KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ
ANABİLİM DALI

YÜKSEK LİSANS TEZİ

NESNELERİN İNTERNETİ TEKNOLOJİLERİNDE FARKLI HABERLEŞME SENARYOLARININ DDS-XRCE STANDARDI İÇİN PERFORMANSININ BELİRLENMESİ

ŞEVVAL ŞOLPAN

KOCAELİ 2023

KOCAELİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

BİLGİSAYAR MÜHENDİSLİĞİ
ANABİLİM DALI

YÜKSEK LİSANS TEZİ

NESNELERİN İNTERNETİ TEKNOLOJİLERİNDE FARKLI HABERLEŞME SENARYOLARININ DDS-XRCE STANDARDI İÇİN PERFORMANSININ BELİRLENMESİ

ŞEVVAL ŞOLPAN

Prof.Dr. Kerem KÜÇÜK
Danışman, Kocaeli Üniversitesi .....
Dr. Öğr. Üyesi Meltem KURT PEHLİVANOĞLU
Jüri Üyesi, Kocaeli Üniversitesi .....
Dr. Öğr. Üyesi Yunus ÖZEN
Jüri Üyesi, Yalova Üniversitesi .....

Tezin Savunulduğu Tarih: 27.12.2022

ETİK BEYAN VE ARAŞTIRMA FONU DESTEĞİ

Kocaeli Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmasında,

- Bu tezin bana ait, özgün bir çalışma olduğunu,
- Çalışmamın hazırlık, veri toplama, analiz ve bilgilerin sunumu olmak üzere tüm aşamalarında bilimsel etik ilke ve kurallara uygun davrandığımı,
- Bu çalışma kapsamında elde edilen tüm veri ve bilgiler için kaynak gösterdiğimi ve bu kaynaklara kaynakçada yer verdiğimi,
- Bu çalışmanın Kocaeli Üniversitesi'nin abone olduğu intihal yazılım programı kullanılarak Fen Bilimleri Enstitüsü'nün belirlemiş olduğu ölçütlere uygun olduğunu,
- Kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- Tezin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı,

beyan ederim.

☒ Bu tez çalışmasının herhangi bir aşaması hiçbir kurum/kuruluş tarafından maddi/alt yapı desteği ile desteklenmemiştir.

☐ Bu tez çalışması kapsamında üretilen veri ve bilgiler ..... tarafından ..... no'lu proje kapsamında maddi/alt yapı desteği alınarak gerçekleştirilmiştir.

Herhangi bir zamanda, çalışmamla ilgili yaptığım bu beyana aykırı bir durumun saptanması durumunda, ortaya çıkacak tüm ahlaki ve hukuki sonuçları kabul ettiğimi bildiririm.

.....

Şevval ŞOLPAN

YAYIMLAMA VE FİKRİ MÜLKİYET HAKLARI

Fen Bilimleri Enstitüsü tarafından onaylanan lisansüstü tezimin tamamını veya herhangi bir kısmını, basılı ve elektronik formatta arşivleme ve aşağıda belirtilen koşullarla kullanıma açma izninin Kocaeli Üniversitesi'ne verdiğimi beyan ederim. Bu izinle Üniversiteye verilen kullanım hakları dışındaki tüm fikri mülkiyet hakları bende kalacak, tezimin tamamının ya da bir bölümünün gelecekteki makale, kitap, tebliğ, lisans, patent gibi çalışmalarda kullanımı, danışmanımın isim hakkı saklı kalmak koşuluyla ve her iki tarafın bilgisi dâhilinde bana ait olacaktır.

Tezin kendi özgün çalışmam olduğunu, başkalarının haklarını ihlal etmediğimi ve tezimin tek yetkili sahibi olduğumu beyan ve taahhüt ederim. Tezimde yer alan telif hakkı bulunan ve sahiplerinden yazılı izin alınarak kullanılması zorunlu metinlerin yazılı izin alarak kullandığımı ve istenildiğinde suretlerini Üniversiteye teslim etmeyi taahhüt ederim.

Yükseköğretim kurulu tarafından yayınlanan ***"Lisanüstü Tezlerin Elektronik Ortamda Toplanması, Düzenlenmesi ve Erişime Açılmasına İlişkin Yönerge"*** kapsamında tezim aşağıda belirtilen koşullar haricinde YÖK Ulusal Tez Merkezi/ Kocaeli Üniversitesi Kütüphaneleri Açık Erişim Sisteminde erişime açılır.

☐ Enstitü yönetim kurulu kararı ile tezimin erişime açılması mezuniyet tarihinden itibaren 2 yıl ertelenmiştir.

☐ Enstitü yönetim kurulu gerekçeli kararı ile tezimin erişime açılması mezuniyet tarihinden itibaren 6 ay ertelenmiştir.

☒ Tezim ile ilgili gizlilik kararı verilmemiştir.

.....

Şevval ŞOLPAN

ÖNSÖZ VE TEŞEKKÜR

Gelişen teknoloji çeşitli yeni buluşları meydana getirmiştir. Bir zamanlar imkansız gibi görünen ve dünyanın birbirinden uzak herhangi iki noktası arasındaki iletişimi mümkün kılan İnternet bunlardan biridir. Bu buluşların insan hayatını kolaylaştırmasının yanında, diğer teknolojik cihazlarla birlikte kullanımları teknolojinin gelişimine ivme kazandırmıştır. İnternet'in doğuşu, Nesnelerin İnterneti (Internet of Things, IoT) kavramının oluşmasına zemin hazırlayan „birbirleriyle haberleşen makineler", „akıllı cihazlar" gibi fikirleri ortaya çıkarmıştır.

Çeşitli ortamlardan sensörler yardımıyla verinin toplandığı, birçok kişi veya cihazın erişmesine imkan sağlayan bulut platformlarda verinin depolandığı ve analiz edildiği karmaşık sistemler Nesnelerin İnterneti teknolojisinin bir örneğidir. Nesnelerin İnterneti teknolojileri veriye erişimi kolaylaştırmakla birlikte insan müdahalesini azaltmaktadır. Bütün bunları kendisini etkin kılan çeşitli disiplinlerin birbiriyle uyumlu çalışmasıyla başarmaktadır. Buradan hareketle, bir Nesnelerin İnterneti sistemi oluşturmak istendiğinde dikkatle ele alınması gereken çeşitli hususlar da olacaktır. Nesnelerin İnterneti, birbirleriyle haberleşen çeşitli cihazlar üzerine kurulu bir teknoloji olduğu için iletişim protokolleri bu hususlardan biridir.

Bu nedenle bu tez çalışmasında, yakın zamanda yayınlandığı için hakkında çok çalışma yapılmamış ve bir uygulama katmanı protokolü olan Son Derece Kaynak Kısıtlı Ortamlar İçin Veri Dağıtım Hizmeti (Data Distribution Service for Extremely Resource-Constrained Environments, DDS-XRCE) standardının ağ performansını değerlendirmeye yönelik birtakım testler yapılmıştır. Testler, oluşturulan birkaç senaryoda DDS-XRCE ağının izlenmesi ve ağdan verilerin toplanmasıyla gerçekleştirilmiştir.

Yüksek Lisans öğrenimim boyunca desteğini esirgemeyen ve karşılaştığım güçlüklerde yol gösteren danışmanım Prof. Dr. Kerem KÜÇÜK'e teşekkürlerimi sunarım. Hayatım boyunca desteğini esirgemeyen ve karşılaştığım güçlüklerde yol gösteren aileme teşekkürlerimi sunarım.

Kasım – 2022 Şevval ŞOLPAN

İÇİNDEKİLER

ETİK BEYAN VE ARAŞTIRMA FONU DESTEĞİ i
YAYIMLAMA VE FİKRİ MÜLKİYET HAKLARI ii
ÖNSÖZ VE TEŞEKKÜR iii
ŞEKİLLER DİZİNİ vi
TABLOLAR DİZİNİ vii
SİMGELER VE KISALTMALAR viii
ÖZET ix
ABSTRACT x
1. GİRİŞ 1
2. LİTERATÜR ÖZETİ 4
3. KULLANILAN MATERYALLER 9
3.1. Veri Dağıtım Hizmeti (Data Distribution Service, DDS) 9
3.1.1. DDS'nin Yapısı 9
3.1.2. DDS İle İlgili Nesneler ve Kavramlar 10
3.1.3. Veri Merkezli Yayım Yapma Abone Olma (Data-Centric Publish-Subscribe, DCPS) 10
3.1.4. DDS Varlıkları ve Ara Yüzleri Oluşumu 11
3.1.5. DDS Mesajları ve Yapısı 13
3.2. Son Derece Kaynak Kısıtlı Ortamlar İçin Veri Dağıtım Hizmeti (Data Distribution Service for Extremely Resource-Constrained Environments, DDS-XRCE) 15
3.2.1. DDS-XRCE İle İlgili Kavramlar 15
3.2.2. DDS ve DDS-XRCE'nin Bütünleşmesi 16
3.2.3. DDS-XRCE'nin Yapısı 17
3.2.4. DDS-XRCE Nesne Modeli 20
3.2.5. DDS-XRCE Varlıkları Oluşumu 21
3.2.6. DDS-XRCE Mesajları ve Yapısı 21
3.3. eProsima Yazılım Ürünleri 23
3.3.1. eProsima Micro-XRCE-DDS-Agent v2.0.0 23
3.3.2. eProsima Micro-XRCE-DDS-Client v2.0.0 24
3.3.3. eProsima Micro-XRCE-DDS-Gen 24
3.3.4. eProsima Fast-DDS v2.4.0 25
3.3.5. eProsima Fast-DDS-Gen 25
3.4. Wireshark 26
3.5. Gnome Terminator 26
4. KULLANILAN TEST SENARYOLARI VE YÖNTEMLER 27
4.1. Hazırlık Aşaması 27
4.1.1. Senaryoların Oluşturulması 27
4.1.2. Farklı Koşulların ve Konu Çalışma Alanlarının Oluşturulması 32
4.1.3. DDS-XRCE Çalışma Alanlarındaki İlgili Kaynak Kodların Düzenlenmesi 37
4.1.4. "Deployment" Özellikli DDS-XRCE Çalışma Alanlarındaki Kaynak Kodların Düzenlenmesi 40
4.1.5. DDS Çalışma Alanlarındaki Kaynak Kodların Düzenlenmesi 46

4.2. Simülasyon ve Veri Toplama Aşaması 47
4.3. Analiz Aşaması 48
5. PERFORMANS VE TEST SONUÇLARI 52
6. SONUÇLAR VE ÖNERİLER 60
KAYNAKLAR 62
EKLER 66
KİŞİSEL YAYIN VE ESERLER 68
ÖZGEÇMİŞ 69

ŞEKİLLER DİZİNİ

Şekil 3.1. DDS'nin yapısı 9
Şekil 3.2. DDS varlıklarının oluşum ağacı 12
Şekil 3.3. Dinleyici ara yüzlerinin oluşum ağacı 12
Şekil 3.4. RTPS protokolünün mesaj yapısı 13
Şekil 3.5. DDS ve DDS-XRCE'nin bütünleşmesi 16
Şekil 3.6. Dağıtım modelleri 18
Şekil 3.7. DDS-XRCE varlıklarının oluşum ağacı 21
Şekil 3.8. DDS-XRCE protokolünün mesaj yapısı 21
Şekil 4.1. Dağıtım Modeli-1 28
Şekil 4.2. Dağıtım Modeli-2 29
Şekil 4.3. Dağıtım Modeli-3 30
Şekil 4.4. Bir DDS çalışma alanının dosya ağacı 34
Şekil 4.5. Bir DDS-XRCE çalışma alanının dosya ağacı 35
Şekil 4.6. Deployment özellikli bir DDS-XRCE çalışma alanının dosya ağacı 36
Şekil 4.8. Yayımcı ve abone kaynak kodlarındaki TCP taşıma kod parçacıkları 38
Şekil 4.7. Yayımcı ve abone kaynak kodlarındaki UDP taşıma kod parçacıkları 37
Şekil 4.10. Abone kaynak kodlarındaki en iyi çaba akış modu kod parçacıkları 39
Şekil 4.9. Yayımcı kaynak kodlarındaki en iyi çaba akış modu kod parçacıkları 38
Şekil 4.11. Yayımcı kaynak kodlarındaki güvenilir akış modu kod parçacıkları 39
Şekil 4.12. Abone kaynak kodlarındaki güvenilir akış modu kod parçacıkları 40
Şekil 4.13. ConfiguratorClient, yayımcı ve abone kodlarındaki UDP kod parçacıkları 41
Şekil 4.14. ConfiguratorClient, yayımcı ve abone kodlarındaki TCP kod parçacıkları 41
Şekil 4.15. ConfiguratorClient kodlarındaki en iyi çaba akış modu kod parçacıkları 42
Şekil 4.16. Yayımcı kaynak kodlarındaki en iyi çaba akış modu kod parçacıkları 43
Şekil 4.17. Abone kaynak kodlarındaki en iyi çaba akış modu kod parçacıkları 43
Şekil 4.18. ConfiguratorClient kodlarındaki güvenilir akış modu kod parçacıkları 44
Şekil 4.20. Abone kaynak kodlarındaki güvenilir akış modu kod parçacıkları 45
Şekil 4.19. Yayımcı kaynak kodlarındaki güvenilir akış modu kod parçacıkları 45
Şekil 4.21. Farklı konular için IDL dosya kodları 46
Şekil 4.22. Simülasyon adımları 47
Şekil 4.23. Yayımlanan bir veri paketinin protokol ve boyut bilgileri 50

TABLOLAR DİZİNİ

Tablo 3.1. RTPS varlık alt mesajı tipleri 14
Tablo 3.2. RTPS yorumlayıcı alt mesaj tipleri..... 14
Tablo 3.3. Dağıtım modelleri ve detayları..... 18
Tablo 3.4. DDS-XRCE alt mesaj tipleri 22
Tablo 4.1. Dağıtım Modeli-1 ve detayları 28
Tablo 4.2. Dağıtım Modeli-2 ve detayları 29
Tablo 4.3. Dağıtım Modeli-3 ve detayları 31
Tablo 4.4. Yapılan testler ve sonlandırılma limitleri..... 48
Tablo 5.1. Dağıtım Modeli-1'in gecikme sonuçları 52
Tablo 5.2. Dağıtım Modeli-2'nin gecikme sonuçları 53
Tablo 5.3. Dağıtım Modeli-3'ün gecikme sonuçları..... 54
Tablo 5.4. Dağıtım Modeli-1'in verim sonuçları..... 55
Tablo 5.5. Dağıtım Modeli-2'nin verim sonuçları..... 56
Tablo 5.6. Dağıtım Modeli-3'ün verim sonuçları..... 56
Tablo 5.7. Dağıtım Modeli-1'in paket kaybı sonuçları 57
Tablo 5.8. Dağıtım Modeli-2'nin paket kaybı sonuçları 58
Tablo 5.9. Dağıtım Modeli-3'ün paket kaybı sonuçları..... 58

SİMGELER VE KISALTMALAR DİZİNİ

Bps.	:	Bit per second (saniye başına bit)
Mbps.	:	Mega bit per second (saniye başına mega bit)
Ms.	:	Millisecond (milisaniye)

Kısaltmalar

ABD	:	Amerika Birleşik Devletleri
AMQP	:	Advanced Message Queuing Protocol (Gelişmiş Mesaj Sıralama Protokolü)
CAN FD	:	Controller Area Network Flexible Data-Rate (Denetleyici Alan Ağı Esnek Veri Hızı)
CDR	:	Common Data Representation (Ortak Veri Temsili)
CoAP	:	Constrained Application Protokol (Kısıtlı Uygulama Protokolü)
CPU	:	Central Processing Unit (Merkezi İşlem Birimi)
DCPS	:	Data-Centric Publish-Subscribe (Veri Merkezli Yayım Yapma-Abone Olma)
DDS	:	Data Distribution Service (Veri Dağıtım Hizmeti)
DDS-XRCE	:	Data Distribution Service for Extremely Resource-Constrained Environments (Son Derece Kaynak Kısıtlı Ortamlar İçin Veri Dağıtım Hizmeti)
IDL	:	Interface Definition Language (Arayüz Tanımlama Dili)
IoT	:	Internet of Things (Nesnelerin İnterneti)
IP	:	Internet Protocol (İnternet Protokol)
K8s	:	Kubernetes
MQTT	:	Message Queuing Telemetry Transport (Mesaj Kuyruğu Telemetri Aktarımı)
MQTT-SN	:	Message Queuing Telemetry Transport for Sensor Networks (Sensör Ağları için Mesaj Kuyruğu Telemetri Aktarımı)
OMG	:	Object Management Group, Inc. (Nesne Yönetim Grubu A.Ş.)
OSI	:	Open Systems Interconnection (Açık Sistem Arabağlantısı)
OPC UA	:	Open Platform Communications United Architecture (Açık Platform İletişim Birleşik Mimarisi)
QoS	:	Quality of Service (Hizmet Kalitesi)
REST	:	Representational State Transfer (Temsili Durum Transferi)
ROS2	:	Robot Operating System 2 (Robot İşletim Sistemi 2)
RPC	:	Remote Procedure Call (Uzaktan Yordam Çağrısı)
RTPS	:	Real-Time Publish Subscribe (Gerçek Zamanlı Yayım Yapma Abone Olma)
SADF	:	Scenario Aware Data Flow (Senaryoya Duyarlı Veri Akışı)
SATCOM	:	Satellite Communications (Uydu İletişimi)
TCP	:	Transmission Control Protocol (Geçiş Kontrol Protokolü)
UDP	:	User Datagram Protocol (Kullanıcı Veribloğu Protokolü)
Wi-Fi	:	Wireless Fidelity (Kablosuz Bağlantı Alanı)
XMPP	:	Extensible Messaging and Presence Protocol (Genişletilebilir Mesajlaşma ve Varlık Protokolü)

NESNELERİN İNTERNETİ TEKNOLOJİLERİNDE FARKLI HABERLEŞME SENARYOLARININ DDS-XRCE STANDARDI İÇİN PERFORMANSININ BELİRLENMESİ

ÖZET

Teknolojik ürünlerin sayısının artışı Nesnelerin İnterneti (Internet of Things, IoT) uygulamaları için birçok çözüm getirmesine rağmen, ayrıca bazı dezavantajlara sebep olmaktadır. Örneğin, söz konusu ürünün Veri Dağıtım Hizmeti (Data Distribution Service, DDS) gibi yüksek performansla işleyecek şekilde yapılandırılmış bir sistemle fiziksel açıdan uyumlu olarak çalışıp çalışmayacağı, ürünler arasında fiziksel açıdan uyum varsa performanslarının neler olacağı gibi. Bu nedenle, söz konusu teknolojik ürünün ilgili uygulama için uyumundan ve yeterliliğinden bahsedebilmek için yeteneklerinin ve yetkinliklerinin tanımlanması gerekmektedir. Bu çalışma, Son Derece Kaynak Kısıtlı Ortamlar İçin Veri Dağıtım Hizmeti (Data Distribution Service for Extremely Resource-Constrained Environments, DDS-XRCE) standardının çalışma mekanizmasını gözlemlerken performansını değerlendirmeyi amaçlamaktadır. Test senaryoları olarak; alıcı ve vericinin türüne, veri paketlerinin takip ettiği yola ve kullanılan protokoller sebebiyle meydana gelen DDS-XRCE'nin üç dağıtım modelinden yararlanılmıştır. Test koşulları akış modları ile taşıma profillerini değiştirerek ve paket gönderimlerini kısıtlayarak ayarlanmıştır. Test ortamı standartların birkaç eProsima uygulama ve kaynak kodu üretim aracını kullanarak DDS ve DDS-XRCE nesnelerinin oluşturulmasıyla elde edilmiştir. Ağ mesajları iki şekilde izlenmiştir. 1) Test esnasında insan gözü ile gözlem yapabilmek için çoklu Gnome Terminator terminalleri kullanarak. 2) Test sonrasında incelemek amacıyla veri paketlerinin bilgilerini kaydetmek için Wireshark kullanarak. Gecikme, verim ve paket kaybına odaklanarak 36 deney gerçekleştirilmiştir. Çalışmamızın sonucu olarak, DDS-XRCE standardının Nesnelerin İnterneti uygulamaları için uygun olduğuna karar verilmiştir.

Anahtar Kelimeler: DDS-XRCE, Gecikme, Paket Kaybı, Performans Değerlendirmesi, Verim.

DDS-XRCE STANDARD PERFORMANCE EVALUATION OF DIFFERENT COMMUNICATION SCENARIOS IN INTERNET OF THINGS (IOT) TECHNOLOGIES

ABSTRACT

Although the increasing number of technological products brings many solutions for Internet of Things (IoT) applications, it also causes some drawbacks, such as whether the product in question would run accordingly to a system structured to enable high-performance like Data Distribution Service (DDS). So, the capabilities of the products must be defined to say that it is compatible enough. This paper aims to evaluate the performance of the DDS-XRCE standard while observing its working mechanism. As test scenarios, we benefit from three DDS-XRCE deployments that occurred due to the kinds of the receiver and sender, the path that packets follow, and the protocols used. Test conditions were set by switching stream modes, transport profiles and limiting packet deliveries. We obtained the test environment by creating the DDS and DDS-XRCE objects using several eProsima implementations and tools of the standards. We monitored the network messages in two ways. 1) Using multiple Gnome Terminator terminals for observation via the human eye during testing. 2) Using Wireshark to save the information of the packets for further examination. We conducted 36 experiments focusing on latency, throughput, and packet loss. As a result of our study, the DDS-XRCE standard is deemed suitable for Internet of Things applications.

Keywords: DDS-XRCE, Latency, Packet Loss, Performance Evaluation, Throughput.

1. GİRİŞ

"Nesnelerin İnterneti" terimi son zamanlarda gündeme gelmektedir ve nesneleri fiziksel ortamda kurulmuş ve İnternet üzerinden bağlanmış sistemler için kullanılmaktadır (Holler ve diğ., 2014). Son on yılda, IoT savunma, otomasyon, sağlık hizmeti gibi alanlarda ün kazanmıştır. Müdahaleyi azaltma konusunda, sensörlere dayalı teknolojisi sebebiyle IoT kayda değer potansiyele sahiptir (Wang ve Payne, 2022). Böylelikle, uzmanların dikkatini daha fazla çekmektedir. Bunu kendisini etkin kılan teknolojisi sayesinde başarmaktadır.

Bir IoT sistemi, cihazları işbirliği ve veri paylaşımı yapan geniş bir ağ olarak tanımlanabilmektedir. Sensörler sistemin veri üreten bileşenleridir ve sistem kısa zamanda çok büyük miktarda veri üretmektedir (Laghari ve diğ., 2021). Aynı zamanda, artan veri onun idaresi veya koruması hakkında endişelerle sonuçlanmaktadır. Bu nedenle, IoT'yi etkin kılan teknolojiler ve disiplinler de IoT sistemlerinin geliştirilmesi amacıyla dikkat çekmektedir.

Örneğin, insan-makine etkileşimini geliştirmek için, bazı araştırmacılar makinelerin duyguları tanıyabilmesini ve kendi duygularını üretebilmesini istemektedir. Sebebi, IoT'nin sadece makine-makine etkileşiminden meydana gelmemesidir. IoT insan-makine etkileşimi de içermektedir (Sheng Huang ve diğ., 2021). Bazı araştırmacılar, bulut-sunucu ortamında işleyen bir video yayın servisini değerlendirmek için bir platform oluşturarak bulut bilişim bağlamında katkıda bulunmuşlardır (Laghari ve diğ., 2022). Bazı araştırmacılar IoT sistemlerinin güvenliğine odaklanmaktadır. Çünkü saldırganlar çoğunlukla ağı çeşitli şekillerde manipüle etmeyi denemektedir, bu da ağın risk altında olmasına neden olmaktadır (Waqas ve diğ., 2021). Böylelikle, eğer makine öğrenmesi algoritmalarının sistemdeki nesnelerin ve cihazların işleyişini öğrenmesine izin verilirse, sonrasında normal olmayan davranışların tespit edilmesiyle bilinmeyen kaynaklar tarafından sebep olunan bozulmaları sistem önleyebilmektedir (Farea ve Küçük, 2022).

Geniş bir ağ olması sebebiyle görülmektedir ki bir IoT sistemi farklı teknolojilerin çeşitli cihazlarını içermektedir. Ancak, cihazları farklı amaçlar taşıyan ve önceliklere sahip heterojen bir ortamda IoT uygulandığında cihazlar arasındaki iletişimi sağlamak güç hale gelmektedir. Haberleşme cihazların türü, uygulamanın amacı ve öncelikleri, işlev alanı,

konum, protokoller, yazılım, bağlantının diğer ucu gibi etkenlere bağlı olarak şekil almaktadır. Çeşitlilik, zorlayıcı konulardaki olası problemler için bir çözüm olarak görünmektedir. Diğer yandan, sonuç olarak nesnelerin uyumlu çalışması zor hale gelmiştir (Bouloukakis ve diğ., 2021). Endüstrinin isteklerinde veri ve gecikmeden kolaylıkla etkilenen IoT uygulamalarına ilişkin bir artış vardır (Kang ve diğ., 2021). Bu, külfetli iletişim sorunlarına katkı sağlamaktadır.

Bir IoT sistemindeki cihazların uyumluluğundan ayrı olarak, nesnelerin veriye duyarlı davranışlarını anlatabilmek de önemli bir konudur. Diğer bir deyişle, işlem türünden bağımsız olarak işleme zamanı, periyodik olarak iletilen verinin miktarı, kayıp verinin miktarı, veri paketlerinin yeniden gönderimleri bilinmelidir. Bu yüzden, örneğin, denilebilmelidir ki nesne düşük gecikmeyle çalışır veya çalışmaz. Belirli bir uygulama için nesnenin beklendiği gibi performans sergileyip sergilemediğine karar verilirken bu tür karakteristikleri bilmek esastır.

IoT yapısı üç katmandan meydana gelmektedir: veri toplanması için algılama cihazlarının kurulduğu algı katmanı, veri iletimi için cihazların ağa erişim sağladığı ağ katmanı, uygulamaların aktif olarak çalıştığı uygulama katmanı (Wang ve Payne, 2022). IoT uygulamaları istek/cevap operasyonlarını uygulama katmanı protokolleri üzerinden işlemektedir. Kısıtlı Uygulama Protokolü (Constrained Application Protocol, CoAP), Mesaj Kuyruğu Telemetri Aktarımı (Message Queue Telemetry Transport, MQTT), Genişletilebilir Mesajlaşma ve Durum Protokolü (Extensible Messaging and Presence Protocol, XMPP), Gelişmiş Mesaj Sıralama Protokolü (Advanced Message Queuing Protocol, AMQP) ve Nesne Yönetim Grubu A.Ş. (Object Management Group Inc., OMG) tarafından benimsenen DDS en çok bilinen uygulama katmanı protokolleridir (Al-Fuqaha ve diğ., 2015), (Schlesselman ve diğ., 2004). Çeşitli protokoller var olmasına rağmen haberleşme standartları geniş bir talep aralığı, ortam koşulları, kısıtlı cihazlar, sınırlı kaynaklar, bütçe ve teknolojik kısıtlar sebebiyle bütün ihtiyaçları karşılamamaktadır. Kullanımları, IoT'de kullanıcıların iyice düşünmesine neden olan zorlu bir konudur (Bayılmış ve diğ., 2022).

Nesne Yönetim Grubu A.Ş., 2020 yılında DDS-XRCE standardını bu kısıtlamalara ve sınırlara çözüm olarak yayınlamıştır. eProsima tarafından geliştirilen DDS-XRCE'nin hem temsilcisi hem de alıcısının uygulamalarının bulunduğu Micro-XRCE-DDS, sadece

temsilcinin uygulamasının bulunduğu Micro-XRCE-DDS-Agent ve sadece alıcının uygulamasının bulunduğu Micro-XRCE-DDS-Client kütüphaneleri vardır (URL-1). “micro-ROS”, Robot İşletim Sistemi 2’nin (Robot Operating System 2, ROS2) bir sürümüdür ve mikrokontrolörler üzerinde çalıştırılır. DDS-XRCE’nin eProsima tarafından sağlanan bir uygulamasını içermektedir (Phueakthong ve Varagul, 2021). DDS-XRCE kullanan uygulamalar görmek mümkündür.

Buna rağmen, DDS-XRCE standardı ile ilgilenen çalışma eksikliği vardır. Bu sebeple, DDS-XRCE’nin ağ performansı karakteristikleri belirsizliğini sürdürmeye devam etmektedir. Bu da DDS-XRCE’nin bir sistemde uyumlu olarak işleyeceği veya işlemeyeceğine nasıl karar verileceği konusunda karışıklığa sebep olmaktadır. Bu nedenle, bu çalışmanın çözmeyi amaçladığı problem DDS-XRCE’nin bilinmeyen performans karakteristiklerinin o şekilde kalmaya devam etmesidir. Bu çalışmada, eProsima tarafından sağlanan uygulaması üzerinden DDS-XRCE’nin performansı değerlendirilmiştir. Bu çalışmanın katkıları şu şekildedir:

- DDS standardının, nesnelerinin ve işleyiş mekanizmasının analizi
- DDS-XRCE standardının, nesnelerinin ve işleyiş mekanizmasının analizi
- DDS ve DDS-XRCE’nin bütünleşmesinin analizi
- Performans değerlendirme deneyleri ve DDS-XRCE standardının değerlendirilmesinin sonuçları

2. LİTERATÜR ÖZETİ

Dehnavi ve arkadaşları, DDS-XRCE'nin bir uygulamasını modellemişlerdir ve modeli çok işlemcili gerçek zamanlı gömülü sistemlerde çalıştırmışlardır. Ek olarak, sistemler üzerinde DDS-XRCE Agent için zaman sınırının aşılmasına periyodik olarak izin veren gerçek zamanlı bazı deneyler ve DDS-XRCE Client için herhangi bir zaman sınırının aşılmasını sistem hatası varsayan gerçek zamanlı bazı deneyler gerçekleştirilmiştir. Yazarların uzun vadedeki verimin beklenen değerini analiz etmek için önerdiği Senaryoya Duyarlı Veri Akışı (Scenario Aware Data Flow, SADF) modeli kullanılarak yayımcı ve abonenin en kötü durum yanıt zamanı ölçülmüştür (Dehnavi ve diğ., 2021).

Kang ve arkadaşları, veri ve gecikmeye duyarlı IoT uygulamalarının yayım/abone olma yardımcı programlarının uç ve bulut uygulamalarına sebep olacağı problemler konusunda endişe duymaktadır. Bu problemlere bir çözüm getirebilmek için bulutta konteynıra alınmış uygulamaları yöneten Kubernetes (K8s) ve DDS'ten yararlanmışlardır. Değerlendirme, DDS uygulamalarının K8s kümelerinde çalıştırılmasıyla sağlanmıştır ve farklı Hizmet Kalitesi (Quality of Service, QoS) politikaları etkin olduğunda K8s'lerin DDS performansı üzerindeki etkisi verim ve gecikmeye odaklanarak gözlenmiştir (Kang ve diğ., 2021).

Chul-Hwan Kim ve arkadaşları, DDS'nin performansını değerlendirmek için bir benzetim aracı geliştirmişlerdir. Simülasyonun geliştirilmesi, birkaç ağ protokolünün kullanımına olanak sağlayan QualNet olarak adlandırılan benzetim platformunda gerçekleştirilmiştir. Yazarların odaklandığı performans metrikleri keşif tamamlanma zamanı, mesaj iletim gecikmesi, veri mesajlarının miktarı ve kullanıcı verisinin işlenmesi için geçen zamandır (Chul-Hwan Kim ve diğ., 2015).

Krinkin ve arkadaşları, DDS'nin performansını değerlendirmiştir. Prismtech tarafından sağlanan OpenDDS, Vortex tarafından sağlanan OpenSplice ve eProsima tarafından sağlanan FastRTPS gibi DDS'nin açık kaynaklı farklı uygulamalarının karşılaştırılmasıyla gecikme ve "jitter"a odaklanılarak deneyler yürütülmüştür (Krinkin ve diğ., 2018).

Andrei ve arkadaşları, DDS ve AMQP'nin performansını değerlendirmişlerdir. Değerlendirme, DDS'nin OpenSplice ile FastRTPS uygulamalarının ve AMQP'nin RabbitMQ uygulamasının kullanılmasıyla yapılmıştır. Gecikme, mesaj kuyruğunun soruşturması ve verim en yüksek seviyesine ulaştığında mesajların boyutlarının ve frekanslarının ne olduğuna odaklanılmıştır (Andrei ve diğ., 2020).

Benzer olarak Profanter ve arkadaşları, OPC UA'nın open62541, ROS'un ROS C++, DDS'nin eProsima FastRTPS ve MQTT'nin Eclipse Paho MQTT C uygulamalarını karşılaştırılarak değerlendirmişlerdir (Profanter ve diğ., 2019). Değerlendirme; sistem boşta olduğunda, yüksek CPU yükü ve yüksek ağ yükü koşullarındayken mesajların gidiş-dönüş süresinin ölçülmesiyle gerçekleştirilmiştir.

Chen ve Khun, bir ağ değerlendiricisi kullanarak MQTT, CoAP, DDS ve UDP'ye dayanan özel bir protokolün performanslarını tıbbi amaçlarla değerlendirmişlerdir (Chen ve Kunz, 2016). Sistemin tükettiği bant genişliği, gecikme ve paket kaybına odaklanmışlardır.

Ağ uygulamalarının ve IoT protokollerinin ağ performansları iki test uygulaması üzerinde deney yaparak Babovic ve arkadaşları tarafından değerlendirilmiştir. İlk uygulamada, çeşitli metrikler konusunda ağ platform uygulamaları değerlendirilmiştir. İkinci uygulamada MQTT, AMQP, XMPP ve DDS IoT protokolleri gecikme ve verime odaklanılarak değerlendirilmiştir (Babovic ve diğ., 2016).

MQTT, mesajlarının yayımcı ve abone arasında MQTT aracısı tarafından aracılık edilen bir protokoldür. Mesajlar aracı üzerinden geçtiğinden, aracı tıkanıklığın ağda yaşanabileceği en muhtemel yer olarak düşünülmüştür. Buna dayanaraktan, aracının performansı MQTT'nin performansını temsil etmektedir. MQTT 5.0 versiyonunun performansını ve yeni işlevlerini, Banno ve arkadaşlarının yük testi yapmak için geliştirdiği MQTTLoader'ı kullanarak onun aracısı üzerinden değerlendirmişlerdir (Banno ve diğ., 2021).

Thulasiraman ve arkadaşları, belirli bir senaryoda verim ve gecikmeye odaklanarak DDS'nin performansını değerlendirmişlerdir. Nedeni, ABD Donanması otonom sistemlerinin bütün ağ varlıkları ile çalışabilecek bir haberleşme protokolü arayışı içinde olmasıdır ve DDS birincil aday olarak düşünülmüştür. Uydu haberleşmeleri (Satellite

Communications, SATCOM) ve Wi-Fi bağlantıları içeren bir deney sistemi modellemişlerdir. Ayrıca, ağ değerlendirmesi ve ağ parametreleri işlemleri sırasında Mininet kullanılmıştır (Thulasiraman ve diğ., 2022).

Chen ve arkadaşlarının amacı DDS'nin gerçek zaman temellerine göre çalışıp çalışmayacağını kontrol etmektir. PREEMPT_RT Linux sistemi ve Loongson platformu üzerinde gecikme, jitter ve veri verimi hususlarında DDS'nin performansının değerlendirildiği bir çalışmadır. Sonuç olarak, DDS'nin performansı ve ağ kartı arasında bir ilişki vardır (Chen ve diğ., 2021).

MQTT ve CoAP protokolleri teorik ve pratik olarak Palmese ve arkadaşları tarafından değerlendirilmiştir. MQTT'nin bir başka formu, MQTT-SN, yayımla/abone ol haberleşme şemasına göre çalışmaktadır. Bu yüzden, adil şekilde karşılaştırabilmek için MQTT-SN gibi aynı haberleşme şemasını takip etmesi amacıyla CoAP'de bazı değişiklikler yaplmıştır. İki protokol de haberleşmede UDP'ye dayanmaktadır (Palmese ve diğ., 2021).

Sasaki ve arkadaşlarına göre, OSI katmanının diğer protokolleri ve IoT protokolleri arasındaki işbirliği ilginç bir konudur ve değerlendirmeye değerdir. Örneğin, IP (Internet Protocol Address) ve ilgili protokoller MQTT'nin işlevlerini yerine getirir. Ek olarak, MQTT, Hizmet Kalitesi politikalarıyla birlikte gelir. Veri iletimindeki MQTT-TCP işbirliğinin performansı ve MQTT Hizmet Kalitesi mekanizması analiz edilmiştir (Sasaki ve diğ., 2018).

Basavaraju ve arkadaşları; gecikme, veri oranı, farklı mesaj yükleri ile mesajların sayılarına odaklanarak RabbitMQ ve ActiveMQ mesaj aracılarının karşılaştırmasını yaparak AMQP protokolünü değerlendirmişlerdir. Mesaj aracılarından biri AMQP 0-9-1 sürümünü ve diğeri AMQP 1-0 sürümünü uygulamıştır (Basavaraju ve diğ., 2021).

E-sağlık, IoT'nin yüksek önem vererek ilgilendiği bir alandır ve e-sağlık uygulamalarının performansı özellikle düşünülmüştür. Bu yüzden, Kassem ve Sleit e-sağlık senaryoları üzerinden CoAP ve MQTT protokollerini incelemişlerdir ve yazarların metrik olarak seçtiği geçmiş zamanda karşılaştırmalı olarak performanslarını değerlendirmişlerdir (Kassem ve Sleit, 2020).

Bender ve arkadaşları, kendilerinin oluşturduğu bir test sistemi kullanarak MQTT'nin performansını onun birkaç açık kaynaklı uygulamaları üzerinden birlikte çalışabilirlik, kaynak tüketimi ve gecikmeye odaklanarak test etmişlerdir (Bender ve diğ., 2021). Bu uygulamalar Mosquitto, HiveMQ, EMQX, VerneMQ, MQTT.js ve Paho'dur. Kullanılan test sistemi ise MQTT uygulamasından veya programlama dilinden bağımsız olarak çalışabilmektedir.

Uygulama katmanının protokolleri, IoT uygulamalarında ağ trafiğinin azaltılmasında kritik öneme sahiptir. Protokollerin dikkatli seçimi ağ trafiğinin yükünü azaltabilir ve başarılı mesaj iletimini arttırabilir. Bu yüzden, Tandale ve arkadaşlarının Raspberry Pi3 üzerinde geçit olarak uyguladıkları ve operasyonların harcadığı zaman ile protokollerin tükettiği bant genişliğine odaklanarak değerlendirdikleri CoAP, MQTT ve REST protokollerinin performansını tartışmışlardır (Tandale ve diğ., 2018).

Pohl ve arkadaşları, bir iş uygulamasında performans metriği olarak bant genişliği kullanımı, güvenilebilirlik, gecikme ve verime odaklanarak AMQP, MQTT ve XMPP protokollerini değerlendirmiştir. Tasarladıkları test sistemi değişebilir gecikme ve paket kayıp oranıyla birlikte üç katmana sahiptir (Pohl ve diğ., 2018).

Önceki çalışmalar değerlendirmelerini çoğunlukla farklı protokolleri karşılaştırarak veya tek bir protokole odaklanarak gerçekleştirmiştir. Sadece bir protokole odaklanan çalışmalar, protokolün birkaç uygulamasını veya özelliğini kullanarak değerlendirmişlerdir. İki protokolü, DDS ve DDS-XRCE, incelemesine rağmen, bu çalışma bir protokole odaklanan bir çalışma olarak sınıflandırılabilmektedir. DDS-XRCE, DDS ile bütünleşerek işleyebilmektedir. Böylelikle, DDS-XRCE'nin kullanımı DDS'nin kullanımını bazı yönlerde zorunlu kılmaktadır. Bu, aynı zamanda bu çalışmayı önceki çalışmalardan ayıran özelliktir. Protokoller arası gerekli bir bütünleşme nadiren meydana gelmektedir. Değerlendirme adımlarında DDS'nin yer almasına rağmen, çalışmanın ana odağı DDS-XRCE üzerindedir. Sonuç olarak DDS-XRCE bu çalışmanın konusunu oluşturmaktadır.

Değerlendirme, DDS-XRCE birkaç yapılandırma altında çalışırken devam etmiştir. Asıl amaç bir protokolün performansını değerlendirmek olduğundan, birçok değerlendirme ölçütü vardır. Bu çalışmada da olduğu gibi, bu ölçütler çoğunlukla gecikme, veri paketi

kaybı ve verim olmaktadır. Test sistemi, probleme göre olduğu kadar değerlendirme ölçütlerine göre de şekil almaktadır. Bazı çalışmaların yazarlar tarafından geliştirilen veya tasarlanan oluşumlar içerdiğini görmek mümkündür. Bu oluşumlar benzetim araçları veya uygulamalar olarak değişmektedir. Bu çalışmadaki oluşumlar kod üretim araçları kullanılarak oluşturulan farklı konular ve alıcılardır. Ek olarak alıcılar, bu çalışmanın ağını oluşturmaktadır.

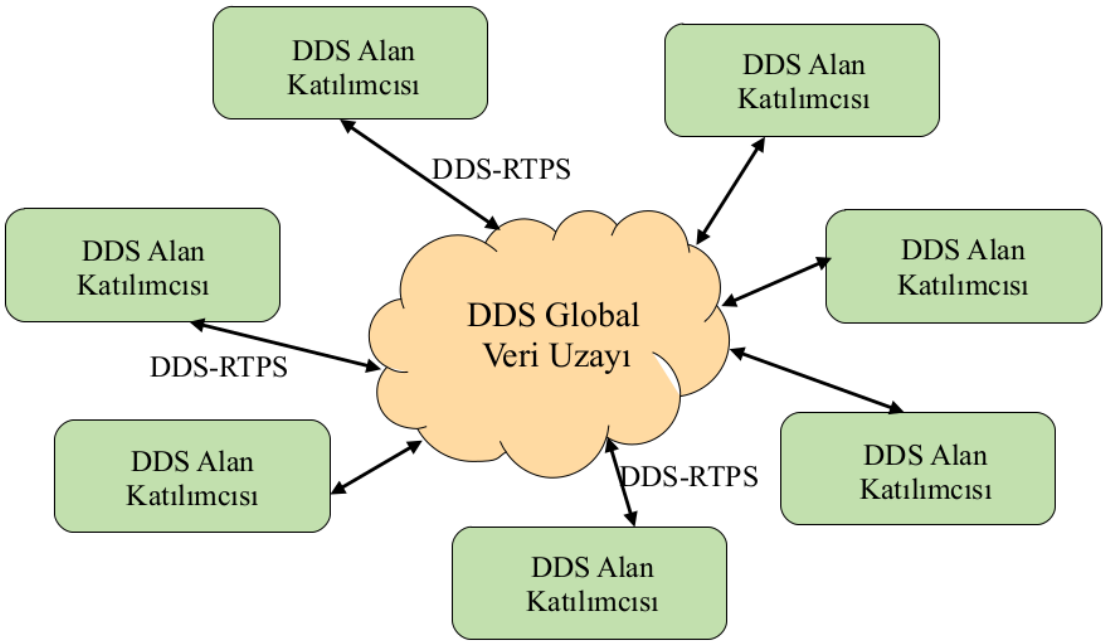
3. KULLANILAN MATERYALLER

3.1. Veri Dağıtım Hizmeti (Data Distribution Service, DDS)

2006 yılında yayınlandığı için günümüzde yaygın olarak bilinen DDS, Nesne Yönetim Grubu A.Ş. tarafından sağlanan etkili bilgi teslimatı ile yüksek performanslı iletişim standardıdır. Nesne Yönetim Grubu A.Ş. ise, bilgisayar endüstrisinde aktif olan bir standartlar kuruluşudur (URL-2). Robot İşletim Sistemi 2 (Robot Operating System 2, ROS2), DDS'yi kullanan bir uygulamadır. DDS'yi yapılandırmaları sebebiyle gerçek zamanlı gömülü sistemler için daha uygun olduğu için kullanmaktadır (Maruyama ve diğ., 2016).

3.1.1. DDS'nin Yapısı

DDS fiziksel olarak iki kısımdan oluşmaktadır: DDS Global Veri Uzayı (DDS Global Data Space) ve DDS katılımcıları. DDS katılımcıları, birbirleriyle sadece DDS Global Veri Uzayı üzerinden bağlantı kurarak iletişim sağlarlar. Her katılımcının arasında DDS Global Veri Uzayı yer aldığı için DDS'nin fiziksel yapısı Şekil 3.1'de gösterildiği gibi yıldız topolojisine benzemektedir.



Şekil 3.1. DDS'nin yapısı

3.1.2. DDS İle İlgili Nesneler ve Kavramlar

Daha iyi bir anlayabilmek için DDS'de yer alan ve yaygın olarak kullanılan nesneler ve kavramlar açıklanmıştır.

Yayımcı (Publisher), veriyi ulaşması gereken uç noktaya gönderen nesnedir. Veri yazarı (DataWriter), uygulamalar tarafından yayımcıyı veri ve verinin bilgisi hakkında bilgilendirmek için kullanılan bir nesnedir. Yayım, yayımcı ve veri yazarı arasındaki ilişkidir. Abone (Subscriber), bir yayımcıdan gönderilen mesajları alan bir nesnedir. Veri okuyucusu (DataReader), veri alan uygulamalar tarafından aboneye iliştirilerek kullanılan bir nesnedir. Abonelik, abone ve veri okuyucusu arasındaki bir ilişkidir.

Konu (Topic), yayım ve abonelik arasındaki bir kavramdır. Hizmet Kalitesi (Quality of Service, QoS), sistemin bazı eylemlerini yöneten ayarlanabilir özelliklerin bir politika listesidir. Her Hizmet Kalitesi politikası özel bir veya birkaç varlıkla ilgilenmektedir. Alan (Domain), alan üyeleri arasındaki onların birbirleriyle iletişim kurmalarına yardımcı olan bir dizi kavramsal bağlantılardır. Alan katılımcısı (Domain Participant), bir alana üye olan bir uygulamadır. Bir uygulama birden fazla alanın üyesi olabilmektedir.

3.1.3. Veri Merkezli Yayım Yapma Abone Olma (Data-Centric Publish-Subscribe, DCPS)

DDS varlıklar, operasyonlar ve işleyiş mekanizması açısından incelenmiştir. DDS beş modülden oluşan bir nesne modeli olan Veri Merkezli Yayım Yapma Abone Olma'yı (Data-Centric Publish-Subscribe, DCPS) tanımlamaktadır. DCPS, nesnelerini veya ara yüzlerini modüllerle sınıflandırmaktadır. Örneğin, dinleyici ara yüzleri Altyapı Modülü'ne aittir.

Altyapı Modülü (The Infrastructure Module): Varlık (Entity), Alan Varlığı (DomainEntity), Hizmet Kalitesi Politikası (QoSPolicy), Dinleyici (Listener), Durum (Status), Bekleme Takımı (WaitSet), Koşul (Condition), Koruma Koşulu (GuardCondition) ve Durum Koşulu (StatusCondition) sınıflarını ve ara yüzlerini içerir. Ara katman yazılımına bildirim ve bekleme tabanlı etkileşimler sağlanmasında yardımcı olmaktadır. Altyapı Modülü'nün sınıfları ve ara yüzleri tamamen soyuttur ve diğer modüller tarafından işlenmektedir.

Alan Modülü (The Domain Module): Birçok sınıf için bir fabrika gibi çalışmaktadır ve aynı zamanda Alan Katılımcısı'nın (DomainParticipant) ait olduğu modüldür. Alan Katılımcısı'na ek olarak, Alan Katılımcısı Fabrikası (DomainParticipantFactory) sınıfı ve Alan Katlımcısı Dinleyicisi (DomainParticipantListener) ara yüzünü içerir.

Konu Tanımlama Modülü (The Topic-Definition Module): Konu Tanımı (TopicDescripton), Konu (Topic), İçerik Filtrelenmiş Konu (ContentFilteredTopic), Çoklu Konu (MultiTopic), Konu Dinleyicisi (TopicListener), Tip Desteği (TypeSupport) sınıfları ve ara yüzleri içerir. Bunlar, konu oluşturma işlemlerinde kullanılacak ve aynı zamanda konuların Hizmet Kalitesi politikalarının iliştirildiği şeylerdir.

Yayım Yapma Modülü (The Publication Module): Yayımlama süreci için kullanılacak ve yardımcı olacak sınıfları ve ara yüzleri içermektedir. Bunlar Yayımcı (Publisher), Veri Yazarı (DataWriter), Yayımcı Dinleyicisi (PublisherListener) ve Veri Yazarı Dinleyicisi'dir (DataWriterListener).

Abonelik Modülü (The Subscription Module): Abonelik süreci için kullanılacak ve yardımcı olacak sınıfları ve ara yüzleri içermektedir. Bunlar Abone (Subscriber), Veri Okuyucusu (DataReader), Veri Örneği (DataSample), Örnek Bilgisi (SampleInfo), Abone Dinleyicisi (SubscriberListener), Veri Okuyucusu Dinleyicisi (DataReaderListener), Okuma Koşulu (ReadCondition) ve Sorgu Koşulu'dur (QueryCondition).

3.1.4. DDS Varlıkları ve Ara Yüzleri Oluşumu

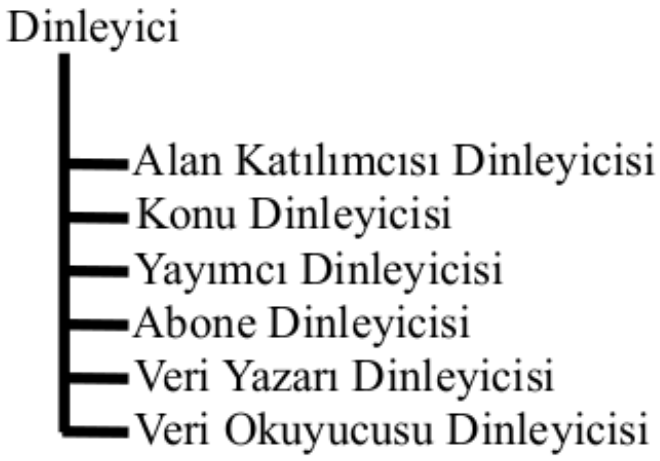
Alan Katılımcısı varlığı Alan Katılımcısı Fabrikası tarafından üretilmektedir. Alan Katılımcısı ise Yayımcı, Abone, Konu ve Çoklu Konu varlıklarını üretir, çünkü bu varlıklar için fabrika gibi çalışmaktadır. Veri Yazarı Yayımcı tarafından ve Veri Okuyucusu Abone tarafından üretilmektedir.

Hangi varlığın ne tarafından üretildiğine dayanarak söylenebilir ki üretici ürettiği şey için fabrika olarak çalışmaktadır. Şekil 3.2'de oluşum ağacı resmedilmiştir. Ağaçta yer alan nesnelerden Çoklu Konu nesneleri hariç hepsi Varlık (Entity) sınıfına aittir.



Şekil 3.2. DDS varlıklarının oluşum ağacı

Varlıkların ağ trafiğini izlemesine yardımcı olan Dinleyici ara yüzü DDS'ye özeldir ve DDS-XRCE'de bulunmamaktadır. Dinleyici ara yüzü Altyapı Modülü'nde yer alır ve Altyapı Modülü'nün Dinleyici sınıfı diğer Dinleyici ara yüzleri için soyut bir kaynak (kök) olarak çalışmaktadır. Her Alan Katılımcısı Dinleyicisi, Konu Dinleyicisi, Yayımcı Dinleyicisi, Abone Dinleyicisi, Veri Yazarı Dinleyicisi ve Veri Okuyucusu Dinleyicisi kaynak Dinleyici'den türetilmiştir ve sonrasında ilgili varlığa iliştirilmektedir. Örneğin, oluşum işleminden sonra Veri Okuyucusu Dinleyicisi ilgili Veri Okuyucusu'na iliştirilir. Dinleyici oluşum ağacı Şekil 3.3'te resmedilmiştir.



Şekil 3.3. Dinleyici ara yüzlerinin oluşum ağacı

Varlık veya dinleyici ara yüzü olarak kabul edilmemesine rağmen, DDS sistemine ve operasyonlarına yardımcı olan diğer sınıfların nesneleri de vardır. Dahası, bazılarının fabrikası yoktur ve doğrudan oluşturulmaktadır. Örneğin, söz konusu olan bir uygulama vardır. Bekleme Takımı (WaitSet) nesnesi, o uygulamaya iliştirilmiş bazı Koşul (Condition) nesneleri gerekli şartları sağlayana kadar uygulamanın işlemlerini ertelemektedir.

3.1.5. DDS Mesajları ve Yapısı

DDS Katılımcıları birbirleriyle haberleşmek için Gerçek Zamanlı Yayım Yap-Abone Ol (Real-Time Publish-Subscribe, RTPS) protokolünü kullanmaktadır. RTPS, koordineli bir şekilde haberleşmeleri amacıyla DDS katılımcıları, nesneler ve cihazlar için bir kablolu protokoldür (URL-2). Şekil 3.4, RTPS protokolünün mesaj yapısını temsil etmektedir.



Şekil 3.4. RTPS protokolünün mesaj yapısı

RTPS protokolünün mesaj yapısı iki kısımdan oluşmaktadır: Başlık ve alt mesaj. Her RTPS mesajı başlangıç olarak başlık kısmını içermek zorundadır. Başlık; protokol, protokol versiyonu, sağlayıcı kimliği (vendorId) ve daha sonraları yeniden oluşturma eylemi için kullanılan bir önek olan guidPrefix hakkında bilgiler taşımaktadır. Bütün mesajlar alt mesaj kısımlarına sahiptir ve alt mesajların miktarı, mesajlara bağlı olarak değişmektedir.

Şekil 3.4'te mor dikdörtgen çerçeve şeklinde gösterilmiş alt mesaj da iki kısımdan oluşmaktadır: Alt mesaj başlığı ve alt mesaj öğesi. Alt mesaj başlığı, bir alt mesajın kesinlikle sahip olması gereken kısımdır ve alt mesaj kimliği, etiketler ve alt mesaj uzunluğu hakkında bilgiler içermektedir. Alt mesaj öğeleri, sistemin alt mesajları oluşturmak için kullandığı yapı taşlarıdır. Önceden tanımlıdır.

RTPS protokolü 2.2 sürümü birkaç çeşit alt mesaj tanımlamaktadır ve alt mesajlar iki grup halinde sınıflandırılmaktadır. Bunlar, RTPS varlıklarını hedefleyen varlık alt mesajları ile RTPS alıcısı durumunu değiştiren ve sonraki varlık alt mesajlarının işlenmesine yardımcı olan bağlamı sağlayan yorumlayıcı alt mesajlarıdır. Varlık alt mesajları çeşitleri Tablo 3.1'de listelenmiştir.

Tablo 3.1. RTPS varlık alt mesajı tipleri

Varlık Alt Mesajı Tipleri	
Tip	**Açıklama**
Data	Bir uygulama Date nesnesinin değeriyle ilgili bilgiler içermektedir. Yazıcılar tarafından Okuyuculara gönderilmektedir.
DataFrag	Data'ya eşdeğerdir, ancak yeni değerin sadece bir bölümü içermektedir. Mesaj taşımadaki mesaj boyutu kısıtlamalarının üstesinden gelmek için verinin çoklu parçalar halinde iletilmesine olanak sağlamaktadır.
Heartbeat	Bir yazıcıdaki ulaşılabilir olan bilgiyi ifade etmektedir. Bir yazıcı tarafından bir veya daha fazla okuyucuya gönderilmektedir.
HeartbeatFrag	Parçalanmış veri için bir yazıcıda hangi parçaların ulaşılabilir olduğunu ifade etmektedir. Bir yazıcıdan bir veya daha fazla okuyucuya gönderilmektedir.
Gap	Okuyucularla artık ilgili olamayan bilgiyi ifade etmektedir. Bir yazıcı tarafından bir veya daha fazla okuyucuya gönderilmektedir.
AckNack	Bir okuyucunun durumu hakkında bir yazıcıya bilgi sağlamaktadır. Bir okuyucu tarafından bir ya da daha fazla yazıcıya gönderilmektedir.
NackFrag	Bir okuyucunun durumu hakkında bir yazıcıya bilgi sağlamaktadır. Daha detaylı olarak, okuyucuda hangi parçaların hala kayıp olduğu hakkında bilgi sağlamaktadır. Bir okuyucu tarafından bir ya da daha fazla yazıcıya gönderilmektedir.

Yorumlayıcı alt mesajları ise Tablo 3.2'de listelenmiştir.

Tablo 3.2. RTPS yorumlayıcı alt mesaj tipleri

Yorumlayıcı Alt Mesaj Tipleri	
Tip	**Açıklama**
InfoSource	Bir sonraki varlık alt mesajının meydana geldiği kaynak hakkında bilgi sağlamaktadır.
InfoDestination	Bir sonraki varlık alt mesajının son varış yeri hakkında bilgi sağlamaktadır.
InfoReply	Sonraki alt mesajlarda görünen varlıklara nereden cevap verileceği hakkında bilgi sağlamaktadır.
InfoTimestamp	Sonraki varlık alt mesajları için bir kaynak zaman damgası sağlamaktadır.
Pad	Eğer gerekirse hafıza hizalanması için bir mesaja dolgu eklemek için kullanılmaktadır.

3.2. Son Derece Kaynak Kısıtlı Ortamlar İçin Veri Dağıtım Hizmeti (Data Distribution Service for Extremely Resource-Constrained Environments, DDS-XRCE)

DDS-XRCE, DDS'nin bir dalı olan, alıcı-sunucu düzenini takip eden kablolu bir protokoldür. DDS-XRCE, temel olarak kaynak kısıtlı cihazların DDS Global Veri Uzayı'na erişim sağlaması amacıyla benimsenmiştir (URL-2).

3.2.1. DDS-XRCE İle İlgili Kavramlar

DDS ile ilgili kavramlara ek olarak DDS-XRCE'de fazladan iki kavram daha yer almaktadır: Temsilci (Agent) ve XRCE alıcısı (XRCE Client).

XRCE alıcısı abonelik, yayım yapma, kaynak yönetimi vb. talepler ile temsilciden yararlanmaktadır. Örneğin, XRCE alıcıları kaynak kısıtlı cihazlar oldukları için periyodik olarak uyuma ve uyanma döngüsündedir. Bir XRCE alıcısı uyuma döngüsündeyken, temsilci ilgili alıcının mesajlarını depolamakta ve alıcı uyandığında depolanan mesajları kendisine iletmektedir.

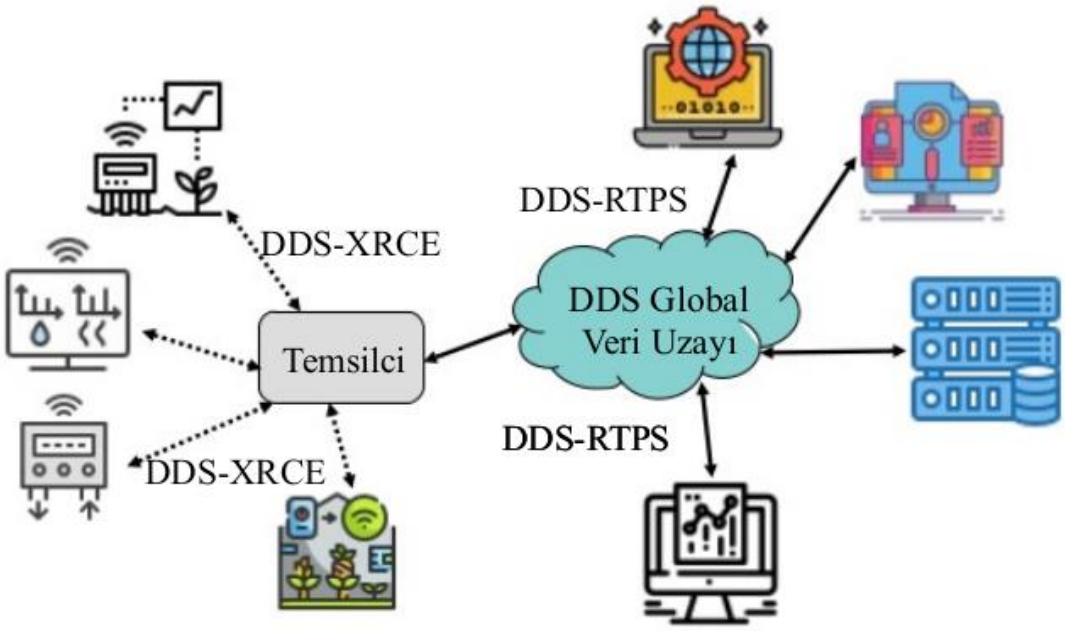
Temsilci, DDS-XRCE'de sunucu ve DDS'de katılımcı olarak görev yapmaktadır. Temsilci, XRCE alıcıları ve DDS katılımcıları arasındaki iletişimi köprü görevi görerek sağlamaktadır. DDS Global Veri Uzayı üzerinden bağlantı kurmak DDS-XRCE'de sadece temsilcinin yeteneğidir.

Ek olarak; DDS Global Veri Uzayı'nda bir katılımcının açısından gözlem yapıldığında, DDS Global Veri Uzayı üzerinden bağlantı kurulan katılımcılar için "DDS katılımcısı" veya "DDS-XRCE'nin temsilcisi" şeklinde herhangi bir ayrım yapılmamaktadır. DDS Global Veri Uzayı üzerinden bağlantı kurulan her katılımcı sadece DDS katılımcısı olarak görülmektedir. Temsilci bunu XRCE alıcıları için DDS Global Veri Uzayı'nda vekil DDS katılımcıları oluşturarak başarmaktadır.

Kaynak dağıtımı, veri iletimi sırasında protokoller arası dönüşüm, parametreler ve modların ayarlanması, DDS-XRCE modeli içinde iletişimin sağlanması temsilcinin diğer görevleri ve davranışlarındandır.

3.2.2. DDS ve DDS-XRCE'nin Bütünleşmesi

DDS Global Veri Uzayı'nda, üyeler diğer üyeleri DDS modelinden bir üye veya DDS-XRCE modelinden bir üye şeklinde sınıflandırmamaktadır. DDS-XRCE'nin temsilcisi DDS Global Veri Uzayı'ndaki diğer üyelerle haberleşmesine rağmen, her üye sadece DDS katılımcısı olarak görünmektedir. Şekil 3.5 DDS ve DDS-XRCE'nin bütünleşmesini temsil etmektedir.



Şekil 3.5. DDS ve DDS-XRCE'nin bütünleşmesi

Bu bütünleşme, uyumluluk sorunlarına çözüm olarak basitçe "tak ve çalıştır" şeklinde ifade edilmektedir. DDS'de, katılımcılar bir alanın konu mesajlarına ulaşmak için o alana dahil olmak zorundadır. Temsilci, DDS Global Veri Uzayı'nda vekil bir DDS alan katılımcısı oluşturarak bağlantı kurmakta ve iletişimi sürdürmektedir.

DDS alan katılımcıları ağ trafiğini sürekli olarak izlemektedir. Sürekli ağı izlemek veya mesajları işlemek kayda değer miktarda kaynak gerektirmektedir. Buna göre DDS'nin ilgilendiği cihazlar son derece donanımlı ve gelişmiştir. Ancak DDS-XRCE'nin açılımı Son Derece Kaynak Kısıtlı Cihazlar İçin Veri Dağıtım Hizmeti olduğu üzere DDS-XRCE'nin ilgilendiği cihazlar kaynak kısıtlıdır. Dahası, XRCE alıcıları periyodik olarak uyumakta ve uyanmaktadır. XRCE alıcıları uyku döngüsündeyken, bir temsilci onlar için işlemektedir. Örneğin, onların uyanma döngüsü sırasında iletmek için konu mesajlarını

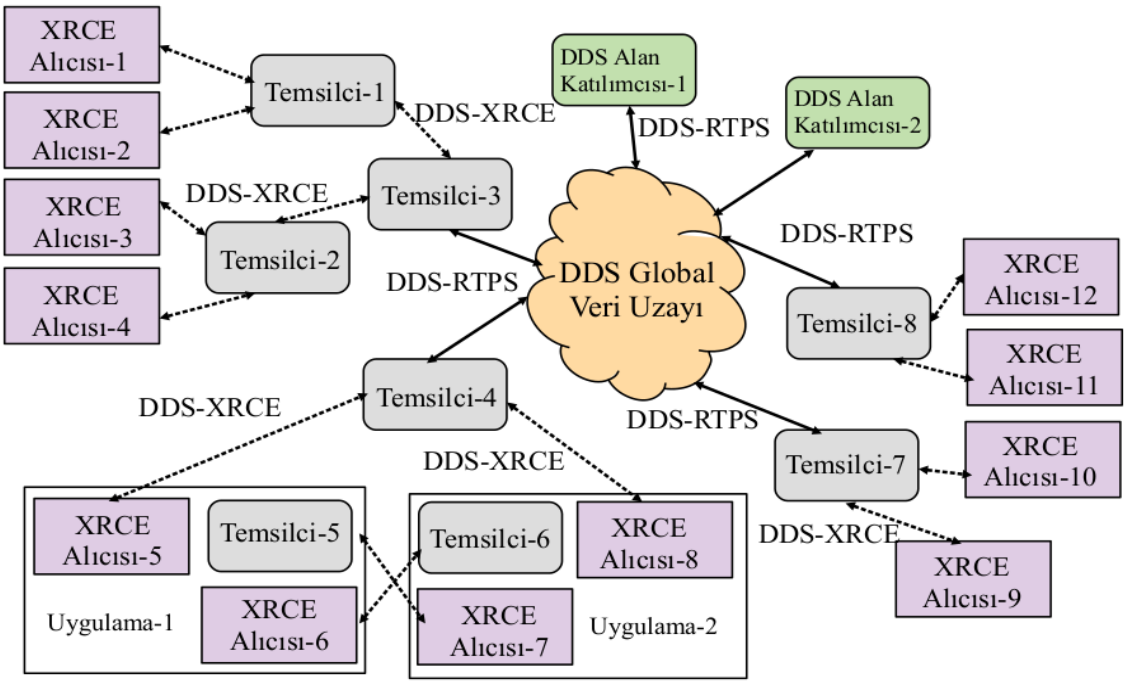
XRCE alıcıları için depolamaktadır. DDS'nin yirmi iki tanesini desteklediği Hizmet Kalitesi politikaları, standartlar için ortaktır. DDS-XRCE tarafından da desteklenmektedir. DDS-XRCE genel olarak DDS ile çalışmaktadır. Dağıtım modelleri incelendiğinde, DDS-XRCE'nin uygulanmak için bazı durumlarda DDS'ye ihtiyaç duyduğunu söylemek bile mümkündür. Bu nedenle, DDS-XRCE DDS'ye çoğunlukla bağımlıdır ve DDS ile uyumlu bir şekilde çalışabilme yetisine sahip olmak zorundadır.

Ek olarak DDS-XRCE, XRCE varlıkları için Hizmet Kalitesi politikalarının konfigürasyonunu da içeren, bir dereceye kadar yapılandırma yetenekleri sağlayan on profil tanımlamaktadır. Bu profillerden bazıları, varlıklara parametreleri ayarlama otoritesini veren gelişmiş yetenekler sağlamaktadır. Örneğin, konuların Hizmet Kalitesi politikasını ayarlayabilmek gibi. Bu tür otoriteye sahip olmak alıcıları gelişmiş yapmaktadır. Böylelikle, DDS-XRCE'nin asıl amacı sıklıkla kullanılan diğer cihazlardan kaynak kısıtlı cihazlara erişim sağlamak olmasına rağmen, DDS-XRCE gelişmiş cihazlar ve/veya alıcılarla da ilgilenmektedir. DDS-XRCE cihazları ve alıcıları basit cihazlar, daha yetkin cihazlar, gelişmiş alıcılar ve karmaşık alıcılar olarak sınıflandırmaktadır. Cihazlar ve alıcıların çeşitliliği XRCE alıcılarının ihtiyaçlarındaki farklılık sebebiyle meydana gelmektedir.

3.2.3. DDS-XRCE'nin Yapısı

DDS modeli yıldız topolojisine benzerken, bu durum DDS-XRCE için daha faklıdır. İletim yolu, gönderici/alıcı türü ve iletim protokolüne bağlı olarak"Dağıtım (Deployment)" adı verilen altı model oluşturulmaktadır. Dağıtım modelleri DDS-XRCE standardının ne çeşitlerde uygulanabileceği sorusunun cevabıdır. Modeller Şekil 3.6'da resmedilmiştir.

Şekil 3.6, DDS-XRCE tanım belgesinde yer alan altıncı dağıtım modeli üzerinde değişiklikler yapılarak oluşturulmuştur. Tanım belgesinde gösterilen modele Temsilci-1, Temsilci-2, Temsilci-7, Temsilci-8, XRCE Alıcısı-3, XRCE Alıcısı-4, XRCE Alıcısı-9, XRCE Alıcısı-10, XRCE Alıcısı-11 ve XRCE Alıcısı-12 eklenmiştir. Ardından orijinal gösterimde XRCE Alıcısı-1, XRCE Alıcısı-2 ve Temsilci-4 ile bağlantılı olan Temsilci-3'ün Temsilci-4 ile olan XRCE protokolü bağlantısı koparılmış ve RTPS protokolü ile DDS Global Veri Uzayı'na doğrudan erişmesi sağlanmıştır.



Şekil 3.6. Dağıtım modelleri

Birbirleriyle haberleşen nesneler ve dağıtım modellerinde odaklanılan iletim yolu Tablo 3.3'te listelenmiştir. İletim yolu sütunundaki ":::" sembolü nesnelerin iletişim kurmak için DDS-XRCE protokolünü kullandığını ifade etmektedir. "=" sembolü ise nesnelerin iletişim kurmak için RTPS protokolü kullandığını ifade etmektedir.

Tablo 3.3. Dağıtım modelleri ve detayları

No	**Dağıtım Modelleri Detayları**	
	Nesneler	**Şekil 3.6'daki İletim Yolu**
1	XRCE Alıcısı ve DDSAK	XRCE Alıcısı-12:::Temsilci-8=DDSAK-1
2	XRCE Alıcıları	XRCE Alıcısı-11:::Temsilci-8=Temsilci-7:::XRCE Alıcısı-10
3	XRCE Alıcıları	XRCE Alıcısı-10:::Temsilci-7=Temsilci-7:::XRCE Alıcısı-9
4	XRCE Alıcıları	XRCE Alıcısı-1:::Temsilci-1:::Temsilci-3=Temsilci-3 :::Temsilci-2:::XRCE Alıcısı-2
5	XRCE Alıcısı ve Temsilci	Uygulama-1'in XRCE Alıcısı-6:::Uygulama-2'nin Temsilcisi-6
6	XRCE Alıcıları, Temsilciler ve DDSAK'ler	Birkaçı veya hepsinin aynı anda uygulanması

*DDSAK: DDS Alan Katılımcısı

Dağıtım modelleri hakkında anlatmaya değer modele özel bazı özellikler bulunmaktadır.

Üçüncü dağıtım modeli için, bu özelliğe modelin odaklandığı iletim yolunda rastlanmaktadır. Modelde yayım yapan XRCE alıcısına bağlı olan temsilci, yayımcıdan paketleri aldıktan sonra paketlerin izlediği yolun ne olduğu konusunda belirsizlik mevcuttur. Paketlerin izleyebileceği üç olası yol vardır. 1) Temsilci, alıcılar için ayrı ayrı vekil DDS alan katılımcıları oluşturur ve oluşturduğu farklı vekil DDS alan katılımcıları birbirleriyle haberleşiyormuş gibi XRCE alıcılarının iletişimini sağlar. 2) Temsilci, bir tane vekil DDS alan katılımcısı oluşturur ve vekil DDS alan katılımcısı DDS Global Veri Uzayı üzerinden kendisiyle haberleşiyormuş gibi XRCE alıcılarının haberleşmesini sağlar. 3) Temsilci, kendisiyle bağlantı kuran ve birbiriyle haberleşen XRCE alıcıları için kendi üzerinden kısa bir yol oluşturarak iletişimi sağlar. Birinci ve/veya ikinci olası yol tabloda gösterilmiştir.

Üçüncü ve dördüncü dağıtım modellerinin iletim yolları konusunda ortak bir noktası bulunmaktadır. Her iki model de kendisiyle bağlantı kurmuş farklı alıcıları yöneten bir temsilciye sahiptir. Bu yüzden, üçüncü dağıtım modelinde iletim yolu hakkında karşılaşılan belirsizliğe dördüncü dağıtım modelinde rastlamak da muhtemeldir. Dördüncü dağıtımda önceki modellerden farklı olarak, temsilci ile doğrudan bağlantı kurmuş başka temsilcilerin yer aldığı da görülmektedir. Bu nedenle, dördüncü dağıtım modelinin uygulanabilmesi için temsilcilerin XRCE alıcısı olarak da çalışabilmesi gerekmektedir.

Beşinci dağıtım modelinde, diğer uygulamaların temsilcileri ile iletişim kurabilmek için uygulamalar haberleşeceği her uygulama için ayrı olarak XRCE alıcısı oluşturmak zorundadır. Dahası, uygulama temsilciyi diğer uygulamaların XRCE alıcıları ile iletişim kurabilmek için oluşturmaktadır. Uygulamalar arasındaki her bağlantı iletim yolu olarak kabul edilebilir ve beşinci dağıtım modelinin düzeninden dolayı, model uygulandığında var olan iletim yolları çok sayıda olabilir. Beşinci dağıtım modelinin uygulanması genellikle kaynak kısıtlı cihazlar için uygun değildir. Çünkü diğer XRCE alıcılarından gelen mesajları temsilci sürekli işlediği için uygulama periyodik olarak uyuyamaz ve uyanamaz. Bir istisna olarak, beşinci dağıtım modeli DDS ile etkileşimin gerçekleşmediği DDS-XRCE'nin tek dağıtım modelidir.

3.2.4. DDS-XRCE Nesne Modeli

DDS-XRCE nesne modeli incelendiğinde beş sınıf görülmektedir: Tek Kök (The Root Singleton), Vekil Alıcı (ProxyClient), Uygulama (Application), Erişim Kontrolcüsü (AccessController), Alan Katılımcısı (DomainParticipant).

- DDS-XRCE'de nesne modeli modüller içermemektedir.
- DDS'den farklı olarak özel bir ismi bulunmamaktadır.
- En yüksek seviyede sadece sınıflar içermektedir.

Tek Kök (The Root Singleton) bütün nesneler için fabrika olarak çalışmaktadır ve temsilci bu nesnelerden sorumludur. Ayrıca, Tek Kök sisteme giriş yeridir.

XRCE alıcısı uygulaması ve temsilci birbiriyle XRCE protokolü üzerinden haberleştiğinde, Vekil Alıcı (ProxyClient) sınıfı XRCE alıcısı uygulamasını temsil etmektedir. Her Uygulama nesnesi tek bir XRCE Vekil Alıcısıyla ilişkili olarak Vekil Alıcının erişim haklarını elde etmektedir.

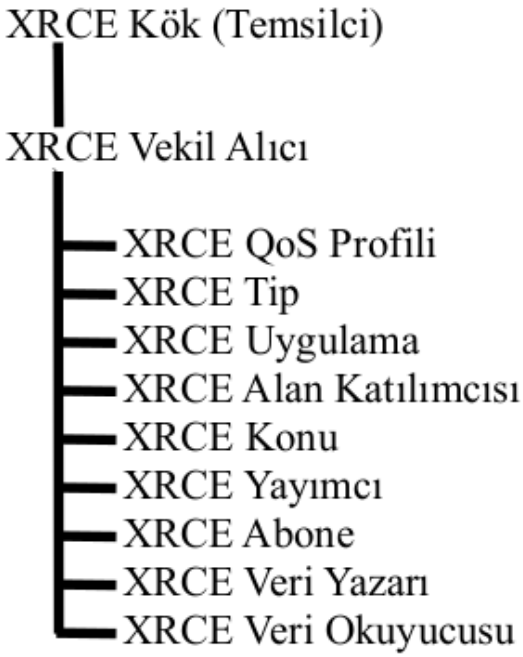
Uygulama (Application) sınıfı, DDS Alanlarında yayım ve abonelik işlemleri için kullanılan DDS nesnelerinden sorumlu bir yazılım uygulamasını XRCE alıcısıyla ilişkilendirerek ifade eder. Bir XRCE Uygulaması birçok Alan Katılımcısı ile ilgili olabilir veya hiçbiriyle olmayabilir. Buradan hareketle, bir XRCE Uygulaması vekil nesneler kullanarak birçok DDS Alanı'nda aktif olabilir veya hiçbirinde olmayabilir.

Bir XRCE Vekil Alıcısı, kaynaklar ve operasyonlarla ilgili sınırlı otoriteye sahiptir. Hakları olan bir alıcıyla ilgili kuralları barındırdığı için, söz konusu otorite Erişim Kontrolcüsü (AccessController) tarafından bir Vekil Alıcısı için belirlenir ve sağlanır. Bu haklar sahip olana, bir uygulama bir alıcı için vekil varlıklar oluşturmaya ve çalıştırmaya niyetlendiğinde DDS alanını seçmek ve bir uygulama yayım yapmak/abone olmak istediğinde DDS konularına karar vermek gibi otorite vermektedir.

DDS Global Veri Uzayı'nda bir vekil olarak çalışan DDS-XRCE'nin Alan Katılımcısı (DomainParticipant) bir DDS alanıyla bağlantıyı ve Uygulama'nın o alanda çalışırken ne yapabildiğini temsil etmektedir.

3.2.5. DDS-XRCE Varlıkları Oluşumu

Temsilci'yi temsil eden XRCE Kök XRCE Vekil Alıcısı'nı oluşturmaktadır. XRCE Vekil Alıcısı ise Hizmet Kalitesi Profili, Tip, Uygulama, Alan Katılımcısı, Yayımcı, Abone, Veri Yazarı ve Veri Okuyucusu oluşturur. DDS'den farklı olarak DDS-XRCE'de, Yayımcı ve Abone Veri Yazarı ve Veri Okuyucusu oluşturmaz. DDS-XRCE'de fabrika gibi çalışan tek nesne, Temsilci tarafından kontrol edilen nesnelerden sorumlu olduğu için Tek Kök'tür. Şekil 3.7 DDS-XRCE nesnelerinin oluşumunu göstermektedir.



Şekil 3.7. DDS-XRCE varlıklarının oluşum ağacı

3.2.6. DDS-XRCE Mesajları ve Yapısı

DDS-XRCE alıcıları ve temsilcileri arasındaki iletişim DDS-XRCE protokolü ile sağlanmaktadır. Şekil 3.8'de gösterildiği gibi, bir DDS-XRCE mesajı başlık ve alt mesaj kısımları içermektedir.



Şekil 3.8. DDS-XRCE protokolünün mesaj yapısı

Başlık; oturum kimliği, akış kimliği, dizi numarası ve alıcı anahtarı hakkında bilgiler taşımaktadır. Alt mesaj, alt mesaj başlığı ve yük kısımlarından oluşmaktadır. Alt mesaj başlığı; alt mesaj kimliği, etiket ve alt mesaj uzunluğu kısımlarından meydana gelmektedir. Yük ise alt mesaj kimliğine bağlı olarak alt mesaj hakkında bilgiler sağlamaktadır. DDS-XRCE alt mesaj tipleri Tablo 3.4'te listelenmiştir.

Tablo 3.4. DDS-XRCE alt mesaj tipleri

Alt Mesaj Tipleri	
Tip	**Açıklama**
Create_Client	Alıcıdan temsilciye gönderilmektedir. Temsilcide bir ProxyClient oluşturmaktadır ve alıcı ile temsilci arasındaki bağlantıyı başlatmaktadır.
Create	Alıcıdan temsilciye gönderilmektedir. Bir XRCE nesnesi oluşturmaktadır.
Get_Info	Alıcıdan temsilciye gönderilmektedir. Bir XRCE nesnesi hakkında bilgi talep etmektedir.
Delete	Alıcıdan temsilciye gönderilmektedir. Bir veya bir grup XRCE nesnesini silmektedir.
Status_Agent	Temsilciden alıcıya gönderilmektedir. Create_Client alt mesajına cevap olarak gönderilmektedir ve Temsilci hakkında bilgi içermektedir.
Status	Temsilciden alıcıya gönderilmektedir. Create, Update ya da Delete alt mesajlarına cevap olarak kullanılmaktadır. Bir XRCE nesnesinin durumu hakkında bilgi içermektedir.
Info	Temsilciden alıcıya gönderilmektedir. Get_Info alt mesajına cevap olarak gönderilmektedir. Bir XRCE nesnesi veya XRCE temsilcisi hakkında detaylı bilgi içermektedir.
Write_Data	Alıcıdan temsilciye gönderilmektedir. Bir XRCE veri yazarı kullanılarak verinin yazılması için kullanılmaktadır.
Read_Data	Alıcıdan temsilciye gönderilmektedir. Bir XRCE veri okuyucusu kullanarak veriyi okumak için kullanılır.
Data	Temsilciden alıcıya gönderilmektedir. Read_Data alt mesajına cevap olarak kullanılmaktadır. Bir XRCE veri okuyucusu tarafından alınan veriyi sağlamaktadır.
Acknack	Çift yönlüdür. Bir dizi sıra numarasına olumlu ve/veya olumsuz kabul bilgisi göndermektedir.
Heartbeat	Çift yönlüdür. Ulaşılabilir sıra numarası aralığı hakkında bilgi vermektedir.
Reset	Çift yönlüdür. Bir oturumu yeniden ayarlamaktadır.
Fragment	Çift yönlüdür. Veri parçası iletmektedir. Mesaj boyutu desteklenenden daha büyük olan mesajlar için kullanılmaktadır.
Timestamp	Çift yönlüdür. Zaman damgası bilgisini iletmektedir.
Timestamp_Reply	Çift yönlüdür. Zaman damgası mesajına cevap vermek için kullanılmaktadır.

3.3. eProsima Yazılım Ürünleri

eProsima ağ iletişimi yüksek performanslı ara katman yazılımı çözümleri sağlar. OMG'nin bir üyesi olarak eProsima ara katman yazılımı standartları oluşturmakta ve uygulamaktadır. Bunlar DDS, RTPS, DDS-XRCE, CDR, DDS üzerine RPC ve benzeridir. Buna ek olarak eProsima, sağladığı ana ürünlerin kaynak kodlarını Github'da Apache 2.0 lisansıyla yayınlamaktadır.

DDS-XRCE, bu çalışmada başlıca odaklanılan standarttır. DDS-XRCE'nin ağ gecikmesi ve verimindeki performansını değerlendirirken eProsima tarafından sağlanan bazı ürünler kullanılmıştır.

Bütün performans değerlendirme testleri Ubuntu 18.04 LTS işletim sisteminin çalıştığı bir bilgisayar ortamında gerçekleşmiştir. Bu nedenle kullanılan bütün eProsima ürünlerinin kurulum işlemleri Ubuntu için olan kurulum kılavuzuna göre tamamlanmıştır (URL-3).

3.3.1. eProsima Micro-XRCE-DDS-Agent v2.0.0

Alıcı ve sunucu haberleşmesi DDS-XRCE protokolünde temeldir. Sunucu, güvenliği sağlamak ve iletişimi emniyete almak için işlevi XRCE alıcıları ile DDS dünyası arasında bir köprü olarak temsilci tarafından temsil edilir. Daha detaylı olarak, temsilci XCRE alıcılarına iletmek için DDS dünyasından mesajlar alır ve/veya mesajları DDS dünyasına iletmek için XRCE alıcılardan alır.

DDS-XRCE protokolündeki temsilcinin kütüphane uygulaması eProsima tarafından Github'da sağlanan Micro-XRCE-DDS-Agent kaynak kodudur. Bu uygulama mikrokontrolörler ve mikrosensörler gibi cihazların DDS dünyasıyla haberleşmesine olanak sağlar.

Ayrıca Temsilci kütüphane uygulamasının bir özelliği UDP sürüm 4, UDP sürüm 6, TCP sürüm 4, TCP sürüm 6 ve seri iletişim gibi bazı yerleşik taşımalar sağlamasıdır. XRCE alıcısı kütüphanesi ile temsilci kütüphanesi arasındaki iletişim daha önce bahsedilen yerleşik taşımalar aracılığıyla sağlanmaktadır (URL-1). Bu çalışmada, Micro-XRCE-DDS-Agent kaynak kodu değişiklik yapılmadan kullanılmıştır.

3.3.2. eProsima Micro-XRCE-DDS-Client v2.0.0

Alıcı ve sunucu haberleşmesinin DDS-XRCE protokolünde temel olduğu ve temsilcinin işlevinin ne olduğundan bahsedilmiştir. Böylelikle, temsilci DDS dünyasıyla iletişim halindeyken diğer iletim ucunun XRCE alıcısı olduğu açıktır. Temsilci, XRCE alıcılarının isteklerine göre onların yerine konular yayımlamaktadır ve/veya konulara abone olmaktadır. Konular, yayımcılar, katılımcılar ve aboneler gibi XRCE alıcıları tarafından DDS dünyasında kullanmak üzere ihtiyaç duyulabilecek varlıklar seçilmiş bir vekil alıcısı tarafından oluşturulmaktadır.

Micro-XRCE-DDS-Client kütüphanesi ayarlanabilmektedir. Bazı CMake etiketlerini etkinleştirip veya etkisizleştirilebilen profil seçenekleri gibi değiştirilebilir özelliklere sahiptir. Kütüphanenin yetkinliklerini kontrol etmek için kullanılabilen bazı diğer parametrelere de sahiptir. Dahası, XRCE alıcısı kütüphanesi temsilci kütüphanesinin yaptığı gibi yerleşik taşımalar sağlamaktadır (URL-1).

Çalışma için bahsedilen değiştirilebilir özelliklerden yararlanılmıştır. Test senaryoları için farklı konuların yayımını yapacak yayımcılara ve bu konulara abone olacak abonelere ihtiyaç duyulmuştur. Bu yüzden, çalışma için kaynak kodlarda bazı değişiklikler yapmak zorunluluktur. Micro-XRCE-DDS-Client kütüphanesi, kütüphanenin nasıl çalıştığını daha iyi anlamak ve hangi parametre/parametrelerin kullanılacağına karar vermek amacıyla çalışmanın deneme sürecinde referans olarak kullanılmıştır.

3.3.3. eProsima Micro-XRCE-DDS-Gen

Bir kod üretim aracı olan Micro-XRCE-DDS-Gen, konular ve bazı tamamlayıcı dosyalar oluşturmak için kullanılmaktatır. Micro-XRCE-DDS-Gen kullanılarak oluşturulan kod, Micro CDR kütüphanesi olmadan oluşturulamamaktadır. Araç, konuları kaynak dosyası olarak bir IDL dosyası ve Micro CDR kütüphanesi kullanarak oluşturur. Bu yüzden, bu kütüphanenin sahip olduğu tek bağımlılık Micro CDR kütüphanesinedir (URL-1).

Nem, sıcaklık, basınç ve irtifa gibi farklı konular için farklı çalışma alanları elde etmek amacıyla IDL dosyaları oluşturulmuştur. Oluşturma tamamlandıktan sonra, yayımcı ve abone dosyaları çalışmanın ihtiyacına göre düzenlenmiş, derlenmiş ve yüklenmiştir. Sonuç olarak, yayımcılarının ve abonelerinin farklı ulaşım profilleri ve akış modları olan

farklı konuların uygulamaları elde edilmiştir. Bu uygulamalar kullanılarak birbiriyle haberleşen yayımcı ve abonelerin yer aldığı bir ağ oluşturmak mümkündür.

3.3.4. eProsima Fast-DDS v2.4.0

Fast-DDS, eProsima tarafından DDS için sağlanan kaynak koddur. Önceki adı Gerçek Zamanlı Yayım Yap-Abone Ol'dur (Real-Time Publish-Subscribe, RTPS). RTPS, bazı taşımalar üzerinden iletişimi sağlayan bir kablolu protokoldür ve DDS için üretilmiştir.

Fast-DDS kütüphanesi, RTPS protokolünün uygulamasını ve onun işlevlerinin tümüne erişim sağlamaktadır (URL-1). Çalışmada, bu uygulama nasıl çalıştığını anlamak amacıyla referans olarak kullanılmıştır.

3.3.5. eProsima Fast-DDS-Gen

Micro-XRCE-DDS-Gen kütüphanesiyle benzer olarak, bu uygulama da bir IDL dosyasından kod üretme aracıdır. Aralarındaki fark, Micro-XRCE-DDS-Gen aracı DDS-XRCE içinken Fast-DDS-Gen aracının DDS için olmasıdır. Fast-DDS-Gen uygulamasının oluşturmasıyla elde edilen kod, bütün Fast-DDS uygulamalarında fazladan bir özelliğe ihtiyaç duymadan çalışabilmektedir (URL-4).

DDS çalışma alanlarının oluşturulması DDS-XRCE çalışma alanlarının oluşturulmasıyla benzerdir. Adımlar kabaca aşağıda anlatılmıştır.

- İlk olarak, konu çalışma alanının kaynağı olacak bir IDL dosyası oluşturulmuştur.
- Fast-DDS-Gen aracına kaynak kod olacak IDL dosyası belirtilerek araç çalıştırılmıştır.
- IDL dosyasından çalışma alanı kodları oluşturulması tamamlandıktan sonra yayımcı, abone ve diğer tamamlayıcı dosyalar elde edilmiştir.
- Son adımda, yayımcı ve abone dosyaları düzenlenmiş, derlenmiş ve yüklenmiştir.

Böylelikle, farklı konuların yayımcıları ve abonelerinin olduğu çalışma alanları DDS için elde edilmiştir. DDS alan katılımcıları bu çalışmada çoğunlukla DDS Global Veri Uzayı'nı gözlemlemek amacıyla kullanılmıştır. Ancak her DDS alan katılımcısı bir konuyla ilgilendiği için, her konu için DDS alan katılımcısını ağa dahil etmek gerekebilir.

3.4. Wireshark

Wireshark, çeşitli işletim sistemlerinde ve diğer platformlarda çalışabilen bir ağ protokolü analizörüdür (HyunHo Kim ve diğ., 2020). Paketleri ağdan yakalamakta ve kullanıcının ağ trafiğini gerçek zamanlı olarak izlemesine olanak sağlamaktadır. Kullanıcı, ağdan yakaladığı paketlerin bilgisini Wireshark 1.8 ve sonraki sürümlerin varsayılan olarak ürettiği pcapng dosyası olarak kaydedebilmektedir. Pcapng dosyaları, diğer formatlı dosyalarda olduğu gibi paket bilgilerini içinde depolamak için veya başka amaçlarla kullanılabilir.

Wireshark, ağ trafiği bilgisini simülasyon sırasında elde etmek amacıyla kullanılmıştır. Ağdan elde edilenler pcapng dosyası olarak kaydedildikten sonra, yakalanan bütün paketler gerekli özellikleri ayıklamak amacıyla ayrı bir dosya olarak yazdırılmıştır. Bunlar protokol, kaynak portu, hedef port, paket uzunluğu ve zaman damgasıdır. Ayıklanmış özellikler, başlangıç olarak, çalışmada odaklanılan mesajları ağda dolaşan diğer mesajlardan ayırt edebilmek için kullanılmıştır ve ikincil olarak ağ gecikmesini ölçmek ile verimi hesaplamak için kullanılmıştır.

3.5. Gnome Terminator

Python diyalogu olarak geliştirilmiş bir komut istemidir. Çoklu terminallerin aynı pencerede çalışması işlevine sahiptir. Kullanıcı çoklu terminal özelliğinden pencereyi bölerek yararlanabilmektedir. Ayrıca, bölünmüş pencereler sonradan birleştirilebilmektedir (URL-5).

Çalışma için çoklu konu mesajlarını aynı anda gösterebilecek bir ortama ihtiyaç duyulmuştur. Bu yüzden Gnome Terminator birçok temsilci, yayımcı ve aboneyi çalıştırabilmek ve ağ trafiği mesajlarını gerçek zamanlı olarak insan gözüyle takip edebilmek için kullanılmıştır. Terminallerin düzeni organize edildikten sonra, terminallerdeki kodlar test planına göre çalıştırılmıştır ve ağ trafiği mesajları izlenebilmiştir.

4. KULLANILAN TEST SENARYOLARI VE YÖNTEMLER

4.1. Hazırlık Aşaması

Hazırlık aşaması test senaryolarının oluşturulduğu ve farklı konular için farklı özellikler ve koşullara göre çalışan kaynak kodların oluşturulduğu iki aşamadan oluşmaktadır.

4.1.1. Senaryoların Oluşturulması

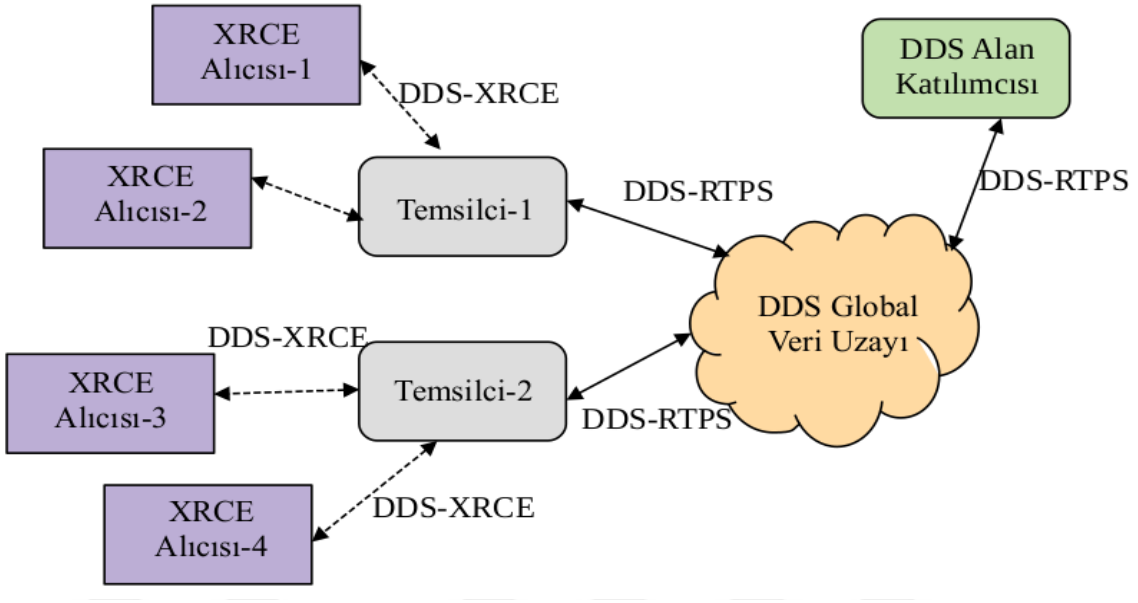
Başlangıç olarak, test senaryosu için DDS-XRCE'nin kullanıldığı bir ortama ihtiyaç duyulmuştur. OMG'nin paylaştığı DDS-XRCE tanımlama belgesinde, dağıtım modelleri DDS-XRCE kullanılarak ne tür ağlar oluşturulabilir veya DDS-XRCE ne şekillerde uygulanabilir sorularına cevap olarak sunulmuştur.

Dağıtım modelleri temel olarak haberleşen tarafların türü, iletim yolu ve kullanılan protokollere bağlı olarak oluşmaktadır. Bu nedenle dağıtım modelleri; bir test senaryosu elde etmek amacıyla ana odak noktası bozulmadan fazladan XRCE alıcıları, temsilciler ve DDS katılımcıları eklenerek yeniden düzenlenmiştir. Bu tez çalışması için birinci, ikinci ve üçüncü dağıtım modellerinin referans alınarak kullanılmasına karar verilmiştir.

Şekil 4.1, DDS-XRCE'nin birinci dağıtım modelinin bu çalışma için yeniden düzenlenmiş halini temsil etmektedir. Dağıtım Modeli-1'in oluşturulması için DDS-XRCE tanım belgesinde sunulan şekline Şekil 4.1'deki XRCE Alıcısı-2, Temsilci-2, XRCE Alıcısı-3 ve XRCE Alıcısı-4 eklenmiştir.

Temsilciler DDS Global Veri Uzayı'nın üyeleridir ve kendilerinin ilgili XRCE alıcılarına doğrudan bağlılardır. Temsilciler, DDS Global Veri Uzayı'nda diğer DDS katılımcıları ile RTPS protokolü üzerinden iletişim kurmaktadır. Bağlanmak amacıyla ilgili temsilciyi tanımak için XRCE alıcılarına yardımcı olan farklı port numaralarına sahiplerdir. Temsilci-1 ve Temsilci-2'nin port numaraları sırasıyla 2018 ve 2019'dur.

Dağıtım Modeli-1'in testleri sırasında DDS Katılımcısı ve XRCE Alıcısı-2 arasındaki iletişime odaklanılmıştır. Birbirleriyle ilgili XRCE alıcısının temsilcisi aracılığıyla iletişim kurmuşlardır.



Şekil 4.1. Dağıtım Modeli-1

Dağıtım Modeli-1'in diğer nesneleri ilgilendikleri konu, bağlantı kurdukları nesneler ve iletişim kurdukları nesnelerle Tablo 4.1'de listelenmiştir.

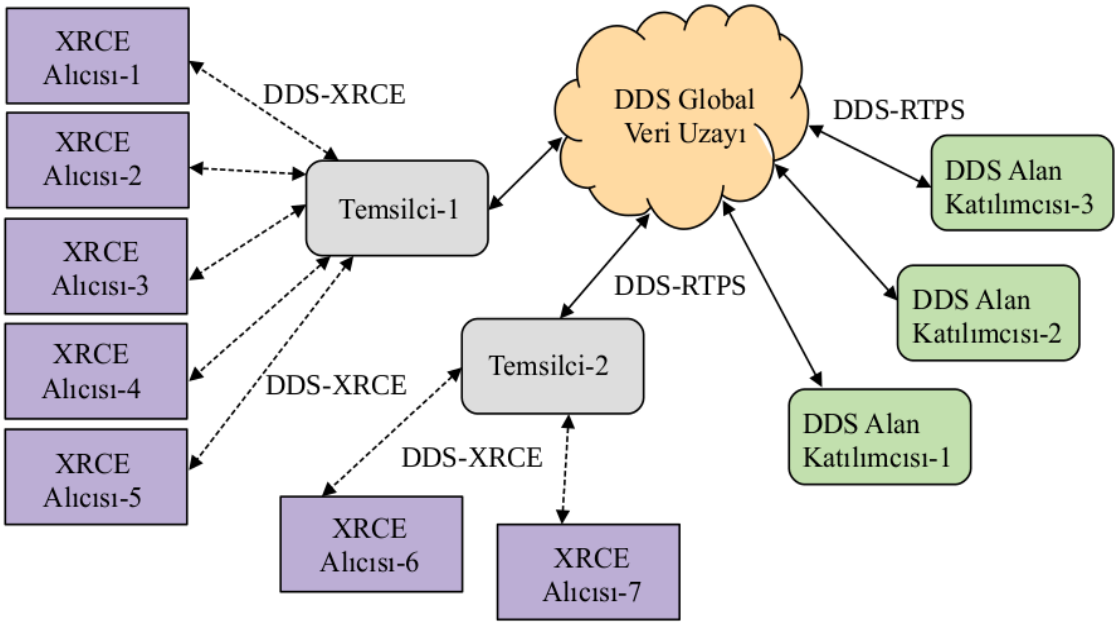
Tablo 4.1. Dağıtım Modeli-1 ve detayları

Nesneler	**Dağıtım Modeli-1 Detayları**			
	Yayımcı/Abone	**Konu**	**Bağlantı**	**İletişim**
DDSAK	Yayımcı	Nem	Temsilci-1 (vekil DDSAK)	XRCE Alıcısı-2
XRCE Alıcısı-1	Yayımcı	Sıcaklık	Temsilci-1	-
XRCE Alıcısı-2	Abone	Nem	Temsilci-1	DDSAK
XRCE Alıcısı-3	Yayımcı	İrtifa	Temsilci-2	-
XRCE Alıcısı-4	Abone	Basınç	Temsilci-2	-

*DDSAK: DDS Alan Katılımcısı

Şekil 4.2, DDS-XRCE'nin ikinci dağıtım modelinin bu çalışma için yeniden düzenlenmiş halini temsil etmektedir. Dağıtım Modeli-2'nin oluşturulması için DDS-XRCE tanım belgesinde sunulan şekline Şekil 4.2'deki XRCE Alıcısı-2, XRCE Alıcısı-3, XRCE Alıcısı-4, XRCE Alıcısı-5, XRCE Alıcısı-7 ve DDS Alan Katılımcıları eklenmiştir.

Dağıtım Modeli-2'nin temsilcilerinin özellikleri Dağıtım Modeli-1'in temsilcilerinin özellikleriyle aynıdır. Dağıtım Modeli-2'nin diğer nesneleri ilgilendikleri konu, bağlantı kurdukları nesneler ve iletişim kurdukları nesnelerle Tablo 4.2'de listelenmiştir.



Şekil 4.2. Dağıtım Modeli-2

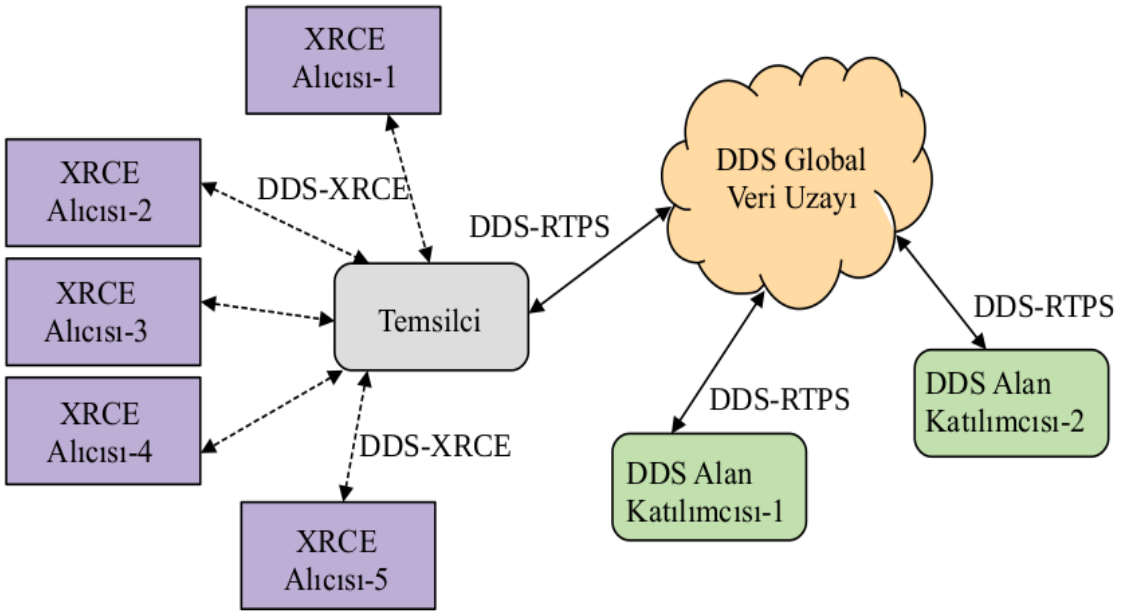
Tablo 4.2. Dağıtım Modeli-2 ve detayları

Nesneler	**Dağıtım Modeli-2 Detayları**			
	Yayımcı/Abone	**Konu**	**Bağlantı**	**İletişim**
DDSAK-1	Abone	HelloWorld	Temsilci-1 (vekil DDSAK)	XRCE Alıcısı-1, XRCE Alıcısı-2
DDSAK-2	Abone	Nem	Temsilci-1 (vekil DDSAK)	XRCE Alıcısı-5
DDSAK-3	Abone	Nem	-	-
XRCE Alıcısı-1	Yayımcı	HelloWorld	Temsilci-1	DDSAK-1, XRCE Alıcısı-3, XRCE Alıcısı-4
XRCE Alıcısı-2	Yayımcı	HelloWorld	Temsilci-1	DDSAK-1, XRCE Alıcısı-3, XRCE Alıcısı-4
XRCE Alıcısı-3	Abone	HelloWorld	Temsilci-1	XRCE Alıcısı-1, XRCE Alıcısı-2
XRCE Alıcısı-4	Abone	HelloWorld	Temsilci-1	XRCE Alıcısı-1, XRCE Alıcısı-2
XRCE Alıcısı-5	Yayımcı	İrtifa	Temsilci-1	XRCE Alıcısı-6, DDSAK-2
XRCE Alıcısı-6	Abone	İrtifa	Temsilci-2	XRCE Alıcısı-5
XRCE Alıcısı-7	Abone	Nem	Temsilci-2	-

*DDSAK: DDS Alan Katıımcısı

Dağıtım Modeli-2'nin testleri sırasında XRCE Alıcısı-5 ve XRCE Alıcısı-6 arasındaki iletişime odaklanılmıştır. XRCE Alıcısı-5 ve XRCE Alıcısı-6, birbirleriyle kendilerinin ilgili temsilcileri aracılığıyla iletişim kurmuşlardır.

Şekil 4.3, DDS-XRCE'nin üçüncü dağıtım modelinin bu çalışma için yeniden düzenlenmiş halini temsil etmektedir. Dağıtım Modeli-3'ün oluşturulması için DDS-XRCE tanım belgesinde sunulan şekline Şekil 4.3'teki XRCE Alıcısı-3, XRCE Alıcısı-4, XRCE Alıcısı-5, ve DDS Alan Katılımcıları eklenmiştir.



Şekil 4.3.Dağıtım Modeli-3

Dağıtım Modeli-3'ün temsilcisinin özellikleri Dağıtım Modeli-1'in Temsilci-1'inin özellikleriyle aynıdır. Dağıtım Modeli-3'ün diğer nesneleri ilgilendikleri konu, bağlantı kurdukları nesneler ve iletişim kurdukları nesnelerle Tablo 4.3'te listelenmiştir.

Dağıtım Modeli-3'ün testleri sırasında XRCE Alıcısı-2 ve XRCE Alıcısı-4 arasındaki iletişime odaklanılmıştır. XRCE Alıcısı-2 ve XRCE Alıcısı-4, birbirleriyle bağlı oldukları aynı temsilci aracılığıyla iletişim kurmuşlardır.

Fark edilecektir ki bazı XRCE alıcıları ilgilendikleri konularla ilgili hiçbir mesaj almamaktadır. Bütün aboneler gerçek zamanlı uygulamalarda bile sürekli olarak mesaj almayabilmektedir. Bu nedenle, herhangi bir mesaj almayan XRCE alıcıları deneyleri etkilemeyecektir.

Tablo 4.3. Dağıtım Modeli-3 ve detayları

Nesneler	**Dağıtım Modeli-3 Detayları**			
	Yayımcı/Abone	**Konu**	**Bağlantı**	**İletişim**
DDSAK-1	Abone	HelloWorld	Temsilci (vekil DDSAK)	XRCE Alıcısı-1, XRCE Alıcısı-5
DDSAK-2	Abone	Nem	Temsilci (vekil DDSAK)	XRCE Alıcısı-2
XRCE Alıcısı-1	Yayımcı	HelloWorld	Temsilci	DDSAK-1, XRCE Alıcısı-3
XRCE Alıcısı-2	Yayımcı	Nem	Temsilci	DDSAK-2, XRCE Alıcısı-4
XRCE Alıcısı-3	Abone	HelloWorld	Temsilci	XRCE Alıcısı-1, XRCE Alıcısı-5
XRCE Alıcısı-4	Abone	Nem	Temsilci	XRCE Alıcısı-2
XRCE Alıcısı-5	Yayımcı	HelloWorld	Temsilci	DDSAK-1, XRCE Alıcısı-3

*DDSAK: DDS Alan Katılımcısı

Taşıma profili seçenekleri, akış modları ve ağ katmanı protokolleri; ilgili fonksiyonlar kullanarak DDS-XRCE'nin yayımcı ve abonelerinin kaynak kodları aracılığı ile ayarlanabilmektedir. Bu seçenekler derleme işleminden önce seçilmeli ve ayarlanmalıdır. Bu yüzden; güvenilir akışta, ağ katmanında IPv4 protokol çalışacak ve mesajlar taşıma katmanında UDP protokol üzerinden gönderilip alınacak şekilde farklı konuların yayımcıları ve aboneleri oluşturulmuştur. Ayrıca UDP-EnİyiÇaba, TCP-Güvenilir ve TCP-EnİyiÇaba gibi taşıma profili-akış modu çiftleri için de aynısı geçerlidir.

Bu çiftler test ortamı koşulları olduğundan, her dağıtım modelinin dört farklı koşula göre test edileceği anlamına gelmektedir. Belirtmek gerekir ki testler belirli bir koşula göre gerçekleştirilirken, örneğin UDP-Güvenilir koşulları. Test senaryosu olarak kullanılan dağıtım modelindeki bütün XRCE Alıcıları; güvenilir akışta, ağ katmanında IPv4 protokolü ve taşıma katmanında UDP protokolü üzerinden mesajları taşıyacak şekilde ayarlanmıştır.

Ancak, UDP-EnİyiÇaba testlerinin ilerleyen aşamalarında bir işlev farklılığı ile karşılaşılmıştır. Farklılık sebebiyle, UDP-EnİyiÇaba çiftlerine göre çalışan haberleşmesine odaklanılan yayımcıların ve abonelerin kaynak kodlarına ek kodlar yazılmıştır. Bu eylemlerin amacı ilerleyen aşamalarda açıklanacaktır.

4.1.2. Farklı Koşulların ve Konu Çalışma Alanlarının Oluşturulması

DDS-XRCE, birçok parametreye sahip çok işlevli bir protokoldür ve iletişimin kalitesini ayarlamak için birçok seçenek sağlamaktadır. Test ortamı koşulları konusunda; DDS-XRCE'nin taşıma profili, akış modu ve ağ katmanı protokolü özelliklerinin kullanımına karar verilmiştir. Ağ katmanında, kullanıcının seçmesi için IP versiyon 4 ve IP versiyon 6 protokolleri sağlanmaktadır. Bu çalışmada ağ katmanında sadece IP versiyon 4 protokolü kullanılmıştır.

DDS-XRCE taşıma katmanında kullanılacak protokol açısından çeşitlilik sunacak şekilde kurgulanmıştır ve eProsima uygulamasında ilgili protokol için hazır fonksiyonlar yer almaktadır. Taşıma profili ise DDS-XRCE'nin kullanıcıya mesajları iletmesi için taşıma protokolü konusunda sağladığı bir seçenektir. Taşıma profili özelliğini kullanarak UDP, TCP, CAN FD, seri ya da özel protokol kullanacağına kullanıcı karar vermektedir. Bu çalışmada UDP ve TCP protokolleri kullanılmıştır. Ağ katmanında IP versiyon 4 protokolü kullanıldığı için taşıma katmanında da UDP versiyon 4 ve TCP versiyon 4 protokolleri kullanılacaktır.

Akış ise konu mesajlarının bağımsız akıntısıdır ve "Güvenilirlik" adıyla Hizmet Kalitesi politikalarından biri olarak yer almaktadır. DDS-XRCE standardının işleyişini sağlayan temel özelliklerden biridir ve iki taraf arasındaki iletişimin kurulup kurulmayacağına karar vermek için kullanılmaktadır.

DDS-XRCE protokolünün iki çeşit akışı vardır: güvenilir akış, en iyi çaba akışı. Güvenilir akışta iletişim, mesajlarda herhangi bir kayıp olmadan sağlanmaktadır. En iyi çaba akışta ise iletişim, taşıma protokolüne güvenerek sağlanmaktadır. Bu nedenle en iyi çaba akışlarda kayıplar gözlemek ihtimal dahilindedir. Bir başka deyişle, güvenilir akışta çalışan bir yayımcı göndermeyi amaçladığı bütün mesajları hedefe ulaştırmaya çalışacaktır.

Güvenilir akışta çalışan bir abone kendisine gönderilen bütün mesajları almaya çalışacaktır. En iyi çaba akışta çalışan bir yayımcı göndermeyi amaçladığı mesajları gönderirken bazı mesajların hedefe ulaşmaması ihtimal dahilindedir. En iyi çaba akışta çalışan bir abone kendisine gönderilen mesajları alırken bazı mesajların alınmaması

ihtimal dahilindedir. Buradan hareketle; güvenilir akışta çalışan bir abone iletişim kurmak istediğinde en iyi çaba akışta çalışan bir yayımcı ile iletişim kuramayacaktır. Çünkü iletişimde kayıplar istenmemektedir ve abone eksiksiz olarak mesajları almak isterken yayımcı eksiksiz olarak gönderemeyecektir.

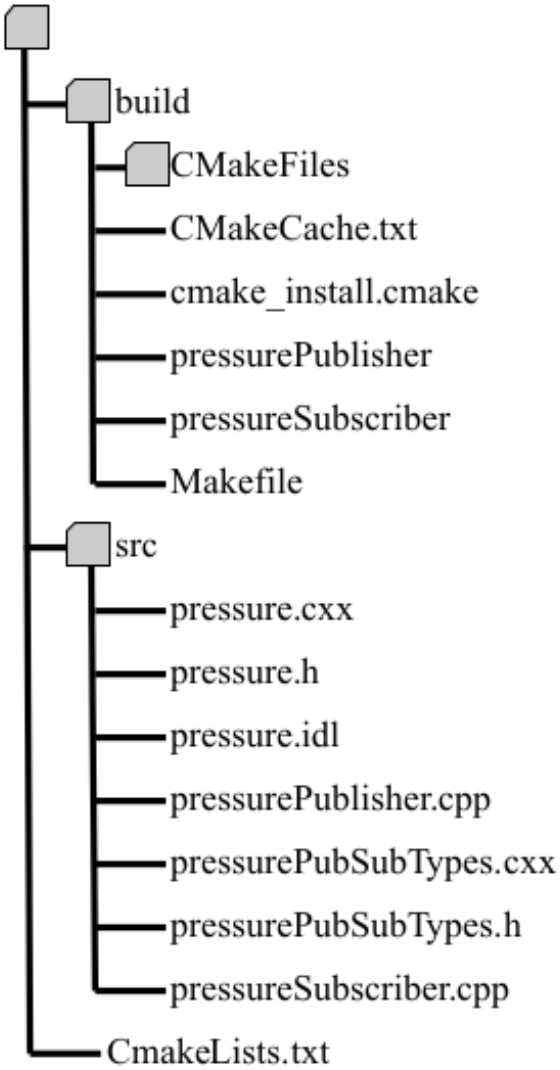
Hizmet Kalitesi politikası gereği aynı akışta çalışan abone ve yayımcıların iletişim kurmasına izin verilmektedir. Diğer yandan en iyi çaba akışta çalışan bir abone ve güvenilir akışta çalışan bir yayımcının iletişim kurmasına da izin verilmektedir. Çünkü beklenenden fazla mesajın başarıyla iletilmesinde bir sakınca bulunmamaktadır. Son olarak güvenilir akışlarda kontrol edilirken, en iyi çaba akışlarında mesajların alınıp alınmadığı kontrol edilmemektedir. Ek olarak güvenilir akışlarda, göndericiye başarılı teslimat gerçekleştiğinde gerçekleştiğine dair ekstra mesajlar gönderilmektedir. Çalışmada iki akış türü de kullanılmıştır.

Mesajların kaynak, hedef ve diğer bilgilerinin takibi; ağda aynı mesajların dolaşmasına sebep olan sadece HelloWorld konusuyla ilgilenen yayımcıların ve abonelerin sayısındaki artış nedeniyle güç hale gelmiştir. Böylelikle, mesajları ayırt etmek için farklı konuların çalışma alanlarından oluşturulmuştur. Çalışma alanları belirli bir konu için genel olarak bir yayımcı, bir abone ve diğer dosyalar içermektedir. Sadece HelloWorld konusu diğer konulara göre farklılık göstermektedir.

Farklı konuların oluşturulması eProsima Micro-XRCE-DDS-Gen ve Fast-DDS-Gen araçlarının kullanımı hakkında sağladığı talimatlar takip edilerek başarılmıştır. Çalışma alanı oluşturmada takip edilen adımlar şu şekildedir.

- Farklı bir konunun çalışma alanını oluşturmak için ilk olarak bir IDL dosyası oluşturulmuştur. Bu çalışma için irtifa, basınç, sıcaklık, nem ve HelloWorld konularının çalışma alanlarından oluşturulmuştur.
- Micro-XRCE-DDS-Gen ve Fast-DDS-Gen araçları kullanılarak yayımcı, abone ve birtakım tamamlayıcı dosyaları içinde barındıran DDS-XRCE ve DDS çalışma alanları oluşturulmaktadır.
- İhtiyaca yönelik olarak gerekli dosyalar düzenlendikten, derlendikten ve yüklendikten sonra kullanıma hazır olarak çalışma alanları elde edilmiştir.

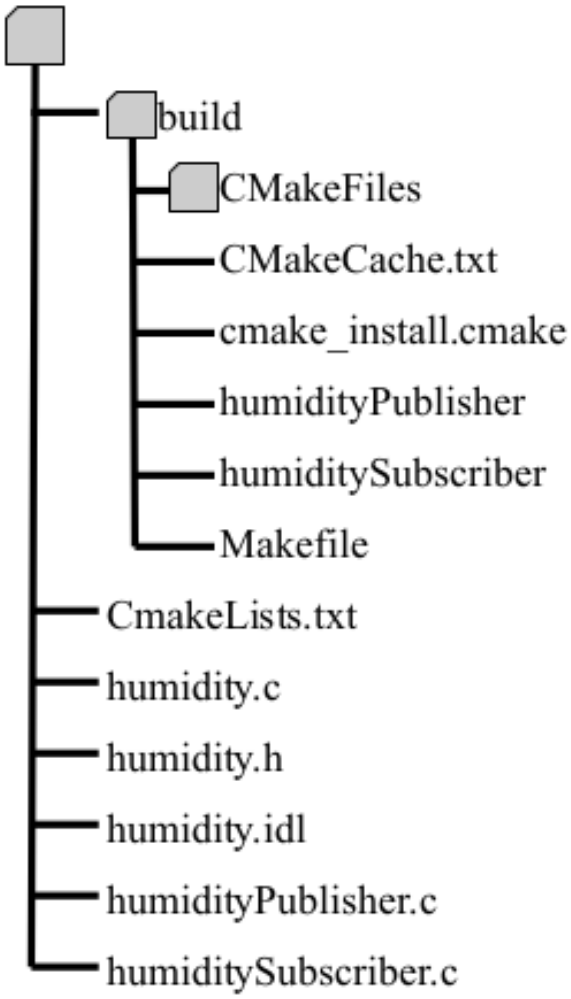
Şekil 4.4'te DDS'nin basınç konusuna ait çalışma alanının dosya ağacı örnek olarak yer almaktadır. "src" klasörü içinde yayımcı, abone ve diğer tamamlayıcı dosyalar yer almaktadır. Bunların "cxx, h" uzantılı olanları Fast-DDS-Gen ve IDL uzantılı belgenin kullanılmasıyla oluşturulan dosyalardır. Yayımcı, abone ve CMakeLists.txt belgeleri kullanıcılar tarafından eklenmektedir.



Şekil 4.4. Bir DDS çalışma alanının dosya ağacı

Terminal penceresinde dosya konumu olarak "build" klasörünün ayarlanmasıyla ilgili belgeler derlenmekte ve yüklenmektedir. Ardından yayımcı ve aboneyi çalıştıracak dosyalar "build" klasöründe oluşturulmaktadır ve terminal penceresinde dosya konumu olarak "build" klasörünün ayarlanmasıyla yayımcı ve abone başlatılmaktadır.

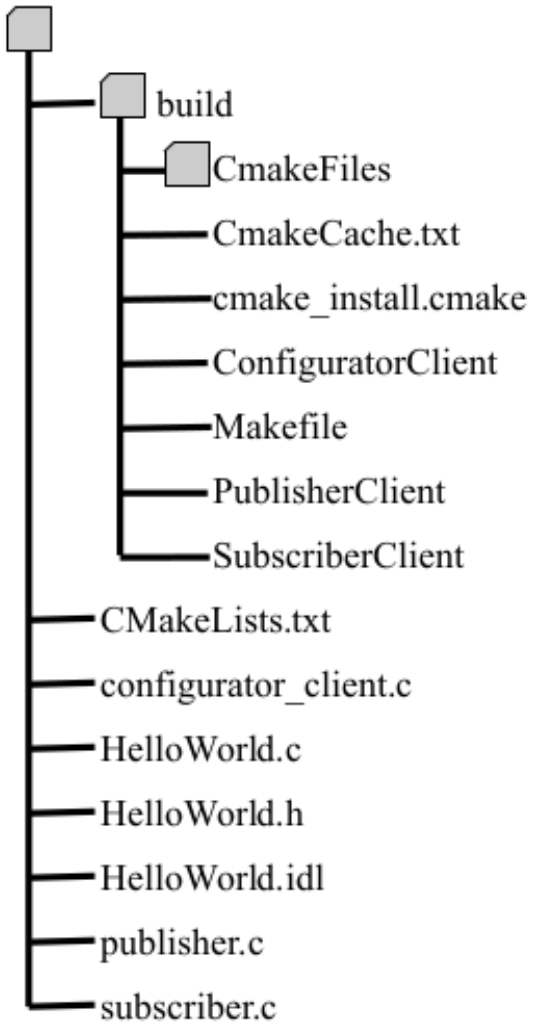
Şekil 4.5'te bir DDS-XRCE çalışma alanında yer alan görünür dosyaların ağacı örnek olarak yer almaktadır. Çalışma alanında IDL uzantılı belge ve Micro-XRCE-DDS-Gen aracı kullanılmasıyla çalışma alanı içindeki "build" klasörü ve içindeki belgeler hariç diğer belgeler oluşturulmaktadır.



Şekil 4.5. Bir DDS-XRCE çalışma alanının dosya ağacı

Ardından çalışma alanına "build" klasörü eklenmektedir. Eklenme amacı, çalışma alanı içindeki belgelerin derlenmesi ve yüklenmesiyle "build" klasörünün içindeki belgelerin oluşturulmasıdır. Ek olarak, sonraki işlemlerde ilgili konunun yayımcı ve abonesinin "build" klasörünün içinden başlatılacak olmasıdır.

Şekil 4.6'da, DDS-XRCE'nin HelloWorld konusuna ait çalışma alanının dosya ağacı örnek olarak yer almaktadır.



Şekil 4.6. Deployment özellikli bir DDS-XRCE çalışma alanının dosya ağacı

Farklı konular için çalışma alanları oluşturulduktan sonra çeşitliliğe bağlı olarak bir sorunla karşılaşılmıştır. Denemeler sırasında bir temsilciye çok çeşitli konuların alıcılarından (yayımcı ve/veya abone) bağlanamadığı gibi, aynı konudan çok sayıda alıcının da farklı alıcılarmış gibi defalarca bağlanamadığı görülmüştür. Buna çözüm olarak eProsima Micro-XRCE-DDS-Client kütüphanesindeki örneklerden "Deployment"tan faydalanılmıştır.

Test edilmesi planlanan Dağıtım Modeli-2 veya Dağıtım Modeli-3 incelendiğinde görülmektedir ki HelloWorld konusuna ait birden fazla alıcı (yayımcı ve/veya abone) yer almaktadır. Bu alıcılar, Micro-XRCE-DDS-Client kütüphanesindeki "Deployment" örneği kullanılarak veya referans alınarak oluşturulan çalışma alanlarına aittir.

UDP-Güvenilir koşulunda gerçekleşen testler için Micro-XRCE-DDS-Client kütüphanesinin “Deployment” örneği kullanılmıştır. UDP-EnİyiÇaba, TCP-Güvenilir ve TCP-EnİyiÇaba koşullarında gerçekleşecek testler içinse, Micro-XRCE-DDS-Client kütüphanesinin “Deployment” örneğinden testin ilgili koşuluna göre çalışacak şekilde türetilmiştir. Bu nedenle, Şekil4.6’daki diğer çalışma alanlarından farklı, “Deployment” özellikli HelloWorld konulu çalışma alanının dosya ağacıdır.

Bu çalışma alanının içinde diğer çalışma alanlarında bulunmayan “configurator_client.c” dosyası bulunmaktadır. “configurator_client.c” dosyasının kullanıcı tarafından eklenmesi gerekmektedir. Bu çalışma alanındaki dosyaların derlenmesi ve yüklenmesiyle “build” klasörünün içindeki dosyalar elde edilmektedir. ConfiguratorClient’ın çalıştırılmasıyla aynı temsilciye bağlı ve aynı konu ile ilgilenen alıcılar oluşturulmaktadır.

4.1.3. DDS-XRCE Çalışma Alanlarındaki İlgili Kaynak Kodların Düzenlenmesi

Micro-XRCE-DDS-Client kütüphanesinde farklı özelliklerde çalışan yayımcı ve aboneler yer almaktadır (URL-1). Bu aşamada farklı özelliklerin nasıl belirlendiği anlatılacaktır (URL-3).

- Taşıma Profili

Şekil 4.7’de hem yayımcı hem de abone için kullanılabilen UDP taşıma kod parçacıkları listelenmiştir. Listelenen kod parçacıkları yayımcı ve abonenin UDP taşıma bağlantısı başlatmak (Şekil 4.7’de 2. satır) ve bağlantıyı kapatmak (Şekil 4.7’de 3. satır) için kullandığı birtakım fonksiyonlardır. Benzer şekilde, Şekil 4.8’de hem yayımcı hem de abone için kullanılabilen TCP taşıma kod parçacıkları listelenmiştir.

UDP Taşıma	
No	**Kod Parçacıkları**
1	#define BUFFER_SIZE UXR_CONFIG_UDP_TRANSPORT_MTU* STREAM_HISTORY

2	// Transport
	uxrUDPTransport transport;
	if (!uxr_init_udp_transport(&transport, UXR_IPv4, ip, port))

3	uxr_close_udp_transport(&transport);

Şekil 4.7. Yayımcı ve abone kaynak kodlarındaki UDP taşıma kod parçacıkları

TCP Taşıma	
No	**Kod Parçacıkları**
1	#define BUFFER_SIZE UXR_CONFIG_TCP_TRANSPORT_MTU* STREAM_HISTORY

2	// Transport
	uxrTCPTransport transport;
	if (!uxr_init_tcp_transport(&transport, UXR_IPv4, ip, port))

3	uxr_close_tcp_transport(&transport);

Şekil 4.8. Yayımcı ve abone kaynak kodlarındaki TCP taşıma kod parçacıkları

UDP taşıma kullanarak çalışan bir yayımcının veya abonenin kaynak kodlarında, UDP taşıma için kullanılan Şekil 4.7'de belirtilen kod parçacıkları; TCP taşıma için kullanılan Şekil 4.8'de belirtilen kod parçacıklarıyla değiştirildikten, derlendikten ve yüklendikten sonra yayımcı veya abone TCP taşıma kullanarak çalışacaktır.

- Akış Modu

Akış modu için gereken kodlar yayımcı ve aboneye göre farklılık göstermektedir. En iyi çaba akış modu kodları, yayımcı için Şekil 4.9'da ve abone için Şekil 4.10'da verilmiştir.

En İyi Çaba Akış Modu - Yayımcı	
No	**Kod Parçacıkları**

1	// Streams
	uint8_t output_reliable_stream_buffer[BUFFER_SIZE];
	uxrStreamId reliable_out = uxr_create_output_reliable_stream(&session, output_reliable_stream_buffer, BUFFER_SIZE,
	STREAM_HISTORY);
	
	uint8_t input_reliable_stream_buffer[BUFFER_SIZE];
	uxr_create_input_reliable_stream(&session, input_reliable_stream_buffer, BUFFER_SIZE, STREAM_HISTORY);
	
	uint8_t output_besteffort_stream_buffer[BUFFER_SIZE];
	uxrStreamId besteffort_out = uxr_create_output_best_effort_stream(&session, output_besteffort_stream_buffer,
	BUFFER_SIZE);

2	uxr_prepare_output_stream(&session, besteffort_out, datawriter_id, &ub, topic_size);

Şekil 4.9. Yayımcı kaynak kodlarındaki en iyi çaba akış modu kod parçacıkları

En İyi Çaba Akış Modu - Abone	
No	**Kod Parçacıkları**

1	// Streams
	uint8_t output_reliable_stream_buffer[BUFFER_SIZE];
	uxrStreamId reliable_out = uxr_create_output_reliable_stream(&session, output_reliable_stream_buffer, BUFFER_SIZE,
	STREAM_HISTORY);
	
	uint8_t input_reliable_stream_buffer[BUFFER_SIZE];
	uxr_create_input_reliable_stream(&session, input_reliable_stream_buffer, BUFFER_SIZE, STREAM_HISTORY);
	
	uxrStreamId besteffort_in = uxr_create_input_best_effort_stream(&session);

2	uint16_t read_data_req = uxr_buffer_request_data(&session, reliable_out, datareader_id, besteffort_in, &delivery_control);

Şekil 4.10. Abone kaynak kodlarındaki en iyi çaba akış modu kod parçacıkları

Güvenilir akış moduyla yayımcının çalışmasını sağlayacak kod parçacıkları Şekil 4.11'de verilmiştir.

Güvenilir Akış Modu - Yayımcı	
No	**Kod Parçacıkları**

1	// Streams
	uint8_t output_reliable_stream_buffer[BUFFER_SIZE];
	uxrStreamId reliable_out = uxr_create_output_reliable_stream(&session, output_reliable_stream_buffer, BUFFER_SIZE,
	STREAM_HISTORY);
	
	uint8_t input_reliable_stream_buffer[BUFFER_SIZE];
	uxr_create_input_reliable_stream(&session, input_reliable_stream_buffer, BUFFER_SIZE, STREAM_HISTORY);

2	uxr_prepare_output_stream(&session, reliable_out, datawriter_id, &ub, topic_size);

Şekil 4.11. Yayımcı kaynak kodlarındaki güvenilir akış modu kod parçacıkları

Güvenilir akış moduyla abonenin çalışmasını sağlayacak kod parçacıkları Şekil 4.12'de verilmiştir. En iyi çaba akışında çalışan yayımcı ve abone kaynak kodlarında, en iyi çaba akışı için kullanılan yayımcı için Şekil 4.9'daki ve abone için Şekil 4.10'daki kod parçacıkları; güvenilir akış için kullanılan yayımcı için Şekil 4.11'deki ve abone için Şekil 4.12'daki kod parçacıklarıyla değiştirildikten, derlendikten ve yüklendikten sonra yayımcı ve abone güvenilir akış modunda çalışacaktır.

Güvenilir Akış Modu - Abone	
No	**Kod Parçacıkları**
1	// Streams uint8_t output_reliable_stream_buffer[BUFFER_SIZE]; uxrStreamId reliable_out = uxr_create_output_reliable_stream(&session, output_reliable_stream_buffer, BUFFER_SIZE, STREAM_HISTORY); uint8_t input_reliable_stream_buffer[BUFFER_SIZE]; uxrStreamId reliable_in = uxr_create_input_reliable_stream(&session, input_reliable_stream_buffer, BUFFER_SIZE, STREAM_HISTORY);
2	uint16_t read_data_req = uxr_buffer_request_data(&session, reliable_out, datareader_id, reliable_in, &delivery_control);

Şekil 4.12. Abone kaynak kodlarındaki güvenilir akış modu kod parçacıkları

4.1.4. "Deployment" Özellikli DDS-XRCE Çalışma Alanlarındaki Kaynak Kodların Düzenlenmesi

Micro-XRCE-DDS-Client kütüphanesinde yer alan ve örnek olarak sunulan "Deployment" adlı çalışma alanının yapısı diğer çalışma alanlarının yapısından biraz farklıdır. Bu nedenle referans alınarak kendisinden türetilen ve farklı modlarda işlemesi için ayarlanan diğer çalışma alanları da ayrı bir şekilde incelenmeyi gerektirmektedir.

- Taşıma Profili

UDP ve TCP taşıma profilleri için kullanılan fonksiyonlar Micro-XRCE-DDS-Client kütüphanesinin Deployment çalışma alanındaki yayımcı, abone ve ConfiguratorClient kaynak kodlarında da kullanılmıştır.

İçinde ConfiguratorClient kaynak kodu bulundurmayan diğer çalışma alanlarından farklı olarak bu çalışma alanının yayımcı ve abonelerinin bağlantı kuracağı temsilci önceden tanımlıdır. Bağlantı kurulması istenen temsilci kodlar aracılığı ile belirtilmektedir.

Şekil 4.13'te UDP taşıma bağlantısı kurmak ve bağlantıyı yönetmek için kullanılacak kod parçacıkları listelenmiştir. Şekil 4.14'te TCP taşıma bağlantısı kurmak ve bağlantıyı yönetmek için kullanılacak kod parçacıkları listelenmiştir.

UDP Taşıma	
No	**Kod Parçacıkları**
1	#define BUFFER_SIZE UXR_CONFIG_UDP_TRANSPORT_MTU* STREAM_HISTORY

2	// Transport
	uxrUDPTransport transport;
	if (!uxr_init_udp_transport(&transport, UXR_IPv4, "127.0.0.1", "2018"))

3	uxr_close_udp_transport(&transport);

Şekil 4.13. ConfiguratorClient, yayımcı ve abone kodlarındaki UDP kod parçacıkları

TCP Taşıma	
No	**Kod Parçacıkları**
1	#define BUFFER_SIZE UXR_CONFIG_TCP_TRANSPORT_MTU* STREAM_HISTORY

2	// Transport
	uxrTCPTransport transport;
	if (!uxr_init_tcp_transport(&transport, UXR_IPv4, "127.0.0.1", "2018"))

3	uxr_close_tcp_transport(&transport);

Şekil 4.14. ConfiguratorClient, yayımcı ve abone kodlarındaki TCP kod parçacıkları

Şekil 4.13'ün ve Şekil 4.14'ün 2 numaralı satırlarından görüldüğü üzere IP ve port numaraları önceden belirlenmiştir. Bunlar, ConfiguratorClient ile oluşturulacak yayımcıların ve abonelerin bağlantı kurması istenen temsilcinin IP ve port numaralarıdır. Bu demektir ki ConfiguratorClient bir uygulama sırasında sadece bir temsilci için çalışmaktadır.

ConfiguratorClient, yayımcı ve abonenin kaynak kodlarındaki TCP taşıma için kullanılan Şekil 4.14'teki kod parçacıkları, UDP taşıma için kullanılan Şekil 4.13'teki kod parçacıklarıyla değiştirildikten, derlendikten ve yüklendikten sonra ConfiguratorClient, yayımcı ve abone UDP taşıma kullanarak çalışacaktır.

- Akış Modu

Şekil 4.15'te en iyi çaba akış modunda çalışan bir ConfiguratorClient'ın kaynak kodundan alınan kod parçacıkları verilmiştir.

En İyi Çaba Akış Modu - ConfiguratorClient	
No	**Kod Parçacıkları**

1	// Streams
	uint8_t output_reliable_stream_buffer[BUFFER_SIZE];
	uxr_create_output_reliable_stream(&session, output_reliable_stream_buffer, BUFFER_SIZE, STREAM_HISTORY);
	
	uint8_t input_reliable_stream_buffer[BUFFER_SIZE];
	uxr_create_input_reliable_stream(&session, input_reliable_stream_buffer, BUFFER_SIZE, STREAM_HISTORY);
	
	uint8_t output_besteffort_stream_buffer[BUFFER_SIZE];
	uxrStreamId besteffort_out = uxr_create_output_best_effort_stream(&session, output_besteffort_stream_buffer,
	BUFFER_SIZE);

2	void create_publisher(
	uxrSession* session,
	uint16_t id)
	{
	uxrStreamId output = uxr_stream_id(0, UXR_BEST_EFFORT_STREAM, UXR_OUTPUT_STREAM);

3	void create_subscriber(
	uxrSession* session,
	uint16_t id)
	{
	uxrStreamId output = uxr_stream_id(0, UXR_BEST_EFFORT_STREAM, UXR_OUTPUT_STREAM);

4	void delete_publisher(
	uxrSession* session,
	uint16_t id)
	{
	uxrStreamId output = uxr_stream_id(0, UXR_BEST_EFFORT_STREAM, UXR_OUTPUT_STREAM);

5	void delete_subscriber(
	uxrSession* session,
	uint16_t id)
	{
	uxrStreamId output = uxr_stream_id(0, UXR_BEST_EFFORT_STREAM, UXR_OUTPUT_STREAM);

Şekil 4.15. ConfiguratorClient kodlarındaki en iyi çaba akış modu kod parçacıkları

Şekil 4.16'da en iyi çaba akış modunda çalışan bir yayımcının kaynak kodundaki kod parçacıkları listelenmiştir.

En İyi Çaba Akış Modu - Yayımcı	
No	**Kod Parçacıkları**

1	// Streams
	uint8_t output_reliable_stream_buffer[BUFFER_SIZE];
	uxrStreamId reliable_out = uxr_create_output_reliable_stream(&session, output_reliable_stream_buffer, BUFFER_SIZE,
	STREAM_HISTORY);
	
	uint8_t input_reliable_stream_buffer[BUFFER_SIZE];
	uxr_create_input_reliable_stream(&session, input_reliable_stream_buffer, BUFFER_SIZE, STREAM_HISTORY);
	
	uint8_t output_besteffort_stream_buffer[BUFFER_SIZE];
	uxrStreamId besteffort_out = uxr_create_output_best_effort_stream(&session, output_besteffort_stream_buffer,
	BUFFER_SIZE);

2	uxr_prepare_output_stream(&session, besteffort_out, datawriter_id, &ub, topic_size);

Şekil 4.16. Yayımcı kaynak kodlarındaki en iyi çaba akış modu kod parçacıkları

Şekil 4.17'de en iyi çaba akış modunda çalışan bir abonenin kaynak kodundaki kod parçacıkları listelenmiştir.

En İyi Çaba Akış Modu - Abone	
No	**Kod Parçacıkları**

1	// Streams
	uint8_t output_reliable_stream_buffer[BUFFER_SIZE];
	uxrStreamId reliable_out = uxr_create_output_reliable_stream(&session, output_reliable_stream_buffer, BUFFER_SIZE,
	STREAM_HISTORY);
	
	uint8_t input_reliable_stream_buffer[BUFFER_SIZE];
	uxr_create_input_reliable_stream(&session, input_reliable_stream_buffer, BUFFER_SIZE, STREAM_HISTORY);
	
	uxrStreamId besteffort_in = uxr_create_input_best_effort_stream(&session);

2	uint16_t read_data_req = uxr_buffer_request_data(&session, reliable_out, datareader_id, besteffort_in, &delivery_control);

Şekil 4.17. Abone kaynak kodlarındaki en iyi çaba akış modu kod parçacıkları

Şekil 4.18'de güvenilir akış modunda çalışan bir ConfiguratorClient'ın kaynak kodundan alınan kod parçacıkları verilmiştir.

Güvenilir Akış Modu - ConfiguratorClient	
No	**Kod Parçacıkları**

1	// Streams
	uint8_t output_reliable_stream_buffer[BUFFER_SIZE];
	uxr_create_output_reliable_stream(&session, output_reliable_stream_buffer, BUFFER_SIZE, STREAM_HISTORY);
	
	uint8_t input_reliable_stream_buffer[BUFFER_SIZE];
	uxr_create_input_reliable_stream(&session, input_reliable_stream_buffer, BUFFER_SIZE, STREAM_HISTORY);

2	void create_publisher(
	uxrSession* session,
	uint16_t id)
	{
	uxrStreamId output = uxr_stream_id(0, UXR_RELIABLE_STREAM, UXR_OUTPUT_STREAM);

3	void create_subscriber(
	uxrSession* session,
	uint16_t id)
	{
	uxrStreamId output = uxr_stream_id(0, UXR_RELIABLE_STREAM, UXR_OUTPUT_STREAM);

4	void delete_publisher(
	uxrSession* session,
	uint16_t id)
	{
	uxrStreamId output = uxr_stream_id(0, UXR_RELIABLE_STREAM, UXR_OUTPUT_STREAM);

5	void delete_subscriber(
	uxrSession* session,
	uint16_t id)
	{
	uxrStreamId output = uxr_stream_id(0, UXR_RELIABLE_STREAM, UXR_OUTPUT_STREAM);

Şekil 4.18. ConfiguratorClient kodlarındaki güvenilir akış modu kod parçacıkları

Şekil 4.15'te ve 4.18'de 2, 3, 4 ve 5 numaralı satırlardaki kod parçacıklarından akış moduyla ilgili olanlar, "uxrStreamId output = " şeklinde başlayarak devam eden kod parçacıklarıdır. Ancak ilgili kod parçacığı ConfiguratorClient kaynak kodunda dört farklı fonksiyonun içinde yer almaktadır. Bu nedenle, akış moduyla ilgili kod parçacığının yerini belirtmek amacıyla tablodaki gibi listelenmiştir. Şekil 4.19'da güvenilir akış modunda çalışan bir yayımcının kaynak kodundan alınan kod parçacıkları verilmiştir.

Güvenilir Akış Modu - Yayımcı	
No	**Kod Parçacıkları**

1	// Streams
	uint8_t output_reliable_stream_buffer[BUFFER_SIZE];
	uxrStreamId reliable_out = uxr_create_output_reliable_stream(&session, output_reliable_stream_buffer, BUFFER_SIZE,
	STREAM_HISTORY);
	
	uint8_t input_reliable_stream_buffer[BUFFER_SIZE];
	uxr_create_input_reliable_stream(&session, input_reliable_stream_buffer, BUFFER_SIZE, STREAM_HISTORY);

2	uxr_prepare_output_stream(&session, reliable_out, datawriter_id, &ub, topic_size);

Şekil 4.19. Yayımcı kaynak kodlarındaki güvenilir akış modu kod parçacıkları

Şekil 4.20'de güvenilir akış modunda çalışan bir abonenin kaynak kodunda kod parçacıkları verilmiştir.

Güvenilir Akış Modu - Abone	
No	**Kod Parçacıkları**
1	// Streams uint8_t output_reliable_stream_buffer[BUFFER_SIZE]; uxrStreamId reliable_out = uxr_create_output_reliable_stream(&session, output_reliable_stream_buffer, BUFFER_SIZE, STREAM_HISTORY); uint8_t input_reliable_stream_buffer[BUFFER_SIZE]; uxrStreamId reliable_in = uxr_create_input_reliable_stream(&session, input_reliable_stream_buffer, BUFFER_SIZE, STREAM_HISTORY);
2	uint16_t read_data_req = uxr_buffer_request_data(&session, reliable_out, datareader_id, reliable_in, &delivery_control);

Şekil 4.20. Abone kaynak kodlarındaki güvenilir akış modu kod parçacıkları

Kaynak kodlarda en iyi çaba akışı için kullanılan ConfiguratorClient için Şekil 4.15'teki, yayımcı için Şekil 4.16'daki ve abone için Şekil 4.17'deki kod parçacıkları; güvenilir akış için kullanılan ConfiguratorClient için Şekil 4.18'deki, yayımcı için Şekil 4.19'daki ve abone için Şekil 4.20'deki kod parçacıklarıyla değiştirildikten, derlendikten ve yüklendikten sonra ConfiguratorClient, yayımcı ve abone güvenilir akış modunda çalışacaktır.

4.1.5. DDS Çalışma Alanlarındaki Kaynak Kodların Düzenlenmesi

DDS çalışma alanlarının oluşturulmasında önerilen adımlar takip edilmiştir (URL-4). Talimatlar takip edildiğinde, HelloWorld konulu yayımcı ve abonenin yer aldığı "Fast-DDS-docs" adlı kütüphanede kaynak kodları örnek olarak sunulan "DDSHelloWorld" adlı çalışma alanı oluşturulmaktadır (URL-1). Diğer konulu çalışma alanları oluşturmak için takip edilmesi gereken iki talimat farklı şekilde yerine getirilmiştir (URL-4).

Fast-DDS-Gen aracı ve IDL dosyası kullanarak çalışma alanı oluşturma aşamasında; HelloWorld konulu IDL dosyasının kodları yerine Şekil 4.21'de belirtilen IDL kodları farklı konular oluşturmak için kullanılmıştır.

Farklı Konu Kod Parçacıkları	
Konu Adı	**Kod Parçacıkları**

İrtifa	// altitude.idl
	
	struct altitude
	{
	unsigned long index;
	string message;
	};

Nem	// humidity.idl
	
	struct humidity
	{
	unsigned long index;
	string message;
	};

Sıcaklık	// temperature.idl
	
	struct temperature
	{
	unsigned long index;
	string message;
	};

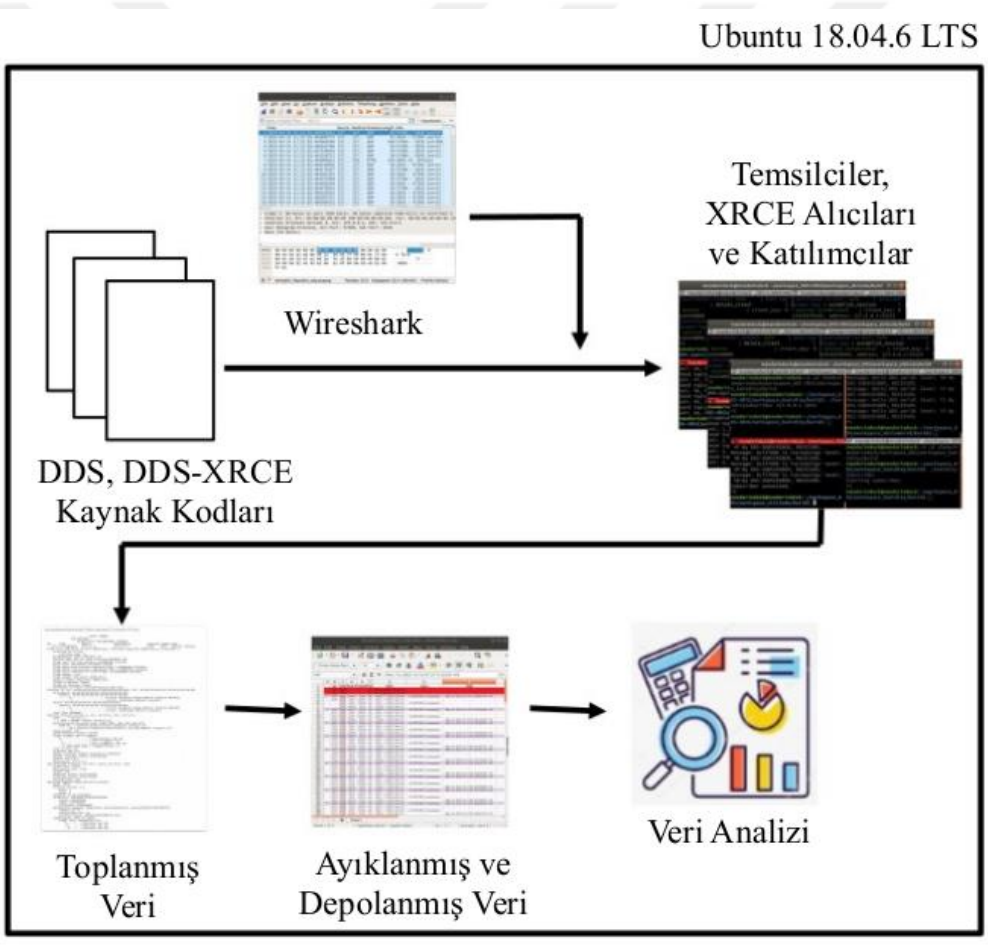
Basınç	// pressure.idl
	
	struct pressure
	{
	unsigned long index;
	string message;
	};

Şekil 4.21. Farklı konular için IDL dosya kodları

Talimatlarda belirtildiği üzere çalışma alanına yayımcı ve abone kaynak kod dosyaları sonradan eklenir. Eklenen bu yayımcı ve aboneler HelloWorld konusuna göre çalışmak isteyecektir ve hataya neden olacaktır. Bu nedenle yayımcı ve abone kaynak kodlarındaki HelloWorld konusuyla ilişkili kod parçacıklarının IDL belgesinde belirtilen konuya göre çalışacak şekilde değiştirilmesiyle yayımcı ve abone istenen şekilde çalışacaktır.

4.2. Simülasyon ve Veri Toplama Aşaması

Toplamda on iki farklı senaryo bulunmaktadır ve bütün yazılım ürünleri çalıştırmak için hazırdır. Şekil 4.22 simülasyon adımlarını göstermektedir. Bütün deneyler Ubuntu 18.04.6 LTS tarafından işletilen bir bilgisayarda gerçekleştirilmiştir.



Şekil 4.22. Simülasyon adımları

Simülasyonun ilk adımında yayımcıların, abonelerin, temsilcilerin ve DDS katılımcılarının kaynak kodlarının dosya konumunu çoklu Gnome Terminator pencereleri gösterecek şekilde ayarlanmıştır ve Wireshark ile bütün kaynak kodları çalıştıracak ilgili

kodlar pencerelere yazılmıştır. İkinci adımda, Wireshark ağdaki mesajları izleyebilmek için komut satısı aracılığıyla çalıştırılmış ve analiz etmek için ağ mesajlarını kaydetmek amacıyla arayüzü aracılığıyla konfigüre edilmiştir. Üçüncü adıma gelindiğinde temsilciler, odaklanılmayan XRCE alıcıları ve DDS katılımcıları sırasıyla çalıştırılmıştır. Dördüncü adımda, birbirleriyle iletişmine odaklanılan abone ve yayımcı sırasıyla komut satırı aracılığıyla çalıştırılmıştır. Beşinci adımda yeterli sayıda mesaj gönderiminden sonra yayımcı ve abone komut satırı aracılığıyla sırasıyla durdurulmuştur. Gönderilecek mesajlar yeterli olduğuna karar vermek için 50, 100 ve 150 paket olarak sınırlandırılmıştır. Son adım olarak, simülasyon verileri analiz etmek için kaydedilmiştir. Simülasyon ve veri toplama aşaması sonunda Tablo 4.4'te görüldüğü gibi otuz altı test testin sonuçları elde edilmiştir.

Tablo 4.4. Yapılan testler ve sonlandırılma limitleri

Test Koşulları	**Testler ve Limitleri (paket)**								
	Dağıtım Modeli-1			**Dağıtım Modeli-2**			**Dağıtım Modeli-3**		
UDP-Güvenilir	50	100	150	50	100	150	50	100	150
UDP-EnİyiÇaba	50	100	150	50	100	150	50	100	150
TCP-Güvenilir	50	100	150	50	100	150	50	100	150
TCP-EnİyiÇaba	50	100	150	50	100	150	50	100	150

4.3. Analiz Aşaması

Gecikme, bir uçtan diğer uca bir miktar verinin iletilmesi için harcanmaya ihtiyaç duyulan zaman olarak tanımlanmaktadır. Bu yüzden, gecikmeyi ölçmek için her mesaj için iletimin ne zaman başladığı ve bittiğinin bilinmesi gerekmektedir. Ağ trafiğini izlemek için Wireshark kullanılmıştır ve ağdaki paketlerin zaman damgası ve boyutu bilinerek izi sürülebilmiştir.

Ancak, Wireshark çeşitli formatlarda zaman damgası olarak her paketin gönderilişinin başlangıç zamanını eklemektedir. Yine de, gecikmeyi ölçmek için bitiş zamanı bilinmelidir. Bir mesaj başarılı şekilde iletildiğinde, güvenilir akışların bir işlevi olarak alıcı göndericiye onaylama mesajı geri gönderir. Onaylama mesajı gönderilmesi için, öncesinde başka bir mesajın başarılı şekilde iletilmesi gerekmektedir. Bu gerçeğe dayanaraktan, farz edilebilir ki veri transferinin bitiş zamanı olarak alıcının göndericiye

geri gönderdiği onaylama mesajının zaman damgası kullanılabilir. En iyi çaba akışı kullanıyor olmasına rağmen TCP-EnİyiÇaba senaryolarında da onaylama mesajlarının varlığı gözlenmiştir. Alınan mesajın zaman damgası, gönderilmiş mesajın zaman damgasına göre geleceği ifade etmektedir. Alınan mesajın zamanı ile gönderilen mesajın zamanı arasındaki süre hesaplanmıştır. Her süre değeri paketin gecikmesini temsil etmektedir. Bir paketin gecikmesi formülü Denklem 4.1'deki gibi ifade edilmiştir.

$$L = T_{CM} - T_{DP} \tag{4.1}$$

T_{CM}, alıcının göndericiye geri gönderdiği onaylama mesajının zamanıdır. T_{DP}, başlangıç olarak gönderilen veri paketinin zamanıdır ve L, Denklem 4.1'de verilen T_{CM} ve T_{DP} arasındaki fark olan paketin milisaniye cinsinden gecikmesidir. Buna rağmen, test edilmesi gereken UDP-EnİyiÇaba senaryoları bulunmaktadır. Simülasyon sırasında, mesajların bitiş zamanını belirlemek için UDP-EnİyiÇaba senaryolarının herhangi bir onaylama mesajının varlığı gözlenmemiştir. UDP-EnİyiÇaba testlerinin işlev farklılığı olarak hazırlık aşamasında anlatılan onaylama mesajlarının olmayışıdır. Buna çözüm olarak, başlangıç olarak yayımcıların ve abonelerin kaynak kodları analiz edilmiştir ve belirli satırların arasına bazı kodlar eklenmiştir. Çalıştırıldığında, kod Epoch'tan beri saniyeler ve saniyeden daha küçük değerleri geri döndürmektedir.

Yayımcı kaynak kodu için, veri mesajını gönderen kod satırının hemen öncesine anlık zamanı hesaplayan ekstra kod yazılmıştır ve önceden hesaplanmış olan zamanı geri döndüren ekstra kod, veri mesajını gönderen kod satırından hemen sonra yazılmıştır. Alıcı kaynak kodu için; anlık zamanı hesaplayan ve geri döndüren ekstra kod, abonenin aldığı mesajı yayımlayan kod satırından hemen sonra yazılmıştır. Terminal penceresinde, veri mesajlarının ilk olarak gelecek şekilde ve veri mesajlarının zaman damgalarının ikinci olarak gelecek şekilde görünmesi amaçlanmıştır.

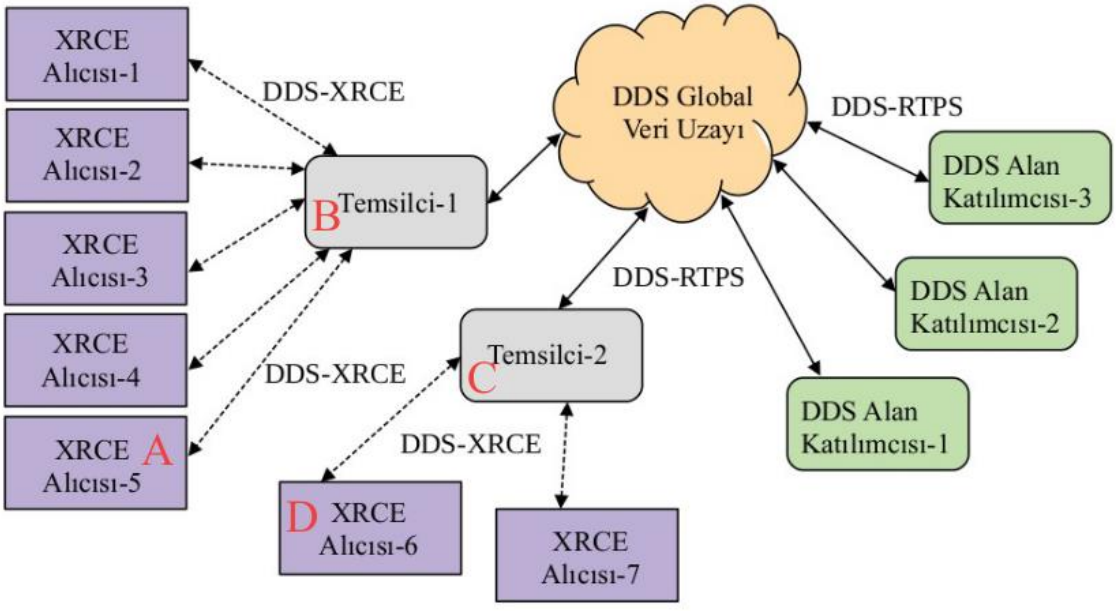
Yine, bir zaman damgası diğerine göre geleceği göstermektedir. Gönderilen ve alınan mesajların arasındaki süre, iletilmiş bütün paketler için hesaplanmıştır. Böylelikle, UDP-EnİyiÇaba senaryolarının her mesajının gecikmesi elde edilmiştir.

Verim, belirli bir süre için ağda bir noktadan diğerine ne kadar verinin başarılı bir şekilde iletildiği sorusuna cevap verir. Saniye başına bit (bits per second, bps) ile ölçülmektedir.

Ancak, verim ve bant genişliğine gelindiğinde bir karışıklık yaşanabilmektedir. Bant genişliği maksimum verime karşılık gelmektedir.

Bütün senaryoların deneyleri sırasında Wireshark ağ trafiğini izlemiştir. Böylelikle, her mesaj için paket boyutu Wireshark tarafından sağlanmıştır. Ancak, protokol dönüşümleri sebebiyle bir veri paketinin boyutu iletim yolu üzerinde birkaç defa değişebilmektedir. Şekil 4.23, odaklanılan iletim yolu üzerindeki bazı noktalarla birlikte Dağıtım Modeli-2'yi ve Dağıtım Modeli-2'nin TCP-Güvenilir-50p koşullarında iletilen otuz dördüncü paketin bilgisini göstermektedir.

DM-2 TCP-GÜVENİLİR 34/50			
PROTOKOL	KAYNAK	HEDEF	BOYUT
TCP	A	B	185
TCP	B	A	66
RTPS	B	C	166
TCP	C	D	110
TCP	D	C	66



Şekil 4.23. Yayımlanan bir veri paketinin protokol ve boyut bilgileri

Şekil 4.23'teki tablodanun ilk satırı, DDS-XRCE protokolü üzerinden A noktasından B noktasına iletilen paketin bilgisidir. Tablodaki ikinci satır, B'den A'ya geri gönderilen onay mesajının bilgisidir. Tablodaki üçüncü satır, RTPS protokolü üzerinden B noktasından C noktasına iletilen paketin bilgisidir. Tablodaki dördüncü satır, DDS-XRCE

protokolü üzerinden C noktasından D noktasına iletilen paketin bilgisidir. Tablodaki beşinci satır, göndericiyi başarılı teslimat hakkında bilgilendiren onay mesajının bilgisidir. A noktası iletimin başladığı yerdir. B noktası DDS-XRCE protokolünün RTPS protokolüne dönüştürüldüğü yerdir. C noktası RTPS protokolünün DDS-XRCE protokolüne dönüştürüldüğü yerdir. D noktası iletimin bittiği yerdir.

Tabloda "boyut" sütunundaki bilgiler, sırasıyla A noktasından B, C ve D noktasına iletilen paketin "byte" cinsinden boyutlarını göstermektedir. Boyuttaki değişiklik protokol dönüşümleri sebebiyle meydana gelmektedir. Verimi hesaplayabilmek için, alıcı ve verici arasında başarılı teslimat gerçekleşmeli ve paketin boyutu bilinmelidir. Bu yüzden verimin hesaplanması, belirli bir sürede uzunluğu bilinen bir yolda bir noktadan diğerine giden bir aracın hızının hesaplanması ile ilişkilendirilebilir. İletilen paketin gecikme değeri, bir noktadan diğerine giden aracın harcadığı zaman olarak düşünülebilir. Toplam paket boyutu, aracın gittiği toplam yol uzunluğu olarak düşünülebilir. Verim formülü aşağıdaki Denklem 4.2'deki gibi ifade edilmiştir.

$$Th = \sum_{n=1}^{N} \left(\frac{S_{AB_n} + S_{BC_n} + S_{CD_n}}{L} \right) \quad (4.2)$$

Verimin hesaplanması için paketin toplam boyutu, protokol dönüşümleri arasındaki paket boyutlarının tamamının toplanmasıyla elde edilmiştir. S_{AB}, A'dan B'ye iletilen paketin "bit" cinsinden boyutudur. S_{BC}, B'den C'ye iletilen paketin "bit" cinsinden boyutudur. S_{CD}, C'den D'ye iletilen paketin "bit" cinsinden boyutudur. "n", yan altına yazıldığı ifadenin hangi başarılı teslimata ait olduğunu belirtmektedir. L, paketin "saniye" cinsinden gecikmesidir ve "bps" cinsinden bir değer elde etmek için paketin toplam boyutu ilgili paketin gecikmesine bölünmektedir. Odaklanılan iletim yolundaki başarılı olarak iletilen her paket için bps değerleri değerleri hesaplanmıştır. Bu yüzden, N iletim yolundaki başarılı teslimat sayısını ifade etmektedir. Sonuç olarak, iletim yolundaki paketlerin bps değerlerinin toplamı Denklem 4.2'deki Th'nin gösterdiği gibi toplam verim değerini vermektedir.

5. PERFORMANS VE TEST SONUÇLARI

Paketlerin gecikmesi, boyutu, gönderilen paket sayısı ve alınan paket sayısı ölçülmüştür. Diğer ölçütler deneylerden elde edilen bilgilere göre hesaplanmıştır. Tablolardaki "Minimum", ilgili ölçümün minimum değerini ifade etmektedir. "Maksimum", ilgili ölçümün maksimum değerini ifade etmektedir. "Ortalama", ilgili ölçümün ortalama değerini ifade etmektedir. "Std. Sapma", ilgili ölçümün standart sapmasını ifade etmektedir. "Varyans", ilgili ölçümün varyansını ifade etmektedir. Tablolardaki "Toplam", başarılı teslimatların toplam paket boyutunu ifade etmektedir. Tablolardaki "Gönderim sayısı", gönderilen paketlerin sayısını ifade etmektedir. Tablolardaki "Alım sayısı", alınan paketlerin sayısını ifade etmektedir. Diğer bir deyişle, başarılı teslimat sayısını ifade etmektedir. Tablolardaki UDP/TCP'ler ilgili test sırasında kullanılan taşıma protokolünü belirtmektedir. Güvenilir/EnİyiÇaba ise ilgili test sırasında kullanılan akış modunu temsil etmektedir. Buradan hareketle; koşulu UDP-Güvenilir olan bir testte, UDP taşıma protokolü kullanılarak ve Güvenili akış modunda çalışarak mesaj iletimi sağlanmaktadır.

Dağıtım Modeli-1'in gecikme sonuçları Tablo 5.1'de verilmiştir.

Tablo 5.1. Dağıtım Modeli-1'in gecikme sonuçları

Koşullar	**Kriterler**					
	Limit (paket)	**Minimum (ms)**	**Maksimum (ms)**	**Ortalama (ms)**	**Std. Sapma (ms)**	**Varyans (ms2)**
UDP-Güvenilir	50	0,186	0,651	0,406	0,082	0,007
UDP-Güvenilir	100	0,151	0,674	0,377	0,095	0,009
UDP-Güvenilir	150	0,172	0,810	0,398	0,108	0,012
TCP-Güvenilir	50	0,135	47,726	14,026	20,187	407,527
TCP-Güvenilir	100	0,155	48,310	14,619	20,472	419,111
TCP-Güvenilir	150	0,170	47,935	14,281	20,080	403,194
UDP-EnİyiÇaba	50	0,134	0,594	0,352	0,092	0,009
UDP-EnİyiÇaba	100	0,148	0,705	0,358	0,079	0,006
UDP-EnİyiÇaba	150	0,114	0,448	0,262	0,060	0,004
TCP-EnİyiÇaba	50	0,152	0,743	0,333	0,096	0,009
TCP-EnİyiÇaba	100	0,136	0,634	0,363	0,077	0,006
TCP-EnİyiÇaba	150	0,148	0,728	0,358	0,090	0,008

UDP-EnİyiÇaba-150p koşullarında gerçekleştirilen test, tüm testlerin içinde en düşük ortalama gecikmeye sahiptir. Yüz elli paket gönderimiyle sınırlandırılmış testler arasında muhtemelen en hızlısı olarak yorumlanmaktadır. TCP-Güvenilir-100p koşullarında gerçekleştirilen test, tüm testlerin içinde en yüksek ortalama gecikmeye sahiptir. Yüz paket gönderimiyle sınırlandırılmış testler arasında muhtemelen en yavaşı olarak yorumlanmaktadır. En düşük minimum gecikme, UDP-EnİyiÇaba-150p koşullarında gerçekleştirilen test sırasında gözlenmiştir. En yüksek maksimum gecikme, TCP-Güvenilir-100p koşullarında gerçekleştirilen test sırasında gözlenmiştir.

Dağıtım Modeli-2'nin gecikme sonuçları Tablo 5.2'de verilmiştir.

Tablo 5.2. Dağıtım Modeli-2'nin gecikme sonuçları

Koşullar	**Kriterler**					
	Limit (paket)	**Minimum (ms)**	**Maksimum (ms)**	**Ortalama (ms)**	**Std. Sapma (ms)**	**Varyans (ms2)**
UDP-Güvenilir	50	0,387	0,873	0,656	0,111	0,012
UDP-Güvenilir	100	0,301	1,175	0,615	0,135	0,018
UDP-Güvenilir	150	0,274	0,902	0,617	0,128	0,016
TCP-Güvenilir	50	0,249	48,129	14,792	20,985	440,379
TCP-Güvenilir	100	0,248	48,427	15,002	20,537	421,788
TCP-Güvenilir	150	0,238	48,800	14,689	20,435	417,588
UDP-EnİyiÇaba	50	0,272	1,467	0,558	0,188	0,035
UDP-EnİyiÇaba	100	0,318	2,661	0,731	0,242	0,058
UDP-EnİyiÇaba	150	0,248	1,385	0,573	0,193	0,037
TCP-EnİyiÇaba	50	0,251	0,630	0,404	0,085	0,007
TCP-EnİyiÇaba	100	0,272	0,917	0,570	0,120	0,014
TCP-EnİyiÇaba	150	0,236	1,124	0,555	0,136	0,019

TCP-EnİyiÇaba-50p koşullarında gerçekleştirilen test, tüm testlerin içinde en düşük ortalama gecikmeye sahiptir. Elli paket gönderimiyle sınırlandırılmış testler arasında muhtemelen en hızlısı olarak yorumlanmaktadır. TCP-Güvenilir-100p koşullarında gerçekleştirilen test, tüm testlerin içinde en yüksek ortalama gecikmeye sahiptir. Yüz paket gönderimiyle sınırlandırılmış testler arasında muhtemelen en yavaşı olarak yorumlanmaktadır.

En düşük minimum gecikme, TCP-EnİyiÇaba-150p koşullarında gerçekleştirilen test sırasında gözlenmiştir. En yüksek maksimum gecikme, TCP-Güvenilir-150p koşullarında gerçekleştirilen test sırasında gözlenmiştir.

Dağıtım Modeli-3'ün gecikme sonuçları Tablo 5.3'te verilmiştir. TCP-EnİyiÇaba-150p koşullarında gerçekleştirilen test, tüm testlerin içinde en düşük ortalama gecikmeye sahiptir. Yüz elli paket gönderimiyle sınırlandırılmış testler arasında muhtemelen en hızlısı olarak yorumlanmaktadır. TCP-Güvenilir-100p koşullarında gerçekleştirilen test, tüm testlerin içinde en yüksek ortalama gecikmeye sahiptir. Yüz paket gönderimiyle sınırlandırılmış testler arasında muhtemelen en yavaşı olarak yorumlanmaktadır. En düşük minimum gecikme, TCP-EnİyiÇaba-50p koşullarında gerçekleştirilen test sırasında gözlenmiştir. En yüksek maksimum gecikme, TCP-Güvenilir-100p koşullarında gerçekleştirilen test sırasında gözlenmiştir.

Tablo 5.3. Dağıtım Modeli-3'ün gecikme sonuçları

Koşullar	**Kriterler**					
	Limit (paket)	**Minimum (ms)**	**Maksimum (ms)**	**Ortalama (ms)**	**Std. Sapma (ms)**	**Varyans (ms2)**
UDP-Güvenilir	50	0,261	1,440	0,576	0,200	0,040
UDP-Güvenilir	100	0,252	0,846	0,571	0,115	0,013
UDP-Güvenilir	150	0,252	0,890	0,579	0,111	0,012
TCP-Güvenilir	50	0,242	48,112	15,058	21,105	445,402
TCP-Güvenilir	100	0,238	48,589	15,274	20,885	436,164
TCP-Güvenilir	150	0,253	47,084	14,584	20,099	403,954
UDP-EnİyiÇaba	50	0,303	0,808	0,628	0,133	0,018
UDP-EnİyiÇaba	100	0,256	0,807	0,606	0,148	0,022
UDP-EnİyiÇaba	150	0,254	1,469	0,550	0,183	0,033
TCP-EnİyiÇaba	50	0,167	0,682	0,450	0,106	0,011
TCP-EnİyiÇaba	100	0,201	0,761	0,493	0,100	0,010
TCP-EnİyiÇaba	150	0,169	0,983	0,444	0,128	0,016

Dağıtım Modeli-1'in verim sonuçları Tablo 5.4'te verilmiştir. TCP-Güvenilir-150p koşullarında gerçekleştirilen test, tüm testlerin içinde en düşük ortalama verime sahiptir. Yüz elli paket gönderimiyle sınırlandırılmış testler arasında periyodik olarak muhtemelen en az veri ileten olarak yorumlanmaktadır.

UDP-EnİyiÇaba-150p koşullarında gerçekleştirilen test, tüm testlerin içinde en yüksek ortalama verime sahiptir. Yüz elli paket gönderimiyle sınırlandırılmış testler arasında periyodik olarak muhtemelen en çok veri ileten olarak yorumlanmaktadır. En düşük minimum verim, TCP-Güvenilir-100p koşullarında gerçekleştirilen test sırasında gözlenmiştir. En yüksek maksimum verim, UDP-EnİyiÇaba-150p koşullarında gerçekleştirilen test sırasında gözlenmiştir.

Tablo 5.4. Dağıtım Modeli-1'in verim sonuçları

Koşullar	**Kriterler**					
	Limit (paket)	**Minimum (Mbps)**	**Maksimum (Mbps)**	**Ortalama (Mbps)**	**Std. Sapma (Mbps)**	**Varyans (Mbps2)**
UDP-Güvenilir	50	3,294	9,997	5,436	1,234	1,523
UDP-Güvenilir	100	3,181	14,233	6,115	1,901	3,613
UDP-Güvenilir	150	2,648	12,447	5,897	2,010	4,039
TCP-Güvenilir	50	0,051	17,357	5,441	4,683	21,931
TCP-Güvenilir	100	0,048	15,117	4,744	3,772	14,232
TCP-Güvenilir	150	0,049	13,771	4,636	3,578	12,803
UDP-EnİyiÇaba	50	3,608	16,027	6,602	2,213	4,896
UDP-EnİyiÇaba	100	3,040	14,506	6,353	1,839	3,383
UDP-EnİyiÇaba	150	4,782	18,885	8,638	2,130	4,537
TCP-EnİyiÇaba	50	3,146	15,326	7,615	2,365	5,592
TCP-EnİyiÇaba	100	3,686	17,225	6,796	1,948	3,794
TCP-EnİyiÇaba	150	3,210	15,747	7,062	2,361	5,575

Dağıtım Modeli-2'nin verim sonuçları Tablo 5.5'te verilmiştir. TCP-Güvenilir-100p koşullarında gerçekleştirilen test, tüm testlerin içinde en düşük ortalama verime sahiptir. Yüz paket gönderimiyle sınırlandırılmış testler arasında periyodik olarak muhtemelen en az veri ileten olarak yorumlanmaktadır.

TCP-EnİyiÇaba-50p koşullarında gerçekleştirilen test, tüm testlerin içinde en yüksek ortalama verime sahiptir. Elli paket gönderimiyle sınırlandırılmış testler arasında periyodik olarak muhtemelen en çok veri ileten olarak yorumlanmaktadır. En düşük minimum verim, TCP-Güvenilir-50p koşullarında gerçekleştirilen test sırasında gözlenmiştir. En yüksek maksimum verim, TCP-Güvenilir-100p koşullarında gerçekleştirilen test sırasında gözlenmiştir.

Tablo 5.5. Dağıtım Modeli-2'nin verim sonuçları

Koşullar	Kriterler					
	Limit (paket)	**Minimum (Mbps)**	**Maksimum (Mbps)**	**Ortalama (Mbps)**	**Std. Sapma (Mbps)**	**Varyans (Mbps2)**
UDP-Güvenilir	50	3,096	6,992	4,263	0,874	0,765
UDP-Güvenilir	100	2,300	8,997	4,683	1,377	1,897
UDP-Güvenilir	150	2,998	9,869	4,651	1,372	1,881
TCP-Güvenilir	50	0,077	12,426	4,259	3,601	12,964
TCP-Güvenilir	100	0,064	12,462	4,220	3,319	11,016
TCP-Güvenilir	150	0,063	12,999	4,554	3,581	12,821
UDP-EnİyiÇaba	50	1,843	9,947	5,247	1,381	1,908
UDP-EnİyiÇaba	100	1,016	8,491	3,988	1,188	1,411
UDP-EnİyiÇaba	150	1,952	10,883	5,248	1,718	2,951
TCP-EnİyiÇaba	50	4,901	12,295	7,995	1,795	3,221
TCP-EnİyiÇaba	100	3,368	11,368	5,693	1,430	2,045
TCP-EnİyiÇaba	150	2,747	13,062	5,999	1,919	3,681

Dağıtım Modeli-3'ün verim sonuçları Tablo 5.6'da verilmiştir.

Tablo 5.6. Dağıtım Modeli-3'ün verim sonuçları

Koşullar	Kriterler					
	Limit (paket)	**Minimum (Mbps)**	**Maksimum (Mbps)**	**Ortalama (Mbps)**	**Std. Sapma (Mbps)**	**Varyans (Mbps2)**
UDP-Güvenilir	50	0,956	5,264	2,680	0,987	0,974
UDP-Güvenilir	100	1,626	5,469	2,541	0,700	0,490
UDP-Güvenilir	150	1,547	5,459	2,494	0,640	0,409
TCP-Güvenilir	50	0,049	7,262	2,458	1,999	3,996
TCP-Güvenilir	100	0,036	7,403	2,431	1,890	3,573
TCP-Güvenilir	150	0,037	8,070	2,610	2,144	4,599
UDP-EnİyiÇaba	50	1,704	4,544	2,335	0,701	0,492
UDP-EnİyiÇaba	100	1,705	5,368	2,468	0,859	0,738
UDP-EnİyiÇaba	150	0,937	5,425	2,773	0,917	0,841
TCP-EnİyiÇaba	50	2,580	10,543	4,233	1,482	2,197
TCP-EnİyiÇaba	100	2,311	8,753	3,810	1,249	1,561
TCP-EnİyiÇaba	150	1,791	10,445	4,371	1,544	2,383

UDP-EnİyiÇaba-50p koşullarında gerçekleştirilen test, tüm testlerin içinde en düşük ortalama verime sahiptir. Elli paket gönderimiyle sınırlandırılmış testler arasında periyodik olarak muhtemelen en az veri ileten olarak yorumlanmaktadır. TCP-EnİyiÇaba-150p koşullarında gerçekleştirilen test, tüm testlerin içinde en yüksek ortalama verime sahiptir. Yüz elli paket gönderimiyle sınırlandırılmış testler arasında periyodik olarak muhtemelen en çok veri ileten olarak yorumlanmaktadır. En düşük minimum verim, TCP-Güvenilir-100p koşullarında gerçekleştirilen test sırasında gözlenmiştir. En yüksek maksimum verim, TCP-EnİyiÇaba-50p koşullarında gerçekleştirilen test sırasında gözlenmiştir.

Dağıtım Modeli-1'in veri paketlerine dair sonuçları Tablo 5.7'de verilmiştir. UDP-Güvenilir ve UDP-EnİyiÇaba koşullarında gerçekleştirilen testler, tüm testlerin içinde en düşük ortalama veri paketine sahiptir. Kendi paket limitine sahip testler içinde en az veri ileten olarak yorumlanmaktadır. TCP-Güvenilir-50p koşullarında gerçekleştirilen test, tüm testlerin içinde en yüksek ortalama veri paketine sahiptir. Elli paket gönderimiyle sınırlandırılmış testler arasında en çok veri ileten olarak yorumlanmaktadır. Dağıtım Modeli-1'in testleri sırasında herhangi bir paket kaybı gözlenmemiştir.

Tablo 5.7. Dağıtım Modeli-1'in paket kaybı sonuçları

Koşullar	**Kriterler**				
	Limit (paket)	**Ortalama (bit)**	**Toplam (bit)**	**Gönderim sayısı**	**Alım sayısı**
UDP-Güvenilir	50	2144	107200	50	50
UDP-Güvenilir	100	2144	214400	100	100
UDP-Güvenilir	150	2144	321600	150	150
TCP-Güvenilir	50	2357,6	117880	50	50
TCP-Güvenilir	100	2336	233600	100	100
TCP-Güvenilir	150	2345,6	351840	150	150
UDP-EnİyiÇaba	50	2144	107200	50	50
UDP-EnİyiÇaba	100	2144	214400	100	100
UDP-EnİyiÇaba	150	2144	321600	150	150
TCP-EnİyiÇaba	50	2336	116800	50	50
TCP-EnİyiÇaba	100	2336	233600	100	100
TCP-EnİyiÇaba	150	2336	350400	150	150

Dağıtım Modeli-2'nin veri paketlerine dair sonuçları Tablo 5.8'de verilmiştir. UDP-Güvenilir-50p, UDP-EnİyiÇaba-50p ve UDP-EnİyiÇaba-100p koşullarında gerçekleştirilen testler, tüm testlerin içinde en düşük ortalama veri paketine sahiptir.

Kendi paket limitine sahip testler içinde en az veri ileten olarak yorumlanmaktadır. TCP-Güvenilir-100p koşullarında gerçekleştirilen test, tüm testlerin içinde en yüksek ortalama veri paketine sahiptir. Yüz paket gönderimiyle sınırlandırılmış testler arasında en çok veri ileten olarak yorumlanmaktadır. TCP-Güvenilir-150p koşullarında gerçekleştirilen test, tüm testler içinde en yüksek paket kaybına sahiptir. Diğer testlerin paket kaybı miktarları sıfır ile iki arasında değişmektedir.

Tablo 5.8. Dağıtım Modeli-2'nin paket kaybı sonuçları

Koşullar	**Kriterler**				
	Limit (paket)	**Ortalama (bit)**	**Toplam (bit)**	**Gönderim sayısı**	**Alım sayısı**
UDP-Güvenilir	50	2704	132496	50	49
UDP-Güvenilir	100	2706,6	267952	100	99
UDP-Güvenilir	150	2704,9	400320	150	148
TCP-Güvenilir	50	3287,2	164360	50	50
TCP-Güvenilir	100	3296,5	326352	100	99
TCP-Güvenilir	150	3296,2	484536	150	147
UDP-EnİyiÇaba	50	2704	132496	50	49
UDP-EnİyiÇaba	100	2704	267696	100	99
UDP-EnİyiÇaba	150	2706	403200	150	149
TCP-EnİyiÇaba	50	3088	151312	50	49
TCP-EnİyiÇaba	100	3088	305712	100	99
TCP-EnİyiÇaba	150	3088	460112	150	149

Dağıtım Modeli-3'ün veri paketlerine dair sonuçları Tablo 5.9'da verilmiştir.

Tablo 5.9. Dağıtım Modeli-3'ün paket kaybı sonuçları

Koşullar	**Kriterler**				
	Limit (paket)	**Ortalama (bit)**	**Toplam (bit)**	**Gönderim sayısı**	**Alım sayısı**
UDP-Güvenilir	50	1376	67424	50	49
UDP-Güvenilir	100	1376	136224	100	99
UDP-Güvenilir	150	1376	205024	150	149
TCP-Güvenilir	50	1963,3	96200	50	49
TCP-Güvenilir	100	1961,2	194160	100	99
TCP-Güvenilir	150	1970,2	293560	150	149
UDP-EnİyiÇaba	50	1376	67424	50	49
UDP-EnİyiÇaba	100	1376	136224	100	99
UDP-EnİyiÇaba	150	1376	205024	150	149
TCP-EnİyiÇaba	50	1760	86240	50	49
TCP-EnİyiÇaba	100	1760	174240	100	99
TCP-EnİyiÇaba	150	1760	262240	150	149

UDP-Güvenilir ve UDP-EnİyiÇaba koşullarında gerçekleştirilen testler, tüm testlerin içinde en düşük ortalama veri paketine sahiptir. Kendi paket limitine sahip testler içinde en az veri ileten olarak yorumlanmaktadır. TCP-Güvenilir-150p koşullarında gerçekleştirilen test, tüm testlerin içinde en yüksek ortalama veri paketine sahiptir. Yüz elli paket gönderimiyle sınırlandırılmış testler arasında en çok veri ileten olarak yorumlanmaktadır. Dağıtım Modeli-3'ün bütün testleri bir paket kaybı ile sonuçlanmıştır.

Referansta, TCP-EnİyiÇaba akışlarının UDP-Güvenilir akışlarına benzer davranış sergilediği belirtilmiştir (URL-3). Bu yüzden, örneğin, farklı taşıma profili-akış modu çiftlerinde dağıtım modellerinin gecikme seviyelerine bakıldığında, ilgili dağıtım modellerinin TCP-EnİyiÇaba ve UDP-Güvenilir çiftlerin yaklaşık ölçümleri sonuçların doğruluğunu desteklemektedir.

6. SONUÇLAR VE ÖNERİLER

Bazı deneylerden sonra, XRCE alıcılarının, temsilcilerin ve DDS alan katılımcılarının davranışları gözlemlenerek ve deneylerden elde edilen veriler analiz edilerek birkaç çıkarımda bulunulmuştur.

Dağıtım Modeli-1'in odaklanılan iletim yolu düşünüldüğünde, iletim yönü sebebiyle abonelik açısından test edilmiştir. Çünkü Dağıtım Modeli-1'in odaklanılan iletim yolundaki XRCE alıcısı bir abonedir. Başarılı teslimat düşünüldüğünde, Dağıtım Modeli-1'in bütün deneyleri için her teslimat başarılı olarak tamamlanmıştır. Bu yüzden ilk çıkarım için denilebilir ki paketler hedefe herhangi bir sorunla karşılaşmadan ve kayıp yaşanmadan ulaşmıştır.

İkinci çıkarım için Dağıtım Modeli-2'nin odaklanılan iletim yolu hatırlanmalıdır. İletim yolu iki uç açısından da aynıdır. XRCE alıcıları arasındaki bağlantı Dağıtım Modeli-2'nin düzeni sebebiyle simetriktir ve DDS-XRCE'nin hem yayım hem de abonelik açısından test edilmesi için fırsat sağlamaktadır. Paketleri gönderen bir XRCE alıcısının ve onları alan başka bir XRCE alıcısının var olduğu anlamına gelmektedir. Dağıtım Modeli-2'nin deneylerinde bütün paketler yayımcı XRCE alıcısı tarafından gönderilmiş olmasına rağmen, başarılı teslimat sayısına bakıldığında, abone tarafından birkaç paket alınmamıştır. Mesajlar incelendiğinde, kayıp paketler yayımcı XRCE alıcısına bağlı temsilci tarafından alındıktan sonra kaybolmuştur. Bu yüzden Dağıtım Modeli-1'in deneylerinden elde edilen ilk çıkarım, aynı zamanda Dağıtım Modeli-2'nin deneyleri için de geçerlidir. Çünkü eğer yayımcı tarafı bütün paketleri gönderebilseydi, abone tarafı bütün paketleri alabilecekti. Dahası, kaybolan paketler her zaman gönderilen ilk üç paket arasındandır. Sanki temsilci, ilk birkaç paketi aldığı anda onları göndermek zorunda olduğunu fark etmemekte gibi görünmektedir. Göndermeye başladığında ise, temsilci kalan paketlerin hepsini göndermektedir. Bu yüzden ikinci çıkarım için denilebilir ki yayımcı XRCE alıcısına bağlı temsilcinin ilk birkaç paketi göndermemesi ihtimal dahilindedir.

Dahası ağ mesajları incelenerek Dağıtım Modeli-2'nin deneyleri sırasında bir etkileşim düzeni fark edilmiştir ve mesajlarına odaklanılan konu ile ilgilidir. Merkezde DDS Global Veri Uzayı düşünüldüğünde, mesajlarına odaklanılan konu ile ilgilenmeleri sebebiyle

DDS Global Veri Uzayı üzerinden bağlanmış bir DDS alan katılımcısı ve iki temsilci bulunmaktadır. DDS alan katılımcısı, Dağıtım Modeli-2'nin deneylerine abone olarak DDS Global Veri Uzayı'nda dolaşan konu mesajlarını gözlemlemek için eklenmiştir. Ağ mesajlarının incelenmesinden sonra; yayımcı XRCE alıcısına bağlı temsilcinin, DDS Global Veri Uzayı üzerinden iki farklı DDS alan katılımcısına iki farklı mesaj serisi halinde veri paketleri gönderdiği gözlemlenmiştir. Sonraki çıkarım için hatırlanması önem arz etmektedir.

Birinci ve ikinci çıkarımlar ayrıca Dağıtım Modeli-3 için de geçerlidir. Ancak, iletim yolunun nasıl ilerlediği belirsizdir. Gözlem için Dağıtım Modeli-3'ün deneylerine abone olarak bir DDS alan katılımcısı eklenmiştir. Ağ mesajları incelendiği sırada, ilgili konunun verilerini taşıyan ve verileri RTPS protokolü kullanarak taşıyan sadece bir mesaj serisinin bulunduğu fark edilmiştir. Ancak, iki abone sebebiyle Dağıtım Modeli-2'nin deneyleri için iki mesaj serisi bulunmaktaydı ve Dağıtım Modeli-3'te de iki abone yer almaktadır. Bunun anlamı; temsilci, XRCE alıcıları ile iletişimini sürdürürken, DDS Global Veri Uzayı'nın iki abone üyesi için paketleri DDS Global Veri Uzayı üzerinden hiç göndermemiştir. Bunu DDS Global Veri Uzayı'nın sadece bir üyesi için yapmıştır, o da DDS alan katılımcısı'dır. XRCE alıcıları arasındaki iletişimi kısa bir yol oluşturarak sağlamıştır. Temsilcinin seçtiği yol bu çalışmanın üçüncü çıkarımıdır.

Bütün çıkarımlar bir kez daha göz önünde bulundurulduğunda, dördüncü çıkarımın var olması yüksek ihtimaldir. Dağıtım Modeli-2'nin testlerinden sonra, temsilcinin paket kaybı diğer senaryolar için de muhtemeldir. Dağıtım Modeli-3 testleri sırasında paket kaybeden temsilcinin gözlemlenmesi bu ihtimali daha gerçekçi hale getirmektedir. Bu yüzden; temsilci yayımcı bir XRCE alıcısıyla bağlantı kurmuşken Dağıtım Modeli-1 test edilmemesine rağmen, temsilcinin Dağıtım Modeli-1'in testleri sırasında da aynı şekilde işlemesi muhtemeldir. Temsilcinin paket kaybı ile ilgili Dağıtım Modeli-1 için göstereceği muhtemel davranış dördüncü çıkarımdır.

KAYNAKLAR

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M. ve Ayyash, M. (2015). Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys & Tutorials*, *17*(4), 2347-2376.

Andrei, G., Marlen, B., Sergey, T. ve Krinkin, K. (2020). Industrial Messaging Middleware: Standards and Performance Evaluation. *2020 IEEE 14th International Conference on Application of Information and Communication Technologies (AICT)*, Özbekistan, 7-9 Ekim 2020.

Babovic, Z. B., Protic, J. ve Milutinovic, V. (2016). Web Performance Evaluation for Internet of Things Applications. *IEEE Access*, *4*, 6974-6992.

Banno, R., Ohsawa, K., Kitagawa, Y., Takada, T. ve Yoshizawa, T. (2021). Measuring Performance of MQTT v5.0 Brokers with MQTTLoader. *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*, ABD, 9-12 Ocak 2021.

Basavaraju, N., Alexander, N. ve Seitz, J. (2021). Performance Evaluation of Advanced Message Queuing Protocol (AMQP): An Empirical Analysis of AMQP Online Message Brokers. *2021 International Symposium on Networks, Computers and Communications (ISNCC),* Birleşik Arap Emirlikleri, 31 Ekim-2 Kasım 2021.

Bayılmış, C., Ebleme, M. A., Çavuşoğlu, Ü., Küçük, K., Sevin, A. (2022). A survey on communication protocols and performance evaluations for Internet of Things. *Digital Communications and Networks*.

Bender, M., Kirdan, E., Pahl, M. –O. ve Carle, G. (2021). Open-Source MQTT Evaluation. *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC),* ABD, 9-12 Ocak 2021.

Bouloukakis, G., Georgantas, N., Kattepur, A. ve Issarny, V. (2021). Timed protocol analysis of interconnected mobile IoT devices. *Journal of Internet Services and Applications*, *12*(12), 1-31.

Chen, X., Kong, X., Ling, Y. ve Cao, X. (2021). DDS Performance Evaluation for PREEMPT_RT Linux. *2021 International Conference on Computer, Blockchain and Financial Development (CBFD),* Çin, 23-25 Nisan 2021.

Chen, Y. ve Kunz, T. (2016). Performance evaluation of IoT protocols under a constrained wireless access network. *2016 International Conference on Selected Topics in Mobile & Wireless Networking (MoWNeT),* Mısır, 11-13 Nisan 2016.

Chul-Hwan Kim, Gunjae Yoon, Wonjoon Lee, Jungdo Park ve Hoon Choi. (2015). A performance simulator for DDS networks. *2015 International Conference on Information Networking (ICOIN)*, Kamboçya, 12-14 Ocak 2015.

Dehnavi, S., Goswami, D., Koedam, M., Nelson, A. ve Goossens, K. (2021). Modeling, implementation, and analysis of XRCE-DDS applications in distributed multi-processor real-time embedded systems. *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, Fransa, 1-5 Şubat 2021.

Farea A. H. ve Küçük, K. (2021). Detections of IoT Attacks via Machine Learning-Based Approaches with Cooja. *EAI Endorsed Transactions on IoT*, *7*(28), e1.

Holler, J., Tsiatsis, V., Mulligan, C., Karnouskos, S., Avesand, S. ve Boyle, D. (2014). *From Machine-to-Machine to the Internet of Things - Introduction to a New Age of Intelligence* (1. Basım). Academic Press, Birleşik Krallık: Elsevier.

HyunHo Kim, HoonJae Lee and HyoTaek Lim. (2020). Performance of Packet Analysis between Observer and WireShark. *2020 22nd International Conference on Advanced Communication Technology (ICACT)*, Kore (Güney), 16-19 Şubat 2020.

Kang, Z., An, K., Gokhale, A. ve Pazandak, P. (2021). A Comprehensive Performance Evaluation of Different Kubernetes CNI Plugins for Edge-based and Containerized Publish/Subscribe Applications, *2021 IEEE International Conference on Cloud Engineering (IC2E)*, ABD, 4-8 Ekim 2021.

Kassem, I. ve Sleit, A. Elapsed Time of IoT Application Protocol for ECG: A Comparative Study Between CoAP and MQTT. *2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, Türkiye, 12-13 Haziran 2020.

Krinkin, K., Filatov, A., Filatov, A., Kurishev, O. ve Lyanguzov, A. (2018). Data Distribution Services Performance Evaluation Framework. *2018 22nd Conference of Open Innovations Association (FRUCT)*, Rusya, 15-18 Mayıs 2018.

Laghari, A. A., Hui He, Khan, A., Laghari, R. A., Shoulin Yin ve Jiachi Wang. (2022). Crowdsourcing Platform for QoE Evaluation for Cloud Multimedia Services. *Computer Science and Information Systems*, *19*(3), 1305-1328.

Laghari, A. A., Wu, K., Laghari, R. A., Ali, M. ve Ayub Khan, A. (2021). A Review and State of Art of Internet of Things (IoT). *Archives of Computational Methods in Engineering*, *29*(2), 1395–1413.

Maruyama, Y., Kato, S. ve Azumi, T. (2016). Exploring the performance of ROS2. *2016 International Conference on Embedded Software (EMSOFT)*, ABD, 2-7 Ekim 2016.

Palmese, F., Longo, E., Redondi, A. E. C. ve Cesana, M. (2021). CoAP vs. MQTT-SN: Comparison and Performance Evaluation in Publish-Subscribe Environments. *2021 IEEE 7th World Forum on Internet of Things (WF-IoT)*, ABD, 14 Haziran-31 Temmuz 2021.

Phueakthong, P. ve Varagul, J. (2021). A Development of Mobile Robot Based on ROS2 for Navigation Application. *2021 International Electronics Symposium (IES),* Endonezya, 29-30 Eylül 2021.

Pohl, M., Kubela, J., Bosse, S. ve Turowski, K. (2018). Performance Evaluation of Application Layer Protocols for the Internet-of-Things. *2018 Sixth International Conference on Enterprise Systems (ES)*, Kıbrıs, 1-2 Ekim 2018.

Profanter, S., Tekat, A., Dorofeev, K., Rickert, M. ve Knoll, A. (2019). OPC UA versus ROS, DDS, and MQTT: Performance Evaluation of Industry 4.0 Protocols. *2019 IEEE International Conference on Industrial Technology (ICIT)*, Avustralya, 13-15 Şubat 2019.

Sasaki, Y., Yokotani, T. ve Mukai, H. (2018). Comparison with Assured Transfer of Information Mechanisms in MQTT. *2018 International Japan-Africa Conference on Electronics, Communications and Computations (JAC-ECC)*, Mısır, 17-19 Aralık 2018.

Schlesselman, J. M., Pardo-Castellote, G. ve Farabaugh, B. (2004). OMG data-distribution service (DDS): architectural update, *IEEE MILCOM 2004. Military Communications Conference*, ABD, 31 Ekim-3 Kasım 2004.

Sheng Huang, Yu-Hsuan Lu, Shafiq, M., Laghari, A. A. ve Yadav, R. (2021). A Generative Adversarial Network Model Based on Intelligent Data Analytics for Music Emotion Recognition under IoT. *Mobile Information Systems*, *2021*(1).

Tandale, U., Momin, B. ve Seetharam, D. P. (2018). An empirical study of application layer protocols for IoT. 2017 International Conference on Energy, *Communication, Data Analytics and Soft Computing (ICECDS)*, Hindistan, 1-2 Ağustos 2017.

Thulasiraman, P., Cheng, Y. K. D. ve Allen, B. (2022). Evaluation of the Data Distribution Service for a Lossy Autonomous Hybrid System. *2022 IEEE International Systems Conference (SysCon)*, Kanada, 25-28 Nisan 2022.

URL-1: https://github.com/orgs/eProsima/repositories (Ziyaret tarihi: 24 Kasım 2022)

URL-2: https://www.omg.org/spec/About (Ziyaret tarihi: 24 Kasım 2022)

URL-3: https://micro-xrce-dds.docs.eprosima.com/en/latest/ (Ziyaret tarihi: 24 Kasım 2022)

URL-4: https://fast-dds.docs.eprosima.com/en/latest/fastddsgen/introduction/introduction.html (Ziyaret tarihi: 24 Kasım 2022)

URL-5: https://gnome-terminator.org/about/ (Ziyaret tarihi: 24 Kasım 2022)

Wang, J. J., Payne, R. (2022). A survey of Internet of Things in Healthcare. *EAI Endorsed Transactions on Internet of Things*, *7*(27), 1-11.

Waqas, M., Kumar, K., Laghari, A. A., Saeed, U., Rind, M. M., Shaikh, A. A., Hussain, F., Rai, A. ve Qazi, A. Q. (2021). Botnet attack detection in Internet of Things devices over cloud environment via machine learning. *Concurrency and Computation*, *34*(5), 1-23.

EKLER

Ekler dijital ortamda teslim edilmiştir.

KİŞİSEL YAYIN VE ESERLER

Şolpan, Ş. ve Küçük, K. (2022). DDS-XRCE Standard Performance Evaluation of Different Communication Scenarios in IoT Technologies. *EAI Endorsed Transactions on Internet of Things*, *8*(4):e1. 23 Kasım 2022. https://doi.org/10.4108/eetiot.v8i4.2691.

ÖZGEÇMİŞ

İlk, orta ve lise eğitimini Sakarya'da tamamladı. 2014 yılında kazandığı Kocaeli Üniversitesi Mekatronik Mühendisliği Bölümü'nden 2019 yılında mezun oldu. Mezun olduktan kısa bir süre sonra Kocaeli Üniversitesi Fen bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı'nda Yüksek Lisans öğrenimine başladı. 2022 yılında Yüksek Lisans öğrenimine devam etmektedir. Yüksek Lisans öğrenimi boyunca "DDS-XRCE Standard Performance Evaluation of Different Communication Scenarios in IoT Technologies" adında bir çalışma yapmıştır.