı ve enerji seviyesi gibi birden çok metriktan elde edilen ağırlıklı skorlar üzerinden KB seçimini gerçekleştirmektedir. Bu yaklaşımın öncülerinden biri, Ağırlıklı Kümeleme Algoritması (WCA)'dır [7].

WCA, KB seçiminde tek bir parametreye dayalı karar mekanizmalarının yetersizliğini aşmak amacıyla geliştirilmiş, çok kriterli bir yaklaşımdır. WCA, düğümün iletim gücü, hareketliliği, ideal komşu sayısından sapması ve kalan pil gücü gibi farklı parametreleri dikkate alarak her düğüm için bir ağırlık değeri hesaplar. Bu parametrelerin her biri,

belirlenen ağırlık katsayıları ile çarpılarak toplam ağırlık değeri oluşturulur ve en düşük ağırlığa sahip düğüm KB olarak seçilir. Diğer bir önemli özelliği ise, KB seçim sürecinin yalnızca ihtiyaç anlarında tetiklenmesidir. Bu sayede, ağın yapısında herhangi bir değişiklik olmadıkça küme yapısı korunur ve gereksiz yeniden yapılandırmaların önüne geçilir [16]. WCA, birden fazla metriği birlikte değerlendirmesi sayesinde daha dengeli ve sürdürülebilir kümeler oluşturmaktadır. Ancak, bu yöntemin en önemli zayıflığı, çok sayıda parametrenin işlenmesinden doğan hesaplama karmaşıklığıdır. Ayrıca, her parametrenin ağırlık katsayısının manuel olarak belirlenmesi, algoritmanın farklı senaryolara uygulanabilirliğini zorlaştırmakta ve kullanıcı bağımlılığı oluşturmaktadır [8], [12].

WCA algoritmasının ağırlık tabanlı çok kriterli yaklaşımı önemli kazanımlar sağlasa da KB seçiminde yalnızca küresel minimuma odaklanması, özellikle yoğun düğüm alanlarında lokal optimizasyon fırsatlarını göz ardı etmesine neden olmuştur. Bu sınırlılığı aşmak amacıyla geliştirilen Ağırlık Tabanlı Adaptif Kümeleme Algoritması (WBACA), kümeleme sürecinde yerel minimumları dikkate alarak daha verimli ve dengeli bir kümeleme yapısı oluşturmayı hedeflemektedir [17]. WBACA, WCA'dan farklı olarak iletişim hızı ve düğüm derecesi gibi ilave parametreleri de değerlendirmekte; toplam ağırlık değeri üzerinden en düşük skora sahip düğümü KB olarak atamaktadır. Bu yaklaşım, ağın dinamik yapısına daha etkin yanıt vererek gereksiz küme yeniden yapılandırmalarını önler ve KB değişim oranlarını azaltarak ağın kararlılığını artırır. Ayrıca, WBACA'nın adaptif karakteri, KB seçim sürecinin ihtiyaç anlarında optimize edilmesini sağlayarak genel ağ performansına olumlu katkı sunar. Ancak, WCA gibi WBACA da birden fazla parametreyi eş zamanlı olarak işlediği için hesaplama karmaşıklığı yüksektir. Ayrıca, düğüm hareketliliği arttıkça WBACA'nın küme stabilitesi azalabilmekte ve daha sık KB değişimi gerektirebilmektedir [8], [18].

Ağırlık tabanlı kümeleme algoritmalarında, parametre sayısının artmasıyla birlikte ağ yapısına duyarlılık gelişmiş olsa da yüksek hareketlilik senaryolarında KB değişimlerinin sıklığı hala önemli bir problem olarak varlığını sürdürmektedir. Bu sorunu azaltmak amacıyla geliştirilen algoritmalarından biri olan Optimize Edilmiş Kararlı Kümeleme Algoritması (OSCA), ağ kararlılığını artırmaya yönelik özel bir mekanizma olan yedek düğüm yapısını kullanmaktadır. OSCA, KB seçiminin yanı sıra her küme için bir yedek düğüm tanımlar ve bu düğüm, ana KB'nin devre dışı kalması durumunda doğrudan görevi devralarak ağ kesintisini önler. Ayrıca algoritma, çoklu parametre değerlendirmesi

yaparak küme kararlılığını ve performansını artırmayı hedeflemektedir [19]. Bu yapı sayesinde KB değişimlerinin sıklığı azalarak, yeniden kümeleme süreçleri minimum düzeye indirilmekte ve böylece ağ üzerindeki kontrol yükü önemli ölçüde azaltılmaktadır. Bununla birlikte, her kümede yedek KB bulundurulması kaynak kullanımını artırmakta, ayrıca öncelik faktörlerinin hesaplanması ve yedek düğüm mekanizması algoritmanın genel hesaplama karmaşıklığını yükseltmektedir [20], [21].

Ağırlık tabanlı kümeleme algoritmalarının gelişiminde önemli bir aşama olarak kabul edilen Enerji Verimli Ağırlıklı Kümeleme Algoritması (EE-WCA), KB seçim sürecinde klasik WCA parametrelerinin ötesine geçerek daha fazla kalite ölçütünü dikkate almaktadır. EE-WCA, tampon uzunluğu, düğüm hareketliliği, pil seviyesi, bağlantı derecesi ve Sinyal-Gürültü Oranı (SNR) gibi beş QoS parametresini değerlendirerek her düğüm için ağırlıklı bir skor üretir ve bu skora göre KB atamasını yapar. Bu yaklaşım sayesinde, özellikle ağ kaynaklarının dengeli kullanımı ve enerji tüketiminin optimize edilmesi hedeflenmiştir. EE-WCA, yük dengeleme özelliğiyle ağdaki aşırı yüklenmeleri önlemekte, topoloji kontrolünü sürekli izleyerek ağ ömrünü uzatmakta ve genel performansı artırmaktadır [22], [23]. Bununla birlikte, çoklu parametrelerin kullanımı algoritmanın hesaplama karmaşıklığını artırmakta, bu da özellikle kaynak kısıtlı düğümlerde performans düşüşüne yol açabilmektedir. Ayrıca, düğüm hareketliliğinin yüksek olduğu senaryolarda, küme kararlılığı azalmakta ve KB değişim oranı artabilmektedir [22], [24].

Ağırlık tabanlı kümeleme algoritmaları, tek metrikli yaklaşımların sınırlı uyarlanabilirliğini aşarak çok kriterli karar mekanizmalarıyla daha dengeli ve verimli küme yapıları sunmayı başarmıştır. WCA ile başlayan bu yaklaşım, WBACA ile yerel optimuma duyarlılık, OSCA ile kararlılık ve yedek yapılar, EE-WCA ile enerji verimliliği ve QoS hassasiyeti gibi farklı gelişim basamaklarından geçmiştir. Bununla birlikte, bu algoritmaların büyük çoğunluğu parametre ağırlıklarını sabit olarak kabul etmekte ve her ağ koşuluna özel dinamik çözüm üretememektedir. Bu noktada, özellikle farklı ağ senaryoları için optimum çözümlerin elde edilmesini sağlayan optimizasyon tabanlı kümeleme algoritmaları ön plana çıkmıştır. Bu algoritmalar, KB seçiminde çok sayıda metriği eş zamanlı optimize ederek hem enerji verimliliğini artırmakta hem de iletişim kalitesini iyileştirmektedir. Bu sınıfın dikkat çeken örneklerinden biri, Çok Amaçlı Bir Parçacık Sürü Optimizasyonu (MOPSO)'dur. MOPSO, klasik PSO algoritmasının çok amaçlı optimizasyon problemlerine uyarlanmış halidir ve dinamik yapıya sahip mobil ad-

hoc ağlarda kümeleme süreçlerinin iyileştirilmesi amacıyla kullanılmaktadır. MOPSO, KB seçimi ve küme yapısının oluşturulmasında enerji tüketimi, bağlantı kalitesi, iletişim maliyeti gibi birden fazla metriği eş zamanlı olarak optimize eder. Algoritma optimal çözüm kümesi üreterek, farklı ağ koşulları altında uygulanabilir, çoklu çözümler sunma kapasitesine sahiptir. Bu yapı sayesinde, ağ ömrünü uzatmakta ve kaynak kullanımını dengelemektedir. MOPSO'nun en önemli avantajı, farklı performans kriterleri arasında denge kurarak küresel çözüm uzayında etkin arama yapabilmesidir. Ancak, algoritmanın çok amaçlı yapısı geniş çözüm uzayında hesaplama maliyetini artırmakta ve bazı durumlarda parçacıkların belirli bölgelerde sıkışması nedeniyle küresel en iyi çözüme ulaşma başarısını düşebilmektedir [25].

Çok amaçlı optimizasyon yaklaşımının bir başka sezgisel formu olan Karınca Kolonisi Optimizasyonuna Dayalı Kümeleme Algoritması (CACONET), Karınca Kolonisi Optimizasyonu (ACO) algoritmasının temel prensiplerini kullanarak, dinamik mobil ağlarda küme başı seçimini ve yol belirleme süreçlerini optimize etmektedir. Bu algoritma, feromon izleri ve sezgisel değerler yardımıyla düğümler arasında en verimli yolları keşfeder; böylece özellikle Araç Tabanlı Ad-Hoc Ağlardaki (VANET) ortamlarında iletişim maliyetlerini azaltmakta ve kümeleme kararlılığını artırmaktadır [26]. CACONET'in en dikkat çeken yönlerinden biri, çoklu hedef fonksiyonlarını optimize etmesi sayesinde, kümelerdeki düğüm sayısını ve iletişim mesafesini dengede tutabilmesidir. Bununla birlikte, algoritmanın ACO temelli yapısı nedeniyle feromon güncellemeleri yüksek hesaplama maliyeti doğurmakta ve ağ yoğunluğu arttıkça karar süreçleri daha karmaşık hale gelmektedir. Bu durum, özellikle zaman hassasiyeti gerektiren uygulamalarda yanıt süresinin uzamasına neden olabilmektedir [27].

Optimizasyon tabanlı kümeleme algoritmalarının gelişiminde bir sonraki aşama, farklı sezgisel yöntemlerin hibrit olarak birleştirilmesiyle daha güçlü ve dengeli sonuçlar elde edilmesini amaçlamaktadır. Bu yaklaşımın başarılı örneklerinden biri olan Ateşböceği Algoritması ile Parçacık Sürü Optimizasyonunun Hibrit Yaklaşımı (HFAPSO) algoritması, özellikle Kablosuz Sensör Ağlarında (WSN) enerji tüketimini azaltmak ve ağ ömrünü uzatmak için geliştirilmiştir. HFAPSO, Ateşböceği Algoritmasının (FA) küresel arama kabiliyeti ile PSO'nun yerel optimizasyon gücünü bir araya getirerek KB seçiminde daha hızlı ve dengeli çözümler üretmektedir. Bu yapı sayesinde ağda daha fazla aktif düğüm kalmakta ve düşük enerji tüketimi ile daha uzun süreli hizmet sağlanmaktadır [28]. Ayrıca, iki algoritmanın sinerjisi sayesinde hem keşif hem de ince ayar süreçleri

optimize edilmiştir. Ancak, bu güçlü yapının doğal bir sonucu olarak, algoritmanın hesaplama maliyeti artmakta ve daha uzun iterasyon süreleri nedeniyle özellikle gecikmeye duyarlı uygulamalarda performans sınırlamaları yaşanabilmektedir [29], [30].

Karmaşık MANET ortamlarında yalnızca enerji verimliliği değil, güvenli ve istikrarlı veri iletimi de öncelikli hedefler haline gelmiştir. Bu doğrultuda geliştirilen Enerji Verimli Ve Güvenli Veri İletimi İçin Bulanık Kaos Uyarlanabilir Parçacık Sürü Optimizasyonu (F-CAPSO) algoritması, bulanık mantık sistemleri, Salp Sürüsü Algoritması (SSA) ve Sinüs Kosinüs Algoritması (SKA) gibi sezgisel yöntemleri birleştirerek hibrit bir kümeleme yaklaşımı sunmaktadır [31]. F-CAPSO, KB seçim sürecini optimize ederek hem ağ ömrünü uzatmayı hem de Çift Dalga Şifreleme stratejisi ile veri iletiminde güvenilirliği sağlamayı hedeflemektedir. Bu sayede, karmaşık MANET topolojilerinde bile hem enerji verimli hem de güvenli yönlendirme elde edilmektedir. Ayrıca, uyarlanabilir optimizasyon yapısı sayesinde düğüm hareketliliğine karşı oldukça dayanıklıdır [94]. Ancak algoritmanın hibrit ve çok katmanlı yapısı, yüksek hesaplama maliyeti ve artan gecikme süresi gibi sınırlamaları da beraberinde getirmektedir. Özellikle KB rotasyonlarının sık yaşandığı durumlarda, ek enerji tüketimi oluşmakta ve gerçek zamanlı uygulamalarda performans düşüşü gözlemlenebilmektedir [32].

MANET’lerde kümeleme yaklaşımlarından bir diğeri de makine öğrenmesi temelli yöntemlerdir. Bu yöntemler, ağdaki verilerin örüntülerini analiz ederek küme yapısını dinamik biçimde şekillendirebilmekte ve özellikle değişken topolojilere sahip ortamlarda adaptif çözümler sunabilmektedir. Makine öğrenmesi tabanlı kümeleme algoritmaları, genellikle denetimsiz öğrenme modellerine dayanmakta; enerji durumu, hareketlilik, güvenlik seviyesi gibi çok sayıda ölçütü değerlendirerek düğümler arasında benzerlik temelli gruplamalar yapmaktadır. Bu algoritmaların KB seçimi, yönlendirme ve saldırı tespiti gibi alanlardaki başarısı, onları geleneksel yöntemlere güçlü bir alternatif haline getirmiştir.

Literatürde bu yaklaşımların bir örneği olarak geliştirilen Tepe Tırmanma Yöntemli Genetik Algoritması (GAHC), MANET’ler için geliştirilmiş optimize edilmiş bir çoklu yol yönlendirme protokolü olup, Genetik Algoritma (GA) ve Tepe Tırmanışı (HC) yöntemlerini birleştirerek hibrit bir yapı sunmaktadır. Bu algoritma, KB seçiminde ve rota belirleme süreçlerinde düğüm güvenilirliği, bağlantı kalitesi ve enerji durumu gibi çoklu parametreleri değerlendirmekte ve ağıın genel performansını artırmayı hedeflemektedir. Hibrit yapısı sayesinde, yalnızca enerji verimliliği sağlamakla kalmaz; aynı zamanda

güvenlik saldırılarını tespit etme ve önleme yeteneği de kazandırmaktadır [33]. GAHC, güvenilir bağlantılar seçerek veri iletiminin sürekliliğini sağlarken, düğüm ömrünü uzatmaya da katkıda bulunur. Ancak bu çok katmanlı yapı, hesaplama yükünü artırmakta ve KB seçim sürecinde oluşan kontrol mesajları nedeniyle ağ trafiğini yükseltebilmektedir. Özellikle dinamik topolojilerde KB'nin sık güncellenmesi, toplam enerji tüketiminin artmasına neden olmaktadır [34], [35].

Makine öğrenmesi tabanlı yaklaşımların enerji verimliliği ve ağ kararlılığı üzerindeki olumlu etkileri, kümeleme algoritmalarının yalnızca KB seçiminde değil, aynı zamanda yönlendirme kararlarında da optimizasyon teknikleriyle desteklenmesini beraberinde getirmiştir. Bu doğrultuda geliştirilen, Enerji Verimli – Değiştirilmiş Afrika Akbabası Ve Değiştirilmiş Mayıs Sineği Algoritması (E-MAVMMF), MANET ortamında hem KB seçimi hem de veri iletimi süreçlerini çok ölçütlü yapay zekâ bileşenleri ile optimize etmektedir. Kümeleme aşamasında geliştirilen Afrika Akbaba Optimizasyon Algoritması (AVOA) kullanılarak enerji tüketimi dengelenmekte, yönlendirme aşamasında ise Modifiye Edilmiş Mayıs Sineği Algoritması (MMA) algoritması ile en verimli iletim yolu seçilmektedir. Bu iki katmanlı yapı sayesinde, ağın genel enerji tüketimi düşürülmekte ve ağ ömrü uzatılmaktadır [36]. Ancak, AVOA ve MMF gibi iki farklı optimizasyon algoritmasının aynı sistemde bütünleştirilmesi, hesaplama karmaşıklığını artırmakta ve özellikle yönlendirme sırasında gereken fazladan bilgi alışverişi nedeniyle bant genişliği kullanımını yükseltmektedir [37].

Literatürde yaygın olarak kullanılan algoritmalarından biri olan Filipin Kartalı Tabanlı Geliştirilmiş K-Ortalamlar Algoritması (MKMPE), KB seçiminde modifiye edilmiş k-means (MKM) algoritmasını, rota seçiminde ise Filipin Kartalı Optimizasyon (PEO) yöntemini bir araya getirerek hibrit bir kümeleme ve yönlendirme yaklaşımı sunmaktadır. MKM, düğüm güvenilirliğini esas alan Doğrudan Güven (DT), Dolaylı Güven (IDT) ve Güncel Güven (RT) gibi ölçütleri değerlendirerek en uygun küme başlarını seçmektedir. Ardından PEO algoritması, ağda düşük gecikmeli ve enerji tasarruflu rotaları belirleyerek veri iletiminin verimli biçimde gerçekleşmesini sağlar. Bu yapı sayesinde, KB seçimi ve rota optimizasyonu birlikte ele alınarak düğümler arası enerji tüketimi dengelenmekte ve ağ ömrü uzatılmaktadır [38]. Ancak, bu algoritmanın çok katmanlı yapısı nedeniyle hesaplama yükü artmakta, ayrıca KB değişimlerinin sabit periyotlarla gerçekleşmesi, düşük enerjili düğümlerin aşırı yüklenmesine neden olabilmektedir [37].

1.3. TEZİN AMACI

MANET'ler sabit bir altyapıya ihtiyaç duymadan hareketli düğümler arasında geçici iletişim kuran ağlardır. Ancak bu ağların dinamik yapısı, enerji tüketimi, yönlendirme maliyetleri, KB değişim sıklığı ve yeniden kümeleme işlemleri gibi birçok zorluğu beraberinde getirmektedir. Özellikle düğümlerin sınırlı enerji kaynaklarına sahip olması, ağır kararlılığını ve verimliliğini doğrudan etkileyen temel bir sorundur.

Bu tezde, MANET'lerin dinamik yapısındaki bu zorlukların üstesinden gelmek amacıyla, ağırlığa dayalı optimizasyon yöntemleri ve makine öğrenmesi tabanlı kümeleme algoritmaları kullanılarak daha verimli bir kümeleme mekanizması geliştirilmesi hedeflenmiştir.

Bu kapsamda, tezin temel amacı, dinamik MANET'lerde:

- Enerji tüketimini azaltmak,
- Ağ yapısını daha kararlı hale getirmek,
- Yeniden kümeleme işlemlerinin sıklığını azaltmak,
- KB değişim süresini minimize etmek,
- Yönlendirme maliyetlerini düşürmek ve
- Ağın topolojik değişikliklere uyum sağlama yeteneğini artırmaktır.

Bu hedeflere ulaşmak için, tez çalışmasında KB seçiminde ağırlık faktörlerinin optimizasyonu ve kümeleme süreçlerinde adaptif yaklaşımlar benimsenmiştir. Özellikle KB değişim süresinin azaltılması, ağdaki kararlılığın artırılması ve iletişim kesintilerinin minimize edilmesi açısından önemli bir katkı sağlamaktadır. Bu sayede, MANET'lerde daha az kümeleme işlemi gerçekleştirilerek ağın ömrü uzatılmakta ve genel verimlilik artırılmaktadır.

1.4. TEZİN KAPSAMI

Bu tez çalışması, MANET'lerde enerji tüketimini azaltmak, ağır kararlılığını artırmak ve topolojik değişikliklere hızlı uyum sağlayan daha verimli bir yönlendirme mekanizması geliştirmek amacıyla kümeleme ve KB seçimi üzerine odaklanmaktadır. Özellikle, KB seçim sürecinin optimize edilmesi ve ağ yapısının daha uzun süre kararlı kalmasını

sağlamak için Yerçekimi Arama Algoritması (GSA) kullanılmıştır. Ayrıca literatürde yer alan Gürültülü Veriler İçin Yoğunluk Tabanlı Uzamsal Kümeleme (DBSCAN), temel parametreleri (Eps ve MinPts) üzerinde yapılan iyileştirmeler ve yapısal düzenlemeler ile geliştirilmiş; bu sayede kümeleme işlemleri ve düğümlere rol atamaları daha verimli şekilde gerçekleştirilmiştir.

Tez kapsamında, MANET'lerde yönlendirme maliyetlerinin azaltılması, yeniden kümeleme sıklığının düşürülmesi ve ağ kaynaklarının verimli kullanımı hedeflenmektedir. Bununla birlikte, KB'lerinin seçiminde enerji verimliliğinin artırılması ve düğümler arasındaki iletişimin daha istikrarlı hale getirilmesi için mevcut yöntemlerdeki eksikliklerin giderilmesi amaçlanmaktadır.

Bu çalışmada, yönlendirme algoritması olarak Tasarsız İsteğe Bağlı Uzaklık Vektörü (AODV) protokolü tercih edilmiştir. Ancak, AODV'ye bazı ek veri paketleri eklenerek kümeler arası iletişim süreçlerinin daha etkili hale getirilmesi sağlanmıştır. KB'ler arasında Geçityolu Düğümleri (GD) kullanılarak, farklı kümelerdeki düğümlerin birbirleriyle güvenilir bir şekilde iletişim kurması hedeflenmiştir.

Geliştirilen model, simülasyon tabanlı bir değerlendirme ile test edilmiş ve farklı topolojilerde ağ performansı analiz edilmiştir. Yapılan simülasyonlar, ağ kararlılığının artırılması, enerji tüketiminin azaltılması ve yeniden kümeleme sayısının düşürülmesi açısından önerilen modelin mevcut protokollere kıyasla daha etkili olduğunu göstermektedir.

1.5. BİLİME KATKISI

Bu tez çalışması, dinamik MANET ler için KB seçimi ve kümeleme süreçlerini iyileştirmek amacıyla WCA-GSA ve Geliştirilmiş-DBSCAN algoritmalarının kullanımına dayalı özgün bir yaklaşım sunmaktadır. Literatürdeki mevcut kümeleme yöntemlerine kıyasla, önerilen yaklaşımın MANET'lerin dinamik yapısına daha iyi uyum sağladığı ve ağ verimliliğini artırdığı gözlemlenmiştir.

Bu doğrultuda tez çalışmasının bilimsel katkıları şu şekilde özetlenebilir:

- Geliştirilen yöntem, KB'lerin görevde kalma süresini uzatarak enerji tüketimini azaltmakta ve dolayısıyla düğüm ömrünü artırmaktadır.
- Yeniden kümeleme sıklığının düşürülmesi ve KB değişim süresinin kısaltılması

sayesinde, ağ topolojisindeki değişikliklere daha hızlı uyum sağlanmakta ve ağın genel kararlılığı yükselmektedir.

- Makine öğrenmesi tabanlı kümeleme yaklaşımı olan Geliştirilmiş-DBSCAN algoritmasının düğüm rol atama ve kümeleme süreçlerinde kullanılmasıyla, ağ yapısının verimliliği artırılmıştır. Bu yenilikçi yaklaşım, literatürdeki benzer çalışmalara kıyasla farklı bir bakış açısı sunmakta ve alandaki araştırmalara katkı sağlamaktadır.
- KB değişim sürelerinin kısaltılması, yeniden kümeleme işlemlerinin seyrekleştirilmesi ve yönlendirme süreçlerinin geliştirilmesi sayesinde, ağdaki iletişim maliyetleri azalmış ve paket teslim oranları iyileştirilmiştir.

1.6. TEZ ORGANİZASYONU

Bu tez çalışmasının bütünsel yapısı aşağıda özetlenen bölümler doğrultusunda yapılandırılmıştır.

Bölüm 2’de, MANET yapısında yönlendirme protokolleri ve kümeleme yaklaşımları ele alınmaktadır. İlk olarak MANET’lerde yönlendirme süreci, protokol sınıflandırmaları ve AODV gibi reaktif protokoller açıklanmaktadır. Ardından, MANET’lerde kümeleme kavramı incelenmekte ve KB seçimi, kümeleme stratejileri ile ilgili yöntemler literatür bağlamında değerlendirilmektedir. Bu bölümde ayrıca, ağırlık tabanlı yaklaşımlar ve makine öğrenmesi destekli yöntemlere yer verilerek literatürdeki eksikliklere dikkat çekilmektedir.

Bölüm 3’te, tez kapsamında geliştirilen Dinamik MANET’ler için Hibrit Adaptif Kümeleme Algoritması (DMHAKA) ayrıntılı olarak tanıtılmaktadır. Bu kapsamda, KB seçiminde WCA ve GSA algoritmalarının entegrasyonu açıklanmış; düğüm rollerinin belirlenmesinde kullanılan Geliştirilmiş-DBSCAN algoritmasının uyarlanmış yapısı detaylandırılmaktadır. Ayrıca kullanılan yönlendirme protokolü olan AODV yapısında yapılan iyileştirmeler ve simülasyon ortamına ilişkin parametreler paylaşılmaktadır.

Bölüm 4’te, önerilen yönteme ait simülasyon bulguları ve analizleri sunulmuştur. Önce klasik WCA algoritması ile karşılaştırmalı performans analizleri yapılmış, ardından EE-WCA, E-MAVMMF, TSDR ve MORS-ASO gibi güncel algoritmalarla kapsamlı değerlendirmeler gerçekleştirilmektedir. Elde edilen veriler doğrultusunda

DMHAKA'nın enerji verimliliđi, gecikme süresi, paket teslim oranı ve veri aktarım kapasitesi gibi performans metriklerinde sağladığı kazanımlar detaylı grafik ve tablolarla açıklanmaktadır.

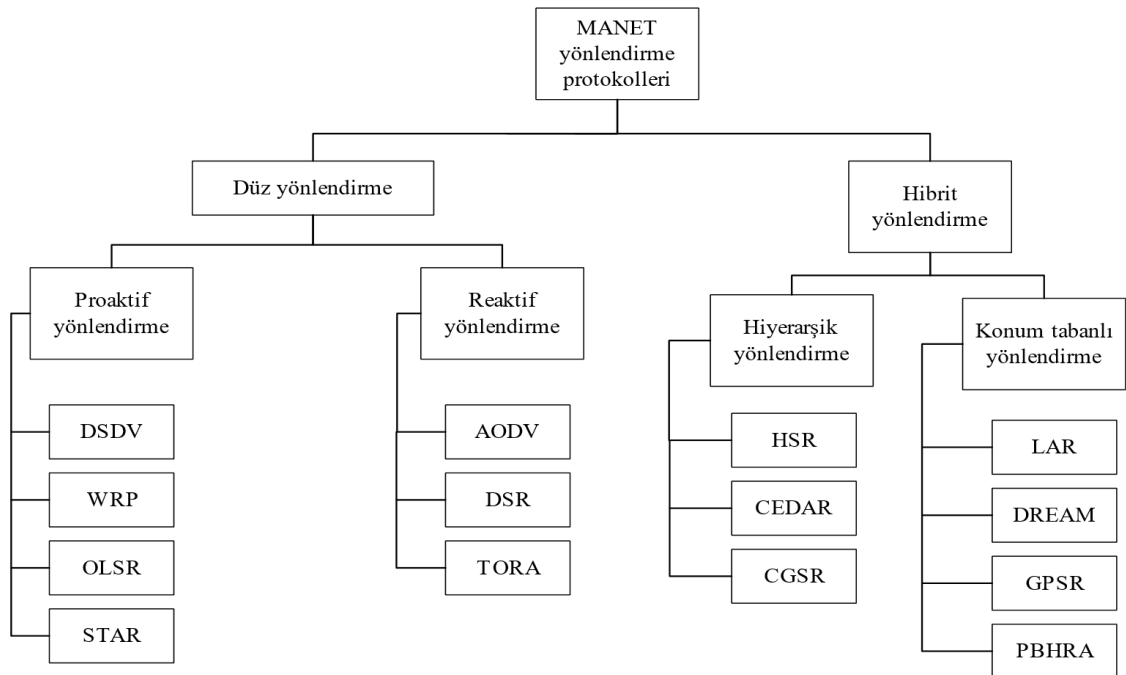
Bölüm 5'te, tez çalışmasından elde edilen sonuçlar özetlenerek, çalışmanın literatüre ve uygulamaya sağladığı katkılar maddeler halinde sunulmaktadır. Ayrıca gelecek çalışmalara ışık tutabilecek çeşitli öneriler sıralanmaktadır.



2. MANET'LERDE YÖNLENDİRME VE KÜMELEME YAKLAŞIMLARI

2.1. MANET'LERDE YÖNLENDİRME PROTOKOLLERİ

MANET'lerdeki düğümler sıklıkla değişen çok sekmeli bir ağ topolojisine sahip oldukları için, verimli yönlendirme protokollerine ihtiyaç duyulmaktadır. Bu doğrultuda, bazı protokoller Internet Engineering Task Force (IETF) tarafından standartlaştırma sürecine alınmış ve literatürde çok sayıda çözüm önerilmiştir. Söz konusu protokoller hem periyodik olarak hem de topolojideki değişiklik anlarında yönlendirme bilgilerini güncelleyerek ağın dinamik yapısına uyum sağlamaktadır [39]. MANET'ler için çok sayıda yönlendirme protokolü geliştirilmiştir. Şekil 2.1'de gösterildiği üzere, yönlendirme stratejilerine bağlı olarak literatürde yaygın biçimde kullanılan bu protokoller; düz yönlendirme başlığı altında proaktif (tabloya dayalı), reaktif (isteğe bağlı ya da kaynak başlatmalı) ve hibrit (her iki yaklaşımın birleşimi) olmak üzere üç temel kategoriye ayrılmaktadır [40], [41]. Bu çalışmada ise, bazı protokoller hibrit yönlendirme başlığı altında hiyerarşik yönlendirme ve konum tabanlı yönlendirme olmak üzere iki alt sınıfa ayrılarak ele alınmaktadır.



Şekil 2.1. Yönlendirme stratejisine göre bazı MANET yönlendirme protokolleri.

2.1.1. Düz Yönlendirme Protokolleri

MANET’lerde ağın tamamını tek bir düzeyde ele alarak düğümler arasında iletişim yollarını belirlemek amacıyla kullanılan bir yaklaşımdır. Bu tür protokoller, ağ yapısında herhangi bir hiyerarşi oluşturmaz ve tüm düğümler eşit öneme sahiptir. Düz yönlendirme protokolleri, yönlendirme bilgilerini ağdaki topolojik değişikliklere uyum sağlamak için düzenli aralıklarla ya da ihtiyaç duyulduğunda güncellemektedirler [41], [42].

Bu gruptaki protokoller üç ana kategoriye ayrılır:

2.1.1.1. Proaktif (Tabloya Dayalı) Protokoller

Proaktif yönlendirme protokolleri, ağdaki tüm düğüm çiftleri arasında tutarlı ve güncel yönlendirme bilgilerini korumak amacıyla, yönlendirme tablolarını düzenli aralıklarla günceller. Bu protokoller, ağdaki topolojik değişikliklere uyum sağlamak için sık sık bilgi yayarak, düğümlerin her zaman mevcut bir yönlendirme rotasına sahip olmasını amaçlar. Proaktif protokoller, genellikle kablolu ağlarda kullanılan geleneksel bağlantı durumu ve mesafe vektörü protokollerinin, dinamik ve hareketli MANET ortamına uyarlanmış versiyonlarıdır. Ancak, bu protokollerde tüm düğümler için yönlendirme bilgilerinin sürekli olarak güncellenmesi her zaman gerekli olmamaktadır [43].

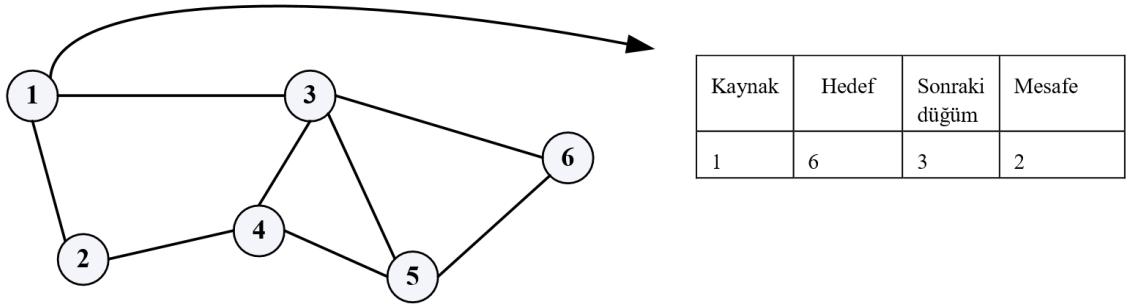
Literatürde geçen proaktif protokollerden bazıları şunlardır:

Hedef Sıralı Uzaklık Vektörü (DSDV) yönlendirme, Kablosuz Yönlendirme Protokolü (WRP) ve Optimize Edilmiş Bağlantı Durumu Yönlendirme (OLSR), Kaynak Ağacı Uyarlamalı Yönlendirme (STAR) [39], [43].

2.1.1.1.1. DSDV

Perkins ve Bhagwat tarafından 1994 yılında geliştirilen bu protokol, MANET’lerde kullanılan ilk proaktif yönlendirme protokollerinden biridir ve Bellman-Ford yönlendirme algoritmasının optimize edilmiş bir versiyonuna dayanmaktadır. Mesafe vektörü yönlendirme mekanizması kullanan DSDV protokolü, diğer yöntemlerle kıyaslandığında sıçrama etkisi nedeniyle daha az güvenilir olarak değerlendirilmektedir. Bu protokolda, her düğüm ağdaki tüm cihazları içeren bir yönlendirme tablosu tutar ve bu tabloyu belirli aralıklarla günceller. Yönlendirme bilgilerinin doğruluğunu sağlamak amacıyla düğüm, yönlendirme taleplerini tüm diğer düğümlere yayınlar. Komşu düğümler, mevcut yönlendirme maliyetini değerlendirir ve bu maliyeti yönlendirme tablosundaki mevcut değerle karşılaştırır. Eğer bir güncelleme gerekiyorsa, düğüm

maliyeti yeniden hesaplayarak yönlendirme tablosunu günceller [2], [44]. Şekil 2.2’de DSDV protokolünde paket yönlendirilmesi gösterilmektedir.

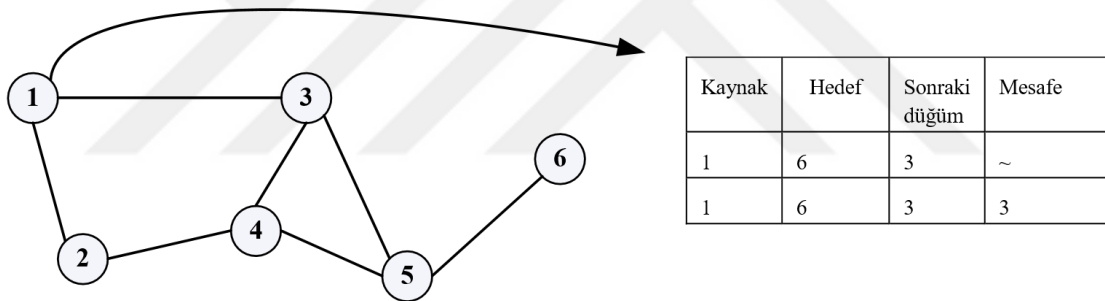


Şekil 2.2. DSDV protokolü ve yönlendirme tablosu [45].

Bellman-Ford algoritmasına göre amaç en kısa yolu bulmak olduğu için;

- Kaynak düğüm 1 ve hedef düğüm 6 sayıldığında paketler düğüm 1’den düğüm 3’e gider ve oradan hedef olan düğüm 6’ya varırlar.

Şekil 2.3’te ise rota bozulması durumunda yönlendirmenin nasıl olacağı gösterilmektedir.



Şekil 2.3. Düğüm 3 ile düğüm 6 arasındaki yolun kopması durumu ve yönlendirme tablosu [45].

Şekil 2.3’teki gibi eğer bir komşu düğüm bir bağlantı kopması algıladığında (bu durumda düğüm 3):

- Bozuk bağlantı üzerinden hedefe giden tüm rotaları sonsuz olarak ayarlar.
- Yönlendirme tablosunu yayınlar.
- Düğüm 5 güncelleme mesajını alır, komşuları düğüm 6’ya en kısa mesafe hakkında bilgilendirir.
- Bu bilgiler ağ üzerinden yayılır ve tüm düğümler yönlendirme tablolarını günceller.

- Düğüm 1 artık paketlerini 1-3-6 yerine 1-3-5-6 yoluyla göndermektedir.

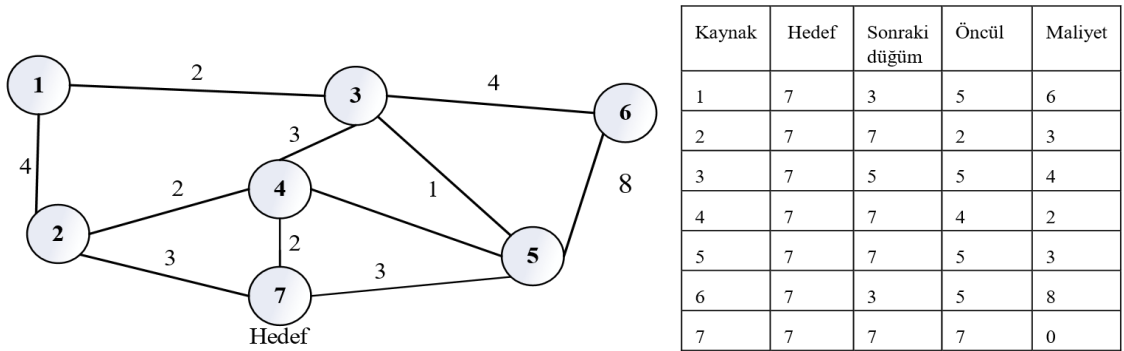
2.1.1.1.2. WRP

Murthy ve Acaves tarafından geliştirilen bu algoritma, DSDV'ye benzer şekilde yönlendirme bilgilerini saklamak için bir tablo yapısından faydalanan bir protokoldür. Ağdaki her düğüm, hedeflere yönelik yönlendirme bilgilerini içeren bir tablo tutar. Bu tablolar, her hedefe ulaşmak için kullanılacak en kısa yolları içerir. Döngü içermeyen bir yapıya sahip olan bu protokol, proaktif bir doğaya sahiptir ve yalnızca iletilen komşu için mesafeyi güncellemekle kalmaz, diğer komşuların mesafelerini de değerlendirir. Ayrıca, sonsuza kadar sayma problemine çözüm sunan bir yol bulma mekanizması barındırır.

Bu algoritmaya göre, ağdaki her düğüm aşağıda belirtilen dört farklı tabloyu tutar:

- Mesafe tablosu (DT): Hedefe olan mesafeyi ve son olarak kullanılan düğümü içerir.
- Mesaj yeniden iletim listesi (MRL): Yeniden iletilecek mesajların kaydını tutar ve her bir mesajın yeniden iletilme işleminden sonra sayaç değerini azaltır.
- Bağlantı maliyet tablosu (LCT): Her bağlantının iletişim maliyetini ve son başarılı güncellemeden bu yana geçen süreyi kaydeder. Bu tablo, bağlantı kesintilerini tespit etmek için kullanılmaktadır.
- Yönlendirme tablosu (RT): En kısa mesafeyi, hedefe giden önceki ve sonraki düğümleri ve rotanın durumunu içermektedir [2], [46].

Düğümler arasında bağlantı hatası olması durumunda komşular bilgilendirilir. Şekil 2.4'te WRP paket yönlendirmesi hedef 7 için her düğümdeki yönlendirme girişleri gösterilmektedir.

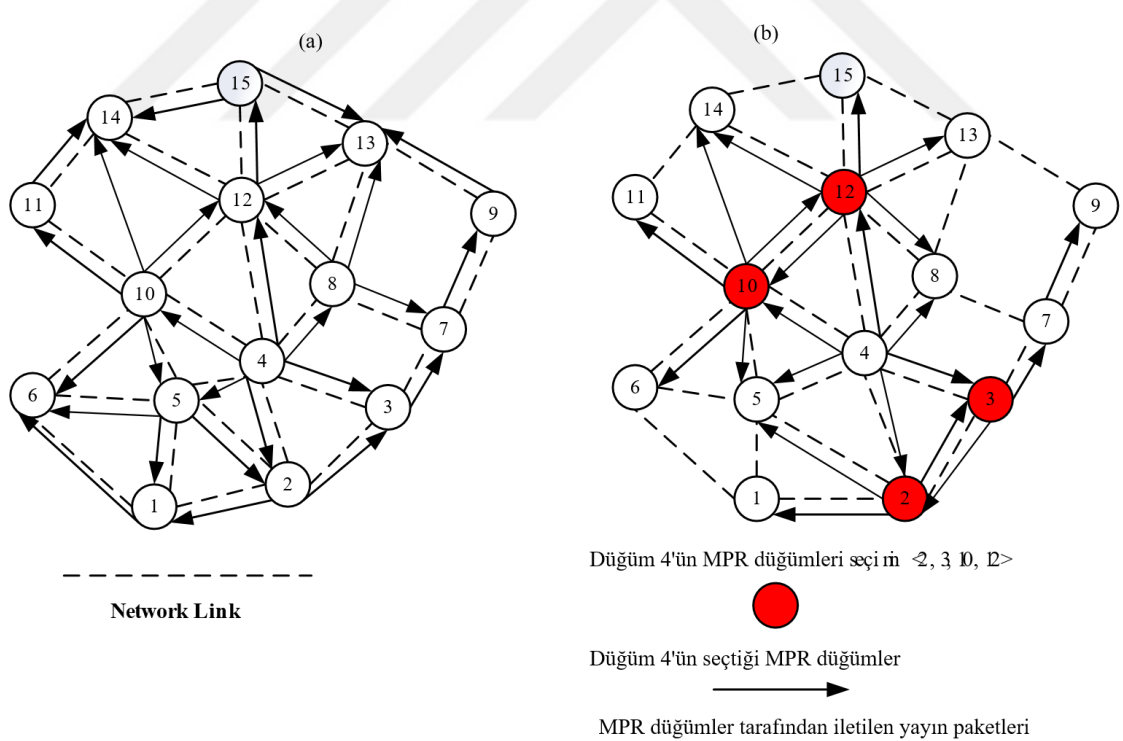


Şekil 2.4. WRP deki Yönlendirme tablosu [45].

Örneğin Şekil 2.4'e göre, düğüm 1'in hedef düğüm 7'ye paket göndermek için izlediği rota belirlenmiştir. Düğüm 1, yönlendirme tablosunda hedef düğüm 7'ye ulaşmak için bir sonraki düğümü düğüm 3 olarak seçmektedir. Düğüm 3'ün yönlendirme tablosuna göre ise bir sonraki düğüm, düğüm 5 olarak tanımlanmıştır. Bu durumda, düğüm 1'in öncülü düğüm 5'tir. Paketler, düğüm 1'den düğüm 3'e, ardından düğüm 5 üzerinden hedef düğüm 7'ye iletilmektedir. Bu rotadaki toplam maliyet ise 6 birimdir.

2.1.1.1.3. OLSR

Jacquet ve arkadaşları tarafından önerilen OLSR protokolü, noktadan noktaya bir bağlantı durumu protokolüdür ve yönlendirme işlemleri için güçlü bir mekanizma olan Çok Noktalı Aktarma (MPR) yöntemini kullanır. MPR düğümleri, çift yönlü bağlantıya sahip tek atlamalı komşu düğümler arasından seçilir ve flooding sırasında yayın mesajlarını ileten özel düğümler olarak görev yapar. Geleneksel flooding mekanizmasında her düğüm alınan her mesajı yeniden iletirken, MPR yöntemi yalnızca seçilmiş düğümlerin mesajları iletilmesiyle kontrol mesajı yükünü büyük ölçüde azaltmaktadır [47]. Şekil 2.5'te OLSR protokolü için MPR seçimi gösterilmektedir.

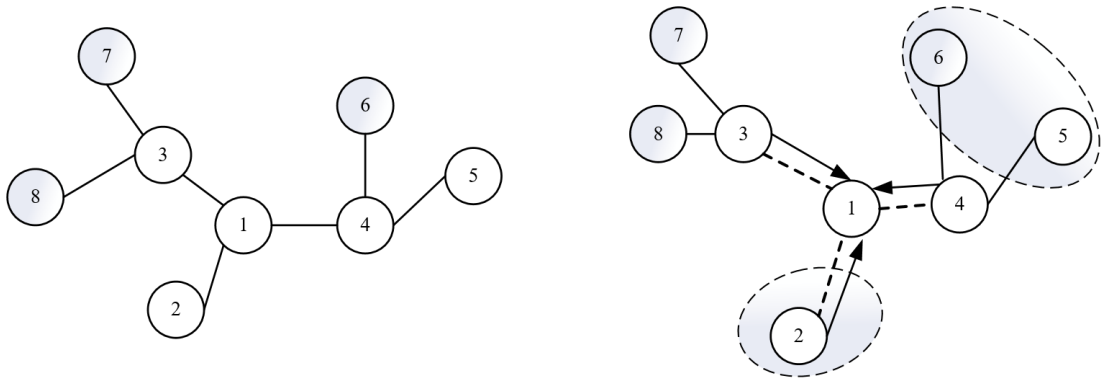


Şekil 2.5. OLSR'de MPR seçiminin bir örneği [48]. (a) ağın klasik flooding mekanizması ile düğüm sayısı kadar iletim alması. (b) MPR şemasını kullanarak tüm ağın altı iletimle doldurulması.

MPR tabanlı bu yaklaşım, tek yönlü bağlantılarda veri aktarımıyla ilişkili sorunları otomatik olarak ortadan kaldırır ve aynı zamanda düğümlerin pil tüketimini düşürmektedir. Bunun nedeni, OLSR protokolünde bağlantı durumu bilgilerinin yalnızca MPR olarak seçilen düğümler tarafından oluşturulmasıdır. Ayrıca, bir MPR düğümü, yalnızca kendisi ile MPR seçicileri arasındaki bağlantıları raporlamayı tercih eder, bu da ağda kısmi bağlantı durumu bilgisinin dağıtılmasını sağlamaktadır. Her düğüm, bu bilgiler ışığında bir optimal rota hesaplar ve en kısa yol algoritması yardımıyla en iyi rotayı belirler. Böylelikle bir veri iletimi gerektiğinde, tüm düğümlere giden yollar önceden hesaplanmış olur ve veri, mümkün olan en kısa yoldan hemen iletilmektedir [49].

2.1.1.1.4. STAR

Sophon ve arkadaşlarının geliştirdiği STAR protokolü, her düğümün başlatma aşamasında komşularına bir güncelleme mesajı göndermesini gerektirir. Bu mesaj, yeni hedefler, olası yönlendirme döngüleri ve yolların maliyetleri hakkında bilgi içermektedir. STAR protokolünde her düğüm, kaynak ağacı bilgisini tercih edilen hedef yolunda kullandığı kablosuz bağlantılar üzerinden yayımlar. Yönlendirici, ad-hoc ağdaki bilinen tüm hedeflere ve adres aralıklarına ulaşmak için gereken bağlantıları içeren kaynak yönlendirme ağacının parametrelerini komşularına iletir. Protokol, ağdaki her düğümün tüm hedeflere ulaşmak için bir yol bulundurmasını sağlar. Eğer bir düğüm, belirli bir hedefe giden yolu bilmiyorsa ve o hedefe paket göndermesi gerekiyorsa, komşularına “yol yok” mesajı göndererek durumu bildirmektedir [50], [51].



Şekil 2.6 STAR protokolü genel yapısı [39].

Şekil 2.6'da görüldüğü üzere, düğüm 1'in, hedef düğüm 6'ya veri göndermek istediği bir senaryo ele alınmıştır. Ancak, düğüm 1'in kaynak yönlendirme ağacında düğüm 6'ya giden bir yol bulunmamaktadır. STAR protokolü bu durumda devreye girer ve düğüm 1, aşağıdaki adımları izleyerek yönlendirme sürecini gerçekleştirir:

- D ğ m 1, t m komşu d ğ mlerine bir g ncelleme mesajı g ndererek Hedef d ğ m 6'ya giden bir yolu olmadığını bildirir. Bu mesaj, komşu d ğ mlerden y nlendirme bilgisi talep eder.
- G ncelleme mesajını alan komşular, y nlendirme bilgilerine dayanarak yanıt verir.
- D ğ m 2, hedef d ğ m 6'ya bir yolunun olmadığını ‘‘yol yok’’ mesajı ile bildirir.
- D ğ m 3 ve d ğ m 4, eğer varsa hedef d ğ m 6'ya giden yollarını ve bu yolların maliyetlerini d ğ m 1'e iletir.
- D ğ m 4, d ğ m 6'ya dođrudan bir bađlantıya sahip olduğunu ve bu bađlantının maliyetinin 1 birim olduğunu bildirir.
- D ğ m 1, komşularından aldığı bilgileri kullanarak kaynak y nlendirme ađacını g nceller. Bu g ncelleme sonrasında, d ğ m 1 i in hedef d ğ m 6'ya giden en uygun yol belirlenir: D ğ m 1 → d ğ m 4 → d ğ m 6.
- G ncellemeler tamamlandıktan sonra, d ğ m 1, kaynak y nlendirme ađacında belirlediđi bu en kısa ve maliyeti d ş k yol  zerinden veri iletimine bařlar.

STAR protokol nde, ađ topolojisi y nlendirilmiř bir grafik olarak modellenir: $G = (V, E)$. Burada, V d ğ mler k mesini, E ise d ğ mleri birbirine bađlayan kenarları ifade eder. Eđer bir komşu, hedefe giden bir yol biliyorsa, yanıt olarak kendi kaynak ađacını g nderir. Aksi takdirde, komşu d ğ m bu mesajı diđer komşularına ileterek bir yanıt alınana kadar iřlemi s rd r r. Bu s re , STAR protokol nde bađlantı kesintilerini y netmek i in kullanılan bir bakım mekanizması olarak kabul edilmektedir [51].

2.1.1.2. Reaktif (İsteđe Bađlı) Protokoller

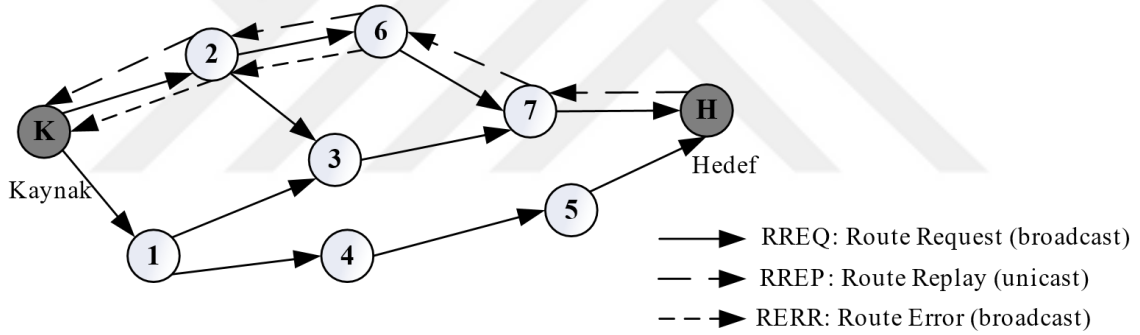
Reaktif protokoller, MANET'lerde y nlendirme i in alternatif bir yaklařım sunar. Bu protokoller, proaktif protokollerden farklı olarak, yalnızca bir talep  zerine ve kaynak d ğ m tarafından bařlatılan bir s re le yolları oluřturur. Genellikle, ađ i inde bir yol keřif mekanizması aracılıđıyla hedefe ulařan yollar dinamik olarak kurulur. Proaktif y nlendirmeden farklı olarak, reaktif protokoller s rekli olarak ađ bađlantılarını belirlemeye  alışmaz. Bunun yerine, yalnızca veri iletimi gerektiđinde talep  zerine bir yol oluřturma prosed r  bařlatılmaktadır.

Reaktif y nlendirme protokollerine  rnek olarak řu protokoller verilebilir:

- Tasarsız İsteğe Bağlı Uzaklık Vektörü (AODV)
- Dinamik Kaynak Yönlendirme (DSR)
- Geçici Sıralı Yönlendirme Algoritması (TORA) [2].

2.1.1.2.1. AODV

Lee ve arkadaşları tarafından geliştirilen AODV protokolü, MANET’lerde yaygın olarak kullanılan bir tek yöllü, isteğe bağlı yönlendirme protokolüdür. Bu protokol, yalnızca mobil ad-hoc ağlarda değil, diğer mobil ağ türlerinde de uygulanmaktadır. Cincinnati Üniversitesi, Santa Barbara ve Kaliforniya Üniversitesi gibi çeşitli akademik kurumların iş birliğiyle geliştirilmiştir. AODV’de, bir düğüm belirli bir hedefe paket göndermek için yalnızca ihtiyaç duyulduğunda bir rota oluşturur. Protokol, Rota Talebi (RREQ), Rota Yanıtı (RREP) ve Rota Hatası (RERR) mesajlarını kullanarak rota bulma ve rota bakımı olmak üzere iki ana süreçten oluşmaktadır [52]. Şekil 2.7’deki görsel, bu mesajların nasıl kullanıldığını göstermektedir.



Şekil 2.7. AODV protokolünün genel yapısı [40].

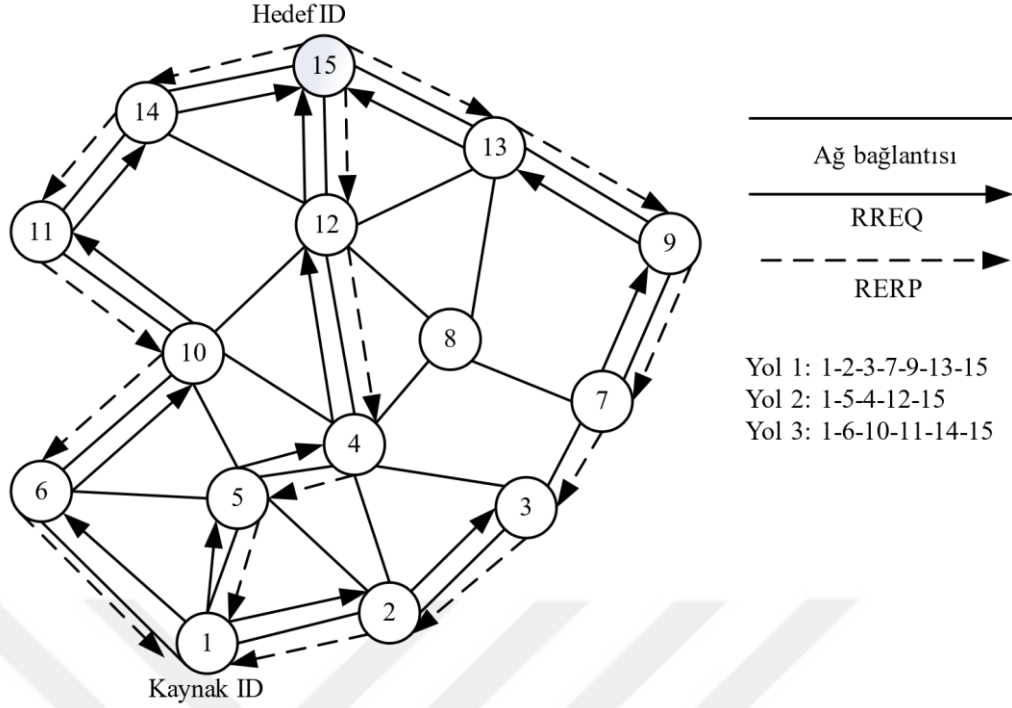
Yol keşfi sürecinde, kaynak düğüm öncelikle hedef düğümüne ulaşmak için bir RREQ paketi yayınlar. Hedef düğümünden gelen ilk yanıt, haberleşme için kullanılacak yolu belirler. Ara düğümlerden biri ilk kez bir RREQ paketi aldığı anda, kaynak düğümüne doğru bir ters rota kaydeder ve paketi yeniden iletir. Eğer aynı RREQ paketini tekrar alırsa, bu paketleri yok sayar. İlk gelen RREQ paketine yanıt olarak, hedef düğüm tek noktaya yayın yöntemiyle kaynak düğümüne bir RREP paketi gönderir. Bu paketi alan ara düğümler, hedefe giden yolu kaydeder ve RREP paketini ters rota boyunca bir önceki düğümüne iletmektedir. Sonunda RREP paketi, kaynak düğümüne ulaşarak veri iletimi için bir yol oluşturulur. AODV protokolünde, paketlerin izleyeceği yol kaynak düğüm tarafından önceden belirlenmez. Bunun yerine, kaynak ve ara düğümler yalnızca bir sonraki düğümün adres bilgilerini tutmaktadır [53].

Rota bakım aşamasında, bir düğüm bağlantı hatası tespit ettiğinde bir RERR paketi oluşturur ve bunu yayın yoluyla diğer düğümlere iletir. Bu paket, ilgili rotaları geçersiz hale getirirken ağdaki rotalar boyunca yayılır. Eğer RERR paketi kaynak düğüme ulaşırsa, kaynak düğüm yeni bir yol keşif sürecini başlatmaktadır [40].

2.1.1.2.2. DSR

Johnson ve arkadaşları tarafından geliştirilen DSR protokolü, kaynak düğümün veri paketinin izleyeceği tüm rotayı önceden belirlemesini sağlamaktadır. Bu rota bilgisi, paketin başlık kısmına eklenerek iletim sırasında kullanılacak yollar belirlenir. Her düğüm, ağdaki diğer düğümlere ait yolları saklayan bir yol önbelleği tutar. Kaynak düğüm, hedef düğüme bir paket göndermeden önce öncelikle yol önbelleğini kontrol eder. Bu mekanizma, periyodik güncelleme gereksinimini ortadan kaldırmaktadır. Önbellek, yakın zamanda kullanılan yolları bulmak ve yeniden kullanmak için bir referans görevi görür. Eğer önbellekte uygun bir yol mevcutsa, veri iletimi bu rota üzerinden gerçekleştirilir. Ancak, uygun bir yol bulunamazsa, kaynak düğüm bir RREQ mesajı göndererek hedefe bir yol arar. Bu süreç, hedef düğüme ulaşmak için uygun bir yol bulunana kadar devam eder. Uygun bir yol keşfedildiğinde, veri iletimi başlatılır [3], [54].

DSR protokolü, iki temel bileşenden oluşur: rota keşfi ve rota bakımı. Rota keşfi, bir kaynak düğümün hedef düğüme giden uygun bir yol bulmasını sağlayan süreci ifade eder. Rota bakımı ise mevcut yolların geçerliliğini kontrol ederek bağlantı kesintileri durumunda alternatif yolların bulunmasını sağlar. Bu mekanizmalar, DSR'nin dinamik ve esnek bir yönlendirme protokolü olarak işlev görmesini sağlamaktadır [55].



Şekil 2.8. DSR protokolünün genel yapısı [45].

Şekil 2.8’de gösterildiği gibi, bu protokolün temel adımları şu şekilde sıralanabilir:

- Rota keşfi sırasında DSR protokolü, ağ genelinde bir RREQ paketi yayımlar.
- Ara düğümler, RREQ paketini kontrol eder ve yalnızca gerekli olduğu durumlarda paketi bir sonraki düğüme iletir.
- Hedef düğüm, RREQ paketine bir RREP mesajı ile yanıt verir.
- RREP paketi, RREQ paketinin geçtiği veya kesiştiği yolu içerir.
- Hedef düğüm, yalnızca alınan RREQ paketi bir ilk mesaj ise yanıt oluşturur.

Şekil 2.8’deki örneğe göre en iyi veri akışı yol 2 üzerinden sağlanmaktadır.

2.1.1.2.3. TORA

Park ve Corson’un geliştirdiği bu algoritma, oldukça uyarlanabilir bir yönlendirme protokolü olarak bilinmekte ve aynı zamanda ‘Bağlantı Tersine Çevirme Protokolü’ olarak da adlandırılmaktadır. TORA, bir hedefe yönelik rotaların oluşturulması için Doğrudan Döngüsel Olmayan Grafik (DAG) yapısını ve rastgele bir yükseklik metriğini kullanır. Bu yapı, belirli bir hedefe genellikle birden fazla yol sunar ancak bu yolların en kısa yol olması gerekmez. TORA, en kısa yolun hesaplanması yerine, paketlerin iletileceği bir sonraki hedefin yönünü belirlemeye odaklanır. Bu sayede, kaynak düğüm

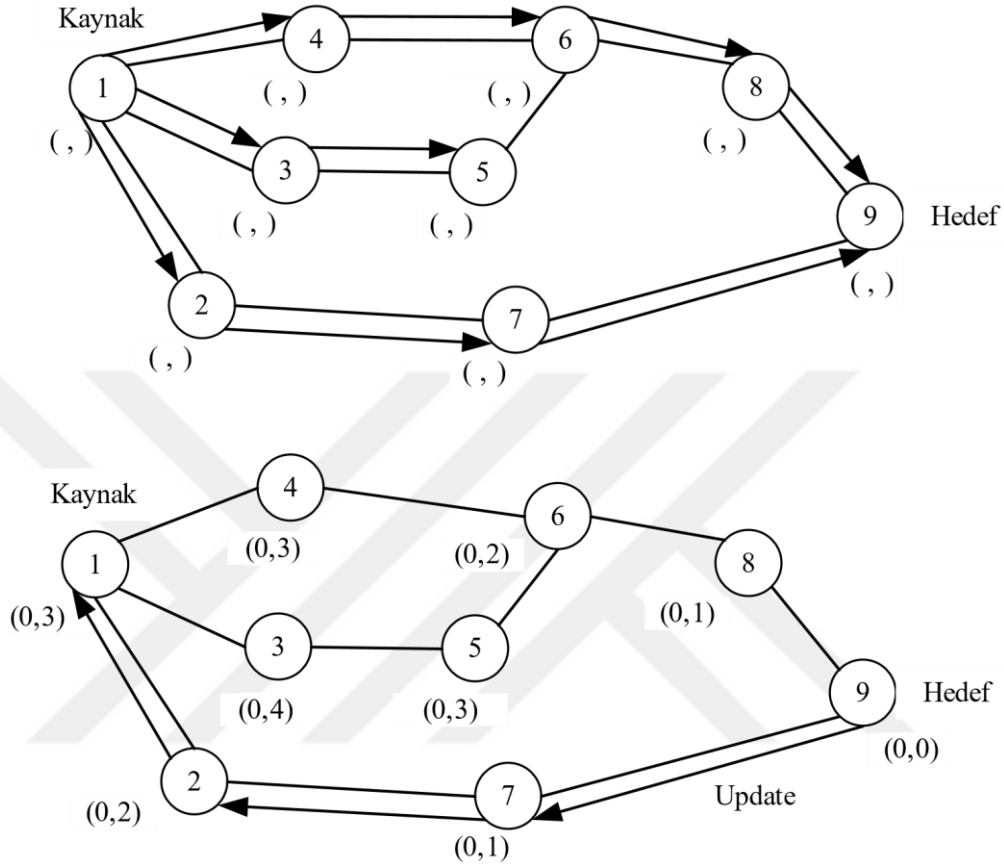
bir veya birden fazla hedefe veri akışı sağlayabilir. Ayrıca, ağdaki kontrol mesajı trafiğini azaltmak için yalnızca bir hedefe paket gönderilmesi gerektiğinde bir yol sorgulaması başlatmaktadır.

TORA protokolü, bir ağ kurulumunda üç temel adımdan oluşmaktadır.

- Kaynaktan hedefe rotaların oluşturulması:
 - Kaynak düğüm, hedef düğüme bir yol oluşturmak için bir QUERY paketi yayınlar.
 - QUERY paketi, hedef düğüme veya hedefe giden bir rotaya sahip bir ara düğüme ulaşana kadar ağda yayılır.
 - QUERY paketini alan düğüm, yüksekliğini listeleyen bir UPDATE paketi oluşturur ve bu paketi diğer düğümlere gönderir.
 - UPDATE paketi alındığında, her düğüm yüksekliğini bu paketi gönderen komşusunun yüksekliğinden daha yüksek bir değere ayarlar. Bu işlem, yönlendirilmiş bağlantılarla hedefe doğru bir yol oluşturur.
- Rotaların bakımının yapılması:
 - Bir düğüm, hedefe olan yolun artık geçerli olmadığını fark ettiğinde, yüksekliğini yerel maksimum olarak ayarlayıp komşularına bir UPDATE paketi gönderir.
 - Eğer bir düğüm, hedefe yönelik geçerli bir yola sahip komşusu kalmadığını görürse, yeni bir yol keşfetmek için QUERY paketini tekrar yayar.
- Geçersiz rotaların silinmesi:
 - Ağda bölünmeler olduğunda, bir CLEAR paketi yayınlanır ve bu paket, tüm rotaları sıfırlayarak yönlendirme tablolarını temizler.

Başlangıçta bir rota oluşturmak için, kaynak düğüm komşularına bir QUERY paketi gönderir. Bu QUERY paketi, hedef düğüme veya hedefe giden bir rota bilgisine sahip ara düğüme ulaşana kadar ağda yeniden iletilir. QUERY paketinin alıcısı, kendi yüksekliğini hedefe göre listeleyen bir UPDATE paketi oluşturur ve bu paketi ağ üzerinden yayar. UPDATE paketi alındığında, her düğüm kendi yüksekliğini, paketi gönderen komşunun yüksekliğinden daha büyük bir değere ayarlar. Bu işlem, QUERY paketinin orijinal göndericisinden, hedefe ulaşan yönlendirilmiş bir bağlantı zinciri oluşturmaktadır. Bir

düğüm, hedefe giden yolun artık geçerli olmadığını tespit ettiğinde, komşularına göre yerel maksimum olacak şekilde kendi yüksekliğini günceller ve ardından bir UPDATE paketi iletir. Eğer düğüm, hedefe giden geçerli bir yola sahip komşusu olmadığını fark ederse, yeni bir yol bulmak için tekrar rota keşfine başlamaktadır [56], [57].



Şekil 2.9. TORA' da rota oluşturma [57].

Şekil 2.9'da TORA protokolünde örnek bir rota oluşturma işlemi gösterilmektedir. Buna göre, rota keşfi sırasında, bir QUERY paketi kaynak düğümden (düğüm 1) başlar ve hedef düğümüne (düğüm 9) ulaşana kadar ağdaki diğer düğümler arasında yayılır. Ancak, düğüm 6 gibi bir düğüm, aynı QUERY paketini birden fazla kez işlemekten kaçınmaktadır. Örneğin, düğüm 6, düğüm 5'ten gelen QUERY paketini, daha önce düğüm 4'ten aldığı ve işlediği için yeniden iletmez.

UPDATE paketleri, hedef düğümden (düğüm 9) başlatılır ve QUERY paketinin geçtiği yollar boyunca kaynak düğümüne doğru geri yayılır. Bu süreçte her düğüm, kendisine gelen UPDATE paketiyle bir yükseklik değeri atar. Paketler, ağda döngüsüz yönlendirme sağlayan DAG mekanizmasına göre hareket eder. Şekil 2.9'da görüldüğü gibi, düğüm 1 en yüksek yükseklik değerine (0,3) sahipken, hedef düğüm, düğüm 9 en düşük yükseklik

değerine (0,0) sahiptir. Bu düzenleme sayesinde paketler, en yüksek yükseklikten en düşük yüksekliğe doğru bir ‘yukarıdan aşağıya’ yaklaşımıyla hareket etmektedir.

Bir düğüm, ağda bir bölünme tespit ettiğinde, tüm yönlendirme tablolarını sıfırlayan bir CLEAR paketi oluşturur. Bu işlem, ağdaki tüm rotaların sıfırlanmasını sağlar ve yeni bir rota keşif süreci başlatır. DAG mekanizmasına dayalı olarak rota oluşturma, ağdaki tüm yolların döngüsüz olmasını garanti etmektedir [57].

2.1.2. Hibrit Yönlendirme Protokolleri

Hibrit yönlendirme protokolleri hem reaktif hem de proaktif protokollerin güçlü yönlerini birleştirerek esnek ve etkili bir yönlendirme mekanizması sunmaktadır. Bu protokoller, ağdaki kaynak ve hedef düğümlerin konumlarına ve bölgesel yapısına bağlı olarak dinamik bir şekilde uyum sağlar. Bölgesel komşuların belirlenmesi için proaktif yöntemler kullanılırken, bu bölgeler arasındaki yönlendirme yolları reaktif yöntemlerle oluşturulmaktadır. Bu yaklaşım, özellikle dinamik ve büyük ölçekli ağlarda ağ kaynaklarının daha verimli bir şekilde kullanılmasını sağlamaktadır [58].

Bu çalışmada hibrit yönlendirme protokolleri, hiyerarşik ve konum tabanlı olmak üzere iki ana başlık altında incelenmektedir.

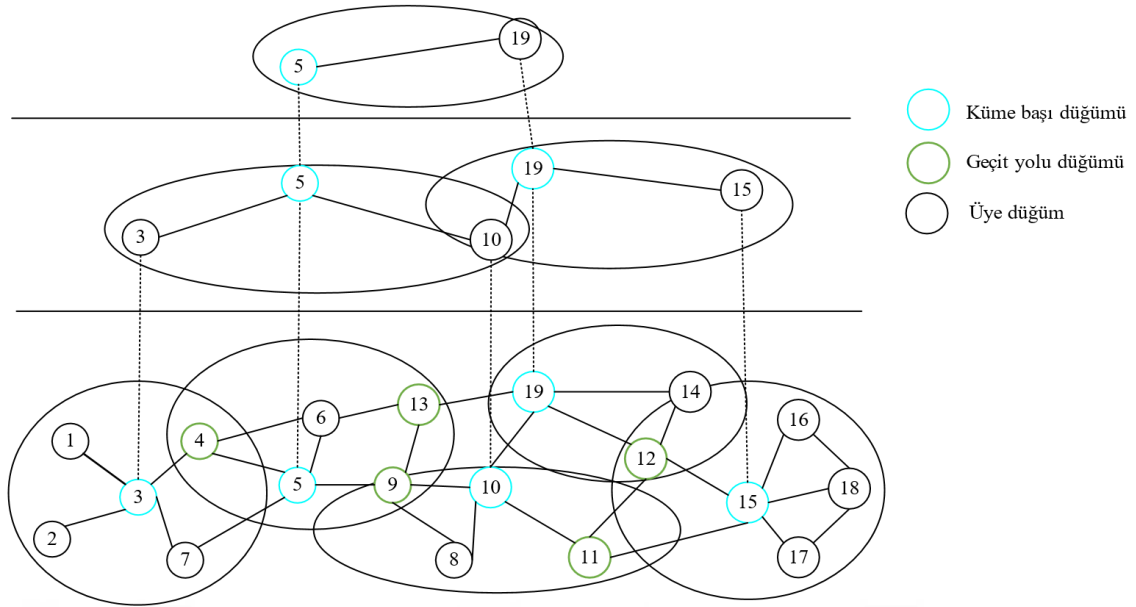
2.1.2.1. Hiyerarşik Yönlendirme Protokolleri

Hiyerarşik ad-hoc yönlendirme protokolleri, ağdaki düğümleri organize etmek ve yönlendirme sürecini daha ölçeklenebilir hale getirmek amacıyla bir hiyerarşi oluşturur. Bu hiyerarşi genellikle kümeleme teknikleri kullanılarak yapılandırılır. Hiyerarşinin üst seviyelerinde yer alan düğümler, yönlendirme sürecini optimize eden ve ağın verimliliğini artıran özel hizmetler sunar. Hiyerarşik yapı sayesinde, bir rota arızası durumunda tüm yolun yeniden hesaplanmasına gerek kalmaz; bunun yerine, yalnızca ilgili bölüm güncellenmektedir [59].

Literatürde en çok geçen hiyerarşik yönlendirme protokollerinden bazıları: Hiyerarşik durum yönlendirme (HSR), Çekirdek Tabanlı Dağıtık Ad-Hoc Yönlendirme (CEDAR), Küme Ağ Geçidi Anahtar Yönlendirme Protokolü (CGSR) protokolleridir.

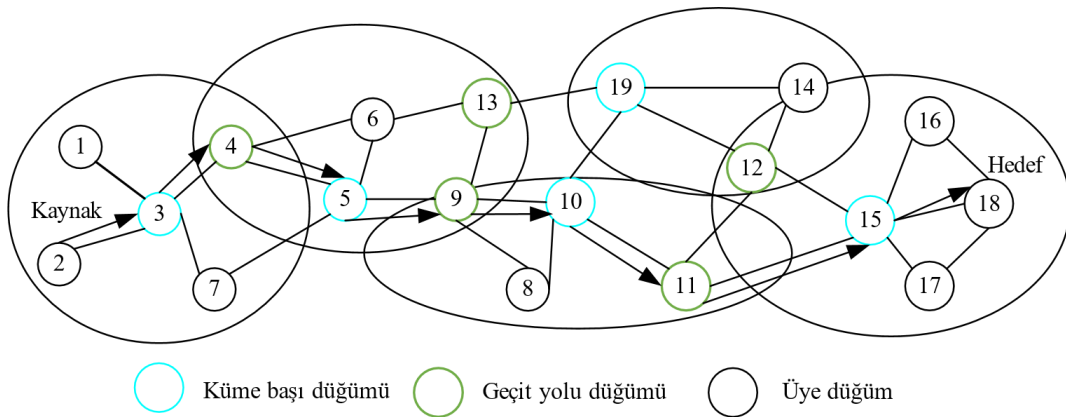
2.1.2.1.1. HSR

Iwata ve arkadaşları, çok düzeyli kümelemeye dayalı bir protokol sınıfı geliştirmiştir. Bu protokolün temel amacı, kontrol bilgilerinin ağ genelinde yayılmasını sınırlandırmak, bu bilgileri yerel KB’lerde toplamak ve ardından diğer KB’lere dağıtmaktır.



Şekil 2.10. HSR protokol yapısı [45].

HSR, düğümlerin fiziksel yakınlıklarına bağlı olarak, Şekil 2.10'da görüldüğü gibi, alt düzeyinde tek atlamalı mesafelere dayalı kümeler oluşturur ve bu kümeler için bir KB seçer. Orta düzeyinde, KB'ler sanal bağlantılar kullanarak birbirine bağlanır ve daha üst düzeydeki kümeler halinde organize edilir. Üst düzeydeki kümelerde bulunan KB'ler bir araya gelerek ağ geçidi düğümlerini seçmektedirler. Bir düğümün konumu değiştiğinde, bağlantı durumu bilgileri sanal bağlantılar aracılığıyla KB'leri arasında paylaşılır. KB'ler, küme içindeki düğümlerin bağlantı durumu bilgilerini toplar ve bu bilgileri ağ geçidi düğümleri üzerinden diğer KB'lere iletmektedirler [45], [60].



Şekil 2.11. HSR paket gönderme [61].

HSR yönlendirme işlemleri, Şekil 2.11'de görüldüğü gibi, KB'lerinin yönlendirici işlevi gördüğü hiyerarşik bir adresleme şeması kullanılarak gerçekleştirilir. Bir düğüm paket

göndermek istediğinde, paketi önce kendi yerel KB'ye iletir. KB, hedef düğümü arar ve paketi en yakın ağ geçidi düğümüne yönlendirir. Ağ geçidi düğümü, paketi hiyerarşinin bir sonraki düzeyindeki en yakın ağ geçidine veya KB'ye iletmektedir. Bu işlem, paket hedef kümenin ağ geçidi düğümüne ulaşana kadar devam eder. Son olarak, hedef kümenin ağ geçidi düğümü, paketi hedef KB'ye yönlendirir ve KB'de paketi hedef düğümüne teslim etmektedir [60], [61].

2.1.2.1.2. CEDAR

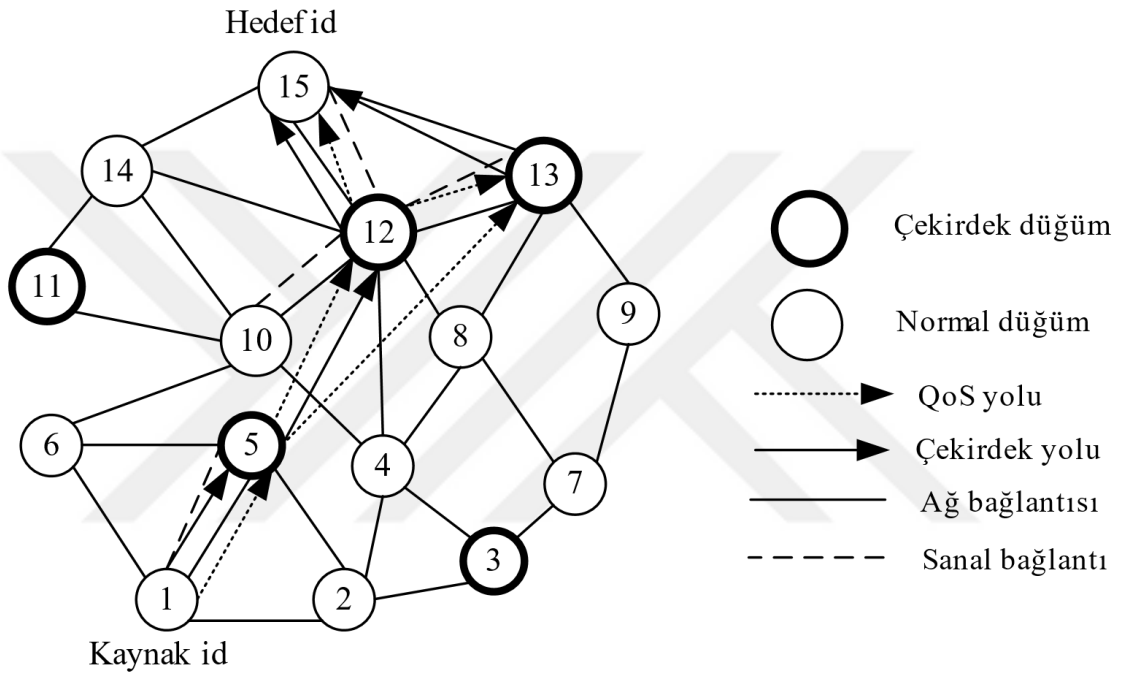
Sivakumar ve çalışma arkadaşları tarafından geliştirilen CEDAR protokolü, ad hoc ağlarda Servis Kalitesi (QoS) gereksinimlerini karşılayabilecek bir yapı sunar ve ağda çekirdek düğümler olarak adlandırılan özel düğümlerin oluşturulmasına dayanır. Protokol, ağdaki düğümlerin bir alt kümesini seçerek ağın çekirdeğini oluşturan minimum bir Dominatör Kümesi (DS) tanımlar. DS, ağdaki her düğümün ya DS içerisinde yer almasını ya da DS'deki bir düğümün komşusu olmasını sağlayacak şekilde oluşturulan bir düğüm kümesidir. CEDAR protokolünde, her üç atlamada bir çekirdek düğüm bulunmaktadır. Bir çekirdek düğümü seçen düğümler, bu çekirdek düğümün çekirdek üyesi olarak adlandırılır. Çekirdek düğümler arasındaki bağlantılar ise sanal bağlantılar olarak ifade edilmektedir [62].

CEDAR'da rota oluşturma süreci iki aşamada gerçekleştirilir:

- Birincil rota keşfi:
 - İlk aşamada, kaynaktan hedefe bir temel rota bulunur. Bu rota, kaynak düğümün çekirdeğinden (kaynak çekirdek) hedef düğümün çekirdeğine (hedef çekirdek) uzanan yol olarak tanımlanmaktadır.
- QoS yolunun seçimi:
 - İkinci aşamada, çekirdek yol üzerinde hizmet kalitesi gereksinimlerini karşılayan uygun bir QoS yolu belirlenmektedir.

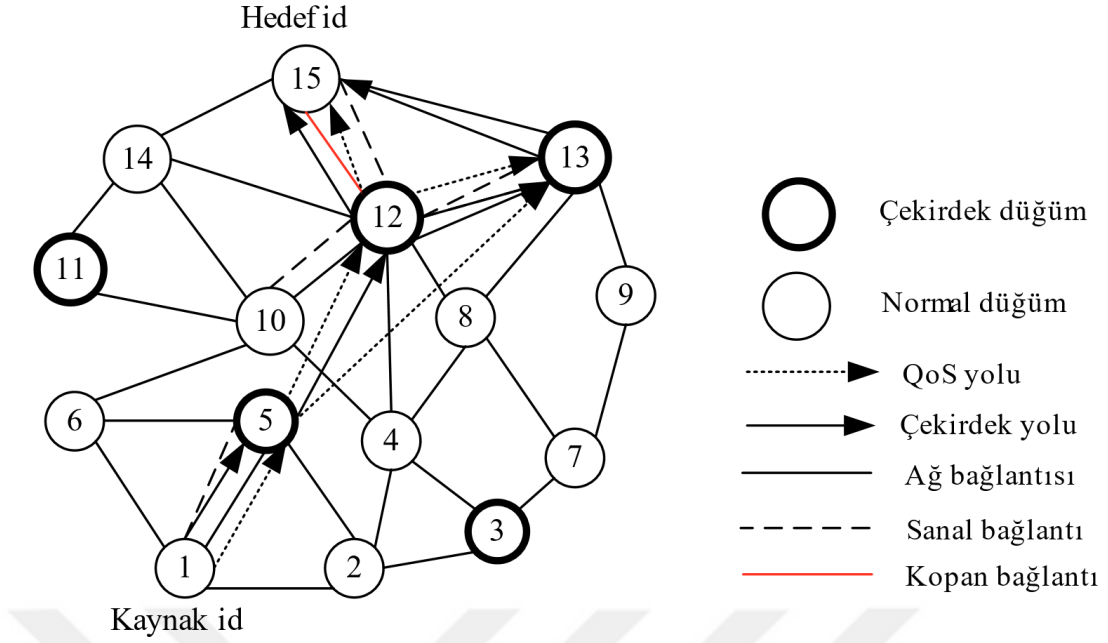
CEDAR mimarisinde yönlendirme, kaynak düğümden başlayarak gerçekleştirilir. Kaynak düğüm, kaynağı, hedefi ve talep edilen bant genişliği bilgisini içeren bir yol talebi paketini yerel çekirdek düğümüne iletir. Şekil 2.12'de gösterildiği gibi, bu talep, dominator düğüm tarafından değerlendirilir ve mümkünse bir QoS rotası hesaplanarak kurulur. Her kümedeki dominator düğümleri, yerel bağlantı durumu bilgilerini saklar ve sanal bağlantılar üzerinden diğer dominator düğümleriyle iletişim kurar. Rota

hesaplaması yalnızca ana yol üzerinde gerçekleştirilir. Ana yol belirlendikten sonra, QoS gereksinimlerini karşılayan uygun bir yol seçilir. CEDAR, rota hesaplamalarında optimallikten ziyade sağlamlığı önceliklendirir. Çekirdek düğümler, yalnızca komşu çekirdek düğümleri hakkında bilgi sahibidir ve küresel ağ yapısı hakkında detaylı bilgiye sahip değildir. Bu durum, düğümlerin hareket etmesi veya başarısız olması durumunda çekirdek ağın bakımını kolaylaştırır. Çekirdek yollar, yalnızca bağlantı talepleri alındığında isteğe bağlı olarak oluşturulur. Yol hesaplama işlemi ise yalnızca bir yol talebi geldiğinde gerçekleştirilir [59], [62].



Şekil 2.12. CEDAR’da rota kuruluşu [45].

CEDAR protokolünde, Şekil 2.13’te gösterildiği gibi, bir rota kesintisi meydana geldiğinde kesilen rota yerel olarak onarılmaya çalışılmaktadır.



Şekil 2.13. CEDAR’ da rota bakımı [45].

Kesinti sonrasında bir düğüm şu işlemleri gerçekleştirmektedir:

- Bir arıza bildirim mesajı gönderir.
- Hedefe yeni bir rota bulmak için arama sürecini başlatır.
- Hedefe giden yeni yol bulunana kadar gelen tüm paketleri reddeder.

Aynı zamanda, kaynak düğüm bir arıza bildirimini aldığı anda:

- Veri iletimini durdurur.
- Hedef düğüme yeni bir rota bulmaya çalışır.

Eğer bu yeni rota, kaynak veya hedef düğümden biri tarafından başarıyla oluşturulursa, kaynak ile hedef arasında yeni bir yol kurulur [59], [62].

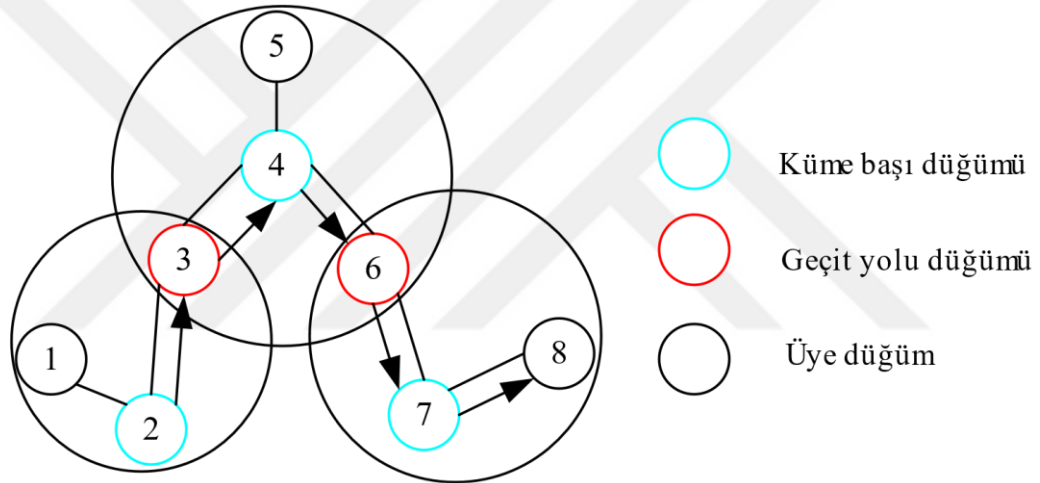
2.1.2.1.3. CGSR

Dinamik ağ ortamlarında KB düzenlemeleri, sık KB seçimleri nedeniyle performans düşüşlerine yol açmaktadır. Bu durumu ele almak için Chiang ve arkadaşları tarafından geliştirilen CGSR protokolü, En Az Küme Değişikliği (LCC) adlı dağıtılmış bir algoritma kullanır. CGSR, düğümleri KB’ler tarafından kontrol edilen kümeler halinde organize ederek kanal erişimi, bant genişliği tahsisi ve yönlendirme için ek özellikler sunar. Bu yöntemde düğümler, KB’ler aracılığıyla diğer kümelerle iletişim kurar. LCC algoritması, komşu düğümler arasında en düşük ID’ye sahip olan mobil düğümü KB olarak

seçmektedir. Yeniden kümeleme işlemi olay tabanlıdır ve yalnızca belirli durumlarda gerçekleştirilir. Bu nedenle, LCC algoritması kararlı bir yapı olarak kabul edilmektedir [59], [63].

CGSR protokolünde yeniden kümeleme iki temel durumda meydana gelmektedir:

- İki KB'nin erişim menziline girmesi:
 - Bu durumda, KB'lerden biri KB rolünden vazgeçerek ağdaki yapı korunmaktadır.
- Bir düğümün KB'ye erişememesi:
 - Eğer bir mobil düğüm herhangi bir KB'ye erişemezse, ağ yapısı yeniden oluşturulur. Bu işlem, ağdaki en düşük ID'ye sahip düğümün yeni KB olarak atanmasıyla gerçekleştirilmektedir [64].



Şekil 2.14. CGSR protokolü paket iletimi [45].

CGSR protokolünde yönlendirme işlemi, KB'lerin DSDV protokolünü kullanarak birbirleriyle iletişim kurmasıyla gerçekleştirilir. İki küme arasındaki bağlantı, ağ geçidi düğümleri aracılığıyla sağlanır. Şekil 2.14'te gösterildiği gibi, bir kaynak düğümünden gönderilen bir paket önce en yakın KB'ye iletilir. Ardından, paketi alan KB, bunu ilgili ağ geçidine yönlendirir. Ağ geçidi düğümleri, birden fazla kümenin üyesi olabilir ve bu nedenle farklı KB'ler ile iletişim kurarken farklı Kod Bölmeli Çoklu Erişim (CDMA) kodları kullanır. Geçidi geçen paket, hedef kümenin KB'ye iletilene kadar diğer KB'lere doğru ilerlemeye devam etmektedir. Son aşamada, hedef kümenin KB'si paketi alır ve doğrudan hedef düğüme iletmektedir [65].

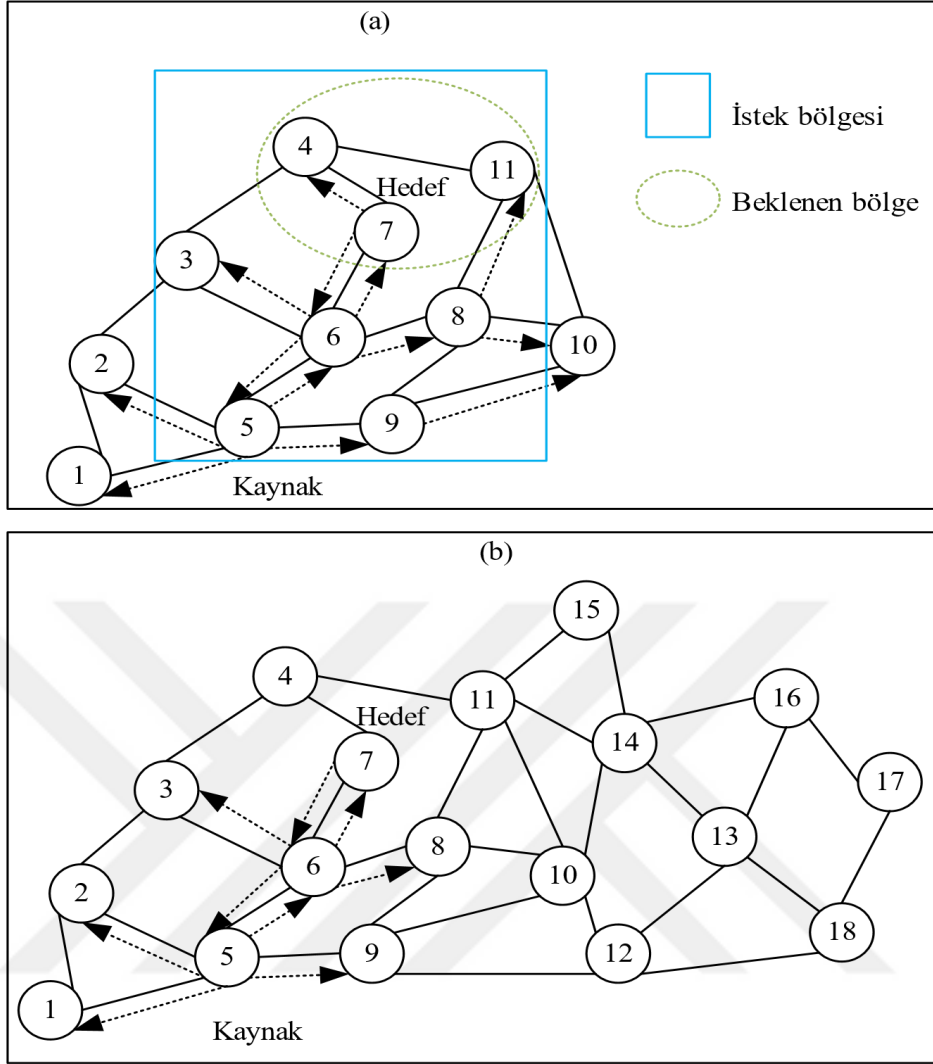
2.1.2.2. Konum Tabanlı Yönlendirme Protokolleri

Mobil ad hoc ağlarda kullanılan konum tabanlı yönlendirme protokolleri, bireysel düğümlerin ağdaki diğer tüm düğümlerin konum bilgilerine sahip olduğunu varsayar. Bu protokoller, daha iyi organize edilmiş ve ölçeklenebilir bir yapı sunmaktadır. Topoloji bilgisine minimum düzeyde ihtiyaç duyar ve yönlendirme tablolarını sürekli olarak güncelleme gereksinimini ortadan kaldırmaktadır. Bu tür algoritmalarda en yaygın ve kolay uygulamalardan biri, düğümlerin coğrafi konumlarını belirlemek için Küresel Konumlandırma Sistemi'nin (GPS) kullanılmasıdır. GPS tarafından sağlanan bu konum bilgisi, yönlendirme protokolleri tarafından rota belirleme işlemlerinde kullanılmaktadır. Literatürde en çok bilinen coğrafi yönlendirme protokolleri arasında Konum Destekli Yönlendirme (LAR), Hareketlilik İçin Mesafe Yönlendirme Etkisi Algoritması (DREAM) ve kablosuz ağlar için Açgözlü Çevre Durumsuz Yönlendirme (GPSR) gibi protokoller öne çıkmaktadır [59], [66].

2.1.2.2.1. LAR

Ko ve Vaidya, hedef düğüme doğru rota keşfi sırasında arama alanını daraltmak için konum bilgisini kullanan LAR protokolünü geliştirmiştir. Bu protokol, rota keşfi sürecinde yönlendirme ek yükünü azaltmayı hedefler ve düğümlerin konum bilgilerini elde etmek için GPS'den yararlanmaktadır. LAR, kontrol paketlerinin yönlendirilmesini kolaylaştırmak ve gereksiz paket iletimlerini önlemek amacıyla Şekil 2.15 (a)'da gösterildiği gibi iki farklı bölge bulunmaktadır: Beklenen bölge ve istek bölgesi. Bu bölgeler, protokolün rota keşfi sırasında daha etkin bir yönlendirme sunmasını sağlar [67], [68].

- Beklenen Bölge:
 - Hedef düğümün geçmiş konum bilgileri ve hareketlilik durumu dikkate alınarak tahmin edilen coğrafi bölgedir. Eğer hedef düğümün önceki konumu hakkında herhangi bir bilgi mevcut değilse, tüm ağ bu bölge olarak kabul edilir.
- İstek Bölgesi:
 - Bir kaynak düğüm, bir paketi göndermeden önce hedef düğümün olası konumunu belirler. Kaynak düğüm, yol istek paketlerini göndererek bu coğrafi alanı istek bölgesi olarak adlandırır. Yol istek paketleri, yalnızca istek bölgesi içerisindeki düğümler tarafından iletilir. Eğer hedef düğüm ilk istek bölgesi içinde bulunamazsa, istek bölgesinin boyutu genişletilir [69].



Şekil 2.15. (a) LAR protokolünün bölgeleri ve LAR 1 yönlendirme durumu
(b) LAR 2 yönlendirme durumu [45].

Düğüm, bir paketin iletilip ileilmeyeceğini belirlemek için LAR1 ve LAR2 olarak adlandırılan iki farklı algoritmaya göre hareket etmektedirler [67].

- LAR1: Şekil 2.15 (a)'da gösterildiği gibi, kaynak düğüm, hedef düğümün konumuna bağlı olarak bir istek bölgesi oluşturmaktadır. Bu bölge, hedef düğümün başlangıç konumunu, mevcut hızını ve yönünü dikkate alarak belirlenir. İstek bölgesi, hedef düğümün hareket hızına bağlı bir yarıçapla oluşturulmuş bir dairenin etrafını saran en küçük dikdörtgen şeklinde tanımlanmaktadır. Dairenin merkezi, hedef düğümün başlangıç konumudur. Kaynak düğüm, istek bölgesinin sınırlarını ve koordinatlarını içeren yol talep paketlerini gönderir. Bu paketleri alan düğümler, kendi konumlarının istek bölgesi içinde olup olmadığını kontrol eder. Eğer düğüm istek bölgesinin içinde ise paketi iletir, aksi takdirde paketi iletmez.

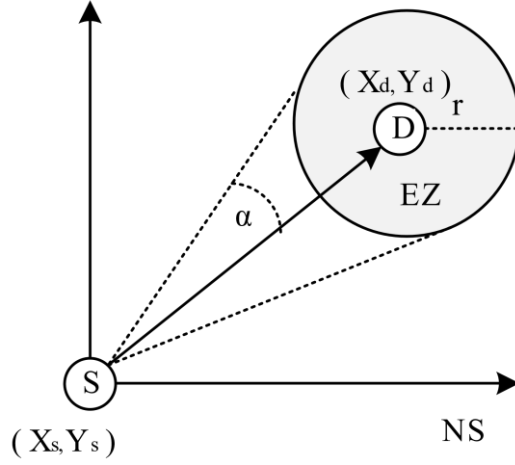
- Şekil 2.15 (b)'de gösterildiği üzere, kaynak düğüm GPS koordinatlarını kullanarak kendisi ile hedef düğüm arasındaki mesafeyi hesaplar. Bu mesafe bilgisi, yol talep paketlerine dahil edilir. Bu paketi alan bir ara düğüm, hedef düğüme olan mesafesini hesaplar. Eğer ara düğümün hedefe olan mesafesi, kaynak düğümünden hedefe olan mesafeden daha az ise, ara düğüm bu paketi istek bölgesinde kabul eder ve komşularına iletir. Ancak, hedefe daha yakın değilse paketi iletmez. Ara düğüm tarafından güncellenen mesafe bilgisi, her bir düğümde hedefe olan uzaklığı daha da azaltacak şekilde pakete eklenir. Bu süreç, hedef düğüme en uygun yolun belirlenmesini sağlamaktadır [59], [67], [69].

2.1.2.2.2. DREAM

Basagni ve arkadaşları, düğümlerin GPS sistemi aracılığıyla elde edilen konum bilgilerini kullanarak yönlendirme ek yükünü optimize etmek amacıyla DREAM protokolünü geliştirmiştir [70]. Bu protokolde, her düğümün GPS ile belirlenen coğrafi koordinatları periyodik olarak diğer düğümlerle paylaşılmakta ve bu bilgiler yönlendirme tablolarında saklanmaktadır. Bu yaklaşım, tam bağlantı durumu veya mesafe vektör bilgisinin değiş tokuş edilmesine kıyasla önemli ölçüde daha az bant genişliği tüketmekte ve böylece protokolün ölçeklenebilirliğini artırmaktadır [71].

DREAM protokolü, güncelleme mesajlarının gönderim sıklığını düğümlerin hareketliliği ve aralarındaki mesafeyle orantılı hale getirerek yönlendirme ek yükünü daha da azaltmayı hedefler. Sabit düğümlerin herhangi bir güncelleme mesajı göndermesi gerekmez, bu da ağdaki gereksiz kontrol mesajlarını en aza indirmektedir [72].

Bu algoritmada, paket iletimi Şekil 2.16'da gösterildiği üzere gerçekleştirilir. Kaynak düğüm (S), paketi hedef düğüme (D) göndermek için öncelikle konum tablosunu kontrol ederek hedef düğümün coğrafi konumu hakkında bilgi edinmektedir. Elde edilen bu bilgilere dayanarak, hedef düğümün yönü geçerli ise, paketi hedef yönündeki birinci derece komşularına (1-hop uzaklıktaki düğümlere) göndermektedir. Paketi alan düğüm, öncelikle kendisinin hedef düğüm olup olmadığını kontrol eder. Eğer bu düğüm hedef düğüm ise, kaynak düğüme (S) bir "paket alındı" bildirimini göndermektedir. Hedef düğüm değilse, paketi hedef yönündeki en yakın komşularına iletmektedir. Bu işlem, paket hedef düğüme ulaşınca kadar her düğüm tarafından tekrarlanmaktadır. DREAM protokolü, bu yönüyle reaktif algoritmalara benzerlik göstermektedir [73].



Şekil 2.16. DREAM algoritmasının mesafe etkisi, [74].

DREAM protokolünde mesafe etkisi, Şekil 2.16’da gösterildiği gibi açıklanmaktadır. S düğümü, D için yönlendirme bölgesini belirlemek amacıyla, D’nin etrafında bir daire olarak tanımlanan Beklenen Bölge’yi (EZ) hesaplar.

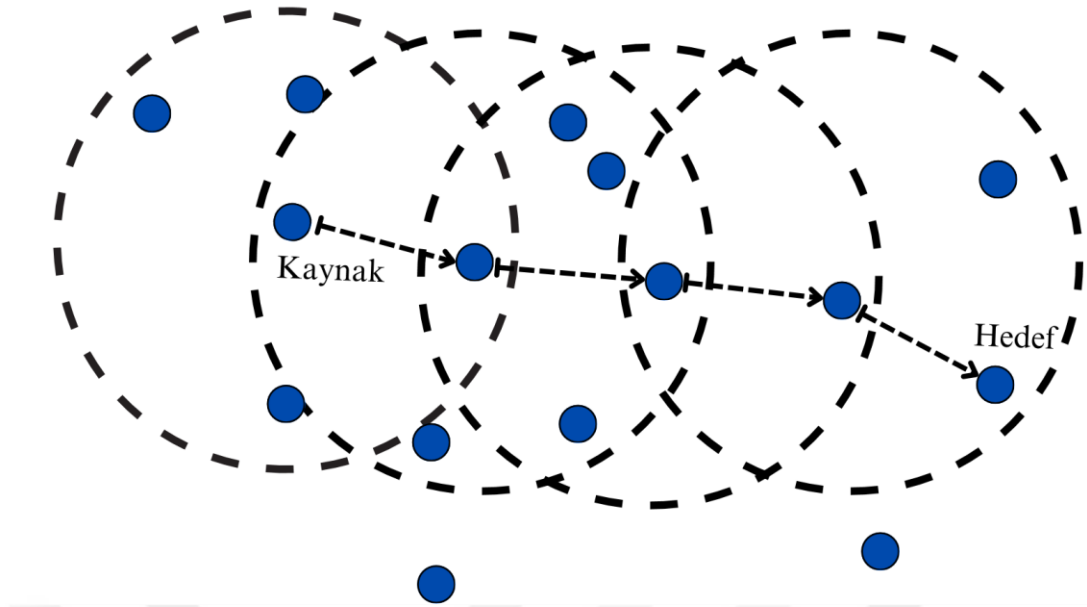
Bu bölgenin yarıçapı $r = (t_1 - t_0)v_{max}$ formülüyle ifade edilmektedir. Burada:

- t_0 : S düğümünün, D düğümüne ait konum bilgisinin zaman damgasını,
- t_1 : Güncel zamanı,
- v_{max} : D düğümünün ad hoc ağda bilinen maksimum hareket hızını temsil etmektedir.

S düğümü, D düğümü için hesapladığı EZ’ye teğet bir açıyla yönlendirme bölgesini tanımlar. Daha sonra, paketi yönlendirme bölgesindeki en yakın komşularına iletmekte ve bu sayede D’ye doğru bir yol oluşturmaktadır [74].

2.1.2.2.3. GPSR

GPSR protokolü, literatürde konuma dayalı protokoller arasında öne çıkan bir yaklaşımdır. Bu protokolda her düğüm, komşularına kendi tanımlayıcısını ve konumunu içeren “işaret” adı verilen bir mesajı periyodik olarak gönderir. Bu mesajlar, düğümlerin 1-hop mesafesindeki komşularının konumlarını bilmesine yardımcı olur ve yerel düğüm bilgileri oluşturur [75]. Bu mekanizma sayesinde paketlerin iletimi, yerel bilgilere ve paket başlığındaki konum bilgisine dayanarak gerçekleştirilmektedir.



Şekil 2.17. GPSR Paket iletim sistemi [69].

GPSR protokolü, iletişim mesajlarının iletiminde iki farklı mod kullanır:

- Açgözlü Mod:
 - GPSR protokolü, paketi öncelikle açgözlü modda iletmeyi dener. Bu modda, her düğüm, komşularının coğrafi konum bilgilerini kullanarak, paketi bir sonraki sıçramada hedefe en yakın olan komşuya yönlendirir. Böylece paket, hedefe ulaşana kadar art arda coğrafi olarak daha yakın düğümlere iletilir. Diğer bir ifadeyle, kaynak düğüm, veri paketini hedef düğüme olan mesafesi daha kısa olan bir ara düğüme göndermektedir. Ara düğüm, Şekil 2.17’de gösterildiği gibi aynı işlemi tekrarlayarak paketi hedef yönünde iletmektedir.
- Çevre Modu:
 - Açgözlü iletimin mümkün olmadığı bir bölgeye, yani “yerel maksimum alan” olarak bilinen bir yere ulaşıldığında, protokol çevre moduna geçer. Çevre moduna geçen bir paket, başlık kısmına konum bilgisi kaydedilerek döngülerin oluşması engellenmektedir. Bu modda paketler, düzlemsel bir grafikte sağ el kuralına göre ilerletilir. Eğer açgözlü modda iletim tekrar mümkün hale gelirse, protokol çevre modundan çıkarak açgözlü moda geri dönmektedir [69].

Bu bölümde anlatılan yönlendirme protokollerinin özeti, avantaj ve dezavantajları Çizelge 2.1’de gösterilmektedir.

Çizelge 2.1. Yönlendirme protokollerinin özeti, avantaj ve dezavantajları.

Düz Yönlendirme Protokolleri		Proaktif (Tabloya dayalı) Yönlendirme Protokolleri			
		Protokol	Protokol Özeti	Avantajlar	Dezavantajlar
		DSDV	Bellman-Ford algoritmasının optimize edilmiş bir versiyonuna dayanır. Her düğüm, ağdaki tüm cihazları kapsayan bir yönlendirme tablosu tutar. Yönlendirme bilgilerini ağ genelinde belirli aralıklarla günceller. Döngüleri önler ve güvenilir rotalar oluşturur.	Döngüsüz yönlendirme sağlar. Proaktif yapı sayesinde düşük gecikme. Küçük ve orta ölçekli ağlarda verimlidir. Uygulaması basit ve hesaplama yükü düşüktür. Düzenli güncellemelerle sürekli güncel yönlendirme tabloları sunar.	Sürekli güncellemeler nedeniyle ek yük fazladır. Dinamik ağlarda performans düşüktür. Büyük ağlarda ölçeklenebilirlik zordur. Düğümlerin enerji tüketimi yüksektir. Ağın ilk kurulumu gecikmelere yol açabilmektedir [76], [77].
		WRP	DSDV'ye benzer şekilde tablo yapısı kullanır, ancak döngü içermeyen bir yapıya sahiptir. Mesafe, bağlantı maliyeti ve yönlendirme bilgilerini içeren dört tablo tutar. Yönlendirme ek yükünü azaltır ve "sonsuzluğa kadar sayma" probleminde çözüm sunar. Her hedef için en kısa rotaları belirler.	Döngüsüz yönlendirme sağlar. Bağlantı kesintilerini hızlı algılar. Doğru ve güncel yönlendirme tabloları sunar. Güncellemeleri hızlı ve etkili bir şekilde yapar.	Yüksek bellek ve işlem gücü gereksinimi. Kontrol mesajı trafiği yüksektir. Dinamik topolojilerde performans düşülebilmektedir. Küçük ölçekli ağlar için fazla karmaşıktır [2], [77].
		OLSR	MPR yöntemi ile kontrol mesajı yükünü azaltır. MPR düğümleri, yalnızca seçilen düğümler arasında veri aktarımı yapar. Bağlantı durumu bilgileri yalnızca MPR düğümleri tarafından raporlanır. En kısa yol algoritması ile optimal rotalar hesaplanır. En kısa yol algoritması ile veri iletimi için en iyi rota belirlenir. Veri iletimi gerektiğinde tüm düğümlere giden yollar önceden hesaplanmış olur ve veri, mümkün olan en kısa yoldan iletilir.	MPR yöntemi kontrol mesaj yükü ve düğüm enerji tüketimi düşüktür. Dinamik topolojilere hızlı uyum sağlamaktadır. Geniş ölçekli ağlar için uygundur. Düşük uçtan uca gecikme sağlar. Hızlı kaynak ve hedef değişikliklerine sahip ağlarda etkilidir. Büyük düğüm grupları arasında iletişimi optimize eder.	Düğüm sayısı arttıkça kontrol mesajı yükü artar. Alternatif rota keşfinde yüksek işlem gücü gerektirir. Sabit kontrol mesajı yükü, düşük hareketlilikte gereksiz enerji harcamasına sebep olmaktadır. Ağ büyüklüğüne bağlı olarak depolama karmaşıklığı artar [49], [78].
		STAR	Her düğüm, ağdaki komşularına güncelleme mesajları gönderir. Bu mesajlar yeni hedefler, döngü olasılıkları ve yolların maliyet bilgilerini içerir. Her düğüm, ağdaki tüm hedeflere ulaşmak için gereken bağlantıları kaynak yönlendirme ağacıyla komşularına iletir. "Yol yok" mesajları ve kaynak ağacı yanıtlarıyla, bağlantı kesintilerini yönetmek için bakım mekanizması sağlar.	Kontrol mesajı yükü düşüktür. Dinamik topolojilere uyumludur. Bağlantı kesintisi bakım mekanizması vardır.	Ağın kurulum aşamasında ek yük fazladır. Hızlı topoloji değişikliklerinde performans düşebilir. Yüksek bellek gereksinimi olmaktadır. Küçük ağlarda gereksiz bir karmaşık yapıya sebep olmaktadır [45], [50].

Çizelge 2.1. (devam) Yönlendirme protokollerinin özeti, avantaj ve dezavantajları.

Düz Yönlendirme Protokolleri		Protokol	Protokol Özeti	Avantajlar	Dezavantajlar
		Reaktif (İsteğe Bağlı) Yönlendirme Protokolleri		AODV	<p>Kaynak düğüm, hedefe bir rota bulmak için RREQ paketi yayımlar.</p> <p>Hedef düğüm, alınan ilk RREQ'ye yanıt olarak RREP gönderir.</p> <p>Ara düğümler, alınan pakete göre hedefe giden ters rotaları kaydeder.</p> <p>Bağlantı hatası algılandığında, düğüm bir RERR paketi oluşturur ve yayar.</p> <p>RERR paketi kaynak düğüme ulaşırsa, kaynak düğüm yeni bir yol keşif sürecini başlatır.</p> <p>Paketlerin izleyeceği yol, kaynak tarafından önceden belirlenmez.</p> <p>Kaynak ve ara düğümler yalnızca bir sonraki düğümün adres bilgisini tutar.</p>
		DSR	<p>Kaynak düğüm, gönderdiği paketin izleyeceği tüm yolu önceden belirler ve bu bilgiyi paketin başlığına ekler.</p> <p>Her düğüm, farklı hedeflere ait yolları saklayan bir yol önbelleği tutar.</p> <p>Kaynak düğüm, hedefe bir rota bulmak için öncelikle yol önbelleğini kontrol eder.</p> <p>Eğer önbellekte uygun bir yol bulunmazsa, bir Rota Talebi (RREQ) mesajı gönderir.</p> <p>Uygun rota keşfedildiğinde, paket hedefe iletilir.</p> <p>DSR, periyodik güncelleme yapmaz, bu da yönlendirme ek yükünü azaltır.</p>	<p>Yönlendirme tablosu gerekmez; yol bilgisi paket başlığında bulunur.</p> <p>Bir hedef için birden fazla rota bulunabilir, yük dengeleme ve güvenilirlik sağlar.</p> <p>Kontrol mesajı yükü azdır.</p> <p>Periyodik mesaj olmadığından, bant genişliğini az kullanır.</p> <p>Tek bir rota keşfi ile birden fazla rota bulunabilir.</p> <p>Rota önbelleği, keşif ek yükünü azaltır.</p>	<p>Büyük ağlarda ölçeklenemez ve yüksek işlem gücü gerektirir.</p> <p>Düğümler, kontrol verilerini işlemek için zaman harcar.</p> <p>Rota bakım mekanizması bağlantı hatalarını yerel olarak onaramaz.</p> <p>Bağlantı kurulum gecikmesi yüksektir.</p> <p>Yüksek hareketlilikte performans düşer.</p> <p>Rota taleplerinin yayılması ağda aşırı yük oluşturabilir [78].</p>
		TORA	<p>Rota oluşturma, Doğrudan DAG yapısına dayanır ve döngüsüz yollar sağlar.</p> <p>Rastgele bir yükseklik metriği kullanılarak birden fazla rota oluşturulur; ancak bunlar en kısa rotalar olmayabilir.</p> <p>QUERY ve UPDATE paketleri ile ağda rotalar oluşturulur.</p> <p>Düğümler, hedef yüksekliğine göre yerel ayarlamalar yapar.</p> <p>Geçerli olmayan yollar CLEAR paketiyle sıfırlanır.</p> <p>Paketler, yüksek yükseklığe sahip düğümden düşük yükseklığe sahip düğüme doğru iletilir.</p> <p>Yalnızca ihtiyaç duyulduğunda yol sorgulaması yapılarak kontrol mesajı yükü azaltılmaktadır.</p>	<p>Kaynak ve hedef arasındaki çoklu yolları destekler.</p> <p>Büyük mobil ağlarda etkili bir şekilde çalışır.</p> <p>Bağlantı kesintilerinde hızlı uyarlanabilirlik sağlamaktadır.</p>	<p>Yönlendirme ek yükü yüksektir.</p> <p>Rota oluşturma süresi uzun olduğu için gecikme fazladır.</p> <p>Küçük ağlar için karmaşıklık fazladır [57].</p>

Çizelge 2.1. (devam) Yönlendirme protokollerinin özeti, avantaj ve dezavantajları.

Hibrit Yönlendirme Protokolleri		Protokol	Protokol Özeti	Avantajlar	Dezavantajlar
		Hiyerarşik Yönlendirme Protokolleri		HSR	Düğüm, fiziksel yakınlığa göre tek-hop kümeler oluşturup KB'ler seçmektedir. KB'ler, sanal bağlantılarla daha yüksek seviyeli kümeler oluşturmaktadır. Ağ geçidi düğümleri, KB'ler arasında bağlantı sağlamaktadır. Paketler, önce yerel KB'ye, ardından hiyerarşik olarak ağ geçidi düğümleri üzerinden hedef KB'ye iletilmektedir. Düğüm hareketliliği durumunda bağlantı bilgileri sanal bağlantılarla güncellenmektedir.
		CEDAR	Çekirdek düğümlerden oluşan bir Minimum DS belirlemektedir. Çekirdek düğümler, sanal bağlantılarla iletişim kurarak, ağır çekirdeğini oluşturur. Kaynaktan hedefe temel yol oluşturulur. Çekirdek yol üzerinde uygun bir QoS yolu bulunur. Çekirdek düğümler, yerel bilgileri kullanarak QoS rotalarını hesaplamaktadırlar. Rota kesintileri durumunda, düğümler yerel olarak yolları onarmaya çalışmakta veya yeni bir yol keşfetmektedir.	QoS destekli yönlendirme sağlamaktadır. Çekirdek tabanlı yapı, kontrol mesajı yükünü azaltmaktadır. Rota kesintilerini yerel olarak onarabilmektedir. Hareketlilik ve topoloji değişikliklerine uyum sağlamaktadır. Çekirdek ağ yapısı bakımını basitleştirmektedir.	Çekirdek oluşturma ve sürdürme işlemleri ek yük getirmektedir. Küçük ağlarda etkili olsa da, büyük ve yoğun ağlarda performansı sınırlanabilmektedir. Rota oluşturma, alternatif rotalara göre daha fazla zaman alabilmektedir. Yüksek işlem gücü gerektirmektedir [77].
		CGSR	En Az Küme Değişikliği LLC algoritmasını kullanır. Komşular arasındaki en düşük ID değeri KB'leri belirlemektedir. Yeniden kümeleme yalnızca iki durumda gerçekleşir: KB'lerin yaklaşması veya bir düğümün KB'ye erişememesi. KB'ler, DSDV ile iletişim kurup paketleri ağ geçidi düğümleri üzerinden hedef kümenin KB'sine iletmektedir. Ağ geçidi düğümleri birden fazla küme arasında bağlantıyı sağlar ve farklı CDMA kodları kullanır. Sık küme değişikliklerini azaltarak kararlılığı artırmak ve kanal erişimi ile bant genişliği tahsisini optimize etmektedir.	Yönlendirme tablosu ve güncelleme mesajları küçüktür. Her düğüm yalnızca KB'ye yönelik yolları tuttuğundan, yönlendirme yükü DSDV ve WRP'ye göre daha düşüktür. Adresleme yapısı basittir.	Yol uzunluğunda artış olabilir. KB kararsız olabilir, özellikle yüksek mobilite durumlarında. KB'ler daha hızlı enerji tüketir. KB'lerin sık değişimi çoklu yola neden olabilmektedir [77].

Çizelge 2.1. (devam) Yönlendirme protokollerinin özeti, avantaj ve dezavantajları.

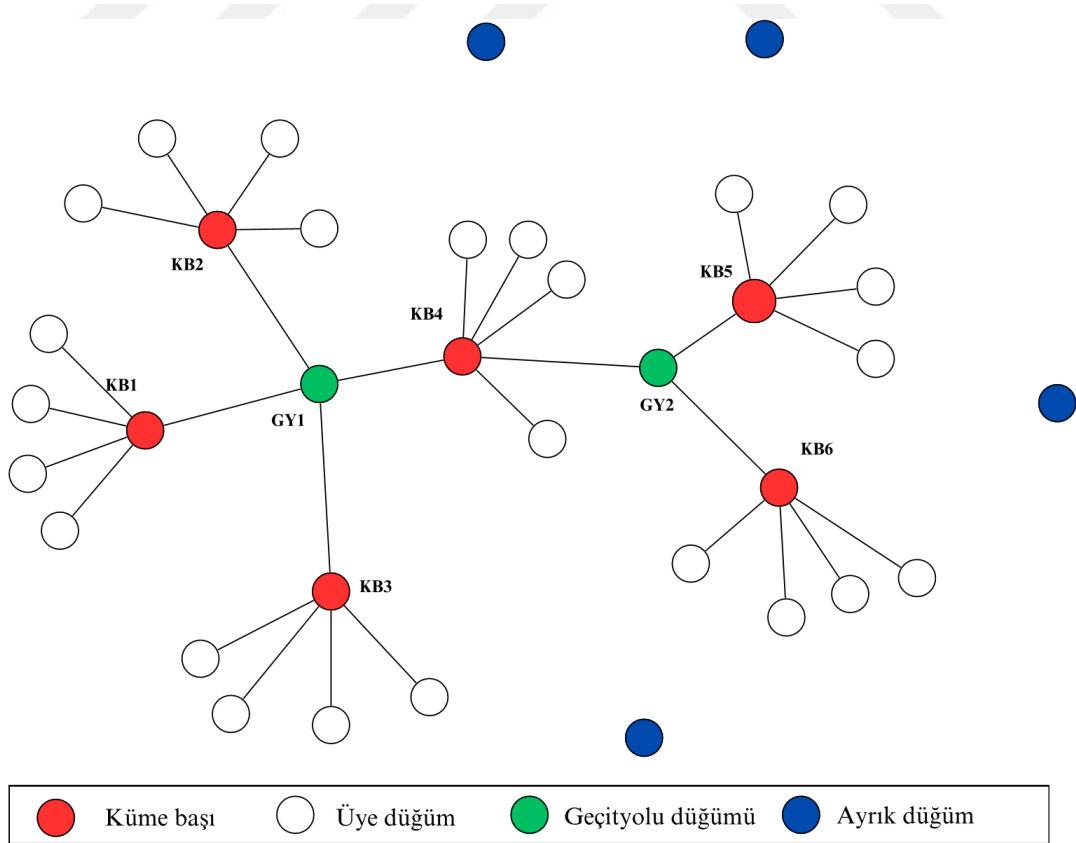
Hibrit Yönlendirme Protokolleri		Konum Tabanlı Yönlendirme Protokolleri			
		Protokol	Protokol Özeti	Avantajlar	Dezavantajlar
		LAR	<p>GPS tabanlı konum bilgisini kullanarak yönlendirme ek yükünü azaltmayı hedefleyen bir protokoldür. Hedef düğümün konumu, geçmiş konum ve hareketliliğe bağlı olarak tahmin edilir ve bu bilgiler doğrultusunda yönlendirme yapılacak alan sınırlandırılmaktadır. LAR protokolü iki farklı algoritma sunar: LAR1, hedefin beklenen bölgesini bir dikdörtgen olarak tanımlarken, LAR2, kaynak ve hedef düğüm arasındaki mesafeyi dikkate alarak paket iletimini gerçekleştirmektedir. Bu mekanizmalar sayesinde gereksiz paket iletimleri önlenir ve ağdaki yönlendirme verimliliği artırılmaktadır.</p>	<p>Yakın komşuların bilgisini içerdiğinden yönlendirme ek yükü düşüktür. Paket teslim oranı yüksektir. Döngü içermeyen yollar sunar ve verimli bir yönlendirme sağlar. Düğüm yoğunluğundan etkilenmez, yoğun ağlarda da performansı stabildir.</p>	<p>Komşu tabloların tutulması nedeniyle kontrol ek yükü fazladır. Yönlendirici düğümlerin enerji tüketimi diğer düğümlere göre daha yüksektir. İşlem yükü fazladır, çünkü konum hesaplamaları ve komşu tablosu güncellemeleri gereklidir [77], [80].</p>
		DREAM	<p>Her düğüm, GPS ile elde edilen konum bilgilerini periyodik olarak paylaşır ve bu bilgiler yönlendirme tablolarında saklanır. Hareketlilik ve mesafe etkisine göre güncelleme sıklığı ayarlanarak bant genişliği tasarrufu sağlanır. Paket iletiminde, kaynak düğüm hedefin konumunu belirler ve hedefe doğru en yakın komşusuna paketi gönderir. Mesafe etkisiyle EZ hesaplanır ve paketler bu bölge içindeki komşulara yönlendirilir. Bu yaklaşım, gereksiz paket iletimlerini önlerken ağın ölçeklenebilirliğini artırır.</p>	<p>Yönlendirme ek yükünü, mesafe etkisi ve hareketlilik oranını kullanarak azaltır. Sabit düğümlerin güncelleme mesajı göndermemesi yönlendirme ek yükünü azaltır. Döngü içermeyen yollar sağlamaktadır.</p>	<p>Mobilitesi çok olan düğümler daha sık konum güncelleme mesajı gönderir. Yüksek hareketliliği olan ağlarda ek yük artmaktadır [79].</p>
		GPSR	<p>GPSR, komşuların konum bilgisini içeren “işaret” adı verilen 1-hop mesajları kullanarak yerel bilgiye dayalı bir yönlendirme protokolüdür. Paket iletimi için iki mod kullanılır: Açgözlü Mod: Paketler, hedefine en yakın komşuya yönlendirilerek adım adım iletilmektedir. Çevre Modu: Paketler, açgözlü modun başarısız olduğu “yerel maksimum alan” durumunda, döngülerden kaçınmak için çevre modunda iletilmektedir.</p>	<p>Yüksek mobilite durumlarında başarılı paket teslimi sağlar. Düğüm yoğunluğu fazla olan ağlarda verimli çalışır. Verilerin başlık kısmında şeffaflık sağlar. Yerel maksimum durumları yönetebilmektedir. Yakın komşularının bilgisini içerdiğinden yönlendirme ek yükü azdır.</p>	<p>Kontrol karmaşıklığı yüksektir. Hedef düğüm aralığını tanımlamak zor olabilmektedir. İzleme problemlerinde artan maliyetlere yol açabilmektedir. Yönlendirici düğümlerin enerji tüketimleri fazladır. İşlem yükü fazladır [77], [81].</p>

2.2. MANETLERDE KÜMELEME

MANET, düğümlerin rastgele hareket ettiği, kablosuz bağlantıya sahip, kendi kendine organize olabilen ve yapılandırılabilen bir mobil düğüm ağıdır [82]. Her düğüm bağımsız bir şekilde hareket ederek veri iletimine katılır. Merkezi bir yönetimden uzak bu yapı, düğümlerin enerji tüketimini artırmakta, ağ ömrünü kısaltmakta ve paket iletim performansını olumsuz etkilemektedir. Literatürde bu sorunlara yönelik en etkili çözüm yöntemi olarak kümeleme öne çıkmaktadır [83].

Kümeleme, bir yönlendirme protokolü olmayıp, mobil düğümlere sahip ağların ölçeklenebilirliğini artıran ve düğümleri mantıksal gruplara ayırarak ağın genel verimliliğini yükselten bir mekanizmadır. Bu mekanizma, ağın karmaşıklığını ve dinamik yapısını etkili bir şekilde yönetir. Kümeleme sayesinde, mobil düğümler belirli bir düzene göre gruplandırılır ve bu grupların yönetiminden sorumlu düğümler seçilmektedir. Seçilen bu düğümler, iletişimi koordine ederek ağın enerji tüketimini optimize edip verimliliğini artırmaktadır [7], [84].

Şekil 2.18’de, MANET’teki kümeleme stratejisine göre mobil düğümlerin durumlarına göre üstlendikleri roller gösterilmektedir [85].



Şekil 2.18. MANET’te düğüm rolleri gösterimi.

Bu roller Őu Őekilde tanımlanmaktadır:

Küme BaŐı (KB): Küme içindeki iletiŐimi yöneterek veri toplama, iŐleme ve iletme görevlerini üstlenene düĐümlerdir.

Üye DüĐüm (ÜD): KB'ye veri gönderip alan küme üyesi düĐümlerdir.

Geçityolu DüĐümü (GD): Aynı anda en az iki KB'nin iletim aralıĐında bulunan düĐümlerdir.

Ayrık DüĐüm (AD): Hiçbir kümeye ait olmayan ve baŐka düĐümlerin iletim aralıĐı dıŐında kalan düĐümlerdir.

Bu yapı, kümeleme mekanizmasının düĐümler arası görev daĐılımını ve iletiŐim organizasyonunu etkili bir Őekilde yönetmesine olanak tanımaktadır [86].

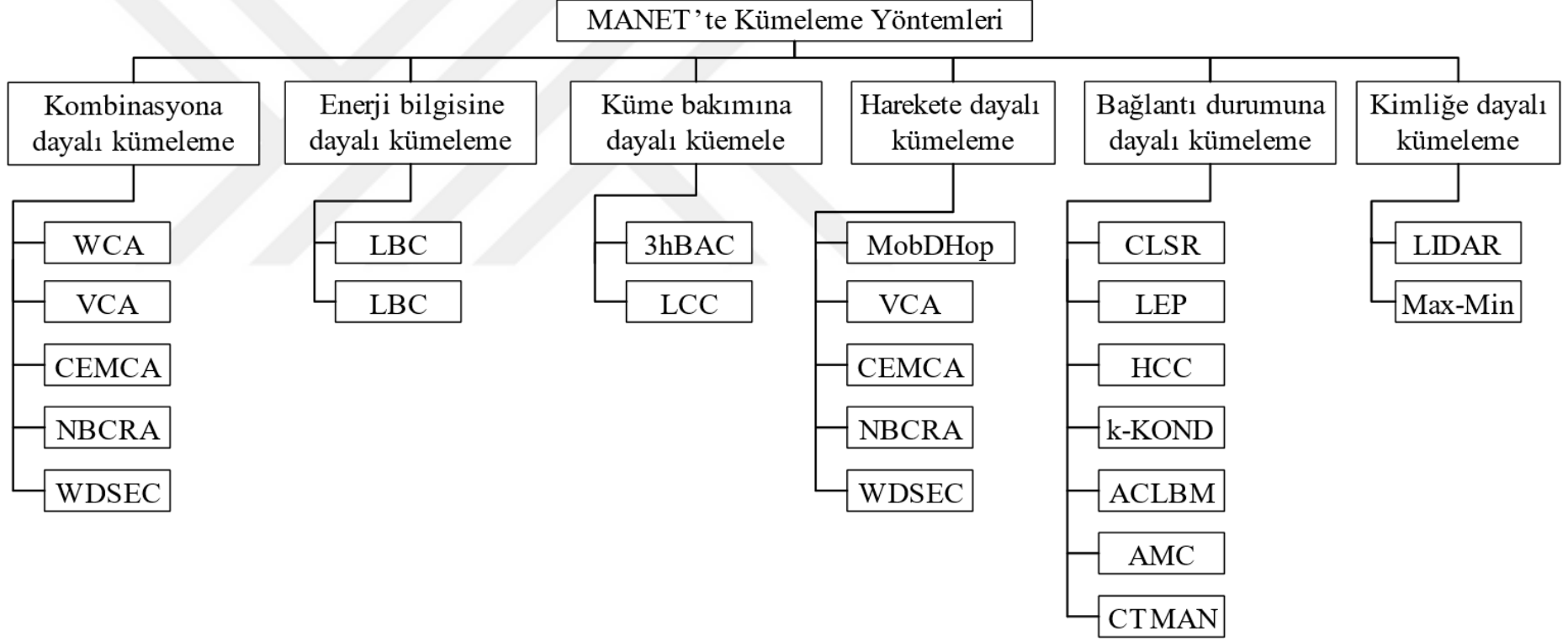
Kümeleme yapısı Őu avantajları saĐlamaktadır:

- Dinamik ve büyük ölçekli aĐların daha verimli yönetilmesini saĐlamaktadır.
- DüĐümlerin enerji tüketimini azaltarak aĐın ömrünü uzatmaktadır.
- AĐdaki yönlendirme ve kontrol ek yüklerini minimize ederek, iletiŐim maliyetlerini düşürmektedir.
- Dinamik aĐlarda hızlı baĐlantı, daha iyi yönlendirme ve topoloji yönetimi saĐlamaktadır.

Dezavantajları ise:

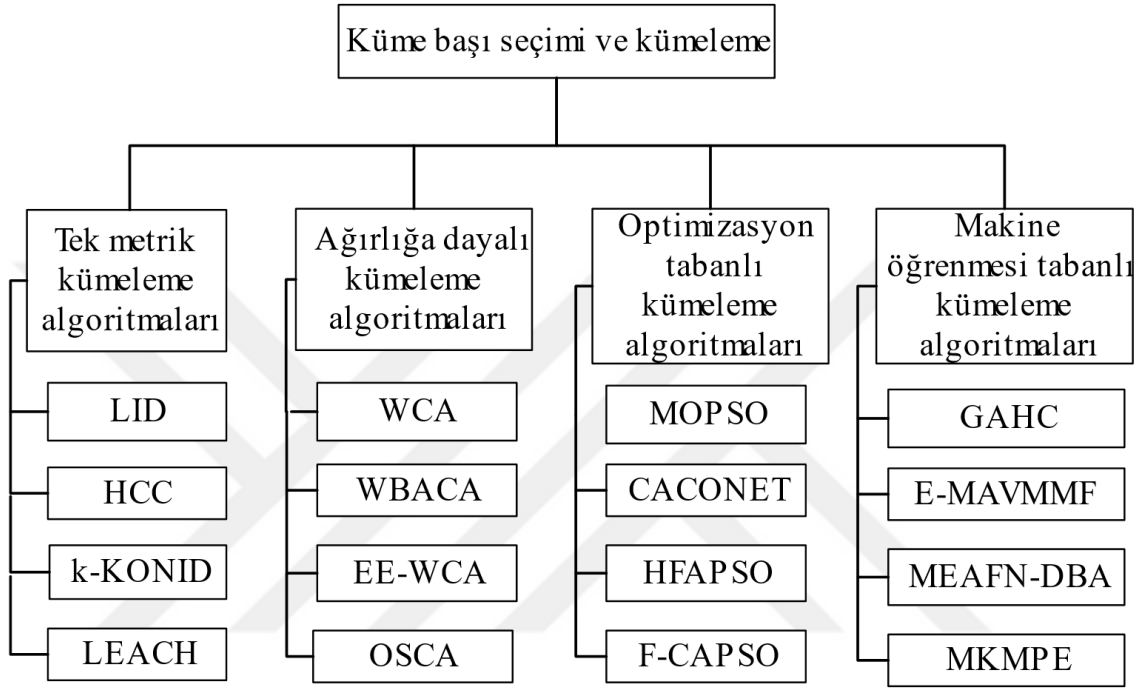
- Kümelemeyle ilgili veri deĐiŐimi, lokal aĐlarda ek yükü artırmaktadır.
- AĐ yapısının deĐiŐmesi durumunda küme yapısının yeniden yapılandırılması hesaplama maliyetini artırmaktadır.
- Kontrol mesajı alıŐveriŐi nedeniyle iletiŐim karmaŐıklıĐı artmaktadır [87].

Literatürde kablosuz aĐlar için çeŐitli kümeleme yöntemleri önerilmektedir. Bu yöntemlerin bazıları kararlı kümeler oluŐturarak yönlendirme ve kontrol ek yüklerini azaltmaya odaklanırken, bazıları enerji tüketimini düşürerek aĐ ömrünü uzatmayı hedeflemektedir. Ayrıca literatürde, yaygın olarak kullanılan kümeleme algoritmaları Őekil 2.19'te gösterildiĐi gibi farklı yöntemlere göre çeŐitli Őekillerde sınıflandırılmaktadır [6], [7].



Şekil 2.19. MANET'te kümeleme yöntemleri [7].

Bu tez çalışmasında ise, MANET'lerde kümeleme algoritmaları için temel sınıflandırma yöntemleri ve KB seçimine yönelik bir metodoloji sunulmaktadır. Literatürde kullanılan mevcut yaklaşımlara dayanarak, bu tezde önerilen metodoloji, tek metrik kümeleme, ağırlık tabanlı kümeleme, optimizasyon tabanlı kümeleme ve makine öğrenmesi tabanlı kümeleme olmak üzere dört ana sınıfa ayrılmaktadır. Şekil 2.20'de, bu sınıfları ve bu sınıflar altında bazı kümeleme algoritmaları gösterilmektedir.



Şekil 2.20. MANET'te kümeleme algoritmalarının sınıflandırılması.

2.2.1. Tek Metrik Kümeleme Algoritmaları

Bu sınıftaki algoritmalar, KB seçimini yalnızca tek bir parametre kullanarak gerçekleştirmektedir. Bu parametreler, ağın genel durumu ve ihtiyaçlarına bağlı olarak değişiklik göstermektedir. Tipik olarak kullanılan parametreler şunlardır:

- **Düğüm kimliği:** Düğüm kimlik numarasına dayalı seçim, genellikle daha basit ve hızlı bir yöntem sağlar. En düşük ID'ye sahip düğümün KB olarak seçilmesi gibi yaklaşımlar kullanılmaktadır.
- **Düğüm enerjisi:** Ağın genel enerji verimliliğini artırmayı hedefleyen bu metrik KB seçimi sırasında düğümün kalan enerjisini dikkate alır. Daha yüksek enerjiye sahip düğümler önceliklidir.
- **Düğüm hareketliliği:** Düğümün hareket hızına dayalı olarak, daha az hareket eden

düğümleler KB olarak seçilir. Bu yaklaşım, küme stabilitesini artırmayı hedeflemektedir.

- **Düğüm derecesi:** Düğümün komşu düğümlerle olan bağlantı sayısına (derecesine) bağlı olarak, daha yüksek dereceli düğümler KB olarak seçilmektedir. Bu yöntem, ağın bağlanabilirliğini artırmayı amaçlamaktadır.
- **Düğüm iletişim gücü:** Düğümün iletişim gücünü dikkate alan bu yaklaşım, daha yüksek iletişim gücüne sahip düğümleri KB olarak seçerek daha geniş bir kapsama alanı sağlamayı hedeflemektedir.

Bu metrikler, ağın yoğunluğu, topolojisi ve dinamik gereksinimlerine göre belirlenmektedir. Tek metrikli kümeleme algoritmaları, genellikle hesaplama açısından daha basit ve hızlıdır. Ancak, yalnızca tek bir parametreye dayandıkları için ağ performansını olumsuz etkileyebilecek sınırlamalara sahiptirler.

Bu başlık altında, literatürde yaygın olarak kullanılan dört farklı tek metrikli kümeleme algoritması ele alınmaktadır. Bu algoritmalar; En Düşük Kimlik Kümeleme Algoritması (LID), En Yüksek Bağlantılılık Kümeleme Algoritması (HCC), k-Komşuluk Kümeleme Algoritması (k-KONID) ve Düşük Enerjili Uyarlanabilir Kümeleme Hiyerarşisi (LEACH) algoritmalarıdır. Bu algoritmaların temel prensipleri ve uygulama yaklaşımları aşağıdaki alt başlıklarda ayrıntılı olarak açıklanmıştır.

2.2.1.1. LID

MANET'lerde kararlı ve verimli bir kümeleme sağlamak amacıyla M. Gerla ve J.T. Tsai tarafından geliştirilmiştir. Bu algoritmada her düğüme rastgele benzersiz bir kimlik atanır ve her düğüm, kendisi dahil olmak üzere 1-hop mesafedeki komşularının kimliklerini düzenli aralıklarla yayınlar. Algoritma, en düşük ID'ye sahip düğümü KB olarak seçer ve bu seçim kriteri, yalnızca düğüm kimliklerine dayanır. Düğüm kimliklerinin değeri zamana göre değişmediğinden, belirli düğümler daha uzun süre KB olarak hizmet verebilir. LID algoritmasında, bir düğüm birden fazla KB'nin iletim aralığında yer alıyorsa (yani kendisine KB olarak adlandırılan en az iki düğümden paket alıyorsa), bu düğüm ağ geçidi olarak adlandırılır ve kümeler arası yönlendirme işlemleri için kullanılır [10]. LID algoritması, basit bir metrik kullanarak kararlı kümeleme sağlayarak ağın ölçeklenebilirliğini artırmaktadır. Ancak, sadece kimlik değerine dayalı seçim, ağın yük dengesini olumsuz etkilemektedir. Dolayısıyla düşük kimlikli düğümler genellikle daha fazla enerji tüketerek ağın ömrünü azaltmaktadır [11].

2.2.1.2. HCC

M. Gerla ve C.R. Lin'in geliřtirdikleri bu yaklařım, MANET'te KB seiminde dğüm baėlantı sayısını temel almaktadır. Bu algoritmada, her dğüm kendi ID'sinin ve komřu dğümlerinin ID'lerini belirli aralıklarla yayımlar, böylece her dğüm komřularını tanır. Bir dğümün derecesi, o dğümün baėlı olduėu komřu sayısını ifade etmektedir. En yüksek dereceye sahip dğüm, KB olarak seilir. Eėer aynı menzilde iki veya daha fazla dğüm aynı dereceye sahipse, aralarından en düşük ID'ye sahip olan dğüm KB olarak atanır. Bu yöntem, aėın topolojik yapısını ve baėlantı yoğunluėunu dikkate alarak daha dengeli bir kümeleme saėlamayı amaçlamaktadır [88].

HCC algoritması, KB sayısını azaltarak paket teslim gecikmesini düşürme avantajına sahiptir. Ancak, dğümlerin hareketliliėine baėlı olarak KB deėiřimi sık yařanabilmekte, bu da aėın kararlılıėını olumsuz etkilemektedir. Özellikle yüksek hareketlilik durumlarında, KB deėiřimlerinin artması, yeniden baėlanma süreçlerini ve dolayısıyla aė üzerindeki kontrol mesajlarının sayısını artırabilir. Bu durum, aėın genel performansını ve enerji verimliliėini düşürebilmektedir [11], [13].

2.2.1.3. k-KONID

Chen ve arkadaşları tarafından geliřtirilen k-KONID algoritması, LID ve HCC algoritmalarının bir kombinasyonudur [89]. HCC algoritmasında olduėu gibi tek bir kriter olarak dğüm baėlantı sayısının kullanılması, dğümler arasında çok sayıda baė oluşmasına yol açarken, LID'de olduėu gibi yalnızca en düşük ID'nin kriter olarak seilmesi ise daha fazla küme oluşmasına neden olmaktadır. Bu algoritma, bu iki yöntemin eksikliklerini gidermek amacıyla tasarlanmıřtır. k-KONID algoritmasının temel amacı, aėda oluşan küme sayısını ve her kümedeki dğüm sayısını minimize ederek, daha küçük ve etkili baskın kümeler oluřturmaaktır [13].

Algoritmada her dğüme bir did = (d, ID) çifti atanır. Burada:

- d, dğümün baėlantı sayısını ifade eder.
- ID, dğümün kimliėini belirtir.

Algoritma işleyiři:

- KB seimi: En yüksek baėlantı sayısına sahip dğüm KB olarak seilir. Eėer baėlantı sayısı eřitse, daha düşük ID'ye sahip dğüm öncelikli olarak KB olur.
- Kümelerin oluşumu: Bir KB, kendi k-hop mesafesindeki komřu dğümlere bilgi

göndererek kümesini oluşturur.

- Uzantı: Bu yaklaşımda kümeler, bir KB'den ve KB'ye en fazla k-hop uzaklıktaki düğümlerden oluşmaktadır.

HCC ile İlişkisi:

$k = 1$ olduğunda, k-KONID algoritması HCC algoritmasının bir uzantısı olarak çalışır. Ancak $k > 1$ olduğunda, k-hop uzaklıktaki düğümleri de kapsayan daha geniş kümeler oluşturmaktadır [7].

2.2.1.4. LEACH

LEACH algoritması, sensör ağlarındaki enerji tüketimini optimize etmek ve ağın ömrünü uzatmak için Heinzelman ve arkadaşları tarafından geliştirilmiş bir kümeleme algoritmasıdır. Bu algoritma, sensör düğümlerini kümelemek ve enerji tasarrufu sağlamak amacıyla, KB olarak görev yapan düğümleri dinamik bir şekilde seçer. KB'ler, ÜD'den gelen verileri toplayarak birleştirir ve sıkıştırır. Ardından bu verileri doğrudan ya da çok atlamalı yollarla Baz İstasyonu'na (BS) iletmektedir.

Bu süreçte, KB seçimi rastgele olarak yapılmakta ve rotasyonel bir mekanizma ile enerji tüketimi tüm düğümler arasında dengelenmektedir. Bu sayede, belirli düğümlerin enerji tüketimi yoğunlaşarak ağın erken çökmesi engellenmiş olur [14].

LEACH algoritmasının işleyişi;

- KB seçimi:
 - Her düğüm, bir KB olma kararını belirli bir olasılık ile kendi kendine verir. Bu olasılık, düğümün önceki turlarda KB olup olmadığına ve düğümün kalan enerjisine dayalıdır.
 - Rastgele sayı üreten her düğüm, bu sayının belirli bir eşik değerinden küçük olması durumunda KB olarak seçilir.
- Küme oluşumu:
 - KB olarak seçilen düğümler, KB olduklarını diğer düğümlere duyurur.
 - ÜD'ler, alınan sinyal gücüne dayanarak bir KB'ye bağlanır. Bu bağlantı, düğümlerin KB'ye olan mesafesini minimize ederek enerji tüketimini azaltmayı amaçlamaktadır.

- Veri toplama ve iletim:
 - KB, kümesindeki düğümlerden gelen verileri toplar, birleştirir ve sıkıştırır.
 - Daha sonra bu sıkıştırılmış veri, KB tarafından BS'ye iletilir. Bu iletim, genellikle daha güçlü iletişim kaynakları gerektirir ve KB'lerin enerji tüketimini artırır.
- Döngüsel KB rotasyonu:
 - Enerji tüketimini eşit şekilde dağıtmak için KB pozisyonu düzenli olarak değiştirilir. Bu, düğümlerin aşırı enerji tüketimini önleyerek, ağın genel ömrünü uzatmaktadır.
- Çok turlu işleyiş:
 - Algoritma, kurulum ve veri iletimi aşamalarından oluşan çok turlu bir yapıda çalışır. Her tur, yukarıdaki süreçlerin tekrarını içermektedir [14], [90].

2.2.2. Ağırlığa Dayalı Kümeleme Algoritmaları

Ağırlığa dayalı kümeleme algoritmaları, KB seçimi ve kümeleme sürecinde birden fazla parametreyi dikkate alarak ağ performansını optimize etmeyi hedefleyen algoritmalar. Bu algoritmalar, düğümlerin kimliği, enerji seviyesi, hareketliliği, düğüm derecesi (komşu sayısı), iletim gücü gibi birden fazla metriğe belirli ağırlıklar atayarak, karar verme sürecinde bu parametrelerin önemini dikkate alır. Literatürde, ağırlığa dayalı yaklaşımlar, düğüm hareketliliğini azaltma, ağ ömrünü uzatma ve ağın genel kararlılığını artırma gibi avantajlar sağlamaktadır [7].

Bu tür algoritmalar, özellikle dinamik ve yoğun ağlarda ölçeklenebilirliği ve verimliliği artırmada etkili bir çözüm sunmaktadır. Ancak, birden fazla parametreyi hesaba katmak, bu algoritmaların karmaşıklığını artırmakta ve karar verme sürecinde ek işlem yüküne yol açabilmektedir. Ağırlıkların doğru şekilde belirlenmesi, algoritmanın başarısını doğrudan etkileyen kritik bir faktördür.

Bu bölümde, literatürde sıkça karşılaşılan ağırlığa dayalı kümeleme algoritmalarından olan Ağırlıklı Kümeleme Algoritması (WCA), Ağırlık Tabanlı Adaptif Kümeleme Algoritması (WBACA), Enerji Verimli Ağırlıklı Kümeleme Algoritması (EE-WCA) ve Optimize Edilmiş Kararlı Kümeleme Algoritması (OSCA) ele alınacaktır.

2.2.2.1. WCA

WCA, MANET’te dinamik topolojilerin yönetimi için tasarlanmış, dağıtık ve isteğe bağlı bir kümeleme algoritmasıdır. Bu algoritma, KB seçiminde düğümlerin ideal derece, iletim gücü, hareketlilik ve pil gücü parametrelerini dikkate alarak ağın kararlılığını ve performansını artırmayı hedeflemektedir.

Buna göre WCA’nın temel özellikleri;

- Çoklu parametre değerlendirmesi: WCA, her düğüm için bir ağırlık değeri hesaplar ve bu hesaplamada aşağıdaki faktörleri göz önünde bulundurmaktadır.
 - Derece farkı (Δ_v): düğümlerin sahip oldukları komşu sayıları için hesaplanan derece farkıdır ve $\Delta_v = |d_v - \delta|$ şeklinde gösterilmektedir. Burada d_v mobil düğümün komşu sayısı ve δ ise KB’nin önceden tanımlanmış ideal olarak işleyebileceği düğüm sayısını göstermektedir ve bir eşik değeri olarak kabul edilmektedir.
 - Düğümün mesafeler toplamı (D_v): Düğümden tüm komşularına olan mesafelerin toplamıdır.
 - Hareketlilik (M_v): Düğümün hareketlilik hızı, yani zaman içindeki konum değişikliğine bağlı mobilite seviyesidir.
 - Pil gücü (P_v): Düğümün harcanan enerji seviyesidir.

Bu parametrelerin birleşimi, her düğüm için bir ağırlık değeri oluşturur ve en düşük ağırlık değerine sahip düğümler KB olarak seçilmektedir.

- KB seçimi: WCA, KB seçim sürecini periyodik olarak değildir, ağın ihtiyaçlarına göre başlatır. Her bir düğümün ağırlığı hesaplandıktan sonra iletim aralığındaki düğümler arasında en küçük ağırlığa sahip düğüm KB olarak seçilmektedir.
- İsteğe bağlı KB seçimi: WCA, KB seçim sürecini periyodik olarak değildir, ağın ihtiyaçlarına göre başlatır. Bu sayede hesaplama ve iletişim maliyetleri azaltılmaktadır [7], [16].

2.2.2.2. WBACA

WBACA, MANET’ler için tasarlanmış bir kümeleme algoritmasıdır. Ağın kararlılığını artırmak ve KB seçim sürecini optimize etmek amacıyla düğümlerin iletim gücü, iletim hızı, hareketlilik, pil gücü ve düğüm derecesi gibi çeşitli parametreleri dikkate almaktadır.

Buna göre WBACA'nın temel özellikleri;

- Çoklu parametre değerlendirmesi: WBACA, her düğüm için bir ağırlık değeri hesaplar ve bu hesaplamada aşağıdaki faktörleri göz önünde bulundurur:
 - İletim gücü (T_x): Düğümün komşularına veri iletmek için gereken güç miktarıdır.
 - İletim oranı (T_r): Düğümün veri iletim kapasitesidir.
 - Hareketlilik (M): Düğümün hareketlilik hızı, yani zaman içindeki konum değişikliği.
 - Pil Gücü (B): Düğümün mevcut enerji seviyesidir.
 - Derece farkı (D): Bir düğümün komşu sayısı, yani doğrudan iletişim kurabileceği düğüm sayısıdır [17].

Bu parametrelerin birleşimi, her düğüm için bir ağırlık değeri oluşturur ve en düşük ağırlık değerine sahip düğümler KB olarak seçilmektedir.

- Kararlılık ve uyarlanabilirlik: WBACA, ağın dinamik doğasına uyum sağlayarak KB seçim sürecini optimize ederek, gereksiz yeniden yapılandırmaları önlemektedir.
- Yük dengesi: Algoritma, her bir KB'nin yönetebileceği düğüm sayısını sınırlamaktadır. Bu durum, KB'lerin aşırı yüklenmesini önler ve ağ performansını optimize eder.

WBACA algoritmasının işleyişi:

- Ağırlık hesaplama: Her düğüm, komşularından aldığı parametreleri kullanarak ağırlığını hesaplar. Ağırlık formülü Denklem (2.1)' deki gibidir.

$$W_N = w_1 * M + w_2 * B + w_3 * T_x + w_4 * D + w_5 / T_r \quad (2.1)$$

Burada w_1, w_2, w_3, w_4 ve w_5 ağırlık faktörleri ağın ihtiyaçlarına göre manuel ayarlanmaktadır.

- KB seçimi:
 - Her düğüm, kendisinin ve komşularının ağırlıklarını karşılaştırır.
 - En düşük ağırlığa sahip düğüm, KB olarak atanır.
 - KB seçilen düğüm, komşularına bu durumu bildirir.

- Kümelerin oluşumu:
 - KB'ler, komşularına küme lideri olduklarını duyurur.
 - Diğer düğümler, sinyal gücüne ve en düşük ağırlık değerine göre bir KB'ye bağlanır.
 - Bir düğüm, birden fazla KB'nin iletim aralığında ise, en düşük ağırlığa sahip KB'ye bağlanır.
- Süreç güncellemesi:
 - Düğümler, hareketlilik veya enerji durumlarına göre ağırlıklarını düzenli aralıklarla yeniden hesaplar.
 - Bir KB'nin enerji seviyesi kritik seviyeye düştüğünde, küme liderliği en düşük ağırlığa sahip başka bir düğüme devredilir.

KB seçimi düzenli olarak yenilenir ve bu da yük dengesini sağlamaktadır [17].

WBACA, MANET'lerde dinamik topolojilerin yönetimi için etkili bir çözüm sunmaktadır. Birden fazla parametreyi dikkate alarak daha kararlı ve dengeli bir kümeleme yapısı oluşturmakta ve uyarlanabilir KB seçimi ile ağırlık kararlılığını ve performansını artırmaktadır.

2.2.2.3. EE-WCA

Kablosuz sensör ağları'nda enerji tüketimini minimize etmek ve ağ ömrünü uzatmak amacıyla geliştirilmiş bir kümeleme algoritmasıdır. Bu algoritma, düğüm hareketliliği ve enerji tüketimi gibi kritik konuları ele alarak, topoloji kontrolü üzerinden ağırlık performansını iyileştirmeyi hedeflemektedir.

Buna göre EE-WCA'nın temel özellikleri;

- QoS gereksinimlerinin değerlendirilmesi: EE-WCA, KB seçiminde düğümlerin tampon uzunluğu, düğüm yer değiştirmesi, pil seviyesi, bağlantı durumu Ve Sinyal-Gürültü Oranı (SNR) gibi QoS parametrelerini dikkate alır. Bu sayede, uygulamaların QoS gereksinimlerini karşılayacak şekilde KB seçimi yapılmaktadır.
- Ağırlık hesaplaması ile hesaplama yükünün azaltılması: Algoritma, düğümlerin hesaplama yükünü azaltmak ve enerji tasarrufu sağlamak için ağırlık hesaplamasına dayalı bir teknik kullanır. Bu yaklaşım, KB seçim sürecini optimize etmektedir.

- Kaynak tüketiminin dengeli dağılımı: EE-WCA, ağın tamamında kaynak tüketimini eşit olarak dağıtmak için KB güncelleme sürecini tanımlamaktadır. Bu mekanizma, belirli düğümlerin aşırı enerji tüketimini önleyerek ağın genel performansını artırmaktadır [22].

EE-WCA algoritmasının işleyişi:

- EE-WCA’da KB seçimi için kullanılan parametreler:
 - Tampon uzunluğu (FA): Tamponda kullanılan alan, düğümün iş yükünü gösterir. Denklem (2.2)’deki gibi gösterilmektedir.

$$F = B - X \quad (2.2)$$

Burada B: tampon boyutu, X ise kullanılan tampon miktarıdır.

- Enerji tüketimi (Δe): Düğümün pil tüketim durumunu yansıtır. Denklem (2.3)’teki gibi gösterilmektedir.

$$\Delta e = \frac{|e_2 - e_1|}{\Delta t} \quad (2.3)$$

Burada düğümün ilk enerjisi e_1 Δt zaman sonra kalan enerjisi e_2 dir.

- Düğüm hareketliliği (d): Düğümün hareket hızını temsil etmektedir. Düşük hareketlilik daha kararlı düğümler demektir. Formülü Denklem (2.4)’te gösterildiği gibi dir.

$$d = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2} \quad (2.4)$$

Burada iki düğüm arasındaki zamana bağlı değişiminin mesafe farkı hesaplanmaktadır.

- Düğüm derecesi (D_A): Düğümün A’nın iletim menziline bulunan komşularının toplam sayısıdır. Denklem (2.5)’deki gibi normalize edilerek ağırlık hesabına eklenir.

$$D_A = \frac{\text{Mevcut komşu sayısı}}{\text{Maksimum komşu sayısı}} \quad (2.5)$$

- SNR: iletişim kalitesini ölçmek için Denklem (2.6)'daki formül kullanılmaktadır.

$$SNR = 10 \cdot \log_{10} \frac{A_{signal}}{A_{noise}} \quad (2.6)$$

- Ağırlıkların hesaplanması: EE-WCA'da her düğümün ağırlığı Denklem (2.7)'deki gibi hesaplanır.

$$W = w_1 * F1 + w_2 * F2 + \dots F_n * w_n \quad (2.7)$$

- Burada W: düğümün toplam ağırlığı, F1, F2, ..., Fn: Belirlenen QoS parametreleri ve w_1, w_2, \dots, w_n ise bu parametrelerin ağırlıklarıdır.
- . Ağırlık parametreleri olan F1, F2, ..., Fn, ağıdaki QoS gereksinimlerini karşılamak için manuel olarak veya Denklem (2.8)'deki gibi normalize edilerek belirlenmektedir.

$$NorVal = \frac{Val - Min}{Max - Min} \quad (2.8)$$

- Ağırlık faktörleri olan w_1, w_2, \dots, w_n , QoS parametrelerinin değerlerini normalleştirmek için kullanılır Ayrıca, bu faktörler ağı kurulumundaki QoS gereksinimlerine dayanarak manuel belirlenmektedir. Burada, geleneksel olarak Denklem (2.9)'daki kısıtın takip edilmesi gerekmektedir.

$$w_1 + w_2 + \dots w_n = 1 \quad (2.9)$$

- Parametrelerin normalizasyonu Denklem (2.10)'da gösterilmektedir.

$$W = \frac{FA - Fmin}{Fmax - Fmin} - \frac{dA - dmin}{dmax - dmin} - \frac{\Delta eA - \Delta emin}{\Delta emax - \Delta emin} + \frac{DA - Dmin}{Dmax - Dmin} + \frac{SNRA - SNRmin}{SNRmax - SNRmin} \quad (2.10)$$

- Bu formüle göre pozitif işaretler yüksek tampon kapasitesi, yüksek bağlantı derecesi ve yüksek SNR için kullanılırken negatif işaretler düşük hareketlilik ve düşük enerji tüketimi için tercih edilmektedir.
- Her düğümün ağırlığı hesaplandıktan sonra, en yüksek ağırlığa sahip düğüm KB olarak seçilir. Her bölgede birden fazla KB olabilir. KB seçimi, tüm ağ genelinde

kaynakların dengeli kullanımını sağlayarak yüksek performanslı bir ağ oluşturmaktadır.

- Kümelerin oluşumu:
 - KB olarak seçilen düğümler, durumlarını diğer düğümlere duyururlar.
 - Diğer düğümler, sinyal gücü ve enerji durumu gibi parametrelere dayanarak en uygun KB'ye katılır ve küme üyeleri olarak atanmaktadırlar.
- KB güncelleme süreci:
 - Belirli bir zaman aralığı sonunda KB rotasyonu gerçekleştirilmektedir.
 - Bu mekanizma, kaynak kullanımını tüm ağ düğümleri arasında eşit olarak dağıtmayı ve düğümlerinin aşırı yüklenmesini önlemeyi amaçlamaktadır.
 - Mevcut KB, enerjisinin belirli bir eşik seviyesinin altına düştüğünü tespit ettiğinde, KB seçimi sürecini başlatır.
 - Böylece, düşük enerjiye sahip düğüm yerine daha fazla enerjiye sahip bir düğüm KB olarak atanmaktadır.
 - Eğer daha parametreleri yüksek yeni bir düğüm ağa katılırsa, mevcut KB yerine bu düğüm KB olarak seçilmektedir [22], [23].

2.2.2.4. OSCA

MANET'lerde KB değişimlerinin etkisini azaltarak ağ kararlılığını artırmayı ve kontrol yükünü en aza indirmeyi hedefleyen bir kümeleme algoritmasıdır. OSCA, dinamik topolojilere sahip ağlarda verimliliği artırmak için yedek düğüm mekanizmasını ve çoklu parametre değerlendirmesini içeren bir yöntem sunmaktadır.

Algoritmanın genel işleyişi ve adımları:

- Kümeleme ve KB seçimi:
 - Yedek düğüm tanıtımı: OSCA, her küme için bir ana KB ve bir yedek düğüm tanımlar. Yedek düğüm, KB'nin görevini devam ettiremeyeceği durumlarda otomatik olarak devreye girmektedir.
 - Öncelik faktörü hesaplaması: KB seçimi için düğüm derecesi, kalan pil enerjisi ve düğüm hareketliliği gibi parametreler kullanılarak bir öncelik faktörü hesaplanır. Bu faktör, ağ kararlılığı ve enerji verimliliği sağlamak amacıyla KB

seçiminde belirleyici rol oynamaktadır.

- Kümelerin oluşumu:
 - Bir KB, kendi iletim aralığındaki düğümlerden oluşan kümesini kurar ve düğümler arasında verimli bir iletişim sağlar.
 - Kümeler, KB'ye belirli bir uzaklıkta olan düğümlerden oluşur. Düğümler, sinyal gücü ve enerji gibi metriklere dayanarak hangi kümede yer alacaklarına karar vermektedir.
- Yedek düğüm mekanizması:
 - KB düğümü enerjisini tükettiğinde veya iletişim yeteneğini kaybettiğinde, yedek düğüm otomatik olarak KB rolünü üstlenir. Bu süreç, yeniden kümeleme ihtiyacını azaltarak ağın sürekliliğini sağlamaktadır.
- Enerji ve kaynak yönetimi:
 - Yerel veri işleme ve sıkıştırma gibi yöntemlerle iletişim sırasında enerji tasarrufu sağlar. Ayrıca, KB ve yedek düğüm arasında iş yükü dengelenmektedir [19].

2.2.3. Optimizasyon Tabanlı Kümeleme Algoritmaları

Optimizasyon tabanlı kümeleme algoritmaları, ad hoc ağlarda düğüm kümeleme süreçlerini iyileştirmek ve ağın genel performansını artırmak için çeşitli meta-sezgisel ve matematiksel optimizasyon tekniklerini kullanmaktadır. Bu algoritmalar, enerji verimliliği, küme kararlılığı, iletişim maliyeti ve düğüm hareketliliği gibi çeşitli metrikleri optimize ederek daha etkili kümeleme yapıları oluşturmayı hedeflemektedir. Literatürde Parçacık Sürü Optimizasyonu (PSO), Arı Kolonisi Optimizasyonu (ACO) gibi yöntemler sıklıkla kullanılmaktadır. Optimizasyon tabanlı yöntemler, geleneksel kümeleme algoritmalarının aksine, birden fazla metriği aynı anda dikkate alarak KB seçimi ve küme oluşumu süreçlerini geliştirmektedir. Bu sayede, ağın enerji tüketimini azaltmak, yaşam süresini uzatmak ve yük dengelemesini sağlamak gibi avantajlar sunarlar. Ancak, bu algoritmalar genellikle yüksek hesaplama maliyeti ve uygulama karmaşıklığı ile sınırlı olabilmektedir [18], [91].

Bu bölümde, literatürde sıkça karşılaşılan dört farklı optimizasyon tabanlı kümeleme algoritması ele alınmaktadır. Bu algoritmalar; Çok Amaçlı Parçacık Sürü Optimizasyonu (MOPSO), Karınca Kolonisi Optimizasyonuna Dayalı Kümeleme Algoritması

(CACONET), Ateşböceği Algoritması İle Parçacık Sürü Optimizasyonunun Hibrit Yaklaşımı (HFAPSO) ve Enerji Verimli Ve Güvenli Veri İletimi İçin Bulanık Kaos Uyarlanabilir Parçacık Sürü Optimizasyonu (F-CAPSO) kümeleme algoritmalarıdır. Bu algoritmaların temel prensipleri ve uygulama yaklaşımları aşağıdaki alt başlıklarda ayrıntılı olarak açıklanmıştır.

2.2.3.1. MOPSO

M.S. Lechuga ve arkadaşlarının geliştirdikleri MOPSO, PSO'nun çok amaçlı bir uzantısıdır [92]. Bu algoritma, MANET'lerdeki dinamik topolojilerin yönetimi ve enerji verimliliğinin artırılması gibi kritik problemlere çözüm sunmaktadır. Parçacıklar, çözüm uzayında bireysel ve küresel en iyi konumlarına göre hareket ederek, belirli bir problem için optimum çözümleri arar. MOPSO, kümeleme problemlerinde, düğüm kümeleme süreçlerini optimize etmek için kullanılır. Bu algoritma, enerji verimliliği, bağlantı kalitesi ve iletişim maliyetleri gibi birden fazla metrik üzerinde eş zamanlı optimizasyon yapabilmektedir [25].

MOPSO algoritmasının işleyişi:

- Başlangıç ayarları:
 - Algoritma, her bir düğüm için bir parçacık oluşturur.
 - Parçacıklar, çözüm uzayındaki konum ve hız vektörleri ile temsil edilmektedir.
 - Çok amaçlı optimizasyon için uygun bir uygunluk (fitness) fonksiyonu tanımlanır. Bu fonksiyon genellikle enerji verimliliği, küme stabilitesi ve bağlantı derecesi gibi metrikleri içerir.
- Parçacıkların konum ve hız güncellemesi:
 - Parçacıklar, bireysel en iyi konumlarına (pBest) ve küresel en iyi konuma (gBest) göre hareket eder.
 - Güncellenmiş hız Denklem (2.11)'de ve konum hesaplamaları Denklem(2.12)'de gösterilmektedir.

$$V_i^{t+1} = wV_i^t + c_1 * r_1(pBest_i - X_i) + c_2 * r_2(gBest_i - X_i) \quad (2.11)$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \quad (2.12)$$

Burada: V_i : parçacığın hızını, X_i : parçacığın konumunu, c_1, c_2 : öğrenme katsayılarını, r_1, r_2 : rastgele sayıları ve w : atalet ağırlığını göstermektedir.

- Kümeleme süreci:
 - Her parçacık, uygunluk fonksiyonu tarafından değerlendirilen bir çözüm önerir.
 - En iyi çözüm, KB'lerin optimum dağılımını belirlemektedir.
- Optimizasyon süreci:
 - Parçacıklar, belirlenen metriklere göre iterasyonlar boyunca konumlarını günceller.
 - Optimizasyon süreci, ağın enerji tüketimini minimize ederken kümeleme yapısını stabilize etmeyi hedeflemektedir.
- Durdurma kriteri:
 - Algoritma, önceden belirlenen bir iterasyon sayısına ulaştığında veya uygunluk fonksiyonu belirli bir eşik değerine ulaştığında sona ermektedir [25].

2.2.3.2. CACONET

CACONET, ACO'nun meta-sezgisel doğasını kullanarak dinamik ve değişken bir ağ yapısında KB'lerin seçimini optimize etmeyi hedeflemektedir. ACO, karıncaların doğal yollarını bulma davranışından esinlenerek geliştirilmiş bir optimizasyon yöntemidir. Algoritmada, feromon ve sezgisel değer gibi temel kavramlar yer alır. Feromon, karıncaların yollarına bıraktığı kimyasal izlerden esinlenerek modellenmiş bir değerdir ve düğümler veya kenarlar üzerindeki çözüm kalitesini temsil etmektedir. Daha iyi çözümler, daha fazla feromon birikimi sağlar ve bu yolların daha sık tercih edilmesine yol açar. Sezgisel değer (η), karar verme sürecinde kullanılan bir ek bilgidir ve genellikle düğümler arasındaki mesafe veya bağlantı kalitesi gibi ölçütlere dayanmaktadır. CACONET, özellikle Araç Tabanlı Ad-Hoc Ağlardaki (VANET) düğümler arası iletişim maliyetlerini azaltmayı ve kümeleme kararlılığını artırmayı amaçlamaktadır [26].

CACONET algoritmasının işleyişi:

- Başlangıç ayarları:
 - Araçların pozisyonları ve yönleri rastgele olarak belirlenir.
 - Araçların hızları rastgele atanır.

- VANET ortamındaki tüm araçlar bir mesh topolojisiyle temsil edilir ve her araç bir düğüm olarak adlandırılır.
- Düğümler arası bağlantılara başlangıç feromon değerleri atanmaktadır.
- Arama alanı ve feromon değerlerinin tanımlanması:
 - Düğümler arası mesafeler hesaplanır ve bu mesafeler normalize edilerek ağdaki kenarlara atanır.
 - Feromon değerleri, düğümler arası bağlantıların tercih edilme olasılığını artırmak için başlangıç değerlerine göre güncellenmektedir.
- Çözüm değerlendirme:
 - Her bir yapay karınca, bir başlangıç düğümünden rastgele seçilerek hareketine başlamaktadır.
 - Her adımda, bir sonraki düğüm, feromon ve sezgisel değerlere bağlı olarak olasılıkla seçilmektedir.
 - Seçim süreci Denklem (2.13)' te gösterildiği gibidir.

$$P_{i,j} = \frac{\text{Feromon}_{i,j} * \text{Sezgisel değer}_{i,j}}{\sum_{k \in S} \text{Feromon}_{i,j} * \text{Sezgisel değer}_{i,j}} \quad (2.13)$$

Burada i , mevcut düğümü, j ise bir sonraki aday düğümü temsil etmektedir. S , seçim için uygun düğüm setidir.

- Kısıtlar:
 - Her bir turda, bir düğüm yalnızca bir kez seçilir.
 - Aynı küme içindeki düğümler yalnızca tek bir KB'ye bağlı olabilmektedir.
- Çözüm değerlendirme:
 - Her karınca tarafından oluşturulan tur, iki çoklu hedef fonksiyonuna göre değerlendirilir.
 - Bu fonksiyonlardan $f1$: KB'ye düşen düğüm sayısı ile kullanıcı tarafından belirlenen ideal küme boyutu arasındaki farkı minimize eder.
 - Fonksiyon $f2$: KB'ler ile küme üyeleri arasındaki toplam mesafeyi minimize eder.

- Feromon güncellemesi:
 - Her tur sonunda, daha yüksek kaliteli çözümler üzerinde daha fazla feromon birikmesi sağlanır.
 - Feromon buharlaşması uygulanarak eski izler silinmektedir. Feromon buharlaşma formülü Denklem (2.14)'teki gibidir.

$$T_{ik}(t + 1) = (1 - p) * T_{ik}(t) + \frac{1}{1 + f_n} \quad (2.14)$$

Burada p , feromon buharlaşma oranıdır ve f_n , karınca n tarafından oluşturulan turun değerlendirme skorudur.

- Döngü kriterleri:
 - Algoritma, belirli bir iterasyon sayısına ulaşıldığında veya art arda 20 iterasyon boyunca en iyi çözümde iyileşme olmadığında sonlandırılmaktadır.
 - En iyi tur, kümeleme süreci için nihai çözüm olarak kullanılmaktadır.
- Sonuçlar ve kümeleme:
 - Algoritma, ağdaki düğümleri en uygun sayıda KB ve küme üyesi ile gruplar.
 - Sonuçlar, daha az sayıda küme oluşturularak yönlendirme maliyetlerini azaltmakta ve veri iletimi verimliliğini artırmaktadır [93].

2.2.3.3. HFAPSO

Bu algoritma, Kablosuz Sensör Ağları'nda (WSN) enerji verimliliğini artırmak ve ağ ömrünü uzatmak amacıyla geliştirilmiş bir kümeleme algoritmasıdır. HFAPSO, Ateşböceği Algoritması (FA) ve PSO tekniklerinin birleşimini kullanarak, KB seçimini optimize etmektedir. Bu hibrit yaklaşım, FA'nın küresel arama yeteneğini PSO'nun yerel arama kapasitesiyle birleştirerek, düğüm enerjisi ve KB'ye olan mesafeye dayalı olarak en uygun KB'yi belirlemektedir. HFAPSO, özellikle enerji tüketimini minimize etmek ve ağdaki canlı düğüm sayısını artırmak için tasarlanmıştır.

HFAPSO algoritmasının temel bileşenleri şunlardır:

FA: Ateşböceklerinin ışık yayma davranışından ilham almaktadır. Daha parlak ateşböcekleri, daha az parlak olanları çeker ve bu sayede çözüme yönelik bir optimizasyon sağlanmaktadır.

PSO: Kuş sürüsü ve balık sürüsü hareketlerinden esinlenerek, parçacıkların en iyi konumu bulmalarına dayanan bir optimizasyon yöntemidir.

HFAPSO algoritmasının işleyişi:

- Başlangıç ayarları:
 - Sensör düğümleri rastgele bir coğrafi bölgeye yerleştirilir.
 - LEACH-C algoritması kullanılarak kümeler oluşturulur.
- İlk KB seçimi:
 - Kümeler oluşturulduktan sonra, kalan enerjisi en yüksek olan düğümler KB olarak belirlenir.
- HFAPSO algoritmasının uygulanması:
 - KB adayları, belirlenen bir popülasyon boyutuna göre atanır ve başlangıç parametreleri olan atalet ağırlığı, bilişsel ve sosyal faktörler belirlenir.
 - Aday düğümler enerji seviyeleri ve mesafeleri baz alınarak uygunluk fonksiyonuna tabi tutulur.
- FA kullanımı:
 - Ateşböceklerinin çekiciliği ve parlaklığı belirli kurallara göre hesaplanmaktadır.
 - Ateşböceklerinin parlaklık yoğunluğu, hedef fonksiyonunun değerine bağlıdır. Parlaklık ve çekicilik formülü Denklem (2.15)'de gösterilmektedir.

$$I(r) = I_0 e^{-\gamma r^2} \quad (2.15)$$

Burada: $I(r)$ mesafeye bağlı parlaklık seviyesi, I_0 başlangıç parlaklık değeri, γ ortamın ışık absorpsiyon katsayısı ve r iki ateşböceği arasındaki mesafedir.

- Hareket denklemi: Her ateşböceğinin hareketi, çözümün doğruluğunu sağlar ve yakınsama hızını artırmaktadır. Matematiksel gösterimi Denklem (2.16)'daki gibidir.

$$x_i^{t+1} = x_i^t + \beta e^{-r_{ij}} (x_j^t - x_i^t) + \alpha \cdot (rand - 0.5) \quad (2.16)$$

Burada: x_i^{t+1} güncellenmiş konum, x_i^t mevcut konum, β çekicilik katsayısı, α rastgelelik faktörü, r_{ij} ateşböcekleri arasındaki mesafe dir.

- PSO kullanımı:

- PSO, ateşböceği algoritmasının küresel arama yeteneğini iyileştirmek için kullanılmaktadır.
- Parçacıkların hız ve konum güncellemeleri Denklem (2.17) ve Denklem (2.18)'deki gibi hesaplanmaktadır.

$$v_i^{k+1} = w \cdot v_i^k + c_1 \cdot r_1 (p_{best} - x_i^k) + c_2 \cdot r_2 (g_{best} - x_i^k) \quad (2.17)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (2.18)$$

Burada: w eylemsizlik ağırlığı, c_1 ve c_2 bilişsel ve sosyal öğrenme faktörleri, r_1 ve r_2 rastgele sayılar, p_{best} en iyi kişisel konum, g_{best} ise küresel konumdur.

- KB seçimi:

- KB seçimi için uygunluk fonksiyonu Denklem (2.19)'daki gibi hesaplanmaktadır.

$$F = f_1 * a + f_2 * (1 + a) \quad (2.19)$$

Burada f_1 düğümlerin enerji seviyelerine dayalı faktör, f_2 ise düğümlerin KB'lere olan mesafelerinin ortalamasıdır.

- Veri iletimi:

- Seçilen KB'ler verileri toplar ve ana istasyona iletmektedirler.

- Enerji güncellemeleri ve KB'nin güncellenmesi:

- Her zaman aralığında kalan enerji ve canlı düğüm sayısı hesaplanmaktadır.
- Eğer bir KB'nin enerjisi kritik seviyeye düşerse, yeni bir KB seçimi gerçekleştirilmektedir [28].

2.2.3.4. F-CAPSO

F-CAPSO algoritması, MANET'ler için enerji verimli ve güvenli veri iletimi sağlamak amacıyla geliştirilmiştir. Algoritma, bulanık mantık tabanlı optimizasyon yöntemlerini kullanarak KB seçim sürecini optimize etmektedir. Bununla birlikte, Salp Sürüsü Algoritması (SSA) ve Sinüs Kosinüs Algoritması (SKA) ile birlikte çalışarak ağ ömrünü uzatmayı ve veri iletiminde güvenliği artırmayı hedeflemektedir. F-CAPSO algoritması, güvenilir bir yönlendirme sağlamak için güven temelli çift dalga şifreleme stratejisini

kullanmaktadır.

F-CAPSO algoritmasının işleyişi:

- Bulanık mantık kullanımı:
 - Algoritma, düğüm yoğunluğu, kalan enerji ve rota uzunluğu gibi giriş parametrelerini kullanarak bulanık mantık çıkarım motoru aracılığıyla KB'leri belirlemektedir.
 - Giriş parametreleri üç dilsel değişken (Düşük, Orta, Yüksek) ile değerlendirilir.
 - Sonuç olarak, düğümlere bir rekabet yarıçapı ve öncelik sıralaması atanır.
 - LEACH-C algoritması kullanılarak kümeler oluşturulur
- Kaotik optimizasyon:
 - Kaos tabanlı optimizasyon KB seçiminde lokal minimumlardan kaçınmak için kullanılır.
 - Lojistik haritalama yöntemiyle düğümlerin başlangıç konumları belirlenmektedir.
- PSO kullanımı:
 - PSO algoritması, KB seçiminde enerji ve bağlantı gibi ölçütleri optimize etmek için uygulanır.
 - Hız ve konum güncellemeleri de Denklem (2.17) ve Denklem (2.18)'deki gibi hesaplanmaktadır.
- SSA kullanımı:
 - SSA, lider ve takipçiler olarak iki bölüme ayrılan parçacıklardan oluşmaktadır. Liderin güncellenmiş pozisyonu Denklem (2.20)'ye göre hesaplanmaktadır.
$$X_1^n = F_n + c_1 \cdot ((\gamma b_n - l_{bn}) \cdot c_2 + l_{bn}) \cdot c_3 \quad (2.20)$$
 - Buradaki X_1^n güncellenmiş salp konumu, F_n en iyi bilinen konum, c_1 , c_2 ve c_3 ise keşif parametreleridir. Ayrıca b_n üst sınır değeri, l_{bn} ise alt sınır değeridir.
- SKA kullanımı:
 - SKA algoritması, yerel arama ve keşif süreçlerini dengelemek için Denklem (2.21) ve Denklem (2.22)'deki sinüs ve kosinüs fonksiyonlarını kullanmaktadır.

$$x_{i+1}^{m,n} = x_i^{m,n} + \eta * \sin(r()) * |r()x_{best}^n - x_i^{m,n}| \quad (2.21)$$

$$x_{i+1}^{m,n} = x_i^{m,n} + \eta * \cos(r()) * |r()x_{best}^n - x_i^{m,n}| \quad (2.22)$$

Burada $X_1^{m,n}$ mevcut pozisyon, X_{best}^n en iyi bilinen çözüm, $r()$ 0- 1 aralığında rastgele bir sayı ve η adaptif keşif faktörüdür.

- Kümeleme ve veri iletimi:
 - Seçilen KB'ler, küme üyeleri ile iletişime geçerek veri iletim süreçlerini yönetmektedir.
 - Düğümler, seçilen KB'ye bağlı olarak Zaman Bölmeli Çoklu Erişim (TDMA) çerçevesinde veri iletmektedir.
- Çift dalga şifreleme:
 - Güvenli veri iletimi için çift aşamalı bir şifreleme yöntemi kullanılarak ağın güvenliği artırılmaktadır.
 - Bu süreç, ağın ömrünü uzatmayı, güvenli veri iletimini sağlamayı ve enerji tüketimini optimize etmeyi hedeflemektedir [31].

2.2.4. Makine Öğrenmesi Tabanlı Kümeleme Algoritmaları

Makine öğrenmesi tabanlı kümeleme algoritmaları, verinin içsel yapısını keşfetmek ve düğümleri belirli özelliklerine göre gruplamak için kullanılan güçlü tekniklerdir. Bu algoritmalar, denetimsiz öğrenme prensiplerine dayanarak veri noktaları arasındaki benzerlikleri analiz eder ve belirli ölçütlere göre kümeleme işlemi gerçekleştirmektedir. Geleneksel kümeleme yaklaşımlarına kıyasla, makine öğrenmesi tabanlı yöntemler büyük veri kümelerinde daha dinamik ve adaptif çözümler sunabilmektedir.

MANET'lerde makine öğrenmesi algoritmalarının kullanımı, ağın dinamik doğası ve değişken topolojisi nedeniyle giderek artmaktadır. Mobil düğümlerin rastgele hareket etmesi ve sınırlı enerji kaynakları gibi kısıtlamalar, makine öğrenmesi teknikleri sayesinde daha verimli bir şekilde yönetilebilmektedir. Makine öğrenmesi algoritmaları, MANET'lerde KB seçimi, yönlendirme, saldırı tespit algoritmaları gibi alanlarda başarıyla uygulanmaktadır. Literatürde MANET'lerde makine öğrenmesi tabanlı kümeleme algoritmalarının etkinliği üzerine birçok çalışma yapılmıştır. Örneğin, Yapay Sinir Ağları (ANN), Destek Vektör Makineleri (SVM), bulanık mantık sistemleri ve derin

öğrenme yöntemleri gibi çeşitli teknikler, KB seçimi, enerji verimliliği, küme stabilitesi, saldırı tespit yöntemleri gibi ağın performansını ve güvenliğini artırmak amacıyla uygulanmıştır [29], [94], [95], [96].

Bu bölümde, literatürde sıkça karşılaşılan makine öğrenmesi yaklaşımlarından olan Tepe Tırmanma Yöntemli Genetik Algoritma (GAHC), Enerji Verimli – Değiştirilmiş Afrika Akbabası Ve Değiştirilmiş Mayıs Sineği Algoritması (E-MAVMMF), uçan ağlar için Yoğunluk Tabanlı Karşılıklı Dışlama Algoritması (MEAFN-DBA) ve Filipin Kartalı Tabanlı Geliştirilmiş K-Ortalamalar Algoritması (MKMPE) algoritmaları ele alınmaktadır.

2.2.4.1. GAHC

GAHC algoritmasının işleyişi, ağırlıklı olarak Genetik Algoritma (GA) ve Tepe Tırmanışı (HC) tekniklerinin birleşimi ile gerçekleştirilmiştir. Bu algoritma, çok yollu güvenli yönlendirme protokolü sağlamak için geliştirilmiştir.

GAHC algoritmasının işleyişi:

- Güven yönetim sistemi doğrudan güven (DT):
 - Bir kaynaktan hedef düğüme olan bağlantı süresi baz alınarak hesaplanır. Bu süre, gerçek ve tahmini bağlantı sürelerinin farkına göre değerlendirilmektedir. Matematiksel gösterimi Denklem (2.23)'te gösterildiği gibidir.

$$DT_{di}(T) = \frac{1}{3} [DT_{di}(T - 1) - \left(\frac{T_{appx} - Test}{T_{appx}} \right) + w] \quad (2.23)$$

Burada T_{appx} tahmini süre, $Test$ gerçekleşen süre ve w ise düğümlerin görüş değişkenidir.

- Güven yönetim sistemi dolaylı güven (IDT):
 - Tanıklık değişkeni içermeyen düğümler için, doğrudan güven değerlerinin komşular üzerinden ortalaması alınarak hesaplanmaktadır. Matematiksel Formülü Denklem (2.24)'te gösterildiği gibidir.

$$IDT_{di}(T) = \frac{1}{r} \sum_{i=1}^r DT_{di}(d) \quad (2.24)$$

Burada r , düğümün komşuluk alanındaki düğüm sayısını ifade etmektedir.

- Güven yönetim sistemi güncel güven (RT):
 - Doğrudan ve dolaylı güvenin ağırlıklı ortalaması alınarak hesaplanır. Matematiksel gösterimi Denklem(2.25)'de gösterildiği gibidir.

$$RT_{di}(T) = a * DT_{di}(T) + (1 - a) * IDT_{di}(T) \quad (2.25)$$

- KB seçimi:
 - KB seçiminde İyileştirilmiş Fuzzy C-Means (FCM) algoritması kullanılmıştır. Düğümlerin doğrudan, dolaylı ve güncel güven değerleri göz önünde bulundurularak, yoğunluk zirve bazlı kümeleme yapılmaktadır. Kümeleme formülü Denklem (2.26)'daki gibidir.

$$J(U, V) = \sum_{i=1}^c \sum_{j=1}^n (u_{ij})^m (d_{ij})^2 \quad (2.26)$$

Burada (u_{ij}) küme üyelik derecesi, (d_{ij}) öklid mesafesi ve m bulanıklaştırma katsayısıdır.

- Saldırı tespiti ve eşik değer kontrolü:
 - Saldırıları tespit etmek amacıyla, düğümlerin kalan enerji seviyeleri ve bağlantı durumu kontrol edilmektedir.
 - Önceden belirlenen bir eşik değeri aşıldığında, saldırı yapan düğümler tespit edilir.
- GA ile yönlendirme:
 - Yönlendirme sürecinde GA kullanılarak en uygun yol bulunur.
 - GA'da kullanılan temel adımlardan biri kromozom temsilidir. Her olası yol bir kromozom olarak kodlanmaktadır.
 - Diğer temel adım ise uygunluk fonksiyonudur. Rotaların uygulunun hesaplanmasında kullanılır. Matematiksel gösterimi Denklem (2.27)'deki gibidir.

$$f_i = \frac{1}{\sum_{i=1}^c \sum_{j=1}^{l_i-1} C(g_i(j), g_i(j+1))} \quad (2.27)$$

Burada f_i kromozomun uygunluk değeri, l_i kromozomun uzunluğu ve $g_i(j)$ ise kromozomdaki j 'nci düğümdür.

- HC ile yerel optimizasyon:
 - HC algoritması, GA tarafından belirlenen rotayı daha da optimize etmek için kullanılır. Komşu çözümler değerlendirilerek daha iyi bir çözüm aranmaktadır.
- Veri iletimi:
 - Yukarıdaki algoritma ve yöntemler ile belirlenen en uygun yollar üzerinden veri paketleri iletilmektedir [33].

2.2.4.2. E-MAVMMF

E-MAVMMF algoritması, MANET ortamında enerji verimliliğini artırmak ve ağ ömrünü uzatmak amacıyla geliştirilmiştir. Algoritma, iki aşamadan oluşmaktadır: 1. Aşama olan KB seçiminde, Afrika Akbaba Optimizasyon Algoritması'nın (AVOA) geliştirilmiş bir versiyonu kullanmakta ve Brownian hareketi ile optimize edilmektedir. Optimum rota seçiminde ise, Modifiye Edilmiş Mayıs Sineği Algoritması (MMA) kullanılarak en iyi çözüm bulunmaktadır.

E-MAVMF algoritmasının işleyişi:

- KB seçimi AVOA yaklaşımı:
 - Başlangıçta belirlenen düğüm sayısı kadar akbaba popülasyonu rastgele oluşturulmaktadır.
 - İki grup akbaba fitness fonksiyonuna göre ayrılmaktadır. En iyi akbabanın belirlenmesi, Denklem (2.28)'deki gibi rulet tekerleği seçimi yöntemi kullanılarak yapılmaktadır.

$$k(i) = \begin{cases} \text{Best vulture1} & \text{if } l_i = o_1 \\ \text{Best vulture2} & \text{if } l_i = o_2 \end{cases} \quad (2.28)$$

Burada $k(i)$ seçilen en iyi akbabanın pozisyonu, l_i rastgele belirlenen bir parametre, o_1 ve o_2 rastgele üretilen olasılık değerleridir.

- Açlık modeli:
 - Akbabaların açlık durumuna bağlı olarak hareketleri belirlenir. Açlık faktörü şu Denklem (2.29) ile belirlenmektedir.

$$g = (2 * rand_1 + 1) * e * \left(1 - \frac{iteration_1}{Maxiterations}\right) + b \quad (2.29)$$

Burada g açlık faktörü, $rand_1$ rastgele sayı, e açlık katsayısı, $iteration_1$ mevcut iterasyon sayısı, $Maxiterations$ maksimum iterasyon sayısı ve b ise dengeleme faktörüdür.

- Keşif ve sömürü aşamaları:
 - Akbabalar en iyi çözüme yönlendirilerek keşif aşamasında yeni pozisyonlar Denklem (2.30)'daki formüle göre belirlenmektedir.

$$l(j + 1) = k(j) - d(j) * g \quad (2.30)$$

Burada $l(j + 1)$ güncellenmiş akbaba pozisyonu, $k(j)$ en iyi akbaba pozisyonu, $d(j)$ akbabanın mesafesi ve g ise açlık faktörüdür.

- KB ataması:
 - Seçilen KB'ler, enerji, mesafe, düğüm derecesi gibi kriterlere göre değerlendirilerek KB olarak atanmaktadır.

Optimum rota seçimi için MMA ile düğümlerin hız ve konum güncellemeleri, eylemsizlik ağırlığı, rastgele parametreler ve en iyi yerel/küresel çözümler göz önünde bulundurulurularak yapılmaktadır [36].

2.2.4.3. MEAFN-DBA

MEAFN-DBA, Uçan Tasarsız Ağlar (FANET) için önerilmiş bir kümeleme ve kaynak paylaşımı algoritmasıdır. Algoritma, DBSCAN tabanlı bir kümeleme mekanizmasını kullanarak ağdaki İnsansız Hava Araçları (İHA) yoğunluk temelli gruplara ayırmaktadır. Bunun yanı sıra, dağıtık sistemlerde kaynak paylaşımını yöneten karşılıklı dışlama ilkelerini kullanarak veri iletimini ve kaynak tahsisini optimize etmeyi amaçlamaktadır.

DBSCAN algoritması, belirli bir yoğunluk eşliğini aşan düğümleri kümeleyerek uç birimlerin daha verimli bir şekilde gruplandırılmasını sağlar. Algoritmanın temel

özellikleri şunlardır:

- Yoğunluk temelli kümeleme: DBSCAN, önceden belirlenmiş bir epsilon (ϵ) mesafesi içinde belirli bir sayıda komşuya (MinPts) sahip olan noktaları, çekirdek nokta olarak tanımlamaktadır.
- Bağlantılı düğümler: Bir düğüm, belirlenen yoğunluk eşiğini aşan başka bir çekirdek düğümlerle doğrudan veya dolaylı olarak bağlantılıysa aynı kümede bulunur.
- Gürültü ve ayrık düğümler: Yeterli komşuya sahip olmayan noktalar gürültü olarak kabul edilir ve kümeye dahil edilmezler.

MEAFN-DBA algoritmasının işleyişi:

- Kümeleme süreci:
 - İHA'ların belirli bir iletim menzili (TR) belirlenir.
 - DBSCAN algoritması kullanılarak, belirlenen TR ve MinPts parametreleri ile kümeleme işlemi gerçekleştirilir.
 - KB'ler seçilir ve her İHA, kendi komşularının yoğunluklarına göre kümelere atanmaktadır.
- Jeton tabanlı karşılıklı dışlama:
 - KB'ler belirlenen küme içi iletişimi yönetir.
 - Jeton tabanlı karşılıklı dışlama yöntemi kullanılarak, kaynak paylaşımı düzenlenir.
 - Her KB, kendisine bağlı düğümlerin kaynak taleplerini takip eder ve jeton sahibine öncelikli erişim sağlar.
 - Eğer bir düğüm Kritik Bölge (CS) erişimi talep ederse, KB bunu kaydeder ve uygun sırayla jeton atamasını gerçekleştirir.
 - CS talebinde bulunan düğümlerin talepleri sırasıyla işlenir ve jeton transfer edilir.
- Jeton dolaşımı ve kaynak paylaşımı:
 - İlk olarak rastgele bir düğüme jeton atanır.
 - Jeton sahibi düğüm, CS erişimini kullanıp kullanmayacağına karar verir.
 - Eğer kullanmayacaksa, jeton en yakın KB'ye devredilir.

- KB, taleplerini kaydederek uygun düğüme jetonu yönlendirir.
 - Eğer belirli bir kümede CS talebi yoksa, jeton başka bir kümeye yönlendirilmektedir.
- Kaynak yönetimi ve senkronizasyon gecikmesi:
 - Algoritma, ağ içindeki mesaj karmaşıklığını azaltarak veri iletim süreçlerini optimize etmektedir.
 - Jeton geçişleri, en uygun düğümler üzerinden gerçekleştirilerek ağ yükü dengelenir.
 - İHA'ların hareketliliği dikkate alınarak dinamik olarak kümeler yeniden yapılandırılmaktadır [97].

2.2.4.4. MKMPE

MKMPE algoritması, MANET'lerde KB seçimi ve güvenli yönlendirme süreçlerini optimize etmek için geliştirilmiş hibrit bir yöntemdir. Algoritma, KB'leri belirlemek için Modifiye Edilmiş K-Means algoritmasını (MKM), güvenli ve enerji verimli yolların bulunması için ise Filipin Kartalı Optimizasyon (PEO) yöntemini kullanmaktadır.

MKM algoritması, güven unsurlarına dayalı olarak en uygun KB'leri belirler. Bu güven unsurları DT, IDT ve RT olarak tanımlanır. Ardından, PEO yöntemi kullanılarak ağda enerji kaybını en aza indiren, güvenli ve gecikme süresi düşük yönlendirme yolları oluşturulmaktadır.

MKMPE algoritmasının işleyişi:

- KB seçimi:
 - Ağda bulunan tüm düğümler, MKM algoritmasıyla KB adayı olarak değerlendirilir.
 - Düğümlerin KB seçimi güven faktörlerine dayalı olarak belirlenir. Bu faktörlerden DT, düğümlerin geçmiş paket iletimlerine göre hesaplanmaktadır. IDT, komşu düğümlerin verdikleri geri bildirimlere dayanmaktadır. RT ise DT ve IDT'nin ağırlıklı ortalaması alınarak hesaplanmaktadır. Güven hesaplama formülü Denklem (2.31)'deki gibidir.

$$T = \frac{1}{3}(DT + IDT + RT) \quad (2.31)$$

- Bu hesaplamalar sonunda güven skoru en yüksek olan düğüm KB olarak seçilmektedir. KB'ler küme üyeleri ile iletişim kurarak kümeleri oluşturmaktadırlar.
- Optimum rota seçimi:
 - İlk olarak ağdaki düğümler, kartal popülasyonu olarak modellenir.
 - Kartalların avlanma stratejisi kullanılarak en uygun yönlendirme yolu belirlenir.
 - Rotalar belirlenirken düşük enerji tüketimi olan yollar öncelikli seçilir.
 - Düğümlerdeki kalan enerjiyi hesaplamak için Denklem (2.32)'deki formül kullanılmaktadır.

$$E_{rem}(t) = E_{rem}(t-1) - E_{trans}(t-1, t) - E_{rec}(t-1, t) \quad (2.32)$$

- Burada $E_{rem}(t)$, t zamanındaki kalan enerji, $E_{trans}(t-1, t)$, t-1 ve t arasındaki veri iletimi için harcanan enerji ve $E_{rec}(t-1, t)$ ise t-1 ve t arasındaki veri alımı için harcanan enerjidir.
- Kartalların uçuş davranışlarına göre, yönlendirme rotası güncellenerek en iyi yol keşfedilmektedir [38].

Bu bölümde anlatılan yönlendirme protokollerinin özeti, avantaj ve dezavantajları Çizelge 2.2'de gösterilmektedir.

Çizelge 2.2. Kümeleme algoritmalarının özeti, avantaj ve dezavantajları.

Protokol	Algoritma Özeti	Avantajlar	Dezavantajlar
LID	En düşük kimlik (ID) değerine sahip düğümü KB olarak seçen, basit ve merkezi olmayan bir kümeleme yöntemidir. Düğümler periyodik olarak kimliklerini yayınlar ve en düşük ID'ye sahip olan düğüm KB olarak atanır.	Düğüm Algoritma, yalnızca düğüm kimliklerine dayandığı için uygulaması ve anlaşılması oldukça kolaydır. Sadece belirli düğümler KB olarak görev aldığından, kontrol yükü düşer.	En düşük kimliğe sahip düğümler sürekli KB olarak seçildiği için enerji tüketimleri hızla artar ve ağ performansını olumsuz etkiler. Büyük ağlarda, kimlik tabanlı KB seçim mekanizması yeterince verimli olmayabilir. Yalnızca ağ GD düğümleri aracılığıyla kümeler arası iletişim sağlandığından, bu düğümlere aşırı yük binebilir [11], [8].
HCC	Kümelere oluşturmak için bağlantı derecesine dayanır. En yüksek sayıda komşuya sahip düğüm KB olarak seçilir. İki veya daha fazla düğümün aynı bağlantı derecesi varsa, en düşük kimliğe sahip düğüm KB olarak seçilmektedir.	LID algoritmasına kıyasla daha az KB sayısına sahiptir. Yüksek dereceli düğümlerin seçimi sayesinde ağın ölçeklenebilirliğini artırmaktadır.	Yüksek hareketlilik durumunda KB değişimi sık yaşanır, bu da kararlılığı azaltmaktadır. Güncelleme paketlerinin fazlalığı nedeniyle ağ üzerindeki verim düşmektedir. Yeniden bağlanma süreçleri nedeniyle kontrol mesajları ve enerji tüketimi artmaktadır [12], [9].
k-KONID	LID ve HCC algoritmalarını birleştirerek KB seçiminde bağlantı derecesini birinci, düşük kimlik numarasını ise ikinci kriter olarak kullanmaktadır. Ağda oluşturulan küme sayısını minimize etmeyi amaçlayan bu yaklaşımda her KB, kendisine en fazla k-hop uzaklıktaki düğümleri içerecek şekilde kümeler oluşturmaktadır.	Ağda oluşan küme sayısını ve kümedeki düğüm sayısını en aza indirmektedir. Ağdaki ek yük LID algoritmasına göre daha düşüktür. KB değişimi azdır, bu da küme kararlılığını artırmaktadır.	KB'lerin enerjileri hızlı tükenmektedir. Her düğüm, k-hop mesafesindeki komşularına kümeleme kararını duyurduğundan, süreç zaman açısından verimli değildir [7].
LEACH	Enerji verimliliğini artırmak için kümeleme tabanlı çalışan bir yönlendirme protokolüdür. Ağ, belirli aralıklarla rastgele seçilen KB'ler tarafından yönetilen kümelere ayrılır ve KB'ler, veri toplama ve ana istasyona iletme görevlerini üstlenerek düğümler arasındaki enerji tüketimini dengelemektedir.	KB'ler tarafından yapılan veri birleştirme iletişim maliyetini ve iletişim gecikmesini düşürmektedir. Düğümleri uyku modunda tutarak enerji tasarrufu yapmaktadır. Merkezi kontrol gerektirmeyen basit ve dağıtılmış bir yapıya sahiptir.	KB'lerin enerjileri hızlı tükenmektedir. Her turun başında KB'lerin rastgele seçilmesi, ek iletişim maliyeti ve kontrol yükü oluşturmaktadır. KB seçimi rastgele olduğundan düşük enerjili düğüm KB olabilmektedir [15].

Çizelge 2.2. (devam) Kümeleme algoritmalarının özeti, avantaj ve dezavantajları.

Protokol	Algoritma Özeti	Avantajlar	Dezavantajlar
WCA	İletim gücü, düğüm hareketliliği, ideal düğüm derecesi ve pil gücü gibi parametreleri dikkate alarak KB seçiminde en düşük ağırlığa sahip düğümü KB olarak belirler. KB seçim süreci talebe bağlı olarak gerçekleştirilerek hesaplama ve iletişim maliyeti azaltılmaktadır.	Ağın dinamik doğasına uyum sağlayarak KB seçim sürecini ağın ihtiyaçlarına göre başlatır ve böylece gereksiz yeniden yapılandırılmaları önlemektedir. Birden fazla parametreyi dikkate alarak daha dengeli ve verimli bir kümeleme yapısı oluşturmaktadır.	Birden fazla parametrenin değerlendirilmesi, algoritmanın hesaplama karmaşıklığını artırabilmektedir. Farklı parametrelerin ağırlıklarının manuel belirlenmesi, algoritmanın etkinliği açısından önemlidir ve bu da ek bir zorluk oluşturmaktadır [8], [12].
WBACA	WCA'nın kümeleme sürecindeki küresel minimum yaklaşımına karşılık yerel minimumları kullanarak daha verimli bir kümeleme yapısı oluşturmaktadır. İletim gücü, iletim hızı, hareketlilik, pil gücü ve düğüm derecesi gibi parametreleri dikkate alarak en düşük ağırlığa sahip düğümü KB olarak seçmektedir.	Ağın kararlılığını artırarak KB değişimlerini minimize eder ve böylece ağın genel performansını iyileştirir. Algoritma, ağın dinamik doğasına uyum sağlayarak KB seçim sürecini optimize eder ve gereksiz yeniden yapılandırılmaları önlemektedir.	Birden fazla parametrenin değerlendirilmesi, algoritmanın hesaplama karmaşıklığını artırmaktadır. Düğüm hareketliliği arttıkça küme stabilitesi azalabilir ve daha sık KB değişimi gerektirebilmektedir [18], [8].
EE-WCA	KB seçiminde tampon uzunluğu, düğüm hareketliliği, pil seviyesi, bağlantı derecesi ve sinyal-gürültü oranı gibi beş QoS parametresini kullanarak daha verimli ve dengeli bir kümeleme yapısı oluşturmaktadır.	Düğümlerin enerji tüketimini optimize ederek ağ ömrünü uzatır. Kaynak kullanımını ağda dengeli bir şekilde dağıtarak aşırı yüklenmeleri engeller. Düğüm hareketliliği ve enerji durumuna göre sürekli olarak topoloji kontrolü yapar, böylece ağ performansını artırmaktadır.	Ağırlık hesaplama süreçleri ve çoklu parametre değerlendirilmesi yüksek hesaplama yüküne neden olmaktadır. Düğüm hareketliliği çok yüksek olan ağlarda küme kararlılığı düşmektedir [22], [24].
OSCA	MANET'lerde KB değişimlerini azaltarak ağ kararlılığını artırmayı ve kontrol yükünü en aza indirmeyi amaçlayan bir kümeleme algoritmasıdır. Yedek düğüm mekanizması ve çoklu parametre değerlendirmesi ile dinamik topolojilerde verimliliği artırmayı hedeflemektedir.	Yedek düğüm mekanizması sayesinde, KB değişimleri minimize edilir ve ağın kararlılığı artırılmaktadır. KB değişimlerinin azalması, yeniden kümeleme süreçlerini minimize eder ve böylece ağ üzerindeki kontrol yükünü azaltmaktadır.	Yedek düğüm tanıtımı, her kümede ek bir düğümün KB rolünü üstlenmeye hazır olmasını gerektirir, bu da kaynak kullanımını artırmaktadır. [20], [21].

Çizelge 2.2. (devam) Kümeleme algoritmalarının özeti, avantaj ve dezavantajları.

Protokol	Algoritma Özeti	Avantajlar	Dezavantajlar
MOPSO	PSO'nun çok amaçlı bir uzantısı olup, MANET'lerde dinamik topolojilerin yönetimi ve enerji verimliliğini artırmak için kullanılır. Düğüm kümeleme süreçlerini optimize ederek, bağlantı kalitesi ve iletişim maliyetleri gibi birden fazla metriği eş zamanlı olarak iyileştirmektedir.	Kümeleme sürecinde enerji tüketimini minimize ederek MANET'lerde ağ ömrünü uzatmaktadır. Algoritma, optimal çözümler üreterek farklı ağ koşullarına göre uyarlanabilir çözümler sunmaktadır.	PSO'nun çok amaçlı optimizasyon için geniş arama alanlarında kullanılması, yüksek hesaplama maliyetine neden olabilmektedir. Optimizasyon sürecinde parçacıkların belirli bölgelerde sıkışması nedeniyle, en iyi çözüme ulaşamama riski taşımaktadır [25].
CACO NET	ACO'nun sezgisel doğasını kullanarak dinamik ağlarda KB seçimini optimize eden bir algoritmadır. Feromon izleri ve sezgisel değerler kullanarak en iyi yolları belirler, böylece VANET'lerde iletişim maliyetini azaltmakta ve kümeleme kararlılığını artırmaktadır.	Feromon izlerini kullanarak düğümler arasındaki en uygun yolların seçimi, yönlendirme maliyetlerini azaltır. Çoklu hedef fonksiyonlarını optimize ederek, kümelerdeki düğüm sayısını ve toplam iletişim mesafesini dengelemektedir.	ACO tabanlı olması nedeniyle, feromon güncellenmesi yüksek hesaplama maliyeti gerektirmektedir. Ağ yoğunluğu arttıkça feromon güncellemeleri ve karar verme süreci daha karmaşık hale gelmekte bu da yanıt süresini artırmaktadır [27].
HFAPSO	Kablosuz sensör ağlarında enerji verimliliğini artırmak ve ağ ömrünü uzatmak için FA ve PSO tekniklerini birleştiren hibrit bir kümeleme algoritmasıdır. FA'nın küresel arama yeteneği ve PSO'nun yerel optimizasyon gücünü kullanarak en uygun KB'leri seçerek enerji tüketimini minimize etmeyi hedeflemektedir.	PSO ve FA algoritmalarının birleşimi sayesinde daha hızlı küresel arama ve yerel optimizasyon sağlamaktadır. Enerji tüketimini azaltarak ağın ömrünü uzatmaktadır. Düşük enerji tüketimiyle daha fazla aktif düğüm sayısı sağlamaktadır.	Karmaşık yapı nedeniyle hesaplama maliyeti artmaktadır. Daha uzun iterasyon süreleri gerektirir, bu da gecikmelere yol açmaktadır [29], [30].
F-CAPSO	MANET'lerde enerji verimli ve güvenli veri iletimi sağlamak için bulanık mantık, (SSA) ve (SKA) tabanlı bir kümeleme algoritmasıdır. KB seçim sürecini optimize ederek ağ ömrünü uzatmayı ve güvenilir yönlendirme için Çift Dalga Şifreleme stratejisini kullanmayı hedeflemektedir.	KB seçiminde hibrit yöntemler kullandığı için enerji verimliliği ve ağ ömrü artırmaktadır. Karmaşık topolojilerde güvenli veri iletimi sağlar. Uyarlanabilir optimizasyon yaklaşımı sayesinde düğüm hareketliliğine karşı dayanıklıdır.	Algoritmanın hesaplama yükü yüksektir ve büyük ölçekli ağlarda gecikmeyi artırmaktadır. KB rotasyonu sık gerçekleştiğinde ek enerji tüketimine yol açmaktadır [32].

Çizelge 2.2. (devam) Kümeleme algoritmalarının özeti, avantaj ve dezavantajları.

Protokol	Algoritma Özeti	Avantajlar	Dezavantajlar
GAHC	MANET’lerde çoklu yol yönlendirme için optimize edilmiş hibrit bir protokoldür. GA ve HC algoritmalarının birleşimi sayesinde, KB seçiminde ve en iyi rotanın belirlenmesinde etkin bir çözüm sunmaktadır. Algoritma, düğüm güvenilirliği, bağlantı durumu ve enerji verimliliği gibi parametreleri değerlendirerek ağ performansını artırmayı hedeflemektedir.	Enerji tüketimini optimize ederek düğümlerin ömrünü uzatmaktadır. Hibrit yaklaşımı sayesinde güvenlik saldırılarını tespit etme ve önleme kabiliyeti yüksektir. Algoritma, güvenilir bağlantılar seçerek ağ performansını artırmaktadır.	Hibrit algoritma kullanılması, hesaplama maliyetini artırmaktadır. KB seçimi ve yönlendirme sürecinde ek kontrol mesajları nedeniyle ağ trafiği artmaktadır. Dinamik topolojilerde KB sık güncellendiğinden enerji tüketimini artırmaktadır [34], [35].
E-MAVM MF	MANET’te enerji verimliliğini artırmak ve ağ ömrünü uzatmak için geliştirilmiştir. KB seçiminde geliştirilmiş AVOA algoritması ve rota seçiminde ise MMA algoritması kullanılarak optimum yönlendirme sağlanmaktadır.	KB seçiminde AVOA kullanarak enerji tüketimi dengeli bir şekilde dağıtmaktadır. MMF algoritması ile en optimum yolu belirleyerek enerji tasarrufunu artırmaktadır. Ağın genel enerji tüketimi minimize ederek ağ ömrü uzatmaktadır.	AVOA ve MMF algoritmalarının birlikte kullanılması ek işlem yükü getirmektedir. Kümeleme ve yönlendirme süreçlerinde fazladan bilgi değişimi gerektirdiğinden dolayı bant genişliği tüketimi artmaktadır. [37].
MEAFN-DBA	DBSCAN tabanlı kümeleme mekanizmasını kullanarak İHA’ları belirli yoğunluk eşiklerine göre gruplandırırken, karşılıklı dışlama prensibiyle veri iletimi ve kaynak tahsisini yönetmektedir. Algoritma, çekirdek ve bağlantılı düğümleri belirleyerek kümeleri oluşturur ve yeterli komşuya sahip olmayan düğümleri gürültü olarak değerlendirir, böylece ağın verimliliğini artırmayı hedefler	DBSCAN algoritması sayesinde düğümler arasındaki yoğunluk temelli kümeleme sağlanarak daha dengeli ve kararlı kümeler oluşturulur. Kaynak paylaşımı için kullanılan jeton bazlı yöntem, veri çakışmalarını önleyerek senkronize bir iletişim sağlamaktadır.	DBSCAN’ın epsilon (ϵ) ve MinPts parametreleri doğru ayarlanmazsa, kümeleme performansı olumsuz etkilenmektedir. Token yönetimi ve küme güncellemeleri, ekstra hesaplama yükü oluşturmaktadır [97].
MKMPE	KB seçiminde MKM algoritması kullanılırken, güvenli ve enerji verimli yolların belirlenmesi için PEO yöntemi uygulanmaktadır. MKM, güven unsurlarını (DT, IDT, RT) değerlendirerek en uygun KB’leri seçerken, PEO ise düşük gecikmeli ve enerji tasarruflu yönlendirme yolları oluşturmayı hedeflemektedir.	KB seçimi ve rota optimizasyonu, düğümlerin enerji tüketimini minimize edecek şekilde gerçekleştirilmektedir. Düğüm hareketliliği ve kalan enerjiyi dikkate alarak KB seçimi yaparak ağın daha uzun süre çalışmasını sağlamaktadır.	KB seçimi ve rota belirleme süreçleri optimizasyon algoritmalarına dayandığı için işlem yükü fazladır. KB değişim mekanizması belirli periyotlarda gerçekleştiğinden, düşük enerjili düğümlerin yükü artmaktadır [37].

3. DİNAMİK MANETLER İÇİN HİBRİT ADAPTİF KÜMELEME ALGORİTMASI

MANET'ler, dinamik yapıları ve merkezi bir yönetim mekanizmasının bulunmaması nedeniyle kümeleme algoritmalarına ihtiyaç duymaktadır. Mevcut kümeleme yaklaşımları, düğüm hareketliliği, enerji tüketimi ve sık KB değişimleri gibi temel zorluklarla karşı karşıya kalmaktadır. Özellikle, klasik WCA algoritması gibi yöntemler, KB seçiminde belirli parametrelere dayalı sabit ağırlıklar kullanmakta olup, bu durum ağırlık değişken topolojisine uyum sağlamada yetersiz kalmaktadır.

Bu çalışmada, WCA tabanlı kümeleme algoritmasının KB seçimindeki manuel ağırlık faktörü seçme yaklaşımı, GSA kullanılarak optimize edilmiş ve dinamik bir KB seçim mekanizması önerilmiştir. KB seçiminde ağırlık faktörleri, ağırlık mevcut durumuna göre adaptif olarak ayarlanarak daha kararlı ve enerji verimli kümeler oluşturulması sağlanmıştır. Bunun yanı sıra, küme üyelerinin belirlenmesi sürecinde DBSCAN kümeleme algoritması kullanılmış ve bu algoritma geliştirilerek daha etkili bir düğüm gruplama mekanizması oluşturulmuştur.

Önerilen Dinamik Manet'ler için Hibrit Adaptif Kümeleme Algoritması (DMHAKA) yöntemi, WCA'nın kümeleme mantığını, GSA'nın optimizasyon yeteneğini ve DBSCAN'ın yoğunluk tabanlı kümeleme yapısını birleştirerek MANET'ler için ölçeklenebilir, enerji verimli ve kararlı bir kümeleme mekanizması sunmayı amaçlamaktadır.

DMHAKA yöntemi iki aşamadan oluşur: KB seçimi ve kümeleme. Aşağıdaki alt bölüm 3.1, en iyi KB'yi seçmek için GSA-WCA'nın Entegrasyonunu anlatılmaktadır. Alt bölüm 3.2, geliştirilmiş DBSCAN kullanılarak seçilen en uygun KB'ye göre düğüm rollerini belirleyerek kümelemeyi sunmaktadır.

3.1. EN İYİ KB'Yİ SEÇMEK İÇİN GSA- WCA ENTEGRASYONU

Önerilen yöntemde, KB seçim sürecinin iyileştirilmesi amacıyla WCA'nın kümeleme mekanizmasının KB seçim fazı ve GSA'nın optimizasyon yeteneği birleştirilerek dinamik, enerji verimli ve uyarlanabilir bir yapı oluşturulmuştur.

Bu bölümde, öncelikle WCA'nın temel prensipleri ve kümeleme yaklaşımı ele alınacak,

ardından GSA'nın optimizasyon süreçleri ve karar mekanizmasındaki rolü açıklanacaktır. Son olarak, GSA-WCA entegrasyonu ile önerilen hibrit KB seçim sürecinin çalışma prensibi detaylandırılacaktır.

3.1.1. WCA'ya Genel Bakış

WCA, MANET'te dinamik topolojilerin yönetimi için tasarlanmış, dağıtık ve isteğe bağlı bir kümeleme algoritmasıdır. Bu algoritma, KB seçiminde düğümlerin derecesi, düğümler arası mesafe toplamı, hareketlilik ve pil gücü parametrelerini dikkate alarak her düğüm için bir ağırlık hesaplamaktadır. Böylece en iyi KB'yi bulup, ağın kararlılığını ve performansını artırmayı hedeflemektedir.

Buna göre WCA'nın temel özellikleri;

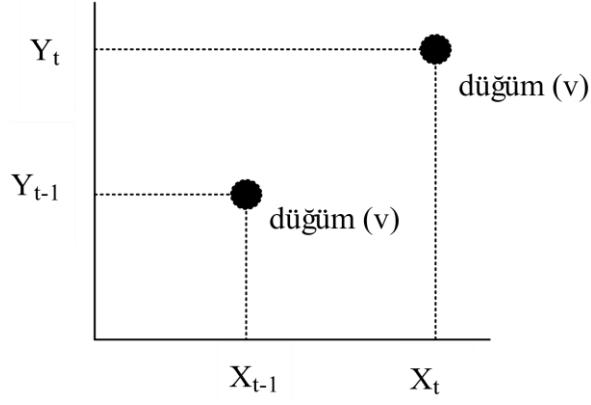
- Çoklu parametre değerlendirmesi: WCA, her düğüm için bir ağırlık değeri hesaplar ve bu hesaplamada aşağıdaki faktörleri göz önünde bulundurmaktadır.
 - Derece farkı (Δ_v): düğümlerin sahip oldukları komşu sayıları için hesaplanan derece farkıdır ve $\Delta_v = |d_v - \delta|$ şeklinde gösterilmektedir. Burada d_v mobil düğümün komşu sayısı ve δ ise KB'nin önceden tanımlanmış ideal olarak işleyebileceği düğüm sayısını göstermektedir ve bir eşik değeri olarak kabul edilmektedir.
 - Düğümün mesafeler toplamı (D_v): Düğümün tüm komşularına olan mesafelerin toplamıdır. Matematiksel ifadesi Denklem (3.1)'deki gibidir.

$$D_v = \sum_{v' \in N(v)} \{dist(v, v')\} \quad (3.1)$$

- Hareketlilik (M_v): Düğümün hareketlilik hızı, yani zaman içindeki konum değişikliğine bağlı mobilite seviyesidir. Formülü Denklem (3.2)'deki gibidir.

$$M_v = \frac{1}{T} \sum_{i=1}^T \sqrt{(X_t - X_{t-1})^2 + (Y_t - Y_{t-1})^2} \quad (3.2)$$

Buradaki (X_t, Y_t) Şekil 3.1)'de gösterilen koordinat düzlemindeki düğüm v 'nin (t) zamanındaki koordinatı (X_{t-1}, Y_{t-1}) ise $(t - 1)$ zamanındaki koordinatlarıdır.



Şekil 3.1. v düğümünün belirli zamanlardaki gösterimi [16].

- Pil gücü (P_v): Düğümün harcanan enerji seviyesidir.

Bu parametrelerin birleşimi, her düğüm için bir ağırlık değeri oluşturur ve en düşük ağırlık değerine sahip düğümler KB olarak seçilmektedir.

- Düğüm ağırlığının hesaplanması: Her bir düğümün ağırlığı Denklem (3.3)'te gösterildiği gibi hesaplanmaktadır.

$$W_v = w_1 \Delta_v + w_2 D_v + w_3 M_v + w_4 P_v \quad (3.3)$$

Buradaki ağırlık faktörleri olan w_1, w_2, w_3 ve w_4 manuel olarak verilmekte ve $\sum_{i=1}^4 c_i = 1$ şeklinde bir kısıt bulunmaktadır.

- KB Seçimi: WCA, KB seçim sürecini periyodik olarak değildir, ağırlık ihtiyaçlarına göre başlatır. Her bir düğümün ağırlığı hesaplandıktan sonra iletim aralığındaki düğümler arasında en küçük ağırlığa sahip düğüm KB olarak seçilmektedir.
- İsteğe Bağlı KB Seçimi: WCA, KB seçim sürecini periyodik olarak değildir, ağırlık ihtiyaçlarına göre başlatır. Bu sayede hesaplama ve iletişim maliyetleri azaltılmaktadır [7], [16].

Geleneksel WCA yönteminde, ağırlık faktörleri ilk ağ kurulumunda manuel olarak atanır, sabit kalır ve en düşük ağırlığa sahip düğüm KB olarak seçilir. Ancak, önerilen DMHAKA yönteminde, düğümlerin ağırlık faktörleri GSA algoritması kullanılarak ağ koşullarına göre optimize edilir ve en yüksek ağırlığa sahip düğüm KB olarak seçilmektedir.

3.1.2. GSA'ya Genel Bakış

Newton'un kütle çekim yasasından esinlenen bir sezgisel optimizasyon algoritmasıdır. Bu algoritma, parçacıkların (ajanların) birbirine kütle çekimi kuvveti uyguladığı ve en iyi çözüme doğru hareket ettiği bir mekanizma üzerine kuruludur. Önerilen DMHAKA yönteminde GSA, ağ topolojisinin dinamik yapısını dikkate alarak KB seçim sürecinde WCA'nın ağırlık faktörlerini optimize ederek en uygun düğümü belirlemek için kullanılmaktadır.

Önerilen çalışmada GSA'nın tercih edilme nedeni; esnekliği, uygulanabilirliği ve hızlı yakınsama özelliğidir. GSA'daki arama ajanları, performansları kütlelerine göre ölçeklenen nesnelere kabul edilir. Tüm bu nesnelere birbirlerini yerçekimi ile çeker ve bu kuvvet tüm nesnelere daha büyük kütleyle sahip nesnelere doğru hareket etmesine neden olmaktadır [98].

GSA'nın temel çalışma prensibi şu şekildedir:

- Kütle çekim kuvveti: Her ajan, çözüme olan uygunluğuna bağlı olarak farklı bir kütleyle sahiptir. Daha iyi çözümler daha büyük kütleyle sahip olur ve diğer ajanları kendine doğru çekmektedir.
- Hareket denklemi: Ajanlar, kütle çekim kuvvetine bağlı olarak hızlarını ve konumlarını güncelleyerek en iyi çözüm noktasına doğru ilerler.
- Zamanla optimizasyon: Algoritma iterasyonlar boyunca güncellenerek en iyi çözüme yakınsama sağlamaktadır.

KB seçimi sürecinde, GSA, düğümlerin enerji seviyesi, bağlantı derecesi ve hareketlilik parametrelerinin ağırlık faktörlerini optimize ederek en uygun KB'yi belirlemekte kullanılır. Böylece enerji verimli ve dengeli bir kümeleme yapısının ilk aşaması oluşturulmaktadır.

GSA'nın algoritmasının başlangıcında N ajanlı bir düğümün i 'ninci ajanının konumu Denklem (3.4)'te tanımlandığı gibidir.

$$X_i = (x_i^1, \dots, x_i^d, \dots, x_i^n) \quad i = 1, 2, \dots, N \quad (3.4)$$

Formülde, n , problemin boyutunu ifade eder ve x_i^d , i 'ninci ajanın d 'ninci boyuttaki konumunu tanımlamaktadır. Ajanların başlangıç konumları rastgele seçilmektedir.

Her iterasyonda ajanların uygunluk (fitness) değerleri hesaplanarak en iyi (best) ve en kötü (worst) fitness değerleri Denklem (3.5) ve Denklem (3.6)'da gösterildiği gibi belirlenmektedir.

$$best(t) = \max fit_i(t) \quad \forall i \in N \quad (3.5)$$

$$wors(t) = \min fit_i(t) \quad \forall i \in N \quad (3.6)$$

Burada $fit_i(t)$, i 'ninci ajanın t zamanındaki fitness değerini ifade etmektedir.

GSA'da yerçekimi sabiti olan $G(t)$ Denklem (3.7)'de gösterilmektedir.

$$\begin{aligned} G(t) &= G(G_0, t) \\ G(t) &= G_0 e^{-\frac{\alpha t}{T}} \end{aligned} \quad (3.7)$$

$G(t)$, başlangıçtaki değeri G_0 ve zaman t 'ye bağlı olarak azalmaktadır. Burada α azalma katsayısı ve T ise toplam iterasyon sayısını göstermektedir.

Her ajan için güncellenmiş kütle değerleri Denklem (3.8)'de gösterilmektedir.

$$\begin{aligned} m_i(t) &= \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \\ M_i(t) &= \frac{m_i(t)}{\sum_{k=1}^N m_k(t)} \end{aligned} \quad (3.8)$$

Burada $M_i(t)$, i 'ninci ajanın toplam kütlelerini temsil ederken, $m_i(t)$ ise fitness değerine bağlı olarak normleştirilmiş kütle değerini göstermektedir.

Yukarıdaki denklemler ışığında, GSA ajanlarının uygunluklarına bağlı olarak ağırlıkları güncellenmekte ve ajanlar en iyi çözüme doğru optimize edilmektedir. Yerçekimi sabiti zamanla azalırken, ajanların hareketi de hassaslaşmaktadır. Böylece küresel optimizasyondan yerel iyileştirmeye geçiş sağlanmış olmaktadır.

Belirli bir t zamanındaki j kütlelerinin i kütleleri üzerindeki kuvvet etkisi Denklem (3.9) ile verilmektedir.

$$F_{ij}^d(t) = G(t) \frac{M_{pi}(t) x M_{aj}(t)}{R_{ij}(t) + \varepsilon} (x_j^d(t) - x_i^d(t)) \quad (3.9)$$

$$R_{ij}(t) = \|X_i(t), X_j(t)\|_2 \quad (3.10)$$

$F_{ij}^d(t)$, Newton'un çekim yasasını kullanarak ajanların birbirine çekim uyguladığını göstermektedir. Bu çekim, ajanların birbirlerine daha yakın hareket etmelerini ve optimum çözüme yönelmelerini sağlamaktadır.

Denklem (3.10)'daki $R_{ij}(t)$ ajanların çekim kuvvetinin büyüklüğünü belirleyen önemli bir faktördür. Daha kısa mesafeler, daha güçlü çekim kuvveti oluşturmaktadır. Ayrıca bu denklemde i ve j ajanları arasındaki öklid mesafesinin hesaplanmasını gösterilmektedir. Bu denklemler sayesinde GSA, optimizasyon sürecinde daha iyi çözümlere ulaşmak için ajanların etkileşimini ve hareketini belirlemektedir. Algoritmaya stokastik bir özellik kazandırmak amacıyla, i ajanına uygulanan toplam kuvvetin d 'nci boyutundaki bileşeni, Denklem (3.11)'de gösterildiği gibi rastgele ağırlıklandırılmış bir toplam olarak tanımlanmaktadır.

$$F_i^d(t) = \sum_{j=1, j \neq i}^N rand_j F_{ij}^d(t) \quad (3.11)$$

Denklemdaki $F_i^d(t)$, i ajanının d boyutundaki toplam çekim kuvvetini göstermektedir. $rand_j$ parametresi (0- 1) aralığında olup rastgele belirlenebilmektedir. Böylece optimizasyon sürecine rastgelelik ekleyerek yerel minimumlara takılmak engellenmiş olmaktadır. Denklem (3.11) ve hareket yasası kullanılarak, i ajanının ivmesi, Denklem (3.12)'de gösterildiği gibi ifade edilmektedir.

$$a_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)} \quad (3.12)$$

Burada $a_i^d(t)$, i ajanının d boyutundaki ivmesidir. $F_i^d(t)$, i ajanının d boyutundaki toplam çekim kuvvetini göstermektedir. M_{ii} terimi ise, i ajanının atalet kütesini temsil etmektedir. Ajanın atalet kütesi büyüdükçe, ivmesi azalır. Yani, daha büyük kütleli ajanlar daha yavaş hareket eder, böylece küresel optimizasyon süreci dengeli olmaktadır.

Ayrıca daha büyük kütleli ajanlar, diğer ajanları kendilerine çekmekte ve böylece sistem dengeye ulaşmaktadır. Bir ajanın gelecekteki hızı Denklem (3.13)'te gösterildiği gibi mevcut hızı ile ivmesinin toplamı olarak hesaplanmaktadır.

$$v_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t) \quad (3.13)$$

Denklem (3.13)'teki $v_i^d(t+1)$, i ajanının d 'ninci boyutundaki yeni hızı, $rand_i$ (0- 1) aralığındadır ve ajanın keşif sürecine stokastik bir özellik kazandırmak için rastgele belirlenmektedir. $v_i^d(t)$, i ajanının önceki hızı ve $a_i^d(t)$ ise i ajanının önceki ivme değerini göstermektedir.

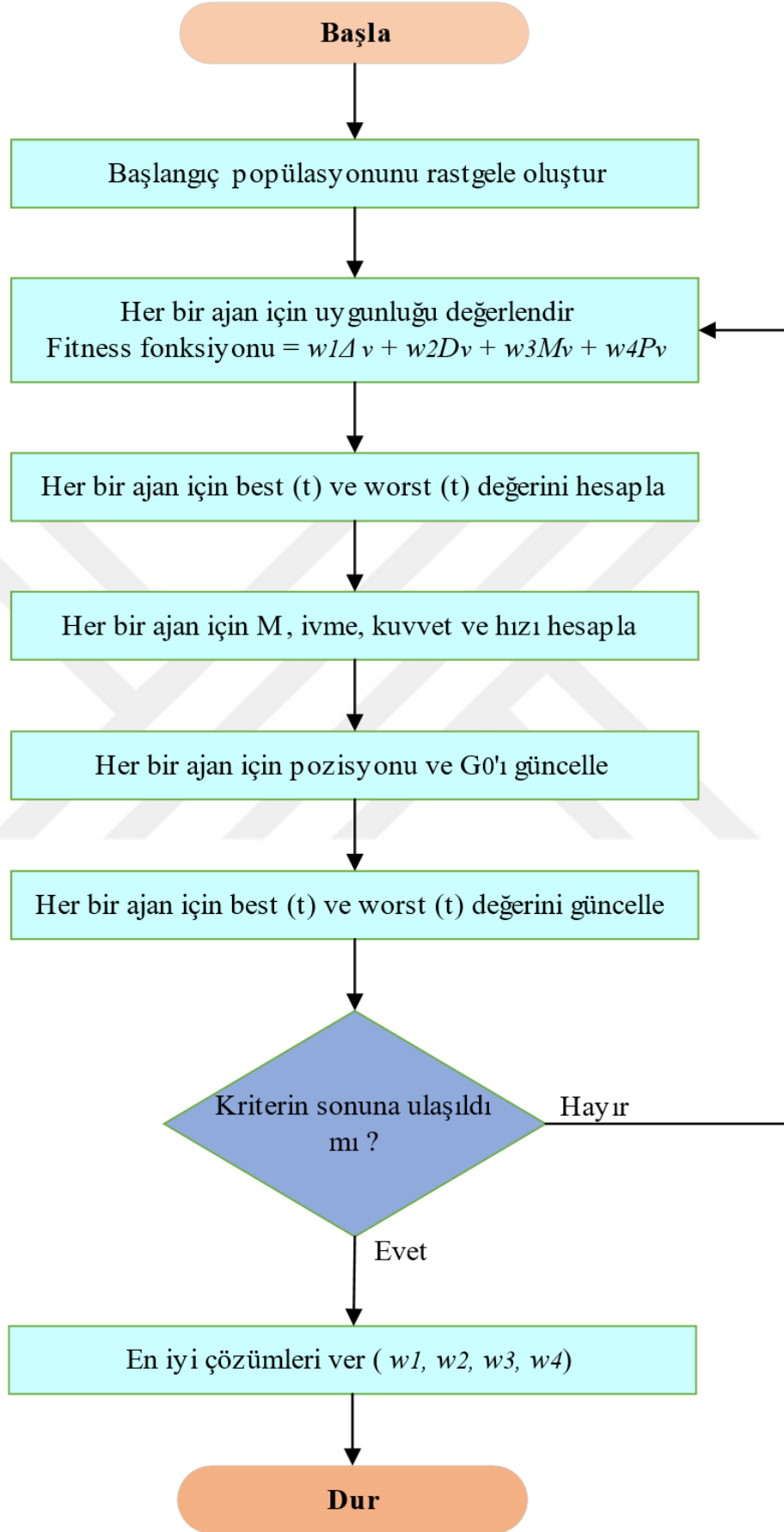
Denklem (3.14)'te bir ajanın konum güncelleme denklemi gösterilmektedir. Bu denkleme göre ajanın yeni konumunu belirlerken önceki konumuna yeni hızını ekleyerek güncelleme yapılmaktadır. Hızdaki değişim, ajanın önceki hareketinden etkilenecek daha optimal çözümlere ulaşmasını sağlamaktadır.

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \quad (3.14)$$

Burada $x_i^d(t+1)$ ifadesi, i ajanının d 'ninci boyutundaki yeni konumunu, $x_i^d(t)$, i ajanının önceki konumunu ve $v_i^d(t+1)$ i ajanının yeni hız değerini göstermektedir.

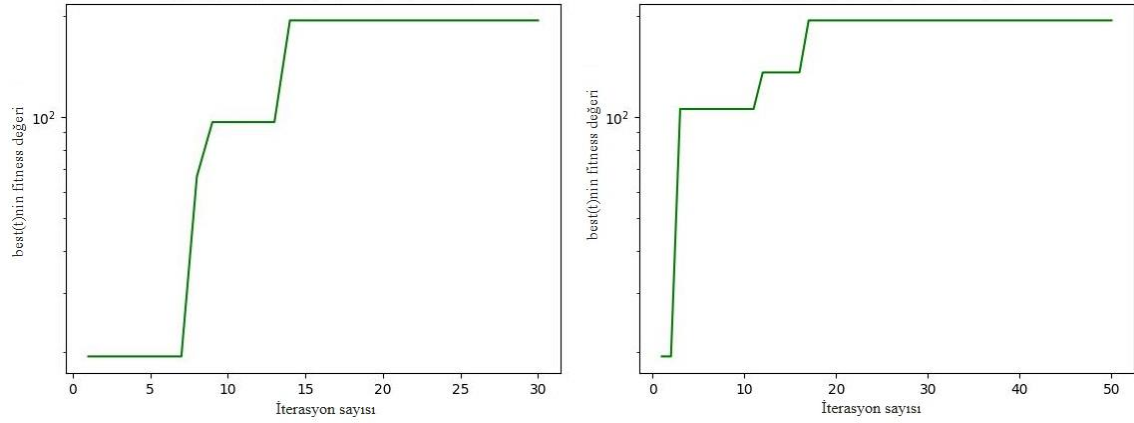
Denklem (3.13), ajanın yönlendirilmesini ve arama alanındaki keşif sürecini belirlerken, Denklem (3.14) ise ajanın bir sonraki iterasyondaki yerini belirlemektedir. Böylece algoritma, çözüme ulaşmak için Newton'un hareket yasaları ile yönlendirilmiş stokastik bir optimizasyon süreci oluşturmaktadır.

GSA'nın arama stratejisi ve algoritmanın akış diyagramı Şekil 3.2'te gösterilmektedir.



Şekil 3.2. GSA'nın akış diyagramı.

Şekil 3.3'te GSA'nın her düğüm için 30 ve 50 döngü süreli yakınsama grafikleri verilmektedir.



Şekil 3.3. GSA'nın yakınsama grafiği.

Bu çalışmada grafiklerde görüldüğü gibi en ideal döngü sayısı her düğüm için 20 olarak belirlenmiştir.

WCA algoritmasında, KB seçiminde kullanılan ağırlık faktörleri (w_1 , w_2 , w_3 ve w_4) ağın başlangıç kurulumu sırasında manuel olarak atanmakta ve ağ sonlandırılıncaya kadar sabit kalmaktadır. Ancak, MANET gibi dinamik ve sürekli değişen topolojilere sahip ağlarda bu yaklaşım yetersiz kalmakta ve ağ performansını olumsuz etkilemektedir. Önerilen DMHAKA yöntemi, GSA optimizasyon algoritmasını kullanarak ağırlık faktörlerini uyarlanabilir şekilde ayarlamakta ve böylece ağ topolojisindeki değişikliklere hızlı tepki verebilen dinamik KB seçimini mümkün kılmaktadır.

Denklem (3.3)'te, GSA'da maksimizasyon için kullanılan uygunluk fonksiyonunu tanımlamaktadır. Uygunluk fonksiyonundaki ağırlık faktörleri (w_1 , w_2 , w_3 ve w_4) için alt ve üst sınırlar, parametrelerin göreceli önemine göre şu şekilde belirlenmiştir:

$$w_1=0,1- 0,4, w_2= 0,01- 0,1, w_3= 0,01- 0,1 w_4= 0,2- 0,6.$$

Ayrıca, ağırlık faktörlerinin toplamı, parametreler arasında dengeyi sağlayan geleneksel $\sum_{i=1}^4 w_i = 1$ kısıtlamasını karşılamalıdır [98].

3.2. KÜMELEME VE DÜĞÜM ROLLERİNİN BELİRLENMESİ İÇİN GELİŞTİRİLMİŞ DBSCAN ALGORİTMASI

GSA-WCA ile ağırlıklarını hesaplayan düğümler, Node_probs paketi aracılığıyla bu bilgileri iletim aralığındaki diğer düğümlere iletmektedir. Elde edilen ağırlık değerleri doğrultusunda, en yüksek ağırlığa sahip düğüm KB olarak seçilir. Kümedeki düğümlerin rollerini içeren Cluster_info paketi, KB tarafından güncellenerek iletim aralığındaki düğümlere iletilir.

Ancak, WCA yönteminde KB, yalnızca kendi iletim aralığında bulunan bütün düğümlerle doğrudan kümeleme yapmaktadır. Bu yapı, KB'nin iletim aralığındaki tüm düğümleriyle doğrudan iletişim kurmasını gerektirerek KB'nin hızlı enerji tüketmesine ve küme elemanları fazla olduğu içinde küme içi iletişim ek yükünde artmasına neden olmaktadır. Bundan dolayı KB'nin ömrü kısaltmakta, sık sık KB değişimi gerçekleşmekte ve küme kararlılığı hızlı bozulmaktadır. Bu da yeniden kümeleme sürecinin erken başlamasına yol açarak ağın genel performansını olumsuz etkilemektedir.

Bu sorunu gidermek amacıyla, çalışmamızda WCA algoritmasının kümeleme mekanizması yerine, DBSCAN algoritması temel alınarak geliştirilmiş bir kümeleme yöntemi önerilmiştir. Makine öğrenmesinde denetimsiz öğrenme yöntemlerinden biri olan DBSCAN, kümeleme sürecinde düğümlerin topolojik konumları, yoğunluk dağılımı ve bağlantı durumlarını dikkate alarak daha dinamik, esnek ve enerji verimli kümeler oluşturulmasını sağlamaktadır. Böylece, KB'lerin dengeli seçimi ve düğüm rollerinin daha adaptif atanması mümkün hale gelmektedir.

DBSCAN, yoğunluk tabanlı kümeleme algoritmalarından biridir. Algoritma, verileri yoğunluklarına göre kümelendirerek, yeterli sayıda komşuya sahip olan noktaları bir küme içinde gruplamaktadır. Bağlantılı noktalardan oluşan yoğun bölgeleri kümeler hâlinde ayırırken, düşük yoğunluklu ve izole noktaları ise gürültü olarak sınıflandırmaktadır. DBSCAN, özellikle düzensiz ve değişken yoğunlukta veri kümelemek için uygun bir yöntemdir.

DBSCAN algoritması, kümeleme sürecinde iki temel parametreye dayanır:

Komşuluk mesafesi (Eps):

Bir noktanın komşu olarak kabul edilebilmesi için belirlenen maksimum mesafedir. Eğer bir düğüm, belirlenen eps mesafesi içinde belirli bir sayıda komşuya sahipse, bu düğüm

çekirdek nokta (core point) olarak değerlendirilmektedir. Eps, Denklem (3.15)'deki gibi öklid mesafe formülü ile hesaplanmaktadır.

$$d(p, q) = \sqrt{(X_p - X_q)^2 + (Y_p - Y_q)^2} \quad (3.15)$$

Burada:

- p ve q , iki farklı düğümü temsil eder.
- (X_p, Y_p) ve (X_q, Y_q) , bu düğümlerin koordinatlarıdır.
- $d(p, q)$ hesaplanan mesafeyi temsil etmektedir.
- Eğer $d(p, q) \leq \text{eps}$ ise, q düğümü, p düğümünün komşusu olarak kabul edilmektedir.

Minimum komşu sayısı (MinPts):

Bir noktanın çekirdek nokta olarak kabul edilebilmesi için sahip olması gereken minimum komşu sayısıdır. Matematiksel gösterimi Denklem (3.16)'da gösterildiği gibidir.

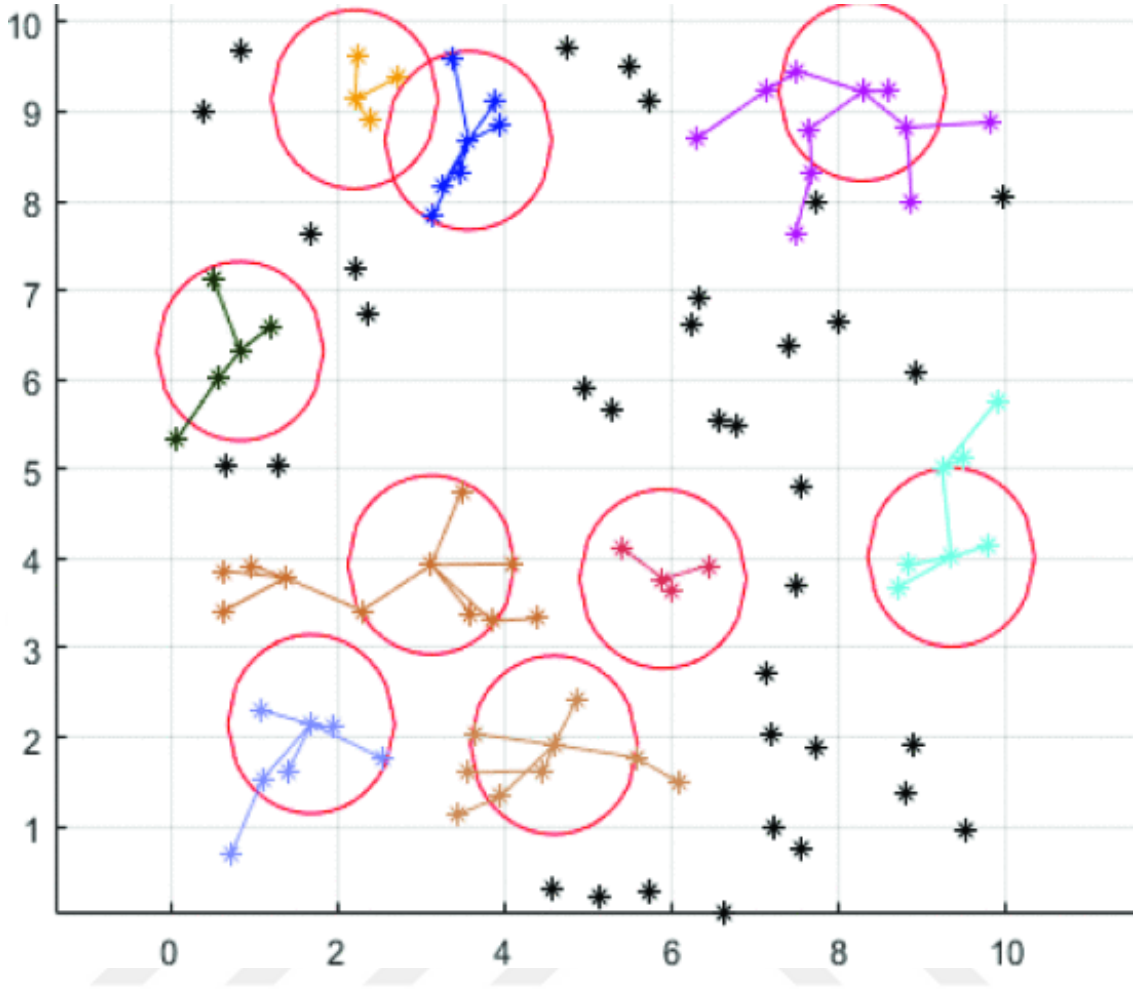
$$\text{Minpts} \geq \ln(N) \quad (3.16)$$

Burada:

- N , toplam düğüm sayısını ifade eder.
- $\ln(N)$, doğal logaritma fonksiyonunu temsil etmektedir.

Eğer bir düğüm, belirlenen MinPts değerinden daha az komşuya sahipse, bu düğüm sınır noktası veya gürültü noktası olarak değerlendirilmektedir [99].

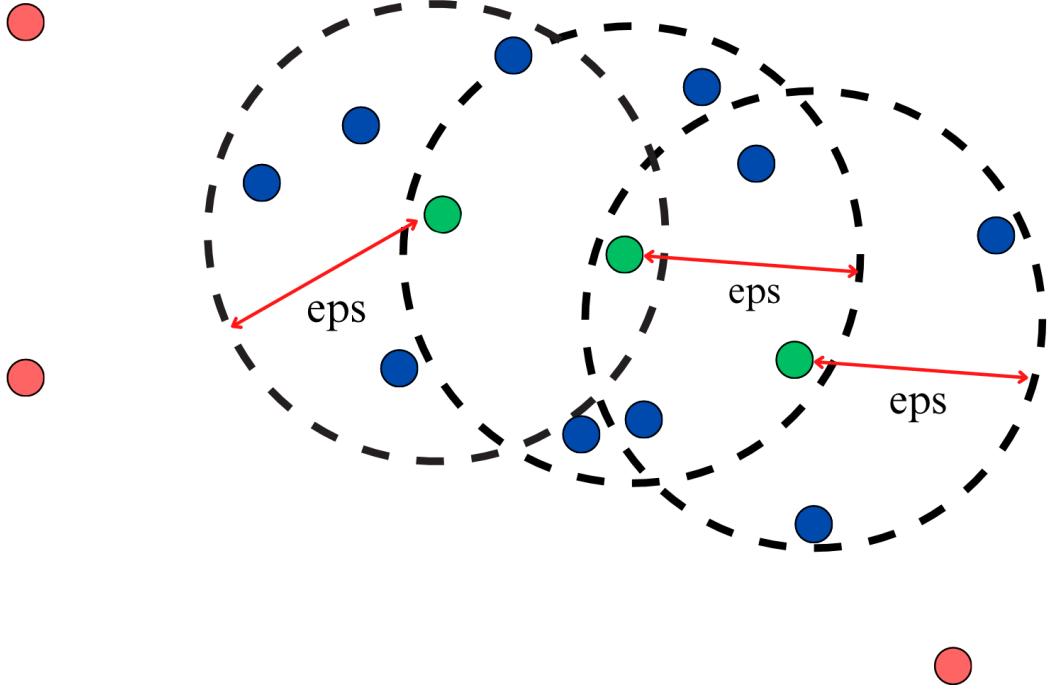
DBSCAN algoritmasının şekilsel gösterimi Şekil 3.5'te gösterilmektedir.



Şekil 3.4. DBSCAN kümeleme örneği [100].

DBSCAN algoritması, hiçbir kümeye katılmayan düğümleri belirleyebilme avantajına sahiptir. Bu çalışmada, DBSCAN algoritmasının çekirdek noktaları, GSA-WCA ile seçilen KB'ler olacak şekilde yapılandırılmıştır. Bu sayede, daha kararlı KB'ler seçilerek küme kararlılığı artırılmıştır. Ancak, eps parametresinin iletim mesafesi olarak belirlenmesi bazı sorunlara yol açmaktadır. Özellikle, uzak mesafelere veri göndermek, düğümlerin enerji tüketimini artırarak KB'lerin hızlı tükenmesine neden olur. Küme içi paket dağıtımının KB tarafından yönetilmesi, KB'nin enerji maliyetini artırarak küme stabilitesinin daha çabuk kaybolmasına ve yeniden kümeleme sürecinin erken başlamasına sebep olmaktadır. Bu durum, MANET gibi kısıtlı kaynaklara sahip ağlarda istenmeyen bir durumdur.

Ayrıca, DBSCAN algoritmasının MANET'lere uygulanmasında bazı dezavantajları vardır. Bunlardan biri Şekil 3.5'teki gibi örtüşen kümeler oluşturmasıdır.



● Çekirdek nokta ● Sınır noktası ● Gürültü noktası

Şekil 3.5. DBSCAN algoritmasında örtüşen kümeler.

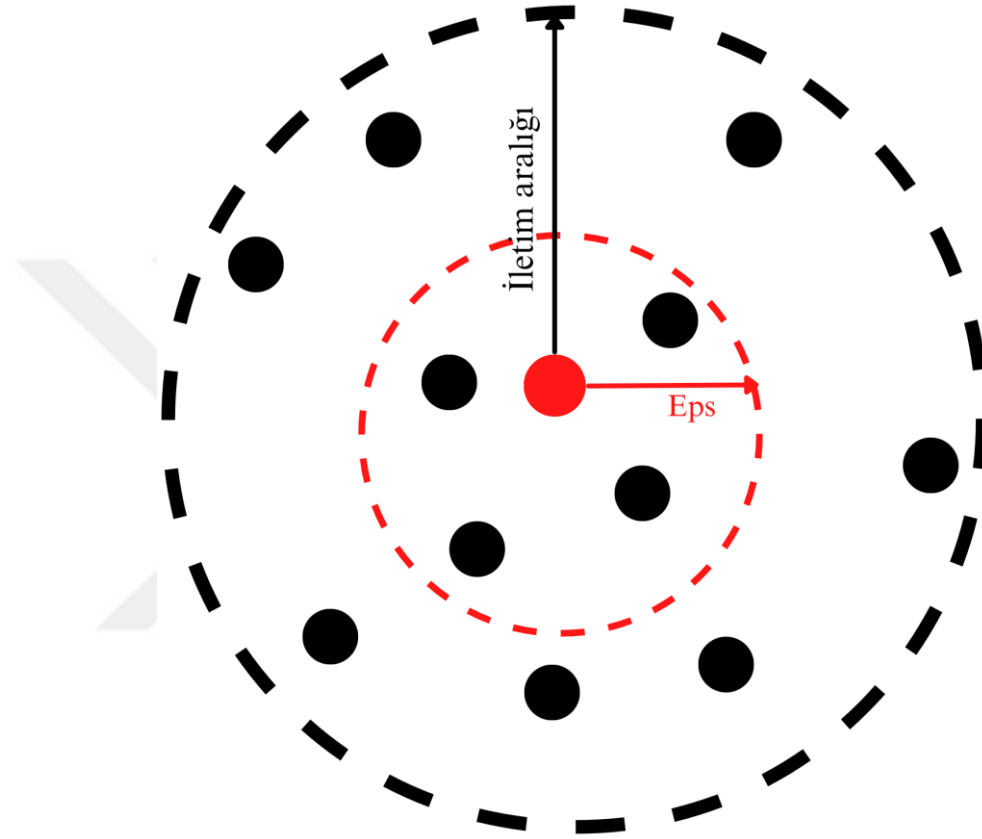
Bu sorun bir kümede KB haricinde küme elemanlarından birinin daha çekirdek nokta olması yani KB olmasıdır. Bu hem küme kararlılığını azaltacak hem de küme içi ve kümeler arası karmaşayı artıracaktır. Bundan dolayı ağ trafiği ve düğümlerin maliyetleri artacaktır.

Bu çalışmada, geleneksel DBSCAN algoritmasındaki sabit eps parametresi yaklaşımı yerine, önerilen Geliştirilmiş-DBSCAN yöntemi kapsamında, KB'ler için eps değeri dinamik olarak hesaplanmıştır. Bu hesaplama, Denklem (3.17)'de sunulan ve yerel düğüm yoğunluğunu esas alan özgün matematiksel formüle dayanmaktadır.

$$r_j = \sqrt{\frac{\sum_{i=1}^N \text{mesafe}^2(K_j, x_{ij})}{D_j}} \quad (3.17)$$

Burada K_j , GSA-WCA ile hesaplanan KB'nin konumunu, x_{ij} ise iletim aralığındaki düğümlerin konumlarını ifade etmektedir. D_j , KB hariç olmak üzere iletim aralığında bulunan düğüm sayısını temsil etmektedir. Denklemde hesaplanan değer her küme için

dinamik olup, kümelerin etkin mesafeyi adaptif olarak ayarlamasını sağlamaktadır. Bu yaklaşım, Şekil 3.6’da gösterildiği gibi WCA tarafından oluşturulan kümenin iletim aralığını azaltarak daha verimli bir kümeleme yapısı elde edilmesini sağlamaktadır. Ayrıca, Minpts değeri, bir KB’nin kendisi hariç olmak üzere bir küme oluşturabilmesi için gerekli olan minimum düğüm sayısını belirlemektedir ve her küme için iki olarak atanmıştır.



Şekil 3.6. KB iletim aralığı ve eps mesafesi.

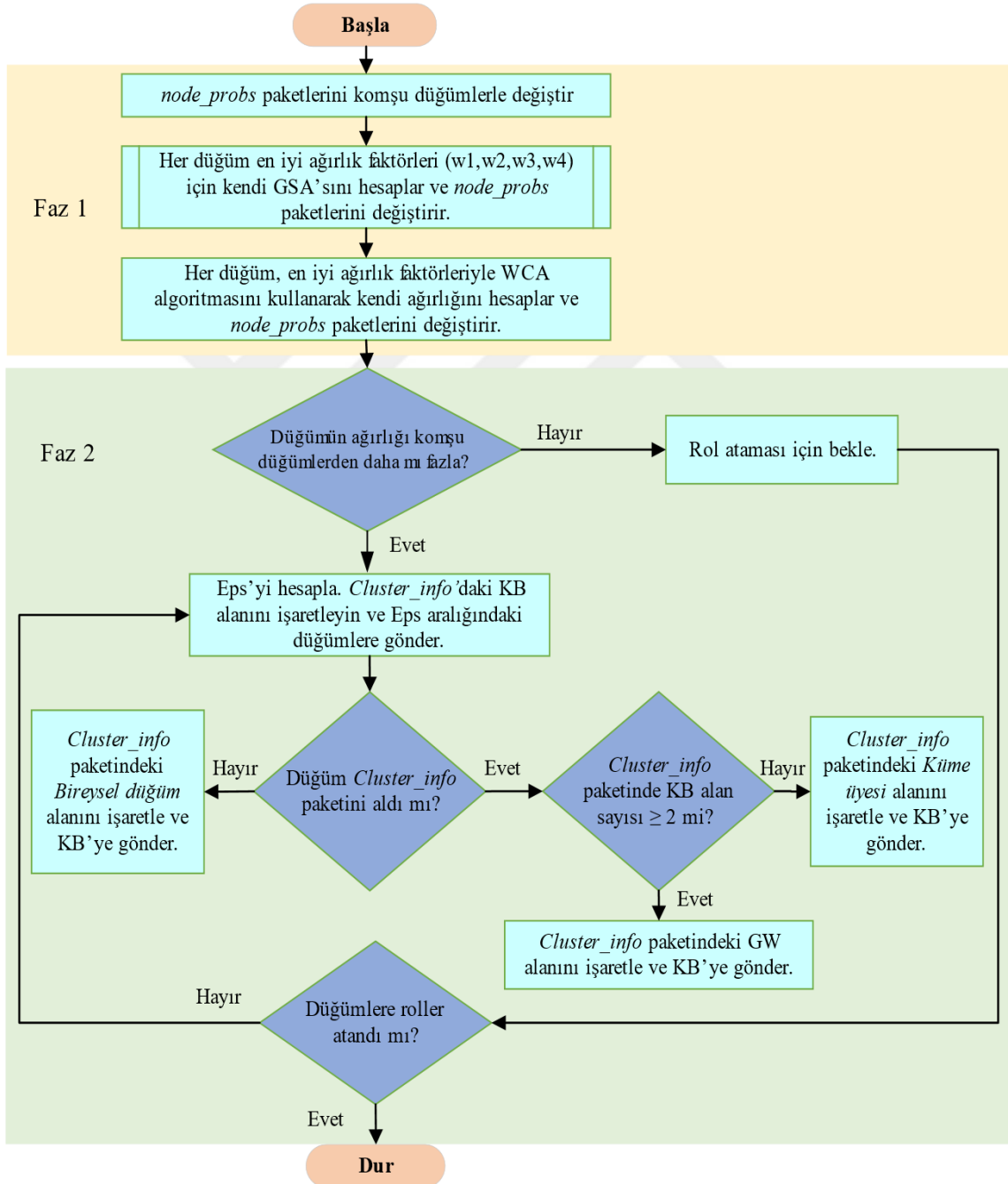
Her düğümde *Cluster_info* adında bir paket mevcuttur. *Cluster_info* paketi, düğümlerin ağ içindeki rollerini belirlemek ve kümeleme sürecini yönetmek amacıyla aşağıdaki bilgileri içermektedir:

- *Cluster_info*: Kümenin genel yapısına dair bilgiler,
- *KB_info*: KB olarak seçilen düğümün kimlik bilgisi,
- *Üye_bilgisi*: Küme üyesi (KÜ) olarak atanan düğümlerin bilgisi,
- *Geçityolu bilgisi*: Geçityolu Düğümlerinin (GD) kimlik bilgisi,
- *Bireysel_düğüm bilgisi*: Bireysel Düğümleri (BD) tanımlayan bilgi,

- *Düğüm enerji bilgisi*: Düğümlerin mevcut enerji seviyelerini gösteren bilgi,
- *GD_istek bilgisi*: Geçityolu isteğinde bulunan düğümlerin bilgisi.

İlk olarak bu paket, eps parametresi hesaplandıktan sonra KB tarafından iletim aralığındaki düğümlere gönderilmektedir.

Bu çalışma kapsamında önerilen hibrit dinamik kümeleme yöntemi ile düğümlere rol atanmasının akış diyagramı Şekil 3.7’de gösterilmektedir.



Şekil 3.7. Önerilen yöntemin akış diyagramı.

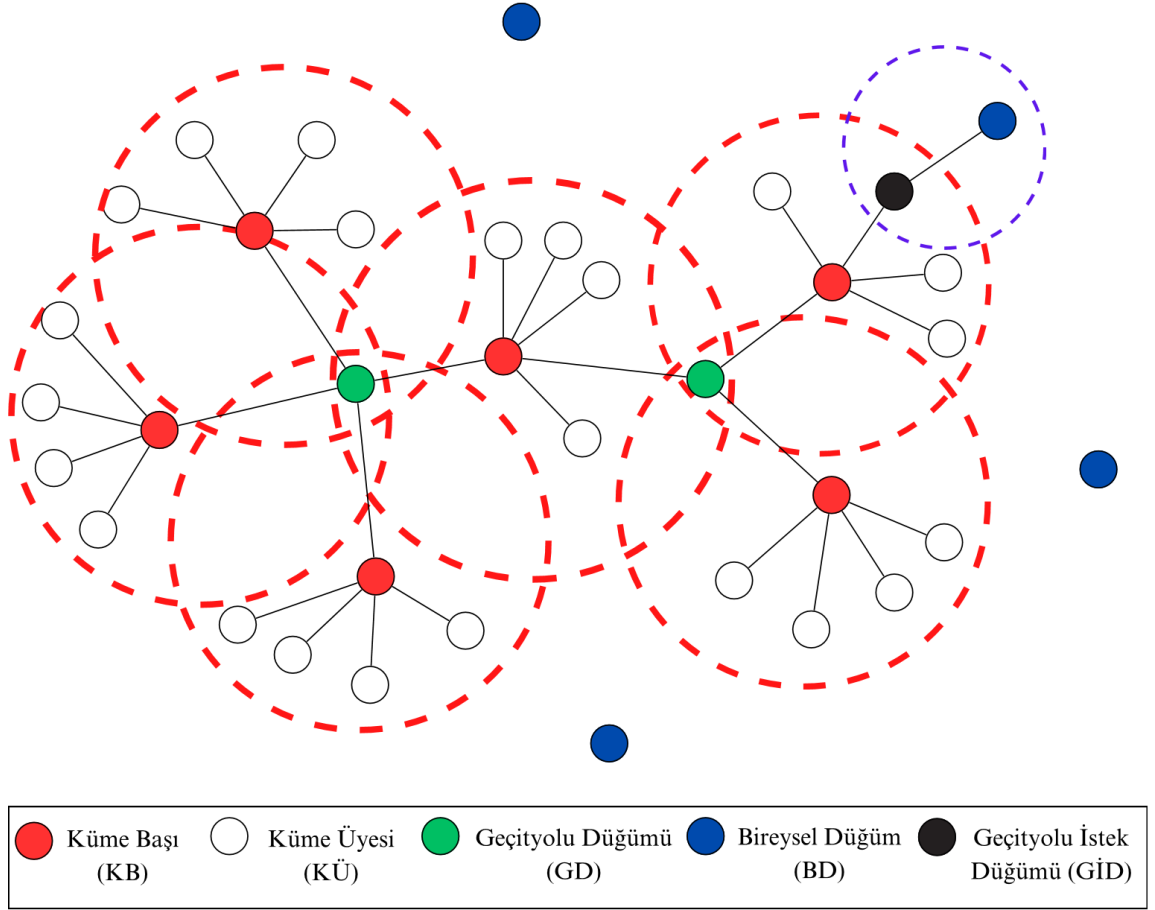
Şekil 3.7’de gösterildiği gibi, KB seçiminden sonra KB düğümü, *Cluster_info* paketinin *KB_info* bölümünü kendi düğüm adresiyle günceller ve güncellenmiş paketi iletim aralığındaki düğümlere gönderir.

KB seçimi yapıldıktan sonra, diğer düğümlere rol atama işlemi aşağıdaki gibi gerçekleştirilir.

- Üye Düğümlerin belirlenmesi: Eğer bir düğüm, *KB_info* bölümü işaretlenmiş yalnızca bir *Cluster_info* paketi aldıysa, *Cluster_info* paketindeki “*Üye_bilgisi*” bölümünü işaretler ve bunu KB’ye gönderir.
- GD’lerin belirlenmesi: Bir düğüm, *KB_info* bölümü işaretlenmiş birden fazla *Cluster_info* paketi aldıysa, bu düğüm kendini GD olarak işaretler ve bu bilgiyi diğer KB düğümleriyle paylaşır. GD tarafından gönderilen paketin *KB_info* bölümünde en az iki adres varsa, KB düğümü bu düğümü bir GD düğümü olarak tanımlar. Aynı KB adresine sahip birden fazla GD varsa, KB düğümü en yüksek enerjiye sahip olanı GD olarak işaretler ve diğerini KÜ düğümü olarak atar ve bilgileri ilgili düğümlere iletir. Eğer GD’ler arasında enerji seviyeleri eşitse, ortam erişim kontrolü (MAC) adresi büyük olan düğüm GD olarak seçilir. Diğer düğüm KÜ olarak işaretlenir ve ilgili düğümlere iletilir.
- Geçityolu istek düğümünün (GİD) belirlenmesi: Eğer bir düğüme hiç *KB_info* bilgisi içeren bir *Cluster_info* paketi almadıysa ancak iletim aralığında “hello” paketleriyle gelen komşu bilgileri varsa, bu komşuya bir *GD_istek bilgisi* işaretli bir *Cluster_info* paketi gönderir. Bu paketi alan komşu düğüm, kendini GİD olarak işaretler ve hem KB’ye hem de diğer düğüme gönderir.
- Adım 4: Eğer bir düğüme hiç *KB_info* işaretli *Cluster_info* paketi veya “hello” paket bilgisi gelmemiş ise düğüm *Cluster_info* paketindeki *Bireysel_düğüm bilgisi* kısmını işaretler ve BD olur.

Böylece kümeleme, ağın verimli bir şekilde yönetilmesini sağlamak için tüm düğümlere belirli roller atayarak gerçekleştirilir. Bu roller, düğümlerin konumu ve kapasitesine göre belirlenir ve böylece daha düzenli bir iletişim ve verimli veri iletimi sağlanmaktadır.

Önerilen yöntemden elde edilen kümeler ve düğüm rollerinin gösterimi Şekil 3.8’de gösterilmektedir.



Şekil 3.8. Önerilen yönteme göre kümeler ve düğüm rolleri örneği.

Şekil 3.8’de gösterilen ağ topolojisi, kümeleme ve düğüm rolü atamasının bir MANET’te nasıl etkili bir şekilde gerçekleştirilebileceğini göstermektedir. Kırmızı noktalar KB’leri ve beyaz noktalar Küme Üyesi (KÜ) düğümleri temsil etmektedir. Yeşil noktalar Geçityolu Düğümlerini (GD) temsil ederken, mavi noktalar Bireysel Düğümleri (BD) temsil etmektedir. Siyah noktalar bir BD’nin iletim aralığında olan ve yalnızca bu düğümler için ağ geçidi rolünü üstlenen Geçityolu İstek Düğümlerini (GİD) göstermektedir. Şekil 3.8’de gösterildiği üzere mavi noktalı çizgi ile çevrili BD’nin küme dışı iletişim için GİD üzerinden haberleşmeye katıldığını göstermektedir. Bu topolojik yapı, ağın enerji verimliliğini ve iletişim verimliliğini artırmak için geliştirilmiştir.

4. BULGULAR VE TARTIŞMA

Bu bölümde, simülasyon deneyleri ve elde edilen sonuçlar karşılaştırmalı olarak ele alınmaktadır. Simülasyon sonuçları üzerinden enerji verimliliği, ağ performansı, paket teslim oranı, düğüm hareketliliği, KB değişim oranı, uçtan uca gecikme ve veri iletim kapasitesi gibi performans kriterleri ile temel değerlendirmeler yapılmıştır.

4.1. SİMÜLASYON ORTAMI

Araştırmacılar ve sistem tasarımcıları, bir sistemin uygulanabilirliğini ve verimliliğini analiz etmek amacıyla, maliyetli fiziksel kurulumlar yerine simülasyon ortamlarından faydalanmaktadır. İletişim sistemlerinin performans analizi, gerçek bir sistem üzerinden gerçekleştirilebileceği gibi, matematiksel olarak modellenmiş bir eşdeğeri kullanılarak da incelenebilmektedir.

Kablosuz ağlar gibi karmaşık sistemlerin simülasyon süreçlerinde, kullanıcı davranışları, iletişim ortamının özellikleri ve sistem toleransları gibi birçok faktör başarıyı doğrudan etkilemektedir. Ancak, kablosuz algılayıcı ağlarda donanım ve yazılım açısından belirli bir standartlaşmanın olmaması, popüler simülasyon yazılımlarının bazı kablosuz ağ modelleri ve protokollerine sınırlı destek sunmasına neden olmaktadır. Bu nedenle, kablosuz ağlar için geliştirilen protokol, yöntem veya algoritmaların modellenmesi ve simülasyonlarının gerçekleştirilmesi oldukça zorlu bir süreçtir.

Son yıllarda yapılan performans değerlendirmeleri, NS-2, OMNeT++, THE ONE, OPNET, NetSim, NS-3 ve JiST gibi simülasyon araçlarının büyük ölçekli ağların modellenmesi ve simüle edilmesi konusunda başarılı sonuçlar verdiğini göstermektedir [101], [102].

Bu tez çalışmasında, açık kaynak kodlu ve esnek yapısı sayesinde mevcut protokollerin değiştirilmesine ve yeni protokollerin eklenmesine olanak tanıyan NS-2.35 ağ simülatörü kullanılmıştır. Bu simülatör, birçok farklı fonksiyon ve yönlendirme protokolünü bünyesinde barındırarak, simülasyon süreçlerinde geniş kapsamlı analizler yapılmasını sağlamaktadır.

NS-2 simülasyon aracı, kablolu ve kablosuz ağ arařtırmalarında kullanılan nesne tabanlı, ayrık olay simülatörüdür. İletim Kontrol Protokolü (TCP) ve kullanıcı Datagram Protokolü (UDP) gibi iletim protokollerinin yanı sıra, Dosya Transfer Protokolü (FTP), Telekomünikasyon Ağı Üzerinden Terminal Eriřim Protokolü (TELNET) ve Sabit Bit Oranı (CBR) gibi farklı trafik türlerini desteklemekte ve kablosuz ağlarda yönlendirme protokollerinin simülasyonunu gerçekleřtirebilmektedir. Ayrıca, çoklu yönlendirme protokollerini ve bazı MAC katmanı protokollerini de destekleyerek ağ performans analizleri için kapsamlı bir ortam sunmaktadır.

NS-2, C++ ve OTcl (Tcl betik dilinin nesneye dayalı genişletilmiş hali) dillerinde geliştirilmiştir. OTcl, simülasyonların komut ve yapılandırma işlemlerinde kullanılırken, C++ ise daha hızlı çalışan protokol uygulamaları için tercih edilmektedir. Simülatörün temel kullanımında OTcl bilgisi yeterli olurken, yeni protokol ve mekanizmaların geliştirilmesi için hem C++ hem de OTcl dillerine hakim olmak gerekmektedir.

Simülasyonlar için kullanılan sistem özellikleri Çizelge 4.1’de gösterilmektedir.

Çizelge 4.1. Kullanılan bilgisayarın sistem özellikleri.

İřletim sistemi	(Linux OS) Ubuntu 16.04 LTS 64bit
İřlemci	Intel® Core™ i5-4460 @3.20GHz (4CPU)
RAM	20GB
Grafik kartı	NVIDIA GeForce 8600 GT

Bu çalışmada önerilen yöntemin deęerlendirilmesi amacıyla, rastgele hareket paternine sahip mobil düęümler içeren heterojen bir MANET simülasyonu NS-2 simülasyon aracı versiyon sürümü 2.35’te gerçekleştirilmiştir. Simülasyon süresi 100 saniye olarak belirlenmiştir. Kullanılan simülasyon parametreleri Çizelge 4.2’de gösterilmektedir.

Çizelge 4.2. Simülasyonlarda kullanılan parametreler.

Parametreler	Değer
Düğüm sayısı	20– 100
Simülasyon alanı	1000 m × 1000 m
İletim aralığı	250 m
Simülasyon süresi	100 s
Hareketlilik modeli	Random-way point
Düğüm başlangıç enerjileri	100Joule
Paket boyutu	512 bytes
Yönlendirme protokolü	(Changed) AODV
Anten	Omni anten
Bağlantı katmanı sınıfı	LL
Kanal	Kablosuz
MAC versiyonu	IEEE 802.11
MAC Protokolü	CSMA/CA
Radyo yayılım modeli	İki ışınımlı zemin
Düğümün hareket hızları	2-10 mps

4.2. DÜĞÜMLERİN SAHİP OLMASI GEREKEN ÖZELLİKLER VE SİMÜLASYON KISITLARI

Bu tez kapsamında önerilen DMHAKA algoritmasının simülasyon ortamında doğru ve kararlı bir şekilde çalışabilmesi için ağdaki düğümlerin belirli donanımsal ve yazılımsal özelliklere sahip olduğu varsayılmıştır. Algoritmanın kümeleme ve yönlendirme süreçleri aşağıda belirtilen sistem gereksinimleri doğrultusunda yapılandırılmıştır:

- Düğümlerin konum bilgisine sahip olması gerekmektedir. Bu nedenle düğümlerin küresel konum belirleme sistemi (GPS) ile donatıldığı kabul edilmiştir.
- Düğümler arası bağlantıların çift yönlü (simetrik) olması sağlanmalıdır.
- Her düğümün en az 6 MB bellek kapasitesine sahip olduğu varsayılmıştır. Bu bellek, yönlendirme tablolarının tutulması ve kontrol paketlerinin işlenmesi için gereklidir.
- Düğümlerin en az 200 MHz işlemci kapasitesine sahip olduğu kabul edilmiştir. Bu işlem gücü, algoritmanın ağırlık hesaplamaları, EPS değeri hesaplaması ve

karar mekanizmalarının yürütülmesi için gereklidir.

- Her düğüm, küme kimliği ve düğüm rolüne ilişkin bilgileri içeren, yaklaşık 1 KB boyutunda Cluster_info adlı bir kontrol paketi barındırmaktadır. Bu paket, küme yapısının oluşturulması ve güncellenmesi amacıyla düğümler arasında iletilmektedir.

Belirtilen donanım ve kontrol yapıları, yalnızca simülasyon ortamında algoritmanın kararlı çalışabilmesi amacıyla tanımlanmıştır.

4.3. DMHAKA -WCA KARŞILAŞTIRMASININ PERFORMANS KRİTERLERİ

MANET'ler, dinamik yapıları nedeniyle ağ performansını etkileyen birçok faktöre sahiptir. Bu tür ağlarda yönlendirme, kümeleme ve kaynak yönetimi gibi süreçlerin etkinliğini ölçmek için belirli performans kriterleri kullanılmaktadır. Ağın enerji verimliliği, küme kararlılığı, uçtan uca gecikme süresi, paket teslim oranı ve ağ ömrü gibi parametreler, önerilen yöntemin başarımını değerlendirmek için kritik öneme sahiptir. Bu çalışmada, önerilen yöntemin performansı, literatürde yaygın olarak kullanılan aşağıdaki değerlendirme metrikleri doğrultusunda analiz edilmiştir.

- Ortalama kalan enerji ve ağ ömrü: Düğümlerin bilgi gönderme, alma veya iletme gibi farklı ağ görevlerini gerçekleştirmek için başlangıçtaki enerjilerini harcarlar. Bu görevler sırasında düğümün pilindeki enerjinin azalmasına ise enerji tüketimi denir. Düğümün düşük enerji tüketimi, ağın kararlılığını artırır ve daha uzun bir ağ ömrü sağlar. Dolayısıyla ağ ömrü ile düğümlerin kalan enerjileri arasında doğru orantılı bir ilişki vardır [103]. Ağın yüzdelik olarak kalan enerjisi Denklem (4.1) ve Denklem (4.2)'de gösterildiği gibi hesaplanmaktadır.

$$\text{Kalan enerji} = \text{Toplam ağ enerjisi} - \text{Toplam tüketilen enerji} \quad (4.1)$$

$$\text{Ortalama kalan enerji} = \frac{\text{Kalan enerji}}{\text{Toplam ağ enerjisi}} \times 100 \quad (4.2)$$

Daha yüksek kalan enerji, düğümlerin daha az enerji tükettiğini ve ağ ömrünün daha uzun olacağını göstermektedir. Eğer ortalama kalan enerji düşükse, bu ağın hızlı bir şekilde tükendiği ve düğümlerin ömrünün kısa olduğu anlamına gelmektedir.

- KB değişim oranı (KBDO): KB'lerin belirli bir süre içinde ne sıklıkla değiştiğini

gösteren bir metriktir. Ağ kararlılığını ve kümeleme mekanizmasının etkinliğini değerlendirmek için kullanılmaktadır. Matematiksel gösterimi Denklem (4.3)'te gösterildiği gibidir [104].

$$KBDO = \frac{10 \text{ saniyede değişen KB sayısı}}{\Delta t} \quad (4.3)$$

Burada Δt zaman aralığını göstermektedir. Bu çalışmada 10 sn olarak belirlenmiştir.

Daha düşük KB değişim oranı, küme yapısının daha kararlı olduğunu ve düğümlerin daha uzun süre KB olarak hizmet ettiğini gösterir. Yüksek KB değişim oranı ise, ağda fazla kümeleme işlemi yapıldığını ve düğümlerin sürekli yeniden düzenlendiğini gösterir. Bu durum ağın ek yükünü artırarak enerji tüketimini olumsuz etkilemektedir. Dolayısıyla ağ kararlılığını düşürmektedir.

- Ortalama KB ömrü: Bir KB'nin simülasyon süresi boyunca ne kadar süre aktif kaldığını gösterir [105]. KB'lerin ömrü arttıkça küme kararlılığı da artmaktadır. Formülü Denklem (4.4)' te gösterilmektedir.

$$\text{Ortalama KB ömrü} = \frac{\sum_{i=1}^N TKB_i}{N} \quad (4.4)$$

Burada TKB_i , i 'ninci KB'nin toplam aktif süresi ve N simülasyon süresince görev yapan toplam KB sayısıdır.

- Ortalama küme ömrü: Bir kümenin ilk oluşturulduğu andan yapısal değişikliğe (KB değişimi veya üyelik bozulması) uğradığı ana kadar geçen süredir [106]. Matematiksel gösterimi Denklem (4.5)'de gösterildiği gibidir.

$$\text{Ortalama Küme ömrü} = \frac{\sum_{i=1}^M TK_i}{M} \quad (4.5)$$

Burada TK_i , i 'ninci kümenin ömrüdür. Yani kaç saniye boyunca aynı KB ve üyelerle aktif kaldığıdır. M ise toplam küme sayısını göstermektedir.

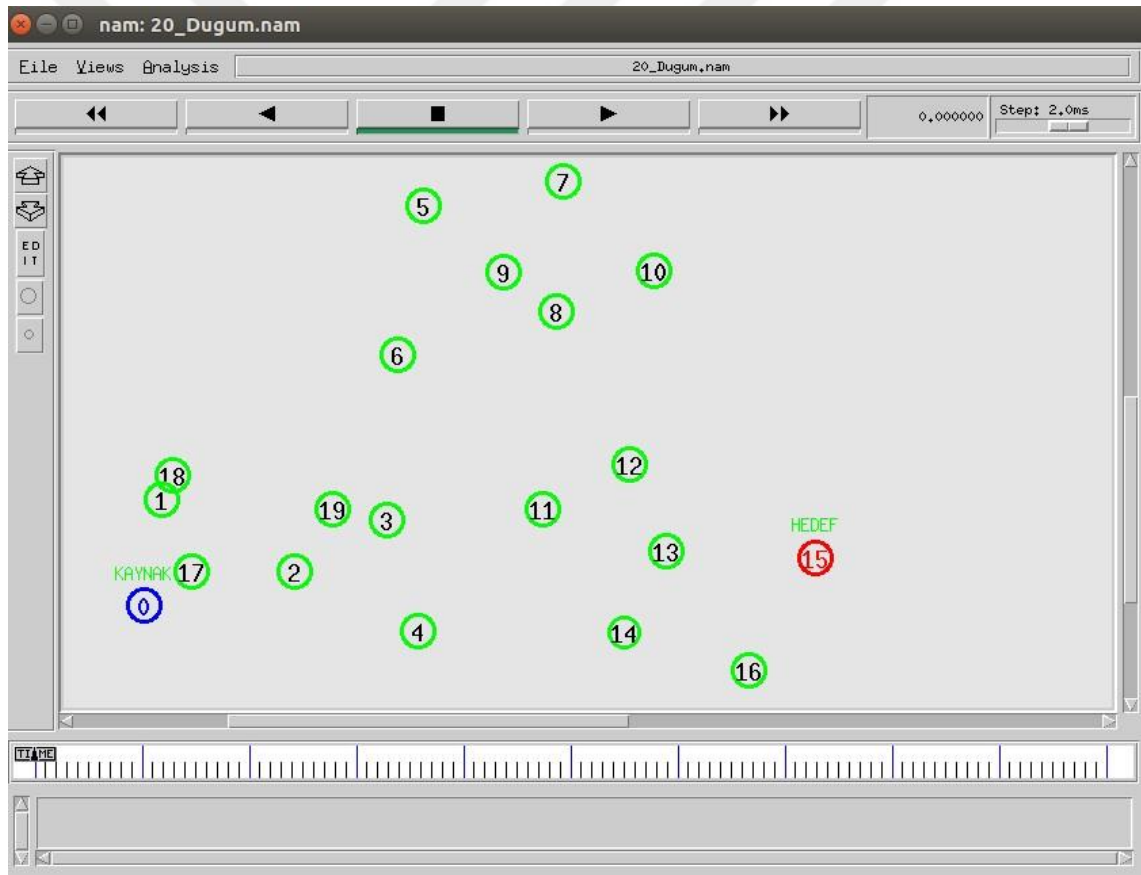
4.3.1. DMHAKA Yönteminin WCA ile Karşılaştırmalı Performans Analizi

KB seçimi, MANET'lerde kümeleme süreçlerinin temel unsurlarından biri olup ağın enerji verimliliği, kararlılığı ve ömrü üzerinde doğrudan etkiye sahiptir. KB seçiminde

kullanılan yöntemlerin etkinliđi, düđüm sayısı arttıkça daha belirgin hale gelmekte ve ađ ölçeklenebilirliđi açısından kritik bir rol oynamaktadır. Bu bölümde, önerilen DMHAKA yöntemi ile klasik WCA yönteminin farklı düđüm sayılarında (20, 40, 60, 80 ve 100 düđüm) KB seçimi performansları karşılaştırılmaktadır. Performans deđerlendirmesi; seçilen KB sayısı, KB ömrü, KB deđişim oranı ve ađ üzerindeki enerji tüketimi gibi metrikler kullanılarak yapılmaktadır. Her bir simülasyon senaryosu için elde edilen sonuçlar analiz edilerek, önerilen yöntemin klasik WCA'ya kıyasla sağladığı iyileştirmeler ortaya konulmaktadır.

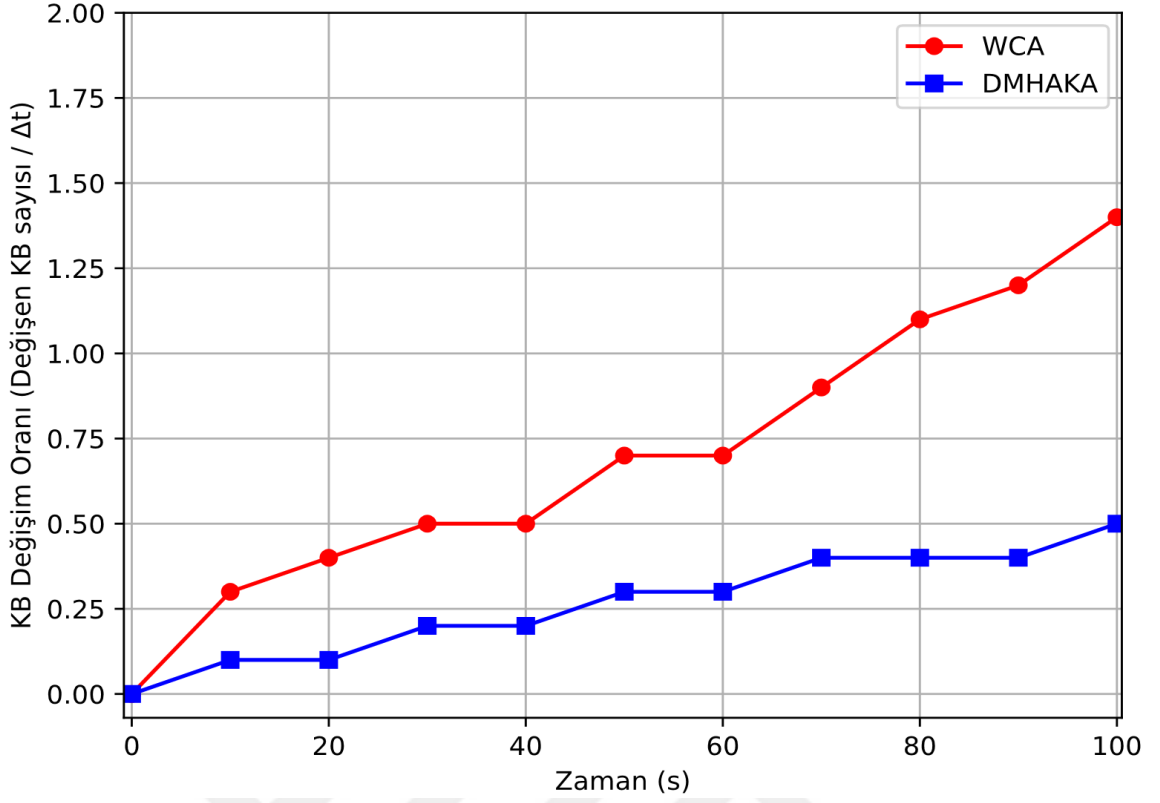
4.3.1.1. 20 düđümlü simülasyon çalışmasının performans karşılaştırması

Şekil 4.1 'de gösterildiđi gibi 20 düđümden oluşan bir simülasyon ortamı hazırlanmıştır. Düđümler rastgele konumlandırılmış ve Çizelge 4.2'deki parametreler kullanılmıştır.



Şekil 4.1. 20 düđümlü simülasyon gösterimi.

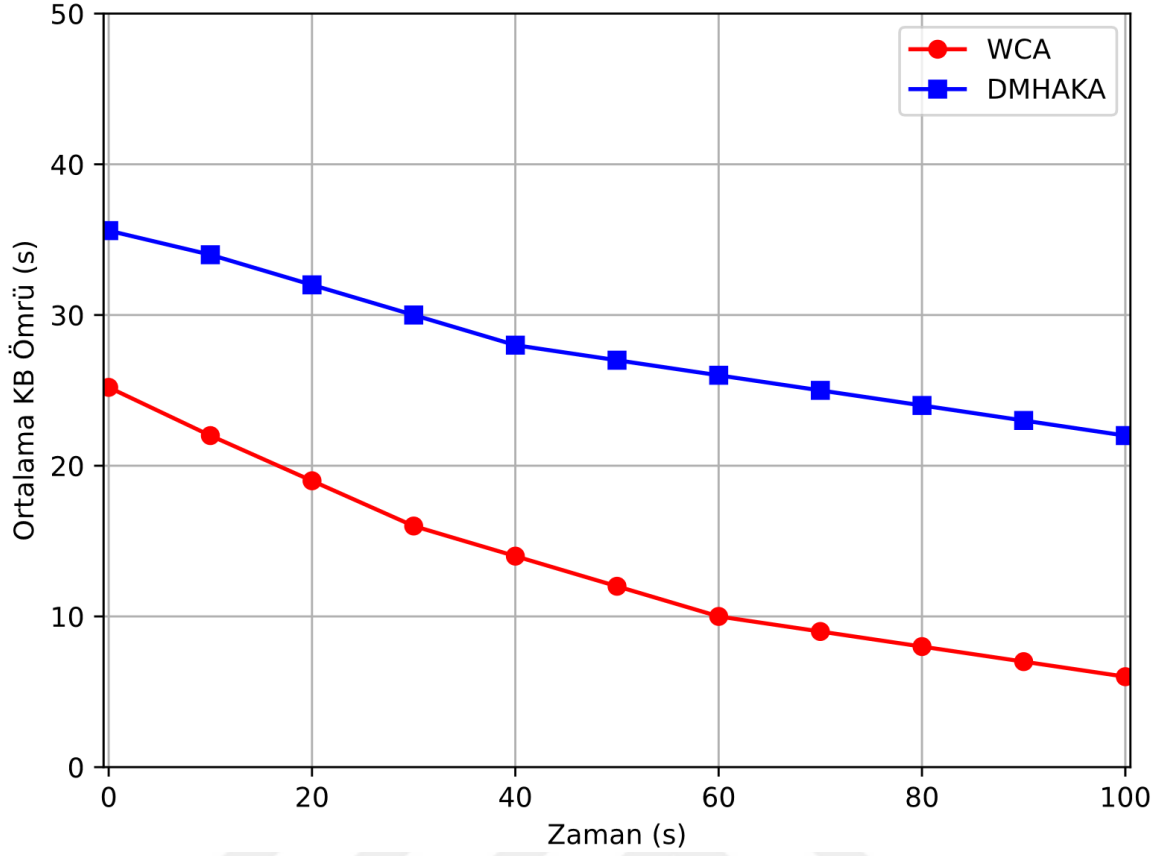
Şekil 4.1'e göre kaynak düđüm 0'dan hedef düđüm 15'e paket gönderilmektedir. 100 sn'lik periyotta Şekil 4.2'de iki yöntemin zamana bađlı KB deđişim oranı gösterilmektedir.



Şekil 4.2. 20 düğümlü senaryoda KB değişim oranı.

Şekil 4.2'ye göre simülasyonun başlangıcında, WCA yönteminde 5 KB, DMHAKA yönteminde ise 7 KB oluşmaktadır. Zaman ilerledikçe, WCA'da KB değişim oranı sürekli artarken, DMHAKA yöntemi KB değişim oranını daha düşük seviyelerde tutmaktadır. İlk 10 saniyede, WCA'da 3 KB değişmiş, yani başlangıçta oluşturulan KB'lerin bir kısmı enerji veya bağlantı durumu nedeniyle değiştirilmiştir. DMHAKA'da ise sadece 1 KB değişmiştir. Buna bağlı olarak WCA'nın KB değişim oranı 0,3, DMHAKA'nın ise 0,1 olarak hesaplanmıştır. 10- 50 saniye aralığında, WCA'da 3 yeni KB eklenmiş ve 6 KB değişmiş, DMHAKA'da ise 3 yeni KB eklenmiş ve sadece 2 KB değişmiştir. Bu dönemde WCA'nın KB değişim oranı 0,6'ya yükselirken, DMHAKA'nın KB değişim oranı 0,2 civarında sabit kalmıştır. 50-100 saniye aralığında, WCA'da toplam 9 KB değişmiş, DMHAKA'da ise 4 KB değişmiştir. Bu süreçte WCA'nın KB değişim oranı 1,4 seviyesine çıkarken, DMHAKA 0,6 oranında kalmıştır. WCA'nın değişim oranındaki bu hızlı artış, KB'lerin stabil olmamasından ve sürekli değişime uğramasından kaynaklanmaktadır. DMHAKA ise daha stabil KB seçim mekanizmasına sahip olduğu için değişim oranını düşük seviyelerde tutmaktadır. Dolayısıyla DMHAKA'nın küme kararlılığı yüksek olduğu görülmektedir.

20 düğümlü senaryo için ortalama KB ömrü Şekil 4.3'te gösterilmektedir.



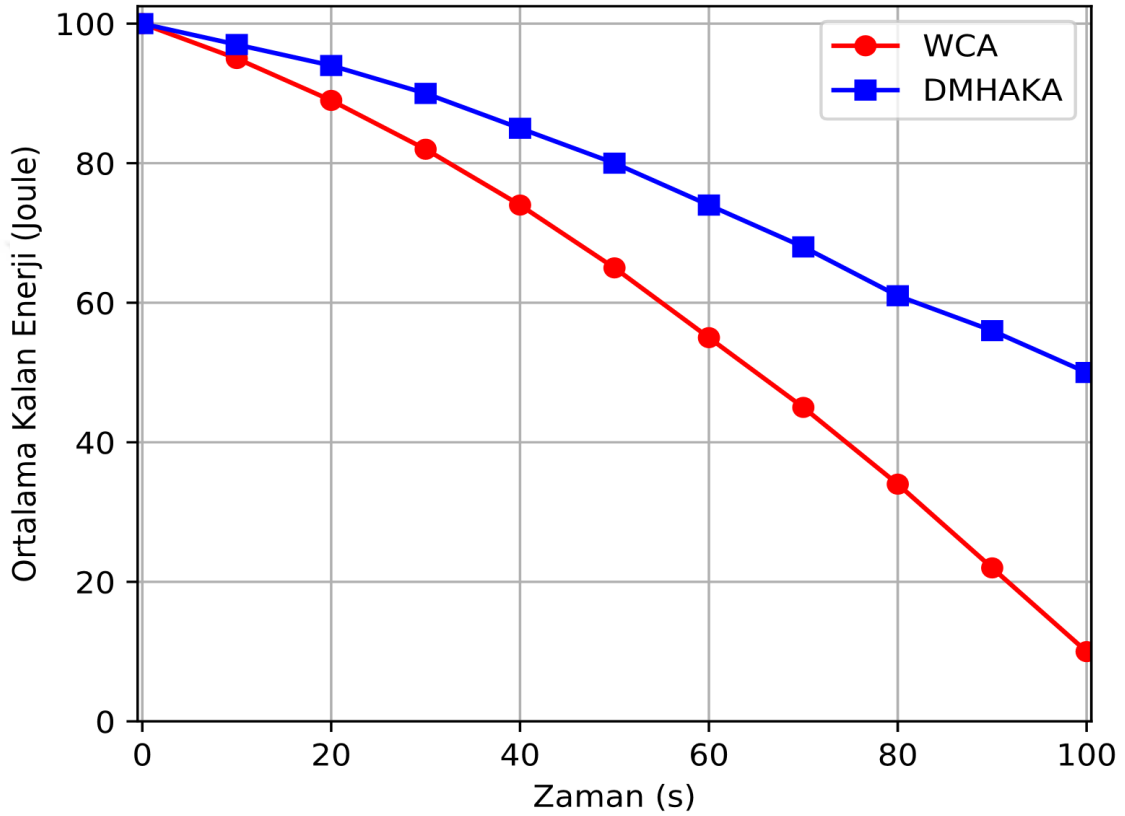
Şekil 4.3. 20 düğümlü senaryo için ortalama KB ömrü.

Yukarıdaki şekle göre ağ ilk kurulumunda WCA’da KB’lerin ortalama ömrü yaklaşık 25 saniye, DMHAKA’da ise yaklaşık 35 saniye olarak hesaplanmıştır. Bu durum, DMHAKA yönteminin daha stabil KB seçim mekanizmasına sahip olduğunu ve KB’lerin daha uzun süre hayatta kaldığını göstermektedir. İlk 20 saniyede ise WCA yönteminde KB ömrü hızla 18 saniyeye kadar düşerken, DMHAKA yönteminde bu düşüş daha azdır ve 30 saniye civarındadır. WCA’nın KB ömründeki bu hızlı azalma, KB’lerin daha sık değişmesine neden olmaktadır. 20-50 saniye arasında WCA’da KB’lerin ömrü düşmeye devam ederek 10-12 saniyeye kadar gerilemektedir. Buna karşılık DMHAKA’da KB ömrü 27-28 saniye civarında korunmaktadır. Bu fark, DMHAKA’nın daha kararlı kümeler oluşturduğunu göstermektedir. 50-100 saniye arasında ise WCA yönteminde KB ömrü 5-7 saniye civarına kadar düşmüştür. Bu süreçte, düğümlerin enerji seviyeleri de düştüğünden KB’ler çok sık değişmekte dolayısıyla kümeleme yapısının kararlılığı önemli ölçüde bozulmaktadır. DMHAKA’da ise KB ömrü hala 22-23 saniye seviyelerinde kalmakta, yani daha uzun süre stabil KB’ler ile ağ yönetilmektedir.

20 düğümlü ağda ortalama KB ömrünün zamana bağlı grafiğine göre WCA yönteminde KB’ler kısa ömürlü olup sık sık değişmektedir, bu da kümeleme yapısının stabil

olmadığını göstermektedir. DMHAKA yöntemi ise KB seçiminde daha stabil bir mekanizma sunduğundan, KB'lerin ömrü daha uzun olmaktadır. Daha uzun KB ömrü, ağda daha az KB değişimi olmasını sağladığı için enerji verimliliği artmakta ve kümeleme kararlılığı korunmaktadır.

Bu durum Şekil 4.4'te ortalama kalan enerjinin zamana bağlı grafiğinde de açıkça görülmektedir.



Şekil 4.4. 20 düğümlü senaryoda ortalama kalan enerjinin zamana göre değişimi.

Bu grafiğe göre, simülasyon başlangıcında her iki yöntemde de düğümlerin enerji seviyeleri 100 Joule olarak başlatılmıştır. İlk 50 saniyede, genelde WCA yönteminde KB değişim oranının yüksek olması, KB'lerin daha uzak düğümlerle haberleşmesi ve daha fazla düğüme hizmet vermesi nedeniyle enerji tüketimi hızla artmaktadır. Buna karşılık, DMHAKA yöntemi optimizasyon mekanizmaları sayesinde daha stabil KB seçimleri yaparak enerji tüketimini dengeli bir şekilde yönetmektedir.

50'inci saniyede, WCA'daki ortalama düğüm enerjisi yaklaşık 65 Joule'e kadar düşerken, DMHAKA'da bu değer yaklaşık 80 Joule seviyesinde kalmaktadır. Bu fark, DMHAKA'nın daha verimli KB seçim ve kümeleme mekanizması ile gereksiz enerji tüketimini azalttığını göstermektedir.

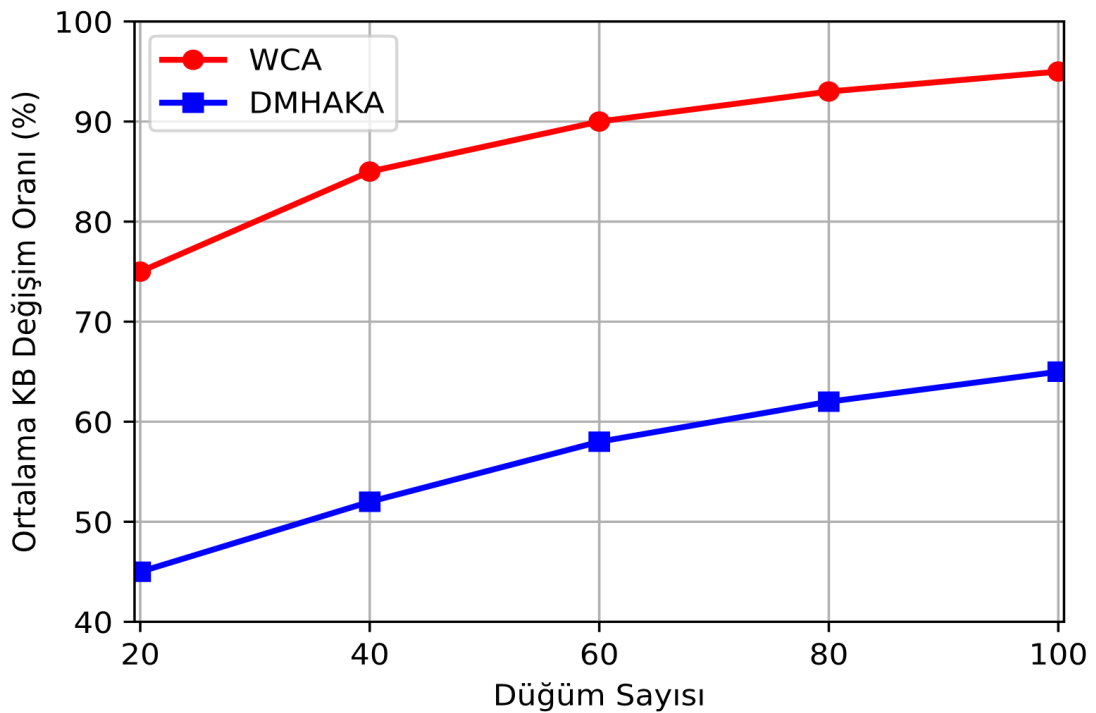
Simülasyon süresinin sonuna doğru, WCA'daki ortalama enerji seviyesi 40J'nin altına inerken, DMHAKA ile bu değer yaklaşık 55J seviyesinde korunmaktadır. 100. saniyede ise WCA'da ortalama kalan enerji 20J civarına düşerken, DMHAKA ile bu değer yaklaşık 45J seviyesinde kalmıştır. Bu sonuçlar, DMHAKA'nın ağ ömrünü önemli ölçüde uzattığını ve enerji tüketimini optimize ettiğini göstermektedir.

4.3.1.2. Farklı düğüm sayılarına göre performans karşılaştırması

Bu bölümde, önerilen DMHAKA yöntemi ile klasik WCA yönteminin farklı düğüm sayılarındaki (20, 40, 60, 80, 100 düğüm) performansları karşılaştırmalı olarak analiz edilmiştir. Her bir performans metriği için ayrı ayrı grafiksel analiz sunulmuş ve iki yöntemin başarımları değerlendirilmiştir.

Bu çalışmada, ortalama KB değişim oranı yüzde (%) cinsinden hesaplanmıştır. Her simülasyon senaryosunda değişen toplam KB sayısı, simülasyon süresine bölünmüş ve sonuç 100 ile çarpılarak yüzde formatına dönüştürülmüştür. Böylece ağdaki KB stabilitesi daha sezgisel bir biçimde karşılaştırılabilir hale getirilmiştir. Değerin düşük olması KB seçimlerinin daha kararlı olduğunu ve ağ üzerindeki yükün daha az olduğunu göstermektedir.

Şekil 4.5'te düğüm sayılarına göre ortalama KB değişim oranı grafiği gösterilmektedir.



Şekil 4.5. Farklı düğüm sayılarına göre ortalama KB değişim oranı.

Şekil 4.5'e göre, düğüm sayısı arttıkça her iki yöntemde de KB değişim oranı artmaktadır. Ancak DMHAKA yöntemi, her senaryoda WCA'ya göre daha düşük değişim oranı sergileyerek kümeleme kararlılığı açısından daha başarılı bir sonuç ortaya koymaktadır.

Örneğin, 20 düğüm için:

WCA yönteminde KB değişim oranı %75 iken,

DMHAKA yönteminde bu oran sadece %45 olarak gerçekleşmiştir.

Düğüm sayısı 40'a çıkarıldığında:

WCA'nın değişim oranı %85,

DMHAKA'nın ise %52 seviyesindedir.

100 düğüm gibi yoğun bir ağda ise fark daha da belirginleşmektedir:

WCA'nın oranı %95'e ulaşırken,

DMHAKA yalnızca %65 seviyesine yükselmektedir.

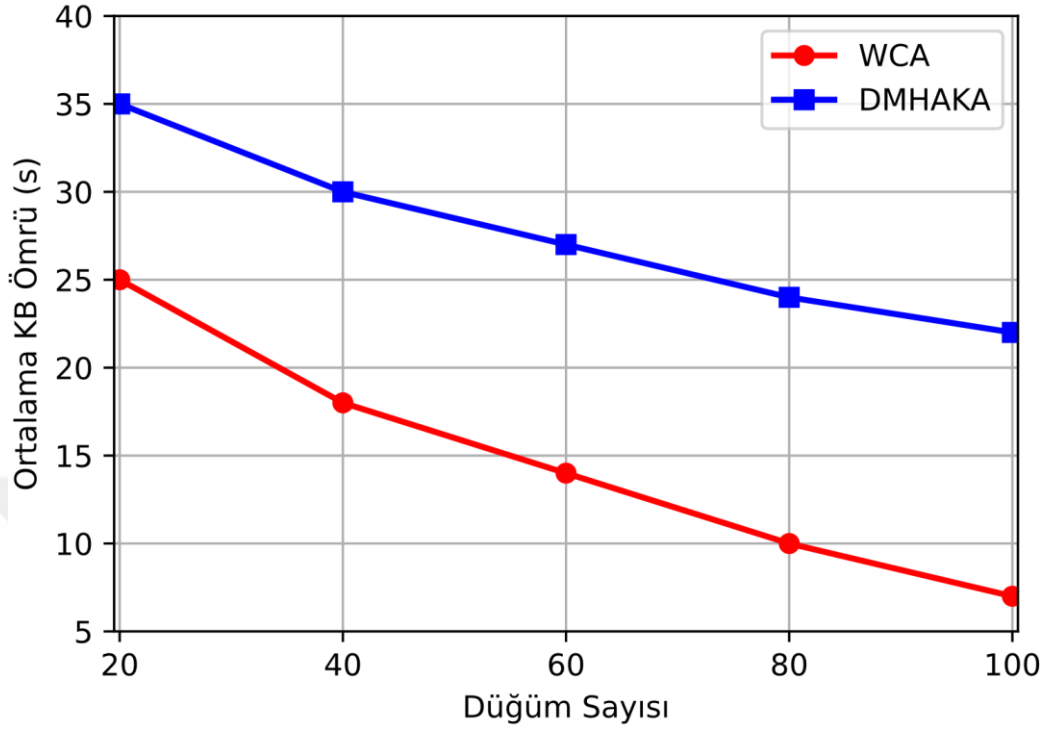
Bu sayısal bulgular, DMHAKA yönteminin artan düğüm sayısına rağmen KB değişim oranlarını kontrollü biçimde yönetebildiğini ve bu sayede daha kararlı kümeler oluşturabildiğini ortaya koymaktadır. Özellikle WCA yönteminde gözlemlenen yüksek KB değişim oranları, ağda sık sık yeniden kümeleme işlemleri yapılmasına neden olmaktadır. Bu durum ise hem işlem yükünü artırmakta hem de düğümlerin enerji tüketimini olumsuz yönde etkilemektedir.

DMHAKA yöntemi, daha istikrarlı ve dengeli bir KB seçim mekanizmasına sahip olduğundan, yeniden yapılanma sıklığını azaltmaktadır. Böylece ağda enerji verimliliği korunmakta ve performans sürdürülebilirliği sağlanmaktadır. Bu durum yalnızca enerji tasarrufuna değil, aynı zamanda paket iletim süresi, gecikme süresi ve genel ağ performansı üzerinde de doğrudan olumlu etkiler yaratmaktadır.

Dolayısıyla elde edilen veriler, DMHAKA yönteminin yalnızca küçük ölçekli MANET yapılarında değil, aynı zamanda yüksek düğüm yoğunluğuna sahip, daha karmaşık ağ senaryolarında da ölçeklenebilir ve sürdürülebilir bir kümeleme performansı sunduğunu açıkça göstermektedir.

Bu bağlamda incelenen bir diğer önemli performans kriteri ise, KB'lerin ağ üzerinde ne kadar süreyle görevde kaldığını gösteren ortalama KB ömrüdür. Bu metrik, doğrudan kümeleme yapısının kararlılığı hakkında bilgi vermektedir. Her bir simülasyon senaryosu

için, seçilen tüm KB'lerin görev süreleri ortalama olarak hesaplanmış ve sonuçlar Şekil 4.6'da sunulmuştur.



Şekil 4.6. Farklı düğüm sayılarına göre ortalama KB ömrü.

Yukarıdaki grafik sonuçlarına göre, düğüm sayısı arttıkça hem WCA hem de DMHAKA yöntemlerinde ortalama KB ömrü azalmaktadır. Bu durum, ağ yoğunluğu arttıkça KB'lerin daha kısa sürelerle görevde kalmasından ve daha sık değişmesinden kaynaklanmaktadır. Ancak, DMHAKA yöntemi her düğüm sayısında WCA'ya göre daha uzun ömürlü KB'ler üretmeyi başarmıştır.

Örneğin;

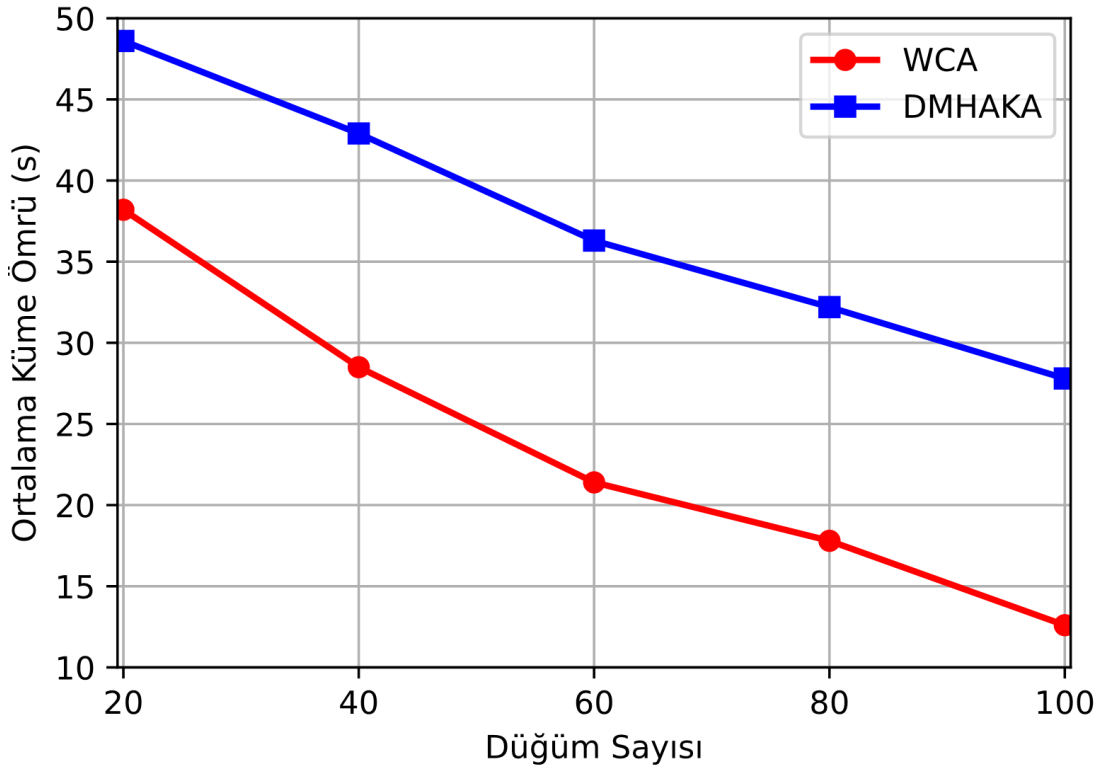
- 20 düğümde WCA ile elde edilen ortalama KB ömrü yaklaşık 25 saniye iken, DMHAKA ile bu süre yaklaşık 35 saniye olarak ölçülmüştür.
- 100 düğümde ise WCA'nın KB ömrü yaklaşık 7 saniyeye, DMHAKA'nın ise yaklaşık 22 saniyeye kadar düşmektedir.
- Benzer şekilde, 60 düğüm için WCA değeri yaklaşık 14 saniye iken DMHAKA yaklaşık 27 saniye ömür sağlamıştır.

Bu veriler, DMHAKA'nın sadece düşük düğüm yoğunluklarında değil, aynı zamanda yoğun MANET senaryolarında da daha kararlı bir KB yapısı sunduğunu ve yeniden

kümeleme sıklığını azaltarak hem enerji verimliliği hem de ağ kararlılığı açısından önemli avantajlar sağladığını göstermektedir.

Geliştirilen DMHAKA yönteminin KB değişim oranını düşürerek küme kararlılığını artırdığı, KB ömrünü uzattığı ve enerji tüketimini optimize ederek genel verimliliği yükselttiği bu çalışmada ortaya konulmuştur. Bu olumlu etkilerin bir diğer yansıması da ağda oluşturulan kümelerin ne kadar süreyle stabil kaldığını gösteren küme ömrü metriğinde gözlemlenmektedir. Her bir simülasyon senaryosunda, küme yapısının bozulmadan sürdürüldüğü ortalama süre hesaplanarak, farklı düğüm sayılarındaki değişim eğilimi Şekil 4.7’de gösterilmektedir.

Her ne kadar küme ömrü metriği, ortalama KB ömrü ile benzer kavramsal çerçeveye sahip gibi görünse de aralarında temel bir fark bulunmaktadır. Ortalama KB ömrü yalnızca KB’nin görev süresini ifade ederken; küme ömrü hem KB’nin hem de bağlı üyelerin yapısal bütünlüğünü dikkate almaktadır. Bu yönüyle küme ömrü, ağdaki genel kararlılığın daha geniş ve bütüncül bir göstergesi olarak değerlendirilmelidir.



Şekil 4.7. Farklı düğüm sayılarına göre ortalama küme ömrü.

Şekil 4.7’de sunulan verilere göre, düğüm sayısı arttıkça her iki yöntemde de ortalama küme ömrü azalma göstermektedir. Bu durum, ağda düğüm yoğunluğu arttıkça küme yapılarının daha sık değişmesiyle açıklanabilir. Ancak geliştirilen DMHAKA yöntemi,

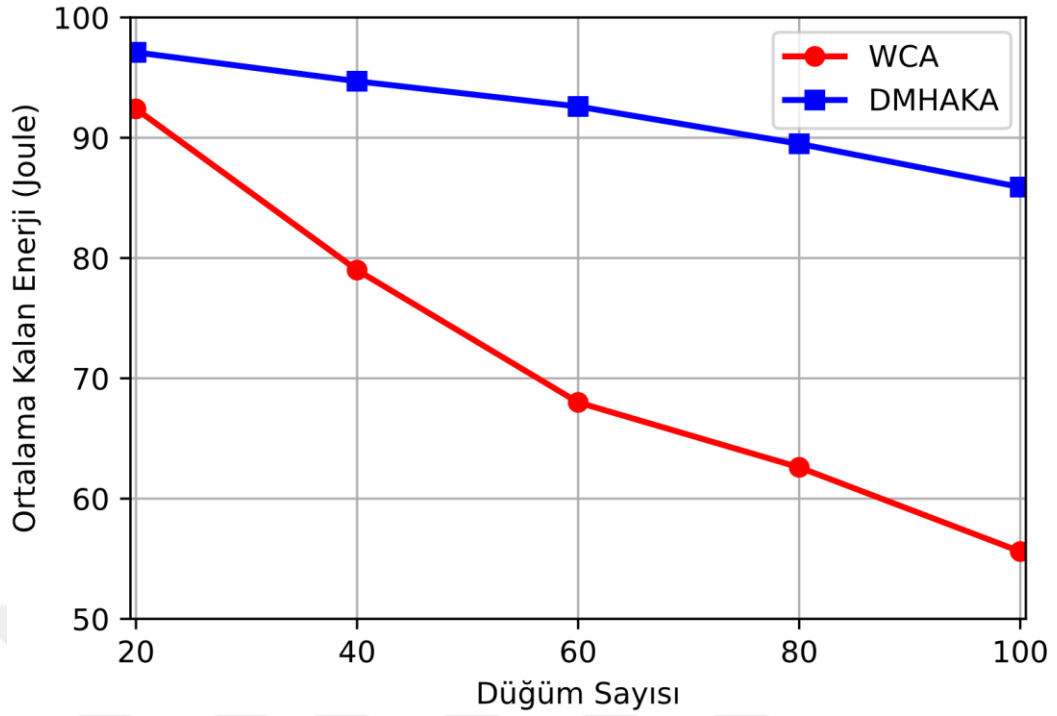
her bir düğüm sayısında WCA'ya göre daha uzun süreli kümeler oluşturarak kararlılık açısından önemli bir avantaj sağlamıştır.

Örneğin:

- 20 düğüm senaryosunda, WCA ile elde edilen ortalama küme ömrü yaklaşık 38 saniye iken, DMHAKA ile bu süre yaklaşık 48 saniye olarak hesaplanmıştır.
- 60 düğüm için WCA'da küme ömrü yaklaşık 21 saniye, DMHAKA'da ise yaklaşık 36 saniyeye kadar yükselmektedir.
- 100 düğüm gibi yoğun ağ yapılarında ise WCA ile kümeler yaklaşık 12 saniye, DMHAKA ile ise yaklaşık 27 saniye boyunca stabil kalabilmiştir.

Bu sonuçlar, DMHAKA'nın sadece bireysel KB kararlılığını değil, aynı zamanda bütünsel küme yapısını da daha uzun süre koruyabildiğini göstermektedir. Uzun ömürlü kümeler, ağda yeniden yapılanma ihtiyacını azaltmakta, kontrol trafiğini düşürmekte ve bu sayede enerji tüketimini dengelemektedir. Aynı zamanda paket yönlendirme sürekliliği açısından da fayda sağlayarak, genel ağ performansına doğrudan olumlu katkıda bulunmaktadır.

Geliştirilen yöntemin, farklı düğüm sayılarına sahip ağlardaki enerji verimliliği üzerindeki etkisini değerlendirmek amacıyla ortalama kalan enerji analizi gerçekleştirilmiştir. Bu analizde, her bir senaryo için simülasyon süresi sonunda düğümlerde kalan ortalama enerji miktarı hesaplanmış ve elde edilen karşılaştırmalı sonuçlar Şekil 4.8'de sunulmuştur.



Şekil 4.8. Farklı düğüm sayılarına göre ortalama kalan enerji.

Şekil 4.8’de görüldüğü üzere düğüm sayısı arttıkça her iki yöntemde de ortalama kalan enerji miktarında azalma gözlenmektedir. Bununla birlikte, geliştirilen DMHAKA yöntemi, tüm düğüm sayılarında WCA’ya kıyasla daha yüksek seviyede kalan enerji değerleri üretmiş ve enerji verimliliği açısından üstün bir performans sergilemiştir.

Örneğin:

- 20 düğüm için simülasyon sonunda WCA yöntemiyle ortalama kalan enerji yaklaşık 93 Joule iken, DMHAKA ile bu değer yaklaşık 96 Joule olarak ölçülmüştür.
- 60 düğüm durumunda WCA’da kalan enerji yaklaşık 68 Joule, DMHAKA’da ise yaklaşık 92 Joule’ dur.
- 100 düğüm gibi yoğun bir ağda WCA’nın ortalama kalan enerjisi yaklaşık 55 Joule civarlarına kadar gerilerken, DMHAKA bu değeri yaklaşık 86 Joule düzeyinde tutabilmiştir.

Bu farklar, DMHAKA’nın enerji açısından daha dengeli bir kümeleme yapısı sunduğunu ve KB seçimlerinin daha uzun ömürlü, daha az değişen ve daha verimli şekilde gerçekleştiğini göstermektedir. WCA yönteminde sık sık değişen KB’ler ve yeniden yapılanma süreçleri, düğümlerin iletişim yükünü artırarak enerji tüketimini

hızlandırmakta; bu da kalan enerji miktarını doğrudan olumsuz etkilemektedir.

DMHAKA yöntemi ise, daha kararlı KB yapısı sayesinde düğümler arası iletişim maliyetlerini azaltmakta, yeniden kümeleme işlemlerini en aza indirmekte ve böylece genel enerji tüketimini optimize etmektedir. Özellikle yüksek düğüm sayılı senaryolarda bu fark daha da belirginleşmekte ve DMHAKA'nın ölçeklenebilir, enerji açısından sürdürülebilir bir kümeleme çözümü sunduğu anlaşılmaktadır.

4.3.2. DMHAKA Yönteminin Güncel Kümeleme Algoritmaları ile Karşılaştırmalı Performans Analizi

Bu bölümde, geliştirilen DMHAKA yöntemi, literatürde sıkça referans verilen ve enerji verimliliği, gecikme süresi, yönlendirme başarımı gibi yönleriyle öne çıkan dört farklı güncel kümeleme algoritması ile karşılaştırmalı olarak analiz edilmiştir. Karşılaştırma yapılan yaklaşımlar sırasıyla EE-WCA [22], TSDR [107], E-MAVMMF [36] ve MORS-ASO [108] algoritmalarıdır. Bu algoritmalar ile DMHAKA'nın performansı; veri iletim kapasitesi, paket teslim oranı, ortalama kalan enerji ve ortalama uçtan uca gecikme kriterleri üzerinden değerlendirilmiştir. Bu karşılaştırmalı analizlerde, DMHAKA algoritması simülasyon ortamında değerlendirilmiş; EE-WCA, E-MAVMMF, TSDR ve MORS-ASO gibi algoritmalara ait performans verileri ise ilgili çalışmaların grafiklerinden elde edilen nicel değerler esas alınarak karşılaştırmaya dahil edilmiştir. Bu veriler, literatürde sunulan grafikler üzerinden yaklaşık değerlerle alınmış ve aynı metrikler üzerinden kıyaslama yapılmıştır.

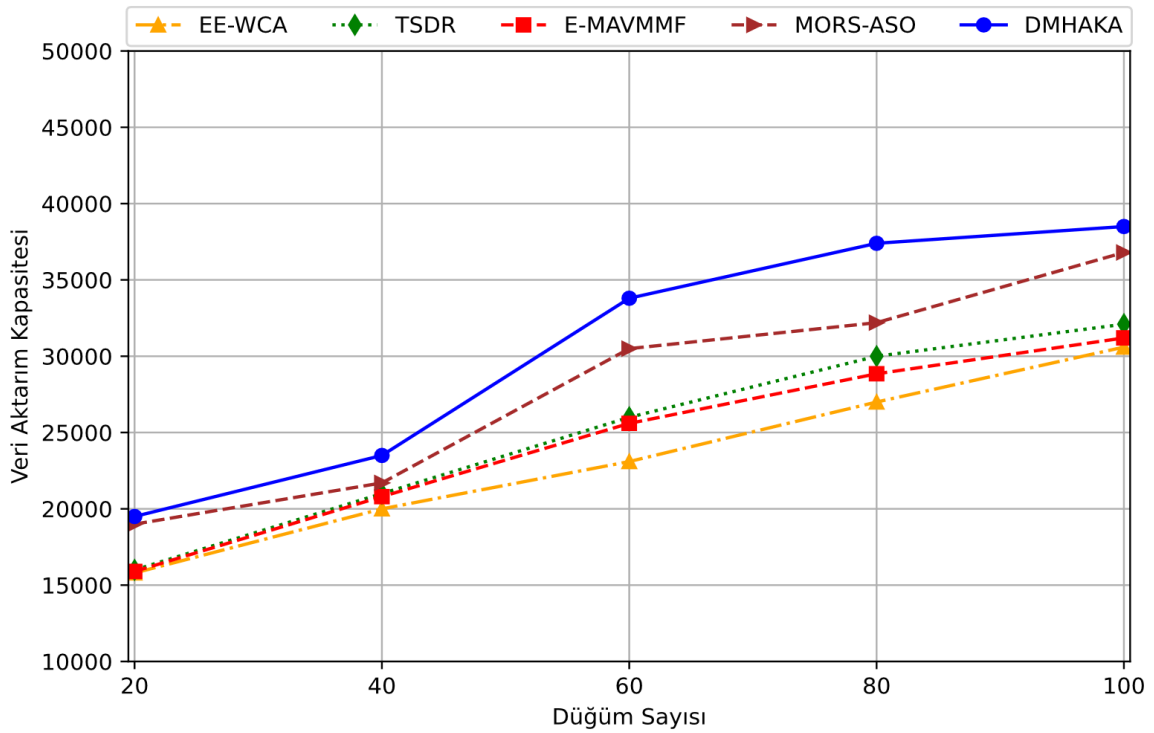
Söz konusu performans kriterlerinin tanımları, hesaplama yöntemleri ve kullanılan formüller aşağıdaki alt başlıklarda detaylı biçimde açıklanmıştır. Bu analizde amaç, önerilen yöntemin yalnızca geleneksel algoritmalara karşı değil, aynı zamanda modern, enerji ve gecikme odaklı geliştirilmiş yöntemlere karşı da nasıl bir performans sergilediğini ortaya koymaktır.

4.3.2.1. Veri aktarım kapasitesine (Throughput) göre performans analizi

Veri aktarım kapasitesi, birim zamanda başarıyla iletilen paketlerin sayısıdır [36]. Dolayısıyla ağ çıkışı bir veri aktarım hızıdır. Saniye başına bit (bps) cinsinden ölçülür. Bu metrik ağın veri iletim performansını değerlendirmek için kullanılmaktadır. Veri aktarım kapasitesi hesaplama formülü Denklem (4.6)'de gösterildiği gibidir.

$$\text{Veri aktarım kapasitesi} = \frac{\text{Toplam iletilen paket sayısı}}{\text{Simülasyon süresi}} \quad (4.6)$$

Daha yüksek veri aktarım kapasitesi, ağın daha fazla veriyi daha kısa sürede iletebildiğini göstermektedir. Düşük veri aktarım kapasitesi, ağ tıkanıklığı, paket kaybı, düşük bant genişliği veya kümeleme sorunları gibi problemlerden kaynaklanabilmektedir. DMHAKA yaklaşımının, EE-WCA, TSDR, E-MAVMMF ve MORS-ASO yaklaşımları ile 20, 40, 60, 80 ve 100 düğümlü ağ ortamları için ağ çıkışı kriterine göre karşılaştırılması Şekil 4.9'ta gösterilmektedir.



Şekil 4.9. Düğüm sayısına göre veri aktarım kapasitesi karşılaştırma grafiği.

Şekil 4.9 incelendiğinde, önerilen yöntemin diğer yöntemlere kıyasla daha yüksek veri aktarım kapasitesi sağladığı görülmektedir. Örneğin, 60 düğüm içeren bir ağ senaryosunda önerilen yöntem 33.800 bps veri aktarım kapasitesi elde ederken, karşılaştırılan diğer yöntemlerin veri aktarım kapasitesi değerleri sırasıyla 23.100, 26.200, 25.600 ve 30.500 bps olarak ölçülmüştür.

Özellikle düğüm sayısı arttıkça, önerilen yöntemin veri aktarım kapasitesi değerlerinde diğer yöntemlere kıyasla daha istikrarlı ve düzenli bir artış eğilimi sergilediği dikkat çekmektedir. Diğer yöntemlerde de düğüm sayısının artmasına paralel olarak veri aktarım kapasitesinde bir artış gözlenirse de bu artış DMHAKA kadar belirgin değildir.

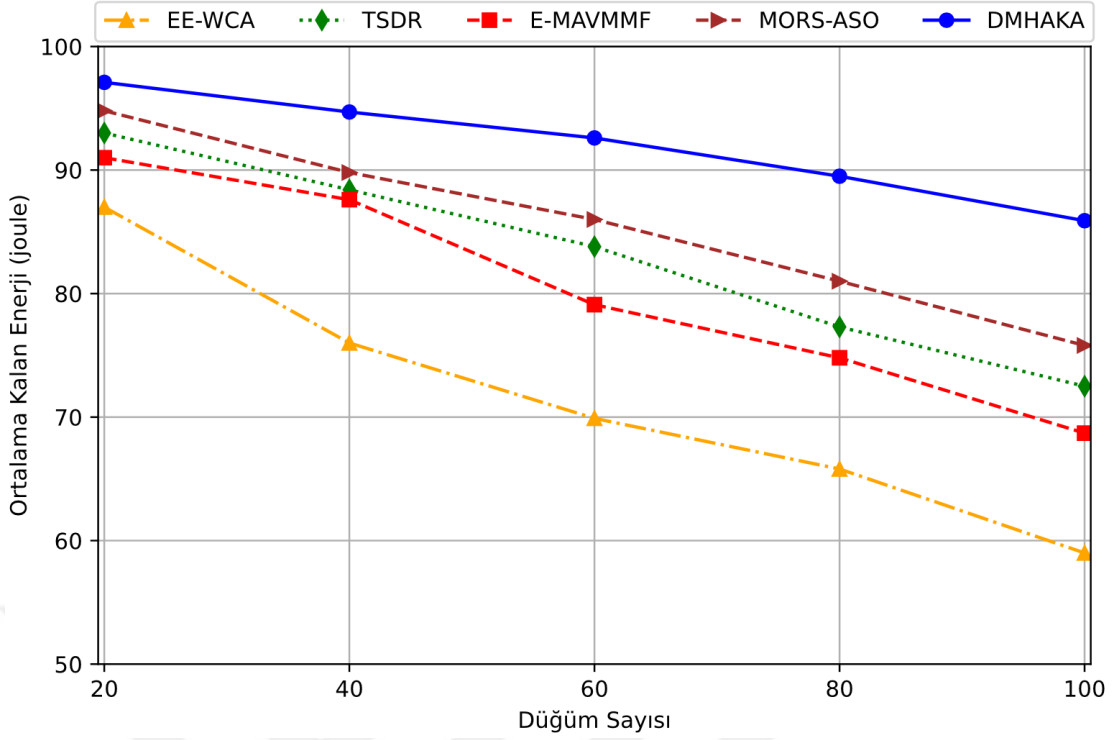
Bu sonuçların temel nedeni, önerilen yöntemin küme kararlılığının yüksek olması ve KB değişim oranının düşük seviyelerde tutulmasıdır. Yüksek küme kararlılığı, ağ içi iletişimin daha tutarlı ve verimli gerçekleşmesini sağlamakta, bu da veri aktarım kapasitesi performansını doğrudan olumlu etkilemektedir. Elde edilen bulgular, önerilen yöntemin hem ölçeklenebilir hem de yüksek düğüm sayılı ağlarda dahi güçlü performans gösterdiğini ortaya koymaktadır.

4.3.2.2. Ortalama kalan enerjiye göre performans analizi

Düğümün bilgi gönderme, alma veya iletme gibi farklı ağ görevlerini gerçekleştirmek için başlangıçtaki enerjilerini harcarlar. Bu görevler sırasında düğümün pilindeki enerjinin azalmasına ise enerji tüketimi denir. Düğümün düşük enerji tüketimi, ağın kararlılığını artırır ve daha uzun bir ağ ömrü sağlar. Dolayısıyla ağ ömrü ile düğümlerin kalan enerjileri arasında doğru orantılı bir ilişki vardır. Ağın yüzdelik olarak kalan enerjisi Denklem (4.1) ve Denklem (4.2)'deki gibi hesaplanmaktadır.

Daha yüksek kalan enerji, düğümlerin daha az enerji tükettiğini ve ağ ömrünün daha uzun olacağını göstermektedir. Eğer ortalama kalan enerji düşükse, bu ağın hızlı bir şekilde tükendiği ve düğümlerin ömrünün kısa olduğu anlamına gelmektedir.

Önerilen yöntemin farklı düğüm sayıları için ortalama kalan enerji değerlerinin, EE-WCA, TSDR, E-MAVMMF ve MORS-ASO yöntemleriyle karşılaştırması Şekil 4.10'da gösterilmektedir.



Şekil 4.10. Düğüm sayısına göre kalan enerji miktarlarının karşılaştırılma grafiği.

Bu grafik, karşılaştırılan yöntemlerin düğüm sayısı arttıkça gösterdiği enerji tüketim performansını ortaya koymaktadır. EE-WCA, TSDR, E-MAVMMF, MORS-ASO ve önerilen DMHAKA yöntemi için 20, 40, 60, 80 ve 100 düğümlü ağlarda elde edilen ortalama kalan enerji değerleri (Joule cinsinden) yüzde olarak karşılaştırılmıştır.

Elde edilen bulgulara göre, düğüm sayısı arttıkça tüm yöntemlerde enerji seviyelerinde azalma gözlenmiştir. Ancak bu azalma oranları arasında belirgin farklar bulunmaktadır. Özellikle DMHAKA yöntemi, artan düğüm sayısına rağmen en istikrarlı enerji verimliliği performansını sergileyerek, diğer yöntemlere göre çok daha yavaş enerji kaybı göstermiştir. Düğüm sayısı 100'e ulaştığında, DMHAKA ile elde edilen ortalama kalan enerji değeri, diğer tüm yöntemlerden kayda değer biçimde yüksek kalmıştır. Bu sonuçlar, önerilen yöntemin büyük ölçekli ağlarda enerji optimizasyonu açısından önemli bir potansiyele sahip olduğunu göstermekte ve uzun vadede enerji tasarrufu sağlayabilecek etkili bir yaklaşım olduğunu ortaya koymaktadır.

4.3.2.3. Paket teslim oranına göre performans analizi

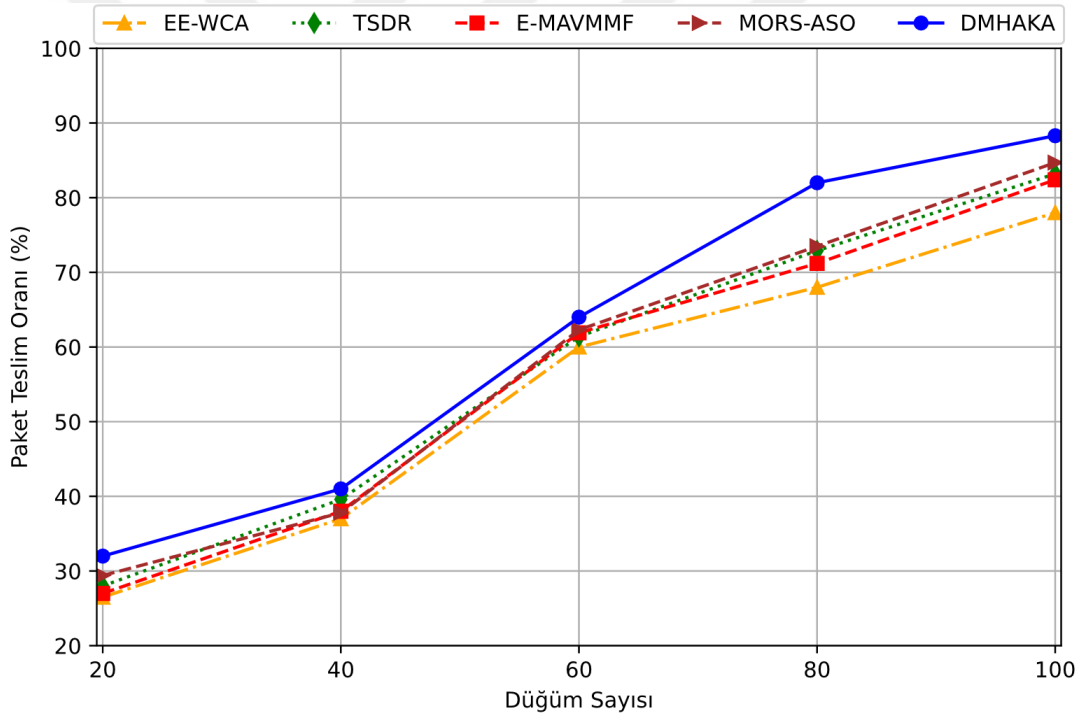
Hedef tarafından alınan toplam paket sayısının kaynaktan gönderilen toplam paket sayısına oranıdır [36]. Bu ölçü hedefe ulaşan sağlıklı paketlerin sayısı anlamına gelir. Formülü Denklem (4.7)'de gösterildiği gibidir.

$$Paket\ teslim\ oranı = \frac{NRD}{NSD} \times 100 \quad (4.7)$$

Burada NRD hedef düğüm tarafından alınan toplam paket sayısı ve NSD kaynak düğüm tarafından gönderilen toplam paket sayısıdır.

Daha yüksek paket teslim oranı, ağın paket kaybının düşük olduğunu ve küme içi-kümeler arası iletimin verimli çalıştığını gösterir. Düşük paket teslim oranı ise yüksek paket kaybına ve düşük güvenilirliğe işaret eder ve ağ tıkanıklığı, yönlendirme hataları veya düğüm hareketliliği gibi sorunlardan kaynaklanabilmektedir.

DMHAKA yaklaşımının, EE-WCA, TSDR, E-MAVMMF ve MORS-ASO yaklaşımları ile 20, 40, 60, 80 ve 100 düğümlü ağ ortamları için paket teslim oranı kriterine göre karşılaştırılması Şekil 4.11’de gösterilmektedir.



Şekil 4.11. Düğüm sayısına göre paket teslim oranı grafiği.

Şekil 4.11’deki grafiğe göre, tüm yöntemlerde paket teslim oranı artış göstermektedir. Ancak önerilen DMHAKA yöntemi, diğer yaklaşımlara kıyasla daha yüksek bir paket teslim oranı değeri elde etmiştir. Bu iyileşme büyük ölçüde, yöntemin optimize edilmiş KB seçim mekanizmasına dayanmakta olup, ağ içi trafik yükünün dengeli dağıtılmasını sağlamakta ve paket kayıplarını en aza indirmektedir.

Ayrıca, önerilen yönteme entegre edilen geliştirilmiş DBSCAN algoritması, düğümlere optimal rollerin atanmasını kolaylaştırarak, uyarlanabilir kümeleme yeteneğini önemli ölçüde artırmaktadır. Bu sayede, gereksiz iletimler ve çakışmalar azaltılmakta, ağ kararlılığı ise güçlendirilmektedir.

Önerilen yöntemin verimli ağ yönetimi ve enerji farkındalığına dayalı tasarımı, ağ ömrünü uzatmakta ve sürekli iletişimi desteklemektedir. Bu da genel anlamda paket teslim oranı değerinin artmasına doğrudan katkı sağlamaktadır.

Elde edilen bulgular, önerilen yaklaşımın daha yoğun ağ yapılarında ve dinamik koşullarda bile daha güvenilir veri iletimi sunduğunu ve bu yönüyle mevcut yöntemlere göre daha üstün bir performans sergilediğini ortaya koymaktadır.

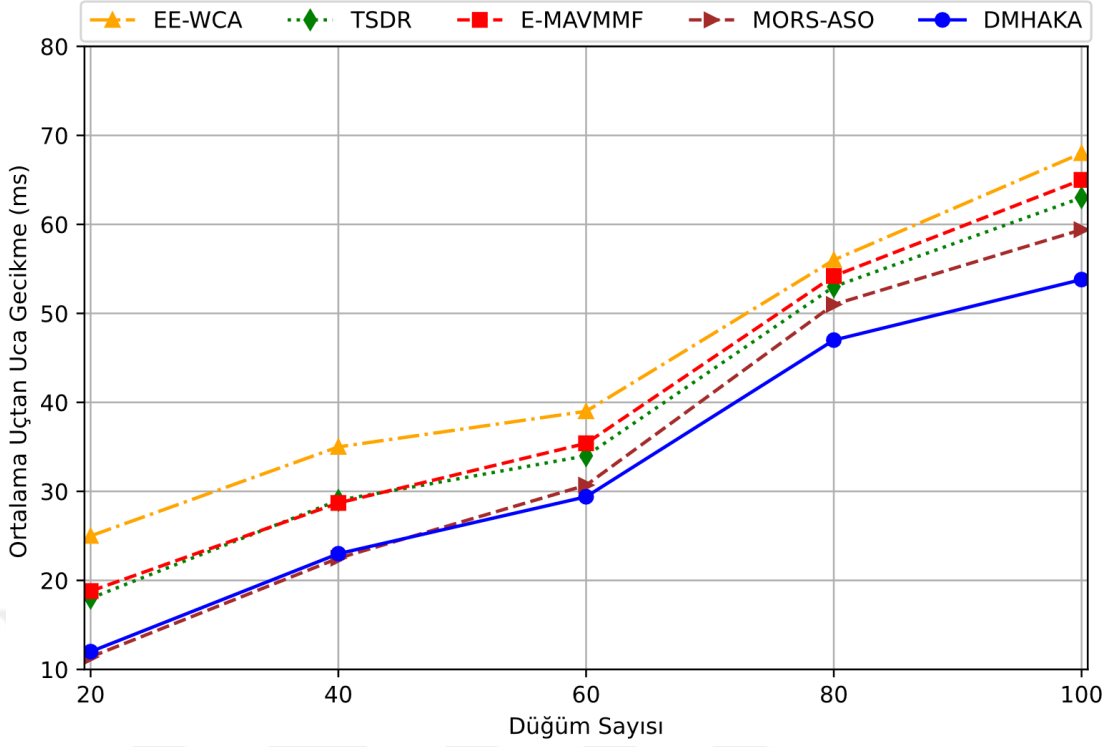
4.3.2.4. Ortalama uçtan uca gecikmeye göre performans analizi

Bir veri paketinin kaynak düğümden hedef düğüme aktarılması için geçen milisaniye (ms) cinsinden tüm sürelerin ortalamasıdır. Bu süre, her paketin ulaşım zamanı ile gönderim zamanı arasındaki farkların, toplam başarıyla alınan paket sayısına bölünmesiyle hesaplanır [109]. Formülü Denklem (4.8)'de gösterildiği gibidir.

$$\text{Ortalama uçtan uca gecikme} = \frac{\sum_{i=1}^N T_{\text{alım},i} - T_{\text{gönderim},i}}{N} \quad (4.8)$$

Burada $T_{\text{alım},i}$ i 'ninci paketin alım zamanı, $T_{\text{gönderim},i}$ i 'ninci paketin gönderim zamanı ve N ise Başarıyla alınan toplam paket sayısını ifade etmektedir. Düşük gecikme değerleri, ağın daha hızlı veri aktarımı yapabildiğini ve daha kararlı çalıştığını gösterir. Yüksek gecikme ise yoğun trafik, yönlendirme hataları veya düğüm hareketliliği nedeniyle ağın verimli çalışmadığını gösterebilmektedir.

Önerilen yöntemin ortalama uçtan uca gecikme değerlerinin, EE-WCA, TSDR, E-MAVMMF ve MORS-ASO yöntemleriyle karşılaştırılması Şekil 4.12'de gösterilmektedir.



Şekil 4.12. Düğüm sayısına göre uçtan uca gecikme grafiği.

Şekil 4.12’deki grafiğe göre, tüm yöntemlerde düğüm sayısındaki artışla birlikte ortalama uçtan uca gecikme de artmaktadır. Ancak, önerilen yöntem gecikmedeki artışı minimum seviyede tutarak, EE-WCA, TSDR, E-MAVMMF ve MORS-ASO yaklaşımlarına kıyasla üstün bir performans sergilemektedir.

60 düğümden daha az sayıda düğüm içeren ağlarda, MORS-ASO yöntemi önerilen yöntemden daha iyi sonuçlar vermektedir. Ancak 60 düğüm ve üzeri senaryolarda, önerilen yöntemin gecikme açısından daha başarılı olduğu görülmektedir. Bu sonuçlar, önerilen yöntemin yüksek düğüm yoğunluklarına sahip ağ yapılarında gecikmeye duyarlı uygulamalar için daha uygun olduğunu ve daha etkili bir performans sergileyebileceğini göstermektedir.

Ayrıca, DMHAKA yönteminin uyarlanabilir yapısı, yüksek düğüm hareketliliğinin olduğu dinamik ağ koşullarında ağ kararlılığının korunmasında önemli bir rol oynamaktadır. Bu bağlamda, dinamik KB seçiminde kullanılan GSA algoritması ve esnek küme yapılarının oluşturulmasına olanak tanıyan Geliştirilmiş- DBSCAN algoritması, zamanında rol değişimi gerçekleştirilmesini sağlamak ve yeniden kümeleme yükünü azaltarak ağın genel performansını korumaktadır.

Çizelge 4.3'te yer alan iyileşme oranları, önerilen DMHAKA yönteminin her bir performans metriği açısından diğer yöntemlere göre sağladığı yüzde (%) bazlı iyileştirmeleri ifade etmektedir.

Çizelge 4.3. Farklı düğüm sayılarında DMHAKA yönteminin diğer yöntemlere göre iyileştirme oranları.

Metrikler	DMHAKA – EE-WCA Karşılaştırması	DMHAKA – E-MAVMMF Karşılaştırması	DMHAKA –TSDR Karşılaştırması	DMHAKA –MORS-ASO Karşılaştırması
Veri	%23,42 (N=20)	%22,01 (N=20)	%21,88 (N=20)	%4,63 (N=20)
Aktarım	%17,50 (N=40)	%12,98 (N=40)	%11,90 (N=40)	%8,29 (N=40)
Kapasitesi	%46,32 (N=60)	%32,03 (N=60)	%30,03 (N=60)	%10,82 (N=60)
	%38,52 (N=80)	%29,68 (N=80)	%24,67 (N=80)	%16,15 (N=80)
	%29,08 (N=100)	%26,60 (N=100)	%23,05 (N=100)	%10,64 (N=100)
Ortalama	%11,61 (N=20)	%6,70 (N=20)	%4,41 (N=20)	%2,42 (N=20)
Kalan	%24,61 (N=40)	%8,10 (N=40)	%7,13 (N=40)	%5,46 (N=40)
Enerji	%32,46 (N=60)	%17,08 (N=60)	%10,52 (N=60)	%7,67 (N=60)
	%36,02 (N=80)	%19,60 (N=80)	%15,74 (N=80)	%10,49 (N=80)
	%45,59 (N=100)	%24,97 (N=100)	%18,48 (N=100)	%13,34 (N=100)
Paket	%20,75 (N=20)	%18,11 (N=20)	%9,71 (N=20)	%5,44 (N=20)
Teslim	%12,11 (N=40)	%5,26 (N=40)	%4,01 (N=40)	%7,82 (N=40)
Oranı	%26,67 (N=60)	%19,38 (N=60)	%14,22 (N=60)	%10,24 (N=60)
	%20,59 (N=80)	%15,15 (N=80)	%12,42 (N=80)	%12,56 (N=80)
	%13,21 (N=100)	%12,16 (N=100)	%10,13 (N=100)	%9,75 (N=100)
Ortalama	%52,00 (N=20)	%36,17 (N=20)	%33,33 (N=20)	%-5,26(N=20)
Uçtan Uca	%34,29 (N=40)	%19,86 (N=40)	%20,69 (N=40)	%-2,22(N=40)
Gecikme	%24,62 (N=60)	%16,95 (N=60)	%13,53 (N=60)	%4,23 (N=60)
	%16,07 (N=80)	%13,31 (N=80)	%11,32 (N=80)	%7,84 (N=80)
	%20,88 (N=100)	%17,23 (N=100)	%14,60 (N=100)	%9,43 (N=100)

Çizelge 4.3, önerilen DMHAKA yönteminin dört farklı güncel kümeleme algoritmasına (EE-WCA, E-MAVMMF, TSDR ve MORS-ASO) karşı çeşitli performans kriterleri açısından sağladığı iyileşme oranlarını ortaya koymaktadır. Veri aktarım kapasitesi, ortalama kalan enerji, paket teslim oranı ve ortalama uçtan uca gecikme gibi temel metrikler üzerinden yapılan bu karşılaştırma, DMHAKA'nın özellikle yüksek düğüm sayılı ve yoğun ağ yapılarında belirgin bir avantaj sunduğunu göstermektedir.

Özellikle veri aktarım kapasitesi ve enerji verimliliği metriklerinde DMHAKA'nın sağladığı iyileşme oranları, diğer yöntemlerle kıyaslandığında oldukça dikkat çekicidir. Örneğin, 60 düğüm senaryosunda veri aktarım kapasitesi açısından EE-WCA'ya göre %46'lık, E-MAVMMF'ye göre %32, TSDR'ye göre %30 ve MORS-ASO'ya göre ise yaklaşık %11'lik bir iyileşme elde edilmiştir. Aynı senaryoda kalan enerji değerleri de benzer şekilde DMHAKA'nın enerji farkındalığı yüksek tasarım yapısının etkili

olduđunu ortaya koymaktadır. Bunun yanında, paket teslim oranı aısından elde edilen kazanımlar ađın veri iletim gvenilirliđini artırırken, ortalama utan uca gecikme deđerlerindeki dşşler de gecikmeye duyarlı uygulamalar iin DMHAKA'yı daha uygun hale getirmektedir.

Genel olarak deđerlendirildiđinde, nerilen yntem hem kararlılık hem de verimlilik aısından mevcut algoritmalara karşı oklu metriklerde stnlk sađlamakta ve zellikle dinamik, yođun ve enerjiye duyarlı MANET ortamlarında leklenebilir bir zm sunmaktadır.



5. SONUÇLAR VE ÖNERİLER

Bu bölümde, geliştirilen DMHAKA kümeleme yönteminin simülasyon ortamında elde edilen bulgular doğrultusunda değerlendirilmesi yapılmakta ve tez çalışmasından elde edilen genel sonuçlar sunulmaktadır. Ayrıca, önerilen yöntemin klasik ve güncel algoritmalara kıyasla sağladığı katkılar açıklanmakta ve gelecekte yapılabilecek çalışmalara yönelik öneriler tartışılmaktadır. Performans analizlerinde kullanılan metrikler doğrultusunda, yöntemin ağ kararlılığı, enerji verimliliği ve veri iletim kapasitesi açısından elde ettiği iyileşmeler detaylı olarak ele alınmıştır.

5.1. SONUÇLAR

Bu tez çalışması kapsamında, KB seçim sürecini iyileştirmek amacıyla GSA ve WCA birlikte kullanılarak optimize edilmiş bir KB belirleme yaklaşımı geliştirilmiştir. Bu sürecin ardından, Geliştirilmiş-DBSCAN algoritması ile düğüm rollerinin atanması sağlanarak esnek, dengeli ve enerji açısından verimli kümeler oluşturulmuştur. Geliştirilen DMHAKA yöntemi, klasik kümeleme algoritmalarından WCA ve literatürde öne çıkan EE-WCA, E-MAVMMF, TSDR ve MORS-ASO gibi güncel yöntemlerle karşılaştırmalı olarak değerlendirilmiştir.

Simülasyon ortamında gerçekleştirilen kapsamlı analizlerde DMHAKA yöntemi; veri aktarım kapasitesi, ortalama kalan enerji, ortalama KB ömrü, ortalama küme ömrü, KB değişim oranı, uçtan uca gecikme ve paket teslim oranı gibi kritik performans metriklerinde karşılaştırılan diğer yöntemlere kıyasla anlamlı iyileşmeler sağlamıştır. Özellikle:

WCA'ya karşı:

- KB değişim oranı: %95 → %65'e düşürüldü
- Ortalama KB ömrü (100 düğüm): 7 sn (WCA) → 22 sn (DMHAKA)
- Ortalama kalan enerji (100 düğüm): 55 J (WCA) → 86 J (DMHAKA)
- Küme ömrü ve stabilitesi gözle görülür şekilde arttı

EE-WCA, E-MAVMMF, TSDR, MORS-ASO'ya karşı:

- Enerji tüketimi iyileşme oranları: %46,38 (EE-WCA), %18,35 (E-MAVMMF), %13,08 (TSDR), %8,33 (MORS-ASO)
- Paket teslim oranı iyileşme oranları: %18,67 (EE-WCA), %14,01 (E-MAVMMF), %10,10 (TSDR), %9,16 (MORS-ASO)
- Ortalama uçtan uca gecikme: 60 düğümden sonra tüm yöntemlerden daha düşük gerçekleşmiş; iyileşme %29,58'e kadar ulaşmıştır,
- Veri aktarım kapasitesi iyileşme oranları: %30,97 (EE-WCA), %24,66 (E-MAVMMF), %22,33 (TSDR), %10,11 (MORS-ASO)

Ayrıca, DMHAKA yöntemi düğüm sayısı arttıkça performansını sürdürebilmiş ve kümeleme yapısını kararlı şekilde korumuştur. Bu durum, özellikle dinamik ve yoğun düğüm dağılımına sahip ağ senaryolarında dahi önerilen yöntemin ölçeklenebilirliğini ve sürdürülebilirliğini göstermektedir.

5.2. ÇALIŞMANIN GETİRDİĞİ KATKILAR

Bu çalışmanın bilime ve teknolojiye getirdiği ana katkılar aşağıda maddeler halinde sunulmaktadır:

1. Bilime ve teknolojiye yenilik getirme:
 - a) KB seçiminde GSA ve WCA algoritmaları birlikte kullanılarak, yeni bir KB seçim yaklaşımı geliştirilmiştir.
 - b) Geliştirilen Geliştirilmiş-DBSCAN algoritmasında, eps parametresi her KB için yerel düğüm yoğunluğuna göre dinamik olarak hesaplanmış ve manuel parametre ayarlama ihtiyacı ortadan kaldırılmıştır. Bu özellik, yöntemin değişken ağ topolojilerine ve düğüm yoğunluklarına karşı daha uyurlanabilir ve esnek çalışmasını mümkün kılmıştır.
 - c) KB değişim oranı düşürülerek ağ kararlılığı artırılmış, yeniden kümeleme ihtiyacı azaltılmıştır. Bu durum hem enerji tüketiminin azaltılmasına hem de ağ ömrünün uzatılmasına katkı sağlamıştır. Bu katkılar, önerilen yöntemin kararlı ve enerji farkındalığı yüksek bir kümeleme yaklaşımı sunduğunu göstermektedir.

d) DMHAKA yöntemi, veri aktarım kapasitesi, kalan enerji, gecikme ve paket teslim oranı gibi temel performans metrikleri üzerinden WCA, EE-WCA, TSDR, E-MAVMMF ve MORS-ASO algoritmaları ile karşılaştırılarak çok yönlü ve ölçülebilir bir değerlendirme modeli oluşturmuştur. Her bir metrikte sayısal iyileşme oranları hesaplanarak somut kazanımları ortaya koymuştur.

2. Bilinen bir yöntemin yeni bir alana uygulanması:

a) GSA algoritması, bu çalışmada MANET ortamındaki KB seçim problemine entegre edilmiş ve çok kriterli seçim mekanizmasının bir parçası olarak kullanılmıştır. Bu yaklaşım, GSA'nın dinamik ve mobil ağ yapılarında küme kararlılığına yönelik katkı sağlayabileceğini göstermektedir.

b) DBSCAN algoritması, geleneksel kullanımının ötesine geçerek, MANET ortamında KB çevresinin belirlenmesi ve düğüm rolü atama amacıyla yapılandırılmıştır. Bu kapsamda geliştirilen Geliştirilmiş-DBSCAN yapısı, DBSCAN'ın mobil ve değişken topolojilere sahip ağlarda kullanılabilirliğini artırmıştır.

c) Bu çalışmada sadece belirli düğüm yoğunluklarında performans ölçümleri yapılmakla kalmamış, aynı zamanda klasik (WCA) ve güncel yöntemlerle yapılan karşılaştırmalar, her bir metrik için sayısal iyileşme oranları ile desteklenmiştir. Böylece, DMHAKA'nın farklı ağ senaryolarındaki başarımı yalnızca grafiksel olarak değil, nicel karşılaştırma verileriyle de değerlendirilmiş, sonuçlar açık biçimde karşılaştırılabilir hâle getirilmiştir.

5.3. TARTIŞMALAR VE ÖNERİLER

Bu tez çalışmasında geliştirilen DMHAKA yöntemi, klasik ve güncel kümeleme algoritmalarıyla kapsamlı biçimde karşılaştırılmış ve performans kriterleri açısından anlamlı iyileşmeler sağlamıştır. Ancak her bilimsel çalışmada olduğu gibi, bu çalışmanın da bazı sınırlılıkları bulunmaktadır ve gelecekte yapılacak çalışmalara ışık tutacak öneriler aşağıda sunulmuştur:

1. Gerçek zamanlı uygulama ve protokol dönüşümü:

Bu tezde DMHAKA yöntemi simülasyon ortamında test edilmiştir. Gerçek donanım ortamında, özellikle mobil cihazlar veya kablosuz algılayıcı ağlar gibi sınırlı kaynaklı sistemler üzerinde uygulanarak protokol düzeyinde bir uygulamaya dönüştürülmesi, yöntemin pratik kullanılabilirliğini artıracaktır.

2. Farklı mobilite modelleri ile genişletilmiş performans değerlendirmesi:

Bu çalışmada DMHAKA yöntemi, 2–10 m/s hız aralığında hareket eden düğümlerle test edilmiştir. Ancak yalnızca rastgele yön hareket modeli kullanılmıştır. Farklı mobilite modelleri ile algoritmanın farklı hareket desenlerine karşı dayanıklılığı ve kararlılığı test edilerek, yöntem daha geniş ağ senaryolarında değerlendirilebilir.

3. Yeni performans metrikleri ve optimizasyon hedefleri:

Bu çalışmada veri aktarım kapasitesi, ortalama kalan enerji, KB değişim oranı, ortalama KB ömrü, ortalama küme ömrü, paket teslim oranı ve ortalama uçtan uca gecikme metrikleri temel alınmıştır. Yük dengesi, bant genişliği kullanımı, güvenlik farkındalığı ve bağlantı süresi gibi metriklerle genişletilmiş değerlendirmeler yapılabilir. Ayrıca DMHAKA yönteminin çok hedefli optimizasyon algoritmaları ile birlikte performansı araştırılabilir.

4. Gelişmiş sezgisel algoritmalarla entegrasyon:

Bu çalışmada GSA'nın çok kriterli optimizasyon yeteneklerinden yararlanılarak önemli başarımlar elde edilmiştir. Ancak, literatürde ACO ve PSO gibi diğer sezgisel algoritmalara dayalı birçok KB seçimi çalışması mevcuttur. Bu bağlamda, gelecekteki çalışmalarda klasik sezgisel algoritmalar yerine, bu algoritmaların hibritleştirilmiş, dinamik MANET ortamlarına özel olarak uyarlanmış veya geliştirilmiş sürümleri ile DMHAKA yönteminin entegrasyonu sağlanabilir. Böylece mevcut yaklaşımın performansı, gelişmiş meta-sezgisel stratejilerle daha ileri düzeye taşınabilir ve farklı ağ koşulları altında kararlılığı derinlemesine analiz edilebilir.

5. Derin öğrenme destekli yönlendirme ve GD seçimi:

DMHAKA'nın küme yapısı üzerine derin öğrenme temelli yönlendirme protokolleri, özellikle yoğun ağ yapılarında daha verimli veri akışı sağlayabilir.

GD seçimi için çevresel bağlamı (örneğin kalan enerji, trafik yoğunluğu, sinyal gücü) öğrenen modeller geliştirilebilir.



6. KAYNAKLAR

- [1] A. Ahmad, *Wireless and Mobile Data Networks*, 1. baskı, New Jersey, A.B.D: Wiley-Interscience, 2005.
- [2] R. Yadav ve S. Rao, “A Survey of various routing protocols in MANETs”, *International Journal of Computer Science and Information Technologies* c. 6, sayı 5, ss. 4587–4592, 2015.
- [3] R. Kara, D. Özçeliz, ve H. Ekiz, “Analysis of Fuzzy Logic Usage in PBHRA Routing Algorithm in Wireless Ad Hoc Networks”, *2010 6th International Conference on Wireless and Mobile Communications*, IEEE, 2010, (ss. 337–342), Valencia, İspanya.
- [4] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, ve E. Cayirci, “Wireless sensor networks: A survey”, *Computer Networks*, c. 38, sayı 4, ss. 393–422, 2002.
- [5] L. Hanzo ve R. Tafazolli, “A survey of qos routing solutions for mobile ad hoc networks”, *IEEE Communication Survey Tutorials*, c. 9, sayı 2, ss. 50–70, 2007.
- [6] O. Boyinbode, H. Le, ve M. Takizawa, “A survey on clustering algorithms for wireless sensor networks”, *International Journal of Space-Based and Situated Computing*, c. 1, sayı 2–3, ss. 130–136, 2011.
- [7] S. A. Sharifi ve S. M. Babamir, “Evaluation of clustering algorithms in Ad Hoc mobile networks”, *Wireless Personal Communications*, c. 109, sayı 4, ss. 2147-2186, 2019.
- [8] S. M. Almheiri ve H. S. Alqamzi, “MANETs and VANETs clustering algorithms: A survey”, *2015 IEEE 8th GCC Conference and Exhibition, GCCCE 2015*, IEEE, 2015, ss. 1–6.
- [9] M. Alinci, E. Spaho, A. Lala, ve V. Kolicic, “Clustering Algorithms in MANETs: A Review”, *Proceedings - 2015 9th International Conference on Complex, Intelligent, and Software Intensive Systems, CISIS 2015*, IEEE, 2015, ss. 330–335.
- [10] M. Gerla ve J. Tzu-Chieh Tsai, “Multicluster, mobile, multimedia radio network”, *Wireless Networks*, c. 1, sayı 3, ss. 255–265, 1995.

- [11] S. Pathak ve S. Jain, “A novel weight based clustering algorithm for routing in MANET”, *Wireless Networks*, c. 22, sayı 8, ss. 2695–2704, 2016.
- [12] R. Agarwal ve D. M. Motwani, “Survey of clustering algorithms for MANET”, *arXiv Preprint, arXiv0912.2303*, 2009, doi: doi.org/10.48550/arXiv.0912.2303.
- [13] D. M. Bokhari, H. S. A. Hamatta, ve S. T. Siddigui, “A review of clustering algorithms as applied in MANETs”, *International Journal of Advanced Research in Computer Science and Software Engineering*, c. 2, sayı 11, 2012.
- [14] W. R. Heinzelman, A. Chandrakasan, ve H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks”, *Proceedings of the Hawaii International Conference on System Sciences*, 2000, ss. 223.
- [15] I. Daanoune, B. Abdennaceur, ve A. Ballouk, “A comprehensive survey on LEACH-based clustering routing protocols in Wireless Sensor Networks”, *Ad Hoc Networks*, c. 114, ss. 102409, 2021.
- [16] M. Chatterjee, S. K. Das, ve D. Turgut, “On-demand weighted clustering algorithm (WCA) for ad hoc networks”, *Conference Record / IEEE Global Telecommunications Conference*, IEEE, 2000, ss. 1697–1701.
- [17] S. K. Dhurandher ve G. V. Singh, “Weight based adaptive clustering in wireless ad hoc networks”, *IEEE International Conference on Personal Wireless Communications*, IEEE, 2005, ss. 95–100.
- [18] S. Pathak, S. Jain, ve S. Borah, “Clustering Algorithms for MANETs: A Review on Design and Development”, *Advances in Intelligent Systems and Computing*, ss. 563–578, 2021.
- [19] S. Pathak ve S. Jain, “An optimized stable clustering algorithm for mobile ad hoc networks”, *Eurasip Journal on Wireless Communications and Networking*, c. 2017, sayı 1, ss. 1–11, 2017.
- [20] S. Yang, T. L. Li, D. Wu, T. Hu, W. Deng, ve H. Gong, “Bio-inspired multi-hop clustering algorithm for FANET”, *Ad Hoc Networks*, c. 154, ss. 103355, 2024.
- [21] S. Pathak ve S. Jain, “Comparative study of clustering algorithms for MANETs”, *Journal of Statistics and Management Systems*, c. 22, sayı 4, ss. 653–664, 2019.

- [22] P. Gulganwa ve S. Jain, “EE-WCA: energy efficient weighted clustering algorithm to regulate application’s quality of service requirements”, *Wireless Personal Communication*, c. 124, sayı 4, ss. 3647–3660, 2022.
- [23] P. Gulganwa ve S. Jain, “EES-WCA: Energy efficient and secure weighted clustering for WSN using machine learning approach”, *International Journal of Information Technology.*, c. 14, sayı 1, ss. 135–144, 2022.
- [24] O. Younis ve S. Fahmy, “HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks”, *IEEE Transactions on Mobile Computing*, c. 3, sayı 4, ss. 366–379, 2004.
- [25] H. Ali, W. Shahzad, ve F. A. Khan, “Energy-efficient clustering in mobile ad-hoc networks using multi-objective particle swarm optimization”, *Applied Soft Computing Journal*, c. 12, sayı 7, ss. 1913–1928, 2012.
- [26] K. M. Awan, H. H. R. Sherazi, A. Ali, R. Iqbal, Z. A. Khan, ve M. Mukherjee, “Energy-aware cluster-based routing optimization for WSNs in the livestock industry”, *Transactions on Emerging Telecommunications Technologies*, c. 33, sayı 3, ss. e3816, 2022.
- [27] M. B. Kokare ve D. Kakkar, “A survey on clustering algorithms for cluster-head selection in VANET”, *Proceedings of the 2nd International Conference on Electronics and Sustainable Communication Systems, ICESC 2021*, IEEE, 2021, ss. 992–996.
- [28] B. Pitchaimanickam ve G. Murugaboopathi, “A hybrid firefly algorithm with particle swarm optimization for energy efficient optimal cluster head selection in wireless sensor networks”, *Neural Computing and Applications*, c. 32, sayı 12, ss. 7709–7723, 2020.
- [29] M. R. Jose ve S. M. C. Vigila, “A Comparative Analysis of MANET Routing Protocols on Cluster Head selection: with fuzzy logic approach and without fuzzy logic approach”, *9th International Conference on Smart Computing and Communications: Intelligent Technologies and Applications, ICSCC 2023*, IEEE, 2023, ss. 369–373.
- [30] O. K. Abbas, F. Abdullah, N. A. M. Radzi, A. D. Salman, ve S. J. Abdulkadir, “Survey on clustered routing protocols adaptivity for fire incidents: architecture

- challenges, data losing, and recommended solutions”, *IEEE Access*, c. 12, ss. 113518–113552, 2024.
- [31] M. R. Jose ve S. M. C. Vigila, “F-CAPSO: Fuzzy chaos adaptive particle swarm optimization for energy-efficient and secure data transmission in MANET”, *Expert System Application*, c. 234, ss. 120944, 2023.
- [32] Y. Zhang, J. Lu, C. Zhao, Z. Li, ve J. Yan, “Chaos Optimization Algorithms: A Survey”, *International Journal of Bifurcation and Chaos*, c. 34, sayı 16, ss. 2450205, 2024.
- [33] U. Srilakshmi, N. Veeraiah, Y. Alotaibi, S. A. Alghamdi, O. I. Khalaf, ve B. V. Subbayamma, “An improved hybrid secure multipath routing protocol for MANET”, *IEEE Access*, c. 9, ss. 163043–163053, 2021.
- [34] V. K. Quy, V. H. Nam, D. M. Linh, ve L. A. Ngoc, “Routing Algorithms for MANET-IoT Networks: A comprehensive survey”, *Wireless Personal Communication*, c. 125, sayı 4, ss. 3501–3525, 2022.
- [35] B. Prabu ve G. Jagatheeshkumar, “Hex-Cell Clustering Based Distributed Service Discovery Algorithm for MANETs”, *2024 15th International Conference on Computing Communication and Networking Technologies, ICCCNT 2024*, IEEE, 2024, ss. 1–7.
- [36] P. Arulprakash, A. S. Kumar, ve S. P. Prakash, “Optimal route and cluster head selection using energy efficient-modified African vulture and modified mayfly in manet”, *Peer-to-Peer Network Application*, c. 16, sayı 2, ss. 1310–1326, 2023.
- [37] S. Hemalatha vd., “A Review of Power Management Approaches for Mobile Ad Hoc Networks”, *Journal Européen des Systèmes Automatisés*, c. 57, sayı 1, ss. 137–145, 2024.
- [38] R. Saravanan, K. Suresh, ve S. S. Arumugam, “A modified k-means-based cluster head selection and Philippine eagle optimization-based secure routing for MANET”, *The Journal of Supercomputing*, c. 79, sayı 9, ss. 10481–10504, 2023.
- [39] J. Hemagowri, C. Baranikumari, ve B. Brindha, “A study on proactive routing protocol in Ad-hoc network”, *International Journal of Modern Engineering Research*, c. 4, ss. 1–4, 2004.

- [40] Y. Sakurai ve J. Katto, “AODV multipath extension using source route lists with optimized route establishment”, *2004 International Workshop on Wireless Ad-Hoc Networks*, IEEE, 2005, ss. 63–67.
- [41] X. Hong, K. Xu, ve M. Gerla, “Scalable routing protocols for mobile ad hoc networks”, *IEEE Network*, c. 16, sayı 4, ss. 11–21, 2002.
- [42] N. H. Saeed, M. F. Abbod, ve H. S. Al-Raweshidy, “MANET routing protocols taxonomy”, *2012 International Conference on Future Communication Networks, ICFCN 2012*, IEEE, 2012, ss. 123–128.
- [43] S. Mohseni, R. Hassan, A. Patel, ve R. Razali, “Comparative review study of reactive and proactive routing protocols in MANETs”, *4th IEEE International Conference on Digital Ecosystems and Technologies - Conference Proceedings of IEEE-DEST*, 2010, ss. 304–309.
- [44] C. E. Perkins ve P. Bhagwat, “Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers”, *ACM SIGCOMM Computer Communication Review*, c. 24, sayı 4, ss. 234–244, 1994.
- [45] M. Horton, (2024, 7 Nisan). *Routing Protocols*, [Çevrimiçi]. Erişim: <https://slideplayer.com/slide/13159494>
- [46] S. Murthy ve J. J. Garcia-Luna-Aceves, “Routing protocol for packet radio networks”, içinde *Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM*, 1995, ss. 86–95.
- [47] P. Jacquet, P. Mühlethaler, T. Clausen, A. Laouiti, A. Qayyum, ve L. Viennot, “Optimized link state routing protocol for ad hoc networks”, *Proceedings - IEEE International Multi Topic Conference 2001: Technology for the 21st Century, IEEE INMIC 2001*, IEEE, 2001, ss. 62–68.
- [48] N. Enneya, K. Oudıdı, ve M. Elkoutbı, “Enhancing Delay in MANET Using OLSR Protocol”, *International Journal of Communications, Network and System Sciences*, c. 02, sayı 05, ss. 392–399, 2009.
- [49] F. De Rango, M. Fotino, ve S. Marano, “EE-OLSR: Energy efficient OLSR routing protocol for mobile ad-hoc networks”, *Proceedings IEEE Military Communications Conference MILCOM*, IEEE, 2008, ss. 1–7.

- [50] J. J. Garcia-Luna-Aceves ve M. Spohn, “Source-tree routing in wireless networks”, *Proceedings. Seventh International Conference on Network Protocols*, IEEE, 1999, ss. 273–282.
- [51] M. G. Zapata ve N. Asokan, “Securing ad hoc routing protocols”, *Proceedings of the 1st ACM workshop on Wireless security*, 2002, ss. 1–10.
- [52] S.-J. Lee, E. M. Belding-Royer, ve C. E. Perkins, “Ad hoc on-demand distance-vector routing scalability”, *ACM SIGMOBILE Mobile Computing and Communications Review*, c. 6, sayı 3, ss. 94–95, 2002.
- [53] G. Aggelou, *Mobile Ad Hoc Networks: From Wireless Lans to 4G Networks, 1. baskı*, Newyork, ABD: McGraw-Hill Professional, 2004.
- [54] D. B. Johnson ve D. A. Maltz, “DSR : The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks”, *Computer Science Department Carnegie Mellon University Pittsburgh*, c. 5, sayı 1, ss. 139–172, 1996.
- [55] S. Bitam, A. Mellouk, ve S. Zeadally, “HyBR: A Hybrid Bio-inspired Bee swarm Routing protocol for safety applications in Vehicular Ad hoc NETWORKS (VANETs)”, *Journal of Systems Architecture*, c. 59, sayı 10 PART B, ss. 953–967, 2013.
- [56] V. D. Park ve M. S. Corson, “A performance comparison of the temporally-ordered routing algorithm and ideal link-state routing”, *Proceedings - 3rd IEEE Symposium on Computers and Communications, ISCC 1998*, IEEE, 1998, ss. 592–598.
- [57] D. Kaur ve N. Kumar, “Comparative Analysis of AODV, OLSR, TORA, DSR and DSDV Routing Protocols in Mobile Ad-Hoc Networks”, *International Journal of Computer network and information security*, c. 5, sayı 3, ss. 39–46, 2012.
- [58] K. Raheja ve S. K. Maakar, “A Survey on Different Hybrid Routing Protocols of MANET”, *International Journal of Computer Science and Information Technologies*, c. 5, sayı 4, ss. 5512–5516, 2014.
- [59] A. Boukerche, B. Turgut, N. Aydin, M. Z. Ahmad, L. Bölöni, ve D. Turgut, “Routing protocols in ad hoc networks: A survey”, *Computer Networks*, c. 55, sayı 13, ss. 3032–3080, 2011.

- [60] A. Iwata, C. C. Chiang, G. Pei, M. Gerla, ve T. W. Chen, “Scalable routing strategies for ad hoc wireless networks”, *IEEE Journal on Selected Areas in Communications*, c. 17, sayı 8, ss. 1369–1379, 1999.
- [61] S. Patel, S. Rizvi, ve K. Elleithy, “Hierarchically Segmented Routing (HSR) Protocol for MANET”, *6th International Conference on Information Technology New Generations ITNG*, 2009, Las Vegas Nevada, A.B.D.
- [62] R. Sivakumar, P. Sinha, ve V. Bharghavan, “CEDAR: a core-extraction distributed ad hoc routing algorithm”, *IEEE Journal on Selected Areas in Communications*, c. 17, sayı 8, ss. 1454–1465, 1999.
- [63] W. Liu, C. Chiang, H. Wu, ve C. Gerla, “Routing in Clustered Multihop Mobile Wireless Networks with Fading Channel”, *Proceedings of IEEE Singapore International Conferans on Networks-SICON*, 1997, Singapur, Singapur ss. 197–211.
- [64] R. Xu ve D. Wunsch, “Survey of clustering algorithms”, *IEEE Transactions on Neural Networks*, c. 16, sayı 3, ss. 645–678, 2005.
- [65] A. K. Maurya, “Survey : Comparison Estimation of Various Routing P Rotocols in Mobile Ad -Hoc”, *International Journal of Distributed and Parallel Systems*, c. 5, sayı 1, ss. 87–96, 2014.
- [66] B. Zhou, Y. Z. Lee, ve M. Gerla, “Direction assisted geographic routing for mobile ad hoc networks”, *Proceedings - IEEE Military Communications Conference MILCOM*, IEEE, 2008, ss. 1–7.
- [67] Y.-B. Ko ve N. H. Vaidya, “Location-aided routing (LAR) in mobile ad hoc networks”, *Proceedings of the 4th annual ACM/IEEE International Conference on Mobile Computing and Networking*, 1998, ss. 66–75.
- [68] Y. B. Ko ve N. H. Vaidya, “GeoTORA: a protocol for geocasting in mobile ad hoc networks”, *International Conference on Network Protocols*, IEEE, 2000, ss. 240–250.
- [69] R. B. Al-Bayram ve R. M. Abdullah, “Network size variation of geographical aided routing protocols in MANET”, *Indonesian Journal of Electrical Engineering and Computer Science*, c. 21, sayı 1, ss. 420–428, 2021.

- [70] S. Basagni, I. Chlamtac, V. R. Syrotiuk, ve B. A. Woodward, “A distance routing effect algorithm for mobility (DREAM)”, *Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM*, 1998, ss. 76–84.
- [71] T. Camp, J. Boleng, B. Williams, L. Wilcox, ve W. Navidi, “Performance comparison of two location based routing protocols for ad hoc networks”, *Proceedings. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, IEEE, 2002, ss. 1678–1687.
- [72] M. Abolhasan, T. Wysocki, ve E. Dutkiewicz, “A review of routing protocols for mobile ad hoc networks”, *Ad Hoc Networks*, c. 2, sayı 1, ss. 1–22, 2004.
- [73] R. Kara, I. Ozcelik, ve H. Ekiz, “A new routing algorithm in MANETs: Position based hybrid routing”, *Scientific Research and Essays*, c. 5, sayı 3, ss. 328–338, 2010.
- [74] M. Bakhouya, J. Gaber, ve M. Wack, “Performance evaluation of DREAM protocol for inter-vehicle communication”, *Proceedings of the 2009 1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace and Electronic Systems Technology, Wireless VITAE 2009*, IEEE, 2009, ss. 289–293.
- [75] B. Karp ve H. T. Kung, “GPSR: Greedy Perimeter Stateless Routing for wireless networks”, *Proceedings of the Annual International Conference on Mobile Computing and Networking, MOBICOM*, 2000, ss. 243–254.
- [76] G. He, “Destination-sequenced distance vector (DSDV) protocol”, *Networking Laboratory, Helsinki University of Technology*, c. 135, ss. 1–9, 2002.
- [77] M. Nagaraju, R. K. Rao, ve K. Vijayalakshmi, “A Study and Comparison of Wireless Routing Protocols Performance Parameters”, *International Journal of Computer Technology & Applications*, c. 8, sayı 6, ss. 640–649, 2017.
- [78] L. A. Hassnawi, R. B. Ahmad, A. Yahya, S. A. Aljunid, ve M. Elshaikh, “Performance analysis of various routing protocols for motorway surveillance system cameras’ network”, *International Journal of Computer Science*, c. 9, sayı 2, ss. 369–373, 2012.
- [79] D. Ghosh ve P. Thakur, “Comparison and Analysis of Pro-Active Routing Protocols in MANET”, *International Journal of Engineering Research and*

- Applications*, c. 3, sayı 4, ss. 2176–2181, 2013.
- [80] K. Habak, M. Abdelatif, H. Hagrass, K. Rizec, ve M. Youssef, “A location-aided routing protocol for cognitive radio networks”, *2013 International Conference on Computing, Networking and Communications, ICNC 2013*, IEEE, 2013, ss. 729–733.
- [81] A. Setiabudi, A. A. Pratiwi, Ardiansyah, D. Perdana, ve R. F. Sari, “Performance comparison of GPSR and ZRP routing protocols in VANET environment”, *Proceedings - 2016 IEEE Region 10 Symposium, TENSYP 2016*, IEEE, 2016, ss. 42–47.
- [82] A. K. Gupta, H. Sadawarti, ve A. K. Verma, “Performance Analysis of MANET Routing Protocols in Different Mobility Models”, *International Journal of Information Technology and Computer Science*, c. 5, sayı 6, ss. 73–82, 2013.
- [83] P. Piyalikar, P. Kar, ve M. K. D. Barma, “Forecast Weighted Clustering in MANET”, *Procedia Computer Science*, c. 89, ss. 253–260, 2016.
- [84] D. Ergenç, L. Eksert, ve E. Onur, “Dependability-based clustering in mobile ad-hoc networks”, *Ad Hoc Networks*, c. 93, s. 101926, 2019.
- [85] R. Popli, M. Sethi, I. Kansal, A. Garg, ve N. Goyal, “Machine Learning Based Security Solutions in MANETs: State of the art approaches”, *Journal of Physics: Conference Series*, 2021. doi: 10.1088/1742-6596/1950/1/012070.
- [86] S. Singh, N. Rajpal, ve A. Sharma, “Address allocation for MANET merge and partition using cluster based routing”, *Springerplus*, c. 3, sayı 1, ss. 1–13, 2014.
- [87] V. V. Ziyadinov ve M. V. Tereshonok, “Analytical Survey on MANET and VANET Clusterisation Algorithms”, *2020 Systems of Signal Synchronization, Generating and Processing in Telecommunications, SYNCHROINFO 2020*, IEEE, 2020, ss. 1–5.
- [88] C. R. Lin ve M. Gerla, “Adaptive clustering for mobile wireless networks”, *EEE Journal on Selected areas in Communications*, c. 15, sayı 7, ss. 1265–1275, 1997.
- [89] G. Chen, F. G. Nocetti, J. S. Gonzalez, ve I. Stojmenovic, “Connectivity based k-hop clustering in wireless networks”, *Proceedings of the Annual Hawaii International Conference on System Sciences*, IEEE, 2002, ss. 2450–2459.

- [90] G. H. Alshammri, “Enhancing wireless sensor network lifespan and efficiency through improved cluster head selection using improved squirrel search algorithm”, *Artificial Intelligence Review*, c. 58, sayı 3, s. 79, 2025.
- [91] S. Çınaroğlu ve H. Bulut, “New initialization approaches for the k-means and particle swarm optimization based clustering algorithms”, *Journal of the Faculty of Engineering and Architecture of Gazi University*, c. 33, sayı 2, ss. 413–423, 2018.
- [92] C. A. Coello Coello ve M. S. Lechuga, “MOPSO: A proposal for multiple objective particle swarm optimization”, *Proceedings of the 2002 Congress on Evolutionary Computation, CEC 2002*, IEEE, 2002, ss. 1051–1056.
- [93] F. Aadil, K. B. Bajwa, S. Khan, N. M. Chaudary, ve A. Akram, “Caconet: Ant colony optimization (ACO) based clustering algorithm for VANET”, *PLoS One*, c. 11, sayı 5, 2016.
- [94] F. Barani ve S. Gerami, “ManetSVM: Dynamic anomaly detection using one-class support vector machine in MANETs”, *2013 10th International ISC Conference on Information Security and Cryptology, ISCISC 2013*, IEEE, 2013, ss. 1–6.
- [95] O. Kaynar, H. Arslan, Y. Görmez, ve Y. E. Işık, “Makine öğrenmesi ve öznelik seçim yöntemleriyle saldırı tespiti”, *Bilişim Teknolojileri Dergisi*, c. 11, sayı 2, ss. 175–185, 2018.
- [96] S. Pandey ve V. Singh, “Blackhole Attack Detection Using Machine Learning Approach on MANET”, *Proceedings of the International Conference on Electronics and Sustainable Communication Systems, ICESc 2020*, IEEE, 2020, ss. 797–802.
- [97] A. S. Parihar, S. K. Chakraborty, A. Sharma, ve M. Saini, “A comparative study and proposal of a novel distributed mutual exclusion in UAV assisted flying ad hoc network using density-based clustering scheme”, *Wireless Networks*, c. 29, sayı 6, ss. 2635–2648, 2023.
- [98] M. Srinivas ve M. R. Patnaik, “Clustering with a high-performance secure routing protocol for mobile ad hoc networks”, *The Journal of Supercomputing*, c. 78, sayı 6, ss. 8830–8851, 2022.

- [99] M. Ester, H. P. Kriegel, J. Sander, ve X. Xu, “A Density-Based Algorithm for Discovering Clusters A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise”, *Proceedings - 2nd International Conference on Knowledge Discovery and Data Mining, KDD 1996*, 1996, ss. 226–231.
- [100] S. S. Muratchaev, A. S. Volkov, V. S. Martynov, ve I. A. Zhuravlev, “Application of Clustering Methods in MANET”, *Proceedings of the 2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering, EICoN Rus 2020*, IEEE, 2020, ss. 1711–1714.
- [101] E. Weingärtner, H. Vom Lehn, ve K. Wehrle, “A performance comparison of recent network simulators”, *IEEE International Conference on Communications*, IEEE, 2009, ss. 1–5.
- [102] F. Irani, “VSIM: a new simulation and performance evaluation tool for MANET and VANET”, *International Journal of Information Technology*, ss. 1–17, 2024.
- [103] F. Hamza ve S. M. C. Vigila, “An energy-efficient cluster head selection in MANETs using emperor penguin optimization fuzzy genetic algorithm”, *Proceedings of International Conference on Recent Trends in Computing: ICRTC 2022*, 2023, ss. 453–468.
- [104] Z. Xing, L. Gruenwald, ve K. K. Phang, “A robust clustering algorithm for mobile Ad-hoc networks”, *Next Generation Mobile Networks and Ubiquitous Computing*, IGI Global, 2011, ss. 187–200.
- [105] M. Shayesteh ve N. Karimi, “An innovative clustering algorithm for MANETs based on cluster stability”, *International Journal of Modeling and Optimization*, c. 2, sayı 3, ss. 239–244, 2012.
- [106] T. Parameswaran, C. Palanisamy, A. Paul, ve S. Rho, “A Topology Control Method with Prediction of Lossy Links in Cluster Based Mobile Ad-Hoc Networks”, *International Journal of Control and Automation*, c. 8, sayı 12, ss. 403–418, 2015.
- [107] A. A. Mahamune ve M. M. Chandane, “Trust-based co-operative routing for secure communication in mobile ad hoc networks”, *Digital Communications and Networks*, c. 10, sayı 4, ss. 1079–1087, 2024.

- [108] K. S. Prasanna ve B. Ramesh, “Multiobjective Secure Trust Aware Redundant Array Shifting Encryption and Clustering Based Routing in Mobile Ad Hoc Networks”, *International Journal of Communication Systems*, ss. e6074, 2024.
- [109] B. Kheradmand, A. Ghaffari, F. S. Gharehchopogh, ve M. Masdari, “Cluster-Based Routing Schema Using Harris Hawks Optimization in the Vehicular Ad Hoc Networks”, *Wireless Communications and Mobile Computing*, c. 2022, 2022.



ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Kudret YILMAZ

Yabancı Dili : İngilizce

ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Doktora	Bilgisayar Müh.	Düzce Üniversitesi	2025
Y. Lisans	Bilgisayar Müh.	Düzce Üniversitesi	2019
Lisans	Bilgisayar Müh.	Azerbaycan Teknik Üniversitesi	2008

TEZ YAYINLARI

- [1] K. Yılmaz, R. Kara, ve F. Katircioglu, “Energy-Efficient Hybrid Adaptive Clustering for Dynamic Manets”, IEEE Access, 2025, doi: 10.1109/ACCESS.2025.3552232.
- [2] K. Yılmaz ve R. Kara, “MANET’te Küme Başı Seçimi ve Kümeleme Algoritmalarının Karşılaştırılması”, 11th International Vocational Schools Symposium, ss. 24-30, 2023.
- [3] K. Yılmaz ve R. Kara, “Manet’de Ağırlık Tabanlı Hibrit Dinamik Kümeleme Yöntemi”, 3rd International Congress On Contemporary Scientific Research, ss 510-520, 2023.