

T.C.  
SAKARYA ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

**BÜYÜK ÖLÇEKLİ KABLOSUZ ALGILAYICI AĞLAR  
İÇİN OĞUL ZEKASI TABANLI YENİ BİR  
YÖNLENDİRME ALGORİTMASI TASARIMI**

**DOKTORA TEZİ**

**Fatih ÇELİK**

**Enstitü Anabilim Dalı : ELEKTRONİK VE BİLGİSAYAR EĞİTİMİ**

**Tez Danışmanı : Doç. Dr. Ahmet ZENGİN**

**Eylül 2012**

T.C.  
SAKARYA ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

BÜYÜK ÖLÇEKLİ KABLOSUZ ALGILAYICI AĞLAR  
İÇİN OĞUL ZEKASI TABANLI YENİ BİR  
YÖNLENDİRME ALGORİTMASI TASARIMI

DOKTORA TEZİ

Fatih ÇELİK

Enstitü Anabilim Dalı : ELEKTRONİK VE BİLGİSAYAR EĞİTİMİ

Bu tez 07/09/2012 tarihinde aşağıdaki jüri tarafından Oybirliği ile kabul edilmiştir.



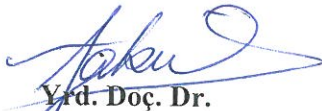
Prof. Dr.  
Hüseyin EKİZ  
Jüri Başkanı



Doç. Dr.  
Ahmet ZENGİN  
Üye



Doç. Dr.  
Resul KARA  
Üye



Yrd. Doç. Dr.  
Mehmet YAKUT  
Üye



Yrd. Doç. Dr.  
İhsan PEHLİVAN  
Üye

## TEŐEKKÜR

Eđitimim süresince olduđu gibi bu doktora alıőmasında da bana destek olan, hocam ve danışmanım Do. Dr. Ahmet ZENGİN'e teőekkür ederim. alıőmam boyunca desteklerini esirgemeyen Yrd. Do Dr. Sinan TUNCEL, Dr. Bülent OBANOĐLU ve Elektronik-Bilgisayar Eđitimi bölümündeki arkadaşlarıma teőekkür ederim. Bunun yanında alıőmam boyunca maddi ve manevi desteklerini aldıđım, moral motivasyonlarımı her zaman yanımda bulduđum eőime ve aileme teőekkür ederim.

# İÇİNDEKİLER

TEŞEKKÜR.....	ii
İÇİNDEKİLER .....	iii
SİMGELER VE KISALTMALAR LİSTESİ.....	ix
ŞEKİLLER LİSTESİ .....	xi
TABLolar LİSTESİ.....	xiv
ÖZET.....	xv
SUMMARY.....	xvi
BÖLÜM 1.	
GİRİŞ.....	1
1.1. Problem Tanımı .....	1
1.2. Literatür Taraması .....	4
1.3. Tezin Amacı.....	7
1.4. Tezin Kapsamı.....	8
1.5. Tez Çalışması ve Bilime Katkısı.....	9
1.6. Tez Düzeni.....	10
BÖLÜM 2.	
KABLOSUZ ALGILAYICI AĞLAR.....	12
2.1. Giriş.....	12
2.2. Kablosuz Algılayıcılar.....	12
2.3. Kablosuz Algılayıcılar İçin Geliştirilmiş İşletim Sistemleri.....	13
2.4. IEEE 802.15.4/ZigBee Kablosuz Algılayıcı Ağ İletişim Protokolü.	14
2.5. Kablosuz Algılayıcı Ağların Kullanım Alanları.....	15
2.5.1. Çevresel uygulamalar .....	15
2.5.2. Sağlık uygulamaları.....	15
2.5.3. Ticari uygulamalar .....	15

2.5.4. Askeri uygulamalar .....	16
2.6. Kablosuz Algılayıcı Ağ Yönlendirme Protokolleri.....	16
2.7. Yönlendirme Protokolleri Tasarım Kriterleri.....	17
2.7.1. Enerji tüketimi.....	17
2.7.2. Ölçeklenebilirlik.....	17
2.7.3. Veri toplama.....	18
2.7.4. Ağ ömrü.....	18
2.7.5. Hata toleransı.....	18
2.7.6. Paket teslim gecikmesi.....	18
2.7.7. Yerleşim.....	18
2.7.8. Servis kalitesi.....	19
2.7.9. Veri teslim modelleri.....	19
2.7.10. Operasyon çevresi.....	19
2.8. Oğul Zekası Tabanlı Yönlendirme Protokolleri.....	19
2.8.1. Karınca kolonisi temelli (ACO) yönlendirme protokolleri....	19
2.8.1.1. Enerji etkili karınca temelli yönlendirme (EEABR)...	20
2.8.1.2. Algılayıcı-Güdümlü ve Maliyet-Farkındalıklı Yönlendirme (SC-ANT).....	21
2.8.1.3. Akıcı ileri karınca yönlendirme (FF-ANT).....	21
2.8.1.4. Akıcı taşıma karınca yönlendirme (FP-ANT).....	21
2.8.1.5. Karınca Kolonisi Yer Farkındalıklı Yönlendirme (ACLR).....	21
2.8.1.6. T-ANT kablosuz algılayıcı ağ protokolü .....	22
2.8.1.7. Ant-chain kablosuz algılayıcı ağ protokolü.....	22
2.8.1.8. Kraliçe karınca farkındalıklı yönlendirme (QAAB)...	23
2.8.1.9. Karınca tabanlı minimum veri birleştirme ağacı (MADFT).....	24
2.8.1.10. Ant-0, Ant-1 ve Ant-2 kablosuz algılayıcı ağ protokolleri	24
2.8.1.11. E&D ANTS kablosuz algılayıcı ağ protokolü .....	24
2.8.1.12. Kendi kendine uyum sağlayan çoklu-yol yönlendirme .....	25
2.8.1.13. Ant Colony kablosuz algılayıcı ağ protokolü .....	26

2.8.1.14. AR, IAR kablosuz algılayıcı ağ protokolü.....	26
2.8.1.15. Kablosuz algılayıcı ağlar için karınca kolonisi temelli bir yönlendirme .....	26
2.9.2. Bal Arısı kolonisi tabanlı yönlendirme protokolleri.....	27
2.9.2.1. Beesensor... kablosuz algılayıcı ağ protokolü.....	27
2.9.3. Küf mantarı tabanlı yönlendirme protokolleri.....	28
2.9.3.1. Kablosuz algılayıcı ağlarda çoklu-hedef için oğul zekası tabanlı yönlendirme protokolü.....	28
2.10. Kümeleme Tabanlı Yönlendirme .....	29
2.10.1. Kümeleme tabanlı yönlendirme parametreleri.....	29
2.10.2. Kümeleme tabanlı yönlendirme protokolleri.....	31
2.10.2.1. Bağlantılı kümeleme algoritması (LCA) .....	31
2.10.2.2. Rasgele çekişme tabanlı kümeleme (RCC).....	32
2.10.2.3. CLUBS kümeleme protokolü.....	32
2.10.2.4. Hiyerarşik kontrollü kümeleme.....	33
2.10.2.5. GS <sup>3</sup> kümeleme protokolü .....	33
2.10.2.6. Enerji etkili hiyerarşik kümeleme (EEHC).....	34
2.10.2.7. Düşük enerji uyumlu kümeleme hiyerarşisi (LEACH).....	35
2.10.2.8. Hızlı yerel kümeleme (FLOC) .....	35

### BÖLÜM 3.

AYRIK OLAYLI MODELLEME ve BENZETİM .....	37
3.1. Giriş.....	37
3.2. Ayrik Olaylı Sistemler ve DEVS Modelleme ve Benzetim Teorisi.	37
3.4.1. Ayrik olaylı modelleme yaklaşımı.....	38
3.4.2. Ayrik olaylı sistemlerde kullanılan terimler.....	39
3.4.3. Ayrik olaylı benzetim stratejileri .....	41
3.4.3.1. Olay zamanlama benzetim stratejisi.....	41
3.4.3.2. Aktivite tarama benzetim stratejisi.....	43
3.4.3.3. Süreç etkileşim benzetim stratejisi.....	44
3.4.4. Ayrik olaylı benzetim stratejileri arasındaki ilişkiler.....	45
3.5. Atomik DEVS Modelleme Yaklaşımı.....	46

3.6. Birleşik DEVS Modelleme Yaklaşımı.....	48
3.7. Hiyerarşik Model Tasarımı: DEVS Birleşim Çerçevesi.....	50
3.8. Modelleme ve Benzetim Araçları.....	51
3.8.1. Ağ benzeticileri ve problemleri.....	51
3.8.2. Benzeticiler ile ilgili problemlere çözüm önerisi.....	52
<b>BÖLÜM 4.</b>	
<b>BEEWS: OĞUL ZEKASI TABANLI YENİ BİR ÖLÇEKLENEBİLİR</b>	
<b>YÖNLENDİRME PROTOKOLÜ .....</b>	
4.1. Giriş.....	56
4.2. KAA Model Sentezi ve BeeWS Algoritması.....	56
4.2.1. Veri oluşturucu.....	58
4.2.2. Gözcüler.....	58
4.2.3. Veri taşıyıcı.....	62
4.3. BeeWS Mimarisi.....	62
4.3.1. Giriş bölümü.....	62
4.3.2. Değerlendirme bölümü.....	63
4.3.3. Dans bölümü.....	65
4.3.4. Paketleme bölümü.....	66
4.3.5 Yönlendime tablosu.....	67
4.3.6. BeeWS paket yapıları.....	68
4.3.6.1. İleri yönlü gözcü mesajı.....	68
4.3.6.2. Geri yönlü gözcü mesajı.....	70
4.3.7. Kümeleme yöntemi.....	71
4.3.7.1. Kümebaşı seçimi.....	73
4.3.7.1. Kümeleme yöntemi çalışması.....	73
<b>BÖLÜM 5.</b>	
<b>DEVS-SENSOR MODELLEME ve BENZETİM ORTAMININ</b>	
<b>GELİŞTİRİLMESİ.....</b>	
5.1. Giriş.....	75
5.2. Java Programlama Dili ve DEVS-Suite ortamı.....	75
5.3. DEVS-Sensor Modelleme Süreci.....	76

5.4. KAA Modelleme Yaklaşımı ve Bileşenlerin Tasarımı.....	77
5.4.1. Kablosuz algılayıcı atomik modeli.....	81
5.4.2. Yerleşim atomik modeli.....	83
5.4.3. Ağ paketleri.....	84
5.4.4. Yönlendirme tabloları.....	87
5.4.5. Birleşik (Coupled) model.....	88
5.4.6. Ağların ölçeklenmesi.....	89
5.5. Topoloji Üretici.....	90
5.6. Görsel Takip.....	93
5.7. Benzetim Onaylama ve Geçerleme.....	94
BÖLÜM 6.	
DENEYLER ve SONUÇLAR .....	96
6.1. Giriş.....	96
6.2. Deneysel Çerçeve ve Benzetim Ortamı.....	97
6.2.1. Değerlendirme ölçütleri.....	99
6.3. Benzetim Sonuçları.....	100
6.3.1. BeeWS ile BeeSensor, FP-Ant, EEABR ve AODV karşılaştırması.....	100
6.3.1.1. Paket teslim oranı.....	100
6.3.1.2. Ortalama gecikme.....	100
6.3.1.3. Enerji tüketimi.....	102
6.3.2. BeeWS için kümelemesiz ve kümelemeli uygulamaların karşılaştırılması.....	102
6.3.2.1. Paket teslim oranı.....	103
6.3.2.2. Ortalama gecikme.....	104
6.3.2.3. Enerji tüketimi.....	104
6.3.3. Büyük ölçekli KAA' da kümeleme.....	105
6.3.3.1. BeeWS ve diğer kümeleme yöntemlerinin karşılaştırılması.....	106
BÖLÜM 6.	
SONUÇLAR ve DEĞERLENDİRME .....	108



7.1. Sonular.....	108
7.2. Tartıřma ve neriler.....	110
KAYNAKLAR.....	111
EKLER.....	118
ZGEMİř.....	122



## SİMGELER VE KISALTMALAR LİSTESİ

ACO	: Karınca kolonisi temelli
ACLR	: Yer farkındalıklı karınca kolonisi yönlendirme
AI	: Yapay Zeka (Artificial Intelligence)
AM	: Atomik model
AODV	: Ad Hoc İsteğe Bağlı Uzaklık Vektörü Yönlendirme Protokolü
BRITE	: The Boston University Representative Internet Topology Generator
CM	: Birleşik Model (Coupled Model)
DEVS	: Ayrık Olaylı Sistem Tanımı (Discrete Event System Specification)
EEABR	: Enerji etkili karınca temelli yönlendirme
EECH	: Enerji etkili hiyerarşik kümeleme
EESH	: Enerji etkili güçlü kümeleme
E&D ANT	: Enerji ve gecikme temelli yönlendirme
EF	: Deneysel Çerçeve (Experimental Frame)
FF	: Akıcı-ileri yönlendirme
FIFO	: İlk Giren İlk Çıkar (First-In, First-Out)
FLOC	: Hızlı yerel kümeleme
FP	: Akıcı-taşıma yönlendirme
GloMoSim	: Global Mobile Information System Simulator
GUI	: Grafiksel kullanıcı arayüzü (Graphical User Interface)
h	: Yükseklik
ID	: Kimlik
IEEE	: Elektrik Elektronik Mühendisleri Enstitüsü (Institute of Electrical and Electronical Engineers)
IETF	: İnternet Engineering Task Force
I/O	: Giriş ve Çıkış
IP	: İnternet Protokolü
JNS	: Java Network Simulator

JVM	: Java Sanal Mekanizması (Java Virtual Machine)
KAA	: Kablosuz Algılayıcı Ağ
LAN	: Yerel alan ağı (Local Area Network)
LCA	: Bağlantı kümeleme
LEACH	: Düşük enerji uyumlu kümeleme
MAC	: Veribağı katmanı
M&S	: Modelleme ve Benzetim
n	: Düğüm sayısı
Ns-2	: Network Simulator 2
Ns-3	: Network Simulator 3
OMNET++	: Objective Modular Network Test-bed in C++
OS	: İşletim Sistemi (Operating System)
QAAB	: Kraliçe karınca farkındalıklı yönlendirme
QoS	: Hizmet Kalitesi (Quality of Service)
QSS	: Nitel Sistem Tanımlama (Qualitative System Specification)
PDNS	: Paralel/Dağıtık ağ benzeticisi
r	: Kapsama yarı çapı
RCC	: Rasgele çekişme tabanlı kümeleme
RX	: Veri alma
SC	: Algılayıcı – sürücülü yönlendirme
SSFnet	: Ölçeklenebilir benzetim çerçevesi (Scalable Simulation Framework)
TCP	: İletim Kontrol Protokolü (Transmission Control Protocol)
TX	: Veri gönderme
TTL	: Yaşama Zamanı (Time to Live)
UDP	: Kullanıcı Veribloğu İletişim Kuralları (User Datagram Protocol)
w	: Genişlik,
WLAN	: Kablosuz yerel alan ağı
VV&A	: Verification, Validation & Accreditation

## ŞEKİLLER LİSTESİ

Şekil 2.1.	Çeşitli boyuttaki Kablosuz Algılayıcılar .....	13
Şekil 2.2.	Kablosuz Algılayıcı birimleri .....	13
Şekil 2.3.	Kablosuz algılayıcı ağ örneği.....	17
Şekil 2.4.	Oğul zekası tabanlı yönlendirme algoritmalarının sınıflandırılması.....	20
Şekil 2.5.	QAAB protokolü ızgara yapısı.....	23
Şekil 2.6.	G düğümünün araştırma bölgesinin genişletmesi.....	25
Şekil 2.7.	Hiyerarşik kümeleme.....	33
Şekil 2.8.	Bir hücrenin merkezine büyük düğüm yerleştirilerek oluşturulan sanal altıgen hücreler.....	34
Şekil 3.1.	Kasiyer / kuruk sisteminin davranış örneği.....	39
Şekil 3.2.	Olay zamanlama benzetim çekirdeği.....	43
Şekil 3.3.	Aktivite tarama benzetim stratejisi.....	44
Şekil 3.4.	Benzetim stratejilerinin sınıflandırılması.....	45
Şekil 3.5.	DEVS işleyiş mekanizması.....	47
Şekil 3.6.	Ayrık olaylı sistemde giriş, durum, geçen süre ve çıkışlar.....	48
Şekil 3.7.	Birleşik DEVS yaklaşımında bağlantılar.....	49
Şekil 3.8.	DEVS birleşik modeli .....	50
Şekil 3.9.	DEVS-Suite benzetim altyapısı.....	54
Şekil 4.1.	Bir kovanın yapısı.....	59
Şekil 4.2.	BeeWS mimarisi.....	63
Şekil 4.3.	Giriş bölümü .....	64
Şekil 4.4.	Değerlendirme bölümü.....	65
Şekil 4.5.	Dans bölümü.....	66
Şekil 4.6.	Paketleme bölümü .....	67
Şekil 4.7.	BeeWS mesaj gönderme süreci .....	69
Şekil 4.8.	İleri yönlü gözcü mesaj paketi içeriği.....	69

Şekil 4.9.	İleri yönlü gözcü mesaj iletimi.....	70
Şekil 4.10.	İleri yönlü gözcü mesaj iletimi.....	70
Şekil 4.11.	İleri yönlü gözcü yönlendirme tablo oluşumu.....	71
Şekil 4.12.	Geri yönlü gözcü mesaj bileşenleri.....	71
Şekil 4.13.	Geri yönlü gözcü mesaj iletimi yayılımı .....	72
Şekil 4.14.	Geri yönlü gözcü yönlendirme tablosu .....	72
Şekil 4.15.	HELLO paketi içeriği .....	73
Şekil 4.16.	Kümeleme yöntemi .....	74
Şekil 4.17.	BeeWS protokol mesajları .....	74
Şekil 5.1.	KAA modelleme ve benzetim süreci .....	78
Şekil 5.2.	KAA'ların model bileşenleri ile DEVS-Sensor' in kavramsal görünümü.....	79
Şekil 5.3.	DEVS-Sensor ile ağ katmanlarının eşleşmesi .....	80
Şekil 5.4.	DEVS-Suite ve DEVS-Sensor benzeticisi kavramsal modeli gösterimi .....	81
Şekil 5.5.	Algılayıcı düğümün kavramsal modeli.....	82
Şekil 5.6.	Düğüm atomik modelinin ekran çıktısı .....	83
Şekil 5.7.	Yerleşim atomik modelinin kavramsal modeli (a) ve ekran çıktısı (b).....	83
Şekil 5.8.	DEVS-Sensor sınıflar ve yapısal ilişkileri .....	85
Şekil 5.9.	DEVS-Sensor birleşik model sınıfları.....	86
Şekil 5.10.	Protokol başlığı ve veri içeren veri paketlerinin gösterimi .....	86
Şekil 5.11.	Bir düğüme ait sadeleştirilmiş durum diyagramı .....	87
Şekil 5.12.	Bir yönlendirme tablosu örneği .....	88
Şekil 5.13.	Bir yönlendirme tablosunun DEVS-Suite altında ekran görünümü.....	88
Şekil 5.14.	Düğümlerin iletişim kurması .....	89
Şekil 5.15.	DEVS-Sensor ortamında bir ağ modeli birleşimi .....	90
Şekil 5.16.	DEVS-Sensor benzetim ortamında kümeleme .....	91
Şekil 5.17.	DEVS-Sensor BRITE topoloji üretici ekran görüntüsü .....	93
Şekil 5.18.	DEVS-Sensor görselleştirme ekranında 350X 350 lük bir alanda topoloji dağılımı .....	94

Şekil 5.19.	DEVS modeli süreci ve deneysel çerçevenin doğrulanması ve geçerlemesi .....	95
Şekil 6.1.	BeeWS modelinin DEVS-Sensor benzetim ortamında çalışma grafiği .....	97
Şekil 6.2.	Paket teslim oranları karşılaştırması .....	101
Şekil 6.3.	Ortalama gecikme karşılaştırması .....	101
Şekil 6.4.	Enerji tüketimi karşılaştırması .....	102
Şekil 6.5.	Kümelemeli ve kümelemesiz yöntemlerin paket teslim oranları...	103
Şekil 6.6.	Kümelemeli yöntemin ortalama gecikmeye etkisi .....	104
Şekil 6.7.	Kümelemenin enerji tüketimine etkisi .....	105
Şekil 6.8.	Ortalama kümebaşı sayısı değişimi .....	106
Şekil 6.9.	Ortalama küme sayısı karşılaştırma .....	107

## TABLolar LİSTESİ

Tablo 2.1.	IEEE 802.15.4 Radyo frekansları ve veri aktarım hızları .....	15
Tablo 2.2.	Yönlendirme protokollerinin karşılaştırılması.....	30
Tablo 2.3.	Kümeleme yönlendirme algoritmaları sınıflandırması.....	36
Tablo 3.1.	Ağ benzetimlerinin karşılaştırılması.....	53
Tablo 4.1.	KAA ile bal arısı kolonileri arasındaki benzerlik .....	58
Tablo 4.2.	Gözcü oluşturma algoritması .....	60
Tablo 4.3.	Gözcülerin çalışma algoritması .....	61
Tablo 4.4.	Yönlendirme tablosu güncelleme algoritması .....	61
Tablo 6.1.	MICAz parametreleri .....	98
Tablo 6.2.	Benzetim parametre ve formülleri.....	99

## ÖZET

Anahtar Kelimeler: KAA, Oğul Zekası, Yönlendirme, Modelleme ve Benzetim, DEVS, Ölçeklenebilirlik, Kümeleme

Kablosuz algılayıcı ağlar için ölçeklenebilir, enerji etkin ve performansa dayalı yönlendirme protokollerinin tasarımı son yıllarda oldukça ilgilenilen konular arasındadır. Geliştirilen yönlendirme protokolleri incelendiğinde kullanılan yöntemler arasında sosyal böceklerden ilham alınarak geliştirilen oğul zekası teknikleri diğer tekniklere göre öne çıkmaktadır. Bunun nedeni kablosuz algılayıcıların sosyal böceklerde olduğu gibi kendi kendine örgütlenebilme, iş bölümü yapabilme, çok merkezlilik, esneklik, ölçeklenebilirlik yeteneklerine sahip olmasıdır.

Benzetim araçları (Ns-2, Ns-3, OPNET, Omnet++, QualNet vb.) küçük ağları çalıştırmak için ideal ortamlar oluştururken, günümüzde sürekli büyüyen ağ sistemlerini modellemede ve değişken yapıları ağ sistemlerinin başarımını test etmede yetersiz kalmaktadırlar. Aynı zamanda benzetim araçları temelde KAA'ların ihtiyaçlarına göre tasarlanmadığından gerçek dünya verilerine uygun KAA topolojileri oluşturmada sıkıntılar yaşanmaktadır. Bu da KAA'ların ihtiyaçlarını karşılayacak büyük ölçekleri destekleyecek yeni modelleme ve benzetim araçlarının geliştirilmesini gerekli hale getirmektedir.

Hazırlanan bu tez çalışmasında, yönlendirme protokollerindeki paket teslimi, gecikme, enerji ve küme sayısı kriterleri dikkate alınarak uyum yeteneği, ölçeklenebilirlik, beka ve batarya ömrü vb. problemlerine çözüm getirmek amacıyla;

1. KAA'lar için oğul zekası tabanlı yeni bir yönlendirme algoritması geliştirilmiştir. Geliştirilen bu algoritma BeeWS olarak adlandırılmıştır.
2. Büyük ölçekli KAA'lar için yeni bir kümeleme algoritması geliştirilmiştir.
3. Geliştirilen protokolleri modellemek ve benzetimini yapmak amacıyla büyük ölçekleri destekleyen, kolay kullanımlı, KAA topolojilerinin otomatik oluşturulabilen DEVS-Sensor modelleme ve benzetim aracı geliştirilmiştir.

Geliştirilen protokolün üstünlüklerini göstermek amacıyla farklı ölçeklerden oluşan ağlar modellendi. Modellenen ağlar farklı trafik yükleri altında çalıştırılarak, geliştirilen BeeWS algoritması literatürden seçilen önemli algoritmalarla karşılaştırılarak başarımı incelendi. Gerçekleştirilen uygulamalarda, geliştirilen yönlendirme algoritmasının belirlenen problemlere çözüm getirdiği gözlemlendi.



# **DESIGN AND IMPLEMENTATION OF A NEW SWARM-BASED ROUTING ALGORITHM FOR LARGE-SCALE WIRELESS SENSOR NETWORKS**

## **SUMMARY**

Keywords: WSN, Swarm Intelligence, Routing, Modeling & Simulation, DEVS, Scalability, Clustering

In the last decade, design of scalable, energy-efficient and high performance routing protocols for wireless sensor networks (WSN) is attracted by many researchers. In the analysis of routing protocols, among the methods inspired by social insects are more preferred than other techniques. This is because the social insects have similar properties with wireless sensor networks such as self-organization, division of labor, decentralization, flexibility, scalability and fault tolerance.

Simulation tools such as ns-2, ns-3, Opnet, Omnet++, QualNet and Glomosim are ideal environments when creating and running small-scale networks. However, they are incapable of modeling today's growing networks and variable structure systems in order to test the performance of such systems. At the same time, these simulation tools are not designed according to the requirements of WSN's which those basically correspond to real-world problems are being experienced when creating WSN topologies. Hence, there need for developing new swarm based routing protocols to meet the needs of WSN's for supporting large scale and highly flexible systems.

In this thesis, in order to bring solutions to the above problems in WSN's, considering the criteria such as packet delivery, latency, energy, adaptation ability, number of clusters, scalability, survivability and battery life, following transactions are performed:

1. A new routing algorithm for WSN based on honeybee scout-recruit system called BeeWS was developed.
2. A new clustering algorithm was developed for large-scale WSN's.
3. In order to model and simulate developed protocols, scalable, easy-to-use and deploy, capable of automated topology generation simulation tool called DEVS-Sensor was developed.

# BÖLÜM 1. GİRİŞ

## 1.1. Problem Tanımı

Kablosuz iletişimin son yıllarda hızla gelişmesiyle beraber bu alanda yeni sistemler ve teknolojiler ortaya çıkmaktadır. Bu teknolojilerden biri Kablosuz Algılayıcı Ağ (KAA) teknolojisidir. 1996 yılında düşük güçlü entegre kablosuz mikro algılayıcıların (LWIMs) UCLA ve Rockwell Bilim Merkezi tarafından üretilmesiyle, kablosuz algılayıcılar ticari alanlarda kullanılmaya başlamıştır [1]. Kablosuz algılayıcılar, kapsama alanının artırılması ve enerji tüketiminin azaltılması amacıyla IEEE 802.15.4 teknolojisi ve bu teknolojinin belirlediği 2.4 Ghz radyo frekansını kullanmaktadır.

Karıncalar ve bal arısı gibi sosyal canlıların karmaşık kolektif davranışları ve çok merkezli yapıları kablosuz algılayıcı ağlar gibi dinamik sistemler ile benzerlikler taşımaktadır. Sosyal canlılar tek başlarına herhangi bir zeka ortaya koyamadıkları halde birlikte hareket ederek mükemmel bir sistem oluşturabilmektedirler. Bir lidere sahip olmamalarına rağmen sistemin devamlılığını sağlayabilmektedirler ve bu sayede sistem ölçeği ne kadar büyürse büyüsün problemsiz bir şekilde sistem işleyebilmektedir. Mesela arı kolonisinin birkaç binden kırk elli bine çıkması onların çalışmalarını etkilemez. Sistemde var olan canlılar enerji kaybını azaltmak amacıyla buldukları yiyeceklere en kısa yolu tespit ederek gitmektedirler. Birçok özellik oğul zekasının kablosuz algılayıcı ağlar ile ne kadar çok örtüştüğünü göstermektedir.

Kablosuz algılayıcı ağları önemli kılan başlıca özellikleri; buldukları yerde kendi kendilerine işlem yapabilme kabiliyetlerinin olması, kendi aralarında dış müdahale olmadan organize olabilmeleri, çevreye istenildiği şekilde yerleştirilebilmeleri, fiziki olarak çok küçük olmaları, kullanım kolaylığı ve esnekliği, uzaktan takip edilebilmeleri ve geniş çaplı kablosuz algılayıcı ağların düşük maliyet ve güç

tüketimi ile oluşturulmasında kullanılmalarıdır. Üzerlerindeki pillerin uygulamaya bağlı olarak uzun süreler algılayıcıları çalıştırabilmesi, algılayıcı ağları konum belirleme, veri toplama, olay tanımlama gibi alanlarda oldukça başarılı kılmiştir. Fakat bunun yanında KAA'lar belli başlı problemlere sahiptir.

Kablosuz algılayıcı ağlar da karşılaşılan başlıca problemler şunlardır:

Ölçeklenebilirlik [2]: Bir ortama yüzler hatta binlerce düğüm yerleştirilebilir. Uygulamaya bağlı olarak KAA milyonlarca düğüme ulaşabilir yada küçük bir ortama çok fazla yerleştirilerek yüksek yoğunluklu ağlar oluşturulabilir. Bundan dolayı, çok fazla düğüm üzerinde çalışabilecek veya yüksek yoğunluklu KAA'larda verimli çalışabilecek yeni algoritmalara ihtiyaç duyulmaktadır.

Uyum yeteneği [2]: Algılayıcı ağların düğüm ölümleri ya da hareketli senaryolarda düğümlerin hareket etmesiyle topolojileri değişime uğramaktadır. Algılayıcı düğümlerin yüzlerce hatta binlerce olduğu düşünüldüğünde değişimin ne kadar çok büyük olabileceği anlaşılmaktadır. Dolayısıyla topoloji değişimlerinden en az oranda etkilenen yönlendirme algoritmalarına ihtiyaç duyulmaktadır.

Sistemin bekası [2]: KAA sisteminde düğüm ölümlerinin gerçekleşmesi ve ağda çıkabilecek arızalar ağ topolojisinin değişmesine neden olmaktadır. Bu değişimler ağ sisteminin tamamen çökmesine neden olabilmektedir. KAA sistemlerinin kendi kendini yönetmesi nedeniyle bu probleme yönlendirme algoritmalarının çözüm getirerek sistemin devamlılığını sağlamaları gerekmektedir.

Batarya ömrü [2]: Kablosuz algılayıcılar uygulama alanlarına bırakılır ve bazen birkaç hafta bazen birkaç yıl onarıma tabi tutulmazlar. Bu da kablosuz algılayıcıların enerji ihtiyacını en az seviyede tutmalarını gerektirmektedir. Batarya ömrü kablosuz algılayıcıların en önemli problemi olduğu için yönlendirme algoritmaları enerji etkin yöntemlere odaklanmıştır.

Yukarıda sayılan yönlendirme problemlerinin yanısıra yönlendirme algoritmalarının test edileceği benzetim ortamları ile ilgili problemler de karşımıza çıkmaktadır.

Günümüzde çeşitli modelleme ve benzetim araçları ağ tasarımı için şirketler / akademik araştırma grupları tarafından planlanan amaca göre pratik veya eğitim amaçlı olarak kullanılmaktadır. Bu araçların işlevselliği, avantajları, süreçleri ve özellikleri açısından detaylı olarak incelenmesi gerekmesine rağmen hemen hemen tüm ağ benzetim araçlarının benzer yolla (bazen tamamlayıcı hedefler) çalıştığı açıktır [3].

İletişim ağlarını doğrudan modellemek için kullanılacak yazılımlar (OPNET, Ns-2, Ns-3, GloMoSim, OMNET++, JNS, vb.) yanında donanım bileşenlerini modellemek için kullanılacak araçlar (VHDL, Verilog, SystemC, vb.) bulunmaktadır [4,5]. En yaygın kullanılan ağ benzetimcileri olan Ns-2, Ns-3, OPNET, OMNET++, vb. benzetimciler küçük boyutlu ağları çalışmak için ideal platformlardır. Bu programları büyük ölçekli sistemlerin modellenmesi ve benzetiminde kullanmanın zor olmasının yanında, değişik teknolojilerin içine katıldığı sistemleri modelleme yeteneğine de sahip değildirler [6]. Ağ sistemlerinin boyutlarının süreç içerisinde üstel olarak artması nedeni ile statik topoloji üreten bu benzetimciler yetersiz kaldılar ve gelişen sistemlerin başarımını doğru bir şekilde test edemez / ölçemez duruma düştüler. Belirtilen eksikliklerin ortaya çıkmasının sebepleri olarak; basit benzetimcilerin yapısal sınırlamaları ve büyük ölçekli karmaşık yapıları ağların yetersiz bir şekilde analiz edilmesi sıralanabilir. Belirtilen kısıtlamaları / yetersizlikleri bir ölçüde karşılayabilen ve binlerce düğümü modelleyebilen GloMoSim, PDNS, vb. benzetimciler bulunsa da, bunların hiçbiri dinamik, gelişebilir, yeniden boyutlanabilir ve değişik trafik şartlarına uyarlanabilir (adaptif) bir ağı modellemeye uygun değildir [6-8]. Ayrıca, bu tür benzetimcilerin bileşenleri modüler ve hiyerarşik bir yapıda olmadığından bileşenlerin yeniden kullanımı, değişik uygulamalara uyarlanabilirliği ve hiyerarşik tasarımı zordur. Klasik benzetimcilerin çalıştırıldıkları bilgisayarlarda oldukça yüksek kaynak kullanım gereksinimleri, gelişmiş ve büyük uygulamaların meydana getirilmesini zorlaştırmaktadır.

Bu tez çalışmasında yukarıdaki problemleri ortadan kaldırmak için, sosyal canlılardan ilham alınarak bal arı kolonilerini temel alan oğul zekası tabanlı yeni bir yönlendirme algoritması ve bu algoritmayı test edebilmek için KAA'ların ihtiyaçlarına cevap verebilen bir modelleme ve benzetim aracı geliştirilmiştir.

## 1.2. Literatür Taraması

Yönlendirme protokollerinin hemen hemen tamamı enerjinin etkin kullanılmasına odaklanmıştır. Bunun yanında diğer problemlerde çözüm getirmeyi hedeflemişlerdir. Aşağıda literatürde yer alan yönlendirme protokolleri ve bunların çözüm getirdiği alanlar verilmiştir.

- Enerji etkili karınca temelli yönlendirme (EEABR) algoritması Camilo tarafından 2006 yılında karınca kolonisi tabanlı optimizasyon olarak kablosuz algılayıcı ağlar için geliştirilmiştir [9]. Enerji tüketimi, ağ ömrü ve hata toleransı konularında çözüm getirmiştir.
- SC-Ant, FF-Ant ve FP-Ant yönlendirme algoritmaları Zhang, Kuhn, Fromherz [10] tarafından 2004 yılında karınca kolonisi tabanlı optimizasyon olarak kablosuz algılayıcı ağlar için geliştirilmiştir. SC-Ant enerji tüketimi konusuna, FF-Ant paket teslim ortalama gecikmesi konusuna, FP-Ant veri teslim başarımlı oranı konusuna çözüm getirmiştir.
- Karınca Kolonisi Yer Farkındalıklı Yönlendirme (ACLR) algoritması Wang tarafından 2008 yılında geliştirilmiştir [11]. Enerji tüketimi ve ortalama paket teslim gecikmesi konularına çözüm getirmiştir.
- T-ANT yönlendirme algoritması Selvakennedy tarafından 2006 yılında geliştirilmiştir [12]. Enerji tüketimi, ölçeklenebilirlik, veri toplama, ağ ömrü konularına çözüm getirmiştir.
- Ant-chain yönlendirme algoritması Ding ve Liu tarafından 2005 yılında geliştirilmiştir [13]. Enerji tüketimi, veri toplama ve ağ ömrü konularına çözüm getirmiştir.

- Kraliçe karınca tabanlı yönlendirme (QAAB) algoritması Sun tarafından 2006 yılında geliştirilmiştir [14]. Ağ ömrü ve enerji tüketimi konularına çözüm getirmiştir.
- Karınca tabanlı minimum veri birleştirme ağacı (MADFT) yönlendirme algoritması Juan tarafından 2007 yılında geliştirilmiştir [15]. Enerji tüketimi konusuna çözüm getirmiştir.
- Ant-0, Ant-1 ve Ant-2 yönlendirme algoritmaları Liao tarafından 2007 yılında geliştirilmiştir [16]. Veri toplama ve enerji tüketimi konularına çözüm getirmiştir.
- E&D ANTS yönlendirme algoritması Wen tarafından 2008 yılında geliştirilmiştir [17]. Enerji tüketimi ve ortalama paket teslim gecikmesi konularına çözüm getirmiştir.
- Kendi kendine uyum sağlayan çoklu-yol yönlendirme algoritması Saleem tarafından 2009 yılında geliştirilmiştir [18]. Enerji tüketimi ve ortalama paket teslim gecikmesi konularına çözüm getirmiştir.
- Karınca kolonisi yönlendirme algoritması Chao tarafından 2008 yılında geliştirilmiştir [19]. Enerji tüketimi ve ortalama paket teslim gecikmesi konularına çözüm getirmiştir.
- AR ve IAR yönlendirme algoritmaları Aghaeil tarafından 2007 yılında geliştirilmiştir [20]. Enerji tüketimi ve ortalama paket teslim gecikmesi konularına çözüm getirmiştir.
- Kablosuz algılayıcı ağlar için karınca kolonisi temelli bir yönlendirme algoritması Okdem ve Karaboğa tarafından 2009 yılında geliştirilmiştir [21]. Enerji tüketimi konusuna çözüm getirmiştir.

- Beesensor yönlendirme algoritması Saleem ve Farooq tarafından 2007 yılında kablolu ağlar için geliştirilmiş olan BeeHive [22] yönlendirme protokolünden ilham alınarak bal arısı kolonileri yöntemi kullanılarak geliştirilmiştir [23]. Enerji tüketimi ve ortalama paket teslim gecikmesi konularına çözüm getirmiştir.
- Kablosuz algılayıcı ağlarda çoklu-hedef için oğul zekası tabanlı yönlendirme protokolü Paone tarafından 2009 yılında geliştirilmiştir [24]. Enerji tüketimi, hata toleransı konularına çözüm getirmiştir.

Yukarıda bahsedilen algoritmalar küçük ölçekli ağlarda uygulanmış fakat büyük ölçeklere ulaşan ağlarda test edilmemiştir. Bunun nedeni kümeleme yönteminin yukarıdaki algortimalara uygulanmamasıdır. Aşağıdaki bahsedilen algoritmalar ise kümeleme yöntemini kullanarak büyük ölçeklere çıkabilmişlerdir.

- Bağlantılı kümeleme algoritması (LCA) Baker ve Ephremides tarafından geliştirilmiştir [25,26]. Küme sayısı değişken olup kümeler bir atlamadan oluşur. Kümebaşı seçimi rasgele yapılmaktadır.
- Rasgele çekişme tabanlı kümeleme (RCC) algortması Xu ve Gerla [27] tarafından geliştirilmiştir. Küme sayısı değişken olup kümelerin atlama sayısı oluşturulan yapıya göre değişir. Kümebaşı seçimi rasgele yapılmaktadır.
- CLUBS kümeleme algoritması Nagpal ve Coore [28] tarafından geliştirilmiştir. Küme sayısı değişken olup kümeler iki atlamadan oluşur. Kümebaşı seçimi rasgele yapılmaktadır.
- Hiyerarşik kontrollü kümeleme algoritması Banerjee ve Khuller [29] tarafından geliştirilmiştir. Küme sayısı değişken olup kümelerin atlama sayısı oluşturulan yapıya göre değişir. Kümebaşı seçimi rasgele yapılmaktadır.

- GS<sup>3</sup> yönlendirme algoritması Zhang ve Arora [30] tarafından geliştirilmiştir. Küme sayısı ön tanımlı olarak belirlenir. Kümelerin atlama sayısı oluşturulan yapıya göre değişir. Kümebaşı seçimi ön tanımlı olarak yapılır.
- Enerji etkili hiyerarşik kümeleme (EEHC) algoritması Bandyopadhyay ve Coyle [31] tarafından geliştirilmiştir. Küme sayısı değişken olup kümelerin atlama sayısı oluşturulan yapıya göre değişir. Kümebaşı seçimi rasgele yapılmaktadır.
- Düşük enerji uyumlu kümeleme hiyerarşisi (LEACH) algoritması Heinzelman, Chandrakasan, Balakrishnan [32] tarafından geliştirilmiştir. Küme sayısı değişken olup kümeler bir atlamadan oluşur. Kümebaşı seçimi rasgele yapılmaktadır.
- Hızlı yerel kümeleme (FLOC) algoritması Demirbaş, Arora ve Mittal tarafından geliştirilmiştir [33]. Küme sayısı değişken olup kümeler iki atlamadan oluşur. Kümebaşı seçimi rasgele yapılmaktadır.

Literatürde taranan algoritmaların tamamı kablosuz algılayıcıların başlıca problemi olan batarya ömrüne çözüm getirmeye çalışmıştır. Bunun yanında diğer problemlere de çözüm getirmeye çalışmışlardır. Ölçeklenebilirlik problemine çözüm getirmeye çalışan algoritmalar ortalama beş yüz civarında düğüme ulaşabilmişler büyük bir kısmı daha yüksek düğümlere ulaşamamışlardır. Paket teslim oranı olarak yüksek oranda paket teslimi gerçekleştiren algoritma sayısı çok azdır. Algoritmaların kullandığı kontrol paketleri ortalama gecikmeleri artırmıştır.

### 1.3. Tezin Amacı

KAA'ların oluşumunda karşımıza çıkan uyum yeteneği, sistemin bekası, ölçeklenebilirlik ve batarya ömrü gibi problemlerin giderilmesinde veri akışının kontrolünü sağlayan yönlendirme protokollerinin önemi büyüktür. Bu problemlerin çözümü için geliştirilen yönlendirme algoritmalarının test ortamı olan modelleme ve benzetim araçlarının da KAA topolojileri üretmeye uygun ortamlar olması, algoritma



geliştirilirken KAA'larda yapılacak deęişikliklere anında cevap vermesi, büyük ölçekleri desteklemesi gerekmektedir.

Gerçekleştirilen tez çalışmasının amacı, KAA'ların problemlerini karşılamak üzere sosyal canlıların davranışlarından ilham alınarak geliştirilen oęul zekası teknikleri kullanılarak büyük ölçekli kablosuz algılayıcı ağlar için yeni bir yönlendirme algoritması geliştirmek ve uygulamaktır.

Yöntem olarak oęul zekası tekniklerinin kullanılmasının başlıca nedeni kablosuz algılayıcılar ile sosyal böcekler arasında olan benzerliktir. Sosyal böceklerden karıncalar, termitler, bal arıları vb. canlılar kendi kendilerine organize olabilmeleri (self-organization), iş bölümü yapabilmeleri, bölünerek büyük ölçeklere çıkabilmeleri ve sürü zekasını kullanarak problemlere çözüm getirmeleri gibi özellikleri kablosuz algılayıcılarla olan benzerliklerinden bazılarıdır. Bu nedenle tasarım aşamalarında oęul zekası teknikleri kullanılmıştır.

Geliştirilen yönlendirme algoritmasının test ortamı olan modelleme ve benzetim aracı olarak KAA topolojileri üretmeye uygun, algoritma geliştirilirken KAA'larda yapılacak deęişikliklere anında cevap verebilen, büyük ölçekleri destekleyen bir benzetim aracı geliştirilmiştir. Bu benzetim aracı DEVS yöntemini, DEVS-Suite benzetim aracının kolay kullanım özellikleri ve BRITE topoloji üreticinin esnek ortamı kullanılarak geliştirilmiştir.

#### **1.4. Tezin Kapsamı**

KAA'ların problemlerini karşılamak üzere geliştirilen algorithmada:

- Sosyal böceklerden ilham alınarak geliştirilen oęul zekası tabanlı bir yönlendirme algoritması geliştirilmiştir.
- Geliştirilen algoritmanın modelleme ve benzetimini gerçekleştirmek için DEVS modelleme ve benzetim yaklaşımı kullanılmıştır.

- Modellenen algoritmanın benzetimi için Java dilinin esnek yapısı kullanılarak DEVS-Sensor adını verdiğimiz bir benzetim aracı geliştirilmiştir.
- Geliştirilen algoritmayı test etmek amacıyla literatürde öne çıkan yönlendirme algoritmaları ile karşılatırmalar yapılmış ve elde edilen sonuçlar sunulmuştur.
- Geliştirilen yeni benzetim aracının onaylama ve geçерleme işlemleri yapılmış ve sonuçlar sunulmuştur.

### **1.5. Tez Çalışmasının Bilime Katkısı**

Tez sürecinde KAA'lar için geliştirilen DEVS-Sensor benzetim ortamının bilime olan katkıları aşağıdaki şekilde özetlenebilir:

- KAA için DEVS yöntemini kullanan DEVS-Sensor adında bir benzetim ortamı geliştirildi. Geliştirilen bu benzetim aracı KAA'ların modelleme ve benzetim ihtiyaçlarını karşılayacak şekilde tasarlandı.
- Geliştirilen DEVS-Sensor benzetim ortamı büyük ölçeklere çıkabilme kabiliyetine, esnek ve kolay kullanıma sahip olarak geliştirildi.
- Otomatik olarak topoloji üretmek amacıyla literatürde kabul görmüş bir topolji üretici olan BRITE topoloji üretici sisteme dahil edildi.
- Geliştirilen DEVS-Sensor benzetim aracı kümeleme yöntemlerinin uygulanabilmesine olanak sağlayacak şekilde tasarlandı.

Geliştirilen KAA'lar için oğul zekası tabanlı yönlendirme algoritmasının bilime olan katkıları aşağıdaki şekilde özetlenebilir;

- KAA'larda sistemin bekasını ilgilendiren ve başlıca problemlerden biri olan enerji problemine oğul zekası teknikleriyle çözüm getirildi.
- KAA'larda veri toplama olayını ilgilendiren paket teslim oranı problemine oğul zekası teknikleriyle yüksek paket teslim oranları elde edilerek çözüm getirildi.
- Geliştirilen KAA'lar için oğul zekası teknikleriyle ağ problemlerinin başında gelen ve KAA'lar da önemli yer tutan ortalama gecikme problemine çözüm getirildi.
- Geliştirilen kümeleme tekniği ile KAA'larda büyük ölçeklere çıkıldı.
- Kümeleme yöntemlerinde etkin yer tutan küme sayısı veya kümebaşı sayısında çoklu-atlamalı sistemler dikkate alınarak daha düşük küme sayısı elde edildi.

Özet olarak, gerçekleştirilen KAA'lar için oğul zekası tabanlı yönlendirme algoritması literatüre üstün özelliklere sahip, enerji yönünden verimli, veri toplama yönünden düşük kayıp oranıyla büyük ölçeklere çıkabilen yeni bir yönlendirme algoritması kazandırdı.

## 1.6. Tez Düzeni

Yapılan tez çalışması aşağıdaki gibi düzenlenmiştir:

Bölüm 1'de problemin tanımı, yapılan çalışmanın amacı, yapılan tez çalışmasını diğerlerinden farklı kılan yönler, bilime katkısı ve tez organizasyonu hakkında bilgi verilmektedir.

Bölüm 2'de kablosuz algılayıcılardan bahsedilmektedir. Literatürde yer alan KAA'lar için oğul zekası tabanlı yönlendirme algoritmaları ve KAA'lar için geliştirilen kümeleme algoritmaları incelenmektedir.

Bölüm 3'te sistemin tanımlanması, Ayrık Olaylı Sistem Tanımında (Discrete Event System Specification – DEVS) kullanılan temel kavramlar ele alınıp, ayrık olaylı benzetim stratejileri ve birbirleri ile olan ilişkileri anlatılmaktadır. DEVS yaklaşımı detaylandırılıp, atomik, birleşik DEVS kavramları tanımlanmakta ve hiyerarşik model tasarımından bahsedilmektedir. KAA'ların modellenmesinde ve benzetiminde kullanılan ortamlarından ve geliştirilen benzetim ortamının getirdiği çözümlerden bahsedilmektedir.

Bölüm 4'de, geliştirilen KAA'lar için oğul zekası tabanlı yönlendirme algoritmasının modeli, mimarisi ve paket yapılarından bahsedildi. Bunun yanında yönlendirme algoritması kullanılarak geliştirilen kümeleme yönteminden de bahsedilmektedir.

Bölüm 5'de geliştirilen algoritmanın benzetim ortamına uygulanması ve kullanılan DEVS-Sensor benzetim ortamından, ortamı tasarlarken geçilen aşamalar, DEVS ortamı, görselleştirme ortamı ve analiz ortamı anlatılmaktadır.

Bölüm 6'da geliştirilen KAA'lar için oğul zekası tabanlı yönlendirme algoritmasının DEVS-Sensor ortamına uygulamasından elde edilen sonuçlar grafik ortamına aktarıldı ve literatürdeki algoritmalar ile karşılaştırılmaktadır. Yapılan bu karşılaştırmalar yorumlanmaktadır.

Bölüm 7'de yapılan çalışmalardan elde edilen sonuçlar genel hatlarıyla özetlenmekte, tez çalışmasının bilime katkıları tartışılmakta ve ileride bu çalışmanın devamı olarak yapılabilecek çalışmalara ışık tutabilecek önerilerde bulunmaktadır.

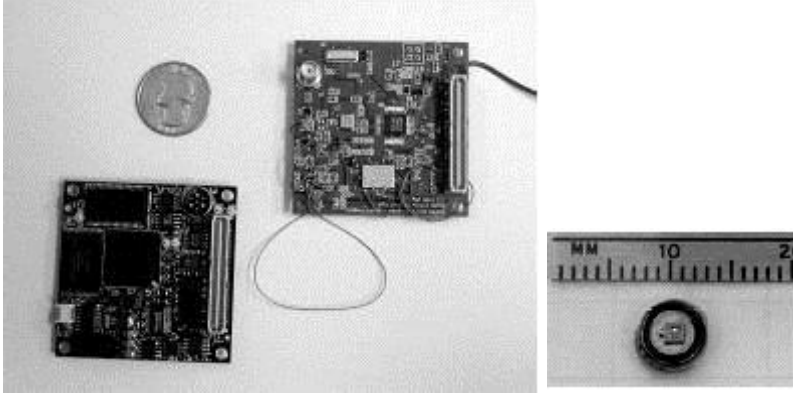
## **BÖLÜM 2. KABLOSUZ ALGILAYICI AĞLAR**

### **2.1. Giriş**

Kablosuz algılayıcı ağlar (KAA), çok fazla sayıda küçük boyutlu, düşük maliyetli ve kısa mesafede kablosuz ortam üzerinden haberleşebilen algılayıcı düğümlerinden meydana gelen bir ağdır. Bu ağda, düğümler rastgele olarak ortama bırakılabilmekte ve geliştirilen protokoller sayesinde kablosuz ortam üzerinden birbirleri ile haberleşerek kendi kendine organize olabilmektedir. Bu özellik, düğümlerin ortamdaki fiziksel büyüklük (ışık, sıcaklık, nem, basınç vb.) değişimlerini çok atlamalı (multihop) yollar üzerinden merkezi ağ birimine iletmesini mümkün kılmaktadır. Kablosuz algılayıcı düğümlerin düşük maliyetli olması, normal şartlarda erişimin imkânsız olduğu bölgelere kolaylıkla yerleştirilebilmesi ve uzun süre bakım istemeden çalışabilmesi gibi özellikler kablosuz algılayıcı ağlarının çok çeşitli alanlarda kullanılabilmesini sağlamaktadır. Genel bir algılayıcı ağ; gözlem alanı, algılayıcı düğümler, çıkış düğümü ve görev yönetim düğümünden meydana gelmektedir.

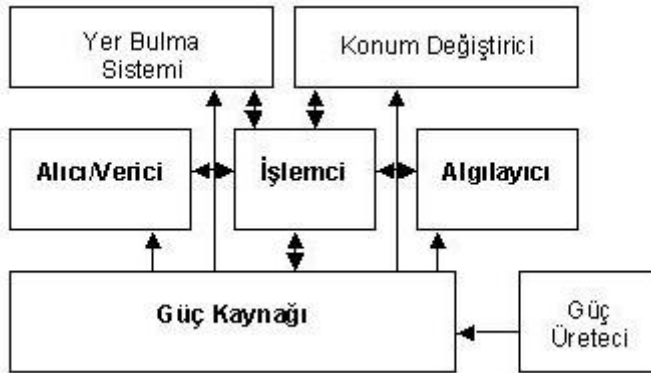
### **2.2. Kablosuz Algılayıcılar**

Mikroelektronik sistemlerde son yıllarda meydana gelen gelişmeler düşük maliyetli, düşük güç ihtiyacı olan, küçük boyutlu ve çok fonksiyonlu algılayıcıların üretilmesine olanak sağlamaktadır. Kablosuz algılayıcılar olarak adlandırılan araçlar kullanım alanlarına göre çevrelerindeki nem, sıcaklık vb. değerleri ölçüp, işleyerek hedefe gönderebilmektedirler. Çoğu zaman bir kibrit kutusu kadar olup, uygulamalara göre daha da küçük üretilmektedirler.



Şekil 2.1. Çeşitli boyuttaki kablosuz algılayıcılar

Kablosuz algılayıcı ağlarda görev yapan bir kablosuz algılayıcının Şekil 2.1’de fiziksel görüntüsü görülmektedir. Şekil 2.2’de ise kablosuz algılayıcının iç yapısı görülmektedir. Bu yapıda algılayıcı, işlemci, alıcı/verici ve güç birimleri olmak üzere dört ana elemanı vardır. Bunlara ilave olarak kullanım amacına göre bir kablosuz algılayıcı, yer bulma sistemi, güç üretim birimi ve konum değiştirici bulundurabilir [2].



Şekil 2.2. Kablosuz algılayıcı birimleri [2]

### 2.3. Kablosuz Algılayıcılar İçin Geliştirilmiş İşletim Sistemleri

Gittikçe küçülen kablosuz algılayıcıların enerji problemini çözüm getirebilecek ve bellekte az yer kaplayan yazılımlara ihtiyacı bulunmaktadır. Bunun için piyasada

geliştirilmiş olan ticari ve açık kaynak kodlu işletim sistemleri bulunmaktadır. Açık kaynak kodlu olarak;

- TinyOS
- MANTIS
- SOS

Ticari olarak ise;

- $\mu$ COS
- AVRX

İşletim sistemleri bunlardan bazıları olarak sayılabilir [34].

#### **2.4. IEEE 802.15.4/ZigBee Kablosuz Algılayıcı Ağ İletişim Protokolü**

Kişisel kablosuz ağlarda, düşük güç ile sınırlı kapasite veri iletimi sağlamak amacıyla ZigBee firması tarafından geliştirilmiş ve IEEE tarafından 802.15.4 adıyla standartlaştırılmış iletişim protokolüdür [35]. Tablo 2.1’de bu iletişim protokolünün radyo frekansları ve veri aktarım hızları gösterilmektedir.

ZigBee’nin diğer IEEE standartlarına göre ayırt edici özellikleri [36];

- 10 ile 115.2Kbps arasında düşük veri hızı,
- Standart bir batarya ile birkaç yıl süren düşük güç tüketimi,
- Çoklu izleme ve uygulama kontrolü sağlayan ağ topolojisi,
- Düşük maliyet, basit ve kolay kullanım,
- Yüksek güvenlik.

Tablo 2.1. IEEE 802.15.4 Radyo frekansları ve veri aktarım hızları

<b>Band</b>	<b>Etki Sahası</b>	<b>Kanal</b>	<b>Veri Hızı</b>
2.4GHz	Dünya geneli	16 kanal	250kbps
915MHz	Amerika	10 kanal	40kbps
868MHz	Avrupa	1 kanal	20kbps

## 2.5. Kablosuz Algılayıcı Ağların Kullanım Alanları

Kablosuz algılayıcı ağlar çok geniş kullanım alanlarına sahiptir. Bunlar aşağıdaki başlıklar altında toplanmıştır.

### 2.5.1 Çevresel uygulamalar

Orman yangını, sel, deprem, gibi doğal afetlerin ölçümlendirilmiş olarak hızlı bir şekilde ihbar edilmesinde, hava kirliliği tespiti ve ayrıntılı rapor alınmasında ve doğal yaşamın gözlenmesinde kullanılır [2].

### 2.5.2. Sağlık uygulamaları

İnsanların fizyolojik verilerinin uzaktan izlenmesi, hastanede bulunan doktorların yerinin ve hastaların durumunun (kalp atışı, kan basıncı vb.) izlenmesi ve hastanedeki ilaç dağıtımının yönetiminde kullanılır [2].

### 2.5.3. Ticari uygulamalar

Küçük çocukların konumlarının aileleri tarafından takip edilmesi, güvenlik ihtiyaçları, hırsızlarının tespiti, envanter yönetim yardımcı aracı, araçların izlenmesi ve tespit edilmesinde kullanılır [2].



#### 2.5.4. Askeri uygulamalar

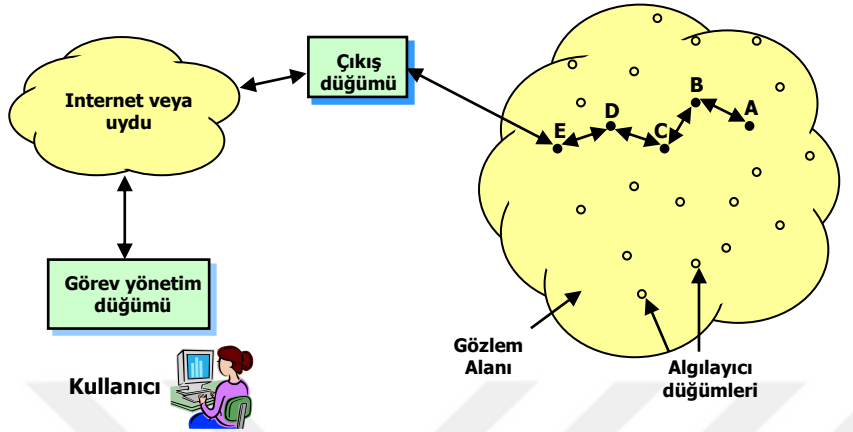
Dost kuvvetlerin teçhizat ve cephanesinin izlenmesi, savaş alanının gözlenmesi, arazi hakkında keşifte bulunma, hedefin konumu, sürati gibi hedef bilgilerinin tespiti, düşmana verdirilen hasar miktarının tespit edilmesi, nükleer, biyolojik ve kimyasal (NBC) saldırıları ihbarının alınması ya da keşfinde kullanılır [2].

#### 2.6. Kablosuz Algılayıcı Ağ Yönlendirme Protokolleri

Daha öncede ifade edildiği gibi, KAA'ların başlıca problemleri karmaşıklık, ölçeklenebilirlik, uyum yeteneği, sistemin bekası ve batarya ömrü olarak sıralanabilir. Yönlendirme protokolleri bu problemlere çözüm getirmeyi hedeflemektedir. KAA'larda yönlendirmenin temel amaçlarından biri ağ ömrünün artırılması ve yoğun enerji yönetimi tekniklerinin kullanımıyla ortaya çıkan bağlantı hatalarının önlenmesidir [2]. KAA'ların batarya ömrü sınırlıdır. Bununla birlikte, algılayıcıların genel olarak yerleşim şekli daha önceden belirlenmemekte, uygulama alanlarına (afet bölgeleri, ulaşılması zor alanlar) rastgele bırakılmaktadırlar. Bu nedenle algılayıcı düğümlerin herhangi bir bakım ve onarım gerektirmeden çalışmalarını sürdürmeleri gerekir. Batarya ömrünün artırılabilmesi için, algılayıcı düğümler arasında önemli ölçüde dayanışma ve işbirliğinin gerçekleştirilmesi gerekmektedir. Bu sebepten herhangi bir kablosuz algılayıcı ağa klasik bir ağ protokolü uygulanamamasından dolayı KAA'lara özgü gerekli kriterleri dikkate alan yeni ağ protokollerinin tasarımına ihtiyaç duyulmaktadır.

Karınca ve bal arısı gibi sosyal canlıların karmaşık kolektif davranışları ve çok merkezli yapıları kablosuz algılayıcı ağlar gibi dinamik sistemler ile benzerlikler taşımaktadır. Sosyal canlılar tek başlarına herhangi bir zeka ortaya koyamadıkları halde birlikte hareket ederek mükemmel bir sistem oluşturabilmektedirler. Bir lidere sahip olmamalarına rağmen sistemin devamlılığını sağlayabilmekte ve bu sayede sistem ölçeği ne kadar büyürse büyüsün problemsiz bir şekilde sistem işleyebilmektedir. Mesela arı kolonisinin birkaç binden kırk elli bine çıkması çalışmalarını etkilemez. Sistemde var olan canlılar enerji kaybını azaltmak amacıyla

buldukları yollara en kısa yolu tespit ederek gitmektedirler. Birçok özellik oğul zekasının kablosuz algılayıcı ağlar ile ne kadar çok örtüşüğünü göstermektedir. Şekil 2.3’de bir KAA modeli gösterilmektedir.



Şekil 2.3. Kablosuz algılayıcı ağ örneği [2]

## 2.7. Yönlendirme Protokolleri Tasarım Kriterleri

### 2.7.1. Enerji tüketimi

Algılayıcı düğümler, alınan verilerin gönderilmesinde ve işlenmesinde sınırlı ömre sahip olan bataryaları kullanırlar [37]. Algılayıcı düğümlerin ömrü büyük oranda batarya ömrüne bağlıdır [38]. Çok-atlamalı KAA'larda, her düğüm alıcı, işleyici ve göndericidir. Düğümler bataryanın bitmesinden dolayı devre dışı kaldıkça ağın topolojisi değişir. Bu değişime bağlı olarak yönlendirme işlemi yeniden yapılmalıdır.

### 2.7.2. Ölçeklenebilirlik

Tipik bir KAA uygulamasında algılayıcı düğümlerin yüzlercesi ya da binlercesi algılama yapılacak alana yerleştirilebilir. Yönlendirme protokollerinin çok büyük sayıda algılayıcı düğümlerle çalışabilmesi gerekir. Ayrıca algılayıcı düğüm sayısına ek olarak, bir yönlendirme protokolü çevredeki olayları da yeterince ölçekleyebilmelidir. Olay meydana geldiğinde veriye ulaşması gereken düğümlerin dışındakiler uyku durumunda kalabilmelidir [39].

### **2.7.3. Veri toplama**

Düğümler kendilerine birden fazla düğümden gelen benzer paketleri işleyip birleştirerek iletimi düşürmek amacıyla sayılarını azaltır [40]. Çünkü gelen tüm verilerin aynen gönderilmesi, işlenerek gönderilmesinden daha fazla enerji harcanmasına neden olur.

### **2.7.4. Ağ ömrü**

Bir algılayıcı ağın ömrü birkaç saat ile birkaç yıl arasında değişebilir. Ağ ömrünü enerji tüketimi ve düğümlerin sağlamlığı büyük oranda etkiler [41].

### **2.7.5. Hata toleransı**

Bazı algılayıcı düğümler güçlerinin azalması, fiziksel zararlar ve çevresel etkilerden dolayı kullanılamaz hale gelebilirler. Algılayıcı düğümlerin bu durumu algılayıcı ağdaki veri akışını etkilememelidir. Eğer çok fazla düğüm kullanılamaz hale gelirse, yönlendirme protokolleri yeni bağlantılar üzerinden yönlendirme yaparak bu durumu telafi edebilmelidir [42].

### **2.7.6. Paket teslim gecikmesi**

Paket teslim gecikmesi yönlendirme protokollerinin tasarım kriterlerinin en önemlilerinden biridir. Veri toplamadaki ve çoklu-atlamalı yönlendirmedeki gecikmeler paket teslim gecikmesine neden olur [43].

### **2.7.7. Yerleşim**

Algılayıcı düğümlerin yerleşimi fiziksel alana bağlı olarak değişir. Bu da yönlendirme protokolünün performansını etkiler. Düğüm yerleşimi daha önceden belirlenmiş ya da rastgele yapılabilir [44].

### **2.7.8. Servis kalitesi**

Servis kalitesi uygulamanın kalitesini ve ayakta kalma gücünü yansıtır [43]. Uygulamalar veri güvenilirliği, enerji verimi, yerel farkındalık ve ortaklaşa işlev görebilme gibi özelliklere gereksinim duyar. Bu faktörler özel bir uygulama için yönlendirme protokolü seçiminde etkilidir [43].

### **2.7.9. Veri teslim modelleri**

Her algılayıcı periyodik olarak veri gönderimi yapar. Olay ve sorgu temelli olmak üzere iki tür veri teslim yöntemi vardır. Olay temelli yaklaşımda, olay başladığında veri iletimi gerçekleştirilir. Sorgu temellide ise, hedef tarafından sorgulama başladığında veri iletimi başlar. Melez yaklaşım, bu iki modeli birleştirerek kullanır [45].

### **2.7.10. Operasyon çevresi**

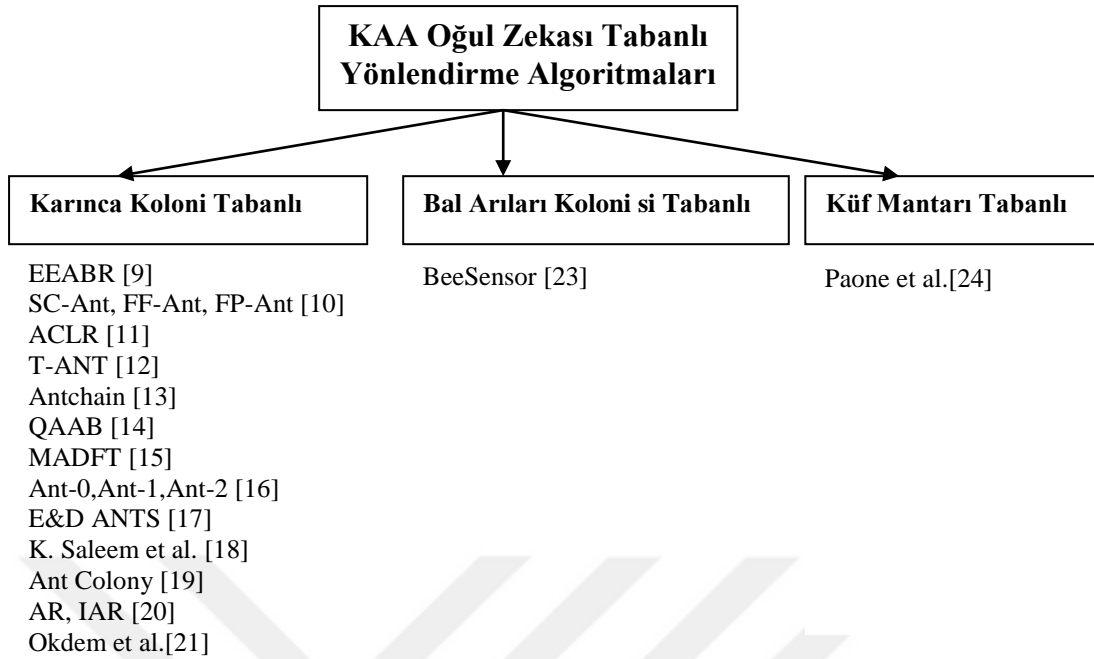
Algılayıcı düğümler uygulamalara bağlı olarak okyanuslara, ormanlara, bir evin içine veya daha farklı alanlara uygulanabilir. Bu şartlara bağlı olarak yönlendirme protokolleri değişim gösterir [46].

## **2.8. Oğul Zekası Tabanlı Yönlendirme Protokolleri**

Oğul zekası tabanlı yönlendirme algoritmaları karınca kolonisi tabanlı, bal arıları tabanlı ve küf mantarı tabanlı olmak üzere üç farklı sınıfa ayrılır (Şekil 2.4) [46].

### **2.8.1. Karınca kolonisi tabanlı (ACO) yönlendirme protokolleri**

Karınca kolonisi tabanlı (ACO) yönlendirme algoritmaları karıncaların yiyecek ararken ortaya koyduğu davranışları [47], bir karınca kolonisinin yuva inşası ve yiyecek hedefine ulaşma gibi durumlarda izlediği yolları dikkate alır [48].



Şekil 2.4. Oğul zekası tabanlı yönlendirme algoritmalarının sınıflandırılması [46].

### 2.8.1.1 Enerji etkili karınca temelli yönlendirme (EEABR)

Camilo tarafından 2006 yılında karınca kolonisi tabanlı optimizasyon olarak kablosuz algılayıcı ağlar için geliştirilmiştir [9].

Bu algortimada önceki düğüm, sonraki düğüm, karınca kimliği ve zaman aşımı değerini tutan kayıt alanları oluşturulur. İleri yönde hareket eden karınca alındığında, düğüm yönlendirme tablosunda o karıncaya ait bilgilerin olup olmadığına bakar. Eğer yoksa bilgiler tabloya kaydedilir. Zamanlayıcı yeniden başlatılır ve karınca bir sonraki düğüme gönderilir. Eğer bilgiler tabloda varsa karınca paketi düşürülür. Geri yönde hareket eden karınca alındığında, karıncayı hedefine göndermek için yönlendirme tablosunda bir sonraki düğüm araştırılır. Eğer hedefine ulaşamazsa zamanlayıcı yeniden kurulur ve karınca paketinin kayıtları silinir.

Protokolün karşılaştırma deneyleri Temel Karınca Kolonisi Yönlendirme (BABR) [20] ve Geliştirilmiş Karınca Kolonisi Yönlendirmesi (IABR) [20] algoritmaları ile Ns-2 [49] benzetim ortamında gerçekleştirilmiştir [9].

### **2.8.1.2. Algılayıcı-Güdümlü ve Maliyet-Farkındalıklı Yönlendirme (SC-ANT)**

Bu yaklaşım ileri yönde hareket eden karıncaların ilk gidecekleri en iyi yönü tahmin etmelerini sağlar. Ek olarak olasılık dağılımını güçlendirir [10]. Geliştirilen bu yaklaşımla birlikte FF ve FP olmak üzere iki farklı algoritma daha gerçekleştirilmiştir. SC yaklaşımını enerji tüketimini kriter olarak belirlemiştir.

### **2.8.1.3. Akıcı ileri karınca yönlendirme (FF-ANT)**

Bu yaklaşım yol bulma kontrol paketlerinin hedefi bulmak maksadıyla ağda dolaşmalarını ve böylece tüm düğümlere ulaşmalarını sağlar. Hedef bilinmediğinde veya maliyet tahmin edilemediğinde karıncalar tüm alanı dolaşırlar [10]. FF yaklaşımında sistemin bekası dikkate alınır. SC yaklaşımına ek yetenekler getirmiştir.

### **2.8.1.4. Akıcı taşıma karınca yönlendirme (FP-ANT)**

Kablosuz algılayıcı ağlardaki tüm düğümlere ulaşmak karmaşık, dinamik ve büyük oranda dağıtık bir yetenek gerektirir. Bu yaklaşım, ileri listelerini taşıyan veri karıncaları ismiyle yeni bir karınca türü ortaya koymuştur. Veri karıncalarının yönlendirme kontrolü FF'deki ileri karıncalara benzer şekilde halledilir [10].

### **2.8.1.5. Karınca Kolonisi Yer Farkındalıklı Yönlendirme (ACLR)**

Bu yaklaşım Wang tarafından 2008'de geliştirilmiştir [11]. Döngüden kaçınmak için hedef düğüme doğru veri paketlerinin teslimini garanti eden komşular yerine, karınca paketi düğümün komşularının bir altkümesini bir sonraki atlama düğümü olarak seçer. Algoritma karıncaların seçtiği bir sonraki atlama düğümlerinden formül yardımıyla tahminde bulunur. Bir karınca tarafından salgılanacak feromon miktarına karar vermek amacıyla, kalan enerjiye göre bazı yolların feromonlerini ve düğümlerin yerel bilgilerini ortadan kaldıracak bir algoritma kullanır.

Karşılaştırmalar BAR [50], FPAR [50], SCAR [51] ve IAR [20] ile OPNET [52] benztim ortamında yapılmıştır.

#### **2.8.1.6. T-ANT kablosuz algılayıcı ağ protokolü**

Bu yaklaşım Selvakennedy tarafından 2006 yılında geliştirilmiştir [12]. İki metod kullanır: değişim tahmini ve kümeleme metodu. Kümeleme metodu, kümebaşı seçilen karınca yerleştirilir. Düğüm kurulumu boyunca hedefe karınca (kontrol mesajları) gönderir. Karıncalardan elde edilen bilgiler ile komşu tabloları oluşturulur. Karınca gezintileri maksimum adım sayısı (time to live - TTL) ile sınırlandırılmıştır. Küme kurulumu CS zamanlayıcı ile kontrol edilir. Bu zamanlayıcı sonlandığında düğüm bir karıncaya sahip olup olmadığını kontrol eder. Eğer bir karıncaya sahipse kendisinin küme başı olduğunu bir reklam (advertisement – ADV) mesajı ile yayınlar.

Veri toplama işleminde bir düğümün katılımına karar vermek amacıyla verisinin verimli olup olmadığına karar verme sürecine değişim tahmini adı verilir.

Karşılaştırmalar TCCA [53] ve m-LEACH [54] ile kendine özel bir benzetici ortamında yapılmıştır.

#### **2.8.1.7. Ant-chain kablosuz algılayıcı ağ protokolü**

Ding ve Liu tarafından 2005'te geliştirilmiştir [13]. Enerji tüketimi, veri bütünlüğü ve düğüm ömrü konularına odaklanmıştır.

Bu yöntemde karınca kolonisi bir zincir formundadır. Zincir bilgisi algılayıcı düğümler için yönlendirme bilgisi olarak yayınlanır. Üç farklı zincir şeması ile KAA'ların farklı durumlarında veri toplama işlemi gerçekleştirirler. İki yönlü Ant-chain küçük topoloji değişimlerine karşı kendi kendine uyum sağlayabilir. Tek yönlü Ant-chain veri toplama dolaşımını sınırlandırmak için tasarlanmıştır. Sorgu

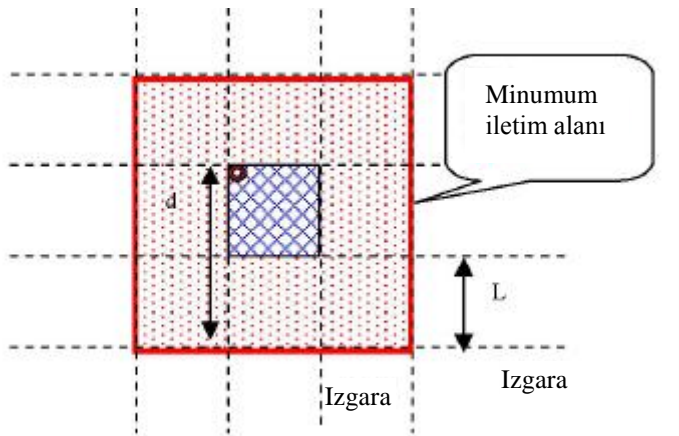
zincir ilgilenilen algılayıcı düğümlerinden veri toplamak için kullanılır. Zincir tipi ve bilgisi alındıktan sonra düğüm veri toplama işlemi bağımsız olarak çalışır.

Karşılaştırmalar PEGASIS [55] ve LEACH [54] protokolleri ile, deneyler ise Ns-2 benzetim ortamında gerçekleştirilmiştir.

### 2.8.1.8. Kraliçe karınca tabanlı yönlendirme (QAAB)

Sun tarafından 2006 yılında geliştirilmiştir [14]. Coğrafi yönlendirme protokolü olan GAF protokolünü temel alarak İnternet üzerinden düğümlere erişim sağlamayı hedeflemiştir.

Bu modelde GPS tarafından sıklıkla düğümlerin pozisyonları güncellenerek sanal ızgaralar çizilir. Ağın manevra kabiliyetini ve İnternet arayüzünü sağlamak amacıyla protokol düğümleri beş türde sınıflandırılır; Kaynak düğüm (olay başlangıcı), hedef düğüm (alıcı), Kraliçe karınca düğümü (internet arayüzü), ana düğüm (izleyiciler) ve normal düğümler. Şekil 2.5’de protokolün ızgara yapısı verilmiştir.



Şekil 2.5. QAAB protokolü ızgara yapısı

Karşılaştırmalar SPIN [56] protokolü ile GloMoSim [57] ağ benzetim ortamında gerçekleştirilmiştir.



### **2.8.1.9. Karınca tabanlı minimum veri birleştirme ağacı (MADFT)**

Juan tarafından 2007 yılında geliştirilmiştir [15]. Karınca kolonisi yaklaşımı ile yönlendirme problemine çözüm getirecek sezgisel bir hedef seçim algoritması geliştirilmiştir.

Öncelikle kaynak düğüm için karıncalar belirlenir. Karıncalar yolları belirler. Dönüş yolunda keşfedilen yollara en yakın noktadan karıncalar tarafından yeni yollar belirlenir. Bu yollar arasında geliştirilen olasılık hesabı formülü ile en kaliteli yol seçilir. Olasılık fonksiyonu en az maliyetli yolu bulabilmek amacıyla maliyet ve feromon değerlerini kullanır.

Testler C++ da hazırlanmış bir benzetim ortamında gerçekleştirilmiştir.

### **2.8.1.10. Ant-0, Ant-1 ve Ant-2 kablosuz algılayıcı ağ protokolleri**

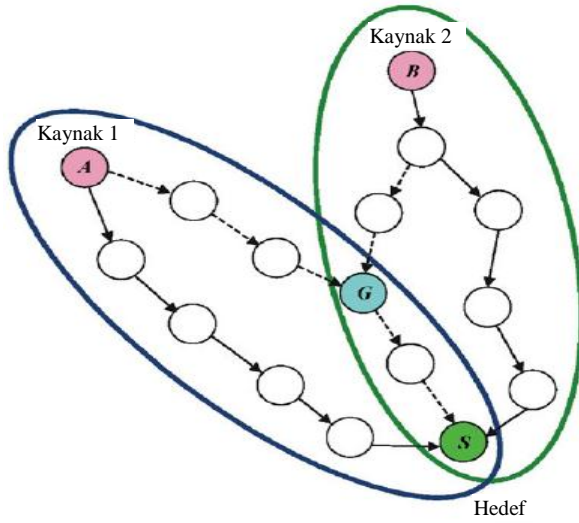
Liao tarafından 2007 yılında geliştirilmiştir [16]. Veri toplama problemine çözüm getirmek amacıyla karınca kolonisi yaklaşımı kullanılmıştır.

Karıncalar aracılığıyla kaynak ile hedef düğümler arasında en kısa yollar kurulur. Oluşturulmuş olan ağaç yolun alt dallarından benzer veriler geliyorsa birleştirilip tek bir mesajla gönderilir. Daha sonra birikmiş feromonlar tarafından veri yolları ağacı kurulur. Şekil 2.6'da iki ayrı kaynaktan aynı hedefe doğru oluşturulan yollar gösterilmektedir.

DD [58] yönlendirme protokolü ile karşılaştırılmıştır.

### **2.8.1.11. E&D ANTS kablosuz algılayıcı ağ protokolü**

Wen tarafından 2008 yılında geliştirilmiştir [17]. Karınca kolonisi yaklaşımı kullanılarak enerji tüketimi ve gecikmeyi minimum seviyeye indirmektedir.



Şekil 2.6. G düğümünün araştırma bölgesinin genişletilmesi

Anahtar fikir olarak düğümlerin feromonlarını güncellemek amacıyla kablosuz iletişimdeki enerji ve gecikmenin değerlerini kullanır. Güçlendirilmiş Öğrenme (Reinforce Learning-RL) algoritmasının yeni bir çeşidini kullanarak maksimum seviyede ağ ömrü ve minimum yayılma gecikmesi sağlamayı hedeflemiştir.

Karşılaştırmalar AntNet [13] ve Antchain [13] yönlendirme protokolleri ile OPNET benzetim ortamında gerçekleştirilmiştir.

### 2.8.1.12. Kendi kendine uyum sağlayan çoklu-yol yönlendirme

Saleem tarafından 2009 yılında geliştirilmiştir [18]. Enerji, gecikme ve hız konularına çözüm getirmeyi hedeflemiştir. KAA'larda tıkanma problemini giderebilmek için çoklu yol (multi-hop) yaklaşımını kullanmıştır.

Yönlendirmeye karar vermek için paket alma oranı, enerji ve gecikme olmak üzere üç parametre kullanılır. Enerji tüketimini azaltmak için algoritmada çapraz-katman yöntemi kullanılır. Çoklu-yol (multi-hop) özelliği ise iki veya daha fazla yol üzerinde yönlendirmeyi dağıtıp trafik yükünü azaltmaktır. Böylece veri çıkış oranını yükseltip veri tıkanıklığını önlemektedir.

Testler Ns-2 [49] benzetim ortamında gerçekleştirilmiştir.

#### **2.8.1.13. Ant Colony kablosuz algılayıcı ağ protokolü**

Chao tarafından 2008 yılında geliştirilmiştir [19]. Enerji tüketimi, uçdan uca gecikme ve ağ ömrü ile ilgili çözümler getirmektedir. Bu yaklaşım ağ ömrü, enerji tüketimi ve uçdan uca gecikme problemlerine DD yönlendirme protokolünün getirdiği iyileştirmeleri iletirmektedir.

Karşılaştırmalar DD [58] yönlendirme protokolü ile JiST/SWANS [59] benzetim ortamında gerçekleştirilmiştir.

#### **2.8.1.14. AR, IAR kablosuz algılayıcı ağ protokolleri**

Aghaeil tarafından 2007 yılında geliştirilmiştir [20]. Enerji tüketimi, başarımları oranı ve gecikme problemlerine çözüm getirmeyi hedeflemiştir.

ADR algoritmasına Güçlendirilmiş Öğrenme (RL) algoritması ekleyerek geliştirilmiştir. AR algoritmasında istenen sonuçlar elde edilemediğinden IAR algoritmasında önerilmiştir. Her düğüm için olasılık dağılımını kullanır.

Karşılaştırmalar SC-Ant [10], FF-Ant [10], FP-Ant [10], AntNet [13] yönlendirme protokolleri ile Java tabanlı geliştirilen bir benzetim ortamında gerçekleştirilmiştir.

#### **2.8.1.15. Kablosuz algılayıcı ağlar için karınca kolonisi temelli bir yönlendirme**

Ökdem ve Karaboğa tarafından 2009 yılında geliştirilmiştir [21]. Çözüm önerilerini dört aşamada ele almıştır. İlk olarak, bir düğüm farklı yollar kullanabilmek için göndereceği veriyi parçalara ayırır. Gönderdiği verinin ulaştığını garanti edebilmek için de onay mesajı kullanır. Eğer onay mesajı gelmezse farklı yol kullanarak paket yeniden gönderilir. İkinci olarak, yolun uzunluğu kadar düğümlerin enerji seviyelerinde gözönüne alınır. Bu işlem fazla enerji seviyesine sahip düğümler

tarafından gerçekleştirilir. Böylece ortalama ağ ömrü artar. Üçüncü olarak, karınca ajanları tekniği kullanılarak yeni bir iletişim tekniği geliştirilmiştir. Dördüncü olarak, düğümlerin yer değişimleri göz önünde bulundurulmamasına rağmen değişimler ağ güvenliğini etkilemez. Yolları organize etmek için bölüm kurulur ve veri iletimi bu bölümlerden gerçekleştirilir.

Karşılaştırmalar EEABR [9] yönlendirme protokolü ile MATLAB’da paralel ayrık olaylı benzetim aracında gerçekleştirilmiştir.

### **2.9.2. Bal Arısı kolonisi tabanlı yönlendirme protokolleri**

Bal arılarının yiyecek bulma davranışlarını temel alarak geliştirilmiş protokollerdir. Bal arıları bir çok alanda ilham kaynağı olduğu gibi yönlendirme protokollerine de ilham kaynağı olmuştur. Arıların iş bölümü yapmaları ve kendi kendilerine organize olabilmeleri nedeniyle KAA'lara benzemektedirler. Bu benzerlik KAA alanına da uygulanmalarına neden olmuştur.

#### **2.9.2.1. Beesensor kablosuz algılayıcı ağ protokolü**

Saleem ve Farooq 2007 yılında daha önce kablolu ağlar için geliştirilmiş olan BeeHive [22] yönlendirme protokolünü KAA'lara uyarlamışlardır [23].

BeeSensor dört farklı ajan kullanır: paketleyici, izci, toplayıcı ve oğul [85]. Paketleyici durağan ajanlar olarak çalışır. Çünkü algılayıcı düğüm içinde çalışırlar. İziciler ağda yol bulmak için dolaşırlar. İziciler ikiye ayrılır. İleri izciler ve geri izciler. İzci kaynak düğüm tarafından etiketlenir. Önce ileri izciler yayınlanır ve bunlar hedefe giden yolları araştırır. Hedefe ulaşan izci yol bilgilerini geri izciler teslim eder ve kaynağa geri döner. Toplayıcılar alınan bu yol bilgisini kullanarak paketleyiciler tarafından hazırlanan paketleri taşırlar. Oğullar, toplayıcılara benzerler. Toplayıcılar yükleri fazla olduğunda oğullar birden fazla toplayıcı taşıyabilir.

Karşılaştırmalar FP-Ant [10], EEABR [9] ve AODV [60] yönlendirme protokolleri ile Ns-2 benzetim ortamında gerçekleştirilmiştir.

### **2.9.3. Küf mantarı tabanlı yönlendirme protokolleri**

Küf mantarı tek-hücreli bir türdür. Tek hücreliler ile kablosuz algılayıcı ağlar arasında kıyas olarak güçlü bir bağlantı vardır. Bir algılayıcı ağ basit bir koloni gibi ele alınır. Sınırlı kaynaklı düğümler, özerklik, sadece basit işlemleri yapabilme gibi özellikleri KAA ile benzerlik taşır. Fakat koloni çok karmaşık problemlere çözüm getirmek zorundadır [24]. Bu anlamda benzerlikler ele alınarak küf mantarı davranışları KAA'lara uygulanmıştır.

#### **2.9.3.1. Kablosuz algılayıcı ağlarda çoklu-hedef için oğul zekası tabanlı yönlendirme protokolü**

Paone tarafından 2009 yılında gerçekleştirilmiştir [24]. Küf mantarı davranışlarından ilham alınmıştır. Çoklu-hedefe ulaşırken hata toleransını azaltmayı hedefler.

Yönlendirme işlemini iki farklı süreç olarak ele almıştır. İlk olarak sinyal işleme, veri paketlerinin hedefe doğru yönlendirme sürecinde yönlendirme tabloları oluşturma ve güncelleme sürecidir. Sinyal işleme dört işlemde oluşur.

Birinci süreç ileri davranış tahmini; düğüm iç durumunu kontrol edip çevreyi algılayarak ileri yönde davranışını tahmin eder. Feromen çıkarma, düğüm sinyal işleme paketlerini kullanarak ileri davranışını yayınlar. Feromen algılama, düğüm komşusuyla ilgili olan ileri davranış yönlendirme tablosuna kaydeder. Feromen kaldırma, düğüm komşusunun ileri davranışını azaltır.

İkinci süreç yönlendirme işlemi; her düğüm yönlendirme tablosundaki feromen seviyelerine bakarak komşuları arasından bir sonraki atlamayı seçer. Her düğüm ileri veri paketlerini en yüksek feromen atlamasına göre gönderir. Veri akışı feromenini izleyerek hedefe ulaşır ve parametreleri ayarlar.

Testler OMNET++ [61] benzetim ortamında gerçekleştirilmiştir.

İncelenen KAA'lar için oğul zekası kullanan yönlendirme algoritmalarının yönlendirme kriterleri açısından karşılaştırmaları, temel aldıkları oğul zekası teknikleri ve algoritmaların test edildiği benzetim ortamları Tablo 2.2'de verilmiştir. Tabloya bakıldığında oğul zekası tekniklerinin karınca kolonisi üzerine yoğunlaştığı görülmektedir. Enerji tüketimi tüm algoritmalarda başlıca problem olarak ele alınmıştır. Algoritmaların bir kısmı birkaç kriter üzerinde dururken, bazı algoritmalar tek bir kriter üzerinde yoğunlaşmıştır. Tabloda kriterler zayıf, güçlü ve çok güçlü olmak üzere üç aşamada seviyelendirilmiştir.

## 2.10. Kümeleme Tabanlı Yönlendirme

Ağ topolojisi oluşturmada en önemli yöntemlerden biri kümelemedir. KAA'larda kümeleme işleminde temel anlamda kümeler kümebaşı ve ona bağlı düğümlerden oluşur. Küme üyesi olan düğümler çevrelerinden aldıkları verileri kümebaşına iletirler. Kümebaşlarında bu verileri oluşturulmuş olan bir omurga ağ vasıtasıyla hedefe ulaştırırlar.

Kümeleme teknikleri düğüm yerleşimine ve önceden belirlenmiş düğüm şemalarına, kümebaşı düğümlerinin karakteristiğine ve ağ işlem modellerine göre değişkenlik gösterebilir. Kümebaşı kaynakları açısından daha zengin bir düğüm veya düğümlerin içinden seçilen bir düğüm olabilir. Küme üyeleri sabit veya değişken olabilir. Kümeleme ölçeklenebilirlik açısından diğer tekniklere göre daha fazla avantaja sahiptir.

### 2.10.1. Kümeleme tabanlı yönlendirme parametreleri

Kümeleme tabanlı yönlendirmeyi etkileyen bazı önemli parametreler vardır. Bunları aşağıdaki gibi sıralayabiliriz.

Küme sayısı: Olasılıklı yaklaşımlar ve rasgele yerleşimde küme sayılarının ne kadar olduğu bilinmemektedir. Ortaya konan algoritmalarla küme sayısı ve kümebaşı seçimi değişiklik göstermektedir. Bu nedenle küme sayısı kümelemede önemli bir parametre olarak ortaya çıkmaktadır.

Tablo 2.2. Yönlendirme protokollerinin karşılaştırılması [46].

Yönlendirme Protokolleri	Enerji tüketimi	Ölçeklenebilirlik	Veri Toplama	Ağ Ömrü	Hata Toleransı	Paket Teslim Gecikmesi	Veri Teslim Başarım Oranı	Benzetim Ortamı	Oğul Zeka Tabanı
<b>T-ANT</b>	Güçlü	Güçlü	Çok Güçlü	Güçlü	Zayıf	Zayıf	Zayıf	Ayrık olaylı benzetici	ACO
<b>AntChain</b>	Güçlü	Zayıf	Güçlü	Çok Güçlü	Zayıf	Zayıf	Zayıf	NS2	ACO
<b>QAAB</b>	Çok Güçlü	Zayıf	Zayıf	Güçlü	Zayıf	Zayıf	Zayıf	Glomosim 2.0	ACO
<b>Paone et al.</b>	Güçlü	Zayıf	Zayıf	Zayıf	Çok Güçlü	Zayıf	Zayıf	Omnnet++	Küf mantarı
<b>ACLR</b>	Çok Güçlü	Zayıf	Zayıf	Zayıf	Zayıf	Güçlü	Zayıf	Opnet	ACO
<b>MADFT</b>	Güçlü	Zayıf	Zayıf	Zayıf	Zayıf	Zayıf	Zayıf	C++	ACO
<b>Ant-0, Ant-1, Ant-2</b>	Çok Güçlü	Zayıf	Çok Güçlü	Zayıf	Zayıf	Zayıf	Zayıf		ACO
<b>E&amp;D ANTS</b>	Güçlü	Zayıf	Zayıf	Zayıf	Zayıf	Güçlü	Zayıf	Opnet	ACO
<b>K. Saleem et al.</b>	Güçlü	Zayıf	Zayıf	Zayıf	Zayıf	Güçlü	Zayıf	NS2	ACO
<b>SC-Ant</b>	Çok Güçlü	Zayıf	Zayıf	Zayıf	Zayıf	Zayıf	Zayıf	Farklı yol Takibi oyunu (Peg)	ACO
<b>FF-Ant</b>	Zayıf	Zayıf	Zayıf	Zayıf	Zayıf	Çok Güçlü	Zayıf	Farklı yol Takibi oyunu (Peg)	ACO
<b>FP-Ant</b>	Zayıf	Zayıf	Zayıf	Zayıf	Güçlü	Zayıf	Çok Güçlü	Farklı yol Takibi oyunu (Peg)	ACO
<b>Ant Colony</b>	Güçlü	Zayıf	Zayıf	Güçlü	Zayıf	Güçlü	Zayıf	Jist/SWANS	ACO
<b>Okdem et al.</b>	Güçlü	Zayıf	Zayıf	Zayıf	Zayıf	Zayıf	Zayıf	Matlab	ACO
<b>AR, IAR</b>	Güçlü	Zayıf	Zayıf	Zayıf	Zayıf	Çok Güçlü	Çok Güçlü	Java-Based Simulation	ACO
<b>EEABR</b>	Çok Güçlü	Zayıf	Zayıf	Güçlü	Güçlü	Zayıf	Zayıf	NS2	ACO
<b>Beesensor</b>	Çok Güçlü	Zayıf	Zayıf	Güçlü	Güçlü	Çok Güçlü	Zayıf	NS2	Arı kolonisi

Kümelerarası iletişim: Kümeleme yaklaşımları ortaya konulurken kümeler arası iletişimin veya kümebaşlarının hedef ile olan iletişimlerinin gerçekleşme şekli önemlidir [32]. Kümebaşlarının hedefe ulaşmaları tek atlamalı ve çoklu atlamalı

olmak üzere iki türdür. Tek atlamalı düğümlerde kümebaşlarının kaynakları daha geniştir. Çok atlamalılarda ise kümebaşları düğümler arasından seçilir.

**Düğümler ve kümebaşı hareketliliği:** Düğümlerin ve kümebaşlarının durumu yapılacak çalışmaya göre durağan veya hareketli olarak belirlenir. Hareketli düğümlerin özellikleri durağan düğümlere göre farklılık arzeder [62].

**Düğümlerin tipleri ve rolleri:** Bazı KAA'larda kümebaşları üye düğümlerden veri işleme ve iletişim açısından daha fazla özelliğe sahiptir. Çoğu ağ modellerinde tüm düğümler benzer özelliklere sahiptir [63].

**Kümeleme metodları:** Kümebaşı normal bir düğüm iken ve zaman verimliliği birinci kriter iken, kümeleme koordinasyonsuz bir şekilde dağıtık olarak yapılır. Merkezi yaklaşımda, bir veya daha fazla koordinatör düğüm birbirine bağlı olmayan düğümleri ve küme bağlantılarını organize eder [62].

**Kümebaşı seçimi:** Bazı algoritmalarda kümebaşı önceden belirlenir. Çoğu algortmada ise kümebaşı tamamen rasgele, olasılıklara bağlı olarak ya da belirli ölçütlere (kalan enerji, bağlanabilirlik vb. ) bağlı olarak belirlenir [63].

**Kümeleme hiyerarşisi:** Çoklu seviyedeki hiyerarşik yaklaşım toplam enerji tüketimi (bir seviyede enerji tüketmek yerine) ve daha iyi bir enerji dağılımı sağlamaya çalışır. Büyük ölçeklere çıktığında hiyerarşik yaklaşım önerilmektedir [63].

## **2.10.2. Kümeleme tabanlı yönlendirme protokolleri**

### **2.10.2.1. Bağlantılı kümeleme algoritması (LCA)**

Baker ve Ephremides tarafından geliştirilmiştir [25,26]. Kablosuz ağlar için geliştirilen ilk kümeleme algoritmalarındandır. Ağ topolojisi düğümlerin hareketlerine göre düzenlenebilir. Kümebaşları düğümler hareket ederken bağlanabilecekleri bir omurga ağ oluşturur. Kümedeki tüm düğümler kümebaşlarına



doğrudan bağlanır. LCA bu yüzden maksimum bağlanabilirliğe sahiptir. İlk olarak her düğüm kendi kimliğini yayınlar ve ortamdaki iletimi dinler. Her düğüm 1 atlama ve 2 atlama ötedeki komşularını kaydeder. Bir düğüm komşuları arasındaki en yüksek kimliğe sahipse o kümebaşı olur.

### **2.10.2.2. Rasgele çekişme tabanlı kümeleme (RCC)**

RCC [27] adhoc ağlar için geliştirilmesine rağmen, kablosuz algılayıcı ağlara da uygulanmıştır. Sabit kümelemeye odaklanır. Eğer herhangi bir düğüm kümebaşı olmak isterse “ilk deklarasyon kazanma kuralları” algoritmasını uygular. Kümebaşı olma talebi diğer düğümler tarafından duyulduktan sonra komşu düğümler kümesine dahil olurlar. Kümenin sürmesi için her kümebaşı periyodik olarak kümebaşı talep paketi yayınlar. Paketlerin yayınlanması ve alınması arasında bir gecikme varsa, eşzamanlı yayında bir çekişme olur. Bunun sonucunda komşu paketler kümebaşı talep paketi yayımlayabilir. RCC bu problemi çözmek için rasgele zamanlayıcı ve düğüm kimliği kullanır. Her düğüm kümebaşı talep paketinden önce zamanlayıcıyı sıfırlar. Zamanlayıcı zamanı boyunca başka bir düğümden böyle bir paket alırsa, bu talebini durdurur.

### **2.10.2.3. CLUBS kümeleme protokolü**

Nagpal and Coore [28] tarafından geliştirilmiştir. Üç farklı karakteristik özelliğin üzerine inşa edilmiştir.

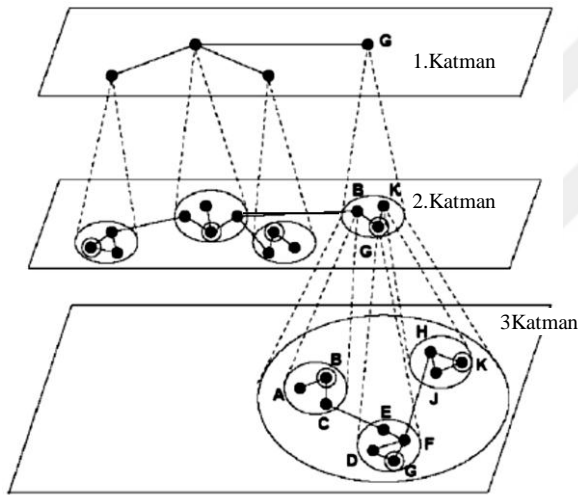
1. Ağdaki her düğüm bir kümeye bağlanmalı,
2. Ağdaki tüm kümelerin maksimum çapı aynı olmalı ve
3. Kümeler küme içinde ve diğer kümeler ile iletişim kurmalıdır.

Kümeler maksimum iki atlamadan oluşur. Ağdaki her düğüm sabit bir tamsayı dizisinden seçilen rasgele bir sayı tarafından bir küme içine yerleştirilir. Sonra bu

sayıcı geri doğru sayar. Eğer bu sayıcı komşuluk bittiği için kesilirse veya sıfırlanır, kendini kümebaşı ilan eder ve bir mesaj yayınlar. İki atlama uzağındakiler bu mesajı aldığı anda, geri sayıcılarını durdurur ve bu kümeye dahil olurlar.

#### 2.10.2.4. Hiyerarşik kontrollü kümeleme

Banerjee and Khuller [29] tarafından geliştirilmiştir. Kümelemenin katmanlar şeklinde gerçekleştirildiği bir yöntemdir (Şekil 2.7). Kümelemeyi başlatan düğüm ilk düğüm kimliğini alır. İki aşamadan oluşur: ağaç keşfi ve kümeleme işlemi. Ağaç keşfi, kümelemeyi başlatan düğümden başlayarak dağılır.

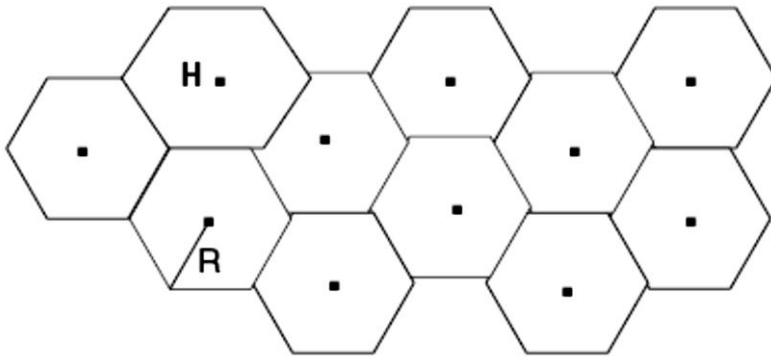


Şekil 2.7. Hiyerarşik kümeleme

#### 2.10.2.5. GS<sup>3</sup> kümeleme protokolü

Zhang and Arora [30] tarafından geliştirildi. Altıgen yapılı hücrelerden oluşur. Coğrafi sınırların görmezden gelinmesini savunur. Geometrik büyüklüğü bir ölçü olarak kümedeki tüm düğümleri içeren bir dairenin yarıçapını tanımlar. Büyük küme yarıçapı enerji tüketimini, hücre için iletim güvenilirliğini ve ağdaki radyo frekansların yeniden kullanımını artırır. Düğümler büyük ve küçük olmak üzere iki farklı türde varsayılır. Büyük düğümler kümeleme işlemi başlatmaktan sorumludur

ve diğ er ağ ve hücrelerle küçük düğ ümler arasında iletişim kurar. Hüresel altıgen yapı sanaldır ve gruplamaya ve düğ üm dağılımına yardım etmek için kullanılır. Şekil 2.8’de büyük düğ ümlerden biri komş u hücreler arasında komş u baş larını seçerek kümeleme iş lemini baş latır. Seçilmeyen hücreler hücre üyesi olur. Hücre merkezleri yeniden belirlenir. Hücre baş ları tarafından komş u hücre kurulumları baş lar. Tüm hücreler eklenene kadar bu iş lem devam eder.



Şekil 2.8. Bir hücrenin merkezine büyük düğ üm yerleştirilerek oluşturulan sanal altıgen hücreler

#### 2.10.2.6. Enerji etkili hiyerarş ik kümeleme (EEHC)

Bandyopadhyay and Coyle [31] tarafından gerçekleştirilmiştir. Ağ ömrünü maksimum yapmaya çalış an dağı tık, rasgele kümeleme algoritmasıdır. Kümebaş ları kendi kümelerindeki algılayıcıların okuduğ u bilgileri bir rapor halinde baz istasyonuna gönderir. Giriş ve geniş letme olmak üzere iki teknik kullanır. Giriş bölümü, her algılayıcı düğ ümün komş ularıyla kurduğ u iletişim oranı ihtimaline dayanarak kendini kümebaş u duyurduğ u bölümdür. Bu kümebaş ları gönüllü kümebaş ları olarak anılır. Bir kümebaş ının önceden belirlenen bir atlama sayısının içinde bulunan tüm düğ ümler, doğ rudan veya dolaylı olarak bu duyuruyu alır. Duyuruyu alan her düğ üm en yakın kümenin üyesi olur. Bir kümeye ait olup olamaması önemli olmayan kümebaş ları güçlü kümebaş larıdır. İkinci durum geniş letme ise kümelemeyi hiyerarş ik hale getirmektedir.

### 2.10.2.7. Düşük enerji uyumlu kümeleme hiyerarşisi (LEACH)

KAA'lar için en bilinen kümeleme algoritmasıdır [32]. Kümebaşı düğümleri baz istasyonuna yönlendirir ve kümeler sinyal gücü seviyesine göre oluşturulur. LEACH merkezi olmaksızın düğümlerin otonom kararlar vererek kümeler oluşturduğu dağıtık bir algoritmadır. İlk önce herhangi bir düğüm kümebaşı olmak için karar verir ve bu kararını yayımlar. Kümebaşı olmayan düğümler en az enerji harcanarak ulaşılabilecek kümeyi ve kümebaşını seçmek için karar verirler. Kümebaşı olma yük dengelemek için düğümler arasında periyodik olarak döndürülür. Bu dönüşüm "T" rasgele sayısının 0 ve 1 arasından seçilerek elde edilir. Eğer bu sayı eşik değerinden küçükse o düğüm kümebaşı olabilir.

Kümebaşı değişimi yapılırken düşük enerjili bir düğümünde kümebaşı olma şansı vardır. Bu düğüm öldüğünde tüm hücre bozulur. Küme başlarının doğrudan baz istasyonuna verileri ulaştırdığı varsayılır. Bu yüzden kümebaşlarının iletim menzili geniştir. Bu yaklaşım sıradan bir düğüm kullanıldığında doğru bir yaklaşım değildir.

### 2.10.2.8. Hızlı yerel kümeleme (FLOC)

FLOC [33] yaklaşık olarak eşit büyüklükte küme oluşturan dağıtık bir algoritmadır. Kabul edilen radyo modeli düğümlerin kümebaşına yakınlığını i-band ve o-band olarak sınıflandırmıştır. O-band düğümlerinde mesajlar kaybolurken, i-band düğümleri kümebaşı ile iletişimde çok az müdahale yaşayacaktır. Bir düğüm boştayken potansiyel kümebaşından davet alır. Eğer hiçbir davet almamışsa, kümebaşı aday olur ve adaylık mesajını yayımlar.

Tablo 2.3'de kümeleme algoritmalarının yönlendirme kriterlerine göre sınıflandırılması gösterilmektedir. Tabloda kümeleme algoritmalarının problemlere çözüm getirme şekillerinde verilmiştir.

Tablo 2.3. Kümeleme yönlendirme algoritmaları sınıflandırması [33]

Kümeleme yaklaşımları	Küme Özellikleri					Kümebaşı Yetenekleri				Kümeleme İşlemi			
	Küme sayısı	Küme topolojisi	Küme bağlantıları	İstikrar	Hareket	Düğümler tipi	Rol	Metod	Düğümler gruplama amacı	KB seçimi	Algoritma karmaşıklığı		
<b>LCA</b>	Değişken	Sabit(1- atlama)	Direk bağlantı/ çoklu atlama	Sağlar	Hareketli	Algılayıcı	Toplama	Dağıtık	Bağlanabilme	Rasgele	Değişken		
<b>RCC</b>	Değişken	Uyumlu	Direk bağlantı	Sağlar	Hareketli	Algılayıcı	Yayınlama	Melez	İstikrar ve basitlik	Rasgele	Değişken		
<b>CLUBS</b>	Değişken	Sabit(2- atlama)	Çoklu atlama	Değişken	Yeniden yerleşebilir	Algılayıcı	Toplama ve yayınlama	Dağıtık	Yönetim ve ölçekleme	Rasgele	Değişken		
<b>Hiyerarşik kontrollü kümeleme</b>	Değişken	Uyumlu	Çoklu atlama(Hiyer arşik)	Sağlar	Yeniden yerleşebilir	Algılayıcı	Yayınlama	Dağıtık	Yönetim ve ölçekleme	Rasgele	Değişken		
<b>GS<sup>3</sup></b>	Ön tanımlı	Uyumlu	Direk bağlantı	Sağlar	Yeniden yerleşebilir	Zengin kaynaklı	Yayınlama	Dağıtık	Ölçekleme ve hata toleransı	Ön tanımlı	Değişken		
<b>EEHC</b>	Değişken	Uyumlu	Direk bağlantı/ çoklu atlama(hiyer arşik)	Değişken	Sabit	Algılayıcı	Toplama ve yayınlama	Dağıtık	Enerji korunumu	Rasgele	Değişken		
<b>LEACH</b>	Değişken	Sabit(1- atlama)	Direk bağlantı	Sağlar	Sabit	Algılayıcı	Yayınlama	Dağıtık	Enerji korunumu	Rasgele	Sabit		
<b>FLOC</b>	Değişken	Sabit(2- Üniteli)	Direk bağlantı	Sağlar	Yeniden yerleşebilir	Algılayıcı	Toplama ve yayınlama	Dağıtık	Ölçekleme ve hata toleransı	Rasgele	Sabit		

## **BÖLÜM 3. AYRIK OLAYLI MODELLEME ve BENZETİM**

### **3.1. Giriş**

Mühendislik dallarının hemen hemen hepsinde modelleme ve benzetim araçları kullanılmaktadır. Genel olarak modelleme ve benzetim çoğu kez Sistem Teorisi, Kontrol Teorisi, Yapay Zeka Araştırmalarının bir alt kümesi olarak görülmektedir [64].

DEVS ayrık olaylı sistemler için sistem teorisi ve modelleme kavramlarını içeren ayrık olaylı sistem tanımlama yöntemidir [65].

Bu bölümde modelleme ve benzetim yaklaşımlarının problemleri, getirdiği çözümler ve DEVS yaklaşımı incelenmektedir.

### **3.2. Ayrık Olaylı Sistemler ve DEVS Modelleme ve Benzetim Teorisi**

DEVS (Discrete Event System Specification - DEVS), ayrık olaylı sistemler için sistem teorisi ve modelleme kavramlarını daha özel bir biçimde ifade eden ayrık olaylı bir sistem tanımlama yaklaşımıdır. Sadece ayrık olaylı modeller için değil aynı zamanda ayrık zamanlı ve diferansiyel denklemlilerle de ifade edilen davranışları uyarlamak için bir hesaplama temeli oluşturması nedeni ile önemlidir [65].

Paralel DEVS, modern hesaplama teorisinde çok önemli bir yere sahip olan paralelliğin faydalı bir şekilde kullanılmasını engelleyen ve ilk zamanlardaki bilgisayarların sıralı operasyonları nedeniyle ortaya çıkan sınırlandırmaları ortadan kaldırmaktadır [65]. Bu bölümde daha sonraki kısımlarda tartışılacak DEVS'in

nesneye-yönelik çerçeveye uygulanması işlemine hazırlık amacı ile DEVS modelleme yaklaşımı konusu tanıtılmaya çalışılacaktır.

### 3.4.1. Ayrık olaylı modelleme yaklaşımı

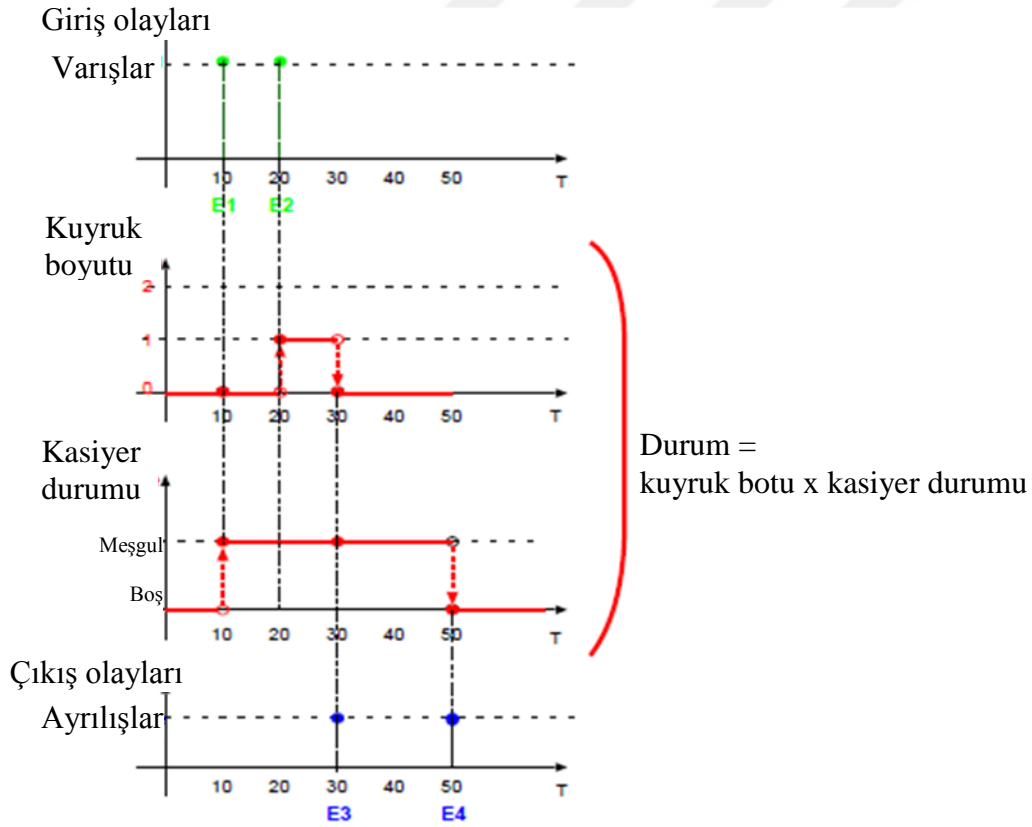
Zaman ekseninin sürekli olduğu, ancak sadece sınırlı bir zaman periyodunda sonlu sayıda olayların meydana geldiği sistemler, ‘ayrık olay’ olarak adlandırılan bir soyutlama seviyesinde ele alınmıştır. Bu olaylar sistemin durumunda bir değişikliğe neden olabilmelerine karşılık, olaylar arasında sistemin durumu değişmez. Bu sebeple ‘ayrık olaylı modeller’, sistemin durumunun zaman üzerinde sürekli değiştiği sürekli modellerden ayrılırlar [65].

Ayrık olaylı modelleme yaklaşımları yüksek bir soyutlama düzeyinde ele alınır. Bu soyutlama düzeyinde sistemin gerçek ortamdaki durumu DEVS ortamına aktarılır.

Yüksek bir soyutlama düzeyi, gerçek dünya davranışıyla ilgisi olmayan benzetim araçlarını getirebilir. Özellikle birden fazla olayın aynı zaman diliminde meydana geldiği olay çakışmaları, ayrık olaylı modelin yeteri kadar detaylandırılmamasından kaynaklanabilir. DEVS modelleme yaklaşımı ve onun türevleri, oluşan olay çakışmalarını başarılı bir şekilde yönetir ve bu durumların ortadan kaldırılması konusunda güvenilir bir çözüm yolu sunar [65].

Şekil 3.1’de kasiyer/kuyruk sisteminin örnek bir davranışı görülmektedir. Müşteri gelişi ile gerçekleşecek işlem tarif edilmiştir.

Fiziksel seviyede kasiyer bir anda bir müşteriye hizmet vermektedir. Kasiyerin meşgul olduğu durumda müşteriler kuyrukta beklemektedir. Burada sistem durumu müşterilerin özelliklerine bakmaksızın meydana gelir (yaş, alınan ürün adedi, vb.). Böylece model kuyruk durumu doğal sayılardan oluşan bir kuyruk uzunluğu manasına gelir. Kasiyer “boş veya “meşgul” durumunda olabilir. Sistem dinamikleri kuyruk ve kasiyer durumları ile kararlaştırılır. Kuyruk idaresi, ‘ilk giren ilk çıkar’(First In First Out - FIFO) yöntemine göre ele alınır [4].



Şekil 3.1. Kasiyer / kuyruk sisteminin davranış örneği [66]

### 3.4.2. Ayrık olaylı sistemlerde kullanılan terimler

Ayrık olaylı benzetimde kullanılan terimler aşağıda tanımlanmaktadır [65,4]:



An: Bir nesnenin en az bir karakteristiğine atanabilen bir zaman değeridir.

Aralık: Birbirini izleyen iki an arasındaki süredir.

Zaman Periyodu (time-span): Bir veya daha fazla aralığının ardı ardına gelmesidir.

Nesnenin Durumu: Belirli bir anda bir nesnenin tüm karakteristik değerlerinin listesidir.

Aktivite: Bir aralık boyunca bir nesnenin durumudur.

Olay: Bir nesnenin durumunda bir anda olan bir değişimdir ve o ana kadar olması engellenen bir aktiviteyi / işlemi başlatır. Olayın oluş şartı yalnızca sistem zamanına bağlıysa, bu olay 'belirlenmiş' bir olaydır ve bu özelliğe sahip modeller, benzetim modellemede, 'zaman olayı' olarak isimlendirilir. Zamana bağımlı olmadan sistemin şartlarına bağlı olarak gelişen / oluşan olaylar tesadüfidir (contingent) ve bu tür olaylar modellemede, 'durum olayı' olarak isimlendirilir.

Nesne aktivitesi: Durum değişimlerine sebep olan birbirini izleyen iki olay arasındaki nesnenin durumudur. Diğer olaylar, diğer nesnelerdeki durum değişimlerine göre meydana gelebilir.

İşlem (process): Bir zaman periyodu boyunca bir nesnenin birbirini takip eden durumlarıdır. Bir veya daha fazla nesne aktivitesinin ardı ardına gelmesi, 'işlem' olarak adlandırılabilir.

Belirli bir problem için, olayların ne olduğunu belirlemek / anlayabilmek için aşağıdaki adımlar takip edilebilir:

- Nesnelere ve nesne karakteristiklerinin tespit edilmesi,
- Sistemin karakteristiklerinin belirlenmesi ve

- Bir olay olarak, herhangi bir karakteristik değerde değişimin nedenlerinin tanımlanmasıdır.

Çoğu zaman sayıcılar, minimum değerler, maksimum değerler, ortalamalar, değişkenlerin frekans dağılımları, vb. başarıml ölçütlerinin hesaplanmasına yardımcı olacak ilave durum değişkenleri modele eklenir. Ayrık olaylı benzetimde, ortalama kuyruk uzunluğu ve kaynakların kullanımı gibi başarıml ölçütleri sık sık kullanılır. Başarıml ölçütleri bir benzetim çalışmasının sonucunda elde edilen çıkışlardır.

Aşağıda, ayrık olaylı benzetimin temelini oluşturan dünya görüşleri sunulmaktadır [65,4].

### **3.4.3. Ayrık olaylı benzetim stratejileri**

Benzetim dillerinde yaygın olarak kullanılan üç farklı ayrık olaylı benzetim stratejisi bulunmaktadır [67].

- Olay zamanlama (event scheduling),
- Aktivite tarama (activity scanning) ve
- Süreç etkileşimi (process interaction) .

“Dünya görüşleri” olarak da adlandırılan bu stratejilerin her biri diğerlerinden daha doğal bir şekilde ifade edilebilir bir model tanımlama biçimi ortaya koyar. Bir başka deyişle, her bir benzetim stratejisi dünyanın çalışma şeklini ele alırken orijinal yaklaşımlara sahiptirler. Yukarıda isimleri verilen stratejilerden sonuncusunun ilk ikisinin bir kombinasyonu olması yanında günümüzde geliştirilen benzetim ortamlarının çoğu bu stratejilerin bir kombinasyonunu kullanmaktadır [65].

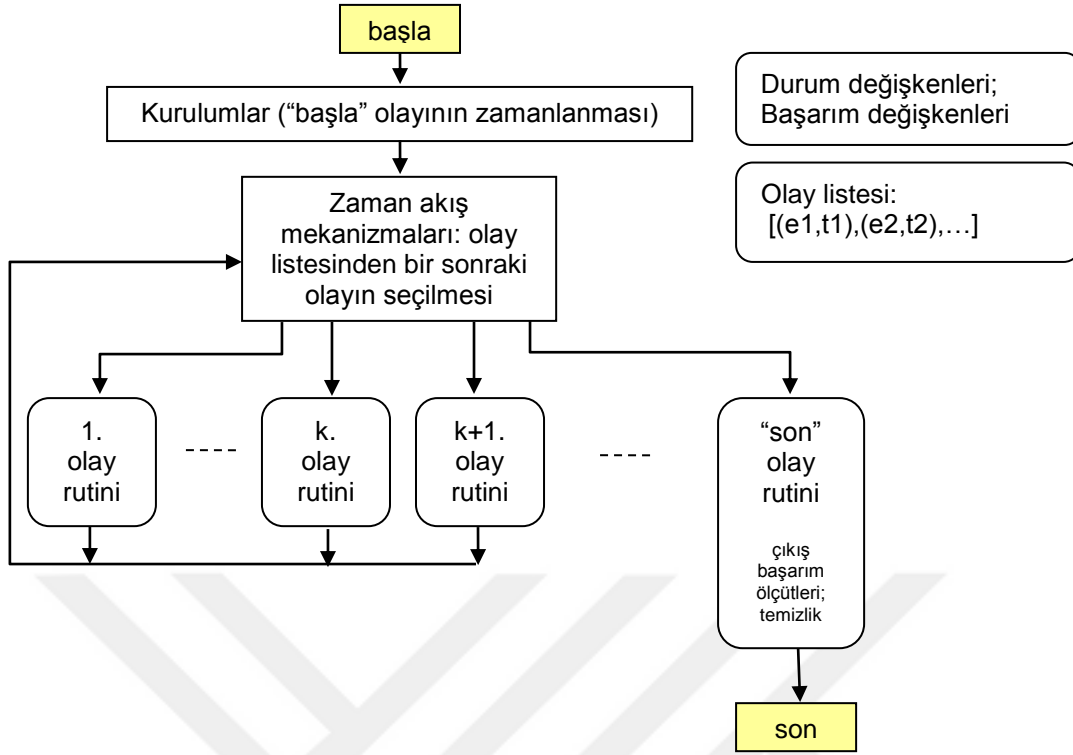
#### **3.4.3.1. Olay zamanlama benzetim stratejisi**

Olaya yönelik modelleme ve benzetimde bütün olaylar önceden planlanır. Bir olayın planlanması, olayın gerçekleşmesi için gerekli bütün şartlar önceden bilinebildiği

durumlarda mantıklı olur. Olay zamanlama benzetim stratejisinde model; her bir olay için olayın durum üzerindeki ve sistemin gelecekteki davranışı üzerindeki etkisini tanımlar. Bu, yeni olayların gelecekteki bir zamana programlanmasıyla yapılır [65].

Şekil 3.2’de görüldüğü gibi bir olay zamanlama benzetim çekirdeği iki (global) veri yapısı kullanır. Veri yapılarından bir tanesi modelde bildirilen durum değişkenlerini, diğeri ise artan zamanla ve azalan öncelikte sıralı bir olay listesinde zamanlanmış olay bildirimlerini içerir. Bir olay programlandığı zaman listede en sona eklenir. Öncelikler aynı zamanda gerçekleşen (çakışan) olaylar arasında seçim yapmak için kullanılır. Durum değişkenleri, durum değişkenlerinin ve kombinasyonlarının minimumunu, maksimumunu, ortalamasını ve standart sapmasının hesaplanması için ilave başarımla değişkenleriyle artırılabilir. Bir olay zamanlama çekirdeği, olayları ‘olay listesi’ şeklinde düzenleyerek (artan zaman sırasına göre), listenin en üstündekini kaldırıp işleyerek ve bunu liste tamamen boşalana kadar tekrarlayarak çalışır. Olay bildirimindeki olay zamanı, benzetim zamanını ilerletmek için kullanılır. Olay bildirimindeki olay türüne bağlı olarak, uygun olay bildirim yapılabılır. Bu işlem, olay bildirimlerini olay listesine koyarak sistemin durumunu değiştirebilir ve gelecekte yeni olaylar zamanlayabilir. Olay zamanlama yaklaşımında:

- Sistemin kurulumu ve ön zamanlamalı olaylar bir ‘başlangıç’ olayına konulabilir. Bu olay otomatik olarak olay listesine yerleştirilir ve daha sonra normal bir olay gibi işlenir.
- Belirli bir benzetim zamanında benzetimin sonlandırılması, benzetim prosedürü tarafından son olay olarak işlenecek olan özel bir ‘son’ olayının zamanlanmasıyla halledilir. Bu olay başarımla ölçümleri çıkışını ve elde edilen diğer istatistikleri üreten işlem yönergelerini içerir.

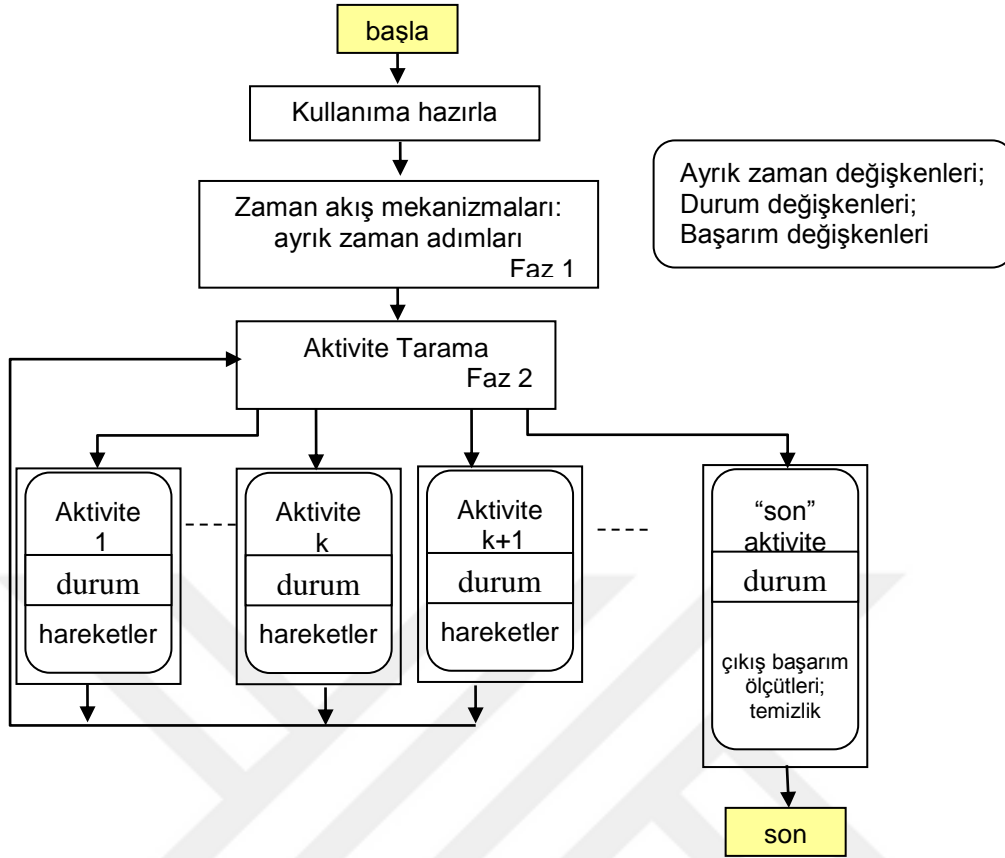


Şekil 3.2. Olay zamanlama benzetim çekirdeği [65]

Şekil 3.2’de gösterilen bir olay zamanlama modeli, benzetim zamanını ilerleten, olay listesini ve sistem durumunu güncelleyen yinelemeli bir benzetim prosedürüyle benzetim edilir [65].

### 3.4.3.2. Aktivite tarama benzetim stratejisi

Aktivite tarama yönteminde, bir takım aktiviteleri etkinleştiren şartlar tanımlanır. Bu yaklaşım, Prolog gibi bildirimli yapay zekâ (Artificial Intelligence - AI) dillerinde kullanılan yaklaşımla benzerlikler taşır. Bir aktivite tarama benzetimi zamanı ilerletmek için ayrık zaman adımını kullanır. Aktivite tarama safhasında, çözücü (solver) aktivitenin şartı sağlayıp sağlamadığını kontrol eder ve eğer şart doğruysa işler. Bu tarama işlemi en azından bir aktivite şartı doğru olarak değerlendirilene kadar devam eder. Eğer aktivitelerin hiçbiri şartı sağlamazsa, zaman akış safhası zamanı ilerleterek tekrar çalıştırılır. Petri ağları, aktivite tarama tabanlı bir modelleme yaklaşımına örnektir [65]. Şekil 3.3’de aktivite tarama benzetim stratejisi gösterilmiştir.

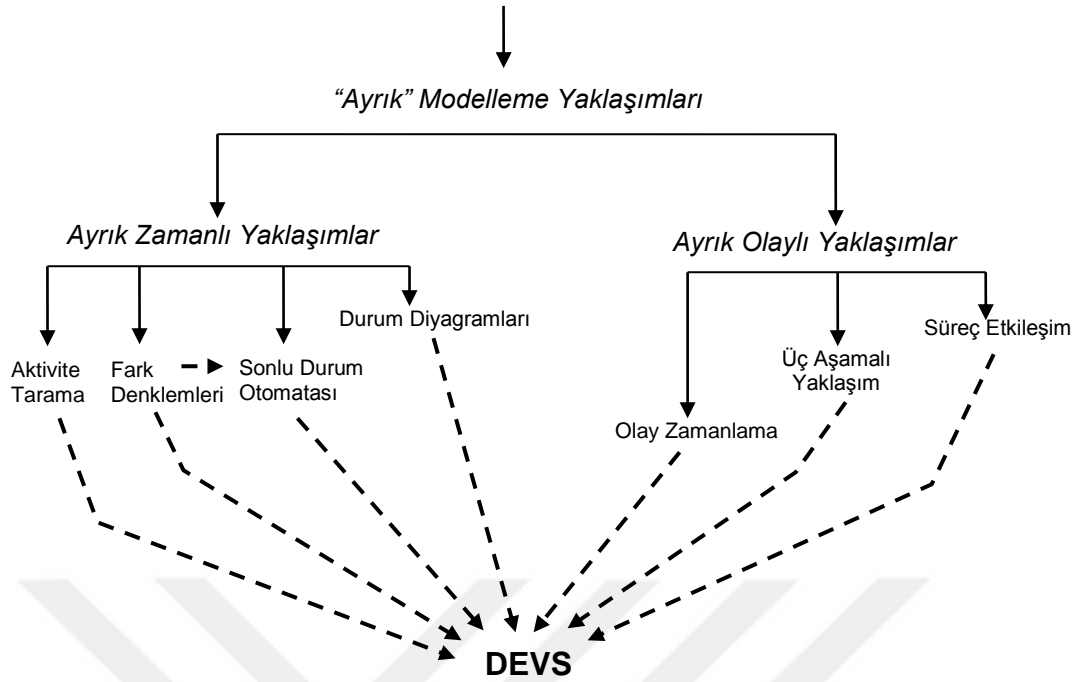


Şekil 3.3. Aktivite tarama benzetim stratejisi [65]

### 3.4.3.3. Süreç etkileşim benzetim stratejisi

Aktivite tarama sabit bir zaman adımını kullanır ve bu pek çok durumda verimli değildir. Davranışı doğru bir şekilde modellemek için iki olay arasındaki zaman aralığı olabildiğince küçük seçilmelidir. Diğer taraftan, bazı olaylar zaman bakımından çok seyrek olabilir. Bu gibi uzun zaman aralıklarında, aktivite tarama benzeticisi, şartlar değişmemesine rağmen zamandaki her noktada tüm şartları gereksiz bir şekilde kontrol eder.

Süreç etkileşim benzetim stratejisinde, aktivite tarama yaklaşımı olay zamanlama yaklaşımıyla birleştirilir. Aktiviteler, olay zamanlama benzetim stratejisindeki gibi gelecekteki bir zamana programlanır. Buna ek olarak, olay zamanlarında, bütün aktivite şartları aktivite tarama benzetim stratejisindeki gibi kontrol edilir. Şekil 3.4’de benzetim stratejileri sınıflandırılmıştır.



Şekil 3.4. Benzetim stratejilerinin sınıflandırılması [65]

### 3.4.4. Ayrık olaylı benzetim stratejileri arasındaki ilişkiler

Şekil 3.4, farklı ayrık yaklaşımlar arasındaki ilişkilerin bir özetini göstermektedir. Şekilde sol tarafta zaman akış mekanizmasının sabit bir zaman ilerlemesi şeklinde olduğu yaklaşımlar sağ tarafta ise, ‘ayrık olaylı’ zaman akışına (saat, sadece olayların olduğu zamanlarda ilerler) sahip yaklaşımlar görülmektedir.

Ayrık olaylı modelleme yaklaşımları modüler olmayan bir formdadırlar. Yani olay işleyicisi (event handler), aktiviteler, işlem blokları, vb. model bileşenleri bir takım arabirimler aracılığıyla kendi ortamıyla etkileşen varlıklar değildir. Bundan çok, direk olarak global durum değişkenleri ile birlikte diğer bileşenleri etkilerler.

Şekil 3.4’deki kesikli çizgiler, bir kaynak modelleme yaklaşımında tanımlı bir modelin hedef yaklaşımda tanımlı aynı modele dönüşümünü göstermektedir. Orijinal ve dönüşmüş modeller, aynı başlangıç şartlarında benzetim edildiğinde aynı durum eğrilerini üretirlerse “denk” olarak adlandırılırlar. Şekilde tüm dönüşümler DEVS yaklaşımına yönelmiştir. Tüm bu dönüşümlerin özü, modüler olmayan

tanımlamalardan modüler bir 'DEVS Birleşik modelinin' kurulmasıdır. DEVS birleşik modeli, modüler bileşenler arasındaki bağlantılar aracılığıyla bağımlılıklar açık bir şekilde gösterilerek elde edilir [65].

### 3.5. Atomik DEVS Modelleme Yaklaşımı

Atomik DEVS modelleme yaklaşımı, sistemin ayrık olaylı davranışının değişik yönlerini tanımlayan bir yapıdadır. DEVS atomik model tanımı, değişken zaman periyotlarına sahip parçalı sabit eğriler şeklinde durumları (değişken değerler) ve ilişkili zaman eksenini tanımlar. Atomik model tanımı yeni durum değerlerinin nasıl üretileceğini ve ne zaman bu yeni değerlerin etkin olacağını da açıklar. Paralel bir atomik DEVS modeli aşağıdaki yapıdadır [6,5]:

$$M = \langle X, S, Y, \delta_{int}, \delta_{ext}, \delta_{conf}, \lambda, ta \rangle$$

Burada;

X, giriş değerleri kümesi,

S, durumlar kümesi,

Y, çıkış değerleri kümesi,

$\delta_{int} : S \rightarrow S$  dahili geçiş fonksiyonu,

$\delta_{ext} : Q \times X \rightarrow S$  harici geçiş fonksiyonu,

burada;  $Q = \{(s,e) | s \in S, 0 \leq e \leq ta(s)\}$  toplam durum kümesi,

e, en son olan geçişten bu yana geçen süre,

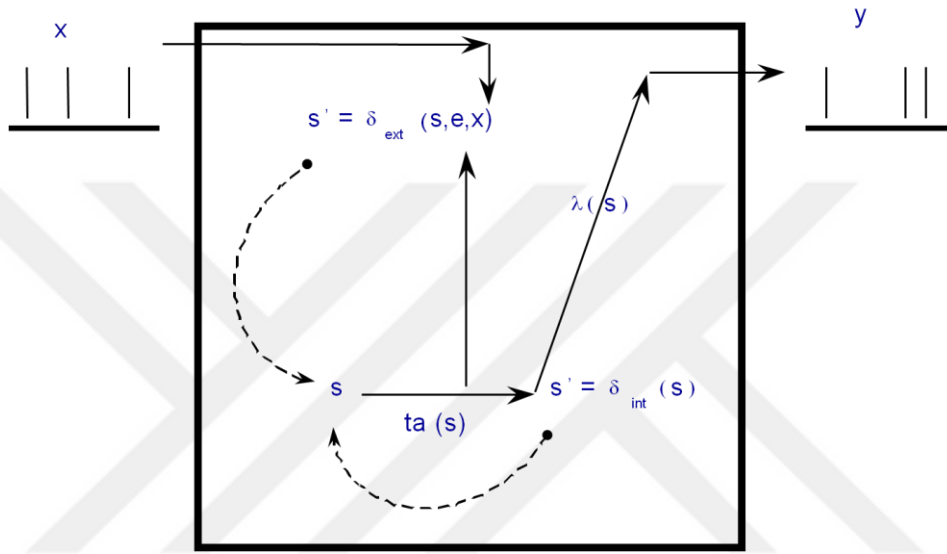
$\delta_{ext} : Q \times X \rightarrow S$  çakışma (confluent) geçiş fonksiyonu,

$\lambda : S \rightarrow Y$  çıkış fonksiyonu,

ta:  $S \rightarrow \mathbb{R}^+_{0,\infty}$  zaman ilerleme (time advance) fonksiyonu, 0 ve  $\infty$  arasındaki pozitif reel sayılar kümesidir.

Bu elemanların yorumu Şekil 3.5'de görülmektedir. Herhangi bir anda sistem bir 's'durumundadır ve hiçbir harici olay meydana gelmemesi durumunda, sistem ta(s) zamanı süresince 's'durumunda kalır. Yukarıda yapılan tanımdan anlaşılacağı üzere 'ta(s)' bir reel sayı olmasına rağmen '0' ve ' $\infty$ ' değerlerini de alabilmektedir. Zaman ilerleme fonksiyonu ta(s) = 0 olduğunda, 's'durumunun süresi araya başka olaylar giremeyecek kadar çok kısadır – bu durumda 's'durumunun bir geçici durum

olduğunu belirtmek yanlış olmaz. İkinci durumda ( $ta(s) = \infty$  olduğunda), harici bir olay bu durumu bozmadıkça sistem sonsuza kadar 's'durumunda kalır ve bu durumda 's'pasif bir durum olarak tanımlanır. Bir durumda kalma süresi dolduğu zaman (geçen süre  $e=ta(s)$  olduğu zaman), sistem  $\lambda(s)$  değerini çıkış olarak verir ve  $\delta_{int}(s)$  durumuna geçer. Bu anda, çıkışın dâhili geçişlerden hemen önce üretildiğine dikkat edilmelidir.

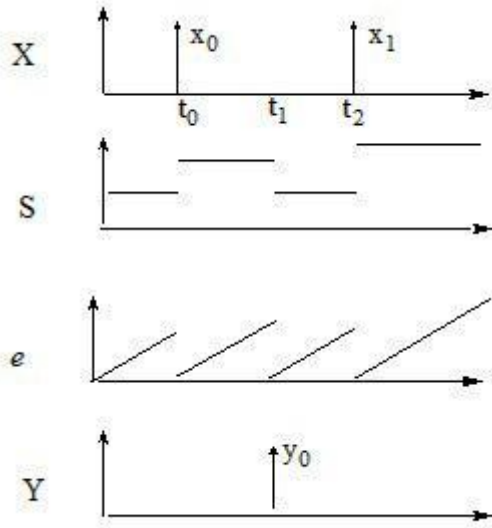


Şekil 3.5. DEVS işleyiş mekanizması.

Eğer bir  $x \in X$  harici olayı bitiş zamanından önce meydana gelirse (sistem  $e \leq ta(s)$  ile  $(s, e)$  durumundaysa), sistem  $\delta_{ext}(s, e, x)$  durumuna geçer. Bu nedenle dâhili geçiş fonksiyonu, en son geçişten itibaren hiçbir olay olmadığı zaman sistemin yeni bir duruma geçmesine neden olur. Bu durum, 'x' girişi, 's'şimdiki durumu ve sistemin bu durumda ne kadar kaldığını gösteren 'e' tarafından belirlenir. Her iki durumda da sistem, yeni bir  $ta(s')$  sükûnet zamanı ile yeni bir  $s'$  durumundadır ve aynı olay devam eder.

Aşağıda Şekil 3.6'da ayrık olaylı sistemde giriş, durum, geçen süre ve çıkışların zamana bağlı gösterimi görülmektedir [67].





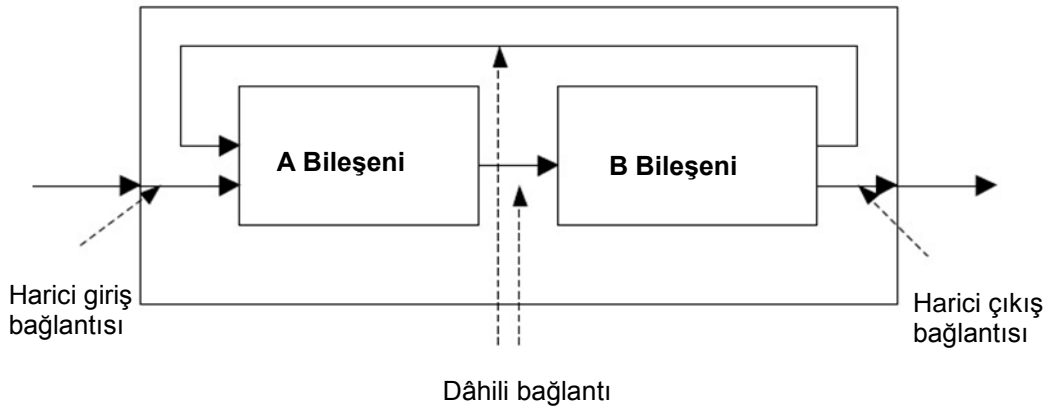
Şekil 3.6. Ayrık olaylı sistemde giriş, durum, geçen süre ve çıkışlar

### 3.6. Birleşik DEVS Modelleme Yaklaşımı

Birleşik model, bir takım bileşen modellerden ve bunların birbirleriyle olan bağlantılarından oluşur. Daha öncede ifade edildiği üzere bileşenler ‘atomik’ veya ‘birleşik’ olabilir. Birleşik modelin davranışı bileşenlerin davranışıyla ve / veya bağlantısıyla tanımlanabilir. Bağlantı üç türe ayrılır: Harici giriş bağlantısı, harici çıkış bağlantısı ve dâhili bağlantı (Şekil 3.7).

Harici giriş bağlantısı; birleşik modelin kendisi ve bileşenlerin bir veya birden fazlasının arasında olur. Harici çıkış bağlantısı ise; bileşen çıkışları ile birleşik modelin çıkışları arasındaki bağlantıdır. Dâhili bağlantı; bileşen çıkışları ve bileşen girişleri arasında yapılır.

DEVS formalizmi temel olarak ayrık olaylı sistemlerin tanımlanmasında kullanılır. DEVS formalizminin önemli bir özelliği de modelleme ve benzetim bölümleri arasında açık bir ayrıma izin vermesidir. Bunun anlamı herhangi bir sistemin davranışını fazından bağımsız modelleyebiliriz [5].



Şekil 3.7. Birleşik DEVS yaklaşımında bağlantılar

Birleşik bir DEVS modelleme yaklaşımı aşağıdaki yapıdadır;

$$CM = \langle X, Y, D, \{M_i\}, EIC, EOC, IC, Select \rangle$$

Burada;

$X, Y$ : giriş ve çıkış kümeleridir

$D$ : birleşik modelin bileşenler kümesidir

Her  $i \in D$  için,

$M_i$ : atomik veya birleşik olabilen bir bileşenin DEVS modelidir;

$EIC \subseteq X \times \cup_i X_i$ , harici giriş bağlantı ilişkisi;

$EOC \subseteq \cup_i Y_i \times Y$ , harici çıkış bağlantı ilişkisi;

$IC \subseteq \cup_i Y_i \times \cup_j X_j$ , dâhili bağlantı ilişkisi;

$Select: 2^{\{M_i\}} - \emptyset \rightarrow \{M_i\}$ , eşitlik fonksiyonudur.

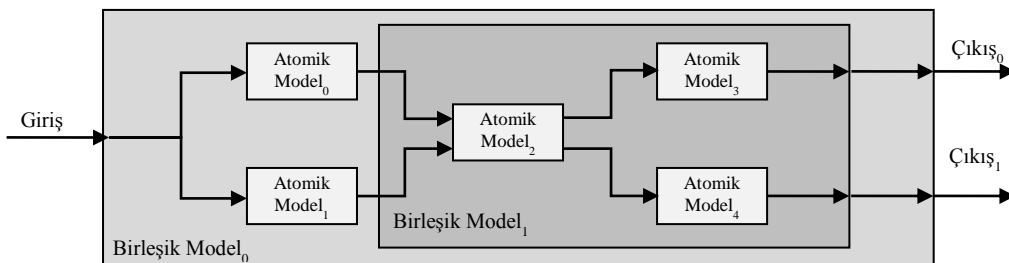
DEVS, modelin davranışının tanımlanması kadar sistem modelini de çeşitli düzeylerde tanımlayan güçlü bir modelleme ve benzetim aracıdır. DEVS formalizminin önemli bir özelliği her bir modelin otomatik olarak benzetimini desteklemesidir. DEVS modelleme ve benzetimi birbirinden ayırır. Bu yüzden her bir model özel bir benzetici olmaksızın çalıştırılabilir. Her bir model, davranışını yöneten bir benzeticiye, her bir birleşik modelde, bileşenlerinin uyumlu çalışmasını sağlayan bir koordinatöre sahiptir. DEVS benzetim sistemlerinin mimarisi hiyerarşik ve modüler DEVS formalizmi ile benzetim kavramlarından gelir [5].

### 3.7. Hiyerarşik Model Tasarımı: DEVS Birleşim Çerçevesi

Birleşik bir model DEVS modelleme yaklaşımında temel bir model olarak ifade edilebilir. Bu temel model daha büyük bir birleşik model içinde kullanılabilir. Bu, DEVS yaklaşımının hiyerarşik model yapısı için gerekli olan birleşim altında kapalı olduğunun göstergesidir. Bir birleşik modeli denk bir temel model olarak ifade etmek, tüm davranışı vermek üzere bileşenlerin etkileşmesi yoluyla yapılır. Daha önce de ifade edildiği gibi, birleşik modeller atomik modüller gibi daha büyük sistemlerde bir bileşen olabilir ve böylece hiyerarşik bir yapı olur (Şekil 3.8) [4].

Bir bağlantı tanımının bir takım modellere uygulanması yoluyla nasıl bir birleşik model üretildiği Şekil 3.8’de gösterilmektedir. Bu modeli daha büyük bir sistemde yeni bileşenlerle birlikte bir bileşen olarak kullanarak ve bağlantı bilgisini de ekleyerek, hiyerarşik bir birleşik model elde edilir.

Şekil 3.8’de birleşik modelin hiyerarşik yapısı görülmektedir. BirleşikModel<sub>0</sub> bir giriş portu, iki çıkış portu, iki adet atomik model (AM<sub>0</sub>, AM<sub>1</sub>) ve bir birleşik modele sahiptir. Birleşik Model<sub>1</sub> atomik model AM<sub>2</sub>, AM<sub>3</sub> ve AM<sub>4</sub>’ü kapsamaktadır. Dahili bağlantılar (AM<sub>0</sub>→CM<sub>1</sub> ve AM<sub>1</sub>→CM<sub>1</sub>), harici giriş bağlantıları (IN→AM<sub>0</sub> ve IN→AM<sub>1</sub>), harici çıkış bağlantıları (CM<sub>1</sub>→OUT<sub>0</sub> ve CM<sub>1</sub>→OUT<sub>1</sub>) ve alt modelin etkisi (CM<sub>1</sub>={AM<sub>0</sub>, AM<sub>1</sub>} veya AM<sub>0</sub> and AM<sub>1</sub> CM<sub>1</sub> etkiler [5].



Şekil 3.8. DEVS birleşik modeli

### 3.8. Modelleme ve Benzetim Araçları

#### 3.8.1. Ağ benzetimleri ve problemleri

İletişim protokolleri, bağlantı teknolojileri, trafik akışları ve yönlendirme algoritmalarının birlikte şekillendirdiği ağların tasarım süreci oldukça karmaşıktır [3]. Karmaşıklığın azaltılması ile ilgili açık bir yaklaşım, modelleme ve benzetim tekniklerinin kullanılmasıdır [3]. Donanım eksiklikleri, ekipmanların geliştirilmesindeki zorluklar, gerçek dünyadaki karmaşık yapılandırma, güvenilir istatistiklerin toplanması, vb. sebeplerden dolayı, yeni fikirleri, ağın davranışını ve başarımını analiz etmede ağ benzetimi araştırmacılar ve mühendisler için önemli bir araçtır [68].

İnternet'in mevcut boyutuna ve karmaşıklığına erişmeden önce, küçük homojen ağlarda yönlendirme algoritmaları tasarlamak, test etmek ve modelleme / benzetim yoluyla prototip ağları incelemek kısmen mümkündür. İnternet gibi büyük ölçekli ağlar üzerinde hatalara karşı hassasiyet derecesini tespit etmek, daha sağlam yönlendirme algoritmaları tasarlamak / test etmek amacıyla deney yapmak günümüzde mümkün olmamakta, bütün ağ sisteminin davranışını çözmek veya hatalar karşısında ağın çökmesini önlemek gibi problemleri çözmek konusunda mevcut benzetim araçları yetersiz kalmaktadır. İnternet ortamının davranışının modellerini oluşturmak amacıyla yeterince veri ve analiz yöntemi bulunsaydı, kritik hataların ve yapısal zayıflıkların tespit edilebilmesi yanında olası hatalara karşı da önlemler alınabilirdi [69].

Günümüzde çeşitli modelleme ve benzetim araçları ağ tasarımı için şirketler / akademik araştırma grupları tarafından planlanan amaca göre pratik veya eğitim amaçlı olarak kullanılmaktadır. Bu araçların işlevselliği, avantajları, süreçleri ve özellikleri açısından detaylı olarak incelenmesi gereksinimi bulunmasına rağmen hemen hemen tüm ağ benzetim araçlarının benzer yolla (bazen tamamlayıcı hedefler) çalıştığı açıktır [3].

İletişim ağlarını doğrudan modellemek için kullanılacak yazılımlar (COMNET, OPNET, Ns-2, Ns-3, GloMoSim, OMNET++, JNS, vb.) yanında donanım bileşenlerini modellemek için kullanılacak araçlar (VHDL, Verilog, SystemC, vb.) bulunmaktadır [4,5]. En yaygın kullanılan ağ benzeticileri olan Ns-2, Ns-3, OPNET, OMNET++, vb. benzeticiler küçük boyutlu ağları çalışmak için ideal platformlardır. Bu programları büyük ölçekli sistemlerin modellenmesi ve benzetiminde kullanmanın zor olmasının yanında, değişik teknolojilerin içine katıldığı sistemleri modelleme yeteneğine de sahip değildirler [6]. Ağ sistemlerinin boyutlarının süreç içerisinde üstel olarak artması nedeni ile statik topoloji üreten bu benzeticiler yetersiz kaldılar ve gelişen sistemlerin başarımını doğru bir şekilde test edemez / ölçemez duruma düştüler. Belirtilen eksikliklerin ortaya çıkmasının sebepleri olarak; basit benzeticilerin yapısal sınırlamaları ve büyük ölçekli karmaşık yapıli ağların yetersiz bir şekilde analiz edilmesi sıralanabilir. Belirtilen kısıtlamaları / yetersizlikleri bir ölçüde karşılayabilen ve binlerce düğümü modelleyebilen GloMoSim, PDNS, vb. benzeticiler bulunsa da, bunların hiçbiri dinamik, gelişebilir, yeniden boyutlanabilir ve değişik trafik şartlarına uyarlanabilir (adaptif) bir ağı modelleyememektedir [6-8]. Ayrıca, bu tür benzeticilerin bileşenleri modüler ve hiyerarşik bir yapıda olmadığından bileşenlerin yeniden kullanımı, değişik uygulamalara uyarlanabilirliği ve hiyerarşik tasarımı zordur. Klasik benzeticilerin çalıştırıldıkları bilgisayarlarda oldukça yüksek kaynak kullanım gereksinimleri, gelişmiş ve büyük uygulamaların meydana getirilmesini zorlaştırmaktadır.

Benzeticilerin farklı açıdan karşılaştırılması ile elde edilen özellikler, Tablo3.1’de görülmektedir. Farklı özellikler açısından tablodaki benzeticilerin incelenmesinden, mükemmel bir benzetim aracı olmadığı görülmektedir. Begg ve arkadaşları [70] tarafından yapılan çalışmada, maliyet ve süre göz ardı edildiğinde özel amaçlı hazırlanan benzeticilerin kullanıcı taleplerini karşılayabileceği vurgulanmıştır.

### **3.8.2. Benzeticiler ile ilgili problemlere çözüm önerisi**

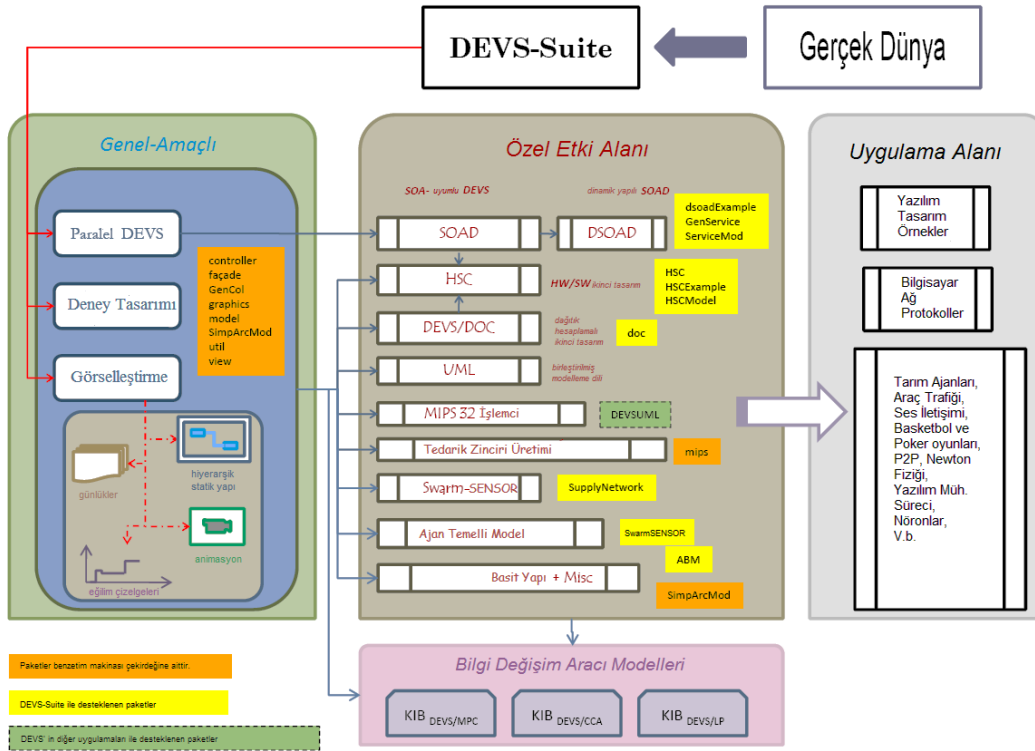
KAA’ların enerji, karmaşıklık, ölçeklenebilirlik, vb. problemlerinin çözülmesi amacıyla yönlendirme protokolleri tasarlamak, protokollerin uygulanacağı

topolojileri oluşturmak ve bu topolojileri oluşturan sistemlerin bileşenlerini test edip hangi seviyelerde soyutlama yapılacağına karar vermek için modelleme ve benzetim gerekmektedir. Yapılan çalışma, yukarıdaki gereksinimleri karşılamak üzere KAA'lara yönelik modelleme ve benzetim araçlarının tasarlanmasını ve uygulanmasını içermektedir.

Tablo 3.1. Ağ benzeticilerinin karşılaştırılması

Karşılaştırma Ölçütleri	Ns-2	Pdns	OPNET	OMNeT++	J-sim	SSFnet	GloMoSim	DEVS-Sensor
Nesne Yönelimli	Orta	Orta	Güçlü	Orta	Çok Güçlü	Çok Güçlü	Orta	Çok Güçlü
Ağ model kütüphanesi	Güçlü	Güçlü	Güçlü	Güçlü	Orta	Zayıf	Orta	Zayıf
Sonuçların analizi	Orta	Orta	Çok Güçlü	Zayıf	Zayıf	Zayıf	Güçlü	Çok Güçlü
Genişletilebilirlik	Orta	Orta	Güçlü	Güçlü	Çok Güçlü	Çok Güçlü	Çok Güçlü	Çok Güçlü
Uzman ihtiyacı	Çok Güçlü	Çok Güçlü	Zayıf	Çok Güçlü	Zayıf	Güçlü	Zayıf	Orta
Kurulum Kolaylığı	Zayıf	Zayıf	Güçlü	Orta	Çok Güçlü	Güçlü	Güçlü	Çok Güçlü
Dokümantasyon	Orta	Orta	Çok Güçlü	Güçlü	Zayıf	Zayıf	Orta	Orta
Erişilebilirlik	Çok Güçlü	Güçlü	Zayıf	Çok Güçlü	Çok Güçlü	Çok Güçlü	Zayıf	Çok Güçlü
Görsellik	Zayıf	Zayıf	Güçlü	Güçlü	Orta	Çok Güçlü	Çok Güçlü	Çok Güçlü
Kullanıcı tabanı	Çok Güçlü	Zayıf	Güçlü	Güçlü	Orta	Zayıf	Zayıf	Zayıf
Ölçeklenebilirlik	Zayıf	Çok Güçlü	Orta	Orta	Güçlü	Çok Güçlü	Çok Güçlü	Çok Güçlü
Başarım	Güçlü	Çok Güçlü	Orta	Orta	Güçlü	Çok Güçlü	Orta	Çok Güçlü
Rastgelelik	Çok Güçlü	Çok Güçlü	Zayıf	Güçlü	Zayıf	Zayıf	Zayıf	Çok Güçlü
Hata Modelleme	Çok Güçlü	Çok Güçlü	Çok Güçlü	Orta	Zayıf	Orta	Zayıf	Çok Güçlü
Web Erişim	Yok	Yok	Yok	Yok	Güçlü	Yok	Yok	Çok Güçlü

Bir KAA sisteminin modellenmesi işlemi; ağ bileşenleri belirlenmesi, bu bileşenlerin yazılım nesnelere oluşturulması, yazılım nesnelere işlem yapan düğümlere ile bağlantılanması ve KAA topolojilerine yönlendirme protokolünün uygulanmasıdır. Yönlendirme algoritmalarının test edilmesi, büyük ölçekli KAA'ların analizi, modellenmesi, vb. ileri KAA uygulamalarını gerçekleştirmek amacıyla modellenen sistemi oluşturan parçalar ve bileşenler 'Java' programlama dili kullanılarak gerçekleştirilmiştir. DEVS 'birleşik model'(coupled model) kavramı kullanılarak sistem bileşenleri birbirine bağlanıp, benzetim deneyleri ile model davranışı üzerinde gözlem yapmak amacıyla deneysel çerçeve [6] aracı kullanıldı. Java dilinde yazılan DEVS-Suite [71] modelleme ve benzetim ortamında, KAA'lar için geliştirilen oğul zekası tabanlı yönlendirme algoritması çalıştırılmak üzere geliştirilen ortam 'DEVS-Sensor' olarak adlandırıldı. Şekil 3.9'da DEVS-Sensor ortamının geliştirilmesinde kullanılan DEVS-Suite ve teknolojileri gösterilmiştir.



Şekil 3.9. DEVS-Suite benzetim altyapısı

Kolay tasarım ve anlaşılır arabirimiyle model tasarım ve eğitiminde etkili bir araç olan DEVS-Suite ortamı; nesneye-yönelik modellemeyi, eş zamanlı paralel çalışan benzetimleri, etkileşen benzetim nesneleri arasında uyumluluğu ve web tabanlı benzetimleri olanaklı kılmaktadır [6]. DEVS modelleme ve benzetim yaklaşımının ve Java programlama dilinin sağladığı esneklik, değişen ortama adapte olabilen zeki bileşenlerin tasarımını kolaylaştırmaktadır. DEVS-Suite'in nesneye yönelik yapısı, bir ağı oluşturan düğümlerin, linklerin, yazılım varlıklarının ve deneysel çerçevelerin modüler bir yapıda tasarımını, yeniden kullanımını ve sistemlerin sistemini oluşturmayı kolaylaştırmaktadır [67].





## **BÖLÜM 4. BEEWS: OĞUL ZEKASI TABANLI YENİ BİR ÖLÇEKLENEBİLİR YÖNLENDİRME PROTOKOLÜ**

### **4.1. Giriş**

Sosyal canlılardan esinlenilerek geliştirilen yönlendirme protokolleri düğümlerin ağda dolaşan gözcü görevliler aracılığıyla ağ topolojisinin tamamından haberdar olmadan çok merkezli yönlendirme kararı vermesine olanak sağlayan algoritmalarlardır [72].

Arılar sosyal canlılar içinde yol bulma davranışı olarak en üstün yeteneğe sahip canlılardır [72]. Yiyecek kaynaklarına erişmek için en az enerjiyi kullanır, alternatif kaynakları da ihmal etmezler. Aynı zamanda arılar çoğaldıkça yeni kovanlar oluşturarak sayının artmasının çalışma sistemini etkilemesini önlerler. Kablosuz algılayıcıların en önemli problemi enerji tüketimi ve ölçeklenebilirliktir. Arılar ile bu bu açıdan kablosuz algılayıcılar arasında bir benzerlik söz konusudur. Bu tezde, bu benzerlikten yola çıkarak arıların enerji tüketimi modeli temel alınmıştır. Aynı zamanda büyük ölçeklere çıkabilmek için de arıların koloniyi yeni kovanlara bölme işlemi kümeleme olarak uygulanmıştır.

Bu bölümde, KAA'lar için oğul zekası tabanlı yeni bir yönlendirme algoritmasının işleyişi, geliştirilen algoritma mimarisi, paket yapıları, paketlerin çalışması ve kullanılan yöntemler anlatılacaktır.

### **4.2. KAA Model Sentezi ve BeeWS Algoritması**

Bal arılarından türetilen oğul zekası yaklaşımları ve bu etkileşimler bal arılarının bal toplama sürecinde kullandıkları nektar sahasını arayan gözcülerin çalışma şekli ele alınarak geliştirildi [72]. En önemli nokta “ Gerçek arılar bal toplama sürecinde en

uygun çözümleri bulurlar.” fikri ile temellendirilmiştir [72]. KAA için modelleme ve benzetimde sosyal canlılardan etkilenilerek geliştirilen yaklaşımlar verimli olmuştur.

Bal arıları topluluklarındaki nektar arama davranışları kendi kendine organize olabilme gibi sosyal böcek davranışlarını araştırmak için iyi bir örnektir [72]. Bal arıları polen ve nektar kaynakları ve kendi kendine organize olma yoluyla çeşitli görevlere işçi arıların tahsisi gibi konulara birlikte karar verirler. Bir kovadaki bal arıları arasındaki bu seçme ve ayırma işlemleri, herhangi bir merkezi yönetim otoritesinin yokluğunda yapılır. Merkezi olmayan ve eşzamanlı olarak, her bal arısı nektar konsantrasyonu gibi ölçütlerin yanı sıra gıda kaynakları için mesafe ve seyahat süresine dayalı basit kurallar kümesine uyarlar. Kovadaki gıda depolamak için sorumlu arıların sayısı gibi parametreler de dahil olmak üzere, bu ölçütler, bir nektar kaynağının karlılığını belirlemektedir. Koloni birden fazla nektar kaynağı ile karşılaşır, en karlı olanı toplayıcılar tarafından tercih edilmektedir. Toplayıcılar nektar toplama süreci boyunca karlılık kriterlerini kullanarak nektar kaynakları arasında dağıtılır. Eğer bir yerde nektar miktarı değişirse, nektar kaynağının önemi tüm koloni için azaltılır. Ayrıca, koloninin küçük bir kısmını oluşturan bal arılarına nektar araştırmacı gözcüler denir. Gözcüler çevredeki mevcut nektar sahalarını izler ve zengin kaynakları gözlemler [84]. Karlılık kriterlerine göre gıda kaynakları için toplayıcı arıların atama işlemi bal arıları içinde gözcü-işe alma (scout-recruit) mekanizması olarak bilinir. En iyi gözcü-işe alma mekanizmalarından birinin matematiksel modeli Seeley tarafından geliştirilmiştir [72].

Bal arılarının temel aktiviteleri arama, işletme ve nektar kaynaklarını terkden oluşur. Bal arılarındaki bu kolektif davranışlar şu aşamalardan oluşur: Kovandan yiyecek kaynağına seyahat, nektar gözcülüğü, yiyecek kaynağından nektar toplama, kovana dönüş, diğer arılara çevrede gözlenen durumu iletme, kovanda gerçekleşen değişimlere uyum sağlarken çevredeki değişimlere cevap verme.

Tablo 4.1’de görüldüğü üzere KAA’lar ile bal arıları arasında bir benzerlik söz konusudur. KAA’lar zengin nektar kaynaklarına ulaşmak için kaynakları araştıran bal arısı kolonilerine karşılık gelir. KAA’lardaki her düğüm bir kovana benzer.

Kapsam alanları bal arılarının besin bölgelerine karşılık gelir. KAA'lardaki düğümler veri ve kontrol paketlerinin kendi aralarında alış verişini yaparlar. Her KAA düğümü en iyi yolu bulmak için gözcüler gönderir. Veri paketleri kovana nektar taşıyan toplayıcılara karşılık gelir. Veri paketleri kontrol paketleri tarafından yönlendirme bilgisi oluşturulan kaynaktan hedefe doğru sevk edilirler.

Tablo 4.1. KAA ile bal arısı kolonileri arasındaki benzerlik

KAA	Bal arısı kolonileri
Düğüm	Arı kovanı
Kapsam alanı	Nektar/ kovan dan uçmak
Kontrol paketleri	Gözcüler
Veri paketleri	Veri oluşturucu, Veri taşıyıcı
Yönlendirme tabloları	Dans ve ipuçları

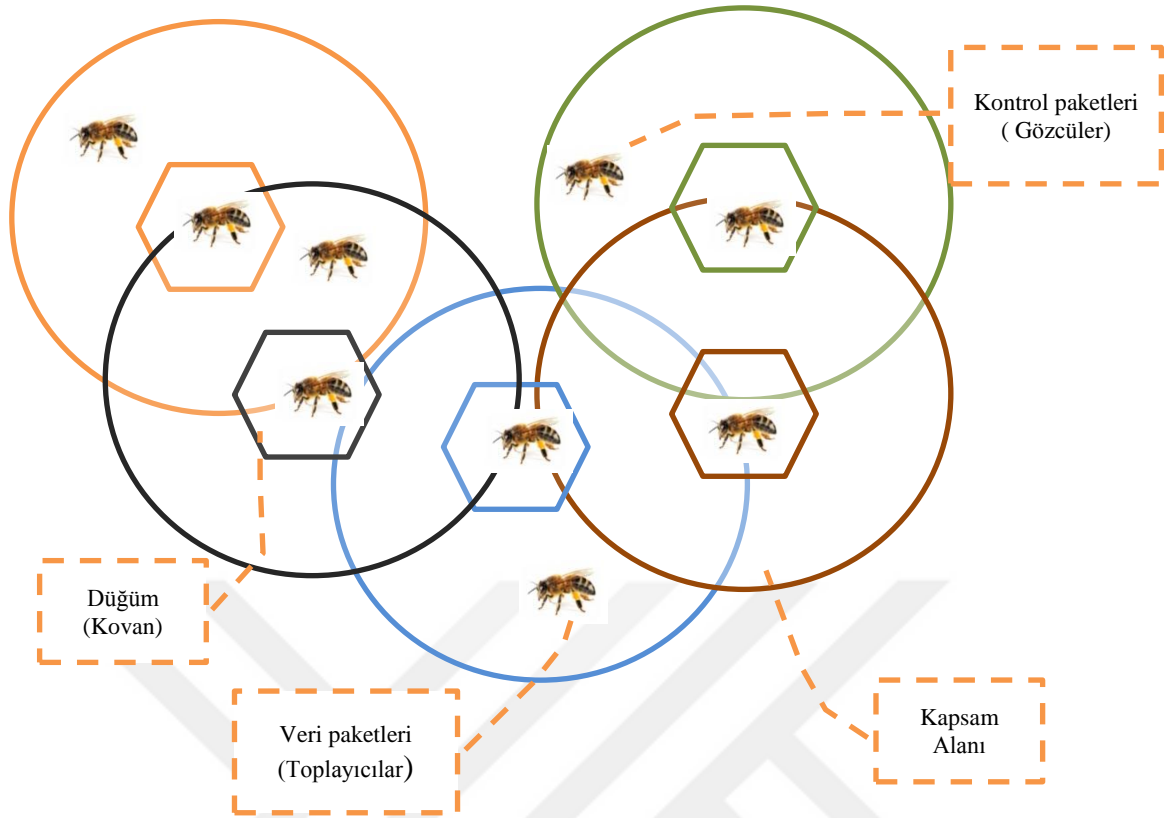
Yapılan çalışmada BeeWS modeli, bal arısı kolonilerinin yiyecek bulma tekniklerinden esinlenerek geliştirildi. Model üç farklı tipde arı ajanına sahiptir. Bunlar; veri oluşturucular, gözcüler ve veri taşıyıcılarıdır. Şekil 4.1' de arıların KAA'a uygulanış biçimi görülmektedir.

#### 4.2.1. Veri oluşturucu

Veri oluşturucu, gelen yol bilgisine göre üretilen paketleri veri taşıyıcılarına yükleme işlemini yapar. Veri oluşturucu oluşturulan veri paketlerini yönlendirme katmanından taşıma katmanına çıkarır. Veri paketlerini veri taşıyıcılarına teslim ettikten sonra görevlerini tamamlamış olurlar.

#### 4.2.2. Gözcüler

Bal arıları kolonilerinin varlıklarını sürdürebilmeleri için gözcülerin nektar araştırmaları çok önemli olduğu gibi KAA'larda da sistemin bekasının sağlanabilmesi için kontrol paketleri durumu da aynı prensiplere bağlı olarak o kadar önemlidir. Bunun yanında kontrol paketleri ağ yükünü dengeler.



Şekil 4.1. Bir kovannın yapısı

Yapay bal arılarının kendi kendine organize olmaları bireysel böcek davranışlarından türetilen basit kurallara sahiptir. Bu yapay bal arıları gözcü- işe alma olarak isimlendirilen mekanik özel bir sınıf davranışına karşılık gelir. Bu davranış önceden belirlenmiş kurallar çerçevesinde çevreden gelen bilgilere gösterilen tepkidir. BeeWS algoritması kuralları şunlardır:

- Kural 1: Olay tetiklemeli olarak, her kovan yiyecek kaynakları hakkında bilgi toplamak ve en iyi yolu seçmek için gözcü arıları gönderir.
- Kural 2: Toplama işlemi boyunca, gözcü arıların amacı mümkün olduğunca çok nektar toplamaktır.
- Kural 3: Gözcü arılar yiyecek kaynakları ve yollar hakkında bilgi toplarlar.

- Kural 4: KAA'da, belirli bir yol boyunca toplanan nektar miktarıyla yol maliyeti ters orantılıdır.
- Kural 5: Bir gözcü hedefe ulaştığında, çıktığı kaynağa geldiği yönden geri döner.
- Kural 6: Bir gözcü aynı yönde bir düğümü asla iki kez ziyaret etmez. Bunun için her düğümde ziyaret eden gözcülerin belli bir süre listesi tutulur.
- Kural 7: Bir gözcü maksimum hop sayısına ulaşırsa ölür.
- Kural 8: Tüm veya bazı gözcüler arı kovanına geri döndüğünde, yol bilgileri ve yol maliyetleri yönlendirme tablosuna kaydedilir.

Tablo 4.2. Gözcü oluşturma algoritması

---

**Algoritma 1 : Gözcülerin başlangıç algoritması**

---

```

gözcüolustur()
{
    oluştur yeni paketformat adı IIZC(ileri yönlü gözcü)
    {
        paket_yol ← kaynak_düğüm_kimliği
        paket_id ← veri_numarası
        paket_kaynak_adresi ← kaynak_düğüm_kimliği
        paket_enerji ← düğüm_enerji
    }
}

```

---

Gözcülerin görevi veri paketleri için yeni yollar bulmaktır. Veri oluşturucu bir veri oluşturduğunda bir atlama mesafede hedef olup olmadığı kontrol edilir. Eğer hedef bir atlama mesafede ise paket veri taşıyıcısına yüklenir. Eğer hedef bir atlama mesafesinde değilse gözcüler aktif hale getirilir, etraftaki düğümlere yayın yapılarak gözcüler yola çıkartılır. Gözcüler iki çeşittir: İleri yönlü gözcüler ve geri yönlü gözcüler. İleri yönlü gözcüler, yola çıktıklarında hedefe giden tüm yolları keşfetmeye çalışırlar. Bu işlem akış tekniği ile yapılır. Hedefe bir atlama mesafedeki düğümü bulan ileri yönlü gözcü geri yönlü gözcüye dönüşür ve keşfettiği yoldan geri döner.

Tablo 4.3. Gözcülerin çalışma algoritması

<b>Algoritma 2 : Gözcülerin çalışma algoritması</b>
<pre> gözcü_işlem() { <b>eğer</b> gelen ileri yönlü gözcü &amp;&amp; bu düğüme ait &amp;&amp; gözcü kimliği listesinde yok <b>ise</b> {     <b>ekle</b> enerji     <b>ekle</b> gözcü kimliği listesi     <b>eğer</b> hedefe bir atlama <b>ise</b>     {         <b>oluştur</b> yeni paketformat adı GIZC (geri yönlü gözcü)         {             paket_yol <math>\leftarrow</math> kaynak_düğüm_kimliği             paket_id <math>\leftarrow</math> veri_numarası             paket_üreten_adres <math>\leftarrow</math> düğüm_kimliği             paket_hedef_adres <math>\leftarrow</math> hedef_düğüm_kimliği             paket_enerji <math>\leftarrow</math> toplam_enerji             paket_maliyet <math>\leftarrow</math> toplam_maliyet         }         <b>eğer</b> maliyet 15 den büyük <b>ise</b>             gözcü iptal edildi         <b>değilse</b>         {             maliyet++             IIZC gönder         }     } } <b>eğer</b> gelen geri yönlü gözcü &amp;&amp; bu düğüme ait <b>ise</b> {     GIZC gönder     hedef düğüm güncelle } } </pre>

Tablo 4.4. Yönlendirme tablosu güncelleme algoritması

<b>Algoritma 3 : Yönlendirme tablosu güncelleme algoritması</b>
<pre> yönlendirmetablosu_güncelleme() { <b>eğer</b> gelen ileri yönlü gözcü &amp;&amp; bu düğüme ait &amp;&amp; gözcü kimliği listesinde yok <b>ise</b> {     <b>eğer</b> hedefe bir atlama <b>ise</b>         tablo_güncelle <math>\leftarrow</math> yol <math>\leftarrow</math> sonraki_düğüm <math>\leftarrow</math> enerji <math>\leftarrow</math> atlama_sayısı+1     <b>değilse</b>         tablo_güncelle <math>\leftarrow</math> yol <math>\leftarrow</math> sonraki_düğüm <math>\leftarrow</math> atlama_sayısı+1     } } <b>eğer</b> gelen geri yönlü gözcü &amp;&amp; bu düğüme ait <b>ise</b> {     <b>eğer</b> kaynak düğüm <b>ise</b>     {         <b>ekle</b> kuyruk <math>\leftarrow</math> yol_kodu <math>\leftarrow</math> toplam_enerji/atlama_sayısı         tablo_güncelle <math>\leftarrow</math> yol <math>\leftarrow</math> önceki_düğüm <math>\leftarrow</math> toplam_enerji <math>\leftarrow</math> atlama_sayısı     }     <b>değilse</b>         tablo_güncelle <math>\leftarrow</math> yol <math>\leftarrow</math> önceki_düğüm <math>\leftarrow</math> toplam_enerji <math>\leftarrow</math> atlama_sayısı     } } } </pre>

Tablo 4.2’de hedefe giden yolu bulma ihtiyacı oluştuğunda kaynak tarafından oluşturulan gözcülerin oluşumu ile ilgili algoritma verilmiştir. Tablo 4.3’de gözcülerin hedefe ulaşmada kullanılacak yolu bulurken üzerinden geçtiği düğümlerde oluşturduğu davranışlar verilmektedir. Tablo 4.4’de gözcüler tarafından oluşturulan yönlendirme tablolarının oluşum algoritması verilmektedir.

### 4.2.3. Veri taşıyıcı

Geri yönlü gözcüler yol ve bu yola ait enerji bilgilerini getirir. İlk gelen geri yönlü gözcüden itibaren zamanlayıcı çalıştırılır ve belli bir süre beklenir. Gelen bu yol ve enerji bilgisi hesaplanarak herbiri numaralandırılır. Bu yollarda en uygun olanı seçilir ve paket veri taşıyıcısına yüklenerek bu yoldan gönderilir. Veri taşıyıcısı en uygun yolu seçmekle yükümlüdür. Bu sebepten ağ ömrü için önemi büyüktür.

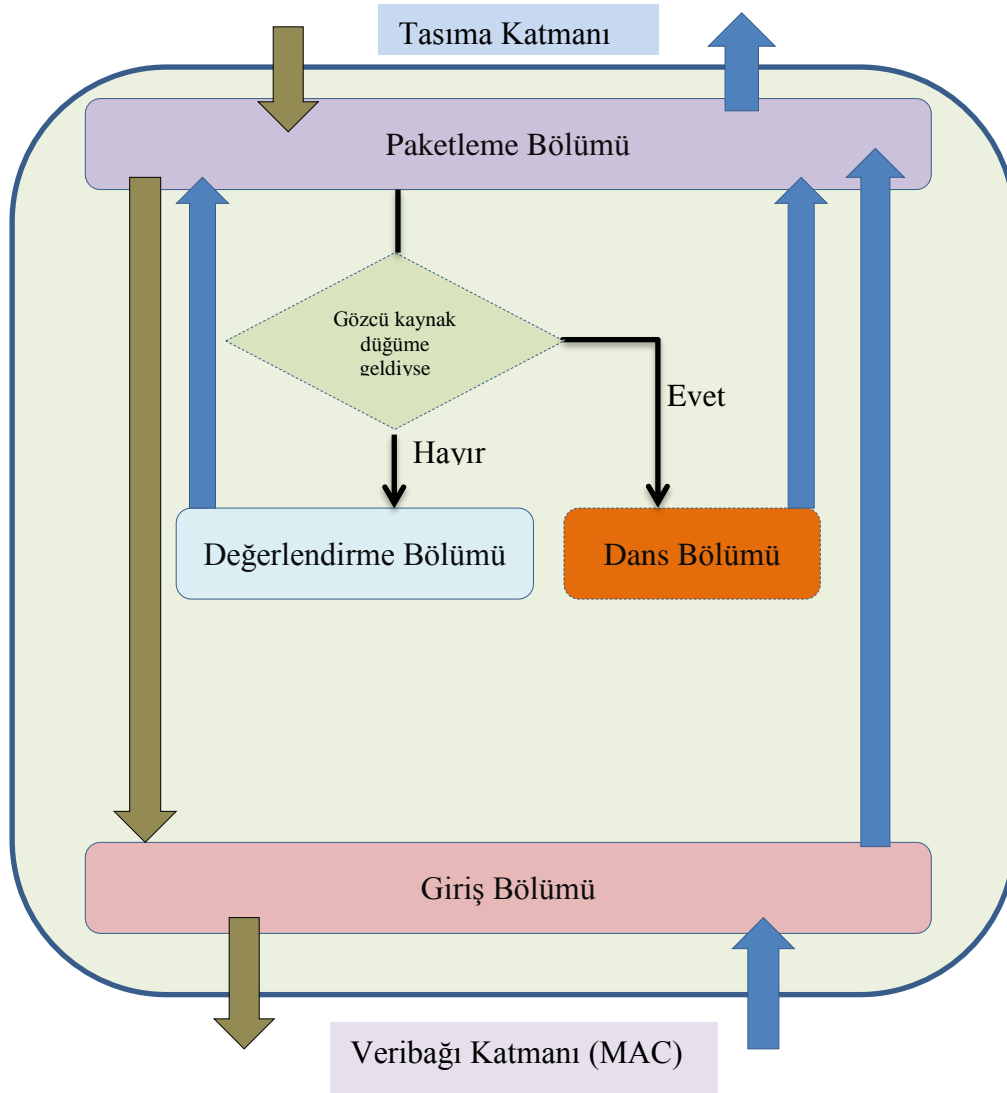
## 4.3. BeeWS Mimarisi

BeeWS, yazılım olarak kendi yönlendirme katmanına sahiptir. Her düğüm de bulunan bu yazılım bir bal arısı kovani olarak düşünülebilir. Şekil 4.2’de gösterildiği gibi bu sistem dört aşamadan oluşur: kovan, gelen arılar bölümü, değerlendirme bölümü, dans bölümü ve kovandan çıkış bölümü. Veribağı katmanından giriş bölümüne gelen paket, türüne göre ayrıştırılıp gerekli eklemeler yapılarak paketleme bölümüne gönderilir. Paketleme bölümüne gelen paket gözcü ise değerlendirme veya dans bölümüne gönderilir. Değilse taşıma katmanına gönderilir. Eğer paketin ulaştığı düğüm kaynağa dans bölümüne gönderilerek burada paketdeki bilgiler alınarak bekleme durumuna geçilir. Değilse düğüm için gerekli yol bilgileri alınır ve paketleme katmanına gönderilir.

### 4.3.1. Giriş bölümü

Bu bölüm MAC katmanından gelen paketleri karşılar. Şekil 4.3’de gösterildiği gibi paketin tipine bağlı olarak gelen paketler veri oluşturucu, gözcü ve veri taşıyıcısı şeklinde ayrıştırılır. Ayrıştırılan paketler tiplerine göre uygun bölümlere

değerlendirmek üzere gönderilir. Gelen paket kaynağa ulaştığında bekleme durumuna geçer. Kaynak değilse gözcü paketleme bölümüne gönderilir. Paket veri taşıyıcı ise yol varsa paketleme bölümüne gönderilir. Hedefe bir atlama varsa paketleme bölümüne gönderilir. Hiçbiri yoksa yeni yol oluşturmak için paketleme bölümüne gönderilir.



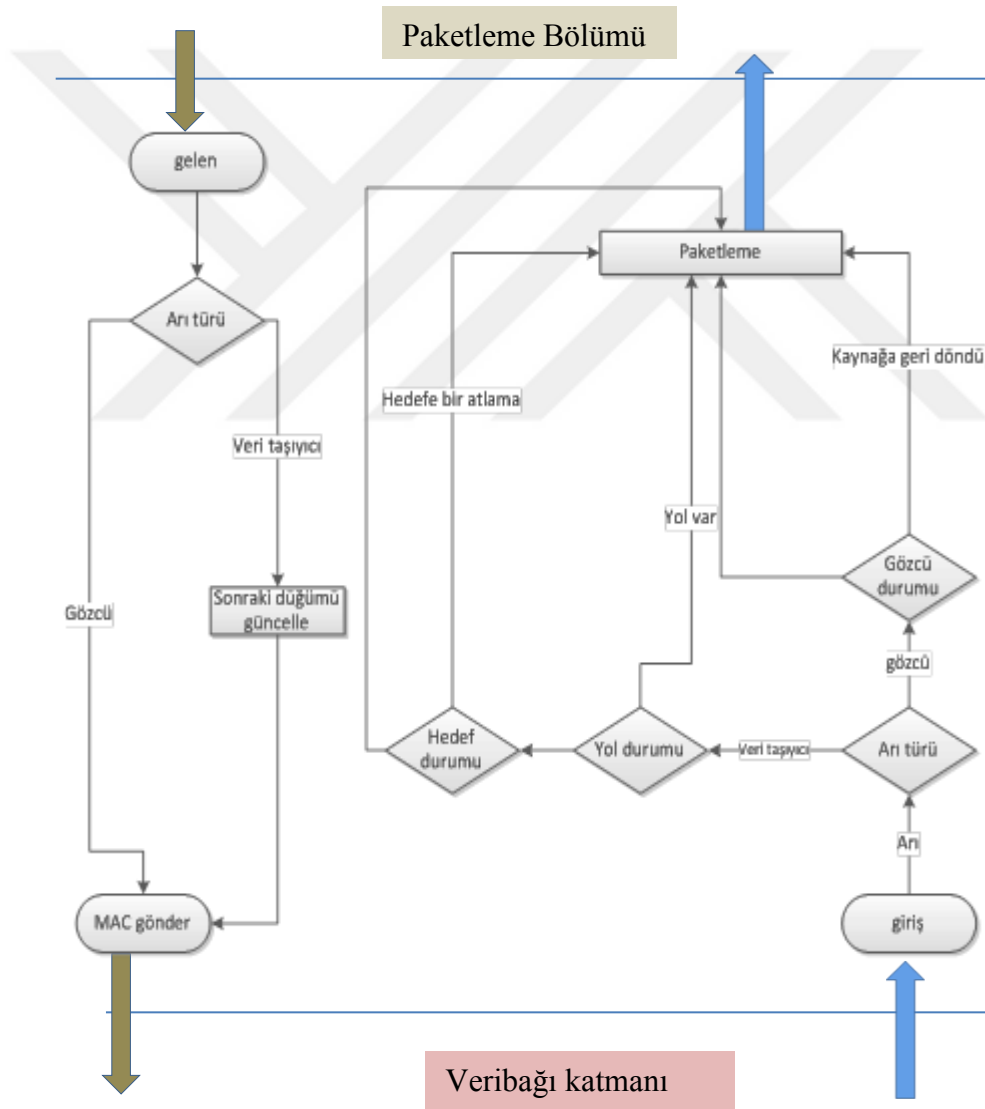
Şekil 4.2. BeeWS mimarisi

#### 4.3.2. Değerlendirme bölümü

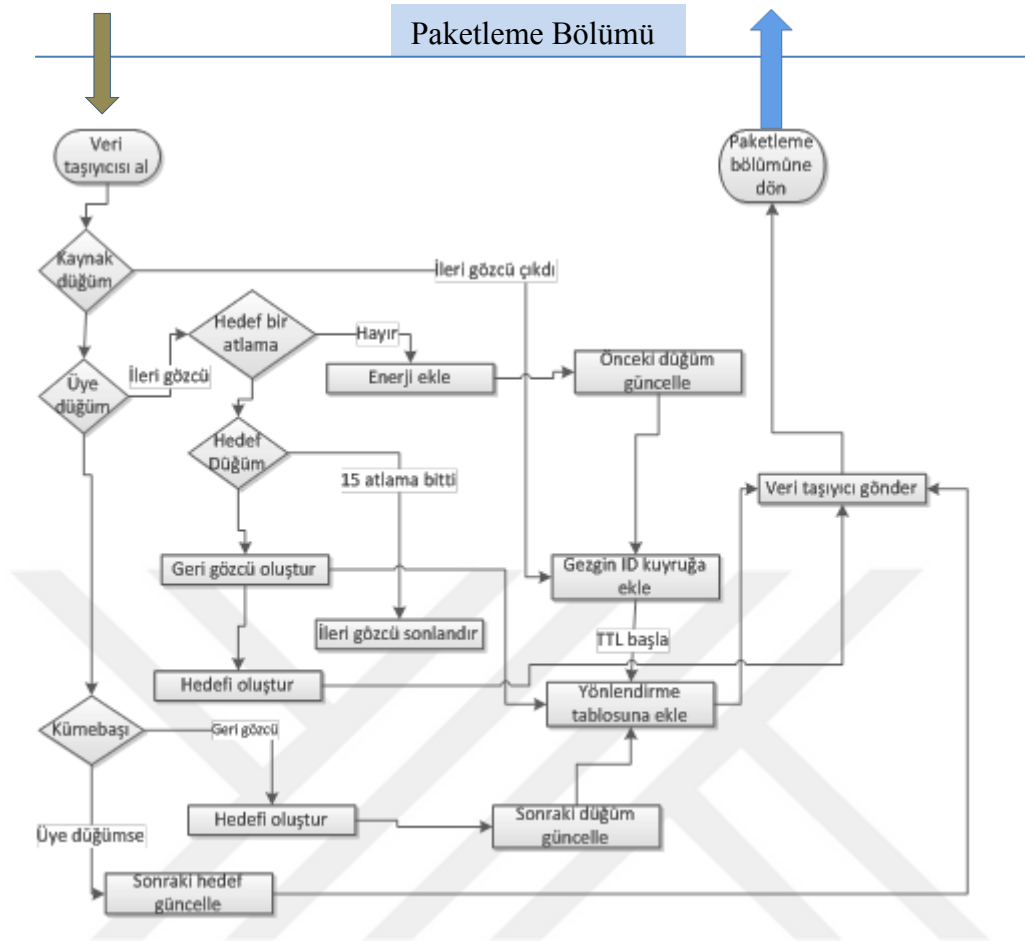
Gelen tüm bilgiler burada değerlendirilip yönlendirme işlemi yapılır. Şekil 4.4'de gösterildiği gibi. Alınan paket veri paketi ise ve düğümün kapsam alanında hedef



varsa paket veri taşıyıcısına yüklenir. Yoksa daha önce bulunan yollar bir süre saklandığı için bu düğümden hedefe giden yollar kontrol edilir. Yol varsa o yol kullanılır. Yol yoksa yeni yol bulunması için ileri yönlü gözcüler harekete geçirilir. Paket ileri yönlü gözcü paketi ise paketin geldiği düğüm kaydedilir. Düğüm hedefden önceki son düğümse geri yönlü gözcü oluşturulur ve paketin geldiği düğüm kaydedilerek aynı düğüme geri gönderilir. Eğer gelen paket geri yönlü gözcü ise paketin geldiği düğüm kaydedilir. Düğümde kayıtlı olan önceki düğüme bakılarak geri yönlü gözcü o düğüme gönderilir.



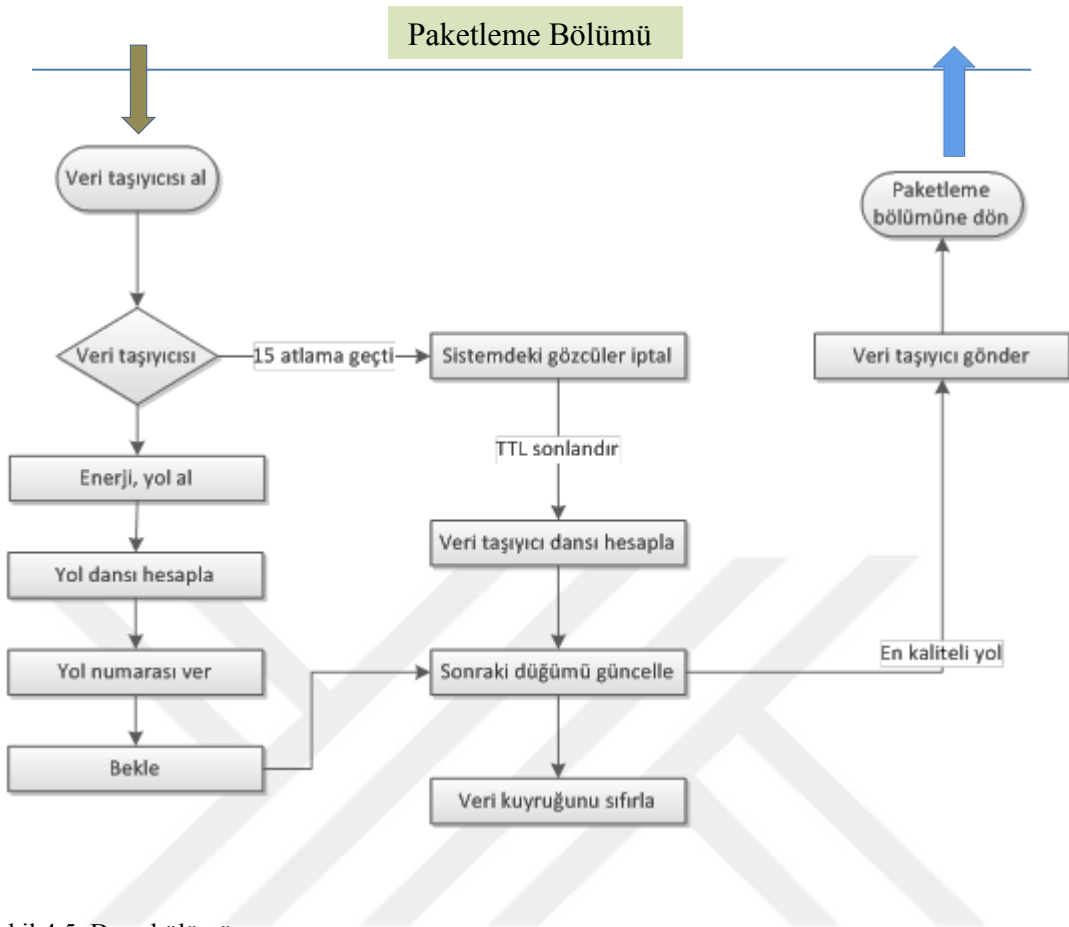
Şekil 4.3. Giriş bölümü



Şekil 4.4. Değerlendirme bölümü

### 4.3.3. Dans bölümü

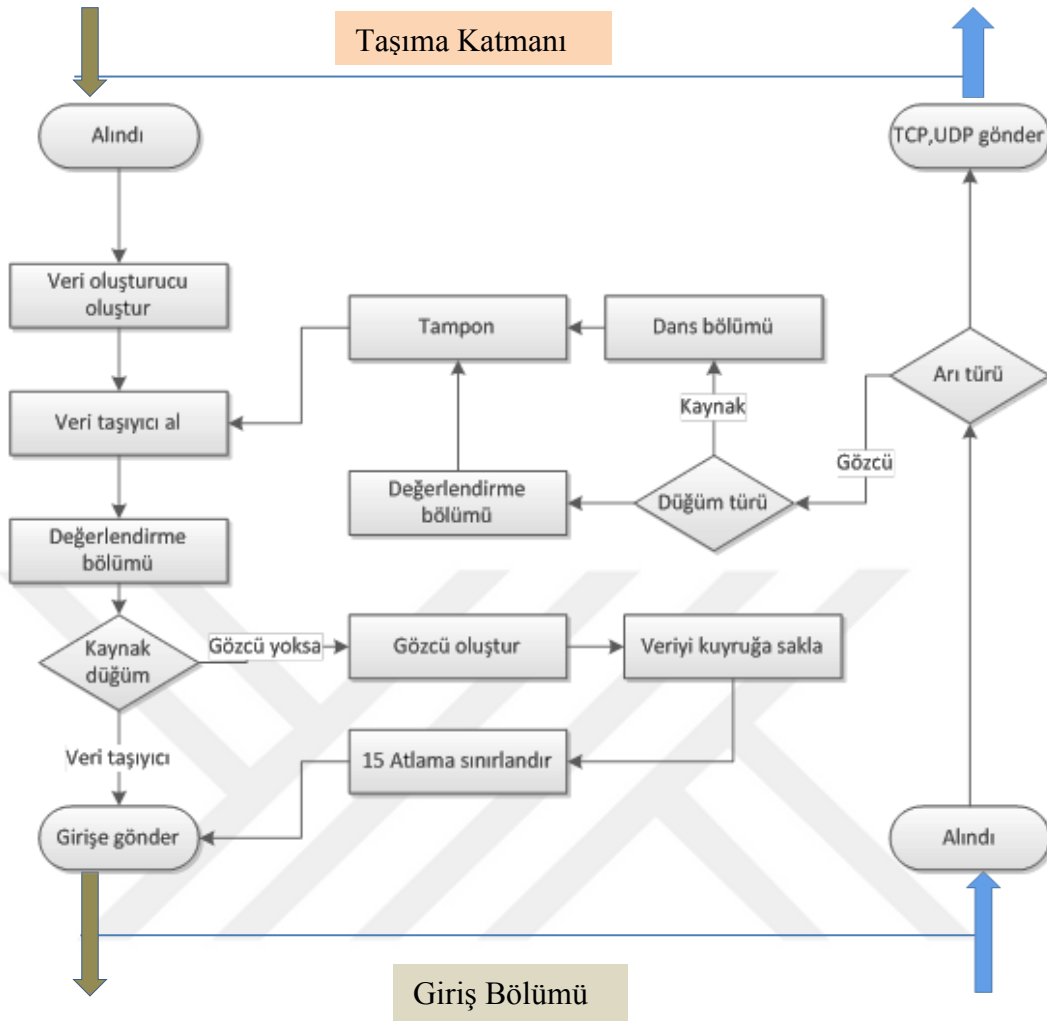
Bu olay kaynak düğümde gerçekleşir. Gelen geri yönlü gözcü paketi kaynak düğüme ulaştıysa geldiği yoldaki atlama sayısını ve yoldaki toplam enerjiyi kaynak düğüme aktarır. Alınan bu bilgi değerlendirilerek bir yol numarası verilir. Gelen ilk geri yönlü gözcü paketinden sonra zamanlayıcı kurulur ve ortalama 15 atlamının gerçekleşeceği bir süre beklenir. Bu sürede gelen geri yönlü gözcü paketlerinin bilgileri değerlendirilerek numaralandırılır. Süre sonunda zamanlayıcı sıfırlanır. Kaynağa ait ağda bulunan ileri ve geri yönlü gözcü paketleri sonlandırılır. Bu yollara ait değerlerden en küçük değere sahip olan yol tercih edilir. Paket veri taşıyıcılarına teslim edilir.



Şekil 4.5. Dans bölümü

#### 4.3.4. Paketleme bölümü

Paket veri taşıyıcılarına yüklendi ise paket taşıma katmanına yönlendirilir. Paket veri taşıyıcılarına yüklenmedi ise tipine göre ileri yönlü gözcü veya geri yönlü gözcü olarak taşıma katmanına gönderilir (Şekil 4.6). Bu bölüm taşıma katmanı ile olan ilişkiyi kurar. Paketleme bölümüne gelen paket veri taşıyıcısı ise taşıma katmanına gönderilir. Gözcüyse düğüm türüne bakılır. Kaynak düğümse dans bölümüne; değilse değerlendirme bölümüne gönderilir.



Şekil 4.6. Paketleme bölümü

### 4.3.5 Yönlendirme tablosu

BeeWS bir yönlendirme protokolüdür ve yönlendirme tablosu kullanır. Yönlendirme tablosu kısa süreli olarak yolları saklar. BeeWS yönlendirme tabloları aşağıdaki bilgileri tutar.

- Yol numarası
- Kaynak kimliği
- Gözcü kimliği
- Önceki adım

- Sonraki adım
- Adım sayısı (hedefe ulaşmak için gerekli hop sayısı)
- Enerji
- Yaşam süresi (yolun son veya silme süresi)

Düğümüne geri yönlü gözcüler tarafından getirilen tüm yollar yol numarası ile numaralandırılır. Yolun hangi kaynağa ait olduğunun bilinmesi amacıyla gözcüler yönlendirme tablolarına kaynak kimliği bırakır. Ağdaki gözcülerin düğümler tarafından birbirinden ayırt edilebilmesi için her bir gözcüye ayrı gözcü kimliği verilir. Düğümün kendinden önceki ve sonraki düğümleri bilmesi için tabloya önceki ve sonraki düğümler kaydedilir. Düğümden hedefe kaç adımda ulaşıldığını tespit etmek amacıyla adım sayısı bilgisi tabloya eklenir. Düğümün enerji bilgisi de tabloya eklenir. Tabloda var olan yol bilgisi belli bir süre saklandığı için bu süreyi tutan yaşam süresi alanı mevcuttur.

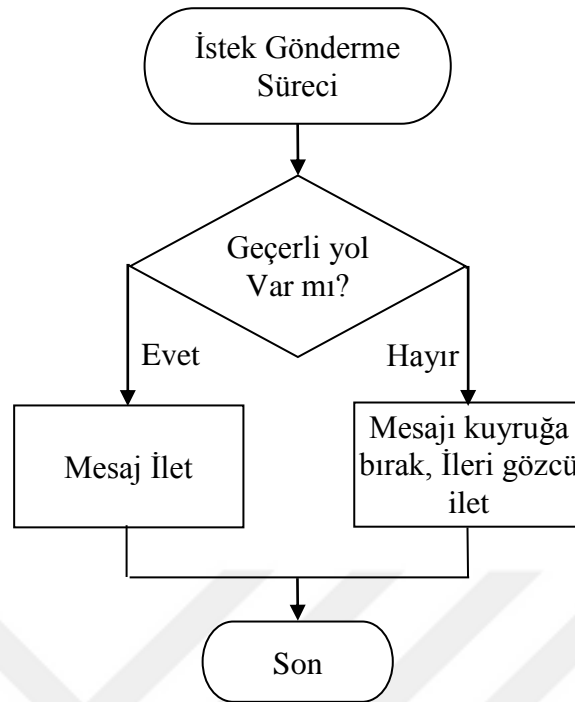
#### **4.3.6. BeeWS paket yapıları**

##### **4.3.6.1. İleri yönlü gözcü mesajı**

Bir düğüm bir hedef için yol bilgisine ihtiyaç duyduğu zaman ileri yönlü gözcü mesajı yayınlar (broadcast). Hedefe ait bilgiye sahip düğüm tarafından gönderilen geri yönlü gözcü mesajının (unicast) alınmasıyla hedefe ait bir yol elde edilir. Bu iletişim sırasında, ara düğümler aldıkları mesajlardan kendi yönlendirme tablolarında güncelleme yaparlar. Bir veri gönderme isteği olduğunda düğümün sergilediği davranış Şekil 4.7'deki akış şemasında gösterilmektedir.

Düğüm geçerli bir yola sahip olmadığı zaman yol kurmak üzere Şekil 4.8'de gösterilen biçimde bir ileri yönlü gözcü mesajı yayınlar.

Şekil 4.9'da kaynak düğüm (1) hedef düğüme (0) bir veri göndermek istemektedir. Yönlendirme tablosunda 0 düğümü için bir yol bilgisi mevcut değildir. Bu nedenle ileri yönlü gözcü mesajı yayınlar.

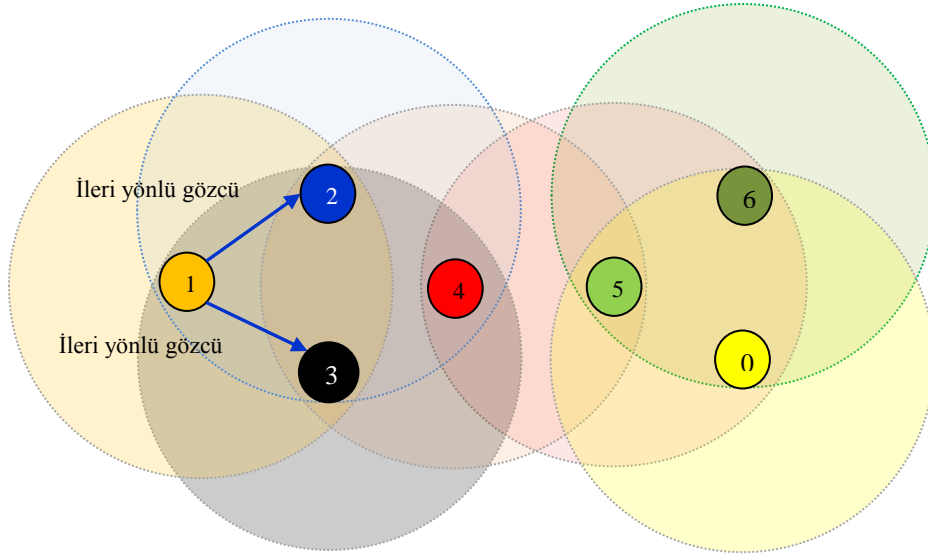


Şekil 4.7. BeeWS mesaj gönderme süreci

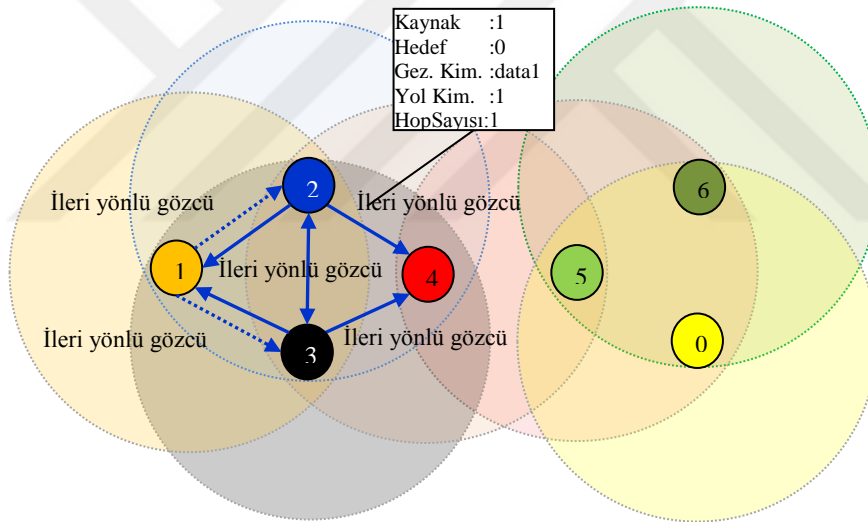


Şekil 4.8. İleri yönlü gözcü mesaj paketi içeriği

Mesajı alan 2 ve 3 düğümleri öncelikle 1 düğümüne ait yol bilgileriyle ilgili güncelleme yaparlar. Bir düğüm ileri yönlü gözcü aldığı zaman iki seçeneği vardır: (1) Hedefe ait geçerli bir yol bilgisine sahip ise geri yönlü gözcü yayınlar, (2) değil ise ileri yönlü gözcü mesajını iletir. Burada 2 ve 3 düğümleri 0 düğümüne ait bir geçerli yol bilgisine sahip olmadıkları için ileri yönlü gözcü mesajını iletirler. (Şekil 4.9, Şekil 4.10 ve Şekil 4.11) İleri yönlü gözcü mesajında bulunan gözcü kimliği sayesinde düğümler kendilerine ulaşan mesajlarından sadece biri için işlem yaparlar. Böylece sonsuz döngüler engellenmiş olur. (Tablo 4.2 ve Tablo 4.3)



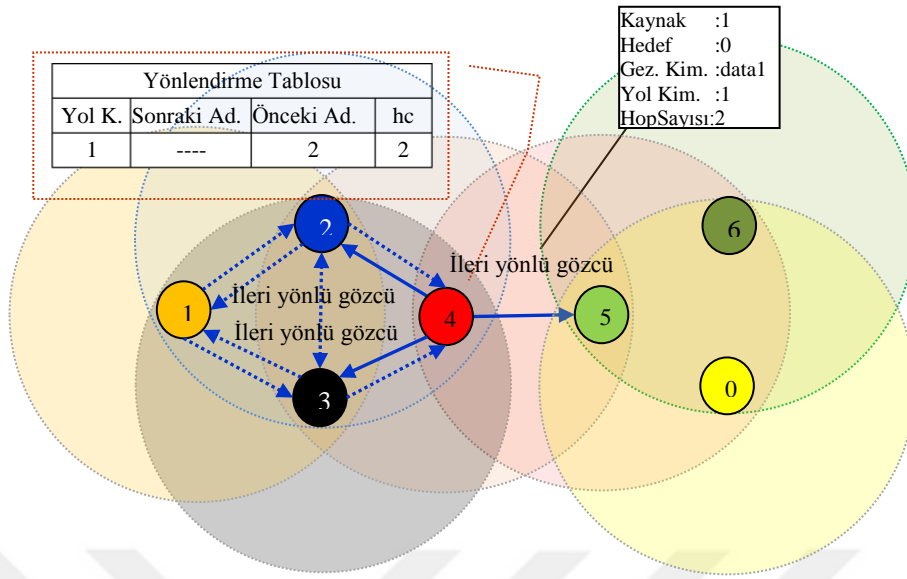
Şekil 4.9. İleri yönlü gözcü mesaj iletimi



Şekil 4.10. İleri yönlü gözcü mesaj iletimi

#### 4.3.6.2. Geri yönlü gözcü mesajı

Bir düğüm kendisine gelen ileri yönlü gözcü mesajının hedefine ait geçerli bir yola sahip ise, düğümde geri yönlü gözcü mesajı oluşturulur ve Şekil 4.12'de içeriği verilen gibi bir geri yönlü gözcü mesajı yayınlar (Şekil 4.13 ve Şekil 4.14).



Şekil 4.11. İleri yönlü gözcü yönlendirme tablo oluşumu

Yol kimliği
Gözcü kimliği
Üreten adres
Hedef adres
Toplam enerji
Toplam maliyet
Yaşam süresi

Şekil 4.12. Geri yönlü gözcü mesajı bileşenleri

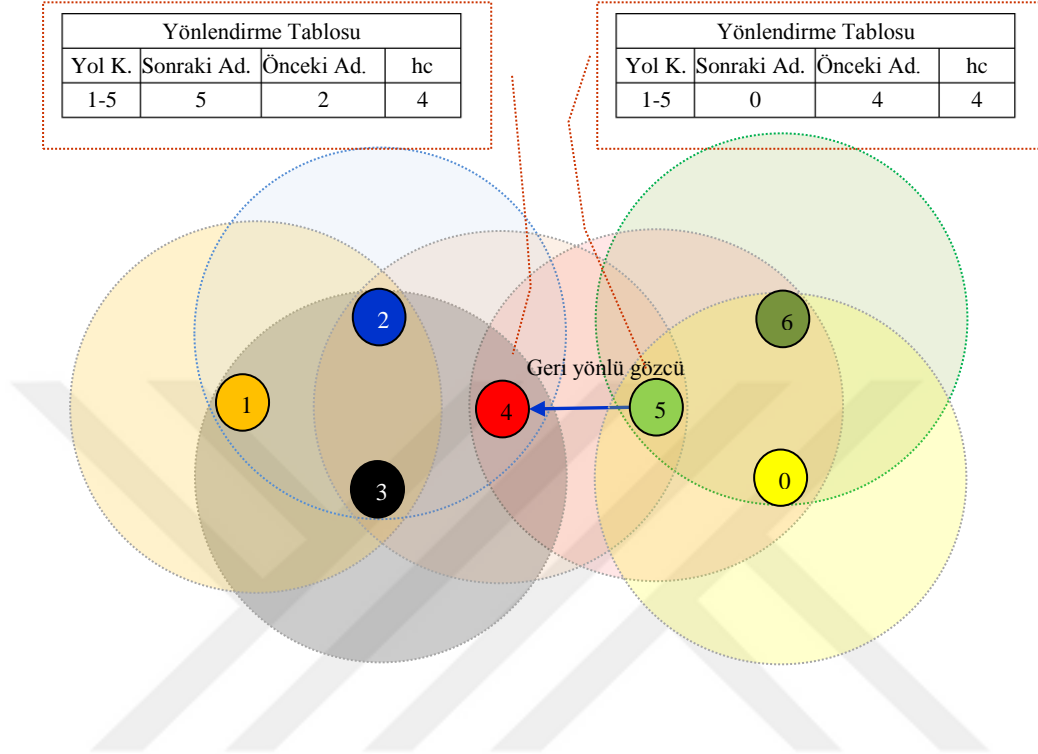
Düğüm geri yönlü gözcü üretirken, hedef adresine, ileri yönlü gözcü mesajını gönderen düğümün kimliğini, geri yönlü gözcü mesajında karşılık gelen hedef adres alanları koyar. Yol da gelirken hesaplanan toplam enerji ve toplam maliyet bilgilerini de gereken alanlara yerleştirir. Kendi adresini yol kimliğini ekleyerek ve gözcü kimliğini de yerleştirerek geri yönlü gözcü paketini oluşturur.

#### 4.3.7. Kümeleme yöntemi

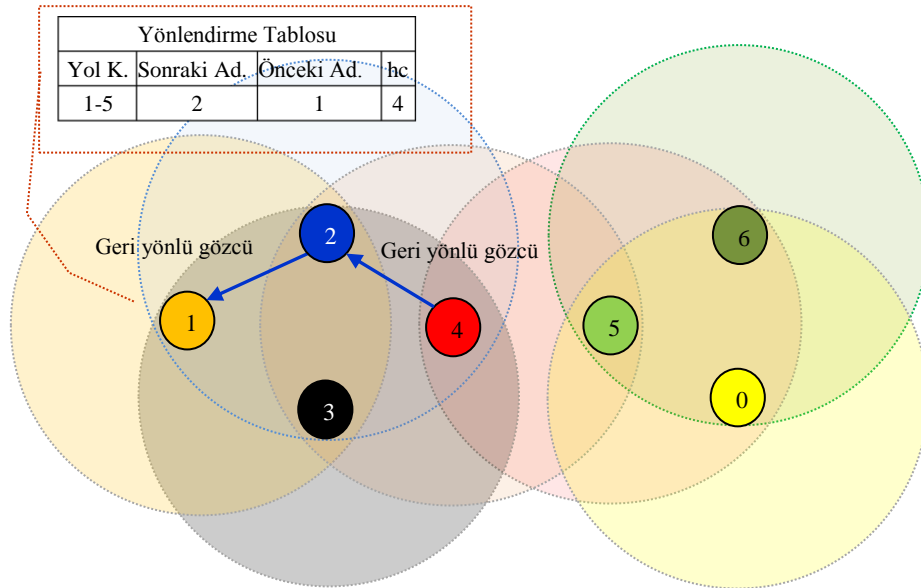
Arıların kovandaki birey sayısı artınca oğul verme yöntemiyle bölünerek yeni kovanlar oluşturulur. Bu sebepten sayılarının çoğalması sistemin çalışmasını



etkilemez. Arıların bu davranışından ilham alınarak yapılan çalışmada, yüksek ölçeklere çıkabilmek için geliştirilen kümeleme algoritmasıyla büyük ölçekli KAA'ları oluşturuldu.



Şekil 4.13. Geri yönlü gözcü mesaj iletimi yayılışı



Şekil 4.14. Geri yönlü gözcü yönlendirme tablosu

Yöntem geliştirilirken, Bölüm 2’de anlatılan kümeleme yöntemlerinden faydalanıldı.

#### 4.3.7.1. Kümebaşı seçimi

Kümeleme işlemi, başlamadan önce tüm düğümler kapsama alanındaki düğümlerden haberdar olabilmek için HELLO paketi yayınlarlar. Hello paketi komşu düğüm kimliğini, enerjisini ve yaşam süresini içerir (Şekil 4.15).

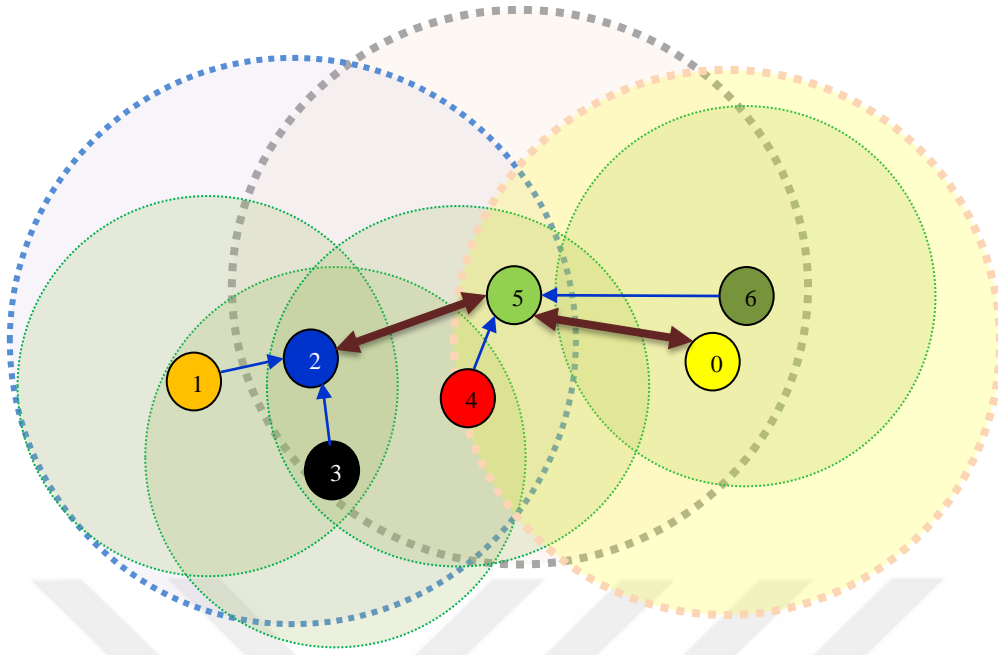
Düğüm kimliği
Enerji
Yaşam süresi

Şekil 4.15. HELLO paketi içeriği

Hello paketleri dolaşırken düğümlerin hepsi kendisini kümebaşı kabul eder. Bu yüzden kapsama alanları kümebaşlarının sahip olduğu kapsama alanı kadar olur. Bu durumda her düğüm kapsama alanındaki düğümleri tespit eder. Komşu sayısı 15 ve üzerinde olanlar kendilerini kümebaşı ilan ederler ve kapsama alanındaki düğümler kümebaşlarına bağlanır. Eğer bir üye düğüme, iki veya daha fazla kümebaşı tarafından ilan ulaşılmış ise bağlantı gücü fazla olana bağlanır. Kümebaşlarının kapsama alanı üye düğümlerin kapsama alanının üç katıdır. Bu işlem ikinci ve daha sonraki süresi belirlenen dönüşümlerde komşuların enerjileri de kümebaşı seçimine katılarak yenilenir. Kümede bulunan komşu sayısı ve enerjisi en yüksek olan düğüme kümebaşı görevi devredilir.

#### 4.3.7.1. Kümeleme yöntemi çalışması

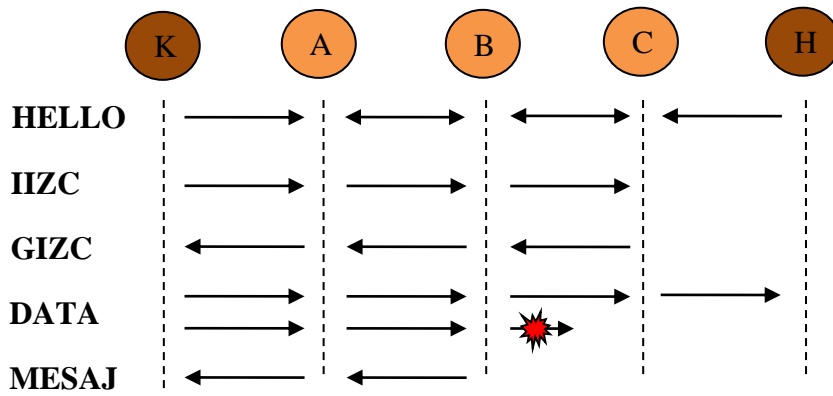
Kümebaşı seçimi sonucunda oluşturulan kümelerdeki üye düğümler sadece algılama işlemi yapıp veri paketi oluşturular ve bu paketleri kümebaşlarına gönderirler. Kümebaşları kendi aralarında oluşturdukları KAA vasıtasıyla çoklu-atlama yöntemini kullanarak hedefe ulaşırlar. Kümebaşları kendi aralarındaki veri aktarımı ve yol bulma işlemlerinde BeeWS mimarisini kullanırlar.



Şekil 4.16. Kümeleme yöntemi

Şekil 4.16'da gösterildiği gibi 2 ve 5 numaralı düğümler kümebaşı seçildi. Diğerleri üye düğüm olarak atandı. 0 numaralı düğümde hedef olarak kabul edildi. 2 numaralı kümebaşı hedefe ulaşmak için 5 numaralı kümebaşını kullanmak zorundadır. Bu şekilde oluşan kümeler çoklu-atlama yöntemi ile verileri hedefe ulaştırırlar.

Şekil 4.17'de kümeleme yönteminde paketlerin çalışma şeması gösterilmiştir. Protokolün benzetim aracındaki çalışması ekler bölümünde yer almaktadır.



Şekil 4.17. BeeWS protokol mesaj

## **BÖLÜM 5. DEVS-SENSOR MODELLEME VE BENZETİM ORTAMININ GELİŞTİRİLMESİ**

### **5.1. Giriş**

Bu bölümde, KAA'lardaki yönlendirme problemlerine çözüm getirmek amacıyla tasarlanan oğul zekası tabanlı yönlendirme algoritmasını uygulamak amacıyla DEVS-Suite üzerinde DEVS-Sensor benzeticisinin tasarım aşamaları tanımlanacaktır. Geliştirilen benzetici, DEVS [6,73] modelleme ve benzetim yaklaşımı kullanmaktadır. Kolay ara yüze sahip olması, esnek bir şekilde programlanabilmesi, büyük ölçeklere çıkabilmesi ve otomatik izleme araçlarına sahip olması gibi özellikleri vardır.

Tipik bir KAA algılayıcı düğümleri, kapsama alanları ve iletişim paketlerinden oluşur. Bu bileşenler DEVS yaklaşımının bir uyarlaması olan DEVS-Suite benzetim ortamında modellenmiştir. Geliştirilen bu yeni modelleme ve benzetim ortamı algılayıcı ağ teknolojileriyle çalıştığı için DEVS-Sensor adı verilmiştir. Geliştirilen DEVS-Sensor ortamı tasarlanan oğul zeka tabanlı yönlendirme algoritmasını test etme ve değerlendirme imkanı sağlar.

DEVS-Sensor ortamını oluşturan bileşenler, nesnelere, kavramlar ile ilgili teorik bilgiler ve DEVS-Sensor ortamından elde edilen ekran çıktıları bölüm içerisinde detaylandırılmaktadır.

### **5.2. Java Programlama Dili ve DEVS-Suite ortamı**

Java programlama dili C ve C++'programlama dilleri ile benzer söz dizimine sahip nesneye dayalı bir programlama dilidir [74]. Java dilinin taşınabilirliği ve web sayfalarında kullanılabilen Java Appletleri nedeniyle sık sık 'World Wide Web' ile

birlikte anılmaktadır [75]. Java programlama dili C++ programlama dilinden daha yüksek bir başarıma sahiptir [74].

Java programlama dili, Java Sanal Makinesi (Java Virtual Machine –JVM) ve Java programlama dili ile ilişkili sınıf kütüphaneleri yardımıyla 3 farklı alanda taşınabilirlik özelliği sunar [76]. Bunlar;

1. Kaynak kodu taşınabilirliği: Bir Java programı, temel CPU, işletim sistemi veya Java derleyicisi dikkate alınmaksızın aynı sonuçları üretmelidir.
2. CPU mimarisi taşınabilirliği: JVM CPU mimarisi taşınabilirliğini sağlar. Java derleyicisi, JVM için nesne kodunu (J-code) üretir ve belirli bir CPU üzerinde JVM ilgili CPU için J-kodunu derler.
3. OS/GUI taşınabilirliği: OS/GUI taşınabilirliğini tesis etmek için, Java bazı paketleri (awt, util ve lang) sağlar.

Java tabanlı ayırık olaylı sistem modelleme araçlarının kullanımı gittikçe artmaktadır [77-79]. Java basitlik, %100 nesneye yönelim, hız, dağıtık çalışabilme, taşınabilirlik, vb. pek çok faydadan dolayı modelleyiciler tarafından tercih edilmektedir [81].

DEVS-Suite modelleme ve benzetim ortamı paralel DEVS formalizmine dayanır ve modüler, hiyerarşik, ayırık olaylı sistem ortamı, nesneye dayalı Java programlama dili kullanılarak gerçekleştirilmiştir [78,80]. DEVS-Sensor benzetim ortamının modellenmesi ve benzetimi DEVS-Suite ortamında Java'nın avantajlarından faydalanılarak hazırlanmıştır.

### **5.3. DEVS-Sensor Modelleme Süreci**

Dağıtık ağ sistemlerinin tasarım ve analizini sistematik bir biçimde ortaya koymak için sistemlerin modellenmesi ve benzetimi beş aşamada gerçekleştirilebilir (Şekil 5.1).

Birinci aşama, modelleme ve benzetim hedeflerinin belirlenmesidir (Şekil 5.1). Modelleme hedefleri, modelin tasarımında, yönetiminde ve kontrolünde tanımlanan rolleriyle ilişkili olmalıdır. Önceden hedeflerin bilinmesi, model tasarlama işleminin daha belirgin problemler üzerine odaklanmasına neden olur.

İkinci aşama, benzetim bileşenlerinin tanımlanması işlemidir. Bu aşamada benzetimin içerdiği alanlar ve bu alanlara ait özellikler tanımlanır. Düğümlerin ve varlıkların ayrı ayrı ele alındığına dikkat edilmelidir.

Üçüncü aşama, geliştirilen varlıkların ve düğümlerin birleştirilmesi aşamasıdır. Bu bileşenler DEVS-Suite modelleme ve benzetim ortamında birbiriyle bağlanarak değişik topolojiler ve ağ modelleri meydana getirilebilir.

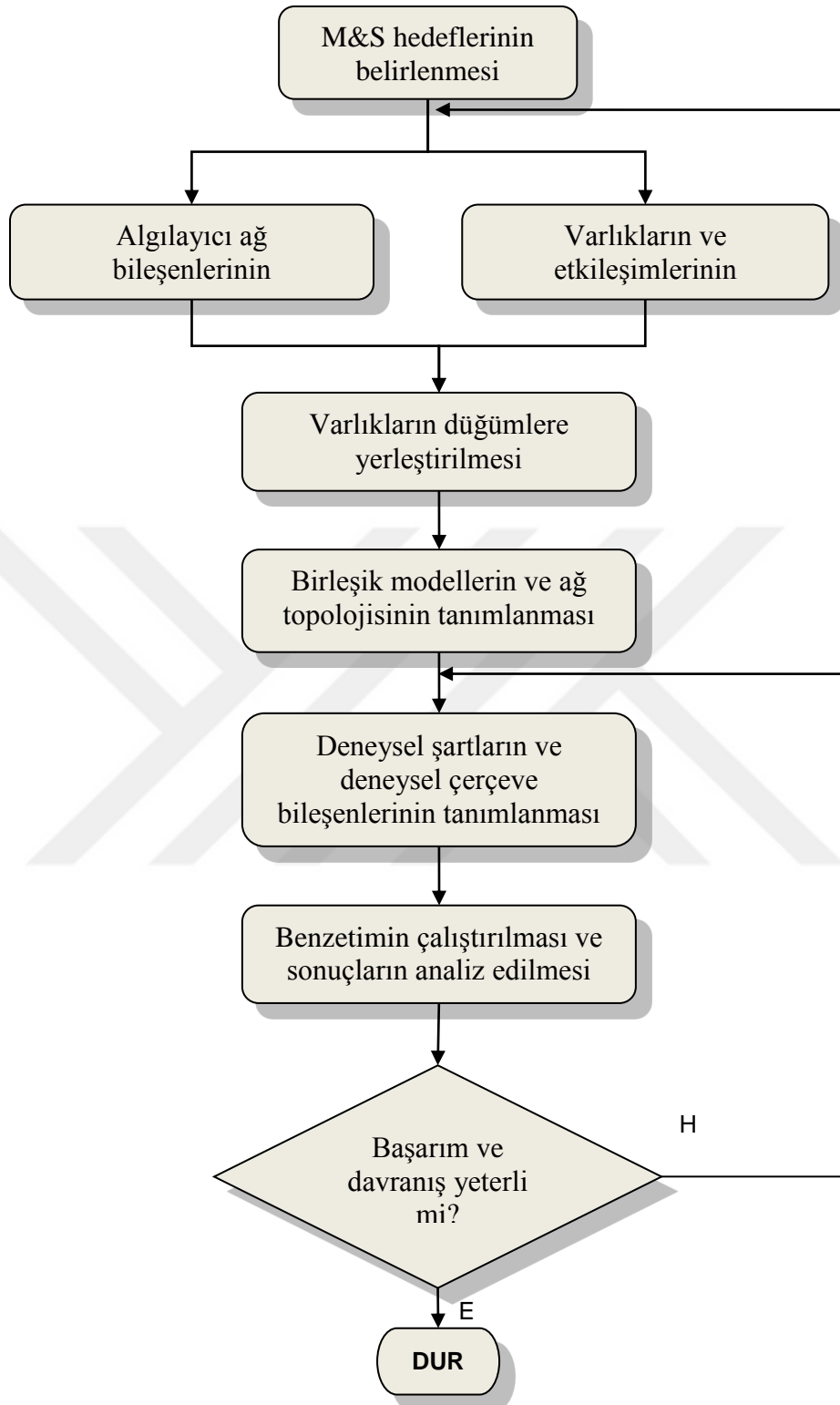
Dördüncü aşama, modelleme ve benzetim hedeflerine ve geliştirilen modellere uygun bir deneysel çerçeve tanımlanması işlemidir. Bu deneysel çerçeve modelin belirlenen şartlar altında test edilmesine ve yapılan çalışmanın değerlendirilmesine olanak tanır.

Son aşamada, benzetim çalıştırılır ve sonuçlar DEVS-Suite ortamında gözlemlenir. Sonuçlar kabul edilebilir bir aralıkta ise benzetim işlemi sonlandırılır. Aksi halde, tekrar başa dönülür ve geliştirilen modellerin parametreleri ayarlanır. Böyle bir süreçte ileri geri hareket edilerek en uygun model geliştirilmiş olur.

#### **5.4. KAA Modelleme Yaklaşımı ve Bileşenlerin Tasarımı**

Bir KAA topolojisi, kısıtlı enerjiye sahip algılayıcı düğümlerinden oluşur. KAA'lardaki algılayıcı düğümler, bulunduğu ortamdan topladığı bilgileri merkezi bir erişim noktasına göndermektedir [81]. Şekil 5.2'de görüldüğü gibi KAA modelinin bileşenleri geliştirilen DEV-Sensor yaklaşımına uyarlanmış hali görülmektedir.

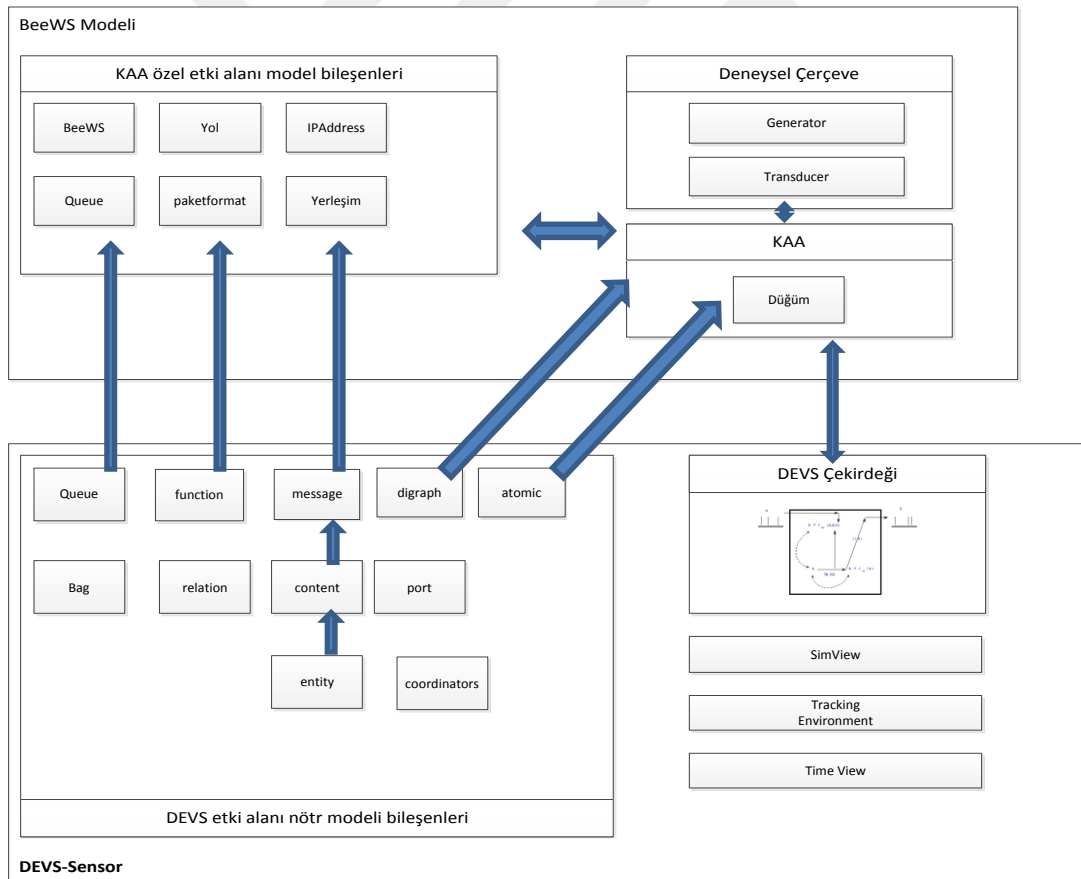
KAA topolojilerini modelleme amacına yönelik oluşturulan düğümler, düğümlerin kapsama alanları ve bu düğümlerin haberleşmesini sağlayan paketler



Şekil 5.1. KAA modelleme ve benzetim süreci [65].

‘temel ağ bileşenleri’ olarak tanımlanır. Düğümler kullanılarak çeşitli ağ topolojileri oluşturulur. Düğümler, kapsama alanları ve paketlerin bir araya gelmesinden oluşan ağ modeli ‘DEVS birleşik ağ modeli’ olarak adlandırılır. Kullanılan atomik ve birleşik modeller, Bölüm 3’te tanımlanan ‘Paralel DEVS yaklaşımı’ kullanılarak tanımlanır ve ‘DEVS-Sensor’ modelleme ve benzetim ortamında uygulanır. Bu yöntemde, düğümler ve diğer ağ bileşenleri, ağ modelinin davranışını (çıkış zamanı gibi) belirlemek için kullanılır [65].

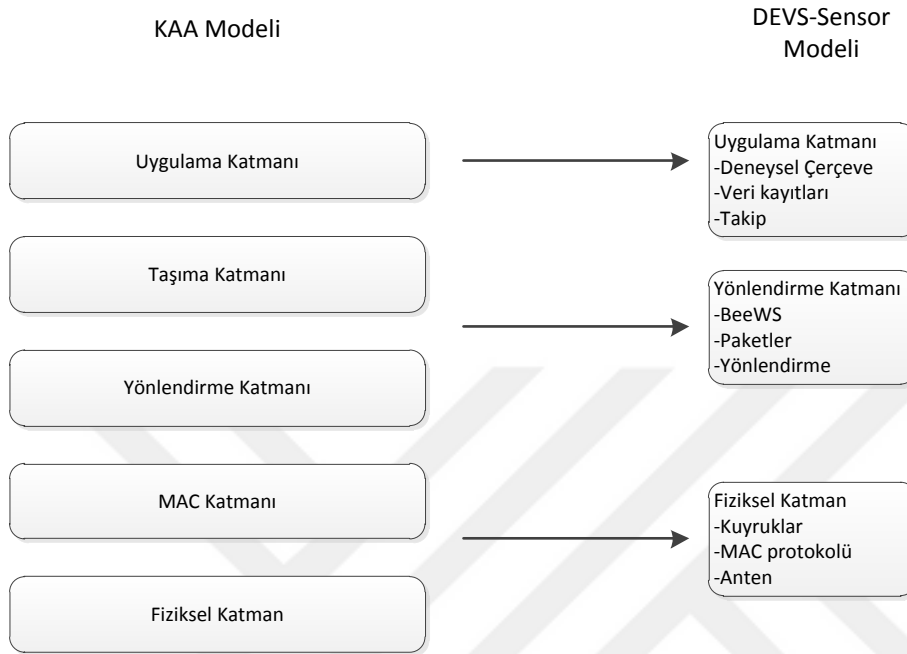
Şekil 5.2’de geliştirilen yönlendirme algoritmasının DEVS-Sensor benzetim aracında modellenme ve benzetim süreci şema olarak gösterilmiştir. Şekil 5.2’de geliştirilen BeeWS modeli, deneysel çerçevesi ile DEVS çekirdeği, bu çekirdeğe bağlı olan DEVS-Sensor modülleri ile bunlar arasındaki bağlantılar görülmektedir.



Şekil 5.2. KAA'ların model bileşenleri ile DEVS-Sensor'ın kavramsal görünümü



Şekil 5.3’de KAA katmanlarının benzetim ortamında modellenirken karşılık gelen araçları verilmiştir.



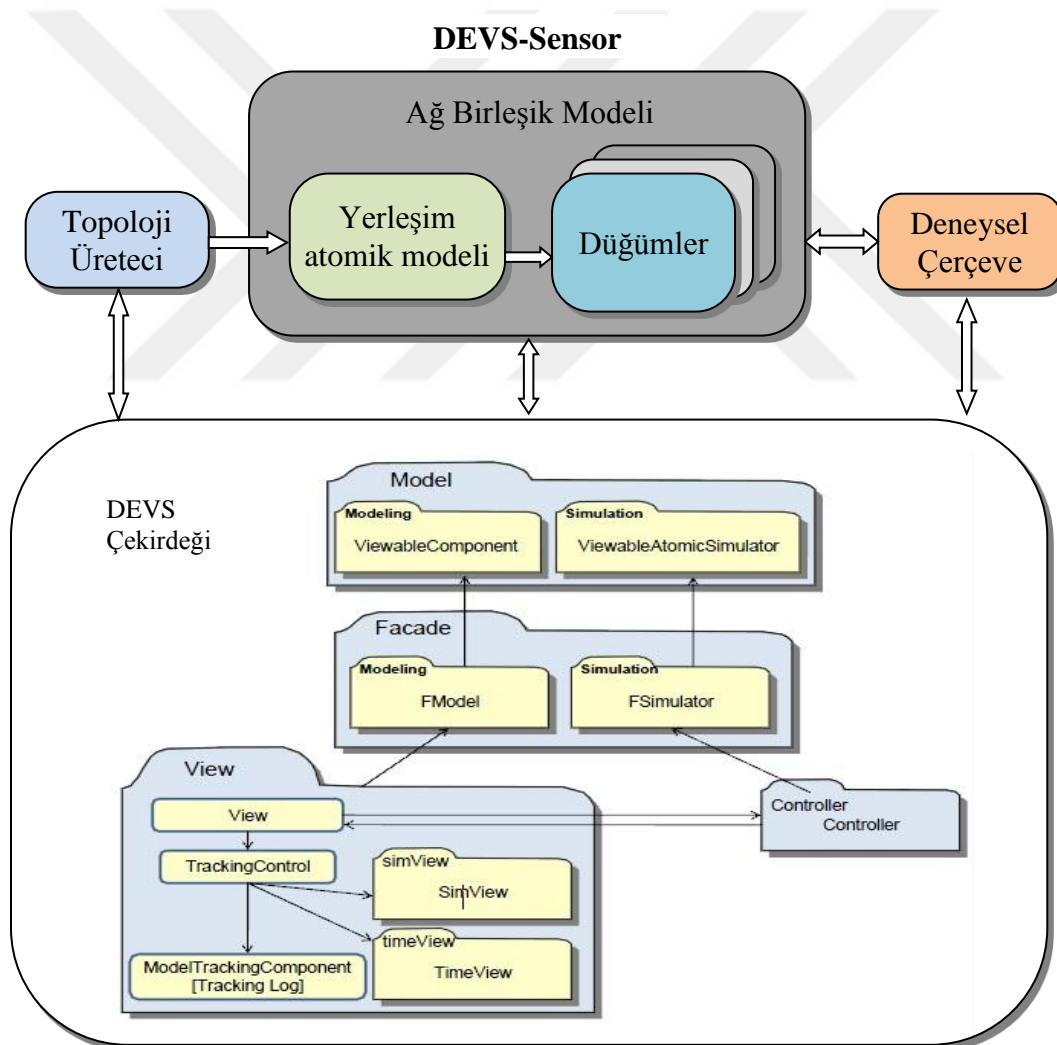
Şekil 5.3. DEVS-Sensor ile ağ katmanlarının eşleşmesi

Gerçekleştirilen çalışmada deneyler için, bir modelin deney yapılacağı ve gözlemleneceği şartları tanımlayan ‘DEVS deneysel çerçeve’ kavramından faydalanılmıştır. Ağ trafiğini üretmek ve düğüm gibi ağ bileşenlerinin arızalanmaları ve hizmet verememeleri gibi ağın çalışması sırasında meydana gelebilecek muhtemel olayları önceden programlamak ve zamanlamak amacıyla ağdaki ilgili bileşenlere DEVS mesajları gönderen bir ‘olay üretici’ kullanıldı [65].

DEVS-Sensor benzeticisinin kavramsal modeli en temel haliyle Şekil 5.4’de görülmektedir. Şekilde görüldüğü gibi DEVS-Sensor benzeticisindeki bileşenler DEVS çekirdeği ile beraber çalışmaktadır.

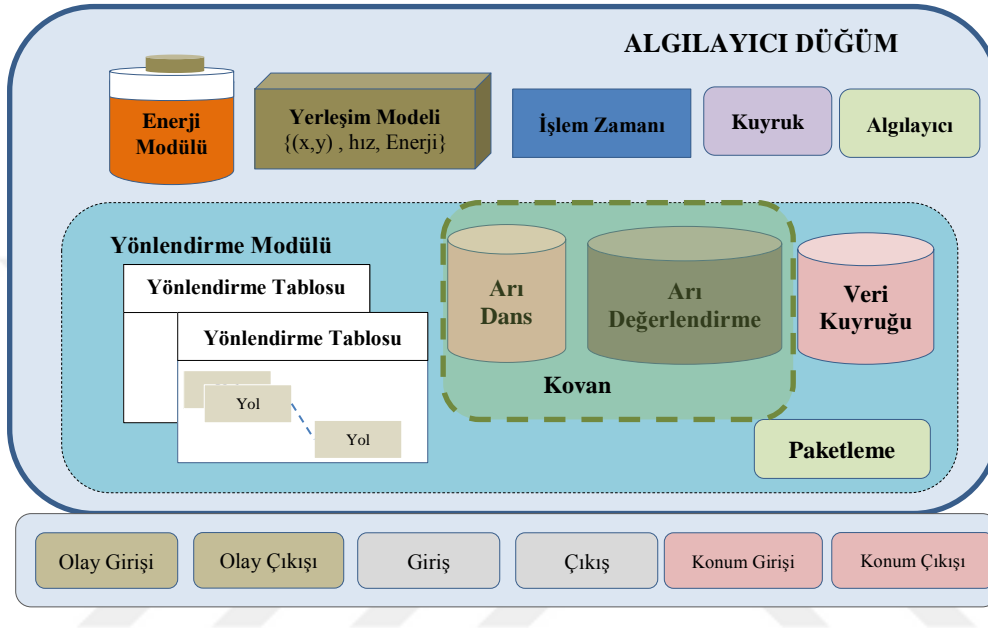
### 5.4.1. Kablosuz algılayıcı atomik modeli

Düğümün her biri, oluşturulan topolojideki paketleri işleyebilecek ve bu paketleri istenilen yerlere yönlendirebilecek şekilde tasarlanmıştır. Düğümler sistemde gerçekte olduğu gibi kapsama alanı dâhilindeki düğümlerle bağlanıp haberleşebilen atomik modellerdir. Düğümlerin davranış karakteristikleri, trafiğin işlenmesi için bant genişliği, işlem yapma hızı ve trafiği işleyebilecek kapasitede sınırlı bir tampon boyutudur. Tanımlanan karakteristiklerle oynanarak değişik kapasiteye sahip ağ bileşenleri oluşturulabilir ve farklı ağ senaryoları geliştirilebilir [65].



Şekil 5.4. DEVS-Suite ve DEVS-Sensor benzetici kavramsal modeli gösterimi

KAA kendi kendine organize olabilen düğümlerden oluşur. Bu nedenle topolojideki sürecin tamamı düğümler üzerinde yürür. Düğümler birer karar merkezi gibi çalışırlar. Yönlendirme işlemlerinin kararlarını verebilmek amacıyla yönlendirme tabloları oluştururlar. (Şekil 5.5)

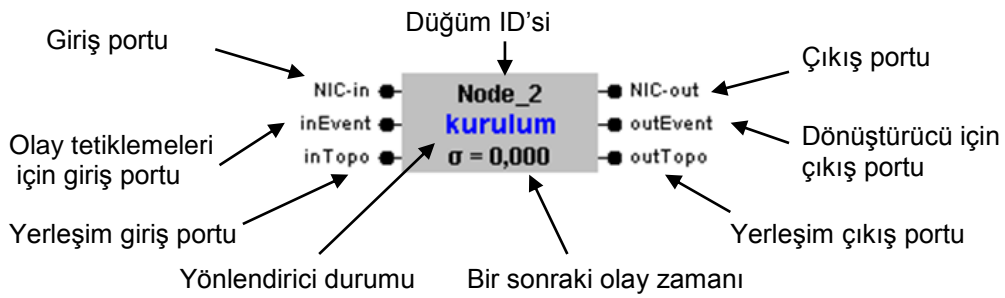


Şekil 5.5. Algılayıcı düğümün kavramsal modeli

Şekil 5.5’de bir düğüm mimarisi görülmektedir. Şekilde görüldüğü gibi en önemli modül yönlendirmedir. Bu modül gelen paketlerden yollar oluşturup bu yolları kullanarak veri paketini gönderme işlemini gerçekleştirir.

Sistemde basit bir MAC protokolü tasarlanmıştır. MAC protokolleri MAC katmanında iki düğüm arasındaki veri akışını kontrol eder. Fiziksel katman ile yönlendirme katmanı arasında bulunur. Kullanılan MAC protokolünün çalışması ideal kabul edilmiştir.

Bir düğümün DEVS-Suite ekranında görsel olarak görünümü Şekil 5.6’da görülmektedir. Düğümde paket gönderme ve alma işlemini gerçekleştiren portlar, düğüm kimliği, yönlendiricinin durumu, bir sonraki olay zamanı bulunmaktadır.

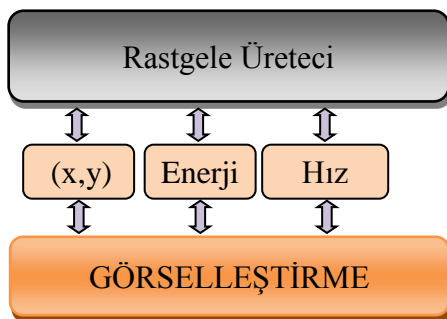


Şekil 5.6. Düğüm atomik modelinin ekran çıktısı

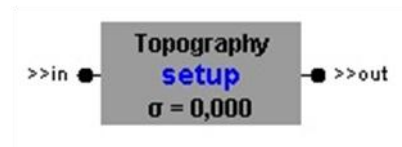
Şekil 5.8’de DEVS-Sensor benzetim aracında kullandığımız sınıfları ve bunlar arasındaki bağlantılar gösterilmektedir. Şekil 5.9’da ise DEVS-Sensor için geliştirilen birleşik modellerin birbirleriyle olan yapısal ilişkisi verilmiştir.

#### 5.4.2. Yerleşim atomik modeli

Düğümün yerleştirilmesi, enerji kontrolü ve hareketlilik için yerleşim (topography) atomik modeli kullanılır. Hareketli topolojilerde bağlantıların dinamik olarak güncellenmesi gerekmektedir. Benzetim ortamının hızlı çalışması için bu görevi yerleşim atomik modeli yerine getirir. Bunun yanında enerjisi biten düğümlerin sistem tarafından algılanma sürecinide gerçekleştirir. Şekil 5.7 (a)’da yerleşim atomik modelinin kullandığı değişkenler ve çalışma sistemi gösterilmiştir. Şekil 5.7 (b)’de yerleşim düğüm atomik modeli gösterilmiştir



(a)



(b)

Şekil 5.7. Yerleşim atomik modelinin kavramsal modeli (a) ve ekran çıktısı (b)

### 5.4.3. Ağ paketleri

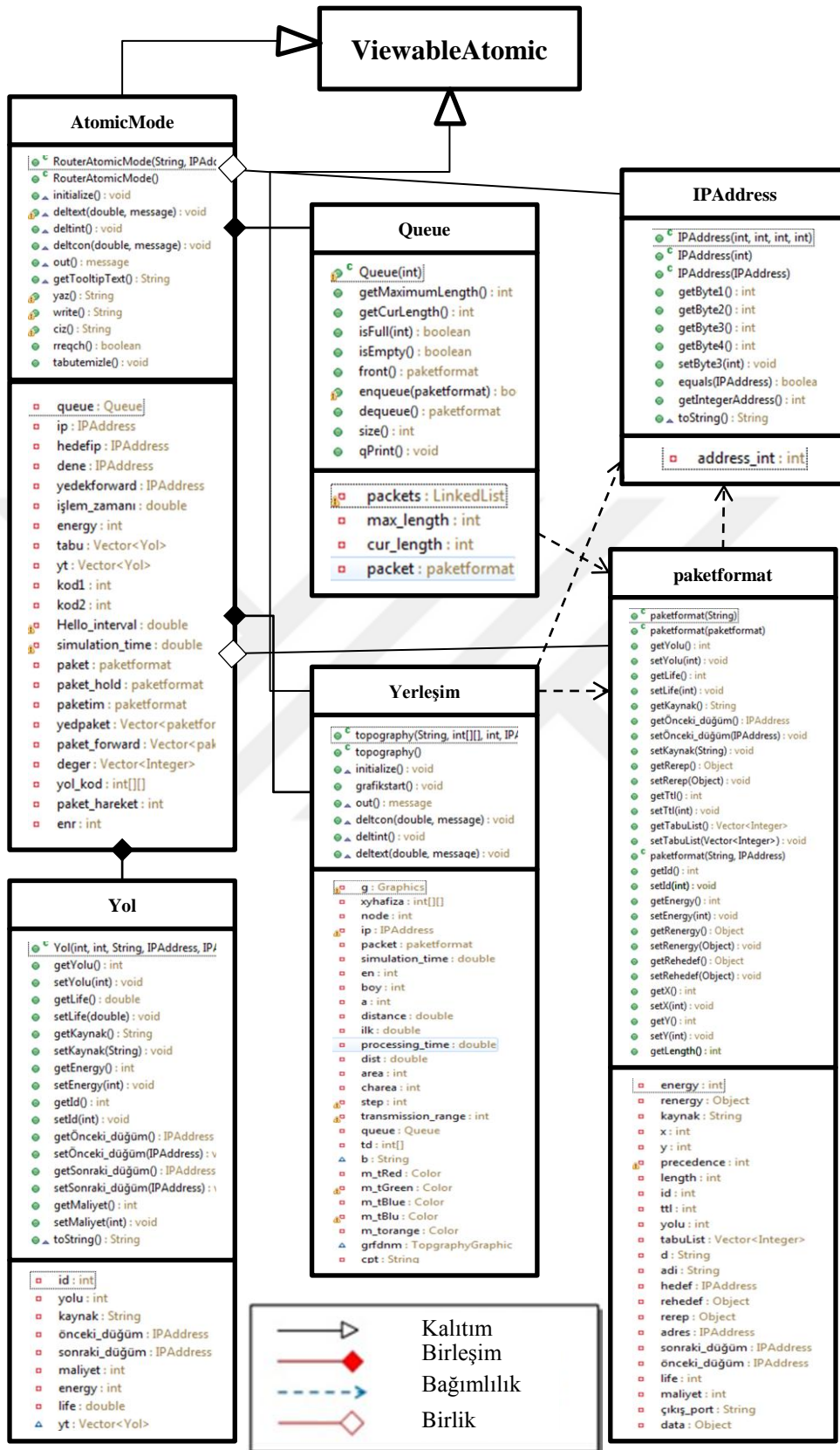
Ağ topolojisinde iki farklı tipte paket vardır. Bu paketler, veri ve kontrol paketleridir. Veri paketleri, oluşturulan verilerin hedefe taşınma sürecinde kullanılırken, kontrol paketleri yol bulma ve komşu tanıma gibi ağ izleme ve yönetimi işlerini yerine getirirler. Kontrol ve veri paketlerinin yapıları birbirlerinden farklıdır.

Kontrol paketleri arı veya karınca gibi sosyal canlıların kullanıldığı algoritmalarda gözcü, izci, paketleyici vb. olarak isimlendirilirler. Geliştirilen kontrol paketleri, aktif, proaktif, öğrenme ve iletişim kurma yeteneğine sahip, hareket edebilen zeki ajan paketleri olarak modellenmiştir.

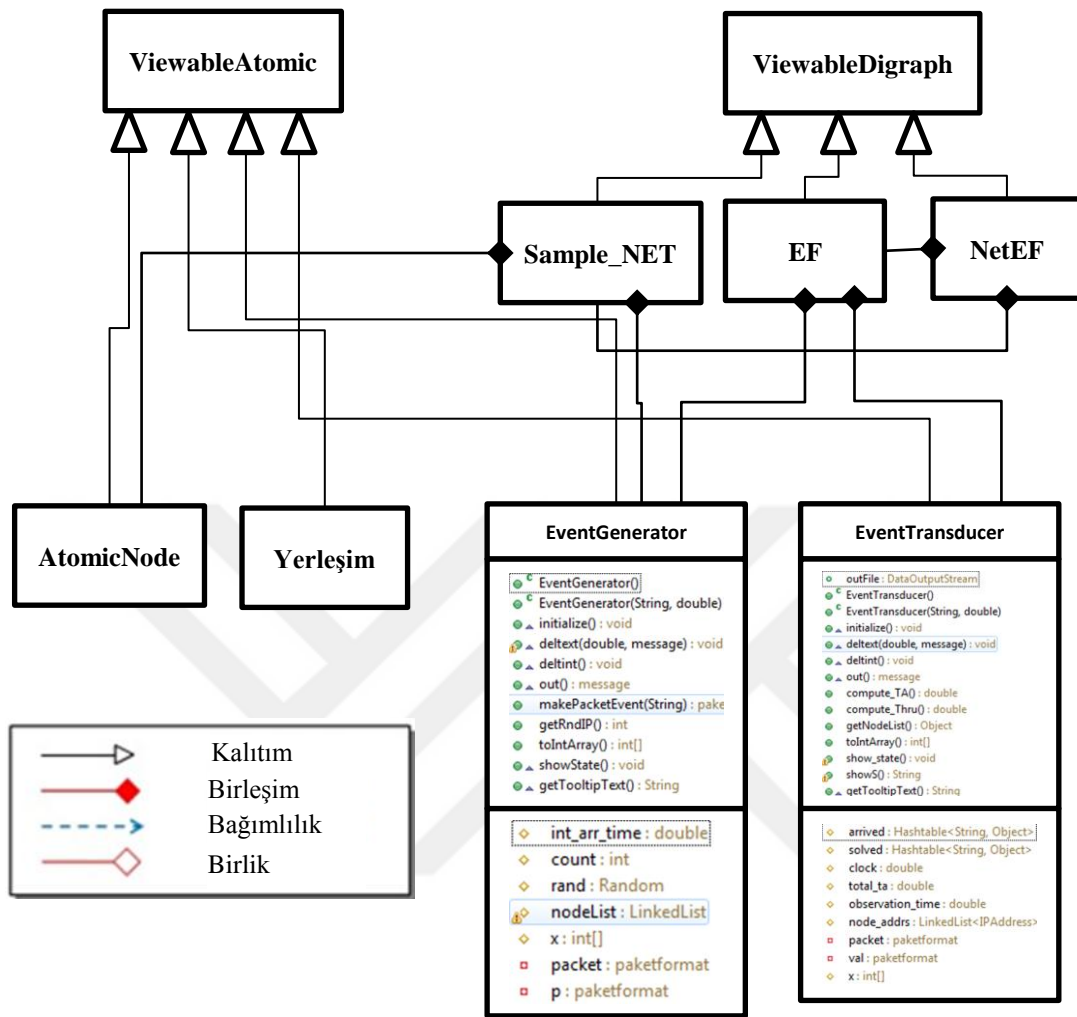
Paketler ağ topolojisinde DEVS modelinden geliştirilmiştir ve mesajlar halinde dolaşırlar. Java'nın nesnel yapısı sayesinde bu paketler istenilen kontrol paketi şekline dönüştürülebilmektedir. Düğümlere gelen paketler FIFO mantığına göre kuyruğa yerleştirilir. Algılayıcı atomik model gelen paketi yönlendirme kuralları çerçevesinde türüne göre işleme tabi tutar. Düğümlerin kuyrukları gerçekte olduğu gibi sınırlandırılmıştır. Bundan dolayı kuyruk dolu olduğunda paketlerde veri kaybı yaşanması söz konusudur. Kontrol paketlerinde ise yönlendirme kuralları çerçevesinde belirlenen paketin yaşam ömrü (TTL - Time to Live) sonlandığında paket düşürülmektedir.

Protokol başlığını (protocol header) oluşturan paket kimlik numarası, türü, yaşam ömrü, kaynak adresi, hedef adresi vb. bilgiler DEVS-Sensor ortamında paket nesnesi içerisindeki değişkenlerde saklanır. Paket nesnesi benzetim ortamına uygun şekilde gerçek ortam ile uyumlu olarak hazırlanmıştır. Şekil 5.10'da DEVS-Sensor ortamında bir paket nesnesinin alanları gösterilmektedir.

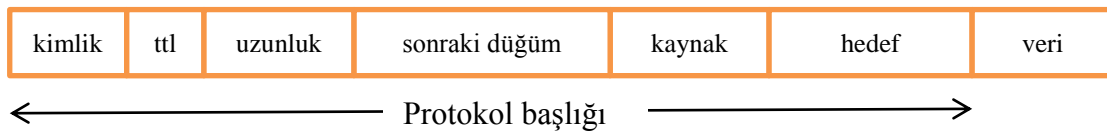
Veri paketinin takip edeceği yol düğümde bulunmuyorsa düğüm bir kontrol paketi üretir. Veriyi gönderecek düğümde veri paketleri farklı bir kuyrukta tutulur ve yol bulunana kadar bu kuyrukta bekletilir. Böylece veri paketlerinin içeriği bozulmadan en iyi şekilde hedefe ulaştırılması sağlanmış olur.



Şekil 5.8. DEVS-Sensor sınıfları ve yapısal ilişkileri



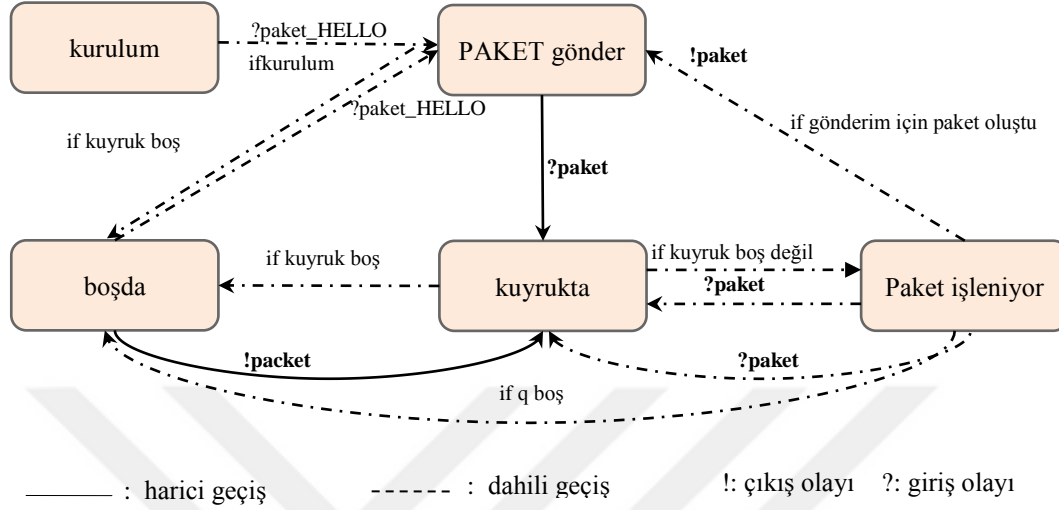
Şekil 5.9. DEVS-Sensor birleşik model sınıfları



Şekil 5.10. Protokol başlığı ve veri paketlerinin gösterimi

Şekil 5.11’de DEVS-Sensor benzetim aracında geliştirilen bir düğüm modelinde meydana gelebilecek durumlar diyagram olarak verilmiştir. Benzetim aracının çalışması başlangıç aşamasındaysa düğüm “kurulum” durumundadır. Bir paket

geldiyse “PAKET gönder” durumunda, “kuyrukta”, “Paket işleniyor” durumunda veya “boşta” olabilir.



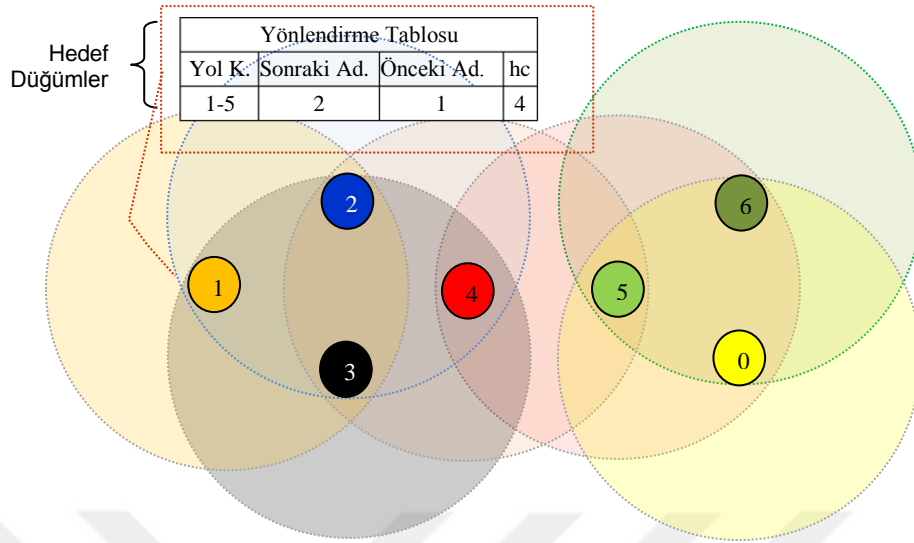
Şekil 5.11. Bir düğüme ait sadeleştirilmiş durum diyagramı

#### 5.4.4. Yönlendirme tabloları

Ağdaki hedefe bağlı yol bilgilerini yönlendirme tabloları tutar. Geliştirilen modeldeki yönlendirme tabloları hedefe düğümün kendi ulaşacağı veya düğümü kullanarak ulaşılan yolları tutmaktadır. Yönlendirme tabloları sahip olunan yol bilgisine göre değişkenlik göstermektedir. Şekil 5.12’de temsili olarak bir KAA sunulmuştur. 1 numara ile gösterilen düğüm kaynak düğüm olarak belirlenmiştir. 0 numara ise hedef düğüm olarak belirlenmiştir. Şekilde, 1’den 0’a giden yolda oluşturulan yönlendirme tablosu gösterilmektedir.

Yönlendirme tabloları yönlendirme algoritmaları tarafından oluşturulur. Oğul zeka tabanlı yönlendirme algoritmaları dinamik olarak çalışır. Bu sebeple yönlendirme tabloları ağ kurulumunda oluşturulmaz. Yönlendirme tabloları veri paketleri üretildikçe oluşturulur.





Şekil 5.12. Bir yönlendirme tablosu örneği

BeeWS yönlendirme protokolünde örnek bir yönlendirme tablosu Şekil 5.12’de gösterilmektedir. Yönlendirme tablosunda yol kimliği, hedefe ulaşmak için sonraki adım, kaynak tespiti için önceki adım ve yolun atlama sayısı tutulur. DEVS-Sensor ortamında bir düğümün yönlendirme tablosu Şekil 5.13’de görülmektedir.

IP : 0.1.0.1									
YT:									
Yol	id	izci.id	önceki_düğüm	Sonraki_düğüm	hop sayısı	Enerji	Ömür		
12	12	data0	0.1.0.2	0.1.0.0	3	799927	0.0		
17	17	data2	0.1.0.2	0.1.0.0	3	799918	0.0		
1	1	data9	0.1.0.1	0.1.0.0	1	199974	0.0		
12	12	data5	0.1.0.2	0.1.0.0	3	799906	0.0		
2	2	data16	0.1.0.2	0.1.0.0	1	399904	0.0		
18	18	data11	0.1.0.2	0.1.0.0	2	599857	0.0		
17	17	data9	0.1.0.2	0.1.0.0	3	799804	0.0		

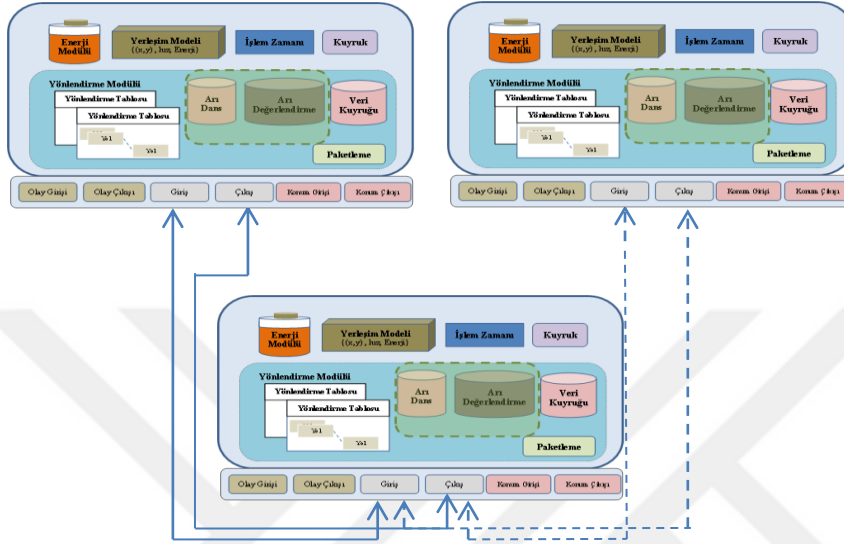
Gezgin kimliği      Önceki düğüm      Sonraki düğüm      Atlama sayısı      Enerji      Düğüm IP adresi

Şekil 5.13. Bir yönlendirme tablosunun DEVS-Sensor ekran görüntüsü

#### 5.4.5. Birleşik (Coupled) model

Atomik modellerini oluşturduğumuz düğümleri DEVS bileşik model yaklaşımını kullanarak bir araya getirip birleşik bir KAA modeli oluşturabiliriz. Birleşik model

yaklaşımı sayesinde farklı tipte ağ topolojileri oluşturabiliriz. Şekil 5.14’de, 3 düğümden oluşan bir birleşik model görülmekte ve Şekil 5.15’de ise DEVS-Sensor ortamında düğümlerin birbirleri ile olan bağlantıları görülmektedir.



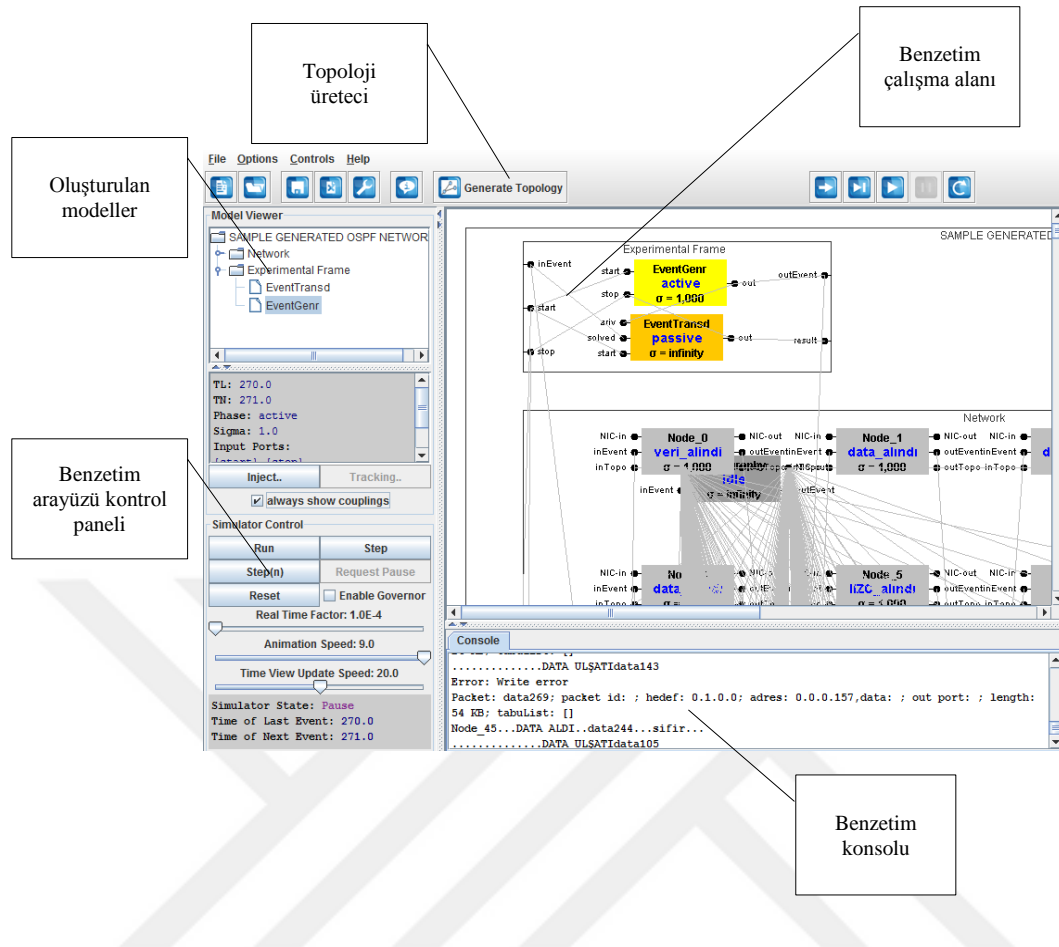
Şekil 5.14. Düğümlerin iletişim kurması

DEVS-Sensor DEVS birleşik model kavramından faydalanılarak geliştirilmiş bir benzetim aracıdır. Bu özelliği sayesinde farklı ağ topolojileri oluşturulmuş ve test edilmiştir.

#### 5.4.6. Ağların ölçeklenmesi

Daha öncede ifade edildiği gibi, büyük ölçekli ağların oluşturulmasında başlıca problem ölçeklenebilirliktir. Yönlendirme protokollerinde ağların ölçeklenebilmesi için Bölüm 2’de bahsedilen kümeleme yöntemleri kullanılmaktadır.

Yapılan çalışmada yüksek sayıda düğümlere ulaşabilmek için yeni bir kümeleme yöntemi geliştirildi. Bölüm 4’de mimarisi anlatılan BeeWS yönlendirme algoritmasının kümeleme yöntemi kullanılarak gerçekleştirilmesi için DEVS-Sensor benzetim ortamı kümeleme yöntemine uygun hale getirildi. DEVS-Sensor benzetim



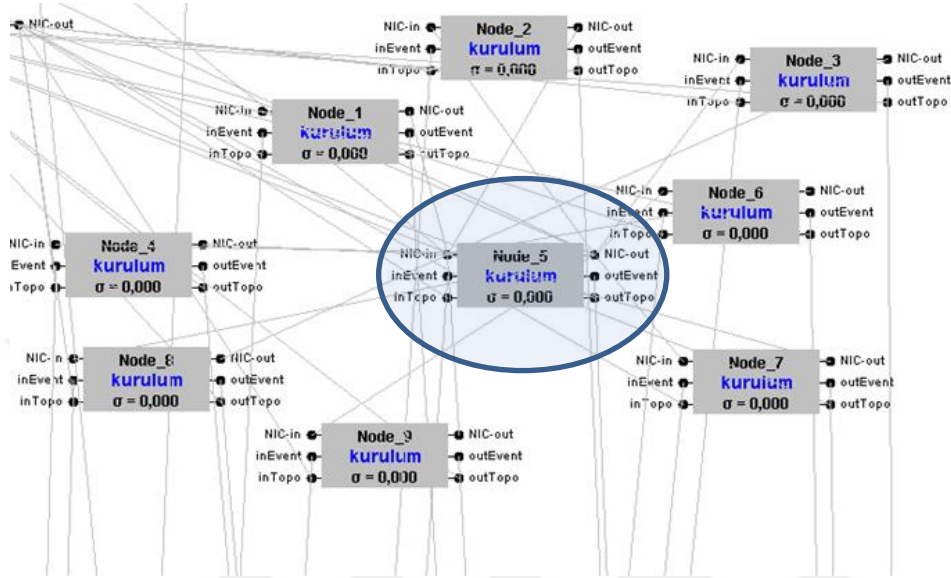
Şekil 5.15. DEVS-Sensor ortamında bir ağ modeli

ortamında oluşturulan kümeler kümebaşı seçilen düğümlere diğer üye düğümlerin bağlanmasıyla meydana gelir. Kendisine üye düğümlerin bağlandığı kümebaşları da kendi aralarında organize olarak bir ağ topolojisi oluşturur. Oluşturulan bu ağ topolojisinde hedefe ulaşma çoklu-atlama (multi-hop) yöntemi kullanılarak yapılır. DEVS-Sensor benzetimin ortamı Şekil 5.16'da görüldüğü üzere, 5 numaralı düğüm kümebaşı olarak belirlenmiş ve kapsam alanında bulunan düğümler kendisine bağlanmıştır. Üye düğümler sadece 5 numaralı düğümle bağlantı kurmuşlardır. 5 numaralı düğüm ise diğer kümebaşları ve üye düğümlerle bağlantı kurmuştur.

## 5.5. Topoloji Üretici

Benzetim ortamları modelleme ve benzetim araçlarını geliştirir. Bu araçlar ile gerçek dünya ortamlarına uygun ağ topolojileri oluşturma görevini ise topoloji üreticileri

üstlenir. Bundan dolayı topoloji üretici benzetim araçları için önemli bir bileşendir. İyi bir topoloji üreticine sahip olmayan benzetim aracı gerçek dünya verilerini kullanamayacağından verimli sonuçlar üretmez.



Şekil 5.16. DEVS-Sensor benzetim ortamında kümeleme

BRITE topoloji üretici gerçeğe yakın benzetim çalışmaları ve ağın başarımlı karakteristiklerinin incelenmesi için araştırmacılar tarafından sıklıkla kullanılmaktadır. BRITE topoloji yapısının iyi modellenmiş olması iletişim tekniklerinin analizi ve geliştirilmesi için önemli katkı sağlamaktadır. Ağ iletişimde, verimli protokol tasarımında, problemlerin (yönetim, yönlendirme, kaynakların kullanılması vb.) çözümünde, benzetim için doğru modelin kurulmasında ve hata toleransı çalışmalarında BRITE topoloji üreticinin önemi çok büyüktür. Özetle ağın başarımlı önemli ölçüde BRITE topoloji üreticine dayanmaktadır [82].

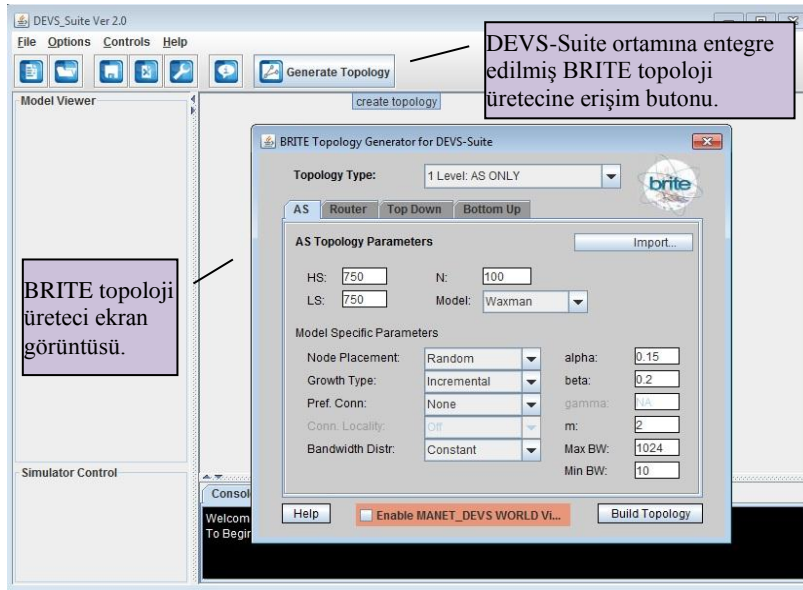
BRITE topoloji üretici İnternet topolojilerini doğru bir şekilde üreterek protokol ve algoritmaların doğruluğunu araştıran ve yeni nesil güçlü modeller üzerinde çalışan araştırmacıların kullanımına sunar [83].

Aşağıda BRITE topoloji üreticinin karakteristik özellikleri listelenmektedir [83].

- Temsil edilebilirlik: Doğru yapay topolojiler üretir. İnternet topolojilerini farklı açılardan doğru yansıtır.
- Kapsamlılık: Tek bir topoloji üreticinde mümkün olduğunca fazla üreteç modeli barındırır.
- Esneklik: Geniş ölçekte topoloji üretebilir, mantıklı sayı aralığında minimum ve maksimum düğüm sınırlaması getirmez.
- Verimlilik: Büyük ölçeklerde topoloji üretirken CPU ve belek tüketiminde ekonomik davranır.
- Genişleyebilirlik: Kullanıcı tarafından yeni modellerin eklenebildiği mekanizmaları destekler.
- Kullanıcı dostu: Standart kullanıcı arayüzü prensiplerini kullanır. Kullanıcılar tarafından üreticinin mekanizmasının kolay öğrenilmesini sağlar.
- Karşılıklı çalışabilirlik: Benzetim ve görselleştirme araçları için arayüzler sağlar. Geniş kullanıma sahip benzetimler tarafından kullanılmasına imkân verir.
- Sağlamlık: Geniş hata sezme yeteneğine sahiptir.

Topoloji üreteçleri ile ilgili önemli bir çalışma M. A. Rahman ve arkadaşları tarafından yapılmıştır [82]. BRITE (The Boston University Representative Internet Topology Generator) [82] topoloji üreticinin büyük ölçekli ağları desteklemesi, kolay topoloji oluşturmaya imkan tanınması, açık kaynak kodlu olması nedeniyle bu çalışmada tercih edilmiştir. (Şekil 5.17)

BRITE topoloji üretici DEVS-Sensor benzetim ortamına uyarlanarak eklenmiştir. BRITE topoloji üretici dahil edilerek DEVS-Sensor ortamında istenilen boyutta KAA oluşturulmaktadır. BRITE topoloji üreticinde oluşturulmak istenen KAA alanı,



Şekil 5.17. DEVS-Sensor BRITE topoloji üretici ekran görüntüsü

düğüm sayısı, ağ modeli, düğüm yerleşimi gibi seçenekler bulunmaktadır. bu özellikler sayesinde oluşturulan KAA modeline tam bir müdahale gerçekleştirmek mümkün olmaktadır.

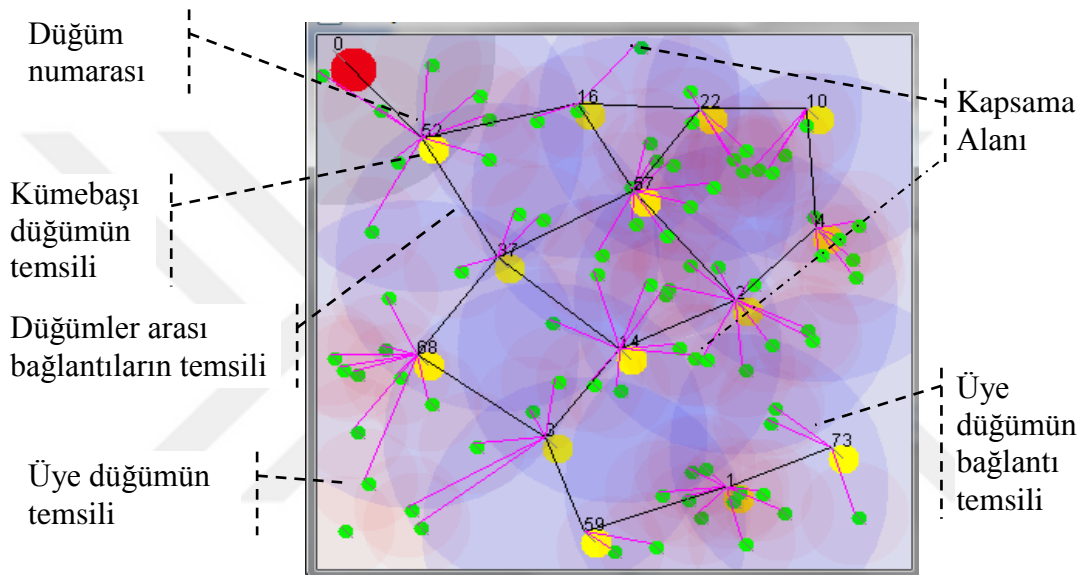
## 5.6. Görsel Takip

Oluşturulan ağ topolojilerinin gerçek ortamdaki konumlar ve hareketlerini gözlemleyebilmek için bir grafiksel arayüz geliştirilmiştir. Böylece yapılan çalışmanın rasgele düğüm yerleşimleri, dağılım alanları, kapsam alanları ve birbirleriyle olan bağlantıları gözlenebilmektedir.

DEVS-Sensor görselleştirme ekranında iki pixel arası 1m olarak kabul edilmiştir. Şekil 5.18'de 350 X 350 bir alanda 100 düğümden oluşan bir topolojinin ekran görüntüsü verilmiştir.

Geliştirilen görselleştirme ekranı DEVS-Sensor ortamında oluşturulan KAA'nın gerçek dünyaya uyarlanmış halini göstermektedir. Gerçekte olduğu gibi Şekil 5.18'de 350m X 350m bir alanda 100 düğüm rasgele ortama dağıtılmıştır. Yeşil

düğümler üye düğümleri göstermekte, sarı düğümler ise kümebaşlarını ifade etmektedir. Kırmızı düğüm hedef ya da baz istasyonu olarak gösterilmektedir. Kırmızı saydam daireler üye düğümlerin kapsam alanı, mavi saydam yuvarlaklar ise kümebaşlarının kapsam alanını göstermektedir. Siyah bağlantılar kümebaşları arasında kurulan ağı, mor bağlantılar ise üye düğümlerin kümebaşlarıyla olan bağlantısını göstermektedir. DEVS-Sensor ortamında KAA'da olan değişimler aynen görselleştirme ortamında görüntülenmektedir.

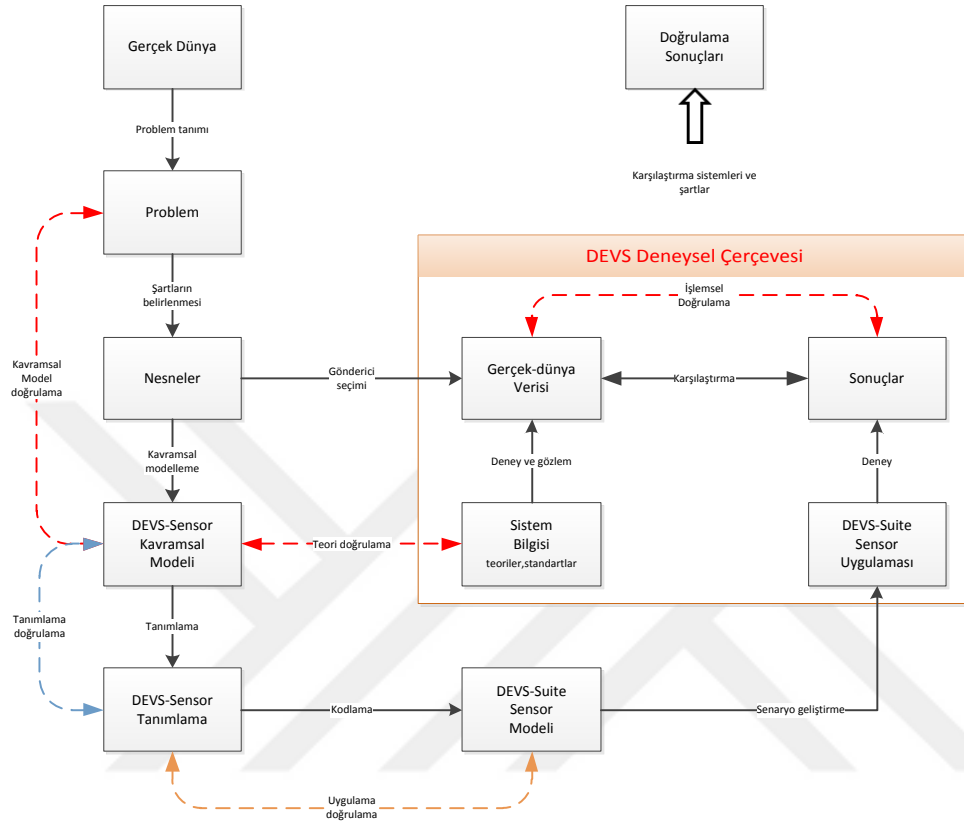


Şekil 5.18. DEVS-Sensor görselleştirme ekranında 350X 350 lük bir alanda topoloji dağılımı

## 5.7. Benzetim Onaylama ve Geçerleme

Şekil 5.19'da görüldüğü gibi onaylama ve geçerleme süreci dağıtık olaylı modelin ve benzetimin deneysel çerçevede çalışmasıyla gerçekleşir. Bu süreç DEVS-Sensor modelinin onaylama ve geçerleme için deneysel çerçeve ile birlikte kullanılarak örneklenmesinden oluşmuştur. DEVS modelleme sürecinde deneysel çerçeve model hakkında ayrıntılı karar vermek için kullanılmıştır. Şekil 5.19'da gerçekleştirilen onaylama ve geçerleme süreci görülmektedir. Problemin tanımı ve nesnelerin tanımı yapılmıştır. Kavramsal modeli onaylama önerilen modelin detay seviyesini belirlemek için yapılır. Teorik onaylama geliştirilen modelin standart, yaklaşım,

biçimsel tanımları gibi teknik kriterler ile ilgilidir. Örneğin BeeWS geliştirilirken Bölüm 4’de verilen kriterler gibi.



Şekil 5.19. DEVS modelleme süreci ve deneysel çerçeve doğrulanması ve geçerliliği

Çalışmanın doğrulanması sürecinde gerçek dünyadan doğru veriler elde edilmiş ve gerekli modeller doğrulanmıştır. Doğrulanmış modeller, elde edilen bilgiler, benzetim algoritması, yaklaşımlar olarak gerçekleştirilmiştir.



## **BÖLÜM 6. DENEYLER ve SONUÇLAR**

### **6.1. Giriş**

Bölüm 5’de anlatılan DEVS-Sensor benzetim ortamında Bölüm 4’de anlatılan KAA’lar için geliştirilen oğul zeka tabanlı yönlendirme algoritması gerçekleştirildi. Bu bölümde gerçekleştirilen çalışma ile ilgili test ve deney sonuçları ortaya konulmaktadır.

BeeSensor algoritması literatürdeki kablosuz algılayıcı ağlar için geliştirilen algoritmalar içinde tek arı kolonisi tabanlı algoritmadır. EEABR enerji alanında literatürde verimli sonuçlar veren oğul zekası tabanlı bir algoritmadır. FP-Ant algoritması paket teslimi alanında literatürde verimli sonuçlar veren oğul zekası tabanlı algoritmadır. AODV algoritması kablosuz araçlar alanında temel esasları oluşturan bir algoritmadır.

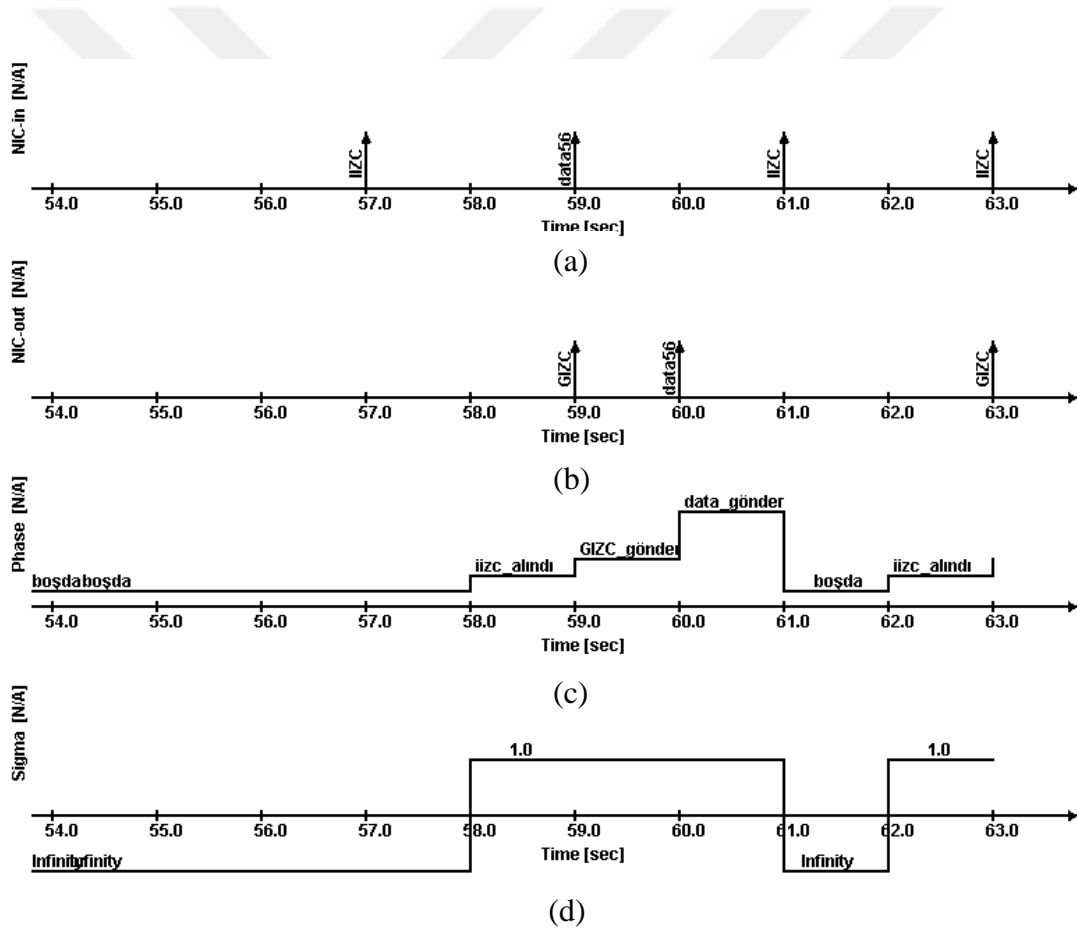
Bu sebeplerden dolayı çalışmada geliştirdiğimiz KAA’lar için oğul zekası tabanlı yönlendirme algoritması modelini literatürde var olan EEABR, FP-Ant, AODV ve BeeSensor algoritmalarıyla enerji, gecikme ve paket teslim oranı açısından karşılaştırdık. Bu karşılaştırmalardan elde ettiğimiz sonuçları grafikler ile gösterdik.

Çalışmadaki önemli noktalardan biri ölçeklemedir. Geliştirdiğimiz kümeleme mantığını benzetim ortamına uygulayarak büyük ölçekli ağlar kurmayı başardık. Bunun sonucunda elde edilen sonuçları grafik ortamında, enerji, gecikme, paket teslim oranı ve küme sayısı grafikleriyle ortaya koyduk.

Gerçekleştirilen deney ve sonuçlarla, yapılan KAA’lar için oğul zeka tabanlı yönlendirme algoritmasının üstün özellikleri ve DEVS-Sensor benzetim ortamının KAA’ların benzetiminde uygun bir ortam olduğu gösterilmektedir.

## 6.2. Deneysel Çerçeve ve Benzetim Ortamı

KAA'ların modellenmesinde en uygun yöntem olan ayrık olaylı modelleme yöntemi DEVS yaklaşımı kullanılarak geliştirilen DEVS-Sensor benzetim ortamı yapılan çalışmada kullanılmaktadır. DEVS-Sensor benzetim ortamının ayrık olay çalışma grafiği Şekil 6.1'de görülmektedir. Şekil 6.1 (a)'da bir düğüme gelen paketler ve gelme zamanları görülmektedir. Şekil 6.1 (b)'de bir düğümden giden paketler ve çıkış zamanları görülmektedir. Şekil 6.1 (c)'de düğümde oluşan durumlar ve zamanları görülmektedir. Şekil 6.1 (d)'de düğümde işlem oluşumuna gerçekleştiren sigma değeri zamana bağlı olarak görülmektedir.



Şekil 6.1. BeeWS modelinin DEVS-Sensor benzetim ortamında çalışma grafiği

Karşılaştırmalar, EEABR, FP-Ant, AODV, BeeSensor yönlendirme algoritmalarının sonuçları kullanılarak yapıldı. Bu karşılaştırmalar sonucunda kendi geliştirdiğimiz protokolle beraber beş farklı protokolü karşılaştırma imkanı bulduk. Uygulamada

birçok düğümden tek bir hedefe (Converge-cast scenario) veri akışı olacak şekilde senaryo oluşturuldu. Uygulamalar boyunca olay üreticimiz her saniyede bir olay olacak şekilde 512 bit büyüklüğünde paketler üretti.

Benzetim deneyleri üç aşamada gerçekleştirildi. Birinci aşama, EEABR, FP-Ant, AODV, BeeSensor yönlendirme algoritmalarıyla BeeWS algoritmasının karşılaştırılması, ikinci aşamada kümeleme yöntemi uygulanarak geliştirilen BeeWS kümelemeli ile kümelemesiz algoritmaların karşılaştırılması, üçüncü aşamada ise kümeleme yöntemi ile büyük ölçekli ağlar oluşturulması ve deney sonuçlarının grafik ortama aktarılması gerçekleştirilmiştir.

Düğüm deneyler boyunca rastgele yerleştirilip, 300 saniye boyunca sonuçlar gözlemlendi. Her grafik on ayrı çalışmanın ortalamasından elde edildi. Yapılan deneylerde Berkeley mote CC2420 MICAz algılayıcı düğümünün parametreleri kullanıldı.

Tablo 6.1. MICAz parametreleri [81].

<b>Parametre</b>	<b>MICAz</b>
Chipset	CC2420
Radyo frekansı(MHz)	2400-2483,5
Maksimum veri oranı (kbps)	250
RX gücü (mW)	19,7
TX gücü (mW)	17,4
Uyku gücü ( $\mu$ A)	1
Modulasyon	DSSS-O-QPSK
Alım hassasiyeti	-94 dBm
Kapsam alanı	100 m ye kadar
Çoklu kanal	Evet

Kablosuz algılayıcılarda uyku durumu, enerji tüketimi hesaplanırken çok küçük olduğundan önemsenmedi. Çalışmada pixeller arası 1m olacak şekilde kabul edildi. Sonuçlar % olarak raporlandı.

Tablo 6.2. Benzetim parametre ve formülleri

Parametre	Açıklama	Formül
Benzetim alanı	Topolojinin kurulduğu alan	$w * h$
Düğüm yoğunluğu	Benzetim alanındaki düğüm yoğunluğu	$\frac{n}{w * h}$
Düğüm kapsama alanı	Düğümün kapsama alanlarının kapsama alanı	$\pi * r^2$
Benzetim alanı yüzdesi	Düğümün benzetim alanında kapsadığı alanın yüzdesi	$\frac{(\pi * r^2)}{w * h} * 100$
Maksimum yol	Kaynaktan hedefe paketin doğrusal olarak aldığı maksimum uzaklık	$\sqrt{w^2 + h^2}$
Ağ çapı	Birbirine en uzak iki nokta arasındaki minimum atlama sayısı	$\frac{\sqrt{w^2 + h^2}}{r}$
Komşu sayısı	Benzetim ve iletim alanına dayalı olarak komşu sayısı	$\frac{\pi * r^2}{\left(\frac{w * h}{n}\right)}$
w= genişlik, h= yükseklik, r= kapsama alanı, n= düğüm sayısı		

### 6.2.1. Değerlendirme ölçütleri

**Ortalama gecikme:** Bir kaynaktan üretilen bir olayın hedefe ulaşması için uçtan uca geçen zaman olarak tanımlanır. Ortalama gecikme değerlendirilirken kayıp olaylar dikkate alınmamıştır.

**Paket teslim oranı:** Ağdaki tüm kaynaklar tarafından üretilen olayların hedef tarafından alınan tüm olaylara oranıdır.

**Enerji tüketimi:** Hedefe varan 1000 bit verinin tükettiği enerjiye denir(j/Kbit).

**Küme sayısı:** Kümeleme işlemi gerçekleştirilirken oluşan küme sayısıdır.

### 6.3. Benzetim Sonuçları

#### 6.3.1. BeeWS ile BeeSensor, FP-Ant, EEABR ve AODV karşılaştırması

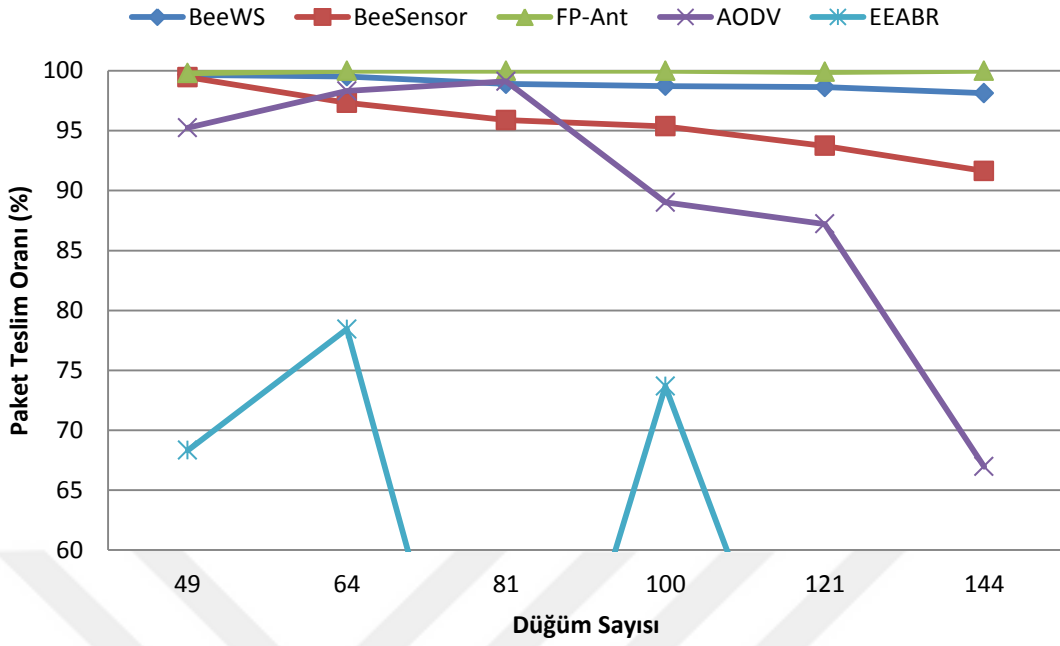
Deney ortamında, 49, 64, 81, 100, 121 ve 144 olmak üzere altı farklı KAA kullanıldı. 49 düğümlü ağda algılayıcılar 140x140 m alana rasgele dağıtıldı. Diğer topolojilerde de aynı düğüm yoğunluğu kullanıldı. Düğümlerin kapsama alanı 35 m olarak kabul edildi. Durağan (static) senaryo da deneyler gerçekleştirildi. Her bir düğümün başlangıç enerji seviyesi 20 joule olarak kabul edildi.

##### 6.3.1.1. Paket teslim oranı

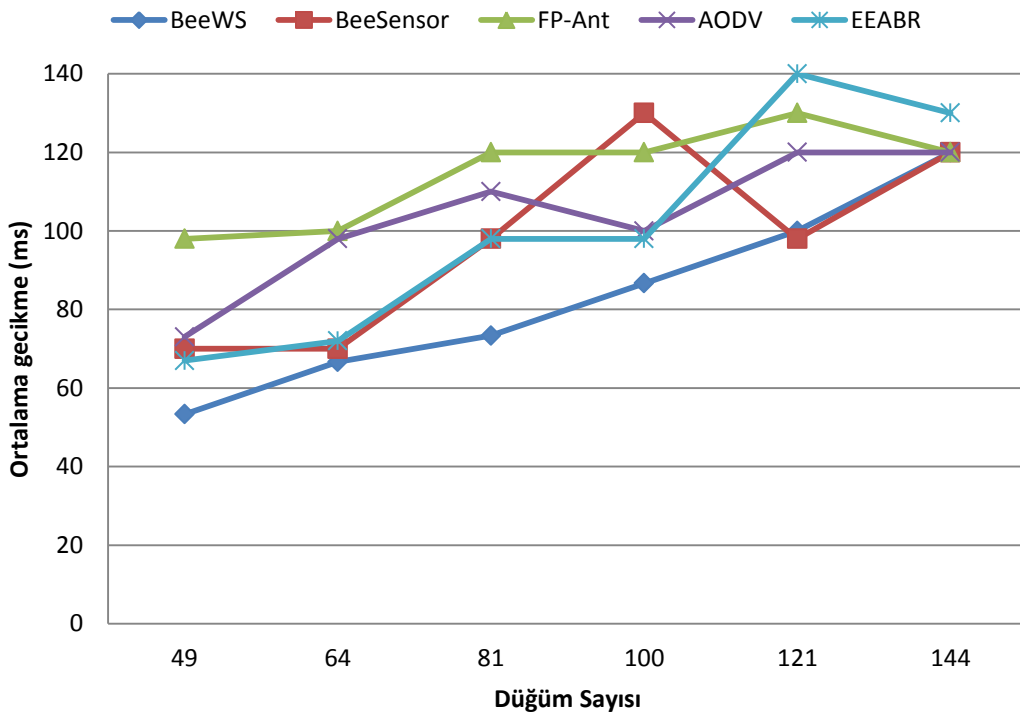
Şekil 6.2’de protokollerin paket teslim oranları görülmektedir. Şekilde görüldüğü üzere FP-Ant ve BeeWS en yüksek paket teslim oranına sahip protokollerdir. Bunun nedeni her iki protokolünde akış yöntemi kullanmasıdır. Ölçek büyüdükçe AODV ve EEABR protokollerinde teslim oranları büyük farklarla azalmasına karşın BeeWS yüksek teslim oranını korumuştur. Önemli bir gözlemde EEABR’deki tutarsızlıktır. Bunun nedeni EEABR’nin çok fazla kontrol paketi kullanmasıdır. Bu da topoloji büyüdükçe tutarsızlığa sebep olmaktadır.

##### 6.3.1.2. Ortalama gecikme

Şekil 6.3’de ortalama gecikme grafiği gösterilmektedir. Tüm protokoller yaklaşık olarak benzer ortalama gecikmelere sahiptirler. Küçük topolojilerde protokoller arasında az da olsa ortalama gecikme farkı gözlenirken bu fark topolojiler büyüdükçe azalmıştır. Ortalama gecikme topolojideki düğüm sayıları arttıkça artmaktadır. Geliştirilen protokolde ortalama gecikmeyi artıran etkenlerin başında dans bölümü denilen yol seçim bölümü vardır. Buna rağmen ortalama gecikmenin diğer protokollere göre az olmasının nedeni oluşan yolların belli bir süre saklanmasıdır.



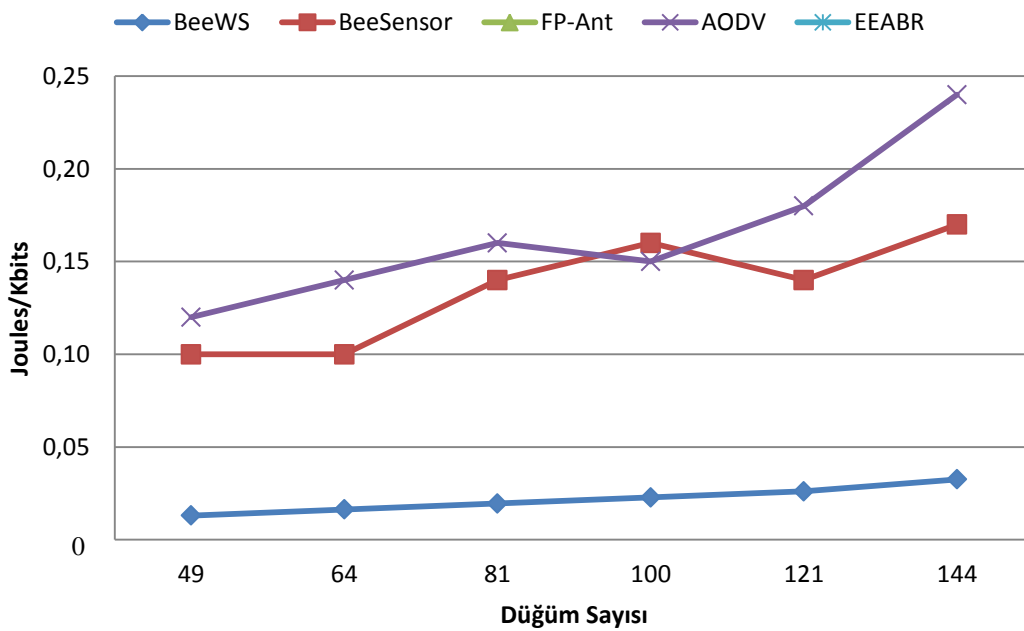
Şekil 6.2. Paket teslim oranları karşılaştırması



Şekil 6.3. Ortalama gecikme karşılaştırması

### 6.3.1.3. Enerji tüketimi

Şekil 6.4’de enerji tüketimi grafiği gösterilmektedir. Şekilde görüldüğü üzere AODV, BeeSensor ve BeeWS en iyi verimliliğe sahip olan protokollerdir. AODV enerji tüketimi açısından literatürde kabul görmüş bir protokoldür. BeeWS’ün en verimli olmasının nedeni AODV protokolünden ilham alıp bunu en verimli yol tekniği ile geliştirmesidir. Bunun yanında yüksek paket teslim oranına sahip olması enerji tüketimi açısından AODV ve BeeSensor’ün önüne geçmesine neden olmuştur. Sistem de kayıp oranı arttıkça enerji tüketimide düşmektedir. Grafikte FP-Ant ve EEABR yönlendirme protokollerinin gözükmeme nedeni yüksek enerji tüketmelerinden dolayı diğer üç protokolle aynı aralıkta olamamalarından kaynaklanmaktadır.



Şekil 6.4. Enerji tüketimi karşılaştırması

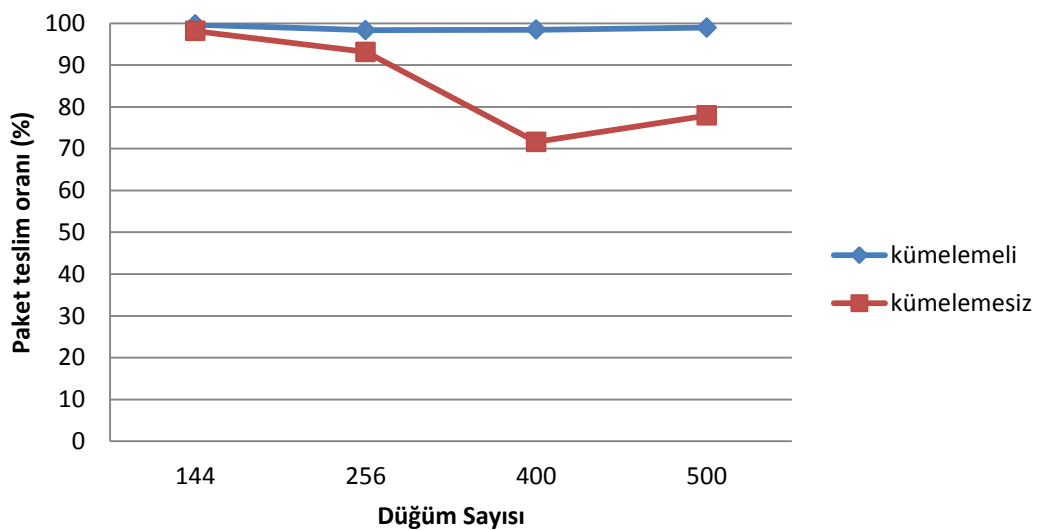
### 6.3.2. BeeWS için kümelemesiz ve kümelemeli uygulamaların karşılaştırılması

Daha önce de ifade edildiği gibi kümeleme yöntemi DEVS-Sensor ortamında modellendi. Yapılan bu çalışmayla büyük ölçeklere çıkıldıkça kümeleme yöntemine olan ihtiyacı ortaya koyduk ve etkilerini gözlemledik.

Deneyleerde, durağan (static) senaryoda 256, 400 ve 500 algılayıcı düğüm kullanarak KAA için rasgele topolojiler oluşturdu. Düğümler kümebaşı olarak kullanıldığında 105 m kapsam alanına sahipken üye düğüm olarak kullanıldığında 35 m kapsam alanına sahiptir. Başlangıç enerjisi tüm düğümler için 20 joule olarak belirlenmiştir. Olay boyutu 512 bit olarak kabul edilmiştir. Her saniyede bir olay olacak şekilde 300 saniye boyunca test yapıldı. Kümebaşıları çoklu-atlama yöntemini kullanarak hedef düğüme ulaşır. Değerlendirme amacıyla paket teslim oranı, ortalama gecikme ve enerji tüketimi ölçütleri kullanılmıştır.

### 6.3.2.1. Paket teslim oranı

Şekil 6.5’de paket teslim oranı grafiği görülmektedir. Paket teslim oranı kümelemesiz yönlendirmede algılayıcı düğüm sayısı arttıkça azalma eğilimine girdi. Bunun sebebi topolojide düğüm sayısının artması yol bulma yeteneğinin azalmasına neden olmasındır. Kümeleme yaklaşımında ise kümebaşıları kullanılarak topolojideki düğüm sayısı azaltıldı. Gelen olayların neredeyse tamamı hedefine ulaştırıldı. Kümelemesiz yöntem 500 düğümden daha fazla düğüme uygulanamamıştır. Benzetim aracı 400 düğümünden sonra kararsızlaştığından paket teslim oranı yüksek çıkmaktadır.

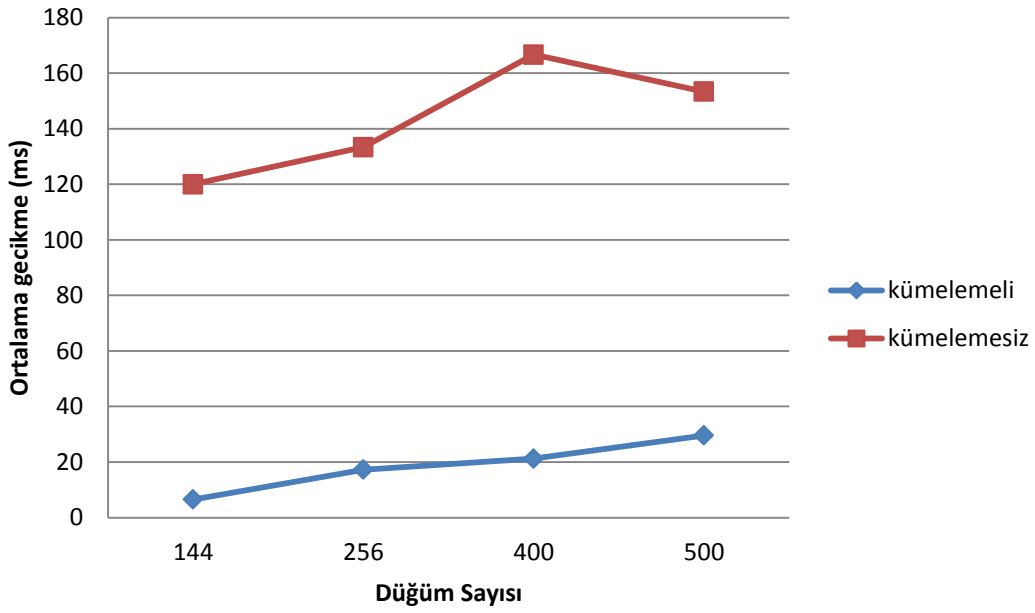


Şekil 6.5. Kümelemeli ve kümelemesiz yöntemlerin paket teslim oranları



### 6.3.2.2. Ortalama gecikme

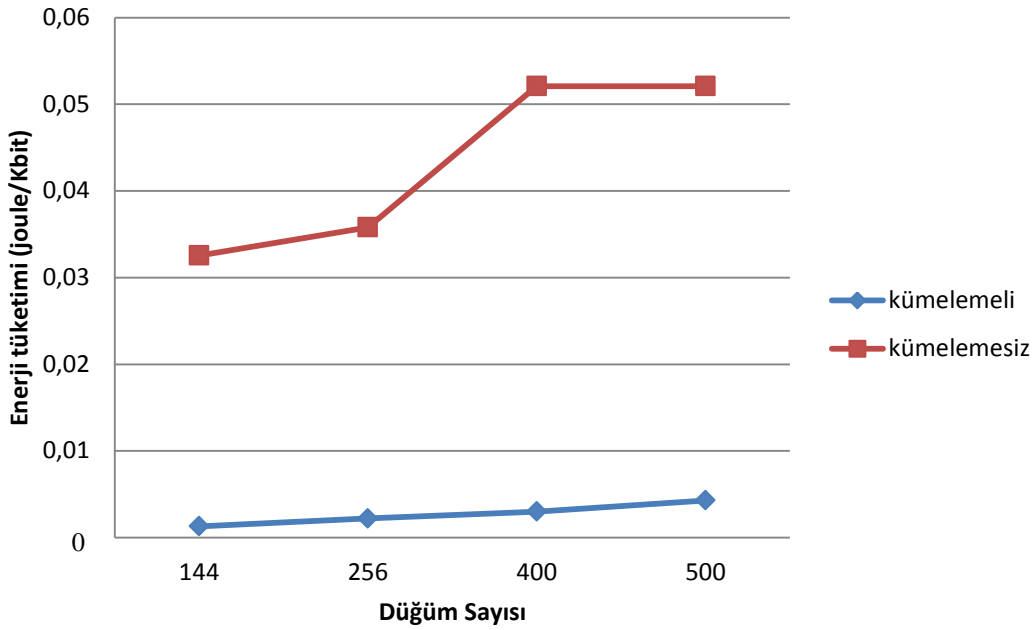
Düğüm sayısının artmasına bağlı olarak topolojide oluşan olay sayısı artmaktadır. Buna bağlı olarak da yüksek trafikten dolayı kümelemesiz protokolde ortalama gecikme artmaktadır. Kümeleme yaklaşımı kümebaşlarını kullanarak düğüm sayısını azalttığından dolayı ağdaki trafik düşmekte ve ortalama gecikmede azalmaktadır.



Şekil 6.6. Kümelemeli yöntemin ortalama gecikmeye etkisi

### 6.3.2.3. Enerji tüketimi

Şekil 6.7’de kümelemesiz ve kümelemeli yöntemlerinin enerji tüketimi grafikleri görülmektedir. Enerji tüketimi kümeleme yönteminin önemli kazanımlarındandır. Kümelemede tüm düğümler veri akışında rol almadığından sadece belli düğümlerin enerjileri kullanılmaktadır. Bu düğümler değiştirilerek enerjisi daha fazla olan düğümlerden kümebaşları seçilmektedir. Bu da enerjinin kontrollü bir şekilde kullanılmasını sağlar. Grafikte de kümelemesiz protokolle kümeleme arasında enerji tüketimi açısından büyük fark olduğu görülmektedir.

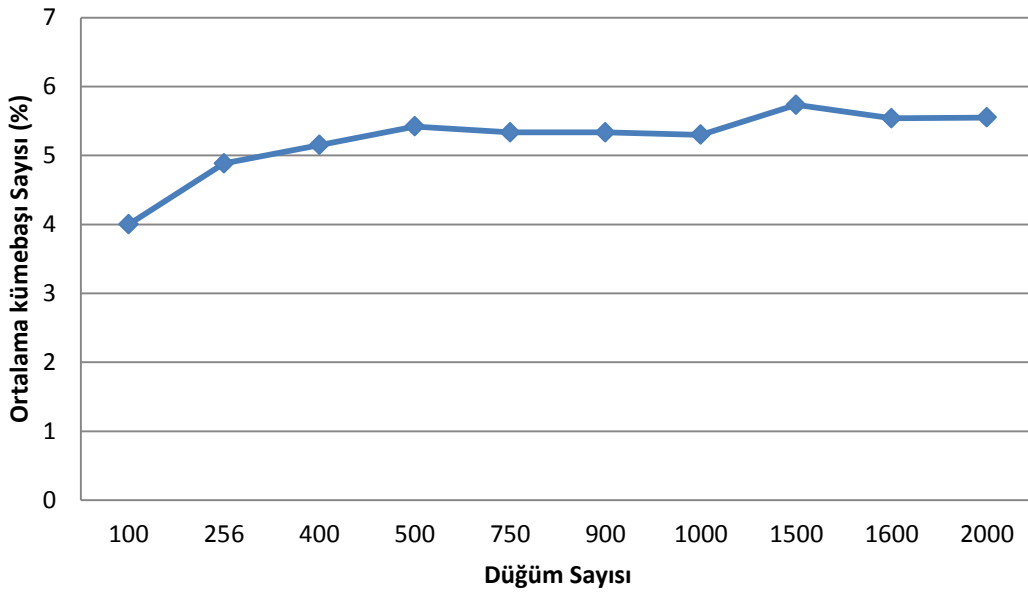


Şekil 6.7. Kümelemenin enerji tüketimine etkisi

### 6.3.3. Büyük ölçekli KAA'larda kümeleme

Büyük ölçeklere çıkabilmek için kümeleme yöntemi en çok kullanılan yöntemlerdendir. Kümeleme yöntemlerinde kullanılan ölçütlerden en önemlisi küme sayısıdır. Çünkü kümeleme yöntemlerinde trafiğin çoğunluğu kümebaşları tarafından yönlendirilir. Buda enerji kaybının büyük miktarda kümebaşlarında meydana geldiğini gösterir. Bundan dolayı küme sayısı ne kadar az olursa enerji kaybı da o kadar az olur.

Deneylerde oluşan küme sayısı ve ölçeklenebilen düğüm sayısına odaklanıldı. Şekil 6.8'de görüldüğü üzere kümeleme yöntemi kullanarak 2000 düğüme kadar ulaşıldı. Oluşan kümebaşı sayısına bakıldığında %5 civarında bir kümebaşı oluşumu gözlenmektedir. LEACH [54] kümeleme yöntemi literatürde en fazla atıf alan bir yöntem olarak %5 kümebaşını kümeleme yaparken standart olarak kabul etmiştir. Bu da geliştirilen yöntemin uygunluğunu göstermektedir.

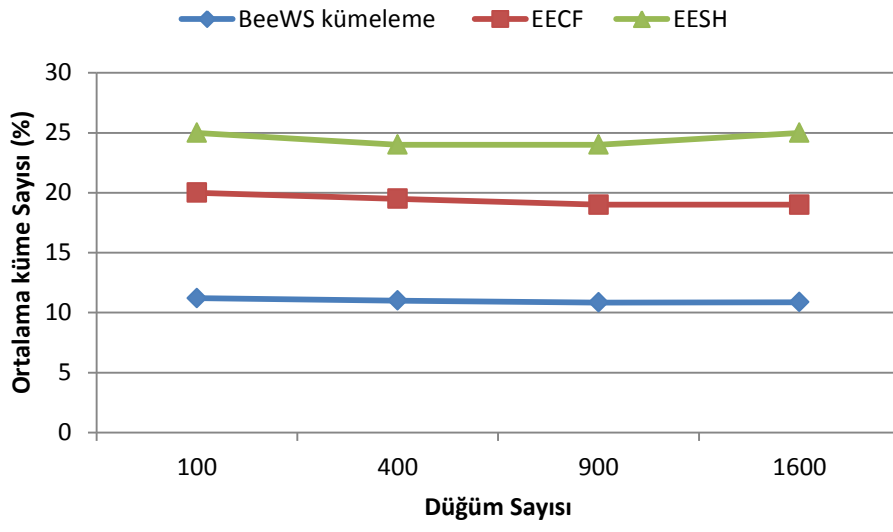


Şekil 6.8. Ortalama kümebaşı sayısı değişimi

### 6.3.3.1. BeeWS ve diğer kümeleme yöntemlerinin karşılaştırılması

Geliştirilen kümeleme yaklaşımı çoklu-atlamalı kümeleme yöntemler olan EECF [82] ve EESH [83] algoritmalarıyla karşılaştırılmıştır. Kümeleme 100m x 100m bir alanda yapıldı. Başlangıç enerjisi 2 joule olarak belirlenmiştir. Düğümlerin kapsam alanı 10 m' dir. Kümebaşı ve üye düğümlerin kapsam alanı aynıdır. Değerlendirmede kümebaşı sayısı dikkate alınmıştır. Deney alanı değiştirilmeden algılayıcı düğüm sayısı artırılmıştır.

Şekil 6.9'da görüldüğü gibi BeeWS kümeleme yöntemi literatürdeki EECF ve EESH kümeleme yöntemlerinden daha az küme üretmiştir. Yukarıda da anlatıldığı gibi daha az küme sayısı daha az enerji harcanmasına sebep olmaktadır.



Şekil 6.9. Ortalama küme sayısı karşılaştırması

## BÖLÜM 7. SONUÇLAR ve DEĞERLENDİRME

### 7.1. Sonuçlar

Kablosuz algılayıcılar tek başlarına bir şey ifade etmemelerine rağmen birbirleriyle organize olduklarında çevreden veri toplayıp hedeflerine ulaştırabilmektedirler. Bir arının tek başına bir şey ifade etmeyip arıların bir araya gelerek kovan oluşturup bal yapmaları örneği, kablosuz algılayıcılarla sosyal canlıların benzerliğini ortaya koymaktadır. Örnekte de görüldüğü gibi her bir sosyal canlı veya kablosuz algılayıcı sistemden bağımsız olarak karar verebilir. Aynı zamanda sistem ile uyumlu çalışabilir. Bu ayrık olaylı yaklaşımın KAA'ları modellemeye ne kadar uygun olduğunu göstermektedir.

Tez çalışmasında, oğul zekasının yönlendirme için ve ayrık olaylı benzetim yaklaşımının modelleme ve benzetim ortamı gelişiminde kullanılmasıyla verimli sonuçlar alındı. Bu sonuçlar Bölüm 6'da gösterilmektedir. Sonuçlar incelendiğinde aşağıdaki çıkarımlar yapılabilir;

- DEVS ortamında geliştirilen ve DEVS-Suite benzetim ortamından türetilen DEVS-Sensor benzeticisi büyük ölçekli KAA'ları modellemede kullanılacak ve verimli sonuçlar alınabilecek bir ortamdır.
- DEVS-Sensor ortamı java dilinin esnek nesne yönelimli yapısını kullandığından dolayı java sanal makinesi olan her ortamda çalışabilmektedir. Bu esnek yapıya BRITE topoloji üreticinin eklenmesiyle kolayca KAA için rasgele topolojiler üretilebilmekte; deneyler ve sonuçlar büyük ölçeklerde dahi kısa zamanda gerçekleştirilebilmektedir.

- DEVS-Sensor ortamı odaklanılan çalışma ile ilgili tüm detayları herhangi bir eklentiye bağlı kalmadan değiştirebilme ve sistemde olmayan özellik yada eklentileri yeniden oluşturabilme imkanı sunar. Açık kaynak koda sahip olduğundan geliştirilmeye açıktır.
- DEVS-Sensor benzetim ortamını kullanarak KAA'lara uygulanan BeeWS yönlendirme protokolü veri toplama problemine çözüm getirmektedir. Bu açıdan bakıldığında BeeWS klasik yönlendirme algoritmalarına göre daha yüksek paket teslim oranına sahiptir.
- KAA'lara uygulanan BeeWS yönlendirme protokolü Bölüm 6'daki grafiklerde de görüldüğü gibi diğer protokollere göre ortalama gecikmeyi azaltmaktadır.
- BeeWS protokolü enerji korunumu olarak diğer protokollerin önüne geçmektedir. Geliştirilen yöntemin yol bulma teknikleri ve yolları bir müddet saklaması bu alanda etkili oldu.
- BeeWS algoritması kümeleme yöntemiyle genişletilerek büyük ölçeklere ulaşıldı. Bu anlamda kümeleme yönteminin kullanılmasının katkı sağladığı ve beklenen sonuçlar verdiği görülmektedir.
- BeeWS kümeleme yönteminin küme sayısı bakımından çoklu-atlamalı yöntemlere üstünlük sağladığı görülmektedir.

Sonuç olarak; DEVS-Sensor ortamı esnek yapısıyla, kolay kullanımıyla ve büyük ölçekli ağlar oluşturup bu uygulamalardan sonuç alınmasıyla KAA'ların modelleme ve benzetiminde üstünlüğünü ortaya koymaktadır. Geliştirilen BeeWS kümelemesiz yönlendirme protokolüde veri toplama ve enerji problemlerine uygun çözümler getirmektedir. BeeWS kümeleme protokolüde büyük ölçeklere çıkıp elde ettiği düşük sayıda kümebaşı ile kümeleme yöntemlerinin küme sayısı problemine çözüm getirmektedir.

## 7.2. Tartışma ve Öneriler

DEVS-Sensor modelleme ve benzetim ortamı, KAA'ların yönetimi ve karakteristikleri ile ilgili konuları çalışmaya uygundur. DEVS-Sensor ortamında tasarlanan sistemler için belli kabuller ortaya kondu ve soyutlama işlemi buna göre gerçekleştirildi. DEVS yönteminin ve java dilinin avantajlarını kullanan DEVS-Sensor modelleme ve benzetim ortamı büyük ölçekli KAA'ları modelleyebilmektedir.

KAA için geliştirilen oğul zekası tabanlı yönlendirme algoritması BeeWS'de KAA'lardaki veri toplama, enerji problemlerine çözüm getirmektedir. BeeWS kümeleme protokolünde KAA'larda büyük ölçeklere çıkılmasına ve kümelemedeki kümebaşı problemine çözüm getirmektedir. BeeWS protokolünün verimli çalışmasının başlıca nedeni KAA'ların sosyal canlılarla benzerliğidir. Kümelemede olduğu gibi kendilerine yaşam kümeleri oluşturarak sayılarını artırabilmeleri ve kendi kendilerine organize olabilmeleri bu benzerlikleri ortaya koymaktadır. Sosyal canlılardan ilham alınarak, belli problemlere çözüm getirebilen bir yönlendirme protokolü geliştirildi.

KAA için geliştirilen protokolün DEVS-Sensor ortamına uygulanması oğul zekası tabanlı geliştirilen yöntem ve tekniklerin ayrık olaylı modelleme ve benzetim (DEVS) ortamlarında gerçekleştirildiğinde alınan neticelere daha kolay, hızlı ve esnek bir şekilde erişilebildiğini göstermektedir. Bunun nedeni sosyal canlılarda olduğu gibi ayrık olaylı modelleme ve benzetim (DEVS) ortamı dağıtık ve paralel çalışmaktadır. Bu da gerçekleştirilen çalışmanın daha verimli sonuçlar vermesine neden olmaktadır.

Geliştirilen yönlendirme protokolü ve benzetim aracı nihai bir hedef değil literatürde bir basamaktır. Hedefe giden yolda daha detaylı modeller oluşturan ve daha verimli çalışan modelleme ve benzetim ortamları oluşturmak mümkündür. Yapılan çalışma açık kaynak kodlu olduğundan her türlü bilimsel katkıya açık ve geliştirilebilir bir özelliğe sahiptir.

## KAYNAKLAR

- [1] HANDE, A., POLK, T., WALKER, W., BHATIA, D., Self-Powered Wireless Sensor Networks for Remote Patient Monitoring in Hospitals, Molecular Diversity Preservation International, 10.3390/s6091102, Switzerland, 22 September 2006.
- [2] AKYILDIZ, I. F., SU, W., SANKARASUBRAMANIAM, Y., CAYIRCI, E., Wireless Sensor Networks: A Survey, Elsevier Computer Networks, 38 (2002) 393–422, USA, 2002.
- [3] RAHMAN, M.A., PAKŠTAS, A., WANG, F. Z., Network Modelling and Simulation Tools, Simulation Modelling Practice and Theory, International Journal of the Federation of European Simulation Societies – EUROSIM, 17, pp. 1011-1031, USA, 2009.
- [4] VANGHELUWE, H., Multi-Formalism Modelling and Simulation, Doktora tezi, Universiteit Gent Faculteit Wetenschappen, Belgium, 2001.
- [5] ANTOINE-SANTONI, T., SANTUCCI, J.F., DE GENTILI, E., COSTA, B., Discrete Event Modeling and Simulation of Wireless Sensor Network Performance, Simulation, volume: 84, issue: 2-3, pp. 103-121, USA, 2008.
- [6] ZEIGLER, B.P., PRAEHOFER, H., KIM, T.G., Theory of Modelling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems, Academic Press, second edition, USA, 2000.
- [7] ZENG, X., BAGRODIA, R., GERLA, M., GloMoSim: a Library for Parallel Large-scale Wireless Networks, Proceedings of the 12th Workshop on Parallel Simulations -- PADS '98, pages: 154 – 161, Banff, Alberta, Canada, 1998.
- [8] Parallel/Distribute NS, <http://www.cc.gatech.edu/computing/compass/pdns/>, Haziran 2012.
- [9] CAMILO T., CARRETO, C., An Energy-Efficient Ant-Based Routing Algorithm for Wireless Sensor Networks, 5th International Workshop on Ant Colony Optimization and Swarm Intelligence, volume: 4150, pages: 49-59, Belgium, 2006.
- [10] ZHANG, Y., KUHN, L. D., FROMHERZ, M. P. J., Improvements on Ant Routing for Sensor Networks, 4th International Workshop, ANTS 2004, volume 3172, pp 154-165, Belgium, 2004.



- [11] WANG X., LI, Q., XIONG, N., PAN, Y., Ant Colony Optimization-Based Location-Aware, 3rd International Conference on Wireless Algorithms, Systems and Applications, volume: 5258, pages: 109-120, USA, 2008.
- [12] SELVAKENNEDY, S., SINNAPAN, S., SHANG, Y. T-ANT: A Nature-Inspired Data Gathering Protocol for Wireless Sensor Networks, Journal of Communications, Vol. 1, no. 2 (May 2006), pp. 22-29, Australia, May 2006.
- [13] DING, N., LIU, P. X., A Centralized Approach to Energy-Efficient Protocols for Wireless Sensor Networks, 2005 IEEE International Conference on Mechatronics and Automations, pages: 121-124, Canada, 2005.
- [14] SUN, H., JIANG, J., TAN, X., Queen-Ant-Aware-Based Algorithm for Wireless Sensor Networks Routing, 2006 IEEE International Conference on Information Acquisition, Vols 1 and 2, Conference Proceedings Pages: 622-626, China, 2006.
- [15] JUAN, L., CHEN, S., CHAO, Z., "Ant System based Anycast Routing in Wireless Sensor Networks", in the proceedings of International Conference on Wireless Communications, Networking and Mobile Computing (WiCom 2007), Sept. pp. 2420-2423, 2007.
- [16] LIAO, W. KAO, Y., FAN, C., Data aggregation in wireless sensor networks using ant colony algorithm, Journal Of Network And Computer Applications, volume: 31, issue: 4, pages: 387-401, February 2008.
- [17] WEN, Y., CHEN, Y., PAN, M., Adaptive ant-based routing in wireless sensor Networks using Energy Delay metrics, Journal Of Zhejiang University-Science, volume: 9, issue: 4, pages: 531-538, 8 March 2008.
- [18] SALEEM, K., FISAL, N., HAFIZAH, S., A Self-Optimized Multipath Routing Protocol, International Journal of Recent Trends in Engineering, , vol 2, no. 1, November 2009.
- [19] CHAO, W., QIANG, L., Swarm Intelligence Optimization Based Routing Algorithm For Wireless Sensor Networks, 2008 International Conference On Neural Networks And Signal Processing, vols 1 and 2, pages: 136-141, China, June 8~10, 2008.
- [20] AGHAEIL, R. G., RAHMAN, A., GUEAIEB, W., EL SADDIK, A., Ant Colony-Based Reinforcement Learning Algorithm for Routing in Wireless Sensor Networks, 2007 IEEE Instrumentation & Measurement Technology Conference, vols 1-5, pages: 2050-2055, Poland, 1-3 May, 2007.
- [21] ÖKDEM, S., KARABOĞA, D., Routing in Wireless Sensor Networks Using an Ant Colony Optimization (ACO) Router Chip, Sensors, volume: 9, issue: 2, pages: 909-921, 2009.

- [22] KARABOGA, D., AKAY, B., A survey: algorithms simulating bee swarm intelligence, *Artificial Intelligence Review*, volume: 31, issue: 1-4, pages: 61-85, 2009.
- [23] SALEEM, M., FAROOQ, M., Beesensor: A bee-inspired power aware routing protocol for wireless sensor networks. In *Proceedings of EvoWorkshops (EvoCOMNET)*, LNCS 4448, pgs 81-90, Apr 2007.
- [24] PAONE, M., PALADINA, L., SCARPA, M., PULIAFITO, A., A Multi-Sink Swarm-based Routing Protocol for Wireless Sensor Networks, 2009 IEEE Symposium On Computers And Communications, vols 1 and 2, pages: 28-33 Tunisia, Jul 05-08, 2009.
- [25] BAKER, D.J., EPHREMIDES, A., The architectural organization of a mobile radio network via a distributed algorithm, *IEEE Transactions on Communications*, COM-29 (11) 1694–1701, 1981.
- [26] BAKER, D.J., Ephremides, A., FLYNN, J.A., The design and simulation of a mobile radio network with distributed control, *IEEE Journal on Selected Areas in Communications* 226–237, 1984.
- [27] XU, K., GERLA, M., A heterogeneous routing protocol based on a new stable clustering scheme, *Proceeding of IEEE Military Communications Conference (MILCOM 2002)*, vols 1 and 2, pages: 838-843, Anaheim, CA, October 2002.
- [28] NAGPAL, R., COORE, D., An algorithm for group formation in an amorphous computer, *Proceedings of the 10th International Conference on Parallel and Distributed Systems (PDCS'98)*, pages: 452-455, Las Vegas, NV, October 1998.
- [29] BANERJEE, S., KHULLER, S., A clustering scheme for hierarchical control in multi-hop wireless networks, *Proceedings of 20th Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'01)*, volume: 2, pages: 1028 – 1037, Anchorage, AK, April 2001.
- [30] ZHANG, H., ARORA, A., GS3: scalable self-configuration and self-healing in wireless networks, *Proceedings of the 21st ACM Symposium on Principles of Distributed Computing (PODC 2002)*, volume: 43, issue: 4, pages: 459-480, Monterey, CA, July 2002.
- [31] MARIA, A., Introduction to Modeling and Simulation, *Proceedings of the 1997 Winter Simulation Conference*, pp. 7-13, USA, 1997.
- [32] HEINZELMAN, W.B. CHANDRAKASAN, A.P., BALAKRISHNAN, H. Application specific protocol architecture for wireless microsensor networks, *IEEE Transactions on Wireless Networking*, volume: 1, issue: 4, pages: 660-670, October 2002.

- [33] DEMIRBAS, M., ARORA, A., MITTAL, V., FLOC: a fast local clustering service for wireless sensor networks, Proceedings of Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks (DIWANS'04), pages: 700-709, San Jose, CA, June 2004.
- [34] TAYŞI, Z., Telsiz Algılayıcı Düğüm Tasarımı ve Gerçeklenmesi, Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı Yüksek Lisans Tezi, İstanbul, 2006.
- [35] KARASULU, B.; TOKER, L.; KORUKOĞLU, S., ZigBee - IEEE 802.15.4 Standartı Temelli Kablosuz Algılayıcı Ağları, XIV. Türkiye'de İnternet Konferansı - Inet-tr'09, sayfalar: 131-134 İstanbul, 2009.
- [36] CHANDRA, P., DOBKIN, D. M., BENSKY, A., OLEXA, R., LIDE, D. A., DOWLA, F., RF & Wireless Networking, Elsevier, s-510, USA, 2008.
- [37] ÖKDEM, S., KARABOĞA, D., Kablosuz Algılayıcı Ağlarda Yönlendirme Teknikleri, IX. Akademik Bilişim Konferansı, sayfalar: 4090-415 Kütahya, Şubat 2007.
- [38] DI CARO, G., DORİGO M., Ant colonies for adaptive routing in packet-switched communications networks. In Parallel Problem Solving from Nature – PPSN V, LNCS 1498, pages: 673–682, Amsterdam, Netherlands, Sept 1998.
- [39] AL-KARAKI, J.N., KAMAL, A.E., Routing Techniques in Wireless Sensor Networks: a Survey, Wireless Communications IEEE, Volume 11, pp. 6-28, 2004.
- [40] KRISHNAMACHARI, B., ESTRIN D., WICKER S., The Impact of Data Aggregation in Wireless Sensor Networks, International Workshop on Distributed Event-Based Systems, (DEBS '02), held in conjunction with IEEE ICDCS, pages: 575 – 578, Vienna, Austria, July 2002.
- [41] ROMER, K., FRIEDEMANN, M., The Design Space of Wireless Sensor Networks, IEEE Wireless Communications, pp. 54-61, Zurich, Switzerland December 2004.
- [42] VILLALBA, L., OROZCO, A., CABRERA, A., ABBAS, C., Routing Protocols in Wireless Sensor Networks, Sensors, pages: 8399-8421, October 2009.
- [43] RAJASHREE., V., BIRADAR, V.C., Classification And Comparison Of Routing Protocols In Wireless Sensor Networks, Ubiquitous Computing Security Systems, volume 4, 2009.
- [44] AKKAYA, K., YOUNIS, M., A survey on routing protocols for wireless sensor networks, Ad Hoc Networks Elsevier, volume 3, issue 3, pages 325–349, May 2005.

- [45] TILAK, A., taxonomy of wireless microsensor network models, *Mobile Computing and Communications Review*, volume: 6, pages: 28–36, 2002.
- [46] ÇELİK, F., ZENGİN A., TUNCEL, S., A survey on swarm intelligence based routing protocols in wireless sensor networks, *International Journal of the Physical Sciences* Vol. 5(14), pp. 2118-2126, 4 November, 2010.
- [47] BONABEAU, E., DORIGO, M., THERAULAZ, G., Inspiration for optimization from social insect behaviour. *Nature*, volume: 406, issue: 6791, pages: 39-42, July 2000.
- [48] DORIGO, M., DI CARO, G., GAMBARDELLA, L.M., Ant algorithms for discrete optimization, *Artificial Life*, volume: 5(2), pages: 137–172, 1999.
- [49] Network Simulator-2, <http://www.isi.edu/nsnam/ns/>, Haziran 2012.
- [50] ZHANG, Y., KUHN, L.D., FROMHERZ, M.P.J.: Improvements on ant routing for sensor networks. In: Dorigo, M., Birattari, M., Blum, C., Gambardella, L.M., Mondada, F., Stützle, T. (eds.) ANTS 2004. LNCS, vol. 3172, pp. 154–165. Springer, Heidelberg, 2004.
- [51] MASCOLO, C., MUSOLESI, M., SCAR: Context-aware Adaptive Routing in Delay Tolerant Mobile Sensor Networks, *IWCMC'06*, July 3–6, 2006.
- [52] Opnet Modeler, <http://www.opnet.com/products/modeler/home.html>, Haziran 2012.
- [53] SELVAKENNEDY, S., SINNAPPAN, S., "The Time-Controlled Clustering Algorithm for Optimized Data Dissemination in Wireless Sensor Networks," in *Proc. IEEE Conference on LCN*, pp. 509-510, Sydney, Australia, 2005.
- [54] KIM, S., SARJOUGHIAN H. S., ELAMVAZHUTHI, V., DEVS-Suite: A Simulator Supporting Visual Experimentation Design and Behavior Monitoring, *Spring Simulation Multiconference*, Article no 161, San Diego, California, 2009.
- [55] LINDSEY, S., RAGHAVENDRA, C., SIVALINGAM, K. M., "Data gathering algorithms in sensor networks using energy metrics," *IEEE Transactions on Parallel and Distributed Systems*, 13(9), 924-935, 2002.
- [56] AKKAYA, K. YOUNIS, M., A Survey on Routing Protocols for Wireless Sensor Networks, *Ad Hoc Networks*, volume: 3, issue: 3, pages: 325–349, May 2005.
- [57] <http://pcl.cs.ucla.edu/projects/glomosim/>, Haziran 2012.
- [58] INTANAGONWIWAT, C., GOVINDAN, R., Directed Diffusion for Wireless Sensor Networking, *Proceedings of the 1st ACM international workshop on wireless sensor networks and applications*, Page: 2, 2003.

- [59] <http://jist.ece.cornell.edu/>, Haziran 2012.
- [60] PERKINS, C., BELDING-ROYER, E., DAS, S., Ad hoc On-Demand Distance Vector (AODV) Routing, <http://tools.ietf.org/html/rfc3561>, 2010.
- [61] <http://www.omnetpp.org/>, Haziran 2012.
- [62] MAMALIS, B., GAVALAS, D., KONSTANTOPOULOS, C., Zhang/RFID and Sensor Networks AU7777\_C012 Page Proof Page 331, 2009.
- [63] ABBASI, A., YOUNIS, M., A survey on clustering algorithms for wireless sensor networks, Published by Elsevier B.V. doi:10.1016/j.com.2007.05.024, 21 June 2007.
- [64] ZEIGLER, B. P., PRAEHOFER, H., KIM, T., Theory of Modelling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems. Academic Press, second edition, 2000.
- [65] ZENGİN, A., Dağıtık Simülasyon Sistemleri İçin Yeni Bir Yönlendirme Algoritması ve Uygulaması, Doktora tezi, Sakarya, Temmuz 2004.
- [66] TUNCEL, S., Gezgin Ağ Yönlendirme Protokolleri İçin Devs Tabanlı Benzetim Aracı Tasarımı, Doktora tezi, Ekim 2010.
- [67] ZEIGLER, B.P., Theory of Modelling and Simulation, Malabar, Florida, 1984.
- [68] MALOWIDZKI, M., Network Simulators: A Developer's Perspective, Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS'04), pp. 412-419, July 2004.
- [69] ZEIGLER, B.P., MITTAL, S., Modeling and Simulation of Ultra-large Networks: A Framework for New Research Directions, supported by NSF Grant ANI-0135530, ULN Workshop, July 2002.
- [70] BEGG, L., LIU, W., PAWLIKOWSKI, K., PERERA, S., SIRISENA, H., Survey of Simulators of Next Generation Networks for Studying Service Availability and Resilience, Technical Report TR-COSC 05/06, University of Canterbury, Christchurch, New Zeland, 2006.
- [71] KIM, S., SARJOUGHIAN H. S., ELAMVAZHUTHI, V., DEVS-Suite: A Simulator Supporting Visual Experimentation Design and Behavior Monitoring, Spring Simulation Multiconference, Article no 161, San Diego, California, 2009.
- [72] SEELY, T., The Wisdom of the Hive, Harvard University Press, Cambridge, 1995.

- [73] HILD, D.R., Discrete Event System Specification (DEVS) Distributed Object Computing (DOC) Modeling And Simulation, Doktora Tezi, The University of Arizona, 2000.
- [74] REINHOLTZ, K., Java Will Be Faster Than C++, ACM SIGPLAN Notices, Volume 35, Issue 2, pp 25 – 28, 2000.
- [75] CASSEL, R. A., PIDD, M., Distributed Discrete Event Simulation Using the Three-Phase Approach and Java, Simulation Practice and Theory, 8, pp 491-507, 2001.
- [76] PIDD, M., CASSEL, R., Using Java to Develop Discrete Event Simulations, Journal of the Operational Research Society, 51, 2000.
- [77] NICOL, D. M., Scalability of Garbage Collection in Java-Based Discrete-Event Simulators, Conference on Computer Simulation, Cambridge England, 2003.
- [78] KULJIS, J., PAUL R.J., An Appraisal of Web-Based Simulation: Whither We Wander?, Simulation Practice and Theory, 9, pp37-54, 2001.
- [79] KILGORE, R. A., KLEINDORFER, G. B., The Future Of Java-Based Simulation, Winter Simulation Conference, pp 1707-1712, 1998.
- [80] [www.acims.arizona.edu/SOFTWARE/devsjava.../devsjava-user-ref.pdf](http://www.acims.arizona.edu/SOFTWARE/devsjava.../devsjava-user-ref.pdf), Haziran 2012.
- [81] CEKEN, C., “An Energy Efficient and Delay Sensitive Centralized MAC Protocol for Wireless Sensor Networks”, Computer Standards & Interfaces, 30.1-2, 20-31, 2008.
- [82] RAHMAN, M. A., PAKŠTAS A., WANG F.Z., Network Topology Generation and Discovery Tools, Simulation Modelling Practice and Theory, Vol 17 Issue 6, pp 1011-1031, July 2009.
- [83] MEDINA, A., LAKHINA, A.I, MATTA, I., BYERS, J., BRITE: Universal Topology Generation from a User's Perspective, Technical Report, Boston University, Boston, MA, USA, 2001.
- [84] ANDERSON, C., The adaptive value of inactive foragers and the scout-recruit system in honey bee *apis mellifera* colonies, Behavioral Ecology 111–119, 2001.
- [85] SALEEM, M., ULLAH, I., FAROOQ, M., BeeSensor: An energy-efficient and scalable routing protocol for wireless sensor networks, Elsevier, doi:10.1016/j.ins., 2012.

## EKLER

### Ek A. DEVS-Sensor Dügüm Atomik Model Tanımı

//DEVS-Sensor düğüm atomik model tanımı

$M_{\text{BeeWS\_node}} = \langle X, Y, S, \delta_{\text{ext}}, \delta_{\text{int}}, \delta_{\text{con}}, \lambda, ta \rangle$

where;

//giriş portları ve değerleri

$X = \text{inport} \times \text{invalues}$

invalues : {DATA, HELLO, IIZC, GIZC, mesaj },

inports : {NIC-in, inEvent, inTopo},

//çıkış portları ve değerleri

$Y = \text{outport} \times \text{outvalues}$

outvalues : {DATA, HELLO, IIZC, GIZC, mesaj },

outports : {NIC-out, outEvent, outTopo},

//durum kümeleri

$S = \text{phase} \times \sigma \times Q$

phase : {"boşda", "kurulum", "hello\_alındı", "Komşu eklendi", "data\_alındı",  
"veri\_alındı", "data\_gönder", "data\_al", "IIZC\_alındı", "GIZC\_alındı",

“IIZC\_gönder”, “GIZC\_gönder”, “mesaj\_alındı”, “data\_güncellendi”,  
 “data\_forward”, “enerji\_gönder”, “BEKLEME” }

$$\sigma = \mathfrak{R}^+_{0,\infty}$$

Q = queue × yedekqueue

Burada, kuyruk gelen ve gönderilen paketleri ve iletilecek veri paketlerini tutar.

// Harici geçiş fonksiyonu

$\delta_{\text{ext}}((\text{phase}, \sigma, Q), e, X) =$

```

if packet=hello and queue has enough space
  enqueue the packet and s ← ("hello_alındı", σ', x)
else s ← ("congested", σ', x)

if packet = IIZC
  enqueue the packet and s ← ("IIZC_alındı ", σ', x)
else s ← ("congested", σ', x)

if packet = GIZC
  enqueue the packet and s ← ("GIZC_alındı ", σ', x)
else s ← ("congested", σ', x)

if packet = DATA
  enqueue the packet and s ← ("data_alındı" , σ', x)
else s ← ("congested", σ', x)

if packet = mesaj
  enqueue the packet and s ← ("data_alındı" , σ', x)
else s ← ("mesaj_alındı", σ', x)

```



//Dahili geiş fonksiyonu

$\delta_{int}(\text{phase}, \sigma, Q) =$

```

if queue size =0 and deęer = 0 s ← ("bořda",σ',x );
if queue size =0 and deęer ≠ null s ← ("data_forward",σ',x );
else dequeue packet;
if packet=hello s ← ("komřu_eklendi",σ',x );
else if ttl >=15 s ← ("data_forward ",σ',x );
else if packet=data s ← ("komřu_eklendi",σ',x );
if komřu=hedef s ← ("veri_alındı",σ',x );
else if komřu≠ hedef s ← ("data_gönder",σ',x);
else if its route = routing table s ← ("data_al",σ',x );
else s ← ("IIZC_gönder",σ',x );
else if packet= IIZC and komřu = hedef s ←
("GIZC_gönder",σ',x );
else s ← ("IIZC_gönder" , σ',x );
else if packet= GIZC and hedef = kaynak s ←
("BEKLEME",σ',x );
else s ← ("GIZC_gönder" , σ',x );
else if düęüm≠ kümebařı s ← ("data_güncellendi",σ',x);
where s ∈ S

```

// Confluent geiş fonksiyonu

$\delta_{con}((\text{phase}, \sigma, Q), e, X) = \delta_{ext}(\delta_{int}(\text{phase}, \sigma, Q), 0, X)$

//Çıkış fonksiyonu

$\lambda(\text{phase}, \sigma, Q) =$

```

if enerji=1 or 2000 s ← ("enerji_gönder" ,  $\sigma$ , x );
when phase ="initialize" generate HELLO and output
y← (NIC-out,hello_olustur);
when phase ="enerji_gönder" y← (outTopo,enerji_gönder());
when phase ="IIZC_gönder" y← (NIC-out,paketim);
when phase ="IIZC_alındı" y← (NIC-out,paketim);
when phase ="GIZC_gönder" y← (NIC-out,paketim);
when phase ="data_forward" y← (NIC-out,paketim);
when phase ="data_al" y← (NIC-out,paketim);
when phase ="data_gönder" y← (NIC-out,paketim);
when phase ="data_güncellendi" y← (NIC-out,paketim);
when phase ="veri_alındı" y← (NIC-out,paketim);

where y ∈ Y

```

//Zaman ilerleme fonksiyonu

$\tau_a(s) = \sigma$

## ÖZGEÇMİŞ

Fatih ÇELİK, 1980 Karasu / SAKARYA doğumludur. İlk ve orta öğretimini Adapazarı'nda tamamladı. Lise öğrenimini Adapazarı Ali Dilmen Lisesi'nde tamamladı. 1997 yılında girdiği Gazi Üniversitesi Teknik Eğitim Fakültesi Elektronik ve Bilgisayar Eğitimi Bölümü Bilgisayar Sistemleri Öğretmenliği Programından 2001 yılında mezun oldu. Aynı yıl Sapanca Çok Programlı Lisesi'nde Bilgisayar Öğretmeni olarak göreve başladı. 2002 yılında başladığı yüksek lisans eğitimini "Devre Analizi Dersinin Uzaktan Eğitime İçeriği Geliştirilerek Uyarlanması" isimli yüksek lisans tez çalışması ile 2006 yılında bitirerek bilim uzmanı unvanını aldı. 2003 yılından itibaren görevine Sakarya Fatih Endüstri Meslek Lisesi'nde devam etti. Üniversite destekli, ulusal ve Avrupa Birliği projelerinde görev aldı. Sakarya Fatih Endüstri Meslek Lisesi'nde görevine devam etmektedir. Evli ve iki çocuk babasıdır.