



# Contiki İşletim Sisteminde Cooja Simulatörü Kullanılarak Örnek Bir Nesnelerin İnterneti Uygulaması

İbrahim Karataş<sup>1\*</sup>, Selim Bayraklı<sup>2</sup>

<sup>1</sup> Maltepe Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, İstanbul, Türkiye (ORCID: 0000-0002-5558-3691)

<sup>2</sup> Milli Savunma Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Bölümü, İstanbul, Türkiye (ORCID: 0000-0003-3115-6721)

(İlk Geliş Tarihi 22 Mayıs 2020 ve Kabul Tarihi 26 Ağustos 2020)

(DOI: 10.31590/ejosat.741578)

**ATIF/REFERENCE:** Karataş, İ. & Bayraklı, S. (2020). Contiki İşletim Sistemi ve Cooja Simülasyon Programı. *Avrupa Bilim ve Teknoloji Dergisi*, (19), 763-769.

## Öz

Günümüzde pek çok alanda kullanılan düşük kaynağa sahip sensör düğümler, nesnelerin internetinin çok önemli bir parçasıdır. Özellikle kablosuz sensör ağların sanayi, tıp gibi birçok sektörde kullanılması, akıllı üretim tesislerinin kurulması, hastaların uzaktan izlenmesi gibi birçok önemli gelişmeye neden oldu. Bu cihazları fiziksel ortamda test etmek, büyük ağlar kurmak oldukça zor bir süreç olduğundan Contiki işletim sistemi bize bu donanımı sanal ortamda birebir sunmakta ve Cooja simulatörü ise bu işlemleri gerçek zamanlı bir şekilde test etmemize imkân sağlamaktadır. Bu çalışma kapsamında Contiki işletim sistemi ve Cooja simülasyon programının yapısı anlatılacak ve örnek simülasyonlar gerçekleştirilecektir.

**Anahtar Kelimeler:** Contiki, Cooja, IoT.

## An Application of Internet of Things Using Cooja Simulator in Contiki Operating System

### Abstract

Low-resource sensor nodes, which are used in many areas today, are a very important part of the internet of things. Especially the use of wireless sensor networks in many sectors such as industry, medicine, establishment of smart production facilities, remote monitoring of patients caused many important developments. Testing these devices in a physical environment is a very difficult task, setting up large networks. The Contiki operating system provides this hardware in a virtual environment, and the Cooja simulator allows us to test these processes in real time. In this study, the structure of Contiki operating system and Cooja simulation program will be explained and sample simulations will be carried out.

**Keywords:** Contiki, Cooja, IoT.

## 1. Giriş

Contiki, algılayıcı cihazlar için geliştirilmiş açık kaynak kodlu bir işletim sistemidir (Sehgal, 2013). Düşük güç ve düşük maliyetli mikrodenetleyicilerin (skymote, wismote) internete açılmasını sağlamaktadır. Nesnelerin interneti ise global olarak tanımlanabilen fiziksel objelerin ve bunların internet ile birleşiminin bir aşımıdır. Nesnelerin interneti ağını oluşturabilmek için birçok teknoloji altyapısı kullanılmaktadır (RFID, WAN protokolleri, Web teknolojileri vb.). Bu teknolojiler ise fiziksel bir dünya üzerinde makineden makineye (M2M) ya da makineden insana (H2M) iletişime olanak sağlayan sanal bir dünya oluşturur. Kablosuz sensör ağları (WSN) ise düğümlerin birbirleriyle otonom şekilde iletişim kurduğu, fiziksel ve sanal dünyaya bağlanabildiği akıllı sistemlerdir. Sensör düğümleri limitli hafıza, işlem kapasitesi, batarya ömrü gibi kısıtlara sahiptir (Raza, Slabbert, Voigt, Landernas, 2009). Contiki işletim sisteminin

\* Sorumlu Yazar: Maltepe Üniversitesi, Mühendislik Fakültesi, Bilgisayar Mühendisliği Bölümü, İstanbul, Türkiye, ORCID: 0000-0002-5558-3691, [ikaratas2515@gmail.com](mailto:ikaratas2515@gmail.com)

iletişimi düşük maliyetle gerçekleştirmesi, nesnelerin internetinde bulunan kaynak kısıtlı cihazların test edilmesini kolaylaştırmaktadır. IPv6 ve IPv4 desteğinin yanında nesnelerin interneti için geliştirilen 6LoWPAN, RPL, CoAP protokollerini desteklemektedir (Bagula, Colina, Pietrosemilo, Vives, Zennaro, 2016). Nesnelerin internetinin gelişimiyle beraber bu alanda yapılan çalışmaların artması, test için gerekli olan donanıma ihtiyacı da arttırmıştır. Cooja simülatörü sayesinde de yazılan kod donanıma yazılmadan önce simülasyonu gerçekleştirilebilmektedir. Bu da yapılacak çalışmalar için büyük kolaylık sağlamaktadır. Contiki uygulamaları standart C dilinde yazılmıştır.

## 2. Materyal ve Metot

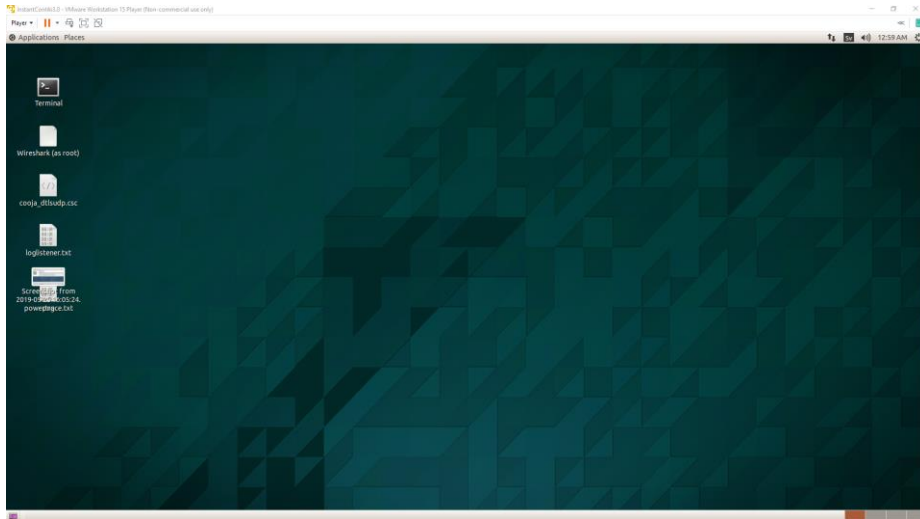
### 2.1. Contiki İşletim Sistemi

#### 2.1.1. Contiki İşletim Sisteminin Yüklenmesi

Contiki ortamında çalışmak için gerekli olan araçlar;

- 1- Contiki kaynak kodu <http://www.contiki-os.org/> adresinden indirilebilir.
- 2- Sanal Makine (VMware) <https://www.vmware.com> adresinden indirilebilir.
- 3- Contiki'nin son geliştirilen sürümü 2015 yılında çıkarılmış olan contiki-3.0 sürümüdür. Bu çalışma kapsamında da bu sürümü kullanılacaktır.

Yükleme işlemi tamamlandıktan sonra sanal makine VMware Workstation programı açılarak "Open a Virtual Machine" seçilir ve indirilen Contiki kaynak kodu gösterilir. Daha sonra "Play Virtual Machine" seçeneğine tıklanarak simülasyon başlatılır. Açılırken bir şifre istenecektir. Standart şifre olarak "user" girilir ve devam edilir. Bu basamaklar tamamlandığında Contiki işletim sistemine giriş yapılmış olur. Ekran görüntüsü Şekil 1'deki gibidir (Duquenooy ve Gonizzi, 2013).

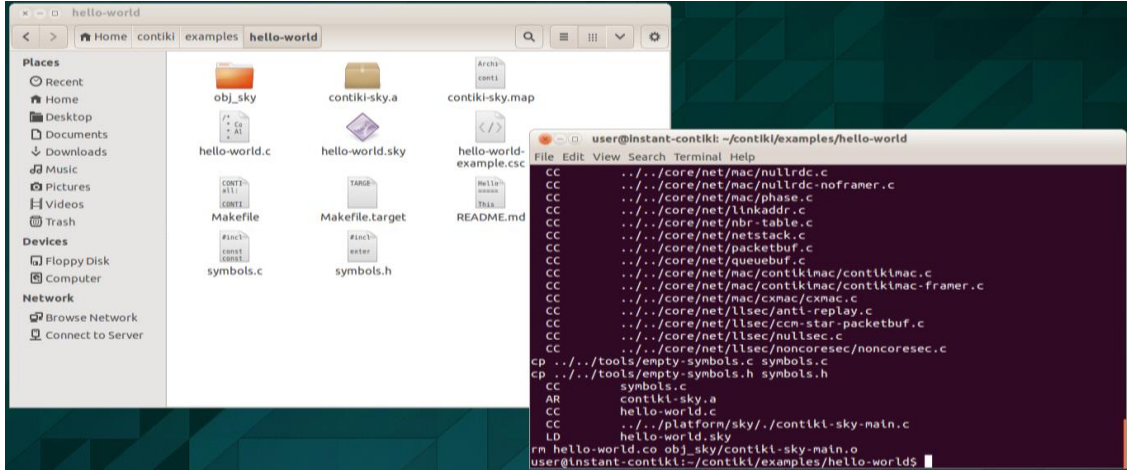


Şekil 1. Contiki işletim sistemi

#### 2.1.2. Terminal Üzerinden "hello World" uygulaması

Masaüstünde bulunan terminal açılır ve aşağıdaki komutları sırasıyla uygulanır.

- cd contiki/examples/hello-world komutuyla hello-world
- make TARGET=sky savetarget komutuyla hedef düğüm olarak sky düğümünü seçilir ve bu hedef kaydedilir.
- make komutuyla kod çalıştırılır ve sonuç Şekil 2'deki gibidir.



Şekil 2. Contiki hello World örneği

-*gedit hello-world.c* komutuyla kod üzerinde istenilen değişiklik gerçekleştirilebilir.

Bu uygulamanın standart kod yapısı Şekil 2'deki gibidir.

```

PROCESS(hello_world_process, "Hello world process");
// gerçekleştirilen işlemin adı
AUTOSTART_PROCESSES(&hello_world_process); //Önyükleme adımı tamamlandığında
kod başlatılacağını belirtir.
    PROCESS_THREAD(hello_world_process, ev, data) //işlemin içeriğinin
    belirtildiği bölüm
    {
        PROCESS_BEGIN(); //işlemin başlagıcı
        printf("Hello, world\n"); //yapılacak işlemler
        PROCESS_END(); //işlemin sonu
    }
    Uygulamaların derlenmesi için ayrıca bir makefile dosyasına ihtiyaç duyulur. Onunda yapısı
    aşağıdaki gibidir.
    CONTIKI_PROJECT = hello-world //derlenecek olan uygulamayı belirtir.
    all: $(CONTIKI_PROJECT) // all kullanıldığında tüm kod derlenecek anlamına gelir.
    CONTIKI = ../../ //contiki ana dosyasının adresi belirtir.
    include $(CONTIKI)/Makefile.include //contiki sisteminin makefile dosyasını
    ekler.

```

## 2.2. Cooja Simülasyon Programı

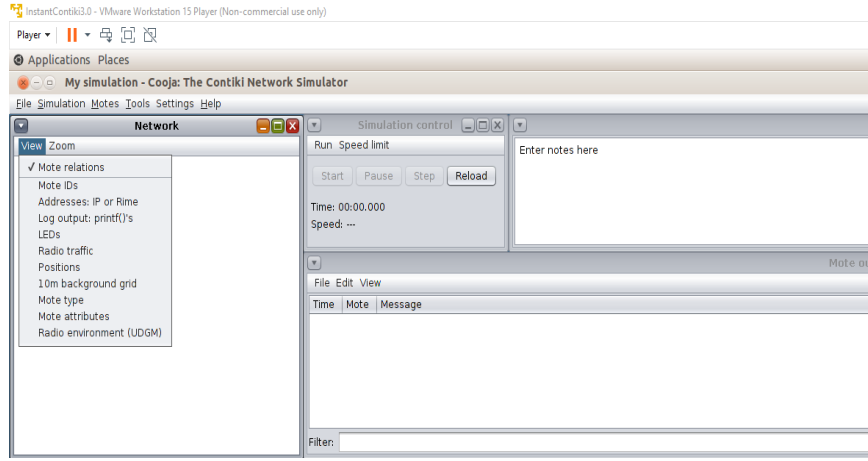
Cooja, sensör düğümlerin bilgisayar ortamında simülasyonun yapılabilirdiği bir ağ simülatördür. Düğüm tarafından yürütülen kod, fiziksel düğümlere yükleyebileceğiniz aynı ürün yazılımıdır (Duquenooy ve Gonizzi, 2013).

Cooja simülasyonunu çalıştırmak için sırasıyla aşağıdaki komutlar terminale yazılır.

-*cd contiki/tools/cooja*

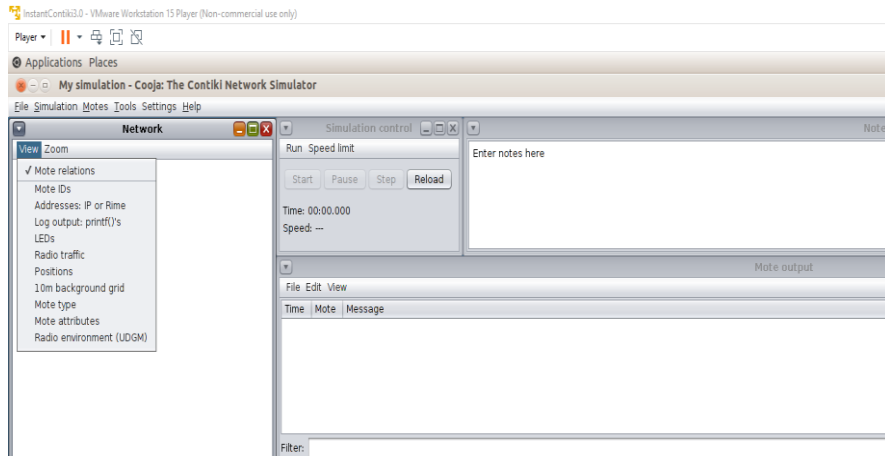
-*ant run*

Cooja yazılımı başlayacak ve karşınıza Şekil 3'teki ekran gelecektir.



Şekil 3. Cooja simülâtör programı

*File > New Simulation* adımları ile yeni bir simülasyon başlatılır. Simülasyonun ismi girildikten sonra *Create* seçeneğine tıklanarak simülasyon başlatılır. İşlem sonucunda Şekil 4’teki ekran ile karşılaşılacaktır.



Şekil 4. Cooja simülâtör ana ekranı

**Network:** Ağın fiziksel düzenini gösterir. View penceresi ile düğüm ve ağ ile ilgili ayrıntıların gösterilmesi sağlanır (düğüm id, düğüm çeşidi, düğüm çıktısı, ağ trafiği vb). Buraya düğüm fiziksel olarak konularak ağın yapısına göre düzenlenebilir.

**Simulation Control:** Simülasyonun başlatılıp durdurulması ve yüklenmesinin gerçekleştirildiği bölümdür. Ayrıca simülasyon hızı da kontrol edilebilir.

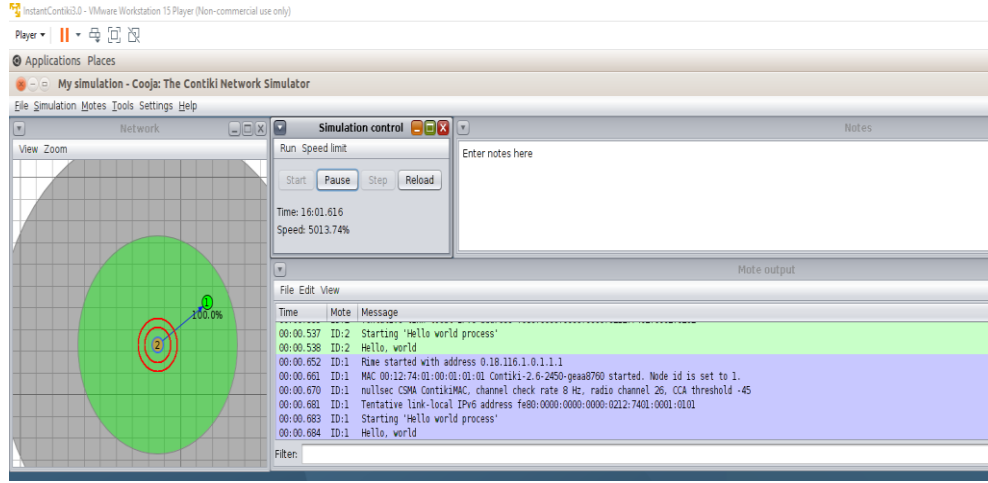
**Mote Output:** Tüm düğümlerden elde edilen ekran çıktılarını gösteren bölümdür. Ancak Filter alanından elde edilen çıktılar ayıklanabilir. Örneğin sadece 1. düğümle ilgili çıktılar isteniyorsa ona göre filtreleme yapılabilir.

**Timeline:** Zaman çizelgelerini, her karede simülasyonun zaman çizelgesi üzerinde gerçekleşen olayları gösterir. Bu çıktılar *file>save to file* ile kaydedilebilir.

**Notes:** Simülasyon ile ilgili notların alınabildiği bölümdür (Al-Dubai, Romdhani, Qasem, Thompson, 2016 ).

### 2.2.1. Cooja “hello world” uygulaması

Örnek bir simülasyon gerçekleştirmek için öncelikle *Motes > Add Motes > Create new mote type > Sky Mote* adımları izlenir. Açılan pencerede *Browse* sekmesinde istenilen kod düğüme yüklenerek derlenebilir. *Contiki / examples / hello-world* klasöründe bulunan *hello-world.c* dosyası düğüme yüklenerek kod derlenmiştir. Burada kodda herhangi bir hata olması durumunda *Create* butonu aktif olmayacaktır. *Create* butonuna tıklanarak düğüm eklenir. Daha sonra *Start* butonuna basıldığında simülasyon Şekil 5’teki gibi başlayacaktır (Al-Dubai ve diğerleri, 2016).

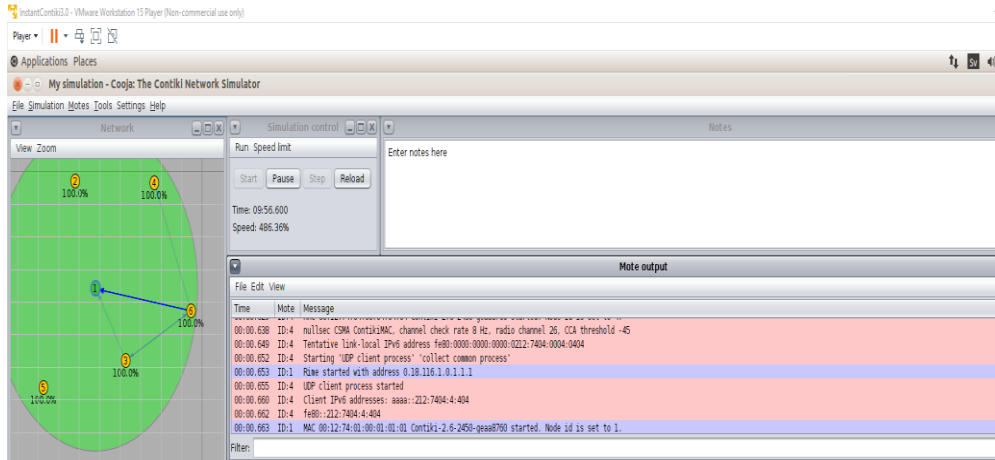


Şekil 5. Hello World Simülasyonu

### 2.2.2. Ipv6-Rpl-Udp Örneği

Nesnelerin internetinin temel protokolleri olan IPv6, RPL ve UDP'yi kullanarak düğümler arası iletişimi gösteren örnek bir simülasyon için aşağıdaki adımlar izlenir;

Motes > Add Motes > Create new mote type > Sky Mote adımları izlenir. Browse seçeneğine tıklanarak *user/contiki/examples/ipv6/rpl-collect* klasöründeki *udp-sink.c* kodu düğüme eklenir. Daha sonra *Clean* seçeneğine tıklanarak önceki derlemelerden kalan kod düğümünden temizlenerek *Compile* seçeneğine tıklanır ve kod düğüme yüklenir. Düğüm sayısı olarak 1 seçilir. Aynı adımlar tekrarlanarak yeni bir düğüm eklenir ve klasör içerisinde bulunan *udp-sender.c* kodunu düğüme yüklenir. Düğüm sayısı olarak 4 seçilir. *Network* alanından düğümlerin topolojisi istenilen şekilde ayarlanabilir. Aynı şekilde *View* sekmesinden de düğüm ve ağ ile ilgili istenilen ayarlar eklenebilir. Oluşturulan ağ yapısı Şekil 6'deki gibidir.

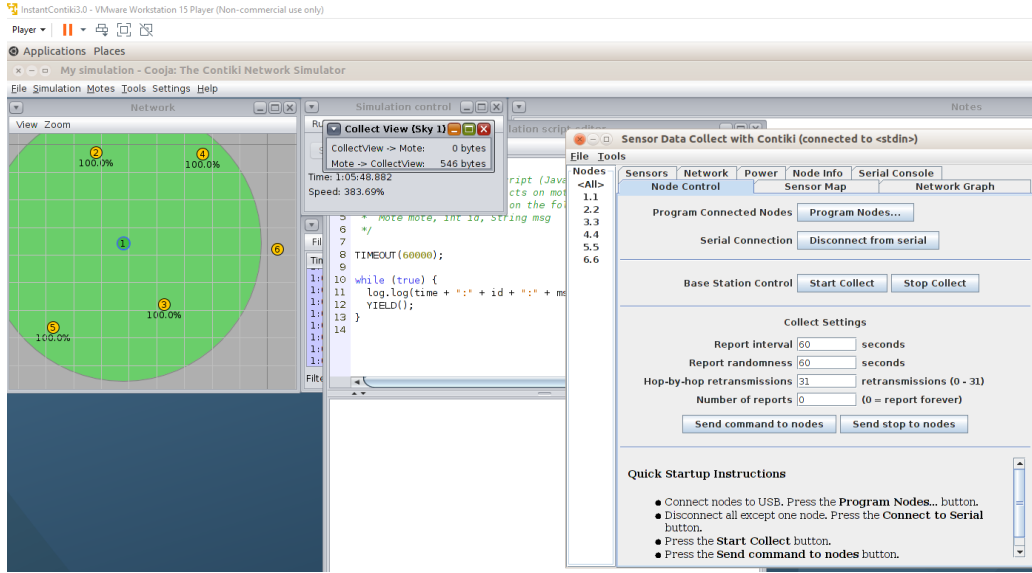


Şekil 6. IPv6-RPL-UDP protokolleri ile düğümler arası iletişim

### 2.2.3. Algılayıcı Veri Toplama

Cooja, düğümlerden veri toplamak için farklı araçlara sahiptir. Yukarıda gerçekleştirilen örneklerde gönderilen verileri, döngü sayıları, düğümlerin tükettiği ortalama güç değerleri mW cinsinden görüntülenebilir.

Bunun için düğüme sağ tıklanarak *Mote Tools for Skyl > Collect View* seçeneğine tıklanır. Açılan pencerede farklı ayrıntılar görüntülenmektedir. Şekil 7'deki görüldüğü gibi *Start Collect* seçeneğine tıklanarak veriler elde edilmeye başlanır. Örneğin, *Power* seçeneğine tıklanıldığında düğümlerin tükettiği ortalama enerji miktarı elde edilebilir, *Sensors* seçeneğinden düğümlerin anlık sıcaklık verisi alınabilir, *Radio Duty Cycle* seçeneğinden ise düğümlerin döngü sayısını elde edilebilir.



Şekil 7. Düğümlerden veri toplama

#### 2.2.4. Sınır Yönlendirici ile CoAP Protokolü Örneği

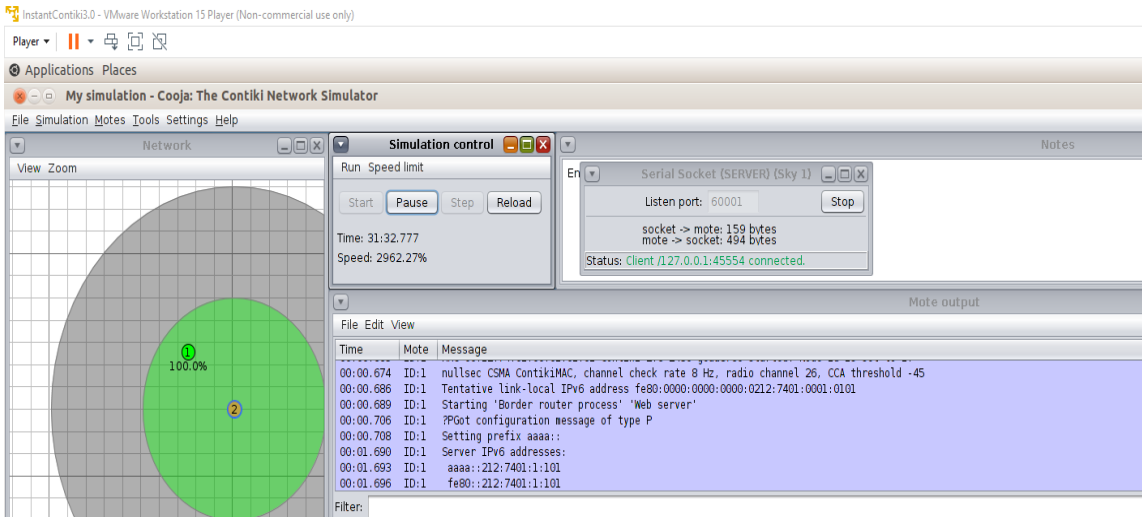
Motes > Add Motes > Create new mote type > Sky Mote adımları izlenir. Browse seçeneğine tıklayarak *contiki / examples / IPv6 / rpl-border-router* klasöründe bulunan *border-router.c* kodunu düğüme yüklenir. Tekrar aynı adımları izleyerek bu sefer düğüme *examples / er-rest-example* klasöründeki *er-example-server.c* kodu yüklenir. Simülasyonu başlatmadan önce sınır yönlendirici ile sunucu arasında bağlantı kurulması gerekmektedir. Bunun için öncelikle sınır yönlendirici (*border-router.c*) kodu eklenen düğüme sağ tıklayarak *Mote Tools for Sky > Serial Socket (server)* seçilip açılan pencereden *Start*'a tıklanır. Daha sonra terminal açılarak sırasıyla aşağıda yazılan komutları girilir.

```
cd contiki/ examples / ipv6/ rpl-border-router
```

```
pwd
```

```
make connect-router-cooja
```

Komutlarını sırasıyla çalıştırdıktan sonra Şekil 8'de görüldüğü gibi bağlantı sağlanmış olur.



Şekil 8. CoAP protokolü aracılığı ile sınır yönlendiri ve sunucu iletişimi

Sonrasında *Application > Internet > Firefox* tıklanarak tarayıcı açılır ve arama kısmına *coap://[ipv6 adresi]* yazılır. Açılan pencereden CoAP protokolü aracılığı ile GET ve POST metodlarını kullanarak düğüme veri gönderip alınabilir. Düğümlerin IPv6 adresi düğüm üzerinden View sekmesi aracılığı ile veya terminalden elde edilebilir.

### 3. Sonular ve Tartışma

Nesnelerin İnternetinin yaygınlaşması ve geleceğın önemli teknolojilerinden biri görölmesi sebebiyle sensör cihazlara olan ilgi ve üzerine yapılan alıřmalar artmıřtır (Li ve Xu, 2017). Cooja Simölasyon programı aracılığı ile sensör düğömler fiziksel olarak olmasa da gerek donanım birebir taklit edilmiř ve fiziksel cihazın ierisine yüklenecek olan yazılım simölasyona yüklenmiř ve düğömlerin alıřması saėlanmıřtır. Bu sayede gerek donanıma ihtiya olmadan yapılan alıřma simölö edilebilmiřtir. Cooja simölasyon programının büyük ve küçük aėların kurulması ve test edilmesinde oldukça kullanıřlı görölmüřtür (Bagula, Colina, Pietrosevoli, Vives, Zennaro, 2016). Nesnelerin İnternetinin yaygınlaşması ve geleceğın önemli teknolojilerinden biri görölmesi sebebiyle sensör cihazlara olan ilgi ve alıřma artmıřtır (Görmüş, Aydın, Ulutař, 2017). Bu sebeple yapılan alıřmalarda geliřtirilen protokollerin test edilmesinde kullanımı oldukça avantajlıdır. Bu alıřma kapsamında Contiki iřletim sistemi ve Cooja simölasyon programı anlatılmıř ancak simölasyonlar tek sensör düğüm üzerinden (SkyMote) gerekleřtirilmiřtir. Gelecek alıřmada Cooja simölasyon programı kullanılarak nesnelerin internetinde güvenlik üzerine geliřtirilen alıřmaların testleri simölö edilecektir. Özellikle var olan řifreleme yöntemlerin performansları karřılařtırılarak gelecek alıřmalar iin öneriler sunulacaktır.

### Kaynaka

- Sehgal A. (2013). Using the Contiki Cooja Simulator. *Computer Science, Jacobs University Bremen Campus Ring 1, 28759*.
- Bagula, A. Colina, A. L. Pietrosevilo, E. Vives, A. Zennaro, M. (2016). Internet of Things in Five Days. pp, 41-87.
- Duquenooy, S. Gonizzi, P. and (2013). Hands on Contiki OS and Cooja Simulator: Exercises (Part II). *Internet of Things and Smart Cities Ph.D. School, University of Parma*.
- Al-Dubai, A. Y. Romdhani, C. Qasem, M. Thompson, I. (2016). Cooja Simulator Manual. *IoT Networking Research Group, Edinburgh Napier University*.
- Li, S. Xu, L. D. (2017). Securing the internet of things. *Syngress*.
- Bagula, A. Colina, A. L. Pietrosevilo, E. Vives, A. Zennaro, M. (2016). Internet of Things in Five Days.
- King J. (2015). A Distributed Security Scheme to Secure Data Communication between Class-0 IoT Devices and the Internet.
- Görmüş, S. Aydın, H. Ulutař, G. (2017). Nesnelerin interneti teknolojisi iin güvenlik: var olan mekanizmalar, protokoller ve yařanılan zorlukların arařtırılması. *Gazi Üniversitesi Mühendislik-Mimarlık Faköltesi Dergisi, 5000211237*.
- Raza, S. Slabbert, A. Voigt, T. Landernas, K. (2009) Security Considerations for the WirelessHART Protocol. In proceedings of 14th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA'09).