

Received: 29.06.2020

Accepted:07.10.2020

Nesnelerin İnterneti için MQTT ile Hiyerarşik Haberleşme

Oğuz MISIR^{1*}, Levent GÖKREM²

^{1,2}Gaziosmanpaşa University, Faculty of Engineering and Architecture, Mechatronics Engineering Department, 60250, Tokat, Turkey

Özet

Çevremizde birçok amaca hizmet eden nesnelerin internet ağına bağlanarak aralarında çeşitli iletişim protokolleri ile haberleşmesi istenmektedir. Bu istek nesnelerin interneti (Internet of Things- IoT) kavramını günlük hayatımıza soktu. Her nesnenin internete bağlanması gereksinimi beraberinde çeşitli haberleşme ağı teknolojilerinin de gelişmesine sebep oldu. Sadece günlük hayatımızda değil endüstride kullanılan nesnelerin aralarında iletişim kurarak kendi kendini organize etme talebi ağ protokollerin gelişimine ivme kazandırdı. Çok yakın gelecekte milyarlarca cihazın haberleşmek isteği platformların esneklik, hız, güvenlik kavramlarının ön plana çıkması önemli bir etken haline dönüştü. Bu çalışmada TCP/IP alt yapısını kullanan bir IoT haberleşme protokolü olan MQTT (Message Quening Telemetry Transport) incelenmiştir. MQTT çalıştığı ağda bir karmaşa oluşturmayan hafif, esnek ve hızlı haberleşme yeteneğine sahip bir iletişim protokolüdür. Yayınlama/abone olma prensibini kullanır. İletilmek istenen mesajları bir konu başlığı altında toparlar ve bilgiye erişmek isteyen kullanıcılar yayımlanan konuya abone olarak veriye erişebilirler. MQTT 'nin konuları hiyerarşik olarak sınıflandırma yeteneği ile bilgiler, veri mesajlarının sayısının oldukça fazla olduğu ortamlarda mesajlara filtre özelliği kazandırır. Bu çalışmada MQTT protokolü kullanılarak CC3200 ve ESP8266 kablosuz mikro denetleyiciler yardımıyla birbirinden farklı konularda hiyerarşik olarak sınıflandırılmış bir yayımlama /abone olma örneği uygulanmıştır. Ayrıca farklı seviyelerde sınıflandırılmış mesajlara erişmek için "Paho" yazılımı kullanılmıştır. Birbirinden farklı 9 hiyerarşik sınıfta veri yayımlama/abone olma yeteneği incelenmiştir.

Anahtar Kelimeler: *IoT, MQTT, Yayınlama/Abone, cc3200, esp8266, Nesnelerin İnterneti*

Hierarchical Communication with MQTT for Internet of Things

Oğuz MISIR^{1*}, Levent GÖKREM²

Abstract

Every object that serves many purposes around us needs to be connected to the internet network and communicate by using various electronic communication protocols between them. This request placed the concept of Internet of Things (IOT) in my daily life. The necessity of connecting each object to the internet also caused the development of various communication network technologies. Beginning processes such as the ability to self-organize by communicating objects that are used only in our everyday lives, rather than just our daily lives, gave the impetus to the development of network protocols that concepts such as industry 4.0 require for a network. In the foreseeable future, the concepts of flexibility, speed and security have become an important factor in the platforms where billions of devices have to communicate with each other. In this work, we investigated MQTT (Message Quenching Telemetry Transport), an IoT communication protocol using TCP / IP infrastructure. MQTT is a communication protocol with lightweight, flexible and fast communication capability that does not create any confusion in the network it is running. Uses the principle of publishing / subscribing. Users who wish to access information can access the database by subscribing to the published topic. MQTT provides filtering features to messages in environments where there is

*Corresponding Author, e- mail: oguz.misir@gop.edu.tr

a large number of data messages to send, with the ability to classify topics hierarchically. In this study, using MQTT protocol, CC3200 and ESP8266 wireless microcontrollers were used to implement a sample publishing / subscribing example hierarchically categorized in different fields. In addition, "Paho" software is used to access messages classified at different levels. the ability to publish / subscribe to data in 9 different hierarchical levels is investigated.

Keywords: *IoT, MQTT, Publish / Subscribe, cc3200, esp8266.*

1. Giriş

Nesnelerin interneti İnterneti (IoT) terimi ile yaygın bir şekilde güncel olarak karşılaşılmaktadır. Nesnelerin interneti kavramındaki nesnelere, ortamı algılayabilen ve verileri kullanıcılara yada işbirliği kurabileceği nesnelere paylaşan bir iletişim ağını içermektedir [1]. Nesnelerin interneti terimindeki nesne tanımını kullandığımız her cihazı içermektedir. Bunlar buzdolabı, çamaşır makinesi, otomobil, otopark için dolu/boş algılayıcısı, sensörler, bir marketin kişi sayacı gibi sayabileceğimiz birçok örnekten oluşabilmektedir. Nesnelerin internetinde her bir nesnenin verileri algılaması ve kendi aralarında veya kullanıcılarla paylaşması bir akıllı uygulama ekosisteminin oluşumunu sağlayabilmektedir [2]. Bu ekosistemde akıllı şehirler, sağlık hizmetleri, üretim sektörü gibi bir çok alanda verilerin izlenebildiği ve bu verilere göre karar verme sürecinde önemli ölçüde kolaylık sağlandığı görülmektedir [3]. Özellikle mobil internet ağının gelişimi, çevremizdeki kullandığımız her nesnenin(sensörler, ev aletleri, trafik bilgisi gb.) kolayca internete erişmesini sağlamaktadır [4]. Nesnelerin internete erişimi ile her bir nesnenin verisinin kullanıcılara ve kendi aralarında paylaşımı için internet erişimi önem taşımaktadır [5]. İnternete erişebilecek nesne sayısının artmasıyla birlikte verilerin internetteki dolaşımı büyük bir veri trafiğine yol açabilmektedir. Bu sebepten internet ağına bağlanan her bir nesne olabildiğince az bir şekilde interneti az veri yüküne tabi tutmalıdır. İnternet ağına bağlanan nesnelerin kullanacakları ağ alt yapısına bağlı olarak kullanacakları protokoller büyük önem taşımaktadır.

İnternet ağına katılan bileşenler TCP/IP alt yapısını kullanarak haberleşmektedir. Bu ağ alt yapısı altında IoT(Internet of Things) protokolleri TCP/IP desteğine ihtiyaç duyarlar. Bu kapsamda IoT uygulamalarını genişlemesiyle CoAP(Constrained Application Protokol), REST(Representational State Transfer) XMPP(Extensible Messaging and Presence Protokol), AMQP(Advanced Message Queuing Telemetry Transport), MQTT(Message Queue Telemetry Transport) gibi çeşitli protokol uygulamaları geliştirilmiştir [6]. Bu uygulamaların en önde gelen özelliklerinin başında nesnelerin interneti tanımlamasına uygun nesne donanımlarını destekleyebilmesidir. IoT protokollerinden beklenen özelliklerin başında sunucu ve istemciler arasında olabildiğince kolay bir haberleşme yöntemi kullanması, mobil ve küçük işlem kapasitesine sahip cihazlar üzerinde verimli çalışabilmesi, işlem gördükleri cihazlar üzerinde az enerji harcayabilmesi beklenmektedir. IoT protokolü standardına sahip olan MQTT(Message Quening Telemetry Transport) protokolü hafif ve esnek oluşu ile uygulamalarda oldukça çok tercih edilmektedir [7]. MQTT ile haberleşirken istemciler ve sunucu arasında bir IP numarasına ihtiyaç duyulmadan nesnelere arasında haberleşme yapılabilmektedir. MQTT ile yayınlama/abone olma prensibine göre konu tabanlı haberleşme yapılmaktadır [8]. MQTT 'de sunucu yerine arabulucu olarak adlandırılan "Broker" kullanılmaktadır. Arabulucu istemcilerin oluşturdukları konulara etiketlenen mesajları, konulara abone olan istemcilere filtre ederek dağıtma işlemini yapmaktadır. Konu tabanlı haberleşme geniş, ölçeklenebilir bir algılayıcı tasarımının kurulmasını sağlayabilmektedir. Ayrıca MQTT konuları mesaja özgü konu etiketine sahip olduğundan dolayı, bir istemcinin hem birden fazla konu hakkında aynı anda yayın yapılmasına ve abone olunmasına izin vermektedir. Konu tabanlı haberleşme mesajların doğrudan arabulucu sayesinde ayrıştırılıp konu seviyelerine bağlı olarak dağıtılmasını sağlamaktadır. Bu özellik mesajların seviyeleme yaparak sınıflandırılmasını sağlamaktadır.

MQTT protokolü IBM tarafından ortaya çıkmıştır [9]. Başlangıçtaki hedefi petrol boru hatlarındaki algılayıcıları uydulara bağlanmasıydı [10]. Protokolün en önemli özelliklerinin başında düşük işlem kapasitesine sahip cihazlarda ve bant genişliği düşük ağlarda verimli bir şekilde çalışabilmesidir. MQTT günümüzde birçok kablolu ve kablosuz haberleşme yeteneğine sahip mikro denetleyiciler tarafından desteklenmektedir. Ayrıca bu duruma bağlı olarak ticari ve ticari olmayan çeşitli sunucular tarafından arabulucu hizmeti alma bilmektedir [11].

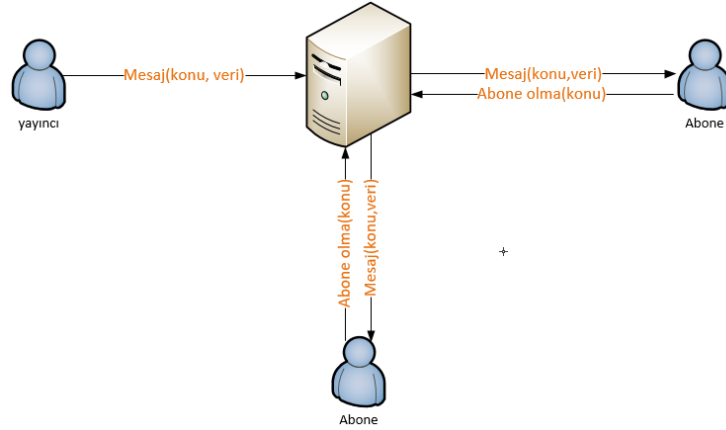
Bu çalışmada MQTT protokolünün yayımlama/abone olma haberleşme yöntemi kullanılarak konu tabanlı haberleşme uygulaması incelenmiştir. Ayrıca farklı seviyelerde konular oluşturulmuştur. Aynı ve farklı seviyelerde konular hakkında veri paylaşımı yapılmış buna bağlı olarak sadece aynı seviyedeki konulara veri gönderme veya abone olma sağlanmış tek seviyeli haberleşme yapılmıştır. Benzer olarak farklı seviyelerde birçok konuya abone olma sağlanarak çok seviyeli haberleşmede yapılmıştır. 2. bölümde MQTT protokolü tanıtılmaktadır. 3. bölümde MQTT protokolü ile tek ve çok seviyeleri konu hiyerarşi uygulaması yapılmıştır.

2. Materyal ve Metot

2.1. MQTT

MQTT yayımlama/abone olma metodolojisine dayanan IBM 'in geliştirdiği bir haberleşme protokolüdür [9]. OASIS tarafından standartlaştırılmıştır [12]. TCP-IP altyapısını kullanarak haberleşme yapmaktadır [8]. MQTT 'nin en önemli özelliklerinin başında hafif yani protokollün kullanıldığı cihazlar arasında ve ağda bir yük/karmaşa oluşturmayacak yeteneğe sahip olmasıdır. Hafif düzeyde haberleşme yapan protokolün bu yeteneği az enerji tüketimi, hız gibi ek özellikleri de beraberinde getirmiştir. MQTT 'nin yayımlama/abone olma prensibi konu merkezli veri alışverişine dayanmaktadır. Konu tabanlı haberleşme daha önceden tanımlanan bir konuya abone olarak, konuyla ilgilenen birimin konuya abone olarak veri alışverişi yapılmasıdır [14].

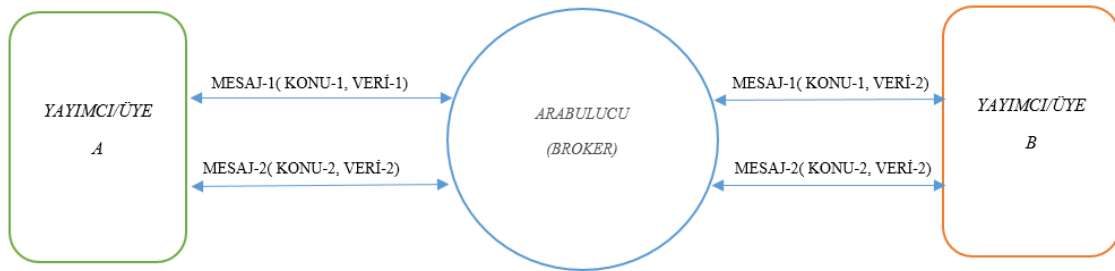
MQTT protokolü istemci ve yönlendirici/arabulucu unsurlarından oluşmaktadır. İstemci tarafından gönderilen mesajlar arabulucu/broker yoluyla denetlenir ve iletir. İstemci tarafından gönderilen her bir mesaj bir konuya etiketlenir. Arabulucu tanımlanan konuya abone olan istemcilere mesajları ayırır ve gönderir. Mesajını ilgili alıcıya iletmek isteyen istemci arabulucuya göndermek istediği mesaj konusu kaydettirmelidir. Bir konuya bir istemci abone olabiliyorken birden fazla istemci de abone olabilmektedir. Şekil 1. 'de gösterildiği gibi verinin paylaşılacağı yayıncı ilgili konuyla birlikte veriyi göndermektedir. Arabulucu yoluyla aboneler ilgili konuya üye olur ve üyeler üye olunan konu ile ilgili veri alışverişi yapabilmektedir. Konuya abone olan istemciler arasında konu tabanlı haberleşme uçtan uca (Peer to Peer/P2P) hiyerarşik olarak yapılabilmektedir. Konu tabanlı mesaj yönlendirme prensibi MQTT protokolünün ana mekanizmasını oluşturmaktadır.



Şekil 1. MQTT Mesaj Yayını

MQTT IoT (Internet of Things) ve M2M (Machine to Machine) uygulamaları için tercih edilebilir nitelik taşımaktadır. Gömülü sistem uygulamalarında enerji ve ileti performansı verimliliği http (Hyper Text Transfer Protocol) gibi protokollerle karşılaştırıldığında MQTT daha iyi performans vermektedir [15]. Ayrıca MQTT ile benzer özellikler taşıyan CoAP (Constrained Application Protocol) [16] yoğun veri alışverişi performansı MQTT 'den daha düşüktür [17]. Bu sayede MQTT küçük işlem kapasitesine sahip mikro denetleyiciler için hafif ve kullanışlı hale gelmektedir.

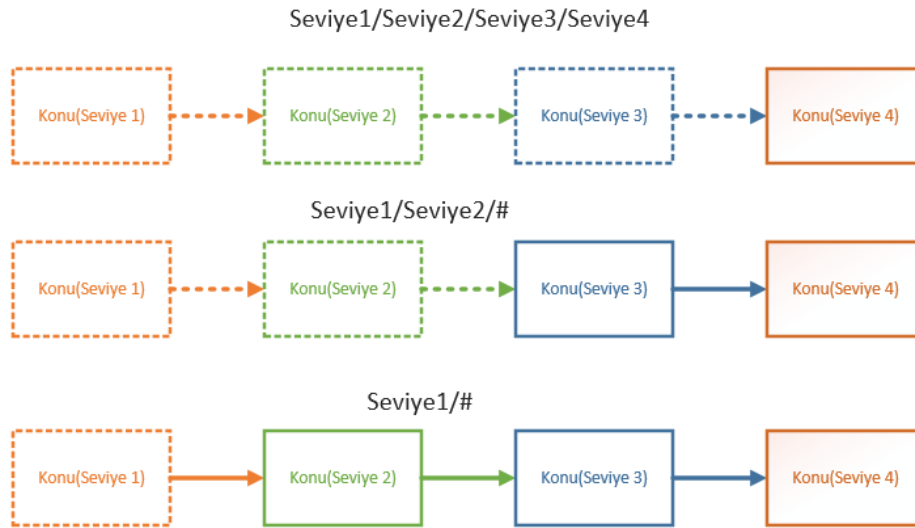
Bir MQTT yayıncısı başarılı bir bağlantı yapabilmesi için arabulucu (Broker) ile iletişime geçmelidir. Arabulucudan gelen yanıtla bu bağlantı sağlanmaktadır. Yayıncı paylaşmak istediği ilgili konuyu arabulucuya kaydettirmelidir. Konu kaydı yapıldıktan sonra yayıncı yayınladığı konu ile veri paylaşımını abonelere yapabilmektedir. Bağlantıyı sonlandırmak isteyen yayıncı arabulucuya konuyu veya haberleşmeyi sonlandırdığını bildirmektedir. Eğer üye /yayıncılar ve arabulucu arasında bir bağlantı sonlandırma paylaşımı veya belirlenen sürede haberleşme yapılmamışsa broker tarafından bağlantının koptuğu bildirilmektedir [18]. MQTT konuları veri paylaşımının karşılıklı bir şekilde yapılabilmesi için konuların üyeler tarafından bilinmelidir. Bu durum yayıncıların da üye olmak istedikleri ilgilendikleri konudan veri alışverişi yapabilmelerini sağlayabilmektedir. Bir konu hakkında veri paylaşan bir yayıncı başka bir konu hakkında veri alabilmektedir. Bu durum Şekil 2. 'de gösterildiği gibi istemcilerin iletilmek istenen mesajların bir istemciden birden fazla konu paylaşımına ve aynı istemciden birden fazla konuya abone olmalarını sağlayabilmektedir.



Şekil 2. Birden fazla yayıncının aynı anda farklı konularda üye ve yayıncı şeması.

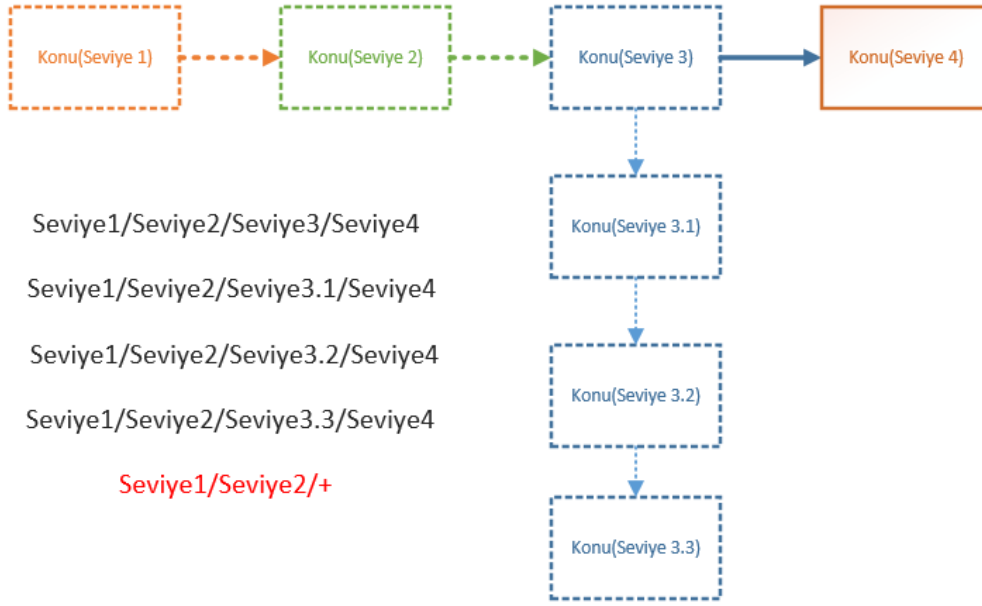
2.2. MQTT Mesajlarının Konu Hiyerarşisi

MQTT 'de mesajlar bir konuya abone olunan abonelere iletilebilmektedir. Konular mesaj alışverişinde filtre amacı ile kullanılmaktadır. Aboneler konular vasıtası ile seviyelere ayrılabilir. Bu durum abonelerin amacına hitap eden konulara üye olabilirken amacı dışında kalan konulara mesaj erişimi kısıtlana bilmektedir. MQTT mesajları ayraç karakteri “/” ile seviyelere ayrılabilir. Örnek olarak “Market/Gıda/içecekler/sudolabı/sıcaklık” gibi bir konu oluşturulabilir [18]. Bu örnekte bir marketin gıda ünitesinin içecekler bölümünün su dolabının sıcaklık parametreleri konusuna üye olunarak bu parametre hakkında bilgiye erişilebilir. Abone konusu özel karakterler kullanılarak birden fazla konuya abone olunabilir. MQTT 'de bu ayırım için çok seviyeli ayırım olarak isimlendirilmektedir. Çok seviyeli ayırım için “#” karakteri kullanılmaktadır. Bir konudaki herhangi bir sayıda konuyla eşleşen karakteri temsil etmektedir. Bir konudaki Şekil 3. 'de görüldüğü gibi Seviye1, Seviye2, Seviye3, Seviye4 şeklinde hiyerarşik olarak oluşturulan bir konu “#” karakteri ile Seviye1 karakterinin sonuna yerleştirilen “#” karakteri ile Seviye1 ve altındaki konuları kapsayabilmektedir. Yine aynı şekilde Seviye2 ve alt seviyeler, Seviye2 konusunun ayırım karakteri sonuna yerleştirilmiştir. Bu sayede Seviye2 ve alt seviye konularına erişilebilir. ”#” karakteri sadece seviye ayırım “/” karakterinden sonra istenilen kapsam konularına göre yerleştirilebilir. Bu sayede “#” karakteri alt seviyelerinde bulunan bütün konulara abone olunabilir.



Şekil 3. “#” ile birden fazla konuya aynı anda kapsama

MQTT 'de birden fazla seviyede konuya abone olunabiliyorken aynı seviyede bulunan birden fazla konuya da abone olunabilir. Aynı seviyede birden fazla konuya abone olabilmek için seviye ayırım karakteri “/” sonuna “+” karakteri kullanılmaktadır. Şekil 4. 'de görüldüğü gibi dört farklı seviyede yedi konu oluşturulmuştur. “Seviye1/Seviye2/Seviye3/Seviye4” 4 seviyeli konu için Seviye3 sınıfına bağlı aynı seviyede Seviye3, Seviye3.1, Seviye3.2, Seviye3.3 oluşturulmuştur. Örneğin “Seviye1/Seviye2/+” konusu Seviye3 ve aynı seviyedeki üç konuya da abone olunmasını sağlamaktadır. Bu abonelik aynı seviyede farklı konuları aynı anda abone olunmasını sağlamaktadır.



Şekil 4. Tek seviyeli “+” konu aboneliği

Hem çok seviyeli abone olma karakteri “#” hem de tek seviyeli abone olma karakteri “+” aynı anda kullanılabilir. Şekil 3. ‘de görüldüğü gibi örneğin “Seviye1/Seviye2/+/#” konusuna abone olan bir istemci 3. seviyede bulunan bütün konulara abone olabiliyor iken 3. seviye altındaki tüm seviyelerdeki konulara da abone olabilmektedir. Bu sayede hem aynı seviyedeki konulara abone olmayı hem de farklı seviyede abone olunması istenen konulardaki verilere erişilebilmektedir.

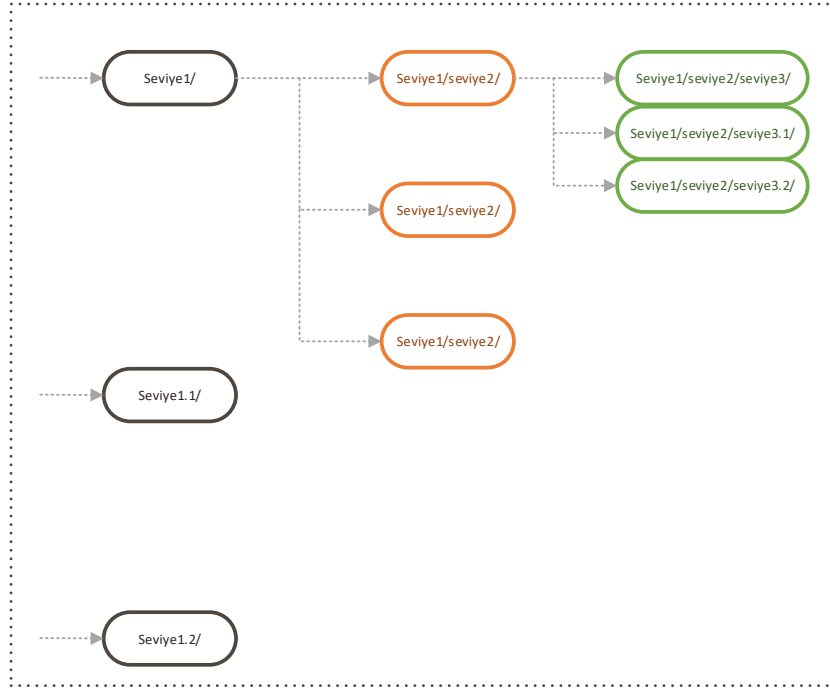
2.3. MQTT Mesaj İletimi Metotları

MQTT TCP/IP alt yapısı kullanmasına karşın QoS mesaj iletim metodu kullanmaktadır [19]. QoS, 3 ayrı iletim yöntemi ile çalışmaktadır. Bu metotlar QoS0, QoS1 ve QoS2 şeklinde sınıflandırılmıştır. QoS0 yöntemi, herhangi bir doğrulama gerektirmeden mesaj paketlerini iletmektedir. Mesajın iletilmediği kontrol edilmez. Sürekli gerçek zamanlı veri aktarılması gereken anlık sıcaklık, hız süreç denetim uygulamalarında kullanılabilir. QoS1 ile mesajlar denetimli bir şekilde aktarılma sağlanmaktadır. Bu yöntemde mesaj iletilene kadar paketler gönderilir ve denetimli veri alışverişi yapılmaktadır. QoS2 ise mesaj paketlerinin en az bir kez iletilmesini sağlar. Hem iletilerin ulaştırılmasını garanti eder hem de mesajları sadece alıcıya bir kez iletir. QoS 0 ve QoS 1 iletim yöntemlerinde mesaj birçok kez alıcılara iletelebilmektedir. MQTT ile istemci ve arabulucu (Broker) arasında veri iletimi güvenliği için SSL (Secure Sockets Layer) ve TLS (Transport Layer Security) kullanılarak mesajların şifreli olarak alıcılara iletilmesi sağlanabilmektedir. Ancak bu durum işlem kapasitesi düşük cihazlar için işlem yükü oluşturmaktadır [20].

3. Bulgular ve Tartışma

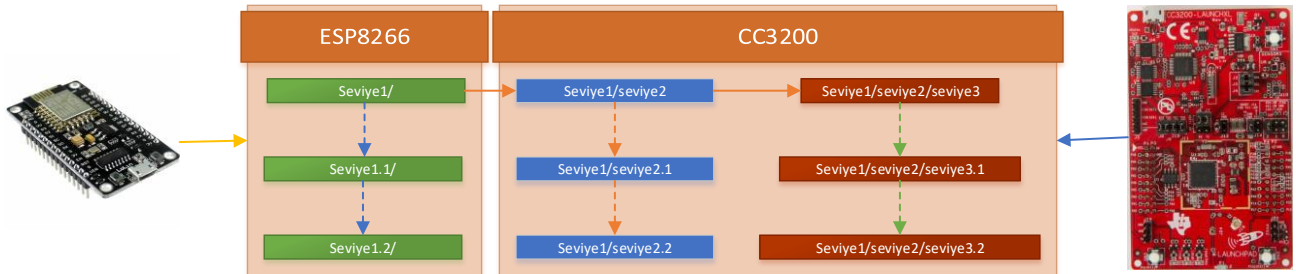
Çalışmada Şekil 5. ‘de gösterildiği gibi hiyerarşik seviyeli bir konu dağılımı için dokuz farklı konu oluşturulmuştur. Bu hiyerarşik konular 3 farklı seviyeden oluşmaktadır. 1. Seviye konular “Seviye1/”, “Seviye1.1” ve “Seviye1.2” şeklindedir. 2. seviye konular “Seviye1/” konusu altında “Seviye1/Seviye2”, “Seviye1/Seviye2.1” ve “Seviye1/seviye2.1” 3 alt seviyeden oluşmaktadır. 3. seviye konular “Seviye1/Seviye2” konusu altında “Seviye1/Seviye2/Seviye3”,

“Seviye1/Seviye2/Seviye3.1” ve “Seviye1/Seviye2/Seviye3.2” alt konularından oluşmaktadır. Her bir seviye altındaki konulara dolaylı olarak erişilebildiği gibi her seviyedeki konuya da doğrudan erişim sağlanabilmektedir.



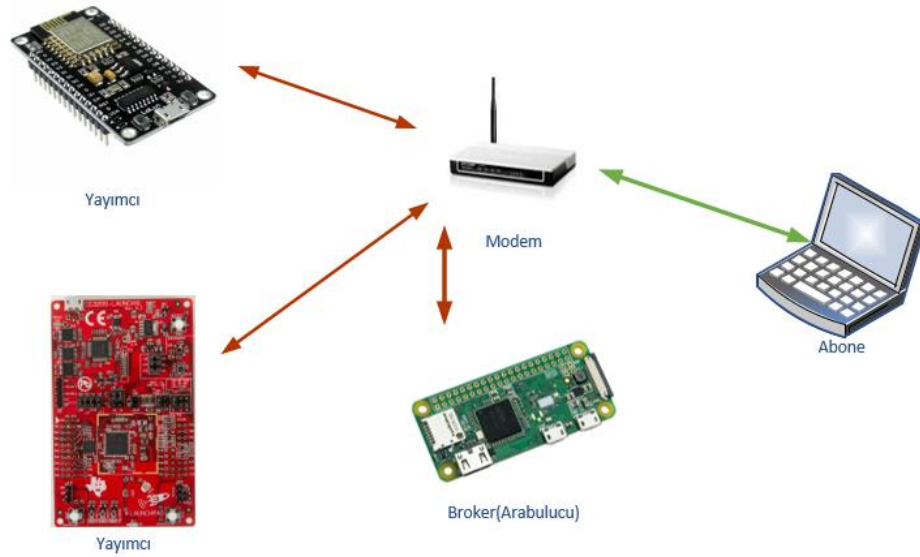
Şekil 5. Hiyerarşik konu sınıflandırma diyagramı

Şekil 6. 'da gösterildiği gibi yayımcı olarak 1. seviye konuları için Wemos NodeMCU V3 Lolin kit ve sonraki 2. ve 3. seviye konuları için Texas Instruments CC3200 Launch Pad [21] kiti kullanılmıştır. Seviye1, Seviye1.1 ve Seviye1.2 konuları için NodeMCU [22] arabulucuya konuları başlangıçta kaydetmektedir. Konularla ilgili olarak değişken iç sayaç değerini göndermektedir. CC3200 ile 2 farklı seviyede 6 konu için yayın yapılmaktadır. Seviye2, Seviye2.1, Seviye2.2 için CC3200 geliştirme kiti üzerinde bulunan BMA222 ivme algılayıcısının x eksen parametresi için “Seviye1/Seviye2/”, y eksen parametresi için “Seviye1/Seviye2.1/” ve z eksen parametresi için “Seviye1/Seviye2.2/” konusu oluşturulmuştur. Aynı geliştirme kiti üzerinde bulunan TMP006 sıcaklık algılayıcısı hem ortam sıcaklığı hem de sensor üzerindeki obje sıcaklığını ölçebilmektedir. Ortam sıcaklığı “Seviye1/Seviye2/Seviye3” konusu ile sensor üzerindeki obje sıcaklığı “Seviye1/Seviye2/Seviye3.1” konusu ile gönderilmektedir. Ayrıca test amaçlı olarak “Seviye1/Seviye2/Seviye3.2” üzerinden değişken sayaç değeri gönderilmektedir.



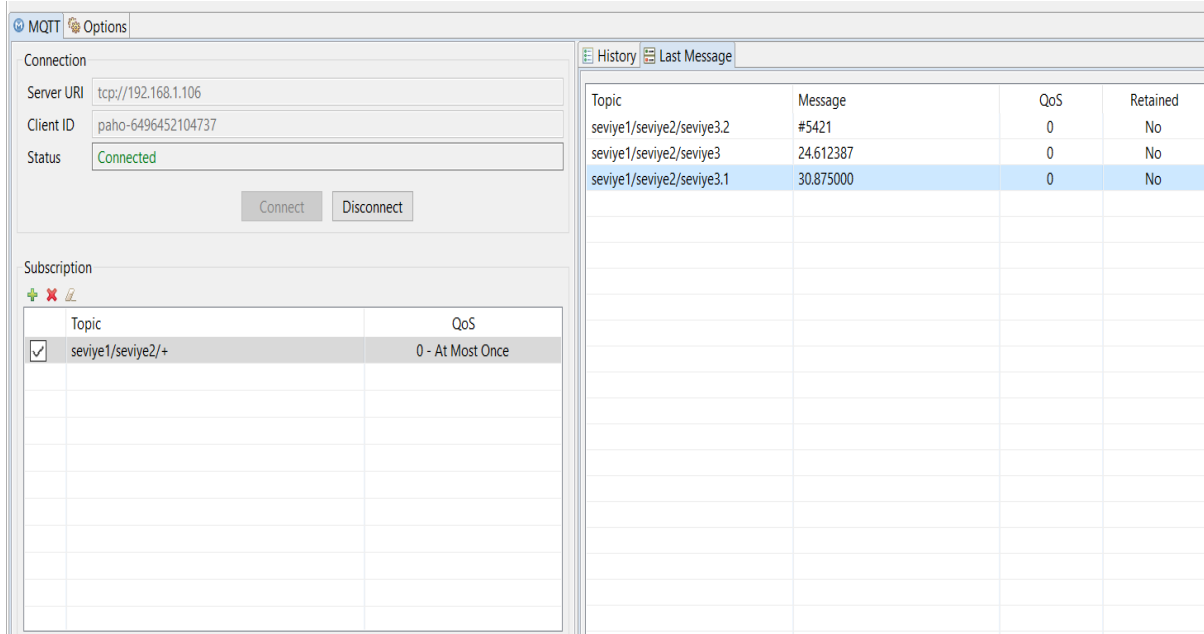
Şekil 6. Deney kiti için konu hiyerarşisi

Şekil 7. 'da görüldüğü gibi yayımcılar bir modem vasıtasıyla arabulucu(broker) ile haberleşmektedir. Arabulucu donanımı olarak Raspberry Pi ZERO kullanılmıştır. Arabulucu yazılımı olarak IoT Eclipse projesi olarak geliştirilen açık kaynaklı MQTT arabulucu Mosquitto [23] kullanılmıştır. Mosquito aracılığı ile yayımcılar ve aboneler aralarından doğrudan veri alışverişi yapabilmektedir. Çalışmada abone istemci olarak Eclipse tarafından geliştirilen Paho [24] (MQTT Client) ile yayımcılar tarafından yayınlanan konulara abone olunmaktadır. Bir yayımcı farklı seviyelerde konularla veri paylaşabilmektedir. Çalışmada 3 farklı seviye için 2 yayımcı kullanılmıştır. Her bir seviyenin aboneliği 2. bölümde anlatıldığı gibi birden fazla konuya aynı ve farklı seviyelere abone olmak için tek seviyeli ve çok seviyeli abonelik yöntemleri uygulanmıştır.



Şekil 7. MQTT yayımcı ve abone ağı

Tek seviyeli ve çok seviyeli abonelik için kullanılan “+” ve “#” karakterleri seviye ayraçları sonuna “/” eklenerek farklı seviyelerde çoklu abonelik oluşturulmuştur. Sadece abonelik konusu “#” olduğunda bütün konulara erişilebilmektedir. Sadece “+” karakteri kullanıldığında “+” seviyesindeki Seviye1, Seviye1.1 ve Seviye1.2 konularına erişilebilmektedir. Aynı şekilde “Seviye1/#” konusuna abone olduğunda seviye1 konuları haricindeki bütün konulara abone olunmaktadır. “Seviye1/#” konusuna abone olduğunda “Seviye2”, “Seviye2.1” ve “Seviye2.2” konularından CC3200 geliştirme kiti üzerinde bulunan BMA22 ivme algılayıcısından konu sıralamasına göre x, y, z ivme verileri yayınlanmaktadır. Şekil 8. 'de gösterildiği gibi “Seviye1/Seviye2/+” konusuna abone olduğunda CC3200 geliştirme kiti tarafından yayınlanan Seviye3, Seviye3.1 ve Seviye3.2 konuları tarafından yayınlanan obje sıcaklığı, algılayıcı sıcaklığı ve değişken bir iç sayaç değeri verilerine erişilebilmektedir.



Şekil 8. Paho istemci (Client) ile yayınlanan bütün konulara erişim.

4. Sonuç

MQTT konu tabanlı haberleşme desteği, konuları hiyerarşik bir şekilde sınıflandırılmasına sağlamaktadır. Hiyerarşik sınıflandırma özelliği birbirinden farklı algılayıcıların amacına uygun olarak alıcılara veya istemcilere hiyerarşik mesajı ulaştırmaktadır. Nesnelere arası iletişimde, çeşitli hiyerarşik sınıfların verilerinin filtre edilmesinde, kablosuz ve endüstriyel algılayıcı ağı teknolojilerinde tercih edilebilir nitelikler taşımaktadır. Özellikle nesnelere interneti uygulamaları için hafif bir protokol oluşu ve protokolün işlem gördüğü cihazlar üzerindeki enerji verimliliği özelliği ile MQTT önemli bir yere sahiptir. MQTT arabulucuları TCP/IP protokolü alt yapısı sayesinde internet yoluyla doğrudan kullanılabilir. Bu erişim yöntemi sayesinde arabulucular vasıtasıyla MQTT ağına bağlanan istemciler tek bir konu veya birden fazla konulara hakkında veri alışverişi yapabilmektedirler. İnternete bağlanan oldukça fazla sayıda istemci yayın yapabilir veya konulara abone olabilir. MQTT 'nin konuları hiyerarşilere ayırıp filtre edebilme yeteneği verilerin erişim karmaşasını ortadan kaldırmaktadır.

Bu çalışmada MQTT protokolü ile seviyelere ayrılmış konuları aynı seviyedeki konulara abone olmayı veya farklı seviyedeki birden çok konuya abone olma uygulanmıştır. CC3200 ve NodeMCU geliştirme kitleri kullanılarak 3 ayrı seviyede tek seviyeli konulara üye olma ve çok seviyeli konulara üye olarak haberleşme örneği gerçekleştirilmiştir. Aynı ve farklı seviyelerdeki konulara üye olma ve konu filtreleme yeteneği, verilerin doğrudan haberleşme sırasında ayrılması sağlamaktadır. Bu yetenek hem verilerin sınıflandırılmasını sağlar hem de MQTT özelliği gereği haberleşme işlemi gerçekleştirilen cihazlarda enerji ve işlem verimliliği sağlamaktadır.

Açıklama

Bu çalışma 11-13 Nisan 2018 tarihleri arasında I. Uluslararası Bilimsel Çalışmalarda Yenilikçi Yaklaşımlar Sempozyumu 'unda özet bildiri olarak sunulmuştur.

5. Kaynaklar

- [1] J. Huang, Y. Meng, X. Gong, Y. Liu, Q. Duan (2014), A Novel Deployment Scheme for Green Internet of Things, *IEEE Internet of Things Journal*, 1(2) ss. 196-205, doi: 10.1109/JIOT.2014.2301819.
- [2] M. A. Salahuddin, A. Al-Fuqaha, M. Guizani, K. Shuaib, F. Sallabi (2017), Softwarization of Internet of Things Infrastructure for Secure and Smart Healthcare, *50(7)*,74-79, doi: 10.1109/MC.2017.195.
- [3] M. Weyrich, C. Ebert (2016), Reference Architectures for the Internet of Things, *IEEE Software*, 33(1), 112-116, doi: 10.1109/MS.2016.20.
- [4] L. Gökrem, M. Bozuklu (2016), Nesnelerin İnterneti: Yapılan Çalışmalar ve Ülkemizdeki Mevcut Durum, *Gaziosmanpaşa Bilimsel Araştırma Dergisi*, 13, 47-68.
- [5] S. A. Çeltik, M. Durgun, L. Gökrem, Y. Durgun (2017), Nesnelerin İnterneti Tabanlı Yangın Alarm Sistemi Tasarımı ve Uygulaması”, *Gaziosmanpaşa Bilimsel Araştırma Dergisi*, 6(3), 66-72.
- [6] A. Al-Fuqaha, A. Khreishah, M. Guizani, A. Rayes, M. Mohammadi (2015), Toward better horizontal integration among IoT services, *IEEE Communications Magazine*, 53(9), 72-79, doi: 10.1109/MCOM.2015.7263375.
- [7] R. Kazala, A. Taneva, M. Petrov, St. Penkov (2015), Wireless Network for Mobile Robot Applications, *IFAC-PapersOnLine*, 48(24), 231-236, doi: 10.1016/j.ifacol.2015.12.088.
- [8] R. N. Rodrigues, A. V. Canez, G. M. Simas, R. Barwaldt (2016), Publish-Subscribe Architecture for Delivering Assistance to Visually Impaired People, *IFAC-PapersOnLine*, 49(30), 150-155, doi: 10.1016/j.ifacol.2016.11.145.
- [9] IBM (2010), *MQTT V3.1 Protocol Specification*. International Business Machines Corporation: IBM.
- [10] Michael Yuan (2017), Getting to know MQTT, IBM, developerWorks, <http://www.ibm.com/developerworks/library/iot-mqtt-why-good-for-iot/index.html>.
- [11] Emq (2017), EMQ- The Massively Scalable Open Source MQTT Broker, <http://emqtt.io>.
- [12] OASIS Message Queuing Telemetry Transport (MQTT) TC | OASIS, https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=mqtt
- [13] ISO/IEC 20922:2016- (MQTT) v3.1.1 (2018), <https://www.iso.org/standard/69466.html>.
- [14] U. Hunkeler, H. L. Truong, A. Stanford-Clark (2014), MQTT-S #x2014; A publish/subscribe protocol for Wireless Sensor Networks, 3. *International Conference on Communication Systems Software and Middleware and Workshops, COMSWARE 2008*, doi: 10.1109/COMSWA.2008.4554519.
- [15] L. Hou (2016), Internet of Things Cloud: Architecture and Implementation, *IEEE Communications Magazine*, 54(12), 32-39, doi: 10.1109/MCOM.2016.1600398CM.

- [16] Z. Shelby, K. Hartke, C. Bormann (2014), *The Constrained Application Protocol (CoAP)*.
- [17] M. Collina, M. Bartolucci, A. Vanelli-Coralli, G. E. Corazza (2014), Internet of Things application layer protocol analysis over error and delay prone links, *2014 7th Advanced Satellite Multimedia Systems Conference and the 13th Signal Processing for Space Communications Workshop (ASMS/SPSC)*, doi: 10.1109/ASMS-SPSC.2014.6934573.
- [18] M. H. Amaran, N. A. M. Noh, M. S. Rohmad, H. Hashim (2015), A Comparison of Lightweight Communication Protocols in Robotic Applications, *Procedia Computer Science*, sub(76), 400-405, doi: 10.1016/j.procs.2015.12.318.
- [19] S. Spinsante (2017), A LoRa enabled building automation architecture based on MQTT, 15, doi: 10.23919/AEIT.2017.8240560.
- [20] Ö. Yerlikaya, G. Dalkılıç (2017), Mesaj Kuyruk Telemetri Taşıma Protokol Güvenliği, program adı: 25. IEEE SINYAL ISLEME ve ILETISIM UYGULAMALARI KURULTAYI, May. 18.
- [21] TI (2018), CC3200 LaunchPad | TI.com, <http://www.ti.com/tool/cc3200-launchxl>.
- [22] NodeMCU (2018), NodeMcu -- An open-source firmware based on ESP8266 wifi-soc., http://www.nodemcu.com/index_en.html.
- [23] Eclipse Mosquitto (2018), *Eclipse Mosquitto*, <https://mosquitto.org/> (erişim May. 11, 2018).
- [24] PAHO (2018), Eclipse Paho- MQTT and MQTT-SN software, <https://www.eclipse.org/paho/>.