

Obtaining Node Position and Energy Dynamically in NS2



To read this post online visit: <http://wp.me/piGUb-Ic>

Happy to meet you with another interesting task in NS2!

This time we will “dynamically” calculate a nodes position and remaining energy. This post is based on the valuable tips mentioned <here> by Elmurod Talipov.

Now to the task...

I will be modifying the sample script **simple-wireless.tcl** found at ~/ns-allinone-2.34/ns-2.34/tcl/ex and all my tasks in ns2 would be based on AODV protocol.

1. In **simple-wireless.tcl**, change the routing protocol to AODV, enable Energy model and set the initial energy value.

```
set val(rp)          AODV                ;# routing protocol
set val(energymodel)  EnergyModel         ;# Energy Model
set val(initialenergy) 100                ;# value
```

2. In the node configuration set the EnergyModel, Initial Energy value, power spent in receiving mode, transmit mode, idle mode and sleep mode.

visit <http://getch.wordpress.com>

<http://getch.wordpress.com>

```
# configure node
```

```
$ns_ node-config -adhocRouting $val(rp) \  
                  -llType $val(ll) \  
                  -macType $val(mac) \  
                  -ifqType $val(ifq) \  
                  -ifqLen $val(ifqlen) \  
                  -antType $val(ant) \  
                  -propType $val(prop) \  
                  -phyType $val(netif) \  
                  -channelType $val(chan) \  
                  -topoInstance $topo \  
                  -agentTrace ON \  
                  -routerTrace ON \  
                  -macTrace OFF \  
                  -movementTrace OFF \  
                  -energyModel $val(energymodel) \  
                  -initialEnergy $val(initialenergy) \  
                  -rxPower 35.28e-3 \  
                  -txPower 31.32e-3 \  
                  -idlePower 712e-6 \  
                  -sleepPower 144e-9
```

3. Now we will move to the backend code(c++). Our edits would be performed only in aadv.cc and aadv.h.

4. In aadv.h, first include the following header file that contains the procedures/functions needed to access node position, energy and several other functions.

```
#include <mobilenode.h>
```

You will find class AODV, something like the one shown below.

```
/*  
    The Routing Agent  
*/  
class AODV: public Agent {  
  
    /*  
     * make some friends first  
     */  
};
```

visit <http://getch.wordpress.com>

5. In that, declare the following variables in “protected” scope.

```
271
272     /*
273     * History management
274     */
275
276     double          PerHopTime(aodv_rt_entry *rt);
277
278
279     nsaddr_t         index;                // IP Address of this node
280     u_int32_t        seqno;               // Sequence Number
281     int              bid;                 // Broadcast ID
282
283 /* declared by Manoj for mobilenode.h related work*/
284     double xpos;
285     double ypos;
286     double zpos;
287     double iEnergy;
288     MobileNode *iNode;
289
290 /*-----*/
```

With that, no more work to be done in aodv.h. What we have actually done here is to declare the variables that we will using in aodv.cc to record a nodes position and energy.

6. In aodv.cc, we will initialize the variables that we have just declared in aodv.h. This we do in the AODV “constructor”(invoked only once during node creation).

```
/*
 * Constructor
 */

AODV::AODV(nsaddr_t id) : Agent(PT_AODV),
                          btimer(this), htimer(this), ntimer(this),
                          rtimer(this), lrtimer(this), rqueue() {

    index = id;
    seqno = 2;
    bid = 1;
/*manoj: mobilenode.h work*/
    xpos = 0.0;
    ypos = 0.0;
    zpos = 0.0;
    MobileNode *iNode;
    iEnergy = 0.0;
/*-----*/
```

<http://getch.wordpress.com>

7. Next, we include the code to access the functions in mobilenode.h that fetch the nodes position and energy.

```
1005 Packet Transmission Routines
1006 */
1007
1008 void
1009 AODV::forward(aodv_rt_entry *rt, Packet *p, double delay) {
1010 struct hdr_cmh *ch = HDR_CMH(p);
1011 struct hdr_ip *ih = HDR_IP(p);
1012 /***Code to print Node position and Energy: Manoj
1013 iNode = (MobileNode *) (Node::get_node_by_address(index));
1014 xpos = iNode->X();
1015 ypos = iNode->Y();
1016 iEnergy = iNode->energy_model()->energy();
1017 printf("at Time (%.6f), Position of %d is X: %.4f and Y : %.4f \n", CURRENT_TIME, index, xpos, ypos);
1018 printf("at Time (%.6f): Updated Energy for Node %d is Energy %.4f \n", CURRENT_TIME, index, iEnergy);
1019 *****/
1020
1021 if(ih->tll_ == 0) {
1022
1023 #ifdef DEBUG
1024 fprintf(stderr, "%s: calling drop()\n", __PRETTY_FUNCTION__);
1025 #endif // DEBUG

```

I have included that code **AODV::forward()** function, so that the nodes position and energy are printed each time it forwards a packet. Where and When you want them to be printed is your choice. Alternatively, you may have the code include in the function of your choice and have the output redirected to a file.

8. Thats all! All we have to do now is to run simple-wireless.tcl

\$ns simple-wireless.tcl

and the output would be something like this:

```
root@user-desktop: ~
File Edit View Search Terminal Help
at Time (149.644802), Position of 0 is X: 5.0000 and Y : 15.0000
U (149.644802): UPDATE ENERGY, for Node 0, Energy 95.4630
at Time (149.646727), Position of 0 is X: 5.0000 and Y : 15.0000
U (149.646727): UPDATE ENERGY, for Node 0, Energy 95.4630
at Time (149.656812), Position of 1 is X: 50.0000 and Y : 50.0000
U (149.656812): UPDATE ENERGY, for Node 1, Energy 95.0719
at Time (149.666736), Position of 1 is X: 50.0000 and Y : 50.0000
U (149.666736): UPDATE ENERGY, for Node 1, Energy 95.0715
at Time (149.668681), Position of 0 is X: 5.0000 and Y : 15.0000
U (149.668681): UPDATE ENERGY, for Node 0, Energy 95.4623
at Time (149.678746), Position of 1 is X: 50.0000 and Y : 50.0000
U (149.678746): UPDATE ENERGY, for Node 1, Energy 95.0711
at Time (149.680750), Position of 0 is X: 5.0000 and Y : 15.0000
U (149.680750): UPDATE ENERGY, for Node 0, Energy 95.4619
at Time (149.690795), Position of 1 is X: 50.0000 and Y : 50.0000
U (149.690795): UPDATE ENERGY, for Node 1, Energy 95.0707
at Time (149.700679), Position of 1 is X: 50.0000 and Y : 50.0000
U (149.700679): UPDATE ENERGY, for Node 1, Energy 95.0704
at Time (149.703650), Position of 0 is X: 5.0000 and Y : 15.0000
U (149.703650): UPDATE ENERGY, for Node 0, Energy 95.4612
```

<http://getch.wordpress.com>

If you find any discrepancies in this post, please feel free to drop me a mail at micman.manoj@hotmail.com. By doing so, You would be saving the valuable time of many researchers.

A Personal Appeal From This Blogger

There are more than hundred IEEE research papers that have used NS2 for their research work, but the codes available online dont equal the proposed findings. This is because of the over conservative nature of the researchers. If only they could open up their minds and provide guidance to beginner researchers, the quality of simulation that is taking place would be very nice. So I humbly appeal that researchers "open out their code" or provide an outline on "how to get tasks done" in NS2. Blogging and appropriately tagging on NS2 tasks would answer hundreds of queries that remain unanswered in the ns forums. So help us "not to reinvent the wheel".

Start Blogging! Start Sharing!

-- ManojKumar.A

visit <http://getch.wordpress.com>

This article is licensed under Creative Commons License by

ManojKumar.A

Read the following License information before distributing this article.



Attribution 3.0 Unported (CC BY 3.0)

You are free:



to Share — to copy, distribute and transmit the work



to Remix — to adapt the work



Under the following conditions:



Attribution — You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).

With the understanding that:

Waiver — Any of the above conditions can be **waived** if you get permission from the copyright holder.

Public Domain — Where the work or any of its elements is in the **public domain** under applicable law, that status is in no way affected by the license.

Other Rights — In no way are any of the following rights affected by the license:

- Your fair dealing or **fair use** rights, or other applicable copyright exceptions and limitations;
- The author's **moral** rights;
- Rights other persons may have either in the work itself or in how the work is used, such as **publicity** or privacy rights.

Notice — For any reuse or distribution, you must make clear to others the license terms of this work. The best way to do this is with a link to this web page.