

KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

IETF 6TİSCH AĞLARINDA GÜVENLİ DAĞITIK KİMLİK DOĞRULAMA
MEKANİZMALARININ GELİŞTİRİLMESİ

DOKTORA TEZİ

Hakan AYDIN

EKİM 2024
TRABZON



**KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

**IETF 6TİSCH AĞLARINDA GÜVENLİ DAĞITIK KİMLİK DOĞRULAMA
MEKANİZMALARININ GELİŞTİRİLMESİ**

Hakan AYDIN

**Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsünde
"DOKTOR (BİLGİSAYAR MÜHENDİSLİĞİ)"
Unvanı Verilmesi İçin Kabul Edilen Tezdir.**

**Tezin Enstitüye Verildiği Tarih : 02/09/2024
Tezin Savunma Tarihi : 20/09/2024**

Tez Danışmanı : Doç. Dr. Sedat GÖRMÜŞ

Trabzon 2024

ÖNSÖZ

“IETF 6TiSCH Ağlarında Güvenli Dağıtık Kimlik Doğrulama Mekanizmalarının Geliştirilmesi” isimli bu tez, Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı, Doktora Programı’nda hazırlanmıştır.

Tez danışmanım Sayın Doç. Dr. Sedat Görmüş’e, bana yol gösterici ve destekleyici yaklaşımı için minnettarım. Sedat hocam, tez çalışmamın her aşamasında sağladığı bilgi, tecrübe ve vizyon ile bu çalışmanın yönünü belirlemiş ve zenginleştirmiştir. Onun sabrı, sürekli teşviki ve kıymetli geri bildirimleri olmasaydı, bu çalışmanın bu noktaya gelmesi mümkün olamazdı. Bu vesileyle hocama, bana sağladığı tüm bu katkılar ve gösterdiği ilgi için en derin şükranlarımı sunarım. Ayrıca, değerli tez izleme komitesi üyelerim Prof. Dr. Güzin Ulutaş’a ve Prof. Dr. Gökçe Hacıoğlu’na da teşekkürlerimi sunarım. Değerli araştırma görevlisi arkadaşlarım Sefa Aras, Musa Aslan, Mustafa Topsakal, Özge Nur Özaras ve İbrahim Uğur Yılmaz’a destekleri için teşekkür ederim. Manevi desteklerinden dolayı Prof. Dr. Özcan Özyurt’a da teşekkür etmek istiyorum. Zorlu süreçlerde her zaman yanımda olmuş, bilgi ve tecrübesiyle beni yönlendirmiştir. Desteklerinden dolayı araştırma görevlileri Burak Aydın’a, Mehmet Ali Küçük’e, Mervenur Çakır’a ve Mavi Alp Bilgi Teknolojileri çalışanlarına da özel olarak teşekkür etmek istiyorum. Son olarak, ikizim Gökhan Aydın’a, bana her zaman sağladığı sarsılmaz destek ve yüksek moral için minnettarım. Onun varlığı, dostluğu ve cesaretlendirmesi, bu zorlu süreçte bana büyük bir güç kaynağı olmuştur. Ailemin koşulsuz sevgisi, sonsuz sabrı ve derin anlayışı olmasaydı, bu çalışmayı tamamlamak çok daha zor ve hatta imkansız olurdu. Onların desteğiyle bu noktaya gelebilmek, benim için tarif edilemez bir mutluluk ve gurur kaynağıdır. Ayrıca, tezimi büyük bir minnettarlıkla Türk toplumuna atfetmek istiyorum. Atalarımızın ‘Yufka yürekliyle çetin yollar aşılmaz; Çünkü bu yol kutludur, gider Tanrı Dağı’na.’ sözü, akademik yolculuğumdaki kararlılık ve cesaretime ilham olmuştur. Bu tezin, ülkemizin bilimsel birikimine ve toplumsal gelişimine katkı sağlaması en büyük dileğimdir.

Hakan AYDIN

Trabzon 2024

TEZ ETİK BEYANNAMESİ

Doktora Tezi olarak sunduğum "IETF 6TiSCH Ağlarında Güvenli Dağıtık Kimlik Doğrulama Mekanizmalarının Geliştirilmesi" başlıklı bu çalışmayı baştan sona kadar danışmanım Doç. Dr. Sedat GÖRMÜŞ'nin sorumluluğunda tamamladığımı, verileri/örnekleri kendim topladığımı, deneyleri/analizleri ilgili laboratuvarlarda yaptığımı/yaptırdığımı, başka kaynaklardan aldığım bilgileri metinde ve kaynakçada eksiksiz olarak gösterdiğimi, çalışma sürecinde bilimsel araştırma ve etik kurallara uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul ettiğimi beyan ederim. 20/09/2024

Hakan AYDIN

İÇİNDEKİLER

	<u>Sayfa No</u>
ÖNSÖZ.....	III
TEZ ETİK BEYANNAMESİ.....	IV
İÇİNDEKİLER.....	V
ÖZET.....	VIII
SUMMARY	IX
ŞEKİLLER DİZİNİ	X
TABLolar DİZİNİ.....	XIII
SEMBOLLER DİZİNİ	XV
1. GENEL BİLGİLER	1
1.1. Motivasyon	1
1.2. Giriş.....	3
2. NESNELERİN İNTERNETİ İÇİN PROTOKOL YIĞINI	8
2.1. IEEE 802.15.4 Standardı	11
2.1.1. Beacon Çerçevesi.....	15
2.1.2. Veri Çerçevesi	16
2.1.3. Onay (Acknowledgment) Çerçevesi	17
2.2. İnternet Protokolü Versiyon 6	21
2.3. 6LoWPAN	25
2.3.1. Adres Modları.....	27
2.3.2. Parçalama	29
2.3.3. Başlık Sıkıştırma.....	30
2.3.4. Yönlendirme	32
2.3.5. Otomatik Yapılandırma ve Komşu Keşfi.....	42
2.4. 6TiSCH.....	44
2.4.1. 6TiSCH Mimarisinde Çizelgeleme ve Yönlendirme	51
2.5. CoAP	57
2.5.1. OSCORE	61
2.6. Contiki ve Cooja	62

2.6.1. Bellek Tahsisi ve Yönetimi	65
2.6.2. Güç Yönetimi Teknikleri.....	67
3. SENSÖR AĞLARI İÇİN GÜVENLİK GEREKSİNİMLERİ	71
3.1. Kimlik Doğrulama (Authentication)	71
3.2. Veri Bütünlüğü (Data Integrity)	71
3.3. Veri Güncelliği (Data Freshness)	72
3.4. Hizmet Bütünlüğü (Service Integrity)	73
3.5. Senkronizasyon (Synchronization)	73
3.6. Erişilebilirlik (Availability).....	74
4. LİTERATÜR	75
4.1. Kablosuz Sensör Ağlarında Merkezi ve Dağıtık Kimlik Doğrulama Üzerine Yapılan Çalışmalar	76
4.2. Kablosuz Sensör Ağlarında Uygulanan Gelişmiş Kimlik Doğrulama Teknikleri Üzerine Yapılan Çalışmalar.....	85
5. YAPILAN ÇALIŞMALAR	94
5.1. Ölçeklenebilir IETF 6TiSCH Ağları için Merkezi Olmayan Proxy-JRC Kimlik Doğrulama Sistemi	94
5.2. Dağıtık Proxy-JRC Kimlik Doğrulama Noktalarının Belirlenmesi	99
5.3. IETF 6TiSCH Ağları için Hafif Üç Faktörlü Sensör ve Kullanıcı Kimlik Doğrulama Mekanizması.....	113
5.3.1. Sistem Kurulum Aşaması.....	115
5.3.2. Sistem Kayıt Aşaması	117
5.3.3. Sistem Giriş Aşaması	120
5.3.4. Kullanıcı ve Cihaz Kimlik Doğrulama Aşaması	121
5.3.5. Sistem Anahtarı Güncelleme Aşaması.....	125
6. BULGULAR VE İRDELEME	130
6.1. Önerilen Proxy-JRC Seçim Yöntemlerinin Deneysel Sonuçları.....	130
6.2. Önerilen Hafif Üç Faktörlü Sensör ve Kullanıcı Kimlik Doğrulama Mekanizmasının Deneysel Sonuçları	144
6.2.1. Güvenlik Analizi.....	144
6.2.2. Performans Değerlendirmesi ve İşlevsellik Analizi.....	155

7. SONUÇLAR	161
8. ÖNERİLER VE TARTIŞMA	163
9. KAYNAKLAR.....	165
ÖZGEÇMİŞ	



Doktora Tezi

ÖZET

IETF 6TiSCH AĞLARINDA GÜVENLİ DAĞITIK KİMLİK DOĞRULAMA MEKANİZMALARININ GELİŞTİRİLMESİ

Hakan AYDIN

Karadeniz Teknik Üniversitesi

Fen Bilimleri Enstitüsü

Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Sedat GÖRMÜŞ

2024, 174 Sayfa

Bu tezde, IETF 6TiSCH ağlarında güvenli dağıtık kimlik doğrulama mekanizmalarının geliştirilmesi üzerine yapılan araştırma ve geliştirme çalışmaları sunulmaktadır. Endüstriyel Nesnelerin İnterneti ortamlarında kullanılan RPL protokolü, düşük güçlü ve kayıplı ağlar için yaygın bir çözüm olmasına rağmen güvenlik açısından bazı zafiyetler içermektedir. Özellikle, ağ topolojisi ve düğüm kaynaklarına yönelik saldırılar, ağın bütünlüğünü ve güvenilirliğini tehdit etmektedir. Bu çalışmanın amacı, RPL tabanlı 6TiSCH ağlarında güvenliği artırmak için dağıtık kimlik doğrulama mekanizmaları geliştirmektir. Bu doğrultuda, JRC benzeri merkezi otoritelerin ağda stratejik olarak konumlandırılmasına odaklanılmıştır. Bu merkezi doğrulama noktaları, belirli bölgelerde düğüm kimlik doğrulama işlemlerini gerçekleştirerek, merkezi otoritenin yükünü dağıtmakta ve genel güvenliği artırmaktadır. Amaç, merkezi doğrulama adımlarını stratejik noktalarda gerçekleştirerek düğüm doğrulama işlemlerini dağıtmaktır. Böylece her düğüm, en yakın doğrulama noktası aracılığıyla kimlik doğrulama yapabilir ve merkezi otoriteye bağımlılık azalır. Bu yaklaşım, merkezi yöntemlerdeki ölçeklenebilirlik sorununu çözmeyi hedeflemektedir. Ağın genişlemesi durumunda yeni doğrulama noktaları eklenerek sistemin ölçeklenebilirliği sağlanabilir. Ayrıca, sensör, kullanıcı ve sunucu arasında hızlı ve güvenilir bir kimlik doğrulama yöntemi önerilmiştir. Sonuç olarak, bu çalışma RPL tabanlı 6TiSCH ağlarında güvenliği sağlamak için yeni ve etkili bir çözüm sunmakta olup, gelecekteki IIoT uygulamalarında kullanılabilirliği artırmayı hedeflemektedir. Önerilen yöntem, ağ güvenliğini sağlarken performans üzerindeki olumsuz etkileri en aza indirmeyi amaçlamaktadır.

Anahtar Kelimeler: 6TiSCH, Endüstriyel Nesnelerin İnterneti, Anahtar Dağıtımı, Anahtar Yönetimi, Güvenli Katılım.

PhD. Thesis

SUMMARY

DEVELOPMENT OF SECURE DISTRIBUTED AUTHENTICATION MECHANISMS IN IETF 6TiSCH NETWORKS

Hakan AYDIN

Karadeniz Technical University

The Graduate School of Natural and Applied Sciences

Department of Computer Engineering Graduate Program

Supervisor: Assoc. Prof. Sedat GÖRMÜŞ

2024, 174 Pages

This thesis presents research and development studies on developing secure distributed authentication mechanisms in IETF 6TiSCH networks. Although the RPL protocol is widely used in Industrial Internet of Things (IIoT) environments for low-power and lossy networks, it contains some security vulnerabilities. Specifically, attacks targeting the network topology and node resources threaten the integrity and reliability of the network. The main objective of this study is to enhance security in RPL-based 6TiSCH networks by developing distributed authentication mechanisms. To achieve this, the focus is on strategically positioning JRC-like central authorities within the network. These central authentication points perform node authentication in specific regions, distributing the load of the central authority and increasing overall network security. The aim is to carry out central authentication steps at strategic points, thereby enabling distributed node authentication. Consequently, each node can perform authentication through the nearest central point, reducing dependency on a central authority. This approach aims to solve the scalability issues inherent in centralized methods. In case of network expansion, new authentication points can be added to ensure scalability. Additionally, a simple and efficient authentication method is proposed between sensors, users, and servers. This method aims to optimize network performance and ensure security by facilitating rapid and reliable authentication between sensors and users. In conclusion, this study offers a novel and effective solution for ensuring security in RPL-based 6TiSCH networks, aiming to enhance its applicability in future IIoT applications. The proposed method aims to ensure network security while minimizing any adverse effects on performance.

Key Words: 6TiSCH, Industrial Internet of Things, Key Distribution, Key Management, Secure Join.

ŞEKİLLER DİZİNİ

	<u>Sayfa No</u>
Şekil 1. Nesnelerin İnterneti kullanım alanları	4
Şekil 2. Nesnelerin İnterneti teknolojisinin yıllara göre kullanım sayıları	5
Şekil 3. Nesnelerin İnterneti teknolojisinin katmanlara göre saldırı türleri	6
Şekil 4. Nesnelerin İnterneti teknolojisinde yer alan bileşenlerin örnek bir gösterimi	8
Şekil 5. 6LoWPAN ile KSA'yı İnternete bağlayan tipik bir ağ gösterimi	9
Şekil 6. Standart TCP/IP modeli ve 6LoWPAN tabanlı protokol yığını	9
Şekil 7. Nesnelerin İnterneti cihazı ile geleneksel İnternet ana makinesi için protokol yığını ve bu cihazların 6LoWPAN sınır yönlendirici kullanılarak entegrasyonu	12
Şekil 8. Star ve Peer to Peer topoloji örnekleri	13
Şekil 9. IEEE 802.15.4 standardının desteklediği frekans bantları	14
Şekil 10. Veri paketlerinin radyo katmanından çıkış süreci	14
Şekil 11. Beacon çerçevesi	15
Şekil 12. Veri çerçevesi	17
Şekil 13. ACK çerçevesi	18
Şekil 14. MAC komut çerçevesi	18
Şekil 15. ACL giriş formatı	20
Şekil 16. IPv4 ve IPv6 başlıklarını karşılaştırma	21
Şekil 17. IPv6 adresleme türleri: (I) Unicast, (II) Broadcast, (III) Multicast, (IV) Anycast	23
Şekil 18. 6LoWPAN'ın çeşitli standartlar ile ilişkisi	25
Şekil 19. Farklı 6LoWPAN mimarileri	27
Şekil 20. 6LoWPAN Mesh adres başlığı	28
Şekil 21. IPv6 paketinin 6LoWPAN ağında iletilmek üzere parçalandığında gönderilen ilk parçası	29
Şekil 22. IPv6 paketinin geri kalanını içeren parçalar	30
Şekil 23. 6LoWPAN HC1 yapısı	30

Şekil 24.	6LoWPAN HC2 yapısı	31
Şekil 25.	Mesh-under paket yönlendirme	32
Şekil 26.	Route-over paket yönlendirme	33
Şekil 27.	RPL kontrol mesajı	35
Şekil 28.	RPL ile DODAG oluşum aşamaları	36
Şekil 29.	RPL protokolünün kontrol mesajlarının akış diyagramı	38
Şekil 30.	RPL DIO paketi	38
Şekil 31.	RPL Depolama (Storing) modu	40
Şekil 32.	RPL Depolama olmayan (Non-Storing) modu	41
Şekil 33.	6TiSCH protokol yığınının oluşturan farklı standartlar	45
Şekil 34.	Çizelgeleme, ASN ve örnek bir 6TiSCH ağı.....	46
Şekil 35.	6TiSCH çizelgeleme örneği	46
Şekil 36.	IEEE 802.15.4e TSCH protokolünde zaman senkronizasyonunu sağlayan iki farklı yaklaşım.....	49
Şekil 37.	IEEE 802.15.4e TSCH enhanced beacon paket formatı	50
Şekil 38.	Çizelgeleme stratejilerinin sınıflandırılması (Urke vd., 2022)	52
Şekil 39.	A ve B düğümleri arasında 6P ekleme işlemi örneği.....	53
Şekil 40.	6TiSCH merkezi kontrol yapıları	55
Şekil 41.	CoRE ReSTful mimarisi	58
Şekil 42.	CoAP protokol yığını, mesaj ve başlık formatı	58
Şekil 43.	CoAP mesaj aktarımı	60
Şekil 44.	CoAP mesajlarının OSCORE ile şifrelenmesi	62
Şekil 45.	Contiki OS mimarisi	64
Şekil 46.	msp430-size aracı ile bellek ölçümü	67
Şekil 47.	50 Düğümlü bir ağ için PowerTrace modülü	69
Şekil 48.	Normal ve saldırı durumlarında düğüm yapıları.	72
Şekil 49.	KSA'da kimlik doğrulama şemalarının sınıflandırılması (Sheikh vd., 2019)	88
Şekil 50.	Ağ üzerinde yer alan sensör cihazları ile son kullanıcı arasındaki haberleşme yapısı	89
Şekil 51.	CoJP ile ağa dahil olma süreci	95

Şekil 52.	Proxy-JRC tabanlı kimlik doğrulama altyapısı	97
Şekil 53.	Wireshark üzerinde ağa katılım paketlerinin incelenmesi	98
Şekil 54.	Aday Proxy-JRC'lerin belirlenmesi.....	100
Şekil 55.	32 düğümlü bir ağda komşuluk ilişkisi	102
Şekil 56.	32 düğümlü bir ağda yönlendirme ilişkisi.....	103
Şekil 57.	21 düğümlü RPL tabanlı Cooja simülasyonu.....	106
Şekil 58.	Düğüm 4 ve 14 için Oracle Proxy-JRC	112
Şekil 59.	Maliyet Fonksiyonu denklem 5.1'e göre seçilen Proxy-JRC düğümleri	113
Şekil 60.	Maliyet Fonksiyonu denklem 5.2'ye göre seçilen Proxy-JRC düğümleri	114
Şekil 61.	Ağa katılma sürecindeki aktörler.....	115
Şekil 62.	6TiSCH ağlarında kullanıcı-sensör-kimlik doğrulama sunucu kontrol sistemine bir örnek.....	117
Şekil 63.	Kullanıcı kayıt şeması.....	118
Şekil 64.	Cihaz (sensör düğümü) kayıt şeması	119
Şekil 65.	Kullanıcı giriş şeması.....	120
Şekil 66.	Kullanıcı kimlik doğrulama şeması	121
Şekil 67.	Cihaz anahtar güncelleme şeması.....	126
Şekil 68.	Kullanıcı anahtar güncelleme şeması.....	127
Şekil 69.	CPU, LPM, TX, RX ve toplam enerji tüketiminin zamana bağlı analizi.....	134
Şekil 70.	20 düğümlü bir ağ için ortalama anahtar güncelleme gecikmesi	135
Şekil 71.	35 düğümlü bir ağ için ortalama anahtar güncelleme gecikmesi	136
Şekil 72.	50 düğümlü bir ağ için Ortalama anahtar güncelleme gecikmesi.....	137
Şekil 73.	20 düğümlü bir ağda anahtar güncellemesi için ortalama güç tüketimi.....	137
Şekil 74.	35 düğümlü bir ağda anahtar güncellemesi için ortalama güç tüketimi.....	138
Şekil 75.	50 düğümlü bir ağda anahtar güncellemesi için ortalama güç tüketimi.....	138
Şekil 76.	20 düğümlü ağda 2 Proxy-JRC için ortalama anahtar güncelleme gecikmesi	138
Şekil 77.	20 düğümlü ağda ortalama cihaz anahtarı güncelleme gecikmesi.....	139
Şekil 78.	20 düğümlü ağda anahtar güncelleme işlemi sırasında tüketilen ortalama güç değerleri (2 Proxy-JRC)	140

TABLolar DİZİNİ

Sayfa No

Tablo 1.	IEEE 802.15.4 radyo özellikleri	13
Tablo 2.	IEEE 802.15.4 tarafından desteklenen güvenlik mekanizmaları (Mesaj Kimlik Doğrulama Kodu - Message Authentication Code)	20
Tablo 3.	IPv6 adres türleri	23
Tablo 4.	IPv4 ve IPv6'nın karşılaştırılması	24
Tablo 5.	CoAP istek yöntemleri	59
Tablo 6.	CoAP yanıt kodları	59
Tablo 7.	Literatürde yer alan merkezi ve dağıtık kimlik doğrulama konulu çalışmalar	86
Tablo 8.	Literatürde yer alan merkezi ve dağıtık kimlik doğrulama konulu çalışmalar-2	87
Tablo 9.	Simülasyon Parametreleri	101
Tablo 10.	20 Düğümlü bir ağ için ağırlık katsayılarının seçimi.....	108
Tablo 11.	35 Düğümlü bir ağ için ağırlık katsayılarının seçimi.....	109
Tablo 12.	50 Düğümlü bir ağ için ağırlık katsayılarının seçimi.....	110
Tablo 13.	35 ve 50 Düğümlü ağlar için ağırlık katsayılarının seçimi	111
Tablo 14.	Önerilen protokolde kullanılan tanımlar listesi	116
Tablo 15.	OSCORE ve IPsec Protokolleri Karşılaştırması.....	129
Tablo 16.	Exp5438 cihazı için akım tüketim değerleri.....	131
Tablo 17.	Cooja simülatörü ve EXP5438 donanım cihazının karşılaştırmalı özellikleri	131
Tablo 18.	Görev döngüsü (Duty Cycle) değerleri	135
Tablo 19.	Proxy-JRC sayısına göre 20 düğüm için ortalama anahtar güncelleme süresi (saniye).....	141
Tablo 20.	Anahtar güncelleme sırasında ortalama güç tüketimi (mW) (20 düğüm).....	142
Tablo 21.	Anahtar güncelleme sırasında ortalama güç tüketimi (mW) (35 düğüm).....	143

Tablo 22.	Önerilen mekanizmanın güvenlik özelliklerinin ilgili çalışmalarla karşılaştırılması: (A: Kullanıcı Anonimliği, B: Karşılıklı Kimlik Doğrulama, C: Ortadaki Adam Saldırısı, D: İçeriden Saldırı, E: Kötü Niyetli Kullanıcı Saldırısı, F: Yeniden Oynatma Saldırısı, G: Çevrimdışı Tahmin Saldırısı, H: Gizli Sunucu Saldırısı, I: Sensör Taklit Saldırısı, K: Mesaj Gizliliği).....	156
Tablo 23.	Performans karşılaştırması: (T_H , T_{ECC} , T_{ECM} , T_S , T_X , T_{FE} , T_{PUF})	158
Tablo 24.	Farklı yöntemlerin toplam bit ve enerji tüketimi bazında karşılaştırılması...	159



SEMBOLLER DİZİNİ

<i>IoT</i>	:	Nesnelerin İnterneti (Internet of Things)
<i>IIoT</i>	:	Endüstriyel Nesnelerin İnterneti (Industrial Internet of Things)
<i>IEEE</i>	:	Elektrik ve Elektronik Mühendisleri Enstitüsü (Institute of Electrical & Electronic Engineers)
<i>IETF</i>	:	İnternet Mühendisliği Görev Gücü (Internet Engineering Task Force)
<i>WSN</i>	:	Kablosuz Sensör Ağı (Wireless Sensor Network)
<i>DWSN</i>	:	Dağıtık Kablosuz Sensör Ağı (Distributed Wireless Sensor Networks)
<i>HWSN</i>	:	Hiyerarşik Kablosuz Sensör Ağı (Hierarchical Wireless Sensor Networks)
<i>WSAN</i>	:	Kablosuz Sensör ve Aktüatör Ağı (Wireless Sensor And Actuator Network)
<i>SoC</i>	:	Çip Üzerinde Sistem (System on Chip)
<i>CPU</i>	:	Merkezi İşlem Birimi (Control Processing Unit)
<i>RAM</i>	:	Rastgele Erişimli Bellek (Random Access Memory)
<i>ROM</i>	:	Salt Okunur Bellek (Read Only Memory)
<i>MAC</i>	:	Ortam Erişim Kontrolü (Media Access Control)
<i>IP</i>	:	İnternet Protokolü (Internet Protocol)
<i>IPv6</i>	:	İnternet Protokolü sürüm 6 (Internet Protocol version 6)
<i>MTU</i>	:	Maksimum İletim Birimi (Maximum Transmission Unit)
<i>6BR</i>	:	6LoWPAN Sınır Yönlendirici (6LoWPAN Border Router)
<i>6LoWPAN</i>	:	Düşük Güçlü Kişisel Alan Ağlarında IPv6 (IPv6 over Low-Power Wireless Personal Area Networks)
<i>AES</i>	:	Gelişmiş Şifreleme Standardı (Advanced Encryption Standard)
<i>ECC</i>	:	Eliptik Eğri Kriptografisi (Elliptic Curve Cryptography)
<i>CoAP</i>	:	Kısıtlı Uygulama Protokolü (Constrained Application Protocol)
<i>TLS</i>	:	Taşıma Katmanı Güvenliği (Transport Layer Security)
<i>DTLS</i>	:	Datagram Taşıma Katmanı Güvenliği (Datagram Transport Layer Security)
<i>IPHC</i>	:	IP Başlık Sıkıştırması (IP Header Compression)

<i>HTTP</i>	: Hiper Metin Transfer Protokolü (Hyper Text Transfer Protocol)
<i>LwM2M</i>	: Hafif Makineler Arası İletişim (Lightweight Machine-to-Machine)
<i>PKC</i>	: Açık Anahtar Kriptografisi (Public Key Cryptography)
<i>PSK</i>	: Önceden Paylaşılan Anahtar (Pre-Shared Key)
<i>SDK</i>	: Yazılım Geliştirme Kiti (Software Development Kit)
<i>CSMA</i>	: Taşıyıcı Algılama Çoklu Erişim (Carrier Sense Multiple Access)
<i>TSCH</i>	: Zaman Dilimli Kanal Atlama (Time Slotted Channel Hopping)
<i>6TiSCH</i>	: IEEE 802.15.4 TSCH modunda IPv6 (IPv6 over the TSCH mode of IEEE 802.15.4)
<i>ROLL</i>	: Düşük Güçlü ve Kayıplı Ağlar Üzerinde Yönlendirme (Routing over Low-Power and Lossy Networks)
<i>RPL</i>	: Düşük Güçlü ve Kayıplı Ağlar için Yönlendirme Protokolü (Routing Protocol for Low-power and Lossy Networks)
<i>DAG</i>	: Yönlendirilmiş Asiklik Grafik (Directed Acyclic Graph)
<i>DODAG</i>	: Hedef Yönelimli Yönlendirilmiş Asiklik Grafik (Destination-Oriented Directed Acyclic Graph)
<i>OF</i>	: Amaç Fonksiyonu (Objective Function)
<i>MRHOF</i>	: Histerezisli Minimum Sıralama Amaç Fonksiyonu (Minimum Rank with Hysteresis Objective Function)
<i>OF0</i>	: Sıfır Amaç Fonksiyonu (Objective Function Zero)
<i>CA</i>	: Sertifika Yetkilisi (Certificate Authority)
<i>CBC</i>	: Şifre Blok Zinciri (Cipher Block Chaining)
<i>CCM</i>	: CBC-MAC ile Sayıcı (Counter with CBC-MAC)
<i>Mac</i>	: Mesaj Doğrulama Kodu (Message Authentication Code)
<i>SHA</i>	: Güvenli Hash Algoritması (Secure Hash Algorithm)
<i>ECDHE</i>	: Eliptik Eğri Diffie-Hellman (Elliptic Curve Diffie-Hellman)
<i>ECDSA</i>	: Eliptik Eğri Dijital İmza Algoritması (Elliptic Curve Digital Signature Algorithm)
<i>RSA</i>	: Rivest-Shamir-Adleman
<i>UDP</i>	: Kullanıcı Datagram Protokolü (User Datagram Protocol)
<i>NH</i>	: Sonraki Başlık (Next Header)
<i>NHC</i>	: Sonraki Başlık Sıkıştırması (Next Header Compression)
<i>P</i>	: Pledge

<i>CoJP</i>	: Kısıtlı Katılma Protokolü (Constrained Join Protocol)
<i>JP</i>	: Katılma Proxy (Join Proxy)
<i>JN</i>	: Katılmış Düğüm (Joined Node)
<i>JRC</i>	: Katılma Kaydedici/Koordinatörü (Join Registrar/Coordinator)
<i>P – JRC</i>	: Proxy Katılma Kaydedici/Koordinatörü (Proxy Join Registrar/Coordinator)
<i>DIO</i>	: DODAG Bilgi Nesnesi (DODAG Information Object)
<i>DAO</i>	: Hedef Reklam Nesnesi (Destination Advertisement Object)
<i>DIS</i>	: DODAG Bilgi İsteği (DODAG Information Solicitation)
<i>DAO – ACK</i>	: Hedef Reklam Nesnesi Onayı (Destination Advertisement Object Acknowledgment)
<i>EB</i>	: Gelişmiş İşaretçi (Enhanced Beacon)
<i>IE</i>	: Bilgi Elemanı (Information Element)
<i>ASN</i>	: Mutlak Zaman Numarası (Absolute Slot Number)
<i>Tx</i>	: Gönderim (Transmit)
<i>Rx</i>	: Alım (Reception)
<i>PAN</i>	: Kişisel Alan Ağı (Personal Area Network)
<i>RFD</i>	: Azaltılmış Fonksiyon Cihazı (Reduced Function Device)
<i>CSMA/CA</i>	: Taşıyıcı Algılama Çoklu Erişim/Çakışma Önleme (Carrier Sense Multiple Access/Collision Avoidance)
<i>CCA</i>	: Boş Kanal Değerlendirmesi (Clear Channel Assessment)
<i>FFD</i>	: Tam Fonksiyon Cihazı (Full Function Device)
<i>PCE</i>	: Yol Hesaplama Elemanı (Path Computation Element)
<i>QoS</i>	: Hizmet Kalitesi (Quality of Service)
<i>L2S</i>	: Katman 2 Güvenliği (Layer 2 Security)
<i>L3S</i>	: Katman 3 Güvenliği (Layer 3 Security)
<i>IPsec</i>	: İnternet Protokolü Güvenliği (Internet Protocol Security)
<i>IKEv2</i>	: İnternet Anahtar Değişim Sürüm 2 (Internet Key Exchange version 2)
<i>IDS</i>	: Saldırı Tespit Sistemi (Intrusion Detection System)
<i>PDR</i>	: Paket Teslim Oranı (Packet Delivery Ratio)
<i>AE2ED</i>	: Ortalama Uçtan Uca Gecikme (Average End-to-end Delay)
<i>MP2P</i>	: Çok Noktadan Noktaya (Multi-Point-to-Point)
<i>P2MP</i>	: Noktadan Çok Noktaya (Point-to-Multipoint)

<i>P2P</i>	: Noktadan Noktaya (Point-to-Point)
<i>ICMPv6</i>	: IPv6 için İnternet Kontrol Mesaj Protokolü (Internet Control Message Protocol for IPv6)
<i>IQR</i>	: Çeyrekler Arası Aralık (Interquartile Range)
<i>ETX</i>	: Beklenen İletim Sayısı (Expected Transmission Count)
<i>SRH</i>	: Kaynak Yolu Başlığı (Source Route Header)
<i>LPM</i>	: Düşük Güç Modu (Low Power Mode)
<i>ACE</i>	: Kimlik Doğrulama Koordinasyon Elemanı (Authentication Coordination Element)
<i>MLME</i>	: MAC Katmanı Yönetim Birimi (MAC Layer Management Entity)
<i>6top</i>	: 6TiSCH Operasyon Alt Katmanı (6TiSCH Operation Sublayer)
<i>CoRE</i>	: Kısıtlı RESTful Ortamlar (Constrained RESTful Environments)
<i>REST</i>	: Temsili Durum Transferi (Representational State Transfer)
<i>UA</i>	: Kullanıcı Kimlik Doğrulama (User Authentication)
<i>ID</i>	: Kimlik (Identity)

1. GENEL BİLGİLER

1.1. Motivasyon

Nesnelerin İnterneti (Internet of Things - IoT), günümüzde teknolojinin en hızlı gelişen ve yaygınlaşan alanlarından biri olarak öne çıkmaktadır (Madakam vd., 2015; Görmüş vd., 2018). IoT, günlük yaşamımızın birçok alanına nüfuz etmiş, ev otomasyonundan akıllı şehir uygulamalarına, sağlık hizmetlerinden endüstriyel otomasyona kadar geniş bir yelpazede kullanılmaktadır. IoT'nin bu denli yaygınlaşması, cihazların birbirleriyle ve merkezi sistemlerle kesintisiz ve güvenli bir şekilde iletişim kurmasını gerektirmektedir. Bu noktada, güvenlik ve verimlilik önemli birer sorun olarak karşımıza çıkmaktadır. IoT'nin özellikle Endüstriyel Nesnelerin İnterneti (Industrial Internet of Things - IIoT) olarak bilinen endüstriyel uygulamalarda kullanımı, bu teknolojinin güvenlik ve verimlilik gereksinimlerini daha da kritik hale getirmektedir (Sengupta vd., 2020; Boyes vd., 2018). IIoT, fabrikalar, enerji ağları ve lojistik gibi sektörlerde cihazların birbirleriyle ve merkezi kontrol sistemleriyle iletişim kurarak operasyonları optimize etmesini sağlar (Vilajosana vd., 2017; Alizadeh & Aydın, 2019; Dehghan vd., 2020). Bu tür uygulamalarda kullanılan cihazlar genellikle düşük güçlü ve kaynak kısıtlıdır. Bu durum, cihazların güvenli ve verimli bir şekilde çalışmasını sağlamak için özel protokollerin ve mekanizmaların geliştirilmesini zorunlu kılmaktadır.

Bu bağlamda, IPv6 üzerinden Zaman Paylaşımlı Kanal Atlamalı (6TiSCH) protokolü, düşük güçlü ve kayıplı ağlar için özel olarak tasarlanmış bir çözüm sunmaktadır (Kalita & Khatua, 2022; Külcü & Görmüş, 2022; Kulcu vd., 2021). 6TiSCH, IEEE 802.15.4e TSCH modunu kullanarak, düşük güç tüketimi ve yüksek güvenilirlik sağlayan bir ağ yapısı oluşturur. Ancak, 6TiSCH ağlarının güvenliği, özellikle kimlik doğrulama ve yetkilendirme süreçlerinde ciddi zorluklarla karşı karşıyadır (Kalita & Khatua, 2022; Morabito & Jiménez, 2020). Bu tür ağlarda, merkezi bir otoritenin tüm kimlik doğrulama işlemlerini gerçekleştirmesi, ölçeklenebilirlik ve verimlilik sorunlarına yol açabilir. Bu nedenle, dağıtık kimlik doğrulama mekanizmalarının geliştirilmesi, 6TiSCH ağlarının güvenliğini artırmak için kritik bir gerekliliktir (Vučinić vd., 2021). CoJP (Constrained Join Protocol) protokolü ile merkezi kimlik doğrulama işlemlerinin gerçekleştirilmesi, ölçeklenebilirlik ve verimlilik sorunlarına yol açabilir. Bu nedenle, merkezi yapının en önemli sorunlarını çözmek amacıyla Proxy-JRC (Join Registrar/Coordinator) kimlik doğrulama noktalarının dağıtık olarak belirlenmesi üzerine

çalışılmıştır. Dağıtık kimlik doğrulama, ağ içerisindeki belirli düğümlerin kimlik doğrulama işlemlerini üstlenmesini ve bu işlemleri merkezi otoriteye bağımlı olmadan gerçekleştirmesini sağlar.

Bu yaklaşım, ağın genel verimliliğini artırırken, ölçeklenebilirlik sorunlarını da minimize eder. Tezde, RPL (Routing Protocol for Low-Power and Lossy Networks) tabanlı 6TiSCH ağlarında Proxy-JRC kimlik doğrulama noktalarının belirlenmesi ve bu mekanizmaların geliştirilmesi üzerine yoğunlaşmıştır. RPL, düşük güçlü ve kayıplı ağlar için özel olarak tasarlanmış bir yönlendirme protokolüdür. Ancak, RPL'nin güvenlik zafiyetleri, ağın bütünlüğünü ve güvenilirliğini tehdit edebilir. Bu nedenle, RPL tabanlı 6TiSCH ağlarında kimlik doğrulama mekanizmalarının güçlendirilmesi büyük önem taşımaktadır. Çalışmada, özellikle düşük güçlü cihazların kaynak kısıtlı olması göz önünde bulundurularak, enerji verimli ve güvenli kimlik doğrulama mekanizmaları geliştirilmiştir. Düşük güçlü cihazlar, sınırlı işlem kapasitesi, bellek ve enerji kaynaklarına sahiptir. Bu kısıtlamalar, kimlik doğrulama işlemlerinin verimli ve düşük maliyetli olmasını gerektirir. Çalışmamızda, bu gereksinimleri karşılayan, merkezi otoritelerin yükünü dağıtan ve ağın genel güvenliğini artıran bir çözüm sunulmuştur.

Dağıtık kimlik doğrulama, ağ içerisindeki belirli düğümlerin kimlik doğrulama işlemlerini üstlenmesini ve bu işlemleri merkezi otoriteye bağımlı olmadan gerçekleştirmesini sağlar. Bu yaklaşım, ağın genel verimliliğini artırırken, ölçeklenebilirlik sorunlarını da minimize eder. Tezde, RPL tabanlı 6TiSCH ağlarında dağıtık kimlik doğrulama noktalarının belirlenmesi ve bu mekanizmaların geliştirilmesi üzerine yoğunlaşmıştır. Bu nedenle, RPL tabanlı 6TiSCH ağlarında kimlik doğrulama mekanizmalarının güçlendirilmesi büyük önem taşımaktadır.

Önerilen kimlik doğrulama mekanizması, dağıtık kimlik doğrulama noktalarının seçimine ek olarak, sensör ve kullanıcı arasındaki kimlik doğrulama mekanizmasını da içermektedir. Sensör ve kullanıcı kimlik doğrulama mekanizması, düşük güçlü ve kaynak kısıtlı cihazların güvenli ve verimli bir şekilde kimlik doğrulama işlemlerini gerçekleştirmesini sağlar. Önerilen mekanizmanın uygulanabilirliği ve etkinliği, çeşitli simülasyon ve deneysel çalışmalarla test edilmiştir.

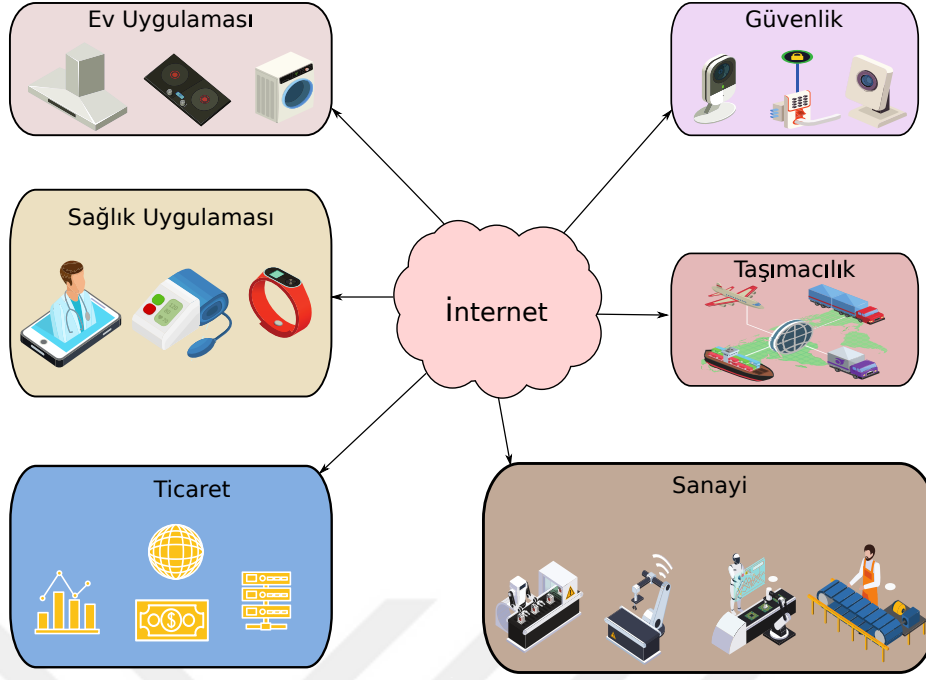
Sonuçlar, önerilen çözümün hem güvenlik hem de verimlilik açısından mevcut yöntemlere kıyasla üstün performans sergilediğini göstermektedir. Özellikle, düşük güçlü cihazların enerji tüketimini minimize ederek, ağın genel çalışma süresini uzattığı ve güvenlik tehditlerine karşı daha dirençli hale getirdiği gözlemlenmiştir. Dağıtık kimlik doğrulama

mekanizmalarının geliştirilmesi, sadece IIoT uygulamalarında değil, aynı zamanda genel olarak IoT teknolojisinin geleceği için de kritik bir öneme sahiptir. Çalışmamız, bu alanda önemli bir boşluğu doldurarak, daha güvenli ve verimli IIoT ağlarının oluşturulmasına katkıda bulunmayı amaçlamaktadır. Çalışmanın temel katkıları şu şekilde özetlenebilir:

- IIoT uygulamaları için Proxy-JRC tabanlı dağıtık kimlik doğrulama mekanizması geliştirilmiştir.
- 6TiSCH protokolü kapsamında Proxy-JRC kimlik doğrulama noktalarının belirlenmesi ve bu noktaların dağıtık olarak konumlandırılması sağlanmıştır.
- CoJP protokolü ile merkezi kimlik doğrulama süreçlerinin neden olduğu ölçeklenebilirlik ve verimlilik sorunlarına çözüm getirilmiştir.
- Önerilen dağıtık kimlik doğrulama mekanizmasının uygulanabilirliği ve etkinliği, çeşitli simülasyon ve deneysel çalışmalarla test edilmiştir.
- Geliştirilen yöntem, düşük güçlü cihazların enerji tüketimini minimize ederek, ağın genel çalışma süresini uzatmış ve güvenlik tehditlerine karşı daha dirençli hale getirmiştir.
- Sensör, kullanıcı ve sunucu arasında güvenli kimlik doğrulama mekanizması önerilmiştir; bu yöntem, düşük güçlü ve kaynak kısıtlı cihazların, sensör ve kullanıcı arasında hızlı, güvenilir ve enerji verimli kimlik doğrulama işlemlerini gerçekleştirmesini mümkün kılmaktadır.

1.2. Giriş

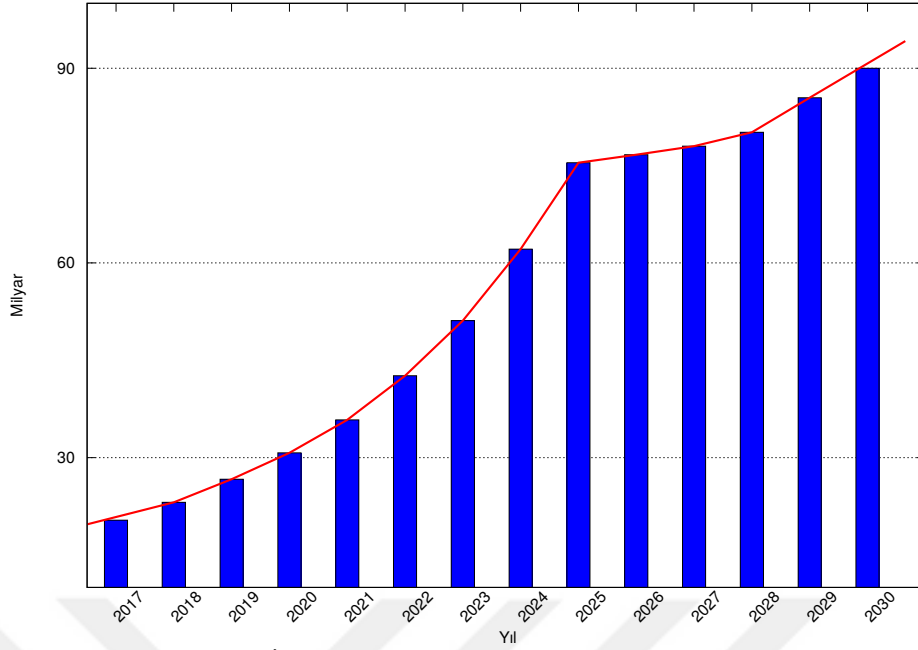
Kablosuz sensör ağları (KSA), küçük boyutlu, düşük maliyetli ve kısa mesafelerde kablosuz olarak iletişim kurabilen duyarga düğümlerinin bir araya gelmesiyle oluşan ağlardır. Bu ağlarda, düğümler düzenli veya rastgele olarak yerleştirilebilir ve protokoller yardımıyla kablosuz ortam üzerinden birbirleriyle iletişim kurabilir. Günümüzde KSA'lar sağlık, inşaat, kimya, çevre izleme ve ev otomasyonu gibi farklı alanlarda kullanılmaktadır. İnternete bağlanmış kablosuz duyarga ağlarına ait örnekler Şekil 1'de gösterilmektedir. Bu şekilde İnternete bağlı küçük cihazların oluşturduğu yeni nesil ağ yapısı "Nesnelerin İnterneti" olarak adlandırılmaktadır. Bu teknolojinin, akıllı sayaçlar ve akıllı şehirler gibi kavramlarla hayatımızda önemli bir yer tutması beklenmektedir (Kevin, 2009).



Şekil 1. Nesnelerin İnterneti kullanım alanları

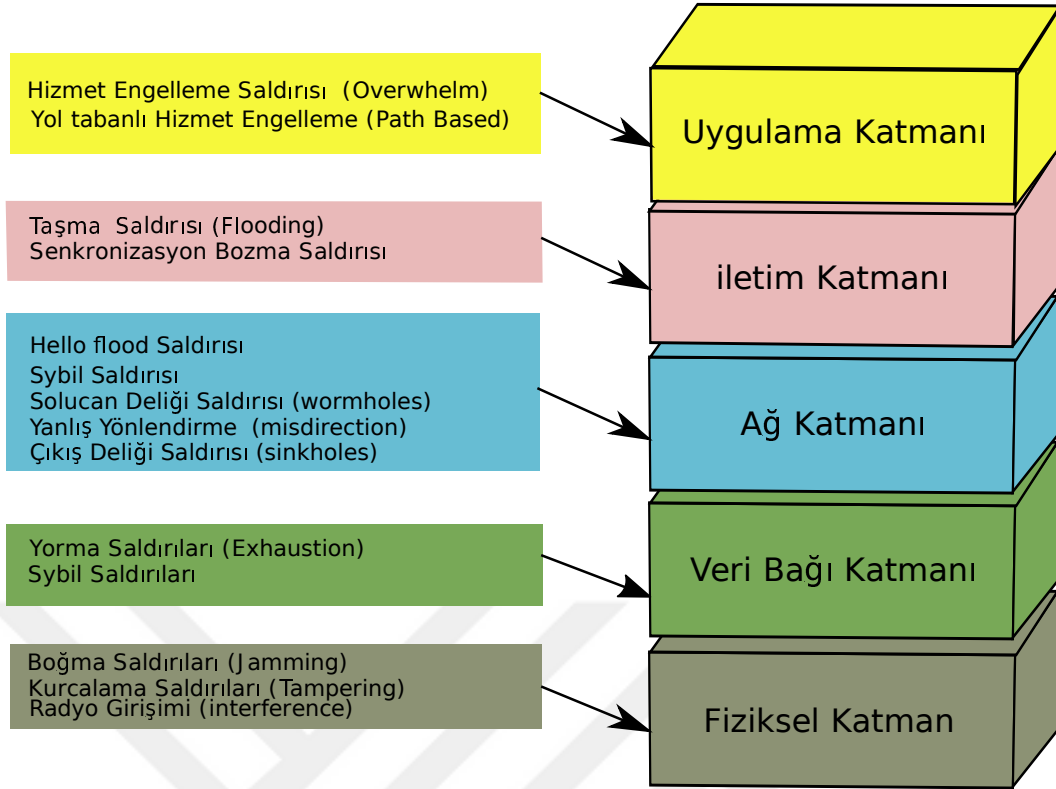
Nesnelerin İnterneti kavramı, Uluslararası Telekomünikasyon Birliği (ITU) tarafından herhangi bir yer ve zamanda her cihazın birbirleriyle iletişim kurabileceği bir teknoloji olarak tanımlanmaktadır (Siow vd., 2018). Bu teknoloji, çeşitli iletişim protokolleri ve algılama yöntemleri aracılığıyla tüm cihazların tanımlanıp birbirleriyle ve İnternet ile bağlantı kurabildiği akıllı ağlardan oluşan bir sistemler ağı olarak tanımlanabilir. Nesnelerin İnterneti fikrinin temelleri, 1991 yılında Cambridge Üniversitesi'nde 15 akademisyenin kahve makinesinin durumunu izlemek için kurduğu kameralı sistemle atılmıştır. 1999 yılında IoT kavramının önerilmesi ve 2005 yılında ITU tarafından bu yeni kavrama dair bir raporun yayımlanması, Nesnelerin İnterneti'ne olan ilgiyi büyük ölçüde artırmıştır.

Cisco tarafından yapılan araştırmaya göre, 2003 yılında internete bağlı cihaz sayısı 500 milyon iken, 2010 yılında bu sayı 12,5 milyara ulaşmış ve kişi başına düşen cihaz sayısı ise 1.84'e çıkmıştır (Ali vd., 2015). Aynı araştırma, 2020 yılında internete bağlanacak cihaz sayısının 50 milyara ulaşacağını öngörmüştür. 2030 yılına gelindiğinde ise bu sayının daha da artarak 90 milyara ulaşması beklenmektedir. Bu hızlı artış, IoT pazarının büyüklüğünü ve önemini göstermektedir. Şekil 2 yıllara göre artış göstermektedir. Bu cihazların internet erişiminin getirdiği güvenlik sorunları ile başa çıkmaları gerekecektir. IoT cihazları, kısıtlı bant genişliği, hafıza ve hesaplama yeteneğine sahip olmaları nedeniyle bu tür tehditlere karşı savunmasız kalmaktadır. Özellikle düşük güçlü ve kısıtlı kaynaklara sahip cihazlar, geleneksel güvenlik protokollerini uygulamada zorlanmaktadır. Bu nedenle,



Şekil 2. Nesnelerin İnterneti teknolojisinin yıllara göre kullanım sayıları

cihazların geleneksel güvenlik tekniklerini kullanarak internetin ortaya koyduğu güvenlik sorunlarıyla başa çıkmaları mümkün görülmemektedir. IoT ağlarındaki cihazların güvenlik açıkları, saldırganların bu cihazları hedef almasına neden olmaktadır. Fiziksel katmanda parazit ve sinyal bozma, MAC (Medium Access Control) katmanında çarpışma ve uykudan uyandırma saldırıları, ağ katmanında ise yönlendirme saldırıları gibi çeşitli saldırı türleri mevcuttur. Örneğin, DDoS (Distributed Denial of Service) saldırıları, IoT cihazlarının sınırlı kaynaklarını hızla tüketerek hizmet dışı kalmalarına yol açabilir. Ayrıca, Ortadaki Adam (Man-in-the-Middle - MITM) saldırıları, cihazlar arasındaki iletişimi ele geçirerek veri gizliliğini ihlal edebilir. Şekil 3'te kablosuz duyarga ağ katmanlarını etkileyen saldırı türleri özetlenmiştir. Bu saldırılar, ağın performansını ve güvenliğini ciddi şekilde tehdit edebilir. IoT cihazlarının artan pazar payı, bu cihazların güvenliğinin sağlanması gerekliliğini daha da önemli hale getirmektedir. 2030 yılına kadar, dünya genelinde milyarlarca cihazın internete bağlı olacağı düşünüldüğünde, bu cihazların güvenlik açıkları ciddi sorunlara yol açabilir. Bu nedenle, IoT ağlarında kullanılan cihazlar için yeni ve etkin güvenlik çözümlerinin geliştirilmesi büyük bir önem taşımaktadır. Örneğin, 2015 Ukrayna elektrik şebekesi saldırısı, saldırganların "Black Energy" adlı kötü amaçlı yazılımı kullanarak internet üzerinden elektrik şirketinin bilgisayar sistemlerine sızdığı ve altyapı hasarına neden olduğu tipik bir endüstriyel internet saldırısı vakasıdır. Bu olay, altyapıya yönelik saldırıların neden olabileceği sonuçları göstermiştir (Yang vd., 2017). Dolayısıyla, geleneksel saldırı tespit yöntemleri günümüz



Şekil 3. Nesnelerin İnterneti teknolojisinin katmanlara göre saldırı türleri

endüstriyel internet ortamına uygun değildir ve bu tür sistemlerin güvenliğini korumak için yeni yöntem ve teknolojilerin geliştirilmesi ve uygulanması gerekmektedir.

Tezde, RPL tabanlı 6TiSCH ağlarında güvenli dağıtık kimlik doğrulama mekanizmalarının geliştirilmesi üzerine odaklanılmaktadır. Çalışmanın temel amacı, merkezi doğrulama adımlarını ağın stratejik noktalarında yaparak, dağıtık olarak düğüm doğrulama işlemlerini gerçekleştirmek ve böylece merkezi yöntemlerin dezavantajı olan ölçeklenebilirlik sorununu çözmektir. Bu yaklaşım, IoT cihazlarının güvenliğini artırarak, tehditlere karşı daha dayanıklı bir ağ yapısı oluşturmayı hedeflemektedir.

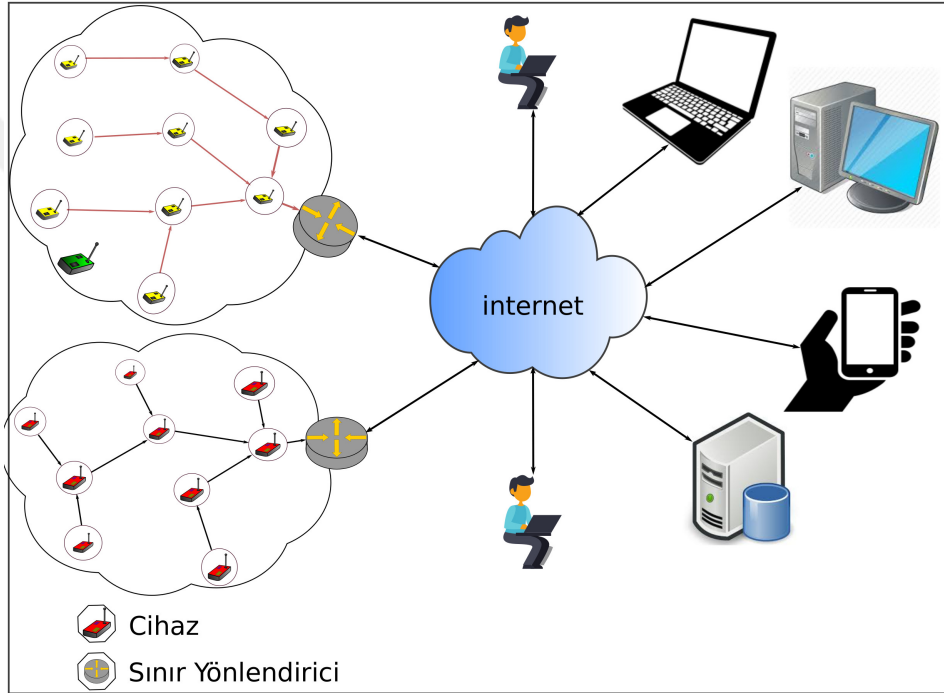
Tez çalışması altı ana bölümde organize edilmiştir. İlk olarak, 2. Bölümde, Nesnelerin İnterneti için kullanılan protokol yığını ve bu alandaki ilgili çalışmalar detaylı bir şekilde incelenmiştir. Bu bölümde, IoT'nin temel bileşenleri ve iletişim protokollerinin yanı sıra, literatürdeki güncel gelişmeler ve çalışmalar ele alınmıştır. 3. Bölümde, 6TiSCH ağlarının güvenlik gereksinimleri ve bu gereksinimleri karşılamaya yönelik mevcut çözümler ele alınmıştır. 4. Bölüm, çalışmanın ana odağını oluşturan güvenli dağıtık kimlik doğrulama mekanizmalarının geliştirilmesi ve teknik detaylarının sunulmasına ayrılmıştır. Bu bölümde, önerilen çözümler ve bu çözümlerin teknik bileşenleri detaylandırılmaktadır. Geliştirilen yöntemlerin nasıl çalıştığı ve ağ üzerindeki güvenlik ve performans etkileri ayrıntılı olarak

açıklanmaktadır. 5. Bölümde, simülasyon sonuçları ve performans analizleri sunulmuş olup, önerilen mekanizmaların etkinliği ve verimliliği üzerine yapılan değerlendirmelere yer verilmiştir. Bu bölümde, simülasyon sonuçlarına dayalı olarak, önerilen mekanizmaların mevcut yöntemlere göre etkili performans sergilediği gösterilmiştir. Çalışmanın genel sonuçları ve gelecekteki çalışmalar için öneriler 6. Bölümde sunulmuştur. Ayrıca, tez çalışmasının bulguları özetlenmiş ve gelecekte yapılacak araştırmalar için potansiyel yönler tartışılmıştır. Dağıtık kimlik doğrulama mekanizmasının ve sadeleştirilmiş kimlik doğrulama yönteminin IIoT ve IoT teknolojilerinin geleceği için kritik bir öneme sahip olduğunu vurgulamaktadır.

Son olarak, 7. Bölümde de gelecekte yapılacak çalışmalar için öneriler ve tartışmalar sunulmaktadır. Bu bölümde, çalışma kapsamında elde edilen sonuçların genişletilmesi ve daha ileri araştırmaların hangi alanlara odaklanması gerektiği tartışılmaktadır. Bu yapısal düzenleme, çalışmanın amaçlarına ulaşmak için izlenecek adımları ve elde edilen bulguları sistematik bir şekilde sunmayı hedeflemektedir.

2. NESNELERİN İNTERNETİ İÇİN PROTOKOL YIĞINI

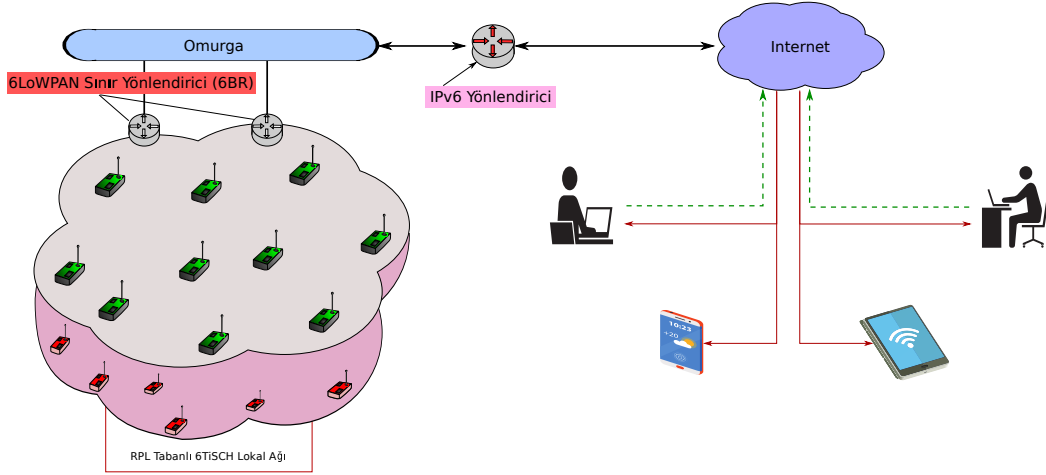
Nesnelerin İnterneti, çeşitli cihazların birbirleriyle ve internetle iletişim kurmasını sağlayan geniş bir ekosistemdir. IoT cihazlarının verimli ve güvenli bir şekilde çalışabilmesi için belirli protokoller ve standartlar kullanılır. Bu bölümde, IoT için kullanılan protokol yığını detaylı bir şekilde açıklanacaktır. Şekil 4, tipik bir kablosuz sensör ağı örneğini göstermektedir.



Şekil 4. Nesnelerin İnterneti teknolojisinde yer alan bileşenlerin örnek bir gösterimi

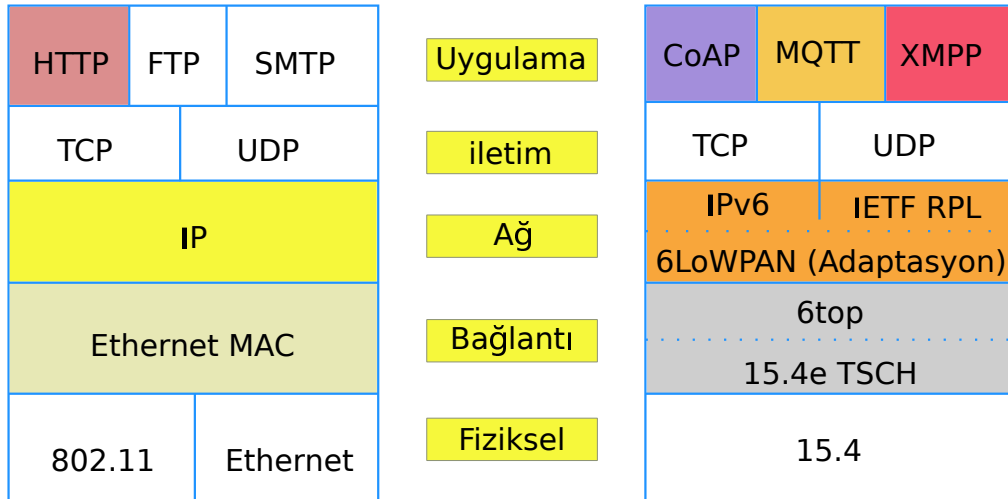
Şekilde gösterilen ağda, 6LoWPAN (IPv6 over Low Power Personal Area Network) sınır yönlendiricisi (border router), KSA'yı internete bağlamakla görevlidir. Bu cihaz, düşük güçlü ve kayıplı ağlardaki (Low-Power and Lossy Networks - LLNs) cihazların internete erişimini sağlamaktadır. Hem IPv6 hem de 6LoWPAN protokollerini destekleyerek, sensör düğümlerinin daha geniş ağlara entegre olmasına olanak tanır. Böylece, IoT cihazlarının verimli bir şekilde veri toplaması ve iletmesi mümkün hale gelir. KSA'larda, çeşitli sensör düğümleri fiziksel çevreyi izler ve topladıkları verileri kablosuz olarak sınır yönlendiriciye gönderir. Sınır yönlendirici, verileri internete aktararak kullanıcılar ve diğer uygulamalar için erişilebilir kılar.

Şekil 5, RPL tabanlı 6TiSCH lokal ağı ve 6LoWPAN sınır yönlendiricisinin geleneksel KSA üzerindeki kullanımını gösterir. RPL tabanlı 6TiSCH lokal ağı, küçük ve düşük güçlü



Şekil 5. 6LoWPAN ile KSA'yı İnternete bağlayan tipik bir ağ gösterimi

cihazların (yeşil ve kırmızı düğümler) birbirleriyle iletişim kurduğu bir sistemdir. Sensörler, çevresel verileri toplayarak sınır yönlendiriciye (6BR) iletir. RPL, düşük güçlü cihazlar arasında güvenilir ve verimli veri iletimi sağlar. 6TiSCH, zaman dilimli kanal atlama (Time Slotted Channel Hopping - TSCH) ile birlikte çalışarak, ağın enerji verimliliğini ve güvenilirliğini artırır. 6LoWPAN sınır yönlendirici ise düşük güçlü ve kayıplı ağları internete bağlayan bir geçiş noktasıdır. 6BR, IPv6 ve 6LoWPAN protokollerini destekleyerek, IoT cihazlarının internete erişimini sağlar. Sınır yönlendirici, veri paketlerini sıkıştırır ve parçalara ayırır, böylece düşük bant genişliği ve düşük güç tüketimi ile veri iletimi gerçekleştirilebilir.



Şekil 6. Standart TCP/IP modeli ve 6LoWPAN tabanlı protokol yığını

Şekil 6, standart TCP/IP modeli ile 6LoWPAN tabanlı protokol yığını arasındaki farkları ve benzerlikleri göstermektedir. Sol tarafta yer alan standart TCP/IP modeli, geleneksel internet protokollerini temsil eder ve fiziksel katmandan başlayarak uygulama katmanına kadar uzanır. Fiziksel katmanda, Wi-Fi olarak bilinen 802.11 ve kablolu yerel alan ağları (Local

Area Networks - LAN) için kullanılan Ethernet bulunur. Bağlantı katmanında, Ethernet'in MAC katmanı, veri çerçevelerini adresler ve fiziksel ortamda veri iletimini sağlar. Ağ katmanında, IP adresleme ve yönlendirme işlevlerini yerine getirir. İletim katmanında, TCP ve UDP protokolleri bulunur; TCP, bağlantı tabanlı ve güvenilir veri iletimi sağlarken, UDP bağlantısız, hızlı ama güvenilir olmayan veri iletimi sağlar. Uygulama katmanında ise, web sayfalarının iletimi için kullanılan HTTP (HyperText Transfer Protocol), dosya transferi için kullanılan FTP (File Transfer Protocol) ve e-posta iletimi için kullanılan SMTP (Simple Mail Transfer Protocol) yer alır.

Sağ tarafta yer alan 6LoWPAN tabanlı protokol yığını ise, düşük güçlü ve kayıplı ağlar için optimize edilmiştir. Fiziksel katmanda, düşük veri hızlı kişisel alan ağları (LR-WPANs) için bir standard olan IEEE 802.15.4 bulunur. Bağlantı katmanında, zaman dilimli kanal atlama standardı olan IEEE 802.15.4e TSCH ve zaman dilimlerinin planlanmasını ve yönetimini sağlayan 6top yer alır. Ağ katmanında, IPv6 paketlerini düşük güçlü ve kayıplı ağlara uyarlamak için kullanılan 6LoWPAN (adaptasyon katmanı), geniş adresleme kapasitesi sağlayan IPv6 ve LLNs için özel olarak tasarlanmış bir yönlendirme protokolü olan IETF RPL (Routing Protocol for Low-Power and Lossy Networks - Düşük Güçlü ve Kayıplı Ağlar için Yönlendirme Protokolü) bulunur. İletim katmanında, TCP ve UDP protokolleri, güvenilir ve hızlı veri iletimi sağlar. Uygulama katmanında ise, kısıtlı cihazlar için hafif bir uygulama protokolü olan CoAP (Constrained Application Protocol - Kısıtlı Uygulama Protokolü), hafif bir mesajlaşma protokolü olan MQTT (Message Queuing Telemetry Transport - Mesaj Kuyruklu Telemetri Aktarımı) ve gerçek zamanlı iletişim için kullanılan XMPP (Extensible Messaging and Presence Protocol - Genişletilebilir Mesajlaşma ve Varlık Protokolü) yer alır. Bu protokoller, IoT cihazlarının çeşitli uygulama ihtiyaçlarını karşılamak için özel olarak tasarlanmış olup, verimli ve güvenli veri iletimi sağlar.

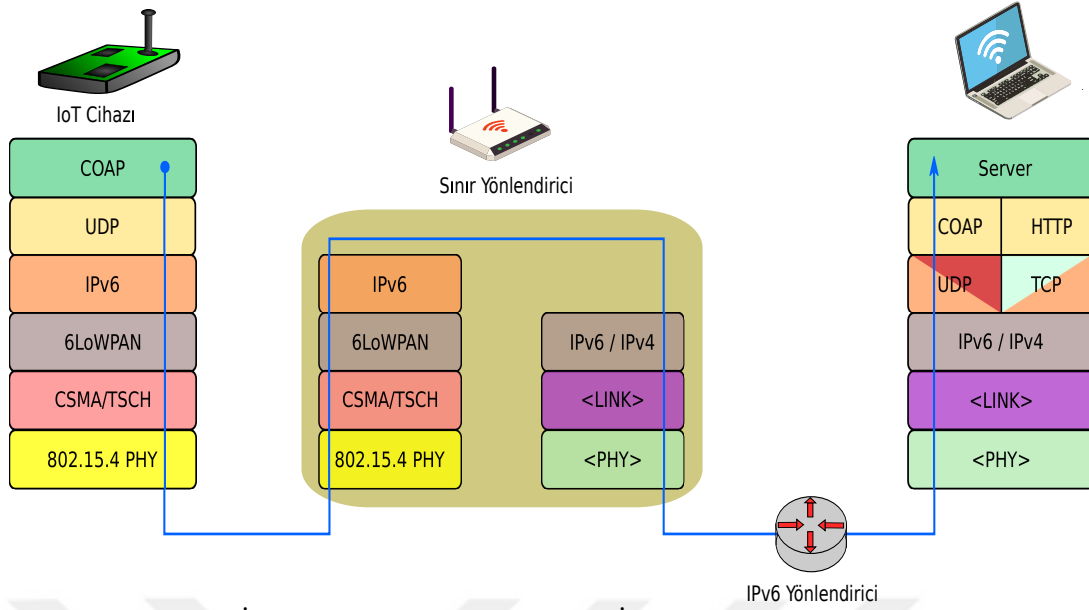
Standart TCP/IP modeli ile 6LoWPAN tabanlı protokol yığını arasındaki temel farklılıklar, 6LoWPAN'ın düşük güçlü ve kayıplı ağlar için optimize edilmiş olması ve IoT uygulamalarına özel çözümler sunmasıdır. Örneğin, TCP/IP modelinde yüksek bant genişliği ve güçlü işlem kapasitesi olan cihazlar hedeflenirken, 6LoWPAN modeli düşük güç tüketimi ve sınırlı kaynaklara sahip cihazlar için daha uygun hale getirilmiştir. Bu durum, 6LoWPAN'ın özellikle IoT uygulamaları için ideal bir protokol yığını olmasını sağlar. Ayrıca, 6LoWPAN'ın düşük güç tüketimi ve geniş adresleme kapasitesi gibi avantajları, IoT ağlarının ölçeklenebilirliğini ve verimliliğini artırmaktadır.

2.1. IEEE 802.15.4 Standardı

IEEE 802.15.4 standardı, düşük güçlü ve düşük veri hızlı kişisel alan ağları için tasarlanmış bir protokoldür. Bu standardın temel amacı, düşük maliyetli ve enerji tasarruflu iletişim sağlamak olup, özellikle Nesnelerin İnterneti ve endüstriyel otomasyon uygulamaları gibi alanlarda yaygın olarak kullanılmaktadır. Ayrıca, onlarca metreye kadar mesafelerde ve 250 kbps'ye kadar veri hızlarında iletişim sağlamayı hedeflemektedir. Bir cihaz internete bağlanmak için IP kullanmalıdır. Çok sayıda potansiyel IoT cihazı ve bunları barındırmak için IPv4'teki (2^{32} benzersiz adres) adres eksikliği nedeniyle, genellikle IPv6 (2^{128} benzersiz adres) kullanılır. Kişisel Alan Ağı (Personal Area Network - PAN), kişisel alan ağlarını ifade eder ve bu ağlar genellikle küçük bir coğrafi alan içinde, düşük güçlü cihazların birbirleriyle iletişim kurmasını sağlar. IEEE 802.15.4 standardı, PAN koordinatörleri tarafından yönetilen bu tür ağları destekler. Koordinatör, ağdaki diğer cihazların (düğümlerin) haberleşmesini yönetir ve senkronize eder. IEEE 802.15.4 standardında cihazlar iki kategoriye ayrılır: Full Function Device (FFD) ve Reduced Function Device (RFD). FFD, tüm IEEE 802.15.4 fonksiyonlarını destekleyen ve PAN koordinatörü olabilen cihazlardır. RFD ise sınırlı işlevselliğe sahip olup, yalnızca son cihaz olarak görev yapar ve genellikle düşük enerji tüketimine sahip uygulamalarda kullanılır. FFD'ler, ağın diğer üyeleriyle doğrudan iletişim kurabilme yeteneğine sahiptir ve gerektiğinde yeni bağlantılar kurarak ağın genişlemesine yardımcı olabilir.

RFD'ler, sensörler ve aktüatörler gibi basit veri toplama ve kontrol görevlerini yerine getiren cihazlar olarak yaygın bir şekilde kullanılır. IEEE 802.15.4 standardı, FFD ve RFD cihazlarının birlikte çalışarak enerji verimli ve esnek ağlar oluşturmasını sağlar. FFD cihazlarının ağın merkezi işlevlerini üstlenmesi, RFD cihazlarının ise düşük güç tüketimiyle belirli görevleri yerine getirmesi, ağın genel performansını optimize eder (Şekil 7). Bu yapı, özellikle geniş alanlarda dağıtık sensör ağlarında, ev otomasyon sistemlerinde ve endüstriyel kontrol uygulamalarında yaygın olarak kullanılmaktadır. Cihazların etkileşimi, ağın esnekliğini ve ölçeklenebilirliğini artırarak, çeşitli uygulamalarda güvenilir ve verimli iletişim sağlar.

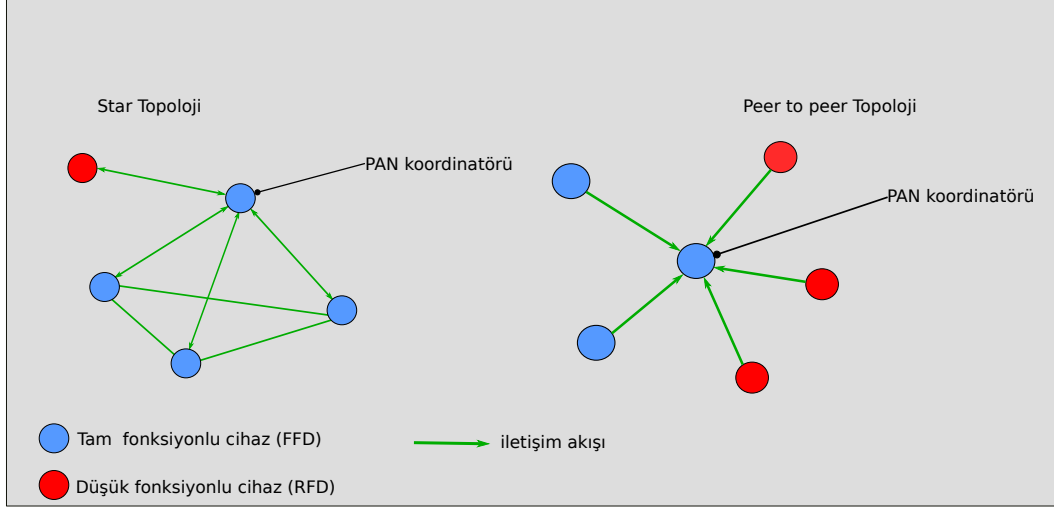
IEEE 802.15.4 standardı, veri iletimi sırasında çakışmaların önlenmesi için Taşıyıcı Algılama Çarpışma Önleme ile Çoklu Erişim (Carrier Sense Multiple Access with Collision Avoidance - CSMA/CA) mekanizmasını kullanır. Çakışma (collision), iki veya daha fazla cihazın aynı anda veri göndermeye çalışması durumunda ortaya çıkan bir durumdur. Bu durum, gönderilen verilerin karışmasına ve her iki cihazın da iletiminin başarısız olmasına



Şekil 7. Nesnelerin İnterneti cihazı ile geleneksel İnternet ana makinesi için protokol yığını ve bu cihazların 6LoWPAN sınır yönlendirici kullanılarak entegrasyonu

neden olur. CSMA/CA yöntemi, cihazın veri göndermeden önce kanalın boş olup olmadığını kontrol etmesini sağlar. Eğer kanal boşsa, cihaz veriyi iletir; değilse, belirli bir süre bekleyip tekrar kontrol eder. Bu mekanizma, çakışmaları azaltarak veri iletiminin daha güvenilir olmasını sağlar. IEEE 802.15.4 standardı, yıldız (star) ve eşler arası (peer-to-peer) gibi çeşitli ağ topolojilerini destekler. Şekil 8’de, bu ağ topolojilerinde bulunan cihazların ve ağ kordinatörünün sahip olduğu bağlantılar görülmektedir. Yıldız topolojisine sahip ağlarda iletişim, cihazlar ve PAN koordinatörü olarak adlandırılan tek bir merkezi kontrolcü arasındadır. Bu topolojide bir cihaz ağ iletişiminin başlatıcısı ya da son alıcısıdır. PAN koordinatörü, bir FFD olarak, ağ etrafında iletişimi başlatan, sonlandıran ve yönlendiren birincil denetleyicidir. Bir cihaz PAN’daki başka bir cihazla iletişim kurmak isterse mesajı PAN koordinatörüne gönderir, koordinatör de mesajı hedef cihaza iletir.

Eşler arası bir ağda, her cihaz kendi ortamındaki diğer tüm cihazlarla doğrudan iletişim kurabilir. Ayrıca bir PAN koordinatörü bulunur; ancak bu koordinatör iletişimin merkezi noktası değildir. Bu teknoloji, örgü (mesh) topolojisini de içeren daha karmaşık ağ oluşumlarının uygulanmasına olanak tanır. Örgü topolojisinde, her cihaz birden fazla cihazla bağlantı kurabilir, bu da ağın esnekliğini ve yeniden yönlendirme yeteneklerini artırır. Eğer bir cihaz veya bağlantı arızalanırsa, veri farklı bir yol üzerinden iletilebilir. Dinamik yönlendirme protokolleri ile cihazlar arasındaki veri iletimi yönetilir, bu da ağdaki her cihazın en iyi yolu bulmasını ve veriyi hedef cihaza iletmesini sağlar. Dinamik yönlendirme, ağın genişlemesini ve yeni cihazların eklenmesini kolaylaştırır, çünkü her yeni cihaz otomatik olarak ağın bir



Şekil 8. Star ve Peer to Peer topoloji örnekleri

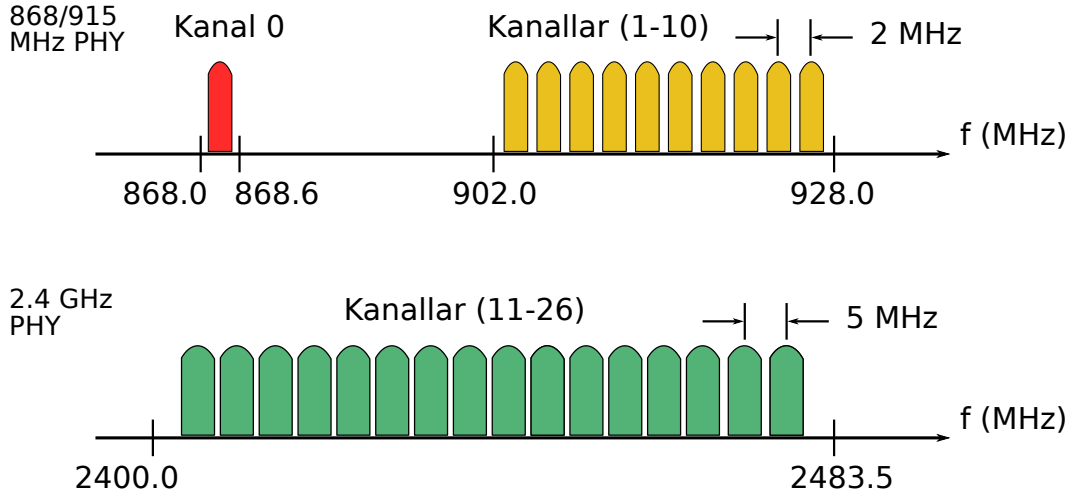
Tablo 1. IEEE 802.15.4 radyo özellikleri

Frekans (MHz)	Kanal	Bölge	Veri Hızı (kbit/s)	Baud Hızı (kBaud)	Modülasyon
868-868.6	0	Avrupa	20	20	BPSK
902-928	1- 10	Amerika	40	40	BPSK
2400-2483.5	11 - 26	Dünya	250	65.5	O-QPSK

parçası haline gelir ve mevcut cihazlarla iletişim kurabilir. IEEE 802.15.4 standardı, düşük güç tüketimi ve düşük veri hızı gerektiren uygulamalar için optimize edilmiştir. Tipik veri hızları 20 kbps, 40 kbps, 100 kbps, 250 kbps ve 1 Mbps arasında değişir. Bu düşük veri hızları, cihazların pil ömrünü uzatır ve enerji verimliliğini artırır. IEEE 802.15.4 standardının radyo özellikleri Tablo 1’de gösterilmiştir.

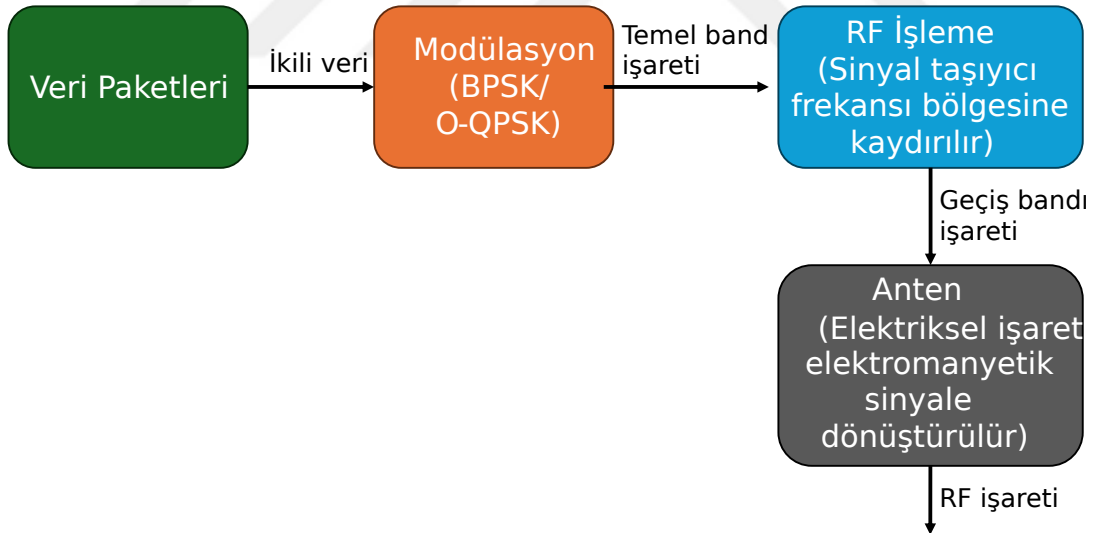
Her frekans bandının kendine özgü parametreleri bulunmaktadır ve bu, Şekil 9’da gösterilmiştir. Bu parametreler arasında taşıyıcı frekansı, iletim gücü, iletim görev devri, maksimum ve minimum iletim süreleri, ve anten özellikleri yer almaktadır. Fiziksel adres kullanımı, cihazların aynı fiziksel bağlantıyı kullanmalarını sağlarken, her cihazın benzersiz şekilde tanımlanmasına yardımcı olur. Bu sayede, cihazlar arasında veri iletimi daha düzenli ve verimli bir şekilde gerçekleşir. Bir radyo kanalı, bir sayı ile adlandırılan belirli bir frekans, frekans çifti veya bandıdır. IEEE 802.15.4-2003, 0’dan 26’ya kadar numaralandırılmış 27 kanala sahiptir. Yük alanının uzunluğu değişkendir; ancak MAC çerçevesinin tamamının uzunluğu 127 baytı geçemez.

IEEE 802.15.4 standardında kullanılan farklı frekans bantları, dünya genelinde farklı bölgesel düzenlemelere uyum sağlamak ve çeşitli uygulama senaryolarında esneklik sağlamak amacıyla bulunmaktadır. Her bölgenin radyo spektrumu kullanımına ilişkin kendi



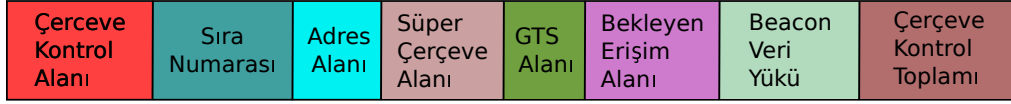
Şekil 9. IEEE 802.15.4 standardının desteklediği frekans bantları

düzenlemeleri ve kısıtlamaları vardır. Bu nedenle, IEEE 802.15.4 standardı, Avrupa, Amerika ve dünya genelinde uyumlu olabilmek için farklı frekans bantları belirlemiştir. Örneğin, Avrupa’da 868-868.6 MHz bandı kullanılırken, Amerika’da 902-928 MHz bandı kullanılır. Ayrıca, dünya genelinde yaygın olarak kullanılan 2400-2483.5 MHz bandı da mevcuttur. Bu çeşitlilik, IEEE 802.15.4 standardının global olarak benimsenmesini ve farklı coğrafi bölgelerdeki cihazların birbirleriyle uyumlu bir şekilde çalışmasını sağlar.



Şekil 10. Veri paketlerinin radyo katmanından çıkış süreci

IEEE 802.15.4 standardında, veri paketleri radyo katmanından radyo sinyalleri ile iletilir ve alıcı tarafında tekrar veri paketlerine dönüştürülür. Gönderim sürecinde, veri paketleri belirli bir modülasyon tekniği kullanılarak radyo dalgalarına dönüştürülür. Örneğin, 868 MHz ve 902-928 MHz bantlarında İkili Faz Kaydırmalı Anahtarlama (Binary Phase Shift Keying-BSPK) modülasyonu kullanılırken, 2400-2483.5 MHz bandında O-QPSK (Offset Quadrature Phase Shift Keying) modülasyonu kullanılır. Bu modülasyon teknikleri, veri



Şekil 11. Beacon çerçevesi

bitlerini radyo sinyallerine dönüştürerek fiziksel ortamda iletilmesini sağlar. Alıcı cihaz, gelen radyo sinyallerini alır ve demodülasyon işlemi ile tekrar veri bitlerine dönüştürür. Bu süreçte, radyo sinyalleri anten tarafından alınır, radyo frekans (Radio Frequency-RF) devresi tarafından işlenir ve dijital sinyallere dönüştürülerek alıcı cihazın veri katmanına iletilir. Böylece, gönderilen veri paketleri başarılı bir şekilde alınmış olur ve iletişim tamamlanır. Şekil 10, bu süreci detaylı olarak göstermektedir.

Ortam Erişim Katmanı (MAC) çerçeveleri, IEEE 802.15.4 standardında dört temel mesaj formatında tanımlanmıştır: beacon çerçeveleri, veri çerçeveleri, onay (acknowledgment) çerçeveleri ve MAC komut çerçeveleri. Bu çerçevelerin her biri belirli bit uzunluklarına sahip alanlardan oluşur. Bu çerçevelerin detaylı açıklamaları ve ilgili bit uzunlukları devam eden bölümlerde açıklanmıştır.

2.1.1. Beacon Çerçevesi

Beacon çerçevesi, IEEE 802.15.4 MAC katmanında ağ iletişiminin koordinasyonu ve senkronizasyonu için kritik öneme sahiptir. Bu çerçeveler, özellikle zaman dilimli ağlarda, düğümlerin veri iletimi ve alımı için belirli zaman dilimlerinde uyanmasını sağlayarak enerji verimliliğini artırır. Beacon çerçeveleri, ağdaki düğümlere senkronizasyon bilgisi, ağ kimliği ve süper çerçeve yapısı gibi temel bilgileri iletir. Bu çerçeveler, ağ kurulumunda ve düğümlerin ağa katılımında önemli rol oynar. Aynı zamanda, düğümler arası iletişimin zamanlamasını ve sırasını belirleyerek çakışmaları ve veri kaybını önler. IEEE 802.15.4 standardı, beacon çerçevelerinin düzenli aralıklarla yayınlanmasını ve bu sayede düğümlerin zamanlama bilgilerini sürekli güncelleyerek ağın tutarlılığını ve verimliliğini sağlamasını öngörür. Bu özellikler, beacon çerçevelerini, düşük güçlü ve kayıplı ağlarda güvenilir ve enerji verimli bir iletişim sağlamada vazgeçilmez kılar. Şekil 11’de beacon çerçevesinin yapısı detaylı olarak gösterilmektedir.

Maddeler halinde beacon çerçeve yapısı verilmiştir:

- Çerçeve Kontrol Alanı (16 bit): Çerçeve türü, güvenlik etkinleştirme, ACK isteği gibi kontrol bilgilerini içerir.

- Sıra Numarası (8 bit): Beacon çerçevesindeki Sequence Number alanı, 8 bit uzunluğundadır ve her bir çerçeveye özgü bir numarayı temsil eder. Bu numara, alıcı cihazların çerçeveleri doğru sıralamayla işlemesine yardımcı olur ve aynı zamanda çerçeve kaybı veya tekrarını tespit etmelerine olanak tanır.
- PAN ID (16 bit): PAN ID alanı, 2 bayt uzunluğundadır ve belirli bir PAN kimliğini temsil eder. Bu alan, ağ içindeki cihazların hangi PAN'e ait olduklarını tanımlamalarını sağlar ve ağlar arasında doğru iletişimi güvence altına alır.
- Kaynak Adresi (16 bit veya 64 bit): Bu alan, beacon çerçevesini gönderen cihazın adresini içerir. Cihazların iletişim kurduğu ağdaki diğer cihazlar, bu alanı kullanarak veri paketlerinin kaynağını belirler ve gerektiğinde geri dönüş yapar.
- Süper Çerçeve Tanımları (16 bit): Bu birim, ağın süper çerçeve yapısını tanımlar. Süper çerçeve, aktif ve pasif dönemler içeren bir zaman dilimidir ve bu alan, süper çerçevenin uzunluğu, aktif ve pasif dönemlerin süresi gibi bilgileri içerir. Bu sayede, cihazlar veri iletimi için uygun zaman dilimlerini bilirler.
- GTS Alanı (8 bit): Bu alan, garantili zaman dilimlerinin (Guaranteed Time Slot - GTS) yer aldığı bilgileri içerir. GTS, belirli cihazlara veri iletimi için ayrılmış özel zaman dilimleridir. Bu alan, GTS talep eden ve kullanan cihazların adreslerini ve bu dilimlerin başlangıç ve bitiş zamanlarını belirtir.
- Bekleyen Erişim Alanı (değişken): Bu alan, ağdaki cihazlara henüz teslim edilmemiş bekleyen veri paketlerinin adreslerini içerir. Cihazlar, bu alanı kontrol ederek kendilerine yönelik bekleyen veri paketlerinin olup olmadığını öğrenir ve gerekli ise veri iletimini gerçekleştirir. Bu, ağdaki veri iletimini düzenli ve verimli hale getirir.
- Beacon Veri Yüğü (değişken): Beacon çerçevesinin veri yükü, ağın yapılandırması ve çalışma durumu hakkında ek bilgiler içerebilir. Bu bilgiler, özel uygulama ihtiyaçlarına göre özelleştirilebilir ve ağın genel performansını optimize etmek için kullanılabilir.

2.1.2. Veri Çerçevesi

IEEE 802.15.4 standardında kullanılan veri çerçeveleri, ağdaki cihazlar arasında veri iletimi sağlamak amacıyla tasarlanmıştır. Bu çerçeveler, çeşitli alanlardan oluşur ve her bir



Şekil 12. Veri çerçevesi

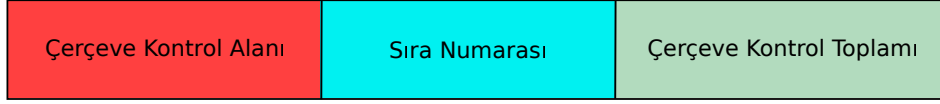
alan, çerçevenin iletimi sırasında gerekli olan bilgileri taşır. Şekil 12, veri çerçevesinin detaylı halini göstermektedir.

- **Çerçeve Kontrol Alanı (16 bit):** Bu alan, çerçevenin türünü, güvenlik etkinleştirmesini, ACK isteğini ve PAN ID sıkıştırmasını belirler. Örneğin, Frame Type (3 bit) çerçevenin türünü belirtirken, Security Enabled (1 bit) güvenlik özelliklerinin etkin olup olmadığını gösterir. Ayrıca, Frame Pending (1 bit) çerçevenin ardından başka veri çerçevelerinin olup olmadığını belirtir ve Acknowledgment Request (1 bit) çerçevenin alıcısından ACK isteyip istemediğini ifade eder. PAN ID Compression (1 bit) ise PAN ID alanlarının sıkıştırılıp sıkıştırılmadığını belirtir.
- **Sıra Numarası (8 bit):** Çerçeve kontrol alanının ardından gelen Sequence Number (8 bit) alanı, çerçevenin sıra numarasını içerir ve bu da çerçevelerin sırasıyla işlenmesini sağlar.
- **Adres Alanı (16 bit):** Adres alanı, hedef ve kaynak PAN kimliklerini ve adreslerini içerir. Destination PAN ID (16 bit) ve Source PAN ID (16 bit), hedef ve kaynak PAN kimliklerini belirtirken, Destination Address (16 bit veya 64 bit) ve Source Address (16 bit veya 64 bit) hedef ve kaynak cihazların adreslerini içerir. Bu adresleme bilgileri, çerçevenin doğru bir şekilde hedefe yönlendirilmesini sağlar.
- **Veri Yüğü (değişken):** Bu veri alanı, uygulama katmanından gelen bilgileri içerir ve uzunluğu ağdaki yapılandırmaya ve gereksinimlere bağlı olarak değişir. Veri çerçevelerinin sonunda yer alan MAC Footer (MFR) bölümü ise Frame Check Sequence (16 bit) alanını içerir.
- **Çerçeve Kontrol Toplamı (16 bit):** Bu alan, hata tespiti için kullanılan bir CRC (Cyclic Redundancy Check) değeri içerir ve çerçevenin bütünlüğünü doğrulamak için kullanılır.

2.1.3. Onay (Acknowledgment) Çerçevesi

Onay çerçevesi, IEEE 802.15.4 standardında ağdaki cihazlar arasında veri iletimini doğrulamak amacıyla kullanılan basit çerçevelerdir. Bu çerçeveler, veri çerçevesinin başarılı

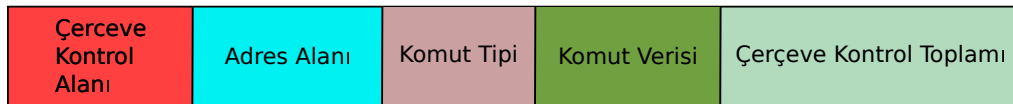
bir şekilde alındığını gönderen cihaza bildirmek için kullanılır. Çerçeve Kontrol Alanı, Sıra Numarası ve Çerçeve Kontrol Toplamı olmak üzere üç bölümden oluşan ACK çerçevesi, Şekil 13'te gösterilmiştir.



Şekil 13. ACK çerçevesi

- Çerçeve Kontrol Alanı (16 bit): Bu alan, ACK çerçevesinin yapısını ve işlevselliğini tanımlayan bilgileri barındırır. Örneğin, çerçevenin tipini belirten bitler ve kontrol bilgilerini içeren diğer bitler burada yer alır.
- Sıra Numarası (8 bit): Bu bölüm, alınan veri çerçevesinin sıra numarasını içerir. Bu numara, gönderilen veri çerçevesiyle ilişkilendirilir ve veri çerçevesinin başarılı bir şekilde alındığını doğrulamak için kullanılır. Alıcı cihaz, bu numarayı kullanarak hangi veri çerçevesinin alındığını belirler ve buna uygun olarak ACK çerçevesini oluşturur.
- Çerçeve Kontrol Sırası (FCS) (16 bit): hata kontrol bilgilerini içerir. Bu alan, çerçevenin doğruluğunu sağlamak için kullanılır ve gönderilen verilerin hatasız bir şekilde alındığını doğrulamak için hata tespiti yapar. FCS, çerçevenin sonunda yer alır ve CRC algoritması kullanılarak hesaplanır.

Bu bileşenler bir araya gelerek, MAC Komut Çerçevesi'nin ağdaki kontrol işlemlerini verimli ve güvenilir bir şekilde gerçekleştirmesini sağlar. Şekil 14'de gösterilen MAC Komut Çerçeveleri, ağ yönetimi, cihazların bağlanması ve bağlantılarının kesilmesi gibi çeşitli işlemler için kritik öneme sahiptir. Bu çerçeveler ayrıca, ağın genel performansını optimize etmek ve iletişim protokollerinin etkinliğini artırmak için tasarlanmıştır.



Şekil 14. MAC komut çerçevesi

- Çerçeve Kontrol Alanı (16 bit): Bu alan, çerçevenin türü, güvenlik etkinleştirme ve ACK isteği gibi kontrol bilgilerini içerir. Çerçeve türü, gönderilen çerçevenin bir MAC komut çerçevesi olduğunu belirtir ve diğer kontrol bitleri, çerçevenin nasıl işleneceğini belirler.

- Sıra Numarası (8 bit): Sıra numarası alanı, çerçevenin sıra numarasını içerir. Sıra numarası, gönderilen çerçevelerin izlenmesini ve doğru sıralamayla işlenmesini sağlar.
- Hedef PAN ID (16 bit): Hedef ağ kimliğini belirtir. Bu alan, çerçevenin yönlendirileceği hedef ağın kimliğini içerir ve çerçevenin doğru ağa ulaşmasını sağlar.
- Hedef Adresi (16 bit veya 64 bit): Hedef düğümün adresini içerir. Bu alan, çerçevenin gönderileceği belirli cihazın adresini belirler. Adres uzunluğu, ağ yapısına ve kullanılan adresleme moduna bağlı olarak 16 bit veya 64 bit olabilir.
- Kaynak PAN ID (16 bit): Kaynak ağ kimliğini belirtir. Bu alan, çerçevenin gönderildiği ağın kimliğini içerir ve alıcı cihazın göndericiyi tanımasını sağlar.
- Kaynak Adresi (16 bit veya 64 bit): Kaynak düğümün adresini içerir. Bu alan, çerçevenin gönderildiği belirli cihazın adresini belirler. Adres uzunluğu, ağ yapısına ve kullanılan adresleme moduna bağlı olarak 16 bit veya 64 bit olabilir.
- Komut Tipi (8 bit): Gönderilen komutun türünü belirtir. Örneğin, "Bağlanma İsteği" (Association Request) veya "Bağlantı Kesme Bildirimi" (Disassociation Notification) gibi farklı komut türlerini içerir.
- Komut Veri Yüğü (değişken): Komut ile ilgili ek bilgileri içerir ve uzunluğu değişkendir. Bu alan, gönderilen komutun gerçekleştirilmesi için gerekli ek verileri barındırır.

IEEE 802.15.4 standardı, veri iletimi sırasında güvenliği sağlamak amacıyla erişim kontrolü, mesaj bütünlüğü ve mesaj gizliliği gibi çeşitli güvenlik özelliklerini içerir. Erişim Kontrolü, ağdaki cihazların yetkisiz erişimlerini engellemeyi amaçlar. Bu, Erişim Kontrol Listesi (Access Control List - ACL) kullanılarak gerçekleştirilir. ACL, belirli cihazların iletişim kurabilmesi için önceden tanımlanmış izinleri içerir. Bu liste, yalnızca yetkili cihazların ağa erişimini sağlayarak güvenliği artırır. Mesaj Bütünlüğü gönderilen verinin alıcıya bozulmadan ve değiştirilmeden ulaşmasını garanti eder. IEEE 802.15.4 standardı, mesaj bütünlüğünü sağlamak için çeşitli kriptografik yöntemler kullanır. Bu yöntemlerden biri de AES-CBC-MAC (Cipher Block Chaining Message Authentication Code) yöntemidir. Bu yöntem, veri bütünlüğünü korumak için veri blokları arasında kriptografik bir zincir oluşturur. Mesaj Gizliliği, Mesaj gizliliği, iletilen verinin yetkisiz kişiler tarafından okunmasını engeller. IEEE 802.15.4 standardı, mesaj gizliliğini sağlamak için AES-CCM (Counter with Cipher Block Chaining-Message Authentication Code) yöntemini kullanır. Bu yöntem, hem veri

şifreleme hem de mesaj bütünlüğü sağlar, böylece verinin güvenli bir şekilde iletilmesini temin eder.

IEEE 802.15.4 standardında, farklı güvenlik gereksinimlerini karşılamak için sekiz farklı güvenlik durumu tanımlanmıştır. Bu güvenlik durumları, çeşitli kombinasyonlarda erişim kontrolü, mesaj bütünlüğü ve mesaj gizliliği sağlar. Bu durumlar Tablo 2’de detaylandırılmıştır: 0x00 değeri, herhangi bir şifreleme işlemi gerçekleştirmez; bu nedenle,

Tablo 2. IEEE 802.15.4 tarafından desteklenen güvenlik mekanizmaları (Mesaj Kimlik Doğrulama Kodu - Message Authentication Code)

Kod	Güvenlik Durumu	Açıklama
0x00	NULL	Güvenlik kullanılmaz.
0x01	AES-CTR	Şifreleme için sayıcı modunda AES kullanılır.
0x02	AES-CBC-MAC-32	Mesaj bütünlüğü için 32 bit CBC-MAC kullanılır.
0x03	AES-CBC-MAC-64	Mesaj bütünlüğü için 64 bit CBC-MAC kullanılır.
0x04	AES-CBC-MAC-128	Mesaj bütünlüğü için 128 bit CBC-MAC kullanılır.
0x05	AES-CCM-32	Hem şifreleme hem de mesaj bütünlüğü sağlanır, CBC-MAC için 32 bit kullanılır.
0x06	AES-CCM-64	Hem şifreleme hem de mesaj bütünlüğü sağlanır, CBC-MAC için 64 bit kullanılır.
0x07	AES-CCM-128	Hem şifreleme hem de mesaj bütünlüğü sağlanır, CBC-MAC için 128 bit kullanılır.

veriler şifrelenmez (veri gizliliği sağlanmaz) ve veri orijinalliği doğrulanmaz. 0x01’den 0x03’e kadar olan değerler, verilerin kimliğini doğrulamak için şifrelenmiş Mesaj Kimlik Doğrulama Kodu (MAC) kullanır. 0x04 değeri, veri gizliliğini sağlamak amacıyla veriyi şifreler. 0x05 ila 0x07 aralığındaki değerler ise hem veri gizliliğini hem de veri orijinalliğini sağlamaktadır. Bu değerler, güvenlik seviyelerini belirlemek ve verilerin hem gizliliğini hem de bütünlüğünü korumak için kritik öneme sahiptir.



Şekil 15. ACL giriş formatı

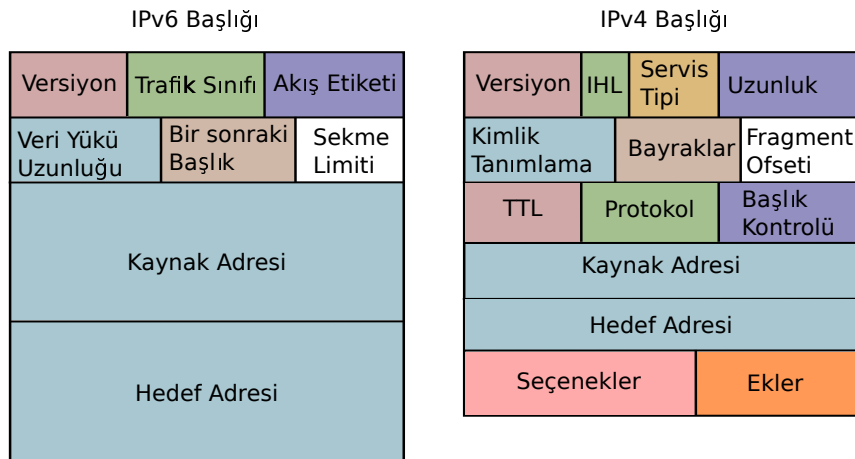
Erişim Kontrol Listesi, IEEE 802.15.4 standardında ağ güvenliğini sağlamak için kullanılan kritik bir mekanizmadır. Kontrol Listesi, belirli cihazların ağa erişim izni olup olmadığını belirler ve yalnızca yetkili cihazların iletişim kurmasını sağlar. Şekil 15’te gösterilen ACL giriş formatı, her bir cihaz için gerekli erişim haklarını ve güvenlik seviyelerini tanımlar. Bu mekanizma, ağda bulunan her cihazın kimlik bilgilerini ve izinlerini içeren bir liste tutarak çalışır. ACL, ağ yönetimi ve güvenlik politikaları doğrultusunda dinamik olarak güncellenebilir. Bu sayede, yetkisiz erişimler engellenir ve ağ güvenliği artırılır.

ACL'nin önemli özelliklerinden biri, güvenli veri iletimi ve ağ kaynaklarının yetkisiz erişimlere karşı korunmasıdır. ACL, cihazların kimlik doğrulamasını gerçekleştirir ve sadece tanımlı cihazların ağa erişimine izin verir. Böylece, potansiyel güvenlik tehditleri minimize edilir ve ağın bütünlüğü korunur. ACL ayrıca ağ performansını iyileştirir ve ağ trafiğinin daha verimli yönetilmesini sağlar.

ACL'nin sağladığı güvenlik seviyeleri, özellikle endüstriyel ve kritik uygulamalarda büyük önem taşır. Örneğin, sağlık izleme sistemleri, akıllı şehirler ve endüstriyel kontrol sistemleri gibi uygulamalarda, veri güvenliği ve ağ bütünlüğü kritik öneme sahiptir. Bu tür uygulamalarda ACL, veri güvenliğini sağlamak ve operasyonel sürekliliği korumak için önemli bir araçtır.

2.2. Internet Protokolü Versiyon 6

Internet Protokolü Sürüm 6 (IPv6), ağlar arasında veri iletimi sağlayan bir ağ protokolüdür. Ancak, verilerin teslimatını garanti etmez, doğru sırada iletilmesini sağlamaz veya yinelenen teslimatı önlemez. Bu tür işlevler, UDP ve TCP gibi üst katman protokolleri tarafından gerçekleştirilir. IPv6, IPv4'ün yerini almış olup, adres boyutu, başlık formatları ve adres ataması gibi alanlarda yenilikler sunar. Ayrıca, IPv6'da ağ güvenliği tasarımın içine entegre edilmiştir. IPv6 iki ana başlık türü kullanmaktadır: Ana IPv6 Başlığı ve yeni IPv6 Uzatma Başlıkları. Ana IPv6 başlığı, daha iyi verimlilik için getirilen bazı alan farklılıkları ile IPv4 başlığına eşdeğerdir. Şekil 16, bu iki başlık türünü karşılaştırmaktadır. Versiyon alanı, IP protokol sürümünü belirten 4 bitlik bir tanımlayıcıdır; IPv4'te bu alan 4, IPv6'da ise 6 olarak ayarlanmıştır.

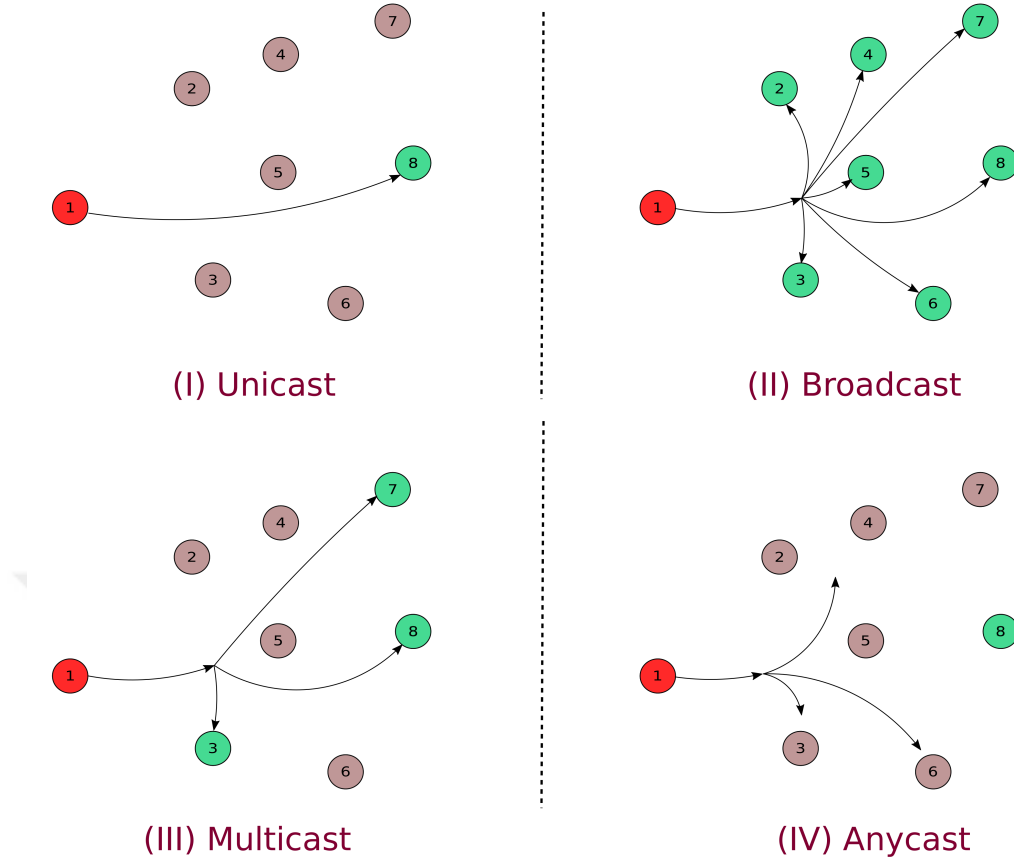


Şekil 16. IPv4 ve IPv6 başlıklarını karşılaştırma

Trafik Sınıfı alanı, paket sınıfının veya önceliğinin 8 bitlik uzunluğundaki tanımlayıcısıdır. Bu alan, IPv4 başlığındaki Hizmet Türü (Type of Service) alanı ile aynı konseptte sahiptir. Trafik Sınıfı alanının ilk 6 biti, RFC 2474'te tanımlanan Farklı Hizmet Kod Noktası (Differentiated Services Code Point, DSCP) alanını temsil eder (Baker vd., 1998). Bu alan, paketlerin ağdaki öncelik seviyelerini belirler ve ağ kaynaklarının verimli bir şekilde kullanılmasını sağlar. Trafik Sınıfının alanının son 2 biti ise, RFC 3168'de tanımlanan Tıkanıklık Bildirimi (Explicit Congestion Notification, ECN) için kullanılır (Floyd vd., 2001). ECN, ağdaki tıkanıklığı bildirmek için kullanılan bir mekanizmadır. Bu mekanizma, veri paketlerinin yönlendirilmesi sırasında tıkanıklığın önlenmesi ve ağ performansının artırılması amacıyla tasarlanmıştır. ECN, paketlerin düşürülmesi yerine, ağdaki tıkanıklığı bildirerek göndericiye ve alıcıya tıkanıklığın varlığını bildirir. Böylece, gönderici veri gönderme hızını ayarlayarak ağdaki tıkanıklığı azaltabilir.

IPv6, IPv4'e göre çok daha büyük bir adres alanına sahiptir: IPv6 adresleri 128 bit uzunluğundadır, IPv4 adresleri ise 32 bittir. Ayrıca, sistemlerin yapılandırma ve yönetim süresini azaltmayı hedefler. IPv6 sistemleri, ağ yönlendiricisi tarafından sağlanan bir ön eki kendi MAC adresiyle birleştirerek benzersiz IP adreslerini otomatik olarak oluşturabilen durumsuz otomatik yapılandırma (stateless autoconfiguration) sürecine katılabilir. Bu nedenle, yapılandırmada adres ataması için DHCP (Dynamic Host Configuration Protocol)'ye ihtiyaç duyulmazken, IPv4'te bu bir temel ihtiyaçtır. IPv6 başlıkları sabit boyutludur ve IPv4 başlıklarına göre daha düzenlidir. Bu da daha iyi performans ve daha fazla genişleme olanağı sağlar. Çok noktaya yayın (Multicast) IPv6'nın zorunlu bir özelliğidir ve IPv4'te desteklenmesine rağmen, her yönlendirici ve ana bilgisayar bunu desteklemez. Ayrıca, Komşu Keşfi (Neighbor Discovery) IPv6'nın bir parçasıdır. Bu özellik, IPv4'te kullanılan Adres Çözümleme Protokolü (Address Resolution Protocol - ARP), ICMP Yönlendirici Keşfi ve ICMP Yönlendirme mesajlarının yerini alan bir dizi ICMPv6 mesajını belirler.

Şekil 17'de, tekli yayın (Unicast), genel yayın (Broadcast), çoklu yayın (Multicast) ve yakın yayın (Anycast) olmak üzere dört farklı adres türü gösterilmektedir. Unicast adresler, belirli bir alıcıya veri göndermek için kullanılır. Bu adresler, kapsamlarına göre küresel (global), site-yerel (site-local) ve yerel bağlantı (link-local) olarak sınıflandırılabilir. Broadcast adresler, aynı anda ağdaki tüm cihazlara veri göndermek için kullanılır. Multicast adresler, bir cihazın bir grup alıcıya veri göndermesini sağlar. Bu alıcılar bir multicast grubuna ait olmalıdır ve grup adresine gönderilen paket, bu gruba ait tüm düğümlere teslim edilir. IPv6'da multicast adresler 0xFF ile başlar ve Flags ve Scope ID alanlarını içerir. Flags, multicast adresinin doğasını belirtirken, Scope ID adresin geçerlilik alanını tanımlar. IPv6 multicast



Şekil 17. IPv6 adresleme türleri: (I) Unicast, (II) Broadcast, (III) Multicast, (IV) Anycast

adresleri, iyi bilinen (IANA tarafından atanmış) veya geçici olabilir ve çeşitli kapsamlar altında kullanılabilir. Anycast adresler ise bir grup alıcıyı tanımlar ve bu adrese gönderilen paket, gruba ait en yakın alıcıya yönlendirilir. Anycast, bir grup içinde en verimli iletişimi sağlamak amacıyla kullanılır. Unicast adresler belirli bir alıcıya, multicast adresler bir grup alıcıya veri gönderirken, anycast adresler bir grup içindeki en yakın alıcıya veri gönderir. IPv4'te yaygın olarak kullanılmayan anycast adresler, IPv6'da daha yaygın ve standart hale getirilmiştir.

Tablo 3. IPv6 adres türleri

Adres Türü	Binary Ön Ek	Ipv6 Notasyonu
Belirtilmemiş (Unspecified)	00...00 128 bit	::/128
Döngüsel (Loopback)	00...01 128 bit	::1/128
Çoklu Gönderim (Multicast)	11111111 bit	FF00::/8
Bağlantı Yerel Tekli Gönderim (Link-local Unicast)	1111111010 bit	FE80::/10
Küresel Tekli Gönderim (Global Unicast)	Her şey (Everything else)	Boş (Empty)

Tablo 3, IPv6 adres türlerinin çeşitli binary ön ekler ve IPv6 gösterimleri ile nasıl tanımlandığını gösterir. Belirtilmemiş adres (Unspecified), belirli bir adresi belirtmez ve genellikle başlangıç adresi olarak kullanılır. Döngüsel adres (Loopback), cihazın kendi kendine veri göndermesini sağlar ve bu adres ::1/128 olarak tanımlanmıştır. Çoklu gönderim adresleri (Multicast), belirli bir grup alıcıya veri gönderimini sağlar ve FF00::/8 ile tanımlanır. Bağlantı yerel tekli gönderim adresleri (Link-local Unicast), sadece aynı ağ bağlantısı üzerindeki cihazlarla iletişim kurmak için kullanılır ve FE80::/10 ile tanımlanmıştır.

IPv6, IPv4'e göre daha geniş bir adres alanı sunar ve bu sayede daha fazla cihazın internete bağlanmasını mümkün kılar. Ayrıca, IPv6, daha verimli başlık formatları ve otomatik adres yapılandırma gibi özellikler sunarak ağ yönetimini kolaylaştırır. Bunun yanında, IPv6'nın multicast ve anycast adresleme desteği, ağ verimliliğini ve performansını artırır.

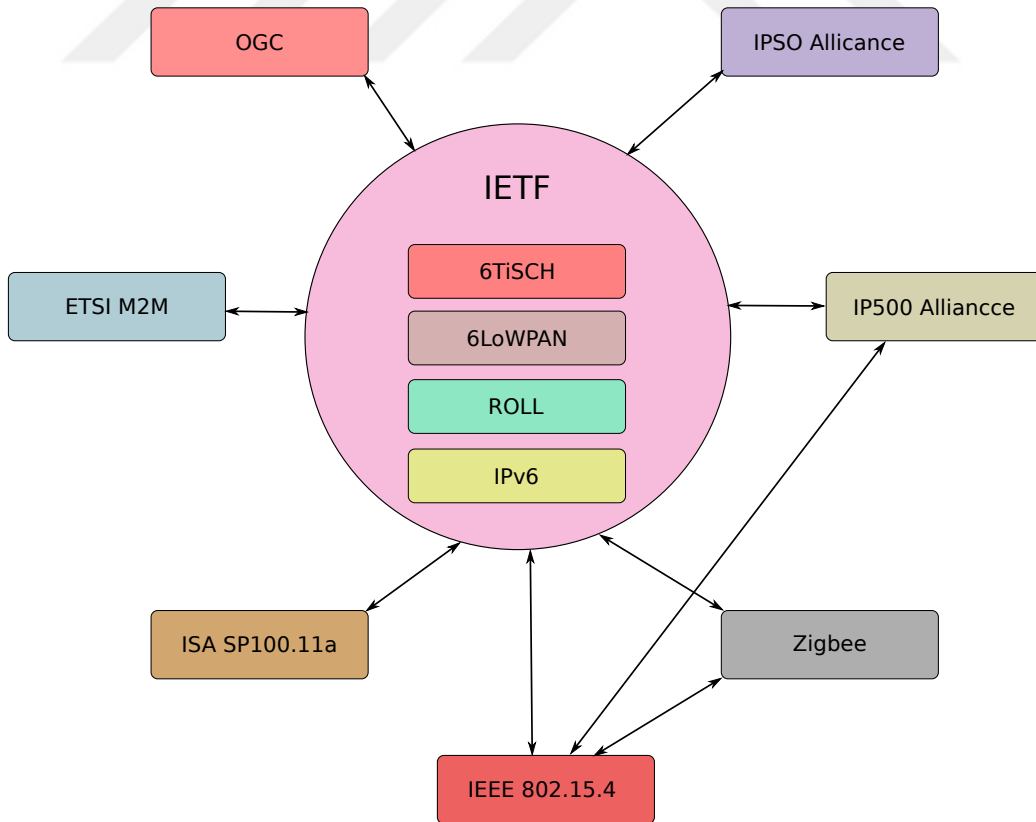
IPv4 ve IPv6, internet protokolleri arasında önemli farklar içermektedir ve bu farklar ağ performansını, güvenilirliğini ve ölçeklenebilirliğini doğrudan etkilemektedir. Tablo 4'te görüldüğü üzere, IPv6, 128 bitlik adres uzunluğuna sahip olup 3.4×10^{38} adres sayısı sunar, bu da IPv4'ün 4.3 milyar adreslik kapasitesine kıyasla çok daha geniş bir adres alanı sağlamaktadır. Bu geniş adres alanı, IoT cihazlarının sayısının hızla artmasıyla birlikte, gelecekteki ağ gereksinimlerini karşılamada kritik bir rol oynar (Kevin, 2009). Ayrıca, IPv6'nın adresleme yöntemi Unicast, Anycast ve Multicast'i içerirken, IPv4'te Broadcast de bulunmaktadır. IPv6'da Broadcast adresleme yöntemi bulunmamaktadır; bunun yerine Multicast kullanılır. IPv6'nın otomatik yapılandırma özelliği, DHCP gereksinimini ortadan kaldırarak sistemlerin kendi benzersiz IP adreslerini otomatik olarak oluşturmaya olanak tanır (Baker vd., 1998; Floyd vd., 2001).

Tablo 4. IPv4 ve IPv6'nın karşılaştırılması

Özellik	IPv4	IPv6
Adres Uzunluğu	32 bit	128 bit
Adres Sayısı	4.3 milyar	3.4×10^{38} milyar
Başlık Boyutu	Değişken (20-60 bayt)	Sabit (40 bayt)
Adresleme Yöntemi	Unicast, Broadcast, Multicast	Unicast, Anycast, Multicast
Otomatik Yapılandırma	Sınırlı (DHCP gerekli)	Evet (otomatik yapılandırma)
Komşu Keşfi	ARP kullanır	ICMPv6 kullanır

2.3. 6LoWPAN

6LoWPAN, düşük güçlü ve sınırlı kaynaklara sahip kablosuz cihazların IPv6 protokolü üzerinden iletişim kurmasını sağlayan bir protokol yığınıdır (Ashrif vd., 2023b; Ko vd., 2011; Le vd., 2016). Temel amacı, küçük, düşük maliyetli cihazların geniş ölçekli ağlara, özellikle de İnternet'e, sorunsuzca entegre olmasını sağlamaktır. Protokol, IEEE 802.15.4 standardı üzerine inşa edilmiştir ve bu sayede kaynak kısıtlı cihazların IP tabanlı ağlarla iletişim kurmasına olanak tanır (Raza vd., 2014). IPv6, genellikle 802.15.4'ün işleyebileceğinden daha büyük paket boyutlarını destekler. Bu yüzden, 6LoWPAN'da bir adaptasyon katmanı tanımlanmıştır. Bu katman, IPv6 başlıklarını sıkıştırır ve IP katmanının altında parçalama ve yeniden birleştirme işlemlerini destekler. Bu özellikler, düşük güçlü cihazların daha verimli bir şekilde çalışmasını sağlar. Bu protokol, özellikle Nesnelerin İnterneti uygulamalarında kullanılır. Bununla birlikte, düşük enerji tüketimi, sınırlı bellek ve işlem kapasitesine sahip cihazların geniş alan ağlarıyla uyumlu hale gelmesini sağlar. 6LoWPAN, akıllı şehirler, ev otomasyonu, sağlık izleme sistemleri ve endüstriyel otomasyon gibi birçok alanda yaygın olarak kullanılmaktadır.

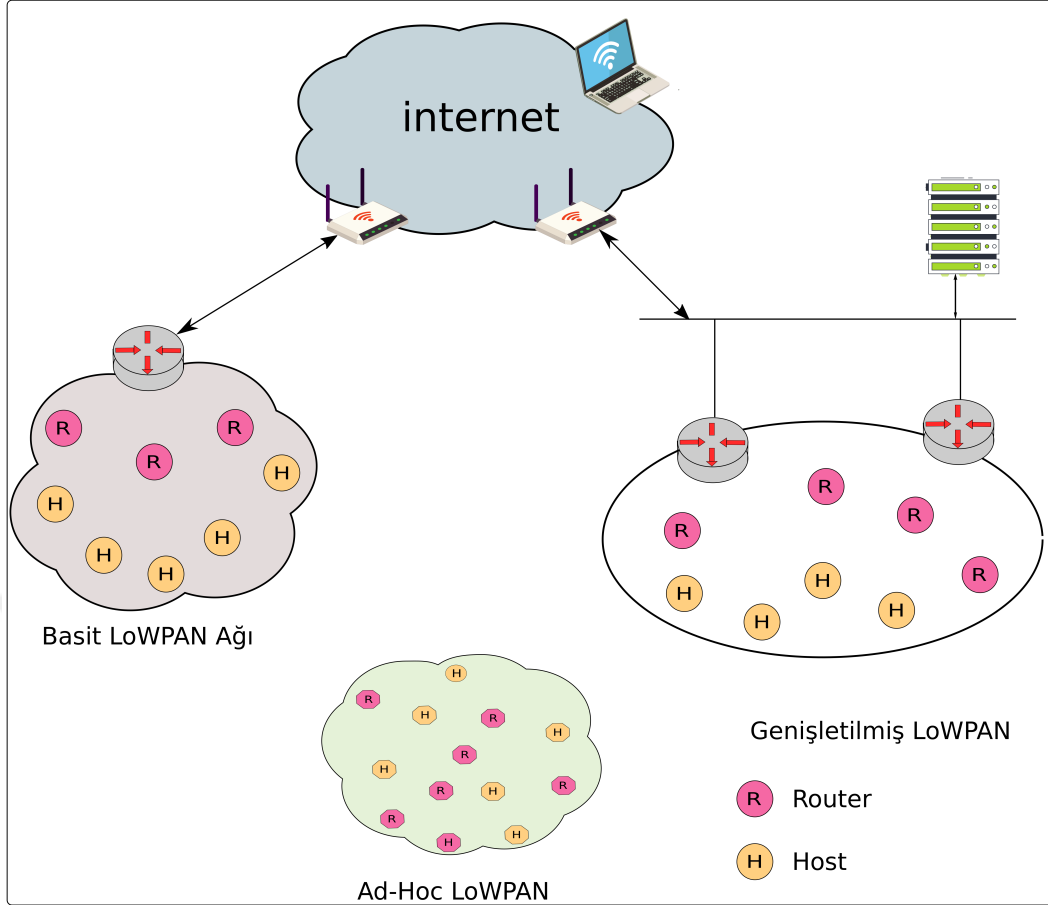


Şekil 18. 6LoWPAN'ın çeşitli standartlar ile ilişkisi

2008 yılında, Düşük Güçlü ve Kayıplı Ağlar Üzerinde Yönlendirme (Routing Over Low Power and Lossy Networks - ROLL) adlı yeni bir IETF çalışma grubu kurulmuştur (Krishna vd., 2021; Al-Amiedy vd., 2023). Bu grup, düşük güçlü, kablosuz ve güvenilir olmayan ağlar için yönlendirme gereksinimlerini ve çözümlerini belirlemektedir. 2008'de ISA, 6LoWPAN'a dayalı bir kablosuz endüstriyel otomasyon sistemi olan SP100.11a'nın standardizasyonuna başladı. 6LoWPAN ile ilgili son faaliyetler arasında, akıllı nesnelerde IP kullanımını ve Nesnelerin İnterneti iş modellerini teşvik etmek amacıyla 2008'de kurulan Akıllı Nesneler için IP ittifakı (Internet Protocol for Smart Object - IPSO) ve IEEE 802.15.4 alt GHz radyo iletişimleri üzerinden 6LoWPAN için geliştiren IP500 İttifakı bulunmaktadır (Vasseur & Dunkels, 2010; Vasseur vd., 2011; Sarb vd., 2015). Şekil 18 ise bu standart kuruluşları ve ittifakların arasındaki bağlantıları görsel olarak açıklamaktadır. Açık Coğrafi Konsorsiyumu (Open Geospatial Consortium - OGC), coğrafi ve algılama uygulamaları için IP tabanlı çözümler belirtir. 2009 yılında, Avrupa Telekomünikasyon Standartları Enstitüsü (European Telecommunication Standards Institute - ETSI), 6LoWPAN ile uyumlu uçtan uca bir IP mimarisi içeren makineden makineye (Machine to Machine - M2M) standardizasyonu için bir çalışma grubu başlatmıştır.

6LoWPAN çalışma grubu, IP paketlerinin düşük güçlü ağlarda sorunsuz taşınmasını aşağıdaki özelliklerle sağlamaktadır:

- IP ve TCP/UDP/ICMP başlıklarının (header) sıkıştırılması.
- IPv6 paketlerinin parçalanması ve yeniden birleştirilmesi.
- Otomatik ağ yapılandırması için komşu keşfi yöntemleri.
- Örgü ağ adresleme desteği.
- 16 bit (kısa) ve 64 bit (uzun) adreslemeyi destekleme.
- Güvenli veri iletimi için şifreleme ve kimlik doğrulama mekanizmaları.
- Çoklu yol seçimi ve yönlendirme protokolleri ile ağ dayanıklılığını artırma.
- Düşük bant genişliği ve düşük gecikme süreleri için protokol iyileştirmeleri.
- Farklı cihazlar arasında uyumluluk sağlamak için standartlaştırılmış protokol desteği.



Şekil 19. Farklı 6LoWPAN mimarileri

Genel 6LoWPAN mimarisi şekil 19’da gösterilmektedir. Üç farklı LoWPAN türü tanımlanmıştır: Basit LoWPAN’lar, Genişletilmiş LoWPAN’lar ve Ad hoc LoWPAN’lar. LoWPAN, ortak bir IPv6 adres önekini (bir IPv6 adresinin ilk 64 bit) paylaşan 6LoWPAN Dğümleri topluluğudur. Bir Ad hoc LoWPAN, internete bağlı değildir ve herhangi bir merkezi altyapı olmadan çalışır. Bu, dğümelerin kendi aralarında doğrudan iletişim kurarak veri alışverişi yaptığı anlamına gelir. Basit bir LoWPAN, LoWPAN Edge Router aracılığıyla başka bir IP ağına bağlanır. Genişletilmiş bir LoWPAN, birden fazla kenar yönlendiricinin LoWPAN’larını ve bunları birbirine bağlayan bir omurga bağlantısını (örneğin Ethernet) kapsar. Kenar yönlendirici, LoWPAN için 6LoWPAN sıkıştırma ve komşu keşfi işlemlerini gerçekleştirirken LoWPAN’a giren ve çıkan trafiği yönlendirdiği için ağ yönetiminde önemli bir rol oynar (Vandervelden vd., 2024).

2.3.1. Adres Modları

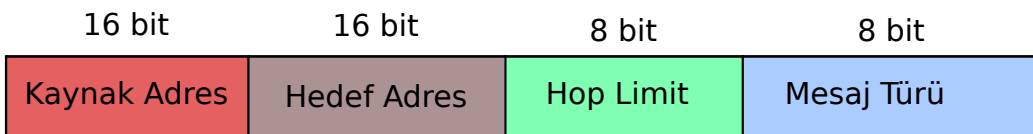
IEEE 802.15.4, PAN içinde benzersiz 16-bit kısa adresler veya IEEE 64-bit genişletilmiş adres kullanımı gibi çeşitli adresleme modlarını destekler. Kısa adresler, bir PAN koordinatörü

ile ilişkilendirildikten sonra kullanılabilir. Bu standart her iki adresleme modunu da destekler ve bir PAN'ın belirli bir IPv6 bağlantısına eşlendiği varsayılır. Ancak, IPv6 doğal olarak multicast'i desteklerken IEEE 802.15.4'te multicast, link-layer broadcast kullanılarak gerçekleştirilir. Bu, multicast mesajlarının sadece belirli bir grubun alıcıları olmasını sağlar. Aşağıda, multicast işleminin nasıl yapıldığı detaylandırılmıştır:

1. Çerçeve başlığındaki hedef PAN ID'si, ağın PAN ID'si ile eşleşmelidir.
2. Çerçeve başlığındaki kısa hedef adresi, broadcast adresi olan 0xFFFF ile eşleşmelidir.
3. Bu mesajı alan düğümler, mesajın kendileri için olup olmadığını anlamak için daha yüksek bir seviyede işlem yapar.
4. 0xxxxxxxxxxxxxxxxx
İlk bit (bit 0), 16-bit adresin bir unicast adres olduğunu belirtmek için sıfır olmalıdır. Bu, gerçek adres için 15 bit bırakır.
5. 100xxxxxxxxxxxxxxxxx
Bitler 0, 1 ve 2, multicast adresler için yukarıdaki desene göre ayarlanır. Kalan 13 bit, gerçek multicast adresini içerir.

0xFFFF değeri, broadcast'i ifade eder ve tüm düğümler tarafından kabul edilmelidir. Ayrıca, 6LoWPAN ağlarında, 16-bit adresler 5 farklı sınıfa ayrılmıştır. Şu anda sadece iki sınıf tanımlanmıştır, diğerleri gelecekte kullanılmak üzere ayrılmıştır.

Diğer önemli bir özellik ise 6LoWPAN Mesh Adresleme Başlığıdır. Düşük güçlü ve kayıplı ağlarda IPv6 paketlerinin çoklu atlama (multi-hop) üzerinden iletilmesini sağlamak için kullanılan önemli bir mekanizmadır. Mesh adresleme, paketlerin kaynak düğümden hedef düğüme ara düğümler üzerinden güvenli ve verimli bir şekilde ulaşmasını sağlar. Mesh adresleme başlığı, 6LoWPAN ağlarında paketlerin yönlendirilmesi ve iletilmesi için gerekli olan adresleme ve yönlendirme bilgilerini içerir. Mesh adresleme başlığı, şekil 20'de gösterilen bileşenlerden oluşur.



Şekil 20. 6LoWPAN Mesh adres başlığı

Kaynak adresi, paketi gönderen düğümün 16-bitlik kısa adresini içerir. Bu adres, ağdaki her düğüm için benzersizdir ve paketin başlangıç noktasını belirtir. Hedef Adresi, paketin

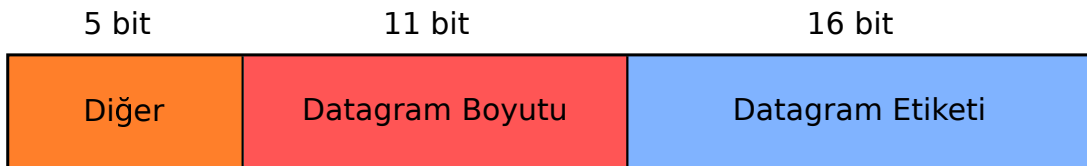
ulaşması gereken son düğümün 16-bitlik kısa adresini içerir. Bu adres, paketlerin hedefini belirler ve nihai varış noktasını gösterir. Hop Limiti, paketin kaç kez ara düğüm üzerinden iletebileceğini belirleyen bir sayaçtır. Mesaj Türü, paketlerin türünü belirten bir alan içerir. Bu, paketin veri, kontrol mesajı veya başka bir türde olduğunu belirlemek için kullanılır.

Mesh adresleme başlığı, ağdaki düğümler arasında esnek veri iletimi sağlar. Paketler, ağdaki en uygun yolu takip ederek hedeflerine ulaşabilir. Çoklu atlama mekanizması, paketlerin ara düğümler üzerinden güvenli bir şekilde iletilmesini sağlar. Bu, ağın genel güvenilirliğini artırır. Mesh adresleme, ağın boyutunun büyümesine ve yeni düğümlerin eklenmesine olanak tanır. Bu sayede, geniş ölçekli ve karmaşık ağlar oluşturulabilir. Mesh ağlar, enerji tüketimini optimize ederek düğümlerin daha uzun süre çalışmasını sağlar. Ara düğümler, enerjiyi verimli kullanarak paketlerin iletimini gerçekleştirir.

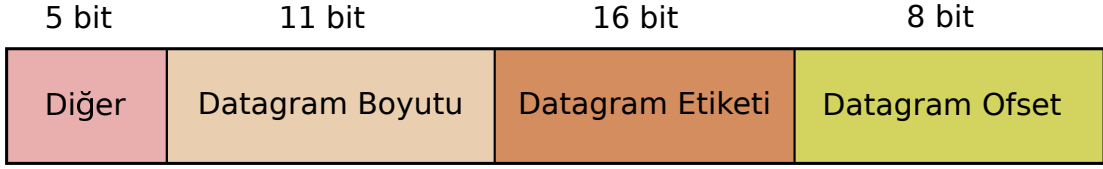
2.3.2. Parçalama

6LoWPAN adaptasyon katmanının temel işlevlerinden biri, veri parçalama ve yeniden birleştirme süreçleridir. Parçalama başlığı, veri yükünün IEEE 802.15.4 çerçevesine sığamayacak kadar büyük olduğu durumlarda kullanılır. Bu başlık üç ana bileşenden oluşur: datagram boyutu, datagram etiketi ve datagram ofseti.

Datagram boyutu, parçalanmamış veri yükünün toplam boyutunu belirtir ve alıcı tarafında bellek tahsisini sağlamak için her parçaya eklenir. Datagram etiketi, belirli bir veri yüküne ait parçaları tanımlamak ve eşleştirmek için kullanılır (Tanaka vd., 2019). Ofset alanı ise yalnızca ikinci ve sonraki parçalar için kullanılır; bu alan, veri yükünün başlangıcından itibaren parçanın sekizli katları halinde ofsetini belirtir. Şekil 21'deki IPv6 paketinin 6LoWPAN ağında iletmek üzere parçalandığında gönderilen ilk parça, datagram boyutu (11 bit) ve datagram etiketi (16 bit) içeren bir başlık taşır. Şekil 22'deki sonraki parçalar ise, datagram boyutu, datagram etiketi ve ofset (8 bit) içeren bir başlık taşır (Khan & Salah, 2018). Bu başlıklar, parçalanmış veri yüklerinin doğru bir şekilde yeniden birleştirilmesini sağlar (Tanaka vd., 2019; Ashrif vd., 2023b).



Şekil 21. IPv6 paketinin 6LoWPAN ağında iletmek üzere parçalandığında gönderilen ilk parçası

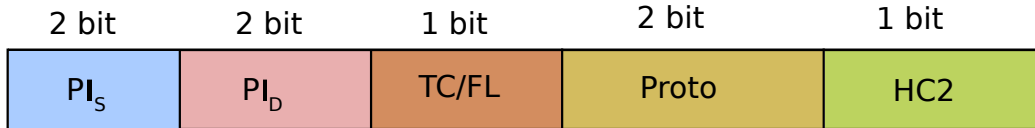


Şekil 22. IPv6 paketinin geri kalanını içeren parçalar

6LoWPAN, IP katmanından aldığı ve parçalama eşiğini aşan büyük paketleri önce küçük parçalara böler. Protokol, parçalama, iletim ve yeniden birleştirme işlemlerini 60 saniye içinde gerçekleştirerek iki düğüm arasındaki veri aktarımını sağlamalıdır. Paket başlığı, ilk parçaya yerleştirilir ve hedef düğüme iletmek üzere ortam erişim katmanına gönderilir. Diğer parçalar ise, parça başlığı ve sıra numarası bilgisi ile hedef düğüme yönlendirilir. Hedef düğüm, aldığı parçaları sıralar ve bu paketi IP katmanına iletir. IP katmanında parçalama ve birleştirme işlemleri yapılmaz. Eğer haberleşme sırasında herhangi bir parça kaybolursa, IP paketinin tamamı kaybolmuş sayılır.

2.3.3. Başlık Sıkıştırma

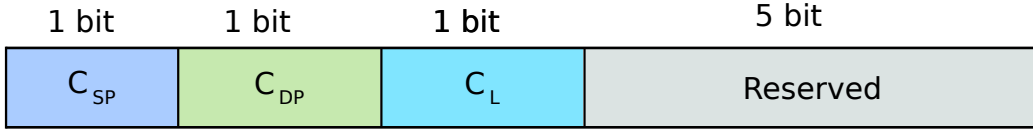
6LoWPAN, IPv6 paketlerinin IEEE 802.15.4 çerçevelerine sığması için başlık sıkıştırma teknikleri kullanır (Kushalnagar vd., 2007). Sıkıştırma, paketlerin daha az yer kaplamasını ve daha verimli bir şekilde iletilmesini sağlar. IPv6 başlıkları genellikle 40 bayt uzunluğundadır, bu da düşük bant genişliğine sahip ağlarda verimli bir şekilde iletilmeleri zor olabilir. IEEE 802.15.4 çerçeveleri, yalnızca 127 baytlık bir maksimum çerçeve boyutuna sahiptir, bu nedenle başlık sıkıştırma, veri iletimini optimize etmek için kritik bir öneme sahiptir (Montenegro vd., 2007). RFC4944, standart olarak iki farklı sıkıştırma seviyesi sunar: Başlık sıkıştırma 1 (header compression 1 - HC1) ve Başlık sıkıştırma 2 (header compression 2 - HC2).



Şekil 23. 6LoWPAN HC1 yapısı

HC1, IPv6 başlıklarının sıkıştırılmasını sağlayan ilk sıkıştırma şemasıdır. HC1, IPv6 başlığındaki sabit alanları ve genellikle aynı kalan alanları sıkıştırır (Kushalnagar vd., 2007). HC1, aşağıdaki alanları sıkıştırır:

- PI_S (Prefix Indicator Source): 2 bit uzunluğundadır ve kaynak adres ön ek göstergesini belirtir. Kaynak adresin sıkıştırma durumunu ifade eder.



Şekil 24. 6LoWPAN HC2 yapısı

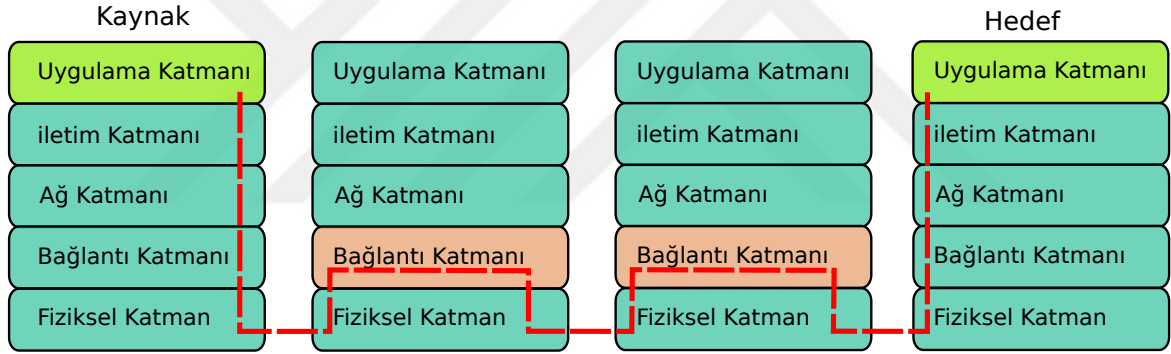
- PI_D (Prefix Indicator Destination): 2 bit uzunluğundadır ve hedef adres ön ek göstergesini belirtir. Hedef adresin sıkıştırma durumunu ifade eder.
- TC/FL (Traffic Class / Flow Label): 1 bit uzunluğundadır. Trafik sınıfı ve akış etiketi genellikle sıkıştırılır veya düşük güçlü ağlarda kullanılmaz, bu nedenle sıkıştırma yapılabilir.
- Proto (Next Header Protocol): 2 bit uzunluğundadır. Bir sonraki başlık protokolünü belirtir ve genellikle belirli protokoller için sıkıştırma yapılır.
- HC2 (Header Compression 2): 1 bit uzunluğundadır. HC2'nin kullanılıp kullanılmadığını belirtir. HC2 alanı, daha ileri sıkıştırma işlemleri için kullanılır.

Şekil 23'te, 6LoWPAN HC1 paket formatı gösterilmektedir. HC2, HC1 sıkıştırmasının ardından UDP başlıklarının sıkıştırılmasını sağlar. UDP başlıkları da genellikle 8 bayt uzunluğundadır. Source Port ve Destination Portlar, eğer bu alanlar belirli bir aralıkta ise (örneğin, sık kullanılan port numaraları), sıkıştırılarak daha az bit ile temsil edilir. Length ve Checksum, bu alanlar sıkıştırılır ve yalnızca gerektiğinde iletilir. Şekil 24'te, 6LoWPAN HC2 için detaylı paket içeriği gösterilmektedir.

C_{SP} (Compressed Source Port), kaynak port numarasının sıkıştırılmış halini ifade eder. Bu alan, belirli aralıklarda sık kullanılan port numaralarını daha az bit ile temsil eder. Örneğin, sık kullanılan port numaraları (1024-49151 aralığındaki portlar) sıkıştırılarak daha az bit ile gösterilebilir. C_{DP} (Compressed Destination Port), hedef port numarasının sıkıştırılmış halini ifade eder. Benzer şekilde, hedef port numarası da belirli aralıklarda sık kullanılan port numaralarını daha az bit ile temsil eder. Bu, ağ trafiğinde sık kullanılan hizmetlerin port numaralarını daha verimli bir şekilde iletilmesini sağlar. C_L (Compressed Length), UDP başlığındaki uzunluk alanının sıkıştırılmış halini ifade eder. UDP başlığı genellikle 8 bayt uzunluğundadır, ancak sıkıştırma ile bu uzunluk daha az bit kullanılarak gösterilir. Bu alan, yalnızca gerektiğinde iletilir. Reserved, gelecekte kullanım için ayrılmıştır ve şu anki implementasyonlarda kullanılmaz. Sıkıştırma şemalarında esneklik sağlamak ve ileride eklenebilecek yeni özelliklere yer açmak için ayrılmıştır. HC2 başlığı, 6LoWPAN ağlarında başlık sıkıştırma teknikleri kullanılarak paketlerin daha küçük hale getirilmesine ve ağ trafiğinin optimize edilmesine yardımcı olur.

2.3.4. Yönlendirme

Yönlendirme, veri paketinin bir cihazdan diğerine, bazen birden fazla ara düğüm üzerinden iletilmesi yeteneği olarak tanımlanabilir (Olsson, 2014). Yönlendirme mekanizmasının hangi katmanda gerçekleştirildiğine bağlı olarak, bağlantı katmanı üzerinden (mesh-under) ve ağ katmanı üzerinden (route-over) olmak üzere iki kategoriye ayrılır. Şekil 25'te, Mesh-under yönteminin veri paketlerini iletmek için bağlantı katmanı adreslerinin (IEEE 802.15.4 MAC veya kısa adres) kullanılmasını, Şekil 26'da ise Route-over yönteminin ağ katmanı adreslerini (IP adresleri) kullanılmasını gösterilmektedir. Contiki OS, genellikle Route-over (ağ katmanı üzerinden) yönlendirme mekanizmasını kullanır. Bu yöntem, IP tabanlı yönlendirme sağladığı için geniş ağlar ve internete bağlı cihazlar için daha uygundur. Contiki OS'de bu yönlendirme mekanizmasını sağlayan ana bileşenlerden biri RPL protokolüdür.

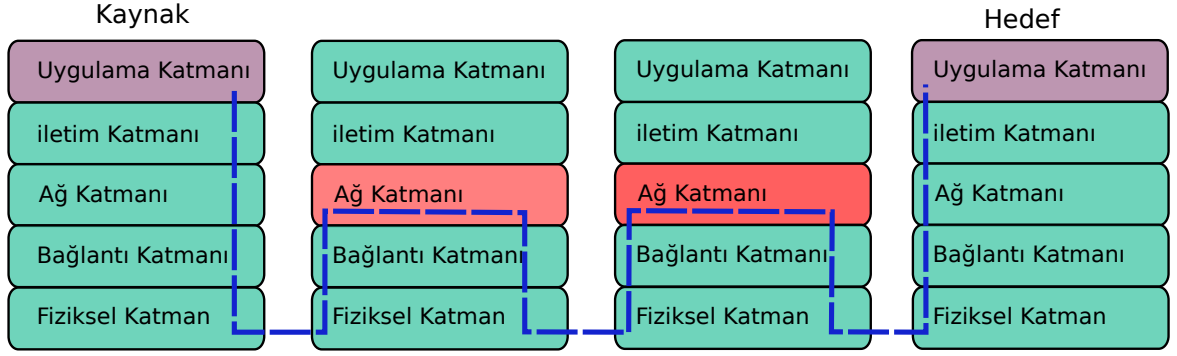


Şekil 25. Mesh-under paket yönlendirme

Mesh-under yönlendirme, veri paketlerinin bağlantı katmanı adreslerini kullanarak iletilmesini sağlar. Bu yöntemde, her ara düğüm, paketi bir sonraki düğüme ileterek hedefine ulaşmasını sağlar. Mesh-under yönlendirme, bağlantı katmanında gerçekleştirildiği için düşük gecikme ve düşük güç tüketimi avantajları sunar. Bu yöntem, özellikle düşük güçlü ve kayıplı ağlarda yaygın olarak kullanılır.

Route-over yönlendirme, veri paketlerinin ağ katmanı adreslerini (IP adresleri) kullanarak iletilmesini sağlar. Bu yöntemde, yönlendirme kararları ağ katmanında alınır ve her ara düğüm, paketin hedef IP adresine ulaşması için en uygun yolu belirler. Route-over yönlendirme, daha esnek ve ölçeklenebilir bir çözüm sunar, ancak daha fazla işlem gücü ve enerji tüketimi gerektirebilir.

Mesh-under ve Route-over yönlendirme yöntemlerinin avantajları ve dezavantajları aşağıda özetlenmiştir:



Şekil 26. Route-over paket yönlendirme

• Mesh-under Yönlendirmenin Avantajları:

- Düşük Gecikme: Mesh-under yönlendirme, veri paketlerinin doğrudan bağlantı katmanında yönlendirilmesini sağlar. Bu, paketlerin daha az işlem sürecinden geçmesini ve böylece daha düşük gecikme sürelerine ulaşmasını sağlar.
- Düşük Enerji Tüketimi: Bu yöntem, düğümlerin bağlantı katmanında doğrudan iletişim kurmasını sağladığı için, enerji açısından verimlidir. Özellikle enerji sınırlı cihazlar için ideal bir çözümdür.
- Basitlik: Mesh-under yönlendirme, yönlendirme kararlarının bağlantı katmanında alınması nedeniyle daha basit bir yapıdadır. Bu, ağın kurulumu ve yönetimini kolaylaştırır.
- Hızlı Veri İletimi: Düğümden düğüme hızlı veri iletimi sağlar, bu da gerçek zamanlı uygulamalar için avantajlıdır.

• Mesh-under Yönlendirmenin Dezavantajları:

- Sınırlı Ölçeklenebilirlik: Mesh-under yönlendirme, düğüm sayısının artmasıyla birlikte ölçeklenebilirlik sorunları yaşayabilir. Ağ büyüdükçe, her düğümün diğer düğümler hakkında bilgi sahibi olması gerektiğinden, yönetim ve performans zorlukları ortaya çıkar.
- Düğüm Bilgisi Gerekliği: Bu yöntemde, her düğümün ağdaki diğer düğümler hakkında bilgi sahibi olması gerekmektedir. Bu, ağ yönetimi karmaşıklığını arttırabilir ve sürekli bilgi güncellemeleri gerektirebilir.
- Kısıtlı Yönlendirme Seçenekleri: Bağlantı katmanı üzerinden yönlendirme, ağ katmanı üzerinden yönlendirmeye göre daha kısıtlı seçenekler sunabilir. Ağ katmanında yönlendirme, ağ topolojisi ve ağ durumu değiştikçe daha esnek ve

dinamik yönlendirme kararları alabilir. Bağlantı katmanında yapılan yönlendirme ise genellikle sabit yollar kullanır ve bu da değişen ağ koşullarına uyum sağlamayı zorlaştırır.

- Route-over Yönlendirmenin Avantajları:

- Yüksek Ölçeklenebilirlik: Route-over yönlendirme, ağın büyük ölçekte genişlemesine olanak tanır. IP tabanlı yönlendirme kullanılarak, geniş alan ağlarında bile etkili bir şekilde çalışabilir.
- Esneklik: Bu yöntem, farklı ağ topolojileri ve uygulama senaryolarına uyum sağlayabilir. IP tabanlı yönlendirme, çeşitli protokollerle entegrasyonu kolaylaştırır.
- IP Tabanlı Yönlendirme Avantajları: IP adresleme ve yönlendirme kullanılarak, mevcut internet altyapısı ile uyumlu çalışır. Bu, geniş alan ağlarında ve internete bağlı cihazlarda büyük bir avantaj sağlar.
- Gelişmiş Yönlendirme Kararları: Route-over yöntemi, daha gelişmiş yönlendirme algoritmalarının kullanılmasına olanak tanır. Bu, veri iletimi için en optimal yolun seçilmesini sağlar.

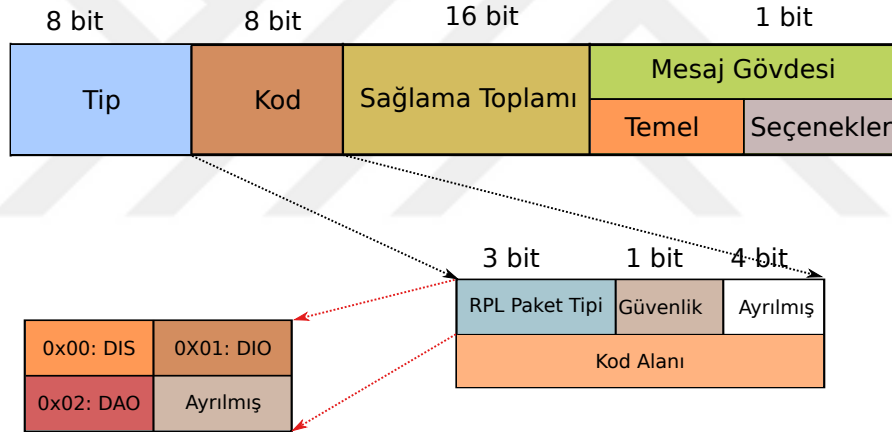
- Route-over Yönlendirmenin Dezavantajları:

- Daha Yüksek Gecikme: Route-over yönlendirme, ağ katmanında yönlendirme kararları aldığı için, paketlerin daha fazla işlemden geçmesine neden olabilir. Bu da gecikme sürelerini artırabilir.
- Daha Fazla İşlem Gücü ve Enerji Tüketimi: Bu yöntemde, yönlendirme kararlarının alınması için daha fazla işlem gücü ve enerji tüketimi gereklidir. Bu, özellikle enerji sınırlı cihazlar için dezavantaj olabilir.
- Karmaşıklık: Route-over yöntemi, daha karmaşık bir yapıya sahip olabilir ve ağ yönetimi ve yapılandırması daha zor olabilir. Bu, ağın kurulumu ve bakım maliyetlerini artırabilir.

6LoWPAN, tek bir IPv6 alt ağı olarak çalışır ve bu nedenle geleneksel ön-ek tabanlı yönlendirme yöntemleri burada etkili olmaz. Ara yönlendiriciler, değişen kanal koşulları, düğüm hareketliliği veya düğümlerin uyku moduna geçmesi gibi nedenlerle bağlantı kayıpları yaşayabilir. Bu yüzden, 6LoWPAN ağlarında yönlendirme protokolleri, bu değişken koşullara

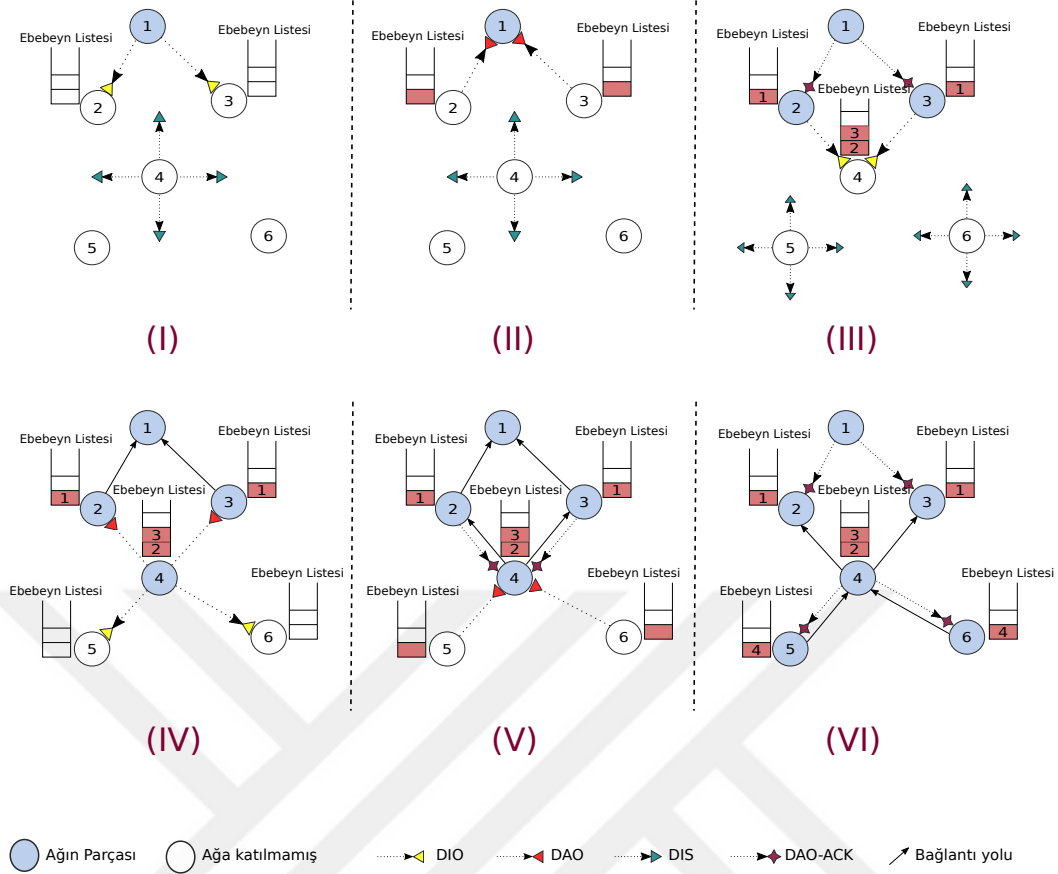
uyum sağlayacak şekilde optimize edilmelidir. Özellikle, cihazların sınırlı kaynaklara sahip olması, yönlendirme protokollerinin düşük güç tüketimi, düşük gecikme ve yüksek güvenilirlik sağlamasını gerektirir. Geleneksel yönlendirme yaklaşımları kaynak sınırlı ağlar için uygun olmayabilir ve verimli ve güvenilir veri iletimi sağlamak için 6LoWPAN ekosistemine göre uyarlanmış özel protokoller gereklidir.

Özetle, 6LoWPAN ağının dinamik ve kaynak sınırlı özellikleri, tutarlı veri iletimini sürdürmek için enerji verimliliği, düşük gecikme süresi ve sağlam bağlantıya odaklanarak bu ortamda sunulan benzersiz zorlukları ele almak için özel olarak tasarlanmış yönlendirme protokollerinin kullanılmasını gerektirir. Bu bağlamda, düşük güçlü ve kayıplı ağlarda yaygın olarak RPL yönlendirme protokolü kullanılır. RPL, özellikle IoT uygulamaları için optimize edilmiş bir protokoldür ve ağdaki düğümler arasında veri paketlerinin verimli bir şekilde yönlendirilmesini sağlar. Bu protokol, 6LoWPAN ağlarının ihtiyaçlarına yönelik olarak geliştirilmiştir ve çeşitli avantajlar sunar.



Şekil 27. RPL kontrol mesajı

Özellikle sensör ağları, akıllı şehirler ve IIoT uygulamaları gibi geniş ve heterojen ağlar için uygundur. RPL, yönlendirme kararlarını almak için DAG (Directed Acyclic Graph) adı verilen bir yapı kullanır. Şekil 27, ICMPv6 kontrol mesajı türü olan RPL mesajlarının yapısını göstermektedir. RPL kontrol mesajı aşağıdaki bileşenlerden oluşur: tip (type), kod (code) ve sağlama toplamı (checksum). İletinin gövdesi bir temel ileti ve birçok seçenekten oluşur. RPL için 155 olarak ayarlanan tip alanı, ICMPv6 kontrol mesajının türünü tanımlar. RPL kontrol mesajının türü kod alanında belirtilir. RPL paket tipi DIO (DODAG Information Object), DAO (Destination Advertisement Object), DIS (DODAG Information Solicitation) ve Ayrılmış (Reserved) olmak üzere dört önemli paket tipine sahiptir ve bunların her biri aşağıda ayrı ayrı açıklanmıştır.



Şekil 28. RPL ile DODAG oluşum aşamaları

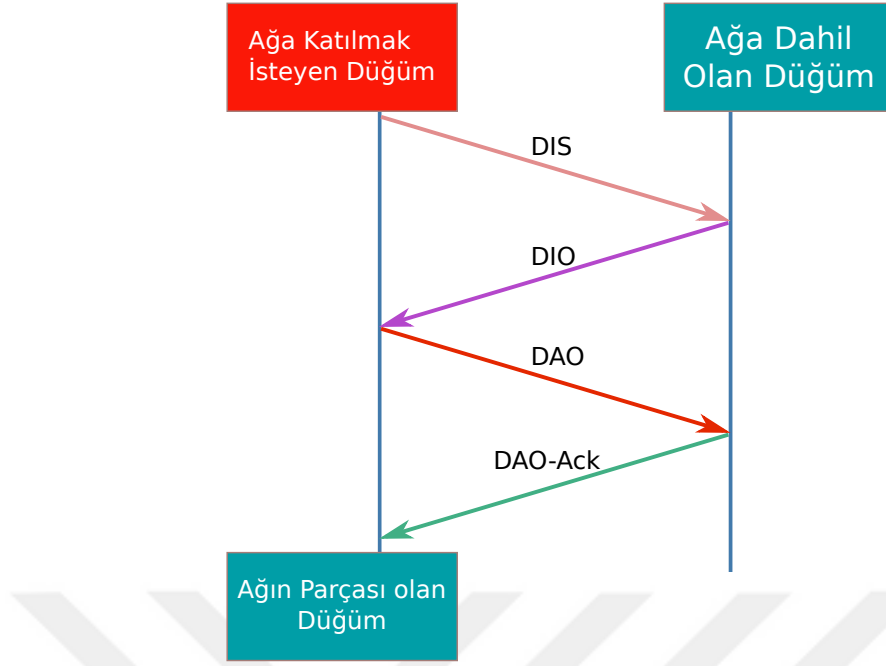
RPL, düğümlerin ağı dahil olması ve yönlendirme işlemlerini gerçekleştirmesi için şekil 28'deki adımları izler. İlk olarak, kök düğüm bir DIO mesajı yayar. Düğümler, DAG yapısına dahil olmak için çeşitli mesajlar gönderir ve alır. Bu süreç, düğümlerin ağda uygun yerlerini belirlemelerini ve ebeveyn listesi oluşturmalarını sağlar. DIO mesajları, DAG yapı bilgilerini yaymak için kullanılır. Bir DAG kökü (root), DIO mesajları göndererek diğer düğümleri DAG'a katılmaya davet eder. Bu mesajlar, düğümlerin DAG hakkında bilgi sahibi olmasını sağlar. DIO paketini alan düğüm, DIO göndericisini yönlendirme tablosundaki ebeveyn listesine ekler ve alınan DIO'da belirtilen amaç fonksiyonuna (Objective Function - OF) göre rank (konum bilgisi) değerini hesaplar. Rank değeri, düğümün ağdaki konumunu yansıtır ve ebeveyn rankından büyük olmak zorundadır. OF, düğümlerin yönlendirme kararlarını alırken kullandıkları kriterleri belirler. Bu fonksiyonlar, düğümlerin rank hesaplamalarını ve ebeveyn seçimlerini etkileyerek ağın performansını, enerji verimliliğini ve güvenilirliğini optimize eder. Contiki'de OF, ContikiRPL modülünde tanımlanmıştır ve kullanıcılar, ağın ihtiyaçlarına göre farklı amaç fonksiyonları tanımlayabilir. Yaygın olarak kullanılan OF'ler arasında Histeresisli Minimum Sıralama Amaç Fonksiyonu (Minimum Rank with Hysteresis Objective

Function - MRHOF) ve OF0 (Objective Function Zero) bulunmaktadır. MRHOF, düğümlerin en düşük rank değerine sahip yolları seçmesini sağlayarak ağın genel enerji verimliliğini artırır. MRHOF, rank hesaplamalarında histerezis kullanarak, yönlendirme kararlarının daha kararlı olmasını ve gereksiz yönlendirme değişikliklerinin önlenmesini sağlar. Bu sayede, ağdaki enerji tüketimi dengelenir ve düğümlerin pil ömrü uzar. MRHOF'un bu enerji verimliliği odaklı yapısı, özellikle enerji sınırlı cihazların bulunduğu IoT uygulamalarında önemli avantajlar sunar. OF0 ise, düğümlerin en kısa yolu seçmesini sağlayarak gecikmeyi minimize eder. Bu amaç fonksiyonu, veri iletiminin hızlandırılmasına ve ağdaki gecikmelerin azaltılmasına odaklanır. OF0, düğümlerin en düşük rank'a sahip yolları tercih etmelerini sağlar, bu da veri paketlerinin en kısa mesafeden hedefine ulaşmasını temin eder. Bu yaklaşım, özellikle zaman kritik uygulamalarda büyük bir avantaj sağlar. Örneğin, acil durum yanıt sistemlerinde veya gerçek zamanlı veri iletimi gerektiren endüstriyel otomasyon sistemlerinde, OF0 kullanılarak veri iletiminin gecikmesi en aza indirilebilir ve sistemin hızlı tepki vermesi sağlanabilir.

Seçilen amaç fonksiyonlarına göre DIO mesajları daha sonra güncellenir ve komşu düğümlere gönderilir. Düğüm, ebeveyn listesinden tercih edilen bir ebeveyni seçer ve bu, verileri DODAG kök düğüme göndermek için varsayılan geçit olarak kullanılır. DAO mesajları, sensör düğümlerinin DODAG yapısına katılması ve ağ içinde belirli bir hedef yönüne doğru yer alması için kullanılır. Sensör düğümleri, DAO mesajlarını kullanarak kendi konumlarını ve rotalarını ağa bildirir. Bu sayede, ağ içinde iletişim ve veri aktarımı gerçekleşebilir. DAO-ACK mesajları ise, DAO mesajlarına verilen yanıtlardır. Bir sensör düğümünün gönderdiği DAO mesajına, DODAG'ın kök düğümü veya ağ yöneticisi tarafından DAO-ACK mesajı ile onay gönderilir. Bu süreç, sensör düğümlerinin ağa başarılı bir şekilde katılmasını ve ağ bağlantısının kurulmasını sağlar. DAO ve DAO-ACK mesajları, RPL protokolünün temel unsurlarıdır. Özellikle düşük güçlü ve kayıplı ağlarda, kısıtlı kaynakları en verimli şekilde kullanmak için bu mesaj türleri kritik öneme sahiptir. Ağ yapısının oluşturulması, optimal rotalama ve güç tüketiminin azaltılması gibi işlevler, DAO ve DAO-ACK mesajları aracılığıyla gerçekleştirilir.

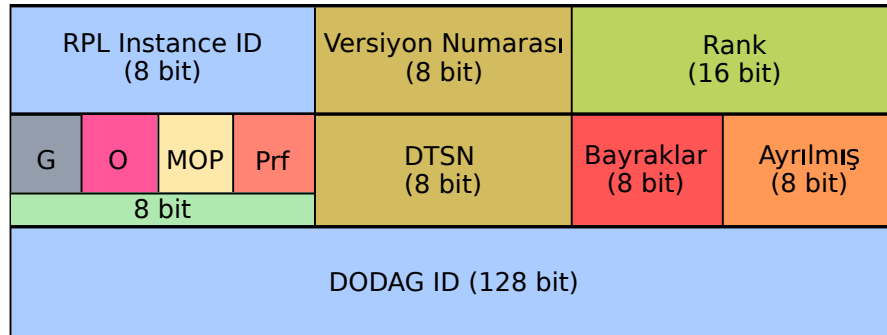
DIS mesajları, RPL protokolünde düğümlerin ağdaki DODAG yapısı hakkında bilgi toplamak için kullandıkları önemli bir mesaj türüdür (Şekil 29). DIS mesajlarının temel işlevi şu şekildedir:

1. Yeni bir düğüm, mevcut DODAG yapısına katılmak istediğinde, DIS mesajı göndererek ağdaki diğer düğümlerden DODAG hakkında bilgi toplar.



Şekil 29. RPL protokolünün kontrol mesajlarının akış diyagramı

2. Mevcut bir düğüm, ağdaki DODAG yapısında herhangi bir değişiklik olup olmadığını kontrol etmek için DIS mesajları gönderebilir.
3. DODAG kök düğümü, ağdaki tüm düğümlerin DODAG bilgilerini güncellemesi için DIS mesajları yayınlayabilir.



Şekil 30. RPL DIO paketi

DIS mesajları içerisinde, DODAG kök düğümünün IP adresi, DODAG Kimliği, düğümün DODAG'a katılma tercihini belirten bir değer ve düğümün DODAG'a katılma şekli, zaman aşımı değerleri vb. diğer parametreler yer alır. Bu mesaj formatı, RPL ağındaki dinamik değişikliklere hızlı bir şekilde uyum sağlamayı mümkün kılar. Şekil 30'da gösterilen DIO mesajları, ağla ilgili kontrol mesajlarının iletim sıklığını en üst düzeye çıkarmak için

belirlenen zamana bağılı olarak Trickle algoritması ile aralıklı olarak iletilir (Fawwaz & Chung, 2023). Trickle algoritması, kontrol mesajlarının iletimini belirli bir zaman dilimi içinde düzenler. Bu zaman dilimi, minimum bir değerden (I_{min}) maksimum bir değere (I_{max}) kadar değişir. Belirli aralıklarla, algoritma kontrol mesajlarını yayınlar. Bu mesajlar, ağdaki düğümlerin en güncel yönlendirme bilgilerine sahip olmasını sağlar. Algoritma, ağdaki bir değişiklik tespit edildiğinde zamanlayıcıyı sıfırlar ve kontrol mesajlarının iletim sıklığını artırır. Bu, ağın hızla güncellenmesini sağlar. Değişiklikler ortadan kalktığında ve ağ kararlı hale geldiğinde, zamanlayıcının süresi yavaşça artırılarak, mesaj gönderme sıklığı azaltılır.

Trickle algoritmasının çalışma mantığı, belirli bir zaman diliminde mesaj sayısını minimize etmek ve ağın kararlılığını korumaktır. Algoritmanın temel formülleri şu şekildedir:

$$I = I_{min} \cdot 2^k \quad (2.1)$$

Burada, I periyot uzunluğunu, I_{min} minimum periyot uzunluğunu, ve k güncelleme sayısını temsil eder. Periyot süresi, her güncellemede iki katına çıkarak ağın güncel duruma hızlı adapte olmasını sağlar.

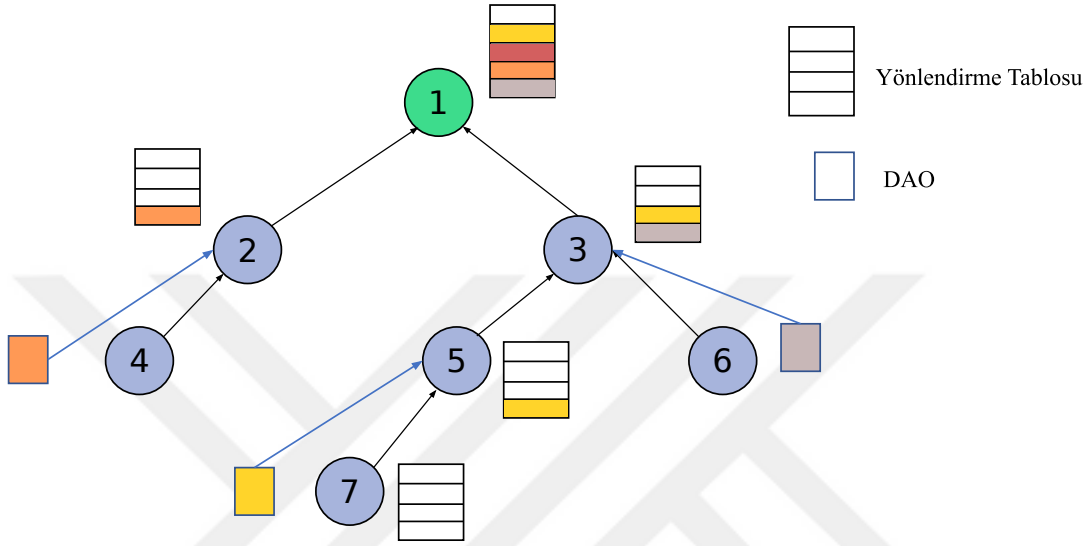
$$T = I/2 + \text{rand}(0, I/2) \quad (2.2)$$

Bu formülde, T rastgele seçilen zaman dilimini, $[0, \frac{I}{2}]$ aralığında değişen uniform dağılımlı bir rastgele değişkeni temsil eder. Bu sayede, mesajların iletimi rasgele aralıklarla gerçekleştirilir, bu da çakışmaların ve gereksiz tekrarların önlenmesine yardımcı olur.

Örneğin, bir düğümün eklenmesi veya bir bağlantının kesilmesi gibi durumlarda, ağın yapısını ve yönlendirme bilgilerini güncellemek için daha sık DIO mesajları gönderilir. Bu, düğümlerin en güncel yönlendirme bilgilerini edinmelerine olanak tanır ve böylece ağın kararlılığını ve verimliliğini korur. Ağ kararlı hale geldiğinde, yani düğümlerin yerleri ve bağlantıları sabit olduğunda ve ağın genel durumu değişmediğinde, DIO mesajlarının gönderim sıklığı azaltılır. Bu azaltma, ağdaki gereksiz veri trafiğini ve enerji tüketimini minimize eder. Böylece, ağın genel performansı ve enerji verimliliği artırılır. DIO mesajlarının gönderim sıklığının bu şekilde dinamik olarak ayarlanması, ağın hem değişikliklere hızlı tepki vermesini hem de stabil durumlarda gereksiz kaynak tüketimini önlemesini sağlar.

Trickle algoritması, gereksiz kontrol mesajlarının iletimini önleyerek enerji tüketimini optimize eder, bu da özellikle düşük güçlü ve kaynak sınırlı cihazlar için önemli bir avantajdır.

(Shetty vd., 2024). Algoritma, ağdaki değişikliklere dinamik olarak tepki verir ve ağın hızlı bir şekilde güncellenmesini ve kararlı hale gelmesini sağlar. Ancak, kararsız ağlarda algoritmanın sürekli olarak mesaj iletim sıklığını ayarlamak zorunda kalması enerji tüketimini artırabilir. Ayrıca, yanlış zamanlama ayarları gereksiz mesaj iletimine veya ağın yavaş güncellenmesine neden olabilir.

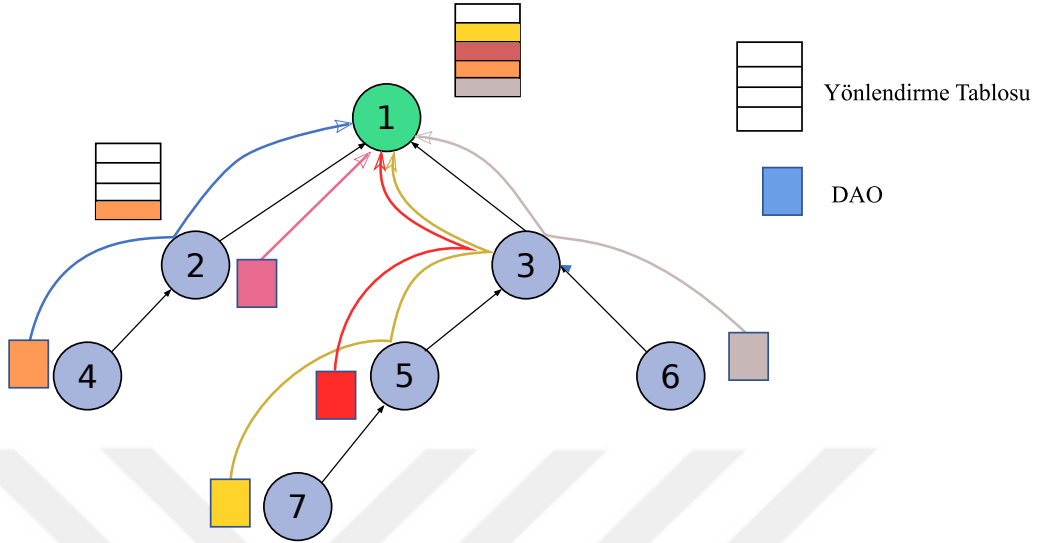


Şekil 31. RPL Depolama (Storing) modu

DODAG yapısındaki yönlendirici düğümler, DIO kontrol paketlerinde tanımlanan hizmet türüne göre düğümlerin yönlendirme tablolarını yönetebilir. RPL örneğinde aşağıya doğru yolları korumak için, RPL yönlendirme protokolü Storing Mode (Depolama Modu) ve Non-Storing Mode (Depolama Olmayan Mod) olmak üzere iki çalışma moduna sahiptir. Şekil 31 depolama modunu, Şekil 32 ise depolama olmayan modu göstermektedir.

Storing Mode (Depolama Modu), her düğümün yönlendirme tablolarını kendi yerel belleğinde sakladığı bir moddur. Bu mod, düğümlerin birbirlerine doğrudan veri paketleri iletmelerini sağlar. Storing Mode'un detaylarına bakıldığında, her düğümün, kendisiyle doğrudan iletişimde olan düğümlerin yönlendirme bilgilerini tuttuğu bir yönlendirme tablosuna sahip olduğu görülür. Bu tablolar, ağdaki her düğümün hangi yollardan veri ileteceğini belirler. Düğümler, DAO mesajları kullanarak yönlendirme bilgilerini DAG köküne iletirler. Bu mesajlar, her düğümün yönlendirme bilgilerini içerir ve kök düğüme ulaştırır. Veri iletimi, doğrudan hedef düğüme yapılır, bu da hızlı ve verimli veri iletimi sağlar. Storing Mode'un avantajları arasında düşük gecikme süreleri ve yüksek performans yer alır. Yönlendirme kararları yerel olarak alındığı için ağın genel verimliliği artar. Ancak,

yönlendirme tablolarının tutulması için daha fazla bellek kullanımı gerektirir, bu da kaynak sınırlı cihazlar için sorun yaratabilir.



Şekil 32. RPL Depolama olmayan (Non-Storing) modu

Non-Storing Mode (Depolama Olmayan Mod), yönlendirme bilgilerini yalnızca DAG kökünde tutan bir moddur. Bu modda, düğümler yönlendirme tablolarını saklamaz ve tüm yönlendirme işlemleri DAG kökü üzerinden gerçekleştirilir. Non-Storing Mode'un detaylarına bakıldığında, yönlendirme bilgilerinin sadece DAG kökünde tutulduğu ve düğümlerin yönlendirme tablolarını depolamadığı görülür. Düğümler, yönlendirme bilgilerini DAG köküne iletmek için DAO mesajları kullanır ve bu mesajlar kökte depolanır. Veri paketleri DAG köküne yönlendirilir ve kök düğüm, paketi hedef düğüme iletecek en uygun yolu belirler. Non-Storing Mode'un avantajları arasında düğümlerin daha az bellek kullanması yer alır, bu da kaynak sınırlı cihazlar için avantaj sağlar. Ancak, veri paketlerinin kök düğüm üzerinden iletilmesi daha yüksek gecikmelere ve potansiyel olarak daha fazla enerji tüketimine neden olabilir.

RPL, ağdaki yönlendirme hatalarını düzeltmek ve ağın kararlılığını sağlamak için iki onarım mekanizması sunar: Yerel onarım (Local Repair) ve Küresel onarım (Global Repair). Local Repair (Yerel Onarım), ağdaki küçük ölçekli yönlendirme hatalarını düzeltmek için kullanılan bir mekanizmadır. Yerel onarımın amacı, düğümün belirli bir bölümünde oluşan sorunları hızlı bir şekilde düzeltmektir. Bir düğüm, bağlantı sorunları veya yönlendirme hataları tespit ettiğinde, yeni bir DIO mesajı yayarak yerel onarım sürecini başlatır. Bu mesaj, ağdaki diğer düğümlere yönlendirme bilgilerini güncellemeleri gerektiğini bildirir. Yerel onarım hızlı ve düşük maliyetli bir çözüm sunar, ağın genel performansını etkilemeden

sorunları düzeltir. Ancak, sadece küçük ölçekli sorunları çözebilir ve büyük ölçekli ağ hatalarını düzeltmekte yetersiz kalabilir.

Global Repair (Küresel Onarım) ise ağdaki büyük ölçekli yönlendirme hatalarını düzeltmek için kullanılan bir mekanizmadır. Küresel onarımın amacı, ağ genelinde büyük ölçekli sorunları düzeltmek ve ağın tamamını yeniden yapılandırmaktır. DAG kökü, büyük bir yönlendirme hatası tespit ettiğinde, tüm ağda yeni bir DAG oluşturulmasını sağlayarak küresel onarım sürecini başlatır. Bu, tüm düğümlerin yönlendirme bilgilerini ve bağlantılarını güncellemelerini gerektirir. Küresel onarım büyük ölçekli ağ hatalarını etkili bir şekilde düzeltebilir ve ağın genel sağlığını iyileştirir. Ancak, yüksek maliyetli ve zaman alıcı olabilir ve ağın genel performansını geçici olarak etkileyebilir.

2.3.5. Otomatik Yapılandırma ve Komşu Keşfi

Otomatik yapılandırma ve komşu keşfi, IPv6 tabanlı ağların verimli çalışmasını sağlayan kritik mekanizmalardır. Bu mekanizmalar, özellikle dinamik ve geniş ağlarda düğümlerin kendilerini doğru bir şekilde yapılandırmalarını ve çevrelerindeki diğer düğümleri keşfetmelerini sağlar. Otomatik yapılandırma, ağdaki cihazların manuel müdahale gerektirmeden kendi IP adreslerini ve diğer ağ yapılandırmalarını almasını sağlayan bir süreçtir. Bu süreç, çeşitli mesaj tiplerinin kullanılmasıyla etkin bir şekilde gerçekleştirilir. IPv6 ağlarında otomatik yapılandırma ve komşu keşfi süreçlerinde kullanılan temel mesajlar şunlardır:

1. Yönlendirici Duyurusu (Router Advertisement - RA): Yönlendiriciler tarafından ağdaki düğümlere yönlendirme bilgileri ve ağ yapılandırma bilgilerini iletmek için kullanılır. RA mesajları, ağ ön ek bilgilerini, varsayılan yönlendirici bilgilerini ve diğer konfigürasyon bilgilerini içerir.
2. Yönlendirici Talebi (Router Solicitation - RS): Düğümler tarafından ağdaki yönlendiricilerden RA mesajlarını talep etmek için kullanılır. RS mesajları, yönlendiriciye RA mesajı göndermesi için tetikleyici bir istekte bulunur.
3. Komşu Duyurusu (Neighbor Advertisement - NA): Düğümler tarafından diğer düğümlere kendi varlıklarını ve MAC adres bilgilerini bildirmek için kullanılır. NA mesajları, bir düğümün IPv6 adresinin MAC adresiyle eşleştirilmiş halini içerir.

4. Komşu Talebi (Neighbor Solicitation - NS): Düğüm tarafından belirli bir IPv6 adresinin sahibini öğrenmek veya Yineleyici Adres Tespiti (Duplicate Address Detection - DAD) işlemi için kullanılır. NS mesajları, hedef IPv6 adresinin sahibini belirlemek için gönderilir.

Otomatik yapılandırma, ağdaki cihazların manuel müdahale gerektirmeden kendi IP adreslerini ve diğer ağ yapılandırmalarını almasını sağlayan bir süreçtir. IPv6, bu süreci destekleyen birkaç yöntem sunar:

- Durumsuz Adres Otomatik Yapılandırma (Stateless Address Auto Configuration - SLAAC): SLAAC, bir düğümün kendine özgü bir IPv6 adresi oluşturmasını sağlar. Bu süreç, düğümün MAC adresini kullanarak global olarak benzersiz bir IP adresi oluşturmasına olanak tanır. Bir düğüm, ağdaki bir RA mesajını aldığı anda, bu mesajdaki bilgilere dayanarak kendine bir IPv6 adresi atar. RA mesajları, ağda mevcut olan ön ek bilgilerini içerir. SLAAC, düğümün hızlı ve verimli bir şekilde IP adresi almasını sağlar.
- Durumlu Yapılandırma (Stateful Configuration - DHCPv6): DHCPv6, DHCP sunucusundan dinamik olarak IPv6 adresi ve diğer yapılandırma bilgilerini almak için kullanılır. Bir düğüm, DHCPv6 İstemci Kimlik Bilgisi mesajı gönderir ve DHCP sunucusundan Yanıt mesajını alır. Bu mesajlaşma süreci, düğümün uygun bir IPv6 adresi ve diğer ağ bilgilerini almasını sağlar.

Komşu keşfi, IPv6 ağlarında düğümün birbirlerini keşfetmelerini ve iletişim kurmalarını sağlayan bir dizi protokol ve mesajdan oluşur. Bu protokol, ağdaki düğümün varlıklarını, konumlarını ve ulaşılabilirliklerini belirlemelerine yardımcı olur. IPv6 ağlarında komşu keşfi, düğümün hem ağ içinde doğru bir şekilde adreslenmesini hem de etkin bir iletişim kurulmasını sağlar. Bu süreç, özellikle büyük ve dinamik ağlarda düğümün birbirleriyle güvenilir ve verimli bir şekilde iletişim kurabilmesi için kritik öneme sahiptir. Komşu keşfi protokolü, hem yeni düğümün ağa katılması sırasında hem de mevcut düğümün iletişim bilgilerini güncellemeleri sırasında kritik bir rol oynar. Bu süreç, düğümün IP adreslerini doğrulamalarına, yönlendiricilerden bilgi almalarına ve diğer düğümlerle iletişim kurmalarına olanak tanır. Protokol, aynı zamanda ağdaki yönlendirme tablolarının güncel kalmasını ve düğümün en verimli yolları kullanarak veri iletimini gerçekleştirmesini sağlar. Bu süreç birkaç önemli bileşenden oluşur:

Adres Çözümleme (Address Resolution): Komşu keşfi, düğümlerin IP adreslerini, veri bağlantı katmanı adreslerine (örneğin, MAC adresleri) eşleştirmelerini sağlar. Bu, bir düğümün belirli bir IP adresine veri göndermek için doğru fiziksel adresi bulmasını mümkün kılar. Bu işlem, NS ve NA mesajları ile gerçekleştirilir.

Adres Yenileme ve Kontrol (Address Reachability): Düğümler, komşularının ulaşılabilir olup olmadığını belirlemek ve mevcut bağlantılarının durumunu izlemek için periyodik olarak NS mesajları gönderir. NA mesajları, bir düğümün ulaşılabilir olduğunu ve veri alabileceğini doğrular. Bu işlem, düğümlerin ağdaki değişikliklere hızlıca uyum sağlamasına yardımcı olur.

Yönlendirici Keşfi (Router Discovery): Düğümler, ağdaki yönlendiricileri keşfetmek ve onlardan yönlendirme bilgileri almak için RS ve RA mesajlarını kullanır. Yönlendirici keşfi, düğümlerin ağ yapılandırma bilgilerini almasını ve yönlendirme tablolarını güncellemesini sağlar. Bu, yeni bir düğümün ağa katılması sırasında kritik bir adımdır.

Adres Çakışma Tespiti (Duplicate Address Detection – DAD): Komşu keşfi protokolü, bir düğümün kendisine atadığı IP adresinin benzersiz olup olmadığını doğrulamak için DAD işlemini kullanır. Bu işlem, aynı IP adresine sahip başka bir düğümün olup olmadığını belirlemek için NS mesajları gönderir. Eğer bir yanıt alınmazsa, adresin benzersiz olduğu kabul edilir. Bu, ağda IP adresi çakışmalarını önler ve ağın kararlılığını artırır.

Yönlendirme (Routing): Komşu keşfi, yönlendirme protokollerinin ağdaki en verimli yolları belirlemesine yardımcı olur. Bu, düğümlerin veri paketlerini en uygun rotalardan iletmelerini sağlar ve ağın genel performansını optimize eder. Yönlendirme işlemi, düğümlerin sürekli olarak birbirleriyle iletişim kurmasını ve yönlendirme tablolarını güncel tutmasını gerektirir.

2.4. 6TiSCH

IETF 6TiSCH Çalışma Grubu, IEEE802.15.4 TSCH (Time Slotted Channel Hopping) bağlantı katmanı üzerine IP katmanı adaptasyonu ve kontrol düzlemi standardizasyonu üzerinde çalışmak için oluşturulmuştur (Aydın vd., 2018). 6TiSCH, endüstriyel alanda kullanılan düşük güçlü kablosuz ağlara IPv6 teknolojisini getirmek amacıyla yapılan önemli bir protokoldür (Aydın & Görmüş, 2018; Vučinić vd., 2020). IETF tarafından yapılan çalışmalar, düşük kapasiteli ağlarda IPv6'yı kullanmanın getirdiği birçok zorluğu belirlemiş ve bu sorunları çözmeye odaklanmıştır. 6TiSCH, ağ katmanları arasında bir köprü görevi görerek, verimli ve basit bir yapı sunmayı hedeflemiştir. Bununla birlikte, başlık sıkıştırma,

IP içinde IP kapsülleme ve 6LoWPAN Komşu Keşfi gibi konularda önemli iyileştirmeler yapmıştır. Bu iyileştirmeler sayesinde, daha yetenekli yönlendirme şemaları ve güvenlik yönetimi sağlanmış ve ağların daha verimli çalışması mümkün olmuştur.

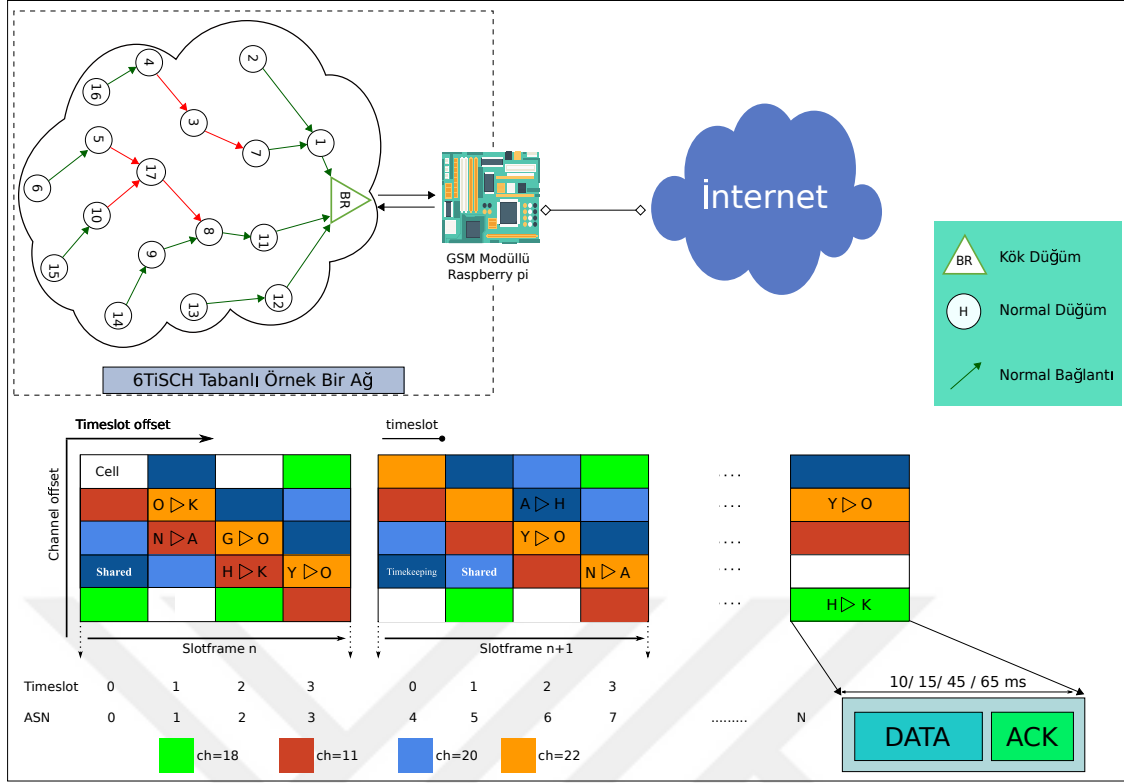
Uygulama (CoAP)	RFC8613 RFC7252	(2019) OSCORE (2014) COAP
Yönlendirme (RPL)	RFC6554 RFC6553 RFC6552 RFC6550	(2012) Yönlendirme Başlığı (2012) Başlık Formatı, RPL v3 (2012) OF Tanımlanması, RPL v2 (2012) RPL v1
Adaptasyon Katmanı (6LoWPAN)	RFC8505 RFC8138 RFC8025 RFC6282 RFC4944	(2018) Komşu Keşfi, Kayıt Olma (2017) Yönlendirme Başlık Sıkıştırma (2016) 6LoWPAN v3 (2011) 6LoWPAN v2 (2007) 6LoWPAN v1
Tarifeleyici (6TiSCH)	RFC9033 RFC8480 RFC8137 RFC 9031 RFC8180	(2018) Dağıtık Tarifeleyici Algoritması v2 (2018) 6P v2, Dağıtık Tarifeleyici Algoritması v1 (2017) 6P v1 (2017) 6TiSCH Güvenlik Eklentisi (2017) 6TiSCH v1
Fiziksel Katman	IEEE802.15.4	(2015) 2.4 GHz, 250 kbps, 127 bayt

Şekil 33. 6TiSCH protokol yığını oluşturulan farklı standartlar

IETF 6TiSCH protokol yığını, IEEE802.15.4 TSCH bağlantı katmanına dayanmaktadır ve bu yığının tüm detayları 6TiSCH Mimarisinde açıklanmaktadır. Şekil 33, standart 6TiSCH yığının oluşturulan ana yapı taşlarını özetlemektedir. Bu yapı taşları, ağın verimli ve güvenli bir şekilde çalışmasını sağlayan çeşitli protokollerden oluşur (Pradhan & Chaudhari, 2022).

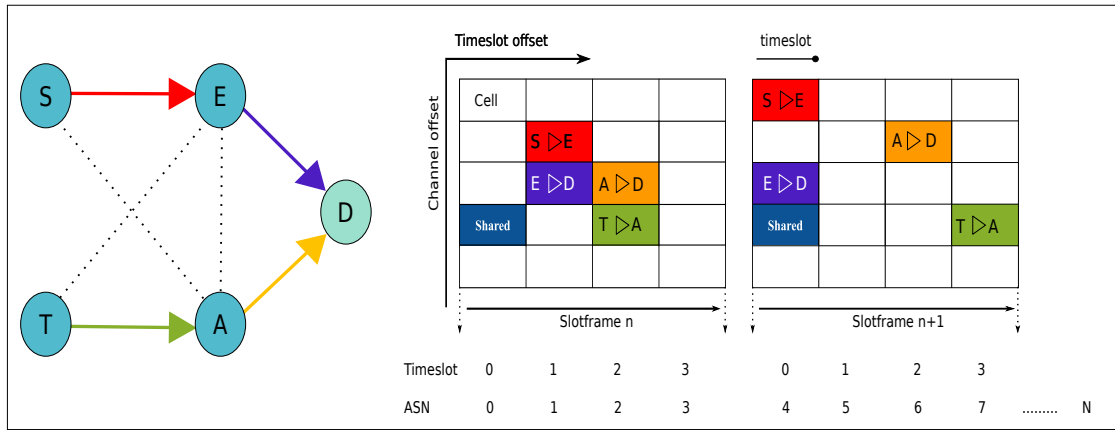
İlk olarak, ağın başlatılması ve çalıştırılması için minimal bir protokol bulunmaktadır. Bu protokol, ağın temel işlevlerini yerine getirmek için gereken en temel özellikleri içerir. Ayrıca, güvenli birleştirme prosedürlerini desteklemek için yalın bir güvenlik mekanizması da mevcuttur. Bu mekanizma, ağdaki cihazların güvenli bir şekilde birbirleriyle iletişim kurmalarını sağlar. Ağ zamanlama yönetimi için, dağıtık bir çizelgeleme protokolünden oluşan bir paket kullanılır. Bu protokol, ağdaki veri iletiminin zamanlamasını ve düzenini belirler. Ayrıca, bu zamanlama yönetim paketine ek olarak, bir dizi çizelgeleme politikası da bulunur. Bu politikalar, ağın performansını optimize etmek için kullanılır.

6TiSCH, bağlantı katmanı topolojisinin yönlendirme topolojisi ile eşleştirildiği bir mekanizma sağlayarak düğümlerin en iyi bağlantılı ebeveynleri ile senkronizasyonu sürdürmelerini sağlar. Şekil 34, bu mekanizmanın nasıl çalıştığını göstermektedir. Bu mekanizma, ağın kararlılığını ve verimliliğini artırır. Güvenlik açısından, CoAP ve Kısıtlı RESTful Ortamlar için Nesne Güvenliği (Object Security for Constrained RESTful



Şekil 34. Çizelgeleme, ASN ve örnek bir 6TiSCH ağı

Environments - OSCORE), güvenli bir birleştirme sürecini desteklemenin yanı sıra yığını tamamlar ve düşük yükte güvenli RESTful etkileşimi sağlayan araçlar sağlar. Bu, ağdaki veri iletiminin güvenli bir şekilde gerçekleştirilmesini sağlar. Sonuç olarak, IETF 6TiSCH, IEEE802.15.4 TSCH'nin endüstriyel performansını IPv6'ya hazır bir üst yığınla birleştirerek iyi ayarlanmış ve güvenli bir komple ağ çözümü oluşturur.



Şekil 35. 6TiSCH çizelgeleme örneği

Bir zaman dilimi çerçevesi (slotframe) içinde kanal erişimini düzenler. Şekil 35'te 4 slottan oluşan örnek bir slot çerçevesi gösterilmektedir. İletimler, bir çift zaman aralığı (zaman alanında yatay olarak) ve kanal ofsetinden (frekans alanında dikey olarak) oluşan

hücrelerde planlanır. Her hücre, zaman ve frekans koordinatlarıyla tanımlanır. Hücrenin zaman ofseti, slotframe'in başlangıcına göre zaman içindeki konumunu gösterir. Her hücre, tek bir düğümde iletim (TX) veya alım (RX) için zamanlanır. Hücrede herhangi bir etkinlik planlanmadığında, düğüm uyku moduna geçer. TX hücresinde, çerçeve hazır olduğunda veya tamponda bir çerçeve beklediğinde iletim gerçekleşir; aksi takdirde hücre optimize edilmiş bir uygulamada uyku moduna geçer ve bir sonraki planlanan zamanı bekler. Bu nedenle, 6TiSCH ağı düşük güç tüketimine sahiptir. Kanal ofseti ise her slotframe döngüsünde belirli bir frekansa denk gelir. Slot çerçevesi sonsuza kadar tekrar eder ve her zaman dilimi Mutlak Slot Numarası (Absolute Slot Number - ASN) adı verilen monoton olarak artan bir tanımlayıcı ile ilişkilendirilir. Bu slot numarası her zaman diliminde artar ve 5 baytlık bir sayaçtır. ASN değeri, 10 ms zaman dilimi varsayıldığında, 300 yıldan fazla süre boyunca ağı kesintisiz çalışmasını destekleyebilir (Accettura vd., 2015).

$$f = F[(ASN + ch_{of}) \mod Nc] \quad (2.3)$$

Denklem 2.3, TSCH protokolünde hangi frekansın kullanılacağını belirlemek için kullanılır. Denklemden, f kullanılacak frekans, F ise frekans listesini temsil eder. ASN mutlak zaman numarasıdır (Hauweele vd., 2020). Kanal ofseti (ch_{of}), belirli bir zaman diliminde kullanılacak frekans belirlemek için kullanılan bir değerdir. Nc , kullanılabilir frekans sayısını ifade eder. Hesaplama sonucunda bir indeks elde edilir ve düğüm frekans listesindeki ilgili indeksli frekans kullanılır.

Bu süreç, her zaman diliminde farklı bir frekansın kullanılmasını sağlar ve böylece TSCH ağlarında, yol sönmemesini ve dış müdahaleyi azaltarak daha güvenilir bir iletişim sunar (Kurunathan vd., 2018). Yol sönmemesi, bir sinyalin gönderici anteninden çıktıktan sonra alıcı antene ulaşana kadar geçtiği ortamda enerjisinin azalması olarak tanımlanır. Sinyalin yayılım sürecinde enerjisini kaybetmesine neden olan çeşitli faktörler bulunmaktadır. Fiziksel engeller, sinyalin enerjisini soğurarak, yansıtarak veya dağıtarak zayıflamasına neden olur. Mesafe arttıkça sinyal gücü azalır, bu serbest uzayda yayılma kaybı olarak bilinir ve sinyal gücünün mesafenin karesi ile ters orantılı olarak azalmasına neden olur. Dış müdahale ise, bir ağdaki iletişimin diğer elektronik cihazlardan veya ağlardan gelen sinyallerle bozulmasıdır. Diğer kablosuz cihazlar, aynı frekans bandını kullanan cihazlar (Wi-Fi, Bluetooth, mikrodalga fırınlar vb.) parazit oluşturabilir. RF gürültüsü, çeşitli elektrikli ve elektronik cihazlar tarafından oluşturulan gürültüdür ve alıcıda algılanan işareti bozarak

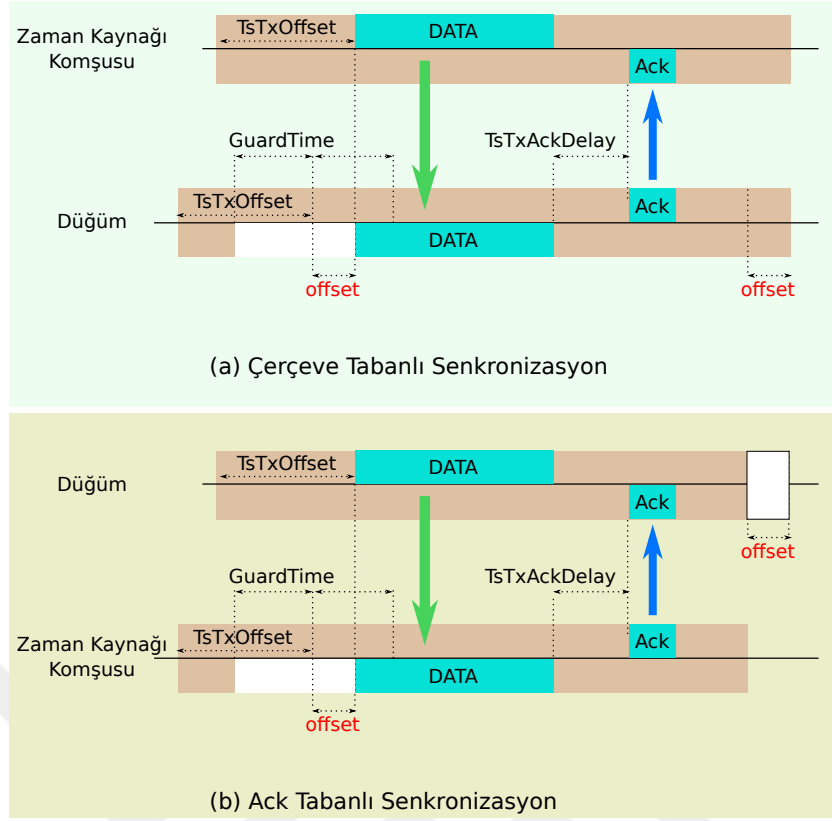
iletişimi etkileyebilir. Ağ yoğunluğu, aynı frekans bandını paylaşan çok sayıda cihazın bulunması durumunda frekansların aşırı yüklenmesine ve sinyal kalitesinin düşmesine neden olabilir. Yan kanal etkisi, bir frekans kanalının hemen yanında bulunan başka bir kanalın sinyali tarafından oluşturulan parazit nedeniyle iletişimi bozabilir. TSCH ağlarında, frekans atlama mekanizması kullanılarak bu sorunlar minimize edilir. Örneğin, 16 kullanılabilir frekans olduğu varsayıldığında ($N_c = 16$) ve frekans listesi $F, [f_1, f_2, \dots, f_{16}]$ şeklindedir. Eğer $ASN = 5$ ve $ch_{of} = 3$ ise, toplam 8 olur ($5 + 3$) ve 8 mod 16 işlemi yine 8 verir. Bu durumda, 8. indeksteki frekans seçilerek düğümler arasındaki haberleşme sağlanmış olur. Bu süreç, her zaman diliminde farklı bir frekansın kullanılmasını sağlayarak TSCH ağlarında daha güvenilir bir iletişim sunar.

TSCH, zaman dilimlerini ve frekansları bir matris gibi yapılandırarak düğümlerin komşularıyla iletişim kurmasını sağlar (Tanaka vd., 2021). Bu yapı, bir çizelgeleme fonksiyonu (Scheduling Function - SF) tarafından yönetilir. Bağlantılar, dağıtık veya merkezi bir çizelgeleme mekanizması kullanılarak belirli bir kanal ofsetine atanır ve her düğümün alım için hangi hücreyi dinleyeceğini veya belirli bir komşuya iletim için kullanacağını belirlemesine olanak tanır. Genellikle, seyrek bir ağda, zaman dilimi başına tek bir hücre kullanılır. Ancak hiçbir şey eşzamanlı iletişimin aynı anda farklı kanallarda gerçekleşmesini kısıtlamaz.

TSCH zaman çizelgesinde iki tür slot bulunur: paylaşımlı (shared) slot ve özel tahsis edilmiş (dedicated) slot. Shared slot, ağdaki tüm düğümlerin belirli yönetim paketlerini göndermek için kullanabileceği paylaşımlı bir zaman dilimidir. Bu tür slotlar EB, KA (keep alive), RPL kontrol paketleri, katılım isteği ve katılım yanıtı paketlerinin iletimi için kullanılır. Shared slotlar, yeni düğümlerin ağa katılmasına izin vermek ve ağın genel yönetimini sağlamak için kullanılır. Bu slotlar, genellikle yayın iletimine izin verir ve bu nedenle ağın genel koordinasyonu için kritiktir. Dedicated slotlar ise belirli bir düğüm çifti arasında veri iletimi için ayrılmış zaman dilimleridir. Bu slotlar, belirli bir kanal ofseti ve zaman diliminde yalnızca belirli bir düğüm çifti tarafından kullanılmak üzere ayrılır.

IEEE802.15.4e TSCH protokolünde, düğümler arasındaki zaman senkronizasyonu, ağın verimli ve güvenilir bir şekilde çalışması için kritik öneme sahiptir (Külcü & Görmüş, 2022). Zaman senkronizasyonunu sağlamak için iki farklı yaklaşım kullanılmaktadır: Çerçeve Tabanlı Senkronizasyon ve Onay Tabanlı Senkronizasyon.

Çerçeve tabanlı senkronizasyon, düğümlerin zaman senkronizasyonunu veri çerçeveleri üzerinden sağladığı bir yöntemdir. Şekil 36a'da gösterildiği gibi, zaman kaynağı komşusu bir veri çerçevesi gönderir ve bu çerçeve düğüm tarafından alınır.



Şekil 36. IEEE 802.15.4e TSCH protokolünde zaman senkronizasyonunu sağlayan iki farklı yaklaşım

- TsTxOffset: Veri çerçevesinin gönderildiği zaman dilimindeki ofset.
- GuardTime: Veri çerçevesinin alınması için belirlenen koruma süresi.
- TsTxAckDelay: Veri çerçevesi alındıktan sonra onay (ACK) mesajının gönderilmesi için geçen süre.

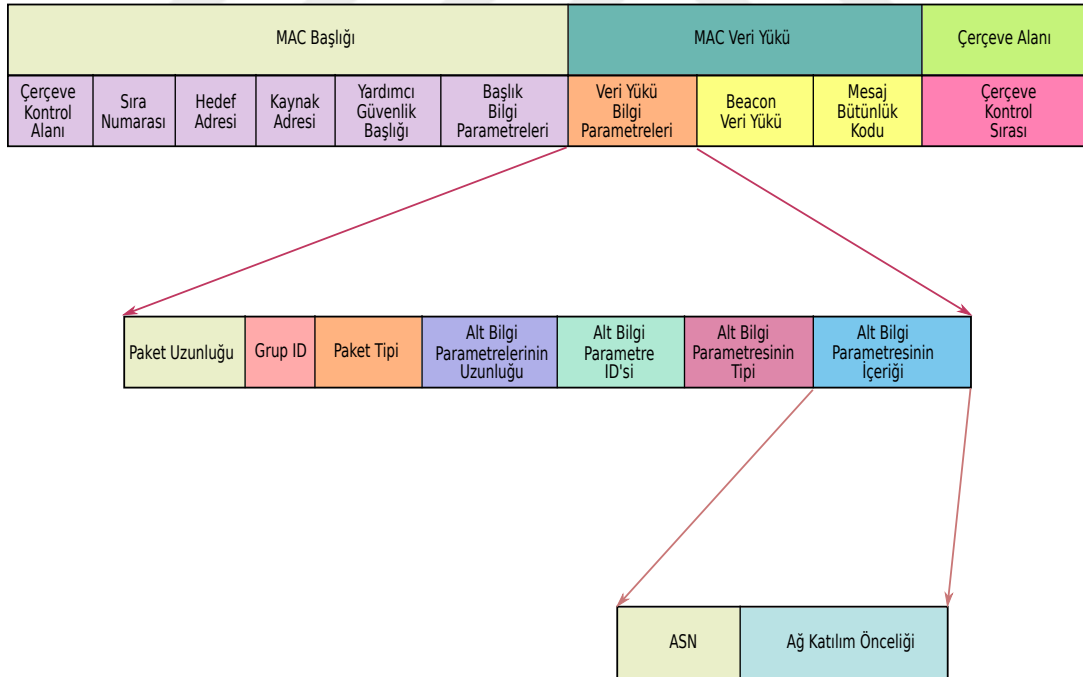
Düşüm, veri çerçevesini alır ve içindeki zaman bilgisini kullanarak kendi zaman bilgisini günceller. Bu güncelleme, düğümün diğer düğümlerle senkronize olmasını sağlar. Çerçeve tabanlı senkronizasyon, veri iletim ve alım zamanlarının doğru bir şekilde hizalanmasını sağlar ve ağdaki zaman senkronizasyonunu korur.

Onay tabanlı senkronizasyon, düğümlerin zaman senkronizasyonunu onay mesajları üzerinden sağladığı bir yöntemdir. Şekil 36b'de gösterildiği gibi, düğüm veri çerçevesini gönderir ve zaman kaynağı komşusu bu çerçeveyi alıp bir ACK mesajı gönderir.

- TsTxOffset: Veri çerçevesinin gönderildiği zaman dilimindeki ofset.
- GuardTime: Veri çerçevesinin alınması için belirlenen koruma süresi.

- TsTxAckDelay: Veri çerçevesi alındıktan sonra onay mesajının gönderilmesi için geçen süre.

Düğüm, veri çerçevesini gönderir ve zaman kaynağı komşusu bu çerçeveyi alıp bir ACK mesajı gönderir. Düğüm, ACK mesajını aldığı anda, içindeki zaman bilgisini kullanarak kendi zaman bilgisini günceller. Bu güncelleme, düğümün diğer düğümlerle senkronize olmasını sağlar. Onay tabanlı senkronizasyon, özellikle veri iletiminden sonra alınan geri bildirimler üzerinden zaman senkronizasyonunu sağlar. TSCH protokolünde Enhanced Beacon mesajları, ağdaki zaman senkronizasyonunu sağlamak ve sürdürmek için kritik bir rol oynar (Şekil 37). EB mesajları, ağdaki düğümlerin birbirleriyle senkronize olmasını ve yeni düğümlerin ağa katılmasını kolaylaştırmak için periyodik olarak yayınlanır. EB mesajları, ağdaki tüm düğümlerin aynı zaman diliminde senkronize olmasını sağlar. Bu mesajlar, düğümlerin zaman bilgilerini güncellemelerine yardımcı olur ve böylece tüm ağ düğümleri aynı zaman çizelgesine göre veri iletimi yapabilir. Yeni düğümler, ağa katılabilmek için EB mesajlarını dinler. Bu mesajlar, ağın zaman dilimini ve kanal atlama sekansını içerir. Yeni düğümler, bu bilgileri alarak mevcut ağa senkronize olabilir ve veri iletimine başlayabilir.



Şekil 37. IEEE 802.15.4e TSCH enhanced beacon paket formatı

EB mesajları, düğümlerin zaman çizelgesini ve kanal atlama sırasını içerir (De Guglielmo vd., 2016; Görmüş, 2015). Kanal atlama sırası, her zaman diliminde hangi frekans kanalının kullanılacağını belirleyen bir düzen anlamına gelir. Bu bilgiler, düğümlerin doğru zaman diliminde ve doğru kanalda veri gönderip almasını sağlar. Bu sayede, ağdaki

veri iletimi verimli ve güvenilir bir şekilde gerçekleşir. EB mesajları, ağdaki düğümlerin senkronizasyonunu düzenli olarak güncelleyerek ağın kararlılığını artırır. Düğümler, bu mesajları alarak kendi zaman bilgilerini periyodik olarak kontrol eder ve senkronizasyon kayıplarını minimize eder. Böylece, ağdaki veri iletiminin kesintisiz ve doğru bir şekilde devam edilmesi sağlanır. Ayrıca, EB mesajları, yeni düğümlerin ağa katılmasını ve mevcut düğümlerle uyumlu bir şekilde çalışmasını sağlayan diğer önemli bilgileri de içerir. Bu mesajlar, ağın genel verimliliğini ve güvenilirliğini artırarak, düğümlerin zaman çizelgesine uyum sağlamasını ve veri iletiminde doğru kanalı kullanmasını sağlar.

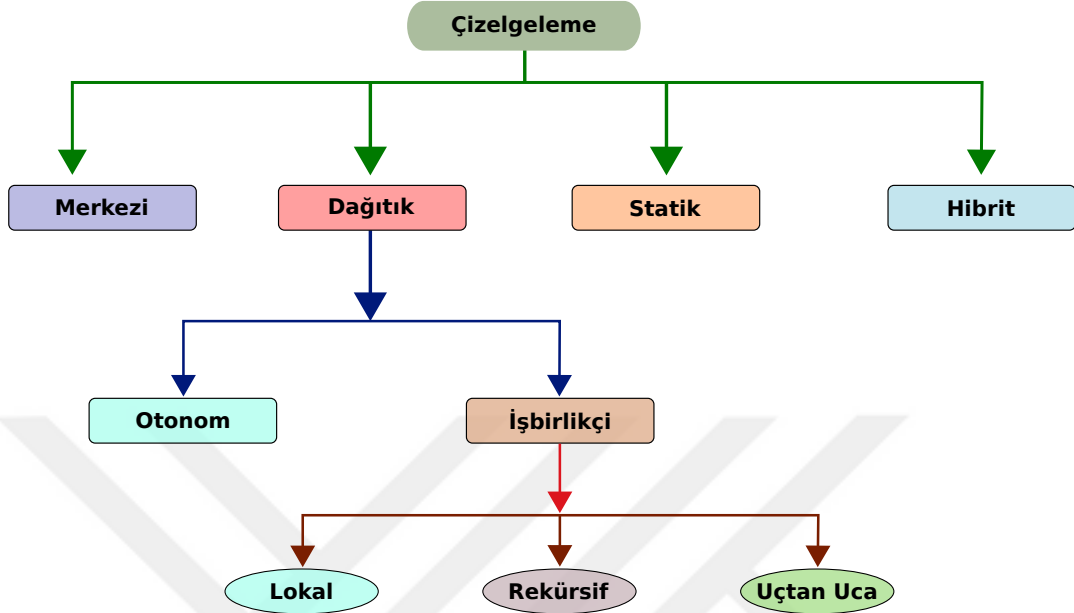
EB paketinin başında yer alan MAC Başlığı (MAC Header), çerçevenin türünü ve kontrol bilgilerini içeren Çerçeve Kontrolü (FRAME CONTROL), çerçevenin sıra numarasını belirten Sıra Numarası (Sequence Number), çerçevenin gönderileceği düğümün adresini belirten Hedef Adresi, çerçevenin gönderildiği düğümün adresini belirten Kaynak Adresi ve ek güvenlik bilgilerini içeren Yardımcı Güvenlik Başlığı (Auxiliary Security Header - AUX. SEC. HEADER) gibi alanlardan oluşur. Ayrıca, başlık bilgi elementlerini içeren Başlık Bilgi Elementleri (Header Information Elements, IEs) de MAC başlığı içinde yer alır. Beacon Yükü kısmı, yük bilgi elementlerini içeren Yük Bilgi Elementleri (Payload Information Elements) ve beacon mesajının içeriğini taşıyan Beacon Yükü (Beacon PAYLOAD) kısımlarını içerir. Bu bölüm, ağdaki düğümlerin doğru zaman diliminde senkronize olmalarını sağlar.

Çerçeve Alanı (MAC Footer - MFR) ise mesaj bütünlüğünü sağlayan Mesaj Bütünlük Kodu (Message Integrity Code - MIC) ve çerçevenin doğru iletilip iletilmediğini kontrol eden Çerçeve Kontrol Sırası (Frame Check Sequence - FCS) alanlarından oluşur. Bu alanlar, veri iletiminin güvenilirliğini ve bütünlüğünü korur. Beacon Paket Formatı, çeşitli alanlarından oluşur. Bit 0-10 (Uzunluk) paketin uzunluğunu belirler ve 0-2047 arasında bir değer alır. Bit 11-14 (Grup Kimliği) genellikle 1 olarak ayarlanır ve Bit 15 (Tür) paket türünü belirler, genellikle 1 olarak ayarlanır. İkili gruplar içinde, alt bilgi elementlerinin uzunluğunu belirleyen Uzunluk, alt bilgi elementinin kimliğini belirleyen Alt Kimlik (Sub-identifier - Sub-ID), alt bilgi elementinin türünü belirleyen Tür ve alt bilgi elementinin içeriğini taşıyan Alt Bilgi Elementi İçeriği (Sub-Information Element Content - Sub-IE Content) yer alır. Alt bilgi elementleri arasında ASN değeri, mutlak zaman numarasını belirler ve Ağ Katılım Önceliği (Join Priority), ağ katılım önceliğini belirtir.

2.4.1. 6TiSCH Mimarisinde Çizelgeleme ve Yönlendirme

6TiSCH, endüstriyel düşük güçlü kablosuz ağlarda zaman dilimlerinin ve yönlendirme yollarının verimli bir şekilde yönetilmesini sağlayan bir protokoldür (Görmüş & Yavuz, 2017;

Dehghan vd., 2020). Bu protokol yığını, merkezi çizelgeleme, dağıtık çizelgeleme, statik çizelgeleme ve hibrit çizelgeleme olmak üzere dört farklı çizelgeleme modunu içerir (Aydın vd., 2024). Şekil 38, bu dört çizelgeleme modunu göstermektedir.

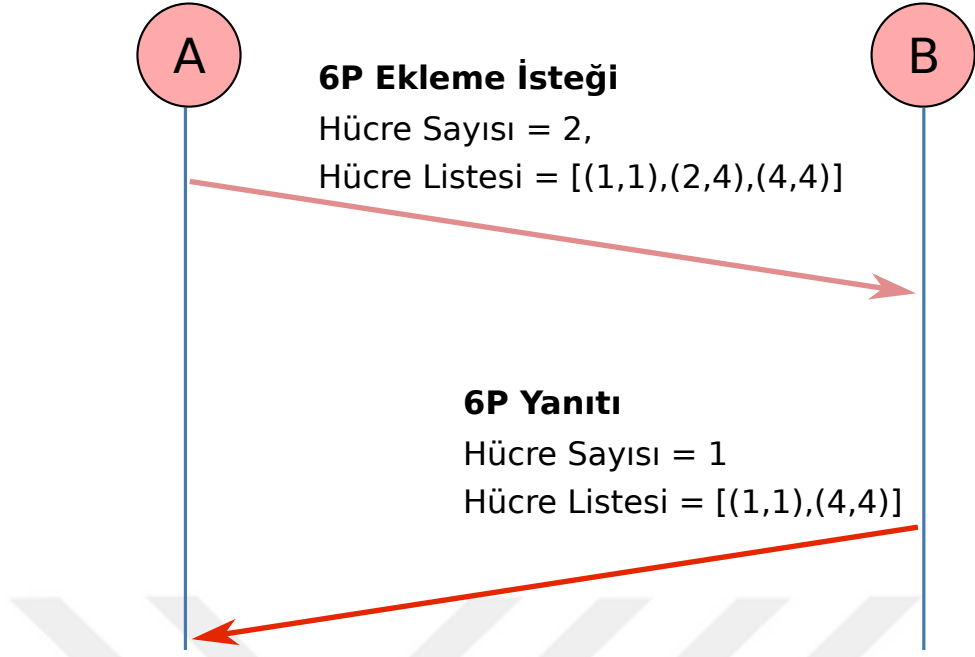


Şekil 38. Çizelgeleme stratejilerinin sınıflandırılması (Urke vd., 2022)

Her bir çizelgeleme modu, ağın ihtiyaçlarına göre farklı avantajlar ve dezavantajlar sunar. Çizelgeleme, farklı çizelgeleme işlevleri tarafından dinamik olarak uyarlanabilir. Bu işlevler, slotframe yapılandırmasını yönetir ve kontrol eder. Slotframe kullanımı (etkin bir şekilde çizelgeleme) ayarlamak için 6P protokolü gereklidir (Wang vd., 2018). Zamanlama işlevleri, bir hücre eklemek, silmek veya yerini değiştirmek için 6P işlemlerini tetikler. Bu işlemler, genellikle bir tarafın işlemi başlattığı ve ilgili hücreleri önerdiği, diğer tarafın ise sonuçla yanıt verdiği bir istek-yanıt modelini takip eder. Örneğin, Şekil 39'da 2 adımlı bir hücre ekleme işlemi gösterilmektedir. Bu işlemde, A düğümünde çizelgeleme fonksiyonu iki yeni hücreye ihtiyaç olduğuna karar verir ve bir 6P işlemi başlatır. A düğümü, ADD komutu ile birlikte aday hücrelerin listesini içeren bir 6P istek mesajı gönderir. B düğümü, çizelgeleme fonksiyonuna danışır ve kabul edilebilir hücrelerin bir alt kümesiyle yanıt verir. Böylece işlem tamamlanır ve A ile B düğümleri arasında iki yeni hücre planlanır.

6P protokolü, 6TiSCH ağlarında zamanlama ve kaynak yönetimi işlemlerini gerçekleştirmek için çeşitli işlem ve mesaj türlerini destekler. 6P paket türleri ve bunların detaylı açıklamaları maddeler halinde verilmiştir:

- 6P ADD (Hücre Ekleme): Bir düğüm, başka bir düğümden belirli bir zaman aralığında veri iletimi için hücre eklemesini isteyebilir. Bu işlem, hücrelerin zaman çizelgesine eklenmesini sağlar.



Şekil 39. A ve B düğümleri arasında 6P ekleme işlemi örneği

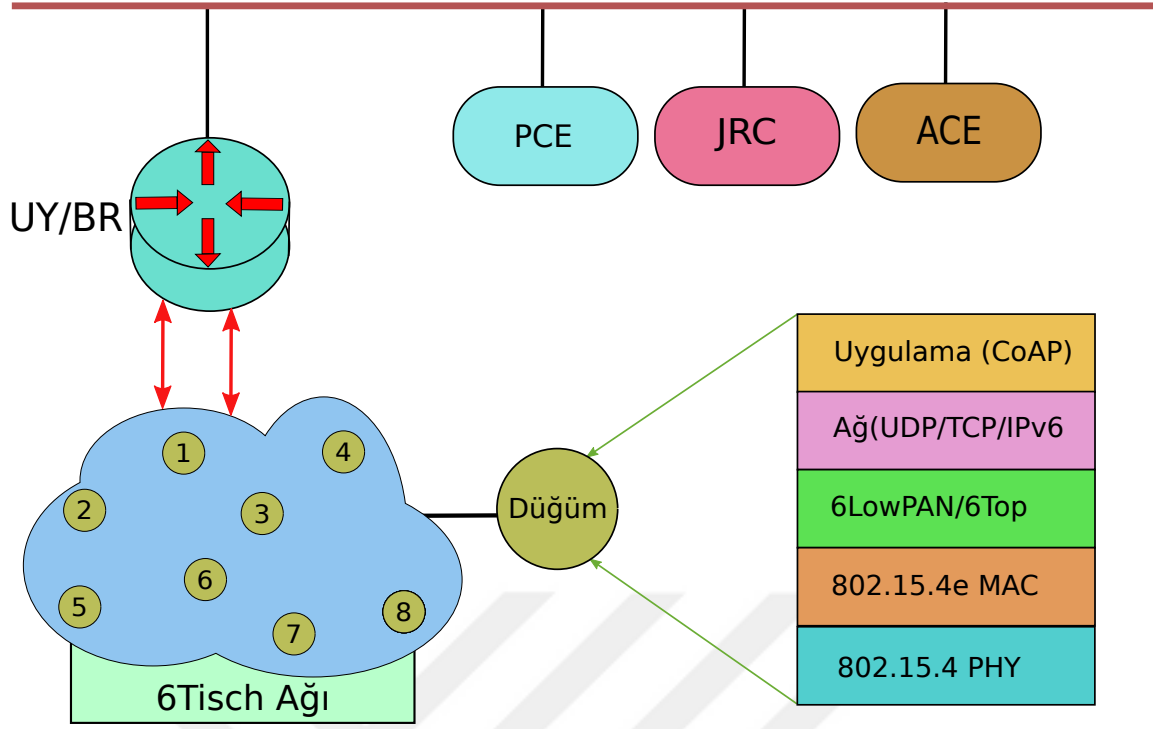
- **6P DELETE (Hücre Silme):** Bir düğüm, başka bir düğümden belirli bir zaman aralığındaki hücreleri serbest bırakmasını isteyebilir.
- **6P LIST (Hücre Listesi):** Bir düğüm, başka bir düğümden mevcut hücrelerin listesini istemek amacıyla LIST mesajı gönderebilir. Böylece, düğümün hangi hücrelerin kullanılabilir olduğunu bilmesini sağlar.
- **6P COUNT (Hücre Sayısı):** Bir düğüm, başka bir düğümden mevcut hücrelerin sayısını istemek amacıyla COUNT mesajı gönderebilir. Bu yöntem, düğümün mevcut kaynakların miktarını bilmesini sağlar.
- **6P CLEAR (Zaman Çizelgesini Temizleme):** Bir düğüm, başka bir düğümden tüm hücreleri temizlemesini ve zaman çizelgesini sıfırlamasını isteyebilir. Bu talep, düğümün zaman çizelgesini yeniden başlatmasını sağlar.
- **6P RESPONSE (Yanıt):** Yukarıdaki istek mesajlarına yanıt olarak gönderilir. İstek mesajlarına olumlu veya olumsuz yanıtlar içerir.
- **6P SIGNAL (Sinyal):** Düğümler arası belirli olayları veya durumları bildirmek için kullanılır. Örneğin, bir düğüm belirli bir hücrede çakışma tespit ettiğinde bu mesajı kullanabilir.

Her 6P mesajı, mesajın türüne ve işlevine bağlı olarak belirli alanlardan oluşur. Bu alanlar, mesajların doğru şekilde anlaşılmasını ve işlenmesini sağlar. Her 6P mesajında bulunabilecek temel alanlar ve bunların işlevleri:

- **Komut Tanımlayıcı (Command Identifier):** Bu alan, mesajın türünü belirtir ve mesajın amacını tanımlar. Örneğin, ADD, DELETE, LIST gibi komutlar bu alan aracılığıyla belirlenir. Bu tanımlayıcı, düğümün mesajı nasıl işleyeceğini belirlemesine yardımcı olur.
- **Hücre Seçenekleri (Cell Options):** Bu alan, hücrelerin özelliklerini belirtir. Hücrelerin giden, gelen veya paylaşılan olup olmadığını bu seçenekler belirler. Ayrıca, hücrelerin hangi amaçlarla kullanılacağını da (örneğin, veri iletimi veya alımı) belirler.
- **Hücre Listesi (Cell List):** Bu alan, mesajda belirtilen hücrelerin zaman ve kanal bilgilerini içerir. Her hücre için zaman dilimi ve kanal ofseti gibi bilgiler bu listede yer alır. Bu bilgiler, hücrelerin doğru şekilde planlanmasını ve kullanılmasını sağlar.
- **İşlem Tanımlayıcı (Transaction Identifier):** Bu alan, isteği ve yanıt eşleştirmek için kullanılır. Bir işlem başlatıldığında, bu tanımlayıcı sayesinde istek mesajı ve yanıt mesajı birbirine bağlanır. Bu, mesajların doğru eşleşmesini ve işlem akışının takip edilmesini sağlar.
- **Sıra Numarası (Sequence Number):** Mesajların sırasını izlemek için kullanılır. Her mesaj, belirli bir sıra numarası ile etiketlenir. Bu numara, mesajların doğru sıralamada işlenmesini ve olası mesaj kayıplarının tespit edilmesini sağlar.
- **Yük (Payload):** Ek veri veya parametreler içerir. Bu alan, mesajın türüne ve amacına bağlı olarak çeşitli bilgileri taşıyabilir. Örneğin, bir ADD mesajında, ek hücre bilgileri bu alanda yer alabilir.

Bu alanlar, 6P protokolü ile gönderilen her mesajın doğru ve etkin bir şekilde işlenmesini sağlar. Mesajların içeriği, ağdaki düğümlerin kaynakları etkin yönetmesini ve zamanlama işlemlerini koordine etmesini mümkün kılar.

Her düğüm, çizelgelemeyi uyarlamak için komşularıyla iletişim kurmak üzere 6P'yi kullanabildiğinden, farklı çizelgeleme yönetimlerini kullanabilir. Bu nedenle, her düğüm MAC katmanında sağlanan kapasitenin kendi uygulaması için yeterli olmaması durumunda hücre (slot) ekleyerek veya silerek performansını artırabilir. Ayrıca, bir hücrenin başka bir



Şekil 40. 6TiSCH merkezi kontrol yapıları

slot ofsetine ve/veya kanal ofsetine taşınması, bu bağlantıda çarpışma olasılığının diğerlerine kıyasla önemli ölçüde daha yüksek olması durumunda kullanılır.

Merkezi çizelgeleme, ağdaki zaman dilimlerini ve yönlendirme yollarını bir merkezi denetleyici aracılığıyla planlar ve kontrol eder (Kumar & Kolberg, 2023). Bu yaklaşım, ağın tüm planlama işlevlerinin merkezi bir birim tarafından yönetilmesini sağlar. Şekil 40'da gösterildiği gibi, merkezi çizelgelemenin amacı, özellikle küçük ve orta ölçekli ağlarda verimlilik ve koordinasyon sağlamaktır. Merkezi çizelgelemenin avantajları arasında yüksek verimlilik ve etkili koordinasyon yer alır. Bu avantajlar, küçük ve orta ölçekli ağlar için özellikle uygundur. Ancak, merkezi birimin arızalanması durumunda ağın işleyişi olumsuz etkilenebilir ve merkezi birimin yüksek hesaplama kapasitesine sahip olması gereklidir. Ayrıca, büyük ağlarda ölçeklenebilirlik sorunları yaşanabilir. Merkezi çizelgeleme, ağın genel performansını optimize etmek için zaman dilimlerini ve yönlendirme yollarını dikkatlice planlar ve bu sayede veri iletimini en verimli şekilde yönetir.

Dağıtık çizelgeleme, ağdaki her düğümün kendi zaman dilimlerini ve yönlendirme yollarını yerel olarak planladığı ve yönettiği bir moddur (Righetti vd., 2020). Bu mod, merkezi bir denetleyiciye ihtiyaç duymadan, düğümlerin kendi aralarında işbirliği yaparak ağın genel işleyişini sağlar. Dağıtık çizelgelemenin amacı, merkezi birim gerektirmeden ölçeklenebilir ve esnek bir ağ yönetimi sağlamaktır. Avantajları arasında merkezi birime

bağımlılık olmaması, esneklik ve ölçeklenebilirlik bulunur. Ayrıca, merkezi arızalardan etkilenmez. Ancak, yerel hesaplamalar ve koordinasyon eksikliği nedeniyle verimlilik düşebilir ve büyük ağlarda karmaşıklık artabilir. Dağıtık çizelgeleme, otonom çizelgeleme ve işbirlikçi çizelgeleme olmak üzere iki kategoriye ayrılır. Otonom çizelgelemede her düğüm, kendi çizelgesini bağımsız olarak belirlerken, işbirlikçi çizelgelemede ise düğümler, çizelgelerini birbirleri ile işbirliği yaparak belirler. İşbirlikçi çizelgeleme, kendi içinde yerel (Local), yinelemeli (Recursive) ve uçtan uca (End-to-end) çizelgeleme olarak alt kategorilere ayrılır. Yerel işbirlikçi çizelgeleme, düğümlerin sadece komşu düğümleri ile işbirliği yaparak çizelgelerini belirlemesini sağlar. Bu yöntem, yerel optimizasyonlar yaparak verimliliği artırabilir. Yinelemeli işbirlikçi çizelgeleme, düğümlerin belirli bir alan içindeki tüm düğümlerle işbirliği yaparak çizelgelerini belirlemesini içerir. Bu yöntem, daha geniş bir alanda optimizasyon sağlar ve ağın genel verimliliğini artırabilir. Ancak, yinelemeli işbirliği daha fazla hesaplama ve iletişim gerektirebilir. Uçtan uca işbirlikçi çizelgeleme, düğümlerin ağın uç noktalarındaki düğümlerle işbirliği yaparak çizelgelerini belirlemesini içerir. Bu yöntem, ağın uç noktalarındaki düğümler arasında doğrudan iletişim ve koordinasyon sağlayarak, veri iletiminde gecikmeleri minimize eder ve ağın genel performansını artırır.

Statik çizelgeleme, ağın başlangıç aşamasında belirlenen sabit bir çizelgeyi kullanır ve bu çizelge ağın çalışması sırasında değişmez (Thubert, 2019). Bu modun amacı, sabit ve öngörülebilir bir veri iletimi sağlamaktır. Statik çizelgelemenin avantajları arasında basitlik ve düşük yönetim gereksinimi bulunur. Değişmeyen ağ yapıları için oldukça uygundur. Ancak, dinamik değişikliklere uyum sağlayamaz ve esneklikten yoksundur. Statik çizelgeleme, özellikle ağın yapısının ve veri iletiminin önceden belirlenmiş ve sabit olduğu senaryolarda kullanılır. Merkezi çizelgelemeden farklı olarak, statik çizelgeleme merkezi bir denetleyici tarafından yönetilmez. Merkezi çizelgeleme, ağdaki zaman dilimlerini ve yönlendirme yollarını merkezi bir birim aracılığıyla planlar ve kontrol ederken, statik çizelgeleme, başlangıçta belirlenen çizelgenin tüm ağ ömrü boyunca değiştirilmeden kullanıldığı bir yaklaşımdır. Bu nedenle, statik çizelgeleme daha basit ve düşük yönetim gereksinimi olan ağlar için uygundur, ancak ağın dinamik değişikliklere uyum sağlama kabiliyeti sınırlıdır.

Hibrit çizelgeleme, merkezi ve dağıtık çizelgeleme yöntemlerinin bir kombinasyonunu kullanır (Thubert, 2019). Bu mod, merkezi bir denetleyici ve yerel düğümlerin birlikte çalışmasını sağlar. Hibrit çizelgelemenin amacı, hem merkezi hem de dağıtık çizelgeleme yöntemlerinin avantajlarını birleştirmektir. Bu yaklaşım, merkezi yönetimin koordinasyon ve verimliliğini, dağıtık yönetimin esneklik ve ölçeklenebilirlik avantajlarıyla bir araya getirir. Hibrit çizelgelemenin avantajları arasında yüksek verimlilik ve esneklik bulunur, bu da

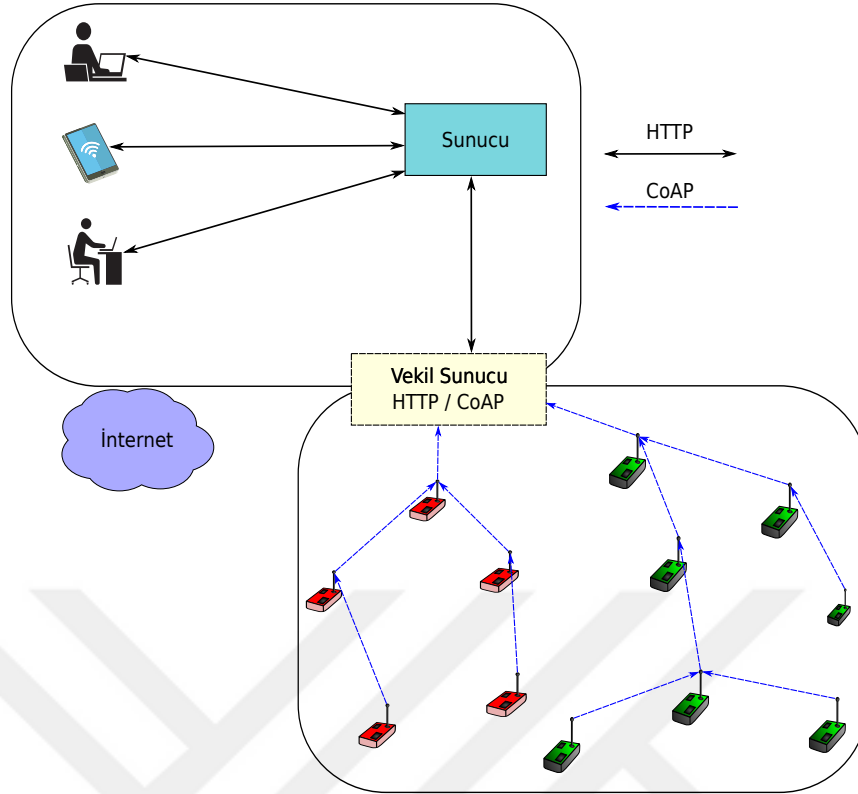
büyük ve dinamik ağlar için uygundur. Merkezi bir denetleyici, kritik görevler ve zaman dilimleri için ana planlamayı yaparken, yerel düğümler daha az kritik görevler ve değişen ağ koşullarına göre kendi planlamalarını yapabilirler. Bu sayede, ağın genel performansı ve güvenilirliği artar. Ancak, bu yaklaşımın karmaşıklık ve yönetim gereksinimi yüksektir. Hem merkezi hem de yerel bileşenlerin uyum içinde çalışması gerektiği için, koordinasyon ve senkronizasyon mekanizmaları gereklidir. Hibrit çizelgeleme, ağın belirli bölümlerinin merkezi olarak yönetilmesini sağlarken, diğer bölümlerinin yerel olarak planlanmasına olanak tanır. Örneğin, bir akıllı şehir uygulamasında, merkezi denetleyici trafik ışıkları gibi kritik altyapıları yönetirken, sensör ağları yerel olarak veri toplayabilir ve iletebilir. Bu mod, ağın genel verimliliğini artırmak için merkezi ve yerel kararların dengeli bir şekilde alınmasını sağlar. Bu sayede, hem büyük ölçekli koordinasyon gerektiren durumlarda hem de dinamik ve değişken koşullara hızlı adaptasyon gerektiren durumlarda etkili bir çözüm sunar.

2.5. CoAP

CoAP, IoT cihazlarındaki kısıtlı düğümler için hafif bir uygulama katmanı protokolüdür (Oliver & Purusothaman, 2022). CoAP, düşük ek yük, multicast desteği ve kısıtlı ortamlar için basitlik ile uygulama uç noktaları arasında etkileşimi destekleyen farklı istek/yanıt yöntemleri ile çalışır (Alhaidari & Alqahtani, 2020). Şekil 41, Constrained RESTful Environments (CoRE) ReSTful mimarisini göstermektedir. Representational State Transfer (REST), web servisleri geliştirirken kullanılacak bir dizi kısıtlamayı belirleyen yazılım tasarım stilidir (Krech vd., 2023). REST mimarisiyle uyumlu olan web servislerine RESTful Web Services denir. CoRE çalışmasının amacı, REST mimarisini en kısıtlı düğümler ve ağlar için kabul edilebilir bir biçimde uygulamaktır. Kısıtlı cihazların REST mimarisiyle uyumlu hale getirilmesi için bir çözüm, paketlerin parçalanmasıdır, ancak bu çözüm veri aktarım hızını düşürür.

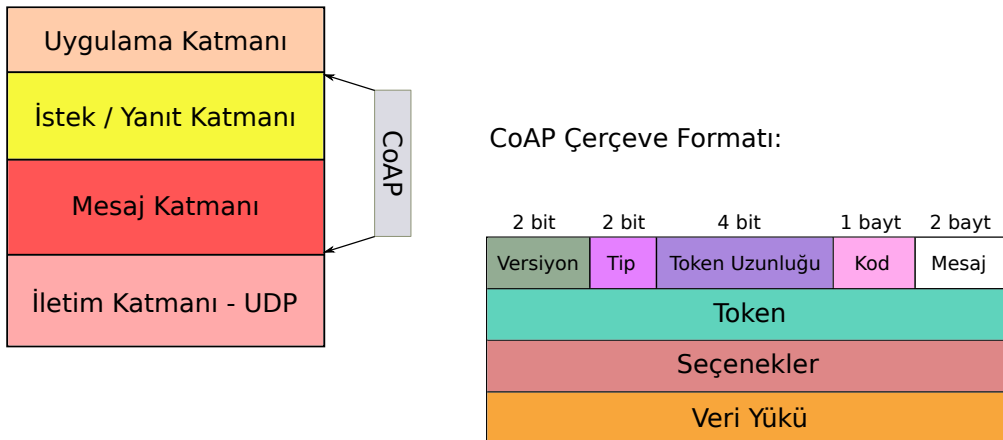
CoAP, kısıtlı cihazlar için parçalama gereksinimini ortadan kaldırırken düşük ek yükle REST mimarisinin uygulanması için yeni bir yaklaşım sunar (Majumder vd., 2021). Şekil 42’de, CoAP protokol yığını, mesaj ve başlık formatı gösterilmektedir. Bu şekil, CoAP’nin katmanlı yapısını ve her katmanın işlevlerini detaylandırır.

CoAP’nin uygulama katmanı, HTTP’ye benzer şekilde çalışır ve RESTful web hizmetlerinin uygulanmasını sağlar. Bu katman, CoAP’nin uygulama uç noktaları arasında veri alışverişini yönetir ve belirli istek/yanıt işlemlerini gerçekleştirir. Tablo 5’te gösterildiği



Şekil 41. CoRE ReSTful mimarisi

gibi, uygulama katmanında GET, POST, PUT ve DELETE yöntemleri kullanılır. İstek/yanıt katmanı, CoAP'nin istemci ve sunucu arasındaki etkileşimini yönetir. Bu katman, isteklerin nasıl yapılacağını ve yanıtların nasıl alınacağını belirler. HTTP'den farklı olarak, kısıtlı cihazlar için optimize edilmiştir. İstek ve yanıt, CoAP mesajları aracılığıyla asenkron olarak değiş tokuş edilir. Tablo 6'da istemci isteklerinin başarılı olup olmadığını ve yanıt türlerini göstermek için kullanılan CoAP yanıt kodları sunulmuştur. Ayrıca, CoAP mesajı bazı isteğe bağlı istek ve yanıt bilgilerini taşır.



Şekil 42. CoAP protokol yığını, mesaj ve başlık formatı

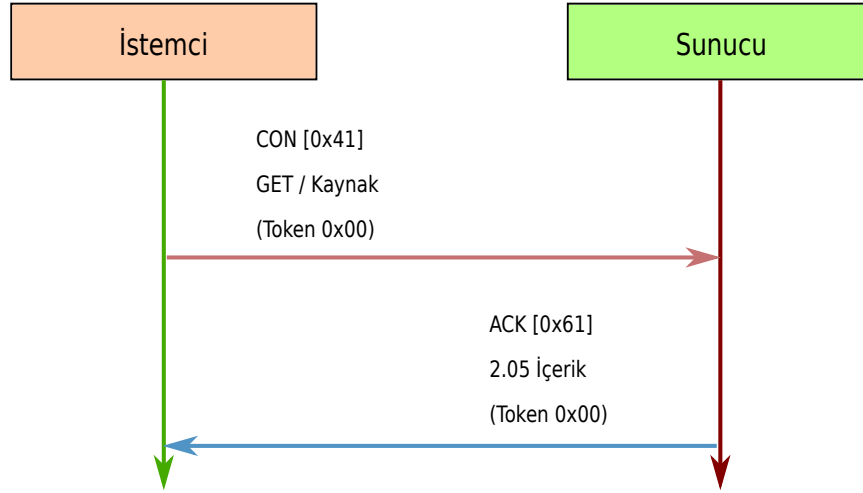
Tablo 5. CoAP istek yöntemleri

Mesaj Tipi	Açıklama
GET	0 URI isteğiyle tanımlanan kaynağa şu anda karşılık gelen bilgileri elde etmek için kullanılır.
POST	İstek'te kullanılan açıklamanın işlenmesini önerir.
PUT	URI gereksinimiyle belirtilen kaynağın kapalı temsil ile değiştirilmesini veya oluşturulmasını önerir.
DELETE	URI isteğiyle belirtilen kaynağın kaldırılmasını önerir.

Tablo 6. CoAP yanıt kodları

Mesaj Tipi	Açıklama
2.XX (Başarılı)	İstemci isteğinin alındığını ve başarıyla kabul edildiğini gösterir. Son iki XX şu anlama gelir: .01 (Oluşturuldu), .02 (Silindi), .03 (Geçerli), .04 (Değiştirildi) ve .05 (İçerik).
4.XX (İstemci Hatası)	Sunucunun isteği anlamadığını gösterir. Son iki XX şu anlama gelir: .00 (Kötü İstek), .01 (Yetkisiz Erişim), .02 (Kötü Seçenek), .03 (Yasak), .04 (Bulunamadı), .05 (Yöntem İzin Verilmiyor), .06 (Kabul Edilemez), .12 (Önkoşul Başarısız), .13 (İstek Varlığı Çok Büyük) ve .15 (Desteklenmeyen İçerik Formatı).
5.XX (Sunucu Hatası)	Sunucu hatasını gösterir. Son iki değer, .00 (Dahili Sunucu Hatası), .01 (Uygulanmadı), .02 (Kötü Ağ Geçidi), .03 (Sunucu Kullanılamaz), .04 (Ağ Geçidi Zaman Aşımı) ve .05 (Vekil Sunucu Desteklenmiyor) gibi farklı anlamlara gelir.

Mesaj katmanı, CoAP'nin temel veri iletim mekanizmasını sağlar. Bu katman, Onaylanabilir (Confirmable - CON), Onaylanamaz (Non-confirmable - NON), Onay (Acknowledgement - ACK) ve Sıfırlama (Reset - RST) olmak üzere dört ana mesaj türü kullanır. Şekil 43, CoAP protokolü kullanılarak istemci ve sunucu arasında gerçekleştirilen mesaj aktarım sürecini ayrıntılı bir şekilde göstermektedir. ACK ve RST, CoAP sunucu yanıt türleridir; ACK mesajı belirli bir onay mesajın alındığını belirtir ve RST mesajı CON ve NON mesajlarında bazı bağlamların eksik olduğunu gösterir. CON ve NON mesajları, CoAP İstek/Yanıt yöntemleridir; CON, bir mesajın güvenilirliğini gösterir ve ACK gerektirir, NON ise güvenilir iletim veya ACK gerektirmez, ancak bir kopyalama tanımlama Mesaj Kimliği (Message-ID) içerir. Mesaj katmanı, mesajların iletilmesi, tekrar iletilmesi ve doğrulanması süreçlerini yönetir. Ayrıca, mesajların güvenilir bir şekilde iletilmesini sağlar. Şekil 42'de ayrıca CoAP başlık formatı detaylandırılmıştır. Bu format, V (Version), T (Type), TLK (Token Length), Kod (Code), Mesaj Kimliği (Message ID), Token, Seçenekler (Options) ve Veri Yüğü (Payload) gibi bileşenlerden oluşur. Her bir bileşen, CoAP mesajlarının belirli



Şekil 43. CoAP mesaj aktarımı

bir işlevi yerine getirmesine yardımcı olur ve mesajın içeriği ve meta verileri hakkında bilgi sağlar. Version alanı, CoAP sürümünü gösteren 2-bitlik bir tamsayıdır. Tip, mesaj türünü gösteren 2-bitlik bir tamsayıdır. Token Uzunluğu, jeton (token) alanının uzunluğunu (0-8 bayt) gösteren 4-bitlik bir tamsayıdır. Kod alanı, 8-bitlik bir tamsayıdır. İlk 3 bit, sınıfı belirtir (örneğin, istemci isteği veya sunucu yanıtı) ve kalan 5 bit, bu sınıfa ait daha spesifik bir durumu veya eylemi belirtir. Örneğin, 0.01 değeri GET isteğini belirtirken, 2.05 değeri İçerik (Content) yanıtını, 4.04 değeri Bulunamadı (Not Found) yanıtını ve 5.00 değeri İç Sunucu Hatası (Internal Server Error) yanıtını belirtir. Mesaj alanı, CON/NON türündeki mesajları ACK/RST ile eşleştirmek ve mesaj çoğaltmasını tespit etmek için kullanılan 16-bitlik bir tamsayıdır (Khalil vd., 2024).

CoAP mesajlarını uygulama katmanında korurken, meta verilerin çoğaltılmasını önlemek ve yeniden oynatma ile sıra dışı mesaj ekleme gibi saldırılara karşı koruma sağlamak gerekir. CoAP veri yükü dışında, uçtan uca korumaya ihtiyaç duyan hassas veriler de bulunmaktadır. Örneğin, CoAP URI'si, Hafif Makineler Arası İletişim (Lightweight Machine to Machine - LwM2M) ile kullanıldığında, uygulamaya sahip olduğu hedef LwM2M nesnesini ve kaynağını, ayrıca kaynak üzerinde gerçekleştirilecek işlemi belirten REST yöntemini gösterir. Bu tür verilerin bir kısmının hem şifrelenmesi hem de bütünlüğünün uçtan uca korunması gereklidir. Bu korumanın sağlanması için COSE (CBOR Object Signing and Encryption), CoAP mesajlarını korumak için kısıtlı cihazlarda çalışacak şekilde tasarlanmıştır (Gunnarsson vd., 2021). COSE sadece CoAP yükünü koruyabilir, CoAP protokolünün meta verilerini korumaz. IETF, CoAP başlık ve seçeneklerindeki meta veriler de dahil olmak üzere CoAP mesajlarını korumak için COSE'nin nasıl kullanılacağını belirten OSCORE standardını tanımlar. Standart, CoAP mesajlarının formatlarını, hangi CoAP başlık alanlarının korunup

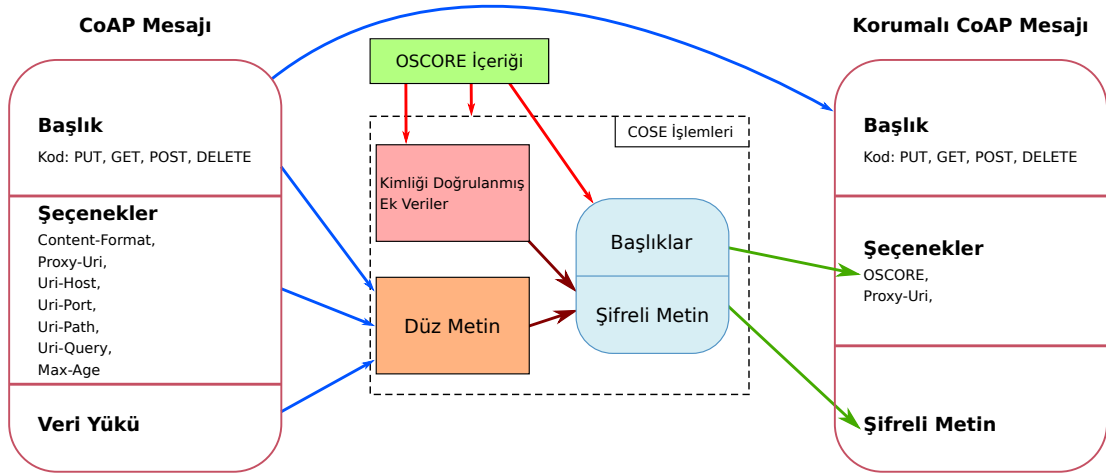
korunmayacağını, şifrelenmesi ve bütünlüğünün korunması gereken CoAP seçeneklerini açıklar.

2.5.1. OSCORE

OSCORE, kısıtlı düğümler ve ağlar için tasarlanmış bir web transfer protokolü olan CoAP'ı korumak için geliştirilmiş uygulama katmanı güvenlik protokolüdür (Gunnarsson vd., 2021). HTTP gibi RESTful protokollerle uyumlu çalışarak, uçtan uca güvenliği sağlamak için kullanılır (Gunnarsson vd., 2022). OSCORE, kısıtlı düğümler ve ağlar için tasarlanmış olup, İlişkili Verilerle Kimlik Doğrulama Şifreleme (Authenticated Encryption with Associated Data - AEAD) şifrelemesi sayesinde uçtan uca güvenlik sağlar. Bu güvenlik, gizlilik, bütünlük ve mesaj doğruluğunu kapsar. Mesajın her bir kısmının kritikliği doğrultusunda çeşitli koruma modları sunar: Gizlilik ve bütünlük, en hassas veriler için uygulanırken, bazı seçenekler yalnızca bütünlük korumasına tabi tutulur ve hassas olmayan veriler görünür kalır (örneğin, CoAP Token, CoAP URI-Host, vb.). Bu özellikler sayesinde, OSCORE ara düğümler tarafından herhangi bir mesaj alanının değiştirilmesini veya erişilmesini önlerken, onların operasyonlar gerçekleştirmesine (örneğin, CoAP istek/yanıtlarını yönlendirme ve zamanlama, protokoller arasında çeviri yapma) izin verir (Díaz & Almenares, 2023).

Ayrıca, OSCORE, TCP, UDP veya IP olmayan taşıma katmanındaki protokoller gibi herhangi bir iletişim protokolü ile uyumludur. Bu uyumluluk, OSCORE'un farklı ağ yapılandırmaları ve iletişim senaryoları ile kolayca entegre olmasını sağlar. Bu sayede, uygulama katmanındaki verilerin güvenliği sağlanırken, farklı taşıma katmanı protokolleri üzerinden güvenli veri alışverişi yapılabilir. OSCORE, uygulama verilerini korurken, taşıma katmanında TLS/DTLS gibi güvenlik protokollerinin de kullanılmasına olanak tanır. Bu, veri iletiminin her aşamasında güvenliği artırır ve veri bütünlüğünü sağlar.

Şekil 44, OSCORE ile korunan mesaj içeriğinde gerçekleşen işlemleri gösterir. Bu süreçte, CoAP mesajı belirli adımlardan geçerek şifrelenir ve bütünlük kontrolü sağlanır. İlk olarak, CoAP mesajının düz metin içeriği (payload) ve bazı başlık alanları, kimlik doğrulamalı ek veriler (Authenticated Additional Data) ile birlikte COSE işlemlerine tabi tutulur. Bu aşamada, şifreleme ve bütünlük kontrolü uygulanarak mesajın güvenliği sağlanır. Başlık alanları, şifrelenmiş metin ve kimlik doğrulamalı ek veriler, COSE işlemleri ile korunur. Bu işlem, AEAD algoritması kullanılarak gerçekleştirilir. Şifrelenmiş metin, mesajın gizliliğini korurken, kimlik doğrulamalı ek veriler, mesajın bütünlüğünü ve kimliğini doğrular. Son



Şekil 44. CoAP mesajlarının OSCORE ile şifrlenmesi

olarak, OSCORE seçeneği eklenir ve bu seçenek, anahtar tanımlayıcı, algoritma ve kısmi başlangıç vektörü gibi bilgileri içerir. Bu bilgiler, mesajın doğru bir şekilde işlenmesini ve güvenliğini sağlar. Bu adımlar tamamlandıktan sonra, OSCORE ile korunan mesaj, alıcıya gönderilmeye hazır hale gelir. Bu süreç, mesajın gizliliğini, bütünlüğünü ve kimliğini korurken, aynı zamanda ara düğümler tarafından yapılan işlemleri destekler.

2.6. Contiki ve Cooja

Contiki ve Cooja, özellikle IoT uygulamaları için geliştirilmiş önemli araçlardır (Simha vd., 2020). Bu bölümde, Contiki işletim sistemi ve Cooja simülatörü hakkında ayrıntılı bilgi verilecek ve bu araçların IoT ağlarında nasıl kullanıldıkları açıklanacaktır. Contiki, düşük güçlü cihazlar ve kısıtlı ağ kaynakları için optimize edilmiş bir işletim sistemi iken, Cooja, bu sistemin simülasyon ve test edilmesi için kullanılan güçlü bir simülatördür. Bu araçlar, IoT ağlarının geliştirilmesi, test edilmesi ve değerlendirilmesi süreçlerinde önemli bir rol oynar ve birçok araştırmacı ve geliştirici tarafından tercih edilmektedir.

Contiki, gömülü sistemler ve IoT cihazları için tasarlanmış açık kaynaklı, çok görevli bir işletim sistemidir (Österlind, 2006). 2002 yılında Adam Dunkels tarafından geliştirilen işletim sistemi, özellikle sınırlı kaynaklara sahip cihazlarda çalışmak üzere optimize edilmiştir. Düşük bellek ve enerji tüketimi ile dikkat çeken Contiki, bu özellikleri sayesinde sensör ağları ve IoT uygulamaları için idealdir (Zikria vd., 2018).

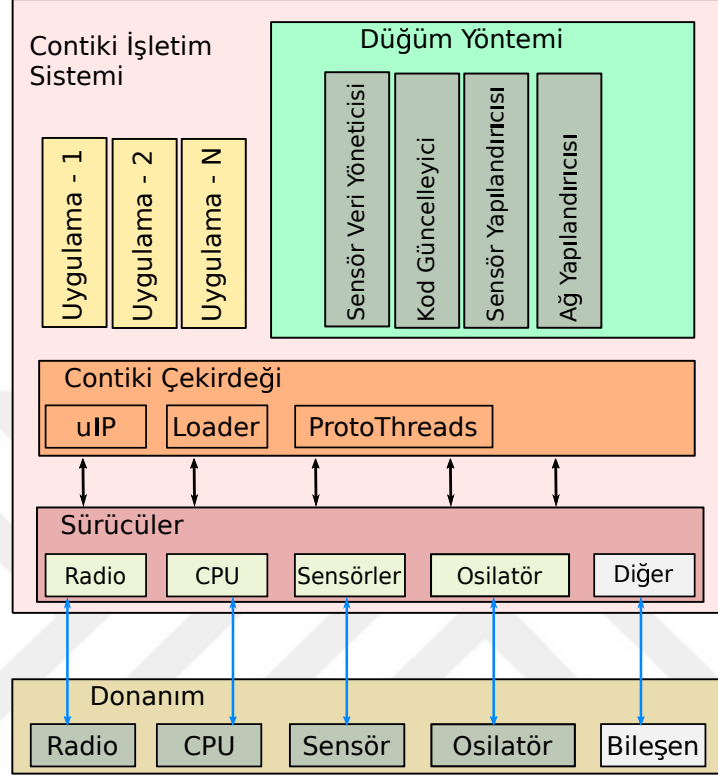
Modüler bir yapıya sahip olan Contiki'nin çekirdek mimarisi, sistemin esnek ve genişletilebilir olmasını sağlar. Temel bileşenler şunlardır:

- Çekirdek (Kernel): Çoklu görev yönetimi ve süreçler arası iletişim sağlar. Bu, sistem kaynaklarının etkin kullanılmasını ve farklı işlemlerin paralel olarak yürütülmesini mümkün kılar.
- Protothreads: İş parçacığı yönetimi için hafif bir çözüm sunar. Protothreads, bellek kullanımı açısından verimli olup, düşük bellekli cihazlarda iş parçacığı oluşturma ve yönetme işlemlerini kolaylaştırır.
- UIP: Hem IPv4 hem de IPv6 destekleyen bir ağ yığınıdır. UIP, IoT cihazlarının internet üzerinden veri iletişimini sağlar ve geniş bir protokol desteği sunar.
- Rime: Düşük güçlü kablosuz iletişim için kullanılan bir ağ protokolüdür. Rime, enerji verimliliği yüksek iletişim yöntemleri sunarak, sensör ağları gibi enerji kısıtlı ortamlarda etkili bir şekilde çalışır.

Contiki işletim sistemi, özellikle IoT uygulamaları için birçok avantaj sunar. Aşağıda, Contiki'nin sunduğu avantajlar ve özellikler detaylı bir şekilde açıklanmaktadır:

- Geniş Platform Desteği: Contiki, birçok donanım platformunu destekler. Bunlar arasında Telos, Z1, Sky, MicaZ ve MSP430 tabanlı cihazlar bulunur. Bu geniş destek, farklı donanımlar üzerinde uygulamaların kolayca geliştirilip dağıtılabilmesini sağlar.
- Zengin Ağ Protokolü Desteği: Contiki, IoT ağ protokollerini geniş bir yelpazede destekler. 6LoWPAN, RPL, CoAP gibi protokoller sayesinde, IoT cihazları arasında düşük güçlü ve güvenilir iletişim sağlayan yapılar geliştirilebilir.
- Enerji Verimliliği: Contiki, enerji tasarrufu sağlamak amacıyla düşük güçlü çalışma modları sunar. Bu modlar, pil ömrünü uzatır ve enerji tüketimini minimumda tutarak cihazların daha uzun süre çalışabilmesini sağlar.
- Esneklik ve Modülerlik: Modüler yapısı sayesinde, kullanıcılar ihtiyaçlarına göre sistemi özelleştirebilirler. Bu esneklik, farklı uygulama senaryolarında Contiki'nin kolayca adapte edilmesini mümkün kılar.
- Güvenilirlik: Contiki, düşük güçlü ve kayıplı ağlar için optimize edilmiştir. Bu özellik, IoT cihazlarının güvenilir ve kesintisiz iletişim sağlamasına yardımcı olur.
- Gelişmiş Simülasyon Desteği: Contiki, Cooja simülatörü ile birlikte gelir. Cooja, büyük ölçekli IoT ağlarının simülasyonunu yaparak, ağ performansını analiz etme ve olası sorunları önceden tespit etme imkanı sunar.

- Açık Kaynak ve Topluluk Desteği: Contiki, açık kaynak bir işletim sistemi olup geniş bir geliştirici topluluğu tarafından desteklenmektedir. Bu, sürekli güncellemeler ve yeni özelliklerin eklenmesi anlamına gelir.



Şekil 45. Contiki OS mimarisi

Bu özellikler, Contiki'yi IoT uygulamaları için güçlü ve esnek bir çözüm haline getirir. Özellikle düşük güçlü ve kayıplı ağlarda kullanımı, onu enerji verimliliği ve güvenilirlik açısından öne çıkarır.

Cooja, Contiki işletim sistemi ile birlikte kullanılan güçlü bir ağ simülatörüdür (Bagula & Erasmus, 2015). Kullanıcıların sanal bir ağ ortamında IoT cihazlarını ve ağ protokollerini simüle etmelerine olanak tanır, böylece gerçek donanım üzerinde test yapmadan önce ağ senaryolarını ve protokolleri test etmek mümkündür (Nandhini vd., 2023). Simülatör, düğüm seviyesinde (Node level) ve paket seviyesinde (Packet level) simülasyonlar yapabilir. Sanal düğümler, gerçek donanım düğümleri ve hibrit düğümler simüle edilebilir. Kullanıcılar, farklı ağ topolojilerini ve senaryolarını oluşturabilir ve bu senaryoları gerçek zamanlı olarak izleyebilir. Simülasyon sürecinde ağ trafiğini, düğüm enerji tüketimini ve diğer performans metriklerini görselleştirir ve analiz eder. Contiki ile tam uyumlu çalışması, kullanıcıların uygulamalarını simülasyon ortamında kolayca test etmelerini sağlar.

Cooja, radyo paketlerinin izlenmesine olanak tanır ve kullanıcıların ağ trafiğini ve paket kaybını detaylı bir şekilde analiz edebilmesini sağlar. Simülatör, simülasyon senaryolarını

otomatikleştirmek ve özelleştirmek için script yazma imkanı sunar. NoGUI (grafik kullanıcı arayüzü olmadan) ile çalışabilme özelliği sayesinde, ağ simülasyonları komut satırı üzerinden başlatılabilir ve yönetilebilir. Bu özellik, büyük ölçekli simülasyonların otomatik olarak yürütülmesini ve analiz edilmesini sağlar. Contiki ve Cooja, IoT araştırmalarında ve geliştirmelerinde önemli araçlar olarak öne çıkmaktadır. Contiki, düşük güçlü cihazlar ve kısıtlı ağ kaynakları için optimize edilmiş bir işletim sistemi sunarken, Cooja bu sistemin simülasyon ve test edilmesi için güçlü ve esnek bir platform sağlar. Şekil 45, Contiki OS'nin mimarisini göstermektedir. Cooja, diğer simülasyon platformlarına kıyasla düşük seviye radyo modellemeleri, düğüm davranışlarının detaylı simülasyonu, radyo paketlerinin izlenmesi, script yazma ve NoGUI ile çalışabilme gibi özellikleriyle öne çıkar. Bu özellikler, Cooja'nın IoT ağlarının gerçekçi ve ayrıntılı bir şekilde simüle edilmesini sağlayarak, araştırma ve geliştirme süreçlerini destekler.

2.6.1. Bellek Tahsisi ve Yönetimi

Bellek tahsisi ve yönetimi, Contiki işletim sisteminde genellikle statik tahsis yoluyla yapılır, ancak dinamik ihtiyaçlar için memb ve mmem modülleri de kullanılabilir. Statik yöntem, belleğin program başlatılırken tahsis edilmesini ve program sona erdiğinde serbest bırakılmasını sağlar. Bu yöntem, bellek yönetimini basitleştirir ve bellek sızıntılarını önler. Ancak, statik tahsis dinamik bellek ihtiyaçlarına esnek bir şekilde yanıt veremez.

Contiki, dinamik bellek tahsisi için memb ve mmem modüllerini sağlar. Bu modüller, belleğin daha verimli ve esnek bir şekilde yönetilmesine yardımcı olur.

- memb modülü: Belirli boyutta ve sayıda bellek blokları tahsis eder. Bu modül, genellikle küçük ve sabit boyutlu veri yapılarının yönetimi için kullanılır. Örneğin, ağdaki düğümlerin adres bilgilerini tutan bir yapı düşünüldüğünde, bu yapının her biri 20 bayt olsun ve ağda en fazla 50 düğüm olabileceği biliniyorsa, memb modülü kullanılarak 50 adet 20 baytlık bellek blokları tahsis edilebilir. Bu, bellek kullanımını optimize eder ve gereksiz bellek tahsisinden kaçınılmasını sağlar.

```
MEMB(node_addresses, struct node_addr, 50);
```

```
// Bellek bloklarını kullanma
```

```
struct node_addr *addr = memb_alloc(&node_addresses);
```

```

if (addr != NULL) {

    // Adresi kullanır

}

// Kullanımı bittikten sonra belleği serbest bırakır

memb_free(&node_addresses, addr);

```

- **mmem modülü:** Daha büyük ve dinamik olarak boyutlandırılabilen bellek blokları için kullanılır. Bu modül, belleğin daha esnek yönetimini sağlar ve genellikle değişken boyutlu verilerin yönetiminde kullanılır. Örneğin, bir sensör ağında düğümlerden gelen veri paketlerinin boyutu değişebilir. Bu durumda, mmem modülü kullanılarak gerektiğinde farklı boyutlarda bellek blokları tahsis edilebilir.

```

struct mmem packet_memory;

// Bellek bloğunu tahsis eder

if (mmem_alloc(&packet_memory, 100) == 0) {

    // Tahsis edilen bellek bloklarını kullanır

}

// Kullanımı bittikten sonra belleği serbest bırakır

mmem_free(&packet_memory);

```

Bu örnekler, Contiki işletim sisteminde bellek yönetiminin nasıl yapıldığını ve memb ve mmem modüllerinin nasıl kullanıldığını göstermektedir. Bu modüller, dinamik bellek ihtiyaçlarını karşılayarak bellek yönetimini daha esnek ve verimli hale getirir.

Contiki OS’de bellek tahsisini optimize etmek ve kullanılan bellek miktarını analiz etmek için msp430-size aracı tercih edilmektedir. msp430-size, MSP430 mikrodenetleyicileri için derlenmiş ikili dosyaların bellek kullanımını analiz eden bir araçtır (Eriksson vd., 2008). Bu araç, programın .text, .data, .bss, dec ve hex gibi bellek bölümlerinin boyutlarını raporlayarak geliştiricilerin bellek tüketimini izlemelerine ve optimize etmelerine olanak tanır.

Şekil 46’da görüldüğü gibi, msp430-size komutu, programın farklı bellek segmentlerini ölçmek için kullanılır. Bu çıktıda:

```
$ msp430-size udp-server.exp5438
text      data      bss      dec      hex filename
112940    1098    13828    127866    1f37a udp-server.exp5438
```

Şekil 46. msp430-size aracı ile bellek ölçümü

- .text bölümü, programın çalıştırılabilir kodunu içerir ve ROM’da yer alır.
- .data bölümü, başlangıç değerleri ile birlikte RAM üzerinde depolanan global ve statik değişkenleri içerir.
- .bss bölümü, başlangıçta sıfır olarak başlatılan global ve statik değişkenleri içerir. Bu değişkenler de RAM’de yer alır.
- dec, toplam bellek kullanımını ondalık sayı sistemi ile gösterir.
- hex, toplam bellek kullanımını onaltılık sayı sistemi ile gösterir.

Bu bilgilere dayanarak, bir programın ROM ve RAM tüketimini hesaplamak mümkündür. ROM tüketimi .text ve .data bölümlerinin toplamına eşittir. RAM tüketimi ise .data ve .bss bölümlerinin toplamına eşittir.

$$\text{ROM} = \text{.text} + \text{.data}, \text{RAM} = \text{.data} + \text{.bss}$$

ROM tüketimi, programın kalıcı bellekte kapladığı alanı gösterir; bu, .text ve .data segmentlerinin toplamıdır. RAM tüketimi ise, programın çalışma zamanında ihtiyaç duyduğu bellek miktarını belirler; bu da .data ve .bss segmentlerinin toplamıdır. Bu hesaplamalar, bellek sınırlamaları olan gömülü sistemlerde, bellek kullanımını optimize etmek için kritiktir. Bu sayede, sistemin stabilitesi ve güvenilirliği artırılabilir, böylece donanım kaynakları daha verimli kullanılabilir.

2.6.2. Güç Yönetimi Teknikleri

Contiki OS’da güç yönetimi, enerji verimliliğini artırmak ve düşük güçlü gömülü sistemlerde pil ömrünü uzatmak için çeşitli teknikler ve araçlar aşağıda sunulmuştur.

Görev Döngüsü (Duty Cycling), cihazın aktif ve uyku modları arasında geçiş yapmasını sağlayarak enerji tüketimini azaltır. Contiki, düşük güçlü iletişim protokollerini desteklemek için duty cycling mekanizmalarını kullanmaktadır. Görev döngüsü, cihazın belirli zamanlarda veri gönderip almasını ve geri kalan zamanlarda uyku modunda kalmasını sağlar. Bu sayede, cihazların sürekli olarak aktif kalmaları engellenir ve enerji tüketimi minimize edilir. Cihazlar,

görev döngüsü sayesinde belirli aralıklarla kısa süreliğine aktif hale gelir, veri alışverişini gerçekleştirir ve ardından tekrar uyku moduna geçerler.

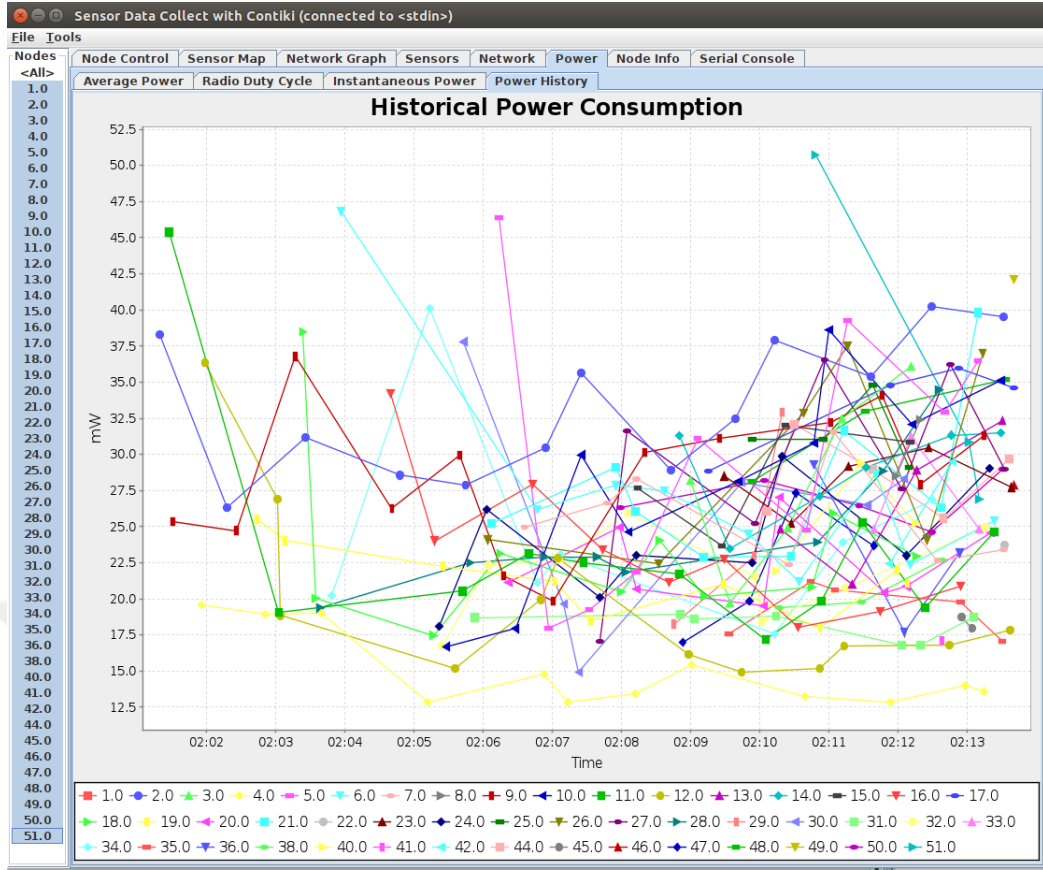
Düşük Güç Modu (Low Power Mode - LPM), radyo iletişimi sırasında enerji tasarrufu sağlamak için kullanılan bir diğer önemli tekniktir. LPM, cihazın belirli aralıklarla radyo kanalını kısa süreliğine dinlemesini ve yalnızca veri paketleri alındığında aktif hale gelmesini sağlar. Bu yöntem, sürekli radyo dinlemeye kıyasla enerji tüketimini önemli ölçüde azaltır. LPM, cihazların enerji verimliliğini artırırken aynı zamanda veri iletişiminin güvenilirliğini de korur. Radyo modülü, belirlenen aralıklarla aktif hale gelerek veri paketlerini dinler ve bu esnada enerji tasarrufu sağlanır. Veri paketi algılandığında, radyo modülü tam güçle çalışarak veri alışverişini gerçekleştirir.

Contiki, ayrıca güç yönetimi stratejilerini optimize etmek için yazılım ve donanım bileşenleri arasında koordinasyon sağlar. Bu koordinasyon, cihazların enerji tüketimini daha verimli bir şekilde yönetmelerine olanak tanır. Örneğin, sensör düğümleri, enerji tasarrufu sağlamak için uyku moduna geçmeden önce gerekli tüm işlemleri tamamlar ve ardından uyku moduna geçerler. Bu süreç, enerji tüketimini minimize ederken cihazların görevlerini etkin bir şekilde yerine getirmelerine yardımcı olur.

Contiki OS, çeşitli güç modlarını destekler:

- Aktif Mod: CPU ve radyo aktif durumdadır.
- Düşük Güç Modu: CPU düşük güç modundadır, ancak hızlı bir şekilde uyanabilir.
- Uyku Modu: CPU ve radyo minimum enerji tüketir, ancak uyanma süresi daha uzundur.

Powertrace, Contiki OS’de enerji tüketimini izlemek ve analiz etmek için kullanılan bir araçtır (Dunkels vd., 2011). Şekil 47, 50 düğümlü bir simülasyon için PowerTrace modülünü göstermektedir. Şekilde görüldüğü gibi, her bir düğümün enerji tüketimi zamana bağlı olarak izlenmektedir. Farklı renkler ve semboller, ağdaki farklı düğümlerin enerji tüketimini temsil eder. PowerTrace modülü, düğümlerin enerji tüketimindeki zamansal değişiklikleri detaylı bir şekilde raporlayarak enerji yönetim stratejilerinin etkinliğini değerlendirmeye yardımcı olur. Bu modül, ağdaki düğümlerin ortalama güç tüketimini, radyo görev döngüsünü ve anlık güç tüketimini analiz etme imkanı sağlar.



Şekil 47. 50 Düğümlü bir ağ için PowerTrace modülü

Energiest, Contiki'nin enerji tüketimini izlemek için kullandığı bir başka araçtır (Simha vd., 2020). Energiest, farklı bileşenlerin (CPU, radyo, vb.) enerji tüketimini zaman bazlı olarak izler. Aşağıda, tez kapsamında güç yönetimi modları için enerji tüketimi hesaplama örneği verilmiştir:

- CPU Aktif Modu: Aktif modda CPU'nun akım tüketimi I_{CPU} , bu modda geçen süre T_{CPU} ve voltaj değeri V 'dir. Enerji tüketimi $E_{CPU} = I_{CPU} \times V \times T_{CPU}$ olarak hesaplanır.
- Radyo Dinleme Modu: Radyo dinleme modunda akım tüketimi I_{Listen} , bu modda geçen süre T_{Listen} ve voltaj değeri V 'dir. Enerji tüketimi $E_{Listen} = I_{Listen} \times V \times T_{Listen}$ olarak hesaplanır.
- Radyo İletme Modu: Radyo iletme modunda akım tüketimi $I_{Transmit}$, bu modda geçen süre $T_{Transmit}$ ve voltaj değeri V olarak belirlenir. Enerji tüketimi $E_{Transmit} = I_{Transmit} \times V \times T_{Transmit}$ olarak hesaplanır.
- Düşük Güç Modu (LPM): LPM modunda akım tüketimi I_{LPM} , bu modda geçen süre T_{LPM} ve voltaj değeri V 'dir. Enerji tüketimi $E_{LPM} = I_{LPM} \times V \times T_{LPM}$ olarak hesaplanır.

Bu hesaplamalar, Powertrace ve Energiest mod  lleri tarafından toplanan verilerle ger  ekle  tirilir. Her bir mod i  in ayrı ayrı enerji t  ketimi hesaplanarak toplam enerji t  ketimi belirlenir. Bu bilgiler, g     y  netimi stratejilerinin etkinli  ini de  erlendirmek i  in kullanılır.



3. SENSÖR AĞLARI İÇİN GÜVENLİK GEREKSİNİMLERİ

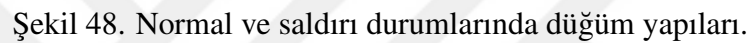
Sensör ağlarının güvenliği, topladıkları verilerin hassasiyeti ve kullanım alanlarının kritik doğası göz önüne alındığında büyük bir önem taşımaktadır. Bu ağlar, genellikle sağlık hizmetleri, askeri uygulamalar, çevresel izleme ve endüstriyel kontrol sistemleri gibi kritik alanlarda kullanılmaktadır. Bu tür uygulamalarda veri bütünlüğü, gizliliği ve doğruluğu hayati önem taşıyabilir. Bu nedenle, sensör ağlarının güvenliğinin sağlanması, bu ağların etkin ve güvenilir bir şekilde çalışabilmesi için zorunludur. Güvenlik tehditlerine karşı korunmak amacıyla, çeşitli güvenlik gereksinimlerinin eksiksiz bir şekilde karşılanması gerekmektedir. Bu gereksinimler, veri şifreleme, kimlik doğrulama, erişim kontrolü ve saldırı tespit sistemleri gibi çeşitli güvenlik önlemlerini içermektedir. Bu bölümde, sensör ağlarının güvenlik gereksinimleri ve bu gereksinimlerin nasıl karşılanacağı detaylı bir şekilde incelenmiştir.

3.1. Kimlik Doğrulama (Authentication)

Kimlik doğrulama, ağ güvenliğinde yetkisiz erişimlere karşı korunmak için kritik bir unsurdur. Bu mekanizma, sadece yetkili cihazların ağa katılmasını ve veri iletimini sağlar. Kimlik doğrulamanın eksik veya zayıf olduğu durumlarda, kötü niyetli saldırganlar ağa sızarak yetkisiz erişim sağlayabilir ve veri iletimini tehlikeye atabilir. Şekil 48’de gösterildiği gibi, yetkisiz düğümler ağ güvenliğini tehdit etmeye çalışmaktadır. Ancak, doğru kimlik doğrulama mekanizmaları sayesinde bu düğümlerin ağa erişimi engellenir, yalnızca yetkili düğümler güvenli bir şekilde veri iletimine katılabilir.

3.2. Veri Bütünlüğü (Data Integrity)

Veri bütünlüğü, iletilen ya da saklanan bilgilerin izinsiz müdahalelere karşı korunmasını ifade eder. Bu konsept, verilerin kaynağından hedefe bozulmadan ulaşmasını ve aktarım esnasında herhangi bir değişikliğe uğramamasını garanti altına alır. Bilgilerin güvenilirliği ve doğruluğu bakımından büyük öneme sahip olan veri bütünlüğü, özellikle hatalı ya da değiştirilmiş verilerin kullanılmasını engellemek açısından gereklidir. Kritik bilgilerin



Veri bütünlüğünü temin etmek amacıyla yaygın olarak kullanılan tekniklerden biri kriptografik özet fonksiyonlarıdır. Bu fonksiyonlar, verilerin sabit boyutlu bir özetini çıkararak, verilerin değiştirilip değiştirilmediğini tespit etmeye yardımcı olur. Eğer veriler üzerinde bir değişiklik yapılırsa, özet değeri farklılaşır ve bu da bütünlüğün ihlal edildiğini gösterir. Dijital imzalar ise verilerin bütünlüğünü ve kaynağını doğrulamak amacıyla kullanılır; bu sayede, verilerin değiştirilip değiştirilmediği ve kaynağının güvenilir olup olmadığı kontrol edilebilir.

Zaman damgaları, verilerin belirli bir zaman diliminde oluşturulduğunu ve gönderildiğini belirtir, bu da alıcıların verilerin geçerliliğini doğrulamalarını sağlar. Zaman damgası içeren paketler, belirli bir süre sonra geçersiz sayılır, bu da eski paketlerin yeniden

kullanılmasını önler. Ancak, zaman damgası yönteminin etkin bir şekilde kullanılabilmesi için ağdaki tüm cihazların senkronize olması gereklidir. Bu, ağdaki saatlerin doğru bir şekilde senkronize edilmesi gerektiği anlamına gelir; aksi takdirde zaman damgaları arasında uyumsuzluklar oluşabilir.

Sıralama numaraları, her veri paketine benzersiz bir numara atanmasını sağlayarak veri iletiminde bütünlük ve doğruluğu sağlar. Bu numaralar, alıcı cihazlar tarafından izlenir ve her yeni paketin numarası önceki numaradan daha büyük olmalıdır. Bu yöntem, herhangi bir paketin yeniden gönderilmesi durumunda alıcı cihazın bu paketin eski olduğunu anlamasını sağlar. Veri güncelliği, özellikle kritik zamanlı uygulamalar için önemlidir. Örneğin, gerçek zamanlı izleme sistemlerinde, sağlık izleme cihazlarında ve endüstriyel kontrol sistemlerinde verilerin güncel olması hayati önem taşır. IEEE 802.15.4 TSCH protokolü, zaman senkronizasyonu ve sıralama numaraları kullanarak güvenilir veri iletimi sağlar. TSCH protokolü, her verinin belirli zaman dilimlerinde gönderilmesini düzenler ve sıralama numaraları ile eski paketlerin yeniden iletilmesini engeller, bu da özellikle ağın güvenilirliğini ve performansını artırır.

3.4. Hizmet Bütünlüğü (Service Integrity)

Hizmet bütünlüğü, sensör ağının sunduğu hizmetlerin güvenilir ve kesintisiz bir şekilde çalışmasını garanti eder. Bu, hizmetlerin yetkisiz müdahalelere karşı korunmasını ve doğru bir şekilde işlenmesini sağlar. Hizmet bütünlüğünü sağlamak amacıyla, ağın çeşitli seviyelerinde güvenlik politikaları ve mekanizmaları uygulanabilir. Örneğin, hizmet kesintilerini tespit etmek ve önlemek için izleme sistemleri ve saldırı tespit sistemleri devreye alınabilir. Ayrıca, ağın yönlendirme protokolleri (örneğin, RPL) ve çizelgeleme algoritmaları (örneğin, 6P) sayesinde, hizmetlerin etkin bir şekilde yönetilmesi ve dağıtılması sağlanır. Bu yöntemler, ağın güvenilirliğini ve performansını artırarak, hizmetlerin kesintisiz sunulmasını mümkün kılar.

3.5. Senkronizasyon (Synchronization)

Senkronizasyon, sensör düğümlerinin zaman açısından uyumlu çalışmasını sağlar. Senkronizasyon, veri toplama ve iletilme süreçlerinin düzgün bir şekilde gerçekleştirilmesi için önemlidir. Zaman senkronizasyonu, düğümlerin veri toplama zamanlarını koordine

etmelerini ve böylece veri çatışmalarını ve enerji israfını önlemelerini sağlar. Senkronizasyonu sağlamak için ağda merkezi bir saat kaynağı veya dağıtık zaman senkronizasyon protokolleri kullanılabilir. 6TiSCH ağları, TSCH mekanizmasını kullanarak zaman senkronizasyonunu gerçekleştirir. Bu mekanizma, her düğümün belirli zaman dilimlerinde veri gönderme ve alma işlemlerini yapmasını sağlar, böylece düğümler arasında çakışma olmadan iletişim kurulur.

3.6. Erişilebilirlik (Availability)

Erişilebilirlik, ağ hizmetlerinin her zaman ulaşılabilir olmasını ve kesintisiz çalışmasını sağlar. Hizmetlerin sürekli sunulabilmesi için ağın, donanım arızaları ve hizmet reddi saldırıları gibi çeşitli tehditlere karşı dayanıklı olması gerekir. Bu dayanıklılığı sağlamak için yedekleme sistemleri, yük dengeleme teknikleri ve saldırı önleme mekanizmaları uygulanabilir. Bu sayede, ağın sürekli ve kesintisiz bir şekilde çalışması güvence altına alınır.

Güvenlik gereksinimleri, sensör ağlarının güvenli ve verimli çalışmasında kritik bir rol oynar. Her bir gereksinim, ağın farklı bir güvenlik yönünü ele alarak genel güvenlik seviyesini yükseltir. Sensör ağlarının bu güvenlik standartlarını karşılaması, ağın güvenilirliğini ve performansını artırarak farklı uygulama alanlarında etkin bir kullanım sağlar.

4. LİTERATÜR

Bu bölümde, tez kapsamında yayınlanan "A Decentralized Proxy-JRC Authentication System for Scalable IETF 6TiSCH Networks" ve "Lightweight Three-Factor Sensor and User Authentication for RPL-Based 6TiSCH Networks" makaleleri üzerine literatürde yapılan çalışmalar ele alınacaktır (Aydın vd., 2024). Kablosuz sensör ağları, düşük güçlü ve kayıplı ağlar olarak da bilinen enerji kısıtlı ve dinamik ortamlarda çalışır. Bu ağlarda kimlik doğrulama, yetkisiz erişimlerin önlenmesi ve veri güvenliğinin sağlanması açısından kritik bir öneme sahiptir. Kimlik doğrulama mekanizmaları, genellikle merkezi otoriteye dayalı yöntemler ve dağıtık yöntemler olarak ikiye ayrılır.

Merkezi otoriteye dayalı kimlik doğrulama yöntemleri, ağın güvenilirliğini ve yönetilebilirliğini artırırken, merkezi otoritenin bir arıza noktası oluşturması ve yüksek enerji tüketimi gibi dezavantajlara sahiptir. Bu dezavantajlar, özellikle enerji kısıtlı ve dinamik yapıya sahip sensör ağları için önemli bir sorun teşkil eder.

Dağıtık kimlik doğrulama yöntemleri ise merkezi otoritenin getirdiği bu dezavantajları ortadan kaldırarak, ağın güvenilirliğini ve ölçeklenebilirliğini artırır. Bu yöntemler, düğümlerin kendi aralarında işbirliği yaparak kimlik doğrulama işlemlerini gerçekleştirmelerine olanak tanır ve böylece enerji tüketimini azaltarak ağın genel performansını iyileştirir. Özellikle, IETF 6TiSCH protokolü gibi düşük güçlü ve kayıplı ağlarda kullanılan yöntemler, ağın dinamik yapısına uyum sağlayarak güvenli ve enerji verimli iletişim sağlar. Bu çalışmalarda, merkezi kimlik doğrulama mekanizmalarının getirdiği potansiyel güvenlik açıkları ve enerji tüketimi problemleri üzerinde durulmuş, bu sorunların üstesinden gelmek için dağıtık proxy-JRC tabanlı kimlik doğrulama sistemleri önerilmiştir. Dağıtık proxy-JRC sistemleri, ağdaki düğümlerin merkezi bir otoriteye bağlı kalmadan kendi aralarında güvenilir bir şekilde kimlik doğrulama ve anahtar yönetimi yapmalarını sağlamaktadır. Böylece, ağda tek bir arıza noktası oluşmasının önüne geçilmekte ve ağın ölçeklenebilirliği artırılmaktadır.

Kablosuz sensör ağlarındaki kimlik doğrulama süreci, ağ içinde iletişim kuran düğümlerin birbirlerini doğrulaması işlemidir. Bu süreç, düğümlerin kimlik bilgilerini doğrulayarak yetkisiz erişimleri engellemeyi ve ağın bütünlüğünü korumayı amaçlar. Özellikle dağıtık sistemlerde, düğümler, gelişmiş şifreleme teknikleri ve anahtar yönetimi protokolleri

kullanarak birbirlerini doğrular. Bu yöntemler, ağın güvenlik düzeyini artırırken, enerji verimliliğini ve genel performansını iyileştirir, böylece ağın dinamik ve değişken ortamlarda etkin bir şekilde işlemlerini sağlar. Kablosuz sensör ağları için kimlik doğrulama, sadece güvenlik sağlamakla kalmaz, aynı zamanda ağın sürdürülebilirliğini ve ölçeklenebilirliğini de destekler. Bu nedenle, ağdaki düğümler arasında güvenilir kimlik doğrulama mekanizmaları, bu tür ağların temel taşlarından biridir.

Yukarıda, kimlik doğrulama süreçlerinin kablosuz sensör ağları açısından taşıdığı kritik öneme ve bu süreçlerde kullanılan temel yöntemlere genel olarak değinilmiştir. Takip eden bölümlerde ise, bu yöntemler üzerine yapılmış araştırmalar ayrıntılı şekilde ele alınacaktır. Özellikle, merkezi ve dağıtık kimlik doğrulama mekanizmalarının avantajları, dezavantajları ve bu alandaki güncel çalışmalar üzerinde durulacaktır. İlk olarak, kablosuz sensör ağlarında merkezi ve dağıtık kimlik doğrulama yöntemleri üzerine yapılan çalışmalara odaklanılacak ve bu yöntemlerin karşılaştırılması yapılacaktır.

Ardından, kablosuz sensör ağlarında uygulanan gelişmiş kimlik doğrulama teknikleri üzerine gerçekleştirilen araştırmalar detaylandırılacak ve bu tekniklerin güvenlik, performans ve enerji verimliliği gibi kritik yönleri değerlendirilecektir. Böylece, kimlik doğrulamanın farklı boyutları ve kablosuz sensör ağlarına uygulanabilirliği hakkında kapsamlı bir analiz sunulmuş olacaktır.

4.1. Kablosuz Sensör Ağlarında Merkezi ve Dağıtık Kimlik Doğrulama Üzerine Yapılan Çalışmalar

Anand vd., 2023 yılında yazılım tanımlı ağlar (software defined networks - SDN) ile entegre edilen kablosuz sensör ağlarının güvenliğini artırmak amacıyla merkezi hafif bir kimlik doğrulama ve anahtar anlaşma protokolü önermişlerdir (Anand & Saifulla, 2023). Çalışma, modern ağlarda artan bağlı cihaz sayısının getirdiği zorluklara değinerek başlamakta ve sensör düğümlerinin kısıtlı kaynaklarla çalıştığı ortamlarda güvenli iletişim sağlanmasının önemini vurgulamaktadır. Yazarlar, ağ trafiğini ve kontrolünü yönetmek için SDN'den yararlanan, ağın artan talepleri performans darboğazları olmadan karşılayabilmesini sağlayan yeni bir protokol önermektedir. Hafif Kimlik Doğrulama ve Anahtar Anlaşma Protokolü (Lightweight Authentication and Key Agreement Protocol - LAKP) olarak bilinen bu protokol, hesaplama ve iletişim yükünü en aza indirirken güçlü güvenlik gereksinimi sağlamayı amaçlamaktadır. Makale, geleneksel güvenlik yöntemlerinin sensör ağları için uygun olmadığını, çünkü bu

yöntemlerin yüksek enerji tüketimine neden olduğunu belirtmektedir. Bunun yerine, LAKP, iletişimi verimli bir şekilde güvence altına almak için hash fonksiyonları, XOR işlemleri ve birleştirme işlemleri (concatenation operations) kullanmaktadır. Çalışma ayrıca, LAKP'yi mevcut protokollerle karşılaştıran bir performans analizi içermektedir. Analiz, LAKP'nin sadece daha iyi güvenlik sağlamakla kalmayıp aynı zamanda daha düşük hesaplama ve iletişim maliyetlerine sahip olduğunu göstermektedir. Bu durum, kaynak kısıtlamalarının önemli bir sorun olduğu SDN özellikli KSA'lar için LAKP'yi özellikle uygun kılmaktadır. Ek olarak, protokol, dinamik sensör düğümü ekleme ve kullanıcı kimlik bilgisi güncellemelerini içermekte olup, gerçek dünya senaryolarında uygulanabilirliğini artırmaktadır. Ancak, çalışmanın dezavantajları arasında SDN'nin merkezi yapısının potansiyel tek nokta arızası oluşturabileceği ve protokolün büyük ölçekli dağıtımlarda uygulanabilirliğinin test edilmesi gerektiği belirtilmektedir. Merkezi doğası, ağın belirli bir bileşeninin arızalanması durumunda tüm ağın işleyişinin etkilenmesine yol açabilir.

Yazarlar, düşük güçlü ve kayıplı ağlar için hafif ve verimli bir kimlik doğrulama mekanizması olan Nesnelerin İnterneti için Hafif Çapraz Alan Karşılıklı Kimlik Doğrulama Şeması (Lightweight Cross-Domain Mutual Identity Authentication Scheme for Internet of Things - LCDMA) önermektedir (Gong vd., 2023). Çalışma, mevcut kimlik doğrulama yöntemlerinin yüksek hesaplama maliyetleri ve güvenlik açıkları nedeniyle IoT ortamında uygulanabilir olmadığını belirtmektedir. Önerilen protokol, simetrik polinom kullanarak kimlik doğrulama işlemlerini gerçekleştirir. Ayrıca, makale Nesnelerin İnterneti-Güven Zinciri (Internet of Things-Trust Chain - IoT-TC) yapısını ele almaktadır. IoT-TC, IoT cihazları arasında güvenli veri iletimi ve kimlik doğrulama sağlamak için bir güven zinciri oluşturur ve merkezi otorite kullanarak bir kimlik doğrulama mekanizması sağlar. IoT-TC, düğümler arasında güvenli anahtar değişimi ve veri iletimi sağlar. Bu yapı, ağın güvenilirliğini ve yönetilebilirliğini artırırken, merkezi otoritenin bir arıza noktası oluşturması ve yüksek enerji tüketimi gibi dezavantajlara sahiptir. LCDMA yöntemi, yüksek enerji verimliliği sağlarken güvenlik açıklarına karşı dayanıklıdır. Ayrıca, güvenli anahtar anlaşması ve anonimlik koruması sunarak mobil IoT ortamında güvenli veri iletimi sağlar. Önerilen LCDMA protokolünün avantajları arasında düşük hesaplama ve iletişim maliyetleri, güvenli izlenebilirlik ve anonimlik koruması bulunmaktadır. Simetrik polinom kullanımı sayesinde, daha düşük enerji tüketimi ve hızlı kimlik doğrulama işlemleri gerçekleştirilir.

Ito, IoT destekli akıllı tarım izleme sistemleri için güvenli ve gizliliği koruyan hafif bir kimlik doğrulama ve anahtar değişim algoritması önermişlerdir (Ito vd., 2023). Bu algoritma, eliptik eğri kriptografisi ve hash fonksiyonu kullanarak çeşitli güvenlik

saldırılarına karşı koruma sağlar. Önerilen çerçeve, karşılıklı kimlik doğrulama ve anahtar değişimi için Burrows-Abadi-Needham (BAN) mantığı kullanılarak doğrulanmış ve Scyther aracı ile şifreleme güvenliği simülasyonu yapılmıştır (Hofer-Schmitz & Stojanović, 2020). Ayrıca, Rastgele Oracle Modeli (Random Oracle Model - ROR) kullanılarak sistemin güvenliği formelleştirilmiştir. Performans değerlendirmeleri, önerilen protokolün benzer protokollere kıyasla üstün güvenlik ve verimlilik sağladığını göstermektedir. Önerilen protokol, düşük enerji tüketimi ve hızlı kimlik doğrulama süreçleri ile kaynak kısıtlı cihazlar için uygundur. Ancak, merkezi yapısı nedeniyle büyük ölçekli dağıtımlarda uygulanabilirliği zordur. Çalışmanın sonuçları, önerilen protokolün tarım sensör ağlarında güvenlik ve gizlilik sağlama konusunda etkin olduğunu ve diğer mevcut protokollere göre daha az hesaplama ve iletişim yükü ile daha iyi güvenlik sağladığını göstermektedir. Bu, özellikle kaynak kısıtlı cihazlar için kritik bir avantajdır çünkü düşük hesaplama ve iletişim yükü, cihazların daha uzun süre çalışmasını ve daha az enerji tüketmesini sağlar.

Sai, enerji kısıtlı kablosuz iletişim cihazlarının güvenliğini sağlamak amacıyla hafif bir kimlik doğrulama çerçevesi önermektedir (Sai vd., 2023). Çalışmada, kablosuz sensör ağları için Eliptik Eğri Diffie-Hellman (Elliptic Curve Diffie-Hellman - ECDH) ve eliptik eğri tabanlı güvenli iletişim ve Eliptik Eğri Entegre Şifreleme Şeması (Elliptic Curve Integrated Encryption Scheme - ECIES) tabanlı kimlik doğrulama yöntemleri kullanılmaktadır. Çalışmada iki farklı senaryo ele alınmıştır: tek bir veri toplayıcı düğümün (sink node) olduğu senaryo ve tüm düğümlerin veri toplayıcı olduğu senaryo. Bu senaryolar üzerinde yapılan performans değerlendirmeleri, önerilen çerçevenin işlem süresi ve bellek kullanımı açısından üstün olduğunu göstermiştir. Merkezi kimlik doğrulama yönünden değerlendirildiğinde, bu çalışma merkezi bir kontrol mekanizmasına dayanarak güvenliğini sağlamaktadır. ECC ve ECDH yöntemleri kullanılarak merkezi bir otorite aracılığıyla güvenli iletişim sağlanmaktadır. Bu yaklaşım, merkezi yönetim sayesinde yüksek güvenlik ve koordinasyon sağlamaktadır. Ancak, merkezi bir otoritenin varlığı, tek bir hata noktası oluşturma riskini beraberinde getirebilir ve yüksek işlem gücü gereksinimi olabilir. Özellikle, her bir sensör düğümünün kendi başına kimlik doğrulama ve veri şifreleme işlemlerini gerçekleştirmesi, ağın genel dayanıklılığını artırır ve merkezi bir otoritenin arızalanması durumunda bile ağın işleyişini sürdürebilir.

Oliveira vd., merkezi kimlik doğrulama yöntemi üzerine bir çalışma yapmışlardır (Oliveira vd., 2016). Çalışmada, 6LoWPAN ağları için düğüm kontrol mekanizması önerilmiştir. Bu çözüm, yetkisiz düğümlerin ağda iletişim kurmasını engelleyerek güvenlik saldırılarını azaltmayı amaçlamaktadır. Merkezi bir sınır yönlendirici, düğümlerin kimlik

doğrularmasını ve yetkilendirilmesini gerçekleştirmektedir. AES simetrik anahtar algoritması kullanılarak düğümlerin kimlik doğrulaması ve veri çerçevelerinin bütünlüğü sağlanmaktadır. Ayrıca, önerilen çözüm laboratuvar ortamında test edilmiş ve doğrulanmıştır. Bu çalışmanın avantajları arasında, merkezi otorite sayesinde ağın yönetilebilirliğinin ve güvenliğinin artırılması yer almaktadır. AES simetrik anahtar algoritması ile düğümlerin kimlik doğrulaması ve veri bütünlüğü sağlanmıştır. Laboratuvar testlerinde düşük uygulama maliyeti ve yüksek güvenlik performansı gösterilmiştir. Merkezi yapının dinamik anahtar değiştirme kapasitesi sınırlıdır ve tüm düğümlerin yeniden yapılandırılması gerekebilir. Çalışma sonuçlarında, önerilen çözümün doğru çalıştığı ve güvenlik hedeflerini başarıyla gerçekleştirdiği gösterilmiştir. Ancak, çözümün daha da geliştirilmesi gerektiği belirtilmiştir. Özellikle, global anahtarın dinamik olarak değiştirilmesi ve RPL mesajlarının güvenliğinin sağlanması gibi alanlarda iyileştirmeler önerilmektedir.

Wang vd., hafif bir şifreleme ve mesaj doğrulama çerçevesi önererek kablosuz iletişimde güvenliği artırmayı hedeflemiştir (Wang vd., 2023). Yöntem, fiziksel katmanda sunulan zincir anahtarı (physical layer offered chain key - PHYLOCK) yapısını kullanarak bir defalık anahtarlar üretir ve bu anahtarlarla şifreleme yapar. Ayrıca, PHYLOCK-CRC tabanlı mesaj doğrulama şemaları geliştirerek, önerilen yöntemin güvenlik ve hata tespit yeteneklerini analiz etmişlerdir. Merkezi kimlik doğrulama açısından, LAKF'nin anahtarlama ve doğrulama işlemleri, fiziksel katmanda gerçekleşen anahtar üretimi ile sağlanmaktadır. Bu, merkezi bir otoriteye ihtiyaç duymadan güvenlik sağlar ve sistemin tek bir arıza noktası olmasını önler. Avantajları arasında, düşük hesaplama karmaşıklığı, enerji verimliliği ve yüksek güvenlik seviyeleri bulunmaktadır. Fiziksel katman anahtar üretimi ve CRC tabanlı mesaj doğrulama sayesinde, önerilen çerçeve düşük enerji tüketimi ile yüksek güvenlik sağlar. Ayrıca, tek seferlik anahtarlar kullanarak veri gizliliği ve bütünlüğü sağlanır. Dezavantajları ise, önerilen şemanın sadece belirli türdeki saldırılara karşı güvenli olması ve fiziksel katman anahtar üretiminin güvenliği, fiziksel ortamın koşullarına bağlı olmasıdır. Yüksek gürültü seviyeleri veya çoklu yol etkileri gibi faktörler, anahtar üretiminin güvenilirliğini etkileyebilir.

Berini vd., tarafından önerilen çalışma, merkezi bir kontrol mekanizmasına dayanmaktadır (Berini vd., 2023). Yazarlar, dronların ve kullanıcıların birbirlerini güvenli bir şekilde doğrulamalarını ve oturum anahtarını paylaşmalarını sağlayan verimli ve güvenli bir kimlik doğrulama protokolü geliştirmeyi amaçlamaktadır. HCALA (Hyperelliptic Curve-based Anonymous Lightweight Authentication Scheme for Internet of Drones), hipereliptik eğri kriptografisi (Hyperelliptic Curve Cryptography - HECC) kullanarak, düşük bant genişliği, enerji ve işlem kaynaklarına sahip dronlar için uygun bir çözüm sunar.

Çalışmanın avantajları arasında, önerilen protokolün bilinen aktif ve pasif saldırılara karşı dayanıklı olması ve diğer mevcut benzer şemalara kıyasla hesaplama, iletişim maliyetleri ve enerji tüketimi açısından daha verimli olması bulunmaktadır. Bununla birlikte, çalışma sonuçları, önerilen protokolün hesaplama ve iletişim yüklerini azalttığını, enerji verimliliğini artırdığını ve güvenlik yeteneklerini geliştirdiğini göstermiştir. Ayrıca, Otomatik Doğrulama Araçları Kümesi (Automated Validation of Internet Security Protocols and Applications - AVISPA) aracı kullanılarak yapılan resmi güvenlik doğrulaması, HCALA'nın çeşitli saldırılara karşı dayanıklılığını göstermiştir. Ancak, merkezi bir otorite olan yer istasyonu sunucusu (Ground Station Server - GSS) kullanılması, potansiyel bir tek arıza noktası oluşturabilir. Bu durum, GSS'nin arızalanması veya saldırıya uğraması durumunda ağır güvenliğini tehlikeye atabilir. Ayrıca, merkezi yapı nedeniyle ağır ölçeklenebilirliği sınırlanabilir ve büyük ölçekli dağıtımlar için ek önlemler gerekebilir.

Ashrif vd., tarafından önerilen SLAE6 (Secure and lightweight authenticated encryption scheme for 6lowpan networks), 6LoWPAN ağları için güvenli ve hafif kimlik doğrulama ve şifreleme şemasıdır (Ashrif vd., 2023a). SLAE6, kimlik doğrulama ve anahtar kurulum mekanizmalarını güvenli hale getirmek için hafif bir hash fonksiyonu ve kimlik doğrulama yöntemi (An Authenticated Encryption Algorithm) kullanır. SLAE6, diğer yöntemlerle karşılaştırıldığında daha az hesaplama ve iletişim maliyeti gerektirir. Merkezi kimlik doğrulama yöntemine dayanan bu makalede, tüm güvenlik ve kimlik doğrulama işlemleri merkezi bir sunucu tarafından yönetilmektedir. Bu yöntem, yüksek güvenlik ve yönetim kolaylığı sağlarken, merkezi otoritenin dezavantajlarına sahiptir. SLAE6, düşük hesaplama karmaşıklığı ve enerji tüketimi ile yüksek güvenlik sağlar. Özellikle kimlik doğrulama ve anahtar değişim süreçlerinde sağladığı düşük gecikme ve yüksek performans ile 6LoWPAN ağlarında verimli ve güvenli iletişimi mümkün kılar.

Adarbah, 6LoWPAN ağlarında güvenlik tehditlerine karşı güvenli bir eliptik eğri Diffie-Hellman tabanlı kimlik doğrulama protokolü geliştirmişlerdir (Adarbah vd., 2022). Tüm kimlik doğrulama işlemleri, ağır merkezindeki bir sunucu tarafından yönetilir. Bu durum, merkezi bir güvenlik otoritesine ihtiyaç duyar ve bu otorite, kimlik doğrulama ve anahtar yönetimini gerçekleştirir. Çalışmanın avantajları arasında yüksek güvenlik seviyeleri, düşük hesaplama maliyetleri ve düşük iletişim maliyetleri bulunmaktadır. Merkezi otorite sayesinde güvenlik yönetimi kolaylaşır ve güvenlik tehditlerine karşı etkili çözümler sunulur. Merkezi yapının getirdiği bu dezavantajlar, ağır genişletilebilirliğini ve esnekliğini kısıtlayabilir. Çalışma sonuçları, önerilen protokolün düşük hesaplama ve iletişim maliyetleri ile enerji verimliliğini artırdığını ve güvenlik yeteneklerini geliştirdiğini göstermektedir.

Usha, IIoT ağlarında cihaz kimlik doğrulaması ve oturum anahtarı oluşturma için yeni bir protokol önerdi (Jain vd., 2023). Bu protokol, XOR ve hash fonksiyonları gibi basit kriptografik teknikler kullanarak IIoT cihazlarının güvenliğini artırmayı amaçlamaktadır. Önerilen protokolde, IIoT cihazları kimlik doğrulama isteği içeren bir mesaj gönderir ve sunucu, gelecekteki iletişimler için oturum anahtarları ile yanıt verir. Protokolün güvenliği, AVISPA kullanılarak değerlendirilmiş ve düşük maliyetli, güvenli ve etkili bir kimlik doğrulama mekanizması sağladığı kanıtlanmıştır (Yogesh vd., 2020). Avantajları arasında düşük enerji tüketimi, basit uygulanabilirlik ve güçlü güvenlik önlemleri bulunmaktadır. Önerilen protokol, XOR ve hash işlemleri sayesinde düşük hesaplama yükü gerektirir ve oturum anahtarları kullanarak ileri ve geri gizlilik sağlar. Ayrıca, kimlik doğrulama sürecinde kullanılan zaman damgaları ve rastgele sayılar, DoS, yeniden oynatma, Sybil, mesaj gecikmesi ve gizli değer tahmini saldırılarına karşı koruma sağlar. Ancak, protokolün bazı potansiyel güvenlik açıkları bulunmaktadır. Örneğin, XOR işlemleri tek başına kullanıldığında, özellikle düşük entropili anahtarlar veya sabit veri ile çalışıyorsa, saldırganlar tarafından analiz edilebilir ve bu durum protokolün güvenliğini zayıflatır. Ayrıca, hash fonksiyonlarının zayıf seçilmesi durumunda, çarpışma saldırıları gibi kriptografik saldırılarla karşı karşıya kalınabilir. Farklı saldırı türleri arasında yan kanal saldırıları, kriptanaliz teknikleri ve düşük güçlü cihazların sınırlı hesaplama yeteneklerinden yararlanan zamanlama saldırıları sayılabilir.

Goyat vd., kablosuz multimedya sensör ağlarında merkezi olmayan kimlik doğrulama için PRIBADI adlı bir çözüm önermiştir (Goyat vd., 2023). Çalışma, özellikle akıllı şehir senaryolarında enerji ve depolama zorluklarını ele almak için geliştirilmiştir. PRIBADI, sensör düğümleri ve baz istasyonları arasındaki iletişimi doğrulamak için blokzincir tabanlı bir çerçeve kullanır ve güvenilir küme başlarını seçmek için ağırlık metriklerine dayalı olarak küme başları seçer. Bu küme başları, bilgileri Güvenilir Otorite'ye (Trusted Authority - TA) iletir ve TA, büyük verileri buluta depolamak için iletir. Ayrıca, TA kötü niyetli sensör düğümlerinin sertifikalarını iptal eder ve onları blokzincirden çıkarır. PPRIBADI'nin deneysel sonuçları, mevcut modellerden enerji verimliliği, veri iletim güvenliği ve depolama maliyeti açısından 31,7% daha iyi performans gösterdiğini ortaya koymaktadır. Merkezi olmayan kimlik doğrulama ile tüm güvenlik ve kimlik doğrulama işlemleri, ağırlıklı olarak blokzincir tarafından yönetilmektedir. Bu, merkezi bir güvenlik otoritesine olan bağımlılığı ortadan kaldırır ve sistemin tek bir arıza noktası olmasını önler. Avantajları arasında, merkezi olmayan yapının getirdiği yüksek güvenlik ve yönetim kolaylığı bulunmaktadır. Dezavantajları ise, blokzincir teknolojisinin yüksek enerji tüketimi ve veri depolama gereksinimleri yer almaktadır.

Khashan vd., IoT sistemleri için merkezi ve blokzincir tabanlı bir kimlik doğrulama mimarisini birleştirerek hibrit bir yaklaşım sunmaktadır (Khashan & Khafajah, 2023). Çalışma, IoT cihazları için zaman açısından verimli ve enerji tasarrufu sağlayan bir kimlik doğrulama yöntemi geliştirmeyi amaçlamaktadır. Bu mimari, merkezi kontrol mekanizmalarının yanı sıra blokzincir teknolojisinin güvenlik avantajlarını kullanarak IoT cihazlarının doğrulanmasını ve veri iletişimini güvence altına alır. Merkezi kimlik doğrulama, IoT cihazlarının ağın uç noktalarındaki kenar sunucular tarafından yönetilmesiyle gerçekleştirilirken, dağıtık kimlik doğrulama, farklı ve heterojen IoT sistemlerine ait cihazların blockchain ağı üzerinden doğrulanmasını sağlar. Çalışmanın avantajları arasında, düşük enerji tüketimi, yüksek güvenlik seviyeleri ve hızlı kimlik doğrulama süreçleri bulunmaktadır. Bununla birlikte, dezavantajları arasında, blockchain tabanlı doğrulama süreçlerinin yüksek hesaplama maliyetleri ve gecikme süreleri bulunabilir.

Kang vd., tıbbi sensör ağları için blokzincir tabanlı hafif bir kimlik doğrulama şeması sunmaktadır (Kang vd., 2024). Çalışmada, geleneksel merkezi kimlik doğrulama yöntemlerine kıyasla, blokzincir teknolojisinin merkezi arıza noktalarını ortadan kaldırarak veri bütünlüğünü ve güvenliğini artırdığı belirtilmiştir. Çalışma, merkezi kimlik doğrulama yöntemi ile başlarken, merkezi sunucuların güvenlik açıklarına dikkat çekmiş ve bu sorunları blokzincir teknolojisi ile çözmeyi hedeflemiştir. Blokzincirinin dağıtık yapısı sayesinde, merkezi bir otoriteye ihtiyaç duyulmadan güvenli ve verimli bir kimlik doğrulama sağlanmaktadır. Ayrıca, blokzincir tabanlı yapı sayesinde ağdaki paketlerin veri bütünlüğü ve gizliliği korunurken merkezi sunucuların yükü azaltılmıştır. Ancak, çalışmanın dezavantajları arasında blokzincir teknolojisinin getirdiği karmaşıklıklar ve yüksek işlem maliyetleri yer almaktadır.

Raza vd., hafif ve güvenli bir anahtar değişim mekanizması olan TinyIKE (Tiny Internet Key Exchange)'yi tanıtmaktadır (Raza & Magnússon, 2018). Bu mekanizma özellikle nesnelerin interneti cihazları için tasarlanmıştır. TinyIKE, IoT cihazlarının sınırlamalarını ele alarak, bu cihazların internetle güvenli bir şekilde iletişim kurmalarını sağlar. Çalışma, IKEv2 protokolünün hafif bir versiyonunu sunmakta ve çeşitli IoT protokollerinde güvenli anahtar kurulum ve yönetimini kolaylaştırmaktadır. TinyIKE, eliptik eğri kriptolojisi kullanarak düğüm kimlik doğrulamasını geliştirir ve ağ bütünlüğü ile güvenliğini artırır. Çalışmada, TinyIKE'nin uygulanabilirliğini değerlendirmek için gerçek IoT donanım platformlarında kapsamlı bir değerlendirme yapılmıştır. TinyIKE'nin avantajları arasında, birçok IoT protokolü için tek bir anahtar değişim çözümü sunması ve eliptik eğri kullanarak yüksek güvenlik sağlaması bulunmaktadır. Ancak, eliptik eğri kriptolojisi yüksek güvenlik sağlarken,

işlem yükünü artırarak cihaz performansını ve enerji tüketimini olumsuz etkileyebilir. Bu durum, kaynak kısıtlı ağlarda bu tür bir çözümün ölçeklenebilirliğini zorlaştırabilir.

Vučinić vd., tarafından oluşturulan "Constrained Join Protocol (CoJP) for 6TiSCH" standardı, 6TiSCH ağlarına güvenli bir şekilde yeni cihazların katılımını sağlayan bir protokolü tanımlamaktadır (Vućinić vd., 2021). Bu protokol, merkezi bir otorite olan Katılım Kayıt Yetkilisi/Koordinatör (Join Registrar/Coordinator - JRC) tarafından yönetilir. CoJP, yeni bir cihazın, "pledge" olarak adlandırılan, ağa katılım isteği göndermesi ve bu isteğin JRC tarafından işlenmesi sürecini içerir. Protokol, OSCORE tarafından sağlanan uçtan uca güvenlik özelliklerine dayanır ve güvenli kimlik doğrulama, yetkilendirme ve parametre dağıtımını sağlar. JRC, gerekli parametreleri sağlayarak cihazın ağa katılımını onaylar. Çalışma merkezi kimlik doğrulama yöntemine dayanmaktadır. Tüm kimlik doğrulama ve güvenlik işlemleri, ağın merkezinde bulunan JRC tarafından yönetilir. Bu, merkezi bir güvenlik otoritesine ihtiyaç duyar ve bu otorite, kimlik doğrulama ve anahtar yönetimi işlemlerinin güvenliğini sağlar. Çalışmanın sonuçlarına göre, CoJP protokolü 6TiSCH ağlarında yeni cihazların güvenli ve verimli bir şekilde katılımını sağlar. Protokol, ağın genel güvenliğini artırırken, enerji verimliliği ve ağ performansını optimize eder.

Aydın vd., tarafından IETF 6TiSCH protokolü için güvenli bir dağıtık kullanıcı kimlik doğrulama mekanizması önerildi (Aydın vd., 2018). Bu mekanizma, kimlik doğrulama anahtarlarının IoT ağının güvenilir düğümleri arasında dağıtılmasını sağlayarak, merkezi kimlik doğrulama sürecine kıyasla iletişim yükünü azaltmayı ve ağın enerji verimliliğini artırmayı amaçlamaktadır. Avantajları arasında, dağıtık kimlik doğrulama yaklaşımının iletişim yükünü azaltarak enerji tasarrufu sağlaması ve kimlik doğrulama sürecini ağın kenarlarında gerçekleştirmesi bulunmaktadır. Bu durum, özellikle büyük ölçekli IoT ağlarında performans iyileştirmeleri sağlayabilir. Ayrıca, güvenilir düğümler arasında kimlik doğrulama anahtarlarının dağıtılması, merkezi otoritenin tek hata noktası olma riskini azaltır. Dezavantajları ise, merkezi olmayan kimlik doğrulama düğümlerinin seçimi ve bu düğümlerin ağda ideal olarak konumlandırılması gibi problemlerin çözülmesi gerektiğidir. Çalışmanın sonuçları, önerilen dağıtık kimlik doğrulama mekanizmasının iletişim yükünü merkezi kimlik doğrulama mekanizmasına kıyasla 50%'ye kadar azaltabildiğini ve enerji tüketimini önemli ölçüde düşürdüğünü göstermektedir. Bu sonuçlar, dağıtık kimlik doğrulama yaklaşımının 6TiSCH tabanlı IoT ağlarında güvenliği ve performansı artırmada etkili bir çözüm olduğunu kanıtlamaktadır.

Haj-Hassan vd., Endüstriyel IoT ağları için merkezi ve dağıtık kimlik doğrulama yöntemlerini birleştiren yeni bir protokol sunmaktadır (Haj-Hassan vd., 2022). Çalışmada,

ağ koordinatörünün sertifikalar kullanarak yeni katılan düğümleri doğruladığı ve ardından yeni düğümün, Shamir sır paylaşımı temelli bir uzlaşma mekanizması ile ağ koordinatörünü doğruladığı bir yöntem önerilmektedir. Bu yöntem, ağ üyeleri arasında bir uzlaşma oluşturulmasını sağlayarak, yeni düğümlerin güvenli bir şekilde ağa katılmasını mümkün kılar. Dağıtık kimlik doğrulama ile güvenlik ve kimlik doğrulama işlemleri ağın farklı noktalarındaki düğümler arasında gerçekleştirilir. Bu, merkezi bir otoriteye olan bağımlılığı azaltır ve ağın tek bir arıza noktasına sahip olmasını önler. Avantajları arasında, yüksek güvenlik seviyeleri, düşük enerji tüketimi ve ağın esnekliği bulunur. Dezavantajları ise düğümler arasında koordinasyon gerektirmesi ve ağın karmaşıklığını artırmasıdır. Simülasyon sonuçları, teorik analizlerle tutarlı olup, protokolün etkinliğini ve saldırılara karşı dayanıklılığını ortaya koymaktadır.

Literatürde, 6TiSCH ağlarında kimlik doğrulama üzerine yapılan çalışmalar, genellikle merkezi otoriteye dayalı yöntemler ve bu yöntemlerin güvenlik ve performans üzerindeki etkileri üzerinde durmuştur. Ancak, merkezi otoritenin yükünün azaltılması ve ağın belirli noktalarına çekilerek dağıtık bir yapıya dönüştürülmesi, enerji tüketiminin düşürülmesi ve güvenlik açıklarının minimize edilmesi açısından önemli avantajlar sunmaktadır. Dağıtık kimlik doğrulama mekanizmaları, ağın çeşitli noktalarına yayılarak merkezi otoritenin yükünü azaltmakta ve ağ güvenliğini artırmaktadır. Bu yaklaşım, kimlik doğrulama ve anahtar güncelleme adımlarında çeşitli kazanımlar sağlamaktadır. Gerçekleştirilen literatür taramasında, bu alandaki mevcut çalışmalar detaylı bir şekilde incelenmiş ve dağıtık kimlik doğrulama yöntemlerinin endüstriyel IoT ağlarında sağladığı faydalar ortaya konulmuştur. Ayrıca, dağıtık kimlik doğrulama yöntemlerinin uygulanması, ağın ölçeklenebilirliğini artırarak, IIoT cihazlarının güvenli ve verimli bir şekilde entegrasyonunu sağlamaktadır.

Tablo 7 ve Tablo 8, literatürdeki merkezi ve dağıtık kimlik doğrulama yöntemlerine ilişkin kapsamlı bir analiz sunmaktadır. Bu tablolar, ilgili çalışmaların kullandığı kriptografi teknikleri, kimlik doğrulama türleri, avantajlar, dezavantajlar ve ölçeklenebilirlik açısından detaylı bir karşılaştırma sağlamaktadır. ECC, AES, XOR gibi yaygın kriptografik yöntemlerin yanı sıra, bazı çalışmalar blokzinciri ve PUF gibi daha gelişmiş tekniklere dayanan güvenlik çözümlerini içermektedir. Tabloların genel analizi göstermektedir ki, merkezi kimlik doğrulama sistemleri, yönetim kolaylığı ve yüksek güvenlik sağlarken, ölçeklenebilirlik ve tek arıza noktası gibi yapısal sorunlara eğilimlidirler. Özellikle, merkezi otoriteye bağımlı olan sistemler, geniş ölçekli dağıtımlarda güvenlik ve işlem maliyetlerinde ciddi zorluklarla karşılaşmaktadır. Bu zorluklar, yüksek hesaplama gereksinimleri ve işlem yükü ile daha da belirgin hale gelmektedir. Buna karşın, dağıtık kimlik doğrulama çözümleri,

merkezi otoritenin yükünü hafifleterek ağın güvenlik performansını artırmakta ve tek hata noktası sorununu ortadan kaldırmaktadır. Ancak bu tür çözümler, ağda düğümler arası koordinasyon gerektirdiği için karmaşıklık ve iletişim maliyetlerini artırabilmektedir. Ayrıca, tabloların ortaya koyduğu bir diğer önemli bulgu, ölçeklenebilirlik açısından sadece birkaç yöntemin verimli bir çözüm sunduğudur. Bu durum, özellikle IIoT ağlarında yaygın olarak kullanılabilecek daha ölçeklenebilir ve enerji verimli güvenlik çözümlerine ihtiyaç olduğunu ortaya koymaktadır.

Sonuç olarak, tablolar, mevcut yöntemlerin avantaj ve dezavantajlarını gözler önüne sermekte, ancak büyük ölçekli IIoT ağları için optimum güvenlik ve verimlilik dengesinin sağlanmasının hala bir araştırma alanı olarak öne çıktığını vurgulamaktadır.

4.2. Kablosuz Sensör Ağlarında Uygulanan Gelişmiş Kimlik Doğrulama Teknikleri Üzerine Yapılan Çalışmalar

Kimlik doğrulama, bir cihazın iddia ettiği cihaz olduğunu doğrulama sürecidir. Bu süreç, mesaj alıcısının, mesajın iddia edilen kaynağını (kaynak doğrulama) onaylamasını sağlar ve mesajın içeriğinin değiştirilmediğinden emin olur. Kimlik doğrulama, sensör ağlarında güvenli iletişimin temel yapı taşlarından biridir. Bu sayede, mesajların güvenilir kaynaklardan geldiği ve iletim sırasında değiştirilmediği doğrulanarak, ağın güvenliği ve bütünlüğü sağlanır.

Doğrulama süreci, genellikle iki ana bileşene dayanır: kimlik belirleme ve anahtar oluşturma. Kimlik belirleme, cihazın sahip olduğu halka açık bir tanımlayıcı veya başka bir benzersiz veri gibi unsurlara dayanır. Anahtar oluşturma ise, cihazlar arasında güvenli bir iletişim kanalı kurmak için gerekli olan şifreleme anahtarlarının paylaşılmasıdır. Bu süreç, mesajların şifrelenmesini ve yalnızca yetkili cihazlar tarafından okunabilmesini sağlar. Ayrıca, kimlik doğrulama ve anahtar oluşturma işlemleri, sensör ağlarının güvenliğini artırarak, ağın güvenilirliğini ve işleyişini korur. Sensör ağlarında kimlik doğrulama ve anahtar yönetimi, ağın dinamik yapısına ve enerji kısıtlamalarına uygun olarak tasarlanmalı ve uygulanmalıdır. Bu nedenle, hafif ve etkili kimlik doğrulama protokollerinin geliştirilmesi, sensör ağlarının güvenliğini sağlamak için kritik öneme sahiptir.

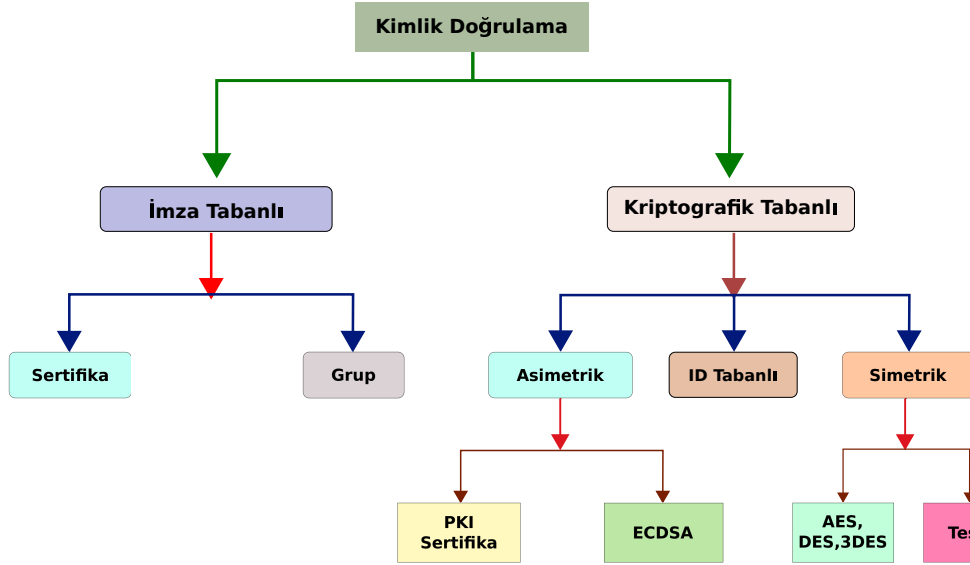
Şekil 49, kablosuz sensör ağlarında kullanılan kimlik doğrulama şemalarının sınıflandırılmasını göstermektedir. Kimlik doğrulama yöntemleri, imza tabanlı ve kriptografik tabanlı olmak üzere iki ana kategoriye ayrılmıştır. İmza tabanlı yöntemler, dijital imzalar ve sertifikalarla kimlik doğrulama sağlar (Velásquez vd., 2018). Bu kategoride sertifika

Tablo 7. Literatürde yer alan merkezi ve dağıtık kimlik doğrulama konulu çalışmalar

İlgili Çalışma	Şifreleme için kullanılan yöntem	Kimlik doğrulama türü	Avantaj	Dezavantaj	Ölçeklenebilirlik
(Anand & Saifulla, 2023)	ECC, XOR	Merkezi	Dinamik sensör düğümü ekleme, kimlik bilgilerini güncelleme	Yüksek hesaplama gereksinimi	Hayır
(Bei vd., 2023)	ECCDH, XOR, Hash	Merkezi	Anonimlik, izlenemezlik	Uygulama karmaşıklığı	Hayır
(Ittoo vd., 2023)	ECC, XOR, Hash	Merkezi	Dinamik sensör ekleme	Ölçeklenebilirlik sorunları	Hayır
(Sai vd., 2023)	ECC, ECIES	Merkezi	Hata toleranslı	ECC işlem yükü, Anahtar yönetiminde potansiyel karmaşıklık, Yüksek güç tüketimi	Hayır
(Oliveira vd., 2016)	AES, Hash	Merkezi	Verimli düğüm varlığı tespiti, İdari Düğüm Yetkilendirmesi	Ölçeklenebilirlik	Hayır
(Wang vd., 2023)	Hash, XOR, CRC	Merkezi	Hata algılama	Ölçeklenebilirlik	Hayır
(Berini vd., 2023)	HECC, XOR, Hash, Blok Zinciri	Merkeziyetsiz	Anonimlik, mahremiyet	Hesaplama ve iletişim yükü, Blok zinciri işlem maliyeti	Evet
(Ashrif 2023a)	ECC, XOR, Hash	Merkezi	Tekrar oynatmaya, Taklit saldırılarına, DoS saldırılarına	Ölçeklenebilirlik, Hesaplama Yükü	Hayır
(Adarbah 2022)	ECC, ECDH, AES, Hash	Merkezi	Dinamik anahtar üretimi	Uygulama karmaşıklığı, Ek işlem gücü	Hayır

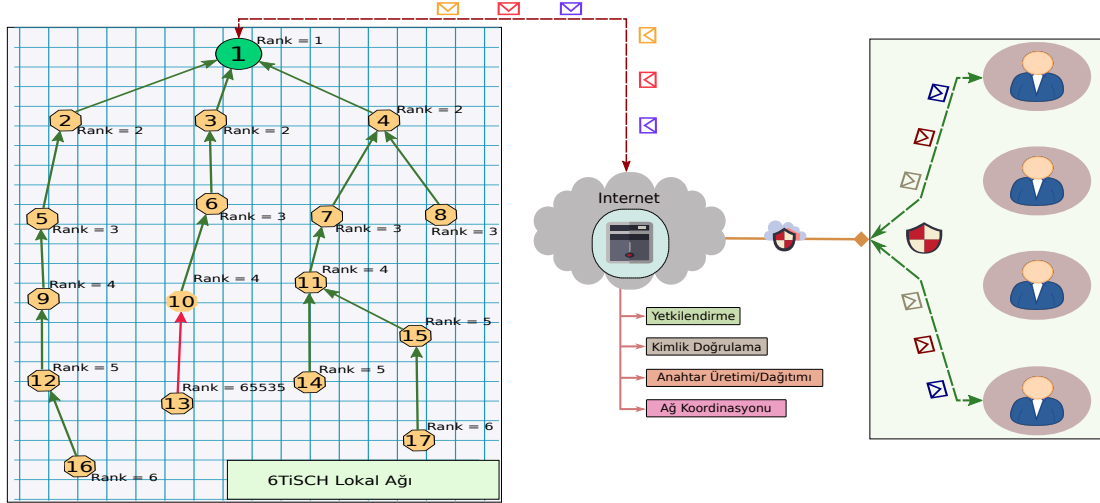
Tablo 8. Literatürde yer alan merkezi ve dağıtık kimlik doğrulama konulu çalışmalar-2

İlgili Çalışma	Şifreleme için kullanılan yöntem	Kimlik doğrulama türü	Avantaj	Dezavantaj	Ölçeklenebilirlik
(Jain vd., 2023)	XOR, Hash	Merkezi	Düşük yük, Enerji Tasarruflu	Ölçeklenebilirlik	Hayır
(Goyat vd., 2023)	Blok Zinciri, XOR, Hash	Dağıtık	Gelişmiş gizlilik koruması, Enerji verimliliği	Yüksek hesaplama, Blok zinciri kaynak tüketimi	Evet
(Khashan & Khafajah, 2023)	AES, ECC, Blok Zinciri	Merkezi	Takip edilemezlik	Blok zinciri maliyeti	Evet
(Kang vd., 2024)	Blok Zinciri, XOR, Hash, PUF	Merkeziyetsiz	PUF güvenliği	Anahtar yönetimindeki karmaşıklık, Blok zinciri maliyeti	Evet
(Raza & Magnússon, 2018)	ECC, AES, Hash, XOR	Merkezi	IoT cihazları arasında uçtan uca güvenlik	Ölçeklenebilirlik ve esneklikten yoksundur	Hayır
(Vučinić vd., 2021)	AES, Hash, XOR	Merkezi	İlk standart	Düğüm güvenliği, ağ ölçeklenebilirliği	Hayır
(Aydin vd., 2018)	AES, XOR, Hash	Merkezi	Düşük hesaplama maliyetine sahiptir	Anahtar dağıtımı zorluğu	Hayır
(Haj-Hassan vd., 2022)	AES, ECC, Hash, XOR	Merkezi	Dinamik sensör düğümü ekleme, Düşük hesaplama maliyeti	Yüksek hesaplama gereksinimleri, Ölçeklenebilirlik	Hayır
(Thakur 2024)	AES, Hash	Merkezi	Yüksek güvenlik ve verimlilik, Minimum yük	Kullanıcı, ağ geçidi ve sensör düğümü kimliğine karşı saldırılara açık	Evet



Şekil 49. KSA'da kimlik doğrulama şemalarının sınıflandırılması (Sheikh vd., 2019)

tabanlı ve grup imzası gibi alt yöntemler yer alır. Sertifika tabanlı yöntemler PKI (Public Key Infrastructure - Açık Anahtar Altyapısı) kullanırken, grup imzası ise grup içindeki kullanıcıların kimliklerini korur. Kriptografik tabanlı yöntemler ise asimetrik, ID tabanlı ve simetrik olarak sınıflandırılmıştır. Asimetrik şifreleme, PKI sertifikası ve ECDSA gibi teknolojilerle iki anahtar kullanarak kimlik doğrulama yapar. ID tabanlı yöntemler ise kimlik bilgilerine dayanarak daha az anahtar yönetimi gerektirir. Simetrik şifreleme ise AES, DES ve 3DES gibi algoritmaları kullanır. Ayrıca, TESLA gibi zaman tabanlı yöntemler, düşük gecikmeli güvenli iletişim sağlar. Bu sınıflandırma, farklı güvenlik ihtiyaçlarına göre kimlik doğrulama yöntemlerinin avantaj ve dezavantajlarını ortaya koymaktadır. Bu çalışmanın amacı, ağa dahil olacak yeni cihazların kimliğini doğrulamak ve yetkilendirmek için güvenli bir yöntem geliştirmektir. Ağ dahil olan cihazların güvenilir ve güvenli bir şekilde eklenmesi sağlanırken, yetkisiz ve kötü niyetli cihazların erişiminin engellenmesi de hedeflenmektedir. Bu doğrultuda, yalnızca yetkili cihazların ağa katılmasına izin verilmekte, ağın güvenliği korunmaktadır. Şekil 50, ağ üzerinde yer alan cihazlar ile son kullanıcı arasında gerçekleşen iletişim yapısını detaylı bir şekilde göstermektedir. Sol tarafta, 6TiSCH lokal ağındaki cihazlar, birbirleriyle haberleşme sırasındaki hiyerarşik yapıları ile gösterilmektedir. Cihazlar, ağdaki sıralamalarına (Rank) göre organize edilmiş olup, en üstte kök cihaz bulunmaktadır. Kök cihaz, kimlik doğrulama ve yetkilendirme işlemleri için internet üzerinden merkezi bir sunucuya bağlanmaktadır. Merkezi sunucu, ağ üzerindeki yetkilendirme, kimlik doğrulama, anahtar üretimi ve dağıtımını gibi kritik işlevleri yerine getirir. Ayrıca, ağ koordinasyonu da



Şekil 50. Ağ üzerinde yer alan sensör cihazları ile son kullanıcı arasındaki haberleşme yapısı

merkezi sunucu üzerinden sağlanır. Sağ tarafta ise son kullanıcılar, ağ üzerinden veri almakta ve işlemlerini gerçekleştirmektedirler. Kötü niyetli cihazların ağa sızma girişimleri, merkezi sunucu tarafından tespit edilerek engellenir. Böylece, ağ sadece yetkili cihazlara izin vermekte ve veri akışının güvenli bir şekilde sürdürülmesini sağlamaktadır.

IIoT alanında, kimlik doğrulama, düşük güçlü ve kayıplı ağlarda kullanıcı kimlik doğrulaması ve veri güvenliğini sağlamada önemli bir rol oynar (Szymoniak, 2024). Sensör verilerini güvenli bir şekilde merkezi iletişim noktalarına iletmek, bu tür ağlarda büyük bir zorluk teşkil eder. Geleneksel kimlik doğrulama yöntemleri ve güvenlik protokolleri, kaynak kısıtlamaları nedeniyle uygulanabilir olmayabilir. Bu nedenle, IIoT uygulamalarında güvenlik sorunlarını ele almak için özel kullanıcı kimlik doğrulama yöntemleri ve anahtar anlaşma şemaları kritik öneme sahiptir (Lee vd., 2023). Bu alandaki literatür taramaları, IIoT ve kaynak kısıtlı ağlardaki güvenlik sorunları hakkında değerli bilgiler sunar ve güvenli veri iletişimi, kullanıcı/sensör kimlik doğrulama yöntemleri ve anahtar anlaşma protokolleri gibi konuları kapsar.

Turkanović vd., enerji verimliliği, sağlam güvenlik ve minimal hesaplama yükü hedefleyen IoT odaklı bir kullanıcı kimlik doğrulama şeması sunmaktadır (Turkanović vd., 2014). Ancak, bu şemanın çevrimdışı şifre tahmin saldırılarına, sahtecilik saldırılarına ve şifre güncelleme işlevselliğinde zayıflıklara karşı savunmasızlık gibi güvenlik açıkları bulunmaktadır. Hu ve arkadaşları, hastalar için donanım ve yazılım bileşenlerini birleştiren gerçek zamanlı bir sağlık izleme sistemi önermektedir (Hu vd., 2007). Sistem düşük maliyetli ve düşük güçlü sensörlere vurgu yapsa da, sağlam kullanıcı kimlik doğrulamasından yoksundur ve bu durum yetkisiz erişim ve olası veri ihlallerine karşı savunmasız hale

getirmektedir. Das, kablosuz sensör ağlarına dayalı sağlık uygulamaları için iki faktörlü bir kimlik doğrulama şeması sunmaktadır (Das, 2009). Bu şema çeşitli saldırılara karşı direnç gösterse de, hizmet reddi saldırıları ve düğüm ele geçirme saldırılarına karşı yeterli güvenliği sağlayamamaktadır. Bae vd., çoklu ağ geçitli IoT ortamlarında hesaplama ve iletişim maliyetlerini azaltmak için akıllı kartlara dayalı güvenli bir kimlik doğrulama protokolü önermektedir (Bae & Kwak, 2020). Ancak, bu protokoller sahtecilik saldırılarına, ağ geçidi sahtekarlığına, izlenebilirliğe ve oturum anahtarı sızıntısına karşı savunmasızdır. Ayrıca, güvenli kimlik doğrulama ve anonimliği sağlamada yetersiz kalmaktadırlar.

Challa vd., tarafından kablosuz sağlık sensör ağları için üç faktörlü kullanıcı kimlik doğrulama ve anahtar uzlaşısı protokolü önermektedir (Challa vd., 2018). Bu protokol, mevcut protokollerdeki güvenlik açıklarını gidermeyi amaçlamakta ve kablosuz sağlık uygulamaları için geliştirilmiş bir çözüm sunmaktadır. Protokol güvenli kimlik doğrulama sağlasa da, eliptik eğri kriptografisinin uygulanma karmaşıklığı ve hesaplama gereksinimleri, kablosuz sağlık sensör ağlarında kullanılan kaynak kısıtlı cihazlarda zorluklar oluşturmaktadır. Wu vd., çoklu ağ geçitli kablosuz tıbbi sensör ağlarında sağlık sistemleri için hafif bir kimlik doğrulama protokolü önermektedir (Wu vd., 2018). Protokol, hasta takibi, içerden saldırılar ve çevrimdışı tahmin saldırıları gibi potansiyel kötü niyetli saldırıları hafifletmek üzere tasarlanmıştır. Yazarlar, çoklu ağ geçitli ortamların kablosuz sağlık sensörlerini kullanarak hastaların sağlık verilerini toplamak için uygun olduğunu göstermektedir. Muhal vd., IoT cihazları için uç cihazlar ile ağ geçitleri arasında karşılıklı kimlik doğrulama sağlayan bir kimlik doğrulama şeması önermektedir (Muhal vd., 2018). Bu şema ayrıca, akıllı telefonlar veya giyilebilir cihazlar aracılığıyla kullanıcı kimlik doğrulamasına olanak tanıyarak, oturum anahtarı ile güvenli iletişim kurmaktadır. Zaman damgaları, tekrar saldırılarına karşı koruma sağlar ve güvenliği artırır. Ancak, şemanın yüksek işlem gerektiren şifre çözme gereksinimleri gerçek dünya uygulamaları için pratik bir çözüm değildir. Kablosuz sensör ağları için eliptik eğri tabanlı hafif karşılıklı kimlik doğrulama ve anahtar anlaşma protokolü üzerine bir çalışma (Fariss vd., 2022)'de yapılmıştır. Çalışma, IoT ekosistemindeki nesneler arasında güvenli iletişimi sağlamak ve sınırlı kaynaklara sahip sensör düğümlerinin kimlik doğrulamasını kolaylaştırmayı amaçlamaktadır. Çalışmanın avantajları, PUF ve ECC kullanarak daha yüksek güvenlik sağlama, akıllı kart iptali ve klonlama saldırılarına karşı dirençli olmasıdır. Kullanıcının şifresi ve biyometrik bilgilerini kullanan kablosuz sağlık sensör ağları için verimli bir üç faktörlü kimlik doğrulama protokolü Saini vd., tarafından önerilmektedir (Saini vd., 2024). Protokolde, sensör cihazında depolanan verilerin güvenliğini sağlamak için ECC ve karma fonksiyonlar kullanılır. Protokolün doğruluğunu ve güvenliğini analiz etmek için BAN

ve ProVerif araçları kullanılmıştır. Ancak, hekimlerin sensör cihazına erişiminin doğrulanması ve hem oturum yöneticisi hem de ağ geçidi varlıklarının yetkilendirilmesi çalışmada eksiktir. Bu eksiklikler, protokolün genel güvenilirliği ve kullanılabilirliği açısından önemlidir.

Aydın vd., RPL tabanlı 6TiSCH ağları için enerji verimli bir kimlik doğrulama protokolü önermektedir (Aydın vd., 2018). Yöntemleri, kaynak kısıtlı cihazlarda kimlik doğrulama için AES şifrelemesi kullanarak sensör cihazlarını yönlendirici aracılığıyla merkezi bir sunucuya kimlik doğrulaması yapmaya odaklanmaktadır. Ancak, çalışma iç ağ saldırılarına karşı bir çözüm sunmamakta ve anahtar değişim mekanizması sağlamamaktadır. Hamidi vd., çeşitli biyometrik tekniklerin akıllı sağlık uygulamaları için uygulanabilirliğini analiz ettikleri güvenli bir cihaz kimlik doğrulama yaklaşımı tanıtmaktadır (Hamidi, 2019). Yazarlar, biyometrik özelliklere dayalı kimlik doğrulama gereksinimlerini ayrıntılı olarak açıklamaktadır. Fakat, çalışmaları kullanıcı taklit saldırıları gibi güvenlik açıklarını ele almamaktadır.

IoT için ECC kullanan yeni bir anonim kimlik doğrulama şeması sunulmaktadır (Durairaj & Muthuramalingam, 2018). Protokol, kayıt ve kimlik doğrulama olmak üzere iki ana aşamadan oluşmaktadır. Yazarlar, farklı boyutlardaki anahtarlar için şifreleme ve şifre çözme sürelerine odaklanarak ECC ve RSA arasında karşılaştırmalı bir analiz yapmaktadır. Ancak, kaynak kısıtlı cihazlar için yüksek şifreleme maliyeti, önerilen yaklaşımın pratikte uygulanabilirliğini engellemektedir. Mukhopadhyay vd., tarafından önerilen diğer bir donanım tabanlı yaklaşım, IoT cihazlarını kimlik doğrulamak için parmak izi ve PUF kullanmaktadır (Mukhopadhyay, 2016). Araştırma, PUF davranışını simüle etmek için bir yazılım modeli oluşturulması sonucunda makine öğrenimi tabanlı saldırıları incelemektedir. Ancak, ağ içindeki kötü niyetli saldırganları ele almak için bir çözüm önerilmemektedir. Lu vd., Das (Das, 2009) tarafından önerilen güvenlik şemasının güvenlik açıklarını ortaya koymaktadır; önerilen şema, üç faktörlü güvenlik, kullanıcı anonimliğini ve kullanıcı taklit saldırılarına karşı koruma sağlamamaktadır. Şemanın bilinen saldırılara karşı güvenli olduğunu iddia etseler de, güçlü oturum anahtarı güvenliği sağlamamakta ve bu da kaynak kısıtlı cihazlardan oluşan ağlar için uygunluğunu azaltmaktadır.

Vijaykumar vd., kaynak kısıtlı IoT cihazları için RFID teknolojisi kullanarak yeni bir hafif donanım kimlik doğrulama protokolü önermektedir (Vijaykumar vd., 2024). Protokol, kaynak yoğun veya hesaplama açısından pahalı olmayan güvenli karşılıklı kimlik doğrulama ihtiyacını ele almaktadır. Farklı rastgele kontrol girdileri için farklı mantıksal çıktılar üreterek güvenliği artıran çok işlevli dijital mantık tabanlı bir kodlayıcı mimarisi kullanmaktadır. Deneysel sonuçlar, alan, gecikme ve güç tüketiminde iyileştirmeler ile hafif uygulamalar için

uygunluğunu göstermektedir. Protokolün güvenliği BAN mantığı kullanılarak resmi olarak doğrulanmış ve Contiki OS kullanılarak gerçek zamanlı testleri gerçekleştirilmiştir.

Yazarlar, basit düzeyde ECC kullanan medikal nesnelerin interneti için gizlilik koruyucu dört faktörlü bir kimlik doğrulama protokolü sunmaktadır (Guo vd., 2024). Bu protokol, Nesnelerin İnterneti için Medikal Teknolojiler (Internet of Medical Things - IoMT) uygulamalarında güvenlik ve gizliliği artırmak için şifreler, akıllı kartlar, biyometrik özellikler ve PUF'u birleştirmektedir. Protokol, sistem kurulumu, kayıt, oturum açma, karşılıklı kimlik doğrulama, şifre değiştirme ve kullanıcı iptali olmak üzere altı aşamaya ayrılmaktadır. Yazarlar, önerilen protokolün güvenliğini BAN mantığı ve ProVerif (Rao & Prema, 2021) aracı kullanarak doğrulamakta ve güvenli karşılıklı kimlik doğrulama sağlama kabiliyetini göstermektedir. Ancak, ECC ile ilgili hesaplama maliyeti nedeniyle bu protokolün kaynak kısıtlı IoMT cihazlarında uygulanması zorluklar yaratabilir. Ayrıca, protokol birçok güvenlik endişesini ele alırken, kullanıcı deneyiminin pratik yönlerini ve gerçek zamanlı uygulamalardaki potansiyel gecikmeyi dikkate almamaktadır.

Huang vd., ECC tabanlı üç faktörlü kimlik doğrulama ve anahtar anlaşma şeması önermektedir. Önerilen yöntem, kablosuz sensör ağları için kullanıcılar, ağ geçitleri ve sensörler arasında kimlik doğrulama ve güvenli oturum anahtarı anlaşmasını gerçekleştirmek üzere biyometrik, akıllı kart ve şifre kimlik doğrulama teknolojilerini birleştirmektedir (Huang, 2024). Mekanizma, mevcut şemaların çeşitli güvenlik açıklarını ele almayı ve saldırılara karşı yüksek güvenlik sağlamayı amaçlamaktadır. Ancak, çalışma, özellikle kaynak kısıtlı cihazlar için yüksek hesaplama maliyetleri nedeniyle pratik uygulamalarda zorluklarla karşılaşabilir.

Bu bölümde sunulan çalışmalar, düğümlerin ağa nasıl güvenli bir şekilde entegre edildiğini veya ağ geçitlerinin bu cihazları nasıl doğruladığını tam olarak açıklığa kavuşturamamaktadır. Ayrıca, önerilen çalışmaların şifre tahmin, taklit ve kullanıcı izlenebilirliği saldırılarına karşı yeterli koruma sağlamadığı gözlemlenmiştir. Genel olarak literatür, sınırlı kaynaklara sahip cihazların yetkilendirme ve kimlik doğrulama süreçlerini tam anlamıyla ele almamaktadır. Ayrıca, çalışmalar genellikle pratik uygulamalarda gerekli enerji tasarrufunu yeterince dikkate almamakta ve bu da kaynak kısıtlı ağlarda istenen enerji verimliliğinin sağlanamamasına yol açmaktadır. Tezde önerilen mekanizma, sensör düğümlerinin ağa güvenli bir şekilde entegre edilmesini ve bu düğümlere güvenli kullanıcı erişimi sağlamayı amaçlamaktadır. Çalışmanın önemli bir yönü, kaynak kısıtlı cihazların gerçekçi testlerini bilgisayar ortamında mümkün kılan COOJA simülatörünün kullanılmasıdır (Osterlind vd., 2006). COOJA ortamında ağ senaryolarını simüle ederek, çalışma, gerçek dünya koşullarına yakın sonuçlar üretmeyi amaçlamaktadır. Kimlik doğrulama ve anahtar

yönetim mekanizması, sınırlı kaynaklara sahip sensör düğümleri için tasarlanmış bir gömülü işletim sistemi olan Contiki OS üzerinde çalışan Exp5438 cihazlarında uygulanmaktadır (Dunkels vd., 2004; Instruments, 2024). Önerilen şema, karşılıklı kimlik doğrulama ve yeniden saldırı zorluklarını ele alarak güvenlik gereksinimlerini karşılamaktadır. Performans ve güvenlik analiz sonuçları, çalışmanın 6TiSCH tabanlı IIoT sistemlerine uygulanabileceğini göstermektedir.



5. YAPILAN ÇALIŞMALAR

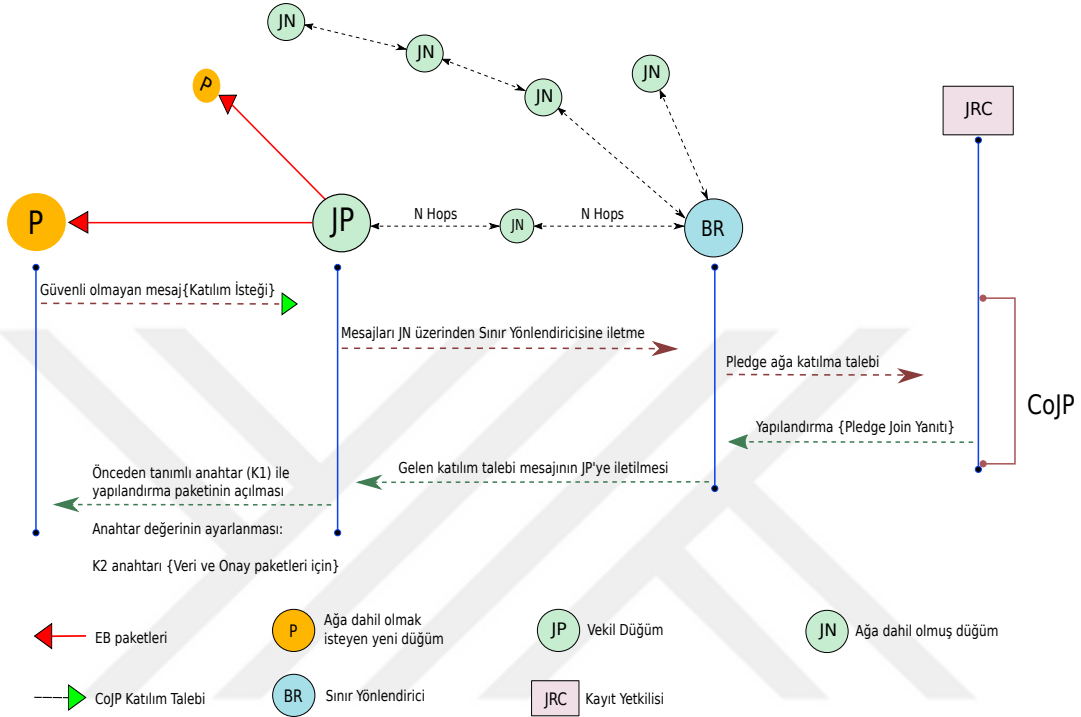
Literatürdeki mevcut çalışmalar kapsamlı bir şekilde incelenmiş ve Literatür başlığı altında bu alanda gerçekleştirilen çalışmalara ilişkin ayrıntılı bir karşılaştırma sunulmuştur. Tez kapsamında geliştirilen yöntemler, ölçeklenebilir IETF 6TiSCH ağları için merkezi Olmayan Proxy-JRC kimlik doğrulama sistemi başlığı altında detaylandırılmış olup, bu yöntem, yeni düğümlerin güvenli bir şekilde 6TiSCH ağına entegre edilmesine olanak tanımaktadır. Ayrıca, merkezi kimlik doğrulama yaklaşımlarına alternatif olarak önerilen dağıtık Proxy-JRC kimlik doğrulama noktalarının belirlenmesi süreçleri ayrıntılı bir şekilde ele alınmıştır. Son olarak, IETF 6TiSCH ağları için geliştirilen hafif üç faktörlü sensör ve kullanıcı kimlik doğrulama mekanizması kapsamında, sensör düğümleri ile kullanıcılar arasındaki güvenli veri iletişiminin sağlandığı gösterilmiştir.

5.1. Ölçeklenebilir IETF 6TiSCH Ağları için Merkezi Olmayan Proxy-JRC Kimlik Doğrulama Sistemi

6TiSCH ağına, yeni düğümlerin ağ koordinatörü tarafından kabul edilmek ve bağlantı düzeyinde güvenlik kimlik bilgilerini almak için katılma aşamasından geçmesi gerekir. CoJP, yeni düğümlerin 6TiSCH ağına katılması için gereken parametrelerin dağıtımını gerçekleştirir. CoJP’de, katılma aşamasında üç varlık ana rol oynar. (1) JRC ağı yönetmekten ve yeni düğümlere erişim vermekten sorumlu varlıktır. (2) Pledge, ağa katılmak isteyen düğümdür. (3) JP, halihazırda ağda bulunan ve rehin ile JRC arasındaki alışverişlerde aracılık yapan bir düğümdür. Şekil 51, Pledge’nin ağa dahil olma sürecini göstermektedir. CoJP’de katılma aşamasında üç ana bileşen rol oynar:

1. Join Registrar/Coordinator (JRC): Ağın yönetiminden sorumlu olan ve yeni düğümlere erişim izni veren merkezi bir birimdir. JRC, katılım taleplerini alır ve değerlendirir, gerekli güvenlik parametrelerini sağlar.
2. Pledge: Ağa katılmak isteyen yeni düğümdür. Pledge, ağa katılmak için bir dizi mesaj gönderir ve yanıt alır.

3. Join Proxy (JP): Ağa halihazırda bağlı olan ve pledge ile JRC arasındaki mesaj alışverişine aracılık eden düğümdür. JP, mesajları güvenli bir şekilde ileterek sürecin tamamlanmasına katkı sağlar. Ayrıca, katılmış düğüm (Joined Node - JN) olarak da adlandırılır.



Şekil 51. CoJP ile ağa dahil olma süreci

Bu aşama sırasında farklı türde mesajlar kullanılır:

- EB: JP'ler tarafından düzenli olarak shared slot/lar kanalları üzerinden gönderilir ve pledge'lerin ağı keşfetmelerine ve katılmalarına davet eder. EB mesajları, ağın kimlik bilgilerini ve zamanlama bilgilerini içerir.
- Katılım Talebi (Join Request) mesajları: Pledge tarafından JP aracılığıyla JRC'ye gönderilir ve ağa katılmak için talep edilen rolü ve ağın tanımlayıcısını içerir. Ayrıca, pledge'in kimliğini doğrulamak için gerekli bilgileri, şifreleme anahtarlarını ve güvenlik kimlik bilgilerini sağlar. Bu bilgiler, JRC'nin pledge'in kimliğini doğrulamasına ve ağın güvenliğine uygun olup olmadığını belirlemesine olanak tanır. Mesajda ayrıca, pledge'in ağ hizmetlerine ve kaynaklarına erişim talepleri belirtilir, bu da pledge'in ağ içinde üstleneceği görevleri ve kullanacağı kaynakları tanımlar.
- Katılım Yanıtı (Join Response) mesajları, JRC'den JP aracılığıyla pledge'e gönderilir ve pledge'in tam işlevsel bir ağ düğümü olması için gereken farklı parametreleri içerir.

Bu parametreler arasında bağlantı katmanı anahtarları, ağ kimlik bilgileri, güvenlik yapılandırmaları, veri iletimi için gerekli zamanlama bilgileri ve diğer ağ ayarları bulunur. Ayrıca, bu mesajlar, pledge'in ağ içinde doğru bir şekilde çalışabilmesi için gereken belirli frekansları ve zaman dilimlerini de sağlar. Katılım Yanıtı mesajı, pledge'in ağ içindeki yerini ve rolünü belirleyen bilgileri de içerir, bu da düğümün hangi görevleri üstleneceğini ve hangi düğümlerle iletişim kuracağını belirler. Bu kapsamlı bilgi seti, pledge'in ağ kaynaklarını etkin bir şekilde kullanmasını ve güvenilir bir şekilde veri iletimi yapmasını sağlar.

Katılma süreci, belirli adımları takip eder:

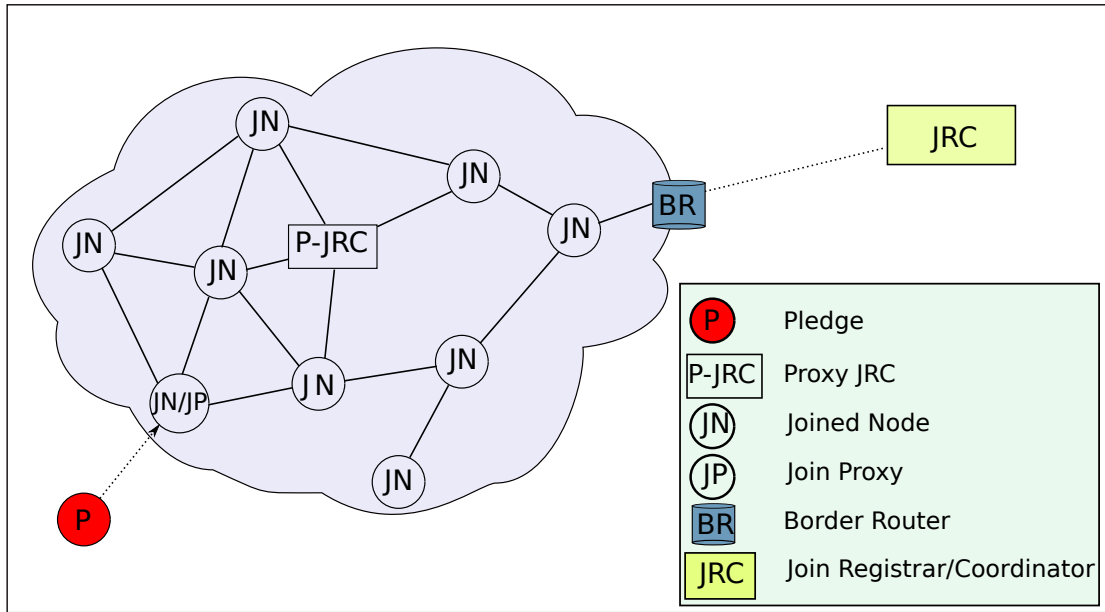
1. Pledge, JP tarafından gönderilen EB mesajını alır ve bu mesaj sayesinde ağı senkronize olur. EB mesajı, JP'nin bağlantı katmanı adresini ve bazı durumlarda yerel IPv6 adresini içerir. Bu süreç, pledge'in ağı keşfetmesine ve uygun bir JP bulmasına olanak tanır. EB mesajları, TSCH zamanlamasını ve ağın genel yapılandırma bilgilerini içerir. Pledge, birden fazla EB mesajı alabilir ve en uygun olanı seçer.
2. Düğüm, kendi bağlantı yerel IPv6 adresini yapılandırır ve bu adresi JP'ye reklam eder. Bu adım, pledge'in JP ile güvenli bir şekilde iletişim kurabilmesi için gereklidir. Komşu keşfi süreci, IEEE 802.15.4 standardına göre gerçekleştirilir ve adreslerin benzersizliğini sağlar. Pledge ve JP, link-lokal adresler üzerinden iletişim kurar ve bu adresler, katılım süreci boyunca kullanılır.
3. Pledge, CoJP katılım talebi mesajını JP'ye gönderir ve bu mesaj JP tarafından JRC'ye iletilir. CoJP, güvenli bir kanal üzerinden haberleşmeyi sağlar ve mesajların bütünlüğünü, gizliliğini ve tekrarlanabilirliğini garanti eder. Katılım talebi, pledge'in ağda hangi rolü oynamak istediğini ve hangi ağı katılmak istediğini belirtir. JRC, bu talebi değerlendirir ve onaylarsa, gerekli ağ parametrelerini içeren bir yanıt mesajı gönderir. Bu yanıt mesajı, bağlantı katmanı anahtarlarını, kısa adresi ve diğer ağ ayarlarını içerir.

Bu aşamaların tümü boyunca, K1 ve K2 anahtarları önemli bir güvenlik rolü üstlenir. K1 anahtarı, CoJP katılım sürecinde ilk olarak EB mesajlarının ve katılım taleplerinin güvenliğini sağlamak amacıyla kullanılır. K1, TSCH katmanındaki veri çerçevelerini şifreleyerek iletilen mesajların gizliliğini ve bütünlüğünü korur. Bu sayede, Pledge ve JP arasında güvenli bir iletişim kanalı oluşturulur ve ağdaki hassas bilgilere yetkisiz erişim önlenmiş olur. CoJP,

OSCORE ile güvenli hale getirilmiş bir kanal üzerinden iletişim kurar. OSCORE, uçtan uca gizlilik, veri bütünlüğü, tekrar koruması ve taleplerle yanıtların güvenli bir şekilde bağlanmasını sağlar. Pledge ve JRC, önceden paylaşılmış bir anahtar (Pre Shared Key - PSK) kullanarak güvenli bir kanal oluşturur. Bu kanal, katılım sürecindeki tüm mesajların güvenli bir şekilde iletilmesini sağlar.

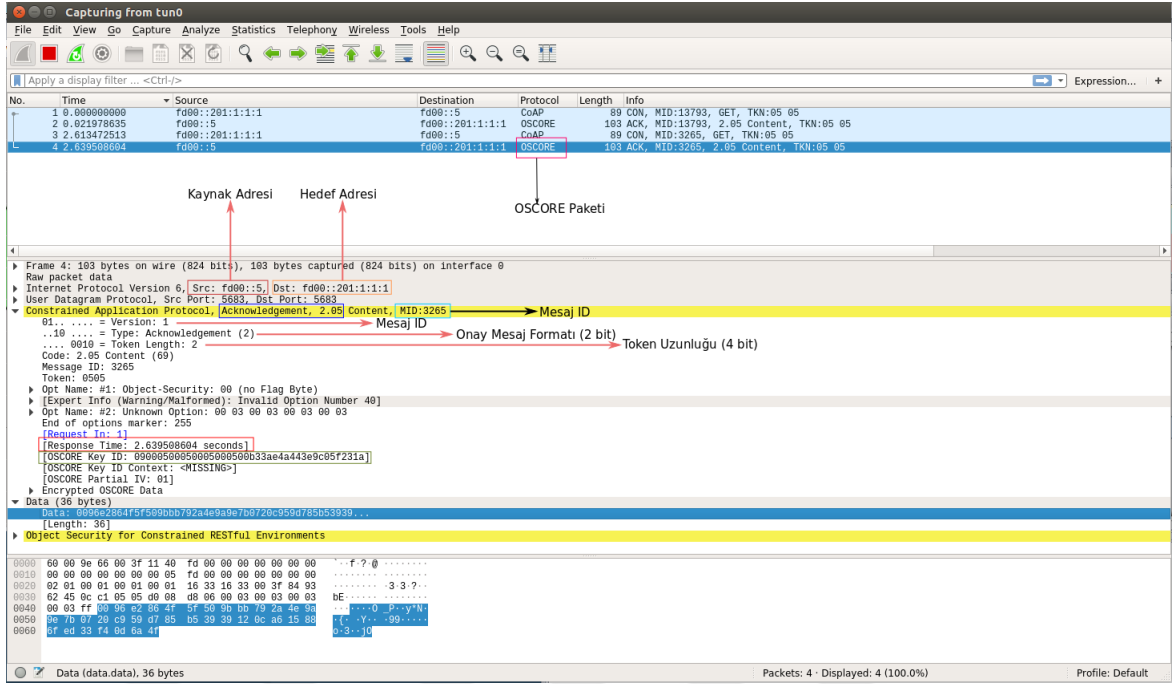
CoJP sürecinde K1, IEEE 802.15.4 standardına uygun olarak MAC katmanındaki mesajları şifreler ve bu sayede veri ve MAC komut çerçevelerinin güvenliğini sağlar. K2 anahtarı ise, CoJP'nin daha ileri aşamalarında kontrol mesajları gibi ağ yönetimi için kritik olan trafiği şifreleyerek katmanlar arası güvenliği destekler. K1 anahtarı Pledge'in ağa güvenli şekilde katılımını sağlarken, K2 anahtarı, ağda devam eden yönetim trafiğinin güvenliğini sağlamaya odaklanır. Bu iki anahtar, ağın güvenliğini ve katılım sürecinin etkinliğini birlikte destekler.

CoJP'nin güvenlik çerçevesine ek olarak, daha fazla güvenlik sağlamak için ek mekanizmalar da eklenebilir. Örneğin, ağ katmanı koruması, kimlik doğrulama yöntemleri ve saldırı tespit sistemleri, CoJP sürecinin güvenliğini artırmak için kullanılabilir. Bu ek güvenlik önlemleri, DoS saldırılarına karşı koruma sağlar ve katılım sürecinin güvenliğini artırır. CoJP protokolü, OSCORE ile güvenli hale getirilmiş bir kanal üzerinden iletişim kurar ve mesajların bütünlüğünü, gizliliğini ve tekrarlanabilirliğini sağlar. Bu süreç, 6TiSCH ağının güvenli ve verimli bir şekilde çalışmasını sağlar.



Şekil 52. Proxy-JRC tabanlı kimlik doğrulama altyapısı

Şekil 53, Cooja simülatörü tarafından gönderilen ağa katılım (join request) OSCORE paketinin Wireshark üzerinde nasıl parse edildiğini göstermektedir. Görüntüde, OSCORE



Şekil 53. Wireshark üzerinde ağa katılım paketlerinin incelenmesi

protokolü kullanarak gönderilen bir CoAP paketinin ayrıntıları incelenmiştir. Paketin kaynağı ve hedef adresleri belirgin bir şekilde gösterilmiştir; kaynak adresi "fd00::5" ve hedef adresi "fd00::201:1:1:1". Protokol olarak OSCORE kullanıldığı ve paketin uzunluğu 103 bayt olduğu belirtilmiştir. Bu paketin bir ACK mesajı olduğu, CoAP versiyonu ve mesajın içerik kodu ile birlikte gösterilmektedir. Mesaj ID'si "3265" olarak belirlenmiş ve onay mesaj formatı ile token uzunluğu gibi ayrıntılar vurgulanmıştır.

Paketin içeriğinde, OSCORE protokolüne özgü güvenlik bilgileri ve şifrelenmiş veri bulunmaktadır. OSCORE anahtarının (Key ID) ve bağlamının (Context) ayrıntıları ile birlikte OSCORE veri doğrulama etiketi gösterilmektedir. Bu bilgiler, mesajın güvenliğini sağlamak ve iletim sırasında bütünlüğünü korumak amacıyla kullanılır. Alt kısımda, şifrelenmiş OSCORE verisi ve bu verinin uzunluğu belirtilmiştir. Bu, güvenli iletişimin sağlanması için gerekli olan tüm güvenlik parametrelerinin doğru bir şekilde iletiliğini ve işlendiğini göstermektedir. Wireshark arayüzü, bu tür detaylı analizler yaparak ağ güvenliği ve protokol doğrulaması için güçlü bir araç olarak kullanılmaktadır. Bu simülasyon, ağa katılım sürecinin güvenliğini test etmek ve OSCORE protokolünün uygulanabilirliğini doğrulamak amacıyla gerçekleştirilmiştir.

Yukarıda açıklanan merkezi kimlik doğrulama süreci, 6TiSCH ağlarındaki güvenlik gereksinimlerini karşılamakla birlikte, özellikle büyük ölçekli ve dinamik ağ yapılarında bazı sınırlamalara sahip olabilir. Bu sınırlamalar, tek bir merkezi doğrulama otoritesine bağımlı olmanın doğurduğu güvenlik riskleri ve ölçeklenebilirlik sorunları ile ilgilidir. Merkezi

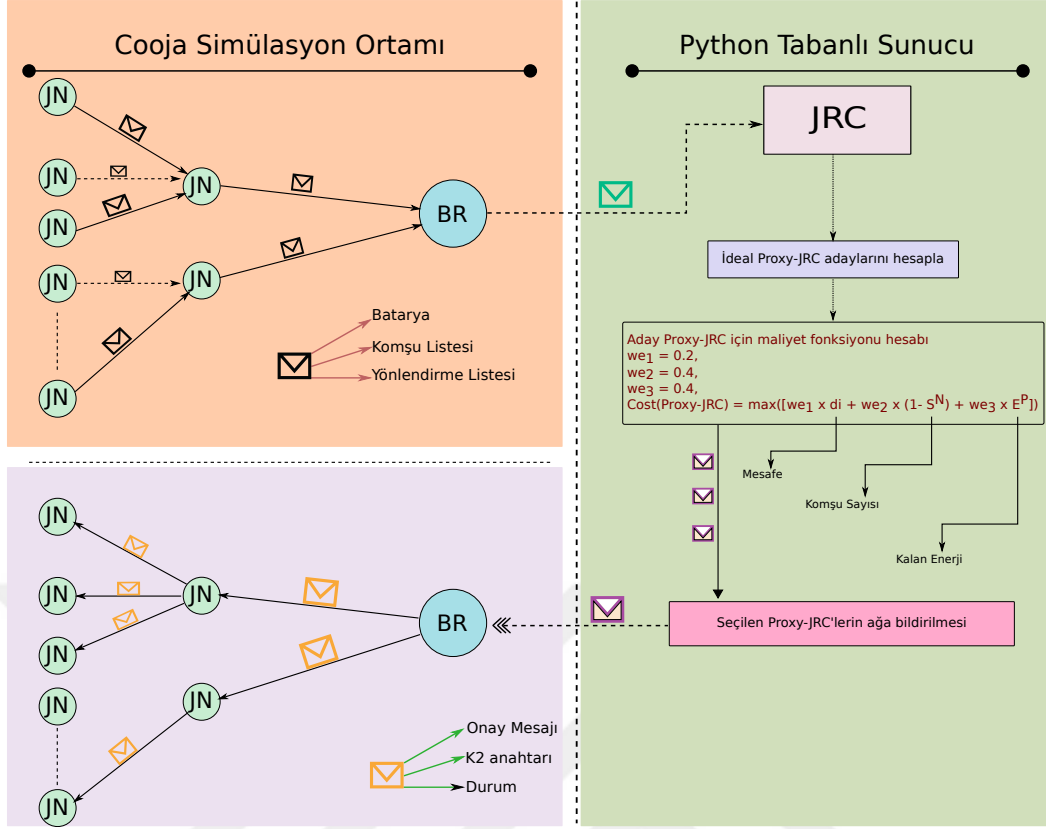
kimlik doğrulama süreçleri, yüksek yük ve gecikme sürelerine yol açabilir, bu da ağın genel performansını ve güvenliğini olumsuz etkileyebilir. Bu nedenle, merkezi olmayan kimlik doğrulama çözümleri, daha esnek, güvenli ve ölçeklenebilir bir yapıya olanak tanımaktadır. Aşağıda, dağıtık Proxy-JRC kimlik doğrulama noktalarının belirlenmesi ve uygulanması süreci ayrıntılı bir şekilde ele alınacaktır. Bu süreç, ağ güvenliğini ve performansını artırmayı hedefleyen önemli bir adım olarak öne çıkmaktadır.

5.2. Dağıtık Proxy-JRC Kimlik Doğrulama Noktalarının Belirlenmesi

Şu ana kadar açıklanan yöntemler, merkezi bir kimlik doğrulama altyapısına dayanmaktadır. Merkezi kimlik doğrulama adımları, JRC'nin tüm ağdaki yeni düğümleri doğrulaması üzerine kuruludur. Bu yöntem, belirli durumlarda ölçeklenebilirlik ve güvenlik açısından sınırlamalara sahip olabilir. Çalışmada, dağıtık Proxy-JRC noktalarının belirlenmesi ve uygulanması üzerine odaklanılmıştır. Dağıtık Proxy-JRC kimlik doğrulama altyapısı, merkezi doğrulama yükünü azaltarak, ağın güvenlik ve performansını artırmayı hedefler. Her bir Proxy-JRC, kendi bölgesindeki düğümleri doğrular ve bu şekilde ağın tamamında daha etkin ve güvenli bir kimlik doğrulama sağlanmış olur.

Proxy-JRC'ler, kimlik doğrulama ve anahtar güncellemeleri gibi işlemleri merkezi JRC'ye ihtiyaç duymadan gerçekleştirebilen dağıtık kimlik doğrulama birimleridir. Bu doğrulama mekanizması, RPL topolojisinde en iyi proxy kimlik doğrulama cihazı konumlarını belirlemek için gereklidir. Böylece, sensör doğrulama ve anahtar güncellemelerinin ağ içinde gerçekleştirilmesine olanak tanınır. Şekil 54'te, JRC'nin gelen sensör verilerine dayalı olarak dağıtık düğümleri seçme sürecini gösterilmektedir. Veriler, düğümün komşu sayısı, kalan enerjisi, yönlendirme tablosu ve düğümün rankı gibi bilgilerden oluşur. Seçilen bu düğümler Proxy-JRC olarak adlandırılır ve kimlik doğrulama bilgileri JRC'den Proxy-JRC'lere gönderilir. Algoritma 1 ve Algoritma 2, Proxy-JRC seçim mekanizmasını detaylı olarak açıklamaktadır.

Proxy-JRC seçim sürecini değerlendirmek ve doğrulamak amacıyla çeşitli simülasyon senaryoları gerçekleştirilmiştir. Bu simülasyonlar için kullanılan parametreler, Tablo 9'da belirtilmiştir. Simülasyonlar, Contiki Cooja simülatörü kullanılarak gerçekleştirilmiş ve IEEE 802.15.4e TSCH MAC modeli, IPv6/RPL ağ katmanı ve ızgara topolojisi gibi standart ayarlar kullanılmıştır. Düğümlerin düzenli bir şekilde yerleştirildiği ve birbirleriyle belirli bir düzen içinde iletişim kurduğu ızgara (grid) topolojisi, simülasyonların temelini oluşturmuştur. İletim



Şekil 54. Aday Proxy-JRC'lerin belirlenmesi

ve alım menzili, her düğümün diğer düğümlerle iletişim kurabileceği mesafe olarak $50 \text{ m} \times 100 \text{ m}$ olarak belirlenmiştir.

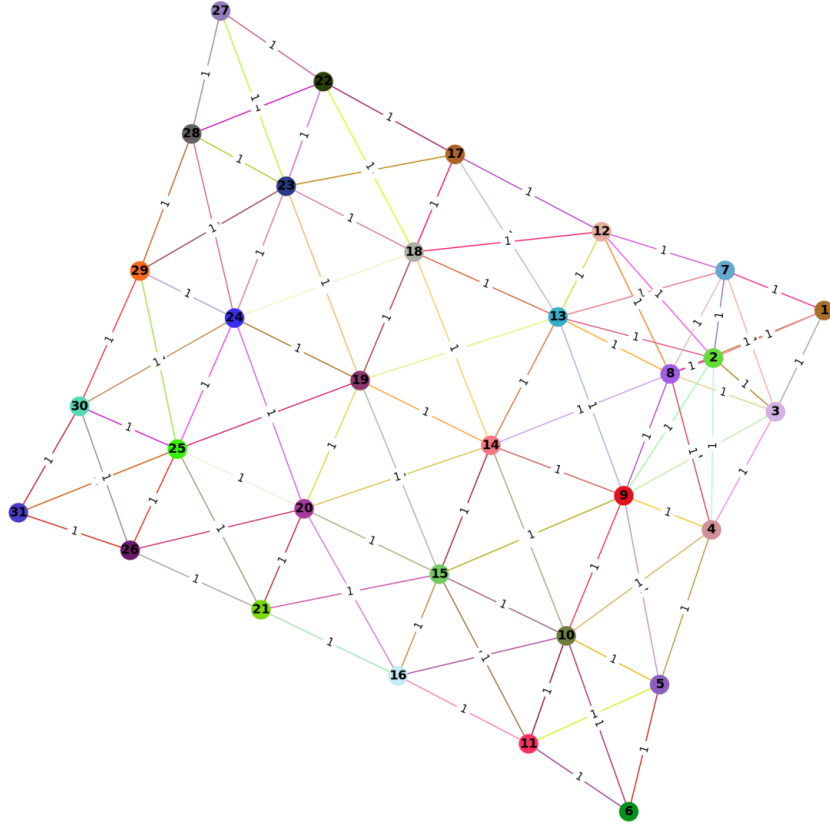
Simülasyon alanı $200 \text{ m} \times 200 \text{ m}$ olup, her bir senaryonun çalıştırıldığı süre ağ performansını ve enerji tüketimini değerlendirmek için yeterli bir süre olarak 1 saat belirlenmiştir. Her bir simülasyon, MRHOF (ata düğüm seçme eşik değeri $6 \cdot ETX_D$) amaç fonksiyonu kullanılarak gerçekleştirilmiştir. Simülasyon süresince, farklı düğümler arasındaki mesafe, komşu sayısı, yönlendirme tablosu ve düğümlerin enerji seviyeleri gibi parametreler dikkate alınarak maliyet fonksiyonları hesaplanmıştır. Komşuluk tablosu ve yönlendirme tablosu her 60 saniyede bir güncellenmiştir. Düğümlerin pil seviyeleri her 90 saniyede bir güncellenerek enerji tüketimi izlenmiştir. Rank bilgisi ise her 120 saniyede bir yenilenmiştir.

Şekil 55, 32 düğümlü bir ağda düğümlerin komşuluk ilişkilerini göstermektedir. Düğümlerin düzenli yerleşimi ve komşu sayıları, ağın genel iletişim yapısını değerlendirmek açısından kritik öneme sahiptir. Her düğümün komşu düğümleri ile olan ilişkisi, ağın yoğunluğu, kapsama alanı ve veri iletim yollarının oluşturulmasında belirleyici bir faktördür. Örneğin, bir düğümün çok sayıda komşuya sahip olması, düğümün iletişim yollarında daha fazla tercih edilmesine ve bu düğümün üzerindeki trafik yükünün artmasına neden olabilir. Bu durum, enerji tüketimini ve yönlendirme performansını doğrudan etkileyebilir. Düşük sayıda

Tablo 9. Simülasyon Parametreleri

Parametre	Değer
Simülatör	Contiki Cooja
Yayılım Modeli	Unit Disk Graph Medium
MAC Modeli	IEEE 802.15.4e TSCH
Ağ Katmanı	IPv6 / RPL
Adaptasyon	6LoWPAN
Fiziksel Katman	IEEE 802.15.4
Topoloji	Izgara (Grid)
Tx/Rx Menzili	50 m X 100 m
Simülasyon Alanı	200 m X 200 m
Simülasyon Süresi	1 saat
Hedef Fonksiyonu	MRHOF (OF-6ETX iyileştirilmiş)
Keep-alive süresi	30 saniye
Çizelgeleme Fonksiyonu	Dağıtık Çizelgeleme
TSCH Slot Süresi	15 ms
Radyo Kanal Sayısı	16
Timekeeping Slot Sayısı	1
Shared Slot Sayısı	2
Kuyruk (Queue) Boyutu	4
Kuyruk Modeli	FIFO
Maliyet Fonksiyonu 1	Mesafe, Komşu, Enerji
Maliyet Fonksiyonu 2	Komşu, Enerji
Komşuluk Tablosu	1 Paket / 60 saniye
Rank Bilgisi	1 Paket / 120 saniye
Yönlendirme Tablosu	1 Paket / 60 saniye
Pil Seviyesi Güncelleme Aralığı	1 Paket / 90 saniye
Enerji Modeli	Contiki Powertrace

komşuya sahip düğümler ise yönlendirme ağının kenarlarında kalabilir ve bu düğümler için daha düşük enerji tüketimi gözlemlenebilir. Ayrıca, komşuluk yapısı, düğümler arası doğrudan iletişimin mümkün olup olmadığını ve hangi düğümlerin yönlendirme yollarında kilit rol oynadığını belirler. Şekil 56 ise 32 düğümlü bir ağda yönlendirme ilişkilerini göstermektedir. Yönlendirme ilişkileri, düğümlerin veri paketlerini nasıl ilettiğini ve ağın genel performansını nasıl etkilediğini anlamak açısından büyük önem taşır. Yönlendirme yollarının uzunluğu, düğümler arasındaki mesafe, düğüm başına düşen yönlendirme yükü gibi faktörler, ağın genel verimliliğini belirler. Örneğin, bazı düğümler, diğer düğümlerin veri paketlerini sürekli ileten ana yönlendirme düğümleri (relay nodes) haline gelebilir. Bu tür düğümler, yoğun bir şekilde veri iletimi yaparken diğer düğümlere kıyasla daha fazla enerji tüketir ve ağdaki kaynakların dengesiz kullanımına yol açabilir. Ayrıca, yönlendirme yolundaki düğümlerin fazla olması, veri iletim gecikmelerine ve potansiyel tıkanıklıklara neden olabilir. Bu nedenle, yönlendirme

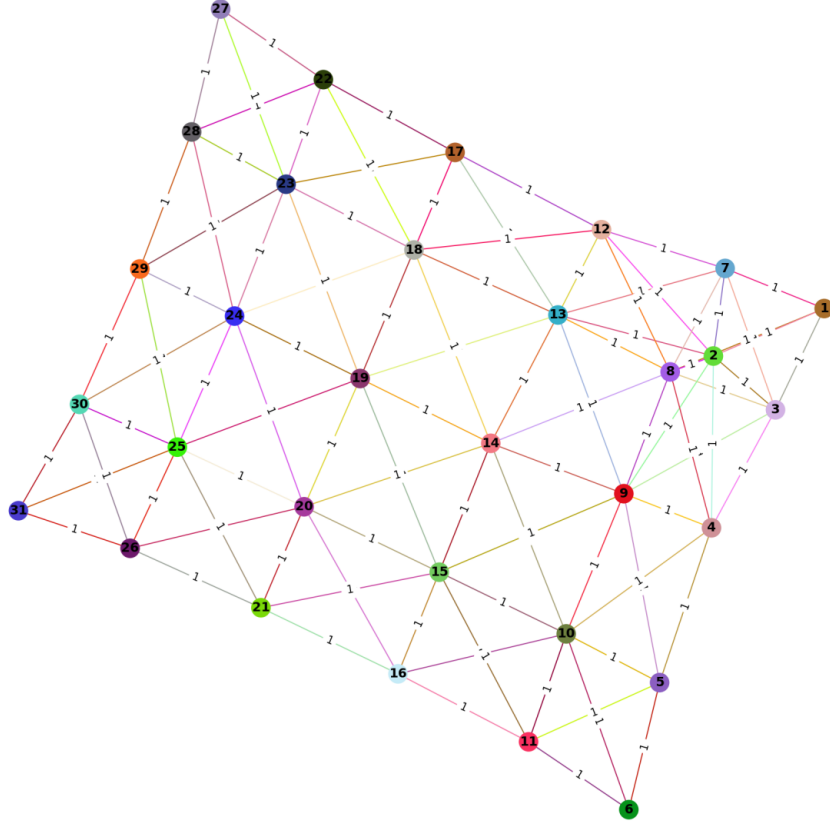


Şekil 55. 32 düğümlü bir ağda komşuluk ilişkisi

ilişkileri üzerinde yapılan analizler, veri trafiğinin dengeli bir şekilde dağılıp dağılmadığını, düğümler arası iletişimin verimliliğini ve ağın genel performansını artırmak için yapılması gereken iyileştirmeleri ortaya koyar. Yönlendirme ilişkileri ayrıca ağın ölçeklenebilirliği ve farklı düğüm yoğunluklarına karşı dayanıklılığı açısından da önemli bilgiler sağlar.

Bu parametreler doğrultusunda gerçekleştirilen simülasyonlar, ağ performansını ve enerji verimliliğini değerlendirmek için detaylı analizler sunmuştur. Elde edilen sonuçlar, Proxy-JRC seçim mekanizmasının etkinliğini ve ağın genel performansını değerlendirmeye yardımcı olmuştur.

Sınır yönlendirici, ağ kurulum paketlerini (EB'ler, DIO'lar) yayınlarak ağ oluşumunu başlatır. Ancak, sınır yönlendirici dışındaki düğümlerin ağa katılabilmesi için JRC tarafından onaylanması gerekmektedir. Tüm düğümler güvenli bir şekilde ağa dahil olduktan sonra, komşuluk tablosunu, yönlendirme tablosunu ve kalan enerji değerlerini düzenli aralıklarla JRC'ye iletir. JRC, iletilen bilgileri Algoritma 1 yardımıyla işler ve ağdaki Proxy-JRC düğümlerini seçer. Algoritma 2, seçilen Proxy-JRC düğümlerinin aynı yönlendirme yolunda olup olmadığını kontrol eder. İki düğüm aynı yönlendirme yolunda ise, algoritma alternatif bir Proxy-JRC bulmaya çalışır. Ağ kurulumu tamamlandığında, mevcut cihazların anahtar güncellemeleri RPL topolojisinde bulunan Proxy-JRC düğümleri tarafından gerçekleştirilir.



Şekil 56. 32 düğümlü bir ağda yönlendirme ilişkisi

Cooja düğümleri için batarya modülü tasarlanarak düğümlerin enerji kullanımı değerlendirilmiştir. Burada, önerilen mekanizmaları değerlendirmek için Exp5438 cihazının enerji tüketim modeli kullanılmıştır (Instruments, 2024). Her bir düğümün enerji tüketimi, düğümün her alt sisteminin aktif olduğu süreler üzerinden ölçülmüştür. CPU, LPM, Tx ve Rx, her düğümün çalışma modlarını temsil eder ve enerji tüketimini değerlendirmek için kullanılır. CPU, düğüm CPU'sunun aktif olduğu süreleri temsil eder. LPM, düğümün düşük güç modunda olduğu süreleri ifade eder. Radyo dinleme (Rx) ve iletme (Tx) parametreleri ise en fazla enerji tüketen radyo etkinliğini ölçmek için kullanılır.

Önerilen Proxy-JRC seçim mekanizması için şu kriterler göz önünde bulundurulmuştur:

1. Önerilen yöntem, RPL protokolünü yeni bir yönlendirme mekanizması ile değiştirmez. Bunun yerine, yönlendirme için RPL'yi kullanır ve merkezi olmayan alt ağları seçer.
2. Ağ derinliği arttıkça, ideal merkezi olmayan varlığı seçmek zorlaşır. İdeal Proxy-JRC seçimi kombinatoryal bir problemdir ve NP-zordur (Zhu vd., 2012).
3. Çözümde; rank, komşuluk tablosu, yönlendirme tablosu ve kalan enerji Proxy-JRC seçim süreci için dikkate alınmıştır.

Algorithm 1 Proxy-JRC seçiminde kullanılan maliyet analizi

Input: we_1, we_2, we_3, NOS (*number of subnet*)

Definition: $we_1 = 0.2, we_2 = 0.4, we_3 = 0.4$ ▷ Coefficients

Input: $di_1, di_2 \dots di_N$ ▷ Distance to the border router

Input: $S_1^N, S_2^N \dots S_N^N$ ▷ Number of nodes at a hop distance

Input: $(1 - E_1^P), (1 - E_2^P) \dots (1 - E_N^P)$ ▷ Energy values of the nodes

Input: $algorithm_list\{alg_1, alg_2\}$

Output: $output\{Proxy_JRC\}$

```

1: procedure PROXY-JRC_SELECTION
2:    $selected\_algorithm \leftarrow \{algorithm\_list\}$ 
3:   for all  $i \in network[Node_{1..N}]$  do
4:     if  $selected\_algorithm \leftarrow alg\_1$  then
5:        $cost(i) = we_1 \times di\_JRC_{N_i} + we_2 \times S^{N_i} + we_3 \times (1 - E_{N_i}^P)$ 
6:     else if  $selected\_algorithm \leftarrow alg\_2$  then
7:        $cost(i) = we_1 \times (1 - E_{N_i}^P) + we_2 \times Neighbor_i$ 
8:     else
9:        $cost(i) = we_1 \times Neighbor_i$ 
10:    sort $\{cost[Node_{1..N}]\}$ 
11:    for  $s \in (rpl\_subnet[N], NOS)$  do
12:      if  $size[s] \geq 2$  then
13:         $new\_list\{\} = combination(s, 2)$ 
14:      else
15:         $new\_list\{\} = combination(2, 1)$ 
16:      if  $size[s] \geq 2$  then
17:        for  $sub\_list \in (new\_list\{\})$  do
18:          if  $!best\_proxy\_check(sub\_list)$  then
19:             $output.add(sub\_list\_node)$ 
20:      else
21:         $output.add(s)$ 

```

4. Öneride, Proxy-JRC'lerin sayısının optimizasyonu ve belirli parametre atamalarının ağ yükü analiz edilmemiştir.
5. Önerilen yöntem, kümeleme yöntemine benzerlik göstermektedir ancak, küme seçiminde yönlendirme adımı da değişirken, bu yöntem yönlendirme protokolünü değiştirmemektedir. Küme başı seçiminde, yönlendirme algoritması genellikle küme başına göre optimize edilirken, bu yöntemde RPL protokolü sabit kalır ve yalnızca Proxy-JRC seçimi gerçekleştirilir. Bu farklılık, önerilen yöntemin kümeleme algoritmalarından ayrılmasını sağlar ve mevcut yönlendirme protokolünün kullanılmasına olanak tanır.

Algorithm 2 Seçilen Proxy-JRC’lerin sınır yönlendirici ve yönlendirme tablosuna göre kontrolü

Input: routing_table[], neighbor_table[]

Input: from_node, to_node

Input: penalty_node = Border Router ▷ Border router can never select Proxy-JRC

Output: state

1: **procedure** BEST_PROXY_CHECK

2: state $\leftarrow$ *True*

3: **for** $i = 0$ to length(routing_table) **do**

4: **if** routing_table[from_node][i] = 1 **then**

5: state $\leftarrow$ *True*

 state $\leftarrow$ (neighbor_table[to_node][from_node] = 1)

and(neighbor_table[from_node][to_node] = 1)

and(neighbor_table[penalty_node][from_node] = 1)

and(neighbor_table[from_node][penalty_node] = 1)

and(neighbor_table[penalty_node][to_node] = 1)

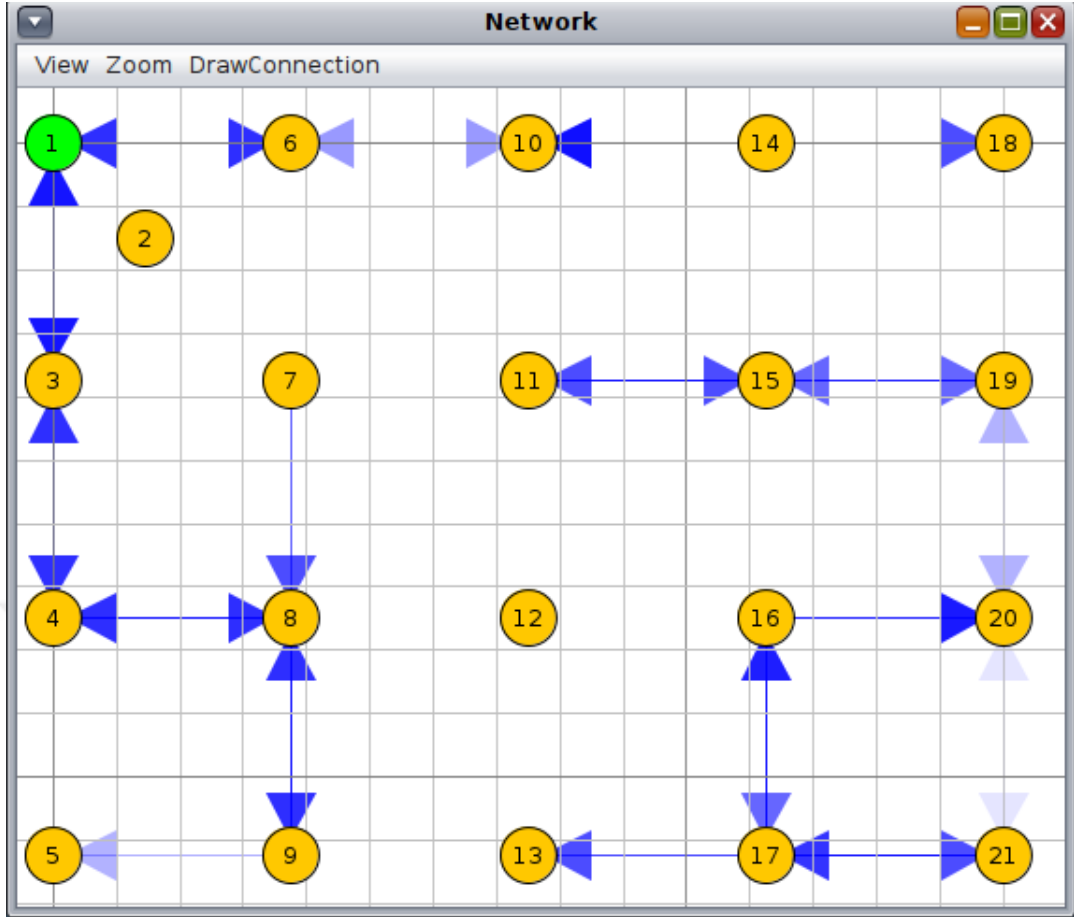
and(neighbor_table[to_node][penalty_node] = 1)

6: **return** state

6. Kümeleme yaklaşımı, sürekli olarak küme başlarının değişmesini gerektirir. Ancak, gerçek dünyadaki ağlarda bu tür sürekli değişimler pratik değildir. Küme başlarının sıkça değiştirilmesi, gerçek ağlarda ek yük ve gecikmelere neden olur ve bu durum, ağ performansını olumsuz etkileyebilir.

Şekil 57, Cooja’da simüle edilen 20 düğümlü örnek bir ağı göstermektedir. Bu ağda, doğru konumlandırılmış 2 Proxy-JRC’nin seçimi, toplamda 190 farklı alt kümenin taranmasını gerektirir. Bu alt kümeler, kimlik doğrulama için geçen ortalama süre ve enerji tüketimi gibi parametrelere dayalı olarak değerlendirilir. Doğrusal arama ile bu 190 farklı çözüm alanından elde edilen ağın topolojisindeki optimal noktalar, Oracle Proxy-JRC olarak adlandırılır.

Oracle Proxy-JRC, tüm çözüm uzayının taranmasını ifade eden bir yaklaşımdır. Bu yöntem, ağdaki her bir düğümün tek tek test edilmesi ve maliyet analizi kriterlerine göre en uygun Proxy-JRC düğümlerinin belirlenmesi sürecini kapsar. Oracle’nin temel amacı, düğümlerin kimlik doğrulama sürelerini ve enerji tüketimini minimize etmektir. Oracle Proxy-JRC, tüm olasılıkları değerlendirerek, en iyi performans (düşük enerji tüketimi, düşük uçtan uca gecikme, kısa sürede düğüm kimliklerinin doğrulanması) gösteren Proxy-JRC düğümlerini belirler. Özellikle, Proxy-JRC’lerin doğru konumlandırılması, düğümler arasında güvenli iletişimin hızla sağlanması, ağdaki uçtan uca gecikmenin azaltılması ve enerji



Şekil 57. 21 düğümlü RPL tabanlı Cooja simülasyonu

verimliliğinin artırılması açısından büyük önem taşır. Ancak, bu yöntemin en büyük zorluğu, ağ büyüklüğünün artmasıyla birlikte çözüm uzayının üstel olarak genişlemesidir.

Az sayıda düğüm içeren ağlarda Oracle Proxy-JRC düğümleri kısa süre içinde belirlenebilir. Fakat düğüm sayısı arttıkça arama alanı üstel olarak genişler ve bu durum, çözüm bulmayı zorlaştırır. Örneğin, 20 düğümlü bir ağ senaryosunda, sınır geçidi (düğüm id'si 1) çözüm alanından çıkarıldığında 171 farklı çözüm elde edilebilir. Ayrıca, sınır yönlendiriciden bir atlama mesafesindeki düğümlerin ve ağın uç noktalarındaki düğümlerin çıkarılmasıyla arama alanı daha da daraltılabilir.

Doğrusal arama yaklaşımının hesaplama karmaşıklığının üstesinden gelmek için yeni bir sezgisel algoritma olan Algoritma 1 kullanılmıştır. Bu durumda, Proxy-JRC'ler önerilen maliyet fonksiyonlarına (Maliyet fonksiyonu 5.1, Maliyet fonksiyonu 5.2) dayalı olarak seçilir. Denklem 5.1, sınır yönlendiriciye olan mesafeye ($di_JRC_{N_i}$), kalan enerjiye ($1 - E_{N_i}^P$) ve komşu sayısına (S^{N_i}) farklı ağırlıklar verir. Örnek senaryo için, doğrusal arama yöntemi ile belirlenen Oracle Proxy-JRC'ler Şekil 58'da gösterilmiştir. Önerilen yaklaşımlar, bu ağ için farklı maliyet fonksiyonları kullanarak Oracle Proxy-JRC algoritmasına karşı değerlendirilmiştir.

$$cost(N_i) = [we_1 * di_JRC_{N_i} + we_2 * S^{N_i} + we_3 * (1 - E_{N_i}^P)] \quad (5.1)$$

Denklem 5.2, düğümlerin kalan enerjisini $(1 - E_{N_i}^P)$ ve bir sekme uzaklıktaki komşularının sayısını ($Neighbor_i$) dikkate alan bir maliyet fonksiyonunu belirtmektedir. Bu maliyet fonksiyonu için bulunan optimum ağırlık katsayıları (we_1, we_2) sırasıyla 0.6 ve 0.4 olarak bulunmuştur. Bu katsayılar, kalan enerjinin maliyet hesaplamasında daha büyük bir etkiye sahip olduğunu göstermektedir. Bu durum Şekil 60'de detaylı olarak gösterilmiştir.

$$cost(N_i) = [we_1 * (1 - E_{N_i}^P) + we_2 * Neighbor_i] \quad (5.2)$$

Önerilen mekanizma, enerji tüketimi, atlama mesafesi, rank, yönlendirme ve komşuluk tabloları gibi parametreleri kullanarak Proxy-JRC seçimi yapar. Ağdan toplanan veri seti, Proxy-JRC seçim maliyet hesaplamalarını gerçekleştirmek için kullanılır. Proxy-JRC seçim süreci tamamlandığında, seçilen Proxy-JRC'ler hakkında bilgi içeren bir mesaj ağdaki ilgili düğümlere gönderilir. Bu iletişim adımı, ağdaki kimlik doğrulanmış düğümlerin seçilen Proxy-JRC'leri kullanarak kimlik doğrulama bilgilerini güncellemelerini sağlar.

Tablo 13, 35 ve 50 düğümlü ağlar için farklı ağırlık katsayıları (we_1, we_2) kullanılarak Maliyet Fonksiyonu 2'nin (Denklem 5.2) sonuçlarını göstermektedir. Her bir katsayı kombinasyonu için enerji tüketimi (mW) ve kimlik doğrulama süreleri analiz edilmiştir. Tablodaki veriler, farklı katsayı çiftlerinin (örneğin, $we_1 = 0.32$ ve $we_2 = 0.68$) ağ performansını nasıl etkilediğini ortaya koymaktadır. En düşük enerji tüketimi ve kimlik doğrulama sürelerine ulaşan ideal katsayılar, 35 düğümlü ağlar için $we_1 = 0.60$ ve $we_2 = 0.40$ olarak tespit edilmiştir. Bu katsayı kombinasyonu, ağın enerji verimliliğini optimize ederken kimlik doğrulama süresini de minimum seviyede tutmaktadır. Aynı şekilde, 50 düğümlü ağlar için de bu katsayıların ($we_1 = 0.60$ ve $we_2 = 0.40$) enerji tüketimi ve kimlik doğrulama süreleri açısından en ideal sonuçları verdiği gözlemlenmiştir. Bu nedenle, ağırlık katsayılarının bu şekilde seçilmesi, ağın performansını optimize etmek için önemli bir kriterdir. Tabloda yer alan diğer katsayı kombinasyonları ise referans olarak kullanılabilir ve farklı performans ihtiyaçlarına göre değerlendirilebilir.

Tablo 10. 20 Dügümlü bir ağ için ağırlık katsayılarının seçimi

D.S	we_1	we_2	we_3	Maliyet Fonksiyonu 1 ▷ Eq. 5.1		Maliyet Fonksiyonu 2 ▷ Eq. 5.2	
				mW	Kimlik Doğrulama(s)	mW	Kimlik Doğrulama(s)
20	0.18	0.25	0.57	12.7	180	15.2	240
	0.30	0.15	0.55	12.9	200	13.8	210
	0.12	0.40	0.48	14.9	220	15.1	230
	0.25	0.35	0.40	15.0	229	16.9	350
	0.22	0.28	0.50	13.8	210	13.9	205
	0.37	0.18	0.45	13.9	222	13.2	213
	0.10	0.50	0.40	13.3	199	15.6	298
	0.28	0.22	0.50	14.2	320	14.5	247
	0.33	0.27	0.40	13.1	223	15.7	287
	0.20	0.30	0.50	13.2	205	17.3	380
	0.15	0.30	0.55	14.9	210	13.8	230
	0.24	0.26	0.50	13.1	278	17.5	430
	0.11	0.47	0.42	12.9	210	15.1	220
	0.35	0.18	0.47	12.0	200	15.5	250
	0.28	0.25	0.47	12.5	223	15.6	261
	0.20	0.40	0.40	11.3	160	10.9	217
	0.24	0.33	0.43	14.5	227	14.5	290
	0.32	0.28	0.40	15.1	251	15.3	284
	0.15	0.50	0.35	13.6	215	17.7	490
	0.40	0.20	0.40	13.2	270	13.8	180
	0.25	0.25	0.50	12.9	220	14.2	200
	0.35	0.35	0.30	13.5	240	13.5	210
	0.28	0.32	0.40	14.1	210	15.1	225
	0.22	0.38	0.40	13.8	260	14.3	250
	0.15	0.55	0.30	14.5	250	16.2	270
	0.30	0.25	0.45	12.7	280	14.8	260
	0.40	0.15	0.45	13.2	200	15.5	230
	0.18	0.42	0.40	13.1	210	14.4	220
	0.25	0.35	0.40	12.9	290	15.2	280
	0.28	0.30	0.42	14.0	220	14.9	300
	0.32	0.28	0.40	13.4	230	15.3	240
	0.24	0.40	0.36	13.7	240	14.7	250
	0.20	0.45	0.35	13.6	250	15.1	270
	0.35	0.25	0.40	14.1	260	14.5	280
	0.33	0.30	0.37	14.0	270	14.3	290
	0.18	0.30	0.52	13.2	220	15.4	230
	0.22	0.32	0.46	13.4	210	14.9	260
	0.38	0.22	0.40	13.7	250	15.0	270
	0.28	0.35	0.37	13.8	240	14.2	280
	0.30	0.32	0.38	14.1	250	14.5	300
	0.25	0.40	0.35	14.3	220	14.8	240
	0.27	0.33	0.40	13.5	230	15.1	280
	0.30	0.35	0.35	13.7	240	14.6	260

Tablo 11. 35 Dügümlü bir ağ için ağırlık katsayılarının seçimi

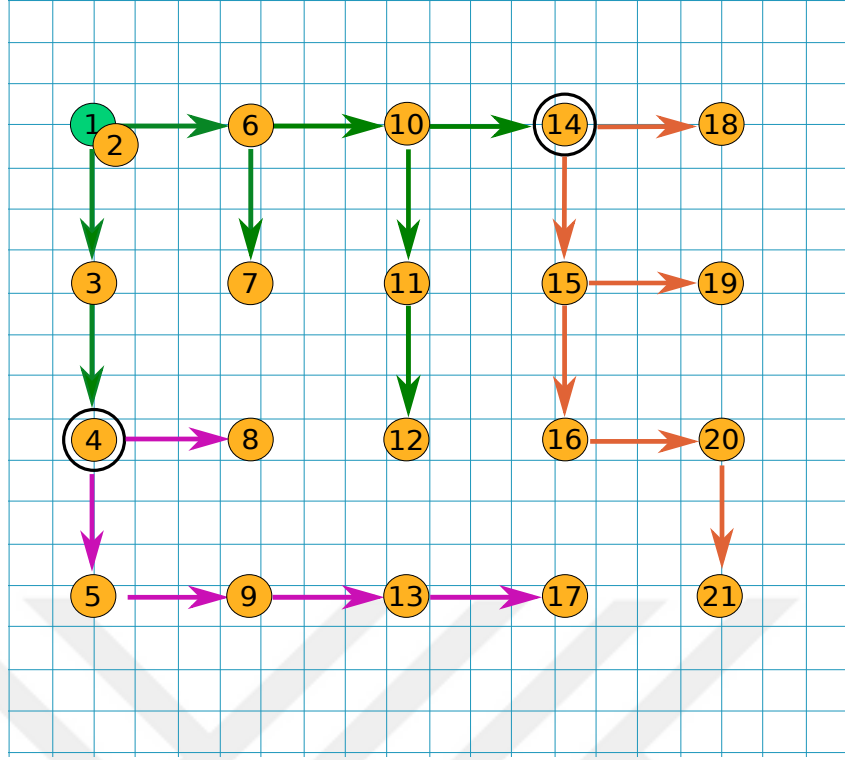
D.S	we_1	we_2	we_3	Maliyet Fonksiyonu 1 ▷ Eq. 5.1		Maliyet Fonksiyonu 2 ▷ Eq. 5.2	
				mW	Kimlik Doğrulama(s)	mW	Kimlik Doğrulama(s)
35	0.18	0.25	0.57	15.7	230	17.5	320
	0.30	0.15	0.55	16.1	240	16.2	280
	0.12	0.40	0.48	16.9	250	18.1	290
	0.25	0.35	0.40	17.3	270	19.1	330
	0.22	0.28	0.50	15.8	240	17.9	300
	0.37	0.18	0.45	15.9	260	17.2	310
	0.10	0.50	0.40	16.0	220	19.0	340
	0.28	0.22	0.50	17.2	290	18.3	360
	0.33	0.27	0.40	15.9	260	18.4	330
	0.20	0.30	0.50	15.7	240	19.3	370
	0.15	0.30	0.55	17.5	250	16.4	320
	0.24	0.26	0.50	16.8	280	19.5	380
	0.11	0.47	0.42	16.2	240	17.1	310
	0.35	0.18	0.47	15.1	220	18.6	340
	0.28	0.25	0.47	15.4	260	18.9	345
	0.20	0.40	0.40	13.5	200	12.7	300
	0.24	0.33	0.43	17.8	270	17.5	350
	0.32	0.28	0.40	18.5	290	18.6	360
	0.15	0.50	0.35	15.9	240	20.2	400
	0.40	0.20	0.40	15.4	300	16.0	320
	0.25	0.25	0.50	15.6	240	16.2	310
	0.35	0.35	0.30	15.7	250	15.9	320
	0.28	0.32	0.40	16.4	230	18.1	330
	0.22	0.38	0.40	16.1	260	17.7	340
	0.15	0.55	0.30	17.2	250	19.8	360
	0.30	0.25	0.45	15.1	280	17.8	330
	0.40	0.15	0.45	15.6	220	18.4	340
	0.18	0.42	0.40	15.4	230	17.6	350
	0.25	0.35	0.40	15.2	290	18.9	340
	0.28	0.30	0.42	16.1	240	18.1	360
	0.32	0.28	0.40	15.5	250	18.3	330
	0.24	0.40	0.36	15.9	260	17.7	340
	0.20	0.45	0.35	15.8	270	18.1	350
	0.35	0.25	0.40	16.2	280	17.5	370
	0.33	0.30	0.37	16.0	290	17.3	380
	0.18	0.30	0.52	15.4	230	18.4	290
	0.22	0.32	0.46	15.6	220	17.9	340
	0.38	0.22	0.40	16.0	260	18.2	350
	0.31	0.29	0.40	16.5	277	18.9	377
	0.21	0.37	0.42	16.2	269	18.5	361
	0.29	0.31	0.40	16.8	275	19.0	365
	0.26	0.34	0.40	16.7	265	18.7	355
	0.29	0.32	0.39	16.5	260	18.6	350

Tablo 12. 50 Dügümlü bir ağ için ağırlık katsayılarının seçimi

D.S	we_1	we_2	we_3	Maliyet Fonksiyonu 1 ▷ Eq. 5.1		Maliyet Fonksiyonu 2 ▷ Eq. 5.2	
				mW	Kimlik Doğrulama(s)	mW	Kimlik Doğrulama(s)
50	0.18	0.25	0.57	17.5	500	19.2	550
	0.30	0.15	0.55	18.1	510	18.5	540
	0.12	0.40	0.48	18.7	520	19.8	560
	0.25	0.35	0.40	19.1	530	20.3	570
	0.22	0.28	0.50	17.8	515	19.5	555
	0.37	0.18	0.45	17.9	520	19.0	560
	0.10	0.50	0.40	18.0	510	20.2	580
	0.28	0.22	0.50	19.0	540	19.9	590
	0.33	0.27	0.40	17.9	520	20.1	570
	0.20	0.30	0.50	17.5	515	20.5	600
	0.15	0.30	0.55	19.2	530	18.2	565
	0.24	0.26	0.50	18.7	540	20.7	610
	0.11	0.47	0.42	18.3	525	19.3	575
	0.35	0.18	0.47	17.2	510	20.1	590
	0.28	0.25	0.47	17.5	530	20.2	595
	0.20	0.40	0.40	16.2	447	16.4	474
	0.24	0.33	0.43	19.5	550	19.2	600
	0.32	0.28	0.40	20.1	570	20.1	610
	0.15	0.50	0.35	18.0	520	21.5	650
	0.40	0.20	0.40	17.5	580	18.1	600
	0.25	0.25	0.50	17.7	525	18.3	590
	0.35	0.35	0.30	17.8	535	18.0	600
	0.28	0.32	0.40	18.5	520	20.3	610
	0.22	0.38	0.40	18.1	550	19.9	620
	0.15	0.55	0.30	19.2	540	22.0	640
	0.30	0.25	0.45	17.2	580	20.1	610
	0.40	0.15	0.45	17.7	520	20.3	620
	0.18	0.42	0.40	17.5	525	19.5	630
	0.25	0.35	0.40	17.3	585	20.2	620
	0.28	0.30	0.42	18.1	540	20.3	640
	0.32	0.28	0.40	17.6	550	20.1	610
	0.24	0.40	0.36	17.9	560	19.9	620
	0.20	0.45	0.35	17.8	570	20.3	630
	0.35	0.25	0.40	18.2	580	19.8	650
	0.33	0.30	0.37	18.0	590	19.6	660
	0.18	0.30	0.52	17.5	520	20.5	620
	0.22	0.32	0.46	17.6	510	19.9	630
	0.38	0.22	0.40	18.0	550	20.4	640
	0.43	0.27	0.30	19.0	578	21.2	680
	0.25	0.39	0.36	18.2	561	20.6	670
	0.27	0.35	0.38	18.3	565	20.6	675
	0.29	0.34	0.37	18.5	570	20.8	680
	0.31	0.33	0.36	18.7	575	21.0	685

Tablo 13. 35 ve 50 Düğümlü ağlar için ağırlık katsayılarının seçimi

Maliyet Fonksiyonu 2 ▷ Eq. 5.2							
D.S	we_1	we_2	mW	Kimlik Doğrulama(s)	D.S	mW	Kimlik Doğrulama(s)
35	0.32	0.68	17.9	334	50	20.4	558
	0.54	0.46	19.1	351		22.1	597
	0.65	0.35	19.3	351		21.2	587
	0.28	0.72	20.4	366		23.4	602
	0.49	0.51	19.9	371		24.5	625
	0.75	0.25	20.7	372		24.7	655
	0.22	0.78	21.4	387		26.4	689
	0.61	0.39	12.9	231		16.8	499
	0.41	0.59	13.4	261		17.9	535
	0.33	0.67	13.8	277		18.3	571
	0.45	0.55	13.2	241		18.2	578
	0.23	0.77	21.5	394		26.5	695
	0.72	0.28	20.3	377		24.8	674
	0.50	0.50	13.4	261		18.7	593
	0.38	0.62	13.6	285		18.5	577
	0.66	0.34	18.2	353		21.3	591
	0.55	0.45	13.2	259		18.5	580
	0.29	0.71	20.4	370		23.5	612
	0.67	0.33	18.5	360		21.4	600
	0.43	0.57	13.5	251		18.3	569
	0.62	0.38	13.0	245		17.0	506
	0.21	0.79	21.6	403		26.7	693
	0.77	0.23	20.5	365		24.7	659
	0.52	0.48	13.2	365		18.9	587
	0.64	0.36	14.2	252		18.5	502
	0.30	0.70	20.1	359		22.2	567
	0.73	0.27	20.3	362		24.3	620
	0.73	0.27	20.3	362		24.3	620
	0.46	0.54	19.9	364		25.1	630
	0.46	0.54	19.9	364		25.1	630
	0.57	0.43	19.3	353		24.5	620
	0.48	0.52	19.6	359		25.0	635
	0.60	0.40	12.8	312		16.5	479
	0.35	0.65	21.4	390		27.3	700
	0.25	0.75	20.0	400		27.6	710
	0.69	0.31	19.0	350		24.3	618
	0.36	0.64	21.0	380		27.0	680
	0.53	0.47	19.2	357		23.7	630
	0.63	0.37	20.6	369		25.5	640
	0.22	0.78	20.4	401		27.0	691
	0.39	0.61	20.9	373		25.8	653
	0.24	0.76	21.9	400		27.1	710
	0.44	0.56	19.5	355		24.8	635



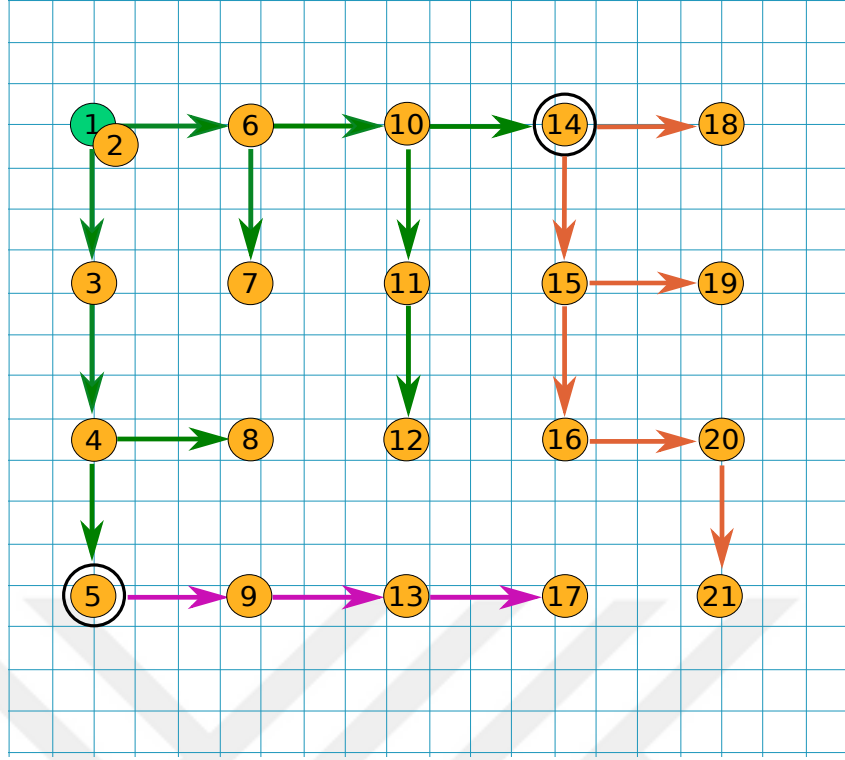
Şekil 58. Düzüm 4 ve 14 için Oracle Proxy-JRC

Önerilen algoritma, düğümün sınır yönlendiriciye olan uzaklığı, komşuluk tablosu, yönlendirme tablosu ve düğümün kalan enerjisini kullanan bir bileşik maliyet fonksiyonu kullanır. Bu bileşik maliyet fonksiyonunun ağırlıkları, Proxy-JRC seçim performansını etkiler. Burada, 20 düğümlü ağ için kimlik doğrulama performansı, farklı maliyet fonksiyonu ağırlıklarıyla analiz edilerek optimum değerler simülasyon ortamında bulunmuştur. Tablo 10, Tablo 11, Tablo 12’de gösterildiği gibi, optimum ağırlık katsayıları w_{e1} , w_{e2} ve w_{e3} için sırasıyla 0.2, 0.4 ve 0.4 olarak bulunmuştur. Önerilen Proxy-JRC seçim algoritmasının değerlendirilmesinde, daha büyük düğüm sayısına sahip ağlar için de aynı katsayılar kullanılmıştır.

Şekil 59, Maliyet fonksiyonu 1’e göre Proxy-JRC seçim algoritmasının kullanıldığı 20 düğümlü ağdaki Proxy-JRC’lerin konumlarını göstermektedir.

Topolojide değişiklikler olduğunda JRC varlıklarının yeniden değerlendirilmesi gerekebilir. Periyodik olarak güncellenen sensör verileriyle, JRC varlıkları yeniden atanır ve gerekli güncellemeler yapılır. Ancak, bu çalışmada topolojinin yeniden değerlendirilmesi ele alınmamıştır. Çalışmanın performans sonuçları, ağ stabil hale geldikten sonra anahtar güncelleme işlemlerini temsil etmektedir.

Merkezi anahtar güncelleme mekanizması, ağdaki tüm cihazların JRC gibi bir anahtar saklama varlığı ile iletişim kurmasını gerektirir. Bu durumda, düğümlerin anahtar

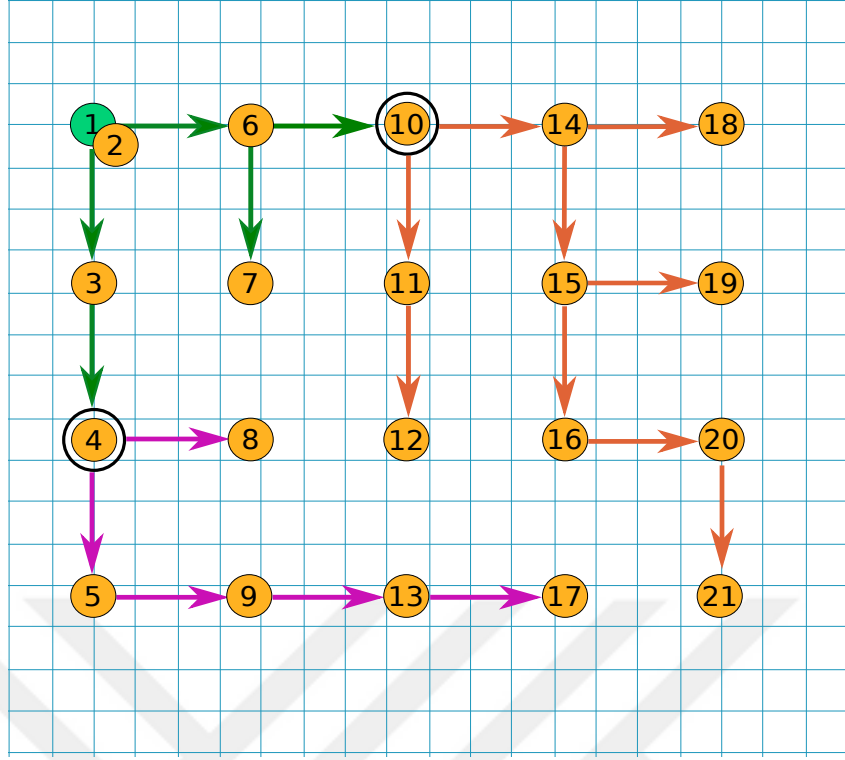


Şekil 59. Maliyet Fonksiyonu denklem 5.1'e göre seçilen Proxy-JRC düğümleri

güncellemeleri, ağ büyüdükçe ölçeklenebilirlik sorunları ortaya çıkarır. Bu nedenle, bu çalışmada sunulan yöntem, literatürde kullanılan merkezi yaklaşımlara kıyasla ağ yönetimi açısından daha etkili performans göstermektedir (Vučinić vd., 2019).

5.3. IETF 6TiSCH Ağları için Hafif Üç Faktörlü Sensör ve Kullanıcı Kimlik Doğrulama Mekanizması

6TiSCH çalışma grubu, bir cihazın 6TiSCH ağına katılmasını sağlamak için gerekli minimum mekanizmaları açıklar (Sarıkaya vd., 2012; Duquennoy vd., 2018; Aydın vd., 2024). Bu mekanizmaların amacı, ağa katılmak isteyen sensör düğümü ile zaten ağın bir parçası olan düğümler arasında güvenli bir iletişim sağlamaktır (Haj-Hassan vd., 2024). Sensör, belirli adımları gerçekleştirmek için bağlantı katmanı anahtarlarını ayarlayarak ağ ile senkronize edilir. Ağ ile senkronize olan düğüm, kanal atlama bilgisi, zaman dilimi süresi gibi parametreleri öğrenir. Bağlantı katmanı paketlerini güvenli hale getirirken, IEEE 802.15.4'te tanımlanan standart güvenlik mekanizmaları kullanılır. Standart, iki farklı şifreleme anahtarı (K1, K2) kullanarak doğrulama ve şifreleme işlemlerini gerçekleştirir. K1 anahtarı, işaretçileri doğrulamak için kullanılır (başlangıçta tüm cihazlara atanmış olduğu varsayılır). K2 anahtarı, Veri ve Onay çerçevelerini kimlik doğrulama ve şifrelemek için

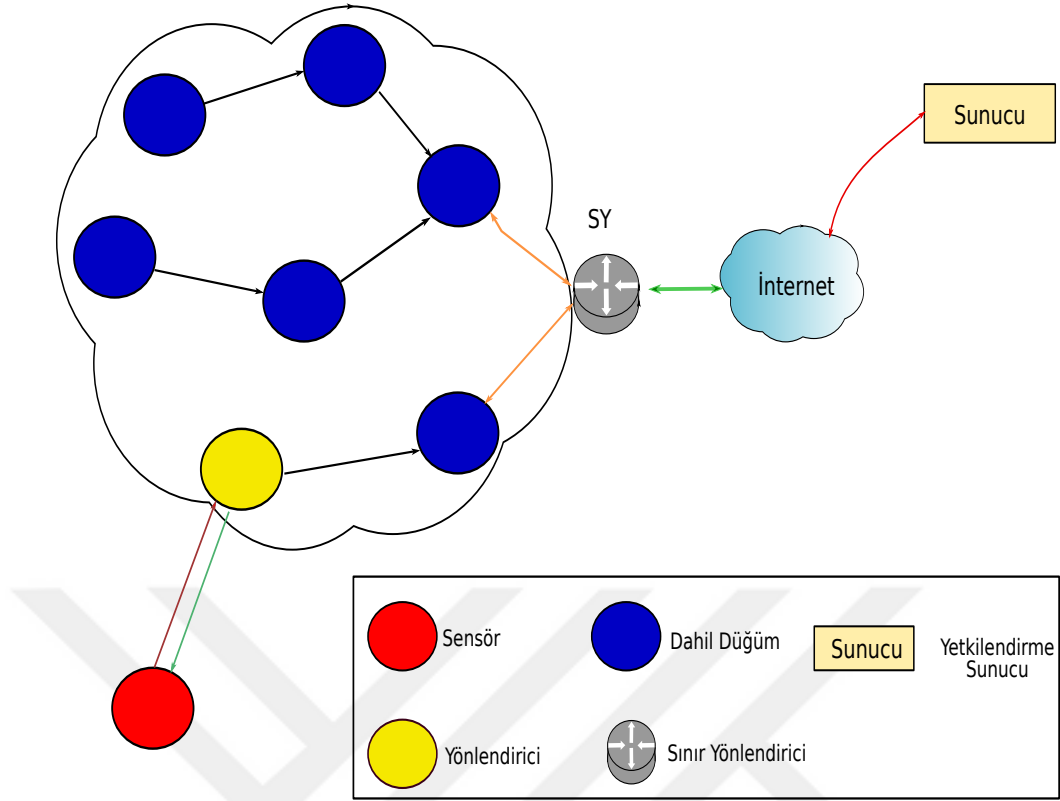


Şekil 60. Maliyet Fonksiyonu denklem 5.2'ye göre seçilen Proxy-JRC düğümleri

kullanılır. Bu anahtarlar, sensörlere önceden yüklenebilir veya anahtar dağıtım aşamasında cihazlara iletebilir. Cihazdan sunucuya yapılan istekler üzerine, K2 anahtar değeri ve diğer kimlik doğrulama parametreleri cihazlara gönderilir. Düğüm, sunucudan aldığı yanıt mesajını doğruladıktan sonra ağa dahil edilir ve komşuları ile iletişim kurar. Şekil 61, katılım sürecindeki aktörleri göstermektedir.

Tezde, IETF 6TiSCH ağları için alternatif üç adımlı kimlik doğrulama ve anahtar anlaşma şemasını sunulmuştur. Şekil 62, sensör cihazlarını içeren bir ağ modelini göstermektedir. Sistem dört tür varlık içerir: kullanıcı (K), sınır yönlendirici (SY), sunucu (S) ve kaynak kısıtlı cihazlar (C). SY ve merkezi bir varlık tarafından kolaylaştırılan karşılıklı kimlik doğrulama işleminden sonra, kullanıcılara sensör cihazları tarafından iletilen verileri izlemek amacıyla ağa erişim izni verilir. Kimlik doğrulama aşamasında, cihaz ağa katılır ve verilerini güvenli kanal üzerinden bulut tabanlı sunucuya iletir. Kullanıcı, verileri güvenli bir kanal üzerinden bulut tabanlı sunucu aracılığıyla veya doğrudan cihazdan alabilir.

Önerilen şema, sistem kurulumu, kullanıcı kaydı, giriş işlemi, kimlik doğrulama prosedürü ve şifre değiştirme olmak üzere beş aşamadan oluşmaktadır. Şemada kullanılan kısaltmalar ve açıklamaları Tablo 14'te verilmiştir.



Şekil 61. Ağa katılma sürecindeki aktörler

5.3.1. Sistem Kurulum Aşaması

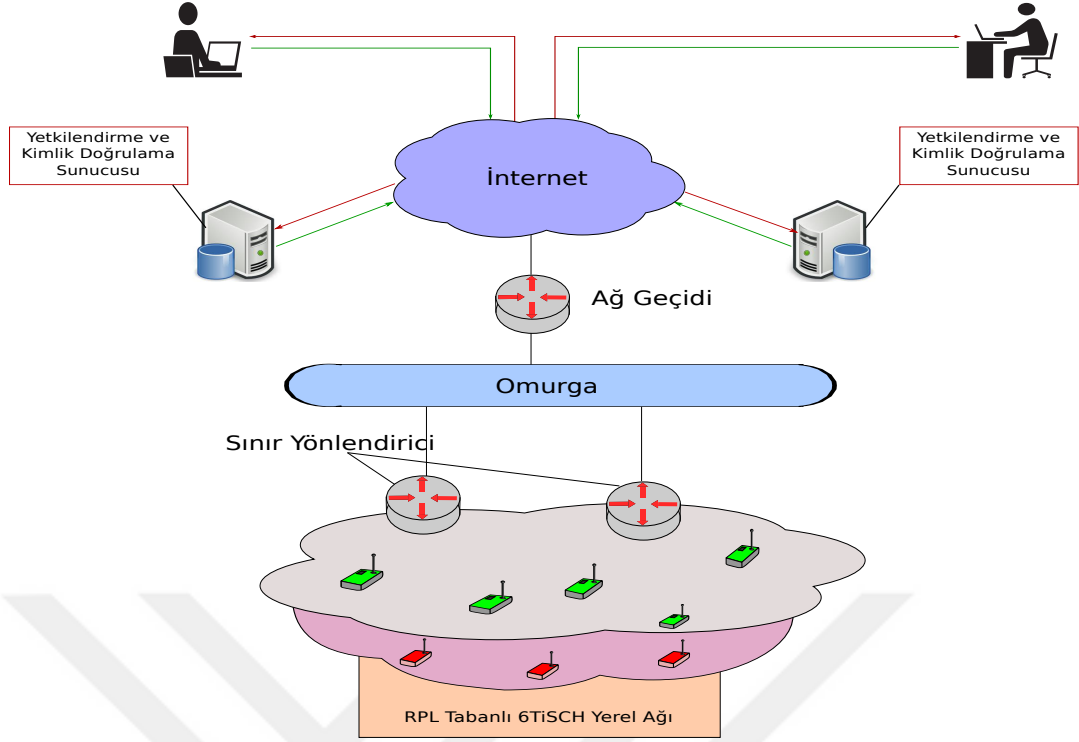
Bu aşama, sensör düğümlerine kimlik doğrulama parametrelerinin sunucu (S - Kimlik Doğrulama ve Yetkilendirme - Tam güvenilen varlık) tarafından programlanmasını içerir ve bu işlem dağıtım öncesinde gerçekleştirilir. Ayrıca, bu adımda sunucu, hem kullanıcılar hem de cihazlar için gerekli sertifikaları ve kimlik doğrulama belirteçlerini oluşturur. Sunucu, kullanıcılar (k) ve cihazlar (c) için sadece sunucu tarafından bilinen iki global anahtar değeri (GK_{k-c} ve GK_{k-s}) rastgele olarak oluşturur. Sunucu, cihazların ve kullanıcıların kimlik bilgilerini kullanarak her varlığa özel dinamik kimlik bilgileri oluşturur. Kullanıcı için dinamik kimlik bilgisi, sunucu tarafından belirlenen şifre ve kimlik bilgilerinin hash fonksiyonundan geçirilmesiyle elde edilir. Bu adımda, hem kullanıcı hem de sensör düğümü tarafından oluşturulan şifre değerleri hash fonksiyonundan geçirilir ve belirtilen cihazların hafızalarında saklanır. Bu şekilde, cihazlar fiziksel olarak ele geçirilse bile saldırganlar verilerin saf haline erişemezler.

$$DID_k = H(ID_k || H(PWD_k)) \quad (5.3)$$

Tablo 14. Önerilen protokolde kullanılan tanımlar listesi

Kısaltmalar	Tanım
S	Sunucu (Yetkilendirme ve Kimlik Doğrulama Birimi)
SY	Sınır Yönlendirici
C	Cihaz
K	Kullanıcı
ID_c	Cihaz D_i kimliği ID_c
ID_k	Kullanıcı K_i kimliği ID_k
DID_c	Dinamik cihaz kimliği ID_c
DID_k	Dinamik kullanıcı kimliği ID_k
PWD_c	Cihaz şifresi
PWD_k	Kullanıcı şifresi
$H(PWD_c)$	Cihaz şifresinin hash değeri
$H(PWD_k)$	Kullanıcı şifresinin hash değeri
PK_{k-s}	Kullanıcı ve sunucu arasındaki özel anahtar
GK_{k-s}	Kullanıcı ve sunucu arasındaki global simetrik anahtar
GK_{k-c}	Cihaz ve sunucu arasındaki global simetrik anahtar
PK_{s-c}	Kimlik doğrulamadan sonra sunucu ve cihaz arasındaki simetrik özel anahtar (16 bayt)
K_1	EB doğrulama anahtarı (16 bayt)
K_2	Yetkilendirici tarafından sağlanan özel anahtar
TS	Veri tazeliği için zaman damgası
$\oplus$	XOR Operatörü
$\parallel$	Birleştirme operatörü
Δt	Sabit iletim gecikmesi
$H(.)$	Tek yönlü hash fonksiyonu (SHA-1)
OSCORE	CoAP üzerinde güvenli veri iletimi sağlayan uçtan uca güvenlik protokolü
IPsec	IP katmanında çalışan güvenli ağ protokolü

$$DID_c = H(ID_c \parallel H(PWD_c)) \quad (5.4)$$



Şekil 62. 6TiSCH ağlarında kullanıcı-sensör-kimlik doğrulama sunucu kontrol sistemine bir örnek

$$XS_{k-s} = H(DID_k || K_{k-s}) \quad (5.5)$$

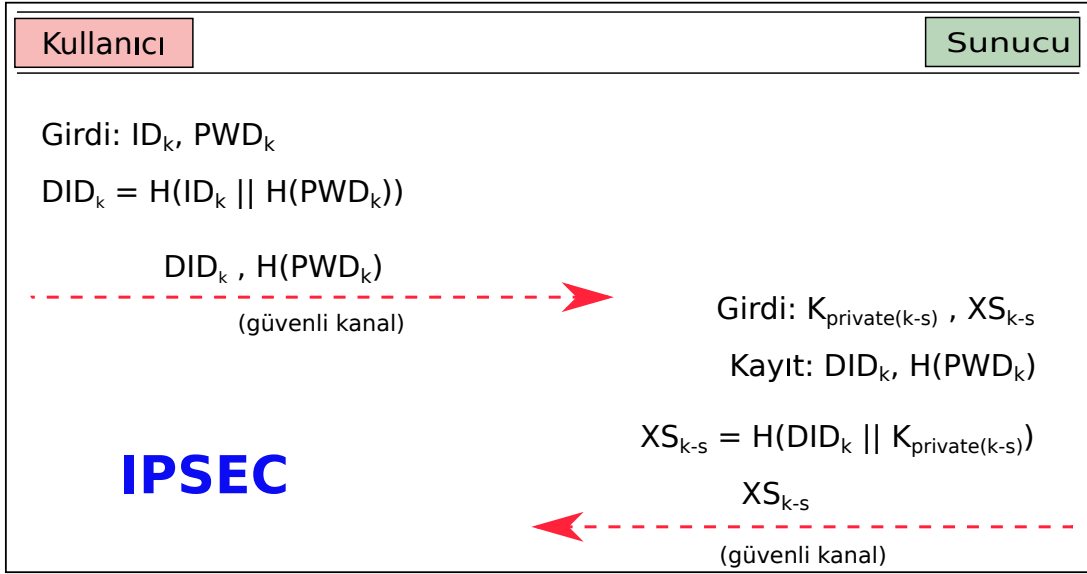
Sensör ve kullanıcı cihazlarının dinamik kimlik bilgileri belirlendikten sonra, bu bilgiler güvenli bir şekilde cihaz hafızasına kaydedilir ve cihazlar, ağda yerlerini almak üzere hedef alana yerleştirilir. Oluşturulan dinamik kimlik bilgileri, sensörlerin ve kullanıcıların sunucuyu, sunucunun ise bu cihazları doğrulamasını sağlar ve cihaz ile kullanıcı arasındaki güvenlik gereksinimini karşılar. XS_{k-s} değeri, sunucu tarafından belirlenen bir parametredir ve hem kullanıcıda hem de sunucuda saklanır. Bu verinin gerçek değeri yalnızca yasal cihazlar tarafından bilinir.

5.3.2. Sistem Kayıt Aşaması

Ağdan veri erişimi sağlamak isteyen kullanıcı, Şekil 63'te gösterildiği gibi gerekli kayıt adımlarını sunucu üzerinden takip eder.

Adım 1: Kullanıcı, rastgele bir sayı (PWD_k) seçtikten sonra dinamik kimlik bilgisi ve şifre oluşturmaktadır.

Adım 2: Sunucu, özel bir anahtar değeri (PK_{k-s}) hesaplayarak bu değeri kullanıcının cihazına kaydeder. Bu özel anahtar değeri, sunucu ve kullanıcı arasındaki iletişimi güvence



Şekil 63. Kullanıcı kayıt şeması

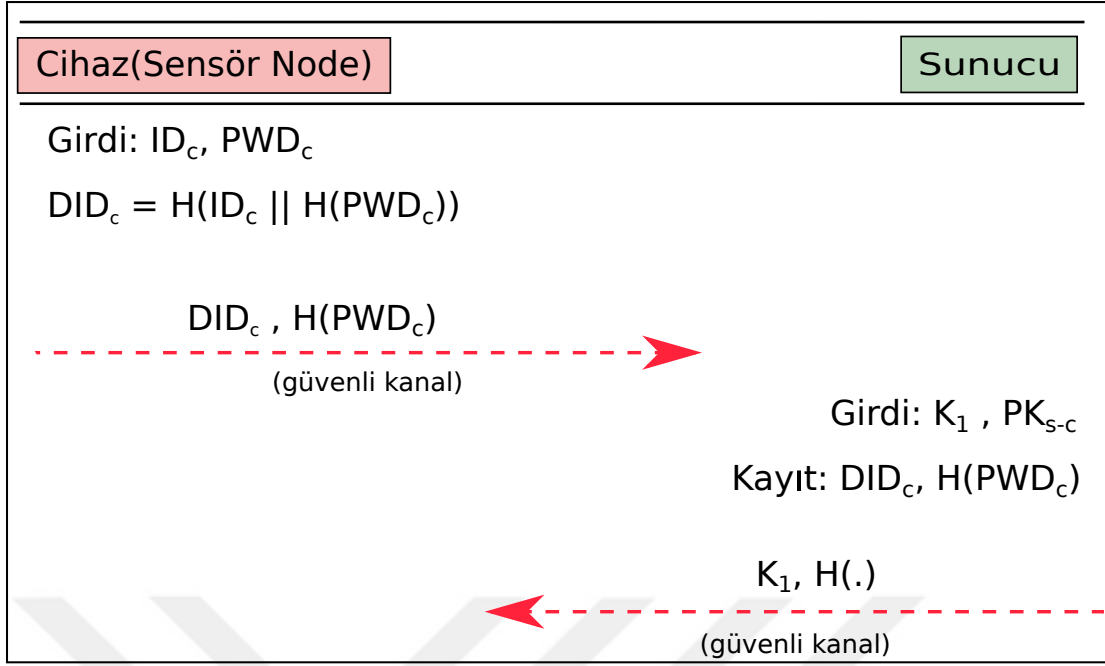
altına almak ve çeşitli işlemleri gerçekleştirmek için kullanılır. Sunucu ile kullanıcı arasındaki güvenli iletişim IPSEC kullanılarak sağlanır. Hem sunucu hem de kullanıcı bu değeri kendi sistemlerinde saklar.

Adım 3: Kullanıcı, sunucudan aldığı bilgilerle birlikte giriş talebinde bulunmak için DID_k, M_k paketini gönderir.

Sensör düğümlerinin ağ üzerinde güvenli bir şekilde iletişim kurabilmeleri için, bu cihazların sunuculardan güvenli bir kanal üzerinden doğrulama parametrelerini almaları gerekmektedir. Bu güvenli kanalı sağlamak için cihazlar, dağıtım öncesinde fabrikalarda veya kurulum yerlerinde $K_1, H(.)$ ile programlanır. Bu, sunucu ile düğüm arasında paylaşılan kimlik doğrulama parametreleri yardımıyla sistem güvenlik kontrolünü sağlamayı gerektirir. Cihazlar güvensiz bir ağ ortamına yerleştirildiğinde, daha önce güvenli ortamdan elde edilen kimlik doğrulama parametreleri, düğümün doğrulamasını gerçekleştirmek için kullanılır. Şekil 64, sensör düğümlerinin ağ üzerindeki kayıt aşamasını göstermektedir.

Bu aşamada, cihazlar sunucudan güvenli bir kanal üzerinden doğrulama parametrelerini alır ve bu parametreler kullanılarak güvenlik kontrolleri sağlanır. Güvenli iletişim kanalı, OSCORE protokolü kullanılarak sağlanmaktadır. Bu protokol, mesajların gizliliğini ve bütünlüğünü koruyarak, ortadaki adam saldırıları ve hizmet reddi saldırılarına karşı koruma sağlar. OSCORE, veri paketlerini şifreler ve doğrular, böylece sadece yetkili taraflar bu verilere erişebilir ve onları değiştirebilir. Bu, sunucu ve düğüm arasındaki iletişimin güvenliğini artırır ve cihazların güvenilir bir şekilde ağa katılmasını sağlar.

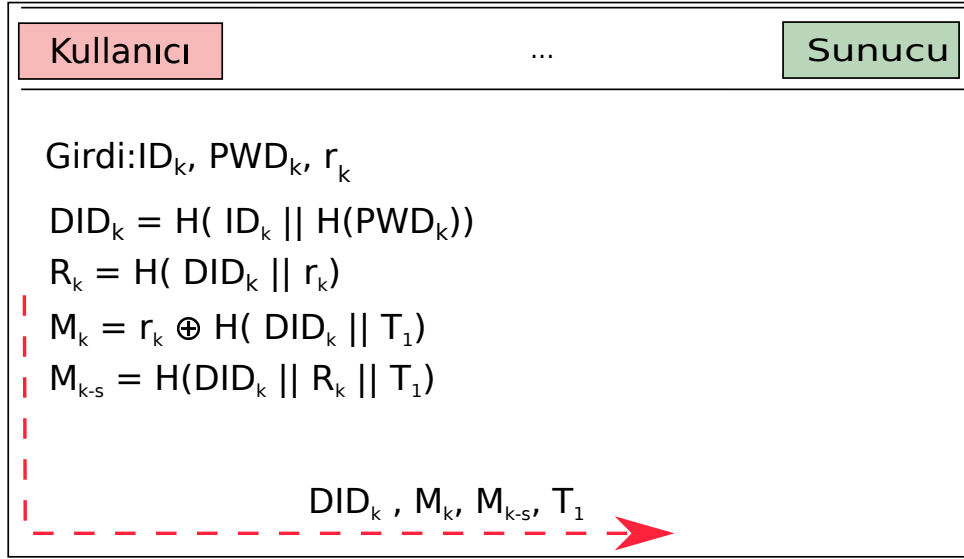
Adım 1: Rastgele bir gizli değer (geçerli şifre) seçildikten sonra, sensör düğümü dinamik kimlik bilgisi (DID_c) ve şifre (PWD_c) oluşturur. Şifre hash fonksiyonu yardımıyla 20 baytlık bir çıktı ($H(PWD_c)$) üretir. Bu veri, güvenli kanal üzerinden sunucuya iletilir.



Şekil 64. Cihaz (sensör düğümü) kayıt şeması

Adım 2: Sunucu, birinci adımda elde edilen sensörün dinamik kimlik bilgisi ve şifresinin hash değerini hafızasına kaydeder. Sunucu, EB paketlerini şifrelemek ve şifresini çözmek için özel anahtar değerini (K_1) hesaplar. Hesaplanan anahtar değeri, cihazların ağa katılmasına yardımcı olmak için üretim aşamasında güvenli bir şekilde sensör düğümünün hafızasına yazılır. Bu değer, hem sunucuda hem de kullanıcının tarafında kaydedilir veya kayıt altına alınır.

Tablo 15’te OSCORE ve IPsec protokollerinin sunduğu güvenlik özelliklerinin karşılaştırılması verilmiştir. OSCORE, kaynak kısıtlı cihazlar için optimize edilmiş bir protokoldür ve özellikle IoT cihazlarında yaygın olarak kullanılır. Uygulama katmanında çalışarak uçtan uca mesaj şifreleme ve kimlik doğrulama sağlar. Mesajların bütünlüğünü korur ve yeniden oynama saldırılarına karşı koruma sağlar. OSCORE’nin düşük işlem yükü sayesinde, IoT cihazlarının sınırlı işlem gücü ve enerji kaynaklarını verimli kullanması mümkün olur. Bununla birlikte, OSCORE yalnızca mesaj yükünü şifreler ve başlıklar açıkta kalır. Öte yandan, IPsec protokolü ağ katmanında çalışır ve daha geniş ağ ortamlarında, güçlü cihazlar ve ağ altyapılarında kullanılır. IPsec, tüm IP trafiğini koruyarak güvenli iletişim sağlar. IPsec hem uçtan uca hem de tünel şifreleme yapabilme yeteneğine sahiptir ve geniş güvenlik özellikleri sunar.



Şekil 65. Kullanıcı giriş şeması

5.3.3. Sistem Giriş Aşaması

Kullanıcı, ID_k , PWD_k ve önceden tanımlanmış özel bir parametre (r_k) kullanarak 6tisch ağına güvenli bir şekilde dahil olmak için giriş aşamasını başlatır. Önerilen şema için giriş aşaması Şekil 65'te gösterilmiştir. Bu aşama aşağıdaki adımlardan oluşur.

Adım 1: Kullanıcı, ID_k , PWD_k ve r_k parametrelerini girdi olarak kullanır.

Adım 2: DID_k değeri hesaplanır. Dinamik kimlik bilgisi, hash fonksiyonuna önceden belirlenmiş r_k parametresi ile birlikte girdi olarak verilir.

$$DID_k = H(ID_k || H(PWD_k)) \quad (5.6)$$

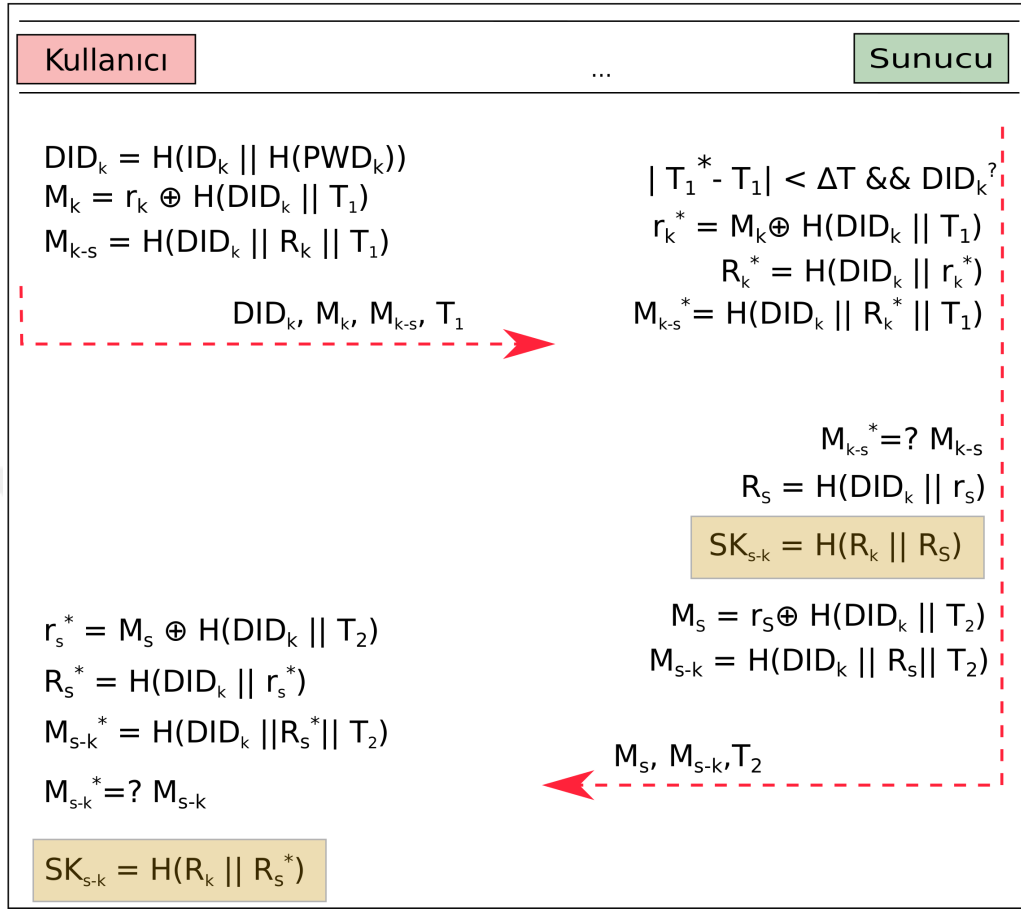
Ardından R_k , M_k , M_{k-s} değerleri hesaplanır.

$$R_k = H(DID_k || r_k) \quad (5.7)$$

$$M_k = r_k \oplus H(DID_k || T_1) \quad (5.8)$$

$$M_{k-s} = H(DID_k || R_k || T_1) \quad (5.9)$$

Adım 3: Hesaplanan DID_k , M_k , M_{k-s} ve T_1 değerleri doğrulama için sunucuya gönderilir.



Şekil 66. Kullanıcı kimlik doğrulama şeması

Bu aşamada, sunucu ve kullanıcı arasındaki iletişimin güvenliğini sağlamak amacıyla IPSEC kullanılmaktadır. Bu, ortadaki adam saldırılarına karşı ek bir güvenlik katmanını ekler. Özellikle, açık giden zaman damgasının T_1 değiştirilmesi gibi saldırılar IPSEC kullanılarak önlenabilir. Böylece, ortadaki adam saldırısının sebep olduğu DoS (Hizmet Reddi) saldırıları engellenmiş olur.

5.3.4. Kullanıcı ve Cihaz Kimlik Doğrulama Aşaması

Kimlik doğrulama sürecinin ilk aşaması, sensör cihazının sunucuya gönderdiği ağa katılım talebi ile başlar. Bu aşamada, sensör cihazının kimliği ve güvenlik parametreleri doğrulanır. Bu işlemlerin ardından, sensör cihazlarının ağa güvenli bir şekilde dahil edilmesi sağlanır ve bu, sunucu aracılığıyla kullanıcı ve cihaz arasında güvenli bir iletişim kanalı oluşturur. Kullanıcı ve sunucu arasında karşılıklı kimlik doğrulama ve oturum anahtarı

anlaşmasını sağlamak için bu aşama, aşağıda belirtilen bir dizi ardışık adımı içerir. Kullanıcı ve sensör cihazı için kimlik doğrulama aşamaları ayrı ayrı incelenmiştir. Şekil 66, kullanıcı kimlik doğrulama mekanizmasının aşamalarını göstermektedir.

Adım 1: Sunucu, kullanıcının gönderdiği parametreler için önce zaman damgasını kontrol eder ($|T_1^* - T_1| < \Delta t$) ve veritabanında DID_k değerini sorgular. Kullanıcının yasal bir cihaz olduğunu doğrulamak için aşağıdaki hesaplamaları yapar.

$$r_k^* = M_k \oplus H(DID_k || T_1) \quad (5.10)$$

$$R_k^* = H(DID_k || r_k^*) \quad (5.11)$$

$$M_{k-s}^* = H(DID_k || R_k^* || T_1) \quad (5.12)$$

Hesaplanan M_{k-s}^* değeri, kullanıcıdan alınan M_{k-s} parametresi ile karşılaştırılır. Bu koşul sağlanmazsa, sunucu bu aşamayı sonlandırır. Aksi takdirde, sunucu K'yi yetkili bir kullanıcı olarak kabul eder. Sunucu, daha önce tanımlanmış r_k değeri ile R_s parametresini hesaplar ve kullanıcı ile sunucu arasında bir oturum anahtarı oluşturur.

$$R_s = H(DID_k || r_s) \quad (5.13)$$

$$SK_{s-k} = H(R_k || R_s) \quad (5.14)$$

Oluşturulan oturum anahtarı, kullanıcı ile sunucu arasındaki iletişimin artık bu anahtar değeri ile güvenli şekilde yapılabileceğini gösterir. Bu adımdan sonra, sunucu kendisini kullanıcıya yasal bir varlık olarak kanıtlamalı ve karşılıklı iletişim için oturum anahtarı kullanıcısı tarafından hesaplanmalıdır.

$$M_s = r_s \oplus H(DID_k || T_2) \quad (5.15)$$

$$M_{s-k} = H(DID_k || R_s || T_2) \quad (5.16)$$

Yukarıda hesaplanan değerler $\langle M_s, M_{s-k}, T_2 \rangle$ kullanıcıya gönderilir.

Adım 2: Kullanıcı, sunucudan alınan parametreler için zaman damgasını kontrol eder ($|T_2^* - T_2| < \Delta t$) ve yeni M_{k-s}^* değerini hesaplar. Hesaplanan bu değer, gönderilen M_{k-s} ile karşılaştırılır. Bu koşul sağlanmazsa, kullanıcı aşamayı sonlandırır. İşlem başarılı olursa, kullanıcı sunucunun yasal bir varlık olduğunu onaylar ve oturum anahtarını oluşturmak için aşağıdaki hesaplamaları yapar.

$$r_s^* = M_s \oplus H(DID_k || T_2) \quad (5.17)$$

$$R_s^* = H(DID_k || r_s^*) \quad (5.18)$$

$$M_{s-k}^* = H(DID_k || R_s^* || T_2) \quad (5.19)$$

M_{s-k}^* ve M_{s-k} eşit olduğunda, oturum anahtarı $SK_{s-k} = H(R_k || R_s^*)$ olarak oluşturulur ve artık kullanıcı ile sunucu arasında hem kimlik doğrulama hem de ortak anahtar değeri kullanımı sağlanır.

Burada, IETF 6TiSCH protokolü için geliştirilmiş bir önyükleme (bootstrapping) süreci tanıtılmaktadır. Bu süreç, sunucu ile karşılıklı kimlik doğrulama için dinamik kimlikler ve cihaz şifrelerini içermektedir. Önyükleme sürecindeki iyileştirmeler, RPL'de belirtilen güvenlik mekanizmalarını kullanarak ağın güvenli bir şekilde başlatılmasını ve yönetilmesini sağlar. Aşağıda, sensör cihazlarının ağa katılımı için oluşturulan prosedürlerin adımları verilmiştir:

Adım 1: Sensör cihazı AUT_REQ, M_{c-s} , DID_c ve T_1 parametrelerini girdi olarak kullanır.

$$M_{c-s} = E(H(DID_c || H(PWD_c) || T_1)) \quad (5.20)$$

Adım 2: Vekil (proxy) düğümü, ağa katılmak isteyen cihazın ağa katılım paketlerini doğrudan sunucuya yönlendirir.

Adım 3: Sunucu, sensörden aldığı parametreler için önce zaman damgasını kontrol eder ($|T_1^* - T_1| < \Delta T$) ve hafızasında DID_c değerini sorgular. Zaman damgasının kontrolü, tekrar ataklarına karşı koruma sağlar. Bir eşleşme sağlanırsa, M_{c-s} paket içeriğini yeniden hesaplayarak bir karşılaştırma yapar.

$$M_{c-s}^* = E(H(DID_c^* || H(PWD_c)^* || T_1)) \quad (5.21)$$

Hesaplanan M_{c-s}^* değeri alınan M_{c-s} değerine eşit değilse, sunucu cihaz kimlik doğrulamasını sonlandırır. Bu adım, saldırganın hizmet engelleme saldırısını engellemeye yardımcı olur. İşlem başarılı olursa, sunucu katılan sensör düğümünün yasal olduğunu kabul eder ve paket içeriği $E(HELLO)$, M_{s-c} , $E(PK)$ ve T_2 'yi katılan düğüme gönderir. Burada mesaj, katılan düğümün ebeveyni olan ata düğüme gönderilir.

$$M_{s-c}^* = E(H(DID_c || H(PWD_c))) \quad (5.22)$$

Adım 4: Ata düğüm, sunucudan aldığı parametreler için zaman damgasını kontrol eder ($|T_2^* - T_2| < \Delta T$) ve hafızasında katılan cihazın bağlantı adresini arar. Sunucunun adresini mesaja ekler ve paketi sensör düğüme yönlendirir. Ağa katılma isteğinde bulunan düğüme iletilen paket içeriği:

$$\leq E(HELLO), M_{s-c}, server_address, E(PK), T_3 \geq$$

Adım 5: Ağa dahil olma isteğinde bulunan düğüm, ata düğümden aldığı parametreler için zaman damgasını kontrol eder ($|T_2^* - T_2| < \Delta T$) ve değeri yeniden hesaplar ve karşılaştırır.

$$M_{s-c}^* = H(DID_c^* || H(PWD_c)^*) \quad (5.23)$$

M_{s-c}^* , M_{s-c} değerine eşit değilse, sensör cihazı kimlik doğrulamasını sonlandırır ve katılma istek paketini yeniden gönderir. İşlem başarılı olursa, sensör sunucunun istekte bulunan yasal bir varlık olduğunu kabul eder ve şifreli özel anahtar değerini açık anahtar değeri ile şifreler.

$D(PK)$, değeri sunucu ile sensör cihazı arasında artık paylaşılabilecek olan özel ortak anahtar değeridir. Şifrelenmiş $D(HELLO)$ çerçevesi doğrulanırsa, sensör düğümü RPL ağının bir

parçası olur. Artık düğüm ve sunucu birbirini karşılıklı olarak doğrulamış olduğundan, güvenli bir şekilde iletişim kurabilirler. $PK_{(K_2)}$, veri ve ACK trafiğini şifrelemek ve doğrulamak için kullanılır.

5.3.5. Sistem Anahtarı Güncelleme Aşaması

ASN değeri, sunucu tarafından oluşturulan ağda anahtar değişimi için kullanılır. Bunun temel nedeni, ASN değerinin ağdaki paylaşılan küresel bir değer olması ve her zaman aralığında artmasıdır. Bu referans değeri ile ağdaki güncellemeler $ASN + Time$ kapsamında hesaplanır. Algoritmanın adımları Şekil 67’de ifade edilmiştir ve aşağıda bu adımların açıklamaları verilmiştir.

Adım 1: Sunucu tarafından belirlenen yeni anahtar değeri (K_2^{new}), mevcut ASN değerine bir süre eklenerek ve XOR işlemine tabi tutularak eklenir.

$$M_s = K_2^{new} \oplus (ASN + \Delta_{ASN}) \oplus T_1 \quad (5.24)$$

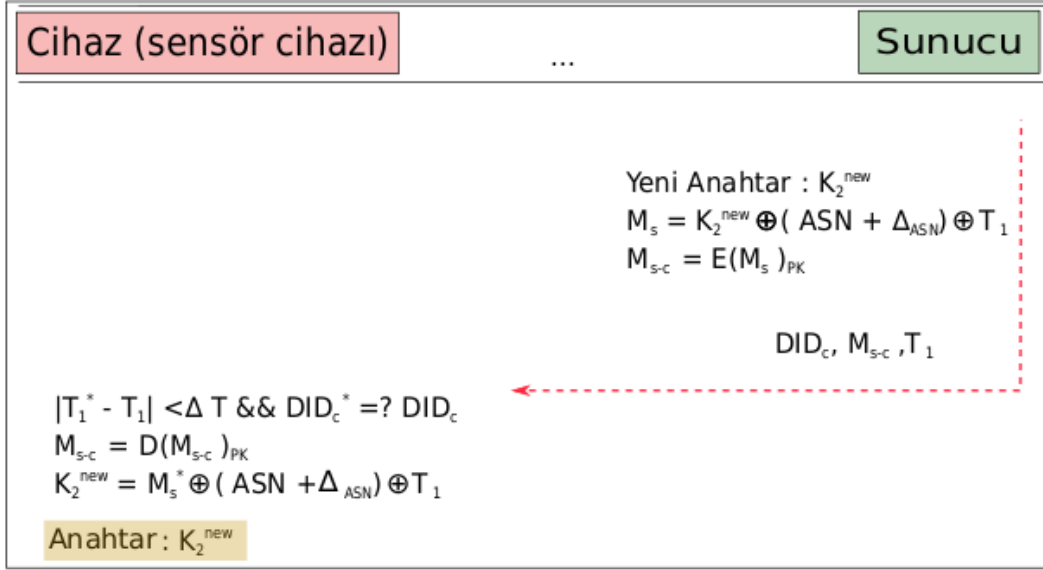
$$M_{s-d} = E(M_s)_{PK} \quad (5.25)$$

Δ_{ASN} değeri, ağdaki en uzak noktada bulunan sensör düğümünün sunucu tarafından gönderilen mesajı alması için geçen süreyi ifade eder. DID_c , M_{s-c} , T_1 parametreleri, ilgili sensör düğümüne iletmek üzere ağa gönderilir.

Adım 2: Sensör düğümü, sunucudan aldığı parametreler için zaman damgasını kontrol eder ($|T_1^* - T_1| < \Delta T$) ve istekte bulunan cihazın hafızasında olup olmadığını kontrol eder. Sensör sunucudan gelen mesajı aldıktan sonra yeni anahtar değerini hesaplar. Ancak, bu değer belirtilen zaman aralığına ulaşana kadar kullanılamaz. Bu anahtar değeri, belirlenen zaman aralığına ulaşana kadar bekletilir. Bu işlem, sensör düğümünün ve ağın senkronizasyon kaybını önler ve tüm düğümlerin aynı anda yeni anahtarı kullanmasını sağlar.

$$M_s = D(M_{s-c})_{PK} \quad (5.26)$$

$$K_2^{new} = M_s \oplus (ASN + \Delta_{ASN}) \oplus T_1 \quad (5.27)$$



Şekil 67. Cihaz anahtar güncelleme şeması

Sensör cihazı ile sunucu arasındaki anahtar güncelleme adımı, ağda ekstra trafik yükü pahasına dinamik bir koruma sağlar. Diğer yandan, veri paketlerini şifrelemek için aynı anahtarın kullanılması, saldırganlar tarafından istismar edilebilecek bir güvenlik açığı oluşturur. Önerilen yaklaşım, anahtar tahmin saldırısını ortadan kaldırır. Ayrıca, simetrik şifreleme ve karma tekniklerin kullanılması, düşük bellek gereksinimi ile düşük güçlü sensör cihazları için uygun hale getirir. Şekil 68, kullanıcı için anahtar güncelleme aşamasını göstermektedir. Bu aşamada, kullanıcı ile sunucu arasındaki oturum anahtarı yeni değerini alır.

Kullanıcı ile sunucu arasındaki anahtar güncelleme sürecinin adımları özetlenmiştir.

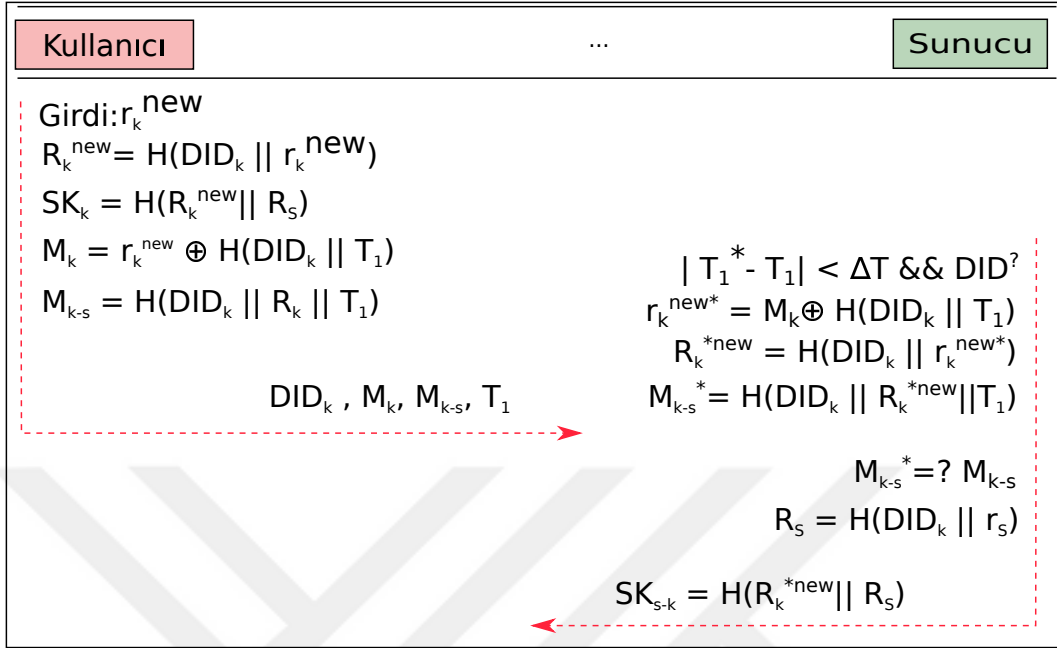
Adım 1: Kullanıcı, yeni belirlenen r_k^{new} parametresi ile R_k değerini yeniden hesaplar. Ardından M_k , M_{k-s} değerleri hesaplanır.

$$R_k = H(DID_k || r_k^{new}) \quad (5.28)$$

$$SK_{s-k} = H(R_k || R_s) \quad (5.29)$$

$$M_k = r_k^{new} \oplus H(DID_k || T_1) \quad (5.30)$$

$$M_{k-s} = H(DID_k || R_k || T_1) \quad (5.31)$$



Şekil 68. Kullanıcı anahtar güncelleme şeması

Adım 2: Hesaplanan DID_k, M_k, M_{k-s}, T_1 değerleri doğrulama için sunucuya gönderilir.

Adım 3: Sunucu, kullanıcıdan aldığı parametreler için zaman damgasını kontrol eder ($|T_1^* - T_1| < \Delta T$) ve veritabanındaki DID_k değerini sorgular.

$$r_k^{new*} = M_k \oplus H(DID_k || T_1) \quad (5.32)$$

$$R_k^{new} = H(DID_k || r_k^{new*}) \quad (5.33)$$

$$M_{k-s} = H(DID_k || R_k^{new} || T_1) \quad (5.34)$$

Hesaplanan M_{k-s}^* değeri, kullanıcıdan alınan M_{k-s} parametresi ile karşılaştırılır. Koşul sağlanmazsa, sunucu bu aşamayı sonlandırır. Ancak, koşul sağlanırsa, sunucu kullanıcı k'yı yasal bir kullanıcı olarak kabul eder. Ardından, sunucu yeni hesaplanan R_k^{new} değeri ile kullanıcı ve sunucu arasında bir oturum anahtarı oluşturur.

$$R_s = H(DID_k || r_s) \quad (5.35)$$

$$SK_{s-k} = H(R_k^{new} || R_s) \quad (5.36)$$

Oluşturulan oturum anahtarı, kullanıcı ile sunucu arasındaki iletişimin artık bu anahtar değeri ile gizli bir şekilde yapılabileceğini gösterir.



Tablo 15. OSCORE ve IPsec Protokolleri Karşılaştırması

Güvenlik Özelliği	OSCORE	IPsec
Şifreleme	Uçtan uca mesaj şifreleme sağlar.	IP katmanında uçtan uca veya tünel şifreleme sağlar (ESP).
Kimlik Doğrulama	Mesaj bazlı kimlik doğrulama sağlar.	IP başlıklarını ve yükü kimlik doğrulamak için AH veya ESP kullanır.
Bütünlük	Mesajın bütünlüğünü korur, değişiklikleri tespit eder.	Bütünlük sağlamak için AH veya ESP kullanır.
Yeniden Oynama Saldırısı Koruması	Yeniden oynama saldırılarına karşı koruma sağlar (mesaj sayacı).	Yeniden oynama saldırılarına karşı AH veya ESP ile koruma sağlar.
Çalışma Katmanı	Uygulama katmanında çalışır (RESTful API'ler için optimize edilmiştir).	Ağ katmanında (IP katmanı) çalışır.
Bağlam Yönetimi	Tek bir iletişim oturumu boyunca güvenlik bağlamı sağlar.	Tüm IP trafiği için güvenlik bağlamı sağlar (oturum bazlı).
Veri Gizliliği	Yalnızca mesaj yükünü şifreler.	IP başlıkları hariç tüm yükü şifreleyebilir (ESP).
Veri Kaynağı Doğrulama	Mesajın orijinini doğrular.	Hem başlık hem de yük orijini doğrulanabilir.
Bağlantısız Güvenlik	Her mesaj bağımsız olarak korunur.	Özellikle AH ile bağlantısız güvenlik sağlar.
Kullanım Alanı	Kaynak kısıtlı IoT cihazları için optimize edilmiştir.	Daha geniş ağ ortamlarında ve güçlü cihazlar için kullanılır.
Anahtar Yönetimi	Önceden paylaşılan anahtarlar veya anahtar değişim mekanizmaları ile çalışır.	IKE ile anahtar yönetimi sağlar.
Ağ Trafiği Yönetimi	Uygulama katmanındaki mesajlar şifrelenir.	Tüm IP trafiği korunur, ağ katmanındaki her şey güvence altına alınır.
Performans	Düşük işlem yükü sağlar.	Ağ katmanında daha yüksek işlem yükü yaratır, daha fazla kaynak kullanır.
Protokol Destekleri	CoAP ve RESTful protokoller ile entegre çalışır.	IP ile çalışan tüm protokollerle çalışır (TCP, UDP, vb.).

6. BULGULAR VE İRDELEME

Bu bölümde, çalışmanın sonuçları detaylı bir şekilde analiz edilmektedir. Çeşitli ağ senaryolarını analiz etmek için Contiki-Cooja simülatörü kullanılmıştır. Bu senaryolarda düğüm sayıları değişkenlik göstermektedir. Her bir senaryo, farklı seed¹ değerleri ile 5 kez çalıştırılmış ve her çalışmanın sonuçlarının ortalaması alınmıştır. Bu yaklaşımın temel nedeni, rastgelelik içeren simülasyonların sonuçlarında oluşabilecek değişkenliği minimize etmektir. Rastgele seed değerleri, simülasyonun her seferinde farklı başlangıç koşullarıyla çalıştırılmasını sağlar ve bu da sonuçların farklılık göstermesine neden olabilir. Dolayısıyla, her senaryonun birden fazla kez çalıştırılması ve bu çalışmalardan elde edilen sonuçların ortalamasının alınması, simülasyon sonuçlarının doğruluğunu ve güvenilirliğini artırır. Her bir senaryoda, farklı düğüm sayıları ve ağ topolojileri kullanılarak, ağ performansı ve enerji tüketimi ölçümleri gerçekleştirilmiştir. Düğüm sayıları 10, 20, 30 ve 50 olarak belirlenmiş ve her bir durumda ağın enerji verimliliği, paket teslim oranı, gecikme süresi ve ağ ömrü gibi metrikler değerlendirilmiştir. Ayrıca, ağda meydana gelen olası hataların ve saldırıların etkileri de incelenmiş ve bu durumlarda ağın performansı analiz edilmiştir.

6.1. Önerilen Proxy-JRC Seçim Yöntemlerinin Deneysel Sonuçları

Contiki OS'nin Powertrace ve Energest güç profili modülleri, düğümlerin anahtar güncelleme prosedürü için güç tüketimini değerlendirmek amacıyla kullanılmıştır (Dunkels vd., 2007). Bu modüller, her bir durumda (iletim, alım, uyku) harcanan süreyi kaydeder. Tablo 16, Cooja² cihazları için her bir durumdaki akım tüketim seviyelerini göstermektedir. Bu değerler, düğümlerin enerji tüketim seviyelerini ölçmek ve ağın genel enerji verimliliğini analiz etmek için kullanılmıştır. Ayrıca, düğümlerin farklı durumlarda harcadığı enerji miktarları, ağın performansını optimize etmek amacıyla analiz edilmiştir. Önerilen Proxy-JRC seçim yöntemlerinin deneysel sonuçları, enerji tüketimi ve ağ performansı açısından detaylı bir şekilde analiz edilmiştir. Bu analizlerde, düğümlerin enerji tüketim seviyeleri, ağın genel enerji verimliliği ve paket teslim oranı gibi metrikler değerlendirilmiştir.

¹ Seed, rastgele sayı üreteçlerinin başlangıç noktası olarak kullanılan bir değerdir. Simülasyonlarda farklı seed değerleri kullanılarak, her çalıştırmada farklı rastgelelik koşulları elde edilir. Bu, sonuçların güvenilirliğini ve doğruluğunu artırır.

²Exp5438 güç tüketimi değerleri referans alınmıştır

Tablo 16. Exp5438 cihazı için akım tüketim değerleri

Parametre	Değer
CPU	3V'da 1.9 mA
LPM	3V'da 0.0545 mA
Tx	3V'da 20 mA
Rx	3V'da 17.7 mA
Vcc	3V
Çalışma Süresi (saat frekansı)	32768 ticks

Tablo 17. Cooja simülatörü ve EXP5438 donanım cihazının karşılaştırmalı özellikleri

Özellik	Cooja Mote	EXP5438 Cihazı
Enerji Tüketimi Ölçümü	Tahmini Değerler	Gerçek Zamanlı Ölçümler
Platform	Yazılımsal Platform	MSP430 Tabanlı Donanım
Hafıza Kapasitesi	Sanal Hafıza	Fiziksel Hafıza
Batarya Modeli	Teorik Batarya Modeli	Gerçek Batarya Tüketimi
Komşu Tablosu	100	7
Yönlendirme Tablosu	100	15

Tablo 17, Cooja simülatöründe kullanılan yazılım tabanlı Cooja Mote ile donanım tabanlı EXP5438 cihazının özelliklerini karşılaştırmaktadır. Cooja Mote, enerji tüketimi ve batarya modeli gibi parametrelerde teorik ve tahmini değerlere dayanırken, EXP5438 gerçek zamanlı ölçümler ve fiziksel batarya tüketimi ile daha kesin sonuçlar sunmaktadır. Ayrıca, EXP5438'in donanımsal sınırları nedeniyle komşu ve yönlendirme tablolarında daha sınırlı bir kapasiteye sahip olduğu görülmektedir. Bununla birlikte, MSP430 tabanlı donanım platformu, enerji tüketimi açısından daha güvenilir ve gerçekçi sonuçlar sağlar.

Toplam güç tüketimi hesabı, düğümde belirli bir zamanda mevcut enerjiye dayanarak hesaplanır. Bu hesaplama, her düğüm için elde edilen değerlerin toplanması ve ortalamasının alınması ile gerçekleştirilir. Düğümün belirli bir zamanda toplam enerji tüketimini hesaplamak için kullanılan formül aşağıda verilmiştir:

$$E_{toplamlam} = V_{cc} \times \sum_{i=1}^N (I_{CPU,i} \times T_{CPU,i} + I_{LPM,i} \times T_{LPM,i} + I_{Tx,i} \times T_{Tx,i} + I_{Rx,i} \times T_{Rx,i}) / ticks \quad (6.1)$$

Genel güç tüketimi, bir düğümün belirli bir zaman diliminde mevcut enerji seviyesine dayalı olarak hesaplanır. Bu hesaplama, her düğüm için ayrı ayrı gerçekleştirilir ve sonuçlar Formül 6.1’de gösterildiği gibi elde edilir. Burada, i indeksi her bir düğümü temsil eder ve N toplam düğüm sayısını ifade eder. Öncelikle, her düğümde belirli bir zaman diliminde harcanan enerji miktarı belirlenir. $I_{CPU,i}$, $I_{LPM,i}$, $I_{Tx,i}$ ve $I_{Rx,i}$ sırasıyla her bir düğümün CPU, düşük güç modu, iletim ve alım akımlarını temsil ederken; $T_{CPU,i}$, $T_{LPM,i}$, $T_{Tx,i}$ ve $T_{Rx,i}$ bu modlarda harcanan süreleri temsil eder. Ayrıca, besleme gerilimi *voltage* olarak tanımlanmıştır ve değeri 3V’dir.

$$P_{cpu(mW)} = \frac{1.9 * voltage * T_{cpu}}{T_{cpu} + T_{lpm}} \quad (6.2)$$

$$P_{lpm(mW)} = \frac{0.0545 * voltage * T_{lpm}}{T_{cpu} + T_{lpm}} \quad (6.3)$$

$$P_{transmit(mW)} = \frac{17.7 * voltage * T_{transmit}}{T_{cpu} + T_{lpm}} \quad (6.4)$$

$$P_{listen(mW)} = \frac{20.0 * voltage * T_{listen}}{T_{cpu} + T_{lpm}} \quad (6.5)$$

Aşağıdaki örnek, Contiki kullanarak güç ve enerji tüketiminin nasıl hesaplandığını göstermektedir. Örnekte, Powertrace modülü kullanılarak düğümün farklı modlarda harcadığı enerji hesaplanmaktadır:

Time	CPU	LPM	TX	RX
0	512803	14227588	153188	195436
10	531519	14535272	158359	203847

20	549549	14844701	163409	211794
30	560341	15161365	165821	216534
40	575755	15473409	169844	223419
50	599333	15777318	178173	231767
60	610307	16093852	180704	236651
70	625674	16406020	184722	243660
80	649197	16710040	193402	252940
90	660144	17026626	195811	258453
100	670942	17343372	198221	263249

Bu verilere dayanarak, enerji tüketimi hesaplamaları aşağıdaki şekilde hesaplanmaktadır:

- CPU için:

$$P_{CPU} = \frac{(531519 - 512803) \times 1.9 \times 3}{32768 \times 10} \quad (6.6)$$

- LPM için:

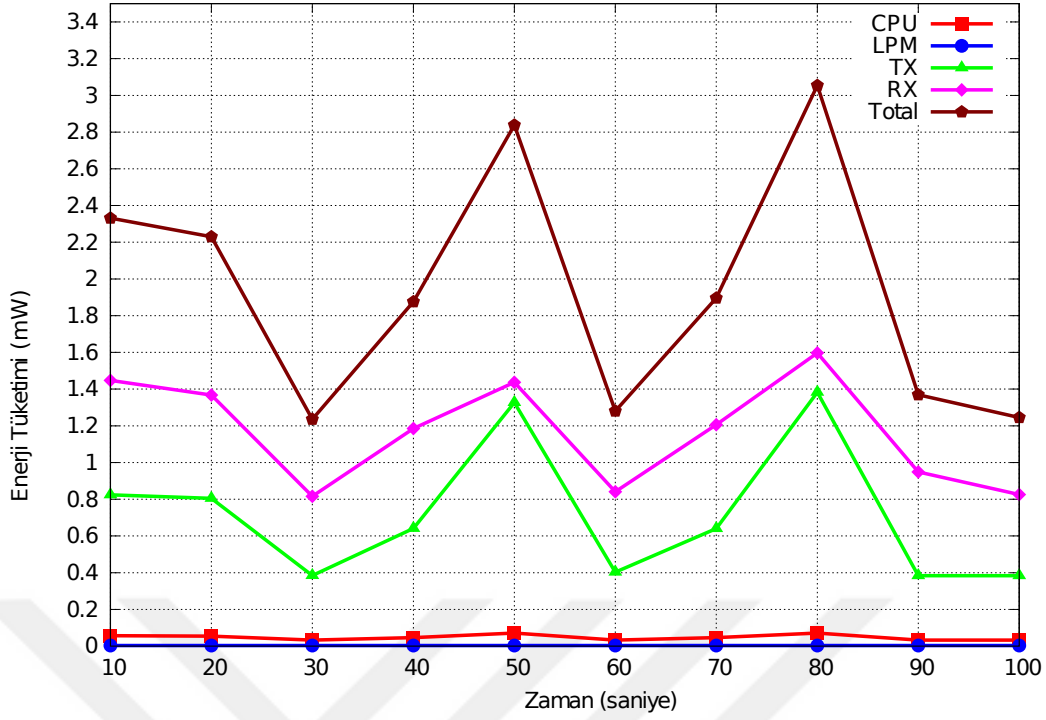
$$P_{LPM} = \frac{(14535272 - 14227588) \times 0.0545 \times 3}{32768 \times 10} \quad (6.7)$$

- TX için:

$$P_{TX} = \frac{(158359 - 153188) \times 20.0 \times 3}{32768 \times 10} \quad (6.8)$$

- RX için:

$$P_{RX} = \frac{(203847 - 195436) \times 17.7 \times 3}{32768 \times 10} \quad (6.9)$$



Şekil 69. CPU, LPM, TX, RX ve toplam enerji tüketiminin zamana bağlı analizi

Bu formüller, cihazların her bir durumda ne kadar güç tükettiğini hesaplamak için kullanılır. Enerji ve güç arasındaki ilişki ise şu şekildedir: Güç (P), bir cihazın birim zamanda harcadığı enerji miktarıdır ve genellikle watt (W) cinsinden ölçülür. Enerji (E) ise belirli bir zaman diliminde harcanan toplam güç miktarıdır ve joule (J) cinsinden ölçülür. Bu bağlamda, güç tüketimi, enerji tüketimini anlamamıza yardımcı olur ve cihazların performansını optimize etmemizde kritik bir rol oynar. Burada 10 ile çarpmanın nedeni, hesaplamaların her 10 saniyede bir yapılan ölçümler üzerinden yapılmasıdır. Her 10 saniyede bir ölçüm alındığı için enerji tüketimini doğru bir şekilde yansıtmak adına 10 saniyelik periyodun hesaba katılması gerekmektedir. Bu çarpma, güç tüketiminin toplam zaman dilimiyle orantılı olarak hesaplanmasını ve sonuçların doğru bir şekilde elde edilmesini sağlar.

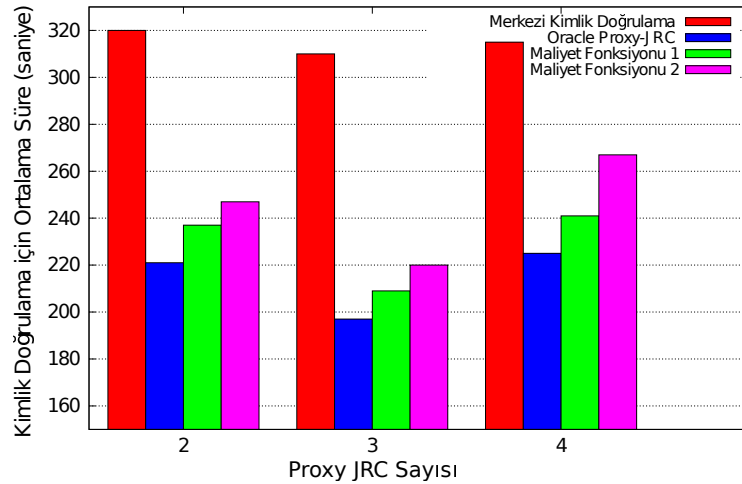
Şekil 69'da 20 düğümden oluşan bir ağda örnek güç hesabı gösterilmektedir. Cihazların enerji tüketimi hesaplanırken, görev döngüsü (duty cycle) değerleri de önem arz eder. Görev döngüsü, cihazın belirli bir zaman diliminde aktif olduğu sürenin toplam süreye oranıdır. Bu oran, cihazın enerji tüketimini ve performansını anlamak için kritik bir ölçüttür. Özellikle kaynak kısıtlı cihazlar için düşük görev döngüsü, enerji verimliliğini artırır ve cihazın pil ömrünü uzatır. Tablo 18, farklı zaman dilimlerinde CPU, LPM, TX ve RX görev döngüsü değerlerini sunmaktadır.

Tablo 18'de görüldüğü üzere, görev döngüsü değerleri farklı zaman dilimlerinde değişiklik göstermektedir. Bu değişiklikler, cihazın enerji tüketimi ve performansı üzerinde

Tablo 18. Görev döngüsü (Duty Cycle) değerleri

Zaman (saniye)	CPU Görev Döngüsü (%)	LPM Görev Döngüsü (%)	TX Görev Döngüsü (%)	RX Görev Döngüsü (%)	Toplam Görev Döngüsü (%)
10	1.58	2.58	1.58	2.58	4.16
20	1.54	2.43	1.54	2.43	3.97
30	0.74	1.45	0.74	1.45	2.18
40	1.23	2.10	1.23	2.10	3.33
50	2.54	2.55	2.54	2.55	5.09
60	0.77	1.49	0.77	1.49	2.26
70	1.23	2.14	1.23	2.14	3.37
80	2.65	2.83	2.65	2.83	5.48
90	0.74	1.68	0.74	1.68	2.42
100	0.74	1.46	0.74	1.46	2.20

doğrudan etkiye sahiptir. Örneğin, 10. saniyede TX görev döngüsü 1.58%, RX görev döngüsü 2.58% ve toplam görev döngüsü 4.16% iken, 50. saniyede TX görev döngüsü 2.54%, RX görev döngüsü 2.55% ve toplam görev döngüsü 5.09% olarak ölçülmüştür. Bu tür veriler, enerji verimliliği stratejilerinin geliştirilmesinde önemli bir rol oynar.



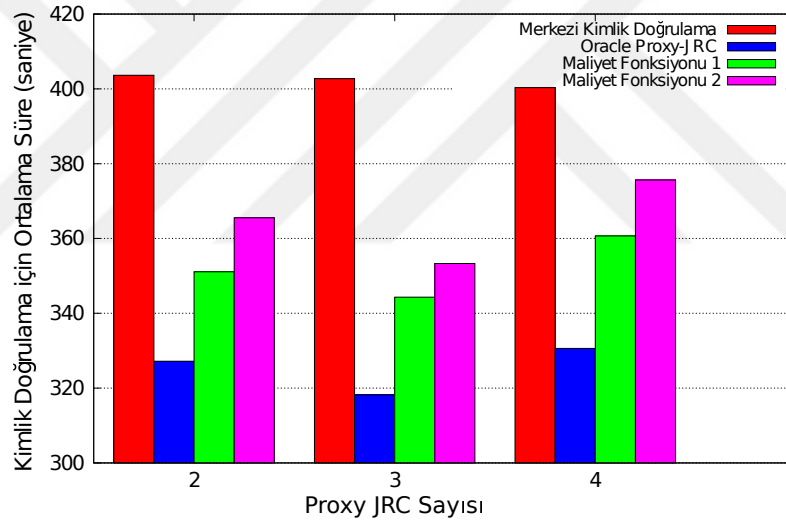
Şekil 70. 20 düğümlü bir ağ için ortalama anahtar güncelleme gecikmesi

Şekil 70, 71 ve 72, bağlantı başarı oranının 100% olduğu durumda, 20, 35 ve 50 düğümden oluşan bir ağ için ortalama anahtar güncelleme süresini göstermektedir. Uçtan uca gecikme, anahtar güncelleme talebinde bulunan düğümün ağ kimlik doğrulama paketindeki

zaman damgası ile JRC/Proxy-JRC tarafından bu düğümün talep mesajına yanıt olarak gönderilen paket arasındaki zaman farkından elde edilir.

Şekil 70 incelendiğinde, 20 düğümlü ağda daha düşük ağırlıklı maliyet fonksiyonlarının daha hızlı anahtar güncelleme süreleri sağladığı gözlemlenmektedir. Bu durum, daha düşük ağırlıkların düğümler arasındaki iletişim yoğunluğunu azaltarak ağdaki tıkanıklığı engellemesi ve böylece anahtar güncelleme süreçlerinin hızlanması ile açıklanabilir. Benzer şekilde, Şekil 71 ve Şekil 72 incelendiğinde, 35 ve 50 düğümlü ağlarda da maliyet fonksiyonlarının ağırlıklarının artırılmasıyla ortalama anahtar güncelleme sürelerinin azaldığı görülmektedir.

Bu gözlemler, önerilen mekanizmanın özellikle daha büyük ölçekli ağlarda verimliliği artırmak için uygun olduğunu göstermektedir. Düşük gecikme süreleri ve hızlı anahtar güncellemeleri, ağ güvenliğinin yanı sıra performansını da artırmaktadır. Dolayısıyla, maliyet fonksiyonlarının uygun şekilde optimize edilmesi, ağdaki anahtar güncellemelerinin etkinliğini önemli ölçüde iyileştirebilir.

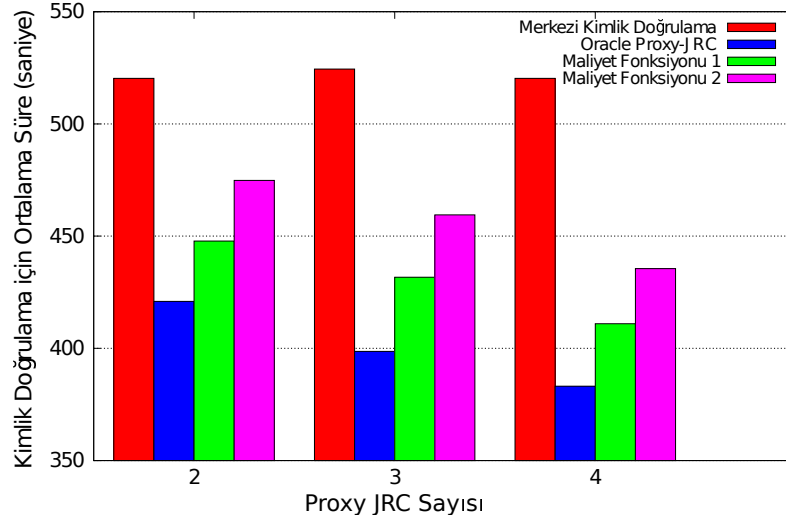


Şekil 71. 35 düğümlü bir ağ için ortalama anahtar güncelleme gecikmesi

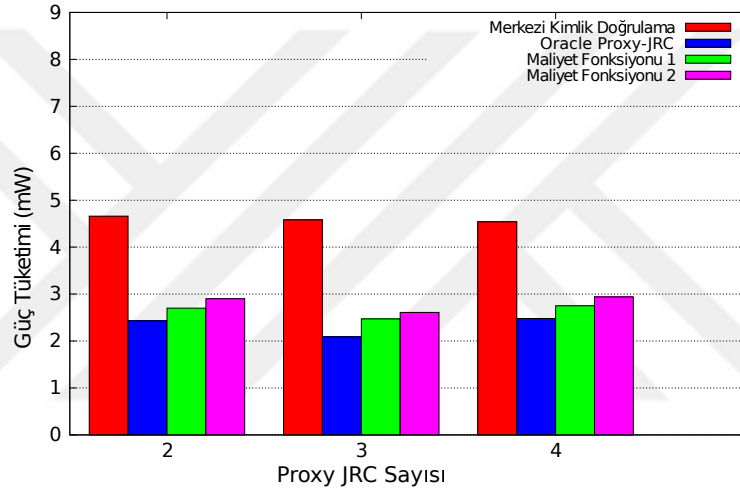
Şekiller 73, 74, ve 75 sırasıyla 20, 35 ve 50 düğümlü ağlar için anahtar güncelleme işlemi sırasında ortalama güç tüketimini göstermektedir. Bağlantı başarı oranının 100% olduğu durumlar için yapılan bu ölçümler, 2 proxy JRC seçimi yöntemlerinin güç tüketimi üzerindeki etkilerini değerlendirmek amacıyla gerçekleştirilmiştir.

Şekillerde görüldüğü üzere, merkezi kimlik doğrulama yöntemleri en yüksek güç tüketimine sahipken, Oracle Proxy-JRC ve maliyet fonksiyonlarına dayalı dağıtık kimlik doğrulama yöntemleri daha düşük güç tüketimi göstermektedir. Bu, dağıtık kimlik doğrulama yöntemlerinin enerji verimliliği açısından daha avantajlı olduğunu ortaya koymaktadır.

Şekil 76, 20 düğümlü ağda ve 2 Proxy-JRC'li farklı bağlantı kalitesi seviyeleri için ortalama anahtar güncelleme gecikmesini göstermektedir. Sonuçlar, bağlantı kalitesi



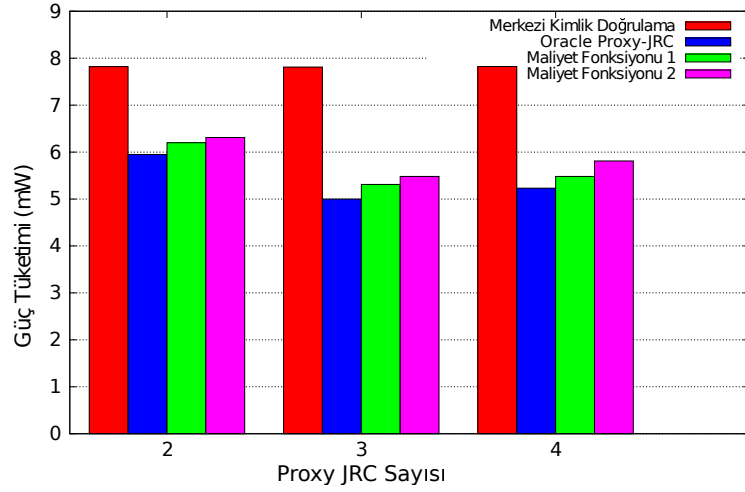
Şekil 72. 50 düğümlü bir ağ için Ortalama anahtar güncelleme gecikmesi



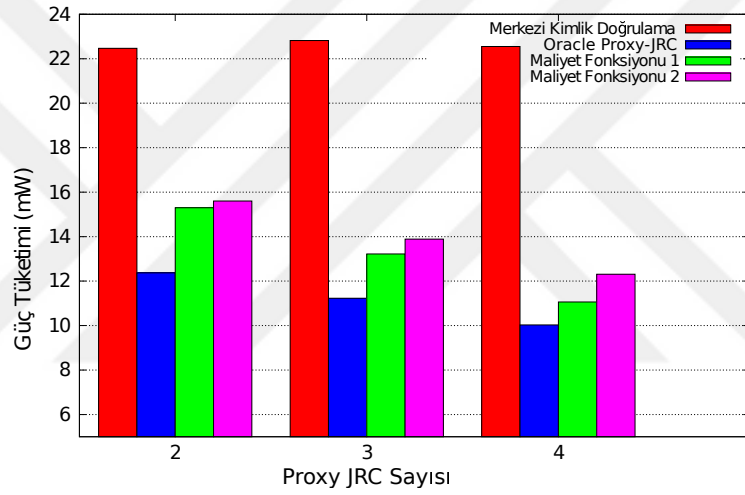
Şekil 73. 20 düğümlü bir ağda anahtar güncellemesi için ortalama güç tüketimi

iyileştikçe tüm kimlik doğrulama yöntemleri için anahtar güncelleme gecikmesinin azaldığını göstermektedir. Merkezi kimlik doğrulama yöntemi, anahtar güncellemelerini yönetme konusundaki verimsizliğini yansıtarak, sürekli olarak en yüksek gecikmeyi göstermektedir. Oracle Proxy-JRC yöntemi, en düşük gecikmeyi sergileyerek, anahtar güncellemelerini verimli bir şekilde yönetme konusundaki üstün performansını göstermektedir. Hem Maliyet Fonksiyonu 1 hem de Maliyet Fonksiyonu 2, merkezi yöntemden daha iyi performans gösterirken, Oracle Proxy-JRC'den biraz daha az verimlidir. Performans farkı, daha düşük Rx değerlerinde (70%-80%) daha belirgindir; burada Oracle Proxy-JRC diğer yöntemlere göre önemli ölçüde daha iyi performans göstermektedir. Daha yüksek Rx değerlerinde (90%-100%), tüm yöntemlerin gecikmeleri yakınsamakta, ancak merkezi kimlik doğrulama yine de dağıtık yöntemlerin gerisinde kalmaktadır.

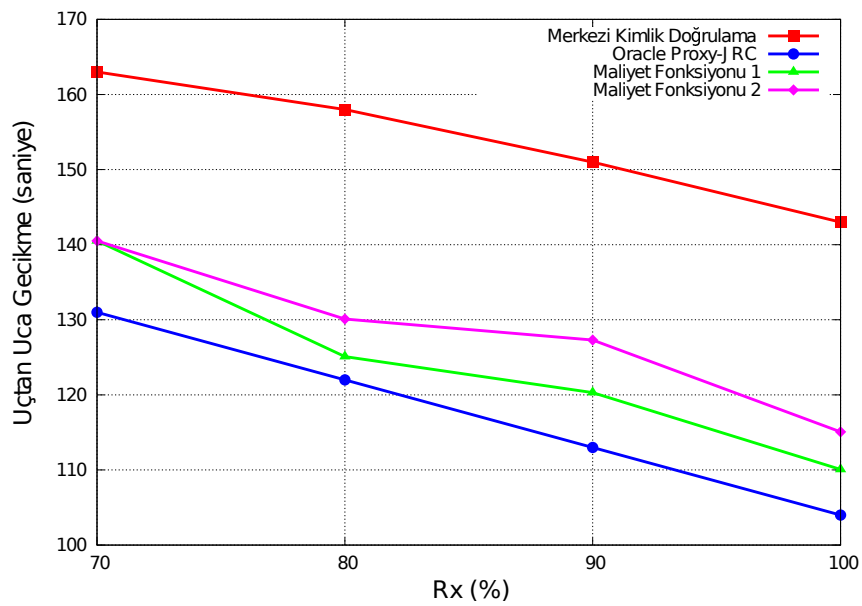
Şekil 77, farklı bağlantı kalitesi seviyeleri için ortalama cihaz anahtarı güncelleme gecikmesini sunmaktadır. Düğüm, bu süre zarfında birden fazla anahtar güncelleme isteği



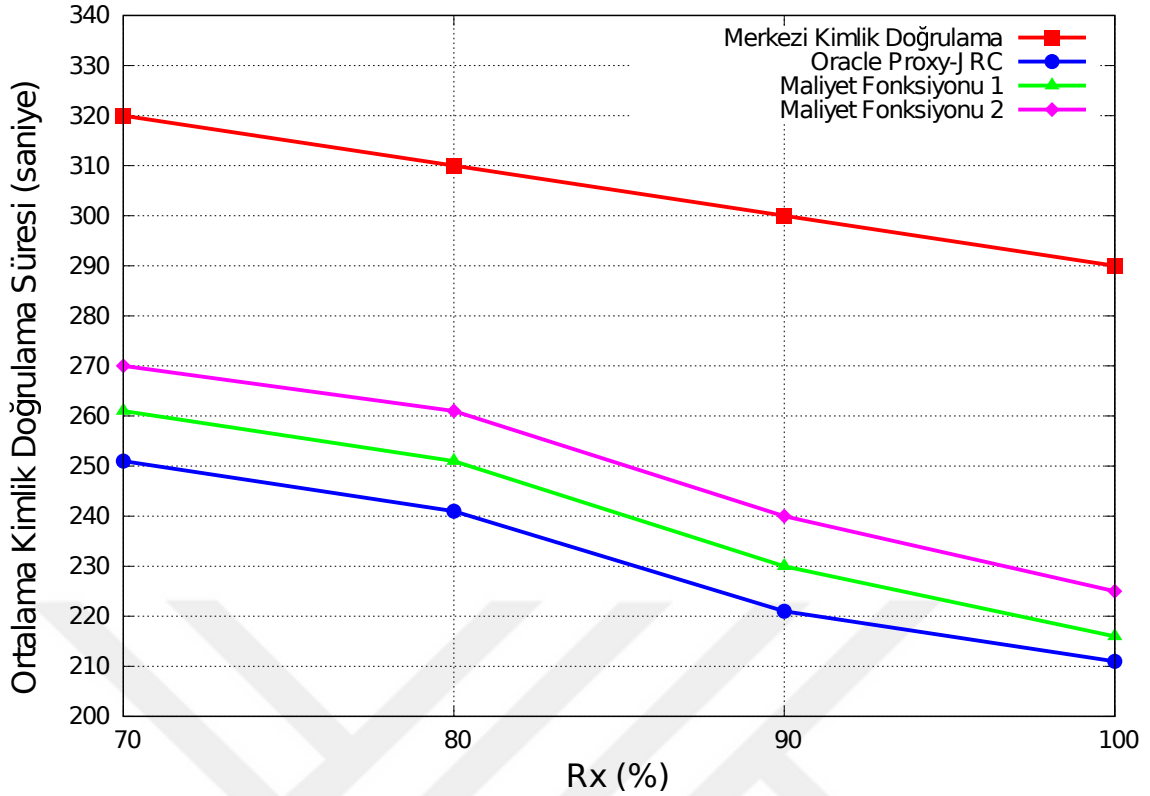
Şekil 74. 35 düğümlü bir ağda anahtar güncellemesi için ortalama güç tüketimi



Şekil 75. 50 düğümlü bir ağda anahtar güncellemesi için ortalama güç tüketimi



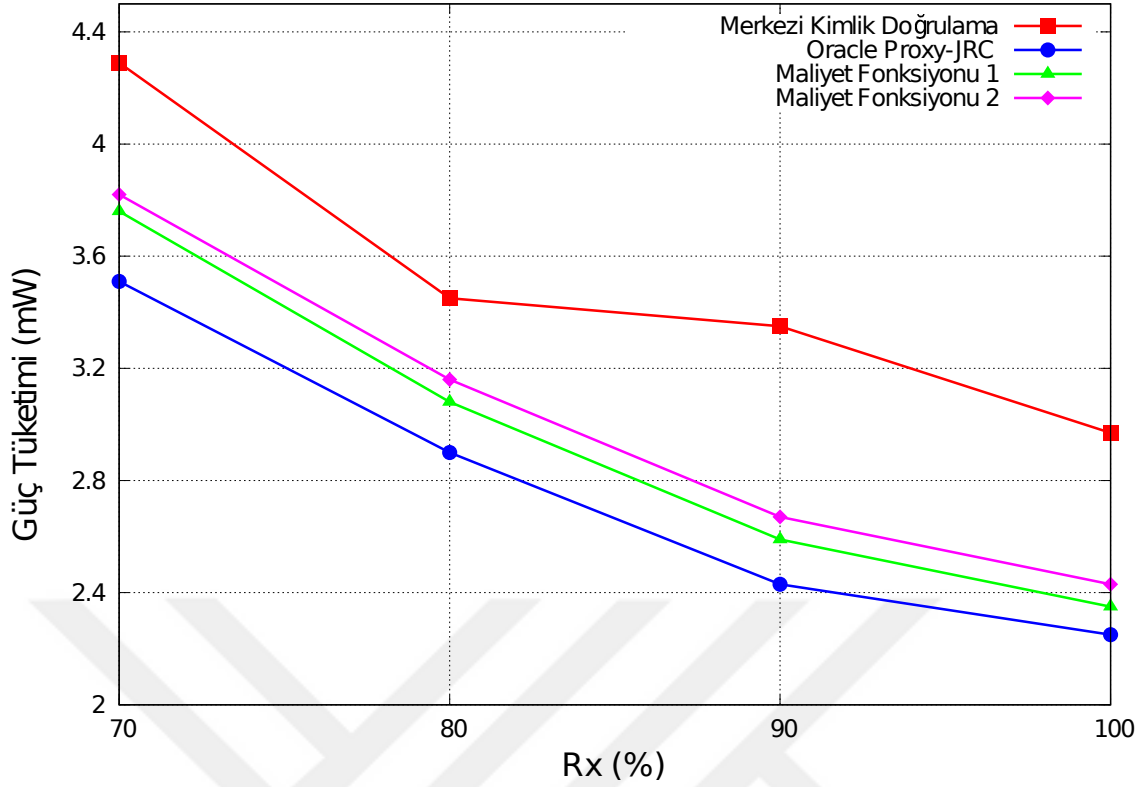
Şekil 76. 20 düğümlü ağda 2 Proxy-JRC için ortalama anahtar güncelleme gecikmesi



Şekil 77. 20 düğümlü ağda ortalama cihaz anahtarı güncelleme gecikmesi

gönderebilir. Ancak, istek paketleri, kanal çakışması nedeniyle ara düğümlerde düşebilir, bu da doğrulama gecikmesini artırır. Sonuçlar, daha yüksek bağlantı kalitesi ile tüm yöntemler için güncelleme gecikmelerinin azaldığını net bir şekilde göstermektedir. Merkezi kimlik doğrulama, sürekli olarak en yüksek gecikmeyi sergileyerek, ölçeklenebilirlik sorunlarını ve verimsizliğini ortaya koymaktadır. Buna karşılık, Oracle Proxy-JRC yöntemi, en az gecikme ile en iyi performansı göstermekte, anahtar güncellemelerini yönetmede etkinliğini kanıtlamaktadır. Daha düşük bağlantı kalitesi seviyelerinde (70%-80%), merkezi kimlik doğrulama ile diğer yöntemler arasındaki gecikme farkı önemlidir.

Şekil 78, 20 düğümlü ağda ve 2 Proxy-JRC'li farklı bağlantı kalitesi seviyeleri boyunca anahtar güncelleme işlemi sırasında ortalama güç tüketimini göstermektedir. Merkezi kimlik doğrulama yöntemi, en yüksek güç tüketimini göstererek, verimsizliğini ve daha yüksek enerji gereksinimlerini vurgulamaktadır. Diğer yandan, Oracle Proxy-JRC yöntemi, en düşük güç tüketimini sergileyerek, RPL tabanlı 6TiSCH ağlarında anahtar güncellemeleri için en enerji verimli seçeneği ortaya koymaktadır. Önerilen yöntemler olan Maliyet Fonksiyonu 1 ve Maliyet Fonksiyonu 2 de merkezi kimlik doğrulamadan daha iyi enerji verimliliği sergilemekte olup, Maliyet Fonksiyonu 1, Maliyet Fonksiyonu 2'den daha enerji verimlidir. Bağlantı kalitesi arttıkça, tüm yöntemler için güç tüketimi azalmaktadır. 100% Rx seviyesinde, Oracle Proxy-JRC ve Maliyet Fonksiyonu 1'in güç tüketimi neredeyse aynı seviyeye gelmekte



Şekil 78. 20 düğümlü ağda anahtar güncelleme işlemi sırasında tüketilen ortalama güç değerleri (2 Proxy-JRC)

ve ideal koşullarda en iyi performansı göstermektedirler. Örneğin, 70% Rx bağlantı kalitesinde merkezi kimlik doğrulama yöntemi yaklaşık 4.29 mW güç tüketirken, Oracle Proxy-JRC yöntemi sadece 3.51 mW güç tüketmektedir. Maliyet Fonksiyonu 1 ise 3.76 mW, Maliyet Fonksiyonu 2 ise 3.82 mW güç tüketimi ile merkezi yönteme kıyasla daha iyi bir performans sergilemektedir. 80% Rx seviyesinde merkezi kimlik doğrulama yöntemi 3.45 mW güç tüketirken, Oracle Proxy-JRC yöntemi 2.90 mW güç tüketmekte ve Maliyet Fonksiyonu 1 ve Maliyet Fonksiyonu 2 sırasıyla 3.08 mW ve 3.16 mW güç tüketimi ile merkezi yönteme göre daha verimli olmaktadır. 90% Rx seviyesinde merkezi kimlik doğrulama yöntemi 3.35 mW güç tüketirken, Oracle Proxy-JRC yöntemi 2.43 mW güç tüketmekte ve Maliyet Fonksiyonu 1 ve Maliyet Fonksiyonu 2 sırasıyla 2.59 mW ve 2.67 mW güç tüketimi ile merkezi yönteme kıyasla daha düşük güç tüketmektedir. 100% Rx seviyesinde, merkezi kimlik doğrulama yöntemi 2.97 mW güç tüketirken, Oracle Proxy-JRC yöntemi 2.25 mW güç tüketmekte ve Maliyet Fonksiyonu 1 ve Maliyet Fonksiyonu 2 sırasıyla 2.35 mW ve 2.43 mW güç tüketimi ile daha verimli bir performans sergilemektedir.

Tablo 19, 20 düğümlü bir ağ senaryosunda 2, 3 ve 4 Proxy-JRC seçildiğinde, anahtar güncelleme adımı için ağdaki tüm düğümler tarafından harcanan ortalama süreyi sunmaktadır. Ağdaki Proxy-JRCs sayısı, belirli bir eşişge kadar anahtar güncelleme süreleri üzerinde olumlu

Tablo 19. Proxy-JRC sayısına göre 20 düğüm için ortalama anahtar güncelleme süresi (saniye)

Rx	Proxy Sayısı	Merkezi Kimlik Doğrulama	Oracle Proxy-JRC	Maliyet Fonksiyonu 1	Maliyet Fonksiyonu 2
%70	2	324	251	261	270
	3		238	247	256
	4		265	278	295
%80	2	314	241	251	261
	3		227	235	243
	4		251	267	288
%90	2	303	221	230	240
	3		209	218	228
	4		240	254	283
%100	2	296	211	216	225
	3		197	206	216
	4		231	246	279

bir etkiye sahiptir. Ancak çok sayıda Proxy-JRC kullanılması, iletişim ve senkronizasyon maliyetleri anahtar güncelleme sürelerini artırabileceğinden anahtar güncelleme sürelerini olumlu yönde etkilemeyebilir. Bu senaryoda, Proxy-JRC sayısı ile ağ boyutu arasında bir maliyet dengesi olduğu görülebilir.

Tablo 20, farklı bağlantı kalitesi seviyelerinde (70%, 80%, 90% ve 100%) 20 düğümlü ağda ve 2, 3 ve 4 Proxy-JRC ile yapılan simülasyonlar sırasında ortalama güç tüketimini göstermektedir. Örneğin, 70% Rx bağlantı kalitesinde merkezi kimlik doğrulama yöntemi yaklaşık 4.29 mW güç tüketirken, Oracle Proxy-JRC yöntemi sadece 3.51 mW güç tüketmektedir. Bu, Oracle Proxy-JRC'nin merkezi kimlik doğrulama yöntemine göre 18% daha verimli olduğunu göstermektedir. Maliyet Fonksiyonu 1 ise 3.76 mW, Maliyet Fonksiyonu 2 ise 3.82 mW güç tüketimi ile merkezi yönteme kıyasla daha iyi bir performans sergilemektedir. 80% Rx seviyesinde merkezi kimlik doğrulama yöntemi 3.45 mW güç tüketirken, Oracle Proxy-JRC yöntemi 2.90 mW güç tüketmektedir.

Maliyet Fonksiyonu 1 ve Maliyet Fonksiyonu 2 sırasıyla 3.08 mW ve 3.16 mW güç tüketimi ile merkezi yönteme göre daha verimlidir. 90% Rx seviyesinde merkezi kimlik doğrulama yöntemi 3.25 mW güç tüketirken, Oracle Proxy-JRC yöntemi 2.43 mW güç

Tablo 20. Anahtar güncelleme sırasında ortalama güç tüketimi (mW) (20 düğüm)

Rx	Proxy Sayısı	Merkezi Kimlik Doğrulama	Oracle Proxy-JRC	Maliyet Fonksiyonu 1	Maliyet Fonksiyonu 2
%70	2	4.29	3.51	3.76	3.82
	3		3.52	3.74	3.79
	4		3.59	3.96	4.01
%80	2	3.45	2.90	3.08	3.16
	3		2.85	3.00	3.08
	4		2.97	3.24	3.24
%90	2	3.25	2.43	2.59	2.67
	3		2.40	2.48	2.57
	4		2.49	2.76	2.83
%100	2	2.97	2.25	2.35	2.43
	3		2.23	2.28	2.39
	4		2.33	2.58	2.65

tüketmektedir. Maliyet Fonksiyonu 1 ve Maliyet Fonksiyonu 2 sırasıyla 2.59 mW ve 2.67 mW güç tüketimi ile merkezi yöntemle kıyasla daha düşük güç tüketmektedir. 100% Rx seviyesinde, merkezi kimlik doğrulama yöntemi 2.97 mW güç tüketirken, Oracle Proxy-JRC yöntemi 2.25 mW güç tüketmektedir. Maliyet Fonksiyonu 1 ve Maliyet Fonksiyonu 2 sırasıyla 2.35 mW ve 2.43 mW güç tüketimi ile daha verimli bir performans sergilemektedir. Bu bulgular, Oracle Proxy-JRC yönteminin enerji verimliliği açısından en iyi performansı sergilediğini ve önerilen Maliyet Fonksiyonu 1 ve Maliyet Fonksiyonu 2'nin de merkezi kimlik doğrulama yöntemine kıyasla önemli ölçüde daha verimli olduğunu göstermektedir.

Alım oranı (Rx), ağdaki düğümlerin veri iletiminde ne kadar başarılı olduğunu ifade eder. Yüksek Rx değerleri, daha az veri kaybı ve daha verimli bir ağ anlamına gelir. Düşük Rx değerleri ise, veri iletiminde daha fazla başarısızlık ve dolayısıyla daha fazla yeniden iletim ve enerji tüketimi anlamına gelir. Bu nedenle, farklı Rx değerlerinin ağın güç tüketimi ve performansı üzerinde önemli bir etkisi vardır. Yüksek Rx değerlerinde, ağın genel verimliliği artarken, düşük Rx değerlerinde güç tüketimi ve gecikme süreleri artar. Bu sonuçlar, ağın boyutu ve Proxy-JRC sayısı gibi faktörlerle birlikte değerlendirildiğinde, optimum performans için uygun dengeyi bulmayı sağlar.

Tablo 21. Anahtar güncelleme sırasında ortalama güç tüketimi (mW) (35 düğüm)

Rx	Proxy Sayısı	Merkezi Kimlik Doğrulama	Oracle Proxy-JRC	Maliyet Fonksiyonu 1	Maliyet Fonksiyonu 2
%70	2	7.51	6.14	6.58	6.69
	3		6.16	6.55	6.63
	4		6.28	6.93	7.02
%80	2	6.54	5.08	5.39	5.53
	3		4.99	5.25	5.39
	4		5.20	5.67	5.67
%90	2	5.6	4.25	4.53	4.67
	3		4.20	4.34	4.50
	4		4.36	4.83	4.95
%100	2	4.80	3.94	4.11	4.25
	3		3.90	3.99	4.18
	4		4.08	4.52	4.64

Tablo 21, 35 düğümlü bir ağda farklı Proxy-JRC sayıları ve Rx bağlantı kalitesi için merkezi ve dağıtık anahtar güncelleme işlemleri sırasında ortalama güç tüketimini göstermektedir. Tablo 21’den de görülebileceği gibi, Oracle Proxy-JRC, merkezi kimlik doğrulama yöntemine göre her durumda daha az güç tüketmektedir. Rx 70% ve 2 Proxy-JRC kullanıldığında, dağıtık anahtar güncelleme (Oracle Proxy-JRC) 6.14 mW güç tüketirken, merkezi kimlik doğrulama 7.51 mW tüketmektedir. Bu, dağıtık yöntemin merkezi anahtar güncelleme mekanizmasından daha enerji verimli olduğunu göstermektedir. Proxy sayısı 4’e çıkarıldığında bile, dağıtık yöntemin güç tüketimi hala merkezi yöntemin altında kalmakta ve 6.28 mW olmaktadır. Rx 80% ve 2 Proxy-JRC kullanıldığında, dağıtık anahtar güncelleme yöntemi, merkezi kimlik doğrulama yöntemine (6.54 mW) göre daha avantajlıdır ve 5.08 mW güç tüketmektedir. Proxy sayısı arttıkça (3 ve 4 Proxy-JRC), güç tüketimi artsa da merkezi yönteme kıyasla hala daha düşük kalmaktadır. Rx 90% ve 2 Proxy-JRC kullanıldığında, dağıtık yöntemin güç tüketimi 4.25 mW iken, merkezi kimlik doğrulama yöntemi 5.6 mW tüketmektedir. Proxy sayısı arttığında (3 ve 4 Proxy-JRC) bile bu durum değişmemekte ve dağıtık yöntem hala daha az güç tüketmektedir. Rx 100% ve 2 Proxy-JRC kullanıldığında, dağıtık anahtar güncelleme yöntemi merkezi kimlik doğrulama yönteminden 18% daha

verimli olup 3.94 mW güç tüketmektedir. Proxy sayısı arttıkça güç tüketimi artmasına rağmen (4 Proxy için 4.08 mW), bu değer hala merkezi yöntemle göre 15% daha verimlidir.

Maliyet Fonksiyonları 1 ve 2 incelendiğinde, dağıtık anahtar güncelleme yönteminin enerji verimliliği daha belirgin hale gelmektedir. Rx 70% ve 2 proxy kullanıldığında, Maliyet Fonksiyonu 1 için güç tüketimi 6.58 mW ve Maliyet Fonksiyonu 2 için 6.69 mW'dır. Bu değerler, merkezi kimlik doğrulama yöntemine göre daha düşük güç tüketimini göstermektedir. Rx değeri arttıkça (örneğin 80%, 90%, 100%), dağıtık yöntemin enerji verimliliği korunmakta ve merkezi kimlik doğrulamaya göre daha düşük güç tüketimi sağlanmaktadır. Bu durum, dağıtık yöntemin yüksek Rx değerlerinde bile enerji verimliliğini sürdürdüğünü göstermektedir. Ayrıca, Proxy-JRC sayısının artmasıyla birlikte güç tüketiminde bir artış gözlenmektedir (2, 3, 4 proxy). Ancak, bu artış merkezi kimlik doğrulama yöntemine kıyasla yine de daha düşük kalmaktadır. Örneğin, 80% Rx oranı ve 4 proxy ile, dağıtık anahtar güncelleme yöntemi Maliyet Fonksiyonu 1 için 5.67 mW ve Maliyet Fonksiyonu 2 için 5.67 mW güç tüketmektedir, bu da merkezi kimlik doğrulama yönteminin 6.54 mW güç tüketiminden daha düşüktür. Bu veriler, dağıtık anahtar güncelleme yönteminin hem ölçeklenebilirliğini hem de enerji verimliliğini vurgulamaktadır. Sonuç olarak, dağıtık anahtar güncelleme yöntemi, ağda daha enerji verimli bir çözüm sunmakta ve hem Rx değeri hem de proxy sayısına bağlı olarak ölçeklenebilirliğini korumaktadır. Bu bulgular, büyük ölçekli ve dinamik ağ ortamlarında dağıtık yöntemlerin üstünlüğünü göstermektedir.

6.2. Önerilenn Hafif Üç Faktörlü Sensör ve Kullanıcı Kimlik Doğrulama Mekanizmasının Deneysel Sonuçları

6.2.1. Güvenlik Analizi

Bu bölümde, önerilen mekanizmanın potansiyel saldırı modelleri ve güvenlik gereksinimlerine karşı dayanıklılığını değerlendirmek için kapsamlı bir güvenlik analizi yapılmıştır. Analiz, önerilen modelin yüksek bir güvenlik seviyesi sergilediğini ve geniş bir saldırı yelpazesini etkili bir şekilde hafiflettiğini göstermektedir. Güvenlik analizinde BAN mantığı kullanılmaktadır. BAN mantığı, protokollerin güvenlik özelliklerini formal olarak analiz etmek için kullanılan bir yöntemdir. Bu mantık, protokol katılımcıları arasındaki güven ilişkilerini ve bu ilişkilerin zaman içindeki evrimini değerlendirir. BAN mantığı, katılımcıların belirli mesajları aldıklarında ne tür güven çıkarımlarında bulunabileceklerini

ifade eder. Önerilen kimlik doğrulama mekanizması, sensör ve sunucu arasındaki güvenli iletişimi sağlamak için OSCORE kullanır. Kullanıcı ve sunucu arasındaki iletişim ise IPSEC protokolü ile korunmaktadır. OSCORE, CoAP üzerinden uçtan uca güvenlik sağlarken, IPSEC internet protokolü üzerinden veri akışının güvenliğini sağlar.

- **Kullanıcı Anonimliği:** KSA'larda kullanıcı anonimliği, ağ güvenliği ve gizliliği açısından büyük önem taşır. Bu özellik veri toplayan veya bu ağlarla etkileşimde bulunan kullanıcıların kimliklerinin gizlenmesini sağlar. Böylece, kullanıcıların kişisel bilgilerini ve konumlarını koruyarak gizliliklerini muhafaza etmelerine yardımcı olur. Kullanıcı anonimliği, aynı zamanda verilerin güvenli bir şekilde iletilmesini ve kullanıcıların izlenmesini veya hedef alınmasını engeller. Anonimleştirme protokolleri, sensörlerin ve kullanıcıların kimliklerinin gizlenmesine yardımcı olurken, verilerin bütünlüğünü ve doğruluğunu korur.

Sunucu, kullanıcıdan $(DID_k, M_k, M_{k-s}, T_1)$ talep mesajını aldığı anda, $M_{k-s} = H(DID_k || R_k || T_1)$ denklemini değerlendirir. Sunucu paketi onaylarsa, bu paketin meşru bir kullanıcıdan geldiği anlamına gelir. Saldırganın, açık kanaldan alınan mesajdan meşru bir kullanıcının kimliğini (DID_k) elde etmeye çalıştığı varsayılın. Gerçek DID_k 'yi başarılı bir şekilde elde edebilmek için, saldırganın $ID_k \oplus H(PWD_k)$ hesaplamasını yapması gerekir. Ancak, $H(PWD_k)$ yalnızca meşru sunucu tarafından bilinir ve ayrıca hash fonksiyonu ile korunur, bu da belirli bir sürede hesaplanmasını imkansız hale getirir. Bu nedenle, önerilen şema kullanıcı anonimliği sağlar. Kullanıcı anonimliğini ve önerilen şemanın genel güvenliğini BAN mantığını kullanarak değerlendirmek için protokoldeki her katılımcının (sensör, kullanıcı ve sunucu) çeşitli varsayımları dikkate alınır.

Mesaj Yorumlama ve Varsayımlar:

- $K \rightarrow S : \{DID_k, M_k, M_{k-s}, T_1\}, S \rightarrow K : \{R_s, T_s\}$
- S, K 'nin K_{s-k} 'nin güncel olduğunu varsayar. K, S 'nin K_{s-k} 'nin güncel olduğunu ve S, K 'nin R_k 'yi ürettiğini kabul eder. K, S 'nin R_s 'yi ürettiğini varsayar.

Mantıksal Çıkarımlar:

- $\{DID_k, M_k, M_{k-s}, T_1\}$ 'yi aldığı anda, *Sunucu* aşağıdaki kontrolleri yapar:
 1. S, T_1 'in güncel olup olmadığını kontrol eder. $S, M_{k-s} = H(DID_k || R_k || T_1)$ 'i yeniden hesaplar.

2. Yeniden hesaplanan M_{k-s} , alınan M_{k-s} ile eşleşirse, sunucu, kullanıcının kimliğini doğrular ve DID_k 'nin geçerli olduğunu kabul eder.

– Kullanıcı, R_s, T_s mesajını aldığı anda, aşağıdaki kontrolleri gerçekleştirir:

1. Kullanıcı, T_s 'nin güncel olup olmadığını kontrol eder. Kullanıcı, R_s 'nin sunucu tarafından üretildiğini doğrular. R_s geçerliyse, kullanıcı, sunucunun kimliğini doğrular ve oturum anahtarının (K_k) geçerli olduğunu kabul eder.

Sonuçlar:

- S , M_{k-s} mesajını doğrularsa K 'nin kimliğinin geçerli olduğunu kabul eder. S , oturum anahtarı K_{s-k} 'nin güncel olduğunu kabul eder. S , yalnızca DID_k ve $H(PWD_k)$ 'yi doğrulamada kullandığı için K 'nin anonim kalmasını sağlar.
- K , R_s ve T_s parametrelerini doğrularsa S 'nin kimliğinin geçerli olduğunu kabul eder. K , anonim kalır çünkü DID_k , ID_k ve $H(PWD_k)$ 'den türetilmiş olup, bu bilgiler asla açık metin olarak iletilmez.

Özetle, karma şifreler ve oturum anahtarları kullanılarak, önerilen kimlik doğrulama mekanizması kimlik hırsızlığı ve yeniden oynatma saldırılarına karşı güçlü bir güvenlik sağlar ve iletişim sürecinin bütünlüğünü ve gizliliğini korur.

- Oturum Anahtarı Güvenliği: Saldırgan tüm iletişim mesajlarına (M_s , M_{s-k} , T_2 , $E("HELLO")$, M_{s-c} , $E(PK)$, T_3) erişse bile, oturum anahtarını (SK_{s-k}) oluşturamaz. SK_{s-k} 'yi oluşturmak için saldırganın kullanıcının R_k ve sunucunun r_s , R_s değerlerini bilmesi gereklidir. Saldırgan R_k ve r_k değerlerini elde etse bile, R_s ve r_s 'yi bilmeden oturum anahtarını (SK_{s-k}) hesaplayamaz. Bu nedenle, önerilen şema oturum anahtarı güvenliğini sağlar.

Mesaj Yorumlama ve Varsayımlar:

- $K \rightarrow S : \{DID_k, M_k, M_{k-s}, T_1\}$
- $S \rightarrow K : \{R_s, T_s\}$
- Sunucu, kullanıcının K_{s-k} 'nin güncel olduğunu bildiğini varsayar. Kullanıcı, sunucunun K_{s-k} 'nin güncel olduğunu kabul eder.
- Sensör, SK_{s-k} 'nin güncel ve gizli olduğunu varsayar.

Mantıksal Çıkarımlar:

- $\{DID_k, M_k, M_{k-s}, T_1\}$ 'yi aldığıında, *Sunucu* aşağıdaki kontrolleri yapar:
 1. S, T_1 zaman damgasının güncelliğini kontrol eder. Ardından, $M_{k-s} = H(DID_k || R_k || T_1)$ ifadesini yeniden hesaplar.
 2. Yeniden hesaplanan M_{k-s}^* , alınan M_{k-s} ile eşleşirse, S, K 'nin meşru olduğuna ve DID_k 'nin geçerli olduğuna inanır.
- $\{R_s, T_s\}$ 'yi aldığıında, K aşağıdaki kontrolleri yapar:
 1. K, T_s 'nin güncelliğini kontrol eder. K, R_s 'nin S tarafından oluşturulduğunu doğrular. R_s geçerliyse, K, S 'nin meşru olduğuna ve oturum anahtarı K_{s-k} 'nin güncel olduğuna inanır.
- $\{SK_{s-k}, T_s\}$ 'yi aldığıında, sensör aşağıdaki kontrolleri yapar:
 1. Sensör, T_s 'nin güncelliğini kontrol eder. SK_{s-u} 'nin S tarafından üretildiği doğrulanır. SK_{s-k} geçerliyse, sensör S 'nin meşru olduğuna ve oturum anahtarı K_{s-k} 'nin güncel ve gizli olduğuna inanır.

Sonuçlar:

- S, M_{k-s} mesajını uygun şekilde doğrularsa K 'nin meşru olduğunu varsayar.
- S , oturum anahtarı K_{s-k} 'nin güncel ve gizli olduğuna inanır.
- S, DID_k ve $H(PWD_k)$ 'yi doğrulamada kullandığı için U 'nun anonim kalmasını sağlar. U, R_s ve T_s doğru şekilde doğrulanırsa S 'nin meşru olduğuna inanır.

Sonuç olarak, önerilen şema, haberleşme mesajlarını engelleyebilen saldırganın varlığında bile oturum anahtarı SK_{s-k} 'nin güvenliğini ve güncelliğini sağlar.

- Karşılıklı Kimlik Doğrulama: İletişimin güvensiz bir kanal üzerinden gerçekleşmesi nedeniyle karşılıklı kimlik doğrulama sağlamak son derece önemlidir. Sunucunun hem sensör cihazının hem de kullanıcının kimliklerini doğrulaması gereklidir. Benzer şekilde, kullanıcı da herhangi bir mesaj göndermeden önce sunucunun kimliğini doğrulaması gerekmektedir. Önerilen şemada, sunucu, cihaz ve kullanıcı arayüzü arasında karşılıklı kimlik doğrulama sağlanarak iletişim sürecinde yer alan her bir tarafın kimliğinin doğruluğu güvence altına alınır. Karşılıklı kimlik doğrulamayı değerlendirmek için, protokoldeki her katılımcının (cihaz, kullanıcı ve sunucu) çeşitli varsayımları dikkate alınır.

Varsayımlar:

- $K \rightarrow S : \{DID_k, M_k, M_{k-s}, T_1\}$
- $S \rightarrow K : \{R_s, T_s\}, S \rightarrow Cihaz : \{SK_{s-k}, T_s\}$
- $Cihaz \rightarrow S : DID_c, M_c, T_c$
- S, K 'nin K_{s-k} 'nin güncel olduğunu bildiğine inanır. K, S 'nin K_{s-k} 'nin güncel olduğunu kabul eder. S, K 'nin R_k 'yi ürettiğini varsayar. K, S 'nin R_s 'yi ürettiğini varsayar. $S, Cihaz$ 'ın M_c 'yi ürettiğini varsayar. $Cihaz, SK_{s-k}$ 'nin güncel ve gizli olduğunu kabul eder.

Mantıksal Çıkarımlar:

- $\{DID_k, M_k, M_{k-s}, T_1\}$ 'yi aldığı anda, S aşağıdaki kontrolleri gerçekleştirir:
 1. S, T_1 'in güncelliğini kontrol eder. $S, M_{k-s} = H(DID_k || R_k || T_1)$ 'i yeniden hesaplar.
 2. Yeniden hesaplanan M_{k-s}^* , alınan M_{k-s} ile eşleşirse, S, K 'nin meşru olduğuna ve DID_k 'nin geçerli olduğuna kanaat getirir.
- $\{R_s, T_s\}$ 'yi aldığı anda, K aşağıdaki kontrolleri yapar:
 1. K, T_s 'nin güncelliğini kontrol eder ve R_s 'yi doğrular.
 2. R_s geçerliyse, K, S 'nin meşru olduğuna ve oturum anahtarı K_{s-k} 'nin güncel olduğuna inanır.
- $\{SK_{s-k}, T_s\}$ 'yi aldığı anda, sensör cihazı aşağıdaki kontrolleri yapar:
 1. $Cihaz, T_s$ 'nin güncelliğini kontrol eder. SK_{s-k} 'nin S tarafından üretildiği cihaz tarafından doğrulanır. SK_{s-k} geçerliyse, $Cihaz S$ 'nin meşru olduğuna ve oturum anahtarı K_{s-k} 'nin güncel ve gizli olduğuna kanaat getirir.
- $\{DID_c, M_c, T_c\}$ 'yi aldığı anda, S aşağıdaki kontrolleri yapar:
 1. S, T_c 'nin güncelliğini kontrol eder. S, M_c 'nin cihaz tarafından üretildiğini doğrular. M_c geçerliyse, S cihaz'ın meşru ve iletişimin güvenli olduğunu kabul eder.

Sonuçlar:

- S, M_{k-s} 'yi doğrularsa K 'nin meşru olduğunu kabul eder. S, DID_k ve $H(PWD_k)$ 'yi doğrulamada kullandığı için K 'nin anonim kalmasını sağlar.
- K, R_s ve T_s paketlerini doğrularsa S 'nin meşru olduğuna inanır. $K, oturum anahtarı K_{s-k} 'nin güncel ve gizli kaldığını varsayar.$

- K anonim kalır çünkü DID_k , ID_k ve $H(PWD_k)$ 'den türetilir ve bunlar hiçbir zaman açık metin olarak iletilmez.
- Cihaz, SK_{s-k} ve T_s paketlerini doğrularsa S 'nin meşru bir birim olduğunu kabul eder. Cihaz, oturum anahtarı K_{s-k} 'nin güncel ve gizli olduğuna inanır.

Sonuç olarak önerilen şema, cihaz, sunucu ve kullanıcı arasında karşılıklı kimlik doğrulamayı etkin bir şekilde sağlar.

- Çevrimdışı Tahmin Saldırılarına Karşı Dayanıklılık: Saldırgan, sensör cihazında depolanan parametreleri çıkartarak kimlik bilgilerini, şifreleri ve anahtarları tahmin etmeye çalışabilir. Ancak, saldırgan ID_c 'yi kullanarak, sensör düğümünün şifresini bilmeden DID_c 'yi elde edemez. Aynı şekilde, r_k bilinmeden rastgele olarak R_k , M_{k-s} ve SK_{s-k} değerleri doğru şekilde yeniden üretilemez. Bu nedenle, saldırgan r_k ve r_s bilgilerine sahip olmadan doğru R_k , R_s ve oturum anahtarını tahmin edemez. Sonuç olarak, önerilen mekanizma çevrimdışı tahmin saldırılarına karşı etkili bir şekilde direnç gösterir.

Bu durum, hash fonksiyonlarının kullanımının, hassas verilerin korunmasında ne kadar kritik olduğunu göstermektedir. Hash fonksiyonları, giriş değerlerini tahmin edilemez çıktılara dönüştürerek, saldırganların doğru değerleri tahmin etme olasılığını önemli ölçüde azaltır. Bu özellik, özellikle kimlik doğrulama sistemlerinde, saldırıya uğrayabilecek bilgilerin güvenliğini sağlamak için kritik öneme sahiptir.

- Kullanıcı Taklit Saldırılarına Karşı Koruma: Kullanıcı taklit saldırısı, saldırganın kullanıcının kimlik bilgilerini elde ederek veya önceki oturumlardan elde ettiği mesajları değiştirerek meşru bir kullanıcı gibi davranmasını içerir. Bu tür saldırılarda, saldırganın geçerli bir oturum başlatabilmesi için kullanıcının kimlik doğrulama parametrelerini (DID_k , M_k , M_{k-s} ve R_k) bilmesi gereklidir. Ancak, bu değerleri elde etmek oldukça zordur, çünkü şifreleme ve hash teknikleri ile korunmaktadırlar ve oturum süresi sınırlıdır.

Önerilen şema, kullanıcı taklit saldırılarına karşı aşağıdaki güvenlik mekanizmaları ile korunmaktadır:

1. Kullanıcı Parametrelerinin Elde Edilmesi:

- Saldırganın, kullanıcıya ait DID_k , R_k ve PWD_k gibi hassas değerlere erişmesi gerekir.

- Bu değerler şifreleme ve hash teknikleri ile korunur. Kullanıcı, DID_k ve PWD_k değerlerini düz metin olarak değil, tersine mühendislik yapılması oldukça zor olan şifrelenmiş biçimde saklar.

2. Mesajların Yakalanması:

- Saldırgan, kullanıcı ve sunucu arasında değişen mesajları yakalamalı ve değiştirmelidir.
- Önerilen yöntem, mesaj bütünlüğü ve güncelliğini garanti etmek için şifreleme ve zaman damgası mekanizmalarını içeren güvenli iletişim kanalları kullanır. Mesaj yakalansa bile, hash ve dijital imzaların kullanımı sayesinde saldırı tarafından fark edilmeden değiştirilemez.

3. Geçerli Giriş İsteği Oluşturma:

- Saldırganın, kullanıcıyı taklit etmek için geçerli bir giriş isteği oluşturması gerekir.
- Güvenlik Mekanizması:
 - * Benzersiz oturum parametreleri, her oturum için benzersiz rastgele değerler (R_k ve R_s) ve zaman damgaları (T_1 , T_2) kullanır. Bu değerler, oturuma özgü anahtarlar ve kimlik doğrulama kodları (M_k , M_{k-s}) oluşturmak için kullanılır.
 - * Hash, bu değerleri güvenli bir şekilde birleştirmek için kullanılır. Örneğin, $M_{k-s} = H(DID_k || R_k || T_1)$ yalnızca geçerli DID_k , R_u ve T_1 bilgilerini bilen meşru kullanıcının doğru M_{k-s} 'yi üretebilmesini sağlar.

Bu adımlar, önerilen şemanın kullanıcı taklit saldırılarına karşı etkili bir koruma sağladığını göstermektedir.

- Sensör Düşümü Taklit Saldırısı: Saldırgan, sunucuya $\{AUT_{REQ}, M_{c-s}, DID_c, T_1\}$ istek mesajı göndererek sensör düşümlerini taklit etmeye çalıştığı kabul edilsin. Sunucu, gönderilen istek mesajını aldıktan sonra $M_{c-s} = E(H(DID_c || H(PWD_c) || T_1))$ formülünü hesaplayarak mesajın yasal bir sensör düşümünden gelip gelmediğini doğrular. M_{c-s}^* ile M_{c-s} 'nin eşit olup olmadığını kontrol eder. Önerilen şemada, hem sensör hem de sunucu, doğrulanabilir M_{c-s} 'yi hesaplayabilmek için gizli özel anahtar değeri (K_1) bilgisine sahip olmalıdır. K_1 , sunucu tarafından üretilir ve ağdaki EB mesajlarının doğrulanması için sensör düşümlerin belleklerinde saklanır. Hem sensör hem de sunucu, kimlik doğrulama ve oturum anahtarı aşamasında gösterildiği gibi paylaşılan

gizli anahtarlarını kullanmalıdır. Uygun anahtar olmadan, istek mesajı asla başarılı bir şekilde doğrulanamaz. Ağa katılım başarısız olur çünkü DATA ve ACK paketleri açılmaz. Bu nedenle, önerilen şema sensör düğümü taklit saldırısına karşı dirençlidir.

Bu süreç, ağ güvenliğinin kritik yönünü temsil eder ve kullanılan güvenlik mekanizmaları, sensör düğümlerinin kimliklerinin doğrulanmasında merkezi bir rol oynar. Sunucu ve sensör düğümler arasındaki bu etkileşim, her iki tarafın da sahip olduğu gizli anahtarlar aracılığıyla güçlendirilmiş güvenlik önlemleri sayesinde sağlanır. Bu yaklaşım, potansiyel saldırganların yetkisiz erişimlerini engeller ve ağın bütünlüğünü korur.

- **Tekrar Saldırılarına Karşı Dayanıklılık:** Önerilen şema, kimlik doğrulama sürecine zaman damgaları ve nonce değerleri ekleyerek yeniden oynatma saldırılarına karşı etkili bir şekilde direnç gösterir. Saldırgan giriş aşamasında giriş isteğini $(DID_k, M_k, M_{k-s}, T_1)$ yakalar. Eğer saldırgan elde ettiği bu giriş isteğini yeniden kullanmaya çalışırsa, sunucu T_1 'in geçerliliğini kontrol ederek saldırıyı kolayca tespit edebilir. Ayrıca, saldırgan yeni bir zaman damgası (Te_1) oluşturup yakalanan mesajları yeniden kullanmaya çalışsa bile, sunucu bu uyumsuzluğu tespit eder. Çünkü yeni oluşturulan zaman damgası Te_1 , meşru zaman damgası T_1 kullanılarak hesaplanan M_{k-s} içindeki beklenen değerle eşleşmez. Bu dinamik unsurların dahil edilmesi, her kimlik doğrulama oturumunun benzersiz olmasını ve kötü niyetli şekilde tekrarlanamamasını sağlar. Sonuç olarak, önerilen şema, eski veya değiştirilmiş mesajların tanınmasını ve reddedilmesini sağlayarak yeniden oynatma saldırılarına karşı güçlü bir koruma sunar.

Varsayımlar:

- K, S 'nin K_{S-K} üzerinde yetkisi olduğunu varsayar.
- S, K 'nin K_{S-K} üzerinde yetkisi olduğunu kabul eder.

Mesaj Alışverişi:

- $K \rightarrow S : \{DID_k, M_k, M_{k-s}, T_1\}, S \rightarrow K : \{M_s, M_{s-k}, T_2\}$

Mesaj Alışverişinden Sonrası:

- $S, \{DID_k, M_k, M_{k-s}, T_1\}$ mesajını alır ve T_1 'i kontrol ederek tazeliğini doğrular.
- S, K 'nin $\{DID_k, M_k, M_{k-s}, T_1\}$ dediğine inanır.
- $S, \{DID_k, M_k, M_{k-s}, T_1\}$ 'in güncel olduğuna inanır.

- S , K_{S-K} 'nin iletişim için iyi bir anahtar olduğuna inanır.
- K , $\{M_s, M_{s-k}, T_2\}$ mesajını alır ve T_2 'yi kontrol ederek güncelliğini doğrular.
- K , S 'nin $\{M_s, M_{s-u}, T_2\}$ dediğine inanır.
- K , $\{M_s, M_{s-k}, T_2\}$ 'in güncel olduğuna inanır.

Bu kabuller, her kimlik doğrulama oturumunun benzersiz olduğunu ve tekrar mesajlarının tanınıp reddedildiğini sağlayarak yeniden oynatma saldırılarına karşı güçlü bir koruma sağlar.

- Anahtar Güncelliği: Önerilen şemada, hesaplanan oturum anahtarı $SK_{s-k} = H(R_k || R_s)$ olarak belirlenir; burada R_k , rastgele değere bağlı bir parametredir ve R_s ise r_s kullanılarak hesaplanır. Rastgele sayıların ve zaman damgalarının kullanımı, her oturum için yeni ve benzersiz bir oturum anahtarı üretilmesini sağlar. Dolayısıyla, kimlik doğrulama mekanizması anahtar tazeliğini garanti eder.

ASN değeri, anahtar değişimi için kullanılır. Bu değer her zaman aralığında artırılır, böylece anahtarlar düzenli olarak güncellenir.

Mesaj Yorumu: Örneğin, kullanıcıdan sunucuya giden bir mesaj $K \rightarrow S : \{M\}_K$ olarak gösterilir, burada $\{M\}_K$, M mesajının K anahtarı ile şifrelenmiş halini temsil eder.

Başlangıç Varsayımları: Katılımcıların (örneğin, sunucu ve sensör düğümü) mesajların güncelliği ve doğruluğu hakkında başlangıçta sahip oldukları varsayımları ifade eder. S , K 'nin paylaşılan gizli bir anahtarı bildiğini ve bu anahtarın güncel olduğunu kabul eder.

- $K \rightarrow S : \{R_k, T_k\}, S \rightarrow K : \{R_s, T_s\}$
- S , K 'nin K_{s-k} 'nin güncel olduğunu kabul eder.

Analiz, önerilen şemanın karşılıklı kimlik doğrulama, oturum anahtarı güvenliği ve yeniden oynatma saldırılarına direnç sağladığını doğrular.

- Ortadaki Adam Saldırısı: Saldırgan, kimlik doğrulama istek mesajını yakaladığında, sunucu kullanıcıdan M_s , M_{s-k} ve T_2 içeren mesajı alır. Saldırgan, M_s^* , M_{s-k}^* , T_2^* paketlerini değiştirmeye çalışabilir. Ancak, saldırgan, nonce ve hash fonksiyonlarının kullanımı nedeniyle gerçek zaman damgası T_2 'ye karşılık gelen parametreleri doğru şekilde hesaplayamaz. Saldırgan mesajları değiştirmeye çalışsa bile, sunucu doğrulama süreci sırasında tutarsızlıkları tespit eder. Bu, kimlik doğrulama sürecinin ortadaki

adam saldırılarına karşı güvenli kalmasını sağlar.

Mesaj Aışverişı:

$$- K \rightarrow S : \{M_k, M_{k-s}, T_1\}, S \rightarrow K : \{M_s, M_{s-k}, T_2\}$$

Mesaj Aışverişinden Sonra:

- $S, \{M_k, M_{k-s}, T_1\}$ mesajını alır ve T_1 'i güncelliğı doğrulamak için kontrol eder.
- S, K 'nin $\{M_k, M_{k-s}, T_1\}$ değerlerinin doğru olduğunu kabul eder.
- $S, \{M_k, M_{k-s}, T_1\}$ 'in güncel olduğunu kabul eder.
- $K, \{M_s, M_{s-k}, T_2\}$ mesajını alır ve T_2 'yi güncelliğini doğrulamak için kontrol eder.
- K, S 'nin $\{M_s, M_{s-k}, T_2\}$ parametrelerinin uygun biçimde olduğunu varsayar.
- $K, \{M_s, M_{s-k}, T_2\}$ 'in güncel olduğunu kabul eder.

Bu varsayımlar, kullanıcı ve sunucu arasında karşılıklı kimlik doğrulamasını sağlar ve iletişimin güvenli olduğunu doğrular, böylece ortadaki adam saldırılarına karşı koruma sağlanır.

- İçeriden Gelen Saldırı: Bu saldırı türü, ağ içindeki meşru bir kullanıcı veya sensör düğümünün, oturum anahtarları gibi diğer kullanıcıların veya sensör düğümlerinin anahtar değerlerine yetkisiz erişim sağlayarak bu anahtarları güvenlik ihlallerini başlatmak için kullanması durumunda gerçekleşir. Bu tür saldırıya karşı savunma yapmak zordur çünkü saldırgan sistem içinde belirli ayrıcalıklara zaten sahiptir. Önerilen şema, içeriden saldırılarla ilişkili riskleri birkaç ana mekanizma aracılığıyla etkili bir şekilde azaltır:

İlk olarak, kullanıcı veya sensör düğümü ile sunucu arasında paylaşılan oturum anahtar çiftleri (SK_{s-k} veya SK_{s-c}), her bir varlık ve sunucuya özgü benzersiz unsurların kombinasyonu kullanılarak üretilir. Böylece, meşru bir kullanıcı veya sensör düğümü kendi oturum anahtarına erişse bile, diğer kullanıcıların veya sensör düğümlerinin oturum anahtarlarını çıkarabilmesini engeller.

İkinci olarak, kimlik doğrulama sürecinde kullanılan dinamik kimlik (DID_k kullanıcılar için ve DID_c sensör düğümleri için) ve diğer kimlik bilgileri, kullanıcının veya sensörün şifresini ve kimliğini girdi olarak alan güvenli hash fonksiyonları aracılığıyla üretilir. Bu, her dinamik kimliğin benzersiz olmasını ve aynı ağa erişimi olan diğer kullanıcılar veya sensör düğümleri tarafından kopyalanamamasını sağlar.

Ayrıca, anahtar kurulum aşamasında, sunucu iletilen tüm verilerin şifrelenmesini sağlar ve yeniden oynatma saldırılarını önlemek için zaman damgaları ve nonce'lar içerir. Zaman damgaları ve nonce'lar, oturum anahtarlarının güncel kalmasını ve saldırgan tarafından önceki iletişimlerin tekrarlanmasını veya manipüle edilmesini önleyerek ek bir güvenlik katmanı ekler. Örneğin, (K) veya (C) sunucuyla bir oturum başlattığında, içeriden saldırılara karşı güvenliği sağlamak için aşağıdaki adımlar uygulanır:

1. Kullanıcı veya sensör düğümü, dinamik kimliklerini (DID_k veya DID_c), nonce ve zaman damgası içeren bir giriş isteği gönderir. Sunucu, isteğin güncelliğini zaman damgası ve nonce kullanarak doğrular. Böylece tekrar saldırıları bertaraf edilmiş olur.
2. S, dinamik kimlik ve sunucuya özgü parametreler kullanarak oturum anahtarı (K_{s-k} veya K_{s-c}) oluşturur. Bu oturum anahtarı benzersizdir ve başka hiçbir kullanıcı veya sensör düğümü tarafından türetilemez.
3. Kullanıcı, sensör düğümü ve sunucu tarafından karşılıklı kimlik doğrulaması gerçekleştirir, böylece haberleşmede bulunan birimlerin meşru olduğu ve oturum anahtarının güvenle kurulabileceği garanti edilir.

Ek olarak, önerilen şema, oturum anahtarlarının sürekli izlenmesini ve periyodik olarak yeniden değerlendirilmesini içerir. Oturum anahtarının tehlikeye girdiği şüphelenilirse, hemen iptal edilir ve yeni bir anahtar üretilir. Sensörler, sunucu ile etkileşimde bulunarak periyodik olarak oturum anahtarlarını yeniler, bu da anahtarların güvenli ve güncel kalmasını sağlar. Bu sağlam kriptografik mekanizmaları ve güvenlik önlemlerini içeren önerilen şema, kullanıcılar, sensör düğümleri ve sunucu arasında paylaşılan anahtarların yalnızca hak sahiplerine özel kalmasını ve ilgisiz taraflara ifşa edilmemesini sağlar.

Sonuç olarak, önerilen şemanın tasarımı ve uygulaması, kullanıcılar ve sensör düğümleri için oturum anahtarlarının benzersizliğini ve gizliliğini sağlayarak ve titiz kimlik doğrulama ve şifreleme teknikleri kullanarak içeriden saldırılara karşı etkili bir şekilde koruma sağlar.

- Sunucu Taklit Saldırısı: Bu saldırı türünde, saldırganın meşru bir sunucu gibi davranarak kullanıcıya veya sensör düğümüne istek mesajı göndermesi ve bu istek mesajının kullanıcı veya sensör tarafından meşru bir sunucu olarak başarılı şekilde doğrulanması durumunda gerçekleşir. Saldırgan, sensör için $\{E("HELLO"), M_{s-c}, \text{cihaz bağlantı}$

adresi, $E(PK), T_3$ } veya kullanıcı için $\{M_s, M_{s-k}\}$ gibi bilgileri içeren bir istek mesajı T_2 göndererek sunucuyu taklit etmeye çalıştığı varsayalım. Saldırgan, M_{s-c}, M_s, M_{s-k} ve $E(PK)$ hesaplamak için ortak kullanılan anahtar K_1 ve $SK_{s-k} = H(R_k || R_s)$ kullanılmalıdır. Ancak, r_k ve r_s bilgileri olmadan, hash işlemi özellikleri nedeniyle bu parametreler hesaplanamaz.

Sunucu taklit saldırısını önlemek için kullanılan güvenlik mekanizmaları şunlardır:

1. Saldırgan, sunucuya ait M_{s-c}, M_s, M_{s-k} ve $E(PK)$ gibi kimlik doğrulama kodlarını ve şifreleme anahtarlarını hesaplamak zorundadır. Bu hesaplamaları yapabilmek için saldırırganın ortak kullanılan anahtar K_1 ve oturum anahtarı $SK_{s-k} = H(R_k || R_s)$ bilgilerine sahip olması gerekir. Ancak, bu anahtarlar sadece meşru sunucu tarafından bilinir ve güvenli bir şekilde korunur.
2. İletişimde kullanılan mesajlar, zaman damgaları (T_3, T_2) ile birlikte gönderilir. Zaman damgaları, mesajların güncelliğini ve tekrar saldırılarına karşı korunmasını sağlar. Saldırganın, geçerli ve güncel zaman damgalarını tahmin etmesi veya üretmesi zordur.
3. Hash fonksiyonları ve şifreleme teknikleri, mesajların bütünlüğünü ve gizliliğini sağlar. Örneğin, $M_{s-c} = H(DID_c || H(PWD_c) || T_1)$ formülü ile hesaplanır ve sadece meşru sunucu bu hesaplamaları yapabilir. Ayrıca, $E(PK)$ gibi şifreli mesajlar, yalnızca doğru anahtarlarla çözülebilir. Saldırgan, bu anahtarlara sahip olmadan şifreli mesajları çözemez veya yeni geçerli mesajlar üretemez. Bu mekanizmalar sayesinde, kullanıcı ve sensör düğümleri, aldıkları mesajların meşru bir sunucudan gelip gelmediğini doğrulayabilirler. Kullanıcı ve sensör, meşru bir sunucudan gelen mesajların doğrulamasını yapamadığında, iletişimi reddeder ve saldırı başarısız olur. Böylece, sunucu taklit saldırılarına karşı etkili bir koruma sağlanmış olur. Bu tür güvenlik önlemleri, sistemin bütünüyle korunmasını sağlar ve yetkisiz erişim girişimlerini başarıyla engeller. Sunucu, kullanıcı ve sensör düğümleri arasında güvenli ve doğrulanabilir bir iletişim ortamı oluşturarak, sistemin güvenliğini ve bütünlüğünü garanti altına alır.

6.2.2. Performans Değerlendirmesi ve İşlevsellik Analizi

Kimlik doğrulama şemasının etkili olabilmesi için ilgili varlıkları doğrularken hesaplama maliyetinin göz önünde bulundurulması gerekir. Bu bölümde, önerilen

mekanizmanın hesaplama maliyeti ve güvenlik özellikleri açısından performansı değerlendirilmiş ve mevcut yaklaşımlarla karşılaştırılmıştır. Hesaplama maliyeti, herhangi bir kimlik doğrulama şemasının performansını ölçmede en önemli faktörlerden biridir (Höglund vd., 2023). Tablo 22, önerilen yaklaşımın güvenlik özellikleri ilgili mekanizmalarla karşılaştırılmasını gösterir. Yapılan karşılaştırma, önerilen şemanın gerekli güvenlik özelliklerini sağladığını ve bilinen saldırılara karşı direnç gösterdiğini ortaya koymuştur.

Tablo 22. Önerilen mekanizmanın güvenlik özelliklerinin ilgili çalışmalarla karşılaştırılması: (A: Kullanıcı Anonimliği, B: Karşılıklı Kimlik Doğrulama, C: Ortadaki Adam Saldırısı, D: İçeriden Saldırı, E: Kötü Niyetli Kullanıcı Saldırısı, F: Yeniden Oynatma Saldırısı, G: Çevrimdışı Tahmin Saldırısı, H: Gizli Sunucu Saldırısı, I: Sensör Taklit Saldırısı, K: Mesaj Gizliliği)

Çalışma	A	B	C	D	E	F	G	H	I	K
Önerilen Şema	Evet	Evet	Evet	Evet	Evet	Evet	Evet	Evet	Evet	Evet
(Turkanović vd., 2014)	Hayır	Hayır	Evet	Hayır	Hayır	Evet	Hayır	Evet	Hayır	Evet
(Bae & Kwak, 2020)	Evet	Evet	Hayır	Hayır	Evet	Evet	Hayır	Hayır	Evet	Hayır
(Challa vd., 2018)	Hayır	Hayır	Evet	Evet	Evet	Evet	Hayır	Hayır	Evet	Evet
(Wu vd., 2018)	Evet	Evet	Evet	Evet	Evet	Hayır	Hayır	Evet	Hayır	Evet
(Aydin vd., 2018)	Evet	Evet	Hayır	Hayır	Hayır	Evet	Evet	Evet	Hayır	Evet
(Durairaj & Muthuramalingam, 2018)	Evet	Hayır	Hayır	Hayır	Evet	Evet	Hayır	Evet	Hayır	Evet
(Lu vd., 2019)	Evet	Hayır	Evet	Evet	Evet	Evet	Hayır	Evet	Evet	Evet
(Fariss vd., 2022)	Evet	Hayır	Evet	Hayır	Evet	Hayır	Evet	Hayır	Hayır	Evet
(Saini vd., 2024)	Evet	Hayır	Evet	Evet	Hayır	Evet	Evet	Evet	Evet	Evet
(Vijaykumar vd., 2024)	Evet	Evet	Evet	Evet	Evet	Evet	Evet	Hayır	Hayır	Evet
(Guo vd., 2024)	Evet	Evet	Evet	Evet	Hayır	Evet	Evet	Hayır	Evet	Evet
(Huang, 2024)	Evet	Evet	Evet	Evet	Evet	Evet	Evet	Hayır	Hayır	Evet

Tablo 23, önerilen yaklaşımın literatürde yer alan çalışmalarla karşılaştırıldığında kimlik doğrulama ve sisteme giriş aşamalarının hesaplama maliyetini göstermektedir. Tabloda, T_H hash fonksiyonu için geçen süreyi, T_{FE} bulanık doğrulayıcı hesaplama süresini, T_{ECC} eliptik eğri ile şifrelemede kullanılan nokta çarpımını ve T_S şifreleme ve şifre çözme süreçlerindeki

zaman karmaşıklığını temsil eder. T_X , XOR işlemi çalışma süresini; T_R , tek bir rastgele sayı üretim maliyetini; T_{MFC} ise önerilen çok fonksiyonlu devreyi ifade eder. T_{PUF} , kimlik doğrulama sürecinde kullanılan PUF işlemleriyle ilişkili zamanı temsil eder. Sunulan yöntem, XOR ve tek yönlü hash fonksiyonu kullanarak daha düşük hesaplama karmaşıklığı sağlar. Sonuç olarak, üs alma ve nokta çarpımı işlemleri kullanan önceki çalışmalara kıyasla daha hafif bir kullanıcı kimlik doğrulama şemasıdır. Ancak bu karşılaştırmada, OSCORE ve IPsec protokollerinin hesaplama maliyetleri dahil edilmemiştir. Bu protokollerin ek yüklerinin hesaba katılmaması, önerilen yöntemin genel hesaplama maliyeti açısından eksik bir değerlendirme sunabilir. Bu nedenle, önerilen yöntemin pratik uygulamalardaki performansını tam olarak anlayabilmek için OSCORE ve IPsec protokollerinin ek yükleri de göz önünde bulundurulmalıdır.

Tablo 24'te, çeşitli mevcut yöntemlerin IoT ağlarındaki toplam bit ve enerji tüketimi açısından karşılaştırmalı bir analizi sunulmaktadır. Örneğin, (Turkanović vd., 2014) yöntemi, toplamda 4000 bit gerektirirken, enerji tüketimi 12.517 mJ olarak ölçülmüştür. Bu, diğer yöntemlere kıyasla daha yüksek bit ve enerji tüketimi anlamına gelir. Benzer şekilde, (Bae & Kwak, 2020) yöntemi, 2720 bit ile 13.633 mJ enerji tüketimiyle dikkat çekmektedir. Bu yöntem, daha az bit gerektirse de, enerji tüketimi açısından daha yüksek maliyetli olabilir. (Challa vd., 2018) yöntemi ise 2528 bit gerektirir ve 11.633 mJ enerji tüketimi ile daha verimli bir seçenek sunar. Bu yöntemin, bit ve enerji tüketimi açısından dengeli bir performans sergilediği görülmektedir. Ayrıca, (Wu vd., 2018) yöntemi de benzer şekilde 2720 bit gerektirir ve enerji tüketimi 12.78 mJ olarak kaydedilmiştir. (Aydın vd., 2018) yöntemi, 2586 bit gerektirirken, enerji tüketimi 10.87 mJ olarak ölçülmüştür. Bu, yöntemlerin çoğuna kıyasla daha düşük enerji tüketimi anlamına gelir ve bu da özellikle enerji verimliliğinin kritik olduğu IoT uygulamalarında avantaj sağlar. (Durairaj & Muthuramalingam, 2018) yöntemi, 3000 bit gerektirir ve enerji tüketimi 10.45 mJ ile düşük enerji tüketimi sunar, bu da onu enerji verimliliği açısından öne çıkan bir yöntem yapar.

(Lu vd., 2019) yöntemi, 2800 bit gerektirir ve 10.90 mJ enerji tüketimiyle verimli bir performans sergiler. (Fariss vd., 2022) yöntemi ise 3200 bit ile 11.00 mJ enerji tüketimi arasında dengeli bir sonuç sunar. Bu yöntemler, düşük bit gereksinimi ve enerji tüketimi ile dikkat çeker. (Saini vd., 2024) yöntemi, 2304 bit ile 11.50 mJ enerji tüketimi sağlar ve en düşük bit gereksinimi ile verimli bir çözüm sunar. (Vijaykumar vd., 2024) yöntemi, 3100 bit ile 11.20 mJ enerji tüketimi arasında dengeli bir performans sunar. (Guo vd., 2024) yöntemi ise, 3200 bit gerektirir ve 11.70 mJ enerji tüketimi ile dikkat çeker. Benzer şekilde, (Huang, 2024) yöntemi de 3180 bit gerektirir ve enerji tüketimi 11.80 mJ olarak kaydedilmiştir.

Tablo 23. Performans karşılaştırması: (T_H , T_{ECC} , T_{ECM} , T_S , T_X , T_{FE} , T_{PUF})

Yöntem	Kullanıcı (K)	Cihaz (C)	Sunucu (S)	Maliyet	Süre (s)
(Turkanović vd., 2014)	$4T_H + 2T_S$	$5T_H$	$3T_H + 4T_S + T_X$	$12T_H + 6T_S + T_X$	0.3195
(Bae & Kwak, 2020)	$4T_H + 3T_S + T_X$	$3T_H + 2T_S + T_X$	$6T_H + 4T_S + 2T_X$	$13T_H + 9T_S + 4T_X$	0.4105
(Challa vd., 2018)	$2T_{ECM} + 10T_H + T_{FE}$	$T_{ECM} + 4T_H$	$4T_H + T_{ECM}$	$4T_{ECM} + 18T_H + T_{FE}$	0.6845
(Wu vd., 2018)	$14T_H + 2T_S$	$4T_H + 2T_S + T_X$	$13T_H + 4T_S + T_X$	$31T_H + 8T_S + 2T_X$	0.6112
(Aydin vd., 2018)	$6T_H + 2T_X$	$2T_H + 4T_S + 3T_X$	$2T_H + 4T_S + 4T_X$	$10T_H + 8T_S + 9T_X$	0.0259
(Durairaj & Muthuramalingam, 2018)	$2T_H + 6T_{ECC}$	$T_H + 2T_{ECC}$	$4T_{ECC}$	$3T_H + 12T_{ECC}$	1.5234
(Lu vd., 2019)	$7T_H + T_S$	$4T_H + 2T_S$	$6T_H + T_S + T_X$	$17T_H + 4T_S + T_X$	0.0519
(Fariss vd., 2022)	$4T_{ECM} + 8T_H + T_S$	$2T_{ECM} + 3T_H$	$2T_{ECM} + 5T_H + T_S$	$8T_{ECM} + 16T_H + 2T_S$	0.0377
(Saini vd., 2024)	$28T_H + 12T_X + T_{ECM}$	$12T_H + 3T_X$	$18T_H + 6T_X + 4T_{ECM} + T_{FE}$	$58T_H + 21T_X + 5T_{ECM} + T_{FE}$	0.4134
(Vijaykumar vd., 2024)	-	$2T_R + 4T_{MFC}$	-	$2T_R + 4T_{MFC}$	0.421
(Guo vd., 2024)	$11T_H$	$7T_H + T_{PUF} + 2T_{ECC}$	$12T_H + T_{ECC}$	$30T_H + 2T_{PUF} + 5T_{ECC}$	N/A
(Huang, 2024)	$5T_{ECM} + 22T_H + T_{FE}$	$4T_{ECM} + 18T_H$	$3T_{ECM} + 8T_H$	$12T_{ECM} + 48T_H + T_{FE}$	0.0260
Önerilen Şema	$3T_H + 2T_S + 3T_X$	$5T_H + 2T_S + 4T_X$	$4T_H + 5T_S + 5T_X$	$12T_H + 9T_S + 12T_X$	0.0281

Bu hesaplama sürecinde, ENERGEST aracı kullanılarak düğümdeki enerji tüketimi ölçülmektedir. Her bir kriptografik işlem sırasında CPU'nun ne kadar süre aktif kaldığı ENERGEST ile tespit edilir ve işlem süresi sonunda bu süre ile CPU'nun güç tüketimi çarpılarak işlem başına enerji tüketimi hesaplanır. ENERGEST aracı, özellikle düşük güçlü

Tablo 24. Farklı yöntemlerin toplam bit ve enerji tüketimi bazında karşılaştırılması

Yöntem	Toplam bit	Toplam enerji tüketimi
Turkanović vd. (2014)	4000	12.517 mj
Bae & Kwak (2020)	2720	13.633 mj
Challa vd. (2018)	2528	11.633 mj
Wu vd. (2018)	2720	12.78 mj
Aydin vd. (2018)	2586	10.87 mj
Durairaj & Muthuramalingam (2018)	3000	10.45 mj
Lu vd. (2019)	2800	10.90 mj
Fariss vd. (2022)	3200	11.00 mj
Saini vd. (2024)	2304	11.50 mj
Vijaykumar vd. (2024)	3100	11.20 mj
Guo vd. (2024)	3200	11.70 mj
Huang (2024)	3180	11.80 mj
Önerilen Şema	2900	11.25 mj

cihazlarda yapılan şifreleme ve veri iletimi işlemlerinin enerji maliyetlerini hesaplamak için güvenilir bir yöntem sunar. Böylece, önerilen yöntemin enerji tüketim verimliliği daha doğru ve güvenilir bir şekilde hesaplanmaktadır.

Önerilen yöntem, 2900 bit gerektirerek 11.25 mJ enerji tüketimi ile dengeli bir sonuç elde eder. Bu karşılaştırma, önerilen yöntemin enerji verimliliği açısından avantajını vurgulamakta ve enerji tasarrufunun düğümlerin operasyonel ömrünü uzatmak için kritik olduğu IoT uygulamaları için uygun olduğunu göstermektedir. Önerilen yöntem, düşük enerji tüketimi ve orta düzeyde bit gereksinimi ile dengeli bir performans sunarak, IoT ağlarının verimliliğini artırmayı amaçlamaktadır. Bu verimlilik, IoT cihazlarının uzun süreli operasyonlarını destekler ve enerji tasarrufu sağlayarak genel ağ performansını iyileştirir.

IIoT ağında konuşlandırılan cihazlar kaynak kısıtlı olduğundan, sensör düğümlerinin enerji tüketimi önemlidir. Ağdaki cihazların enerji harcaması, gerçekleştirilen şifreleme işlemlerinin sayısına ve iletilen veri miktarına bağlıdır. Yapılan analizden, önerilen şemanın hesaplama açısından verimli olduğu açıktır çünkü önerilen yöntem, güncel çalışmalara kıyasla daha az enerji tüketmektedir. Bu durum, önerilen yöntemin, diğerlerine göre daha az işlem gücü gerektirmesi ve veri iletiminde daha etkin yöntemler kullanması sayesinde enerji

tasarrufu sağladığını göstermektedir. Bu özellikler, IIoT ortamlarında enerji kullanımını optimize ederken aynı zamanda güvenlik ve performans standartlarını koruyarak, ağın uzun vadeli sürdürülebilirliğine katkıda bulunur. Ayrıca, düşük enerji tüketimi, cihazların pil ömrünü uzatır ve bakım maliyetlerini azaltır. Dolayısıyla, önerilen yöntem, enerji verimliliği yüksek, maliyet etkin bir çözüm sunarak, kaynak kısıtlı IIoT cihazları için ideal bir seçenektir.



7. SONUÇLAR

Yapılan doktora tezinde, RPL tabanlı IETF 6TiSCH ağlarında güvenli kimlik doğrulama mekanizmalarının geliştirilmesi ve bu yöntemlerin dağıtık şekilde uygulanması üzerine yoğunlaşmıştır. Çalışmanın temel amacı, düşük güçlü ve kayıplı ağlarda güvenliğini artırarak, özellikle IIoT uygulamaları için güvenilir ve ölçeklenebilir bir kimlik doğrulama sistemi geliştirmektir. Bu bağlamda, önerilen mekanizmalar farklı açılardan incelendi ve mevcut yöntemlerle karşılaştırılarak üstünlükleri ortaya kondu.

Merkezi kimlik doğrulama sistemlerinin ölçeklenebilirlik ve verimlilik açısından yetersiz kaldığı tespit edilmiştir. Merkezi kimlik doğrulama yöntemlerinde, tek bir otoriteye olan bağımlılık, ağın genişlemesi ve performansı üzerinde olumsuz etkiler yaratmaktadır. Bu yöntemler, ağın büyümesiyle birlikte artan işlem yükünü yönetmekte zorlanmakta ve tek bir hata noktası oluşturma riski taşımaktadır. Ayrıca, merkezi sistemlerdeki gecikmeler ve veri işleme süreleri, düşük gecikme gerektiren IIoT uygulamaları için uygun olmayabilir.

Bu sorunları aşmak için dağıtık kimlik doğrulama yöntemlerinin geliştirilmesi gerektiği belirlenmiştir. Önerilen mekanizmalarda, ağ içerisindeki belirli düğümlerin kimlik doğrulama işlemlerini üstlenmesi ve bu işlemleri merkezi otoriteye bağımlı olmadan gerçekleştirmesi sağlanmıştır. Bu yaklaşım, ağın genel verimliliğini artırırken, ölçeklenebilirlik sorunlarını da minimize etmektedir. Dağıtık sistemler, her düğümün yerel olarak kimlik doğrulama yapabilmesi sayesinde, ağdaki işlem yükünü eşit şekilde dağıtarak performansı artırır ve merkezi otoriteye bağımlılığı azaltır.

Yapılan simülasyonlar ve deneysel çalışmalar, önerilen mekanizmaların güvenlik ve performans açısından mevcut yöntemlere göre üstün olduğunu göstermiştir. Özellikle, düşük güçlü cihazların enerji tüketimini minimize ederek, ağın genel çalışma süresini uzattığı ve güvenlik tehditlerine karşı daha dirençli hale getirdiği gözlemlenmiştir. Önerilen kimlik doğrulama yöntemi, XOR, AES ve hash işlemleri gibi hafif kriptografik teknikleri kullanarak, cihazların hesaplama yükünü azaltmakta ve enerji verimliliğini artırmaktadır. Bu sayede, IIoT cihazlarının pil ömrü uzatılmakta ve ağın sürdürülebilirliği artırılmaktadır.

Ayrıca, merkezi kimlik doğrulama noktalarının stratejik olarak yerleştirilmesi, ağın güvenliğini sağlamada kritik bir rol oynamaktadır. Bu merkezler, ağın belirli bölgelerinde düğüm kimlik doğrulama işlemlerini gerçekleştirerek, merkezi bir otoritenin yükünü

dağıtmaktadır. Bu strateji, ağın genel güvenliğini artırırken, kimlik doğrulama işlemlerinin hızını ve verimliliğini de optimize etmektedir.

Geliştirilen mekanizmaların uygulanabilirliği ve etkinliği, çeşitli simülasyon ve deneysel çalışmalarla test edilmiştir. Sonuçlar, önerilen çözümün hem güvenlik hem de verimlilik açısından mevcut yöntemlere kıyasla üstün performans sergilediğini göstermektedir. Özellikle, düşük güçlü cihazların enerji tüketimini minimize ederek, ağın genel çalışma süresini uzattığı ve güvenlik tehditlerine karşı daha dirençli hale getirdiği gözlemlenmiştir. Bu bağlamda, dağıtık kimlik doğrulama mekanizmalarının geliştirilmesi, sadece IIoT uygulamalarında değil, aynı zamanda genel olarak IoT teknolojisinin geleceği için de kritik bir öneme sahiptir.

Önerilen kimlik doğrulama mekanizmalarının 6TiSCH protokolü ile entegrasyonu, ağın performansını ve güvenliğini daha da artırmaktadır. 6TiSCH protokolü, düşük güçlü ve kayıplı ağlar için belirli zaman aralıklarında veri iletimi sağlayan zamanlama mekanizmalarını kullanır. Bu protokol, belirli zaman dilimlerinde belirli düğümler arasında veri iletimini düzenleyerek, veri kaybını ve çakışmaları minimize eder. Önerilen kimlik doğrulama mekanizmalarının bu protokol ile uyumlu olarak çalışması, ağın genel verimliliğini artırmakta ve güvenli veri iletimini sağlamaktadır.

Sonuç olarak, geliştirilen güvenli dağıtık kimlik doğrulama mekanizmaları, RPL tabanlı 6TiSCH ağlarında güvenliğini sağlamak için yeni ve etkili bir çözüm sunmakta olup, gelecekteki IIoT uygulamalarında kullanılabilirliği artırmayı hedeflemektedir. Bu mekanizmalar, ağ güvenliğini sağlarken, performans üzerindeki olumsuz etkileri en aza indirmeyi amaçlamaktadır. Bu çalışmanın bulguları, gelecekteki araştırmalar ve uygulamalar için önemli bir referans niteliği taşımaktadır. Özellikle, enerji verimliliği ve güvenlik gereksinimlerinin ön planda olduğu IIoT uygulamaları için kritik çözümler sunmaktadır. Geliştirilen yöntemler, çeşitli gerçek dünya senaryolarında test edilerek ve optimize edilerek, daha geniş kapsamlı ve güvenli IoT uygulamalarına olanak tanıyacaktır. Bu bağlamda, önerilen araştırma konuları ve yöntemler, IIoT ağlarının gelecekteki güvenlik ve performans gereksinimlerine cevap verecek şekilde geliştirilmesine katkıda bulunacaktır.

8. ÖNERİLER VE TARTIŞMA

Önerilen dağıtık kimlik doğrulama mekanizmalarının gerçek dünya uygulamalarında test edilmesi, bu mekanizmaların etkinliğini ve geçerliliğini ortaya koymak için kritik öneme sahiptir. Teorik olarak güçlü görünen yöntemler, pratikte karşılaşılabilecek zorluklar ve değişken koşullar altında beklenen performansı göstermeyebilir. Gerçek dünya senaryolarında, özellikle endüstriyel IoT (IIoT) ağları gibi dinamik ve karmaşık yapılarda bu mekanizmaların ne kadar başarılı olacağı dikkatlice incelenmelidir. Farklı ağ topolojileri, cihaz özellikleri, ağ yoğunluğu, veri iletim hızları ve güvenlik tehditleri gibi değişkenler, bu tür mekanizmaların performansını doğrudan etkileyen faktörlerdir. Özellikle büyük ölçekli IIoT ağlarında dağıtık kimlik doğrulama mekanizmalarının sunduğu avantajlar daha belirgin hale gelmekte, ancak aynı zamanda bu mekanizmaların ölçeklenebilirliği ve performansı da ciddi sınavlardan geçmektedir. Bu nedenle, bu mekanizmaların zayıflıklarının, eksikliklerinin ve iyileştirilmesi gereken yönlerinin ortaya konulması, güvenlik açıklarının giderilmesi için yapılacak çalışmalar açısından oldukça önemlidir.

Ağ güvenliğini artırmanın bir diğer önemli yolu, gelişmiş kriptografik yöntemlerin entegrasyonudur. Kriptografik algoritmalar, ağdaki veri trafiğini ve iletişim süreçlerini güvence altına alarak, yetkisiz erişim ve veri manipülasyonu gibi tehditlere karşı koruma sağlar. Ancak klasik kriptografik algoritmalar, düşük güçlü ve kaynak kısıtlı cihazlarda yüksek enerji tüketimi ve işlem maliyetine yol açabilir. Bu bağlamda, kuantum kriptografisi gibi yeni nesil kriptografik tekniklerin, IIoT ağlarında uygulanabilirliği araştırılmalıdır. Kuantum kriptografisi, kuantum fiziğinin ilkelerine dayanarak yüksek düzeyde güvenlik sağlar ve özellikle uzun vadeli veri gizliliği ve bütünlüğü gerektiren sistemler için cazip bir çözümdür. Ancak, bu tür ileri düzey teknolojilerin, IIoT ağlarındaki düşük güçlü cihazlar üzerindeki etkilerinin ayrıntılı bir şekilde analiz edilmesi gerekmektedir. Kuantum dayanıklı algoritmaların uygulanabilirliği, bu cihazlarda enerji ve işlem yükü açısından optimize edilmelidir. Bu tür araştırmalar, kuantum dayanıklı algoritmaların IIoT cihazlarına uyarlanması, güvenlik gereksinimlerini karşılarken kaynak verimliliğini sağlamayı amaçlayacaktır.

Buna ek olarak, dağıtık kimlik doğrulama mekanizmalarının farklı sektörlerdeki IoT uygulama senaryolarına adapte edilebilirliği de büyük bir önem arz etmektedir. Her

sektör, kendine özgü tehdit modelleri ve güvenlik gereksinimlerine sahiptir. Örneğin, sağlık sektöründe hasta verilerinin gizliliği ve bütünlüğü en yüksek önceliğe sahiptir. Bu sektörde kullanılan kimlik doğrulama protokollerinin, hastaların sağlık verilerinin gizliliğini koruyacak şekilde tasarlanması gerekmektedir. Öte yandan, enerji sektöründe ise akıllı şebekelerde kullanılan cihazların kimlik doğrulama süreçleri ve güvenli veri iletimi hayati önem taşır. Dağıtık enerji üretim sistemlerinde veri bütünlüğü ve güvenliği sağlanmazsa, şebekenin işleyişi sektöre uğrayabilir ve ciddi güvenlik açıkları oluşabilir. Bu nedenle, sektöre özel tehdit modelleri ve güvenlik ihtiyaçları dikkate alınarak özelleştirilmiş kimlik doğrulama ve güvenlik protokolleri geliştirilmelidir. Sektöre özgü bu tür çözümler, hem güvenliği artıracak hem de operasyonel verimliliği sağlayacaktır.

Blokzinciri teknolojisinin de, IIoT cihazları arasında dağıtık kimlik doğrulama süreçlerinin yönetilmesi ve veri bütünlüğünün sağlanması açısından büyük bir potansiyele sahip olduğu görülmektedir. Blokzincirinin merkezi olmayan yapısı, özellikle tek bir otoritenin veya sunucunun arızalanması durumunda sistemin tüm güvenliğini tehlikeye atma riskini ortadan kaldırır. Blokzinciri teknolojisinin entegre edilmesiyle, IIoT ağlarında her bir cihazın kimlik doğrulama süreci güvence altına alınabilir ve bu da sistemin güvenilirliğini artırabilir. Blokzinciri tabanlı kimlik yönetim sistemleri, her bir cihazın güvenilirliğini doğrulamak ve bu cihazlar arasında güvenli bir şekilde veri iletimi sağlamak için kullanılabilir. Ayrıca, blokzincirinin değiştirilemez kayıt tutma yapısı sayesinde, verilerin bütünlüğü korunur ve yetkisiz erişim riskleri minimize edilir. Blokzincirinin entegre edilmesiyle sağlanan güvenlik ve veri bütünlüğü, özellikle kritik altyapılarda ve büyük ölçekli IIoT ağlarında oldukça önemli avantajlar sağlayacaktır.

Makine öğrenimi ve yapay zeka tabanlı güvenlik çözümlerinin IIoT ağlarına entegrasyonu, gelişen tehditlere karşı savunma mekanizmalarının etkinliğini artırabilir. Bu bağlamda, anomali tespit sistemleri ve tehdit önleme çözümleri, yapay zeka ile desteklenerek daha akıllı hale getirilebilir. Makine öğrenimi algoritmaları, ağ trafiği ve cihaz davranışlarını analiz ederek anormal aktiviteleri tespit edebilir ve potansiyel tehditlere karşı proaktif bir savunma sağlar. Özellikle büyük veri setlerinden öğrenme yeteneği ile yapay zeka sistemleri, daha önce görülmemiş tehditleri tespit edebilir ve gerçek zamanlı çözümler üretebilir. Derin öğrenme teknikleri, IIoT ağlarında olası güvenlik ihlallerini tespit etmek ve önlemek için kullanılabilir. Bu tür yapay zeka tabanlı sistemlerin, sürekli değişen ağ tehditlerine karşı daha dirençli ve adaptif bir güvenlik altyapısı oluşturması hedeflenmektedir.

Tüm bu öneriler, gelecekteki araştırmalar için kapsamlı bir yol haritası sunmaktadır. Bu çalışmalar, IIoT ağlarının güvenliğini ve verimliliğini artırmaya yönelik sürekli bir

arařtırma ve geliřtirme abasını teřvik etmektedir. Bu alıřmanın bulguları, gelecekteki arařtırmalar ve uygulamalar iin deęerli bir referans nitelięi tařımakta olup, IoT teknolojisinin geliřimine nemli katkılar saęlamayı hedeflemektedir. Geliřtirilen yntemler, eřitli gerek dnya senaryolarında test edilerek ve optimize edilerek, daha geniř kapsamlı ve gvenli IoT uygulamalarına olanak tanıyacaktır. Bu baęlamda, nerilen arařtırma konuları ve yntemler, IIoT aęlarının gelecekteki gvenlik ve performans gereksinimlerine cevap verecek řekilde geliřtirilmesine katkıda bulunacaktır.



9. KAYNAKLAR

- Accettura, N., Vogli, E., Palattella, M. R., Grieco, L. A., Boggia, G., & Dohler, M. (2015). Decentralized traffic aware scheduling in 6tisch networks: Design and experimental evaluation. *IEEE Internet of Things Journal*, 2(6), 455–470.
- Adarbah, H. Y., Moghadam, M. F., Maata, R. L. R., Mohajerzadeh, A., & Al-Badi, A. H. (2022). Security challenges of selective forwarding attack and design a secure ecdh-based authentication protocol to improve rpl security. *IEEE Access*, 11, 11268–11280.
- Al-Amiedy, T. A., Anbar, M., Belaton, B., Bahashwan, A. A., Hasbullah, I. H., Aladaileh, M. A., & Mukhaini, G. A. (2023). A systematic literature review on attacks defense mechanisms in rpl-based 6lowpan of internet of things. *Internet of Things*, 22, 100741.
- Alhaidari, F. A., & Alqahtani, E. J. (2020). Securing communication between fog computing and iot using constrained application protocol (coap): A survey. *J. Commun.*, 15(1), 14–30.
- Ali, Z. H., Ali, H. A., & Badawy, M. M. (2015). Internet of things (iot): definitions, challenges and recent research directions. *International Journal of Computer Applications*, 128(1), 37–47.
- Alizadeh, S., & Aydin, H. (2019). Energy efficient communication protocol for 6tisch networks. *2019 Seventh International Conference on Digital Information Processing and Communications (ICDIPC)*, (Bildiriler Kitabı: 63–67).
- Anand, N., & Saifulla, M. (2023). En-lakp: Lightweight authentication and key agreement protocol for emerging networks. *IEEE Access*, 11, 28645–28657.
- Ashrif, F. F., Sundarajan, E. A., Ahmed, R., & Hasan, M. K. (2023a). Slae6: Secure and lightweight authenticated encryption scheme for 6lowpan networks. *SENSORNETS*, (Bildiriler Kitabı: 67–78).
- Ashrif, F. F., Sundararajan, E. A., Ahmad, R., Hasan, M. K., & Yadegaridehkordi, E. (2023b). Survey on the authentication and key agreement of 6lowpan: Open issues and future direction. *Journal of Network and Computer Applications*, (103759).
- Aydin, H., Goermues, S., & Jin, Y. (2018). A distributed user authentication mechanism for ietf 6tisch protocol. *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*, (Bildiriler Kitabı: 1–7). IEEE.
- Aydin, H., Gormus, S., & Aydin, B. (2024). A decentralized proxy-jrc authentication system for scalable ietf 6tisch networks. *IEEE Access*, 12, 90952–90970.
- Aydin, B., Aydin, H., & Görmüş, S. (2024). A security mechanism against man in the middle attack in 6tisch networks. *2024 32nd Signal Processing and Communications Applications Conference (SIU)*, (Bildiriler Kitabı: 1–4).

- Aydın, H., & Görmüş, S. (2018). A new user authentication mechanism for ietf 6tisch protocol. *2018 26th Signal Processing and Communications Applications Conference (SIU)*, (Bildiriler Kitabı: 1–4).
- Bae, W.-i., & Kwak, J. (2020). Smart card-based secure authentication protocol in multi-server iot environment. *Multimedia Tools and Applications*, 79(23), 15793–15811.
- Bagula, B., & Erasmus, Z. (2015). Iot emulation with cooja. *ICTP-IoT workshop*, (99).
- Baker, F., Black, D. L., Nichols, K., & Blake, S. L. (1998). Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. RFC 2474.
- Berini, A. D. E., Ferrag, M. A., Farou, B., & Seridi, H. (2023). Hcala: Hyperelliptic curve-based anonymous lightweight authentication scheme for internet of drones. *Pervasive and Mobile Computing*, 92, 101798.
- Boyes, H., Hallaq, B., Cunningham, J., & Watson, T. (2018). The industrial internet of things (iiot): An analysis framework. *Computers in industry*, 101, 1–12.
- Challa, S., Das, A. K., Odelu, V., Kumar, N., Kumari, S., Khan, M. K., & Vasilakos, A. V. (2018). An efficient ecc-based provably secure three-factor user authentication and key agreement protocol for wireless healthcare sensor networks. *Computers & Electrical Engineering*, 69, 534–554.
- Das, M. L. (2009). Two-factor user authentication in wireless sensor networks. *IEEE transactions on wireless communications*, 8(3), 1086–1090.
- De Guglielmo, D., Brienza, S., & Anastasi, G. (2016). Ieee 802.15. 4e: A survey. *Computer Communications*, 88, 1–24.
- Dehghan, P., Aydın, H., & Görmüş, S. (2020). Resource management in distributed 6tisch networks. *2020 5th International Conference on Computer Science and Engineering (UBMK)*, (Bildiriler Kitabı: 349–354).
- Díaz, J. P., & Almenares, F. (2023). Integrating an optimised puf-based authentication scheme in oscore. *Ad Hoc Networks*, 140, 103038.
- Dunkels, A., Eriksson, J., Finne, N., & Tsiftes, N. (2011). Powertrace: Network-level power profiling for low-power wireless networks.
- Dunkels, A., Gronvall, B., & Voigt, T. (2004). Contiki - a lightweight and flexible operating system for tiny networked sensors. *29th Annual IEEE International Conference on Local Computer Networks*, (Bildiriler Kitabı: 455–462).
- Dunkels, A., Osterlind, F., Tsiftes, N., & He, Z. (2007). Software-based on-line energy estimation for sensor nodes. *Proceedings of the 4th workshop on Embedded networked sensors*, (Bildiriler Kitabı: 28–32).
- Duquennoy, S., Elsts, A., Al Nahas, B., & Oikonomou, G. (2018). Tsch and 6tisch for contiki: Challenges, design and evaluation. *2017 13th International Conference on Distributed Computing in Sensor Systems*, (Bildiriler Kitabı: 11–18). IEEE.
- Durairaj, M., & Muthuramalingam, K. (2018). A new authentication scheme with elliptical curve cryptography for internet of things (iot) environments. *Int. J. Eng. Technol*, 7(2.26), 119–24.

- Eriksson, J., Dunkels, A., Finne, N., Österlind, F., Voigt, T., & Tsiftes, N. (2008). Demo abstract: Mspsim-an extensible simulator for msp430-equipped sensor boards. *5th European Conference on Wireless Sensor Networks (EWSN 2008)*, 30 Jan-1 Feb 2008, Bologna, Italy.
- Fariss, M., El Gafif, H., & Toumanari, A. (2022). A lightweight ecc-based three-factor mutual authentication and key agreement protocol for wsns in iot. *International Journal of Advanced Computer Science and Applications*, 13(6).
- Fawwaz, D. Z., & Chung, S.-H. (2023). Adaptive trickle timer for efficient 6tisch network formation using q-learning. *IEEE Access*.
- Floyd, S., Ramakrishnan, D. K. K., & Black, D. L. (2001). The Addition of Explicit Congestion Notification (ECN) to IP. RFC 3168.
URL <https://www.rfc-editor.org/info/rfc3168>
- Gong, B., Zheng, G., Waqas, M., Tu, S., & Chen, S. (2023). Lcdma: Lightweight cross-domain mutual identity authentication scheme for internet of things. *IEEE Internet of Things Journal*.
- Görmüş, S., Aydın, H., & Ulutaş, G. (2018). Security for the internet of things: a survey of existing mechanisms, protocols and open research issues. *Journal of the Faculty of Engineering and Architecture of Gazi University*, 33(4), 1247–1272.
- Goyat, R., Kumar, G., Saha, R., & Conti, M. (2023). Pribadi: A decentralized privacy-preserving authentication in wireless multimedia sensor networks for smart cities. *Cluster Computing*, (Bildiriler Kitabı: 1–17).
- Görmüş, S. (2015). Synchronisation in 6tisch networks. *2015 23rd Signal Processing and Communications Applications Conference (SIU)*, (Bildiriler Kitabı: 535–539).
- Görmüş, S., & Yavuz, A. F. (2017). A protocol for internet of things : Ietf 6tisch. *2017 25th Signal Processing and Communications Applications Conference (SIU)*, (Bildiriler Kitabı: 1–4).
- Gunnarsson, M., Brorsson, J., Palombini, F., Seitz, L., & Tiloca, M. (2021). Evaluating the performance of the oscore security protocol in constrained iot environments. *Internet of Things*, 13, 100333.
- Gunnarsson, M., Malarski, K. M., Höglund, R., & Tiloca, M. (2022). Performance evaluation of group oscore for secure group communication in the internet of things. *ACM Transactions on Internet of Things*, 3(3), 1–31.
- Guo, P., Liang, W., & Xu, S. (2024). A privacy preserving four-factor authentication protocol for internet of medical things. *Computers & Security*, 137, 103632.
- Haj-Hassan, A., Imine, Y., Gallais, A., & Quoitin, B. (2022). Zero-touch mutual authentication scheme for 6tisch industrial iot networks. *2022 International Wireless Communications and Mobile Computing (IWCMC)*, (Bildiriler Kitabı: 354–359). IEEE.
- Haj-Hassan, A., Imine, Y., Gallais, A., & Quoitin, B. (2024). Detecting malicious proxy nodes during iot network joining phase. *Computer Networks*, 243, 110308.

- Hamidi, H. (2019). An approach to develop the smart health using internet of things and authentication based on biometric technology. *Future generation computer systems*, 91, 434–449.
- Hauweele, D., Koutsiamanis, R.-A., Quoitin, B., & Papadopoulos, G. Z. (2020). Pushing 6tisch minimal scheduling function (msf) to the limits. *2020 IEEE Symposium on Computers and Communications (ISCC)*, (Bildiriler Kitabı: 1–7). IEEE.
- Hofer-Schmitz, K., & Stojanović, B. (2020). Towards formal verification of iot protocols: A review. *Computer Networks*, 174, 107233.
- Höglund, J., Furuheid, M., & Raza, S. (2023). Lightweight certificate revocation for low-power iot with end-to-end security. *Journal of Information Security and Applications*, 73, 103424.
- Hu, F., Jiang, M., Wagner, M., & Dong, D.-C. (2007). Privacy-preserving telecardiology sensor networks: Toward a low-cost portable wireless hardware/software codesign. *IEEE Transactions on Information Technology in Biomedicine*, 11(6), 619–627.
- Huang, W. (2024). Ecc-based three-factor authentication and key agreement scheme for wireless sensor networks. *Scientific Reports*, 14(1), 1787.
- Instruments, T. (2024). Msp-exp430f5438 experimenter board user's guide, 2013. Available: <http://www.ti.com/lit/ug/slau263i/slau263i.pdf>.
- Ito, S., Khan, A. A., Ahmad, M., & Idrisi, M. J. (2023). A secure and privacy-preserving lightweight authentication and key exchange algorithm for smart agriculture monitoring system. *IEEE Access*.
- Jain, U., Tripathi, A., Kumar, S., & Kumar, G. (2023). Simple, secure and lightweight authentication protocol with session-key generation for iiot device in iiot networks. *Microsystem Technologies*, (Bildiriler Kitabı: 1–13).
- Kalita, A., & Khatua, M. (2022). 6tisch–ipv6 enabled open stack iot network formation: A review. *ACM Transactions on Internet of Things*, 3(3), 1–36.
- Kang, T., Woo, N., & Ryu, J. (2024). Enhanced lightweight medical sensor networks authentication scheme based on blockchain. *IEEE Access*.
- Kevin, A. (2009). That'internet of things' thing. *RFID journal*, 22.
- Khalil, K., Mohaidat, T., Darwich, M., Kumar, A., & Bayoumi, M. (2024). An efficient hardware design of coap protocol for the internet of things. *2024 IEEE 17th Dallas Circuits and Systems Conference (DCAS)*, (Bildiriler Kitabı: 1–5). IEEE.
- Khan, M. A., & Salah, K. (2018). Iot security: Review, blockchain solutions, and open challenges. *Future generation computer systems*, 82, 395–411.
- Khashan, O. A., & Khafajah, N. M. (2023). Efficient hybrid centralized and blockchain-based authentication architecture for heterogeneous iot systems. *Journal of King Saud University-Computer and Information Sciences*, 35(2), 726–739.
- Külcü, S., & Görmüş, S. (2022). Improving synchronization time in 6tisch networks with smart antennas. *2022 30th Signal Processing and Communications Applications Conference (SIU)*, (Bildiriler Kitabı: 1–4).

- Ko, J., Terzis, A., Dawson-Haggerty, S., Culler, D. E., Hui, J. W., & Levis, P. (2011). Connecting low-power and lossy networks to the internet. *IEEE Communications Magazine*, 49(4), 96–101.
- Krech, F.-A., Rota, A.-I., & Stangaciu, C.-S. (2023). An overview of different topologies for coap protocol using contiki operating system. *2023 27th International Conference on System Theory, Control and Computing (ICSTCC)*, (Bildiriler Kitabı: 416–421). IEEE.
- Krishna, R. R., Priyadarshini, A., Jha, A. V., Appasani, B., Srinivasulu, A., & Bizon, N. (2021). State-of-the-art review on iot threats and attacks: Taxonomy, challenges and solutions. *Sustainability*, 13(16), 9463.
- Külcü, S., & Görmüş, S. (2022). Improving synchronization time in 6tisch networks with smart antennas. *2022 30th Signal Processing and Communications Applications Conference (SIU)*, (Bildiriler Kitabı: 1–4). IEEE.
- Kulcu, S., Gormus, S., & Jin, Y. (2021). Integration of steerable smart antennas to ietf 6tisch protocol for high reliability wireless iot networks. *IEEE Access*, 9, 147780–147790.
- Kumar, K., & Kolberg, M. (2023). Dessr: a decentralized, broadcast-based scalable scheduling reservation protocol for 6tisch networks. *IEEE Internet of Things Journal*.
- Kurunathan, H., Severino, R., Koubaa, A., & Tovar, E. (2018). Ieee 802.15. 4e in a nutshell: Survey and performance evaluation. *IEEE Communications Surveys & Tutorials*, 20(3), 1989–2010.
- Kushalnagar, N., Montenegro, G., & Schumacher, C. (2007). Rfc 4919: Ipv6 over low-power wireless personal area networks (6lowpans): overview, assumptions, problem statement, and goals.
- Le, A., Loo, J., Chai, K. K., & Aiash, M. (2016). A specification-based ids for detecting attacks on rpl-based network topology. *Information*, 7(2), 25.
- Lee, J., Oh, J., & Park, Y. (2023). A secure and anonymous authentication protocol based on three-factor wireless medical sensor networks. *Electronics*, 12(6), 1368.
- Lu, Y., Xu, G., Li, L., & Yang, Y. (2019). Anonymous three-factor authenticated key agreement for wireless sensor networks. *Wireless Networks*, 25, 1461–1475.
- Madakam, S., Ramaswamy, R., & Tripathi, S. (2015). Internet of things (iot): A literature review. *Journal of Computer and Communications*, 3(5), 164–173.
- Majumder, S., Ray, S., Sadhukhan, D., Khan, M. K., & Dasgupta, M. (2021). Ecc-coap: Elliptic curve cryptography based constraint application protocol for internet of things. *Wireless Personal Communications*, 116(3), 1867–1896.
- Montenegro, G., Hui, J., Culler, D., & Kushalnagar, N. (2007). Transmission of IPv6 Packets over IEEE 802.15.4 Networks. RFC 4944.
URL <https://www.rfc-editor.org/info/rfc4944>
- Morabito, R., & Jiménez, J. (2020). Ietf protocol suite for the internet of things: Overview and recent advancements. *IEEE Communications Standards Magazine*, 4(2), 41–49.

- Muhal, M. A., Luo, X., Mahmood, Z., & Ullah, A. (2018). Physical unclonable function based authentication scheme for smart devices in internet of things. *2018 IEEE International Conference on Smart Internet of Things (SmartIoT)*, (Bildiriler Kitabı: 160–165). IEEE.
- Mukhopadhyay, D. (2016). Pufs as promising tools for security in internet of things. *IEEE Design & Test*, 33(3), 103–115.
- Nandhini, P., Kuppuswami, S., & Malliga, S. (2023). Energy efficient thwarting rank attack from rpl based iot networks: a review. *Materials Today: Proceedings*, 81, 694–699.
- Oliveira, L. M. L., Rodrigues, J. J., de Sousa, A. F., & Denisov, V. M. (2016). Network admission control solution for 6lowpan networks based on symmetric key mechanisms. *IEEE transactions on industrial informatics*, 12(6), 2186–2195.
- Oliver, S. G., & Purusothaman, T. (2022). Lightweight and secure mutual authentication scheme for iot devices using coap protocol. *Computer Systems Science & Engineering*, 41(2).
- Olsson, J. (2014). 6lowpan demystified. *Texas Instruments*, 13, 1–13.
- Österlind, F. (2006). *A sensor network simulator for the Contiki OS*. Swedish Institute of Computer Science.
- Osterlind, F., Dunkels, A., Eriksson, J., Finne, N., & Voigt, T. (2006). Cross-level sensor network simulation with cooja. *Proceedings. 31st IEEE conference on local computer networks*, (Bildiriler Kitabı: 641–648).
- Pradhan, N., & Chaudhari, B. S. (2022). 6tisch scheduling for iiot by slotframe fragmentation. *2022 IEEE 2nd International Conference on Mobile Networks and Wireless Communications (ICMNWC)*, (Bildiriler Kitabı: 1–5).
- Rao, V., & Prema, K. (2021). A review on lightweight cryptography for internet-of-things based applications. *Journal of Ambient Intelligence and Humanized Computing*, 12(9), 8835–8857.
- Raza, S., Duquennoy, S., Höglund, J., Roedig, U., & Voigt, T. (2014). Secure communication for the internet of things—a comparison of link-layer security and ipsec for 6lowpan. *Security and Communication Networks*, 7(12), 2654–2668.
- Raza, S., & Magnússon, R. M. (2018). Tinyike: Lightweight ikev2 for internet of things. *IEEE Internet of Things Journal*, 6(1), 856–866.
- Righetti, F., Vallati, C., Das, S. K., & Anastasi, G. (2020). Analysis of distributed and autonomous scheduling functions for 6tisch networks. *IEEE Access*, 8, 158243–158262.
- Sai, K. S., Bhat, R., Hegde, M., & Andrew, J. (2023). A lightweight authentication framework for fault-tolerant distributed wsn. *IEEE Access*, 11, 83364–83376.
- Saini, K. K., Kaur, D., Kumar, D., & Kumar, B. (2024). An efficient three-factor authentication protocol for wireless healthcare sensor networks. *Multimedia Tools and Applications*, (Bildiriler Kitabı: 1573–7721).
URL <https://doi.org/10.1007/s11042-024-18114-1>

- Sarb, A., Brad, S., Stan, O., & Plugaru, S. (2015). Management innovation in designing buildings: from sustainable and intelligent buildings to inclusive buildings. a survey. *Acta Technica Napocensis: Civil Engineering & Architecture*, 58(1).
- Sarikaya, B., Ohba, Y., Moskowitz, R., Cao, Z., & Cragie, R. (2012). Security Bootstrapping Solution for Resource-Constrained Devices. Internet-Draft draft-sarikaya-core-sbootstrapping-05, Internet Engineering Task Force. Work in Progress.
URL <https://datatracker.ietf.org/doc/draft-sarikaya-core-sbootstrapping/05/>
- Sengupta, J., Ruj, S., & Bit, S. D. (2020). A comprehensive survey on attacks, security issues and blockchain solutions for iot and iiot. *Journal of network and computer applications*, 149, 102481.
- Sheikh, M. S., Liang, J., & Wang, W. (2019). A survey of security services, attacks, and applications for vehicular ad hoc networks (vanets). *Sensors*, 19(16).
URL <https://www.mdpi.com/1424-8220/19/16/3589>
- Shetty, S. P., Shetty, M., Kishore, V., & Shetty, P. (2024). Trickle timer modification for rpl in internet of things. *Soft Computing*, 28(3), 2621–2635.
- Simha, S. V., Mathew, R., Sahoo, S., & Biradar, R. C. (2020). A review of rpl protocol using contiki operating system. *2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)*(48184), (Bildiriler Kitabı: 259–264). IEEE.
- Siow, E., Tiropanis, T., & Hall, W. (2018). Analytics for the internet of things: A survey. *ACM computing surveys (CSUR)*, 51(4), 1–36.
- Szymoniak, S. (2024). Key distribution and authentication protocols in wireless sensor networks: A survey. *ACM Computing Surveys*, 56(6), 1–31.
- Tanaka, Y., Minet, P., Vučinić, M., Vilajosana, X., & Watteyne, T. (2021). Ysf: A 6tisch scheduling function minimizing latency of data gathering in iiot. *IEEE Internet of Things Journal*, 9(11), 8607–8615.
- Tanaka, Y., Minet, P., & Watteyne, T. (2019). 6lowpan fragment forwarding. *IEEE Communications Standards Magazine*, 3(1), 35–39.
- Thakur, G., Prajapat, S., Kumar, P., & Chen, C.-M. (2024). A privacy-preserving three-factor authentication system for iot-enabled wireless sensor networks. *Journal of Systems Architecture*, 154, 103245.
URL <https://www.sciencedirect.com/science/article/pii/S1383762124001826>
- Thubert, P. (2019). An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4. Internet-Draft draft-ietf-6tisch-architecture-21, Internet Engineering Task Force. Work in Progress.
URL <https://datatracker.ietf.org/doc/draft-ietf-6tisch-architecture/21/>
- Turkanović, M., Brumen, B., & Hölbl, M. (2014). A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the internet of things notion. *Ad Hoc Networks*, 20, 96–112.
- Urke, A. R., Kure, i., & Øvsthus, K. (2022). A survey of 802.15.4 tsch schedulers for a standardized industrial internet of things. *Sensors*, 22(1).

- Vandervelden, T., Deac, D., Van Glabbeek, R., De Smet, R., Braeken, A., & Steenhaut, K. (2024). Evaluation of 6lowpan generic header compression in the context of a rpl network. *Sensors*, 24(1).
URL <https://www.mdpi.com/1424-8220/24/1/73>
- Vasseur, J., Agarwal, N., Hui, J., Shelby, Z., Bertrand, P., & Chauvenet, C. (2011). Rpl: The ip routing protocol designed for low power and lossy networks. *Internet Protocol for Smart Objects (IPSO) Alliance*, 36, 1–20.
- Vasseur, J.-P., & Dunkels, A. (2010). *Interconnecting Smart Objects with IP: The Next Internet*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
- Velásquez, I., Caro, A., & Rodríguez, A. (2018). Authentication schemes and methods: A systematic literature review. *Information and Software Technology*, 94, 30–37.
- Vijaykumar, V. R., Sekar, S. R., Jothin, R., Diniesh, V. C., Elango, S., & Ramakrishnan, S. (2024). Novel light weight hardware authentication protocol for resource constrained iot based devices. *IEEE Journal of Radio Frequency Identification*, 8, 31–42.
- Vilajosana, X., Pister, K., & Watteyne, T. (2017). Minimal ipv6 over the tsch mode of ieee 802.15. 4e (6tisch) configuration. Teknik rapor.
- Vučinić, M., Chang, T., Škrbić, B., Kočan, E., Pejanović-Djurišić, M., & Watteyne, T. (2020). Key performance indicators of the reference 6tisch implementation in internet-of-things scenarios. *IEEE Access*, 8, 79147–79157.
- Vučinić, M., Simon, J., Pister, K. S., & Richardson, M. (2019). Constrained join protocol (cojp) for 6tisch. *Internet Engineering Task Force*.
- Vučinić, M., Simon, J., Pister, K. S., & Richardson, M. (2021). RFC9031: Constrained Join Protocol (CoJP) for 6TiSCH. *Internet Engineering Task Force*, (RFC9031).
URL <https://inria.hal.science/hal-02407587>
- Vučinić, M., Simon, J., Pister, K., & Richardson, M. (2021). Constrained Join Protocol (CoJP) for 6TiSCH. RFC 9031.
URL <https://www.rfc-editor.org/info/rfc9031>
- Wang, Q., Vilajosana, X., & Watteyne, T. (2018). 6tisch operation sublayer (6top) protocol (6p). Teknik rapor.
- Wang, S., Huang, K., Ma, K., Xu, X., & Hu, X. (2023). A lightweight encryption and message authentication framework for wireless communication. *IET Communications*, 17(3), 265–278.
- Wu, F., Li, X., Sangaiah, A. K., Xu, L., Kumari, S., Wu, L., & Shen, J. (2018). A lightweight and robust two-factor authentication scheme for personalized healthcare systems using wireless medical sensor networks. *Future Generation Computer Systems*, 82, 727–737.
- Yang, Y., Wu, L., Yin, G., Li, L., & Zhao, H. (2017). A survey on security and privacy issues in internet-of-things. *IEEE Internet of things Journal*, 4(5), 1250–1258.
- Yogesh, P. R., vd. (2020). Formal verification of secure evidence collection protocol using ban logic and avispa. *Procedia Computer Science*, 167, 1334–1344.

- Zhu, C., Zheng, C., Shu, L., & Han, G. (2012). A survey on coverage and connectivity issues in wireless sensor networks. *Journal of Network and Computer Applications*, 35(2), 619–632.
- Zikria, Y. B., Afzal, M. K., Ishmanov, F., Kim, S. W., & Yu, H. (2018). A survey on routing protocols supported by the contiki internet of things operating system. *Future Generation Computer Systems*, 82, 200–219.



ÖZGEÇMİŞ

Hakan AYDIN, ilköğrenimini Dereli Eğriarı İlköğretim Okulu'nda bitirdikten sonra lise eğitimini 2009 yılında Giresun Aksu Anadolu Lisesi'nde ikinci olarak tamamladı. 2009 - 2010 öğretim yılında Karadeniz Teknik Üniversitesi Mühendislik Fakültesi Bilgisayar Mühendisliği bölümünde lisans eğitimine başladı. Üniversite yıllarında yazılım geliştirme, ağ güvenliği ve gömülü sistemler konularında çeşitli projelerde yer aldı. 2015 yılında lisans eğitimini tamamlayarak bilgisayar mühendisi ünvanını kazandı. Aynı yıl, Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Bilgisayar Bilimleri Anabilim Dalı'nda tezli yüksek lisans programına başladı. Yüksek lisans eğitimi boyunca, Nesnelerin İnterneti ve güvenlik protokolleri üzerinde yoğunlaştı. 2018 yılında, "IETF 6TiSCH protokolü için dağıtık kullanıcı kimlik doğrulama mekanizması" başlıklı yüksek lisans tezini başarıyla sunarak yüksek lisans derecesini aldı. Yüksek lisans eğitiminin ardından doktora eğitimine Karadeniz Teknik Üniversitesi'nde devam eden AYDIN, araştırmalarını Endüstriyel Nesnelerin İnterneti ve ağ güvenliği üzerine yoğunlaştırdı. 2016 yılından itibaren Karadeniz Teknik Üniversitesi Of Teknoloji Fakültesi Yazılım Mühendisliği Bölümü'nde araştırma görevlisi olarak görev yapmaktadır. Akademik çalışmalarının yanı sıra, çeşitli ulusal ve uluslararası konferanslarda bildiriler sunmuş, akademik dergilerde hakemlikler yapmıştır.

Hakemli konferans/sempozyumların bildiri kitaplarında yer alan yayınlar

1. S. Alizadeh and H. Aydın, "Energy Efficient Communication Protocol For 6TiSCH Networks," 2019 Seventh International Conference on Digital Information Processing and Communications (ICDIPC), Trabzon, Turkey, 2019, pp. 63-67, doi: 10.1109/ICDIPC.2019.8723860.
2. H. Aydın and I. Bozkurt, "An Energy Efficient and Lightweight Cryptographic User Authentication Scheme for 6Tisch Networks," 2020 28th Signal Processing and Communications Applications Conference (SIU), Gaziantep, Turkey, 2020, pp. 1-4, doi: 10.1109/SIU49456.2020.9302128.
3. Hakan Aydın, P Dehghan, Cavdar Tugrul, "Yazılım Tanımlı IoT Ağlarında Kimlik Doğrulama," 2020 28th Signal Processing and Communications Applications Conference (SIU), Poster Sunumu, 2020.
4. P. Dehghan, H. Aydın and S. Görmüş, "Resource Management in Distributed 6TiSCH Networks," 2020 5th International Conference on Computer Science and Engineering (UBMK), Diyarbakir, Turkey, 2020, pp. 349-354, doi: 10.1109/UBMK50275.2020.9219388.

5. B. Aydın, S. Görmüş, H. Aydın and S. Külcü, "A New Routing Objective Function for IETF 6TiSCH Protocol," 2022 30th Signal Processing and Communications Applications Conference (SIU), Safranbolu, Türkiye, 2022, pp. 1-4, doi: 10.1109/SIU55565.2022.9864711.
6. T. Bekar, S. Görmüş, B. Aydın and H. Aydın, "Q-Learning Algorithm Inspired Objective Function Optimization For IETF 6TiSCH Networks," 2023 International Conference on Smart Applications, Communications and Networking (SmartNets), Istanbul, Türkiye, 2023, pp. 1-6, doi: 10.1109/SmartNets58706.2023.10216124.
7. B. Aydın, H. Aydın and S. Görmüş, "A Security Mechanism Against Man in the Middle Attack in 6TiSCH Networks," 2024 32nd Signal Processing and Communications Applications Conference (SIU), Mersin, Türkiye, 2024, pp. 1-4, doi: 10.1109/SIU61531.2024.10600741.
8. H. Aydın and B. Aydın, "A DIS Attack Resistant Algorithm for RPL Routing Protocol in 6TiSCH Networks," 2024 32nd Signal Processing and Communications Applications Conference (SIU), Mersin, Türkiye, 2024, pp. 1-4, doi: 10.1109/SIU61531.2024.10600756.
9. B. Aydın, H. Aydın ve S. Gormus, "Advanced Detection and Prevention of Sinkhole Attacks in 6TiSCH Networks," 9th International Conference on Computer Science and Engineering (UMBK 2024), Antalya, Türkiye, 2024.

Uluslararası hakemli dergilerde yayınlanan makaleler (SCI/SCI-E)

1. Görmüş, S., Aydın, H., and Ulutaş, G. (2018). Security for the internet of things: a survey of existing mechanisms, protocols and open research issues. Journal of the Faculty of Engineering and Architecture of Gazi University, 33(4), 1247-1272.
2. Aydın, H., Gormus, S., & Aydın, B. (2024). A Decentralized Proxy-JRC Authentication System for Scalable IETF 6TiSCH Networks. IEEE Access.
3. Aydın, H., Aydın, B., & Gormus, S. (2024). Lightweight Three-Factor Sensor and User Authentication for RPL-Based 6TiSCH Networks. IEEE Sensors Journal.

Projeler

1. Akay, B., Çakır, O., Canpolat G. A., ve Aydın, H. "Hasta Takip Sistemli Glukoz Sensörü ve Modüler Acil İnsülin Rezerv- Enjeksiyon Ünitesi", TÜBİTAK - BiGG 1512, 2023 - 2024.