This is a brief help for the use of our simulator. However, you may need to read our paper first to better understand the main assumptions that we make. You can find this paper online "**Network Coding for Wireless Mesh Networks: A Case Study**".

In this simulator, we evaluate four cooperation strategies between nodes:

1- Blind-forwarding without network coding
2- Blind-forwarding with network coding
3- Selective-forwarding without network coding
4- Selective -forwarding with network coding

The principle of network coding is to encode different packets together. We implement this feature using the Galois field. The main characteristic of the Galois field is that the result of any operation remains within this field. A complete C++ package of Galois field can be found at this site: http://www.partow.net/projects/galois/

This script takes 9 arguments as inputs:

1- c ------------- Number of chunks the file is split into.

2- r ------------- Transmission radius of nodes. We assume a fixed and homogeneous radius for all nodes.

3- x ------------- The length of the area S where the nodes are deployed.

4- y ------------- The width of the area S where the nodes are deployed.

5- n ------------- The number of sources that hold the entire file at the beginning of the simulation. We allow up to four sources but it can be easily extended to support a larger number.

6- s ------------- The placement of the sources in the network. We consider two placements *Random* and *Extreme*. s='r' means that the source is placed at random. s='e' means that the source is placed at one of the corners of the area S.

7- l ------------- The loss rate in the network. We assume a very simplified model for losses; a node receives chunks from its neighbors with a probability of (100-l)%.

8- b ------------- The run number. This parameter allows to run a same simulation many times.

9- d ------------- The directory where the results will be stored.

As a result of this simulator, we store in Matlab files the following metrics:

Service time: Time needed to distribute the file to all nodes.
Completed nodes: Number of nodes that complete against time.
Useless chunks: Number of useless chunks received by all nods during the entire simulation.

Transmitted chunks: Number of emitted chunks by all nodes during the entire simulation

Chunks evolution: Distribution of chunks in the system. We compute at any moment the number of nodes that hold k chunks, with $0 < k < c$.

The script can be run as follows:

```
./script -c < > -r < > -x < > -y < > -d < > - n < > -s < > -l < > -b
```

We recommend you to respect this order when giving the arguments. Also do not forget to leave a space between each argument and its value, e.g., -c 3 AND NOT –c3

One final remark: The number of nodes is not given by the user. Instead it is computed as a function of r, x, and y using the following formula: $N = (10*x*y)/3.14*r^2$.