

Opnet Modeler Ortamında WSN Protokollerinin Gerçekleştirme Süreçlerinin İncelenmesi

¹Gürkan Doğan, ²Güngör Yıldırım, ³Yetkin Tatar

^{1,2,3} Faculty of Engineering, Department of Computer Engineering Fırat University, Elazığ, Turkey

Özet

Kablosuz sensör ağları (KSA), bilgisayar ağlarına benzer olarak katmanlı bir ağ modeline göre oluşturulur ve çalıştırılırlar. Temelde bu katmanlar fiziksel, ortama erişim kontrolü (MAC), ağ ve uygulama katmanlarıdır. KSA sistemini oluşturan her bir düğümde bu katmanlar için belirlenmiş görevler, ilgili katman protokolleri tarafından gerçekleştirilir. Önerilen yeni bir katman protokolünün veya mevcut protokole yapılacak bir iyileştirmenin ilgili düğüm yapısına ve KSA'ya etkisinin analiz edilmesi için düğümünün bu katmanlı yapıya göre modellenip simüle edilmesi son derece önemlidir. Bu bildiride OPNET Modeler simülasyon ortamında 4 katmanlı WSN ağ modeline uygun düğüm modelleri oluşturulmuş olup, belirlenmiş bir KSA ağına katılım isteğinde bulunan bu düğümlerin ve koordinatör düğümün, ağa katılım sürecindeki davranışları simüle edilerek elde edilen sonuçlar tartışılmıştır.

Anahtar kelimeler : OPNET, Simülasyon, Kablosuz Sensör Ağları, Protokoller.

A Study on The Processes of Realization of WSN Protocols in Opnet Modeler

Abstract

Wireless sensor networks (WSN) are run and created according to a layered network model similar to computer networks. These fundamental layers are physical, medium access control (MAC), network and application layers. In each node that makes up the WSN system, the tasks assigned to these four layers are performed by the corresponding layer protocols. It is quite important that the node is modeled and simulated according to a layered structure in order to analyze the effect of a proposed new layer protocol or any optimization to be made for the existing protocol on the node. In this study, node models were created in OPNET Modeler simulation environment in accordance with 4-layered WSN network model, and the results obtained by simulating the behaviors of all WSN nodes in the process of the joining a WSN are discussed.

Key words: OPNET, Simulation, Wireless Sensor Network, Protocols

1. Giriş

Kablosuz Sensör Ağları (KSA-WSN) günümüz dünyasında yaşam konforunu artırıcı teknolojilerden biri haline gelmiştir. Akıllı ev, akıllı tarım, akıllı sağlık v.b birçok uygulama alanı mevcuttur. IoT (Nesnelerin İnterneti) ve CPS (Siber Fiziksel Sistemler) teknolojilerinin de temel taşı olan WSN'ler artık paylaşılabili sistemler halini almış olup buna paralel olarak, farklı WSN teknolojileri ve protokolleri ile ilgili yeni geliştirmelerin yapılması da zorunlu hale gelmiştir. Bu geliştirme sürecinde, modelleme ve simülasyon işlemlerinin çok önemli bir yer tutacağı açıktır.

*Corresponding author: Yetkin TATAR Address: Faculty of Engineering, Department of Computer Engineering Fırat University, 23119 , Elazığ TURKEY. E-mail address: ytatar@firat.edu.tr, Phone: +904242370000/6302

Onlarca veya yüzlerce düğümden oluşan WSN sistemlerinde katmanlı yapıda düğüm modelleri oluşturulması ve bunlarla ilgili geliştirilecek protokollerin denenmesi süreci simülasyon ve analiz aşamasını gerektirir. WSN sistemleri için birçok simülasyon platformu aracı mevcuttur. Bunlardan en önemlilerinden birisi de OPNET (Optimized Network Engineering Tools) simülasyon platformudur. OPNET simülasyon platformunda düğüm modellemesi yapabilmek için C programlama dili ile birlikte OPNET'in kendi fonksiyonları da kullanılmaktadır. OPNET simülasyon aracı birçok araştırmalarda kullanılmaya devam etmektedir. Güncel örnekler olarak; slotlu CSMA / CA kanal erişim mekanizmasını kullanan ağların MAC katman performansının analiz edilmesi[4], bir Mobil Ad-hoc ağı (MANET) veya Kablosuz Sensör Ağı (WSN) tasarlamak için bir algoritma geliştirilirken ve bazı hizmet kalitesi parametrelerine dayanan farklı ağ ve fiziksel katman saldırılarının (karadelik, bizans vb. gibi) etkilerinin karşılaştırılması[5], kablosuz sensör ağlarının hiyerarşik yönlendirme protokollerinden LEACH, PEGASIS ve TEEN kullanılarak performans değerlendirmesi [6], mevcut SSLE yönteminin geliştirilmiş sürümü olan modifiye bir sektörel tarayıcı tabanlı lokalizasyon tahmin (SSLE) yönteminin simülasyonu [7], ZigBee protokolünü uygulayan bir WSN'in simülasyonu ve uçtan uca gecikmenin farklı ağlara göre değerlendirilmesi [8] gibi araştırma çalışmaları verilebilir.

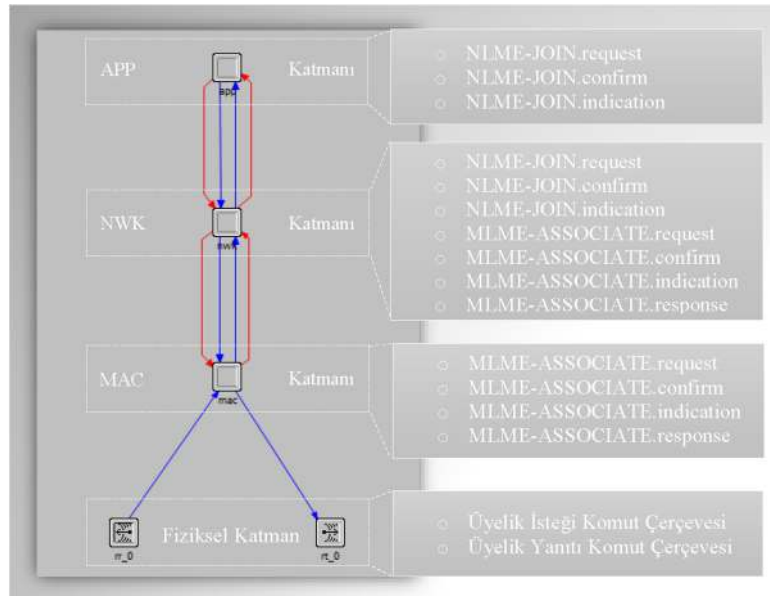
Bu bildiride, ZigBee teknolojisinin temel alındığı ağlarda çalışabilen bir WSN düğümün katmanlı yapısının gerçeğine yakın bir şekilde OPNET ortamında modellenip, simülasyon süreci etraflıca açıklanmıştır. Çalışmada simülasyon için modellenen WSN düğümlerin mevcut bir WSN ağına dışarıdan üye olarak bağlanma isteğinde bulunduğu bir senaryo oluşturularak, WSN düğümlerinin ve bağlanılacak ağıdaki koordinatör düğümünün bu süreçteki davranışları gözlemlenmiştir. Simülasyon sonuçları incelenerek modelin doğru çalıştığı gösterilmiştir. Bu bildiri bir WSN sisteminin katmanlı mimari yapısındaki mevcut veya yeni geliştirilecek protokol yapılarının OPNET platformunda modellenmesi, simüle edilmesi sürecinde yapılması gerekenleri uygulamalı olarak açıklaması bakımından da önemlidir. Bildirinin sonraki bölümlerinde, OPNET Modeller platformunda oluşturulan düğüm modeli, süreç yapıları, çerçeve formatları ve bir WSN ağına katılma sürecinde düğümlerin izleyecekleri davranışların modellenmesi ve OPNET simülasyon ortamında simülasyon ve çıktıların analizi açıklanmıştır. Son bölümde ise bu çalışmanın sonuçları değerlendirilmiştir.

2. Opnet Simülasyon Modeli

OPNET Modeller'de bir WSN düğümünün katmanlı yapısının modellenip simüle edilmesini açıklamak için, bir koordinatör düğümün ağına bağlanmak isteyen uç cihazların (kenar düğümlerin) bu isteklerini kabul ederek ağ adresleri ataması sürecini kapsayan bir senaryo oluşturulmuştur. Düğüm modellemesi IEEE 802.15.4 standardı üzerinde çalışan ZigBee protokolü spesifikasyonlarına uygun olarak gerçekleştirilmiştir.

2.1. Düğüm Modeli

Şekil 1'de bir WSN düğümün, OPNET platformunda ZigBee spesifikasyonuna uygun katmamsal yapısı gösterilmiştir. Bunlar, fiziksel (PHY), MAC, ağ (NWK) ve uygulama (APP) katmanlarıdır. Burada kırmızı ve mavi okların bağlı olduğu kare şeklinde görülen yapılara işlemci (processor) ya



Şekil 1. OPNET Modeler'de oluşturulan katmansal düğüm yapısı

da modül denilirken, oklara da akış (stream) denilmektedir. Akışlar, her bir katman arasındaki iletişimi sağlayan SAP'ları (Service Access Point) birbirine bağlar. ZigBee spesifikasyonunda bu iletişim noktaları, Veri-SAP (DE-SAP) ve Yönetim-SAP (ME-SAP) olarak sınıflandırılmıştır. Şekil 1'deki mavi akışlar, modüller arasındaki Veri-SAP'larını birbirine bağlar ve veri primitifleri bu iletişim noktaları üzerinden alınıp gönderilirken, kırmızı akışlar ise Yönetim-SAP'larını birbirine bağlar ve yönetim primitifleri de bu iletişim noktaları üzerinden alınıp gönderilir. Katmanlarda kullanılan primitiflerden bazıları (MLME-ASSOCIATE.request v.b) yine Şekil 1'de gösterilmiştir. Gösterilen bu primitifler, bir düğümün bir ağa bağlanma isteği ve bu isteğin yerine getirilmesi sürecinde kullanılan primitiflerdir. Fiziksel katmanda kullanılan iletişim çerçevelerinin isimleri de Şekil.1'de görülmektedir.

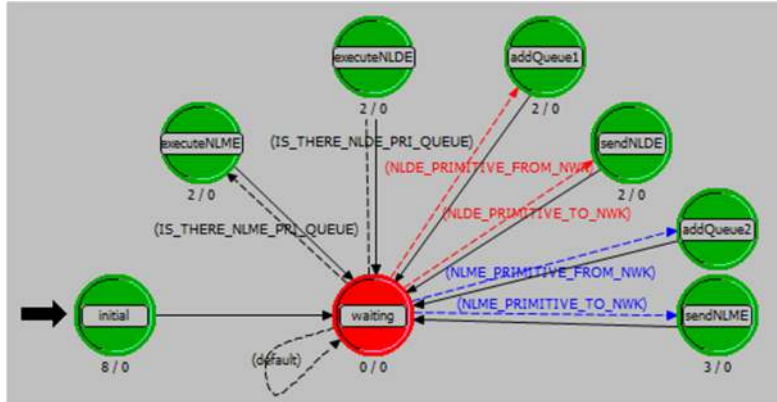
(rt_0) Attributes	
Attribute	Value
- name	rt_0
- channel	(...)
- Number of Rows	1
Row 0	
- data rate (bps)	250,000
- packet formats	gd_ACK_MPDU_frame, gd_COMMAND...
- bandwidth (kHz)	2,000
- min frequency (MHz)	2,405
- spreading code	disabled
- power (W)	0.005
- bit capacity (bits)	infinity
- pk capacity (pks)	1,000
- modulation	qpsk
- rxgroup model	dra_rxgroup
- txdel model	dra_txdel
- closure model	dra_closure
- chanmatch model	dra_chanmatch
- taqain model	dra_taqain
Extended Attrs.	

Şekil 2. Kullanılan bazı PHY parametreleri

Düğüm modelinin fiziksel katmanında yer alan iki tane modül vardır; radyo alıcı modülü (rr_0) ve radyo gönderici modülü (rt_0). Alıcı ve gönderici modülleri kablosuz ortamı kullanarak çerçeve almayı ya da göndermeyi sağlar. Gönderici modülü ilgili akışın bağlı olduğu SAP noktasından herhangi bir paket gelmesi durumunda onu kablosuz ortama modüle edilmiş sinyal olarak aktarır. Alıcı modülü, düğümün 802.15.4 spesifikasyonlarında tanımlı FFD (Full Function Device) veya RFD (Reduced Function Device) olmasına bağlı olarak sürekli veya periyodik dinleme yapmaktadır. Oluşturulan WSN düğümlerinin kullanmış olduğu genel 802.15.4 spesifikasyonunun PHY parametreleri Şekil 2’de gösterilmektedir.

2.2. Süreç (Process) Yapıları

Uygulama, ağ ve MAC katmanı modellemesi için OPNET Modeler’de oluşturulan süreçler, sonlu durum makineleri (FSM) ile tasarlanmaktadır. FSM ‘ler, katmanlar arasındaki primitifleri daha esnek bir şekilde yönetmeye imkan vermektedir. Şekil 3’te gösterilen durum geçiş diyagramındaki kırmızı ve yeşil pullar ilgili katman için belirlenmiş her bir farklı duruma (state) karşılık gelmektedir. Kırmızı durumlar (unforced state), kod yürütümü için OPNET kesme çağrılarını eklerken, yeşil durumlar kod yürütümlerini tamamlar tamamlamaz bir sonraki duruma geçiş sağlamaktadır.



Şekil 3. Uygulama katmanı FSM

Her WSN düğümü bir uygulama katmanına sahiptir. Uygulama katmanı için oluşturulan süreçte, WSN düğümü ilk çalışmaya başladığında başlangıç durumu (initial state) bir *app_initial()* çağrımı ile ilgili düğüme varsayılan parametrelerin yüklenmesini sağlar ve bekleme durumuna (Waiting state) geçer. Bekleme durumu sürecinde, art arda gelebilecek farklı primitifleri işlemek veya işlenmiş primitifleri göndermek için iki kuyruk yapısı (addQueue1 ve addQueue2) oluşturulmuştur. Bekleme durumunda, NWK katmanından bir NLDE (Ağ katmanı veri varlığı) ya da NLME (Ağ katmanı yönetim varlığı) primitifinin gelmesi olayı beklenir. Eğer bu olaylardan biri gerçekleşirse gelen primitif ilgili kuyruğa alınır ve tekrar bekleme durumuna geçilir. Birden çok primitifin bu şekilde uygulama katmanına gelmesi halinde bu primitiflerin hepsi ilgili kuyrukta tutulur. Bekleme durumunda eğer kuyruklarda herhangi bir primitif varsa ilgili kesmenin üretilmesi üzerine işlenmek üzere yürütme durumuna (*execute state*) geçiş yapılır. Kuyruklarda bir primitifin olup olmadığı, geliştirilen kesmeler yardımıyla kontrol edilmektedir. *executeNLME* veya *executeNLDE* durumlarında ilgili primitiflerin görevlerini icra edebilecek *execute*

NLME_on_APP() ve *execute NLDE_on_APP()* gibi geliştirilen çağrılar bulunmaktadır. Bu çağrılar ilgili primitifin türünü belirleyerek kendisine has alt çağrıların başlatılmasında kullanılır. Bu alt çağrılar ise ilgili primitif parametrelerinin alınması ve yine kendisine özel başka bir primitifin oluşturulmasında kullanılır. İlgili yürütme durumundaki fonksiyonların icrası bittikten sonra eğer yeni bir primitif oluşturulursa bu yeni primitif NWK gönderimi için kuyruğuna alınır. Herhangi bir yürütme durumuna geçildiğinde, yapılması gereken işlemlere örnek bir FSM Şekil 3'te verilmiştir. Yine örnek olması bakımından, bu FSM 'deki executeNLME durumuna ait primitif alma, işleme ve yeni primitif oluşturma kod örneği ise Şekil 4'te gösterilmiştir.

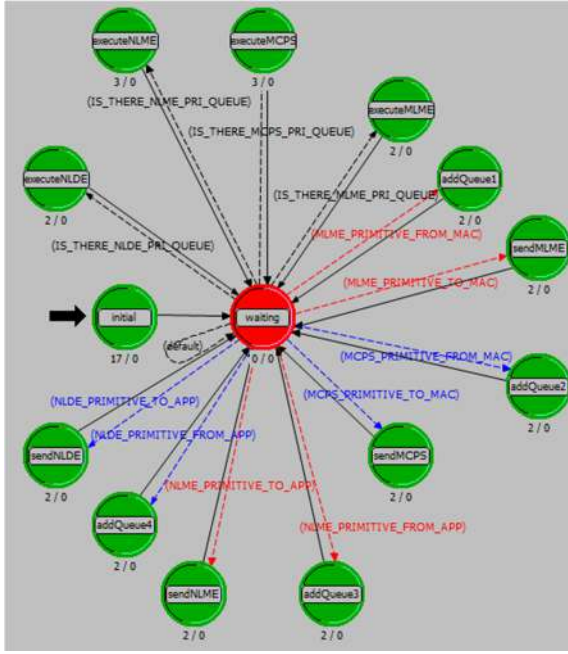
```

130  /* -----
131  * receive_NLME_from_NWK_pk_list kuyrugunda hiç MLME primitive paketi var mı
132  * varsa gerekli primitive fonksiyonunu cagiran fonksiyondur.
133  * -----
134  void execute_NLME_on_APP(){
135      int i=0;
136      int num_pkts;
137      int prim_id;
138      Packet *pkptr = OPC_NIL;
139      num_pkts = op_prg_list_size(receive_NLME_from_NWK_pk_list);
140
141      if(num_pkts > 0){
142          pkptr = op_prg_list_access(receive_NLME_from_NWK_pk_list,OPC_LISTPOS
143          op_pk_nfd_access(pkptr,"Primitive Id",&prim_id);
144
145          switch(prim_id){
146
147              case NLME_JOIN_REQUEST :sprintf(msg,"Primitive ID: %d ",prim_id);
148              case NLME_JOIN_INDICATION :NLME_JOIN_indication_on_APP(pkptr);op
149              case NLME_JOIN_CONFIRM :NLME_JOIN_confirm_on_APP(pkptr);op_prg_l
150
151
152              default : sprintf(msg,"Primitive ID: %d Taninmadi.!!!",prim_id);o
153
154          }
155
156      }
157
158  }
159
160  execute_NLME_pk_timer = op_intrpt_schedule_self(op_sim_time()+NLME_EXE_P
161
162

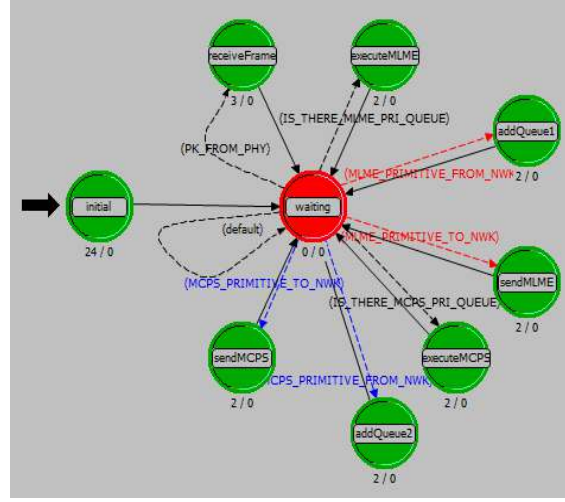
```

Şekil 4. executeNLME durumuna ait örnek bir kodlama

Ağ katmanı ise Şekil 5'te gösterildiği gibi, düğüm çalışmaya başladığında, *initial state*'de varsayılan olarak *nwk_initial()* çağrısını gerçekleştirerek gerekli parametreleri yükler ve bekleme durumuna geçer. Uygulama katmanından NLDE, NLME primitifleri veya MAC katmanından MLME, MCPS primitiflerinden birinin gelmesi olayı burada bir durum (state) değişimine yol açacaktır. Bir olayın gerçekleşmesi halinde alınan primitif ilgili NWK kuyruğuna alınır ve oluşturulan kesme yardımıyla yürütme durumuna (execute state) geçilir. Bu durumda (state) uygulama katmanındaki işlevlere benzer olarak ilgili primitifler, geliştirilmiş olan *execute_NLME_on_NWK()*, *execute_NLDE_on_NWK()*, *execute_MCPS_on_NWK()* ve *execute_MLME_on_NWK()* gibi çağrıların yürütümleri ile icra edilir. Katmanlar arasında primitif gönderimi, ön tanımlı bir kesme yardımıyla gönderme durumunda (*send state*) gerçekleştirilir. Şekil 6'da görüleceği üzere yine APP ve NWK katman işleyişlerine benzer olarak, MAC katmanı, düğüm ilk çalışmaya başladığında başlangıç durumunda *mac_initial()* çağrısıyla gerekli parametre yüklenimi ve dinleme durumu geçişlerini sağlar. Bekleme durumunda ağ katmanından, MLME, MCPS primitifinden birinin gelmesi veya fiziksel katmanından bir paketin gelmesi olayı beklenir. Eğer bu olaylardan biri gerçekleşirse, gelen primitif kuyruğa alınır ve ilgili yürütme durumuna geçilir.



Şekil 5. Ağ katmanı FSM



Şekil 6. MAC katmanı FSM

İlgili primitif işlemleri *execute_MLME_on_MAC()*, *execute_MCPS_on_NWK()* gibi çağrılarla icra edilir. MAC katmanında bir çerçeve *receiveFrame* durumunda oluşturulmaktadır. Oluşturulan MAC çerçevesi geliştirilen ön tanımlı bir PHY kesmesi yardımıyla fiziksel katmana gönderilerek istenilen paket gönderimi gerçekleştirilir.

2.3. Çerçeve Formatları

Bu çalışmada modellenen koordinatör ve uç düğümlerin iletişimi için kullanılan çerçeve formatları IEEE 802.15.4 standartlarına göre OPNET Paket Editörü kullanılarak oluşturulmuştur. Bir uç düğüm bir koordinatörden ağ adresi isterken, Şekil 7'deki üyelik isteği 802.15.4 MAC komut çerçevesini kullanır. Koordinatör ise bu isteğe yanıt vermek için Şekil 8'deki 802.15.4 üyelik yanıt MAC komut çerçevesini kullanır.

Octets: 17/23	1	1
MHR Alanı	Komut Çerçevesi Tanımlayıcı	Yetenek Bilgisi

Şekil 7. Üyelik isteği MAC komut çerçevesinin formatı

Octets: 23	1	2	1
MHR Alanı	Komut Çerçevesinin Tanımlayıcısı	Kısa Adres	Üyelik Durumu

Şekil 8. Üyelik yanıtı MAC komut çerçevesinin formatı

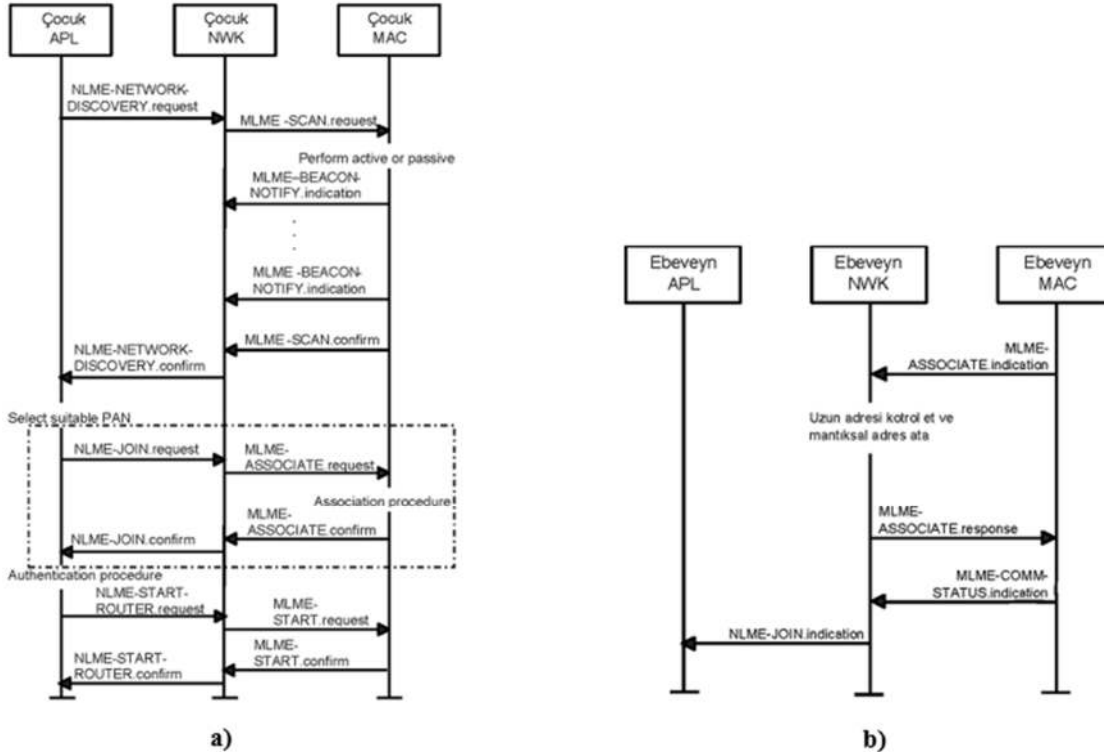
Bu çerçevelerdeki Komut Çerçevesi Tanımlayıcı alanı, hangi tür komutun kullanılacağını belirtir. Örneğin 0x01 üyelik isteği komut çerçevesini tanımlarken 0x02 üyelik yanıtı komut çerçevesini tanımlar. Üyelik isteği komut çerçevesinin Yetenek Bilgisi alanında uç düğümler için “kısa adres ayırma”, “güç kaynağı tipi”, “düğüm boşta iken alıcı devre durumu” gibi bilgiler mevcuttur. Eğer adres ayırma alanı *true* ise Koordinatör düğüm kendisinden bir kısa adres istendiğini anlar ve MAC adresine karşılık ağ adresi tahsis eder. Tahsis ettiği ağ adresini üyelik yanıtı komut çerçevesinin Kısa Adres alanına ekler ve Üyelik Durumu alanına da bu üyeliğin başarılı, başarısız vs. gibi durumları ekler ve gönderir. Bu iki çerçeve türündeki MHR (MAC Header – MAC başlığı) alanı IEEE 802.15.4 standartlarında belirtildiği gibi yapılmıştır.

2.4. Kablosuz Sensör Ağına Katılma Süreci

Bir ağ koordinatör düğümü, ağa ilk defa katılmak isteyen yeni bir cihaza izin verdiği zaman bu iki cihaz arasında ebeveyn-çocuk (parent-child) ilişkisi gerçekleşmektedir. Yeni cihazın ağa katılması üzerine ağda zaten üyeliği bulunan FFD cihaz ebeveyn olurken, yeni düğüm katılım sürecinde çocuk cihaz olur [1]. Katılan uç düğümün FFD olması durumunda, katılım sonrasında ağa yeni düğümlerin katılması için bu uç düğüm ebeveyn düğüm görevi üstlenebilecektir.

2.4.1. Uç Düğüm (Çocuk) Prosedürü

Bir uç düğüm bir koordinatörün ağına katılmak istediğinde, ilk olarak bağlanabileceği mevcut ağların keşfini yapmalıdır.



Şekil 9. Bir ağa katılma süreci; a) Çocuk düğüm için, b) Ebeveyn düğüm için [3].

Bu ağ keşfi süreci Şekil 9.a’da görüldüğü gibi NLME-NETWORK-DISCOVERY.confirm primitifini alınmasıyla tamamlanmış olur. Ağ keşfinin sonunda, PAN tanımlayıcı ve haberleşilecek kanal gibi temel parametreler belirlenmiş olur. Bu çalışmada ağ keşfinin yapılmış olduğu varsayılmıştır. Bir uç düğümün bir koordinatörün ağına katılma süreci Şekil 9.a’da kesikli dikdörtgen içindeki primitiflerle gösterilmiştir. Bu katılma sürecinde ağa katılacak düğüm, uygulama katmanında oluşturduğu *NLME-JOIN.request* primitifini kendi ağ katmanına gönderir. Bu isteği alan ağ katmanı *MLME-ASSOCIATE.request* primitifi ile MAC katmanına bu katılma isteğini iletir. MAC katmanı bu primitife göre üyelik isteği komut çerçevesini oluşturup fiziksel katmana yayınlanmak üzere gönderir. Koordinatör bu çerçeveyi aldığı anda, kendi ağına katılmak isteyen bir uç düğüm olduğunu ve kendisinden bir kısa adres istendiğini anlar, cevabını uç düğüme üyelik yanıtı komut çerçevesiyle gönderir. Uç düğüm bu çerçeveyi aldığı anda, MAC katmanında *MLME-ASSOCIATE.confirm* primitifini oluşturur ve ağ katmanına gönderir. Ağ katmanı kendi bilgi tabanına (*Information Base*) bu primitif ile birlikte gelen kısa adres parametresini ekler ve *NLME-JOIN.confirm* primitifini oluşturarak uygulama katmanına, artık kendisine tahsis edilen bu kısa adresin kullanılacağını bilgisini gönderir.

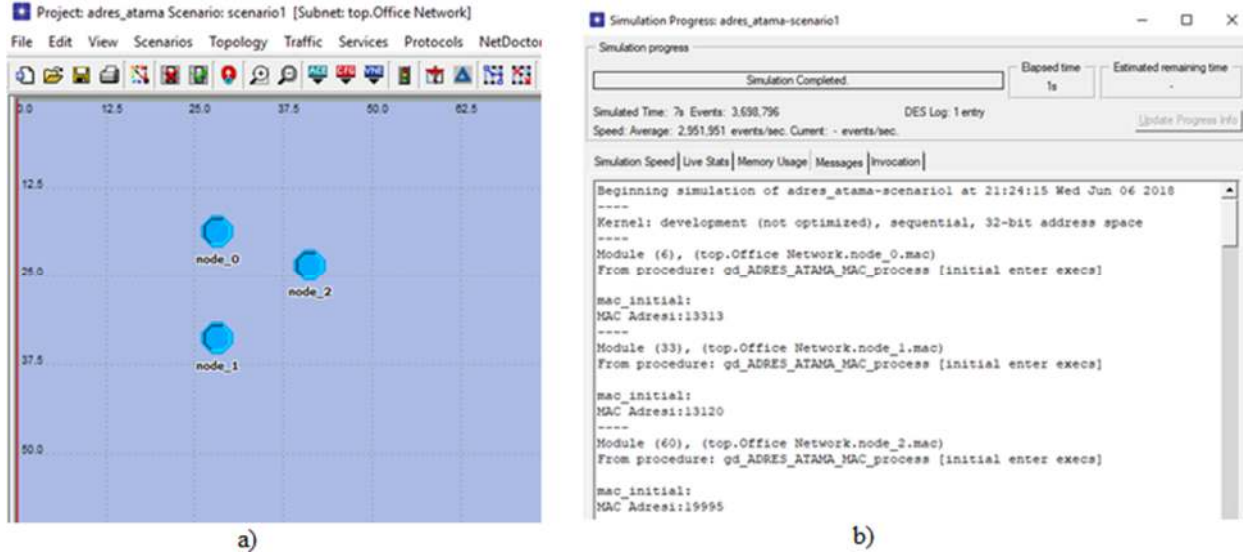
2.4.2. Koordinatör (Ebeveyn) Prosedürü

Bir ağa katılım için öncelikle uç düğüm tarafından üyelik isteği komut çerçevesinin koordinatör düğüme ulaştırılması gereklidir. Bu çerçeve koordinatör düğüme geldiğinde, MAC katmanında *MLME-ASSOCIATE.indication* primitifi oluşturularak ağ katmanına çıkarılır. Ağ katmanına gelen bu primitif ile, uç düğüm için bir kısa adres tahsis edilmesi istenmektedir. Düğüm, kısa adres tahsis sonucunu hem *MLME-ASSOCIATE.response* primitifi ile MAC katmanına hem de *NLME-JOIN.indication* primitifi ile uygulama katmanına gönderir. MAC katmanı *MLME-ASSOCIATE.response* primitifini aldıktan sonra üyelik yanıtı komut çerçevesini oluşturup cevabını uç düğüme gönderir. Ebeveyn düğümdeki bu süreç Şekil.9.b’de gösterilmiştir.

3. Simülasyon Ortamı ve Sonuçları

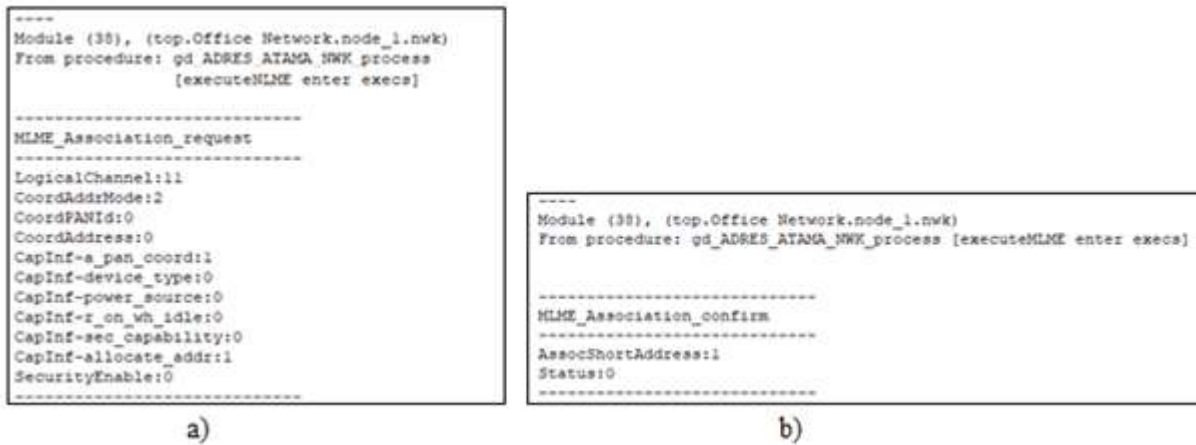
Bu çalışma için oluşturulan simülasyonda iki tip düğüm kullanılmıştır; koordinatör ve uç cihaz. Bu düğümler yukarıda açıklanan modellemelere göre OPNET Modeler platformunda oluşturulmuştur. Koordinatör cihazının kaynak ağ adresi “0” olarak atanmıştır. Tüm cihazlar aynı PAN adresine sahiptirler. Simülasyonda düğümlerin paket transferleri zaman-düzenli çalışma modunda gerçekleştirilmiştir. Zaman-düzenli çalışma modu her düğüm için asal bir sayıyı temsil eder ve düğümlerin aynı anda paket göndermesini engellemektedir. Simülasyon, 100 m2 lik bir alanda tek hop ‘luk iletişime sahip bir WSN ‘i temel almıştır. Şekil 10.a’daki simülasyon görüntüsünde, node_0 koordinatörü, node_1 ve node_2 ise uç cihazları temsil etmektedir. Bu simülasyonda kullanılan tüm cihazlara rastgele olarak MAC adresleri atanmıştır. Şekil 10.b’de simülasyon sonucunda atanan MAC adresleri çıktısı görülmektedir. Burada node_0’ın MAC adresi 13313, node_1’in 13120, node_2’nin 19995 olarak atandığı görülmektedir. PanID’si 0 olan ve 11. kanaldan iletişim yapan bir WSN ağına katılmak için, Node_1 ‘in ağ katmanı tarafından oluşturulan *MLME-ASSOCIATE.request* primitif yapısı Şekil 11.a’da verilmiştir. Bu primitifin ilgili MAC katmanı tarafından alınmasının ardından üyelik isteği komut çerçevesi oluşturulur ve bu çerçeve fiziksel katmana gönderilerek yayınlanır. İlgili ebeveyn düğümün (node_0) ağ katmanı bu

çerçeveyle birlikte gelen ilgili bilgileri alıp değerlendirir ve Node_1'in ağı bağlanma isteğinin uygun bulunup bulunmadığı durumu ve ona atanan kısa adres gibi bilgilerinin belirtildiği MLME-ASSOCIATE.response primitivini oluşturur ve MAC katmanına gönderir. MAC katmanı bu primitive göre üyelik yanıt komut çerçevesini oluşturarak Node_1'e yayınlanmak üzere fiziksel katmanına iletir.



Şekil 10. a) Simülasyon ortamından bir görüntü, b) Simülasyon çıktısı, düğümlere MAC adreslerinin atanması

Node_1'in Mac katmanı ise, gelen bu çerçeveden oluşturduğu *MLME-ASSOCIATE.confirm* primitifi ile ağ katmanına katılım durumunu bildirir. Bu primitifin içeriği Şekil 11.b'de verilmiştir. Şekilden de görüldüğü gibi Node_1 için atanan kısa adres "1" dir. Ayrıca bağlantı isteğine olumlu cevap verildiği de "status = 0" değeriyle bildirilmiştir. Bu değer bu katmanın bilgi tabanında bulunan ilgili değişkene atanır ve bundan sonra bu değer ağ adresi olarak kullanılır. Her düğüme atanan ağ adresi, düğümün üst katmanına da bildirilmektedir. Bu bildirimi yapmak için *NLME-JOIN.confirm* primitifi kullanılmaktadır.



Şekil 11. a) MLME-ASSOCIATE.request primitifi, b) MLME-ASSOCIATE.confirm primitifi

4. Sonular

Kablosuz sensör ağıları belirli bir katmansal protokol yapısı ile alışan kendine ait özel karakter yapısına sahip teknolojilerdir. Enerji verimi ve performans açısından bu teknolojilerin tasarımında modelleme ve simülasyonlar hayati rol oynamaktadır. Katmansal alışma tasarımlarının simülasyon ortamında gereğe yakın yapılması sistem başarımını doğrudan etkileyecektir. Bu alışmada, OPNET Modeler platformunda, 802.15.4 ve ZigBee protokol spesifikasyonlarına uygun olarak adresleme yapabilen bir WSN sistemi için, modelleme ve simülasyon süreçlerinde yapılması gerekenler açıklanıp gerekleştirilmiştir. Bu süreçler için oluşturulan primitif yapıları ve paket biçimleri ilgili spesifikasyonlara göre oluşturulmuştur. Her bir katmandaki gerekli protokol yürütümleri geliştirilen sonlu durum makineleri tarafından gerekleştirilmiştir. Bir koordinatör ağına katılacak N adet düğümün hatasız bir şekilde adreslendirilmesi için bu süreçte gerekleştirilen tüm temel protokol prosedürleri detaylandırılmıştır.

Bu alışmada genel ZigBee protokolü kullanılmış olsa da, WSN sistemler için yeni geliştirilebilecek protokol yapılarının, OPNET platformunda modellenip simüle edilmesi sürecinde yapılması gerekenler için bir yol gösterici niteliğinde olması bu alışmanın dikkate değer önemli bir sonucudur.

Bundan sonraki alışmalarda 802.15.4, ZigBee, LoRaWAN ve diğer WSN protokollerinin tüm süreçlerinin gerekleştirilebildiği, iyileştirmelerin yapılabilmesine imkan verebilen ve OPNET platformunda eksikliği hissedilen simülasyon kütüphanelerinin oluşturulması amaçlanmaktadır.

Referanslar

- [1] ZigBee Alliance, ZigBee Specification, ZigBee Document Number:053474r20, ZigBee Alliance, Inc., 2012.
- [2] IEEE Computer Society, IEEE Std 802.15.4™-2003, The Institute of Electrical and Electronics Engineers, Inc., New York, 2003.
- [3] ZigBee Alliance, ZigBee Specification, ZigBee Document Number: 053474r06, ZigBee Alliance, Inc., 2005.
- [4] S. Biswas, "Simulation model of beacon enabled 802.15.4 networks with OPNET modeler," *2017 International Electrical Engineering Congress (iEECON)*, Pattaya, 2017, pp. 1-4.
- [5] R. Das et al., "Performance analysis of various attacks under AODV in WSN & MANET using OPNET 14.5," *2016 IEEE 7th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, New York, NY, 2016, pp. 1-9.
- [6] M. A. F. Bakheet, "Hierarchical routing protocols of Wireless Sensor Networks: A comparison," *2015 International Conference on Computing, Control, Networking, Electronics and Embedded Systems Engineering (ICCNEEE)*, Khartoum, 2015, pp. 107-113.
- [7] K. Kucuk, N. Bandirmali and A. Kavak, "Modeling of the modified SSLE in OPNET for large scale wireless sensor networks," *The 5th International Conference on Communications, Computers and Applications (MIC-CCA2012)*, Istanbul, 2012, pp. 53-58.
- [8] Deepika and M. Sharma, "Effective data flow in ZigBee network using OPNET," *2014 International Conference on Communication and Signal Processing*, Melmaruvathur, 2014, pp. 1155-1158.