## TW-9: Understanding the working of Ipv6 in Low power lossy network

Theory for Joournal Writing:

IPv6 in Low-Power and Lossy Networks (LLNs) works to enable communication between devices in constrained environments, like sensor networks or IoT deployments. LLNs are characterized by devices with limited power, processing capabilities, and sometimes unreliable or lossy connections.

IPv6 was designed to accommodate the challenges of LLNs:

1. **Addressing:** IPv6 provides a vast address space, allowing unique addresses for each device. 128-bit IPv6 addresses are used, ensuring a much larger pool of addresses compared to IPv4, essential for the vast number of devices in LLNs.
2. **Neighbor Discovery:** IPv6 in LLNs uses a protocol known as 6LoWPAN (IPv6 over Low-Power Wireless Personal Area Networks) to adapt IPv6 packets for transmission over low-power wireless networks. Neighbor Discovery Protocol (NDP) allows devices to find and maintain relationships with neighboring devices, enabling packet routing and connectivity.
3. **Routing Protocols:** Routing protocols are optimized for LLNs. RPL (IPv6 Routing Protocol for Low-Power and Lossy Networks) is designed for resource-constrained devices, allowing for efficient routing, adaptability to changing network conditions, and minimal energy consumption.
4. **Adaptation and Compression:** 6LoWPAN focuses on compressing IPv6 packets, reducing overhead to fit within the constrained frame sizes of low-power networks. This helps in efficient transmission and reception of data in LLNs.
5. **Security:** IPv6 in LLNs includes security measures like IPSec to ensure secure communication in potentially hostile or untrusted environments. This is crucial in maintaining the integrity and confidentiality of data transmitted across the network.
6. **Efficient Protocols:** The protocols in IPv6 for LLNs are designed to be efficient in terms of power consumption, transmission overhead, and adaptability to the dynamic nature of these networks.

IPv6 in LLNs faces challenges due to the constraints of the network devices, such as limited processing power, memory, and energy. Therefore, optimization and specialized protocols are crucial to ensure effective communication.

#### **Steps to open the cooja simulator**

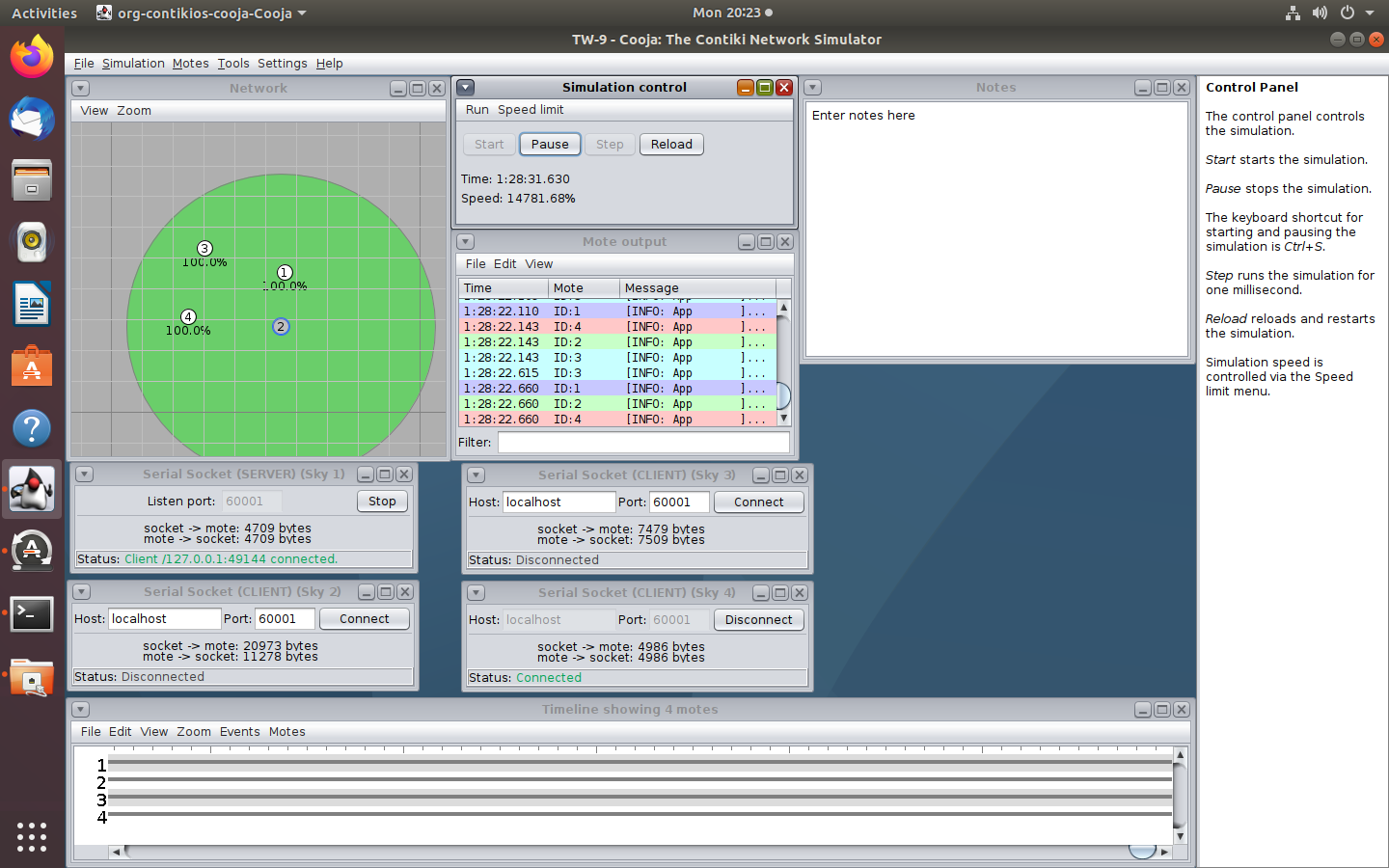
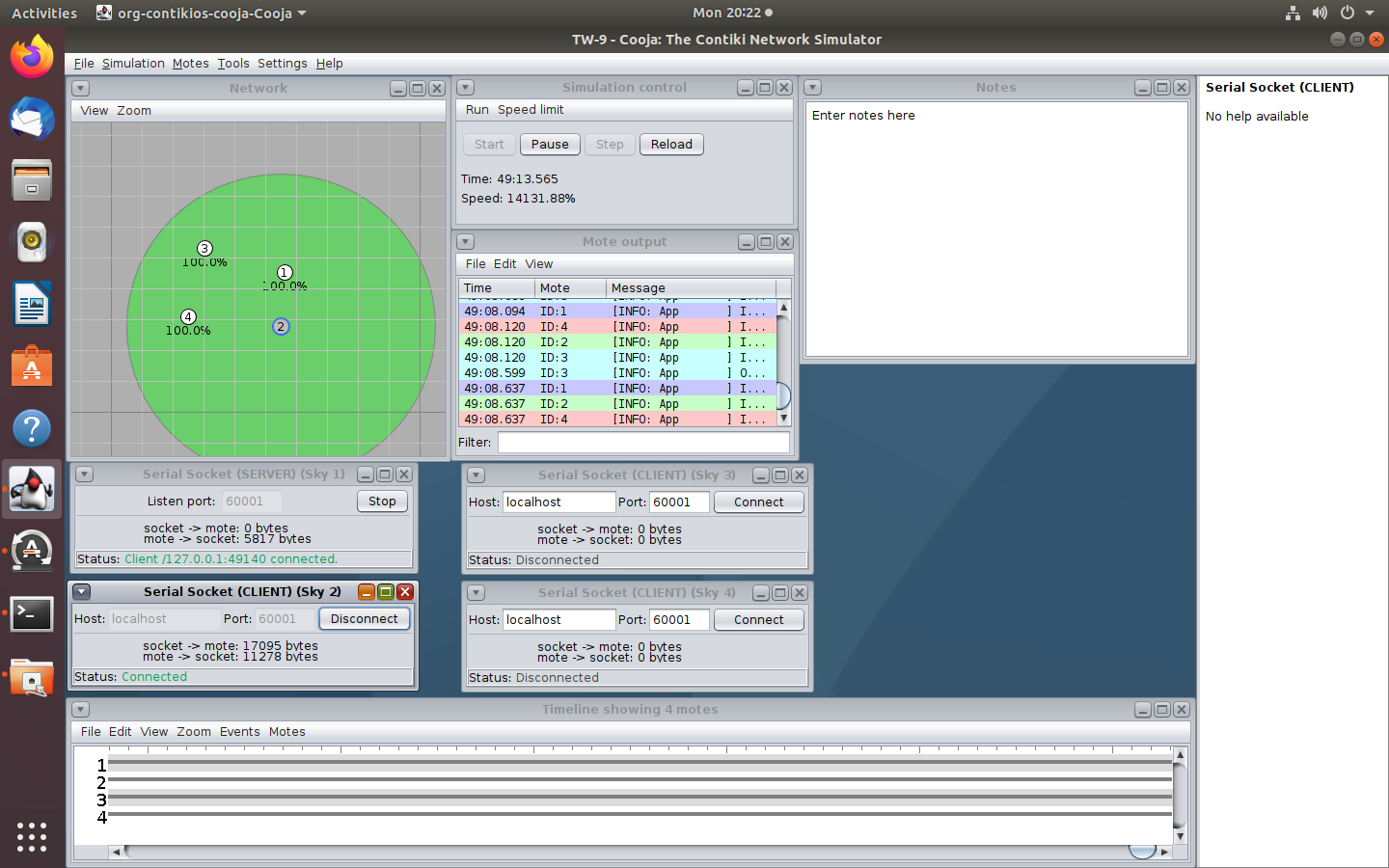
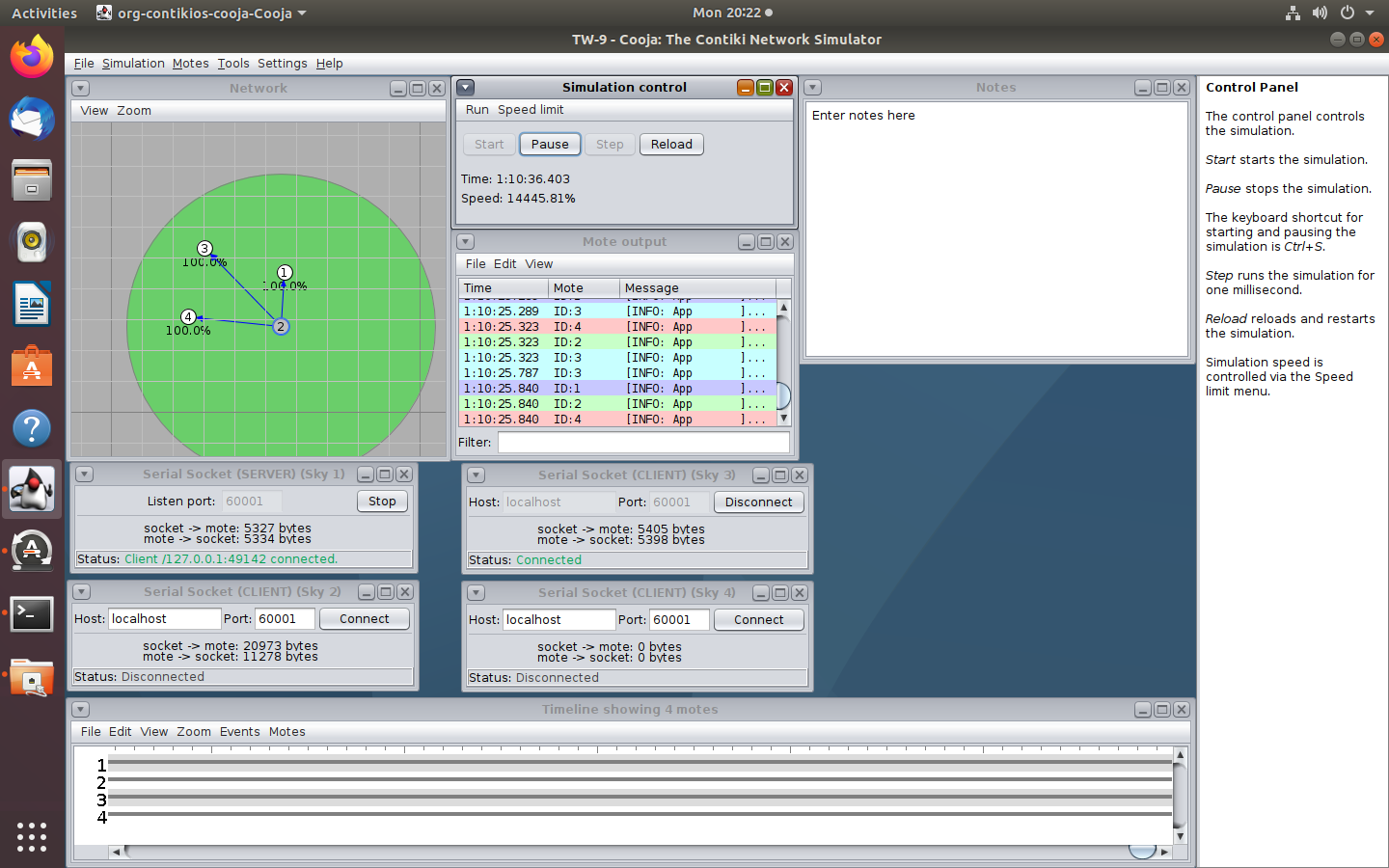
Vmware->Open Virtual machine \_>E drive-> Ubuntu\_Cooja->Ubuntu 1804\_64 bit->Open->Resume

Username-> lab2 Password->git12345

1. Goto root directory
2. cd contiki-ng
3. cd tools
4. cd cooja
5. ant run

#### Steps to create motes and configure them as server and client

1. Goto File -> New Simulation
2. Name the simulation and click on create
3. Click on Motes -> Add motes -> Create a new mote type -> Sky mote
4. Click on Browse and select ipv6-hooks.c (/contiki-ng/examples/libs/ipv6-hooks)
5. Click on open and then on compile and then on create
6. Enter the number of motes as 4 and click on Add motes
7. Place all motes close to each other such that the coverage is 100% for each of them
8. Right click on mote 1 and then click More tools for Sky 1 and then on Serial Socket (SERVER). Mote 1 has been configured as Server.
9. Similarly, configure motes 2, 3 and 4 as clients.
10. Copy the server’s listening port number and paste it as the port number for all clients.
11. Start the server and connect the client to the server.
12. Run the simulation by clicking on Simulation -> Run Simulation



## TW-10: Understanding the working of IoT routing using RPL protocol

Theory for Journal Writing:

The Routing Protocol for Low-Power and Lossy Networks (RPL) is specifically designed for IPv6-based Low-Power and Lossy Networks (LLNs), which are characteristic of the Internet of Things (IoT). RPL is a distance-vector, proactive routing protocol that helps devices efficiently communicate within such constrained networks.

### Working of RPL Protocol:

1. **Objective Function:** RPL uses an objective function to determine the best route based on various metrics, such as energy consumption, latency, and reliability. This function helps nodes select the most suitable path for data transmission.
2. **DODAG Structure:** RPL organizes the network in a Directed Acyclic Graph (DODAG). A DODAG is a tree-like structure that starts from a root node (often a border router) and expands downward to the leaf nodes (end devices). This structure helps in efficient routing and ensures a loop-free topology.
3. **Parent-Child Relationships:** Each node in the DODAG has a parent node except for the root. Nodes maintain parent-child relationships to create the structure. They periodically exchange control messages, allowing them to discover and maintain the optimal paths to the root and to other nodes within the network.
4. **Rank and Trickle Algorithm:** Nodes in the DODAG have a rank, indicating their position within the DODAG hierarchy. The Trickle algorithm helps manage the exchange of control messages, reducing overhead by controlling the frequency of updates based on changes in the network.
5. **Constraints and Objective Metrics:** RPL takes into account the constraints of LLNs, such as limited power, memory, and intermittent connectivity. It optimizes routing decisions based on these constraints and the chosen objective metrics, which could be minimizing energy consumption or reducing latency.
6. **Adaptability and Fault Tolerance:** RPL is adaptive to changes in the network, enabling nodes to reconfigure and find alternative paths in case of failures or changing network conditions. This adaptability ensures fault tolerance and resilience within the network.
7. **Secure Communication:** RPL can be used in conjunction with security protocols like IPSec to ensure secure communication within the IoT network.

### Advantages of RPL in IoT Routing:

* **Energy Efficiency:** RPL is designed to optimize energy usage, critical in IoT networks where many devices are battery-powered.
* **Scalability:** It can scale to accommodate a large number of devices, essential in IoT deployments.
* **Adaptability:** RPL is flexible and can adapt to the dynamic nature of IoT networks, accommodating changes in topology and device status.

RPL protocol is an essential component in enabling efficient and scalable communication within Low-Power and Lossy Networks typically found in IoT deployments.

**Follow the exact same steps to create 2 motes (client and server).**

1. Click on Browse and select rpl-udp(/contiki-ng/examples/libs/rpl-udp)
2. Create udp-server.c and add 1 mote by clicking Motes -> Add new Mote -> Browse
3. Create udp-client.c and add 1 mote
4. Place both the motes close to each other
5. Configure 1 as server and 2 as client
6. Copy the server’s port number to the client.
7. Start the server and connect the client.
8. Run the simulation.

