

T.C.
MUĞLA SITKI KOÇMAN ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLİŞİM SİSTEMLERİ MÜHENDİSLİĞİ
ANABİLİM DALI

HETEROJEN VERİ TRAFİĞİ İÇEREN NESNELERİN
İNTERNETİ SİSTEMLERİ İÇİN DİNAMİK BİR KANAL
TAHSİS YÖNTEMİNİN MARKOV ZİNCİRİ MODELİ
VE BAŞARIM ANALİZİ

FEYYAZ OKUL

YÜKSEK LİSANS TEZİ

OCAK 2025

MUĞLA

MUĞLA SITKI KOÇMAN ÜNİVERSİTESİ

Fen Bilimleri Enstitüsü

TEZ ONAYI

Feyyaz OKUL tarafından hazırlanan **Heterojen Veri Trafikçi İçeren Nesnelerin İnterneti Sistemleri için Dinamik Bir Kanal Tahsis Yönteminin Markov Zinciri Modeli ve Başarım Analizi** başlıklı tezinin, 14/01/2025 tarihinde aşağıdaki jüri tarafından Bilişim Sistemleri Mühendisliği Anabilim Dalı'nda yüksek lisans derecesi için gerekli şartları sağladığı **oybirliği** ile kabul edilmiştir.

TEZ SINAV JURİSİ

Prof. Dr. İlhan TARIMER (**Jüri Başkanı**)

İmza:

Bilişim Sistemleri Mühendisliği Anabilim Dalı,
Muğla Sıtkı Koçman Üniversitesi, Muğla

Doç. Dr. Sedat ATMACA (**Danışman**)

İmza:

Bilişim Sistemleri Mühendisliği Anabilim Dalı,
Muğla Sıtkı Koçman Üniversitesi, Muğla

Doç. Dr. Necla BANDIRMALI ERTÜRK (**Üye**)

İmza:

Bandırma Onyediy Eylül Üniversitesi
Bilgisayar Mühendisliği Bölümü

ANA BİLİM DALI BAŞKANLIĞI ONAYI

Doç. Dr. Osman ÖZKARACA (**Anabilim Dalı Başkanı**)

İmza:

Bilişim Sistemleri Mühendisliği Anabilim Dalı,
Muğla Sıtkı Koçman Üniversitesi, Muğla

Doç. Dr. Sedat ATMACA (**Danışman**)

İmza:

Bilişim Sistemleri Mühendisliği Anabilim Dalı,
Muğla Sıtkı Koçman Üniversitesi, Muğla

Savunma Tarihi: 14/01/2025

Tez çalışmalarım sırasında elde ettiğim ve sunduğum tüm sonuç, doküman, bilgi ve belgelerin tarafımdan bizzat ve bu tez çalışması kapsamında elde edildiğini; akademik ve bilimsel etik kurallarına uygun olduğunu beyan ederim. Ayrıca, akademik ve bilimsel etik kuralları gereği bu tez çalışması sırasında elde edilmemiş başkalarına ait tüm orijinal bilgi ve sonuçlara atıf yapıldığını da beyan ederim.

Feyyaz OKUL

14/01/2025



ÖNSÖZ

Bu tez çalışmasının her aşamasında bilgi birikimi, rehberliği ve desteğiyle beni yönlendiren, akademik gelişimime büyük katkılarda bulunan değerli danışman hocam Doç. Dr. Sedat ATMACA'ya en içten teşekkürlerimi sunarım. Hocamın titiz yaklaşımları, sağladığı kıymetli geri bildirimler ve akademik vizyonumu genişletmemdeki katkıları, bu çalışmanın başarıyla tamamlanmasında temel bir rol oynamıştır. Kendisi, yalnızca bir danışman değil, aynı zamanda bilimsel bir yol gösterici olarak bana ilham kaynağı olmuştur.

Ayrıca, hayatımın her anında yanımda olan, karşılaştığım tüm zorluklarda sabrı ve anlayışıyla bana güç veren sevgili eşim Seda OKUL'a gönülden teşekkür ederim. Psikolog kimliğiyle mesleğinin getirdiği içgörü ve empatiyi, benim için bir yaşam desteğine dönüştüren eşim, bu süreçte yalnızca bir eş değil, aynı zamanda en büyük ilham kaynağım ve destekçim olmuştur. Yoğun temposuna rağmen her zaman yanımda bulunarak, özellikle stresli ve yorucu anlarda gösterdiği özveriyle bu tezin tamamlanmasına sağladığı katkılar unutulmazdır.

Saygılarımla,

Feyyaz Okul

İÇİNDEKİLER

ÖNSÖZ.....	iv
İÇİNDEKİLER	v
ŞEKİLLER DİZİNİ	vii
TABLolar DİZİNİ	viii
SEMBOLLER VE KISALTMALAR DİZİNİ	ix
1. GİRİŞ	1
1.1. Tez Çalışmasının Önemi ve Amaçları.....	1
1.2. Tez Organizasyonu	2
1.3. Tez Çalışmasının Motivasyonu ve Katkıları	3
2. LİTERATÜR ARAŞTIRMASI	4
2.1. Akıllı Şehir Uygulamaları	6
2.2. Sağlık Uygulamaları.....	7
2.3. Tarım Uygulamaları	8
2.4. Ulaşım Uygulamaları	9
2.5. Enerji Uygulamaları	10
2.6. IoT Heterojen Veri Trafiklerinin Dinamik Yönetimi İlgili Literatürde Sunulan Çalışmalar.....	11
3. IoT ORTAM ERİŞİM KONTROL YÖNTEMLERİ ve VERİ TRAFİK TÜRLERİ	15
3.1 IoT Ağlarında Kullanılan Ortam Erişim Kontrol Teknikleri	15
3.1.1. Çekişme tabanlı (Contention-Based) protokoller.....	15
3.1.1.1. ALOHA.....	16
3.1.1.2. Slotted ALOHA.....	16
3.1.1.3. CSMA.....	17
3.1.2. Tahsis tabanlı (Scheduled Access) protokoller	18
3.1.2.1. FDMA (Frequency Division Multiple Access).....	18
3.1.2.2. TDMA (Time Division Multiple Access).....	19
3.1.2.3. CDMA (Code Division Multiple Access)	20
3.2. IoT Uygulamalarında Kullanılan Heterojen Veri Trafikleri ve Özellikleri.....	21
3.2.1. Veri trafik türleri	22
3.2.2. Veri hızı	22
3.2.3. Gecikme	23

3.3. Servis Kalitesi (Quality of Service, QoS) Kavramı ve Önemi.....	23
3.3.1. Bant genişliği	24
3.3.2. Gecikme ve gecikme değişimi (jitter).....	25
3.3.3. Veri kaybı yönetimi	26
4. GELİŞTİRİLEN DİNAMİK KANAL TAHSİS YÖNTEMİ	27
4.1. Önerilen Kanal Tahsis Yönteminin Analitik Modeli	30
4.2. Önerilen Dinamik Kanal Tahsis Yönteminin Benzetimi.....	36
4.3. Bulgular	37
5. GELİŞTİRİLEN DİNAMİK KANAL TAHSİS YÖNTEMİNİN ÖRNEK BİR IoT AĞ SENARYOSUNDA UYGULANMASI VE BAŞARIM DEĞERLENDİRMESİ.....	38
5.1. Kanal kullanım oranı (Utilization)	38
5.2. Çağrı tıkanma (bloke) olasılığı (CBP).....	40
5.3. Çağrı tamamlama oranı (CCR).....	41
5.4. Throughput (iş çıkarma oranı).....	43
5.5. Değerlendirme	45
6. SONUÇLAR VE ÖNERİLER	47
6.1. Gelecek Çalışma Önerileri	48
KAYNAKLAR	49
ÖZGEÇMİŞ.....	53

ŞEKİLLER DİZİNİ

Şekil 2.1. IoT ile akıllı şehir uygulamaları.....	6
Şekil 2.2. IoT ile sağlık uygulamaları	8
Şekil 2.3. IoT ile tarım uygulamaları	9
Şekil 2.4. IoT ile ulaşım uygulamaları	10
Şekil 2.5. IoT ile enerji uygulamaları	11
Şekil 3.1. FDMA tekniğinde kanal kullanımı	19
Şekil 3.2. TDMA tekniğinde kanal yapısı.....	20
Şekil 3.3. CDMA kanal yapısı	21
Şekil 4.1. Önerilen kanal tahsis yönteminin kullanılğı ağ modeli	28
Şekil 4.2. Kanal tahsis yönteminin pseudo kodu	29
Şekil 4.3. Kanal tahsis yöntemi uygulama kodu giriş değerleri.....	30
Şekil 4.4. Önerilen dinamik kanal tahsis yönteminin $N_{irt}=1$, $N_{rt}=3$ ve $N=12$ için 2- boyutlu Markov Zinciri modeli	32
Şekil 5.1. Önerilen dinamik kanal tahsis algoritmasının kullanıldığı ağ modelinde kanal kullanımı oranları.....	39
Şekil 5.2. Önerilen dinamik kanal tahsis algoritmasının kullanıldığı ağ modelinde çağrı bloke olasılıkları	41
Şekil 5.3. Önerilen dinamik kanal tahsis algoritmasının kullanıldığı ağ modelinde çağrı tamamlama oranları	43
Şekil 5.4. Önerilen dinamik kanal tahsis algoritmasının kullanıldığı ağ modelinde iş çıkarma oranları.....	44

TABLolar DİZİNİ

Tablo 1. Önerilen dinamik kanal tahsis tablosunun Markov zinciri modelinde kullanılan semboller ve kısaltmalar	32
Tablo 2. Önerilen Kanal Tahsis Yöntemin Benzetim Modelinde Kullanılan Parametreler	36



SEMBOLLER VE KISALTMALAR DİZİNİ

IoT	Internet of Things (Nesnelerin İnterneti)
GZ	Gerçek zamanlı
GZO	Gerçek zamanlı olmayan
$P(i)$	i durumundaki olasılık
N	Ağda kullanılan toplam kanal sayısı
N_{NRT}	Ağdaki gerçek zamanlı olmayan veri trafiği kullanıcı sayısı
N_{RT}	Ağdaki gerçek zamanlı veri trafiği kullanıcı sayısı
N_{C_NRT}	Ağda kullanılan gerçek zamanlı olmayan veri trafiği kanal sayısı
N_{C_RT}	Ağda kullanılan gerçek zamanlı veri trafiği kanal sayısı
λ_{NRT}	Gerçek zamanlı olmayan kullanıcılara ait varış oranı
λ_{RT}	Gerçek zamanlı kullanıcılara ait varış oranı
λ_T	Ağdaki kullanıcılara ait toplam varış oranı
μ_{NRT}	Gerçek zamanlı olmayan kullanıcılara ait servis oranı
μ_{RT}	Gerçek zamanlı kullanıcılara ait servis oranı

ÖZET

HETEROJEN VERİ TRAFİĞİ İÇEREN NESNELERİN İNTERNETİ SİSTEMLERİ İÇİN DİNAMİK BİR KANAL TAHSİS YÖNTEMİNİN MARKOV ZİNCİRİ MODELİ VE BAŞARIM ANALİZİ

Feyyaz OKUL

Yüksek Lisans Tezi

Fen Bilimleri Enstitüsü

Bilişim Sistemleri Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Sedat ATMACA

Ocak 2025, 53 Sayfa

Bu tez çalışmasında, heterojen veri trafiği içeren IoT (Nesnelerin İnterneti) sistemleri için önerilen bir dinamik kanal tahsis yönteminin Markov zinciri modeli ve örnek bir ağ senaryosunda başarımların analizi gerçekleştirilmiştir. Geliştirilen dinamik kanal tahsis yöntemi ile hem gerçek-zamanlı hem de gerçek-zamanlı olmayan farklı veri trafik türlerinin IoT ağı üzerinden başarılı bir şekilde iletimi gerçekleştirilmiştir. Tez çalışmasında önerilen dinamik kanal tahsis yöntemi, 2-boyutlu sürekli Markov zinciri kullanılarak analitik olarak modellenmiş ve sonlu sayıda IoT düğümünün kullanıldığı örnek bir IoT ağ senaryosunda başarımların değerlendirilmesi gerçekleştirilmiştir. Ayrıca, önerilen modelin ayrık olay benzetim (discrete event simulation) modeli de geliştirilmiştir. Geliştirilen kanal tahsis yönteminin analitik model sonuçları benzetim modeli sonuçları ile doğrulanmıştır. Sonuç olarak; kanal sayısı 12, gerçek-zamanlı ve gerçek zamanlı-olmayan kullanıcı sayılarının 10 olduğu bir ağ senaryosunda ve toplam yük 10 iken iş çıkarma oranları, gerçek-zamanlı olmayan kullanıcılar için 3,63 ve gerçek-zamanlı kullanıcılar için 2,55 olarak elde edilmiştir. Ayrıca, aynı yük altında kanal kullanım oranları ise gerçek-zamanlı olmayan kullanıcılar için %15,16 ve gerçek-zamanlı kullanıcılar için ise %53,31 olarak elde edilmiştir. Böylece, geliştirilen dinamik kanal tahsis yönteminin IoT uygulamalarında, özellikle gerçek-zamanlı heterojen veri trafiklerinin etkin bir şekilde iletiminde önemli bir başarımlar sağladığı gösterilmiştir.

Anahtar Kelimeler: Nesnelerin İnterneti, Algılayıcı Ağlar, Ayrık Olay Benzetimi, Sürekli Markov Zinciri

ABSTRACT

MARKOV CHAIN MODEL AND PERFORMANCE ANALYSIS OF A DYNAMIC CHANNEL ALLOCATION METHOD FOR INTERNET OF THINGS SYSTEMS WITH HETEROGENEOUS DATA TRAFFIC

Feyyaz OKUL

Master Thesis

Graduate School of Engineering and Natural Sciences

Information System Engineering Department

Supervisor: Associate Professor Sedat ATMACA

January 2025, 53 Pages

In this thesis work, a Markov chain model of a dynamic channel allocation method proposed for IoT (Internet of Things) systems with heterogeneous data traffic and its performance analysis in a sample network scenario were performed. With the developed dynamic channel allocation method, both real-time and non-real-time data traffic types were successfully transmitted over the IoT network. In the thesis, the proposed dynamic channel allocation method was analytically modeled by using a 2-dimensional continuous Markov chain and its performance evaluation was performed in a sample IoT network scenario where a finite number of IoT nodes were used. In addition, the discrete event simulation model of the proposed model was also developed. The analytical model results of the developed channel allocation method were verified with the simulation model results. As a result; in a network scenario where the number of channels is 12, the number of real-time and non-real-time users is 10 and the total load is 10, the throughput rates were obtained as 3.63 for non-real-time users and 2.55 for real-time users. In addition, under the same load, the channel utilization rates are 15.16% for non-real-time users and 53.31% for real-time users. Thus, it has been shown that the developed dynamic channel allocation method provides significant performance in IoT applications, especially in the efficient transmission of real-time heterogeneous data traffic.

Keywords: Internet of Things (IoT), Sensor Networks, Medium Access Control (MAC), Discrete Event Simulation, Continuous Markov Chain

1. GİRİŞ

IoT (Internet of Things, Nesnelerin İnterneti), fiziksel cihazların ve algılayıcıların ağ üzerinden veri paylaşmasına olanak sağlayan bir teknolojidir. Günümüzde IoT'nin hızla yaygınlaşmasıyla birlikte, veri iletiminin etkinliğini ve güvenilirliğini artırmaya yönelik çeşitli yaklaşımlar geliştirilmektedir. Bu bağlamda, IoT ağlarında karşılaşılan en önemli sorunlardan biri, heterojen veri trafikleri olarak adlandırılan, farklı tiplerdeki verilerin aynı ağ üzerinden iletilmesidir (Atzori, Iera, & Morabito, 2010). Heterojen veri trafikleri, ağ üzerinde farklı servis kalitesi (QoS) gereksinimlerine sahip çeşitli uygulamaların veri paylaşımını mümkün kılmak için kritik bir unsurdur.

IoT sistemlerinde veri iletim başarımı, düğümlerin yoğunluğu, kanal kapasitesi ve veri kaynaklarının çeşitliliği gibi birçok faktörden etkilenmektedir. Bu nedenle, IoT ağlarında veri iletim süreçlerinin optimize edilmesi hem başarımları iyileştirmesi hem de düğüm enerji verimliliği açısından önem arz etmektedir. Ancak mevcut yaklaşımlar, özellikle yoğun ağ ortamlarında, bu sorunların çözümünde yeterli olmamaktadır (Gubbi, Buyya, Marusic, & Palaniswami, 2013). Dolayısıyla, IoT ağlarında daha etkin ve adil bir veri iletimini sağlamak için yenilikçi çözümlerin geliştirilmesine ihtiyaç duyulmaktadır.

Bu tez çalışmasında, IoT uygulamalarında heterojen veri trafiklerinin etkili bir şekilde iletimini sağlamak amacıyla dinamik bir kanal tahsis yöntemi geliştirilmiş ve örnek bir ağ senaryosunda başarımları değerlendirilmiştir. Önerilen kanal tahsis yöntemi, iki-boyutlu sürekli Markov Zinciri kullanarak analitik olarak modellenmiştir. Ayrıca önerilen yöntemin ayrık-zamanlı olay modellemesi ile benzetim modeli geliştirilmiştir ve benzetim modelinden elde edilen sonuçlar analitik modelden elde edilen sonuçlar ile doğrulanmıştır.

1.1. Tez Çalışmasının Önemi ve Amaçları

IoT ağlarının giderek yaygınlaşması ve gelişmesiyle birlikte daha karmaşık hale gelmesi, ağ içindeki düğümlerin farklı veri trafik türleri kullanarak iletişim kurmasını zorlaştırmaktadır. Heterojen veri trafikleri, farklı veri türlerinin aynı ağ üzerinden etkin

bir şekilde iletilmesi için uygun mekanizmaların geliştirilmesini zorunlu kılmaktadır. Mevcut literatürde, bu sorunun çözümüne yönelik çeşitli yöntemler geliştirilmiş olsa da özellikle gerçek ağ koşullarını yansıtan çözümlerin sınırlı olduğu görülmektedir (Niu et al., 2015).

Dinamik kanal tahsisi, IoT ağlarında hem etkin hem de adil veri iletimi sağlamak için öne çıkan bir yaklaşım olarak dikkat çekmektedir. Bu yöntem, ağın mevcut durumunu göz önünde bulundurarak kanalların yeniden düzenlenmesini sağlar. Böylece sınırlı ağ kaynaklarının verimli kullanılması mümkün olurken, veri iletimindeki gecikmeler de minimize edilebilmektedir. Özellikle heterojen veri trafikleri için adil bir iletim ortamı oluşturulması, dinamik kanal tahsisinin temel hedefleri arasında yer almaktadır (Gubbi, Buyya, Marusic, & Palaniswami, 2013).

Sunulmakta olan bu tez çalışmasının en önemli amaçları, sonlu sayıda düğümden oluşan bir IoT ağ uygulama ortamında, heterojen veri trafikleri kullanan düğümlere etkin ve adil bir şekilde iletim kanalı tahsisi gerçekleştiren dinamik bir kanal tahsis algoritması geliştirmek ve geliştirilen algoritmanın örnek bir ağ senaryosunda başarımlarını analizini elde etmektedir. Böylece, hem ağ kaynaklarının verimli bir şekilde kullanımı hem de farklı servis kalitesi gereksinimlerinin karşılanması hedeflenmektedir.

1.2. Tez Organizasyonu

Bu tez, IoT ağlarında heterojen veri trafiklerinin etkin iletimi ve ağın başarımı konularını ele alan altı ana bölümden oluşmaktadır. Birinci bölüm, çalışmanın amaçlarını, önemini ve motivasyonunu açıklamaktadır. Bu bölümde ayrıca, heterojen veri trafiklerinin IoT ağları üzerindeki olumsuz etkileri ve çözümüne yönelik ihtiyaçlar vurgulanmaktadır. İkinci bölümde, IoT'nin temel bileşenleri ve geniş uygulama alanları ele alınmakta ve üçüncü bölümde heterojen veri trafiklerinin karakteristik özellikleri ile servis kalitesi üzerindeki etkileri detaylandırılmaktadır. Dördüncü bölümde, önerilen kanal tahsis yöntemi, analitik ve benzetim modelleri ile birlikte ayrıntılı olarak sunulmaktadır. Beşinci bölümde, önerilen kanal tahsis yönteminin örnek bir ağ senaryosu üzerinde gerçekleştirilen başarımlarına ilişkin değerlendirmelere yer verilmektedir. Bu bölümde ayrıca ayrık olay benzetimi gerçekleştirilen kanal tahsis yönteminin benzetim sonuçları ile analitik modelden elde

edilen sonuçlar doğrulanmaktadır. Altıncı bölümde, araştırmanın ve elde edilen sonuçların genel bir değerlendirmesi yapılarak gelecekteki çalışmalar için öneriler sunulmaktadır.

1.3. Tez Çalışmasının Motivasyonu ve Katkıları

IoT ağlarının giderek daha yoğun bir şekilde kullanılması, veri iletim süreçlerini karmaşık hale getirmektedir. Bu durum hem veri türleri arasındaki çeşitlilik hem de servis kalitesi gereksinimlerinin farklılığına bağlı olarak mevcut yaklaşımların yetersiz kalmasına neden olmaktadır. Özellikle, heterojen veri trafiklerinin yönetimindeki sorunlar, IoT ağlarının başarımını olumsuz etkilemektedir (Gubbi et al., 2013).

Bu tez çalışmasının motivasyonu, IoT ağlarında son zamanlarda sıklıkla karşılaşılan heterojen veri trafiklerinin iletim sorunlarına bir çözüm sunmaktır. Çalışmada, heterojen veri trafiklerini dinamik bir şekilde yöneterek, IoT ağlarında servis kalitesi destekli ve sürdürülebilir bir veri iletişim altyapısı oluşturulması hedeflenmektedir. Bu yaklaşım, farklı servis kalitesi gereksinimlerine sahip düğümlerin, aynı IoT ağı üzerinden verilerini etkili bir şekilde iletebilmesini mümkün hale getirecektir.

Gerçekleştirilen tez çalışmalarının IoT'nin gelecekteki uygulamalarına ve ağ başarım değerlendirmelerine önemli katkılar sağlaması beklenmektedir. Tezde önerilen yeni dinamik kanal tahsis yöntemi, IoT ağlarının geniş ölçekli kullanımında önemli bir temel oluşturacak ve bu ağların başarımını artırmaya yönelik yeni yaklaşımların önünü açacaktır.

2. LİTERATÜR ARAŞTIRMASI

IoT, düğüm adı verilen fiziksel cihazlar ve dijital sistemler arasında sürekli veri alışverişi sağlayan bir teknolojiyi temsil etmektedir. IoT'nin temel işlevi, farklı cihazların internet üzerinden bağlanmasını ve bu cihazlardan elde edilen verilerin analiz edilerek çeşitli amaçlar doğrultusunda kullanılmasını mümkün kılmaktır (Xia, Yang, Wang, & Vinel, 2012). Bu teknoloji, sağladığı faydalar nedeniyle tarımdan sağlık hizmetlerine kadar birçok sektörde önemli bir yere sahiptir.

IoT uygulamalarında verimlilik, düğüm enerji tüketimi, gecikme ve güvenilirlik gibi başarımlarını iyileştirmek için ağına farklı katmanlarında değişik protokoller ve yenilikçi yaklaşımlar geliştirilmiştir. Bu bölümde, IoT sistemlerinin temel bileşenleri ve mimarileri incelenmekte, ardından IoT'nin çeşitli uygulama alanları detaylandırılmaktadır. Ayrıca, literatürde sunulan IoT ağlarındaki farklı kanal tahsis teknikleri detaylı bir şekilde incelenerek güçlü ve zayıf yönleri ile açıklanmıştır. Literatürde IoT'nin kullanımına yönelik sunulan çalışmalar ve bu alandaki gelişmeler, IoT teknolojisinin özellikle heterojen veri trafiklerinin etkin yönetimindeki rolünü anlamak açısından ele alınmaktadır.

IoT sistemleri, her biri farklı görevlere sahip dört temel bileşenden oluşmaktadır: düğümler (cihazlar), ağlar, veri işleme ve depolama sistemleri ve uygulama bileşenleri (Bandyopadhyay & Sen, 2011). Fiziksel ortamdan veri toplama işlevi, algılayıcılar ve aktüatörler gibi cihazlar tarafından gerçekleştirilirken, bu verilerin iletimi ağlar üzerinden sağlanmaktadır. İletilen veriler genellikle merkezi sistemlerde işlenerek anlamlı bilgiye dönüştürülmekte ve farklı uygulamalarda kullanılmaktadır (Wang, Xu, & Bi, 2018).

IoT ağ mimarileri; algılayıcı katmanı, ortam erişim kontrol katmanı, ağ katmanı ve uygulama katmanı olmak üzere dört temel yapı üzerinden şekillenmektedir (Zanella et al., 2014). Algılayıcı katman, algılayıcılar aracılığıyla fiziksel verilerin toplanması ve dijital verilere dönüştürülmesini sağlamaktadır. Ortam erişim kontrol katmanı, ortak kanalın düğümler arasında paylaşılmasından sorumludur. Ağ katmanı, verilerin cihazlar arasında taşınması işlemini kablolu ve kablosuz teknolojiler kullanılarak

gerçekleştirir. Uygulama katmanı ise IoT sistemlerinin asıl işlevsel değerini ortaya çıkaran ve verilerin anlamlandırılarak analiz edildiği katmandır. IoT ağlarında genellikle yukarıda sıralanan katmanlar kullanılmasına rağmen, bazı örneklerde bu katmanlardan iki veya daha fazlasının birlikte kullanıldığı hibrit yapılar da mevcuttur.

Son yıllarda IoT mimarileri üzerinde yapılan çalışmalar, geleneksel merkezi sistemlerin yerine daha esnek ve verimli çözümleri gündeme getirmiştir. Sis bilişim, uç bilişim ve dağıtık bulut sistemleri, veri işleme süreçlerini cihazlara daha yakın noktalarda gerçekleştirmeyi mümkün kılmakta ve bu sayede gecikme süreleri azaltılmaktadır (Chiang & Zhang, 2016). Ayrıca bu yaklaşımlar, verilerin daha güvenli bir şekilde işlenmesine olanak tanımaktadır (Satyanarayanan, 2017).

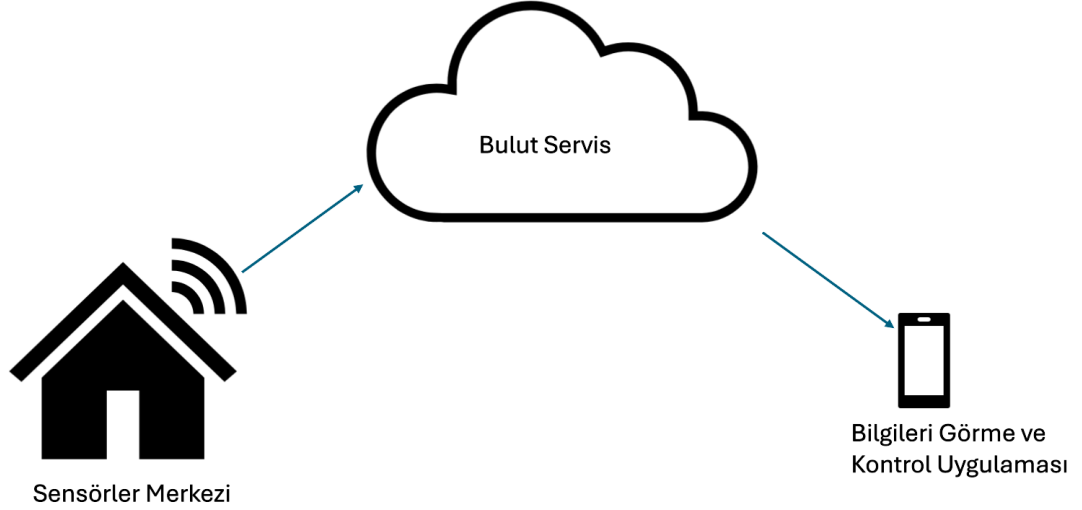
IoT teknolojisi, üretimden sağlık hizmetlerine, tarımdan enerji sektörüne, akıllı evlerden akıllı şehirlere kadar geniş bir yelpazede uygulanmaktadır. IoT'nin bu alanlardaki katkıları, verimliliği artırmak, maliyetleri düşürmek ve kullanıcı deneyimlerini geliştirmek gibi çok yönlü faydalar sunmaktadır (Atzori, Iera, & Morabito, 2010).

Farklı sektörlerde güncel IoT uygulamaları, büyük miktarda veri toplama, bu verileri analiz etme ve karar alma süreçlerini optimize etme yeteneği ile öne çıkmaktadır. Özellikle algılayıcılar, aktüatörler ve bulut bilişim altyapısı, IoT'nin temel unsurlarını oluşturur. Örneğin, akıllı şehirlerde trafik akışını optimize etmekten, sağlık sektöründe riskli hastaların uzaktan gerçek zamanlı (real-time) takibini sağlamaya kadar birçok kritik kullanım alanı vardır. Bu bağlamda, IoT yalnızca bireylerin yaşam kalitesini artırmakla kalmaz, aynı zamanda endüstriyel verimlilik ile iş süreçlerini daha sürdürülebilir ve çevre dostu hale getirir (Gubbi et al., 2013).

IoT'nin başarısında en önemli faktörlerden biri, cihazlar arasındaki bağlantının güvenilirliği ve ölçeklenebilirliğidir. IoT ağları, heterojen veri trafiklerini yönetme yeteneği sayesinde çok çeşitli cihaz ve uygulamaları destekleyebilmektedir. Ayrıca, bu ağlar, gerçek zamanlı veri işleme ve analiz imkânı sunarak hızlı ve doğru kararlar alınmasına katkı sağlanmaktadır. Bu durum, işletmelerin ve kamusal kurumların IoT uygulamalarını benimsemelerini hızlandırmıştır. Örneğin, enerji sektöründe akıllı sayaçlar ve enerji yönetim sistemleri hem enerji tasarrufu sağlamakta hem de karbon ayak izini azaltmaktadır (Perera et al., 2014). Alt bölümlerde IoT ağlarının yaygın kullanıldığı uygulamalar detaylıca açıklanmaktadır.

2.1. Akıllı Şehir Uygulamaları

Akıllı şehir uygulamaları, IoT teknolojisinin kentsel altyapıların ve hizmetlerin yönetiminde nasıl bir dönüşüm sağladığını ortaya koymaktadır. Trafik yönetimi, enerji optimizasyonu ve atık yönetimi gibi alanlarda IoT cihazları ve ağlar aracılığıyla toplanan veriler analiz edilerek etkin çözümler sunulmaktadır (Huang, Wang, & Xu, 2015). Bu sistemler hem kaynakların daha verimli kullanılmasını hem de çevresel sürdürülebilirliğin desteklenmesini sağlamaktadır. Şekil 2.1’de akıllı şehirlerde IoT uygulaması görülmektedir. Bu uygulama ile şehir altyapılarının modernize edilerek algılayıcılardan alınan verilere göre, trafik sıklığı azaltılabilmektedir (Solanas et al., 2014). Ayrıca, güvenlik hizmetlerinde IoT tabanlı kameralar ve algılayıcılarla suç oranları düşürülürken, afet yönetim süreçleri de geliştirilmiştir (Zanella et al., 2014).



Şekil 2.1. IoT ile akıllı şehir uygulamaları

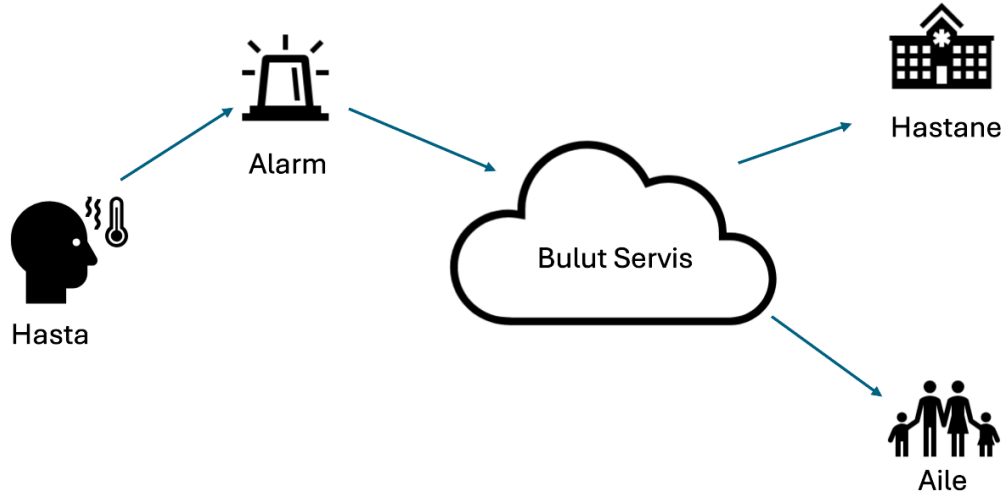
Akıllı şehir projelerinde, gerçek zamanlı veri analitiği ve IoT cihazlarının sağladığı hızlı iletişim altyapısı sayesinde daha esnek ve dinamik bir yönetim mümkün olmaktadır. Örneğin, trafik sıklığını azaltmak için trafik ışıkları, algılayıcılardan tarafından toplanan verilere göre ayarlanabilmektedir (Solanas et al., 2014). Bu tür sistemler, şehirlerin günlük işleyişini iyileştirmekte ve buralarda yaşayanların yaşam kalitesine katkıda bulunmaktadır.

Ayrıca, akıllı şehir çözümleri kapsamında güvenlik hizmetleri de geliştirilmektedir. Kameralar ve çeşitli IoT algılayıcıları kullanılarak gerçekleştirilen izleme ve takip

sistemleri suç oranlarının azaltılmasına katkıları sağlamaktadır. Ayrıca, izleme ve takip sistemleri sayesinde afet yönetimi süreçleri de iyileştirilebilmektedir (Zanella et al., 2014). Böylece IoT'nin, şehirlerin daha yaşanabilir ve güvenli hale gelmesinde önemli rol oynadığı görülmektedir.

2.2. Sağlık Uygulamaları

IoT ağları, sağlık sektöründe önemli bir dönüşüm sağlayarak hasta izleme, takip ve veri toplama süreçlerini daha verimli hale getirmiştir. Giyilebilir teknolojiler ve sağlık cihazları aracılığıyla hastaların kalp atış hızı, kan basıncı oksijen ve şeker seviyeleri gibi önemli parametreler sürekli olarak izlenmekte ve bazı durumlarda gerekli müdahaleler de yapılabilmektedir (Islam et al., 2015). Ayrıca bu ağlardan elde edilen veriler, sağlık profesyonelleri tarafından değerlendirilip analiz edilerek hastalara uygun tedavi yöntemlerinin belirlenmesine katkıları sağlamaktadır. Şekil 2.2'de sağlık sektöründe IoT uygulama örneği gösterilmiştir. Burada kullanılan sistem ile uzaktan hasta izleme ve kontrol etme süreçleri hızlandırılmıştır. Giyilebilir cihazlar, hastaların kan basıncı, kalp atış hızı gibi parametrelerini sürekli izler ve uzaktan takip sağlar (Islam et al., 2015).



Şekil 2.2. IoT ile sağlık uygulamaları

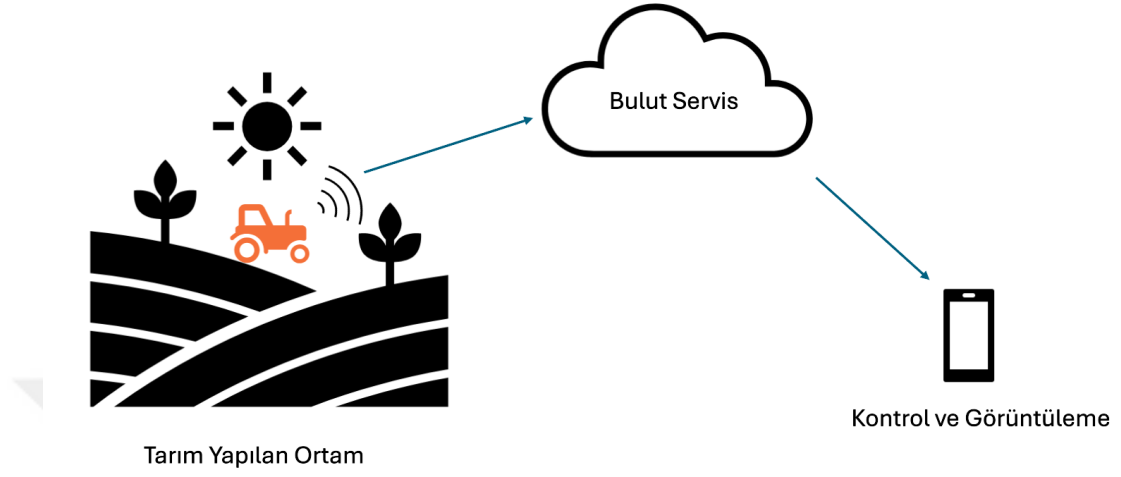
Sağlık hizmetlerinde IoT, özellikle kronik hastalıkların yönetiminde büyük katkılar sağlamıştır. Örneğin, diyabet ve hipertansiyon gibi sürekli takip gerektiren hastalıklarda, IoT cihazları aracılığıyla uzaktan takip yapılabilen ve hasta ile doktor arasındaki iletişim hızlandırılmaktadır (Satyanarayanan, 2017). Bu durum hem hasta memnuniyetini artırmakta hem de sağlık sistemlerindeki yükü oldukça azaltmaktadır.

Bunun yanı sıra, IoT'nin sağlık sektöründe acil durum yönetimi ve epidemiyolojik analizlerde de kullanıldığı bilinmektedir. Pandemi dönemlerinde IoT cihazlarının, hasta takibi ve kaynak dağıtımı gibi kritik süreçlerin yönetiminde önemli rol oynadığı da bilinmektedir (Niu et al., 2015).

2.3. Tarım Uygulamaları

IoT'nin tarım sektöründeki uygulamaları, üretim süreçlerinin modernizasyonunda önemli bir yer tutmaktadır. Toprak nemi, hava durumu ve sıcaklık gibi kritik parametreler, IoT tabanlı algılayıcılar aracılığıyla ölçülmekte ve bu veriler çiftçilere anlık bilgi sağlamaktadır (Kamilaris, Kartakoullis, & Prenafeta-Boldú, 2017). Bu durum, tarımsal üretimde daha bilinçli kararlar alınmasını mümkün hale getirmektedir. Şekil 2.3'te IoT'nin tarım sektöründe üretim süreçlerinin dijitalleşmesine olanak

sağladığı durumlar gösterilmiştir. Örneğin, toprak nemi ve hava durumu verileri algılayıcılarla ölçülerek çiftçilere bilinçli sulama ve gübreleme imkânı tanımaktadır (Kamilaris et al., 2017).



Şekil 2.3. IoT ile tarım uygulamaları

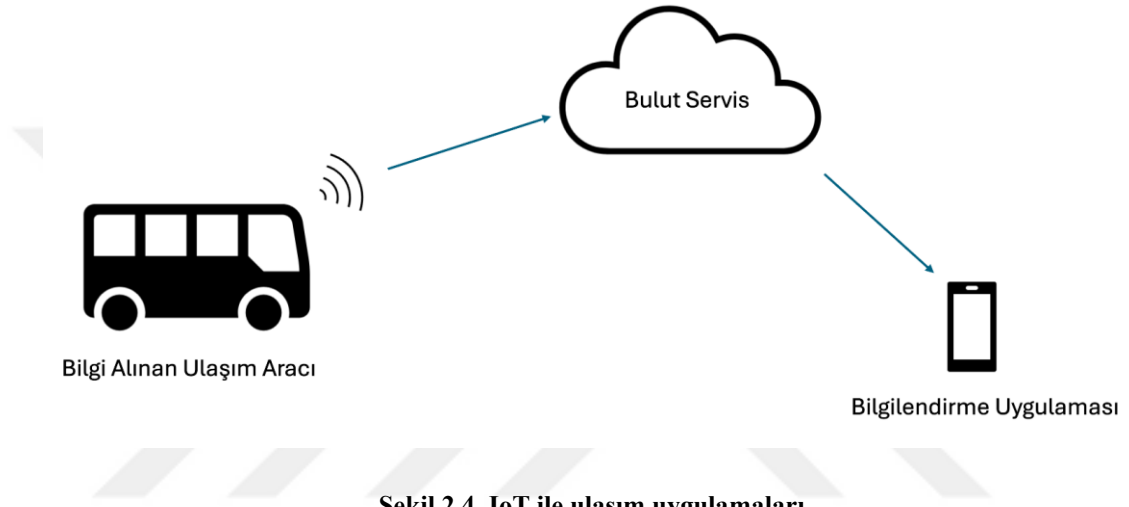
Akıllı tarım uygulamaları kapsamında IoT cihazları, sulama sistemlerini optimize ederek su kaynaklarının verimli kullanımına katkı sağlamaktadır. Aynı şekilde, gübreleme süreçleri de gerçek zamanlı verilerle desteklenmekte ve çevresel etkiler en aza indirilmektedir (Wang et al., 2018). Bu yaklaşımlar, tarımsal üretkenliği artırırken sürdürülebilirlik açısından da olumlu sonuçlar sunmaktadır.

Hayvancılık sektöründe de IoT'nin önemli katkıları olduğu bilinmektedir. Hayvanların sağlık durumları ve hareketleri IoT cihazlarıyla izlenmekte ve takip edilmekte, böylece verimlilik artırılmakta ve olası sağlık sorunlarına hızlı müdahale imkanı sağlanmaktadır (Chiang & Zhang, 2016).

2.4. Ulaşım Uygulamaları

IoT, ulaşım sektöründe trafik yönetim sistemlerinden otonom araç teknolojilerine kadar çeşitli yenilikler sunmaktadır. Trafik akışını izlemek ve sıkışıklığı azaltmak için kullanılan IoT tabanlı sistemler, gerçek zamanlı veri analizine dayalı dinamik çözümler sunmaktadır (Khan et al., 2012). Bu durum hem seyahat sürelerini

kısaltmakta, hem daha fazla konfor sunmakta, hem de yakıt tüketimini azaltarak ekonomiye katkılar sağlamaktadır. Şekil 2.4'te IoT'nin ulaşımında trafik yönetimi ve otonom araç teknolojileri kullanımı gösterilmiştir. Gerçek zamanlı veri analizi ile trafik sıkışıklığı azaltılarak yol güvenliği artırılmaktadır (Khan et al., 2012). Otonom araçlar ise IoT ağları sayesinde birbirleriyle iletişim kurarak kazaların önlenmesine ve azaltılmasına yardımcı olmaktadır (Zanella et al., 2014). Bunun yanında, IoT teknolojileri, taşımacılık hizmetlerinde, filo yönetimi ve yakıt optimizasyonu gibi süreçlerde de aktif olarak faydalar sağlamaktadır.



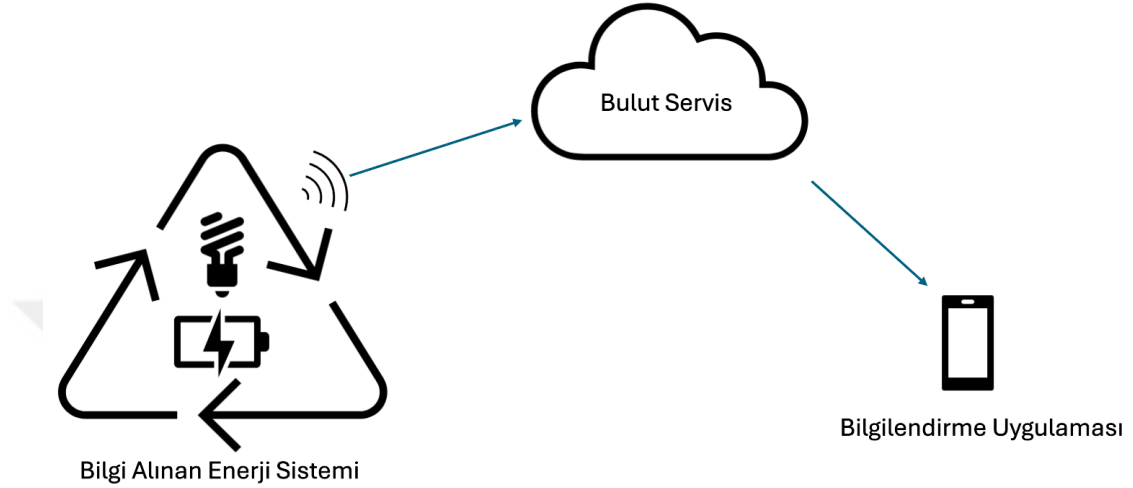
Şekil 2.4. IoT ile ulaşım uygulamaları

Ayrıca, akıllı ulaşım sistemleri kapsamında toplu taşıma araçlarının verimliliği artırılmakta ve şehirlerin altyapı sistemleri de daha iyi yönetilmektedir (Gungor et al., 2011). Bu tür sistemler hem ekonomik hem de çevresel faydalar sunmaktadır.

2.5. Enerji Uygulamaları

IoT'nin enerji sektöründeki uygulamaları, enerji tüketiminin izlenmesi ve optimize edilmesi konularında önemli bir dönüşüm sağlamaktadır. Akıllı sayaçlar aracılığıyla enerji tüketimi sürekli olarak izlenmekte ve kullanıcıların enerji tasarrufu yapmasına olanak tanınmaktadır (Gungor et al., 2011). Şekil 2.5'te enerji sektöründe IoT kullanımına örnek verilmiştir. Akıllı sayaçlar ile enerji tüketimi uzaktan takip

edilebilmektedir. (Gungor et al., 2011). Ayrıca yenilenebilir enerji kaynaklarının entegrasyonu, IoT verileri ile daha etkili bir şekilde sağlanmaktadır. Örneğin, güneş panelleri ve rüzgâr türbinlerinden elde edilen veriler analiz edilerek enerji arz-talep dengesi optimize edilmektedir (Chiang & Zhang, 2016). Bu süreçler, enerji altyapılarının daha sürdürülebilir hale getirilmesine katkılar sağlamaktadır.



Şekil 2.5. IoT ile enerji uygulamaları

Ayrıca IoT teknolojileri, enerji sektöründe cihazların bakım süreçlerinin iyileştirilmesinde de rol almaktadır. Önleyici bakım stratejileri, IoT algılayıcılarından sağlanan verilerle desteklenerek enerji kesintilerinin önlenmesine yardımcı olmaktadır (Huang et al., 2015).

2.6. IoT Heterojen Veri Trafiklerinin Dinamik Yönetimi İlgili Literatürde Sunulan Çalışmalar

Literatür incelendiğinde, IoT teknolojilerinde heterojen veri trafiklerinin iletimi üzerine yapılan çalışmaların önemli rol oynadığı görülmektedir. Bu çalışmalar, farklı veri türlerinin aynı ağ üzerinde iletilmesinde karşılaşılan zorluklara odaklanmaktadır (Niu et al., 2015). Özellikle servis kalitesi gereksinimlerinin karşılanması ve veri iletim süreçlerinin optimize edilmesi yaygın olarak ele alınan konulardır. Bu durum, özellikle yoğun düğüm ortamlarında, mevcut çözümlerin yetersiz kaldığını ve daha esnek ve ölçeklenebilir yöntemlerin geliştirilmesi gerektiğini ortaya koymaktadır (Gubbi, Buyya, Marusic, & Palaniswami, 2013).

Son yıllarda yapılan çalışmalarda, yoğun ağ ortamlarında klasik kanal tahsis yöntemlerinin yetersiz kaldığı ve yenilikçi yaklaşımların gerekliliği vurgulanmaktadır (Chiang & Zhang, 2016). Bu bağlamda, dinamik kanal tahsis yöntemleri öne çıkan çözüm seçeneklerinden biri olarak dikkat çekmektedir. Burada kanal tahsisinin düğüm gereksinimlerine göre dinamik olarak ayarlanması gerekliliği vardır. Literatürde, özellikle yoğun ağ ortamlarında, sabit kanal tahsis yöntemlerinin gerekli servis kalitesi desteğini sağlayamadığı vurgulanmıştır. Dinamik yaklaşımlar sayesinde düğümlerin değişken bant-genişliği talepleri karşılanarak hem başarımların iyileştirilmesi hem de sınırlı kaynaklar için adil bir paylaşım sağlandığı vurgulanmaktadır (Kuo & Ko, 2017). Bununla birlikte, IoT ağlarının etkin yönetimi için heterojen veri trafiklerinin bütüncül bir yaklaşımla ele alınması gerektiği de ifade edilmektedir (Satyanarayanan, 2017).

Atmaca (2025), yaptığı bir çalışmada IoT uygulamalarında heterojen veri trafiklerinin etkin ve adil iletimi için servis kalitesi destekli yeni bir ortam erişim kontrol protokolü tasarlamıştır. Önerilen çalışma, analitik ve benzetim modellerinin kolaylığı açısından ağ ortamında sonsuz sayıda IoT düğümünün olduğunu kabul etmiştir. Bu çalışmadan farklı olarak, sunulan tez çalışmasında önerilen kanal tahsis yönteminin sonlu sayıda IoT düğümlerinin kullanıldığı gerçek ağ ortamına daha yakın bir ağ ortamı için analitik ve benzetim modelleri geliştirilmiştir.

Banerjee vd. (2016), IEEE 802.15.4 tabanlı ağlardaki başarımların sorunlarını çözmek için BSMAC (Backoff Freezing and Sleep MAC) protokolünü geliştirmiştir. Bu protokol, yüksek veri trafiği durumlarında oluşan düşük verimlilik ve yüksek gecikme gibi problemleri ele almaktadır. BSMAC protokolü, veri iletimi sırasında mevcut süper çerçevede zaman yetersizliği ortaya çıktığında "backoff" sayacını durdurarak düğümü uyku moduna geçirir. Uyku modu, enerji tüketimini önemli ölçüde azaltırken belirli bir süre sonra düğüm tekrar etkinleştirilir ve sayaç kaldığı yerden devam eder. BSMAC protokolü, özellikle enerji verimliliğini artırmada başarılıdır. Ancak, rastgele uyku süreleri karmaşık ağ senaryolarında verimliliği sınırlayabilmektedir. Bu durum, güç tüketimi ve verimlilik arasındaki dengeyi sağlayacak şekilde optimize edilmiştir. Bu protokolün başarımların değerlendirilmesi, üç boyutlu bir Markov zinciri ile gerçekleştirilmiş ve analitik modelden elde edilen sonuçlar ns-2 benzetim aracı ile gerçekleştirilen benzetim modelinden elde edilen sonuçlar ile doğrulanmıştır. Başarımların kriterleri olarak uçtan uca gecikme, iş çıkarma oranı ve güvenilirlik olarak

belirlenmiştir. Sonuçlar, enerji tüketiminin azaltıldığını ve düşük gecikme ile yüksek verimlilik sağlandığını göstermiştir.

Nurzaman vd. (2016), IoT ağlarında yaygın olarak kullanılan IEEE 802.15.4 protokolü ile IEEE 802.11ah protokolünün karşılaştırmalı başarımlarını değerlendirmesini gerçekleştirmiştir. Bu çalışmada IEEE 802.11ah protokolünün daha geniş kapsama alanı ve daha yüksek veri iletim hızları sunduğu vurgulanmıştır. Özellikle, büyük ölçekli IoT uygulamaları için ilişkilendirme süresi, uçtan uca gecikme ve ağ kapsama alanı gibi parametrelerde üstünlük sağlamaktadır. Protokolün kullanıldığı ağ senaryosundan elde edilen benzetim sonuçlarına göre, 802.11ah, veri iletim hızında ve iş çıkarma oranında belirgin bir iyileşme sunarken, enerji verimliliği açısından IEEE 802.15.4'e göre daha düşük başarımlar göstermiştir. Örneğin, enerji tasarrufu gerektiren uygulamalarda IEEE 802.15.4 daha uygun bulunurken, geniş kapsama alanı gerektiren senaryolarda 802.11ah tercih edilmektedir. Bu durum, her iki protokolün de belirli kullanım alanlarına göre optimize edilmesi gerektiğini göstermektedir.

Mondal vd. (2023), IoT ağlarında aşağı bağlantı (downlink) trafiğini etkin şekilde yönetmek için IoT-MAC protokolünü geliştirmiştir. Protokolde, Kısıtlı Erişim Penceresi (RAW) yöntemi kullanılarak çarpışma oranı azaltılmıştır. Bu yöntem, düğümleri belirli gruplara ayırarak her birinin belirli zaman dilimlerinde kanala erişmesini sağlamaktadır. Böylece, çarpışma olasılığı da önemli ölçüde azaltılmıştır. IoT-MAC, yukarı bağlantı (uplink) trafiğinin periyodikliğini göz önünde bulundurarak aşağı bağlantı etkinliğini optimize etmektedir. Bu süreçte, kritik trafıklere öncelik verilmiş ve daha fazla çekişme olmadan yeni bir RAW planlanmıştır. Başarımların analizleri, iş çıkarma oranı, uçtan uca gecikme, güç tüketimi ve paket kaybı kriterlerinde önemli iyileşmeler göstermiştir. Ancak, gruplama işlemi yoğun ağlarda ek işlem yüküne neden olabilmektedir. Bu durum, ağ verimliliği ve işlem karmaşıklığı açısından bir denge kurulması gerektiğini ortaya koymaktadır.

Mayssa vd. (2022), gerçek zamanlı IoT uygulamalarının gecikme gereksinimlerini karşılamak için CaWuQoS-MAC (Collision Avoidance and Wake-up QoS MAC) protokolünü geliştirmiştir. Bu protokol, paketin iletilmesi gereken maksimum gecikme süresini belirler ve süresi dolan paketlerin silinmesini sağlar. Ayrıca, çarpışma önleme mekanizması ile birlikte uyandırma radyosu (WuR) teknolojisi de kullanılmaktadır. Bu protokol hem düşük hem de yüksek trafik yüklerinde güvenilirlik ve düşük gecikme sağlama konusunda başarılıdır. Özellikle ağır trafik altında iş çıkarma oranını yüksek

tutabilmektedir. Ancak, WuR teknolojisi, düğüm enerji tüketimini artırabilecek ek donanım gereksinimlerini de beraberinde getirmektedir. Protokolün başarımı, ayrık zamanlı Markov zinciri modeliyle analiz edilmiş ve benzetim sonuçları ile analitik modelden elde edilen sonuçlar doğrulanmıştır.

Liu vd. (2019), IoT ağlarında dinamik trafik koşullarını yönetmek için bulanık mantık tabanlı Fuzzy-MAC protokolünü geliştirmiştir. Protokol, düğümlerin mevcut trafik yüklerine ve enerji durumlarına göre iletim sıklığını uyarlayarak enerji verimliliğini artırmaktadır. Bu çalışmada ayrıca, çarpışma oranını düşürmek için trafik yoğunluğuna duyarlı bir kanal erişim mekanizması da geliştirilmiştir. Bulanık mantık yaklaşımı, protokolün değişken koşullara uyum sağlama kabiliyetini artırmıştır. Başarım değerlendirmelerinde güç tüketimi, uçtan uca gecikme ve iş çıkarma oranı metrikleri benimsenmiştir. Ancak, bulanık mantık kurallarının belirlenmesi ve uygulanması ek işlem yükü getirmektedir. Bu durum, özellikle düşük işlem gücüne sahip cihazlarda başarım sınırlamalarına yol açabilmektedir.

3. IoT ORTAM ERİŞİM KONTROL YÖNTEMLERİ ve VERİ TRAFİK TÜRLERİ

IoT uygulamalarında kullanılan kanal erişim ve tahsis yöntemleri genellikle algılayıcı ağlarda kullanılan Ortama Erişim Kontrol (Medium Access Control, MAC) protokolleri veya bu protokollerin yeniden düzenlenmiş şekilleridir. Bu yöntemler, IoT ağlarında düğümlerin ağ kaynaklarını verimli şekilde paylaşmalarını ve veri çakışmalarını önlemek için kullanılan temel mekanizmalardır. Bu yöntemler, cihazların veri iletim zamanlarını, erişim yöntemlerini ve kaynak tahsis süreçlerini düzenleyerek ağın başarımını optimize etmeyi hedefler. IoT ağlarının çeşitliliği, ortam erişim kontrol protokollerinin uygulanmasında farklı gereksinimlerin dikkate alınmasını zorunlu kılar. Bu bölümde, yaygın olarak kullanılan ortam erişim kontrol protokolleri temel özellikleriyle birlikte kısaca tanıtılmaktadır.

3.1 IoT Ağlarında Kullanılan Ortam Erişim Kontrol Teknikleri

IoT ağlarında kullanılan ortam erişim kontrol teknikleri, farklı ağ gereksinimlerine ve uygulama senaryolarına uygun çözümler sunmak için farklı yaklaşımlar ve teknikler kullanılmaktadır. Ağın büyüklüğü, ağda bulunan kullanıcı sayısı, ağda iletilecek veri trafiği türü ve diğer ağ parametreleri de dikkate alınarak uygun ortam erişim ve kontrol tekniği benimsenmektedir. Ortam erişim ve kontrol teknikleri, çekişme tabanlı (Contention-based) teknikler, tahsis tabanlı (Scheduled-based) teknikler ve bu iki tekniğin bazı yönlerinin birlikte kullanıldığı hibrit yöntemler olarak genellikle 3 farklı şekilde gerçekleştirilir (Rom & Sidi, 1990). Alt bölümlerde bu yöntemler açıklanmaktadır.

3.1.1. Çekişme tabanlı (Contention-Based) protokoller

Bu protokoller, ağdaki cihazların ortak kanalı paylaşarak verilerini iletmek için kullandığı protokollerdir. Bu protokollerde, cihazlar arasında önceden belirlenmiş bir

kanal tahsisi yoktur. Bunun yerine, cihazlar verilerini göndermek istediklerinde kanala erişmek için rastgele bir zamanlama mekanizması kullanmaktadır. Bu durum, özellikle yoğun ağlarda, kanal erişimi sırasında çakışmalar veya çarpışmalar (collision) yaşanmasına neden olabilmektedir (Rom & Sidi, 1990). Bunun için bu protokollerde çarpışma durumunda yeniden gönderim veya başka çözüm yöntemleri kullanılmaktadır. Bu protokoller özellikle dağıtık ve esnek yapıları nedeniyle IoT ve kablosuz ağlar gibi dinamik sistemlerde yaygın olarak kullanılmaktadır. Ancak çarpışmaların ve gecikmelerin en aza indirilebilmesi için tasarım optimizasyonlarına ihtiyaç duymaktadır. Aşağıdaki alt bölümlerde çekişme tabanlı bazı rastgele erişim teknikleri sunulmaktadır.

3.1.1.1. ALOHA

ALOHA, ortam erişimi için kullanılan ilk protokoller arasında yer almakta ve oldukça basit bir yapı sunmaktadır. Bu protokolle, cihazlar herhangi bir zaman kısıtlaması olmaksızın veri iletimlerini gerçekleştirirler. Eğer iletilen veriler çakışırsa, cihazlar bir süre bekledikten sonra aynı veriyi tekrar iletirler. ALOHA'nın bu özelliği, küçük ölçekli ve düşük yoğunluklu ağlarda kabul edilebilir bir başarımları sağlamaktadır (Abramson, 1970).

Ancak ALOHA protokolü, çakışma oranlarının yüksek olması nedeniyle büyük ölçekli ağlarda yetersiz kalmaktadır. Verilerin çakışması durumunda ağ kaynakları israf olur ve ağ başarımları düşer. Bu durum, özellikle veri trafiğinin yoğun olduğu IoT ağlarında ciddi sorunlara yol açabilmektedir. Buna rağmen, protokolün basit yapısı ve uygulama kolaylığı, ALOHA'yı düşük maliyetli ağlar için cazip bir seçenek haline getirmektedir.

ALOHA'nın başarımlarını artırmak amacıyla çeşitli iyileştirmeler yapılmıştır. Örneğin, cihazların rastgele veri iletimi yerine belirli zaman aralıklarında iletim yapmasını sağlayan Slotted ALOHA, bu protokolün geliştirilmiş bir versiyonudur (Rom & Sidi, 1990).

3.1.1.2. Slotted ALOHA

Slotted ALOHA, klasik ALOHA protokolünün çakışma oranlarını azaltmak için geliştirilmiş bir versiyonudur (Rom & Sidi, 1990). Bu protokolle, veri iletimi önceden tanımlanmış zaman dilimlerine bölünür. Cihazlar yalnızca bu zaman dilimlerinin

başlangıcında veri iletimlerini gerçekleştirebilirler. Bu sayede, iki cihazın aynı anda veri iletme olasılığı önemli ölçüde düşürülmüştür (Roberts, 1972).

Slotted ALOHA, paket çarpışma oranlarını klasik ALOHA'ya kıyasla yarı yarıya azaltarak ağ başarımını artırmaktadır. Ancak, cihazların veri iletimi için uygun bir zaman dilimi beklemesi gerektiğinden, bu protokol gecikmelere yol açabilmektedir. Buna rağmen, düşük yoğunluklu IoT ağlarında Slotted ALOHA, çarpışma ve veri kaybını azaltmak için etkili bir çözüm sunmaktadır.

Bu yöntemin bir diğer avantajı, protokolün uygulanabilirliğinin hala basit ve maliyet etkin olmasıdır. Slotted ALOHA, özellikle sabit bir cihaz sayısına sahip küçük ölçekli IoT uygulamaları için ideal bir protokol seçeneği olarak öne çıkmaktadır.

3.1.1.3. CSMA

Carrier Sense Multiple Access (CSMA), bir cihazın veri iletimi yapmadan önce kanalı kontrol ederek kullanılıp kullanılmadığını belirlemesini sağlayan bir ortam erişim tekniğidir (Rom & Sidi, 1990). Bu teknik, çarpışmaların önlenmesine yönelik basit bir mekanizma sunmaktadır. Kanal boşsa cihaz hemen veri iletimine başlar; aksi durumda belirli bir süre bekleyerek yeniden deneme yapar (Kleinrock & Tobagi, 1975). Bu özellik, CSMA'yı düşük yoğunluklu ağlar ve düşük maliyetli uygulamalar için uygun bir teknik haline getirmektedir.

CSMA'nın temel avantajı, paket çarpışmalarını azaltarak ağ başarımını artırmasıdır. Bu teknik, cihazların kanalı aktif bir şekilde dinlemesini sağladığı için ağdaki veri trafiğini düzenler. Ancak, kanalın kontrol edilmesi için geçen süre ve çarpışma sonrası yeniden deneme işlemleri, gecikmelere neden olabilmektedir. Ayrıca, yoğun trafiğe sahip ağlarda çarpışmaların tamamen önlenmesi mümkün olmayabilir ve bu durum başarımların azalmasına yol açabilmektedir (Kleinrock & Tobagi, 1975).

CSMA, IoT uygulamalarında yaygın olarak kullanılan tekniklerden biridir. Özellikle düşük yoğunluklu IoT ağlarında, cihazların veri iletimi sırasında paket çarpışmalarını minimum düzeyde tutar. Ancak, yoğun trafiğe sahip büyük ölçekli ağlarda CSMA'nın başarımlarını sınırlı olabilir ve bu gibi durumlarda daha gelişmiş ortam erişim protokollerine ihtiyaç duyulabilir. Ancak, düşük maliyeti ve uygulanabilirliği nedeniyle IoT ağlarının birçok farklı türünde yaygın olarak tercih edilmektedir. (Kleinrock & Tobagi, 1975).

3.1.2. Tahsis tabanlı (Scheduled Access) protokoller

Bu tür protokoller, ağ kaynaklarının (örneğin bant genişliği veya zaman dilimleri) önceden belirlenmiş bir düzende kullanıcılara tahsis edilmesini sağlar. Çarpışma olasılığı yok denecek kadar azdır çünkü her kullanıcıya iletim önceki belirli bir kaynak tahsisi gerçekleştirilir. Alt bölümlerde temel tahsis tabanlı ortam erişim ve kontrol teknikleri sunulmaktadır.

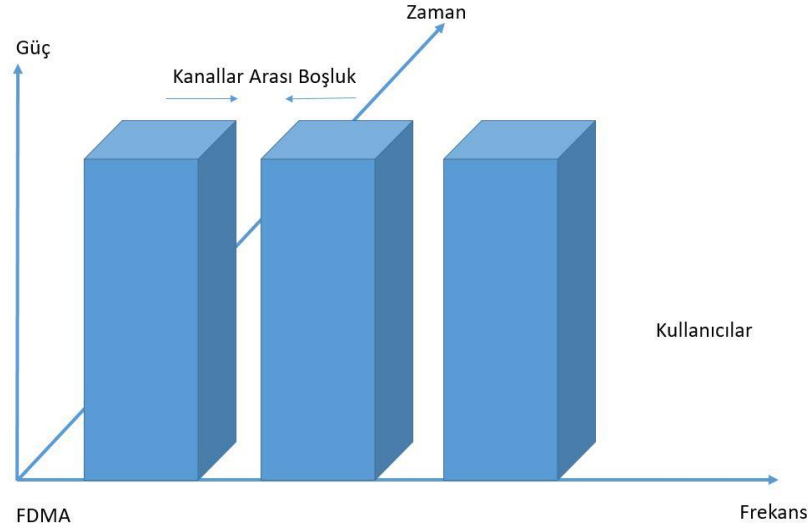
3.1.2.1. FDMA (Frequency Division Multiple Access)

FDMA, cihazların veri iletiminde farklı frekans bantları kullanmasını sağlayan bir tekniktir. Bu teknikte, her cihaza belirli bir frekans bandı atanır ve bu bant yalnızca o cihaz tarafından kullanılır. Böylece, cihazlar arasında çakışma tamamen önlenir (Goldsmith, 2005).

FDMA'nın en büyük avantajı, aynı anda birden fazla cihazın veri iletimi yapabilmesine olanak tanınmasıdır. Ayrıca, bu yöntem, frekans bantlarının sabit bir şekilde tahsis edilmesi sayesinde ağın kararlı bir başarımlı göstermesini sağlar. Ancak, bant genişliğinin sınırlı olması, çok sayıda cihazın bulunduğu büyük IoT ağlarında bu protokolün uygulanabilirliğini sınırlandırmaktadır.

Bu teknik, genellikle sabit ve yüksek bant genişliği gereksinimi olan IoT uygulamalarında tercih edilir. Örneğin, endüstriyel IoT ağlarında FDMA, veri iletimini optimize etmek için etkili bir çözüm sunmaktadır.

Şekil 3.1'de, FDMA tekniğinde kanal dağılımı gösterilmiştir. Bu yöntemde, her cihaz farklı bir frekans bandına atanır ve bu bant yalnızca ilgili cihaz tarafından kullanılır. Kanallar arasındaki boşluk frekans çakışmalarının önlenmesi için kullanılır. Bu yaklaşım, ağdaki parazitlerin en aza indirilmesi ve veri iletiminde daha yüksek güvenilirlik sağlanması açısından etkili bir çözümdür. FDMA'nın bu özelliği, özellikle endüstriyel IoT uygulamaları ve kritik görevlerde sıklıkla kullanılmaktadır (Goldsmith, 2005).



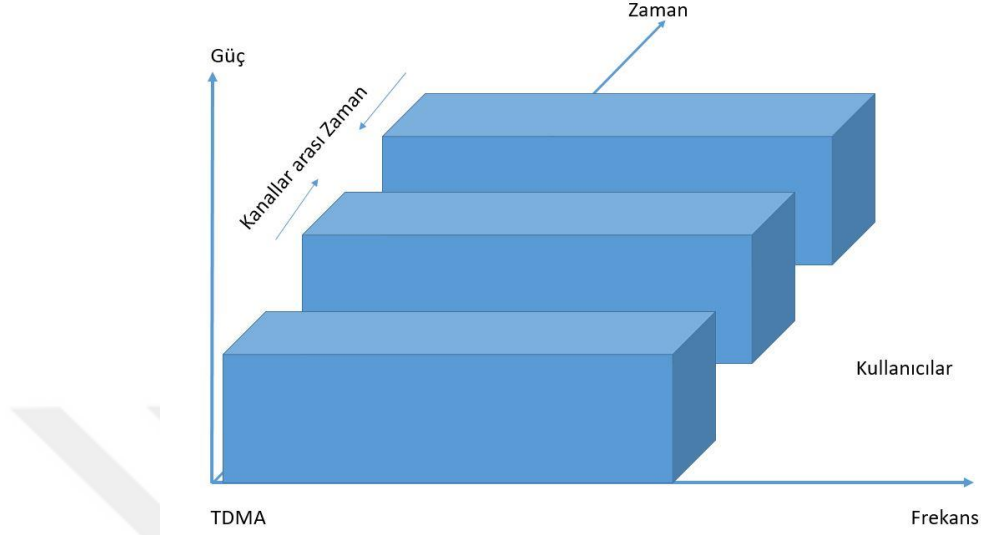
Şekil 3.1. FDMA tekniğinde kanal kullanımı

3.1.2.2. TDMA (Time Division Multiple Access)

TDMA, çizelge tabanlı bir ortam erişim kontrol yöntemi olarak, cihazların veri iletim zamanlarını önceden belirleyerek paket çarpışmalarını önler. Bu teknikle, her düğüme belirli bir zaman dilimi atanır ve ilgili düğüm yalnızca bu zaman diliminde veri iletimini gerçekleştirir (Rom & Sidi, 1990). Bu teknik, ağ kaynaklarının daha verimli bir şekilde kullanılmasını sağlar (Proakis & Salehi, 2008).

TDMA'nın en büyük avantajı, paket çarpışma olasılığını ortadan kaldırmasıdır. Bununla birlikte, cihazların yalnızca belirlenen zaman dilimlerinde aktif olması, enerji tüketimini önemli ölçüde azaltır. Bu özellik, pille çalışan IoT cihazları için kritik öneme sahiptir. Ayrıca, ağın tahmin edilebilir yapısı, güvenilir veri iletimi sağlar. Ancak, TDMA, hassas zaman dilimi senkronizasyonu gerektirdiğinden, tekniğin uygulanmasını daha zor hale getirir. Dinamik IoT ağlarında cihazların sürekli eklenip çıkarılması, zaman çizelgesinin yeniden düzenlenmesini zorunlu kılar. Bu nedenle, TDMA genellikle sabit yapıya sahip IoT ağlarında daha yaygın olarak kullanılmaktadır. Şekil 3.2'de, TDMA'nın kanal yapısı görülmektedir. Bu yöntemde, her cihazın belirli bir zaman diliminde veri iletmeye izin verilmektedir. Şekilde, cihazların zaman dilimlerine göre sıralandığı ve yalnızca kendi zaman dilimlerinde veri gönderdiği görülmektedir. Bu düzenleme, paket çarpışmalarını önler ve enerji

tasarrufu sağlar. Özellikle pille çalışan IoT cihazlarında TDMA'nın düşük enerji tüketimi, ağ verimliliğini artırmak için kritik bir rol oynar (Proakis & Salehi, 2008).



Şekil 3.2. TDMA tekniğinde kanal yapısı

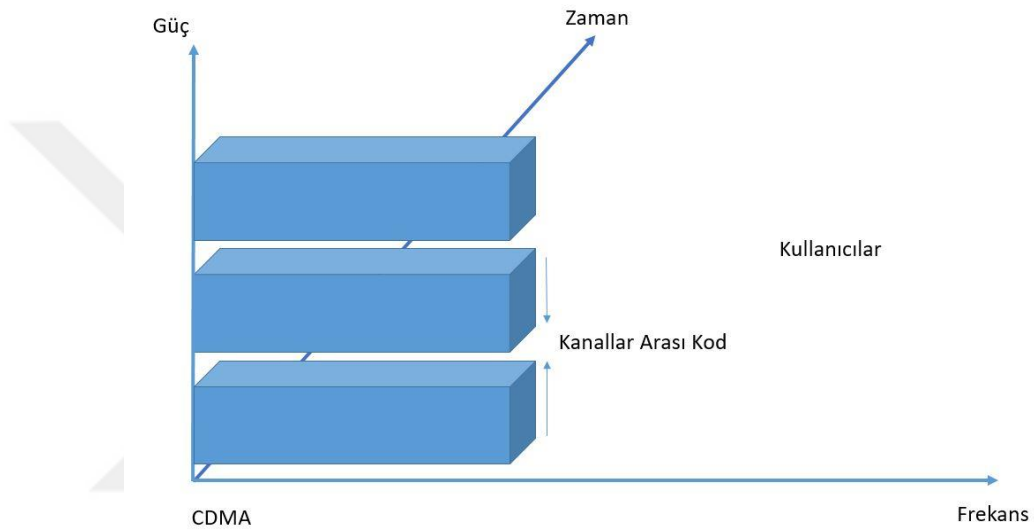
3.1.2.3. CDMA (Code Division Multiple Access)

CDMA, cihazların aynı frekans bandını kullanmasına rağmen farklı kodlarla veri iletimi yapmasını sağlayan bir tekniktir (Rom & Sidi, 1990). Bu teknikte, her cihaz kendine özgü bir kod kullanarak veri gönderir. Bu sayede, aynı anda birden fazla cihazın veri iletimi yapması mümkün hale gelir ve paket çarpışmaları büyük ölçüde önlenir (Viterbi, 1995).

CDMA'nın en büyük avantajı, ağ kaynaklarının çok daha verimli bir şekilde kullanılmasıdır. Farklı kodlarla veri iletimi, aynı frekans bandında daha fazla cihazın çalışabilmesini sağlar. Ayrıca, CDMA, ağ yoğunluğu arttıkça başarımını koruyabilen bir yöntemdir ve bu nedenle geniş ölçekli IoT ağları için tercih edilmektedir.

CDMA tekniğini kullanan protokol geliştirilmesi, FDMA ve TDMA'ya kıyasla daha karmaşıktır. Kod yönetimi ve tahsisi, ek bir ağ yönetimi gerektirir. Ancak, CDMA'nın sağladığı yüksek verimlilik ve esneklik, bu zorlukları telafi etmektedir. Özellikle mobil IoT ağlarında CDMA, güvenilir ve hızlı veri iletimi için sıkça tercih edilmektedir (Rom & Sidi, 1990).

Şekil 3.3, CDMA tekniğinin kanal yapısı gösterilmiştir. Şekilde, aynı frekans bandını paylaşan cihazların, her birinin kendine özgü bir kod kullanarak veri ilettiği gösterilmektedir. Bu yöntem, aynı anda birden fazla cihazın veri iletilmesini mümkün kılar ve frekans çakışmalarını önler. CDMA'nın bu özelliği, ağ kaynaklarının verimli kullanımını ve cihaz yoğunluğu arttıkça başarımın korunmasını sağlar (Viterbi, 1995). Özellikle geniş ölçekli IoT ağlarında CDMA'nın sunduğu bu avantajlar, onu kritik bir yöntem haline getirmektedir.



Şekil 3.3. CDMA kanal yapısı

3.2. IoT Uygulamalarında Kullanılan Heterojen Veri Trafikleri ve Özellikleri

IoT uygulamaları da diğer ağ uygulamaları gibi birbirinden farklı cihazların iletişimini mümkün kılarak çok çeşitli veri türlerinin aynı ağ üzerinden iletilmesine olanak tanımaktadır. Bu ağlar, farklı servis gereksinimlerine, veri hızlarına sahip heterojen veri akışlarını desteklemektedir. Farklı veri türlerinin bu şekilde bir araya gelmesi, ağ yönetiminde karmaşıklığa neden olmaktadır (Zanella et al., 2014). Bu bağlamda, IoT ağlarının başarımlarının artırılması için veri hızları, gecikme toleransı ve veri çeşitliliği gibi parametrelerin dikkate alınması gerekmektedir (Niu et al., 2015).

Heterojen veri trafiklerinin etkili bir şekilde yönetimi, ağ başarımı üzerindeki yükün azaltılmasında kritik bir rol oynamaktadır. Bu durum, özellikle farklı uygulama türleri için tasarlanmış IoT cihazlarının servis ihtiyaçlarının değişkenlik göstermesi nedeniyle önem kazanmaktadır. Örneğin, video akışı yüksek bant genişliği gerektirirken, sıcaklık sensörlerinden gelen veriler daha düşük bant genişliğiyle aktarılabilmektedir (Atzori et al., 2010).

Heterojen veri trafiklerinin temel özellikleri, veri hızı ve gecikme toleransı başlıkları altında ele alınmaktadır. Bu başlıklar, ağ tasarımı ve yönetimi açısından önemlidir.

3.2.1. Veri trafik türleri

IoT cihazları tarafından iletilen veri türleri, ağ yönetimi açısından büyük önem taşır. Bu veriler; algılayıcı verileri, video, ses ve metin gibi farklı formatlarda olabilir ve her bir veri türü, kendine özgü yönetim gereksinimleri gerektirir. Örneğin, ses ve video verileri gerçek zamanlı iletimi gerektirirken, metin tabanlı veriler daha düşük önceliklendirme ile işlenebilmektedir (Zanella et al., 2014).

Bu çeşitlilik, ağ başarımının optimize edilmesi ve hizmet kalitesinin artırılması için detaylı bir planlama yapılmasını zorunlu kılmaktadır. Ayrıca, veri türlerinin ihtiyaç duyduğu bant genişliği ve gecikme gereksinimlerinin bilinmesi, ağ kaynaklarının etkili bir şekilde tahsis edilmesini sağlamaktadır. Sensör verileri genellikle düşük bant genişliği gerektirirken, yüksek çözünürlüklü video akışları daha yüksek bant genişliğine ihtiyaç duymaktadır (Islam et al., 2015).

Bununla birlikte, veri türlerinin çeşitliliği, ağ yük dengeleme mekanizmalarını da etkilemektedir. Veri iletim sürecinde karşılaşılan farklı gereksinimlerin etkili bir şekilde bir arada çalıştırılması, özel tasarlanmış protokollerin geliştirilmesini gerektirmektedir (Bandyopadhyay & Sen, 2011).

3.2.2. Veri hızı

IoT uygulamalarında kullanılan cihazlar arasında, farklı hızlarda veri iletimi yapılmaktadır. Örneğin, bir güvenlik kamerasından elde edilen veriler saniyede birkaç megabit hızında iletilirken, sıcaklık sensörlerinden gelen veriler yalnızca birkaç kilobit

hızında iletilmektedir (Atzori et al., 2010). Bu durum, ağ üzerindeki bant genişliği yönetimini karmaşıklaştırmaktadır.

Ağ başarımını artırmak amacıyla, yüksek hızlı veri gereksinimi olan cihazlara öncelik verilmesi ve düşük hızlı cihazların bu süreçte gecikmeye uğramamasını sağlamak için özel algoritmalar kullanılmaktadır. Özellikle video ve ses akışı gibi uygulamalarda yüksek hız ve düşük gecikme gereksinimleri kritik öneme sahiptir (Niu et al., 2015). Bu farklılıklar dikkate alınmadan yapılan ağ tasarımları, veri iletim sürecinde sorunlara ve veri kayıplarına yol açabilmektedir. Bu nedenle, heterojen veri trafiklerinin etkili bir şekilde iletilmesi, ağ başarımını artırmada önemli rol oynamaktadır (Zanella et al., 2014).

3.2.3. Gecikme

IoT cihazlarından iletilen verilerin bir kısmı düşük gecikme toleransına sahipken, bir kısmı bu konuda daha esnektir. Örneğin, sağlık uygulamalarında kullanılan cihazlar genellikle milisaniyeler düzeyinde bir gecikme gereksinimine sahipken, tarım sensörleri için bu süre saatlerle ifade edilebilmektedir (Bandyopadhyay & Sen, 2011). Bu durum, ağ üzerinde veri akışlarını önceliklendiren mekanizmaların geliştirilmesini zorunlu kılmaktadır. Düşük gecikme gerektiren uygulamalara öncelik verilmesi hem veri güvenilirliğini artırmakta hem de kritik uygulamalardaki olası hataları önlemektedir (Islam et al., 2015).

Heterojen veri trafiklerinin gecikme toleransı dikkate alınmadan yönetilmesi, ağ üzerinde ciddi başarımların kayıplarına neden olabilmektedir. Bu nedenle, bu tür parametrelerin etkili bir şekilde analiz edilmesi ve veri iletimi sırasında dinamik bir şekilde yönetilmesi büyük önem taşımaktadır (Zanella et al., 2014).

3.3. Servis Kalitesi (Quality of Service, QoS) Kavramı ve Önemi

IoT ağlarında Servis Kalitesi (QoS), farklı veri türlerinin ihtiyaçlarının karşılanması ve ağ başarımının optimize edilmesi amacıyla kullanılan bir terimdir Boyan (2018). IoT uygulamaları, yüksek çeşitlilikte veri türleri ve iletişim gereksinimlerini barındıran geniş ölçekli ağlar olarak dikkat çekmektedir. QoS, bu ağların etkin ve

verimli çalışmasını sağlamak için gecikme, bant genişliği, jitter (gecikme değişimi) ve veri kaybı gibi parametrelerin kontrol edilmesine odaklanır (Gubbi, Buyya, Marusic, & Palaniswami, 2013).

QoS mekanizmaları, IoT ağlarındaki cihazların farklı gereksinimlerini karşılamak için özelleştirilmiş çözümler sunmaktadır. Örneğin, sağlık izleme cihazları düşük gecikme süreleri gerektirirken, çevresel sensörler için bu gereksinim daha az kritik olabilir. QoS, bu tür farklılıkları ele alarak cihazların ağ üzerindeki başarımlarını optimize eder. Bu hem kullanıcı deneyiminin iyileştirilmesine hem de ağ kaynaklarının daha verimli kullanılmasına olanak sağlamaktadır.

Bununla birlikte, QoS yalnızca başarımların artırımı değil, aynı zamanda güvenilirlik ve veri bütünlüğü açısından da önemlidir. IoT uygulamalarında veri iletimindeki herhangi bir hata veya kesinti, özellikle kritik görevlerde büyük sorunlara yol açabilmektedir. Bu nedenle QoS hem ağ yönetimi hem de veri iletimi süreçlerinde önemlidir.

3.3.1. Bant genişliği

IoT cihazları tarafından kullanılan bant genişliği, cihazın türüne ve gönderdiği verinin formatına bağlı olarak değişiklik gösterir. Örneğin, video gözetim sistemleri yüksek çözünürlüklü video akışları nedeniyle geniş bant gerektirirken, sıcaklık sensörleri gibi cihazlar düşük bant genişliği kullanır (Bandyopadhyay & Sen, 2011). Bu durum, IoT ağlarının heterojen veri trafiğini yönetmek için bant genişliği tahsisini optimize etmesini zorunlu kılar.

Bant genişliği yönetiminde QoS protokolleri, cihazlar arasında kaynakların adil bir şekilde dağıtılmasını sağlar. Bu, yüksek öncelikli uygulamalara ihtiyaç duydukları kaynakların tahsis edilmesiyle mümkün olur. Örneğin, acil durum uyarıları gönderen bir cihazın veri akışı, çevresel izleme yapan bir cihazdan daha fazla bant genişliği gerektirebilir. QoS mekanizmaları, bu tür önceliklendirmeleri dinamik bir şekilde düzenleyerek ağ verimliliğini artırır.

Ayrıca, bant genişliği yönetimi, veri trafiği yoğunluğunun değişkenlik gösterdiği durumlarda kritik öneme sahiptir. Trafik yoğunluğunun artması durumunda, QoS algoritmaları, daha az öncelikli uygulamaların bant genişliğini sınırlayarak ağın

çökmesini önler. Bu, IoT ağlarının hem ölçeklenebilirliğini artırır hem de kritik uygulamalarda güvenilir veri iletimi sağlar.

3.3.2. Gecikme ve gecikme değişimi (jitter)

IoT ağlarında gecikme, veri paketlerinin kaynak cihazdan hedefe ulaşması için geçen süre olarak tanımlanır. Gecikme, özellikle gerçek zamanlı uygulamalarda ağ başarımını doğrudan etkileyen kritik bir parametredir. Örneğin, sağlık izleme sistemleri veya video konferans uygulamaları gibi düşük gecikme gerektiren uygulamalarda, veri paketlerinin hızlı bir şekilde iletilmesi hayati öneme sahiptir (Niu et al., 2015).

Gecikmeyi yönetmek için QoS gereksinimleri, düşük gecikme toleransına sahip uygulamaları önceliklendiren mekanizmalar kullanır. Bu mekanizmalar, veri iletim sürecindeki sorunları minimize ederek kritik görevlerin kesintisiz bir şekilde sürdürülmesini sağlar. Örneğin, ağ trafiği yoğunluğu arttığında, sağlık monitörlerinden gelen veriler öncelikli olarak işlenir.

Jitter, veri paketlerinin hedefe ulaşma süreleri arasındaki farklılık olarak tanımlanır ve ağ başarımını olumsuz etkileyen önemli bir faktördür. Özellikle gerçek zamanlı uygulamalarda jitter, ses ve video kalitesinde bozulmalara neden olabilir. Örneğin, bir IoT tabanlı video gözetim sisteminde yüksek jitter, görüntünün akıcılığını bozabilir ve izleme sürecini kesintiye uğratabilir (Gubbi et al., 2013).

Jitter'ı yönetmek için tamponlama mekanizmaları kullanılabilir. Bu mekanizmalar, veri paketlerini geçici olarak depolayarak sabit bir akış hızında iletim sağlar. Bunun yanı sıra, QoS algoritmaları, ağ üzerindeki veri trafiğini izleyerek düşük jitter seviyelerini koruyacak şekilde trafik yönlendirme yapar. Bu hem veri akışını optimize eder hem de kullanıcı deneyimini iyileştirir.

Dinamik ağ koşullarında jitter kontrolü daha da önem kazanır. Örneğin, ağ trafiği yoğunlaştığında veya cihazlar arasında frekans çakışmaları meydana geldiğinde, jitter değerleri artabilir. Bu durumları önlemek için QoS mekanizmaları, cihazlar arasındaki iletişim yollarını yeniden düzenleyerek jitter'ı minimize eder. Jitter yönetiminin başarılı bir şekilde uygulanması, IoT ağlarının güvenilirliğini ve verimliliğini artırır.

3.3.3. Veri kaybı yönetimi

IoT uygulamalarında veri kaybı, özellikle kritik görevlerde ciddi sonuçlara yol açabilmektedir. Örneğin, bir sağlık izleme cihazından alınan verilerin kaybolması, yanlış teşhislere ve tedavi hatalarına neden olabilmektedir (Islam et al., 2015). Veri kaybını önlemek için QoS, ağ kaynaklarının doğru tahsis edilmesini ve veri akışlarının sürekli olarak izlenmesini sağlar.

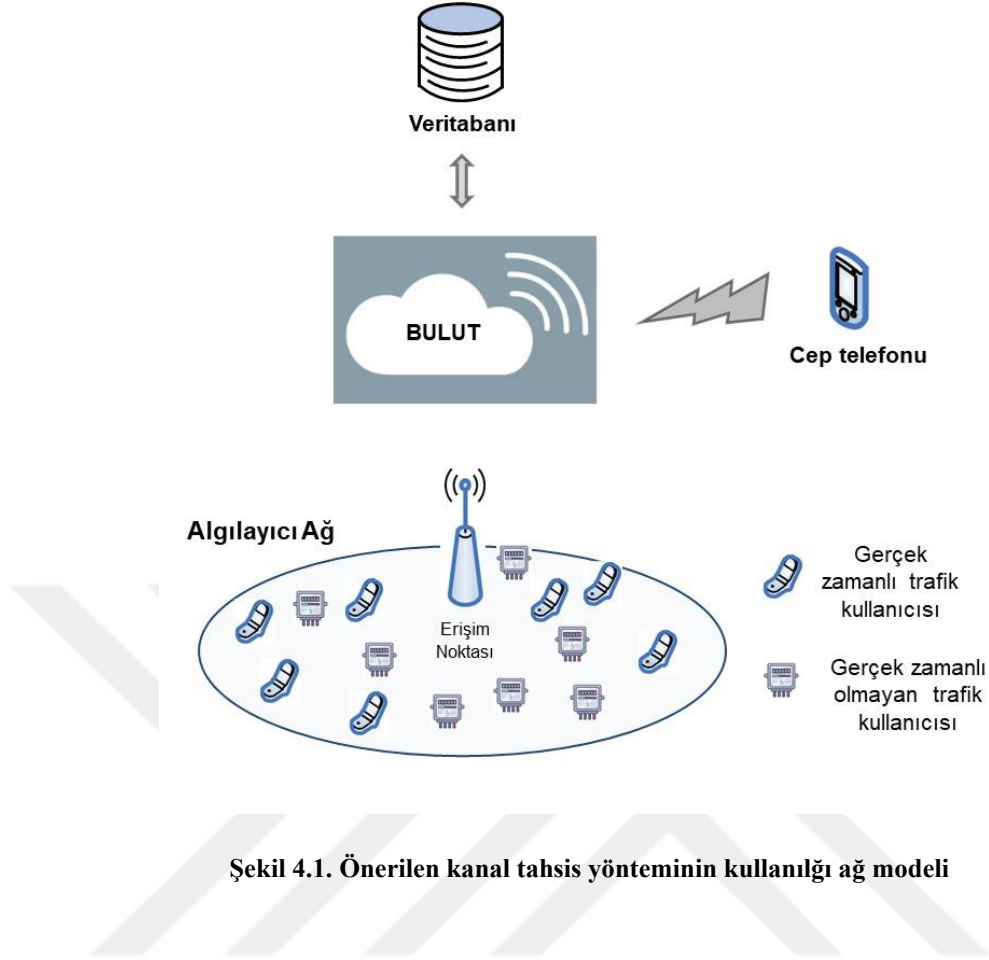
Veri kaybı yönetiminde kullanılan mekanizmalardan biri, yeniden iletim teknikleridir. Bu teknikler, kaybolan veri paketlerinin algılanmasını ve yeniden gönderilmesini sağlar. Ancak, bu süreç ek gecikmelere neden olabilir, bu yüzden yalnızca gerekli durumlarda kullanılması önerilir. Bunun yerine, hata düzeltme kodları gibi ileri düzey çözümler, veri kaybını minimum düzeye indirmek için daha verimli bir yöntem sunar.

Ayrıca, veri kaybının önlenmesi için ağ üzerindeki trafik yükü dengelenmelidir. Yoğun trafik, cihazlar arasında çakışmalara ve veri kaybına neden olabilir. QoS desteği, ağ trafiğini izleyerek cihazlar arasında dinamik yük dengeleme gerçekleştirir. Bu hem veri kaybını önler hem de ağın genel başarımını artırır.

4. GELİŞTİRİLEN DİNAMİK KANAL TAHSİS YÖNTEMİ

IoT ağ yapısı, genel olarak dört temel katmandan oluşur: (1) Fiziksel Katman (Physical Layer): Bu katman, fiziksel ortamda veri işlemlerini gerçekleştiren algılayıcılar, RFID cihazları, kameralar ve diğer algılayıcı birimlerden oluşur. Çevresel bilgiler (örneğin; sıcaklık, nem ve hareket) bu katmanda algılanır ve ağa iletilir. (2) Ortam Erişim Kontrol Katmanı (Medium Access Control Layer): Fiziksel katmanından gelen verilerin erişim noktalarına veya veri toplama (Data Acquisition) birimlerine iletimini sağlayan katmandır. Verilerin etkili ve güvenilir bir şekilde iletilmesi için veri iletim protokollerini ve ağ erişim mekanizmalarını içerir. (3) Ağ Katmanı (Network Layer): Bu katman, algılayıcı ağdaki erişim noktalarının daha geniş bir ağa veya internete bağlanmasını sağlar. Verilerin ağ boyunca doğru bir şekilde yönlendirilmesi ve iletilmesi işlemlerini gerçekleştirir. (4) Uygulama Katmanı (Application Layer): IoT sistemlerinden elde edilen verilerin işlendiği ve kullanıcılarla etkileşime geçtiği katmandır.

Genellikle bu dört katmanlı mimari yapıdan oluşan IoT ağları, düzenli, etkili ve ölçeklenebilir bir şekilde çalışmayı sağlar ve farklı cihazların uyumlu bir şekilde veri toplamasına, paylaşmasına ve analiz etmesine olanak tanır. Bu tez çalışmasında önerilen kanal tahsis algoritmasının kullanıldığı örnek bir IoT ağ yapısı Şekil 4.1’de görülmektedir.



Şekil 4.1. Önerilen kanal tahsis yönteminin kullanıldığı ağ modeli

Örnek ağ modelinde, merkezi ağ topolojisine sahip bir algılayıcı ağ yapısı yer alır. Bu algılayıcı ağ, bir erişim noktası ve sonlu sayıda algılayıcı düğümden meydana gelmektedir. Ağda bulunan bu sonlu sayıda algılayıcı düğümün iletişim alanına tekdüze dağılımla rassal olarak konumlandırıldığı varsayılmıştır. Algılayıcı düğümler, gerçek zamanlı (GZ) ve gerçek zamanlı olmayan (GZO) iki farklı veri trafik türü kullanmaktadır. Algılayıcı düğümlerin ürettikleri veri trafik türü, ağdaki tüm düğümlere homojen bir şekilde (düzgün dağılımla) dağıtılmıştır. Önerilen kanal tahsis yönteminin kullanıldığı IoT ağ yapısında heterojen veri trafiklerini kullanan tüm IoT düğümleri ortak kanala erişim için eşit önceliğe sahiptir, ancak kullandıkları trafik türüne göre, farklı sayıda kanal istekleri vardır. IoT düğümleri, fiziksel katmandan bir algılayıcı veri paketi geldiğinde bu paketi erişim noktasına iletmek için öncelikle kontrol kanalını kullanırlar. Kontrol kanalı aracılığıyla, iletecekleri veri trafiğinin türüne uygun olarak erişim noktasından kanal talebinde bulunurlar. Erişim noktası, o anda sistemde yeterli sayıda kanal varsa, bu algılayıcı düğüm için ilgili kanalları tahsis ederek, düğümü bilgilendirir. Algılayıcı düğüm de kendisine tahsis edilen kanalları

kullanarak, paket iletimini gerçekleştirir. Eğer istekte bulunan IoT düğümü için sistemde yeterli sayıda kanal yoksa, bu düğüm için çağrı bloke durumu oluşur. Bu durumda düğüm, üstel dağılımla rassal olarak üretilen bir süre sonunda tekrar erişim noktasından kanal isteğinde bulunur. Şekil 4.2’de önerilen kanal tahsis yaklaşımının pseudo kodu görülmektedir.

Kanal Tahsis Algoritması

```

1: begin
2:    $N=12, N_{NRT}=10, N_{RT}=10, N_{C,NRT}=1, N_{C,RT}=3$ 
3:    $\lambda_{NRT}, \lambda_{RT}, \mu_{NRT}=2.0, \mu_{RT}=1.2$ 
4:   düğümleri ağ alanına tek düze dağılım ile rassal dağıt
5:   while (true) do
6:     düğümlerden yeni kanal isteği bekle
7:     if (yeni kanal isteği) then
8:       istekte bulunan IoT düğüm tipini ve
9:       bu düğüm için gerekli kanal sayısını belirle
10:      if (yeterli sayıda kanal varsa) then
11:        bu düğüm için gerekli sayıda kanal tahsis et
12:        bu düğüm için rassal bir servis zamanı üret
13:      else
14:        çağrı bloke sayısını 1 arttır
15:        bu düğüm için rassal bir varış zamanı üret
16:        tüm başarımların metriklerini güncelle
17:      end if
18:    end if
19:    else if (yeni bir servis tamamlandı) then
20:      servisi tamamlanan IoT düğümünün kullandığı
21:      kanalları boşalt
22:      servisi tamamlanan IoT düğümü için yeni bir varış
23:      zamanı üret
24:      tüm başarımların metriklerini güncelle
25:    end else if
26:  end while
27: end

```

Şekil 4.2. Kanal tahsis yönteminin pseudo kodu

Önerilen kanal tahsis yönteminde uygulama python dili kullanılarak oluşturulmuş benzetime ait verilerin girdileri Şekil 4.3’de görülmektedir.

```
9 import numpy as np
10 import matplotlib.pyplot as plt
11
12 # Total number of channels
13 N = 12
14
15 #Total number of Real-time Users
16 Nu_rt = 10
17
18 # Total number of Non-Real-time Users
19 Nu_nrt = 10
20
21 # The number of channels used for Non_Real_Time Users
22 Nc_nrt = 1
23
24 # The number of channels used for Real_Time Users
25 Nc_rt = 3
26 Nc_nrt_Max = (N // Nc_nrt) * Nc_nrt
27 Nc_rt_Max = (N // Nc_rt) * Nc_rt
28 print(f"Nc_nrt_Max = {Nc_nrt_Max}")
29 print(f"Nc_rt_Max = {Nc_rt_Max}")
```

Şekil 4.3. Kanal tahsis yöntemi uygulama kodu giriş değerleri

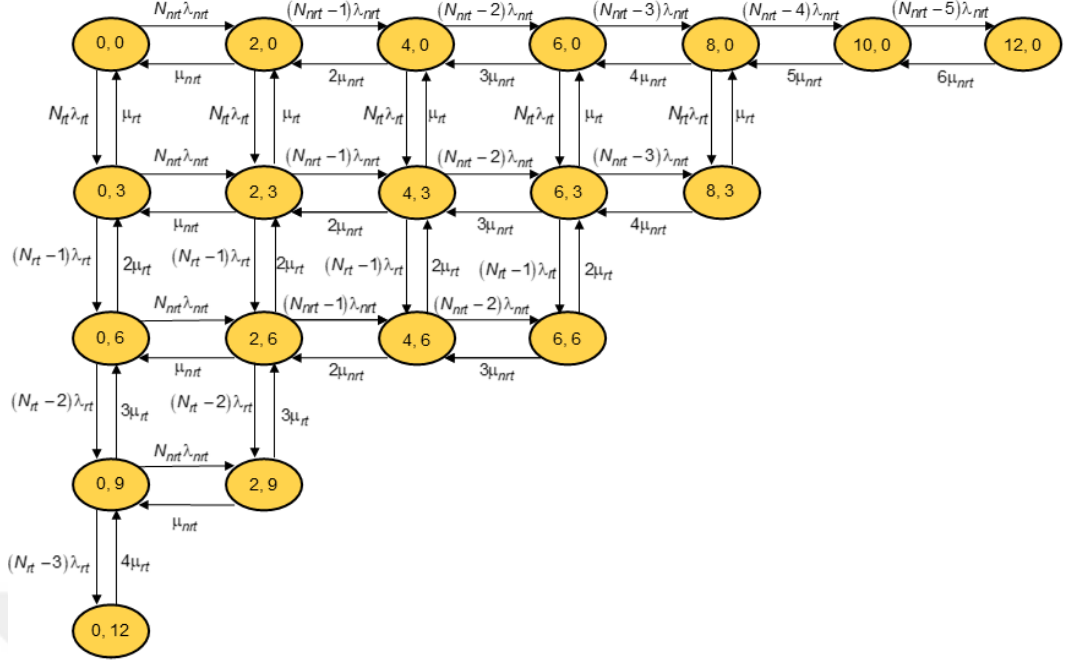
4.1. Önerilen Kanal Tahsis Yönteminin Analitik Modeli

Tez çalışmasında önerilen tahsis yöntemi, 2-boyutlu sürekli bir Markov zinciri kullanılarak analitik olarak modellenmiştir. Bir Markov Zinciri modelinde sistemin başlangıç durumu genellikle tüm kanalların boş olduğu varsayımıyla ele alınır; bu nedenle başlangıç oran matrisi rassal değerlerle yüklenir ve normalize edilir. Bu işlem, sistem kararlı hale geldikten sonra başlangıç durumları önemli olmadığından dolayı gerçekleştirilir. Sistem, başlangıç değerleri ve parametreleri girildikten sonra düğümler için yeni çağrılara açık hale gelir.

Markov zinciri modelinde, sistemin her bir durumda bulunma olasılığı hesaplamaktır. Bu hesaplama iteratif bir süreçle gerçekleştirilir. Örneğin, i durumundaki olasılık $P(i)$, bir önceki durum olan $(i-1)$ 'de bulunma olasılığı $P(i-1)$ ve $(i-1)$ durumundan i 'ye geçiş olasılığı kullanılarak belirlenir. Bu süreç, belirli bir sistem kapasitesi N için tüm durumları kapsar ve sistem kararlı hale gelinceye kadar devam eder. Sistem kararlı hale geldikten sonra durum olasılıkları değişmez (Kleinrock, 1975).

Bu hesaplamaları gerçekleştirmek için bir döngü kullanılır. Döngü her durumda çağrı başlatma ve tamamlama olasılıklarını iteratif olarak işler. Markov modellerinde olasılık dağılımının toplamı 1 olmalıdır. Bu nedenle, tüm olasılıklar normalizasyon adı verilen bir adımda yeniden ölçeklendirilir. Bu adım, bütün $P(i)$ toplam değerlerini 1 yaparak hesaplanır.

Şekil 4.1'de önerilen modelin gerçek-zamanlı olmayan kullanıcıların 1 kanal kullandığı ($N_{C_NRT} = 1$), gerçek-zamanlı kullanıcıların 3 kanal kullandığı ($N_{C_RT} = 3$) ve toplam kanal sayısının 12 ($N = 12$) olduğu durum için Markov Zinciri modeli görülmektedir. Burada, N_{C_NRT} gerçek zamanlı olmayan (non-real-time) trafikler için gerekli kanal sayısını, N_{C_RT} gerçek zamanlı trafikler için gerekli kanal sayısını ve N ise sistemdeki toplam kanal sayısını göstermektedir. Markov zinciri modelinde her bir durum, (i, j) ikilileri şeklinde ifade edilir. Bu (i, j) sıralı ikilileri, sistemin durumlarını ayrıntılı bir şekilde tanımlamak için kullanılır. Burada, ikilinin bileşenleri sırasıyla, sistemde kullanılan gerçek-zamanlı olmayan ve gerçek-zamanlı kanal sayılarını temsil etmektedir.



Şekil 4.4. Önerilen dinamik kanal tahsis yönteminin $N_{nrt}=1$, $N_{rt}=3$ ve $N=12$ için 2-boyutlu Markov Zinciri modeli

Tablo 1.'de Markov zinciri modelinde kullanılan semboller ve kısaltmalar sunulmuştur.

Tablo 1. Önerilen dinamik kanal tahsis tablosunun Markov zinciri modelinde kullanılan semboller ve kısaltmalar

Sembol	Açıklaması
N	Ağda kullanılan toplam kanal sayısı
N_{NRT}	Ağdaki gerçek zamanlı olmayan veri trafiği kullanıcı sayısı
N_{RT}	Ağdaki gerçek zamanlı veri trafiği kullanıcı sayısı
$N_{C,NRT}$	Ağda kullanılan gerçek zamanlı olmayan veri trafiği kanal sayısı
$N_{C,RT}$	Ağda kullanılan gerçek zamanlı veri trafiği kanal sayısı
λ_{NRT}	Gerçek zamanlı olmayan kullanıcılara ait varış oranı
λ_{RT}	Gerçek zamanlı kullanıcılara ait varış oranı
λ_T	Ağdaki kullanıcılara ait toplam varış oranı
μ_{NRT}	Gerçek zamanlı olmayan kullanıcılara ait servis oranı
μ_{RT}	Gerçek zamanlı kullanıcılara ait servis oranı

Önerilen dinamik kanal tahsis yönteminde kullanılan Markov zinciri modelinin durum uzayı Denklem (4.1) ile tanımlanmıştır.

$$\Omega = \{(i, j) | 0 \leq i, j \leq N, i + j \leq N\} \quad (4.1)$$

Markov zinciri modelinde (i, j) durumuna ait denge eşitlikleri Denklem (4.2)'de verilmiştir.

$$P(0,0) = \left(\frac{1}{(N_{NRT} \lambda_{NRT} + N_{RT} \lambda_{RT})} \right) \left(P(N_{C,NRT},0) \mu_{NRT} + P(0, N_{C,RT}) \mu_{RT} \right) \quad (4.2)$$

Önerilen Markov zinciri modelinde $i = N_{C,NRT,MAX}, j = 0$ durumuna ait denge eşitlikleri Denklem (4.3)'de verilmiştir.

$$P(N_{C,NRT,MAX},0) = \left(\frac{1}{(N_{C,NRT,MAX} / N_{C,NRT}) \mu_{NRT}} \right) \left(P(N_{C,NRT,MAX} - N_{C,NRT},0) \left((N_{C,NRT,MAX} - N_{C,NRT}) / N_{C,NRT} \right) \lambda_{NRT} \right) \quad (4.3)$$

Burada, $N_{C,NRT,MAX}$, gerçek zamanlı olmayan kullanıcılar tarafından kullanılacak en yüksek kanal sayısını göstermektedir. $N_{C,NRT,MAX}$ Denklem (4.4) ile elde edilir;

$$N_{C,NRT,MAX} = \lfloor N / N_{C,NRT} \rfloor N_{C,NRT} \quad (4.4)$$

Denklem (4.4) ile verilen eşitlikte $\lfloor \cdot \rfloor$ işlemi, aşağı yuvarlama fonksiyonunu göstermektedir. Markov zinciri modelinde $N_{C,NRT,MAX} + N_{C,NRT} > N$ durumunda denge eşitliği Denklem (4.5)'te verilmiştir.

$$P(0, N_{C,RT,MAX}) = \left(\frac{1}{(N_{C,RT,MAX} / N_{C,RT}) \mu_{RT}} \right) \left(P(0, N_{C,RT,MAX} - N_{C,RT}) \left((N_{C,RT,MAX} - N_{C,RT}) / N_{C,RT} \right) \lambda_{RT} \right) \quad (4.5)$$

Markov zinciri modelinde $N_{C,NRT,MAX} + N_{C,NRT} \leq N$ durumuna ait denge eşitliği Denklem (4.6)'da verilmiştir.

$$\begin{aligned}
P(0, N_{C,RT,MAX}) &= \left(1 / \left((N_{C,RT,MAX} / N_{C,RT}) \mu_{RT} + N_{NRT} \lambda_{NRT} \right) \right) \\
&\left(P(0, N_{C,RT,MAX} - N_{C,RT}) \left(N_{RT} - (N_{C,RT,MAX} - N_{C,RT}) / N_{C,RT} \right) \lambda_{RT} + \right. \\
&P(N_{C,NRT}, N_{C,RT,MAX}) \mu_{NRT} \left. \right)
\end{aligned} \tag{4.6}$$

Markov zinciri modelinde $N_{C,NRT} \leq i < N_{C,NRT,MAX}$ ve $i + N_{C,NRT} \leq N_{C,NRT,MAX}$ durumuna ait denge eşitliği Denklem (4.7)'da verilmiştir.

$$\begin{aligned}
P(i, 0) &= \left(1 / \left((N_{NRT} - i / N_{C,NRT}) \lambda_{NRT} + N_{RT} \lambda_{RT} + (i / N_{C,RT}) \mu_{NRT} \right) \right) \\
&\left(P(i - N_{C,NRT}, 0) \left(N_{NRT} - (i - N_{C,NRT}) / N_{C,NRT} \right) \lambda_{NRT} + \right. \\
&P(i + N_{C,NRT}, 0) \left((i + N_{C,NRT}) / N_{C,NRT} \right) \mu_{NRT} \left. \right)
\end{aligned} \tag{4.7}$$

Markov zinciri modelinde $N_{C,NRT} \leq i < N_{C,NRT,MAX}$ ve $i + N_{C,NRT} > N_{C,NRT,MAX}$ durumuna ait denge eşitliği Denklem (4.8)'da verilmiştir.

$$\begin{aligned}
P(i, 0) &= \left(1 / \left((N_{NRT} - i / N_{C,NRT}) \lambda_{NRT} + (i / N_{C,NRT}) \mu_{NRT} \right) \right) \\
&\left(P(i - N_{C,NRT}, 0) \left(N_{NRT} - (i - N_{C,NRT}) / N_{C,NRT} \right) \lambda_{NRT} + \right. \\
&P(i + N_{C,NRT}, 0) \left((i + N_{C,NRT}) / N_{C,NRT} \right) \mu_{NRT} \left. \right)
\end{aligned} \tag{4.8}$$

Markov zinciri modelinde $N_{C,RT} \leq i < N_{C,RT,MAX}$ durumuna ait denge eşitliği Denklem (4.9)'da verilmiştir.

$$\begin{aligned}
P(0, j) &= \left(1 / \left(N_{NRT} \lambda_{NRT} + (N_{RT} - j / N_{C,RT}) \lambda_{RT} \right) + (j / N_{C,RT}) \mu_{RT} \right) \\
&\left(P(0, j - N_{C,RT}) \left(N_{RT} - (j - N_{C,RT}) / N_{C,RT} \right) \lambda_{RT} + \right. \\
&P(0, j + N_{C,RT}) \left((j + N_{C,RT}) / N_{C,RT} \right) \mu_{RT} + P(N_{C,NRT}, j) \mu_{NRT} \left. \right)
\end{aligned} \tag{4.9}$$

Markov zinciri modelinde $N_{C,NRT} \leq i < N$, $N_{C,RT} \leq j \leq N$ ve $N - N_{C,NRT} \leq i + j \leq N$ durumuna ait denge eşitliği Denklem (4.10)'da verilmiştir.

$$\begin{aligned}
P(i, j) = & \left(1 / \left(\left(i / N_{C,NRT} \right) \mu_{NRT} + \left(j / N_{C,RT} \right) \mu_{RT} \right) \right) \\
& \left(P(i - N_{C,NRT}, j) \left(N_{NRT} - (i - N_{C,NRT}) / N_{C,NRT} \right) \lambda_{NRT} + \right. \\
& \left. P(i, j - N_{C,RT}) \left(N_{RT} - (j - N_{C,RT}) / N_{C,RT} \right) \lambda_{RT} \right)
\end{aligned} \tag{4.10}$$

Markov zinciri modelinde $N_{C,NRT} \leq i < N$, $N_{C,RT} \leq j < N$ ve $(i + j) \leq (N - N_{C,RT})$ durumuna ait denge eşitliği Denklem (4.11)'de verilmiştir.

$$\begin{aligned}
P(i, j) = & \left(1 / \left(\left(N_{NRT} - i / N_{C,NRT} \right) \lambda_{NRT} + \left(N_{RT} - j / N_{C,RT} \right) \lambda_{RT} + \right. \right. \\
& \left. \left(i / N_{C,NRT} \right) \mu_{NRT} + \left(j / N_{C,RT} \right) \mu_{RT} \right) \left(P(i - N_{C,NRT}, j) \left(N_{NRT} - \right. \right. \\
& \left. \left(i / N_{C,NRT} - 1 \right) \right) \lambda_{NRT} + P(i, j - N_{C,RT}) \left(N_{RT} - \left(j / N_{C,RT} - 1 \right) \right) \lambda_{RT} + \\
& P(i, j + N_{C,RT}) \left(\left(j + N_{C,RT} \right) / N_{C,RT} \right) \mu_{RT} + \\
& \left. P(i + N_{C,NRT}, j) \left(\left(i + N_{C,NRT} \right) / N_{C,NRT} \right) \mu_{NRT} \right)
\end{aligned} \tag{4.11}$$

Markov zinciri modelinde $N_{C,NRT} \leq i < N$, $N_{C,RT} \leq j < N$ ve $N - N_{C,RT} < (i + j) \leq N - N_{C,NRT}$ durumuna ait denge eşitliği Denklem (4.12)'de verilmiştir.

$$\begin{aligned}
P(i, j) = & \left(1 / \left(\left(N_{NRT} - i / N_{C,NRT} \right) \lambda_{NRT} + \left(j / N_{C,RT} \right) \mu_{RT} + \right. \right. \\
& \left. \left(j / N_{C,RT} \right) \mu_{RT} \right) \left(P(i - N_{C,NRT}, j) \left(N_{NRT} - (i - N_{C,NRT}) / N_{C,NRT} \right) \lambda_{NRT} + \right. \\
& \left. + P(i + N_{C,NRT}, j) \left(\left(i + N_{C,NRT} \right) / N_{C,NRT} \right) \mu_{NRT} + \right. \\
& \left. P(i, j - N_{C,RT}) \left(N_{RT} - (j - N_{C,RT}) / N_{C,RT} \right) \lambda_{RT} \right)
\end{aligned} \tag{4.12}$$

Markov zinciri modelinde tüm durumlara ait olasılıkların toplamı 1 olduğundan, bu durum Denklem (4.13)'te verilen eşitlik ile ifade edilir.

$$\sum_{i=0}^N \sum_{j=0}^N P(i, j) = 1 \tag{4.13}$$

Sürekli zamanlı Markov zincirinin kararlı durum olasılık dağılımı, durum geçiş oranı matrisini kullanarak ve Gauss-Seidel yöntemi kullanarak kolayca elde edilebilir (Stewart, 1994).

4.2. Önerilen Dinamik Kanal Tahsis Yönteminin Benzetimi

Önerilen dinamik kanal tahsis yönteminin benzetimi, Python programlama dili kullanılarak ayrık olay benzetim (Discrete-event simulation) tekniğiyle gerçekleştirilmiştir. Python, ayrık olay benzetim kütüphanelerine (SimPy) sahiptir. Ancak, bu tez çalışmalarında benzetim için hazır kütüphane kodları kullanılmamıştır. Önerilen dinamik kanal tahsis yönteminin benzetiminde örnek ağ senaryosu göz önünde bulundurularak, bu ağa ait parametreler ve diğer sistem değişkenleri kullanılmıştır. Sistem benzetiminde Tablo 2’de sunulan parametreler kullanılmıştır.

Tablo 2. Önerilen Kanal Tahsis Yöntemin Benzetim Modelinde Kullanılan Parametreler

	Parametre	Değer
Sistem	Kanal Sayısı (N)	12
	Kanal sayısı ($N_{C,RT}$)	3
Gerçek zamanlı kullanıcı	Kullanıcı sayısı (N_{RT})	10
	Mean Arrival Rate (λ_{RT})	0,1 – 1 çağrı/saniye
	Mean Service Rate (μ_{RT})	1,2 çağrı/saniye
	Kanal sayısı ($N_{C,NRT}$)	1
Gerçek zamanlı olmayan kullanıcı	Kullanıcı sayısı (N_{NRT})	10
	Mean Arrival Rate (λ_{NRT})	0,1 – 1 çağrı/saniye
	Mean Service Rate (μ_{NRT})	2 çağrı/saniye

Benzetim modelinde kanal tahsisi, ağdaki kullanıcıların veri iletimlerini gerçekleştirebilmeleri için uygun kanalların dinamik olarak atanmasını sağlar. Burada kanal için frekans veya zaman dilimi kullanılabilir. Dinamik kanal tahsisi ağda kullanılan kanalların yoğunluğunu ve trafik koşullarını sürekli olarak izler ve kanal isteğinde bulunan IoT düğümü için gerekli sayıda kanal tahsisini gerçekleştirir. Kanal tahsisi sonrasında IoT düğümü kontrol kanalı üzerinden bilgilendirilir ve veri iletimi başlatılır. Kanal tahsis edilen IoT düğümüne veri iletim süresi (servis süresi) üstel dağılımla rassal olarak üretilir. Servis süresi sona eren IoT düğümünün veri iletimi tamamlanır ve bu düğümüne ait kanallar diğer düğümlerin kullanımı için boşa çıkartılır. Sistemden hizmet alımı sonlanan bir IoT düğümü, üstel dağılımla üretilen bir rassal süre sonunda tekrar servis isteğinde bulunur.

Benzetim modelinde, önerilen dinamik kanal tahsis yönteminin başarımını değerlendirmek amacıyla, analitik modelde kullanılan başarım ölçütleri elde edilerek her iki modelden elde edilen sonuçlar karşılaştırılmıştır.

4.3. Bulgular

Geliştirilen dinamik kanal tahsis yöntemine yönelik analitik ve benzetim çalışmaları, IoT ağlarının heterojen veri trafiğini daha etkin ve verimli bir şekilde yönetebildiğini ortaya koymaktadır. Özellikle gerçek zamanlı ve gerçek zamanlı olmayan veri trafiklerini kullanan bu yöntem, sınırlı kaynakların optimum düzeyde kullanılmasına olanak tanımaktadır. Markov zinciri modeli ile gerçekleştirilen analitik çözümler, sistemin kararlı durum olasılıklarının hesaplanmasını ve bu olasılıkların başarım metriklerine dönüştürülmesini mümkün kılmaktadır (Kleinrock, 1975).

Benzetim sonuçları, önerilen yöntemin dinamik kanal tahsisi sırasında IoT düğümlerinin çağrı bloke oranını ve kanal kullanım etkinliğini başarılı bir şekilde dengelediğini göstermektedir. Özellikle, sistemde mevcut kanal sayısının sınırlı olduğu durumlar için önerilen yaklaşımın kanal bloke oranını minimum seviyede tuttuğu ve IoT düğümlerinin veri iletim sürecini optimize ettiği görülmüştür. Bu durum hem analitik modelden hem de benzetim sonuçlarından elde edilen verilere dayanmaktadır.

Çalışmada kullanılan Python tabanlı benzetim modeli, dinamik kanal tahsisi sürecini sonlu sayıda kullanıcı sayısı ile gerçek ağ modeline yakın bir şekilde modelleyerek analitik çözüm ile benzer sonuçlar üretmiştir. Analitik model ve benzetim model sonuçları karşılaştırıldığında, her iki yöntemin tutarlı sonuçlar vermesi önerilen yöntemin başarımının doğrulandığını ortaya koymaktadır. Bu durum, yöntemin güvenilirliğini artırmakta ve IoT ağları için uygulanabilirliğini kanıtlamaktadır.

5. GELİŞTİRİLEN DİNAMİK KANAL TAHSİS YÖNTEMİNİN ÖRNEK BİR İoT AĞ SENARYOSUNDA UYGULANMASI VE BAŞARIM DEĞERLENDİRMESİ

Bu bölümde, önerilen dinamik kanal tahsis yönteminin başarımını analiz etmek için örnek bir IoT ağ senaryosu oluşturulmuş ve bu ağ üzerinden başarım metrikleri elde edilmiştir. Başarım analizinde dört temel metrik benimsenmiştir: (1) kanal kullanım oranı (utilization), (2) çağrı bloke oranı (call blocking probability, CBP), (3) çağrı tamamlama oranı (call completion rate, CCR) ve (4) iş çıkarma oranı (throughput). Başarım metrikleri ağdaki yükün artan değerleri için elde edilmiştir ve toplam yük, ağdaki IoT düğümlerine düzgün ve rassal olarak dağıtılmıştır. Alt bölümlerde, önerilen kanal tahsis yönteminin kullanıldığı ağ senaryosundan elde edilen başarım sonuçları ayrıntılı olarak sunulmaktadır.

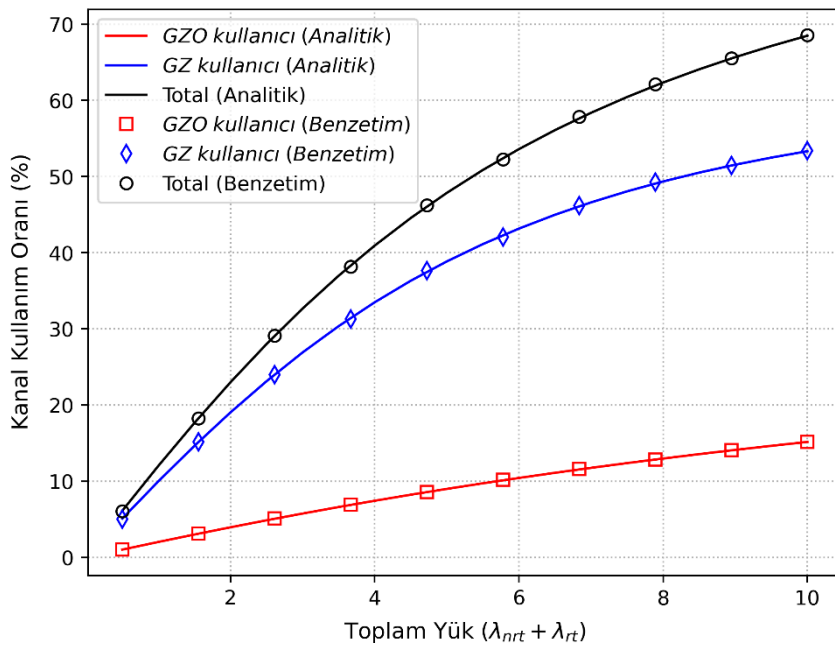
5.1. Kanal kullanım oranı (Utilization)

Kanal kullanım oranı, bir ağdaki kaynakların (haberleşme kanallarının) ne ölçüde verimli kullanıldığını ifade eder ve genellikle bir kanalın veya ağın toplam kapasitesinin ne kadarının veri iletimi için ayrıldığını gösterir (Atmaca, 2025). Diğer bir ifadeyle, kullanılan kanal sayısının ağdaki toplam kanal sayısına oranı olarak tanımlanabilir (Rappaport, 2002). Önerilen ağ modelindeki gerçek zamanlı ve gerçek zamanlı olmayan kullanıcılara ait kanal kullanım oranları sırasıyla Denklem (5.1) ve Denklem (5.2)'de verilmiştir.

$$U_{RT} = \sum_i^N \sum_{j, i+j \leq N}^N P(i, j) \frac{j}{N} \quad (5.1)$$

$$U_{NRT} = \sum_i^N \sum_{j, i+j \leq N}^N P(i, j) \frac{i}{N} \quad (5.2)$$

Şekil 5.1’de gerçek zamanlı ve gerçek zamanlı olmayan düğümlerin (GZ0 ve GZ) kanal kullanım oranlarının (utilization) toplam yük ($\lambda_{RT} + \lambda_{NRT}$) ile ilişkisi hem analitik sonuçlar hem de benzetim sonuçları ile birlikte sunulmaktadır. Grafikler incelendiğinde, gerçek zamanlı olmayan düğümlerin kanal kullanım oranlarının bütün yük değerleri için gerçek zamanlı düğümlere göre düşük olduğu görülmektedir. Bunun nedeni gerçek zamanlı düğümlerin veri trafiği türünden dolayı daha fazla kanal kullanmalarıdır. Ayrıca, gerçek zamanlı olmayan düğümlerin kanal kullanım oranlarının artan yük ile gerçek zamanlı düğümlere göre daha yavaş arttığı görülmektedir. Gerçek zamanlı olmayan düğümlerin kanal kullanım oranları toplam yük 2 iken %3,95, toplam yük 5 iken %8,99 ve toplam yük 10 iken %15,16 olduğu görülmektedir. Gerçek zamanlı düğümlere ait kanal kullanı oranları ise toplam yük 2 iken %19,03, toplam yük 5 iken %38,85 ve toplam yük 10 iken %53,31 olarak elde edilmiştir. Bu sonuçlar incelendiğinde gerçek zamanlı düğümlerin sistem kaynaklarını daha yüksek oranda kullandığı ve toplam ağ kapasitesinden büyük pay aldığı anlaşılmaktadır.



Şekil 5.1. Önerilen dinamik kanal tahsis algoritmasının kullanıldığı ağ modelinde kanal kullanımı oranları

5.2. Çağrı tıkanma (bloke) olasılığı (CBP)

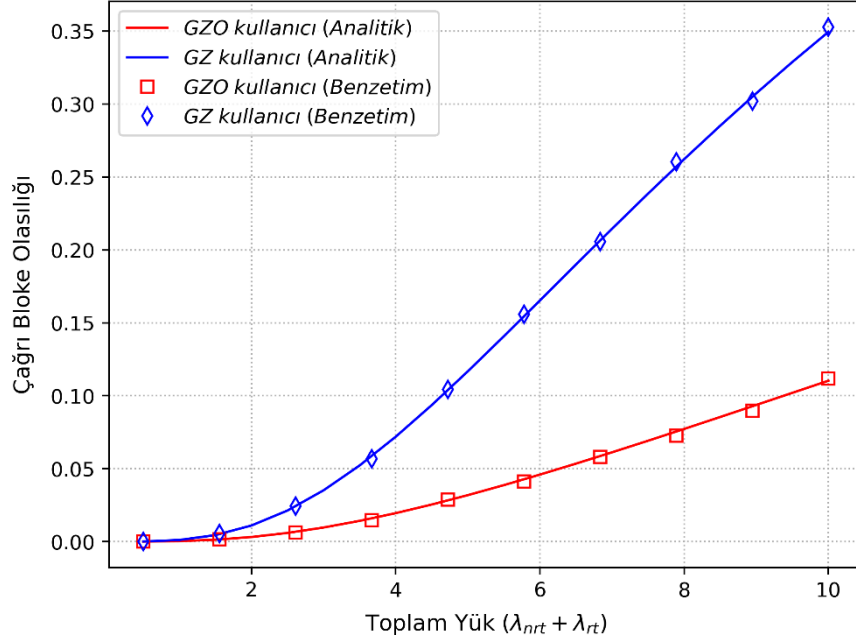
Çağrı bloke olasılığı (Call Blocking Probability), ağda yeterli kaynak bulunmaması nedeniyle hizmet sağlanamayan ve engellenen bağlantı isteklerinin toplam bağlantı isteği sayısına oranıdır (Atmaca, 2025; Kumar & Misra, 2014). Bu çalışmada sunulan örnek ağ modelindeki gerçek zamanlı ve gerçek zamanlı olmayan düğümlere ait çağrı bloke olasılıkları sırasıyla Denklem (5.3) ve Denklem (5.4)'te sunulmuştur.

$$P_{B,RT} = \frac{\sum_i^N \sum_{j, N-N_{C,RT} < i+j \leq N}^N P(i, j) (N_{U,RT} - j / N_{C,RT}) \lambda_{RT}}{\sum_i^N \sum_{j, i+j \leq N}^N P(i, j) (N_{U,RT} - j / N_{C,RT}) \lambda_{RT}} \quad (5.3)$$

$$P_{B,NRT} = \frac{\sum_i^N \sum_{j, N-N_{C,NRT} < i+j \leq N}^N P(i, j) (N_{U,NRT} - i / N_{C,NRT}) \lambda_{NRT}}{\sum_i^N \sum_{j, i+j \leq N}^N P(i, j) (N_{U,NRT} - i / N_{C,NRT}) \lambda_{NRT}} \quad (5.4)$$

Şekil 5.2'de verilen grafikte gerçek zamanlı ve gerçek zamanlı olmayan düğümlerin (GZ0 ve GZ) çağrı bloke olasılıklarının toplam yük ($\lambda_{RT} + \lambda_{NRT}$) ile ilişkisi hem analitik sonuçlar hem de benzetim sonuçları ile birlikte sunulmaktadır. Grafiklerden de görüleceği gibi her iki kullanıcı türü için de ağdaki toplam yük arttığında çağrı bloke olasılıkları artmaktadır. Ancak gerçek zamanlı olmayan düğümlerin çağrı bloke olasılıklarının gerçek zamanlı düğümlere göre, aynı yük altında daha düşük olduğu görülmektedir. Bunun sebebi gerçek zamanlı düğümlerin kaynak ihtiyacının gerçek zamanlı olmayan düğümlere göre daha yüksek olması ve sistemde bu kullanıcıların ihtiyaç duyduğu yeterli kaynak sayısının karşılanmasının daha zor olmasıdır. Diğer bir ifade ile, gerçek zamanlı olmayan düğümlerin kaynak ihtiyacı daha düşük olduğundan sistem kaynaklarından yararlanma olasılıkları daha yüksektir. Örneğin gerçek zamanlı olmayan düğümlerin için toplam yük değeri 2 iken çağrı bloke olasılığı 0,0031, toplam yük değeri 5 iken çağrı bloke olasılığı 0,0318 ve toplam yük değeri 10 iken bu değer 0,11 olmaktadır. Gerçek zamanlı düğümler için ise, toplam yük değeri 2 iken çağrı bloke olasılığı 0,011, toplam yük değeri 5 iken çağrı bloke olasılığı 0,116 ve toplam yük değeri 10 iken bu değer 0,349 olmaktadır. Grafiklerdeki örtüşen benzetim ve

analitik model sonuçlarından da görüleceği üzere, gerçek zamanlı düğümlere ait çağrı bloke olasılıkları artan toplam yük ile birlikte daha hızlı artmaktadır.



Şekil 5.2. Önerilen dinamik kanal tahsis algoritmasının kullanıldığı ağ modelinde çağrı bloke olasılıkları

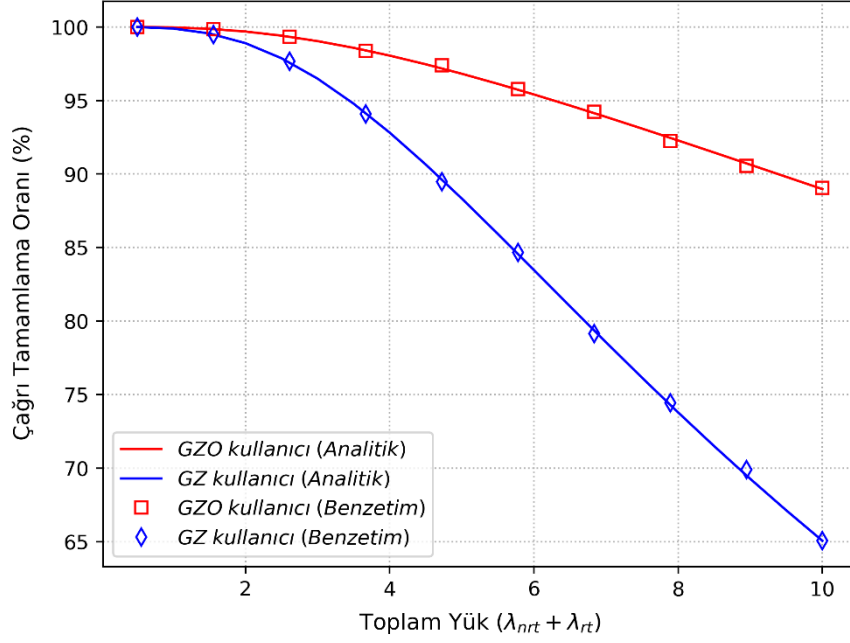
5.3. Çağrı tamamlama oranı (CCR)

Çağrı tamamlama oranı, ağda hizmet sunulan bağlantı sayısının toplam bağlantı isteği sayısına oranı olarak tanımlanır (Chen et al., 2021). Bu çalışmadaki örnek ağ modelinde gerçek zamanlı ve gerçek zamanlı olmayan düğümlere ait çağrı tamamlama oranlarına ait eşitlikler sırasıyla Denklem (5.5) ve Denklem (5.6)'da sunulmuştur.

$$CCR_{RT} = \frac{\sum_i^N \sum_{j,i+j \leq N}^N P(i, j) (j / N_{C,RT}) \mu_{RT}}{\sum_i^N \sum_{j,i+j \leq N}^N P(i, j) (N_{U,RT} - j / N_{C,RT}) \lambda_{RT}} \quad (5.5)$$

$$CCR_{NRT} = \frac{\sum_i^N \sum_{j,i+j \leq N}^N P(i, j) (i / N_{C,NRT}) \mu_{NRT}}{\sum_i^N \sum_{j,i+j \leq N}^N P(i, j) (N_{U,NRT} - i / N_{C,NRT}) \lambda_{NRT}} \quad (5.6)$$

Şekil 5.3'te sunulan grafikte, örnek ağ modelinde bulunan gerçek zamanlı ve gerçek zamanlı olmayan düğümlerin (GZ0 ve GZ) çağrı tamamlama oranlarının toplam yük ($\lambda_{RT} + \lambda_{NRT}$) ile ilişkisi hem analitik sonuçlar hem de benzetim sonuçları ile birlikte gösterilmiştir. Grafik genel olarak, sistemdeki toplam yük arttıkça hem gerçek zamanlı düğümler için hem de gerçek zamanlı olmayan düğümler için çağrı tamamlama oranının düştüğünü göstermektedir. Bu durum, sınırlı sistem kaynaklarının toplam yükün artmasıyla birlikte yetersiz hale geldiğini ve daha fazla kullanıcıya hizmet veremediğini göstermektedir. Ayrıca, gerçek zamanlı olmayan düğümlere ait çağrı tamamlama oranı, gerçek zamanlı düğümlere göre daha yüksektir. Bunun nedeni, bu düğümlerin daha az sınırlı sistem kaynaklarından yararlanması ve hizmetlerini tamamlamasıdır. Gerçek zamanlı düğümlerin kaynak ihtiyacı daha yüksek olduğundan ve sistem kaynaklarından yararlanma olasılıkları daha düşüktür ve bununla birlikte çağrı tamamlama oranları da düşüktür. Grafiklerden gösterilen sonuçlara göre, gerçek zamanlı olmayan düğümler için toplam yük değeri 2 iken çağrı tamamlama oranı %99,68, toplam yük değeri 5 iken çağrı tamamlama oranı %96,81 ve toplam yük değeri 10 iken çağrı tamamlama oranı %88,97 olmaktadır. Gerçek zamanlı düğümler için ise, toplam yük değeri 2 iken çağrı tamamlama oranı %98,89, toplam yük değeri 5 iken çağrı tamamlama oranı %88,32 ve toplam yük değeri 10 iken çağrı tamamlama oranı %65,05 olmaktadır. Bu sonuçlara göre, gerçek zamanlı düğümlere ait çağrı tamamlama oranları artan yük ile birlikte gerçek zamanlı olmayan düğümlere göre daha hızla düşmektedir.



Şekil 5.3. Önerilen dinamik kanal tahsis algoritmasının kullanıldığı ağ modelinde çağrı tamamlama oranları

5.4. Throughput (iş çıkarma oranı)

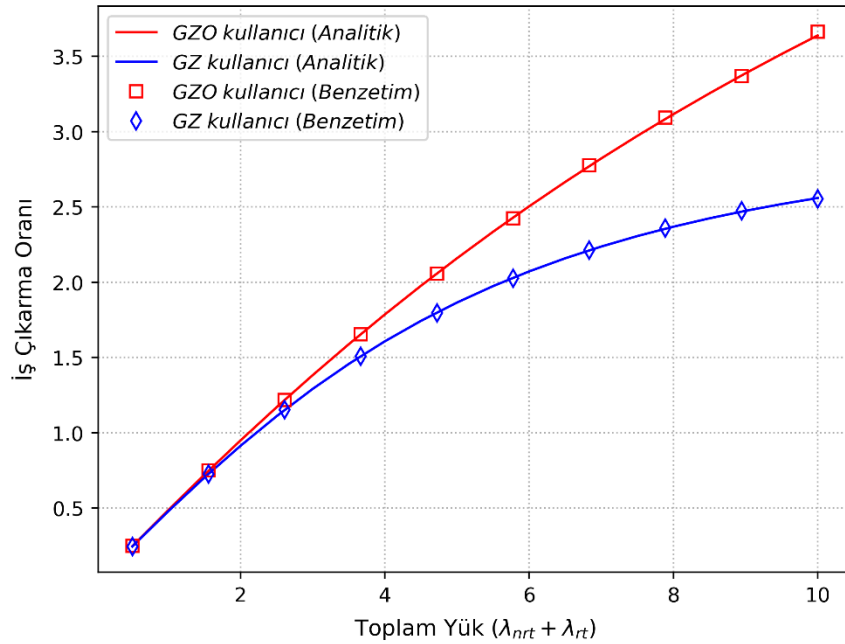
Throughput (iş çıkarma oranı), ağda birim zamanda tamamlanan toplam bağlantı sayısı olarak tanımlanır (Tang et al., 2020). Önerilen ağ modelinde Markov zinciri modeli kullanılarak gerçek zamanlı ve gerçek zamanlı olmayan düğümlere ait iş çıkarma oranları sırasıyla Denklem (5.7) ve Denklem (5.8)'de verilmiştir.

$$\eta_{RT} = \sum_i^N \sum_{j, i+j \leq N}^N P(i, j) (j / N_{C,RT}) \mu_{RT} \quad (5.7)$$

$$\eta_{NRT} = \sum_i^N \sum_{j, i+j \leq N}^N P(i, j) (i / N_{C,NRT}) \mu_{NRT} \quad (5.8)$$

Şekil 5.4'te sunulan grafikte, örnek ağ modelinde bulunan gerçek zamanlı ve gerçek zamanlı olmayan düğümlere ait toplam iş çıkarma oranları, artan yük ($\lambda_{RT} + \lambda_{NRT}$) değerleri ile sunulmuştur. Grafiklerden de görüleceği gibi, ağdaki toplam yük arttıkça hem gerçek zamanlı düğümler için hem de gerçek zamanlı olmayan düğümler için iş çıkarma oranının arttığı görülmektedir. Grafiklerde sunulan sonuçlara göre, toplam

yük değeri 5 olana kadar gerçek zamanlı ve gerçek zamanlı olmayan düğümlerin iş çıkarma oranları arasında anlamlı bir fark oluşmazken, toplam yük değeri 5'in üzerine çıktığında bu fark belirgin hale gelmeye başlamıştır. Bunun nedeni, artan yük değerleri ile birlikte, sınırlı sistem kaynaklarının azalması ve kaynak ihtiyacı daha fazla olan gerçek zamanlı düğümlere yeterli kaynağın sağlanamamasıdır. Bu nedenle gerçek zamanlı olmayan düğümler daha yüksek bir iş çıkarma oranına sahiptir. Grafiklerde sunulan sonuçlara göre, gerçek zamanlı olmayan düğümlerin için toplam yük değeri 2 iken iş çıkarma oranı 0,94, toplam yük değeri 5 iken 2,15 ve toplam yük değeri 10 iken 3,63 olmaktadır. Gerçek zamanlı düğümler için ise, toplam yük değeri 2 iken iş çıkarma oranı 0,91, toplam yük değeri 5 iken 1,86 ve toplam yük değeri 10 iken 2,55 olmaktadır. Bu sonuçlara göre, artan yük ile birlikte gerçek zamanlı düğümlerden, gerçek zamanlı olmayan düğümlere göre daha düşük iş çıkarma oranları elde edilmektedir.



Şekil 5.4. Önerilen dinamik kanal tahsis algoritmasının kullanıldığı ağ modelinde iş çıkarma oranları

5.5. Değerlendirme

Bu bölümde, önerilen dinamik kanal tahsis yönteminin başarımı, analitik modelleme ve benzetim sonuçları ile değerlendirilmiştir. Elde edilen sonuçlar, önerilen yöntemle hem gerçek zamanlı (GZ) hem de gerçek zamanlı olmayan (GZ0) düğümler için ağ başarımında anlamlı iyileşmeler sağlandığını göstermektedir. Başarım analizinde kullanılan temel metrikler, kanal kullanım oranı, çağrı bloke olasılığı, çağrı tamamlama oranı ve iş çıkarma oranı olarak belirlenmiştir.

Elde edilen sonuçlara göre, kanal kullanım oranı incelendiğinde, gerçek zamanlı düğümlerin daha yüksek kanal kullanım oranlarına sahip olduğu, dolayısıyla sistem kaynaklarını daha yoğun kullandığı gözlemlenmiştir. Bu durum, gerçek zamanlı trafiğin kaynak ihtiyaçlarının daha yüksek olmasından kaynaklanmaktadır (Rappaport, 2002).

Çağrı bloke olasılığı analizlerinde, artan toplam yük ile birlikte çağrı bloke olasılıklarının arttığı görülmüştür. Gerçek zamanlı düğümler için çağrı bloke olasılıkları, aynı yük değerlerinde gerçek zamanlı olmayan düğümlere kıyasla daha yüksek bulunmuştur. Bunun temel nedeni, gerçek zamanlı düğümlerin daha yüksek kaynak ihtiyacı ve bu kaynakların sağlanmasının daha zor olmasıdır (Kumar & Misra, 2014).

Çağrı tamamlama oranı analizleri, artan toplam yük ile birlikte her iki kullanıcı türü için de oranların azaldığını göstermiştir. Ancak, gerçek zamanlı olmayan düğümler, daha düşük kaynak ihtiyaçlarından dolayı daha yüksek çağrı tamamlama oranlarına sahiptir (Chen et al., 2021). Bu durum, sistem kaynaklarının artan yük altında yetersiz hale gelmesiyle açıklanabilir.

İş çıkarma oranı analizlerinde ise, her iki kullanıcı türü için de toplam yük arttıkça iş çıkarma oranlarının arttığı, ancak gerçek zamanlı olmayan düğümlerin daha yüksek iş çıkarma oranlarına ulaştığı görülmüştür. Bu fark, artan yük değerlerinde sınırlı kaynakların öncelikli olarak gerçek zamanlı düğümlerin ihtiyaçlarına yönlendirilmesinden kaynaklanmaktadır (Tang et al., 2020).

Sonuç olarak, önerilen dinamik kanal tahsis yöntemi, farklı kullanıcı türlerinin ihtiyaçlarını karşılamak üzere geliştirilmiş ve benzetim sonuçları analitik

modellemelerle tutarlı bir şekilde, yöntemin etkili olduğunu göstermiştir. Bu çalışma, dinamik kanal tahsisi ile IoT ağlarının başarımını artırmaya yönelik gelecekteki çalışmalara temel oluşturabilecek önemli bir katkı sunmaktadır.



6. SONUÇLAR VE ÖNERİLER

Bu tez çalışmasında, heterojen veri trafiği içeren IoT sistemleri için dinamik bir kanal tahsis yönteminin analitik ve benzetim modelleri geliştirilmiş ve bu yöntemin örnek bir ağ senaryosunda başarımlarını değerlendirmesi gerçekleştirilmiştir. Önerilen dinamik kanal tahsis yönteminin analitik modelinde 2 boyutlu sürekli Markov zinciri kullanılmıştır. Ayrıca, Python programlama dili ile ayrık-zamanlı modelleme tekniği kullanılarak benzetim modeli elde edilmiştir. Önerilen dinamik kanal tahsis yöntemi, IoT ağlarının ortam erişim ve kontrol katmanında, farklı türdeki veri trafiğini (heterojen veri trafiği) etkili bir şekilde yönetmek ve sınırlı ağ kaynaklarının kullanımını optimize etmek üzere geliştirilmiştir. Bu yöntem, gerçek zamanlı (GZ) ve gerçek zamanlı olmayan (GZO) veri trafiği türlerine servis kalitesi desteği sağlamak için düğüm gereksinimlerini dikkate alarak sistem kaynaklarının etkin şekilde paylaşılmasını hedeflemektedir. Böylece, hem gerçek zamanlı (real-time) hem de gerçek zamanlı olmayan (non-real time) iki farklı veri trafik türünün IoT ağı üzerinden başarılı bir şekilde iletimi gerçekleştirilmiştir. Önerilen dinamik kanal tahsis yönteminin başarımlarını değerlendirmesinde, kanal kullanım oranı, çağrı tamamlama oranı, iş çıkarma oranı ve çağrı bloke oranı başarımlarını metrikleri olarak benimsenmiştir. Elde edilen sonuçlara göre, kanal sayısı 12, gerçek-zamanlı ve gerçek zamanlı-olmayan düğüm sayılarının 10 olduğu bir ağ senaryosunda, gerçek zamanlı olmayan düğümlerin kanal kullanım oranları toplam yük 2 iken %3,95, toplam yük 5 iken %8,99 ve toplam yük 10 iken %15,16 olduğu görülmektedir. Gerçek zamanlı düğümlere ait kanal kullanım oranları ise toplam yük 2 iken %19,03, toplam yük 5 iken %38,85 ve toplam yük 10 iken %53,31 olarak elde edilmiştir. Bu sonuçlar değerlendirildiğinde gerçek zamanlı düğümlerin sistem kaynaklarını daha yüksek oranda kullandığı ve toplam ağ kapasitesinde büyük pay aldığı anlaşılmaktadır. Çağrı bloke olasılığı ise, gerçek zamanlı olmayan düğümler için toplam yük değeri 2 iken 0,003, toplam yük değeri 5 iken 0,031 ve toplam yük değeri 10 iken 0,11 olmaktadır. Gerçek zamanlı düğümler için ise, toplam yük değeri 2 iken 0,01, toplam yük değeri 5 iken 0,11 ve toplam yük değeri 10 iken 0,34 olmaktadır. Bu sonuçlara göre, gerçek zamanlı düğümlere ait çağrı bloke olasılıkları artan toplam yük ile birlikte artmaktadır.

Örnek ağ modelinden elde edilen çağrı tamamlama oranları sonuçları ise, gerçek zamanlı olmayan düğümler için toplam yük değeri 2 iken %99,68, toplam yük değeri 5 iken %96,81 ve toplam yük değeri 10 iken %88,97 olmaktadır. Gerçek zamanlı düğümler için ise, toplam yük değeri 2 iken %98,89, toplam yük değeri 5 iken %88,32 ve toplam yük değeri 10 iken %65,05 olmaktadır. Bu sonuçlara göre, gerçek zamanlı düğümlere ait çağrı tamamlama oranları artan yük ile birlikte gerçek zamanlı olmayan düğümlere göre daha hızla düşmektedir. Örnek ağ senaryosundan elde edilen yüksek çağrı tamamlama oranları, önerilen dinamik kanal tahsis yönteminin servis kalitesi desteği ile IoT uygulamalarına hizmet verdiğini göstermektedir. Önerilen modellerden elde edilen iş çıkarma oranları ise, gerçek zamanlı olmayan düğümler için toplam yük değeri 2 iken 0,94, toplam yük değeri 5 iken 2,15 ve toplam yük değeri 10 iken 3,63 olmaktadır. Gerçek zamanlı düğümler için ise, toplam yük değeri 2 iken 0,91, toplam yük değeri 5 iken 1,86 ve toplam yük değeri 10 iken 2,55 olmaktadır. Bu sonuçlara göre, artan yük ile birlikte gerçek zamanlı düğümlerden, gerçek zamanlı olmayan düğümlere göre daha düşük iş çıkarma oranları elde edilmektedir.

6.1. Gelecek Çalışma Önerileri

Bu tez çalışmasının IoT ağlarında dinamik kanal tahsil yöntemlerinin analitik ve benzetim modellerinin geliştirilmesinde temel teşkil etmesi düşünülmektedir. Sunulan modeller kullanılarak, sonraki çalışmalarda öncelik-tabanlı (priority-based), ara-bellek tabanlı (queueing-based) ve hibrid kanal tahsis yöntemleri gibi daha kapsamlı yöntemlerin geliştirilmesi hedeflenmektedir.

KAYNAKLAR

- Abramson, N. (1970). The ALOHA system: Another alternative for computer communication, *Proceedings of the 1970 Fall Joint Computer Conference, Part I*, 281–285. IEEE.
- Atmaca, S. (2025). Nesnelerin İnterneti Uygulamalarında Heterojen Veri Trafiklerinin Etkin ve Adil İletimi için Servis Kalitesi Destekli Yeni Bir Ortam Erişim Kontrol Protokolü Tasarım ve Başarım Analizi, Kabul Edildi.
- Atzori, L., Iera, A. & Morabito, G. (2010). The Internet of Things: A survey, *Computer Networks*, 54(15), 2787–2805.
- Bandyopadhyay, D. & Sen, J. (2011). Internet of Things: Applications and challenges in technology and standardization, *Wireless Personal Communications*, 58(1), 49–69.
- Boyan, F. (2018). *QoS-Quality of Service Nedir?* Retrieved from <https://www.firatboyan.com/qos-quality-of-service-nedir.aspx>
- Chen, J., Zhang, X., & Wang, Y. (2021). Resource allocation in IoT networks: A survey. *IEEE Communications Surveys & Tutorials*, 23(3), 1-24.
- Chiang, M. & Zhang, T. (2016). Fog and IoT: An overview of research opportunities, *IEEE Internet of Things Journal*, 3(6), 854–864.
- Goldsmith, A. (2005). *Wireless communications*. Cambridge University Press.

- Gubbi, J., Buyya, R., Marusic, S. & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions, *Future Generation Computer Systems*, 29(7), 1645–1660.
- Huang, J., Wang, X. & Xu, B. (2015). IoT-based real-time urban traffic information collection, *IEEE Sensors Journal*, 15(5), 2697–2704.
- Islam, S. M. R., Kwak, D., Kabir, H. M., Hossain, M. & Kwak, K. S. (2015). The Internet of Things for health care: A comprehensive survey, *IEEE Access*, 3, 678–708.
- Kamilaris, A., Kartakoullis, A. & Prenafeta-Boldú, F. X. (2017). A review on the practice of big data analysis in agriculture, *Computers and Electronics in Agriculture*, 143, 23–37.
- Khan, S. U., Zaheer, R., & Khan, S. (2012). Future Internet: The Internet of Things architecture, possible applications and key challenges. In *10th International Conference on Frontiers of Information Technology (FIT 2012)* (pp. 257–260). IEEE.
- Kleinrock, L., & Tobagi, F. (1975). Packet switching in radio channels: Part I—Carrier sense multiple-access modes and their throughput-delay characteristics. *IEEE Transactions on Communications*, 23(12), 1400-1416.
- Kleinrock, L. (1975). *Queueing systems: Theory*. Wiley-Interscience.
- Kumar, A., & Misra, S. (2014). Call blocking probability analysis in wireless networks. *Wireless Networks*, 20(5), 1291-1303.

- Niu, J., Zhang, X., Yang, L. T., Liu, M. & Ji, W. (2015). A trustworthy data collection approach with error detection in wireless sensor networks, *IEEE Transactions on Industrial Informatics*, 11(6), 1486–1494.
- Perera, C., Zaslavsky, A., Christen, P. & Georgakopoulos, D. (2014). Context-aware computing for the Internet of Things: A survey, *IEEE Communications Surveys & Tutorials*, 16(1), 414–454.
- Proakis, J. G. & Salehi, M. (2008). *Communication systems engineering*. Pearson.
- Rappaport, T. S. (2002). *Wireless communications: Principles and practice*. Prentice Hall.
- Raza, U., Kulkarni, P., & Sooriyabandara, M. (2016). Low power wide area networks: An overview. *IEEE Communications Surveys & Tutorials*, 19(2), 855-873.
- Roberts, L. G. (1972). ALOHA packet system with and without slots and capture. *Computer Communication Review*, 5(2), 28-42.
- Rom, R. & Sidi, M. (1990). *Multiple access protocols: Performance and analysis*. Springer.
- Satyanarayanan, M. (2017). The emergence of edge computing, *Computer*, 50(1), 30–39.
- Solanas, A. et al. (2014). Smart health: A context-aware health paradigm within smart cities, *IEEE Communications Magazine*, 52(8), 74–81.

Stewart, W. J. (1994). *Introduction to the numerical solution of Markov chains*. Princeton University Press.

Viterbi, A. (1995). *CDMA: Principles of spread spectrum communication*. Addison-Wesley.

Wang, P., Xu, L. D. & Bi, Z. (2018). Internet of Things in healthcare: Architecture, applications, and future directions, *IEEE Internet of Things Journal*, 5(5), 3209–3221.

Zanella, A., Bui, N., Castellani, A., Vangelista, L. & Zorzi, M. (2014). Internet of Things for smart cities, *IEEE Internet of Things Journal*, 1(1), 22–32.

ÖZGEÇMİŞ

Kişisel Bilgiler

Ad Soyad : F***** O***
Uyruk : T.C.
Doğum Yeri ve Tarihi : B*****. **/**/19**
Medeni Hali : Evli
Telefon : 0 5*****
E-posta : f*****@mu.edu.tr

Eğitim

Alınan Derece	Aldığı Kurum/Üniversite	Mezuniyet Yılı
Lise	Bilecik Refik Arslan Öztürk Fen Lisesi	2008
Lisans	Kocaeli Üniversitesi Bilgisayar Mühendisliği	2014

İş Tecrübesi

Yıl	Yer	Görev
08/2014 – 01/2021	Jandarma Genel Komutanlığı MEBS Başkanlığı	Muhabere Subayı
03/2021 – 12/2022	Netelsan	AR-GE Proje Sorumlusu
01/2023 – 02/2024	Kurumsalkiraci.com	Yazılım Geliştirme Sorumlusu
03/2024 – Halen	Kredinet.net	Yazılım Geliştirme Sorumlusu

Yabancı Dil

Yabancı Dil	Konuşma-Anlama	Okuma-Yazma
İngilizce	Orta	Orta