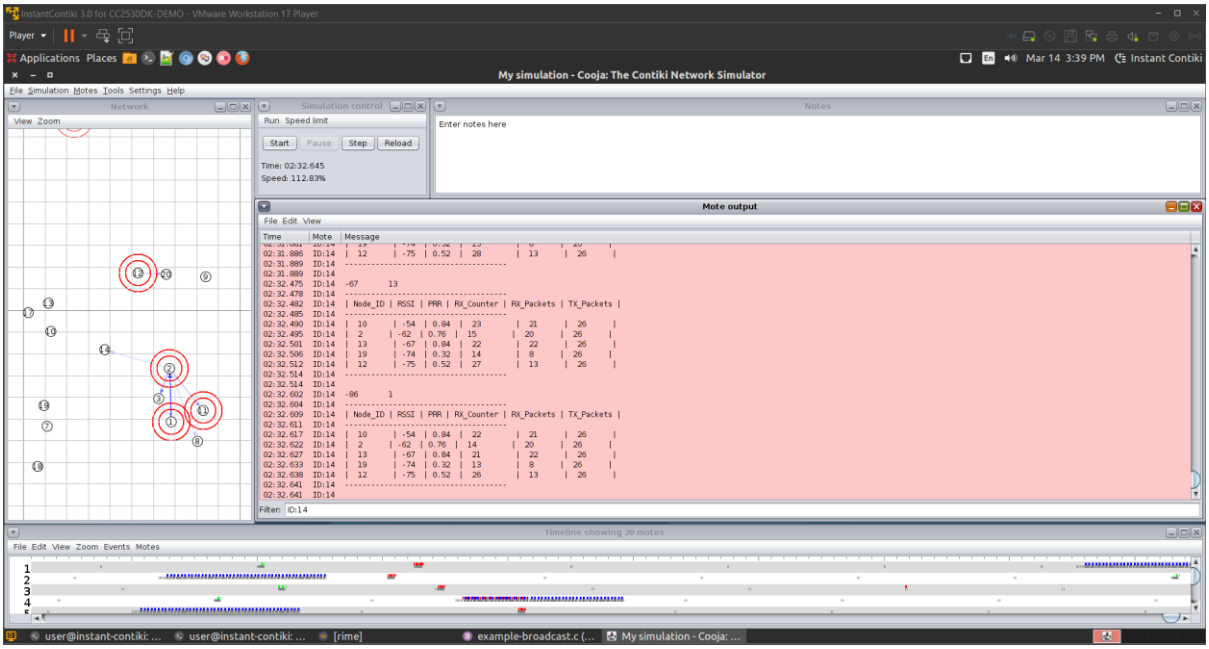
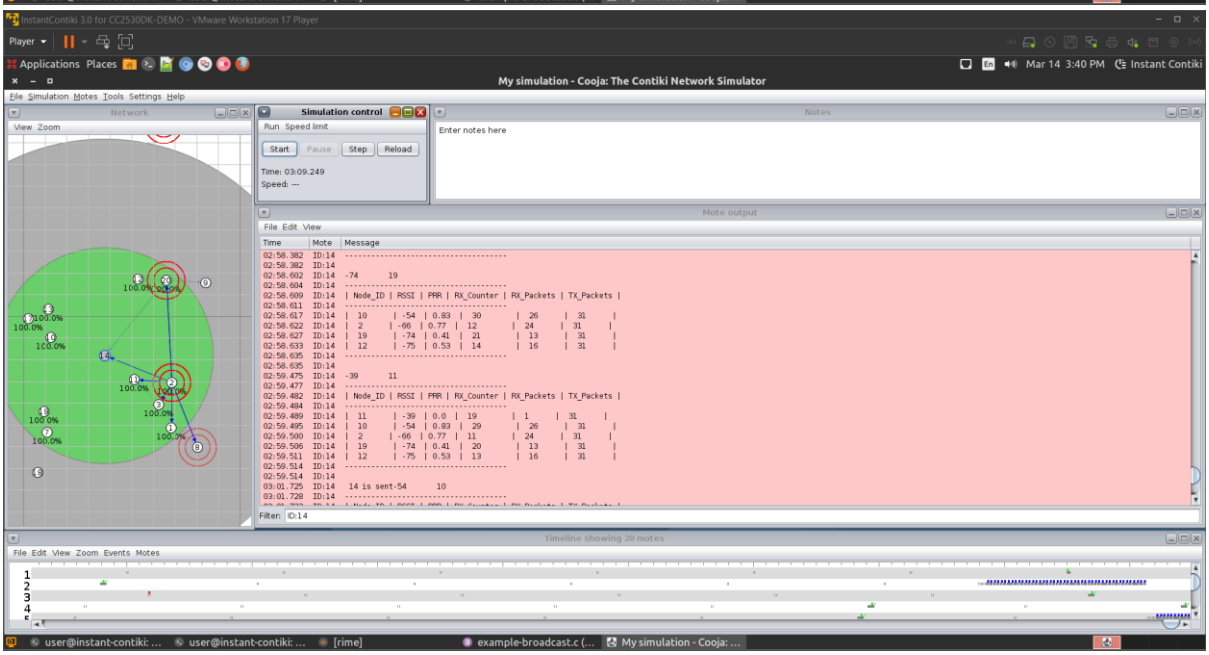
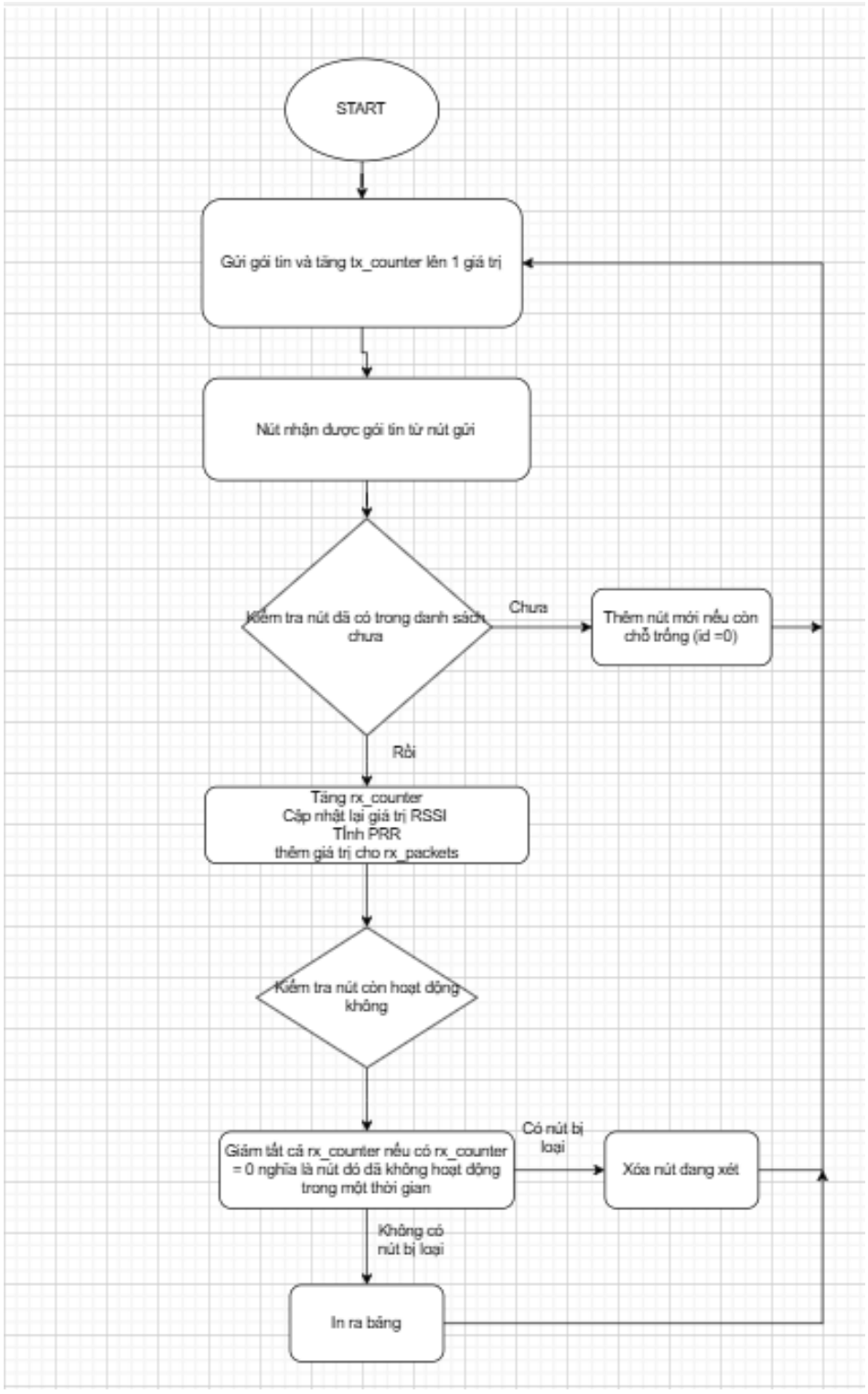

InstantContiki 3.0 for CC2530DK-DEMO - VMware Workstation 17 Player
Player
Applications Places
Mar 14 3:39 PM
Instant Contiki
My simulation - Cooja: The Contiki Network Simulator
File Simulation Motes Tools Settings Help
Network
View Zoom
Simulation control
Run Speed limit
Start
Pause
Step
Reload
Time: 02:32.645
Speed: 112.83%
Notes
Enter notes here
Mote output
File Edit View
Time
Mote
Message
02:32.475 ID:14 -67 13
02:32.482 ID:14 | Node_ID | RSSI | PRR | RX_Counter | RX_Packets | TX_Packets |
02:32.490 ID:14 | 10 | -54 | 0.84 | 23 | 21 | 26 |
02:32.495 ID:14 | 2 | -62 | 0.76 | 15 | 20 | 26 |
02:32.501 ID:14 | 13 | -67 | 0.84 | 22 | 22 | 26 |
02:32.506 ID:14 | 19 | -74 | 0.32 | 14 | 8 | 26 |
02:32.512 ID:14 | 12 | -75 | 0.52 | 27 | 13 | 26 |
02:32.602 ID:14 -86 1
02:32.609 ID:14 | Node_ID | RSSI | PRR | RX_Counter | RX_Packets | TX_Packets |
02:32.617 ID:14 | 10 | -54 | 0.84 | 22 | 21 | 26 |
02:32.622 ID:14 | 2 | -62 | 0.76 | 14 | 20 | 26 |
02:32.627 ID:14 | 13 | -67 | 0.84 | 21 | 22 | 26 |
02:32.633 ID:14 | 19 | -74 | 0.32 | 13 | 8 | 26 |
02:32.638 ID:14 | 12 | -75 | 0.52 | 26 | 13 | 26 |
Filter: ID:14
Timeline showing 20 motes
File Edit View Zoom Events Motes
user@instant-contiki: ...
user@instant-contiki: ...
[rime]
example-broadcast.c (...
My simulation - Cooja: ...


InstantContiki 3.0 for CC2530DK-DEMO - VMware Workstation 17 Player
Player
Applications Places
Mar 14 3:40 PM
Instant Contiki
My simulation - Cooja: The Contiki Network Simulator
File Simulation Motes Tools Settings Help
Network
View Zoom
Simulation control
Run Speed limit
Start
Pause
Step
Reload
Time: 03:09.249
Speed: ---
Notes
Enter notes here
Mote output
File Edit View
Time
Mote
Message
02:58.382 ID:14
02:58.602 ID:14 -74 19
02:58.609 ID:14 | Node_ID | RSSI | PRR | RX_Counter | RX_Packets | TX_Packets |
02:58.617 ID:14 | 10 | -54 | 0.83 | 30 | 26 | 31 |
02:58.622 ID:14 | 2 | -66 | 0.77 | 12 | 24 | 31 |
02:58.627 ID:14 | 19 | -74 | 0.41 | 21 | 13 | 31 |
02:58.633 ID:14 | 12 | -75 | 0.53 | 14 | 16 | 31 |
02:59.475 ID:14 -39 11
02:59.477 ID:14 | Node_ID | RSSI | PRR | RX_Counter | RX_Packets | TX_Packets |
02:59.489 ID:14 | 11 | -39 | 0.0 | 19 | 1 | 31 |
02:59.495 ID:14 | 10 | -54 | 0.83 | 29 | 26 | 31 |
02:59.500 ID:14 | 2 | -66 | 0.77 | 11 | 24 | 31 |
02:59.506 ID:14 | 19 | -74 | 0.41 | 20 | 13 | 31 |
02:59.511 ID:14 | 12 | -75 | 0.53 | 13 | 16 | 31 |
02:59.514 ID:14
03:01.725 ID:14 14 is sent-54 10
Filter: ID:14
Timeline showing 20 motes
File Edit View Zoom Events Motes
user@instant-contiki: ...
user@instant-contiki: ...
[rime]
example-broadcast.c (...
My simulation - Cooja: ...




```
#include "contiki.h"

#include "net/rime/rime.h"

 #include "random.h"

#include "dev/leds.h"
```

```
#include <stdio.h>

#define MAX_NEIGHBORS 5

PROCESS(example_broadcast_process, "Broadcast Example");
AUTOSTART_PROCESSES(&example_broadcast_process);

typedef struct {

 uint8_t id_0;

int rssi;

 int prr;

int rx_counter;

 int rx_packets;

int tx_packets;

 } Neighbor;

static int tx_counter;

static Neighbor neighbors[MAX_NEIGHBORS];

 static struct broadcast_conn broadcast;

int find_neighbor(uint8_t id_0) {

for (int i = 0; i < MAX_NEIGHBORS; i++) {

                if (neighbors[i].id_0 == id_0) {

                return i;

               }

      }

return -1; }

void update_rx_packets(uint8_t id_0, int rssi) {

int index = find_neighbor(id_0);

if (index != -1) {

 neighbors[index].rx_packets++;

neighbors[index].rssi = rssi;

if (neighbors[index].tx_packets > 0) {

 neighbors[index].prr = neighbors[index].rx_packets * 100 /neighbors[index].tx_packets; }
```

```
 if (neighbors[index].rx_counter < 30) {
 neighbors[index].rx_counter += 10; }
} else {
for (int i = 0; i < MAX_NEIGHBORS; i++) {
if (neighbors[i].id_0 == 0) {
neighbors[i].id_0 = id_0;
neighbors[i].rssi = rssi;
neighbors[i].rx_counter = 20; neighbors[i].rx_packets = 1;
neighbors[i].tx_packets = tx_counter; return;
}
}
}
}
void update_tx_packets() {
for (int i = 0; i < MAX_NEIGHBORS; i++) {
neighbors[i].tx_packets = tx_counter;
}
}
void sort_neighbors_by_rssi() {
for (int i = 0; i < MAX_NEIGHBORS - 1; i++) {
for (int j = i + 1; j < MAX_NEIGHBORS; j++) {
 if (neighbors[i].rssi < neighbors[j].rssi) {
Neighbor temp = neighbors[i]; neighbors[i] = neighbors[j]; neighbors[j] = temp;
}
}
}
}
void remove_inactive_neighbors() {
```

```
for (int i = 0; i < MAX_NEIGHBORS; i++) {
if (neighbors[i].id_0 != 0) {
if (neighbors[i].rx_counter > 0) {
neighbors[i].rx_counter--;
}
if (neighbors[i].rx_counter == 0) {
memset(&neighbors[i], 0, sizeof(Neighbor));
}
}
}
}
void convert_inttofloat(int num, int *quotient, int *remainder) {
*quotient = num / 100; *remainder = num % 100;
}
void print_neighbors_table() {
sort_neighbors_by_rssi();
printf("\n--------------------------------------\n");
printf("| Node_ID | RSSI | PRR | RX_Counter | RX_Packets | TX_Packets |\n");
printf("--------------------------------------\n");
for (int i = 0; i < MAX_NEIGHBORS; i++) {
if (neighbors[i].id_0 != 0) {
int quotient, remainder; convert_inttofloat(neighbors[i].prr, &quotient, &remainder);
printf("| %d | %d | %d.%d | %d | %d | %d | \n", neighbors[i].id_0, neighbors[i].rssi, quotient, remainder,
neighbors[i].rx_counter, neighbors[i].rx_packets, neighbors[i].tx_packets);
}
}
printf("--------------------------------------\n\n"); }
static void broadcast_recv(struct broadcast_conn *c, const linkaddr_t *from) {
int16_t rssi = packetbuf_attr(PACKETBUF_ATTR_RSSI);
```

```
printf("%d %d", rssi, from->u8[0]); update_tx_packets();

update_rx_packets(from->u8[0], rssi);

remove_inactive_neighbors(); print_neighbors_table();

}

static const struct broadcast_callbacks broadcast_call = {broadcast_recv};

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) {
   etimer_set(&et, CLOCK_SECOND * 4 + random_rand() % (CLOCK_SECOND * 4));
   PROCESS_WAIT_EVENT_UNTIL(etimer_expired(&et));
   printf(" %d is sent", linkaddr_node_addr.u8[0]);

   packetbuf_copyfrom(" ", 6);
   broadcast_send(&broadcast);
   tx_counter++;
}

PROCESS_END();


}
```