

NS2 Notebook: How to Get Neighbors of a Mobile Node in Wireless Simulation

[Back to NS2 Menu](#)

1. Get the number of nodes in the network

```
#include <god.h>

God::instance()->nodes();
```

2. In order to use God object in C++ code, in your simulation script, you should have

```
# Create God
set god_ [create-god $opt(nn)]

for {set i 0} {$i < $opt(nn)} {incr i} {
    set node_($i) [$ns_ node]
    $node_($i) random-motion 0 ;# disable random motion
    $god_ new_node $node_($i)
}
```

3. In order to check that if node j is node i's neighbor, use the following function.

```
God::instance()->IsNeighbor(i , j);
```

4. In order to use IsNeighbor function in C++ code, either add EnergyModel in simulation script, (please refer to NS Manual Chapter 19 for detail)

```
$ns_ node-config -energyModel $energymodel \
-rxPower $p_rx \
-txPower $p_tx \
```

```
-initialEnergy $initialenergy \
```

or change the `IsNeighbor` function in `mobile/god.cc` (disable energy checking)

```
bool God::IsNeighbor(int i, int j)
{
    assert(i<num_nodes && j<num_nodes);

    /*
    if (mb_node[i]->energy_model()->node_on() == false ||
    mb_node[j]->energy_model()->node_on() == false ||
    mb_node[i]->energy_model()->energy() <= 0.0 ||
    mb_node[j]->energy_model()->energy() <= 0.0 ) {
        return false;
    }
    */

    vector a(mb_node[i]->X(), mb_node[i]->Y(), mb_node[i]->Z());
    vector b(mb_node[j]->X(), mb_node[j]->Y(), mb_node[j]->Z());
    vector d = a - b;

    if (d.length() < RANGE)
        return true;
    else
        return false;
}
```