



**T.C. İSTANBUL TİCARET
ÜNİVERSİTESİ**

FEN BİLİMLERİ ENSTİTÜSÜ

**NESNELERİN İNTERNETİNDE UYGULAMA KATMANI
ÜZERİNDEKİ HABERLEŞME PROTOKOLLERİNİN İNCELENMESİ
VE DENEYSEL KARŞILAŞTIRILMASI**

Cem GÜLTUNCA

**Danışman
Prof. Dr. Abdül Halim ZAIM**

**YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜH ABD ANABİLİM DALI
İSTANBUL- 2018**

AKADEMİK VE ETİK KURALLARA UYGUNLUK BEYANI

İstanbul Ticaret Üniversitesi, Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmada,

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanılması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- ve bu tezin herhangi bir bölümünü bu üniversitede veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

beyan ederim.

Tarih
06/03/2018
İmza

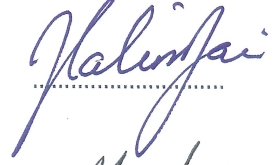
Cem GÜLTUNCA


KABUL VE ONAY SAYFASI

Cem Gültunca tarafından hazırlanan "NESNELERİN İNTERNETİNDE UYGULAMA KATMANI ÜZERİNDEKİ HABERLEŞME PROTOKOLLERİNİN İNCELENMESİ VE DENEYSEL KARŞILAŞTIRILMASI" adlı tez çalışması 06/02/2018 tarihin de aşağıdaki jüri üyeleri önünde başarı ile savunularak, İstanbul Ticaret Üniversitesi Fen Bilimleri Enstitüsü **Bilgisayar Mühendisliği Anabilim Dalı**'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

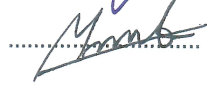
Danışman

Prof. Dr. Abdül Halim ZAİM
İstanbul Ticaret Üniversitesi



Jüri Üyesi

Yrd.Doç.Dr. Metin TURHAN
İstanbul Ticaret Üniversitesi



Jüri Üyesi

Yrd.Doç.Dr.Selim BAYRAKLI
Maltepe Üniversitesi



Onay Tarihi: 16.02.2018

Prof. Dr. Necip ŞİMŞEK
Enstitü Müdürü



İÇİNDEKİLER

Sayfa

ÖZET	III
ABSTRACT	IV
ŞEKİLLER	VI
ÇİZELGELER.....	VII
SİMGELER VE KISALTMALAR	VIII
1. GİRİŞ.....	1
2. LİTERATÜR ÖZETİ	2
2.1. Nesnelerin İnterneti (IoT).....	2
2.2. Geçmiş Çalışmalar.....	4
3. UYGULAMA KATMANI PROTOKOLLERİ.....	6
3.1. AMQP.....	6
3.1.1. AMQP nedir?	6
3.1.2. AMQP nasıl çalışır	6
3.1.3. Mesajlaşma mimarileri	7
3.1.4. Teknik özellikleri.....	11
3.1.5. AMQP mesaj brokerları.....	11
3.1.6. RABBITMQ	11
3.1.7. Operasyonel özellikler.....	12
3.1.8. AMQP teknoloji sağlayıcıları ve kullanıcılar	12
3.1.9. AMQP'nin tarihçesi	12
3.2. MQTT.....	13
3.2.1. MQTT nasıl çalışır	13
3.2.2. MQTT çalışma prensibi	14
3.2.3. MQTT mesaj türleri.....	17
3.2.4. MQTT protokolünün genel özellikleri.....	17
3.3. XMPP.....	18
3.4. COAP.....	19
3.4.1. Mesajlaşma modeli.....	19
3.4.2. İstek/Cevap modeli.....	20
3.4.3. Mesaj formatı	21
4. PROTOKOLLERİN KARŞILAŞTIRILMASI	23
4.1. Protokollerin Özelliklerinin Karşılaştırılması	23
4.2. Protokollerin Uygulama Alanlarına Göre Karşılaştırılması	23
5. DENEYSEL ÇALIŞMALAR	25
5.1. Deneysel Çalışma Ortamı.....	25
5.2. Kullanılan Programlama Dilleri ve Uygulamalar	26
5.3. Kullanılan Mesajlar.....	26
5.4. MQTT Performans Testi.....	27
5.4.1. Test ortamının oluşturulması.....	27
5.4.2. Performans sonuçları	28
5.5. AMQP Performans Testi.....	28
5.5.1. Test ortamının oluşturulması.....	28
5.5.2. Performans sonuçları	29
5.6. COAP Performans Testi.....	30

5.6.1.	Test ortamının oluşturulması.....	30
5.6.2.	Performans sonuçları.....	31
5.7.	XMPP Performans Testi	31
5.7.1.	Test ortamının oluşturulması.....	31
5.7.2.	Performans sonuçları.....	32
5.8.	İşlemci ve Ağ kullanım Oranları.....	33
6.	SONUÇ VE ÖNERİLER	35
6.1.	Araştırma Bulguları	35
6.2.	Gelecek Çalışmalar.....	35
6.3.	Öneriler.....	36
6.3.1.	Farklı bağlantı türleri ile birlikte kullanımı.....	36
6.3.1.1.	WI-FI (kablosuz bağlantı) kullanımı	36
6.3.1.2.	Bluetooth kullanımı.....	36
	KAYNAKLAR.....	38
	ÖZGEÇMİŞ	40



ÖZET

Yüksek Lisans Tezi

NESNELERİN İNTERNETİNDE UYGULAMA KATMANI ÜZERİNDEKİ HABERLEŞME PROTOKOLLERİNİN İNCELENMESİ VE DENEYSEL KARŞILAŞTIRILMASI

Cem GÜLTUNCA

İstanbul Ticaret Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Abdül Halim ZAIM

2018 / 39 sayfa

Günümüzde internet artık her alanda yaygınlaşmış, dünyanın neredeyse her köşesinde insanların yaşamında yer almaya başlamıştır. Şimdi ise artık çok çeşitli cihazların internete bağlanabilir olduğu bir döneme giriş yapmaktayız. Yani nesnelerin interneti olarak adlandırılan bir dönemin başlangıcı olarak düşünebiliriz. Nesnelerin internetini birçok cihazın internete bağlanması, birbirleri ile haberleşmesi ve elde ettiği verileri başka noktalara iletebilmesi olarak tanımlayabiliriz. Buradaki cihazlar telefonlar, tabletler, ev aletleri olabileceği gibi küçük sensörler de olabilir. Özellikle küçük yani kaynakların daha kısıtlı olduğu cihazlar göz önünde bulundurulduğunda nesnelerin internetinde iletişimin nasıl sağlanacağı konusu önem kazanmaktadır. Nesneleri internetinde iletişimde kullanılan dört ayrı protokol bulunmaktadır. Bunlar AMQP, CoAP, MQTT ve XMPP dir. Bu çalışmada ise bu protokoller yapıları ve çalışma mantıkları anlatılmış ve yapılan deneysel çalışmalar ile performans karşılaştırılması yapılmıştır.

Anahtar Kelimeler: AMQP, COAP, Nesnelerin İnterneti, MQTT, XMPP

ABSTRACT

M.Sc. Thesis

EXAMINATION AND COMPARISON OF THE COMMUNICATION PROTOCOLS ON THE APPLICATION LAYER IN IOT

Cem GÜLTUNCA

**İstanbul Commerce University
Graduate School of Applied and Natural Sciences
Department of Computer Engineering**

Supervisor: Prof. Dr. Abdül Halim ZAIM

2018 / 39 Pages

Today the Internet has become ubiquitous, has touched almost every corner of the globe, and is affecting human life in unimaginable ways. We are now entering an era of even more pervasive connectivity where a very wide variety of appliances will be connected to the web. We are entering an era of the “Internet of Things” (abbreviated as IOT). IOT is defined as a paradigm in which objects equipped with sensors, actuators, and processors communicate with each other to serve a meaningful purpose. Several IOT protocols have been introduced in order to provide an efficient communication for resource-constrained applications. However, their performance is not as yet well understood. I evaluated and compared four communication protocols, namely, AMQP, MQTT, XMPP, and COAP. I implemented a some IOT application using open source software for these protocols and measured their performance. In our tests, we compare these protocols.

Keywords: AMQP, COAP, Internet of Things, MQTT, XMPP

TEŐEKKÜR

Bu arařtırma için beni yönlendiren, karşılařtıđım zorlukları bilgi ve tecrübesi ile ařmamda yardımcı olan deđerli Danıřman Hocam Prof. Dr. Abdül Halim ZAIM'e teőekkürlerimi sunarım.

Tezimin her ařamasında beni yalnız bırakmayan aileme sonsuz sevgi ve saygılarımı sunarım.

Cem GÜLTUNCA
İSTANBUL, 2018



ŞEKİLLER

	Sayfa
ŞEKİL 2.1-1 NESNELERİN İNTERNETİ (IOT).....	2
ŞEKİL 2.1-2 AKILLI EVLER	3
ŞEKİL 3-1 AMQP MİMARİSİ	7
ŞEKİL 3-2 DIRECT EXCHANGE	8
ŞEKİL 3-3 FANOUT EXCHANGE	9
ŞEKİL 3-4 TOPIC EXCHANGE	10
ŞEKİL 3-5 HEADER EXCHANGE.....	10
ŞEKİL 3-6 MQTT ABONE OLMA	13
ŞEKİL 3-7 MQTT MESAJ GÖNDERİMİ	14
ŞEKİL 3-8. MQTT ÇOKLU BROKER KULLANIMI	14
ŞEKİL 3-9 MQTT	15
ŞEKİL 3-10 COAP MESAJ FORMATI.....	21
ŞEKİL 5-1 TEST CİHAZLARI	25
ŞEKİL 5-2 MQTT TEST ORTAMI	27
ŞEKİL 5-3 MQTT PERFORMANS SONUÇLARI.....	28
ŞEKİL 5-4 AMQP TEST ORTAMI	29
ŞEKİL 5-5 AMQP PERFORMANS SONUÇLARI.....	29
ŞEKİL 5-6 CoAP TEST ORTAMI	30
ŞEKİL 5-7 CoAP GÖNDERİM PERFORMANSI	31
ŞEKİL 5-8 XMPP TEST ORTAMI	32
ŞEKİL 5-9 XMPP SONUÇLARI.....	32
ŞEKİL 5-10 RASPBERRY Pİ KAYNAK KULLANIMI.....	34

ÇİZELGELER

	Sayfa
ÇİZELGE 3-1AMQP'NİN TARİHÇESİ	12
ÇİZELGE 4-1 PROTOKOL KARŞILAŞTIRMALARI	23



SİMGELER VE KISALTMALAR

AMQP	Advanced Message Queuing Protocol – İleri Mesaj Dizisi Protokolü
API	Application Programming Interface – Uygulama Programlama Arayüzü
COAP	Constrained Application Protocol – Kısıtlı Uygulama Protokolü
CONNECTACK	Connect Acknowledgement
CPU	Central Processing Unit – Merkezi İşlem Birimi
DPWS	Devices Profile for Web Services
DSS	Data Distribution Service
DTLS	Datagram Transport Layer Security
GBİT	Gigabit
HTTP	Hypertext Transfer Protocol
IEC	International Electrotechnical Commission - Uluslararası Elektroteknik Komisyonu
IETF	Internet Engineering Task Force – İnternet Mühendisliği Görev Gücü
IOT	Internet of Things – Nesnelerin İnterneti
IP	Internet Protocol
ISO	International Organization for Standardization - Uluslararası Standartlar Teşkilâtı
MBİT	Megabit
MQTT	Message Queuing Telemetry Transport – Telemetry Mesajlaşma Protokolü
OASIS	Advancing Open Standards for the Information Society
PINGREQ	Ping Request
PINGRESP	Ping Response
PUBACK	Publish Acknowledgement
PUBCOMP	Publish Complete
PUBREC	Publish Received
PUBREL	Publish Release
QOS	Quality of Service – Hizmet Kalitesi Seviyesi
RESTFUL	Representational State Transfer
S2S	Server-to-Server
SASL	Simple Authentication and Security Layer
S	Second
SSL	Secure Sockets Layer - Güvenli Soket Katmanı
SUBACK	Subscribe Acknowledgement
TCP	Transmission Control Protocol
TKL	Token Length
TLS	Transport Layer Security – Taşıma Katmanı Güvenliği
UDP	User Datagram Protocol
UNSUBACK	Unsubscribe Acknowledgement
URI	Uniform Resource Identifier
XMPP	Extensible Messaging and Presence Protocol- Genişletilebilir Mesajlaşma ve Varlık Protokolü

1. GİRİŞ

İnternet'in günümüzde artan önemi ile birlikte, nesnelerin interneti de oldukça artan bir ivme ile gelişmektedir. Akıllı ev sistemleri, akıllı şehir uygulamaları, giyilebilir teknoloji cihazları ve daha birçok internete bağlanabilen cihazlar hayatımızı kolaylaştırmak için ortaya çıkmaya ve gelişmeye devam etmektedir. Nesnelerin interneti üzerine pek çok proje geliştirilmekte ve yeni ürünler piyasaya sunulmaktadır. Yenilikçi teknoloji firmaları da bu akıma ayak uydurarak ve Ar-Ge yatırımları yaparak bu alanda gelişen trendi takip etmek için yüksek bir çaba sarf etmektedir.

Ancak bu cihazların internete ya da birbirlerine bağlanarak sürekli iletişim halinde olması, iletişimin nasıl sağlanacağı, hangi yöntemlerin kullanılacağı gibi soruları da beraberinde getirmektedir. Bu alanda birçok çalışma söz konusudur. Öncelikle burada önemli olan yapılacak olan uygulamanın ihtiyaçlarını iyi çıkarmaktır. Geliştirme sırasında birçok parametreyi göz önünde bulundurmak gerekmektedir. Örnek vermek gerekirse nasıl bir cihaz üretileceği, cihazların kullanılacağı ortam, cihazların enerji ihtiyaçlarının nasıl karşılanacağı bunları yaparken enerji tasarrufunun nasıl sağlanacağı gibi dikkat edilmesi gereken konular vardır. Bunlardan en önemlilerinden biride bu cihazların nasıl iletişim kuracağıdır.

İletişim altyapısını kurarken ilk etapta bağlantının nasıl sağlanacağına karar vermeliyiz. Sonraki aşamada ise önemli konulardan biride bu bağlantı üzerinden iletişim kurulurken hangi protokolü kullanacağımızdır. Burada birden çok uygulama katmanı protokolü mevcuttur. Her protokolün kullanım alanına göre avantajları ve dezavantajları bulunmaktadır. Uygulama alanımıza göre bize en uygun protokolü seçerek ürünümüzü geliştirmeliyiz. Bu araştırmada da bu konuyla ilişkili olarak iletişimde kullanılan MQTT, AMQP, COAP ve XMPP gibi protokoller incelenerek, farklılıkları ve performansları karşılaştırılacaktır. Yapılacak olan deneysel çalışmalar ile birlikte de desteklenecektir.

sistemler tanımlanırken genellikle akıllı kelimesi kullanılmaktadır. Birkaç örnek vermek gerekirse, akıllı saatler, akıllı gözlükler, akıllı şehirler vb.



Şekil 2.1-2 Akıllı Evler

Yakın gelecekte yapılan yatırımlar sayesinde birçok cihaz Wi-Fi veya Bluetooth teknolojisi ile internete bağlanabilir hale gelecektir. Nesnelerin interneti kavramı altında geliştirilen cihazlar genellikle mobil telefon veya tabletlerle birlikte kullanılabilir ve bu mobil cihazlar üzerinde bulunan uygulamalar sayesinde takip edilip yönetimi sağlanabiliyor. Ayrıca geliştirilen mobil bildirim uyarı sistemleri sayesinde kullanıcılar istedikleri kriterlere göre tanımlamalar yaparak, cihazlar üzerinden bildirimler alabiliyorlar.

Teknolojinin hızla ilerlemesi ile birlikte nesnelerin interneti hayatımızın içinde daha çok yer almaya devam edecektir. Ayrıca geliştirilen akıllı ev sistemleri olsun, akıllı şehir sistemleri olsun, bu kavramdaki birçok uygulama veya cihaz hayatımızı kolaylaştırmada çok önemli bir rol oynayacaktır.

2.2. Gemiř alıřmalar

Jordan niversitesinde yapılan bir alıřmada nesnelerin internetinin kritik amacı; nesnelere arasında etkin iletiřim saęlamak ve aralarında farklı trde uygulamalar kullanarak srekli bir baę kurmak olarak tanımlanmıřtır. Ayrıca bu alıřmada XMPP, MQTT, COAP, RESTFUL, DSS, AMQP ve WebSocket karřılařtırılmıřtır. Uygulama katmanı, hizmet sunumundan sorumludur ve uygulama dzeyinde ileti geiři iin bir dizi protokol belirler. Nesnelerin interneti ortamı farklı ve geniř senaryolardan oluřur. Her senaryo farklı ortamı, farklı uygulamalar ve farklı cihazlar ierir ve her cihaz hesaplama kabiliyetinde deęiřiklik gsterir. Ayrıca her uygulamanın amacına istenen Őekilde ulařmak iin farklı gereksinimlere ihtiya duyulur. Bir protokol semek, bir uygulamadır ve evreye baęımlıdır. Dřk gte sınırlı bir iletiřim ortamında, destekli protokol verimli bir Őekilde alıřabilir, ancak pil ile alıřan cihazlarda MQTT protokol iyi bir seim olarak kabul edilir.

Uygulama sıklıkla ihtiya duyulduęunda ve byk miktarda veri gncelleřtirildięinde, yayınlama / abone olma protokol daha sonra iyi bir seim olarak kabul edilir. Uygun uygulama katmanı protokolnn seimi, cihazlar, uygulama ve evre ile ilgili farklı faktrlere baęlıdır. Cihazlara dayalı olarak, hesaplama ve iletiřim yeteneęi nemli bir faktrdr. evrenin dřk g ile ilgili olarak, kısıtlı iletiřim, uygulama katmanı protokoln semek iin kritik bir rol oynamaktadır; bunun yanı sıra, uygulamanın kendisine baęlıdır, farklı uygulamalar farklı ihtiya ve gereksinimler anlamına gelir, bu nedenle farklı uygulamalar farklı protokolleri tercih eder.

Yapılan bařka bir arařtırmada ise MQTT, COAP ve WebSocket zerinde durulmuřtur. Ayrıca iletiřim kurmak iin uygulama katmanı protokollerinin gerekli olduęu kısımlar aıklanarak ortak bir mimari sunulmuřtur. Protokoller arasından COAP'ı UDP zerinden alıřan tek protokol olarak tanımlanmıř ve UDP kullanması sayesinde hafif bir protokol haline gelmiřtir. COAP'ın ardından iletiřim ykn nemli lde azaltan WebSocket gelmektedir. İlgili cihazların hesaplama ve iletiřim yeteneęi de en uygun protokol seerken dikkate alınmalıdır. Kısıtlı iletiřim ve pil tketicimi bir sorun deęilse, RESTful hizmetler kolayca uygulanabilir ve dnya apında HTTP kullanarak

internet ile etkileşim kurabilir. Facebook Messenger tarafından kullanılan MQTT, HTTP kadar yaygın olarak kullanılmaz, ancak pil ile çalıştırılan cihazlar için daha verimli olduğu kanıtlanmıştır. Buna ek olarak, hedeflenen nihai uygulamalar aynı değere göre büyük çapta güncelleme gerektiriyorsa, yayınlama / abone olma protokolleri daha uygundur. Özetlemek gerekirse, en önemlileri cihazların hesaplama ve iletişim yeteneği, pil tüketimi ve nihai uygulamanın akılda tutulması gereken bir uygulama katmanı protokolünün seçimini etkileyen çeşitli faktörlerdir.

Farklı bir araştırmada ise DPWS, COAP ve MQTT protokolleri CPU, memory ve güç kullanımı açısından karşılaştırılmıştır. Yapılan mesaj gönderme testlerinde DPWS 130 milisaniye, COAP 100 milisaniye, MQTT 97 milisaniyede 100 adet mesaj gönderimini sağlamıştır. CPU açısından en iyi değeri DPWS, memory açısından da MQTT olmuştur.

3. UYGULAMA KATMANI PROTOKOLLERİ

3.1. AMQP

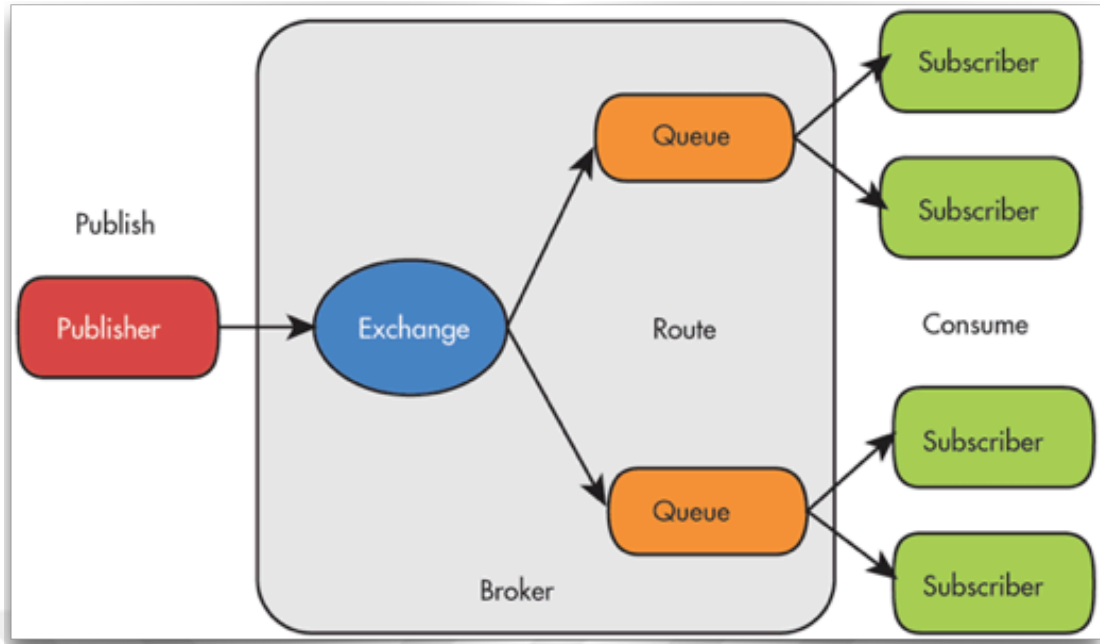
3.1.1. AMQP nedir?

AMQP (Advanced Message Queuing Protocol – İleri Mesaj Dizisi Protokolü), IOT protokolleri arasında anılır. Sunucular arasındaki iş emri mesajlarını iletir. Banka endüstrisiyle ortaya çıkmış mesaj-merkezli özel bir protokoldür ve sıraya konulmuş binlerce iş emrini güvenle yerine getirir. AMQP protokolünün mesaj kaybına çok fazla tahammülü yoktur. Bu sebeple mesaj verilerini kaybetmemek üzerine odaklanmıştır. AMQP protokolü güvenilir noktadan noktaya bir bağlantı sağladığı için TCP üzerinden çalışır. Hatta uç birimler her bir ileti için mesajı aldıklarına dair bir onay bildiriminde bulunmalıdır.

Protokol aynı zamanda çok-fazlı dizi işleme özelliğine sahip opsiyonel bir iş emri modunu da tanımlar. Köklerine bakıldığında AMQP tüm mesajları takip etmeye odaklanmış, her bir mesajın hatadan bağımsız, istenildiği şekilde eriştiğinden emin olmaya çalışan bir protokoldür. AMQP genellikle ticari mesajlaşmada kullanılmaktadır. AMQP'ye göre “cihaz” kavramı back-office veri merkezleri ile haberleşen mobil el-aletleri olarak tanımlanır. IOT tarafında ise AMQP kontrol katmanı veya sunucu tabanlı analiz fonksiyonları için uygundur. Daha çok sunucudan sunucuya (S2S) sistemler için hizmet görür.

3.1.2. AMQP nasıl çalışır

- Yayıncı tarafından üretilen mesaj Exchange'e gönderilir.
- Exchange kurallara(route) göre mesajları kuyruklara iletir.
- Kuyruklar üzerinden de son olarak mesajlar alıcılara iletilir

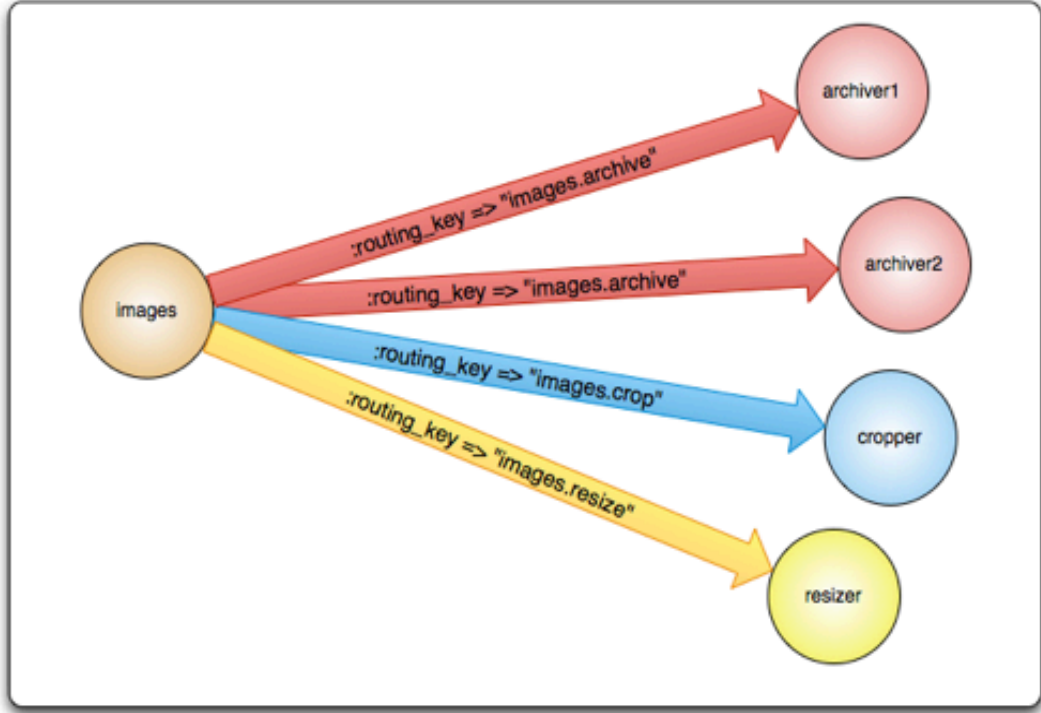


Şekil 3-1 AMQP Mimarisi

3.1.3. Mesajlaşma mimarileri

3.1.3.1. Noktadan noktaya (direct exchange)

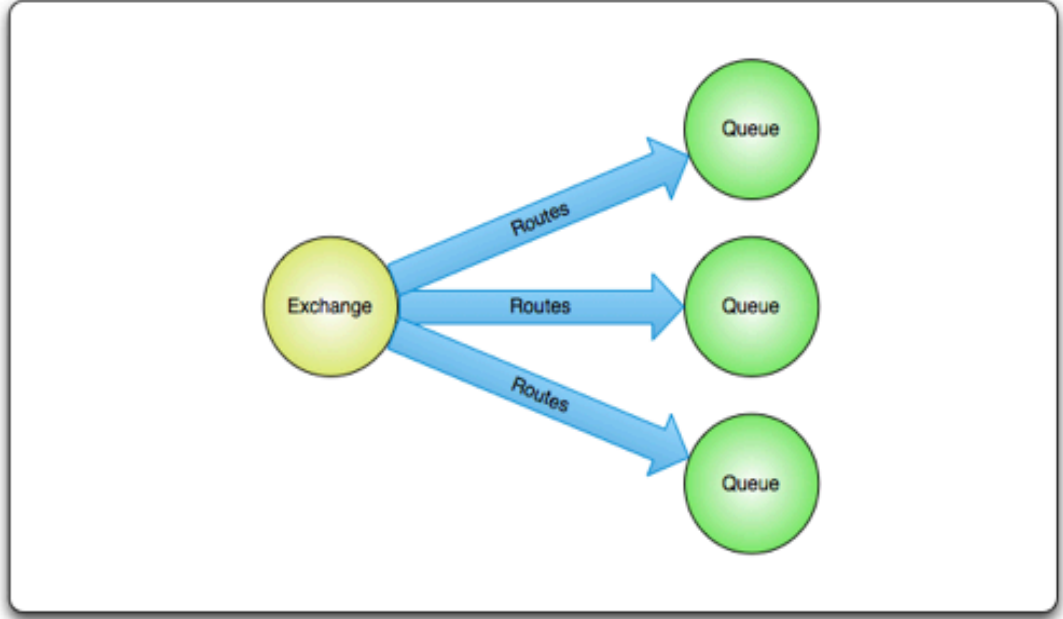
- Bir mesajın üreticisi tarafından (producer, sender) sadece bir tüketiciye (consumer, receiver) iletildiği yapıdır.
- Burada birden fazla tüketicinin olmasında bir sakınca yoktur. Fakat tek bir mesaj sadece tek bir tüketici tarafından işlenir.
- Mesaja exchange'e gönderilirken gideceği kuyruğun key'i eklenir.



Şekil 3-2 Direct Exchange

3.1.3.2. Yayın / abone (fanout exchange)

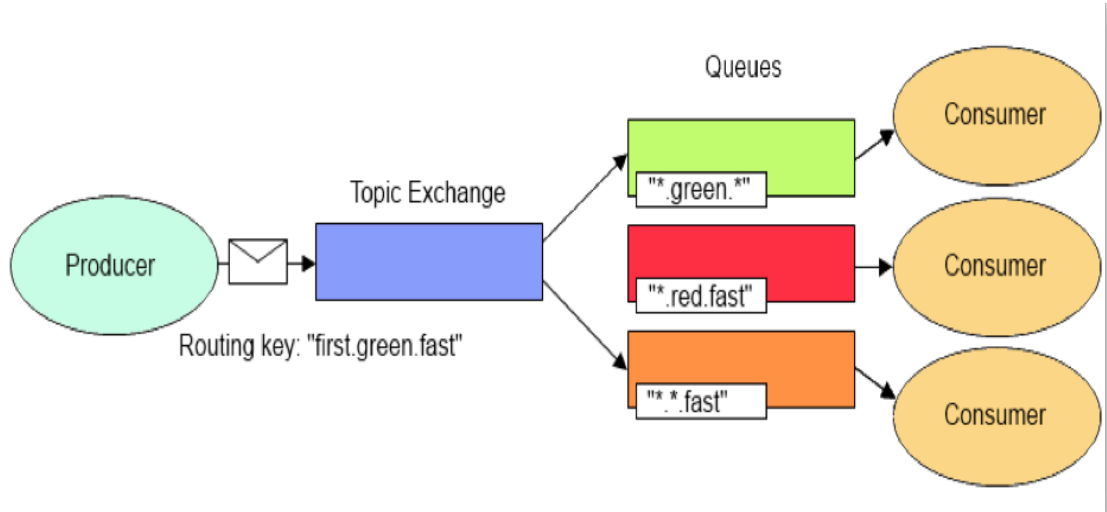
- Bir mesajın üreticisi tarafından tüm tüketicilere iletiildiği yapıdır.
- Bir nevi observer pattern'in başka uygulamalar arasında dağıtılmış hali gibi düşünebiliriz.
- Bir etkinlik oluştuğunda tüm dinleyenler tetikleniyor. Bir exchange'e farklı kuyruklar bağlanıyor. Fanout exchange, gelen mesajı bağlı olan tüm kuyruklara dağıtıyor.
- Burada gönderilen route key göz ardı ediliyor.



Şekil 3-3 Fanout Exchange

3.1.3.3. Topic exchange

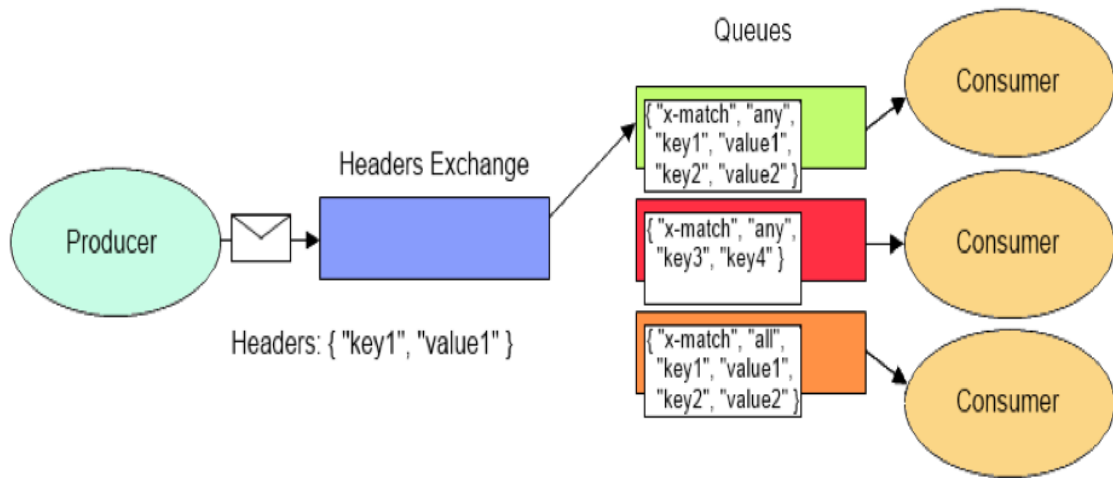
- Bu tip mesajlaşma kullanıldığında mesaj iletimi farklı kurallara göre point-to-point ya da publish-subscribe şeklinde davranabiliyor.
- Bu tip bir yapıda kullanacağımız anahtarımızın belirli özelliklere sahip olması gerekiyor. Yani belirli bir şablonda olması gerekiyor.
- Anahtarımızda arka arkaya gelen kelimeler bulunur ve aralarında ayraç olarak . (nokta) kullanılır. **Örneğin:** kelime1.kelime2.kelime3
- Şablon ayrıca * ve # içerebilir.
- O pozisyonda herhangi bir kelime olmasını, # ise hiçbir kelime olmaması ya da birden çok kelime olmasını temsil ediyor.



Şekil 3-4 Topic Exchange

3.1.3.4. Header exchange

- Kontrol etmek istenilen bilgiler mesajın başlık kısmında tanımlanır.
- Başlıkta olduğu gibi hem noktadan noktaya (point-to-point) hem de yayıncı-abone(publish-subscribe) şeklinde davranabiliyor.
- Eşleşme tipi olarak (x-match argümanı) "Any" ya da "All" seçilebilir.
- "Any" seçildiğinde anahtarlardan herhangi birinin eşleşmesi yeterlidir.
- "All" seçilirse anahtarlardan hepsinin eşleşmesi gerekmektedir.



Şekil 3-5 Header Exchange

3.1.4. Teknik özellikleri

- Verimli: AMQP protokol talimatları ve bunun üzerinden aktarılan iş mesajları için ikili kodlamayı kullanan bir bağlantı yönelimli protokoldür. Bu ağın kullanımını ve bağlı bileşenleri maksimize etmek için gelişmiş akış-kontrol şemaları içerir. Yani protokol verimliliği, esneklik ve birlikte çalışabilirlik arasında bir denge için tasarlanmıştır,
- Güvenilir: Noktadan noktaya bir bağlantı sağlar. TCP üzerinden çalışır.
- Esnek: AMQP farklı topolojileri desteklemek için kullanılan esnek bir protokoldür. Hem client-to-client, client-to-broker, hem de broker-to-broker iletişimi için kullanılabilir
- Broker modeli bağımsız: AMQP spesifikasyonu bir komisyoncu tarafından kullanılan mesajlaşma modeli üzerinde herhangi bir gereksinime sahip değildir. Bu kolayca mevcut mesajlaşma brokerına AMQP desteği eklemenin mümkün olduğu anlamına gelir.
- Standartlaşmış: AMQP 1.0 uluslararası bir standart olan ISO ve ISO/IEC 19464:2014 olarak IEC tarafından onaylanmıştır.

3.1.5. AMQP mesaj brokerları

- RabbitMQ
- Windows Azure Service Bus
- Apache Qpid
- Apache ActiveMQ
- SwiftMQ
- StormMQ
- WSO2 Message Broker
- OpenAMQ

3.1.6. RABBITMQ

- RabbitMQ mesaj broker uygulamasıdır
- Açık kaynaklı ve Erlang dilinde geliştirilmiştir.
- AMQP versiyon desteği: 0.8, 0.9 ve 0.9.1

3.1.7. Operasyonel özellikler

- Yüksek Kullanılabilirlik (High Availability)
- Kümeleme (Clustering)
- Yönetim: HTTP API için yönetim eklentisi, Web Ara yüz, Komut Sistemi Araçları
- Güvenlik: Sanal Sunucular, Kimlik Doğrulama & İzinler

3.1.8. AMQP teknoloji sağlayıcıları ve kullananlar

Teknoloji Sağlayıcıları: Axway Software, Huawei Technologies, IIT Software, INETCO Systems, Kaazing, Microsoft, Mitre Corporation, Primeton Technologies, Progress Software, Red Hat, SITA, Software AG, Solace Systems, VMware, WSO2, Fenika

Kullanan Firmalar: Bank of America, Credit Suisse, Deutsche Boerse, Goldman Sachs, JPMorgan Chase

3.1.9. AMQP'nin tarihçesi

Çizelge 3-1AMQP'nin Tarihçesi

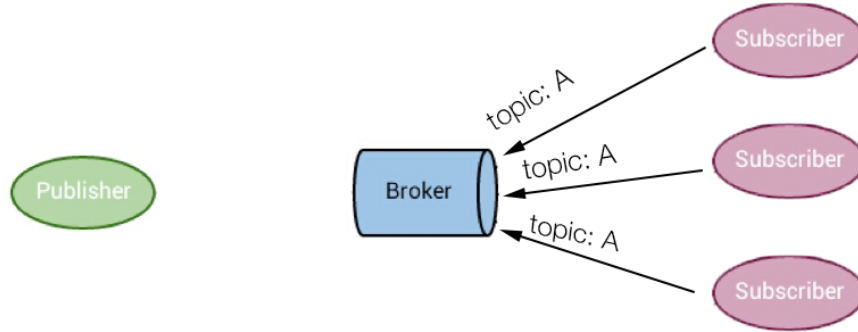
Tarih	
2003	AMQP JP Morgan da geliştirilmeye başlandı.
2005	AMQP çalışma grubu kuruldu
2006	Haziran AMQP 0.8 / Aralık AMQP 0.9 yayınlandı
2006	Rabbit teknoloji RabbitMQ ilk sürümünü yayınladı
2011	AMQP 1.0 yayınlandı
Ekim 2012	AMQP 1.0 OASIS standartlarında kabul edildi.

3.2. MQTT

MQTT (Message Queuing Telemetry Transport – Telemetry Mesajlaşma Protokolü), abone olma ve yayınlama mantığına dayanan telemetri mesajlaşma protokolüdür. Nesnelerin internetinde cihazlar arasındaki haberleşmenin sağlanmasında yaygın olarak kullanılmaktadır. Diğer benzer haberleşme protokollerinden MQTT’yi ayıran en önemli özelliği ise hafif olması ve bu nedenle birçok platformda rahatlıkla kullanılabilmesidir. MQTT IBM de çalışan Dr Andy Stanford-Clark ve Arcom da çalışan Arlen Nipper tarafından 1999 yılında oluşturulmuştur. Uygun maliyetli, görüntülenecek aygıtların bağlantısı kolay olan bu protokol, ilk başlarda uzaktan erişim cihazlarıyla yağ ve gaz endüstrisinde kullanılmak üzere yapılmıştır.

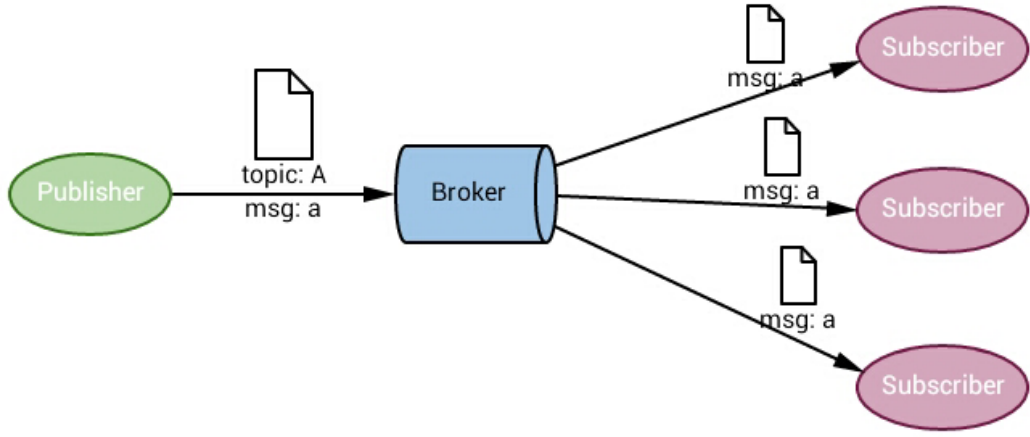
3.2.1. MQTT nasıl çalışır

Abone birimleri mesaj almak için abone olan birimlerdir ve hangi birimlerden hangi mesajları alacaklarını bulutta bulunan broker birimine “konu” olarak bildirirler.



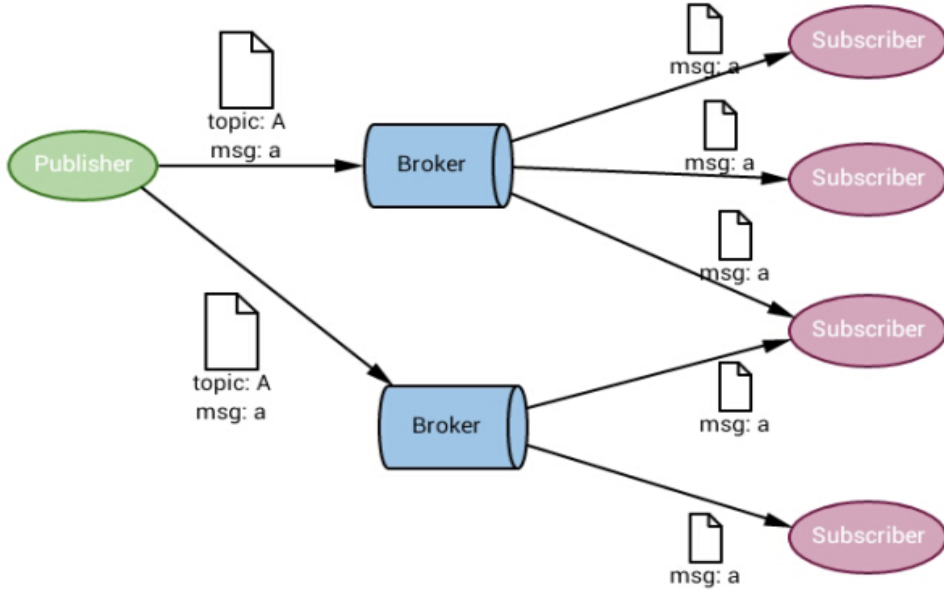
Şekil 3-6 MQTT Abone Olma

Yayıncı birimleri ise mesaj yayınlayan birimlerdir ve yayınlamak istedikleri mesajı, konu bilgisi ile birlikte broker birimine iletirler. Broker, ilgili konuya üye olan abonelere ilgili mesaj iletir.



Şekil 3-7 MQTT Mesaj Gönderimi

Yayıncı ve abone birimleri gerekirse birden fazla broker birimi üzerinden diğer birimler ile iletişim kurabilirler.



Şekil 3-8. MQTT Çoklu Broker Kullanımı

3.2.2. MQTT çalışma prensibi

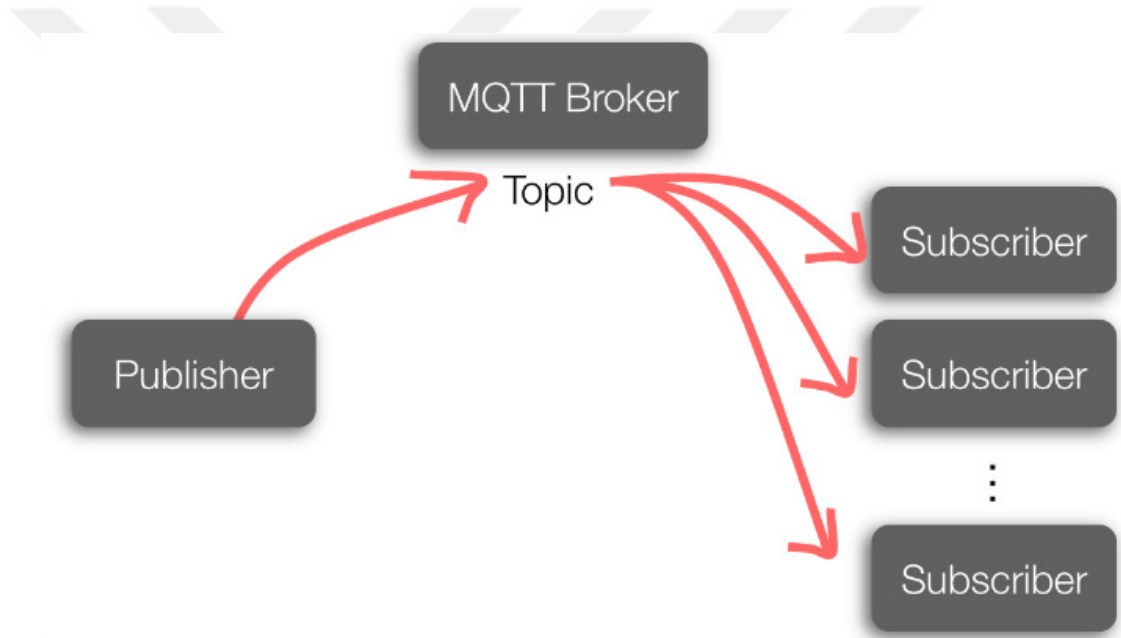
MQTT çalışma yapısı dört farklı bölümden oluşur

- Bağlantı,
- Kimlik Doğrulama,

- İletim,
- Sonlandırmadır.

Müşteri, öncelikli olarak broker birimine TCP/IP bağlantısı kurar. Bunu kurarken, broker'in işlemcisi tarafından belirlenmiş standart port ya da özel bir port numarası kullanır.

Standart port numaraları; şifresiz bağlantılar için 1883, şifreli bağlantılar için 8883'tür. Bu portlar SSL/TLS kullanırlar. SSL/TLS uyuştuğu sırada, müşteri birim sunucu sertifikasını doğrular ve kimlik doğrulama işlemi gerçekleşmiş olur. Müşteri birim ayrıca broker birimine müşteri sertifikasını da açabilir.



Şekil 3-9 MQTT

MQTT protokolü genel olarak kaynak açısından daha kısıtlı olan cihazlara yönelik bir protokol olarak kalmayı hedeflediği için, SSL/TLS her zaman bir seçenek olmayabilir. Hatta bazı durumlar içinde istenmeyebilir. Bu gibi durumlarda, kimlik doğrulama cihaz üzerinden sunucuya gelen herhangi bir metot ile şifrelenmemiş açık bir metin şeklinde kullanıcı adı ve parola ile yapılır. Bu bağlantı/bağlantı geri bildirim paket sıralamasının bir parçasıdır. Bazı broker birimleri, özellikle internette yayınlanan broker birimleri, kimliği belirli olmayan müşteri cihazlarını kabul ederler. Böyle durumlarda kullanıcı adı ve şifresi bölümü boş olarak bırakılır.

MQTT hafif protokol olarak anılır. Çünkü tüm mesajlar küçük kodlara sahiptir. Her mesaj sabit ve 2 byte büyüklüğünde bir başlığa sahiptir. Ek bir seçenek olarak değişken bir başlığa da sahip olabilirler. Mesajın taşıma kapasitesi bilgi ve “Quality of Service” seviyesi olarak 256 MB’a sınırlandırılmıştır. Hizmet Kalitesi, kısaca QoS, ağ iletişimi hizmet kapasitesi olarak bilinmektedir. Ağ üzerindeki uygulamaları önceliklendirerek zaman kaybını azaltmayı hedefleyen bir ağ servsidir. Bir ağ bağlantısı üzerinden çalışan bir trafik veya program türüne öncelik veren çeşitli tekniklere karşılık gelir. 3 farklı hizmet kalitesi seviyesi vardır. Bu seviyeler MQTT protokolünün içerikleri nasıl yöneteceğini belirler. Yüksek seviyeli hizmet kalitesi daha güvenilir olsa da daha fazla gecikme yaşarlar ve bant genişliği ihtiyaçları vardır. Bu nedenle abone birimler kendileri ne almak istiyorlarsa ona göre yüksek hizmet kalitesi seviyelerini belirleyebilir.

Hizmet kalitesi seviyelerinden ilki “Unacknowledged Service” yani onaylanmamış servistir. Bu hizmet kalitesi seviyesi yayınlama (publish) paket sıralamasını kullanır. Yayıncı, broker birimine bir kere mesajı yollar ve broker birimi de bu mesajı bir kez abone birime yollar. Mesajın ulaşmış ulaşmadığından emin olunacak bir mekanizma yoktur ve mesaj kaydedilmez. Bu seviye “QoS0” olarak da adlandırılır. Kısaca özetlenirse bu hizmet kalitesi seviyesinde mesaj iletmeyebilir ancak minimum ağ trafiği kullanımı vardır.

Hizmet kalitesi seviyelerinden ikincisi ise “Acknowledged Service” yani onaylanmış servistir. Bu hizmet kalitesi seviyesi yayınlama/onay (PUBLISH/PUBACK) paket sıralamasını kullanır. Onaylı bir paket, içeriğin alındığını ve eğer alınmadıysa belirli zaman aralıklarında içeriğin tekrar yollandığını doğrular. Bu durumda abone birim aynı mesajdan birden çok kopya alabilir. Bu seviye “QoS1” olarak da adlandırılır. Özetlemek gerekirse bu seviyede mesaj kesin iletilir fakat mesaj birden fazla kez iletilebilir.

Hizmet kalitesi seviyelerinden üçüncüsü “Assured Service” yani garanti servis denilebilir. Bu hizmet kalitesi seviyesi mesajı iki çift paket halinde iletir. İlk çift PUBLISH/PUBREC olarak adlandırılırken ikinci mesaj paketi çifti PUBREL/PUBCOMP olarak adlandırılır. Bu servis seviyesinde mesaj kesin iletilir.

Tek seferde iletilen bu hizmet kalitesi seviyesinde maksimum ağ trafiği kullanımı vardır.

3.2.3. MQTT mesaj türleri

- CONNECT: Sunucuya bağlantı kurmak için gönderilen mesaj türüdür.
- CONNECTACK: Bağlantı mesajına karşılık gönderilen onay mesajıdır.
- PUBLISH: Mesajı yayınlama gönderilen mesajlardır.
- PUBACK: Yayınlama mesajına karşılık gönderilen onay mesajıdır.
- PUBREC: Yayınlama mesajlarının alındığına dair kullanılan mesaj türüdür.
- PUBREL: Yayınlama mesajlarının yayınlandığına dair gönderilen mesaj türüdür.
- PUBCOMP: Yayınlama işleminin tamamlandığına dair gönderilen mesajlardır.
- SUBSCRIBE: Abonelik için gönderilen mesaj türüdür.
- SUBACK: Abonelik mesajının onayıdır.
- UNSUBSCRIBE: Abonelikten çıkmak için gönderilen mesaj türüdür.
- UNSUBACK: Abonelik iptal mesajının onayıdır.
- PINGREQ: Bağlantı kontrolü için gönderilen bir mesajdır.
- PINGRESP: Bağlantı kontrolü mesajına karşılık cevap olarak gönderilen bir mesajdır.
- DISCONNECT: Bağlantının koptuğuna dair gönderilen bir mesaj türüdür.

3.2.4. MQTT protokolünün genel özellikleri

- Asenkron bir protokoldür.
- Broker temelli haberleşme mekanizmasına sahiptir.
- Konuya dayalı adresleme vardır.
- Güvenlik olarak SSL/TLS destekler.
- En az kaynak kullanımını hedefler.
- Milisaniyeler seviyesinde bir haberleşme sunar.

- TCP/IP bağlantı türünü kullanır.
- Varsayılan port 1883'tür.
- TCP/IP protokolünün yazılabildiği Linux, Windows, Android, iOS, MacOS işletim sistemlerinde çalışır.

3.3. XMPP

XMPP (Extensible Messaging and Presence Protocol- Genişletilebilir Mesajlaşma ve Varlık Protokolü) önceki adıyla Jabber, Internet'teki iki ucun herhangi bir yapısal bilgiyi birbirleri arasında karşılıklı ve neredeyse eş zamanlı aktarmalarına olanak sağlayan XML protokol ve teknolojilerinden oluşan açık bir yapıdır. XMPP esnek bir yapıya sahip olduğundan geliştirilebilir bir teknoloji olarak karşımıza çıkmaktadır. Esnek olduğu için dosya gönderimi, oyun oynama veya herhangi yeni fikri uygulama imkânı sağlamaktadır. XMPP aynı zamanda birçok popüler anlık mesajlaşma uygulamasının da mesajlaşma altyapısında kullanılan protokol olmuştur. Bu uygulamalar arasında WhatsApp, Facebook Messenger ve Google GTalk uygulamalarını sayabiliriz.

- Açıktır: XMPP protokolleri açık ve özgürdür, kolayca anlaşılabilir; istemciler, sunucular, bileşenler ve kod kütüphaneleri için birden fazla uygulama mevcuttur.
- Standarttır: IETF çekirdek XML sürekli yayın protokollerini kabul edilen mesajlaşma teknolojisi olacak şekilde standartlaştırmıştır. XMPP'da kullanılmakta ve IETF'nin standartlaşma sürecine uygun olarak XMPP standartlar kuruluşu tarafından geliştirilmeye devam etmektedir.
- Güvenilirdir: XMPP teknolojisi ilk olarak 1998'de geliştirilmeye başlanmıştır ve şu anda yüksek oranda kararlı bir yapı haline gelmiştir. Şu anda XMPP teknolojileri yüzlerce geliştirici tarafından geliştirilmekte, on binlerce sunucusu Internet üzerinde çalıştırılmakta ve milyonlarca insan mesajlaşma için kullanmaktadır.
- Merkezi değildir: XMPP ağının mimarisi e-posta mimarisine çok benzer, herhangi bir kişi bir XMPP sunucu işletebilir.

- Güvenlidir: Herhangi bir XMPP sunucu, XMPP ağından tamamen izole edilebilir. Örneğin sadece yerel bir ağ içerisinde kullanılabilir ve çekirdek XMPP spesifikasyonlarında yer verilen SASL ve TLS ile güçlü ve sağlam bir güvenlik sunar.
- Genişletilebilirdir: Herhangi birisi XML'den faydalanarak çekirdek protokol üzerine kendi istediği fonksiyonları inşa edebilir.
- Esnektir: XMPP uygulamaları XMPP'ın sunduğu altyapı sayesinde mesajlaşmanın ötesinde, ağ yönetmek, dosya ve içerik paylaşmak, uzaktaki sistemlerin durumunu görüntülemek, oyun oynamak için de kullanılabilir.
- Kozmopolittir: Çok sayıda şirket ve açık kaynak kodlu proje XMPP protokolünü kullanmaktadır, gerçek zamanlı uygulamalar ve servisler geliştirmektedir.

3.4. COAP

COAP (Constrained Application Protocol – Kısıtlı Uygulama Protokolü), IETF tarafından tasarlanmış bir uygulama katmanı protokolüdür. Adından da anlaşılacağı gibi birincil amacı kısıtlı kaynaklara sahip cihazlar üzerinde ve kısıtlı bant genişliğine sahip ağlarda çalışmak olan COAP, tasarımı basit tutmak için UDP üzerinde çalışır. COAP, uygulama uç birimleri arasında etkileşimli bir istek/cevap modeli sunar; servisler ve kaynakların keşfi için yerleşik desteğe sahiptir; URI gibi anahtar web kavramlarını barındırır.

COAP'ın etkileşim modeli HTTP'nin istemci/sunucu modeline benzer ancak CoAP makineden-makineye çalıştığı için COAP hem istemci hem sunucu rollerini üstlenir. HTTP'den farklı olarak bu etkileşim UDP üzerinde asenkron olarak gerçekleştirilir. Bu süreç, opsiyonel olarak güvenilirliği desteklemek için mesaj katmanlarının mantıksal kullanımı ile gerçekleştirilebilir.

3.4.1. Mesajlaşma modeli

COAP'ta dört farklı mesajlaşma tipi vardır: Onay gerektiren (confirmable), onay gerektirmeyen (Non-Confirmable), onay (Acknowledgment) ve reset.

Onay gerektiren mesajların karşılığında bir onay mesajının gönderilmesi gerekmektedir. Bu onay mesajları cevapla birlikte gönderilebileceği gibi ayrı olarak da gönderilebilir.

Onay gerektirmeyen mesajların karşılığında onay mesajı gönderilmesine gerek yoktur.

Onay mesajları mesajlar onay gerektiren mesajlara karşılık olarak gönderilirler.

Reset mesajları, bir nedenden dolayı yerine getirilemeyen istek karşılığında gönderilir. Ayrıca boş bir reset mesajı bir cihazın ulaşılabilir olup olmadığını anlamak için de kullanılabilir.

COAP, 4 byte'lık sabit bir başlık kullanır. Her mesajda 16 bitlik mesaj kimliğini içeren bir bilgi bulunur. Bu da saniyede 250 tane mesaja kimlik bilgisi eklemek için yeterlidir.

Güvenilirlik, onay gerektiren mesajlar ile sağlanır. Bir onay gerektiren mesaj, bu mesaj için bir onay mesajı alınana kadar bir zaman aşımı mekanizması kullanılarak yeniden gönderilir. Bir cihaz bir isteğin içeriğini yerine getiremeyeceği zaman onay mesajı yerine reset mesajı gönderir. Güvenilir olması gerekmeyen mesajlar ise onay gerektirmeyen mesaj türünde gönderilirler. Bu mesajların taşıdığı isteğin yerine getirilememesi halinde ise aynı şekilde reset mesajı gönderilebilir.

3.4.2. İstek/Cevap modeli

Metot kodu veya cevap kodu içeren COAP istek/cevap semantiği COAP mesajları üzerinde taşınır. Bir istek onay gerektiren veya onay gerektirmeyen mesaj üzerinde taşınabilir; eğer bir onay gerektiren mesaja verilebilecek cevap hazırsa, oluşturulan onay mesajı ile birlikte gönderilebilir. Bu cevap verme yöntemine “piggybacked” denmektedir. Bu cevap tipinde ayrı onay ve ayrı cevap yoktur. İki mesaj birleştirilmiş olarak gönderilmektedir.

Eğer sunucu, isteği anında cevaplayacak durumda değilse basitçe boş bir onay üretir ve karşı tarafa gönderir. Cevap hazırlandığı zaman yeni bir onay gerektiren mesaj oluşturulur ve cevap bu mesaj vasıtası ile gönderilir. Bu cevap verme türüne “ayrılmış

cevap” denir. Bu durumda istemci, kendisine onay gerektiren mesaj içerisinde gelen cevap karşılığında onay mesajı göndermek zorundadır. Aksi taktirde sunucu sürekli olarak aynı cevabı gönderecektir. COAP, cevapları isteklerle eşleştirmek için bir jeton (token) kullanır. Jeton kavramı mesaj kimliğinden farklı bir kavramdır.

Eğer bir istek onay gerektirmeyen olarak gönderilmişse, cevap onay gerektirmeyen veya onay gerektiren mesaj olarak gönderilebilir.

COAP, GET, PUT, POST, DELETE gibi temel http komutlarını http’dekine benzer şekilde kullanır. Ayrıca unicast iletişimin yanı sıra multicast iletişime de olanak tanır.

3.4.3. Mesaj formatı

0				1				2				3																			
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Ver		T		TKL				Code				Message ID																			
TOKEN																															
OPTIONS																															
1	1	1	1	1	1	1	1	1	1	PAYLOAD																					

Şekil 3-10 COAP Mesaj Formatı

- Versiyon (Ver): iki bit işaretli tamsayıdır. CoAP’ın versiyon numarasını gösterir.
- Tip (T): iki bit işaretli tamsayıdır. Mesaj tipini gösterir. Onay Gerektiren (0), onay gerektirmeyen (1), onay (2) ve reset (3).
- Jeton Uzunluğu (Token Length – TKL): 4 bitlik işaretli tamsayıdır. Değişken uzunluklu jeton alanının uzunluğunu gösterir.
- Kod (Code): 8 bitlik işaretli tamsayı. Bu alan bir isteği, başarılı bir cevap bilgisini, bir istemci hata cevabını veya bir sunucu hata cevabını gösterebilir.
- Mesaj Kimliği (Message ID): 16 bitlik işaretli tamsayı. Mesaj tekrarlarını tespit etmek için kullanılır. Ayrıca, mesaj tiplerini eşleştirmek için kullanılır.

- Bařlıktan sonra jeton(TOKEN) deęeri gelmektedir. Jeton deęeri istekleri ve cevapları iliřkilendirmek iin kullanılır.
- Bařlık ve Jeton alanlarından sonra seenekler(OPTIONS) alanı gelmektedir. CoAp mesajı hibir opsiyon barındırmayacaęı gibi birden fazla opsiyon da barındırabilir. Seenek alanını yk (PAYLOAD) alanından yk iřaretisi adı verilen 1 byte'lık alan ayırır. Yk alanları iletilen verilerde gerek istenilen mesajın parasıdır.
- Son olarak opsiyonel yk alanı gelmektedir. Eęer bir yk alanı varsa, bu yk alanından nce 8 tane bitten oluřan yk iřaretisi gelir. Yk alanı, yk iřaretisinden sonra UDP data gramının sonuna kadar uzatılabilir.

4. PROTOKOLLERİN KARŞILAŞTIRILMASI

4.1. Protokollerin Özelliklerinin Karşılaştırılması

Çizelge 4-1 Protokol Karşılaştırmaları

	MQTT	AMQP	XMPP	COAP
Tanım	Yayınlama / Abonelik	Noktadan-Noktaya ya da Yayınlama / Abonelik	Noktadan-Noktaya ya da Yayınlama / Abonelik	İstek/Cevap
Uygulama Mimarisi	Broker	Broker	XMPP Server (broker)	Client-Server
Hizmet Kalitesi	3	3	Yok	Onaylı veya Onaysız mesaj seçeneği
Birlikte Çalışabilirlik	Kısmi	Var	Var	Var
Gerçek Zamanlı	Yok	Yok	Yok	Yok
İletişim Protokolü	TCP	TCP	TCP	UDP
Üyelik Kontrolü	Hiyerarşik eşleşme ile konular	Santral, kuyruk ve mesaj filtreleme	Konuya bağlı düğümler	Çoklu Adresleme desteği
Veri Serileştirme	Tanımlanmamış	Sistem veya kullanıcı tanımlı	XML	Konfigüre edilebilir
Standartlar	OASIS MQTT Standartları	OASIS AMQP	XMPP Standartlar Kurumu	IETF COAP Standartları
Kodlama	İkili	İkili	Düz Metin	İkili
Lisanslama Modeli	Açık Kaynak & Kurumsal Lisans	Açık Kaynak & Kurumsal Lisans	Açık Kaynak & Kurumsal Lisans	Açık Kaynak & Kurumsal Lisans
Dinamik Keşif	Yok	Yok	Var	Var
Mobil Cihaz Desteği (Android, iOS)	Var	Var	Var	HTTP vekil üzerinden
6LoWPAN cihaz desteği	Var	Uygulamaya özgü	Yok	Var
Çok Fazlı İşlemler	Var	Var	Yok	Yok
Güvenlik / Şifreleme Yöntemleri	Basit- Kullanıcı Adı/Şifre modeli, Veri Şifreleme için SSL desteği	SASL kimlik doğrulama, Veri şifreleme için TLS desteği	SASL kimlik doğrulama, Veri şifreleme için TLS desteği	DTLS

4.2. Protokollerin Uygulama Alanlarına Göre Karşılaştırılması

Nesnelerin internetindeki bu protokollerin uygulama alanlarına göre avantaj ve dezavantajları vardır. İletişimin kurulacağı iki noktayı düşündüğümüzde bu bağlantı

sunucular arasında yapılıyorsa AMQP burada en avantajlı protokoldür. Cihazlar ve sunucular arasında bir veri iletişimi var ise diğer protokoller tercih edilebilir. Ayrıca gönderilen mesajların eksiksiz iletişi konusunda da AMQP diğer protokollere göre bir adım öndedir.

Kullandığımız cihazlar için enerji kullanımının düşük olmasının önemli olduğu durumlarda ise MQTT öne çıkmaktadır. Özellikle hafif bir yapıya sahip olması işlem gücünün düşük olduğu ve pil ile çalışan bir uygulama ortamında avantaj sağlamaktadır. Ayrıca eğer bizim için iletilen mesajların tümünün anlık olarak iletilmesi önemli değilse hizmet seviyesi en düşük seviye olarak ayarlanarak aynı zamanda ağ kullanımından da tasarruf sağlanabilmektedir. Aynı zamanda COAP protokollüde bu tarz bir kullanıma uygun olarak kısıtlı kaynaklarda çalışabilecek ve düşük ağ kullanımı yaratacak şekilde tasarlanmıştır.

Örnek vermek gerekirse; bir veri merkezinde sensörler ile sıcaklık ölçtüğümüzü ve sıcaklık değerine göre soğutma sistemlerini çalıştırdığımızı düşünelim. Sıcaklık ölçümlerini 10 saniyede bir kontrol sunucumuza iletirken arada birkaç mesajın iletilmemesi bizim için sorun teşkil etmeyebilir. Bu kısımda hafif özelliklere sahip protokolleri kullanabiliriz. Ancak kontrol sisteminden soğutucuyu çalıştıracak mesajın iletilmemesi bizim için kabul edilebilir bir durum olmayacaktır. Bu yüzden kullanacağımız protokolleri seçerken genel olarak tüm sistemi düşünerek hareket etmeliyiz.

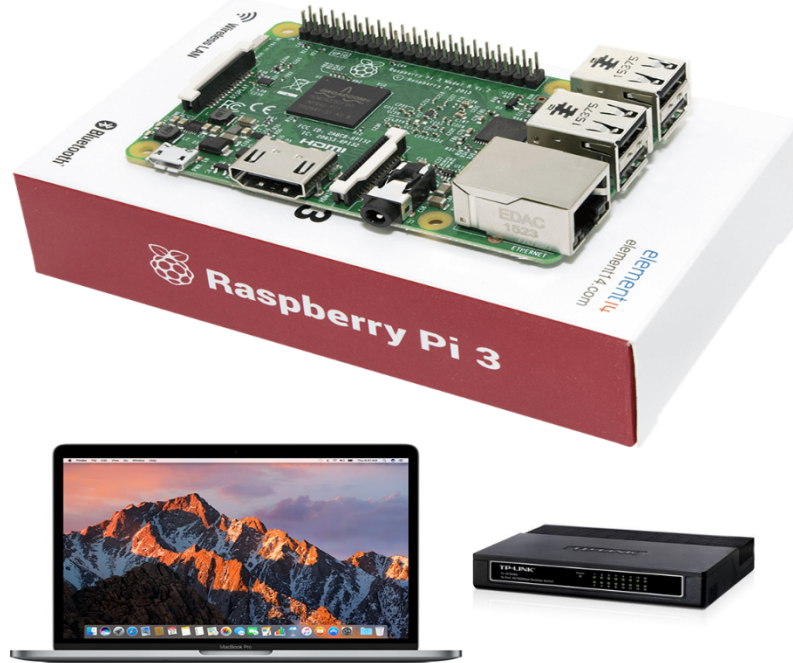
XMPP protokolüne gelecek olursak daha çok kullanım alanı olarak mesajlaşma uygulamaları gelmektedir. Ayrıca esnek bir yapıya sahip olması kendi uygulamalarımızı geliştirirken bize büyük bir avantaj sağlamaktadır. Sunucu yazılımında gönderilen mesajı iletilmemesi durumunda tutmasında çevrimdışı mesaj bir mesajlaşma altyapısı da sunmaktadır. Bu nedenle XMPP mail sistemine de benzetilmektedir.

5. DENEYSEL ÇALIŞMALAR

Bu arařtırmada nesnelere internetinde uygulama katmanında kullanılan protokoller arasındaki karřılařtırmayı grebilmek iin bazı testler yapılmıřtır. Yapılan bu testler sırasında ıkan sonular ayrı ayrı kaydedilerek karřılařtırılma yapılması saėlanmıřtır.

5.1. Deneysel alıřma Ortamı

Bu alıřmada 2 adet Raspberry pi 3 modeli cihaz kullanılmıřtır. Cihazlar arasındaki baėlantıyı saėlamak iin 1 adet TP-Link 1 Gbit bant geliřtiėine sahip switch ile kategori 5 network kabloları kullanılarak cihazlar arasında lokal bir network baėlantısı oluřturulmuřtur. Baėlantının hızı Raspberry pi cihazları destekleyebildiėi maksimum hız olan 100 mbit/sn dır. Raspberry pi cihazlar Raspbian jessie iřletim sistemi ile alıřmaktadır. Ayrıca programlama, sunucu ve kullanıcı tarafı testler yapılabilmesi iin python, javascipt ve C# dilleri kullanılarak kk test uygulamaları yapılmıřtır. Ayrıca testlerde MacOSX iřletim sistemine, i7 iřlemci, 16 GB memory ve ssd disk'e sahip MacbookPro bilgisayar kullanılmıřtır.



řekil 5-1 Test Cihazları

Yazılan test programları ile dört protokol üzerinden de mesaj gönderim denemeleri yapılmıştır. Bu denemeler sırasında 5 farklı boyutta mesaj kullanılmıştır ve her boyuttaki mesaj seri olarak bir döngü içerisinde 1000 kere gönderilerek toplam gönderim süresi ölçümlenmiştir.

5.2. Kullanılan Programlama Dilleri ve Uygulamalar

- Python
- Javascript
- C#, .Net
- NodeJS
- RabbitMQ
- Mosquitto
- eJabber

5.3. Kullanılan Mesajlar

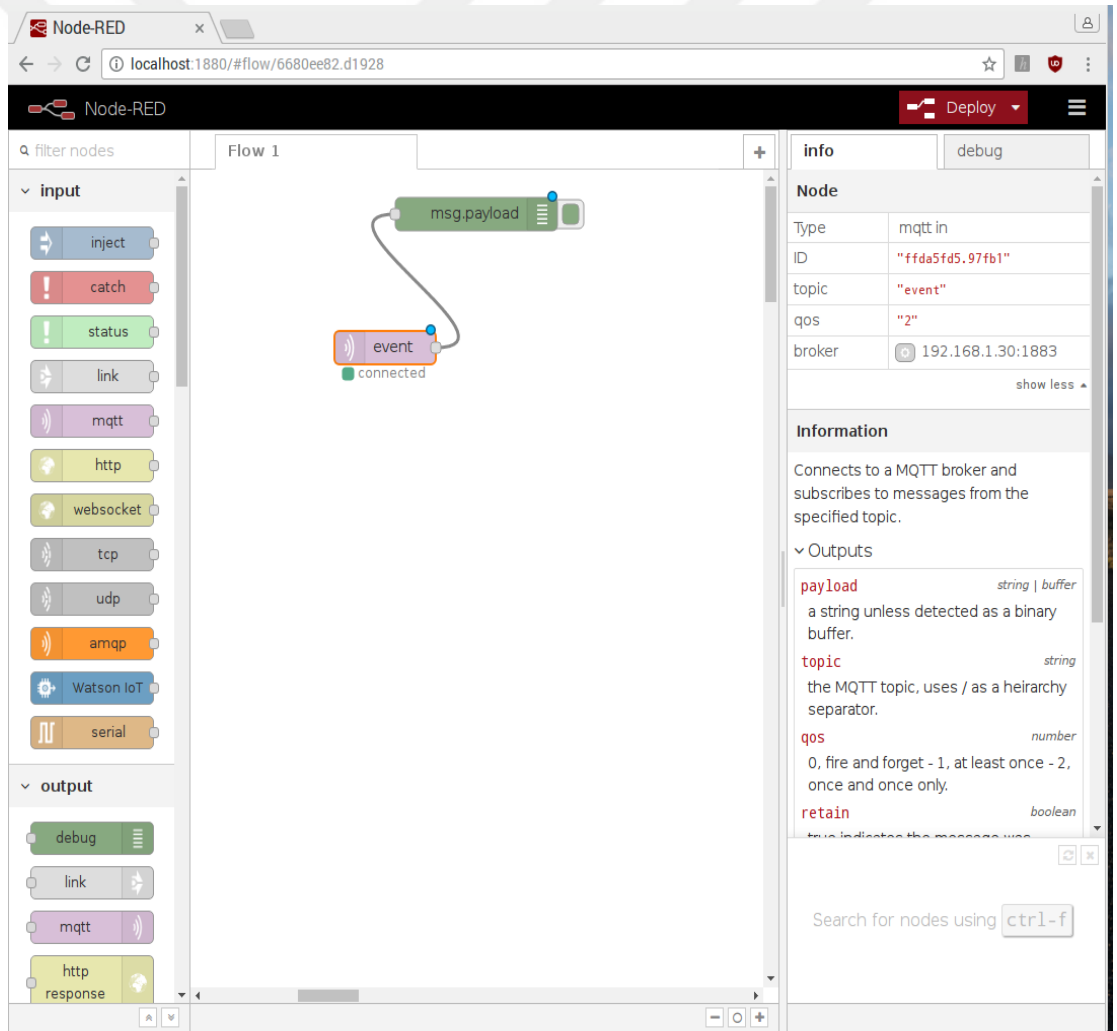
Yapılan testlerde 5 farklı boyuta sahip mesajlar kullanılmıştır. Online “lorem ipsum” üretici kullanılarak üretilen mesajlar boyutlarına göre şu şekildedir;

- 50 byte: “Praesent sagittis dignissim ipsum, eu dapibus sed.”
- 100 byte: “Nam consequat pellentesque lacus pretium interdum. Sed aliquet ante id velit eleifend bibendum amet.”
- 150 byte: “Nulla facilisi. Vivamus vehicula nisi purus, et vulputate libero sodales vitae. Sed sapien metus, efficitur quis eleifend posuere, bibendum non metus.”
- 200 byte: “Vestibulum quis ullamcorper sapien. Vivamus imperdiet tincidunt arcu, sit amet fringilla nisl pharetra vitae. Nullam id magna a nunc placerat blandit. Sed faucibus nulla ac est dignissim, luctus amet.”
- 250 byte: “Lorem ipsum dolor sit amet, consectetur adipiscing elit. Maecenas non purus vitae mi sagittis placerat id a ligula. Vivamus ultrices consequat arcu, quis luctus justo tempor sit amet. Nunc in consectetur sapien. Vestibulum felis diam, vulputate amet.”

5.4. MQTT Performans Testi

5.4.1. Test ortamının oluşturulması

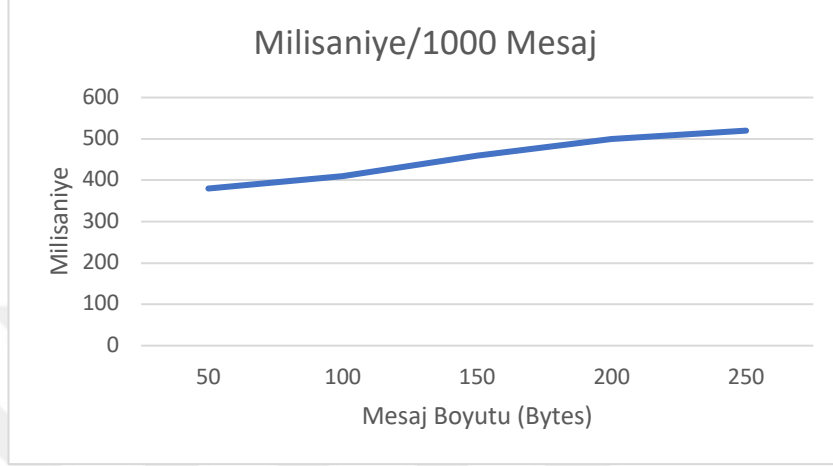
Öncelikli olarak MQTT testleri için C programlama dilinde yazılmış olan açık kaynak Mosquitto mesajlaşma broker'ı kullanılmıştır. Bu broker yazılımı Raspberry Pi a yüklenerek cihaz üzerinden çalışması sağlanmıştır. Ayrıca python programlama dili kullanılarak bir abone ve yayıncı uygulamaları yazılmıştır. Ayrıca Node-RED isimli NodeJS üzerinde çalışan uygulama ile MQTT broker üzerinde konfigürasyonlar yapılmıştır. Başlık ve hizmet kalitesi seviyesi belirlenerek sistem çalıştırılmıştır. Yapılan testler sırasında hizmet kalitesi seviyesi 2 olarak seçilmiştir.



Şekil 5-2 MQTT Test Ortamı

5.4.2. Performans sonuçları

Yayıncı uygulama üzerinden mesaj gönderimi başlatılarak süre ölçümlenmesi yapılmıştır. Çıkan sonuçlar şekil 5.3'deki gibidir.



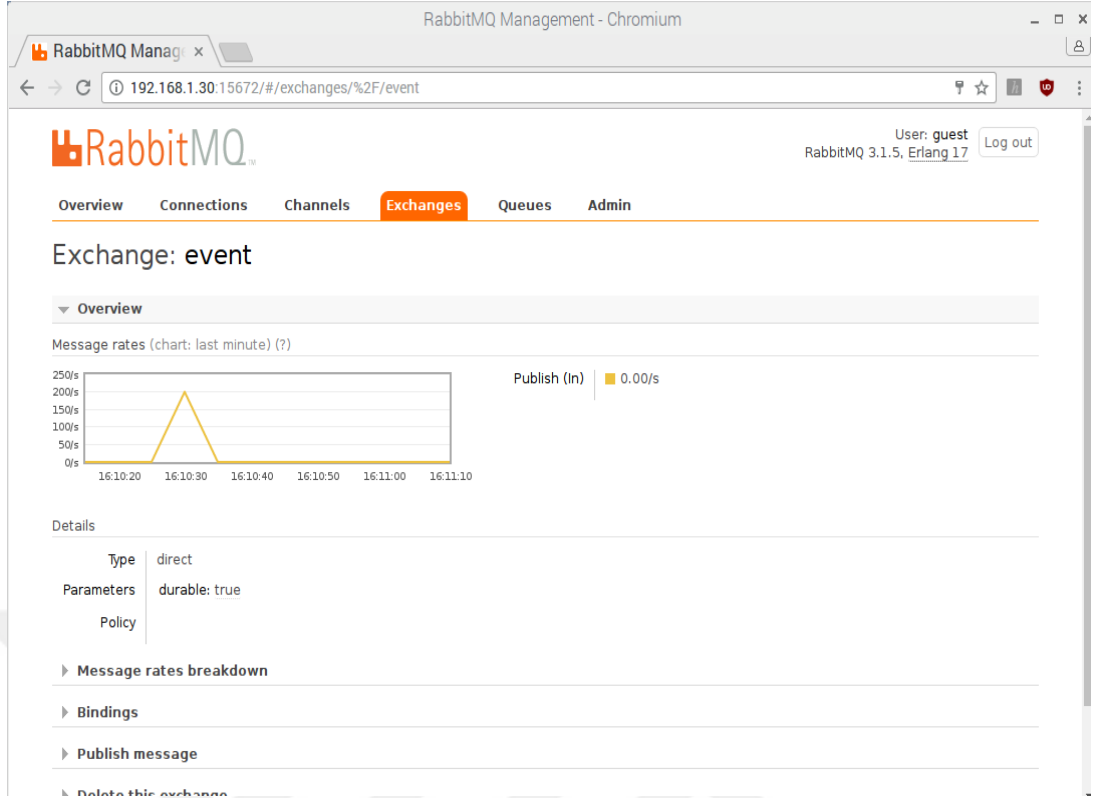
Şekil 5-3 MQTT Performans Sonuçları

MQTT testlerinde şekil 5-3 'deki grafiği de göz önünde bulundurursak mesaj boyutunun artması gönderim süresini etkilemiş hafifte olsa bir yükselme olmasına neden olmuştur. Ancak arka arkaya seri bir şekilde 1000 mesaj gönderildiğini düşünürsek süreleri kabul edilebilir değerler olarak görebiliriz.

5.5. AMQP Performans Testi

5.5.1. Test ortamının oluşturulması

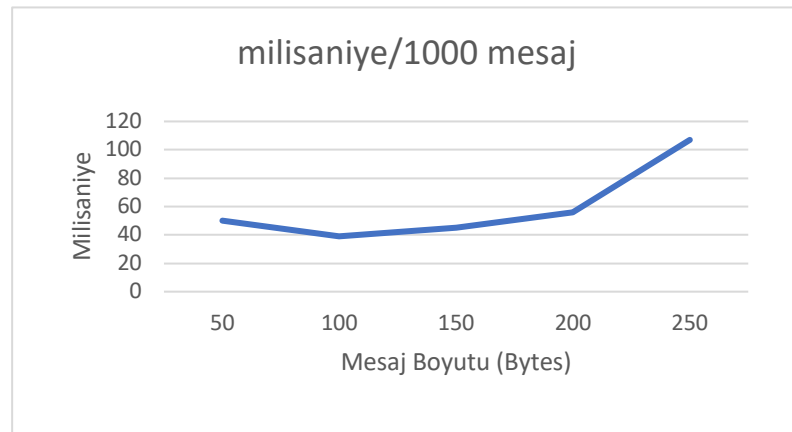
Öncelikli olarak AMQP protokolü için ise erlang dilinde yazılmış olan RabbitMQ mesajlaşma broker'ı kullanılmıştır. Bu broker yazılımı Raspberry Pi a yüklenerek cihaz üzerinden çalışması sağlanmıştır. Ayrıca python programlama dili kullanılarak bir abone ve yayıncı uygulamaları yazılmıştır.



Şekil 5-4 AMQP Test Ortamı

5.5.2. Performans sonuçları

Yayıncı uygulama üzerinden mesaj gönderimi başlatılarak süre ölçümlenmesi yapılmıştır. Çıkan sonuçlar şekil 5.5'deki gibidir.



Şekil 5-5 AMQP Performans Sonuçları

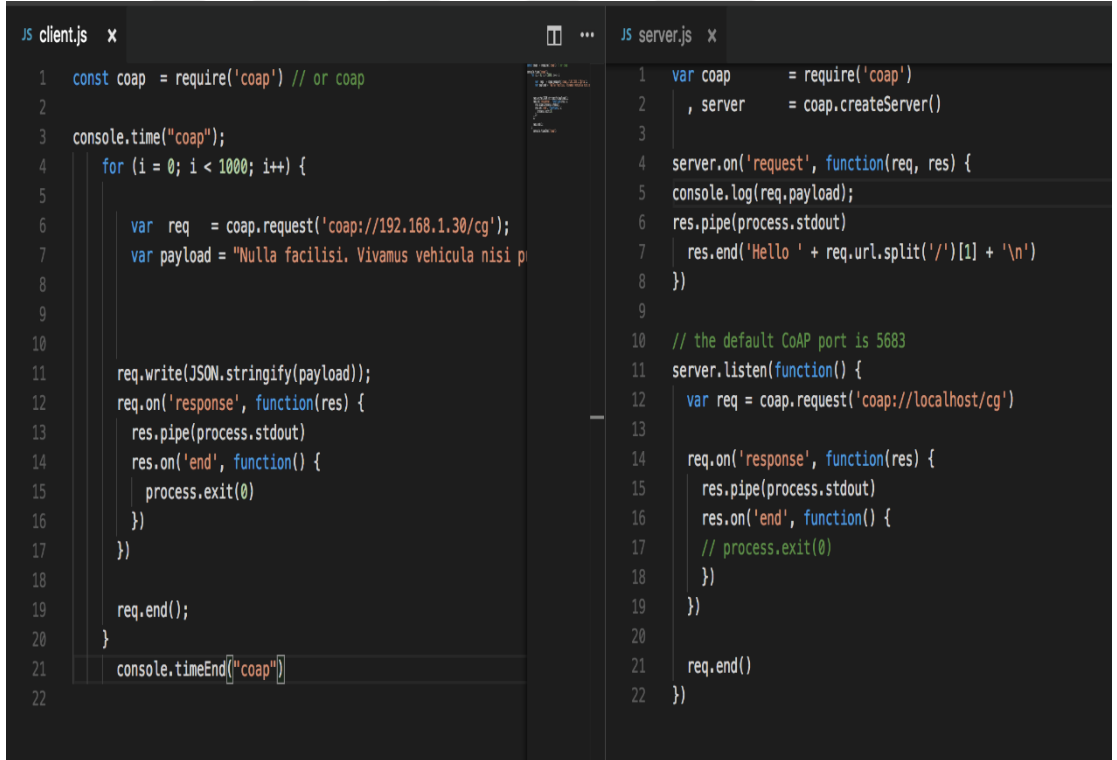
AMQP testlerinde şekil 5-5 'deki grafiği de göz önünde bulundurursak mesaj boyutunun artması önce gönderim süresinde bir azalmaya sonra hafifte olsa bir

yükselme olmasına neden olmuştur. AMQP diğer protokollere göre daha iyi performans sağlamış ve en düşük gönderim süreleri elde edilmiştir. Ayrıca testler sırasında kullanılan AMQP brokerlarından biri olan RabbitMQ bu alanda en iyi gelişim ve sonuçlara sahip brokerlardan biridir. Ayrıca hem AMQP'nin yapısı hemde brokerın özellikleri ile birlikte mesajları iletim durumu çok yüksektir.

5.6. COAP Performans Testi

5.6.1. Test ortamının oluşturulması

Öncelikli olarak COAP protokolü için ise javascript dilinde yazılmış olan ve NodeJS üzerinde çalışan bir server yazılımı kullanılmıştır. Ayrıca yine javascript programlama dili kullanılarak NodeJS üzerinde çalışan bir abone ve yayıncı uygulamaları yazılmıştır.



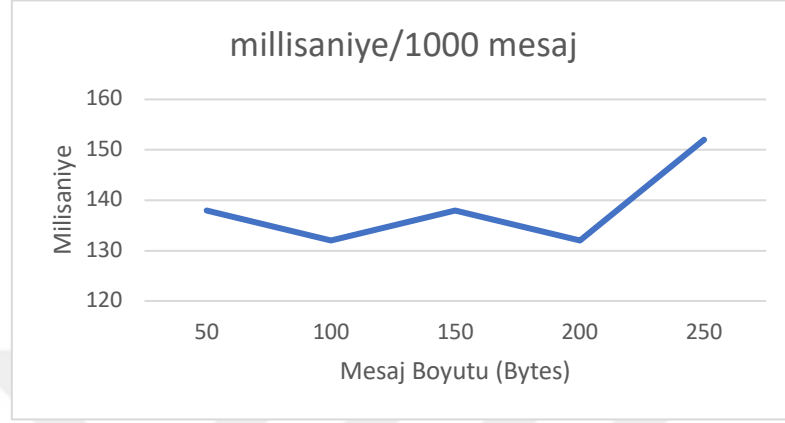
```
JS client.js x
1  const coap = require('coap') // or coap
2
3  console.time("coap");
4  for (i = 0; i < 1000; i++) {
5
6      var req = coap.request('coap://192.168.1.30/cg');
7      var payload = "Nulla facilisi. Vivamus vehicula nisi p
8
9
10
11     req.write(JSON.stringify(payload));
12     req.on('response', function(res) {
13         res.pipe(process.stdout)
14         res.on('end', function() {
15             process.exit(0)
16         })
17     })
18
19     req.end();
20 }
21 console.timeEnd("coap")
22

JS server.js x
1  var coap = require('coap')
2  , server = coap.createServer()
3
4  server.on('request', function(req, res) {
5  console.log(req.payload);
6  res.pipe(process.stdout)
7  res.end('Hello ' + req.url.split('/')[1] + '\n')
8  })
9
10 // the default CoAP port is 5683
11 server.listen(function() {
12     var req = coap.request('coap://localhost/cg')
13
14     req.on('response', function(res) {
15         res.pipe(process.stdout)
16         res.on('end', function() {
17             // process.exit(0)
18         })
19     })
20
21     req.end()
22 })
```

Şekil 5-6 CoAP Test Ortamı

5.6.2. Performans sonuçları

Yayıncı uygulama üzerinden mesaj gönderimi başlatılarak süre ölçümlenmesi yapılmıştır. Çıkan sonuçlar şekil 5.7 'deki gibidir.



Şekil 5-7 COAP Gönderim Performansı

COAP testlerinde şekil 5-7 'deki grafiği de göz önünde bulundurursak mesaj boyutunun artması gönderim süreleri küçük düzeyde oynamalar ile birbirine yakın değerler olarak gözlemlenmiştir. Ayrıca cihazın network ve işlemci kaynak durumuna göre diğer protokollerde de olabilecek ufak dalgalanmalar gözlemlenebilmektedir.

5.7. XMPP Performans Testi

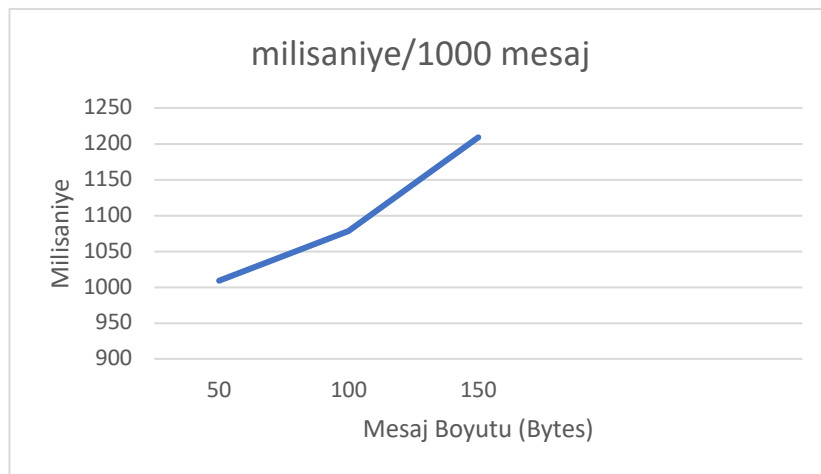
5.7.1. Test ortamının oluşturulması

Öncelikli olarak XMPP protokolü için ise ejabbered XMPP sunucusu kullanılmıştır. Ayrıca yine python programlama dili kullanılarak nodejs üzerinde çalışan bir abone ve bir yayıncı uygulaması yazılmıştır.

Şekil 5-8 XMPP Test Ortamı

5.7.2. Performans sonuçları

Yayıncı uygulama üzerinden mesaj gönderimi başlatılarak süre ölçümlenmesi yapılmıştır. Çıkan sonuçlar şekil 5.9 'deki gibidir.



Şekil 5-9 XMPP Sonuçları

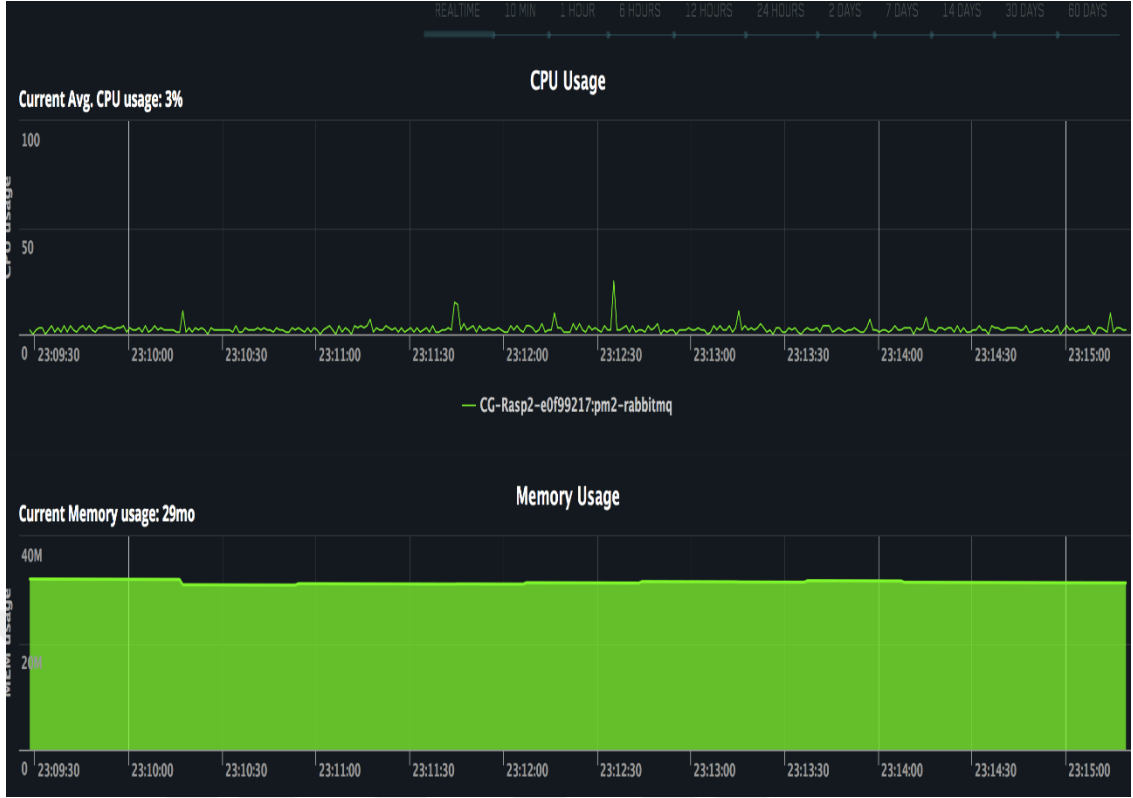
XMPP protokolü üzerinden yapılan denemelerde bir takip sorunlar ile karşılaşmıştır. 150 byte'ın üzerinde seri olarak 1000 mesaj gönderimi sunucunun kilitlemesine ve cevap veremez hale gelmesine neden olmuştur. Bunun üzerinde bu boyutlarda farklı sayıda denemeler yapılmıştır ve maksimum anlık olarak 300 mesaj gönderimi sağlanmıştır. Bu rakamın üzerine çıkıldığında sistemsel sorunlar yaşandığı görülmektedir.

Bu sorunun başlıca nedenleri olarak Raspberry cihazların yetersiz kalması, sunucu yazılımının eksikleri, hatalı konfigürasyon yapılmış olması protokolün mimarisinden kaynaklanan yetersizlikler olarak düşünebiliriz. Bu bağlamda bu sorunu aşmak için daha yüksek özellikli altyapı ve daha iyi optimize edilmiş bir sunucu alt yapısı kullanılabilir.

Örnek vermek gerekirse dünya üzerinde en çok kullanılan iletişim uygulaması olan WhatsApp altyapı olarak protokolünü kullanmaktadır. WhatsApp altyapı uygulama yazılımlarını Erlang programlama dili kullanarak geliştirmiş ve dağıtık arttırılabilir bir sunucu altyapısı oluşturmuştur. Bu yüzden yapılan testlerde gönderim sırasında hatalar alınsa da daha geliştir altyapı ve yazılımlar kullanarak yüksek bir performans sağlanabilmektedir.

5.8. İşlemci ve Ağ kullanım Oranları

Yapılan testler sırasında ayrıca işlemci ve ağ kullanımları da gözlenmiştir. Protokoller için genel olarak düşünecek olursak mesaj gönderimleri sırasında anlık olarak işlemci kullanımları %30-40 seviyesinde olmuştur. Ağ kullanımı ise yaklaşık olarak %3-6 seviyesinde seyretmiştir. Nesnelerin interneti için düşündüğümüzde cihazlar üzerinden belirli aralıklarda mesaj iletimi olacağını göz önünde bulundurursak bu değerlerin kabul edilebilir seviye de olduğunu varsayabiliriz.



Şekil 5-10 Raspberry Pi Kaynak Kullanımı

6. SONUÇ VE ÖNERİLER

Sonuç olarak çağımızın teknoloji çağı olduğunu ve internetin hızla daha çok cihaza eriştiğini düşünürsek, gelişmekte olan sanayide nesnelerin interneti konusu oldukça önemli bir hal almaktadır. Özellikle hayatı kolaylaştıran uygulamalar yapmada, akıllı şehirler ve akıllı evler tasarlamada hızla yaygınlaşan bu teknoloji ilerde daha çok yaşamın bir parçası olacaktır. Bu yüzden bu iletişimde kullanacağımız altyapı ve protokollerin seçimi, sistemin kendini hızla geliştirmesi, enerji tasarrufu ve kullanılabilirlik açısından çok dikkat edilmesi gereken hassas bir konu olmuştur. Yapılan incelemeler dikkate alınarak bu teknolojiler daha da gelişmeye ve hayatımızın bir parçası olmaya artarak devam edecektir.

6.1. Araştırma Bulguları

Bu araştırmada nesnelerin internetinde uygulama katmanında kullanılan protokoller incelenerek, her bir protokolün mimari yapısı, iletişim modeli, güvenliği ve hizmet kalitesinin değerlendirmesi açısından incelemeler ve testler yapılmıştır. Yapılan bu testler sonucunda protokoller arasında AMQP veri iletimi açısından en hızlı protokol olmuştur. Her protokolün kendine has avantajları ve dezavantajları vardır. Bu bağlamda geliştirmek istediğimiz ürünün gerekliliklerini düzgün olarak belirlememiz ve kullanacağımız altyapı teknolojilerini buna göre belirlememiz gerekmektedir. Bu araştırmada önemli noktalardan biri olan cihazlar arasındaki iletişim sağlanması konusuna değinilmiştir. Burada avantajlara ve dezavantajlara da dikkat edip uygulama aşamasına geçerken bizim için en uygun seçimi yapmamız gerekmektedir.

6.2. Gelecek Çalışmalar

Bu araştırma iletişim altyapısı kurulurken kablolu bir bağlantı kullanılmıştır. Ancak her uygulama ortamında böyle bir yapı kurulması mümkün olmayabilir. Bu yüzden sonraki çalışmalarda kablosuz teknolojilerden Wi-fi ve özellikle düşük güç tüketimi ile dikkat çeken Bluetooth'un BLE teknolojisi üzerinden testler ve incelemeler yapılması hedeflenmektedir. Bunların yanı sıra yine Bluetooth üzerinden iletişim kuran iBeacon teknolojisi de incelenmek istenmektedir.

6.3. Öneriler

6.3.1. Farklı bağlantı türleri ile birlikte kullanımı

Bu araştırmada oluşturduğumuz test ortamında cihazların birbirleriyle olan iletişimi kablolu bir ağ altyapısı oluşturarak sağlamıştır. Ancak nesnelerin internetindeki cihazların kullanım yerlerini düşündüğümüzde farklı ağ altyapılarına ihtiyaç olabileceğini gözlemleyebiliriz. Oluşturulacak uygulamaya yapılacak olan kurgu ya göre daha sıklıkla kablosuz iletişim daha sıklıkla kullanılacaktır. Buradaki bağlantı yapılarından kısaca şöyle bahsedebiliriz.

6.3.1.1. WI-FI (kablosuz bağlantı) kullanımı

Gerekli olan ağ cihazları kullanılarak oluşturulacak bir ağ altyapısının kullanımı sağlanabilir. Burada dikkat edilmesi gereken konular ise, sisteme bağlanılacak cihazlarla iletişimin yeteri kadar sağlanabilmesidir. Yani cihazların bulunduğu noktadaki çekim gücü olarak da söyleyebiliriz. Oluşturulan altyapı üzerinden seçilmiş olan protokol kullanılarak cihazlar arası iletişim sağlanabilir. Burada cihazların güç durumları da çok önemli bir rol oynamaktadır. Üzerlerinde bulunduracak olan Wi-Fi modülünün güç ihtiyaçlarını karşılayabilecek düzeyde olması gerekmektedir.

6.3.1.2. Bluetooth kullanımı

Nesnelerin internetinde cihazlar arasındaki iletişimi sağlamanın bir yolu da Bluetooth teknolojisidir. Cihazlar üzerindeki Bluetooth modüllerini kullanarak bir iletişim altyapısı oluşturulabilir. Özellikle son zamanlarda bu teknolojinin gelişmesi, Bluetooth'un üst versiyonlarında bağlantı mesafesinin artması, anlık oluşturulabilecek bağlantı sayısının artması bu teknolojinin kullanımında önemli rol oynamaktadır. Ayrıca son olarak çıkan BLE teknolojisi Bluetooth iletişimine düşük enerji kullanımı ile avantaj sağlamıştır.

Nesnelerin internetinde kullanacağımız cihazların enerji ihtiyaçları göz önünde bulundurduğumuzda daha düşük enerjili bir iletişim altyapısı oluşturmak çok önemli bir hal almaktadır. Çünkü cihazları her zaman doğrudan bir enerji hattına bağlamak

mümkün olmayabilir. Bu yüzden BLE teknolojisi iletişim konusunda büyük bir avantaj sağlayabilir.

6.3.2. Sektörel öneriler

Nesnelerin interneti günümüzde özellikle elektronik eşyalar alanında çok büyük ilerlemeler kaydetmekte ve bu alanda birçok Ar-Ge çalışması yapılmaktadır. Teknoloji alanındaki üst düzey firmalar özellikle akıllı ev konseptinde birçok ürün çıkarmış ve çıkarmaya devam etmektedir. Bunların yanında birçok beyaz eşya da evlerimizde ki internete bağlanarak cep telefonundaki uygulamalar yardımı ile kontrol edilebilir hale geldi. Örneğin eve gelmeden evdeki klimanızın çalışmasını sağlayabilirsiniz. Bunların yanında nesnelerin interneti ile birlikte akıllı asistanlarda evlerimizde yer almaya başladı. Firmalarda bu konuda geliştirdikleri ürünlerle sadece seslenerek evimizdeki birçok cihazı kontrol etmemizi sağlayacak ürünler geliştirmekte ve yapay zekayı da bu alana entegre ederek yaşamı kolaylaştıracak ürünlerle tüketicinin karşısına çıkmaktadır.

Bunun yanında kurumsal alanda kullanılacak birçok uygulamada çıktı ve çıkmaya da devam etmektedir. Kurumsal alanda genellikle otomasyon uygulamaları göz önünde olduğu gibi iş takibi, ortam durum takibi gibi kullanımlara da sıkça rastlanmaktadır. Örnek vermek gerekirse mesela bir çiftlikte hayvanların durumu, ortamdaki nem, sıcaklık, ışık miktarı vb. gibi çeşitli parametreler sensörler yardımıyla ana sistemlere aktarılıp çiftçinin takibini kolaylaştırmakta aynı zamanda olası acil durumlarda da hızlı müdahaleyi kolaylaştırmaktadır.

Her ne kadar birçok ürün ve uygulama piyasaya çıkmış olsa da nesnelerin interneti gelişmeye çok açık bir alandır. Bu alanda yapılabilecek bir sürü uygulama fikri şu an geliştirilmekte ya da geliştirme için beklemektedir. Burada en önemli konular özellikle hangi sektöre hitap edileceği, nasıl bir ortama kurulum yapılacağı veya hangi ortamlarda kullanılacağı, sağlanacak fayda gibi bir sürü kritere dikkat edilmesi gerekmektedir. Özellikle günümüzde enerji tasarrufu çok önemli olduğundan önem verilmesi gereken bir noktadır. Bunun yanında doğru hedefe doğru ürünü yapmak en önemli konudur.

KAYNAKLAR

- M. B. Yassein, M. Q. Shatnawi et al., "Application layer protocols for the internet of things: A survey," in Engineering And MIS (ICEMIS), International Conference on. IEEE, 2016, pp. 1–4.
- Karagiannis, V.: A survey on application layer protocols for the internet of things. Trans. IoT Cloud Comput. 3(1), 11–17 (2015)
- Fysarakis, K., Askoxylakis, I., Soultatos, O., Papaefstathiou, I., Manifavas, C., Katos, V.: Which IoT protocol? Comparing standardized approaches over a common M2M application. In: 2016 IEEE Global Communications Conference (GLOBECOM), pp. 1–7. IEEE (2016)
- AMQP Organization, 2018- AMQP Differentiated, Eriřim Tarihi: 14.05.2017
<http://www.amqp.org/product/different>
- Greer D., 2012, RabbitMQ for Windows: Exchange Types, Eriřim Tarihi:15.05.2017
<https://lostechies.com/derekgreer/2012/03/28/rabbitmq-for-windows-exchange-types/>
- Kefeli C., Jabber(Extensible Messaging and Presence Protocol-XMPP) nedir?
<http://www.cemkefeli.com/post/2009/10/19/Jabber-nedir.aspx>
- Nesta Elektronik, IoT Protokollerine Genel Bir Bakıř, Eriřim Tarihi:01.06.2017
http://www.nestaelektronik.com/blog/index.php?article=iot_review
- PrismTech Corporation, 2017, Messaging Technologies for the Industrial Internet and the Internet of Things Whitepaper, Eriřim Tarihi:05.07.2017
<http://www.prismttech.com/sites/default/files/documents/Messaging-Whitepaper-051217.pdf>
- RabbitMQ Organization, AMQP 0-9-1 Model Explained, Eriřim Tarihi:14.05.2017
<https://www.rabbitmq.com/tutorials/amqp-concepts.html>
- Shelby, Z., Hartke, K., & Bormann, C. (2014). The constrained application protocol
- řİMřEK M., CoAP (Constrained Application Protocol) Eriřim Tarihi:14.05.2017
<http://www.iotturkiye.com/post/144770011545/coap-constrained-application-protocol-kısıtlı>
- Yelis B. / Haberleřme Protokollerinde Endüstri 4.0 Devrimi | MQTT Eriřim Tarihi: 14.05.2017, <http://www.endustri40.com/haberlesme-protokollerinde-endustri-4-0-devrimi-mqtt/>
- Yetimler E. Nesnelerin İnterneti Nedir? Cihazların Etkileřim Trendleri, Eriřim Tarihi:06.01.2017, <https://www.karel.com.tr/blog/internet-things-nesnelerin-interneti-nedir-cihazlarin-etkilesim-trendleri>

Gültunca C., Zaim A., 2017, Examination and Comparison of the Communication Protocols on the Application Layer in IOT. 6th International Conference on Advanced Technology & Sciences



ÖZGEÇMİŞ

Adı Soyadı : Cem GÜLTUNCA
Doğum Yeri ve Yılı : İstanbul, 04/08/1990
Medeni Hali : Bekar
Yabancı Dili : İngilizce
E-posta : info@cemgultunca.com.tr

Eğitim Durumu

Lise : Kültür Fen Lisesi, 2007
Lisans : İstanbul Ticaret Üniversitesi, Mühendislik ve Tasarım
Fakültesi, Bilgisayar Mühendisliği Bölümü

Mesleki Deneyim

Kariyer.net,
Yazılım Uzmanı Eylül 2012/ Kasım 2014
Nesine.com,
Yazılım Uzmanı Aralık 2014/ Eylül 2017
Triodor Ar-Ge,
Yazılım Uzmanı Ekim 2017/devam ediyor

Yayınları

Çelik D., Karakaş A., Bal G., Gültunca C., Elçi A., Buluz B., Aevli M.C., 2013.
Towards an Information Extraction System Based on Ontology to Match
Resumes and Jobs. Computer Software and Applications Conference
Workshops (COMPSACW), 2013 IEEE 37th Annual
Karakaş A., Bal G., Gültunca C., Tüngör T., 2014. Özgeçmişlerden ve İş
İlanlarından Doğal Dil İşleme Yöntemleri İle Bilgi Çıkarımı. Bilişim 2014-
TBD 31.Ulusal Bilişim Kurultayı
Gültunca C., Zaim A., 2017, Examination and Comparison of the Communication
Protocols on the Application Layer in IOT. 6th International Conference on
Advanced Technology & Sciences