

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

**KABLOSUZ ALGILAYICI AĞLAR İÇİN HİBRİT  
KAYIP TELAFİ MEKANİZMASINA DAYANAN YENİ  
BİR TAŞIMA PROTOKOLÜ TASARIMI**

**DOKTORA TEZİ**

**Ayhan KİRAZ**

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

**Tez Danışmanı : Doç. Dr. Murat ÇAKIROĞLU**

**Kasım 2013**

T.C.  
SAKARYA ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

**KABLOSUZ ALGILAYICI AĞLAR İÇİN HİBRİT  
KAYIP TELAFİ MEKANİZMASINA DAYANAN YENİ  
BİR TAŞIMA PROTOKOLÜ TASARIMI**

**DOKTORA TEZİ**

**Ayhan KİRAZ**

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

**Bu tez 29/11/2013 tarihinde aşağıdaki jüri tarafından Oybirliği ile kabul edilmiştir.**

**Doç. Dr.  
Murat ÇAKIROĞLU  
Jüri Başkanı**

**Doç. Dr.  
Ahmet ZENGİN  
Üye**

**Yrd. Doç. Dr.  
Murat İSKEFİYELİ  
Üye**

**Prof. Dr.  
Hüseyin EKİZ  
Üye**

**Doç. Dr.  
Resul KARA  
Üye**

## **TEŐEKKÜR**

Eđitimim süresince olduđu gibi bu doktora alıőmasında da bana destek olan, hocam ve danıőmanım Do. Dr. Murat AKIROĐLU'na teőekkür ederim. alıőmam boyunca desteklerini esirgemeyen Elektronik-Bilgisayar Eđitimi bölümündeki arkadaşlarıma teőekkür ederim. Bunun yanında alıőmam boyunca maddi ve manevi desteklerini aldıđım, moral motivasyonlarını her zaman yanımda bulduđum eőime ve aileme teőekkür ederim.

# İÇİNDEKİLER

TEŞEKKÜR.....	ii
İÇİNDEKİLER .....	iii
SİMGELER VE KISALTMALAR LİSTESİ.....	vii
ŞEKİLLER LİSTESİ .....	x
TABLOLAR LİSTESİ.....	xiii
ÖZET.....	xiv
SUMMARY.....	xv
BÖLÜM 1.	
GİRİŞ.....	1
1.1. Literatür Özeti .....	2
1.2. Tezin Amacı, İzlenen Çalışma Yöntemi ve Katkıları .....	4
1.3. Tez Düzeni.....	6
BÖLÜM 2.	
KABLOSUZ ALGILAYICI AĞLAR.....	8
2.1. Giriş.....	8
2.2. Algılayıcı Düğümlerin Tarihçesi.....	9
2.3. Kablosuz Algılayıcı Ağ Uygulama Alanları.....	10
2.3.1. Askeri alanlar .....	11
2.3.2. Tıbbi alanlar.....	11
2.3.3. Çevresel alanlar.....	12
2.3.4. Ev otomasyon alanları .....	12
2.3.4. Ticari alanlar .....	13
2.4. Kablosuz Algılayıcı Düğüm Yapısı.....	13
2.5. Kablosuz Algılayıcı Ağ Mimarisi .....	16

2.5.1. Fiziksel katman .....	17
2.5.2. Veri bağı katmanı .....	18
2.5.2.1. Çizelge tabanlı ortam erişim protokolleri.....	18
2.5.2.2. Çarpışmasız ortam erişim protokolleri.....	20
2.5.2.3. Çekişme tabanlı ortam erişim protokolleri.....	21
2.5.3. Yönlendirme katmanı .....	24
2.5.4. Taşıma katmanı .....	26
2.5.5. Uygulama katmanı .....	27
2.6. Kablosuz Algılayıcı Ağ Tasarımını Etkileyen Faktörler.....	27
2.7. Sonuçlar .....	31

### BÖLÜM 3.

TAŞIMA KATMANI .....	33
3.1. Giriş .....	33
3.2. KAA'larda Taşıma Katman Protokolünün Görevi .....	33
3.3. KAA'lardaki Ulaşım Katmanı Protokolleri Tasarım Ölçütleri .....	35
3.3.1. Güvenilirlik.....	35
3.3.2. Servis kalitesi.....	36
3.3.3. Adillik.....	36
3.3.4. Enerji verimliliği.....	36
3.4. KAA'lar için Önerilen Taşıma Katman Protokolleri.....	37
3.4.1. Tıkanıklık kontrolü protokolleri.....	38
3.4.1.1. Fusion.....	38
3.4.1.2. CODA .....	39
3.4.1.3. CCF .....	39
3.4.1.4. PCCP.....	39
3.4.1.5. ARC.....	40
3.4.1.6. Siphon .....	40
3.4.1.7. Trickle .....	41
3.4.1.8. PHTCPP.....	42
3.4.1.9. SenTCP.....	42
3.4.1.10. HCCP.....	43
3.4.2. Güvenilirlik protokolleri.....	43

3.4.2.1. DTSN .....	44
3.4.2.2. DTC.....	44
3.4.2.3. RMST.....	45
3.4.2.4. RBC.....	46
3.4.2.5. GARUDA.....	46
3.4.2.6. PSFQ.....	47
3.4.2.7. ERTF .....	48
3.4.3. KAA’larda hem tıkanıklığı hem de güvenilirliği destekleyen protokoller .....	48
3.4.3.1. STCP .....	50
3.4.3.2. ESRT .....	50
3.4.3.3. TRCCIT .....	51
3.4.3.4. CRRT .....	52
3.4.3.5. RT <sup>2</sup> .....	52
3.4.3.6. ART .....	53
3.4.3.7. RCRT .....	53
3.4.3.8. Flush .....	53
3.4.3.9. PORT .....	54
3.4.3.10. SRCP .....	54
3.5. Sonular.....	55

#### BÖLÜM 4.

HİBRİT KAYIP TELAFİ MEKANİZMASINA DAYANAN YENİ BİR TAŞIMA PROTOKOLÜ.....	56
4.1. Giriş.....	56
4.2. Motivasyon.....	56
4.3. Hibrit Kayıp Telafi Mekanizmasına Dayanan Taşıma (HİKMET) Protokolü.....	58
4.3.1. Ortam koşullarının tespiti için eşik değerlerinin..... belirlenmesi.....	58
4.3.2. Kayıp telafi mekanizmasının adaptif olarak değiştirilmesi .....	61
4.3.3. Uçtan-uca paket gönderimi .....	62
4.3.3.1. Ara NACK gönderimi.....	64

4.3.4. Düğüm-den-düğüm-e kayıp telafi mekanizması.....	68	
4.3.4.1. Segment gönderme işlemi.....	68	
4.3.4.2. Sırasız segment gönderme işlemi.....	69	
4.3.4.3. Adaptif hata kurtarma yapısı.....	69	
4.3.4.4. Adaptif proaktif kayıp tespiti ve telafisi.....	70	
4.3.4.5. Kaynak düğüm-e bilgi mesajı gönderilmesi.....	72	
4.5. Sonuçlar.....	74	
BÖLÜM 5.		
HİKMET PROTOKOLÜNÜN BAŞARIM DEĞERLENDİRMESİ.....		75
5.1. Giriş.....	75	
5.2. Benzetim kriterleri.....	75	
5.3. Paket Başarım Oranı.....	76	
5.4. Uçtan Uca Gecikme.....	79	
5.5. Enerji Tüketimi.....	82	
5.6. Paket Maliyeti.....	85	
5.7. Sonuçlar.....	89	
BÖLÜM 6.		
SONUÇLAR VE DEĞERLENDİRME .....		90
6.1. Sonuçlar.....	90	
6.2. Tartışma ve Öneriler.....	92	
KAYNAKLAR.....		93
EKLER.....		103
ÖZGEÇMİŞ.....		129

## SİMGELER VE KISALTMALAR LİSTESİ

ACK	: Acknowledgement
ADC	: Analog-Digital Convertor
AIMD	: Additive Increase Multiplicative Decrease
AQM	: Active Queue Management
ART	: Asymmetric and Reliable Transport
B-MAC	: Berkeley Media Access Control
CCF	: Congestion Control and Fairness
CDMA	: Code Division Multiple Access
CN	: Congestion Notification
CODA	: Congestion Detection and Avoidance
CRRT	: Congestion-Aware and Rate-Controlled Reliable
CSMA	: Carrier Sense Multiple Access
CTS	: Clear to Send
DCF	: Distributed Coordinator Function
D-MAC	: Directional Media Access Control
DSN	: Distributed Sensor Network
DTC	: Distributed TCP Caching
ESRT	: Event-to-Sink Reliable Transport
FDMA	: Frequency Division Multiple Access
FHSS	: Frequency-Hopping Spread Spectrum
GloMoSim	: Global Mobile Information System Simulator
GPS	: Global Position System
HACK	: Hybrid Acknowledgement
HCCP	: Hybrid Congestion Control Protocol
IACK	: Implicit Acknowledgement
ID	: Identification



IEEE	: Institute of Electrical and Electronics Engineers
ISM	: Industrial, Scientific, Medical
JNS	: Java Network Simulator
KAA	: Kablosuz Algılayıcı Ağlar
KAD	: Kablosuz Algılayıcı Düğümler
LLC	: Logical Link Control
LMAC	: A Lightweight Medium Access
MAC	: Medium Access Control
MIT	: Massachusetts Institute of Technology
NACK	: Negative Acknowledgement
NASA	: National Aeronautics and Space Administration
NAV	: Network Allocation Vector
NS-2	: Network Simulator 2
NS-3	: Network Simulator 3
OMNET++	: Objective Modular Network Test-bed in C++
PCCP	: Priority-based Congestion Control
PHTCCP	: Prioritized Heterogeneous Traffic Prioritized Heterogeneous Traffic
PORT	: Price-Oriented Reliable Transport
PSFQ	: Pump-Slowly, Fetch-Quickly
QoS	: Quality of Service
RAP	: Real-Time Communication Architecture Protocol
RBC	: Reliable Bursty Convergecast
RCRT	: Rate-Controlled Reliable Transport
RF	: Radio Frequency
RMST	: Reliable Multi-Segment Transport
RT <sup>2</sup>	: Real-Time and Reliable
RTS	: Request to Send
RTT	: Round Trip Time
RX	: Received Data
SMAC	: Sensor Medium Access Control
SOSUS	: Sound Surveillance System
SRCP	: Sensor Reliability and Congestion Control Protocol

STCP	: Sensor Transmission Control Protocol
TCP	: Transmission Control Protocol
TDMA	: Time Division Multiple Access
TinyOS	: Tiny Operating System
T-MAC	: Timeout MAC
TRCCIT	: Tunable Reliability with Congestion Control for Information Transport
TX	: Transmit Data
UDP	: User Datagram Protocol
VHDL	: VHSIC hardware description language
VS	: Virtual Sink
WFQ	: Weighted Fair Queuing

## ŞEKİLLER LİSTESİ

Şekil 1.1.	CAA'lar için önerilen taşıma katman protokolleri.....	3
Şekil 2.1.	Kablosuz algılayıcı ağ örneği .....	9
Şekil 2.2.	MIT tarafından geliştirilen örnek algılayıcı ağ sistemi.....	10
Şekil 2.3.	Kablosuz algılayıcı ağ uygulama alanları.....	11
Şekil 2.4.	Askeri bir uygulamada kullanılan algılayıcı ağı.....	11
Şekil 2.5.	Kablosuz algılayıcı ağ düğüm mimarisi .....	15
Şekil 2.6.	Kablosuz algılayıcı ağ mimarisi .....	16
Şekil 2.7.	Ortam erişim protokol ailesi .....	19
Şekil 2.8.	TDMA yönteminin yapısı.....	19
Şekil 2.9.	CSMA protokollerinde ortaya çıkan gizli düğüm problemi.....	22
Şekil 2.10.	IEEE 802.11 ortam erişim fonksiyonu .....	23
Şekil 2.11.	S-MAC protokolünün görev çevrimi .....	24
Şekil 2.12.	Kablosuz algılayıcı ağları için sunulan yönlendirme protokollerinin sınıflandırılması .....	25
Şekil 2.13.	Kablosuz algılayıcı ağ tasarımını etkileyen faktörler .....	31
Şekil 3.1.	Taşıma katman protokolünün genel yapısı .....	34
Şekil 3.2.	CAA taşıma katman protokollerinin sınıflandırılması.....	37
Şekil 3.3.	Siphon taşıma katman protokolü yerleşim yapısı.....	41
Şekil 3.4.	DTC protokolünün çalışma algoritması.....	45
Şekil 3.5.	RMST Protokol Yapısı .....	45
Şekil 3.6.	RBC protokolü çalışma mantığı .....	46
Şekil 3.7.	GARUDA güvenlik şemaları a) Tüm düğümlere güvenli teslim b) Alt bölgeye güvenli teslim c) Gözlem alanını kapsayan en az düğümle güvenli teslimat d) Düğümlerin %80'ine güvenli teslim.....	47
Şekil 3.8.	ESRT protokolü durum modeli ve geçişleri.....	51

Şekil 3.9.	WSAN mimarisinin gösterim şeması .....	52
Şekil 3.10.	SRCP paket başlığı .....	55
Şekil 4.1.	Uçtan-uca ve düğümden-düğüme geçiş parametrelerinin belirlenmesi.....	57
Şekil 4.2.	3, 5, 7, 9-atlama için düğümden-düğüme ve uçtan-uca gecikme değerleri.....	60
Şekil 4.3.	Uçtan-uca ve düğümden-düğüme kayıp tespiti geçiş grafiği.....	60
Şekil 4.4.	Senkronizasyon mesajı başlıkları.....	62
Şekil 4.5.	HİKMET protokolünde uçtan-uca kayıp segment telafisi.....	64
Şekil 4.6.	Ara NACK gönderim şeması.....	66
Şekil 4.7.	Düğümden-düğüme tekniğinin çalışmasını gösteren örnek bir senaryo.....	70
Şekil 5.1.	HİKMET, DTC ve PSFQ protokollerinin farklı atlama değerlerindeki %30 hata oranı için paket başarımların yüzdeleri.....	77
Şekil 5.2.	HİKMET, DTC ve PSFQ protokollerinin farklı atlama değerlerindeki %50 hata oranı için paket başarımların yüzdeleri.....	77
Şekil 5.3.	HİKMET, DTC ve PSFQ protokollerinin farklı atlama değerlerindeki %70 hata oranı için paket başarımların yüzdeleri.....	78
Şekil 5.4.	3-atlamalı bir ağ için HİKMET, PSFQ ve DTC protokolleri için uçtan-uca gecikme değerleri grafiği .....	79
Şekil 5.5.	5-atlamalı bir ağ için HİKMET, PSFQ ve DTC protokolleri için uçtan-uca gecikme değerleri grafiği.....	80
Şekil 5.6.	7-atlamalı bir ağ için HİKMET, PSFQ ve DTC protokolleri için uçtan-uca gecikme değerleri grafiği.....	80
Şekil 5.7.	9-atlamalı bir ağ için HİKMET, PSFQ ve DTC protokolleri için uçtan-uca gecikme değerleri grafiği .....	81
Şekil 5.8.	HİKMET, PSFQ ve DTC protokollerinin rasgele hata ortamındaki uçtan-uca gecikme değerleri grafiği .....	81
Şekil 5.9.	3-atlamalı bir ağ için HİKMET, PSFQ ve DTC protokollerinin enerji tüketim grafiği .....	82
Şekil 5.10.	5-atlamalı bir ağ için HİKMET, PSFQ ve DTC protokollerinin enerji tüketim grafiği .....	83

Şekil 5.11.	7-atlamalı bir ağ için HİKMET, PSFQ ve DTC protokollerinin enerji tüketim grafiği .....	83
Şekil 5.12.	9-atlamalı bir ağ için HİKMET, PSFQ ve DTC protokollerinin enerji tüketim grafiği .....	84
Şekil 5.13.	HİKMET,PSFQ ve DTC protokolleri rasgele hata ortamlarındaki paket başına enerji tüketimi grafiği .....	85
Şekil 5.14.	3-atlamalı bir ağ için HİKMET, PSFQ ve DTC protokollerinin paket maliyeti grafiği .....	86
Şekil 5.15.	5-atlamalı bir ağ için HİKMET, PSFQ ve DTC protokollerinin paket maliyeti grafiği .....	87
Şekil 5.16.	7-atlamalı bir ağ için HİKMET, PSFQ ve DTC protokollerinin paket maliyeti grafiği .....	87
Şekil 5.17.	9-atlamalı bir ağ için HİKMET, PSFQ ve DTC protokollerinin paket maliyeti grafiği .....	88
Şekil 5.18.	HİKMET, PSFQ ve DTC protokolleri için rasgele hata ortamlarındaki ortalama paket maliyeti grafiği .....	88

## TABLolar LİSTESİ

Tablo 2.1.	Başlıca kablosuz algılayıcı düğümleri ve özellikleri .....	15
Tablo 3.1.	Tıkanıklık kontrol protokolleri .....	38
Tablo 3.2.	Güvenilirlik Protokolleri.....	43
Tablo 3.3.	Tıkanıklık ve güvenilirlik protokolleri.....	49
Tablo 4.1.	Kayıp Telafi Mekanizmasının değişimini gösteren sahte kodu.....	61
Tablo 4.2.	HİKMET protokolünün uçtan-uca çalışma yöntemi .....	67
Tablo 4.3.	HİKMET protokolünün düğümden-düğüme çalışma yöntemi .....	73
Tablo 5.1.	Benzetim ortamı ayarları.....	76

## ÖZET

Anahtar Kelimeler: Kablosuz Algılayıcı ağlar, Taşıma Protokolleri, Kayıp Telafi, Düğüm-den-duğüme, Uçtan-uca

Kablosuz Algılayıcı Ağ (KAA)'lar, bakım gerektirmeden uzun yıllar çalışabilmeleri ve çok çeşitli alanlarda kullanılabilmesi sebebiyle hem endüstriyel uygulamalarda hem de akademik çalışmalarda çok popüler bir alan haline gelmiştir. KAA'ları meydana getiren düğümler, düşük fiyatlı, sınırlı enerjiye ve basit radyoya sahip cihazlardır. Ancak, maliyeti düşürmek için kablosuz algılayıcı düğümlerinde düşük kaliteli radyo alıcı/vericilerin kullanılması, düğüm iletişimlerinin ortam koşullarından kolaylıkla etkilenmesine ve paket kayıplarının ciddi manada fazla olmasına yol açabilmektedir.

KAA'larda güvenilir iletişim gerçekleştirmek birçok uygulama için son derece önemlidir. Özellikle de askeri ve tıbbi alanlardaki kritik uygulamalarda paket kayıpları ciddi sorunlara yol açabilir. Ayrıca, bu gibi uygulamalarda paketlerin düşük gecikmeler ile hedeflere iletilmesi de son derece önemlidir. Bu sebeple, birçok KAA uygulamasında yüksek güvenilirlik sağlayan ve gecikmeye-duyarlı bir taşıma katmanının bulunması gerekmektedir.

Bu çalışmada, yukarıda bahsedilen bu gereksinimler doğrultusunda KAA'lar için Hibrit Kayıp telafi Mekanizmasına dayanan yeni bir Taşıma (HİKMET) protokolünün tasarımı gerçekleştirilmiştir. Önerilen protokol, ortam koşullarına göre adaptif olarak kullandığı kayıp telafi mekanizmasını değiştirerek uygun güvenilirlik ve gecikme sağlayabilmektedir. HİKMET'in başarımlı değerlendirilmesi için Paket Başarımlı Oranı, Uçtan Uca gecikme, Enerji Maliyeti ve Paket Maliyeti olmak üzere dört farklı kıstas kullanılmıştır. Gerçekleştirilen detaylı benzetim sonuçlarına göre HİKMET, PSFQ ve DTC gibi iki önemli protokolden daha yüksek performans sağlamıştır.

# **A NEW TRANSPORT PROTOCOL DESIGN BASED ON HYBRID LOSS RECOVERY MECHANISM FOR WIRELESS SENSOR NETWORKS**

## **SUMMARY**

Keywords: Wireless Sensor Networks, Transport Protocols, Loss Recovery, Hop-by-Hop, End-to-End

Wireless Sensor Networks (WSNs) has become a very popular topic in academic research and industrial applications since they can work without maintenance for many years. Sensor nodes building a WSN are generally low-cost devices which have a simple radio and limited energy sources. However, the use of low-quality radio in sensor nodes to reduce the cost may cause the node communications to be affected by environmental conditions and this may result in serious packet losses.

The reliable communication in WSN is extremely important for many applications. Packet losses can lead to serious problems, particularly on critical applications such as military and medical fields. In addition, the transmission of packets with low end-to-end delay is extremely important in some applications. Therefore, delay-sensitive transport layer must be used to provide high reliability in many WSN applications.

In this thesis, hybrid loss recovery mechanism transport layer protocol is designed to meet the requirements mentioned above for WSNs. The proposed transport protocol provides optimum reliability and end-to-end delay by changing loss recovery mechanism according to environmental conditions adaptively. For performance evaluation of proposed algorithm, four different criterions are used. They are packet delivery ratio, end to end delay, energy cost and packet overhead. According to the performance results of a detailed simulation, proposed protocol has better performance characteristic than PSFQ and DTC which are one of the major protocols.



## **BÖLÜM 1. GİRİŞ**

Teknolojinin hızlı bir şekilde gelişmesiyle birlikte son yıllarda, daha yenilikçi ve faydalı teknolojiler üretilmektedir. Geçmiş zamanlarla kıyaslandığında her geçen gün daha fazla bilgi üretimi yapılmakta, bilgi erişimini kolaylaştıracak cihazların geliştirilmesine daha da fazla ihtiyaç duyulmaktadır. Özellikle, taşınabilirliğinin ve veri iletiminin daha kolay olması, ürünlerin ucuz maliyetli ve ergonomik olarak üretilmesi, kablosuz teknoloji cihazlarının milyonlarca insan tarafından kullanılmasını sağlamıştır.

Gelişen teknoloji ile birlikte işlemci, hafıza ve radyo frekans ile çalışan alıcı/verici cihazlara çok çeşitli algılayıcıların (sensör) entegre edilebilmesi söz konusu olmuş ve böylece Kablosuz Algılayıcı Düğüm (KAD) olarak adlandırılan yeni nesil mobil cihazlar kullanılmaya başlanmıştır. Üzerinde nem, ısı, basınç gibi çeşitli fiziksel büyüklükleri algılayan algılayıcılar barındıran bu düğümler bir araya gelerek ve ortak gayret sarf ederek bir bölgenin uzaktan gözlemlenmesine, izlenmesine ve çeşitli multimedya iletişimlerin gerçekleştirilmesine olanak sağlayan Kablosuz Algılayıcı Ağ (KAA)'larının geliştirilmesine olanak sağlamıştır. Algılayıcı düğümlerin düşük maliyetli olması, düşük güç tüketmesi, küçük boyutlarda üretilebilmesi, taşınabilirliğinin kolaylığı, uzun süre bakım istemeden çalışabilmeleri gibi avantajlarından dolayı KAA'lar çok değişik alanlarda kullanılmaya başlanmıştır.

Çok çeşitli kullanım alanları ve avantajlarına rağmen KAA'lar, diğer kablosuz ağlardan kendine has bazı farklılıklar sebebiyle ayrılmaktadır. Örneğin, KAA'larda düğüm yoğunluğu diğer ağlara oranla son derece fazla iken veri iletim hızları oldukça düşüktür. Bunun yanında, ağı meydana getiren düğümlerin boyutları diğer ağlardaki düğümlere göre daha küçük ve radyo alıcı/verici kalitesi ise son derece düşüktür. KAA'larda kalitesi düşük alıcı/vericilerin kullanılması düğüm maliyetini azaltmaya önemli bir katkı sağlarken paket kayıplarının da fazla olmasına yol

açmaktadır. Bu kayıplar, özellikle de askeri ve medikal alanlarındaki gibi kritik uygulamalarda son derece ciddi sıkıntılara yol açabilmektedir. Bu sebeple, araştırmacılar paket kayıplarının önemli olduğu kritik uygulamalarda güvenilir iletişim gerçekleştirmek için taşıma katman protokollerinin kullanılmasının bir gereklilik olduğunu vurgulamışlardır [1].

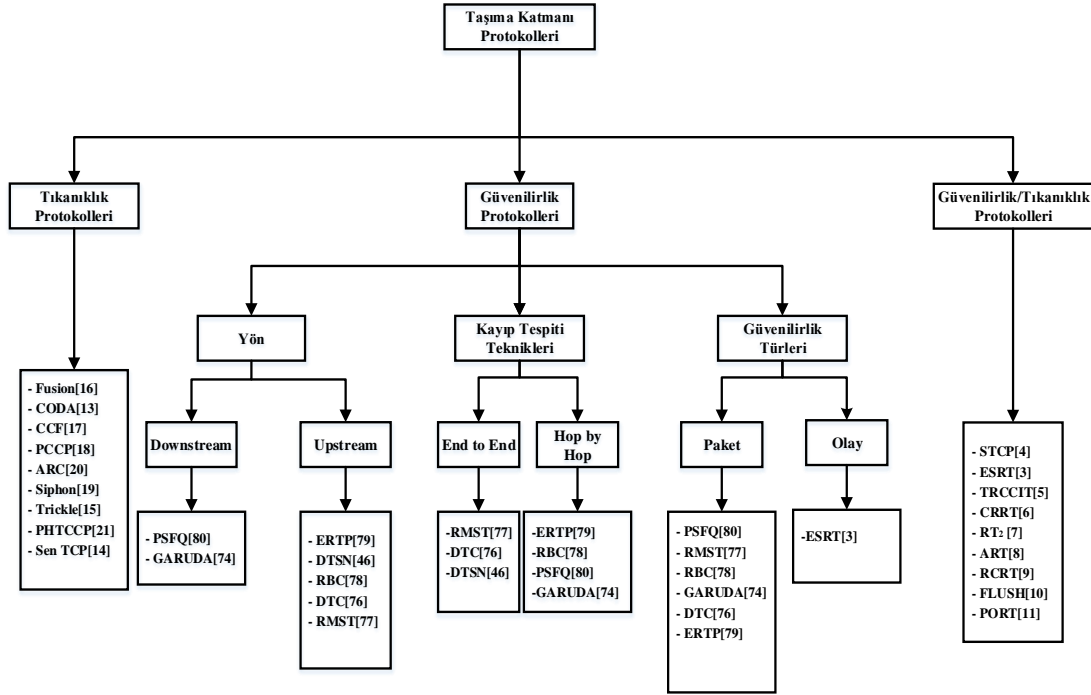
Taşıma katmanı protokolleri, genelde büyük boyutlu paketlerin küçük segmentlere ayrılarak uçtan-uca güvenilir bir şekilde iletilmesini garanti etmektedir. Buna ek olarak, tıkanıklık gibi paket kayıplarına yol açan durumları da gidererek paket kayıplarını minimize etmeye çalışmaktadır. Literatürde, KAA'lar için önerilen çeşitli taşıma katman protokolleri mevcuttur [1]. Ancak, özellikle de gerçek zamanlı hedef tespiti, bina güvenlik ve sızma tespit sistemleri gibi kritik uygulamalar için hem yüksek güvenilirliğe sahip hem de düşük güç tüketen, düşük uçtan-uca gecikmeli ve yüksek bant genişliği imkânı sunan taşıma katman protokollerine ihtiyaç duyulmaktadır.

Yukarıda bahsedilen ihtiyaç, bu tezin de çıkış noktasını oluşturmuş ve kötü ortam koşullarında bile yüksek paket başarımı sağlayabilen, düşük uçtan-uca gecikme ile paketlerin kaynaktan hedefe ulaştırılmasına imkân tanıyan ve düşük enerji tüketimini destekleyen bir taşıma katman protokolünün tasarımı hedeflenmiştir. Bu sebeple, bir sonraki başlıkta literatürde KAA'lar için önerilen taşıma katman protokolleri incelenmiştir. Ayrıca, Literatür özetinde verilen çalışmalara Bölüm 3'de detaylı bir şekilde değinilmiştir.

### **1.1. Literatür Özeti**

KAA'lar da taşıma katmanı genel olarak iki görev üstlenmektedir. Bunlardan birincisi, ağda meydana gelebilecek tıkanıklık durumlarını tespit ederek sorunu gidermek, ikincisi ise paketlerin kaynaktan hedefe güvenilir bir şekilde ulaşmasını garanti etmektir [2]. Literatürdeki bazı taşıma katman protokolleri hem tıkanıklık hem de güvenilirlik kontrolünü beraber sağlarken [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], birçok protokolde bu görevlerden sadece birisi yürütülmektedir. Örneğin, [13], [14], [15], [16], [17], [18], [19], [20], [21], [22] sadece tıkanıklık problemini ele

alan protokollerdir. Sadece güvenilirlik sağlayan protokoller ise güvenilir iletişim üzerine odaklanmaktadır ve literatürde birçok farklı açıdan kategorize edilmektedir. Bu tez çalışmasında, güvenilirlik protokollerini, güvenilirlik yönü, güvenilirlik türü ve kayıp telafi tekniği açısından 3 ana kategori altında inceledik. Şekil 1.1’de literatürdeki farklı özelliklere sahip taşıma katmanı protokollerin sınıflandırılması görülmektedir.



Şekil 1.1. KAA’lar için önerilen taşıma katman protokolleri

Şekil 1’de görüldüğü üzere güvenilirlik sağlayan protokollerin kategorilerinden birincisi güvenilirliğin sağlanma yönüne göredir. Güvenirliğin sink’den düğümlere doğru sağlandığı protokoller downstream, düğümlerden sink’e doğru sağlandığı protokoller ise upstream protokoller olarak adlandırılmaktadırlar. Bir diğer kategori ise protokollerin güvenilirlik türüne göredir. Literatürdeki çalışmalarda genelde iki farklı tür güvenilirlikten bahsedilmektedir. Bunlardan birincisi, paket güvenilirliği diğeri ise olay güvenilirliğidir. Paket güvenilirliği sağlayan protokollerde, tüm segmentler başarılı bir şekilde iletmeye çalışılmalı ve kayıp segmentler tespit edilerek hızlı bir şekilde telafi edilmelidir. Olay tabanlı güvenilirlik protokollerinde ise her paketin başarılı bir şekilde iletilmesi önemli değildir. Güvenirlik tabanlı protokollerde bir başka sınıflama türü de kayıp telafi yöntemine göredir. Birinci yöntemde, kayıp paketler düğümden-düğüme (hop-by-hop) telafi edilirken ikinci yöntemde uçtan-uca (end-to-end) kayıp telafisi yapılmaktadır. Düğümden-düğüme

hata tespiti ve kayıp telafisinde ara düğümler kayıpların tespitinde ve kurtarılmasına görev almaktadır. Bu yöntemde, kendisine yanlış sıra numarası ile segment geldiğini fark eden ara düğümler kayıp olduğunu kabul ederek eksik olan segmentleri bir önceki komşusundan Negatif ACK (NACK) ile isterler. Eksiklerin tamamlanması sonrasında ilgili segmentler bir sonraki düğüme gönderilebilir. Bu yöntemde, ara düğümlerden eksiklerin kurtarılabilmesi için segmentlerin tamponlanması gerekmektedir. Diğer bir yöntem olan uçtan-uca kayıp telafisinde ise eksik segmentlerin tespit edilmesinden yalnızca hedef düğüm sorumludur. Hedef düğüm belirli bir zaman aşımına ya da sırasız gelen segment numaralarına göre bir NACK paketi oluşturarak eksikleri diğer uçtaki kaynak düğümden istemektedir. Kaynak düğüm ise eksik segmentleri tampon belleğinden bularak tekrar hedefe iletmelidir. KAA'lar için önerilen güvenilir taşıma protokollerinde her iki yöntemde kullanılmasına karşın düğümden-düğüme hata tespiti kayıp telafi tekniği daha fazla tercih edilmektedir.

Şekil 1.1'de görüldüğü üzere KAA'lar için hem tıkanıklığın tespiti ve giderilmesi hem de güvenilir iletişimin sağlanması için birçok taşıma protokolü önerilmiştir. Bununla ilgili birçok araştırma makalesi literatürde yer bulmuştur [23], [24], [25], [2]. Ancak, sınır gözetleme, hedef takibi, Vücut Alan Ağları ve multimedya algılayıcı ağlar gibi paket kayıplarının son derece önemli olduğu ve verilerin, zor iletişim koşullarına rağmen düğümlerden sink'e zamanında iletilmesi gerektiği uygulamalarda güvenilirliği ve gecikme duyarlılığı yüksek protokol tasarımına ihtiyaç duyulmaktadır. Bu çalışmada, bahsedilen ihtiyaçlar doğrultusunda kayıp telafi tekniğini iletişim koşullarına göre otomatik olarak seçen ve böylelikle her türlü koşulda yüksek güvenilirlik ve düşük uçtan-uca gecikme sağlayabilen güvenilir bir taşıma protokolü tasarlanması hedeflenmiştir. Bu çalışmayı literatürdeki eşleniklerinden ayıran en önemli ayrıntı bildiğimiz kadarıyla her iki kayıp telafi yöntemini beraber bir şekilde kullanan ilk taşıma protokolü olmasıdır.

## **1.2. Tezin Amacı, İzlenen Çalışma Yöntemi ve Katkıları**

Sınırların uzaktan gözetlenmesi, kimyasal/nükleer sızıntıların tespit edilmesi, riskli hastalıkları bulunan kişilerin kontrol altında tutulması, bir binanın güvenliğinin

sağlanması gibi çok önemli ve hayati görevleri yerine getiren kablosuz algılayıcı ağlarda paket kayıpları ciddi sıkıntıların ortaya çıkmasına yol açabilmektedir. Bu gibi paket kayıplarına ve paketlerin bir uçtan diğer uca zamanında iletilmesine oldukça duyarlı uygulamalarda yüksek güvenilirliğe sahip bir taşıma protokolünün kullanılması gerekmektedir.

Bu tez çalışmasının ana amacı, KAA'lardaki veri güvenilirliğini sağlayacak, düğümden-düğüme ve uçtan-uca kayıp kurtarma yöntemlerini adaptif olarak kullanabilecek yeni bir taşıma protokolü tasarlamak, benzetim yoluyla tasarımını gerçekleştirmek ve başarımını incelemektir. Bu amaç doğrultusunda ilk olarak literatürde sunulan popüler taşıma modelleri incelenmiş avantaj/dezavantajları ele alınmıştır. Daha sonra ise literatürdeki çalışmaların eksiklerinden yola çıkarak adaptif hata kurtarma metodolojisini kullanan yeni bir taşıma protokolünün tasarımı gerçekleştirilmiştir. Son olarak da geliştirilen protokolün başarım analizi gerçekleştirilerek literatürdeki popüler taşıma protokolleri ile kıyaslanmıştır.

Bu tez çalışmasında önerilen yöntemler ve bu çalışmayı klasik eşleniklerinden ayıran katkılar özetle şunlardır:

1. Tezin tamamlanma sürecine kadar literatürde sunulan önemli taşıma protokolleri detaylı bir şekilde incelenerek sistematik bir araştırma gerçekleştirilmiştir.
2. Düğümden-düğüme hata kurtarma tekniği ile çalışan yeni bir taşıma protokolünün tasarımı gerçekleştirilmiştir.
3. Uçtan-uca hata kurtarma tekniği ile çalışan yeni bir taşıma protokolü tasarlanmıştır.
4. Tez kapsamında, literatürde ilk kez bir taşıma protokolünün ortam koşullarının değişimine göre kullandığı hata kurtarma tekniğini adaptif olarak değiştirebileceği önerilmiştir. Bu bağlamda, düğümden-düğüme ve uçtan-uca hata kurtarma teknikleri içeren ve bu tez çalışmasında tasarlanan iki taşıma

protokolü adaptif hata kurtarma tekniğini ile birleştirilerek enerji ve gecikme duyarlı yeni bir taşıma protokolünün tasarımı gerçekleştirilmiştir.

5. KAA'larının katmanlı ağ yapısı, güç tüketimi, farklı konumlandırma teknikleri gibi birçok detayın modellendiği modüler, kullanıcı arabirim destekli ve OMNET++ ve MIXIM tabanlı benzetim arayüz yazılımının tasarımı gerçekleştirilmiştir. KAA'lar için tasarlanan benzetim yazılımındaki bu hazır modeller kullanılarak birçok uygulamanın benzetimi kolaylıkla gerçekleştirilebilmektedir.

### 1.3. Tez Düzeni

Yapılan tez çalışması aşağıda özetlenen altı bölümden oluşmaktadır:

Bölüm 1: Giriş: Bu bölümde, tez çalışmasına konu olan problemin tanımı, yapılan çalışmanın amacı, yapılan tez çalışmasını diğerlerinden farklı kılan yönler, bilime katkısı ve tez organizasyonu hakkında bilgi verilmektedir.

Bölüm 2: Kablosuz Algılayıcı Ağlar: Bu bölümde tez konusunun temel çalışma alanını oluşturan kablosuz algılayıcı ağları hakkında detaylı bilgi verilmektedir. Algılayıcı düğümlerinin yapısı, kablosuz algılayıcı ağ mimarisi, kablosuz algılayıcı ağ tasarımını etkileyen faktörler ve kablosuz algılayıcı ağının uygulama alanları bu bölümde anlatılan konuları oluşturmaktadır.

Bölüm 3: Taşıma Protokolleri: 3.Bölümde ilk olarak taşıma katmanı açıklanmış, görevleri anlatılmış sonra ise KAA'lar için önerilen taşıma protokollerinin detayları verilmiştir.

Bölüm 4: Hibrit Kayıp Telafi Mekanizmasına Dayanan Yeni Bir Taşıma (HİKMET) Protokolünün Tasarımı: Bu bölümde, ortam koşullarının değişimine göre kullandığı hata kurtarma tekniğini değiştiren HİKMET protokolünün mimarisi ve çalışma mantığı ayrıntılı bir şekilde ele alınmıştır.

Bölüm 5: HİKMET Protokolünün Başarım Değerlendirmesi: Bu bölümde, önerilen HİKMET Protokolünün OMNET++ Benzetim ortamında detaylı benzetimleri gerçekleştirilmiş ve literatürdeki popüler protokoller ile başarımları kıyaslaması gerçekleştirilmiştir.

Bölüm 6: Sonuçlar ve Değerlendirmeler: Bu bölümde, 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.

Ek-A: Kablosuz Algılayıcı Ağ Modelleme ve Benzetim Araçları: Bu bölümde, modelleme/benzetim kavramları üzerinde durulmakta, günümüzde yaygın olarak kullanılan bilgisayar tabanlı benzetim yazılımlarının üstünlükleri ve zayıflıkları özetlenmekte ve son olarak da OMNET++ benzetim ortamının özellikleri, yapısı ve kullanım şekli kısaca açıklanmaktadır.

Ek-B: Kablosuz Algılayıcı Ağlara Yönelik OMNET++ Tabanlı Benzetim Modelinin Tasarımı: Bu bölümde, kablosuz algılayıcı ağlara yönelik olarak geliştirilen OMNET++ ve MIXIM tabanlı benzetim modelinin tasarım detayları ve kullanım şekli açıklanmaktadır.

## **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 gelmiş bir ağıdır. Bu ağlarda, düğümler rasgele olarak ortama bırakabilmekte ve geliştirilen protokoller sayesinde kablosuz ortam üzerinden birbirileri ile haberleşerek kendi kendine organize olabilmektedirler. Bu özellik, düğümlerin ortamdaki fiziksel büyüklük (ışık, sıcaklık, nem, basınç v.b.) değişimlerini çok atlamalı (multihop) yollar üzerinden merkezi ağ birimine iletmelerini mümkün kılmaktadır. Kablosuz algılayıcı düğümlerinin düşük maliyetli olması, normal şartlarda erişimin imkânsız olduğu bölgelere kolaylıkla yerleştirilebilmesi ve uzun süreler boyunca bakım istemeden çalışabilmesi gibi özellikler kablosuz algılayıcı ağlarının çok çeşitli alanlarda kullanılabilmesini mümkün kılmaktadır [26]. Şekil 2.1’de görüldüğü gibi kablosuz algılayıcı ağlar gözlem alanından, algılayıcı düğümlerden, çıkış düğümünden ve görev yönetim düğümünden meydana gelmektedir [27].

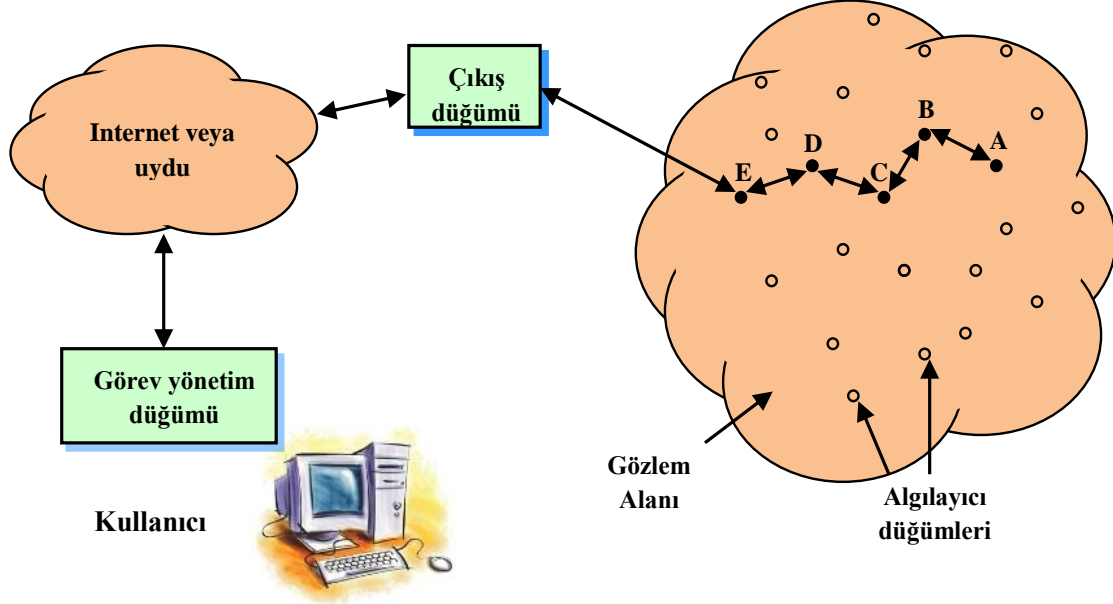
**Gözlem Alanı:** Belirli olayların olması beklenen ve algılayıcı düğümlerin yerleştirildiği alandır.

**Algılayıcı düğümler:** Ortamdaki verileri toplama ve çıkış düğümüne iletme görevini üstlenen düğümlerdir.

**Çıkış düğümü:** Çıkış düğümü, algılayıcı düğümlerden gelen paketlerin alınması işlenmesi ve saklanması ile görevli olan düğümdür. Gönderilecek toplam mesaj sayısının azalmasına yardım etmesi sebebiyle ağın toplam enerji tüketiminin azalmasına ve dolayısıyla ağ ömrünün uzamasına katkı sağlar. Çıkış düğümü,



mevcut enerji miktarı, tampon bellek doluluk oranı v.b ölçütlere göre ağ tarafından dinamik olarak seçilebilir. İcra ettikleri görev gereğince veri toplama noktası olarak da adlandırılabilirler.



Şekil 2.1. Kablosuz algılayıcı ağ örneği [27]

Görev yönetim düğümü ya da baz istasyon: Baz istasyon, ağdan gerekli olan bilgileri alan ve ağa kontrol bilgilerini göndermekle sorumlu olan merkezi kontrol noktasıdır. Ayrıca diğer ağlarla bağlantıyı sağlayan, güçlü veri işleme/saklama yeteneğine sahip ve kullanıcı ile arabirim sağlayan bir erişim noktasıdır. Baz istasyon olarak kullanılabilen dizüstü bilgisayar veya iş istasyonuna bilgiler radyo frekans, uydu veya internet ile iletilebilir [26].

## 2.2. Algılayıcı Düğümlerinin Tarihçesi

Algılayıcı düğümlerin tarihine bakıldığında ilk olarak Amerika Birleşik Devletleri (A.B.D) tarafından savaş yıllarında kullanıldığı görülmektedir [28]. Okyanus tabanındaki kritik bölgelere yerleştirilen akustik algılayıcı içeren düğümler Sovyet denizaltılarını gözetlemek amacıyla kullanılmış ve geliştirilen algılayıcı ağı "Ses Gözetleme Sistemi" (Sound Surveillance System - SOSUS) olarak adlandırılmıştır. Kablolu algılayıcı düğümlerin kullanıldığı bu sistemde veriler farklı katmanlarda işlendikten sonra kablolu ortam üzerinden kıyılarıdaki merkezlere iletilmiştir. Modern

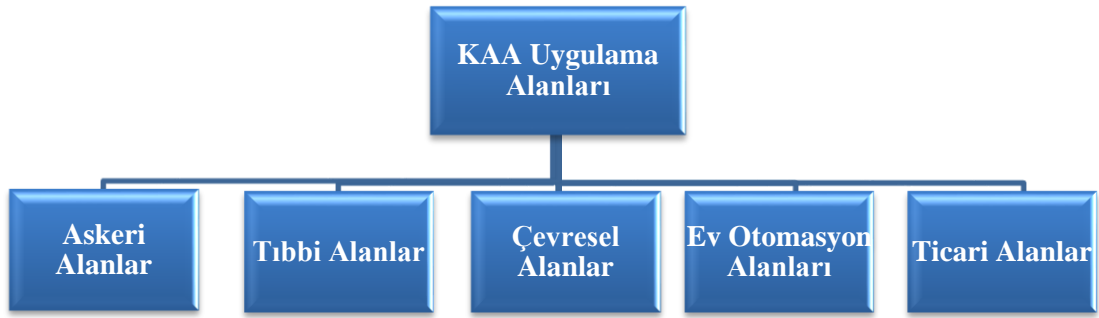
algılayıcı ađ arařtırmaları 1980'lerin bařlarında yine A.B.D'de DARPA'da bařlatılmıřtır. Dađıtık algılayıcı ađ (Distributed Sensor Network - DSN) programı olarak adlandırılan projede dūřuk maliyetli, bađımsız birçok dūđmūn dađıtık olmasına rađmen birbirileri ile iřbirliđi ierisinde olmaları hedeflenmiřtir [28]. 1980 ortalarında MIT (Massachusetts Institute of Technology) ses algılayıcılarından oluřan ve alak uuř gerekleřtiren uakların takibini sađlayan ōrnek sistem geliřtirmiřtir. Bu yapı Őekil 2.2'de gōrūlmektedir. Altıřarlı diziler halinde yerleřtirilen mikrofonlar sayesinde uakların ses sinyalleri ile algılanması hedeflenmiřtir. Akustik algılayıcılardan gelen ses sinyallerinin iřlenmesini sađlayan 512 KB hafızaya sahip bir bilgisayar ve ū adet iřlemciden oluřan hareketli ara, algılama dūđmūnū meydana getirmektedir. Dūđmūler birbirleri ile mikrodalga sinyaller yardımıyla haberleřmektedirler [28].



Őekil 2.2. MIT tarafından geliřtirilen ōrnek algılayıcı ađ sistemi [28]

### 2.3. Kablosuz Algılayıcı Ađ Uygulama Alanları

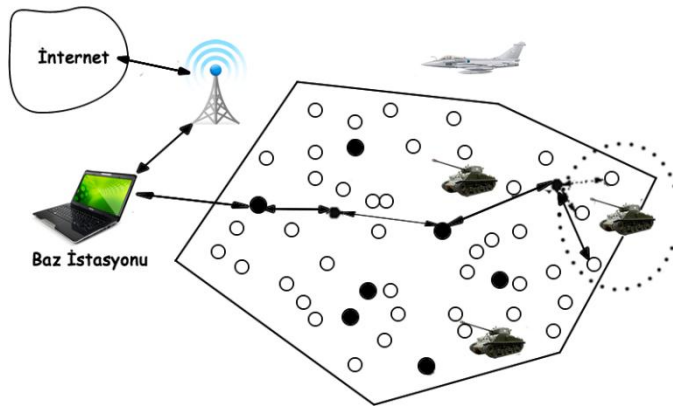
Mikro elektronik ve mekaniksel sistem (MEMS) tasarımındaki geliřmeler sıcaklık, nem, basın, titreřim, ses, gōrūntū ve kimyasal sızıntı gibi fiziksel būyūklüklerin sezilmesini sađlayan algılayıcıların kablosuz haberleřebilen dūđmūlere entegre edilebilmesini mūmkūn kılmaktadır. Bu sebeple gūnūmūzde kablosuz algılayıcı ađlar birok alanda gōzlem ve denetim amacıyla kullanılmaktadır. Algılayıcı ađların kullanım alanları Őekil 2.3'te gōrūldūđ gibi beř ana kategori altında incelenebilir [27]. Ancak algılayıcı ađların uygulama alanları bunlarla sınırlı deđildir.



Şekil 2.3. Kablosuz algılayıcı ağ uygulama alanları [27]

### 2.3.1. Askeri alanlar

Birçok teknolojik başarı askeri ihtiyaçlar sayesinde elde edilmektedir. Yine bir askeri gereksinim sebebiyle ortaya çıkmış olan algılayıcı ağlar günümüzde havacılık uygulamaları [29] gibi çok çeşitli askeri uygulamalarda karşımıza çıkmaktadır. Örneğin nükleer, biyolojik ve kimyasal saldırı tespiti, radyasyona maruz kalmadan kablosuz algılayıcı ağları ile yapılabilir [30]. Kritik araziler, yollar, patikalar, geçitler, boğazlar kolay ve hızlı bir şekilde algılayıcı ağları ile kaplanabilir ve düşman kuvvetleri gözlemlenebilir. Şekil 2.4'te askeri alanlarda uygulanan örnek bir uygulama görülmektedir.



Şekil 2.4. Askeri bir uygulamada kullanılan algılayıcı ağı [14]

### 2.3.2. Tıbbi alanlar

Kablosuz Algılayıcı ağları, özürllüler için arabirim oluşturma, hastalara teşhis koyma ve gözlem altında tutma, insanların psikolojik davranışlarının izlenmesi, hastane

içerisindeki hasta ve doktorların izlenmesi ve gözlemlenmesi, insan vücudunu inceleme gibi [31], [32], [33] birçok sağlık ile ilgili uygulama alanlarında kullanılmaktadır. Fransa'nın Grenoble tıp fakültesinde kurulan algılayıcı ağları yaşlıların düşme vb. davranışlarının gözlemlenmesine olanak sağlamaktadır [27]. Gerçekleştirilen bir başka uygulamada [34] ise hastane içinde ve dışında hastalarla ilgili detaylı bilgilerin toparlanması ve bu sayede kolay ve hızlı bir şekilde hasta tedavilerinin gerçekleştirilmesi hedeflenmektedir. Bu amaçla hastalara kalp atışını ve kalpteki oksijen yoğunluğunu ölçen küçük ve hafif algılayıcı düğümleri takılmaktadır [26].

### **2.3.3. Çevresel alanlar**

Kablosuz algılayıcı ağları bitki ve hayvanların ortamlarında gözlemlenmesi [35], [36], soyu tükenen hayvan türlerinin izlenmesi, okyanuslarda yeni canlıların keşfedilmesi gibi birçok ekolojik uygulamada kullanılmaktadır. Bunun dışında felaketlerin önceden tespiti algılayıcı ağ uygulamalarının en önemlilerindedir. Orman yangınlarının ve selin önceden tespit edilmesi [37], volkanik hareketliliğin sismik ve ses altı (infrasonic) algılayıcılar ile belirlenmesi [38], Tsunami'lerin ve depremlerin sismik algılayıcılar yardımıyla önceden tespiti [39], trafik kontrolü [40], hedef izleme [41] gibi birçok alanda kablosuz algılayıcı ağları kullanılabilir. Kablosuz algılayıcı ağlarının bir diğer ilginç uygulama alanı ise uzay keşifleridir. NASA gezegen keşiflerinde kablosuz algılayıcı ağlarından faydalanmayı hedeflemektedir [42].

### **2.3.4. Ev otomasyon alanları**

Kablosuz algılayıcı ağlar, merkezi ısıtma ya da soğutma sistemlerin verimini arttırmak amacıyla kullanılabilir. Merkezi ısıtma sistemlerinde bir oda diğerinden daha sıcak ya da daha soğuk olabilir veya gelen hava akışı her tarafa eşit dağılmayabilir. Kablosuz algılayıcı düğümleri sıcaklık ve hava akışını kontrol etmek için odaların farklı bölgelerine yerleştirilebilir. Bu sayede tahminen %44 enerji tasarrufu sağlanabilir [43]. Bir diğer uygulama alanı ise bina güvenliğinin sağlanmasıdır. Kritik bölgelere yerleştirilen düğümler izinsiz kişilerin binaya girmesini

engellenen yanında meydana gelebilecek gaz kaçağı, yangın [44] vb ev otomasyon sistemlerinde [45] herhangi bir felaketten önceden tespit edilmesi amacıyla kullanılabilir.

### 2.3.5. Ticari alanlar

Stokların yönetilmesi [46], ürün kalitesinin ölçülmesi, zeki ofis alanlarının oluşturulması, kablosuz algılayıcı ağlarının ticari uygulama alanları arasındadır. British Petrol, petrol saklama ortamlarında meydana gelebilecek tehlikeli durumların tespiti ve petrol tankerlerinin motorlarındaki titreşim kontrolü için [47], Boeing firması ise uçak kanatlarındaki basıncın gözlemlenmesi [48] amacıyla algılayıcı ağları kullanmaktadır. Kablosuz algılayıcı ağlarının bir diğer ilginç uygulama örneği ise kuzey kutbundaki petrol borularının sıcaklığının kontrol edilmesidir. Isıtılmayan borular donma sebebiyle patlayacağından sürekli olarak sıcaklarının gözetilmesinde tutulması gerekmektedir.

### 2.4. Kablosuz Algılayıcı Düğüm Yapısı

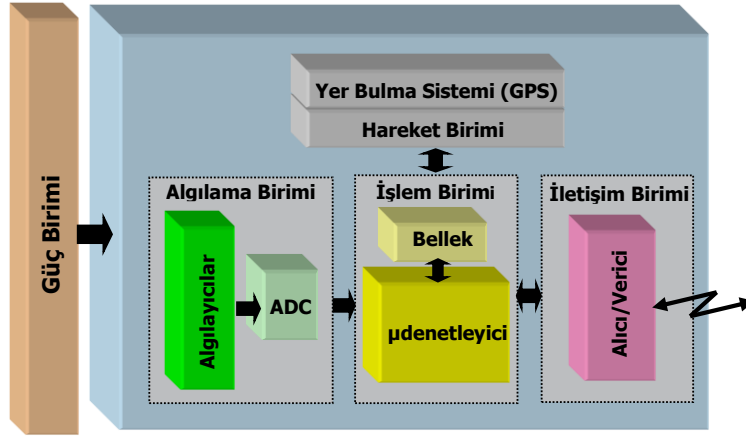
Günümüzde teknolojik gelişmeler düşük güçlü, düşük maliyetli, çok fonksiyonlu ve kısa mesafede kablosuz ortam üzerinden haberleşebilen algılayıcı düğümlerinin gelişmesini sağlamıştır. Şekil 2.5'te bir algılayıcı düğümün genelleştirilmiş mimarisi görülmektedir.

Bir algılayıcı düğümünü meydana getiren birimler ve icra ettikleri görevler şunlardır:

- Algılama Birimi: Algılayıcılar ve ADC (Analog/Digital Converter-Analog/Sayısal Çevirici)'lerden meydana gelen algılama birimi ışık, nem v.b. fiziksel büyüklüklerin ortamdan elde edilmesi ve bu büyüklüklerin işlem birimi tarafından işlenebilecek forma getirilmesinden sorumludur.
- İşlem Birimi: Mikrodenetleyici ve bellek birimlerinden oluşan işlem birimi, kod bellekte yüklü olan ve düğümlerin ağ içerisinde yapmakla yükümlü olduğu görev komutlarının işlenmesinden sorumludur. Kablosuz algılayıcı düğümlerinde algılama, veri işleme, gönderme/alma gibi sürekli kullanılan

yordamların tanımlanmış olduğu ve daha kolay uygulama geliştirebilmeye imkân sağlayan gerçek zamanlı bir işletim sisteminden faydalanılır. TinyOS (Tiny Operating System) [49], kablosuz algılayıcı ağlarda en yaygın biçimde kullanılan işletim sistemlerinden birisidir ve kaynakları sınırlı olan algılayıcı düğümlerine uygun olarak küçük boyutludur.

- İletişim Birimi: Bir düğümü ağ içerisindeki diğer düğümlere bağlayan iletişim birimi, düşük güçlü RF alıcı/vericiden meydana gelmiştir. Algılayıcı düğümlerinde kullanılan alıcı/verici birimleri genellikle gönderme, alma, aylak ve uyku olmak üzere dört çalışma moduna sahiptir. Düğüm içerisinde alıcı/vericinin en fazla güç tüketen birim olduğu ve alıcı/vericide en fazla gücün sırasıyla gönderme, alma, aylak ve uyuma modlarında harcandığı düşünüldüğünde, iletişim biriminin bir düğümün yaşam süresinin belirlenmesinde büyük önem taşıdığı anlaşılmaktadır. Bu sebeple, geliştirilen protokollerde alıcı/verici mümkün olduğu kadar uyuma modunda tutulmaya çalışılır [26].
- Güç Birimi: Güç birimi, dolaylı olarak tüm ağın ömrünü belirlemesi sebebiyle algılayıcı düğümlerinin en önemli birimidir. Boyut sınırlaması nedeniyle algılayıcı düğümlerde genellikle standart AA piller veya kristal hücreler başlıca kullanılan güç kaynaklarıdır. Bazı uygulamalarda güneş enerjisi ile şarj olabilen piller tercih edilebilmektedir. Böylelikle bir düğümün ömrü yaklaşık olarak 7–10 yıla kadar çıkabilmektedir [26].
- Hareket Birimi: Sadece gezgin olan düğümlerde bulunan hareket birimi düğümün hareket yönlerinin ve hızlarının yönetilmesinden sorumludur.
- Yer Bulma Sistemi: Her düğümde olma zorunluluğu olmayan yer bulma sistemi (Global Position System–GPS) düğümlerin küresel olarak konumlarını belirleyebilmesini sağlayan birimdir. Düğüm maliyetlerinin ve boyutlarının artmasına sebep olan GPS cihazlarının genellikle sınırlı sayıdaki düğümde bulunması tercih edilmekte ve diğer düğümler konumlarını bu düğümlere göre belirlemektedirler.



Şekil 2.5. Kablosuz algılayıcı ağ düğüm mimarisi [27]

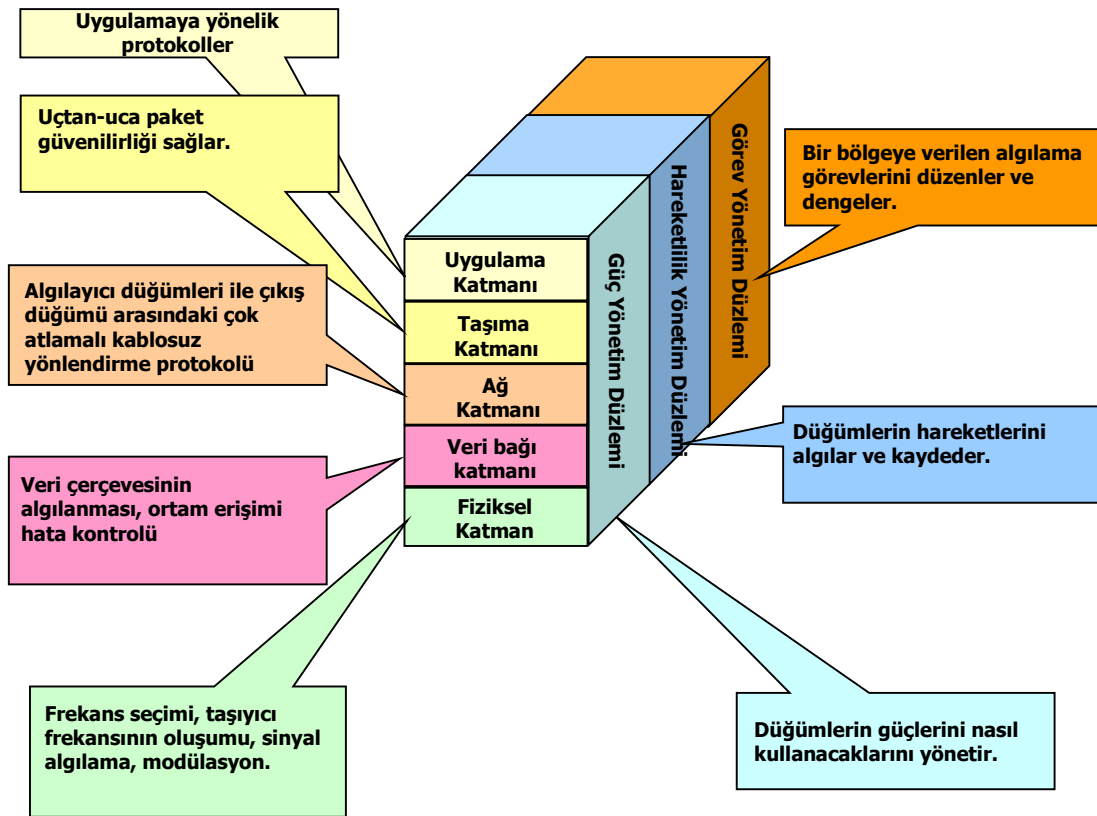
Günümüzde birçok üniversite ve şirket, akademik veya ticari amaçlı olarak algılayıcı düğüm üretmektedir. Tablo 2.1’de başlıca kablosuz algılayıcı düğümlerin özellikleri listelenmektedir.

Tablo 2.1. Başlıca kablosuz algılayıcı düğümleri ve özellikleri [50]

Algılayıcı Düğüm Türü ve Yılı	WeC 1998	Rene 1999	Rene2	Dot 2000	Mica 2001	Mica2Do t2002	Mica2 2002	Telos 2004	
<b>Mikrodenetleyici</b>									
Türü	AT90LS8535		ATMega163		ATmega128			TIMSP430	
Program Belleği(KB)	8		16		128			60	
Veri Belleği (KB)	0.5		1		4			2	
Aktif Güç (mW)	15		15		8		33	3	
Uyku Gücü ( $\mu$ W)	45		45		75		75	6	
Uyanma Süresi ( $\mu$ S)	1000		36		180		180	6	
<b>Kalıcı Saklama Birimi</b>									
Entegre	24LC256				AT45DB041B			STM24M0TS	
Bağlantı Türü	I <sup>2</sup> C				SPI			I <sup>2</sup> C	
Büyüklüğü	32				512			128	
<b>İletişim Birimi</b>									
Alıcı/Verici	TR1000				TR1000	CC1000		CC2420	
Veri Aktarım Hızı	10				40	38.4		250	
Modülasyon Türü	OOK				ASK	FSK		O-QBPSK	
Alım Gücü (mW)	9				12	29		38	
Gönderim Gücü(mW)	36				36	42		35	
<b>Güç Tüketimi</b>									
Min. Çalışma gerilimi	2.7		2.7		2.7			1.8	
Toplam harcanan güç	24				27	44	89	41	
<b>Programlama ve algılayıcı arabirimi</b>									
Genişleme	Yok	51-pin	51-pin	Yok	51-pin	19-pin	51-pin	10-pin	
İletişim	IEEE 1284 (Programlama) ve RS 232							USB	
Tümleşik Algılayıcılar	Yok	Yok	Var	Yok	Yok	Yok	Yok	Var	

## 2.5. Kablosuz Algılayıcı Ağ Mimarisi

Kablosuz algılayıcı ağ mimarisi için beş katmanlı ve üç düzlemlilik protokol yığını tanımlanmıştır [27]. Protokol yığını Şekil 2.6'da görüldüğü üzere uygulama katmanı, taşıma katmanı, ağ katmanı, veri bağı katmanı ve fiziksel katman olmak üzere beş katman ile birlikte güç yönetim düzlemi, hareketlilik düzlemi ve görev yönetim düzlemi ile birlikte meydana gelmektedir. Düğümlerin ortak gayret sarf ederek kaynakların etkin bir şekilde kullanılmasını destekleyen yönetim düzlemleri, özellikle kaynak sıkıntısı olan kablosuz algılayıcı ağlar için büyük önem taşımaktadır [26].



Şekil 2.6. Kablosuz algılayıcı ağ mimarisi [27]

- Güç yönetim düzlemi: Algılayıcı düğümlerinin güçlerini nasıl kullanacaklarını yöneten düzlemdir. Örneğin gücü azalan bir düğüm, komşularına gücünün yeterli olmadığını duyurabilir ve böylelikle mesajların yönlendirilmesine katılmayıp kalan gücünü algılama işlemlerine ayırabilir.



- Hareketlilik yönetim düzlemi: Algılayıcı düğümlerinin hareketlerinin algılanmasından ve kaydedilmesinden sorumlu düzlemdir. Düğümler hareketlilik düzlemi sayesinde komşularının takibini ve bölgesel algılama görevlerini gerçekleştirebilirler.
- Görev yönetim düzlemi: Düğümlere atanacak görevlerin planlanmasından ve yürütülmesinden sorumlu olan düzlemdir. Örneğin bir bölgedeki düğümlerin hepsinin aynı anda algılama işlemini gerçekleştirmesine gerek duyulmayabilir ve bazı düğümler güçlerine göre diğerlerinden daha fazla görev yürütebilirler. Bu durumda düğümler arasında görevlerin taksimini görev yönetim düzlemi gerçekleştirir.

### 2.5.1. Fiziksel katman

Frekans seçimi, taşıyıcı frekansının oluşumu, sinyal algılama, modülasyon, gönderim ve alım işlemlerinin yürütüldüğü katmandır. Fiziksel katman, güç tüketimini doğrudan etkilediği için kablosuz algılayıcı düğüm tasarımında ayrı bir öneme sahiptir. Seçilen modülasyon tekniği, iletim hızı, gönderme gücü ve görev çevrim süresi gibi güç tüketimini etkileyen faktörler fiziksel katman tasarımı ile ilgili parametrelerdir. Fiziksel katman tasarımında önemli olan unsurlardan bir tanesi de haberleşme yöntemidir. Günümüz algılayıcı düğümleri genellikle kısa mesafeli kablosuz iletişim ile haberleşmektedir. Gönderim mesafesi, gönderim gücüne bağlı olduğu için algılayıcı düğümlerde genellikle kısa mesafeli iletişim tercih edilir. Düğümler ISM (Industrial, Scientific, Medical –Endüstriyel, Bilimsel, Tıbbi) bandı olarak bilinen lisansız frekanslarda haberleşmektedir. Avrupa ve Japonya’da 433 MHz/868 MHz frekansları genellikle tercih edilirken Amerika Birleşik Devletlerinde 915 MHz ve 2.4 GHz frekansları kullanılır [27].

Fiziksel katman tasarımında önemli olan unsurlardan bir diğeri de modülasyon tekniğidir. Kablosuz algılayıcı ağları çoğu durumda zor doğa koşulları altında çalışmak zorunda oldukları için seçilen modülasyon tekniğinin gürültüye, girişime ve boğma (jamming) saldırılarına karşı dayanıklı olması gerekmektedir. Frekans atlamalı yayılım spektrumu (Frequency-Hopping Spread Spectrum-FHSS) ve

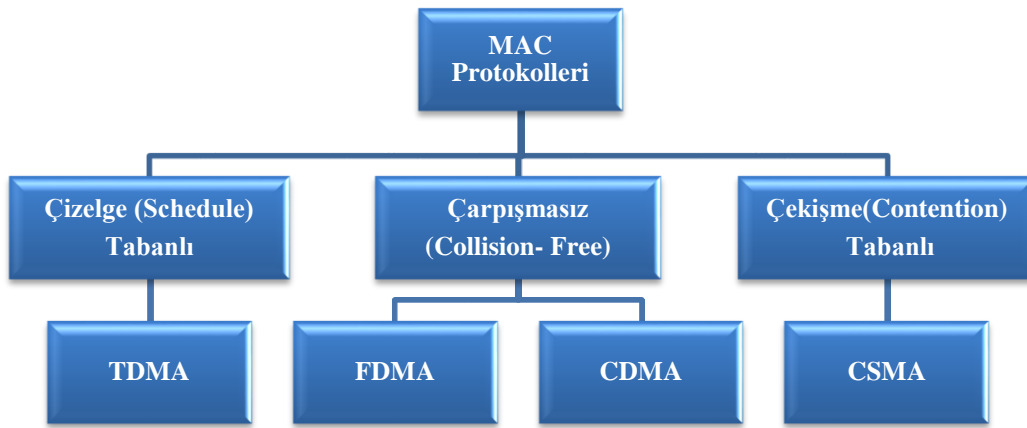
doğrudan sıralı yayılım spektrumu (Direct-Sequence Spread Spectrum-DSSS) kablosuz ağlarda ve kablosuz algılayıcı ağlarda kullanılan modülasyon tekniklerindedir. Her iki teknikte girişime dayanıklı olmasına karşın DSSS tekniği dar bant girişimlerine FHSS'ye oranla daha dayanıklıdır. Ayrıca ultra geniş bant, darbe radyo ve darbe konum modülasyon teknolojilerinin kullanımı KAA'larda enerji tüketiminin azalmasına ve daha güvenilir iletişimin gerçekleşmesine olanak sağlayacaktır [27].

### **2.5.2. Veri bağı katmanı**

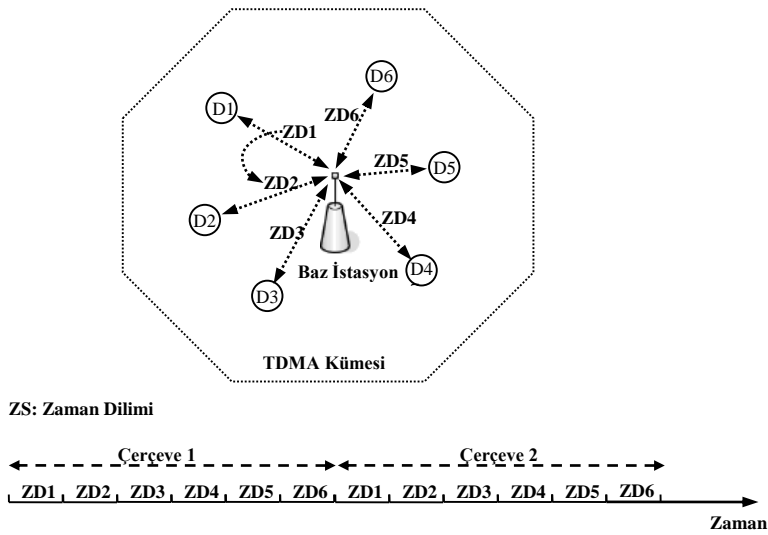
Veri çerçevesinin algılanması, erişim ortamı ve hata kontrolünden sorumlu olan katmandır. Bir iletişim ağında noktadan noktaya ve bir noktadan çok noktaya iletişimin güvenilir ve adil bir şekilde yapılmasını sağlar. Veri bağı katmanı temelde mantıksal bağlantı kontrolü (Logical Link Control-LLC) ve ortam erişim kontrolü (Medium Access Control-MAC) olmak üzere iki kısımdan meydana gelmektedir. MAC, adil ve güvenli haberleşmenin yürütülebilmesi ve enerji tüketiminin düşürülmesi ile ilgili olarak önemli roller üstlenmektedir. Bu sebeple kablosuz algılayıcı ağlarda veri bağı katmanındaki çalışmalar genellikle ortam erişim kontrol mekanizması üzerine odaklanmaktadır. Literatürde kablosuz algılayıcı ağları için geliştirilmiş olan birçok çalışma bulunmaktadır. Geliştirilen MAC protokolleri Şekil 2.7'de görüldüğü gibi üç ana kategoride toplanmaktadır [51].

#### **2.5.2.1. Çizelge tabanlı ortam erişim protokolleri**

Zaman çizelgesi esasına dayanan ortam erişim protokolleri çarpışmayı engellemek için hangi düğümün ne zaman iletişime başlayabileceğine karar veren merkezi çizelge algoritması kullanmaktadır. Zaman bölümlenmeli çoğullama (Time Division Multiple Access-TDMA) ise paylaşımlı olan iletişimin kanalının N tane dilime (slot) ayrıldığı ve her zaman diliminde sadece bir düğümün gönderim yapabildiği çizelge tabanlı algoritmadır. TDMA, düğümlerin iletişim zamanlarının yönetilmesini sağlayan merkezi bir baz istasyona gereksinim duymaktadır. Şekil 2.8'de görüldüğü gibi baz istasyonun kapsama alanındaki düğümler ve baz istasyon, hücre yapısını oluşturmaktadır.



Şekil 2.7. Ortam erişim protokol ailesi



Şekil 2.8. TDMA yönteminin yapısı [51]

TDMA protokolleri düşük enerji ile çarpışmasız iletişim sunmasına rağmen bazı zayıflıklara sahiptir;

- Hareketli düğümler TDMA yapısı için önemli bir sorundur. Hareketli düğümlerin diğer düğümler ile iletişim kurabilmesi için baz istasyonla irtibat halinde olmaları gerekmektedir.
- TDMA düğümler ile baz istasyonlar arasında katı bir zaman senkronizasyonuna ihtiyaç duyulmaktadır.

- Özellikle düğüm yoğunluğu fazla olan ağlarda gönderim sırasının beklenmesi sebebiyle gecikmeler önemli ölçüde artmaktadır.

Literatürde kablosuz algılayıcı ağlar için TDMA esasına dayanan birçok MAC protokolü önerilmiştir. LEACH [52], PACT [53], TRAMA [54] ve LMAC [55] bu protokollerden başlıcalarıdır.

### 2.5.2.2. Çarpışmasız ortam erişim protokolleri

Çarpışmasız ortam erişim protokolleri çarpışmayı farklı radyo kanalları (frekans ya da kod) kullanarak engeller. Böylelikle iki düğüm arasındaki eş zamanlı iletişim girişimsiz ve çarpışmasız olarak gerçekleştirilir. Kablosuz iletişimde iki farklı çarpışmasız ortam erişim yöntemi kullanılmaktadır.

- FDMA (Frequency Division Multiple Access – Frekans Bölümlemeli Çoğullama): FDMA yönteminde frekans spektrumu, ayrı frekanslardaki bantlara ayrılmıştır. İletişim yapacak olan her bir düğüm çifti bu bantlardan birisini seçerek iletişimi gerçekleştirir. Farklı radyo kanalları sayesinde çarpışmasız bir biçimde eş zamanlı iletişim gerçekleştirilebilir.
- CDMA (Code Division Multiple Access- Kod Bölümlemeli Çoğullama): TDMA yönteminde var olan bütün spektrum zamanın belli bir bölümü için sadece bir düğüme tahsis edilirken FDMA yönteminde var olan spektrumun belli bir kısmı sürekli olarak bir düğüme tahsis edilir. Kod bölümlemeli çoğullama tekniğinde var olan bütün spektrum her zaman bir düğüme tahsis edilebilir. CDMA tekniği tek bir taşıyıcı frekansı ve bir dizi dikey kodların kombinasyonları ile iletişimin gerçekleştirilmesi esasına dayanmaktadır [56]. Bu teknikte gönderici düğüm, gönderim yapmadan önce göndereceği paketi belirli dikey kodlarla ÖZEL VEYA işlemine tabi tutar. Alıcı düğüm ise gelen paketi aynı kodlarla tekrar ÖZEL VEYA işlemine tabi tutarak orijinal bilgiyi elde eder.

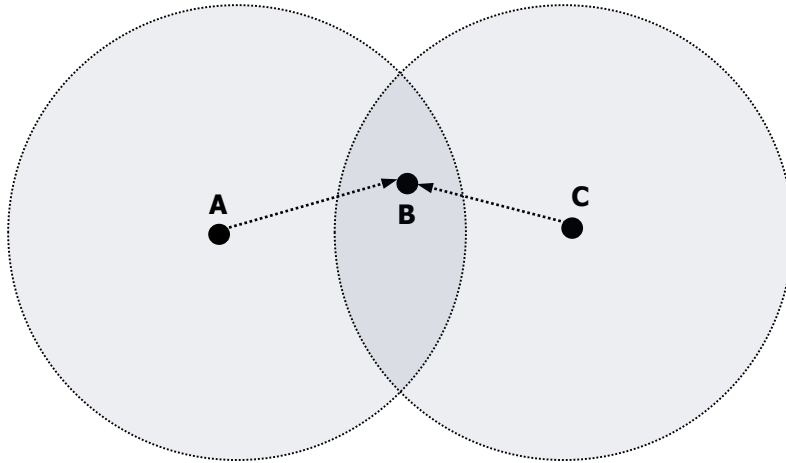
FDMA ve CDMA protokolleri aynı kablosuz ağın farklı kümeleri arasındaki iletişimde kullanılabilir. Her bir kümeye farklı frekans veya kod atanması sayesinde girişimsiz ve çarpışmasız bir biçimde kümeler arasında iletişim gerçekleştirilebilir. Ancak FDMA tekniği farklı radyo kanalları ile dinamik olarak haberleşebilmek için fazladan devrelere ihtiyaç duymaktadır. Daha fazla ve karmaşık devre beraberinde daha yüksek maliyeti getirmektedir. CDMA tekniğinin yüksek işlem yükü gerektirmesi ise düğümlerin enerji tüketimlerinin önemli oranda artması sebep olmaktadır [57]. Bu gibi sebepler CDMA ve FDMA tekniklerinin kablosuz algılayıcı ağlarda kullanılmasına engel teşkil etmektedir. Literatürde FDMA tekniğine göre SMAC [58] protokolü sunulmuştur. CDMA tekniği ile ilgili olarak önerilen çalışmalardan bazıları ise PicoRadio [59] ve DS-CDMA [60] protokolleridir.

### 2.5.2.3. Çekişme tabanlı ortam erişim protokolleri

Çekişme tabanlı protokoller, çarpışmayı tamamen engellemek yerine olma olasılığını azaltmaya çalışırlar. Tek kanallı radyo iletişiminde kanal tüm düğümler tarafından paylaşılmaktadır ve kanal tahsisi isteğe göre yapılmaktadır. Böyle bir durumda aynı anda birden fazla düğüm gönderim isteğinde bulunursa çarpışma kaçınılmazdır. Çarpışmayı engellemek ya da olasılığını azaltmak için iletim hakkını eline geçirmek isteyen düğümler arasında kanal tahsisini gerçekleştirecek dağıtık algoritmalar kullanılmaktadır. Çoğu dağıtık MAC protokolü çekişme esasına dayanır ve taşıyıcı duyarlı iletişim ve/veya çarpışmadan kaçınma mekanizmalarını kullanır. Bu yüzden gönderimden önce dinleme esasına dayanan taşıyıcı duyarlı çoklu iletişim (Carrier Sense Multiple Access -CSMA) olarak bilinirler. Kanalı dinlemenin amacı, gönderime başlamadan önce meşgul olmadığından emin olmaktır. Eğer kanal meşgul değilse düğüm hemen iletme başlar. Eğer kanal meşgul ise rasgele bir zaman boyunca bekler ve bu süre bittikten sonra kanalı yeniden dinler (non-persistent CSMA türlerinde) ya da kanal boş olana kadar dinlemeye devam eder ve kanalın boş olduğunu tespit ettiğinde iletme başlar [51].

Çok atlamalı kablosuz ağlarda CSMA-tabanlı protokoller Şekil 2.9'da görüldüğü gibi gizli düğüm probleminin ortaya çıkmasına yol açmaktadırlar. Gizli düğüm problemi,

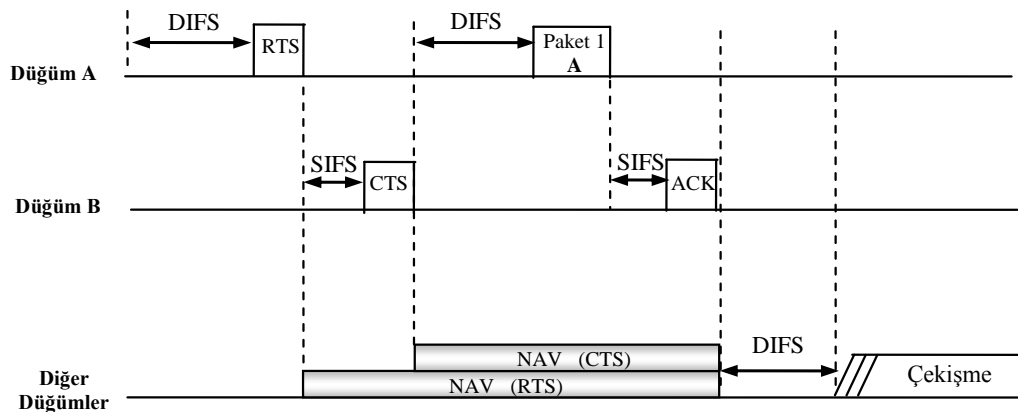
birbirinin kapsama alanında olmayıp aynı komşu düğüme sahip düğümlerin birbirinin iletişimini engellemesi esasına dayanır. A düğümü, B düğümüne bir paket göndermeye başladığında; C düğümü, A'nın kapsama alanında olmaması sebebiyle iletişimi hissedemez ve kanalın boş olduğunu varsayarak B düğümüne paket gönderimini başlatabilir. Böyle bir durumda, B düğümü çarpışma nedeniyle paketi alamaz. Literatürde fazladan kontrol sinyalleri kullanarak gizli düğüm problemi giderilmeye çalışılmıştır. El sıkışması yönteminde, veri paketi gönderiminden önce gönderici düğüm ile alıcı düğüm arasında kontrol paket değişimi olmakta ve böylece alıcı düğümün kapsama alanında olan komşu düğümler iletişimden haberdar olmaktadır. El sıkışması yöntemini kullanan en bilinen protokol IEEE 802.11 tarafından da desteklenen CSMA/CA (Collision Avoidance-Çarpışmadan Kaçınma) protokolüdür. Bu protokolda veri paketi göndermek isteyen düğüm ilk olarak küçük boyutlu RTS (Request to Send – Gönderim İsteği) paketi gönderir. Alım işlemini kabul eden düğüm ise CTS (Clear to Send) paketi ile cevap vererek gönderici ile alıcı arasındaki veri paketi akışının başlamasını sağlar. RTS/CTS paketini duyan komşu düğümler ise backoff (geri çekilme) durumuna geçerek iletimlerini ertelerler.



Şekil 2.9. CSMA protokollerinde ortaya çıkan gizli düğüm problemi

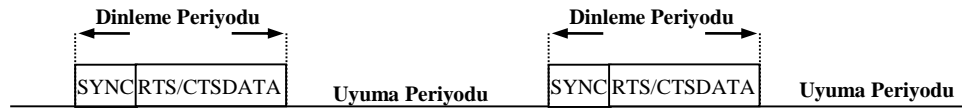
Gizli düğüm problemi, RTS paketlerinin çarpışma olasılığı dışında CSMA/CA tekniği ile büyük ölçüde çözüme kavuşmuştur. CSMA/CA yöntemi esas alınarak geliştirilen bir çalışmada (MACA) [61] RTS ve CTS paketlerine iletimin ne kadar süreceğini gösteren süre bilgisi eklenmiştir. Bu sayede RTS/CTS paketlerini duyan

komşu düğümler, iletim erteleme sürelerini doğru bir şekilde tahmin edebilmekte ve RTS paket çarpışmalarının önüne geçilmektedir. Bir diğer çalışmada ise (MACAW) [62], gönderimin başarılı olup olmadığını gösteren ACK (Acknowledgement-Kabul) paketinin kullanılması önerilmiştir. IEEE 802.11 protokolü, CSMA/CA, MACA ve MACAW protokollerinin birleşimini, dağıtık eş güdüm fonksiyonu (Distributed Coordinator Function-DCF) içerisinde bulundurmaktadır. Şekil 2.10'da IEEE 802.11 ortam erişim fonksiyonunun çalışma mantığı görülmektedir. 802.11 ortam erişim fonksiyonunun çalışma mantığı görülmektedir. 802.11 protokolünde gönderim yapmak isteyen düğüm, ortam erişim hakkını elde etmek için diğer düğümlerle çekişmek zorundadır. Bu sebeple düğümler gönderimden önce ortamı dinler ve eğer ortam meşgul ise iletimidaha sonraki bir zamana ertelerler (rasgele bir süre boyunca bekler). İletişim ortamını DIFS (Distributed Coordination Function Interframe Space)'tan daha uzun bir süre boyunca boş bulan bir düğüm gönderim yapmak istediğini belirten bir RTS paketi gönderir. İlk RTS paketini gönderen düğüm, ortam erişim hakkını kazanmış olur. Gönderici ile alıcı arasında RTS-CTS-DATA-ACK şeklindeki paket değişimi ile iletişim sonlanır. İki düğüm iletişim halindeyken diğer düğümlerin ne kadar beklemesi gerektiği RTS/CTS sinyallerinde bulunan ve iletişimin ne kadar süreceğini gösteren süre kısmında belirtilmektedir. Düğümler kontrol paketlerinden elde edilen süre bilgilerini, NAV (Network Allocation Vector) olarak adlandırılan değişkenlerine kaydeder ve zaman ilerledikçe NAV değişkenini güncellerler ve NAV sıfıra eşit olana kadar ortam erişimi için herhangi bir girişimde bulunmazlar.



Şekil 2.10. IEEE 802.11 ortam erişim fonksiyonu

Literatürde kablosuz algılayıcı ağlar için CSMA tabanlı birçok MAC protokolü önerilmiştir. Kablosuz algılayıcı ağları için tasarlanmış olan ortam erişim protokolleri arasında en yaygın olarak bilinen S-MAC [63], 802.11 protokolünden esinlenerek geliştirilmiş protokollerden birisidir. S-MAC protokolünü 802.11'den ayıran en önemli fark, enerji tüketiminin azaltılmaya çalışılmasıdır. Güç tüketimini azaltmak için ise radyo alıcılarını sürekli çalıştırmak yerine periyodik olarak açılıp kapatılması öngörülmüştür. Şekil 2.11'de görüldüğü gibi düğümler periyodik olarak dinleme/uyuma zamanlamasını kullanmakta ve iletişimlerini dinleme süresi içerisinde gerçekleştirmektedirler. Böylece zamanın büyük bir bölümünde uyuma moduna geçerek enerji tüketimlerini önemli ölçüde azaltmaktadırlar.



Şekil 2.11. S-MAC protokolünün görev çevrimi

Literatürde S-MAC dışında T-MAC [64], D-MAC [65], WiseMAC [66], B-MAC [67] gibi CSMA tabanlı birçok MAC protokolü mevcuttur.

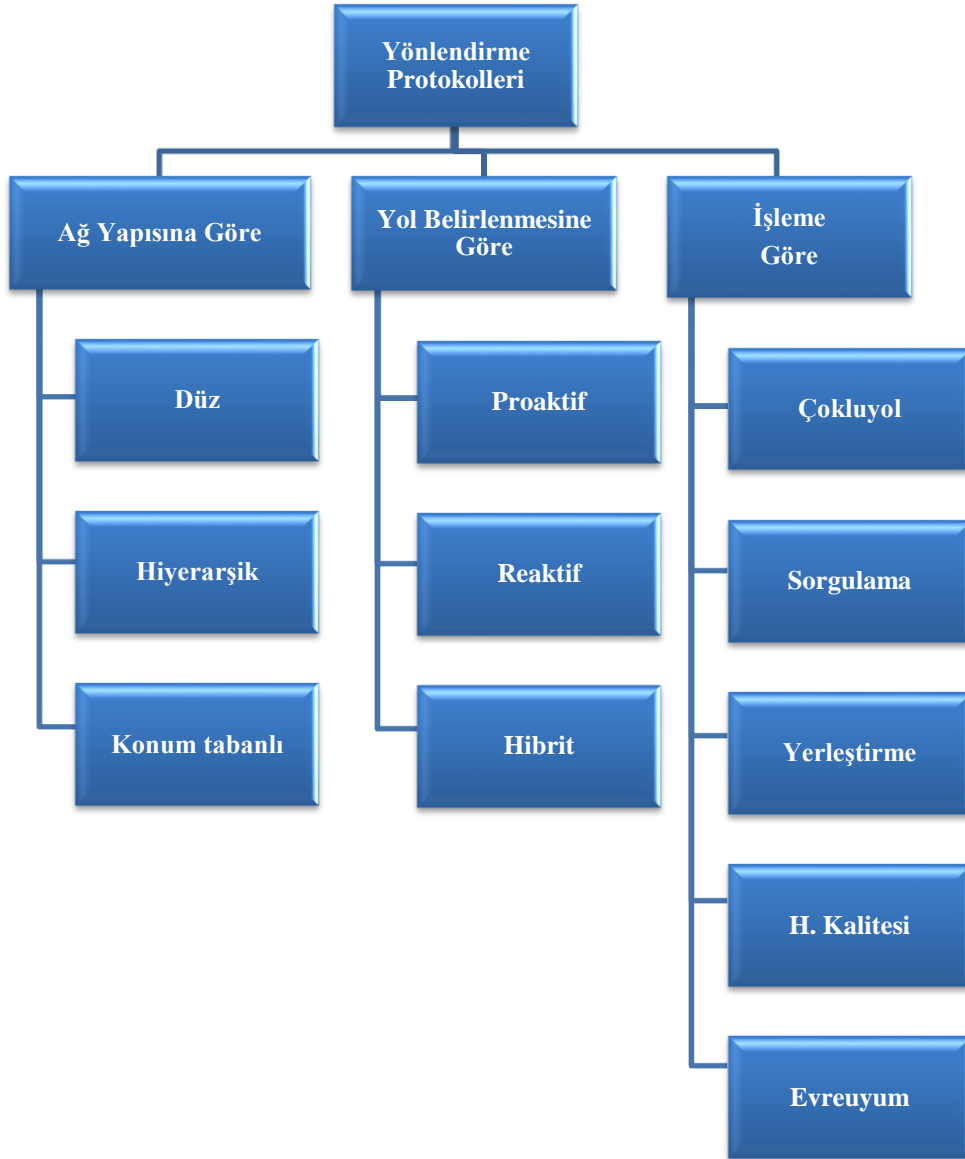
### 2.5.3. Yönlendirme katmanı

Paketlerin çok atlamalı yollar üzerinden çıkış düğümüne iletilmesi için gerekli olan protokolünün gerçekleştiği katmandır. Yönlendirme protokolü paketin hedefe ulaşabilmesi için gerekli olan en etkin yolu bulma görevini üstlenir. Kablosuz algılayıcı ağlar için sunulan yönlendirme protokolleri Şekil 2.12'de görüldüğü gibi ağ yapısına göre, yolların belirlenme şekline göre ve ağ işlemlerine göre üç kategoriye ayrılmaktadır.

Ağ yapısına göre yapılan sınıflandırma da sunulan protokoller düz ağ (Flat Networks) yönlendirme protokolleri, hiyerarşiksel ağ yönlendirme protokolleri ve konum tabanlı yönlendirme protokolleridir. Düz ağ yönlendirme protokollerinde tüm düğümler eşit görev üstlenmekte, hiyerarşiksel protokollerde hiyerarşisine göre farklı görevlere sahip olmakta ve konum bilgisine dayalı protokollerde ise yollar düğüm konumları baz alınarak belirlenmektedir. Diğer sınıflandırma yöntemi, yönlendirme



yollarının belirlenme şekline göredir. Proaktif protokoller, ağ içerisindeki bütün yolları başlangıçta belirlerler ve bütün yollar düğümlerdeki yönlendirme tablolarında saklanır. Eğer yollarda her hangi bir değişiklik olursa değişiklik bilgisi tüm ağa yayılır. Bununla beraber reaktif protokoller, yolları başlangıçta değil sadece gerekli olduğunda belirlerler.



Şekil 2.12. Kablosuz algılayıcı ağları için sunulan yönlendirme protokollerinin sınıflandırılması [68]

Hibrit protokoller ise bu iki yaklaşımı beraber kullanmaktadır. Sabit olan ağlar için proaktif protokollerin kullanılması enerji tüketimi açısından daha verimli olabilir.

Çünkü yolların belirlenmesi önemli ölçüde enerji tüketimini arttırmaktadır. Bir diğer sınıflandırma yöntemi ise ağda gerçekleştirilen işlem türlerine göredir.

Adaptif yönlendirme protokolleri bazı sistem parametrelerini ağ koşullarına göre uyarlayabilirler. Böyle protokoller çokluyol (multipath) tabanlı, sorgulama (query) tabanlı, müzakere (negotiation) tabanlı, hizmet kalitesi (QoS) tabanlı ve evreyum (coherent) tabanlı olmak üzere beş kategoriye ayrılabilirler.

#### **2.5.4. Taşıma katmanı**

Taşıma katmanı, oluşacak paket kayıplarını engelleyerek, ağ üzerindeki tıkanıklıklara çözüm getirerek ağın daha verimli bir şekilde çalışmasını sağlayan katmandır. KAA'larda genellikle bir uygulamanın servis kalitesi ve sistemin bekası gibi problemler bulunmaktadır. Servis kalitesi ve güvenilirlik konularına değişik yönlendirme protokolleri, MAC protokolleri sınırlı da olsa çözüm getirebilmektedir. Örneğin, bir MAC protokolü paketin sadece 1-atlama öteye gidip gitmediğine karar verebilir. Yönlendirme protokolleri de güvenilirliği en uygun yol seçimine karar vererek iyileştirebilir. Var olan çoğu uygulama MAC ve yönlendirme protokolleri üzerine odaklanır. Bununla birlikte, ne yönlendirme [68] ne de MAC protokolleri [57] bağımsız olarak bu problemleri istenen şekilde çözebilir.

Son zamanlarda, taşıma katmanı protokolleri, özellikle tıkanıklık kontrolü ve güvenilirlik [69] garantisi için, KAA'ların performansını iyileştirmek için daha çok dikkat çekmektedir. Taşıma katmanının gerekli ve önemli olmasının aşağıda ifade edildiği gibi bir kaç tane sebebi vardır.

KAA'larda, özellikle yüksek hızda veri akışı olan kablosuz multimedya algılayıcı ağlarda, tıkanıklık kaçınılmazdır [23]. Tıkanıklık paket kayıplarına yol açar ve buna bağlı olarak servis kalitesini (QoS) etkiler, gereksiz enerji tüketimine sebep olur. Buna bağlı olarak düşük servis kalitesi ve düşük enerji verimliliğine yol açar. MAC ve yönlendirme protokolleri tıkanıklıkla yeterince baş edemez. Taşıma katmanı tıkanıklığı azaltmak ve kaçınmak için kullanılır. Bu sayede paket kayıplarını da azaltmış olur. Örneğin, eğer taşıma katmanı tıkanıklığı tespit edip hemen ortadan

kaldırırsa, tıkanıklığa bağlı paket kaybı önemli derecede azalır. Azalan paket kaybı daha az tekrar iletimi sağlar ve ileri de dolaylı yoldan sadece enerji verimliliğini değil, aynı zamanda başarılı paket teslim oranını da iyileştirmiş olur.

Taşıma katmanının güvenilirliği garanti etmesi gereklidir. Genellikle paket kaybı, bit hataları veya bellek doluluğundan meydana gelir. Güvenilirliği garanti etmek için, istenen gereksinimleri karşılamak için, kaybolan paketlerin başarılı bir şekilde iletildiğinden emin olmak gerekir. Her ne kadar MAC katmanı bit hatalarını telafi etse de, tıkanıklığa bağlı kayıp paketleri kurtaramaz. Bu düşünceye göre, taşıma katmanı gereklidir. KAA'lar sadece güvenilirlik ve enerji verimliliğini dikkate almaz. Aynı zamanda ağ üzerindeki her algılayıcı düğüme eşit olarak davranmalı, her düğüm algılayıcı bilgisini adil olarak hedefe iletebilmelidir.

Taşıma katman protokolleri, enerji verimliliği ve servis kalitesini arttırmak için çok önemlidir. Çünkü bu protokoller tıkanıklığı azaltmak ya da kaçınmak için kullanılabilir ve bu sayede paket kayıpları azalır, bant genişliği yerleşiminde adalet sağlanır ve uçtan-uca iletim güvenliği garanti edilir. Kablolu ağlar için kullanılan TCP [70] ve UDP [71] gibi protokoller KAA'lar için uygun değildir. KAA'lar için tasarlanan protokollerin bu gereksinimleri karşılayacak şekilde tasarlanmaları gerekir.

### **2.5.5. Uygulama katmanı**

Özellikle düğümlerin doğrudan internet veya diğer harici ağlarla irtibat halinde olması istediğinde veri akışının sağlanmasında gerekli olan katmandır. Genellikle uygulamalarda kablosuz algılayıcı ağ düğümlerinin harici bir ağa doğrudan bağlanması gerekmez. Bu sebeple literatürde taşıma katmanı ile ilgili çok fazla çalışma bulunmamaktadır.

### **2.6. Kablosuz Algılayıcı Ağ Tasarımını Etkileyen Faktörler**

Kablosuz algılayıcı ağları, gözlem yapılacak ortama kolay ve hızlı bir şekilde yerleştirilebilmeleri, kendi kendine organize olarak uzun yıllar kontrol edilmeksizin

çalışabilmeleri gibi özelliklere sahip olması sayesinde çok çeşitli alanlarda kullanılabilir. Ancak, bu gibi kolaylıklar beraberinde kablosuz algılayıcı ağlarında görevlerin yürütülmesini sağlayan protokol ve algoritmaların tasarımını zorlaştırmaktadır. Şekil 2.13'te kablosuz algılayıcı ağ tasarımı gerçekleştirirken dikkat edilmesi gereken hususlar görülmektedir [27],[72].

- Yerleştirme (Deployment): Kablosuz algılayıcı ağlarında düğümler rasgele dağıtılabilir veya isteğe bağlı olarak seçilmiş yerlere yerleştirilebilir. Yerleştirme işlemi kurulum aşamasında yapılabileceği gibi ağın kapsama alanını genişletmek veya bozulan düğümler ile yenilerini değiştirmek için herhangi bir zamanda da yapılabilir. Bu sebeple KAA'lar için tasarlanan algoritmaların yerleştirme gereksinimlerini karşılayacak özellikte olmaları gerekmektedir [26].
- Gezinlik (Mobility): Düğümlerin konumları yerleştirme sonrası kasıtlı olarak veya kaza ile değişebilir. Örneğin rüzgâr, su v.b doğal etkenler düğümlerin yerleşim noktalarının değişmesine sebep olabilir. Bunun dışında bazı düğümler hareket kabiliyetine sahip olabilir ve bu sayede hareketli olarak gözlem görevlerini yürütebilirler. Sürekli olabileceği gibi zamana ve duruma göre gerçekleşebilen gezinlik, kablosuz algılayıcı ağ protokolleri için önemli bir tasarım ölçütüdür. Özellikle ortam erişim ve yönlendirme protokollerinin tasarım özelliği gezinliğe bağlı olarak değişebilir.
- Maliyet, Boyut, Sınırlı Kaynaklar ve Enerji: Kablosuz algılayıcı ağlarında düğüm sayıları uygulamaya göre binlere, on binlere ve hatta milyona ulaşabilir. Bu sebeple düğüm maliyeti çok önemlidir. Düğüm boyutları ise bir tanecik büyüklüğünde olabileceği gibi bir cep telefonu büyüklüğünde de olabilir. Düğümler, tıbbi uygulamalarda olduğu gibi bir insan vücudunun çeşitli bölgelerine yerleştirilebilir ve bu sebeple düğüm boyutlarının küçük olması beklenir. Düşük maliyet ve küçük boyutlar ise beraberinde kaynak sıkıntısını getirmektedir. Bir düğüm sınırlı işlem yapma kabiliyetine, saklama birimlerine ve enerji kapasitesine sahiptir. Bu sebeple KAA'lar için geliştirilen algoritma ve protokollerin çok karmaşık olmaması ve enerji

tüketimini en aza indirmesi gerekmektedir. Enerjisi biten düğüm, çoğu uygulama senaryosu için bir daha kullanılamaz demektir. Bu sebeple geliştirilen algoritmalar birinci öncelik olarak enerji tüketimine odaklanmalıdır [26].

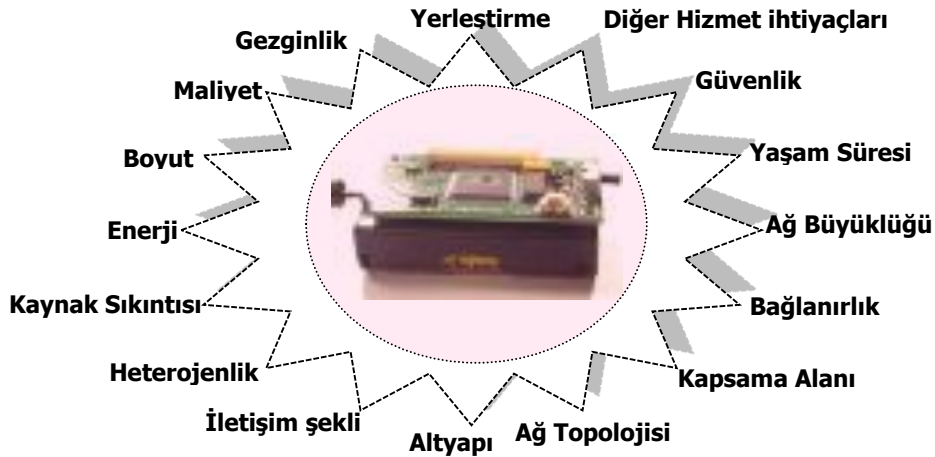
- Heterojenlik: Geçmişteki algılayıcı ağ uygulamalarında ağ içerisindeki düğümlerin tek tip olduğu (homojen) varsayıldı. Ancak günümüzde kablosuz algılayıcı ağlarında her düğüm aynı özellikte olamamaktadır. Örneğin maliyetlerin ve boyutların artmasına sebep olan GPS gibi konumlandırma cihazlarının her düğümde olması gerekmeyebilir. Geliştirilen yöntemlerle GPS cihazlarına sahip olmayan düğümler konumlarını GPS'e sahip olan düğümlerin konumlarına göre belirleyebilirler. Bu sebeple bir algılayıcı ağında heterojenliğin derecesi geliştirilecek olan algoritma ya da protokollerin karmaşıklığının artması ile yakından ilgilidir [26].
- İletişim Şekli: Kablosuz algılayıcı ağlarda radyo frekans ile iletişim dışında lazer, endüktif veya kapasitif bağlaşım (coupling), ses gibi farklı tekniklerle haberleşme gerçekleştirilebilmektedir. Maliyet ve kullanım kolaylığı gibi sebeplerle çoğu uygulamada radyo iletişimi tercih edilmektedir. İletişim şekli fiziksel katman ve ortam erişim katmanlarının fonksiyonları ile yakından ilgilidir [26].
- Altyapı (Infrastructure): Kablosuz ağlarda iletişim ağı altyapı-tabanlı ağlar ve tasarsız (ad-hoc) ağlar olmak üzere iki farklı yöntemle kurulabilir. Tasarsız ağlarda düğümler birbirleri ile herhangi bir altyapıya gerek duymadan haberleşebilirler. Bu sebeple çoğu uygulamada tasarsız ağ yapısı kullanılır. Fakat bunun yanında tasarsız ağlarda düğümler hem veri kaynağı hem de potansiyel bir yönlendirici olduğundan tasarsız ağlar için geliştirilen yönlendirme protokollerinin tasarımı altyapı tabanlı ağlara nazaran daha karmaşıktır [26].
- Ağ Topolojisi: Ağ topolojisi, kablosuz algılayıcı ağlarının önemli tasarım ölçütlerinden birisidir. Tek atlamalı ağlarda düğümler, diğer düğümler ile

doğrudan haberleşebilirken çok atlamalı ağlarda haberleşme keyfi olarak belirlenen atlamalar üzerinden gerçekleşir. Topoloji, gecikme, sağlamlık ve kapasite gibi önemli ağ karakteristiklerini etkiler ve paketlerin yönlendirilme şeklini belirler.

- **Kapsama Alanı:** Bir düğümdeki algılayıcıların etkin alanı o düğümün kapsama alanını belirler. Ağın kapsama alanı ise ağ içerisindeki düğümlerin gözlem yapılması gereken alanının ne kadarını kapsadığıdır. Kapsama alanının derecesi bilgi işlem algoritmalarını yakından ilgilendirmektedir. Düşük kapsama oranı ile güvenilir bir gözlem yapılması mümkün değildir. Örneğin bir ev güvenlik sisteminde eve yerleştirilen düğümlerin evin her tarafını ya da önemli bölgelerini kapsamadığı düşünülürse şüpheli kişiler eve girse bile tehlike ağ tarafından sezilmeyecektir [26].
- **Bağlanırlık (Connectivity):** Düğümlerin fiziksel konumları ve iletişim mesafeleri bir ağın bağlanırlığını göstermektedir. Eğer düğümler arasında daima bir iletişim bağlantısı varsa bu ağ bağlı olduğu söylenebilir. Eğer düğümler genellikle ayrıkça ve bazen diğer düğümlerin iletişim alanına giriyorsa böyle iletişim de düzensiz (sporadic) iletişim olarak adlandırılır. Bağlanırlık veri toplama metotlarının ve iletişim protokollerinin tasarımını etkilemektedir.
- **Ağ Büyüklüğü:** Ağdaki düğüm sayısı ağın bağlanırlığı, kapsama alanı ve gözlem alanının büyüklüğüne göre belirlenir. Ağ birkaç adet düğümden meydana gelebileceği gibi binlerce adet düğümden de meydana gelebilir. Dolayısıyla kablosuz algılayıcı ağları için tasarlanacak olan protokollerin ölçeklenebilir olması yani birkaç düğüm için olduğu gibi birkaç bin düğüm için de uygun şekilde çalışabilmesi gerekmektedir.
- **Yaşam süresi:** Uygulamaya bağlı olarak bir algılayıcı ağının yaşam süresinin birkaç saatten birkaç yıla kadar sürmesi beklenebilir. Yaşam süresi doğrudan enerji tüketimine ve düğümün güvenilirliğine bağlıdır. Enerji tüketimi de

fiziksel düğüm tasarımından kullanılacak protokol tasarımına kadar algılayıcı ağ tasarımının her alanını etkileyen önemli bir faktördür [26].

- **Güvenlik:** Algılayıcı düğümleri güvenli olmayan dış ortamlarda çalışmak zorunda olabilir, bazı doğal koşullar sebebiyle hasara uğrayabilir veya kötü niyetli kişiler tarafından ele geçirilerek saldırgan düğüm olarak yeniden programlanabilir. Dolayısıyla kablosuz algılayıcı ağlar için tasarlanacak olan algoritma ya da protokollerin algılayıcı ağların doğasında olan bu güvenlik açıklarını kapatacak şekilde olması gerekmektedir.
- **Diğer Hizmet Kalite Gereksinimleri:** Kablosuz algılayıcı ağları bir olayın belirli zaman dilimi içerisinde rapor edilmesi, bazı düğümlerde hata olsa bile ağdan beklenen görevlerin aksamaması gibi hizmet kalite gereksinimlerini karşılaması gerekmektedir [26].



Şekil 2.13. Kablosuz algılayıcı ağ tasarımını etkileyen faktörler [26]

## 2.7. Sonuçlar

Bu bölümde son yıllarda bilgisayar ağları alanında oldukça popüler bir konu haline gelen kablosuz algılayıcı ağlarının temel özellikleri açıklanmakta, kablosuz algılayıcı ağ uygulamaları hakkında bilgi verilmekte ve kablosuz algılayıcı ağ tasarımını etkileyen değişik etkenlerden bahsedilmektedir. Kablosuz algılayıcı ağ tasarımını etkileyen bu faktörlerden güvenilirlik konusu son yıllarda akademisyenler tarafından

ilgi görmeye başladığından dolayı, araştırmaya açık olan konuların başında gelmektedir. Bu sebeple tez çalışmasında KAA'ların paket güvenilirliği sorunları üzerine odaklanılmakta ve bir sonraki bölümde KAA'larda güvenilirliği sağlamaktan sorumlu olan taşıma katmanı ve literatürde var olan algoritmalar hakkında detaylı bilgi verilmektedir.



## **BÖLÜM 3. TAŞIMA KATMANI**

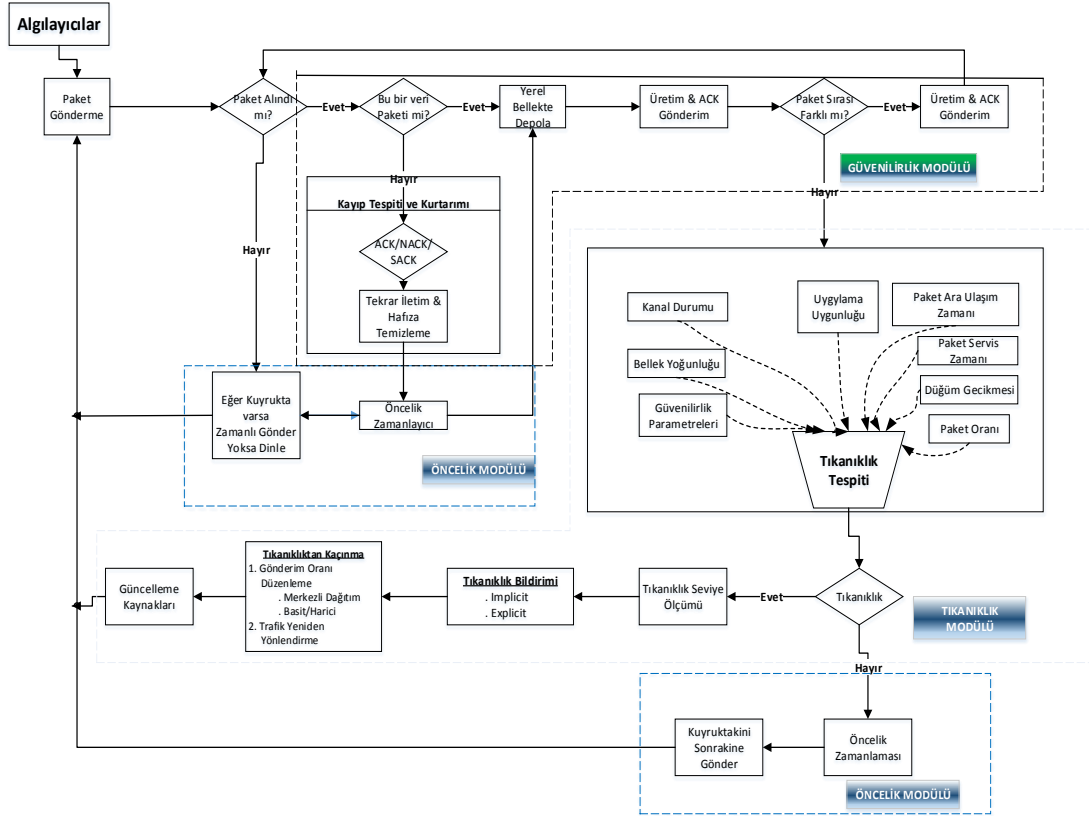
### **3.1. Giriş**

Kablosuz algılayıcı düğümler, ortamdaki farklı nicelikleri ölçebilecek şekilde bir ya da birden fazla algılayıcı ile donatılmıştır. Bu sayede, ortamdaki fiziksel büyüklüklerin algılanarak bölgenin uzaktan izlenmesine imkân sağlarlar. Ancak, algılayıcı düğümlerin maliyetlerini ve enerji tüketimlerini azaltmak için düşük kaliteli radyo alıcı/verici cihazların kullanılması, düğümlerin paket iletim başarımlarının düşük olmasına yol açabilmektedir. Özellikle de askeri ve tıbbi uygulamalar gibi paket hassasiyetinin yüksek olduğu alanlarda oluşan bu paket kayıpları bazen çok büyük sorunlara yol açabilmektedir. Bu yüzden, kritik uygulamalarda, paket kayıplarının tespit edilerek hızlı bir şekilde telafi edilmesi gerekmektedir. Geleneksel ağlarda olduğu gibi KAA'larda da uçtan-uca paket hatalarının tespiti ve telafisinden Taşıma Katmanı sorumludur. Son yıllarda, KAA araştırmacılarının dikkatlerini üzerine çeken güvenilir taşıma katman protokolleri [73] bu tezin de ana konusunu oluşturmaktadır. Bu sebeple, 3.bölümde KAA'lar için önerilen taşıma katman protokolleri detaylı bir şekilde incelenektir. Bölüme, ilk olarak taşıma katmanın temelleri hakkında bilgi verilerek başlanacaktır. Daha sonraki kısımda, taşıma katmanı etkileyen tasarım kriterleri ele alınacaktır. Son kısımda ise literatürde yer bulan başlıca taşıma katman protokolleri sistematik olarak sınıflandırılacak ve çalışma metotları ayrıntılı bir şekilde anlatılacaktır.

### **3.2. KAA'larda Taşıma Katman Protokolünün Görevi**

Kablosuz kanal kalitesinin düşüklüğü, düğümlerin uyku moduna girmeleri ve tıkanıklık gibi nedenlerden dolayı, kablosuz algılayıcı ağlarda önemli oranda paket kaybı olabilmektedir. KAA'larda kullanılan taşıma katmanı protokolleri, meydana gelen paket kayıplarını telafi edebilmeli ve buna bağlı olarak güvenilirliği garanti

etmelidirler. Şekil 3.1’de bir taşıma katman protokolünün çalışma metodolojisini gösteren yapı görülmektedir.



Şekil 3.1. Taşıma katman protokolünün genel yapısı [1]

Şekilden görüldüğü üzere bir taşıma katman protokolünün üç modülü bulunabilmektedir. Bu modüllerden birincisi olan “Güvenilirlik Modülü”, gönderilecek paketlerin küçük boyutlu segmentlere ayrılmasından ve her bir segmente sıra numarası verilmesinden sorumludur. Ayrıca, alınan segmentlerin sırasının kontrol edilerek hataların tespitinden ve telafisinden de sorumludur. Bu modüle eğer hatalı sıra numarasına sahip bir paket geldi ise bu tespit etmeli ve protokolün hata kurtarma tekniğine göre bir Negatif ACK veya Seçimli ACK gibi bir hata kurtarma kontrol paketini komşu düğümlere göndererek hatalı segmentin kurtarılmasını sağlamalıdır. Taşıma Katman Protokollerinde bulunan bir diğer modül ise “Tıkanıklık Modülü” dür. Bu modül, ağdaki herhangi bir düğümün tıkanıklığa maruz kalıp kalmadığını tespit edilmesini ve eğer gerekirse tıkanıklığın olduğu bölgelerdeki trafiğin yavaşlatılarak tıkanıklığın giderilmesini sağlamaktadır. Taşıma katmanında bulunan bir diğer modül ise “Öncelik Modülü” dür. Bu modül ise

kuyruğa eklenen paketlerin önceliklerinin organize edimesini sağlar. Önceliği yüksek olan paketlerin daha kısa süre kuyrukta bekleyerek zaman kaybetmeden gönderilmesine yardımcı olur.

Literatürde, KAA'lar için önerilen taşıma katman protokollerinin bazılarında bu üç modül bulunurken bazılarında ise sadece birisi bulunabilmektedir. Örneğin, sadece tıkanıklığın tespiti ve giderilmesi için önerilen protokollerin yanında sadece paketlerin güvenilir şekilde hedeflere ulaştırılmasını garanti altına alan protokoller mevcuttur.

### **3.3. KAA'lardaki Taşıma Katmanı Protokolleri Tasarım Ölçütleri**

KAA'lar için tasarlanan taşıma katman protokollerinin görevlerinin başında tıkanıklık ve paket güvenilirliğinin sağlanması gelmektedir. Taşıma katman protokollerinin bunları sağlayabilmesi için bazı durumlarda diğer katmanlarla beraber uyum içinde çalışması gerekir. Örneğin, tıkanıklık problemini aşmak için taşıma ve yönlendirme katmanlarının ortaklaşa çalışması gerekirken enerji korunumunu için fiziksel, veribağı, yönlendirme ve taşıma katmanlarının birlikte sorunu ele alması gerekmektedir. Katmanlar arasındaki bu uyum sayesinde daha verimli bir paket iletimi sağlanmış olur. Bunun dışında, başarılı bir taşıma katmanından beklenen başka ölçütler bulunmaktadır. Bu alt başlıkta taşıma katman protokollerinin tasarımını etkileyen güvenilirlik, servis kalitesi, doğruluk ve enerji verimliliği gibi tasarım ölçütleri [69] detaylandırılacaktır.

#### **3.3.1. Güvenilirlik**

KAA'larda güvenilirlik, paketlerin zamanında ve başarılı bir şekilde kaynaktan hedefe ulaştırılması anlamına gelmektedir. KAA'larda farklı uygulamaların ihtiyaçlarına göre iki farklı türde güvenilirlik gerekebilmektedir. Bunlardan birincisi "Paket Güvenilirliği", diğeri ise "Olay Güvenirliği"dir. Paket güvenilirliğinde uygulamalar paket kayıplarına karşı duyarlıdır ve tüm paketlerin başarılı bir şekilde iletilmesine ihtiyaç duyarlar. Olay güvenilirliğinde ise uygulamalar tüm paketlerinin başarılı şekilde iletilmesinden ziyade olayların başarılı şekilde tespitine ihtiyaç

duyarlar [3]. Buna ek olarak, S.J. Park ve arkadaşları [74] hedef bağlantılı güvenilirlikten bahsetmektedir. Sonuç olarak, tasarlanacak taşıma katmanı protokolü, uygulamaların ya da kullanıcıların beklediği güvenilirlik türünü ve seviyesini sağlaması gerekir.

### **3.3.2. Servis kalitesi**

Geleneksel QoS(Quality of Service) ölçütleri, bant genişliği, gecikme ve paket kayıp oranlarını içerir. Kablosuz algılayıcı ağ yapılarının ihtiyaçlarına bağlı olarak, bu ölçütler ya da onların farklı türleri KAA'lar için kullanılabilir. Örneğin kablosuz algılayıcı ağlar bir hedefin izlenmesi için sürekli resim iletiminde kullanılabilir. Bu düğümler yüksek hızda veri akışı oluştururlar ve olay tabanlı uygulamalardan daha fazla bant genişliğine ihtiyaç duyarlar. Gecikme duyarlı uygulamalarda, veri paketinin zamanında teslim edilmesi önemlidir. Bu yüzden, bir taşıma katman protokolü uygulamaların gerektirdiği gecikme, paket kayıp miktarı, bant genişliği gibi servis kalite ölçütlerini sağlamalıdır.

### **3.3.3. Adillik**

Algılayıcı düğümler genellikle dış ortamda ve rasgele dağıtılmış olarak çalışmaktadırlar. Bu sebeple hedef düğümden (sink) uzaktaki düğümler için, elde ettikleri veri paketlerini hedef düğüme (sink) göndermeleri çok daha zordur. Bu eksikliği giderebilmek için, taşıma katman protokolleri, tüm algılayıcı düğümler arasında adaletli bant genişliğini sağlamaları gerekir. Böylece, hedef düğüm tüm algılayıcı düğümlerden veri paketlerini adil ve doğru bir şekilde elde edebilir.

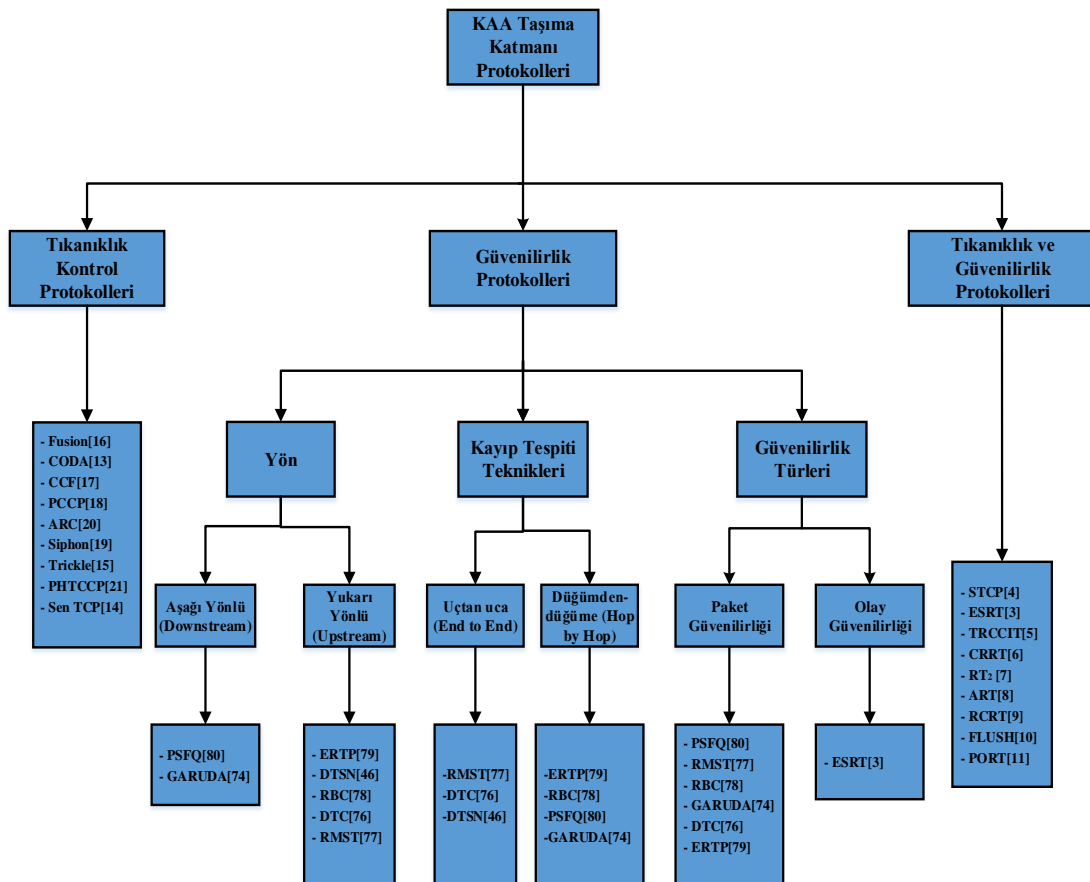
### **3.3.4. Enerji verimliliği**

Algılayıcı düğümler donanımsal kısıtlardan dolayı sınırlı enerjiye sahiptirler. Sonuç olarak, sistemin bekasını en üst seviyeye çıkarabilmek için taşıma katman protokollerinin yüksek enerji verimliliği sağlaması çok önemlidir. Genellikle düğümlerde harcanan gereksiz enerji, paket kayıplarından meydana gelmektedir. Paket kaybı da genel olarak bit hatalarından ya da tıkanıklıktan meydana

gelmektedir. Bu problemlerin uçtan-uca çözümü için taşıma katman protokolü görevlendirilmiştir. Ancak, bir taşıma katmanı hataları tespit ederken ve kayıp paketleri telafi ederken fazladan kontrol paketlerinin gönderimine yol açabilmektedir. Bu sebeple, geliştirilen yöntemin iyi bir şekilde değerlendirilmesi ve gereksiz kontrol paketlerinin engellenmesi gerekmektedir.

### 3.4. KAA'lar için Önerilen Taşıma Katman Protokolleri

KAA'lar için önerilen oldukça fazla taşıma katman protokolü bulunmaktadır. Bunlardan bazıları ya tıkanıklığı ya da güvenilirliği desteklerken, bazıları da her yapıyı da beraber desteklemektedir. Bu kapsamda, taşıma katman protokolleri genel olarak 3 gruba ayrılır. (1) Tıkanıklık kontrol protokolleri; (2) güvenilirlik protokolleri; (3) Tıkanıklık kontrolü ve güvenilirliği beraber destekleyen protokollerdir. Şekil 3.2'te bu protokoller ayrıntılı bir şekilde görülmektedir.



Şekil 3.2. KAA taşıma katman protokollerinin sınıflandırılması.

### 3.4.1. Tıkanıklık kontrolü protokolleri

KAA'ların en büyük sorunlardan birisi olan tıkanıklık problemi için çok sayıda tıkanıklık kontrol protokolü önerilmiştir. Tablo 3.1'de önerilen tıkanıklık kontrol protokolleri görülmektedir. Protokoller, tıkanıklığın tespiti, tıkanıklık durumunun bildirimini ya da paketlerin gönderim oranının ayarlanması gibi farklı ölçütlere göre birbirlerinden farklılık göstermektedir.

Tablo 3.1. Tıkanıklık kontrol protokolleri [75]

Protokoller	Tıkanıklık Tespiti	Tıkanıklık Bildirimi	Tıkanıklıktan Kurtulma Tekniği
<b>Fusion</b>	Kuyruk Uzunluğu	Saklı (Implicit)	Düğümden-düğüm oran ayarlanmasını durdurup başlatarak
<b>CODA</b>	Kuyruk Uzunluğu ve kanal durumu	Açık (Explicit)	Uçtan-uca oran ayarlanması
<b>CCF</b>	Paket Ulaşım Zamanı	Saklı (Implicit)	Tam düğümden-düğüm oran ayarlanması
<b>PCCP</b>	Paket Ulaşım Zamanı	Saklı (Implicit)	Tam düğümden-düğüm oran ayarlanması
<b>ARC</b>	Paket ilerleme Durumu	Saklı (Implicit)	Düğümden-düğüm oran ayarlanması
<b>Siphon</b>	Kuyruk Uzunluğu ve Uygulama Alanı	Uygun değil	Paket trafiği yönlendirme
<b>Trickle</b>	Alınan Paketler Sayısı	Uygun değil	Nazik Dedikodu (Polite Gossip)
<b>PHTCCP</b>	Paket ulaşım zamanı	Saklı (Implicit)	Paket zamanlaması
<b>SenTCP</b>	Paket ulaşım zamanı, bellek durumu	Açık (Explicit)	Tam düğümden-düğüm oran ayarlanması
<b>HCCP</b>	Paket teslim oranı, bellek durumu	Açık (Explicit)	Tam düğümden-düğüm oran ayarlanması

#### 3.4.1.1. Fusion

Fusion, KAA'larda yukarı yönlü veri trafiği için tasarlanan düğümden-düğüm tıkanıklık kontrol protokolüdür. Fusion protokolünde, tıkanıklık tespiti kuyruk uzunluğuna bağlı olarak tespit edilir ve her giden paketin CN bitinin ayarlanmasıyla tıkanıklık bildirimi sağlanır. Tıkanıklık meydana gelen düğümün komşuları CN bitinin yayılımı ile bundan haberdar olurlar. Düğüm CN biti alınca, komşu düğüm

tıkanıklık meydana gelen düğüme paket gönderimini iptal eder. Bu durum bağlantı kullanımını ve adilliği olumsuz yönde etkileyebilir [16].

#### 3.4.1.2. CODA

CODA (Congestion Detection and Avoidance) protokolü [13], tıkanıklığı tampon belleğin doluluk durumu ve kablosuz kanal yükü ile tespit eder. Bunu bir mesaj aracılığıyla yukarı yöndeki düğümlere gönderir. Düğümler bu mesajı aldıktan sonra, yukarı yönlü düğümler kendi gönderim hızlarını azaltırlar. Diğer bir deyişle, yukarı yönlü komşu düğümler belli bir zaman içerisinde tıkanıklık olduğuna dair bir mesaj gelmezse doğrusal olarak gönderim değerlerini arttırlar. CODA, AIMD de olduğu gibi kapalı döngü uçtan-uca yaklaşımını kullanmaktadır. Bu durum özellikle yüksek veri gönderimi ve aralıklı kaynak senaryoları altında güvenilirliğin azalmasıyla sonuçlanabilir.

#### 3.4.1.3. CCF

CCF (Congestion Control and Fairness) [17], tıkanıklığı dolaylı yoldan paket gönderim zamanı ve tıkanıklık bildirim bilgisini kullanarak tespit eden taşıma katman protokolüdür. Özellikle, her ara düğüm ilk olarak, ölçülen bir üst düğümden alınan paketin bir sonraki düğüme iletim değeri olan paket ulaşım zamanı tabanlı kendi servis değeri olan  $r$ 'yi ortaya çıkartır. Daha sonra kendisinin alt düğümlerine gönderme değerini  $r/[N(i) + 1]$  formülüyle hesaplar ve bu bilgiyi  $i$  düğümünün alt düğümü olan  $N(i)$ 'den her giden paketin içerisine yükler. Olasılıklı seçim (PS) ve dönem tabanlı oransal seçim (EPS) gibi iki farklı zamanlama algoritması kullanan CCF, her ne kadar adilliği arttırsa da, paket teslim gecikmesini arttırabilir.

#### 3.4.1.4. PCCP

PCCP (Priority-based Congestion Control) [18] KAA'lar için önerilen tıkanıklık kontrol protokolüdür. PCCP'deki düğüm önceliği, uygulamaya bağlı olduğu varsayılır ve hedef düğüm tarafından güncellenebilir ya da ayarlanabilir. İlk olarak, düğümden-düğüme tıkanıklık kontrol protokolünde olduğu gibi, her ara düğümdeki

yerel tıkanıklık seviyesini tespit etmek için paket ulaşım zamanını kullanır. Bu değerler sadece tıkanıklık seviyesini ölçmez, aynı zamanda etkili bir şekilde paket kayıplarının oluşumu hakkında fikir vermeye yardımcı olur. İkinci olarak, PCCP tam tıkanıklık bildirim yapılarını kullanır ve kontrol mesajları yüzünden meydana gelen aşırı yüklenmelerden kaçınır. Üçüncü olarak, PCCP ölçülen tıkanıklık derecesine bağlı olarak, esnek öncelik oran kontrolünü sağlar. Bu esnek oran kontrolü yüksek çıkış oranı elde etmek için, bazı düğümleri izin verileden daha az trafiğe sahip oldukları zaman, kendi trafik değerlerini arttırarak daha fazla trafik oluşmasına izin verir.

#### 3.4.1.5. ARC

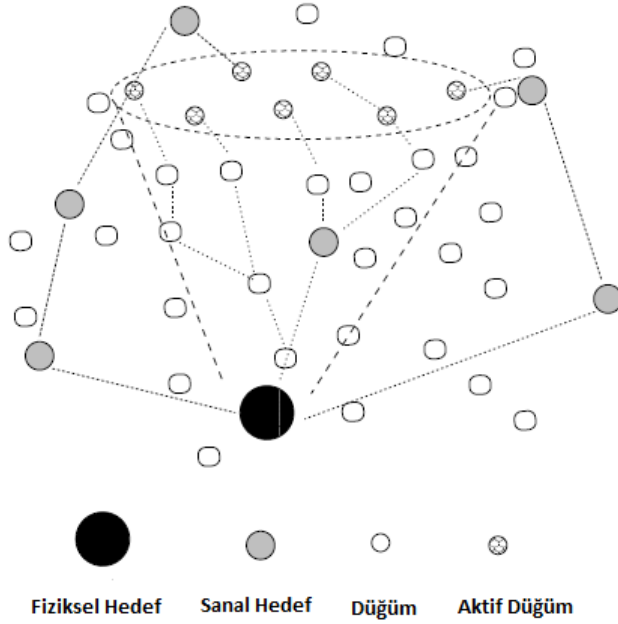
ARC (Adaptive Rate Control) [20] protokolü adaptif gönderim oranı sağlayan bir algoritmadır. ARC protokolünde, ara düğüm kendi komşu düğümünden başarılı bir paket geleceğini duyduğu zaman, kendi gönderim değeri olan  $\alpha$  değerini arttıracaktır. Aksi takdirde, gönderim değeri  $\beta$  etkisi ile çarpılacaktır.  $\beta$  değeri 0 ile 1 arasında bir değerdir. Kaynak ve geçiş trafiği arasındaki adilliği sağlayabilmek için, her düğüm kaynak trafiğini ve ulaşım trafiğini düzenlemek için bağımsız bir değer kullanır. Ulaşım trafiği  $\alpha$  ve  $\beta$  değerlerinden daha büyük bir değere ayarlanır. ARC'de ne açık tıkanıklık tespiti ne de açık tıkanıklık bildirim vardır. Kontrol mesajlarını kullanmaktan kaçınır.

#### 3.4.1.6. Siphon

Siphon[19], çok radyolu sanal hedefler (VS) kullanarak, yukarı yönlü trafik yükünü yöneten bir taşıma katman protokolüdür. Sanal hedefler en az iki farklı radyo arayüzü ile desteklenir. IEEE 802.11'de olduğu gibi, bir tanesi düşük güçlü radyo, diğeri ise uzun mesafeli radyodur. Düşük güçlü radyo, algılayıcı düğümleri birbirine bağlamak için kullanılır. Diğeri güçlü olan radyo ise, diğeri sanal hedefleri birbirine bağlamak ya da internete geçit yolu desteği sağlayan fiziksel hedefe bağlanmak için kullanılır. CODA'da kullanılan tıkanıklık tespitine ek olarak, Siphon aynı zamanda gönderme etkisi (post-facto) yapısını kullanır. Bu yapıda, fiziksel hedef hissedilen uygulama alanını ölçer ve buna bağlı olarak tıkanıklığı anlar. Tıkanıklık meydana



geldiğinde Siphon, trafik yönlendirmesini algılayıcı düğümlerden sanal hedeflere yönlendirir. Sanal hedefler gerçekten trafik tıkanıklığı altında veri teslimi için etkili kısayollar sağlar. Sonuç olarak, tıkanıklık hızlı bir şekilde azaltılır. Şekil 3.3'te Siphon protokolünün yerleşim yapısı görünmektedir.



Şekil 3.3. Siphon taşıma katman protokolü yerleşim yapısı [19]

### 3.4.1.7. Trickle

Trickle [15], 2004 yılında Philip Levis ve arkadaşları tarafından geliştirilen KAA'lardaki kod güncellemelerinin bakımı ve yayılımı için geliştirilmiş kontrollü yayın protokolüdür. Bundan dolayı, aşağı yönlü olarak çalışır. Trickle, komşu düğümler arasındaki kod ve veri değişimi için yerel nazik dedikodu (Polite Gossip) yöntemini kullanır. Nazik dedikodu ile her düğüm belli aralıklarla kendi verilerinin özetinin yayınlanmasını sağlar. Böylece, tüm düğümler aynı anda yeni güncelleme değerlerini alırlar. Her zaman aralığında eğer, komşu düğümlerinden aldığı aynı verinin sayısı eşik değerini aşıyorsa, düğüm nazik bir şekilde kendi yayını bastırır. Diğer bir deyişle, eğer düğüm yeni bir kod ya da veri alıyorsa, düğümler kendi yayın zamanını kısaltabilir. Trickle bağımsız bir şekilde her düğümde çalışır ve ek kontrol mesajı gerektirmez.

### 3.4.1.8. PHTCCP

PHTCCP (Prioritized Heterogeneous Traffic Congestion Control Protocol) [21] protokolü Monowar, Rahman, Pathan ve Hong tarafından 2008 yılında tasarlanmıştır. PHTCCP, düğümden-düğüme tıkanıklık metotunu kullanan tıkanıklık kontrol protokolüdür. Tıkanıklık meydana geldiği zaman, paket iletim oranını düzenler. Bazı düğümler pasif ya da uyku modunda olduğu zaman da bile, bağlantıların verimli bir şekilde kullanılmasına izin verir.

PHTCCP paket zamanlaması için WFQ (Ağırlıklı adil kuyruklama) yöntemini kullanır. WFQ, değişik zamanlama önceliklerine izin veren veri paketi zamanlama tekniğidir. PHTCCP, tıkanıklık seviyesini tespit etmek için paket servis oranı  $R(i)$  olarak adlandırılan parametreyi kullanır.  $R(i)$  değeri, paket servis oranı  $(R_s^i)$  değerinin paket zamanlama değeri  $(R_{sch}^i)$ 'ne bölünmesiyle hesaplanır. Her düğüm, bazı bilgileri diğer düğümlere paket başlığı içinde iletir. Bu bilgiler, paket zamanlama oranı, alt düğümlerin toplam sayısı, belirlenen (t) zamanı sırasındaki aktif alt düğümlerin sayısı ve aktif alt düğümlerin ortalama kuyruk uzunluğudur. Böylece, tüm düğümler tıkanıklık bilgisinden haberdar olurlar.

### 3.4.1.9. SenTCP

SenTCP [14] protokolü Wang, Sohrawy ve Li tarafından 2005 yılında tasarlanmıştır. SenTCP yukarı yönlü veri gönderimi için düğümden-düğüme metotunu kullanır. SenTCP paket ulaşım zamanı ve tampon bellek durumu parametrelerini kullanarak her ara düğümdeki tıkanıklık seviyesini ölçer. Algılayıcı düğüm, tıkanıklık tespit ettiğinde bu bilgiyi komşu düğümlere gönderir. Gönderilen paket içerisinde, tıkanıklık seviye bilgisi ve tampon bellek durum göstergesi bilgileri vardır. Diğer algılayıcı düğümler tıkanıklık mesajını aldığı zaman, kendi paket gönderim değerini düzenlerler. Böylece, ağ tıkanıklık durumu düzeltilmiş olur ve paket kayıpları azaltılmış olur. SenTCP protokolü tıkanıklık kontrolü sağlamasına karşın, güvenilirlik yapısına sahip değildir.

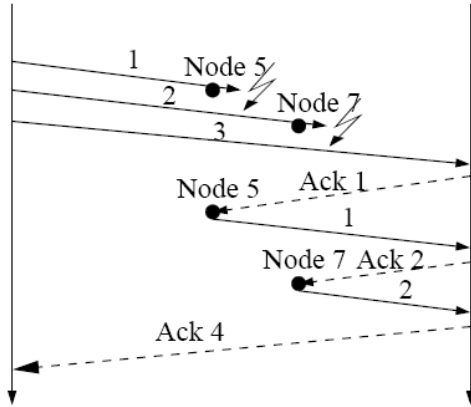


### 3.4.2.1. DTSN

DTSN (Distributed Transport for Sensor Networks) [46] KAA'larda yukarı yönlü güvenilirliği garanti eden uçtan-uca hata tespiti tekniğini kullanan güvenilir taşıma katmanı protokolüdür. DTSN protokolünde, kaynak düğüm paket maliyetini en aza indirebilmek için kontrol ve veri paketlerini kullanarak hata tespiti ve kurtarma işlemini kontrol eder. Temel hata tespiti ve kurtarma algoritması için Seçimli ACK (SACK) yapısını kullanır. SACK tekniğinden pozitif ve negatif ACK'lar beraber kullanılmaktadır. DTSN, bu yapıları kullanarak, bir gönderimdeki tüm paketler kaybolduğunda ya da gelen paketin sıra numarasında yanlışlık olduğunda bunu kolaylıkla tespit edebilir. Ara düğümler, uçtan-uca iletim güvenilirliğini daha da arttırmak için gelen paketleri kendi belleğine kaydeder. Bu sayede, paketlerin ara düğümlerden kurtarılması daha da kolaylaşmaktadır.

### 3.4.2.2. DTC

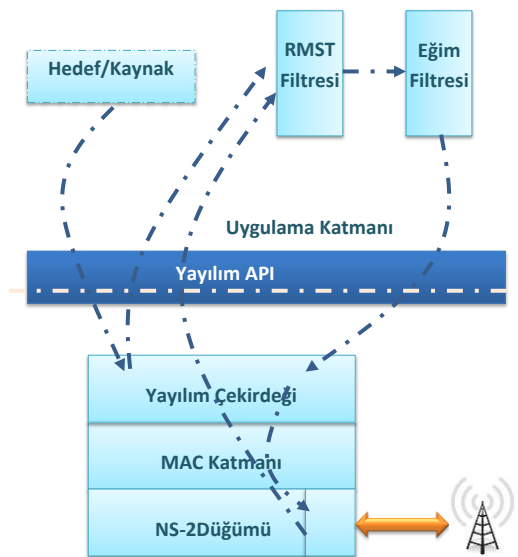
DTC protokolü (Distributed TCP Caching) [76] KAA'larda yukarı yönlü güvenilirliği garanti eden uçtan-uca hata kurtarma tekniğini kullanan güvenilir taşıma katmanı protokolüdür. DTC protokolü KAA'lardaki TCP performansını arttırmak için tasarlanmıştır. TCP [82] protokolünde bir paket bozulduğunda, alıcı ve verici arasındaki tüm yol boyunca tekrar iletimin gerçekleştirilmesi gereklidir. Bu da gereksiz paket maliyetine sebep olur ve gereksiz enerji tüketimine sebep olur. DTC protokolü bu sorunların üstesinden gelebilmesi için kaynaktan gönderilen paketleri ara düğümlerde belleğe kaydeder. Bu sayede hedef düğümde herhangi bir eksik tespit edildiğinde, kaynak düğüme gitmeye gerek kalmadan ara düğümlerden kolaylıkla kurtarılabilir. DTC, bu sayede paketlerin tekrar iletim sayılarını önemli ölçüde azaltır. Şekil 3.4'te DTC protokolünün çalışma algoritması görülmektedir.



Şekil 3.4. DTC protokolünün çalışma algoritması

### 3.4.2.3. RMST

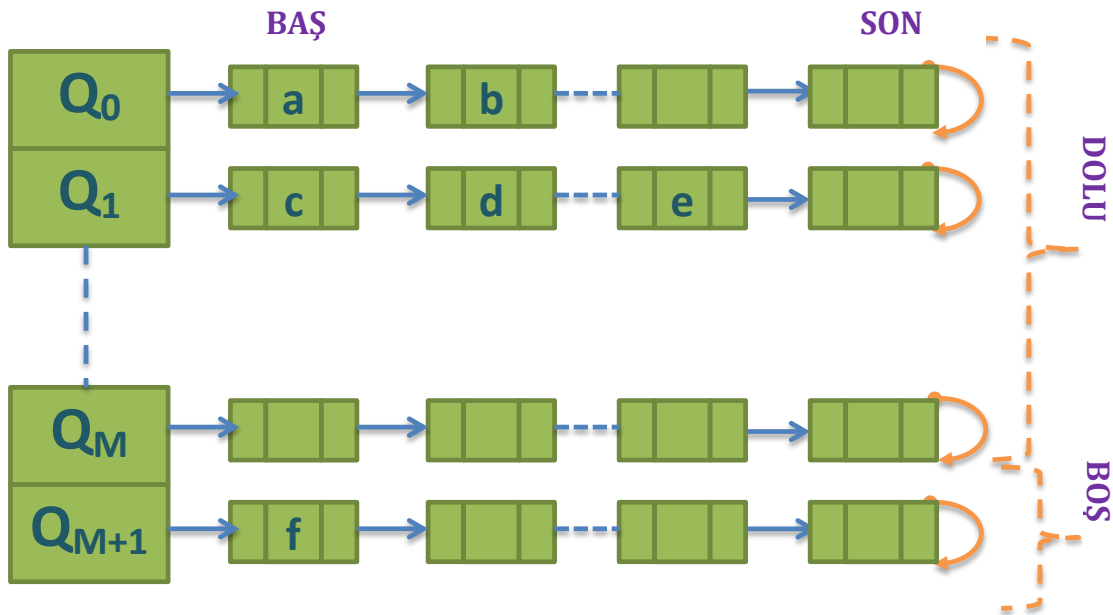
RMST (Reliable Data Transport in Sensor Networks), yukarı yönde her paketin başarılı bir şekilde iletimini garanti eden taşıma katman protokolüdür [77]. Kayıp tespiti ve bildirimini için seçici NACK ve zaman sürgülü yapıyı beraber kullanır. RMST protokolünde ara düğümler tekrar iletim için segmentleri düğüm içerisinde depolama ya da depolamama modunu kullanabilir. Depolama modunda, kayıp paketler ara düğümlerden, düğümden-düğüme hata tespiti ve kayıp kurtarma yapısı ile kurtarılabilir. Eğer bir ara düğüm kayıp paket bulamaz ise ya da depolamama modunda çalışıyorsa, düğüm alınan NACK paketini kaynak düğüme kadar iletir. Şekil 3.5'te RMST protokol yapısı görülmektedir.



Şekil 3.5. RMST protokol yapısı [77]

### 3.4.2.4. RBC

RBC (Reliable Bursty Convergecast) düğümden-düğümüne IACK (Implicit ACK) tabanlı çalışan, her ara düğümde paketleri saklayan güvenilir taşıma katmanı protokolüdür [78]. RBC, güvenilirliği garanti etmeyi ve gerçek zamanlı yukarı yönlü paket gönderimini hedefler. RBC protokolü IACK yapısı ve penceresiz blok onay yapısını kullanır. Bu yapılar daha fazla güvenilirlik ve enerji verimliliği için paket kaybını tespit eder ve bildirir. Alıcı düğüm başarılı bir şekilde paketlerin bilgisini taşır ve gönderici düğüm bu bilgiyi dinleyebilir ve eğer gerekliyse tekrar iletimi başlatır. Eğer, gönderici düğüm, belli bir zaman içinde beklenen bilgiyi almazsa, iletilen her paket için ayarlanan zamanlayıcı sonlandırılır ve gönderici düğüm, düğümden-düğümüne tekrar iletimi başlatır. RBC protokolünde, tekrar iletim zamanlayıcısı değeri bir sonraki düğümdeki kuyruk uzunluğuna bağlıdır. Şekil 3.6'da RBC protokolünün çalışma mantığı görülmektedir.

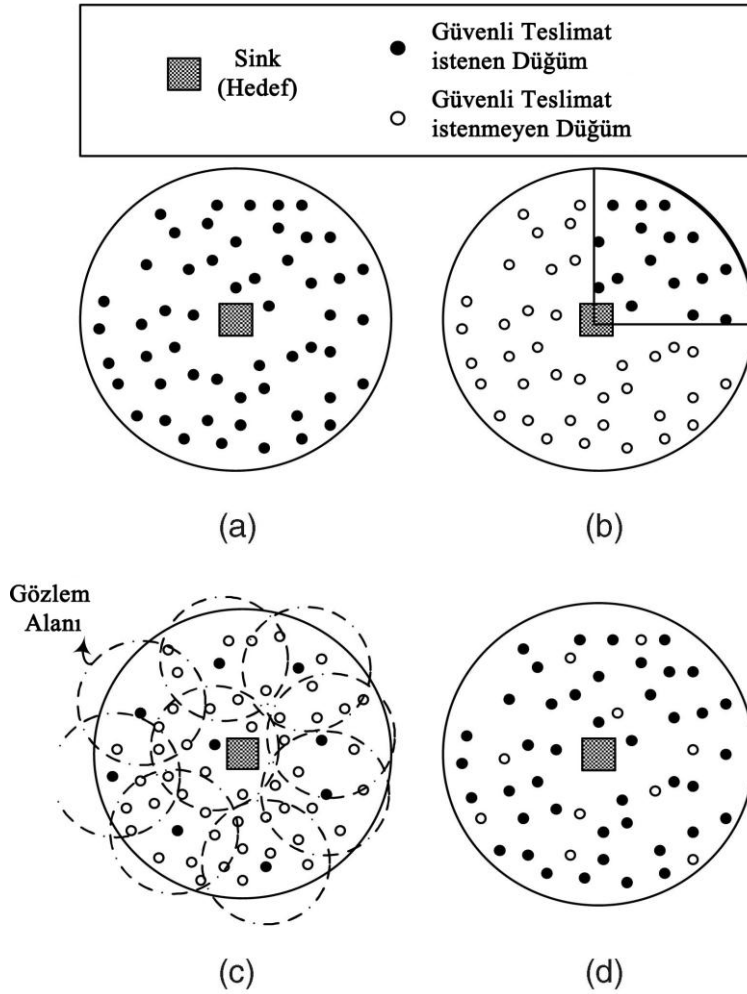


Şekil 3.6. RBC protokolü çalışma mantığı [78]

### 3.4.2.5. GARUDA

GARUDA, KAA'larda aşağı yönlü güvenilirliği garanti eden güvenilir taşıma katmanı protokolüdür [74]. GARUDA, tekrar iletim için iki sıra mimarisini kurar. Hedef düğümden 3-atlama uzaktaki algılayıcı düğümler, eğer onların komşularının

hiçbiri çekirdek düğüm değilse, çekirdek algılayıcı düğüm olarak seçilirler. Çekirdek algılayıcı düğümler ilk sırayı ve diğerleri ise ikinci sırayı oluşturur. Her çekirdek olmayan algılayıcı düğüm kendine kayıp paketleri telafi edebileceği en yakın çekirdek düğümü seçer. GARUDA, kayıp tespiti ve bildirimini için NACK kontrol mesajını kullanır. Şekil 3.7’de GARUDA protokolünün güvenlik şemaları görülmektedir.



Şekil 3.7. GARUDA güvenlik şemaları a) Tüm düğümlere güvenli teslim b) Altbölgeye güvenli teslim c) Gözlem alanını kapsayan en az düğümle güvenli teslim d) Düğümlerin %80'ine güvenli teslim [74]

### 3.4.2.6. PSFQ

PSFQ (Pump Slowly, Fetch Quickly) KAA'lar için önerilen aşağı yönlü ve düğümden-düğüme hata tespiti ve kayıp kurtarma yapısını kullanan taşıma katman

protokolüdür [80]. Düşük hızla paketleri kaynak düğümden algılayıcı düğümlere güvenli bir şekilde dağıtmayı hedefler. PSFQ, düğümden-düğüme hata tespiti ve kayıp kurtarma yapısını kullanır. Bu sayede, eksik segmentleri komşu düğümlerden hızlı bir şekilde telafi edebilir. Pompalama (pump) işlemi, bulup getirme (fetch) işlemi ve raporlama (report) işlemi olmak üzere üç bileşenden oluşur. İlk olarak, kaynak düğüm, yavaşça ve belli aralıklarla paketleri diğer düğümlere gönderir. PSFQ protokolünde yavaş segment gönderimi yüzünden büyük gecikmeler ortaya çıkabilir. İkinci olarak, düğüm eğer paket segmentleri arasında eksik tespit ederse, bulup getirme moduna girer ve kayıp segmenti alabilmek için daha önceden belirlenen eşik değerine ulaşana kadar ters yönde NACK mesajı gönderir. PSFQ, NACK tabanlı hata tespiti tekniğini kullandığı için, tek bir segmentin kaybını tespit edemez. Üçüncü olarak, hedef düğüm algılayıcı düğümlere paket taşıma durumunu geri bildirim bilgisi olarak gönderebilir.

#### **3.4.2.7. E RTP**

E RTP (Energy-efficient and Reliable Transport Protocol) [79] protokolü KAA'larda yukarı yönlü güvenilirliği garanti eden düğümden-düğüme hata kurtarma tekniğini kullanan güvenilir taşıma katmanı protokolüdür. E RTP bir ya da daha fazla algılayıcı düğümlerden hedef düğüme iletilmesi gereken bilgileri, enerji verimliliğini dikkate alarak göndermek için kullanılır. E RTP, güvenilirliği sağlamak için, daha önceden belirlenmiş olan hedefe teslim edilecek veri paketlerinin sayısını garanti eden istatistiksel güvenilirlik ölçütünü kullanır. Her düğümden, dinamik olarak oluşan tekrar iletim sayısını kontrol eder. Bu sayede, önemli ölçüde enerji korunumu sağlar.

#### **3.4.3. KAA'larda hem tıkanıklığı hem de güvenilirliği destekleyen protokoller**

KAA'larda sadece tıkanıklık kontrolü veya güvenilirliği destekleyen taşıma katman protokolleri olduğu gibi bu yapıların ikisini beraber destekleyen taşıma katman protokolleri de vardır [25]. Bunlar STCP [4], ESRT [3], TRCCIT [5], CRRT [6], RT<sup>2</sup> [7], ART [8], RCRT [9], Flush [10], PORT [11], SRCP [12]'dir. Tablo 3.3'te tıkanıklık ve güvenilirlik protokollerinin ayrıntıları gösterilmektedir.



Tablo 3.3. Tıkanıklık ve güvenilirlik protokolleri

Protokoller	Tıkanıklık Tespiti	Tıkanıklık Bildirimi	Tıkanıklıktan Kurtulma Tekniği	Güvenilirlik Yönü	Güvenilirlik seviyesi	Kayıp tespiti ve geri alımı	Kayıp tespiti ve bildirimi
<b>STCP</b>	Kuyruk Doluluğu	Saklı (Implicit)	Gönderim oranı ayarlama	Yukarı	Paket	Uçtan - Uca	NACK, eACK
<b>ESRT</b>	Kuyruk Doluluğu	Saklı (Implicit)	Gönderim oranı ayarlama	Yukarı	Olay	-	-
<b>TRCCIT</b>	Paket oranı	Saklı (Implicit)	Gönderim oranı ayarlama	Yukarı	Paket	Düğümünden- düğümüne	iACK, eACK
<b>CRRT</b>	Kuyruk Doluluğu, Paket oranı	Saklı (Implicit)	Gönderim oranı ayarlama	Yukarı	Paket	Uçtan - Uca	NACK, MAC
<b>RT<sup>2</sup></b>	Düğüm gecikmesi, Kuyruk doluluğu	Saklı (Implicit)	Gönderim oranı ayarlama	Yukarı	Paket	Düğümünden- düğümüne	SACK
<b>ART</b>	Çekirdek düğümlerden alınan ACK	Saklı (Implicit)	Trafığı azaltma	Aşağı/yukarı	Olay	Uçtan - Uca	NACK, eACK
<b>RCRT</b>	Kayıp geri alım zamanı	Saklı (Implicit)	Gönderim oranı ayarlama	Yukarı	Paket	Uçtan - Uca	NACK,
<b>FLUSH</b>	Kuyruk Doluluğu	Saklı (Implicit)	Gönderim oranı ayarlama	Yukarı	Paket	Uçtan - Uca	NACK,
<b>PORT</b>	Düğüm maliyeti, bağlantı kayıp oranı	Saklı (Implicit)	Gönderim oranı ayarlama, Trafik yönlendirme	Yukarı	Olay	-	-
<b>SRCP</b>	Düğüm gecikmesi,	Açık (Explicit)	Gönderim oranı ayarlama	Yukarı	Paket	Düğümünden- düğümüne	SACK

### 3.4.3.1. STCP

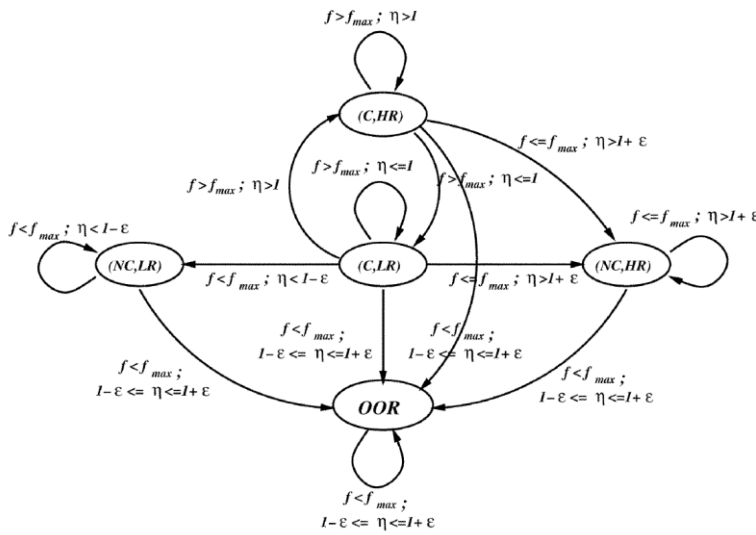
STCP (Sensor Transmission Control Protocol) tıkanıklık kontrolünü ve güvenilirliği beraber destekleyen, uçtan-uca yukarı yönlü çalışan bir taşıma katmanı protokolüdür [4]. STCP protokolünde, ara düğüm, kuyruk uzunluğuna bağlı tıkanıklığı tespit eden algoritma gibi olan rasgele erken tespit (RED) yapısını çalıştırır. Eğer, yerel tıkanıklık meydana gelirse, ara algılayıcı düğüm, gönderilecek normal veri paketlerindeki bit değerini ayarlar. Hedef düğüm bu paketlerdeki bit değerini alır ve kaynak düğümlere, ACK ve NACK mesajlarını ara düğümleri kullanarak bu paketleri göndermesini bildirir. Kaynak düğüm başka bir yol seçebilir ya da ağdaki tıkanıklığı azaltmak için gönderim değerlerini azaltır. STCP protokolündeki yeniliklerden biri de uçtan-uca tekrar iletimlerle, uygulamalar için kontrollü, değişken güvenilirliği sağlamasıdır. Bir başka deyişle, STCP, değişik uygulamalar için değişik yapıları tek bir protokol içinde kullanan ve bunları değişik taşıma fonksiyonlarıyla birleştiren bir protokoldür. Sürekli paket üreten uygulamalar için, hedef düğüm şu anki güvenilirliği ölçer. Eğer ölçülen değer istenen güvenilirliğin altındaysa ve beklenen paketler beklenen zaman içinde ulaşmamışsa, hedef düğüm tekrar iletim için NACK mesajı gönderir. Olaya bağlı uygulamalar için, hedef düğüm paketlerin kaybolduğunu ya da başarılı bir şekilde alındığını kaynak düğüme bildirmek için ACK mesajını kullanır. Eğer istenilen güvenilirlik sağlanmamışsa, daha sonra kaynak düğüm kayıp paketleri tekrar iletebilir.

### 3.4.3.2. ESRT

ESRT (Event-to-Sink Reliable Transport) [3] uçtan-uca gönderim oranını düzenleyerek güvenilirliği sağlamayı hedefleyen yukarı yönlü taşıma katmanı protokolüdür. ESRT protokolünde, her algılayıcı düğümdeki gönderim oranı hedef düğümden alınan o anki güvenilirlik seviyesine bağlıdır.

ESRT gerçekte tıkanıklık kontrolünü ve güvenilirliği beraber destekleyen ilk taşıma katmanı protokolüdür. Her ara düğüm, tampon bellek durumunu kontrol ederek tıkanıklığı tespit eder ve hedef düğüme gönderilecek paketlerin başındaki CN (Congestion Notification) biti ayarlanır. Böylece, hedef düğüm ağın durumu ile ilgili

bilgi sahibi olur. AIMD algoritmasında olduğu gibi, ESRT o anki ağ durumu, olay güvenilirliği ve istenen olay güvenilirliğine göre uygun raporlama frekansını hesaplar. Daha sonra hesaplanan frekans değeri yüksek güçte bir-atlamalık kanal üzerinden diğer düğümlere bildirilir. Algılayıcı düğümler aldıkları rapor frekansına göre gönderme oranlarını düzenlerler. Bunun 2 dezavantajı vardır. (1) Tekrar iletim tabanlı kayıp kurtarma ile karşılaştırıldığında enerji yönünden verimsizdir. (2) Tıkanıklık meydana geldiğinde istenen olay güvenilirliği garanti edilemez. Buna rağmen kayıp kurtarma hala güvenilirliği iyileştirebilir. Şekil 3.8’de ESRT protokolü durum modeli ve geçişleri yapıları görülmektedir.



Şekil 3.8. ESRT protokolü durum modeli ve geçişleri [3]

### 3.4.3.3. TRCCIT

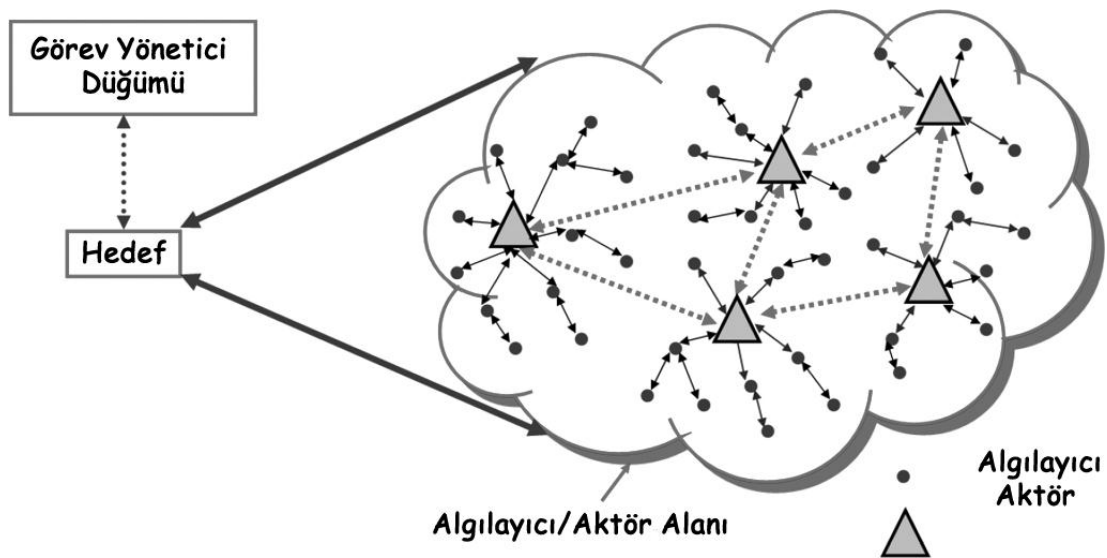
TRCCIT (Tunable reliability with Congestion Control for Information Transport) [5], KAA'lar için tasarlanmış, taşıma katmanındaki tıkanıklık bilgisini kullanarak, ayarlanabilir bir güvenilirlik sağlayan bir taşıma katman protokolüdür. TRCCIT gereksiz paket akışını durdurarak, tekrar iletimleri ayarlayarak ve ağ durumunu geliştirerek, istenen uygulama güvenilirliğini sağlar. Taşıma bilgisinin güvenilirliği, yerel tekrar iletim zaman yönetimi yardımıyla yapılan HACK (Hybrid Acknowledgement) yapısı ile sağlanır. TRCCIT protokolü tıkanıklığı azaltmak için paketleri çoklu yol üzerinden gönderir ve verimli bir şekilde paket akışını sağlar. Bu sayede istenilen güvenilirlikte sağlanmış olur.

### 3.4.3.4. CRRT

CRRT (Congestion-Aware and Rate-Controlled Reliable Transport) [6], tıkanıklık kontrolü sağlayan, en uygun paket gönderimi ve adilliği sağlamak için her düğümün gönderim oranını bireysel olarak kontrol eden, güvenilir bir taşıma katman protokolüdür. CRRT, çok yüksek iletişim verimliliği ve MAC tekrar iletim yapısının kullanılmasına bağlı olarak enerji verimliliği sağlar. Aynı zamanda, diğer protokollere göre daha iyi bir veri akışı sağlamaktadır. CRRT ağ içinde çoklu hedef düğümlerini ve çoklu benzer uygulamaları destekler. Çok hedefli ağlar için, hedefler bireysel olarak kaynakların gönderim oranını değiştirebilirler ve bu sayede adilliği de sağlamış olurlar.

### 3.4.3.5. RT<sup>2</sup>

RT<sup>2</sup> (A Real-Time and Reliable Transport) [7], Kablosuz Aktör ve Sensör Ağlar için Güvenirlik ve Tıkanık kontrolü sağlayan bir protokoldür. RT<sup>2</sup> nin amacı gözlemlenen ortamdaki sezilen olay bilgilerini aktör düğümlere güvenli ve beraber bir şekilde iletmektir ve zamanında doğru karşılık vermesidir. RT<sup>2</sup> kendi ayarlarını dinamik bir şekilde düzenler. Bu sayede, uygulamaların KAA'lardaki aktörlerin işbirliği ile doğru zamanda doğru sonuçlar elde etmesini sağlar. Şekil 3.9'da RT<sup>2</sup> protokolünün mimarisi görünmektedir.



Şekil 3.9. WSAAN mimarisinin gösterim şeması

### 3.4.3.6. ART

ART [8], KAA'larda iki yönlü uçtan-uca güvenilirliği sağlayan bir taşıma katman protokolüdür. ART protokolü esas düğümleri (essential nodes) seçmek için enerji duyarlı düğüm sınıflandırma algoritması olan Greedy [81] algoritmasını kullanır. Bu algortmada düğümlerin pil seviyelerine bakılarak, esas düğüm olup olamayacağına karar verilir. Bu ağdaki düğümlerin sahip olduğu enerjinin esnek ve verimli bir şekilde kullanılabilmesi için çok önemlidir.

ART protokolünde güvenilir olay ve sorgu transferi, kontrol mesajları ve tekrar iletimlere rağmen düşük maliyetle, hedef düğüm ve esas düğümler arasında başarılı bir şekilde gerçekleştirilir. İkincisi, NACK yapısı güvenilir paket teslimatını yaparken, olay transferi için basit bir ACK yapısı kullanılır. Üçüncü olarak, ART dağıtık tıkanıklık kontrol yapısına sahiptir. Bu yapıda tıkanıklık, esas düğüm olmayan düğümlerdeki veri trafiğininin düzenlenmesiyle ortadan kaldırılır.

### 3.4.3.7. RCRT

RCRT (Rate-Controlled Reliable Transport) [9] KAA'lar için tasarlanmış güvenilirlik destekli taşıma katman protokolüdür. RCRT aynı zamanda gönderim oranını esnek bir şekilde ayarlamak ve ağdaki veri trafiğini daha iyi kontrol edebilmek için hedef düğümde tıkanıklık kontrolünü de gerçekleştirir. Bu durumun iki avantajı vardır. Birincisi hedef düğüm, en uygun paket gönderim oranına karar verir ve böylece hedef düğüm ağın genel yapısını daha iyi bildiği için, verimli bir paket gönderimi sağlanmış olur. İkinci olarak, farklı türdeki uygulamaları destekler ve ağ dinamiklerini en uygun şekilde sağlar. RCRT protokolü kayıp paketleri kurtarabilmek için uçtan-uca kayıp hata tespiti ve kurtarma tekniğini kullanır.

### 3.4.3.8. Flush

Flush [10] KAA'lar için güvenilirlik ve tıkanıklık kontrolü destekli, yüksek oranda veri transferi sağlayan bir protokoldür. Flush protokolü uçtan-uca güvenilirliği sağlar, paket gecikmesini azaltır ve zamana bağlı olarak ağ durumunu değiştirir. Bu

özellikleri uçtan-uca ACK paketi ve her düğüm boyunca veri akışı sağlayan oran kontrol algoritması kullanarak gerçekleştirir.

Flush güvenilirliği garanti etmek için ve yüksek paket gönderim oranı sağlamak için birkaç farklı teknik kullanır. İlk olarak, hata tespiti ve kayıp kurtarma için ACK ve SACK yapılarını kullanır. İkinci olarak, veri paketleri hedef düğüme daha önceden belirlenen yol üzerinden gereksiz tıkanıklıktan kaçınabilecek, dikkatli bir şekilde seçilmiş gönderim oranıyla gönderir.

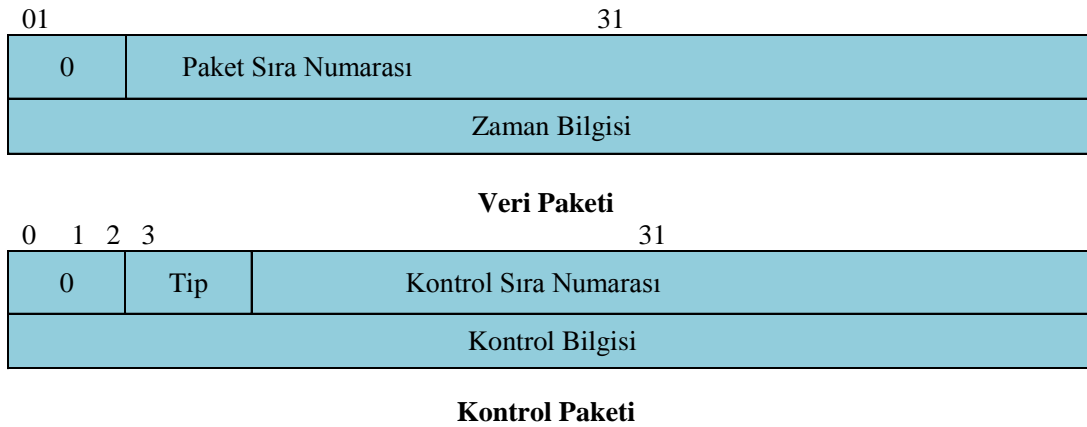
### **3.4.3.9. PORT**

PORT (Price-Oriented Reliable Transport) [11] Zhou ve arkadaşları tarafından KAA'lar için tasarlanmış, kaynak düğümlerden hedef düğüme veri iletimini gerçekleştiren taşıma katman protokolüdür. PORT protokolü, hedef düğümün zor durumlar altında güvenilir paket iletimi elde edebilmesi ve enerji tüketimini en aza indirebilmesi için iki farklı şema kullanır. Bunlardan ilki, her kaynak düğümün en uygun raporlama oranını ve kaynaktan hedefe doğru tüm düğümlerin enerji tüketimi bilgilerinin geri bildirimini sağlayan, hedef düğümün uygulama tabanlı uygunlaştırma yaklaşımıdır. İkincisi, aşağı yönlü paket iletim durumu bilgisinin geri bildirimine göre, ağdaki diğer tüm düğümler için, en uygun yönlendirme şemasının sağlanmasıdır. PORT protokolü, tıkanıklık ve zayıf bağlantı gibi sebeplerden dolayı oluşan paket kayıp bilgilerini hedef düğüme bildirir. Böylece bu kayıplardan dolayı olabilecek ihtimallere göre ağ dinamiği en uygun hale getirilir.

### **3.4.3.10. SRCP**

SRCP (Sensor Reliability and Congestion Control Protocol) [12] Khan ve diğerleri tarafından 2010 yılında tasarlanmıştır. SRCP, hem güvenilirliği hem de tıkanıklığı destekleyen ve aynı zamanda paket gönderim oranını ayarlayabilen, uçtan-uca tıkanıklık kontrol protokolüdür. Şekil 3.10'da SRCP protokolünün paket başlığı yapısı görünmektedir. SRCP bağlantıyı bir istek göndererek başlatır. Hedef düğüm RTT değerini hesaplar ve bu bilgiyi gönderilecek olan paketin başlığına ekler. Daha sonra, bu paketi geri gönderir. Paketler düğüme ulaştığı zaman, düğüm kendi

belleğinde depolanmış olan segmentleri göndermeye başlar. Paket ulaşım zamanı değeri 0.4 sn aralıklarla güncellenir. Veri paketleri alındığı zaman, hedef düğüm paketlerin hedefe ulaşma zamanını hesaplar. Hedef düğüm paketlerin ulaşım zamanlarını bir diziye kaydeder. Eğer paket gönderim zamanlarında bir artma varsa, düğüm RTT değerlerini gönderir ve böylece düğümler RTT değerini günceller. Eğer paket dizi numarası beklenenden farklıysa, hedef düğümü kayıp segmentler konusunda bilgilendirir. Eğer hedef düğüm bu paketi alırsa, başlama evresini (start phase) başlatır ve kontrollü evreyi (controlled phase) tetikler.



Şekil 3.10. SRCP paket başlığı [12]

### 3.5. Sonuçlar

Bu bölümde, kablosuz algılayıcı ağlarda güvenilirliği sağlamaktan sorumlu olan taşıma katmanı hakkında bilgi verilmiştir. Ayrıca, hata tespiti ve kayıp telafisinde kullanılacak olan yöntemlerden bahsedilmiştir. Literatürde var olan taşıma katman protokolleri ve çalışma yöntemleri hakkında özet bilgiler sunulmuştur. Ancak KAA'larda paket güvenilirliğinin sağlanması, enerji korunumunun elde edilebilmesi için, düşük maliyetli ve başarılı çözüm yöntemlerinin geliştirilmesi gerektiği oldukça önemli bir eksiklik olarak karşımıza çıkmaktadır. Bu eksiklik doktora tezinin de temel motivasyonunu oluşturmuştur. Bu tezde, günümüz kablosuz algılayıcı ağlarında paketlerin güvenilir bir şekilde hedefe ulaştırılmasını sağlayacak, enerji tüketimini en aza indirerek düğümlerin yaşam süresini uzatacak bir algoritmanın geliştirilmesi hedeflenmiştir. Bir sonraki bölümde (Bölüm 4) geliştirilen taşıma katman algoritması ile ilgili detaylı bilgi verilecektir.

## **BÖLÜM 4. HİBRİT KAYIP TELAFİ MEKANİZMASINA DAYANAN YENİ BİR TAŞIMA PROTOKOLÜ**

### **4.1. Giriş**

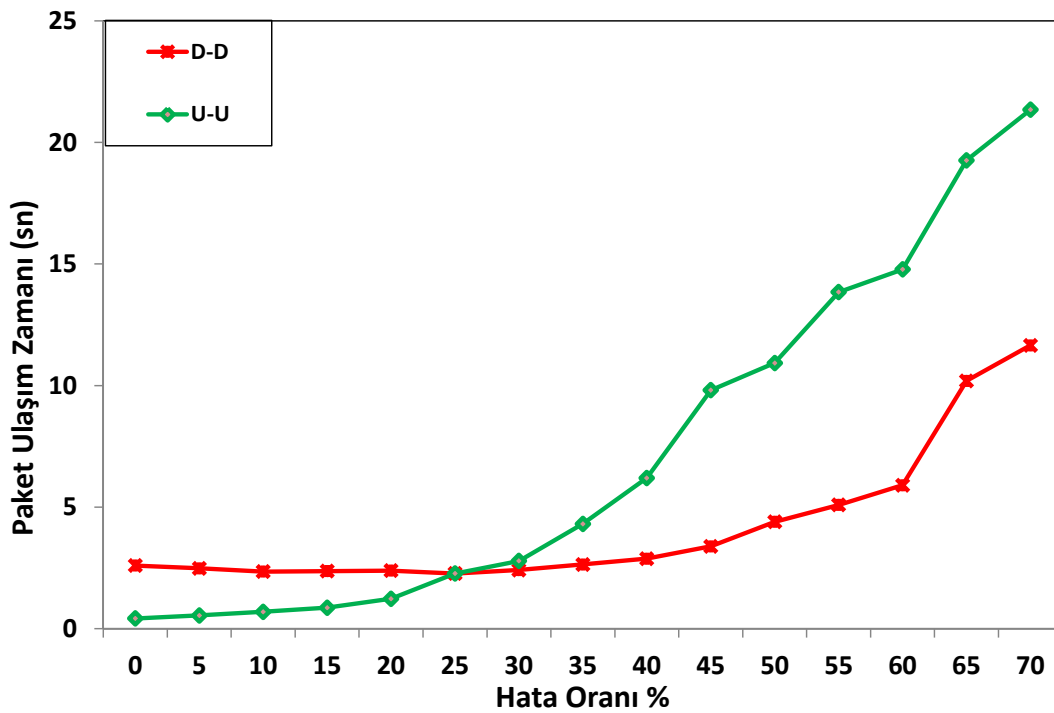
Geleneksel ağlarda bulunan taşıma katmanı, multiplexing (Çoğullaştırma işlemi) ve demultiplexing (Tekilleştirme işlemi) yaparak ağ katmanı ile uygulama katmanı arasında köprüleme yapılmasını sağlar. Buna ek olarak, veri trafiğini düzenleyerek, kaynaktan gönderilen paketlerin uçtan-uca güvenilirliğini garanti etmeye çalışırlar. Kablosuz algılayıcı ağlarda da geleneksel ağlara benzer olarak taşıma katman protokolleri, %100 paket ulaşımı ve güvenilir veri transferini sağlamak, sınırlı gecikme, minimum bant genişliği talebi, multimedya akışı için düzgün trafik yönetimi ve hata kontrol yapıları gibi gereksinimleri sağlaması gerekmektedir. Ancak, geleneksel ağlarda kullanılan TCP/UDP [70],[71] gibi taşıma protokollerinin bazı kısıtlamalar nedeniyle doğrudan KAA'larda kullanılamaması, uygulamaların gereksinimlerini karşılayacak düzeyde güvenilir bir taşıma protokolünün KAA'lara yönelik olarak geliştirilmesini zorunlu kılmıştır. Bu bölümde, KAA'lar önerilen **Hibrit Kayıp Telafi Mekanizmasına Dayanan Yeni Bir Taşıma (HİKMET) Protokolünün** tasarım basamakları, işleyişi ve mimarisi ele alınacaktır.

### **4.2. Motivasyon**

Güvenilir taşıma protokollerinde kayıp paketlerin telafi edilebilmesi için düğümden-düğüme ya da uçtan-uca hata tespiti ve kurtarma tekniği kullanılmaktadır. Diğer bir ifadeyle, eğer, güvenilir bir taşıma protokolü tasarlanmak isteniyorsa bu iki teknikten birisi kullanılarak, gönderim sırasında bozulan paketlerin telafisi gerçekleştirilmelidir. Geleneksel uçtan-uca hata tespiti ve kurtarma tekniğinde hedef düğüm eksik segmentleri kaynak düğümden isteyerek hatayı telafi etmeye çalışırken, düğümden-düğüme tekniğinde ise ara düğümler eksik segmentlerin kurtarılmasından sorumludur. İki hata tespiti ve kurtarma tekniği kıyaslandığında özellikle güvenilir



olmayan bağlantılarda çok fazla paket kayıplarının olduğu düşünülürse, düğümden-düğüme tekniği daha iyi sonuç verebilir. Ancak, bunun yanında eksik segmentlerin ara düğümlerde kurtarılması, beraberinde gecikme değerlerinin de önemli oranda artmasına yol açmaktadır. Uçtan-uca hata tespiti ve kurtarma tekniği ise hata oranların düşük olduğu ortamlarda paketlerin hızlı bir şekilde iletilmesine olanak sağladığı için daha düşük uçtan-uca gecikme değerlerini sağlayabilmektedir. Bu iki yöntemin birbirine göre avantaj/dezavantajlarını kıyaslayabilmek ve hangi yöntemin güvenilir bir taşıma protokolü için daha verimli olduğunu anlayabilmek için ayrıntıları EK.B’de verilen çeşitli benzetimler gerçekleştirilmiştir. Bu benzetimlerde kaynak düğümler farklı uzaklıktaki hedef düğümlere 50 segment içeren 10 adet paket göndermiş ve farklı hata oranlarındaki uçtan-uca gecikme değerlerinin ortalamaları kaydedilmiştir. Şekil 4.1’de 7-atlamalı bir ağda, her iki hata tespiti ve kayıp telafi yöntemi için kaynaktan hedefe gecikme değerlerinin ortalamaları görülmektedir.



Şekil 4.1. Uçtan-uca ve düğümden-düğüme geçiş parametrelerinin belirlenmesi

Şekil 4.1’de görüldüğü üzere iletişim hattındaki hata oranlarının düşük olduğu durumlarda uçtan-uca hata tespiti ve kurtarma yöntemi daha düşük gecikme ile segmentlerin iletilmesini sağlayabilmektedir. Ancak, hata oranları yükseldikçe bozulan segment sayıları artmakta ve bu segmentlerin uçtan-uca istenmesi daha da

zorlaşmaktadır. Sonuç olarak da uçtan-uca gecikme değerleri düğümden-düğüme tekniğine göre daha fazlaşmaktadır. Grafikten ortamdaki hata oranları %25'i aştıktan sonra diğer bir deęişle uçtan-uca gecikme değeri 3.3 sn değerinin üstüne çıktığında uçtan-uca teknięi yerine düğümden-düğüme hata tespiti teknięini kullanmak daha avantajlı olacaktır. Benzer durumlar, 3, 5, 9 ve 11 gibi diğer atlama değerleri için de geçerlidir. Dolayısıyla, bir taşıma protokolünün tasarımında bu iki hata tespiti ve kurtarma yönteminden birisini kullanmak yerine iki yöntemi beraber kullanmak gecikmeye duyarlı bir taşıma protokolünün tasarımını mümkün kılacaktır. Bu tezde, bahsedilen bu motivasyon ile ortam koşullarına göre hata tespiti ve kurtarma teknięini hibrit olarak kullanabilen güvenilir bir taşıma protokol tasarımı üzerine odaklanılmıştır.

### **4.3. Hibrit Kayıp Telafi Mekanizmasına Dayanan Taşıma (HİKMET) Protokolü**

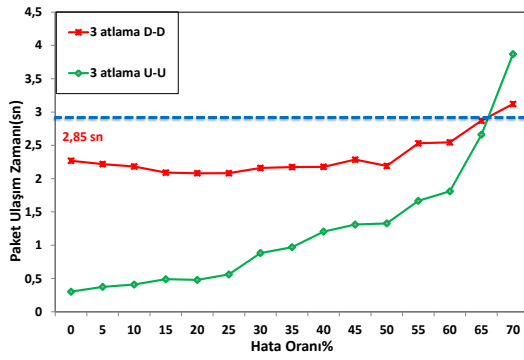
Önerilen HİKMET protokolü, hibrit kayıp telafi teknięini kullanarak kaynaktan-hedefe (upstream) paket güvenilirlięi saęlayan bir taşıma protokolüdür. HİKMET, ortam koşullarına göre kullandığı kayıp telafi teknięini deęiştirerek uçtan-uca gecikme değerleri ile paket teslim oranlarını optimize etmektedir. Bundan sonraki alt bölümlerde, HİKMET protokolünün kayıp telafi yöntemini nasıl deęiştirdięi, deęişimin nasıl gerçekleştięi, uçtan-uca ve düğümden-düğüme telafi yöntemlerinin nasıl çalıştığı gibi detaylar açıklanacaktır.

#### **4.3.1. Ortam koşullarının tespiti için eşik değerlerinin belirlenmesi**

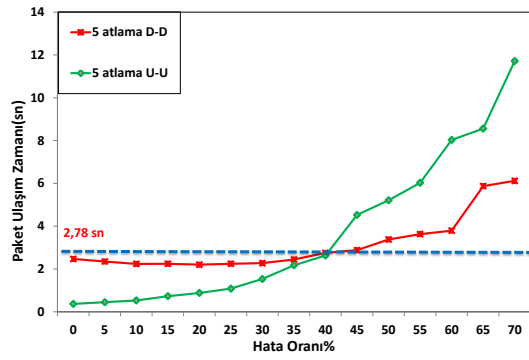
İletişim ortamdaki hata oranlarının tespiti gerçek hayatta oldukça güç bir işlemdir. Her ne kadar RSSI ve LQI gibi fiziksel katman parametreleri ile ortamdaki sinyalin gücü ve kalitesi ölçülebilse de bu parametreler ile Bit Hata Oranları arasında doğrudan bir ilişki her zaman bulunmayabilmektedir. Bu sebeple, HİKMET protokolünde, ortamdaki hata oranlarını ölçmek yerine onunla doğrudan ilişkili olan uçtan-uca gecikme parametresi değerlendirilerek iletim ortamı hakkında bilgi alınacağı varsayılmıştır. Uçtan-uca gecikmenin hangi değerlerinde kayıp telafi mekanizmasının deęiştirileceęine karar verilmesi için ise bir eşikleme mekanizmasından faydalanmaktadır.

Bu eşikleme işleminde, kaynak düğümler teslim edilen paketlerdeki gecikme değerlerinin istatistiğini tutarlar. Bu bilgiler ile önceden belirlenen eşik değerlerini belirli aralıklar ile kıyaslayarak bir sonraki gönderimin hangi kayıp telafi tekniği ile yapılacağına karar verirler. Ancak, tüm bunların yapılabilmesi için uçtan-uca gecikme parametresi için eşik değerlerinin hesaplanması gerekmektedir. Bu çalışmada, kaynak düğümlerin kullanacağı eşik değerlerini belirleyebilmek için farklı ortam koşullarında çeşitli deneyler gerçekleştirilmiştir. Bu deneylerde, 10x10 boyutlarında grid topolojisi kullanılmış ve her bir düğüm, farklı hata oranları için paket gönderimlerini, her iki hata kurtarma tekniğine göre gerçekleştirmiştir. Benzetimlerde kaynak düğümler, toplam 50 segmentten oluşan 10 farklı paketi hedef düğümlere iletmış ve uçtan-uca gecikme değerlerini her iki hata kurtarma tekniği için kaydetmiştir. Şekil 4.2’de 3-atlama, 5-atlama, 7-atlama ve 9-atlama için elde edilen gecikme değerleri görülmektedir. Grafiklerde görülen önemli bir husus farklı hata oranlarına ve atlama sayılarına rağmen her iki tekniğin bir birini kestiği noktalar oldukça yakındır. Diğer bir deyişle, farklı koşullara rağmen kaynak düğümlerin uçtan-uca tekniğinden düğümden-düğüme tekniğine geçmesi gerektiği gecikme değerleri birbirine benzerdir. Dolayısıyla, ağdaki tüm düğümler için global bir eşik değerinin belirlenmesi mümkün gözükmemektedir.

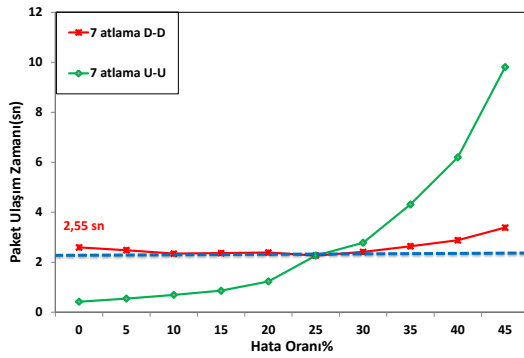
Bu çalışmada, elde edilen grafikler incelenerek geçişin yapılmasının gerekli olduğu gecikme değerleri belirlenmiş ve bu değerlerin normal dağıldığı varsayılmıştır. Normal dağılım, basit olmasına karşın doğadaki birçok problemin etkin bir şekilde modellenmesine olanak sağlamaktadır. Normal dağılıma göre değerlerin yaklaşık %68’i ortalamadan ( $\mu$ ) bir standart sapma ( $\sigma$ ) uzaklıkta, yaklaşık %95’i ise ortalamadan iki standart sapma uzaklıkta bulunmaktadır. Bu çalışmada, elde edilen gecikme değerlerinin ortalama ve standart sapma değerleri kullanılarak tüm düğümler için iki eşik değeri hesaplanmıştır. Şekil 4.3’te görüldüğü gibi uçtan-uca tekniğinden düğümden-düğüme tekniğine geçmek için  $T_A$  adı verilen alt eşik değeri kullanılmıştır. Bu alt eşik değeri, gecikme dağılımına göre  $\mu-2\sigma$  denklemiyle hesaplanmıştır. Benzer şekilde, düğümden-düğüme tekniğinden uçtan-uca tekniğine geçmek ise  $T_U$  adı verilen üst eşik değeri kullanılmıştır. Üst eşik değerini hesaplamak için ise  $\mu+2\sigma$  denkleminde faydalanılmıştır.



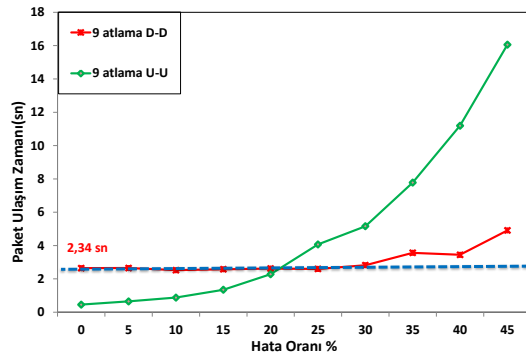
a-) 3-atlama için gecikme değerleri



b-) 5-atlama için gecikme değerleri

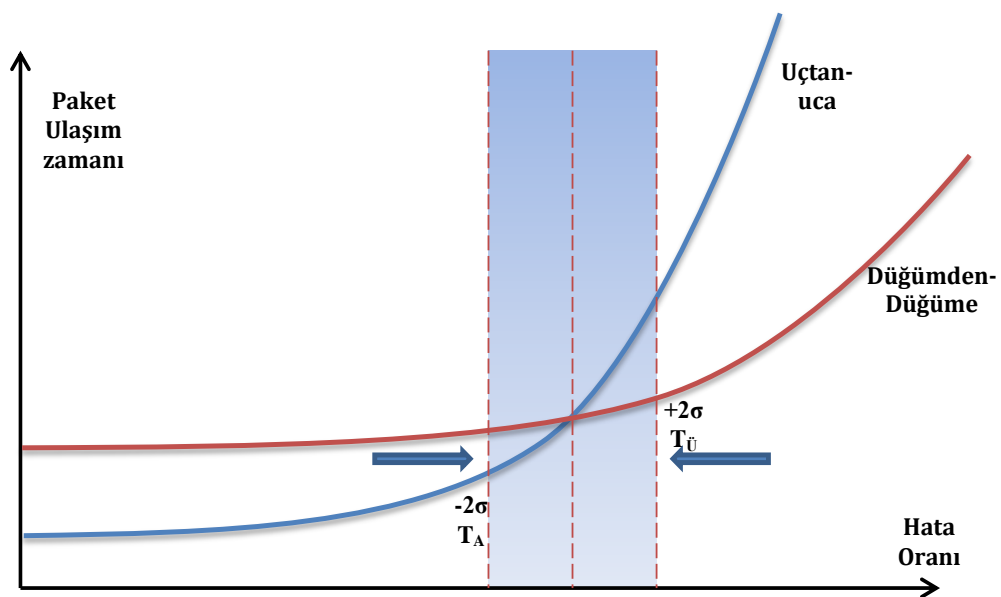


c-) 7-atlama için gecikme değerleri



d-) 9-atlama için gecikme değerleri

Şekil 4.2. 3, 5, 7, 9-atlama için düğümden-düğümüne ve uçtan-uca gecikme değerleri



Şekil 4.3. Uçtan-uca ve düğümden-düğümüne kayıp tespiti geçiş grafiği

### 4.3.2. Kayıp telafi mekanizmasının adaptif olarak değiştirilmesi

HİKMET protokolünde, kaynak düğümler gönderim yapmaya uçtan-uca yöntemi ile başlamakta ve gönderilen her paketin gecikme değerlerini kaydetmektedirler. Daha sonra, kaynak düğümler, Tur (round) adı verilen belirli sayıda gönderim gerçekleştirdikten sonra ilgili turun gecikme ortalaması ile önceden hesaplanmış olan alt eşik değerini kıyaslayarak geçişin gerekli olup olmadığına karar verirler. Eğer, bir turdaki gecikme ortalaması,  $T_A$  değerinden daha fazla ise, bir sonraki turda paket gönderimi düğümden-düğüme olarak yapılacaktır. Kaynak düğümler, düğümden-düğüme tekniğine geçtikten sonra ise o turdaki gecikme ortalaması ile bu sefer  $T_U$  eşik değerini kıyaslanarak geçiş işlemi kontrol edilmektedir. Bu durumda ise o turdaki gecikme ortalaması,  $T_U$  değerinden daha az ise bir sonraki turda paket gönderimi uçtan-uca olarak yapılacaktır. Tablo 4.1’de geçiş işleminin sahte kod olarak gösterimi bulunmaktadır.

Tablo 4.1. Kayıp Telafi Mekanizmasının değişimini gösteren sahte kodu

---

**Algoritma 1: Kayıp Telafi Mekanizmasının değişimini gösteren sahte kodu**

---

```

/*Senkronizasyon*/
KayıpTelafi= Uçtan-Uca;
SenkronizeEt();

/*Segmentlerin ağa enjekte edilmesi*/
SegmentGonder();
Bekle_ACK();

/*ACK alındığında gönderim değerlerinin kaydedilmesi*/
if (ACKAlındı==TRUE)
    teslimEdilenPaketSayısı++;
    gecikmeDegerKaydet();
end if

/*Eşik değerine ulaşıldığında bir sonraki gönderim Türünün belirlenmesi*/
if( teslimEdilenPaketSayısı >=TTUR)
    if( KayıpTelafi == Uçtan-Uca )
        if (gecikmeDeğerleri>TA)
            KayıpTelafi = Düğümden-Düğüme
        end if
    end if ( KayıpTelafi == Düğümden-Düğüme )
        if (gecikmeDeğerleri<TU)
            KayıpTelafi = Uçtan-Uca;
        end if
    end if
end if

```

---

Tablo 4.1’de görüldüğü gibi düğümler, ilk paket gönderimine uçtan-uca tekniği ile başladığı için gönderici ile alıcı arasında bir senkronizasyon yapılması gerekmektedir. Gönderilen bu senkronizasyon mesajı içerisinde bulunan başlıklar Şekil 4.4’te gösterilmektedir. Daha sonrasında ise gönderici segmentleri göndermekte ve paketin başarılı bir şekilde ulaştığından emin olmak için ACK beklemektedir. ACK mesajı kaynak düğüm tarafından başarılı bir şekilde alındığında, alınan paket değeri otomatik olarak bir artırılmakta ve gecikme değerleri düğüm içerisine kaydedilmektedir. Eğer alınan paket değeri, tur sayısına eşit ise, bu turdaki gecikme değerlerinin ortalaması hesaplanmakta ve gerekli kıyaslamalar gerçekleştirilmektedir. Elde edilen değerlere göre bir sonraki gönderim tekniği belirlenmiş olur.

Kaynak Düğüm Kimliği
Hedef Düğüm Kimliği
Toplam Segment Sayısı
ACK Numarası
Atlama Sayısı

Şekil 4.4. Senkronizasyon mesajı başlıkları

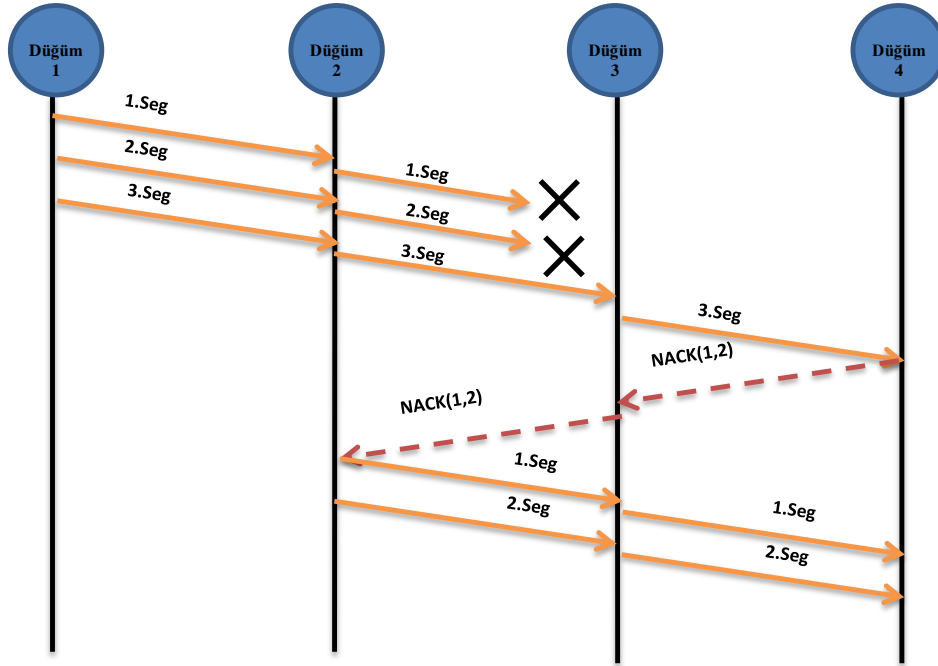
### 4.3.3. Uçtan-uca paket gönderimi

HİKMET protokolünde düğümler, ilk olarak uçtan-uca hata kurtarma tekniğini kullanarak çalışmaya başladığı için bu alt bölümde, bu kayıp telafi yönteminin ayrıntıları açıklanacaktır.

Uçtan-uca paket gönderim yönteminde kaynak düğümler paket gönderimini başlatabilmek için TCP’de olduğu gibi öncelikle hedef düğüme bir senkronizasyon (SYNC) mesajı göndermelidir. Bu mesaj, kaynak düğümün kimliği, hedef düğümün kimliği, toplam segment sayısı, ACK numarası ve atlama sayısı gibi bilgiler içermektedir. Aynı zamanda gönderilen senkronizasyon mesajının bozulma ihtimaline karşın kaynak düğümde  $T_{senk}$  zamanlayıcısı kurulmaktadır. SYNC mesajını alan hedef düğüm, eğer paketi almayı kabul ediyorsa bir SYN\_ACK mesajını kaynak düğüme gönderir.  $T_{senk}$  zamanlayıcısı dolana kadar SYN\_ACK

mesajı kaynak düğüme ulaşmaz ise, kaynak düğüme mesajın bozulduğuna karar verir ve SYNC'yi hedef düğüme tekrar gönderir. Eğer, kaynak düğüme SYN\_ACK mesajını başarılı bir şekilde aldıysa segment gönderimine başlayacağını gösteren bir ACK mesajını hedef düğüme göndermektedir. Bundan sonra, kaynak düğüme ACK beklemeden segmentleri sırayla göndermeye başlar. Komşu düğümler de aldıkları bu segmentleri hiç bekletmeden bir sonraki düğüme iletir. Klasik uçtan-uca sistemlerin yüksek hata oranlarında yeterli performansı verememesinin en önemli sebebi segmentler bozulduğunda eksiklerin kaynak düğümden istenmesidir. Çok atlamalı ağlarda bu işlem çok uzun süreceği için klasik uçtan-uca sistemler KAA'larda pek tercih edilmemektedir. Bu sebeple, HİKMET protokolünün uçtan-uca kayıp telafi yönteminde Tiny TCP [83], [84] protokolünden esinlenilmiştir.

Önerilen uçtan-uca yönteminde, ara düğümlere ulaşan her yeni segment düğümlerin belleklerinde saklanır ve kopyaları hiç bekletilmeden bir sonraki düğüme iletilir. Tamponlama sayesinde segmentler bozulduğunda, ara düğümlerden daha hızlı bir şekilde telafi edilebilecektir. Şekil 4.5'te bu yapı grafiksel olarak gösterilmiştir. 3 segmentli bir paketin gönderildiği düşünülürken, 1. ve 2. Segmentler, 2 numaralı düğüme kadar gelip bundan sonra bozulmuştur. 3. segment ise başarılı bir şekilde hedefe ulaşmıştır. Hedef düğüme aldığı 3. segment ile ilk 2 segmentin kayıp olduğu anlamakta ve bir NACK paketi ile geri yönde eksikleri istemektedir. Hedef düğümden NACK paketi alan 3. düğüme ise eksikleri kendi tamponunda aramakta ve bulunduğu için hedefe göndermektedir. Bu sayede, kaynağa kadar gitmeden eksik segmentler çok daha kısa sürede ve daha az enerji harcanarak telafi edilebilmektedir. Düğüme belleklerinin aşırı yüklenmesini engellemek için ise hedef düğümden gönderilen NACK paketlerinden faydalanılır. Bir ara düğüme gelen NACK paketi içerisinde eksik segmentlerin numarası belli olduğu için ara düğümler, hedefe ulaşan sağlam segmentleri anlayabilir. Örneğin, Şekil 4.5'teki senaryoda 3. düğüme 4. düğümden gönderdiği NACK paketindeki 1 ve 2. segmentlerin eksik olduğunu görerek 3. segmentin 4. düğüme ulaştığını anlar ve kendi tamponunda bulunan bu segmenti siler. Bu şekilde, NACK paketleri yardımıyla başarıyla ulaştığı anlaşılan tüm segmentler ara düğümlerin tamponundan silinmiş olur.



Şekil 4.5. HİKMET protokolünde uçtan-uca kayıp segment telafisi

#### 4.3.3.1. Ara NACK gönderimi

Yüksek hata ortamlarında, segmentlerin çoğu kaynaktan hedefe belirlenen zaman içerisinde ulaşamamaktadır. Bundan dolayı eksik segmentlerin telafi edilmesi çok uzun sürmektedir. Bunu engellemek için HİKMET protokolü ara NACK yapısını kullanmaktadır. Ara NACK yapısına göre, SYN\_ACK paketi hedef düğüme ulaştığında  $T_{NACK}$  adında bir zamanlayıcı kurulur. Denklem 4.1'de  $T_{NACK}$  zamanlayıcısının kurulması için gerekli olan hesaplamalar görülmektedir.

$$T_{NACK} = (\text{Segment Oranı} * T_{\text{Varış}}) * \beta \quad (4.1)$$

Denklemdaki segment oranı, zamanlayıcının kaç adet segment süresi sonrasında taşıacağını gösteren parametredir ve bu parametre bir paketteki toplam segment sayısına göre belirlenmektedir. Yaptığımız benzetimlere göre segment oranı, toplam segment sayısının %10 oranında olduğunda en uygun paket maliyeti ve paket gecikme değerleri elde edildiği görülmüştür. Örneğin, 50 segmentlik bir pakette, her 5 segmentlik zamandan sonra zamanlayıcı devreye girerek o zamana kadar hedefe ulaşması gereken segmentlerin belirlenmesini sağlamaktadır.  $T_{\text{varış}}$  parametresi bir

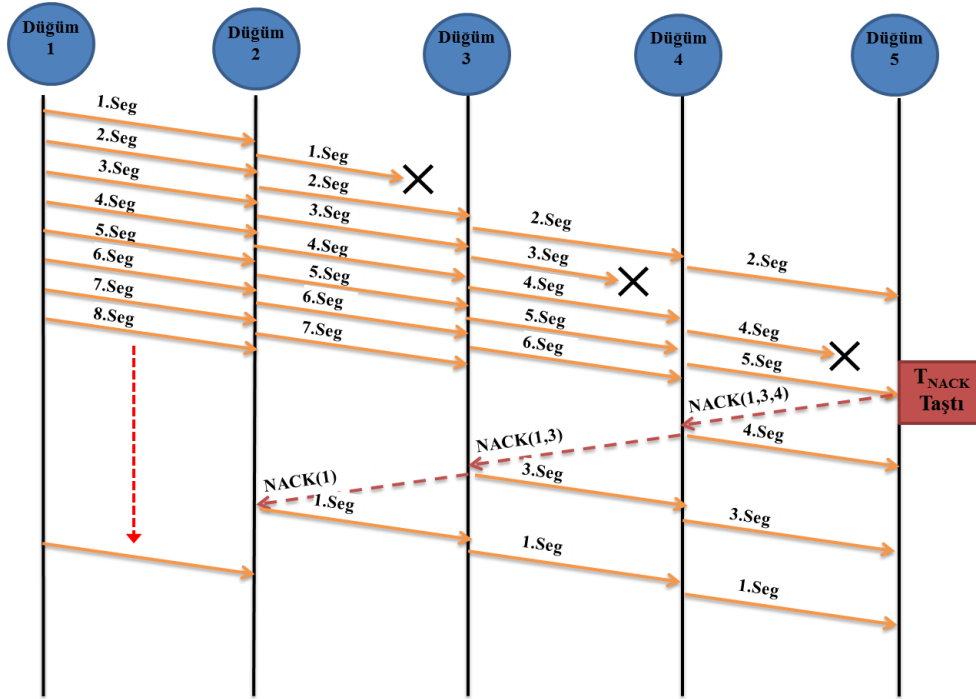


segmentin kaynaktan hedefe ortalama ulaşım süresini gösterirken  $\beta$  parametresi ise ortam koşullarını modellemek için kullandığımız güvenlik faktörünü temsil etmektedir.

Bir hedef düğümde  $T_{NACK}$  zamanlayıcısı taşıdığı düğüm, tamponundaki segmentleri kontrol eder ve ilk segmentten ulaşması gereken segment numarasına kadar olan tüm eksik segmentleri tespit eder. Eğer, eksik segment varsa, düğüm eksik olan segmentlerin numaraları içeren bir NACK mesajı üretir ve bu paketi geri yönde gönderir. NACK mesajını alan ara düğümler kendi tamponlarına bakarak eksik segmentlerin tamponlarında olup olmadığını kontrol ederler. Kendisinde bulunan segmentleri hedef düğüme gönderirken, olmayan segmentleri ise bir önceki düğümden NACK mesajı ile isterler. Eğer, ara düğümlerin gönderdiği NACK mesajı herhangi bir sebeple bozulursa, segment kurtarma işlemi bir sonraki  $T_{NACK}$  zamanlayıcısının taşmasına kadar sona ermiş olur. Bu durumda, bir önceki saykılta kurtarılamayan segmentler için yeniden kurtarma işlemi başlatılır. Kurtarma işlemi son segment aralığına kadar böylece devam eder. Son segment aralığında NACK zamanlayıcısının taşması ile hedef düğümün belleği baştan sona kontrol edilerek tüm eksik segmentler belirlenir. Belirlenen eksik segmentlerin sayısına göre NACK zamanlayıcısı Denklem 4.2'deki formüle göre yeniden yüklenir. Denklemdeki, kayıp segment sayısı, başarılı şekilde alınamamış segmentlerin sayısını gösterirken,  $T_{Varış}$  bir segmentin kaynaktan hedefe ortalama ulaşım süresini,  $\beta$  ise ortam hata koşullarına göre belirlenen güvenlik katsayısını ifade etmektedir. NACK zamanlayıcısının bundan sonraki yükleme değerleri kalan eksik sayısına göre güncellenmektedir.

$$T_{NACK} = (\text{kayıp segment sayısı} * T_{Varış}) * \gamma \quad (4.2)$$

Şekil 4.6'da ara NACK yapısının çalışmasını gösteren örnek bir senaryo görülmektedir.



Şekil 4.6. Ara NACK gönderim şeması

Şekilde, 1 nolu segment, 3 numaralı düğüme ulaşmadan bozulmuştur. 2 nolu segment, hedef düğüme başarılı bir şekilde ulaşmıştır. 3 nolu segment, 4 numaralı düğüme, 4 nolu segment ise 5 numaralı düğüme ulaşmadan bozulmuştur. 5 nolu segment hedefe başarılı bir şekilde ulaşmıştır. Bu anda, hedef düğüme  $T_{NACK}$  zamanlayıcısı devreye girerek, tüm tampon belleğin kontrol edilmesini ve 1 nolu segmentten 5 nolu segmente kadar olan eksik segmentlerin belirlenmesini sağlamıştır. Eksikler belirlendikten sonra eksiklerin Sıra numaraları bir NACK içerisine yerleştirilerek geri yönde diğer düğümlere gönderilmiştir. NACK mesajını alan 4 nolu ara düğüm, tamponunu kontrol ederek, belleğinde bulunan eksik segmentleri hedefe gönderir. Kendisinde olmayanları ise tekrar bir NACK mesajı ile kendisinden önceki düğümlere iletir. Hedef düğüm, her  $T_{NACK}$  taşmasında tamponunu baştan sonra tarayarak eksikleri belirler ve bunları yeni NACK paketleri ile ister. Son segment aralığına kadar bu şekilde kurtarma işlemi gerçekleştirilir. Tüm segmentler başarılı bir şekilde alındıktan sonra hedef düğümler kaynak düğüme paket gönderiminin başarılı bir şekilde bittiğini belirten bir “Başarı ACK (Success ACK)”sı gönderir.

Tablo 4.2’de HİKMET protokolünde kullanılan uçtan-uca hata kurtarma tekniğinin sahte kodları görülmektedir. Hata kurtarma işleminde 3 farklı görev söz konusu olduğu için, gönderici, ara ve alıcı düğümlerin sahte kodları ayrı ayrı verilmiştir.

Tablo 4.2. HİKMET protokolünün uçtan-uca çalışma yöntemi

<b>Algoritma 2: HİKMET Protokolünün uçtan-uca çalışma yöntemi</b>	
<p><b><u>Kaynak Düzüm</u></b>  <i>/*Senkronizasyon*/</i>  Gönder_SYNC();  Ayarla_TSENK();</p> <p><b>if</b>(TSENK taşı ise)      TekrarGonder_SYNC();  <b>end if</b></p> <p><i>/*Segmentlerin ağa enjekte edilmesi*/</i>  <b>if</b> (SYNC_ACK Alındı ise)      iptalEt_TSENK();      Gönder_ACK();      Segment_Gonder();  <b>end if</b></p> <p><i>/*Kayıp segment gönderimi*/</i>  <b>if</b>(NACK Alındı ise)      Gönder_Eksik_Segment();  <b>end if</b></p> <p>Bekle_BaşarıACK();</p> <p><i>/*Başarılı gönderimden sonra tamponun temizlenmesi*/</i>  <b>if</b> (Başarı_ACK Alındı ise)      iletim=Başarılı;  <b>end if</b></p> <p><b><u>Ara Düzüm</u></b></p> <p><i>/*Gelen segmentlerin iletilmesi*/</i>  <b>if</b>(Segment Alındı ise)      <b>if</b>(Segment Bellekte yoksa)          belleğeKaydet();          segmentGonder();      <b>end if</b>  <b>end if</b></p> <p><i>/*Kayıp segmentlerin gönderimi*/</i>  <b>if</b>(NACK Alındı ise)      belleğiKontrolEt();      KayıpSegmentGonder();      başarılıSegmentKontrolEt();      başarılıSegmentSil();  <b>end if</b></p> <p><i>/*Başarı ACK’sı alındığında tamponun temizlenmesi*/</i>  <b>if</b>(Başarı_ACK Alındı ise)      belleğiSil();  <b>end if</b></p>	<p><b><u>Hedef Düzüm</u></b></p> <p><i>/*Senkronizasyon*/</i>  <b>if</b>(SYNC Alındı ise)      Gönder_SYN_ACK();  <b>end if</b></p> <p>Bekle_ACK();</p> <p><i>/*Segmentlerin alınmaya başlaması ve Ara NACK zamanlayıcısının ilk kurlumu*/</i>  <b>if</b>(ACK Alındı ise){      Segmentleri_AI();      Ayarla_TNACK();  <b>end if</b></p> <p><i>/*Eksik Seg.’lerin Ara NACK ile kurtarılması*/</i>  <b>for</b>(SegInt=1;SegInt&lt;LastSegInt;SegInt++)      TNACK_Bekle();      belleğiKontrolEt();      <b>if</b>(eksikSegmentSayisi&gt;0)          Gönder_NACK();      <b>end if</b>      Ayarla_TNACK();  <b>end for</b></p> <p><i>/*Son seg. aralığından sonra kalan eksik Segmentlerin kurtarılması*/</i>  TNACK_Bekle();  belleğiKontrolEt();</p> <p><b>while</b>(eksikSegmentSayisi &gt;0)      Gönder_NACK();      Ayarla_TNACK();  <b>end while</b></p> <p><i>/*Tüm seg. alındıktan sonra kaynak düğüme Başarı ACK gönderilmesi*/</i>  <b>if</b>(tümSegmentler Alındı ise){      Gönder_Başarı_ACK ();      iptalEt_TNACK();  <b>end if</b></p>

#### 4.3.4. Dügümden-dügüme kayıp telafi mekanizması

HİKMET protokolü bir önceki bölümde ayrıntıları verilen uçtan-uca tekniğine ek olarak düğümden-dügüme hata kurtarma tekniğini de kullanmaktadır. HİKMET protokolünde kullanılan düğümden-dügüme tekniği PSFQ protokolünden esinlenerek tasarlanmıştır. Bu teknikte paketi oluşturan segmentler kaynak düğüm tarafından periyodik aralıklarla ağa dâhil edilir. Ara düğümler alınan segmentleri belirli bir zaman aralığında diğer düğümlere iletirler. Hata tespiti, segment sıra numaralarına göre belirlenir. Bu yöntemde, bir düğüme ulaşmasını beklediği segment numarası yerine daha büyük segment numarası gelmişse, düğüm aradaki segmentlerin bozulduğunu varsayar. Eksik segment numaralarını içeren bir NACK paketini geri yöndeki düğüme gönderir ve eksik segmentlerin ulaşmasını bekler. Segmentler kurtarıldıktan sonra ileri yöndeki düğümlere sıra ile iletilir. Dügümden-dügüme tekniğinde eksik segmentleri kurtarıırken ara düğümlerin ısrarcı olması zor ortam koşullarında bile yüksek paket başarımının elde edilmesine imkân tanımaktadır.

HİKMET protokolünün düğümden-dügüme hata tespiti ve kayıp kurtarma kısmı üç fonksiyon içerir. Bunlar paket gönderme, hata tespiti, kayıp geri alımı ve bilgilendirmedir. Kaynak düğüm mesajları ağa dâhil eder. Ara düğümler paketleri belleğine kaydeder ve diğer düğümlere gönderir ve gerekliyse kayıp tespiti ve kayıp kurtarma işlemlerini başlatır. Alt bölümlerde HİKMET protokolü için geliştirilmiş olan düğümden-dügüme hata kurtarma tekniğinin ayrıntıları ele alınacaktır.

##### 4.3.4.1. Segment gönderme işlemi

Segment gönderme işleminde kaynak düğüm, segmentleri periyodik aralıklarla ağa yaymaktadır. Segmentler arasındaki bekleme süresi, bozulan segmentlerin ara düğümlerde kurtarılabilmesine imkân sağlamaktadır. Önerilen düğümden-dügüme tekniğinde segmentler arasındaki bekleme süresi  $T_{Min}$  ve  $T_{Max}$  adındaki iki zamanlayıcı yardımıyla hesaplanmaktadır.

Kaynak düğüm her  $T_{Min}$  aralığında ileri yöndeki komşu düğüme segmentleri sıra ile gönderir. Segmentleri alan ara düğümler ise, öncelikle gelen segmentlerin kendi

tamponlarında bulunup bulunmadığını kontrol ederler. Eğer alınan segment tamponda bulunmuyorsa, diğer bir ifadeyle yeni segment alınmışsa, segmenti tampona kaydeder ve segment numarasını kontrol eder. Eğer sıra numarasında herhangi bir boşluk yoksa ara düğüm segmenti  $T_{Min}$  ve  $T_{Max}$  arasındaki rasgele belirlenen bir gecikme ile gönderir. Segmentler bu şekilde kaynak düğümünden hedef düğüme doğru yayılır. Eğer gelen segment numaralarında bir boşluk varsa, düğüm hata kurtarma moduna geçiş yapar. Eksik segmentleri belirleyerek, bunları geri yöndeki düğümlerden telafi etmeye çalışır.

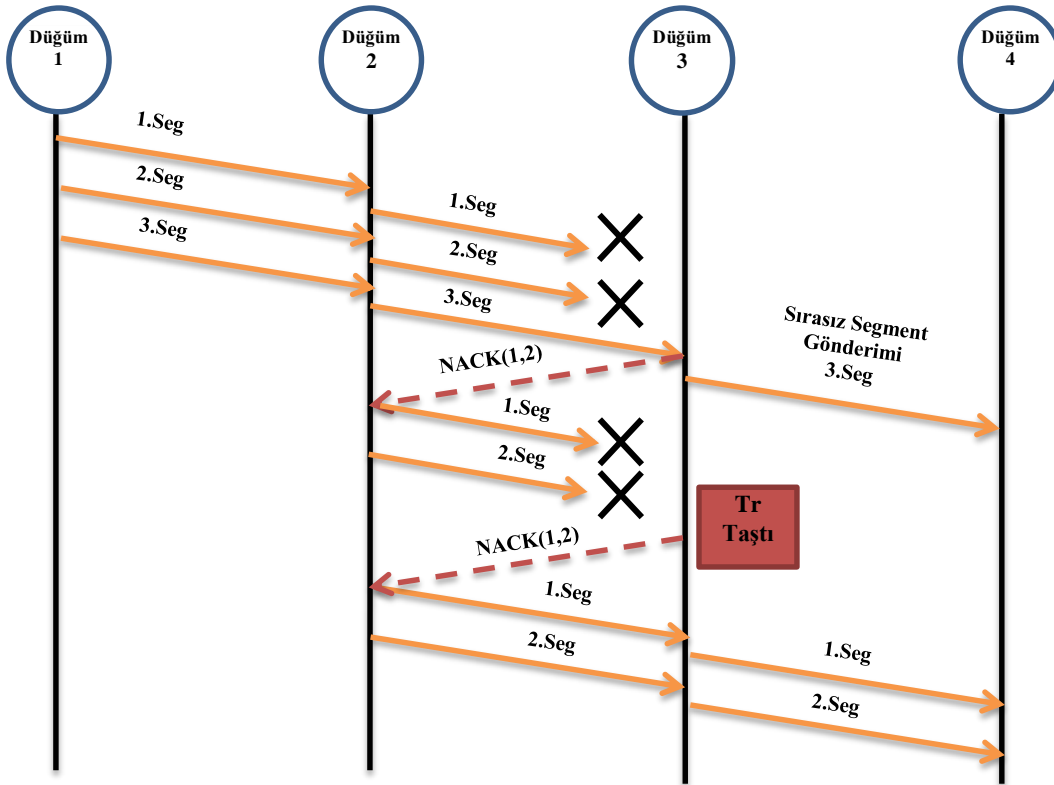
#### 4.3.4.2. Sırasız segment gönderme işlemi

HİKMET protokolü, düğümünden-düğüme tekniğine göre çalıştığı zaman uçtan-uca gecikmeyi iyileştirebilmek için sırasız segment gönderimi yapısını kullanır. PSFQ protokolünde segment düğüme ulaştığı zaman, gelen segment hatalı ise, düğüm hata kurtarma moduna geçiş yapar. Gelen segmenti, aradaki eksik segmentler kurtarılan kadar kendi tampon belleğinde bekletir. Bu da gereksiz gecikmelere sebep olur. HİKMET, bu duruma çözüm üretebilmek için, gelen segmenti bekletmeyerek diğer düğümlere de iletir. Bu durumda, diğer düğümlerin hata kurtarma moduna geçiş yapmaması için segmentin içerisinde bulunan “iletme” adlı biti set eder. Bu sayede düğümler bunun sırasız gönderim ile geldiğini anlar ve hata kurtarma moduna geçiş yapmazlar.

#### 4.3.4.3. Adaptif hata kurtarma yapısı

Gelen segment numarası ile beklenen segment numarası farklı olduğunda, ara düğümler hata kurtarma moduna girer. Bu durumda ilgili ara düğüm, eksik segmentleri belirleyerek geri yöndeki komşu düğümünden eksikleri bir NACK paketi ile ister. Her eksik segment için ayrı bir NACK paketi yollamak yerine tek bir NACK paketi ile bu işlem yapılabilir. NACK gönderen düğüm istemiş olduğu eksik segmenti beklemek için, eksik segmentlerin ortalama ulaşım zamanı kadar olan  $T_R$  adında ek bir zamanlayıcı kurar. Eğer, eksik tüm segmentler  $T_R$  zamanlayıcısı taşmadan kurtarılsa hata kurtarma işlemi sonlanmış olur. Ancak,  $T_R$  süresi içerisinde eksik segmentlerin bir kısmı kurtarılsa kalan eksik segmentler için

yeniden NACK paketi oluşturulur ve  $T_R$  kurur. Bu işlem, HİKMET protokolünde eksik segment sayısına göre maksimum 3 kere tekrar edilebilir. Normal şartlarda bir kurtarma evresinde en fazla 12 tane segment başarıyla kurtarılabilir. Adaptif hata kurtarma yapısında, eğer eksik segment sayısı 0-4 arasında ise 3, 5-6 arasında ise 2, 6'dan fazla ise 1 kere kurtarma işlemi tekrar eder. Tüm denemelere rağmen ilgili segmentler kurtarılamazsa, kurtarma işlemi iptal edilir. Bunun amacı, gereksiz segment gönderimini engellemek ve bu sayede enerji korunumu sağlamaktır. Şekil 4.7'de düğümden-düğüme hata kurtarma tekniğinin çalışmasını gösteren örnek bir senaryo görülmektedir.



Şekil 4.7. Düğümden-düğüme tekniğinin çalışmasını gösteren örnek bir senaryo

#### 4.3.4.4. Adaptif proaktif kayıp tespiti ve telafisi

Düğümden-düğüme hata kurtarma tekniklerinde eksik segmentlerin tespiti, başarılı alınan segmentlerin sıra numarasına göre belirlenmektedir. Örneğin, 6 segmentli bir paket için hedef düğüm 1. ve 2. Segmentleri başarıyla alıp 3.,4. ve 5. Segmentleri alamadıysa, bu düğümün eksik olan 3 segmenti tespit edebilmesi için 6. Segmenti

başarıyla alması gerekmektedir. Ancak, 6. Segmentin de kaybolduğu ve hedef düğüme hiç ulaşamadığı düşünülürse, hedef düğüm eksikleri tespit edemez ve bir kısır döngüye (deadlock) girer. Örnekte bahsedilen bu sıkıntıyı aşmak için düğümden-düğüme hata kurtarma yöntemlerinde genellikle bir periyodik zamanlayıcı kullanılmaktadır. HİKMET protokolünde de, PSFQ' dakine benzer olarak  $T_{PRO}$  adında bir proaktif hata kurtarma mekanizması kullanılmaktadır.  $T_{PRO}$  sayesinde kısır döngüye giren ve eksik kalan segmentleri bekleyen bir hedef düğüm, zamanlayıcının taşmasıyla bekleme modundan çıkıp proaktif hata kurtarma işlemini başlatır. Proaktif modda hedef düğüm, paket içerisindeki tüm eksik segmentleri belirleyerek, bu segmentleri geri yöndeki düğümlerden NACK paketi ile ister. NACK paketini alan ara düğümler hedef düğümün istediği eksik segmentlerin kendi tampon belleğinde olup olmadığını kontrol ederler. Kendi tamponunda bulunan segmentleri hedefe doğru gönderip, eksik olan segmentleri geri yöndeki düğümlerden yeni bir NACK paketi ile isterler.

$T_{PRO}$  zamanlayıcısının ilk kez yüklemesinde kullanılan formül Denklem 4.3'de görülmektedir.

$$T_{PRO} = (S_{MAX} - S_{SON}) \times T_{MAX} * \alpha (\alpha \Rightarrow 1) \quad (4.3)$$

Denklem incelendiğinde,  $\alpha$  gecikmeyi düzenleyen ve ortam koşullarına göre belirlenen bir parametreyi,  $S_{MAX}$  toplam segment sayısını,  $S_{SON}$  en son alınan segment numarasını,  $T_{MAX}$  ise bir segmentin bir düğümden diğer düğüme iletilmesinde geçen maksimum süreyi ifade etmektedir. Denklem 4.3'e göre ilk kez kurulan  $T_{PRO}$  zamanlayıcısının taşması durumunda hedef düğüm eksik segmentleri kontrol edip bu segmentlere göre yeniden bir yükleme değeri belirlemelidir. Bu şekilde tüm eksik segmentler kurtarılan ya da belirli bir deneme sayısına ulaşılan kadar  $T_{PRO}$  zamanlayıcısı tekrar güncellenmelidir.  $T_{PRO}$  zamanlayıcısının güncellenmesinde kullanılan formül ise Denklem 4.4'te görülmektedir.

$$T_{PRO} = (\text{Eksik Segment Sayısı}) \times T_{MAX} * \alpha (\alpha \Rightarrow 1) \quad (4.4)$$

HİKMET protokolünde düğümden-düğüme kurtarma tekniği genelde yüksek hata ortamlarında çalışacağı için proaktif kurtarma modu son derece önemlidir. Ancak, proaktif modda hedef düğümün göndereceği NACK paketlerinin de bozulma ihtimali oldukça yüksektir. NACK paketlerinin bozulması ise eksik segmentlerin kurtarılabilmesi için sonraki  $T_{PRO}$  zamanlayıcısının taşmasının beklenmesini gerektirmektedir. Bu da gecikmelerin ciddi manada artmasına yol açmaktadır. Bu sebeple, önerilen proaktif modunda her  $T_R$  zamanında yeni bir NACK paketi gönderilerek hata kurtarma işleminin daha performanslı olması sağlanır. Bu yeniden gönderme sayısı da adaptif hata kurtarma yapısında olduğu gibi en fazla 3 olmaktadır. Adaptif proaktif hata kurtarma yapısında, eksik segment sayısı 0-4 arasında ise 3, 5-6 arasında ise 2, 6'dan fazla ise 1 kere NACK gönderme işlemini tekrar eder. Buna rağmen hala eksik segmentler kurtarılamamış ise,  $T_{PRO}$  zamanı kadar sonar tekrar eksik segmentler belirlenerek işlemler aynı şekilde tekrar edilir.

#### 4.3.4.5. Kaynak düğüme bilgi mesajı gönderilmesi

HİKMET protokolü düğümden-düğüme tekniği ile çalışırken, paket gönderimi başarılı bir şekilde bittikten sonra kaynak düğüme bir ACK paketi gönderilir. Bu sayede kaynak düğüm paket gönderimin başarılı bir şekilde bittiğini anlar. Bu paket içerisinde aynı zamanda paket gönderimi sırasında elde edilen paketin ulaşım zamanı, toplam NACK sayısı gibi bazı değerler de ulaştırılır. Kaynak düğüm, bu paketi aldığı anda içerisindeki bilgileri kendi tampon belleğine kaydeder. Eğer, hata kurtarma tekniği eşik değerine ulaşılmışsa, paketlerin ortalama ulaşım süresi hesaplanır. Eğer, paket ulaşım zamanı  $T_{\bar{U}}$  eşik değerinden az ise, bir sonraki paket gönderiminin uçtan-uca tekniği ile yapılması sağlanır. Çünkü paketin ulaşım zamanı  $T_{\bar{U}}$  eşik değerinin altında ise, bu durumda bir sonraki paket gönderimi için uçtan-uca tekniği daha performanslı olacaktır.

Tablo 4.3'de HİKMET protokolünde kullanılan düğümden-düğüme kayıp telafi mekanizmasının sahte kodları görülmektedir. Hata kurtarma işleminde 3 farklı görev söz konusu oluşu için, kaynak, ara ve hedef düğümlerin sahte kodları ayrı ayrı verilmiştir.



Tablo 4.3. HİKMET protokolünün düğümden-düğüme çalışma yöntemi

<b>Algoritma 3: HİKMET Protokolünün düğümden-düğüme çalışma yöntemi</b>	
<pre> <b><u>Kaynak Düğüm</u></b> /*Segment gönderimi*/ For(Seg_No=1;Seg_No&lt;=Tot_Seg;Seg_No++)     Segment_Gonder();     Bekle_T<sub>min</sub>(); <b>endfor</b> Bekle Başarı_ACK();  /*NACK alındığında eksikleri gönder*/ <b>if</b> (NACK Alındı ise)     eksikSegmentGonder(); <b>endif</b> /*iletim bittiyse belleği temizle*/ <b>if</b> (Başarı_ACK Alındı ise)     iletim=Başarılı;     bellekTemizle(); <b>endif</b>  <b><u>Hedef Düğüm</u></b> /*Segment alınınca T<sub>pro</sub> avarla*/ <b>if</b> (ilk_segment Alındı ise)     Ayarla_T<sub>Pro</sub>(); <b>if</b> (T<sub>Pro</sub> taşı ise)     BellekKontrolEt();     <b>if</b> (kayıpSegmentSayısı&gt;0)         Gönder_NACK();         Ayarla_T<sub>Pro</sub>();     <b>endif</b> <b>endif</b> /*Gönderim bitince Başarı_ACK gönder*/ <b>if</b> (tümSegmentler Alındı ise)     Gönder_Başarı_ACK();     İptalEt_T<sub>Pro</sub>(); <b>endif</b>  <b>if</b> (T<sub>Pro</sub> taşı ise)     Gönder_Başarısız(); /*Segment alm işlemleri*/ <b>if</b> (Segment Alındı ise)     <b>if</b> (Segment Bellekte Yoksa)         BelleğeEkle();         <b>if</b> (SeqNo Hatalıysa)             <b>if</b> (Seg_Sirasız==0)                 BellekKontrolEt();                 Gonder_NACK();                 Ayarla_TR();                 TR_Sayıcı ++;             <b>endif</b>         <b>endif</b>     <b>endif</b> <b>endif</b> <b>if</b> (T<sub>R</sub> Taştı ise)     <b>if</b> (T<sub>R</sub>_Eşik&lt;= TR_Sayıcı)         BellekKontrolEt();         <b>if</b> (kayıpSegmentSayısı &gt;0){             Gonder_NACK();             Ayarla_TR();             TR_Sayıcı ++;         }     <b>endif</b> <b>endif</b> <b>endif</b> </pre>	<pre> <b><u>Ara Düğüm</u></b> /*Segment alım işlemleri*/ <b>if</b> (Segment Alındı ise)     <b>if</b> (Segment Bellekte yoksa)         BelleğeEkle();         <b>if</b> (Segment_diziNo Yanlış ise)             Ayarla_Segment_İletme();             Segment_Gonder();             BellekKontrolEt();             Gönder_NACK();             Ayarla_TR();             T<sub>R</sub>_Eşik++;         <b>else</b>             RasgeleSec_X_Tmin_Tmax_Arası();             Bekle(X);             Segment_Gonder();         <b>endif</b>     <b>endif</b> <b>endif</b>  /*Eksik segmentleri Gönder*/ <b>if</b> (T<sub>R</sub> taşı ise){     <b>if</b> (T<sub>R</sub>_Eşik&lt;= TR_Sayıcı)         BellekKontrolEt();         <b>if</b> (kayıpSegmentSayısı&gt;0)             Gönder_NACK();             Ayarla_TR();             T<sub>R</sub>_Eşik++;         <b>endif</b>     <b>endif</b> <b>endif</b>  /*NACK alındığında eksikleri Gönder*/ <b>if</b> (NACK Alındı ise)     <b>if</b> (T<sub>Pro</sub> Aktif ise)         BellekKontrolEt();         <b>if</b> (kayıpSegmentler Bellekte yoksa)             gönderNACKKaynağaDogru();         <b>elseif</b> (tümEksikSegmentler varsa)             gönderEksikleriHedefeDogru();         <b>else</b>             gönderEksikleriHedefeDogru();             gönderNACKKaynağaDogru();         <b>endif</b>     <b>endif</b> <b>endif</b> </pre>

#### 4.5. Sonular

Düğömlerin gönderdiği paketlerin zamanında hedefe ulaşmaması bazı kritik uygulamalarda ok önemli sorunlara yol açabilmektedir. Bu soruna özüm üretebilmek için paket kayıplarının hızlı bir şekilde tespit edilerek kurtarılması gerekmektedir. Bu bölümde, paket kayıplarını hızlı bir şekilde tespit ederek ve en uygun kayıp telafi mekanizmasını kullanarak gecikme/enerji duyarlı yeni bir taşıma protokolü geliştirilmiştir. Önerilen yöntem, literatürde var olan kayıp telafi tekniklerinden, ortamdaki hata durumuna göre en uygun olanını kullanarak, güvenilirliği ve enerji korunumunu sağlamaktadır.

## BÖLÜM 5. HİKMET PROTOKOLÜNÜN BAŞARIM DEĞERLENDİRMESİ

### 5.1. Giriş

Bu bölümde, KAA'lar için önerilen HİKMET protokolünün başarımlarını değerlendirme ele alınacaktır. Bölümde ilk olarak benzetim ortamı hakkında bilgi verilerek başlanacaktır. Daha sonra, önerilen protokolün başarımlarını değerlendirmesinde kullanılan Paket Başarım Oranı, Uçtan-uca gecikme, Enerji Tüketimi ve Paket Maliyeti gibi başarımların ölçütleri ile başarımlarını değerlendirmeleri gerçekleştirilecektir. Son olarak da bölüm sonuçları verilerek bölüm sonuçlandırılacaktır.

### 5.2. Benzetim Kriterleri

Önerilen HİKMET protokolünün başarımlarını analiz, ayrıntıları EK.B'de verilen OMNET++ [85] tabanlı benzetim yazılımı altında çalışan Mixim Framework ile gerçekleştirilmiştir. İletişim mesafesi  $r$  olan,  $N$  adet düğüm, uzunluğu  $\ell$  olan bir kare alana konumlandırılmıştır. Bir adet çıkış (sink) düğümü ise alanın uç noktalarından birine yerleştirilmiştir. Düğümlerin güç kapasiteleri, güç tüketimleri, radyo iletim mesafeleri MICAz [86] düğümüne uygun olarak seçilmiştir. Düğümlerin iletim hızı ise 2 Mbit/s olarak seçilmiştir. MAC katmanında, 802.15.4 [87] protokolü kullanılırken yönlendirme katmanında WiseRoute [88] protokolü kullanılmıştır. Benzetimlerde bir paket 50 segmentten oluşmakta ve toplam boyutu 2.5 Kb büyüklüğündedir. HİKMET protokolünün düğümden düğüme mekanizmasında kullanılan parametrelerden  $T_{Max}$  60 ms,  $T_{Min}$  40ms ve  $T_R$  adaptif olarak seçilmiştir. Tablo 5.1'de benzetimde kullanılan parametreler özet olarak görülmektedir.

Önerilen HİKMET protokolünün başarımlarını kıyaslamak için düğüm-den-düğüme hata tespiti ve kurtarma tekniğini kullanan PSFQ [80] protokolü ile uçtan-uca tekniğini

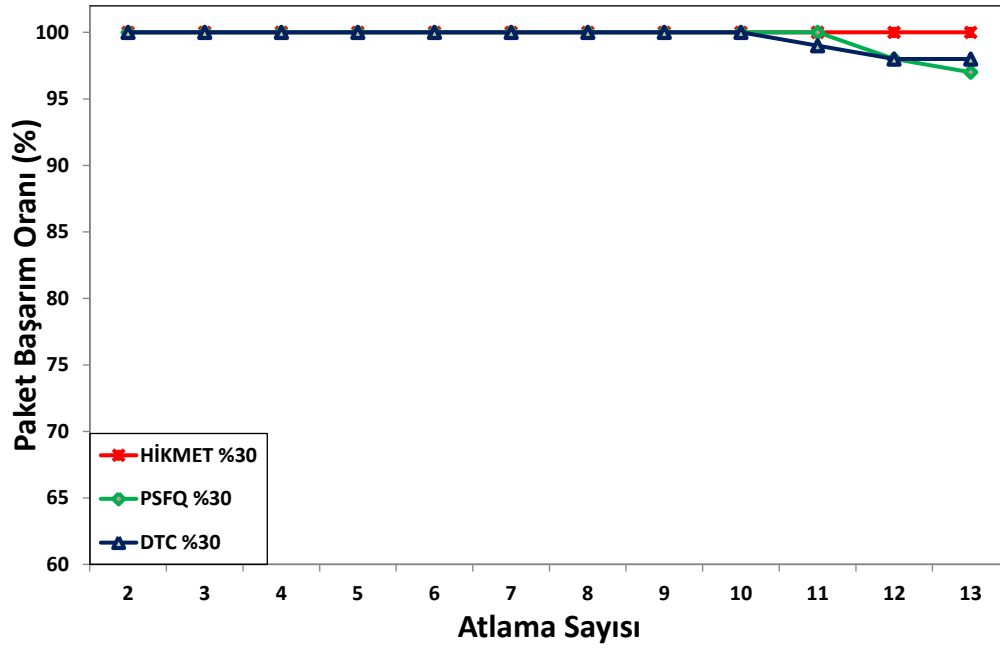
kullanan DTC [76] protokolleri Omnet++ ortamında aslına sadık kalınarak modellenmiştir. Protokollerin başarımlarını değerlendirmesinde paket başarımlar oranı, uçtan-uca gecikme, paket maliyeti ve enerji tüketim miktarı ölçütleri kullanılmıştır.

Tablo 5.1. Benzetim ortamı ayarları

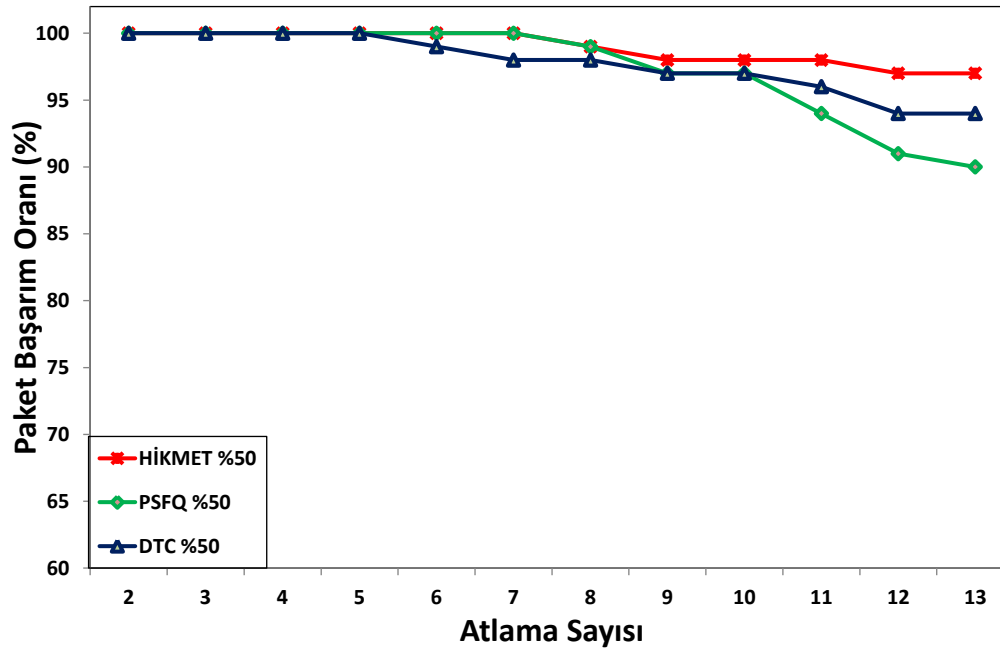
	Değer		
	HİKMET	PSFQ	DTC
Düğüm Sayısı (N)	100		
İletim Mesafesi (r)	100m		
MAC Protokolü	CSMA802.15.4		
Yönlendirme Protokolü	WiseRoute		
İletim Hızı	2 Mbit/sn		
Toplam Segment Sayısı	50		
Paket Boyutu	2.5 Kb		
Bit Hata Oranı	0-80%		
T <sub>Max</sub>	60 ms	100ms	-
T <sub>Min</sub>	40 ms	50 ms	-
T <sub>R</sub>	adaptif		-
NACK Deneme Sayısı (Maks)	adaptif	5	-

### 5.3. Paket Başarımlar Oranı

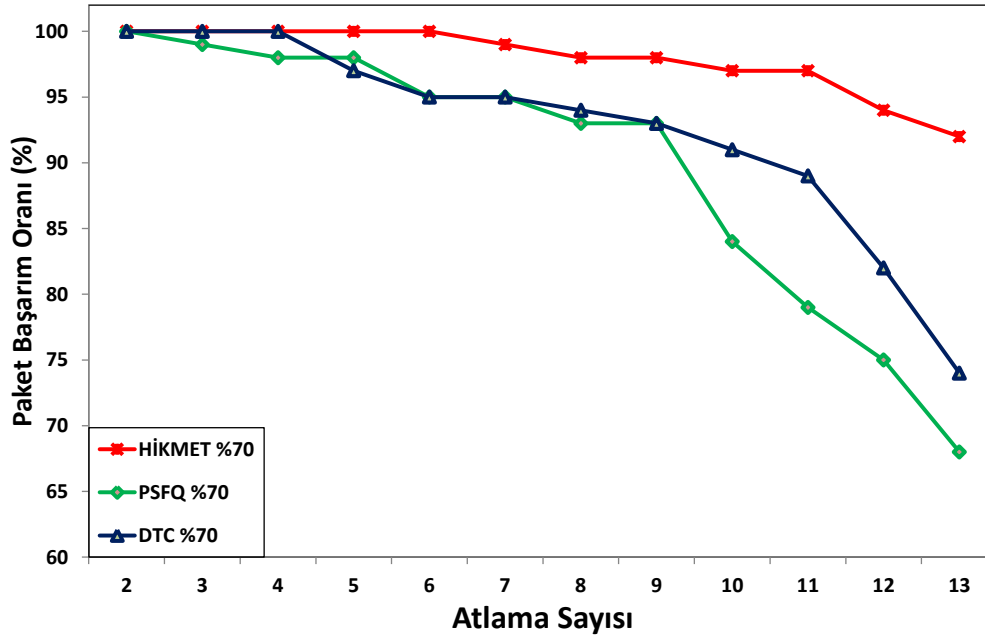
Paket başarımlar oranı, kaynak düğümden gönderilen paketlerin hedef düğüme başarıyla teslim edilme oranını göstermektedir. HİKMET protokolünün paket başarımlar oranını değerlendirmek ve diğer iki protokolle karşılaştırmak için çeşitli benzetimler gerçekleştirdik. Protokollerin farklı ortam koşullarındaki başarımlarını ölçebilmek için %30, %50 ve %70 hata oranlarında ve farklı atlama sayılarında (2-13) paket gönderimleri gerçekleştirilmiştir. Benzetimlerde, kaynak düğümler hedef düğümlere 100'er adet paket gönderimi gerçekleştirmiş ve bu paketlerin hedeflere belirlenen zaman aşımı olmadan ulaşmış ve ulaşmadıkları kontrol edilmiştir. Şekil 5.1'de %30, Şekil 5.2'de %50 ve Şekil 5.3'te %70 hata oranında farklı atlama değerleri için ortalama paket başarımlar oranları görülmektedir.



Şekil 5.1. HİKMET, DTC ve PSFQ protokollerinin farklı atlama değerlerindeki %30 hata oranı için paket başarımlarını gösteren grafik



Şekil 5.2. HİKMET, DTC ve PSFQ protokollerinin farklı atlama değerlerindeki %50 hata oranı için paket başarımlarını gösteren grafik

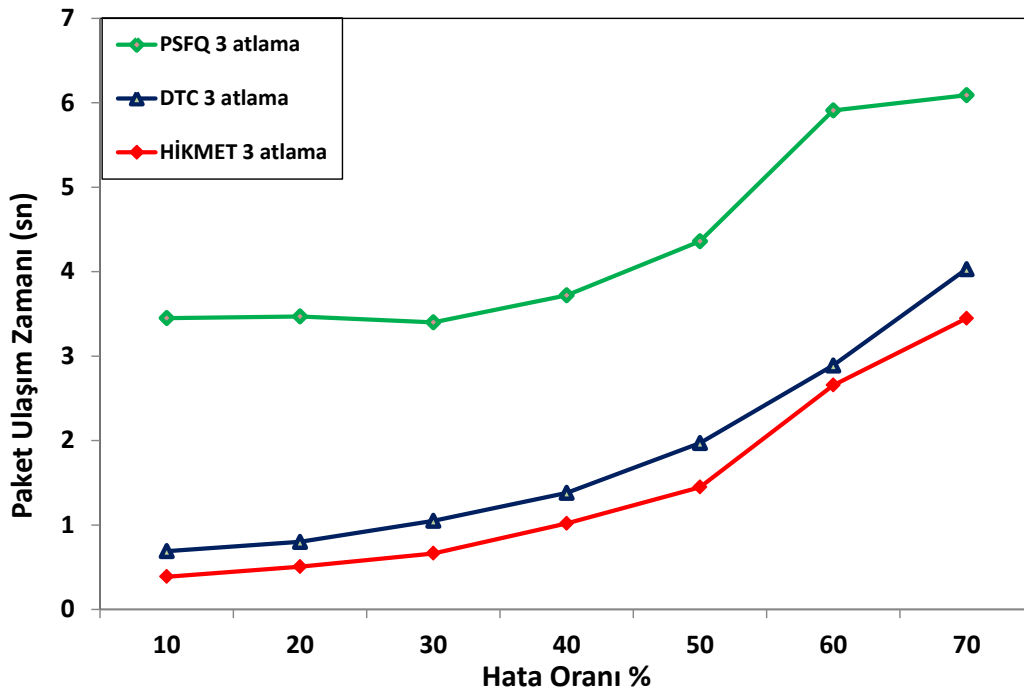


Şekil 5.3. HİKMET, DTC ve PSFQ protokollerinin farklı atlama değerlerindeki %70 hata oranı için paket başarımlar yüzdeleri

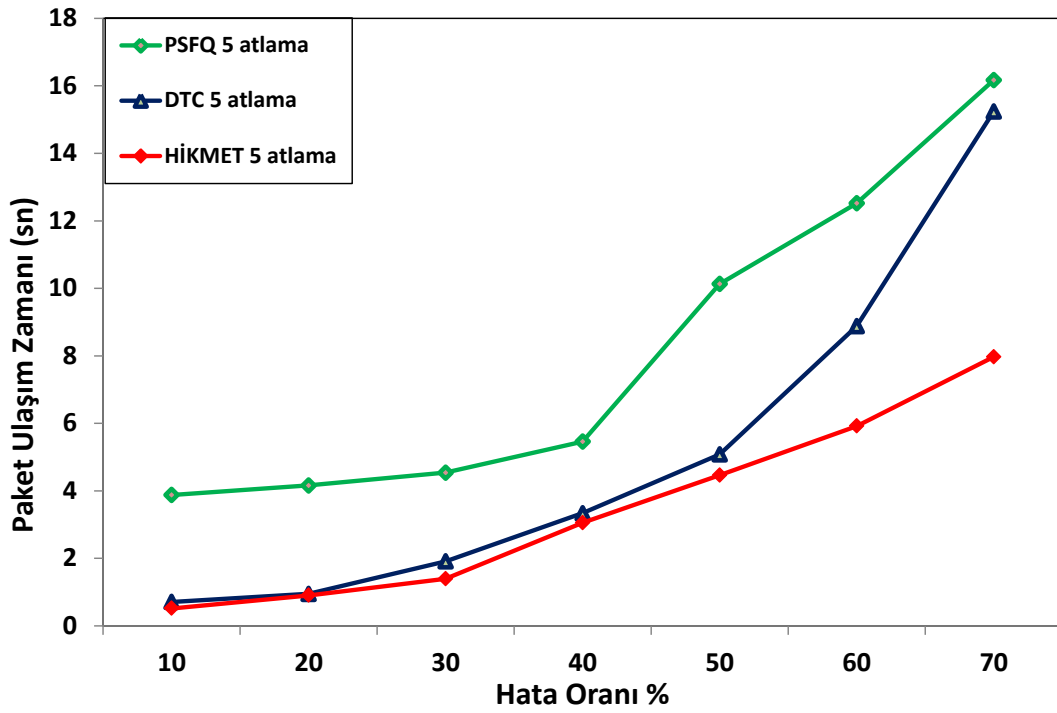
Şekillerden görüldüğü üzere %30 hata oranında özellikle düşük atlama sayıları için 3 protokol de tüm paketleri başarıyla hedeflere iletebilmektedir. Ancak, kaynak ve hedef arasındaki mesafe 10 düğümden fazla olduğunda DTC ile PSFQ protokollerinin paket başarımları düşmektedir. Fakat HİKMET protokolü %30 hata oranında tüm atlama değerleri için %100 paket teslimini gerçekleştirmektedir. %50 hata oranında PSFQ protokolü paketlerin en az %90'ını, DTC protokolü %94'ünü, HİKMET protokolü ise %97'sini başarıyla hedeflere ulaştırabilmiştir. %70 hata oranına çıkıldığında ise, PSFQ ve DTC protokollerinin başarımları önemli ölçüde düşerken, HİKMET protokolü, 100 paketin 92'sini başarıyla ulaştırmıştır. Sonuçlar incelendiğinde, PSFQ ve DTC protokollerinin performansının atlama sayısının artmasıyla önemli ölçüde düştüğü gözlemlenmektedir. Atlama sayısı arttıkça bozulan paketlerin telafi edilmesi daha uzun zaman gerektireceği için paket teslim süresi de uzamaktadır. Dolayısıyla, hedef düğümlere zamanında ulaşamayan paket sayısı da artmakta ve paket başarımları düşmektedir. HİKMET protokolü ise ortam koşullarına göre geliştirilmiş uçtan-uca veya düğümden-düğüme tekniklerini kullandığı için paketleri daha hızlı ve yüksek başarımla gönderebilmektedir.

#### 5.4. Uçtan-Uca Gecikme

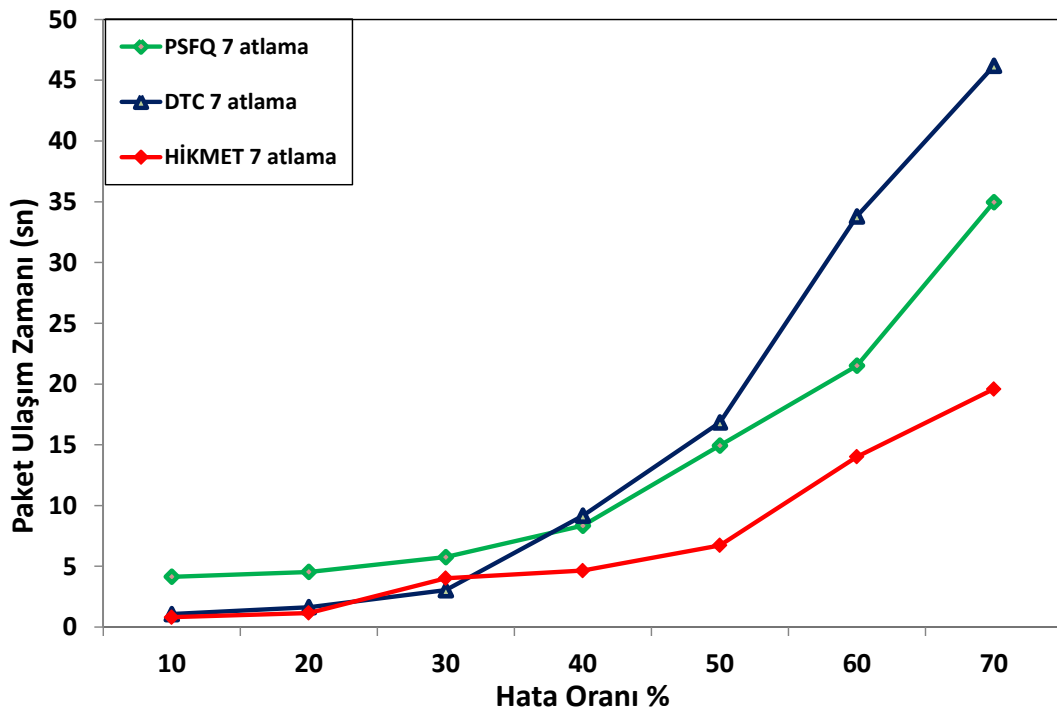
Uçtan-uca gecikme bir paketin kaynaktan hedefe ulaşması sırasında geçen süredir. Uçtan-uca gecikme özellikle gecikmeye duyarlı uygulamalarda son derece önemli bir parametredir. Şekil 5.4'te 3-atlama, Şekil 5.5'te 5-atlama, Şekil 5.6'da 7-atlama, Şekil 5.7'de 9-atlama için HİKMET, DTC ve PSFQ protokollerinin farklı hata ortamlarındaki uçtan-uca gecikme değerleri görülmektedir. Şekiller incelendiğinde düşük hata oranlarında HİKMET ve DTC protokollerinin PSFQ'ye oranla daha düşük gecikme değerleri sağladığı görülmektedir. Bunun sebebi, HİKMET protokolünün düşük hata oranlarında, DTC'nin ise her koşulda uçtan-uca tekniğini kullanmasıdır. PSFQ ise düğümden-düğüme tekniğini kullandığı için diğer iki protokole göre daha yüksek gecikme değerlerine sahiptir. Ancak, yüksek hata oranlarında kayıp paketlerin kurtarılması daha uzadığı için DTC protokolünün gecikme değerleri PSFQ protokolüne yaklaşmaktadır. HİKMET protokolü ise yüksek hata oranlarında düğümden-düğüme tekniğine geçtiği ve proaktif hata kurtarma evresinde ısrarcı NACK gönderimi yaptığı için daha düşük uçtan-uca gecikme değerleri sağlayabilmektedir.



Şekil 5.4. 3-atlamalı bir ağ için HİKMET, PSFQ ve DTC protokolleri için uçtan-uca gecikme değerleri grafiği

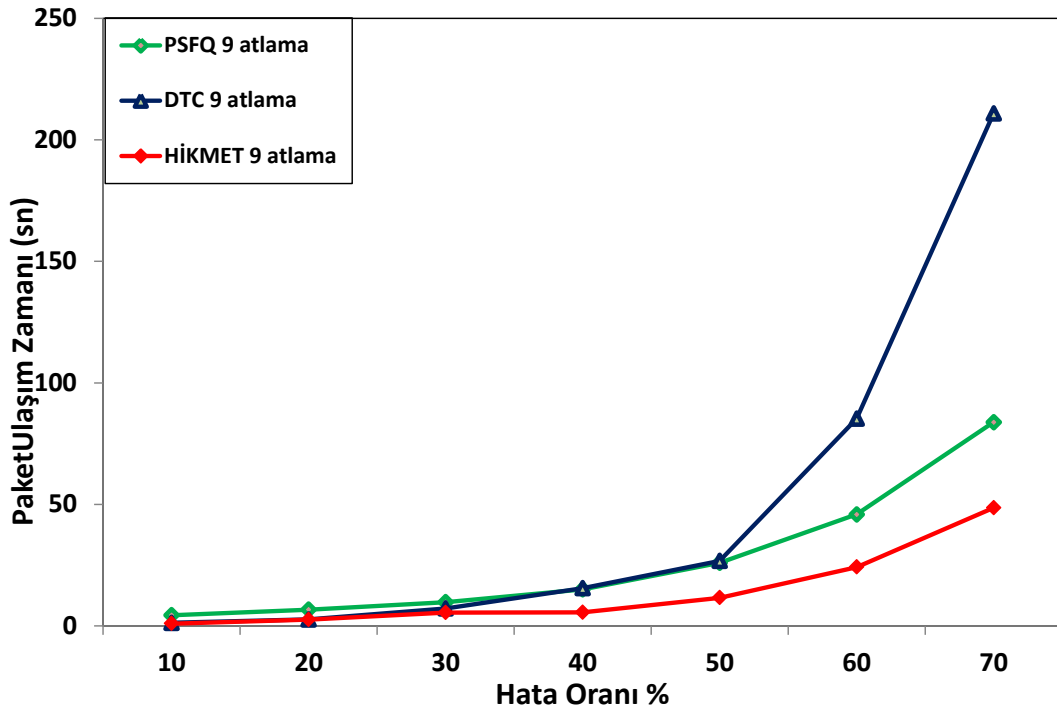


Şekil 5.5. 5-atlamalı bir ağ için HİKMET, PSFQ ve DTC protokolleri için uçtan-uca gecikme değerleri grafiği

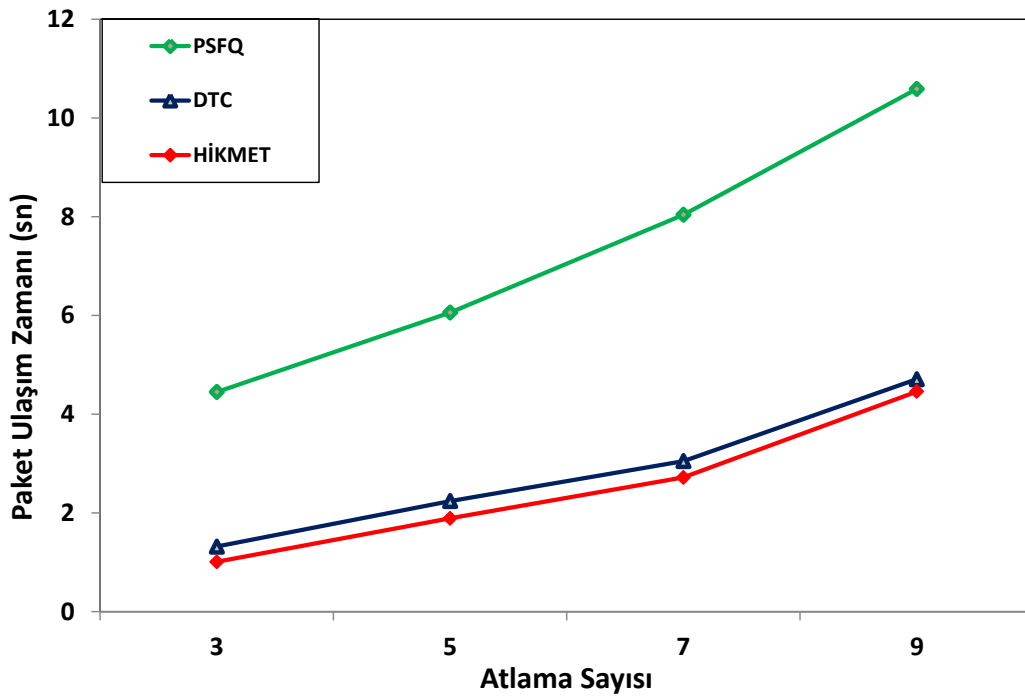


Şekil 5.6. 7-atlamalı bir ağ için HİKMET, PSFQ ve DTC protokolleri için uçtan-uca gecikme değerleri grafiği





Şekil 5.7. 9-atlamalı bir ağ için HİKMET, PSFQ ve DTC protokolleri için uçtan-uca gecikme değerleri grafiği

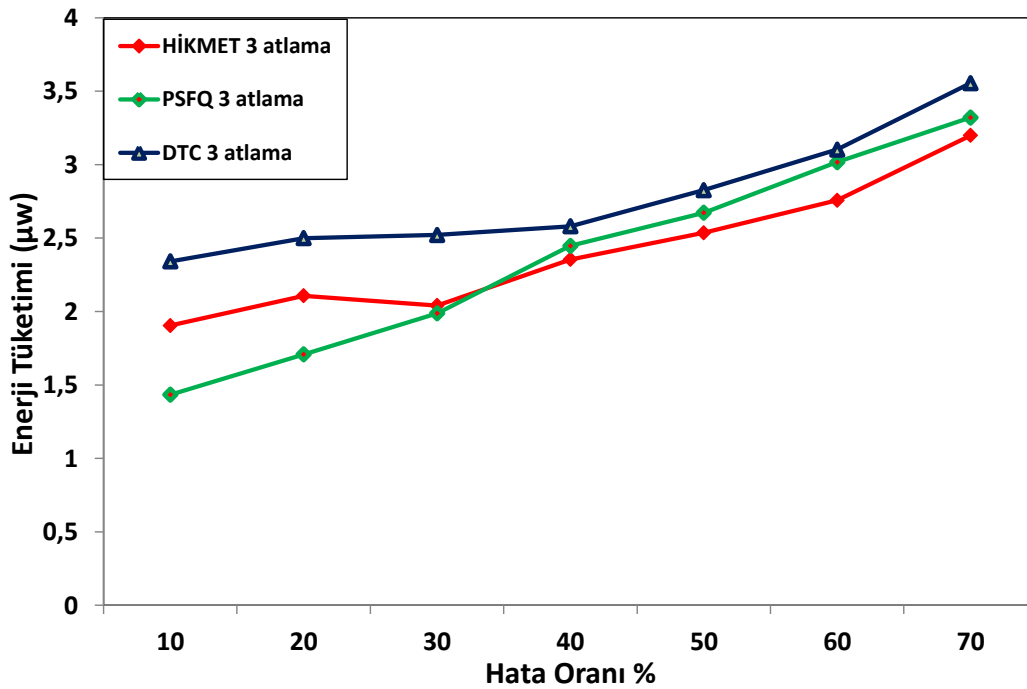


Şekil 5.8. HİKMET, PSFQ ve DTC protokollerinin rasgele hata ortamındaki uçtan-uca gecikme değerleri grafiği

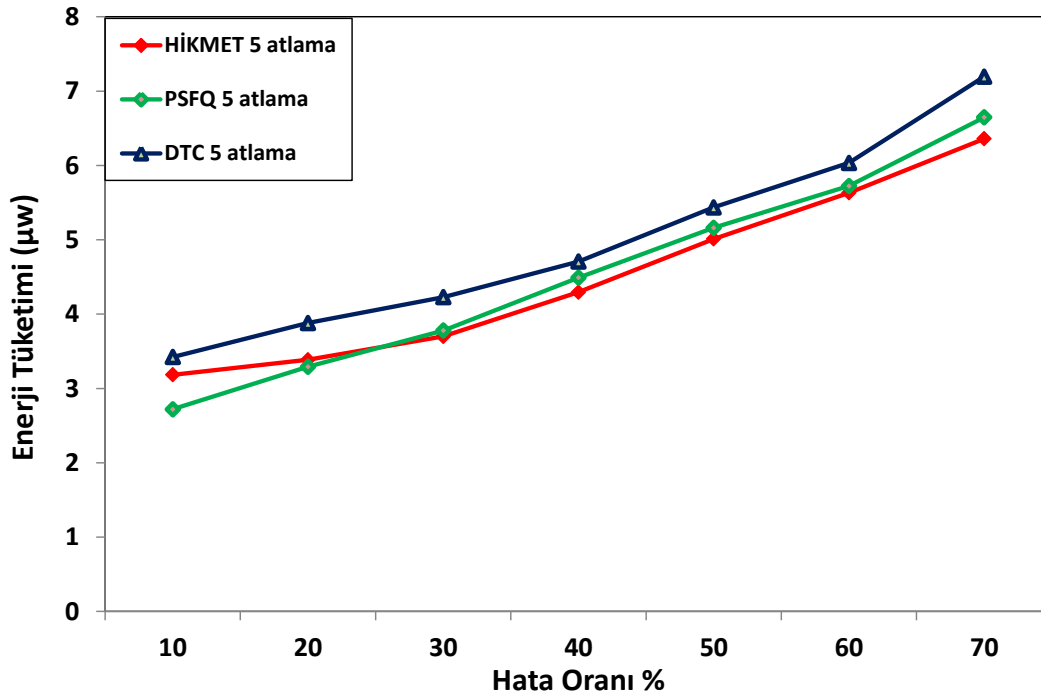
Şekil 5.8’de rasgele hata ortamlarında 1200 sn boyunca çalıştırılan benzetim sonuçları görülmektedir. Elde edilen değerler paket başına ortalama gecikme değerleridir. Şekil incelendiğinde PSFQ protokolünün uçtan-uca gecikme değeri HİKMET ve DTC protokolünün çok üzerinde olduğu görülmektedir. DTC protokolü ile HİKMET kıyaslandığında HİKMET protokolünün her atlama değerinde daha iyi değerler elde ettiği görülmektedir. Örneğin 5-atlamalı bir ağda PSFQ protokolü 1 paketi 6,06 sn, DTC protokolü 2,24 sn’de ulaştırabilirken, HİKMET protokolü 1,89 sn’de hedefe ulaştırmıştır.

### 5.5. Enerji Tüketimi

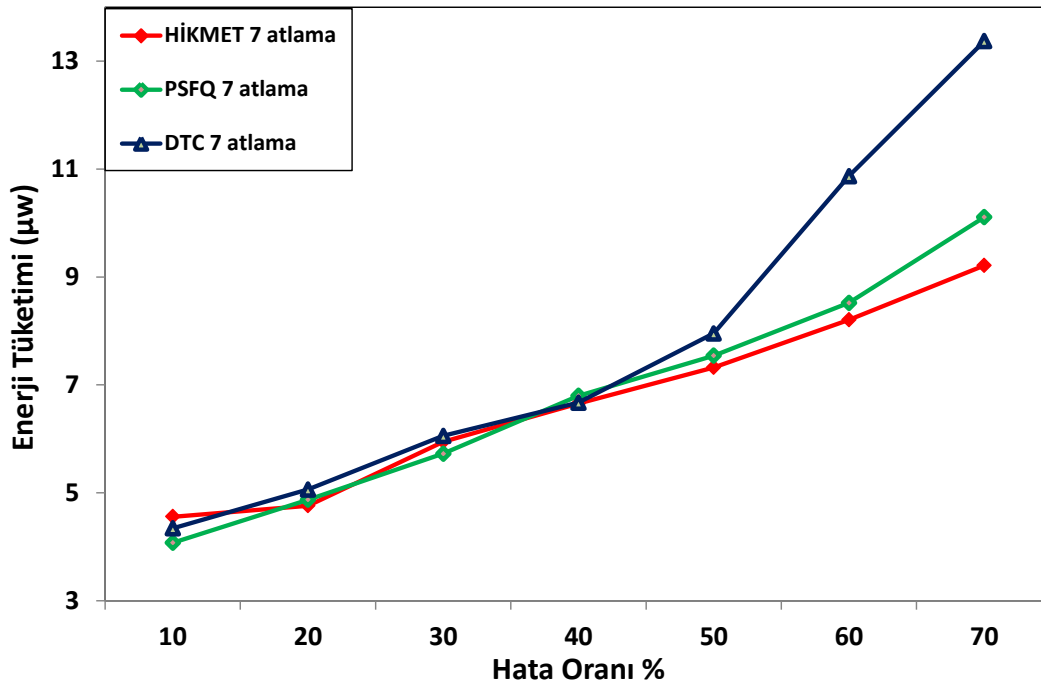
Enerji tüketimi, bir paketin kaynaktan hedefe ulaşması sırasında harcanan toplam enerji miktarıdır. Enerji tüketimi parametresi özellikle, düşük enerjiye sahip uygulamalarda son derece önemli bir parametredir. Şekil 5.9’da 3-atlama, Şekil 5.10’de 5-atlama, Şekil 5.11’de 7-atlama, Şekil 5.12’te 9-atlama için HİKMET, DTC ve PSFQ protokollerinin farklı hata ortamlarındaki paket başına ortalama enerji tüketim değerleri görülmektedir.



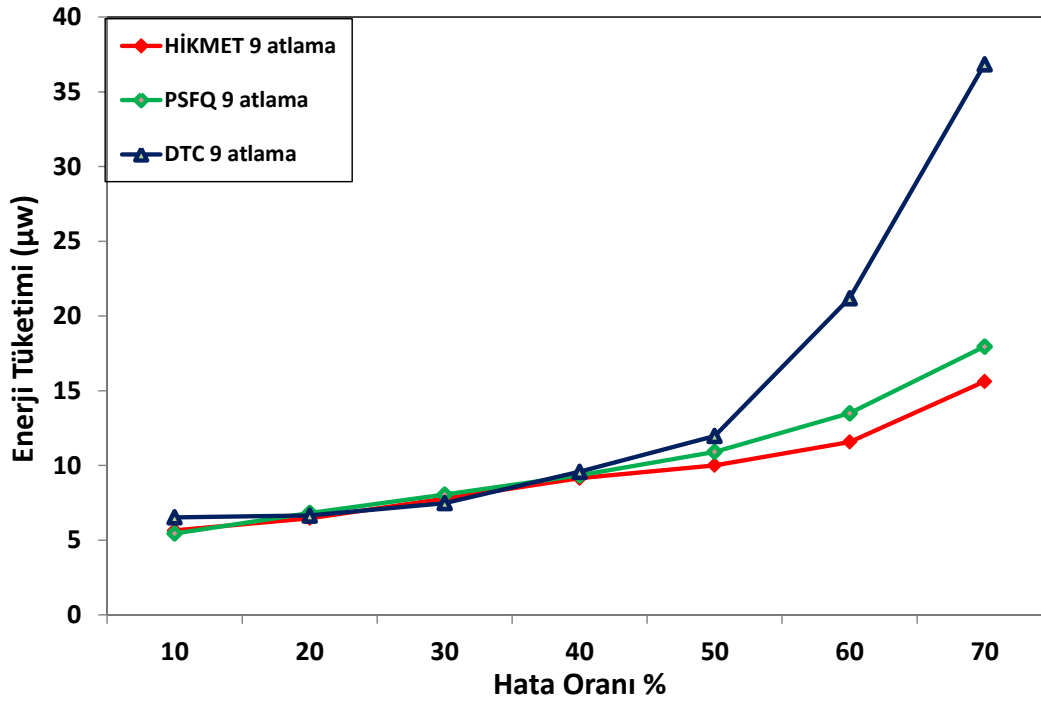
Şekil 5.9. 3-atlamalı bir ağ için HİKMET, PSFQ ve DTC protokollerinin enerji tüketim grafiği



Şekil 5.10. 5-atlamalı bir ağ için HİKMET, PSFQ ve DTC protokollerinin enerji tüketim grafiği



Şekil 5.11. 7-atlamalı bir ağ için HİKMET, PSFQ ve DTC protokollerinin enerji tüketim grafiği

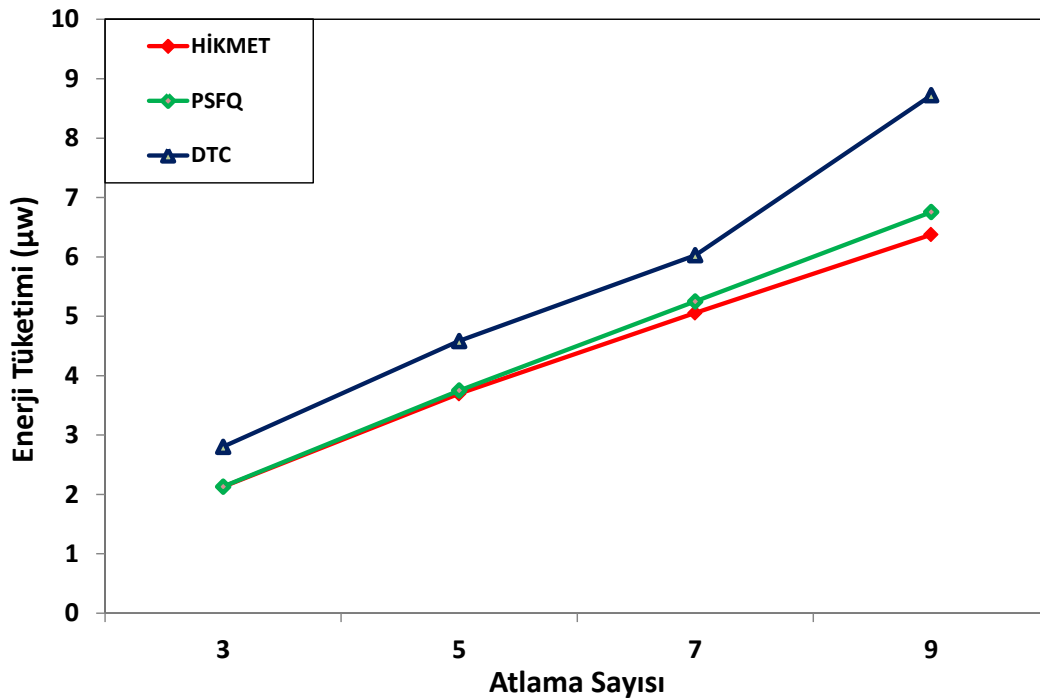


Şekil 5.12. 9-atlamalı bir ağ için HİKMET, PSFQ ve DTC protokollerinin enerji tüketim grafiği

Şekiller incelendiğinde düşük hata oranlarında HİKMET ve DTC protokollerinin PSFQ'ye oranla daha fazla enerji harcadığı görülmektedir. Bunun sebebi, HİKMET ve DTC protokollerinin paket gönderimine başlamadan önce SYNC paketleri göndermesidir. PSFQ protokolü ise düğümden-düğüme tekniğini kullandığı için böyle bir uygulamaya gerek yoktur. Ancak, yüksek hata oranlarında HİKMET protokolü düğümden-düğüme tekniğini kullandığından dolayı enerji tüketimi diğer iki protokole göre daha iyi olmaktadır. Bunun en önemli nedenlerinden birisi de adaptif hata kurtarma yapısını kullanmasıdır. Bu sayede gereksiz segment gönderimleri engellenerek, enerji korunumu sağlanmaktadır. DTC protokolü uçtan-ucaya hata tespiti ve kurtarma tekniğine sahip olduğu için, yüksek hata oranlarında paket bozulmaları aşırı derecede artmakta ve buna bağlı olarak enerji tüketimi çok daha fazla olmaktadır.

Şekil 5.13'te rasgele hata ortamlarında 1200 sn boyunca çalıştırılan benzetimin enerji tüketim sonuçları görülmektedir. Elde edilen değerler paket başına ortalama enerji tüketim değerleridir. Şekil incelendiğinde DTC protokolünün enerji tüketim değeri

HİKMET ve PSFQ protokollerinin çok üzerinde olduğu görülmektedir. HİKMET ile PSFQ protokolleri kıyaslandığında, HİKMET protokolünün düşük atlama değerlerinde PSFQ protokolüne çok yakın enerji tüketim değerleri elde ettiği, yüksek atlama değerlerinde ise HİKMET'un daha az enerji tükettiği görülmektedir. Örneğin 9-atlamalı bir ağda bir paketin kaynaktan hedefe gönderilebilmesi için, DTC protokolü  $8,72\mu w$ , HİKMET protokolü  $6,37\mu w$  ve PSFQ protokolü  $3,75\mu w$  enerji tüketmiştir.

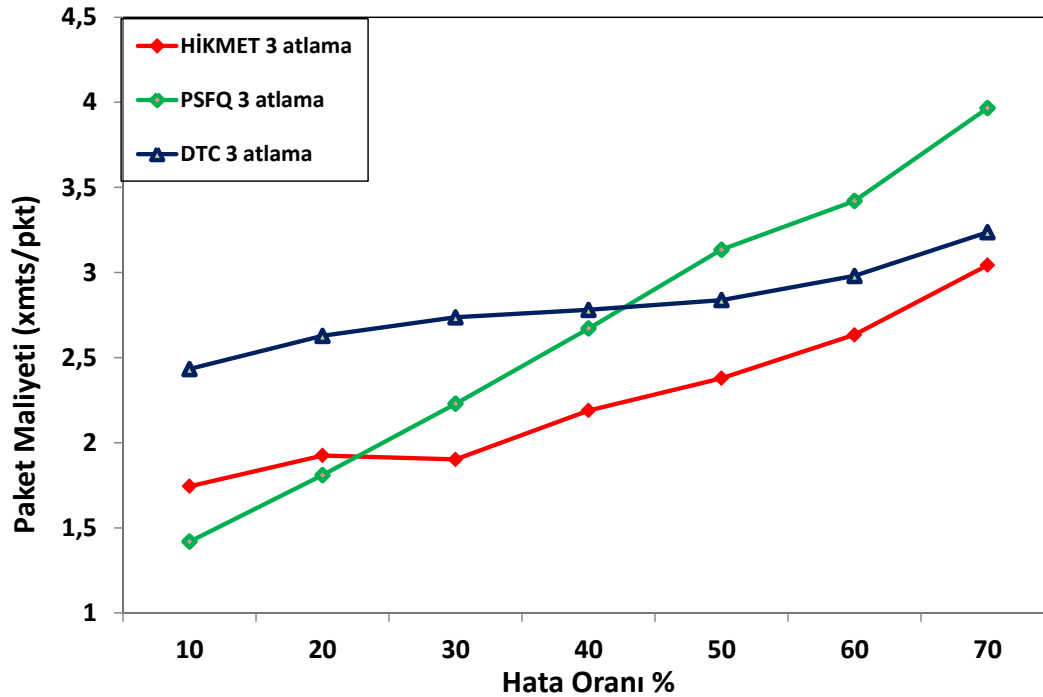


Şekil 5.13. HİKMET,PSFQ ve DTC protokolleri rasgele hata ortamlarındaki paket başına enerji tüketimi grafiği

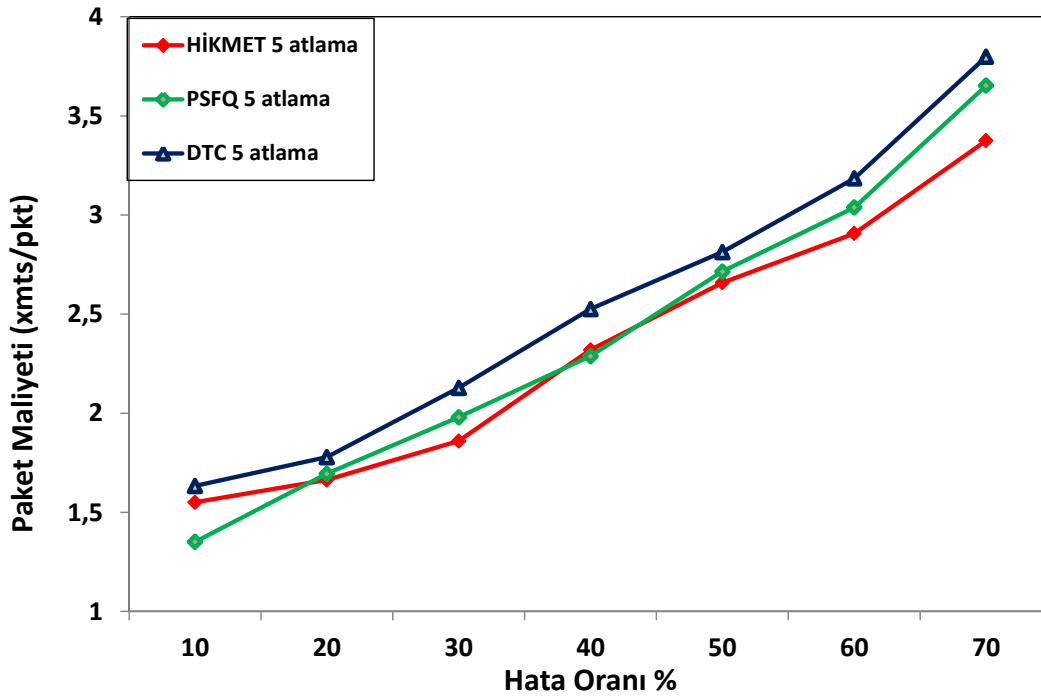
## 5.6. Paket Maliyeti

Paket maliyeti, bir paketin kaynaktan hedefe ulaşması sırasında toplam gönderilen paket sayısının olması gereken paket sayısına oranıdır. Paket maliyeti parametresi ağın performansını belirleyebilmek için kullanılan önemli bir parametredir. Şekil 5.14'de 3-atlama, Şekil 5.15'de 5-atlama, Şekil 5.16'da 7-atlama, Şekil 5.17'de 9-atlama için HİKMET, DTC ve PSFQ protokollerinin farklı hata ortamlarındaki ortalama paket maliyeti değerleri görülmektedir. Şekiller incelendiğinde düşük hata

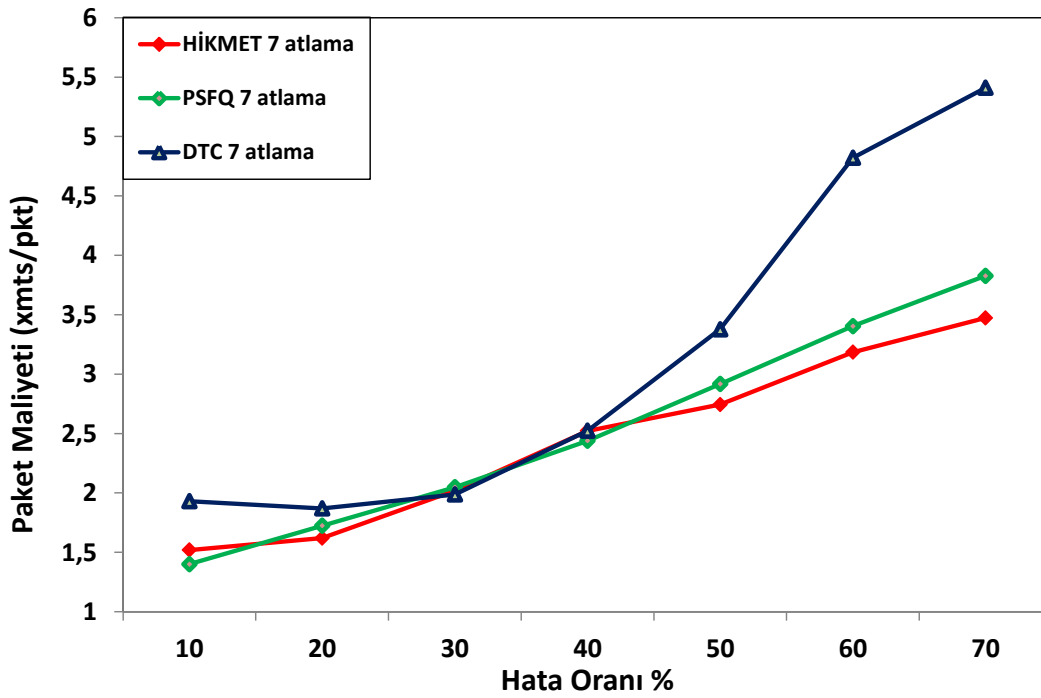
oranlarında HİKMET ve DTC protokollerinin paket maliyet oranının PSFQ'ye oranla biraz daha fazla olduğu görülmektedir. Bunun sebebi, HİKMET ve DTC protokollerinin paket gönderimine başlamadan önce SYNC paketleri göndermesidir. Ancak, yüksek hata oranlarında HİKMET protokolünün paket maliyet oranı diğer iki protokole göre daha iyi olmaktadır. Bunun en önemli nedenlerinden birisi de adaptif hata kurtarma yapısını kullanmasıdır. Bu sayede gereksiz segment gönderimleri engellenerek, paket maliyeti oranını daha da iyi olması sağlanmaktadır. DTC protokolü uçtan-uca tekniğine sahip olduğu için, yüksek hata oranlarında paket bozulmaları aşırı derecede artmakta ve buna bağlı olarak paket maliyet oranı daha fazla olmaktadır.



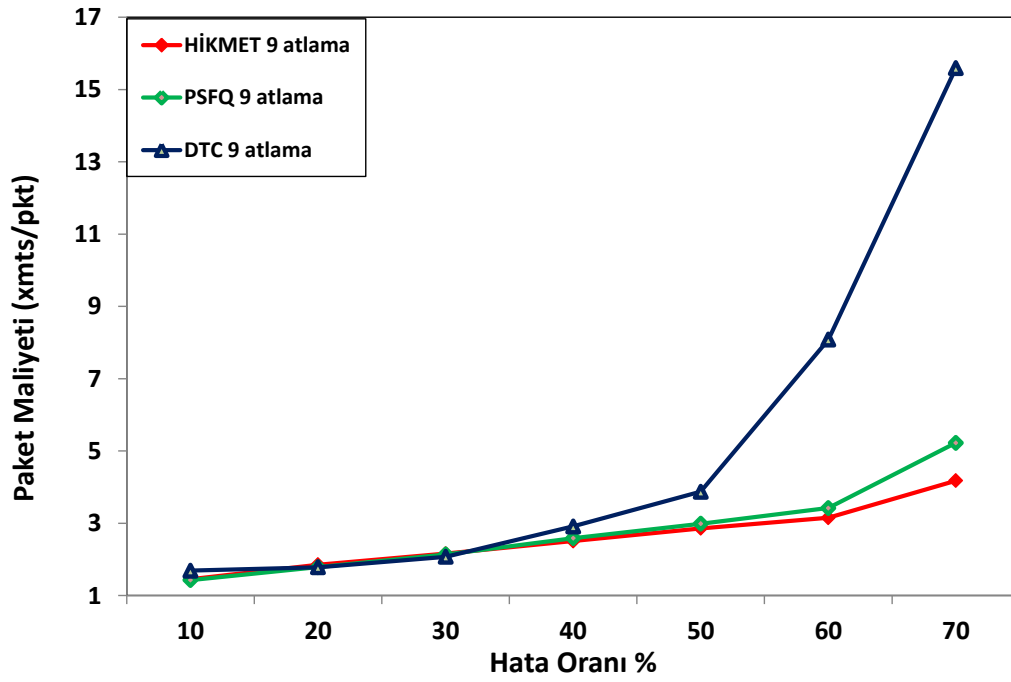
Şekil 5.14. 3-atlamalı bir ağ için HİKMET, PSFQ ve DTC protokollerinin paket maliyeti grafiği



Şekil 5.15. 5-atlamalı bir ağ için HİKMET, PSFQ ve DTC protokollerinin paket maliyeti grafiği

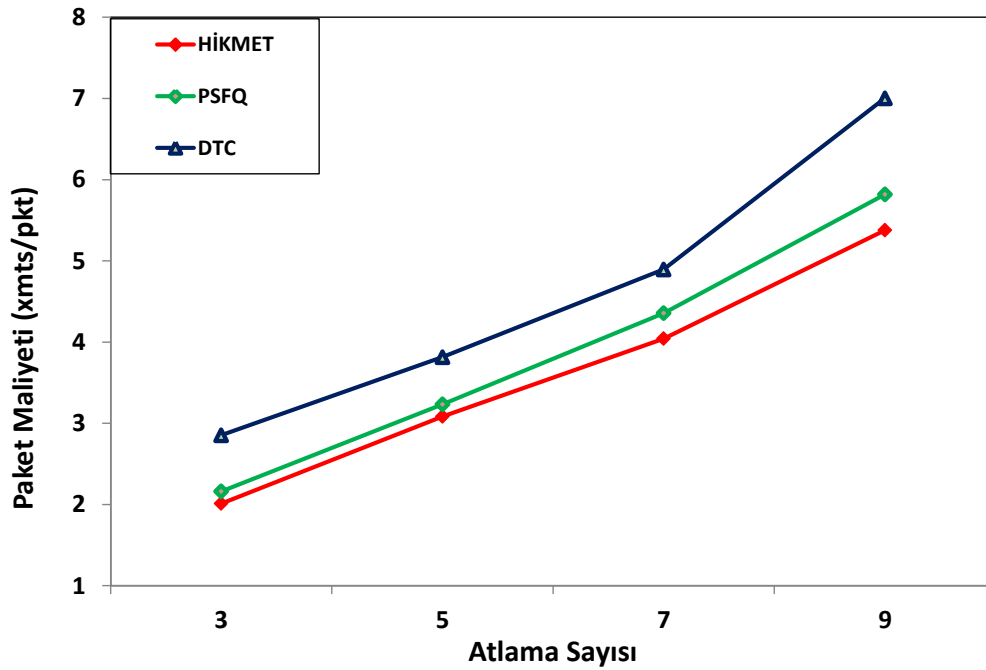


Şekil 5.16. 7-atlamalı bir ağ için HİKMET, PSFQ ve DTC protokollerinin paket maliyeti grafiği



Şekil 5.17. 9-atlamalı bir ağ için HİKMET, PSFQ ve DTC protokollerinin paket maliyeti grafiği

Şekil 5.18’de rasgele hata ortamlarında 1200 sn boyunca çalıştırılan benzetimin paket maliyet oranı sonuçları görülmektedir.



Şekil 5.18. HİKMET, PSFQ ve DTC protokolleri için rasgele hata ortamlarındaki ortalama paket maliyeti grafiği



Elde edilen deęerler paket başına ortalama maliyet oranı deęerleridir. Şekil incelendiğinde DTC protokolünün paket maliyet oranı, HİKMET ve PSFQ protokollerinin üzerinde olduęu görölmektedir. HİKMET ile PSFQ protokolleri kıyaslandığında, HİKMET protokolünün düşük atlama deęerlerinde PSFQ protokolüne çok yakın enerji paket maliyet oranı elde ettięi, yüksek atlama deęerlerinde ise HİKMET'in daha az paket maliyet oranına sahip olduęu görölmektedir. Örneęin 9-atlamalı bir aęda bir paketin kaynaktan hedefe gönderilebilmesi için gerekli olan paket maliyet oranı, DTC protokolü için 7,01, HİKMET protokolü için 5,37 ve PSFQ protokolü için 5,81 olmaktadır.

### **5.7. Sonular**

Bu bölümde, paketlerin hızlı ve güvenilir bir şekilde hedefe ulaşmasını garantileyen HİKMET protokolünün başarımlarını deęerlendirme sonuçları verilmiştir. HİKMET protokolünün başarımlarını, literatürdeki popüler iki taşıma protokolü ile kıyaslanarak analiz edilmiştir. Benzetim sonuçlarına göre, HİKMET, en kötü koşullarda bile paketlerin %92'sini başarıyla iletebilmiştir. Bunun dışında, HİKMET protokolü, PSFQ protokolünün 1/3 oranında, DTC protokolünün de 1/2'si oranında bir sürede paketleri uçtan-uca teslim edebilmektedir. Ayrıca, HİKMET protokolünün enerji tüketimi ve paket maliyetleri bu iki protokole oranla daha düşüktür.

## **BÖLÜM 6. SONUÇLAR VE DEĞERLENDİRME**

### **6.1. Sonuçlar**

Birçok KAA uygulamasında paketlerin hızlı ve güvenilir bir şekilde hedefe ulaştırılması son derece önemli bir gerekliliktir. Bunun yanı sıra, düğümler için büyük önem arz eden enerji verimliliği de dikkate alınması gereken önemli bir diğer ölçütdür. Bu çalışmada, KAA'lar için kayıp telafi yöntemini hibrit olarak kullanabilen güvenilir, enerji ve gecikme duyarlı bir taşıma protokolü tasarlanmış ve benzetim metoduyla detaylı başarımlar değerlendirilmiştir.

Tez çalışması kapsamında elde edilen sonuçların özetleri aşağıdaki gibi sıralanmıştır.

1. Literatürde var olan taşıma katman protokolleri detaylı ve sistematik bir şekilde incelenerek kapsamlı bir literatür taraması gerçekleştirilmiştir.

Taşıma katman protokolleri Bölüm 3'de ilk olarak çeşitli açılardan sınıflandırılmış ve literatürde yer bulan popüler taşıma katman protokolleri bu sınıflara göre tek tek ele alınarak detaylandırılmıştır. Bu sayede, KAA'larda taşıma katmanı konusu üzerine çalışacak araştırmacılar için önemli bir referans çalışma oluşturulmuştur.

2. Taşıma katmanında kayıp telafi yönteminin ortam koşullarına göre adaptif olarak değiştirilmesi ve hibrit bir şekilde kullanılabileceği literatürde ilk kez bu tez kapsamında önerilmiştir.

Literatür incelendiğinde geleneksel ağlarda ve KAA'larda meydana gelen uçtan-uca paket hataları ya düğümden düğüme veya uçtan-uca kayıp telafi

yöntemleri ile giderilmeye çalışılmıştır. Ancak, bu tez çalışmasında bu iki kayıp telafi yönteminin hibrit bir şekilde kullanılabilceği önerilmiştir.

3. Tez kapsamında önerilen hibrit kayıp telafi yönteminin kullanılması fikrine dayanan yeni bir taşıma katmanı protokolü tasarlanmıştır.

Kısaca HİKMET olarak adlandırılan protokol, düğümden-düğüme ve uçtan-uca kayıp telafi mekanizmalarını ortam koşullarına göre beraber bir şekilde kullanarak yüksek güvenilirlik, düşük enerji tüketimi ve düşük uçtan-uca gecikme sağlayabilmektedir.

4. HİKMET protokolü içerisinde tasarlanan ve tekil olarak da kullanılabilcek iki ayrı taşıma katman protokolü de önerilmiştir.

HİKMET protokolü, aslında düğümden-düğüme ve uçtan-uca kayıp paket telafisini gerçekleştiren iki protokol ve bunların ortam koşullarına göre hibrit bir şekilde kullanılmasına sağlayan karar modüllerinin birleşiminden meydana gelmiştir. Bireysel olarak da yüksek başarıya sahip olan bu iki protokol istenirse, ayrı ayrı da kullanılabilir.

5. Tez kapsamında, önerilen taşıma katmanının ve literatürdeki bazı taşıma katman protokollerinin gerçekleştiği OMNET++ ve MIXIM tabanlı ücretsiz benzetim yazılımı tasarlanmıştır.

Kablosuz algılayıcı ağların tüm özellikleri içeren ve hazır modellerin bulunduğu bir ticari benzetim yazılımının henüz bulunmaması kendi benzetim yazılımımızı geliştirmemizi gerektirmiştir. OMNET++ tabanlı olan benzetim yazılımı grafiksel kullanıcı arabirim desteği ve modüler yapısı ile kablosuz algılayıcı ağlarının birçok fonksiyonun benzetilmesine olanak sağlamaktadır. Ayrıca literatürdeki var olan bazı protokollerin benzetim ortamında hazır olarak bulunması gelecekte bu konu üzerine çalışacak araştırmacıların işlerini oldukça kolaylaştıracaktır.

## 6.2. Tartışma ve Öneriler

Bu tez çalışmasında, KAA'lar için yüksek güvenilirliğe sahip bir taşıma protokolünün tasarımı gerçekleştirilmiştir. Bu tezden elde edilen sonuçlar ve katkılar doğrultusunda gelecekte yapılabilecek çalışmalar şunlardır:

1. Tez kapsamında önerilen hibrit kayıp telafi yöntemi, literatürdeki KAA taşıma katman protokollerine entegre edilerek başarımları daha da yükseltilebilir.
2. Önerilen hibrit kayıp telafi yöntemi geleneksel ağlarda kullanılacak taşıma katmanlarında da kullanılarak daha başarımlı yüksek taşıma protokolleri tasarlanabilir.
3. Hibrit kayıp telafi yönteminin karar verme başarımlı farklı sistem parametrelerinin ve farklı yapay zeka tekniklerinin kullanımı ile arttırılabilir.
4. Önerilen HİKMET protokolü benzetim yoluyla gerçekleştirilmiştir. Tasarlanan protokol, günümüz algılayıcı düğümleri için TinyOS v.b. işletim sistemleri üzerinde gerçekleştirilerek gerçek dünya başarımları elde edilebilir.
5. Önerilen HİKMET protokolüne tıkanıklık tespiti ve Giderme modülleri eklenerek daha bütüncül çalışan bir taşıma katman protokolü tasarlanabilir.
6. Önerilen HİKMET protokolü Veri bağı ve yönlendirme katmanları gibi diğer katmanlar ile ortak çalıştırarak daha verimli çalışan yeni bir taşıma katmanı tasarlanabilir.

## KAYNAKLAR

- [1] RATHNAYAKA, A.J. DINUSHA AND POTDAR, VIDYASAGAR M., Wireless Sensor Network transport protocol: A critical review., Journal of Network and Computer Applications. 36 (1), pp. 134-146., 2011.
- [2] D. G. COSTA AND L. A. GUEDES, "A Survey on Transport Protocols for Wireless Multimedia Sensor Networks", KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS, vol. 6, no. 1, 2012.
- [3] SANKARASUBRAMANIAM Y, AKAN O B, AKYILDIZ I F, ESRT: event-to-sink reliable transport in wireless sensor networks. In: Proc. 4th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc '03), New York, NY, USA, pp. 177-188., 2003.
- [4] IYER Y G, GANDHAM S, VENKATESAN S, STCP: A generic transport layer protocol for wireless sensor networks. In: Proc. IEEE ICCCN, San Diego, CA, USA, 2005.
- [5] FAISAL KARIM SHAIKH, ABDELMAJID KHELIL, AZAD ALI AND NEERAJ SURI, TRCCIT: Tunable Reliability with Congestion Control for Information Transport in Wireless Sensor Networks, In: Proceedings of the international wireless internet conference (WICON). Singapore, 2010.
- [6] ALAM M, HONG CS, CRRT: Congestion-Aware and Rate-Controlled Reliable Transport in Wireless Sensor Networks., IEICE Transactions on Communications, E92(B), 184-189, 2009.
- [7] VEHBİ CAGRI GUNGOR, MEMBER, IEEE, ÖZGÜR B. AKAN, A Real-Time and Reliable Transport (RT)<sup>2</sup> Protocol for Wireless Sensor and Actor Networks, IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 16, NO. 2, APRIL 2008.
- [8] TEZCAN N, WANG W., ART: An Asymmetric and Reliable Transport mechanism for wireless sensor networks., International Journal of Sensor Networks, pp 188–200, 2007.
- [9] PAEK J, GOVINDAN R., RCRT: Rate Controlled Reliable Transport for Wireless Sensor Networks., In: Proceedings of the 5th international conference on embedded networked sensor systems. ,Sydney, Australia, pp. 305-319., 2007.

- [10] KIM S, FONSECA R, DUTTA P, TAVAKOLI A, CULLER D, LEVIS P, Flush: a reliable bulk transport protocol for multihop wireless networks. In: Proceedings of the 5th international conference on Embedded networked sensor systems. Sydney, Australia, pp. 351–365, 2007.
- [11] ZHOU Y, LYU MR., PORT: A Price-Oriented Reliable Transport Protocol for Wireless Sensor Network., In: Proceedings of 16th IEEE international symposium on software reliability engineering., Chicago, pp. 117-126, 2005.
- [12] RAEES KHAN, FARRUKH ASLAM KHAN, SRCP: Sensor Reliability and Congestion Control Protocol, 2010 10th IEEE International Conference on Computer and Information Technology, Bradford, West Yorkshire, UK, 2010.
- [13] WAN C.Y, EISENMAN S. B, CAMPBELL A. T, CODA: Congestion Detection and Avoidance in Sensor Networks. In: Proc. the First International Conference on Embedded Networked Sensor Systems (SenSys'03), Los Angeles, CA, USA, pp. 266-279, 2003.
- [14] WANG C, SOHRABY K, LI B, SenTCP: A hop-by-hop congestion control protocol for wireless sensor networks. In: Proc. IEEE INFOCOM, Miami, USA, 2005.
- [15] LEVIS P, PATEL N, CULLER D, SHENKER, Trickle: A self regulating algorithm for code propagation and maintenance in wireless sensor networks. In: Proc. First Symposium Networked Sys. Design and Implementation (NSDI), 2004.
- [16] HULL B, JAMIESON K, BALAKRISHNAN H, Mitigating congestion in wireless sensor networks, In: Proc. ACM Sensys 04, Baltimore, MD, USA, 2004.
- [17] EE C. T, BAJCSY R, Congestion control and fairness for many-to-one routing in Sensor networks. In: Proc. ACM Sensys 04, Baltimore, MD, USA, pp. 148–161, 2004.
- [18] WANG C, SOHRABY K, LAWRENCE V, LI B, HU Y, Priority-based congestion control in wireless sensor networks. In: Proc. IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing, pp. 22–31, 2006.
- [19] WAN C -Y, EISENMAN S B, CAMPBELL A T, CROWCROFT J, Siphon: over-load traffic management using multi-radio virtual sinks in sensor networks. In: Proc. ACM SenSys 05, San Diego, California, USA, pp. 116–129, 2005.

- [20] A. WOO AND D. C. CULLER, A transmission control scheme for media access in sensor networks, in Proceedings of 2001 ACM Annual International Conference on Mobile Computing and Networking (Mobicom '01), Rome, Italy, pp. 221 – 235, July 2004.
- [21] ANTHONY D. WOOD, JOHN A. STANKOVIĆ, AND GANSG ZHOU, DEEJAM Defeating Energy-Efficient Jamming in IEEE 802.15.4-based Wireless Networks, SECON 2007, San Diego, California, USA, June 2007.
- [22] J. P. SHEU, L.J. CHANG, W.K. HU, Hybrid Congestion Control Protocol in Wireless Sensor Networks. // In Proceedings of Journal of Information Science and Engineering, pp 1103-1119, 2009.
- [23] I. F. AKYILDIZ, T. MELODIA and K. R. CHOWDURY, A survey on wireless multimedia sensor networks, Computer Networks ( Elsevier ) , vol. 51, no. 4, pp. 921 – 960., Mar. 2007.
- [24] D. CHEN AND P.K. VARSHNEY, QoS Support in Wireless Sensor Networks: A Survey, in Proceedings of 2004 International Conference on Wireless Networks (ICWN '04), Las Vegas, NV, June 2004.
- [25] S. ZAFAR, A survey of Transport Layer Protocols for Wireless Sensor Networks, International Journal of Computer Applications (0975 – 8887), International Journal of Computer Applications (0975 – 8887).
- [26] ÇAKIROĞLU, M., Kablosuz Algılayıcı Ağlar İçin Dinamik Kanal Atlamalı Güvenlik Sistemi Tasarımı, Doktora Tezi, Sakarya Üniversitesi, Fen Bilimleri Enstitüsü, 2008.
- [27] AKYILDIZ I. F., SU W., SANKARASUBRAMANIAM Y., CAYIRCI E., Wireless Sensor Networks: A Survey, Computer Networks, Vol. 38, No. 4, pp. 393-422, March 2002.
- [28] CHONG C., KUMAR S.P., Sensor Networks: Evolution, Opportunities, and Challenges, in Proceedings of IEEE, vol.91, no.8, pp.1247-1256, 2003.
- [29] STANKUNAS, D.RUDINSKAS, E.LASAUSKAS, Experimental Research of Wireless Sensor Network Application in Aviation, Electronics and Electrical Engineering. ,Kaunas: Technologija, No. 5 (111). , pp. 41–44, 2011.
- [30] R.C. JOHNSON., Sandia enlists MEMS for anti-terror systems.” EE Times,. URL <http://www.eet.com/at/>, March 2002.
- [31] NIE ZEDONG; GUAN FENG; HUANG JIN, Low Power Single-Chip RF Transceiver for Human Body Communication., CHINA COMMUNICATIONS, Vol 9, Issue: 9, pp. 1-10, 2012.

- [32] M. CERNY, M. PENHAKER, Wireless Body Sensor Network in Health Maintenance Systems, Electronics and Electrical Engineering. ,Kaunas: Technologija. , No. 9 (115). , pp. 113–116., 2011.
- [33] ANNA PŁAWIAK-M., ANDRZEJ K., Wireless Body Sensor Network-Fundamental Concepts and Application, Przeglądu, Przegląd Elektrotechniczny, nr 12b, pp. 267-268, 2012.
- [34] <http://fiji.eecs.harvard.edu/CodeBlue>, Erişim tarihi: 10.09.2012.
- [35] MAINWARING A.,CULLER D., POLASTRE J., SZEWCZYK R., ANDERSON J.,Wireless sensor networks for habitat monitoring, Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications, pp. 88-97. ACM Press, 2002. ISBN 1-58113-589-0.
- [36] CERPA A., ELSON J., HAMILTON M., ZHAO J., Habitat monitoring: application driver for wireless communications technology, ACM SIGCOMM'2000, Costa Rica, April 2001.
- [37] <http://www.alertsystems.org>, Erişim tarihi: 05.08.2013.
- [38] GEOFFREY W.A, WELSH M., JOHNSON J., RUIZ M., JONATHAN L., Monitoring volcanic eruptions with a wireless sensor network, Technical Report 27-04, Harvard University, 2004.
- [39] RUPPE D., Nations to discuss using nuclear test sensors as tsunami warning system. Global Security Newswire, January 2005. URL [http://www.nti.org/d\\_newswire/issues/print.asp?story\\_id=5FDDA53C-7385-41B0-A0D5-47652595F5CE](http://www.nti.org/d_newswire/issues/print.asp?story_id=5FDDA53C-7385-41B0-A0D5-47652595F5CE).
- [40] XIAORONG C., MINGXUAN L., The authentication of the grid monitoring system for wireless sensor networks, Przeglądu, Przegląd Elektrotechniczny, nr 01a, pp. 252-254, 2013.
- [41] A. ALAYBEYOGLU, K. ERCIYES, A. KANTARCI, O. DAGDEVIREN, Tracking Fast Moving Targets in Wireless Sensor Networks, IETE Technical Review, Vol 27, Issue 1, pp. 46-53, 2010.
- [42] RABAEY J., ARENS E., FEDERSPIEL C., GADGIL A. , M. D., NAZAROFF W., PISTER K.,. OREN S, VARAIYA P., Smart energy distribution and consumption: Information technology as an enabling force, [http:// bwrc.eecs.berkeley.edu / Publications/2001/samrt\\_energy\\_dist\\_consump / SmartEnergy.pdf](http://bwrc.eecs.berkeley.edu/Publications/2001/samrt_energy_dist_consump/SmartEnergy.pdf).
- [43] KNOTT T. , Smart surrogates. Frontiers, 9, URL [http://www.bp.com/liveassets/bp\\_internet/globalbp/STAGING/global\\_assets/images/fr/downloads/ Frontiers\\_magazine\\_issue\\_09\\_smart\\_surrogates.pdf](http://www.bp.com/liveassets/bp_internet/globalbp/STAGING/global_assets/images/fr/downloads/Frontiers_magazine_issue_09_smart_surrogates.pdf),April 2004.



- [44] DAXIN Z. , DANLIN C., Forest fire monitoring system structure and node design based on wireless sensor network, *Przeglądu, Przegląd Elektrotechniczny*, nr 01b, pp. 64-66, 2013.
- [45] CUI QIMEI; DENG JINGANG; ZHANG XUEFEI, Compressive Sensing Based Wireless Localization in Indoor Scenarios, *CHINA COMMUNICATIONS*, Vol 9, Issue: 4, pp. 1-12, APR 2012.
- [46] MARCHI M, GRILOA, NUNESM, DTSN: Distributed Transport for Sensor Networks, In Proceedings of IEEE symposium on computer sand communications (ISCC). Aveiro, Portugal,2007.
- [47] CATLIN W., ECCLES L., MALCHODI L., Smart sensor project takes flight – boeing pressure belt to measure airplane wing stress. InTech, May 2002, <http://www.isa.org/Content/ContentGroups/InTech2/Features/20023/May6/20020531.pdf>.
- [48] Omnet++ benzetim yazılımı, <http://www.omnetpp.org/>, Erişim tarihi: 09.09.2013.
- [49] <http://www.tinyos.net>, TinyOS işletim sistemi resmi web sayfası, Erişim tarihi: 09.09.2013.
- [50] POLASTRE J., A Unifying Link Abstraction for Wireless Sensor Networks, Doktora Tezi, University of California, Berkeley, A.B.D., 2005.
- [51] KOUBÂA A., ALVES M., TOVAR E., Lower protocol layers for Wireless Sensor Networks: A Survey, Teknik Rapor, 2005.
- [52] HEINZELMAN W., CHANDRAKASAN A., BALAKRISHNAN H.. Energy-efficient communication protocol for wireless microsensor networks, In Proc. 33rd Hawaii Intl. Conf. on System Sciences, January 2000.
- [53] PEI G. , CHIEN C., Low power TDMA in large wireless sensor networks, In Military Communications Conference (MILCOM 2001), volume 1, pp. 347–351, Vienna, VA, October 2001.
- [54] RAJENDRAN V., OBRACZKA K., GARCIA-LUNA-ACEVES J., Energy-efficient, collision-free medium access control for wireless sensor networks, In 1st ACM Conf. on Embedded Networked Sensor Systems, pp. 181–192, Los Angeles, CA, November 2003.
- [55] HOESEL L. VAN, HAVINGA P., A lightweight medium access protocol (LMAC) for wireless sensor networks, In 1st Int. Workshop on Networked Sensing Systems (INSS 2004), Tokyo, Japan, June 2004.
- [56] LANGENDOEN K., HALKES G., Energy-Efficient Medium Access Control, *Embedded Systems Handbook*, CRC Press, 2005.

- [57] I. DEMİRKOL, C. ERSOY and F. ALAGOZ, MAC protocols for wireless sensor networks: A survey, *IEEE Communications Magazine*, vol. 44, no. 4, Apr., pp. 115 – 121, 2006.
- [58] SOHRABI K., GAO J., AILAWADHI V., POTTIE G., Protocols for self-organization of a wireless sensor network, *IEEE Personal Communications*, 7(5), pp. 16–27, October 2000.
- [59] GUO C., ZHONG L., RABAEY J., Low power distributed MAC for ad hoc sensor networks, In *IEEE GlobeCom*, San Antonio, AZ, November 2001.
- [60] LIU B. HUA, NIRUPAMA B., PHAM H., JHA S., CSMAC: A Novel DS-CDMA Based MAC Protocol for Wireless Sensor Networks, *GlobeCom*, 2004.
- [61] KARN P., MACA, A new channel access method for packet radio, presented at the ARRL/CRRL Amateur Radio 9th Computer Networking Conf., 1990.
- [62] BHARGHAVAN V., DEMERS A., SHENKER S., ZHANG L., MACAW: A media access protocol for wireless LANs, in *Proc. ACM SIGCOMM Conf.*, vol. 24, , pp. 212–225, Aug. 1994.
- [63] WEI Y., HEIDEMANN J., ESTRIN D., An energy-efficient MAC protocol for wireless sensor networks, *IEEE Infocom*, pp. 1567–1576, NY, June 2002.
- [64] TIJS V. D., LANGENDOEN K, An adaptive energy-efficient MAC protocol for wireless sensor networks, In *Proceedings of the First ACM Conference on Embedded Networked Sensor Systems*, pp. 171–180, Los Angeles, California, USA, November 2003.
- [65] LU G., KRISHNAMACHARI B, RAGHAVENDRA C.S. , An adaptive energy efficient and low-latency MAC for data gathering in wireless sensor networks, *Proceedings of 18th International Parallel and Distributed Processing Symposium*, pp. 224, 26-30 April 2004.
- [66] EL-HOIYDI A., DECOTIGNIE J. D. , Wisemac: An ultra low power MAC protocol for multi-hop wireless sensor networks. In *Algorithmic Aspects of Wireless Sensor Networks: First International Workshop, ALGOSENSORS 2004*, Turku, Finland, July 16, 2004.
- [67] POLSTRE J., HILL J., CULLER D., Versatile low power media access for wireless sensor networks, in *ACM SENSYS 2004*.
- [68] J. N. AL - KARAKI and A. E. KAMAL, Routing techniques in wireless sensor networks: A survey, *IEEE Wireless Communications*, vol. 11, no. 6, pp. 6 – 28,2004.

- [69] C. W. SOHRABY, K. BO LI; DANESHMAND, M. YUEMING HU, A survey of transport protocols for wireless sensor networks, Network, IEEE pp. 34-40, 2006.
- [70] M. ALLMAN et al.,TCP congestion control, IETF RFC2581, Apr. 1999.
- [71] J. B. POSTEL, “ User datagram protocol ” , IETF RFC 768, Aug. 1980.
- [72] ROMER K., MATTERN F.,The design space of wireless sensor networks, In IEEE Wireless Communications, volume 11, pp. 54 - 61. ETH Zurich, Switzerland, December 2004.
- [73] CHONGGANG WANG, KAZEM SOHRABY, BO LI and WEIWEN TANG, Issues of Transport Control Protocols for WirelessSensor Networks, Proceedings of International Conference on Communications, Circuits and Systems, 2005.
- [74] S.-J. PARK, R. VEDANTHAM, R. SIVAKUMAR and I. F. AKYILDIZ, GARUDA: Achieving Effective Reliability for Downstream Communication in Wireless Sensor Networks, IEEE Transactions on Mobile Computing, vol. 7, no. 2, 2008.
- [75] ZHENG J. JAMALIPOUR A., Wireless Sensor Networks A Networking Perspective, A John Wiley & Sons INC. Publication, ISBN: 978-0-470-16763-2, 2009.
- [76] DUNKELS A, VOIGT T, RITTER H, ALONSO J., Distributed TCP caching for wireless sensor networks. In: Proceedings of the 3rd annual mediterranean ad hoc networking workshop. Turkey, 2004.
- [77] STANN F, HEIDEMANN J, RMST: reliable data transport in sensor networks. In: First IEEE International Workshop on Sensor Network Protocols and Applications, Anchorage, AK, USA, pp. 102-112, 2003.
- [78] ZHANGY H, ARORAY A, CHOIZ Y R, GOUDAZ MG, Reliable bursty convergecast in wireless sensor networks. In: Proc. ACM MOBIHOC 05, Ur-bana-Champaign, IL, USA, 2005.
- [79] LE T, HU W, PETER CORKE, JHA S., E RTP: Energy-Efficient and Reliable Transport Protocol for data streaming in wireless sensor networks. Computer Communications, pp. 1154-71. 2009.
- [80] WAN C -Y, CAMPBELL A T, KRISHNAMURTHY L, PSFQ: A reliable transport protocol for wireless sensor networks. In: Proc. ACM International Workshop on Wireless Sensor Networks and Applications, Atlanta, GA, USA, pp. 1–11, 2002.

- [81] H. GUPTA, S. R. DAS, AND Q. GU., Connected Sensor Cover: Self-Organization of Sensor Networks for Efficient Query Execution. In Proc. of ACM Mobihoc, June 2003.
- [82] B. POSTEL, "Transmission control protocol," RFC 793, September 1981.
- [83] DUNKELS A, VOIGT A T, ALONSO J, RITTER H, SCHILLER J, Connecting wireless sensornets with TCP/IP networks. In: Proc. Second International Conference on Wired/Wireless Internet Communications (WWIC2004). Frankfurt, Germany: Springer Verlag, 2004.
- [84] DUNKELS A, ALONSO J, VOIGHT T, Making TCP/IP Viable for wireless sensor networks, In: european workshop on wireless sensor networks (EWSN), Berlin, Germany, 2004.
- [85] M.M. MONOWAR, M.O. RAHMAN, A. K. PATHAN, C. S. HONG, Congestion control protocol for wireless sensor networks handling prioritized heterogeneous traffic, Proceedings of the 5th Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services, 2008.
- [86] CEKEN, C., An Energy Efficient and Delay Sensitive Centralized MAC Protocol for Wireless Sensor Networks, Computer Standards & Interfaces, 30.1-2, 20-31, 2008.
- [87] DOPICO N.I., GIL-SORIANO C., ARRAZOLA I. AND ZAZO, S., Analysis of IEEE 802.15.4 Throughput in Beaconless Mode on micaZ under TinyOS 2, Vehicular Technology Conference Fall (VTC 2010-Fall), 2010 IEEE 72nd, pp. 1-5, 2010.
- [88] D. PIGUET, J. ROUSSELOT, P. DALLEMAGNE AND C. KASSAP OGLOU-FAIST, Routing for Mobile Wireless Sensor Networks, Scientific and Technical Report, 2009.
- [89] DWIVEDI, A.K., VYAS, O.P., An Exploratory Study of Experimental Tools for Wireless Sensor Networks, Wireless Sensor Network, Vol. 3, pp. 215-240, 2011.
- [90] ERTÜRK İ., Internetworking Between ATM LANs and Legacy LANs Over ATM Networks, Doktora Tezi, Sussex University, The School of Engineering and Information Technology, İngiltere, 2000.
- [91] OPNET benzetim yazılımı, OPNET Modeler 11.5 Documentation, OPNET Technologies, Release 11.5, 2006.
- [92] VANGHELuwe, H., Multi-Formalism Modelling and Simulation, Doktora tezi, Universiteit Gent Faculteit Wetenschappen, 2001.

- [93] ANTOINE-SANTONI, T., SANTUCCI, J. F., DE GENTILI, E., COSTA, B., Discrete Event Modeling and Simulation of Wireless Sensor Network Performance, SIMULATION, 84, 2/3, pp. 103-121, 2008.
- [94] Ns-3 benzetim yazılımı, <http://www.nsnam.org/overview/what-is-ns-3/>, Erişim tarihi: 10.09.2013.
- [95] QualNet benzetim yazılımı, <http://www.scalable-networks.com/boards/>, Erişim tarihi: 10.09.2013.
- [96] Glomosim benzetim yazılımı, <http://pcl.cs.ucla.edu/projects/glomosim/>, Erişim tarihi: 10.09.2013.
- [97] Omnet++ benzetim yazılımı, <http://www.omnetpp.org/>, Erişim tarihi: 10.09.2013.
- [98] RAHMAN, M.A., PAKŠTAS, A., WANG, F. Z., Network Modelling and Simulation Tools, Simulation Modelling Practice and Theory, 17, pp. 1011-1031, 2009.
- [99] Omnet++ benzetim yazılımı kullanıcı kılavuzu, <http://www.omnetpp.org/doc/omnetpp/manual/usman.html>, Erişim tarihi: 10.09.2013.
- [100] Omnet++ Mixim modeli, <http://sourceforge.net/apps/trac/mixim/>, Erişim tarihi: 10.09.2013.
- [101] CC1100 radyo alıcısı, <http://www.ti.com/product/cc1100>, Erişim tarihi: 10.09.2013.
- [102] CC2420 radyo alıcısı, <http://www.ti.com/product/cc2420>, Erişim tarihi: 10.09.2013.
- [103] HU, Y., LIN, Y., LIU, Y., LI, X., LIN, M., A Simulation System for Modeling and Analysis of Large Scale Sensor Networks, 6th International Conference on Information Communications & Signal Processing (ICICS), pp. 1-5, 10-13 Dec. 2007.
- [104] XIAN, X. , SHI, W., HUANG, H., Comparison of Omnet++ and Other Simulator for WSN Simulation, 3rd IEEE Conference on Industrial Electronics and Applications (ICIEA), pp. 1439-1443, 3-5 June 2008.
- [105] KOPKE, A., SWIGULSKI, M., WESSEL, K., WILLKOMM, D., HANEVELD, P.T.K, PARKER, T.E.V., VISSER, O.Q., LICHTÉ, H.S., VALENTIN, S., Simulating wireless and mobile networks in Omnet++ the MiXiM vision, Proceedings of the 1 st international conference on Simulation tools and techniques for communications, networks and systems & workshops, France, 2008.

- [106] WESSEL, K., SWIGULSKI, M., KOPKE, A., WILLKOMM, D., Mixim: the physical layer an architecture overview, in Simutools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques (ICST), Brussels, Belgium, pp. 1-8., 2009.

## **EKLER**

### **Ek A. Kablosuz Algılayıcı Ağ Modelleme Ve Benzetim Araçları**

#### **A.1. Giriş**

Benzetim araçları gerçek dünyada meydana gelen davranışlarının bazı kısımlarının taklit edildiği yazılımlardır. Gerçek dünyanın farklı yapılarını modeller ve benzetimini gerçekleştirirler [89].

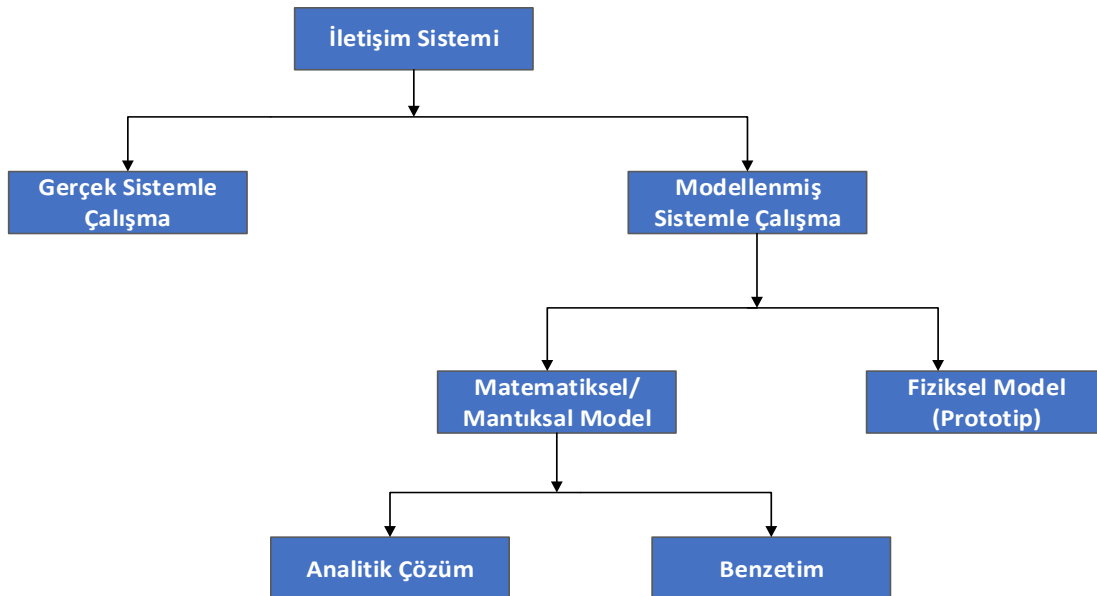
Bir olay gerçekleştirilmeden önce maliyeti daha düşük yöntemlerle çalıştırılır. Sistemin çalışması ve yeterlilik durumları uygun bulunduğu gerçekleştirme işlemi yapılır. Gerçek hayattaki iletişim sistemleri karmaşık yapıya sahiptir ve iletişimin yapıldığı ortam başta olmak üzere sistemin başarımını etkileyen çok sayıda etken vardır. Gerçek hayattaki sistemlerinin başarım analizi gerçek bir sistem kurularak test edilmek istenmesi yüksek maliyetler oluştururken; sistemin model yapısının kullanılmasıyla elde edildiğinde maliyet büyük oranda azalmaktadır [26].

Sistemlerin performans analizi için geleneksel olarak üç teknik kullanılır. Bu yöntemler fiziksel model, analitik çözüm ve benzetim (simülasyon) yöntemleridir. Şekil A.1'de bu teknikler grafiksel olarak görünmektedir. Kablosuz algılayıcı ağların enerji kısıtlılığı, hata toleransı gibi doğasından kaynaklanan sınırlılıklarından dolayı diğer ağlarda yaygın olarak kullanılan analitik metotlar ile kablosuz algılayıcı ağların performans analizinden istenilen başarı elde edilemez [89].

Fiziksel model yönteminde başarım değerlendirilmesi, mevcut sistemin değişik şartlar altında incelenmesiyle elde edilebilir. Fiziksel model, en güvenilir ve en doğru yöntem olmasına rağmen, özellikle karmaşık iletişim sistemleri için planlama ve tasarım aşamaları gibi çeşitli ayarların denenmesinin zorunlu olduğu birçok durumda gerçekleştirilmesi oldukça zordur. Bu yöntem yüksek maliyet ve uzun zaman

gerektiren bir yöntemdir. Analitik model, diğer yöntemler arasında kullanımı en basit olanıdır. Varsayımlar ve ideal kabuller üzerine kurulur. Bu yüzden kesin sonuçlar elde edilmek istendiğinde analitik modeli oluşturmak oldukça zordur. Sistemleri modellemek için kullanılan bir diğer yöntem de benzetimdir. Benzetim somut anlamda belirli bir nesnenin oluşturulan modelidir. Benzetimin bilgisayar ortamında gerçekleştirilmesi işlemine bilgisayar benzetimi adı verilir. Bir sistemi modellemek için kullanılan tüm yöntemler karşılaştırıldığında, olay tabanlı (event-driven) bilgisayar benzetiminin en iyi çözüm sağladığı görülmektedir [90].

Bu tez çalışmasında kablosuz algılayıcı ağlarında kullanılacak, hata tespiti ve kurtarma yapılarını adaptif olarak kullanabilmesine yönelik olarak geliştirilen taşıma katman algoritmasının başarımları için bilgisayar destekli benzetim metodu kullanılmaktadır. Bu bölümde tasarlanan yöntemlerin bilgisayar tabanlı benzetimlerinin gerçekleştirilebilmesi için geliştirilen benzetim ortamının detayları hakkında bilgi verilmektedir.



Şekil A.1. İletişim sistemleri geliştirmede kullanılan yöntemler [26]

## A.2. Benzetim Ortamları

Kablosuz algılayıcı ağlarda donanımsal ve yazılımsal olarak henüz bir standart olmaması popüler birçok benzetim yazılımının kablosuz algılayıcı ağlara yönelik hazır modellerin bulunmamasına neden olmaktadır. Bundan dolayıdır ki, kablosuz



algılayıcı ağlar için geliştirilen protokol, yöntem ya da algoritmaların modellenmesi ve benzetimlerinin gerçekleştirilmesi oldukça zordur [26].

Kablosuz algılayıcı ağları modellemek için kullanılacak 60'dan fazla bilinen benzetim aracı bulunmaktadır [89]. Bunların arasında Opnet [91], VHDL ve Verilog [92], [93], ns-2 [94], Qualnet [95], Glomosim [96] ve Omnet++ [97], en yaygın olarak kullanılanlarıdır. Bu araçların işlevselliği, avantajları, süreçleri ve özellikleri açısından detaylı olarak incelenmesi gerekmesine rağmen hemen hemen tüm ağ benzetim araçlarının benzer yolla çalıştığı açıktır [98].

Bu tez çalışmasında, Omnet++ yazılımının tercih edilmesinin ana nedeni; çalışma zamanı ve bellek kullanımı gibi konularda araştırmacılar tarafından yaygın olarak kullanılan diğer benzetim araçlarına göre daha yüksek performans göstermesidir. Tablo A.1'deki benzetim araçlarının özellikleri incelendiğinde, yüksek çalışma hızı, esnek ve güçlü grafik arabiriminin olması, Omnet++ benzetim aracının seçilmesinde önemli etkenlerdir. Bunların dışında Omnet++ içerisinde kablosuz algılayıcı ağlar için pek çok hazır algoritmanın olması da bu seçimde etkili olmuştur.

Tablo A.1. Yaygın olarak kullanılan benzetim yazılımları ve özellikleri [26]

Özellik	OPNET	QUALNET	NS-2	OMNET++	TOSSIM
Amaç	Genel amaçlı ağ benzeticisi				KAABenzeticisi
Lisans	Ticari	Ticari	Ücretsiz	Ücretsiz	Ücretsiz
Kullanım Kolaylığı	Çok iyi	Çok iyi	Zor	İyi	Çok Zor
Esneklik	İyi	İyi	Orta	Çok İyi	Kötü
Kullanıcı Arabirimi	Güçlü GUI	Güçlü GUI	Yetersiz GUI	Güçlü GUI	Yok
Paralel Çalışabilme	Var	Var	Var	Var	Yok
Ölçeklenebilirlik	Orta	Çok İyi	Orta	İyi	Kötü
Programlama Dili	C++	C++	C++ ve OTcl	C++	NesC
Dokümantasyon	Çok İyi	İyi	İyi	İyi	Orta
Kütüphane	Çok İyi	Çok İyi	İyi	Orta	Orta
Hız	Kötü	Orta	Orta	İyi	Kötü

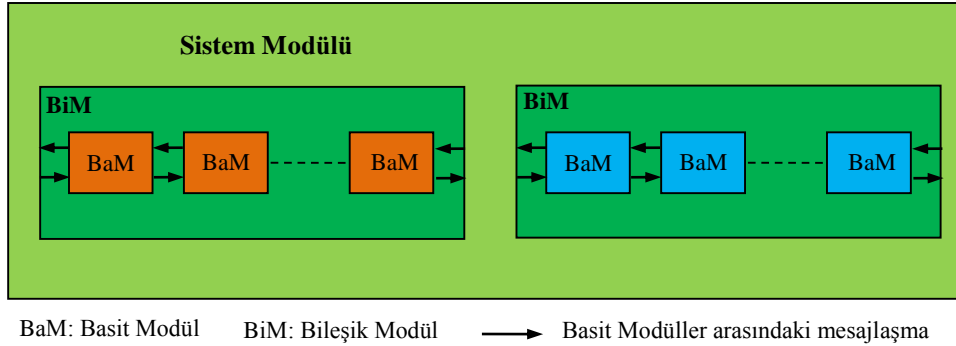
### A.3. OMNET++ benzetim yazılımı

OMNET++ (Objective Modular Network Testbed in C++), nesneye-yönelik (object-oriented) modüler bir ayrık olay ağ benzetim aracıdır ve bu yazılım aşağıda maddeler halinde verilen süreçlerin benzetiminde kullanılabilir.

- Haberleşme trafiğinin modellenmesi
- İletişim protokollerinin modellenmesi
- Çok işlemcili ve diğer dağıtık donanım sistemlerini modelleme
- Donanım yapılarının incelemesi
- Karmaşık sistemlerin başarımlarının analizlerinin değerlendirilmesi
- Ayrık olay yaklaşımının elverişli olduğu diğer sistemlerin modellenmesi [26].

OMNET++ yazılımında bir ağ modeli, Şekil A.2’de görüldüğü gibi iç içe geçmiş modüllerin birleşiminden meydana gelmektedir. En üst seviyedeki modül, sistem modülü ya da ağ olarak isimlendirilir. İç içe geçen modüllerin derinliği kullanıcıya bağlıdır. Modüller, basit ve bileşik olmak üzere iki kategoriye ayrılmaktadır. Basit modül, modellenmek istenen nesnenin davranışlarını tanımlayan C++ dosyasıyla ilişkilendirilir. Bileşik modüller ise basit modüllerin birleşiminden meydana gelmektedir ve doğrudan bir C++ dosyasıyla ilişkili değildir. Modüller kendi aralarından mesajlar göndererek haberleşirler ve benzetim zamanı, bir modül kendisinden veya başka bir modülden mesaj aldığı anda devam eder. OMNET++, zamanlama işlemlerini “self message” adı verilen ve düğümün kendisine gönderdiği mesajlar yardımıyla gerçekleştirir. Modüllerin yapısı ve arabirimleri, Ağ Tanımlama Dili (Network Description Languages – NED) ile oluşturulmakta ve benzetim parametreleri bir ini başlangıç dosyası ile kolaylıkla ayarlanabilmektedir

[26]. C++ ve Tcl/Tk dili ile yazılmış olan OMNET++ benzetim yazılımının avantajları şunlardır;



Şekil A.2. OMNET++ modül yapısı [26]

- Farklı sistemler üzerinde (DOS, UNIX ve Windows) oluşturulan kodlar platformdan bağımsız olarak çalışabilmektedir.
- Görsel kullanıcı arabirim desteği ile kolay hata-ayıklamaya imkân verir.
- Benzetimlerin çalıştırılmasından elde edilen sonuçların vektörel ve sayısal olarak çizilmesine olanak sağlar.
- Paralel yürütme ve çoklu işlemci ile çalışma desteği bulunmaktadır.
- Rasgele sayı üreteçleri, İstatistiksel fonksiyonlar, yoğunluk tahmin fonksiyonları, yönlendirme ve topoloji desteği gibi gelişmiş bir benzetim kütüphanesine sahiptir.
- Benzetimler “.ini” uzantılı bir dosya ile yapılandırılabilir.
- Bir benzetimin farklı parametreler ile çalıştırılabilmesini desteklemektedir.

- Benzetilecek olan tüm nesnelere statik ya da dinamik olarak oluşturulabilmektedir.
- Ücretsiz bir yazılımdır ve zengin dokümantasyon desteği bulunmaktadır.

### A.3.1. OMNET++ bileşenleri

Omnet++ 4.x (Integrated Development Environment - IDE), Eclipse [99] platformu üzerine inşa edilmiş bir benzetim aracıdır. OMNET++, benzetimlerin gerçekleştirilmesi, benzetimler sırasında meydana gelebilecek hataların ayıklanması ve elde edilen sonuçların analizi için aşağıda maddeler halinde listelenen bileşenlere sahiptir.

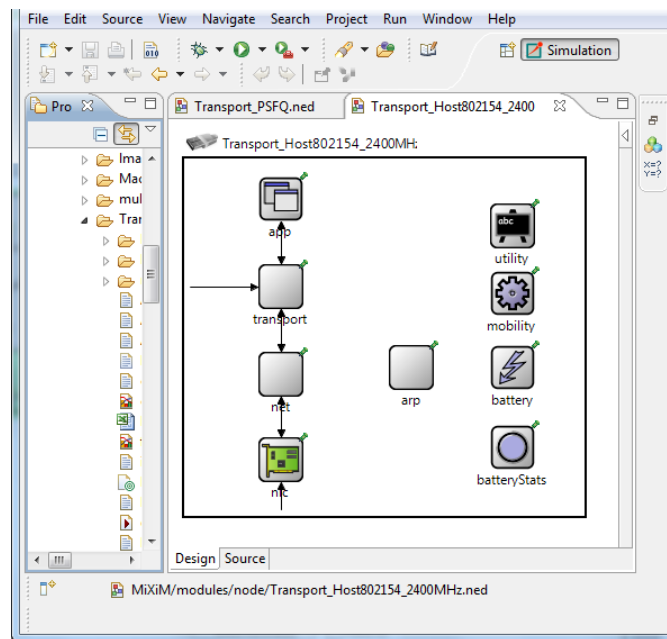
- Gelişmiş benzetim kütüphanesi
- Topoloji tanımlama dili
- Topoloji tanımlamaya yönelik grafiksel ağ editörü
- Benzetimlerin çalıştırılması, hızlı veya yavaş koşturulması, benzetimlerdeki değişkenlerin ve parametrelerin izlenmesine olanak sağlayan kullanıcı arabirimi
- Vektörel ve sayısal olarak saklanan benzetim çıktılarını grafiksel olarak göstermeye yarayan çizim araçları
- Rasgele sayı üreticilerinde kullanılan çekirdeklerin üretilmesine yönelik ve otomatik derleme dosyalarının oluşturulmasını sağlayan araçlar.
- Örnek benzetimler, detaylı kullanıcı el kitabı.

- Hazır benzetim modelleri

Gelişmiş benzetim kütüphanesi: OMNET++, mesajların alınıp gönderilmesi, zamanlama işlemlerinin gerçekleştirilmesi, iletişim kanal yapılarının tanımlanması gibi işlemlerin gerçekleştirilmesini sağlayan çok sayıda kütüphanelere sahiptir.

Topoloji Tanımlama Dili: Derleyicisi ve Grafik Arabirimi: NED (Network Description- Ağ tanımlama), OMNET++ benzetim yazılımının topoloji oluşturma dilidir. Bu sayede ağ topolojisi bağlantıları oluşturulmaktadır. Basit bir komut yapısı olmasına karşın oldukça güçlüdür. GNED ise Şekil A.3’de görüldüğü gibi ağ topolojisini görsel bir şekilde oluşturmayı sağlayan grafik arabirimidir. Görsel tasarımda kodlar otomatik olarak oluşturulmaktadır.

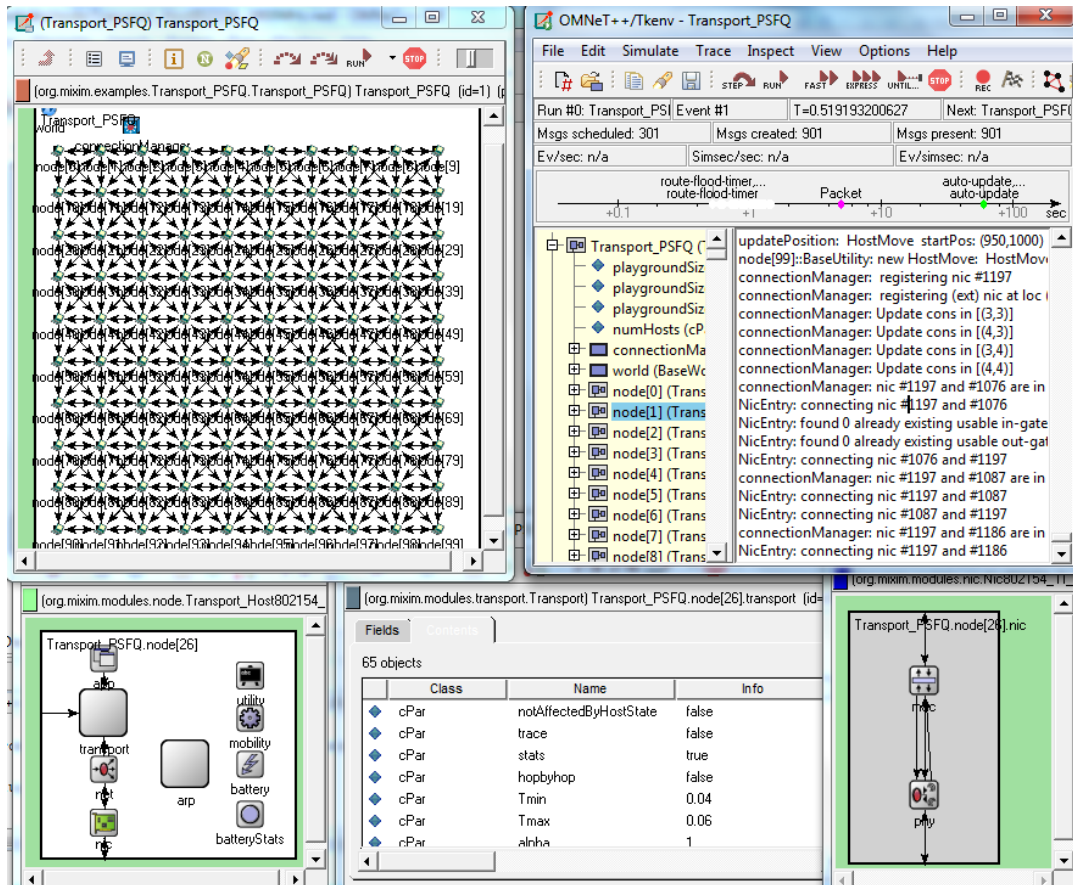
Görsel Kullanıcı Arabirimi (TkEnv): TkEnv, OMNET++ benzetim yazılımının taşınabilir ve görsel kullanıcı arabirimidir. Şekil A.4’de görüldüğü gibi benzetimlerdeki ağ topolojisinin, düğümlerin, paket animasyonlarının, modeller içerisindeki çeşitli değişkenlerin izlenebilmesine olanak sağlayan ve hata ayıklama işlemlerini kolaylaştıran etkileşimli bir arayüzdür.



Şekil A.3. Ağ topolojisi oluşturmak için kullanılan grafik arabirimi (GNED)

Vektörel ve Sayısal Çizim Araçları: OMNET++, benzetim sonuçlarını *.vec* ve *.sca* uzantılı dosyalarda saklar ve bu dosyalar *Plote* ve *Scalar* isimli araçlar sayesinde grafik haline çevrilebilir. Bu sayede elde edilen benzetim sonuçları görsel olarak da incelenebilmektedir.

Çeşitli Araçlar ve Dokümantasyon: OMNET++ farklı sayı üreteçlerine sahiptir ve bu sayı üreteçlerinde kullanılan çekirdeklerin üretilmesini sağlayan çekirdek üretim aracı bulunmaktadır (seedtool). Bunun yanısıra benzetim modellerinin otomatik olarak derlenmesine olanak sağlayan derleme araçları vardır. OMNET++ aynı zamanda kullanıcı el kitabı, çeşitli örnek benzetimler ve forum sayfa desteği gibi güçlü bir dokümantasyon desteği sunmaktadır. Bu da benzetim gerçekleştirmek isteyen kullanıcılar için büyük bir fayda sağlar [26].



Şekil A.4. Grafiksel kullanıcı arabirimi (TkEnv)

Hazır Benzetim Modelleri: Omnet++ açık kaynaklı benzetim yazılımı olmasından dolayı çok sayıda modeller içermektedir. Ücretsiz olması sebebiyle OMNET

kullanan birçok arařtırmacı yeni model ve protokoller geliřtirerek, bu modelleri diđer kullanıcıların da ulaşmasına izin vermektedir.

Omnet++ 4.0/4.1 benzetim yazılımını destekli başlıca hazır modeller řunlardır [97]:

- INET Çerçevesi: Hareketli ve kablosuz benzetimleri destekler. İçerisinde çok sayıda protokol barındırır.
- xMIPv6: INET çerçevesi için hareketli IPv6 benzetim modeli uzantısıdır.
- ReaSE: INET çerçevesinin bir sürümüdür. Hiyerarşik ađ topolojileri, gerçek araçlara dayalı atak trafik ile ilgili benzetim ortamları oluşturabilir ve bunlarla ilgili kullanıcı grafik arabirim sağlar.
- Oversim: Omnet++ benzetim ortamı için geliştirilmiş açık kaynak kodlu model topluluğudur. “Noktadan noktaya” haberleşme ve iletişim tekniklerine yönelik protokolleri desteklemektedir.
- Mixim: Hareketli ve hareketsiz kablosuz ađlar için deđişik modeller içerir. Protokol yığınının daha alt katmanlarına odaklanır ve detaylı radyo yayılım, girişim tahmini, radyo alıcı güç tüketimi ve kablosuz MAC protokolleri içerir [100].
- Castalia; kablosuz algılayıcı ađları, vücut alan ađları ve düşük güçlü gömülü devre ađları için benzetim yapar. Gerçek kablosuz kanal ve radyo modelleri hazır olarak gelmektedir. Model yol kayıp, girişim ve RSSI hesabı, fiziksel süreç modeli ve birkaç popüler MAC protokolünü içerir. Geniş parametrik benzetim çalışmalarını kolaylaştırır [41].
- MF (Mobility Framework): Kablosuz ve hareketli ađların benzetimini destekler. Mixim alt yapısına geçilmesiyle ayrı olarak geliştirilmek yerine Mixim bileşeni olarak geliştirilmektedir. İçerdiği modellere örnek olarak;

802.11 modeli, batarya modeli, BMAC, LMAC, radyo gürültü modeli, CC1100 [101] ve CC2420 [102] için radyo güç tüketim modeli ve IEEE 802.15.4 CSMA modeli verilebilir.

Hu ve arkadaşlarının “Büyük ölçekli algılayıcı ağların modelleme ve analizi için bir benzetim” isimli gerçekleştirmiş oldukları çalışmada; Omnet++ MF, NS-2 ve VWSN [103] benzetim programlarını (sırasıyla 100\*100m, 2000\*2000m ve 5000\*5000m) benzetim alanı ve (sırasıyla 500, 2000 ve 3500) düğüm sayıları ile çalışma zamanı bakımından karşılaştırmıştır. Yapılan karşılaştırma, NS-2 benzetim programının 2000 düğümden fazlasını destekleyemediğini ve Omnet++ MF modelinin NS-2’ye göre işlemleri çok daha kısa sürede tamamladığını göstermektedir. Aynı çalışmada üç benzetim programı 3500 düğüm kullanılarak, bellek kullanımı bakımından kurulum ve algoritma benzetimi için de karşılaştırılmıştır. Bu karşılaştırmalar, NS-2’nin Omnet++ MF modeline göre hem kurulum hem de algoritma benzetimi bakımından daha fazla bellek tükettiğini göstermektedir [103].

Xian ve arkadaşları “Kablosuz algılayıcı ağ simülasyonu için Omnet++ ile diğer benzetim programlarının karşılaştırılması” isimli çalışmalarında; temel MAC ve 802.11 MAC için düğümlerin sırasıyla 10 ve 100 sorgu ürettiği şartlar altında NS-2 ve Omnet++ benzetim araçlarını bellek tüketimi ve çalışma zamanı bakımından karşılaştırmışlardır. Bu karşılaştırmalar, Omnet++ benzetim aracının NS-2 benzetim aracına göre işlemleri daha kısa sürede ve daha az bellek kullanarak tamamladığını göstermektedir [104].

#### **A.4. Mixim Yapısı**

Bu tez çalışmasında kablosuz algılayıcı ağlar için yeni bir taşıma katman protokolünü modelleyebilmek için Omnet++ için hazırlanmış Mixim yapısı kullanılmaktadır.

Mixim, Omnet++’da kablosuz ve hareketli benzetimler için geliştirilmiş bir benzetim modelidir. Omnet++ güçlü bir benzetim yapısı sağlamasına karşın, kablosuz ağlar için desteği çok zayıftır. Mixim Omnet++’da kablosuz ve hareketli benzetimler için geliştirilmiş benzetim çalışma ortamını kapsar ve bu yapıyı genişletir. Özellikle



ayrıntılı kablosuz kanal modelleri, kablosuz bağlantılık özelliği, değişik hareket modelleri, engel modelleri gibi yapılar sağlar. Bunun yanısıra kablosuz ve hareketli ağların benzetiminde kullanılabilir işe yarar görsel öğeler sağlar [105].

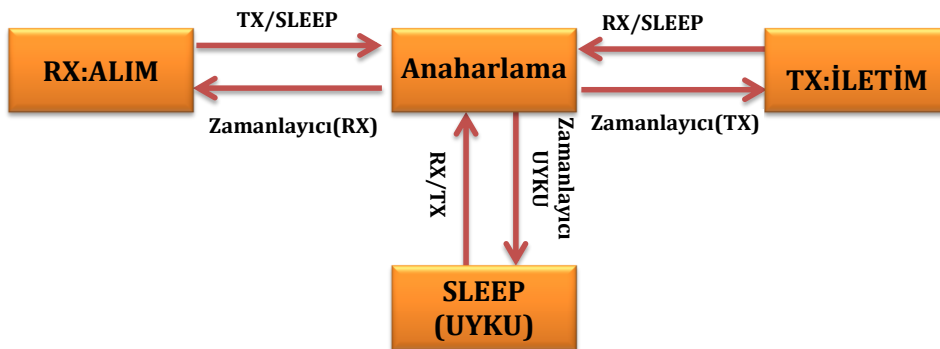
Mixim 3D desteği, radyo sinyal zayıflaması ve hareketliliği etkileyen engel modelleri, farklı frekans ve iletim ortamları, frekans ve uzayda çoklu-kanal desteği, IEEE 802.15.4 dâhil birçok MAC protokolünü içinde barındırır. Mixim düşük bellek tüketimi ve modül yapısının uyumlu olmasından dolayı 1000 düğümden fazla ağların benzetimini de desteklemektedir [105].

#### A.4.1. Mixim yapısı fiziksel katmanı

Ağ yapılarında fiziksel katman MAC katmanına durum bilgisi sağlamaktadır. RX, TX ve SLEEP durumlarını değiştirir, paketleri ortama gönderir, paketleri alır, paketleri dinler, istatistik bilgilerini saklar ve ayarlanabilir düzenlemeler yapar [106].

MAC katmanından istenmesi durumunda MAC katmanına kanalın meşgul, boşta veya RSSI durumunu; şu anki radyo durumunun alma (RX), iletme (TX) veya uyku (SLEEP) radyo durumlarından hangisinde olduğunu ve göndermenin tamamlandığını belirten iletim bitiş bilgilerini sağlar.

Radyo durumları arası anahtarlama süresi Şekil A.5'te görüldüğü gibi radyonun alma, iletme ve uyku durumları arasından hangi radyo durumuna geçiş yapacağına bağlı olarak değişecektir.



Şekil A.5. Radyo durumları arasında anahtarlama ve zamanlama [50]

MAC katmanı paket üzerine zaman üzerindeki bütün boyutlar için bit oranı, TX iletim gücü, frekans / düzlem kanal boyutları ve paketin boyutu gibi bilgileri ekler.

Fiziksel katmandan veri gönderme: İlk olarak MAC katmanı paket ve kontrol bilgisini fiziksel katmana gönderir. Gönderici düğümün fiziksel katmanı alıcı taraftaki fiziksel katmanın ihtiyaç duyduğu bilgileri paketin üzerine ekler ve kanala gönderir. MAC katmanı için mesaj iletiminin sonlanması planlanır.

Alıcı düğümün fiziksel katmanı gelen mesajı değerlendirirken ihtiyaç duyduğu bilgiler:

- Zaman üzerindeki bütün boyutlar için TX iletim gücü ve bit oranı, boyutlu kanal ve öncü süresini tutan sinyal bilgisi.
- Gönderen düğümün hızı, başlangıç yeri, hareket yönü ve ulaşım süresi bilgisi.
- Paket boyutu.

Fiziksel katmandan veri alma: Paketler alıcı düğüme vardığında yayılım gecikmesi, öncü süresi ve yük süresi gibi süreç benzetimleri yapılır. Paket ulaştıktan sonra analog modeller sinyale zayıflama matrisi ekler. Sinyal iletimi sona erdiğinde karar verme bileşeni “sinyal gürültü oranını” değerlendirerek paketi sinyal veya gürültü olarak sınıflandırır. Doğru olarak alınamayan sinyal bit hatası ya da çarpışma işareti ile MAC katmanına geçirilirken, doğru olarak alınan sinyal işaretsiz olarak MAC katmanına geçirilir.

Analog modeller alınan sinyale güç zayıflaması benzetimini yapar. Yol kaybı, gölgeleme ve sönümleme gibi hazır modeller kullanılabileceği gibi yeni bir model geliştirilebilir. Fiziksel katman bir sinyale çoklu analog model uygulayabilir. Analog modeller fiziksel katmandan bağımsız ayarlanabilir.

Karar verici yapı, gelen paketi sinyal veya gürültü olarak sınıflandırır. Paketin sinyal ve girişim gürültüsü temelinde doğru olarak alınıp alınmadığına karar verir. Sinyalin tüm bitlerinin doğruluk bilgisini geri döndürür. Karar verici modeller fiziksel katmandan bağımsız olarak belirlenebilir.

Yapının fiziksel katman ile ilgili erişilebileceği bilgiler: Paket sayısı, alınan sinyal gücü, sinyal gürültü oranı, bit hata oranı ve çarpışmalar.

Yapılandırma “ini” dosyası kullanılarak fiziksel katmana ait aşağıdaki parametre değerleri değiştirilebilir:

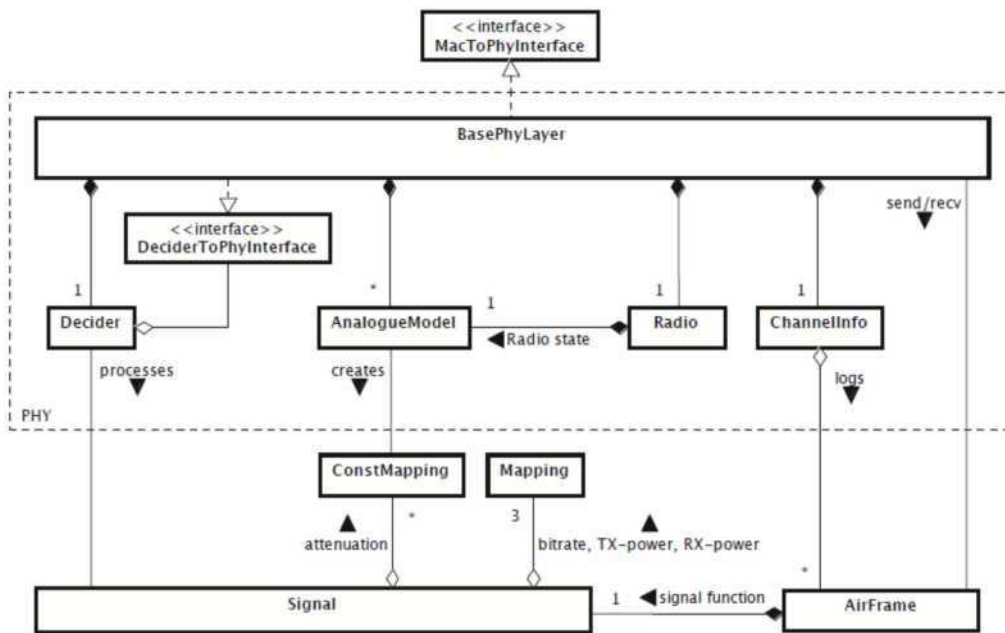
- Seçilecek analog model ve parametre değerleri,
- Termal gürültü, hassasiyet, maksimum TX gücü,
- Yayılım gecikmesinin benzetimi olacak mı?
- Seçilecek karar verici modeli ve parametre değerleri,
- RX, TX ve SLEEP radyo durumları arasında anahtarlama zamanı.

Fiziksel katmanın içerdiği sınıflar ve aralarındaki bağlantılar detaylı olarak Şekil A.6’da görülmektedir.

Temel fiziksel katman arabirimi, Şekil A.6’da gösterildiği gibi MAC katmanına veri kanalı ve kontrol kanalı ile bağlantı sağlar. Fiziksel katman, MAC paketlerini alma - gönderme işlemlerinde veri kanalını kullanılırken; iletim bitti (TX\_OVER) vb. belirli olaylar hakkında MAC katmanını bilgilendirirken kontrol kanalını kullanır [102].

Fiziksel katmanın temel görevleri:

- MAC katmanı kontrol kanalı üzerinden fiziksel katmana kanal algılama isteği gönderir.
- Bir kanal durumu ekleyerek birkaç defa karar vericiye verilir ve sonunda MAC katmanına geri gönderilir.
- Fiziksel katman kanal bilgisine sinyal ekleyebilir ya da kaldırabilir.



Şekil A.6. Fiziksel katman sınıf diyagramı [106]

Şekil A.6'da gösterilen karar verici birimi, fiziksel katmandan gelen paketi alır ve değerlendirir. Karar verici paketin gürültü olup olmadığına karar verebilen zaman noktasını belirler ve fiziksel katmana bu zaman noktası bilgisini geri döndürür. Eğer paket gürültü ise artık onunla ilgilenmez. Paket sinyal olarak sınıflanmışsa fiziksel katmana sinyalin sonunu döndürür. Sonuç olarak fiziksel katman karar sonucu ile birlikte `AirFrame`'i MAC katmanına gönderir [102].

Şekil A.6'da gösterilen temel fiziksel katman ile karar verici arasındaki `DeciderToPhy` arabirimi, karar verici ile birlikte çalışarak mevcut benzetim zamanını

getirir, SNR hesaplamasında kullanmak üzere belirli aralıkta kesişen AirFrame listesini getirir, fiziksel katmana AirFrame'i ve ilgili kontrol mesajını MAC katmanına göndermesini bildirir [102].

AirFrame ve sinyal sınıfları, gönderilecek paket hakkında bilgiyi tutar. AirFrame Omnet++ ilişkili bilgiyi taşıırken; sinyal iletim süreci benzetiminde gerekli olan bilgiyi taşır. Sinyal zaman üzerindeki zayıflama ve iletim gücü için girişleri saklar. Başlık ve yük bit oranı için sabit girişler vardır. Belirli zaman noktalarındaki girişlere erişilebilir. Sinyal ayrıca göndericinin hareket desenini, paket başlık uzunluğu, başlama zamanı ve sinyal uzunluğunu saklar. Paket gönderme sürecini kontrol edebilmek için her AirFrame tekil ID ve AirFrame türüne sahiptir. Ayrıca fiziksel sinyali temsil eden sinyalin bir örneğini tutar [102].

Bir MAC paketinin başarılı bir şekilde alınabilmesi için radyo durumunun TX durumunda ve bir paket gönderilmiyor olması gerekmektedir.

Fiziksel kanala gönderilecek AirFrame oluşturulurken fiziksel katmanın ihtiyaç duyduğu bilgiyi içeren kontrol bilgisi nesnesi MAC paketine eklenir. Fiziksel katman da sinyale yeni bilgiler ekler. AirFrame tamamlandığında gönderilir [102].

Fiziksel katmana bir AirFrame ulaştığında ilgili sinyale analog modeller uygulanır ve AirFrame alınmaya başlar. AirFrame dört aşamada işlenir:

- Paketin alışı başlama noktasına AirFrame planlayarak ve gecikmeye göre sinyal başlama zamanını güncelleyerek yayılım gecikmesi benzetimi yapılır.
- Paket alma işlemi başladığında sinyal işlenmek üzere karar vericiye verilir. Karar verici yeniden sinyali işlemek istediği bir zaman noktası döndürür. Bu zaman noktası sinyalin sonundan önce olmalıdır aksi durumda hata verir.

- AirFrame sinyalinin sonunun zaman noktasını veya negatif zaman noktası döndürene kadar karar verici işleme metodu için keyfi zamanlar planlar. Her iki halde durum AirFrame onun sonuna planlanmadan önce bir artırılır.
- Sonuç olarak alma biter ve AirFrame gerçekte tam olarak alınır [102].

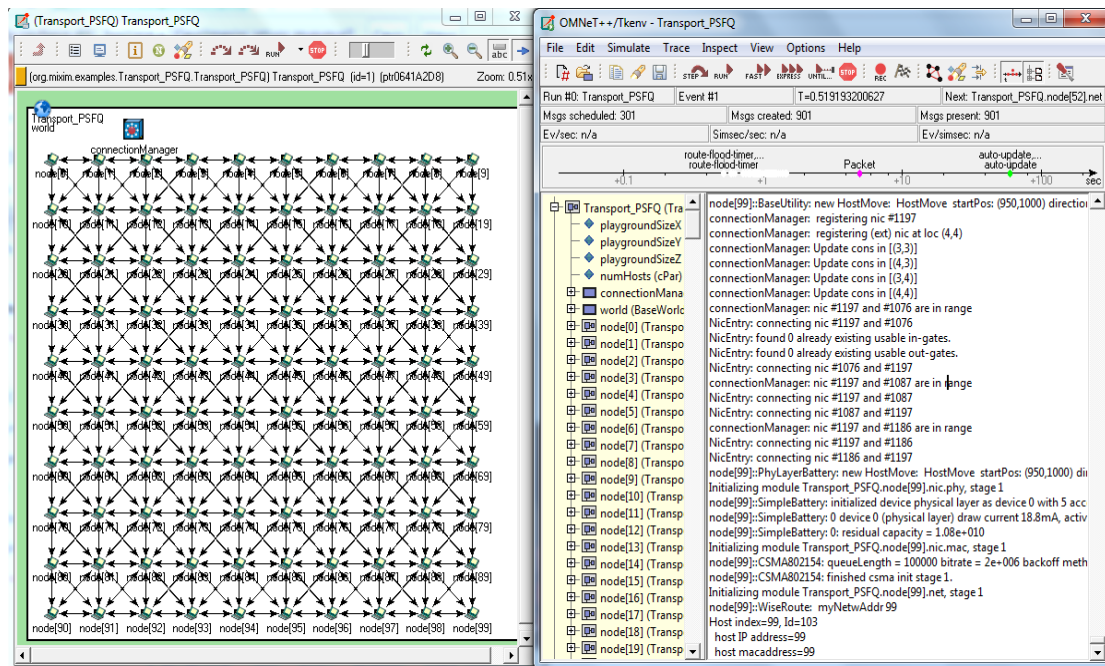
## Ek B. Kablosuz Algılayıcı Ağlara Yönelik OMNET++ Tabanlı Benzetim Modelinin Tasarımı

### B.1. Giriş

Yeni bir benzetim ortamı olan OMNET++'ın hazır model yapıları son yıllarda oldukça hızlı bir şekilde gelişmeye başlamıştır. Günümüzde kablosuz algılayıcı ağları için tasarlanmış OMNET++ tabanlı çeşitli benzetim modelleri bulunmaktadır. Bu sayede hazır modeller kullanılarak daha gerçekçi benzetimler gerçekleştirilebilmektedir.

### B.2. OMNET++ tabanlı benzetim modelinin genel özellikleri

OMNET++ tabanlı kablosuz algılayıcı ağ benzetim modeli, Omnet++ altında çalışan Mixim 2.1 üzerinde geliştirilmiştir. Tasarlanan model, Şekil B.1'de görüldüğü gibi görsel kullanıcı arayüzüne sahiptir. Tasarlanan benzetim modelinde normal (node), ve çıkış (sink) olmak üzere iki farklı düğüm türü bulunmaktadır. Normal düğümler algıladıkları bilgileri komşuları üzerinden çıkış düğümüne iletmekle görevlidir. Çıkış düğümü (Sink), normal düğümlerden gelen bilgilerin toplanması ve işlenmesinden sorumludur.



Şekil B.1. OMNET++ tabanlı kablosuz algılayıcı ağ benzetim modelinin ekran görüntüsü

Benzetim modelindeki birçok parametre omnetpp.ini yapılandırma dosyası ile tanımlanır. Bu sayede herhangi bir kod yazmaya gerek kalmadan farklı benzetimler gerçekleştirmek mümkündür. Şekil B.2’de görüldüğü gibi benzetim alanının boyutu, benzetimdeki toplam düğüm sayıları, düğümlerin yerleşim ayarları, düğümlerin radyo alıcı/verici türü, radyo iletim mesafesi, MAC protokol türü, yönlendirme katmanı gibi birçok parametre bu dosya ile yapılandırılabilir. Ayrıca sık şekilde çalıştırılan benzetimlerin parametreleri [config] isimli kısımlarda tanımlanarak kullanıcının omnetpp.ini dosyasını yapılandırmadan benzetimleri değiştirebilmesi sağlar. Şekil B.3’de omnetpp.ini dosyasında tanımlanmış olan farklı ayarlarının, benzetim başlangıcında seçilmesi görülmektedir. Başlangıçta kullanıcı hangi sekme seçerse benzetim o kısımda olan parametrelere göre çalışmaktadır.

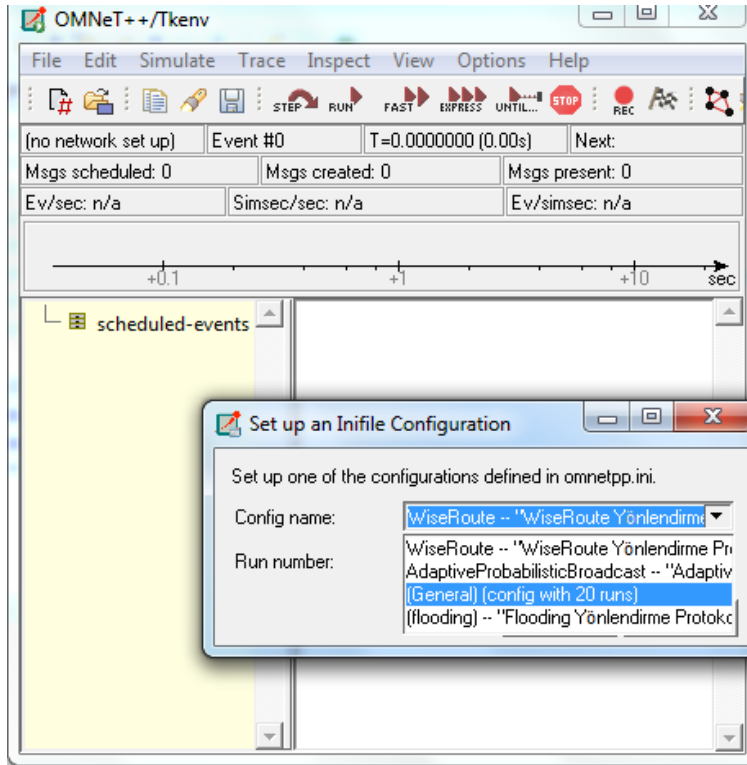
```

4  sim-time-limit = ${simDuration=20} min
5  ned-path = ../../base;../../modules;../../examples;
6
7  num-rngs = 11
8  repeat = 5
9  [ALORT PROTOKOL]
10 #Alan Bilgileri
11 **.playgroundSizeX = 1050 m
12 **.playgroundSizeY = 1050 m
13 **.playgroundSizeZ = 0 m
14
15 #Düğüm Yerleşimi
16 **.node[*].mobility.x = -1
17 **.node[*].mobility.y = -1
18 **.node[*].mobility.z = 0
19
20 #Host Sayısı
21 **.numHosts = ${numHosts=100,101}
22
23 #Fiziksel Katman
24 **.connectionManager.sendDirect = false
25 **.connectionManager.pMax = 1 mW
26 **.connectionManager.sat = -100dBm #-100
27 **.connectionManager.alpha = 2.0
28 **.connectionManager.carrierFrequency = 2.4E+9Hz
29 **.node[*].nic.phy.usePropagationDelay = false #false
30 **.node[*].nic.phy.analogueModels = xmldoc("config.xml")
31 **.node[*].nic.phy.sensitivity = -999999 dBm #-110dBm
32 **.node[*].nic.phy.maxTXPower = 1.1 mW
33 **.node[*].nic.phy.initialRadioState = 0
34 **.node[*].nic.phy.useThermalNoise = false
35
36 # Yönlendirme Katmanı
37 **.node[*].netType = "WiseRoute"
38 **.node[*].net.stats = true
39 **.node[*].net.routeFloodsInterval = 2400 s
40 **.node[*].net.sinkAddress = 99
41 **.node[*].net.headerLength = 3 bit # 3 byte 24 bit
42
43 # MAC Katmanı
44 **.node[*].nic.mac.useMACacks = false
45
46 # Uygulama Katmanı
47 **.Packet_Send_Delay=5
48 **.app.debug = false
49 **.app.headerLength = 256bit #256bit
50 **.app.burstSize = 1
51
52 # Ulaşım Katmanı
53 **.transport.Pac_num = 50 #Packet number that will send
54 **.transport.Tmax = 0.06#Tmax value
55 **.transport.Tmin = 0.04 #Tmin value
56 **.transport.Tr = 0.02
57 **.transport.e2eTime=0.005
58 **.transport.Buffer_length= 1000
59 **.transport.targetNode = 99
60 **.transport.senderNode=67
61 **.transport.alpha=1
62 **.transport.gama=1.1
63 **.transport.proActive_Number_Set=500
64 **.transport.headerLength = 2
65 **.transport.fetch_Counter=3
66 **.transport.hopbyhop=false
67 **.transport.thresholdTime=1000 #3.7 3.1 3.3 3.8

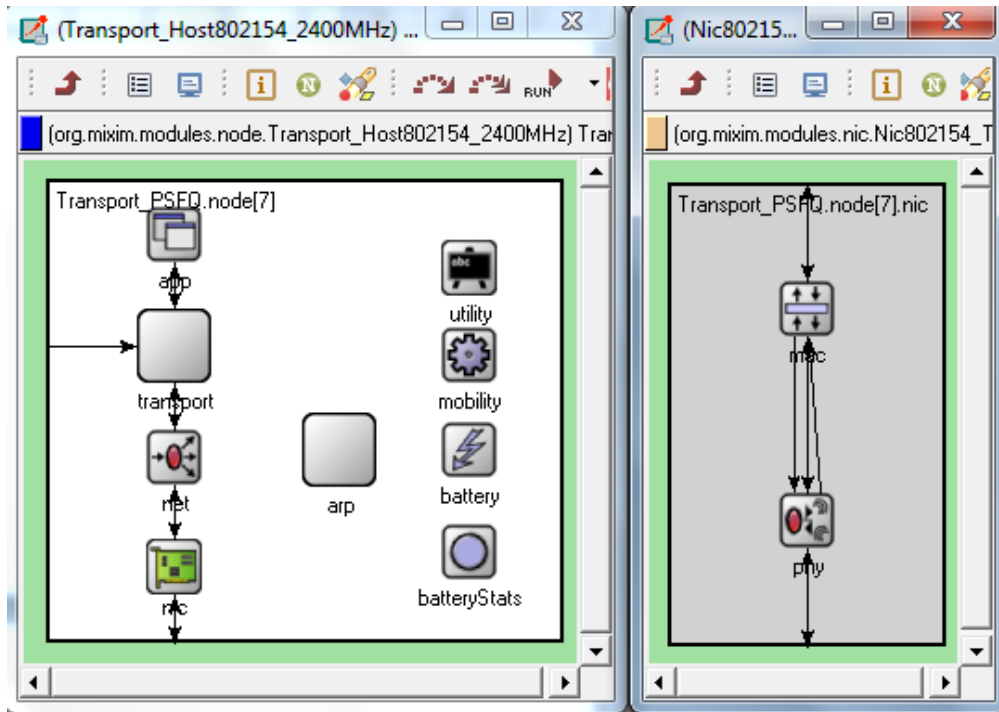
```

Şekil B.2. OMNET++ tabanlı benzetim modelinin yapılandırma dosyası (omnetpp.ini)





Şekil B.3. Farklı benzetim türlerinin benzetim başlangıcında arayüzden seçilmesi



Şekil B.4. Katmanlı düğüm mimarisi

### B.3. Dügüm yapısı

Geliştirilen OMNET++ tabanlı benzetim modelinde kablosuz algılayıcı düğümleri, Şekil B.4’de görüldüğü gibi katmanlı ağ mimarisini destekleyecek şekilde tasarlanmıştır. Böylelikle istenildiğinde herhangi bir katmandaki protokolün değiştirilmesi diğer katmanlar açısından bir sorun teşkil etmeyecektir. Dügüm içerisindeki “app” uygulama katmanını, “transport” taşıma katmanını, “net” yönlendirme katmanını, “nic” MAC ve fiziksel katmanı, “battery” ve “batteryStats” batarya ile ilgili bilgileri, “arp” adres dönüşümlerini, “utility” genel araçları, “mobility” ise düğümlerin hareketliliğini düzenleyen modülleri temsil etmektedir.

#### B.3.1. Fiziksel katman

Paket gönderimi, alımı gibi işlemlerden sorumlu olan fiziksel katman içinde CC2420 radyo alıcı/verici devresi seçilmiştir. Şekil B.5’te görüldüğü gibi, yapılandırma dosyasından (omnetpp.ini) iletim hızı, iletim frekansı, iletim gücü gibi özellikler otomatik olarak değiştirilebilmektedir.

```

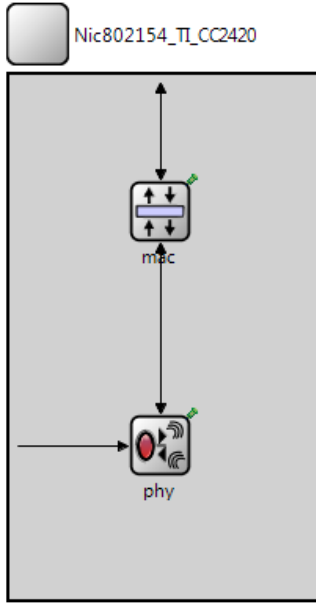
**.node[*].nic.phy.usePropagationDelay = false #false
**.node[*].nic.phy.analogueModels = xmldoc("config.xml")
**.node[*].nic.phy.sensitivity = -999999 dBm #-110dBm
**.node[*].nic.phy.maxTXPower = 1.1 mW
**.node[*].nic.phy.initialRadioState = 0
**.node[*].nic.phy.useThermalNoise = false

```

Şekil B.5. Fiziksel katman ayarları

#### B.3.2. MAC katmanı

MAC katmanı, kanal erişimi, çerçeve filtreleme ve radyo modlarının ayarlanmasından sorumludur. Tasarlanan benzetim programında Şekil B.6’da görüldüğü gibi, kablosuz algılayıcı ağlara yönelik olarak tasarlanmış ve en yaygın MAC protokolü olan CSMA802154 protokolünün modeli bulunmaktadır. Şekil B.7’de görüldüğü gibi, omnetpp.ini yapılandırma dosyası kullanılarak MAC katmanı ile ilgili ayarlar kolaylıkla değiştirilebilmektedir. Kullanılan MAC protokolü Mixim yapısı içinde hazır olup, gerçeği ile birebir olarak modellenmiştir.



Şekil B.6. MAC protokolü yapısı

```

**node[*].nic.mac.txPower = ${txPower=0.1,1} mW
**node[*].nic.mac.useMACAcks = false

```

Şekil B.7. MAC protokolü yapılandırma dosyası ayarları

### B.3.3. Yönlendirme katmanı

Yönlendirme katmanı, düğümlerin çok atlamalı yollar üzerinden çıkış düğümünü bulmasını sağlayan katmandır. Düğümler en kısa yol algoritmasını kullanarak çıkış düğümünü bulmaktadır. Bu benzetim ortamında Omnet++ geliştiricileri tarafından Mixim yapısına katılan WiseRoute protokolü kullanılmaktadır. Şekil B.8’de yönlendirme katmanının yapılandırma dosyası ile yapılan ayarları görülmektedir. İstenilirse yapılandırma dosyası içerisindeki “netwType” özelliği ile farklı yönlendirme protokolleri kolaylıkla modellenerek sisteme katılabilir.

```

**node[*].netwType = "WiseRoute"
**node[*].net.stats = true
**node[*].net.routeFloodsInterval = 2400 s
**node[*].net.sinkAddress = 99
**node[*].net.headerLength = 24 bit

```

Şekil B.8. Yönlendirme katmanı yapılandırma dosyası ayarları

### B.3.4. Taşıma katmanı

Taşıma katmanı kaynaktan hedefe gönderilen paketlerin güvenli bir şekilde hedefe ulaştırılmasından sorumludur. Bunu gerçekleştirmek için değişik algoritmalar geliştirilmiştir. Bu çalışmada HİKMET adı verilen yenilikçi bir taşıma katman algoritması geliştirilerek, bu benzetim içerisinde kullanılmıştır. Şekil B.9'da taşıma katmanının yapılandırma dosyası ile yapılan ayarları görülmektedir.

```

**.transport.Pac_num = 50
**.transport.Tmax = 0.06
**.transport.Tmin = 0.04
**.transport.Tr = 0.02
**.transport.Buffer_length= 1000
**.transport.targetNode = 99
**.transport.senderNode=67
**.transport.alpha=1
**.transport.gama=1.1
**.transport.proActive_Number_Set=500
**.transport.headerLength = 2
**.transport.fetch_Counter=3
**.transport.hopbyhop=false
**.transport.thresholdTime=1000 |
**.transport.thresholdNumber=2
**.transport.thresholdE2etime=1000
**.transport.missPkt = 5

```

Şekil B.9. Taşıma katmanı yapılandırma dosyası ayarları

### B.3.4. Uygulama katmanı

Düğümün hangi sıklıkla veya hangi aralıklarla algılama işlemlerini yürüteceğini ve buna bağlı olarak ne zaman paket gönderimini gerçekleştireceğini belirleyen katmandır. OMNET++ tabanlı benzetim modelinde düğümlerin algılama işlemleri paket üretici ile sağlanmaktadır. Farklı paket dağılım üreticileri yardımıyla trafik üretimi gerçekleştirilebilmektedir.

### B.4. İletişim kanalı

İletişim kanalı düğümlerin kablosuz ortam üzerinden haberleşmesini sağlayan ortamdır. Kablosuz ağlarda düğümlerin haberleşmesini engelleyen en önemli sorunlardan birisi de kötü iletişim ortamıdır. Bunun nedeni maliyeti ve güç

tüketimini azaltmak için basit radyoların tercih edilmesidir. Bunu engellemek için kanal değerleri uygun şekilde ayarlanmalıdır.

Kullanılan benzetim yazılımında istenilen özelliklerdeki kanal parametre ve değerleri Şekil B.10’da görüldüğü gibi ayarlanmaktadır. Burada taşıyıcı frekans değeri “carrierFrequency”, alfa, “alpha”, sinyal zayıflaması, “sat” ve maksimum güç “pMax” parametreleri ile ayarlanır.

```

**.connectionManager.sendDirect = false
**.connectionManager.pMax = 1 mW
**.connectionManager.sat = -100dBm #-100
**.connectionManager.alpha = 2.0
**.connectionManager.carrierFrequency = 2.4E+9Hz

```

Şekil B.10. Kanal parametre ve değerleri

### B.5. Pil tüketim değerleri

Benzetim süresince düğümler gerçekleştirdikleri her bir işlemde enerji harcarlar. “Battery” düğümlerin pili ile ilgili parametreleri, “batteryStats” ise pil ile ilgili istatistiksel bilgileri tutar. Bu sayede her düğümün tükettiği toplam enerji miktarı, uyku, alım, gönderim ve dinlenme gibi radyo durumlarında ayrı ayrı ne kadar enerji tükettikleri bilgisi de alınabilmektedir. Şekil B.11’de “battery” modülüne ait parametreler görülmektedir.

```

battery: SimpleBattery {
    @display("p=292,186;i=block/control");
    nominal = default(1000000 mAh);
    capacity = default(1000000 mAh);
    voltage = default(3 V);
    resolution = default(60 s);
    publishDelta = default(1);
    publishTime = default(60 s);
    numDevices = default(1);
}

```

Şekil B.11. Pil parametre ve değerleri

### B.6. Hata durumlarının benzetimi

Kablosuz algılayıcı ağların yapısından dolayı, donanımsal ve yazılımsal hatalarla çok sayıda karşılaşılabilir. Bundan dolayı, benzetim ortamında hata durumlarının

gerçekleştirilebilmesi gereklidir. Fiziksel katmandan alınan sinyallerin doğru bir şekilde ulaşp ulaşmadığının tespitini Şekil B.12’de gösterilen Config.xml dosyasındaki parametrelerle belirlenmektedir. “berLowerBound” parametresi ile istenilen hata ortamı oluşturulabilir. Paketler alındığında burada belirtilen hata oranına göre rasgele bir değer oluşturularak, paketin bozulup bozulmadığına karar verilir. Bu şekilde benzetim ortamı gerçek hayatta elde edilen sonuçlara yakın değerler elde edebilecektir.

```

<root>
  <AnalogueModels>
    <AnalogueModel type="BreakpointPathlossModel">
      <!-- IEEE 802.15.4 path loss channel model -->
      <parameter name="alpha1" type="double" value="2"/>
      <parameter name="L01" type="double" value="40.2"/>
      <parameter name="breakpointDistance" type="double" value="8.0"/>
      <parameter name="alpha2" type="double" value="3.3"/>
      <parameter name="L02" type="double" value="58.8"/>
      <parameter name="carrierFrequency" type="double" value="2.4E+9"/>
    </AnalogueModel>
    <!-- we add a log-normal shadowing effect on top of the IEEE 802.15.4 path loss -->
    <AnalogueModel type="LogNormalShadowing">
      <parameter name="mean" type="double" value="0.0"/>
      <parameter name="stdDev" type="double" value="6"/>
      <parameter name="interval" type="double" value="1"/>
    </AnalogueModel>
  </AnalogueModels>
  <Decider type="Decider802154Narrow">
    <!--Length of Start Frame Delimiter (used to compute probability of successful
    synchronization)-->
    <parameter name="sfdLength" type="long" value="8"/>

    <!--minimum possible bit error rate (BER floor)-->
    <parameter name="berLowerBound" type="double" value="1e-8"/>

    <!--modulation type-->
    <parameter name="modulation" type="string" value="msk"/>
  </Decider>
</root>

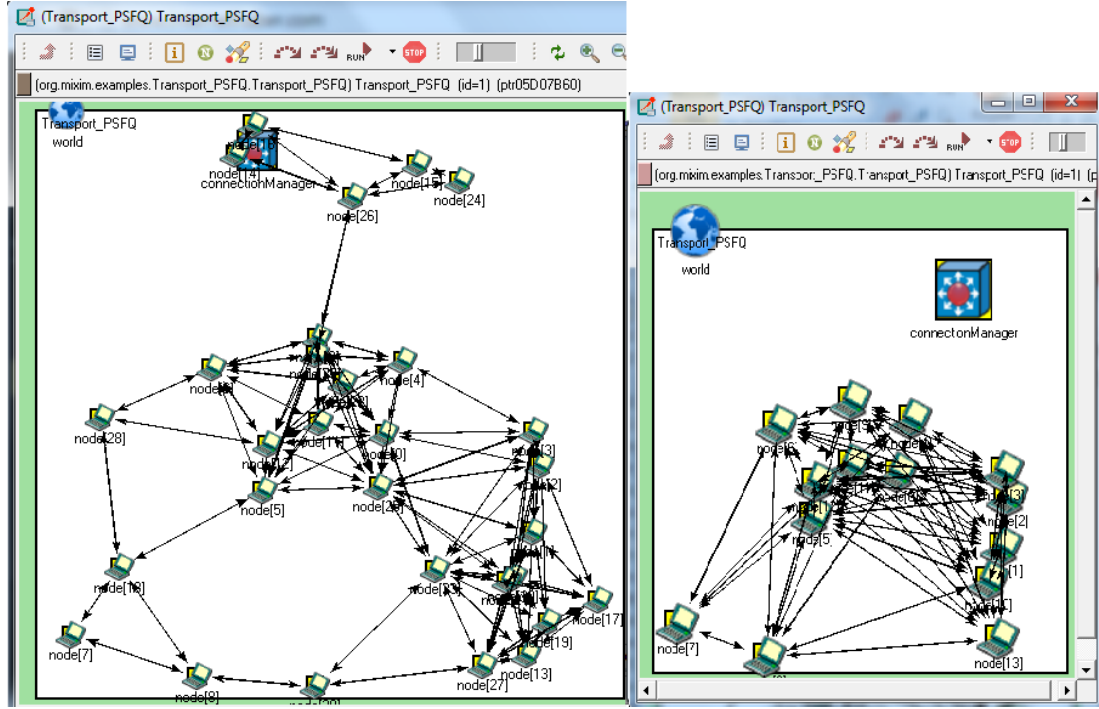
```

Şekil B.12. Analog model ile hata olup olmadığına karar verilmesi

## B.7. Düğümlerin konumlandırılması

OMNET++ tabanlı benzetim yazılımı, *omnetpp.ini* dosyasında belirtilen düğüm sayılarına ve benzetim alanının büyüklüğüne göre düğümlerin rasgele veya sabit şekilde konumlandırabilmesine olanak sağlamaktadır. *PlaygroundSizeX* ve *playgroundSizeY* parametreleriyle benzetim alanının büyüklüğü belirlenmektedir. *NumHosts* parametresiyle belirlenen düğüm sayısı, *mobility.x = -1*, *mobility.y = -1* ve *mobility.z = -1* parametreleri kullanılarak rasgele yerleştirilebildiği gibi, istenilirse sabit bir şekilde de yerleştirilebilir. Bu parametreler sayesinde istenilen ölçekte ve

yoğunlukta benzetim gerçekleştirmek mümkündür. Şekil B.13'te farklı büyüklükte ve farklı düğüm sayılarındaki topoloji örnekleri görülmektedir.



a) 600x600m<sup>2</sup> alan ve 50 düğüm

b) 300x300 m<sup>2</sup> alan ve 15 düğüm

Şekil B.13. Düğüm topoloji örnekleri

### B.8. Yeni bir benzetimin gerçekleştirilmesi

Tasarlanan OMNET++ tabanlı benzetim modeli, kablosuz algılayıcı ağların benzetimine yönelik olarak geliştirilmiş modüller ve kolay konfigüre edilebilir bir benzetim ortamıdır. Yapılandırma dosyasındaki parametreler değiştirilerek farklı amaçlara yönelik birçok uygulamanın benzetimi kolaylıkla gerçekleştirilebilir. Ancak kullanıcının benzetim modelindeki bir özelliği değiştirmesi, yeni özellikler eklemesi veya yeni bir protokol tasarlaması gerekebilir.

Yeni bir taşıma katman protokolü tasarlanacaksa, taşıma katman protokolünün davranışlarının tanımlandığı yeni bir C++ dosyası (.cc), OMNET++ proje dosyasına

eklenmeli ve proje yeniden derlenmelidir ayrıca omnetpp.ini dosyasındaki ayarlar istenilen şekilde deęiştirilmelidir.

Eđer aęa farklı özelliklerde yeni bir düęüm türünün eklenmesi gerekirse düęümün kapı tanımlamaları ve varsa parametreleri GNED editörü kullanılarak tanımlanmalıdır. Ayrıca eklenen düęümün davranışını belirleyen C++ kodları OMNET++ proje dosyasına eklenmeli ve proje yeniden derlenmelidir.



## ÖZGEÇMİŞ

Ayhan KİRAZ, 1980 Pamukova/SAKARYA doğumludur. İlk ve orta öğretimini Pamukova'da tamamladı. Lise öğrenimini ise Adapazarı Fatih Endüstri Meslek Lise'sinde tamamladı. 1998 yılında girdiği Marmara Üniversitesi Teknik Eğitim Fakültesi Elektronik ve Bilgisayar Eğitimi Bölümü Bilgisayar ve Kontrol Öğretmenliği Programından 2002 yılında mezun oldu. Aynı yıl Adapazarı Fatih Endüstri Meslek Lisesinde Bilgisayar Öğretmeni olarak göreve başladı. 2003 yılında başladığı yüksek lisans eğitimini "Java Programlama Dilinin Web Üzerinden Sunumu" isimli yüksek lisans tez çalışması ile 2006 yılında bitirerek bilim uzmanı unvanını aldı. Adapazarı Fatih Endüstri Meslek Lisesi'nde görevine devam etmektedir. Evli ve üç çocuk babasıdır.