Simulating Random Geographical Routing — An Assignment in the Nicta Short Course in Network Simulation

Olivier Mehani
National ICT Australia, and
University of New South Wales, Australia, and
Mines ParisTech, CAOR — Centre de Robotique, Mathématiques et Systèmes, France,
olivier.mehani@nicta.com.au

***Abstract*—We present simulation results of a random geographical routing model in a lattice of vertically and horizontally linked nodes. The effects of the hop-limit and bias on the number of correctly transfered packets are shown. We then expand the model by adding a failure model and compare the new results. We also experiment with various sizes of the lattices, and diagonal links.**

***Index Terms*—geographical routing, random routing, simulation, short course, assignment**

I. Introduction

Routing packets through a network using geographical position information may be an interesting approach. However, to avoid sending all packets to a possible dead end, close but not connected to the destination, it is necessary to introduce a bit of biased randomness into the routing.

Using the discrete event simulator written for [1], we evaluate the efficiency of such a routing scheme, for a various key parameters.

In the following sections, we first study the influence of the maximum time-to-live and routing bias on the delivery success rate in the default model. We also explore the influence of varying the size of the lattice to verify the scalability of the proposed scheme. In section III, we then extend the model to simulate random node failures, and diagonal links. We then summarize our findings in section IV.

II. Random Geographical Routing with Bias

In order to keep the desired property of geographical routing, it is unreasonnable to select the next hop completely randomly. A bias in the random selection is necessary to, overall, get a given packet closer to its destination on each hop. In this section, the biased selection algorithm is first described. Simulation results obtained varying the maximum allowed hop-count for a packet, the routing bias, and the lattice size are then presented.

A. Biased Selection of the Next Hop

The algorithm used to select the next hop is a simple roulette-wheel mechanism. Each neighbor is given a weight depending on whether it is closer to the destination than the current node. Algorithm 1 is used to compute said weights.

```
Input: List N of neighbors of size s
Output: List W of weigths which sum is Sw
for i ← 0 to s − 1 do
    if neighbor N[i] is available then
        if N[i] is closer to the destination in x then
            W[i] ⟵ bias;
        else
            W[i] ⟵ 1;
        end
        if N[i] is closer to the destination in y then
            W[i] ⟵ W[i] + bias;
        else
            W[i] ⟵ W[i] + 1;
        end
        Sw ← Sw + W[i];
    end
end
```

Algorithm 1: Attribution of the weights for the roulette-wheel selection.

The roulette-wheel selection algorithm by itself draws a random number and finds the corresponding entry on the "wheel" by comparing the cumulative weights to the number, as per Algorithm 2.

```
Input: List N of neighbors of size s
Input: List W of weigths which sum is Sw
Output: The selected neighbor n
r ⟵ random number between 0 and Sw;
i ⟵ 0;
cumul ⟵ 0;
while cumul < r and i < s do
    i ⟵ i + 1;
    cumul ⟵ cumul + W[i];
end
n ⟵ N[i − 1];
```

Algorithm 2: Roulette-wheel selection.

TABLE I
Default parameters of the random geographical routing simulation.

number of packets	1,000,000
hop-limit	10
routing bias	2
lattice size	4×4

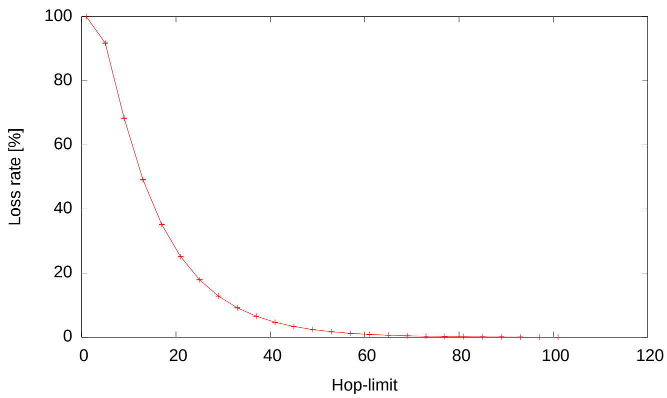

Fig. 1. Packet losses depending on the hop-limit.

B. Simulation Results

The random geographical routing model, with the roulette-wheel selection mechanism from the previous section, has been run for a relevant subsets of the available parameters. Simulations for a given set of parameters were run a hundred times each, varying the seed of the random number generator.

In all the following plots, the mean of the realisations are reported. The error bars show the 95% confidence interval. If not specified or explicitely set, the default parameters of the model are given in TABLE I.

For simplicity, a scenario with single source and destination nodes are considered. The source will always be located at $(0, 1)$ on the lattice while the destination will be at $(s-2, s-1)$, s being the size of the lattice.

C. Hop Limit

Fig. 1 shows the number of lost packets depending on the maximum authorised hop count. Even in a small 4×4 grid (that is 16 nodes), it is not possible to achieve an decent success rate ($> 95\%$) with a TTL lower than 40.

D. Routing Bias

The influence of the bias on the packet loss rate is presented on Fig. 2. For the default TTL of 10, this shows that the bias has to be quite large (> 20) to start achieving reasonnable success rates.

Fig. 3 is the hop-length distribution of the successfully received packets for the default bias. Fig. 4 and 5 show the same distributions, for biases 1, 5, 10 and 20. This shows that, in addition to increasing the success rate, a larger routing bias yields shorter paths overall. It is interesting to note the sawtooth aspect of the distribution. This is due

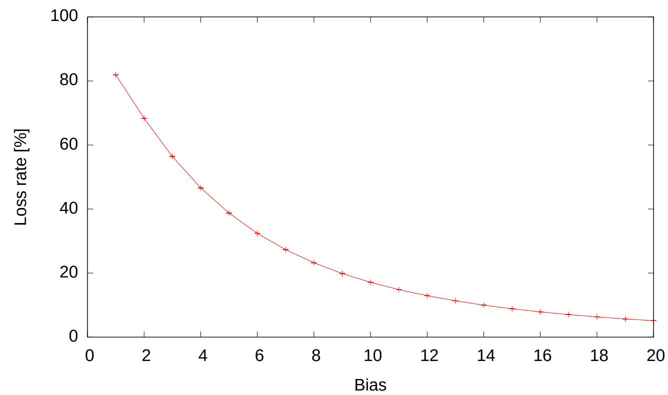

Fig. 2. Packet losses depending on the routing bias.

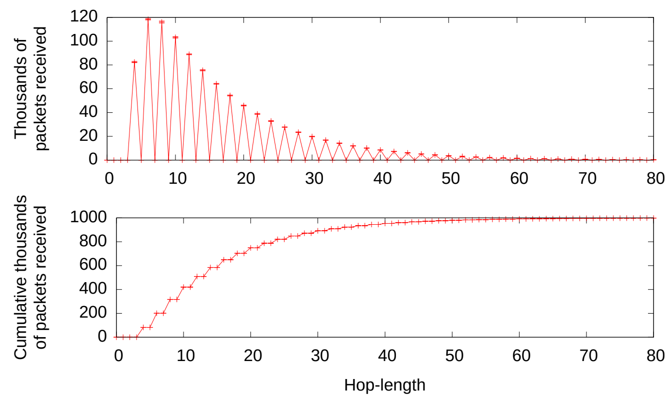

Fig. 3. Hop-length distribution of successfully received packets for the default routing bias.

to the absence of diagonal links between nodes. All valid paths from the source to the destination can only have an even number of hops. It is then logical that no sucessfully received packet has visited an odd number of nodes.

E. Size of the Lattice

Varying the size of the lattice raises the need for a change in the hop-limit range. The minimum number of hops, in the optimal case and given the fixed source and

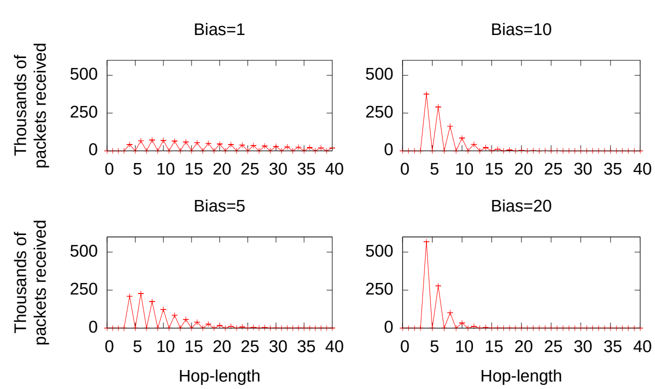

Fig. 4. Hop-length distributions of successfully received packets for various routing biases.

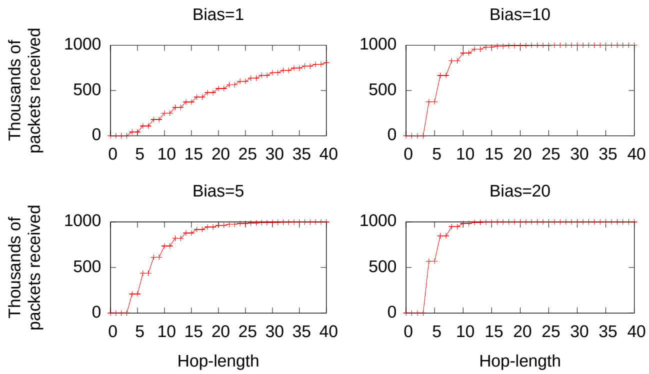

Fig. 5. Cumulative hop-length distributions for different biases.

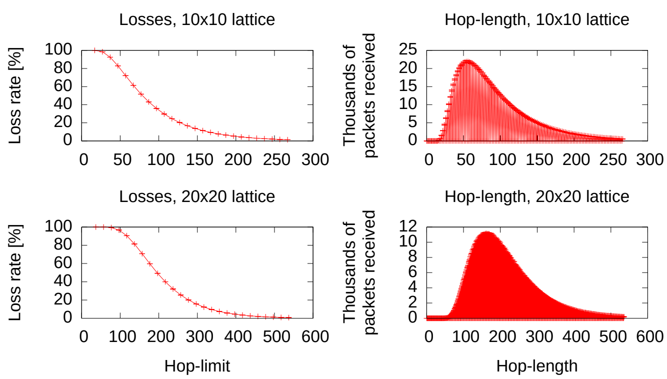

Fig. 6. Packet losses and hop-length distribution for 10×10 (top) and 20×20 lattices. (Note the difference in the x scales.)

destination nodes, is

$$hl = 2s - 3. \qquad (1)$$

This figure is used as the initial value of the hop-limit. It is then incremented by steps of s for 25 iterations.

The loss rates withinin this range and hop-length distributions are shown for 10×10 and 20×20 lattices on Fig. 6. It is interesting to note that doubling the lattice size doubles the number of hops for which similar behaviors (losses or relative number of packets with a given hop-length) are observed[1].

III. A More Complex Model

In order to address some of the behaviors observed in section II-B, the model has been extended to support diagonal links. Additionally, to better model a real worl system, random node failure has been introduced to study how well the system performs when some relay fail. These extension and their impact on the simulated system are described in the following.

A. Diagonal Links

Diagonal links were simply added to the model by considering neighbors with changes in both the x and y coordinates as possible next hops. The bias attribution code has not been changed. Fig. 7 shows the weights attributed to all the neighbors of a node, including diagonal links, for two significant examples.

[1]after the initial offset given by (1)

$$\begin{vmatrix} b+1 & 2b & b+1 \\ 2 & & 2 \\ 2 & 2 & 2 \end{vmatrix} \qquad \begin{vmatrix} 2 & b+1 & 2b \\ 2 & & b+1 \\ 2 & 2 & 2 \end{vmatrix}$$

(a) (b)

Fig. 7. The weights attributed to neighbors for the roulette-wheel selection with bias b for two cases: (a) the destination is towards the North or, (b) in the North-West direction.

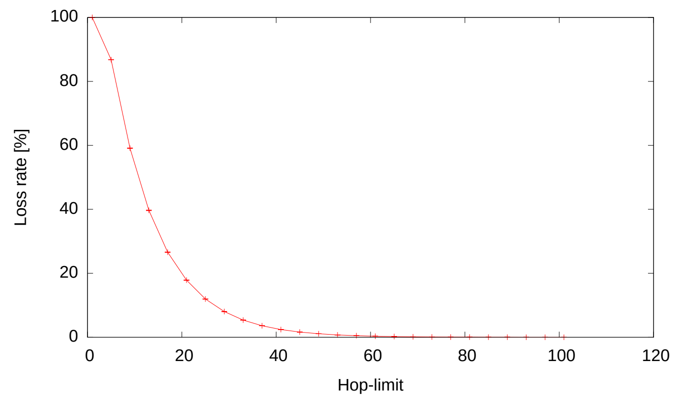

Fig. 8. Packet losses depending on the hop-limit when diagonal links are available.

Quite surprinsingly, allowing routing along diagonal links does not impact the system very much. Packet losses for a given hop-limit (Fig. 8) are only slightly reduced compared to the original model (Fig. 1). Similarly, despite the disappearance of the sawtooth feature of the hop-length distribution, the cumulative distribution only shows a small improvement (Fig. 9).

B. Failing Nodes

Failing nodes were added by incorporating two new events into the simulator: node failures and node recovery. An initial failure event at a random node is submitted to

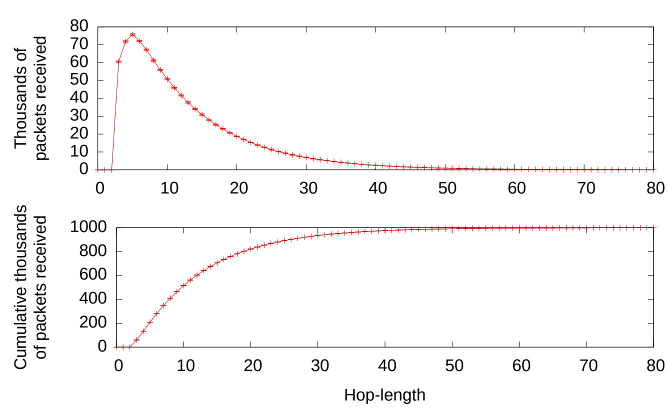

Fig. 9. Hop-length distribution of successfully received packets when diagonal routes are allowed.

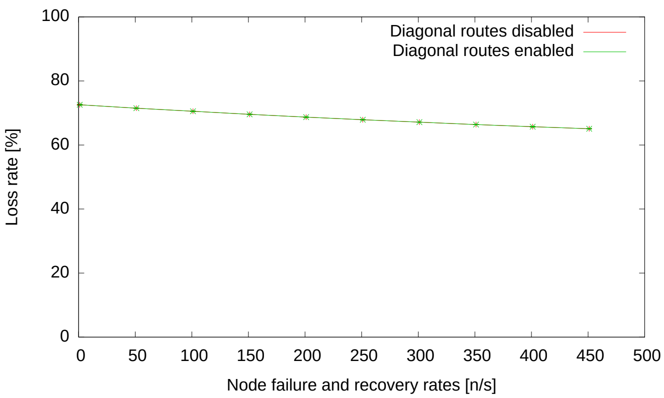

Fig. 10. Packet losses depending on the node failure rate with and without considering diagonal links. (All other simulation parameters are kept to the default from TABLE I.)

the scheduler when the simulator is started. The time at which the event happens is drawn from an exponential distribution with parametrable rate. When such an even happens, the time at which the node will recover is drawn from a similar exponential distribution with separate rate (set at the same value than the failure rate in the following simulations). When a recovery happens, the next failure event in generated and inserted in the event list.

It is worth noting at this stage that the roulette-wheel selection algorithm only selects neighbors from the pool of "up" nodes. Hence, failing nodes will not even be considered during the routing. We are interested here in the trend modifications in the hop-lengths and packets which do not expire before reaching their destination.

Fig. 10 shows the variation in the packet loss rate for a increasing value of the failure rate. Diagonal routing has also been tested as an improvement when node failures occurs, but has not not found to be efficient.

Fig. 11 compares the cumulative hop-length distributions for selected node failure rates, and the influence of diagonal routing. Apart from the removal of the sawtooth behavior (characterized by plateaux at odd hop-lengths in the CDF), the availability of diagonal routes do not significantly improve the performance. In both cases, with an increasing failure rate, successfully delivered packets tends to have a higher hop-length.

IV. Conclusion

We extended and evaluated a random geographical routing model based on a discrete event simulator. Among the findings are the facts that the routing bias has to be rather large to yield good results, or that enabling routing to diagonal nodes do not significantly improve the overall performance.

Adding support for failing nodes showed that the system, particularly due to the implementation of the roulette-wheel selection, was satisfyingly resilient to random nodes going down.

We also found that there may be a scale relation between the size of the lattice and how the hop-limit has to be chosen. This should be covered more in details in future works.

References

[1] V. Sivaraman, "Nicta network simulation short course," Dec. 2008.

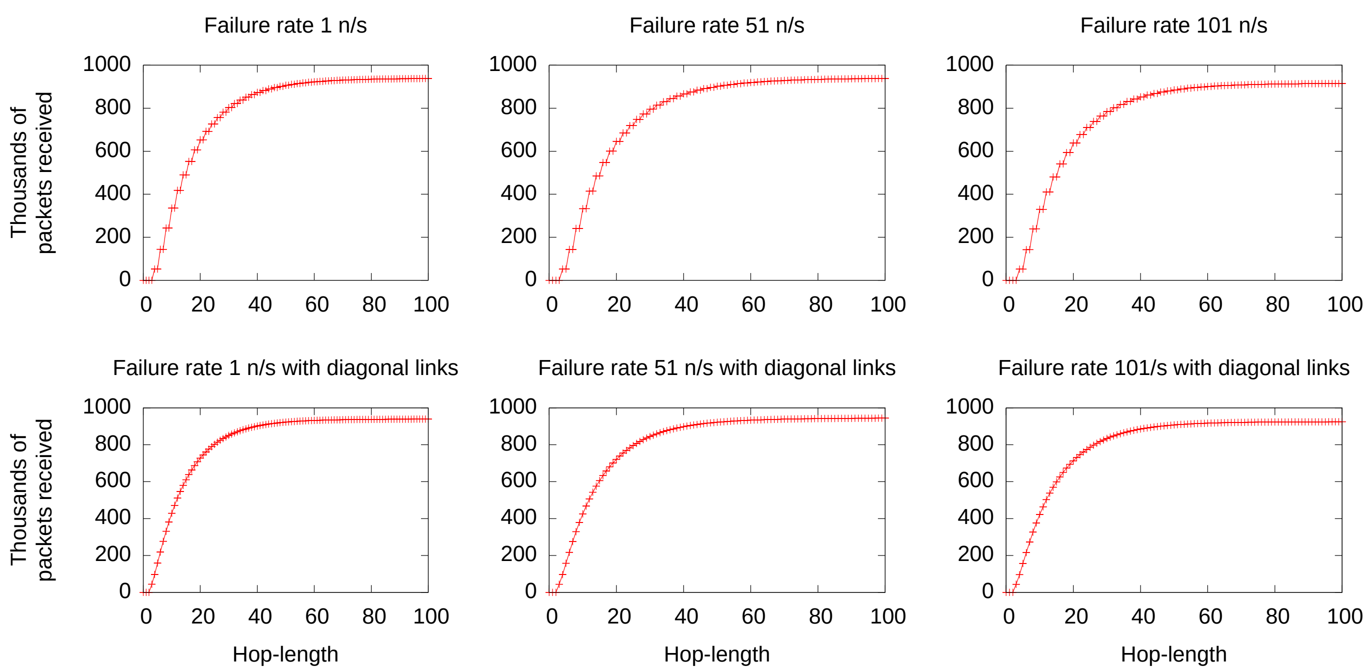

Fig. 11. Cumulative hop-length distribution for successfully received packets for various node failure rates.