



Remote user supported IoT-WSN Laboratory and testbed platform: FiratWSN

Güngör Yıldırım*^{id}, Yetkin Tatar^{id}

Department of Computer Engineering, Engineering Faculty, Firat University, Elazig, 23100, Turkey

Highlights:

- WSN Virtualization
- Virtual WSN Testbed
- WSN laboratory

Keywords:

- Wireless Sensor Network
- Virtual WSN Laboratory
- Virtualization
- CoAP
- Internet of things

Article Info:

Research Article
Received: 03.03.2018
Accepted: 26.03.2019

DOI:

10.17341/gazimmfd.571588

Acknowledgement:

We would like to thank Firat University Scientific Research Projects Unit for the project entitled "MF.16.19 - Design of a distributed-parallel cyber physical system based on virtual wireless network"

Correspondence:

Author: Güngör Yıldırım
e-mail:
gungor.yildirim@firat.edu.tr
phone: +90 533 285 2299

Graphical/Tabular Abstract

In this study, the implementation and development of a virtual WSN testbed platform, called FiratWSN, which enables the running of remote test applications for academic studies, is introduced. Unlike the other similar studies in the literature, with the help of Agilla middleware, the testbed can allow a certain number of different user applications to be run concurrently on the same programmable physical WSN nodes. The testbed also provides an infrastructure for heterogeneous application tests. Users can remotely connect to the system over the internet and test their WSN applications in a flexible lab environment, which this provides an important cost and installation time saving to them.

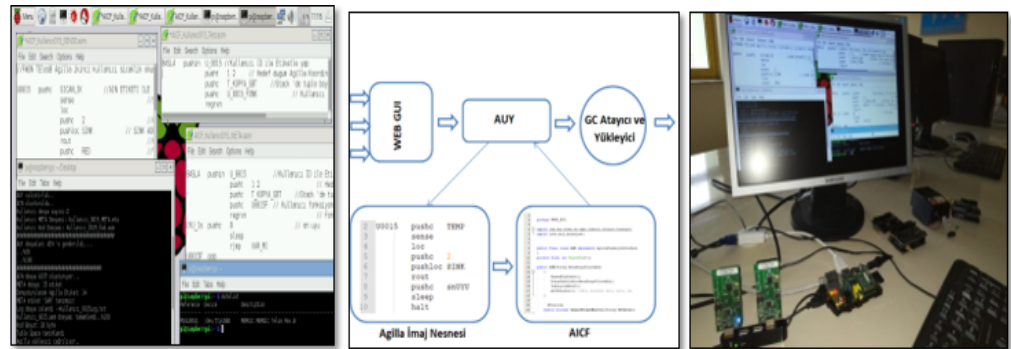


Figure A. The node based virtualization backbone of the IoT-WSN testbed

Purpose: The aim of the study is to develop a virtual IoT-WSN testbed platform that provides multi-remote user support. This will allow more users to use the same shared WSN resources simultaneously.

Theory and Methods:

The basic idea behind this platform is based on three concepts. These are node based WSN virtualization, heterogeneity, and remote user support. The node-based virtualization support is done by the Agilla middleware, and the developed two software, AUY, and AICF. A screenshot from the virtualization backbone is presented in Fig.A. Also, the testbed platform includes heterogeneous WSN technologies in terms of hardware, sensors, actuators, and stack. Multi-remote user support is maintained via world-wide-web. The platform utilizes CoAP and RESTful for IPv6 applications.

Results:

The developed platform has successfully provided all infrastructure supports expected from it. While the success rate for CoAP-based tests is between 90 and 100%, the success rate in the virtual node applications is between 85 and 100 %.

Conclusion:

Designed so as to be a part of large scale WSN projects, the FiratWSN testbed platform, which is compatible with the Client / Server structure, can collaborate with other WSN research laboratories located in different locations to create a broader virtual WSN testing environment over the internet. Hereby, it is clear that it will provide great opportunities.



Uzak kullanıcı destekli bir IoT-WSN sanal laboratuvarı ve test platformu: FıratWSN

Güngör Yıldırım*^{ID}, Yetkin Tatar^{ID}

Fırat Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, Elazığ, 23200, Türkiye

Ö N E Ç İ K A N L A R

- WSN Sanallaştırma
- Sanal WSN test platformu
- WSN Laboratuvarı

Makale Bilgileri

Araştırma Makalesi

Geliş: 03.03.2018

Kabul: 26.03.2019

DOI:

10.17341/gazimmfd.571588

Anahtar Kelimeler:

Kablosuz duyurga ağları,
sanal WSN laboratuvarı,
sanallaştırma,
CoAP,
Nesnelerin interneti

ÖZET

Nesnelerin interneti (IoT) ve siber fiziksel sistemlerin (CPS) gelişimi ile bunların yapı taşlarından olan kablosuz duyurga ağlarının (KDA/WSN) önemi daha da artmış, buna paralel olarak da, klasik WSN sistemlerin IoT/CPS projelere adapte edilmesine yönelik çalışmalar hız kazanmıştır. Tahmin edileceği üzere bu çalışmalar kapsamlı WSN test platformlarına ihtiyaç duymaktadır. Böyle kapsamlı bir WSN test platformunun tesis edilmesi ise masraflı ve detaylı bir süreçtir. Bu çalışmada gerek lisans, gerekse de lisansüstü çalışmalarda uzaktan test uygulamalarına olanak tanıyabilen, FıratWSN isimli bir sanal WSN laboratuvar/test platformu tasarımı ve gerçekleştirilmesi tanıtılmıştır. İstemci/Sunucu yapısına göre çalışan bu IoT-WSN sanal laboratuvar ve test platformu, literatürdeki benzerlerinden ayrı olarak, çoklu uygulamaların, programlanabilir WSN düğümler üzerinde, eş zamanlı çalışabilmesine olanak sağlayabilmektedir. Bunu Agilla ara katman yazılımı sayesinde yapan sistem, ayrıca günümüz güncel IoT teknolojilerinden olan CoAP/RPL uygulama deneylerinin yapılmasına da imkân sunmaktadır. FıratWSN sistemi, FıratZigBee alt WSN, 1322x ZigBee alt WSN ve 802.15.4 teknolojiye dayalı düğümlerden oluşturulan farklı alt WSN yapılarını da içermektedir. FıratWSN sistemi bu şekilde heterojen uygulamalar geliştirmeye de imkân sağlayabilmektedir. Böylece kullanıcılar, internet üzerinden uzaktan bağlanarak kendi WSN uygulamalarını esnek bir laboratuvar ortamında maliyet ve kurulum zamanından tasarruf ederek test edebilmektedirler.

Remote user supported IoT-WSN Laboratory and testbed platform: FıratWSN

H I G H L I G H T S

- WSN Virtualization
- Virtual WSN Testbed
- WSN laboratory

Article Info

Research Article

Received: 03.03.2018

Accepted: 26.03.2019

DOI:

10.17341/gazimmfd.571588

Keywords:

Wireless sensor network,
virtual WSN laboratory,
virtualization,
CoAP,
Internet of things

ABSTRACT

Along with the development of Internet of Things (IoT) and Cyber-Physical Systems (CPSs), the importance of wireless sensor networks (WSNs), which are one of their fundamental subsystems, has obviously increased. In line with this, studies towards adapting traditional WSN systems to IoT/CPS projects have gained speed. As might be expected, these types of studies require comprehensive WSN test platforms, which this process could be quite expensive and detailed. In this study, the implementation and development of a virtual WSN testbed platform, called FıratWSN, which enables the running of remote test applications for academic studies, is introduced. Unlike the other similar studies in the literature, with the help of Agilla middleware, the testbed can allow a certain number of different user applications to be run concurrently on the same programmable physical WSN nodes. On the other hand, FıratWSN also includes three sub-WSNs; the FıratZigBee sub-WSN, a 1322x ZigBee sub-WSN, and an 802.15.4 sub-WSN consisting of nodes of different technologies. In this way, the testbed also provides an infrastructure for heterogeneous application tests. Users can remotely connect to the system over the internet and test their WSN applications in a flexible lab environment, which this provides an important cost and installation time saving to them.

*Sorumlu Yazar/Corresponding Author: gungor.yildirim@firat.edu.tr, ytatar@firat.edu.tr / Tel: +90 533 285 2299

1. GİRİŞ (INTRODUCTION)

Kablosuz duyurga ağları (KDA/WSN) son yılların hem akademik hem de endüstriyel alanda dikkat çekici teknolojilerden birisi haline gelmiştir. Uzaktan gözlemlenimin en verimli ve ekonomik çözümü olan bu teknoloji, askeri, sağlık, çevresel v.b. projelerde aktif rol oynayabilmektedir. Özellikle “Nesnelerin İnterneti - IoT” teknolojilerinin gelişmesiyle birlikte daha da önemi artan WSN’ler, akademik alandaki güncelliklerini de korumaktadırlar. Ancak WSN’ler normal ağlardan farklı olarak; enerji kısıtları, dağıtık yapıları ve düğümlerinin sınırlı işlem kabiliyetleri gibi birçok olumsuzlukları da barındırmaktadır. Bu problemlerin çözümüne yönelik olarak pek çok yaklaşım ve teknolojiler önerilmiş ve geliştirilmiştir. Bunun sonucunda karşımıza farklı özelliklere ve çalışma yapılarına sahip, yazılımsal ve donanımsal olarak geniş yelpazeli WSN teknolojileri çıkmaktadır. Doğal olarak bu farklı teknolojilerin birlikte çalışabilirliğinin sağlanması yani oluşacak büyük ölçekli heterojen WSN yapıların gerek kurulum gerek işletme aşamasında davranışlarının incelenmesi gerekliliği önemli ve karmaşık bir konudur. Büyük ölçekli WSN’ler, oluşturulmadan önce veya yeni topoloji ve teknoloji WSN’lerin geliştirilme sürecinde davranışlarının incelenmesi için ilgili WSN simülasyon araçları kullanılır. Ancak simülasyon sürecinde, gerçek çalışma şartlarının hassas bir şekilde incelenmesi kolay değildir.

Öte yandan WSN’ler gibi teknoloji odaklı konuların öğretilmesi, araştırılması, geliştirilmesi sürecinde teorik çalışmaların yanı sıra laboratuvar ortamında uygulamalar, araştırmalar ve deneylerle desteklenmesi modern üniversite araştırma anlayışının bir parçası haline almıştır. Buna en güzel örneklerden birisi de üniversitelerin birçoğunda WSN’ler ile ilgili araştırma, geliştirme ve uygulamaları için laboratuvar/ test platformlarının oluşturulmasının devam etmesi gösterilebilir [1-3]. Buradaki temel nokta, oluşturulacak WSN laboratuvarının güncel teknolojilere odaklanabilecek bir alt yapıya sahip olması, heterojen WSN yapılarının çalışabilirliği üzerine yapılan araştırmaların gerçek çalışma şartlarına uygun olarak yapılabilmesi ve ekonomik bir şekilde kullanılabilir olmasıdır. Bu bakış açısına göre, tek bir teknoloji üzerine oluşturulmuş lokal WSN laboratuvarlar her ne kadar temel noktalarda yeterlilik sağlasalar da günümüz bilgi teknolojilerinin (IT) spesifik problemlerine yönelik yaklaşım geliştirmede eksik kalabilmektedir. Çünkü çok sayıda sensör/aktuatör ve teknoloji çeşitliliği ve bunların kullanıldığı farklı disiplinler göz önüne alındığında bir WSN laboratuvarının içermesi gereken heterojen yapının genel çizgileri de ortaya çıkmaktadır. Bunlara ilaveten günümüz WSN yapıları lokal ve amaca yönelik yapılar olmaktan çıkıp internet üzerinden birbirleri ile veya diğer teknolojiler ile entegre olabilen sistemler haline gelmiştir. Bu ise zaten var olan heterojen yapı sorunlarına farklı iletişim problemlerini de katmaktadır. Ayrıca WSN sistemleri çok sayıda sensör düğüm içerebilir

ve sık aralıklarla bilgi toplayabilirler. Bu da toplanan verinin boyutunu, karmaşıklığını ve işleme zorluğunu arttırmaktadır. Ayrıca bu büyük veri kümelerinin depolanması, analizi ve işletilmesi şu ana kadar belirtilen sorunlara yenilerini katmaktadır.

Sonuç olarak, bir WSN laboratuvarı veya test platformundan beklenenler; yukarıda açıklanan karmaşık özelliklerin anlaşılması, araştırılması, gerçek çalışma şartlarında deneylerin yapılabilmesi ve çözüm üretilebilmesi süreci için altlık sağlayabilecek bir potansiyele sahip olmasıdır. Donanımsal ve yazılımsal olarak sürekli güncellenebilmeli, veri depolayabilmeli ve en önemlisi de sadece lokal kullanıcılara değil aynı zamanda internet üzerindeki izin verilen uzak kullanıcılara da gerçek zamanlı olarak hitap edebilmelidir. Yani sanal WSN laboratuvar özelliğinde olmalıdır. Böylece tek bir fiziksel WSN deneysel ortamından, çok sayıda ve uzaktaki istemcilerin yararlanabilmesi mümkün olacaktır.

Bu makalede, lokaldeki veya internet üzerindeki uzak kullanıcıların kendi yazmış oldukları WSN uygulamalarını fiziksel bir WSN laboratuvar ortamında test edebilmelerine imkân veren bir sanal IoT WSN test platformunun gerçekleştirilmesi ve tanıtılması amaçlanmıştır. *FıratWSN* ismi verilen bu Sanal IoT WSN laboratuvar ve test platformu iki ana kısımdan meydana gelmiştir. Birincisi fiziksel IoT WSN laboratuvar ortamıdır. Bu ortamda heterojen bir WSN yapısının oluşturulmasına da imkân veren farklı teknoloji ve programlanabilen WSN düğümleri, kenar cihazları, gateway v.b donanımlar mevcuttur. Ayrıca IPv6 temelli IoT desteği ve NoSQL tabanlı bir veritabanı desteği de kullanıcıya sunulmaktadır. İkinci ana kısmı ise; Sanal WSN yönetim yazılımı”, “Web Arayüz” ve “NoSQL Veritabanı” yönetim yazılımlarını içeren sunucu sistemdir. Sanal IoT WSN platformunun hedefi; sisteme üye olan ve izin verilen uzak veya yakın istemcilerin, sunucu üzerindeki yönetim yazılımları aracılığı ile fiziksel sisteme ulaşabilmeleri ve izin verilen sensör düğümlerini kendi isteklerine göre programlayarak WSN yapıları oluşturabilmeleri, denemeler yapabilmeleri ve sonuçlarını da yine bir web arayüzünden alabilmeleridir. İçerdiği farklı teknoloji diğer heterojen alt WSN ‘ler ile de etkileşime olanak sağlayan bu sanal WSN laboratuvarın literatürdeki diğer test platformlarından (testbed) en önemli farkı, düğüm bazlı sanallaştırma ile aynı anda bir WSN düğüm üzerinde birden fazla uygulamanın test edilmesine olanak sağlayabilmesidir. Fırat Üniversitesi Bilgisayar Mühendisliği Bölümü WSN laboratuvarında gerçekleştirilen *FıratWSN* sanal platformu lisans, lisansüstü ve proje tabanlı araştırmalar için hem lokal hem de uzak kullanıcılara olanaklar sağlayabilmektedir.

Amaçlanan sanal WSN laboratuvarının oluşturulması, test edilmesi ve işletilmesi sürecini açıklayan bu makalenin sonraki bölümleri şu şekilde planlanmıştır. Bölüm II konuyla ilgili motivasyon sürecini, Bölüm III önerilen sanal laboratuvar sistem modelini, detaylarını ve testleri, Bölüm IV sonuç ve önerileri içermektedir.

2. MOTİVASYON (MOTIVATION)

Bir WSN simülâtörün hassasiyeti, mevcut ve olası ortam koşullarını dikkate alarak oluşturulmuş olan fiziksel sistem modelinin detayıyla orantılıdır. Genellikle test, değerlendirme ve yeni protokollerin ilk doğrulamaları için kullanılan WSN simülâtörlerde daima bir hata payı mevcuttur [4]. Bir WSN laboratuvarı ve test platformu ise kontrol edilebilir WSN düğümlerinin çoğunlukta bulunduğu bir ortamdır ve WSN'ler ile ilgili teorileri, hesaplama araçlarını ve yenilikleri hassas, esnek, şeffaf ve tekrarlanabilir bir şekilde gerçek zaman ve şartlarda test etmeye izin verir. Ancak WSN test platformlarının sürekli olarak güncel tutulması, işletilmesi oldukça masraflı bir süreçtir. Bunun için WSN konusunda çalışan akademik ve endüstriyel AR-GE kuruluşları işbirliğine giderek ilgili fiziksel WSN platformlarını ortak kullanılabilecekleri şekilde düzenleyebilirler. WSN test platformlarının ekonomik olarak oluşturulabileceği en uygun yapı ise Sanal WSN laboratuvarı/test platformlarıdır. Sanal WSN test platformunun genel tanımı; bir veya birden fazla fiziksel WSN laboratuvarı/ test platformunun, internet üzerindeki kayıtlı üyeler tarafından, belirlenmiş zaman dilimlerinde, kendi istekleri doğrultusunda kontrol edilmesi, programlanması ve kullanılması olarak yapılabilir.

WSN test platformları; “amaçlarına göre” ve barındırdıkları “altyapıya” göre iki temel sınıfa ayrılır. Amaca göre yapılandırılmış WSN test platformları da “çok kullanıcı” ve “önerilen WSN kavram-doğrulama” platformları olarak alt sınıflara ayrılabilir. Amaca yönelik WSN test platformları araştırmacıların yeni ağ teknolojilerini, protokollerini, mimarilerini ve hizmetlerini denemeleri için iyi bir altyapı sağlar. Altyapıya göre WSN test platformu sınıfı ise farklı boyuttaki WSN yapılarının kullanıldığı platformlardır. Bu tür WSN test platformları WSN unsurlarını (düğümler, anahtarlar, yönlendiriciler vb.) ortak kullanılabilecek şekilde bir araya getirir. Uzaktan erişilemeye, kullanıcılara değişik topolojiler üzerinde deneyler yapmaya, veri toplayıp değerlendirmeye, WSN ağ bileşenleri üzerine detaylı işlemler yapmaya imkân verir [4].

Sunulan bu makalede, “amaca göre” ve “alt yapıya” göre yapılandırılmış WSN test platformlarının özelliklerini içinde bulunduran hibrid çok kullanıcı WSN test platform yapısı çalışılmıştır. Bu bakımdan konuyla ilgili olan WSN test platformları ile ilgili daha önce yapılan çalışmalar incelenip değerlendirilmiştir. Birçok üniversitenin ortak katılımı ile gerçekleştirilmiş, büyük ölçekli bir WSN test platform olan WISEBED’de özellikle, donanımsal/yazılımsal katmansal geliştirmeler üzerine çalışmalar gerçekleştirilmektedir [5]. Bu test platformunda TelosB, iSense, T-Node gibi düğümler bulunmakta ve heterojen bir alt yapı sağlanmaktadır. Çoklu uygulama çalıştırabilen WSN test platformu çalışması da literatürde mevcuttur [6]. Belirli bir uygulamaya özel olması, uzak kullanıcı ve heterojenlik desteği sunmaması bu başarılı çalışmanın dikkat çeken eksiklikleri olarak görülebilir. OSAKA Üniversitesi’nde tesis edilmiş olan X-Sensor adlı WSN laboratuvarı da amaca yönelik ağ keşfi, kullanıcılar

için veri arşivleme veri yönetim protokolleri ile uzaktan ağ yönetimlerinin gerçekleştirilmesine imkân tanımıştır [7]. Almanya Karlsruhe Teknoloji enstitüsü ise daha çok aktuatör düğümler üzerine odaklanmış olan WSN sistemler, diğer adıyla WSN’lar ile ilgili SANbed platformunu duyurmuştur [8]. Bu test platformu aktuatör düğümlerde dağıtık enerji yönetimi, protokol iyileştirme ve geliştirme, ağ yönetim yazılımları hazırlama gibi çalışmalar için kurulmuştur. SANbed’deki düğümlerin gözlemlenmesi SNMD (Sensor Node Management Device) denilen modüller tarafından gerçekleştirilirken düğüm teknolojisi olarak MicaZ, TelosB tipi düğümler kullanılmıştır. SensLAB ise Fransa’da kurulmuş ve 1000’in üzerinde düğüm içeren bir WSN test platformudur [9]. Amacı doğrudan büyük ölçekli bir WSN platformunun kurulması olan bu projede, genelde WSN430 tipi düğümler tercih edilmiştir. İşletim sistemi olarak Contiki, TinyOS ve FreeRTOS’un kullanıldığı SensLAB, özellikle katmansal teknoloji geliştirmede kullanılmıştır. Bir diğer WSN laboratuvarı Harvard Üniversitesinde kurulmuş olan MoteLab’dır [10]. Bu laboratuvarın özelliği, kayıtlı kullanıcılar için açık bir platform olmasıdır. Kayıtlı kullanıcılar kendi geliştirdikleri uygulama kodlarına ait image’leri kendilerine atanmış zamanlarda çalışmak üzere, düğümlere yükleyerek çıktılarını elde edebilmeleridir. Bu platform TelosB ve TinyOS teknolojilerini kullanmaktadır. TinyOS platformu üzerine tesis edilmiş bir diğer test platformu Atina Bilgi Teknolojilerinde kurulmuş olan SenseNet’dir [11]. Aynı anda farklı kullanıcıların ulaşımına imkân sağlayan sistemde TelosB ve Mica düğümler kullanılmıştır. Ulusal Tsing Hua Üniversitesi, WSNBTB projesi ile heterojen bir WSN test platformu geliştirmeyi amaçlamışlardır. Bu çalışmada 802.15.4 ve ZigBee desteği sunabilen Octopus I/II düğümleri kullanılmıştır. Bu test platformunda uzak kullanıcıların kaynak seçimine ve bu kaynaklar üzerinde uygulama geliştirmesine imkân veren bir web arayüzü de bulunmaktadır. Benzer şekilde WSN ‘lerde heterojenlikten kaynaklanan problemlerin çözümüne yönelik çalışmalar için alt yapı desteği sağlayan diğer test platformları da mevcuttur [12, 13]. Ancak bu test platformları uygulama tekelliğini kıramamış ve çoklu uygulama desteği açısından lokal bazlı kalmışlardır. Meşhur CitySense projesi Amerika’da Cambridge şehri geneline tesis edilmiş bir WSN uygulaması ve test platformudur [14]. Mesh tipi teknoloji üzerine kurulu olan sistem, 802.11 standardını kullanmaktadır. Wyne State Üniversitesi’nde geliştirilen NetEye test platformu da TelosB düğüm ve TinyOS gömülü işletim sistemi üzerine kurulmuş bir platformdur [15]. MySQL’veri tabanı sisteminin kullanılmış olduğu bu sistem aynı zamanda WSN ağların diğer kablosuz ağlarla aynı ortamlarda bulunması durumunda oluşabilecek girişim problemlerinin analizinde de kullanılmıştır. Sensei ise Uppsala Üniversitesinde kurulmuş olan bir WSN test platformudur ve mobil düğüm desteği sunması ile öne çıkmaktadır. Düğüm teknolojisi olarak TelosB ve uzak bağlantılar için 802.11standartı kullanılmıştır [16]. Öte yandan WSN ‘lerde mobilite desteği içeren test platformları özellikle kablosuz ortamlarda konum tespiti alanında yapılan çalışmalar için önemli alt yapı desteği sunmaktadır [17].

Kablosuz ortam ve kanal analizlerine yönelik test platformları da literatürde bulunmaktadır [18]. Ancak bunlar genellikle teknik kurulum analizlerinde destek sağlamaktadır. WSN test platformlarında uzaktan erişim, heterojenlik ve her bir WSN düğüm üzerinde çoklu uygulama yürütme özelliklerinin bir arada olması istenir. Dikkat edilirse bu üç temel özellikten en az bir tanesinin yukarıda açıklanan WSN test platformlarında bulunmadığı görülmektedir. Bu makalenin odaklandığı amaç ise, yukarıda bahsedilen bu üç özelliğin bir arada bulunduğu gelişmiş bir IoT tabanlı WSN test platformunu, WSN sanallaştırma yöntemleri kullanarak geliştirmektir. Böylece özellikle eğitim ve geliştirme için daha fazla kullanıcıya aynı anda destek sağlanabilmekte ve dolayısıyla ekonomik bir yapı sunma özelliği taşımaktadır. Geliştirilen bu test platformunun sağlayacağı avantajları daha iyi ifade edebilmek için bu sistem üzerinde yapılabilecek bazı araştırma/geliştirme çalışmaları Tablo 1’de sunulmuştur.

Bu çalışmada sistemin yeteneklerinin ve başarımlarının gösterilmesi bakımından iki uygulama senaryosu test edilmiştir. Bunlar, IPv6 destekli CoAP protokolü ve çoklu kullanıcı veri toplama algoritmaları (Data Aggregation) uygulamalarıdır. Bu uygulama senaryoları ilgili sistem alt yapısının detayları ile birlikte verilecektir.

3. GELİŞTİRİLEN TEST PLATFORMU MODELİ VE SİSTEM ALT YAPISI (THE INFRASTRUCTURE OF THE TESTBED PLATFORM)

Geliştirilen test platformu farklı teknoloji heterojen düğümlerden ve kaynaklardan oluşmaktadır. Düğüm olarak 10 adet MEMSIC TelosB, 20 adet FNode, 12 adet Libelium Waspmote, 15 adet Digi standalone XBee S1/S2 ve 4 adet NXP132xx uygulamalar için hazır bulunmaktadır. Analog ve sayısal olarak toplam 23 çeşit farklı sensör kaynak mevcuttur. Ara erişim ve programlama aygıtı olarak 4 adet Raspberry Pi bulunmaktadır. Üzerlerinde bulunan USB çoklayıcılarla aynı anda 4 adreslenebilir düğümü programlayabilmektedir. Bu bağlamda uzaktan programlanabilir düğümlerin ölçülebilirliği, programlama aygıtına sağlanan IP havuzuna bağlı olmakla beraber şu an için bu konuda teknik bir kısıt bulunmamaktadır. Programlanabilir düğümler TinyOS gömülü işletim sistemi çalıştırmakta olup programlama dili olarak nesC ‘yi kullanmaktadır. Uzak kullanıcıların bu test platformuna erişimleri web tabanlıdır. Bu web sitesi, OpenStack Newton bulut işletim sistemli 3 adet Lenovo x3650 sunucu üzerindeki Apache Server hizmeti sunan sanal makine tarafından yürütülmektedir. Geliştirilmiş olan Sanal

WSN laboratuvarı ve test platformunda üç ana hedef gözetilmiştir. İlki farklı kullanıcılara ait WSN uygulamalarının eş zamanlı yürütülmesine olanak sağlamaktır. İkincisi farklı teknoloji WSN yapıları ile heterojen bir altyapı sunmak ve IoT/CPS teknolojilerine uyumlu bir şekilde genişletilebilir olmasına imkân vermektir. Üçüncüsü hem lokal hem de uzak kullanıcılara online veya offline olarak homojen/heterojen WSN ağ oluşturma, analiz edebilme, veri toplama ve değerlendirme hizmeti verebilmektir. Buna göre geliştirilen sistemin yapısı Şekil 1’de verilmiştir. Şekilden görüleceği gibi sistem iki ana kısımdan oluşmaktadır. İlk kısmı, farklı teknoloji WSN düğümler içeren WSN sistemler ve gateway’lerden oluşan fiziksel alt yapı’dır. Fiziksel altyapıda kullanılan düğümlerin seçiminde iki kriter dikkate alınmıştır. Birincisi performans yönetimi, katmansal protokol teknolojileri geliştirme, konumlandırma ve WSN’lerde gömülü yazılım tasarımı gibi akademik araştırmaların yapılabileceği WSN yapılarının oluşturulabilmesidir. Bunun için TinyOS [20], Contiki [21] gibi gömülü işletim sistemleri ile çalışabilen, akademik çalışmalarda sıklıkla kullanılan [22] programlanabilir MEMSIC TelosB düğümler tercih edilmiştir. İkinci kriter ise; özellikle saha uygulamalarının denenmesi için sensör/aktuatör düğüm yapılarının çoğunlukta olduğu WSN’lerin oluşturulabilmesidir. Bunun içinde sensör/aktuatör entegrasyonu daha kolay olan Libelium ve Digi XBee (S1/S2) nin 802.12.4 temelli düğümleri ve ZigBee düğümleri kullanılmıştır. Ayrıca bu makalenin yazarları tarafından geliştirilmiş ve Fnode olarak isimlendirilen sensör düğümlerden oluşan Fırat ZigBee WSN yapısı da sanal laboratuvar sistemi içerisinde veri paylaşımı deneyleri için kullanılmaktadır. Yukarıda açıklanan farklı teknolojilerdeki düğümlerden oluşturulmuş 4 tane WSN gurubu göz önüne alındığında; sanal laboratuvar sistemi kullanıcılara, düğümleri programlanabilir heterojen bir fiziksel WSN sistemi sunabilmektedir. Test platformunun ikinci kısmı ise; “Web Arayüz Yazılımı”, Sanal Laboratuvar işlem ve çoklu uygulama için sanallaştırma modüllerini içeren “Sanal WSN Yönetim Yazılımı”, ZigBee temelli FıratZigBee alt ağının yönetim/erişim modülü “FMonitör Yazılımı” ve “NoSQL Veritabanı yönetim”, yazılımlarını içeren sunucudur. Sunucu kısmında geliştirilmiş olan yazılımlar aracılığı ile yönetici, mevcut fiziksel WSN’leri ve veri tabanı dosyalarını test platformuna ekleyip çıkarabilmekte, her türlü sanal laboratuvar faaliyetini denetleyebilmektedir. İstemciler ise Web Arayüzü yazılımı aracılığı ile sanal WSN laboratuvarı hizmeti alabilmektedirler. Önerilen sistem, farklı teknoloji WSN yapılarının eklenmesine imkân verecek şekilde tasarlanmıştır.

Tablo 1. Test platformu üzerinde yapılabilecek bazı Ar-Ge Çalışmaları (Some R&D studies that can be performed on the platform)

MAC/Ağ ve Uygulama katmanı tasarımları	Enerji temelli veri toplama algoritmaları (Data Aggregation Algorithms)
Topoloji kontrol algoritmaları	Mobilite testleri
Veri çıkarımı/tahmin algoritmaları (Data Fusion Algorithms)	WSN’lerde zaman senkronizasyonu
IPv6 destekli CoAP uygulamaları	Konum tespiti uygulamaları
Dağıtık veri depolama algoritmaları	Büyük sensör verilerinde artıklık analizi ve optimizasyon

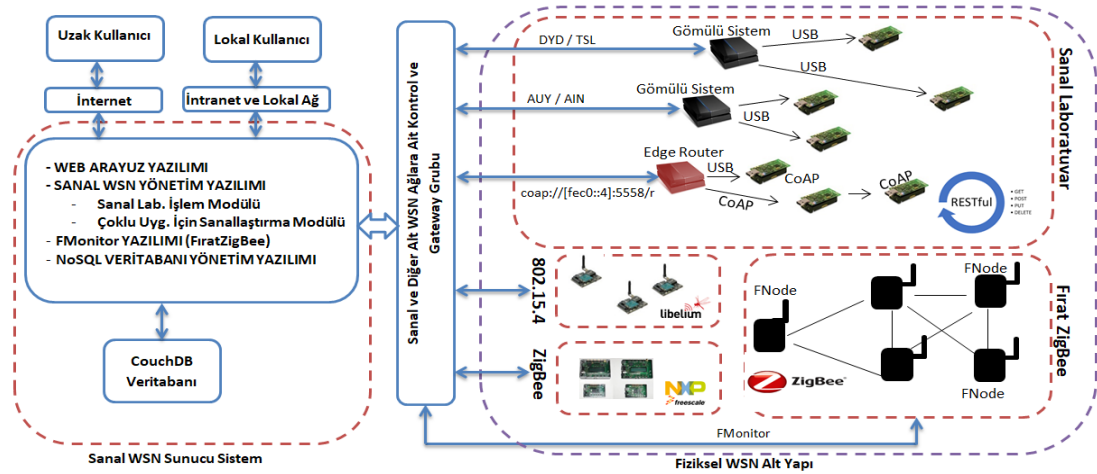
3.1. Sanal WSN Yönetim Yazılımı (Virtual WSN Management Software)

Bu yazılım, hali hazırda WSN kaynaklarına sahip olmayan veya yazılan kodları/protokolleri gerçek bir test ortamında denemeye ihtiyaç duyan, ancak bunun için konum ve zaman problemi yaşayan araştırmacılar için gerçek WSN kaynakları üzerine oluşturulmuş web tabanlı bir sanal WSN laboratuvar özelliği taşır.

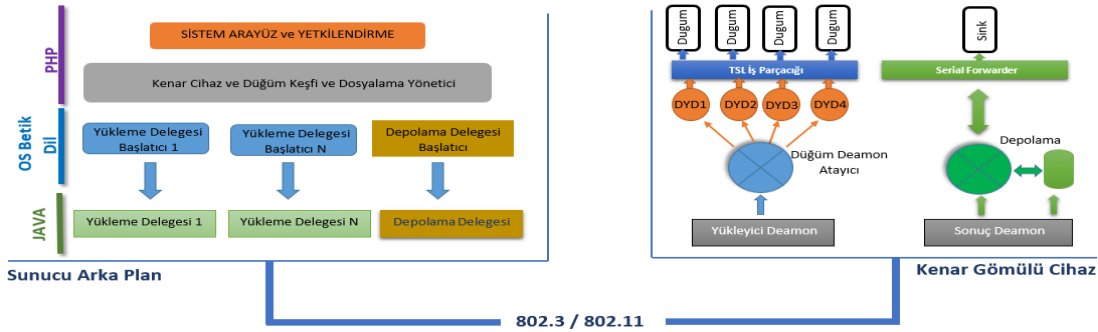
3.1.1. Sanal laboratuvar işlem modülü (The virtual laboratory operation module)

Belirlenmiş bir web adresi üzerinden ulaşılan ve RPL/CoAP teknolojisi desteği sunan sanal laboratuvar modülü dört ana kavram temelinde geliştirilmiştir. İstemci, sunucu sistem, kenar yönlendirici gömülü cihazlar (KGC) ve düğümler. Gerekli yetkilendirme gereksinimlerine sahip istemciler, randevu bazlı olarak, bu WSN laboratuvarının kamusal kullanıma açık olan bölümlerinden faydalanabilmektedir. Örnek olarak, bir yönlendirme protokolü tasarlamak isteyen bir araştırmacı internet üzerinde alacağı randevu saatinde, kendisine verilen kullanıcı adı, şifre ve randevu koduyla sanal laboratuvar kaynaklarına ulaşabilir. Bu randevu saatlerinde kendi sisteminde yazdığı kod imajlarını, özellikleri sunulan WSN düğümlere yine web ortamından yükleyerek çalıştırabilir, elde edilen sonuçları alıp

değerlendirebilir. Şekil 1'den de görüleceği üzere bu sanal laboratuvarın gerçek TelosB WSN düğümleri internet ortamına, KGC olarak düzenlenmiş Raspberry Pi gömülü sistem kenar cihazlar ve bunların koordinasyonunu/işletimini sağlayan sunucu sistem üzerinden çıkmaktadırlar. KGC'ler sadece yönlendirici olarak değil aynı zamanda düğüm kodu yükleyici ve sink düğüm yöneticisi olarak da konfigüre edilmişlerdir. Kodlama platformu olarak açık kaynak kod TinyOS platformu tercih edilmiştir. Ücretsiz olarak indirilen, Windows ve Linux kernel temelli işletim sistemlerinde çalıştırılabilen bu platformda, düğüm kodları yazılabilmekte ve derleme sonucunda bu kodlara ait imaj dosyaları elde edilebilmektedir. Bu imaj dosyaları, sunulan sanal laboratuvar ortamındaki gerçek TelosB WSN düğümler üzerinde çalışmakta ve gerçek sensörler ile denenebilmektedir. Şekil 2, istemcilere bu imkanı sağlayan geliştirilmiş sistem alt yapı modelini göstermektedir. Geliştirilen imaj yükleme sistemi sunucu ve KGC taraflı olmak üzere iki kısımdan meydana gelmektedir. Sunucu sistemin bir yüzü geliştirilen web modül ve ara yüz ile istemci tarafına bakarken diğer yüzü sanal laboratuvar alt yapısına bakmaktadır. Sunucu sistem, öncelikle istemcinin tercih ettiği WSN kaynaklarının gerçek adreslemelerini çözümlenmekte ve kod imajlarının ilgili düğümlere yüklemek için geçici olarak depolamaktadır. İmaj yükleme işlemi



Şekil 1. Geliştirilen test platformu - FıratWSN'in genel yapısı (The general structure of the testbed platform-FıratWSN)



Şekil 2. Geliştirilen kod imajı yükleme sistemi
(The developed system enabling the coding images to be uploaded to WSN nodes)

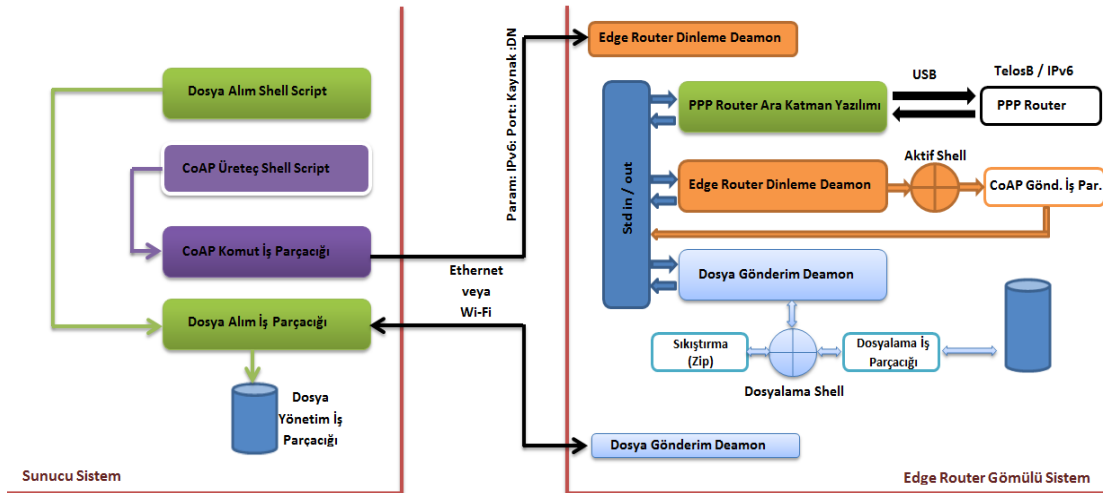
başlatıldığında, sunucu sistem, gerekli yüklemeleri yapacak olan yükleme delege iş parçacıklarını, yazılmış betik dosyaları vasıtasıyla aktif hale getirecektir. Betik dosyaları burada aynı zamanda web ortamından gelecek olan ve ana altyapıya yönelik güvenlik tehditleri içinde bir izolasyon vazifesi sağlamaktadır.

Sunucu sistem ile KGC 'ler arasındaki bağlantı Ethernet ve WiFi teknolojileri üzerinden sağlanmakta olup, oluşabilecek alt yapı değişiklikleri (IP adres değişikliği, KGC değişikliği vb.) sunucu sistem tarafından algılanmakta ve gerekli haritalama işlemleri otomatik olarak yapılmaktadır. KGC üzerinde ise yükleme işlemini yapacak olan düğüm yükleyici iş parçacıkları (DYD) tarafından gerçekleştirilmektedir. İmaj dosyaların yüklenmesi, "TinyOS-tsl operasyonunu" gerçekleştiren iş parçacıkları tarafından yapılmaktadır. Burada da güvenlik amaçlı betik kod dosyaları bulunmaktadır. Bu dosyalar özellikle ilgili DYD 'lerin aktif edilmesinde görev almaktadırlar. Daha sonra açıklanacağı üzere farklı kullanıcılara ait farklı uygulamaların aynı düğüme yüklenmesi durumunda, "ardışıl uygulama yükleyici-AUY" arakatman yazılımı görev alacaktır. Öte yandan her KGC 'da bulunmamakla birlikte, belirli düğümler aynı zamanda sink (toplayıcı) düğüm özelliği de taşıyabilir. Bu düğümlerin olduğu KGC 'lerde toplanan verilerin depo edilmesi "Serial Forwarder " ve bununla iletişim kuran depolama iş parçacıkları ile sağlanmaktadır. Toplanan her veri istemci oturumu sonlandığında KGC 'den kaldırılmakta ve sunucu sistemde depolanmaktadır. KGC ve TelosB düğümler arasındaki bağlantı USB üzerinden sağlanmaktadır.

Sanal laboratuvar modülünün bir diğer önemli özelliği CoAP [23] protokolüne ve IPv6 teknolojisine verdiği destektir. 802.15.4 temelli 6LoWPAN ağların [24] IoT düğümleri haline gelebilmesini sağlayan IPv6, geleceğin M2M sistemleri için önemli bir adresleme altyapısıdır. Sensör düğümlerin internet ortamına çıkabilmesine olanak sağlayan bu teknolojiler yönlendirme için RPL protokolünü kullanmaktadır. Taşıyıcı katman protokolü olarak UDP 'i

kullanan CoAP protokolü HTTP protokolünün kısıtlı kaynaklı cihazlar için optimize edilmiş şeklidir. Böylece günümüzün önemli IoT teknolojilerinden olan RESTful web servislerinin [25] WSN düğümler üzerinde çalışması sağlanabilmektedir. Bu teknolojiler sayesinde temel GET, POST, PUT, DELETE gibi HTTP metotları ile sensör ve aktuatör operasyonları gerçekleştirilebilmektedir. Geliştirilen sanal laboratuvar modülü, bu IoT teknolojisinin kullanılabilmesi için aynı zamanda bir CoAP Edge Router kenar cihazda (CERKC) içermektedir. CERKC çalışma prensibi ve sunucu sistemle olan iletişim şekli Şekil 3'de verilmiştir. Sunucu sistem, CoAP uygulamaları için iki işlemi birbirinden ayırmıştır. Bunlar CoAP komut işletici ve sonuç bildirimlerinin dosyalanması sağlayan operasyonlarıdır.

Sunucu sistemde bunların aktif edilmesi ve yönetilmesi yine betik kod dosyaları ile sağlanmaktadır. CERKC, sunucu sistem ile haberleşmede belirli parametrelere ihtiyaç duymaktadır. Bu parametreler CoAP ve dosyalama için gerekli olan temel bilgileri içerir. CERKC 'lerde iki deamon farklı portlar üzerinde aktiftir. "Edge Router Dinleme" deamonları aktif betik kod dosyaları ile CoAP komut gönderici iş parçacıklarını aktif etmektedir. Edge Router 'lar IPv6 düğümler ile TinyOS PPP Router yazılımı çalıştıran TelosB düğüm üzerinden iletişime geçmektedirler. TinyOS temelli bir coap-client uygulaması çalıştıran CERKC 'de alt modüllerin birbirleri ile haberleşmeleri veya çıktı alımları gömülü işletim sisteminin standart giriş çıkışı özelliği ile sağlanmaktadır. Bu hem çalışma yapısını basitleştirmekte hem de iyi bir loglama performansı sağlamaktadır. Dosya gönderim deamon 'ın görevi, CoAP sorgusu sonucunda elde edilen verileri dosyalamak, log işlemlerini yürütmek ve sunucu tarafından istenildiğinde ilgili parametreler doğrultusunda bu verileri sıkıştırarak geri göndermektir. Bu amaçla dosya gönderim deamon dosyalama iş parçacığı ve sıkıştırma iş parçacığı ile iletişim halindedir. KGC 'lere benzer şekilde istemci oturumu sonunda o oturuma ait veriler CERKC üzerinde silinmekte ve sunucuda istemci klasöründe tutulmaktadır. Bunun yürütümü de yine dosya gönderim



Şekil 3. CoAP Edge Router KGC iletişim alt yapısı (The communication infrastructure of CoAP Edge Router)

daemon tarafından kotarılmaktadır. Geliştirilen test platformu ve sanal WSN yönetim yazılımının çalışmasına örnek olması amacıyla bir senaryo oluşturulmuştur.

Uygulama Senaryosu 1 (The implementation of Scenario 1)

Bu senaryoda uzaktan erişim yöntemi ile test platformunun nasıl kullanılacağı ve IPv6 destekli CoAP uygulamalarının nasıl çalıştırılacağı açıklanmıştır. Geliştirilen sanal WSN laboratuvarı platformunun kullanıcı kodlarını/algortmalarını test edebilmesi için, ilgili kodların kullanıcı tarafından kendi tarafındaki TinyOS ortamında kodlanmış ve imaj dosyasının oluşturulmuş olması gerekmektedir. Daha sonra bu imaj dosyaları, sunucu tarafındaki web arayüzü üzerinden seçilen düğümlere yüklenerek test edilmektedir. Şekil 4'de sanal laboratuvar modülü üzerinde düğümlere kod imajlarının yüklenmesi, sistemin işleyiş süreci ve sonuçların kullanıcıya döndürülmesi süreci gösterilmiştir.

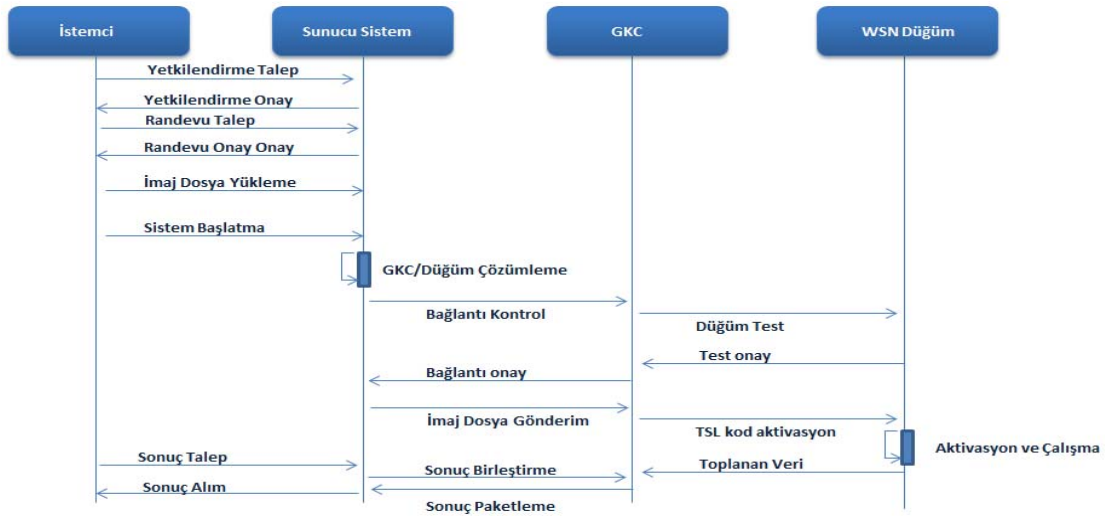
Şekil 4'de gösterilen sürecin, önceden sisteme üye olmuş bir uzak kullanıcı tarafından sunucu üzerinde adım adım gerçekleştirilme aşamaları Şekil 5'de verilmiştir. Buna göre sistemi uzaktan kullanacak olan kullanıcı önceden kendisine verilen kullanıcı adı ve şifre ile ana sayfaya ulaşır ve randevu kodu almak için uygun bir çalışma saati aralığı belirler (Şekil 5a, b, c). Elde ettiği benzersiz randevu kodu ile çalışma zamanı gelince, ana sayfada bulunan dosya yükleme ve sistem başlatma menüsü aracılığı ile TinyOS ortamında yazılmış ve derlenmiş olan düğüm dosyasını (.ihex) seçilen hedef düğüm için sisteme yükler (Şekil 5e). Bu süreçlerden sonra ilgili düğüm kod imaj dosyaları önce web arayüzü üzerinden sunucuda geçici yüklemeye klasörüne yüklenir. Sistem istemci tarafından aktif edildikten sonra sunucu sistem, ilgili GKC/ düğüm ve çalıştırılan portların çözümlenmelerini yaparak gerekli test işlemlerinin gerçekleştirilmesini sağlar.

Test sonucunun olumlu sonuçlanmasından sonra sunucudaki geçici klasörde bulunan düğüm imaj dosyaları ilgili GKC

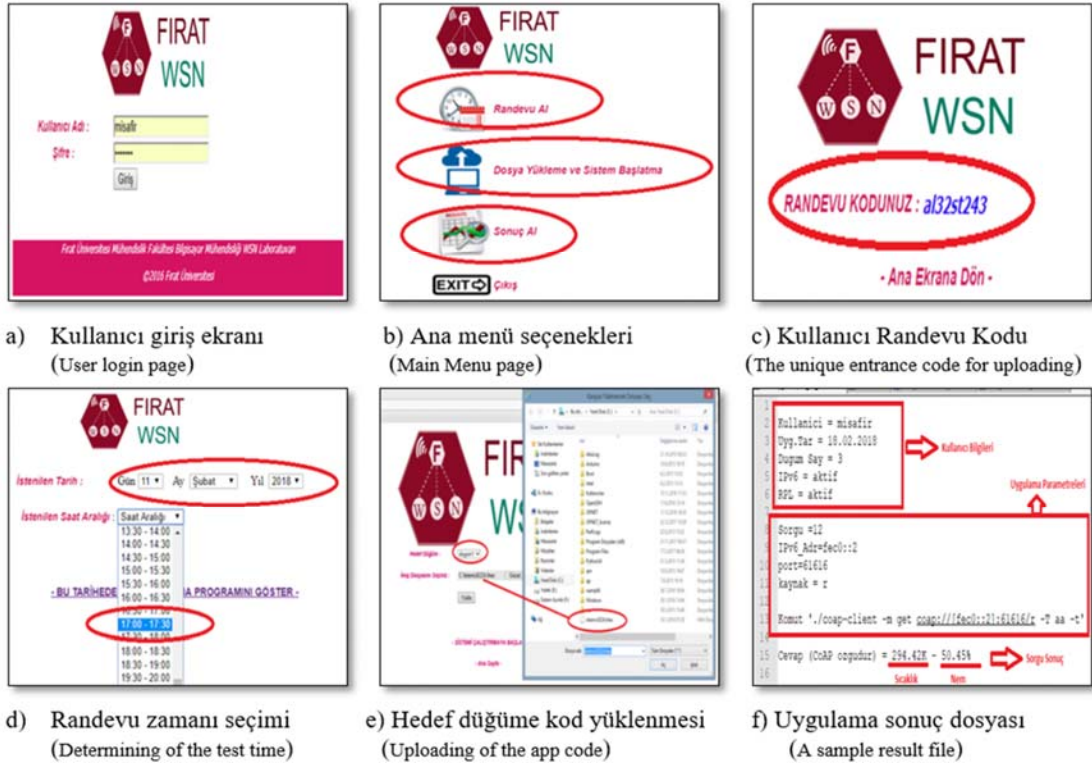
'lara ve oradaki daemon ve diğer iş parçacıkları yoluyla düğümlere yüklenir ve düğümler programlandıkları şekilde çalışmaya başlar. Uzak kullanıcı daha sonra sonuç al menüsü aracılığı ile deney sonuç dosyasını kendi tarafına indirebilmektedir. Şekil.5.f'de gösterildiği gibi, indirilen sonuç dosyasının içeriği iki kısımdan oluşmaktadır. Birinci kısım kullanıcı ve uygulama bilgilerini, ikinci kısım deney verilerini göstermektedir. İkinci kısım tamamen kullanıcının veri toplama modeline göre oluşturulduğundan bu verilerin parçalanması ve anlamlandırılması kullanıcı tanımlıdır.

Şekil 6'da ise CoAP temelli IoT uygulamalarına sanal laboratuvar modülü üzerinden nasıl ulaşılacağı gösterilmektedir. Bunun için iki yol mevcuttur. Bunlardan ilki düğümlerin IPv6 adreslerinin internet ortamında statik olarak tanımlanmış/alınmış olması durumunda gerçekleşmektedir. Bu yolun avantajı CoAP sorgularının internet üzerinden doğrudan Edge Router üzerinden kullanılabilmesine imkân vermesidir. İkincisi ise alt ağ IPv6 adresleme kullanılmaktadır. Bu yöntemde, sanal laboratuvar web sayfası üzerinden gerekli parametrelerin girilmesi ile düğümlere ulaşılabilir. Örnek uygulama senaryosunda CoAP GET metodu için web sayfasında düğüm IPv6 adresi: fec0:2, port adres: 61616, kaynak URI : r, sorgu no parametresi : 12 olarak girilmiştir. Bu parametrelerin girilmesi ile sunucudaki CoAP komut komponentleri, coap-client uygulamasını çalıştırır ve oluşturulacak komut dizgisi, ilgili düğüme gönderilir.

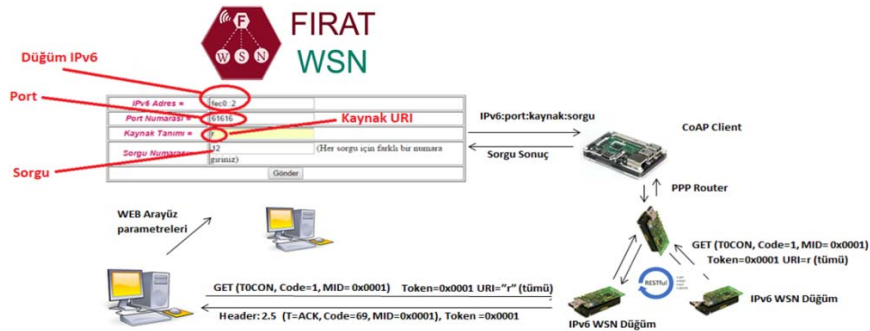
Şekil 7'de uzak kullanıcı tarafından yapılan IPv6/CoAP test sonuçları verilmiştir. Şekil 7a 'da farklı veri boyutları için paket gecikmeleri gösterilmiştir. Her ne kadar WSN sistemlerde 100 byte 'ın üstündeki mesaj boyutları, skaler (sıcaklık, nem gibi temel sensör okumaları içeren) uygulamalar için tercih edilmese de sistem performansının test edilmesi için farklı veri boyutları ile bu gecikme değerleri kaydedilmiştir. Bununla beraber aynı test şartları için ağ performans kriterlerinden olan paket alım oranı (PRR), farklı paket gönderim aralıkları için test edilmiş ve sonuçları Şekil 7b'de paylaşılmıştır. Sistem 5 sn ve üzeri



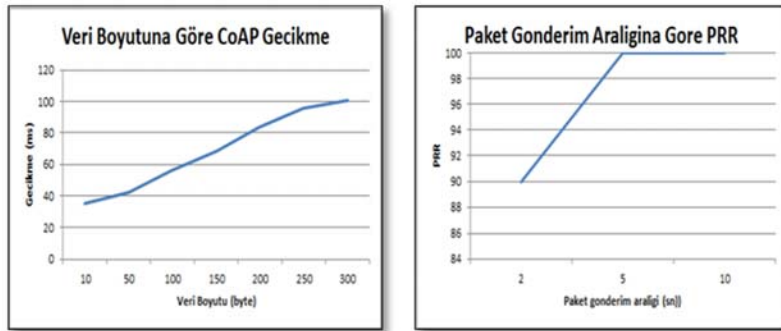
Şekil 4. Düğüm kod imajlarının yüklenmesi ve sonuç akış süreci (The process of code uploading and result acquisition)



Şekil 5. Web arayüzü ile WSN düğümlere kod yüklenmesi (Uploading of WSN node codes into the nodes via the web GUI)



Şekil 6. Sanal WSN laboratuvar modülünde CoAP sorgusu (The CoAP query in the virtual WSN laboratory module)



Şekil 7. Uzak kullanıcıli CoAP/IPv6 test sonuçları (The test results of CoAP/IPv6 applications with remote users)

paket gönderim aralıklarında % 100 değerine ulaşabilmektedir. Tabii ki bu değer yüksek olmasında

düğüm sayısının sınırlı ve kullanıcı spesifik uygulamanın yürütülmesi etkin olmuştur.

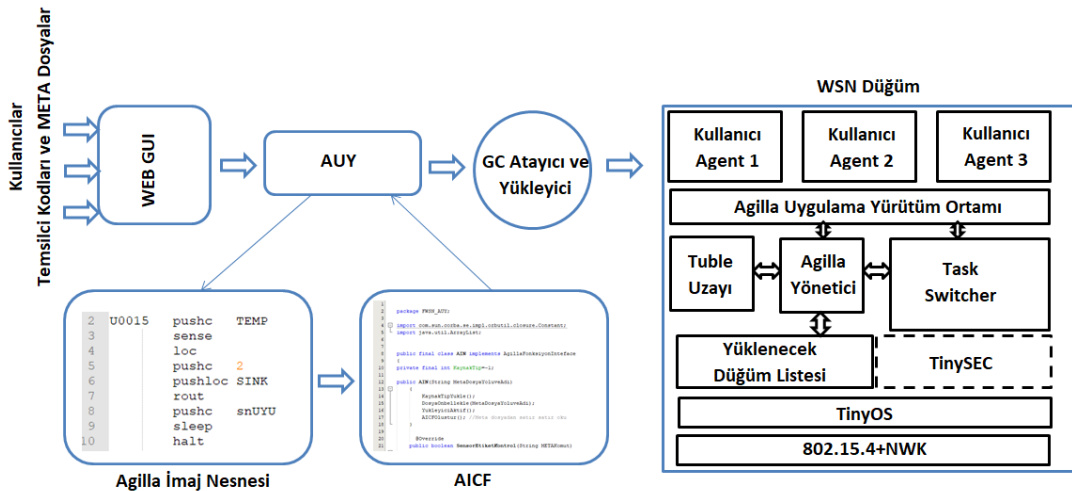
3.1.2. WSN düğümlerde çoklu uygulamalar için sanallaştırma modülü (Virtualization module for multi applications running on WSN nodes)

Şekil 1'deki sanal laboratuvar fiziksel birimiyle etkileşen bu modül ile birden fazla uygulamanın, programlanabilir düğümler üzerinde ardışık (sequential) yürütme tekniğine göre test edilebilmesine olanak sağlanmıştır. Önerilen test platformunun en özgün yanlarından biri de budur. Ardışık çalışma tekniği, bilinen düğüm bazlı WSN sanallaştırma teknikleri arasında yer almaktadır [26, 27]. Bu servis TinyOS tabanlı "Agilla" arakatman yazılımı (middleware-MW) ile sağlanmaktadır [28, 29]. Agilla MW, mobil temsilci (mobile agent) olarak adlandırılan yazılımlar üzerinden bir düğüm üzerinde birden fazla uygulamanın çalıştırılmasına, ağ genelinde kod taşınabilmesine ve WSN ağ düğümlerini yeniden programlanabilmelerine imkân vermektedir. TinyOS 1.x ve 2.x versiyon desteği sunan bu arakatman yazılımı assembly benzeri bir programlama dili kullanmaktadır. Önerilen sistem, bu arakatman yazılımını (middleware-MW) ağ genelinde, kod taşınabilirliğinden ziyade, Agilla uyumlu birden fazla uygulamanın (temsilcinin/agent) bir düğüm üzerinde çalıştırılabilmesine olanak sağlaması için kullanılmıştır. Çoklu uygulama çalıştırma için çalışma takvimi ve zamanları kullanıcılara web üzerinden bildirilmektedir. Çoklu uygulama destekli uygulamalarında öncelikle tüm düğümlere Agilla destekli TinyOS ile yüklü durumdadırlar. Önerilen test platformu, farklı uygulama temsilcilerinin düğümlere yüklenebilmesi için JAVA 'da geliştirilmiş ardışık uygulama yükleyici-AUY yazılımı kullanmaktadır. AUY, ilgili uygulama temsilcisi için uygun "Agilla Injection Configuration File-AICF" dosyasını oluşturmaktadır. AICF dosyaları ilgili kullanıcıların hazırlanmış olduğu düğüm kod meta dosyalarından

elde edilmektedir. Bu işlem ise AUY altında çalıştırılan "Agilla İmaj Nesnesi -AİN" tarafından yapılmaktadır. Bu meta dosyaları aslında Agilla tarafından sıklıkla kullanılan adresleme ve etiket bilgilerinin soyutlanmış biçimleridir.

Örnek olarak düğüm üzerinde bulunan sensörlerin (TEMP, HUM vs.), led tiplerini (1:Kırmızı, 2:Yeşil vs.) tanımlama bilgilerini ön tanımlı olarak "META ETİKET" ler ile kullanıcı kodlarında kullanılabilmesi temsilci (agent) kodlarının AUY-AİN tarafından yorumlanabilir olmasını sağlamaktadır. Böylece AUY, ilgili kullanıcı temsilci kodlarını, paylaşımlı hafıza uzayına (tuple space) sorunsuzca yükleyebilmesine imkân vermektedir. Uygulama temsilcileri ile WSN düğüm kaynak paylaşımına ait genel çalışma modeli Şekil 8'de verilmiştir.

Uygulama temsilcileri (application agents), WSN düğüm üzerinde paylaşımlı hafıza üzerinde çalıştırılmaktadırlar ve bu olay "Tuple Space (Değişkenler Grubu Uzayı)" işlemleri olarak adlandırılır. Bu durumda, her bir uygulama kendine ait bir "tuple field" a sahip olmaktadır. Bu tuple field 'lar hem lokal hem de uzak erişimlerle ulaşılabilir özelliğe sahiptirler. Her ne kadar aynı düğüm üzerinde farklı temsilciler (agent) birbirleri ile haberleşebiliyor olsalar da önerilen test platformu performans temelli hataların önüne geçmek için AUY üzerinden bu işlemi engellemektedir. Agilla MW, farklı uygulama temsilcilerinin beraber çalışabilirliğini bir sanal makine kernel (Agilla Yönetici-Agilla Engine) üzerinden gerçekleştirmektedir. Her bir sanal makine kernel operasyonu, ayrı bir TinyOS task 'i olarak oluşturulur ve bu task 'ler Round-Robin çalışma prensibine göre işletilir. Görevler arasındaki geçişler ve yürütüm düzeni görev değiştirici (Task Switcher) tarafından yürütülmektedir. AUY yazılımı web üzerinden sisteme yüklenen kullanıcı Agilla kod dosyalarını alarak AICF dosyasını oluşturacak olan bir "Agilla İmaj Nesnesi-AİN" yazılımını çalıştırır. Bu yazılım kullanıcı uygulamalarının uygun tuple space 'de bulunmalarını ve olay çağrılarının sorunsuzca koşurulmasını sağlamaktadır. Bir diğer görevi ise yükleniecek kullanıcı uygulamasının doğru düğüm adresi ile etiketlenmesini sağlamaktır. Agilla MW düğümlerin lokasyonunu (x,y) etiketine göre tutmaktadır. Bu amaçla web üzerinden bildirilen düğüm konumları, Agilla tanımlı konumlara bu yazılım aracılığı ile dönüştürülmektedir. Kullanıcı kodlarının çağrımı, düğümler arasında taşınması



Şekil 8. Eş zamanlı çalışma modu için kaynak paylaşım modeli (The resource-sharing model for concurrent operation mode)

ve bu işlemlerin düzenlenmesi ilgili düğümün donanımsal özellikleri ile doğrudan alakalı olduğundan yükleme esnasındaki sorunlar yine bu yazılım tarafından kontrol edilmektedir. Oluşturulan AICF, bash tabanlı bir alt yazılım (GC atayıcı ve yükleyici) ile kenar cihaza (raspberry pi) bağlı TelosB düğümlere yüklenmektedir. Sistem tabanlı hatalar ve geri bildirimler yine bu ara yazılım sayesinde AUY 'a iletilmektedir. Çoklu uygulamaların yanlış adreslemesinde veya çağrım hatalarının olacağını bildiren hata kodlarının alınmasında AUY yeni bir AICF hazırlayarak tekrar yükleme yazılımını çağırılmaktadır. Ön tanımlı (varsayılan olarak bu değer 3 'dür) hata sayısına ulaşan uygulama kodları için deney süreci yapılmamaktadır. AUY yazılımı altında çalışan AİN ve oluşturulan AICF dosyasına ait örnek çalışma şekli, ve sensör okumalarını toplayıcı düğüme (sink) gönderen örnek bir kullanıcı temsilcisi kod bloğu Şekil 9'da verilmiştir. AİN, öncelikle eş zamanlı çalışacak olan kullanıcı Agilla dosyalarına ait uygun yığıt (stack) adreslemelerini kontrol etmektedir veya atamaktadır. Kullanıcı uygulama tetiklemelerinin hangi sensör kaynaklardan olacağı (varsa) veya ilgili çağrım etiketlerinin çıkışmaması için gerekli düzenleme yine AUY tarafından sağlanmaktadır. AİN, farklı uygulamalar için yükleme ve çalışma esnasında belirli bir zaman gecikmesi sağlamaktadır. Her ne kadar bu gecikme uygulamaların yürütümü esnasında

fazladan gecikmeye neden olsa da WSN düğüm işlemci kısıtlarından dolayı bu zorunlu olmaktadır. AİN yazılımı, optimum AICF 'nin oluşturulmasında hayati rol oynamaktadır. Özellikle, "out (lokal tuple space alanına agent tuple yerleştirme işlemi)", "in (belirli bir tuple 'ın kaldırılması)", "rd (bir tuple 'ın okunması)" gibi tuple space fonksiyonlarının herhangi bir çıkışmaya izin vermeden organize edilmesi AİN 'nin görevlerinden birini teşkil etmektedir. AİN her bir tuple için 18 byte 'lık alanın olduğunu varsaymaktadır ve en fazla 3 tuple için düzenleme yapmaktadır.

Öte yandan varsayılan olarak bu fonksiyonların uzak tuple space işlemleri (rout, rinp,rrdp vs.) için AİN destek sunmamaktadır. Ancak bir sonraki versiyonda uzak agent işlemlerinin desteklenmesi sağlanacaktır. İlgili kullanıcı temsilcisinin (agent) hedef düğüme taşınması ve klonlanması (migration) gömülü sistem kenar cihaza bağlı yardımcı düğüm üzerinden sağlanmaktadır. Bu durumda yüklenecek düğümün koordinat bilgileri (x,y) ve gerekli yükleme komutları (pushloc, wclone vs.) bu bilgilere göre düzenlenmektedir. Benzer şekilde düğüm üzerindeki sensör ve ledlerin adresleme kontrolleri de AİN tarafından düzenlenmektedir. Varsayılan olarak bu işlem AİN 'de düzenlenirken "wclone (weak clone)" temelli yapılmaktadır.

```

public final class AIN implements AgillaFonksiyonInterface
{
    private final int KaynakTip=-1;
    public AIN(String MetaDosyaYoluveAdi)
    {
        KaynakTipYukle();
        DepiYolunuOku(MetaDosyaYoluveAdi);
        YukleyiciAktif();
        AICFOlustur(); //Meta dosyadan hata kodu oku
    }
    private
    public boolean SensörEtiketKontrol (String METAkomut)
    {
        if (METAkomut.SensörBul (METAkomut))
        {
            if (KAYNAKTip==MetaEtiketler.FENK)
            {
                AICFKomutOlustur (MetaEtiketler.SICAKLIK, MetaEtiketler.LOKASYON_LAR_AHA_GUYA); //Varsayılan düğüm-1
            }
            else if (KAYNAKTip==MetaEtiketler.HSD)
            {
                AICFKomutOlustur (MetaEtiketler.FENK, MetaEtiketler.LOKASYON_LAR_AHA_PC);
            }
        }
    }
}

```

Düğüm Interface Fonksiyonları Kullanır
 Meta Dosya Okuma ve Düğüm Yükleyici AKTIF
 Kullanılan Sensör Tipini Belirleme
 AICF Çağrımı oluştur ve Lokasyon Ata

a) Meta dosyasından AICF elde eden AIN sınıfı (AIN Class that handles a META file)

```

BASLA    pushin  U_0015      //Kullanıcı ID ile Etiketle yap
         pusht   1 2        // Hedef dugum Agilla Koordinat AIN den
         pushc  T_KOPYA_SBT //Stack 'de tuple boyutu belirle
         pushc  U0015F      // Kullanıcı fonksiyonu etiketi AIN 'den atanmis
         regrxn // Fonksiyon kayıt
UYU_1s   pushc  8          // sn uyu
         sleep
         rjmp   VAR_MI
U0015F   pop
         wclone
         sense

```

b) AICF dosyası (AICF File)

```

1 //FWSN TelosB Agilla ikinci kullanıcı sıcaklık okuma kod bloğu. 15.02.2018
2
3
4 U0015   pushc  SICAKLIK   //AIN ETIKETI ILE
5         sense           //SENSOR OKU
6         loc            //DUGUM KOORDINAT OKU
7         pushc  2        //OKUNACAK TOPLAM VERI BOYUTU
8         pushloc SINK    // SINK ADRES COZUMLE
9         rout           //GONDER
10        pushc  RED      //YAKILACAK LED
11        putled
12        sleep
13        halt

```

c) AİN dönüşümü yapılmış örnek kullanıcı kodu (A sample user code with an AIN conversion)

Şekil 9. AİN yazılımı, AICF dosyasından örnek görüntüler (Sample screenshots from AIN and AICF file)

Böylece sadece kullanıcı kodları hedef düğüme taşınmakta, ilgili temsilciye ait durum (state) bilgileri gönderilmemektedir. Bu durumda yüklenen istemci kodları en baştan çalışmaya başlamaktadır. Her bir tuple için çalıştırılacak "reaction codes (tetikleme kodları)" için 130 byte 'lık alan kullanılmaktadır. Bu alanın yönetimi Agilla Reaction Manager-ARM" tarafından otomatik yürütülmektedir. Yüklenen kullanıcı temsilcilerin kod boyutları 22 byte 'lık bloklar halinde düğüme saklandığından AİN eş zamanlı çalışacak olan temsilci

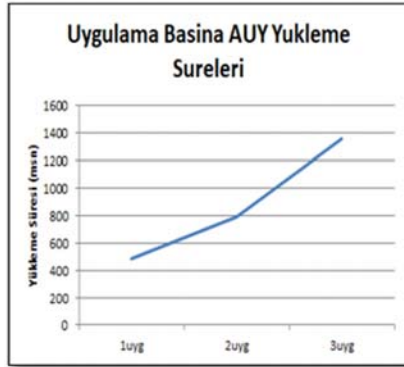
kodlarının uygunluğunu da kontrol etmektedir. Yeterli hafıza boyutu olmadığı durumda kullanıcı temsilci kodları düğüme yüklenmemekte ve hata geri bildirimi üretilmektedir. AİN 'in içerdiği temel fonksiyonlara ait tablo Tablo 2'de verilmektedir.

Uygulama Senaryosu 2 (The implementation of Scenario 2)

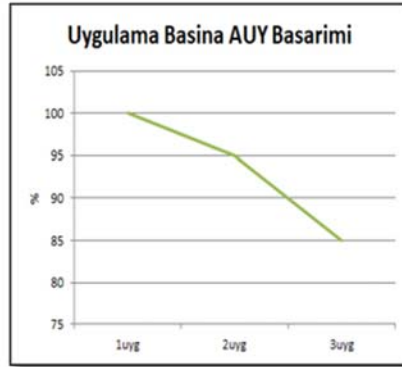
Bu senaryo uygulamasında WSN sanallaştırma yöntemleri ile uzak kullanıcıların aynı düğüm üzerinde ardışık olarak

Tablo 2. Bazı AİN temel fonksiyonları (Some basic functions of AİN)

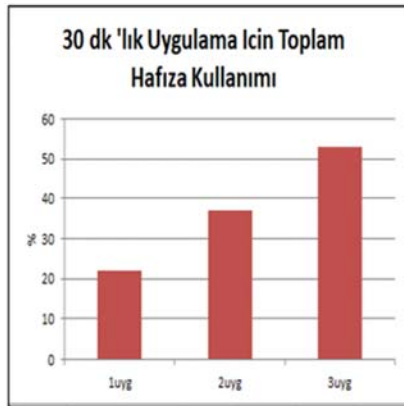
Fonksiyon Adı	İşlevi
KonumEtiketKontrol(String)	Kaynak konum adresini Agilla lokasyon adres uzayına dönüştürür
AICFKomutOlustur(String, String)	Kullanıcı kod ve META dosyasına göre AICF dosyası komutu oluşturur
SensorEtiketKontrol(String)	Kullanıcı kod bloğunda sensör etiketlerini Agilla tanımlı etiketlere dönüştürür
LedEtiketKontrol(String)	Kullanıcı kod bloğunda led etiketlerini Agilla tanımlı etiketlere dönüştürür
BoyutKontrol()	Kullanıcı temsilci kodlarının kod uzayında kapladığı boyutu verir
MetaKomutSensorBul(String)	Kullanıcı temsilci kod bloğunda sensör kaynak tanımlamasını çözümler
MetaKomutLedBul(String)	Kullanıcı temsilci kod bloğunda led kaynak tanımlamasını çözümler
MetaKomutReaction(String)	Kullanıcı temsilci kod bloğunda Reaction kod bloğu başlangıcını verir



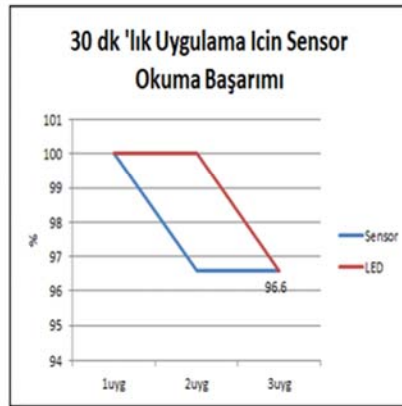
a) Toplam AUY işlem gecikmesi
(Total AUY Delay)



b) Ortalama AUY Başarım oranı
(Average AUY success rate)



c) Ortalama düğüm hafıza kullanımı
(Average Node memory usage)



d) Uygulama kaynak işleme başarımı
(Success rate of resource handling)

Şekil 10. Kaynak paylaşımlı çoklu uygulama test sonuçları (The test results of the multi applications with resource sharing)

farklı uygulamaları gerçekleştirebilmesi açıklanmıştır. Bunun için ilk uygulama skaler sensör/aktuatör uygulaması iken diğer uygulamalar yine skaler sensör okuması yapan ve belirlenen okuma sayısı içerisinde elde edilen verilere göre “data aggregation algoritması” çalıştıran uygulamalardır. Hazırlanan kod imajları ve sisteme yükleme Uygulama Senaryosu 1 ‘de anlatıldığı şekilde yapılmaktadır. Aynı düğüm üzerinde ikinci kod yüklenmesi talebi durumunda KGC üzerinde çalışan bir kontrol iş parçacığı bu talebi AİN ‘e parametre olarak bildirmektedir. Bu senaryoya ait performans test sonuçları Şekil 10‘da verilmiştir. İlk olarak (a) ayrı kullanıcı uygulamalarının KGC üzerinden hedef düğümlere yüklenme süreleri verilmiştir. Bu süre içerisinde tüm AUY işlemleri de dâhildir. Bu işlemler ilgili kullanıcı dosyalarının okunması, çözülmesi, AICF dosyalarının oluşturulmaları ve yüklenme süreçleridir. Şekil.10.a da görüleceği üzere, bu süre, uygulama sayısı ile neredeyse doğru orantılı bir artış göstermektedir. Kaynak kısıtlarından dolayı uygulama sayısı 3 ile sınırlandırılmıştır. Şekil.10.b ‘de toplam 20 kullanıcı uygulaması yükleme denemesi için elde edilen başarılı yükleme sonuçları paylaşılmıştır. Burada arka arkaya yüklenen kodlar, aynı uygulamanın aynı kodları olup yüklemelerle ilgili sonuçlar TelosB düğümlerin silinip tekrar yüklenmesine göre belirlenmiştir. Uygulama başarımının yüksek sayılabilecek bir değerde olduğu görülmektedir. Öte yandan Şekil.10.c ve d’ de ise ilgili senaryodaki tüm uygulamalar için TelosB düğümlerin toplam kullanılan ortalama hafıza oranları ve senaryodaki sensör/aktuatör (Led) uygulaması için kaynak operasyonlarının yürütüm başarımı verilmiştir.

3.2. FMonitör, FıratZigBee Modülü (FMonitor, FıratZigBee Module)

Standartlaşmış olması nedeni ile ZigBee WSN teknolojisi, özellikle sahada, akıllı otomasyon uygulamalarında sıklıkla tercih edilmektedir. Bu yüzden daha çok lisans öğrencilerinin bu tip uygulamalarına yönelik olarak, önerilen *FıratWSN* platformuna, ortamsal gözleme uygulamalarında kullanılmak üzere, mevcut Fırat ZigBee isimli bir alt WSN sistemi de dâhil edilmiştir [30]. Fırat ZigBee alt WSN yapısı, 802.15.4 temelli ZigBee teknolojisine ve mesh topolojisine göre çalışmaktadır. FıratZigBee WSN sistemi, “FNode” düğümleri, “FMonitör” gözleme/kontrol yazılımı, “FİoT ETH”-“FİoT GSM” İoT entegrasyon cihazları ve NoSQL veri tabanı bileşenlerinden oluşmaktadır [30].

3.3. 802.15.4 Alt Ağı (802.15.4 Sub WSN)

Geliştirilen test platformu heterojenliğin sağlanması açısından TelosB sanal ağ ve FıratZigBee alt ağlarının dışında farklı teknoloji alt ağlarda içermektedir. Bunlardan biri Libelium firması tarafından geliştirilen “waspmote” düğümlerden oluşan 802.15.4 temelli alt ağıdır. Bu alt ağ, tak çalıştır sensör/aktuatör kartlar sayesinde birçok farklı uygulama için esnek bir yapı sunmaktadır. Uzaktan kod yükleme desteği içermeyen bu alt ağ daha çok sağlanan API ‘ler yardımı ile belirlenen lokasyonlardaki ölçüm verilerinin

diğer uygulamalar (lokal veya uzak) ile paylaşılmasına yönelik kullanılmaktadır. Geliştirilmekte olan sanallaştırma teknikleri ile bu alt ağlarında kullanıcı kodları ile değerlendirilebilir olması amaçlanmaktadır

3.4. ZigBee Alt Ağ (ZigBee Sub WSN)

Heterojen İoT-WSN platformunun bir diğer alt ağı ise NXP/Freescale tarafından geliştirilen 13226 NCP/SRB ZigBee düğümlerdir. Bu alt ağ daha çok public (kamusal) ve private (özel/ticari) uygulama katmanlarının geliştirilmesinde kullanılmaktadır. Bu sayede farklı ZigBee uygulama projelerinin İoT sistemlere entegrasyonu sağlanan API ‘ler veya web servisler sayesinde sağlanabilmektedir. Şu an için “Zigbee home automation application” desteği sunan bu alt ağ geliştirilebilen farklı uygulama katmanları ile test platformuna esnek bir çalışma ortamı sağlamaktadır.

3.5. NoSQL Veritabanı Yönetim Modülü (NoSQL Database Management Module)

WSN’ler için diğer bir önemli konu ise çok sayıda düğümden toplanabilecek veya çok sık veri okumalarından kaynaklanacak büyük boyutlu verilerdir. Bu büyük boyutlu verilerin depolanmaları, işlenebilmeleri, kullanılacak dosya sistemleri, okuma-yazma işlemleri, hata toleransları, sürdürülebilirlikleri gibi temel yürütüm fonksiyonları özellikle İoT WSN’ler için son derece önemlidir. Büyük verilerin oluşması, her geçen gün sisteme dâhil olan düğümlerin artması beraberinde ölçekleme, bakım-onarım, hata toleransı gibi temel gereksinimleri getirecektir.

Bu nedenle en uygun yapı NoSql veri tabanlarının tercih edilmesidir. Bununla birlikte bir diğer konu da bu verilerin analiz yöntemlerinin nasıl olacağıdır. Büyük boyutlu verilerin analizlerinde günümüzde paralel hesaplama teknolojileri büyük kolaylıklar sağlamaktadır. Bu makalenin yazarlarının da çalışmalarında kullanmış olduğu MapReduce frameworku bu alanda oldukça söz sahibidir [31]. Bu nedenle FıratWSN sisteminde, klasik text veri depolamanın dışında MapReduce desteği sunan CouchDB [32] NoSql veritabanı imkânı da kullanıcılara sunulmaktadır. CouchDB verileri JSON formatında saklamaktadır ve HTTP ve JavaScript desteği de sunan mobil uygulamalar içinde kullanılabilen bir veri tabanıdır. FıratWSN uygulamalarında XML’e göre daha hafif ve esnek olan JSON formatı tercih edilmiştir. FıratWSN’de kullanılan diğer veri depolama birimleri daha önceden de söylendiği gibi FİoT ETH ve FİoT GSM modülleridir. Bu modüller verileri bağlantılı veya bağlantısız şartlarda saklayabilmekte ve gerektiğinde bu verileri ana sistem NoSql veri tabanına gönderebilmektedir.

Geliştirilmiş olan FıratWSN sanal laboratuvar sistemine ve alt modüllerine ait bazı genel görüntüler Şekil 11’de verilmiştir. Bu resimlerde sırası ile gömülü ve kenar sistemler üzerindeki programlanabilir TelosB düğümler (a, b), gömülü sistemler üzerinde çalışan DYD yürütümü (c), çoklu uygulama yürütüm yazılımı AUY ve AİN ‘e ait gömülü sistem yürütümü (d), FıratZigBee alt ağına ait bir



Şekil 11. FiratWSN'den ve alt sistemlerden bazı görüntüler (Some images form FiratWSN and its sub WSNs)

bina içi FNode (e) ve test platformunda kullanılan farklı teknolojili örnek WSN düğümler görülmektedir.

Bu şekilde gerçekleştirilmiş olan ve İstemci/Sunucu modeline benzer olarak çalışan FiratWSN sanal laboratuvar ve test sistemi yukarıda açıklanan özellikleri kapsayan bir düzende kullanıcılarına hizmet vermekte olup, fiziksel yapısı ve gelişim süreci devam etmektedir.

4. SONUÇLAR (CONCLUSIONS)

Bu çalışmada, günümüz WSN teknolojilerinin gelmiş olduğu noktadaki gereksinimleri karşılayabilen, IoT destekli, lokal ve uzak kullanıcılara hizmet verebilen bir sanal WSN laboratuvarı ve test platformu olarak geliştirilmiş olan FiratWSN sistemi açıklanmıştır. Bu platform bu alanda çalışan araştırmacı, eğitimi ve öğrencilere hem zaman hem de ekonomik açıdan pek çok imkân sunmaktadır. Sunduğu sanal laboratuvar imkânı ile bir istemcinin her hangi bir WSN donanımına ücret ödmeden veya bir kablosuz duyurğa ağı dizaynı için zaman harcamadan, yazmış olduğu kodları/protokolleri, internet üzerinden bu laboratuvar kapsamındaki gerçek WSN düğümler üzerinde denemesi mümkün kılınmıştır. Literatürdeki örneklerinden farklı olarak bu sanal laboratuvar, programlanabilir WSN düğümler üzerinde belirli sayıda farklı kullanıcı uygulamalarının eş zamanlı çalıştırılmasına olanak sağlayan bir alt yapı desteği sunmaktadır. Bu özelliği sayesinde daha fazla kullanıcının verimli bir şekilde kendi uygulamalarını denemeleri mümkün olabilmektedir. Sistem bu alt yapıyı Agilla arakatman yazılımı teknolojisi ile sağlamaktadır. Öte yandan bu sanal laboratuvar, istemcilerine CoAP temelli

uygulamaları da test imkânı da sunmaktadır. Ayrıca bu WSN test platformu kapsamında geliştirilen FNode'lar ve FMonitör yazılımının kapsadığı FiratZigBee olarak adlandırılan ikinci bir alt WSN sistemi ile ZigBee, ve 802.15.4 teknolojilerine göre oluşturulabilecek alt ağlar kullanarak heterojen WSN uygulamaları da kolaylıkla yapılabilmektedir. Kurulan laboratuvarda, WSN-IoT entegrasyonun yanı sıra elde edilen verilerin depolanması ve analizi üzerine de uygulamalar geliştirilmesi de amaçlanmıştır. Verilerin depolanmasına yönelik olarak NoSQL veri tabanı teknolojileri tercih edilmiştir.

Büyük ölçekli WSN projelerinin bir parçası olabilecek şekilde tesis edilen ve İstemci/Sunucu yapısına uygun olan FiratWSN sisteminin, farklı coğrafyalardaki farklı merkezlerde bulunan mevcut WSN araştırma laboratuvarları ile internet üzerinden daha büyük bir Sanal WSN test ortamı oluşturabileceği ve dolayısıyla daha çok araştırmacıya, daha az masrafla daha büyük imkânlar sağlayacağı açıktır. Bu fikrin gerçekleştirilebilir ve üzerinde araştırmaya değer bir konu olduğu rahatlıkla söylenebilir.

TEŞEKKÜR (ACKNOWLEDGEMENT)

MF.16.19 - Sanal Kablosuz Duyurğa Ağ Tabanlı Bir Dağıtık-Paralel Siber Fiziksel Sistem Tasarımı isimli proje desteği için Firat Üniversitesi Bilimsel Araştırma Projeleri Koordinasyon Birimine teşekkürlerimizi sunarız. Bu çalışma, makale yazarlarından Güngör Yıldırım 'ın "Sanal Kablosuz Duyurğa Ağ Tabanlı Bir Dağıtık-Paralel Siber Fiziksel Sistem Tasarımı" isimli doktora tezinden türetilmiştir.

KAYNAKLAR (REFERENCES)

1. Iwanicki K., Gaba A., Steen M., KonTest: A wireless sensor network testbed at Vrije Universiteit Amsterdam, Vrije Universiteit, Amsterdam, Development Laboratories (DevLab), Technical Report, Eindhoven, The Netherlands, 2008.
2. Hergenröder A., Wilke J., Meier D., Distributed energy measurements in WSN testbeds with a sensor node management device (SNMD), 23rd Int. Conference Archit. Comput. Syst. Workshop, Hannover, Germany, 1–7, Feb. 2010.
3. Horneber J., Hergenröder A., A Survey on Testbeds and Experimentation Environments for Wireless Sensor Networks, IEEE Communication Surveys & Tutorials, 16 (4), 1820-1839, 2014.
4. Darymli K., Ahmed M. H., Wireless Sensor Network Testbeds: A Survey, Wireless Sensor Networks and Energy Efficiency: Protocols, Routing, IGI Global, USA, doi: 10.4018/978-1-4666-0101-7.ch007, 2012.
5. Fischer S., Pfisterer D., Fekete S. P., Kroeller A., WISEBED—Pan- European wireless sensor network testbeds, Fachgespräch der GI/ITG-Fachgruppe Kommunikation und Verteilte Systeme, Technical Report., 2008.
6. Lattanzi E., Dromedari M., Freschi V., A Scalable Multitasking Wireless Sensor Network Testbed for Monitoring Indoor Human Comfort, IEEE Access, 6,17952-17967, 2018, doi: 10.1109/ACCESS. 2018. 2818191.
7. Kanzaki A., Hara T., Ishi Y., Wakamiya N., Shimojo S., X-Sensor: A sensor network testbed integrating multiple networks, Int. Conf. CISIS, Fukuoka, Japan, 1082–1087, Mar. 2009.
8. Hergenröder A., Horneber J., Wilke J., SANDBed: A WSN testbed for network management and energy monitoring, Hamburg, Technical Report, Germany, Aug. 2009.
9. Ducrocq T., Vandaele J., Mitton N., Simplot R. D., Large scale geolocalization and routing experimentation with the SensLAB testbed, 7th IEEE Int. Conf. MASS, San Francisco, CA, USA, 751–753, 2010, 10.1109/MASS.2010.5663816
10. Allen G. W., Swieskowski P., Welsh M., MoteLab: A wireless sensor network testbed, 4th Int. Conference. IPSN, Apr., Boise, USA, 483–488, 2005, doi: 10.1109/IPSN.2005.1440979.
11. Dimitriou T., Kolokouris J., Zarokostas N., Sensenet: A wireless sensor network testbed, . 10th ACM Conference, MSWiM, Crete Island, Greece, 143–150, 2007.
12. Zhao Z., Yang G.H., Liu Q., Li V.O.K., Cui L., Implementation and Application of a Multi-Radio Wireless Sensor Networks Testbed, IET Wireless Sensor Systems, 1, 4, 191-199, 2011. doi:10.1049/iet-wss.2011.0013.
13. Arora A., Ertin E., Rajiv R., Leal W., Nesterenko M., Kansei: A High-Fidelity Sensing Testbed, IEEE Internet Computing, 10, 2, 35-47, 2006, doi: 10.1109/MIC.2006.37
14. Murty R. N. ., CitySense: A vision for an urban-scale wireless networking testbed, Harvard University, Cambridge, MA, USA, Technical Report, Sep. 2007.
15. Sakamuri D., NetEye: A wireless sensor network testbed,” Master’s thesis, Graduate School of Wayne State University, Detroit, MI, USA, 2008.
16. Rensfelt O., Hermans F., C. Ferm, A. Larzon, Sensei-UU: A nomadic sensor network testbed supporting mobile nodes, Uppsala Universitet, SUppsala, Sweden, Technical Report, Oct. 2009.
17. Lambrou T.P., Panayiotou C.G., A Testbed for Coverage Control Using Mixed Wireless Sensor Networks, Journal of Network and Computer Applications, Vol:35, 527-537, 2012, doi:10.1016/j.jnca.2011.05.010
18. Beshay J.D., Subramani K.S., Mahabeleshwar N., Nourbakhsh E., Wireless Networking Testbed and Emulator (WiNeTestEr), Computer Communications, 73, 99-107, 2016, doi:dx.doi.org/10.1016/j.comcom. 2015.08.007.
19. Kanmaz M., Aydın M.A., Comparison of dv-hop based indoor positioning methods in wireless sensor networks and new approach with k-means ++ clustering method, Journal of the Faculty of Engineering and Architecture of Gazi University, 34 (2), 975-986, 2019.
20. <http://www.tinyos.net/>, 2.03.2018.
21. <http://www.contiki-os.org/>, 2.03.2018.
22. Tatar Y., Yıldırım G., A dynamic location estimation technique based on fingerprint using a reduced radio map in wireless sensor networks, Journal of the Faculty of Engineering and Architecture of Gazi University, 29 (2), 217-226, 2014.
23. <http://coap.technology/>, 2.03.2018
24. Hasbollah A. A., Ariffin S. H. S., Ismi M., Hamini A., Performance analysis for 6LoWPAN IEEE 802.15.4 with IPv6 network, TENCON IEEE Region 10th Conference, Singapore, 2009 DOI: 10.1109/TENCON. 2009. 5396174
25. Mohiuddin J., Bhadram V., Palli S., 6LoWPAN based service discovery and RESTful web accessibility for Internet of Things, Advances in Computing, Communications and Informatics ICACCI, International Conference, New Delhi, India, 2014, doi: 10.1109/ICACCI.2014.6968249
26. Yıldırım G., Sanal Kablosuz Duyarga Ağ Tabanlı Bir Dağıtık-Paralel Siber Fiziksel Sistem Tasarımı, Doktora tezi, Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Elazığ, 2017
27. Yıldırım G., Tatar Y., Simplified Agent-Based Resource Sharing Approach for WSN-WSN Interaction in IoT/CPS Projects, IEEE Access, Vol:6, 78077-78091, 2018, doi: 10.1109/ACCESS.2018.2884741
28. <http://dews.univaq.it/index.php?id=dewsvision0>, Erişim tarihi 2.03.2018.
29. <http://mobilab.wustl.edu/projects/agilla/>, Erişim tarihi 2.03.2018.
30. Yıldırım G., Tatar Y., 802.15.4 Tabanlı Sensör Düzümü Tasarımı ve Geliştirilen Kablosuz Sensör Ağ Uygulama

Ortamı: FıratZigBee, Fırat Üniv. Müh. Bil. Dergisi, 30 (2), 257-267 , 2018.

31. Yıldırım G., Hallac İ. R., Aydın G., Tatar Y., Running genetic algorithms on Hadoop for solving high dimensional optimization problems, Application of

Information and Communication Technologies (AICT) 9th International Conference, Rostov, Russia, 14-16 Oct. 2015, doi: 10.1109/ICAICT.2015.7338506

32. <http://couchdb.apache.org/>, Erişim tarihi 2.03.2018.