Sensor Network as Internet Gateway

Network Sensor Lab - SS11

Oana Comanici, Anupam Ashish, Ignacio Avellino



# Introduction

The starting point of this project is the code provided by the ComSys chair on the project \_\_\_\_ by \_\_\_\_. This project can be found at: <https://github.com/AndreaCrotti/Network-mote>

This code solves the problem of transferring Internet packages over a network of telosb sensors by dividing IPv6 packages into smaller ones; it used blip as an implementation of IPv6 on TinyOS.

Using this staring point, the objective of this project is to enable a sensor network as an Internet gateway. Given a network of sensors where one of the sensors is connected to a computer with Internet access, another computer without Internet access but with one of this motes connected to it, should be able to access the Internet. In order for this to be possible, the following functionalities have to be added to the existing Network as an Internet gateway “project: routing, roaming and multi client support.

In the next section it will be discussed how each of these features is implemented

# Features added

## Routing

The purpose of having routing is that a client mote of the testbed doesn’t need to reach directly the mote connected to the Internet (gateway), but it can reach it through other motes also on the testbed.

In order to implement this routing algorithm, routing tables will be used. This list represents the reachable nodes in the testbed. Each entry of the routing contains:

* node id
* node address
* metric
* id of next hop that reaches this node
* address of next hop that reaches this node
* timeout

To broadcast its existence, each mote sends a beacon packet periodically containing its id only. When this beacon is received by another mote it checks in its routing table if an entry for this mote exists. If the mote doesn’t exist a new entry is added with metric zero and the routing table is forwarded to all the neighbors (nodes with metric zero). Additionally, the routing table is sent to the neighbors periodically even if there are no changes.

Using this data, a node knows to which other node a packet must be forwarded in order to reach a certain destination. The timeout on each entry of the routing table is used to delete an entry after a certain time has passed, reflecting changes in the topology of the sensor network.

An example of this mechanism is provided in Figure 1. Here four nodes are shown with their respective routing tables. At this moment mote 1 sends its beacon, which is received by mote 2. Since there was no entry for mote 1 on this routing table, an entry is added and the new routing table is forwarded to all other neighbor nodes (1, 3 and 4). In Figure 2 it is shown how the routing tables of all motes are updated after mote 2 sends its routing table.

Mote 1

Mote 2

Mote 3

Mote 4

Routing table

{2, ADDR2, 0, null, null, 10}

Routing table

{3, ADDR3, 0, null, null, 10}

{4, ADDR4, 0, null, null, 10}

Routing table

{2, ADDR2, 0, null, null, 10}

Routing table

{2, ADDR2, 0, null, null, 10}

Figure : Routing tables before Mote 1 sends its beacon. Rounting tables contain entries as follows: {node ID, node address, metric, next hop ID, next hop Address, timeout} The lines represent which motes can be reached by a Mote.

Mote 1

Mote 2

Mote 3

Mote 4

Routing table

{2, ADDR2, 0, null, null, 10}

{3, ADDR3, 1, 2, ADDR2, 10}

{4, ADDR4, 1, 2, ADDR2, 10}

Routing table

{1, ADDR1, 0, null, null, 10}

{3, ADDR3, 0, null, null, 10}

{4, ADDR4, 0, null, null, 10}

Routing table

{2, ADDR2, 0, null, null, 10}

{1, ADDR1, 1, 2, ADDR2, 10}

{4, ADDR4, 1, 2, ADDR2, 10}

Routing table

{2, ADDR2, 0, null, null, 10}

{3, ADDR3, 1, 2, ADDR2, 10}

{1, ADDR1, 1, 2, ADDR2, 10}

Figure 2: Routing tables after Mote 1 sends its beacon, Mote 2 receives it, updates its routing table and forwards it. Rounting tables contain entries as follows: {node ID, node address, metric, next hop ID, next hop Address, timeout} The lines represent which motes can be reached by a Mote.