Middleware Lab 05 :
Radiowaves propagation model for more realistic simulations

Guillaume-Jean Herbiet
<guillaume.herbiet@uni.lu>

December 5, 2008

Abstract

During the previous labs, you implemented a complete protocol model for ad hoc instant messenging in OMNeT++ , an event-based discrete network simulator, including a realistic mobility model and instrumentation for statisctics collection.

Using this model, you ran different simulation campaigns in order to assess the performance of the protocol you implemented under different conditions.

In this lab, you will bring your simulation environment closer to reality by implementing a more realistic propagation model that takes into account the presence of obstacles in the simulation field.

1 Propagation model description

In the previous lab, we designed and implemented the **Hotspot mobility** model in which hosts are moving from spot to spot. The latter represent some points of interests (like shops or meeting rooms) and are materialized by cicles of varrying radius inside the simulation field.

As those spots are respesenting some closed area in which the users are likely to enter, stop for a while, and then leave, they should impact the transmission of data over the air. For instance, a node inside a spot should receive with reduced quality a transmission from a node outside the spot; just like a user inside a shop will hardly receive signal from another user located outside the building.

Therefore, we propose a propagation model based on two components, that will reduce the received power of a given message sent over the air:

- **path loss**, that is reduction of the received power due to signal fading with distance. This effect is already modeled by OMNeT++ using a model called **free space** propagation;
- **obstruction of objects**, that is the reduction of the received power due to the fact that nodes may be inside or outside the same or different spots, and that some spots inbetween the nodes may impact propagation of information.

1.1 Free space path loss

Path loss (or path attenuation) is the reduction in power density (attenuation) of an electromagnetic wave as it propagates through space. Path loss is a major component in the analysis and design of the link budget of a telecommunication system.

In free space, all electromagnetic waves (radio, light, X-rays, etc) obey the inverse-square law which states that the power density of an electromagnetic wave is proportional to the inverse of the square of the distance r from the source or:

$$\rho_P \propto \frac{1}{r^2}$$

Doubling the distance from a transmitter means that the power density of the radiated wave at that new location is reduced to one-quarter of its previous value.

1.2 The obstruction matrix

The following figures summarizes the different possible cases of obstruction by spots on the simulation field.

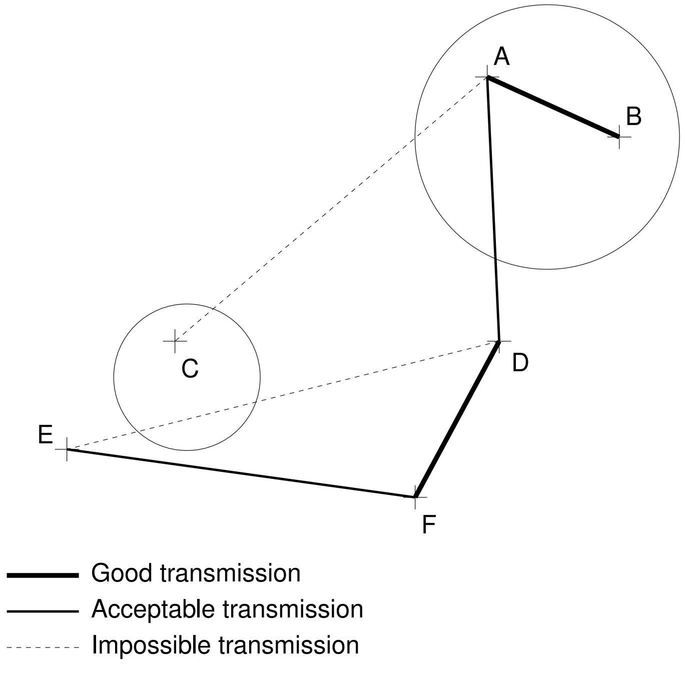

Figure 1: Example of propagation cases

- A and B are inside the same spot, only path loss applies;
- D and F are outside any spot and no obstruction exists, only path loss applies (same for E and F with a bigger distance, so received power is lower);
- A is inside a spot, D outside, the received power should be impacted by the fact that it has to traverse the spot's walls;

- A and C are inside different spots, so we will consider that no communication can occur between them;
- D and E are obstructed by a spot, so we will consider that no communication can occur between them;

Those cases can be summarized in the following obstruction matrix:

		Node j	
		Inside	*Outside*
Node i	*Inside*	δ_{ij}	$o_{ij} * c_s$
	Outside	$o_{ij} * c_s$	o_{ij}

Table 1: The obstruction matrix

where:

- δ_{ij} is 1 if i and j are inside the same spot, 0 otherwise;
- o_{ij} is 1 if i and j are not obstructed by any spot (like E and F in figure 1), 0 otherwise (like D and E in figure 1);
- c_s is the coefficient of the spot s, which is in]0;1[, and represent the signal attenuation due to the spots walls.

As each spot can have walls of different thickness, build with different material, c_s is a spot parameter. It will be read from the `.spots` file.

1.3 The importance of received power

Computing the received power of a packet is very important in order to decide whether this packet is just **noise** (i.e. a reception occured, but the data received is not understandable), or has been received but with too many erroneous bits, or correctly received.

In network simulation, where physical layer is modeled at the packet level, the following principles are applied:

- from the received power of a signal is computed the signal to noise ratio (the power of what we are interested in over the cumulated power of all the things we hear);
- from this ratio is decided a percentage of erroneous bits on the packet;
- then those erroneous bits are mapped on the actual bits of the packet.

Each physical receiver has an error correction mechanism, only able to recover a limited percentage of errors, and on some places of the packet only. If the error correction mechanism can recover the packet, this one can be processed (i.e. passed to the upper layer), otherwise it is discarded.

2 Propagation model implementation

In OMNeT++ with Mobility Framework the physical interface is modeled as a compound module called `nic` which is composed of:

- a **MAC layer module**: in this series of lab we use the `Mac80211` module, which models IEEE802.11 (wi-fi);
- a **decider module** which, based on the signal-to-noise ratio (SNR) of the received packets decides to drop them or not;
- a **SNR module**, which computes the signal-to-noise ratio of the received packets using a propagation model.

2.1 Objectives

In this lab, you will subclass the `SnrEval80211` module (used for wi-fi modeling) to take into account the presence of spots and their characteristics on the radiowaves propagation.

Your implementation should match the description of the propagation model presented in section 1.

2.2 Hotspots characteristics

In the previous lab, the different hotspots were defined by the position of their center and their radius. A `.spots` file, read by the `HotSpotsChannelControl` module upon initialization, was used to generate `HotSpot` objects with the associated characteristics.

For sake of propagation modeling, each spot s shall have an associated coefficient c_s representing the signal attenuation due to the spots walls.

Therefore, based on the new `.spots` file definition you can find in the `/home/user/Development/templates/lab05` directory, apply the following modifications to take this new coefficient into account:

- in the `HotSpot` class definition, add a class attribute of type `double`, called `coeff`;
- in the `HotSpotsChannelControl` module, modify the `readFile()` and `parseLine()` functions, so the coefficent value can be read and stored for each spot;

2.3 Propagation model

To implement this new propagation model, subclass the `SnrEval80211` module. You can use the `like` attribute (as you did for the `Stats*` modules on lab 3) so you don't have to create a new `Ned` file. Don't forget to modify the `StatsNic.ned`, `Host.ned` and `Network.ned` files in consequence.

You will find templates for the `HotSpotsSnrEval80211.h` and `HotSpotsSnrEval80211.cc` files in the `/home/user/Development/templates/lab05` directory.

2.3.1 Path loss

Simple path loss computation (only based on distance) is already computed in a parent class of our module (the `SnrEval` module). You simply have to call the `calcPathLoss()` function of this parent class. It is a good idea to look at how this computation is performed (particularly how the sender's and receiver's positions are retrieved) for your own following calculations.

2.3.2 Hotspots impact

The computation for the hotspots impact on received power will be performed in the `calcHotSpotsImpact()` function.

To complete your computation, you will need the following information:

- Are the nodes inside a spot? Is this the same? Note that in previous lab our mobility model required the same kind of information, so you should already know how to check if a given node is inside a given spot...
- Is there an obstruction (i.e. are the node hidden by one or more spots)? A node is hidden from another by a spot, if the spot is between the two nodes (in terms of direction and distance).

To test this latter point, it is required to do some geometry. In the following figure, we will take the example of node A. The two nodes B and C are hidden from A by a different spot. This can be translated as the direction from A to B (resp. C) being in the range of angles covered by the spot S_1 (resp. S_2) and the distance between the center of S_1 (resp. S_2) and A being smaller than the distance between B (resp. C) and A.

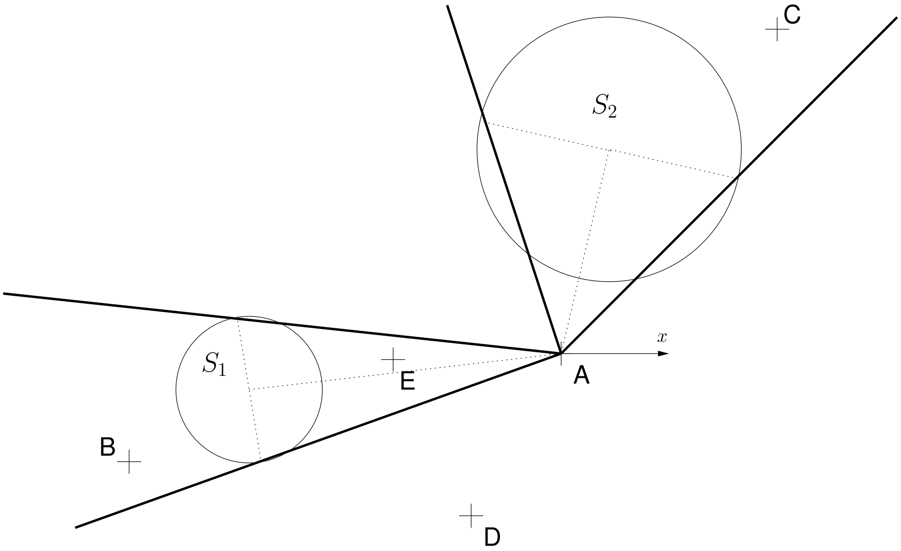

Figure 2: Obstructions geometry

On the contrary, D is not hidden from A, as the direction from A to D is covered neither by S_1 nor S_2. E is not hidden from A as the distance between A and E is smaller than the distance between A and the center of S_2.

On the previous lab, you used some utility functions, relying on trigonometry to compute the angles covered by a spot from a given point (this helped you determine the allowed angles for a node to move towards a spot). By using some variation of these functions, you can easily determine whether or not two nodes are obstructed by a spot and if they are hidden from each other.

2.4 Global received power attenuation

As explained earlier, the global received power attenuation is the combination of the two previously-described effects (path loss and obstruction). Therefore, you will simply have to multiply the received power computed in the `calcPathLoss()` function by the obstruction coefficient computed in `calcHotSpotsImpact()`. This operation is to be done in the `calcRcvdPower()` function.

3 Simulations with the propagation model

Once you have ensured that your propagation model performs correctly, you can launch a real simulation campaign of at least 20 runs using the same configuration files you used for the simulation campaign at the end of last lab.

Analyze and consolidate your results as you did in the previous lab to compair the obtained results and assess the impact of a close-to-reality propagation model on the protocol performance.

Next time:

Now that you have completed the implementation of the propagation model, you have achieved a fairly high-fidelity simulation of a complete ad hoc network stack, from application layer to physical level. You should also have all the tools required to assess the network behavior and its protocol performance.

In the next lab, last of this series, you will have to design and implement several optimizations in order to improve the ad hoc instant messenging protocol robustness.