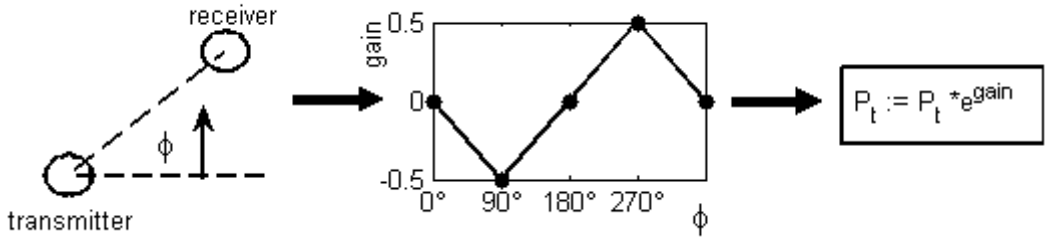
Asymmetric Propagation Proxy

NS-2 supports only symmetric propagation models like Shadowing. It means that the link quality between two nodes is always equal. For example if A hears B with propability of p, B will hear A with the same propability of p. For supporting asymmetric propagation we develop a new propagation model (source). Unlike the other models it is just a proxy, that will connected with a normal propagation model. The proxy changes the transmission power of the transmitter in dependency of the angle (between transmitter and receiver). You can configure the behaviour of each node by setting the gain (or damping) in dependence of the angle.

The following diagram shows the algorithm of the Asymmetric Propagation Proxy:



First the proxy calculates the angle between transmitter and receiver. For each node a profile exists (if not, it will be generated randomly) in which for a number of angles a gain values are given. The angle is linear mapped to a gain value. This gain is used to modify the transmission power (an exponential function). At the end the common routines of the connected propagation model (for example shadowing) are invoked.

Usage

Normally you define, which TYPE of propagation model you want to use and NS-2 will create it for you. This won't work with the Propagation Proxy Model. Here you have to create manually a instance of the Propagation Proxy Model and configure it. Afterwards you can instruct NS-2 to take the model for all nodes. The following script shows the procedure:

```
# variables of the model
Propagation/APProxy set minRandomGain_ -0.5;
Propagation/APProxy set maxRandomGain_ 0.5;
Propagation/APProxy set maxGain_ 1.0;
Propagation/APProxy set angleNumber_ 4;

# create propagation proxy instance
set opt(propInstance) [new Propagation/APProxy]
# configurate propagation proxy instance
$opt(propInstance) propagation [new Propagation/TheModelYouWantToUse]

...
# the script for configurating the nodes
if { [info exists opt(propInstance)] } {
        # so here you use an existent instance of a propagation model
        $ns_ node-config -propInstance $opt(propInstance)
} else {
        # this is the normal case (for instance when using the shadowing model)
        $ns_ node-config -propType $opt(prop)
}

$ns_ node-config \
        -adhocRouting $opt(routing) \
        -llType $opt(linkLayer) \
        -macType $opt(macLayer) \
```

```
        -ifqType $opt(ifq) \
        -ifqLen $opt(lenIfq) \
        -antType $opt(ant) \
        -phyType $opt(netif) \
        -topoInstance $topo_ \
        -channel $chan_ \
        -agentTrace OFF \
        -routerTrace OFF \
        -macTrace OFF \
        -movementTrace OFF
...
```

The model consists of 4 Tcl-binded variable. `angleNumber_` defines the number of points (angles), for which you want to set a gain value. All other gains will be linear approximated. The angles are equally distributed between 0 and 360 degree. For example if `angleNumber_` = 4 you have the angles 0,90,180,270 degree. The variables `minRandomGain_` and `maxRandomGain_` are used for creating random gain values for the nodes, if you don't manually define them. The resulting gain value will be uniformly distributed between the `minRandomGain_` and `maxRandomGain_`. The last variable `maxGain_` is used for calculating the maximum reachable node. So never use Gain values greater than `maxGain_`.

For defining the behaviour (in this context called **profile**) of a specific node you can create the following script:

```
# profile <node id>  <gain for 0 deg>  <gain for 90 deg>  <gain for 180 deg>  <gain for 270 deg>
$opt(propInstance) profile 0 0 -0.5 0 0.5
```

You have to use the exact number of gain values you defined with `angleNumber_`. It's also important to know, that the node (for which you set the profile) must not exists. You only define a entry in a look up table (STL map), which is used dynamically at runtime.

Examples

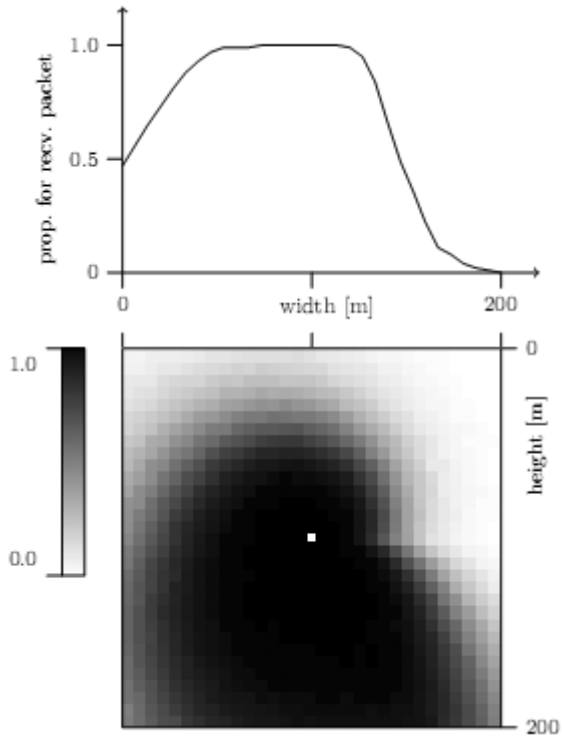
Here some examples for the Asymmetric Propagation Proxy. The script for generating the plots can be found under Visualisation of NS-2 propagation models).

Using the shadowing model (manual definition of profile):

```
...
Propagation/APProxy set minRandomGain_ 0;
Propagation/APProxy set maxRandomGain_ 0;
Propagation/APProxy set maxGain_ 10.0;
Propagation/APProxy set angleNumber_ 6;

# set propagation model
set opt(propInstance) [new Propagation/APProxy]
$opt(propInstance) propagation [new Propagation/Shadowing]

$opt(propInstance) profile [expr [expr $gridX*$gridY-1] /2] -1.0  -0.5 0 0.5 1.5 2.0
...
```

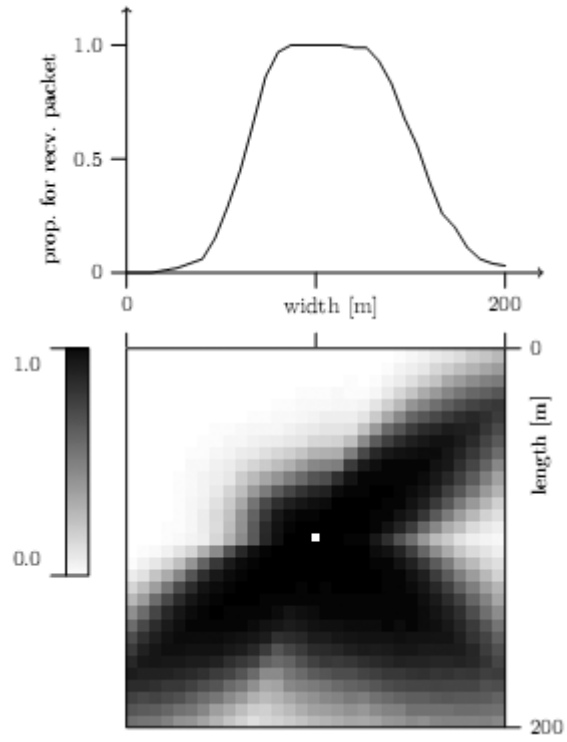


Using the shadowing model (random definition of profile):

```
Propagation/APProxy set minRandomGain_ -2.0;
Propagation/APProxy set maxRandomGain_ 2.0;
Propagation/APProxy set maxGain_ 10.0;
Propagation/APProxy set angleNumber_ 10;

$defaultRNG seed 1000;

# set propagation model
set opt(propInstance) [new Propagation/APProxy]
$opt(propInstance) propagation [new Propagation/Shadowing]
```



Using the Two-ray ground reflection model (manual definition of profile):

```
Propagation/APProxy set minRandomGain_ -2.0;
Propagation/APProxy set maxRandomGain_ 2.0;
```

```
Propagation/APProxy set maxGain_ 10.0;
Propagation/APProxy set angleNumber_ 10;

# set propagation model
set opt(propInstance) [new Propagation/APProxy]
$opt(propInstance) propagation [new Propagation/TwoRayGround]

$opt(propInstance) profile [expr [expr $gridX*$gridY-1] /2]  1.0 -1.0  1.0 -1.0 \
                                                             1.0 -1.0  1.0 -1.0  1.0 -1.0
```