



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

**KABLOSUZ MULTİMEDYA ALGILAYICI AĞLAR İÇİN
GÖRÜNTÜ SIKIŞTIRMAYA DAYALI YENİ BİR ENERJİ
DUYARLI UYGULAMA KATMANI ALGORİTMASI
GELİŞTİRME**

ARAFAT ŞENTÜRK

**DOKTORA TEZİ
ELEKTRİK-ELEKTRONİK VE BİLGİSAYAR MÜHENDİSLİĞİ
ANABİLİM DALI**

**DANIŞMAN
DOÇ. DR. RESUL KARA**

DÜZCE, 2017

T.C.
DÜZCE ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

KABLOSUZ MULTİMEDYA ALGILAYICI AĞLAR İÇİN
GÖRÜNTÜ SIKIŞTIRMAYA DAYALI YENİ BİR ENERJİ
DUYARLI UYGULAMA KATMANI ALGORİTMASI
GELİŞTİRME

Arafat ŞENTÜRK tarafından hazırlanan tez çalışması aşağıdaki jüri tarafından Düzce Üniversitesi Fen Bilimleri Enstitüsü Elektrik-Elektronik ve Bilgisayar Mühendisliği Anabilim Dalı'nda **DOKTORA TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Doç. Dr. Resul KARA

Düzce Üniversitesi

Jüri Üyeleri

Doç. Dr. İbrahim ÖZÇELİK
Sakarya Üniversitesi

Doç. Dr. Resul KARA
Düzce Üniversitesi

Doç. Dr. Pakize ERDOĞMUŞ
Düzce Üniversitesi

Yrd. Doç. Dr. Murat İSKEFİYELİ
Sakarya Üniversitesi

Yrd. Doç. Dr. Mehmet ŞİMŞEK
Düzce Üniversitesi

Tez Savunma Tarihi: 14/02/2017

BEYAN

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

14 Şubat 2017

Arafat ŞENTÜRK

TEŞEKKÜR

Doktora öğrenimimde ve bu tezin hazırlanmasında gösterdiği her türlü destek ve yardımdan dolayı değerli danışmanım Doç. Dr. Resul KARA'ya en içten dileklerimle teşekkür ederim.

Tezin her aşmasında verdikleri fikir ile çalışmalarına ışık tutan değerli hocalarım Doç. Dr. İbrahim ÖZÇELİK, Doç. Dr. Pakize ERDOĞMUŞ ve Yrd. Doç. Dr. Mehmet ŞİMŞEK'e teşekkürü bir borç bilirim.

Eğitimim boyunca yardımlarını ve desteklerini esirgemeyen sevgili eşime sonsuz teşekkürlerimi sunarım. Ayrıca, çalışmalarım sırasında sabırla bana destek olan ailem ve kızım Beyza'ya en kalbi duygularıyla teşekkürlerimi sunarım.

Bu tez çalışması, Düzce Üniversitesi Bilimsel Araştırma Projeleri Koordinatörlüğü tarafından 2015.06.01.308 numaralı Bilimsel Araştırma Projesiyle desteklenmiştir.

14 Şubat 2017

Arafat ŞENTÜRK

İÇİNDEKİLER

İÇİNDEKİLER.....	V
ŞEKİL LİSTESİ.....	IX
ÇİZELGE LİSTESİ.....	XI
KISALTMALAR.....	XIV
ÖZET.....	XV
ABSTRACT.....	XVII
EXTENDED ABSTRACT.....	1
1. GİRİŞ.....	5
1.1 KABLOSUZ MULTİMEDYA ALGILAYICI AĞLAR VE KULLANIM ALANLARI.....	6
1.2 KMAA’LAR İÇİN ENERJİ VERİMLİLİĞİ İLE İLGİLİ YAPILAN ÇALIŞMALAR.....	7
1.2.1 Fiziksel Katman.....	8
1.2.2 MAC (Medium Access Control- Ortam Erişim Kontrolü) Katmanı.....	8
1.2.3 Ağ Katmanı.....	9
1.2.4 Taşıma Katmanı.....	10
1.2.5 Uygulama Katmanı.....	11
1.3 ÇALIŞMANIN AMACI VE ÖNERİLEN ÇÖZÜM YÖNTEMLERİ.....	14
1.4 TEZ ORGANİZASYONU.....	16
2. KABLOSUZ MULTİMEDYA ALGILAYICI AĞLARDA GÖRÜNTÜ SIKIŞTIRMA TEKNİKLERİ.....	18
2.1 KAYIPSIZ GÖRÜNTÜ SIKIŞTIRMA TEKNİKLERİ.....	19
2.2 KAYIPLI GÖRÜNTÜ SIKIŞTIRMA TEKNİKLERİ.....	19
2.2.1 Ayrık Kosinüs Dönüşüm Tabanlı (Discrete Cosine Transform – DCT) Sıkıştırma Teknikleri.....	20
2.2.2 Ayrık Dalgacık Dönüşümü (Discrete Wavelet Transform – DWT) Tabanlı Sıkıştırma Teknikleri.....	21

2.2.3 Gömülü Sıfır Ağaç Dalgacık Tabanlı (Embedded Zerotree Wavelet – EZW) Görüntü Sıkıştırma Teknikleri	22
2.2.4 Hiyerarşik Ağaçlarda Küme Bölümleme (Set Partitioning in Hierarchical Trees – SPIHT).....	23
2.2.5 Düşük Enerjili Görüntü Sıkıştırma Algoritması (Low Energy Image Compression Algorithm - LEICA).....	24
2.2.6 Optimize Kesme ile Gömülü Blok Kodlama (Embedded Block Coding with Optimized Truncation - EBCOT).....	25
2.3 KMAA’LARDA KULLANILAN GÖRÜNTÜ SIKIŞTIRMA TEKNİKLERİNİN DEĞERLENDİRİLMESİ	26
2.4 BELİRLENEN GÖRÜNTÜ SIKIŞTIRMA ALGORİTMALARININ PERFORMANS ANALİZİ	30
2.4.1 Ayrık Kosinüs Dönüşümü (Discrete Cosine Transform - DCT)’nün Performansı	31
2.4.2 Hiyerarşik Ağaçlarda Küme Bölümleme (Set Partitioning in Hierarchical Trees - SPIHT) ’nin Performansı	32
2.4.3 Düşük Enerjili Görüntü Sıkıştırma Algoritması (Low Energy Image Compression Algorithm - LEICA) ’nın Performansı.....	33
2.5 BELİRLENEN GÖRÜNTÜ SIKIŞTIRMA ALGORİTMALARININ KARŞILAŞTIRILMASI	34
2.6 SONUÇ	34
3. KABLOSUZ MULTİMEDYA ALGILAYICI AĞLAR İÇİN GÖRÜNTÜ SIKIŞTIRMA ALGORİTMALARINA DAYALI YENİ BİR ENERJİ DUYARLI UYGULAMA KATMANI ALGORİTMASI GELİŞTİRME	35
3.1 GİRİŞ	35
3.2 ALGILAYICI DÜĞÜMLERİN SAHİP OLMASI GEREKEN ÖZELLİKLER.....	36
3.3 GSD-EDUKA’NIN YAPISI	37
3.4 GSD-EDUKA’NIN ADIMLARI.....	40
3.5 GSD-EDUKA’NIN PARAMETRELERİ	42
3.6 SIKIŞTIRMA ALGORİTMASINI BELİRLEME	44
3.6.1 Düğümler Arası Mesafeye Göre Kullanılacak Algoritmayı Belirleme ...	45

3.6.2 Ağda Kullanılan Düğüm Sayısına Göre Kullanılacak Algoritmayı Belirleme	48
3.6.3 Ağ Üzerindeki Veri Gönderme Sıklığına Göre Kullanılacak Algoritmayı Belirleme	51
3.6.4 Görüntü Sıkıştırma Algoritması Seçimi	53
3.6.5 PSNR Değer Kaybının Korunması	56
3.7 SONUÇ	60
4. KMAA’LARDA ENERJİ TÜKETİMİ.....	62
4.1 ALGILAYICI DÜĞÜMLERİN ENERJİ TÜKETİMİ	62
4.2 GSD-EDUKA’YI KULLANAN ALGILAYICI DÜĞÜMÜN HARCADIĞI ENERJİ.....	63
4.3 SONUÇ	67
5. GSD-EDUKA’NIN PERFORMANS DEĞERLENDİRMESİ.....	68
5.1 KMAA’LARDA PERFORMANS KRİTERLERİ	68
5.2 GSD-EDUKA’NIN PERFORMANS DEĞERLENDİRME İŞLEM ADIMLARI	69
5.2.1 Performans Değerlendirme Ortamı	69
5.2.2 GSD-EDUKA’nın Parametreleri.....	70
5.2.3 Karar Ağacı Yöntemi’nin Kullanılması	70
5.2.4 Bulanık Mantık Yöntemi’nin Kullanılması.....	70
5.3 PERFORMANS DEĞERLENDİRME	71
5.3.1 Benzetim Parametreleri	72
5.3.2 GSD-EDUKA’nın PSNR Değer Kaybına Göre Diğer Sıkıştırma Algoritmalarıyla Karşılaştırılması	73
5.3.3 Enerji Oranı	76
5.3.3.1 Sabit Ağlar için Enerji Oranı Sonuçları	76
5.3.3.2 Hareketli Ağlar için Enerji Oranı Sonuçları	86
5.3.4 Paket Ulaştırma Oranı	96
5.3.5 Harcanan Enerji Değerleri	100
5.3.6 Uçtan Uca Gecikme	101
5.3.7 Yöntemlerin Harcanan Enerjiye göre Karşılaştırılması	103
6. SONUÇLAR VE ÖNERİLER	105
6.1 SONUÇ	105

6.2 ÇALIŞMANIN GETİRDİĞİ KATKILAR	107
6.3 TARTIŞMA VE ÖNERİLER	108
KAYNAKLAR.....	110
EKLER.....	124
A. KARAR AĞACI YÖNTEMİ	124
B. NETWORK SIMULATOR – 2 (NS-2) KODLARI.....	137
ÖZGEÇMİŞ	142



ŞEKİL LİSTESİ

Sayfa No

Şekil 2.1. KMAA’larda kullanılan görüntü sıkıştırma teknikleri [60].	18
Şekil 2.2. Kayıplı JPEG sıkıştırma şeması [74].	21
Şekil 2.3. 2B DWT’nin ayrıştırılmasının iki seviyesi [73].	22
Şekil 2.4. SPIHT kodlama adımları [76].	23
Şekil 2.5. JPEG200 blok diyagramı [76].	26
Şekil 2.6. DCT algoritması için (a) orijinal görüntü (b) sıkıştırılmış görüntü.	31
Şekil 2.7. SPIHT algoritması için (a) orijinal görüntü (b) sıkıştırılmış görüntü.	32
Şekil 2.8. LEICA algoritması için (a) orijinal görüntü (b) sıkıştırılmış görüntü.	33
Şekil 3.1. GSD-EDUKA’nın işlem adımları.	37
Şekil 3.2. GSD-EDUKA’nın OSI katmanlarında gerçekleşen işlemleri.	38
Şekil 3.3. GSD-EDUKA’nın yapısı.	39
Şekil 3.4. GSD-EDUKA’nın akış diyagramı.	42
Şekil 3.5. Bulanık Mantık giriş ve çıkış değişkenlerinin blok görünümü.	45
Şekil 3.6. “hello” paketinin çerçeve yapısı.	45
Şekil 3.7. Değiştirilen “hello” paketinin çerçeve yapısı.	45
Şekil 3.8. Düğümler arasındaki mesafeye göre “enerji oran” değişim grafiği.	47
Şekil 3.9. “Mesafe” giriş değişkeni üyelik fonksiyonlarının grafik görünümü.	48
Şekil 3.10. Kullanılan toplam düğüm sayısına göre “enerji oran” değişimi grafiği.	49
Şekil 3.11. “Düğüm sayısı” giriş değişkeni üyelik fonksiyonlarının grafik görünümü.	50
Şekil 3.12. Veri gönderme sıklığına göre “enerji oran” değişimi.	52
Şekil 3.13. “Sıklık” giriş değişkeni üyelik fonksiyonlarının grafik görünümü.	53
Şekil 3.14. “Kullanılacak Algoritma” çıkış değişkeni üyelik fonksiyonlarının grafik görünümü.	53
Şekil 3.15. Örnek görüntü 1.	57
Şekil 3.16. Örnek görüntü 2.	57
Şekil 3.17. Örnek görüntü 3.	57
Şekil 3.18. Üç algoritma için “örnek görüntü 1” verisinin sıkıştırma oranına göre PSNR değer kaybı grafiği.	58
Şekil 3.19. Üç algoritma için “örnek görüntü 2” verisinin sıkıştırma oranına göre PSNR değer kaybı grafiği.	59
Şekil 3.20. Üç algoritma için “örnek görüntü 3” verisinin sıkıştırma oranına göre PSNR değer kaybı grafiği.	60
Şekil 5.1. Senaryo 2: Mesafe-az ve düğüm sayısı-orta olduğu durumda dört yöntemin veri gönderme sıklığına göre enerji oran grafiği.	77
Şekil 5.2. Senaryo 4: Mesafe-orta ve düğüm sayısı-az olduğu durumda dört yöntemin veri gönderme sıklığına göre enerji oran grafiği.	78
Şekil 5.3. Senaryo 7: Mesafe-çok ve düğüm sayısı-az olduğu durumda dört yöntemin veri gönderme sıklığına göre enerji oran grafiği.	79

Şekil 5.4. Senaryo 11: Mesafe-orta ve veri gönderme sıklığı-az olduğu durumda dört yöntemin düğüm sayısına göre enerji oran grafiği.	80
Şekil 5.5. Senaryo 14: Mesafe-orta ve veri gönderme sıklığı-orta olduğu durumda dört yöntemin düğüm sayısına göre enerji oran grafiği.	81
Şekil 5.6. Senaryo 17: Mesafe-orta ve veri gönderme sıklığı-çok olduğu durumda dört yöntemin düğüm sayısına göre enerji oran grafiği.	82
Şekil 5.7. Senaryo 19: Düğüm sayısı-az ve veri gönderme sıklığı-az olduğu durumda dört yöntemin mesafeye göre enerji oran grafiği.	83
Şekil 5.8. Senaryo 22: Düğüm sayısı-orta ve veri gönderme sıklığı-az olduğu durumda dört yöntemin mesafeye göre enerji oran grafiği.	84
Şekil 5.9. Senaryo 25: Düğüm sayısı-çok ve veri gönderme sıklığı-az olduğu durumda dört yöntemin mesafeye göre enerji oran grafiği.	85
Şekil 5.10. Senaryo 3: Mesafe-az ve düğüm sayısı-çok olduğu durumda dört yöntemin veri gönderme sıklığına göre enerji oran grafiği.	87
Şekil 5.11. Senaryo 5: Mesafe-orta ve düğüm sayısı-orta olduğu durumda dört yöntemin veri gönderme sıklığına göre enerji oran grafiği.	88
Şekil 5.12. Senaryo 8: Mesafe-çok ve düğüm sayısı-orta olduğu durumda dört yöntemin veri gönderme sıklığına göre enerji oran grafiği.	89
Şekil 5.13. Senaryo 12: Mesafe-çok ve veri gönderme sıklığı-az olduğu durumda dört yöntemin düğüm sayısına göre enerji oran grafiği.	90
Şekil 5.14. Senaryo 15: Mesafe-çok ve veri gönderme sıklığı-orta olduğu durumda dört yöntemin düğüm sayısına göre enerji oran grafiği.	91
Şekil 5.15. Senaryo 18: Mesafe-çok ve veri gönderme sıklığı-çok olduğu durumda dört yöntemin düğüm sayısına göre enerji oran grafiği.	92
Şekil 5.16. Senaryo 20: Düğüm sayısı-az ve veri gönderme sıklığı-orta olduğu durumda dört yöntemin mesafeye göre enerji oran grafiği.	93
Şekil 5.17. Senaryo 23: Düğüm sayısı-orta ve veri gönderme sıklığı-orta olduğu durumda dört yöntemin mesafeye göre enerji oran grafiği.	94
Şekil 5.18. Senaryo 26: Düğüm sayısı-çok ve veri gönderme sıklığı-orta olduğu durumda dört yöntemin mesafeye göre enerji oran grafiği.	95
Şekil 5.19. Senaryo 4: Mesafe-orta ve düğüm sayısı-az olduğu durumda dört yöntemin veri gönderme sıklığına göre paket ulaştırma oran grafiği.	97
Şekil 5.20. Senaryo 14: Mesafe-orta ve veri gönderme sıklığı-orta olduğu durumda dört yöntemin düğüm sayısına göre paket ulaştırma oran grafiği.	98
Şekil 5.21. Senaryo 22: Düğüm sayısı-orta ve veri gönderme sıklığı-az olduğu durumda dört yöntemin mesafeye göre paket ulaştırma oran grafiği.	99
Şekil 5.22. Senaryo 3'e göre dört yöntemin harcadığı enerjilerinin zamana bağlı karşılaştırılması.	104

ÇİZELGE LİSTESİ

Sayfa No

Çizelge 2.1. KMAA’larda görüntü sıkıştırma algoritmalarının analizi.	29
Çizelge 2.2. DCT için orijinal ve sıkıştırılmış görüntülerin karşılaştırılması.	31
Çizelge 2.3. DCT algoritmasının görüntüyü sıkıştırma zaman analizi.	31
Çizelge 2.4. SPIHT için orijinal ve sıkıştırılmış görüntülerin karşılaştırılması.	32
Çizelge 2.5. SPIHT algoritmaların görüntüyü sıkıştırma zaman analizi.	32
Çizelge 2.6. LEICA için orijinal ve sıkıştırılmış görüntülerin karşılaştırılması.	33
Çizelge 2.7. LEICA algoritmaların görüntüyü sıkıştırma zaman analizi.	33
Çizelge 2.8. Belirlenen algoritmaların görüntü sıkıştırma performanslarının karşılaştırılması.	34
Çizelge 2.9. Belirlenen algoritmaların görüntüyü sıkıştırma zaman analizi.	34
Çizelge 3.1. Düğümler arasındaki mesafeye göre “enerji oran” değişim değerleri.	46
Çizelge 3.2. “Mesafe” giriş değişkeni üyelik fonksiyonu sınırları.	48
Çizelge 3.3. Kullanılan toplam düğüm sayısına göre “enerji oran” değişim değerleri. .	49
Çizelge 3.4. “Düğüm sayısı” giriş değişkeni üyelik fonksiyonu sınırları.	50
Çizelge 3.5. Veri gönderme sıklığına göre “enerji oran” değişimi değerleri.	51
Çizelge 3.6. “Sıklık” giriş değişkeni üyelik fonksiyonu sınırları.	52
Çizelge 3.7. “Sonuç” çıkış değişkeni üyelik fonksiyonu sınırları.	53
Çizelge 3.8. Örnek giriş değerleri için bulanık mantık yöntemi ile hesaplanmış “Kullanılacak Algoritma” değerleri.	55
Çizelge 4.1. Algoritmaların işlem süreleri ve adım sayıları.	64
Çizelge 4.2. Kullanılan algoritmaların görüntü sıkıştırma performansları.	65
Çizelge 5.1. Gerçekleştirilen senaryolar.	72
Çizelge 5.2. Senaryo 1: Mesafe-az ve düğüm sayısı-az olduğu durumda kullanılan veri gönderme sıklığına göre sıkıştırma oranı ve PSNR değerleri.	74
Çizelge 5.3. Senaryo 14: Düğümler arası mesafe-orta ve veri gönderme sıklığı-orta olduğu durumda düğüm sayısına göre kullanılan sıkıştırma oranı ve PSNR değerleri.	75
Çizelge 5.4. Senaryo 22: Düğüm sayısı-orta ve veri gönderme sıklığı-az olduğu durumda düğümler arası mesafeye göre kullanılan sıkıştırma oranı ve PSNR değerleri.	75
Çizelge 5.5. Senaryo 2: Mesafe-az ve düğüm sayısı-orta olduğu durumda beş yöntemin veri gönderme sıklığına göre enerji oran değerleri.	77
Çizelge 5.6. Senaryo 4: Mesafe-orta ve düğüm sayısı-az olduğu durumda beş yöntemin veri gönderme sıklığına göre enerji oran değerleri.	78
Çizelge 5.7. Senaryo 7: Mesafe-çok ve düğüm sayısı-az olduğu durumda beş yöntemin veri gönderme sıklığına göre enerji oran değerleri.	79
Çizelge 5.8. Senaryo 11: Mesafe-orta ve veri gönderme sıklığı-az olduğu durumda beş yöntemin düğüm sayısına göre enerji oran değerleri.	80

Çizelge 5.9. Senaryo 14: Mesafe-orta ve veri gönderme sıklığı-orta olduğu durumda beş yöntemin düğüm sayısına göre enerji oran değerleri.....	81
Çizelge 5.10. Senaryo 17: Mesafe-orta ve veri gönderme sıklığı-çok olduğu durumda beş yöntemin düğüm sayısına göre enerji oran değerleri.....	82
Çizelge 5.11. Senaryo 19: Düğüm sayısı-az ve veri gönderme sıklığı-az olduğu durumda beş yöntemin mesafeye göre enerji oran değerleri.	83
Çizelge 5.12. Senaryo 22: Düğüm sayısı-orta ve veri gönderme sıklığı-az olduğu durumda beş yöntemin mesafeye göre enerji oran değerleri.	84
Çizelge 5.13. Senaryo 25: Düğüm sayısı-çok ve veri gönderme sıklığı-az olduğu durumda beş yöntemin mesafeye göre enerji oran değerleri.	85
Çizelge 5.14. Senaryo 3: Mesafe-az ve düğüm sayısı-çok olduğu durumda beş yöntemin veri gönderme sıklığına göre enerji oran değerleri.....	87
Çizelge 5.15. Senaryo 5: Mesafe-orta ve düğüm sayısı-orta olduğu durumda beş yöntemin veri gönderme sıklığına göre enerji oran değerleri.....	88
Çizelge 5.16. Senaryo 8: Mesafe-çok ve düğüm sayısı-orta olduğu durumda beş yöntemin veri gönderme sıklığına göre enerji oran değerleri.....	89
Çizelge 5.17. Senaryo 12: Mesafe-çok ve veri gönderme sıklığı-az olduğu durumda beş yöntemin düğüm sayısına göre enerji oran değerleri.....	90
Çizelge 5.18. Senaryo 15: Mesafe-çok ve veri gönderme sıklığı-orta olduğu durumda beş yöntemin düğüm sayısına göre enerji oran değerleri.....	91
Çizelge 5.19. Senaryo 18: Mesafe-çok ve veri gönderme sıklığı-çok olduğu durumda beş yöntemin düğüm sayısına göre enerji oran değerleri.....	92
Çizelge 5.20. Senaryo 20: Düğüm sayısı-az ve veri gönderme sıklığı-orta olduğu durumda beş yöntemin mesafeye göre enerji oran değerleri.	93
Çizelge 5.21. Senaryo 23: Düğüm sayısı-orta ve veri gönderme sıklığı-orta olduğu durumda beş yöntemin mesafeye göre enerji oran değerleri.	94
Çizelge 5.22. Senaryo 26: Düğüm sayısı-çok ve veri gönderme sıklığı-orta olduğu durumda beş yöntemin mesafeye göre enerji oran değerleri.	95
Çizelge 5.23. Senaryo 4: Mesafe-orta ve düğüm sayısı-az olduğu durumda dört yöntemin veri gönderme sıklığına göre paket ulaştırma oran değerleri.	97
Çizelge 5.24. Senaryo 14: Mesafe-orta ve veri gönderme sıklığı-orta olduğu durumda dört yöntemin düğüm sayısına göre paket ulaştırma oran değerleri.	98
Çizelge 5.25. Senaryo 22: Düğüm sayısı-orta ve veri gönderme sıklığı-az olduğu durumda dört yöntemin mesafeye göre paket ulaştırma oran değerleri.....	99
Çizelge 5.26. Sabit ağda senaryo 3, 14 ve 22 için GSD-EDUKA ve sıkıştırma uygulamadan harcanan toplam enerji değerleri.	100
Çizelge 5.27. Hareketli ağda senaryo 6, 16 ve 24 için GSD-EDUKA ve sıkıştırma uygulamadan harcanan toplam enerji değerleri.	100
Çizelge 5.28. GSD-EDUKA ile senaryo 11 boyunca ölçüm yapılan zamana göre düğümün kullandıkları sıkıştırma yöntemleri.....	101
Çizelge 5.29. Senaryo 11 için kullanılan yöntemlere göre toplam ortalama sıkıştırma süreleri.....	102
Çizelge 5.30. 17 düğümlük senaryo için kullanılan yöntemlere göre bir paketin uçtan uca iletilme süresi.	102

Çizelge 5.31. 101 düğümlük senaryo için kullanılan yöntemlere göre bir paketin uçtan uca iletilme süresi.	103
Çizelge 5.32. 197 düğümlük senaryo için kullanılan yöntemlere göre bir paketin uçtan uca iletilme süresi.	103
Çizelge A.1. Toplam oluşabilecek durumlar.	125
Çizelge A.2. Birinci kök için değişken değer dağılımı.	126
Çizelge A.3. Düğümler arası mesafe için veri seti.	127
Çizelge A.4. Düğüm sayısı için veri seti.	127
Çizelge A.5. Veri gönderme sıklığı için veri seti.	128
Çizelge A.6. Birinci kök için bilgi kazancı sonuçları.	129
Çizelge A.7. “Az” durumu için değişken değer dağılımı.	129
Çizelge A.8. “Az” durumu için düğüm sayısı veri seti.	130
Çizelge A.9. “Az” durumu için veri gönderme sıklığı veri seti.	131
Çizelge A.10. “Az” durumu için bilgi kazancı sonuçları.	131
Çizelge A.11. “Orta” durumu için değişken değer dağılımı.	132
Çizelge A.12. “Orta” durumu için düğüm sayısı veri seti.	132
Çizelge A.13. “Orta” durumu için veri gönderme sıklığı veri seti.	133
Çizelge A.14. “Orta” durumu için bilgi kazancı sonuçları.	134
Çizelge A.15. “Çok” durumu için değişken değer dağılımı.	134
Çizelge A.16. “Çok” durumu için düğüm sayısı veri seti.	135
Çizelge A.17. “Çok” durumu için veri gönderme sıklığı veri seti.	136
Çizelge A.18. “Çok” durumu için bilgi kazancı sonuçları.	136

KISALTMALAR

AODV	Ad hoc On Demand Distance Vector Routing
AWICAO	Adaptive Wavelet Image Compression And Organization Algorithm
DCT	Discrete Cosine Transform
DWT	Discrete Wavelet Transform
EBCOT	Embedded Block Coding with Optimized Truncation
EECA	Energy Efficient Congestion Avoidance
EHPF	Elimination High Pass Filter
EZW	Embedded Zerotree Wavelet
FIC	Fractal Image Compression
GPS	Global Positioning System
IEEE	The Institute of Electrical and Electronics Engineers
JPEG	Joint Photographic Experts Group
KAA	Kablosuz Algılayıcı Ağ
KMAA	Kablosuz Multimedya Algılayıcı Ağ
GSD-EDUKA	Kablosuz Multimedya Algılayıcı Ağlar için Görüntü Sıkıştırma Dayalı Yeni Bir Enerji Duyarlı Uygulama Katmanı Algoritması
LEICA	Low Energy Image Compression Algorithm
MAC	Media Access Control
MSE	Mean Square Error
M-AODV	Modify - Ad hoc On Demand Distance Vector Routing
NS-2	Network Simulator - 2
OSI	Open Systems Interconnection
PSNR	Peak Signal to Noise Ratio
RLC	Run Length Coding
ROI	Regions of Interest
SHPS	Skipped High Passsub Band
SNR	Signal to Noise Ratio
SOM	Self Organizing Map
SOT	Spatial Orientation Tree
SPIHT	Set Partitioning in Hierarchical Trees
UDP	User Datagram Protocol
YBS	Yapay Bağışıklık Sistemleri

ÖZET

KABLOSUZ MULTİMEDYA ALGILAYICI AĞLAR İÇİN GÖRÜNTÜ SIKIŞTIRMAYA DAYALI YENİ BİR ENERJİ DUYARLI UYGULAMA KATMANI ALGORİTMASI GELİŞTİRME

Arafat ŞENTÜRK

Düzce Üniversitesi

Fen Bilimleri Enstitüsü, Elektrik-Elektronik ve Bilgisayar Mühendisliği Anabilim Dalı
Doktora Tezi

Danışman: Doç. Dr. Resul KARA

Şubat 2017, 161 sayfa

Kablosuz Algılayıcı Ağ (KAA)'lar, algılayıcı düğümlerin veri işleme ve hesap yapma yeteneğini kablosuz kanal üzerinden gerçekleştirebilen ve birçok iletişim cihazlarından oluşan ağlardır. Kablosuz Multimedya Algılayıcı Ağ (KMAA)'lar ise, ses, resim veya video gibi gerçek zamanlı multimedya verilerini birbirlerine veya baz istasyonuna aktarabilen düşük maliyetli donanıma sahip algılayıcı düğümlerin oluşturduğu ağlardır. KAA'lara göre daha fazla boyuta sahip verileri transfer edebilen KMAA'lar, daha fazla enerjiye ve bant genişliğine ihtiyaç duymaktadır. Algılayıcı düğümlerin birbirlerine ve baz istasyonuna ilettikleri verinin büyüklüğü enerjilerinin harcanmasında önemli bir etken olmaktadır. Enerji tüketimi KMAA'lar için temel bir sorundur. KMAA'ların gelişimini etkileyen diğer önemli sorunlar ise, sınırlı bant genişliği ve bellek sınırlamasıdır. Düğüm batarya ömürlerinin önemli kaynak olduğu bu ağlarda, çevreden alınan verinin işlenerek gereksiz bilgilerin atılması ile iletilen veri boyutu azaltılarak sınırlı kaynakların verimli bir şekilde kullanılması gerekmektedir. Bu tez çalışmasında, KMAA üzerindeki algılayıcı düğümlerin birbirlerine veya baz istasyonuna görüntü aktarımı sağlarken harcadıkları enerjiyi en aza indirerek, algılayıcı düğümler için en önemli kaynak olan batarya ömrünü etkin bir biçimde kullanılmasını sağlayan yeni bir algoritma geliştirilmiştir. Kablosuz Multimedya Algılayıcı Ağlar için Görüntü Sıkıştırma Dayalı Yeni Bir Enerji Duyarlı Uygulama Katmanı Algoritması (GSD-EDUKA) olarak isimlendirilen bu algoritma, KMAA'larda kullanılan en iyi üç görüntü sıkıştırma algoritmasından faydalanmaktadır. GSD-EDUKA, bu görüntü sıkıştırma algoritmalarından belirli parametrelere (düğümler arası mesafe, toplam düğüm sayısı ve veri gönderme sıklığı) göre en verimli olanına anlık olarak karar vererek algılayıcı düğümlerin batarya ömürlerini en etkin biçimde kullanmalarını sağlamaktadır. Öncelikle kullanılacak görüntü sıkıştırma algoritmasını seçme işlemi için "karar ağacı yöntemi" kullanılarak elde edilen çıkarımların değerleri daha aza indirgenmiştir. Bu sayede GSD-EDUKA'nın işlem yükü azaltılmıştır. Daha sonra aza indirilen çıkarımlar, kural tabanlı bilgilere dayandırılarak "bulanık mantık yöntemi" kullanılması sağlanmıştır. Yapılan bu işlemlerin sonucunda, GSD-EDUKA için karar verme mekanizması gerçekleştirilmiştir. Ek olarak, en verimli görüntü sıkıştırma algoritması ile sıkıştırılan görüntünün kablosuz iletişimde kabul edilebilir PSNR (Peak Signal to Noise Ratio) kayıp değeri kontrol edilerek bu değerin sağlandığı duruma kadar algoritmanın sıkıştırma oranı artırılmıştır. Bu sayede kablosuz iletişimde kabul edilebilir PSNR değerinden ödün vermeden görüntüyü daha fazla sıkıştırarak görüntü

aktarımında daha az enerji harcanması sağlanmıştır. Geliştirilen algoritma, Network Simulator – 2 (NS - 2) ile başarımlı değerlendirilerek, enerji oranı, paket ulaşım oranı ve PSNR değeri kaybı açısından diğer algoritmalarla karşılaştırmalı olarak sunulmaktadır.

Anahtar sözcükler: Kablosuz multimedia algılayıcı ağlar, Görüntü sıkıştırma, Enerji verimliliği



ABSTRACT

DEVELOPING A NEW ENERGY-AWARE APPLICATION LAYER ALGORITHM FOR WIRELESS MULTIMEDIA SENSOR NETWORKS BASED ON IMAGE COMPRESSION

Arafat ŞENTÜRK

Duzce University

Graduate School of Natural and Applied Sciences, Department of Electrical-Electronics
and Computer Engineering

Doctoral Thesis

Supervisor: Assoc. Prof. Dr. Resul KARA

February 2017, 161 pages

Wireless Sensor Networks (WSN) are the networks which realizes data processing and computation on wireless channels and they are composed of several communication devices. As for Wireless Multimedia Sensor Networks (WMSN), they are the networks which may transmit multimedia data like audio, image or video to each other (or to sink) with low cost hardware. WMSN needs more energy and bandwidth than WSN since they transfer larger amount of data. The size of the transmitted data is an important factor in terms of the energy consumption of the nodes. Energy consumption is a fundamental issue in WMSN. Other important issues related to WMSN are bandwidth and memory limitations. Limited resources must be used efficiently by removing redundant information after sensing it from the environment, and so decreasing the amount of transmitted data. In this thesis, a new algorithm which uses the battery life of the nodes effectively by reducing the data amount during image transmission to the nodes (or to sink) is developed. This algorithm, named as Energy-Aware Application Layer Algorithm based on Image Compression Algorithms for WMSN (GSD-EDUKA), makes use of the best three image compression algorithms used in WMSN. GSD-EDUKA makes the nodes use their battery efficiently by deciding instantly which compression algorithm is the best according to certain parameters (distance between the nodes, total number of nodes, and data transmission frequency). Firstly, the amount of the arguments for selecting the compression algorithm is minimized by using Decision Tree Method. By this means, the computation load of GSD-EDUKA is reduced. Then, these reduced arguments are based upon rule based information and Fuzzy Logic Method is used. GSD-EDUKA makes a decision at the end of these operations. Additionally, acceptable PSNR (Peak Signal to Noise Ratio) loss values are iteratively checked and the images are compressed by the best compression method until the lower bound of PSNR loss value is reached. Thus, the wireless communication is provided with less energy consumption in image transmission without sacrificing acceptable PSNR value. The developed algorithm is represented comparatively with other algorithms in terms of energy rate, packet access ratio, and PSNR loss value after performance analysis realized on Network Simulator – 2 (NS - 2).

Keywords: Wireless multimedia sensor networks, Image compression, Energy efficiency



EXTENDED ABSTRACT

DEVELOPING A NEW ENERGY-AWARE APPLICATION LAYER ALGORITHM FOR WIRELESS MULTIMEDIA SENSOR NETWORKS BASED ON IMAGE COMPRESSION

Arafat ŞENTÜRK

Duzce University

Graduate School of Natural and Applied Sciences, Department of Electrical-Electronics
and Computer Engineering

Doctoral Thesis

Supervisor: Assoc. Prof. Dr. Resul KARA

February 2017, 161 pages

1. INTRODUCTION

Wireless Multimedia Sensor Networks (WMSN) enables to develop many applications that address different areas like mobile health, environmental monitoring, and traffic monitoring. During those applications, size of the data causes scarcity in memory resources, difficulties in computations, and extreme energy consumption which is the most important one for each node while WMSN is transmitting multimedia data. This problem has a big importance for WMSN which already has limited resources. Image compression could be one of the most effective solutions to overcome those issues. Thus, the lifetime of WMSN could be improved significantly and the bandwidth could be used more effectively.

Meeting the energy needs of sensor nodes is quite difficult in field conditions since generally mobile sensor nodes are preferred in WMSN applications. That's why the nodes must use their energies effectively.

The purpose of this thesis is to make the nodes use their most important resource, battery effectively by minimizing the energy required for image transfer between the nodes and the sink on a WMSN. For this purpose, the nodes will first reduce the size of the image thanks to the embedded image compression algorithm and then they will send

it to other nodes or the base station. Thus, sending energy that is required to send an image to the destination and receiving energy that is required to receive an image from the source will be less. In addition, the image compression algorithms that will be used by the nodes must consume quite few energy. Also, these image compression algorithms must have less operation steps and computational complexity for the nodes with limited CPU and computational power. Computation speed is very important for the applications with real-time image transfer.

In this thesis work, an algorithm named as Energy-Efficient Application Layer Algorithm based on Image Compression Algorithms for Wireless Multimedia Sensor Networks is developed. The most suitable image compression algorithm is selected depending on predetermined parameters to make the nodes use their energies in the best way.

2. MATERIAL AND METHODS

Image compression techniques used for WMSN is divided into two groups as lossy and lossless. Lossy image compression techniques have higher compression rate than the lossless compression techniques. Lossy image compression techniques are used in WMSN since they consume less energy during image transmission. DCT, SPIHT, and LEICA come into prominence after the image compression algorithms are analyzed in terms of energy efficiency. For this purpose, these three image compression algorithms are used in the developed energy-aware application layer algorithm based on image compression algorithms for WMSN (GSD-EDUKA). These three algorithms are coded by C programming language to provide the consistency with the simulator NS-2 (Network Simulator- 2) and the performance measurements are realized.

It is determined that the algorithms show different performances when the distance between the nodes, total number of nodes and data sending frequency are different according to the results of the simulations. The parameters of the developed GSD-EDUKA are determined as “distance between the nodes”, “number of nodes”, and “data frequency” after this inference. GSD-EDUKA considers these parameters and determines which algorithm among the three is the most efficient one. Decision tree method is used to reduce the number of operations for the decision after checking all possible states of these parameters. Thus, the number of states is minimized.

Determination of the most suitable algorithm is realized via fuzzy logic method.

3. RESULTS AND DISCUSSIONS

Wireless multimedia sensor networks are the networks in which high amounts of data is transferred and so the energy consumption is much more than the networks which carry scalar data. Most of the energy in a network is consumed during the transmission of data between the nodes. Energy consumption is proportional to the amount of data to be transmitted. There are several solutions to this energy problem in the literature such as transmission through the shortest path, locating the node to the proper places according to their tasks, putting the nodes to sleep, etc. Recently, besides these solutions, the images (or video and audio data) are compressed after they are captured and the compressed data is transmitted through the route. Therefore, fewer amounts of data are transmitted along the path. This approach provides a more energy efficient way than the solutions which are optimal forwarding schema, optimal node placement and making idle nodes sleep.

In this thesis work, a more effective method than the existing compression methods for WMSN is proposed. The proposed algorithm differs from the others because it decides the most efficient compression algorithm depending on the parameters ‘distance between the nodes’, ‘number of nodes’, and ‘data frequency’. Thus, the maximum efficiency is ensured by the selection of the most suitable compression algorithm according to the instant situation. The method provides up to 51% energy saving in stationary networks and 36 % in mobile networks when compared to the traditional networks that use the same compression method for every situation without considering the instant situation of the network.

4. CONCLUSION AND OUTLOOK

The performance measurements of the developed GSD-EDUKA method are done in terms of some evaluation criteria. These criteria are Energy Ratio (Residual Energy/Total Energy), Packet Transmission Ratio (Received Packet/ Produced Packet), Energy Ratio according to PSNR value loss, and total energy consumption values. Different scenarios are constituted and the results of GSD-EDUKA are obtained for different values of its parameters.

Different sample scenarios are thought and the nodes transfer the received images separately by DCT, SPIHT, LEICA, GSD-EDUKA, and directly without compression to the neighboring node. Different simulations are performed for stationary and mobile networks.

According to the results of the most of the simulations, GSD-EDUKA provides energy efficiency by using the most efficient energy compression algorithm according to the determined parameters. However, it is unable to outperform the existing most efficient compression algorithm when only this algorithm is selected during the simulation. In the future, the topics given below can be studied related to the proposed method:

- a. The proposed application layer method based on image compression for energy saving can be combined with the energy efficiency methods developed for other layers and a hybrid network design with longer lifetime can be realized.
- b. The efficiency of the proposed method can be tested by the cameras with different resolutions, the efficiency of the system can be evaluated in case of high resolution images, and the method can be converted to a more generic form by considering the parameter ‘resolution’.

1. GİRİŞ

Her geçen gün gelişen ve büyüyen dünyamız ile birlikte bilgiye anlık ve daha fazla ihtiyaç duyulması beklenmedik bir durum değildir. Bu ihtiyaçlar zaman içerisinde kablolu iletişim tarafından sağlanamayacak hale gelmiştir. Dolayısıyla birçok alanda kablolu iletişimin yerini kablosuz iletişim almıştır. Kablosuz iletişim ile beraber kablosuz ağ teknolojileri de ilerlemiştir.

Kablosuz Algılayıcı Ağ (KAA)'lar, algılayıcı düğümlerin veri işleme ve hesap yapma yeteneğini kablosuz kanal üzerinden gerçekleştirebilen ve birçok iletişim cihazlarından oluşan ağlardır. KAA'lar doğa koşullarında kablolu iletişime göre daha uygun olması, güvenilir olması, esnek yapıya sahip olması, düşük maliyet ile çözüm üretmesi, kendi kendine organize olabilmesi, enerjisini ağ içinde dengeli bir şekilde kullanabilmesi ve kurulum kolaylığı sağlaması gibi özelliklerinden dolayı kolayca yaygınlaşmıştır. KAA'ların amacı, bilgiye her an, her yerden kolayca erişimi sağlamaktır. Bu işlevi, veriyi toplayarak, işleyerek, çözümleyerek ve yayarak yerine getirirler [1], [2]. Son yıllarda KAA'lar teorik ve pratik zorlukları kolayca çözebildiğinden araştırmacıların dikkatini çekmiştir. Bu artan ilginin sebebi fiziksel ortamlarda veriler üzerinde basit işlemler yapıldıktan sonra uzak mesafelere iletmek için küçük cihazların kullanılmasıdır. Ayrıca, KAA'lar sıcaklık, basınç, nem ve konum gibi fiziksel olguları ölçmekte fayda sağladığı için birçok sivil ve askeri uygulamada kullanılmıştır [3].

Ağlar, yapısal ve rastgele dağıtık ağlar olmak üzere iki gruba ayrılmıştır. KAA'lar, rastgele dağıtık olarak oluşturulmuş ağlar grubuna girmektedir ve yapısal ağlardan bazı yönleriyle ayrılmaktadırlar [4], [5]. Bu yönlerden en önemlisi, KAA'ların uzun süre çalışmak zorunda olup, sınırlı batarya ömrüne sahip olmalarına rağmen, enerjisini etkin bir şekilde kullanmasıdır. Bu ağlar belli bir topolojiye sahip değildir. Veri, algılayıcı düğümler arasında iletilerek baz istasyonuna ulaştırılır ve üzerinde işlemler yapılabilir hale getirilir [6].

Ek olarak, ağda bulunan algılayıcı düğümlerin bazılarının veya tamamının enerji kaynaklarının bitmesi veya hareket halinde olanların da aynı zamanda konumlarının

değişmeleri sebebiyle ağdaki ömürlerini kaybedebilirler [1]. Dolayısıyla enerji kaynaklarının değiştirilemeyeceği veya şarj edilemeyecek bir ortamda bulunan algılayıcı düğümlerden en iyi şekilde faydalanmak ve oluşturulan ağın ömrünü uzatmak için düğümlerin enerji kaynaklarının verimli bir şekilde kullanılması gerekmektedir [7].

1.1 KABLOSUZ MULTİMEDYA ALGILAYICI AĞLAR VE KULLANIM ALANLARI

Kablosuz Multimedya Algılayıcı Ağ (KMAA)'lar gerçek zamanlı multimedya verilerini (ses, resim veya video gibi) birbirlerine veya baz istasyonuna aktaran ucuz donanıma sahip algılayıcı düğümlerin oluşturduğu ağlardır. Tek bir cihazda ses ve görsel bilgi bir arada bulunabildiği gibi ayrı ayrı da bulunabilir. Bunun yanında, KMAA'lar gerçek zamanlı veriyi birçok algılayıcı düğümden alarak depolayabilmektedir [8]. Gerçek zamanlı multimedya izlemede en umut verici gelişmeleri KMAA'lar oluşturmaktadır. Ancak, KAA'lara göre ilettikleri veri boyutunun oldukça fazla olması, KMAA'ların kullanıldığı uygulamalarda ağ ömrünün uzatılması ve verimli enerji kullanımı gibi konular çözüm bekleyen önemli problemler haline gelmiştir [9]. Enerji tüketimi KMAA'lar için temel bir sorundur. Ağdaki algılayıcı düğümler enerjilerinin büyük bir bölümünü veri toplama ve gerçekleştirilen işlemler için harcarlar. Bu nedenle, KAA'larının ömrünü maksimize etmek için uygun bir tasarım gereklidir. KMAA'ların gelişimini etkileyen diğer önemli sorunlar ise, sınırlı bant genişliği ve bellek sınırlamasıdır [1], [6].

KMAA'lardan elde edilen veriler iletişim ve hesaplama açısından sinyal işleme, kontrol teorisi ve gömülü bilgisayar sistemleri gibi çalışma alanlarının da kapsamına girmektedir [10]. KMAA'lar gözetim, trafik uygulama ve kontrol sistemleri, gelişmiş sağlık hizmetleri, yapısal sağlık izleme ve endüstriyel süreç kontrolü gibi yeni uygulamalarda kullanılmıştır. KMAA'lardaki algılayıcı düğümler, video kameralar ve yüksek hesaplama yeteneklerine sahip oldukları için birçok uygulamaya olanak sağlamıştır [11].

KMAA'lar daha çok gözetim amacı ile kullanılmaktadır. Video ve ses algılayıcıları kullanılarak suç ve terörist saldırılarına karşı gözetim sistemleri geliştirilebilir [10], [12]. Ayrıca, KMAA'lardan alınan görüntülerin sınıflandırıcılar kullanılarak sınıflandırılması uygulamanın kullanılabilirliğini büyük ölçüde arttırabilir [13], [14].

Küçük çaptaki habitat gözlemlene sistemlerinin, bitki ve hayvanların doğal yaşam alanlarına kurulması bu canlılar üzerinde psikolojik ve fizyolojik etkiler oluşturabilir [15].

Otoyollarda araç trafiği izlemek ve sıklığı gidermek amacı ile KMAA'lar sıklıkla kullanılmaktadır. Karayollarına yerleştirilen KMAA'lardan araçların ortalama hızları ve yoldaki araç sayısı gibi bilgileri almak üzere faydalanılmaktadır [16], [17].

Çevresel birçok uygulamada video ile çevreyi izleyerek görüntüler kayıt altına alınabilir ve daha sonra bu kayıtlara dayanarak gerekli bilgilere ulaşılabilir [18]. Ayrıca, KMAA'lar makine görme sistemlerinin uyumunu kolaylaştırmak ve sürekli çalışma gerektiren görsel denetim ve otomatik eylem sistemleri için esneklik kazandırabilirler [19].

1.2 KMAA'LAR İÇİN ENERJİ VERİMLİLİĞİ İLE İLGİLİ YAPILAN ÇALIŞMALAR

Multimedya verilerinin aktarılması geleneksel algılayıcı ağlara göre daha yüksek bant genişliği gerektirir ve bu durum da KMAA'ların tasarımını etkilemektedir [20]. Yüksek sıkıştırma kabiliyetinin yanı sıra, işlem karmaşası daha az olan kodlayıcı ve kod çözücülerin tasarlanması, enerji korunumunun oldukça gerekli olduğu KMAA'larda önemli bir uygulama katmanı problemidir [20]. KMAA'larda çevreden alınan ham verinin işlenmesi için algoritmalar geliştirilerek gereksiz bilgiler yerine önemli bilgiler iletilir. Böylece iletilen veri boyutu azaltılarak enerji sarfiyatı düşürülebilir [21]. Güvenilirlik, tıkanıklık kontrolü, gecikme minimizasyonu ve hata kontrolü servis kalitesi bakımından taşıma katmanında sağlanan gereksinimlerdir. Servis kalitesi tabanlı rota bulunması ve yönlendirme şemaları ağ katmanında çalışılan konulardır. MAC katmanı protokolleri ise öncelik tabanlı ve gecikme duyarlı zamanlama algoritmaları içermelidir [20]. Bu katman radyo iletişiminin büyük kısmını kontrol eder ve bu sebeple enerji verimliliği ve gecikmenin azaltılması konusunda önemli rol oynamaktadır [22]. Aşağıda KMAA'lar için önerilen enerji duyarlı protokoller katmanlı mimari baz alınarak açıklanmıştır. Fiziksel katmandan başlanarak uygulama katmanına kadar literatürde yer alan enerji verimliliği ile ilgili çalışmalara yer verilmiştir.

1.2.1 Fiziksel Katman

KAA'lar farklı fiziksel teknolojiler ile uygulanabilir. Genellikle, radyo etkin (beacon enabled) ve radyo etkin olmayan şeklinde iki ayarda çalışan IEEE 802.15.4 teknolojisi kullanılmaktadır [20]. Multimedya iletişimde gecikmenin daha az, veri oranının ise daha yüksek olduğu bir teknolojiye ihtiyaç duyulmaktadır. IEEE 802.15.4 radyo etkin olmayan durumda yüksek veri verimliliği sağlasa da enerji tüketimi oldukça fazla olmaktadır. [23]'de enerji tasarruflu 802.15.4 üzerinde durulmuştur. Yöntem, radyo etkin durumda trafik bilgisine göre aktifliği ayarlayarak daha az gecikme ile birlikte enerjiden de tasarruf sağlamaktadır. [24]'te KMAA'ların performansını uçtan uca gecikmeyi ve paket kayıplarını azaltmak suretiyle artırmayı amaçlayan, bunun için de RED (Random Early Detection- Rastgele Erken Tespit) kuyruk yönetimi ile birlikte radyo aralığı ve çerçeve sürecinin ayarlanmasına dayalı bir yöntem önerilmiştir. Multimedya algılayıcı ağlarda karşılaşılan farklı trafik türleri için farklı servisler sunmak üzere IEEE 802.11e standardı geliştirilmiştir. Geliştirilen pek çok uygulamada da yüksek verim ve daha az gecikme için bu standarttan faydalanılmaktadır [25], [26].

1.2.2 MAC (Medium Access Control- Ortam Erişim Kontrolü) Katmanı

MAC protokolleri, uygulama katmanındaki servis kalitesi gereksinimlerini karşılamaya yönelik çalışırlar. MAC katmanı, çarpışma ve yeniden gönderimlerin sayısını azaltma, enerji tüketimi ve girişimleri minimize etme, uyumluluk ve güvenilirliği maksimize etme gibi sorumlulukları olması sebebiyle diğer katmanlara göre ayrı bir öneme sahiptir [20]. [25]'de CSMA-CA (Carrier Sense Multiple Access with Collision Avoidance – Çarpışma Korumalı Taşıyıcı Duyarlı Çoklu Erişim) tabanlı bir MAC protokolü önerilmiştir. Algılayıcı ağlar üzerinden multimedya iletişimde servis kalitesini sağlarken enerjiyi de korumak amaçlanmıştır. Bu amaç, düğümleri ortamı ele geçirmek için gerçekleştirdikleri çekişme (contention) penceresinin uzunluğunun farklı trafik türleri için dinamik olarak belirlenmesi ile sağlanır. Enerjinin korunumu, görev çevriminin düğüm üzerindeki baskın trafik türüne göre ayarlanması ile sağlanır. Bu yöntemin başka bir versiyonunda [27] KMAA'lardaki farklı trafik akışlarına farklı servisler sağlanması şeklinde bir değişiklik önerilmiştir. [28]'de yine CSMA-CA tabanlı bir MAC protokolü önerilmiştir. Yöntemde atlanan düğüm sayılarına göre trafik sınıflarına öncelik verilmektedir. Ayrıca, çarpışma olasılığını azaltmak için rapor mesajlarına müdahale etmeye izin verir. [29]'da hem TDMA (Time Division Multiple

Access – Zaman Bölütlemeli Çoklu Erişim) hem de çekişme tabanlı CSMA birlikte kullanılarak farklı trafik akışlarının önceliğe göre programlanması amaçlanmıştır. Servis farklılığı kuyruk yönetim şeması ile sağlanmıştır. Yüksek öncelikli trafik ortama erişimde daha yüksek şansa sahip olmakta ve gecikme azaltılmaktadır. Bazı MAC protokolleri yüksek öncelikli verinin kanal erişiminde daha yüksek şansa sahip olması için uyku durumunda da dinleme yapılmasını gerektiren ek süreçler içermektedir [30], [31].

Enerji tüketimini azaltmak amacıyla pek çok düğüm, yaşam süresinin çoğunu uykuda geçirmektedir [32]. Habitat gözlemlene uygulamalarında algılayıcı düğümlerin kısıtlı batarya ile aylarca ya da yıllarca çalışması gerekebilir, bu yüzden bir düğüm zamanının çoğunu uyku durumunda geçirmek zorundadır. Algılayıcı düğümler sadece periyodik olarak hesaplama yapmak ve iletişim kurmak için uyanırlar. Her bir düğümün uyanık kalma süresi hizmet çevrimi olarak bilinir ve değişik yaklaşımlar ile düşük enerji tüketimi sağlanabilir [15].

Genel olarak MAC katmanı için önerilen protokoller farklı trafik akışları için dinamik görev çevrimleri, öncelik tabanlı kuyruk ve kanal erişimi aracılığıyla servis farklılığı sağlamaktadır. Böylece, hızlı erişim ve işleme ile gecikme duyarlı trafik ve gerçek zamanlı uygulamalarda MAC katmanı gecikmeleri azaltılmaktadır. Ayrıca, MAC katmanında çarpışma ihtimalinin azaltılması da gecikmeyi azaltma ve paketin tekrar gönderilmesi için harcanacak enerjiyi azaltma bakımından KMAA'lar için oldukça önemlidir.

1.2.3 Ağ Katmanı

Yönlendirme protokollerinde temel zorluk bir taraftan servis kalitesi gereksinimlerini sağlarken diğer taraftan da ağ performansını optimize etmek ve enerji korunumunu sağlamaktır [20]. Yönlendirme katmanındaki servis kalitesi metrikleri uçtan uca gecikme, paket kaybı, toplam atlama sayısı, bant genişliği, bağlantı kalitesi ve enerjidir. [33]'te coğrafik bir yönlendirme protokolü önerilmiştir. Önerilen protokol ile uçtan uca gecikme, baz istasyonuna göre pozisyon, kalan enerji ve bir sonraki atlanacak düğümde oluşacak kuyruğun uzunluğu değerlendirilerek garantilenmiştir. [34]'te düğümlerdeki enerji tüketimini dengeleyerek ağın ömrünü maksimize eden bir yönlendirme yaklaşımı sunulmuştur. Kabul edilebilir gecikme süresi içinde baz istasyonuna ulaşabilmek için servis ayrımı kavramı kullanılmaktadır. Uçtan uca gecikmeyi azaltmak için trafik birden

fazla alternatifli yollar üzerinden aktarılmaktadır. Veri fazlalığı oluşturmadağı için de genel verim artırılmış olmaktadır. Yöntem atlanacak bir sonraki düğümü düğümlerin kalan enerjisi, bellek büyüklüğü ve SNR (Signal to Noise Ratio – Sinyalin gürültüye oranı) bilgilerini değerlendirerek belirlemektedir. Diğer bir çalışmada, ağdaki küme başları (cluster heads) baz istasyonuna giden optimal yolu bulmada karınca kolonisi optimizasyonunu kullanılmıştır [35]. Gidilen her düğümün mevcut enerjisi, ortalama kuyruk gecikmesi, paket kaybı ve hafıza verileri kaydedilmiştir. Bu verilere bağlı olarak hesaplanan olasılıklarla atlanacak bir sonraki küme başı belirlenmiştir. [36]'da farklı trafik akışları için farklı servis gereksinimleri hesaba katılmıştır. Bunlar, gecikme, paket kaybı, düğümde kalan enerji ve her rota için gereken atlama sayısı gibi çeşitli servis kısıtlarıdır. Yöntem baz istasyonunun ihtiyaçlarını belirten bir sorgu yayınlaması ve alttaki her düğümün kendi kaynaklarını kontrol edip, eğer belirlenen eşğin üzerinde kaynağa sahipse rotaya eklenmesi şeklinde çalışmaktadır. Bu yolla birden fazla rotanın belirlenebilmesi mümkün olmaktadır. Bu durumda ara düğüm sayısı en az olan rota tercih edilmektedir. [37]'de multimedya uygulamaları için gerekli bant genişliğini birbirine karışmayan rotalar aracılığıyla sağlayan ve ağ ömrünü uzatmayı amaçlayan bir çalışma gerçekleştirilmiştir. Önerilen algoritmada kaynaktan baz istasyonuna tek bir rota oluşturulur. Fakat tıkanıklık ya da bant genişliği kısıtlı durumlarında ek rotalar da adım adım oluşturulabilir. Yöntemdeki temel mantık şudur; bir rota seçildiği zaman bu rotadaki iletimi aksatabilecek tüm düğümler uyutulmaktadır. Böylelikle hem çakışmalar azaltılmakta hem de tekrar gönderim sayısı azalacağından enerji tasarrufu sağlanmaktadır. Bunlar ve benzeri iletim katmanı protokolleri optimallığın yukarıda bahsedilen gecikme, tıkanıklık kontrolü, güvenilirlik, enerji gibi servis kalitesi parametreleri ile belirlendiği optimal rotaların bulunmasını amaçlamaktadır. Ayrıca, KMAA'larda düğümlerin yerleştirilmesi yönlendirme açısından ağın performansını, kapasitesini, gecikmesini, verimliliğini ve enerji tüketimini önemli ölçüde etkiler [32], [27]. [39]'da düğüm yerleştirme ile ilgili yapılan çalışmada KMAA'larda enerjinin daha verimli kullanılması için düğümlerin yerleştirileceği konumlar hesaplanarak matematiksel bir formül geliştirilmiş ve bu formüle göre düğümler yerleştirilmiştir.

1.2.4 Taşıma Katmanı

Taşıma katmanı, algılayıcı ağlar üzerinde gecikme kısıtlı iletişimde önemli rol oynamaktadır. Enerji kısıtları ile birlikte tıkanıklık tespiti ve azaltılması, uçtan uca gecikme ve güvenilirlik gereksinimleri karşılanmalıdır [20]. KMAA'larda çok tipli

trafik akışı söz konusudur ve her trafiğin kendine mahsus güvenilirlik, hata ve gecikme kısıtları olabilmektedir. Tüm bu kısıtların her trafik akışı için ayrı ayrı sağlanıyor olması gerekmektedir. Geleneksel KAA'lardaki taşıma katmanı protokolleri KMAA'lar için uygun değildir. KAA'larda servis farkı olmaması, daha esnek gecikme sınırları olması ve uygulamaya mahsus güvenilirliktense olayın güvenilirliğinin önemli olması KMAA'ları KAA'lardan ayıran özelliklerdendir [20].

[40]'ta KAA'lar için güvenilir bir taşıma şeması önerilmiştir. Olay tespitlerinin daha güvenilir şekilde sağlanması ile enerji sarfiyatının minimuma indirildiği bir çözüm düşünülmüştür. Yöntemdeki tıkanıklık kontrol birimi, güvenilirlik ve enerji korunumu için çalışmaktadır. Algoritma, enerji tasarrufu için baz istasyonu üzerinde ve minimum işlev ile çalıştırılmaktadır. ESRT (Event to Sink Reliable Transport) isimli protokolde yapılacak işlemlere elde edilen güvenilirlik ve ağdaki tıkanıklık durumuna göre karar verilir. Eğer güvenilirlik gerekenden az ise kaynak düğümlerin raporlama frekansı hedef güvenilirliğe erişene kadar artırılır. Eğer güvenilirlik gerekenden fazla ise ESRT enerjiyi korumak için güvenilirliği istenen seviyede tutarak rapor sıklığını azaltır. [41]'de yine KAA'lar için enerji etkin tıkanıklık kontrol yapısı önerilmiştir. Çalışmada paket kaybının enerji sarfiyatı üzerindeki büyük etkisi vurgulanmıştır. [42]'de KMAA'lar için video kalitesini artırmak, paket kaybını azaltmak ve enerji tasarrufu sağlamak adına EECA (Energy Efficient Congestion Avoidance) isimli protokol geliştirilmiştir. Bu protokol ağdaki tıkanıklığı öngörerek ve tıkanıklık olmadan önce kaynak düğümlerin çıkış oranını ayarlayarak, KMAA'lardaki paket kaybı oranını en aza indirmek için geliştirilmiştir. Ayrıca, bu çalışmanın dışında yine tıkanıklık kontrolü ve güvenilirlik ile alakalı çalışmalar mevcuttur [43], [44], [45], [46], [47].

1.2.5 Uygulama Katmanı

Piksel tabanlı Wyner Ziv kodlayıcı algılayıcı ağlar için uygun bir uygulama katmanı işlemi olarak önerilmektedir [1], [48]. Fakat bu kodlayıcı, kod çözücü tarafından çok sayıda geri besleme gerektirdiğinden gecikmelere sebep olmaktadır. [49]'da piksel tabanlı Wyner Ziv kodlayıcı otomatik oran seçimi ile birlikte dilimli bir yapıda uygulanmıştır. Gereksiz parite bitlerinin gönderilmemesi ve böylelikle bant genişliğinin etkin kullanımı amaçlanmıştır. Kodlayıcı uç taraftaki bilgi kalitesini alır ve başarılı olarak kodlanacak şekilde ona uygun sayıda parite bitini gönderir. Bir başka çalışmada [50] iki aşamadan oluşan bir protokol önerilmiştir. İlk aşamada entropi tabanlı bir

ayrıklık ölçütü ile birbiriyle ilişkili kameraların birlikte kodlama yaptıklarında elde edilecek sıkıştırma kazancı ölçülmektedir. Daha sonra kodlama için optimal kümeleme hiyerarşisi oluşturulmaktadır. Böylelikle, kod çözücü tarafında sağlanan güvenilirliğin yanında global sıkıştırma kazancı da maksimize edilmektedir. Sıkıştırmanın maksimizasyonu minimum entropiye sahip kümelerin seçilmesi ile güvenilirlik ise seçilen kümelerin en az ikisinde tüm kamera düğümlerinin kapsandığının garanti edilmesi ile sağlanmaktadır.

E.Sun ve ark. [51] geliştirdikleri sıkıştırma algoritması olan Low Energy Image Compression Algorithm (LEICA) sayesinde görüntü kalitesini bozmadan verinin boyutunu küçülterek harcanan enerjiyi en aza indireceklerini savunmuşlardır. LEICA, görüntüyü öncelikle ön plan ve arka plan olmak üzere iki katmana ayırmaktadır. Bu katmanlı yapıya ROI (Regions of Interest) adı verilmektedir. Önemli olan görüntü ön planda, daha önemsiz olan görüntü ise arka planda tutulmaktadır. Görüntünün sıkıştırma oranı, ayrılan katmanlarda değişiklik göstermektedir. Önemsiz olan arka plan görüntüsü fazla, önemli olan ön plan görüntüsü daha az sıkıştırılmaktadır. Bu şekilde görüntünün iletiminde hem görüntünün kalitesi korunduğu hem de sıkıştırma oranı arttırılarak görüntünün iletiminde enerji verimliliği sağlandığı sonucuna varılmıştır.

KMAA'larda ilk olarak kullanılan görüntü sıkıştırma algoritması olan Embedded Zero Tree (EZW) [52], A.Said ve ark. tarafından kod çözmede ve kodlamada daha hızlı olmasını sağlamak amacı ile değiştirilerek yeni bir görüntü sıkıştırma tekniği elde edilmiştir. Bu tekniğe Hiyerarşik Ağaçlarda Küme Bölümleme (Set Partitioning in Hierarchical Trees - SPIHT) adı verilmiştir [53]. SPIHT, görüntünün piksel verilerini ağaç yapısına dönüştürerek birbiriyle yüksek derecede ilişkili olan pikselleri bir küme altına toplamaktadır. Daha sonra sıfır değerli ağaçları tek bir değerle kodlama işlemini yapmaktadır. Dalgacık dönüşümünün doğasındaki bu özellik sayesinde hiyerarşik ağaçlarda kümeleme bölümleme algoritması ile birlikte görüntülerde yüksek sıkıştırma oranı elde edilmiştir.

Discrete Wavelet Transform (DWT), dalgacık fonksiyonları ile görüntü sıkıştırma dönüşüm sınıfı sıkıştırma yöntemleri arasında yer alır. Bu yöntem, dalgacık dönüşüm alanında korunan enformasyon ölçekteki fazlalıkların azalması esasına dayanır. İlk olarak görüntüye dalgacık dönüşümü uygulanır ve katsayıların bir kısmı dönüştürülmüş görüntüden atılır. Bu noktada hangi katsayıların korunup hangilerinin atılacağı

istatistiksel hesaplara, görüntünün işaret enerji konumu ve tecrübe edilmiş eşik değeri gibi uygulamaya bağlı olarak seçilebilir. Kodlama geriye kalan katsayılara uygulanır. Sıkıştırılmış görüntü çözülen katsayılardan tekrar oluşturulur ve ters dönüşüm uygulanarak orijinal görüntü yeniden elde edilir [54]. Çalışmada, paket hata oranını azaltmak için etkili bir kuyruk kontrol stratejisini içermektedir.

[55]'de geliştirilen algoritma sayesinde KMAA'lar kullanılarak iletilen JPEG2000 formatlı görüntülerin kalitesi bozulmadan daha az enerji ile iletileceği düşünülmüştür. Geliştirilen bu algorithmada görüntü içerisinde yer alan gereksiz bitler atılarak görüntünün boyutu azaltılmış ve dolayısıyla iletilen daha az miktardaki verinin daha az enerji harcayacağı savunulmuştur. Ancak, KMAA'larda kullanılan algılayıcı düğümlerin donanım olarak JPEG2000 formatındaki görüntünün sıkıştırılmasına uygun olmadığı anlaşılmış ve beklenenden daha çok enerji harcandığı anlaşılmıştır. [56]'da Mulugeta ve ark. algılayıcı ağlar için güvenli bir yönlendirme protokolü önermişlerdir. Bu çalışmada görüntünün sıkıştırılması, işlenmesi ve iletimi sırasında enerjinin minimum seviyede kullanılması amaçlanmıştır. [57]'de KMAA'larda enerji verimli Yapay Bağışıklık Sistemleri (YBS) tabanlı görüntü tanıma uygulaması geliştirilmiştir. KMAA'larda toplam enerji tüketimi ve algılama kapsamı algılayıcı düğümlerin aktif veya uyku durumlarına göre değişiklik gösterdiği ifade edilmiştir. Çalışmada büyük boy resimler üzerinde veri işleme gerçekleştirilirken minimum enerji maliyeti için uygunluk gösterdiği savunulmuştur.

[58]'de önerilen algoritma ile JPEG'e göre veri miktarında 8,39 kat azalma başarılmıştır. Algoritma, boyut azaltmada kullanılan danışmansız bir makine öğrenmesi algoritması olan SOM (Self Organizing Map)'dur. Algılanan görüntü önerilen algoritma ile kodlanarak baz istasyonuna bir yönlendirme algoritması aracılığıyla gönderilir ve baz istasyonu tarafında, kodlanmış verinin kodu çözülerek orijinal görüntü yeniden elde edilir. SOM çok büyük bir girdi kümesinin, o kümeyi iyi yansıtan bir alt kümesini bularak orijinal veri setine iyi bir yakınsama sağlar. Önerilen yöntemde görüntü verisini sıkıştırmak için bu vektör niceleme özelliği kullanılır. SOM, kod çizelgesi tasarımında kullanılmıştır. Üretilen kod çizelgeleri her bir düğümden ve baz istasyonunda saklanmaktadır. Görüntünün iletilmesi ve baz istasyonunda yeniden oluşturulmasında iki işlem gerçekleşir; bunlar kodlama ve kod çözmedir. Kodlamada, gelen görüntüyü bloklara ayrılır ve ona karşılık gelen kod kelimesi kod çizelgesinden aranarak elde

edilir. Sonrasında da baz istasyonuna iletilir. Kod çözmede ise aktarılan kod kelimesi kod vektörüne dönüştürülür [58].

Ayrık Kosinüs Dönüşüm (Discrete Cosine Transform – DCT) KMAA’larda sıklıkla kullanılan görüntü sıkıştırma tekniğidir. DCT tabanlı görüntü sıkıştırma teknikleri yeterli sıkıştırma verimliliği sağlamaktadır. Kodlama, küçük bireysel görüntü bloklarına ayrılarak düşük bellek uygulanmalarında da kullanılması sağlanır [59].

Uygulama katmandaki çalışmaların yetersizliği ve gerçekleştirilecek iyileştirmelerin diğer katmanlardakilerle birleştirilerek daha fazla enerji duyarlı yaklaşımların geliştirilmesi fikri, bu tez çalışmasında uygulama katmanında çalışacak enerji duyarlı bir yöntem geliştirmeye yönelmiştir. Bu sayede, enerji kısıtının çok önemli bir problem olduğu multimedya algılayıcı ağlarda, ağ ömrü transfer edilen veri boyutunun minimize edilmesi ve ağ durumuna (veri gönderme sıklığı, düğümler arası mesafe ve düğüm sayısı) en uygun sıkıştırma algoritmasının seçilmesi ile uzatılacaktır.

1.3 ÇALIŞMANIN AMACI VE ÖNERİLEN ÇÖZÜM YÖNTEMLERİ

KAA’lara göre daha fazla boyuta sahip verileri transfer edebilen KMAA’lar, daha fazla enerjiye ve bant genişliğine ihtiyaç duymaktadır. KMAA uygulamalarında genellikle mobil algılayıcı düğümler tercih edildiği için algılayıcı düğümlerin enerji ihtiyaçlarını arazi şartlarında karşılamaları oldukça zordur. Bu yüzden algılayıcı düğümlerin var olan enerjilerini en etkin şekilde kullanmaları gerekmektedir.

Algılayıcı düğümlerin birbirlerine ve baz istasyonuna ilettikleri verinin büyüklüğü, enerjilerinin harcanmasında büyük bir etken olmaktadır. Bir ağda enerji tüketimin çoğu, verinin düğümler arasında iletimi sırasında gerçekleşir [10]. Bu sırada harcanan enerji verinin boyutu ile doğru orantılıdır. Literatürde KMAA’larda bu ve benzeri enerji problemlerini aşmak amacıyla, taşınan veriyi en kısa yoldan hedefe ulaştırmak, düğümleri görevlerine göre uygun pozisyonlara yerleştirmek ve düğümleri uyutmak gibi çözümler geliştirilmiştir. Son zamanlarda, geliştirilen bu çözümlerin daha da ilerisine gidilerek görüntüler (video ya da ses verileri) yakalandıktan sonra sıkıştırılıp iletilir. Böylece, daha düşük boyuttaki veri belirlenen uygun hat üzerinde taşınır. Bu da, en uygun yönlendirme şeması, en uygun düğüm dizilimi ya da boştaki düğümlerin uyutulması çözümlerinin çok daha üstünde bir enerji verimli yaklaşım sunar.

Bu tez çalışmasının amacı, KMAA üzerindeki algılayıcı düğümlerin birbirlerine veya baz istasyonuna görüntü aktarımı sağlarken harcadıkları enerjiyi en aza indirerek, algılayıcı düğümler için en önemli kaynak olan batarya ömrünü etkin bir biçimde kullanılmasını sağlamaktır. Bu amaç çerçevesinde algılayıcı düğümler, görüntü sıkıştırma algoritmaları sayesinde elde ettikleri görüntüyü, öncelikle orijinal boyut değerinden daha az değere ulaştıracak daha sonra diğer algılayıcı düğüme veya baz istasyonuna ileteceklerdir. Bu sayede görüntüyü ileten algılayıcı düğümün, boyutu düşürülmüş görüntüyü iletme enerjisi ve görüntüyü alan diğer algılayıcı düğümün görüntüyü alma enerjisi daha az olacaktır. Ayrıca, algılayıcı düğümlerin kullanacakları görüntü sıkıştırma algoritmalarının da oldukça az enerji harcamaları gerekmektedir. Sınırlı işlemciye ve hesaplama gücüne sahip KMAA'lar için kullanılacak olan görüntü sıkıştırma algoritmasının işlem karmaşası ve işlem adımının az olması gerekmektedir. KMAA'larda gerçek zamanlı görüntü aktarımı için geliştirilecek olan uygulamalarda ise kullanılacak görüntü sıkıştırma algoritmasının işlem hızının önemi vardır. Ek olarak, en verimli görüntü sıkıştırma algoritması ile sıkıştırılan görüntünün kablosuz iletişimde kabul edilebilir PSNR (Peak Signal to Noise Ratio) kayıp değeri kontrol edilerek aktarılan görüntünün kalitesi kontrol altında tutulacaktır.

Bu tez çalışmasında, KMAA'lar için kullanılan mevcut popüler sıkıştırma yöntemlerinden belli parametrelere göre enerji verimliliği bakımından en iyi olan algoritma seçilerek iletilecek veri boyutunun küçültülmesi sağlanmıştır. Bu parametreler düğüm sayısı, düğümler arası mesafe ve veri gönderme sıklığıdır. Ayrıca, iletilen görüntünün PSNR değer kaybı kablosuz iletişimde kabul edilebilir değer aralığını koruması sağlanmıştır. Önerilen algoritma, bu değerlendirmeleri yaparak KMAA'larda kullanılan görüntü sıkıştırma algoritmalarından daha etkili bir çözüm sunmaktadır. Bu algoritmaya, Kablosuz Multimedya Algılayıcı Ağlar için Görüntü Sıkıştırmaya Dayalı Yeni Bir Enerji Duyarlı Uygulama Katmanı Algoritması (GSD-EDUKA) olarak isimlendirilmiştir. Çalışmada KMAA'lardaki enerji problemini çözmeyi amaçlarken bir uygulama katmanı protokolü geliştirmeye motive eden sebepler şunlardır:

1. Gönderilen verinin boyutunu küçültmek, taşınan veri boyutunu azaltmak demektir ve bir ağda en çok enerji harcayan işlemin daha az enerji harcar hale getirilmesidir. KMAA için gönderilecek veri görüntü olacağından görüntü boyutunun azaltılması görüntünün sıkıştırılmasını ifade eder.

2. KMAA için diğer katmanlarda gerçekleştirilmiş enerji duyarlı protokoller mevcuttur. Ancak, uygulama katmanındaki çalışmalar henüz olgunlaşmamıştır.
3. KMAA için önerilen çok sayıda enerji duyarlı protokol daha çok diğer katmanlar için, özellikle de yönlendirme katmanı için önerilmiştir. Uygulama katmanında az sayıda çözüm üretilmiş ve bu konuda etkili bir iyileştirmeye ihtiyaç duyulmaktadır.

1.4 TEZ ORGANİZASYONU

Bu tez çalışması aşağıda özetleri verilen bölümler şeklinde organize edilmiştir:

1. Bölümde, KAA'lara kısaca değinilerek, KMAA'lar arasındaki farklar ele alınmıştır. KMAA'ların kullanım alanlarından bahsedilerek, literatürde KMAA'lar için enerji verimliliği ile ilgili yapılan çalışmalara değinilmiştir. Son olarak, bu tez çalışmasının amacı olan KMAA'lar için enerji verimli GSD-EDUKA'nın geliştirilme aşamasından kısaca bahsedilmiş ve katkıları sunulmuştur.

2. Bölümde, literatürde yer alan görüntü sıkıştırma teknikleri üzerinde durulmuş ve bu teknikler sınıflandırılmıştır. Daha sonra KMAA'larda kullanılan görüntü sıkıştırma teknikleri ele alınarak, enerji verimliliği bakımından hangi tekniğin daha iyi sonuç verdiği çıkarılan çizelge aracılığı ile bulunmuştur. Öne çıkan görüntü sıkıştırma algoritmalarının performanslarını ölçmek için kullanılacak benzetim aracına uyum sağlayan programlama dili ile kodlanmıştır. Bu sayede her bir görüntü sıkıştırma algoritmasının sıkıştırma performansı, sıkıştırma süresi ve açma süresi elde edilmiştir. Daha sonra bu değerler karşılaştırılmıştır.

3. Bölümde, öncelikle geliştirilen sistemin adımları kısaca anlatılarak kodlanan GSD-EDUKA'nın adımları ve parametreleri çıkarılmıştır. Belirlenen görüntü sıkıştırma algoritmaları için GSD-EDUKA'nın parametreleri ile çıkarımlar yapılarak, hangi algoritmanın hangi parametrede daha verimli çalışacağı sonucu elde edilmiştir. Elde edilen çıkarımların sayısı "karar ağacı yöntemi" kullanılarak daha aza indirilmiş ve bu sayede geliştirilen GSD-EDUKA'nın hızı arttırılmış ve işlem yükü azaltılmıştır. Daha sonra "bulanık mantık yöntemi" kullanılarak GSD-EDUKA'nın belirlenen parametrelerle kullanacağı görüntü sıkıştırma algoritmasına karar vermesi sağlanmıştır.

4. Bölümde, KMAA'lardaki algılayıcı düğümlerin enerji tüketim değerleri üzerinde

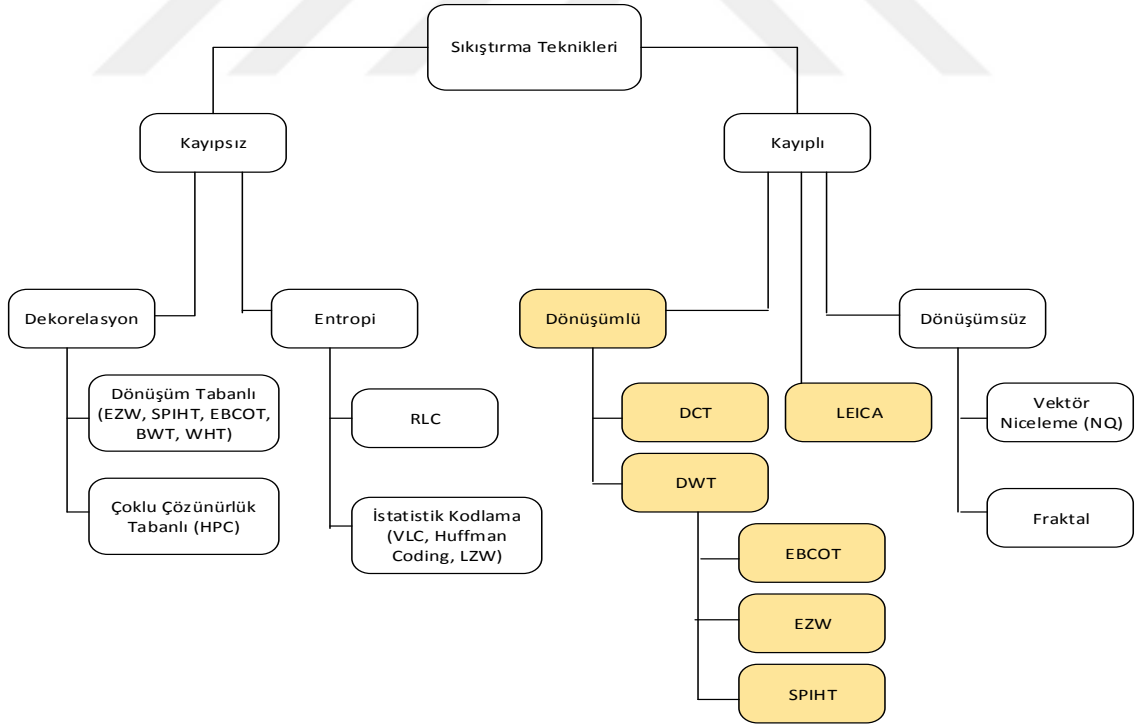
durulmuştur. Geliştirilen GSD-EDUKA'nın harcadığı enerji ifade edilerek, fazladan kullanılan modüllerin harcadığı enerjiden ve algılayıcı düğümlerin var olan enerjisinden sağladığı tasarrufa değinilmiştir.

5. Bölümde, geliştirilen GSD-EDUKA'nın benzetim aracı olan Network Simulator - 2 (NS - 2)'de performans değerlendirilmesi yapılmıştır. Bu değerlendirme için öncelikle belirlenen görüntü sıkıştırma algoritmaları ayrı ayrı kullanılarak ve GSD-EDUKA kullanılarak benzetim gerçekleştirilmiştir. Daha sonra GSD-EDUKA ile hiçbir görüntü sıkıştırma algoritması kullanmadan gerçekleşen sistem karşılaştırılmıştır.

6. Bölümde, yapılan benzetimlerin ve karşılaştırmaların değerlendirilmesi yapılarak geliştirilen algoritmanın üstün kısımları vurgulanmıştır. Bu çalışma ışığında yapılabilecek sonraki çalışmalarla ilgili önerilerde bulunulmuştur.

2. KABLOSUZ MULTİMEDYA ALGILAYICI AĞLARDA GÖRÜNTÜ SIKIŞTIRMA TEKNİKLERİ

KMAA’larda görüntü sıkıştırma teknikleri kayıplı ve kayıpsız olmak üzere ikiye ayrılır. Görüntü üzerinde gereksiz bitleri çıkarıldığında bilgi kaybı olmazsa buna kayıpsız sıkıştırma denir. Eğer bilgi kaybı olursa bu duruma da kayıplı sıkıştırma tekniği denir. Kayıplı görüntü sıkıştırma teknikleri 1/50’ye kadar yüksek sıkıştırma oranına sahiptir. Ancak kayıpsız görüntü sıkıştırma teknikleri, orijinal görüntünün boyutunu en fazla 1/3 veya 1/4 oranında sıkıştırabilirler. Bu oran kayıplı görüntü sıkıştırma tekniklerine göre oldukça azdır. Kayıplı tekniklerin kayıpsız tekniklere göre kodlama ve kod çözme sürelerinin az olması, sıkıştırma oranı ve enerji kısıtlı uygulamalarda kullanabilirliği gibi önemli avantajlar sağlar [60]. KMAA’larda kullanılan görüntü sıkıştırma teknikleri Şekil 2.1’de gösterilmektedir.



Şekil 2.1. KMAA’larda kullanılan görüntü sıkıştırma teknikleri [60].

2.1 KAYIPSIZ GÖRÜNTÜ SIKIŞTIRMA TEKNİKLERİ

Kayıpsız görüntü sıkıştırma teknikleri iki aşamalıdır, bunlar dekorelasyon ve entropi kodlamadır [61].

Dekorelasyon, pikseller arasındaki fazlalıkları kaldırarak kayıpsız görüntü sıkıştırma gerçekleştiren tekniklerdendir. Bu teknikler üç kategoride sınıflandırılır; tahmin tabanlı teknikler [62], dönüşüm tabanlı teknikler ve çoklu çözünürlük tabanlı tekniklerdir [63]. Entropi kodlama ise görüntüdeki fazlalıkları kaldırır. Bu teknik RLC (Run Length Coding) ve istatistiksel kodlama tabanlıdır [64], [65]. Ayrıca, entropi JPEG gibi bazı kayıplı sıkıştırma algoritmaları ile de kullanılır.

Kayıpsız sıkıştırılmış görüntülerin boyutu kayıplılara göre oldukça büyüktür bu sebepten algılayıcı düğümler üzerinde görüntü transfer ederken daha fazla güce ve yüksek band genişliğine ihtiyaç duyar. Dolayısıyla kayıpsız sıkıştırma teknikleri KMAA'larda tercih edilmez [60].

2.2 KAYIPLI GÖRÜNTÜ SIKIŞTIRMA TEKNİKLERİ

Kayıplı sıkıştırma teknikleri, kayıpsız sıkıştırma tekniklerine göre yüksek sıkıştırma oranına sahiptir. Kayıplı sıkıştırmada, sıkıştırılmış görüntü genellikle orijinal görüntü ile aynı değildir. Bu nedenle, bozulma ölçütü kullanılır. Bozulma ölçütü, sıkıştırılmış görüntünün orijinal görüntüye ne kadar yakın bir yaklaşımda olduğunu belirten matematiksel bir büyüklüktür. Görüntü sıkıştırmada en sık kullanılan bozulma ölçütleri PSNR (Peak Signal to Noise Ratio) ve MSE (Mean Square Error)'dir [66], [67]. Kablosuz iletişimde, kabul edilebilir PSNR kayıp değeri yaklaşık 20 ile 25 dB arasındadır [68], [69].

PSNR ve MSE'yi inceleyecek olursak;

$$PSNR = 10 \log_{10} \left(\frac{I_{max}^2}{MSE} \right) dB \quad (2.1)$$

I_{max} değeri, alınan görüntünün en büyük griton değeridir. PSNR karşılaştırmalarında bu değer genel olarak 255 olarak alınır [70]. MSE değerinin hesaplanmasında Denklem (2.2) kullanılır.

$$MSE = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (Y_{i,j} - S_{i,j})^2 \quad (2.2)$$

Burada M görüntünün yatay boyutundaki, N ise dikey boyutundaki piksel sayısıdır. $S_{i,j}$ orijinal görüntü pikselleri, $Y_{i,j}$ ise süzölmüş görüntü pikselleridir [71].

Çalışmanın devamında kayıplı görüntü sıkıştırma tekniklerinden DCT, DWT, EZW, SPIHT, LEICA ve EBCOT üzerinde durulmuş ve bu sıkıştırma tekniklerinin değerlendirilmeleri yapılmıştır.

2.2.1 Ayırık Kosinüs Dönüşüm Tabanlı (Discrete Cosine Transform – DCT) Sıkıştırma Teknikleri

Ayrık Kosinüs Dönüşümü (DCT) en yaygın kullanılan dönüşüm kodlama tekniğidir. DCT tabanlı görüntü sıkıştırma, görüntü kaynağı ilk 8 x 8 piksel boyutu ile görüntülerin bloklara böler ve 64 elemanlı katsayı bloğu oluşturur. Her bir görüntü bloğu birbirinden bağımsız kodlanır [72]. 2 boyutlu DCT'nin eşitliği Denklem (2.3)'teki gibidir;

$$F(u, v) = \frac{1}{4} C(u)C(v) \sum_{x=0}^{k-1} \sum_{y=0}^{k-1} f(x, y) \cos\left(\frac{(2x+1)u\pi}{2k}\right) \cos\left(\frac{(2y+1)v\pi}{2k}\right) \quad (2.3)$$

$$C(z) = \begin{cases} \frac{1}{\sqrt{2}} & \text{eğer } z = 0 \\ 1 & \text{diğer durumlarda} \end{cases} \quad (2.4)$$

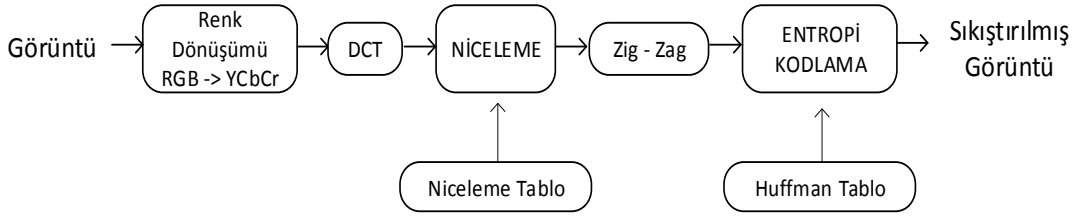
Bu şekilde matrisler oluşturulur. Matrisin her elemanından 128 çıkarılarak P matrisi oluşturulur. Daha sonra en sık kullanılan A matrisi ve bunun tersi ile bu P matrisi çarpılarak DCT matrisi bulunur.

$$F(k \times k) = A(k \times k)P(k \times k)A^T(k \times k) \quad (2.5)$$

Daha sonra niceleme işlemi yapılır. Bu işlemi sayesinde görüntünün ne kadar sıkıştırılacağı isteğe bağlı olarak belirlenmiş olur. Niceleme formülü aşağıdaki gibidir;

$$C_{i,j} = \text{round}\left(\frac{D_{i,j}}{Q_{i,j}}\right) \quad (2.6)$$

DCT tabanlı bilinen en iyi sıkıştırma şeması JPEG'dir [73]. Çok sık kullanılan bir görüntü sıkıştırma standardı olan JPEG, ayrık kosinüs tabanlı görüntü sıkıştırma tekniğini kullanır. Kayıplı JPEG sıkıştırma şeması Şekil 2.2'de verilmiştir.



Şekil 2.2. Kayıplı JPEG sıkıştırma şeması [74].

DCT tabanlı görüntü sıkıştırma teknikleri yeterli sıkıştırma verimliliği sağlamaktadır. Kodlama küçük bireysel görüntü bloklarına ayrılarak düşük bellek uygulamalarında da kullanılması sağlanır [75].

2.2.2 Ayrık Dalgacık Dönüşümü (Discrete Wavelet Transform – DWT) Tabanlı Sıkıştırma Teknikleri

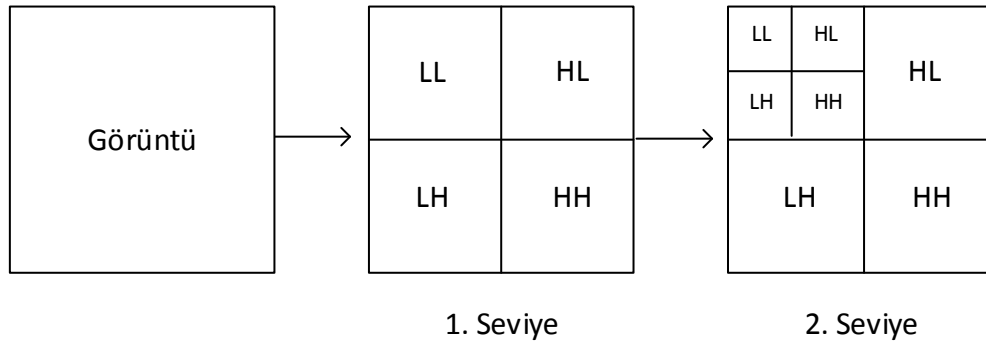
Dalgacık fonksiyonları ile görüntü sıkıştırma, dönüşüm sınıfı sıkıştırma yöntemleri arasında yer alır. Bu yöntem, dalgacık dönüşüm alanında korunan enformasyon ölçekteki fazlalıkların azalması esasına dayanır. İlk olarak görüntüye dalgacık dönüşümü uygulanır ve katsayıların bir kısmı dönüştürülmüş görüntüden atılır. Bu noktada hangi katsayıların korunup hangilerinin atılacağı istatistiksel hesaplara, görüntünün işaret enerji konumu ve tecrübe edilmiş eşik değeri gibi uygulamaya bağlı olarak seçilebilir. Kodlama geriye kalan katsayılara uygulanır. Sıkıştırılmış görüntü çözülen katsayılardan tekrar oluşturulur ve ters dönüşüm uygulanarak orijinal görüntü yeniden elde edilir [54].

Dalgacık analizde temel olan, değişik seviyelerdeki ayrıntılardan enformasyonun tekrar elde edilmesi fikridir. Dalgacık dönüşüm yüksek ve alçak geçiren süzgeçlerden oluşan bir ağaç yapısı şeklinde düşünülebilir. Alçak geçiren süzgeç L_n işareti veya veri dizisindeki enformasyon miktarını azaltırken, yüksek geçiren süzgeç H_n ise kaybolan bu enformasyonu saklar. Dalgacık dönüşümü ile görüntü ortalama şeklinde alt bantlara ayrıştırılır [73].

Genel olarak görüntünün enerji yoğunluğunun büyük bir kısmı alçak frekans bölgelerinde yoğunlaşmıştır. Sıradan bir görüntünün spektral bileşenleri frekans

yükseldikçe azalır. Görüntünün bu özelliği aynen kendisini DWT’de de gösterir. Herhangi bir görüntünün DWT ayrıştırmasını yapmış olduğumuzu varsayalım, bu durumda alçak düzeyler n yüksek frekans alt bantlarına denk düşer. Örneğin, birinci düzey en yüksek frekans alt bandını gösterirken aynı zamanda çözünürlüğün en yüksek olduğu düzeyi de gösterir. Buna bağlı olarak n düzey en alçak frekans alt bandını gösterirken çözünürlüğün en kaba seviyesine de denk düşmektedir. Bu nedenle yüksek düzeylerden alçak düzeylere ilerlerken (düşük çözünürlükten yüksek çözünürlüğe doğru), alt bantların enerji içeriğini azaltmakta olduğu görülür. Aynı yaklaşımla belirli düzeydeki alt bantın piksel genlik değerinin düşük olduğu gözlemlendiğinde, daha düşük düzeydeki alt banttaki piksel değerlerinin daha düşük olması beklenir [54].

Dalgacık tabanlı dönüşümler dalgacık olarak adlandırılan bir dizi basit fonksiyon kullanarak bir sinyali zaman ve frekans kapsamında iyi bir çözünürlükte sunarlar. Görüntü sıkıştırmada kullanılan iki boyutlu dalgacıklar ayrılabilir fonksiyonlardır. Bunların uygulamaları öncelikle L ve H alt bantlarını üretebilmek için satırlar üzerine alçak geçiren filtre uygulayarak daha sonrada LL, LH, HL ve HH alt bantlarını üretebilmek için sütunlara yüksek geçen filtre uygulayarak gerçekleştirilir. Daha sonra ikinci aşamada, bu dört alt bantın her biri tekrar $LL2, LH2, HL2$ ve $HH2$ şeklinde alt dört banda ayrılır ve böyle devam eder. Şekil 2.3’te LL alt bandı gösterilmiştir [76].



Şekil 2.3. 2B DWT’nin ayrıştırılmasının iki seviyesi [73].

2.2.3 Gömülü Sıfır Ağaç Dalgacık Tabanlı (Embedded Zerotree Wavelet – EZW) Görüntü Sıkıştırma Teknikleri

Gömülü Sıfır Ağaç Dalgacık (EZW) basit ve etkili bir görüntü sıkıştırma algoritmasıdır ve özelliği bit dizisindeki bitlerin önemlerine göre üretiliyor olmasıdır [52]. Sonuçta tamamen gömülü bir kod ortaya çıkar, bu gömülü kod görüntüyü boş bir görüntüden ayıran ikili (binary) kararların bir dizisidir. Gömülü kodlama algoritması kullanarak bir

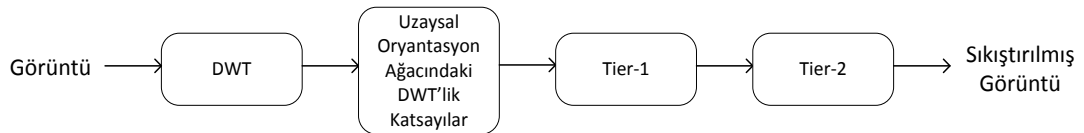
kodlayıcı, kodlamayı herhangi bir noktada sonlandırabilir ve böylece hedef oran ya da hedef bozulma ölçütü tam olarak sağlanır. Ayrıca bir bit dizisi verilmişken, kod çözücü kod çözme işini bit dizisinde herhangi bir noktada durdurabilir ve yine kırılan bit dizisine karşılık gelen bit oranında kodlanan görüntü ile aynı görüntü üretilir. Tamamen gömülü bit dizisi üretmeye ek olarak, EZW sanal olarak bilinen pek çok sıkıştırma algoritmalarıyla rekabet edebilecek sıkıştırma sonuçları üretir. Bu performans, eğitimi önceden yüklenen çizelgeler ya da kod kitapları ve kaynak resmin önceden bilinmesini gerektirmeyen bir teknik ile başılır [52].

EZW algoritması dört kavrama dayalıdır [52]:

- Ayrık dalgacık dönüşümü,
- Kendine benzeyen görüntülerden faydalanarak önemli bilginin yokluğunun tahmin edilmesi,
- Entropi kodlu ardışık tahmin ölçeklemesi ve
- Adaptif aritmetik kodlama ile başılan “evrensel” kayıpsız veri sıkıştırmasıdır.

2.2.4 Hiyerarşik Ağaçlarda Küme Bölümleme (Set Partitioning in Hierarchical Trees – SPIHT)

Hiyerarşik Ağaçlarda Küme Bölümleme (SPIHT), EZW algoritmasının gelişmiş bir versiyonudur ve güçlü bir dalgacık tabanlı sıkıştırma algoritmasıdır [77]. SPIHT ile görüntü sıkıştırmada görüntüye bölütlenmeden önce dalgacık dönüşümü uygulanır. Dalgacık dönüşümü ile elde edilen ayrıntı alt bantları görüntü enerjisinin yüzde olarak çok az bir bölümünü oluşturan küçük değerli katsayılarla sahiptir. Bu küçük değerli katsayıların seçilecek eşik değeri ile sıfır değerine nicelenmesi görüntüde büyük bir kayba neden olmamaktadır. Bu işlemle oluşturulan görüntü büyük oranda sıfır değerine sahip piksellerden oluşmaktadır. SPIHT’in kodlama adımları Şekil 2.4’teki gibidir;



Şekil 2.4. SPIHT kodlama adımları [76].

Dalgacık dönüşümün sağladığı bu özellik hiyerarşik ağaçlarda küme bölümleme algoritmasına temel teşkil etmektedir. Görüntülerin DWT’i dengesiz bir karakteristik gösterir. Diğer bir deyişle ayrıntılı görüntülerin piksel değerleri büyük bir oranda sıfır

veya sıfıra yakın bir değerden oluşur. Bu algorithmada ayırık dalgacık dönüşümü katsayıları uzamsal yönelimli ağaçlar (Spatial Orientation Tree) şeklinde organize edilir. Bu ağaçların oluşturulmasının sebebi birbirleriyle yüksek derecede ilişkili olan piksellerin bir küme altında toplanmasıdır. Bu şekilde küme yapılandırma ile bir seviyedeki katsayı ile diğer seviyedeki benzerlik oranı arttırılmış olmaktadır. Seçilen algorithmada da temel amaç birbirleriyle yüksek derecede ilişkili olan, sıfır değerli ağaçları tek bir değerle kodlamaktır. Dalgacık dönüşümünün doğasındaki bu özellik sayesinde hiyerarşik ağaçlarda kümeleme bölümlleme algoritması ile birlikte görüntülerde yüksek sıkıştırma oranı elde edilmiştir [77]. Sıkıştırılmış bit akımı oluşturulması için Tier-1 ve Tier-2 olmak üzere iki aşamalı katman oluşturulmuştur. Tier-1 ve Tier-2 işlemi aritmetik kodlama ve bağlam düzenlenmesinden sorumludur [78].

2.2.5 Düşük Enerjili Görüntü Sıkıştırma Algoritması (Low Energy Image Compression Algorithm - LEICA)

Görüntü ilgili bölgeler (Regions of Interest – ROI) ve arka plan olarak ikiye bölünür. ROI görünülmesi istenen bölgelerdir. Görüntü kalitesini korumak için daha düşük sıkıştırma oranı ile sıkıştırılır ve görüntünün tamamı işlenir. Arka plan daha az önemlidir, bu yüzden bu kısım daha fazla sıkıştırılabilir. Bu şekilde enerji daha fazla korunmuş olmaktadır [51].

İzlenen nesne görünmediği zaman algılayıcı kamera arka planı tutar ve belleğe kaydeder. Skaler algılayıcı kamera ise, nesneyi izlemek için kullanılır. Nesne görünmüyorsa, kamera pasif durumunda kalır. Aksi halde, sayısal algılayıcı düğüm tarafından tetiklenen kamera düğümü f ile belirtilen görüntüyü yakalar. Yakalanan görüntü (f) ve arka plan görüntüsü (b) arasındaki her bir bloğun farkı aşağıdaki formülle hesaplanır.

$$D_{m,n} = \frac{1}{64} \sum_{i=1}^8 \sum_{j=1}^8 |f(8m+i, 8n+j) - b(8m+i, 8n+j)| \quad (2.7)$$

$$m = 0, 1, \dots, M/8 - 1 ; n = 0, 1, \dots, N/8 - 1$$

Her bloktaki ρ değeri aşağıdaki şekilde hesaplanır [22].

$$\rho_{m,n} = \begin{cases} \rho_{max}, & \text{eğer } D_{m,n} > D_0 \\ \rho_{min}, & \text{diğer durumlarda} \end{cases} \quad (2.8)$$

ROI ve bloklar arasındaki eşik değeri olan D_0 'ın değerine, kamera düğümü ve izlenen sahne tarafından karar verilir. ρ_{max} 'ın değerine ROI'nın görüntü kalitesi karar verir. ρ_{min} 'nin değeri, düğümlerde kalan enerji ve piksellerin görüntü kalitesi ile belirlenir. ρ_{min} 'nin değeri ρ_{max} 'ın değerinden küçüktür.

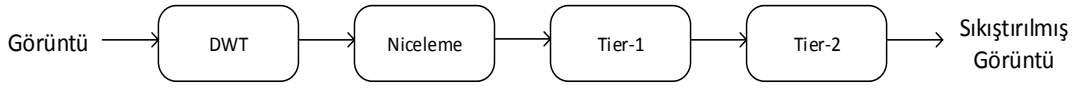
Yakalanan görüntü sıkıştırıldığı zaman, ρ_{min} 'nin değeri, her bir blok için seçilen DCT katsayılarının sayısını belirler. LEICA'nın diğer işlemleri JPEG ile aynıdır [51].

2.2.6 Optimize Kesme ile Gömülü Blok Kodlama (Embedded Block Coding with Optimized Truncation - EBCOT)

Optimize kesme ile gömülü blok kodlama (EBCOT) bir blok tabanlı kodlama algoritmasıdır. EBCOT'un temel fikri, alt bant olan LL, LH, HL ve HH 'leri kod blok denilen küçük bloklar haline dönüştürmektir. Her blok bağımsız kodlanmıştır [73]. EBCOT, bilgi kodlama ve bilgi düzenleme olarak iki bölüme ayrılır. JPEG2000 blok diyagramı Şekil 2.5'te gösterilmiştir. Gömülü kodlama blok bazlı olarak gerçekleştirilir. Popüler görüntü sıkıştırma standardı JPEG2000, EBCOT sıkıştırma tekniğini benimsemiştir [79].

Daha önceden yapılmış ölçeklenebilir görüntü sıkıştırma çalışmalarının bir türü olarak geliştirilmiştir. Sayısallaştırılıp kodlanabilen alt bant örneklerini üretebilmek için diğer algoritmalarda olduğu gibi dalgacık dönüşümünü kullanılmaktadır. Ölçeklenebilir sıkıştırma, gömülü alt kümeler içeren bit dizisi üretmeye karşılık gelmektedir. Bu alt kümelerin her biri çözünürlüğü azaltılmış ya da bozukluk oranı artmış orijinal görüntünün etkin olarak sıkıştırılmış halini gösterir. Bu düşünce ile bağlantılı olarak çözünürlük ölçeklenebilirliği ve SNR ölçeklenebilirliği terimleri yaygın olarak kullanılır. Eğer bir bit dizisinin her biri ardışık çözünürlük seviyesini gösterecek şekilde ayrı alt kümeler içeriyorsa bu bit dizisine çözünürlüğü ölçeklenebilir denir. Eğer bir bit dizisi belli kalite (SNR) seviyesindeki tüm alt bantları gösteren ayrı alt kümeler içeriyorsa bu bit dizisine de SNR'si ölçeklenebilir denir. Ölçeklenebilir sıkıştırmanın en önemli avantajı, hedef bit oranı ya da çözünürlüğün sıkıştırma anında bilinmesine ihtiyaç olmamasıdır. Diğer bir avantaj da, mevcut JPEG2000 sıkıştırma standartlarında

olduğu gibi görüntünün hedeflenen bit oranını başarmak için birden fazla kere sıkıştırılmasına gerek olmamasıdır [80].



Şekil 2.5. JPEG2000 blok diyagramı [76].

2.3 KMAA’LARDA KULLANILAN GÖRÜNTÜ SIKIŞTIRMA TEKNİKLERİNİN DEĞERLENDİRİLMESİ

Bu bölümde, yapılan literatür taramasına göre KMAA’larda kullanılan ve Bölüm 2.2’de açıklanan görüntü sıkıştırma tekniklerinin değerlendirilmesi yapılmıştır.

KMAA’ların kısıtlı gücü, sınırlı bant genişliği ve sınırlı belleği olduğundan dolayı, JPEG ve JPEG2000 gibi geleneksel görüntü sıkıştırma algoritmalarını kullanmaları çok verimli değildir. Bu algoritmaların KMAA’lar için en büyük sorunları algoritma boyutları, işlemci hızı ve bellek erişimidir [1], [81].

Mandelbrot ve ark. yaptıkları çalışmada JPEG’in yüksek işlem yüküne sahip olduğunu göstermişlerdir [82]. Öyle ki, derleme süresi 103,12 ms sürerek kodlama esnasında algılayıcı düğümlerin sahip oldukları enerjisinin çoğunu tüketmişlerdir.

Ghorbel ve ark. KMAA’larda görüntü sıkıştırma için performans analizi gerçekleştirmişlerdir [83]. DCT ve DWT üzerinde yaptıkları analizin sonucu olarak görüntü kalitesi ve enerji tüketimi açısından DCT’nin DWT’den daha iyi olduğunu göstermişlerdir. Sistemde Alkali AA-NiMH 2850 mAh pil kullanılmış, DCT için bu pil 281,34 saat kullanılırken DWT için 245,20 saat kullanılmıştır. Ayrıca çalışmada DWT’nin, DCT’ye göre düşük hafıza uygulamalarına sahip olmasına rağmen KMAA’lar için iyi bir seçenek olmadığı, kodlayıcının karmaşası sebebiyle tüm enerjinin yaklaşık %60’ını tükettiği sonucuna varmışlardır.

Nasri ve ark. hesaplama ve iletişim enerjilerini en aza indiren uyarlayıcı SHPS (Skipped High Passsub Band - Atlamalı Yüksek Geçiren Alt Bant) isminde bir görüntü sıkıştırma yapısı önermişlerdir [84]. Bu görüntü sıkıştırma yapısı, en önemsiz olan alt bandı işleme katmayarak ve işlem yükünü azaltarak enerjinin korunmasını sağlamayı hedeflemiştir. Bunun üzerine EHPF (Elimination High Pass Filter) [85] isimli ek bir teknik daha

geliştirilmiştir. Bu teknikle görüntüye dikey doğrultuda yüksek geçiren filtre uygulanmayıp sadece alçak geçiren filtre uygulanmış ve dolayısıyla, yalnızca görüntü hakkında daha önemli bilgi içeren *LL* ve *LH* alt bantları üretilmiştir. Aslında EHPF yönteminin görüntüde daha az bozulmaya sebep olarak enerjiden bir miktar tasarruf etmesini sağlamışlardır. Fakat SHPS görüntü kalitesinde daha fazla bozulmaya sebep olsa da daha fazla miktarda enerjiden tasarruf edilmesini sağlamıştır. Önerilen bu iki teknik AWICAO (Adaptive Wavelet Image Compression And Organization Algorithm) olarak isimlendirilen uyarlayıcı bir sıkıştırma tekniğini oluşturur. En etkin olan tekniğin seçimi, enerji tüketimi ve görüntü kalitesinin avantaj ve dezavantajlarına bağlı olarak yapılır.

EBCOT, sıkıştırma kalitesi en yüksek performansa sahip görüntü sıkıştırma algoritmasıdır. Bu algortmada Tier-1 işlemi aritmetik kodlama ve bağlam düzenlenmesinden sorumludur. Bu işlem esnasında EBCOT'ta kodlayıcı, algılayıcı düğümün gücünü çok tüketir. Bu yüzden enerji tüketimi fazladır, hesap karmaşıklığından işlem süresi de uzundur ve dolayısıyla hafıza gereksinimi de vardır. Bu yüzden EBCOT, KMAA'lar için çok kullanışlı değildir [80]. Tier-1 işlemi DWT adımından sonra gelen kodlama zamanının %70'den fazlasını tüketmektedir [73], [76].

Fraktal Görüntü Sıkıştırma (FIC) yüksek sıkıştırma oranına sahip, hızlı çözümleme yapabilen ve sıkıştırılan görüntünün çözünürlüğüne müdahale edilebilen yeni nesil kodlama teknolojilerinden biridir [86]. Jackson ve ark. [87] Fraktal ile JPEG'i karşılaştırmışlardır. Yapılan çalışmalar sonucunda, Fraktal'ın daha büyük sıkıştırma oranına sahip olduğu anlaşılmıştır. Ancak, kodlama yaparken çok fazla enerji tüketerek KMAA'ların yaşam süresini aza indirdiği için çok kullanışlı olmadığı sonucuna varılmıştır.

Köse ve ark. [88] SPIHT ve JPEG2000 sıkıştırma tekniklerini 3 boyutlu resimler kullanarak karşılaştırmışlardır. SPIHT tekniğinin daha iyi sonuç verdiğini yaptıkları benzetim ile göstermişlerdir. Ayrıca, Chong ve ark. [89] KMAA'larda çoklu görüntü sıkıştırmak için SPIHT'i kullanmışlardır.

Fu-Xiao ve ark. [90] KMAA'larda görüntü bozulmasını en aza indirmek için SPIHT'i üç aşamada kullanmışlardır. Birinci aşama görüntü üzerine dalgacık dönüşümü uygulanmış, ikinci aşama SPIHT'in dalgacık katsayıları hesaplanmış, son adımda ise

SPIHT için bağlam tabanlı aritmetik kodlama yapılmıştır. Bu işlemler sonucunda KMAA'larda görüntü bozulması en aza indirilerek, SPIHT'in fayda sağladığı belirlenmiştir.

Ma ve ark. SPIHT'in bir versiyonunu geliştirmişler ve bu versiyonda farklı yöntemler kullanılarak görüntünün PSNR değerlerini, aritmetik kodlama kullanmadan 0,2 ile 0,4 arasında geliştirmişlerdir. Ancak, kodlama karmaşıklığı düşürmek için SPIHT'in bu versiyonunun daha fazla bellek alanına ihtiyacı vardır. SPIHT'in çok az enerji tükettiği ve yüksek sıkıştırma oranına sahip olduğu sonucu elde edilmiştir [91].

EBCOT kodlama SPIHT ile karşılaştırıldığında daha yüksek bir sıkıştırma verimliliği sağlamaktadır, ancak EBCOT'ta çok katmanlı kodlama işlemleri vardır. Aritmetik kodlama için birden fazla kodlama tablolarına ihtiyaç duyar, bu da sistemi daha karmaşık ve ekstra bellek ayırarak maliyetli hale getirmektedir. Bu durum SPIHT'i daha ön plana çıkarmaktadır [9], [92]. Ayrıca, SPIHT tekniği yüksek sıkıştırma oranı, az hesaplama karmaşası, düşük enerji tüketimi, EZW [52] ve DCT'den daha az işlem karmaşıklığı ile enerji kısıtlı KMAA'larda daha iyi bir çözüm olmuştur [53].

LEICA'da görüntü ön plan ve arka plan olmak üzere ikiye ayrılır. Arka planı çok önemli sayılmayarak, bu bölgeye çok yüksek sıkıştırma gerçekleştirilir. LEICA'nın diğer görüntü sıkıştırma algoritmalarına göre işlem karmaşası ve enerji tüketimi gibi özellikler bakımından oldukça verimli olduğu görülmektedir. Ancak görüntü kalitesinin önemli olduğu uygulamalarda LEICA etkili değildir [51].

Yapılan değerlendirmeler sonucunda elde edilen bulgular Çizelge 2.1'de sunulmuştur.

Cizelge 2.1. KMAA’larda görüntü sıkıştırma algoritmalarının analizi.

	DCT [73], [75], [76]	EZW [52]	EBCOT [79], [80]	SPIHT [53]	LEICA [51]
Bellek Gereksinimi	Düşük	Orta	Yüksek	Orta	Orta
Hesaplama Yüğü	Düşük	Düşük	Yüksek	Düşük	Düşük
İşlem Karmaşası	Düşük	Düşük	Yüksek	Orta	Orta
Enerji Tüketimi	Orta	Orta	Yüksek	Düşük	Düşük
Sıkıştırma Oranı	Düşük	Orta	Yüksek	Yüksek	Yüksek
İşlem Hızı	Yüksek	Yüksek	Düşük	Yüksek	Yüksek
Sıkıştırılmış Görüntünün Kalitesi	Düşük	Yüksek	Yüksek	Yüksek	Orta
Hafıza Gereksinimi	Düşük	Orta	Yüksek	Orta	Orta
Sistem Karmaşası	Düşük	Orta	Yüksek	Orta	Orta
Kodlama Hızı	Yüksek	Yüksek	Orta	Yüksek	Yüksek
Ağm Yaşam Süresi	Orta	Orta	Düşük	Yüksek	Yüksek

KMAA’larda en çok kullanılan görüntü sıkıştırma algoritmalarını değerdendirildiğinde aşğıdaki sonuçlar çıkarılabilir:

- EZW, KMAA’larda ilk olarak kullanılan sıkıştırma algoritmalarındandır. Kendine benzeyen görüntülerden faydalanarak önemli bilginin yokluğunun tahmin edilmesi üzerine kurulu sistem oldukça çok sıkıştırma sağlmasına rağmen karmaşıklık içermektedir [52].
- DCT algoritması görüntüyü istenilen oranda sıkıştırmaya elverişlidir. KMAA’larda yaygın olarak kullanılır [80].
- EBCOT sıkıştırma algoritması kullanılırken kodlama süresi boyunca toplam enerjinin %70’i harcanır [61].
- SPIHT algoritması EZW’nin geliştirilmiş versiyonudur. SPIHT’in işlem hızı çok yüksektir ve diğer sıkıştırma yöntemlerine göre görüntünün kalitesini arttırmaktadır. Ayrıca, toplam harcadığı enerji oldukça düşük ve işlem hızı yüksektir [54].
- LEICA kullanımında ise, algılayıcı ağlarda ölü düğüm sayısı diğer tekniklere göre oldukça az olmaktadır. Ayrıca, ağ yaşam süresi JPEG ile kıyaslandığında %87 daha fazladır [51].

KMAA'lar için kullanılabilecek görüntü sıkıştırma algoritmaları ve teknikleri üzerinde durularak özellikleri karşılaştırılmış ve Çizelge 2.1'de algoritmalar özetlenmiştir. Yapılan araştırmalar sonucunda KMAA'larda enerji verimliliği bakımından en iyi sonucu veren üç görüntü sıkıştırma algoritması ön plana çıkmaktadır. Bu algoritmalar; DCT, SPIHT ve LEICA'dır.

2.4 BELİRLENEN GÖRÜNTÜ SIKIŞTIRMA ALGORİTMALARININ PERFORMANS ANALİZİ

Bu bölümünde, belirlenen üç görüntü sıkıştırma algoritmasının kodlanması yapılarak görüntü sıkıştırma performansları hesaplanmıştır. Ayrıca, algoritmaların kodlanma kısmında alınan görüntüyü sıkıştırma ve sıkıştırılan görüntüyü açma süreleri de hesaplanmıştır. Görüntüyü sıkıştırma ve açma süreleri, üzerinde işlem yapılan bilgisayarın o anki işlem yüküne göre değişiklik göstermektedir. Üzerinde işlem gerçekleştirilen bilgisayarın teknik özellikleri Bölüm 3.2'de verilen algılayıcı düğümün özellikleri ile aynıdır. Daha doğru sonuca ulaşabilmek için bu algoritmalar farklı zaman dilimlerinde yüz kez çalıştırılmış ve ortalama değerleri alınmıştır. Algoritmaların kodlanma işlemleri, kullanılacak olan benzetim aracı Network Simulator – 2 (NS-2)'ye entegre edilebildiği için C Programlama Dili'nde gerçekleştirilmiştir. Kodlama gerçekleştirildikten sonra aynı görüntü verisi (Şekil 2.6.a) her bir algoritmaya sunulmuş ve çalıştırılmıştır.

Bu tez çalışmasında gerçekleştirilen uygulamalarda yapılan kodlamaların doğruluklarının sağlanması referans alınan çalışmalardaki değerlerle karşılaştırılarak yapılmıştır. Bu amaçla DCT, SPIHT ve LEICA yöntemleri ile sıkıştırılan 256x256 Lena görüntüsünün PSNR değerleri aynı parametrelere göre hesaplanmış ve yöntemlerin önerildiği çalışmalardaki PSNR değerleri ile karşılaştırılmıştır [93], [94], [51]. PSNR değerlerinin bu benzerliği, gerçekleştirilen kodlamanın doğruluğunu göstermektedir. Enerji tüketimleri konusunda DCT'nin değerleri [93]'te verilen değerler ile, SPIHT'in değerleri [94]'te verilen değerleri ile, LEICA'nın değerleri ise [51]'de verilen değerler ile örtüşmektedir.

2.4.1 Ayrık Kosinüs Dönüşümü (Discrete Cosine Transform - DCT)'nin Performansı

DCT algoritmasında kullanıcı tarafından sıkıştırma oran seçimi gerçekleştirilebilir. Ancak, bu çalışmada sıkıştırma oranı sabit yüzde 10 olarak belirlenmiştir. Bunun sebebi kıyaslayacağımız diğer algoritmalar ile sıkıştırma oranlarının yaklaşık aynı olmasını sağlamaktır. DCT algoritmasının arayüzü (orijinal görüntü ve sıkıştırılmış görüntü) Şekil 2.6'da yer almaktadır. Görüntünün orijinal ve sıkıştırılmış boyut değerlerinin karşılaştırılması Çizelge 2.2'de verilmiştir. Algoritmanın ortalama sıkıştırma ve açma süreleri Çizelge 2.3'te yer almaktadır.



(a)

(b)

Şekil 2.6. DCT algoritması için (a) orijinal görüntü (b) sıkıştırılmış görüntü.

Çizelge 2.2. DCT için orijinal ve sıkıştırılmış görüntülerin karşılaştırılması.

Orijinal Görüntü Boyutu	Sıkıştırılmış Görüntü Boyutu	Açılmış Görüntü Boyutu
256 KB (262,144 bayt)	21.1 KB (21.697 bayt)	256 KB (262,144 bayt)

Çizelge 2.3. DCT algoritmasının görüntüyü sıkıştırma zaman analizi.

	Sıkıştırma Süresi	Açma Süresi
DCT	54.843 ms	17.500 ms

2.4.2 Hiyerarşik Ağaçlarda Küme Bölümleme (Set Partitioning in Hierarchical Trees - SPIHT) 'nin Performansı

Hiyerarşik ağaçlarda küme bölümleme (SPIHT) algoritmasının arayüzü (orijinal görüntü ve sıkıştırılmış görüntü) Şekil 2.7'de yer almaktadır. Görüntünün orijinal ve sıkıştırılmış boyut değerlerin karşılaştırılması Çizelge 2.4'te verilmiştir. Algoritmanın ortalama sıkıştırma ve açma süreleri Çizelge 2.5'te yer almaktadır.



Şekil 2.7. SPIHT algoritması için (a) orijinal görüntü (b) sıkıştırılmış görüntü.

Çizelge 2.4. SPIHT için orijinal ve sıkıştırılmış görüntülerin karşılaştırılması.

Orijinal Görüntü Boyutu	Sıkıştırılmış Görüntü Boyutu	Açılmış Görüntü Boyutu
256 KB (262,144 bayt)	16 KB (16,385 bayt)	256 KB (262,144 bayt)

Çizelge 2.5. SPIHT algoritmaların görüntüyü sıkıştırma zaman analizi.

	Sıkıştırma Süresi	Açma Süresi
SPIHT	57,968 ms	19,697 ms

2.4.3 Düşük Enerjili Görüntü Sıkıştırma Algoritması (Low Energy Image Compression Algorithm - LEICA) 'nın Performansı

Düşük Enerjili Görüntü Sıkıştırma ve İletme (LEICA) algoritmasının görüntüyü sıkıştırmasının dışında çalışma prensibi olarak düğümleri istenildiği zaman uyuyup uyandırarak daha da fazla enerji tasarrufu sağlaması mümkündür. Ancak, şu an gerçekleştirdiğimiz benzetimde bütün algoritmaların eşit durumda kıyaslanması için LEICA'nın sadece görüntü sıkıştırma özelliği kullanılmıştır. LEICA algoritmasının ara yüzü (orijinal görüntü ve sıkıştırılmış görüntü) Şekil 2.8'de yer almaktadır. Görüntünün orijinal ve sıkıştırılmış boyut değerlerin karşılaştırılması Çizelge 2.6'da verilmiştir. Algoritmanın ortalama sıkıştırma ve açma süreleri Çizelge 2.7'de yer almaktadır.



(a)

(b)

Şekil 2.8. LEICA algoritması için (a) orijinal görüntü (b) sıkıştırılmış görüntü.

Çizelge 2.6. LEICA için orijinal ve sıkıştırılmış görüntülerin karşılaştırılması.

Orijinal Görüntü Boyutu	Sıkıştırılmış Görüntü Boyutu	Açılmış Görüntü Boyutu
256 KB (262,144 bayt)	19 KB (19,527 bayt)	256 KB (262,144 bayt)

Çizelge 2.7. LEICA algoritmaların görüntüyü sıkıştırma zaman analizi.

	Sıkıştırma Süresi	Açma Süresi
LEICA	54,707 ms	16,250 ms

2.5 BELİRLENEN GÖRÜNTÜ SIKIŞTIRMA ALGORİTMALARININ KARŞILAŞTIRILMASI

Kodlanan görüntü sıkıştırma algoritmalarından alınan performans değerlerinin sonuç karşılaştırılması Çizelge 2.8’de verilmiştir.

Çizelge 2.8. Belirlenen algoritmaların görüntü sıkıştırma performanslarının karşılaştırılması.

	Orijinal Görüntü Boyutu	Sıkıştırılmış Görüntü Boyutu	Açılmış Görüntü Boyutu
DCT	256 KB (262,144 bayt)	21 KB (21,697 bayt)	256 KB (262,144 bayt)
SPIHT	256 KB (262,144 bayt)	16 KB (16,385 bayt)	256 KB (262,144 bayt)
LEICA	256 KB (262,144 bayt)	19 KB (19,527 bayt)	256 KB (262,144 bayt)

Her bir algoritma için görüntüyü sıkıştırma ve sıkıştırılan görüntüyü açma sürelerinin ortalaması alınmıştır ve alınan ortalama değerlerin karşılaştırılması Çizelge 2.9’da gösterilmiştir.

Çizelge 2.9. Belirlenen algoritmaların görüntüyü sıkıştırma zaman analizi.

	Sıkıştırılmış Görüntü Boyutu	Sıkıştırma Süresi	Açma Süresi
DCT	21 KB	54,843 ms	17,500 ms
SPIHT	16 KB	57,968 ms	19,697 ms
LEICA	19 KB	54,707 ms	16,250 ms

2.6 SONUÇ

Bu bölümde, KMAA için literatürde yer alan görüntü sıkıştırma algoritmalarının üzerinde durulmuş ve enerji verimliliği açısından üç görüntü sıkıştırma algoritması belirlenmiştir. Belirlenen görüntü sıkıştırma algoritmalarının sıkıştırma ve açma süreleri açısından performans ölçümleri gerçekleştirilmiştir.

3. KABLOSUZ MULTİMEDYA ALGILAYICI AĞLAR İÇİN GÖRÜNTÜ SIKIŞTIRMA ALGORİTMALARINA DAYALI YENİ BİR ENERJİ DUYARLI UYGULAMA KATMANI ALGORİTMASI GELİŞTİRME

3.1 GİRİŞ

Bölüm 2’de KMAA’lar için kullanılan görüntü sıkıştırma algoritmaları incelenmiştir. Yapılan literatür taraması ve analizler sonucunda KMAA’larda kullanılan ve enerji verimliliği açısından en iyi sonucu veren üç görüntü sıkıştırma algoritmasının kullanılmasına karar verilmiştir. Bu algoritmalar DCT, SPIHT ve LEICA’dır. Daha sonra algoritmaların görüntüyü sıkıştırma performansları, sıkıştırma süreleri ile açma süreleri belirlenmiş ve geliştirilen sistemde kullanılması planlanmıştır.

Bu bölümde KMAA’larda kullanılmak üzere DCT, SPIHT ve LEICA’yı esas alan enerji duyarlı algoritma geliştirmeye yer verilmiştir. Geliştirilen algoritmaya, Kablosuz Multimedya Algılayıcı Ağlar için Görüntü Sıkıştırmaya Dayalı Yeni Bir Enerji Duyarlı Uygulama Katmanı Algoritması, kısaca GSD-EDUKA olarak ifade edilmektedir. Bölümün devamında GSD-EDUKA’nın çalışma prensiplerine değinilecektir.

GSD-EDUKA’da bulunan algılayıcı düğümler Ad hoc On Demand Distance Vector Routing (AODV) yönlendirme protokolündeki “hello” paketinin değiştirilmesiyle elde edilen Modify-Ad hoc On Demand Distance Vector Routing (M-AODV) yönlendirme protokolü kullanılarak birbirleri arasındaki mesafe bilgisine sahip olurlar. Ağın kurulum aşamasında kullanılacak düğüm bilgisi ve veri gönderme sıklığı bilgisi belirlenir. Alınan veriler işlenerek karar ağacı yöntemiyle kural tabanlı bilgilere çevrilir ve bulanık mantık yöntemi aracılığıyla elde edilen bilgilere göre Bölüm 2’de belirlenen KMAA’larda kullanılan görüntü sıkıştırma algoritmalarından en verimli olanı seçme işlemi gerçekleştirilir. Daha sonra, sıkıştırılmış görüntünün Tepe Sinyal Gürültü Oranı (Peak Signal to Noise Ratio – PSNR) hesaplanır. Kablosuz iletişimde, kabul edilebilir PSNR kayıp değeri yaklaşık 20 ile 25 dB arasındadır [68], [69]. Orijinal görüntünün PSNR

değeri ile sıkıştırılmış görüntünün PSNR değeri arasındaki fark, kabul edilebilir kayıp değerleri arasında ise görüntü bir sonraki algılayıcı düğüme veya baz istasyonuna aktarılır. Orijinal görüntünün PSNR değeri ile sıkıştırılmış görüntünün PSNR değeri arasındaki fark, 20 dB'ye eşit veya küçük ise sıkıştırma oranı artırılarak tekrar sıkıştırma işlemi gerçekleştirilir ve görüntü bir sonraki algılayıcı düğüme veya baz istasyonuna aktarılır. Ancak, orijinal görüntünün PSNR değeri ile sıkıştırılmış görüntünün PSNR değeri arasındaki fark, 20 dB'den büyük ise sıkıştırma oranı azaltılarak tekrar sıkıştırma işlemi gerçekleştirilir ve görüntü bir sonraki algılayıcı düğüme veya baz istasyonuna aktarılır.

Algılayıcı düğümlerin harcadıkları gönderme ve alma enerji değerlerinin azaltılması planlanmaktadır. Harcadıkları gönderme ve alma enerjileri azalan algılayıcı düğümlerin, batarya ömürlerini etkin bir biçimde kullanmış olacaklarından dolayı oluşturulan kablosuz ağın yaşam süresinin uzatılmış olması hedeflenmektedir. Ayrıca sıkıştırılmış görüntünün PSNR değeri, kablosuz iletişimde kabul edilebilir PSNR değerinde olması sağlanarak transfer edilen görüntülerin kalitesinden ödün verilmemesi amaçlanmaktadır.

GSD-EDUKA sayesinde, belirlenen durumlara göre en verimli olan görüntü sıkıştırma algoritmasına karar verilerek, algılayıcı düğümlerin harcadıkları enerji değerleri daha da aza indirilecektir. Yapılan enerji hesaplamaları işlemlerinde, kullanılan görüntü sıkıştırma algoritmalarının harcadığı enerji ve GSD-EDUKA'nın ihtiyaç duyduğu enerjiler de göz önünde bulundurulmuştur.

3.2 ALGILAYICI DÜĞÜMLERİN SAHİP OLMASI GEREKEN ÖZELLİKLER

KMAA'ların ilettikleri verilerin boyutları KAA'lara göre oldukça fazladır. Dolayısıyla harcanan enerji de fazla olmaktadır. Ayrıca, KMAA'lardaki algılayıcı düğümlerin elde ettikleri veri üzerinde işlem yapmaları gerekebileceğinden algılayıcı düğümlerin bellek ve hafıza gereksinimleri vardır.

Geliştirilen sistemde algılayıcı düğümlerin sahip olması gereken özellikler aşağıdaki gibidir;

- En az 64 MB RAM içermeli,
- En az 32 MB hafızaya sahip olmalı,
- En az 400 MHz işlemciye sahip olmalı ve

- Üzerinde GPS (Global Positioning System) bulundurmalıdır.

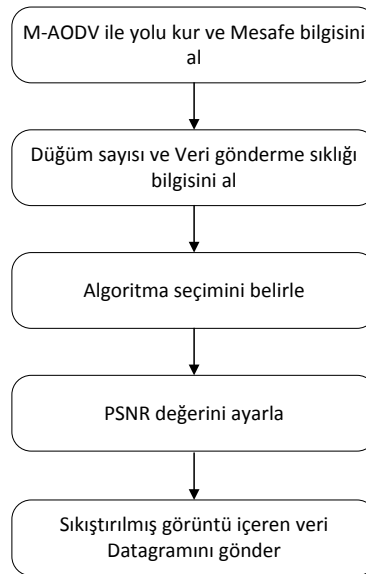
Kullanılan düğümler üzerlerinde görüntü sıkıştırma algoritmaları kullanıldığından 64 MB RAM'e ve 32 MB hafızaya gereksinim duyulmuştur.

3.3 GSD-EDUKA'NIN YAPISI

GSD-EDUKA'nın adımları kısaca şu şekildedir:

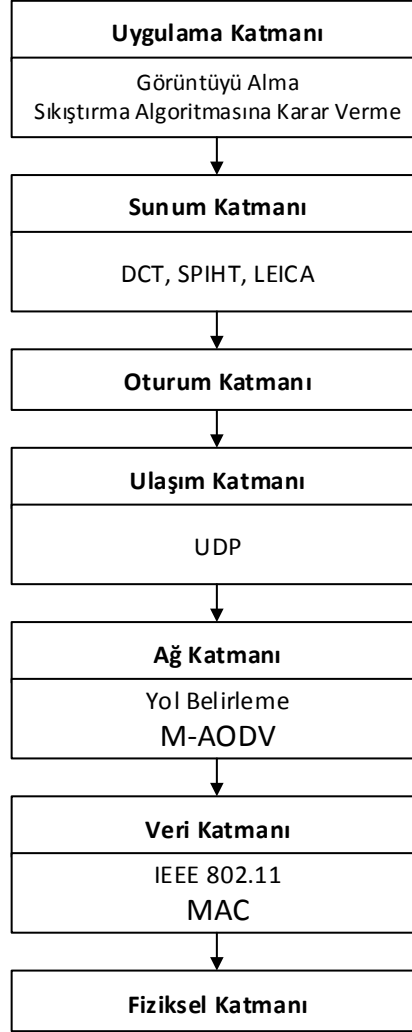
- a. Her bir algılayıcı düğüm, Modify-Ad hoc On Demand Distance Vector Routing (M-AODV) yönlendirme protokolünü kullanarak görüntü alış verışı için yolunu kurar ve mesafe bilgisini hesaplar,
- b. Düğüm sayısı ve veri gönderme sıklığı bilgisi alınır,
- c. Belirlenen parametreler alınarak kullanılacak olan görüntü sıkıştırma algoritmasına karar verilir,
- d. Algılayıcı düğüm bulunduğu ortamın görüntüsünü alır ve karar verilen algoritma ile sıkıştırma işlemi gerçekleştirilir,
- e. Orijinal görüntü ve sıkıştırılmış görüntü arasındaki PSNR değeri farkı, kablosuz iletişimde kabul edilebilir değere ulaştırılır,
- f. Sıkıştırılmış görüntü içeren veri datagramını bir sonraki algılayıcı düğüme gönderir.

Çalışma adımları kısaca verilen sistemin işlem adımları Şekil 3.1'de görülmektedir.



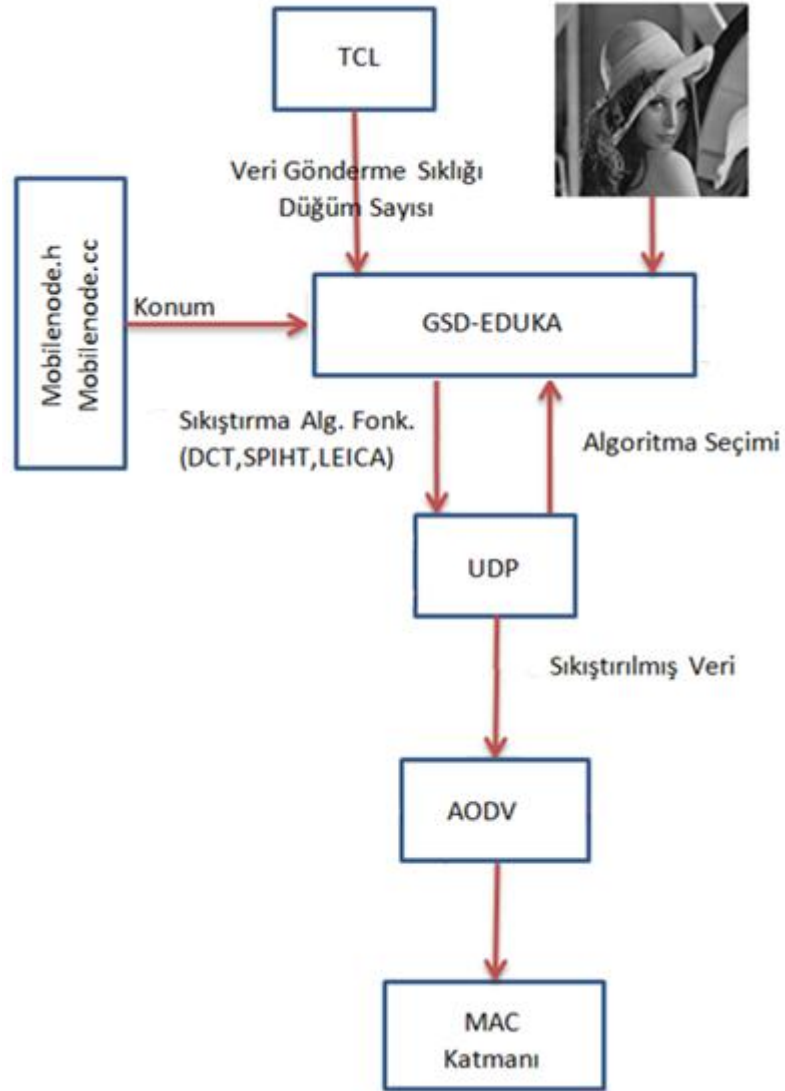
Şekil 3.1. GSD-EDUKA'nın işlem adımları.

GSD-EDUKA'nın her bir katmanda gerçekleştirdiği işlemler veya kullanılan protokoller Şekil 3.2'de yer almaktadır. Algılayıcı düğümler veri alış verişi yaparken bu katman yapısını kullanırlar.



Şekil 3.2. GSD-EDUKA'nın OSI katmanlarında gerçekleşen işlemleri.

GSD-EDUKA'nın yapısı Şekil 3.3'te verilmiştir.



Şekil 3.3. GSD-EDUKA'nın yapısı.

3.4 GSD-EDUKA’NIN ADIMLARI

GSD-EDUKA aşağıdaki adımlara sahiptir;

- a. Her bir algılayıcı düğüm, iletişime geçebildiği düğümler arasındaki mesafe bilgisini, AODV yönlendirme algoritmasındaki “hello” paketinin değiştirilmesiyle elde edilen M-AODV yönlendirme protokolü sayesinde hesaplar ve bu bilgiyi GSD-EDUKA’ya aktarır,
- b. Kullanılacak düğüm sayısı ve veri gönderme sıklığı bilgileri alınır,
- c. Karar ağacı yöntemi aracılığıyla veriler (mesafe, düğüm sayısı ve veri gönderme sıklığı), kural tabanlı bilgilere çevrilir ve bulanık mantık yöntemi ile kullanılacak görüntü sıkıştırma algoritmasına karar verilir,
- d. Algılayıcı düğüm görüntüyü alır,
- e. Orijinal görüntünün PSNR değeri hesaplanır,
- f. Her bir algoritma için sıkıştırma oranı 5 olarak belirlenir,
- g. Belirlenen görüntü sıkıştırma algoritması ile sıkıştırma işlemi gerçekleşir,
- h. Sıkıştırılmış görüntünün PSNR değeri hesaplanır,
 1. Orijinal görüntünün PSNR değeri ile sıkıştırılmış görüntünün PSNR değeri arasındaki fark, 20 dB’den büyük ve 25 dB’ye eşit veya küçük ise, elde edilen görüntü bir sonraki algılayıcı düğüme aktarılır.
 2. Değilse, orijinal görüntünün PSNR değeri ile sıkıştırılmış görüntünün PSNR değeri arasındaki fark, 20 dB’ye eşit veya küçük ise, sıkıştırma oranını artır ve görüntü tekrar sıkıştırılmak için “g” adımına gidilir,
 - i. Değilse, sıkıştırma oranını azaltılır ve görüntüyü tekrar sıkıştırılmak için “g” adımına gidilir (PSNR değeri, kablosuz ağlarda görüntü aktarımında kabul edilebilir kayıp değer aralığına ulaşılan kadar bu işlem gerçekleştirilir. Bu sayede, hem kabul edilebilir kayıp PSNR değerine ulaşılır hem de daha fazla sıkıştırma gerçekleştirilerek görüntünün daha az boyuta indirilmesi ile görüntünün aktarımında daha az enerji harcanması sağlanmış olur).

GSD-EDUKA’nın akış diyagramı Şekil 3.4’te verilmiştir. Kullanılan parametrelerin açıklamaları bölümün devamında yer almaktadır.

GSD-EDUKA'nın sözde kodu aşağıdaki gibidir;

ALGORİTMA GSD-EDUKA (cp, D, Ds)

// input : cp, hello paketinin cevabı, görüntünün boyutu

// output: sıkıştırılmış görüntü

// cp_x, cp_y, komşu düğüm koordinatları

// S: veri gönderim sıklığı, Ds: Ağdaki düğüm sayısı

dugum $\leftarrow$ hello paketini gönderen düğüm

cp_x $\leftarrow$ cevap paketi içinde gelen X koordinatını al

cp_y $\leftarrow$ cevap paketi içinde gelen Y koordinatını al

dugum_x $\leftarrow$ GPS_X_koordinatını_al

dugum_y $\leftarrow$ GPS_X_koordinatını_al

mesafe $\leftarrow$ sqrt((dugum_x-cp_x)^2+(dugum_y-cp_y)^2)

i $\leftarrow$ Görüntü_çek

P $\leftarrow$ psnr(i) // i resminin psnr değeri hesabı

D $\leftarrow$ 5 // Sıkıştırma oranı

Sa $\leftarrow$ algoritmaSecimi(S,Ds,mesafe) // Sıkıştırma algoritması seçimi

While do

Sg $\leftarrow$ goruntuSIKISTIR(D,Sa,i) // Sa'ya göre %D sıkıştırılmış görüntü

Sg_P $\leftarrow$ psnr(Sg) // Sıkıştırılmış görüntünün psnr değeri hesabı

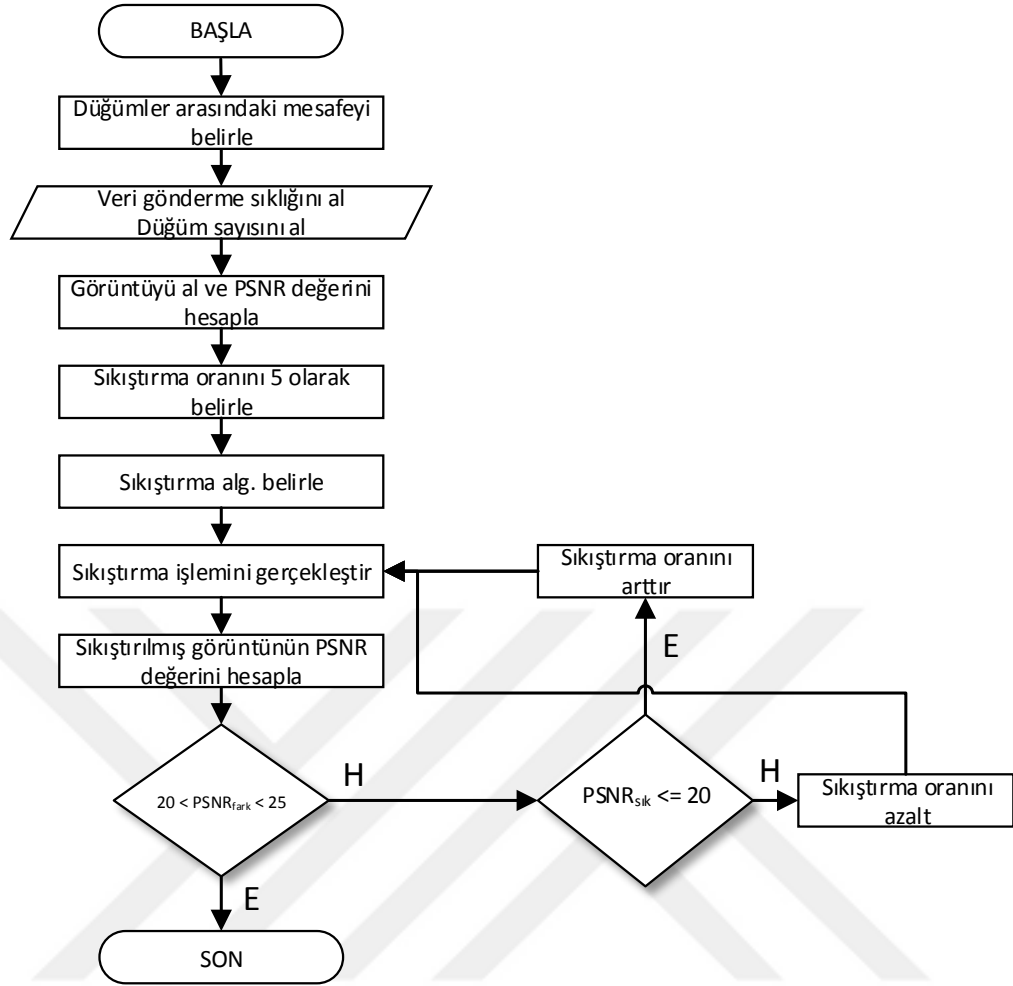
if (P-Sg_P) < 25 ve (P-Sg_P) > 20

return i // cp_x,cp_y koordinatlı düğümüne veri gönder

elseif (P-Sg_P) >= 25

D $\leftarrow$ D + 5

else D $\leftarrow$ D - 5



Şekil 3.4. GSD-EDUKA'nın akış diyagramı.

3.5 GSD-EDUKA'NIN PARAMETRELERİ

GSD-EDUKA, seçim işlemi için öncelikle belirlenen parametreleri işleyerek kullanılabilir algoritmalara karar vermektedir. Daha sonra belirlenen algoritmalar arasında bütün parametreler hesaba katılarak en verimli olanının belirlenmesini sağlamaktadır. Bunun için karar ağacı yöntemi kullanılarak elde edilen durum sayısı daha aza indirgenmiş ve bulanık mantık yöntemi ile de en uygun algoritmaya karar verme işlemi gerçekleştirilmiştir.

GSD-EDUKA'ya bağlı parametreler sistemin ihtiyaçlarına göre belirlenmiştir. Bu parametreler ve belirlenme sebeplerine değinecek olursak;

- Ağ üzerindeki algılayıcı düğümlerin arasındaki mesafe algılayıcı düğümlerin gönderme ve alma enerjilerini etkilemektedir [95]. Geliştirilen sistemde bu durum göz önünde bulundurularak “düğümler arasındaki mesafe” parametre

olarak belirlenmiştir. Sistemde algılayıcı düğümler hareketli olabilmektedir. Dolayısıyla düğümlerin birbirleri arasındaki mesafe her an değişiklik gösterebilir. Kapsama alanı 200 metre olan algılayıcı düğümlerin birbirlerine yakın olduğu durumlar için oluşturulan bulanık üyelik fonksiyonunun dilsel değişkeni “az”, kapsama alanının sınırında olduğu durumlar için “çok” ve her iki durumun arasında olduğu durumlar için “orta” olarak belirlenmiştir.

- b. Ağ üzerinde bulunan algılayıcı düğümlerin sayısı toplamda harcanan enerjiyi etkilemektedir. Fazla düğüm olması halinde enerji sarfiyatının artacağı kaçınılmazdır [96], [97]. Ayrıca, düğüm sayısı uygulamanın kapsadığı alanı da etkilemektedir. Algılayıcı düğümlerin kapsama alanları göz önünde bulundurularak toplam kullanılan düğüm sayısı belirlenebilir [97]. Bu çıkarımlar sonucunda “kullanılan düğüm sayısı” parametre olarak belirlenmiştir. Ağda kullanılan toplam düğüm sayısı değerine ulaşmak için kurulacak sistemin kapsadığı alan, kurulum aşamasında kullanıcı tarafından belirtilmektedir. Algılayıcı düğümlerin de kapsama alanları göz önüne alınarak ortalama düğüm sayısı değeri hesaplanmaktadır. Hesaplanan bu değer uygulamanın hitap ettiği alana göre oluşturulan bulanık üyelik fonksiyonunun dilsel değişkenleri “az”, “orta” ve “çok” olarak belirlenmiştir.
- c. Ağ üzerindeki algılayıcı düğümler elde ettikleri görüntüye gerekli işlemleri uyguladıktan sonra kapsama alanında bulunan bir sonraki algılayıcı düğüme iletirler. Bu görüntü verilerinin belirli zaman dilimlerinde iletilmesi enerji açısından önemlidir. Çünkü her iletimde kullanılan bütün düğümler gönderme ve alma enerjisi harcamaktadır [98]. Bu sebepten “veri gönderme sıklığı” geliştirilen sistemde parametre olarak alınmış ve kullanıcının isteğine bağlı olarak oluşturulan bulanık üyelik fonksiyonunun dilsel değişkenleri “az”, “orta” ve “çok” olarak belirlenmiştir.

Parametrelerden elde edilen değerler sonucunda her bir parametre için enerji verimliliği açısından farklı görüntü sıkıştırma algoritması kullanılması gerekebilir. Örneğin, düğümler arasındaki mesafenin “az” olduğu durumda “DCT” algoritmasının kullanılması verimli olabilirken, ağda kullanılan düğüm sayısının “çok” olduğu durumda “SPIHT” algoritmasının, veri gönderme sıklığının “çok” olduğu durumda da “LEICA” algoritmasının kullanılması verimli olabilir. Ayrıca, birden fazla parametrenin

de etki ettiđi durumlar söz konusu olabilir. Örneđin, düğümler arası mesafe “çok” iken, kullanılan düğüm sayısı “orta” ve veri gönderme sıklığı “az” olabilir. Bu ve bunun gibi bütün durumlarda GSD-EDUKA, bulanık mantık yöntemi kullanarak belirlenen görüntü sıkıştırma algoritmalarından en uygun ve en verimli olan algoritmayı seçerek algılayıcı düğümün kullanmasını sağlar.

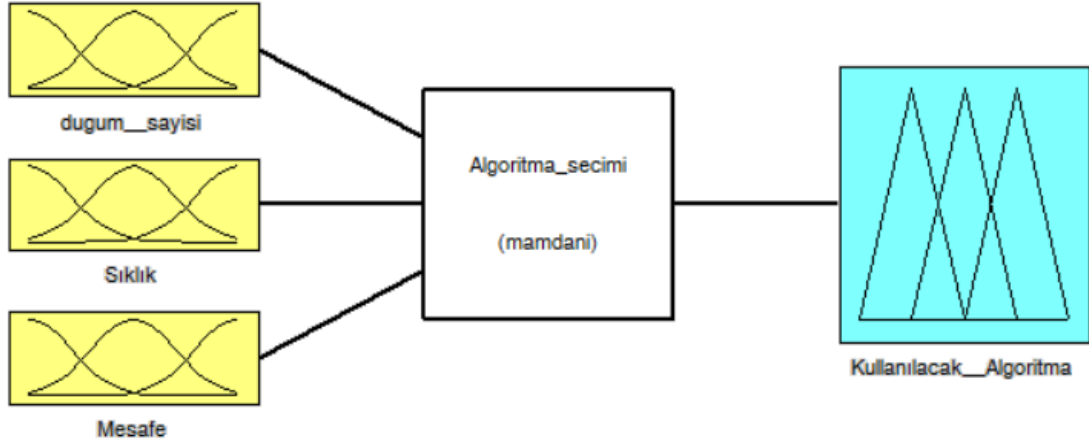
3.6 SIKIŞTIRMA ALGORİTMASINI BELİRLEME

Belirlenen parametrelere göre geliştirilen GSD-EDUKA sayesinde algılayıcı düğümlerin en verimli algoritmayı seçerek kullanmaları sağlanmaktadır. En uygun sıkıştırma algoritmasının belirlenmesinde seçilecek yöntemin sıkıştırma işlemi sırasında harcadığı enerji (işlem karmaşası), sıkıştırma miktarı, çıkış verisindeki kalite kaybı ve mesafeye bağılı değerlendirme etkili olmaktadır. Örneđin mesafeye bağılı değerlendirmede, işlem karmaşası ile sıkıştırılmış verinin iletiminde harcanan enerjilerin toplamı tüm sıkıştırma yöntemleri için hesaplanmakta ve en az enerjiyi harcayan yöntem tercih edilmektedir. Yöntem yüksek sıkıştırma kabiliyetine sahip olabilir fakat karmaşası da yüksek olabilir. Kısa mesafede verinin iletileceğı yol kısa olduğundan sıkıştırma işlemi için fazla enerji harcamak gereksiz olabilir. Veriyi daha basit (işlem maliyeti nispeten düşük) bir yöntemle sıkıştırarak iletmek daha az maliyetli olabilir. Bu sebeple, tercih edilecek sıkıştırma yöntemi mesafeye bağılı olarak değişmektedir. Veri gönderme sıklığına bağılı değerlendirme yaparken gecikmeye sebep olmamak ve oluşacak gecikmeleri minimize etmek amacıyla karmaşası en az olan yöntem tercih edilmektedir. Düğüm sayısı değişken olduğu durumda ise örneđin düğüm sayısı fazla olduğunda, atlama sayısı fazla olacak ve her atlama ek maliyet getirecektir. Bu durumda bir de sıkıştırma işlemi karmaşık olursa toplam maliyet artmakta, dolayısıyla fazla enerji harcanmaktadır. Yani, düğüm sayısına bağılı değerlendirmede işlem karmaşası etkili olmaktadır.

Bu bölümde öncelikle sistemin giriş değişkenleri ve değeri aralıkları belirlenecektir. Daha sonra bu değişkenler bulanık mantık yönteminde kullanılarak oluşturulan ağı üzerindeki algılayıcı düğümlerin en uygun görüntü sıkıştırma algoritması seçimi sağlanacaktır.

Bulanık mantık yöntemi için üç adet bulanık giriş değişkeni mevcuttur. Bunlar GSD-EDUKA için belirlenen parametreler olan “düğüm sayısı”, “sıklık” ve “mesafe”dir.

Bulanık çıkış değişkeni ise “Kullanılacak Algoritma” olarak adlandırılmıştır. Blok görünümü Şekil 3.5’te verilen bulanıklaştırma işlemi, MATLAB 2014a Fuzzy Logic Toolbox ile oluşturulmuştur.



Şekil 3.5. Bulanık Mantık giriş ve çıkış değişkenlerinin blok görünümü.

3.6.1 Döğümler Arası Mesafeye Göre Kullanılacak Algoritmayı Belirleme

Birbirlerinin kapsama alanına giren algılayıcı döğümler arasındaki mesafeyi hesaplamak için kullanılan AODV yönlendirme protokolünün içerdiği “hello” paketi değiştirilmiştir ve protokole Modify-Ad hoc On Demand Distance Vector Routing (M-AODV) adı verilmiştir. Algılayıcı döğümler üzerinde bulunan GPS modülünden alınan konum bilgisi, değiştirilen “hello” paketine eklenmiştir. Bu paket komşu döğümlere dağıtılarak birbirlerinin konum bilgilerinden haberdar olması sağlanmıştır. Daha sonra bu bilgileri kullanarak döğümler arasındaki mesafe hesaplanmıştır. M-AODV kullanılmaz ise coğrafi tabanlı algoritmalar kullanılması gerekmektedir. Coğrafi tabanlı algoritmalar oldukça fazla kontrol paketi kullanarak, çok enerji harcamaktadır [99], [100].

AODV’nin “hello” paketinin çerçeve yapısı Şekil 3.6’da ve değiştirilen “hello” paketinin çerçeve yapısı da Şekil 3.7’de verilmiştir.

Tip 1 Byte	Uzunluk 1 Byte	“hello” Mesajı 4 Byte
---------------	-------------------	--------------------------

Şekil 3.6. “hello” paketinin çerçeve yapısı.

Tip 1 Byte	Uzunluk 1 Byte	“hello” Mesajı 4 Byte	Konum (x, y) 8 Byte
---------------	-------------------	--------------------------	------------------------

Şekil 3.7. Değiştirilen “hello” paketinin çerçeve yapısı.

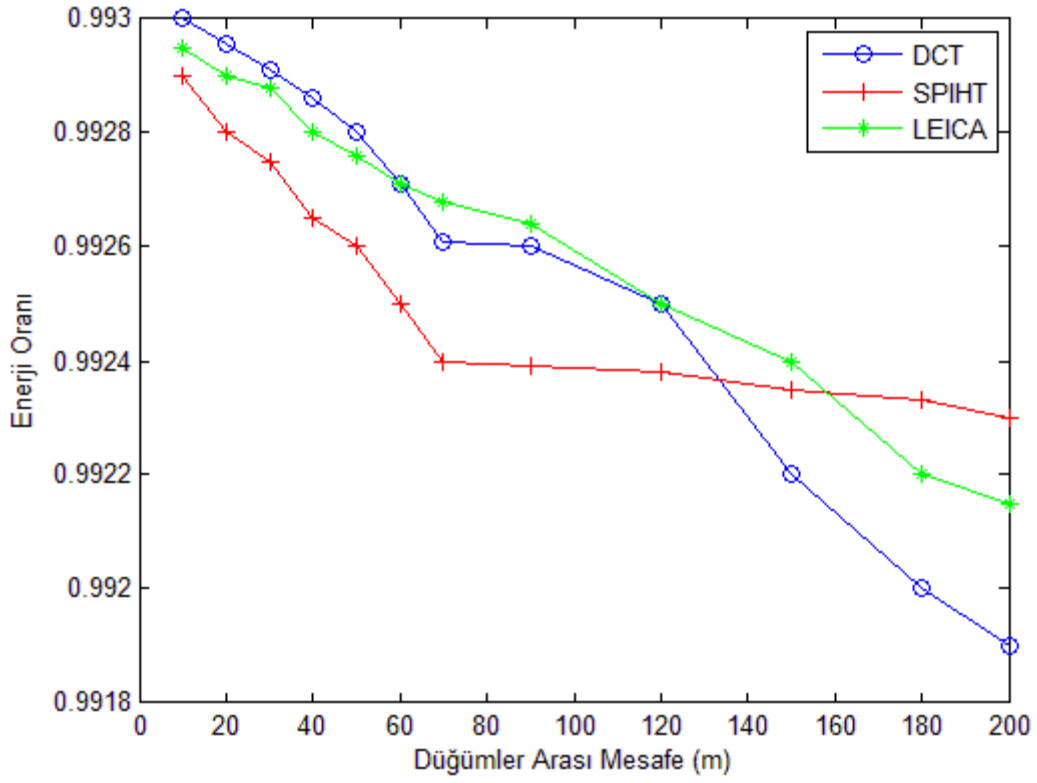
GPS modülü sayesinde her iki düğümden alınan konum bilgileri ile düğümler arasındaki mesafe Denklem (3.1)'de verilen eşitlik kullanılarak hesaplanır. Böylece, sıkıştırılmış görüntü verisini birbirlerine iletilecek olan düğümler arasındaki mesafe hesaplanmış olur.

$$M_{i,j} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \quad (3.1)$$

Düğümlerin birbirleri arasındaki mesafeye göre kullanılacak sıkıştırma algoritması değişiklik göstermektedir. Benzetim sonuçlarına göre, üç sıkıştırma algoritmasının düğümler arasındaki mesafeye göre “enerji oranı” değişim değerleri (kalan enerjinin toplam enerjiye oranı) Çizelge 3.1’de, grafiği ise Şekil 3.8’de verilmektedir.

Çizelge 3.1. Düğümler arasındaki mesafeye göre “enerji oran” değişim değerleri.

Mesafe (m)	Enerji Oranı (Kalan Enerji / Toplam Enerji)		
	DCT	SPIHT	LEICA
0-10	0,993	0,9929	0,99295
10-20	0,992955	0,9928	0,9929
20-30	0,99291	0,99275	0,99288
30-40	0,99286	0,99265	0,9928
40-50	0,9928	0,9926	0,99276
50-60	0,99271	0,9925	0,99271
60-70	0,99261	0,9924	0,99268
70-90	0,9927	0,99262	0,99272
90-120	0,9925	0,99255	0,99262
120-150	0,9922	0,99245	0,9924
150-180	0,992	0,9924	0,9922
180-200	0,9919	0,9923	0,99215



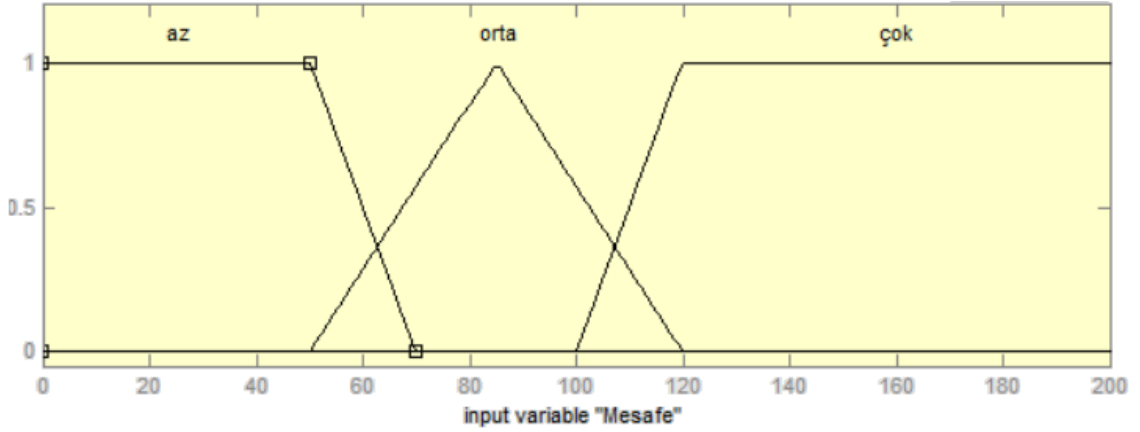
Şekil 3.8. Düşümler arasındaki mesafeye göre “enerji oran” değişim grafiği.

Elde edilen grafiklere göre, düşümler arası mesafesinin oldukça az olduğu durumlarda DCT algoritmasının, orta olduğu durumlarda LEICA algoritmasının ve çok olduğu durumlarda ise SPIHT algoritmasının kullanımı enerji bakımından daha verimli olduğu görülmektedir. Ancak, bazı durumlarda kullanılacak algoritmaya net olarak karar verilememektedir. Yani bazı algılayıcı düşümler arasındaki mesafe aralıklarında birden fazla algoritmanın verimi yüksek olduğu görülmektedir. Bu durumlarda en uygun algoritmaya karar vermek için bulanık mantık yöntemi kullanıldığında;

“Mesafe” parametresi 0 ile 200 metre arasında değişmektedir. Bu giriş değişkeni için üç üyelik fonksiyonu oluşturularak bunlar “az”, “orta” ve “çok” olarak isimlendirilmiştir. “mesafe” giriş değişkenine ait üyelik fonksiyonlarının sınırları Çizelge 3.2’de ve grafik görünümü Şekil 3.9’da verilmiştir.

Çizelge 3.2. “Mesafe” giriş değişkeni üyelik fonksiyonu sınırları.

Dilsel Değişken	Üyelik Fonksiyonu Sınırı	
Az	0	70
Orta	50	120
Çok	100	200



Şekil 3.9. “Mesafe” giriş değişkeni üyelik fonksiyonlarının grafik görünümü.

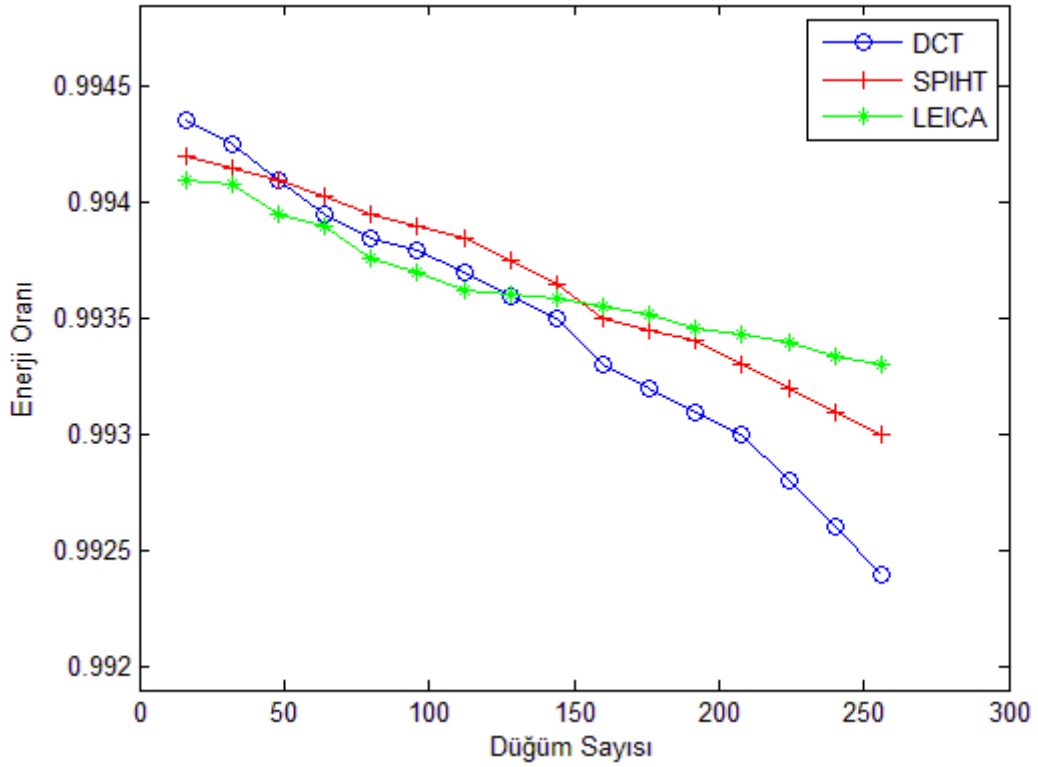
Bu çıkarımda sonuç olarak “mesafe” parametresinin “az ve orta” olduğu durumlarda kullanılacak görüntü sıkıştırma algoritmasının DCT veya LEICA, “orta ve çok” olduğu durumlarda da LEICA veya SPIHT algoritmalarının kullanılacağı anlaşılmaktadır.

3.6.2 Ağda Kullanılan Düğüm Sayısına Göre Kullanılacak Algoritmayı Belirleme

Gerçekleştirilecek olan sistemlerde KMAA’ların hitap edeceği alan kurulum esnasında belirli olduğu için kullanılacak düğüm sayısı da kullanılan alana göre ortalama olarak bilinmektedir. Sıkıştırılmış görüntü verisi iletilirken, ağda kullanılan düğüm sayısına göre kullanılacak algoritma değişmektedir. Gerçekleştirilen benzetim sonuçlarına göre, üç sıkıştırma algoritmasının ağdaki düğüm sayısına göre “enerji oran” değişim değerleri (kalan enerji / toplam enerji) Çizelge 3.3’te, grafiği ise Şekil 3.10’da verilmektedir.

Çizelge 3.3. Kullanılan toplam düğüm sayısına göre “enerji oran” değişim değerleri.

Düğüm Sayısı	Enerji Oranı (Kalan Enerji / Toplam Enerji)		
	DCT	SPIHT	LEICA
16	0,994354	0,9942	0,9941
32	0,99425	0,99415	0,994084
48	0,9941	0,9941	0,99395
64	0,99394	0,99402	0,9939
80	0,99385	0,99395	0,993762
96	0,9938	0,9939	0,993697
112	0,9937	0,99385	0,99362
128	0,9936	0,99375	0,99361
144	0,9935	0,99365	0,99359
160	0,9933	0,9935	0,99355
176	0,9932	0,99345	0,99352
192	0,9931	0,99341	0,99346
208	0,993	0,9933	0,99343
224	0,9928	0,9932	0,9934
240	0,9926	0,9931	0,99334
256	0,9924	0,993	0,9933



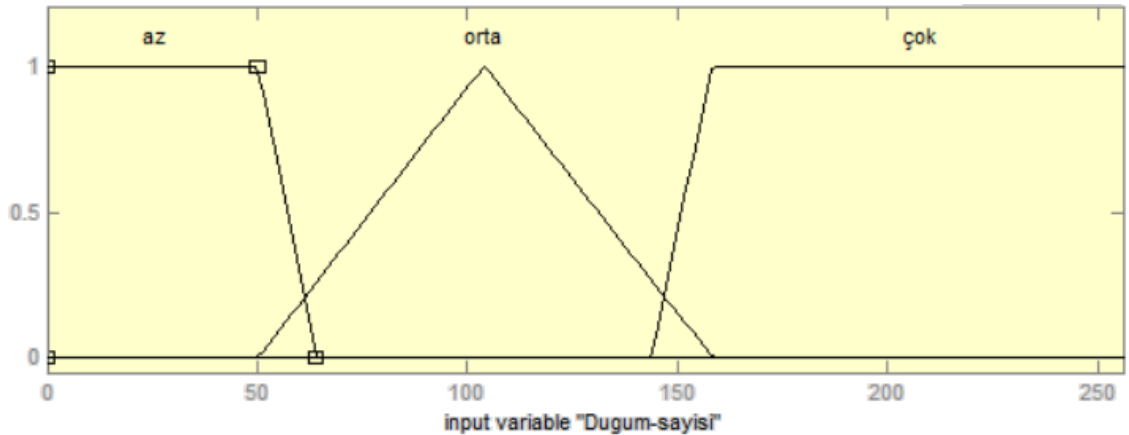
Şekil 3.10. Kullanılan toplam düğüm sayısına göre “enerji oran” değişimi grafiği.

Elde edilen grafiklere göre, ağda en az düğüm sayısı olduğu durumda DCT algoritmasının en iyi enerji oranına sahip olduğu, LEICA algoritmasının ise en kötü enerji oranına sahip olduğu görülmektedir. Ancak, düğüm sayısı arttıkça DCT algoritması etkililiğini kaybettiği SPIHT algoritmasının daha iyi duruma geldiği görülmektedir. Düğüm sayısının daha da fazla olduğu ağlarda ise, LEICA algoritmasının ön planda olduğu görülmektedir. Yani düğüm sayısının en az olduğu durumda DCT, biraz daha fazla olduğu durumda SPIHT, en fazla olduğu durumda ise LEICA algoritmasının daha verimli olduğu görülmektedir. Şekil 3.10’da görüldüğü üzere bazı algılayıcı düğüm sayı aralıklarında kullanılacak uygun görüntü sıkıştırma algoritmasına karar verilememektedir. Bu gibi durumlarda bulanık mantık yöntemi kullanılarak çözüm sağlanmıştır.

“Düğüm sayısı” parametresi değer olarak 0 ile 256 arasında değişmektedir. Bu giriş değişkeni için üç üyelik fonksiyonu oluşturularak “az”, “orta” ve “çok” olarak isimlendirilmiştir. Düğüm sayısı giriş değişkenine ait üyelik fonksiyonlarının sınırları Çizelge 3.4’te ve grafik görünümü ise Şekil 3.11’de verilmiştir.

Çizelge 3.4. “Düğüm sayısı” giriş değişkeni üyelik fonksiyonu sınırları.

Dilsel Değişken	Üyelik Fonksiyonu Sınırı	
Az	0	64
Orta	50	158
Çok	144	256



Şekil 3.11. “Düğüm sayısı” giriş değişkeni üyelik fonksiyonlarının grafik görünümü.

Bu çıkarımda sonuç olarak “düğüm sayısı” parametresinin “az ve orta” olduğu durumlarda kullanılacak görüntü sıkıştırma algoritmasının DCT veya SPIHT, “orta ve

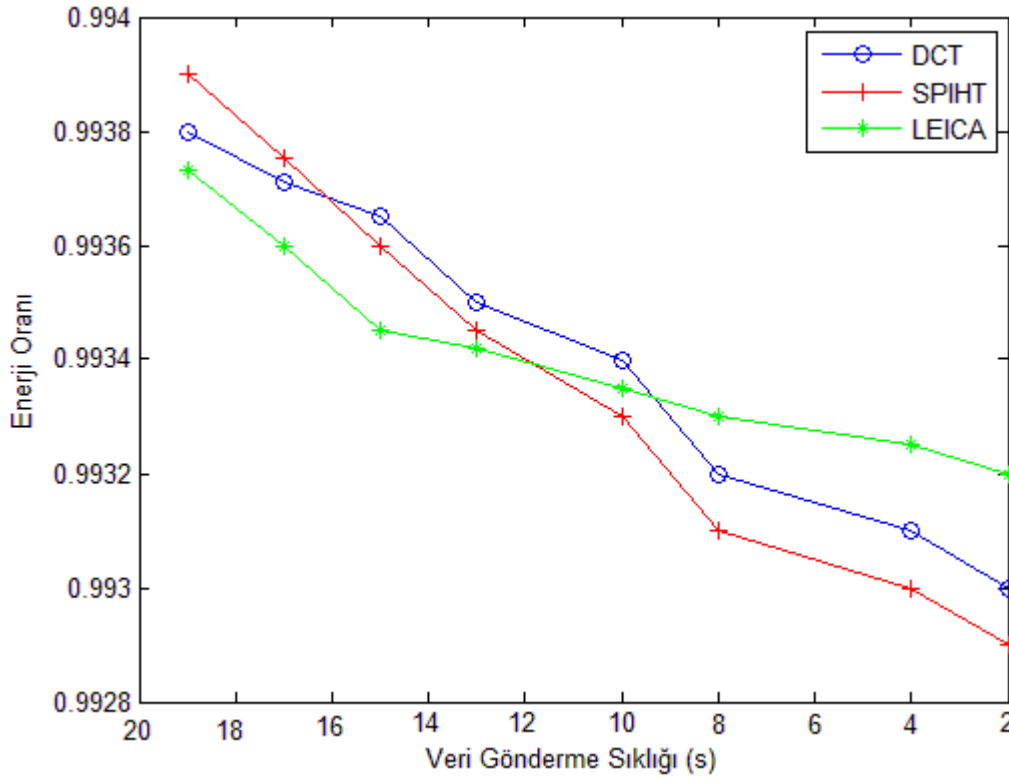
çok” olduğu durumlarda da SPIHT veya LEICA algoritmalarının kullanılacağı anlaşılmaktadır.

3.6.3 Ağ Üzerindeki Veri Gönderme Sıklığına Göre Kullanılacak Algoritmayı Belirleme

Gerçekleştirilen sistemlerde düğümler arasındaki veri alış verişi belirli zaman dilimlerinde yapılmaktadır. Bu zaman dilimi ağ üzerindeki veri yoğunluğunu da göstermekte olup, sistem kurulurken isteğe bağlı olarak belirlenmektedir. Ağ üzerindeki algılayıcıların birbirlerine veri gönderme sıklığı 2 saniye ile 19 saniye arasında değişmektedir. Gerçekleştirilen benzetim sonuçlarına göre, üç sıkıştırma algoritmasının ağdaki veri gönderme sıklığına göre “enerji oran” değişim değerleri (kalan enerji / toplam enerji) Çizelge 3.5’te, grafiği ise Şekil 3.12’de verilmektedir.

Çizelge 3.5. Veri gönderme sıklığına göre “enerji oran” değişimi değerleri.

Veri Gönderme Sıklığı (s)	Enerji Oranı (Kalan Enerji / Toplam Enerji)		
	DCT	SPIHT	LEICA
19	0,9938	0,9939	0,99373
17	0,99371	0,993753	0,9936
15	0,99365	0,9936	0,99345
13	0,9935	0,99345	0,99342
10	0,9934	0,9933	0,99335
8	0,9932	0,9931	0,9933
4	0,9931	0,993	0,99325
2	0,993	0,9929	0,9932



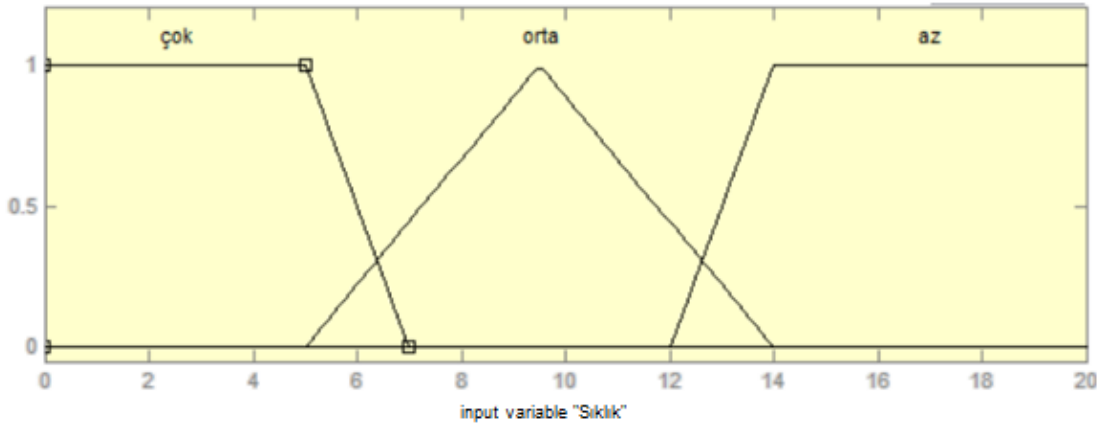
Şekil 3.12. Veri gönderme sıklığına göre “enerji oran” değişimi.

Elde edilen grafiklere göre, ağın en fazla yoğun olduğu durumda SPIHT algoritmasının verimli olduğu, ağın yoğunluğunun biraz azaldığı durumda DCT algoritmasının verimli olduğu ve ağın yoğunluğunun oldukça az olduğu durumda ise LEICA algoritmasının verimli olduğu görülmektedir. Ancak, diğer parametrelerde olduğu gibi bazı veri gönderim sıklık aralıklarında kullanılacak algoritmalarda çakışma söz konusu olmaktadır. Bu durumların üstesinden gelebilmek için bulanık mantık yöntemi kullanılmıştır.

“Sıklık” parametresi 0 ile 19 saniye arasında değişmiştir. Bu giriş değişkeni için üç üyelik fonksiyonu oluşturularak “az”, “orta” ve “çok” olarak isimlendirilmiştir. “Sıklık” giriş değişkenine ait üyelik fonksiyonlarının sınırları Çizelge 3.6’da ve grafik görünümü Şekil 3.13’te verilmiştir.

Çizelge 3.6. “Sıklık” giriş değişkeni üyelik fonksiyonu sınırları.

Dilsel Değişken	Üyelik Fonksiyonu Sınırı	
Az	12	20
Orta	5	14
Çok	0	7



Şekil 3.13. “Sıklık” giriş değişkeni üyelik fonksiyonlarının grafik görünümü.

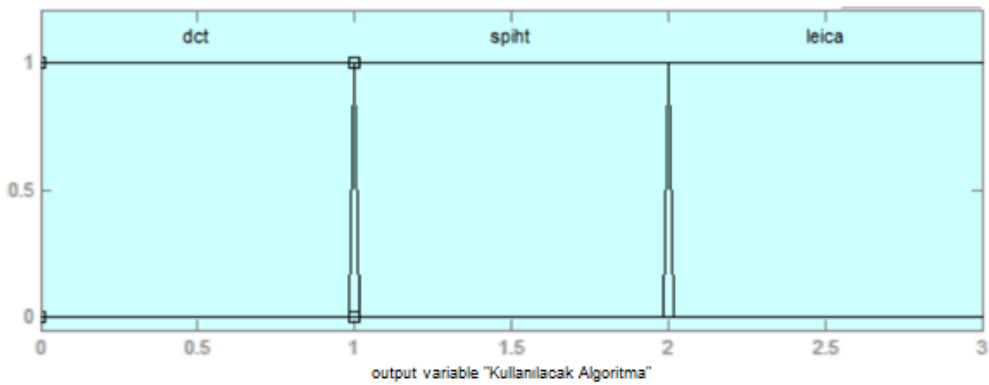
Bu çıkarımda sonuç olarak “sıklık” parametresinin “çok ve orta” olduğu zamanlarda kullanılacak görüntü sıkıştırma algoritmasının SPIHT veya DCT, “orta ve az” olduğu zamanlarda da DCT veya LEICA algoritmalarının kullanılacağı anlaşılmaktadır.

3.6.4 Görüntü Sıkıştırma Algoritması Seçimi

Yukarıda verilen üç giriş değişkeni kullanılarak “Kullanılacak Algoritma” değeri elde edilmektedir. Bu amaçla “Kullanılacak Algoritma” çıkış değişkeni için üç üyelik fonksiyonları oluşturulmuştur. Üyelik fonksiyonlarına yöntem sonucunda kullanılacak algoritma isimleri (DCT, SPIHT ve LEICA) dilsel değişken olarak verilmiştir. “Kullanılacak Algoritma” çıkış değişkenine ait üyelik fonksiyonlarının sınırları Çizelge 3.7’de ve grafik görünümü Şekil 3.14’de verilmiştir.

Çizelge 3.7. “Sonuç” çıkış değişkeni üyelik fonksiyonu sınırları.

Dilsel Değişken	Üyelik Fonksiyonu Sınırı	
DCT	0	1
SPIHT	1	2
LEICA	2	3



Şekil 3.14. “Kullanılacak Algoritma” çıkış değişkeni üyelik fonksiyonlarının grafik görünümü.

Oluşturulan üyelik fonksiyonları kullanılarak EK-A'da gerçekleştirilen karar ağacı yöntemi sayesinde 19 adet'e indirilen durumlar kullanılarak kurallar oluşturulup bulanıklaştırma işlemi yapılmıştır. Oluşturulan kurallar aşağıda verilmiştir.

*IF (Dugum-sayisi is az) AND (Sıklık is az) AND (Mesafe is az)
THEN (Kullanılacak Algoritma is dct)*

*IF (Dugum-sayisi is orta) AND (Sıklık is az) AND (Mesafe is az)
THEN (Kullanılacak Algoritma is spiht)*

*IF (Dugum-sayisi is çok) AND (Sıklık is az) AND (Mesafe is az)
THEN (Kullanılacak Algoritma is leica)*

*IF (Sıklık is az) AND (Mesafe is az)
THEN (Kullanılacak Algoritma is dct)*

*IF (Sıklık is çok) AND (Mesafe is az)
THEN (Kullanılacak Algoritma is leica)*

*IF (Dugum-sayisi is az) AND (Sıklık is az) AND (Mesafe is orta)
THEN (Kullanılacak Algoritma is leica)*

*IF (Dugum-sayisi is orta) AND (Sıklık is az) AND (Mesafe is orta)
THEN (Kullanılacak Algoritma is spiht)*

*IF (Dugum-sayisi is çok) AND (Sıklık is az) AND (Mesafe is orta)
THEN (Kullanılacak Algoritma is leica)*

*IF (Dugum-sayisi is az) AND (Sıklık is orta) AND (Mesafe is orta)
THEN (Kullanılacak Algoritma is dct)*

*IF (Dugum-sayisi is orta) AND (Sıklık is orta) AND (Mesafe is orta)
THEN (Kullanılacak Algoritma is dct)*

*IF (Dugum-sayisi is çok) AND (Sıklık is orta) AND (Mesafe is orta)
THEN (Kullanılacak Algoritma is leica)*

*IF (Sıklık is çok) AND (Sıklık is orta)
THEN (Kullanılacak Algoritma is leica)*

*IF (Sıklık is az) AND (Mesafe is çok)
THEN (Kullanılacak Algoritma is spiht)*

*IF (Dugum-sayisi is az) AND (Sıklık is orta) AND (Mesafe is çok)
THEN (Kullanılacak Algoritma is dct)*

*IF (Dugum-sayisi is orta) AND (Sıklık is orta) AND (Mesafe is çok)
THEN (Kullanılacak Algoritma is spiht)*

*IF (Dugum-sayisi is çok) AND (Sıklık is orta) AND (Mesafe is çok)
THEN (Kullanılacak Algoritma is spiht)*

*IF (Dugum-sayisi is az) AND (Sıklık is çok) AND (Mesafe is çok)
THEN (Kullanılacak Algoritma is spiht)*

*IF (Dugum-sayisi is orta) AND (Sıklık is çok) AND (Mesafe is çok)
THEN (Kullanılacak Algoritma is spiht)*

IF (Dugum-sayisi is cok) AND (Sıklık is cok) AND (Mesafe is cok)
THEN (Kullanılacak Algoritma is leica)

Durulaştırma (defuzzification) ile Kullanılacak Algoritma değerleri elde edilmiştir. Durulaştırma için Mamdani yöntemi kullanılmıştır.

Çizelge 3.8’de örnek giriş değerleri için bulanık mantık yöntemi uygulandığında elde edilen Kullanılacak Algoritma değerleri verilmiştir.

Çizelge 3.8. Örnek giriş değerleri için bulanık mantık yöntemi ile hesaplanmış “Kullanılacak Algoritma” değerleri.

Düğüm Sayısı	Veri Gönderme Sıklığı (saniye)	Düğüm Arası Mesafe (metre)	Çıkış	Kullanılacak Algoritma
20	10	100	0,495	DCT
120	3	56	2,51	LEICA
200	15	20	2,5	LEICA
75	2	190	1,5	SPIHT
16	19	10	0,495	DCT
95	5	175	1,5	SPIHT
256	18	20	2,5	LEICA
20	12	100	0,495	DCT
170	6	43	1,89	SPIHT

Örnek bir senaryo için Çizelge 3.8’de anlaşıldığı üzere belirlenen parametrelerden düğüm sayısının 200 adet, veri gönderme sıklığının 15 saniye, düğümler arası mesafenin de 20 metre olduğu durumda bulanık mantık yöntemi, LEICA algoritmasının en uygun algoritma olduğu karara varmıştır. Diğer bir senaryo için ise, düğüm sayısının 95 adet, veri gönderme sıklığının 5 saniye, düğümler arası mesafenin de 175 metre olduğu durumda bulanık mantık yöntemi, SPIHT algoritmasının en uygun algoritma olduğu karara varmıştır.

3.6.5 PSNR Değer Kaybının Korunması

Gerçekleştirilen sistemde, sıkıştırılmış görüntünün Tepe Sinyal Gürültü Oranı (Peak Signal to Noise Ratio – PSNR) hesaplanır. Orijinal görüntünün PSNR değeri ile sıkıştırılmış görüntünün PSNR değeri arasındaki fark, kablosuz iletişimde kabul edilebilir kayıp değerleri arasında ise görüntü bir sonraki algılayıcı düğüme veya baz istasyonuna aktarılması sağlanır. Orijinal görüntünün PSNR değeri ile sıkıştırılmış görüntünün PSNR değeri arasındaki fark, 20 dB’ye eşit veya küçük ise sıkıştırma oranı artırılarak tekrar sıkıştırma işlemi gerçekleştirilir ve görüntü bir sonraki algılayıcı düğüme veya baz istasyonuna aktarılır. Ancak, orijinal görüntünün PSNR değeri ile sıkıştırılmış görüntünün PSNR değeri arasındaki fark, 20 dB’den büyük ise sıkıştırma oranı azaltılarak tekrar sıkıştırma işlemi gerçekleştirilir ve görüntü bir sonraki algılayıcı düğüme veya baz istasyonuna aktarılır. Sıkıştırılmış görüntünün PSNR değeri, kablosuz iletişimde kabul edilebilir PSNR değerinde olması sağlanarak sistemin kalitesi korunmuş olacaktır.

5. bölümde geliştirilecek olan benzetimlerde ve bu bölümde yapılan PSNR değer kaybı çıkarımlarında aşağıdaki örnek görüntüler kullanılmıştır.

Kullanılan her bir algoritma için verilen “örnek görüntü 1” verisinin sıkıştırma oranına göre PSNR değer kaybı Şekil 3.18’de verilmiştir.



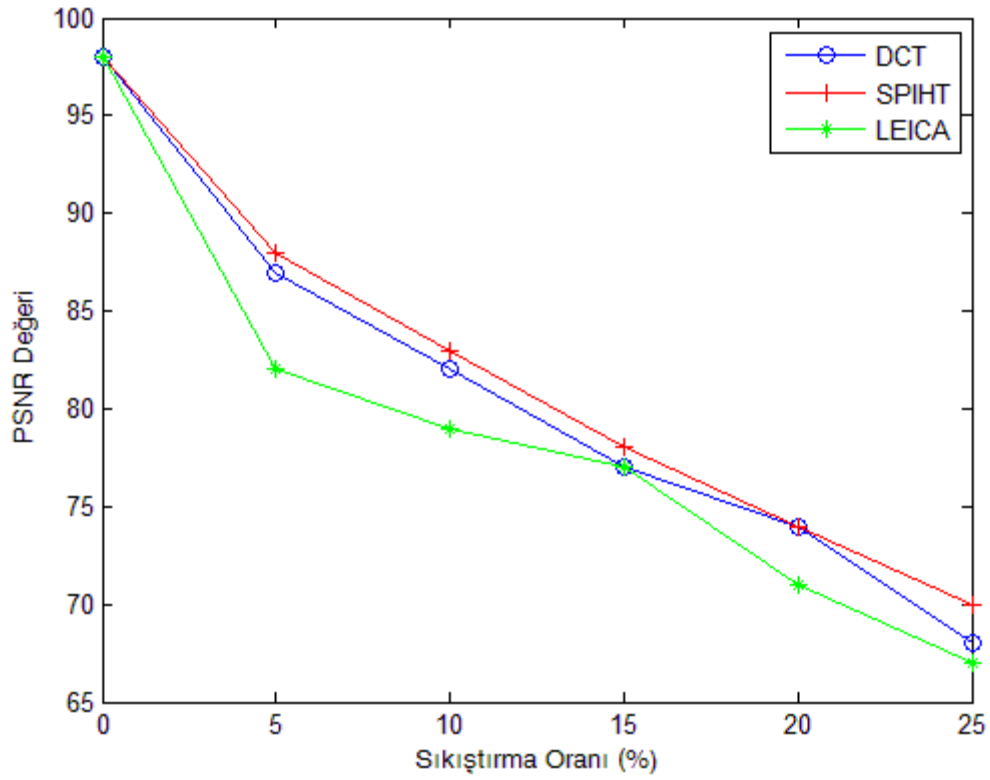
Şekil 3.15. Örnek görüntü 1.



Şekil 3.16. Örnek görüntü 2.



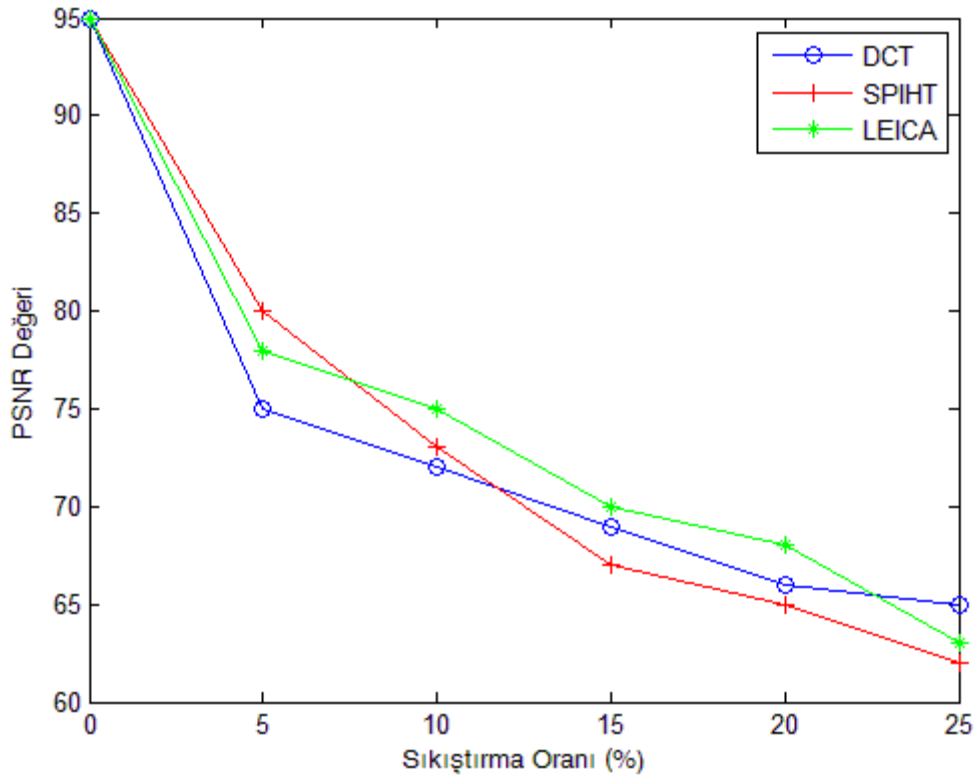
Şekil 3.17. Örnek görüntü 3.



Şekil 3.18. Üç algoritma için “örnek görüntü 1” verisinin sıkıştırma oranına göre PSNR değer kaybı grafiği.

Şekil 3.18’de “örnek görüntü 1” verisine en fazla yüzde 20 sıkıştırma oranı uygulanarak kablosuz iletişimde kabul edilebilir en iyi PSNR kayıp değerine ulaşmakta olduğu görülmektedir. Yani, görüntü sıkıştırılmadan 98 PSNR değerine sahipken yüzde 20 sıkıştırıldığında PSNR değerinin 74’e indiği görülmektedir. Bu durumda PSNR değer kaybı 24 dB olmaktadır. Bir sonraki sıkıştırma oranı olan yüzde 25’de PSNR değer farkı 28 dB olmakta ve bu değer kablosuz iletişimde kabul edilebilir kayıp değerinden fazladır. Dolayısıyla 24 dB kayıp PSNR değerine sahip olan yüzde 20 sıkıştırma oranı en uygun sıkıştırma oranı olduğu anlaşılmaktadır.

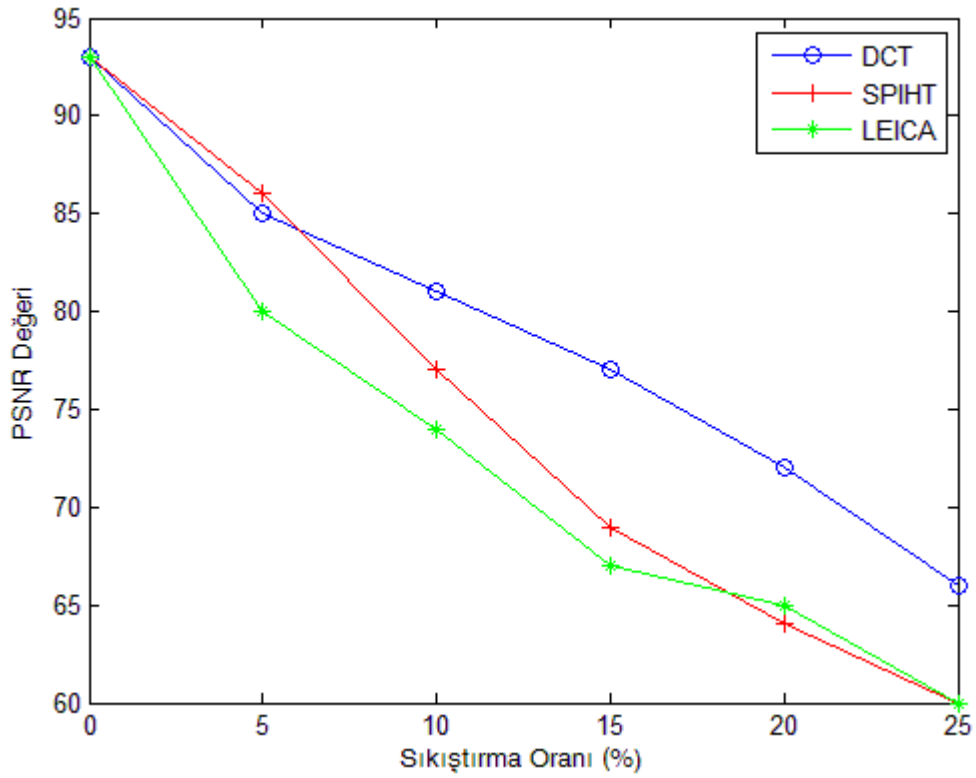
Kullanılan her bir algoritma için verilen “örnek görüntü 2” verisinin sıkıştırma oranına göre PSNR değer kaybı Şekil 3.19’da verilmiştir.



Şekil 3.19. Üç algoritma için “örnek görüntü 2” verisinin sıkıştırma oranına göre PSNR değer kaybı grafiği.

Şekil 3.19’da “örnek görüntü 2” verisine en fazla yüzde 10 sıkıştırma oranı uygulanarak kablosuz iletişimde kabul edilebilir en iyi PSNR kayıp değerine ulaşıldığı görülmektedir. Geliştirilen algoritmaya göre elde edilen en iyi sıkıştırma oranının bir sonraki oran incelendiğinde; sıkıştırma gerçekleşmeden elde edilen PSNR değeri 95 iken, yüzde 15 sıkıştırma oranı ile 69 olduğu görülmektedir. Bu değerler arasındaki fark 26 dB ile kablosuz iletişimde kabul edilebilir PSNR değer farkının üstünde olduğu görülmektedir. Şekil 3.19’da görüldüğü üzere yüzde 15 sıkıştırma oranının üzerindeki oranlarda ise PSNR değerinin daha da azaldığı görülmekte, dolayısıyla PSNR değer farkı daha da artmaktadır. Bu sebepten, bu görüntünün aktarımında görüntüye en fazla yüzde 10 sıkıştırma gerçekleştirilebilir.

Kullanılan her bir algoritma için verilen “örnek görüntü 3” verisinin sıkıştırma oranına göre PSNR değer kaybı Şekil 3.20’de verilmiştir.



Şekil 3.20. Üç algoritma için “örnek görüntü 3” verisinin sıkıştırma oranına göre PSNR değer kaybı grafiği.

Şekil 3.20’de “örnek görüntü 3” verisine en fazla yüzde 10 sıkıştırma oranı uygulanarak kablosuz iletişimde kabul edilebilir en iyi PSNR kayıp değerine ulaşılmaktadır. Yani, bu görüntünün aktarımında görüntüye en fazla yüzde 10 sıkıştırma uygulanabilir.

3.7 SONUÇ

Bu bölümde, GSD-EDUKA’nın çalışma adımları ve parametreleri ele alınmıştır. Belirlenen görüntü sıkıştırma algoritmaları için GSD-EDUKA’nın parametreleri ile çıkarımlar yapılarak, hangi algoritmanın hangi parametrede daha verimli olacağı sonucu elde edilmiştir. Elde edilen çıkarımların sayısı karar ağacı yöntemi kullanılarak daha az indirilmiş ve daha sonra bu çıkarımlar bulanık mantık yöntemi kullanılarak kural tabanlı bilgiler oluşturulup karar verme işlemi gerçekleştirilmiştir.

GSD-EDUKA bulanık mantık yöntemiyle üç sıkıştırma algoritmasından ortam koşullarına göre (veri gönderme sıklığı, ağdaki düğüm sayısı ve düğümler arası mesafe) en uygun olanını seçip, seçilen algoritma PSNR değerini kontrollü olarak kabul edilebilir oranda azaltmaya dayalı olarak geliştirilmiştir. GSD-EDUKA ile yeni bir

sıkıştırma algoritması önerilmeyip, uygun sıkıştırma algoritmasının görüntü kalitesini bozmadan daha yüksek oranda sıkıştırma uygulamaya, buna bağlı olarak taşınacak verinin boyutunun azaltılması sonucunda enerji tasarrufunu arttırmak hedeflenmektedir.



4. KMAA'LARDA ENERJİ TÜKETİMİ

4.1 ALGILAYICI DÜĞÜMLERİN ENERJİ TÜKETİMİ

Algılayıcı düğümler ayakta kalma enerjisi, gönderme enerjisi, alma enerjisi ve hesaplama enerjisi harcamaktadırlar. Bir düğümün toplam harcadığı enerji Denklem (4.1)'de verilmiştir [101].

$$E_{toplamlam} = E_a + E_t + E_r + E_c \quad (4.1)$$

Ayakta kalma enerjisi (E_a): Algılayıcı düğümlerin aktif olduğu ve kapsama alanındaki diğer düğümlerden haberdar olmak için harcadığı enerjidir [101]. Algılayıcı düğümlerin ayakta kalma enerjileri (E_a) bütün karşılaştırmalarda aynı olacağı için göz ardı edilmiştir.

Gönderme enerjisi (E_t): Algılayıcı düğümün bir veriyi herhangi bir mesafe uzaklıktaki başka bir düğüme göndermek için harcadığı enerjidir. Harcanan gönderme enerjisi, mesafeye ve veri paketinin büyüklüğüne göre değişir. Gönderme enerjisi için [102]'de önerilen modele göre, bir düğüm gönderme için $E_{elec} = 50 \text{ nj/bit}$ enerji gerektirir. Gönderici ile alıcı arasında 1-bit'in transfer edilebilmesi için E_{elec} enerjisi tüketilir. d düğümler arası mesafe olmak üzere d^2 enerji kaybı olduğu varsayıldığında, gönderici düğümün harcadığı enerji Denklem (4.2)'de verilen E_{ty} 'nin değeridir.

$$E_{ty} = E_{amp} * d^2 \quad (4.2)$$

Burada $E_{amp} = 100 \text{ pj/bit/m}^2$ 'dir. Böylece d mesafesinden 1-bit göndermek için Denklem (4.3)'de verilen (E_t) enerjisi harcanır [102].

$$E_t = E_{elec} + E_{amp} * d^2 \quad (4.3)$$

Alma Enerjisi (E_r): Alma enerjisi algılayıcı düğümün veriyi almak için harcadığı enerjidir. d mesafesinden 1-bit veriyi almak için harcanan enerji E_{elec} 'dir. Bu enerji

eşitliği Denklem (4.4)'te verilmiştir.

$$E_r = E_{elec} \quad (4.4)$$

Hesaplama Enerjisi (E_c): Algılayıcı düğüm içerisinde işlemci tarafından kullanılan herhangi bir işlem ve yönlendirme için harcanan enerjidir. Harcanan enerjinin değeri algılayıcının gerçekleştirdiği işleme bağlı olarak değişiklik göstermektedir.

4.2 GSD-EDUKA'YI KULLANAN ALGILAYICI DÜĞÜMÜN HARCADIĞI ENERJİ

Geliştirilen sistemdeki algılayıcı düğümler standart düğümlerden farklı olarak sahip oldukları GSD-EDUKA'yı kullanabilmeleri için konum bilgisi elde etmeleri gerekmektedir. Bunun için GPS (Global Positioning System) modülünü kullanmaları gerekir.

Geliştirilen sistemde GPS modülü enerjisi ve hesaplama enerjisinin fazladan olduğu düşünülebilir. Ancak geliştirilen GSD-EDUKA sayesinde yüksek boyutlu olan görüntü verisi sıkıştırılarak daha az boyuta indirgenmiş ve gönderme ile alma enerjilerinden oldukça tasarruf sağlanmış olacaktır. Bu durumda, GSD-EDUKA'nın kullanılabilmesi için harcanan GPS modülü enerjisi ve hesaplama enerjisi daha az olduğu görülmektedir. Denklem (4.1)'de ifade edilen toplam harcanan enerjiye GSD-EDUKA'nın ihtiyacı olduğu GPS modülünün, harcadığı enerjide eklenerek Denklem (4.5) elde edilmiştir. Burada (E_g), GPS modülünün harcadığı enerjidir.

$$E_{toplam} = E_a + E_t + E_r + E_c + E_g \quad (4.5)$$

4.2.1 Hesaplama Enerjisi

Hesaplama enerjisi (E_c), algılayıcı düğüm içerisinde kullanılan GSD-EDUKA ve yönlendirme için harcanan enerjidir. Yönlendirme işlemi bütün karşılaştırmalarda aynı olacağı için göz ardı edilmiştir. Algılayıcı düğümlerin kullandığı GSD-EDUKA'nın içerdiği algoritmalar (DCT, SPIHT ve LEICA) alınan görüntünün sıkıştırma işlem sürelerine ve algoritmanın işlem adım sayılarına göre harcadığı enerjiler hesaplanmaktadır. [103]'e göre kullanılan algılayıcı düğümlerin hesaplama esnasında harcadıkları toplam enerji eşitliği Denklem (4.6)'da verilmiştir.

$$E_c = V_{cc} * I_{cc} * (T_{top}/T_{is}) \quad (4.6)$$

Örnek bir algılayıcı düğüm için V_{cc} anma gerilimi ve I_{cc} anma akımı sırasıyla 1,65 V ve 95 mA verilmiştir [103]. T_{top} toplam işlem süresini, T_{is} 'de algoritmanın işlem sayısını göstermektedir. Kullanılacak olan algoritmalarından bu değerler alınarak harcanan enerji hesaplanır. Çizelge 4.1'de her bir algoritmanın toplam işlem süresi ve adım sayısı yer almaktadır.

Çizelge 4.1. Algoritmaların işlem süreleri ve adım sayıları.

	Sıkıştırılmış Görüntü Boyutu	Sıkıştırma Süresi	İşlem Sayısı
DCT	21 KB	54,843 ms	1000
SPIHT	16 KB	57,968 ms	800
LEICA	19 KB	54,707 ms	900

Bit başına harcadığı enerji değeri Denklem (4.7) ile hesaplanmaktadır. Denklem (4.8)'de ise birim eşitliği verilmiştir.

$$E_c = (V_{cc} * I_{cc} * (T_{top}/T_{is})) / \text{Bit Sayısı} \quad (4.7)$$

$$\text{Joule} = \text{Volt} * \text{Amper} * \text{Saniye} \quad (4.8)$$

[103]'ten elde edilen V_{cc} anma gerilimi ve I_{cc} anma akımı ile Denklem (4.6) sağlanarak DCT, SPIHT ve LEICA algoritmaları için harcanan enerji değerleri sırasıyla Denklem (4.9), Denklem (4.10) ve Denklem (4.11)'de verilmiştir.

$$E_{C_DCT} = 1,65 * 95 * 10^{-3} * (0,054843/1000) = 8,59 \mu\text{joule} \quad (4.9)$$

$$E_{C_SPIHT} = 1,65 * 95 * 10^{-3} * (0,057968/800) = 11,35 \mu\text{joule} \quad (4.10)$$

$$E_{C_LEICA} = 1,65 * 95 * 10^{-3} * (0,054707/900) = 9,5 \mu\text{joule} \quad (4.11)$$

GSD-EDUKA'nın karar verme süresi 1,6 µsaniye ve işlem sayısı ise 19'dur. Karar verme işlemi için harcadığı enerji Denklem (4.12)'de verilmektedir.

$$E_{Karar} = 1,65 * 95 * 10^{-3} * (0,00016/19) = 1,32 \mu\text{joule} \quad (4.12)$$

4.2.2 Gönderme – Alma Enerjileri

GSD-EDUKA’da kullanılan algoritmalarının görüntü sıkıştırma performans değerleri Çizelge 4.2’de verilmiştir. Elde edilen sıkıştırma değerlerine göre örnek alınan algılayıcı düğüme bağlı olarak değişen gönderme ve alma enerjileri Denklem (4.3) ve Denklem (4.4)’te verilen eşitlikler ile hesaplanmaktadır.

Çizelge 4.2. Kullanılan algoritmaların görüntü sıkıştırma performansları.

	Orijinal Görüntü Boyutu	Sıkıştırılmış Görüntü Boyutu
DCT	256 KB	21 KB
SPIHT	256 KB	16 KB
LEICA	256 KB	19 KB

Bu durumda 256 KB’lik bir görüntü DCT algoritması kullanılarak sıkıştırıldığında 21 KB’ye, SPIHT kullanılarak sıkıştırıldığında 16 KB’ye ve LEICA kullanılarak sıkıştırıldığında 19 KB’ye indirilmektedir. Denklem (4.3)’e bağlı olarak sıkıştırılmamış görüntünün ve her bir algoritma kullanılarak sıkıştırılan görüntünün gönderme enerjileri aşağıda verilmiştir;

$$\begin{aligned} E_{t_{orjinal}} &= 50 * 10^{-9} + 100 * 10^{-12} * d^2 * bit \\ &= 10^{-9}(50 + 100 * 10^{-3} * d^2 * 256 * 10^3 * 8) \\ &= 204,8 * d^2 \mu joule \end{aligned} \quad (4.13)$$

$$\begin{aligned} E_{t_{DCT}} &= 50 * 10^{-9} + 100 * 10^{-12} * d^2 * bit \\ &= 10^{-9}(50 + 100 * 10^{-3} * d^2 * 21 * 10^3 * 8) \\ &= 16,85 * d^2 \mu joule \end{aligned} \quad (4.14)$$

$$\begin{aligned} E_{t_{SPIHT}} &= 50 * 10^{-9} + 100 * 10^{-12} * d^2 * bit \\ &= 10^{-9}(50 + 100 * 10^{-3} * d^2 * 16 * 10^3 * 8) \\ &= 12,85 * d^2 \mu joule \end{aligned} \quad (4.15)$$

$$\begin{aligned}
E_{t_{LEICA}} &= 50 * 10^{-9} + 100 * 10^{-12} * d^2 * bit \\
&= 10^{-9}(50 + 100 * 10^{-3} * d^2 * 19 * 10^3 * 8) \\
&= 15,25 * d^2 \mu joule
\end{aligned} \tag{4.16}$$

Aynı durumlarda alma enerjisi ise Denklem (4.4)'e bağılı olarak hesaplanmaktadır.

$$E_{r_{orjinal}} = 50 * 10^{-9} * 256 * 10^3 * 8 = 102,4 \mu joule \tag{4.17}$$

$$E_{r_{DCT}} = 50 * 10^{-9} * 21 * 10^3 * 8 = 8,4 \mu joule \tag{4.18}$$

$$E_{r_{SPIHT}} = 50 * 10^{-9} * 16 * 10^3 * 8 = 6,4 \mu joule \tag{4.19}$$

$$E_{r_{LEICA}} = 50 * 10^{-9} * 19 * 10^3 * 8 = 7,6 \mu joule \tag{4.20}$$

4.2.3 GPS Enerjisi

GPS enerjisi (E_g), algılayıcı düğümlerin konumlarını belirlenmesi için kullanılan GPS modülünün harcadığı enerjidir. Algılayıcı düğümlerde bulunan GPS'ler yer belirleme anında 70 mW, bekleme anında ise 10 mW enerji harcarlar [104]. Bu değerleri saniye de harcanan joule değerine çevirecek olursak, Denklem (4.22)'de yer belirleme anında harcanan enerji, Denklem (4.23)'te de bekleme anında harcanan enerji elde edilmiş olur.

$$watt = joule / saniye \tag{4.21}$$

$$E_k = 70 mW = 70 * 10^{-3} W = 70 \mu joule / saniye \tag{4.22}$$

$$E_b = 10 mW = 10 * 10^{-3} W = 10 \mu joule / saniye \tag{4.23}$$

Bu durumda GSD-EDUKA'da GPS modülünün harcadığı enerjiyi incelersek; örneğin 10 saniyede bir defa çalışan GPS modülü için, 1 saniyede bir defa konum belirleme işlemi yaparak kalan 9 saniyede de bekleme konumunda kalmaktadır. Bu durumda toplamda GPS modülünün harcadığı enerji Denklem (4.24)'te verilmiştir.

$$E_g = \frac{E_k * 1 + E_b * 9}{10} = 16 \mu joule \tag{4.24}$$

4.3 SONUÇ

Bu bölümde, KMAA’larda kullanılan algılayıcı düğümlerin standart olarak harcadığı enerjilere değinilmiştir. Daha sonra, GSD-EDUKA’yı kullanan algılayıcı düğümlerin fazladan harcadığı görüntü sıkıştırma algoritmanın hesaplama enerjisi ve GPS modülünün harcadığı enerji hesaplanmıştır.



5. GSD-EDUKA’NIN PERFORMANS DEĞERLENDİRMESİ

Bu bölümde, öncelikle KMAA’larda performans değerlendirme ölçütleri üzerinde durulmuştur. Daha sonra geliştirilen GSD-EDUKA’nın performans değerlendirmesini yapabilmek amacıyla Network Simulator – 2 (NS - 2)’de oluşturulan KMAA için farklı senaryolar üretilmiş ve geliştirilen yeni yöntem mevcut yöntemlerle karşılaştırılmıştır.

5.1 KMAA’LARDA PERFORMANS KRİTERLERİ

KMAA’ların bellek, işlemci, enerji tüketimi ve yaşam süresi gibi kısıtları vardır. Bellek ve işlemci kısıtı donanımsal problemler olduğu için ancak ilerleyen teknoloji ile giderilebilir. Enerji tüketimini iyileştirmek ve dolayısıyla yaşam süresini uzatmak ise yapılacak çalışmalar ile mümkündür [105], [106], [107].

KMAA’larda en önemli performans kriteri enerji tüketimidir. Multimedya verilerin iletimi için daha fazla enerjiye ve bant genişliğine ihtiyaç duyulmaktadır [1], [60], [108]. KMAA’lar için gerçekleşen birçok çalışmada [55], [73], [81], [83], [84] ve [109] enerji verimliliği için yöntemler geliştirilerek bu yöntemlerin enerji tüketim hesabı yapılmıştır. Yani, KMAA’ların yaşam süresini uzatma ile ilgili geliştirilen yöntemlerde hususiyet ile enerji tüketimi üzerinde durulmuştur. Enerji tüketim hesabında gönderilen paket sayısı, düğümler arası mesafe ve paketin büyüklüğü etkili olmaktadır [1], [73], [76], [110].

Bu çalışmada önerilen yöntemin performansı aşağıdaki kriterler açısından değerlendirilmiştir:

1. Enerji tüketimi: Gerçekleştirilen benzetimlerde, enerji tüketimi üç değişkene göre değerlendirilmiştir. Bu değişkenler; düğümler arası mesafe, düğüm sayısı ve veri gönderme sıklığıdır. Veri iletimi sonrasında kalan enerji, kullanılan görüntü sıkıştırma algoritmalarının performansları ile orantılıdır.
2. PSNR kayıp değeri: Yapılan benzetimlerde sıkıştırılmış olan görüntünün PSNR değeri ölçülmüş ve orijinal görüntünün PSNR değerinin 20 ile 25 dB arasında değer kaybetmesi durumunda görüntü aktarma işlemi gerçekleşmiştir. Bu değer

KMAA'lar için kabul edilebilir kayıp ölçütüdür [68], [69]. Ayrıca, elde edilen orijinal görüntünün PSNR değeri kaybı ile beraber sıkıştırma oranının artması ile iletilecek olan verinin boyutu azalacaktır. Dolayısıyla, algılayıcı düğümün daha az gönderme ve alma enerjisi harcamasını sağlayacaktır. Elde edilen sonuçlar Bölüm 5.3'te yer almaktadır.

5.2 GSD-EDUKA'NIN PERFORMANS DEĞERLENDİRME İŞLEM ADIMLARI

Bu tez çalışmasında, geliştirilen GSD-EDUKA'nın performans değerlendirmesini yapabilmek amacıyla, benzetim aracı olan Network Simulator – 2 (NS - 2) üzerinde örnek ağlar oluşturulmuştur.

5.2.1 Performans Değerlendirme Ortamı

Performans değerlendirme ortamını sabit ve hareketli düğümlerin bulunduğu iki farklı kablosuz ağ yapısı oluşturmaktadır.

Sabit düğümlerin bulunduğu yapıda, algılayıcı düğümlerin yerleşimleri için ızgara modeli kullanılmıştır. KMAA'larda yıldız topolojisi, iki katmanlı hiyerarşik küme topolojisi ve ızgara topolojisi gibi birçok topoloji kullanılmaktadır [111], [112], [110]. Izgara modeli belirli bir alanı kapsamak için kullanılan ağ sistemlerinde en uygun modeldir. Ayrıca, düğümler arasında birden fazla yol bulunması mümkündür. Sık kullanılan düğümün enerjisi bittiği veya azaldığı zaman başka düğüm kullanılarak ağın esnekliği artırılmış olacaktır [113], [114]. Literatürde yer alan çalışmaların birçoğunda enerji verimliliği açısından uygun olduğu için ızgara modeli seçilmiştir [112], [115].

Hareketli düğümlerin bulunduğu yapıyı ise, öncelikle ızgara modeli ile yerleştirilmiş ve daha sonra rastgele 5 m/s ve 10 m/s hız ile hareket eden algılayıcı düğümler oluşturmaktadır. Kablosuz ağdaki düğüm sayısı ve düğümlerin dağıtılacağı ortamın büyüklüğü değiştirilebilmektedir. Bu sayede Bölüm 3.5'te verilen GSD-EDUKA'nın parametrelerinden, düğüm sayısı (ağın büyüklüğü) sağlanmış olmaktadır.

Benzetim süresi için literatürdeki çalışmalarda genellikle 70 veya 100 saniye değerlerinin kullanıldığı görülmüştür. Benzetimde kullanılan toplam algılayıcı düğüm sayısı ise 100 ila 300 arasında değişmektedir [116], [117]. Gerçekleştirilen benzetimde kullanılan parametreler ayrıntılarıyla Bölüm 5.3.1'de verilmiştir.

5.2.2 GSD-EDUKA'nın Parametreleri

GSD-EDUKA'nın Bölüm 3'te verilen adımlarına göre, “düğümler arası mesafe”, “düğüm sayısı” ve “veri gönderme sıklığı” algılayıcı düğümlerin kullanacağı en verimli görüntü sıkıştırma algoritmasına karar verilebilmesi için gerekli olan parametrelerdir. Düğümler arası mesafeler, geliştirilen M-AODV yönlendirme protokolü tarafından hesaplanmaktadır. Birbiri ile iletişime geçebilen iki düğüm, yönlendirme protokolünün “hello” paketi sayesinde birbirlerinin koordinat bilgilerine ulaşırlar ve Denklem (3.1) sayesinde aralarındaki mesafeyi hesaplarlar. Düğüm sayısı ve veri gönderme sıklığı senaryoların kurulumunda belirlenerek giriş sağlanmaktadır. Elde edilen veriler, algılayıcı düğümlerin kullanacakları görüntü sıkıştırma algoritmasına karar vermesinde yardımcı olmaktadır.

5.2.3 Karar Ağacı Yöntemi'nin Kullanılması

Bölüm 3.5'te belirlenen GSD-EDUKA'nın üç parametresinin her birinin içerdiği üç durum (az, orta ve çok) göz önünde bulundurularak 27 durum elde edilmektedir. Kullanılacak görüntü sıkıştırma algoritmasını seçme işlemi için karar ağacı yöntemi kullanılarak elde edilen 27 durum 19'a indirgenmiştir. Bu sayede GSD-EDUKA'nın kullanılacak en verimli görüntü sıkıştırma algoritmasına karar vermesini sağlayan bulanık mantık yöntemi için oluşturulacak kurallar daha aza indirgenmiştir. Dolayısıyla GSD-EDUKA'nın işlem yükü ve süresi azaltılmış olacaktır. Karar ağacı yönteminin uygulanması ve gerçekleştirilen matematiksel işlemler Ek A'da verilmektedir.

5.2.4 Bulanık Mantık Yöntemi'nin Kullanılması

Düğümlerin GSD-EDUKA sayesinde kullanılacak algorithmaya karar vermesi bulanık mantık yöntemi aracılığıyla gerçekleştirilmektedir. Bu amaçla MATLAB 2014a'nın Fuzzy Logic Toolbox kullanılarak *kullanilacak_algoritma.fis* bulanık mantık çıkarım mekanizması oluşturulmuştur.

Bulanık mantığın giriş değişkenleri Bölüm 3'te verilen GSD-EDUKA'nın parametreleridir. Çıkış değişkeni ise kullanılmasına karar verilecek olan görüntü sıkıştırma algoritmasıdır.

5.3 PERFORMANS DEĞERLENDİRME

Geliştirilen GSD-EDUKA Bölüm 5.1’de verilen değerlendirme kriterlerinden enerji verimliliği (enerji oranı) ve paket ulaştırma oranı açısından performans ölçümleri yapılmıştır. Farklı senaryolar geliştirilip, Bölüm 3.5’te verilen GSD-EDUKA’nın parametrelerine farklı değerler verilerek sonuçlar alınmıştır. Ayrıca, alınan görüntünün PSNR değerinin farklı olduğu durumlarda da enerji oranı değerleri hesabı yapılmıştır.

Enerji oranı hesabı yaparken, sıkıştırılarak boyutu azalan görüntünün baz istasyonuna iletilene kadar taşıyıcı görevi yapan algılayıcı düğümlerin gönderme ve alma enerjileri sıkıştırılmış boyut üzerinden gerçekleşmektedir. Yani ağ içerisinde bir düğüm önce görüntüyü çeker, GSD-EDUKA sayesinde uygun görüntü sıkıştırma algoritmasına karar verir ve görüntüye uygulanır. Daha sonra yönlendirme protokolüne bağlı olarak bir sonraki algılayıcı düğüme görüntüyü iletir. Sıkıştırılmış görüntüyü alan komşu düğüm, sadece taşıyıcı görevi yaparak bir sonraki algılayıcı düğüme görüntüyü iletir. Bu esnada taşıyıcı düğümler sadece alma ve gönderme enerjisi harcarlar, hesaplama enerjisi harcamazlar.

Performans değerlendirmesi için örnek farklı senaryolar oluşturularak algılayıcı düğümlerin aldığı görüntüyü aşağıdaki durumlar gerçekleştirilerek komşu düğüme iletilmesi sağlanmıştır. Durumlar;

- Sıkıştırma işlemi yapmadan,
- Sadece DCT algoritmasını kullanarak,
- Sadece SPIHT algoritmasını kullanarak,
- Sadece LEICA algoritmasını kullanarak ve
- GSD-EDUKA kullanarak.

Bu durumlarda enerji oranı (kalan enerji / toplam enerji), alınan görüntünün PSNR değer kaybı ve harcanan toplam enerji değerleri için Çizelge 5.1’de verilen senaryolardan rastgele senaryolar seçilerek sonuçlar elde edilmiştir.

Öncelikle sabit ağlar için senaryolar geliştirilmiş ve benzetimler gerçekleştirilmiştir. Daha sonra algılayıcı düğümlerin 5 m/s ve 10 m/s hız ile hareket ettiği ağlarda oluşturulan senaryolar ile benzetimler gerçekleştirilmiştir.

Çizelge 5.1. Gerçekleştirilen senaryolar.

Senaryo	Düğümler Arası Mesafe (metre)	Düğüm Sayısı	Veri Gönderme Sıklığı (saniye)
Senaryo 1	Az	Az	Az-Orta-Çok
Senaryo 2	Az	Orta	Az-Orta-Çok
Senaryo 3	Az	Çok	Az-Orta-Çok
Senaryo 4	Orta	Az	Az-Orta-Çok
Senaryo 5	Orta	Orta	Az-Orta-Çok
Senaryo 6	Orta	Çok	Az-Orta-Çok
Senaryo 7	Çok	Az	Az-Orta-Çok
Senaryo 8	Çok	Orta	Az-Orta-Çok
Senaryo 9	Çok	Çok	Az-Orta-Çok
Senaryo 10	Az	Az-Orta-Çok	Az
Senaryo 11	Orta	Az-Orta-Çok	Az
Senaryo 12	Çok	Az-Orta-Çok	Az
Senaryo 13	Az	Az-Orta-Çok	Orta
Senaryo 14	Orta	Az-Orta-Çok	Orta
Senaryo 15	Çok	Az-Orta-Çok	Orta
Senaryo 16	Az	Az-Orta-Çok	Çok
Senaryo 17	Orta	Az-Orta-Çok	Çok
Senaryo 18	Çok	Az-Orta-Çok	Çok
Senaryo 19	Az-Orta-Çok	Az	Az
Senaryo 20	Az-Orta-Çok	Az	Orta
Senaryo 21	Az-Orta-Çok	Az	Çok
Senaryo 22	Az-Orta-Çok	Orta	Az
Senaryo 23	Az-Orta-Çok	Orta	Orta
Senaryo 24	Az-Orta-Çok	Orta	Çok
Senaryo 25	Az-Orta-Çok	Çok	Az
Senaryo 26	Az-Orta-Çok	Çok	Orta
Senaryo 27	Az-Orta-Çok	Çok	Çok

5.3.1 Benzetim Parametreleri

Düğüm sayısı: Benzetim için 17, 101 ve 197 algılayıcı düğüm kullanılmıştır. Bu düğümlerden biri baz istasyonudur. Hareketli ağlarda gerçekleştirilen benzetimlerde algılayıcı düğümlerin kapsam dışına çıkması durumunda düğüm sayısının azalması kaçınılmazdır.

Düğümlerin yerleştirilmesi: Sabit ağ benzetiminde ızgara modeli kullanılmıştır. Hareketli ağ benzetiminde de başlangıç olarak ızgara modeli ile algılayıcı düğümler yerleştirilmiş ve daha sonra algılayıcı düğümlerin rastgele hareket etmesi sağlanmıştır.

Düğümler arası mesafe: Algılayıcı düğümlerin arasındaki mesafe gerçekleştirilen benzetime göre 10 ve 200 metre arasında değişiklik göstermektedir.

Ortam boyutu: Sabit ağlarda oluşturulan ızgara modeline bağlı olarak algılayıcı düğümler yerleştirilmiştir ve belirlenen “düğüm arası mesafe” değişkenine göre ortamın boyutu oluşturulmuştur. Değiştirilebilen “düğüm arası mesafe” parametresi ile ortamın boyutu da değişmektedir. Hareketli ağlarda ise, algılayıcı düğümler için rastgele dağılım kullanılarak yatay ve düşeyde algılayıcı düğüm başına 5000 m² düşecek şekilde dağılım kullanılmıştır.

Paket boyutu: 256 Kbyte’lık sabit boyutlarda veri paketleri kullanılmıştır. Kullanılan algılayıcı düğümlere entegre edilen kameranın özelliğinden ötürü 256 Kbyte kullanılmıştır. Ancak kodlama içerisinden bu değer girdi olarak belirlenerek, gerektiği durumda değişiklik gösterebilmesi sağlanmıştır.

Yayılım mesafesi: Düğümlerin kapsama alanı 200 metre yarıçapında dairedir.

Benzetim süresi: Benzetim modelleri 120 saniye çalıştırılmaktadır.

Veri gönderme sıklığı: 3, 9 ve 15 saniye aralıklar ile veri gönderme işlemi gerçekleştirilmiştir.

Tekrar sayısı: Benzetim yazılımı ile ele alınan her farklı ağ 10 defa çalıştırılıp, ortalama sonuçlar alınmıştır.

Kullanılan görüntüler: Gerçekleştirilen benzetimlerde Şekil 3.15 (örnek görüntü 1), Şekil 3.16 (örnek görüntü 2) ve Şekil 3.17 (örnek görüntü 3) verilen görüntüler kullanılmıştır.

5.3.2 GSD-EDUKA’nın PSNR Değer Kaybına Göre Diğer Sıkıştırma Algoritmalarıyla Karşılaştırılması

Kablosuz iletişimde hat boyunca taşınan veri boyutunu mümkün olduğunca küçültürken harcanan enerji miktarını en aza indirmek hedeflenmektedir. Amaç, taşınacak veri boyutunu küçültürken görüntünün PSNR değerini de kabul edilebilir aralıkta tutabilmektir. Böylelikle ağın enerjisi tasarruflu olarak kullanılırken, sistemin kalitesi (teslim edilen verinin kalitesi) de korunmuş olmaktadır. Çizelge 5.1’de belirlenen senaryolardan örnek olarak senaryo 1, 14 ve 22 gerçekleştirilmiştir.

Senaryo 1’deki duruma göre mesafe az, düğüm sayısı az ve veri gönderme sıklığı değişken olan GSD-EDUKA ile (%5 sıkıştırma oranına sahip) DCT, SPIHT ve LEICA için PSNR değerleri ve sıkıştırma oranları Çizelge 5.2’de verilmiştir. Çizelgede, S.O. sıkıştırma oranını ifade etmektedir ve benzetim görüntüsü olarak “örnek görüntü 1” kullanılmıştır.

Çizelge 5.2. Senaryo 1: Mesafe-az ve düğüm sayısı-az olduğu durumda kullanılan veri gönderme sıklığına göre sıkıştırma oranı ve PSNR değerleri.

Veri Gönderme Sıklığı (sn)	DCT		SPIHT		LEICA		GSD-EDUKA		Sıkıştırma Uygulamadan	
	S.O.	PSNR	S.O.	PSNR	S.O.	PSNR	S.O.	PSNR	S.O.	PSNR
15	5	87	5	88	5	82	20	74	0	98
9	5	87	5	88	5	82	10	81	0	98
3	5	87	5	88	5	82	15	77	0	98

Gerçekleşen benzetimde veri gönderme sıklığının 15 saniye olduğu durumda DCT algoritması kullanılmış, sıkıştırma oranı ilk durumda %5 olarak belirlenmiş ve bu oranın kablosuz iletişimde kabul edilebilir minimum PSNR değerini sağlamadığı için sıkıştırma oranı seviyeli bir şekilde arttırılarak %20 olarak kullanılmış ve bu durumda minimum PSNR değerinin sağlandığı görülmüştür. Veri gönderme sıklığının 9 saniye olduğu durumda da DCT algoritması kullanılmış, sıkıştırma oranı olarak ilk aşamada alınan %5 değeri kabul edilebilir minimum PSNR değerini sağlamadığı için sıkıştırma oranı arttırılarak %10 olarak kullanılmış ve bu durumda minimum PSNR değerinin sağlandığı görülmüştür. Aynı şekilde veri gönderme sıklığının 3 saniye olduğu durumda ise LEICA algoritması kullanılarak %15’lik sıkıştırma oranı ile minimum kabul edilebilir PSNR değerine ulaşıldığı görülmektedir.

Senaryo 14’deki duruma göre düğüm sayısı değişken, düğümler arası mesafe orta ve veri gönderme sıklığı orta olan GSD-EDUKA ile standart kullanılan (%5 sıkıştırma oranına sahip) DCT, SPIHT ve LEICA için PSNR değerleri ve sıkıştırma oranları Çizelge 5.3’te verilmiştir. Benzetim görüntüsü olarak “örnek görüntü 2” kullanılmıştır.

Çizelge 5.3. Senaryo 14: Düğümler arası mesafe-orta ve veri gönderme sıklığı-orta olduğu durumda düğüm sayısına göre kullanılan sıkıştırma oranı ve PSNR değerleri.

Düğüm Sayısı	DCT		SPIHT		LEICA		GSD-EDUKA		Sıkıştırma Uygulamadan	
	S.O.	PSNR	S.O.	PSNR	S.O.	PSNR	S.O.	PSNR	S.O.	PSNR
17	5	75	5	80	5	78	10	72	0	95
101	5	75	5	80	5	78	10	70	0	95
197	5	75	5	80	5	78	5	78	0	95

Senaryo 22'deki duruma göre mesafe değişken, düğüm sayısı orta ve veri gönderme sıklığı az olan GSD-EDUKA ile standart kullanılan (%5 sıkıştırma oranına sahip) DCT, SPIHT ve LEICA için PSNR değerleri ve sıkıştırma oranları Çizelge 5.4'te verilmiştir. Benzetim görüntüsü olarak “örnek görüntü 3” kullanılmıştır.

Çizelge 5.4. Senaryo 22: Düğüm sayısı-orta ve veri gönderme sıklığı-az olduğu durumda düğümler arası mesafeye göre kullanılan sıkıştırma oranı ve PSNR değerleri.

Düğümler Arası Mesafe	DCT		SPIHT		LEICA		GSD-EDUKA		Sıkıştırma Uygulamadan	
	S.O.	PSNR	S.O.	PSNR	S.O.	PSNR	S.O.	PSNR	S.O.	PSNR
40	5	85	5	86	5	80	10	72	0	93
90	5	85	5	86	5	80	15	69	0	93
140	5	85	5	86	5	80	10	74	0	93

Gerçekleştirilen örnek senaryolar sonucunda elde edilen Çizelge 5.2, Çizelge 5.3 ve Çizelge 5.4'te görüldüğü üzere GSD-EDUKA, kablosuz iletişimde görüntü aktarımının kabul edilebilir PSNR kayıp değerini koruyarak en fazla sıkıştırma işlemin sağladığı görülmektedir.

5.3.3 Enerji Oranı

Oluşturulan senaryolarda her bir algılayıcı düğüme başlangıç enerjisi olarak 1000 joule atanmıştır. Ağın içerdiği düğüm sayısına göre benzetim yapılmadan önce toplam enerji belirlenmiştir. Benzetim süresi sona erdiğinde kalan enerji hesaplanmıştır. Kalan enerjinin toplam enerjiye oranı “enerji oranı” olarak ifade edilmiştir. Enerji oranı, gerçekleştirilen benzetimin enerji verimliliğini belirleyen bir performans ölçüm kriteridir. Enerji oranı yüksek olan durumlarda enerji verimliliğinin iyi seviyede olduğu saptanmaktadır.

Çizelge 5.1’de belirtilen tüm senaryolar sabit ağlar ve hareketli ağlar için gerçekleştirilmiş ve enerji oranı değeri elde edilmiştir. Enerji oranı değerleri sırasıyla “düğüm arası mesafe”, “düğüm sayısı” ve “veri gönderme sıklığı” parametrelerinin farklı kombinasyonları ile gerçekleştirilmiştir. Değişen parametreye karşılık diğer iki parametre sabit kalmaktadır. Ayrıca, gerçekleştirilen benzetimlerde rastgele olarak farklı görüntüler kullanılmıştır.

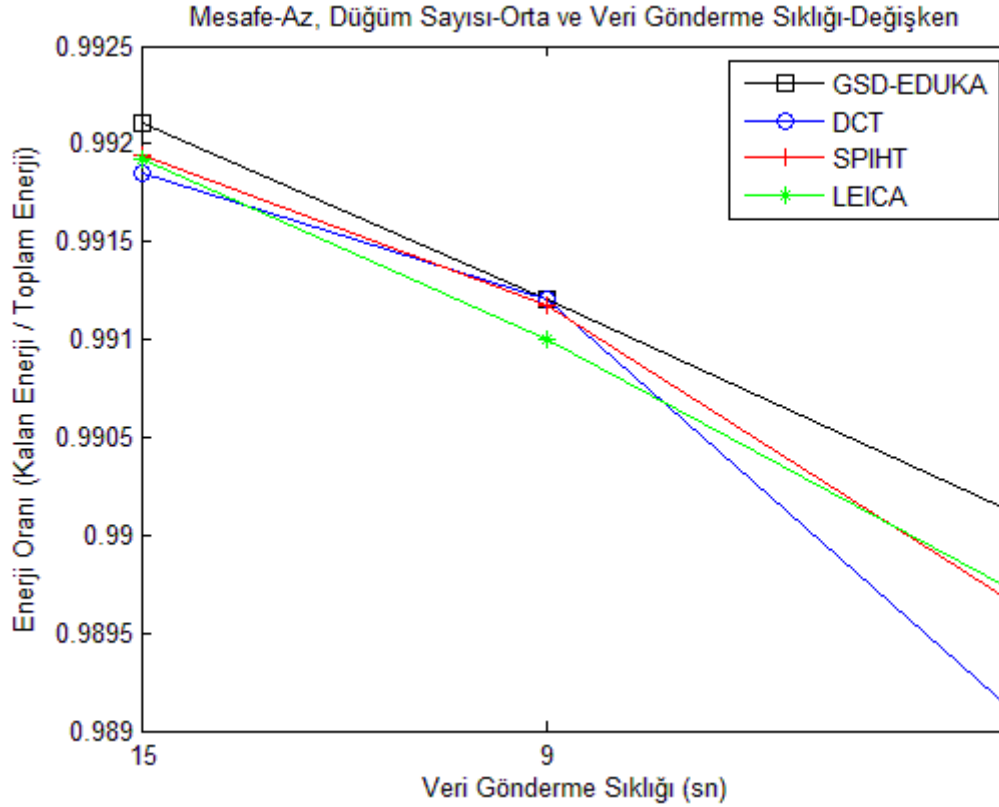
Öncelikle sıkıştırma algoritmaları tek başlarına ayrı ayrı kullanılarak (DCT, SPIHT, LEICA) ve GSD-EDUKA kullanılarak elde edilen sonuçlar karşılaştırılmış, çizelge ve grafik aracılığıyla elde edilen sonuçlar verilmiştir. Daha sonra da GSD-EDUKA ile sıkıştırma işlemi yapmadan elde edilen sonuçlar karşılaştırılmıştır ve sadece çizelge aracılığıyla elde edilen sonuçlar verilmiştir.

5.3.3.1 Sabit Ağlar için Enerji Oranı Sonuçları

Çizelge 5.1’deki senaryolar sabit ağlar için benzetim ortamında gerçekleştirilmiştir. Bölümün devamında gerçekleştirilen senaryolar içerisinde örnek olarak belirlenen senaryo 2, 4, 7, 11, 14, 17, 19, 22 ve 25’e ait sonuç grafikleri ve çizelgeleri verilmiştir. Sabit ağlara ait grafikler Şekil 5.1 - Şekil 5.9’te gösterilmiştir. Grafiklere ait değerler ise Çizelge 5.5 - Çizelge 5.13’te verilmiştir.

Çizelge 5.5. Senaryo 2: Mesafe-az ve düğüm sayısı-orta olduğu durumda beş yöntemin veri gönderme sıklığına göre enerji oran değerleri.

Veri Gönderme Sıklığı (sn)	Enerji Oranı				
	DCT	SPIHT	LEICA	GSD-EDUKA	Sıkıştırma Uygulamadan
15	0,9918469	0,9919395	0,9919179	0,9921035	0,9887133
9	0,9912076	0,9911705	0,9910011	0,9912076	0,9819545
3	0,9891125	0,989661	0,9897173	0,9901256	0,9727593



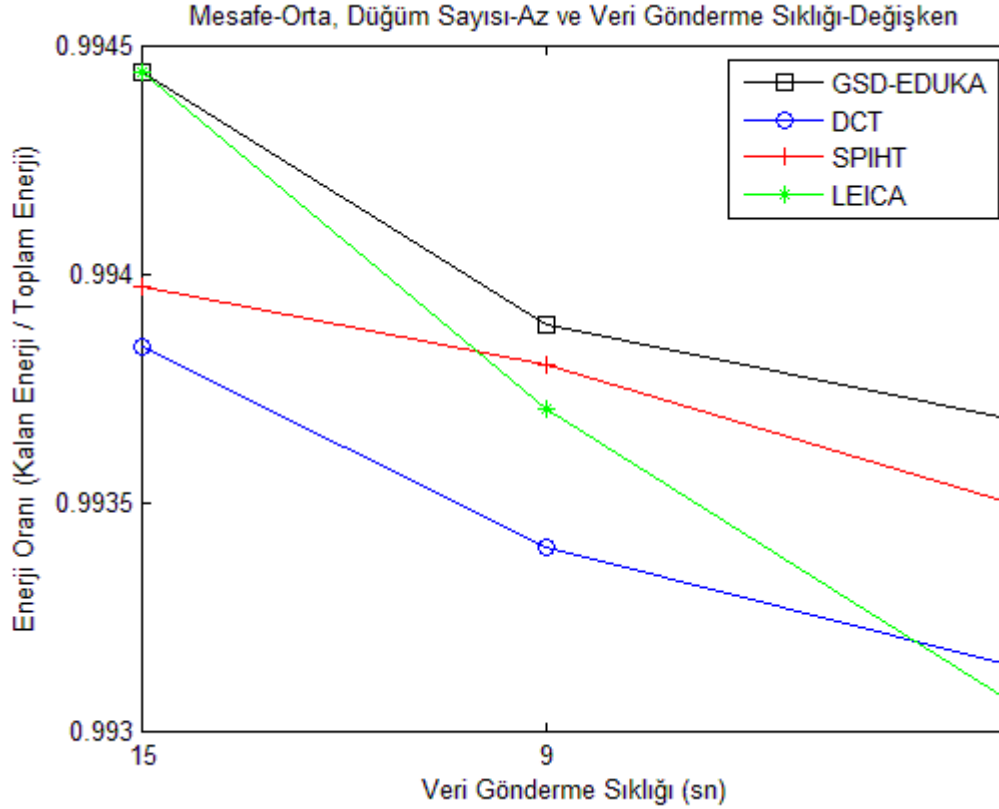
Şekil 5.1. Senaryo 2: Mesafe-az ve düğüm sayısı-orta olduğu durumda dört yöntemin veri gönderme sıklığına göre enerji oran grafiği.

Senaryo 2’de kullanılan görüntü verisi örnek görüntü 1’dir. Benzetimde, GSD-EDUKA, veri gönderme sıklığının 15 saniye olduğu durumda %10 sıkıştırma oranı ile SPIHT algoritmasını, 9 saniye olduğu durumda %5 sıkıştırma oranı ile DCT algoritmasını ve 3 saniye olduğu durumda ise %20 sıkıştırma oranı ile LEICA algoritmasını kullanarak en iyi enerji oranını yakalamıştır.

Çizelge 5.5’teki değerlere veya Şekil 5.1’deki grafiğe göre senaryo 2 için GSD-EDUKA’nın daha yüksek enerji oranına sahip olduğu görülmektedir.

Çizelge 5.6. Senaryo 4: Mesafe-orta ve düğüm sayısı-az olduğu durumda beş yöntemin veri gönderme sıklığına göre enerji oran değerleri.

Veri Gönderme Sıklığı (sn)	Enerji Oranı				
	DCT	SPIHT	LEICA	GSD-EDUKA	Sıkıştırma Uygulamadan
15	0,9938454	0,9939762	0,9944439	0,9944439	0,9928777
9	0,9934073	0,9938033	0,9937094	0,9940211	0,9916103
3	0,9931478	0,9935011	0,9930668	0,9938412	0,9878540



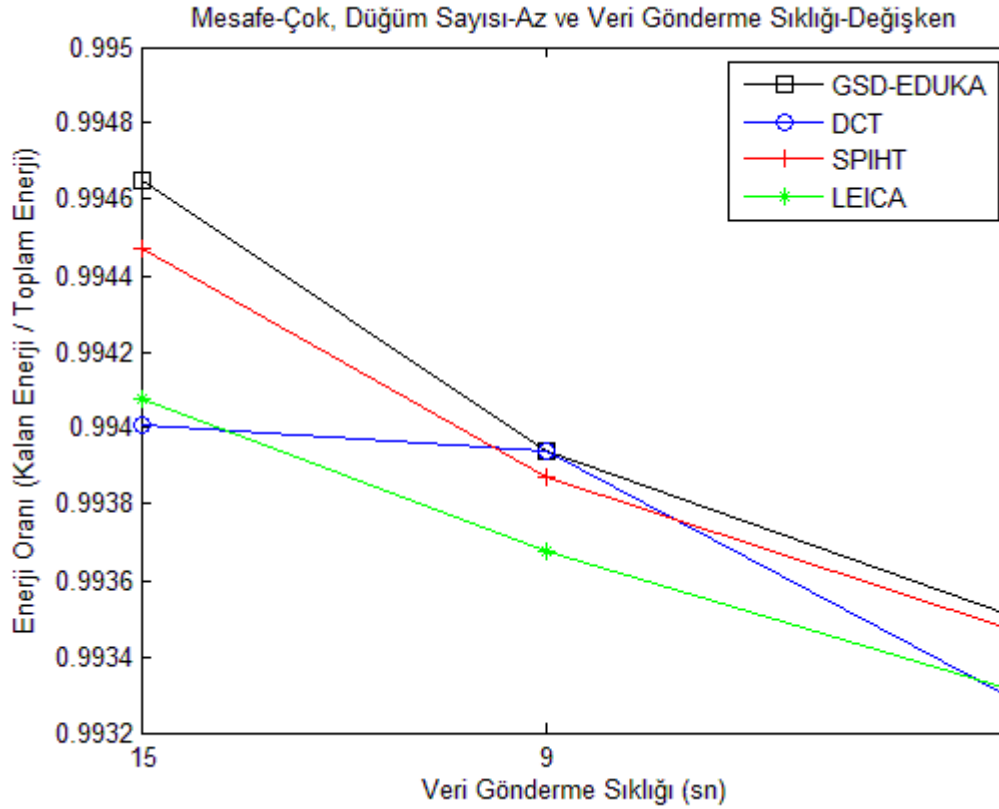
Şekil 5.2. Senaryo 4: Mesafe-orta ve düğüm sayısı-az olduğu durumda dört yöntemin veri gönderme sıklığına göre enerji oran grafiği.

Senaryo 4'te kullanılan görüntü verisi örnek görüntü 2'dir. Benzetimde, GSD-EDUKA, veri gönderme sıklığının 15 saniye olduğu durumda %5 sıkıştırma oranı ile LEICA algoritmasını, 9 saniye olduğu durumda %10 sıkıştırma oranı ile SPIHT algoritmasını ve 3 saniye olduğu durumda ise %15 sıkıştırma oranı ile SPIHT algoritmasını kullanarak en iyi enerji oranını yakalamıştır.

Çizelge 5.6'daki değerlere veya Şekil 5.2'deki grafiğe göre senaryo 4 için GSD-EDUKA'nın daha yüksek enerji oranına sahip olduğu görülmektedir.

Çizelge 5.7. Senaryo 7: Mesafe-çok ve düğüm sayısı-az olduğu durumda beş yöntemin veri gönderme sıklığına göre enerji oran değerleri.

Veri Gönderme Sıklığı (sn)	Enerji Oranı				
	DCT	SPIHT	LEICA	GSD-EDUKA	Sıkıştırma Uygulamadan
15	0,9940079	0,9944714	0,9940742	0,9946501	0,9928071
9	0,9939400	0,9938684	0,9936741	0,9939400	0,9913413
3	0,9932917	0,9934707	0,9933155	0,9935098	0,9878110



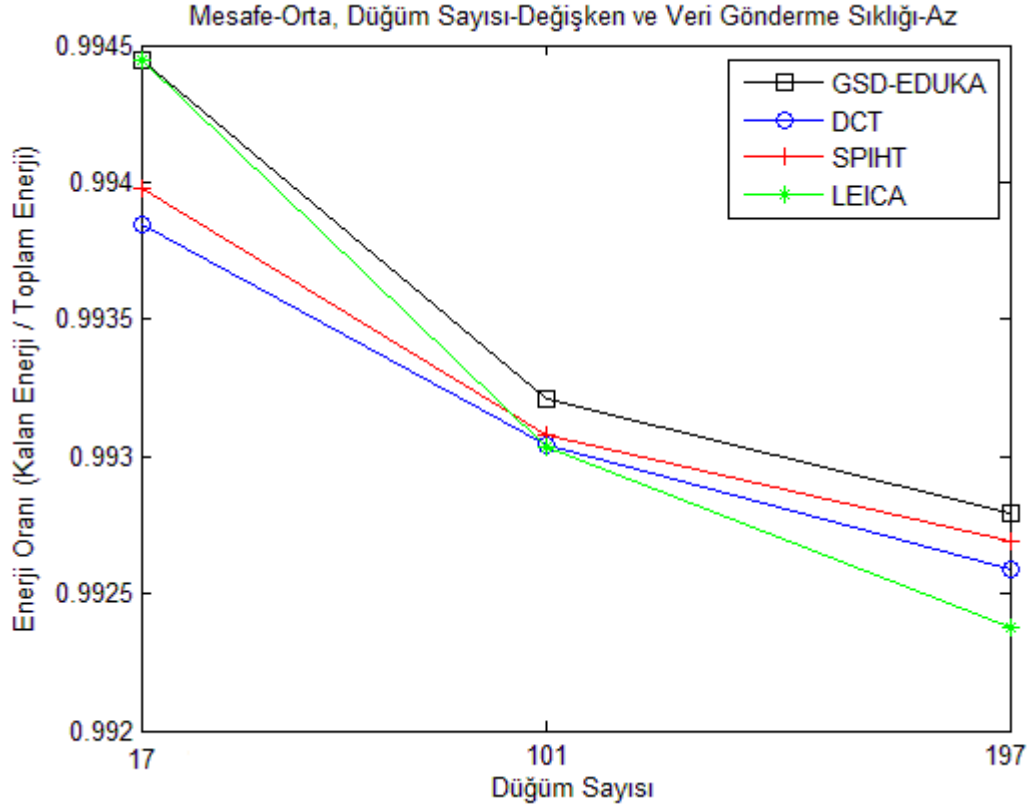
Şekil 5.3. Senaryo 7: Mesafe-çok ve düğüm sayısı-az olduğu durumda dört yöntemin veri gönderme sıklığına göre enerji oran grafiği.

Senaryo 7’de kullanılan görüntü verisi örnek görüntü 3’dür. Benzetimde, GSD-EDUKA, veri gönderme sıklığının 15 saniye olduğu durumda %10 sıkıştırma oranı ile SPIHT algoritmasını, 9 saniye olduğu durumda %5 sıkıştırma oranı ile DCT algoritmasını ve 3 saniye olduğu durumda ise %10 sıkıştırma oranı ile SPIHT algoritmasını kullanarak en iyi enerji oranını yakalamıştır.

Çizelge 5.7’deki değerlere veya Şekil 5.3’teki grafiğe göre senaryo 7 için GSD-EDUKA’nın daha iyi enerji oranına sahip olduğu görülmektedir.

Çizelge 5.8. Senaryo 11: Mesafe-orta ve veri gönderme sıklığı-az olduğu durumda beş yöntemin düğüm sayısına göre enerji oran değerleri.

Düğüm Sayısı	Enerji Oranı				
	DCT	SPIHT	LEICA	GSD-EDUKA	Sıkıştırma Uygulamadan
17	0,9938447	0,9939762	0,9944439	0,9944439	0,9928777
101	0,9930386	0,9930810	0,9930327	0,9932125	0,9900944
197	0,9925863	0,9926916	0,9923728	0,9927954	0,9889077



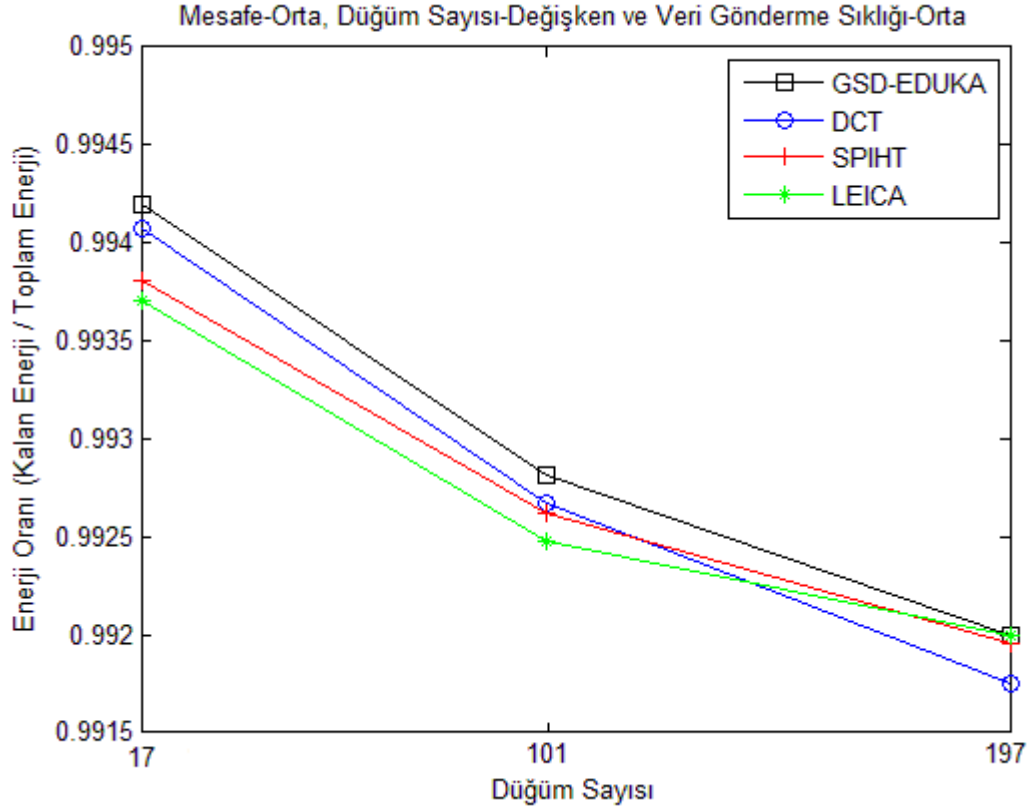
Şekil 5.4. Senaryo 11: Mesafe-orta ve veri gönderme sıklığı-az olduğu durumda dört yöntemin düğüm sayısına göre enerji oran grafiği.

Senaryo 11’de kullanılan görüntü verisi örnek görüntü 1’dir. Benzetimde, GSD-EDUKA, düğüm sayısının 17 olduğu durumda %5 sıkıştırma oranı ile LEICA algoritmasını, 101 olduğu durumda %15 sıkıştırma oranı ile SPIHT algoritmasını ve 197 olduğu durumda ise %10 sıkıştırma oranı ile SPIHT algoritmasını kullanarak en iyi enerji oranını yakalamıştır.

Çizelge 5.8’deki değerlere veya Şekil 5.4’teki grafiğe göre senaryo için GSD-EDUKA’nın daha iyi enerji oranına sahip olduğu görülmektedir.

Çizelge 5.9. Senaryo 14: Mesafe-orta ve veri gönderme sıklığı-orta olduğu durumda beş yöntemin düğüm sayısına göre enerji oran değerleri.

Düğüm Sayısı	Enerji Oranı				
	DCT	SPIHT	LEICA	GSD-EDUKA	Sıkıştırma Uygulamadan
17	0,9940739	0,9938033	0,9937094	0,9942013	0,9916103
101	0,9926702	0,9926158	0,9924812	0,9928129	0,9884496
197	0,9917485	0,9919535	0,9919917	0,9919917	0,9847851



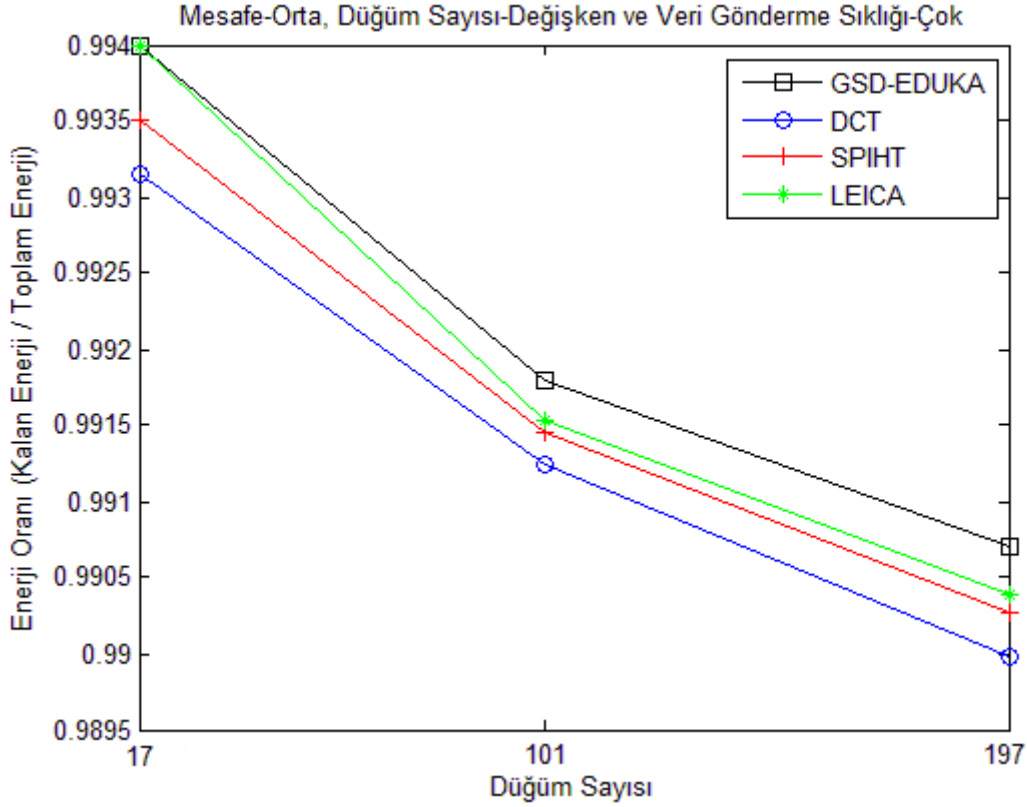
Şekil 5.5. Senaryo 14: Mesafe-orta ve veri gönderme sıklığı-orta olduğu durumda dört yöntemin düğüm sayısına göre enerji oran grafiği.

Senaryo 14’de kullanılan görüntü verisi örnek görüntü 3’dür. Benzetimde, GSD-EDUKA, düğüm sayısının 17 olduğu durumda %10 sıkıştırma oranı ile DCT algoritmasını, 101 olduğu durumda %10 sıkıştırma oranı ile DCT algoritmasını ve 197 olduğu durumda ise %5 sıkıştırma oranı ile LEICA algoritmasını kullanarak en iyi enerji oranını yakalamıştır.

Çizelge 5.9’daki değerlere veya Şekil 5.5’teki grafiğe göre senaryo 14 için GSD-EDUKA’nın daha iyi enerji oranına sahip olduğu görülmektedir.

Çizelge 5.10. Senaryo 17: Mesafe-orta ve veri gönderme sıklığı-çok olduğu durumda beş yöntemin düğüm sayısına göre enerji oran değerleri.

Düğüm Sayısı	Enerji Oranı				
	DCT	SPIHT	LEICA	GSD-EDUKA	Sıkıştırma Uygulamadan
17	0,9931478	0,9935011	0,9939898	0,9939898	0,9878540
101	0,9912436	0,9914603	0,9915305	0,9918255	0,9806634
197	0,9899827	0,9902698	0,9903891	0,9907101	0,9784286



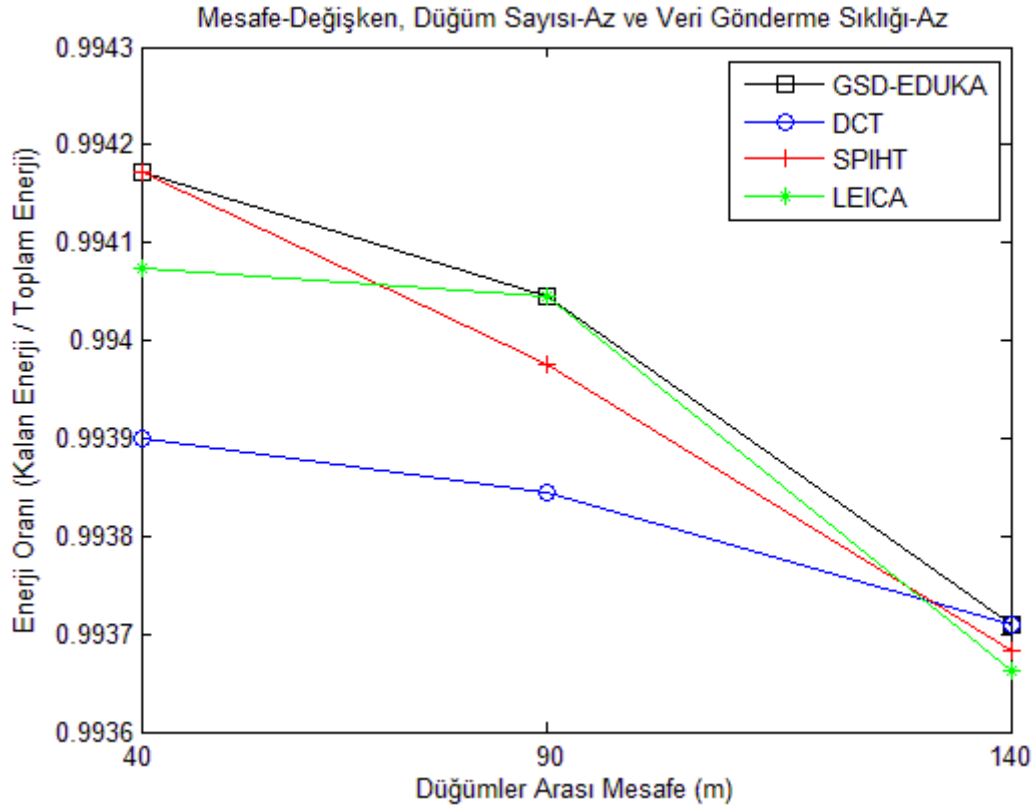
Şekil 5.6. Senaryo 17: Mesafe-orta ve veri gönderme sıklığı-çok olduğu durumda dört yöntemin düğüm sayısına göre enerji oran grafiği.

Senaryo 17’de kullanılan görüntü verisi örnek görüntü 2’dir. Benzetimde, GSD-EDUKA, düğüm sayısının 17 olduğu durumda %5 sıkıştırma oranı ile LEICA algoritmasını, 101 olduğu durumda %15 sıkıştırma oranı ile LEICA algoritmasını ve 197 olduğu durumda ise %10 sıkıştırma oranı ile LEICA algoritmasını kullanarak en iyi enerji oranını yakalamıştır.

Çizelge 5.10’daki değerlere veya Şekil 5.6’daki grafiğe göre senaryo 17 için GSD-EDUKA’nın daha iyi enerji oranına sahip olduğu görülmektedir.

Çizelge 5.11. Senaryo 19: Düğüm sayısı-az ve veri gönderme sıklığı-az olduğu durumda beş yöntemin mesafeye göre enerji oran değerleri.

Mesafe (m)	Enerji Oranı				
	DCT	SPIHT	LEICA	GSD-EDUKA	Sıkıştırma Uygulamadan
40	0,9937096	0,9936819	0,9936627	0,9937096	0,9917281
90	0,9938454	0,9939762	0,9940443	0,9940443	0,9928777
140	0,9939007	0,9941714	0,9940742	0,9941714	0,9928071



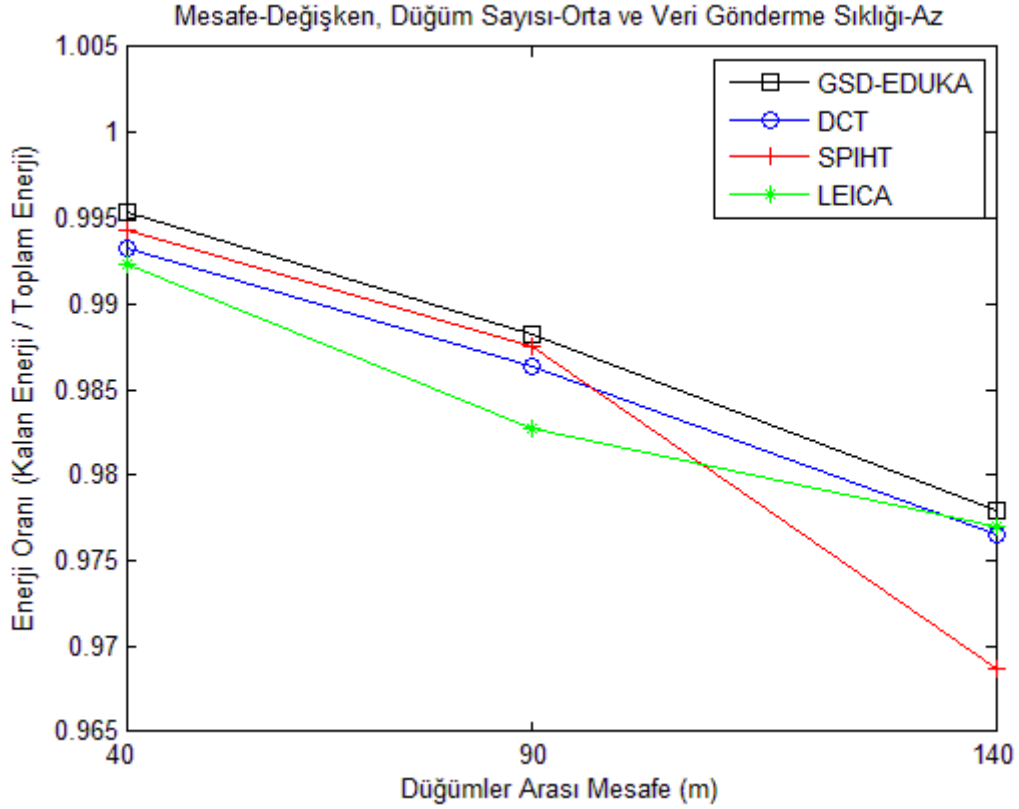
Şekil 5.7. Senaryo 19: Düğüm sayısı-az ve veri gönderme sıklığı-az olduğu durumda dört yöntemin mesafeye göre enerji oran grafiği.

Senaryo 19’da kullanılan görüntü verisi örnek görüntü 1’dir. Benzetimde, GSD-EDUKA, düğümler arası mesafenin 40 metre olduğu durumda %5 sıkıştırma oranı ile SPIHT algoritmasını, 90 metre olduğu durumda %5 sıkıştırma oranı ile LEICA algoritmasını ve 190 metre olduğu durumda ise %5 sıkıştırma oranı ile DCT algoritmasını kullanarak en iyi enerji oranını yakalamıştır.

Çizelge 5.11’deki değerlere veya Şekil 5.7’deki grafiğe göre senaryo 19 için GSD-EDUKA’nın daha iyi enerji oranına sahip olduğu görülmektedir.

Çizelge 5.12. Senaryo 22: Düğüm sayısı-orta ve veri gönderme sıklığı-az olduğu durumda beş yöntemin mesafeye göre enerji oran değerleri.

Mesafe (m)	Enerji Oranı				
	DCT	SPIHT	LEICA	GSD-EDUKA	Sıkıştırma Uygulamadan
40	0,9932325	0,9943369	0,9923319	0,9953312	0,9919021
90	0,9862930	0,9874930	0,9827493	0,9881799	0,9900944
140	0,9764763	0,9686787	0,9769865	0,9779215	0,9691995



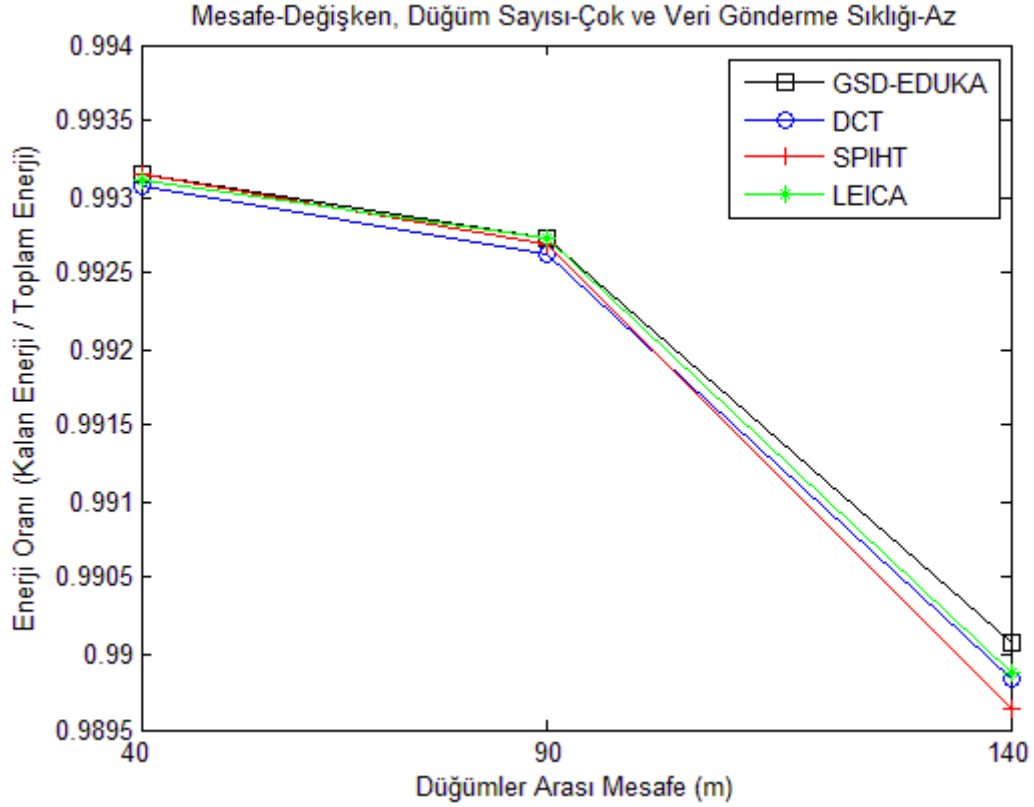
Şekil 5.8. Senaryo 22: Düğüm sayısı-orta ve veri gönderme sıklığı-az olduğu durumda dört yöntemin mesafeye göre enerji oran grafiği.

Senaryo 22’de kullanılan görüntü verisi örnek görüntü 3’dür. Benzetimde, GSD-EDUKA, düğümler arası mesafenin 40 metre olduğu durumda %10 sıkıştırma oranı ile SPIHT algoritmasını, 90 metre olduğu durumda %15 sıkıştırma oranı ile SPIHT algoritmasını ve 190 metre olduğu durumda ise %10 sıkıştırma oranı ile LEICA algoritmasını kullanarak en iyi enerji oranını yakalamıştır.

Çizelge 5.12’deki değerlere veya Şekil 5.8’deki grafiğe göre senaryo 22 için GSD-EDUKA’nın daha iyi enerji oranına sahip olduğu görülmektedir.

Çizelge 5.13. Senaryo 25: Düğüm sayısı-çok ve veri gönderme sıklığı-az olduğu durumda beş yöntemin mesafeye göre enerji oran değerleri.

Mesafe (m)	Enerji Oranı				
	DCT	SPIHT	LEICA	GSD-EDUKA	Sıkıştırma Uygulamadan
40	0,9930786	0,9931499	0,9931062	0,9931499	0,9899057
90	0,9926305	0,9926916	0,9927286	0,9927286	0,9889077
140	0,9898319	0,9896428	0,9898774	0,9900778	0,9837731



Şekil 5.9. Senaryo 25: Düğüm sayısı-çok ve veri gönderme sıklığı-az olduğu durumda dört yöntemin mesafeye göre enerji oran grafiği.

Senaryo 25'te kullanılan görüntü verisi örnek görüntü 2'dir. Benzetimde, GSD-EDUKA, düğümler arası mesafenin 40 metre olduğu durumda %5 sıkıştırma oranı ile SPIHT algoritmasını, 90 metre olduğu durumda %5 sıkıştırma oranı ile LEICA algoritmasını ve 190 metre olduğu durumda ise %10 sıkıştırma oranı ile LEICA algoritmasını kullanarak en iyi enerji oranını yakalamıştır.

Çizelge 5.13'teki değerlere veya Şekil 5.9'daki grafiğe göre senaryo 25 için GSD-EDUKA'nın daha iyi enerji oranına sahip olduğu görülmektedir.

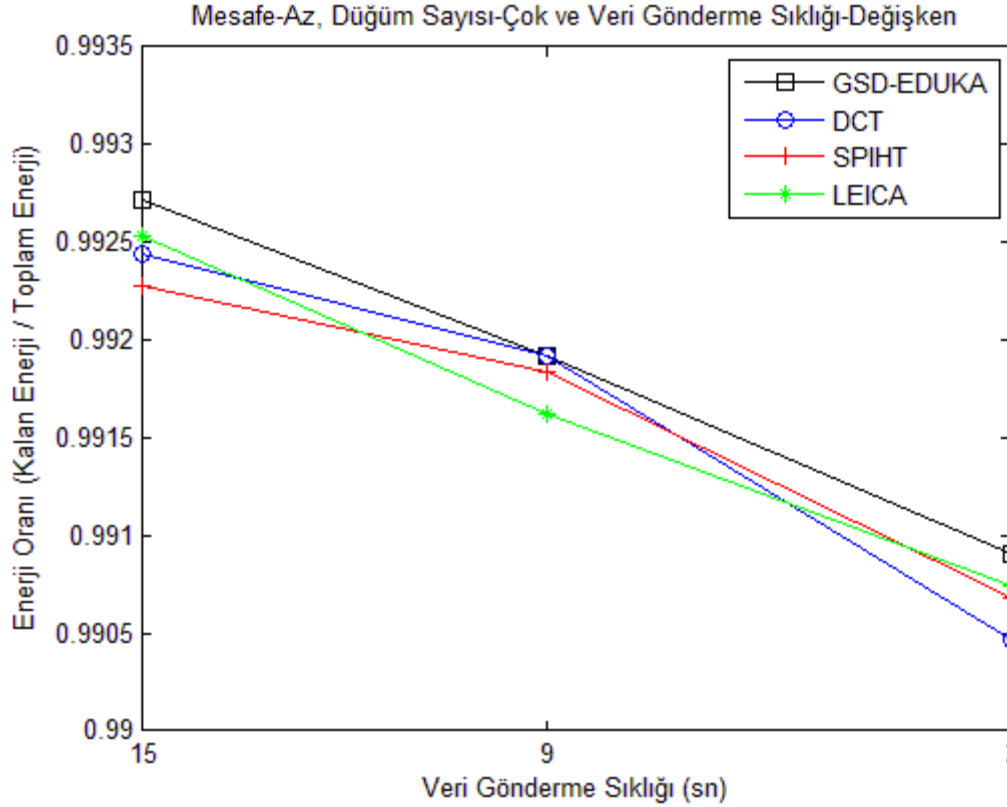
5.3.3.2 Hareketli Ağlar için Enerji Oranı Sonuçları

Çizelge 5.1'deki senaryolar hareketli ağlar için benzetim ortamında gerçekleştirilmiştir. Algılayıcı düğümlerin hareket hızları bazı senaryolarda 5 m/s bazılarında ise 10 m/s olarak belirlenmiştir. Senaryo 3, 5, 12, 15, 20 ve 23'teki algılayıcı düğümlerin hızları 5m/s, senaryo 8, 18 ve 26'daki algılayıcı düğümlerin hızları ise 10 m/s'dir. Bölümün devamında gerçekleştirilen sonuç grafikleri ve çizelgeleri verilmiştir. Hareketli ağlara ait grafikler Şekil 5.10 - Şekil 5.18'de gösterilmiştir. Grafiklere ait değerler ise Çizelge 5.14 - Çizelge 5.22'de verilmiştir.



Çizelge 5.14. Senaryo 3: Mesafe-az ve düğüm sayısı-çok olduğu durumda beş yöntemin veri gönderme sıklığına göre enerji oran değerleri.

Veri Gönderme Sıklığı (sn)	Enerji Oranı				
	DCT	SPIHT	LEICA	GSD-EDUKA	Sıkıştırma Uygulamadan
15	0,9924413	0,9922819	0,9925320	0,9927140	0,9905598
9	0,9919207	0,9918407	0,9916191	0,9919207	0,9885940
3	0,9904621	0,9906785	0,9907414	0,9909013	0,9852178



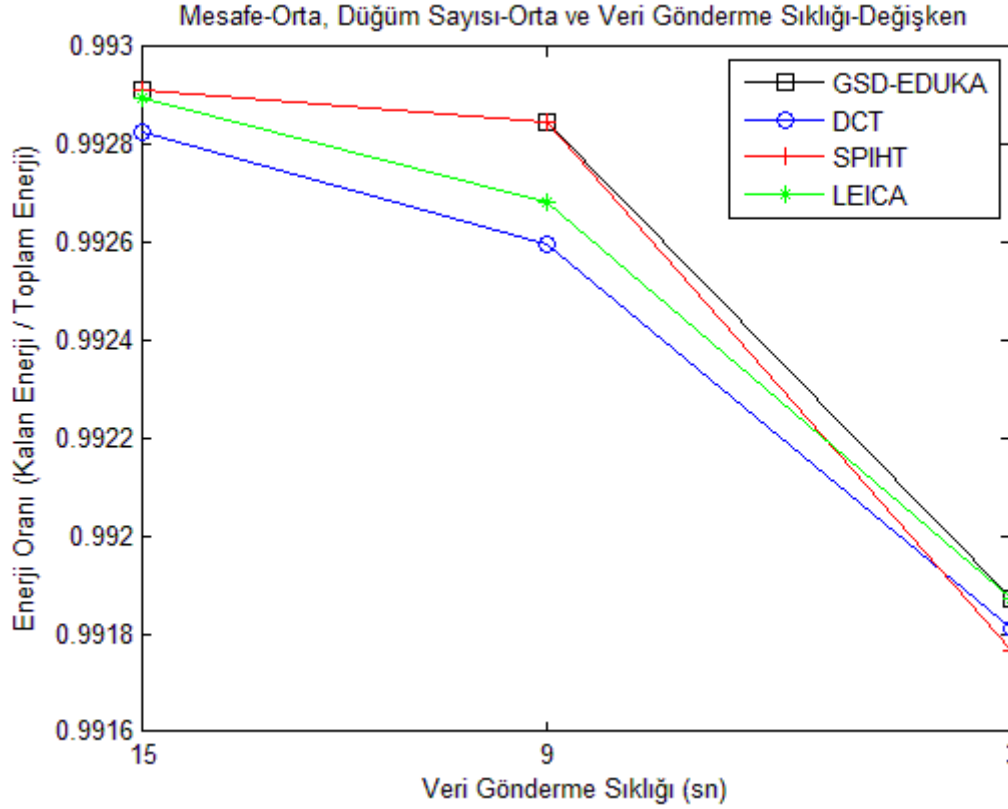
Şekil 5.10. Senaryo 3: Mesafe-az ve düğüm sayısı-çok olduğu durumda dört yöntemin veri gönderme sıklığına göre enerji oran grafiği.

Senaryo 3'te kullanılan görüntü verisi örnek görüntü 1'dir. Benzetimde, GSD-EDUKA, veri gönderme sıklığının 15 saniye olduğu durumda %10 sıkıştırma oranı ile LEICA algoritmasını, 9 saniye olduğu durumda %5 sıkıştırma oranı ile SPIHT algoritmasını ve 3 saniye olduğu durumda ise %10 sıkıştırma oranı ile LEICA algoritmasını kullanarak en iyi enerji oranını yakalamıştır.

Çizelge 5.14'teki değerlere veya Şekil 5.10'daki grafiğe göre senaryo 3 için GSD-EDUKA'nın daha iyi enerji oranına sahip olduğu görülmektedir.

Çizelge 5.15. Senaryo 5: Mesafe-orta ve düğüm sayısı-orta olduğu durumda beş yöntemin veri gönderme sıklığına göre enerji oran değerleri.

Veri Gönderme Sıklığı (sn)	Enerji Oranı				
	DCT	SPIHT	LEICA	GSD-EDUKA	Sıkıştırma Uygulamadan
15	0,9928254	0,9929120	0,9928944	0,9929120	0,9906128
9	0,9925975	0,9928449	0,9926844	0,9928449	0,9897579
3	0,9918096	0,9917640	0,9918717	0,9918717	0,9862466



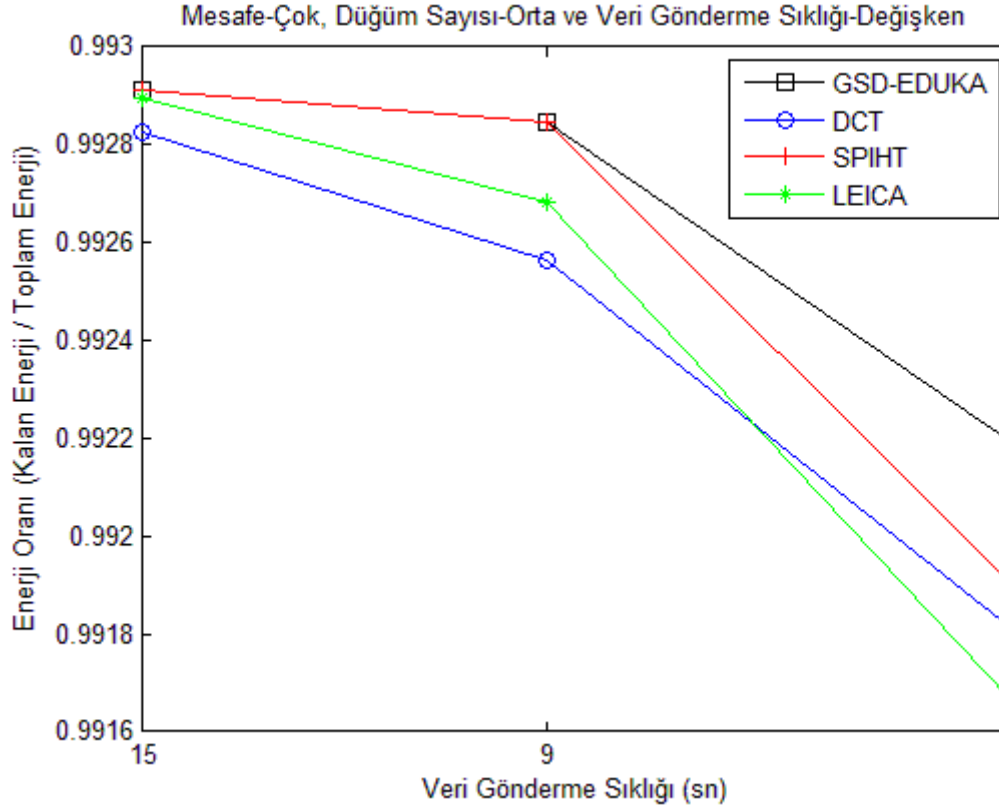
Şekil 5.11. Senaryo 5: Mesafe-orta ve düğüm sayısı-orta olduğu durumda dört yöntemin veri gönderme sıklığına göre enerji oran grafiği.

Senaryo 5'te kullanılan görüntü verisi örnek görüntü 2'dir. Benzetimde, GSD-EDUKA, veri gönderme sıklığının 15 saniye olduğu durumda %5 sıkıştırma oranı ile SPIHT algoritmasını, 9 saniye olduğu durumda %5 sıkıştırma oranı ile SPIHT algoritmasını ve 3 saniye olduğu durumda ise %5 sıkıştırma oranı ile LEICA algoritmasını kullanarak en iyi enerji oranını yakalamıştır.

Çizelge 5.15'teki değerlere veya Şekil 5.11'deki grafiğe göre Senaryo 5 için GSD-EDUKA'nın daha iyi enerji oranına sahip olduğu görülmektedir.

Çizelge 5.16. Senaryo 8: Mesafe-çok ve düğüm sayısı-orta olduğu durumda beş yöntemin veri gönderme sıklığına göre enerji oran değerleri.

Veri Gönderme Sıklığı (sn)	Enerji Oranı				
	DCT	SPIHT	LEICA	GSD-EDUKA	Sıkıştırma Uygulamadan
15	0,9928254	0,9929120	0,9928947	0,9929120	0,9906128
9	0,9925621	0,9928449	0,9926844	0,9928449	0,9897579
3	0,9918096	0,9919019	0,9916648	0,9921896	0,9862466



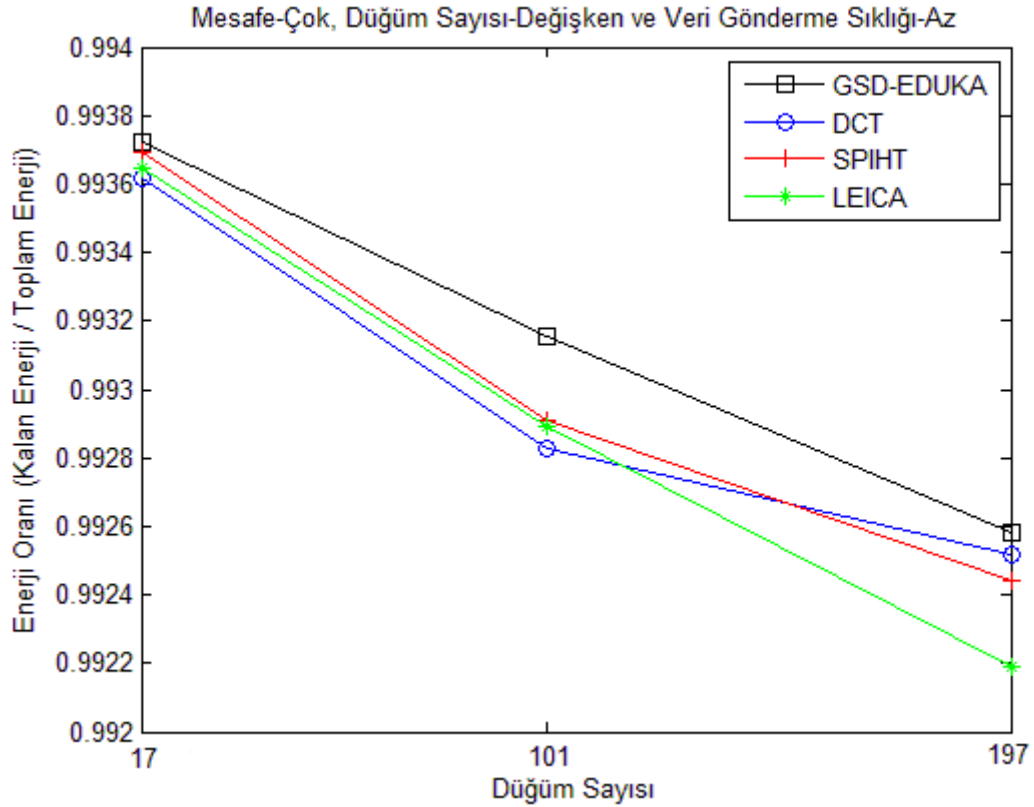
Şekil 5.12. Senaryo 8: Mesafe-çok ve düğüm sayısı-orta olduğu durumda dört yöntemin veri gönderme sıklığına göre enerji oran grafiği.

Senaryo 8’de kullanılan görüntü verisi örnek görüntü 3’dür. Benzetimde, GSD-EDUKA, veri gönderme sıklığının 15 saniye olduğu durumda %5 sıkıştırma oranı ile SPIHT algoritmasını, 9 saniye olduğu durumda %5 sıkıştırma oranı ile SPIHT algoritmasını ve 3 saniye olduğu durumda ise %10 sıkıştırma oranı ile LEICA algoritmasını kullanarak en iyi enerji oranını yakalamıştır.

Çizelge 5.16’daki değerlere veya Şekil 5.12’deki grafiğe göre senaryo 8 için en iyi enerji oranına SPIHT ve GSD-EDUKA’nın sahip olduğu görülmektedir.

Çizelge 5.17. Senaryo 12: Mesafe-çok ve veri gönderme sıklığı-az olduğu durumda beş yöntemin düğüm sayısına göre enerji oran değerleri.

Düğüm Sayısı	Enerji Oranı				
	DCT	SPIHT	LEICA	GSD-EDUKA	Sıkıştırma Uygulamadan
17	0,9936152	0,9936925	0,9936487	0,9937194	0,9923544
101	0,9928254	0,9929120	0,9928945	0,9931551	0,9916128
197	0,9925154	0,9924413	0,9921916	0,9925823	0,9905598



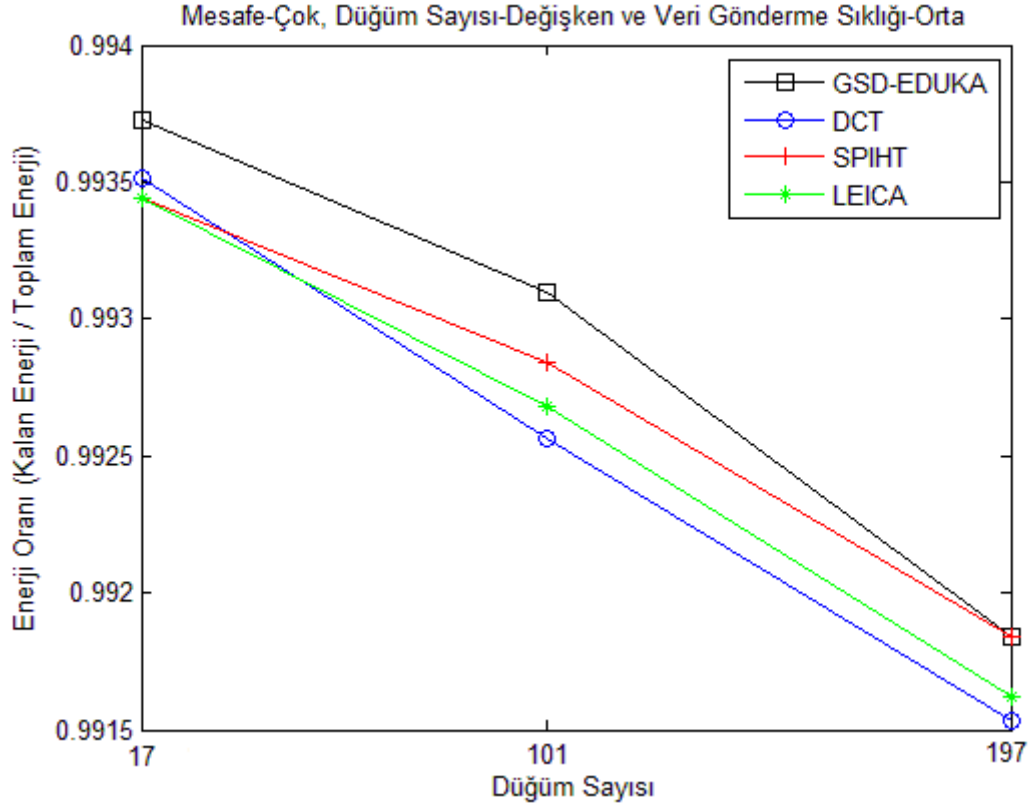
Şekil 5.13. Senaryo 12: Mesafe-çok ve veri gönderme sıklığı-az olduğu durumda dört yöntemin düğüm sayısına göre enerji oran grafiği.

Senaryo 12’de kullanılan görüntü verisi örnek görüntü 1’dir. Benzetimde, GSD-EDUKA, düğüm sayısının 17 olduğu durumda %10 sıkıştırma oranı ile SPIHT algoritmasını, 101 olduğu durumda %15 sıkıştırma oranı ile SPIHT algoritmasını ve 197 olduğu durumda ise %10 sıkıştırma oranı ile DCT algoritmasını kullanarak en iyi enerji oranını yakalamıştır.

Çizelge 5.17’deki değerlere veya Şekil 5.13’teki grafiğe göre senaryo 12 için GSD-EDUKA’nın daha iyi enerji oranına sahip olduğu görülmektedir.

Çizelge 5.18. Senaryo 15: Mesafe-çok ve veri gönderme sıklığı-orta olduğu durumda beş yöntemin düğüm sayısına göre enerji oran değerleri.

Düğüm Sayısı	Enerji Oranı				
	DCT	SPIHT	LEICA	GSD-EDUKA	Sıkıştırma Uygulamadan
17	0,9935127	0,9934378	0,9934378	0,9937258	0,9911955
101	0,9925621	0,9928449	0,9926847	0,9930974	0,9897579
197	0,9915316	0,9918407	0,9916195	0,9918407	0,9885940



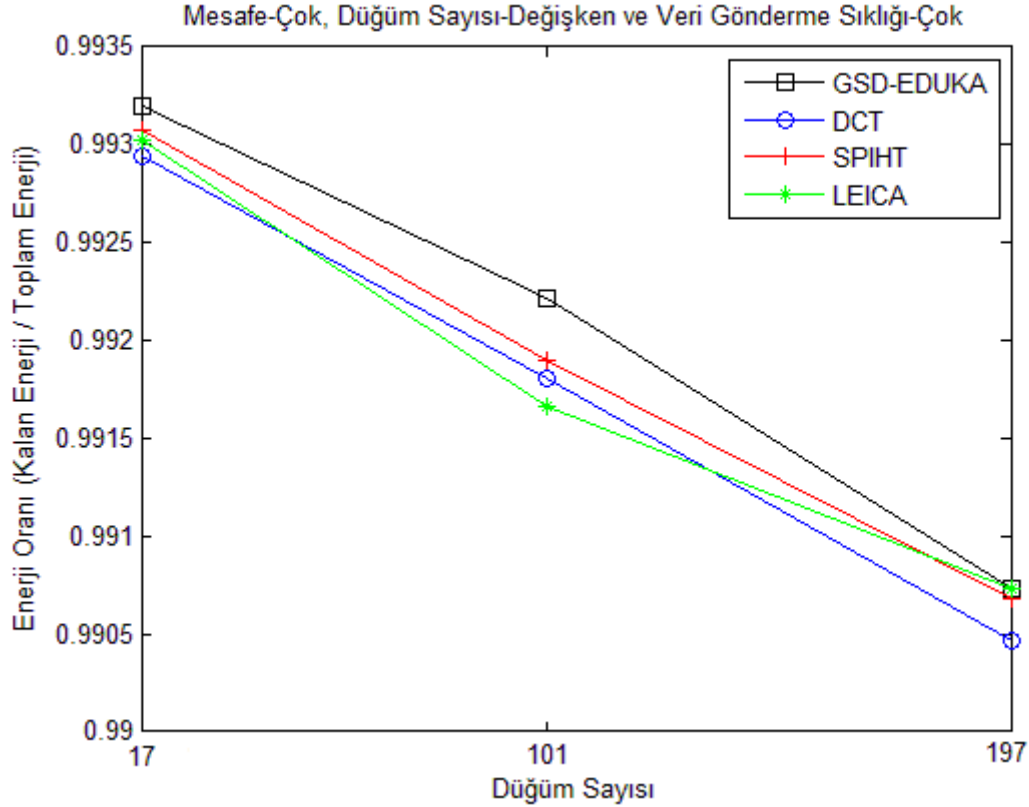
Şekil 5.14. Senaryo 15: Mesafe-çok ve veri gönderme sıklığı-orta olduğu durumda dört yöntemin düğüm sayısına göre enerji oran grafiği.

Senaryo 15'te kullanılan görüntü verisi örnek görüntü 2'dir. Benzetimde, GSD-EDUKA, düğüm sayısının 17 olduğu durumda %10 sıkıştırma oranı ile DCT algoritmasını, 101 olduğu durumda %10 sıkıştırma oranı ile SPIHT algoritmasını ve 197 olduğu durumda ise %5 sıkıştırma oranı ile SPIHT algoritmasını kullanarak en iyi enerji oranını yakalamıştır.

Çizelge 5.18'deki değerlere veya Şekil 5.14'deki grafiğe göre senaryo 15 için GSD-EDUKA'nın daha iyi enerji oranına sahip olduğu görülmektedir.

Çizelge 5.19. Senaryo 18: Mesafe-çok ve veri gönderme sıklığı-çok olduğu durumda beş yöntemin düğüm sayısına göre enerji oran değerleri.

Düğüm Sayısı	Enerji Oranı				
	DCT	SPIHT	LEICA	GSD-EDUKA	Sıkıştırma Uygulamadan
17	0,9929449	0,9930785	0,9930264	0,9932029	0,9889632
101	0,9918096	0,9919019	0,9916649	0,9922109	0,9862466
197	0,9904621	0,9906785	0,9907374	0,9907374	0,9852178



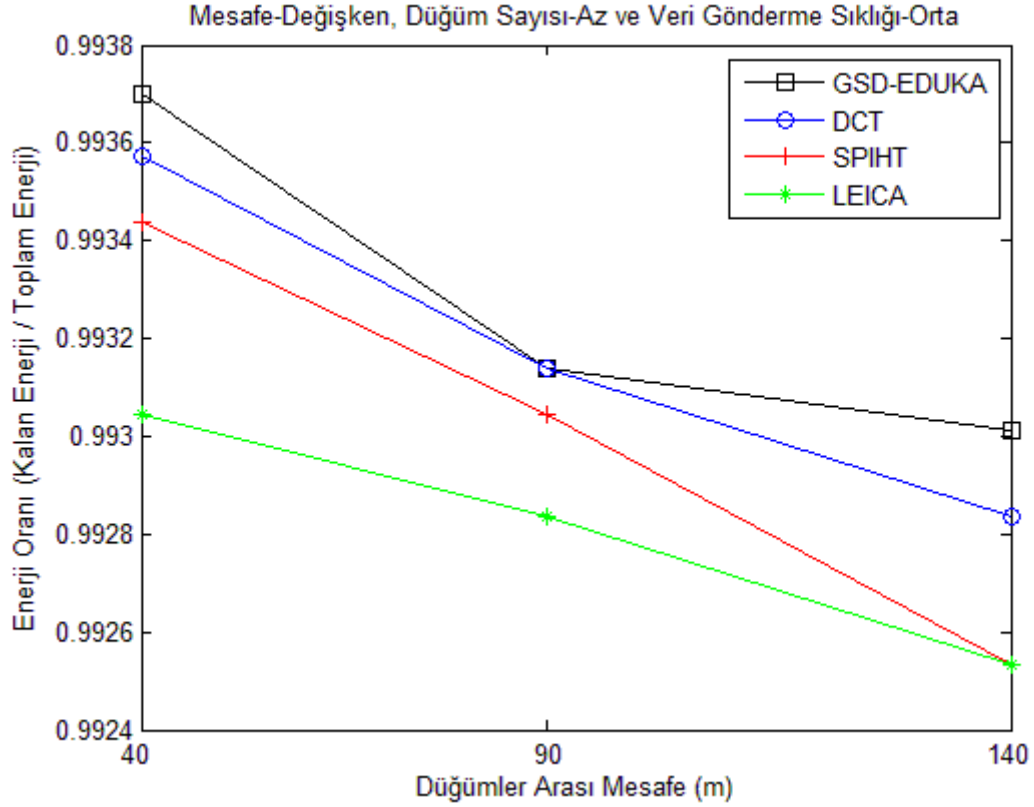
Şekil 5.15. Senaryo 18: Mesafe-çok ve veri gönderme sıklığı-çok olduğu durumda dört yöntemin düğüm sayısına göre enerji oran grafiği.

Senaryo 18’de kullanılan görüntü verisi örnek görüntü 3’dür. Benzetimde, GSD-EDUKA, düğüm sayısının 17 olduğu durumda %10 sıkıştırma oranı ile SPIHT algoritmasını, 101 olduğu durumda %10 sıkıştırma oranı ile SPIHT algoritmasını ve 197 olduğu durumda ise %5 sıkıştırma oranı ile LEICA algoritmasını kullanarak en iyi enerji oranını yakalamıştır.

Çizelge 5.19’daki değerlere veya Şekil 5.15’teki grafiğe göre senaryo 18 için GSD-EDUKA’nın daha iyi enerji oranına sahip olduğu görülmektedir.

Çizelge 5.20. Senaryo 20: Düğüm sayısı-az ve veri gönderme sıklığı-orta olduğu durumda beş yöntemin mesafeye göre enerji oran değerleri.

Mesafe (m)	Enerji Oranı				
	DCT	SPIHT	LEICA	GSD-EDUKA	Sıkıştırma Uygulamadan
40	0,9935715	0,9934378	0,9930447	0,9937011	0,9911955
90	0,9931395	0,9930437	0,9928349	0,9931395	0,9851624
140	0,9928351	0,9925343	0,9925340	0,9930127	0,9810195



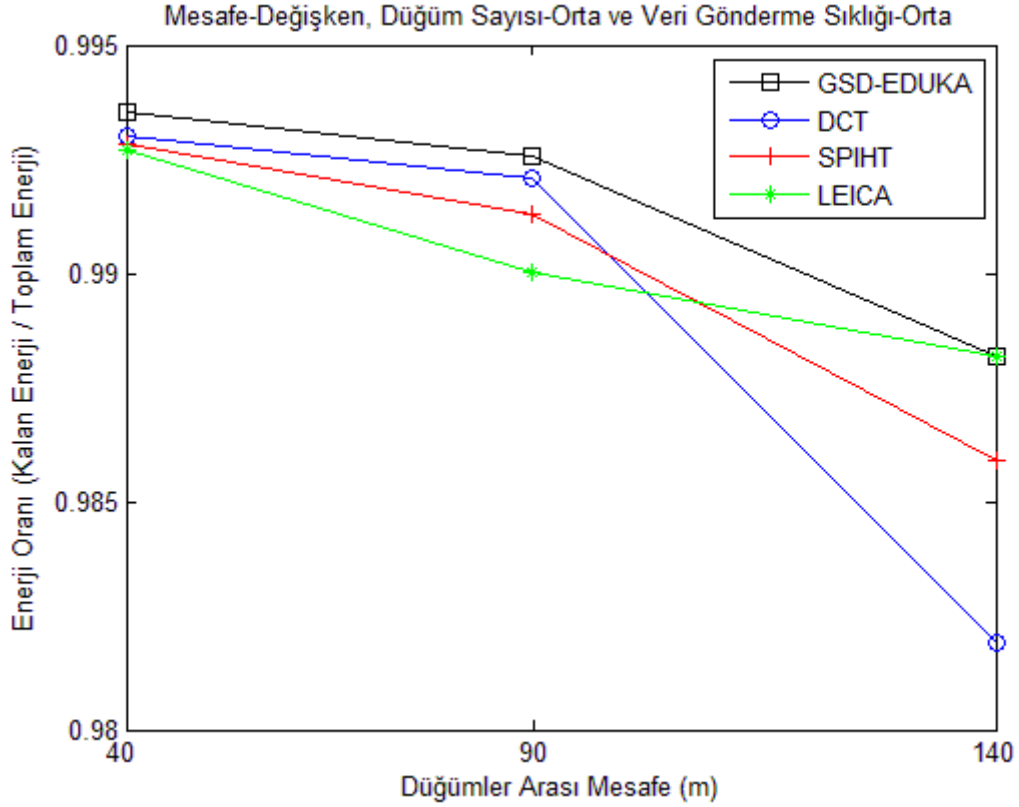
Şekil 5.16. Senaryo 20: Düğüm sayısı-az ve veri gönderme sıklığı-orta olduğu durumda dört yöntemin mesafeye göre enerji oran grafiği.

Senaryo 20’de kullanılan görüntü verisi örnek görüntü 1’dir. Benzetimde, GSD-EDUKA, düğümler arası mesafenin 40 metre olduğu durumda %10 sıkıştırma oranı ile DCT algoritmasını, 90 metre olduğu durumda %5 sıkıştırma oranı ile DCT algoritmasını ve 190 metre olduğu durumda ise %10 sıkıştırma oranı ile DCT algoritmasını kullanarak en iyi enerji oranını yakalamıştır.

Çizelge 5.20’deki değerlere veya Şekil 5.16’daki grafiğe göre senaryo 20 için GSD-EDUKA’nın daha iyi enerji oranına sahip olduğu görülmektedir.

Çizelge 5.21. Senaryo 23: Düğüm sayısı-orta ve veri gönderme sıklığı-orta olduğu durumda beş yöntemin mesafeye göre enerji oran değerleri.

Mesafe (m)	Enerji Oranı				
	DCT	SPIHT	LEICA	GSD-EDUKA	Sıkıştırma Uygulamadan
40	0,9929828	0,9928449	0,9926847	0,9935465	0,9897579
90	0,9920975	0,9912844	0,9900279	0,9925456	0,9545897
140	0,9819125	0,9859102	0,9881926	0,9881926	0,9325489



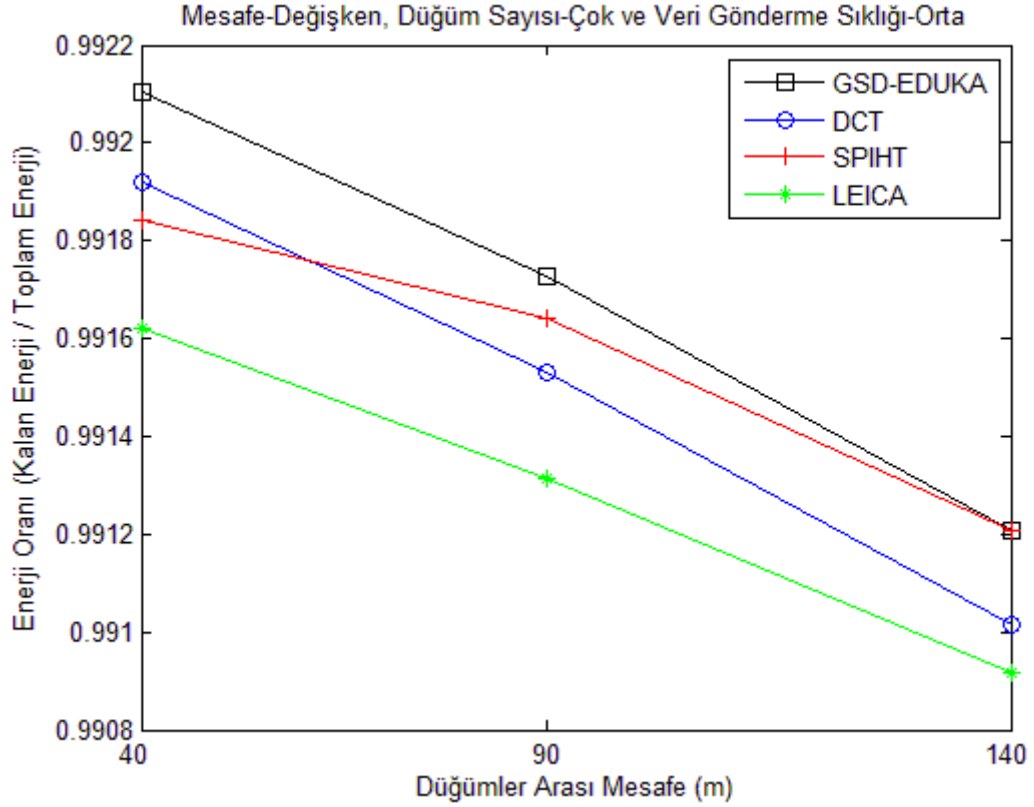
Şekil 5.17. Senaryo 23: Düğüm sayısı-orta ve veri gönderme sıklığı-orta olduğu durumda dört yöntemin mesafeye göre enerji oran grafiği.

Senaryo 23'te kullanılan görüntü verisi örnek görüntü 2'dir. Benzetimde, GSD-EDUKA, düğümler arası mesafenin 40 metre olduğu durumda %10 sıkıştırma oranı ile DCT algoritmasını, 90 metre olduğu durumda %15 sıkıştırma oranı ile DCT algoritmasını ve 190 metre olduğu durumda ise %5 sıkıştırma oranı ile LEICA algoritmasını kullanarak en iyi enerji oranını yakalamıştır.

Çizelge 5.21'deki değerlere veya Şekil 5.17'deki grafiğe göre senaryo 23 için GSD-EDUKA'nın daha iyi enerji oranına sahip olduğu görülmektedir.

Çizelge 5.22. Senaryo 26: Düğüm sayısı-çok ve veri gönderme sıklığı-orta olduğu durumda beş yöntemin mesafeye göre enerji oran değerleri.

Mesafe (m)	Enerji Oranı				
	DCT	SPIHT	LEICA	GSD-EDUKA	Sıkıştırma Uygulamadan
40	0,9919207	0,9918407	0,9916197	0,9921034	0,9885940
90	0,9915316	0,9916418	0,9913138	0,9917284	0,9870215
140	0,9910145	0,9912084	0,9909164	0,9912084	0,9830124



Şekil 5.18. Senaryo 26: Düğüm sayısı-çok ve veri gönderme sıklığı-orta olduğu durumda dört yöntemin mesafeye göre enerji oran grafiği.

Senaryo 26’da kullanılan görüntü verisi örnek görüntü 3’dür. Benzetimde, GSD-EDUKA, düğümler arası mesafenin 40 metre olduğu durumda %15 sıkıştırma oranı ile DCT algoritmasını, 90 metre olduğu durumda %10 sıkıştırma oranı ile SPIHT algoritmasını ve 190 metre olduğu durumda ise %5 sıkıştırma oranı ile SPIHT algoritmasını kullanarak en iyi enerji oranını yakalamıştır.

Çizelge 5.22’deki değerlere veya Şekil 5.18’deki grafiğe göre senaryo 26 için GSD-EDUKA’nın daha iyi enerji oranına sahip olduğu görülmektedir.

5.3.4 Paket Ulaştırma Oranı

Paket ulaştırma oranı (PUO), ağdaki veri gönderici düğümlerin gönderdikleri paketlerin, alıcılarına ulaşma oranlarını belirleyen bir performans ölçüm kriteridir. Yüzde cinsinden hedefine ulaşan paketlerin oranı olarak ifade edilir. Ağda gönderici düğümden çıkan paketler hedeflerine ulaşamamışlarsa, paket dağıtım oranı düşüktür. Alınan paketlerin sayısı artarsa, paket dağıtım oranı da artar. Bu tez çalışmasında, önerilen enerji verimli uygulama katmanı algoritmasında paket ulaştırma oranı ile ilgili bir iyileştirme yapılmamıştır fakat diğer algoritmalar ile karşılaştırılması gerçekleştirilmiştir.

PUO'yu belirlemek için Denklem (5.1)'deki ifade kullanılmıştır.

$$PUO = \frac{n_s}{n_T} \quad (5.1)$$

Bu ifadede n_s alınan paketlerin sayısı ve n_T ise üretilen toplam veri paketlerinin sayısıdır.

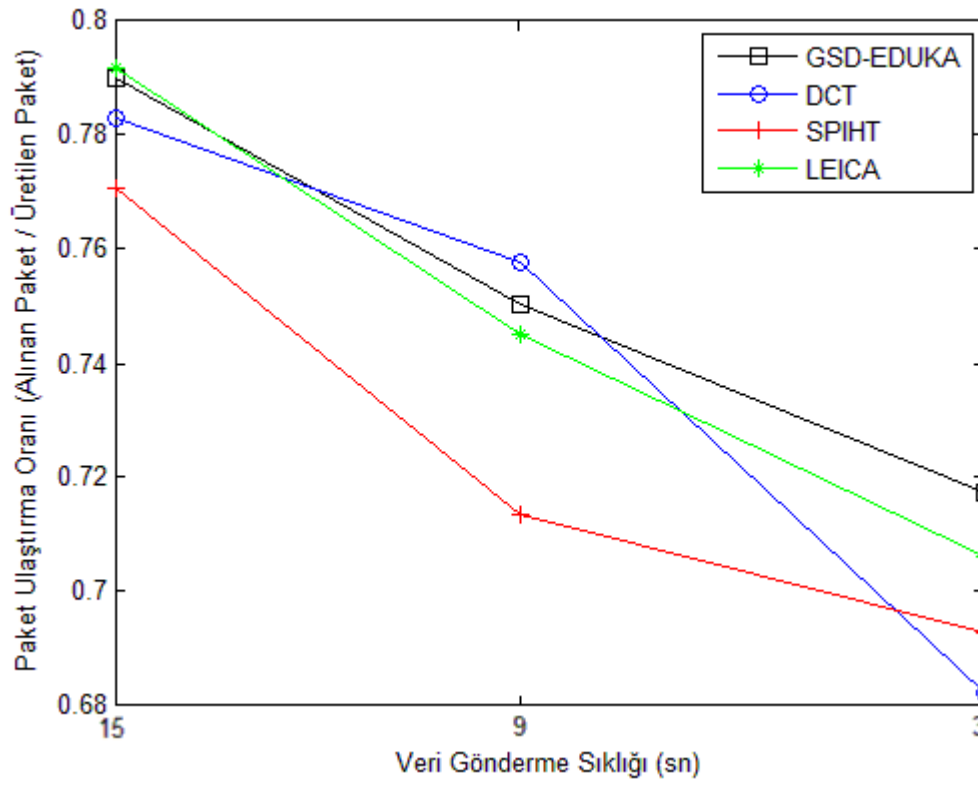
Çizelge 5.1'de belirtilen senaryolardan 4, 14 ve 22'nci senaryolar belirlenerek “paket ulaştırma oranı” değeri elde edilmiştir. PUO değerleri sırasıyla “veri gönderme sıklığı”, “düğüm sayısı” ve “düğümler arası mesafe” parametrelerinin farklı değerleri ile gerçekleşmiştir. Değişen parametreye karşılık diğer iki parametre sabit kalmaktadır.

Öncelikle sıkıştırma algoritmaları tek başlarına ayrı ayrı kullanılarak (DCT, SPIHT, LEICA) ve sonra da GSD-EDUKA kullanılarak elde edilen sonuçlar karşılaştırılmıştır, çizelge ve grafik aracılığıyla elde edilen sonuçlar verilmiştir.

Senaryo 4 için elde edilen değerler çizelge olarak Çizelge 5.23'te, grafik olarak da Şekil 5.19'da verilmiştir.

Çizelge 5.23. Senaryo 4: Mesafe-orta ve düğüm sayısı-az olduğu durumda dört yöntemin veri gönderme sıklığına göre paket ulaştırma oran değerleri.

Veri Gönderme Sıklığı (sn)	Paket Ulaştırma Oranı			
	DCT	SPIHT	LEICA	GSD-EDUKA
15	0,7829166	0,7708333	0,7918750	0,7918757
9	0,7578645	0,7135416	0,7451354	0,7578645
3	0,6822916	0,6927083	0,7060416	0,7060416

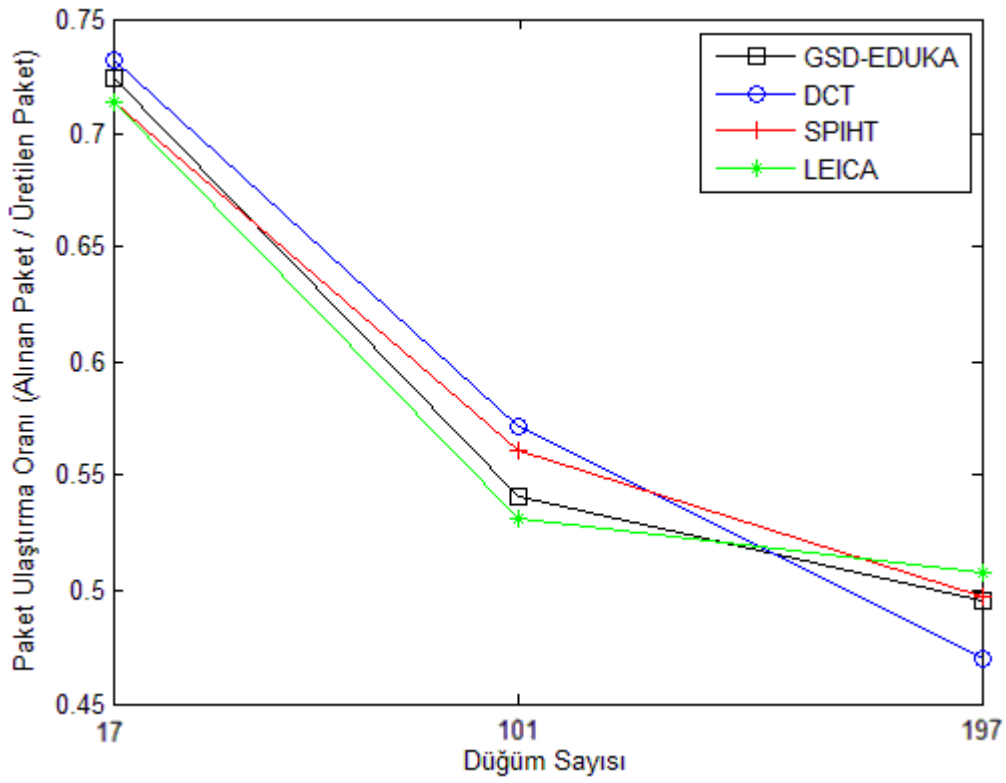


Şekil 5.19. Senaryo 4: Mesafe-orta ve düğüm sayısı-az olduğu durumda dört yöntemin veri gönderme sıklığına göre paket ulaştırma oran grafiği.

Senaryo 14 için elde edilen değerler çizelge olarak Çizelge 5.24’de, grafik olarak da Şekil 5.20’de verilmiştir.

Çizelge 5.24. Senaryo 14: Mesafe-orta ve veri gönderme sıklığı-orta olduğu durumda dört yöntemin düğüm sayısına göre paket ulaştırma oran değerleri.

Düğüm Sayısı	Paket Ulaştırma Oranı			
	DCT	SPIHT	LEICA	GSD-EDUKA
17	0,7322916	0,7135416	0,7135416	0,7225698
101	0,5714648	0,5605468	0,5313671	0,5459852
197	0,4693287	0,4965277	0,5074074	0,4941598

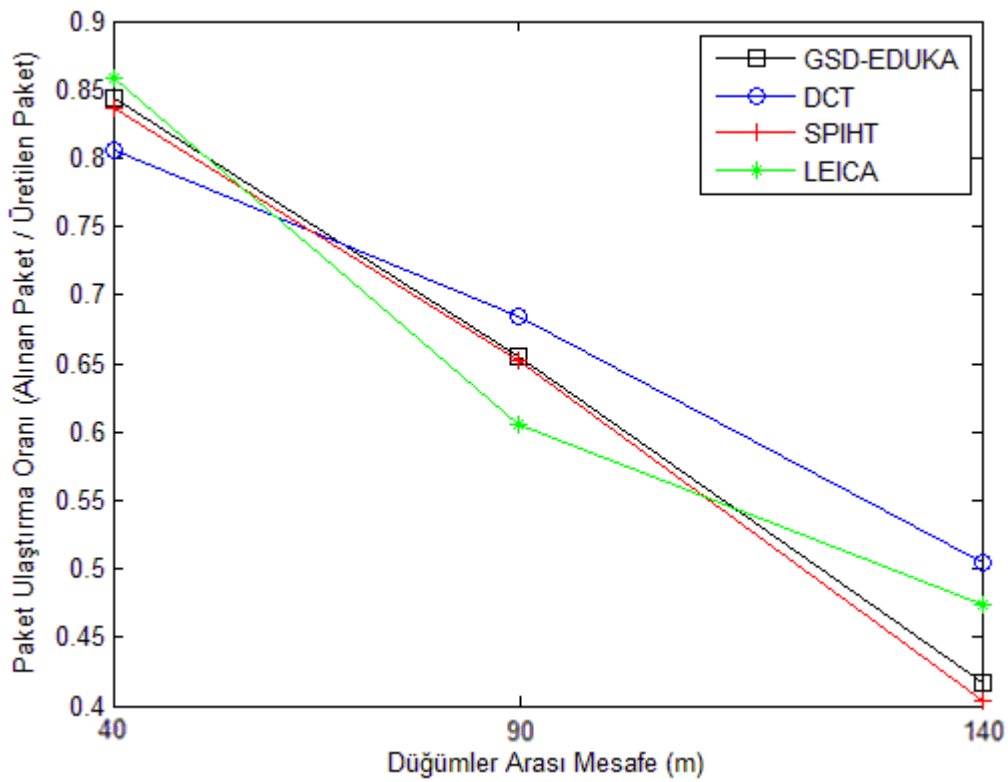


Şekil 5.20. Senaryo 14: Mesafe-orta ve veri gönderme sıklığı-orta olduğu durumda dört yöntemin düğüm sayısına göre paket ulaştırma oran grafiği.

Senaryo 22 için elde edilen değerler çizelge olarak Çizelge 5.25'te, grafik olarak da Şekil 5.21'de verilmiştir.

Çizelge 5.25. Senaryo 22: Düğüm sayısı-orta ve veri gönderme sıklığı-az olduğu durumda dört yöntemin mesafeye göre paket ulaştırma oran değerleri.

Mesafe (m)	Paket Ulaştırma Oranı			
	DCT	SPIHT	LEICA	GSD-EDUKA
40	0,8565972	0,8568287	0,8590601	0,858112
90	0,6973379	0,6556944	0,6117361	0,6656944
140	0,5092740	0,4147480	0,4732252	0,4214804



Şekil 5.21. Senaryo 22: Düğüm sayısı-orta ve veri gönderme sıklığı-az olduğu durumda dört yöntemin mesafeye göre paket ulaştırma oran grafiği.

Elde edilen PUO değerleri, yakın parametreleri ve aynı yönlendirme protokolünü (AODV) kullanarak geliştirilen [118], [119] ve [120]'de ki benzetim çalışmaları ile yakınlık göstermektedir.

Bu çalışmada, POU ile ilgili bir iyileştirme söz konusu değildir. Geçekleştirilen benzetimlerde de görüldüğü üzere GSD-EDUKA'nın PUO değeri, diğer görüntü sıkıştırma algoritmalarına göre önemli ölçüde kötü değerlere sahip değildir. DCT, SPIHT ve LEICA algoritmaları sıkıştırma işlemi sonucunda farklı boyutlar ürettiklerinden dolayı farklı sonuçlar üretilmiştir.

5.3.5 Harcanan Enerji Değerleri

Gerçekleştirilen senaryolarda harcanan enerji değerleri sabit ve hareketli ağlarda olmak üzere ayrı ayrı hesaplanmıştır. Elde edilen değerler aşağıda verilmiştir.

Sabit ağda senaryo 3 (düğümler arası mesafe az, düğüm sayısı çok ve veri gönderme sıklığı değişken), senaryo 14 (düğümler arası mesafe orta, düğüm sayısı değişken ve veri gönderme sıklığı orta) ve senaryo 22'ye (düğümler arası mesafe değişken, düğüm sayısı orta ve veri gönderme sıklığı az) göre GSD-EDUKA ve sıkıştırma uygulamadan harcanan enerji değerleri Çizelge 5.26'da verilmiştir.

Çizelge 5.26. Sabit ağda senaryo 3, 14 ve 22 için GSD-EDUKA ve sıkıştırma uygulamadan harcanan toplam enerji değerleri.

	Harcanan Toplam Enerji (joule)	
	GSD-EDUKA	Sıkıştırma Uygulamadan
Senaryo 3	8786,7094	17993,6416
Senaryo 14	3284,1789	5727,6362
Senaryo 22	5344,2465	10526,2087

Bu durumda, sabit ağlarda gerçekleştirilen benzetimler için senaryo 3'te %51,16, senaryo 14'de %42,66 ve senaryo 33'te ise %49,22'lik enerji verimi sağlandığı görülmektedir.

Hareketli ağda senaryo 6 (düğümler arası mesafe orta, düğüm sayısı çok ve veri gönderme sıklığı değişken), senaryo 16 (düğümler arası mesafe az, düğüm sayısı değişken ve veri gönderme sıklığı çok) ve senaryo 24'e göre (düğümler arası mesafe değişken, düğüm sayısı orta ve veri gönderme sıklığı çok) GSD-EDUKA ve sıkıştırma uygulamadan harcanan enerji değerleri Çizelge 5.27'de verilmiştir.

Çizelge 5.27. Hareketli ağda senaryo 6, 16 ve 24 için GSD-EDUKA ve sıkıştırma uygulamadan harcanan toplam enerji değerleri.

	Harcanan Toplam Enerji (joule)	
	GSD-EDUKA	Sıkıştırma Uygulamadan
Senaryo 6	6351,3718	9156,4681
Senaryo 16	3519,0629	5290,1201
Senaryo 24	3376,4031	5349,6056

Bu durumda, hareketli ağlarda gerçekleştirilen benzetimler için senaryo 6'da %30,63, senaryo 16'da %33,47 ve senaryo 24'de ise %36,88'lik enerji verimliliği sağlanmaktadır.

5.3.6 Uçtan Uca Gecikme

Tez çalışmasında geliştirilen GSD-EDUKA'nın ağ üzerinde sebep olacağı gecikme senaryo 11'e göre hesaplanmıştır. Bu hesaplamada senaryo 11'in düğümler arası mesafesi 100 metre, düğüm sayısı 15 ve veri gönderme sıklığının ise 10 saniye olduğu durum ile hareketli ağda görüntünün iletimi baz alınmıştır. Senaryo süresi 120 saniyedir.

Çizelge 5.28. GSD-EDUKA ile senaryo 11 boyunca ölçüm yapılan zamana göre düğümlerin kullandıkları sıkıştırma yöntemleri.

Ölçüm Yapılan Zaman	Düğüm indeksi														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
10	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D
20	D	D	D	D	D	D	D	D	D	D	D	D	D	D	D
30	D	S	S	S	S	D	L	S	D	D	D	D	L	L	D
40	D	S	L	L	L	S	D	L	S	S	S	D	D	D	S
50	S	L	S	S	D	L	S	D	S	L	S	S	S	S	S
60	S	D	L	L	S	D	L	S	L	D	L	L	L	L	L
70	L	L	D	S	S	S	D	S	S	D	D	D	D	D	S
80	D	S	S	L	L	L	D	L	D	D	S	S	S	D	L
90	L	L	L	S	D	D	S	D	D	S	L	S	L	S	D
100	S	S	S	L	S	D	D	L	S	L	D	L	D	S	S
110	S	S	L	L	L	S	S	D	L	D	L	D	L	L	S
120	L	L	L	S	S	S	S	S	D	D	D	D	D	D	L

Çizelge 5.28'de geliştirilen senaryo 11 için benzetim süresi boyunca belirli zamanlarda (10. saniye, 20. saniye, vs) her bir düğümün kullanmak için seçtiği sıkıştırma yöntemi kısaltmaları ile verilmiştir. D: DCT, S: SPIHT ve L: LEICA'yı ifade etmektedir. Bir düğümün aldığı görüntüyü sıkıştırması DCT ile 54,843 ms, SPIHT ile 57,968 ms ve LEICA ile 54,707 ms sürmektedir (Bkz. Bölüm 4.2.1). GSD-EDUKA'nın hangi sıkıştırma yöntemini kullanacağına karar vermesi ise 1,6 μ s sürmektedir.

Çizelge 5.29'da senaryoda belirtilen 15 düğümün aldıkları görüntüyü sadece DCT, SPIHT, LEICA ve GSD-EDUKA kullanılarak sıkıştırırken harcadıkları zamanın toplam ortalama değerleri verilmiştir.

Çizelge 5.29. Senaryo 11 için kullanılan yöntemlere göre toplam ortalama sıkıştırma süreleri.

	Kullanılan Yöntem			
	DCT	SPIHT	LEICA	GSD-EDUKA
Ortalama Zaman (ms)	658,1160	695,6160	656,4840	669,7924

Bu değerlere göre GSD-EDUKA'nın sıkıştırma işlemini, DCT ve LEICA'dan ortalama 12,4924 ms daha geç, ancak SPIHT'den 25,8236 ms daha erken tamamladığı görülmektedir. GSD-EDUKA, DCT ve LEICA ile karşılaştırıldığında gecikmeye sebep olmakta; fakat görüntünün boyutunu ortalama 13 kB daha aza indirmesi ile toplamda 104 *mjoule* enerji tasarrufu sağlamaktadır. Bu avantaj ise ağ ömrünün uzatılması hedefine erişmede gecikmeye göre daha etkili bir gelişmedir.

Ayrıca, uçtan uca gecikmeyi elde edebilmek için farklı senaryolarda üretilen bütün paketler üzerinde çalışmalar gerçekleştirilmiştir. Bu çalışmalarda, hedefine ulaşan tüm paketler kaynaktan çıktığı andan hedefine varıncaya kadar geçen zamanın ortalaması alınmıştır. DCT, SPIHT, LEICA ve GSD-EDUKA için bu işlem ayrı ayrı gerçekleştirilmiş ve kurulan ağda iletilen paket yolu kullanılan her algoritma için aynı olacak şekilde düzenlenmiştir. Yani, aynı ağda gerçekleştirilen benzetimlerde kullanılan her algoritma için paketlerin izlediği yol aynıdır. Bölüm 2.4'de hesaplanan görüntü sıkıştırma algoritmalarının sıkıştırma süreleri; DCT'nin 54,843 ms, SPIHT'in 57,968 ms ve LEICA'nın ise 54,707 ms'dir.

17 düğüm içeren bir senaryoda bir paketin kaynaktan hedefine ulaşıncaya kadar, her bir algoritma için geçen süre Çizelge 5.30'da verilmiştir. Bu senaryoda GSD-EDUKA en verimli görüntü sıkıştırma algoritması olarak DCT'ye karar vermiş ve kullanmıştır.

Çizelge 5.30. 17 düğümlük senaryo için kullanılan yöntemlere göre bir paketin uçtan uca iletilme süresi.

	Kullanılan Yöntem			
	DCT	SPIHT	LEICA	GSD-EDUKA
Geçen süre (ms)	403,800	406,925	403,664	404,488

Çizelge 5.30'a göre GSD-EDUKA, SPIHT'den daha iyi DCT ve LEICA'dan daha kötü uçtan uca gecikme değerine sahip olduğu görülmektedir.

101 düğüm içeren bir senaryoda bir paketin kaynaktan hedefine ulaşınca kadar her bir algoritma için geçen süre Çizelge 5.31’de verilmiştir. Bu senaryoda GSD-EDUKA en verimli görüntü sıkıştırma algoritması olarak SPIHT’e karar vermiş ve kullanmıştır.

Çizelge 5.31. 101 düğümlük senaryo için kullanılan yöntemlere göre bir paketin uçtan uca iletilme süresi.

	Kullanılan Yöntem			
	DCT	SPIHT	LEICA	GSD-EDUKA
Geçen süre (ms)	756,971	760,096	756,845	762,116

Çizelge 5.31’e göre GSD-EDUKA, diğer algoritmalarından daha kötü uçtan uca gecikme değerine sahip olduğu görülmektedir.

197 düğüm içeren bir senaryoda bir paketin kaynaktan hedefine ulaşınca kadar her bir algoritma için geçen süre Çizelge 5.32’de verilmiştir. Bu senaryoda GSD-EDUKA en verimli görüntü sıkıştırma algoritması olarak LEICA’ya karar vermiş ve kullanmıştır.

Çizelge 5.32. 197 düğümlük senaryo için kullanılan yöntemlere göre bir paketin uçtan uca iletilme süresi.

	Kullanılan Yöntem			
	DCT	SPIHT	LEICA	GSD-EDUKA
Geçen süre (ms)	1005,095	1008,968	1004,959	1006,364

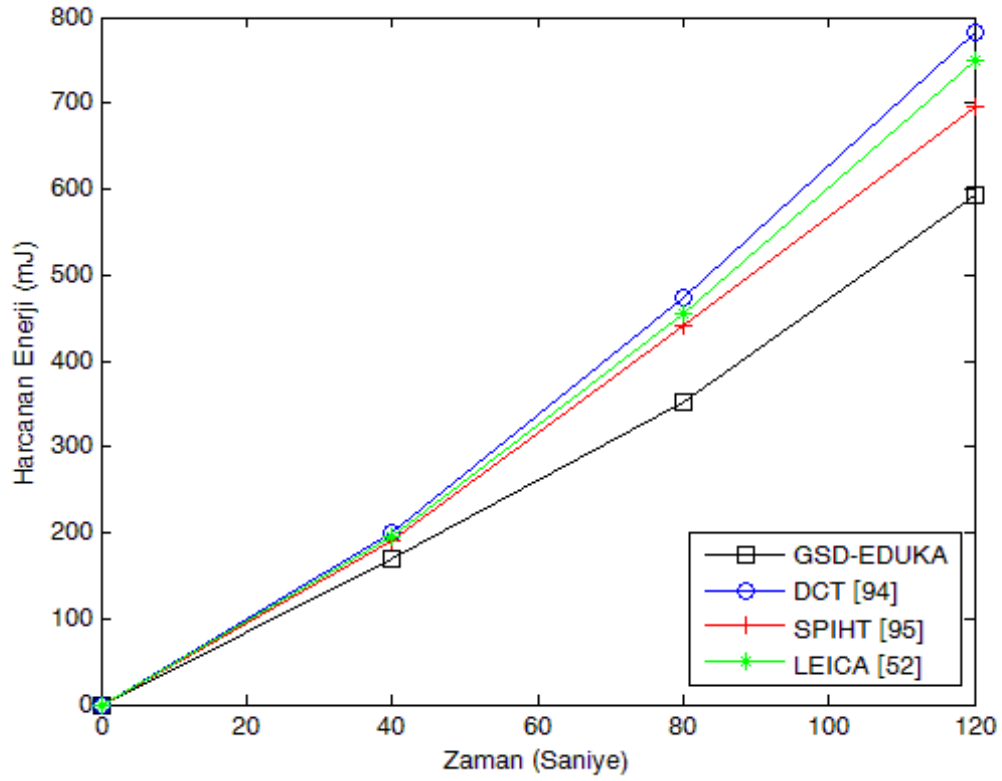
Çizelge 5.32’ye göre GSD-EDUKA, SPIHT’den daha iyi DCT ve LEICA’dan daha kötü uçtan uca gecikme değerine sahip olduğu görülmektedir.

GSD-EDUKA’nın diğer algoritmalarla göre bazı durumlarda sıkıştırma oranını artması işlem süresini uzatmaktadır. Ancak, görüntüyü daha fazla sıkıştırarak gönderilen paket sayısının azaltılması ile uçtan uca gecikmenin azalmasını da sağlayabilir.

5.3.7 Yöntemlerin Harcanan Enerjiye göre Karşılaştırılması

Bu bölümde referans alınan çalışmalardaki yöntemler (DCT, SPIHT ve LEICA), tez çalışmasında geliştirilen GSD-EDUKA ile bir düğümün görüntüyü sıkıştırıp iletirken harcadığı enerji bakımından karşılaştırılması yapılmıştır. Yapılan karşılaştırmada aynı özelliklerdeki ağ modeli benimsenmiştir. Yani, düğüm sayısı, düğümler arası mesafe ve veri gönderme sıklığının aynı olduğu bir ağda DCT [93], SPIHT [94] ve LEICA [51] yöntemleri uygulanarak, harcanan ortalama enerji hesaplanmıştır. Aynı şekilde, bahsedilen ağ modeli kullanılarak GSD-EDUKA’nın zamana bağlı olarak harcadığı

ortalama enerji hesaplanmıştır.



Şekil 5.22. Senaryo 3'e göre dört yöntemin harcadığı enerjilerinin zamana bağlı karşılaştırılması.

Şekil 5.22'de senaryo 3'deki ağ modeli (düğümler arası mesafe az, düğüm sayısı çok ve veri gönderme sıklığı değişken) temel alınarak GSD-EDUKA ve diğer (DCT [93], SPIHT [94] ve LEICA [51]) yöntemlerin bir düğümün harcadığı enerjinin zamana bağlı olarak karşılaştırması verilmiştir. Yöntemler belirli zaman dilimlerinde çalıştırıldıktan sonra ağdaki ortalama harcanan enerji hesaplanmıştır. Buna göre GSD-EDUKA'nın kendisine en yakın olan SPIHT algoritmasına göre 120 saniye gerçekleştirilen benzetimde en az 103 *mjoule* daha az enerji harcadığı görülmüştür.

Her bir düğüme atanan 1000 *joule* değer, en az enerji harcayan SPIHT algoritması ile ortalama 47,6 saat, GSD-EDUKA ile ise ortalama 55,5 saat kullanılabilir. Bu durumda kendisine en yakın olan SPIHT algoritmasına göre geliştirilen GSD-EDUKA'yı kullanan bir düğüm ortalama 7,9 saat daha fazla ömre sahiptir.

6. SONUÇLAR VE ÖNERİLER

6.1 SONUÇ

KMAA'lar, yüksek boyutlarda verilerin taşındığı, dolayısıyla enerji sarfiyatının skaler veri taşıyan geleneksel KAA'lara göre oldukça fazla olduğu ağlardır. Bir ağda en fazla enerji, verinin düğümler arasında iletimi sırasında harcanır. Bu sırada harcanan enerji verinin boyutu ile orantılıdır. Literatürde KMAA'da bu ve benzeri enerji problemlerini aşmak amacıyla, taşınan veriyi en kısa yoldan hedefe ulaştırma, düğümleri görevlerine göre uygun pozisyonlara yerleştirme, düğümleri uyutma gibi çözümler önerilmiştir. Bir diğer çözüm ise multimedya verilerinin sıkıştırılarak düşük boyutlu verilerin iletimidir. Böylece, daha düşük boyuttaki veri belirlenen uygun hat üzerinde taşınır. Bu da, optimal yönlendirme şeması, optimal düğüm dizilimi ya da boştaki düğümlerin uyutulması çözümlerinin çok daha üstünde bir enerji verimli yaklaşım sunar.

Bu tez çalışmasında KMAA için kullanılan mevcut popüler sıkıştırma yöntemlerine göre daha etkili bir yöntem önerilmiştir. Önerilen algoritma diğer algoritmalarından, görüntü aktarımı sırasında hangi sıkıştırma algoritmasının kullanacağına “düğümler arası mesafe”, “düğüm sayısı” ve “veri gönderme sıklığı” değişkenlerine bağlı olarak karar vermesi yönüyle ayrılmaktadır. Böylece, anlık duruma en uygun sıkıştırma algoritması seçilerek maksimum verim sağlanmaktadır. Yöntem, anlık duruma göre hangi sıkıştırma yöntemini kullanacağına karar vererek, ağın o anki durumundan bağımsız, her durumda aynı sıkıştırma yöntemini kullanan geleneksel yöntemlere göre daha az enerji harcamaktadır. GSD-EDUKA, sıkıştırma algoritması kullanmadan uygulanan yöntemle göre sabit ağlarda %51, hareketli ağlarda ise %36 enerji tasarrufu sağlanmaktadır.

Aşağıda yapılan çalışmalara, tezin sunumuna uygun olarak, kısaca değinilmiş ve elde edilen bulgular sunulmuştur.

1. KMAA'da kullanılan algılayıcı düğümlerin, alınan görüntüyü bir sonraki algılayıcı düğüme aktarmadan önce görüntüyü sıkıştırarak iletilen verinin boyutunun azaltması gerekmektedir. Bu sayede algılayıcı düğümlerin

bataryaları daha verimli kullanılarak ağın ömrünün de uzaması sağlanmaktadır (Bölüm 1).

2. Yeni bir algoritma tasarımı yapabilmek için, literatürde KMAA’da kullanılan görüntü sıkıştırma algoritmaları incelenmiş ve enerji bakımından hangi sıkıştırma algoritmasının daha iyi sonuç verdiği çıkarılmıştır (Bölüm 2).
3. Öne çıkan görüntü sıkıştırma algoritmalarının performanslarını ölçmek için kullanılacak Network Simulator – 2 (NS-2) benzetim aracına uyum sağlayan programlama dili ile kodlanmış ve performans değerleri ölçülerek, karşılaştırılmıştır (Bölüm 2).
4. KMAA için verimli olan görüntü sıkıştırma algoritmalarını kullanarak geliştirilen GSD-EDUKA algoritmasının ayrıntılı çalışma adımları verilmiş ve “düğüm sayısı”, “veri gönderme sıklığı” ile “düğümler arası mesafe” parametrelerinin değerleri belirlenmiştir. Belirlenen parametreler ile KMAA’da öne çıkan görüntü sıkıştırma algoritmaları üzerinde benzetimler gerçekleştirilerek, parametrelere göre hangi durumda hangi algoritmanın daha verimli olduğu belirlenmiştir (Bölüm 3).
 - a. Muhtemel tüm durumların sayısı “karar ağacı yöntemi” kullanılarak en aza indirilmiştir.
 - b. Belirlenen parametreler ile kullanılacak görüntü sıkıştırma algoritmasına karar vermek için bulanık mantık yöntemi kullanılmıştır.
5. KMAA’daki algılayıcı düğümlerin enerji tüketim değerleri üzerinde durulmuştur. Geliştirilen GSD-EDUKA’nın harcadığı enerji ifade edilerek, fazladan kullanılan GPS modülünün harcadığı enerji, algoritmaların hesaplanma maliyeti ve algılayıcı düğümlerin var olan enerjisinden sağladığı tasarrufa değinilmiştir (Bölüm 4).
6. Geliştirilen GSD-EDUKA’nın benzetim aracı olan NS - 2’de performans değerlendirilmesi yapılmıştır. Bu değerlendirme için öncelikle belirlenen

görüntü sıkıştırma algoritmaları ayrı ayrı kullanılarak ve hiçbir sıkıştırma algoritması kullanılmadan benzetim gerçekleştirilmiştir. Daha sonra bu görüntü sıkıştırma algoritmalarını, belirlenen parametrelere göre anlık olarak seçme işlemine karar verebilecek bir şekilde geliştirilen GSD-EDUKA ile hiçbir görüntü sıkıştırma uygulamadan gerçekleşen sistem karşılaştırılmıştır (Bölüm 5).

6.2 ÇALIŞMANIN GETİRDİĞİ KATKILAR

Kablosuz multimedya ağlarda enerjinin en çok harcandığı veri iletimi sırasında sarf edilen enerji miktarını azaltma amacı taşıyan, anlık duruma göre en uygun görüntü sıkıştırma algoritmasına karar veren GSD-EDUKA'nın getirdiği katkılar aşağıda sıralanmıştır.

- a. KMAA'lar için görüntü sıkıştırmaya dayalı enerji duyarlı yeni bir algoritma geliştirilmiştir. GSD-EDUKA'nın görüntü sıkıştırma işlemini gerçekleştirmesinde kullanılacak yöntem KMAA'da kullanılan en verimli algoritmalarından olan DCT, SPIHT ve LEICA arasından ağırlıklı olarak durumuna uygun olarak seçilerek belirlenmiştir.
- b. Kablosuz iletişimde kabul edilebilir maksimum PSNR değer kaybına ulaşıncaya kadar sıkıştırma sağlanarak paketlerin minimum boyutlara ulaştırılması ve bu sayede iletimde harcanan enerjinin azaltılması amaçlanmıştır.
- c. Geliştirilen GSD-EDUKA algoritması sabit ağlarda SPIHT'e göre %3, LEICA'ya göre %3,6 ve DCT'ye göre %4, hareketli ağlarda ise SPIHT'e göre %2,7, LEICA'ya göre %3,1 ve DCT'ye göre %3,4 enerji tasarrufu sağlamaktadır.
- d. Gerçekleştirilen benzetimlerde her bir düğüme atanan 1000 *joule* değer ile GSD-EDUKA, her bir düğümün yaşam süresini ortalama 7,9 saat uzattığı görülmektedir.
- e. GSD-EDUKA kullanılırken ağ içerisindeki uçtan uca gecikme hesaplanmıştır. Düğümün hangi sıkıştırma yöntemini kullanacağına karar vermesi 1,6 μ s

sürmektedir. GSD-EDUKA'nın sıkıştırma işlemi, DCT ve LEICA'dan ortalama 12,4924 ms daha geç, ancak SPIHT'den 25,8236 ms daha erken tamamlanmaktadır. GSD-EDUKA, DCT ve LEICA ile karşılaştırıldığında gecikmeye sebep olmakta; fakat görüntünün boyutunu ortalama 13 kB daha azaltması ile toplamda 104 *mjoule* enerji tasarrufu sağlamaktadır. Bu avantaj ise ağ ömrünün uzatılması hedefine erişmede gecikmeye göre daha etkili bir gelişmedir.

- f. Bu tez çalışmasında herhangi bir iyileştirme amacı olmayan paket ulaştırma oranı farklı senaryolara göre gerçekleştirilen benzetimlerde GSD-EDUKA, karşılaştırılan DCT, SPIHT ve LEICA ile ortalama benzer sonuçlar vermiştir.
- g. Literatürde sıkıştırma algoritmalarının uygulandığı çalışmalar genellikle tek düğümün harcadığı enerji üzerine yoğunlaşmıştır. Bu çalışmada ise gerçek ağ ortamına daha fazla ifade eden çok düğümlü ağlar üzerinde durulmuştur.

6.3 TARTIŞMA VE ÖNERİLER

Tez çalışmasında geliştirilen algoritma ve benzetim sonuçlarından faydalanarak gelecekte yapılabilecek çalışmalara yol göstermek üzere aşağıdaki öneriler sunulmaktadır:

- a. Önerilen uygulama katmanına yönelik görüntü sıkıştırmaya dayalı enerji tasarrufu yöntemi diğer katmanlarda geliştirilmiş enerji tasarrufu ile alakalı yöntemlerle birleştirilerek çok daha uzun ömürlü melez bir ağ tasarımı gerçekleştirilebilir.
- b. Önerilen yöntemin etkinliği farklı çözünürlükteki kameralarla da test edilip, yüksek çözünürlüklü görüntüler alındığında sistemin etkinliği değerlendirilebilir ve yöntem çözünürlük parametresini de hesaba katacak şekilde daha genel bir hale dönüştürülebilir.
- c. Bu tez çalışmasında benzetim ortamları kullanılarak yapılan çalışmalar gerçek düğümlerin kullanıldığı ağlarda uygulanabilir.

- d. Kalan batarya ömrü, geliştirilen algoritmaya parametre olarak eklenebilir. Görüntü kalitesinin çok önemli olmadığı durumlarda, batarya azaldığında, iletilen veri miktarını daha da azaltarak ağ ömrünü maksimize etmek için görüntüler kabul edilen PSNR değerini aşacak şekilde, daha yüksek sıkıştırma oranlarında sıkıştırılabilirler. Böylece, mevcut batarya ile daha uzun süre görüntü iletimi sağlanmış olur.



KAYNAKLAR

- [1] I. F. Akyildiz, T. Melodia, and K. Chowdhury, "A survey on wireless multimedia sensor networks," *Computer Networks*, vol. 51, no. 4, pp. 921–960, 2007.
- [2] E. Magli, M. Mancin, and L. Merello, "Low-complexity video compression for wireless sensor networks," *International Conference on Multimedia and Expo ICME*, 2003, pp. 580-585.
- [3] X. Xiaojiang Du and H. Hsiao-Hwa Chen, "Security in wireless sensor networks," *IEEE Wireless Communication*, vol. 15, no. 4, pp. 60–66, Aug. 2008.
- [4] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, pp. 393–422, 2002.
- [5] G. Mali and S. Misra, "TRAST: Trust-Based Distributed Topology Management for Wireless Multimedia Sensor Networks," *IEEE Transactions on Computers*, vol. 65, no. 6, pp. 1978–1991, 2016.
- [6] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless Sensor Networks for Habitat Monitoring," *Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, Atlanta, Georgia, USA, 2002, pp. 88–97.
- [7] M. Cesana, A. Redondi, N. Tiglao, A. Grilo, J. M. Barcelo-Ordinas, M. Alaei, and P. Todorova, "Real-time multimedia monitoring in large-scale wireless multimedia sensor networks: Research challenges," *8th Euro-NF Conference Next Generation Internet NGI*, 2012, pp. 79–86.
- [8] M. Rahimi, R. Baer, O. I. Iroezi, J. C. Garcia, J. Warrior, D. Estrin, and M. Srivastava, "Cyclops: In Situ Image Sensing and Interpretation in Wireless Sensor Networks," *Proceedings of the 3rd international conference on Embedded networked sensor systems*, 2005, pp. 192–204.

- [9] T. Ma, M. Hempel, D. Peng, and H. Sharif, "A Survey of Energy-Efficient Compression and Communication Techniques for Multimedia in Resource Constrained Systems," *IEEE Commun. Survey and Tutorials*, vol. 15, no. 3, pp. 963–972, 2013.
- [10] I. F. Akyildiz, T. Melodia, and K. R. Chowdhury, "Wireless Multimedia Sensor Networks: Applications and Testbeds," *Proceeding of IEEE*, vol. 96, no. 10, pp. 1588–1605, 2008.
- [11] B. Harjito and S. Han, "Wireless Multimedia Sensor Networks Applications and Security Challenges," *Broadband, Wireless Computing, Communication and Applications (BWCCA)*, 2010, pp. 842–846.
- [12] X. Zhang, R. Yu, Y. Zhang, Y. Gao, M. Im, L. Cuthbert, and W. Wang, "Energy-efficient multimedia transmissions through base station cooperation over heterogeneous cellular networks exploiting user behavior," *IEEE Wireless Communication*, vol. 21, no. 4, pp. 54–61, 2014.
- [13] X. Wang, S. Wang, and D. Bi, "Compacted probabilistic visual target classification with committee decision in wireless multimedia sensor networks," *IEEE Sensors Journal*, vol. 9, no. 4, pp. 346–353, 2009.
- [14] Y. Sun, H. Luo, and S. K. Das, "A Trust-Based Framework for Fault-Tolerant Data Aggregation in Wireless Multimedia Sensor Networks," *IEEE Transaction Dependable Secure Computing*, vol. 9, no. 6, pp. 785–797, 2012.
- [15] D. Culler, D. Estrin, and M. Srivastava, "Guest Editors' introduction: Overview of sensor networks," *Computer*, vol. 37, no. 8, pp. 41–49, 2004.
- [16] S. Misra, M. Reisslein, and G. Xue, "A survey of multimedia streaming in wireless sensor networks," *IEEE Communications Survey and Tutorials*, vol. 10, no. 4, pp. 18–39, 2008.
- [17] J. Campbell, P. Gibbons, S. Nath, P. Pillai, S. Seshan, and R. Sukthankar, "IrisNet: an internet-scale architecture for multimedia sensors," *Proceedings of the 13th annual ACM international conference on Multimedia*, Hilton, Singapore, 2005, pp. 81–88.

- [18] R. Holman, J. Stanley, and T. Özkan-Haller, "Applying video sensor networks to nearshore environment monitoring," *IEEE Pervasive Computing*, vol. 2, no. 4, pp. 14–21, 2003.
- [19] S. M. Aziz, "Energy Efficient Image Transmission in Wireless Multimedia Sensor Networks," *IEEE Communications Letters*, vol. 17, no. 6, pp. 1084–1087, 2013.
- [20] Z. Hamid and F. B. Hussain, "QoS in Wireless Multimedia Sensor Networks: A Layered and Cross-Layered Approach," *Wireless Personal Communications*, vol. 75, no. 1, pp. 729–757, 2014.
- [21] S. Pudlewski, A. Prasanna, and T. Melodia, "Compressed-Sensing-Enabled Video Streaming for Wireless Multimedia Sensor Networks," *IEEE Transaction on Mobile Computing*, vol. 11, no. 6, pp. 1060–1072, 2012.
- [22] A. A. Kumar S., K. Ovsthus, and L. M. Kristensen., "An Industrial Perspective on Wireless Sensor Networks", A Survey of Requirements, Protocols, and Challenges," *IEEE Communication Survey and Tutorials*, vol. 16, no. 3, pp. 1391–1412, 2014.
- [23] C. Suh, Z. H. Mir, and Y.-B. Ko, "Design and implementation of enhanced IEEE 802.15.4 for supporting multimedia service in Wireless Sensor Networks," *Computer Networks*, vol. 52, no. 13, pp. 2568–2581, 2008.
- [24] M. Lin, J. Leu, W. Yu, M. Yu, and J. C. Wu, "On Transmission Efficiency of the Multimedia Service over IEEE 802.15.4 Wireless Sensor Networks," *13th International Conference on Advanced Communication Technology (ICACT)*, 2011, pp. 184–189.
- [25] N. Saxena, A. Roy, and J. Shin, "Dynamic duty cycle and adaptive contention window based QoS-MAC protocol for wireless multimedia sensor networks," *Computer Networks*, vol. 52, no. 13, pp. 2532–2542, 2008.
- [26] G. A. Shah, W. Liang, and X. Shen, "Cross-Layer Design for QoS Support in Wireless Multimedia Sensor Networks," *IEEE Global Telecommunications Conference GLOBECOM 2010*, 2010, pp. 1–5.

- [27] M. A. Yigitel, O. Durmaz Incel, and C. Ersoy, "Diff - MAC," *Proceedings of the 6th ACM workshop on QoS and security for wireless and mobile networks - Q2SWinet*, 2010, pp. 62.
- [28] K. Nguyen, T. Nguyen, C. K. Chaing, and M. Motani, "A Prioritized MAC Protocol for Multihop, Event-driven Wireless Sensor Networks," *International Conference on Communications and Electronics*, 2006, pp. 47–52.
- [29] J. Ben-Othman, S. Diagne, L. Mokdad, and B. Yahya, "Performance evaluation of a hybrid MAC protocol for wireless sensor networks," *Proceedings of the 13th ACM international conference on Modeling, analysis, and simulation of wireless and mobile systems - MSWIM '10*, 2010, pp. 327.
- [30] H. Kim and S.G. Min, "Priority-based QoS MAC protocol for wireless sensor networks," *IEEE International Symposium on Parallel & Distributed Processing*, 2009, pp. 1–8.
- [31] O. Farrag, M. Younis, and W. D. Amico, "MAC Support for Wireless Multimedia Sensor Networks," *GLOBECOM IEEE Global Telecommunications Conference*, 2009, pp. 1–6.
- [32] Y. Shouyi, L. Leibo, Z. Renyan, S. Zhongfu, and W. Shaojun, "Design of wireless multimedia sensor network for precision agriculture," *China Communication*, vol. 10, no. 2, pp. 71–88, 2013.
- [33] L. Savidge, Huang Lee, H. Aghajan, and A. Goldsmith, "QoS-based geographic routing for event-driven image sensor networks," *2nd International Conference on Broadband Networks*, 2005, pp. 68–77.
- [34] J. Ben-Othman and B. Yahya, "Energy efficient and QoS based routing protocol for wireless sensor networks," *Journal of Parallel Distributed Computing*, vol. 70, no. 8, pp. 849–857, 2010.
- [35] L. Cobo, A. Quintero, and S. Pierre, "Ant-based routing for wireless multimedia sensor networks using multiple QoS metrics," *Computer Networks*, vol. 54, no. 17, pp. 2991–3010, 2010.

- [36] X. Yan, L. Li, and F. J. An, "Multi-constrained routing in wireless multimedia sensor networks," in *2009 International Conference on Wireless Communications & Signal Processing*, 2009, pp. 1–5.
- [37] M. Maimour and Moufida, "Maximally radio-disjoint multipath routing for wireless multimedia sensor networks," *Proceedings of the 4th ACM workshop on Wireless multimedia networking and performance modeling (WMuNep)*, 2008, pp. 26-31.
- [38] M. Cao, L. T. Yang, X. Chen, and N. Xiong, "Node placement of linear wireless multimedia sensor networks for maximum network lifetime," *Proceedings of the 3rd international conference on Advances in grid and pervasive computing*, 2008, pp. 373–383.
- [39] P. Pace, V. Loscri, E. Natalizio, and T. Razafindralambo, "Nodes placement for reducing energy consumption in multimedia transmissions," *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC*, 2011, pp. 909–914.
- [40] O. B. Akan and I. F. Akyildiz, "Event-to-sink reliable transport in wireless sensor networks," *IEEE/ACM Transactions on Networking*, vol. 13, no. 5, pp. 1003–1016, 2005.
- [41] C.Y. Wan, S. B. Eisenman, and A. T. Campbell, "CODA," *Proceedings of the first international conference on Embedded networked sensor systems - SenSys'03*, 2003, p. 266.
- [42] S. M. Aghdam, V. Heidari, and M. Khansari, "EECA: Energy Efficient Congestion Avoidance in Wireless Multimedia Sensor Network," in *6th International Symposium on Telecommunications (IST)*, 2012, pp. 656–661.
- [43] C. Wang, K. Sohraby, and B. Li, "SenTCP: A Hop-by-Hop Congestion Control Protocol for Wireless Sensor Networks," *IEEE INFOCOM*, 2005.
- [44] F. Stann and J. Heidemann, "RMST: reliable data transport in sensor networks," *Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications*, 2003, pp. 102–112.

- [45] V. C. Gungor, O. B. Akan, and I. F. Akyildiz, "A Real-Time and Reliable Transport Protocol for Wireless Sensor and Actor Networks," *IEEE/ACM Transactions on Networking*, vol. 16, no. 2, pp. 359–370, 2008.
- [46] N. Tezcan and W. Wang, "ART: an asymmetric and reliable transport mechanism for wireless sensor networks," *International Journal of Sensor Networks*, vol. 2, no. 3/4, pp. 188–200, 2007.
- [47] H. Zhang, A. Arora, Y. Choi, and M. G. Gouda, "Reliable bursty convergecast in wireless sensor networks," *Computer Communication*, vol. 30, no. 13, pp. 2560–2576, 2007.
- [48] E. Gürses and Ö. B. Akan, "Multimedia communication in wireless sensor networks," *Annales Des Telecommunications*, vol. 60, no. 7, pp. 872–900, 2005.
- [49] X. Zhuo, K. Loo, J. Cosmas, and P. Yip, "Distributed video coding in wireless multimedia sensor network for multimedia broadcasting", *WSEAS Transaction on Communications*, vol. 7, no. 5, pp.418-427, 2008.
- [50] P. Wang, R. Dai, and I. F. Akyildiz, "Collaborative Data Compression Using Clustered Source Coding for Wireless Multimedia Sensor Networks," *IEEE INFOCOM*, 2010, pp. 1–9.
- [51] E. Sun, X. Shen, and H. Chen, "A low energy image compression and transmission in wireless multimedia sensor networks," *Procedia Engineering*, vol. 15, pp. 3604–3610, 2011.
- [52] J. M. Shapiro, "Embedded Image Coding Using Zerotrees of Wavelet Coefficients," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3445–3462, 1993.
- [53] A. Said and W. A. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 243–250, 1996.

- [54] A. Fakhari and M. Fathy, "A Two Level Architecture for High Throughput DCT-Processor and Implementing on FPGA," *International Conference on Reconfigurable Computing and FPGAs*, 2010, pp. 115–120.
- [55] A. Abraham and G. K. Narendra, "Energy Aware Priority-Based Secure Image Transmission in Wireless Sensor Networks," *8th International Conference on Wireless Communications, Networking and Mobile Computing*, 2012, pp. 1–4.
- [56] T. Mulugeta, L. Shu, M. Hauswirth, M. Chen, T. Hara, and S. Nishio, "Secured two phase geographic forwarding protocol in wireless multimedia sensor networks," *GLOBECOM - IEEE Global Telecommunications Conference*, 2010.
- [57] H. Wang, D. Peng, W. Wang, H. Sharif, J. Wegiel, D. Nguyen, R. Bowne, and C. Backhaus, "Artificial Immune System based image pattern recognition in energy efficient Wireless Multimedia Sensor Networks," *Military Communications Conference*, 2008, pp. 1–7.
- [58] K. Akalu and K. Raimond, "Design and performance analysis of energy efficient technique for wireless multimedia sensor networks using machine learning algorithm," *World Congress on Information and Communication Technologies*, 2011, pp. 1127–1132.
- [59] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete Cosine Transform," *IEEE Transactions Computer*, vol. C-23, no. 1, pp. 90–93, 1974.
- [60] H. Zaineldin, M. A. Elhosseini, and H. A. Ali, "Image compression algorithms in wireless multimedia sensor networks: A survey," *Ain Shams Engineering Journal*, vol. 6, no. 2, pp. 481–490, 2015.
- [61] M. Yang and N. Bourbakis, "An overview of lossless digital image compression techniques," *48th Midwest Symposium Circuits Systems*, 2005, pp. 1099–1102.
- [62] J. Shukla, M. Alwani, and A. K. Tiwari, "A survey on lossless image compression methods," *International Conference on Computer Engineering and Technology*, 2010, pp. 136–141.

- [63] M. F. Carreto-Castro, J. M. Ramirez, J. L. Ballesteros, and D. Baez-Lopez, "Comparison of lossless compression techniques," *Proceedings of 36th Midwest Symposium on Circuits and Systems*, 1993, pp. 1268–1270.
- [64] Z. N. Li, (2016, May 12). *Fundamentals of Multimedia (Springer)* [Online]. Available: <http://www.springer.com/us/book/9783319052892>.
- [65] H. S. Samra, "Image Compression Techniques," *International Journal of Computer and Technology*, vol. 2 no. 2, pp. 49-52, 2012.
- [66] S. T. Welstead, *Fractal and wavelet image compression techniques*. SPIE Optical Engineering Press, 1999.
- [67] D. Saupe and R. Hamzaoui, "A review of the fractal image compression literature," *ACM SIGGRAPH Computer Graphics*, vol. 28, no. 4, pp. 268–276, 1994.
- [68] N. Thomos, N. V. Boulgouris, and M. G. Strintzis, "Optimized transmission of JPEG2000 streams over wireless channels," *IEEE Transactions Image Processing*, vol. 15, no. 1, pp. 54–67, 2006.
- [69] X. Li and J. Cai, "Robust Transmission of JPEG2000 Encoded Images Over Packet Loss Channels," *IEEE International Conference Multimedia Expo*, 2007, pp. 947–950.
- [70] P. Civcioglu Besdok and M. Alci, "Görüntülerdeki gauss gürültüsünü gidermek için akım taşıyıcılı ortalama alıcı süzgeç devresinin tasarımı." *Elektrik-Elektronik-Bilgisayar Mühendisliği Sempozyumu*, Bursa, Türkiye, 2002, ss.70-73
- [71] H. L. Eng and K. K. Ma, "Noise adaptive soft-switching median filter," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 242–251, 2001.
- [72] D. Santa-Cruz, T. Ebrahimi, J. Askelof, M. Larsson, and C. A. Christopoulos "JPEG 2000 still image coding versus other standards," *Applications of Digital Image Processing*, pp. 446–454, 2000.
- [73] L. Chew and L. Ang "Survey of image compression algorithms in wireless sensor networks," *Information Technology*, 2008, pp. 1–9.

- [74] Z.N. Li, M. S. Drew, and L. Jiangchuan, *Fundamentals of Multimedia*, Second Press, USA, 2004.
- [75] C. Chrysafis and A. Ortega, "Line-based, reduced memory, wavelet image compression," *IEEE Transactions of Image Processing*, vol. 9, no. 3, pp. 378–389, 2000.
- [76] A. Mammeri, B. Hadjou, and A. Khoumsi, "A Survey of Image Compression Algorithms for Visual Sensor Networks," *Sensor Networks*, vol. 2012, pp. 1–19, 2012.
- [77] A. Mallaiah, S. K. Shabbir, T. Subhashini, "An Spiht Algorithm With Huffman Encoder For Image Compression And Quality Improvement Using Retinex Algorithm," *International Journal Of Scientific & Technology Research*, Volume 1, Issue 5, June 2012.
- [78] S. Medouakh, Z. E. Baarir, "Entropy Encoding EBCOT (Embedded Block Coding with optimized Truncation) in JPEG 2000," *International Journal of Computer Science Issues*, vol. 8, no. 4, p. 1, 2011.
- [79] W. B. Huang, A. W. Y. SU, and Y. H. KUO, "VLSI Implementation of a Modified Efficient SPIHT Encoder," *IEICE Transaction Fundamentals*, vol. E89–A, no. 12, pp. 3613–3622, 2006.
- [80] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Transaction on Image Process.*, vol. 9, no. 7, pp. 1158–1170, 2000.
- [81] O. Ghorbel, I. Jabri, W. Ayedi, and M. Abid, "Experimental study of compressed images transmission through WSN," *Microelectronics (ICM)*, 2011, pp. 1–6.
- [82] M. J. Kirkby, "Fractals: Form, chance, and dimension, Benoit B. Mandelbrot, W. H. Freeman and Co.," *Earth Surf. Process.*, vol. 4, no. 1, pp. 98–98, 1979.
- [83] O. Ghorbel, W. Ayedi, M. W. Jmal, and M. Abid, "Images compression in WSN: Performance analysis," *IEEE 14th International Conference on Communication Technology*, 2012, pp. 1363–1368.

- [84] M. Nasri, A. Helali, H. Sghaier, and H. Maaref, "Energy-efficient wavelet image compression in Wireless Sensor Network," *Communication in Wireless Environments and Ubiquitous Systems: New Challenges (ICWUS)*, 2010, pp. 1–7, 2010.
- [85] M. Nasri, A. Helali, H. Sghaier, and H. Maaref, "Energy conservation for image transmission over wireless sensor networks," *Microelectronics (ICM)*, 2011, pp. 1–6.
- [86] W. Xing-Yuan, W. Na, and Z. Dou-Dou, "Fractal image coding algorithm using particle swarm optimisation and hybrid quadtree partition scheme," *IET Image Processing*, vol. 9, no. 2, pp. 153–161, 2015.
- [87] D. J. Jackson and S. J. Hannah, "Comparative analysis of image compression techniques," *25th Southeastern Symposium on System Theory*, 1993, pp. 513–517.
- [88] K. Kose, A. E. Cetin, U. Gudukbay, and L. Onural, "3D Model compression using Connectivity-Guided Adaptive Wavelet Transform built into 2D SPIHT," *Journal of Visual Communication and Image Representation*, vol. 21, no. 1, pp. 17–28, 2010.
- [89] W. C. Chia, L.-M. Ang, and K. P. Seng, "Multiview Image Compression for Wireless Multimedia Sensor Network Using Image Stitching and SPIHT Coding with EZW Tree Structure," *International Conference on Intelligent Human-Machine Systems and Cybernetics*, 2009, pp. 298–301.
- [90] F. Xiao, P. Zhang, L. Sun, J. Wang, and R. Wang, "Research on image compression and transmission mechanism for wireless multimedia sensor networks," *International Conference on Electrical and Control Engineering*, 2011, pp. 788–791.
- [91] T. Ma, P. Shrestha, M. Hempel, D. Peng, and H. Sharif, "Low-Complexity Image Coder/Decoder With An Approaching-Entropy Quad-Tree Search Code For Embedded Computing Platforms," *18th IEEE International Conference on Image Processing (ICIP)*, 2011, pp. 297–300.

- [92] Z. Cao, Z. Ji, and M. Hu, "An image sensor node for wireless sensor networks," *International Conference on Information Technology: Coding and Computing (ITCC)*, 2005, pp. 740–745.
- [93] R. Amutha and Y. A. V. Phamila, "Energy efficient image compression scheme using ZonalDCT for wireless sensor networks," *IET Chennai 3rd International Conference on Sustainable Energy and Intelligent Systems (SEISCON 2012)*, 2012, pp. 408–414.
- [94] H. ZainEldin, M. A. Elhosseini, and H. A. Ali, "A modified listless strip based SPIHT for wireless multimedia sensor networks," *Journal Computer and Electrical Engineering*, vol. 56, no. C, pp. 519–532, 2016.
- [95] S. P. Palaskar and N. A. Chavhan, "Design and Implementation of Energy Efficient Node Data Transmission by Using Mobile Sink Node," *Fourth International Conference on Communication Systems and Network Technologies*, 2014, pp. 79–82.
- [96] X. Liu, "An optimal distance based transmission strategy for lifetime maximization of wireless sensor networks," *IEEE Sensors Journal*, vol. 15, no. 6, pp. 3484–3491, 2015.
- [97] R. Ghazalian, A. Aghagolzadeh, and S. M. H. Andargoli, "Energy consumption minimization in wireless visual sensor networks using convex optimization," *7th International Symposium on Telecommunications (IST)*, 2014, pp. 312–315.
- [98] Y. Yao, Q. Cao, and A. V. Vasilakos, "EDAL: An Energy-Efficient, Delay-Aware, and Lifetime-Balancing Data Collection Protocol for Heterogeneous Wireless Sensor Networks," *IEEE/ACM Transaction on Networking*, vol. 23, no. 3, pp. 810–823, 2015.
- [99] S. Ehsan and B. Hamdaoui, "A Survey on Energy-Efficient Routing Techniques with QoS Assurances for Wireless Multimedia Sensor Networks," *IEEE Communications Surveys and Tutorials*, vol. 14, no. 2, pp. 265–278, 2012.

- [100] B. Y. Li and P. J. Chuang, "Geographic energy-aware non-interfering multipath routing for multimedia transmission in wireless sensor networks," *Information Science*, vol. 249, pp. 24–37, 2013.
- [101] V. Raghunathan, C. Schurgers, S. P. S. Park, and M. B. Srivastava, "Energy-aware wireless microsensor networks," *IEEE Signal Processing Magazine*, vol. 19, no. 2, pp. 40–50, 2002.
- [102] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy efficient communication protocol for wireless microsensor networks," *33rd Annual Hawaii International Conference on System Sciences*, 2000, pp. 3005–3014.
- [103] Anonim, (2016, May 20). *INTEL XScale PXA255 Timesys Embedded Linux.*" [Online]. Available: https://linuxlink.timesys.com/products/intel_xscale_pxa255.
- [104] S. Giordano, I. Stojmenovic, and L. Blazevic, "Position based routing algorithms for ad hoc networks: A taxonomy," *Ad Hoc Wireless Networking.*, pp. 103–136, 2003.
- [105] Z. Wei, S. Lijuan, G. Jian, and L. Linfeng, "Image compression scheme based on PCA for wireless multimedia sensor networks," *The Journal of China Universities Posts and Telecommunications*, vol. 23, no. 1, pp. 22–30, 2016.
- [106] D. Kandris, M. Tsagkaropoulos, I. Politis, A. Tzes, and S. Kotsopoulos, "Energy efficient and perceived QoS aware video routing over Wireless Multimedia Sensor Networks," *Ad Hoc Networks*, vol. 9, no. 4, pp. 591–607, 2011.
- [107] A. Alanazi and K. Elleithy, "Energy efficient hidden node detection for improving Quality of Service in wireless multimedia sensor networks," *IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, 2016, pp. 1–6.
- [108] Y. A. Ur Rehman, M. Tariq, and T. Sato, "A Novel Energy Efficient Object Detection and Image Transmission Approach for Wireless Multimedia Sensor Networks," *IEEE Sensors Journal*, vol. 16, no. 15, pp. 5942–5949, 2016.

- [109] V. Kumar, A. Kumar, and A. Bhardwaj, "Performance evaluation of image compression techniques," *International Conference on Devices, Circuits and Systems (ICDCS)*, 2012, pp. 447–450.
- [110] H. Shen and G. Bai, "Routing in wireless multimedia sensor networks: A survey and challenges ahead," *Journal Network and Computer Applications*, vol. 71, pp. 30–49, 2016.
- [111] M. Patel and A. Shah, "Energy Efficient Target Tracking in Wireless Sensor Networks," *International Journal of Advances in Engineering and Technology*, vol. 3, no. 2, pp. 542–549, 2012.
- [112] R. M. Kaur, Gurwinder and Garg, "Energy efficient topologies for wireless sensor networks," *International Journal of Distributed and Parallel Systems (IJDPS)*, pp. 179–192, 2012.
- [113] I. H. Peng and Y. W. Chen, "Energy consumption bounds analysis and its applications for grid based wireless sensor networks," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 444–451, 2013.
- [114] M. Younis and K. Akkaya, "Strategies and techniques for node placement in wireless sensor networks: A survey," *Ad Hoc Networks*, vol. 6, no. 4, pp. 621–655, 2008.
- [115] P. Wang, H. Hou, X. He, C. Wang, T. Xu, and Y. Li, "Survey on Application of Wireless Sensor Network in Smart grid," *Procedia Computer Science*, vol. 52, pp. 1212–1217, 2015.
- [116] K. Malarvizhi, M. Brindha, and M. Kumar, "Evaluation of energy efficient routing in wireless multimedia sensor networks," *2nd International Conference on Electronics and Communication Systems (ICECS)*, 2015, pp. 1387–1391.
- [117] T. F. Silva and D. G. Costa, "Centralized Algorithms for Redundant Coverage Maximization in Wireless Visual Sensor Networks," *IEEE Latin America Transactions*, vol. 14, no. 7, pp. 3378–3384, 2016.

- [118] M. Liu, S. Xu, and S. Sun, “An agent-assisted QoS-based routing algorithm for wireless sensor networks,” *Journal of Network and Computer Applications*, vol. 35, no. 1, pp. 29–36, 2012.
- [119] J. Agrakhed, G. S. Biradar, and V. D. Mytri, “Adaptive Multi Constraint Multipath Routing Protocol in Wireless Multimedia Sensor Network,” *International Conference on Computing Sciences*, 2012, pp. 326–331.
- [120] O. Akan, “Performance of Transport Protocols for Multimedia Communications in Wireless Sensor Networks,” *IEEE Communications Lettters*, vol. 11, no. 10, pp. 826–828, 2007.



EKLER

A. KARAR AĞACI YÖNTEMİ

GSD-EDUKA için gereken düğümler arası mesafe, ağdaki toplam düğüm sayısı ve veri gönderme sıklığı parametreleri belirlendikten sonra bu parametrelerin durumlarına göre kullanılacak algoritmalar belirlenmesi gerekmektedir.

Bölüm 3.6.1, 3.6.2, 3.6.3 ve 3.6.4'den elde edilen benzetim sonuçlarına göre;

- Düğümler arası mesafenin az olduğu durumda DCT algoritmasının, orta olduğu durumda LEICA algoritmasının, çok olduğu durumda SPIHT algoritmasının kullanılması uygun olacağı görülmektedir.
- Düğüm sayısının az olduğu durumda DCT algoritmasının, orta olduğu durumda SPIHT algoritmasının, çok olduğu durumda LEICA algoritmasının kullanılması uygun olacağı görülmektedir.
- Veri gönderme yoğunluğunun az olduğu durumda SPIHT algoritmasının, orta olduğu durumda DCT algoritmasının, çok olduğu durumda LEICA algoritmasının kullanılması uygun olacağı görülmektedir.

Bu sonuçlara göre oluşabilecek durum sayısı her bir parametre için 3 durum içerdiğine göre 3 parametre için toplamda 27 durum oluşmaktadır. Bu durumlar tümü Çizelge A.1'de verilmiştir.

Çizelge A.1. Toplam oluşabilecek durumlar.

Durum	Düğümler Arası Mesafe	Düğüm Sayısı	Veri Gönderme Sıklığı
1	Az	Az	Az
2	Az	Az	Orta
3	Az	Az	Çok
4	Az	Orta	Az
5	Az	Orta	Orta
6	Az	Orta	Çok
7	Az	Çok	Az
8	Az	Çok	Orta
9	Az	Çok	Çok
10	Orta	Az	Az
11	Orta	Az	Orta
12	Orta	Az	Çok
13	Orta	Orta	Az
14	Orta	Orta	Orta
15	Orta	Orta	Çok
16	Orta	Çok	Az
17	Orta	Çok	Orta
18	Orta	Çok	Çok
19	Çok	Az	Az
20	Çok	Az	Orta
21	Çok	Az	Çok
22	Çok	Orta	Az
23	Çok	Orta	Orta
24	Çok	Orta	Çok
25	Çok	Çok	Az
26	Çok	Çok	Orta
27	Çok	Çok	Çok

GSD-EDUKA parametreleri sonucunda elde edilen bu 27 durum, bulanık mantık yönteminde kural tabanını oluşturmak için bilgi olarak kullanılacaktır. Elde edilen bu durumlar karar ağacı yöntemi kullanarak daha aza indirilebilir.

Karar ağacı oluşturmak için öncelikle sistemin entropi değeri hesaplanır. Entropi değerine bağlı olarak da, sistemin sahip olduğu bütün parametreler için ayrı ayrı bilgi kazancı değeri hesaplanır. Bilgi kazancı değeri yüksek olan parametre oluşturulacak olan karar ağacında en üstte yer alır. Bu şekilde karar verilen birinci kök parametresi karar ağacına yerleştirilmiş olur. Geriye kalan parametreler ile işlem tekrarlanarak ağaç oluşturulur. Hesaplanan bilgi kazancı değerlerinin aynı çıkması sonucunda parametrelerden herhangi biri seçilebilir. Bu durum karar ağacının güvenilirliğini kaybetmemektedir, sonuç olarak aynı değer atanmaktadır.

Entropi değeri Denklem (A.1)'deki, bilgi kazancı değeri ise Denklem (A.2)'deki eşitlik ile elde edilmektedir.

$$E(S_1, S_2, \dots, S_m) = - \sum_{i=1}^m \frac{S_i}{S} \log_2 \left(\frac{S_i}{S} \right) \quad (A.1)$$

$m = \text{sınıf sayısı}$

$$\text{BilgiKazancı}(S, A) = \text{Entropy}(S) - \sum P_V * \text{Entropy}(P_V) \quad (A.2)$$

$v = A' \text{nin değerleri}$

A.1. Karar Ağacının Birinci Kökünü Belirleme

Karar ağacında 27 adet örnek mevcuttur, bu örneklerin değişken dağılımı Çizelge A.1'den elde edilerek Çizelge A.2'de verilmiştir.

Çizelge A.2. Birinci kök için değişken değer dağılımı.

Değişken	Değer
DCT	8
SPIHT	9
LEICA	10

Birinci kök için entropi değeri Denklem (A.3)'te hesaplanmıştır. Birinci kökü bulmak için gerçekleştirilecek bütün işlemlerde bu entropi değeri kullanılacaktır.

$$E(S_1, S_2, S_3) = - (8/27) \log_2(8/27) - (9/27) \log_2(9/27) - (10/27) \log_2(10/27) \quad (A.3)$$

$$= 0.4753$$

A.1.1. Düğümler arası mesafe için bilgi kazancı ve entropi değerleri hesabı

Düğümler arası mesafe için elde edilen veri seti Çizelge A.3'te verilmiştir.

Çizelge A.3. D ğ mler arası mesafe i in veri seti.

Durum	DCT	SPIHT	LEICA	Toplam Değer
Az	5	1	3	9
Orta	2	1	6	9
�ok	1	7	1	9

$$\begin{aligned}
 E(S_{az}) &= -(5/9) \log_2(5/9) - (1/9) \log_2(1/9) - (3/9) \log_2(3/9) \\
 &= 0.4068
 \end{aligned}
 \tag{A.4}$$

$$\begin{aligned}
 E(S_{orta}) &= -(2/9) \log_2(2/9) - (1/9) \log_2(1/9) - (6/9) \log_2(6/9) \\
 &= 0.3685
 \end{aligned}
 \tag{A.5}$$

$$\begin{aligned}
 E(S_{ ok}) &= -(1/9) \log_2(1/9) - (7/9) \log_2(7/9) - (1/9) \log_2(1/9) \\
 &= 0.2969
 \end{aligned}
 \tag{A.6}$$

$$\begin{aligned}
 &Bilgi Kazancı(S, Mesafe) = \\
 &0.4753 - \left(((9/27) * 0.4068) + ((9/27) * 0.3685) + ((9/27) * 0.2969) \right) \\
 &= 0.1179
 \end{aligned}
 \tag{A.7}$$

A.1.2. D ğ m sayısı i in bilgi kazancı ve entropi deėerleri hesabı

D ğ m sayısı i in elde edilen veri seti  izelge A.4'de verilmiřtir.

 izelge A.4. D ğ m sayısı i in veri seti.

Durum	DCT	SPIHT	LEICA	Toplam Değer
Az	5	2	2	9
Orta	2	5	2	9
�ok	1	2	6	9

$$E(S_{az}) = -(5/9) \log_2(5/9) - (2/9) \log_2(2/9) - (2/9) \log_2(2/9) \quad (A.8)$$

$$= 0.4321$$

$$E(S_{orta}) = -(2/9) \log_2(2/9) - (5/9) \log_2(5/9) - (2/9) \log_2(2/9) \quad (A.9)$$

$$= 0.4321$$

$$E(S_{çok}) = -(1/9) \log_2(1/9) - (2/9) \log_2(2/9) - (6/9) \log_2(6/9) \quad (A.10)$$

$$= 0.3685$$

$$Bilgi Kazancı (S, Düğüm Sayısı) =$$

$$0.4753 - \left(((9/27) * 0.4321) + ((9/27) * 0.4321) + ((9/27) * 0.3685) \right) \quad (A.11)$$

$$= 0.0644$$

A.1.3. Veri gönderme sıklığı için bilgi kazancı ve entropi değerleri hesabı

Veri gönderme sıklığı için elde edilen veri seti Çizelge A.5'te verilmiştir.

Çizelge A.5. Veri gönderme sıklığı için veri seti.

Durum	DCT	SPIHT	LEICA	Toplam Değer
Az	1	5	3	9
Orta	6	2	1	9
Çok	1	2	6	9

$$E(S_{az}) = -(1/9) \log_2(1/9) - (5/9) \log_2(5/9) - (3/9) \log_2(3/9) \quad (A.12)$$

$$= 0.4068$$

$$E(S_{orta}) = -(6/9) \log_2(6/9) - (2/9) \log_2(2/9) - (1/9) \log_2(1/9) \quad (A.13)$$

$$= 0.3685$$

$$E(S_{çok}) = -(1/9) \log_2(1/9) - (2/9) \log_2(2/9) - (6/9) \log_2(6/9) \quad (A.14)$$

$$= 0.3685$$

$$Bilgi Kazancı (S, Veri Gönderme Sıklığı) =$$

$$0.4753 - \left(((9/27) * 0.4068) + ((9/27) * 0.3685) + ((9/27) * 0.3685) \right) \quad (A.15)$$

$$= 0.0940$$

A.1.4. Birinci kök için sonuç

Tüm parametreler için hesaplanan bilgi kazancı değerleri Çizelge A.6’da verilmiş olup, buna göre “düğümler arası mesafe” birinci kök olarak belirlenmiştir.

Çizelge A.6. Birinci kök için bilgi kazancı sonuçları.

Parametre	Bilgi Kazancı
Düğümler arası mesafe	0.1179
Düğüm sayısı	0.0644
Veri gönderme sıklığı	0.0940

“Düğümler arası mesafe” birinci kök olarak belirlendikten sonra “düğümler arası mesafe”nin “az”, “orta” ve “çok” olduğu durumların kök işlemi gerçekleşecektir. İlk olarak “az” durumu için kök işlemlerini ele alalım.

A.2. “Düğümler Arası Mesafe” için “Az” Durumunun Alt Kökleri Belirleme

“Az” durumu için 9 adet örnek mevcuttur, bu örneklerin değişken dağılımı Çizelge A.7’de verilmiştir.

Çizelge A.7. “Az” durumu için değişken değer dağılımı.

Değişken	Değer
DCT	5
SPIHT	1
LEICA	3

Entropi değeri Denklem (A.16)’da hesaplanmıştır. “Az” durumunun alt köklerini bulmak için gerçekleşecek bütün işlemlerde bu entropi değeri kullanılacaktır.

$$\begin{aligned}
E(S_{Az}) &= \\
&-(5/9) \log_2(5/9) - (1/9) \log_2(1/9) - (3/9) \log_2(3/9) \\
&= 0.4068
\end{aligned} \tag{A.16}$$

A.2.1. “Az” durumu için “düğüm sayısı” bilgi kazancı ve entropi değerleri hesabı

“Az” durumu için düğüm sayısı veri seti Çizelge A.8’de verilmiştir.

Çizelge A.8. “Az” durumu için düğüm sayısı veri seti.

Durum	DCT	SPIHT	LEICA	Toplam Değer
Az	3	0	0	3
Orta	1	1	1	3
Çok	1	0	2	3

$$\begin{aligned}
E(S_{az}) &= -(1/3) \log_2(1/3) \\
&= 0.1590
\end{aligned} \tag{A.17}$$

$$\begin{aligned}
E(S_{orta}) &= -(1/3) \log_2(1/3) - (1/3) \log_2(1/3) - (1/3) \log_2(1/3) \\
&= 0.4771
\end{aligned} \tag{A.18}$$

$$\begin{aligned}
E(S_{çok}) &= -(1/3) \log_2(1/3) - (2/3) \log_2(2/3) \\
&= 0.2764
\end{aligned} \tag{A.19}$$

Bilgi Kazancı (S , Düğüm Sayısı) =

$$\begin{aligned}
&0.4068 - \left(((3/9) * 0.1590) + ((3/9) * 0.4771) + ((3/9) * 0.2764) \right) \\
&= 0.1026
\end{aligned} \tag{A.20}$$

A.2.2. “Az” durumu için “veri gönderme” bilgi kazancı ve entropi değerleri hesabı

“Az” durumu için veri gönderme veri seti Çizelge A.9’da verilmiştir.

Çizelge A.9. “Az” durumu için veri gönderme sıklığı veri seti.

Durum	DCT	SPIHT	LEICA	Toplam Değer
Az	1	1	1	3
Orta	3	0	0	3
Çok	1	0	2	3

$$E(S_{az}) = -(1/3) \log_2(1/3) - (1/3) \log_2(1/3) - (1/3) \log_2(1/3) \quad (A.21)$$

$$= 0.4771$$

$$E(S_{orta}) = -(3/3) \log_2(3/3) \quad (A.22)$$

$$= 0$$

$$E(S_{çok}) = -(1/3) \log_2(1/3) - (2/3) \log_2(2/3) \quad (A.23)$$

$$= 0.2764$$

$$Bilgi Kazancı (S, Veri Gönderme Sıklığı) =$$

$$0.4068 - \left(((3/9) * 0.4771) + ((3/9) * 0) + ((3/9) * 0.2764) \right) \quad (A.24)$$

$$= 0.1556$$

A.2.3. “Az” durumu için alt kök Sonucu

Tüm parametreler için hesaplanan bilgi kazancı değerleri Çizelge A.10’da verilmiştir.

Çizelge A.10. “Az” durumu için bilgi kazancı sonuçları.

Parametre	Bilgi Kazancı
Düğüm sayısı	0.1005
Veri gönderme sıklığı	0.1556

Çizelge A.10’a göre “veri gönderme sıklığı” alt kök olarak belirlenmiştir. Geriye kalan “düğüm sayısı” da üçüncü kök olarak belirlenmiştir.

A.3. “Düğüm Arası Mesafe” için “Orta” Durumunun Alt Kökleri Belirleme

“Orta” durumu için 9 adet örnek mevcuttur, bu örneklerin değişken dağılımı Çizelge A.11’de verilmiştir.

Çizelge A.11. “Orta” durumu için değişken değer dağılımı.

Değişken	Değer
DCT	2
SPIHT	1
LEICA	5

Entropi değeri Denklem (A.25)’de hesaplanmıştır. “Orta” durumunun alt köklerini bulmak için gerçekleştirilecek bütün işlemlerde bu entropi değeri kullanılacaktır

$$\begin{aligned}
 E(S_{Az}) &= \\
 &-(2/9) \log_2(2/9) - (1/9) \log_2(1/9) - (5/9) \log_2(5/9) \\
 &= 0.3930
 \end{aligned} \tag{A.25}$$

A.3.1. “Orta” durumu için “düğüm sayısı” bilgi kazancı ve entropi değerleri hesabı

“Orta” durumu için düğüm sayısı veri seti Çizelge A.12’de verilmiştir.

Çizelge A.12. “Orta” durumu için düğüm sayısı veri seti.

Durum	DCT	SPIHT	LEICA	Toplam Değer
Az	1	0	2	3
Orta	1	1	1	3
Çok	0	0	3	3

$$\begin{aligned}
 E(S_{az}) &= -(1/3) \log_2(1/3) - (2/3) \log_2(2/3) \\
 &= 0.2764
 \end{aligned} \tag{A.26}$$

$$E(S_{orta}) = -(1/3) \log_2(1/3) - (1/3) \log_2(1/3) - (1/3) \log_2(1/3) \quad (A.27)$$

$$= 0.4771$$

$$E(S_{çok}) = -(3/3) \log_2(3/3) \quad (A.28)$$

$$= 0$$

Bilgi Kazancı (S, Düğüm Sayısı) =

$$0.3930 - \left(((3/9) * 0.2764) + ((3/9) * 0.4771) + ((3/9) * 0) \right) \quad (A.29)$$

$$= 0.1418$$

A.3.2. “Orta” durumu için “veri gönderme sıklığı” bilgi kazancı ve entropi değerleri hesabı

“Orta” durumu için veri gönderme veri seti Çizelge A.13’te verilmiştir.

Çizelge A.13. “Orta” durumu için veri gönderme sıklığı veri seti.

Durum	DCT	SPIHT	LEICA	Toplam Değer
Az	0	1	2	3
Orta	2	0	1	3
Çok	0	0	3	3

$$E(S_{az}) = -(1/3) \log_2(1/3) - (2/3) \log_2(2/3) \quad (A.30)$$

$$= 0.2764$$

$$E(S_{orta}) = -(2/3) \log_2(2/3) - (1/3) \log_2(1/3) \quad (A.31)$$

$$= 0.2764$$

$$E(S_{çok}) = -(3/3) \log_2(3/3) \quad (A.32)$$

$$= 0$$

Bilgi Kazancı (*S*, *Veri Gönderme Sıklığı*) =

$$0.3930 - \left(\left((3/9) * 0.2764 \right) + \left((3/9) * 0.2764 \right) + \left((3/9) * 0 \right) \right) \quad (\text{A.33})$$

$$= 0.2087$$

A.3.3. “Orta” durumu için alt kök sonucu

Tüm parametreler için hesaplanan bilgi kazancı değerleri Çizelge A.14’de verilmiş olup, buna göre “veri gönderme sıklığı” alt kök olarak belirlenmiştir. Geriye kalan “düğüm sayısı” da üçüncü kök olarak belirlenmiştir.

Çizelge A.14. “Orta” durumu için bilgi kazancı sonuçları.

Parametre	Bilgi Kazancı
Düğüm sayısı	0.1418
Veri Gönderme sıklığı	0.2087

A.4. “Düğüm Arası Mesafe” için “Çok” Durumunun Alt Kökleri Belirleme

“Çok” durumu için 9 adet örnek mevcuttur, bu örneklerin değişken dağılımı Çizelge A.15’de verilmiştir.

Çizelge A.15. “Çok” durumu için değişken değer dağılımı.

Değişken	Değer
DCT	1
SPIHT	1
LEICA	7

Entropi değeri Denklem (A.34)’de hesaplanmıştır. “Çok” durumunun alt köklerini bulmak için gerçekleştirilecek bütün işlemlerde bu entropi değeri kullanılacaktır

$$E(S_{Az}) =$$

$$-(1/9) \log_2(1/9) - (1/9) \log_2(1/9) - (7/9) \log_2(7/9) \quad (\text{A.34})$$

$$= 0.2969$$

A.4.1. “Çok” durumu için “düğüm sayısı” bilgi kazancı ve entropi değerleri hesabı

“Çok” durumu için düğüm sayısı veri seti Çizelge A.16’da verilmiştir.

Çizelge A.16. “Çok” durumu için düğüm sayısı veri seti.

Durum	DCT	SPIHT	LEICA	Toplam Değer
Az	1	2	0	3
Orta	0	3	0	3
Çok	0	2	1	3

$$E(S_{az}) = -(1/3) \log_2(1/3) - (2/3) \log_2(2/3) \quad (A.35)$$

$$= 0.2764$$

$$E(S_{orta}) = -(3/3) \log_2(3/3) \quad (A.36)$$

$$= 0$$

$$E(S_{çok}) = -(2/3) \log_2(2/3) - (1/3) \log_2(1/3) \quad (A.37)$$

$$= 0.2764$$

$$\text{Bilgi Kazancı } (S, \text{Düğüm Sayısı}) =$$

$$0.2969 - \left(((3/9) * 0.2764) + ((3/9) * 0) + ((3/9) * 0.2764) \right) \quad (A.38)$$

$$= 0.0853$$

A.4.2. “Çok” durumu için “veri gönderme sıklığı” bilgi kazancı ve entropi değerleri hesabı

“Çok” durumu için veri gönderme veri seti Çizelge A.17’de verilmiştir.

Çizelge A.17. “Çok” durumu için veri gönderme sıklığı veri seti.

Durum	DCT	SPIHT	LEICA	Toplam Değer
Az	0	3	0	3
Orta	1	2	0	3
Çok	0	2	1	3

$$E(S_{az}) = -(3/3) \log_2(3/3) = 0 \quad (A.39)$$

$$E(S_{orta}) = -(1/3) \log_2(1/3) - (2/3) \log_2(2/3) = 0.2764 \quad (A.40)$$

$$E(S_{çok}) = -(2/3) \log_2(2/3) - (1/3) \log_2(1/3) = 0.2764 \quad (A.41)$$

$$\begin{aligned} \text{Bilgi Kazancı } (S, \text{Veri Gönderme Sıklığı}) &= \\ 0.2969 - \left(((3/9) * 0) + ((3/9) * 0.2764) + ((3/9) * 0.2764) \right) &= 0.1126 \end{aligned} \quad (A.42)$$

A.4.3. “Çok” durumu için alt kök Sonucu

Tüm parametreler için hesaplanan bilgi kazancı değerleri Çizelge A.18’de verilmiştir.

Çizelge A.18. “Çok” durumu için bilgi kazancı sonuçları.

Parametre	Bilgi Kazancı
Düğüm Sayısı	0.0853
Veri Gönderme Sıklığı	0.1126

Çizelge A.18’e göre “veri gönderme sıklığı” alt kök olarak belirlenmiştir. Geriye kalan “düğüm sayısı” da üçüncü kök olarak belirlenmiştir.

B. NETWORK SIMULATOR – 2 (NS-2) KODLARI

B.1. Hareketli Ağlar için NS-2 Kodları

```
set opt(chan) Channel/WirelessChannel
set opt(prop) Propagation/TwoRayGround
set opt(netif2) Phy/WirelessPhy;
set opt(mac) Mac/802_11
set opt(ifq) Queue/DropTail/PriQueue
set opt(ll) LL
set opt(ant) Antenna/OmniAntenna
set opt(x) 8000 ;# X dimension of the topography
set opt(y) 8000 ;# Y dimension of the topography
set opt(ifqlen) 50 ;# max packet in ifq
set opt(seed) 0.0
set opt(stop) 120.0 ;# simulation time
set opt(tr) Tez.tr ;# trace file
set opt(nam) Tez.nam ;# animation file
set opt(rp) AODV ;# routing protocol script
set opt(lm) "on" ;# log movement
set opt(energymodel) EnergyModel ;
set opt(radiomodel) RadioModel ;
set opt(initialenergy) 1000 ;# Initial energy in Joules
#set opt(logenergy) "on" ;# log energy every 150 seconds
Mac/802_11 set dataRate_ 54.0Mb;
set ns_ [new Simulator]
set topo [new Topography]
set tracefd [open $opt(tr) w]
set namtrace [open $opt(nam) w]
set prop [new $opt(prop)]
$topo load_flatgrid $opt(x) $opt(y)
ns-random 1.0
$ns_ trace-all $tracefd
$ns_ namtrace-all-wireless $namtrace 8000 8000

set god_ [create-god $opt(nn)]
$opt(netif2) set CPTthresh_ 10.0
$opt(netif2) set CSTthresh_ 8.91754e-10 ;#sensing range of 200m
$opt(netif2) set RXThresh_ 8.91754e-10 ;#communication range of 200m
$opt(netif2) set freq_ 914e+6 ;#914MHz
```

```
$ns_ node-config -adhocRouting $opt(rp) \
                -llType $opt(ll) \
                -macType $opt(mac) \
                -ifqType $opt(ifq) \
                -ifqLen $opt(ifqlen) \
```

```

        -antType $opt(ant) \
        -propType $opt(prop) \
        -phyType $opt(netif2) \
        -channelType $opt(chan) \
        -topoInstance $topo \
        -agentTrace ON \
        #-routerTrace ON \
        #-macTrace OFF \
        -energyModel $opt(energymodel) \
        -idlePower 0.05 \
        -sensePower 0.3 \
        -rxPower 0.5 \
        -txPower 1.0 \
        -sleepPower 0.002 \
        -initialEnergy $opt(initialenergy)
$nns_ set WirelessNewTrace_ ON
    for {set i 0} {$i < $opt(nn)} {incr i} {
        set node_($i) [$nns_ node]
        $node_($i) random-motion 1           ;# disable random motion
    }
set null_(0) [new Agent/Null]
$nns_ attach-agent $node_(0) $null_(0)

for {set i 1} {$i < $dugumsayisi} {incr i} {
    set udp_($i) [new Agent/UDP]
    $udp_($i) set packetSize_ 30000
    $nns_ attach-agent $node_($i) $udp_($i)

    set cbr_($i) [new Application/Traffic/CBR]
    $cbr_($i) set packetSize_
    $cbr_($i) set random_ 0
    $cbr_($i) set rate_ $opt(abc)
    $cbr_($i) set maxpkts_ 5000000
    $cbr_($i) attach-agent $udp_($i)
    $nns_ connect $udp_($i) $null_(0)
}
set kenardakidugumsayisi [expr { int(sqrt([expr $dugumsayisi-1])) }]

$node_(0) label "BAZ ISTASYONU"           ;# sets label on node
$nns_ initial_node_pos $node_(0) 20

for {set i 0} {$i < $kenardakidugumsayisi} {incr i} {
    for {set j 1} {$j <= $kenardakidugumsayisi} {incr j} {
        set nodeId [expr $i*$kenardakidugumsayisi + $j];
        $node_($nodeId) set X_ [expr $i*150+100];
        $node_($nodeId) set Y_ [expr $j*150+100];
        $node_($nodeId) set Z_ 0.0
        $nns_ initial_node_pos $node_($nodeId) 20
    }
}
}

```

```

source $val(sc)
for {set i 1} {$i < $dugumsayisi} {incr i} {
$ns_ at [expr sqrt($dugumsayisi)+0.5] "$cbr_($i) start"
$ns_ at 121.00 "$cbr_($i) stop"
}
for {set i 0} {$i < $opt(nn)} {incr i} {
$ns_ at $opt(stop) "$node_($i) reset";
}
$ns_ at $opt(stop) "puts \"NS EXITING...\" ; $ns_ halt"
puts "Starting Simulation..."
$ns_ run

```

B.2. Sabit Ağlar için NS-2 Kodları

```

set opt(chan) Channel/WirelessChannel
set opt(prop) Propagation/TwoRayGround
set opt(netif2) Phy/WirelessPhy;
set opt(mac) Mac/802_11
set opt(ifq) Queue/DropTail/PriQueue
set opt(ll) LL
set opt(ant) Antenna/OmniAntenna
set opt(x) 8000 ;# X dimension of the topography
set opt(y) 8000 ;# Y dimension of the topography
set opt(ifqlen) 50 ;# max packet in ifq
set opt(seed) 0.0
set opt(stop) 120.0 ;# simulation time
set opt(tr) Tez.tr ;# trace file
set opt(nam) Tez.nam ;# animation file
set opt(rp) AODV ;# routing protocol script
set opt(lm) "off" ;# log movement
#set opt(agent) Agent/AODV
set opt(energymodel) EnergyModel ;
set opt(radiomodel) RadioModel ;
set opt(initialenergy) 1000 ;# Initial energy in Joules
Mac/802_11 set dataRate_ 54.0Mb;

set ns_ [new Simulator]
set topo [new Topography]
set tracefd [open $opt(tr) w]
set namtrace [open $opt(nam) w]
set prop [new $opt(prop)]

$topo load_flatgrid $opt(x) $opt(y)
ns-random 1.0
$ns_ trace-all $tracefd
$ns_ namtrace-all-wireless $namtrace 8000 8000
$opt(netif2) set CPTthresh_ 10.0
$opt(netif2) set CSTthresh_ 8.91754e-10 ;#sensing range of 200m
$opt(netif2) set RXThresh_ 8.91754e-10 ;#communication range of 200m

```

```
$opt(netif2) set freq_ 914e+6 ;#914MHz
```

```
$ns_ node-config -adhocRouting $opt(rp) \  
-llType $opt(ll) \  
-macType $opt(mac) \  
-ifqType $opt(ifq) \  
-ifqLen $opt(ifqlen) \  
-antType $opt(ant) \  
-propType $opt(prop) \  
-phyType $opt(netif2) \  
-channelType $opt(chan) \  
-topoInstance $topo \  
-agentTrace ON \  
-routerTrace ON \  
-macTrace OFF \  
-energyModel $opt(energymodel) \  
-idlePower 0.05 \  
-sensePower 0.3 \  
-rxPower 0.5 \  
-txPower 1.0 \  
-sleepPower 0.002 \  
-initialEnergy $opt(initialenergy)
```

```
for {set i 0} {$i < $opt(nn)} {incr i} {  
    set node_($i) [$ns_ node]  
    $node_($i) random-motion 0 ;# disable random motion  
}
```

```
set null_(0) [new Agent/Null]  
$ns_ attach-agent $node_(0) $null_(0)
```

```
set opt(abc) [expr 8*262144*0.2]
```

```
for {set i 1} {$i < $dugumsayisi} {incr i} {  
    set udp_($i) [new Agent/UDP]  
    $udp_($i) set packetSize_ 50000  
    $ns_ attach-agent $node_($i) $udp_($i)
```

```
set cbr_($i) [new Application/Traffic/CBR]  
$cbr_($i) set packetSize_  
$cbr_($i) set rate_ $opt(abc)  
$cbr_($i) set maxpkts_ 5000000  
$cbr_($i) attach-agent $udp_($i)  
$ns_ connect $udp_($i) $null_(0)  
}
```

```
set kenardakidugumsayisi [expr { int(sqrt([expr $dugumsayisi-1])) }]  
$node_(0) label "BAZ ISTASYONU" ;# sets label on node  
$node_(0) set X_ 100;
```

```

$node_(0) set Y_ 100;
$node_(0) set Z_ 0;
$ns_ initial_node_pos $node_(0) 20
for {set i 0} {$i < $kenardakidugumsayisi } {incr i} {
    for {set j 1} {$j <= $kenardakidugumsayisi } {incr j} {
        set nodeId [expr $i*$kenardakidugumsayisi + $j];
        $node_($nodeId) set X_ [expr $i*150+100];
        $node_($nodeId) set Y_ [expr $j*150+100];
        $node_($nodeId) set Z_ 0.0
        $ns_ initial_node_pos $node_($nodeId) 20
        #puts [expr $j*100]
    }
}
for {set i 1} {$i < $dugumsayisi } {incr i} {
    $ns_ at 1.00 "$cbr_($i) start"
    $ns_ at 121.00 "$cbr_($i) stop"
}
for {set i 0} {$i < $opt(nn) } {incr i} {
    $ns_ at $opt(stop) "$node_($i) reset";
}
$ns_ at $opt(stop) "puts \"NS EXITING...\" ; $ns_ halt"
puts "Starting Simulation..."
$ns_ run

```

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı : Arafat ŞENTÜRK
Doğum Tarihi ve Yeri : 20.08.1985 - ANKARA
Yabancı Dili : İngilizce
E-posta : arafatsenturk@gmail.com - arafatsenturk@duzce.edu.tr

ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Doktora	Elektrik Elektronik ve Bilgisayar Müh.	Düzce Üniversitesi	2017
Y. Lisans	Elektrik Elektronik Müh.	Düzce Üniversitesi	2012
Lisans	Bilgisayar Müh.	Atılım Üniversitesi	2009
Lise	Sayısal	Yavuz Sultan Lisesi	2003

TEZ KAPSAMINDA YAPILAN YAYINLAR

A. ÜNİVERSİTELER TARAFINDAN DESTEKLENEN PROJELER

[A.1] Düzce Üniversitesi Bilimsel Araştırma Projeleri (DÜBAP), “Kablosuz Multimedya Algılayıcı Ağlar için Enerji Verimliliği Açısında Adaptif Görüntü Sıkıştırma Algoritması Geliştirilmesi”, 2015.06.01.308.

B. SCI, SCI-E İNDEKSİNE KAYITLI DERGİLERDE YAYINLANAN MAKALELER

[B.1] A. Senturk and R. Kara, “An Analysis of Image Compression Techniques in Wireless Multimedya Sensor Networks”, Technical Gazette, 6, 2016, 1863-1869.

C. DİĞER İNDEKSLERE KAYITLI, HAKEMLİ ULUSAL DERGİLERDE YAYINLANAN MAKALELER

- [C.1] A. Senturk and Z. K. Senturk, “A Study on Energy Efficiency in Wireless Multimedia Sensor Networks”, Düzce University Journal of Science and Technology, 3, 2015, 145-151.

D. İNDEKSLERE KAYITLI, HAKEMLİ ULUSLAR ARASI KONFERANS KİTAPLARINDA YAYINLANAN BİLDİRİLER

- [D.1] A. Senturk and R. Kara, “Kablosuz Multimedya Algılayıcı Ağlarda Kullanılan Görüntü Sıkıştırma Algoritmalarının Düğüm Sayısına Göre Karşılaştırılması” 1st International Conference on Engineering Technology and Applied Sciences Afyon Kocatepe University, Turkey 21-22 April 2016.
- [D.2] A. Senturk and R. Kara, “Comparison of image compression algorithms used in Wireless Multimedia Sensor Networks | Kablosuz Multimedya Algılayıcı Ağlarda Kullanılan Görüntü Sıkıştırma Algoritmalarının Karşılaştırılması”, in 2016 24th Signal Processing and Communication Application Conference, SIU 2016 - Proceedings, 2016.
- [D.3] A. Senturk, Z. K. Senturk and R. Kara, “Image Compression Techniques in Wireless Multimedia Sensor Networks | Kablosuz Multimedya Algılayıcı Ağlarda Görüntü Sıkıştırma Teknikleri”, in 2015 23th Signal Processing and Communication Application Conference, SIU 2015 - Proceedings, 2015.
- [D.3] A. Senturk, Z. K. Senturk and R. Kara, “Comparison of Real Time Image Transfer in Wireless Multimedia Sensor Networks”, International Conference on Electrical and Electronics Engineering, ELECO 2015.