



Düzce Üniversitesi Bilim ve Teknoloji Dergisi

Araştırma Makalesi

Düşük Güçlü, Kayıplı Ağlarda RPL Yönlendirme ve Kenar İşleme Destekli, Merkezi Yönetime Sahip İzleme Sistemi

 Arif CEYLAN^{a,*},  Ali ÇALHAN^b  Murtaza CİCİOĞLU^c

^a MEB, İstanbul, TÜRKİYE

^b Bilgisayar Mühendisliği Bölümü, Mühendislik Fakültesi, Düzce Üniversitesi, Düzce, TÜRKİYE

^c Bilgisayar Mühendisliği Bölümü, Mühendislik Fakültesi, Bursa Uludağ Üniversitesi, Bursa, TÜRKİYE

* Sorumlu yazarın e-posta adresi: arifceylan@gmail.com

DOI: 10.29130/dubited.1246722

ÖZ

Son yıllarda, nesnelerin internetindeki cihaz sayısının devasa artışı sebebiyle, cihazların IPv6 destekli olması önem kazanmaktadır. Düşük güçlü cihazlarda IPv6'ya uyumlu olması amacıyla kullanılan 6LoWPAN teknolojisi ve bu teknolojiyi kullanan ağlara özgü izleme mekanizmalarının önemi her geçen gün artmaktadır. Çalışmamızda 6LoWPAN ağlarında sensör verilerini yönetmek, izlemek, gerektiğinde kritik verileri buluta göndermek için bir mimari önerilmiş ve benzetimi ile uygulaması yapılmıştır. Uygulamada kullanılan teknolojiler kenar işleme, CoAP (kısıtlı uygulama protokolü), 6LoWPAN, düşük güçlü ve kayıplı ağlar için yeni nesil işletim sistemi Contiki-NG, Cooja benzetim programı, RPL yönlendirme protokolü, MQTT mesajlaşma protokolü ve Python uygulamalarıdır. Cooja benzetim programında yerleştirdiğimiz düğümler çok atlamalı bir şekilde yönlendiriciye veri gönderebilmekte, yönlendirici üzerinden düğümlerden veri istenebilmektedir. Yönlendiriciye bağlı bir kenar işleme uygulaması verileri istenen zaman aralıklarıyla alabilmekte, kritik durumlarda bunu buluta gönderebilmektedir. Uygulamamızda çok atlamalı iletişim için RPL yönlendirme protokolü, düğümler ile uygulamamız arasındaki veri iletişimi için CoAP protokolü, veri toplama ve kenar işleme işlemleri için Python programlama dili, veri tabanı olarak SQLite kullanılmaktadır. Kenar ile bulutun iletişimi için abone ol/yayımla mekanizmasına sahip olan MQTT iletişim protokolü kullanılmaktadır.

Literatürde kenar işleme amaçlı grafik ara yüzüne sahip bir uygulamaya rastlanılmamış olup, geliştirdiğimiz uygulamanın merkezi bir şekilde yönetim işlevi olmasının da literatüre katkı sağlayacağı düşünülmektedir. Uygulamamızın benzetim üzerinde sorunsuz bir şekilde çalıştığı görülmüş olup, grafik ara yüzü kolay kullanımı sayesinde askeri, sağlık, tarım, hayvancılık, endüstri gibi birçok alanda kullanılabileceğini düşünmekteyiz.

Anahtar Kelimeler: Nesnelerin İnterneti (IoT), 6LoWPAN, Cooja, Contiki-NG.

Edge Computing Supported Centrally Managed Monitoring System In Low Power and Lossy Networks with RPL Routing

ABSTRACT

In recent years, the massive increase in the number of Internet of Things devices has made it increasingly important for devices to be IPv6-enabled. The use of 6LoWPAN technology, which is used to make low-power devices compatible with IPv6, and the importance of networks specific monitoring mechanisms that use this technology are increasing day by day. In our study, we proposed an architecture for managing, monitoring, and sending critical data to the cloud when necessary in 6LoWPAN networks, with implementation and simulation. The technologies used in the application are edge computing, CoAP (constrained application protocol), 6LoWPAN, the next-generation operating system for low-power and lossy networks as Contiki-NG, the Cooja simulation program, the

RPL routing protocol, the MQTT messaging protocol, and Python applications. In the Cooja simulation program, the nodes that we placed can send data to the router in a multi-hop manner, and data can be requested from the nodes via the router. An edge computing application connected to the router can collect data at the desired intervals, and send it to the cloud in critical situations. In our application, we used the RPL routing protocol for multi-hop communication, the CoAP protocol for data communication between the nodes and the application, the Python programming language for data collection and edge computing operations, and SQLite as the database. We used the MQTT communication protocol for communication between the edge and the cloud, which has a subscribe/publish mechanism.

In the literature, an application with a graphical interface for edge processing could not be found, and it is thought that the central management function of the developed application will also contribute to the literature. It was seen that our application worked seamlessly on the simulation and with its graphic user interface, we think it can be used in many fields such as military, health, agriculture, livestock, and industry.

Keywords: Internet of Things (IoT), 6LoWPAN, Cooja, Contiki-NG

I. GİRİŞ

Nesnelerin interneti (IoT) ağlarında algılayıcı bilgilerinin elde edilmesi ve çeşitli hizmetler sağlamak için çok sayıda cihaz internete bağlanmaktadır [1]. IoT cihazları akıllı ev, akıllı fabrika, endüstri, ulaşım, eğitim, sağlık hizmetleri, doğal afet yardımı, gözetim, eğlence gibi birçok alanda verileri izleme ve yönetme amaçlı kullanılmaktadır [1]-[2]-[3]-[4].

IoT paradigmasında milyarlarca cihazın birbirleriyle ve internet üzerinden haberleşmesi öngörülmektedir. IoT'nin temel amacı; insanların ve nesnelerin herhangi bir zamanda, herhangi bir yerde, herhangi bir nesneyle ve herkesle bağlantı kurmasını sağlamaktır [5]. Bu cihaz ve algılayıcılar arasındaki iletişim RFID, IEEE 802.15.4, Wi-Fi, Bluetooth düşük enerji (BLE) gibi farklı teknolojiler aracılığıyla yapılabilmektedir [5].

Kablosuz algılayıcı ağları, IoT'nin ortaya çıkmasında ve büyümesinde kilit bir rol oynamaktadır. Bu teknoloji sınırlı kaynaklara sahip düşük seviye cihazların internete bağlanmasına ve potansiyel olarak yaşamı değiştiren hizmetler sağlamasına izin vermektedir. Düşük güç ve kayıplı ağları (Low-power and Lossy Networks, LLNs) destekleyen ana standartlardan biri, IoT'nin bir parçası olarak kablosuz algılayıcı ağlarının omurgasını oluşturan IEEE 802.15.4 standardıdır. Bu standart, ağın fiziksel ve veri bağı katmanlarını tanımlamakta ve düşük maliyetlerle bir çalışma çerçevesi sağlamaktadır [6].

Bu düşük seviye cihazları internetin bir parçası yapmak için IETF, algılayıcı düğümlerinin İnternet Protokolü (IP) yığınına uygulamasına ve erişilebilir hale gelmesine izin veren bir uyarlama katmanı olarak kullanılan IPv6 düşük güçlü kablosuz kişisel alan ağlarını (6LoWPAN) geliştirmiştir [3]. 6LoWPAN sayesinde bu düşük güçlü cihazlar IPv6 özelliğine sahip olmakta ve IPv6 üzerinden birbirleri ile haberleşebilmektedirler [6]. 6LoWPAN üzerinde yönlendirme protokolü olarak RPL (Düşük güçlü ve kayıplı cihazlar için yönlendirme protokolü) çalışmaktadır [1]. Tablo 1 6LoWPAN yığınının yapısını ve kullanılan protokolleri göstermektedir [7].

Tablo 1: 6LoWPAN yapısı ve protokolleri

Uygulama Katmanı CoAP, MQTT
Taşıma Katmanı UDP/TCP
Ağ Katmanı IPV6/ICMP6/RPL
6LoWPAN Adaptasyon Katmanı 6LoWPAN
Veri Bağı Katmanı IEEE 802.15.4 MAC
Fiziksel Katman IEEE 802.15.4 PHY

II. IoT PROTOKOLLERİ

İkinci bölümde 6LoWPAN IoT ağlarında kullanılan protokoller detaylı olarak anlatılmıştır. İlk olarak IoT ilişkisi ile RPL yönlendirme algoritması verilmiştir. İkinci olarak, iletişim için MQTT protokolü açıklanmıştır. Son olarak, CoAP protokolü ayrıntılı olarak tartışılmıştır.

A. RPL YÖNLENDİRME PROTOKOLÜ

RPL, IPv6 düşük güçlü cihazlar için tasarlanmış uzaklık vektör tabanlı bit yönlendirme protokolüdür ve 6LoWPAN adaptasyon katmanı desteği ile IEEE 802.15.4 standardında çalışmaktadır [3]. RPL'nin birincil amacı çok sekmeli yönlendirmeyi etkinleştirmek olup çok çeşitli uygulamaları kapsayacak şekilde tasarlanmıştır [8]. Düğümler için çok atlamalı hiyerarşik ağaç yapısında bir topoloji oluşturur. Burada her bir düğüm, gelen veriyi kök düğüme ulaşana kadar üstteki düğümlere iletir. RPL, kısıtlı kaynaklara sahip düğümler için veri yönlendirmesini başarılı ve verimli bir şekilde yönetir ve çift yönlü bağlantı, sağlamlık, güvenilirlik, esneklik ve ölçeklenebilirlik sağlayan bir çerçeve sağlar [3]. Düşük güçlü kablosuz kişisel alan ağı (6LoWPAN) üzerinden IPv6, kaynak kısıtlı algılayıcı/eyleyici cihazları ve istemci uygulamaları açısından birçok avantajı olduğu için IoT'nin önemli bir parçası haline gelmiştir. 6LoWPAN, uzun IPv6 başlığını IEEE 802.15.4 formatında küçük farklı uzunluktaki çerçevelere kapsülleme yeteneğine sahiptir [9].

RPL Depolanmış (Stored) ve Depolanmamış (none-stored) mod olmak üzere iki çalışma modu bulunmaktadır. Depolanmış modda düğümler yerel olarak yönlendirme tablolarını kendi hafızalarında tutarlar. Depolanmamış modda ise düğümler yönlendirme tablosu tutmaz, bunun yerine tüm yönlendirme tablosu kök düğümde tutulur [8]. RPL'nin temel görevi bir veya daha fazla RPL kenar yönlendiricisine dayanan, hedef odaklı yönlendirilmiş çevrimsiz çizge (DODAG) olarak adlandırılan ve ağ cihazları arasında çift yönlü IPv6 iletişimini destekleyen yarı-orman yönlendirme topolojisi oluşturmaktır [10]. RPL topolojisi oluşturulurken ilk olarak sınır yönlendirici aracılığıyla çok noktaya DIO (DODAG bilgi nesnesi) adlı kontrol mesajı yayınlanır [11][8].

Bir düğüm komşu düğümlerden DIO mesajı aldığı anda OF0, MRHOF gibi bir amaç fonksiyonu ile düğüm sırasını hesaplar ve ana düğüm olarak en düşük sıraya sahip komşu düğümleri seçerek DODAG'a katılır [11]. Bir damlama zamanlayıcısı kullanılarak çok noktaya yayın olarak DIO mesajlarının dinamik bir periyotta gönderimi yapılır. Ağa yeni katılan veya bir ağa katılamamış katılma istekli bir düğüm,

komşularından DIO mesajlarını istemek için tipik olarak DIS mesajı gönderir. DIS mesajları, düğümler için bir (grup) komşudan yönlendirme bilgisi istemek için kullanılır. Üst düğümünü seçen düğüm bir DAO mesajı yayımlayarak bunu üst düğümlerine ve kök düğüme iletir. Kök düğüm bir DAO-ACK mesajı ile bunu onayladığını belirtebilir ve bu mesajı alan düğüm artık ulaşılabilir olduğunu bilir. Daha sonra, kök düğüme olan mantıksal mesafenin bir temsili olan Rank'ı içeren çok noktaya yayın DIO'ları ile DODAG'ın reklamını yapmaya başlayabilir. DIO'yu duyan diğer düğümler komşu tablosunu güncelleyecek, buna göre bir ebeveyn seçecek ve böylece çok sekmeli topoloji oluşumuna katılacaktır. Ağ işlemi sırasında, düğümler komşularıyla bağlantı kalitesini izlemeye devam eder ve topolojiyi verimli tutmak için üst öğelerini günceller. [8]

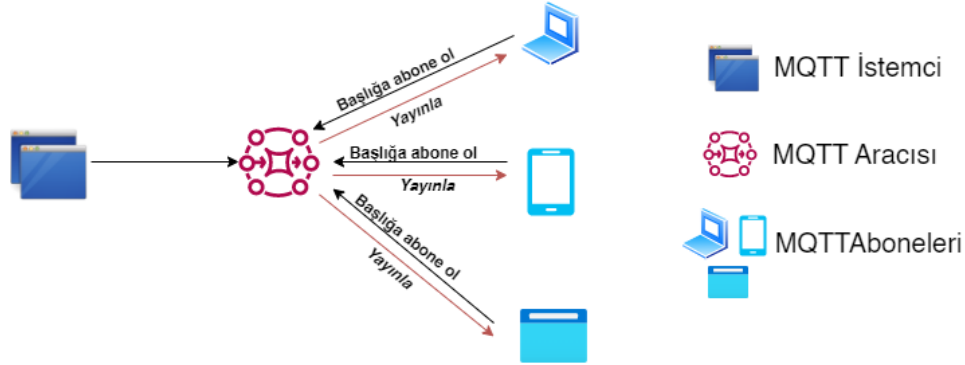
Oturum katmanında yapılan iletişim amaçlı, enerji ve bant genişliği tüketimi açısından verimli olan birkaç protokol vardır. Bu protokollerden bazıları MQTT, XMPP, AMQP ve CoAP'tır [12]

B. MQTT

MQTT, küçük boyut, düşük güç kullanımı ve birçok alıcıya verimli bilgi dağıtımını yapan, abone ol/bildir paradigmasına dayanan hafif bir taşıma protokolüdür. MQTT TCP protokolüne dayanmaktadır. Böylece paket kaybı kontrolü yapılmaktadır [12]. Bir abone ol/bildir mesajlaşma hizmetinde "mesaj" hizmet üzerinde aktarılan veriyi ifade eder. Konu bir mesaj beslemesini ifade eden bir varlıktır ve abonelik bir konu üzerine abone olup mesaj alınımını ifade eder. "Yayıncı veya yapımcı", mesajlar oluşturan ve bunları belirli bir konuda mesajlaşma servisine yayınlayan bir cihaz veya programı ifade eder ve "abone veya tüketici", belirli bir abonelikte mesajları alan bir cihaz veya programı ifade eder [13].

MQTT'nin Şekil 1 'de görüldüğü üzere üç bileşeni vardır:

- 1- Yayıncı veya Yapımcı (MQTT İstemcisi)
- 2- Bir komisyoncu (MQTT Aracısı)
- 3- Tüketici/Abone (MQTT İstemcisi)



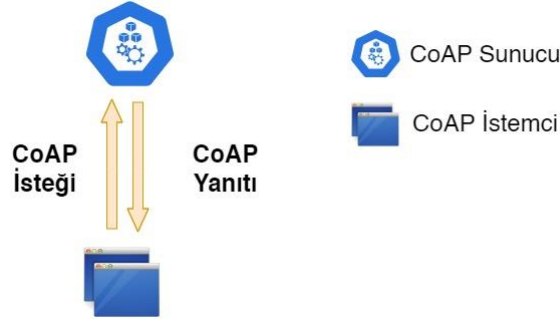
Şekil 1. MQTT Mimarisi

MQTT istemcisi MQTT protokolünü kullanan bir program veya cihazdır. İstemci, sunucuya ağ bağlantısı açmaktan, yayınlanacak mesajları oluşturmaktan, sunucuya uygulama mesajlarını yayınlamaktan, almak istediği uygulama mesajlarına abone olmaktan, uygulama mesajları talebini kaldırmak için abonelikten çıkmaktan ve ağ bağlantısını kapatmaktan sorumludur. MQTT aracısı, yayıncılar ve aboneler arasında postane görevi gören MQTT'ye dayalı bir program veya cihazdır [13].

B. CoAP

CoAP kısıtlı kaynaklara sahip düğümler için geliştirilmiş bir uygulama protokolüdür. HTTP protokolüne benzemekte olup uç noktalar arasında bir istek ve yanıt paradigmasına dayanmaktadır. CoAP çok noktaya yayın desteği, düşük ek yük ve kısıtlı ortamlardaki karmaşıklık gibi özel gereksinimleri karşılarken Web ile entegrasyonu sağlayabilecek kolaylıkta tasarlanmıştır. CoAP paket

kaybının tolere edilip edilmemesine bağı olarak güvenilir olmayan ve güvenilir çalışma modları sunmaktadır [12]. CoAP mimarisi şekil 2’de verilmiştir.



Şekil 2. CoAP Mimarisi

MQTT ve CoAP kısıtlı cihazlar için IoT protokolü olarak kullanılabilir. Bu iki protokolün karşılaştırılmasında ağ koşullarının önemi bulunmaktadır. Örneğin ağın kayıp oranı düşükse CoAP MQTT’den daha az ek yük getirir. Fakat MQTT TCP üzerinden çalıştığı için CoAP’a göre daha güvenilir bir protokoldür [14]. Çalışmamızda bu bölümde anlatılan yöntemler kullanılmakta olup, merkezi bir yönetim sistemi tasarlanmış ve görsel ara yüzle desteklenmiştir.

III. İLGİLİ ÇALIŞMALAR

Tomasic ve arkadaşları uzaktan sağlık izleme işlemleri için Contiki-NG işletim sistemi kurulu Openmote gömülü cihazların üzerinde CoAP protokolünü kullanmıştır. Verileri görebilmek için Firefox web tarayıcısı eklentisi ve Node-RED uygulamasını kullanmışlardır. Verilere istenildiği takdirde sadece bu uygulamalar üzerinden erişilebilmektedir [2].

Esquiagol ve arkadaşları IoT teknolojisini kullanarak iç mekan hava kalitesini ölçme amaçlı bir ağ kurmuşlardır. CoAP protokolünü kullanarak verilerin alındığı bu platformda görsel bir arayüz bulunmamakta olup son kullanıcıya hitap eden bir arayüz görülemez [15].

Khlaifi ve arkadaşları verimli ağ kapsamı ve bağlanabilirlik gerektiren bir kablosuz algılayıcı ağı kullanan askeri sınır izleme uygulaması için yönlendirme protokolleri hakkında bir araştırma sunmuşlardır. Uygulamaları için yönlendirme protokollerindeki gelişmeler sırasında, önemli protokollerden AODV (Ad hoc on Demand Distance Vector) ve RPL örneğini göz önünde bulundurarak, enerji tüketimini ve ağ içinde dolaşan bilgi akışını optimize etmeyi amaçlamışlardır. Ardından, RPL ve AODV protokollerinin performansını karşılaştırmışlardır. Sonuçların, enerji tüketimi açısından RPL protokolünün WSN/IoT uygulamaları için daha uygun olduğunu gösterdiğini belirtmişlerdir [16].

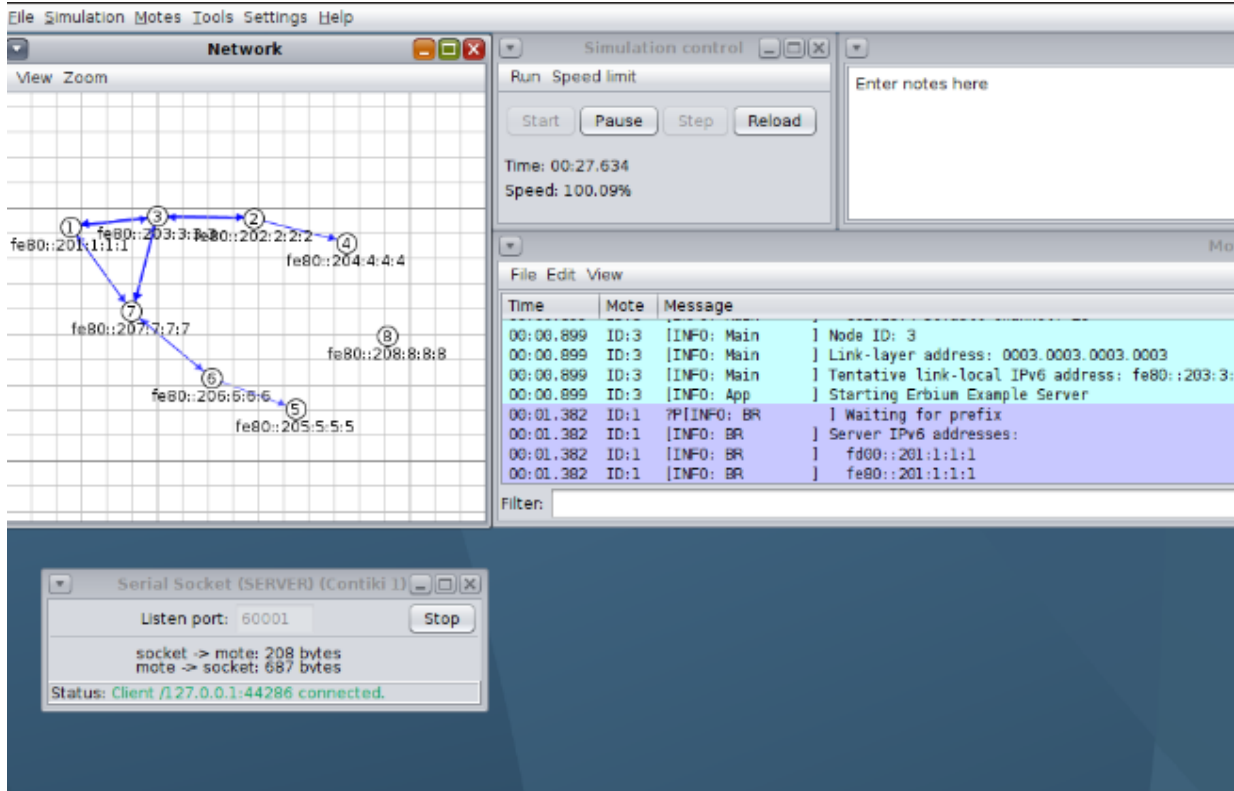
Plageras ve arkadaşları IoT tabanlı, Contiki işletim sistemli cihazlarla akıllı binalarda veri toplama ve işlemeyi ele almışlardır. Tasarladıkları sistemin benzetimini Cooja benzetim programıyla yapmışlar, algılayıcı verilerine Firefox tarayıcısı üzerinden erişebileceklerini belirtmişlerdir [17].

III. YÖNTEM VE TEKNİK

Önerilen IoT sisteminde, Contiki-NG işletim sistemine sahip, düşük güçlü, gömülü cihazlar için RPL yönlendirme algoritmasını kullanan, çok atlamalı bir veri toplama ve kontrol mekanizması sağlanmıştır. Uygulama birçok alanda kullanılabilir nitelikte olup merkezi kontrol sistemiyle uzaktan izlenebilmekte, tetikleyiciler oluşturulabilmektedir. Yerel ağ içerisinde alt katmanda RPL yönlendirme

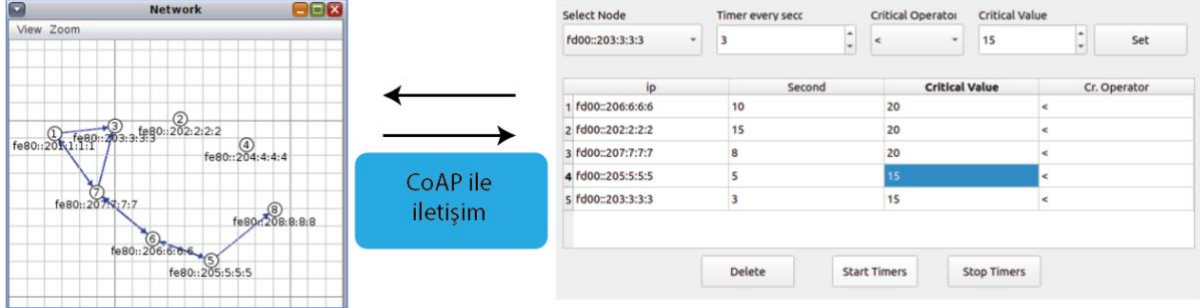
protokolü kullanılırken, taşıma katmanında CoAP ve MQTT protokolü kullanılmaktadır. Merkezi kontrol uygulaması ile istenildiği zaman herhangi bir düğümdeki algılayıcı verisi CoAP istekleri ve yanıtlarıyla alınabilmekte, hatta belirli aralıklarla veri toplaması da sağlanabilmektedir. Geliştirdiğimiz uygulama Contiki-NG işletim sistemi yüklü kenar yönlendirme cihazının seri portu üzerinden düğümlere çok atlamalı bir şekilde ulaşabilmektedir. Uygulamanın görsel arayüzünde zamanlayıcılar oluşturulabilmekte, bu zamanlayıcılar sayesinde düğümlerden istenilen aralıklarla veriler toplanabilmektedir. Hatta yine kullanıcı tarafından oluşturulan bir eşik değeri kontrolüyle verilerin istenilen aralıklarda olmaması halinde uyarı amaçlı uzak sunucuya gönderilebileceği bir ortam da sağlanmış bulunmaktadır. Yerel ağ üzerinde CoAP ile haberleşme sağlanırken, uzak sunucuya gönderim amaçlı MQTT protokolü kullanılmaktadır. Genel işlemler yerel ağda gerçekleştirilirken, kenar işleme sayesinde elenen kritik veya uzak sunucuya gönderilmesi istenilen veriler uzak sunucuya gönderilmektedir.

Uygulama benzetimi yapılırken Contiki-NG içerisindeki araçlardan Cooja benzetim programı kullanılmış olup, Şekil 3'te Cooja benzetim programı ekranı görülebilmektedir. Cooja benzetim programı ile düğümler istenilen bölgelere eklenebilmekte, düğümler içerisinde C programlama dili ile geliştirilmiş programlar çalıştırılabilmektedir.



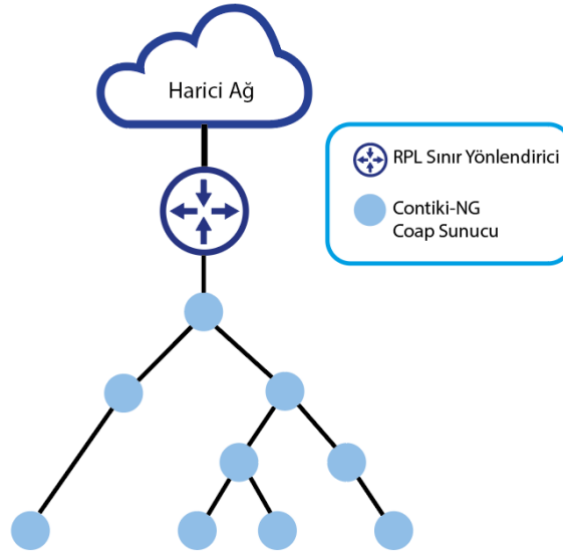
Şekil 3. Cooja Benzetim Programı

Şekil 4'teki görselde görüleceği üzere benzetim programındaki düğümlerle iletişim kurulabilmesi amaçlı bir görsel arayüz geliştirilmiştir. Görsel ara yüzü oluştururken Qt Arayüz Uygulaması kullanılmış olup arka planda Python ile kodlama yapılmıştır.



Şekil 4. Görsel Arayüzlü Uygulama ve Cooja Benzetim Programı

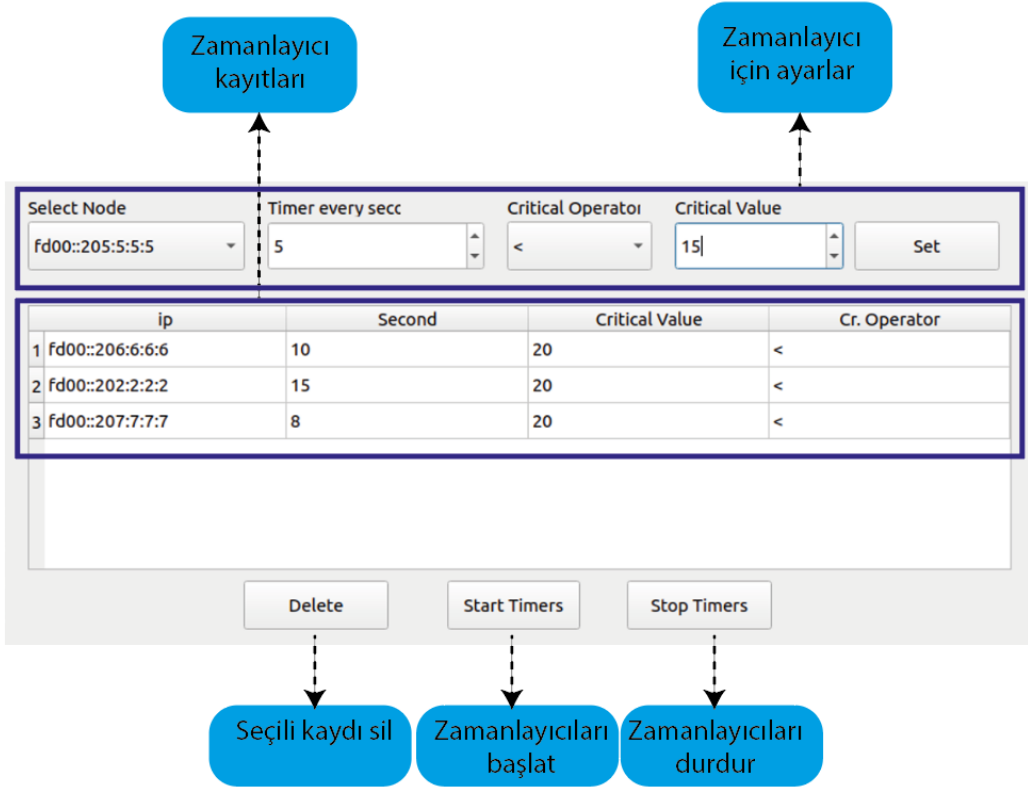
Görsel uygulamamız benzetim programındaki RPL kenar yönlendiricinin seri portu üzerinden iletişim sağlamaktadır. Bütün düğümlere gönderilen mesajlar bu düğüm üzerinden geçmektedir. Algılayıcı düğümleri üzerinde de CoAP Server uygulaması çalışmaktadır. Şekil 5'te ağın genel yapısı görülmektedir.



Şekil 5. RPL ağı ve düğümler

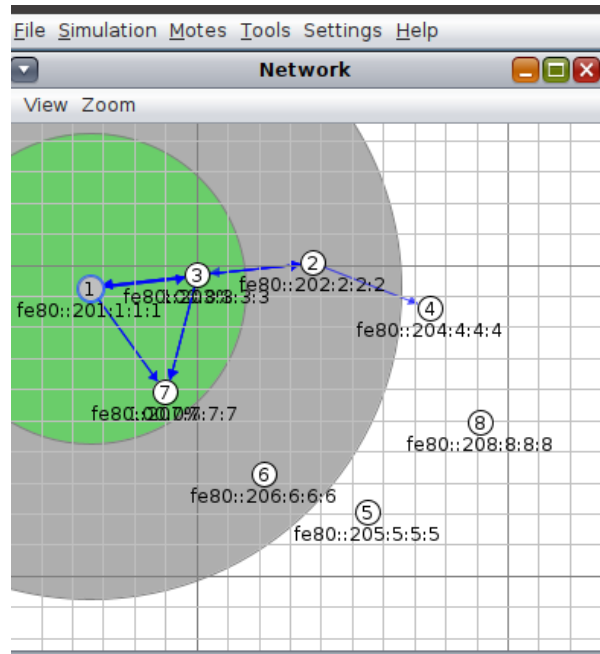
Uygulamamızda benzetim programına eklenen düğümlerin IPv6 numaraları görülmektedir. Düğümler üzerinde CoAP sunucu uygulaması çalışmakta olup uygulama üzerinden istek yapılması halinde düğüm algılayıcı verisini göndermektedir. Alınan veriler alındığı zaman bilgisi ile birlikte SQLite veritabanına kaydedilmekte, ızgara görünümü içerisinde veriler gösterilmektedir.

Uygulamamızda çeşitli senaryolarda kullanılacak bir zamanlayıcı sistemi geliştirilmiştir. İstenilen düğüme, istenilen zaman aralıklarıyla istek gönderilebilmekte, algılayıcı verisi alınabilmektedir. Algılayıcılardaki programda herhangi bir değişiklik yapmaya gerek kalmadan merkezi kontrollü uygulamamızla kolay bir şekilde bu sağlanabilmektedir. Bu özellik bizim geliştirdiğimiz uygulamamızı özgün kılmaktadır. İncelediğimiz kadarıyla literatürde buna benzer bir örneğe rastlanmamıştır. Zamanlayıcı ekranı Şekil 6'da görülmektedir.



Şekil 6. Zamanlayıcı Ekranı

Zamanlayıcı ekranında verisi istenilen düğüm, kaç saniye aralıklarla veri alınacağı, varsa kritik durum operatörü ve değeri bilgisi girilmektedir. Şekil 6'daki örnekte gelen verinin 15'in altında olması durumunda uzak sunucuya bilgi mesajı gönderilmektedir. Zamanlanmış işlemler zamanlayıcı ekranındaki alt bölümde gözükmekte, istenildiği takdirde zamanlayıcının başlatılması, durdurulması, silinmesi işlemleri de düğmeler yardımıyla yapılabilmektedir.



Şekil 7. Cooja RPL İletişimi Görünümü

RPL yönlendirme protokolü sayesinde bir düğüme gidecek mesaj, sadece ilgili düğüme giden yol üzerindeki düğümler üzerinden gitmektedir. Diğer düğümlere gereksiz mesaj gönderimi yapılmamaktadır. Örneğin Şekil 7’de görüldüğü üzere 2 nolu düğüme veri iletilirken 6, 5, 8 nolu düğümlere mesaj uğramamaktadır.

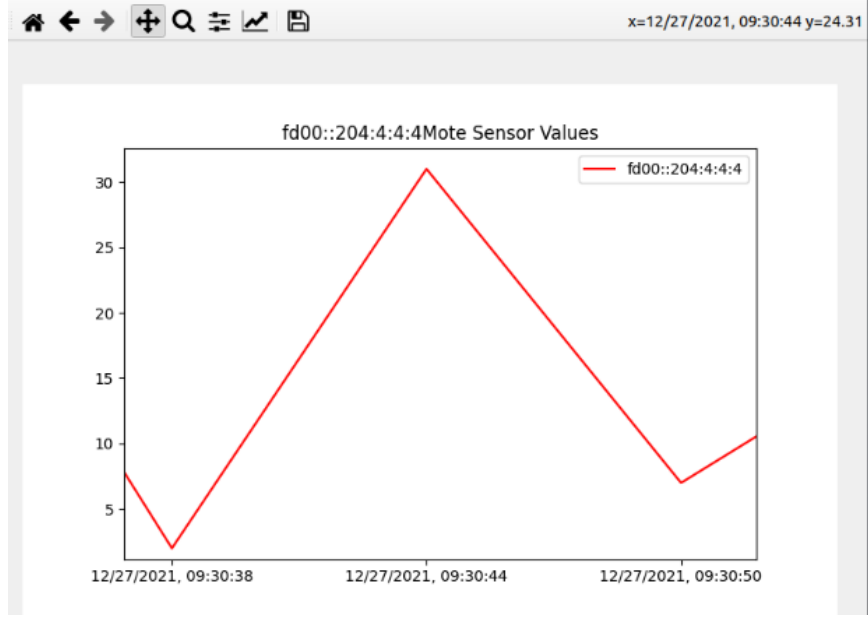
Tüm algılayıcı verileri yerel veritabanında saklanır ve uygulamamızın ana formunda görülebilir. Şekil 8’de örnek bir ekran verilmiştir.

The screenshot displays the application's main interface. At the top, there is a section titled "Notes" containing a list of node identifiers: fd00::202:2:2:2, fd00::203:3:3:3, fd00::204:4:4:4, fd00::205:5:5:5 (highlighted in blue), fd00::206:6:6:6, and fd00::207:7:7:7. To the right of this list are three buttons: "Get Temperature", "Set Timer", and "Graph of Mote". Below the "Notes" section is a table titled "Sensor Values" with three columns: "ip", "value", and "time". The table contains eight rows of data, with the first row highlighted in blue.

	ip	value	time
1	fd00::205:5:5:5	32	12/27/2021, 08:40:57
2	fd00::206:6:6:6	16	12/20/2021, 02:18:50
3	fd00::202:2:2:2	9	12/20/2021, 02:18:50
4	fd00::207:7:7:7	13	12/20/2021, 02:18:44
5	fd00::206:6:6:6	18	12/20/2021, 02:18:40
6	fd00::207:7:7:7	0	12/20/2021, 02:18:36
7	fd00::202:2:2:2	20	12/20/2021, 02:18:35
8	fd00::206:6:6:6	25	12/20/2021, 02:18:29

Şekil 8. Uygulama Ana Ekranı Algılayıcı Verileri

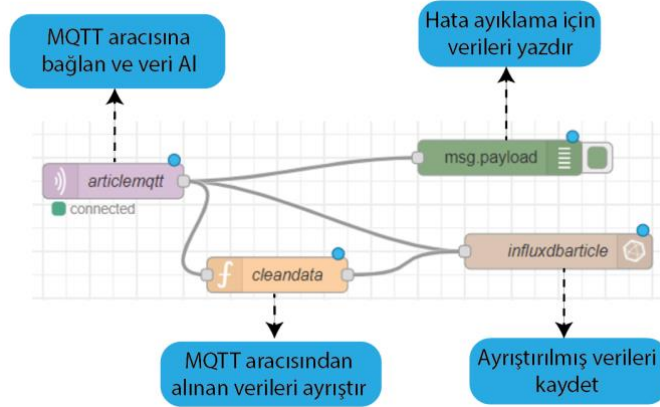
Geliştirilen uygulamada bu verileri grafik üzerinde görsel olarak da görmek mümkündür. Görselleştirme için Matplotlib eklentisi kullanılmış ve Şekil 9'da bir örnek verilmiştir.



Şekil 9. Zamana göre düğüm algılayıcı verileri

Kritik verilerin uzak sunucuya gönderilmesi amaçlı Digital Ocean [18] üzerinden Hollanda merkezli sanal özel sunucu kiralanmış olup üzerinde kullanılan teknolojiler IoT'nin yeni teknolojilerinden olan Node-RED, Mosquitto MQTT broker, InfluxDB ve Grafana'dır. Sanal özel sunucu üzerinde bu uygulamalar çalıştırılmıştır.

Node-RED ile algoritma adımları oluşturulurken, Mosquitto, MQTT mesajlarını alıp gönderme işlevine sahip bir sağlayıcıdır.



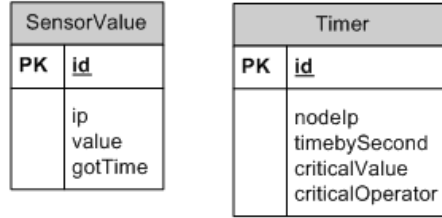
Şekil 10. Uygulamamızın Node-RED Akış Şeması

Şekil 10'da görüleceği üzere Node-RED üzerinde gelen MQTT mesajlarının temizlenerek InfluxDB veritabanına kaydedilmesi sağlanmıştır. InfluxDB veritabanındaki verilerin gösterilmesi amaçlı da Grafana grafik görselleştirme uygulaması kullanılmaktadır. Benzetim üzerindeki düğümlerdeki kritik değerler, kritik operatöre göre uzak sunucuya gönderilmiş ve görselleştirmesi sağlanmıştır. Şekil 11'de bulut tarafında çalışan Grafana uygulamasına ait görsel gözükmemektedir. Kritik veriler burada anlık olarak takip edilebilmektedir.



Şekil 11. Grafana üzerinde kritik verilerin gösterilmesi

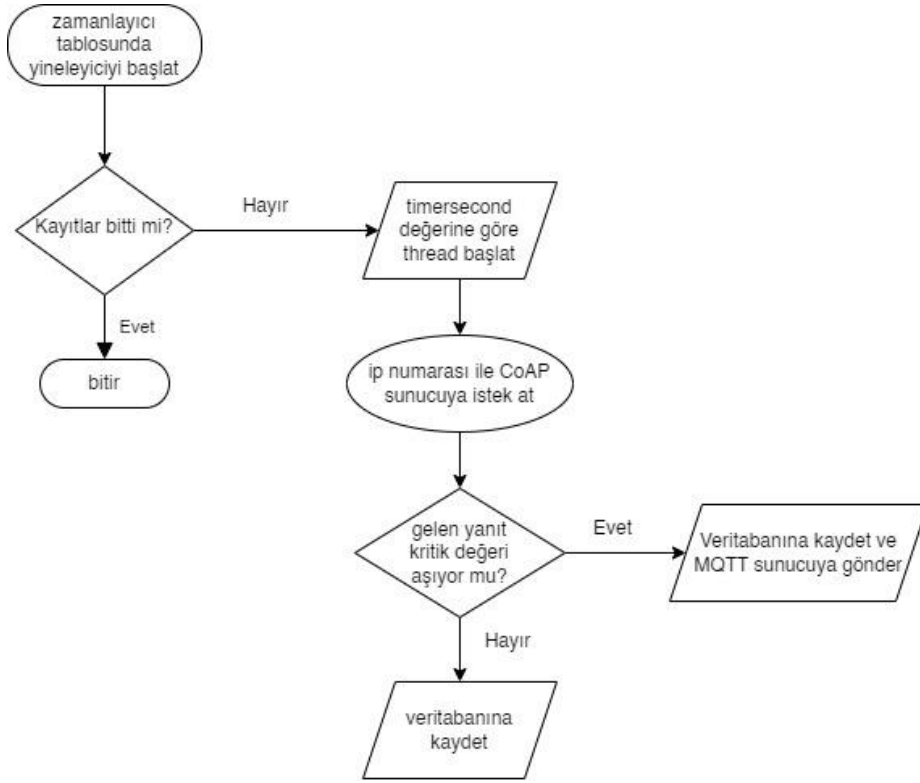
Şekil 11'de görüldüğü gibi, sarı düğüm 3'te (Node3) belirli bir eşik değerinin üzerindeki veriler kritik olarak kabul edilirken, diğer düğümlerde eşik değerinin altındaki veriler kritik olarak kabul edilir. Daha önce de belirttiğimiz gibi bu değerler görsel arayüzden kolaylıkla ayarlanabilmektedir. Uygulamamızın yerel veritabanı diyagramı Şekil 12'de gösterilmektedir.



Şekil 12. Veritabanı-Tablolar Diyagramı

“SensorValue” tablosunda düğüm IP numarası, değeri ve zamanı, “Timer” tablosunda da tetikleyici bilgileri yer almaktadır. Tetikleyiciler başlatıldığı zaman istenilen aralıklarda düğümlerden veriler alınmakta ve işletilmektedir.

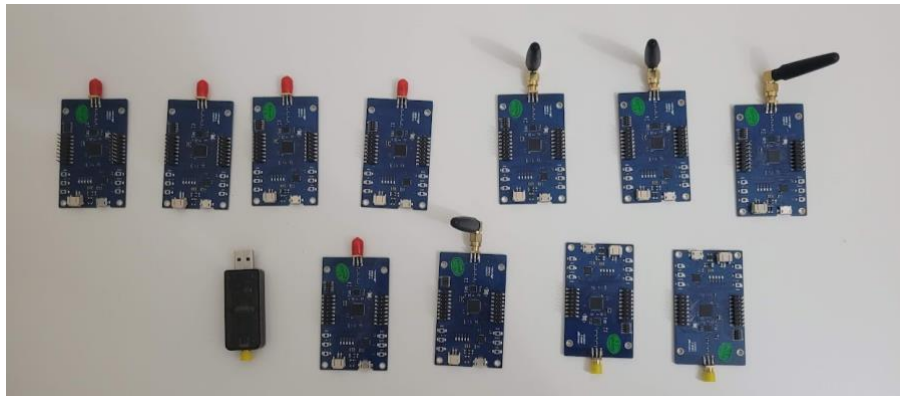
“SensorValue” tablosu düğüm IP numarasını, veri değerini ve alınan zamanı içerirken, “Timer” tablosu tetikleme bilgilerini içermektedir. Tetikleyiciler başlatıldığında düğümlerden istenilen aralıklarla veri alınır ve çalıştırılır. Zamanlayıcıların çalışmasının akış şeması Şekil 13'te gösterilmiştir.



Şekil 13. Zamanlayıcı algoritması akış şeması

Zamanlayıcı bilgileri veri tabanındaki tablodan satır satır alınır ve her bir satır verisi için bir iş parçacığı oluşturulur. İş parçacığı programda tıkanıklık olmaması amaçlı kullanılmıştır. Oluşturulan iş parçacığı üzerinde belirlenen zaman aralıklarında düğümlere CoAP isteği gönderilir. CoAP sunucusu görevi gören düğümler istek geldiğinde algılayıcı verisini gönderir. Gelen veri eğer kritik düzeydeyse buluttaki MQTT sunucusuna gönderilir.

Contiki-NG işletim sisteminde çalışan düğüm yazılımlarını test etmek için CC2538 geliştirme kitleri kullanılmıştır ve Şekil 14’de görülmektedir.



Şekil 14. CC2538 Geliştirme Kitleri

IV. SONUC

Bu çalışmada, esnek yapısı sayesinde geliştirilen uygulamanın sağlık, tarım, hayvancılık, hava kirliliği gibi birçok alanda kullanılabileceği gösterilmiştir. Görsel arayüz sayesinde son kullanıcı kolayca programda ayarlamaları yapabilmektedir. Ayrıca merkezi kontrollü sistem sayesinde program üzerinde sürelerin istenildiği gibi ayarlanabilmesi ile birlikte, gömülü cihazlara sürekli yazılım geliştirilmesinin de gerekliliğini ortadan kaldırmış olmaktadır.

Uygulamamız görsel arayüze sahip olması ve esnek yapısı sebebiyle birçok alana uygulanabilecek nitelikte bulunmaktadır. Benzetim üzerinde sorunsuz çalışması, verilerin düzgün bir şekilde iletilmesi de uygulamamızın kullanılabilirliğini göstermektedir.

TEŞEKKÜR: Bu çalışma Düzce Üniversitesi Bilimsel Araştırma Projeleri tarafından desteklenmiştir (Proje no: 2022.06.01.1307).

V. KAYNAKLAR

- [1] Y. Kim and J. Paek, “NG-RPL for efficient P2P routing in low-power multihop wireless networks,” *IEEE Access*, vol. 8, pp. 182591–182599, 2020.
- [2] I. Tomasic, K. Khosraviani, P. Rosengren, M. Jornten-Karlsson, and M. Linden, “Enabling IoT based monitoring of patients’ environmental parameters: Experiences from using OpenMote with OpenWSN and Contiki-NG,” *2018 41st Int. Conv. Inf. Commun. Technol. Electron. Microelectron. MIPRO 2018 - Proc.*, pp. 330–334, 2018.
- [3] H. Kharrufa, H. A. A. Al-Kashoash, and A. H. Kemp, “RPL-Based Routing Protocols in IoT Applications: A Review,” *IEEE Sens. J.*, vol. 19, no. 15, pp. 5952–5967, 2019.
- [4] B. R. Al-Kaseem, Y. Al-Dunainawi, and H. S. Al-Raweshidy, “End-to-end delay enhancement in 6LoWPAN testbed using programmable network concepts,” *IEEE Internet Things J.*, vol. 6, no. 2, pp. 3070–3086, 2019.
- [5] M. Silva, D. Cerdeira, S. Pinto, and T. Gomes, “Operating Systems for Internet of Things Low-End Devices: Analysis and Benchmarking,” *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10375–10383, 2019.
- [6] M. N. Napiyah, M. Y. I. Bin Idris, R. Ramli, and I. Ahmady, “Compression Header Analyzer Intrusion Detection System (CHA - IDS) for 6LoWPAN Communication Protocol,” *IEEE Access*, vol. 6, pp. 16623–16638, 2018.
- [7] A. H. Farea and K. Küçük, “Detections of IoT Attacks via Machine Learning-Based Approaches with Cooja,” *EAI Endorsed Trans. Internet Things*, vol. 7, no. 28, pp. 1–12, 2022.
- [8] X. Vilajosana, T. Watteyne, T. Chang, M. Vucinic, S. Duquennoy, and P. Thubert, “IETF 6TiSCH: A Tutorial,” *IEEE Commun. Surv. Tutorials*, vol. 22, no. 1, pp. 595–615, 2020.
- [9] R. K. Das, N. Ahmed, F. H. Pohrmen, A. K. Maji, and G. Saha, “6LE-SDN: An Edge-Based Software-Defined Network for Internet of Things,” *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7725–7733, 2020.

- [10] H. S. Kim, J. Ko, D. E. Culler, and J. Paek, “Challenging the IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL): A Survey,” *IEEE Commun. Surv. Tutorials*, vol. 19, no. 4, pp. 2502–2525, 2017.
- [11] H. Fujita, Y. Tanaka, K. Mori, and F. Teraoka, “Efficient RPL Tree Construction Using Passive Link Quality Estimation,” *Proc. - 2021 9th Int. Symp. Comput. Netw. Work. CANDARW 2021*, pp. 8–14, 2021.
- [12] R. Herrero, “Dynamic CoAP Mode Control in Real Time Wireless IoT Networks,” *IEEE Internet Things J.*, vol. 6, no. 1, pp. 801–807, 2019.
- [13] B. Mishra and A. Kertesz, “The use of MQTT in M2M and IoT systems: A survey,” *IEEE Access*, vol. 8, pp. 201071–201086, 2020.
- [14] S. Sinche *et al.*, “A Survey of IoT Management Protocols and Frameworks,” *IEEE Commun. Surv. Tutorials*, vol. 22, no. 2, pp. 1168–1190, 2020.
- [15] J. Esquiagola, M. Manini, A. Aikawa, L. Yoshioka, and M. Zuffo, “Monitoring Indoor Air Quality by using IoT Technology,” *Proc. 2018 IEEE 25th Int. Conf. Electron. Electr. Eng. Comput. INTERCON 2018*, pp. 18–21, 2018.
- [16] H. Khlaifi, A. Zrelli, and T. Ezzedine, “Routing Protocols for A Border Monitoring Application,” *2019 Int. Conf. Internet Things, Embed. Syst. Commun. IINTEC 2019 - Proc.*, pp. 1–6, 2019.
- [17] A. P. Plageras, K. E. Psannis, C. Stergiou, H. Wang, and B. B. Gupta, “Efficient IoT-based sensor BIG Data collection–processing and analysis in smart buildings,” *Futur. Gener. Comput. Syst.*, vol. 82, pp. 349–357, 2018.
- [18] “Digital Ocean Website.” <https://www.digitalocean.com/>. (18.01.2023)