WSN Aggregation Project

Atul Kumar (IIT2018030), Alok Patel (MIT2020088)

B.tech & M.tech

Indian Institute of Information Technology Allahabad

Abstract

CTP, a collection routing protocol for wireless sensor networks, is described. CTP uses data traffic to actively probe the topology, instantly detecting and correcting routing issues. The goal of this project is to establish event aggregation in a wireless sensor network in order to reduce communication overhead and communication energy consumption. In this paper we have to apply changes to collect tree protocol in Cooja Simulator to make it more efficient in terms of resource consumption by applying aggregation at every node.

In wireless sensor networks there are many resource constraints to operate like they work on batteries and small power so they have limited energy to receive. So, making it energy efficient is a key task.

1. Introduction

As wireless sensor networks are low power lossy networks. So in a sensor network there is very limited resource and routing protocols are made in a way that they must consume very little energy.

So every effort is made in research to make routing algorithms very energy efficient. We have run the collect tree protocol with some simulation and made some observations, and then we have made changes to example_collect.c to make it more efficient by method of aggregation.

We are generating events by script.txt. Script generates events and sensors respond to it if in range.

2. Preliminaries

2.1.ContikiOS

contiki only works on ***contention-based*** *MAC protocol.*

- ***contention-based protocol*** *Is based on Carrier Sensing for detecting medium activities and are very easy to implement.*

Medium Access Control protocols take care of the organization of medium access in wireless networks. These are the rules that coordinate when each node is going to transmit/receive packets.

ContikiOS includes Physical and the Network layer, in-between we have 3 different layers: ***Framer****,* ***Radio Duty-Cycle (RDC)*** *and* ***Medium Access Control (MAC)****.*

- *Framer layer is altogether a collection of auxiliary functions that are being called for creating a frame with the data to be transmitted and parsing of the data being received.*

The Radio Duty-Cycle (RDC) layer takes care of the sleeping period of nodes. It's important layer because it is the one responsible for deciding exactly when the packet will be transmitted and also very responsible for making sure that the nodes are awake when packets are to be received.

2.2 Collection Tree Protocol

The majority of sensor networks are used to gather data from the physical world. Sensor networks are used to monitor microclimates in agriculture farms and to measure energy use in commercial and residential buildings, for example. The nodes in these networks use their sensors to collect information about the physical world, and then use multi-hop wireless communication to send the sensor data to a central base station or server.

*The Collection Tree protocol(****CTP****) is a routing* ***protocol*** *for* ***wireless*** *sensor* ***networks****. It is used for transferring data from one or more sensors to one or more root nodes. Collecting information from nodes in a sensor network in a reliable and effective manner is a difficult task, especially given the wireless dynamics. In a* *dynamic wireless environment, multihop routing necessitates the ability of a protocol to react fast to network changes (agility), yet the energy constraints of sensor networks mandate that such mechanisms do not necessitate excessive communication among nodes (efficiency). CTP is a collection routing technique for sensor networks that optimises both agility and efficiency while providing highly reliable data transmission.*

CTP has been used in commercial products, research, and teaching. The IPv6 Routing Protocol for Low Power and Lossy Networks was designed using lessons learned from CTP (RPL).

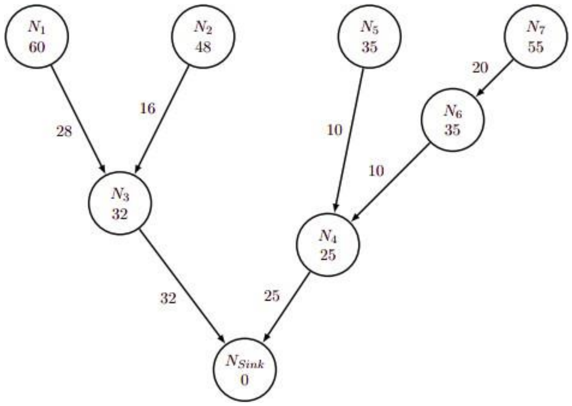
CTP Mechanism:

CTP is a sensor network-specific distance vector routing protocol. CTP calculates the routes between each node in the network and the network's root (given destinations). To solve the obstacles that the distance vector routing protocol faces in a highly dynamic wireless network, CTP employs these three mechanisms:

- a) *Agile and Accurate Link Quality Estimation*
- b) *Datapath Validation*
- c) *Adaptive Beaconing*

In simple words, The Collection Tree System (CTP) is a sensor network routing protocol. As seen in Figure, the protocol builds a collection tree. The root node, also known as the sink node, sits at the bottom of the tree and receives all of the data. The nodes N1 through N7 could be basic relay nodes that forward packets or they might also be data generators. The most crucial aspect is that all data is routed to the root, regardless of which node sends it.

This means that each node assigns a quality rating to all routes to neighbouring nodes, and only the most appropriate route is chosen. To the sink, each node has a name and an expected transmission coefficient (ETX). This ETX is calculated by determining the quality of the node-to-node interconnections and assigning a value to them. The ETX of the node in question is calculated by adding the values of the linkages between the sink and the node. Because each node calculates its own ETX, all a node needs to calculate its own ETX is its parent's ETX and the link quality to its parent. The sink node ETX is set to zero, as can be seen.



3. Proposed approach

The proposed method aims at implementing aggregation in Wireless sensor networks for which we have collect.c file to perform aggregation and we have script.txt to create events in cooja simulator.

3.1 Working of script.txt

Every node during start of simulation tells about their coordinate in the logs and scripts read from the logs and find out how many nodes are there in the simulation.

Script.txt generates random events out of the range of given events in array ref[] at some random points. If sensor nodes are in range they sense the event using the serial port and send it to the sink node. Sensor nodes have a range of 15m any node which senses the event sends message and event id to sink node. If the event is in the range of any sensor node that is within 15m it will receive the message and send it to the sink node otherwise the event will be missed.

The coordinate helps in finding which event will be sensed by which node. After every node has sensed its coordinates then events are generated by scripts.

To run script.txt we have to replace simulation script with script.txt in our simulation and then activate that script.

3.2 Working of Example_collect.c

At start of Example_collect.c random events are generated by script.txt at random points then sensor node try to sense the event if in range with the help of ***PROCESS_THREAD(example_collect_process, ev, data)*** *function in which ev represents the event and data represents the message of the event. if there's any Event message at the serial port then it will store the message in the* ***packetbuf*** *and send it to the* ***sink*** *node via parent's****.***
packetbuf *is a single data structure to carry messages or information of packets in the whole network layer. Messages are sent and received through packetbuf.*

Aggregation technique used:

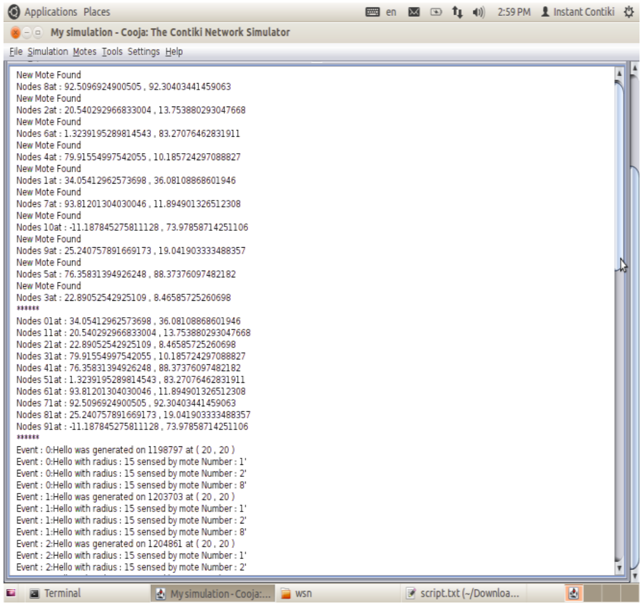
1. We have applied some delay.
2. And between that delay we have collected all the packets and stored their messages in the single array and then we have sent that aggregated array to the sink node.
3. By this way, We have improved efficiency by aggregated sending and reduced power consumption.

Results and Conclusion:

For 10 motes, we have got the following output when running/applying CTP protocol for generated events in our script.txt file.

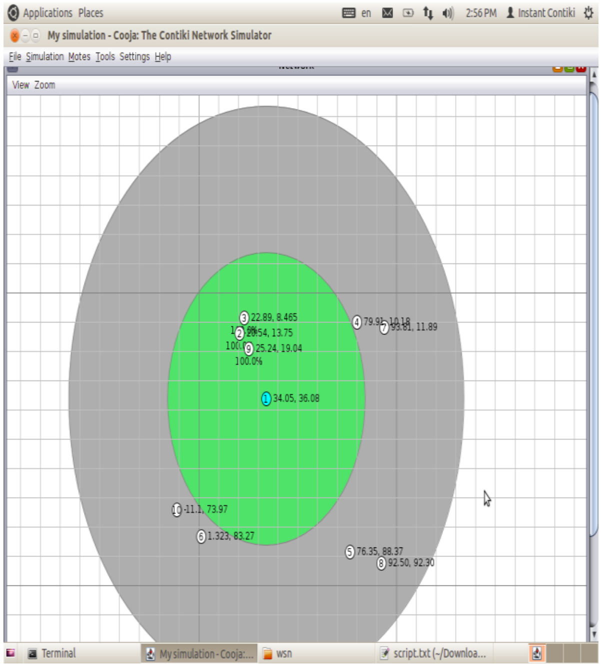
1) Script.txt file generates events sending "Hello" message to sink node(which is by default Node ID:1) when given range satisfies. i.e., radius=15.

Furthermore, it gives output of coordinates of sensor nodes and tells where the event has been generated and which sensor has sensed the generated events and if there are no sensor nodes who sense the generated events then nodes are found to be out of range.


Applications Places
en
2:59 PM
Instant Contiki
My simulation - Cooja: The Contiki Network Simulator
File Simulation Motes Tools Settings Help
Terminal
My simulation - Cooja:...
wsn
script.txt (~/Downloa...

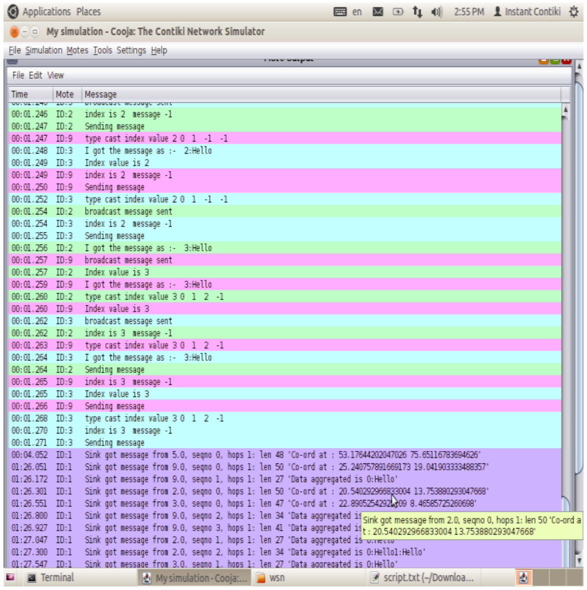
```
New Mote Found
Nodes 8at : 92.5096924900505 , 92.30403441459063
New Mote Found
Nodes 2at : 20.540292966833004 , 13.753880293047668
New Mote Found
Nodes 6at : 1.3239195289814543 , 83.27076462831911
New Mote Found
Nodes 4at : 79.91554997542055 , 10.185724297088827
New Mote Found
Nodes 1at : 34.05412962573698 , 36.08108868601946
New Mote Found
Nodes 7at : 93.81201304030046 , 11.894901326512308
New Mote Found
Nodes 10at : -11.187845275811128 , 73.97858714251106
New Mote Found
Nodes 9at : 25.240757891669173 , 19.041903333488357
New Mote Found
Nodes 5at : 76.35831394926248 , 88.37376097482182
New Mote Found
Nodes 3at : 22.89052542925109 , 8.46585725260698
******
Nodes 01at : 34.05412962573698 , 36.08108868601946
Nodes 11at : 20.540292966833004 , 13.753880293047668
Nodes 21at : 22.89052542925109 , 8.46585725260698
Nodes 31at : 79.91554997542055 , 10.185724297088827
Nodes 41at : 76.35831394926248 , 88.37376097482182
Nodes 51at : 1.3239195289814543 , 83.27076462831911
Nodes 61at : 93.81201304030046 , 11.894901326512308
Nodes 71at : 92.5096924900505 , 92.30403441459063
Nodes 81at : 25.240757891669173 , 19.041903333488357
Nodes 91at : -11.187845275811128 , 73.97858714251106
******
Event : 0:Hello was generated on 1198797 at ( 20 , 20 )
Event : 0:Hello with radius : 15 sensed by mote Number : 1'
Event : 0:Hello with radius : 15 sensed by mote Number : 2'
Event : 0:Hello with radius : 15 sensed by mote Number : 8'
Event : 1:Hello was generated on 1203703 at ( 20 , 20 )
Event : 1:Hello with radius : 15 sensed by mote Number : 1'
Event : 1:Hello with radius : 15 sensed by mote Number : 2'
Event : 1:Hello with radius : 15 sensed by mote Number : 8'
Event : 2:Hello was generated on 1204861 at ( 20 , 20 )
Event : 2:Hello with radius : 15 sensed by mote Number : 1'
Event : 2:Hello with radius : 15 sensed by mote Number : 2'
```

2) Now, we have given a radius of 15. So, when any node comes within this range from the sink node. Then a message will be sent to sink.
Here, node 2,3,9 is falling within this range.
So, transmission of messages will be done between them.



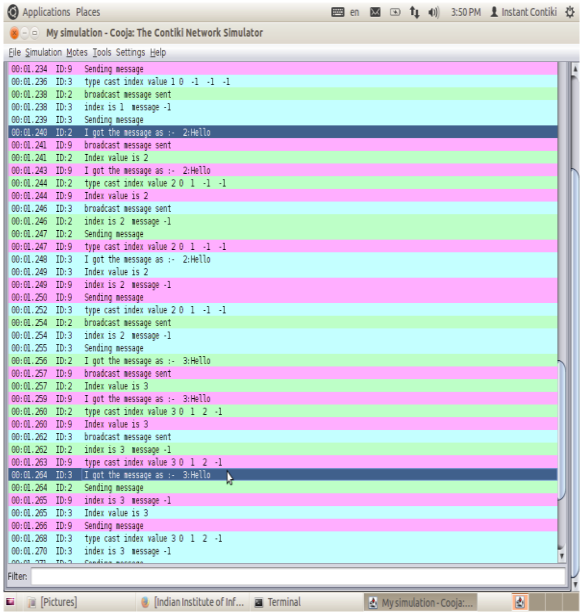
3) Now, this output shows aggregation of messages from 2,3,9 to our sink node 1.

And we know, The first packet with the event id received at the sink node gives the best ERT and the last packet with event id received at the sink node gives the Worst ERT. So, by seeing output node ID:2 arrives first and Node ID:3 arrives last.
Therefore Best ERT: Node ID 2
Worst ERT: Node ID:3


Applications Places
en 2:55 PM Instant Contiki
My simulation - Cooja: The Contiki Network Simulator
File Simulation Motes Tools Settings Help
File Edit View
Terminal
My simulation - Cooja:...
wsn
script.txt (~/Downloa...

```
Time       Mote   Message
00:01.246  ID:2   index is 2  message -1
00:01.247  ID:2   Sending message
00:01.247  ID:9   type cast index value 2 0  1  -1  -1
00:01.248  ID:3   I got the message as :-  2:Hello
00:01.249  ID:3   Index value is 2
00:01.249  ID:9   index is 2  message -1
00:01.250  ID:9   Sending message
00:01.252  ID:3   type cast index value 2 0  1  -1  -1
00:01.254  ID:2   broadcast message sent
00:01.254  ID:3   index is 2  message -1
00:01.255  ID:3   Sending message
00:01.256  ID:2   I got the message as :-  3:Hello
00:01.257  ID:9   broadcast message sent
00:01.257  ID:2   Index value is 3
00:01.259  ID:9   I got the message as :-  3:Hello
00:01.260  ID:2   type cast index value 3 0  1  2  -1
00:01.260  ID:9   Index value is 3
00:01.262  ID:3   broadcast message sent
00:01.262  ID:2   index is 3  message -1
00:01.263  ID:9   type cast index value 3 0  1  2  -1
00:01.264  ID:3   I got the message as :-  3:Hello
00:01.264  ID:2   Sending message
00:01.265  ID:9   index is 3  message -1
00:01.265  ID:3   Index value is 3
00:01.266  ID:9   Sending message
00:01.268  ID:3   type cast index value 3 0  1  2  -1
00:01.270  ID:3   index is 3  message -1
00:01.271  ID:3   Sending message
00:04.052  ID:1   Sink got message from 5.0, seqno 0, hops 1: len 48 'Co-ord at : 53.17644202047026 75.65116783694626'
01:26.051  ID:1   Sink got message from 9.0, seqno 0, hops 1: len 50 'Co-ord at : 25.240757891669173 19.041903333488357'
01:26.172  ID:1   Sink got message from 9.0, seqno 1, hops 1: len 27 'Data aggregated is 0:Hello'
01:26.301  ID:1   Sink got message from 2.0, seqno 0, hops 1: len 50 'Co-ord at : 20.540292966833004 13.753880293047668'
01:26.551  ID:1   Sink got message from 3.0, seqno 0, hops 1: len 47 'Co-ord at : 22.8905254292 09 8.46585725260698'
01:26.800  ID:1   Sink got message from 9.0, seqno 2, hops 1: len 34 'Data aggregated is
01:26.927  ID:1   Sink got message from 9.0, seqno 3, hops 1: len 41 'Data aggregated is
01:27.047  ID:1   Sink got message from 2.0, seqno 1, hops 1: len 27 'Data aggregated is
01:27.300  ID:1   Sink got message from 2.0, seqno 2, hops 1: len 34 'Data aggregated is 0:Hello1:Hello'
01:27.547  ID:1   Sink got message from 3.0, seqno 1, hops 1: len 27 'Data aggregated is 0:Hello'
```

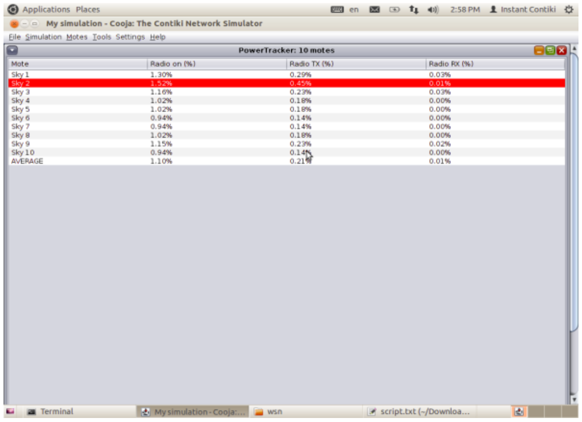
Sink got message from 2.0, seqno 0, hops 1: len 50 'Co-ord a
t : 20.540292966833004 13.753880293047668'


Applications Places
en 3:50 PM Instant Contiki
My simulation - Cooja: The Contiki Network Simulator
File Simulation Motes Tools Settings Help
Filter:
[Pictures]
[Indian Institute of Inf...
Terminal
My simulation - Cooja:...

```
00:01.234  ID:9   Sending message
00:01.236  ID:3   type cast index value 1 0  -1  -1  -1
00:01.238  ID:2   broadcast message sent
00:01.238  ID:3   index is 1  message -1
00:01.239  ID:3   Sending message
00:01.240  ID:2   I got the message as :-  2:Hello
00:01.241  ID:9   broadcast message sent
00:01.241  ID:2   Index value is 2
00:01.243  ID:9   I got the message as :-  2:Hello
00:01.244  ID:2   type cast index value 2 0  1  -1  -1
00:01.244  ID:9   Index value is 2
00:01.246  ID:3   broadcast message sent
00:01.246  ID:2   index is 2  message -1
00:01.247  ID:2   Sending message
00:01.247  ID:9   type cast index value 2 0  1  -1  -1
00:01.248  ID:3   I got the message as :-  2:Hello
00:01.249  ID:3   Index value is 2
00:01.249  ID:9   index is 2  message -1
00:01.250  ID:9   Sending message
00:01.252  ID:3   type cast index value 2 0  1  -1  -1
00:01.254  ID:2   broadcast message sent
00:01.254  ID:3   index is 2  message -1
00:01.255  ID:3   Sending message
00:01.256  ID:2   I got the message as :-  3:Hello
00:01.257  ID:9   broadcast message sent
00:01.257  ID:2   Index value is 3
00:01.259  ID:9   I got the message as :-  3:Hello
00:01.260  ID:2   type cast index value 3 0  1  2  -1
00:01.260  ID:9   Index value is 3
00:01.262  ID:3   broadcast message sent
00:01.262  ID:2   index is 3  message -1
00:01.263  ID:9   type cast index value 3 0  1  2  -1
00:01.264  ID:3   I got the message as :-  3:Hello
00:01.264  ID:2   Sending message
00:01.265  ID:9   index is 3  message -1
00:01.265  ID:3   Index value is 3
00:01.266  ID:9   Sending message
00:01.268  ID:3   type cast index value 3 0  1  2  -1
00:01.270  ID:3   index is 3  message -1
```

4) After all that, our main task was to reduce power consumption. We can check power consumption by observing the mote Radio duty cycle.

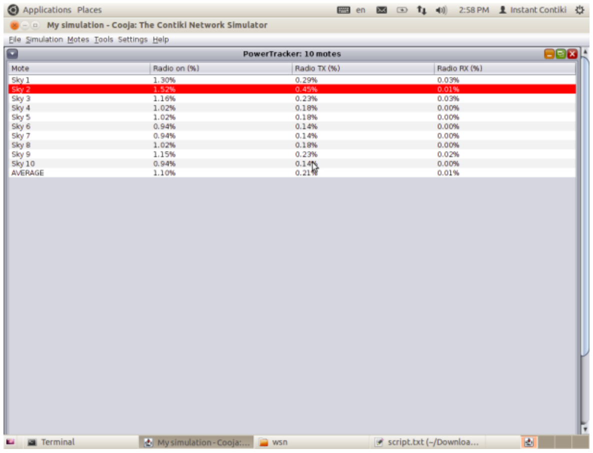
We have got the following output:


Applications Places
en 2:58 PM Instant Contiki
My simulation - Cooja: The Contiki Network Simulator
File Simulation Motes Tools Settings Help
PowerTracker: 10 motes
Terminal My simulation - Cooja:... wsn script.txt (~/Downloa...

Mote	Radio on (%)	Radio TX (%)	Radio RX (%)
Sky 1	1.30%	0.29%	0.03%
Sky 2	1.52%	0.45%	0.01%
Sky 3	1.16%	0.23%	0.03%
Sky 4	1.02%	0.18%	0.00%
Sky 5	1.02%	0.18%	0.00%
Sky 6	0.94%	0.14%	0.00%
Sky 7	0.94%	0.14%	0.00%
Sky 8	1.02%	0.18%	0.00%
Sky 9	1.15%	0.23%	0.02%
Sky 10	0.94%	0.14%	0.00%
AVERAGE	1.10%	0.21%	0.01%

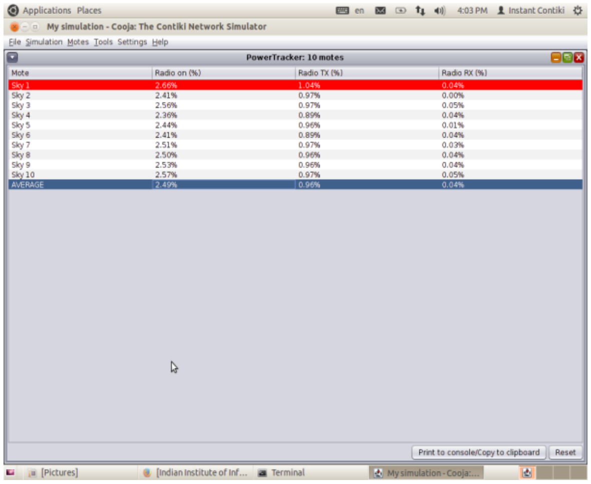
We can clearly see that communication is happening between 2,3,9 and our sink node. And therefore TX and RX are giving some value here.

Avg Radio on(%)= 1.10
Max radio on(%)=1.52


Applications Places
en 2:58 PM Instant Contiki
My simulation - Cooja: The Contiki Network Simulator
File Simulation Motes Tools Settings Help
PowerTracker: 10 motes
Terminal My simulation - Cooja:... wsn script.txt (~/Downloa...

Mote	Radio on (%)	Radio TX (%)	Radio RX (%)
Sky 1	1.30%	0.29%	0.03%
Sky 2	1.52%	0.45%	0.01%
Sky 3	1.16%	0.23%	0.03%
Sky 4	1.02%	0.18%	0.00%
Sky 5	1.02%	0.18%	0.00%
Sky 6	0.94%	0.14%	0.00%
Sky 7	0.94%	0.14%	0.00%
Sky 8	1.02%	0.18%	0.00%
Sky 9	1.15%	0.23%	0.02%
Sky 10	0.94%	0.14%	0.00%
AVERAGE	1.10%	0.21%	0.01%

5) But when we check for powertrace without collect protocol then it gives following output:

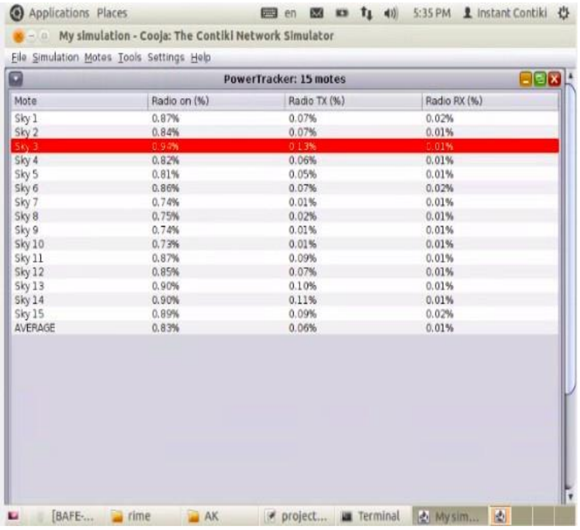

Applications Places
en 4:03 PM Instant Contiki
My simulation - Cooja: The Contiki Network Simulator
File Simulation Motes Tools Settings Help
PowerTracker: 10 motes
Print to console/Copy to clipboard
Reset
[Pictures] [Indian Institute of Inf... Terminal My simulation - Cooja:...

Mote	Radio on (%)	Radio TX (%)	Radio RX (%)
Sky 1	2.66%	1.04%	0.04%
Sky 2	2.41%	0.97%	0.00%
Sky 3	2.56%	0.97%	0.05%
Sky 4	2.36%	0.89%	0.04%
Sky 5	2.44%	0.96%	0.01%
Sky 6	2.41%	0.89%	0.04%
Sky 7	2.51%	0.97%	0.03%
Sky 8	2.50%	0.96%	0.04%
Sky 9	2.53%	0.96%	0.04%
Sky 10	2.57%	0.97%	0.05%
AVERAGE	2.49%	0.96%	0.04%

Similarly powertrace for 15 and 20 nodes for CTP protocol and without CTP protocol is added below:

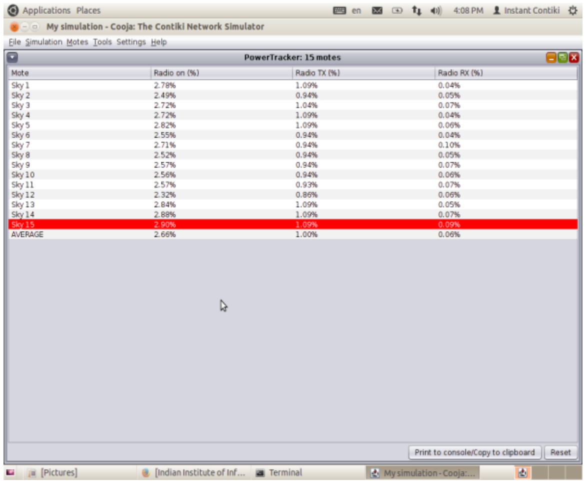
a) For 15 nodes:

With CTP:


Applications Places
en 5:35 PM Instant Contiki
My simulation - Cooja: The Contiki Network Simulator
File Simulation Motes Tools Settings Help
PowerTracker: 15 motes
[BAFE-... rime AK project... Terminal My sim...

Mote	Radio on (%)	Radio TX (%)	Radio RX (%)
Sky 1	0.87%	0.07%	0.02%
Sky 2	0.84%	0.07%	0.01%
Sky 3	0.94%	0.13%	0.01%
Sky 4	0.82%	0.06%	0.01%
Sky 5	0.81%	0.05%	0.01%
Sky 6	0.86%	0.07%	0.02%
Sky 7	0.74%	0.01%	0.01%
Sky 8	0.75%	0.02%	0.01%
Sky 9	0.74%	0.01%	0.01%
Sky 10	0.73%	0.01%	0.01%
Sky 11	0.87%	0.09%	0.01%
Sky 12	0.85%	0.07%	0.01%
Sky 13	0.90%	0.10%	0.01%
Sky 14	0.90%	0.11%	0.01%
Sky 15	0.89%	0.09%	0.02%
AVERAGE	0.83%	0.06%	0.01%

Without CTP:


Applications Places
en
4:08 PM
Instant Contiki
File Simulation Motes Tools Settings Help
Print to console/Copy to clipboard
Reset
[Pictures]
[Indian Institute of Inf...
Terminal
My simulation - Cooja:...

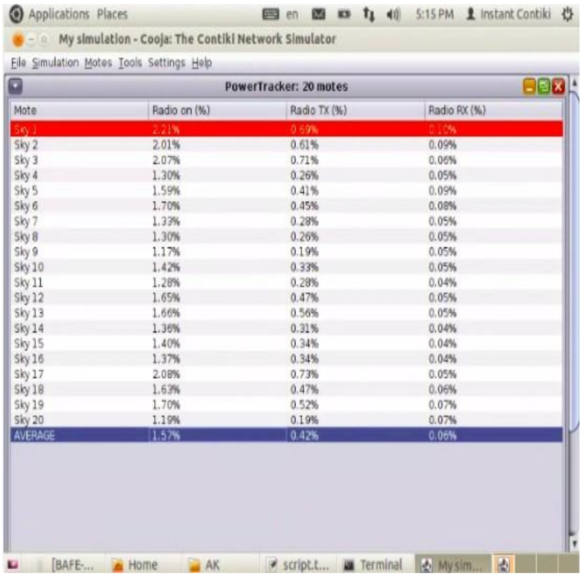
My simulation - Cooja: The Contiki Network Simulator

PowerTracker: 15 motes

Mote	Radio on (%)	Radio TX (%)	Radio RX (%)
Sky 1	2.78%	1.09%	0.04%
Sky 2	2.49%	0.94%	0.05%
Sky 3	2.72%	1.04%	0.07%
Sky 4	2.72%	1.09%	0.04%
Sky 5	2.82%	1.09%	0.06%
Sky 6	2.55%	0.94%	0.04%
Sky 7	2.71%	0.94%	0.10%
Sky 8	2.52%	0.94%	0.05%
Sky 9	2.57%	0.94%	0.07%
Sky 10	2.56%	0.94%	0.06%
Sky 11	2.57%	0.93%	0.07%
Sky 12	2.32%	0.86%	0.06%
Sky 13	2.84%	1.09%	0.05%
Sky 14	2.88%	1.09%	0.07%
Sky 15	2.90%	1.09%	0.09%
AVERAGE	2.66%	1.00%	0.06%

b) For 20 nodes:

With CTP:

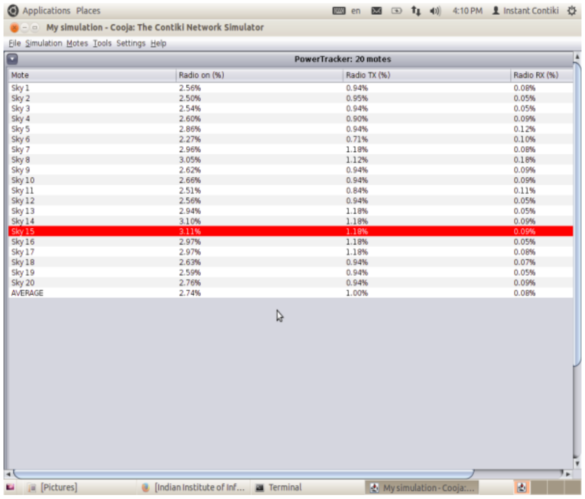


My simulation - Cooja: The Contiki Network Simulator

PowerTracker: 20 motes

Mote	Radio on (%)	Radio TX (%)	Radio RX (%)
Sky 1	2.21%	0.69%	0.10%
Sky 2	2.01%	0.61%	0.09%
Sky 3	2.07%	0.71%	0.06%
Sky 4	1.30%	0.26%	0.05%
Sky 5	1.59%	0.41%	0.09%
Sky 6	1.70%	0.45%	0.08%
Sky 7	1.33%	0.28%	0.05%
Sky 8	1.30%	0.26%	0.05%
Sky 9	1.17%	0.19%	0.05%
Sky 10	1.42%	0.33%	0.05%
Sky 11	1.28%	0.28%	0.04%
Sky 12	1.65%	0.47%	0.05%
Sky 13	1.66%	0.56%	0.05%
Sky 14	1.36%	0.31%	0.04%
Sky 15	1.40%	0.34%	0.04%
Sky 16	1.37%	0.34%	0.04%
Sky 17	2.08%	0.73%	0.05%
Sky 18	1.63%	0.47%	0.06%
Sky 19	1.70%	0.52%	0.07%
Sky 20	1.19%	0.19%	0.07%
AVERAGE	1.57%	0.42%	0.06%

Without CTP:


Applications Places
en
4:10 PM
Instant Contiki
File Simulation Motes Tools Settings Help
[Pictures]
[Indian Institute of Inf...
Terminal
My simulation - Cooja:...

My simulation - Cooja: The Contiki Network Simulator

PowerTracker: 20 motes

Mote	Radio on (%)	Radio TX (%)	Radio RX (%)
Sky 1	2.56%	0.94%	0.08%
Sky 2	2.50%	0.95%	0.05%
Sky 3	2.54%	0.94%	0.05%
Sky 4	2.60%	0.90%	0.09%
Sky 5	2.86%	0.94%	0.12%
Sky 6	2.27%	0.71%	0.10%
Sky 7	2.96%	1.18%	0.08%
Sky 8	3.05%	1.12%	0.18%
Sky 9	2.62%	0.94%	0.09%
Sky 10	2.66%	0.94%	0.09%
Sky 11	2.51%	0.84%	0.11%
Sky 12	2.56%	0.94%	0.05%
Sky 13	2.94%	1.18%	0.05%
Sky 14	3.10%	1.18%	0.09%
Sky 15	3.11%	1.18%	0.09%
Sky 16	2.97%	1.18%	0.05%
Sky 17	2.97%	1.18%	0.08%
Sky 18	2.63%	0.94%	0.07%
Sky 19	2.59%	0.94%	0.05%
Sky 20	2.76%	0.94%	0.09%
AVERAGE	2.74%	1.00%	0.08%

We can now conclude that we have successfully applied collection tree protocol and reduced the power consumption by a good margin.