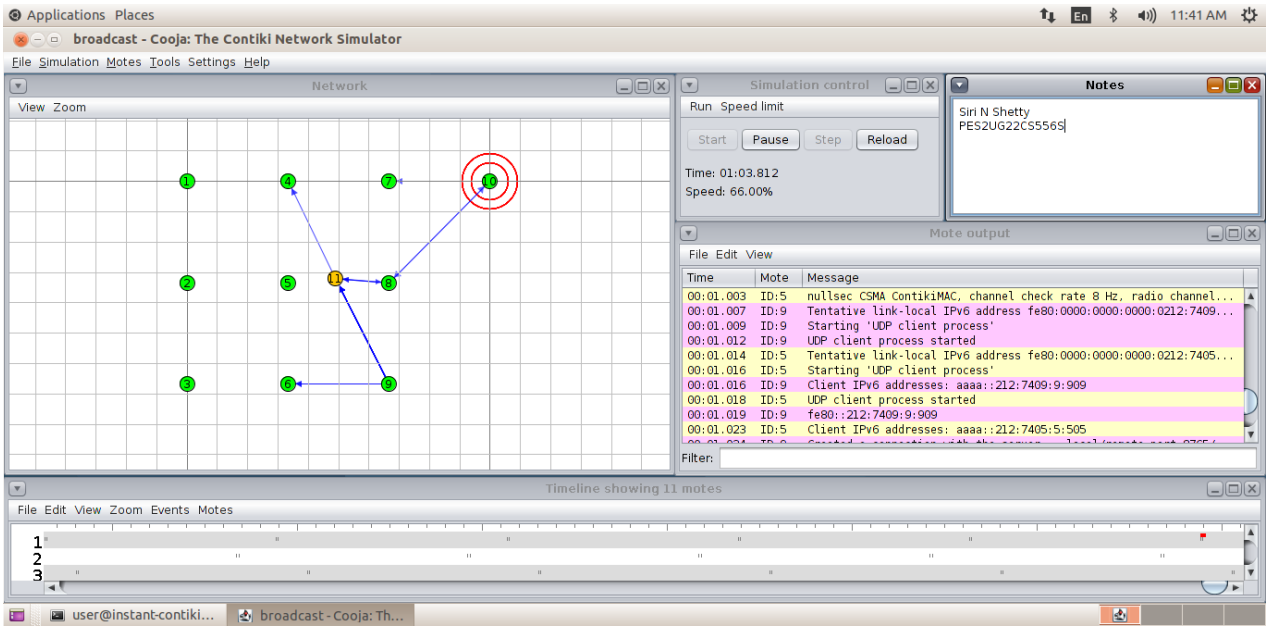
Developing Next-Gen IoT Solutions with Contiki OS and Cooja Simulator

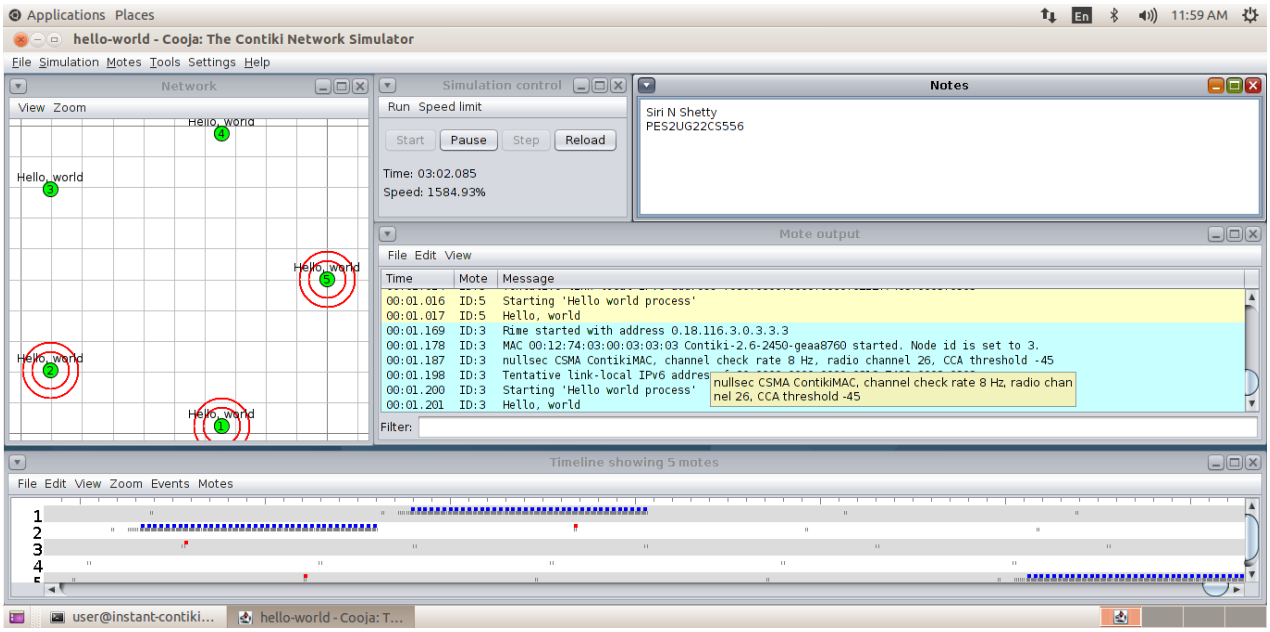
Submissions – Day 1 (8th July 2024)

Siri N Shetty (PES2UG22CS556)

1. Client Server Program

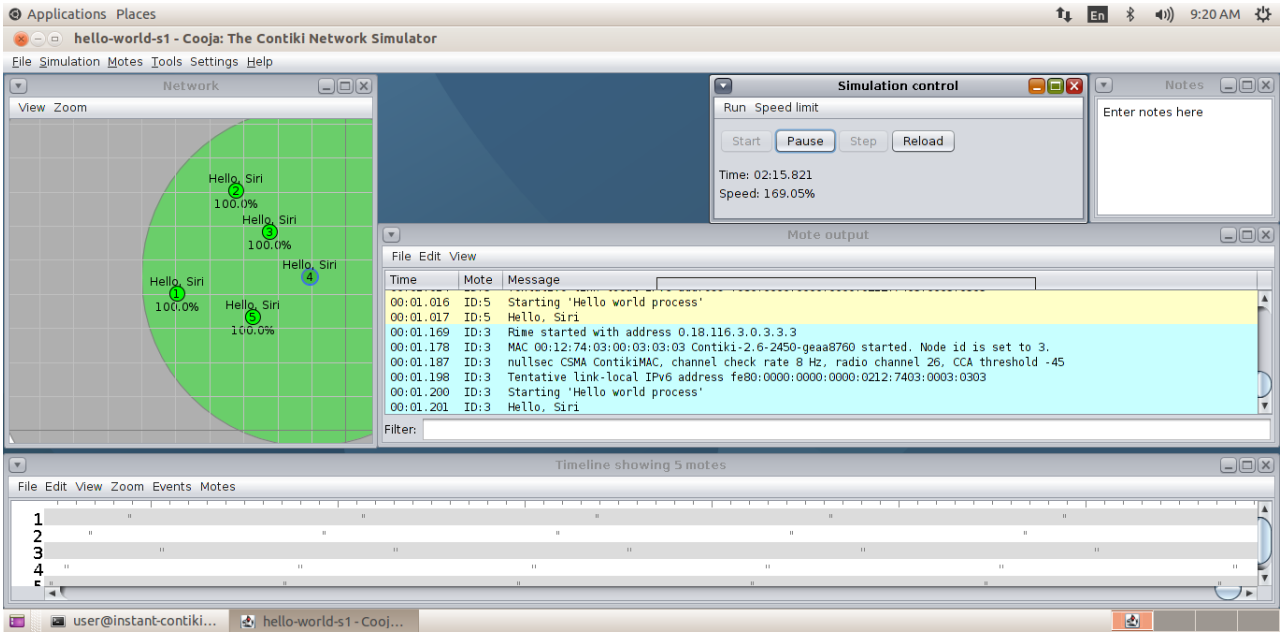


2. Hello World Program

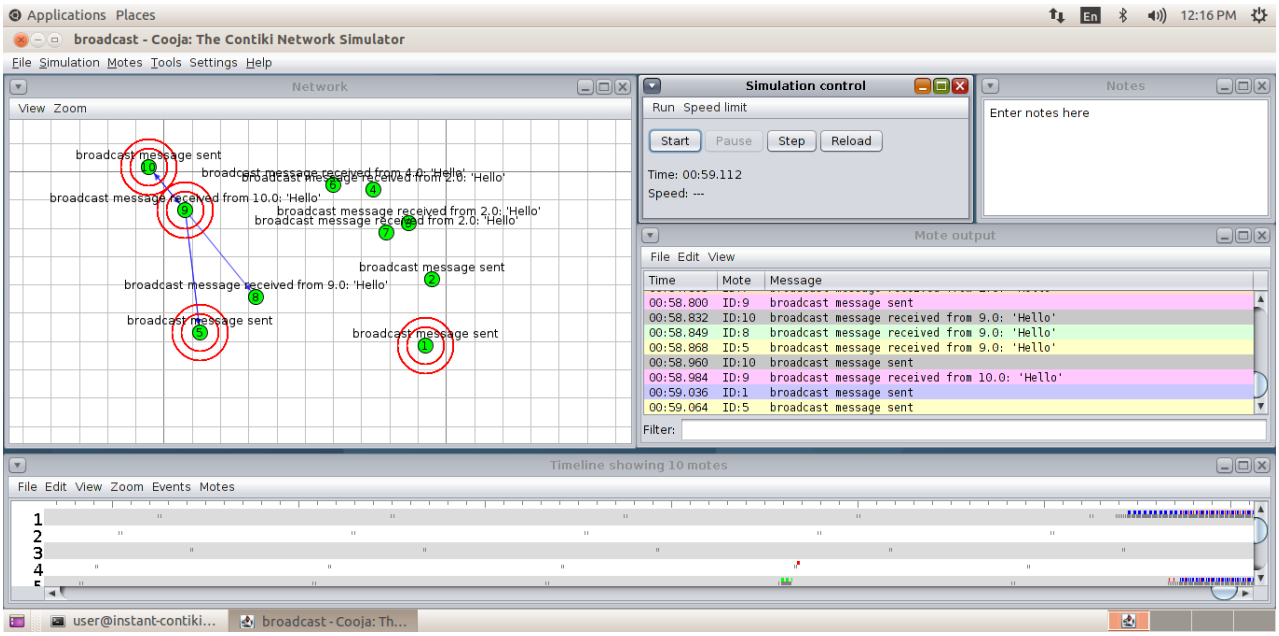


3. Modifying hello-world.c

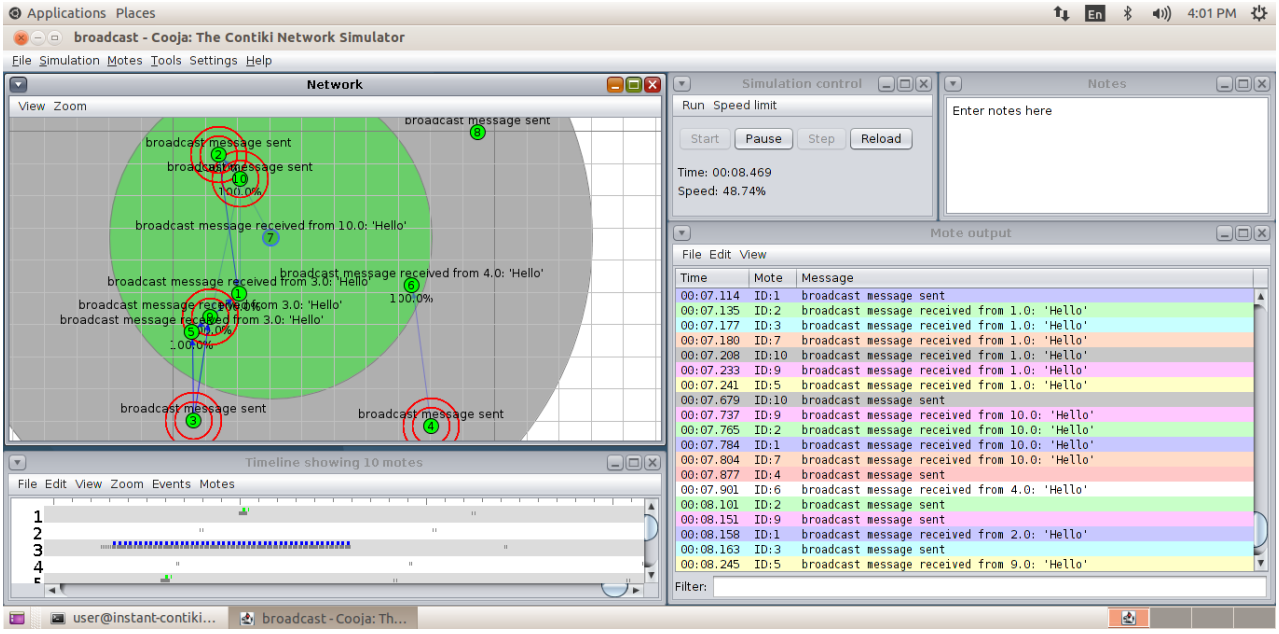
```
 * IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED.  IN NO EVENT SHALL THE INSTITUTE OR CONTRIBUTORS BE LIABLE
 * FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL
 * DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS
 * OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT
 * LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
 * OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF
 * SUCH DAMAGE.
 *
 * This file is part of the Contiki operating system.
 *
 */

/**
 * \file
 *         A very simple Contiki application showing how Contiki programs look
 * \author
 *         Adam Dunkels <adam@sics.se>
 */

#include "contiki.h"

#include <stdio.h> /* For printf() */
/*---------------------------------------------------------------------------*/
PROCESS(hello_world_process, "Hello world process");
AUTOSTART_PROCESSES(&hello_world_process);
/*---------------------------------------------------------------------------*/
PROCESS_THREAD(hello_world_process, ev, data)
{
  PROCESS_BEGIN();

  printf("Hello, Siri\n");

  PROCESS_END();
}
/*---------------------------------------------------------------------------*/
```



4. Broadcast Program

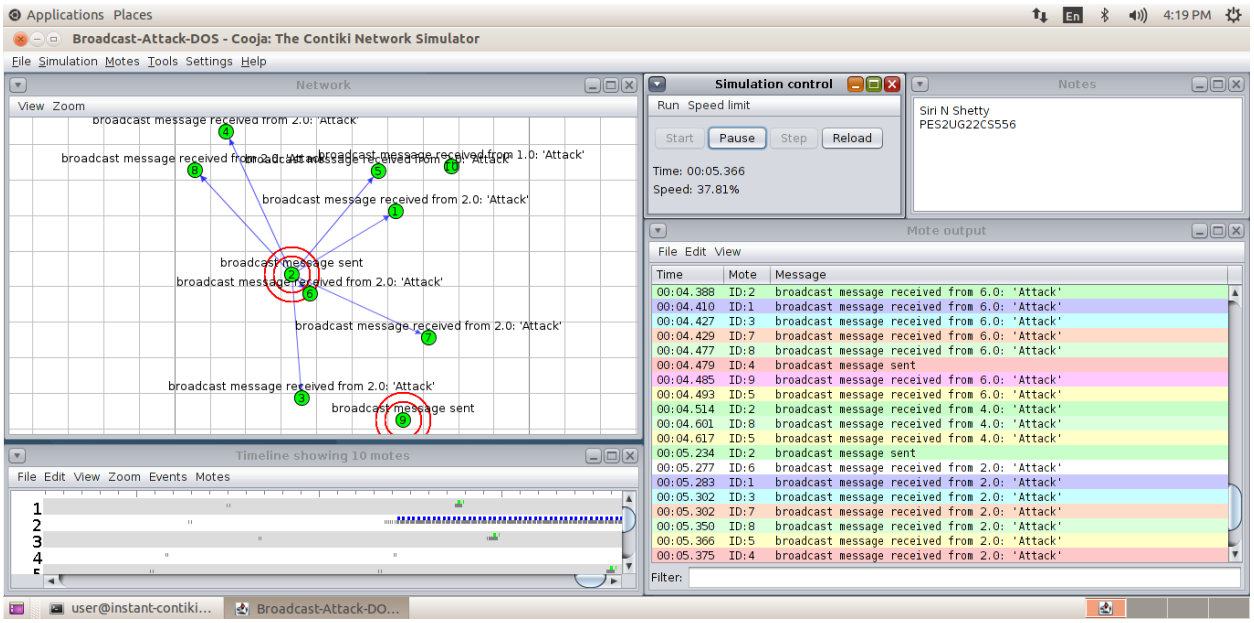



```
PROCESS(example_broadcast_process, "Broadcast example");
AUTOSTART_PROCESSES(&example_broadcast_process);
/*---------------------------------------------------------------------------*/
static void
broadcast_recv(struct broadcast_conn *c, const linkaddr_t *from)
{
  printf("broadcast message received from %d.%d: '%s'\n",
         from->u8[0], from->u8[1], (char *)packetbuf_dataptr());
}
static const struct broadcast_callbacks broadcast_call = {broadcast_recv};
static struct broadcast_conn broadcast;
/*---------------------------------------------------------------------------*/
PROCESS_THREAD(example_broadcast_process, ev, data)
{
  static struct etimer et;

  PROCESS_EXITHANDLER(broadcast_close(&broadcast);)

  PROCESS_BEGIN();

  broadcast_open(&broadcast, 129, &broadcast_call);

  while(1) {

    /* Delay 2-4 seconds */
    etimer_set(&et, CLOCK_SECOND * 2 + random_rand() % (CLOCK_SECOND * 2));

    PROCESS_WAIT_EVENT_UNTIL(etimer_expired(&et));

    packetbuf_copyfrom("Hello", 6);
    broadcast_send(&broadcast);
    printf("broadcast message sent\n");
  }

  PROCESS_END();
}
/*---------------------------------------------------------------------------*/
```



5. Altering sending rate in broadcast program (DOS attack)

```
PROCESS(example_broadcast_process, "Broadcast example");
AUTOSTART_PROCESSES(&example_broadcast_process);
/*---------------------------------------------------------------------------*/
static void
broadcast_recv(struct broadcast_conn *c, const linkaddr_t *from)
{
  printf("broadcast message received from %d.%d: '%s'\n",
         from->u8[0], from->u8[1], (char *)packetbuf_dataptr());
}
static const struct broadcast_callbacks broadcast_call = {broadcast_recv};
static struct broadcast_conn broadcast;
/*---------------------------------------------------------------------------*/
PROCESS_THREAD(example_broadcast_process, ev, data)
{
  static struct etimer et;

  PROCESS_EXITHANDLER(broadcast_close(&broadcast);)

  PROCESS_BEGIN();

  broadcast_open(&broadcast, 129, &broadcast_call);

  while(1) {

    /* Delay 2-4 seconds */
    etimer_set(&et, CLOCK_SECOND * 2 + random_rand() % (CLOCK_SECOND * 2));

    PROCESS_WAIT_EVENT_UNTIL(etimer_expired(&et));

    packetbuf_copyfrom("Attack", 7);
    broadcast_send(&broadcast);
    printf("broadcast message sent\n");
  }

  PROCESS_END();
}
/*---------------------------------------------------------------------------*/
```


Applications Places
4:19 PM
Broadcast-Attack-DOS - Cooja: The Contiki Network Simulator
File Simulation Motes Tools Settings Help
Network
View Zoom
broadcast message received from 2.0: 'Attack'
broadcast message received from 1.0: 'Attack'
broadcast message sent
Simulation control
Run Speed limit
Start Pause Step Reload
Time: 00:05.366
Speed: 37.81%
Notes
Siri N Shetty
PES2UG22CS556
Mote output
File Edit View
Time Mote Message
00:04.388 ID:2 broadcast message received from 6.0: 'Attack'
00:04.410 ID:1 broadcast message received from 6.0: 'Attack'
00:04.427 ID:3 broadcast message received from 6.0: 'Attack'
00:04.429 ID:7 broadcast message received from 6.0: 'Attack'
00:04.477 ID:8 broadcast message received from 6.0: 'Attack'
00:04.479 ID:4 broadcast message sent
00:04.485 ID:9 broadcast message received from 6.0: 'Attack'
00:04.493 ID:5 broadcast message received from 6.0: 'Attack'
00:04.514 ID:2 broadcast message received from 4.0: 'Attack'
00:04.601 ID:8 broadcast message received from 4.0: 'Attack'
00:04.617 ID:5 broadcast message received from 4.0: 'Attack'
00:05.234 ID:2 broadcast message sent
00:05.277 ID:6 broadcast message received from 2.0: 'Attack'
00:05.283 ID:1 broadcast message received from 2.0: 'Attack'
00:05.302 ID:3 broadcast message received from 2.0: 'Attack'
00:05.302 ID:7 broadcast message received from 2.0: 'Attack'
00:05.350 ID:8 broadcast message received from 2.0: 'Attack'
00:05.366 ID:5 broadcast message received from 2.0: 'Attack'
00:05.375 ID:4 broadcast message received from 2.0: 'Attack'
Filter:
Timeline showing 10 motes
File Edit View Zoom Events Motes
user@instant-contiki...
Broadcast-Attack-DO...
