



**HAL**  
open science

## “ARC”: Simulateur d’architecture de réseaux de communications

Pascal Desnogues, Hery Rakotoarisoa

► **To cite this version:**

Pascal Desnogues, Hery Rakotoarisoa. “ARC”: Simulateur d’architecture de réseaux de communications. [Rapport Technique] RT-0187, INRIA. 1996, pp.24. inria-00069984

**HAL Id: inria-00069984**

**<https://inria.hal.science/inria-00069984>**

Submitted on 19 May 2006

**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

**“ARC”:**  
*Simulateur d’architecture de réseaux de  
communications*

Pascal Desnogues - Hery Rakotoarisoa

**N° 0187**

Janvier 1996

PROGRAMME 1



*rapport  
technique*



**“ARC”:  
Simulateur d’architecture de réseaux de  
communications**

Pascal Desnogues - Hery Rakotoarisoa \*

Programme 1 — Architectures parallèles, bases de données, réseaux  
et systèmes distribués  
Projet Mistral

Rapport technique n° 0187 — Janvier 1996 — 24 pages

**Résumé :** ARC est un prototype de simulateur spatial d’architectures de réseaux de communications. Les objets constituant un modèle, tels que les abonnés et les commutateurs dans un réseau de communications, sont la réalisation d’un ensemble de processus ponctuels dans le plan. Les caractéristiques à étudier sur le modèle s’expriment comme des fonctionnelles de ces processus ponctuels, et ces fonctionnelles peuvent être estimées et optimisées par rapport aux paramètres de ces processus ponctuels. La construction d’un modèle et sa manipulation sont basées sur des algorithmes de géométrie aléatoire.

Les différents objets intervenant dans la simulation sont détaillés, ainsi que les contraintes qui les lient pour la construction et la manipulation du modèle. Les structures de données nécessaires sont décrites ainsi que les fonctionnelles actuellement implantées dans ARC. Quelques exemples de résultats de calcul sur ces fonctionnelles sont donnés.

**Mots-clé :** Simulation spatiale, Réseaux de communications, Modèle hiérarchique, Processus ponctuels, Gradient stochastique, Diagramme de Voronoï, Triangulation de Delaunay.

*(Abstract: pto)*

Travaux effectués dans le cadre des conventions de recherche CNET-INRIA CTI 1B 104

\* . Email : {pdesnog | hery}@sophia.inria.fr

# **“ARC”: Communication Networks Architecture Simulator**

**Abstract:** ARC is a prototype of spacial simulator of communication networks architectures. The objects making up a model, such as subscribers and stations in a communication network, are the realization of a set of point processes in the plane. The model characteristics to be evaluated are functionals of these point processes, and these functionals can be estimated and optimized with respect to the point processes parameters. The model construction and handling are based on stochastic geometry algorithms.

The different objects used in the simulation are detailed, as well as connection constraints between them for the construction and the handling of the model. Needed data structures are described, as well as the functionals currently implemented in ARC. Some results of computations on these functionals are given.

**Key-words:** Spacial simulation, Communication networks, Hierarchical model, point process, stochastic gradient, Voronoi diagram, Delaunay triangulation.

# 1 Introduction

Nous présentons dans ce rapport un simulateur d'architectures de réseaux de communications nommé ARC. Ce prototype a été réalisé, dans le cadre de la convention de recherche sur la modélisation macroscopique de réseaux entre le CNET et l'INRIA, dans le but d'expérimenter et de valider les résultats théoriques obtenus ainsi que d'analyser des modèles plus complexes que l'on ne peut manipuler analytiquement.

Il s'agit d'une simulation de processus dans le plan fondée sur des algorithmes de géométrie aléatoire. Les différents objets d'un modèle, tels que les abonnés et les commutateurs dans un réseau de communication, sont considérés comme la réalisation d'un ensemble de processus ponctuels aléatoires dans le plan. Ainsi, le modèle contient toutes les caractéristiques spatiales du réseau en étant simplement décrit par les paramètres des processus stochastiques utilisés, et les caractéristiques discriminantes pour les études que l'on veut faire (optimisation d'architecture, analyse économique ...) s'expriment comme des fonctionnelles de ces processus ponctuels.

Dans la première partie de ce rapport, nous présenterons les différents objets qui interviennent dans la simulation, les caractéristiques de ces objets, ainsi que les contraintes et les paramètres de liaison de ces objets pour la construction du modèle. Nous terminerons cette première partie par une courte énumération des caractéristiques intéressantes d'un modèle qui peuvent être les fonctionnelles des processus ponctuels.

La deuxième partie est consacrée au prototype ARC, ses fonctionnalités, ses objets de base, les algorithmes et les structures de données qu'il utilise, et les fonctionnelles qui y sont actuellement implantées. Un aperçu des évolutions possibles de ARC dans le cadre ou au delà de cette convention de recherche terminera cette partie.

## 2 Les objets de l'étude et les modèles de liaisons

### 2.1 Le modèle

Les objets considérés dans les études que l'on veut faire sont les *abonnés* et les *commutateurs*. Ces derniers sont reliés entre eux par un *réseau de commutation*, et un abonné est *raccordé* à un de ces commutateurs pour faire partie du réseau. Dans

le reste de ce rapport, nous appellerons commutateurs tous les noeuds d'un modèle qui ne sont pas des abonnés.

Nous appellerons *lien physique* ou *lien atomique* un lien reliant uniquement deux points (abonné ou commutateur) et correspondant à un lien de transmission du point de vue réalisation technique.

Nous appellerons *lien logique*, ou tout simplement *lien*, un lien de raccordement d'un abonné à un commutateur ou un lien de commutation entre deux commutateurs. Un lien logique peut donc être composé de plusieurs liens physiques.

Les *flux de communications* sont établis entre deux noeuds du réseau (abonné ou commutateur), et empruntent des liens qui constituent un chemin selon la politique choisie pour l'acheminement des communications.

Enfin, les *cellules*, qui sont chacune associées à un commutateur, définissent les *zones de desserte* de ces commutateurs.

### 2.1.1 Raccordement des abonnés

Un abonné peut être raccordé par des liens sur un ou plusieurs commutateurs. La politique globale d'attachement d'un abonné à un commutateur est à définir, bien que certains abonnés ayant des caractéristiques particulières ne peuvent être attachés qu'à des commutateurs particuliers.

La politique la plus simple est celle du *plus proche* ou celle des  *$N$  les plus proches*, c'est à dire qu'un abonné est raccordé au commutateur le plus proche ou aux  $N$  commutateurs les plus proches en distance euclidienne. Le pavage de l'espace au niveau le plus bas des commutateurs est donc du type Diagramme de Voronoï [1]. Ce pavage définit les cellules associées aux commutateurs de ce niveau.

Si l'on considère que les commutateurs ne sont pas tous identiques mais ayant chacun leurs caractéristiques leurs permettant de servir une zone plus grande que les autres, le pavage de l'espace peut être du type Diagramme de puissance, plus général que celui de Voronoï.

### 2.1.2 Réseau des commutateurs

Si les commutateurs sont tous classés au même niveau hiérarchique, la politique globale de leur liaison est à définir, bien que pour diverses raisons, certains commutateurs sont reliés explicitement à d'autres. On peut par exemple définir un réseau maillé selon la triangulation de Delaunay qui est la structure géométrique duale au

diagrammes de Voronoï. Nous reviendrons sur ce type de triangulation dans la section 3.3.2 ci-après.

Si le réseau des commutateurs est hiérarchique, la politique d’attachement des commutateurs d’un niveau à ceux du niveau supérieur peut être définie de la même manière que l’attachement des abonnés. Un pavage est ainsi défini pour chaque niveau de hiérarchie (sauf pour le plus bas niveau déjà défini lors de l’attachement des clients). Pour les commutateurs de même hiérarchie, notamment ceux de plus haut niveau, une politique de maillage peut également être utilisée.

## 2.2 Les objets de l’étude

Les objets à manipuler pour les études que l’on veut faire sont les abonnés, les commutateurs, les différents types de liens, les flux de communications et les cellules ou zones de desserte des commutateurs. Un *noeud* d’un modèle est un abonné ou un commutateur. Les attributs et les caractéristiques des différents objets sont énumérés ci-après. Les valeurs de certains d’entre eux peuvent être calculées au fur et à mesure des besoins, comme par exemple les différentes longueurs de liens. Ils peuvent donc ne pas figurer directement dans l’objet par soucis d’occupation mémoire.

### 2.2.1 Les abonnés

Ils constituent les noeuds du niveau hiérarchique le plus bas (niveau 0). Un abonné est caractérisé par :

- son identité,
- ses coordonnées,
- la liste des commutateurs auxquels il est raccordé,
- son type fixe ou mobile,
- une indication s’il est *transmetteur* ou non. Cette indication permet de considérer ou non l’abonné comme pouvant transmettre des flux de communication.
- son coût de mise en oeuvre;
- des caractéristiques particulières éventuelles.

### 2.2.2 Les commutateurs

Ils constituent les noeuds des niveaux supérieurs (niveaux  $\geq 1$ ) de la hiérarchie. Les caractéristiques d'un commutateur sont:

- son identité;
- ses coordonnées,
- le niveau de hiérarchie où il se situe;
- les liens qui le relie aux commutateurs de niveaux supérieurs, à ceux du même niveau, et à ceux de niveaux inférieurs;
- indication s'il est transmetteur ou non,
- son coût de mise en oeuvre;
- sa capacité, par exemple en nombre de liens et en quantité de trafic.

D'autres caractéristiques peuvent être ajoutées comme pour les abonnés selon les besoins futurs, par exemple des paramètres pour le modèle équivalent en réseaux de files d'attente.

### 2.2.3 Les liens physiques

Un lien physique, qui modélise un lien de transmission, relie directement un abonné et un commutateur ou deux commutateur. Ses caractéristiques sont :

- son identité;
- les noeuds qui sont attachés à ses deux extrémités (abonné ou commutateur);
- sa longueur ou la distance euclidienne entre ses deux extrémités;
- son support matériel (cuivre, optique, etc..., aérien ou dans une tranchée);
- la liste des liens logiques qui l'utilisent;
- son coût par unité de longueur.

#### 2.2.4 Les liens logiques

Modèle d'un lien de commutation, un lien logique relie également un abonné à un commutateur ou deux commutateurs entre eux. Un lien a les caractéristiques suivantes :

- son identité;
- les noeuds qui sont attachés à ses deux extrémités (abonné ou commutateur);
- la distance euclidienne entre ses deux extrémités et la somme des longueurs des liens physiques utilisés;
- la liste des liens physiques qui le constituent,
- un nombre (1, 2, ou 3 ou ...) indiquant si le lien relie un noeud au 1<sup>er</sup>, au 2<sup>ème</sup> ou au 3<sup>ème</sup> ou ... noeud le plus proche du niveau supérieur. Cet attribut vaudra par exemple 0 si le lien relie deux noeuds ne se trouvant pas dans deux niveaux hiérarchiques voisins.
- son support matériel (cuivre, optique, etc..., aérien ou dans une tranchée),
- son coût par unité de longueur.
- le sens des communications qu'il accepte (bidirectionnel ou non),
- sa capacité en quantité de flux de communication.

Un modèle peut contenir des liens qui relient des objets ne se trouvant pas sur deux niveaux consécutifs.

#### 2.2.5 Les flux de communications

Un flux de communication est établi entre deux noeuds du modèle suivant un chemin pouvant utiliser plusieurs liens logiques. Un flux de communication est caractérisé par :

- son identité;
- les deux noeuds qu'il connecte;

- la distance euclidienne entre ces deux noeuds et la somme des longueurs des liens utilisés;
- la distance en terme de niveaux entre les deux noeuds;
- la liste des liens qu’il utilise;
- des paramètres spécifiques caractérisant le type de trafic qu’il représente.

Les caractéristiques d’acheminement (contraintes de choix des liens à utiliser) sont données pour la mise en place des objets de type flux (notamment l’attribut *liste des liens utilisés*). Par défaut, les flux empruntent les liens dont les extrémités sont les plus proches de leur destination sur le chemin hiérarchique. L’utilisation des liens auxiliaires ( $n^{ième}$  plus proche) est à spécifier (par exemple équirépartition, partage de charge selon telle ou telle règle, ...).

### **2.2.6 Les cellules ou zones de desserte des commutateurs**

Les cellules modélisent les zones de desserte des commutateurs. Les caractéristiques d’une cellule sont :

- son identité;
- le commutateur correspondant;
- la liste des points ou des segments qui composent son contour;
- ses cellules et commutateurs voisins.

## **2.3 Contraintes de liaisons des objets**

La figure 1 résume ces différentes données et contraintes pour la construction du modèle. Les paramètres dans un parallélogramme doivent être donnés par l’utilisateur pour la génération des différentes parties du modèle encadrées par des rectangle.

Le modèle d’un réseau, représenté par des points (abonnés et commutateurs) reliés par des segments (liens), est définie de la manière suivante:

- le nombre de niveaux doit être donné par l’utilisateur;

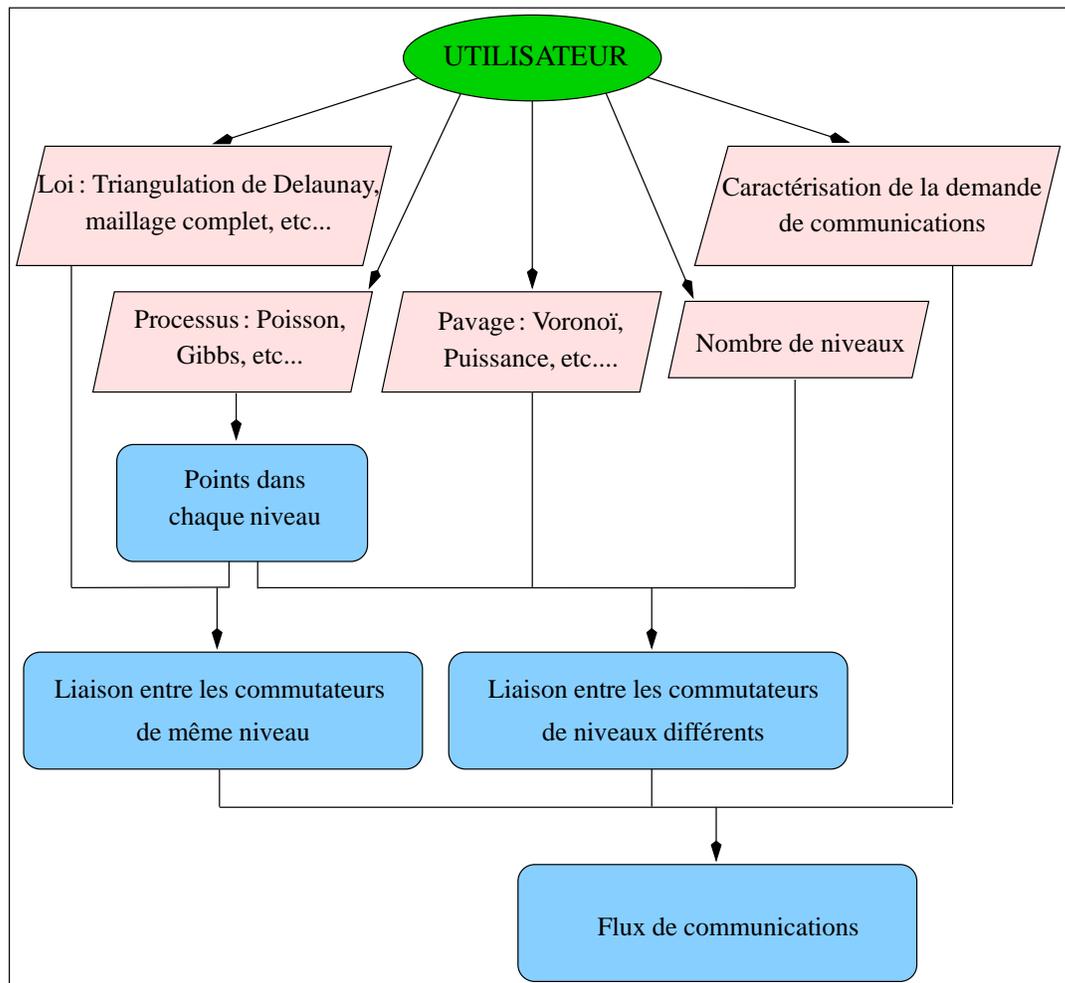


FIG. 1 – *Contraintes de liaison des objets pour la construction d'un modèle.*

- l'ensemble des points dans chaque niveau est défini par un processus ponctuel de génération (par exemple poisson( $\lambda$ )). Pour l'exécution, cet ensemble, ainsi que le nombre de niveaux ci-dessus, peuvent être chargés à partir d'un fichier dans lequel ils ont été préalablement rangés;
- la règle de liaison entre les commutateurs de même niveau doit être donnée (par exemple selon la triangulation de Delaunay);
- la règle de liaison entre les commutateurs de niveaux différents doit également être donnée (par exemple la règle du plus proche - diagramme de Voronoï).

## 2.4 Les caractéristiques intéressantes d'un modèle

Comme énumérées dans [2], les principales grandeurs d'intérêt dans un modèle, qui peuvent être exprimées comme des fonctionnelles des processus ponctuels, sont:

- la distribution des caractéristiques géométriques d'une zone de raccordement typique, par exemple sa surface, la longueur de sa frontière, le nombre de commutateurs adjacents, etc.;
- la distribution du nombre et de la position des abonnés dans une zone typique;
- la distribution de la longueur totale des liaisons entre les abonnés et le commutateur correspondant;
- la distribution des caractéristiques géométriques des liaisons entre les différents commutateurs.

## 3 Le prototype ARC

### 3.1 Les fonctionnalités de ARC

ARC est un simulateur spatial de réseaux de communications. Il permet:

- la construction d'un modèle par génération aléatoire ou par chargement à partir d'un fichier,
- la sauvegarde d'un modèle dans un fichier,

- la visualisation graphique du modèle, et
- différents calculs sur les caractéristiques et les paramètres du modèle.

Les interactions de ARC avec l'utilisateur se font à l'aide d'une interface graphique conviviale où chaque action est associée à un bouton. Toute la partie graphique de ARC, c'est à dire les panels d'interactions avec l'utilisateur et les fenêtres de visualisation, a été réalisée à l'aide de la bibliothèque LEDA.

LEDA [5] est une bibliothèque de structures de données simples et complexes, et de primitives graphiques de visualisation d'objets de base en géométrie.

### 3.1.1 Construction d'un modèle

Lorsque le modèle doit être généré de manière aléatoire, les paramètres de génération du modèle sont définis par l'utilisateur. Un modèle peut être reproduit en choisissant une graine fixe du générateur de nombre aléatoire, ARC en fournit neuf différents et de valeurs fixes. Ce générateur de nombre aléatoire étant portable, un modèle peut être reproduit même sur une autre architecture. La valeur par défaut de la graine est elle-même aléatoire.

Les paramètres fournis par l'utilisateur pour le génération du modèle sont les suivants.

- Les coordonnées de l'espace du modèle, plus précisément les coordonnées  $(x_{min}, x_{max}, y_{min}, y_{max})$  de la fenêtre.
- Le nombre  $n$  de niveaux hiérarchiques dans le modèle, le niveau 0 étant le plus bas et le niveau  $(n - 1)$  le plus haut.
- Le type et les paramètres du processus ponctuel de génération des points sur chaque niveau. Actuellement, le choix de l'utilisateur est limité entre un processus de Poisson homogène dans l'espace et un nombre fixe de points distribués uniformément dans l'espace.
- Le type de connexion des différents points sur le même niveau. Ces points peuvent ne pas être interconnectés, totalement connectés entre eux ou connectés selon la triangulation de Delaunay.

- Le nombre de connexions d'un point donné d'un niveau hiérarchique aux points du niveau supérieur. Ce nombre est actuellement limité à un, c'est à dire qu'un point est connecté au seul point qui lui est le plus proche en distance.
- Une indication si les points du niveau le plus bas (niveau 0) sont des abonnés ou des commutateurs.

Dans la version actuelle de ARC, on ne considère au moment de l'attachement d'un abonné, que les commutateurs de plus bas niveau (ou tous les commutateurs s'ils sont tous sur le même niveau). Il en est de même pour l'attachement d'un commutateur d'un niveau donné à ceux du niveau immédiatement supérieur. Il n'y a donc pas de connexion entre deux éléments se trouvant sur deux niveaux de distance supérieure à 1.

### 3.1.2 Visualisation graphique du modèle

Différentes possibilités de visualisation sont offertes à l'utilisateur.

- Il est possible de faire un *zoom* dans une partie de l'espace en changeant les coordonnées de visualisation. Une grille peut être affichée pour faciliter le repérage des coordonnées.
- Les niveaux hiérarchiques ainsi que les objets à visualiser peuvent être sélectionnés. Par exemple, on peut ne visualiser que les commutateurs d'un niveau donné, les connexions entre eux et leurs cellules correspondantes (figure 4).
- La couleur de visualisation de chaque type d'objet peut être choisie dans une palette de quinze couleurs.

### 3.1.3 Les calculs sur le modèle

Les fonctionnelles que l'on peut actuellement estimer dans ARC sont les suivantes:

- les caractéristiques des zones de desserte des commutateurs sur un niveau hiérarchique donné d'un modèle : la surface, la longueur de la frontière et le nombre de voisins;

- le coût de raccordement des abonnés dans un modèle à trois niveaux hiérarchiques. Le coût en question est le coût total de l'arbre dont la racine est un commutateur du niveau supérieur incluant le coût de réalisation de ce commutateur et de ceux des niveaux intermédiaire et inférieur, le coût des tranchées (l'appellation *tranchée* ici assimile tous les support entraînant des travaux de génie civil), et le coût du câble. Comme nous le montre la figure 2, une seule tranchée relie deux sites du modèle, et un câble relie un abonné (niveau 0) à un commutateur de niveau 2 passant par un commutateur de niveau 1.

L'espérance de chacune des ces fonctionnelles est estimée par le calcul de leurs moyennes sur l'ensemble du modèle. Pour les caractéristiques géométriques des cellules sur un niveau hiérarchique, leurs gradients est estimé par rapport à l'intensité du processus ponctuel associé. Pour le coût, le gradient est estimé par rapport à l'intensité du processus ponctuel du niveau 1. Les résultats obtenus sont toujours comparés au résultat théorique pour ces calculs.

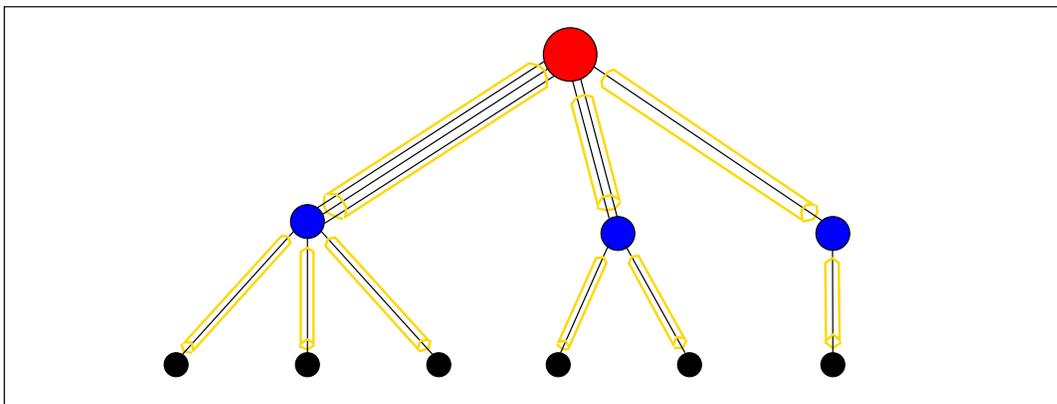


FIG. 2 – *Le coût à optimiser selon l'intensité du processus de niveau 1*

L'estimation des gradients dans ARC est basé sur une généralisation de la technique dite d'analyse de perturbation de processus ponctuels (ou Rare Perturbation Analysis) décrite dans [3]. Dans la pratique, ARC utilise l'algorithme appelé *RPA positif* qui consiste à ajouter successivement des points dans l'espace de manière aléatoire selon le processus ponctuel utilisé, et à calculer les variations engendrées

sur les grandeurs à estimer. Une moyenne des résultats obtenus sur l'ensemble de l'espace donne l'estimation du gradient.

Les résultats obtenus pour l'estimation du gradient du coût dont nous avons parlé plus haut, permet l'optimisation de ce coût en fonction de l'intensité du processus du niveau intermédiaire. Cette optimisation automatique n'est pas encore complètement implantée dans ARC. La possibilité actuellement offerte à l'utilisateur est de régénérer le niveau intermédiaire du modèle avec une autre intensité de processus en fonction du gradient calculé.

### 3.1.4 Exemples de calcul

Les paramètres du modèle sont :

- espace : (0, 50) x (0, 50)
- intensité du processus au niveau abonné : 10.91
- intensité du processus au niveau hiérarchique 1 : 0.27
- intensité du processus au niveau hiérarchique 2 : 0.018
- coûts de réalisation des noeuds : niveau 1 18, niveau 0 4. Le coût d'un noeud au niveau 2 n'influence pas le calcul du gradient.
- coût par unité de longueur d'une tranchée : 9
- coût par unité de longueur d'un câble : 4

Les suivants sont les résultats de l'estimation de l'espérance et du gradient de quelques caractéristiques d'une cellule ou d'une zone de desserte. Le nombre de commutateurs considérés pour les calculs est 494.

	Mean	Theor. mean	Mean/Theor.mean
Surface	3.571713894003	3.666670333337	0.974102815170
Frontier length	7.530710107113	7.659420691762	0.983195780748
Neighbours	5.967611336032	6.000000000000	0.994601889339

	Gradient	Theor.grad.	Grad/Theor.grad
Surface	-12.417761137227	-13.444471333374	0.923633278640
Frontier length	-12.322017528162	-14.042285310516	0.877493745191
Neighbours	0.311845077197	0.000000000000	

Les suivants sont les calculs successifs du coût dont nous avons parlé plus haut, et de son gradient. On peut observer chaque pas de l'optimisation du coût par rapport à l'intensité du processus ponctuel du niveau hiérarchique 1.

```
-----
Level 1 process intensity: 0.27272700
Considered points:      15 (on the higher hierarchy level)

    MEAN COST:          11961.50352868

    COST GRADIENT :    -7361.35478162
-----
```

```
-----
Level 1 process intensity: 0.50000000
Considered points:      16 (on the higher hierarchy level)

    MEAN COST:          10658.34028128

    COST GRADIENT :    -1401.71686228
-----
```

```
-----
Level 1 process intensity: 0.60000000
Considered points:      16 (on the higher hierarchy level)

    MEAN COST:          10851.15125511

    COST GRADIENT :     523.04066513
-----
```

```
-----
Level 1 process intensity: 0.55000000
Considered space:      [0.00000000,50.00000000]x[0.00000000,50.00000000]
Considered points:      16 (on the higher hierarchy level)

    MEAN COST:          10510.39609885

    COST GRADIENT :    -168.12384335
-----
```

## 3.2 Les objets de base dans ARC

La figure 3 nous montre la hiérarchie des classes d'objets utilisés dans ARC. Les classes d'objets composant un modèle, comme nous l'avons vu plus haut (section 2.2), sont les classes *abonné*, *commutateur*, *lien logique*, *lien physique*, *flux* et *cellule*. Les classes d'objets relatives à l'arbre de Delaunay, ainsi que les classes *liste* et *arbre*, sont utilisées pour la manipulation des objets de base dans un modèle et pour la construction d'un modèle, qui, lui-même, est de la classe *modèle*.

Les attributs d'un modèle dans ARC peuvent être classés en trois groupes. Les attributs propres au modèle, les attributs relatifs au rangement et à la manipulation des objets du modèle, et les attributs graphiques et relatifs à l'interaction avec l'utilisateur. Les attributs propres au modèle sont les suivants.

- Le nombre de niveaux de hiérarchie dans le modèle.
- L'espace dans lequel le modèle est construit. C'est une fenêtre dans un espace à deux dimensions.
- Le type de processus ponctuel associé à chaque niveau hiérarchique. A chaque type correspond une fonction qui est appelée pour la génération des points du niveau en question.
- Les paramètres de ces processus ponctuels.
- Pour chaque niveau, le nombre de connexions d'un point de ce niveau aux points du niveau supérieur (par exemple, ce nombre est 2 pour indiquer qu'un point doit être connecté au premier et au deuxième point le plus proche du niveau supérieur). L'algorithme de pavage de l'espace actuellement implanté dans ARC étant pour le moment de type Voronoï simple (d'ordre 1), une seule connexion au point le plus proche peut être faite. La connexion aux  $N$  points les plus proches nécessite l'implantation des diagrammes de Voronoï d'ordre  $N$ . Nous reviendrons sur ce point dans la section 3.3.

Les autres éléments propres au modèle tels que les points dans chaque niveau et les liens entre ces points sont accessibles par les attributs de rangement ci-après. Les points sont rangés dans trois structures de données différentes, et les liens dans de

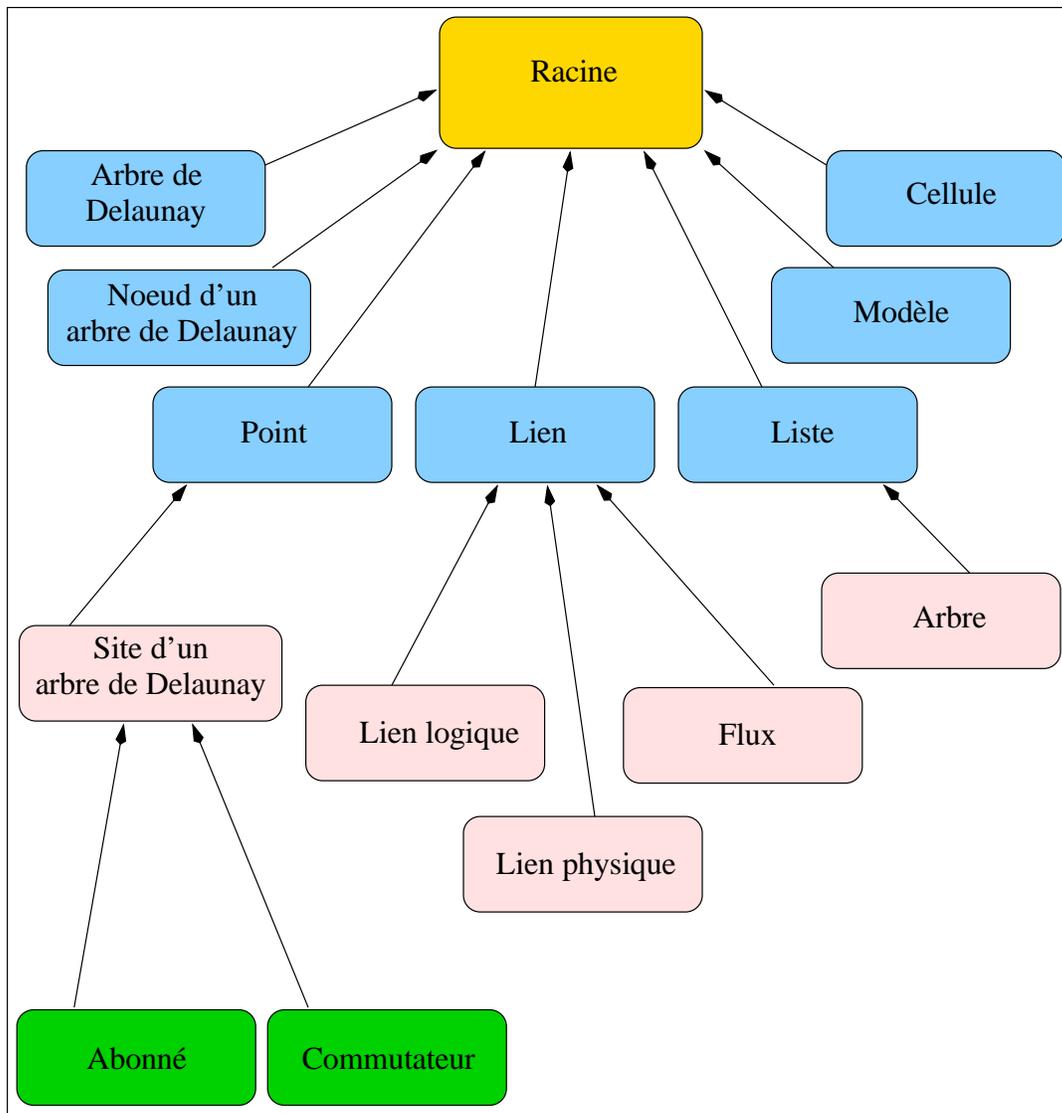


FIG. 3 – *Les objets dans ARC - Hiérarchie des classes*

simples listes. Pour le niveau le plus bas de la hiérarchie dans le modèle, les points peuvent être des abonnés ou des commutateurs. Les attributs de rangement sont:

- La liste des points dans chaque niveau et la liste des liens de différents types.
- L'arbre de Delaunay correspondant à chaque niveau hiérarchique.
- L'arbre quaternaire de repérage spatial des points pour chaque niveau.

### 3.3 Structures de données et algorithmes

Les besoins en accès et en rangement des différents éléments d'un modèle sont les suivants.

- Accès successifs à l'ensemble des points. Une simple liste des points suffit pour cette opération.
- Accès à l'ensemble des points situés dans une fenêtre donnée de l'espace. La structure qui semble adaptée pour cette opération est un arbre quaternaire.
- Accès aux  $n$  points les plus proches d'un point donné, et pavage de l'espace. L'arbre de Delaunay est la structure à utiliser pour ces opérations.

#### 3.3.1 Arbre quaternaire

La structure d'arbre quaternaire utilisée dans ARC est la suivante : un noeud représente une fenêtre de l'espace et les quatre branches d'un noeud représentent quatre subdivisions de cette fenêtre. Une feuille de cet arbre est donc une fenêtre élémentaire. Chaque feuille contient un nombre maximum de points, ce nombre ayant été préalablement évalué pour optimiser le temps d'accès en fonction de la profondeur de l'arbre. Dans chaque feuille, les points sont simplement rangés dans une liste. La subdivision des feuilles de l'arbre en quatre nouvelles feuilles s'effectue au fur et à mesure de la génération des points selon leurs coordonnées en  $x$  et en  $y$ .

### 3.3.2 Arbre de Delaunay

La simulation spatiale d'un modèle repose essentiellement sur cette structure de données. Le modèle de base consiste à connecter un élément (abonné ou commutateur) d'un niveau hiérarchique donné au commutateur le plus proche en distance du niveau supérieur. Le pavage de l'espace sur ce niveau supérieur est ainsi du type *diagramme de Voronoï*. La construction de ce pavage est basée sur la *triangulation de Delaunay* de l'ensemble des éléments, que nous appellerons *sites*, sur ce niveau.

Le diagramme de Voronoï d'un ensemble fini de sites (ici les abonnés ou les commutateurs) dans un espace à deux dimension est la subdivision de cet espace en un ensemble de cellules convexes, chaque cellule étant associée à un site, et tout point dans une cellule est plus proche en distance du site associé que des autres sites. La structure géométrique duale du diagramme de Voronoï est la triangulation de Delaunay, obtenue en traçant une ligne reliant chaque pair de sites dont les cellules sont voisines. L'arbre de Delaunay est une structure contenant la triangulation de l'ensemble des points.

L'algorithme utilisé ici [4] consiste en une construction incrémentale de l'arbre de Delaunay. Les sites sont successivement générés de manière aléatoire et insérés dans l'arbre au fur et à mesure de leur génération, la triangulation étant mise à jour après chaque insertion de site. Nous ne rentrerons pas ici dans les détails de l'algorithme de triangulation qui est bien décrit dans [4], mais nous allons simplement décrire la structure de l'arbre et la manière de le construire.

Un noeud de l'arbre de Delaunay est un triplet de sites formant un triangle appartenant ou non à la triangulation finale. L'arbre, qui est en réalité un graphe orienté sans cycle, contient l'historique des opérations partant du noeud racine (formé par les trois premiers sites) jusqu'à la triangulation finale. A chaque insertion d'un site, des relations *père-fils* sont définies entre les anciens noeuds de l'arbre, qui sont *en conflit* (ne feront plus partie de la triangulation finale) ou non avec le nouveau site, et les nouveaux noeuds créés par l'insertion. Les noeuds en conflit avec un nouveau site sont ceux dont le cercle circonscrit (passant par les trois sites du triplet) contient le nouveau site. La sauvegarde et les relations définies entre les triplets successifs permettent à cet algorithme de localiser très rapidement les noeuds en conflit, et donc d'insérer rapidement le nouveau site.

La figure 4 montre un exemple de triangulation de Delaunay ainsi que le pavage de Voronoï associé pour un niveau hiérarchique d'un modèle.

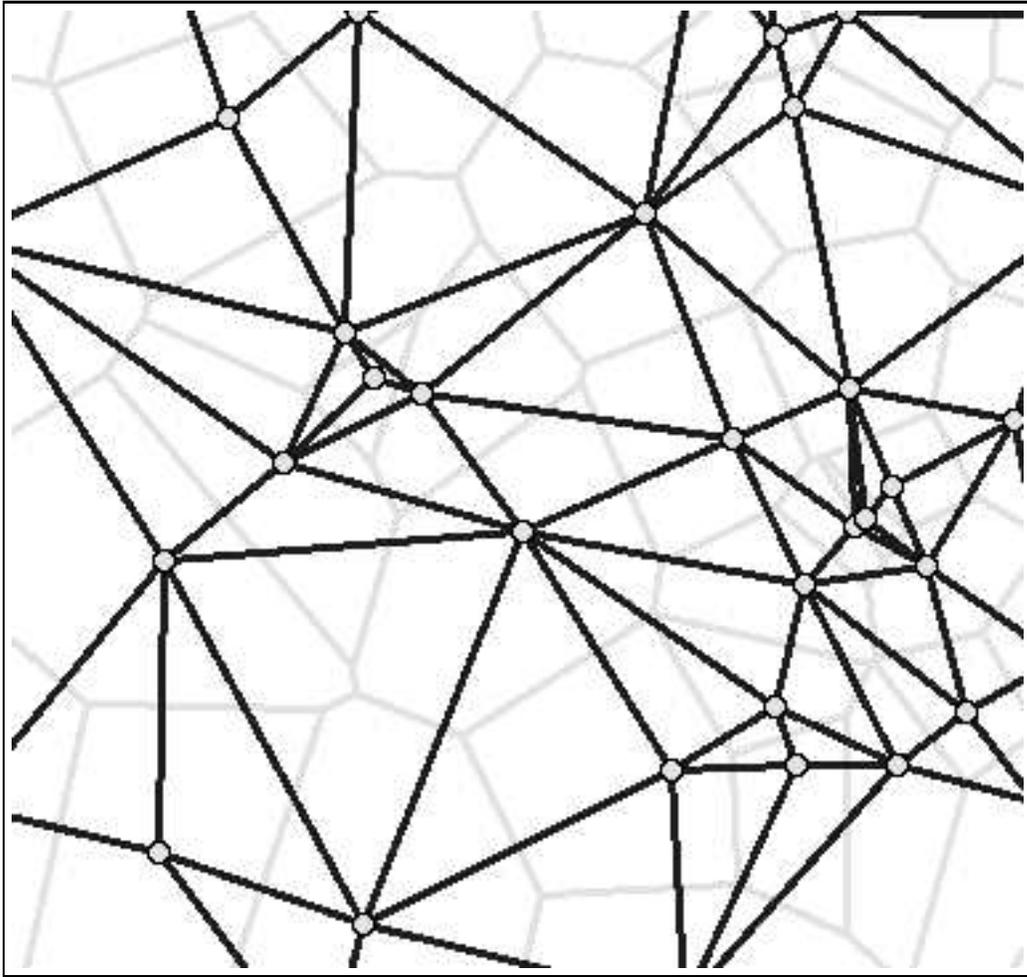


FIG. 4 – Le pavage de type “cellules de Voronoï” (en clair) d’un niveau hiérarchique, et la “triangulation de Delaunay” associée (en foncé).

### 3.4 Les évolutions de ARC

L'approche objet utilisée pour la conception de ARC facilite son extension dans le futur, et les limitations actuelles pourront ainsi être supprimées facilement. Des nouvelles classes d'objets peuvent, soit être créées et ajoutées au système, soit hériter directement des objets déjà existants comme nous le montre la figure 3.

En ce qui concerne les objets de base, les commutateurs peuvent par exemple être différenciés entre eux en créant différentes sous-classes de la classe *commutateur*; il en est de même pour les abonnés, les différents liens et les cellules. Les attributs et les méthodes fournis par chacun des ces objets peuvent ainsi être enrichis et correspondre à l'évolution souhaitée du système, par exemple pour prendre en compte tout ce qui concerne les mobiles dans un modèle.

Les structures de données utilisées dans ARC peuvent être remplacées facilement à condition que les nouvelles classes de structures de données fournissent les mêmes attributs et les mêmes méthodes (éventuellement enrichies) que les anciennes classes. Par exemple, l'arbre quaternaire utilisé actuellement peut être remplacé par une autre structure plus adaptée si la densité des points dans l'espace n'est plus uniforme comme c'est le cas dans les modèles actuels que ARC manipule. En effet, si les points ne sont pas répartis de manière uniforme dans l'espace, l'arbre peut avoir besoin d'un équilibrage fréquent de profondeurs, une opération pouvant être coûteuse.

L'arbre de Delaunay peut être remplacé par une structure plus générale qu'on peut appeler *arbre de Delaunay d'ordre k*, par dualité avec *diagramme de Voronoï d'ordre k*, permettant de réaliser des connexions d'un commutateur de niveau hiérarchique donné aux  $k$  commutateurs les plus proches du niveau hiérarchique supérieur. Cet arbre peut également être remplacé par une toute autre structure permettant un autre pavage de l'espace tel que le diagramme de Puissance.

## 4 Conclusion

Nous avons décrit dans ce rapport le simulateur spatial de réseaux de communications nommé ARC. C'est un simulateur basé sur des algorithmes de géométrie aléatoire. Les buts fixés pour la réalisation de ce prototype ont été atteints:

- Pouvoir expérimenter les résultats théoriques déjà obtenus dans le cadre de la convention de recherche entre le CNET et l'INRIA. Différentes fonction-

nelles des processus ponctuels de définition d'un modèle peuvent être estimées ainsi que leurs gradients. Des optimisations de certains paramètres du modèle peuvent être faites en utilisant les résultats de ces estimations.

- Réaliser un système de structure facilement extensible et adaptable pour la suite des travaux, aussi bien dans le cadre de cette convention qu'indépendamment pour éventuellement en faire un produit réellement utilisable par la suite. Nous avons énuméré quelques exemples d'extensions et d'améliorations de ARC qui peuvent être facilement mise en place.

---

## Références

- [1] F. Aurenhammer. *Voronoi Diagrams: A Survey of a Fundamental Geometric Data Structure*.  
ACM Computing Surveys, 23:345-405, 1991.
- [2] F. Baccelli - M. Klein - M. Lebourges - S. Zuyev. *Géométrie Aléatoire et Architecture de Réseaux de Communication*.  
Rapport INRIA n. 2542, Mai 1995. *A paraître Annales des Télécommunications*, 1996.
- [3] F. Baccelli - M. Klein - S. Zuyev. *Perturbation Analysis of Functionals of random measure*.  
Advances in Applied Probability, 27:306-325, 1995.
- [4] J.-D. Boissonnat - M. Teillaud. *On the Randomized Construction of the Delaunay Tree*.  
Theoretical Computer Science 112 (1993) 339-354, Elsevier.
- [5] S. Näher. *The LEDA User Manual*.  
Max-Planck Institut für Informatik - 66123 Saarbrücken, Germany.

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Les objets de l'étude et les modèles de liaisons</b>	<b>3</b>
2.1	Le modèle . . . . .	3
2.1.1	Raccordement des abonnés . . . . .	4
2.1.2	Réseau des commutateurs . . . . .	4
2.2	Les objets de l'étude . . . . .	5
2.2.1	Les abonnés . . . . .	5
2.2.2	Les commutateurs . . . . .	6
2.2.3	Les liens physiques . . . . .	6
2.2.4	Les liens logiques . . . . .	7
2.2.5	Les flux de communications . . . . .	7
2.2.6	Les cellules ou zones de desserte des commutateurs . . . . .	8
2.3	Contraintes de liaisons des objets . . . . .	8
2.4	Les caractéristiques intéressantes d'un modèle . . . . .	10
<b>3</b>	<b>Le prototype ARC</b>	<b>10</b>
3.1	Les fonctionnalités de ARC . . . . .	10
3.1.1	Construction d'un modèle . . . . .	11
3.1.2	Visualisation graphique du modèle . . . . .	12
3.1.3	Les calculs sur le modèle . . . . .	12
3.1.4	Exemples de calcul . . . . .	14
3.2	Les objets de base dans ARC . . . . .	16
3.3	Structures de données et algorithmes . . . . .	18
3.3.1	Arbre quaternaire . . . . .	18
3.3.2	Arbre de Delaunay . . . . .	19
3.4	Les évolutions de ARC . . . . .	21
<b>4</b>	<b>Conclusion</b>	<b>21</b>



---

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,  
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY  
Unité de recherche INRIA Rennes, Irista, Campus universitaire de Beaulieu, 35042 RENNES Cedex  
Unité de recherche INRIA Rhône-Alpes, 46 avenue Félix Viallet, 38031 GRENOBLE Cedex 1  
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex  
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

---

Éditeur  
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)  
ISSN 0249-6399