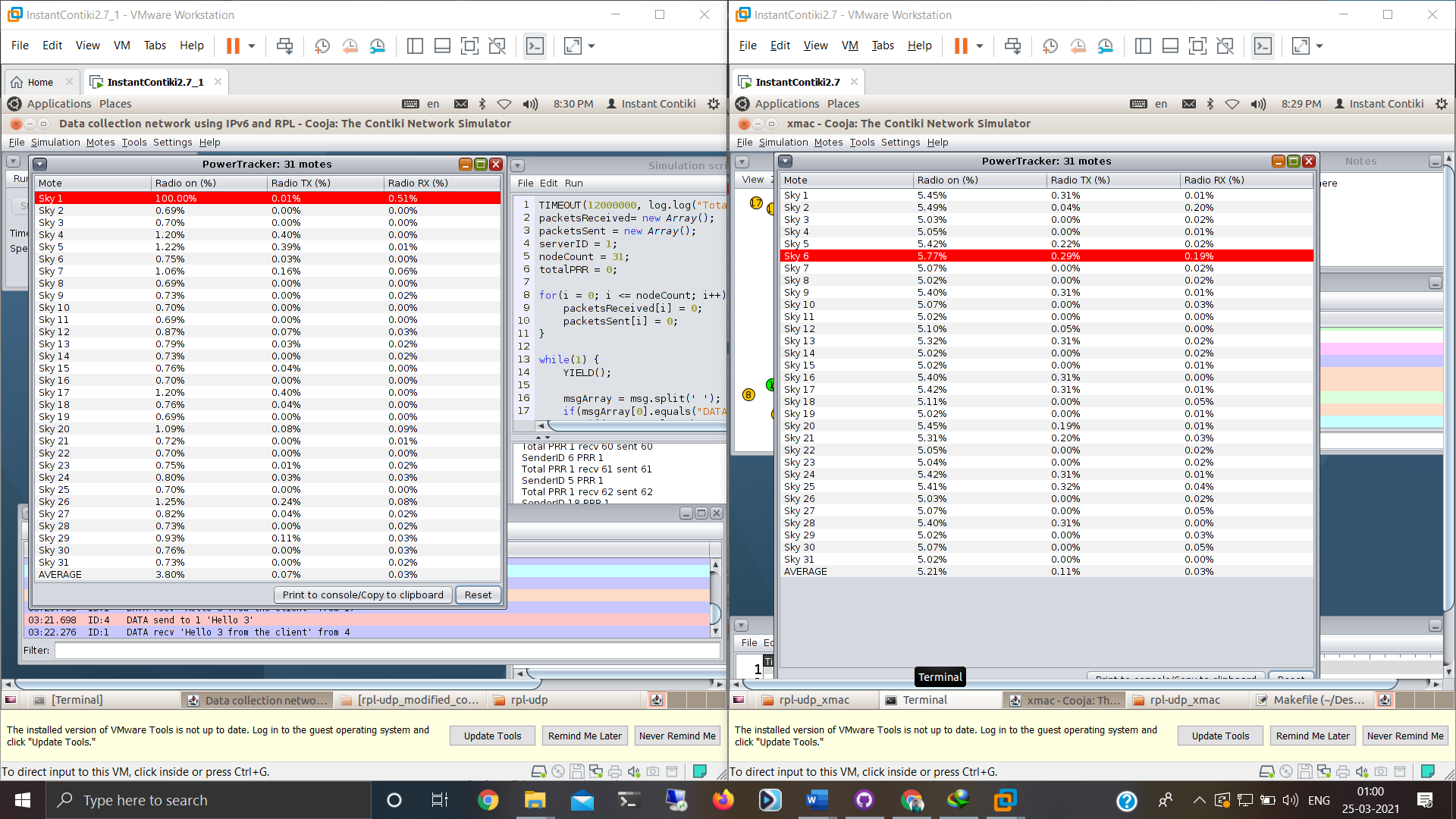
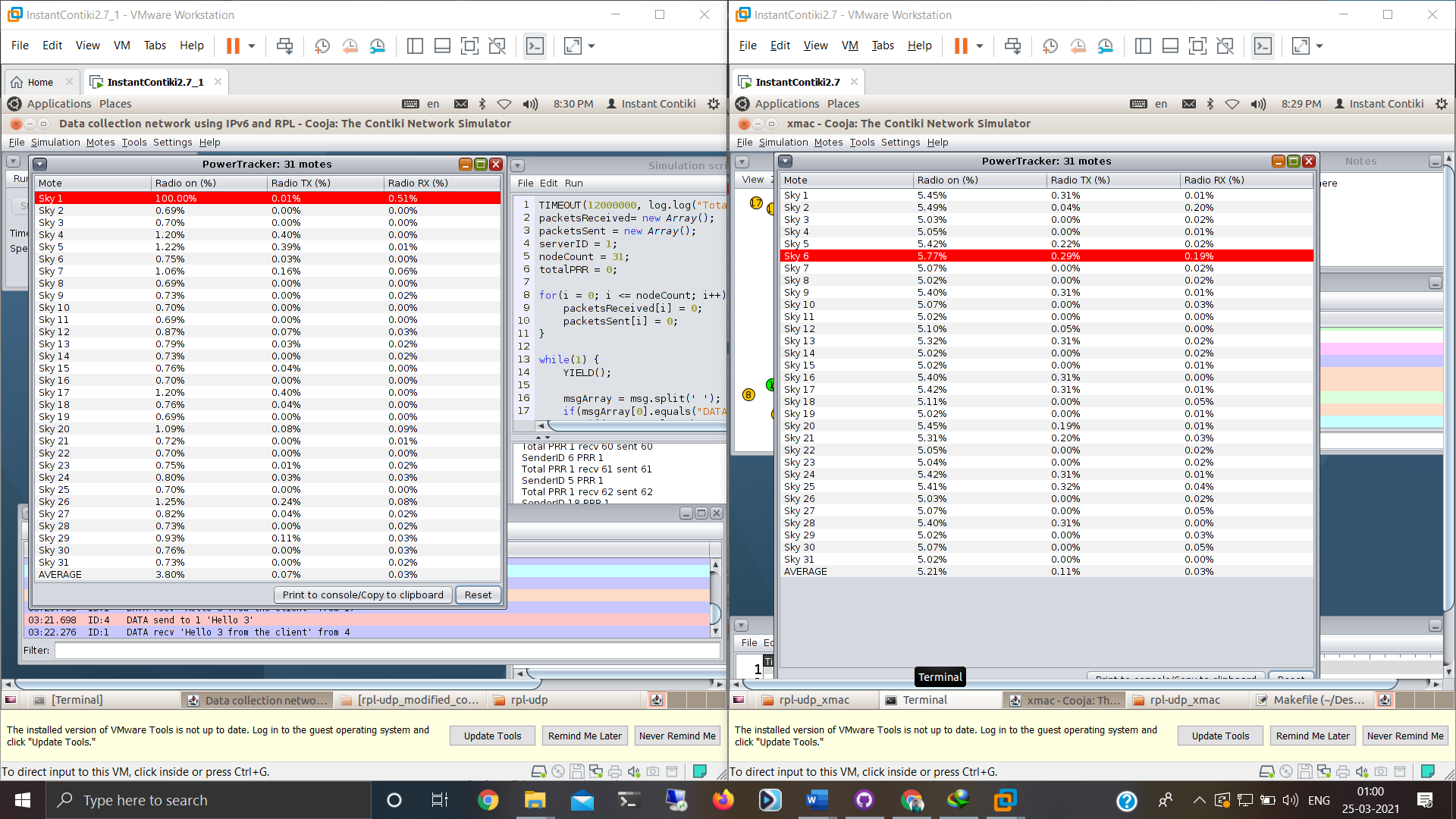
***WSN LAB ASSIGNMENT (C2) -1***

***implementation of X-Mac and Contiki Mac in Contiki OS***

***Chinmay Shravanbal Tayade (IIT2018138)***

****

****

*evaluated the performance of the ContikiMAC and X-MAC. evaluated the Packet Reception Rate (PRR),*

*I considered the firmware in the folder examples/ipv6/rpl-udp/, which is a firmware based on IPv6 and RPL for UDP packet transmission. In the same folder there is one simulation file called rpl-udp-powertrace.csc that is composed of 1 sink node and 30 sensor nodes and traces the energy consumption of all nodes, and also keeps track of PRR values throughout the simulation. We considered the firmwares udp-server.c and udp-client.c and reduced the data transmission interval in udl-client.c to 1 second.*

*By default the simulation is executed with ContikiMAC. We let the simulation run for 5 minutes before extracting the desired statistics. After simulation for ContikiMAC, we changed the RDC protocol, as described before, to X-MAC. And the same statistics were obtained. The figures below show the average duty-cycle of each node for both protocols. The above figure on the left shows results of ContikiMAC and the one on the right shows results of X-MAC.*

*We can see that ContikiMAC reduces the average duty-cycle from 5.2% to 3.8%, which drastically impacts the energy consumption of the nodes.*

*The PRR statistics were also analysed for all 30 sensor nodes. We can conclude that ContikiMAC implementation is more energy and throughput efficient than X-MAC.*

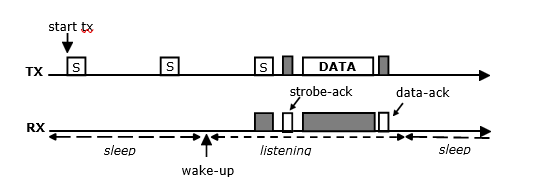
***A Detailed Comparison Between X-Mac And Contiki Mac***

*Contiki MAC is a radio protocol that allows nodes to communicate while keeping their radio turned off most of the time. The radio duty-cycle of Contiki MAC can be as low as 1%. To signal an incoming frame to a receiver, Contiki MAC repeatedly sends the full data frame until it is acknowledged by the receiver. This approach might seem less efficient than what older radio duty cycle protocols have been using. X-MAC for example first sends short strobe frames to signal the receiver of an incoming frame. It is only when those strobes are acknowledged that the actual data frame is transmitted.*

*X-MAC and Contiki MAC are two asynchronous radio duty-cycling protocols. No com- mon wake-up schedule is established a priori among the nodes but instead, each time a frame must be transmitted, the sender needs to synchronize with the receiver.*

*Packet transmission –*

*X-MAC uses different mechanisms to unicast or broadcast a frame In this approach, receivers are notified of an in- coming frame by the transmission of a long preamble, the duration of which is greater than the wake-up interval. X-MAC replaces the use of such a long preamble by a stream of short strobe frames. When the intended receiver catches one strobe it replies with a strobe-ACK. The sender then stops sending strobes and proceeds with the data frame transmission.*

**

*A transmission will fail if any of the following conditions is met : (a) No strobe- ACK is received after a duration equivalent to a wake-up interval, or (b) No data-ACK is received after the data frame transmission. In either cases, it is the responsibility of the above layers to schedule a retransmission.*

*Compared to B-MAC and LPL, this methodology reduces overhearing and latency, resulting in lower energy consumption.*

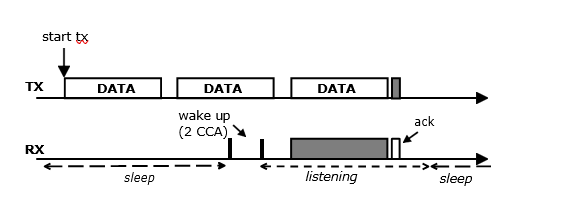
* 1. ***Overhearing reduction*** *is achieved by embedding the destination address in the strobes. Receivers can check this address to determine if they need to stay awake or if they can return to sleep immediately.*
  2. ***Latency reduction*** *is achieved by inserting a short delay between consecutive strobes to allow the destination neighbour to send a strobe-ACK. When a strobe- ACK is received, the stream of strobes is interrupted and the data packet is sent immediately.*

*The broadcast transmission mechanism of X-MAC also relies on a stream of strobes. The difference with unicast is that strobes contain no destination address : every nearby receiver is concerned. Moreover, to allow all the receivers to wake-up, the strobe stream has a duration slightly longer than a wake-up interval. No strobe-ACK is sent by re- ceivers to acknowledge the reception of strobes.*

***ContikiMAC*** *To send a packet, a node running ContikiMAC repeatedly sends the full data frame. The frame destination field allows to reduce overhearing : a node that is not the destination of the frame can immediately go back to sleep. In the opposite case the receiver acknowledges the correct reception of the frame. When an ACK is received, the sender stops sending the data frame and the transmission is successful.*

*A transmission will fail if no ACK is received after a duration equal to a wake-up interval. In this case, it is the responsibility of the above layers to schedule a retrans- mission.*

*Broadcast transmissions are achieved in the same way than unicast, except that no data-ACKs are expected.*

**

***Wake-up procedure –***

*The role of the wake-up mechanism is to determine when a node needs to wake-up and when it can go to sleep. The simplest approach is to wake-up periodically for a fixed duration and then sleep for another fixed duration. However X-MAC and ContikiMAC can decide to go to sleep earlier and hence save additional energy. They do this in conjunction with their transmission mechanisms.*

***X-MAC*** *X-MAC forces each node to wake up at regular interval for a short active*

*period. During its active period, a node listens for potential incoming packets. If no*

*incoming transmissions are detected the node goes back to sleep at the end of its active period and stays asleep until the next scheduled wake-up. The length of an active pe- riod is typically 5 or 10% of the wake-up interval. A typical wake-up interval value is 125 ms. The active period can be extended beyond the cycle length if the node is still involved in transmissions at the end of the cycle.*

***ContikiMAC*** *A node running ContikiMAC also needs to wake up periodically. How- ever, to the contrary of X-MAC, a node does not need to stay awake until the end of its active period. The node quickly performs two successive Clear Channel Assessments (CCA) to determine, based on the radio signal strength, if there is an incoming trans- mission. If the CCAs succeed (channel clear), the node goes back to sleep immediately. If the CCAs fail (channel busy), the node stays awake and a procedure called a fast sleep optimization tries to determine if the received signal is due to noise or to an incoming frame. In case of noise the node goes back to sleep.*

*The timing of the wake-up phase is critical to make data transmission functional. Compared to an X-MAC node which stays awake for a specific time, the use of CCAs by ContikiMAC implies very strict timing constraints.*