KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

IETF 6TİSCH PROTOKOLÜ İÇİN DAĞITIK KULLANICI KİMLİK DOĞRULAMA MEKANİZMASI

YÜKSEK LİSANS TEZİ

HAKAN AYDIN

MAYIS 2018
TRABZON

KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsünce

Unvanı Verilmesi İçin Kabul Edilen Tezdir.

Tezin Enstitüye Verildiği Tarih : / /
Tezin Savunma Tarihi : / /

Tez Danışmanı :

Trabzon

KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Bilgisayar Mühendisliği Anabilim Dalında
Hakan AYDIN

IETF 6TiSCH PROTOKOLÜ İÇİN DAĞITIK KULLANICI KİMLİK DOĞRULAMA MEKANİZMASI

başlıklı bu çalışma, Enstitü Yönetim Kurulunun 08 / 05 / 2018 gün ve 1752 sayılı kararıyla oluşturulan jüri tarafından yapılan sınavda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

Başkan : Prof. Dr. Rıfat YAZICI .....

Üye : Dr.Öğr.Üyesi SEDAT GÖRMÜŞ .....

Üye : Dr.Öğr.Üyesi GÜZİN ULUTAŞ .....

Prof. Dr. Sadettin KORKMAZ
Enstitü Müdürü

ÖNSÖZ

Bu tez, Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Yüksek Lisans Programı'nda yapılan bir çalışmadır.

Tez çalışması içerisinde Nesnelerin İnterneti teknolojisindeki güvenli haberleşme için kullanılan mekanizmalar ve var olan protokoller küçük cihazların mimari yapısı göz önüne alınarak incelenmiştir. Bu katmanlardaki güvenlik mekanizmalarından ve karşılaşılan zorluklardan bahsedilip; güvenli bir Nesnelerin İnterneti için atılması gereken adımlar hakkında öneriler ortaya konulmuştur. Bu bağlamda 6TiSCH gibi yeni nesil Nesnelerin İnterneti protokolü tanıtılarak; bu protokol için muhtemel yeni bir kimlik doğrulama mekanizması önerilmiştir. Önerilen mekanizmanın sonuçları güvenlik gereksinimleri dikkate alınarak tartışılmıştır.

Tez çalışmamın planlanmasında, araştırılmasında, yürütülmesinde ve oluşumunda ilgi ve desteğini esirgemeyen, engin bilgi ve tecrübelerinden yararlandığım, yönlendirme ve bilgilendirmeleriyle çalışmamı bilimsel temeller ışığında şekillendiren ve sürekli motivasyonumu sağlayan, bana değerli zamanını ayıran sayın hocam Dr.Öğr.Üyesi Sedat GÖRMÜŞ'e;

Bu zorlu tez sürecinde benden desteğini esirgemeyen kıymetli arkadaşım, Ahmet Faruk Yavuz'a ve bu çalışmayı hazırlarken geçirdiğim süreçte benden yardımlarını esirgemeyen Mavi Alp Bilgi Teknolojileri Tic. Ltd. Şti. çalışanlarına;

Eğitim hayatım boyunca maddi ve manevi desteklerini esirgemeyen her zaman yanımda olan kardeşim Gökhan olmak üzere tüm hayatım boyunca olduğu gibi bu çalışmalarım süresince de benden her türlü desteklerini esirgemeyen değerli aileme

Sonsuz teşekkürü bir borç bilirim.

Hakan AYDIN

Trabzon 2018

TEZ ETİK BEYANNAMESİ

Yüksek Lisans Tezi olarak sunduğum "IETF 6TiSCH Protokolü için Dağıtık Kullanıcı Kimlik Doğrulama Mekanizması" başlıklı bu çalışmayı danışmanım Dr.Öğr.Üyesi Sedat GÖRMÜŞ'ün sorumluluğunda tamamladığımı, başka kaynaklardan aldığım bilgileri metinde ve kaynakçada eksiksiz olarak gösterdiğimi, tez sürecinde bilimsel araştırma ve etik kurallara uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul ettiğimi beyan ederim. 25/05/2018

Hakan AYDIN

İÇİNDEKİLER

Sayfa No

Yüksek Lisans Tezi

ÖZET

IETF 6TiSCH PROTOKOLÜ İÇİN DAĞITIK KULLANICI KİMLİK DOĞRULAMA MEKANİZMASI

Hakan AYDIN

Karadeniz Teknik Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı
Danışman: Dr.Öğr.Üyesi Sedat GÖRMÜŞ
2018, 100 Sayfa

İnternetin geleceği olarak tanımlanan "Nesnelerin İnterneti" çevreyle etkileşime giren, birbirleriyle iletişim kuran ve internet üzerinden kontrol edilen küçük akıllı nesnelerle çevrili yeni bir dünya vaat etmektedir. IoT ağlarının, milyarlarca küçük cihazın internete entegre edilmesi ve bu sayede şehir otomasyonu ile yaşlılar için evde sağlık çözümleri gibi birçok uygulama alanını kapsamaktadır. Bu geniş uygulama yelpazesi, yüksek güvenilirlik, düşük güç ve düşük gecikme iletişimi gereksinimi gibi benzersiz zorluklar ortaya koymaktadır.

Bu zorluklara ek olarak, bu tür kaynak kısıtlı cihazların, İnternet'in getirdiği güvenlik zorluklarıyla baş etmek için gerekli güvenlik önlemleri ile donatılması gerekecektir. Dahası bu çözümler güvenlik zorluklarını mikro denetleyici vasıtasıyla, mümkün olan az enerji tüketen sınırlı bir işlem gücü ve bellekle ele almak zorundadır. Bu tez kapsamında, yeni tanıtılan IETF 6TiSCH protokolünün güvenli önyükleme protokolüne bir eklenti sunulmuştur; burada kimlik doğrulama anahtarları, etkili kimlik doğrulama ve önyükleme sürecini etkinleştirmek için IoT ağının güvenilir düğümlerinde dağıtılmıştır.

Dağıtık bir yaklaşım kullanılarak standart IETF 6TiSCH kimlik doğrulama protokolünün iletişimini azaltmak ve kimlik doğrulama belirteçlerini IoT ağının kenarında tutarak ağın enerji verimliliğini artırmak hedeflenmiştir.

Anahtar Kelimeler: Kablosuz Duyarga Ağlar, Nesnelerin İnterneti, Gömülü Cihazlar, IEEE, IETF, 6LoWPAN, 6TiSCH.

Master Thesis

SUMMARY

A DISTRIBUTED USER AUTHENTICATION MECHANISM FOR IETF 6TiSCH PROTOCOL

Hakan AYDIN

Karadeniz Technical University
The Graduate School of Natural and Applied Sciences
Computer Engineering Graduate Program
Supervisor: Asst. Prof. Sedat GÖRMÜŞ
2018, 100 Pages

The "Internet of Objects", which is defined as the future of the Internet, promises a new world that interacts with the environment, is surrounded by small intelligent objects that communicate with each other and are controlled over the Internet. IoT networks are expected to integrate billions of small devices to the Internet enabling countless applications ranging from automation of cities to home based healthcare solutions for elderly. Such wide range of application pose unique challenges such as the need for high reliability, low power and low delay communications.

In addition to these challenges, such constrained devices will have to be equipped with the necessary security suits to cope with security challenges posed by the Internet. Furthermore, these solutions have to address such unique challenges via a microcontroller generally with a limited processing power and memory consuming as little energy as possible. This thesis presents an extension to the secure bootstrapping protocol of the newly introduced IETF 6TiSCH protocol where the authentication keys are distributed within the trusted nodes of the IoT network to enable an efficient authentication and bootstrapping process.

It was aimed to reduce the communication of the standard IETF 6TiSCH authentication protocol using a distributed approach and increase the energy efficiency of the network by keeping the authentication tokens at the edge of the IoT network.

Key Words: Wireless Sensor Networks, Internet Of Things, Embedded Devices, IEEE, IETF, 6LoWPAN, 6TiSCH.

ŞEKİLLER DİZİNİ

Sayfa No

TABLOLAR DİZİNİ

Sayfa No

SEMBOLLER DİZİNİ

6LowPAN	IPv6 over Low power Wireless Personal Area Networks
LLN	Low-Power and Lossy Networks
AES	Advanced Encryption Standard
CoAP	Constrained Application Protocol
DODAG	Destination Oriented Directed Acyclic Graph
DTLS	Datagram Transport Layer Security
ECC	Elliptic Curve Cryptography
IoT	Internet of Things
IP	Internet Protocol
IPv6	Internet Protocol Version 6
IPHC	IP Header Compression
KB	Kilobyte
LowPAN	Low-Power Wireless Personal Area Network
MAC	Media Access Control
PHY	Physical Layer
PKC	Public Key Cryptography
PSK	Pre-Shared Key
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol
WSN	Wireless Sensor Network
NH	Next Header
NHC	Next Header Compression
ROM	Read Only Memory
RAM	Random Access Memory
IEEE	Institute of Electrical & Electronic Engineers
IETF	Internet Engineering Task Force
ROLL	Routing over Low-Power and Lossy Networks
RPL	IPv6 Routing Protocol for Low-Power and Lossy Networks
6TiSCH	IPv6 over the TSCH mode of IEEE 802.15.4

JN	Joined Node
JP	Join Proxy
JRC	Join Registrar/Coordinator
P-JRC	Proxy Join Registrar/Coordinator
P	Pledge
DIO	DODAG Information Object
DAO	Destination Advertisement Object
DIS	DODAG Information Solicitation
OF	Objective Function
EB	Enhanced Beacon
IE	Information Element
ASN	Absolute Slot Number
Tx	Transmit
Rx	Reception
MTU	Maximum Transmission Unit
PAN	Personal Area Network
RFD	Reduced Function Device
CSMA/CA	Carrier Sense Multiple Access / Collision Avoidance
CCA	Clear Channel Assessment
FFD	Full Function Device
OF	Objective Function
PCE	Path Computation Element
QoS	Quality of Service
ACE	Authentication Coordination Element
MLME	MAC Layer Management Entity

1. GENEL BİLGİLER

1.1. Motivasyon

Geleceğin İnterneti ile ilişkili yeni kavrama "Nesnelerin İnternet'i (IoT)" denmektedir ve burada gerçek nesnelerin internetin bir parçası haline geldiği yapılardan söz edilmektedir. Her nesnenin benzersiz şekilde tanımlandığı ve ağa erişilebildiği vizyon açıklanmaktadır. Nesnelerin interneti, kablosuz duyargalar ve nano teknoloji gibi önemli alanlarda teknik yeniliklere dayanan teknolojik devrim olarak değerlendirilmektedir. Bu teknolojiyi kullanan ağlar, milyarlarca küçük cihazın internete entegre edilmesi ve bu sayede şehir otomasyonu ile yaşlılar için evde sağlık çözümleri gibi sayısız uygulama alanına hizmet etmesi beklenmektedir.

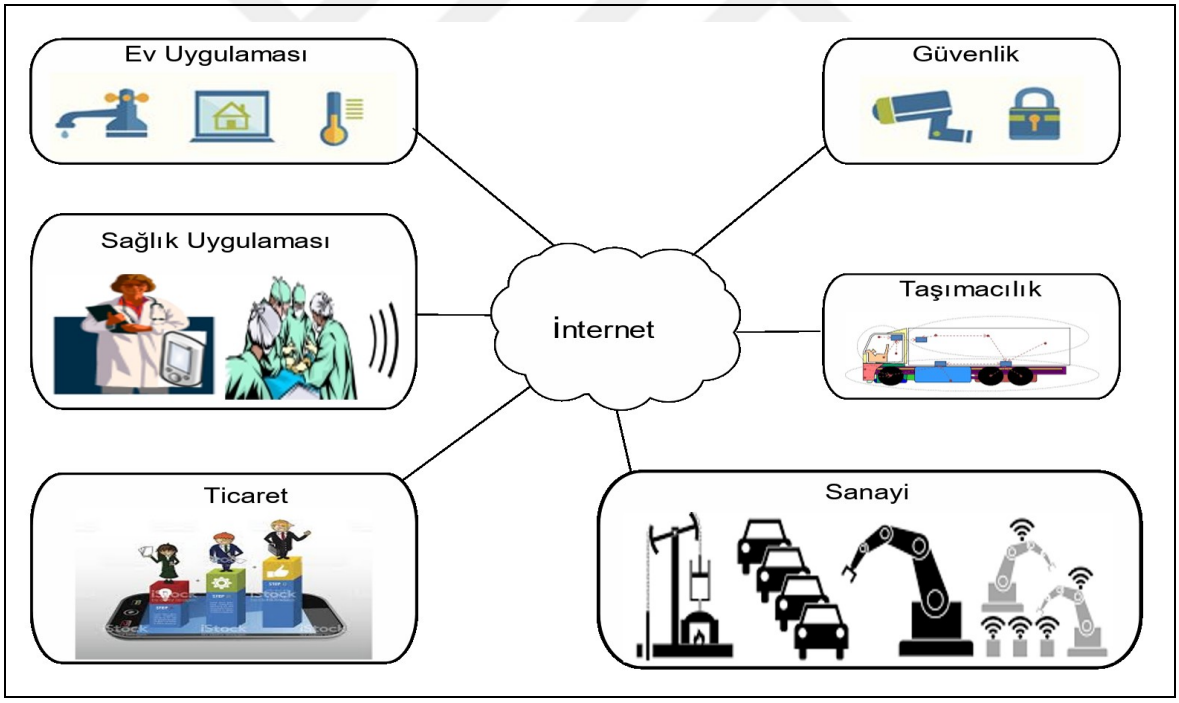
IoT'nin üretim ortamına girişi, endüstriyel süreç izleme ve otomasyon gibi kritik uygulamalarda kullanılan düşük güç kablosuz ağlardan yararlanarak, gelişmekte olan Endüstriyel Nesnelerin İnternetine önderlik etmektedir. Endüstriyel ihtiyaçların karşılanmasında temel yapı taşı, 2012 yılında standart haline gelen 802.15.4 protokolüne zaman paylaşımlı (Time Slotted) ve kanal atlamalı(Channel-Hopping) ortam erişim yönteminin eklenmesidir [1].

Bu zorluklara ek olarak, bu tür İnternet tabanlı küçük cihazların, İnternet'in getirdiği güvenlik zorluklarıyla baş etmek için gerekli güvenlik önlemleri ile donatılması gerekmektedir. Dahası, bu çözümler, benzersiz zorlukları, genellikle sınırlı işlem gücü ve mümkün olduğunca az enerji tüketen mikro denetleyici aracılığıyla çözmek zorundadır. Cihazların kısıtlı enerji, işlem kabiliyeti ve depolama alanlarına sahip olmaları, ağa dahil sürecinde güvenlik gereksinimlerinden olan kimlik doğrulama tasarımını güçleştirmektedir. Bu tezde, yukarıda bahsedilen kısıtlamalar dikkate alınarak; yeni tanıtılan IETF 6TiSCH protokolünün güvenli önyükleme protokolüne bir uzantı sunulmaktadır. Çalışmada dağıtık bir kimlik doğrulama yaklaşım kullanılarak IETF 6TiSCH kimlik doğrulama protokolünün haberleşme maliyetinin azaltılması ve doğrulama parametrelerinin IoT ağının kenarında tutularak ağın enerji verimliliğinin arttırılması hedeflenmiştir.

1.2. Giriş

Kablosuz duyarga ağları (KDA), küçük boyutlu, düşük maliyetli ve kısa mesafede kablosuz ortam üzerinden haberleşebilen duyarga düğümlerinin bir araya gelmesiyle oluştuğu ağlardır. Bu ağlarda, düğümler rastgele olarak ortama bırakılabilmekte ve protokoller yardımı ile kablosuz ortam üzerinden birbirleriyle haberleşebilmektedir.

Günümüzde kablosuz duyarga ağları sağlık, inşaat sektörü, kimya, çevre izleme ve evleri uzaktan kontrol etme gibi geniş bir yelpazede kullanılmaktadırlar. İnternete bağlanmış kablosuz duyarga ağlarına ait bazı örnekler Şekil 1'de verilmiştir. Bu şekilde İnternete bağlı küçük cihazların oluşturacağı yeni İnternete "Nesnelerin İnterneti" adı verilmektedir. Hayatımızda önemli bir yere sahip olacağı düşünülen bu teknoloji akıllı sayaçlar ve akıllı şehirler kavramlarıyla karşımıza çıkmaktadır [2].



Şekil 1. Nesnelerin İnterneti kullanım alanları

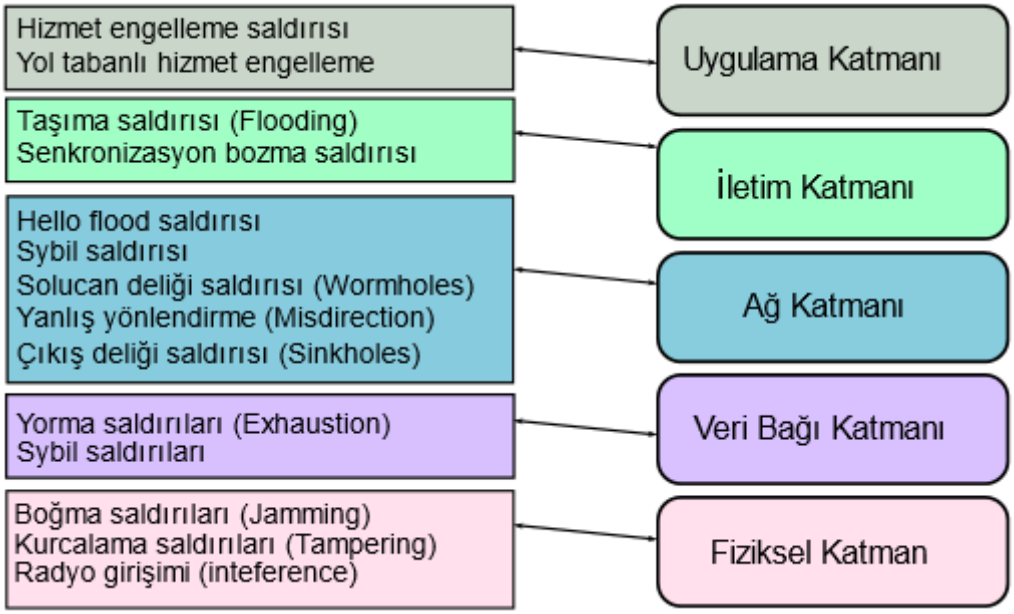
Nesnelerin İnterneti kavramı Uluslararası Telekomünikasyon Birliği (ITU) tarafından, herhangi bir yerde herhangi bir zamanda her cihazın birbirleriyle haberleşebileceği bir teknoloji olarak tanımlanmaktadır. Yapılan tanımlar dikkate alındığında bu teknoloji, tüm cihazların çeşitli iletişim protokolleri ve algılama yöntemleri aracılığıyla tanımlanarak

birbirleri ile iletişime geçebileceği, İnternete erişebileceği akıllı ağlardan oluşan sistemler bütünü olarak tanımlanmaktadır [3].

Nesnelerin İnterneti fikrinin temeli 1991 yılında, Cambridge Üniversitesi'ndeki 15 akademisyenin kahve makinesinin kullanılıp/kullanılmadığı durumunu takip etmek için kurduğu kameralı sistem olarak kabul edilmektedir [4]. 1999 yılında IoT kavramının önerilmesi, 2005 yılında Uluslararası Telekomünikasyon Birliği tarafından bu yeni kavrama dair raporun yayımlanması Nesnelerin İnternetine olan ilgiyi bir hayli artırmış durumdadır [5]. 2010 yılından itibaren İnternet evrimi; 4. Nesil İnterneti oluşturmaktadır. 4. Nesil Internet; Nesnelerin interneti ve Web 3.0'ı kapsayacak şekilde, canlı-cansız tüm cihazların internet ağına ilişkilendirilmesi olarak tarif edilmektedir. Ayrıca Nesnelerin İnterneti fikrini, çeşitli iletişim protokolleri aracılığıyla haberleşebilen ve birbirlerine bağlanarak, bilgi alışverişinde bulunan akıllı cihazlar sistemi olarak da tanımlamak mümkündür.

Artan pazar payında söz sahibi olmak isteyen firmalar nesnelerin interneti kavramının gerçekleştirilmesi için destek vermektedirler. Cisco'nun şehirleri daha etkili yönetmek için "Akıllı kentleşme" ağ projesi, IBM'in şirketi dünyayı ortak bir ağda birleştirmek için "Akıllı Planet" projesi, Microsoft'un çok sayıda Avrupa ülkesinin su ve hava kalitesinin görülebileceği "Eye on Earth" projesi ve General Electric'in kentlerde çevresel sorunları çözmek için "Ecomagination" ağ projesi örnek olarak verilebilir [6]. Cisco tarafından yapılan bir araştırmaya [7] göre 2003 yılında 500 milyon cihazın internete bağlı ve kişi başına düşen cihaz sayısı 0,08 iken 2010 yılında cihaz sayısı 12,5 milyara ve kişi başına düşen cihaz sayısı ise 1.84'e çıkmıştır. 2020 yılında Nesnelerin İnterneti kapsamında İnternete bağlanacak cihaz sayısının 50 milyara ulaşacağı tahmin edilmektedir.

IoT ağlarında yer alan cihazların, İnternet erişiminin getirdiği güvenlik sorunları ile başa çıkması gerekecektir. Kısıtlı bant genişliği, hafıza ve hesaplama yeteneğine sahip olmaları, onları bu tür tehditlere karşı savunmasız bırakmaktadır. Bu sebeple, cihazların geleneksel güvenlik tekniklerini kullanarak İnternetin ortaya koyduğu güvenlik sorunlarıyla başa çıkmaları mümkün görülmemektedir. Yapılan çalışmalar sonucunda kablosuz duyarga ağ mimarisinin farklı özellikteki saldırılara karşı savunmasız olduğu görülmektedir [8, 9, 10, 11]. Şekil 2'de farklı katmanlar için ortaya konulan saldırı türleri özetlenmektedir.



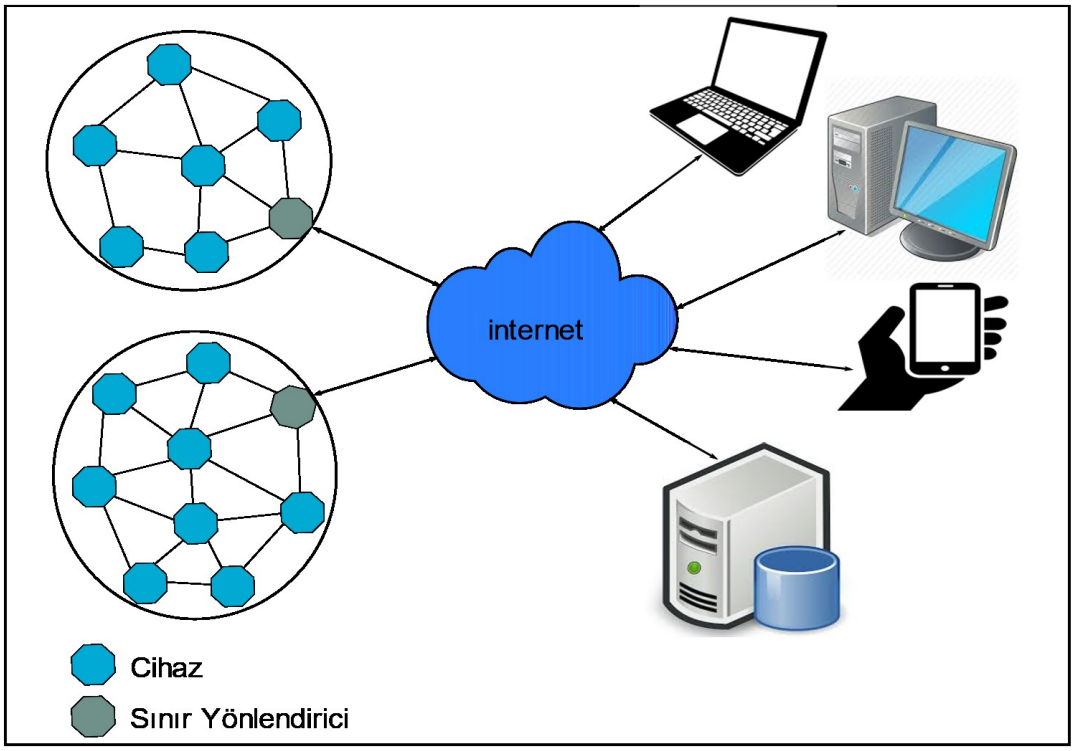
Şekil 2. Kablosuz duyarga ağ katmanlarını etkileyen saldırı türleri

Bu saldırıların ortaya koyduğu ve İnternet'in getirdiği güvenlik zorlukları ile baş etmek için gerekli güvenlik önlemleri ile donatılması gerekecektir. Dahası bu çözümler, benzersiz zorlukları bir mikro denetleyici vasıtasıyla, mümkün olan en az enerji tüketen sınırlı bir işlem gücü ve bellekle ele almak zorundadır.

Tez çalışmasında yeni tanıtılan IETF 6TiSCH protokolünün güvenli önyükleme mekanizmasına bir uzantı sunulmaktadır; burada, kimlik doğrulama anahtarları, etkili bir kimlik doğrulama ve önyükleme işlemi sağlamak için IoT ağının güvenilir düğümlerine dağıtılmıştır. Geliştirilen mekanizmanın sonuçları incelenerek çıkarımlar yapılmıştır. Test sonuçları, önerilen mekanizmanın 6TiSCH protokolünün standart olarak sunduğu mekanizmadan daha iyi olduğunu göstermektedir.

2. NESNELERİN İNTERNETİ İÇİN PROTOKOL YIĞINI

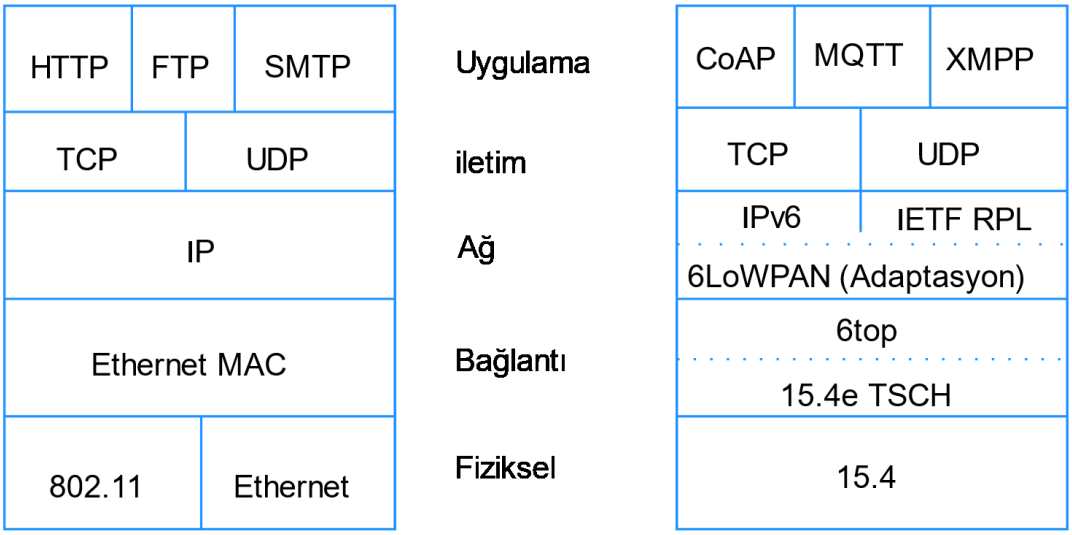
Şekil 3'te Nesnelerin İnternetini gösteren örnek bir ağ modeli verilmiştir. Bu ağda cihazlar (C) birbirleriyle ve sınır yönlendiriciyle (SY) iletişim kurabilmektedir [8]. Burada cihazlar hem veri üretip hem de bu verileri işleyerek sınır yönlendiriciye iletirler. Sınır yönlendirici de cihazlardan aldığı bu verileri "İnternet" ortamına aktarır. Bu işlemleri gerçekleştirmek için TCP/IP protokol yığınının var olan katmanlarında bazı değişiklikler yapılması gerekmektedir.



Şekil 3. Nesnelerin İnterneti Ağı

IEEE ve IETF tarafından Nesnelerin İnternetini gerçeklemeye yönelik bazı protokol eklentileri Şekil 4'te verilmiştir. Şekil 4'te IEEE 802.15.4-e TSCH [12], 6LoWPAN [13], 6Top, CoAP [14] gibi protokol eklentilerinin 6LoWPAN tabanlı protokol yığınındaki yerleri gösterilmiştir. Her bir protokol, kısıtlı kaynaklara sahip cihazların düşük güç tüketimi altında çalışmaları için uyarlanmıştır. Ortam erişim katmanında bulunan TSCH protokolü, yüksek kararlılık ve düşük enerji ile veri paketlerini iletmeyi hedeflemektedir. TSCH kullanan

düğümlerin kaynak ihtiyaçlarını dağıtık olarak karşılamayı hedefleyen 6Top, adaptasyon katmanın hemen altında yer almaktadır. IPv6 ile IEEE 802.15.4 arasındaki uyum işlemlerini 6LoWPAN katmanı gerçekleştirmektedir. RPL protokolü düşük kontrol trafiği gerektiren bir yönlendirme protokolüdür [14]. CoAP ise farklı senaryolar için uygulama protokolü ihtiyacını karşılamak amacıyla tasarlanmıştır. Bu eklentilerin hedefi, düşük güçlü ve kayıplı ağların kararlı bir şekilde İnternet erişiminin gerçekleştirilmesidir.



Şekil 4. Standart TCP/IP Modeli Ve 6LoWPAN Tabanlı Protokol Yığını

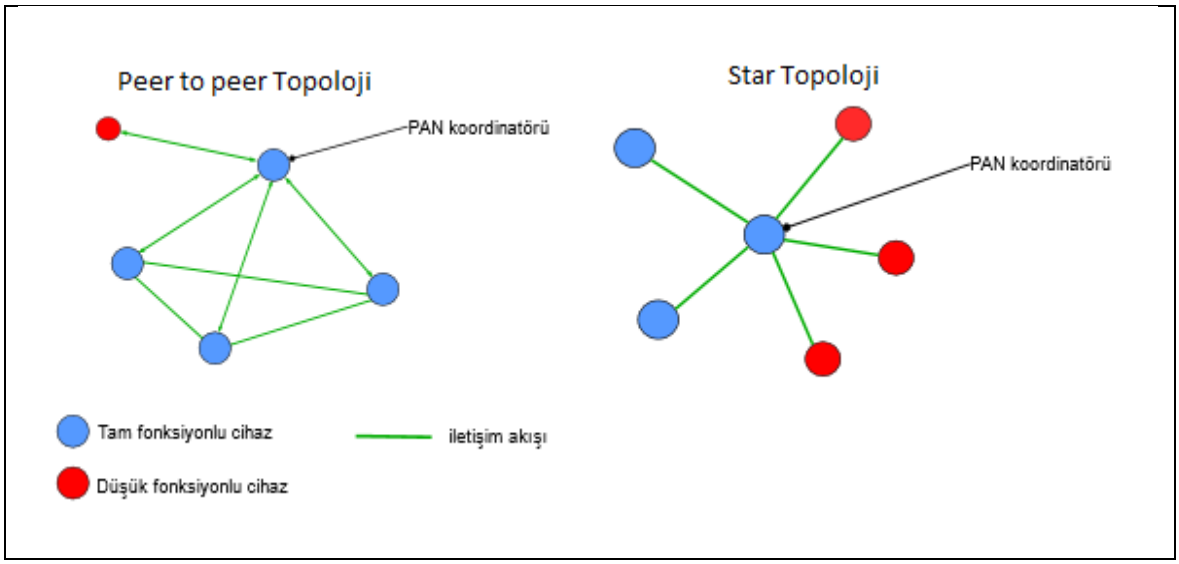
2.1. IEEE 802.15.4 Standardı

IEEE 802.15.4 kablosuz kişisel alan ağı (WPAN) standardıdır ve kısa mesafelerde bilgi aktarmak için kullanılmaktadır [15]. Kablosuz yerel alan ağlarından (WLAN) farklı olarak, WPAN'lardaki bağlantılar çok az altyapı içerir veya hiç içermez. Bu özellik, geniş kapsamlı cihazlar için güç tasarruflu çözümlerin uygulanmasını gerektirmektedir. Standard, cihazlar arası iletişim için fiziksel ve Ortam Erişim Kontrolü (MAC) katmanlarını tanımlamaktadır. Ağdaki maksimum veri iletim hızı 250 kbit/saniye olduğundan düşük güç tüketimi sağlanmış olur. IEEE 802.15.4, 2.4 GHz frekans bandında çalışan başka bir kablosuz kişisel alan ağı standardı olan Bluetooth'a benzerlik göstermektedir. Bluetooth kısa mesafeli kablolamayı ortadan kaldırmaya yöneliktir; IEEE 802.15.4 ise büyük ölçekli

otomasyon ve uzaktan kontrol için daha fazlasını hedeflemektedir. Dolayısıyla kullanım alanları ve amaçları farklıdır.

IEEE 802.15.4 standardı iki farklı cihaz türü tanımlar. Bunlar; Tam Fonksiyonlu Cihazlar (FFD-Full Function Device) ve Düşük Fonksiyonlu Cihazlar (RFD-Reduced Function Device)'dır. Tam fonksiyonlu cihazlar, herhangi bir topolojide bulunabilir, ağ koordinatörlüğü (Personal Area Network (PAN)) yapabilir ve herhangi bir cihazla haberleşebilir. RFD, nem veya sıcaklık algılama gibi düşük karmaşıklığa sahip uygulamalar için tasarlanmıştır; bu cihazlar büyük miktarda veri göndermek zorunda değildirler ve aynı anda tek bir FFD ile ilişkilendirebilirler. Dolayısı ile çok sekmeli topolojilerde yaprak (leaf) düğüm olarak yer alabilirler. 802.15.4. standardı CSMA/CA ortam erişimi mekanizmasını kullanır ve yıldız, eşler arası (peer to peer) gibi farklı topolojilerini destekler.

Şekil 5'te gösterilen yıldız topolojisinde, cihazlar ve PAN koordinatörü adı verilen tek bir merkezi kontrol birimi arasında iletişim kurulur. Cihaz ağ iletişimi için başlatma ya da sonlandırma noktası olarak görev yapmaktadır. PAN koordinatörü ise ağdaki iletişimi başlatmak, sonlandırmak veya yönlendirmek için kullanılabilir. Eşler arası topolojide yıldız topolojisine benzer olarak bir PAN koordinatörü vardır ancak bir cihaz kapsama aralığındaki diğer tüm cihazlarla iletişim kurabildiğinden yıldız topolojisinden farklıdır. Eşler arası topoloji, örgü ağ topolojisi gibi daha karmaşık oluşumlara izin vermekle birlikte kendi kendine organize olup, kendini onaran ağlar olarak karşımıza çıkar. Ayrıca eşler arası topoloji, verileri bir aygıttan ağdaki başka bir aygıta yönlendirmek için birden çok sekmeye izin verir. Her iki topoloji ağında çalışan tüm aygıtların benzersiz 64 bit genişletilmiş adresleri vardır. Bu adresler PAN içinde doğrudan haberleşme için kullanılır veya PAN koordinatörünün izni ile 16 bitlik kısa adresler ile değiştirilir. Bu cihazların oluşturduğu ağlarda iki tür adresleme mevcuttur. İlki 16 bitlik kullanıcı tarafından değiştirilebilen kısa adreslemedir. Bu adresleme kullanıldığında aynı kablosuz kişisel alan ağında aynı kısa adresli iki düğüm bulunmamasına dikkat edilmelidir. İkinci adresleme türü ise dünyada eşsiz bir değere sahip olan genişletilmiş 64 bitlik adrestir. Bu adres IEEE tarafından ağ cihazları üreticilerine dağıtılan 24 bit ve 40 bitlik üretici firma bilgisinden oluşmaktadır. Üretici tarafından atanan bu eşsiz adrese uzun adres denmektedir. Uzun adresleme her ağ ve her düğümde sorunsuzca kullanılmaktadır.



Şekil 5. Star ve Peer to Peer Topoloji Örnekleri

Fiziksel katman radyo alıcı vericisinin etkinleştirilmesi/devre dışı bırakılması, bağlantı kalitesi göstergesi, frekans seçimi, taşıyıcı frekansının oluşumu, sinyal algılama, modülasyon, gönderim ve alım işlemlerinin yürütüldüğü katmandır. Bu katman güç tüketimini doğrudan etkilediği için kablosuz duyarga düğüm tasarımında ayrı bir öneme sahiptir. Seçilen modülasyon tekniği, iletim hızı, gönderme gücü ve görev çevrim süresi gibi güç tüketimini etkileyen faktörler fiziksel katman tasarımı ile ilgili parametrelerdir. Ortam erişim kontrolü (MAC), kablosuz iletim ortamının düğümler arasında etkin bir şekilde paylaştırılmasını sağlayan katmandır. Ayrıca veri paketlerinin parçalanması, kanal erişimi, çerçeve doğrulama, hata düzeltme, hareket yönetimi, güç koruma ve şifreleme gibi işlemleri de kapsamaktadır.

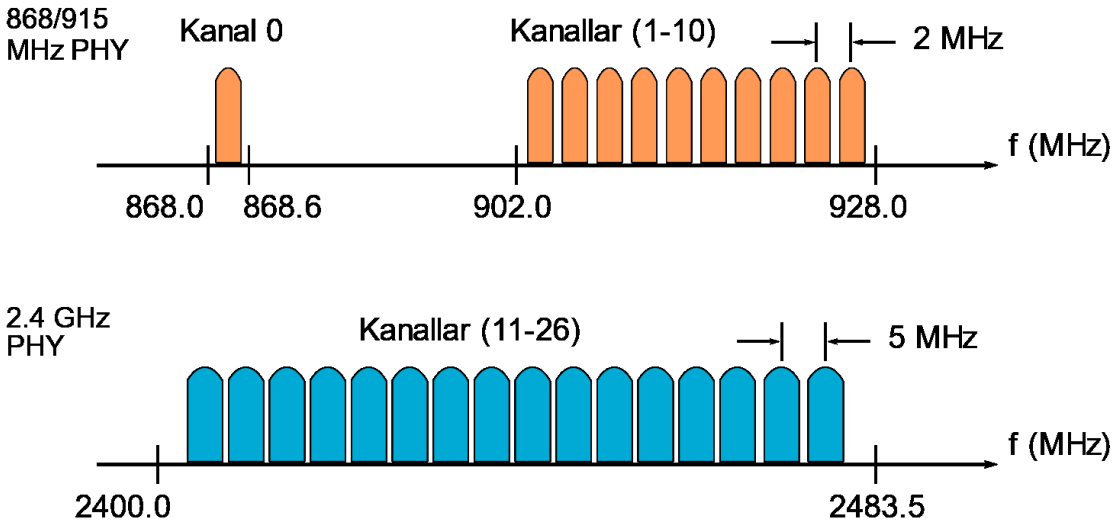
IEEE 802.15.4 standardı, üç farklı frekans bandı tanımlayarak farklı kullanım alanlarına hizmet etmektedir. Tablo 1'de veri hızı, kanal sayısı ve bölge gibi genel özellikleri verilmiştir. Tabloda gösterilen frekans bantları, kablosuz haberleşmede herhangi bir lisansa gerek duymadan sadece belirli bir çıkış gücü sınırlamasına uyularak kullanılan frekans bantlarıdır. Bu frekanslardan 2.4 GHz, dünyada çoğu ülkede lisansız olduğu için daha çok tercih edilir. Fakat bu sertifikasız frekanslarda çalışan cihazlar girişim sinyaline açıktır. Frekans bantlarındaki bazı kanallar, dağılım içerisinde yeniden yapılanmayı mümkün kılmaktadır. Standardın getirmiş olduğu bu özellik ile dinamik kanal seçimi, hedef enerji tespiti, hizmet kalitesi gibi olgular beacon ile kontrol edilebilmektedir. Düşük yayılım

kayıplarına sahip olduğundan düşük frekanslar kullanılarak uzak mesafelere veri iletimi gerçekleştirilebilir [16].

Tablo 1. IEEE 802.15.4 Radyo özellikleri

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

Ayrıca dikkat edilmesi gereken noktalardan biri de Şekil 6'da gösterilen her frekans bandının kendine özgü parametreleri bulundurmasıdır. Bu parametreleri taşıyıcı frekansı, iletim gücü, iletim görev devri, en yüksek iletim zamanı, en düşük iletim zamanı, anten özellikleri olarak sıralamak mümkündür. Fiziksel adresin kullanımı sayesinde cihazların aynı fiziksel bağlantıyı kullanmalarına ve birbirlerinden farklı şekilde tanımlanmalarına yardımcı olur.

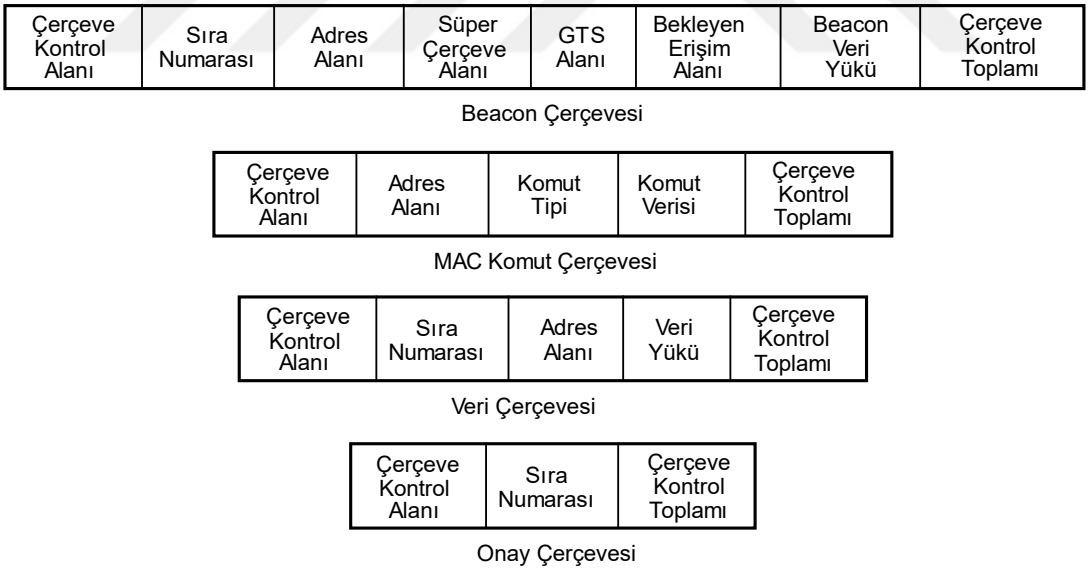


Şekil 6. IEEE 802.15.4 standardı için frekans bantları

MAC katmanı 2 adet servis hizmeti sunmaktadır. Bunlardan biri MAC veri servisi ve diğeri MAC yönetim servisidir. MAC veri servisi fiziksel katmandan gelen verilerin alım ve iletim işlemlerini gerçekleştirir. Yönetim servisi de ağ katmanı ile bağlantı katmanı arasında

yönetim komutlarının gerçekleştirilmesine izin vermektedir. Katmanlar arası veri transferindeki ilişkileri belirleyerek topolojinin etkili bir şekilde hizmet vermesini sağlamaktadır. Buna ek olarak, MAC katmanı, uygulamaya uygun güvenlik mekanizmalarının kullanımına izin vermektedir.

Ortam Erişim katmanı **beacon**, **veri**, **onay**, **MAC komut** çerçeveleri olmak üzere 4 mesaj formatı tanımlamaktadır. Beacon çerçevesi ağdaki aygıtların saatlerinin ayarlanarak senkronize çalışmaları için kullanılmaktadır. Veri çerçevesi veri göndermek için kullanılmaktadır. Onay çerçevesi iletilen mesajın başarılı bir şekilde alındığını belirtmek için kullanılmaktadır. Komut çerçevesi ise MAC komutlarının transferlerini gerçekleştirmek için kullanılmaktadır. Ortam erişim katmanında kullanılan çerçevelerin yapısı Şekil 7'de gösterilmektedir. MAC çerçevelerinde çerçeve kontrol alanı uzunluğu 16 bit, sıra numarası uzunluğu 8 bit ve çerçeve kontrol toplamı 16 bit uzunluğundadır ve bu alanların uzunlukları değişmemektedir. Adres alanı uzunluğu beacon çerçevesinde 32'den 80 bite kadar, veri ve MAC komut çerçevesinde 32'den 160 bit uzunluğuna kadar olabilmektedir.



Şekil 7. MAC katmanı çerçeveleri

IEEE 802.15.4, başarılı veri iletimi olasılığını arttırmak için çeşitli mekanizmalar kullanmaktadır. Bu mekanizmalar, CSMA-CA mekanizması, çerçeve onayı ve veri doğrulamadır. IEEE 802.15.4 LR-WPAN (Low-Rate Wireless Personal Area Network-Düşük Hızlı Kablosuz Kişisel Alan Ağı), ağ yapılandırmasına bağlı olarak beacon ve non-

beacon olmak üzere iki tür kanal erişim mekanizması kullanır. Non-beacon özellikli ağlarda, cihaz veri çerçeveleri veya MAC komutlarını her iletmek istediğinde, rastgele bir süre bekler. Ağdaki tüm cihazlara kanal boş olduğu zaman paket gönderme yetkisi verilir. Kanalın rastgele bekleme süresi içinde meşgul olduğu tespit edilirse, cihaz kanala tekrar erişmeye çalışmadan önce başka bir rastgele süre beklemektedir. Onay çerçeveleri CSMA-CA mekanizması kullanılmadan gönderilir. Beacon özellikli ağlarda ise düğümlere önceden tanımlanmış belirli zaman dilimlerinde paket gönderme yetkisi verilir. Kanalın meşgul olması durumunda üstel bir zaman dilimi beklenerek kanalın boşta olması beklenir. Kanal boşta kalırsa, cihaz bir sonraki mevcut zaman diliminde yayın (beacon) göndermeye başlar. Koordinatör, belirli aralıklarla süper çerçeve (superframe) adı verilen beacon mesajı yollar ve ağdaki tüm düğümler bu çerçeve ile senkronize olurlar. Her bir düğüm süper çerçevede belirtilen zaman diliminde paket alır ve gönderir. Onay ve beacon çerçeveleri CSMA-CA mekanizması kullanılmadan gönderilmektedir.

Veri veya MAC komut çerçevesinin başarılı bir şekilde alınması ve doğrulanması, isteğe bağlı olarak bir onay ile kontrol edilebilir. Hedef cihaz, alınan veri çerçevesini herhangi bir nedenle işleyemiyorsa, mesaj onaylanmaz. Gönderen cihaz, bir süre sonra bildirim almıyorsa, iletimin başarısız olduğunu varsayar ve veri iletimini yeniden başlatır. Birkaç denemeden sonra hala bir onay mesajı alınmıyorsa, gönderen işlemi sonlandırmayı veya tekrar denemeyi seçebilir. Onay kabul edilmediği durumlarda, gönderici iletimin her durumda başarılı olduğunu varsaymaktadır.

Bit hatalarını tespit etmek için 16 bitlik çevrimsel artıklık kontrolü (CRC) kullanan bir FCS mekanizması kullanılır. Bu standardı kullanan birçok uygulama, pille çalışmaktadır ve nispeten kısa aralıklarla pil değişimine neden olmaktadır. Bu nedenle, güç tüketimi önemli bir problemdir. Bununla birlikte, standardın fiziksel olarak uygulanması, bu standardın kapsamı dışındaki ilave güç yönetimi ihtiyacını ortaya çıkarmaktadır. Pille çalışan aygıtlar, güç tüketimini azaltmak için görev döngüsü kullanmaktadır. Bu cihazlar, çalışma ömürlerinin çoğunu uyku modunda geçirir; bununla birlikte, her cihaz bir mesajın beklemede olup olmadığını belirlemek için periyodik olarak radyo kanalını dinler. Bu mekanizma, uygulama tasarımcısının pil tüketimi ve mesaj gecikmesi arasındaki denge üzerinde karar vermesine olanak tanır.

Güvenlik açısından bakıldığında, kablosuz duyarga ağlar, diğer herhangi bir kablosuz ağdan farklı değildir. Bu ağların doğası ve maliyet hedefleri, güvenlik gereksinimlerini optimum bir şekilde karşılamak için ek güvenlik tedbirleri almaktadır. Cihazların bilgi işlem

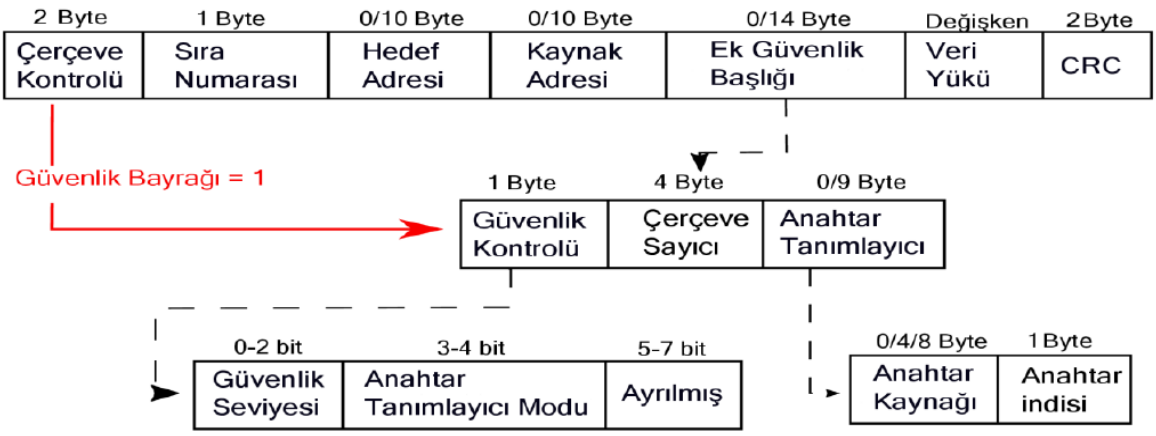
gücü, kullanılabilir depolama alanı ve güç tüketimi açısından sınırlı yeteneklere sahip olması etkili güvenlik mekanizmalarının uygulanmasını zorlaştırmaktadır. Bu kısıtlamalar, kriptografik algoritmaların ve protokollerin seçimini ciddi ölçüde sınırlamaktadır. Buna ek olarak, pil ömrü ve maliyet kısıtlamaları, bu ağların tolere edebileceği güvenlik yüküne ciddi limitler koymaktadır. Bu güvenlik mimari öğelerinin çoğu üst katmanlarda uygulanabilmektedir. IEEE 802.15.4 standardının şifreleme mekanizması, simetrik anahtar tabanlıdır ve ortak paylaşılan anahtarları kullanır. Mekanizma, şifrelemenin uygulanmasını ve anahtar değerinin güvenli bir biçimde depolanmasını varsayar.

Ortam erişim katmanı, iletilen paketlerin güvenlik hizmetlerini sağlamaktan sorumludur. Sunulan güvenlik hizmeti Şekil 4'te gösterilen protokol yığınında kullanılan diğer güvenlik mekanizmalarına yardımcı olmaktadır. IEEE 802.15.4. standardı erişim kontrolü, mesaj bütünlüğü ve mesaj gizliliğini desteklemektedir. Erişim kontrolü ile ağa izinsiz katılım engellenirken mesaj bütünlüğü sağlanarak verilerin orijinalliği korunur. Paket gizliliği ise istenmeyen cihazların veriye erişimini kısıtlamaktadır. Bu standarda sahip cihazlar uygulamanın özelliğine göre gerekli parametreleri ayarlayarak uygun güvenlik hizmeti alabilmektedir. Eğer uygulama herhangi bir güvenlik parametresi belirlemezse, güvenlik mekanizması devre dışı kalmaktadır. Bu sebeple uygulama tarafından istenilen güvenlik seviyesi mutlaka seçili olmalıdır. Güvenlik seviyesi iletilen veri paketinin başlık kısmında tutulan değer ile sağlanır. Her güvenlik seviyesi, farklı bir güvenlik özelliği ve farklı paket formatı sunar. 802.15.4 tanımlamasında Tablo 2'de görülen 8 farklı güvenlik durumu mevcuttur. Güvenlik modu aktif olduğunda IEEE 802.15.4 bağlantı katmanı veri paketi Şekil 8'deki gibidir. Ek güvenlik başlığındaki ilgili alanlar doldurularak tablodaki güvenlik türlerinden biri seçilmektedir.

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

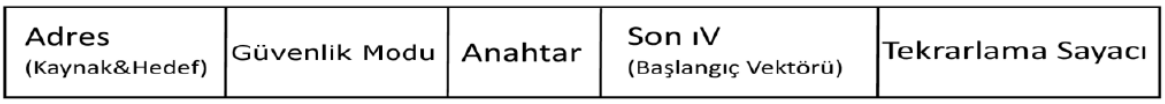
Güvenlik Modu	Tanımlama
NULL	Güvenlik yok
AES-CTR	Sadece şifreleme, CTR Modu
AES-CBC-MAC-128	128 bit MAC
AES-CBC-MAC-64	64 bit MAC
AES-CBC-MAC-32	32 bit MAC
AES-CCM-(32/64/128)	Şifreleme ve (32/64/128) bit MAC

Cihaz iletişim kuracağı hedef adrese göre kendi güvenlik seviyesini oluşturur. 802.15.4 radyo çipleri hangi hedef ile hangi koşulları gerektiren güvenli haberleşmenin sağlanacağına dair bilgileri kontrol eden erişim kontrol listesine sahiptir. Pasif aygıtlar 255 ACL (Erişim Kontrol Listesi) girişine kadar destek verebilirler.



Şekil 8. IEEE 802.15.4 veri ve kontrol alanları

Şekil 9'da görüldüğü gibi her giriş bir 802.15.4 adresi, güvenlik seviyesi tanımlayıcısı ve güvenlik materyalini içerir. Güvenlik materyalleri şifreli anahtar ve şifreleme/şifre çözme işlemlerinde direkt olarak kullanılmayan nonce[1] bilgilerinden oluşur. Nonce şifreleme/şifre çözmenin farklılığını sağlamak için kullanılır ve sadece gönderici ve alıcı tarafından bilinmesi gereken bir bilgidir.



Şekil 9. ACL girişi formatı

Gönderilen paketler için güvenlik talep edilmemiş ise paket olduğu gibi gönderilir. Hedef adresi ACL'de yer almıyorsa, yerine varsayılan olarak bulunan ACL girdisi kullanılır. Varsayılan ACL girdisi tüm hedef adresleri ile eşleşecek şekilde tasarlanmıştır. Eğer varsayılan ACL girdisi boş ise ve cihaz güvenlik gerektiriyorsa, bu durum hataya neden olur.

[1] Nonce: şifreleme sırasında kullanılan eşsiz bir sayıdır ve bu değer aynı şifreleme anahtarı boyunca tekrarlanamaz.

IEEE 802.15.4 Standardı tarafından desteklenen güvenlik seviyeleri NULL, AES-CTR, AES-CBC-MAC ve AES-CCM olmak üzere dört çeşittir.

NULL: Tüm radyo IEEE 802.15.4 donanımlarında bulunmak zorundadır. Kimlik fonksiyonları gibi herhangi bir güvenlik unsuruna ve yönetimine sahip değildir. Herhangi bir güvenlik garantisi sağlamaz.

AES-CTR: Bu güvenlik seviyesi AES blok anahtar ve sayaç değeri kullanarak gizlilik ve güvenliği sağlamaktadır [17]. Sayaç modu altında veriyi şifrelemek için gönderici, şifrelenmemiş veri paketini 16 baytlık bloklara ayırır ve 128-bitlik eşsiz bir anahtarla şifreler. Şifreleme sırasından her blok için bir arttırılan sayaç kullanılır. Alıcı ise şifrelenmiş veriyi kullanılan ortak anahtar yardımıyla çözer. Alıcı 16 baytlık blokları yeniden yapılandırabilmek için sayaç değerine ihtiyaç duyar. Nonce (rastgele değer) veya başlangıç vektörü (IV - Initial Vector) olarak bilinen değer; bayrak alanları, gönderici adresi ve üç farklı sayaç değerleriyle birlikte blok anahtarlama sistemine girdi olarak aktarılır.

AES CBC-MAC: Cihazlar CBC-MAC güvenlik modunu kullanarak bütünlük koruması sağlar [61]. Bu yöntemde veri, ardışık olarak bir anahtar değeri ile şifrelenmektedir. Şifrelenmiş verinin son bloğunun 4, 8 veya 16 baytı doğrulama kodu olarak seçilip gönderilecek paketin sonuna eklenmektedir.

AES-CCM: Bu güvenlik modu doğrulama ve şifreleme için CCM modunu kullanır. Genel olarak CBC-MAC kullanarak başlık ve veri yükü üzerinden bütünlük koruması uygular. Sonrasında faydalı veriyi ve MAC'i AES-CTR modu kullanarak şifreler. Yapılacak işlemlerde MAC, çerçeve ve anahtar sayaçları alanları kullanılmaktadır. Alıcı herhangi bir güvenlik durumunu kullanmak istediğinde tekrarlama korumasını devreye sokar. Bu durum AES-CTR ve tüm AES-CCM çeşitlerini kapsar. Alıcılar, veri paketine gömülü olarak alınan 5 bayt uzunluğundaki çerçeve ve anahtar sayaçlarını tekrarlama sayacı olarak kullanır. Alıcı, tekrarlama sayacını gelen paketin ACL girişinde tutulan en yüksek değer ile karşılaştırır. Eğer gelen paket daha önce alınmış son pakete ait sayaçtan büyük bir tekrarlama sayacına sahip ise kabul edilir ve paketle gelen sayaç yeni tekrarlama sayacı olarak saklanır. Aksi durumda paket kabul edilmez.

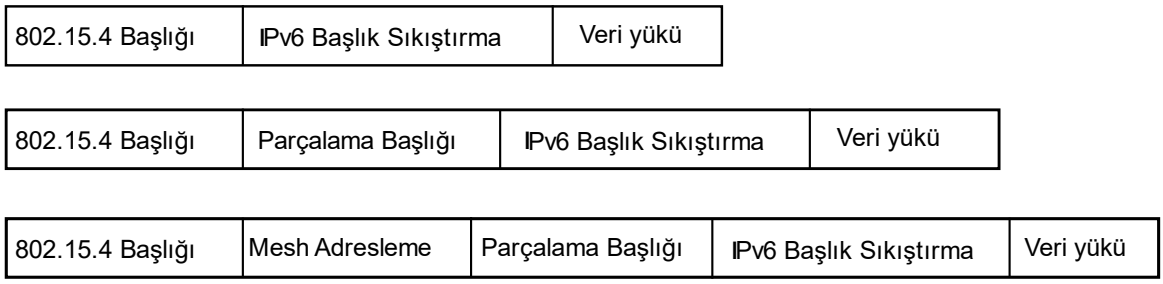
2.2. 6LoWPAN

6loWPAN, düşük güçlü Kablosuz Kişisel Alan Ağlarında IEEE 802.15.4 fiziksel katmanına sahip cihazların IPv6 adresleme standardına uyumlu hale getirmek amacıyla oluşturulmuştur. IEEE802.15.4 standardı küçük paketler ürettiği için IPv6 kullanması uygun değildir, bu nedenle bazı adaptasyon işlemleri gereklidir. IEEE 802.15.4 standardında [18] fiziksel katman için tanımlanan çerçeve boyutu 127 bayttır: 102 bayt MAC katmanına ve 25 baytta maksimum başlık boyutuna bırakılmıştır. Şekil 4'de gösterilen OSI modelinin referansı ile 6LoWPAN, IEEE 802.15.4 üzerinden IPv6 paketlerini iletmektir.

6LoWPAN çalışma grubu, aşağıda verilen özellikler sayesinde IP paketlerinin düşük güçlü ağlarda sorunsuzca taşınmasını sağlamaktadır.

- IP ve TCP/UDP/ICMP başlık (header) sıkıştırma.
- IPv6 paketlerinin parçalara ayrılması / yeniden birleştirilmesi.
- Komşu keşfetme yöntemi ile otomatik ağ konfigürasyonu.
- Örgü ağ adresleme.
- 16 (kısa) ve 64 (uzun) bitlik adreslemeyi desteklemektedir.

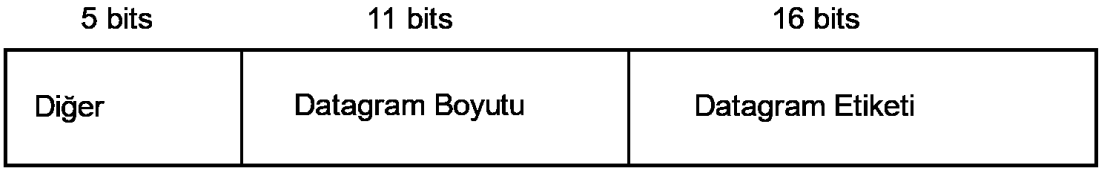
AES-CCM-128 nedeniyle maksimum 21 baytlık ilave bir paket yükü düşünüldüğünde, maksimum veri boyutu 81 bayta inmektedir. Aynı zamanda, IPv6 standardının asgari paket boyutu 1280 bayttır ve bunların arasında 40 baytlık bir başlık bulunmaktadır. Buna ek olarak UDP başlıkları için 8 ve TCP başlıları için 20 bayt yer ayrılmıştır. Bu durum, üst katmanlarda 20 ve 33 bayt arasında veri taşınmasına izin vermektedir [19]. Bu problem göz önüne alındığında, 6loWPAN adaptasyon katmanı, Şekil 10'da olduğu gibi IPv6 ile sıkıştırılmış çerçeve yapısını tanımlamaktadır. Çerçevenin her başlık alanı, bir başlık türü ve bunu izleyen sıfır veya örgü (mesh) adresleme, parçalanma, başlık sıkıştırması veri yükü (payload) seçenekleri gibi başlık alanı içermektedir. Mesh adresleme bağlantı katmanı yönlendirmeyi ve parçalama ise IPv6 minimum MTU gereksinimini desteklemektedir. 6LoWPAN, her başlığın başlangıcına yerleştirilen bir başlık alanı kullanarak tüm başlık (üstbilgi) formatlarını tanımlamaktadır. Belirtilen paket formatının ayrıştırılması kolaydır ve kullanılmasına ihtiyaç duyulmayan alanlar çerçeveden çıkarılabilmektedir.



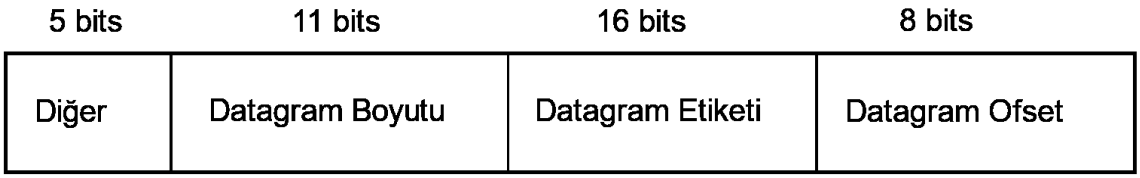
Şekil 10. 6LoWPAN Başlık Yığınları

2.2.1. Parçalama

6LoWPAN adaptasyon katmanının birincil görevi parçalama ve birleştirme işlemleridir. Parçalama başlığı, veri yükünün IEEE 802.15.4 çerçevesine sığmayacak kadar büyük olduğunda kullanılmaktadır. Açıklanan bu başlık, datagram boyutu, datagram etiketi ve datagram ofseti olmak üzere üçe ayrılmaktadır. Datagram boyutu, parçalanmamış veri yükünün toplam boyutunu tanımlar ve alıcıda bellek atamasını gerçekleştirmek için her parçaya eklenmektedir. Datagram etiketi belirli bir yüke karşılık gelen verileri tanımlamaktadır ve aynı verinin parçalarını eşleştirmek için kullanılır. Ofset alanı yalnızca ikinci ve sonraki bağlantı parçalarında bulunur; veri yükünün başlangıcından itibaren parçanın sekizli katları halinde ofset belirtmesi gerekmektedir. Şekil 11'deki birinci parça, datagram boyutunu (11 bit) ve datagram etiketini (16 bit) içeren bir başlık taşımaktadır. Şekil 12'deki sonraki parçalar ise datagram boyutu, datagram etiketi ve ofset (8 bit) içeren bir başlık taşır.



Şekil 11. İlk Parça

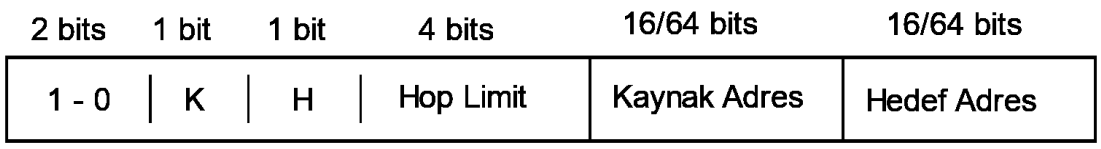


Şekil 12. Diğer Parçalar

6LoWPAN, IP katmanından aldığı parçalama eşiğinden büyük paketleri öncelikle küçük parçalara böler. 6LoWPAN parçalama, iletim ve birleştirme adımlarını 60 saniyede gerçekleştirerek iki düğüm arasındaki iletimi sağlamalıdır. Paket başlığı ilk parçacığa yerleştirilir ve hedef düğüme iletilmek üzere ortam erişim katmanına gönderilir. Diğer parçalar, parça başlığı ve sıra numarası bilgisiyle hedef düğüme yönlendirilir. Hedef düğüm aldığı parçaları sıraya koyar ve bu paketi IP katmanına yönlendirir. IP katmanı tarafında parçalama birleştirme işlemleri söz konusu değildir. Eğer haberleşme sırasında herhangi bir parça düşerse IP paketinin tamamı kaybolmuş sayılır.

2.2.2. Mesh Adresleme

Mesh Adresleme başlığı, çoklu radyo sekmeleri üzerinden 6LoWPAN verilerini iletmek ve bağlantı katmanı yönlendirmesini desteklemek için kullanılır. Bu alan sekme (hop) limiti, kaynak adresi ve hedef adresi olmak üzere üçe ayrılmaktadır. 4 bit olan hop limiti, paketi bir sonraki sekmeye göndermeden önce her yönlendirme düğümü tarafından azaltılmaktadır. Değeri sıfır olunca paket daha ileri yönlendirilmez ve paket düşürülür. Kaynak ve hedef adresleri, IP sekmesinin bitiş noktalarını belirtir. Her iki adres de IEEE 802.15.4 bağlantı adresidir.



Şekil 13. 6LoWPAN Mesh Adresleme Başlığı

Mesh Adresleme biçimi Şekil 13'te gösterilmiştir ve başlık türü sadece iki bittir. Üçüncü ve dördüncü bitler, hangi adresleme modunun kaynak ve hedef adresleri için

kullanılacağını belirtir. Bir bitlik Kaynak (K) ve Hedef (H) alanları 1 seçilirse kaynak ve hedef adresleri için IEEE 64-bit adresleri kullanılmaktadır. Aynı şekilde bu alanlar 0'a atanırsa adresler için 16 bitlik kısa adresler kullanılmaktadır. Diğer bitler hop limiti ve adresleme alanlarını taşır. Sonuçta Mesh Adresleme başlığı, kullanılan adresleme yöntemlerine bağlı olarak 5 ile 17 bayt arasında değişmektedir.

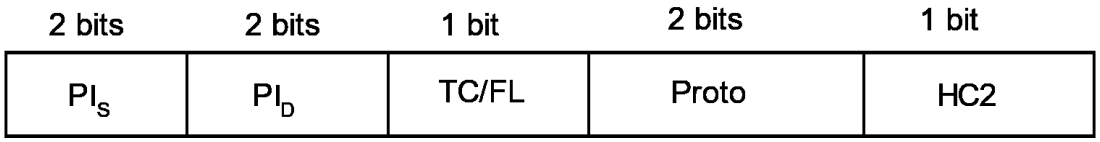
2.2.3. Başlık Sıkıştırma

Başlık sıkıştırması, 6loWPAN standardının en önemli özelliklerinden birisidir. Bu özellik IPv6 iletişimi üzerinden TCP/UDP paketlerinin düşük maliyetle iletilmesini mümkün kılmaktadır. İlk standart olan RFC 4944, HC1 ve HC2 olmak üzere iki farklı sıkıştırma seviyesi tanımlar [19]. Fakat önerilen bu yöntemler, 6loWPAN'da IPv6'nın kullanımları için yetersiz kabul edilmektedir [20].

HC1: Bu sıkıştırmanın temel fikri, IPv6 paketlerinin her zaman bazı alanlar için aynı değerleri almasıdır. Alan bazlı düşünüldüğünde aynı değerleri gösteren ifadeler aşağıdaki gibidir:

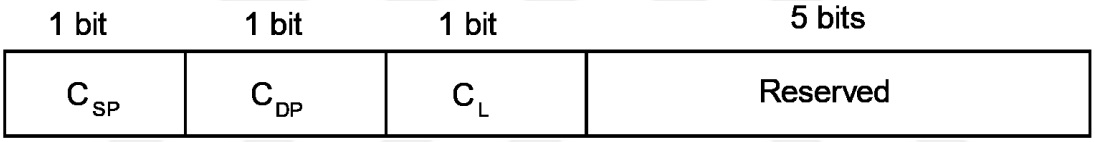
- IP sürümü her zaman 6'dır.
- Adresler MAC adreslerinden çözülebilir.
- Çerçeve uzunluğu, IEEE802.15.4 çerçeve veya boyut alanından çıkarılabilir (Şekil 10).
- Trafik Sınıfı ve Akış Etiketi sıfırdır.
- Bir sonraki başlık UDP / TCP veya ICMP'dir.

HC1 paket formatı Şekil 14'de gösterilmektedir. PI alanları, kaynak ve hedef adreslerinin nasıl taşınacağını belirtmektedir. **TC/FL** alanı Trafik Sınıfı ve Akış Etiketinin sıkıştırılmasını ifade eder; **Proto** alanı hangi protokolün kullanılacağını gösterir (TCP, UDP, ICMP). **HC2** alanı da HC1'den sonra HC2 başlığının kullanılması gerektiği durumları ifade etmektedir.



Şekil 14. 6loWPAN HC1 Yapısı

HC2: UDP ve HC2 bayrağı HC1 tarafından ayarlandığında, HC2 başlığının kullanımı aktif olmaktadır. HC2, UDP'nin paket boyutunu 8 bayttan 4 bayta düşürür. Şekil 15'te genel bir HC2 başlığı gösterilmektedir. C alanları, Kaynak Bağlantı Noktası (SP), Hedef Bağlantı Noktası (DP) ve Uzunluk (L) için sıkıştırmanın etkin olup olmadığını gösterir. Sonuncusu, veri yükü uzunluğunun IPv6 başlığının yük alma uzunluğundan türetildiği anlamına gelir. Bağlantı noktalarının sıkıştırılması, bir portun 4 bitlik kısa bir değer olarak gönderilmesini sağlar [20].



Şekil 15. 6loWPAN HC2 Yapısı

RFC 6282 [20] yukarıda bahsedilen sıkıştırma tekniklerine ek olarak iki farklı teknik sunmaktadır. Bu standart Lokal, Global ve Multicast IPv6 adreslerinin etkin şekilde sıkıştırılması için IPHC ve NHC olmak üzere iki sıkıştırma seviyesi sunmaktadır.

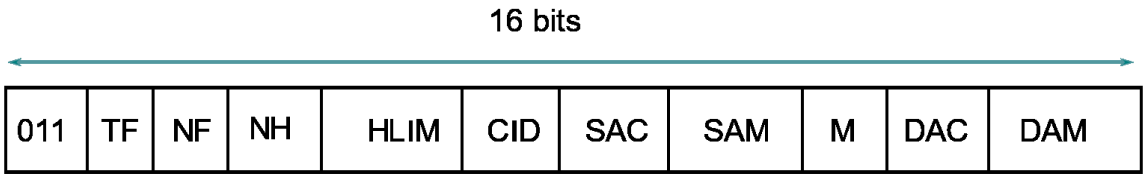
Tablo 3. 6loWPAN IPHC Alanındaki Kısaltmaların Açıklaması

Alan	Uzunluk	Açıklama
011	3	IPHC başlığının dispatch değerini temsil eder.
TF	2	Traffic Class and Flow Label/Trafik Sınıfı ve Akış Etiketi alanları için sıkıştırma seçeneklerini belirtir.
NH	1	Next Header compressed using NHC or not/Bir sonraki başlığın NHC kullanılarak kodlanıp kodlanmadığını belirtir.
HLIM	2	Hop Limit compression/Sekme sınırının nasıl sıkıştırıldığı hakkında bilgi veren bit dizisidir.

Tablo 3'ün devamı.

CID	1	Context Identifier Extension/Eğer bu bit 1 ise, DAM (Hedef Adres Modu) alanından sonra 8 bitlik bir CIE (İçerik Tanımlayıcı Uzantısı) alanı izler.
SAC	1	Source Address Compression/Sıkıştırma durumunu kontrol eden bittir.
SAM	2	Source Address Mode/Kaynak adresi sıkıştırma türünü belirlemek için SAC ile kullanılan bittir.
M	1	Multicast Compression/Hedef adresin multicast adres olup olmadığını belirleyen bittir.
DAC	1	Destination Address Compression/Sıkıştırma durumunu kontrol eden bittir.
DAM	1	Destination Address Mode/Kaynak adresi sıkıştırma türünü belirlemek için M ve DAC ile kullanılan bit dizisidir.

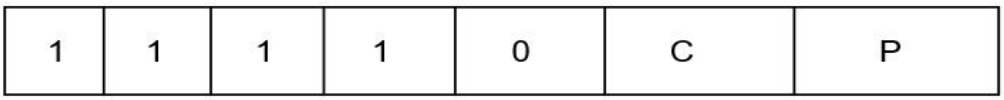
IPHC ve NHC: [20] 'de önerilen bu yeni yöntem, IPv6 başlığını sıkıştırır ve bir NHC baytı kullanarak farklı ardışık başlıklar için 6loWPAN çerçevesini genel yapar. NHC, rastgele sonraki başlıkları sıkıştırmak olarak da adlandırılmaktadır. Bu çerçeve, hangi başlığın takip edileceğini tanımlar ve sıkıştırılmış olup olmadığını gösterir. Şekil 16'da IPHC başlık yapısı verilmiştir. IPHC başlık yapısının alanları Tablo 3'te açıklandığı gibidir. TF bitleri, Trafik Sınıfı/Akış Etiketi alanlarının sıkıştırılıp sıkıştırılmadığını gösterir. HLIM bitleri, sekme sayısını, sıkıştırma durumunu veya bu alanın sırayla taşınıp taşınmadığını kontrol etmektedir. IPHC kodlamasının 8-15 bitleri, IPv6 Kaynak ve Hedef Adresleri için kullanılan sıkıştırma yöntemlerini gösterir. Bağlam Tanımlayıcı (CID) biti sıfır olduğunda, varsayılan bağlam kaynak ve/veya hedef adreslerini sıkıştırmak için kullanılabilir. Bu mod, genellikle hem kaynak hem de hedef adresleri aynı ağdaki düğümlere atandığında kullanılır. Kaynak Adres Sıkıştırması (SAC), sıkıştırmanın lokal veya global haberleşmeye göre şekilleneceğini göstermektedir. Kaynak Adresi Modu (SAM), kaynak adresinin sırayla mı yoksa 16/64 bitinin elendikten sonra iletilip iletilmeyeceğini veya adresin tamamının paketten çıkarılıp çıkarılmayacağını gösterir [20]. SAC değeri setlendiğinde ve kaynak adreslerinin öneki (prefix) paketten çıkarıldığında, tanımlanan bağlam (context information) bitleri kullanılarak adresler elde edilebilir. Çoklu dağıtım alanı (M) hedef adresin unicast veya multicast adresi olup olmadığını belirtir. Hedef adres unicast bir adresi işaret ettiğinde, DAC ve DAM bitleri SAC ve SAM bitlerine benzer. Aynı zamanda hedef multicast bir adres olduğunda, DAM bitleri farklı multicast sıkıştırma biçimlerini tanımlamaktadır.



Şekil 16. 6loWPAN IPHC Başlık Yapısı

Şekil 17'de NHC çerçevesi gösterilmiş ve Şekil 18'de UDP için sıkıştırılmış başlık yapısı sunulmuştur. C alanı paketi kontrol eden "checksum"'ı ifade etmektedir. C alanı, 0'a setlenirse kontrol değeri paketin içinde sırasıyla gönderilmektedir. 1 olması durumunda ise paketten çıkartılarak 6LoWPAN sonlandırma noktasından yeniden hesaplanabilmektedir. P, bağlantı noktalarının sıkıştırılması anlamına gelmektedir. İki bitlik alanın alacağı değere göre portlar 8 veya 12 bitlik sıralar halinde paketlerde iletilmektedir. Bu başlık sıkıştırma, HC1 ve HC2'den daha fazla esneklik sunar.

Şekil 17. 6LoWPAN NHC Yapısı

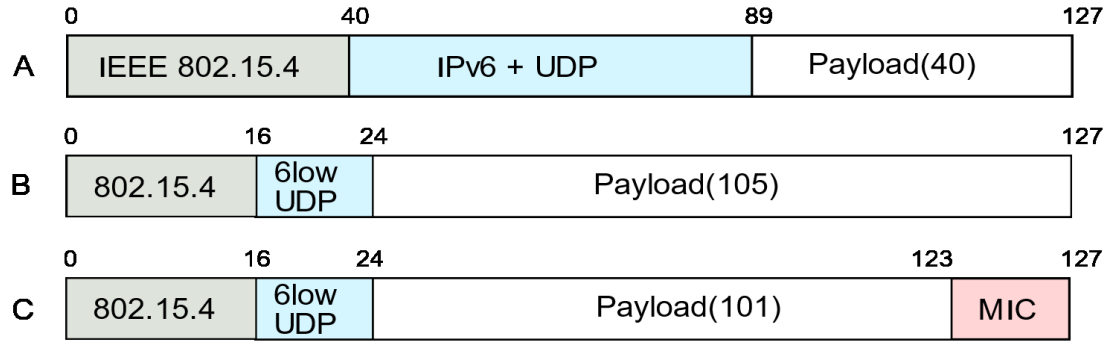


Şekil 18. 6loWPAN UDP Sıkıştırılmış Başlık Yapısı

RFC 6282 standardı, IPHC başlığı için 2-3 bayt ve her bir NHC için 1 bayt alan ayırmaktadır. IPHC'deki farklı alanlar, kaynak ve hedef adreslerinin boyutunu 16/64 bit adresleri olarak belirtilmesini veya tamamen ortadan kaldırılmasını sağlamaktadır. Bu yapı, adreslerin ortadan kaldırılmasıyla UDP/IPv6 paket başlıklarının uzunluğunu 7 bayta düşürür. Şekil 19'daki kırmızı alanların, ek IPv6 başlıkları, adresleri ve büyük bir çerçeve olması durumunda parçalanma gibi isteğe bağlı alanları temsil ettiği gösterilmektedir.

Şekil 19. IPHC ve NHC'yi Kullanan 6loWPAN Paketi (UDP)

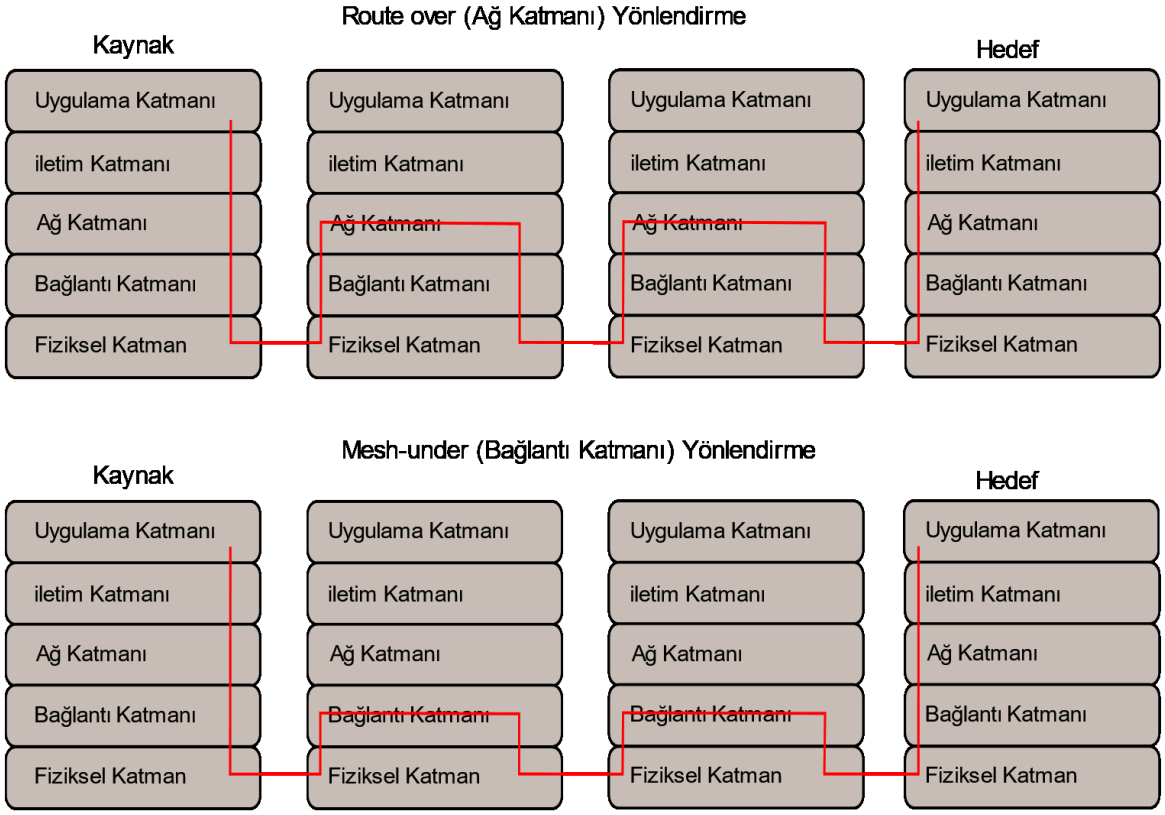
Şekil 20'de, IEEE 802.15.4 için 6lowPAN ve 64-bit adresleri kullanan broadcast yayınlarda veri yükü boyutunun maksimize edildiği çerçeve yapısı görülmektedir. A'da, IPv6 ve UDP başlıklarının normal kullanan ve B'de ise broadcast adresleri ve IPHC/NHC'yi kullanan 6loWPAN için sıkıştırma uygulanmış IEEE802.15.4. paketi gösterilmektedir. Son şekilde çerçevenin sonuna 4 baytlık AES-CCM-128 MIC eklenmiştir.



Şekil 20. IEEE802.15.4, 6loWPAN ve AES-CCM MIC için Çerçevelerin Karşılaştırılması

2.2.4. Yönlendirme

Yönlendirme, bir cihazdan başka bir cihaza, bazen birden fazla sekme kullanarak bir veri paketi gönderme becerisi olarak tanımlanabilir. Yönlendirme mekanizmasının hangi katmanda yapıldığına bağlı olarak, yönlendirme iki kategoriye ayrılmaktadır: Mesh under (bağlantı katmanı üzerinden) veya Route over (ağ katmanı üzerinden). Şekil 21'de Mesh-under veri paketlerini iletmek için bağlantı katmanı adresinin (IEEE 802.15.4 MAC veya kısa adres); Route-over ise ağ katmanı adresinin (IP adresleri) kullanımlarını göstermektedir.



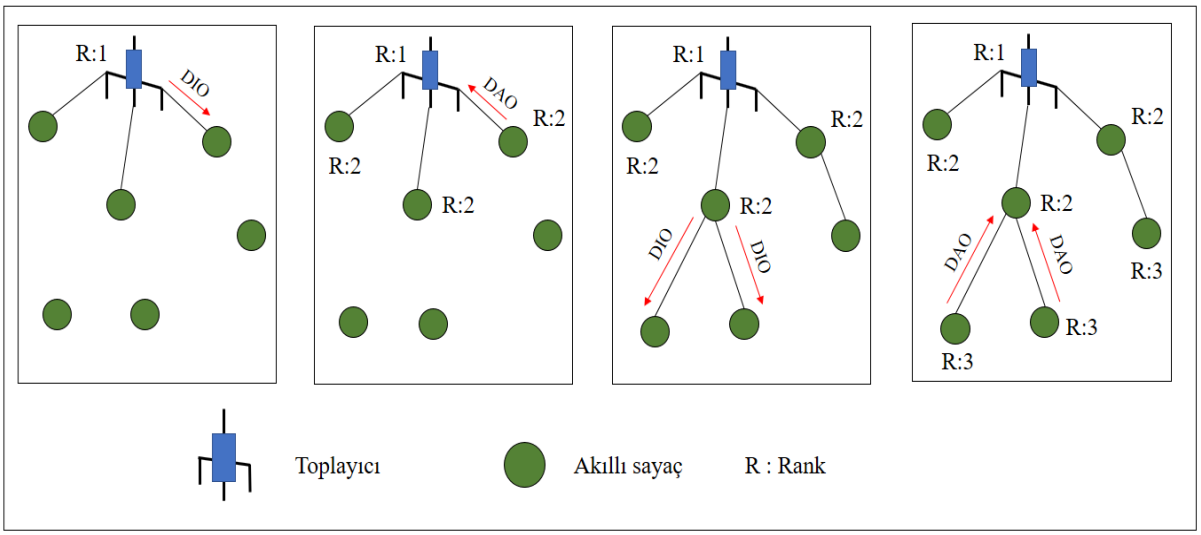
Şekil 21. Mesh-under ve Route-over paket yönlendirme

Mesh-under sistemde, verilerin yönlendirilmesi bağlantı katmanında gerçekleşir, dolayısıyla mesh ağlar IP alt ağı olarak kabul edilebilir. Böyle bir sistemdeki IP yönlendirici sınır yönlendiricidir. Kopya (eş) adres tespiti gibi sorunlarda IPv6 protokolleriyle uyumluluk sağlamak için bir yayın alanı (broadcast domain) oluşturulur. Yayın alanı tarafından iletilen mesajlar, ağdaki tüm cihazlara gönderilmelidir; bu da yüksek ağ yüküne neden olur. Mesh alt ağları, daha küçük ve yerel ağlar için uygundur [21].

Route-over ağlarında yönlendirme, yukarıda açıklandığı gibi IP düzeyinde gerçekleşir; bu nedenle bu tür ağlardaki her atlama, bir IP yönlendiricisini temsil eder. IP yönlendirmesinin kullanımı, daha güçlü ve ölçeklenebilir ağların temelini oluşturur. Her yönlendirici, IP yönlendiricisi tarafından desteklenen tüm özellikleri sağlamaktadır [21]. Her iki yaklaşım da avantaj ve dezavantajlara sahiptir. Mesh-under yaklaşımın en belirgin dezavantajı, IP ve uygulama katmanı performansının optimize edilmesini engelleyen ağ topolojisi ile IPv6 komşusu bulma protokolünün çok noktaya yayın hizmetinin sağlanmasıyla getirmiş olduğu yüksek maliyettir. Bu sınırlamalardan dolayı, genelde mevcut uygulamalarda Route-over iletişimi söz konusudur. Route-over yaklaşımı da bazı zorluklar

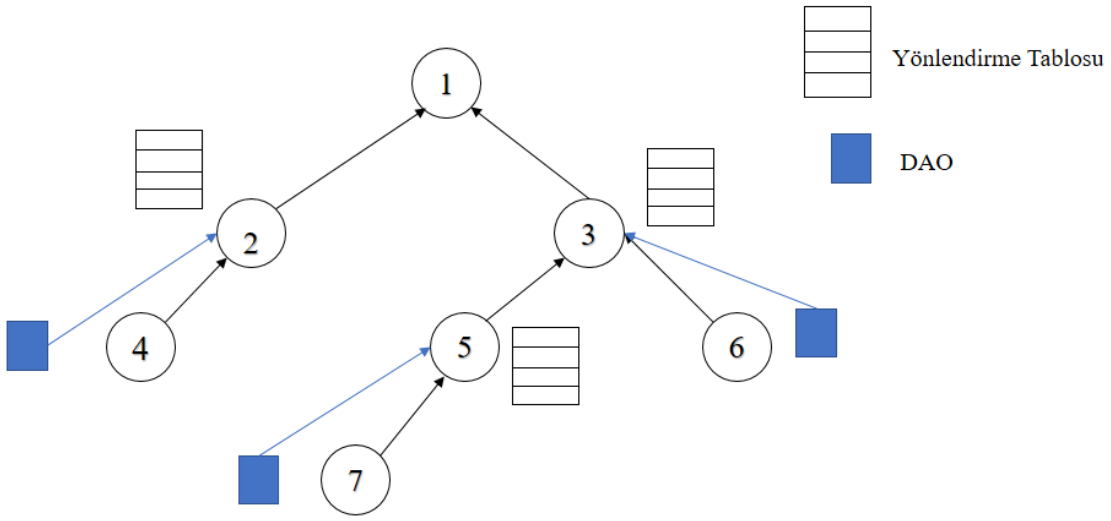
içermektedir. 6LoWPAN tek bir IPv6 alt ağıdır ve geleneksel önek tabanlı gönderim işe yaramaz. Ara yönlendiriciler, değişen kanal koşullarına, düğüm hareketliliğine veya uyku moduna bağlı olarak bağlantı kaybı yaşayabilir. Cihazların kısıtlı kaynaklara sahip olmasından dolayı yönlendirme protokolleri optimize edilerek tasarlanmalıdır.

Günümüzde 6LoWPAN ağları üzerinde popüler olarak kullanılan yönlendirme protokolü IETF tarafından tanımlanan RPL'dir [22]. RPL protokolü kablosuz duyarga ağları ile İnternet arasındaki veri trafiğini sağlamaktadır. Kablosuz duyargaların güç tüketim ihtiyaçları ve iletim kanallarının kayıplı olması gibi özellikler dikkate alınarak yönlendirme için yollar oluşturulur. Bu protokol, IPv6 tarafından oluşturulan bağlantıları kullanarak ağaç tabanlı bir topoloji oluşturur. Topolojide veri iletimi bir kök cihazdan uç cihaza ve tersi yönde olur. Optimum veri akış yolunun belirlenmesi için "Rank" adı verilen ve cihazların ağdaki yerini gösteren bir parametre kullanılır. Rank hesaplanmasında ise amaç fonksiyonu (objective function) ismi verilen ve seçilen bir parametreye göre yolun kalitesini hesaplayan algoritmalar kullanılır. Örneğin, hattın batarya durumu dikkate alınarak Rank hesabı yapılabilir. Şekil 22'de ağaç tabanlı topolojinin oluşturulmasında rol alan RPL protokolünün çalışma adımları özetlenmektedir.



Şekil 22. RPL protokolü

RPL protokolü, ağın oluşturulmasında temel öğe olan kök düğümün DODAG[2] Information Object (DIO) ismi verilen kontrol paketlerini ağa yaymasıyla başlar. Alt düğümler aldıkları DIO mesajını kullanarak rank bilgilerini hesaplarlar. Hesaplamada kullanılan tipik amaç fonksiyonu DIO mesajının geldiği düğümün (ata düğümün) rank değerine kanalın kalitesini ifade eden bir değer (bağlantı kalitesi) ekler ve bu hesaplanan değer alt düğümün rank değeri olarak belirlenir. Alt düğüm yayınlayacağı DIO mesajlarına hesaplanan rank değerini ekler. DIO mesajları Trickle [23] olarak adlandırılan bir algoritmayla değişken periyotlarla gönderilir. Ağın kararlı olduğu durumda DIO yayın periyodu maksimum değere çekilir. Ağ yapılandırması başladığında/değişiklik olduğunda DIO mesajı sık aralıklarla gönderilir. Seyrek gönderilen DIO yayın mesajları kontrol trafiğini azaltmış olur. Bu özellik, düşük güçlü ağlar için RPL'in uygun bir yönlendirme protokolü olmasını sağlar.

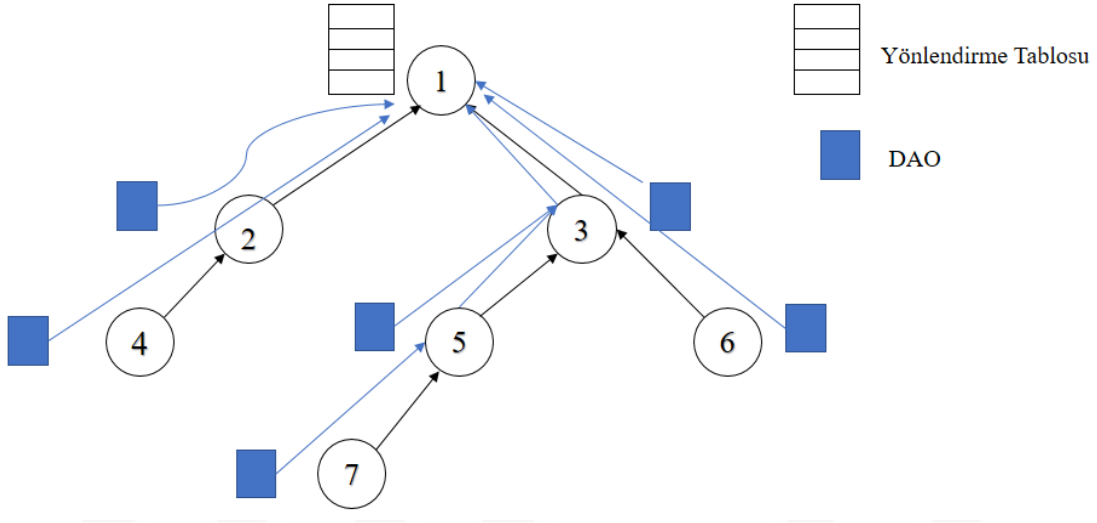


Şekil 23. RPL Storing Modu

DIO mesajı alındıktan sonra her düğüm uygun ranka sahip düğümü atası kabul eder ve bu düğümün IPv6 lokal adresini yönlendirici olarak yönlendirme tablosuna ekler. Düğüm, varsayılan yönlendiriciye ek olarak farklı düğümleri de yönlendirici olarak tabloya ekleyebilir. DIO mesajını alan düğüm kök düğüme doğru yönlendirme yolunu belirlemiş olur. DIO mesajına cevap olarak, alt düğüm Destination Advertisement Object (DAO) adı verilen bir mesaj oluşturur ve bu mesajı yönlendirici düğüme iletir. Ata düğüm alt düğümden

[2] DODAG: Hedef tabanlı yönlü döngüsüz graf.

aldığı DAO mesajını iki farklı şekilde işleyebilir. Depolama modu (storing mode) olarak adlandırılan birinci durumda, düğüm DAO mesajından elde ettiği yol bilgisini yönlendirme tablosuna ekler (Şekil 23). İkinci durumda DAO mesajındaki yönlendirme bilgileri, ara düğümler tarafından depolanmaz. Kök düğüm, DAO içerisindeki farklı düğümlere ait birleştirilmiş yol bilgisini lokal yönlendirme tablosuna ekler. Bu yönteme de depolanmayan mod (non-storing mode) denmektedir (Şekil 24).



Şekil 24. RPL Non-Storing Modu

RPL, DODAG Information Solicitation (DIS) adı verilen bir mesajdan yararlanarak ağa dâhil olma adımlarını kısaltabilmektedir. Düğümler DIS mesajını aldıktan sonra, DIO mesajıyla bu mesaja cevap verirler. Böylece ağa katılmak isteyen düğümler DIO periyodunu beklemeden hızlıca ağa katılabilirler.

Instance (Oluşum) adı verilen ve aynı ağ üzerinde farklı amaç fonksiyonları kullanılarak farklı topolojiler oluşturan yapılar da RPL tarafından desteklenmektedir [22]. Böylece topolojiler, uygulamanın gerektirdiği performans kriterlerine göre şekillendirilmiş olur. Örneğin, batarya optimizasyonu için oluşturulacak topoloji daha uzun sekme sayılarını içerebilir. Fakat bu topoloji, gecikmelerin minimum tutulması gereken mesajlar için uygun olmayacaktır. Bu eksikliği gidermek için aynı ağ içerisinde gecikmelerin minimuma indirildiği bir Oluşum (Instance) kullanılabilir [24].

2.2.5. Otomatik Yapılandırma ve Komşu Keşfi

Otomatik yapılandırma, bir aygıtın IPv6 adresinin otonom bir şekilde üretilmesidir. İşlem, IPv4 ve IPv6 arasında esasen farklıdır. Yapılandırma, IPv6'da bir cihazın IPv6 adresini DHCP sunucusu ya da benzeri herhangi bir dış etkileşim olmadan otomatik olarak oluşturmasına izin vermektedir. Adres almak için, cihaz komşu keşif protokolü üzerinden iletişim kurabilir (NDP-Neighbor Discovery Protocol), fakat NDP özelliklerinin çoğu RPL tarafından da desteklenmektedir. Burada açıklanan prosedür RPL için de geçerlidir ve dört mesaj türünü içerir:

- Yönlendirici talebi (RS-Router solicitation)
- Yönlendirici duyurusu (RA-Router advertisement)
- Komşu talebi (NS-Neighbor solicitation)
- Komşu duyurusu (NA-Neighbor advertisement)

IPv6 komşusu bulma (ND-Neighbor Discovery), cihazın komşuları keşfetmesini, erişilebilirlik bilgilerini sağlaması, varsayılan rotaları yapılandırmasını ve yapılandırma parametrelerini yayınlamasını sağlar. RS mesajı, diğer parametrelerin yanında, ağın IPv6 önekini (prefix) de içerir. Ağdaki tüm yönlendiriciler bu mesajları periyodik olarak gönderir. Bir cihaz 6LoWPAN ağına katılmak istiyorsa, kendisine bir bağlantı-yerel unicast adresi (FE80::IID[3]) atar. Daha sonra bir NS mesajı ile adresin başkası tarafından kullanılıp kullanılmadığını kontrol etmek için alt ağdaki diğer tüm katılımcılara bu adresi gönderir. Cihaz belirli bir zaman içerisinde herhangi NA mesajı duymazsa, adresin benzersiz olduğunu varsayar. Bu işleme yinelenen adres tespiti DAD (Duplicate Address Detection) denir. Cihaz doğru öneki almak için yönlendiriciye bir RS mesajı gönderir. Bu dört mesajı kullanan cihaz kendisine dünya çapında benzersiz bir IPv6 adresi atayabilir [21].

Kaynak adres otomatik konfigürasyonunu kullanarak her cihaz, IEEE 802.15.4 EUI-64 adresinden bağlantı-yerel IPv6 adresi oluşturur. Kaynak adres otomatik konfigürasyonunu kullanarak her ana bilgisayar, IEEE 802.15.4 EUI-64 adresini kullanarak bir bağlantı-yerel IPv6 adresi oluşturur. Mesh-under yapılandırmada, bağlantı-yerel kapsamı tüm 6LoWPAN ağını kapsar ve bir bağlantı-yerel adresi, 6LoWPAN'da gerçekleşen iletişim için yeterlidir. Yönlendirilebilir bir IPv6 adresine, 6LoWPAN ağının dışında iletişim

[3] IID: cihazın MAC adresinden türetilmektedir.

kurulacağı zaman ihtiyaç duyulur. Bir rota-üstü (route-over) konfigürasyonunda, radyo kapsama alanı içindeki düğümlerle iletişim kurmak için bir bağlantı-yerel adresi yeterlidir, ancak birkaç atlama mesafesindeki cihazlarla iletişim kurmak için yönlendirilebilir bir adres gereklidir.

Tüm unicast adreslerini IEEE EUI-64 adresinden türetmek en verimli yöntemdir. 6LoWPAN'ın adaptasyon ve IP başlıkları arasındaki bağlantı adres çözümleme ihtiyacını ortadan kaldırır ve böylece daha küçük başlıklar elde edilir. Benzer şekilde, otomatik konfigürasyon, adreslemeyi ortak bir önek kullanacak şekilde yapılandırmalıdır. 6LoWPAN, IPv6 adresini türetmek için kısa bağlantı adresini kullanabilir ve bu sayede daha kısa başlıklar elde edebilir.

2.3. 6TiSCH

Bölüm 2.1'de özetlenen IEEE 802.15.4 standardı gecikme, güvenilirlik, ölçeklenebilirlik veya tehlikeli ortamlardaki kısıtlı kaynaklara sahip uygulamalar için uygun olmayan bazı sınırlamaların bulunduğunu vurgulamıştır [16]. Bu sınırlamaları aşmak için, IEEE çalışma grubu, 802.15.4 MAC katmanına işlevsellik kazandırmak amacıyla, gömülü uygulamaların ortaya çıkan ihtiyaçlarını gidermek için 2008'de bir Çalışma Grubu (802.15 Görev Grubu 4e) kurmuştur. 2012 yılında 802.15.4e standardının yayımlanması ile ilk nihai sonuç elde edilmiştir [25]. 802.15.4e standardı, endüstriyel uygulamalar için mevcut standartlardan (WirelessHART [26] ve ISA 100.11.a [27]) devralınan kanala erişim, paylaşılan ve tahsis edilmiş alanlar, çok kanallı iletişim ve frekans atlaması da dahil olmak üzere birçok fikri kapsayan bir ortam erişim katmanıdır. Geliştirilen ortam erişim katmanı aşağıdaki genel özellikleri içermektedir:

- Düşük Enerji (Low Energy - LE). Bu mekanizma, enerji verimliliği için gecikmeye neden olan uygulamalar için tasarlanmıştır. Bu özellik bir düğümün çok düşük bir iş çevrimiyle (ör.% 1 veya daha düşük) çalışmasına izin verir.
- Bilgi Öğeleri (Information Elements - IE). MAC alt katmanında bilgi alışverişinde bulunmak için genişletilebilir bir mekanizma sunmaktadır.
- Geliştirilmiş İşaretler (Enhanced Beacons - EB). Geliştirilmiş İşaretleyiciler, 802.15.4 beacon çerçevesinin bir uzantısıdır ve uygulamaya esneklik

sağlamaktadır. EB'ler ilgili IE'ler ekleyerek uygulamalara özel çerçeveler oluşturmalarına izin verir.

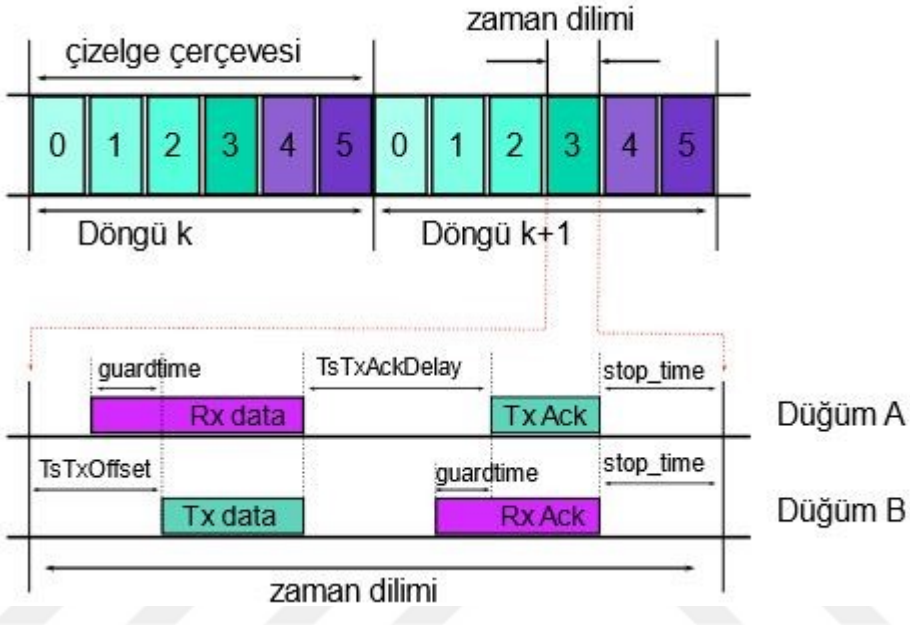
- Çok Amaçlı Çerçeve (Multipurpose Frame - MF). Bu mekanizma, ortam erişim katmanında yapılan işlemlere hitap eden esnek bir çerçeve formatı sunmaktadır.
- MAC Performans Metriği (MAC Performance Metric - MACPM). Bu özellik, uygun kararın alınabilmesi için ağ ve üst katmanların kanal kalitesi hakkında uygun geri bildirim sağlamak için bir mekanizmadır. IP protokolünün kanal koşullarına bağlı olarak veri iletisini dinamik parçalaması örnek olarak verilebilir.

Yukarıda belirtilen özelliklere ek olarak IEEE endüstriyel gereksinimleri karşılamak amacıyla, 802.15.4 protokolüne zaman paylaşımlı (Time Slotted) ve kanal atlamalı(Channel-Hopping) ortam erişim yöntemi eklenmiştir. TSCH modu esas olarak süreç izlemeye odaklı otomasyon uygulamalarının desteklenmesi için tasarlanmıştır. Uygulama alanları petrol ve gaz endüstrisi, gıda ve içecek ürünleri, kimyasal ürünler, su/atık su arıtımı, yeşil enerji üretimi, iklim kontrolü gibi alanlar kapsamaktadır. TSCH zaman paylaşımlı erişimi çoklu kanal atlamalı yeteneklerle birleştirir.

Zaman paylaşımlı erişim, komşu düğümler arasındaki çakışmayı ortadan kaldırarak elde edilebilecek potansiyel işleri (throughput) artırır ve uygulamalara deterministtik gecikme sağlar. Çoklu kanal, daha fazla düğümün aynı anda (diğer bir deyişle aynı zaman diliminde) farklı kanal ofsetlerini kullanarak çizelge (zaman) çerçevesini (slotframe) değiştirmesini sağlar. Dolayısıyla ağ kapasitesini arttırır. Buna ek olarak kanal atlama, girişim ve çok-yollu sönümlenme etkilerini azaltır ve böylece iletişim güvenilirliğini arttırır. Dolayısıyla TSCH zaman paylaşımlı erişim modu sayesinde çok düşük görev döngüsünü (enerji verimliliği) korurken artan ağ kapasitesi, yüksek güvenilirlik ve öngörülebilir gecikme sağlar. Kanal atlama yöntemi özellikle girişimin yüksek olabileceği endüstriyel alanlar için üstün bir kararlılık performansı sağlar. TSCH, herhangi bir ağ topolojisini (yıldız, ağaç, örgü, hibrit) kullanılabileceği gibi aynı zamanda da kullanmayabilir. Frekans atlaması, mevcut kaynakları verimli bir şekilde kullanmasına izin veren çok sekmeli ağlar için özellikle uygundur.

TSCH modunda düğümler, zaman dilimlerinden oluşan periyodik bir çizelge çerçevesine (slotframe) senkronize olurlar. Genellikle 10 milisaniye uzunluğundaki zaman aralığı, çerçevenin iletilmesi ve iletilen mesajın onayı için yeterli bir süredir. Her düğüm, ağı

duyurmak için periyodik olarak diğer düğümler tarafından gönderilen geliştirilmiş işaret çerçevelerinden senkronizasyon, kanal atlama, zaman aralığı ve çizelge çerçeve bilgisini almaktadır. Bu noktadan sonra, çizelge çerçevesi düğümlerin paylaşılan zaman aralığına bağlı olarak kendini tekrar eder ve işaretçinin iletişimi tekrardan başlatmasına gerek duymaz.



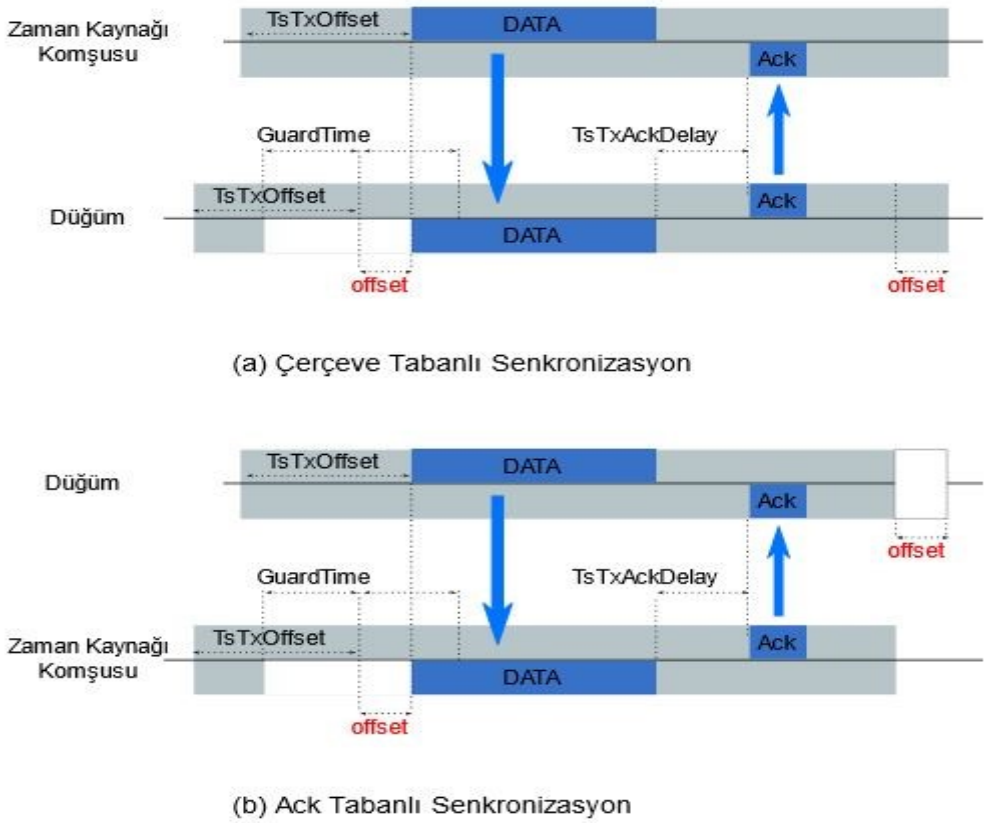
Şekil 25. TSCH Çizelge çerçevesi (üstte) ve zaman paylaşımlı (altta)

Şekil 25'te gösterilen zaman dilimi içerisinde, veri paketleri zaman diliminin başlangıcından itibaren TsTxOffset'ten hemen sonra iletilir. Bununla birlikte, senkronizasyonun bozulması durumunda alıcı düğüm GuardTime kanalını dinlemeye başlar. Buna ek olarak, paketin alımı TsTxOffset'ten sonra GuardTime içinde başlamıyorsa, düğüm enerji tasarrufu yapmak için radyosunu kapatır. Sıcaklık ve besleme gerilimindeki farklılıklar nedeniyle, farklı düğümlerin saatleri tipik olarak farklı frekansta darbe alıp saat kaymasına neden olmaktadır. Bu nedenle, düğümlerin periyodik olarak yeniden senkronize edilmesi gerekmektedir.

TSCH protokolü, düğümler arasında iletilen paketlere (data ve ACK çerçeveleri) zaman bilgisi eklemektedir. Bu işlem, komşu düğümlerin veri alışverişinde bulundukları zaman birbirlerine yeniden senkronize olabileceği anlamına gelmektedir. IEEE 802.15.4-e TSCH protokolünde düğümler arasındaki senkronizasyon işaretçi tabanlı sağlanabileceği

gibi haberleşme sırasında gönderilen ACK tabanlı da gerçekleştirilebilir. Her iki durumda da alıcı çerçeve varışının beklenen zamanı ile gerçek varış süresi arasındaki farkı hesaplar.

- Düğüm, her seferinde komşusundan aldığı veri paketini, paketin alınmaya başladığı anı referans alarak damgalamaktadır. Daha sonra senkronizasyona yardımcı olan komşu düğümle zamansal olarak eşleşmek için zaman aralığının sınırları kaydırılır. Bu adım ağa dahil olan bir cihazın işaretçiyi (EB) duyduktan sonra başlangıçta komşuya eşitlenmesi için kullandığı prosedür gibidir. Buna "çerçeve tabanlı senkronizasyon" da denmektedir [28].
- ACK tabanlı senkronizasyonda, onay mesajını (ACK) ileten cihaz kendi zamanlama bilgileriyle onay iletisini alacak cihazın zamanlama bilgisini karşılaştırır. Paketin düğüm tarafından alınma zamanıyla, düğümde alınması gereken zaman arasındaki fark bulunarak karşılaştırma işlemi gerçekleştirilir. Aradaki zamanlama farkı, paketi gönderen düğüme ACK paketine konulacak bir zaman damgası yardımıyla bildirilir. Hangi düğümün zamanlama için kullanılacağı ise düğümün ağdaki pozisyonuna göre belirlenebilir.



Şekil 26. IEEE802.15.4e TSCH protokolünde senkronizasyonu sağlayan iki farklı yöntem

Şekil 26'da IEEE802.15.4e TSCH protokolünde senkronizasyonu sağlayan iki farklı yöntem gösterilmektedir. Her iki senkronizasyon (çerçeve tabanlı veya ACK tabanlı) durumunda alıcı, offset olarak adlandırılan göreceli senkronizasyon süresini ölçmektedir. offset değeri gelen verinin başlangıç zamanı ile TsTxOffset arasındaki fark olarak hesaplanmaktadır. Çerçeve tabanlı senkronizasyonda, offset alıcının zaman çizelgesine uygulanırken ACK tabanlı senkronizasyonda gönderenin zaman çizelgesine uygulanır. TSCH ağındaki her cihaz, zaman dilimlerinde nasıl davranacağını (iletim, alım ve radyonun durumu) belirleyen bir süreci izler. Ayrıca, her aktif zaman dilimi (slot) için süreç, iletişim kuran komşu düğümleri ve kanal offset değerini de göstermektedir. Belirlenen zaman aralığı tahsisi, öngörülebilir iletim modelini sağlayarak düğümler arasında belli sürelerde haberleşmeyi gerçekleştirmektedir. Bu yöntem, cihazların birbirlerini gereksiz dinlemesini önleyerek pil ömrünü uzatmaktadır.

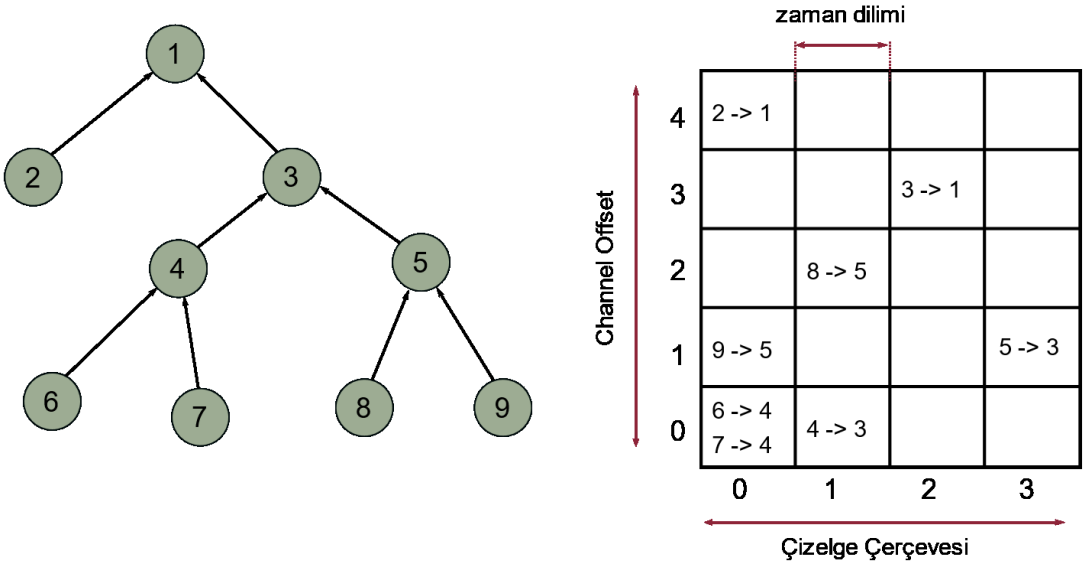
TSCH protokolü kanal atlamaya dayanan çok kanallı iletişimini desteklemektedir. Varsayılan olarak iletişim için başlangıçta 16 farklı kanal mevcuttur (2.4 GHz frekans bandı

16 kanal içerir). Her kanal, 0 ile 15 aralığında bir tam sayı değeri olan channeloffset tarafından tanımlanır. Bununla birlikte kanallardaki bazı zaman dilimleri (düşük kaliteli iletişim nedeniyle) kara listeye alınabilir ve bu nedenle kanal atlamada kullanılabilen mevcut kanal sayısı 16'dan düşük olabilir. TSCH çizelgesindeki zaman dilimleri, düğümler arasındaki iletişimin hangi kanalda gerçekleşeceğini göstermektedir. Dolayısıyla, iletişim kuran düğümler arasındaki bağlantı, çizelge çerçevesindeki zaman dilimi ve bu zaman dilimindeki düğümler tarafından kullanılan kanal ofset değeri ile temsil edilebilir. Frekans f, ağın başlangıcından bu yana geçen toplam zaman dilimini ifade eden ASN (Absolute Slot Number) değerinden ve mod operatöründen oluşmaktadır. ASN parametresi her zaman diliminde artar ve düğümler tarafından zaman aralığı sayıcısı olarak kullanılır. F fonksiyonu kanal ofseti frekansa çevirmek için kullanılır. Çok kanallı mekanizma sayesinde, farklı kanal ofsetlerini kullanarak aynı zaman diliminde birden fazla iletişim gerçekleşebilir. Ayrıca Denklem (1), aynı bağlantı için farklı zaman aralıklarında farklı bir frekansta ilerleyerek kanal atlama mekanizmasını sağlamış olmaktadır. Bu, zaman içinde tüm mevcut kanalların iletişimler için kullanılmasını ve dolayısıyla girişimin (interference[4]) olumsuz etkisinin azaltılmasını sağlar.

$$f = \mathrm{F}[(\mathrm{ASN} + \mathrm{channelOffset})\%\ 16] \quad (1)$$

Şekil 27'de, ağaç tabanlı topolojiye sahip olan duyarga ağında veri toplamak için kendini zaman içinde tekrar eden olası bir çizelge (süreç, zaman) çerçevesi gösterilmektedir. Çizelgenin 4 eşit zaman diliminden ve 5 kanaldan oluştuğu varsayılmaktadır. TSCH tarafından kullanılan çok kanallı yaklaşım sayesinde, 4 zaman aralığının belli dilimlerinde 8 farklı iletişimin gerçekleştiği görülmektedir. Gösterilen çerçevede, paylaşımlı ve belli amaçlara yönelik tahsis edilen zaman dilimleri yer almaktadır. TSCH protokolü zaman dilimlerini sadece iki düğüm arasında tahsis ederek kullanabileceği gibi ikiden fazla düğüme hizmet sunmak için paylaşımlı olarak da kullanabilir. Denklem (1), hem paylaşılan hem de tahsis edilen zaman dilimleri için iletişim frekansını belirlemede de kullanılmaktadır.

[4] Haberleşme sistemini olumsuz yönde etkileyen sinyallerdir.



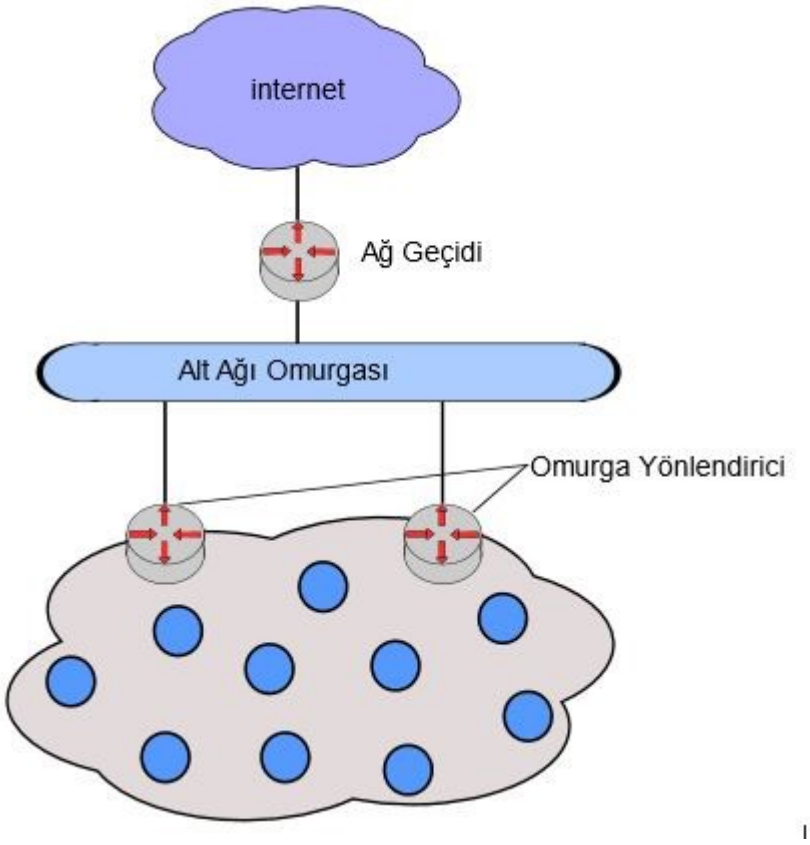
Şekil 27. Olası Bir Ağaç Topolojisine Sahip Duyarga Ağı

IEEE802.15.4-e, TSCH düğümünün iletişim kurması için gerekli mekanizmaları açıklarken, haberleşme planını oluşturmayı ve haberleşmenin devamını sağlamayı dikkate almaz. IPv6 tabanlı bir yönlendirme protokolünün zaman paylaşımlı ve kanal atlamalı bir ortam erişim protokolü ile kullanılacağını tanımlamaz. Ayrıca komşuları keşfetmek, topoloji değişikliklerine tepki vermek veya IP adreslerini kendi kendine yapılandırmak amacıyla 6LoWPAN ve RPL tarafından gerekli olan sinyal mesajları için sürecin nasıl çalışacağı konusunda herhangi bir şart yoktur. Bu nedenle, 802.15.4-e ortam erişim protokolünü Nesnelerin İnternetine dâhil edecek ve Endüstriyel İnternet'in parçası haline getirecek standart çözümler gereklidir. IEEE802.15.4e'ye özgü sorunları çözmek ve böylece RPL tarafından organize edilen çok sekmeli IPv6 tabanlı ağlarda uygun bir entegrasyon sağlamak için IETF 6TiSCH çalışma grubu kurulmuştur ([12, 29, 30, 31]). Bu çalışma grubu her biri belirli amaçlara hitap eden İnternet metinleri oluşturarak protokolün geliştirilmesini amaçlamaktadır.

2.3.1. 6TiSCH Mimarisi

6TiSCH çalışma grubu, belirli bir fiziksel ortamda konuşlandırılan yüzlerce/binlerce düğümden oluşan ve ortam erişim katmanında TSCH kullanan düşük güçlü kayıplı ağları ele almaktadır. Ağdaki tüm düğümler aynı IPv6 alt ağına aittir ve IPv6 üzerinden iletişim

kurarlar. Ağın binlerce düğüme kadar ölçeklenmesine izin vermek ve tek bir IPv6 alt ağı olarak görülebilmek için, tüm fiziksel çevreyi kapsayan yüksek hızlı bir omurga varlığı varsayılmaktadır. Omurga tüm düğümleri birbirine bağlamak ve senkronize etmek için hızlı bir altyapı sağlamaktadır. Kaynak kısıtlı düğümler omurgaya bir veya birden fazla Omurga Yönlendiricisi (BBR) vasıtasıyla bağlanırken tüm omurga bir Ağ Geçidi aracılığıyla İnternet'e bağlanır. Şekil 28, genel mimariyi göstermektedir.



Şekil 28. 6TiSCH Mimarisi

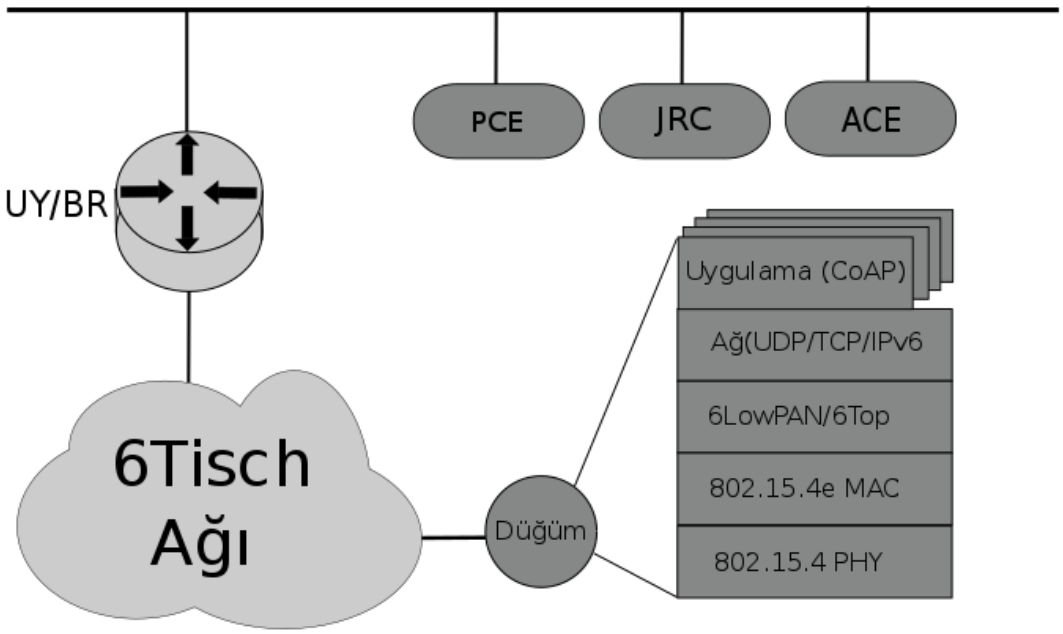
6TiSCH, IEEE 802.15.4 fiziksel ve IEEE 802.15.4e TSCH ortam erişim katmanlarını IETF katmanlarıyla (yani, 6LoWPAN, RPL, CoAP, vb.) birleştirmeyi hedeflemektedir. Şekil 4'te gösterilen 6TiSCH protokol yığınında CoAP, ağın düğümleriyle RESTful etkileşimine imkan vermektedir [32]. 6LoWPAN, kablosuz ortam üzerinden iletilecek paket boyutunu azaltmak için IPv6 başlıklarını sıkıştırırken; RPL yönlendirme topolojisi oluşturur. Bu standart çözümler sayesinde kestirilebilir ve yüksek güvenilirlik sağlayan duyarga ağların gerçekleşmesi mümkün olmaktadır. IETF protokollerinin en iyi şekilde çalışabilmesi

için gerekli boşlukların doldurulması gerekmektedir. Bu amaçla, 6top olarak adlandırılan yeni bir katman, 6TiSCH çalışma grubu tarafından tanımlanmaktadır. 6top katmanı bir Yönetim Birimi'nin (ME) TSCH çizelgesini (schedule), örneğin bağlantıları (6TiSCH terminolojisine göre hücreler) eklemek veya kaldırmak için kontrol etmesini sağlar. Buna ek olarak, 6top üst katmanlar için yararlı olabilecek bağlantı bilgilerini toplar ve beklendiği gibi davranmazlarsa hücrelerin performansını izleyerek yeniden planlamalarını sağlar. Çizelge belirleme işlemini üstlenecek öğe merkezi olarak gerçeklenebileceği gibi, dağıtık bir yöntemle de gerçeklenebilir. 6top bu iki yaklaşım ile birlikte kullanılmak üzere tasarlanmıştır. Bu amaçla, her hücreyi sabit bir hücre veya yumuşak bir hücre olarak sınıflandırır. Sabit hücre 6top tarafından dinamik olarak yeniden tahsis edilemez çünkü genellikle merkezi çizelge öğesi tarafından yüklenir ve kaldırılır. Aksine, yumuşak hücreler kötü performansa sahiplerse dinamik olarak 6top ile yeniden düzenlenebilir. Yumuşak hücreler genellikle 6top üzerinde çalışan dağıtık algoritmalar tarafından belirlenir. Bununla birlikte, çizelge algoritmaları yalnızca belirli bir komşuya kaç tane yumuşak hücrenin planlanması gerektiğini belirtir. Ayrıca, TSCH çizelgesinde her hücrenin belli bir yere (slot, kanal ofseti) eşlenmesi 6top'un sorumluluğundadır. Buna ek olarak, 6TiSCH ağı farklı trafik akışlarını (farklı hizmet gereksinimleri ile) taşıyabildiğinden 6top, farklı trafik akışlarını tanımlayacak şekilde farklı etiketler içeren hücreleri belirleyebilir ve böylece akış izolasyonuna izin verebilir. Ayrıntılı olarak bir paket 6TiSCH ağına gönderildiğinde, paketi alan düğümünün 6top katmanı, paketin ait olduğu hizmet sınıfını tanımlar ve buna göre yapılacak işlemleri gerçekleştirir.

2.3.2. Çizelgeleme ve Yönlendirme

6TiSCH protokolü, TSCH çizelgesini oluşturmak ve sürdürmek için minimal, merkezi ve dağıtık çizelge olmak üzere üç farklı çizelge modu tanımlamaktadır. Minimal yapıda TSCH çizelgesi sabittir ve ağa katılan düğüm tarafından önceden yapılandırılmış veya öğrenilmiştir. Bu zaman çizelgesi IEEE802.15.4e TSCH protokolünün tüm avantajlarından yararlanamaz fakat ağ önyükleme sırasında veya daha iyi bir zamanlama çizelgesi olmadığı durumlarda kullanılabilir. 6TiSCH minimal çizelgeleme için oluşturulan taslak [33], 6TiSCH ağlarında kullanılacak olan bu yapıyı detaylı bir şekilde açıklamaktadır. Şekil 29'da gösterilen merkezi çizelgelemede, ağda yer alan Yol Hesaplama Öğesi (Path Computation Element- PCE), tüm düğümlerin ağ durumu bilgisini ve trafik gereksinimlerini toplar. Tüm

ağ trafiğinin ihtiyaç duyduğu gereksinimler belirlendikten sonra TSCH çizelgesi oluşturulur ve ağdaki düğümlere iletilmek üzere hazır hale getirilir. 6TiSCH, PCE ve ağdaki düğümler arasında mesaj alışverişi için kullanılacak protokolleri ve değiştirilecek kontrol mesajlarının biçimini tanımlamaktadır. Bağlanma Kayıt/Koordinasyon Öğesi (Join Registrar/Coordinator -JRC) ağa bağlanacak düğümün kimliklendirilmesi işlemini koordine eder. Kısıtlı Ortamlar için Kimliklendirme Öğesi (Authentication Coordination Element) düğümün ağa dâhil edilmesi sırasında kimliklendirilmesi ve yetkilendirilmesi işlevlerini yerine getirir. Bütün bu öğeler Şekil 29'da görüldüğü gibi sınır yönlendiricisinin bulunduğu ağda hayata geçirilebileceği gibi, farklı ağların parçası olarak da hayata geçirilebilir.



Şekil 29. 6Tisch Merkezi Kontrol Öğeleri [34]

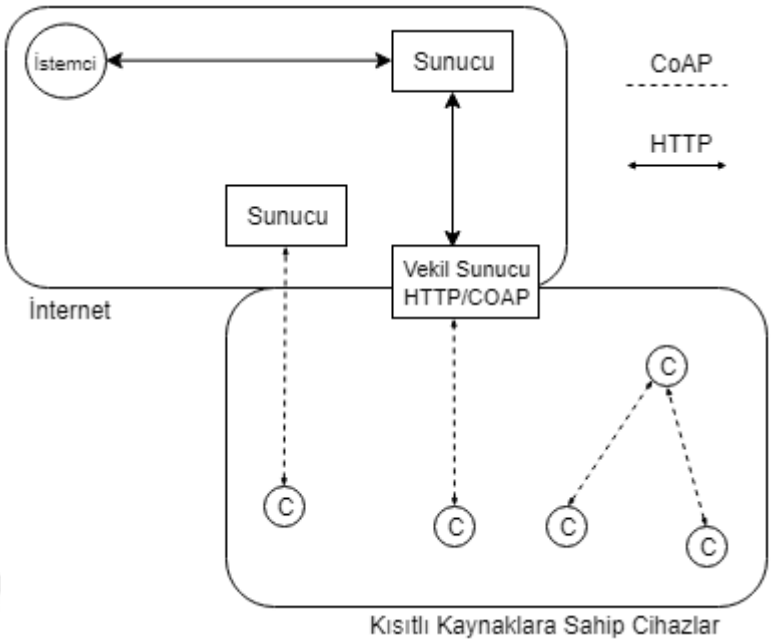
Dağıtık çizelge yönteminde, düğümler merkezi bir kontrolden bağımsız olarak kendi aralarında haberleşebilmeleri için gerekli kaynakları belirlerler. Çizelge belirlemekle yükümlü olan düğümler, komşularının kullandıkları kaynakları takip ederek bu işlemi gerçekleştirmektedir. Bu yöntem sistemin ölçeklenebilir olmasını garanti eder. 6Top protokolü dağıtık bir çizelge yöntemini belirleyerek kaynakların haberleşen düğümler arasında ayrılmasını sağlamaktadır. 6Top ayrıca farklı trafik tiplerini ayrıştırarak, trafik türüne göre öncelikler belirleyebilir. Bu sayede ağ içerisinde uygulama tabanlı farklı servis sınıflarının oluşturulması sağlanabilir.

2.4. CoAP

Uygulama katmanı protokolleri, Nesnelerin İnternetinde büyük öneme sahip cihazların kullanmış olduğu farklı yapıdaki uygulamaların birbirleri ile haberleşmesine olanak sağlamaktadır. Bu özellikleri sayesinde her uygulama için özel olan tasarımlardan bağımsız olarak uygulamalar kendi aralarında haberleşebilmektedir. CoAP (Constrained Application Protocol – Kısıtlı Uygulama Protokolü) Nesnelerin İnternetinde var olan kısıtlı kaynaklara sahip düğümlerin oluşturduğu ağlarda ve Makine – Makine (M2M) uygulamalarında kullanılmak üzere geliştirilmiş bir web aktarım protokolüdür. M2M, makinelerin birbirleriyle, özellikle de kablosuz kanallardaki İnternet protokolleri üzerinden iletişim kurmasını sağlayan tüm teknolojileri ifade ettiği için IoT'nin bir alt kümesi olarak görülebilir. HTTP gibi, CoAP bir dosya aktarım protokolüdür. HTTP'den farklı olarak, CoAP kısıtlı cihazların ihtiyaçları için tasarlanmıştır ve UDP üzerinde çalışmaktadır. Bu protokol HTTP'nin istemci/sunucu yapısına benzemekle birlikte makineden-makineye çalıştığı için istemci/sunucu rollerini birlikte yürütmektedir.

CoRE çalışma grubu, kısıtlı kaynaklara sahip cihazların oluşturduğu ağlar ile İnternet arasındaki haberleşme için Şekil 30'da gösterilen ReST mimarisinin gerçekleştirilmesini amaçlamaktadır [32]. Çalışmaları, aşağıdaki özelliklerle önerilmiş olan CoAP protokolünün özelliklerini içermektedir:

- M2M gereksinimlerini karşılayan kısıtlı web protokolü,
- İsteğe bağlı olarak unicast ve broadcast mesajlarının güvenli iletimi,
- Asenkron mesaj değişimi,
- URI ve İçerik türü desteği,
- HTTP/CoAP ön bellekli vekil sunucu desteği,
- Başlık iletişim yükünün azlığı,
- UDP paketlerinin IPSec veya DTLS ile kullanılmasına izin verir.



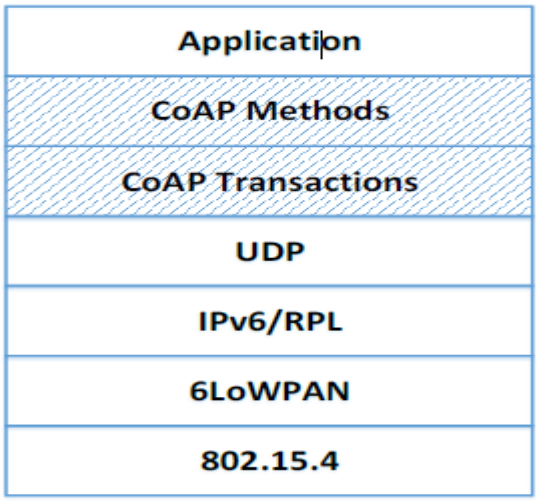
Şekil 30. CoRE ReSTful mimarisi

CoAP protokolü, Şekil 31'de gösterildiği gibi CoAP Methods ve CoAP Transactions olmak üzere 2 alt protokol katmanına ayrılmaktadır. Transactions katmanı güvenilir UDP mesajlaşmasından sorumludur ve Tablo 4'te gösterilen farklı mesaj türlerini kullanmaktadır. CoAP, sunucunun herhangi bir kaynak üzerinde işlem yapabilmek için istemcilerin CoAP istekleri gönderdiği kabul edilmektedir. İstek mesajları çözüldükten sonra, sunucu uygun bir yanıt kodu ve isteğe bağlı olarak kaynak gösterimi içeren yanıt mesajını istemciye geri göndermektedir.

Tablo 4. CoAP Yöntemleri

GET	Belirlenen bir kaynağın bilgilerini alır
POST	İstenen URI'nin altında yeni bir kaynak oluşturur.
PUT	Bir URI tarafından tanımlanan kaynağı günceller
DELETE	Bir URI tarafından tanımlanan kaynağı siler

Sunucu, isteği kabul ettikten sonra mesajı ayrıntılı bir şekilde inceler ve hiçbir hata oluşmazsa istemciye, istek sürecinin sonucunu gösteren bir mesaj gönderir. Yanıt kodunun durumuna göre istemci ve sunucu kısımlarında hatayı düzenleyecek adımlar atılmalıdır. Örneğin, en yaygın HTTP yanıt kodu, istemci isteğinin doğru olduğunu ancak sunucunun URI alanı tarafından gösterilen kaynağı bulamadığını gösteren 404 hatasıdır.



Şekil 31. CoAP Katmanları

CoAP varsayılan olarak UDP'ye bağlı olduğundan istekler ve yanıt kopyalanmış veya eksik olabilmektedir. Bu sorunun üstesinden gelebilmek için, protokol teorik olarak iki mantıksal katmana bölünmüştür; üst katman, daha önce tanıtılan istek/yanıt mekanizmalarını ve alt katman ise güvenlik mekanizmasını işlemektedir. CoAP Transactions katmanı güvenilir UDP mesajlaşmasından sorumludur, 4 farklı mesaj türünü kullanır:

- Confirmable (CON): Taşınan verilerin güvenilirlik işlevini sağlayarak alıcıdan onaylanması gerektiğini gösterir.
- Non Confirmable (NON): onay gerektirmeyen fakat yine de mesaj çoğaltmasından korunması gereken verileri taşır.
- Acknowledgement (ACK): CON iletilerini onaylar.
- Reset (RST): CON veya NON mesajının alınmasında oluşan hataları gösterir.

CON/ACK mesajlarının ve kopya mesajların belirlenmesi CON ve NON mesajından üretilen ve içinde mesaj kimlik bilgisi içeren parametre yardımıyla yapılır. Mesaj kimlik bilgisi, iki uç nokta arasındaki her NON veya CON/ACK iletisi için benzersiz olmalıdır. Aynı hedefe karşılık gelen ACK mesajının alınmasına kadar, CON iletisi, üstel bir bekleme süresi olan varsayılan zaman aşımı kullanarak kanal üzerinden yeniden gönderilir. İstemci isteği, URI adı verilen sunucu kaynağının benzersiz bir tanımlayıcısı ve isteğe bağlı olarak istemci isteğine ilişkin meta verileri içeren bir veri yükünü belirten yöntem içermektedir.

3. NESNELERİN İNTERNETİ İÇİN GÜVENLİK GEREKSİNİMLERİ

Nesnelerin İnternetinde cihazların birbirleriyle ve merkezi birimlerle güvenli bir haberleşme gerçekleştirebilmesi için alt başlıklar halinde bahsedilen güvenlik gereksinimlerini sağlamalıdır. Bu gereksinimler güvenli iletişimin temelini oluşturduğundan tasarlanacak sistemler bu şartları sağlamalıdır.

3.1. Veri Gizliliği

Gizlilik, bilgiyi istenmeyen kişilerden veya tanımlanamayan bölümlerden saklı tutmaktır. Veri gizliliği, ağ güvenliğinin en önemli gereksinimlerinden birisidir ve kablosuz duyarga ağlar gibi potansiyel olarak yüksek riskli bir iletişim ortamına sahip ağlarda çok daha fazla önem taşımaktadır. Güvenli haberleşmeyi gerçeklemek ve mesajın yetkisi olmayan kişiler tarafından kullanılmasını engellemek için iletilen mesajların şifrelenmesi gerekmektedir.

3.2. Veri Bütünlüğü

Kötü niyetli kişilerin bilgileri izinsiz olarak elde etmesi veri gizliliğinin sağlanması ile engellenebilmektedir, ancak saldırganlar bilgileri elde edemese de değiştirebilir ya da bilgilere ekleme yapabilir [35]. Ağda iletilen mesajın herhangi bir ağ öğesi tarafından değiştirildiği algılanmalıdır ve değiştirilmiş mesaj reddedilmelidir. Bir mesajın iletim sırasında değiştirilip değiştirilmediği Mesaj Doğrulama Kodu gibi çeşitli mekanizmalarla kontrol edilebilir.

3.3. Kimlik Doğrulama

Kimlik doğrulama, birbirleriyle haberleşen düğümlerin karşılıklı olarak kimliklerini doğrulaması olarak tanımlanabilir [36]. Bu durumda, saldırgan düğüm, kaynak doğrulama olmadan haberleşmeye dâhil olamaz. İki cihazın güvenli iletişimi göz önüne alındığında,

veri doğrulama/kimlik denetimi için en basit yöntem simetrik bir mekanizmanın kullanılmasıdır. Kaynak ve hedef cihazlar gizli bir anahtarı paylaşarak haberleşmede kullanılacak verilerin doğrulama kodunu hesaplar. Kaynak cihaz, doğrulama kodu hesaplanan mesajı alıcı cihaza gönderdiğinde, alıcı tarafından gelen mesajdan elde edilen yeni kod değeri ile göndericinin doğruluğu ispatlanmış olur [35].

3.4. Veri Güncelliği

Duyarga ağlar fiziki ortamdan elde ettikleri anlık değişen verilerin güvenliğinin yanında aynı zamanda iletilen her mesajın tazeliğini de sağlaması gerekir. Ağda herhangi bir mesaj daha önceki bir mesajın tekrar iletilmiş haliyse, bu veri ağdaki düğümler tarafından değerlendirilmemelidir. Bu gereksinim özellikle paylaşımlı anahtar tekniklerinin kullanıldığı uygulamalarda büyük önem taşımaktadır. Bu tekniklerde paylaşılan anahtarların belirli zaman aralıklarında değiştirilmesi ve anahtar değişimden ağdaki tüm düğümlerin haberdar edilmesi gerekmektedir. Ağdaki veri güncelliğinin sağlanması için standart yaklaşım, iletilen her bir mesajın içerisine mesajın geçerli olduğu süreyi gösteren bir zaman damgasının eklenmesidir.

3.5. Hizmet Bütünlüğü

Cihazlardan elde edilen verilerin kayba uğramadan güvenli bir şekilde toplanabilmesine hizmet bütünlüğü denmektedir [37]. Veriler, düğümlerden alınarak ağ geçidine veya veriyi işleyecek düğümlere iletilir. Hizmet bütünlüğü açısından düğümlerden elde edilen hatalı verilen belirlenmesi ve doğadan elde edilen veri ölçümlerinin doğru hesaplanması önemlidir.

3.6. Esneklik

Duyarga cihazlarının oluşturduğu ağlar, çevresel koşulların hızla değişebileceği çevre senaryolarında kullanılabilmektedir. Cihazların görevlerinin değiştirilmesi, kurulu bir ağdan

kaldırılmasını veya yeni cihazların ağa eklenmesi gerekebilir. Duyarga ağının karşılaşabileceği olası tüm senaryolar için esnek bir yapıda ağlar tasarlanmalıdır.

3.7. Kullanılabilirlik

Anahtar yönetim hizmetleri, gerektiğinde yetkili taraflar için gizlilik ve grup düzeyinde kimlik doğrulama hizmetlerinin mevcut olmasını ve ağdaki hizmetin kesilmesini isteyen saldırılara karşı koruma sağlamalıdır. Mesajın güvenliğini sağlamak ve enerji tüketimini azaltmak için, duyarga ağı ömrünü uzatmak için ana yönetim mesajlarının gereksiz şekilde kullanılmasını engellemelidir. Anahtar yönetimi işlevleri, ağın kullanılabilirliğini sınırlamamalı ve tüm ağ genelinde güvenlik için merkezi anahtar yönetimi düğümü gibi tek bir hata noktası oluşturmamalıdır. Herhangi bir nedenden ötürü, duyarga düğümleri, diğer düğümlerle iletişime geçebilmek için uygun anahtar değerine sahip olsa bile senkronize durumunu gerçekleşmezse, ağın kullanılabilirliği zarar görebilir. Bu nedenle oluşturulan ağ, maksimum seviyede güvenlik sağlamasa dahi ihtiyaç durumlarına cevap verebilmelidir.

3.8. Senkronizasyon

Ağı oluşturan duyargaların gerçekleştirmiş olduğu uygulamalar zaman senkronizasyonu altında çalışmalıdır. Ağın yaşam döngüsünü arttırmak ve güç durumunu minimize etmek amacıyla, duyargalar belirli periyotlarda ve senkronizasyon durumlarında “uyuma” konumunda çalışırlar (radyo uyku(kapalı) moduna geçer). Böylece enerji tasarrufu sağlanmış olmaktadır.

4. LİTERATÜR TARAMASI

Literatür taramasının asıl amacı, arka plan çalışması için yetki ve doğrulama ile ilgili literatürü bulmak, mevcut çalışmaları özetlemek ve mevcut araştırmadaki boşluğu belirlemektir. IEEEXplore, ACM, Scopus ve Google Scholar gibi çevrimiçi araştırma veritabanları araştırılmıştır. Son zamanlarda, küçük aygıtları IoT ağlarına önyüklemek için giderek artan sayıda çalışma bulunmaktadır fakat bu çalışmaların birçoğu gerçek cihazlar üzerinde uygulanmamıştır. Bu eksikliği gidermek için, tezde gerçek cihazlara uygulanabilecek basit bir kimlik doğrulama mekanizması önerilmiştir.

Yazarlar [38] 'de, düğümlerin merkezi bir doğrulama birimi ile kimlik doğrulamasının yapıldığı doğrulama mekanizması önermektedir. Cihazların IoT ağının üyesi olmasını sağlayan parametrelerle önceden yapılandırıldığı varsayılmaktadır. M2M ağları için karşılıklı kimlik doğrulama mekanizması, düğümlerin merkezi olarak 6LoWPAN kenar yönlendiricisi aracılığıyla kimlik doğrulamasının yapıldığı [39] 'da önerilmiştir. Önerilen yöntem kimlik doğrulama ve oturum anahtarı oluşturma aşamalarını içerir. Kimlik doğrulama ve oturum anahtarı yönetimi kenar yönlendirici aracılığıyla merkezi olarak ele alındığından, ağdaki düğüm sayısı büyük olduğunda bu yöntemin iyi ölçeklenemeyeceği öngörülmektedir. Ayrıca, eliptik eğrileri kullanan oturum anahtarının oluşturulması, ağın kaynak kullanımı üzerinde olumsuz bir etkisi olmaktadır. Yazarlar, kimlik doğrulama için XOR tabanlı güvenli mesajlaşma protokolünü kullanan M2M ağları için hafif bir kimlik doğrulama mekanizması önerilmiştir [40].

Altolini vd. 2013 yılında IEEE 802.15.4 uyumlu IoT cihazlarının bağlantı katmanında kullanılan AES-CCM şifreleme yönteminin performansını incelemiş ve diğer birkaç şifreleme mekanizmaları ile karşılaştırmışlardır [41]. Yazarlar, yazılım ve donanım tabanlı güvenlik mekanizmalarını AVR XMEGA platformuna uygulayarak; her bir durum için bellek ve enerji tüketimini değerlendirmiştir. Donanım temelli uygulamaların yalnızca hızlı olmadığı, aynı zamanda yazılım tabanlı güvenliğe kıyasla iki kat daha kısa gecikmelere neden olduğu gözlemlemişlerdir. Ayrıca, bağlantı katmanına donanım tabanlı güvenlik desteğinin eklenmesi, ağ ömrünü çok az etkilemektedir ve güvenlik kullanılmadığı duruma kıyasla yalnızca %2 oranında ağın ömrünün azaldığı belirtilmiştir.

Yazarlar AES şifreleme mekanizmasının tanınmış güvenlik özellikleri ve donanımsal olarak gerçekleştirme imkanı nedeniyle kısıtlı kaynaklara sahip cihazlar üzerinde

uygulanabileceğini tartışmaktadır. [42]'de sunulan çalışma, kriptografik algoritmaların IoT kaynakları üzerinde (bellek, işlem, enerji) üzerindeki etkisini değerlendirmektedir. AES simetrik şifreleme algoritmasının, kod boyutu ve hesaplama maliyeti arasında denge sağlanarak IoT cihazları için etkili şifrelemelerden biri olduğu (diğer blok şifrelere benzer şekilde) gösterilmektedir.

[43] numaralı çalışmada RPL'deki güven yönetiminin farklı konuları ele alınmıştır. Düğümler arasındaki güvenilirliği sağlamak için yalnızca TPM (Trust Platform Module - Güven Platform Modülü) [44] kullanmanın yeterli olmadığı ifade edilmektedir. Yazarlar, RPL'e güvenlik metriği ekleyerek onu güçlendirmeyi önermişlerdir. Bu metrik, ağdaki her bir düğüm için güven düzeyini temsil eder ve bencillik, enerji ve dürüstlük bileşenleri kullanılarak hesaplanmaktadır. Topolojinin oluşturulması sırasında kullanılan metrik değer, bir düğümün diğer düğümlere güvenip güvenmeyeceğine karar vermesine izin vermektedir.

Yapılan bazı çalışmalar, güvenlik işlemleri için donanımsal olarak gerçekleştirilebilecek mekanizmalarının avantajlarını kablosuz duyarga ve Nesnelerin İnterneti ağları üzerindeki etkilerini araştırmaktadır ([45, 46, 47, 48]). MICAZ ve Tmote SKY'yi de içeren birçok duyarga düğümünde bulunan Chipcon CC2420 radyo çipinde bulunan AES şifreleme modülünü kullanarak şifrelemenin maliyetini azaltmak için bir çözüm sunmaktadırlar [45]. Yazarlar, KDA için uygun olan şifreleme algoritmalarının bazılarının yazılım uygulamalarını her iki donanım platformu için analiz etmişlerdir. Huai vd. 2009 yılında IEEE802.15.4 ağları için AES-CCM kullanarak enerji tüketimi açısından etkili bir donanım mimarisi tasarlamışlardır [47]. Görev döngülerini ve enerji tüketimini azaltmak için "CTR" ve CBC-MAC işlemleri beraber kullanılmıştır. Deneysel sonuçlar, uygulamanın, kaynak kısıtlı KDA cihazları için uygun olabileceğini göstermektedir. [48] numaralı çalışmada yazarlar, Nesnelerin İnternetinde kullanılan mikro denetleyicilerde AES şifreleme altyapısını donanım, yazılım ve karma uygulamalarını donanım modülünün işleyişi, bellek alanı ve pil kullanım ömrü gibi faktörleri dikkate alarak incelemişlerdir. Ölçüm sonuçlarına göre AES tasarımının, düşük güçlü bir mikro denetleyici ile birleştirildiğinde, piyasada bulunan TI MSP430 donanımı üzerinde bit başına daha az enerji tükettiği görülmektedir.

[49, 50, 51, 52] numaralı çalışmalarda IEEE 802.15.4 standardındaki güvenlik mekanizmasının ağ performansı üzerindeki etkisi analiz edilmektedir. Daidone vd. 2011 yılında yaptıkları çalışmada bağlantı katmanı şifreleme ve kimlik doğrulama servislerinin Tmote-Sky cihazındaki enerji ve bellek tüketimi üzerindeki etkisini analiz etmişlerdir.

Çalışmada yalnızca donanım uygulaması sunulmaktadır ve TSCH ile entegrasyon da yapılmamaktadır [49]. [50] numaralı çalışmada IEEE 802.15.4 ağlarındaki bağlantı katmanı güvenlik yönteminin uygulama katmanındaki performansı ve diğer protokol parametreleri üzerindeki etkisi analiz edilmiştir. Yazarlar zaman ve enerji tüketimi üzerine odaklanmışlardır. Simülasyonlarda, şifrelemenin ve doğrulamanın uçtan uca uygulama gecikmesini etkilediği gözlemlenmiştir. Fakat bağlantı katmanı güvenliğinin ortam erişim protokolünün performansı üzerindeki etkileri hakkında bir araştırma yapılmamıştır. Yazarlar [51], IEEE802.15.4 güvenliğinin enerji sarfiyatını, enerji toplama aygıtları ve beacon modu bağlamında değerlendirmektedir. IEEE802.15.4 ağlarında güvenlik yükünün bir değerlendirmesi [52] 'de sunulmuştur. Ayrıca, güvenlik hedefleri, güvenlik takımları, güvenlik modları, şifreleme, kimlik doğrulama gibi güvenlik ihtiyaçları incelenmiştir. Çeşitli saldırıları (hizmet reddi saldırısı, yanıt koruma saldırısı, ACK saldırısı vb.) önlemek için bazı güvenlik geliştirmeleri önerilmiştir.

Kimlik doğrulama ve anahtar dağıtımı için açık anahtar şifreleme kullanan DTLS protokolünün kaynak gereksinimleri [53] numaralı çalışmada incelenmiştir. Protokolün neden olduğu kaynak maliyetinin azaltılmasına yönelik ek bir sunucu modeli önerilmiştir. Cihazların merkezi düğüm tarafından doğrulanması sağlanarak merkezi düğüm üzerindeki yoğunluk artırılmaktadır. Yetkilendirme mimarisi, kısıtlı aygıtlar için DTLS korumalı iletişimde kaynak gereksinimlerini önemli ölçüde azaltmaktadır. Değerlendirme sonuçlarına göre, önerilen mekanizma sertifika tabanlı DTLS'e kıyasla daha az bellek gereksinimine ihtiyaç duymaktadır. Önerilen mimari, IP tabanlı IoT için kimlik doğrulama, yetkilendirme ve güvenli veri iletimi için kapsamlı bir çözüm sunmaktadır.

Güvenilir bağlantı ve IoT'nin erişilebilirliğini sağlamak için, uygun kimlik doğrulama ile uçtan uca iletişimi gerçekleştirmede güvenli bağlantılar kurmak önemlidir. Yazarlar, dağıtık IoT uygulamalarında KDA'lar için kapalı (implicit) sertifika tabanlı kimlik doğrulama mekanizması önermişlerdir [54]. Geliştirilen iki aşamalı kimlik doğrulama protokolü, duyarga düğümlerinin ve son kullanıcıların (end user) merkezi bir doğrulama mekanizması tarafından doğrulanmasına ve güvenli bir şekilde ağa dahil edilmesine olanak sağlamaktadır. Önerilen protokol düğümlerin kaynak kıtlığını, ağın heterojenliğini ve ölçeklenebilirliğini desteklemektedir. Performans ve güvenlik analizi, önerilen şemanın KDA ve IoT ağları için uygun olduğunu doğrulamaktadır.

6TiSCH ağına eklenen bir cihazdaki güvenli başlangıç konfigürasyonunu desteklemek için gereken minimum mekanizmalar [55] numaralı çalışmada açıklanmıştır. Bu

konfigürasyonun amacı, bağlantı katmanı anahtarlarını belirlemek ve ağa dahil olmak isteyen düğüm ile JRC arasındaki güvenli oturumu oluşturmaktır. Belirtilen mekanizma JRC denen merkezi bir yetkilendirme sunucusunu kullanarak düğümleri ağa dahil etmektedir. Ayrıca bu çalışma, 6TiSCH ağlarındaki güvenliği ele alan ilk çalışmadır.

Güvenli önyükleme için kullanılabilecek çeşitli protokoller Sarıkaya vd. tarafından belirtilmiştir [56]. Garcia-Morchon vd. ayrıca kablosuz duyarga ağlarına yönelik tehditleri analiz ederek genel olarak önyüklemeyi güvenli hale getirme yaklaşımlarını açıklamışlardır [57].

Nyugen vd. var olan güvenlik gereksinimleri ve zorlukları temel alarak KDA'lar ve IoT için birden fazla güvenli, hafif ve saldırıya dayanıklı çözümler üzerine kapsamlı bir çalışma yapmışlardır [58]. Güvenli bir iletişim kanalı oluşturmak için anahtar önyükleme yaklaşımına dayanan mevcut protokollerin yeni bir sınıflandırması yapılmıştır. Bu protokollerin ve tekniklerin avantaj ve dezavantajları çeşitli kriterlere göre analiz edilmiştir. Simetrik şifreleme yaklaşımlarının artık IoT için varsayılan seçim olmadığına dikkat çekilmiştir. Yazarlar yetkilendirme ve kimlik doğrulama adımları için asimetrik tekniklerin düzgün şekilde optimize edilmiş olması koşuluyla, açık anahtar kriptografisinin IoT'nin vazgeçilmez bir güvenlik mekanizması olacağını belirtmişlerdir.

2015 yılında Hernández-Ramos vd. tarafından kısıtlı kaynaklara sahip nesneler üzerinde kimlik doğrulama ve yetkilendirme işlemlerini gerçekleştirebilecek sade bir kimlik doğrulama ve yetkilendirme mekanizması önerilmektedir [59]. Yetki görevlerinden sorumlu olan merkezi olmayan bir sunucu ve anahtar dağıtımında rol oynayan başka sunucu tarafından ağa dahil olan kısıtlı kaynaklara sahip olan cihazlar birbirleriyle güvenli bir kanal aracılığıyla haberleşebilmektedir. Bu mekanizmalar, IoT cihazının farklı güvenlik seviyelerine hitap edebilmek için diğer standart teknolojilerle optimize edilerek kullanılmaktadır. Sunulan mekanizma kimlik doğrulama yöntemi olarak genişletilebilir. Kimlik Doğrulama Protokolü-Ön Paylaşımlı Anahtar (EAP-PSK - [60]) kullanımına dayanmaktadır ve bu protokol kullanılarak ağların ölçeklenebilirliği sağlanabilmektedir. Belirlenen senaryonun uygunluğunu değerlendirmek için IETF ACE tarafından belirlenen alternatif senaryolarla karşılaştırma yapılarak sonuçları incelenmiştir.

5. CONTİKİ OS

2002 yılında Adam Dunkels yönetiminde C programlama diliyle geliştirilmiş açık kaynaklı bir işletim sistemidir [61]. Adam ve ekibi tarafından TCP/IP yığınının kısıtlı kaynaklara sahip cihazlara başarıyla uygulanmasından sonra geliştirilmiştir. Farklı özelliklere sahip nesneler internete bağlanmak istenmiş ve hızlı bir şekilde gömülü dünyaya yayılmış olan uIP (mikro IP) 4'ü geliştirilmiştir. Gelişmeler birbirini izlemiş ve 2003 yılında ilk Contiki tanıtılmıştır. Normal bir Contiki konfigürasyonu 2 kilobayta RAM ve 40 kilobayt ROM belleğe sahip bir mikro kontrolcü üzerinde çalışabilmektedir. Ayrıca işletim sistemi IPv4 ve IPv6 üzerinden haberleşmeyi sağlayabilmektedir.

Contiki işletim sistemi, kaynak kısıtlı, düşük güçlü kablosuz duyarga ağı düğümleri üzerinde çalışacak şekilde tasarlanmıştır ve yeni donanım platformlarına kolayca yerleştirilebilmesi için yapılandırılmıştır. Dolayısıyla, bileşenlerinin çoğunluğu tamamen platformdan bağımsızdır. Contiki'yi yeni bir platforma uygularken, tüm sistemi çalıştırmak için az sayıda platforma bağlı bileşenlerin değiştirilmesi gerekmektedir.

Açık kaynak kodlu Contiki işletim sistemi aşağıdaki dizinleri içermektedir:

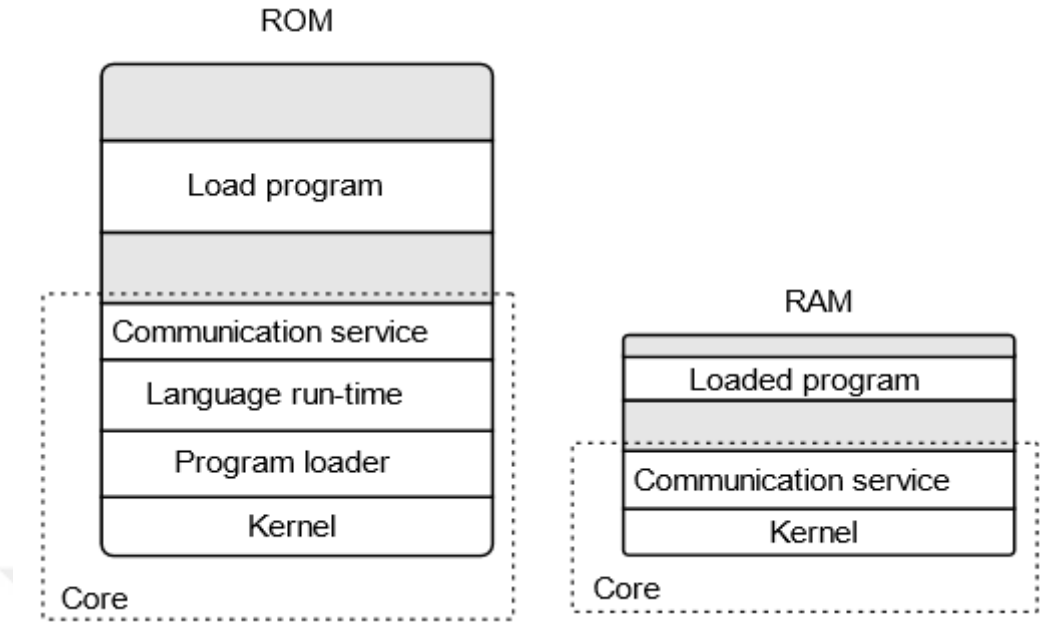
1. doc, Contiki işletim sistemine ait kaynak kodları barındırmaktadır. Dokümantasyonun bir kısmı çok iyi olmasına rağmen, bazı kısımları işletim sistemi geliştirme aşamasında olduğundan eksik kalmaktadır. Çeşitli platformlar için detaylı bilgiler vermektedir.
2. apps, Uygulama kütüphanelerini içermektedir. Bunlar genellikle örnekler dizininde yer alan uygulamalar tarafından kullanılmaktadır. Direkt olarak çalıştırılabilecek Contiki uygulamalarını barındırmaktadır.
3. examples, apps dizinindeki kütüphaneleri kullanan örnekleri kapsamaktadır. Örnek uygulamalar tipik olarak apps kütüphanelerinden süreçleri yürüten işlemleri tanımlarlar. Her bir örneğe ait dosyalar farklı dizinlerde bulunmaktadır.
4. core**,** işletim sistemi çekirdeği için kaynak kodlarını içermektedir. Sistemin farklı katmanları ait alt dizinler bulundurur. Genellikle bu dosyalara dokunmak gerekli değildir, ancak net alt dizindeki dosyalarda değişiklikler yapılarak uygulamalar geliştirilebilir.

5. cpu**,** işletim sisteminin üzerinde çalıştığı donamına ait mikro denetleyici dosyalarını içerir. Örneğin cpu/msp430/ dizini Sky mote için gereken mikro denetleyici sürücülerini içerir**.**
6. platform**,** platforma özgü yapılandırmayı ve donanımlara ait sürücüleri tanımlar.
7. tools**,** seri arabirim üzerinden platforma özgü donanımların yeniden programlanabilmesi için araçlar sağlamaktadır. Contiki'de yazılan uygulamanın simülasyonunu gerçekleştiren COOJA simülatörü bu dizin altında bulunmaktadır.

Contiki bileşenlerinin neredeyse tamamı "Protothread" olarak adlandırılır ve Adam'ın diğer önemli bir buluşudur. Hafıza maliyeti düşük ve yığın kullanmadığından kaynak kısıtlı cihazlar için özel olarak tasarlanmış işlem parçacığı olarak görülebilir [62]. Bunlar C makroları ile gerçekleştirilir. Düşük güçlü kablosuz aygıtlar için tasarlanan bir başka açık kaynaklı işletim sistemi, Kaliforniya Üniversitesi'nde geliştirilen TinyOS 'dur [63]. TinyOS Contiki 'ye kıyasla birçok benzerlik, avantaj ve dezavantaj göstermektedir. TinyOS bu tez kapsamı dışındadır ve yalnızca Contiki'nin mimarisi ve özellikleri aşağıda sunulmuştur.

5.1. Sistem Mimarisi

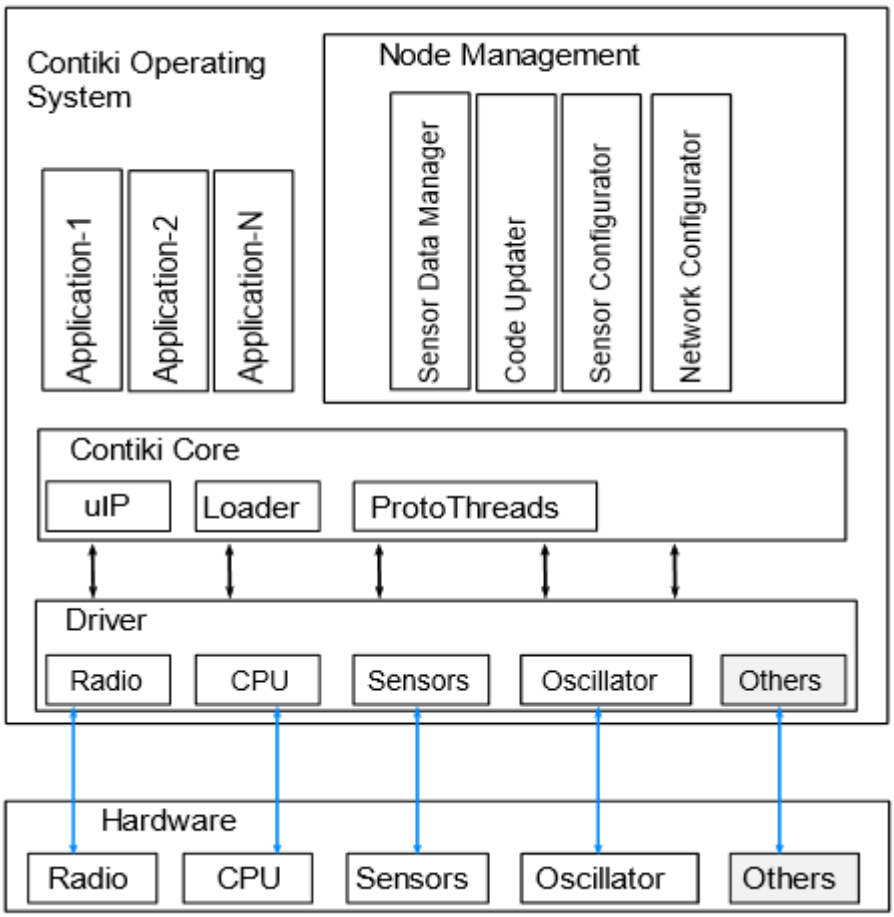
Contiki OS, modüler bir mimariye sahiptir ve çekirdekte olaya dayalı modeli takip ederken, bireysel süreçlere iş parçacığı ile hizmet sağlamaktadır. Çalışmakta olan bir Contiki sistemi, çekirdek, program yükleyici, kütüphaneler ve süreçler olarak 4 ana kısma ayrılabilir. Derleme süresi boyunca sistem Şekil 32'de gösterildiği gibi çekirdek ve yüklü programlar olarak iki bölüme ayrılmıştır.



Şekil 32. Çekirdek ve yüklü programların Gösterimi [61]

"Core" binary olarak derlenir ve cihazların belleğinde depolanır, genelde dağıtımından sonra değiştirilmez. Programlar, iletişim yığınını veya doğrudan depolama birimini kullanan program yükleyicisi tarafından yüklenir. Contiki çekirdeği, olayları çalışan süreçlere gönderen ve süreçlerin yoklama işleyicilerini çağıran sade bir olay zamanlayıcısıdır. Süreçlerin yürütülmesi gönderilen olaylar veya yoklama (polling) mekanizması tarafından tetiklenebilir. Çekirdek (kernel) bir olay işleyicisini tarifeledikten sonra bu olay için işlem önceliği vermez. Bu nedenle, olay işleyicileri (handler) işlem üstünlüğü alabilecek şekilde tasarlanmalıdır.

Contiki çekirdeği tarafından desteklenen iki tür olay vardır: senkron olaylar ve asenkron olaylar. Asenkron olaylar kuyruğa atılır ve daha sonra hedef süreçlere gönderilmek üzere bekletilir. Senkron olaylar ise bekletilmeden hedef süreçlere tarifelenmek üzere gönderilir. Contiki çekirdeğindeki yoklama mekanizması, her bir asenkron olay arasında tarifelenen yüksek öncelikli olaylardan oluşmaktadır. Durum güncellemeleri almak için donanıma yakın çalışan işlemler tarafından kullanılmaktadır. Tüm işletim sistemi olanakları (sensör veri işleme, iletişim, aygıt sürücüleri, vb.) hizmet şeklinde sağlanmaktadır. Şekil 33, Contiki OS mimarisinin blok diyagramını göstermektedir.



Şekil 33. Contiki OS Mimarisi [64]

5.2. Bellek Tahsisi ve Yönetimi

Contiki, düşük bellek kapasiteli gömülü sistemler için tasarlanmıştır. Standart bir Contiki sistemi 2 kilobayt RAM ve 40 kilobayt ROM kullanmaktadır; bu nedenle bellek tahsisi için etkili mekanizmalar kullanmaktadır. Contiki, bellek ayırıcı özel fonksiyon (mmem) ile tahsis edilen hafızayı serbest bırakmanın birincil görevi olan dinamik hafıza yönetimini ve dinamik bağlantıyı desteklemektedir [61].

5.3. Güç Yönetimi

Bu işletim sistemi, bir pil ile aylarca çalıştırılması gereken son derece düşük güç tüketen sistemler için tasarlanmıştır. Cihazlar uyku moduna geçme veya başka türde güç

tasarrufu eylemleri yapılarak yaşam ömürlerini uzatmalıdırlar. Bununla birlikte, Contiki, enerjinin nerede harcandığını görmek için bir sistem güç tüketimi tahmin mekanizması sağlamaktadır.

5.4. COOJA

COOJA, Contiki işletim sistemini çalıştıran duyarga ağlarını taklit etmek için tasarlanmış esnek bir Java tabanlı simülatördür [65]. Yalnızca dahili yazılımda değil aynı zamanda simüle edilmiş donanımda da farklılık gösteren duyarga düğümleri ağlarını taklit etmektedir. Cooja, simülatörün birçok parçasını kolayca değiştirilebileceği veya ek işlevsellikle genişletilebileceği için kullanışlıdır. Genişletilebilen örnek parçalar, simüle edilmiş radyo aracı, simüle düğüm donanımları ve giriş/çıkış eklentileri gibi birçok özellik içermektedir. Cooja'daki düğüm veri belleği, düğüm tipi ve donanım çevre birimleri olarak üç temel özelliğe sahiptir.

Örneğin, aynı türdeki cihazlar, ortak donanım çevre birimleri üzerinde aynı program kodunu çalıştırır. Böylece aynı türdeki düğümler benzer veri belleği ile başlatılır. Yürütme sırasında, düğümlerin veri belleği farklı girişlere bağlı olarak farklılık gösterecektir. Cooja şu anda Contiki programlarını iki farklı şekilde yürütmektedir. Program kodunu derlenmiş yerel kod olarak doğrudan ana CPU üzerinde veya derlenmiş program kodunu TI MSP430 emülatöründe çalıştırılarak programlar yürütülmektedir. Aynı zamanda üzerinde Contiki olmayan Java veya başka işletim sistemine sahip düğümleri de simüle etme yeteneğine sahiptir.

Tüm farklı yaklaşımların yanı sıra dezavantajları da vardır. Java tabanlı düğümler daha hızlı simülasyonlar sağlar, ancak "deployable" kod çalıştırmazlar. Dolayısıyla, dağıtılmış algoritmaların geliştirilmesi için kullanışlıdırlar. Benzetilmiş düğümler Java tabanlı düğümlere veya "native" kodu çalıştıran düğümlere kıyasla daha fine-grained işlem sağlar. Cooja, Java ortamında derlenmiş Contiki sistemine Java Native Interfance (JNI) çağrıları yaparak "native" kodu yürütmektedir. İşletim sistemi, Contiki çekirdeği, önceden seçilmiş kullanıcı işlemleri ve özel simülasyon sürücülerinde oluşmaktadır. Bu, simülasyon ve dağıtım arasındaki gecikmeyi en aza indirgeyen herhangi bir değişiklik yapmadan aynı kodu dağıtmayı ve benzetmeyi mümkün kılar.

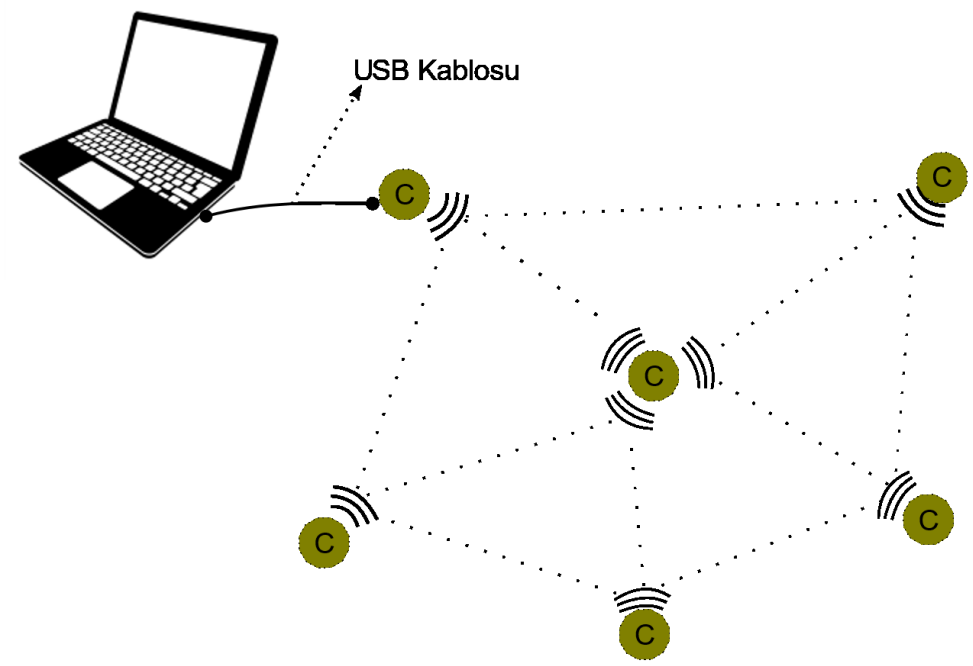
Java simülatörü, benzetimi yapılan düğümlerin belleği üzerinde tam denetime sahiptir. Dolayısıyla, simülatör her zaman Contiki süreç değişkenlerini görüntüleyebilir veya değiştirebilir ve simülatörden çok dinamik etkileşim olanakları sağlanabilir. JNI kullanmanın bir başka sonucu Contiki kodunu herhangi bir hata ayıklayıcı kullanarak JNI çağrısı gerçekleştirildiğinde Java simülatörü üzerinde hata ayıklamayı gerçekleştirmektir. Simüle edilen düğümlerin donanım çevre birimleri arabirimler olarak adlandırılır ve Java simülatörünün, gelen radyo trafiği veya LED gibi olayları algılayıp tetikleyebilmesini sağlamaktadır. Simülasyondaki arabirimler ve benzetilmiş düğümlerle olan tüm etkileşimler eklentiler aracılığıyla gerçekleştirilir. Kullanıcının simülasyonu başlatmasına veya durmasına izin veren simülasyon denetimi eklentiye örnek olarak verilebilir. Kullanıcılar bazı simülasyonların ihtiyaç durumuna göre arabirimleri ve eklentileri simülasyona kolayca ekleyebilir.

Cooja'daki her simülasyon, radyo dalgası yayılımını karakterize eden radyo modeli kullanır. Simülasyon ortamına yeni radyo modelleri eklenebilir. Simülasyon oluşturulduğunda radyo modeli seçilir. Bu, kullanıcının basit bir radyo modeli kullanarak ağ protokolü geliştirmesini ve daha gerçekçi bir model kullanarak test etmesini veya protokolü çok özel ağ koşullarında test etmek için özel olarak hazırlanmış bir modeli test etmesini sağlamaktadır. Cooja, girişim kullanan basit modeli ve simülasyon çalışması sırasında değiştirilebilen iletim aralığı parametresini desteklemektedir. Daha iyi radyo modelleri üzerinde devam eden çalışmalar Cooja'ya, radyo emici materyali destekleyen genel ışın izleme modelini sunacaktır [65].

Contiki olay tabanlı bir işletim sistemidir ve sistemdeki her işlenen olayın tamamlanmasına izin verir. Bu, duyarga düğümleri gibi kısıtlı belleğe sahip cihazlar için önemli bir özelliktir ve TinyOS gibi diğer işletim sistemlerinde de kullanılmaktadır. Cooja, yüklenen Contiki sistemini çağırarak bu özellikten yararlanır; böylece simüle edilen her düğüm sırayla yalnızca bir olayı gerçekleştirmektedir. Her Contiki sistemi, daima belirli bir donanım platformu için derlenmektedir. Platform, mevcut sürücüleri tutar ve gerçek donanımla nasıl iletişim kuracağını tanımlar. Örneğin MSP430 emülatöründe bir düğüm taklit edildiğinde Contiki işletim sistemi MSP430 işlemci mimarisi için özel olarak derlenmektedir.

5.5. Contiki'de IPv6 Ağı

Contiki, istemci ve sınır yönlendirici uygulamaları da dahil olmak üzere 6LoWPAN'ı, RPL'i ve bilgisayarın kablosuz ortama (kanal) IP paketleri göndermesini sağlayan SLIP (ağ katmanı yönlendirmesi) protokolünü kullanan örnek uygulamalar sunmaktadır. Bu uygulamaların aktif durumda olduğu senaryo varsayıldığında, Contiki Şekil 34'teki gibi bilgisayar ve düğümler tarafından oluşturulan bir ağın kurulmasına olanak sağlamaktadır.



Şekil 34. Bir bilgisayarı kablosuz duyarga ağına bağlama örneği

Bilgisayar, 6LoWPAN kullanarak IPv6 destekli düğümlere erişmek için bir USB bağlı gömülü cihazı kullanmaktadır. Ana bilgisayar IP paketlerini, SLIP protokolü vasıtasıyla seri giriş olarak aldığı paketleri kablosuz kanala aktaran cihaza göndermektedir ve tersi olarak da bu cihazdan seri bağlantı üzerinden paketleri almaktadır. Bu nedenle bilgisayar, doğru biçimlendirilmiş 6LoWPAN paketleri göndermeyi bilmelidir çünkü SLIP protokolü burada herhangi bir paket düzenleyici rolü almaz.

Yukarıdaki bilgilerden de çıkarılacağı gibi iki farklı ağ arasında yönlendirici olarak görev yapmak için tipik bir kısıtlı kaynaklara sahip cihazdan daha güçlü bir aygıt kullanmak

gereklidir. Şekil 34’teki örnek dikkate alındığında dönüşümü sağlamak için temel gereksinimler yönlendiricinin Ethernet arabirimine ve 802.15.4 kablosuz radyoya sahip olmasıdır. SLIP arayüzü aracılığıyla sınır yönlendiriciye bağlanılması IEEE 802.15.4’den gelen verilerin anlamlı bir şekilde okunabileceğini göstermektedir. Contiki'de bulunan sınır yönlendirici uygulaması RPL tarafından oluşturulan ağaç tabanlı bir topolojide kök düğüm görevi görmektedir ve IPv6 paketlerini, yerel biçimleri ile kablosuz kişisel alan ağlarında kullanılan 6LoWPAN formatına dönüştürmeye yardımcı olmaktadır. Seri hat üzerinden 2 cihaz arasındaki IP trafiğini gerçekleştirmede kullanılan SLIP ile bir bilgisayara bağlanır. SLIP, ana makine tarafında bir sanal ağ arabirimi oluşturur ve IP trafiğini seri hattın diğer tarafına aktarmak ve bu seri hattın diğer tarafından IP paketlerini almak için SLIP’i (seri hat İnternet protokolü) kullanır.

6. YAPILAN ÇALIŞMA

Bu tez çalışmasında, IETF 6TiSCH ağlarında yer alan bağlantı katmanı güvenli önyükleme protokolüne bir eklenti sunulmaktadır; burada kimlik doğrulama anahtarları, etkili bir kimlik doğrulama ve önyükleme sürecini etkinleştirmek için IoT ağının güvenilir düğümlerinde dağıtılmaktadır. Dağıtık bir yaklaşım kullanılarak standart IETF 6TiSCH kimlik doğrulama protokolünün iletişim maliyetinin azaltılması ve kimlik doğrulama parametrelerinin IoT ağının kenarında tutularak ağın enerji verimliliğinin artırılması hedeflenmiştir.

Çalışma grubu tarafından önerilen yaklaşımda 6TiSCH ağına katılmak isteyen bir cihazın güvenli önyükleme konfigürasyonunu sağlamak için gereken minimum mekanizmalar açıklanmaktadır [55]. Yapılandırmanın amacı, merkezi bir yetkilendirme servisi kullanılarak, ağa dahil olmak isteyen cihaz ile bu cihazı ağa dahil etmede yardımcı olan düğüm arasında güvenli bir oturum oluşturabilmektir. Belirtilen adımların gerçekleşmesi için bağlantı katmanı anahtarları ayarlanarak cihaz ağa senkronize olmalıdır. Bu sayede cihazlar birbirleriyle güvenli olarak haberleşebilir. Bağlantı katmanında gerçekleşen bu olaylara ek olarak üst katmanlarda da ilave güvenlik mekanizmaları eklenebilmektedir.

Bağlantı katmanı çerçeveleri güvenli hale getirilirken, IEEE 802.15.4'te tanımlanan standart güvenlik mekanizmaları kullanılabilmektedir. Bağlantı katmanındaki kimlik doğrulaması, 802.15.4 başlığı da dahil olmak üzere tüm çerçeveye uygulanmalıdır. Şifreleme ise, 802.15.4 standardının oluşturduğu veri yüküne uygulanabilmektedir. Standart iki farklı şifreleme anahtarını (K1, K2) kullanarak doğrulama işlemini gerçekleştirmektedir. K1 anahtarı, işaretçileri doğrulamak için kullanılmaktadır (ilk başta tüm cihazlara atandığı varsayılmaktadır). K2 anahtarı ise, Veri ve Onay çerçevelerini doğrulamak ve şifrelemek için kullanılmaktadır. Bu anahtarlar önceden cihazlara yüklenebilir veya anahtar dağıtım aşamasında cihazlara bildirilebilir. Ağa dahil olmak isteyen düğüm, anahtarların önceden yapılandırıldığına bağlı olarak farklılık göstermektedir. K1 ve K2 anahtarları önceden yapılandırılmışsa sisteme dahil olmak isteyen düğümün, bu anahtarları öğrenmek için bir anahtar dağıtım mekanizmasına ihtiyacı yoktur. K1 anahtarı bilinmesine rağmen K2 anahtarı bilinmiyorsa, düğüm komşu cihazlardan gelen işaretçileri K1 anahtarını kullanarak doğrulamaktadır ve K2'yi öğrenmek için anahtar dağıtım mekanizmasına güvenmektedir

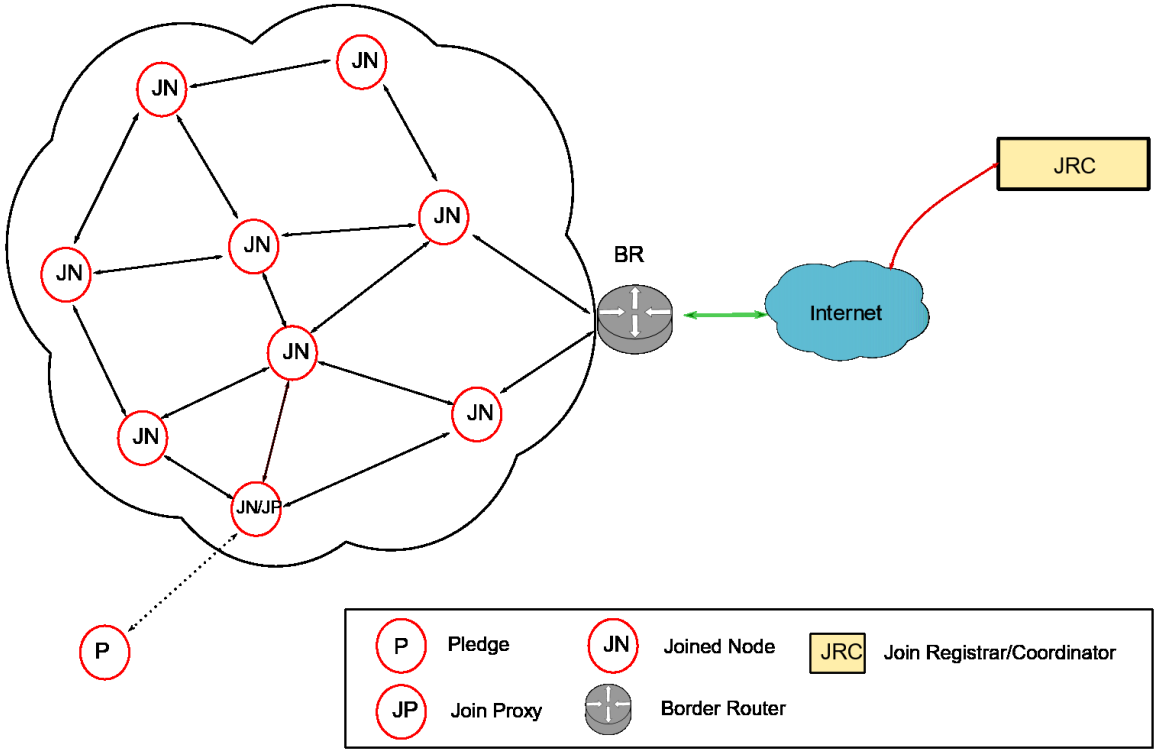
(tezde incelenen yapı bu şema üzerinedir). Merkezi/Dağıtık kimlik doğrulama öğesi tarafından gelen isteklere cevap olarak bu anahtar değeri ve diğer bazı parametreler cihazlara gönderilmektedir. Düğüm, anahtar değerini doğruladıktan sonra sisteme dahil olmaktadır ve artık komşularıyla haberleşebilmektedir.

6TiSCH ağındaki tüm çerçeveler, bağlantı katmanı güvenliğini kullanmalıdır. Güvenlik seçenekleri verilerin kimlik doğrulamasını ve şifrelemesini içermektedir. Aday düğüm, EB mesajını ortak anahtar değeri kullanarak anlamlı hale getirir. Link katmanı çerçeveleri, 16 baytlık anahtar ve şifrelemede büyük bir öneme sahip 13-baytlık Nonce değerlerinden oluşmaktadır. Şekil 35'te gösterilen bu değer, EB gönderen cihazın adresinden (8 bayt) ve ASN (5 bayt)'den oluşmaktadır.

Şekil 35. Bağlantı katmanı nonce yapısı

K1 ve K2 anahtarları önceden yapılandırılmadıysa, düğüm, işaretçileri kullanabilmek için OSCOAP protokolünden faydalanmaktadır. Anahtar dağıtım aşamasında bu protokol kullanılarak bağlantı katmanı anahtar değerleri cihazlara gönderilmektedir. Cihazlar elde ettikleri bu anahtar değerleriyle işaretçileri, verileri ve onay çerçevelerini şifrelemektedir/doğrulamaktadır.

IETF 6TiSCH grubu IoT cihazlarını önyüklemek için Şekil 36'da gösterilen kimlik doğrulama mekanizmasını sunmaktadır. Mevcut öneri, JRC olarak adlandırılan kimlik doğrulamayı etkinleştirmek için merkezi bir yapıyı ve ağa dahil olmak isteyen düğümlere yardımcı olan bu düğümlerle ortak özelliklere sahip yardımcı düğümlerden oluşmaktadır.



Şekil 36. Ağa Dahil Olma İşlemine Genel Bir Bakış

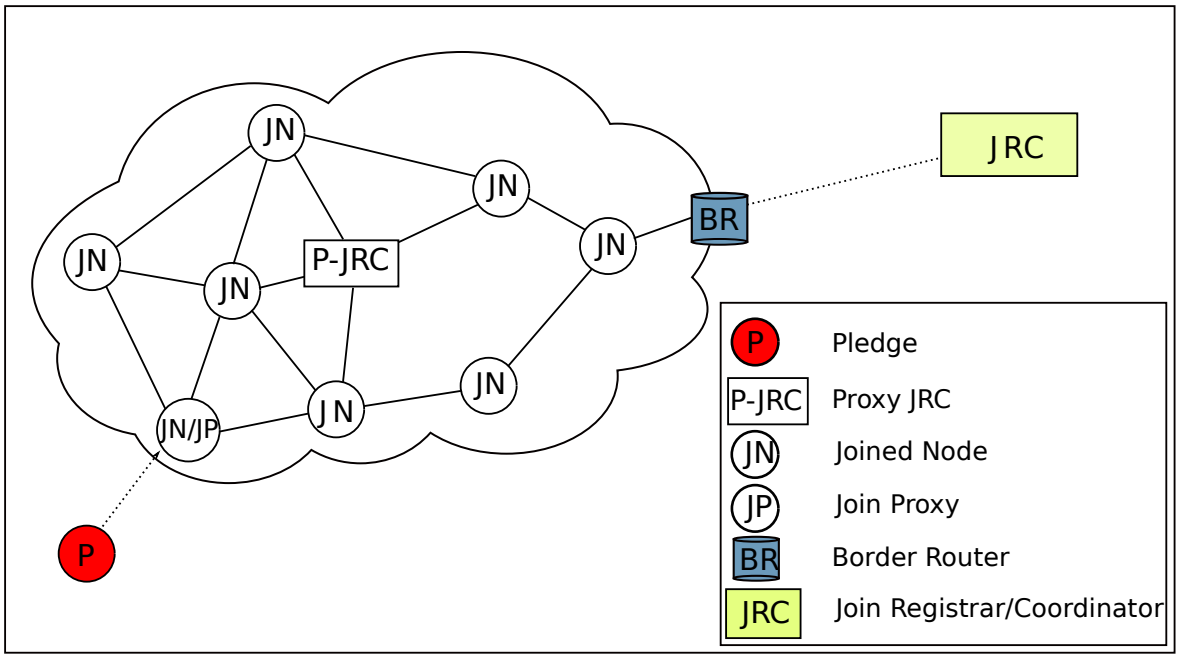
6TiSCH ağına katılmak isteyen bir cihazın önce ağa senkronize edilmesi gerekmektedir. Senkronize işlemi gerçekleştikten sonra haberleşmede önemli rolleri olan kanal atlama sırası, slot çerçeve süresi, slot zamanlaması gibi parametreleri öğrenmiş olmaktadır. Bu adımın ardından, cihaz ağ için uygun kimlik doğrulama bilgilerine sahipse, katılım işlemi kablosuz cihaz tarafından başlatılır. Bu noktada bağlantı katmanı anahtarlarını yapılandırmak için ağ ile etkileşimde bulunulması beklenmelidir. Ancak o zaman düğüm DTLS [66] veya OSCOAP [67] kullanan cihazlarla uçtan uca güvenli bir oturum oluşturabilir. Uygulama gereksinimleri bilinirse, cihaz, gerektiği gibi ek kaynaklar talep etmek için veya ağ değişiklikleri olarak yeniden yapılandırılması için eşleriyle etkileşime girmektedir. Ağa katılma işleminin sonucu olarak, aday cihaz bir veya daha fazla bağlantı katmanı anahtarını ve isteğe bağlı olarak geçici bir ağ tanımlayıcısını beklemektedir.

6TiSCH'ye katılma işlemi, gelişmiş işaretçilerin kimliğini doğrulamak için önceden paylaşılan bir anahtar, JRC'ye aygıtın kimliğini doğrulamak için önceden paylaşılan anahtar veya sertifika ve JRC yanıtını doğrulamada katılımcı düğüm için paylaşılan bir anahtar veya sertifika gibi ön koşullara sahiptir. 6TiSCH ağına katılmak isteyen düğüm Pledge olarak adlandırılır. Ağın bir parçası haline gelen düğüme JN denir. Katılma isteğini kesintisiz olarak

JRC'ye aktaran düğüme JP denir. JRC, katılan düğümlerin kimlik doğrulamasından ve yetkilendirilmesinden sorumludur. 6TiSCH ağına katılma süreci ayrıca, önyüklemeyi gerçekleştirecek şekilde yetkilendirmek için denetim parametresi ve [68] 'da açıklandığı gibi önyükleme işlemi için üretici tarafından kurulmuş bir sertifikayı tanımlamaktadır. Önerilen 6TiSCH katılım sürecindeki aktörler Şekil 36'da gösterilmektedir.

6TiSCH önyükleme modelinde, Pledge'in önceden paylaşılan anahtar veya katılmak istediği ağ için geçerli bir sertifikası olduğu varsayılmaktadır. Ağa dahil olmak isteyen cihaz ilk önce kanal atlama sırası, zamanlama parametreleri gibi değişkenleri öğrenebilmesi için gelişmiş işaretçileri dinlemelidir. Kablosuz cihaz ortamdan işaretçiyi aldığında, önce bu mesajı bilinen bir anahtarla doğrular. Bu adım, alınan EB'nin bir 6TiSCH işareti olduğunu onaylamak için tasarlanmıştır. İşaret, geçerli bir 6TiSCH mesajı olarak doğruladıktan sonra Pledge ağa senkronize olur, fakat ağ tarafından kimliği doğrulanmaz. Veri gönderemez ve alamaz. Bir sonraki adım olarak Pledge, CoAP protokolünü kullanarak EB aldığı düğüme bir katılma isteği göndererek katılma sürecini başlatmalıdır. Ağa dahil olmada yardımcı görev alan JP, kablosuz cihazdan aldığı katılma isteğini JRC'ye iletmekten sorumludur. JP tarafından JRC adresi bilindiği için gelen katılım istekleri kesintisiz olarak OSCOAP protokolü aracılığıyla ana sunucuya yönlendirilir. Yönlendirme işlemi boyunca Pledge JP ile JRC arasındaki düğümleri ve JRC'yi bilmesine gerek yoktur. Ağa katılım istekleri uygulama katmanına çıkmadan ağ katmanında çoklu sekme üzerinden gönderilmektedir.

JRC önceden paylaşılan anahtarı veya önceden yüklenmiş sertifikayı kullanarak JP'nin yollamış olduğu katılım isteğini doğrulamaktadır ve katılma isteğinin durumuna göre JP'ye mesaj göndermektedir. Pledge**,** JP'den aldığı yanıt mesajının geçerli bir JRC'den geldiğinden emin olmak için JRC'nin kimlik bilgileriyle yanıt mesajını doğrular. Doğrulama işleminin sonucuna göre aday düğüm ağa dahil olabilir/olmayabilir.



Şekil 37. Vekil Sunucu Tabanlı Kimlik Doğrulama Altyapısı

Bu çalışmada 6TiSCH önyükleme modeli, JRC için kimlik doğrulama parametrelerini içeren ve JRC adına düğümleri doğrulayarak ağa dahil edilmesini sağlayan dağıtık bir önyükleme modeli oluşturan vekil sunucu tabanlı JRC (P-JRC)'yi içerecek şekilde genişletilmiştir. P-JRC cihazı JN'lerden seçilebilir veya ek donanıma sahip önceden yapılandırılmış bir cihaz olabilir. Bu yaklaşımın ana motivasyonu, ağdaki önyükleme ve kimlik doğrulama parametre güncelleme işleminin haberleşme yükünü azaltmaktır. Şekil 37'de gösterildiği gibi ağ içerisinde tek bir P-JRC uygulanmaktadır. Birden fazla P-JRC düğümüne sahip olmak da mümkündür fakat simülasyon sürecindeki kolaylık açısından, model şu anda ideal bir konumda bulunan tek P-JRC düğümünü kullanmaktadır. Önerilen önyükleme modelinde ağa dahil olma işlemi, standart 6TiSCH önyükleme modelinde olduğu gibi Pledge tarafından ağa katılmaya yardımcı olan JP'ye gönderilen katılma isteği ile başlar.

Pledge düğümüne ağa katılmada yardımcı olan JP, düğümlerden gelen katılma isteklerini kesintisiz olarak merkezi JRC biriminin hedef IPv6 adresine iletir. Mesaj ağ boyunca dolaşırken P-JRC öğesi, JP'nin yönlendirme yolundaysa katılma isteği P-JRC'ye gönderilmektedir. Bu durumda P-JRC, doğrulama işlemini başlatmak için Pledge'in sahip olduğu önceden paylaşılan anahtarı kullanmaktadır. Ağa katılmak isteyen düğüm başarıyla doğrulandıktan sonra P-JRC, önyükleme sürecini tamamlamak için kimlik doğrulama iletisini JP öğesine gönderir. JRC öğesi, kimlik doğrulama belirteçlerini yenilemek için

önceden tanımlanmış aralıklarla P-JRC'ye kimlik doğrulama parametrelerinin güncel hallerini göndermektedir. Bu, merkezi olarak düğümlerin kimlik doğrulama belirteçlerinin doğrulanması ve güncellenmesine kıyasla daha düşük bir iletişim yüküne neden olmaktadır.

Düşük Güçlü ve Kayıplı ağlar (LLN'ler) için IPv6 yönlendirme protokolü olan RPL, ağ içerisindeki cihazlardan merkezi bir kontrol noktasına ve merkez kontrol noktasından LLN içindeki cihazlara kadar çoklu trafiği destekleyen DODAG tabanlı bir topoloji oluşturur. Çalışmada düşük güçlü aygıtlar ağı, düğümlerin simüle edilen ağ ile yerel alan ağı (LAN) arasında sınır yönlendirici olarak yapılandırılan kök düğüm aracılığıyla JRC'ye kendilerini doğrulayabileceği RPL yönlendirme protokolü kullanılarak oluşturulmuştur. Şekil 37'de önerilen mekanizmada JRC'nin güvenilir bir yapı olduğu, her cihaz için anahtar(simetrik) değerinin ve kayıt aşamasında cihaz tarafından üretilen dinamik kimlik bilgisinin (DID) JRC'de tutulduğu varsayılmıştır.

Önerilen kimlik doğrulama süreci, cihazın kaydı ve kimlik doğrulamasından sorumlu olan iki basamaktan oluşmaktadır. Kimlik doğrulama işlemi için kullanılan kısaltmalar aşağıda listelenmiştir:

ID_u: Cihazın kimlik numarası (Cihazın EUI64 adresi olabilir);

DID_u : Cihazın dinamik ID'si;

PWD_u : Cihazın şifresi;

$H(PWD)_u$: Cihaz şifresinin özüt değeri;

GK_{JRC-u} : Cihaz ile JRC arasındaki global simetrik anahtar;

PK_{JRC-u} : Kimlik doğrulamasını takiben JRC ve cihaz arasındaki simetrik özel anahtar;

TS : Zaman damgası;

$H(.)$: Tek yönlü özüt fonksiyonu (SHA-1);

$||$: String birleştirme işlemi;

Güvenlik gereksinimlerinin önemli yapı taşı olan veri bütünlüğü ve veri doğrulaması; herhangi bir veri setinin ilk halini koruduğu ve hiçbir değişikliğe uğramadığı anlamına gelmektedir. Özüt[5] (hash) fonksiyonları kullanılarak dosya ve metin gibi veri bloklarının özgünlüğü koruma altına alınmaktadır. Özüt fonksiyonları, herhangi bir uzunluktaki verileri

[5] SHA-1, 160 bitlik çıktı üreten tek yönlü özetleme fonksiyonudur.

belirli bir özüt algoritmasına tabi tutarak, sabit uzunlukta çıktı üreten fonksiyonlardır. Genellikle veri bütünlüğünü garanti etmesi, hızlı olması, sabit uzunlukta çıktı vermesi, dosya boyutunun alınan özütü etkilememesi ve yüksek performanslı bir haberleşme sağlaması bu yaklaşımın üstünlükleridir. Güvenli Özüt Algoritması (SHA - Secure Hash Algorithm), bir tür mesaj doğrulama yöntemi olmakla birlikte, uygulandığında programların ve dosyaların bütünlüğünü güvence altına almaktadır. Bu teknikte girdi değerinden sabit uzunlukta özüt değeri hesaplamak kolaydır fakat aynı çıktı (özüt) değerini veren girdi değerini üretmek zordur. Giriş değerinde bir bitin değişmesi, ortalama olarak özüt değerindeki bitlerin yarısını değiştirmektedir. Verilen bir özüt değeri ile aynı değeri veren farklı bir girdi değerini bulmak matematiksel olarak çok zordur.

Güvenli Özüt Algoritması SHA-1, SHA-256, SHA-384 ve SHA-512 olmak üzere dört farklı özüt algoritmasına ayrılmaktadır. Bu tez kapsamında, NSA tarafından tasarlanan şifreleme algoritmaları içerisinde en yaygın olarak kullanılan SHA1 algoritması kullanılmıştır [69], 70]. Bu algoritma ile sadece şifreleme işlemi yapılır; şifre çözme işlemi yapılamaz. Şifreleme için kullanılan metin belli işlemlerden geçerek 160 bitlik özütler oluşturur. SHA1, ön işleme ve özüt hesaplama olmak üzere iki aşamadan oluşmaktadır. Ön işleme aşamasında mesajlar 512 bitlik bloklara ayrılır ve gerekirse son bloğun uzunluğu 512 bite tamamlanır. Özüt hesaplaması bu mesajı fonksiyonlar, sabitler ve kelime işlemleri ile bir özet değerleri serisini üretmek için kullanmaktadır. Ön işleme, özüt hesaplaması başlamadan önce yer almalıdır ve mesajı (M) doldurma, doldurulan mesajı bloklara ayırma, başlangıç özet değerini ayarlama olarak üç aşamadan oluşmaktadır.

SHA-1, f_0, f_1,……, f_{79} gibi bir dizi mantıksal fonksiyonlar kullanmaktadır. $0 \leq t \leq 79$ olmak üzere her f_t fonksiyonu üç adet 32 bitlik kelimeler olan x, y, ve z üzerinde işlem yapar ve 32 bitlik bir çıkış üretmektedir. f(t; x, y, z) fonksiyonu aşağıdaki gibi tanımlanır.

$f(t;x,y,z) = (x \wedge y) \vee ((\neg x) \wedge z)$ (0 <= t <= 19)

$f(t;x,y,z) = x \oplus y \oplus z$ (20 <= t <= 39)

$f(t;x,y,z) = (x \wedge y) \vee (x \wedge z) \vee (y \wedge z)$ (40 <= t <=59)

$f(t;x,y,z) = x \oplus y \oplus z$ (60 <= t <= 79)

SHA-1'de sabit kelimeler seksen adet 32 bitlik kelimeden oluşan K_t: K_0, K_1,……, K_{79} dizisini kullanmaktadır. Onaltılık olarak bunlar:

$K(t) = 5a827999, \quad (0 \leq t \leq 19)$

$K(t) = 6ed9eba1, \quad (20 \leq t \leq 39)$

$K(t) = 8f1bbcdc, \quad (40 \leq t \leq 59)$

$K(t) = ca62c1d6, \quad (60 \leq t \leq 79)$

Ek bitlerle doldurmanın amacı, algoritmaya bağlı olarak mesajın 512 bitin katı olduğundan emin olmaktır. Mesaj ilave ekten sonra, özüt hesaplanmasının başlayabilmesi için 16 adet 32 bitlik bloğa bölünmesi gerekmektedir. Giriş bloğu 16 adet 32 bit kelime ile ifade edilebildiği için, i mesaj bloğunun ilk 32 biti $M_0^{(i)}$, bir sonraki 32 biti $M_1^{(i)}$ ve en son $M_{15}^{(i)}$ olarak ifade edilir.

Özüt hesaplaması başlamadan önce, uygulanacak algoritma için başlangıç özüt değerinin ($H^{(0)}$) hazırlanması gerekmektedir. $H^{(0)}$'ın boyutu ve içerdiği kelime sayısı mesaj özetinin boyutuna bağlıdır. SHA-1 için $H^{(0)}$ başlangıç özüt değeri aşağıda belirtilen beş adet 32 bitlik kelime içermektedir.

$H0^{(0)} = 0x67452301$

$H1^{(0)} = 0xefcdab89$

$H2^{(0)} = 0x98badcfe$

$H3^{(0)} = 0x10325476$

$H4^{(0)} = 0xc3d2e1f0$

Hesaplama, her biri 32 bit olan beş adet değişkenden ve seksen adet 32 bitlik mesaj dizisinden oluşan iki tampon (buffer) kullanılarak ifade edilmektedir. Özüt değerinin kelimeleri, H0, H1, H2, H3, H4 ve 80 adet mesaj dizisinin kelimeleri ise W (0), W (1), ..., W (79) olarak etiketlenmiştir. Ana algoritma, her 512-bit mesaj bloğunun durumunu değiştirmek için kullanır. Bir mesaj bloğunun işlenmesi doğrusal olmayan bir fonksiyon, modüler toplama ve sola kaydırma üzerine 80 benzer işlemden oluşmaktadır. Bu işlemler sonucunda 160 bitlik özet hali hesaplanır. Bu değer ilerde kullanılmak üzere saklanmaktadır.

SHA1'de çakışmaların meydana gelmesi kolay değildir. SHA-256, SHA-1'den çok daha fazla işlem gerektirmektedir, ancak benzer yapıdadır. Bu nedenle işlem maliyetini azaltmak için yapılan çalışmada SHA1 seçilmiştir. Doğrulama işlemlerinde kayıt merkezi olan güçlü cihazlar, kullanıcıların gerçek kimlik bilgilerine direkt olarak erişebilmektedir. Bu özellik kötü niyetli kişilerin içerden saldırı yaparak kullanıcıların kimlik bilgisine

erişmesine olanak sağlamaktadır. Meydana gelen bu açığı kapatmak için özüt fonksiyonları kullanılarak kimlik bilgilerinin şifreli halleri veri tabanlarında saklanmaya başlanmıştır. Saldırgan kimlik doğrulama sunucusunu fiziki olarak ele geçirse dahi tek yönlü özüt fonksiyonları kullanıldığından tersine bir işlem yapıp kullanıcıların gerçek kimlik değerlerini elde edemez. Bu nedenle tez kapsamında kullanılan kimlik doğrulama elemanları, kullanıcıların (düğüm) tek yönlü özüt fonksiyonlarının sağlamış olduğu basitlikten yararlanarak oluşturdukları kimlik değerlerini kendi bünyelerinde barındırmaktadırlar. Bu özellik sayesinde elemanlar ele geçirilse bile saldırgan gerçek kimlik parametrelerini elde edemez.

6.1. Kayıt Aşaması

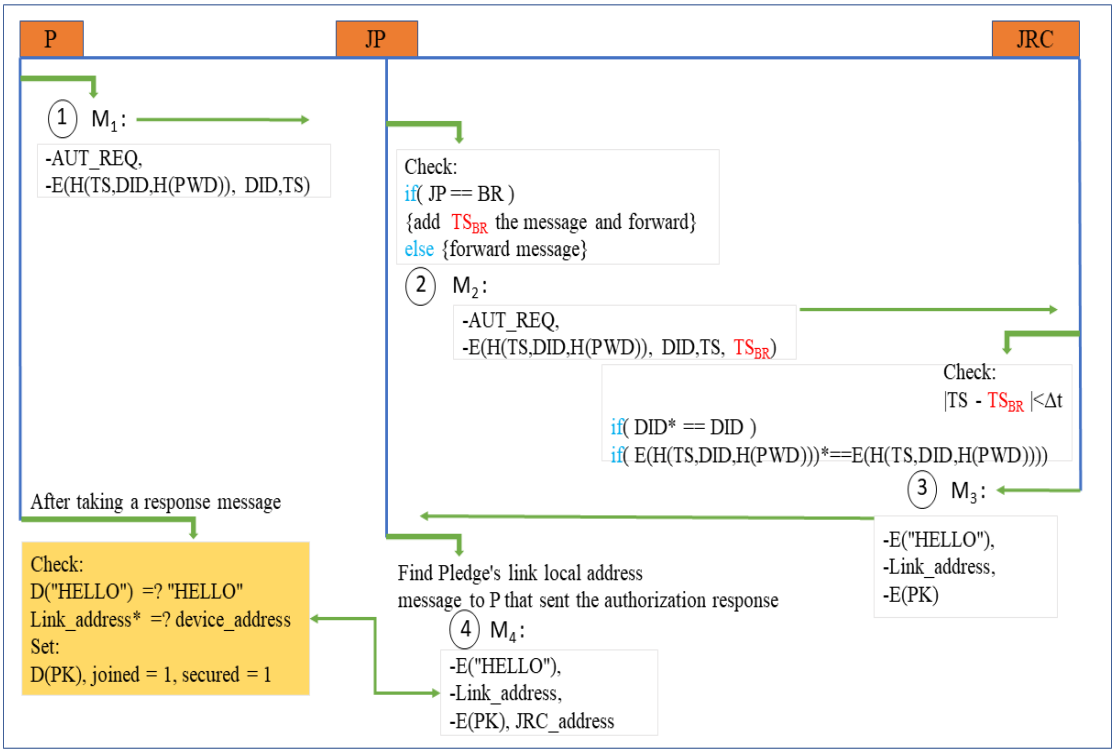
Bu aşamada, cihaz tarafından 8 baytlık kimlik numarası ve 20 baytlık şifre parametreleri üretilmektedir. Üretilen kimlik bilgisi her cihaza özel olan bağlantı katmanı adresinden oluşturulur. Daha sonra bu kimlik bilgisi ile şifrenin özüt hali birleştirilerek yeni bir özüt değeri hesaplanır. Hesaplanan bu yeni değer, dinamik kimlik bilgisini oluşturur. Ayrıca JRC cihazı için güvenli bir simetrik anahtar değeri üretilir ve bu işlem sırasında kayıtlı cihazların her birinde saklanır. Bu işlem, kimlik doğrulama düğümünün JRC yanıtını doğrulamasını sağlar.

$$(DID_u) = (\mathrm{H}\ (\ ID_u \ ||\ H(PWD)_u\ )) \tag{2}$$

Elde edilen dinamik kimlik bilgisi ve şifrenin özüt değeri güvenli bir şekilde JRC'de kayıt altına alınır. Her iki tarafın bu aşamada birbirlerine güvendiği varsayıldığından herhangi bir saldırı durumu dikkate alınmaz. Cihazlarda kimlik değerlerinin özüt hali saklanmasından dolayı, saldırgan JRC'ye saldırması durumunda cihazların gerçek kimlik bilgilerini ele geçiremez. JRC bu aşamada cihaza diğer aşamalarda kullanabilmesi için simetrik bir anahtar değeri verir. Öte yandan JRC, gelen kimlik doğrulama mesajını, gelen DID_u, Aday cihaz için saklanan kimlik bilgilerinden üretilen DID ile karşılaştırarak doğrulayabilir.

6.2. Kimlik Doğrulama

6TiSCH ağına katılmak isteyen cihaz bu bölümde ayrıntılı olarak belirtilen süreçleri izlemelidir. Daha önce de belirtildiği gibi 6TiSCH protokolü, önceden yapılandırılmış veya ağa dahil olma işlemi sırasında elde edilen bir anahtar kullanılarak kimlik doğrulama adımını gerçekleştirir. Cihaz, eşleştiği ağdan aldığı EB'yi bilinen bir anahtar değeriyle doğruladıktan sonra kanal atlama sırası, slot çerçeve süresi gibi ağ parametrelerini öğrenir. Aday cihaz (P), yerel IPv6 adresini yapılandırır ve senkronize edildiği düğüme (JP) bildirir. Bu adımdan sonra cihaz, Şekil 36'da açıklandığı gibi katılma sürecini başlatır. Bu yazıda önerilen ve açıklanan katılım süreci, [55] 'te tanımlanan önergenin biraz değiştirilmiş halini uygular.

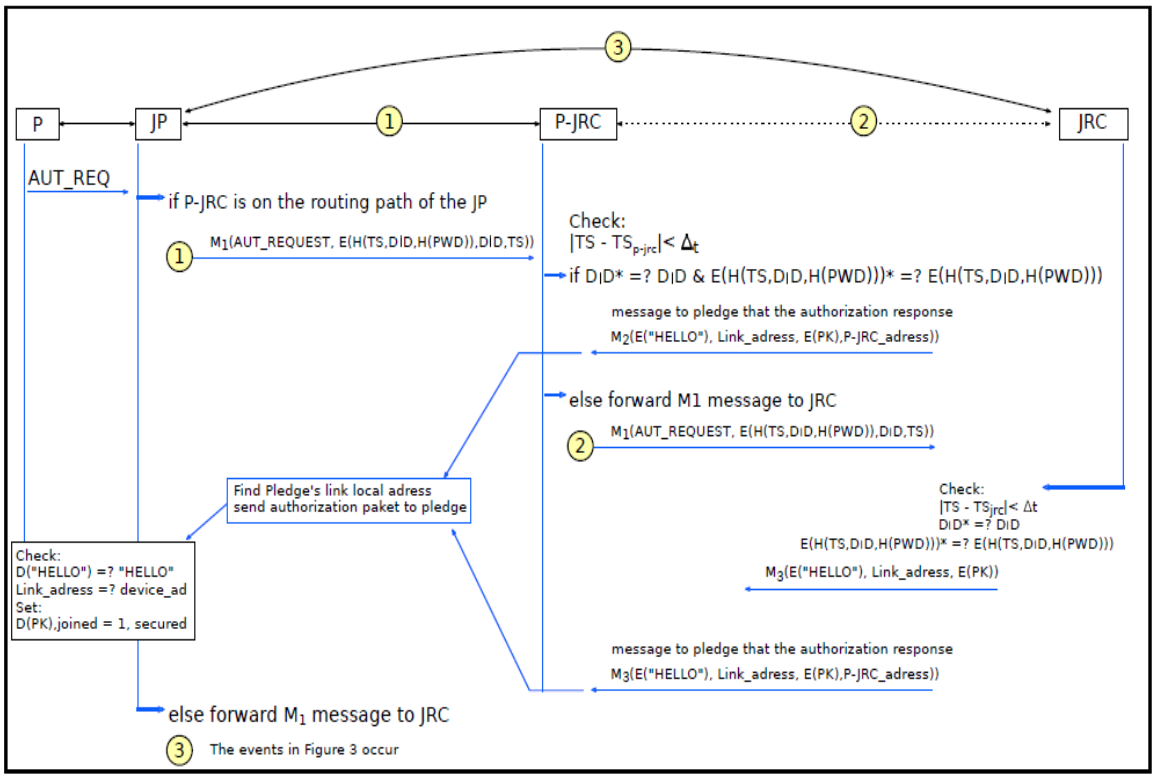


Şekil 38. Kimlik Doğrulama Modeli

6TiSCH ağına katılmak isteyen düğüm, JP cihazı aracılığıyla JRC'ye uçtan-uca şifrelenmiş bir istek mesajı gönderir. Katılma İsteği mesajının şifrelenmesi önceden paylaşılan bir anahtar kullanılarak veya cihaza yüklenmiş sertifikalar yoluyla yapılabilir. Önceden paylaşılan anahtarları kullanan kimlik doğrulama işlemi, cihaz ile JRC arasında

daha az mesaj iletilmesini gerektirir. Böylece, önceden paylaşılan anahtar tabanlı kimlik doğrulama işlemi sertifika tabanlı kimlik doğrulama ile karşılaştırıldığında daha kolay hale gelir. Öte yandan, düğümün önceden paylaşılan anahtarı ele geçirilirse, saldırgan JRC'den cihaz için gönderilen kimlik doğrulama parametrelerini edinebilir. Bu saldırı ile ağ güvenilmez bir hale gelebilir. Tabii ki, bu fiziksel olarak tehlikeye düşmüş bir düğüm için de önceden yüklenmiş sertifikaların ele geçirilmesiyle mümkün olabilir. Önceden paylaşılan anahtarlar kullanıldığında JRC, ağa katılan her düğüm için ayrı anahtarlar bulundurmalıdır.

Cihaz, bağlantı yerel IPv6 adresini aldıktan ve komşuluk tablosuna JP için kayıt oluşturduktan sonra JP düğümüne istek mesajını gönderir. Bu istek paketi 6TiSCH standardına göre EB tarafından belirtilen paylaşılmış alana (shared slot) iletilir. JP, bu paketi JRC'ye iletir. JRC, Şekil 38'de gösterildiği gibi anahtar veritabanında DID_u, DID_u* ve zaman damgası gibi parametrelerden yararlanarak mesajı gönderen cihazların kimlik bilgilerini doğrular. Kimlik bilgileri geçerliyse JRC, JP öğesine ağa katılmak isteyen düğüme özel kimlik doğrulama belirteçleriyle birlikte bir "HELLO" mesajı gönderir. JP öğesi, yanıt mesajından aday cihazın adresini elde eder ve şifrelenmiş mesajı yeni kimlik doğrulama bilgileriyle birlikte bu cihaza iletir. Cevabı aldıktan sonra cihaz, önceden JRC için saklanan kimlik bilgilerini kullanarak paketi doğrular ve bu şekilde cevabın orijinalliği doğrulanmış olur.



Şekil 39. P-JRC tabanlı kimlik doğrulama modeli

Tez kapsamında, merkezi kimlik doğrulama işlemi tarafından getirilen haberleşme yükünü azaltmak amacıyla JRC benzeri öğelerin ağ performansı üzerindeki etkileri incelenmiştir. Burada, tüm ağ için kimlik doğrulama parametrelerinin düşük güçlü bir aygıta sığmayacağı bilinmektedir. Fakat, bazı aygıtların ağın bir bölümü için kimlik doğrulama bilgilerini sakladığı varsayılırsa, daha kısa kimlik doğrulama yollarından yararlanılarak ağın verimliliği arttırılabilir. P-JRC olarak seçilen düğümün ağda ideal bir şekilde bulunduğu ve ağ için kimlik doğrulama parametrelerini saklamak için yeterli bellek depolama alanının bulunduğu varsayılır. Önerilen dağıtık kimlik doğrulama mekanizması Şekil 39'da verilmiştir. Şekilde görüleceği üzere, eğer P-JRC kendi ağının JRC öğesine yönlendirilmiş bir kimlik doğrulama isteği alırsa ve bu istek için kimlik doğrulama bilgilerine sahipse, doğrulama sürecini başlatır. P-JRC öğesi, ağa katılmak isteyen cihazı doğrulayarak kimlik doğrulama belirteçleri ile bu cihaza bir cevap gönderir. Öte yandan P-JRC öğesi, istekte bulunan cihazın yönlendirme yolunda değilse, kimlik doğrulama işlemi, Şekil 33'te açıklandığı gibi merkezi JRC'de gerçekleştirilir.

6TiSCH için önerilen önyükleme protokolü Contiki işletim sisteminde uygulanmıştır ve exp5438 gömülü platform [70]1] için Cooja emülatöründe yöntemin performansı

değerlendirilmiştir. Değerlendirmede kullanılan parametreler Tablo 5'te verilmiştir. Ağ senaryoları, farklı random çekirdeklerde (seed) 5 kez çalıştırılarak ortalama doğrulama gecikmesi hesaplanmıştır. İdeal olarak yerleştirilmiş P-JRC öğesi her iki senaryoda da kullanılır. Her düğüm, 6TiSCH ağına senkron olduktan sonra 30 ile 60 saniye arasında rastgele bir sürede kimlik doğrulama isteği gönderir.

Tablo 5. Çalışmada Kullanılan Parametreler

Parametre	**Değer**
Düğüm Sayısı	20 - 25
Başlama gecikmesi (dakika)	30
Kimlik doğrulama isteği (saniye)	30 - 60
Rx (%)	70-80-90-100
Yayılım modu	Cooja UDGM [65]

7. Performans Sonuçları

Duyarga cihazların kısıtlı kaynaklara sahip olması nedeniyle uygulanan güvenlik mekanizmalarının, şifreleme hızını ve enerji tüketimini ölçmek önemlidir. Bu bölümde, IoT uygulamalarının iş yükü, ROM, RAM ve enerji tüketimi gibi parametreleri ölçmek için kullanılan yöntemler açıklanmıştır. Bu yöntemler daha sonra değerlendirmede kullanılmıştır. Contiki'de bulunan gerçek zamanlı zamanlayıcılar kullanılarak ağın gecikme performansı değerlendirilmiştir. Gerçek zamanlı benzetim kullanılması yürütme zamanında daha yüksek doğruluk payı olan sonuçlar üretir. Bölüm 5'te Contiki işletim sisteminin gerçek zamanlı ve olaya dayalı zamanlayıcıları desteklediği anlatılmıştır.

KDA sınırlı bir güç kaynağı ile donatılmış olduğundan enerji çok kısıtlı bir kaynaktır. Normal uygulamalar bit başına mikro joule aralığında enerji tüketirken; açık anahtar şifreleme karmaşıklığına sahip güvenlik mekanizmaları bit başına mili-joule boyutlarında enerji tüketimine neden olur. Bu nedenle seçilen kimlik doğrulama mekanizmasında enerji tüketimi az olan simetrik şifreleme tekniği kullanılmıştır. Contiki, enerji tahminleri için **ENERGEST** adlı bir araca sahiptir. Energest, duyarga düğümünün enerji tüketimini tahmin eden, yazılım tabanlı bir mekanizmadır [71]2]. Bu mekanizma, Contiki içindeki radyo alıcısı ve CPU gibi tüm bileşenlerin enerji tahminlerini sağlamak için duyarga düğümü tarafından kullanılır. Bu araç CPU, radyo alıcı vericisi ve LED'ler gibi tüm bileşenleri içeren bir tablo tutar. Bir bileşen aktif olduğunda, bir sayaç bu bileşenin aktif olduğu zamanı ölçerek kaydeder. Bileşenin aktif olduğu zaman dilimi bileşenin güç gereksinimi ile çarpılarak tükettiği enerji hesaplanır. Tablo değerlerini normalleştirmek için seçilen donanımın karakteristik özellikleri bilinmelidir. Tablo 6'da normalleştirme için kullanılan parametreler gösterilmektedir. Bu değerler MSP-EXP430F5438 ve CC2538 veri setlerinden elde edilmiştir [70]1, [72]3].

Tablo 6. Bileşenlerin Çektiği Akımlar

Bileşen	**Anlık Tüketim (mA)**
CPU	1.9
LPM	0.0545

Tablo 6'nın devamı.

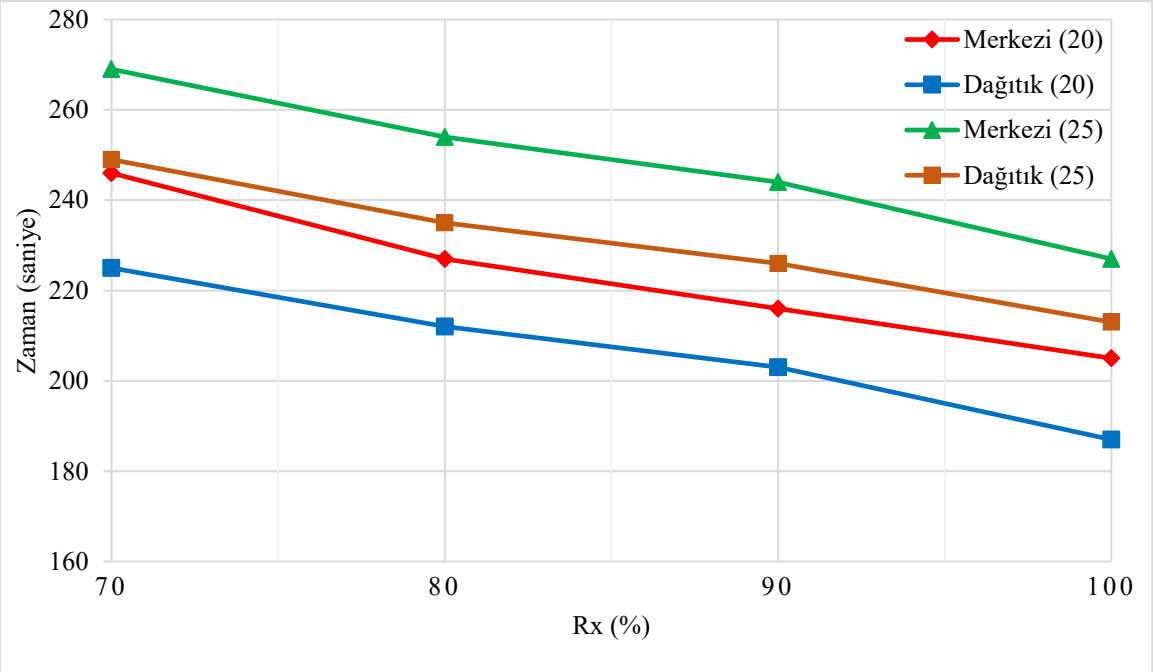
Tx	20
Rx	17.7
Vcc	3.3 V

Her bileşenin kaydı oluşturulduktan sonra, normalize olmayan değerlerden enerji elde etmek için duyarga düğümün voltajıyla çarpılması gerekir. Contiki işletim sisteminin RTIMER_SECOND[6] değişkenine bölünerek gerçek zamanlı enerji tüketiminde kullanılır. Duyarga düğümü tam güç moduna geçtiği için Vcc, 3.3 Volt değerinde kullanılır. Aşağıdaki formül, çalışmada kullanılan enerji hesabındaki kod formatını göstermektedir.

$$avg_{power} = \frac{(1.9cpu_{time} + 0.545lpm_{time} + 20listen_{time} + 17.7transmit_{time})3.3v}{\text{RTIMER\_SECOND}};$$

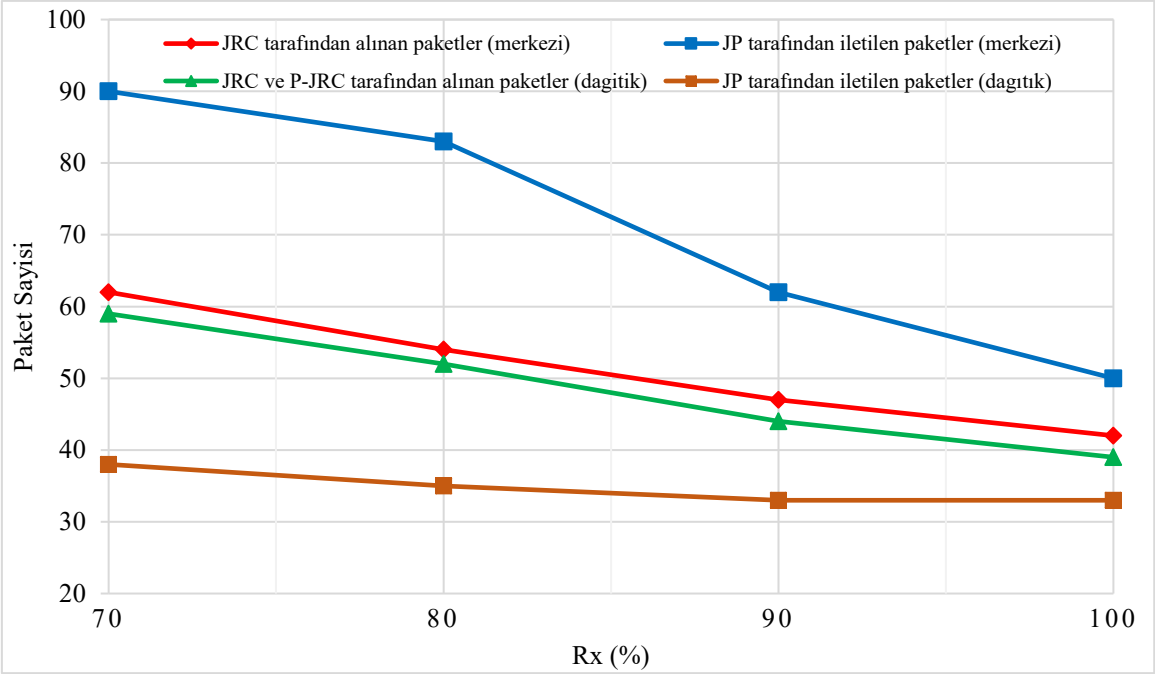
Şekil 40'da, bir adet P-JRC öğesi kullanan 20 ve 25 düğümlük bir ağ için uygulanan iki doğrulama mekanizmasının performans sonuçları gösterilmiştir. Düğüm kimliklerinin doğrulanma süresi hem merkezi hem de dağıtık senaryolarda ağın bağlantı kalitesine bağlı olarak değiştiği gözlemlenmiştir. Düğümlerin önyükleme işlemi, düşük bağlantı kalitesinde daha uzun sürer. P-JRC'ye sahip dağıtık kimlik doğrulama mekanizmasındaki ortalama ağa dahil olma süresinin tüm bağlantı kalitesi olasılıkları için merkezi yaklaşıma göre daha kısa sürdüğü gözlemlenmiştir. Düğüm senaryoları için dağıtık kimlik doğrulama süresi, merkezi kimlik doğrulama mekanizmasına göre yaklaşık %29 daha iyi performans sağlar.

[6] Bu değişken gerçek zamanlı saati ifade etmektedir.



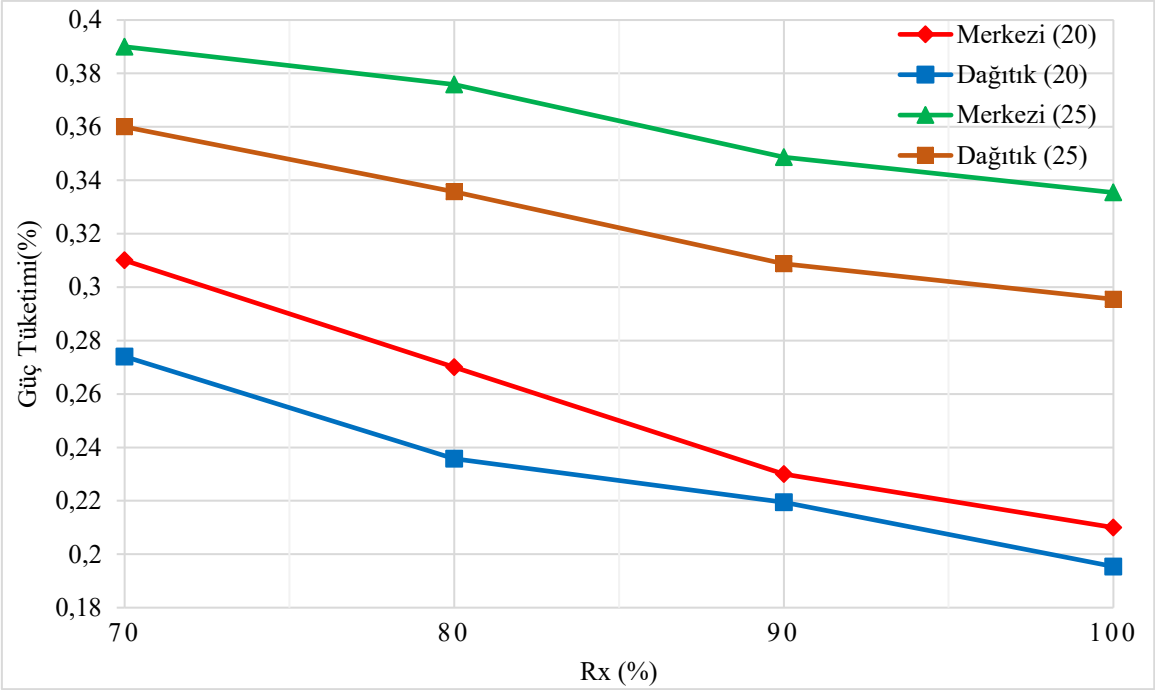
Şekil 40. Değerlendirilen ağlar için kimlik doğrulama süresi (1xP-JRC)

Şekil 41'de 20 düğümlü ağ senaryosu için merkezi ve dağıtık kimlik doğrulama mekanizmalarında gönderilen ve alınan doğrulama paketlerinin sayısı verilmiştir. Şekildeki yatay eksen bağlantı başarı oranını ifade eder. Kimlik doğrulama işlemi için tüm ağı dolaşmak zorunda kalan toplam paket sayısı, önerilen dağıtık kimlik doğrulama mekanizmasına kıyasla merkezi bir kimlik doğrulama yaklaşımı kullanıldığında belirgin olarak daha yüksektir. Çalışmada, P-JRC'lerin testten önce tüm kimlik doğrulama anahtarlarıyla donatılmış olduğu varsayılır. Bununla birlikte, dağıtık kimlik doğrulama mekanizmasının sonuçları, IETF 6TiSCH ağları için uygulanan merkezi kimlik doğrulama ile karşılaştırıldığında %50 daha az paket iletimi gerçekleştirdiğini gösterir.

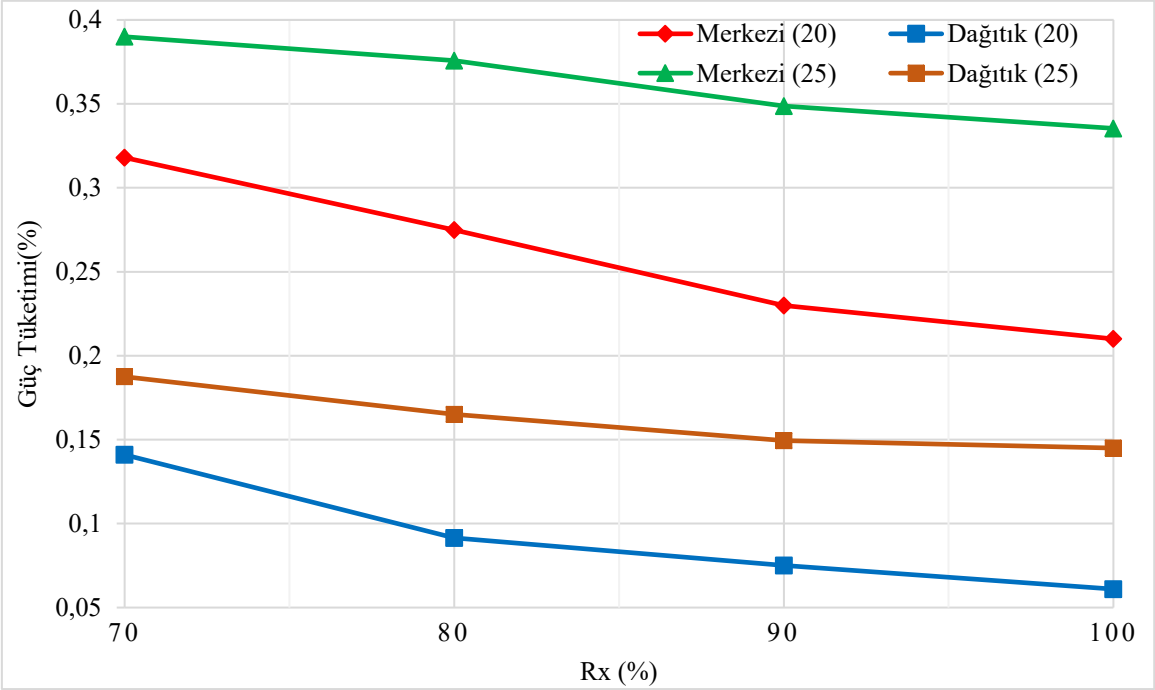


Şekil 41. 20 Düğüm için gönderilen/alınan kimlik doğrulama paketlerinin sayısı (1xP-JRC)

Şekil 42, 3.3V-100 mAh batarya varsayıldığında düğümlerin ağa önyükleme sırasında tükettikleri ortalama enerji miktarını verir. Bu çalışma için, [72] 'te verilen bilgileri kullanarak exp5438 platformunun enerji tüketim rakamları simüle edilmiştir. Sonuçlardan görüleceği üzere, önerilen dağıtık önyükleme mekanizmasının enerji tasarrufu, ağ içindeki düğüm sayısı arttıkça daha belirginleşir. Dahası, beklendiği gibi ağın bağlantı kalitesi, önyükleme sürecinin enerji tüketimini önemli ölçüde etkilemektedir. Sonuçlar, kimlik doğrulama işlemini ağın kenarında gerçekleştirmenin, 6TiSCH tabanlı IoT ağının önyükleme sürecinin iletişim gereksinimi ve enerji tüketimini önemli ölçüde azaltabileceğini göstermiştir. P-JRC sayısının iki olduğu durumda ön yükleme performansı Şekil 43'te görüldüğü gibi tek P-JRC'nin kullanıldığı senaryoya göre %30 kadar artar.



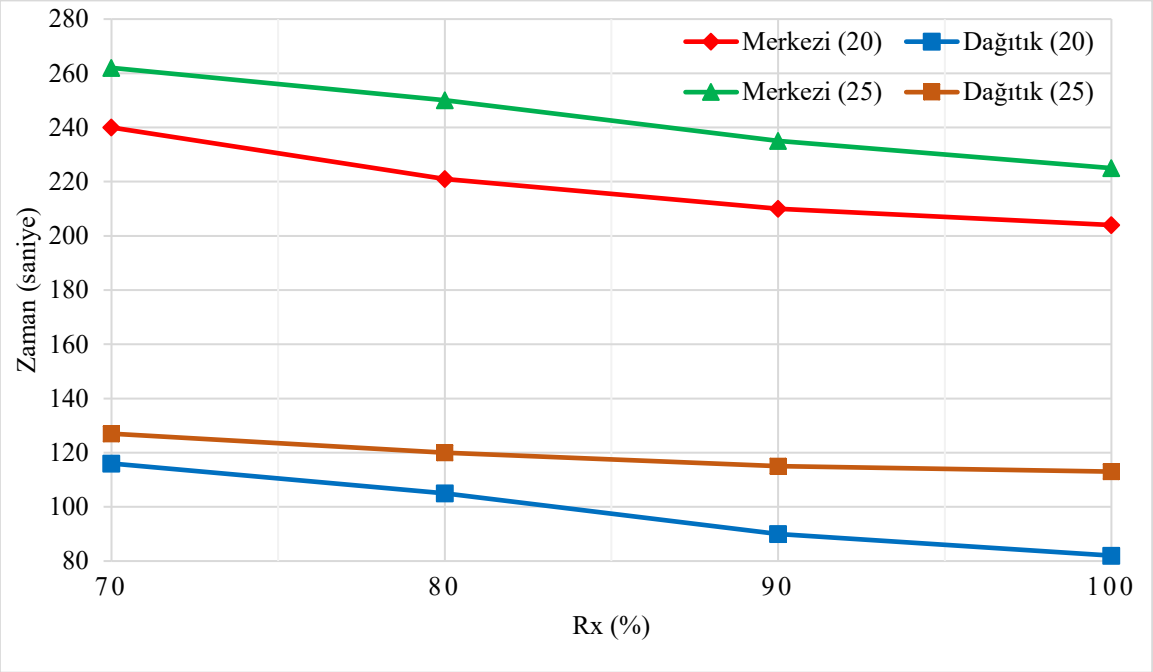
Şekil 42. Önyükleme işlemi sırasında tüketilen ortalama enerji miktarı (1xP-JRC)



Şekil 43: Önyükleme işlemi sırasında tüketilen ortalama enerji miktarı (2xP-JRC)

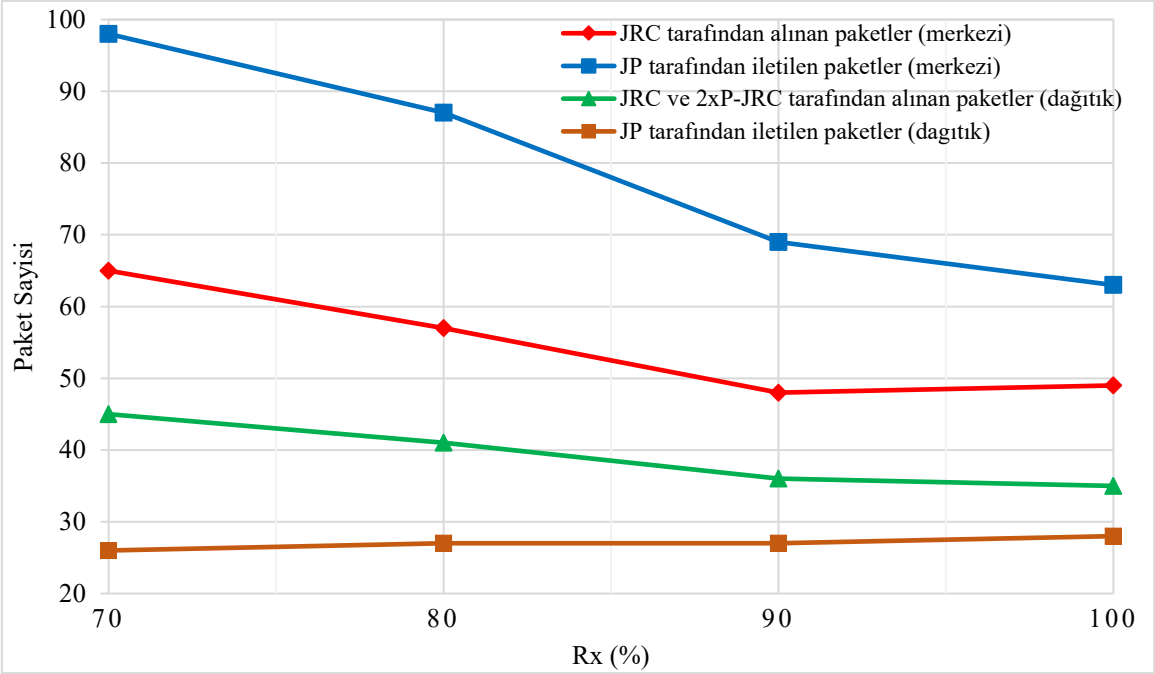
Şekil 44'te, iki adet P-JRC öğesi kullanılan 20 ve 25 düğüm için uygulanan iki doğrulama mekanizmasının performansı verilmiştir. Düğüm kimliklerinin doğrulanma

süresinin hem merkezi hem de dağıtık senaryolarda ağın bağlantı kalitesine bağlı olarak değiştiği gözlemlenmiştir. Düğümlerin önyükleme işlemi, düşük bağlantı kalitesinde daha uzun sürer. Merkezi yapıya kıyasla P-JRC sahip dağıtık kimlik doğrulama mekanizmasındaki ortalama ağa dahil olma süresi tüm bağlantı kalitesi olasılıkları için daha azdır. Düğüm senaryoları için dağıtık kimlik doğrulama süresi, merkezi kimlik doğrulama mekanizmasına göre yaklaşık %46 daha iyi performans sağlar.



Şekil 44. Değerlendirilen ağlar için kimlik doğrulama süresi (2xP-JRC)

Şekil 45'te 25 düğümlü ağ senaryosu için bağlantı kalitesine bağlı olarak merkezi ve dağıtık kimlik doğrulama mekanizmalarında gönderilen/alınan doğrulama paketlerinin sayısını gösterir. Kimlik doğrulama işlemi için tüm ağı dolaşmak zorunda kalan toplam paket sayısı, önerilen dağıtık kimlik doğrulama mekanizmasına kıyasla merkezi bir kimlik doğrulama yaklaşımı kullanıldığında belirgin olarak daha yüksektir. Kullanılan P-JRC cihaz sayısının artması ağın performansında önemli bir kazanca neden olur.



Şekil 45. 25 Düğüm için gönderilen/alınan kimlik doğrulama paketlerinin sayısı (2xP-JRC)

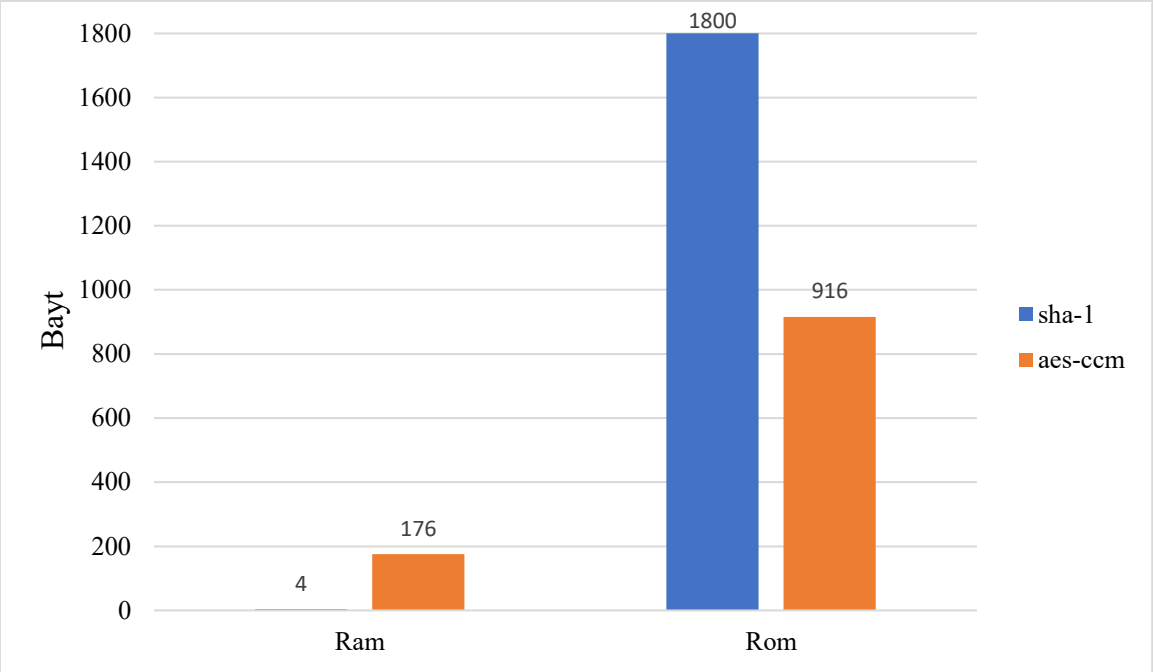
Duyarga düğümlerinin sınırlı bellek kaynakları nedeniyle kullanılan RAM/ROM bellekleri, kablosuz duyarga ağlarının güvenlik katmanı tasarımında dikkate alınması gereken önemli parametrelerdir. Geleneksel şifreleme yöntemleri gömülü aygıtlar için çok büyük olduğundan, bu çalışmada seçilen güvenlik yöntemlerinin düşük maliyetli olmasına özen gösterilmiştir. Contiki işletim sisteminin kullandığı bellek kapasitesi ölçülerek kimlik doğrulama mekanizmasının maliyeti hesaplanmaya çalışılmıştır. Bunu başarmak için, kod tüm şifreleme işlevleriyle derlenmiştir ve **msp430-size** [73] yardımcı programı kullanılarak boyutu ölçülmüştür. Bu değerler, kimlik doğrulama mekanizması için kullanılan RAM/ROM gereksinimlerinin gerçek boyutlarına yakın sonuç vermektedir.

Bununla birlikte, yığının değişken boyutundan dolayı RAM kullanımını ölçmek zordur ve yığın dinamik bellek ayırma için kullanılmaktadır. Tablo 'de exp5438 gömülü platform için çalıştırılabilir alanların ayrıntıları verilmektedir. Çalıştırılabilir bir program, .text, .bss ve .data adlı bölümlere ayrılmıştır. **.text** alanı statik belleği işaret etmektedir ve bu bölüm kodlar ve statik değişkenleri içermektedir. **.data** alanı, program başladığında sıfıra atanan ROM üzerinde herhangi bir alanı kaplamayan, ancak program çalışırken RAM'de saklanan veriler kümesini oluşturmaktadır. Son olarak, **.bss** alanı, program başladığında bir değerle başlatılan veridir, dolayısıyla RAM'de olduğu gibi ROM'da da saklanmalıdır. Tablodan çıkarılacağı üzere İstemci uygulamasında kullanılan kimlik doğrulama

mekanizmaları için RAM'de kullanılan bellek miktarları hemen hemen aynıdır. Dağıtık kimlik doğrulama yönteminde kullanılan ROM miktarı ise merkezi yapıya göre 5 kB daha fazla maliyete sebep olur. Sınır yönlendiricinin ağ oluşumundaki payının büyük olması, İstemci uygulamasına göre RAM/ROM alanlarındaki bellek tüketim miktarını arttırır. Sınır yönlendirici, İstemci uygulamasına benzer olarak merkezi doğrulamaya oranla dağıtık kimlik doğrulama mekanizması kullandığında ROM bellek alanına 5 kB ek bir maliyet getirir. Kullanılan doğrulama mekanizmaları için RAM alanındaki bellek tüketim oranları birbirlerine çok yakındır.

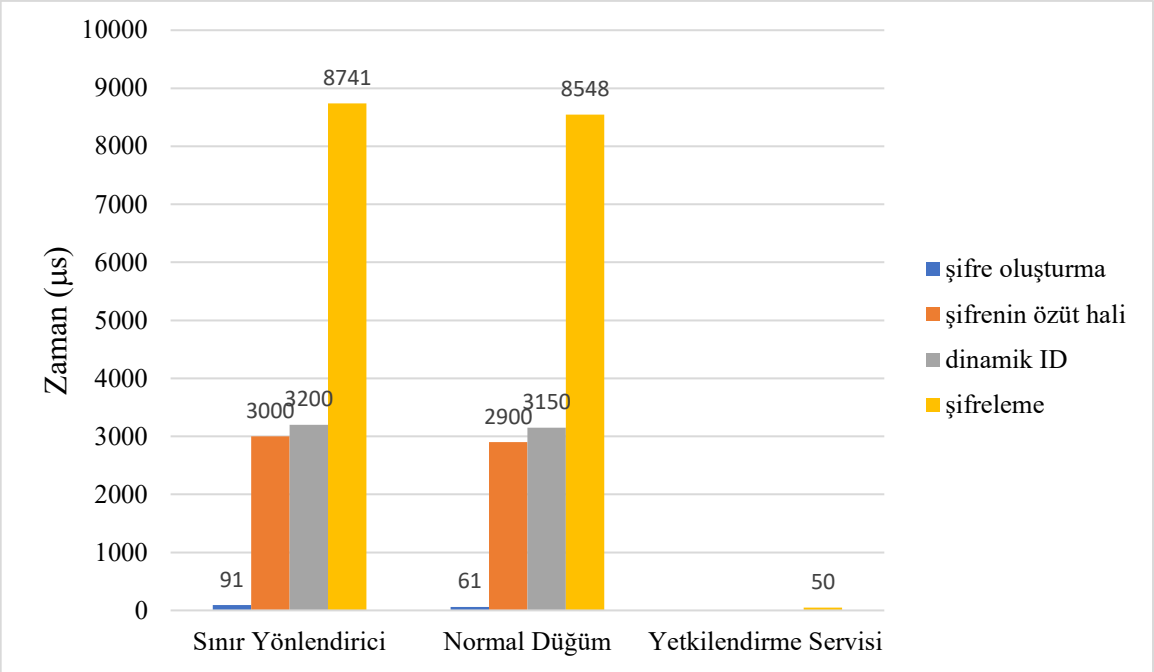
Tablo 7. Merkezi ve Dağıtık Kimlik Doğrulama Protokollerinin Bellek Kullanımı (Bayt)

	Merkezi	Dağıtık
İstemci		
.text	111082	116272
.bss	13438	13440
.data	630	630
RAM (data + BSS)	14068	14070
ROM (text + data)	124520	129712
Sınır Yönlendirici		
.text	112780	117972
.bss	14454	14456
.data	662	662
RAM	15116	15118
ROM	127234	132428



Şekil 46. Aes-128 ve Sha-1 için Bellek Tüketimi (Bayt)

Şekil 46'da kimlik doğrulama mekanizmaları için kullanılan yöntemlerin (AES-128, SHA-1) bellekte kapladığı alanlar gösterilmiştir. Bu yöntemlerin büyük bir kısmı ROM bellekte saklanır. RAM bellekte ise çalışma durumunda oluşturulan değişkenler kaydedilmiştir. Önerilen yöntem simetrik şifreleme ve bütünlük kontrolü yapan basit bir mekanizma içerdiği için düğümlere aşırı bir maliyete sebebiyet vermez.



Şekil 47. Kimlik doğrulama için yapılan işlemlerin ortalama süreleri

Şekil 47, kimlik doğrulamada yer alan cihazların gerçekleştirmiş olduğu işlemlerin ortalama süresini gösterir. Burada sınır yönlendirici ile normal düğüm kısıtlı kaynaklara sahip olduğundan kendi aralarında değerlendirilir. Yetkilendirme servisi ise merkezi olarak ele alındığında kaynak sıkıntısı olmayan cihaz olarak kabul edilir. Şifre oluşturma işlemi her cihaza özel olan 20 baytlık bir değer üretir. Bu değer özüt fonksiyonu yardımıyla yeni bir forma bürünür. Daha sonra bu değer JRC öğesinin veritabanında kullanılmak üzere kayıt altına alınır. Özüt hali ile birlikte yetkilendirme servisi direkt olarak cihazın gerçek kimlik bilgilerine ulaşamaz. Bu durum temel düzeyde kimlik bilgilerinin gizlenmesine yardımcı olur. Dinamik kimlik bilgisinin oluşturulması yukarıda bahsedilen şifrenin özüt halinin getirmiş olduğu fayda ile ortak paya sahiptir. Şifreleme adımında AES-128 simetrik şifreleme tekniği kullanıldığından sistemin ek maliyet ihtiyacı fazla artmamıştır. Doğrulama adımlarından sonra sistem otomatik olarak güvenli moda geçtiğinden bütün mesaj trafiği şifrelenmektedir. Bu özellik, ağa dahil olan cihazların güvenli bir şekilde birbirleriyle haberleşmesini sağlar.

8. SONUÇ

Nesnelerin İnterneti vizyonu başarılı bir şekilde uygulandığında ve entegre edildiğinde hayatımızı kolaylaştıracak bir teknoloji olarak karşımıza çıkacaktır. Akıllı nesnelerle çevrili bir günlük yaşamı hayal etmek, son birkaç yıldır akıllı telefonlar tarafından kazanılan popülariteyi gördükten sonra pek de zor değildir. Fakat bu teknolojilerin en önemli sorunu güvenlik mekanizmalarının tam olarak fiziki dünyaya uygulanamamasıdır. Kısıtlı kaynaklara sahip cihazların kullanılması güvenlik için gerekli olan yapıların kullanılmasını engellemektedir. Klasik şifreleme tekniklerinin dolaylı yollardan bu cihazlara uygulanması var olan güvenlik açıklarını tam anlamıyla kapatamaz. Belirtilen sebeplerden ötürü minimum güvenliği sağlayacak güvenlik mekanizmalarının dizayn edilmesi gerekmektedir. Güvenlik mekanizmalarının donanımı destekleyecek şekilde oluşturulması şifreleme için gerekli olan işlem hızını düşürecektir. İşlem hızının düşmesi ile birlikte zaman dilimleri küçük seçilerek senkronizasyon için gerekli olan radyo on/off durumu daha kararlı bir hal alacaktır. Radyonun kapalı kalma süresi arttırılarak bataryanın daha uzun ömürlü olması sağlanacaktır.

Gelecek çalışmalarda önerilen yöntemin saha testi gerçekleştirilerek sonuçları gözlemlenecektir. IoT ağları için arzulanan güvenlik mekanizmaları, farklı uygulamaların farklı ihtiyaçlarını dikkate alarak esneklik sağlamak için çeşitli seviyelere veya seçeneklere bölünen güçlü bir güvenlik hizmeti sunmalıdır. Bu ağlarda, ağ dinlenilmesinin önüne geçilmesi halinde iletişime, güç kaynağına ve gizliliğe karşı yapılan saldırılar büyük ölçüde engellenecektir. Aynı zamanda düğümler arasında gerçekleşen anahtar değişimindeki kriptolojik saldırıları önlemek için büyük bir fırsat sunacaktır. İletilen içeriklerin bütünlük ve doğruluğu takip edilerek, özellikle sağlık alanı gibi hassas verilen değerlendirildiği uygulama alanlarında güncel bilgilerin kullanılması sağlanmalıdır.

9. KAYNAKLAR

1. http://www.ieee802.org/15/pub/TG4e.html, IEEE 802.15 wpan 4e task group. 11 Kasım 2017.

2. Zhong, F., Kulkarni, P., Gormus, S., Efthymiou, C., Kalogridis, G., Sooriyabandara, M., Zhu, Z., Lambotharan, S., ve Chin, W. H. , Smart grid communications: Overview of research challenges, solutions, and standardization activities, IEEE Communications Surveys and Tutorials, 15, 1 (2013) 21–38.

3. Liu, T. ve Lu D., "The application and development of iot, International Symposium on Information Technology in Medicine and Education, 15, 1 (2012) 991–994.

4. Kevin, A., That internet of things thing. rfid journal (2009). http://www. rfidjournal. com/articles/view 14 Ekim 2017.

5. Yuxi, L. ve Guohui, Z., Key technologies and applications of internet of things, in Intelligent Computation Technology and Automation (ICICTA), 2012 Fifth International Conference on, Şubat 2012, Hunan, Bildiriler Kitabı, 197–200.

6. Harald, S., Patrick, G., Peter, F., ve Sylvie, W., Vision and challenges for realising the internet of things, Cluster of European Research Projects on the Internet of Things, European Commision, 3, 3 (2010) 34–36.

7. Dave, E., The internet of things, april 2011. http://www.cisco.com/c/dam/en-us/about/ac79/docs/innov/IoT_IBSG 0411FINAL. Pdf 10 Ekim 2017.

8. Wood, D., ve Stankovic,A., Denial of service in sensor networks, Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems, 35, 10 (2002) 54–62.

9. Chris, K., ve David, W., Secure routing in wireless sensor networks: Attacks and countermeasures, Ad hoc networks, 1, 2 (2003) 293–315.

10. Wood, A. D., ve Stankovic, J. A., A taxonomy for denial-of-service attacks in wireless sensor networks, Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems, 4 (2004) 739–763.

11. Deepak, G., Govindan, Govindan., Shenker, S., ve Estrin, D., Highly-resilient, energy-efficient multipath routing in wireless sensor networks, ACM SIGMOBILE Mobile Computing and Communications Review, 5, 4 (2001) 11–25.

12. https://datatracker.ietf.org/wg/6tisch/charter/, Ipv6 over the tsch mode of ieee 802.15.4e (6tisch). 28 Eylül 2017.

13. http://datatracker.ietf.org/wg/6lowpan, IETF IPv6 over Low Power Wireless Personal Area Networks (6lowpan) Working Group. 2 Haziran 2017.

14. Shelby, Z., Hartke, K., ve Bormann, C., The constrained application protocol, CoAP, https://tools.ietf.org/html/rfc7252 11 Kasım 2017.

15. Association.I. S., IEEE 802.15. 4-2006 ieee standard for information technology–telecommunications and information exchange between systems–local and metropolitan area networks–specific requirements part 15.4: Wireless medium access control (mac) and physical layer (phy) specifications for low rate wireless personal area networks (lr-wpans), http://standards. ieee. org/getieee802/download/802.15 12 Aralık 2017.

16. Committee, L. S., Part 15.4: wireless medium access control (mac) and physical layer (phy) specifications for low-rate wireless personal area networks (lr-wpans), IEEE Computer Society, 2003.

17. Daemen, J., ve Rijmen, V., The block cipher bksq, in International Conference on Smart Card Research and Advanced Applications, 236–245, Springer, Berlin, Heidelberg, 1998.

18. Committee vd., Ieee standard for information technology telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements part 11: Wireless lan medium-access control (mac) and physical layer (phy) specifications, IEEE Std 802.11 TM-2007, 2002.

19. Montenegro, G., Kushalnagar, N., Hui, J., ve Culler, D., Transmission of ipv6 packets over ieee 802.15. 4 networks, IETF, 2007, 14 Aralık 2017.

20. Thubert P., ve Hui, J. W., Compression format for ipv6 datagrams over ieee 802.15. 4-based networks, 2011.

21. Olsson, J., 6lowpan demystified, Texas Instruments, 13, 2014.

22. Winter, T., Rpl: Ipv6 routing protocol for low-power and lossy networks, Internet Engineering Task Force, 2012.

23. Levis, P., Patel, N., Culler,, D. ve Shenker, S., Trickle: A self-regulating algorithm for code propagation and maintenance in wireless sensor networks, in Proc. of the 1st USENIX/ACM Symp. on Networked Systems Design and Implementation, San Francisco, Mart 2004.

24. Jin, Y., Gormus, S., Kulkarni, P. ve Sooriyabandara, M., Content centric routing in iot networks and its integration in rpl, Computer Communications, 89, (2016) 87–104.

25. https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=6185525, Ieee std. 802.15.4e, part 15.4: Low-rate wireless personal area networks, amendment 1 Mac sublayer. 3 Kasım 2017.

26. Kim, A. N., Hekland, F., Petersen, S., ve Doyle, P., When hart goes wireless: Understanding and implementing the wirelesshart standard, in Emerging Technologies and Factory Automation, ETFA 2008. IEEE International Conference on, Ekim 2008, Hamburg, Bildiriler Kitabı, 899–907.

27. Petersen S. ve Carlsen, S., Wirelesshart versus isa100. 11a: The format war hits the factory floor, IEEE Industrial Electronics Magazine, 5, 4 (2011) 23–34.

28. Chang, T., Watteyne, T., Pister, K., ve Wang, Q., Adaptive synchronization in multi-hop tsch networks, Computer Networks, 76 (2015) 165–176.

29. Palattella, M. R., Accettura, N., Vilajosana, X., Watteyne, T., L. A. Grieco, Boggia, G. ve Dohler, M., Standardized protocol stack for the internet of (important) things, IEEE communications surveys & tutorials, 15, 3 (2013) 1389–1406.

30. Thubert, P., Watteyne, T., Palattella, M. R., Vilajosana, X., ve Wang, Q., Ietf 6tsch: Combining ipv6 connectivity with industrial performance, in Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), Seventh International Conference on, Eylül 2013, Taichung, 541–546.

31. Accettura N., ve Piro, G., Optimal and secure protocols in the ietf 6tisch communication stack, in Industrial electronics (ISIE), IEEE 23rd international symposium on, (2014) 1469–1474.

32. Shelby, Z., Vial, M., Koster, M., ve Groves, C., Reusable Interface Definitions for Constrained RESTful Environments, Internet-Draft draft-ietf-core-interfaces-06, draft-ietf-coreinterfaces-04, 2015.

33. Vilajosana X. ve Pister, K., Minimal 6tisch configuration-draft-ietf-6tisch-minimal-00, IETF, Fremont, 2013.

34. Görmüs, S., Synchronisation in 6tisch networks, in Signal Processing and Communications Applications Conference (SIU), 23th, Haziran 2015, Malatya, Bildiriler Kitabı, 535–539.

35. Perrig, A., Szewczyk, R., Tygar, J. D., Wen, V., ve Culler, D. E., Spins: Security protocols for sensor networks, Wireless networks, 8, 5 (2002) 521–534.

36. Xiao, Y., Security in distributed, grid, mobile, and pervasive computing, CRC Press, Boston, USA, 2007.

37. Shi E., ve Perrig, A., Designing secure sensor networks, IEEE Wireless Communications, 11, 6 (2004) 38–43.

38. Azarmehr, M., Ahmadi, A., ve Rashidzadeh, R., Secure authentication and access mechanism for iot wireless sensors, in Circuits and Systems (ISCAS), IEEE International Symposium on, Ekim 2017, Baltimore, 1–4.

39. Qiu, Y., ve Ma, M., A mutual authentication and key establishment scheme for m2m communication in 6lowpan networks, IEEE Transactions on Industrial Informatics, 12, 6 (2016) 2074–2085.

40. Esfahani, A., Mantas, G., Matischek, R., Saghezchi, F. B., Rodriguez, J., Bicaku, A., Maksuti, S., Tauber, M., Schmittner, C., ve Bastos, J., A lightweight authentication mechanism for m2m communications in industrial iot environment, IEEE Internet of Things Journal, 2017.

41. Altolini, D., Lakkundi, V., Bui, N., Tapparello, C., ve Rossi, M., Low power link layer security for iot: Implementation and performance analysis, in Wireless Communications and Mobile Computing Conference (IWCMC), Ağustos 2013, Sardinia, 919–925.

42. Law, Y. W., Doumen, J., ve Hartel, P., Survey and benchmark of block ciphers for wireless sensor networks, ACM Transactions on Sensor Networks (TOSN), 2, 1 (2006) 65–93.

43. Djedjig, N., Tandjaoui, D., ve Medjek, F., Trust-based rpl for the internet of things, in Computers and Communication (ISCC), Şubat 2016, Larnaca, Bildiriler Kitabı, 962–967.

44. Seeber, S., Sehgal, A., Stelte, B., Rodosek, G. D., ve Schonwalder, J., Towards a trust computing architecture for rpl in cyber physical systems, in Network and Service Management (CNSM), 9th International Conference on, Ocak 2014, Zurich, Bildiriler Kitabı, 34–137.

45. Healy, M., Newe, T. ve Lewis, E., Analysis of hardware encryption versus software encryption on wireless sensor network motes, in Smart Sensors and Sensing Technology, 3–14, Springer, Berlin, 2008.

46. Hamalainen, P., Hannikainen, M., ve Hamalainen, T. D., Efficient hardware implementation of security processing for ieee 802.15. 4 wireless networks, in Circuits

and Systems, 48th Midwest Symposium on, Şubat 2006, Covington, Bildiriler Kitabı, 484–487.

47. Huai, L., Zou, X., Liu, Z., ve Han, Y., An energy-efficient aes-ccm implementation for ieee802. 15.4 wireless sensor networks, in Networks Security, Wireless Communications and Trusted Computing, 9. International Conference on, Mayıs 2009, Wuhan, Bildiriler Kitabı, 394–397.

48. Otero, C. T. O., Tse, J., ve Manohar, R., Aes hardware-software co-design in wsn, in Asynchronous Circuits and Systems (ASYNC), 21st IEEE International Symposium on, Temmuz 2015, Mountain View, Bildiriler Kitabı, 85–92.

49. Daidone, R., Dini, G., ve Tiloca, M., On experimentally evaluating the impact of security on ieee 802.15. 4 networks, in Distributed Computing in Sensor Systems and Workshops (DCOSS), 2011 International Conference on, Ağustos 2011, Barcelona, Bildiriler Kitabı, 1–6.

50. Chen, F., Yin, X., German, R., ve Dressler, F., Performance impact of and protocol interdependencies of ieee 802.15. 4 security mechanisms, in Mobile Adhoc and Sensor Systems, IEEE 6th International Conference on, Kasım 2009, Macau, Bildiriler Kitabı, 1036–1041.

51. Vucinic, M., Tourancheau, B., Rousseau, F., Duda, A., Damon, L., ve Guizzetti, R., Energy cost of security in an energy-harvested ieee 802.15. 4 wireless sensor network, in Embedded Computing (MECO), 3rd Mediterranean Conference on, Temmuz 2014, Budva, Bildiriler Kitabı, 198–201.

52. Xiao, Y., Chen, H.-H., Sun, B., Wang, R., ve Sethi, S., Mac security and security overhead analysis in the ieee 802.15. 4 wireless sensor networks, EURASIP Journal on Wireless Communications and Networking, 1 (2006) 93-830.

53. Hummen, R., Shafagh, H., Raza, S., Voig, T., ve Wehrle, K., Delegation-based authentication and authorization for the ip-based internet of things, in Sensing, Communication, and Networking (SECON), Eleventh Annual IEEE International Conference on, pp. Aralık 2014, Singapore, Bildiriler Kitabı,284–292.

54. Porambage, P., Schmitt, C., Kumar, P., Gurtov, A., ve Ylianttila, M., Two-phase authentication protocol for wireless sensor networks in distributed iot applications, in Wireless Communications and Networking Conference (WCNC), Kasım 2014, Istanbul, Bildiriler Kitabı, 2728–2733.

55. https://tools.ietf.org/html/draft-ietf-6tisch-minimal-security-02, Minimal security framework for 6tisch. 03 Kasım 2017.

56. Sarikaya, B., Ohba, Y., Cao, Z., ve Cragie, R., Security bootstrapping of resource-constrained devices, Security Bootstrapping of Resource-Constrained Devices, https://tools.ietf.org/html/draft-sarikaya-core-sbootstrapping-05 2 Ocak 2018.

57. Garcia-Morchon, O., Kumar, S., Keoh, S., Hummen, R., ve Struik, R., Security considerations in the ip-based internet of things draft-garciacore-security-06, Internet Engineering Task Force, 2013.

58. Nguyen, K. T., Laurent, M., ve Oualha, N., Survey on secure communication protocols for the internet of things, Ad Hoc Networks, 32 (2015) 17–31.

59. Hernandez-Ramos J. L., Pawlowski, M. P., Jara, A. J., Skarmeta, A. F., ve Ladid, L., Toward a lightweight authentication and authorization framework for smart objects, IEEE Journal on Selected Areas in Communications, 33, 4 (2015) 690–702.

60. Bersani F., ve Tschofenig, H., The eap-psk protocol: A pre-shared key extensible authentication protocol (eap) method, https://buildbot.tools.ietf.org/html/rfc4764 16 Ekim 2017.

61. Dunkels, A., Gronvall, B., ve Voigt, T., Contiki-a lightweight and flexible operating system for tiny networked sensors, in Local Computer Networks, 29th Annual IEEE International Conference on, Aralık 2004, Tampa, Bildiriler Kitabı, 455–462.

62. Dunkels A.ve Schmidt, O., Protothreads-lightweight stackless threads in c. http://www.diva-portal.org/smash/get/diva2:1041636 16 Ekim 2017

63. Levis, P., Madden, S., Polastre, J., Szewczyk, R., Whitehouse, K., Woo, A., Gay, D., Hill, J., Welsh, M., ve Brewer, E., Tinyos: An operating system for sensor networks, Ambient intelligence, 35, Springer, Berlin, Bildiriler Kitabı,115–148.

64. Farooq M. O. ve Kunz, T. Operating systems for wireless sensor networks: A survey, Sensors, Molecular Diversity Preservation International, 11, 6 (2011) 5900–5930.

65. Osterlind, F., Dunkels, A., Eriksson, J., Finne, N., ve Voigt, T., Cross-level sensor network simulation with cooja, in Local computer networks, proceedings 31st IEEE conference on, Şubat 2007, Tampa, Bildiriler Kitabı, 641–648.

66. Rescorla E., ve Modadugu, N., Datagram transport layer security version. https://buildbot.tools.ietf.org/html/rfc6347 6 Temmuz 2017.

67. Selander, G., Mattsson, J., Palombini, F., ve Seitz, L., Object security of coap (oscoap), Internet Engineering Task Force Internet-Draft work in progress, 2017.

68. https://tools.ietf.org/html/draft-ietf-6tisch-dtsecurity-zerotouch-join-01, 6tisch zero-touch secure join protocol. 03 Kasım 2017.

69. Eastlake D., ve Jones, P., Us secure hash algorithm 1 (sha1), tech. rep., 2001.

70. Msp-exp430f5438 experimenter board user guide, texas instruments inc., Texas. http://www.ti.com/lit/ug/slau263i/slau263i.pdf 10 Ekim 2017.

71. Dunkels, A., Osterlind, F., Tsiftes, N., ve He, Z., Software-based on-line energy estimation for sensor nodes, in Proceedings of the 4th workshop on Embedded networked sensors, Haziran 2007, New York, Bildiriler Kitabı, 28–32.

72. Lajara, R., Pelegr-Sebastiá, J., ve Solano, J. J. P., Power consumption analysis of operating systems for wireless sensor networks, Sensors, 10, 6 (2010) 5809–5826.

73. http://mspgcc.sourceforge.net/, mspgcc- gcc toolchain for msp430. 28 Ekim 2016.

ÖZGEÇMİŞ

Hakan Aydın, 1991 Giresun doğumludur. 2010 yılında girdiği Karadeniz Teknik Üniversitesi Bilgisayar Mühendisliği Bölümü'nden 2015 senesinde mezun olmuştur. 2015 yılında Trabzon' da yüksek lisans eğitimine başlamıştır. 2016 yılında da Karadeniz Teknik Üniversitesi Yazılım Mühendisliği Bölümünde araştırma görevlisi olarak işe başlamış ve halen de görevini devam ettirmektedir.