
AIM:

Installing contiki OS (I have done through terminal as I have Ubuntu) and running programs in the COOJA simulator and learning about protocols.

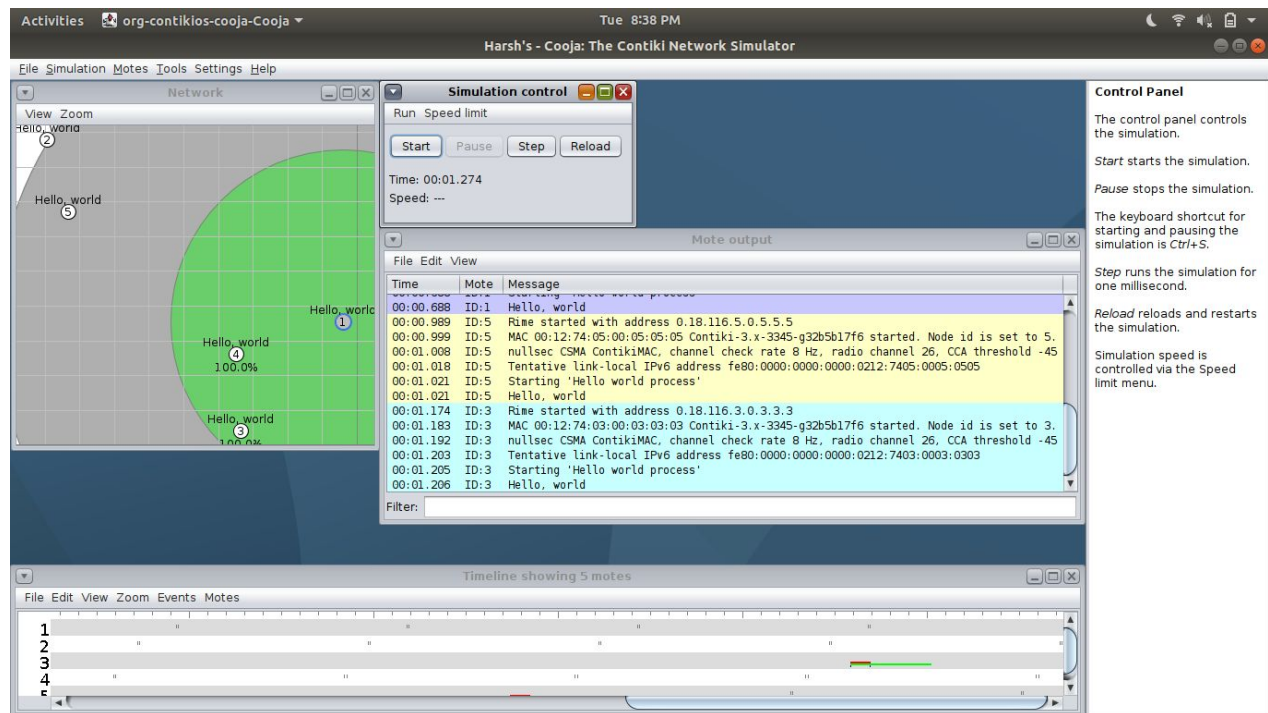
Contiki OS:

Contiki is an operating system for networked, memory-constrained systems with a focus on low-power wireless Internet of Things devices. Extant uses for Contiki include systems for street lighting, sound monitoring for smart cities, radiation monitoring, and alarms.

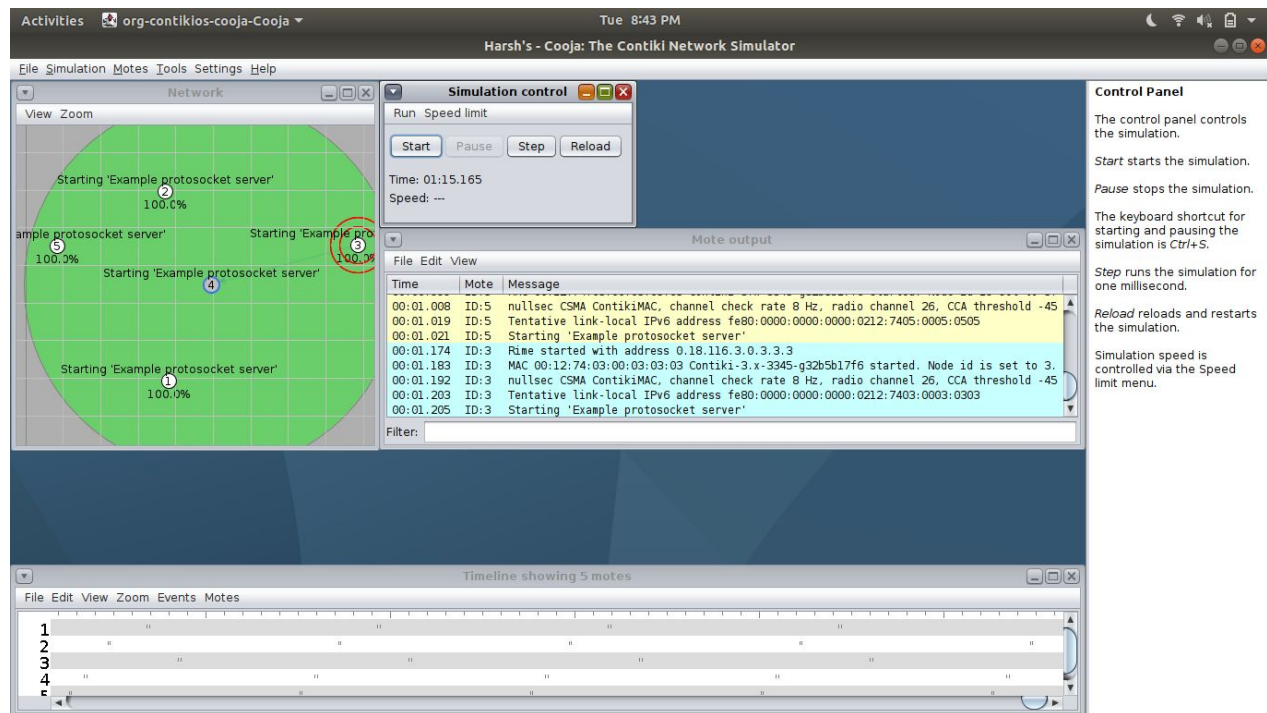
COOJA simulator:

Cooja Simulator is a network simulator specifically designed for Wireless Sensor Networks. Cooja allows the large and small networks of Contiki nodes to be simulated.

Examples:



This is a basic hello world example. Here we have created 5 motes which broadcast a message of hello world to all the other motes.



This is a small example of how to write a TCP server using Contiki's **protosockets**. It is a simple server that accepts one line of text from the TCP connection, and echoes back the first 10 bytes of the string, and then closes the connection. The server only handles one connection at a time. The protosocket library uses **Protothreads** protothreads to provide sequential control flow. This makes the protosockets lightweight in terms of memory, but also means that protosockets inherit the functional limitations of protothreads. Each protosocket lives only within a single function block. Automatic variables (stack variables) are not necessarily retained across a protosocket library function call. Protothreads are a type of lightweight stackless threads designed for severely memory constrained systems such as deeply embedded systems or sensor network nodes. Protothreads provides linear code execution for event-driven systems implemented in C.

Conclusion:

Through this practical we learned a lot about contiki and running programs in the COOJA simulator. We learned about motes and also their implementation in COOJA. We also learned about protosockets library. Protosockets only work with TCP connections.