International Journal of Computing and Business Research (IJCBR)
ISSN (Online) : 2229-6166
International Manuscript ID : 22296166V7I1201702
Volume 7 Issue 1 May - June 2017

Implementation of Secured IPv6 for 6LoWPAN Based Internet of Things

Dr. Vijayaraghavan. A
Professor & Head
Department of CSE
HMSIT, Tumkur, Karnataka, India

Abstract
In the current era of digital world as well as globalization, the interconnectivity is growing at very swift rate. Now days, we are surrounded with number of gadgets, mobile devices, smartphones, wireless nodes and many other objects which are digitally connected in real time. Internet of Things (IoT) is one of the prominent domains in wireless networking which enable the link between the real world objects. With the implementation of IoT, the physical objects in real world can be connected with each other to share the information and communicate in real time with higher degree of performance as well as security. IoT works on the development and integration of smart objects which can be controlled using remote network infrastructure. This manuscript underlines the security and power aware programming in IoT for higher performance in Cooja.

Keywords: Wireless Networks, Internet of Things, Wireless Security

INTRODUCTION

The term Internet of Things was first presented by Kevin Ashton in year 1999. The implementation of IoT is widespread now because of the availability of high performance wireless technologies. Radio Frequency Identification (RFID) tags and Sensors are base in the implementation of IoT. The RFID tags can be embedded in real world devices and objects which can be monitored remotely using software based applications. The RFID readers can be used to locate, read and sense the RFID implanted objects. Very small micro sized transmitting and receiving chips are integrated with RFID which can communicate at distant point.

As per the reports from Forbes.com, the market of Internet of Things will reach around 267 billion dollars by year 2020. The analysis from Gartner underlines that around 8.4 billion objects with investment of 273 billion dollars will be interconnected with each other in current year 2017. This figure of 8.4 billion objects is 31% more than the implementation figures of previous year 2016.

Some of the key applications of IoT include

- Smart Cities
- Smart Retail Points
- Smart Grid
- Smart Agriculture and Farming
- Internet of Vehicles (IoV)
- Connected Cars
- Connected Railways Infrastructure
- Wearable Devices
- Smart Home
- Smart Offices
- Software Defined Networking
- Smart Supply Chain
- Smart Healthcare and Smart Ambulances
- Industrial Internet
- Energy Management
    - and many others

RESEARCH DOMAINS IN INTERNET OF THINGS (IoT)

As the domain of IoT is in the implementation phase especially in developing countries, there is huge scope of research in assorted dimensions. In

IoT, there are enormous government as well corporate projects in assorted sectors. Due to increasing deployment of IoT, there is the need to analyze various factors which can affect the overall performance and efficiency in the implementation phase.

Following are some of key research issues in IoT

- Security, Privacy and Trust Architectures
- Big Data Analysis and Scalability
- Device Interoperability and Compliance
- Robustness and Fault Tolerance
- Cognitive Networking
- Energy Aware Approaches
- Virtualization
- Ontology Models

FREE AND OPEN SOURCE TOOLS FOR IoT PROGRAMMING

OpenIoT (URL: http://www.openiot.eu/) - It is free and open source platform to manage and program the sensors on cloud and Internet based environment. The concept of Sensing as a Service is finely adopted in OpenIoT.

Zetta (URL: http://www.zettajs.org/) - It is free and open source platform that is having base of Node.js. Zetta is used to create the IoT Servers which can control and run the worldwide distributed systems, sensors and computers including on-cloud.

DSA (URL: http://www.iot-dsa.org/) - Distributed services Architecture is one of the powerful IoT library under free and open source distribution. It makes the inter-objects communication very effective with higher degree of performance. DSA provides the toolkit for managing the IoT based applications, services as well as objects.

Node-RED (URL: http://nodered.org/) - Node-RED provides the programming interface and APIs for the Internet of Things. Using Node-RED, the flow based creation of remote IoT objects can be done with an easy web browser based flow editor. In the flow editor of Node-RED, the JavaScript code can be executed and remote objects can be programmed with easy as well powerful functionalities

IoTivity (URL: https://www.iotivity.org/) - IoTivity a powerful open source library which enable to inter-object connectivity with enormous speed and performance. It is written and programmed in C and C++. Most of the performance aware protocols like ANT+, Bluetooth low energy, Wi-Fi Direct, Zigbee, Z-Wave and others can be easy integrated with IoTivity.

Following is the list of other open source implementations for Internet of Things

Development Toolkits and Libraries

- Arduino
- Eclipse IoT Project
- Kinoma
- M2MLabs Mainspring
- Node-RED
- ThingBox

Automation for Home and Offices

- Eclipse SmartHome
- Home Gateway Initiative (HGI)
- Ninja Blocks
- openHAB
- PrivateEyePi
- RaZberry
- The Thing System

Middleware

- IoTSyS
- Kaa
- OpenIoT
- OpenRemote

Operating Systems

- AllJoyn
- Brillo
- Contiki
- FreeRTOS
- Raspbian
- RIOT
- Spark
- TinyOS

IoT Integration Tools and Horizontal Platforms

- Canopy
- Chimera IoT
- DeviceHive
- IoT Toolkit
- M2MLabs Mainspring
- Mango
- Nimbits
- Open Source Internet of Things (OSIoT)

- OpenRemote
- Pico Labs
- prpl Foundation
- RabbitMQ
- SiteWhere
- SiteWhere
- ThingSpeak
- webinos
- Yaler

Protocols

- Advanced Message Queuing Protocol (AMQP)
- Constrained Application Protocol (CoAP)
- Extensible Messaging and Presence Protocol (XMPP)
- OASIS Message Queuing Telemetry Transport (MQTT)
- Very Simple Control Protocol (VSCP)

Implementations for Engineering

- Open Garden
- Open Source Robotics Foundation
- OpenWSN

USING CONTIKI OPERATING SYSTEM WITH COOJA SIMULATOR FOR PROGRAMMING OF INTERNET OF THINGS

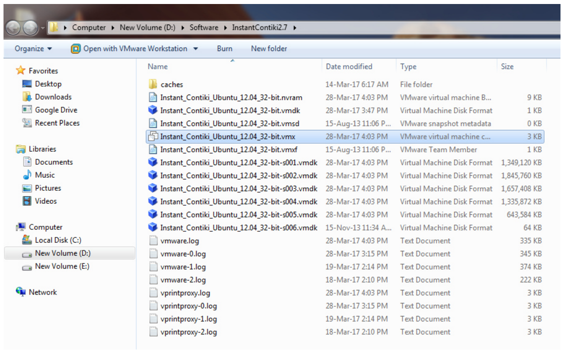
URL: http://www.contiki-os.org/

Contiki is one of the widely used operating system for IoT programming using different types of sensors and RFIDs. It is free and open source operating system under BSD license with the base code of programming language C. Contiki can be used for the communication between low powered RFID chips in wireless networks with higher degree of performance and security.

The programming on Contiki is done using Cooja Network Simulator in which the base libraries of RFID chips and sensors are available in C. To program, control and monitor the remote IoT devices, the back-end C programs and related header files can be customized and recompiled to get the desired results. Contiki works on IPv4 as well as IPv6 networking with the integration of lightweight protocols so that low power chips and radio frequency chips can be connected without performance issues.

URL for Downloading Instant Contiki :
http://sourceforge.net/projects/contiki/files/Instant%20Contiki/

Once the compressed Instant Contiki is downloaded, it can be used on any host operating system. The Instant Contiki is available on sourceforge.net as compressed file which is required to be extracted. The uncompressed or extracted Instant Contiki can be executed on VMWare Player which is a virtualization tool. The VMWare Player can be downloaded free and available on http://www.vmware.com/go/downloadplayer/.
In the extracted folder of Instant Contiki, there is an executable file Instant_Contiki_Ubuntu_12.04_32-bit.vmx.



Name	Date modified	Type	Size
caches	14-Mar-17 6:17 AM	File folder	
Instant_Contiki_Ubuntu_12.04_32-bit.nvram	28-Mar-17 4:03 PM	VMware virtual machine B...	9 KB
Instant_Contiki_Ubuntu_12.04_32-bit.vmdk	28-Mar-17 3:47 PM	Virtual Machine Disk Format	1 KB
Instant_Contiki_Ubuntu_12.04_32-bit.vmsd	15-Aug-13 11:06 P...	VMware snapshot metadata	0 KB
Instant_Contiki_Ubuntu_12.04_32-bit.vmx	28-Mar-17 4:03 PM	VMware virtual machine c...	3 KB
Instant_Contiki_Ubuntu_12.04_32-bit.vmxf	15-Aug-13 11:06 P...	VMware Team Member	1 KB
Instant_Contiki_Ubuntu_12.04_32-bit-s001.vmdk	28-Mar-17 4:03 PM	Virtual Machine Disk Format	1,349,120 KB
Instant_Contiki_Ubuntu_12.04_32-bit-s002.vmdk	28-Mar-17 4:03 PM	Virtual Machine Disk Format	1,845,760 KB
Instant_Contiki_Ubuntu_12.04_32-bit-s003.vmdk	28-Mar-17 4:03 PM	Virtual Machine Disk Format	1,657,408 KB
Instant_Contiki_Ubuntu_12.04_32-bit-s004.vmdk	28-Mar-17 4:03 PM	Virtual Machine Disk Format	1,335,872 KB
Instant_Contiki_Ubuntu_12.04_32-bit-s005.vmdk	28-Mar-17 4:03 PM	Virtual Machine Disk Format	643,584 KB
Instant_Contiki_Ubuntu_12.04_32-bit-s006.vmdk	15-Nov-13 11:34 A...	Virtual Machine Disk Format	64 KB
vmware.log	28-Mar-17 4:03 PM	Text Document	335 KB
vmware-0.log	28-Mar-17 3:15 PM	Text Document	345 KB
vmware-1.log	19-Mar-17 2:14 PM	Text Document	374 KB
vmware-2.log	18-Mar-17 2:10 PM	Text Document	222 KB
vprintproxy.log	28-Mar-17 4:03 PM	Text Document	3 KB
vprintproxy-0.log	28-Mar-17 3:15 PM	Text Document	3 KB
vprintproxy-1.log	19-Mar-17 2:14 PM	Text Document	3 KB
vprintproxy-2.log	18-Mar-17 2:10 PM	Text Document	3 KB

Figure 1: Directory Structure and Files in Instant Contiki

This executable file on executing will automatically open in VMWare Player and we will be ready to work with Contiki in Virtualization Software in parallel with any host operating system. The default password for Contiki operating system is “user”.

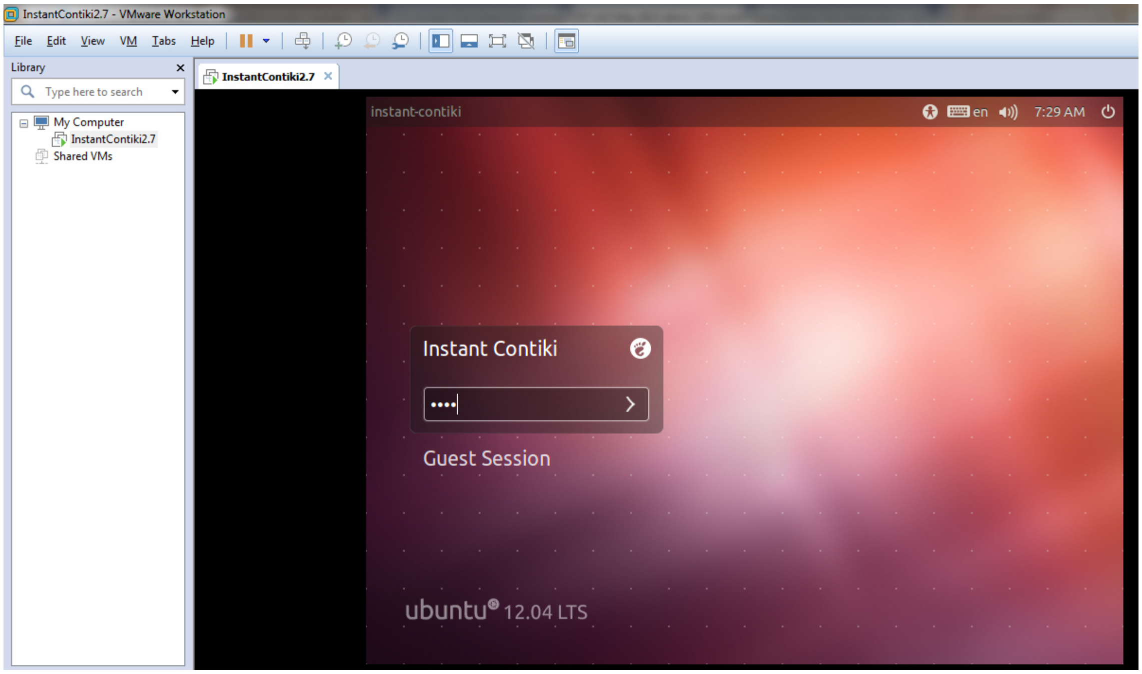
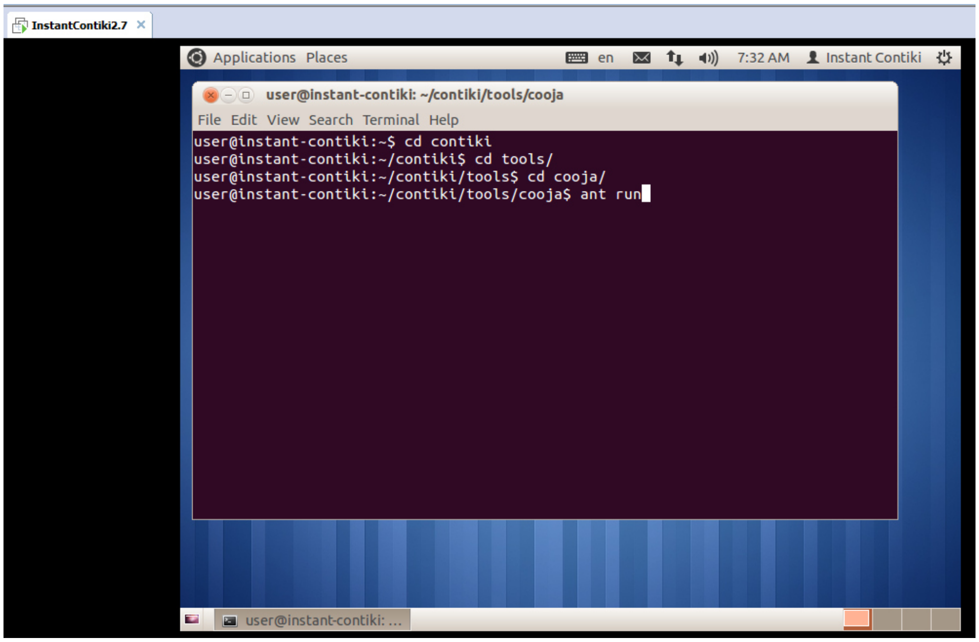


Figure 2: Instant Contiki Login Screen

After loading Contiki O.S., the following commands are executed in the Terminal of Contiki so that the Cooja Simulator gets loaded for implementation of IoT.


InstantContiki2.7
Applications Places
en
7:32 AM
Instant Contiki
user@instant-contiki: ~/contiki/tools/cooja
File Edit View Search Terminal Help
user@instant-contiki: ...

```
user@instant-contiki:~$ cd contiki
user@instant-contiki:~/contiki$ cd tools/
user@instant-contiki:~/contiki/tools$ cd cooja/
user@instant-contiki:~/contiki/tools/cooja$ ant run
```

Figure 3: Loading Cooja Simulator in Contiki

CREATING NEW NETWORK IN COOJA SIMULATOR

In File Menu of Cooja, select New Simulation as follows

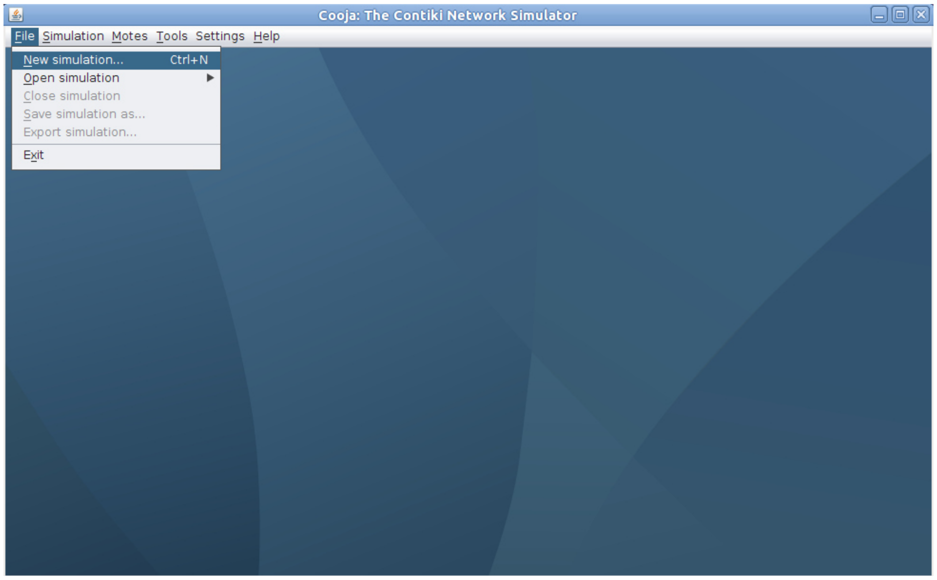
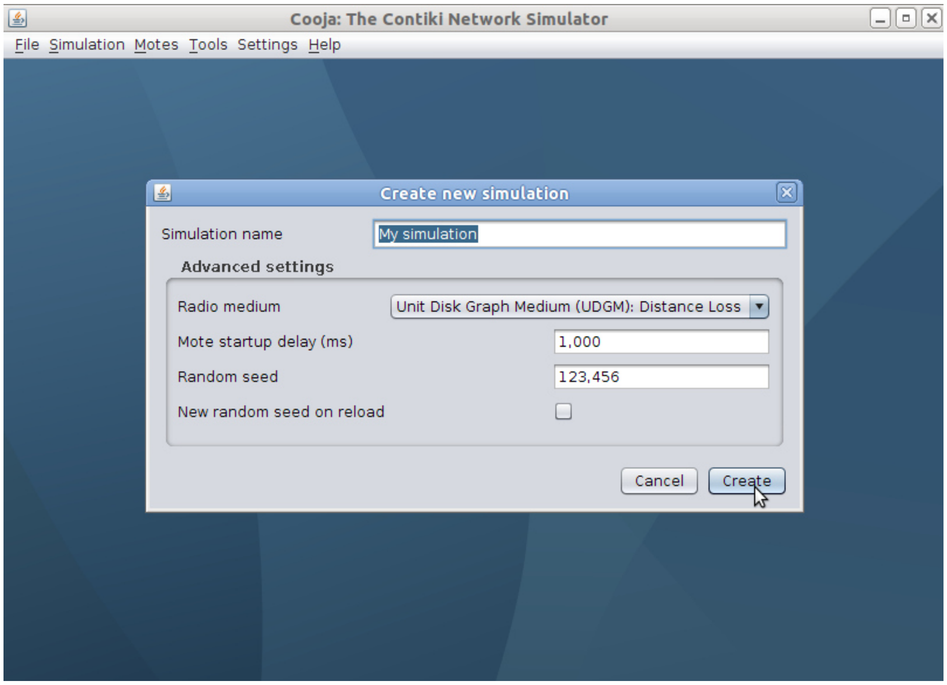


Figure 4: Creating New IoT Simulation in Cooja

In the dialog box, the basic network parameters are set which includes the Name of Simulation, Radio Medium, Startup Delay and Random Seed.


Cooja: The Contiki Network Simulator
File Simulation Motes Tools Settings Help
Create new simulation

Simulation name: My simulation

Advanced settings

Radio medium: Unit Disk Graph Medium (UDGM): Distance Loss

Mote startup delay (ms): 1,000

Random seed: 123,456

New random seed on reload: □

Figure 5: Setup of basic properties of simulation in Cooja

Once the basic layout and working environment is prepared, there is need to import the RFID tags, sensor nodes or any other wireless devices which are required to be connected and communicating in IoT. In wireless networking and IoT, these are known as motes. There are many types of motes in Cooja which can be programmed.

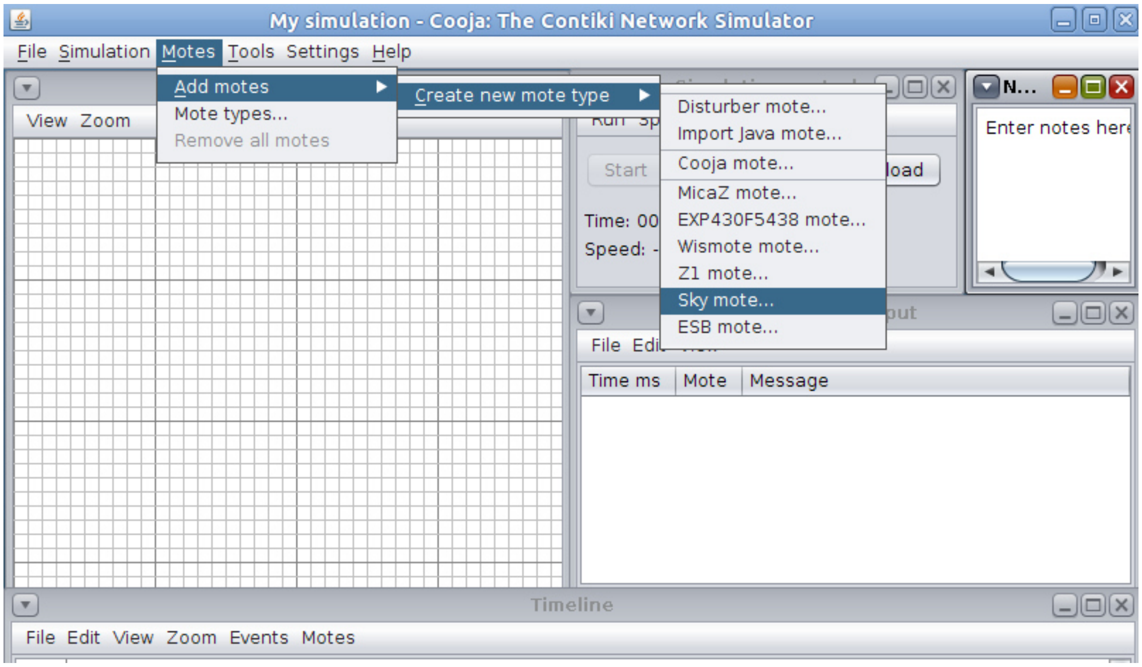
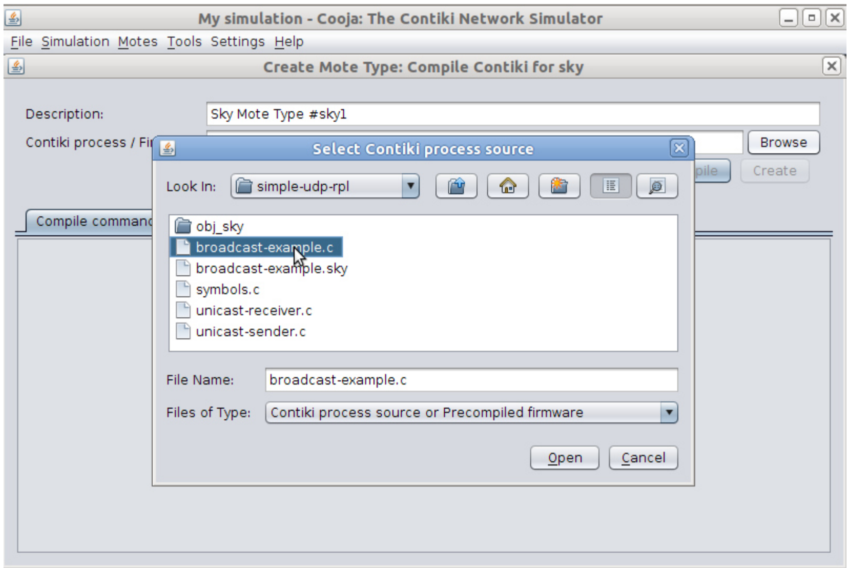


Figure 6: Invoking Wireless and RFID Motes in Cooja Simulator

Even physical motes can be connected using ports on the system so that real time interfacing can be done. Every mote with the base properties and programming APIs are specified in the source code of C at back-end of Cooja. These C source code files can be customized and recompiled to get the new or desired output from these motes.


My simulation - Cooja: The Contiki Network Simulator
File Simulation Motes Tools Settings Help
Create Mote Type: Compile Contiki for sky
Contiki process / Fi
Browse
pile
Create
Compile command
Select Contiki process source
obj_sky
broadcast-example.c
broadcast-example.sky
symbols.c
unicast-receiver.c
unicast-sender.c
Open
Cancel

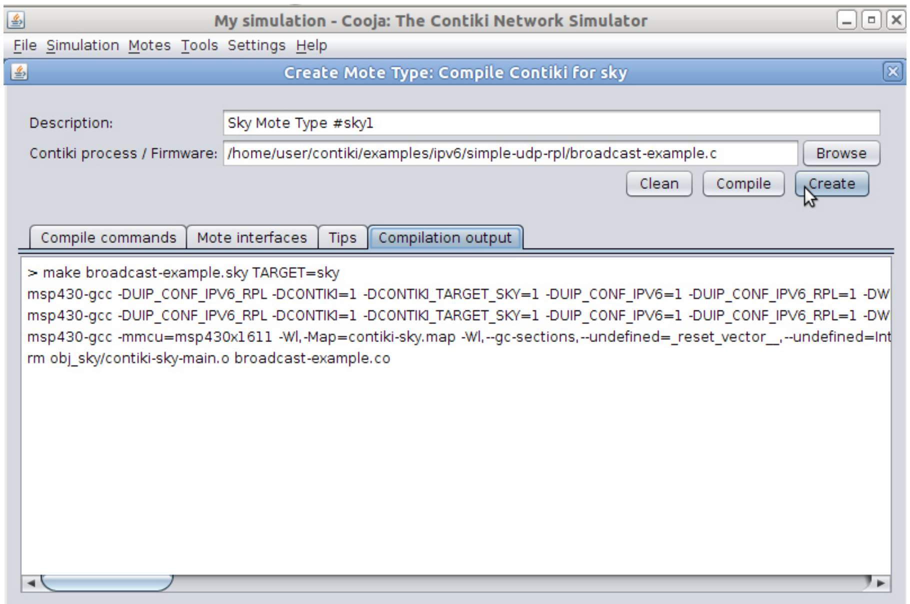
Description: Sky Mote Type #sky1

Look In: simple-udp-rpl

File Name: broadcast-example.c

Files of Type: Contiki process source or Precompiled firmware

Figure 7: Importing C Source Code for Recompilation in Cooja


My simulation - Cooja: The Contiki Network Simulator
File Simulation Motes Tools Settings Help
Create Mote Type: Compile Contiki for sky
Browse
Clean
Compile
Create
Compile commands
Mote interfaces
Tips
Compilation output

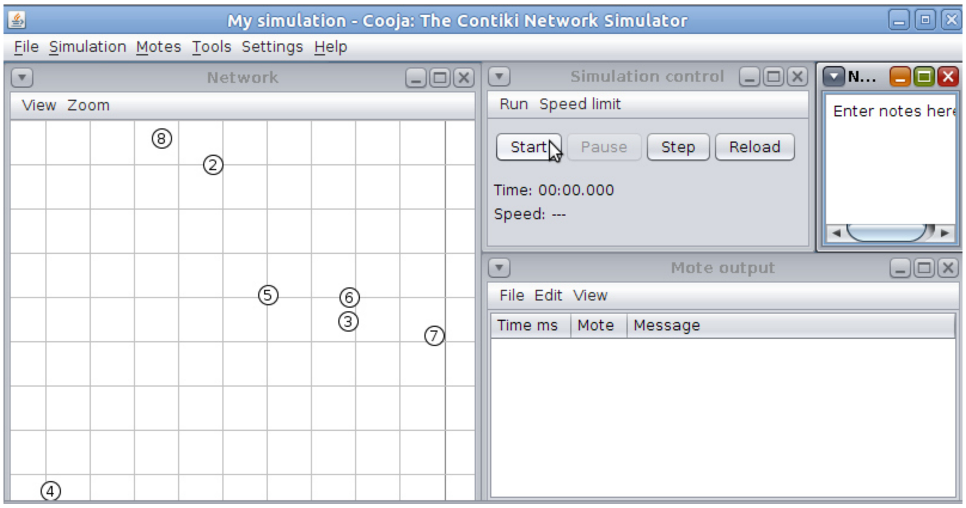
Description: Sky Mote Type #sky1

Contiki process / Firmware: /home/user/contiki/examples/ipv6/simple-udp-rpl/broadcast-example.c

```
> make broadcast-example.sky TARGET=sky
msp430-gcc -DUIP_CONF_IPV6_RPL -DCONTIKI=1 -DCONTIKI_TARGET_SKY=1 -DUIP_CONF_IPV6=1 -DUIP_CONF_IPV6_RPL=1 -DW
msp430-gcc -DUIP_CONF_IPV6_RPL -DCONTIKI=1 -DCONTIKI_TARGET_SKY=1 -DUIP_CONF_IPV6=1 -DUIP_CONF_IPV6_RPL=1 -DW
msp430-gcc -mmcu=msp430x1611 -Wl,-Map=contiki-sky.map -Wl,--gc-sections,--undefined=_reset_vector__,--undefined=Int
rm obj_sky/contiki-sky-main.o broadcast-example.co
```

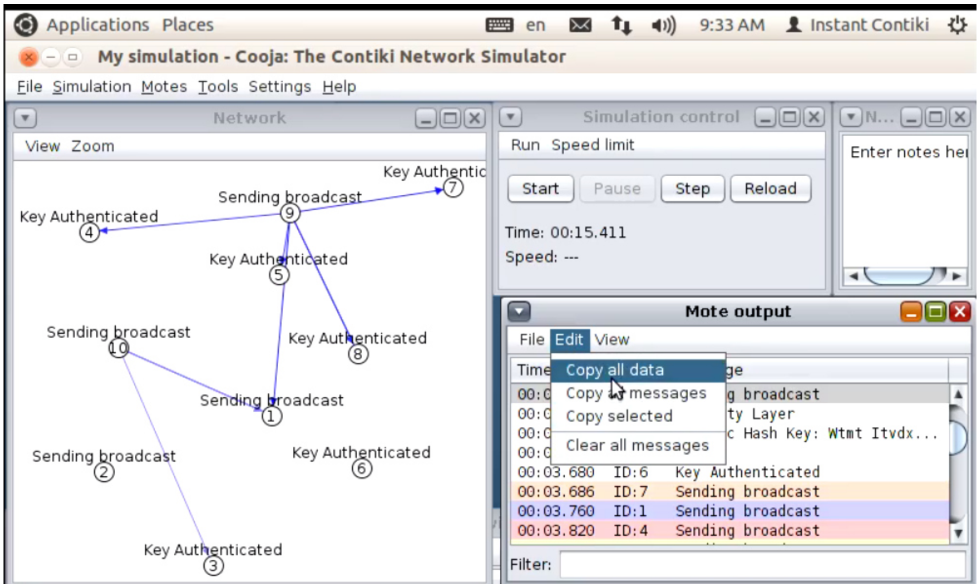
Figure 8: Compiling C Source Code for desired behavior of wireless motes in Cooja

After compilation of C code, the number of virtual motes can be imported in the simulation area so that the transmission of radio signals can be viewed and analyzed.



Time: 00:00.000
Speed: ---

Figure 9: Viewing wireless motes with positions in Cooja



Time: 00:15.411
Speed: ---

```
Time                                  ge
00:0                                 g broadcast
00:0                                 ty Layer
00:0                                 c Hash Key: Wtmt Itvdx...
00:0
00:03.680    ID:6    Key Authenticated
00:03.686    ID:7    Sending broadcast
00:03.760    ID:1    Sending broadcast
00:03.820    ID:4    Sending broadcast
```

Figure 10: Running of Simulation and Behavior Analytics of Motes

In this simulation of IoT network, the scenario of dynamic key exchange between the motes is done in which the dynamic security key is being generated and authenticated for communication. In IoT security, it is necessary to devise and implement the protocols and algorithms by which the overall privacy and security in communication can be enforced to avoid any intrusion. As IoT can be used for military applications, it becomes mandatory to work on highly secured algorithms of key exchange with dynamic cryptography of security keys.

Once the simulation is complete, the network log files are analyzed which includes the source and destination motes, time and overall activities performed during simulation. In the Mote Output Window, the log data can be copied and further analyzed using data mining and machine learning tools for predictive analytics.

Conclusion

Internet of Things is one of emerging domain of research at various academic as well as corporate establishments. Because of increasing number of devices, there are so many segments for research in this area. Following are some of the approaches on which the novel and effectual algorithms can be devised and implemented using Cooja

- Interoperability and Cross-Protocol Compatibility
- Development of Energy Aware IoT Scenarios
- Power Aware Scheduling and Routing
- Prediction and Avoidance of Energy Consumption Attacks
- Lifetime Analytics for Robustness of IoT Environment
- Reproducible and Multi-Interface Implementations
    - and many others

REFERENCES

[1] Chan H., & Perrig A. “Security and privacy in sensor networks, Computer Networks, 36(10), pp. 103-105, Oct. 2003.

[2] Huang Q., Cukier J., Kobayashi H., Liu B., & Zhang J., Fast authenticated key establishment protocols for self-organizing sensor networks, In Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications, pp. 141-150, ACM, Feb. 2003.

[3] Gura N., Patel A., Wander A., Eberle H., & Shantz S., Comparing elliptic curve cryptography and RSA on 8-bit CPUs, In International Workshop on Cryptographic Hardware and Embedded Systems, pp. 119-132, Springer Berlin Heidelberg, 2004.

[4] Watro R., Kong D., Cuti S. F., Gardiner C., Lynn C., & Kruus P., TinyPK: securing sensor networks with public key technology, In proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks, pp. 59-64, ACM, 2004.

[5] Liao W.H., & Huang C. C., SF-MAC: A spatially fair MAC protocol for underwater acoustic sensor networks, IEEE Sensors Journal, 12(6), pp. 1686-1694, Jan. 2012.

[6] Rodríguez-Colina E., Multiple attribute dynamic spectrum decision making for cognitive radio networks, In 2011 Eighth International Conference on Wireless and Optical Communications Networks, pp. 1-5, IEEE July 2011.

[7] Ioannis K., Dimitriou T., & Freiling F. C., Towards intrusion detection in wireless sensor networks, In Proc. of the 13th European Wireless Conference, pp. 1-10, 2007.

[8] Sung W. T., Multi-sensors data fusion system for wireless sensors networks of factory monitoring via BPN technology, Expert Systems with Applications, 37(3), pp. 2124-2131, 2010.