

KABLOSUZ SENSÖR AĞLARINDA
ZigBee TABANLI ADRESLEME: TEORİ,
MODELLEME VE SİMÜLASYON



Arş. Gör. Dr. Gürkan DOĞAN
Doç. Dr. Güngör YILDIRIM
Prof. Dr. Yetkin TATAR

ISBN: 978-625-5923-38-7

Ankara -2025

Kablosuz Sensör Ağlarında ZigBee Tabanlı Adresleme: Teori, Modelleme ve Simülasyon

YAZARLAR

Arş. Gör. Dr. Gürkan Doğan¹

Doç. Dr. Güngör Yıldırım²

Prof. Dr. Yetkin Tatar³

¹Munzur Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği
Bölümü, Tunceli, Türkiye
gurkandogan@munzur.edu.tr
ORCID ID: 0000-0003-2497-8348

²Fırat Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği
Bölümü, Elazığ, Türkiye
gungor.yildirim@firat.edu.tr
ORCID ID: 0000-0002-4096-4838

³Fırat Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği
Bölümü, Elazığ, Türkiye
ytatar@gmail.com
ORCID ID: 0000-0002-7181-0014

DOI: <https://doi.org/10.5281/zenodo.15297229>



Copyright © 2025 by UBAK publishing house

All rights reserved. No part of this publication may be reproduced, distributed or transmitted in any form or by any means, including photocopying, recording or other electronic or mechanical methods, without the prior written permission of the publisher, except in the case of brief quotations embodied in critical reviews and certain other noncommercial uses permitted by copyright law. UBAK International Academy of Sciences Association Publishing House®
(The Licence Number of Publicator: 2018/42945)

E mail: ubakyayinevi@gmail.com

www.ubakyayinevi.org

It is responsibility of the author to abide by the publishing ethics rules.

UBAK Publishing House – 2025©

ISBN: 978-625-5923-38-7

April / 2025

Ankara / Turkey

ÖNSÖZ

Sevgili Okuyucular,

Teknoloji gelişmeye devam ettikçe, fiziksel sistemlerin dijital zeka ile entegrasyonu giderek daha da önemli hale geliyor. Kablosuz sensör ağları (KSA) ve nesnelerin interneti (IoT), cihazların verileri otonom olarak algılamasını, işlemesini ve paylaşmasını sağlayarak bu dönüşümün merkezinde yer alıyor. Akıllı şehirlerden hassas tarıma kadar, bu teknolojiler sayısız uygulamada gerçek zamanlı, veri odaklı karar alma için omurga sağlıyor.

Bu sistemleri desteklemek için güvenilir ve verimli iletişim protokolleri olmazsa olmazdır. Zigbee, minimum enerji tüketimiyle uzun süreler boyunca çalışması gereken cihazlar için optimize edilmiş düşük güç, düşük veri hızı kablosuz iletişimi sunarak bu alanda önemli bir rol oynar. Ağ oluşturma yeteneği, kapsama alanını ve güvenilirliği artırarak akıllı evler, endüstriyel izleme ve sağlık hizmetleri gibi uygulamalar için ideal hale getirir.

Bu kitap, Zigbee teknolojisinin kapsamlı bir teorik incelemesini ve OPNET Modeler simülasyon aracında pratik bir uygulamasını sunarak, sizlere bir iletişim ağı geliştirmek için temel bir alt yapı sağlamaktadır. Bu kapsamlı kaynağın okurlarımız için teorik ve pratik açıdan faydalı ve ufuk açıcı olmasını umuyorum.

28/04/2025

İÇİNDEKİLER

ÖNSÖZ	III
ÖZET	V
SUMMARY	VI
KISALTMALAR LİSTESİ	VII
1. GİRİŞ	1
2. KSA İLE İLGİLİ YAPILAN ÇALIŞMALAR	5
3. ZİGBEE PROTOKOLU VE PRİMİTİF KAVRAMI	11
3.1 Zigbee Protokol Mimarisi	11
3.1.1 Fiziksel Katman	12
3.1.2 MAC Katmanı	12
3.1.3 Ağ Katmanı	13
3.1.4 Uygulama Katmanı	14
3.2 Primitif Kavramı	16
4. ZİGBEE DÜĞÜMÜ MODELLEME	21
4.1 Modellemede Kullanılan Çerçeve Formatları.....	21
4.1.1 Genel MAC Çerçeve Formatı	22
4.1.2 MAC Komut Çerçeve Formatı	30
4.1.3 MAC Komut Çerçeveleri	32
4.1.4 ZigBee düğümünün tasarlanma süreçleri	38
4.2 Modellemede Kullanılan Düğüm Yapısı ve Primitifler	45
4.3 Modellemede Kullanılan Süreç Yapısı.....	52
5. ZİGBEE AĞININ SİMÜLASYONU VE ANALİZİ	58
6. SONUÇLAR	72
KAYNAKÇA	74

ÖZET

Kablosuz sensör ağlarının, sert doğa olaylarını izleme, insanların ve makine sağlığını izleme, deprem gibi doğal afet olaylarını izleme, askeri olarak hudutları gözleme gibi birçok kullanım alanları vardır ve kullanımı gün geçtikçe artmaktadır. Ayrıca IoT ve CPS sistemlerin de temel üniteleri olmalarından dolayı önemleri bir kat daha artmıştır. Bu durumda kablosuz sensör ağ teknolojilerinin ve iletişim protokollerinin sürekli olarak geliştirilmesi ve iyileştirilmesini gerektirmektedir.

Öte yandan büyük ve kapsamlı sistemlerin oluşturulmasındaki ilk aşama sistemin modellenmesi ve simüle edilerek analizlerinin yapılmasıdır. Kablosuz sensör ağ sistemleride bunlardan birisidir. Bu ağlar için en fazla kullanılan modelleme ve simülasyon platformlarından biriside OPNET Modeler'dir. Bu çalışmada, öncelikle kablosuz sensör ağlarının bir teknolojisi olan ZigBee incelenmiştir. Ardından, ZigBee teknolojisinin fiziksel ve mac katmanlarını tanımlayan IEEE 802.15.4 standardı ele alınarak, bu standartlara göre OPNET Modeler simülasyon aracında ağa üyelik sürecindeki uç cihazlara ağ adresi ataması için bir koordinatör ve bir uç cihazın modellenmesi yapılmıştır. Son olarak, bir koordinatör ve 4 adet uç düğüm için bir senaryo hazırlanarak simülasyon yapılmış ve analizi gerçekleştirilmiştir.

Anahtar kelimeler: Opnet, simülasyon, kablosuz sensör ağları, zigbee, protokoller.

SUMMARY

Wireless sensor networks have many using fields such as monitoring hard natural events, monitoring people and machine health, monitoring natural disasters such as earthquakes, monitoring military boundaries, and are increasing day by day. In addition, the wireless sensor networks are also the basic units of the IoT and CPS systems, so its importance has increased one more time. In this case, the technology and communication protocols of wireless sensor networks need to be constantly improved and optimized.

On the other hand, the first stage in creating large and comprehensive systems is modeling and simulating the system and analyzing it. Wireless sensor network systems are one of them. One of the most widely used modeling and simulation platforms for these networks is OPNET Modeler. In this study, ZigBee, a technology of wireless sensor networks, was first examined. Then, the IEEE 802.15.4 standard, which defines the physical and mac layers of ZigBee technology, was taken into consideration and according to these standards, a coordinator and an end device were modeled in the OPNET Modeler simulation tool to assign network addresses to end devices in the network association process. Finally, a scenario was prepared for a coordinator and 4 end nodes, a simulation was performed and its analysis was performed.

Keywords: Opnet, simulation, wireless sensor network, zigbee, protocols.

KISALTMALAR LİSTESİ

- STD: Durum Geçiş Diyagramı (State Transition Diagram)
- ICI: Arayüz Kontrol Bilgisi (Interface Control Information)
- PDF: Olasılık Yoğunluk Fonksiyonu (Probability Density Function)
- SAP: Hizmet Erişim Noktası (Service Access Point)
- PHY: Fiziksel (Physical)
- MAC: Ortam Erişim Kontrolü (Medium Access Control)
- NWK: Ağ (Network)
- APS: Uygulama Desteği Alt Katmanı (Application Support Sub-layer)
- ZDO: ZigBee Cihaz Nesnesi (ZigBee Device Object)
- OSI: Açık Sistemler Arası Bağlantı (Open Systems Interconnection)
- POS: Kişisel Çalışma Alanı (Personal Operating Space)
- FFD: Tam İşlevli Cihaz (Full-Function Device)
- RFD: Düşük İşlevli Cihaz (Reduced-Function Device)
- APL: Uygulama Katmanı (Application Layer)
- MHR: MAC Çerçeve Başlığı (MAC Header)
- MFR: MAC Çerçeve Alt Başlığı (MAC Footer)
- FCS: Çerçeve Kontrol Dizisi (Frame Control Sequence)
- ACK: Onay (Acknowledgment)
- PAN: Kişisel Alan Ağı (Personal Area Network)
- BSN: İşaret Sıra Numarası (Beacon Sequence Number)
- DSN: Veri Sıra Numarası (Data Sequence Number)
- PIB: PAN Bilgi Tabanı (PAN Information Base)
- NLME: Ağ Katmanı Yönetim Varlığı (Network Layer Management Entity)
- NLDE: Ağ Katmanı Veri Varlığı (Network Layer Data Entity)

MLME: MAC Katmanı Yönetim Varlığı (MAC Layer Management Entity)

MCPS: MAC Ortak Bölüm Alt Katmanı (MAC Common Part Sub-Layer)

IoT: Nesnelerin İnterneti (Internet of Things)

CPS: Siber Fiziksel Sistemler (Cyber-Physical Systems)

1. GİRİŞ

Kablosuz Sensör Ağları (KSA - WSN), birçok disiplinde yaşam kalitesini iyileştirmek için temel bir teknoloji olarak ortaya çıkmıştır. Kablosuz Sensör Ağlarının Nesnelerin İnterneti (IoT) ve Siber Fiziksel Sistemlerine (CPS) entegre edilmesi ile birlikte, akıllı tarım, akıllı evler, akıllı şehirler, sağlık hizmetleri ve endüstriyel izleme alanlarında çeşitli uygulamaların gelişimine ön ayak olmuştur (Bhatia vd., 2023; Jawhar vd., 2018; Nasri vd., 2021). Bu sistemler, genellikle belirlenen bir alana dağıtılmış olan bir çok sensör düğümünü içerir. Gerçek zamanlı izleme ve akıllı karar vermeyi mümkün kılan bu sistemler, sıcaklık, nem, basınç ve hareket gibi fiziksel parametreler hakkında veri toplar (Bhatia vd., 2023; Koshariya vd., 2023; Volpi vd., 2023). Bu düğümler daha sonra verileri işler ve kablosuz olarak merkezi bir baz istasyonuna veya ağ geçidine iletir. Bu sayede, sensörlerden elde edilen bilgilerin toplanıp analiz edilmesi ve karar alması için uzak bir sunucuya iletilmesi sağlanabilir (Amxilatidis vd., 2011).

KSA'ların kendi kendini organize edebilme ve çok atlamalı iletişim özellikleri, onları geleneksel kablolu sensör sistemlerinden ayırır ve zor koşullarda esnek dağıtıma olanak tanır (Sun vd., 2020). Bu sistemlerin esnekliği veya uyarlanabilirliği, dinamik koşullarda etkili bir şekilde çalışmalarını ve ağ performansını iyileştirmek için de davranışlarını değiştirmelerini sağlar (Portocarrero vd., 2015). Sensör düğümlerinin pil değişimini veya şarjını güçleştiren sınırlı güç kaynağına sahip olması, enerji açısından verimli iletişim protokolleri

ve veri işleme tekniklerini gerektiren önemli bir tasarım kısıtlamasına yol açar (Liu vd., 2019). Enerji açısından verimli yönlendirme protokollerinin geliştirilmesi, ağ ömrünü uzatmak ve otonom ortamlarda kesintisiz çalışmayı sürdürmek için çok önemlidir (Jabbar vd., 2015). Araştırmacılar bu problemi ele almak için çeşitli optimizasyon stratejileri önermişlerdir. Örneğin, enerji tüketimini azaltmak ve ağ ömrünü uzatmak için genetik algoritmalar ve parçacık sürüsü optimizasyonunu kullanan karmaşık dağıtım stratejileri geliştirilmiştir(Bhatia vd., 2023). Diğer alternatif yaklaşımlar arasında, bulanık tabanlı yönlendirme protokolleri, katmanlar arası yönlendirme protokolleri ve öncelikli olay odaklı enerji verimli algoritmaları bulunur (Koshariya vd., 2023; Nasri vd., 2021). Bunun yanı sıra, WSN'lerin IoT uygulamalarındaki evrimi ile birlikte bu sistemleri etkili bir şekilde yönetmek için yeni protokollerin ve katmanlı yapıların geliştirilmesini gerektirmiştir. EEGT (Şebeke Hücrelerine Dayalı Enerji Verimli Yönlendirme Protokolü) ve EEDC (Enerji Verimli Veri İletişim Şeması) gibi enerji açısından verimli yönlendirme protokolleri, ağ ömrünü ve performansını iyileştirmek için tanıtılmıştır (Duy Tan vd., 2023; Gupta vd., 2023). WSN'ler paylaşılabılır sistemlere dönüşmeye devam ettikçe, güvenlik zorluklarını ele almak ve IoT nesnelere arasındaki iletişimi standartlaştırmak, gelecekteki gelişimleri ve yaygın bir şekilde benimsenmesi açısından çok kritik olacaktır (Gaikwad vd., 2015; Matharu vd., 2014; Pundir vd., 2020).

KSA'ların tasarımı ve optimizasyonu, enerji, işlem gücü ve bellek gibi kısıtlı kaynaklar nedeniyle önemli zorluklar ortaya çıkarmaktadır. Bu

sistemlerin etkili ve maliyeti düşük bir şekilde performans deęerlendirmesi ve doęrulaması için geliřmiř modelleme ve simülasyon araçları kullanılabilir (Saba vd., 2020). Simülasyon araçları, çeřitli topolojilerin etkinlięini deęerlendirmede ve kritik performans ölçümlerini belirlemede kritik bir role sahiptir (Mamun, 2012). WSN ve dięer iletiřim aęlarının performansının modellenmesi, simüle edilmesi ve analiz edilmesi için OMNET++, NS2, J-Sim vb. gibi simülasyon platformları bulunmasına raęmen aę uzmanları; detaylı protokol modelleme, ölçeklenebilirlik, görsel arayüz, kapsamlı kütüphane, özelleřtirme, hiyerarřik modelleme, ayrık olay simülasyonu vb. gibi özelliklerinden dolayı OPNET Modeler simülasyon platformunu yaygın olarak tercih etmektedir (Pan & Jain, 2008; Patel vd., 2019).

Opnet Modeler iletiřim aęları, cihazları, protokolleri ve uygulamaları incelemek için esnek bir şekilde kullanılabilirken ticari bir yazılım saęlayıcısı olmasından dolayı OPNET Modeler, kullanıcılar için nispeten daha güçlü görsel ve grafiksel destek sunar. aę topolojisi ve varlıkları oluřturmak için kullanılır. Grafik düzenleyici arabirimi, uygulama katmanından fiziksel katmana kadar aę topolojisi ve varlıkları oluřturmak için kullanılır. Dahası, Opnet Modeler, ayrık olay sistemi adı verilen bir mekanizmaya dayanmaktadır. Bu da, sistem davranıřının, sistemdeki olayları kullanıcının kurduęu senaryo sırasına göre modelleyerek simüle edebileceęi anlamına gelir. Aęları düzenlemek için de hiyerarřik yapı kullanılır. Dięer aę simülatörleri gibi, OPNET Modeler ayrıca kullanıcıların protokolün paket formatını

tanımlamaları için programlama araçları sağlar. Programlama araçları ayrıca durum geçiş makinesini tanımlama, ağ modelini ve process modülünü tanımlama görevlerini yerine getirmek için gereklidir. Tüm bu özellikleriyle OPNET Modeler, modelleme, simülasyon ve analiz olmak üzere üç temel yeteneğe sahiptir (Doğan vd., 2018; Pan & Jain, 2008; Sharma vd., 2016).

Bu çalışmada, ZigBee protokolünün temel alındığı ağlarda kullanılabilen bir KSA düğüm yapısının ve katmanlı ağ protokol yapısı detaylı bir şekilde ele alınmıştır. Bu protokole göre koordinatör ve uç cihaz olarak çalışabilen bir KSA düğümünün ağ adresi atama sürecinin Opnet Modeler platformunda modellenmesi, simülasyonu ve analizi detaylı bir şekilde açıklanmıştır.

2. KSA İLE İLGİLİ YAPILAN ÇALIŞMALAR

Son yıllarda, akıllı şehirler, çevresel izleme, sağlık hizmetleri, endüstriyel otomasyon ve çevresel izleme gibi çok çeşitli alanlarda kullanılması nedeniyle, KSA'lar büyük ilgi görmüştür (Randhawa, 2014; Sharma vd., 2016). Bu ilgi artışı, kablosuz arayüzlerle donatılmış daha küçük, daha ucuz ve daha akıllı sensörlerin geliştirilmesine yol açmış ve bunların birbiriyle iletişim kurmasını ve dolayısıyla ağlar oluşturmasını sağlamıştır (Agarkhed vd., 2020; Yick vd., 2008). Bu ağlar, gerçek zamanlı izleme, hedef takibi ve fiziksel olayların tanınması için çözümler sunarak, belirli uygulama özelliklerine ilişkin giderek artan sayıda araştırma sorusuyla karşı karşıya kalmıştır (ABBA ARI vd., 2015). Özellikle, protokol geliştirme, sensör düğümlerinin kısıtlı pil ömrü, işlem gücü ve bellek kapasitesi gibi kaynak sınırlamaları nedeniyle benzersiz zorluklar ortaya çıkarır (Liu vd., 2019). Bu nedenle, enerji verimliliği, ağır ömrünü ve operasyonel etkinliğini doğrudan etkilediği için KSA'larda protokol geliştirme süreci en önemli tasarım hususudur. Bu doğrultuda, bazı araştırmacılar, veri iletim yollarını optimize ederek, iletişim yükünü en aza indirerek ve güç yönetimi stratejilerini kullanarak enerji tüketimini azaltan enerji açısından verimli çeşitli yönlendirme protokollerini araştırmışlardır (Jabbar vd., 2015). Örneğin, sensör düğümlerini kümeler halinde gruplayan kümeleme algoritmaları yaygın olarak kullanılır. Bu algoritmaların görevi, verileri toplayıp baz istasyonuna iletmek olan küme başlarını belirlemektir (Aadri & Idrissi, 2017; Aliouat & Harous, 2014). Bu

yöntem, uzun mesafeli iletişimde kullanılan düğüm sayısını azaltarak enerji tasarrufu sağlar ve ağ ömrünü uzatır (Meelu vd., 2010). Başka bir yöntem, düğümlerin görev döngüsü mekanizmalarını değiştirerek etkin ve uyku modları arasında dönüşümlü olarak hareketsiz zamanlarda enerji tasarrufu sağlamaktır (Karimi-Bidhendi vd., 2019). Başka bir çalışmada, Güç Kontrolü ve Gecikme Farkında Yönlendirme ve MAC Protokolü (PCDARM), yönlendirme ve MAC katmanı optimizasyonlarını birleştiren bir protocol sunulmuştur. Uçtan uca gecikmeyi ve enerji tüketimini azaltmak için PCDARM, MAC katmanında TDMA tabanlı bir yuva ataması kullanır (Yang vd., 2012). Boru Hattı Yönlendirme Gelişmiş MAC Protokolü (PRMAC), çok atlamalı iletimlerde uçtan uca gecikmeyi azaltmak için katmanlar arası optimizasyon kullanır (Canli & Khokhar, 2009). Çoklu Jeton Tabanlı MAC-Cum-Routing Protokolü, yönlendirme ve MAC işlevlerini entegre eden gelişmiş bir MAC protokolünün başka bir örneğidir. Bu protokol, güvenilir veri iletimi sağlamak ve enerji tüketimini azaltmak için jeton tabanlı mekanizmalar kullanır (Dash vd., 2019). MAC-aware Yönlendirme protokolü (MAR-WSN), TDMA planlamasını ve iki atlamalı komşu bilgisini kullanarak yönlendirme kararları alır. Geleneksel katmanlı yaklaşımlara kıyasla, MAR-WSN gecikmeyi, enerji tüketimini ve atlama sayısını azalttığı gösterilmiştir (Louail vd., 2015). Gerçek Zamanlı X Katmanı Protokolü (RTXP), WSN'lerde gerçek zamanlı iletişimi sağlamak için MAC ve yönlendirme işlevlerini birleştirir. RTXP protokolü, iletili seçimini ve deterministik ortam erişimini sağlamak için atlama sayısına dayalı bir sanal koordinat sistemi kullanır. Simülasyon

sonuçları, RTXP protokolünün zorlu radyo kanalı koşullarında bile deterministik olmayan çözümlerden daha iyi olduğunu göstermiştir (Mouradian vd., 2014).

KSA protokollerinin yaşam döngüsü boyunca, çeşitli çalışma koşulları altında ağın güvenilirliğini, sağlamlığını ve performansını garanti etmek için test ve doğrulama önemli adımlardır. OPNET Modeler gibi simülasyon araçları, araştırmacıların WSN protokollerinin performansını kontrollü bir ortamda değerlendirmek için uygun maliyetli ve verimli bir yol sunar. Bu araçlar, araştırmacıların dağıtımdan önce olası sorunları belirlemesini ve protokol parametrelerini optimize etmesini sağlar. Nitekim, Çin'de yapılan bir çalışmada, minimum rota maliyetine dayanan geliştirilmiş bir AODV yönlendirme protokolü önerilmiştir ve OPNET Modeler, gelişmiş sensörle kablosuz sensör ağının performansını değerlendirmek için, uçtan uca gecikme ve diğer parametreleri simüle etmek için kullanılmıştır. Simülasyon sonuçları, önerilen AODV protokolü ile performansının daha iyi olduğunu göstermektedir (Yu & Yao, 2012). Hindistan'da yapılan başka bir çalışmada, ZigBee protokolünü uygulayan bir KSA'nın taklit edilmesi ve uçtan uca gecikmenin farklı ağlara göre değerlendirilmesi için OPNET Modeler kullanılmıştır. Simülasyon aracında sabit ve hareketli cihazların yer aldığı iki farklı ağ ve bu ağlardaki yönlendiricilerin aktif ve pasif olmasına göre de iki farklı senaryo uygulanarak uçtan uca gecikme analiz edilmiştir. Sonuçlar, her iki ağda yönlendiricilerin aktif olduğu durumda uçtan uca gecikmenin, yönlendiricinin pasif olduğu duruma göre daha

yüksek olduğu ortaya çıkmıştır (Deepika & Sharma, 2014). Çinde yapılan başka bir çalışmada, mevcut Zigbee WSN modellerindeki düğüm hareketliliği kısıtlamaları sorununu ele alan yeni bir Zigbee uyumlu simülasyon modeli önerilmektedir. Düğüm hareketliliğini desteklemek için Zigbee protokolleriyle uyumlu bir ağ katmanı modeli ve iyileştirilmiş bir AODV yönlendirme algoritması geliştirmektedir. Kapsamlı performans karşılaştırmaları, önerilen modelin orijinal Zigbee modelinden daha iyi performans gösterdiğini ve düğüm hareketliliği yönlendirme hatalarına neden olduğunda alternatif yolları hızla belirleyebildiğini göstermektedir (Song, 2011). Diğer bir çalışmada, LEACH protokolü ile gelişmiş yönlendirme protokolü olan I-LEACH (Improved-LEACH) belirli kriterlere göre kıyaslanmıştır. LEACH ve I-LEACH protokolleri ağ yaşam döngüsü, enerji tüketimleri ve ağ ölçeklenebilirliği OPNET ağ simülasyon aracı kullanılarak simülasyonu ve analizi yapılmıştır. Bu kıyaslama sonucunda I-LEACH algoritması geleneksel LEACH algoritmasından enerji tüketimi daha da azalttığı ve bu sayede ağ ömrü döngüsünü etkin bir şekilde uzatabildiği görülmüştür (Biswas, 2017). Hindistan Ulusal Teknoloji Enstitüsü, OPNET simülasyon modeli kullanarak slotlu CSMA/CA kanal erişim mekanizmaları kullanan ağların MAC katman performansını analiz etti. Analiz bir PAN koordinatörü ve n tane verici düğüm ile yıldız topolojisi altında tasarlanmış bir OPNET simülasyon modeline dayanmaktadır. Simülasyon farklı MAC parametre ayarları ve farklı n sayısı için yapılmıştır. Çalışma, alıcı çerçeve sayısının verici düğüm sayısı ile orantılı olduğunu ve verici düğümlerin uzun bir çerçeve gönderdiğinde koordinatörün daha az

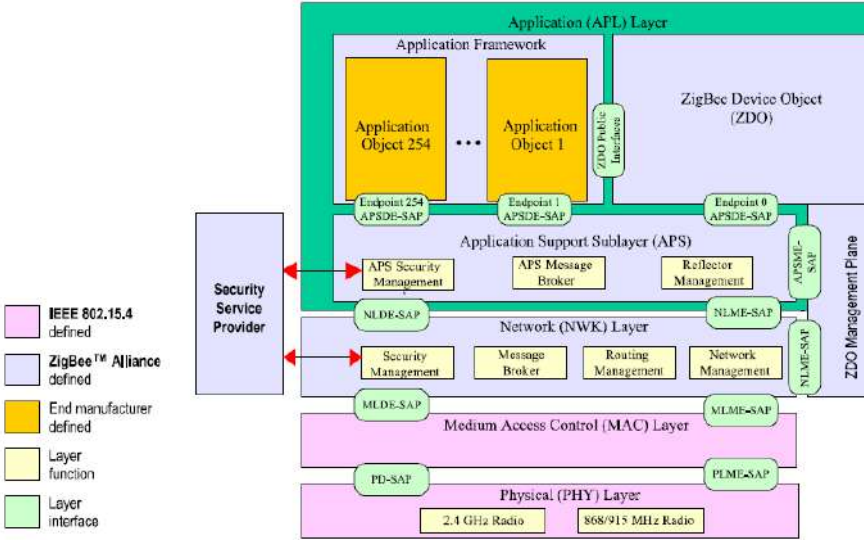
çerçeve aldığını buldu. Alınan çerçeve sayısı ayrıca Back-off Exponent'e ve Backoff sayısına da bağlı olduğu görülmüştür (Biswas, 2017). Sudan Khartoum Üniversitesi, OPNET Modeller kullanarak hiyerarşik yönlendirme protokolleri LEACH, PEGASIS ve TEEN'i kullanan KSA'ların performans değerlendirmesini gerçekleştirmiştir. Bu üç hiyerarşik yönlendirme protokolünün performans değerlendirmesinde iletim gücü, uçtan uca gecikme, iletim esnasında düşen veri, alıcı(sink) tarafından alınan trafik miktarı, hop sayısı, ağ yükü olmak üzere altı tane kriter kullanılarak analiz yapılmıştır. LEACH, tek atlamalı yönlendirme tekniği ve yüksek güç tüketimi nedeniyle büyük ağlar için önerilmemiştir. PEGASIS, veri iletimi için çok atlamalı bir yol kullandığından, zamanın kritik olduğu küçük kablosuz sensör ağları için daha iyidir. TEEN ağları, kullanıcıların ağdaki belirli alanları gözlemlemesine olanak tanıdığı için ani çevresel değişiklikler için kullanışlı hale getiren daha karmaşık bir sensör düğümü mimarisine sahiptir (Al & Bakheet, 2015). Başka bir çalışmada, SSLE yönteminin geliştirilmiş bir versiyonu olan değiştirilmiş sektörel tarayıcı tabanlı lokalizasyon tahmini (SSLE) yöntemini sunulur. Bu teknik, basitleştirilmiş mesaj formatları ve akıllı anten işleme yeteneğine sahip merkezi bir düğüm gerektirir. Değiştirilmiş SSLE, kanal yayılım koşulları dikkate alınarak OPNET Modeller yazılımı kullanılarak modellenir ve simüle edilir. Simülasyon sonuçları, log-normal gölgeleme kanal koşulları altında ortalama %9,5 lokalizasyon hatasıyla lokalizasyon tahmininin yapılabileceğini göstermektedir (Kucuk vd., 2012). Hindistan'da yapılan bir çalışmada, OPNET Modeller kullanarak ağ ve fiziksel katman saldırılarının

etkilerini karşılaştıran Mobil Ad-hoc Ağları (MANET'ler) veya KSA tasarlamak için bir algoritma sunulmaktadır. Her QoS parametresi için tüm saldırılar için bir performans grafiğiyle Kablosuz Mobil Ad-hoc Ağları ve KSA'ları oluşturmak için yeni bir algoritma önerilmektedir. Bu çalışma, mobil düğümler için kullanıcı tanımlı yörüngeyi desteklemekte ve daha genel kullanım için yeni bir yörünge algoritması önerilmektedir (Das vd., 2016). KSA'larda düğüm hataları sırasında Hizmet Kalitesini (QoS) ve bağlantıyı sürdürmek, ağ esnekliği olarak adlandırılır. Ağ esnekliğini artırmak için başka bir çalışmada, KSA'larda olası ağ bağlantısının kesilmesini önlemek amacıyla ana düğümü değiştirmek için proaktif bir mekanizma önerilmiştir. Bu mekanizma, bağlantıyı sürdürmek ve yeni bağlantıları barındırmak için sensör düğümlerinin radyo mesafesi ayarını kullanır. OPNET Modeller kullanılarak test edilen dağıtım şeması, geleneksel yöntemlere göre minimum enerji tüketimi gerçekleştirerek daha iyi performans göstermiştir. Proaktif radyo mesafe ayarlaması, paket düşüşlerini %40 ve gecikmeyi %5,5 oranında azalttığı gözlenmiştir (Verma vd., 2015). Diğer bir çalışmada, OPNET Modeller simülatörünü kullanarak yıldız, mesh ve ağaç yönlendirmesi de dahil olmak üzere farklı ZigBee topolojilerinin performansını araştırır. Sonuçlar, mesh yönlendirmesinin ağaç ve yıldız yönlendirmesinden daha düşük gecikmeye sahip olduğunu, ağaç yönlendirmesinin ise diğer topolojilerden daha yüksek verim elde ettiğini gösterir. Ek olarak, yıldız yönlendirmesi ağaç ve mesh yönlendirmesine kıyasla daha yüksek koordinatör ek yükü ve uçtan uca gecikmeyi sergilediği görülmüştür (Ameen & Nourildean, 2013).

3. ZİGBEE PROTOKOLU VE PRİMİTİF KAVRAMI

3.1 Zigbee Protokol Mimarisi

ZigBee yığın mimarisi, katman adı verilen bir dizi bloktan oluşur. Her katman, üstteki katman için belirli bir hizmet takımını gerçekleştirir. Bir veri varlığı (Data Entity), bir veri aktarımı hizmeti sağlarken bir yönetim varlığı (Management Entity) diğer tüm hizmetleri sağlar. Her bir servis varlığı, bir servis erişim noktası (SAP) aracılığıyla üst katmana bir arayüz sunar ve her SAP, gerekli işlevselliği elde etmek için bir dizi hizmet primitifini destekler (ZigBee Alliance, 2012).



Şekil 3.1. ZigBee yığın mimarisinin anahatları [ZigBee Alliance, 2012].

Şekil 3.1 'de görüldüğü gibi ZigBee yığın mimarisi, standart açık sistemler arası bağlantı (OSI) yedi katmanlı modeli temel alır. IEEE 802.15.4-2003, minimum güç tüketen ve tipik olarak kişisel bir çalışma alanında (POS) çalışan basit cihazları desteklemek için fiziksel katman (PHY) ve ortam erişim kontrol alt katmanının (MAC)

özelliklerini tanımlar. ZigBee Birliği bu temel üzerine ağ (NWK) katmanı ve uygulama desteği (APS) alt katmanı, ZigBee cihaz nesnesi (ZDO) ve üreticinin tanımladığı uygulama nesnelerini içeren uygulama katmanı için bir çatı (framework) sağlar (Lee & Huang, 2006; ZigBee Alliance, 2012).

3.1.1 Fiziksel Katman

IEEE802.15.4 standardının fiziksel katmanı, telsiz alıcı-vericisi ile doğrudan kontrol ve haberleşme sağlayan donanıma en yakın olanlardır. Kanal seçimine yardımcı olmak için ZigBee donanımına erişim, donanımın başlatılması, kanal seçimi, enerji algılama ölçümü, bağlantı kalitesi tahmini ve temiz kanal değerlendirmesi gibi tüm görevleri kontrol eder.

802.15.4 standardını kullanan kablosuz bağlantılar, üç adet endüstriyel bilimsel tıbbi (ISM) frekans bandında çalışabilir. Bunlar, 2.4 GHz bandında 250 kbps, 915 MHz bandında 40 kbps ve 868 MHz bandında 20 kbps kablosuz veri hızlarına uyum sağlar. 802.15.4'te, 2.4 GHz bandında 16 kanal, 915 MHz bandında 10 kanal ve 868 MHz bandında 1 kanal olmak üzere toplam 27 kanal tahsis edilmiştir (Kaushal vd., 2014; Lee & Huang, 2006).

3.1.2 MAC Katmanı

MAC katmanı, fiziksel ve ağ katmanı arasında bir arabirimdir. MAC katmanının fonksiyonu, işaretler (beacon) üretmek ve cihazları işaret sinyali etkinleştirilmiş olan bir şebekede işaret sinyaliyle senkronize etmektir. Ayrıca bağlantı ve bağlantı kesme işlevini gerçekleştirir.

IEEE 802.15.4 MAC'de, dört tip çerçeve yapısını tanımlamıştır: İşaretleyicileri aktarmak için bir koordinatör tarafından kullanılan bir işaret (beacon) çerçevesi. İşaret çerçevesi, onu aldıklarında tekrar uyuyan ve adreslerini dinleyen istemci cihazlarını uyandırır. Tüm veri aktarımları için bir veri çerçevesi kullanılır. Veri çerçevesi, en fazla 104 baytlık bir yük taşır. Çerçevenin başarılı bir şekilde alındığını onaylamak için bir onay (ACK) çerçevesi kullanılır. Alıcıdan göndericiye geri bildirim gönderir ve paketin herhangi bir hata olmadan alındığını onaylar. MAC komut çerçevesi, uzaktan kontrol ve istemci/alıcı düğümlerinin düzeni için bir yöntem sağlar. MAC katmanı, bir CSMA-CA mekanizması kullanarak radyo kanalına erişimi kontrol eder. Sorumlulukları arasında ayrıca, işaret çerçeveleri, senkronizasyon ve güvenilir bir aktarım mekanizması sağlamakta bulunmaktadır (Kaushal vd., 2014; Lee & Huang, 2006).

3.1.3 Ağ Katmanı

Ağ katmanı ZigBee standardının en alt katmanıdır. Ağ katmanı, uygulama katmanı ve MAC Katmanı arasında bir arabirimdir. Bu katmanın işlevleri ağ ve yönlendirmeden ibarettir. Ağ ve yönlendirme işlevleri, bir ağa katılmak ve çıkmak, rota keşfi yapmak, çerçevelere güvenliği uygulamak ve çerçeveleri hedeflenen hedeflere yönlendirmek için kullanılan mekanizmalardan oluşur. Bu katman, düşük güç cihazlarının pil ömrünü uzatmasına yardımcı olur. Koordinatörün ağ katmanı yeni bir ağ başlatır ve yeni bağlanan cihazlara bir adres atar. ZigBee standartlarındaki ağ katmanı yıldız, ağaç ve ağ (mesh) gibi ağ topolojilerini destekler. Tablo 3.1 'de

ZigBee standardındaki ağ katmanını için cihazların işlevsellikleri bakımından karşılaştırılması yapılmıştır (Digi International Inc., 2008; Kaushal vd., 2014; Lee & Huang, 2006).

Tablo 3.1.Ağ katmanındaki ZigBee cihazlarının Karşılaştırılması

ZigBee Ağ Katmanı İşlevleri	Koordinatör	Yönlendirici	Uç Cihaz
ZigBee ağı kurma	✓		
Ağa katılmak veya ayrılmak için diğer cihazlara izin verme	✓	✓	
16-bitlik ağ adresi atama	✓	✓	
Verimli mesaj teslimi için yolları keşfetme ve kaydetme	✓	✓	
Tek-hoplu komşularının listesini keşfetme ve kaydetme	✓	✓	
Ağ paketlerini yönlendirme	✓	✓	
Ağ paketlerini gönderme ve alma	✓	✓	✓
Ağa katılma ve ayrılma	✓	✓	✓
Uyku moduna girme			✓

3.1.4 Uygulama Katmanı

Uygulama Katmanı, protokol yığınının en üst katmanıdır ve uygulama nesnelere tutar. ZigBee standardı, APL katmanını üç ayrı alt katmana böler (Kaushal vd., 2014; Lee & Huang, 2006):

- Uygulama desteği alt katmanı,
- ZigBee cihaz nesnelere,
- Üreticinin tanımladığı nesnelere içeren uygulama çatısı.

3.1.4.1 Uygulama Nesnesi (APO)

Uygulama nesnelere, ZigBee cihazlarındaki katmanları kontrol etmek ve yönetmekle sorumludur. Donanımı kontrol eden bir yazılım türüdür. Her bir uygulama nesnesi, diğer APO'ların etkileşimi için ağ

cihazı adresine bir ek yaparak kullanabileceği belirli bir son nokta (end point) numarasına sahiptir. Bir ZigBee cihazında yaklaşık 240 uygulama nesnesi olabilir (Kaushal vd., 2014).

3.1.4.2 ZigBee Cihaz Nesnesi (ZDO)

ZigBee'nin temel tanımı, güvenlik, servis keşfi ve bağlama gibi üç ana işlevi gerçekleştiren ZigBee cihaz nesnesidir. Keşfin işlevi, düğümleri bulmak ve tek noktaya yayın mesajlarını kullanarak koordinatörün veya yönlendiricinin MAC adresini sormaktır. Bu keşif aynı zamanda profil tanımlayıcıları aracılığıyla bazı hizmetleri bulma prosedürünü de kolaylaştırmaktadır. ZigBee aygıt nesnesi aşağıdaki işlevselliklerden sorumludur (Digi International Inc., 2008; Kaushal vd., 2014; Lee & Huang, 2006);

- APS alt katmanı, NWK katmanı, Güvenlik Hizmeti Sağlayıcısının başlatılması,
- Cihazın çalışma modunun tanımlanması (örn. koordinatör, yönlendirici vb.),
- Cihazın keşfi ve cihazın hangi uygulama servislerini sağladığının belirlenmesi,
- Bağlayıcı istekleri başlatmak ve / veya yanıtlamak,
- Güvenlik yönetimi.

3.1.4.3 Uygulama Desteği Alt Katmanı (APS)

APS alt katmanı, Ağ katmanı ve APL katmanları arasında bir arabirimdir. APS alt katmanı, çerçeveleri güvenli bir şekilde almak / göndermek ve kriptografik anahtarları oluşturmak veya yönetmek için

gelen ve giden çerçeveleri işler. APS alt katmanı aşağıdaki işlevselliklerden sorumludur (Digi International Inc., 2008; Kaushal vd., 2014; Lee & Huang, 2006);

- Bağlayıcı (Binding) tabloları,
- Bağlı cihazlar arasında mesaj iletmek,
- Grup adresi tanımlama ve idaresi,
- 64-bit uzatılmış adreslerden 16-bit NWK adreslerine adres eşleştirmesi,
- Paketlerin parçalanması ve yeniden birleştirilmesi,
- Veri aktarımının güvenliği,
- APS bilgi tabanı olarak bilinen yönetilen nesnelerin veritabanınının bakımı.

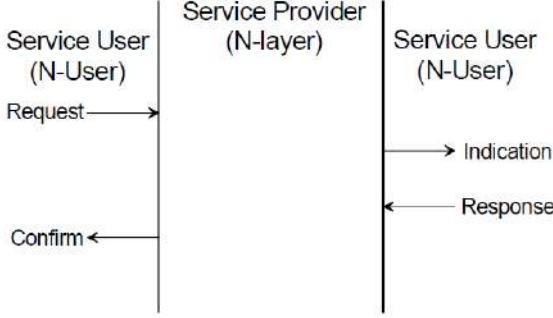
3.1.4.4 Güvenlik Hizmeti Sağlayıcısı

ZigBee, her biri kendi çerçevelerini korumaktan sorumlu olan ağ katmanı ve uygulama katmanı için güvenlik yöntemleri sunar. Güvenlik hizmetleri, anahtar kurma, çerçeve koruması, anahtar taşıma ve cihaz yönetimi için yöntemlere sahiptir (Kaushal vd., 2014).

3.2 Primitif Kavramı

Protokol, bir ağdaki düğümler arasındaki haberleşmeyi yöneten kuralların bir kümesidir. İki düğümün birbiriyle konuşabilmesi için aynı dili kullanması gerekir. Yani protokoller, verilerin düğümler ve ağlar arasında nasıl iletileceği, hata denetiminin nasıl olacağı ve veri sıkıştırmasının nasıl olacağı vb. gibi konuları tanımlar. Bir düğüm (hangi işletim sistemi, ağ kartı veya uygulama kullanıyorsa kullansın)

dünyanın dört bir yanında bulunan başka bir düğümle iletişim kurabildiğinden emin olmak için birçok farklı ağ protokolü ve standardı gerekir (Winkelman, 2025).



Şekil 3.2.Hizmet Primitifi (IEEE Computer Society, 2003).

Katmanlı bir mimaride (örn. OSI Referans Modeline dayanan) iletişim kuran her protokol, uzak protokol varlığı ile uçtan uca bir tarzda iletişim kurar. Komşu protokol katmanları arasındaki iletişim (yani aynı haberleşme düğümü üzerinde), katmanlar arasında primitif olarak adlandırılan fonksiyonlar çağrılarak yönetilir. Her bir primitif, gerçekleştirilecek eylemi belirler veya daha önce talep edilen bir eylemin sonucunu bildirir. Bir primitif, işlevlerini yerine getirmek için gerekli parametreleri taşır. Bir parametre, üst ya da alt katmana gönderilecek / alınacak olan bir veri yapısının (structure) adresidir. Bu veri yapısında ihtiyaç duyulduğunda paket veya diğer parametreler de yer alabilir (Fairhurst, 2025).

Kısaca Primitif, servis sağlayıcı tarafından komşu katmanlardaki kullanıcıya sunulan bir dizi hizmet olarak tanımlanabilir. Şekil 3.2’de hem primitif kavramı gösterilirken hem de hizmet hiyerarşisi ve iki eş

N-kullanıcısının (N-User) ilişkisi ve N-katmanın aynı protokol varlıkları ile ilişkisini göstermektedir. Servisler, N-kullanıcı ve N-katman arasındaki bilgi akışını tanımlayarak belirlenir. Bu bilgi akışı, bir hizmet sunumunu karakterize eden ayrı, anlık olaylar tarafından modellenmiştir. Her olay, bir N-kullanıcısı ile ilişkilendirilmiş bir SAP vasıtasıyla diğer bir katmana geçmesinden meydana gelir. Servis primitifi, belirli bir hizmeti sağlayarak gerekli bilgileri iletir. Bu hizmet ilkeleri bir soyutlamadır, çünkü sağladıkları araçlardan ziyade yalnızca sunulan hizmeti belirtirler.

Hizmetler, onu karakterize eden parametreler ve hizmet primitiflerini açıklayarak belirtilir. Bir hizmet, söz konusu hizmetle ilgili aktiviteyi oluşturan bir veya daha fazla ilgili primitif maddeye sahip olabilir. Her hizmet primitifi, hizmeti sağlamak için gereken bilgileri ileten sıfır veya daha fazla parametreye sahip olabilir. Dört tane genel primitif çeşidi vardır:

- **İstek (Request)** : Bir hizmetin başlatılmasını istemek için N-kullanıcısından N-katmanına geçirilir.
- **İşaret (Indication)** : N-kullanıcı için önemli olan bir iç N-katmanı olayını göstermek için N-katmanından N-kullanıcısına geçirilir. Bu olay bir uzaktan hizmet isteği ile mantıksal olarak ilişkili olabilir veya bir N katmanı iç olayından kaynaklanabilir.
- **Yanıt (Response)** : Daha önce bir işaret primitifi tarafından başlatılan bir prosedürü tamamlamak için N-kullanıcıdan N-katmanına geçirilir.

- **Doğrulama (Confirm)** : Bir veya daha fazla ilişkili önceki hizmet isteklerinin sonuçlarını aktarmak için N katmanından N kullanıcıya geçirilir.

Primitif kavramının daha iyi anlaşılması için IEEE 802.15.4 standartlarında yer alan MLME-ASSOCIATE.request primitifini kısa bir şekilde inceleyelim:

Aşağıda bu primitifin genel yapısı görülmektedir. Bu genel yapı da “MLME-ASSOCIATE.request” ifadesi üyelikle ilgili bir talep primitifinin adını temsil ederken “LogicalChannel”, “CoordAddrMode”, “CoordAddress”, “SecurityEnable” vb. gibi ifadelerde bu primitifin parametrelerini temsil eder.

```
MLME-ASSOCIATE.request(  
    LogicalChannel,  
    CoordAddrMode,  
    CoordPANId,  
    CoordAddress,  
    CapabilityInformation,  
    SecurityEnable  
)
```

Bu primitif bir FFD düğüme katılmak için bir RFD düğümün NWK katmanından MAC katmanına emir göndererek o katmanda bir üyelik komut çerçevesinin yaratılmasını sağlar ve NWK katmanından MAC katmanına gelirken beraberinde NWK katmanından bazı parametreleri

de getirir; bu parametrelerden “LogicalChannel”i fiziksel katmanda kullanmak üzere getirirken Bilgi Tabanında (Information Base) yer alan “*phyCurrentChannel*” parametresine “LogicalChannel” parametresinin deęerini atar ve “CoordPANId” parametresinin deęerini de yine Bilgi Tabanında yer alan “*macPANId*”ye atayarak g¼nceller. Bilgi Tabanında (Information Base) bu Őekilde deęerler saklanarak t¼m fonksiyonlar tarafından kullanılması saęlanır. Eęer bir fonksiyon tarafından bu parametrelerin deęerlerinden biri deęiŐtirilirse anında dięer t¼m fonksiyonlarda deęiŐtirilen son deęer geęerli olur. “CoordAddrMode” parametresi FFD d¼ę¼me g¼nderilecek ęeręevenin 16-bitlik kısa adresi mi ya da 64 bitlik uzun adresi mi kullanacaęını belirtir. “CoordAddress” parametresi de “CoordAddrMode” parametresinde belirtildięi gibi d¼ę¼m¼n adresini ięerir. “CapabilityInformation” parametresinde ¼ye olacak olan d¼ę¼m¼n iŐlevsel yetenekleri belirtilir. Bu parametre bitmap tipindedir ve asıl bu parametrede bir ¼yelik isteęinin talep edildięi belirtilir. “SecurityEnable” parametresi ise g¼nderilecek ęeręevenin g¼venlikli yada g¼venliksiz gideceęi 0 yada 1 Őeklinde belirtilir. Eęer g¼venlikliyse MAC komut ęeręevesinin y¼k alanı (payload field) ilgili Őifreleme anahtarı ile Őifrelenir ve gizli metin (cipher text) olarak g¼nderilir. Aksi halde ęeręevenin y¼k alanı (payload field) aęık metin (plain text) olarak g¼nderilir.

4. ZIGBEE DÜĞÜMÜ MODELLEME

Bu bölümde, Opnet Modellerin eğitimsel versiyonu (deneme sürümü) kullanılarak bir koordinatör düğümünden uç cihazlara ağ adreslerinin atamasını gerçekleştirmek üzere düğüm modellemesinin IEEE 802.15.4 ve ZigBee protokol standartlarına uygun olarak gerçekleştirilmesi açıklanmıştır. İlk olarak, uç cihaz ve koordinatör düğümlerinin birbiriyle haberleşmesini sağlamak için IEEE 802.15.4 Mac çerçeve formatları sunulmuştur. Ardından, OPNET Modeller'de bu mac çerçeve formatlarının ve ve bir düğümün katmanlar arası iletişimini sağlayan ve primitifleri taşıyan çerçeve formatının modellenmesi gerçekleştirilmiştir. İkinci olarak, bir ağa katılma için koordinatör ve uç düğümde gerçekleşen prosedürlerin teorisi primitif türleriyle desteklenerek açıklanmıştır. Üçüncü olarak, bir ZigBee düğümünün fiziksel, mac, ağ ve uygulama katmanlarının yapısı ve her bir katman için süreç yapıları, primitifler ve düğüm ayarlarıyla desteklenerek modellenmesinin nasıl gerçekleştirildiği açıklanmıştır.

4.1 Modellemede Kullanılan Çerçeve Formatları

Bu alt bölümde koordinatör ve uç cihazlar arasındaki haberleşme için kullanılan ZigBee protokol çerçeve formatları IEEE 802.15.4 standartlarına göre ele alınmıştır.

4.1.1 Genel MAC Çerçeve Formatı

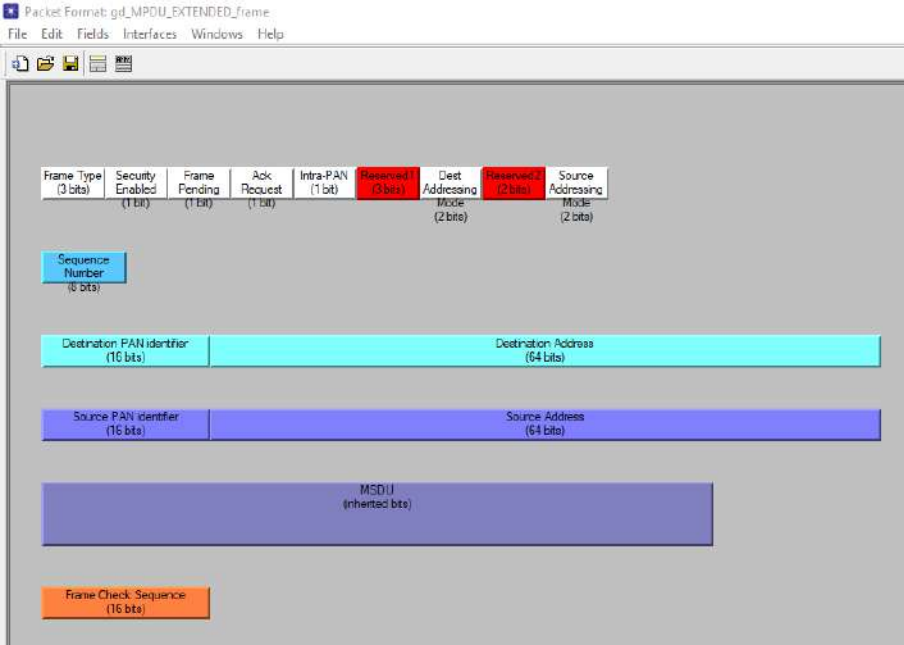
ZigBee protokolünün MAC katmanını tanımlayan IEEE 802.15.4 standartına göre genel bir MAC çerçevesi üç tane temel bileşenden oluşur;

- **MHR (MAC Header- MAC çerçeve başlığı)**, çerçeve kontrolü, sıra numarası ve adres bilgilerini içerir.
- **MAC yükü (payload)**, çerçeve türüne özgü bilgileri yani üst katmanın çerçevesini içerir ve değişken uzunluktadır.
- **MFR (MAC Footer – MAC çerçeve alt başlığı)**, sadece FCS içerir.

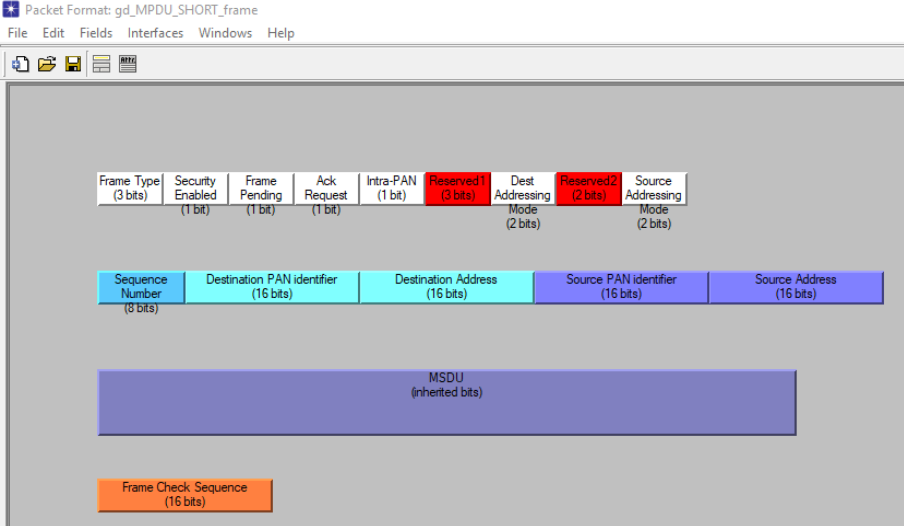
Tablo 4.1.Genel MAC Çerçeve Formatı (IEEE Computer Society, 2003).

Octets:2	1	0/2	0/2/8	0/2	0/2/8	Değişken	2
Çerçeve Kontrolü	Sıra Numarası	Hedef PAN Tanımlayıcı	Hedef Adres	Kaynak PAN Tanımlayıcı	Kaynak Adres	Çerçeve Yükü	FCS
		Adresleme Alanları					
		MHR				MAC yükü	MFR

Tablo 4.1’de verilen genel MAC çerçevesinin formatı OPNET Modeller’de iki farklı çerçeve formatı tasarlanarak gerçekleştirilmiştir. ZigBee düğümleri ağa ilk katılım esnasında veya kısa adres alamadığında uzun adresle iletişim kurmak zorunda olduğu için uzun adresli alana sahip olan bir MPDU_EXTENDED_frame adlı MAC çerçeve formatı tasarlanmıştır ve şekil 4.1’de verilmiştir.



Şekil 4.1. Uzun adresli MAC çerçeve formatı



Şekil 4.2. Kısa adresli MAC çerçeve formatı

Bir diğ er MAC çerç e ve formatı, ZigBee düğ ümleri ađ a katılım sađ l adıđ ında koordinatör düğ üm tarafından uç düğ ümlere atanan kısa adresle iletiř im kurmak zorunda olduđu için ksa adresli alana sahip olan bir MPDU_SHORT_frame adlı MAC çerç e ve formatı tasarlanmıř tır ve ř ekil 4.2’de verilmiř tir.

4.1.1.1 Ç erç e ve Kontrolü Alanı

Ç erç e ve kontrolü alanı 16 bit uzunluđ undadır ve çerç e ve türünü, adresleme alanlarını ve diğ er kontrol bayraklarını tanımlayan bilgileri içerir.

Tablo 4.2.Ç erç e ve Kontrol Alanının Formatı (IEEE Computer Society, 2003).

Bits:0-2	3	4	5	6	7-9	10-11	12-13	14-15
Ç erç e ve Tipi	Güvenlik Etkin	Ç erç e ve Bekleme	Onay İř teđ i	PAN İ ç i	Ayrılmıř	Hedef Adresleme Modu	Ayrılmıř	Kaynak Adresleme Modu

IEEE 802.15.4’e göre çerç e ve kontrol alanı Tablo 4.2’de gösterildiđ i gibidir. Buna göre, OPNET Modelerde tasarlanan çerç e ve kontrol alanının formatı ise ř ekil 4.3’deki gibidir.

Frame Type (3 bits)	Security Enabled (1 bit)	Frame Pending (1 bit)	Ack Request (1 bit)	Intra-PAN (1 bit)	Reserved1 (3 bits)	Dest Addressing Mode (2 bits)	Reserved2 (2 bits)	Source Addressing Mode (2 bits)
------------------------	-----------------------------	--------------------------	------------------------	----------------------	-----------------------	-------------------------------------	-----------------------	---------------------------------------

ř ekil 4.3. OPNET Modelerde tasarlanan çerç e ve kontrol alanı

4.1.1.1.1 Çerçeve Tipi Alt Alanı

Çerçeve türü alt alanı 3 bit uzunluğundadır ve Tablo 4.3'de listelenen ama ayrılmış olmayan değerlerden birine ayarlanır.

Tablo 4.3.Çerçeve tipi alt alanının değerleri (IEEE Computer Society, 2003).

Çerçeve Tipi Değeri	Tanımlama
000	İşaret Çerçevesi
001	Veri Çerçevesi
010	Onay Çerçevesi
011	MAC Komut Çerçevesi
100-111	Ayrılmış

4.1.1.1.2 Güvenlik Etkin Alt Alanı

Güvenlik etkin alt alanı 1 bit uzunluğundadır ve çerçeve MAC alt katmanı tarafından şifreleme yoluyla korunmuyorsa, 0 olarak ayarlanır. Güvenlikli alt alan 1'e ayarlanırsa, çerçeve mevcut çerçeveye gösterilen güvenlik ilişkisi için MAC PIB'de saklanan/depolanan anahtarlar kullanılarak korunur.

4.1.1.1.3 Çerçeve Bekleme Alt Alanı

Çerçeve bekleyen alt alan uzunluğu 1 bittir ve çerçeve gönderen cihazın mevcut aktarımı takiben alıcıya göndermek için ek veri varsa, 1'e ayarlanır. Daha fazla veri bekleniyorsa, alıcı cihaza başka bir veri talep komutu göndererek onları geri alacaktır. Çerçeveyi gönderen cihazın alıcı için daha fazla verisi yoksa, bu alt alan 0'a ayarlanacaktır.

4.1.1.1.4 Onay İsteđi Alt Alanı

ACK isteđi alt alanı 1 bit uzunluđundadır ve alıcı cihazdan bir data ya da MAC komut çerçevesinin alındısının onayı gerekip gerekmediđi belirtilir. Bu alt alan 1'e ayarlanırsa, alıcı cihaz, çerçevenin geçerli olduđunu belirledikten sonra bir onay çerçevesi gönderecektir. Bu alt alan 0'a ayarlanırsa, alıcı cihaz, çerçevenin geçerli olduđunu belirledikten sonra bir onay çerçevesi göndermeyecektir.

4.1.1.1.5 PAN-İçi Alt Alanı

PAN-İçi alt alanı 1 bit uzunluđundadır ve MAC çerçevesinin aynı PAN (intra-PAN) içinde olacađını veya başka bir PAN'a (inter-PAN) gönderileceđini belirtir. Bu alt alan 1'e ayarlanır ve hem hedef hem de kaynak adresleri mevcutsa, çerçeve kaynak PAN tanımlayıcı alanını içermeyecektir. Bu alt alan 0'a ayarlanır ve hem hedef hem de kaynak adresleri mevcutsa, çerçeve hedef ve kaynak PAN tanımlayıcı alanlarını içerecektir.

4.1.1.1.6 Hedef Adresleme Modu Alt Alanı

Hedef adresleme modu alt alanı 2 bit uzunluđundadır ve Tablo 3.4'de listelenen deđerlerden birine ayarlanmalıdır. Bu alt alan 0'a eşitse ve çerçeve türü alt alanı, bu çerçevenin ACK veya işaret çerçevesi olduđunu belirtmezse, kaynak adresleme modu alt alanı sıfır olmamalıdır; bu, çerçevenin, kaynak PAN tanımlayıcı alanında belirtildiđi gibi PAN tanımlayıcısıyla PAN koordinatörüne yönlendirildiđini ima eder.

Tablo 4.4.Hedef ve kaynak adresleme modu alt alanının deęerleri (IEEE Computer Society, 2003).

Adresleme Modu Deęeri	Tanımlama
00	PAN tanımlayıcı ve adres alanı mevcut deęil.
01	Ayrılmıř.
10	Adres alanı 16-bit kısa adres ierir.
11	Adres alanı 64-bit uzun adres ierir.

4.1.1.1.7 Kaynak Adresleme Modu Alt Alanı

Kaynak adresleme modu alt alanı 2 bit uzunluęundadır ve Tablo 4.4'de listelenen deęerlerden birine ayarlanmalıdır. Bu alt alan 0'a eřitse ve ereve tur alt alanı bu erevenin bir onaylama erevesi olduęunu belirtmezse, hedef adresleme modu alt alanı sıfır olmamalıdır; bu, erevenin hedef PAN tanımlayıcısı alanında belirtildięi gibi PAN tanımlayıcısı ile PAN koordinatrnden kaynaklandıęı belirtilir.

4.1.1.2 Sıra Numarası Alanı

Sıra numarası alanı uzunluęu 8 bittir ve ereve iin benzersiz bir sıra tanımlayıcısı belirtir. Bir beacon erevesi iin sıra numarası alanı bir BSN'yi belirtir. Her koordinatr mevcut BSN deęerini MAC PIB znitelięi macBSN'de saklar ve rastgele bir deęerle bařlatır. Rastgele bir sayı seme algoritması, bu standardın kapsamı dıřındadır. Koordinatr, macBSN znitelięinin deęerini bir beacon erevesinin sıra numarası alanına, her biri oluřturulduęunda kopyalar ve macBSN'yi birer birer artırır.

Bir veri, onay veya MAC komut çerçevesi için, sıra numarası alanı, bir onay çerçevesini veri veya MAC komut çerçevesine eşleştirmek için kullanılan bir veri sıra numarası (DSN) belirtir. Her bir cihaz, iletişim kurmak istediği benzersiz cihazların sayısına bakılmaksızın tam olarak bir DSN'yi destekleyecektir. Her bir aygıt mevcut DSN değerini MAC PIB özneliği macDSN'sinde saklar ve rastgele bir değerle başlatır. Rastgele bir sayı seçme algoritması bu standardın kapsamı dışındadır. Cihaz macDSN özneliğinin değerini bir veri veya MAC komut çerçevesinin sıra numarası alanına her ne zaman üretilirse kopyalar ve macDSN'yi birer birer arttıracaktır.

Bir ACK alındığı takdirde, alıcı cihaz ACK çerçevesine karşılık gelen DSN alanının içine veri ya da MAC komut çerçevesinde alınan DSN'yi kopyalar. Eğer ACK, macAckWaitDuration simgelerinden sonra gelmediyse, üreten/gönderen cihazın MAC alt katmanı, orijinal iletimde kullanılanla aynı DSN'yi kullanarak çerçeveyi yeniden iletir.

4.1.1.3 Hedef PAN Tanımlayıcı Alanı

Hedef PAN tanımlayıcı alanı 16 bit uzunluğundadır ve çerçevenin istenen alıcısının benzersiz/eşsiz PAN tanımlayıcısını belirtir. Bu alanda 0xffff'lik bir değer, şu anda kanalı dinleyen tüm cihazlar tarafından geçerli bir PAN tanımlayıcısı olarak kabul edilecek olan yayın (broadcast) PAN tanımlayıcısını temsil edecektir. Bu alan, sadece çerçeve kontrol alanının hedef adresleme modu alt alan sıfırdan farklıysa, MAC çerçevesine dahil edilecektir.

4.1.1.4 Hedef Adres Alanı

Hedef adres alanı, çerçeve kontrol alanının hedef adresleme modu alt alanında belirtilen değere göre 16 bit veya 64 bit uzunluğundadır ve çerçevenin istenen alıcısının adresini belirtir. Bu alandaki 0xffff olan 16 bitlik bir değer, şu anda kanalı dinleyen tüm cihazlar tarafından geçerli bir kısa adres olarak kabul olunan yayın (broadcast) kısa adresi temsil eder. Bu alan, sadece çerçeve kontrol alanının hedef adresleme modu alt alan sıfırdan farklıysa, MAC çerçevesine dahil edilecektir.

4.1.1.5 Kaynak PAN Tanımlayıcı Alanı

Kaynak PAN tanımlayıcı alanı 16 bit uzunluğundadır ve çerçevenin yaratıcısının/göndericisinin benzersiz PAN tanımlayıcısını belirtir. Bu alan, MAC çerçeve içine, yalnızca sırasıyla çerçeve kontrol alanının kaynak adresleme modu ve PAN-İçi alt alanları sıfırdan farklı ve sıfıra eşitse dahil edilecektir.

4.1.1.6 Kaynak Adres Alanı

Kaynak adresi alanı, çerçeve kontrol alanının hedef adresleme modu alt alanında belirtilen değere göre 16 bit veya 64 bit uzunluğundadır ve çerçevenin göndericisinin/yaratıcısının adresini belirtir. Bu alan, sadece çerçeve kontrol alanının kaynak adresleme modu alt alanı sıfırdan farklıysa, MAC çerçevesine dahil edilecektir.

4.1.1.7 Çerçeve Yüğü Alanı

Çerçeve yüğü alanı değişken uzunluktadır ve özel çerçeve türlerine özgü bilgileri içerir. Eğer güvenlik etkin alt alanı çerçeve kontrol

alanındaki 1'e ayarlanırsa, çerçeve yükü bu ilişki için seçilen güvenlik paketi/takımı tarafından tanımlandığı şekliyle korunur.

4.1.1.8 Çerçeve Kontrol Dizisi (FCS) Alanı

FCS alanı 16 bit uzunluğundadır ve bir 16 bit ITU-T CRC içerir. FCS, çerçevenin MHR ve MAC yükü kısımları üzerinden hesaplanır ve FCS alanına eklenir. Çerçevenin gerekli tüm alanları doldurulduktan sonra gönderilir ve bu çerçeveyi alan cihaz tekrar gelen çerçevenin FCS bitlerini oluşturur eğer çerçeveye gelen FCS bitleriyle aynı çıkarsa çerçeve değişikliğe uğramadan cihaza ulaşmıştır. Yani FCS bitleri çerçevenin yolda bozulup bozulmadığı anlamak için gereklidir.

4.1.2 MAC Komut Çerçeve Formatı

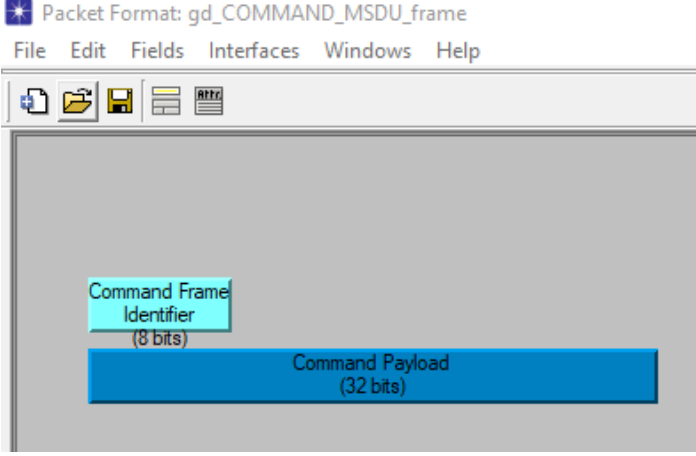
Mac komut çerçevesinin formatı Tablo 4.5'da gösterildiği gibidir. Bu tablodaki Adresleme Alanı Tablo 4.1'de gösterilen genel Mac çerçeve formatındaki gibidir. Bu tablodaki temel farklılık MAC yükü alanında komut çerçevesi tanımlayıcı ve komut yükünün bulunmasıdır.

Tablo 4.5.MAC Komut Çerçevesinin Formatı (IEEE Computer Society, 2003).

Octets:2	1	<u>bkz. Tablo 4.1</u>	1	Değişken	2
Çerçeve Kontrolü	Sıra Numarası	Adresleme Alanları	Komut Çerçevesi Tanımlayıcı	Komut Yükü	FCS
MHR			MAC Yükü		MFR

Tablo 4.5'deki MAC komutu çerçevesinin MAC yükü alanı OPNET Modelerde Şekil 4.4 gibi tasarlanmıştır. Bu çerçeve kullanıldığında

Şekil 4.1 ve 4.2'deki MAC çerçeve formatının MAC yükü alanına gömülerek gönderilir.



Şekil 4.4. Komut çerçevesi için OPNET Modellerde tasarlanan MAC yükü alanı

4.1.2.1 MAC Komut Çerçevesinin MHR Alanı

Bir MAC komut çerçevesi için MHR, çerçeve kontrol alanını, sıra numarası alanını, hedef PAN tanımlayıcısını alanını, hedef adres alanını, kaynak PAN tanımlayıcı alanını ve kaynak adres alanını içermelidir.

Çerçeve kontrolü alanında, çerçeve türü alt alanı Tablo 4.3'de gösterildiği gibi bir MAC komut çerçevesinde belirten değeri içermelidir. Diğer tüm alt alanlar, MAC komut çerçevesinin amaçlanan kullanımına uygun olarak ayarlanacaktır.

Sıra numarası alanı, macDSN'in mevcut değerini içerir. Adresleme alanları, çerçeve kontrol alanındaki ayarlara bağlı olarak hedef adres alanlarını ve / veya kaynak adres alanlarını içerir.

4.1.2.2 Komut Çerçevesi Tanımlayıcı Alanı

Komut çerçevesi tanımlayıcı alanı, kullanılan MAC komutunu tanımlar. Bu alan, Tablo 4.6'de listelenmiş olan fakat ayrılmamış değerlerden birine ayarlanacaktır.

4.1.2.3 Komut Yüğü Alanı

Komut yüğü alanı MAC komutunun kendisini içerir. Giden bir MAC komut çerçevesinde güvenlik gerekliyse, komut yüğü alanındaki oktet dizisi ya hedef adrese mevcutsa yada eğer hedef adres alanı mevcut değilse macCoordExtendedAddress'e (PIB özniteliklerinden biri) karşılık gelen güvenlik paketine göre işlenir. Gelen bir çerçevenin çerçeve kontrol alanının güvenli alt alanı 0'a ayarlanırsa, komut yüğü alanı, istenen MAC komutu içerecektir. Gelen bir çerçevenin çerçeve kontrol alanının güvenli alt alanı 1'e ayarlanırsa, cihaz istenen MAC komutunu belirlemek için seçilen güvenlik paketine/takımına göre komut yüğü alanını işleyecektir.

Bireysel/özel komutların biçimleri 4.1.2'de açıklanmaktadır.

4.1.3 MAC Komut Çerçevesi

MAC alt katmanı tarafından tanımlanan komut çerçevesi Tablo 4.6'da listelenmiştir. FFD, tüm komut çerçevesi tiplerini iletme ve

alma yeteneğine sahipken, bir RFD için gereklilikler tabloda gösterilmektedir.

Tablo 4.6.MAC komut çerçeveleri (IEEE Computer Society, 2003).

Komut Çerçeve Tanımlayıcı	Komut Adı	RFD	
		Tx	Rx
0x01	Üyelik isteği	x	
0x02	Üyelik yanıtı		x
0x03	Üyelikten ayrılma bildirimi	x	x
0x04	Veri isteği	x	
0x05	PAN ID çatışma bildirimi	x	
0x06	Yetim bildirimi	x	
0x07	İşaret isteği		
0x08	Koordinatörün yeniden düzenlenmesi		x
0x09	GTS isteği		
0x0A-0xFF	Ayrılmış		

4.1.3.1 Üye Olma Ve Üyelikten Ayrılma

Üye olma (Association) ve üyelikten ayrılma (disassociation) komutları, cihazların bir PAN ile üye olmasına veya üyelikten ayrılmasına izin vermek için kullanılır.

4.1.3.1.1 Üyelik İsteği Komutu

Üyelik isteği komutu, bir cihazın bir koordinatöre üye olma isteğini sağlar. Bu komut sadece bir PAN ile ilişkilendirmek isteyen ilişkisiz/üye olmayan bir cihaz tarafından gönderilir. Bir RFD'nin bunu almasına ihtiyaç olmamasına rağmen, tüm cihazlar bu komutu gönderme yeteneğine sahiptir. Üyelik isteği komutu Tablo 4.7'de gösterildiği gibidir.

Tablo 4.7. Üye olma isteđi komutunun formatı (IEEE Computer Society, 2003).

Octets: 17/23	1	1
MHR Alanı	Komut Çerçevesi Tanımlayıcı	Yetenek Bilgisi

Bu komut formatının komut çerçevesi tanımlayıcısı ve yetenek bilgisi alanlarında Şekil 4.4'te tasarlanan çerçevede ilgili alanlarla eşleşecek şekilde kullanılır. Ancak modellemeyi kolaylaştırmak için bunların MAC Yükü (MSDU) alanına bu komuttaki yetenek bilgisi alanını tanımlayan bir veri yapısı ile paket oluşturulması kod ortamında tercih edilmiştir.

4.1.3.1.1.1 MHR Alanı

Çerçeve kontrol alanının kaynak adresleme modu alt alanı 3 (64 bit genişletilmiş adresleme) olarak ayarlanacaktır. Güvenlik üye olma isteđi komutu için kullanılıyorsa, güvenli alt alan 1'e ayarlanacak ve çerçeve, hedef adresine karşılık gelen güvenlik paketi/takımı tarafından tanımlanan metoda göre işlenecektir. Aksi halde, güvenli alt alan 0'a ayarlanacaktır. Çerçeve kontrol alanının çerçeve bekleyen alt alanı 0'a ayarlanır ve alım üzerine göz ardı edilir ve ACK isteđi alt alanı 1'e ayarlanır. Hedef PAN tanımlayıcı alanı, üye olacağı PAN'ın tanımlayıcısını içerecektir. Hedef adres alanı, üye olma isteđi komutunun gönderildiđi koordinatöre gönderilen işaret çerçevesindeki adresi içerir. Kaynak PAN tanımlayıcı alanı, broadcast PAN tanımlayıcısını içerecektir (yani, 0 x ffff). Kaynak adres alanı, aExtendedAddress değerini içerir.

4.1.3.1.1.2 Yetenek Bilgisi Alanı

Yetenek bilgisi alanı Tablo 4.8’de gösterildiği gibi biçimlendirilir (IEEE Computer Society, 2003).

Tablo 4.8. Yetenek bilgisi alanının formatı (IEEE Computer Society, 2003).

Bits: 0	1	2	3	4-5	6	7
Alternatif PAN Koordinatörü	Cihaz Tipi	Güç Kaynağı	Boşta Olduğunda Alıcı	Ayrılmış	Güvenlik Yeteneği	Adres Ayırma

Alternatif PAN kodlayıcı alt alanı 1 bit uzunluğundadır ve cihaz bir PAN koordinatörü olabiliyorsa 1 olur. Aksi takdirde, alternatif PAN koordinatör alt alanı 0'a ayarlanacaktır.

Cihaz tipi alt alanı 1 bit uzunluğundadır ve cihaz FFD ise 1 olarak ayarlanır. Aksi takdirde, RFD'yi belirtmek için cihaz tipi alt alanı 0'a ayarlanacaktır.

Güç kaynağı alt alanı 1 bit uzunluğundadır ve cihaz alternatif akım şebekesinden güç aldığı takdirde 1 olarak ayarlanır. Aksi takdirde, güç kaynağı alt alanı 0'a ayarlanacaktır.

Boşta (idle) olduğunda alıcı alt alan uzunluğu 1 bit olduğunda olur ve cihaz bekleme (idle) süresince gücü korumak için alıcıyı devre dışı bırakmazsa 1 olur. Aksi takdirde alıcı, boşta olan alt alan açık olduğunda 0'a ayarlanacaktır.

Güvenlik yeteneği alt alanı 1 bit uzunluğundadır ve cihaz, güvenlik takımı kullanarak güvenli MAC çerçeveleri gönderebiliyor ve

alabiliyorsa 1 olarak ayarlanacaktır. Aksi halde, güvenlik yeteneđi alt alanı 0'a ayarlanacaktır.

Adres ayırma alt alanı bir bit uzunluđundadır ve cihaz koordinatöre, üye olma prosedürünün bir sonucu olarak kısa bir adres tahsis etmek isterse, 1 olarak ayarlanır. Bu alt alan 0'a ayarlanırsa, 0xffff'nin özel kısa adresi cihaza tahsis edilecek ve üye olma yanıt komutuyla geri gönderilecektir. Bu durumda, cihaz sadece 64 bit genişletilmiş adresini kullanarak PAN üzerinden iletişim kuracaktır.

4.1.3.1.2 Üyelik Yanıt Komutu

Üyelik yanıt komutu, koordinatörün bir üye olma girişiminin sonuçlarını, üyelik isteyen cihaza geri iletmesine izin verir. Bu komut, koordinatör tarafından yalnızca şu anda üye olmaya deneyen çalışan bir cihaza gönderilir. RFD'nin bunu iletilebilme yeteneđi olmamasına rağmen, tüm cihazlar bu komutu alabilecek yetenekte olmalıdır. Üyelik yanıt komutu Tablo 4.9'da gösterildiđi gibi biçimlendirilecektir.

Tablo 4.9.Üyelik yanıt komutunun formatı (IEEE Computer Society, 2003).

Octets: 23	1	2	1
MHR Alanı	Komut Çerçevesinin Tanımlayıcısı	Kısa Adres	Üyelik Durumu

4.1.3.1.2.1 MHR Alanı

Çerçeve kontrol alanının hedef adresleme modu ve kaynak adresleme modu alt alanları her biri 3'e ayarlanmalıdır (yani, 64 bit genişletilmiş adresleme).

Güvenlikli bir üyelik yanıt komutu için kullanılıyorsa, güvenlikli alt alan 1'e ayarlanacak ve çerçeve, hedef adresine karşılık gelen güvenlik paketi tarafından tanımlanan metoda göre işlenecektir. Aksi halde, güvenlikli alt alan 0'a ayarlanacaktır.

Çerçeve kontrol alanının çerçeve bekleyen alt alan 1 0'a ayarlanır ve alım üzerine göz ardı edilir ve ACK isteği alt alanı 1'e ayarlanır.

Hedef ve kaynak PAN tanımlayıcı alanları, macPANId değerini içerir. Hedef adres alanı, üyelik talep eden cihazın genişletilmiş adresini içerecektir. Kaynak adres alanı, aExtendedAddress değerini içerir.

4.1.3.1.2.2 Kısa Adres Alanı

Kısa adres alanı 16 bit uzunluğundadır. Koordinatör bu cihazı PAN'a üye yapmadıysa, bu alan 0xffff olarak ayarlanacak ve üyelik durumu alanı başarısızlığın sebebini içerecektir. Koordinatör, cihazı PAN'ına üye olabiliyorsa, bu alan, cihazı üyelikten çıkarana kadar PAN'daki iletişimde kullanabileceği kısa adresi içermelidir. 0xffff'e eşit kısa bir adres alanı değeri, cihazın bir PAN ile başarıyla üye olduğu ancak kısa bir adres tahsis edilmediğini gösterecektir. Bu durumda, cihaz sadece 64 bit genişletilmiş adresini kullanarak PAN üzerinden iletişim kuracaktır.

4.1.3.1.2.3 Üyelik Durumu Alanı

Üyelik durumu alanı 8 bit uzunluğundadır ve Tablo 4.10’da listelenen ayrılmamış değerlerden birini içerecektir (IEEE Computer Society, 2003).

Tablo 4.10.Üyelik durumu alanının geçerli değerleri (IEEE Computer Society, 2003).

Üyelik Durumu	Açıklama
0x00	Üyelik başarılı.
0x01	PAN’nın kapasitesi dolu.
0x02	PAN’a erişim reddedildi.
0x03-0x7F	Ayrılmış.
0x80-0xFF	MAC primitif sayım değerleri için ayrılmıştır.

4.1.4 ZigBee düğümünün tasarlanma süreçleri

Bu bölümde, ZigBee temelli bir düğümün, belirli bir ZigBee ağına katılma sürecindeki kullanılan primitiflerin yapıları ve Opnet Modeler platformunda ZigBee’yi temel alan bir düğüm yapısının tasarlanması için teorik bilgiler bu alt bölümde açıklanacaktır. Bu çalışmada, ZigBee ağına katılacak düğümün ağ keşfini yaptığı, kanal ve PAN tanımlayıcı gibi parametrelerin belirlenmiş olduğu varsayılmıştır. PAN ID’si belirli bir ağa katılma isteğinde bulunan bir düğüme, ağa katılma iznini verildikten sonra ağ adresinin atamasının yapılması süreci üzerine odaklanılmıştır.

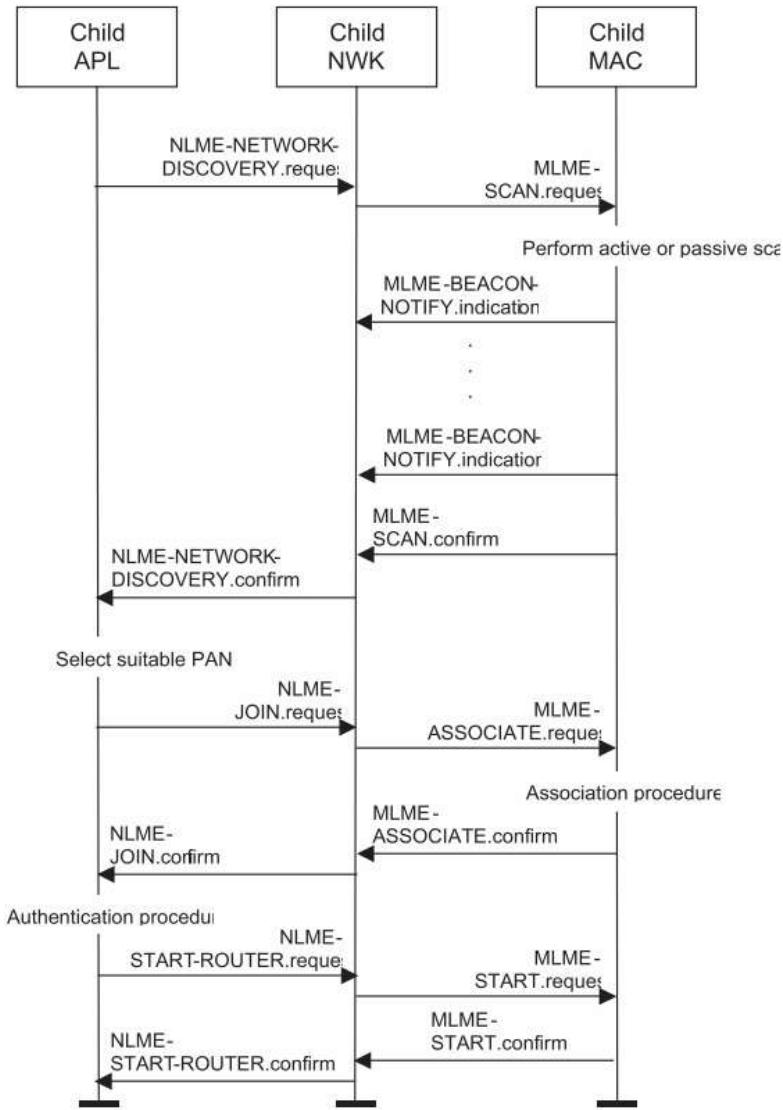
4.1.4.1 ZigBee Ağına Katılma Süreci

Bir ağdaki koordinatör düğüm veya yetkili düğüm, ağa ilk defa katılmak isteyen yeni bir cihaza izin verdiği zaman bu iki cihaz arasında ebeveyn-çocuk (parent-child) ilişkisi oluşur. Yeni cihazın ağa katılması üzerine ağda zaten üyeliği bulunan cihaz ebeveyn olurken, yeni cihaz ise çocuk olur.

Bir çocuk cihaz eğer MAC tarafından sağlanan temel üyelik yeteneklerini kullanarak bir ağa katılmayı tercih ederse, ebeveyn cihaz olan bir ZigBee koordinatör veya yönlendirici prosedürü MAC'den MLME-ASSOCIATE.request primitifi almasını takip etmelidir.

4.1.4.1.1 Çocuk Düğüm (Uç Cihaz) Prosedürü

MAC katman ilişkilendirme prosedürünü kullanarak bir ağa katılma prosedürü, ZigBee Specification-2012'de 3.6.1.3 numaralı alt paragrafta açıklandığı gibi ağ keşfi ile başlamalıdır [ZigBee Alliance, 2012]. Şekil 4.1'de görüldüğü gibi çocuk cihaz NLME-NETWORK-DISCOVERY.confirm primitifini aldıktan sonra, bir sonraki üst katman ya keşfedilen ağlardan katılmak için bir ağ seçmeli ya da ağ keşfini yeniden yapmalıdır. Bir ağ seçilir seçilmez, cihazın bir yönlendirme cihazı olarak katılmak isteyip istemediğini belirtmek için, RejoinNetwork parametresi 0x00 ve JoinAsRouter parametre seti ile birlikte NLME-JOIN.request primitifi aşağı katmana iletilecektir.



Şekil 4.5. Bir ağa katılmak için çocuk düğümde gerçekleşen süreç (ZigBee Alliance, 2005).

Sadece bir ağa henüz katılmamış olan cihazlar yani ilk defa katılacak cihazlar katılma prosedürünü başlatır. Başka herhangi bir cihaz bu prosedürü başlatırsa, NLME prosedürü sonlandırır ve NLME-

JOIN.confirm primitifini INVALID_REQUEST olarak ayarlanan durum parametresiyle yayınlayarak bir sonraki üst katmana yasa dışı olan isteği bildirir.

Çocuk cihaz henüz bir ağa katılmamış ise NLME-JOIN.request primitifi, NWK katmanında komşu tablosunda ebeveyn bir cihaz arar. Eğer komşu tablosunda bağlanabileceği uygun bir ebeveyn cihaz yoksa NLME-JOIN.confirm primitifi durum parametresi NOT_PERMITTED olacak biçimde NLME katmanı üst katmana yanıt verecektir. Eğer komşu tablosunda uygun bir ebeveyn cihaz bulunursa, NLME katmanı MAC alt katmanına MLME-ASSOCIATE.request primitifi gönderecektir.

MLME-ASSOCIATE.request primitifinin LogicalChannel parametresi, potansiyel ebeveynin koordinatör adresine karşılık gelen komşu tablosu bulunana ayarlanmalıdır. Capability Information parametresinin bit alanları, Tablo 4.11'de gösterilen değerlere sahip olacak ve yetenek bilgileri, nwkCapabilityInformation NIB özniteliğinin değeri olarak depolanacaktır. Birden fazla cihaz bu gereksinimleri karşılırsa, çocuk cihaz en küçük ağ derinliğine sahip olan ebeveyni seçebilir. Burada çok kısa olarak verilen bilgiler detaylı olarak ZigBee Specification-2012'de 3.6.1.4 alt başlığı altında verilmektedir (ZigBee Alliance, 2012).

Tablo 4.11.Yetenek bilgisi bit alanları (ZigBee Alliance, 2012).

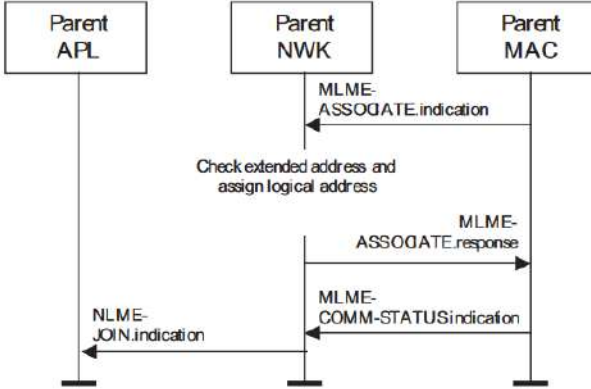
Bit	Adı	Açıklaması
0	Alternatif PAN Koordinatörü	Bu alan ZigBee standartlarındaki her uygulama için 0 değeri seçilecektir.
1	Cihaz Tipi	Katılan cihaz ZigBee yönlendiricisi ise bu alanın değeri 1 olacaktır, katılan cihaz bir ZigBee uç cihaz ya da yönlendirici yetenekli bir cihazsa 0 olacaktır.
2	Güç Kaynağı	0x01 : Şebekeye bağlı Cihaz 0x00 : Diğer güç kaynağı
3	Boşta İken Alıcı	0x01 : Cihaz boştayken alıcı etkinleştirilir. 0x00 : Cihaz boşta iken alıcı devre dışı bırakılır.
4-5	Ayrılmış	Bu alan ZigBee standartlarındaki uygulamalar için 0 değerine ayarlanır.
6	Güvenlik Yeteneği	Bu alan, cihaz Standart Güvenlik Modunda çalışıyorsa 0'a, cihaz Yüksek Güvenlik Modunda çalışıyorsa 1 değerine, sahip olacaktır.
7	Adres Ayırma	Bu alan ağa katılacak cihaza 16-bitlik bir ağ adresi atanması isteniyorsa 1'e, aksine 64-bitlik uzun adresi kullanmak istinirse 0'a ayarlanır.

4.1.4.1.2 Ebeveyn (Koordinatör) Prosedürü

Bir koordinatörün ağna katılmak için; ZigBee koordinatörü veya yönlendirici prosedürü, MAC alt katmanından gelen MLME ASSOCIATE.indication katılma isteği primitifi tarafından başlatılır. Şekil 4.6'da bu durumu ifade eden ebeveyn prosedürü görülmektedir. Sadece bir ZigBee koordinatörü veya bir ZigBee yönlendiricisi olan ve düğümlerin ağa katılmasına izin veren bu FFD düğümler bu

prosedürü başlatmalıdır. Bu prosedür başka bir cihazda başlatılmışsa, NLME bu prosedürü sonlandırır.

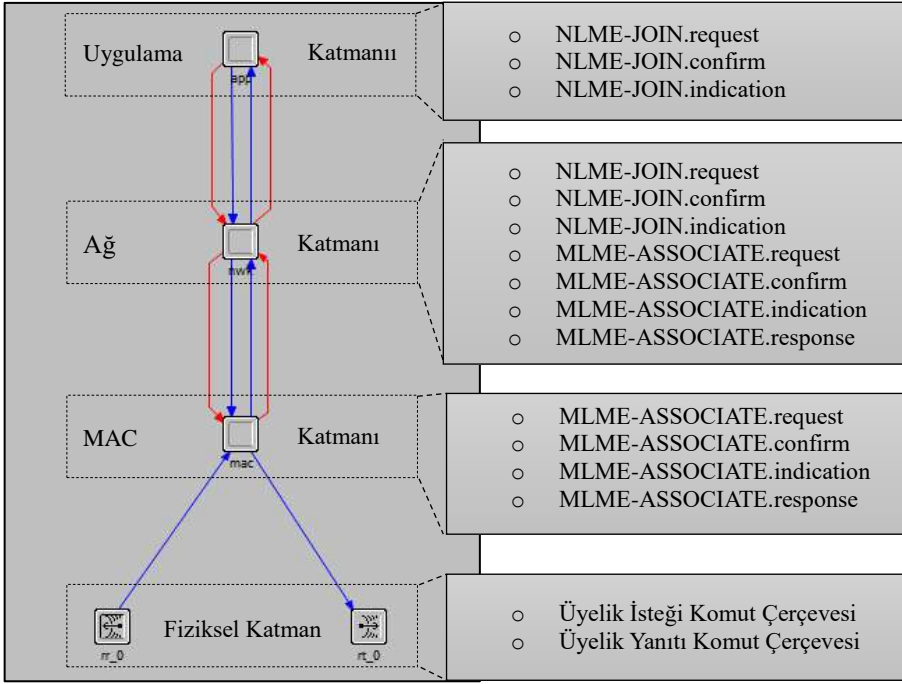
Bu prosedür başlatıldığında, potansiyel bir ebeveyne ait NLME, öncelikle katılmak isteyen cihazın ağında var olup olmadığını belirleyecektir. Bunu yapmak için, NLME, eşleşen 64 bitlik bir genişletilmiş adres bulunup bulunmadığını belirlemek için komşu tablosunda arayacaktır. Genişletilmiş adres eşleşmesi bulunursa, NLME, verilen DeviceCapabilities öğelerinin komşu tablosundaki kaydedilen cihaz türüyle eşleştiğini kontrol edecektir. Cihaz tipi de NLME ile eşleşiyorsa, o zaman karşılık gelen 16-bit ağ adresini elde edecek ve MAC alt-katmanına bir üyelik yanıtı gönderecektir. Cihaz tipi eşleşmesi bulunmazsa, NLME, cihazın tüm kayıtlarını komşu tablosundan çıkarır ve MLME ASSOCIATION.indication işlemini yeniden başlatır. Genişletilmiş adres eşleşmesi bulunamazsa, NLME, mümkünse, yeni cihaz için bir 16 bit ağ adresi tahsis edecektir.



Şekil 4.6.Ebeveyn düğümde gerçekleşen katılma isteği süreci (ZigBee Alliance, 2005).

Potansiyel ebeveyn daha fazla çocuk kabul etme kapasitesine sahip değilse, NLME prosedürü sonlandırır ve bu durumu MAC alt katmanına sonraki MLME-ASSOCIATE.response primitifi ile belirtmelidir. Bu primitifin durum parametresi, PAN'nın kapasitesinin dolduğunu gösterir.

Katılma talebi geldiyse, ebeveynin NLME'si, verilen cihaz bilgilerini kullanarak komşu tablosundaki çocuk için yeni bir kayıt oluşturacak ve MAC alt katmanına sonraki MLME-ASSOCIATE.response primitifinde başarılı bir üyelik belirtilecektir. *NwkSecurityLevel* değeri 0x00 ise, yeni komşu tablosu kaydının ilişki alanı, komşunun bir çocuk olduğunu belirten 0x01 değerine ayarlanmalıdır; aksi takdirde, kimliği doğrulanmamış bir çocuğu belirten 0x05 olarak ayarlanmalıdır. Çocuğa iletilen yanıt durumu, MLME-COMM-STATUS.indication primitifi aracılığıyla ağ katmanına geri iletilir. Eğer iletime başarısız olursa NLME katmanı prosedürü sonlandıracaktır. Eğer iletime başarılı ise NLME katmanı NLME-JOIN.indication primitifini kullanarak ağa bir çocuğun yeni katıldığını sonraki üst katmana (APL) bildirecektir.



Şekil 4.7. Opnet Modeler'de modellenen düğüm yapısı ve modüllerde kullanılan primitifler (Doğan vd., 2018).

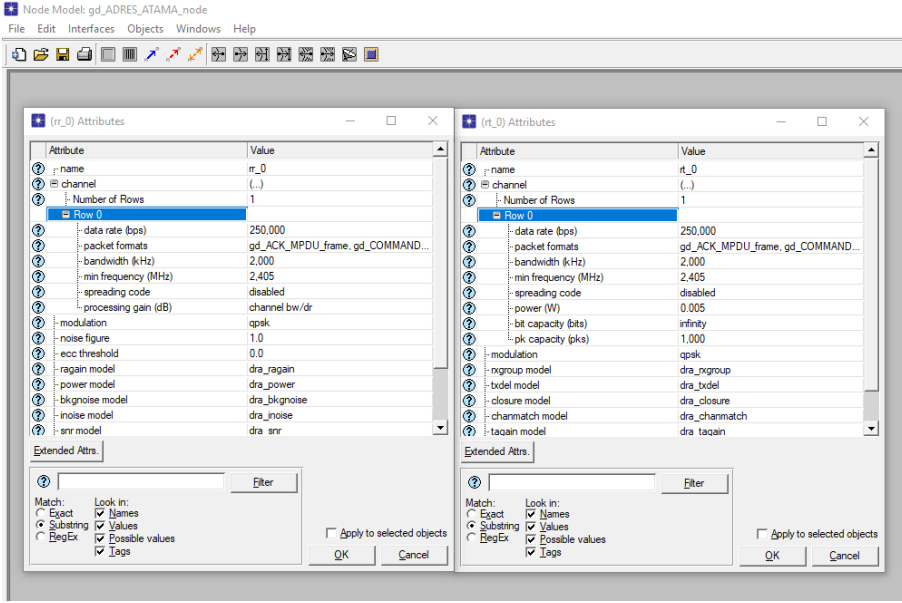
4.2 Modellemede Kullanılan Düğüm Yapısı ve Primitifler

Bu alt bölümde Opnet Modeler'de modellenmesi yapılan düğümlerin yapısına, bu modüllerin birbiriyle olan ilişkilerine ve primitiflere değinilecektir. Düğüm yapısı, bir düğümü oluşturan fiziksel, veri bağı, ağ ve uygulama katmanlarını veya modüllerini ifade ederken primitifler ise bir düğüm yapısında bulunan bu modüllerin birbiriyle haberleşmesini sağlayan hizmet çağrılarınıdır.

Şekil 4.7'de de görüldüğü gibi bir düğümün fiziksel katmanı, MAC katmanı, ağ (NWK) katmanı ve uygulama (APP) katmanlarından her birinin modellenmesi yapılmıştır. Bu şekilde, kırmızı ve mavi okların

birbirine baėlı olduėu kare Őeklinde grnen yapı iŐlemci (processor) veya modl olarak adlandırılırken oklar da akıŐ (stream) olarak adlandırılır. Bu akıŐların modllere baėlandığı yer ise SAP noktası olarak adlandırılır. SAP noktaları temelde iki tiptir; veri SAP noktası ve ynetim SAP noktası. Mavi akıŐlar modllerin veri SAP noktalarına baėlıyken kırmızı akıŐlar ynetim SAP noktalarına baėlıdır. Mavi akıŐlar zerinden veri primitifleri modller arasında gnderilirken kırmızı akıŐlar zerinden de ynetim primitifleri gnderilir.

Alıcı (rr_0) ve gnderici (rt_0) fiziksel katmanın iki modldr. ereveyi kablosuz olarak alıp gndermek gnderici ve alıcı modllere tarafından saėlanır. Herhangi bir paket, mavi akıŐın baėlı olduėu SAP noktasından gnderildiėinde, gnderici modl onu kablosuz ortama modle edilmiŐ sinyal olarak aktarır. Alıcı modl, kablosuz ortamda srekli olarak sinyalleri dinler. Bu sreklilik dėm tasarlayan kullanıcıya baėlı olmakla birlikte bu sinyal dinleme olayı nceden ayarlanılan belirli frekans aralıėı iin geerli olmaktadır.



Şekil 4.8. Alıcı ve gönderici modülünün özellikleri

Şekil 4.8’de alıcı ve gönderici modülünün özelliklerini göstermektedir. İlgili düğümün radyo modülünün özellikleri simülasyon ortamında bu özellik tablosu kanalıyla ayarlanabilir.

Bu çalışmada sadece yönetim primitiflerinden uygulama katmanı için NLME-JOIN.request, NLME-JOIN.confirm, NLME-JOIN.indication primitifleri, ağ katmanı için NLME-JOIN.request, NLME-JOIN.confirm, NLME-JOIN.indication, MLME-ASSOCIATE.request, MLME-ASSOCIATE.confirm, MLME-ASSOCIATE.indication, MLME-ASSOCIATE.response primitifleri, MAC katmanı için MLME-ASSOCIATE.request, MLME-ASSOCIATE.confirm, MLME-ASSOCIATE.indication, MLME-ASSOCIATE.response primitifleri kullanılmıştır. Ayrıca fiziksel katmanda da üyelik isteği komut çerçevesi ve üyelik yanıtı komut çerçevesi kullanılmıştır. Tüm bu

primitifler kendilerine özel fonksiyonlar kullanılarak işlevsellik kazanır. Yani Şekil 4.7'e eğer dikkatli bir şekilde bakılırsa uygulama-ağ ve ağ-mac katmanlarında aynı primitifler kullanılmıştır. Bunun nedeni üst katmanda ayarlanan (set) bir primitif parametrelerinin alt katmanda alınabilmesi (get) ya da tam tersi olarak alt katmanda ayarlanan primitif parametrelerinin üst katmanda alınabilmesi için komşu katmanlar arasında aynı primitiflerin kullanıldığı görülebilir. Bu noktada, dikkat edilmesi gereken başka bir konu da bu primitifler için kullanılan fonksiyonlardır. Bu konu da çok kafa karıştırıcıdır çünkü literatürde primitiflerin nasıl kullanılacağıyla ilgili bir kaynak bulunmamaktadır. Öncelikle bilinmesi gereken ilk şey bir primitif kesinlikle bir fonksiyondan bağımsız olarak düşünülemez. Çünkü bir primitifi işleme koyan ve onun üzerinde bazı işlemler yapabilmek için fonksiyonlar kullanılır. Yani bir fonksiyon içinde primitif kullanılır ve yönetilir. Fonksiyonlar için vurgulanması gereken nokta eğer bir primitif parametrelerini tanımlama (set) üzerinde çalışılmak isteniyorsa farklı primitifler için farklı fonksiyonlar yazılmalıdır. Ama eğer bir primitif için parametre tanımlanılacak ve başka bir primitif için de parametre alınacaksa (get edilecekse) bu iki primitif aynı fonksiyonda kullanılabilir. Yani iki veya daha fazla primitif aynı fonksiyon içinde ayarlanması (set edilmesi) uygun bir kullanım olmaz.

Şekil 4.9'da ağ (NWK) katmanındaki MLME-ASSOCIATE.request primitifinin Opnet Modeler'de kullanımını gösteren bir görüntü verilmiştir. Bu şekilde görüldüğü üzere MLME-ASSOCIATE.request primitifini oluşturmak için parametreleri girilmektedir. Bu

parametrelerden *mar->LogicalChannel = 11;* ve *mar->CoordAddress = 0;* manuel olarak atanmış olmasına rağmen *mar->CoordPANId = njr->ExtendedPANId;* ve *mar->capInf = njr->capInfo;* parametreleri NLME-JOIN.request primitifinin parametrelerinden atanmıştır. Daha sonra bu oluşturulan MLME-ASSOCIATE.request primitifi aşağı katmana gönderilmek üzere kuyruğa eklenmiştir.

```

gd_ADRES_ATAMA_NWK_process.function block
File Edit Options
551
552 /*-----
553 * MLME_ASSOCIATE_request primitive gönderen fonksiyondur.
554 *-----*/
555 static void MLME_Association_request(Packet *pk){
556
557     MLME_ASSOCIATE_request *mar;
558     NLME_JOIN_request *njr;
559     Packet *pkptr;
560
561     njr = (NLME_JOIN_request *)op_prg_mem_alloc(sizeof(NLME_JOIN_request));
562     op_pk_nfd_get(pk,"Primitive Payload",&njr);
563
564     mar = (MLME_ASSOCIATE_request *)op_prg_mem_alloc(sizeof(MLME_ASSOCIATE_request));
565     mar->LogicalChannel = 11;
566     mar->CoordAddrMode = SHORT_ADDRESS;
567     mar->CoordPANId = njr->ExtendedPANId;
568     mar->CoordAddress = 0;
569
570     mar->capInf = (CapabilityInformation *)op_prg_mem_alloc(sizeof(CapabilityInformation));
571     mar->capInf = njr->capInfo;
572
573     mar->SecurityEnable = njr->SecurityEnable;
574
575
576     pkptr = op_pk_create_fmt("gd_primitive_packet");
577     op_pk_nfd_set(pkptr,"Primitive Id",MLME_ASSOCIATE_REQUEST); //primitive id'si gd_tree_adresleme_param.h dosyasındadır.
578     op_pk_nfd_set(pkptr,"Primitive Payload",mar,op_prg_mem_copy_create,op_prg_mem_free,sizeof(MLME_ASSOCIATE_request));
579
580     op_prg_list_insert(MLME_send_pkt_list,pkptr,OPC_LISTPOS_TAIL); //kuyruğa eklendi için daha sonra gönderilecek.
581
582 }
583

```

Şekil 4.9.MLME-ASSOCIATE.request primitifinin kullanımıyla ilgili bir görüntü

Bu primitifin parametrelerine atanan değerler Şekil 4.10'da simülasyon çıktısı olarak da gösterilmektedir. Bu çıktı yazılımsal ortamda bu primitifin parametrelerine atanan değerleri doğruladığı görülmektedir. Bu şekilde yetenek bilgileri (CapabilityInformation) parametresinin tüm alanları ayrıntılı bir biçimde gösterilmektedir.

```

-----
| Module (40), (top.Office Network.node_1.nwk) | Module (67), (top.Office Network.node_2.nwk) |
| From procedure: gd_ADRES_ATAMA_NWK_process [executeNLME enter execs] | From procedure: gd_ADRES_ATAMA_NWK_process [executeNLME enter execs] |
| | |
|-----|-----|
| MLME_Association_request | MLME_Association_request |
|-----|-----|
| LogicalChannel:11 | LogicalChannel:11 |
| CoordAddrMode:2 | CoordAddrMode:2 |
| CoordPANId:0 | CoordPANId:0 |
| CoordAddress:0 | CoordAddress:0 |
| CapInf-a_pan_coord:1 | CapInf-a_pan_coord:1 |
| CapInf-device_type:0 | CapInf-device_type:0 |
| CapInf-power_source:0 | CapInf-power_source:0 |
| CapInf-r_on_wh_idle:0 | CapInf-r_on_wh_idle:0 |
| CapInf-sec_capability:0 | CapInf-sec_capability:0 |
| CapInf-allocate_addr:1 | CapInf-allocate_addr:1 |
| SecurityEnable:0 | SecurityEnable:0 |
|-----|-----|
| | |
|-----|-----|
| Module (705), (top.Office Network.node_3.nwk) | Module (732), (top.Office Network.node_4.nwk) |
| From procedure: gd_ADRES_ATAMA_NWK_process [executeNLME enter execs] | From procedure: gd_ADRES_ATAMA_NWK_process [executeNLME enter execs] |
| | |
|-----|-----|
| MLME_Association_request | MLME_Association_request |
|-----|-----|
| LogicalChannel:11 | LogicalChannel:11 |
| CoordAddrMode:2 | CoordAddrMode:2 |
| CoordPANId:0 | CoordPANId:0 |
| CoordAddress:0 | CoordAddress:0 |
| CapInf-a_pan_coord:1 | CapInf-a_pan_coord:1 |
| CapInf-device_type:0 | CapInf-device_type:0 |
| CapInf-power_source:0 | CapInf-power_source:0 |
| CapInf-r_on_wh_idle:0 | CapInf-r_on_wh_idle:0 |
| CapInf-sec_capability:0 | CapInf-sec_capability:0 |
| CapInf-allocate_addr:1 | CapInf-allocate_addr:1 |
| SecurityEnable:0 | SecurityEnable:0 |
|-----|-----|

```

Şekil 4.10. Ağ katmanındaki MLME-ASSOCIATE.request primitifinin simülasyon çıktısı

MLME-ASSOCIATE.request primitifi ve diğer tüm primitiflerin modellenmesi OPNET Modellerde bir veri yapısı olarak gerçekleştirilebilir. MLME-ASSOCIATE.request primitifinin bir veri yapısı olarak modellenmesi Şekil 4.11'deki gibidir. Bu primitif gibi diğer tüm primitifler, tek bir başlık dosyasında (adresAtama.h) kaydedilmiştir.

```

/*-----
 * NWK katmanında end device'in coordinator'a bağlanması için MAC'a gönderilen
 * primitive'dir.
 * IEEE 802.15.4'e göre oluşturulmuştur.
 * FULL.
 *-----*/
typedef struct{
    int LogicalChannel;
    int CoordAddrMode;
    int CoordPANId;
    long long CoordAddress; //64 bitlik
    CapabilityInformation *capInf;
    Boolean SecurityEnable;
}MLME_ASSOCIATE_request;

```

Şekil 4.11. Primitif kütüphanesinde bulunan MLME-ASSOCIATE.request primitifi

Bazı primitiflerle birlikte veya haricinde kullanılan sabitler ve bilgi tabanı da (Information Base) bulunmaktadır. Sabitler C programlama dilindeki makrolar gibi tanımlanır ve sabitler ihtiyaç duyulan fonksiyonlarda kullanılır ve her zaman için kendisine atanan değeri korurlar. Bilgi tabanı ise modüllerin ana fonksiyonunda ilk değerleri atanan ve daha sonra modüldeki bazı fonksiyonlarda bu ilk değerlerin son değerleriyle değiştirildiği bir nevi programlama dillerindeki değişkenler olarak tanımlanabilir. Bilgi tabanıyla sabitler arasındaki en önemli fark sabitler ilk değerlerini aldıklarında o değerler hiçbir değişikliğe uğramadan kullanılabilir ama bilgi tabanındaki parametreler en son hangi değeri almışlarsa tüm fonksiyonlarda o değer geçerli olur. Bu çalışmada kullanılan bazı sabitlerin tanımlanma görüntüleri Şekil 4.12’de verilmiştir. Bu şekilde de görüldüğü gibi çerçeveler için çerçeve tipi alt alanında alınacak değerler atanmıştır.

```
16
17
18 /*-----
19  * Frame type subfield tanımlamaları -IEEE 802.15.4'e göre
20  *-----
21 #define BEACON_FRAME_TYPE    0
22 #define DATA_FRAME_TYPE    1
23 #define ACK_FRAME_TYPE      2
24 #define COMMAND_FRAME_TYPE  3
```

Şekil 4.12.Bazı sabitlerin görüntüsü

```

/*-----
 * NWK Information Base
 * Özel amaçlı oluşturulmuştur.
 *-----
typedef struct{
  Objid      obj_id; //mevcut modülün id'si
  Objid      parent_id; //dugumun yani cihazın id'si.
  int        nwk_dst_addr;
  int        nwk_src_addr;

  int        device_type;
}NWK_Attributes;

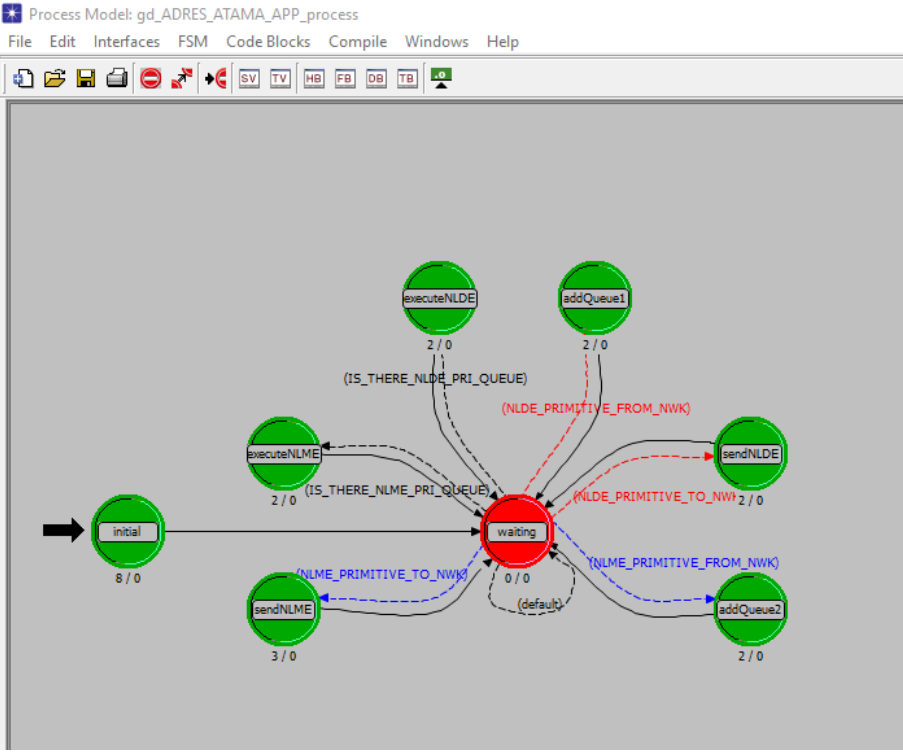
```

Şekil 4.13. Ağ katmanında kullanılan bilgi tabanındaki parametreler

Yine bu çalışmada kullanılan ve özel olarak oluşturulan (yani herhangi bir standartta olmayan) bilgi tabanındaki parametrelerden bazıları Şekil 4.13'te gösterilmektedir. Bu parametrelerden *device_type* ilk değer olarak uç cihazı belirten “3” değeri atanmıştır ama daha sonra kullanıcı bu değeri koordinatör olarak ayarlarsa koordinatörü belirten “1” değeri atanacak ve o düğüm için artık bu değer “1” olarak kullanılacaktır.

4.3 Modellemede Kullanılan Süreç Yapısı

Bu alt bölümde uygulama katmanı, ağ katmanı ve MAC katmanı için Opnet Modeler'de modellenen süreç ya da sonlu durum makinesi olarak adlandırılan yapı üzerinde durulacaktır. Bu süreç yapıları, ilgili katmanda kuyrukların, primitiflerin, olayların davranışlarının kontrol edilmesinde veya yönetilmesinde çok önemlidir.



Şekil 4.14. Uygulama katmanının süreç yapısı (Doğan vd., 2018).

Şekil 4.14'te uygulama katmanı için oluşturulan süreçte düğüm ilk çalışmaya başladığında başlangıç durumundaki fonksiyonlarla düğümün bazı varsayılan özelliklerinin yüklendiği durum icra edilir. Bu durum koşulsuz bir geçişe sahiptir yani fonksiyonların icrası bittikten sonra kırmızı renk (unforced state) ile görülen bekleme durumuna geçer. Bu durum da ise şekilde görüldüğü gibi ağ (NWK) katmanından bir NLDE ya da NLME primitifinin gelmesi gibi bir koşulun gerçekleşmesi olayı beklenir. Eğer bu koşullardan biri gerçekleşirse gelen primitif kuyruğa alınır ve bu kuyrukta belirli bir zamanlayıcı (timer) süresi kadar bekletilir. Zamanlayıcının dolmasını

takriben ilgili primitif için ilgili yürütme durumuna geçilme koşulu sağlanılır ve o duruma geçilir. Ardından ilgili primitif işlemleri icra edilir daha sonra işlemler bittiğinde ilgili primitif gönderilecekler kuyruğuna alınır. Bu noktada da belli bir zamanlayıcı süresinin dolması beklenir. Bu süre dolduğunda ise ilgili gönder duruma (state) geçilerek primitif NWK katmanına gönderilir.

```

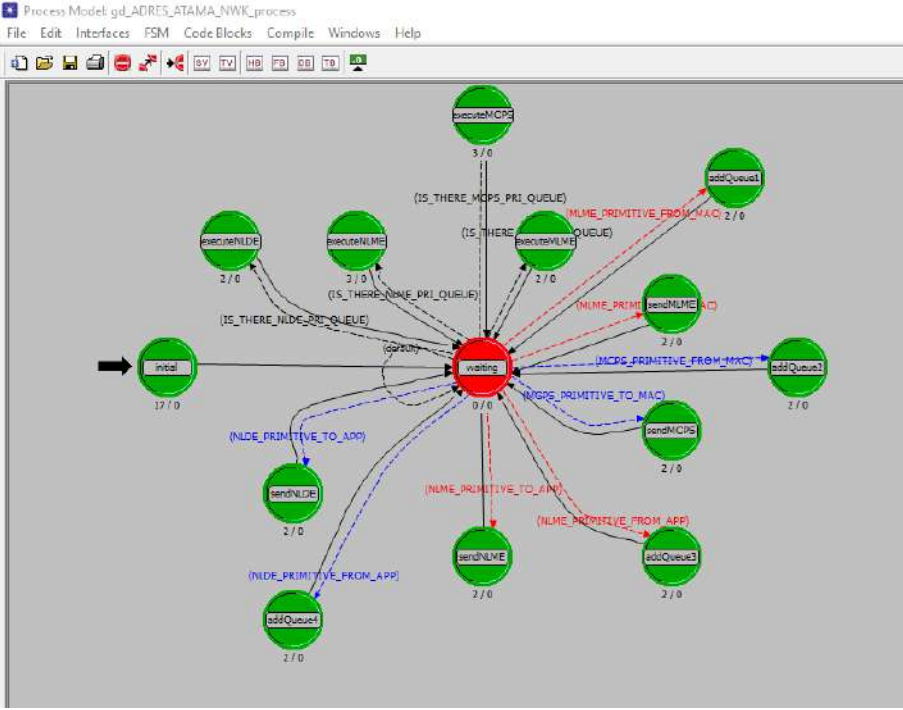
gd_ADRES_ATAMA_APP_process.function block
File Edit Options
130  /*-----
131  /* receive_NLME_from_NWK_pk_list kuyrugunda hiç MLME primitive paketi var mi
132  /* varsa gerekli primitive fonksiyonunu cagiran fonksiyondur.
133  /*-----
134  void execute_NLME_on_APP(){
135      int i=0;
136      int num_pkts;
137      int prim_id;
138      Packet *pkptr = OPC_NIL;
139      num_pkts = op_prg_list_size(receive_NLME_from_NWK_pk_list);
140
141
142
143      if(num_pkts > 0){
144          pkptr = op_prg_list_access(receive_NLME_from_NWK_pk_list,OPC_LISTPOS,
145          op_pk_nfd_access(pkptr,"Primitive Id",&prim_id);
146
147          switch(prim_id){
148              case NLME_JOIN_REQUEST :sprintf(msg,"Primitie ID: %d ",prim_id);
149              case NLME_JOIN_INDICATION :NLME_JOIN_indication_on_APP(pkptr);op_
150              case NLME_JOIN_CONFIRM :NLME_JOIN_confirm_on_APP(pkptr);op_prg_l
151
152
153
154              default : sprintf(msg,"Primitie ID: %d Taninmadi.!!!",prim_id);o
155
156          }
157
158      }
159
160
161      execute_NLME_pk_timer = op_intrpt_schedule_self(op_sim_time()+NLME_EXE_P
162

```

Şekil 4.15. *executeNLME* durumuna ait örnek bir kod bloğu (Doğan vd., 2018).

Şekil 4.14’te görülen *executeNLME* durumunun çalışması halinde icra edilecek kod bloklarından biri Şekil 4.15’te gösterilmiştir. Bu şekildeki fonksiyonda ilk olarak ağ katmanından alınan yönetim primitiflerinin kuyruğuna bakılır. Eğer bu kuyrukta bir primitif varsa o primitifin id’sine erişilerek önceden tanımlanılan bir primitif olup olmadığına bakılır eğer önceden tanımlanmış ise ilgili fonksiyonu da

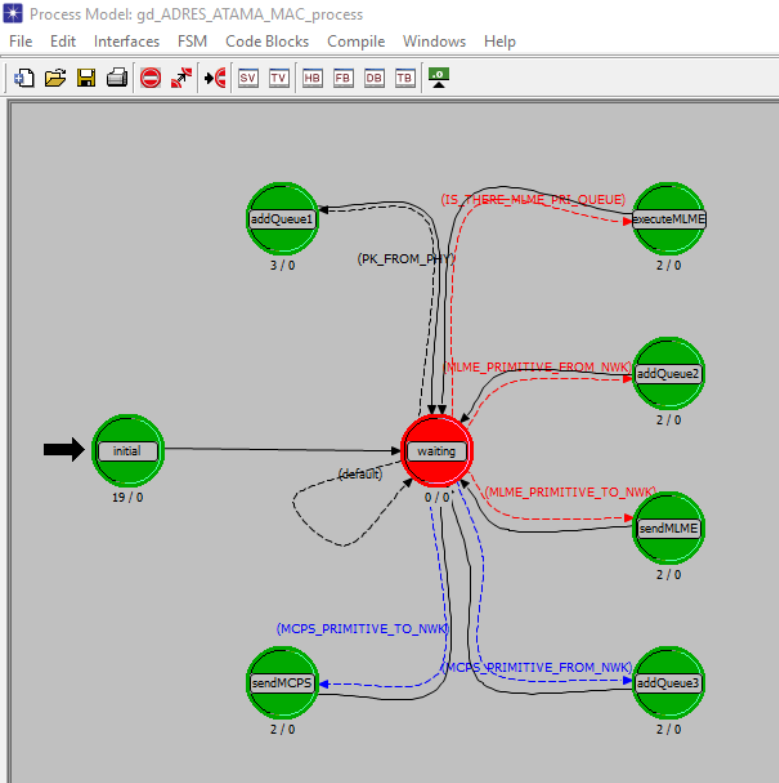
tanımlanmıştır ve o fonksiyon çağrılarak bu primitif de o fonksiyona parametre olarak gönderilir. Eğer kuyrukta yoksa, zamanlayıcı (timer) yeniden aktif edilir ve bu fonksiyondan çıkılır.



Şekil 4.16. Ağ katmanının süreç yapısı (Doğan vd., 2018).

Şekil 4.16’de ağ katmanı için oluşturulan süreçte düğüm ilk çalışmaya başladığında başlangıç durumundaki fonksiyonlarla düğümün bazı varsayılan özelliklerinin yüklendiği durum icra edilir. Bu durum koşulsuz bir geçişe sahiptir yani fonksiyonların icrası bittikten sonra kırmızı renk (unforced state) ile görülen bekleme durumuna geçer. Bu durum da ise şekilde görüldüğü gibi ya uygulama katmanından bir NLDE veya NLME primitifinden herhangi birinin gelmesi ya da MAC katmanından bir MLME veya MCPS

primitifinden herhangi birinin gelmesi gibi bir koşulun gerçekleşmesi olayı beklenir. Eğer bu koşullardan biri gerçekleşirse gelen primitif kuyruğa alınır ve bu kuyrukta belirli bir zamanlayıcı (timer) süresi kadar bekletilir. Zamanlayıcının dolmasını takriben ilgili primitif için ilgili yürütme durumuna geçilme koşulu sağlanılır ve o duruma geçilir ve ilgili primitif işlemleri icra edilir daha sonra işlemler bittiğinde ilgili primitifin gönderilecekler kuyruğuna alınır burada da belli bir zamanlayıcı süresinin dolması beklenir. Bu süre dolduğunda ise ilgili gönder duruma geçilerek primitif ilgili uygulama ya da MAC katmanına gönderilir.

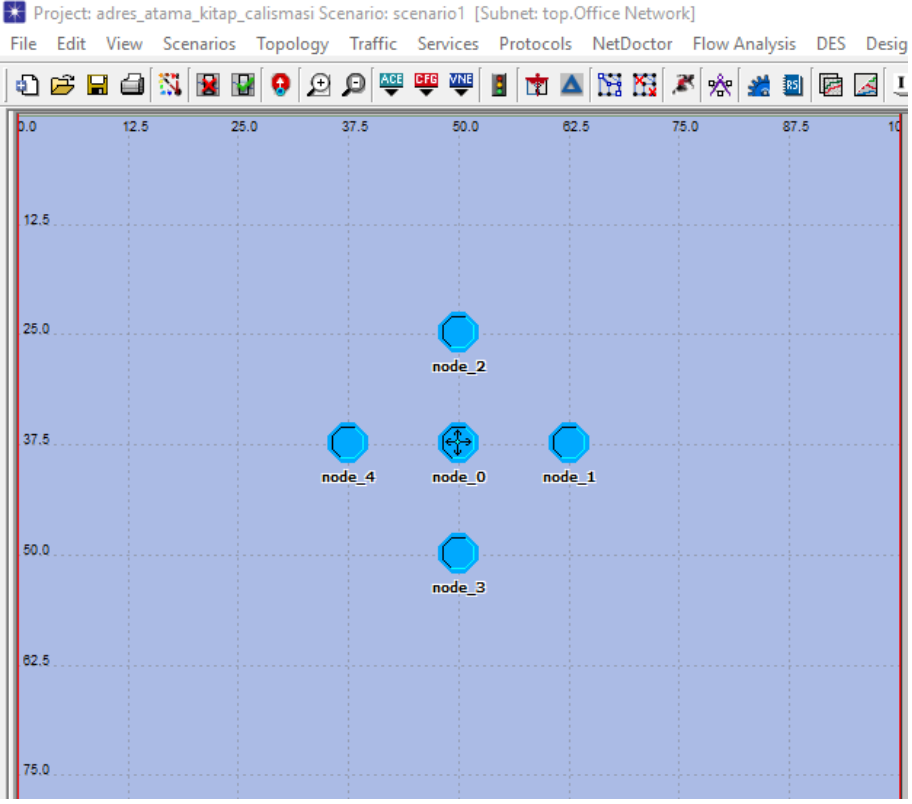


Şekil 4.17.MAC katmanının süreç yapısı (Doğan vd., 2018).

Şekil 4.17’de MAC katmanı için modellenen süreç yapısında, düğüm ilk çalışmaya başladığında başlangıç durumundaki fonksiyonlarla düğümün bazı varsayılan özelliklerinin yüklendiği durum icra edilir. Bu durum koşulsuz bir geçişe sahiptir yani fonksiyonların icrası bittikten sonra kırmızı renk ile görülen bekleme durumuna geçer. Bu durum da ise şekilde görüldüğü gibi ya ağ katmanından bir MLME veya MCPS primitifinden herhangi birinin gelmesi ya da fiziksel katmanından bir paketin gelmesi gibi bir koşulun gerçekleşmesi olayı beklenir. Eğer bu primitif koşullardan biri gerçekleşirse gelen primitif kuyruğa alınır ve bu kuyrukta belirli bir zamanlayıcı (timer) süresi kadar bekletilir. Zamanlayıcının dolmasını takriben ilgili primitif için ilgili yürütme durumuna geçilme koşulu sağlanılır ve o duruma geçilir ve ilgili primitif işlemleri icra edilir daha sonra işlemler bittiğinde ilgili primitifin gönderilecekler kuyruğuna alınır burada da belli bir zamanlayıcı süresinin dolması beklenir. Bu süre dolduğunda ise ilgili gönder durumuna geçilerek primitif ağ katmanına gönderilir. Eğer MAC katmanında bir paket oluşturulmuş ise o paketi fiziksel katmana gönderir ve yayınlanır.

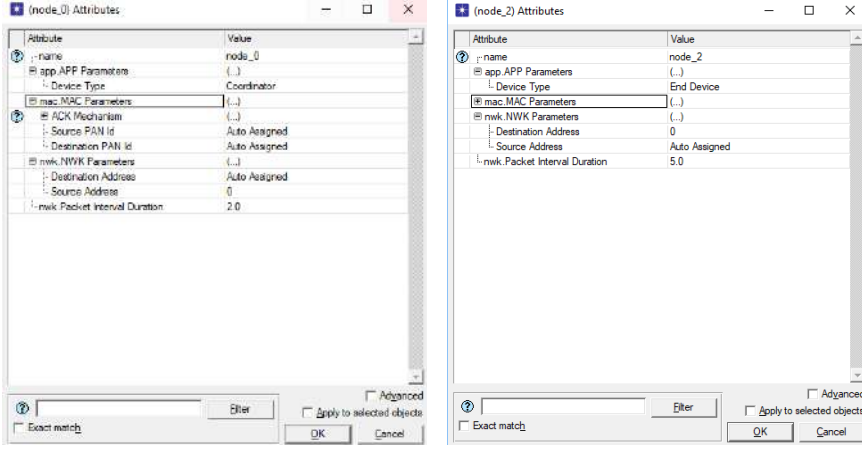
5. ZIGBEE AĞININ SİMÜLASYONU VE ANALİZİ

Bu çalışma için oluşturulan simülasyonda iki tip düğüm vardır; koordinatör ve uç cihaz. Koordinatör cihazının kaynak ağ adresi varsayılan olarak 0 atanmıştır ve bu iki cihazın aynı PAN'da olduğu varsayılarak simülasyon icra edilmiştir. Bu simülasyonda ayrıca düğümlerin paket alıp göndermesi için sinyal çakışmasını da engellemek için aralıklı paket süresi kullanılmıştır. Aralıklı paket süresi her düğüm için asal bir sayıyı temsil eder. Aralıklı paket süresinde asal sayı kullanılmasının nedeni düğümlerin aynı anda paket göndermesini engellemektir. Eğer iki veya daha fazla düğüm aynı anda paket gönderirse bu paketlerin radyo dalgaları bozulmaya uğrayacak ve hiçbir düğüm bu paketleri alamayacaktır. Bu simülasyon ortamının alanı 100 m^2 'dir ve koordinatör düğümü ortada olmak üzere diğer uç düğümlerin koordinatöre olan uzaklığı 12.5 metredir. Ayrıca, düğümler arasındaki iletişim, hava ortamında bir düğümden başka bir düğüme tek hop olacak şekilde paket akışı sağlanmaktadır. Simülasyon 12 saniye boyunca icra edilmiştir.



Şekil 5.1. Simülasyon ortamından bir görüntü

Şekil 5.1’de simülasyon ortamından bir görüntü verilmiştir. Burada görülen node_0 adlı düğüm koordinatörü, node_1, node_2, node_3 ve node_4 adlı düğümler ise uç cihazları temsil etmektedir.



Şekil 5.2.Simülasyonda kullanılan bazı düğümlerin öznelikleri

Bu simülasyonda kullanılan bazı düğümlerin öznelikleri Şekil 5.2’de gösterilmektedir. Buna göre *node_0* düğümü için cihaz tipi koordinatör seçilmiştir, MAC parametreleri altında bulunan ACK mekanizması kullanılmamıştır, kaynak ve hedef PAN tanımlayıcı otomatik atanmıştır yani kod kısmında tanımlanmıştır, ağ (NWK) parametrelerinin altında bulunan kaynak adres sıfır olarak ayarlanmıştır ve hedef adres otomatik atanmıştır. Yine bu düğüm için paketler arası süre 2 saniye olarak ayarlanmıştır. Bu düğümüne benzer olarak *node_2* düğümünde de MAC parametreleri altında bulunan özellikler aynı olmakla birlikte cihaz tipi uç cihaz, ağ parametrelerinden hedef adres sıfır, kaynak adres koordinatör tarafından atanacağı için otomatik atama ve paketler arası süre 5 saniye olarak ayarlanmıştır. Yine bu düğümle benzer ama bu şekilde verilmeyen *node_1*, *node_3*, ve *node_4* düğümü de aynı özneliklere

sahiptir sadece paketler arası süre aralarında asal olacak şekilde değiştirilmiştir.

Bu simülasyonda modellemesi yapılan tüm düğümlere koordinatör hariç olmak üzere rastgele olarak MAC adresleri atanmıştır. Bu çalışmada, ZigBee protokolünde yer alan ağ keşfi süreci gerçekleştirilmediği için koordinatör düğüme varsayılan olarak 00:00:00:00:00:00 MAC adresi tanımlanmıştır ve uç düğümlere de koordinatör düğümün MAC adresi tanımlanmıştır.

```
-----  
| Module (35), (top.Office Network.node_1.mac) | | Module (62), (top.Office Network.node_2.mac) | |
| From procedure: qd_ADRES_ATAMA_MAC_process [initial enter execs] | | From procedure: qd_ADRES_ATAMA_MAC_process [initial enter execs] |  
| | | | |  
| Başlangıç MAC Adresi: | | Başlangıç MAC Adresi: |  
| MAC Adresi:00:00:00:29:48:23 | | MAC Adresi:00:00:08:BE:67:84 |  
-----  
  
-----  
| Module (700), (top.Office Network.node_3.mac) | | Module (727), (top.Office Network.node_4.mac) | |
| From procedure: qd_ADRES_ATAMA_MAC_process [initial enter execs] | | From procedure: qd_ADRES_ATAMA_MAC_process [initial enter execs] |  
| | | | |  
| Başlangıç MAC Adresi: | | Başlangıç MAC Adresi: |  
| MAC Adresi:00:00:0A:E1:3D:6C | | MAC Adresi:00:00:0C:D6:72:AE |  
-----
```

Şekil 5.3. Simülasyon çıktısında MAC adreslerinin atanması

Şekil 5.3’de uç düğümlere MAC adreslerinin rastgele atandığı simülasyon sonuçları görülmektedir. Bu şekilde, *node_1*’ın MAC adresi 00:00:00:29:48:23, *node_2*’in 00:00:08:BE:67:84, *node_3*’nin 00:00:0A:E1:3D:6C, ve *node_4*’ün 00:00:0C:D6:72:AE, olarak atandığı görülmektedir. Bu sürecin ardından bir uç düğüm ağa katılmak için uygulama katmanında NLME-JOIN.request primitifi komşu tablosunda bir koordinatör düğüm arar. Bu çalışmada, ağ keşfinin yapıldığı varsayıldığı için uç düğümlerde koordinatörün MAC adresi tanımlanmıştı ve bu yüzden bu primitif uygun bir koordinatörün olduğunu bilir ve ağ katmanına MLME-ASSOCIATE.request primitifinin hazırlanmasını emreder.

```

-----
| Module (49), (top.Office Network.node_1.app) | Module (76), (top.Office Network.node_2.app) |
| From procedure: gd_ADRES_ATAMA_APP_process [initial enter execs] | From procedure: gd_ADRES_ATAMA_APP_process [initial enter execs] |
| | |
| ----- | ----- |
| NLME_JOIN_request | NLME_JOIN_request |
| ----- | ----- |
| ExtendedPANId:0 | ExtendedPANId:0 |
| RejoinNetwork:0 | RejoinNetwork:0 |
| ScanChannels:0 | ScanChannels:0 |
| ScanDuration:0 | ScanDuration:0 |
| CapInf-a_pan_coord:1 | CapInf-a_pan_coord:1 |
| CapInf-device_type:0 | CapInf-device_type:0 |
| CapInf-power_source:0 | CapInf-power_source:0 |
| CapInf-r_on_wh_idle:0 | CapInf-r_on_wh_idle:0 |
| CapInf-sec_capability:0 | CapInf-sec_capability:0 |
| CapInf-allocate_addr:1 | CapInf-allocate_addr:1 |
| SecurityEnable:0 | SecurityEnable:0 |
| ----- | ----- |
| | |
| ----- | ----- |
| Module (714), (top.Office Network.node_3.app) | Module (741), (top.Office Network.node_4.app) |
| From procedure: gd_ADRES_ATAMA_APP_process [initial enter execs] | From procedure: gd_ADRES_ATAMA_APP_process [initial enter execs] |
| | |
| ----- | ----- |
| NLME_JOIN_request | NLME_JOIN_request |
| ----- | ----- |
| ExtendedPANId:0 | ExtendedPANId:0 |
| RejoinNetwork:0 | RejoinNetwork:0 |
| ScanChannels:0 | ScanChannels:0 |
| ScanDuration:0 | ScanDuration:0 |
| CapInf-a_pan_coord:1 | CapInf-a_pan_coord:1 |
| CapInf-device_type:0 | CapInf-device_type:0 |
| CapInf-power_source:0 | CapInf-power_source:0 |
| CapInf-r_on_wh_idle:0 | CapInf-r_on_wh_idle:0 |
| CapInf-sec_capability:0 | CapInf-sec_capability:0 |
| CapInf-allocate_addr:1 | CapInf-allocate_addr:1 |
| SecurityEnable:0 | SecurityEnable:0 |
| ----- | ----- |

```

Şekil 5.4. Simülasyon çıktısında NLME-JOIN.request primitifinin görüntüsü

Şekil 5.4'te görüldüğü gibi *node_1*, *node_2*, *node_3* ve *node_4* uç düğümlerinin uygulama katmanında oluşturulan NLME-JOIN.request primitifiyle aşağı katmana yani ağ katmanına bu primitifteki parametreleri gerekli yerlerde kullanarak bir MLME-ASSOCIATE.request primitifinin oluşturulmasını emreder ve MLME-ASSOCIATE.request primitifi görevi devraldığında artık NLME-JOIN.request primitifinin görevi sonlanır.

```

-----
Module (40), (top.Office Network.node_1.nwk)
From procedure: gd_ADRES_ATAMA_NWK_process [executeNLME enter execs]
-----
MLME_Association_request
-----
LogicalChannel:11
CoordAddrMode:2
CoordPANId:0
CoordAddress:0
CapInf-a_pan_coord:1
CapInf-device_type:0
CapInf-power_source:0
CapInf-r_on_wh_idle:0
CapInf-sec_capability:0
CapInf-allocate_addr:1
SecurityEnable:0
-----

Module (705), (top.Office Network.node_3.nwk)
From procedure: gd_ADRES_ATAMA_NWK_process [executeNLME enter execs]
-----
MLME_Association_request
-----
LogicalChannel:11
CoordAddrMode:2
CoordPANId:0
CoordAddress:0
CapInf-a_pan_coord:1
CapInf-device_type:0
CapInf-power_source:0
CapInf-r_on_wh_idle:0
CapInf-sec_capability:0
CapInf-allocate_addr:1
SecurityEnable:0
-----

Module (67), (top.Office Network.node_2.nwk)
From procedure: gd_ADRES_ATAMA_NWK_process [executeNLME enter execs]
-----
MLME_Association_request
-----
LogicalChannel:11
CoordAddrMode:2
CoordPANId:0
CoordAddress:0
CapInf-a_pan_coord:1
CapInf-device_type:0
CapInf-power_source:0
CapInf-r_on_wh_idle:0
CapInf-sec_capability:0
CapInf-allocate_addr:1
SecurityEnable:0
-----

Module (732), (top.Office Network.node_4.nwk)
From procedure: gd_ADRES_ATAMA_NWK_process [executeNLME enter execs]
-----
MLME_Association_request
-----
LogicalChannel:11
CoordAddrMode:2
CoordPANId:0
CoordAddress:0
CapInf-a_pan_coord:1
CapInf-device_type:0
CapInf-power_source:0
CapInf-r_on_wh_idle:0
CapInf-sec_capability:0
CapInf-allocate_addr:1
SecurityEnable:0
-----

```

Şekil 5.5. Simülasyon çıktısında MLME-ASSOCIATE.request primitifinin görüntüsü

```

-----
Module (35), (top.Office Network.node_1.mac)
From procedure: gd_ADRES_ATAMA_MAC_process [executeNLME enter execs]
-----
MLME_Association_request
-----
LogicalChannel:11
CoordAddrMode:2
CoordPANId:0
CoordAddress:0
CapInf-a_pan_coord:1
CapInf-device_type:0
CapInf-power_source:0
CapInf-r_on_wh_idle:0
CapInf-sec_capability:0
CapInf-allocate_addr:1
SecurityEnable:0
-----

Module (62), (top.Office_ATAMA.node_2.mac)
From procedure: gd_ADRES_ATAMA_MAC_process [executeNLME enter execs]
-----
MLME_Association_request
-----
LogicalChannel:11
CoordAddrMode:2
CoordPANId:0
CoordAddress:0
CapInf-a_pan_coord:1
CapInf-device_type:0
CapInf-power_source:0
CapInf-r_on_wh_idle:0
CapInf-sec_capability:0
CapInf-allocate_addr:1
SecurityEnable:0
-----

Module (700), (top.Office Network.node_3.mac)
From procedure: gd_ADRES_ATAMA_MAC_process [executeNLME enter execs]
-----
MLME_Association_request
-----
LogicalChannel:11
CoordAddrMode:2
CoordPANId:0
CoordAddress:0
CapInf-a_pan_coord:1
CapInf-device_type:0
CapInf-power_source:0
CapInf-r_on_wh_idle:0
CapInf-sec_capability:0
CapInf-allocate_addr:1
SecurityEnable:0
-----

Module (727), (top.Office Network.node_4.mac)
From procedure: gd_ADRES_ATAMA_MAC_process [executeNLME enter execs]
-----
MLME_Association_request
-----
LogicalChannel:11
CoordAddrMode:2
CoordPANId:0
CoordAddress:0
CapInf-a_pan_coord:1
CapInf-device_type:0
CapInf-power_source:0
CapInf-r_on_wh_idle:0
CapInf-sec_capability:0
CapInf-allocate_addr:1
SecurityEnable:0
-----

```

Şekil 5.6. Mac katmanındaki MLME-ASSOCIATE.request primitifi

Şekil 5.5'te de görüldüğü gibi MLME-ASSOCIATE.request primitifi bazı parametreleri kedisinin bulunduğu ağ katmanından geriye kalan parametreleri ise uygulama katmanından gelen NLME-JOIN.request

primitifinden almıştır. MLME-ASSOCIATE.request primitifi de aynı şekilde MAC katmanına gönderilir.

Şekil 5.6’da görüldüğü gibi MAC katmanına ulaşan bu primitif artık kendisinde olan gerekli parametreleri de kullanarak *node_0*’a yani koordinatöre üyelik isteği komut çerçevesini hazırlar ve fiziksel katmanın özelliklerini de kullanarak bu çerçeveyi iletir.

```

-----
Module (8), (top.Office Network.node_0.mac)
From procedure: gd_ADRES_ATAMA_MAC_process [addQueue! enter execs]
-----
|
|
|-----
| Alınan komut çerçevesi:
|-----
| Security Enabled:0
| Frame Pending:0
| Ack Request:0
| Dest Addressing Mode:2
| Source Addressing Mode:3
| Sequence Number:51
| Intra-PAN:0
| Destination PAN Identifier:0
| Source PAN Identifier:65535
| Destination Address:00:00:00:00:00:00
| Source Address:00:00:00:29:48:23
| MSDU'daki Command Frame Identifier:1
|-----
-----

Module (8), (top.Office Network.node_0.mac)
From procedure: gd_ADRES_ATAMA_MAC_process [addQueue! enter execs]
-----
|
|
|-----
| Alınan komut çerçevesi:
|-----
| Security Enabled:0
| Frame Pending:0
| Ack Request:0
| Dest Addressing Mode:2
| Source Addressing Mode:3
| Sequence Number:43
| Intra-PAN:0
| Destination PAN Identifier:0
| Source PAN Identifier:65535
| Destination Address:00:00:00:00:00:00
| Source Address:00:00:08:BE:67:84
| MSDU'daki Command Frame Identifier:1
|-----
-----

-----
Module (8), (top.Office Network.node_0.mac)
From procedure: gd_ADRES_ATAMA_MAC_process [addQueue! enter execs]
-----
|
|
|-----
| Alınan komut çerçevesi:
|-----
| Security Enabled:0
| Frame Pending:0
| Ack Request:0
| Dest Addressing Mode:2
| Source Addressing Mode:3
| Sequence Number:48
| Intra-PAN:0
| Destination PAN Identifier:0
| Source PAN Identifier:65535
| Destination Address:00:00:00:00:00:00
| Source Address:00:00:08:BE:67:84
| MSDU'daki Command Frame Identifier:1
|-----
-----

Module (8), (top.Office Network.node_0.mac)
From procedure: gd_ADRES_ATAMA_MAC_process [addQueue! enter execs]
-----
|
|
|-----
| Alınan komut çerçevesi:
|-----
| Security Enabled:0
| Frame Pending:0
| Ack Request:0
| Dest Addressing Mode:2
| Source Addressing Mode:3
| Sequence Number:254
| Intra-PAN:0
| Destination PAN Identifier:0
| Source PAN Identifier:65535
| Destination Address:00:00:00:00:00:00
| Source Address:00:00:0C:D6:72:AE
| MSDU'daki Command Frame Identifier:1
|-----
-----

```

Şekil 5.7.Koordinatöre gelen üyelik isteği çerçevesinin görüntüsü

Şekil 5.7’de koordinatöre kaynak adresi *00:00:00:29:48:23*, *00:00:08:BE:67:84*, *00:00:0A:E1:3D:6C* ve *00:00:0C:D6:72:AE* olan uç düğümlerden yani sırasıyla, *node_1*, *node_2*, *node_3* ve *node_4*’ten gelen üyelik isteği komut çerçevesinin bazı alanları gösterilmiştir. Burada önemli olan noktalar, kaynak adresleme modu “3” yani uzun adresleme modunu tüm uç düğümler kullanılırken, *node_0* yani koordinatörün ağ adresi önceden belli olduğu için kısa adres modu olan “2” kullanılmıştır. Ayrıca bu simülasyonda sadece bir

tane PAN kullanıldığından yani PAN-içi haberleşme gerçekleştiğinden hedef PAN tanımlayıcı ve kaynak PAN tanımlayıcı alanının kontrolü yapılmamıştır zaten böyle bir kontrol yapılırsa mutlaka yönlendirme tablolarının tutulması gerekir. Tüm bunların yanı sıra çerçevenin yük alanında komut çerçevesi tanımlayıcısı “/”dir. Yani üyelik isteği komutunun geldiğini ve ona göre gerekli primitifi oluşturulması emri verilir.

<pre>----- Module (8), (top.Office Network.node_0.mac) From procedure: gd_ADRES_ATAMA_MAC_process [addQueueI enter execs] ----- MLME_Association_indication ----- DeviceAddress:00:00:00:29:48:23 CapInf-a_pan_coord:1 CapInf-device_type:0 CapInf-power_source:0 CapInf-r_on_wh_idle:0 CapInf-sec_capability:0 CapInf-allocate_addr:1 SecurityUse:0 ACLEntry:8 -----</pre>	<pre>----- Module (8), (top.Office Network.node_0.mac) From procedure: gd_ADRES_ATAMA_MAC_process [addQueueI enter execs] ----- MLME_Association_indication ----- DeviceAddress:00:00:08:BE:67:84 CapInf-a_pan_coord:1 CapInf-device_type:0 CapInf-power_source:0 CapInf-r_on_wh_idle:0 CapInf-sec_capability:0 CapInf-allocate_addr:1 SecurityUse:0 ACLEntry:8 -----</pre>
<pre>----- Module (8), (top.Office Network.node_0.mac) From procedure: gd_ADRES_ATAMA_MAC_process [addQueueI enter execs] ----- MLME_Association_indication ----- DeviceAddress:00:00:0A:E1:3D:6C CapInf-a_pan_coord:1 CapInf-device_type:0 CapInf-power_source:0 CapInf-r_on_wh_idle:0 CapInf-sec_capability:0 CapInf-allocate_addr:1 SecurityUse:0 ACLEntry:8 -----</pre>	<pre>----- Module (8), (top.Office Network.node_0.mac) From procedure: gd_ADRES_ATAMA_MAC_process [addQueueI enter execs] ----- MLME_Association_indication ----- DeviceAddress:00:00:0C:D6:72:AE CapInf-a_pan_coord:1 CapInf-device_type:0 CapInf-power_source:0 CapInf-r_on_wh_idle:0 CapInf-sec_capability:0 CapInf-allocate_addr:1 SecurityUse:0 ACLEntry:8 -----</pre>

Şekil 5.8.MAC katmanındaki MLME-ASSOCIATE.indication primitifinin görüntüsü

Bu emir için Şekil 5.8’de MLME-ASSOCIATE.indication primitifi görevlendirilir. Bu primitifin görevi üyelik isteği komut çerçevesinden aldığı gerekli parametreleri bir üst katmana yani ağ katmanına çıkarmak ve NLME-JOIN.indication ve MLME-ASSOCIATE.response primitiflerini görevlendirmektir. Bu primitif kısaca görevlendireceği primitifler için *node_1*, *node_2*, *node_3* ve *node_4*’in uzun adresi olan *00:00:00:29:48:23*, *00:00:08:BE:67:84*,

00:00:0A:E1:3D:6C ve 00:00:0C:D6:72:AE için ağ adresinin istendiğini bildirmektedir.

<pre> ----- Module (13), (top.Office Network.node_0.nwk) From procedure: gd_ADRES_ATAMA_NWK_process [executeMLME enter execs] ----- MLME_Association_indication ----- DeviceAddress:00:00:00:29:48:23 CapInf-a_pan_coord:1 CapInf-device_type:0 CapInf-power_source:0 CapInf-r_on_wh_idle:0 CapInf-sec_capability:0 CapInf-allocate_addr:1 SecurityUse:0 ACLEntry:8 ----- </pre>	<pre> ----- Module (13), (top.Office Network.node_0.nwk) From procedure: gd_ADRES_ATAMA_NWK_process [executeMLME enter execs] ----- MLME_Association_indication ----- DeviceAddress:00:00:08:BE:67:84 CapInf-a_pan_coord:1 CapInf-device_type:0 CapInf-power_source:0 CapInf-r_on_wh_idle:0 CapInf-sec_capability:0 CapInf-allocate_addr:1 SecurityUse:0 ACLEntry:8 ----- </pre>
<pre> ----- Module (13), (top.Office Network.node_0.nwk) From procedure: gd_ADRES_ATAMA_NWK_process [executeMLME enter execs] ----- MLME_Association_indication ----- DeviceAddress:00:00:0A:E1:3D:6C CapInf-a_pan_coord:1 CapInf-device_type:0 CapInf-power_source:0 CapInf-r_on_wh_idle:0 CapInf-sec_capability:0 CapInf-allocate_addr:1 SecurityUse:0 ACLEntry:8 ----- </pre>	<pre> ----- Module (13), (top.Office Network.node_0.nwk) From procedure: gd_ADRES_ATAMA_NWK_process [executeMLME enter execs] ----- MLME_Association_indication ----- DeviceAddress:00:00:0C:D6:72:AE CapInf-a_pan_coord:1 CapInf-device_type:0 CapInf-power_source:0 CapInf-r_on_wh_idle:0 CapInf-sec_capability:0 CapInf-allocate_addr:1 SecurityUse:0 ACLEntry:8 ----- </pre>

Şekil 5.9. Ağ katmanındaki MLME-ASSOCIATE.indication primitifinin görüntüsü

<pre> ----- Module (13), (top.Office Network.node_0.nwk) From procedure: gd_ADRES_ATAMA_NWK_process [executeMLME enter execs] ----- MLME_JOIN_indication ----- NetworkAddress:1 ExtendedAddress:00:00:00:29:48:23 CapInf-a_pan_coord:1 CapInf-device_type:0 CapInf-power_source:0 CapInf-r_on_wh_idle:0 CapInf-sec_capability:0 CapInf-allocate_addr:1 SecurityUse:0 ----- </pre>	<pre> ----- Module (13), (top.Office Network.node_0.nwk) From procedure: gd_ADRES_ATAMA_NWK_process [executeMLME enter execs] ----- MLME_JOIN_indication ----- NetworkAddress:2 ExtendedAddress:00:00:08:BE:67:84 CapInf-a_pan_coord:1 CapInf-device_type:0 CapInf-power_source:0 CapInf-r_on_wh_idle:0 CapInf-sec_capability:0 CapInf-allocate_addr:1 SecurityUse:0 ----- </pre>
<pre> ----- Module (13), (top.Office Network.node_0.nwk) From procedure: gd_ADRES_ATAMA_NWK_process [executeMLME enter execs] ----- MLME_JOIN_indication ----- NetworkAddress:3 ExtendedAddress:00:00:0A:E1:3D:6C CapInf-a_pan_coord:1 CapInf-device_type:0 CapInf-power_source:0 CapInf-r_on_wh_idle:0 CapInf-sec_capability:0 CapInf-allocate_addr:1 SecurityUse:0 ----- </pre>	<pre> ----- Module (13), (top.Office Network.node_0.nwk) From procedure: gd_ADRES_ATAMA_NWK_process [executeMLME enter execs] ----- MLME_JOIN_indication ----- NetworkAddress:4 ExtendedAddress:00:00:0C:D6:72:AE CapInf-a_pan_coord:1 CapInf-device_type:0 CapInf-power_source:0 CapInf-r_on_wh_idle:0 CapInf-sec_capability:0 CapInf-allocate_addr:1 SecurityUse:0 ----- </pre>

Şekil 5.10. Ağ katmanındaki NLME-JOIN.indication primitifi

Şekil 5.9’da ise MAC katmanından gelen MLME-ASSOCIATE.request primitifinin parametrelerini almak için ağ

katmanında tekraren kullanılmıştır ama buradaki tek görevi, NLME-JOIN.indication ve MLME-ASSOCIATE.response primitiflerine parametrelerini aktarmaktır.

Görevi devralan Şekil 5.10'daki NLME-JOIN.indication primitifi, MLME-ASSOCIATE.indication primitifiyle gelen ağ adresi isteğine eğer olumlu cevap verirse bir ağ adresi tahsisi yapar ve bunu kendi üst katmanı olan uygulama katmanına bu primitif ile bildirir.

<pre> ----- Module (13), (top.Office Network.node_0.nwk) From procedure: gd_ADRES_ATAMA_NWK_process [executeMLME enter execs] ----- MLME_Association_response ----- DeviceAddress:00:00:00:19:40:23 AssocShortAddress:1 status:0 SecurityEnable:0 ----- </pre>	<pre> ----- Module (13), (top.Office Network.node_0.nwk) From procedure: gd_ADRES_ATAMA_NWK_process [executeMLME enter execs] ----- MLME_Association_response ----- DeviceAddress:00:00:00:BE:67:84 AssocShortAddress:2 status:0 SecurityEnable:0 ----- </pre>
<pre> ----- Module (13), (top.Office Network.node_0.nwk) From procedure: gd_ADRES_ATAMA_NWK_process [executeMLME enter execs] ----- MLME_Association_response ----- DeviceAddress:00:00:0A:E1:3D:6C AssocShortAddress:3 status:0 SecurityEnable:0 ----- </pre>	<pre> ----- Module (13), (top.Office Network.node_0.nwk) From procedure: gd_ADRES_ATAMA_NWK_process [executeMLME enter execs] ----- MLME_Association_response ----- DeviceAddress:00:00:0C:D6:72:AE AssocShortAddress:4 status:0 SecurityEnable:0 ----- </pre>

Şekil 5.11. Ağ katmanındaki MLME-ASSOCIATE.response primitifi

Şekil 5.11'deki MLME-ASSOCIATE.response primitifi ise bir alt katman olan MAC katmanına tahsis edilen ağ adresini ve diğer bilgileri gönderir.

<pre> ----- Module (22), (top.Office Network.node_0.app) From procedure: gd_ADRES_ATAMA_APP_process [executeNLME enter execs] ----- NLME_JOIN_indication ----- NetworkAddress:2 ExtendedAddress:00:00:0C:D6:72:AE CapInf-a_pan_coord:1 CapInf-device_type:0 CapInf-power_source:0 CapInf-r_on_wh_idle:0 CapInf-sec_capability:0 CapInf-allocate_addr:1 RejoinNetwork:0 SecureRejoin:0 ----- </pre>	<pre> ----- Module (22), (top.Office Network.node_0.app) From procedure: gd_ADRES_ATAMA_APP_process [executeNLME enter execs] ----- NLME_JOIN_indication ----- NetworkAddress:1 ExtendedAddress:00:00:00:29:48:23 CapInf-a_pan_coord:1 CapInf-device_type:0 CapInf-power_source:0 CapInf-r_on_wh_idle:0 CapInf-sec_capability:0 CapInf-allocate_addr:1 RejoinNetwork:0 SecureRejoin:0 ----- </pre>
<pre> ----- Module (22), (top.Office Network.node_0.app) From procedure: gd_ADRES_ATAMA_APP_process [executeNLME enter execs] ----- NLME_JOIN_indication ----- NetworkAddress:2 ExtendedAddress:00:00:08:BE:67:84 CapInf-a_pan_coord:1 CapInf-device_type:0 CapInf-power_source:0 CapInf-r_on_wh_idle:0 CapInf-sec_capability:0 CapInf-allocate_addr:1 RejoinNetwork:0 SecureRejoin:0 ----- </pre>	<pre> ----- Module (22), (top.Office Network.node_0.app) From procedure: gd_ADRES_ATAMA_APP_process [executeNLME enter execs] ----- NLME_JOIN_indication ----- NetworkAddress:3 ExtendedAddress:00:00:0A:E1:3D:6C CapInf-a_pan_coord:1 CapInf-device_type:0 CapInf-power_source:0 CapInf-r_on_wh_idle:0 CapInf-sec_capability:0 CapInf-allocate_addr:1 RejoinNetwork:0 SecureRejoin:0 ----- </pre>

Şekil 5.12.Uygulama katmanındaki NLME-JOIN.indication primitifi

Şekil 5.12’de görülen NLME-JOIN.indication primitifi uygulama katmanına ulaşmış ve bildirimini yapmıştır.

<pre> ----- Module (8), (top.Office Network.node_0.mac) From procedure: gd_ADRES_ATAMA_MAC_process [executeNLME enter execs] ----- MLME_Association_response ----- DeviceAddress:00:00:08:BE:67:84 AssocShortAddress:2 Status:0 SecurityEnable:0 ----- </pre>	<pre> ----- Module (8), (top.Office Network.node_0.mac) From procedure: gd_ADRES_ATAMA_MAC_process [executeNLME enter execs] ----- MLME_Association_response ----- DeviceAddress:00:00:0A:E1:3D:6C AssocShortAddress:3 Status:0 SecurityEnable:0 ----- </pre>
<pre> ----- Module (8), (top.Office Network.node_0.mac) From procedure: gd_ADRES_ATAMA_MAC_process [executeNLME enter execs] ----- MLME_Association_response ----- DeviceAddress:00:00:0C:D6:72:AE AssocShortAddress:4 Status:0 SecurityEnable:0 ----- </pre>	<pre> ----- Module (8), (top.Office Network.node_0.mac) From procedure: gd_ADRES_ATAMA_MAC_process [executeNLME enter execs] ----- MLME_Association_response ----- DeviceAddress:00:00:00:29:48:23 AssocShortAddress:1 Status:0 SecurityEnable:0 ----- </pre>

Şekil 5.13.MAC katmanındaki MLME-ASSOCIATE.response primitifi

Şekil 5.13’teki MLME-ASSOCIATE.response primitifi ise üyelik yanıtı komut çerçevesini oluşturarak ve fiziksel katman özelliklerini kullanarak çerçeveyi yayınladı.

```

-----
| Module (35), (top.Office Network.node_1.mac)
| From procedure: qd_ADRES_ATAMA_MAC_process [addQueueel enter execs]
|
|
|-----
| Alinan komut çerçevesi:
|-----
| Security Enabled:0
| Frame Pending:0
| Ack Request:0
| Dest Addressing Mode:3
| Source Addressing Mode:3
| Sequence Number:12
| Intra-PAN:0
| Destination PAN identifier:65535
| Source PAN identifier:65535
| Destination Address:00:00:00:29:48:23
| Source Address:00:00:00:00:00:00
| MSDU'daki Command Frame Identifier:2
|-----
-----
| Module (42), (top.Office Network.node_2.mac)
| From procedure: qd_ADRES_ATAMA_MAC_process [addQueueel enter execs]
|
|
|-----
| Alinan komut çerçevesi:
|-----
| Security Enabled:0
| Frame Pending:0
| Ack Request:0
| Dest Addressing Mode:3
| Source Addressing Mode:3
| Sequence Number:98
| Intra-PAN:0
| Destination PAN identifier:65535
| Source PAN identifier:65535
| Destination Address:00:00:08:BE:67:84
| Source Address:00:00:00:00:00:00
| MSDU'daki Command Frame Identifier:2
|-----
-----
-----
| Module (706), (top.Office Network.node_3.mac)
| From procedure: qd_ADRES_ATAMA_MAC_process [addQueueel enter execs]
|
|
|-----
| Alinan komut çerçevesi:
|-----
| Security Enabled:0
| Frame Pending:0
| Ack Request:0
| Dest Addressing Mode:3
| Source Addressing Mode:3
| Sequence Number:9
| Intra-PAN:0
| Destination PAN identifier:65535
| Source PAN identifier:65535
| Destination Address:00:00:0A:E1:3D:6C
| Source Address:00:00:00:00:00:00
| MSDU'daki Command Frame Identifier:2
|-----
-----
-----
| Module (727), (top.Office Network.node_4.mac)
| From procedure: qd_ADRES_ATAMA_MAC_process [addQueueel enter execs]
|
|
|-----
| Alinan komut çerçevesi:
|-----
| Security Enabled:0
| Frame Pending:0
| Ack Request:0
| Dest Addressing Mode:3
| Source Addressing Mode:3
| Sequence Number:151
| Intra-PAN:0
| Destination PAN identifier:65535
| Source PAN identifier:65535
| Destination Address:00:00:0C:D6:72:AE
| Source Address:00:00:00:00:00:00
| MSDU'daki Command Frame Identifier:2
|-----
-----

```

Şekil 5.14.Uç cihaza üyelik yanıtı komut çerçevesinin görüntüsü

Şekil 5.14’de tüm uç düğümlerin MAC katmanına ulaşan üyelik yanıtı komut çerçevesinin bazı alanları gösterilmiştir. Bu çerçevedeki önemli noktalardan biri hedef ve kaynak adresleme modunda uzun adresleme yani MAC adresleme kullanılmıştır. İkincisi bu çerçevenin yük alanında komut çerçeve tanımlayıcı alanı “2” yani üyelik yanıtının geldiğini düğüme bildirmektedir. Bu bildirimde göre de MLME-ASSOCIATE.confirm primitifine görev devredilir.

<pre> ----- Module (35), (top.Office Network,node_1.mac) From procedure: qd_ADRES_ATAMA_MAC_process [addQueueI enter execs] ----- MLME_Association_confirm ----- AssocShortAddress:1 Status:0 ----- </pre>	<pre> ----- Module (62), (top.Office Network,node_2.mac) From procedure: qd_ADRES_ATAMA_MAC_process [addQueueI enter execs] ----- MLME_Association_confirm ----- AssocShortAddress:2 Status:0 ----- </pre>
<pre> ----- Module (700), (top.Office Network,node_3.mac) From procedure: qd_ADRES_ATAMA_MAC_process [addQueueI enter execs] ----- MLME_Association_confirm ----- AssocShortAddress:3 Status:0 ----- </pre>	<pre> ----- Module (727), (top.Office Network,node_4.mac) From procedure: qd_ADRES_ATAMA_MAC_process [addQueueI enter execs] ----- MLME_Association_confirm ----- AssocShortAddress:4 Status:0 ----- </pre>

Şekil 5.15.MAC katmanındaki MLME-ASSOCIATE.confirm primitifi

Şekil 5.14’teki üyelik yanıtı komut çerçevesi geldiğinde Şekil 5.15’teki MLME-ASSOCIATE.confirm primitifine üyelik durumu ve tahsis edilen ağ adresi parametrelerini sağlar ve bu primitif bu parametreleri edinmesi üzerine ağ katmanına iletilir.

<pre> ----- Module (40), (top.Office Network,node_1.nwk) From procedure: qd_ADRES_ATAMA_NWK_process [assocMLME enter execs] ----- MLME_Association_confirm ----- AssocShortAddress:1 Status:0 ----- </pre>	<pre> ----- Module (67), (top.Office Network,node_2.nwk) From procedure: qd_ADRES_ATAMA_NWK_process [assocMLME enter execs] ----- MLME_Association_confirm ----- AssocShortAddress:2 Status:0 ----- </pre>
<pre> ----- Module (705), (top.Office Network,node_3.nwk) From procedure: qd_ADRES_ATAMA_NWK_process [assocMLME enter execs] ----- MLME_Association_confirm ----- AssocShortAddress:3 Status:0 ----- </pre>	<pre> ----- Module (732), (top.Office Network,node_4.nwk) From procedure: qd_ADRES_ATAMA_NWK_process [assocMLME enter execs] ----- MLME_Association_confirm ----- AssocShortAddress:4 Status:0 ----- </pre>

Şekil 5.16.Ağ katmanındaki MLME-ASSOCIATE.confirm primitifi

Ağ katmanına gelen MLME-ASSOCIATE.confirm primitifi bu katmanda bilgi tabanında (Information Base) bulunan ilgili parametreye atanır ve bundan sonra bu parametreden sürekli olarak ağ adresi alınır. Şekil 5.16’da görüldüğü gibi uç düğüm olan *node_1*, *node_2*, *node_3* ve *node_4*’e ağ adresi olarak sırasıyla, “1”, “2”, “3” ve “4” atanmıştır ve bu atanan ağ adresi üst katmana da bildirilmelidir.

Bu bildirim yapan Şekil 5.17’da görülen NLME-JOIN.confirm primitifidir.

```

-----
| Module (40), (top.Office Network.node_1.nwk)
| From procedure: gd_ADRES_ATAMA_NWK_process [executeNLME enter execs]
|
|-----
| NLME_JOIN_confirm
|-----
| Status:0
| NetworkAddress:1
| ExtendedPANID:0
| ActiveChannel:11
|-----
-----
| Module (67), (top.Office Network.node_2.nwk)
| From procedure: gd_ADRES_ATAMA_NWK_process [executeNLME enter execs]
|
|-----
| NLME_JOIN_confirm
|-----
| Status:0
| NetworkAddress:2
| ExtendedPANID:0
| ActiveChannel:11
|-----
-----
| Module (705), (top.Office Network.node_3.nwk)
| From procedure: gd_ADRES_ATAMA_NWK_process [executeNLME enter execs]
|
|-----
| NLME_JOIN_confirm
|-----
| Status:0
| NetworkAddress:3
| ExtendedPANID:0
| ActiveChannel:11
|-----
-----
| Module (732), (top.Office Network.node_4.nwk)
| From procedure: gd_ADRES_ATAMA_NWK_process [executeNLME enter execs]
|
|-----
| NLME_JOIN_confirm
|-----
| Status:0
| NetworkAddress:4
| ExtendedPANID:0
| ActiveChannel:11
|-----

```

Şekil 5.17.Ağ katmanındaki NLME-JOIN.confirm primitifi

Özetle, bu çalışmada, ZigBee temelli bir kablosuz sensör ağında uç düğümlerin ağa üye olma sürecinin bir bölümü OPNET Modeller simülasyon aracında modelleme ve simülasyon yapılarak gerçekleştirilmiştir. Üyelik sürecindeki prosedürlerden olan ağ keşfi ve kimliklendirme atlanılarak sadece ağ adresi yani kısa adres ataması ele alınmıştır. Simülasyon sonucu olarak, uç düğümlere yani *node_1*, *node_2*, *node_3* ve *node_4*’e koordinatör olan *node_0* otomatik olarak sırasıyla “1”, “2”, “3”, ve “4” ağ adresleri atadığı görülmektedir. Burada dikkat çekilecek nokta koordinatöre yani *node_0* ‘a bağlanmak isteyen ilk uç cihaz kimse ona 1’den başlanarak sırası ile bir artarak ağ adresinin atanması yapıldığı gözlenmiştir. Yani burada paketler arası zaman uç cihazlar için değiştirilirse ona göre ilk ağ adresini alacak olan düğümde değişmiş olur.

6. SONUÇLAR

Kablosuz sensör ağlarında yapılan arařtırmalar günümüzde çok popüler bir hale gelmiştir. Bu popülerliğin altında yatan nedenler arasında uygulama alanlarının çok geniş olması ve nesnelerin interneti gibi geleceğin teknolojilerine ön ayak olması sayılabilir. Öte yandan kablosuz sensör ağlarında kullanılan düğümler sınırlı miktarda enerji, sınırlı miktarda bellek ve sınırlı miktarda işlemciye sahiptir. Tüm bu sınırlamaların da etkisini azaltmak ve bunun gibi birçok problemin çözüm beklemesi arařtırmacıların ilgisini kablosuz sensör ağlarına çekmektedir. Tüm bunları müteakip bu sistemleri kurmak ve üzerinde çalışma yapmak da bir o kadar maliyetli olmaktadır. Bu maliyeti azaltmak için bu sistemler kurulmadan önce modellenip bir simülasyon ortamında simüle edilir ve analiz sonuçları değerlendirilir.

Bu kitap çalışmasında, katmanlı bir protokol mimarisine sahip olan ZigBee tabanlı bir kablosuz sensör ağındaki koordinatör düğümün, katılım isteğinde bulunan uç cihazlara, ağa katılım isteklerinin kabul edilip, ağ adresi ataması yapılması süreci gerçekleştirilmiştir. Bunun için IEEE 802.15.4 ve ZigBee standartlarına uygun olarak düğümler Opnet Modeler'de modellenip simüle edilerek sonuçları analiz edilmiştir. Bu süreç için oluşturulan primitif yapıları ve paket biçimlerinin uyumlu çalıştığı ve uç cihazlara ağ adresleri istenilen şekilde yani ilk ağ adresini hangi düğüm talep etmişse o düğümden başlanarak N tane düğüm kullandığı varsayarsak, sırası ile 1 'den N 'e kadar bir artarak düğümlere ağ adresi atandığı görülmüştür. Ayrıca bu çalışmada düğümlerin hava ortamında yayınladığı paketlerin

çakışmasını önlemek için tüm düğümlerde aralarında asal olan tam sayılar, paketler arası süre olarak ayarlanmıştır.

Ayrıca bu çalışma ile katmanlı bir protokol mimarisinin genel kullanımını uygulamalı olarak sunduğu için okuyuculara hem teorik hem de pratik bir temel sağlaması bakımından önemlidir.

KAYNAKÇA

- Aadri, A., & Idrissi, N. (2017). A Cluster Based Routing Algorithm Minimizing Energy Consumption in WSN. *International Journal of Wireless & Mobile Networks*, 9(4), 49-62.
<https://doi.org/10.5121/ijwmn.2017.9405>
- ABBA ARI, A. A., GUEROUI, A., LABRAOUI, N., & YENKE, B. O. (2015). Concepts And Evolution Of Research In The Field Of Wireless Sensor Networks. *International journal of Computer Networks & Communications*, 7(1), 81-98.
<https://doi.org/10.5121/ijcnc.2015.7106>
- Agarkhed, J., Dattatraya, P. Y., & Patil, S. R. (2020). A Study of Wireless Sensor Networks to Comprehend their Relevance to Different Applications. *Journal of Telecommunications and Information Technology*, 2(2020), 3-13.
<https://doi.org/10.26636/jtit.2020.139219>
- Al, M., & Bakheet, F. (2015). *Hierarchical Routing Protocols of Wireless Sensor Networks: A Comparison*.
- Aliouat, Z., & Harous, S. (2014). Energy efficient clustering for wireless sensor networks. *International Journal of Pervasive Computing and Communications*, 10(4), 469-480.
<https://doi.org/10.1108/IJPC-05-2014-0033>

- Amaxilatis, D., Chatziannakis, I., Koninis, C., & Pyrgelis, A. (2011). *Component Based Clustering in Wireless Sensor Networks*. <http://arxiv.org/abs/1105.3864>
- Ameen, S. Y., & Nourildean, S. W. (2013). Coordinator and router investigation in IEEE802.15.4 ZigBee wireless sensor network. *2013 International Conference on Electrical Communication, Computer, Power, and Control Engineering (ICECCPCE)*, 13, 130-134. <https://doi.org/10.1109/ICECCPCE.2013.6998748>
- Digi International Inc. (2008). *An Introduction to ZigBee* ®.
- Bhatia, S., Jaffery, Z. A., & Mehfuz, S. (2023). Development and Analysis of IoT based Smart Agriculture System for Heterogenous Nodes. *2023 International Conference on Recent Advances in Electrical, Electronics & Digital Healthcare Technologies (REEDCON)*, 62-67. <https://doi.org/10.1109/REEDCON57544.2023.10150466>
- Biswas, S. (2017). Simulation model of beacon enabled 802.15.4 networks with OPNET modeler. *2017 International Electrical Engineering Congress (iEECON)*, 1-4. <https://doi.org/10.1109/IEECON.2017.8075839>
- Canli, T., & Khokhar, A. (2009). PRMAC: Pipelined Routing Enhanced MAC Protocol for Wireless Sensor Networks. *2009 IEEE International Conference on Communications*, 1-5. <https://doi.org/10.1109/ICC.2009.5199364>

- Das, R., Bal, S., Das, S., Sarkar, M. K., Majumder, D., Chakraborty, A., & Majumder, K. (2016). Performance analysis of various attacks under AODV in WSN & MANET using OPNET 14.5. *2016 IEEE 7th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, 1-9. <https://doi.org/10.1109/UEMCON.2016.7777831>
- Dash, S., Kumar, S., Lenka, M. R., & Swain, A. R. (2019). Multi-Token Based MAC-Cum-Routing Protocol for WSN: A Distributed Approach. *Journal of Communications Software and Systems*, 15(3). <https://doi.org/10.24138/jcomss.v15i3.709>
- Deepika, & Sharma, M. (2014). Effective data flow in ZigBee network using OPNET. *2014 International Conference on Communication and Signal Processing*, 1155-1158. <https://doi.org/10.1109/ICCSP.2014.6950035>
- Doğan, G., Yıldırım, G., & Tatar, Y. (2018). Opnet Modeler Ortamında WSN Protokollerinin Gerçekleştirme Süreçlerinin İncelenmesi. *Academic Perspective Procedia*, 1(1), 105-114. <https://doi.org/10.33793/acperpro.01.01.23>
- Duy Tan, N., Nguyen, D.-N., Hoang, H.-N., & Le, T.-T.-H. (2023). EEGT: Energy Efficient Grid-Based Routing Protocol in Wireless Sensor Networks for IoT Applications. *Computers*, 12(5), 103. <https://doi.org/10.3390/computers12050103>

- Fairhurst, G. (2025, Nisan 17). *Service Primitives*.
<https://www.erg.abdn.ac.uk/users/gorry/course/intro-pages/service-prim.html>
- Gaikwad, P. P., Gabhane, J. P., & Golait, S. S. (2015). A survey based on Smart Homes system using Internet-of-Things. *2015 International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC)*, 0330-0335.
<https://doi.org/10.1109/ICCPEIC.2015.7259486>
- Gupta, D., Wadhwa, S., Rani, S., Khan, Z., & Boulila, W. (2023). EEDC: An Energy Efficient Data Communication Scheme Based on New Routing Approach in Wireless Sensor Networks for Future IoT Applications. *Sensors*, 23(21), 8839.
<https://doi.org/10.3390/s23218839>
- IEEE Computer Society (Ed.). (2003). *IEEE Std 802.15.4-2003, Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)*. IEEE.
- Jabbar, S., Minhas, A., Imran, M., Khalid, S., & Saleem, K. (2015). Energy Efficient Strategy for Throughput Improvement in Wireless Sensor Networks. *Sensors*, 15(2), 2473-2495.
<https://doi.org/10.3390/s150202473>
- Jawhar, I., Mohamed, N., & Al-Jaroodi, J. (2018). Networking architectures and protocols for smart city systems. *Journal of*

Internet Services and Applications, 9(1), 26.
<https://doi.org/10.1186/s13174-018-0097-0>

Karimi-Bidhendi, S., Guo, J., & Jafarkhani, H. (2019). Using Quantization to Deploy Heterogeneous Nodes in Two-Tier Wireless Sensor Networks. *2019 IEEE International Symposium on Information Theory (ISIT)*, 1502-1506.
<https://doi.org/10.1109/ISIT.2019.8849271>

Kaushal, K., Kaur, T., & Kaur, J. (2014). *ZigBee based Wireless Sensor Networks*. 7752-7755. www.ijcsit.com

Koshariya, A. K., Kalaiyarasi, D., Jovith, A. A., Sivakami, T., Hasan, D. S., & Boopathi, S. (2023). *AI-Enabled IoT and WSN-Integrated Smart Agriculture System* (ss. 200-218).
<https://doi.org/10.4018/978-1-6684-8516-3.ch011>

Kucuk, K., Bandirmali, N., & Kavak, A. (2012). *MODELING OF THE MODIFIED SSLE IN OPNET FOR LARGE SCALE WIRELESS SENSOR NETWORKS*.

Lee, J.-S., & Huang, Y.-C. (2006). Design and implementation of ZigBee/IEEE 802.15.4 Nodes for Wireless Sensor Networks. *Çinde Feature www.instm.org.uk* (C. 204, Sayı 7).
www.instm.org.uk

Liu, F., Tang, Y., & Wang, L. (2019). eHAPAC: A Privacy-Supported Access Control Model for IP-Enabled Wireless Sensor Networks. *Sensors*, 19(7), 1513. <https://doi.org/10.3390/s19071513>

- Louail, L., Felea, V., Bernard, J., & Guyennet, H. (2015). MAC-aware routing in wireless sensor networks. *2015 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom)*, 225-229.
<https://doi.org/10.1109/BlackSeaCom.2015.7185120>
- Mamun, Q. (2012). A Qualitative Comparison of Different Logical Topologies for Wireless Sensor Networks. *Sensors*, *12*(11), 14887-14913. <https://doi.org/10.3390/s121114887>
- Matharu, G. S., Upadhyay, P., & Chaudhary, L. (2014). The Internet of Things: Challenges & security issues. *2014 International Conference on Emerging Technologies (ICET)*, 54-59. <https://doi.org/10.1109/ICET.2014.7021016>
- Meelu, R., Anand, R., Patel, R. B., & Singh, B. P. (2010). *Energy Efficiency of Cluster-based Routing Protocols used in Wireless Sensor Networks*. 109-113. <https://doi.org/10.1063/1.3526169>
- Mouradian, A., Augé-Blum, I., & Valois, F. (2014). RTXP: A localized real-time MAC-routing protocol for wireless sensor networks. *Computer Networks*, *67*, 43-59.
<https://doi.org/10.1016/j.comnet.2014.03.020>
- Nasri, M., Helali, A., & Maaref, H. (2021). Energy-efficient fuzzy logic-based cross-layer hierarchical routing protocol for wireless Internet-of-Things sensor networks. *International Journal of Communication Systems*, *34*(9). <https://doi.org/10.1002/dac.4808>

- Pan, J., & Jain, R. (2008). A survey of network simulation tools: Current status and future developments. *Email: jp10@cse.wustl.edu*, 2(4), 45.
- Patel, R., Patel, N., & Patel, S. (2019). *An Approach to Analyze Behavior of Network Events in NS2 and NS3 Using AWK and Xgraph* (ss. 137-147). https://doi.org/10.1007/978-981-13-0586-3_14
- Portocarrero, J. M. T., Delicato, F. C., Pires, P. F., Nakagawa, E. Y., & Oquendo, F. (2015). Self-Adaptive Middleware for Wireless Sensor Networks. *Proceedings of the 2015 European Conference on Software Architecture Workshops*, 1-8. <https://doi.org/10.1145/2797433.2797445>
- Pundir, S., Wazid, M., Singh, D. P., Das, A. K., Rodrigues, J. J. P. C., & Park, Y. (2020). Intrusion Detection Protocols in Wireless Sensor Networks Integrated to Internet of Things Deployment: Survey and Future Challenges. *IEEE Access*, 8, 3343-3363. <https://doi.org/10.1109/ACCESS.2019.2962829>
- Randhawa, S. (2014). Research Challenges in Wireless Sensor Network: A State of the Play. *ArXiv, abs/1404.1469*. <https://api.semanticscholar.org/CorpusID:17657469>
- Saba, T., Haseeb, K., Ud Din, I., Almogren, A., Altameem, A., & Fati, S. M. (2020). EGCIR: Energy-Aware Graph Clustering and Intelligent Routing Using Supervised System in Wireless Sensor

Networks. *Energies*, 13(16), 4072.
<https://doi.org/10.3390/en13164072>

Sharma, V., Patel, R. B., Bhadauria, H. S., & Prasad, D. (2016). Deployment schemes in wireless sensor network to achieve blanket coverage in large-scale open area: A review. *Egyptian Informatics Journal*, 17(1), 45-56.
<https://doi.org/10.1016/j.eij.2015.08.003>

Song, X. (2011). Modeling and simulation of WSN routing protocols. *2011 IEEE 3rd International Conference on Communication Software and Networks*, 586-590.
<https://doi.org/10.1109/ICCSN.2011.6014793>

Sun, W., Tang, M., Zhang, L., Huo, Z., & Shu, L. (2020). A Survey of Using Swarm Intelligence Algorithms in IoT. *Sensors*, 20(5), 1420. <https://doi.org/10.3390/s20051420>

Verma, R. K., Pattanaik, K. K., & Bharti, S. (2015). An adaptive mechanism for improving resiliency in Wireless Sensor Networks. *2015 IEEE 10th International Conference on Industrial and Information Systems (ICIIS)*, 525-530.
<https://doi.org/10.1109/ICIINFS.2015.7399067>

Volpi, A., Tebaldi, L., Matrella, G., Montanari, R., & Bottani, E. (2023). Low-Cost UWB Based Real-Time Locating System: Development, Lab Test, Industrial Implementation and Economic Assessment. *Sensors*, 23(3), 1124.
<https://doi.org/10.3390/s23031124>

- Winkelman, R. (2025, Nisan 16). *Protocol*.
<https://fcit.usf.edu/network/chap2/chap2.htm>
- Yang, T., Kulla, E., Oda, T., Barolli, L., Younas, M., & Takizawa, M. (2012). Performance Evaluation of WSNs for MAC and Routing Protocols Considering Depletion and Routing Efficiency Metrics. *2012 Seventh International Conference on Broadband, Wireless Computing, Communication and Applications*, 503-510.
<https://doi.org/10.1109/BWCCA.2012.123>
- Yick, J., Mukherjee, B., & Ghosal, D. (2008). Wireless sensor network survey. *Computer Networks*, 52(12), 2292-2330.
<https://doi.org/10.1016/j.comnet.2008.04.002>
- Yu, Y., & Yao, Y. (2012). Improved AODV routing protocol for wireless sensor networks and implementation using OPNET. *2012 Third International Conference on Intelligent Control and Informatio Processing*, 709-713.
<https://doi.org/10.1109/ICICIP.2012.6391523>
- ZigBee Alliance. (2005). *ZigBee Specification*.
- ZigBee Alliance. (2012). *ZigBee Specification*.

KABLOSUZ SENSÖR AĞLARINDA ZigBee TABANLI ADRESLEME: TEORİ, MODELLEME VE SİMÜLASYON