



**HAL**  
open science

## Evolutionary Epidemic Routing

Sara Alouf, Iacopo Carreras, Daniele Miorandi, Giovanni Neglia

► **To cite this version:**

Sara Alouf, Iacopo Carreras, Daniele Miorandi, Giovanni Neglia. Evolutionary Epidemic Routing. [Research Report] 2007. inria-00130803v1

**HAL Id: inria-00130803**

**<https://inria.hal.science/inria-00130803v1>**

Submitted on 14 Feb 2007 (v1), last revised 22 May 2007 (v3)

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

## *Evolutionary Epidemic Routing*

Sara Alouf — Iacopo Carreras — Daniele Miorandi — Giovanni Neglia

N° ????

February 2007

Thème COM

A large, light gray, stylized letter 'R' that serves as a background for the text.

*R*apport  
de recherche





## Evolutionary Epidemic Routing

Sara Alouf, Iacopo Carreras\*, Daniele Miorandi\*, Giovanni Neglia

Thème COM — Systèmes communicants  
Projet Maestro

Rapport de recherche n° ???? — February 2007 — 12 pages

**Abstract:** We present a framework which allows forwarding schemes to evolve in order to adapt to a changing and a-priori unknown environment. The framework is inspired by genetic algorithms: at each node a genotype describes the forwarding scheme, a selection process fosters the diffusion of the fittest genotypes among the nodes, new genotypes are created by combining existing ones or applying random changes.

**Key-words:** Epidemic Routing, Genetic Algorithms

\* Create-Net, Italia

## Routage Épidémique Évolutionnaire

**Résumé :** Nous présentons une approche permettant des algorithmes de relais d'évoluer afin de s'adapter un environnement variable et inconnu a priori. Cette approche s'inspire des algorithmes génétiques. Ainsi, la politique de relais utilisée par un noeud est décrite par un génotype; un procédé de sélection promeut la diffusion, parmi les noeuds, des génotypes les plus adaptés et, enfin, de nouveaux génotypes sont créés soit en combinant ceux existants soit en appliquant sur ceux-ci des changements aléatoires.

**Mots-clés :** Routage Épidémique, Algorithmes Génétiques

## 1 Introduction

Epidemic routing [11] has been proposed as an approach for routing in sparse and/or highly mobile networks in which there may not be a contemporaneous path from source to destination. It adopts a so-called “store-carry-forward” paradigm – a node receiving a packet buffers and carries that packet as it moves, passing the packet on to new nodes that it encounters. Analogous to the spread of infectious diseases, each time a packet-carrying node encounters a new node that does not have a copy of that packet, the carrier is said to *infect* this new node by passing on a packet copy; newly infected nodes, in turn, behave similarly. The destination receives the packet when it first meets an infected node. Epidemic routing is able to achieve minimum delivery delay at the expense of increased use of resources such as buffer space, bandwidth, and transmission power. Variations of epidemic routing have recently been proposed that exploit the tradeoff between delivery delay and resource consumption, including  $K$ -hop schemes [8, 3], probabilistic forwarding [5, 4], and spray-and-wait [10, 9]. These different schemes differ in their “infection process”, i.e., the spreading of a packet in the network. They need to be combined with a so-called “recovery process” that deletes copies of a packet at infected nodes, following the successful delivery of the packet to the destination. Different recovery schemes have been proposed: some are simply based on timers, others actively spread in the network the information that a copy has been delivered to the destination [4].

Depending on the specific scenario, the main performance metric could be different, e.g. the probability to deliver a copy to the destination, the delivery time, the total energy consumption in the system or a combination of the previous ones. For a given optimization goal the choice of a specific forwarding scheme and the configuration of its parameters depend in general on the number of nodes in the system, on their mobility patterns, on the traffic generated in the networks (see for example [6]). We think that in many scenarios, these characteristics are unknown a priori and also can significantly change across time. For example a palmheld carried by a man can illustrate some of these issues. During the day the node can travel at different speeds (e.g. from zero up to car speed) and with very different trajectories (straight along highway or following a random walk from shop to shop), moving from highly crowded areas (supermarkets, classes,...) to less crowded ones, and with different level of power availability.

In order to deal with these issues, we do not propose a specific forwarding scheme, but a framework which allows forwarding schemes to evolve in order to adapt to a changing and a-priori unknown environment. We can refer to it as an evolutionary delay-tolerant routing service at the network level which can be used from the applications at the nodes. The fundamental components of this framework are: 1) the possibility to share with other nodes a description of the specific forwarding mechanism deployed at each node (the forwarding *genotype* of the node), 2) the possibility for each node to modify its forwarding scheme by applying random changes (genotype *mutations*) or by introducing some characteristics of other nodes genotypes (*crossover*), 3) a selection process which fosters the diffusion of the fittest forwarding genotypes, i.e., of those forwarding schemes which achieve the best performance in the specific environment as regards the performance metric chosen.

In what follows, after a brief description of our assumptions on nodes operation, we are going to describe in more details these three components and the open research issues.

## 2 Nodes Operation

Two nodes are able to exchange packets when they move into the transmission range of each other (when they *meet*). Once it happens, They perform the following steps:

1. exchange identification information, i.e. node ID,
2. exchange header information of data packets,
3. each node decides which packets should be forwarded to the other node,
4. packets are exchanged
5. each node can drop some packets from its buffer in order to free space for new packets

The forwarding of a packet can correspond to two different actions: the packet can be copied, i.e. both the nodes have finally a copy of the packet, or just transferred, i.e. only the receiver has finally a copy of the packet.

We can assume that packet headers contain some information about the history of the infection, like the time the data was generated or the number of nodes the packet has been forwarded from. This information can be used by the nodes for their decision process.

Note that we let the sender select the packet to forward and the specific forwarding action (copy or transfer), while we let the receiver choose packets to drop. Different solutions are possible, for example the receiver could choose which packets at the sender it is interested into, and even if the sender should keep or not a copy. Clearly if the decision criterium is the same, it does not matter where it is deployed, but we are going to have different forwarding schemes at the same time in the network, and the location of these decision processes could interact with the selection process and could drive differently the evolution.

## 3 The Genotype

As we stated in the introduction, a first step is to define a formal description of a (quite) generic forwarding scheme. We are looking for string representations which could be easily manipulated, where each letter of the string, as a gene in the DNA, specifies an action. This string can be transmitted from a node to others nodes, in this sense it represents the *genotype* of the forwarding algorithm. Genotypes require an interpreter at the node, able to make the node perform as specified from the the genotype. This is different from what happens in biological organisms, where the DNA is -quoting Shrödinger [7]- hereditary code-script, but at the same time instrumental in bringing about the development it foreshadows, both law-code and executive power. We do not assume any kind of evolution in the interpreter.

In this section we present some considerations about what a gene could be, without defining rigorously the formal language.

According to the description in Section 2, there are four basic actions in a generic forwarding scheme:

1. Choose a packet (C),
2. Copy the packet (Fc),
3. Transfer the packet (Ft),
4. Drop (D).

These are the basic genes. The first gene (C) is in general accompanied by one of the other two. These genes are in general accompanied by other genes specifying some characteristic of the action or a numerical value. Here are some possible examples:

- Timer (T),
- cOunter (O),
- uniformly Random (R),
- with probability (P)
- at the Head (H), i.e. FIFO service discipline
- at the Bottom (B), i.e. LIFO service discipline

We consider that the DNA string can be divided into two parts, the first specifies what has to be done for each new packet, the second what to do when buffer is full and there is no space for new packets.

As we said we are not going to specify the strings grammar, but we present some possible examples of strings, which we could have among the words of our language. The purpose is to show how these few genes can yet be sufficient to specify many different behaviours. Note that we use parenthesis to identify genes representing a value, and dash to separate the two parts of the string.

**CBFc-CHD** This string describes (could describe) standard epidemic routing with priority for recent packets (the node forwards more recent packets, and discards less recent ones);

**CBFcP(0.1)-CRD** describes probabilistic forwarding with forwarding probability equal to 0.1, where the packet to discard is randomly chosen from the buffer;

**CRFc0(10)-** describes a scheme where the packet to forward (to copy in this case) is randomly chosen, and for each packet a counter is created limiting to 10 the number of times the copy is performed, there is no description about how to chose the packet to drop when the buffer is full, this corresponds (again, it could correspond) to the case where incoming packets are discarded when the buffer is full;



**CRFc0(10)Ft-** describes a scheme similar to the previous, but where the packet is transferred after having been copied 10 times.

More complex schemes can be obtained if we include the possibility to read and change fields of the packet.

In the previous examples we have considered variable length strings, but the dictionary could contain fixed length words, where each position corresponds to a given function/characteristic. In the first case we need in general to introduce genes to mark the begin or the end of different parts of the string, like the dash in the previous examples, in the second case we need to introduce a dumb gene to fill uninformative positions.

While it does not seem difficult to create a formal language to include such simple examples, the tradeoff between the complexity of the description and the variety of possible schemes which can be described (the possible phenotypes) has to be investigated, see also comments in Section 6.

## 4 Generation of new Genotypes

The system can evolve if we let nodes change their genotypes. In this section we address the problem of how new genotypes should be generated.

One possibility is that a node simply inherits the genotype from another node in the network. This can appear reasonable especially for nodes arriving in a new environment. Inheritance clearly provides the node a new genotype, but does not increase the number of different genotypes in the network.

Mutation is another possibility. The node can randomly change one or more of the genes and could experience the new genotype for a while. The change could affect a gene representing a parameter or the action to perform itself.

The last possibility is to cross-over two, or more, different genotypes to create a new one.

## 5 The Selection Process

A selection process is needed in order to select good genotypes and let the network evolve towards the adoption of good forwarding schemes. This aspect of evolutionary delay tolerant routing seems to be the trickiest one. In what follows we are going to discuss what are the important issues and to present some possible approaches.

First, we want to stress the differences with the natural selection process. In nature it is easy to evaluate a posteriori which are the best genes at a given moment: they are simply those present in future generations, i.e. those genes which allowed the organisms carrying them to have more offspring in the long term [1]. The selection process can even be assumed independent from the reproduction process which introduces new genotypes through mutations and crossover. This is not true in our case, good genotypes are for us those able to achieve good performance for a specific metric, and the environment does not

include automatically any selection process<sup>1</sup>. A solution in order to make good genotypes more represented in the long term is to favour them in the generation process. Each node can have its own fitness index and this index could drive the generation. For example nodes with high fitness index could incur mutations less frequently, and when two genotypes are crossed over, the new genotypes could be more similar to the genotype of the node with the highest fitness.

The determination of a fitness index is difficult because the forwarding service is intrinsically a distributed service. The node in general has limited information about the spread of the information and its specific contribution to the delivery of the packet. The node can use the information that stored in the packet header (TTL like field) or that it can retrieve from meetings with other nodes (e.g. how many other copies of the packet are in its neighborhood). The destination has a predominant position, because it can evaluate the final results of the infection process. For this reason we suggest two different ways to build local nodes fitness which are in some sense both destination-driven.

In the first one we assume that each node evaluates the others, in particular when node A receives a new packet from node B it evaluates the current forwarding path characteristics and can decide to reward node B. This reward changes node B fitness index. The higher node A fitness index in comparison to the average, the more significant is the change. The process is destination-driven, because it is the destination to introduce some differences in the fitness levels, by evaluating nodes from which it receives a copy of the packet. The evaluation process can be considered a backward process starting from the destination up to the source. This approach clearly requires that we can trust fitness indexes advertised by nodes.

A more robust approach only relies on evaluations from the destination. In this case we assume that each node adds its own genotype to the data in the packet, hence the destination knows all the genotypes which contributed to the delivery and can decide to reward them. Different strategies with different overheads are possible: the destination could randomly select a genotype to reward from the list or could reward all of them, and it could communicate the reward just when it meets the node using it, or it could use some form of delay tolerant routing to send the reward to the node. Another possibility is to send back to the source the list of genotypes and then let the source send again the list to the destination, so that nodes along the paths can increase their fitness index if they see their own genotype in the list.

## 5.1 Network heterogeneity

An important issue to take into account is that we could want the evolution to select a pool of genotypes rather than a single good genotype. In fact nodes in the network could be very different and the best routing services could derive from the composition of different genotypes. For example nodes with different mobility models could have different

---

<sup>1</sup>Only energy consumption can be considered in some scenarios an intrinsic selection process, forwarding schemes which consume a lot of energy could exhaust nodes batteries and then they would be less represented in the population.

good genotypes. Consider for example a situation where the network is mainly divided into two clusters of nodes located at different localities with a limited number of nodes moving between the two places and making possible in this way to transmit information from one locality to the other. We can easily imagine that good genotypes for these migratory carriers could be different from good genotypes for the other nodes. Note that both selection processes described above implicitly allow nodes to have different genotypes depending on their position in the forwarding chain. Clearly multiple genotypes are not needed if a single genotype is able to specify different behaviours depending on the mobility characteristic of the node. As it occurs in nature, the behaviour of the organism can be different depending on the specific environmental conditions, and node mobility can be considered an environmental data as long as the nodes are carried and cannot determine their movements. As a final remark, the similarity between nodes can also be taken into account in the generation process, advantaging the genes of the more similar nodes.

Another aspect related to heterogeneity is that the goodness of a genotype (or of a pool of genotypes as in the example above) can depend on the specific source-destination pair. We can look for a good average behaviour or we could want nodes to have different behaviours for different pairs. A simple solution is to allow nodes to have multiple genotypes, one for each source-destination pair. Again a complex genotype could intrinsically take into account a wider range of behaviours.

## 5.2 Further remarks

Note that, although the rewards are given on the basis of a specific performance metric, the process itself can introduce some biases. Here we just show two possible examples: 1) even if power issue is not considered in the optimization goal, genotypes leading battery exhaustion, would intrinsically be penalized; 2) a fitness reinforcement process triggered by packet delivery could advantage the genotypes useful to those source-destination pairs generating more traffic, this would lead to unfairness across different flows.

Differently from what happens in nature, a node can choose its own genotype. For example it can decide to revert to a previous genotype, if it considers the new one unsuited to the environment or bad according to some specific local function, e.g. nodes going out of battery can strongly limit their participation into the forwarding process.

## 6 Evolution or Adaptation?

A possible critic to the framework presented, and in particular to the genotype description in Section 3, is that what we are considering is more a complex form of adaptation rather than a form of evolution, because the nodes will finally select one behaviour out of a finite set. We think that the distinction between adaptation and evolution is in part a problem of scale: even human DNA has only a finite number of possible different combinations, hence in some sense natural evolution can only choose in a limited set of behaviours that could be theoretically evaluated a priori. The critical point is the expression capacity of the genotype

description we are going to choose. The higher this capacity (and a higher capacity in general requires a more complex language), the larger the number of possible behaviours, perhaps to the extent that the system will start exhibiting some new (unforeseen) behaviours.

## 7 A Broader Framework

As we have shown above, even the simple scenario of an evolving delay tolerant routing service presents many aspects which need careful investigation. We want to stress that other aspects can be included in a more general evolutive routing service, like for example the choice of the routing protocol to use, moving from the delay tolerant approach to reactive or proactive forms, or the choice of the transmission technique to use (Bluetooth or 802.11).

## 8 A First Simulation Study

In this section we investigate a few basic network scenarii to achieve some insight to the feasibility of the proposed framework. In particular our purpose is to answer to the following questions:

1. Does the distributed genetic algorithm “converge”? and if so, how much time is required for the convergence? is the reproduction timescale a critical factor to ensure convergence?
2. What are the performance in comparison to an optimally configured forwarding scheme?
3. Do nodes exhibit different behaviours in a heterogeneous scenario?

We consider a simple fixed length genotype including only two parameters: the maximum number of copies each node is allowed to make (gene C) and the probability to copy a packet (gene P). When a node can forward many packets to another, it starts transmitting them after a random shuffle of these packets. Similarly, when the buffer is full, the packet to discard is chosen according to uniform random distribution.

As optimization goal we consider the minimization of the delivery time  $T_D$ . We assume that all the nodes are synchronized and packet header contains a field specifying the time the payload was generated at the source<sup>2</sup>. In such a way the destination can easily evaluate the delivery time as soon as it receives the first copy and can immediately start evaluating the infection process. In particular the destination determines a reward as a decreasing function of the delivery time  $r(T_D)$ , and this reward is propagated to all the nodes. This information is transmitted during the initial phase of the communication when two nodes meet (together with packet header exchange). Each node updates its own fitness summing the reward to

---

<sup>2</sup>If local clocks are enough accurate at the packet delivery timescale, then there is another solution which does not require synchronization. Packet header should have a field which indicates the time since the payload was generated. This field can be updated by each node before forwarding the packet. In this case the node should keep track of the time since it received the packet.

the previous value of the fitness ( $\phi(t^+) = \phi(t^-) + g(T_D)$ ). At the same time, the fitness decreases as time goes by ( $\phi(t + \tau) = \phi(t) * \exp(-\alpha\tau)$ ). This feature makes the system more reactive to change of the network scenario (e.g. the number of nodes or the traffic matrix). The reward acts also as an antipacket which let the node erase the corresponding packet.

When two nodes meet, they transmit each other their own genotype and its fitness, hence each node maintains a pool of available genotypes. Every  $T$  seconds, the *intergeneration time*, the node procreates a new offspring, i.e. it updates its own genotype. This event can happen at the same time at all the nodes (in this case different generations can be clearly identified during network evolution) or it can happen asynchronously. Crossover is performed with probability  $p_c$  and requires to select two genotypes from the pool, otherwise only one genotype has to be selected. The genotypes are selected with a probability proportional to their own fitness (rescaled, see below). Finally each bit of the current genotype, directly selected from the pool or obtained through crossover of two genotypes, can mutate with probability  $p_m$ . The genotype pool is emptied after every procreation. If the network is large, the pool size could be not large enough to keep all the genotypes a node discovers between two procreations. In this case we take into account the original fitness of the genotype, using again a fitness-biased distribution or just keeping the genotypes with the highest fitness. If we do not want to erase all the information after a generation, then we can take into account the age of each genotype in the pool by opportunely discounting its original fitness as time goes by<sup>3</sup>. We mentioned above fitness scaling, this is a common procedure in genetic algorithms in order to avoid the phenomenon of *premature convergence* [2].

## 8.1 First scenario

We consider  $N$  nodes moving in finite area according to a standard mobility model like random waypoint or random direction ones. We have only one destination, and only one packet spreading at each time. We expect the forwarding scheme to “converge” to standard epidemic routing, i.e. we expect to have many genotypes with high values of gene C and gene P.

## 8.2 Second scenario

The scenario differs from the first because we consider a higher density and/or more memory in the mobility process (e.g. brownian motion). We consider wireless communication interference. We expect to have smaller values of gene P in the population, because an infected node will have in general a higher number of infected neighbours and an aggressive infection procedure could lead to many useless collisions.

<sup>3</sup>This discount is different from the previous one. The node should not take into account that other nodes fitness decreases over time, because it cannot take into account the rewards they are receiving.

### 8.3 Third scenario

The third scenario differs from the first one because there are many packets spreading in the network at the same time. Even if we neglect wireless communication interference, we expect the population to exhibit smaller values of  $C$ . In fact nodes can gain from limiting the spreading of each packet because this will implicitly promote the packets that have less copies in the system. In presence of wireless interference, the reduction should be more significant.

### 8.4 Fourth scenario

In this scenario we consider heterogeneous mobility patterns for the nodes. There are two disjoint identical areas (island A and island B). There are some nodes moving according to random waypoint in island A, and others moving in the same way in island B. There are also other nodes (migratory carriers) which continuously move from one island to the other. Sources belongs to island A, while the destination belongs to island B. If the traffic is “high” in comparison to the size of the migratory carriers population, but small in comparison to the size of the stationary population, we expect different values of gene  $C$  in the two populations: small values for the migratory population and high values for the stationary one.

## References

- [1] Richard Dawkins. *The selfish Gene*. Oxford University Press, New York, 1976.
- [2] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.
- [3] R. Groenevelt, P. Nain, and G. Koole. The message delay in mobile ad hoc networks. In *Performance*, October 2005.
- [4] Z. J. Haas and T. Small. A new networking model for biological applications of ad hoc sensor networks. *IEEE/ACM Transactions on Networking*, February 2006.
- [5] A. Lindgren, A. Doria, and O. Schelen. Probabilistic routing in intermittently connected networks. In *First International Workshop on Service Assurance with Partial and Intermittent Resources (SAPIR 2004)*, Fortaleza, Brazil, August 2004.
- [6] G. Neglia and X. Zhang. Optimal delay-power tradeoff in sparse delay tolerant networks: a preliminary study. In *ACM SIGCOMM workshop on Challenged Networks (CHANTS 2006)*, September 2006.
- [7] Erwin Schroedinger. *What is life*. Cambridge The University Press, 1992.

- [8] G. Sharma and R. R. Mazumdar. Delay and capacity tradeoffs for wireless ad hoc networks with random mobility. Submitted for publication.
- [9] T. Small and Z. J. Haas. Resource and performance tradeoffs in delay-tolerant wireless networks. In *ACM workshop on Delay Tolerant Networking*, 2005.
- [10] T. Spyropoulos, K. Psounis, and C. S. Raghavendra. Spray and wait: an efficient routing scheme for intermittently connected mobile networks. In *ACM workshop on Delay-tolerant networking*, 2005.
- [11] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. Technical Report CS-200006, Duke University, April 2000.



---

Unité de recherche INRIA Sophia Antipolis  
2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Futurs : Parc Club Orsay Université - ZAC des Vignes  
4, rue Jacques Monod - 91893 ORSAY Cedex (France)

Unité de recherche INRIA Lorraine : LORIA, Technopôle de Nancy-Brabois - Campus scientifique  
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex (France)

Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)

Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier (France)

Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex (France)

---

Éditeur  
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)

<http://www.inria.fr>

ISSN 0249-6399