



Un flot de conception Pour Réseaux de Capteurs sans fil: Modèles d'exécution et Génération de code

Bernard Pottier

Université de Brest (UBO), France

**LabSTICC, UMR 3192*

Dept Informatique, Faculté des Sciences

Plan

En appui sur la conférence de Lundi 25, on montre l'ossature des outils NetGen

1. Forme textuelle du modèle topologique
2. Création interactive de réseaux
3. Principe de la génération de processus communicants
4. Principe de la génération de code CUDA

→ ***Démonstrations sur machine / discussions***

Framework en développement

- Outil de développement principal : Smalltalk
 - VisualWorks (Cincom) : prototypage rapide, constructions de modèles. Licence non commerciale
- Pour l'expression et la simulation systèmes
 - Occam, compilateur de U. de Kent, libre
 - CUDA, compilateur et kit nvcc de Nvidia, licence
 - Graphviz, dot, etc..
- Matériel principal : msp430, mspgcc4, insight

A. Proposition de flot de conception (rappel)

L'atelier NetGen propose un flot descendant inspiré de la conception matérielle:

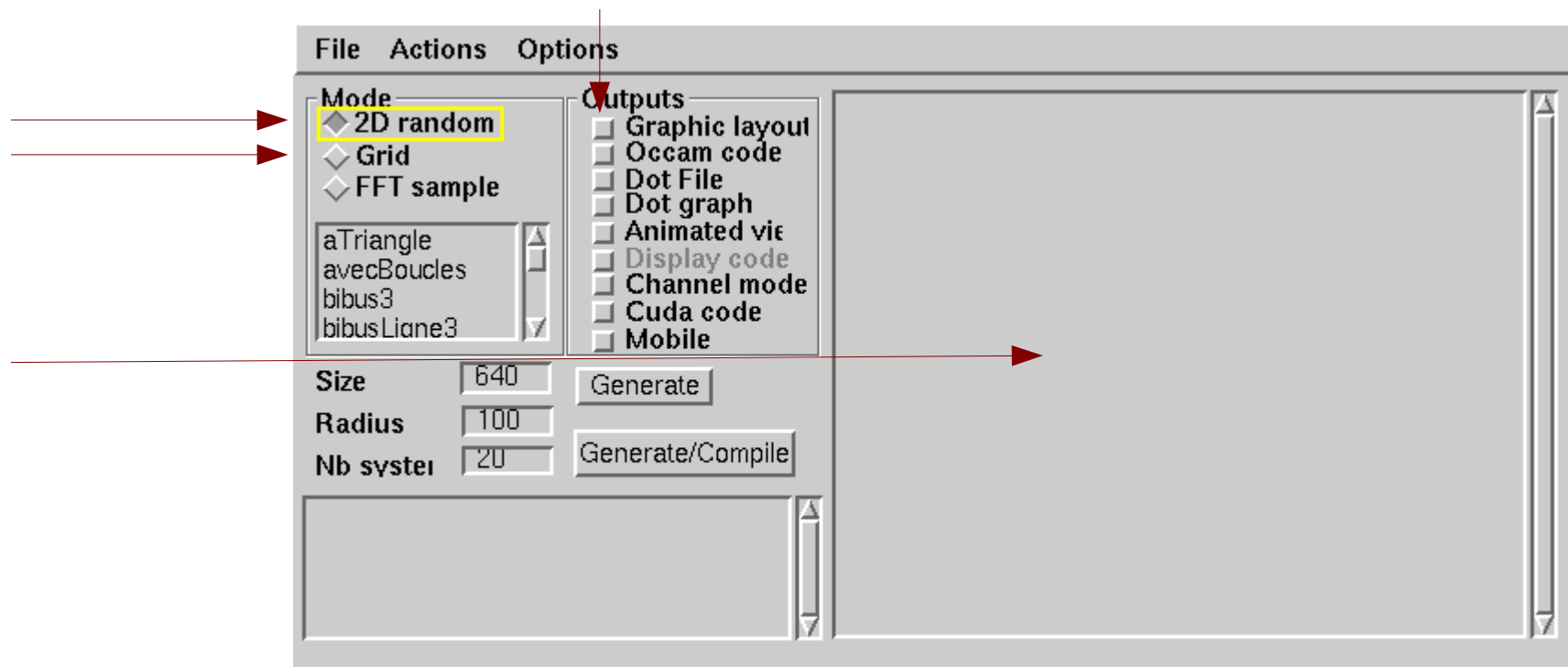
- **Etape 1: conception, simulation, optimisation.**
- **Etape 2 : production du code pour les capteurs, tests et déploiement**

Bouclages sur chacune des étapes et sur le flot complet.

Spécification géométrique (1)

- **Tableau de bord NetGen :**

- Générations de distributions aléatoires
- Réseaux réguliers
- Explications sur la forme textuelle du modèle : $P_i \{ P_j P_k P_n \}$ Programme



Spécification géométrique (2)

File Actions Options

Mode

- 2D random
- Grid
- FFT sample

Outputs

- Graphic layout
- Occam code
- Dot File
- Dot graph
- Animated vic
- Display code
- Channel mode
- Cuda code
- Mobile

Size: 400

Radius: 200

Nb system: 5

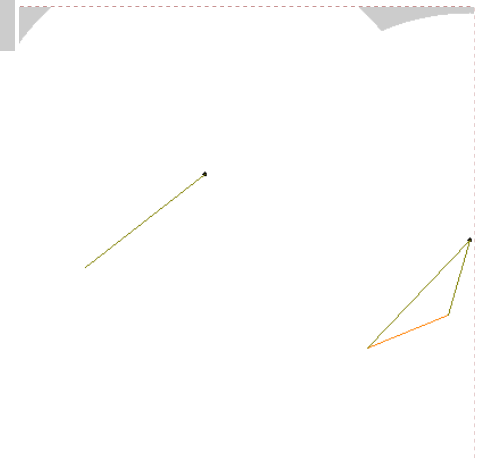
Generate

Generate/Compile

```
genRange200Points5
messages none defined.
P1 { P4, P5 } Node (377 @ 271) (200)
P2 { P3 } Node (58 @ 229) (200)
P3 { P2 } Node (163 @ 147) (200)
P4 { P1, P5 } Node (306 @ 300) (200)
P5 { P1, P4 } Node (396 @ 205) (200)
```

genRange200Points5
processus : 5
min fanout : 1

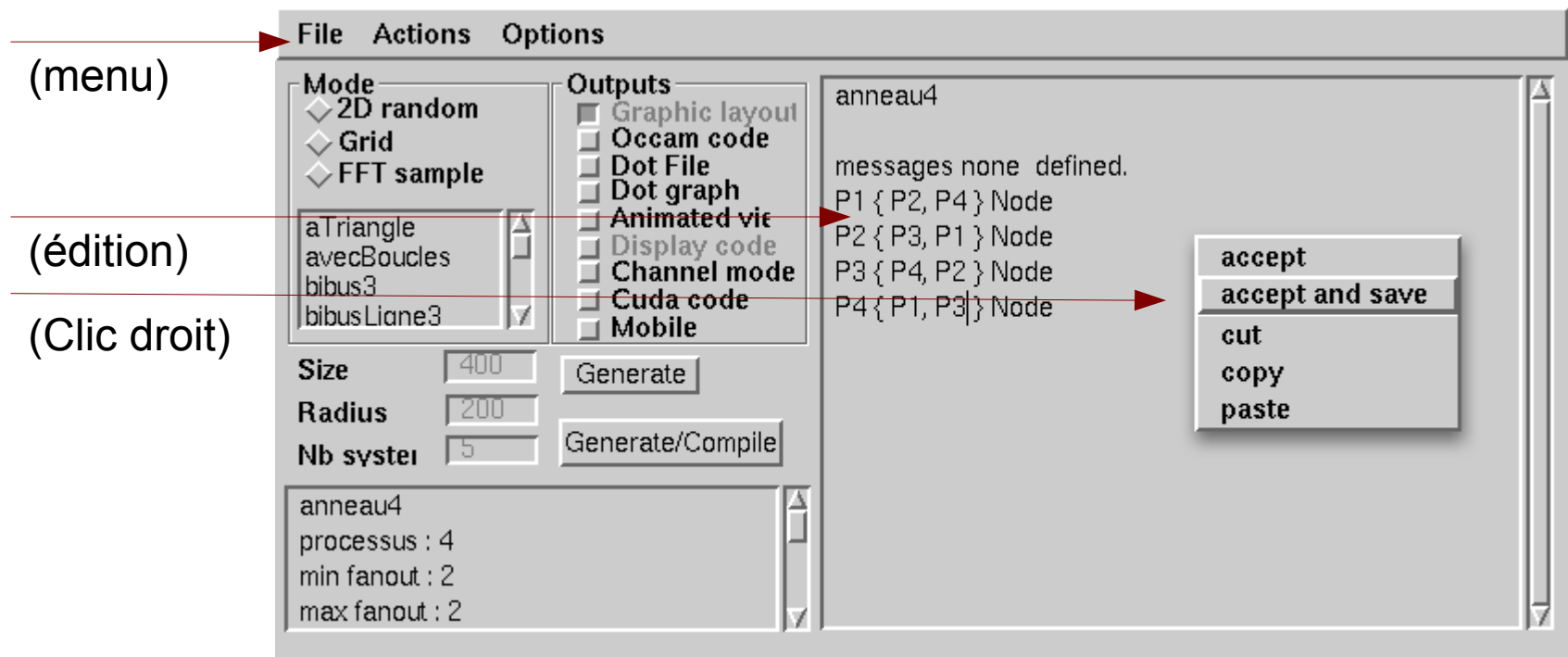
Modèle



Spécification textuelle / fichiers

- **Editeur de spécification**

- Syntaxe spécification : titre messages ($P_i \{ P_j P_k P_n \}$ Programme) ..
- *Load et save* : possibilité d'écrire vos propres générateurs.



Spécifications régulières

- Grilles de connectivité variable

File Actions Options

Mode

- 2D random
- Grid
- FFT sample

anneau4
aTriangle
avecBoucles
bibus3

Outputs

- Graphic layout
- Occam code
- Dot File
- Dot graph
- Animated vie
- Display code
- Channel model
- Cuda code
- Mobile

Size: 200

Radius: 200

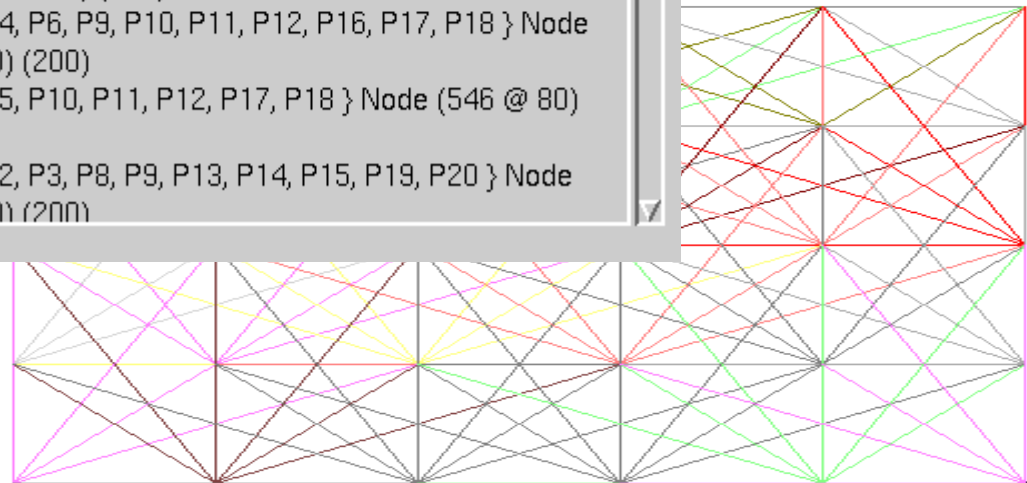
Nb system: 30

Generate

Generate/Compile

```
gridRange200Points30
processus : 30
min fanout : 7
```

```
gridRange200Points30
messages none defined.
P1 { P2, P3, P7, P8, P9, P13, P14 } Node (91 @ 80) (200)
P2 { P1, P3, P4, P7, P8, P9, P10, P13, P14, P15 } Node
(182 @ 80) (200)
P3 { P1, P2, P4, P5, P7, P8, P9, P10, P11, P14, P15, P16 }
Node (273 @ 80) (200)
P4 { P2, P3, P5, P6, P8, P9, P10, P11, P12, P15, P16, P17 }
Node (364 @ 80) (200)
P5 { P3, P4, P6, P9, P10, P11, P12, P16, P17, P18 } Node
(455 @ 80) (200)
P6 { P4, P5, P10, P11, P12, P17, P18 } Node (546 @ 80)
(200)
P7 { P1, P2, P3, P8, P9, P13, P14, P15, P19, P20 } Node
(91 @ 160) (200)
```



Présentation graphe logique

- Utilisation du logiciel graphviz
 - Ré-écriture du modèle en syntaxe *dot*
 - Si graphviz : production de présentations graphiques logiques

The screenshot shows the Graphviz software interface. The 'Mode' section has '2D random', 'Grid', and 'FFT sample' options. The 'Outputs' section has 'Graphic layout', 'Occam code', 'Dot File', 'Dot graph', 'Animated vie', 'Display code', 'Channel mode', 'Cuda code', and 'Mobile' options. The 'Dot graph' option is selected. The 'Size' is set to 400, 'Radius' to 200, and 'Nb system' to 5. The 'Generate' and 'Generate/Compile' buttons are visible. The main window displays the following dot language code:

```
anneau4
messages none defined.
P1 { P2, P4 } Node
P2 { P3, P1 } Node
P3 { P4, P2 } Node
P4 { P1, P3 } Node
```

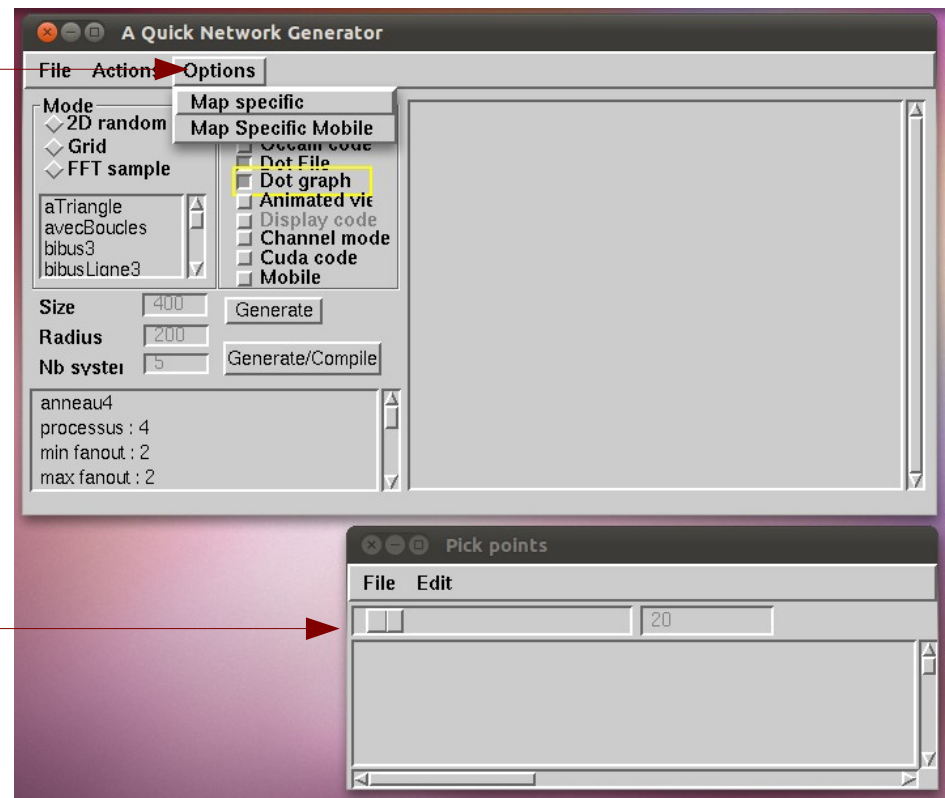
A context menu is open over the code, showing options: 'accept', 'accept and save', 'cut', 'copy', and 'paste'. Below the code, a graph visualization shows four nodes (P1, P2, P3, P4) connected by directed edges. The edges are labeled with pairs of node IDs: P1.P2, P2.P3, P3.P4, P4.P3, P1.P4, P4.P1, P2.P1, and P3.P2.

Saisie sur carte (1)

- **But : positionnement de capteurs sur un plan**
 - Accès fenêtre de saisie graphique

Menu option : map

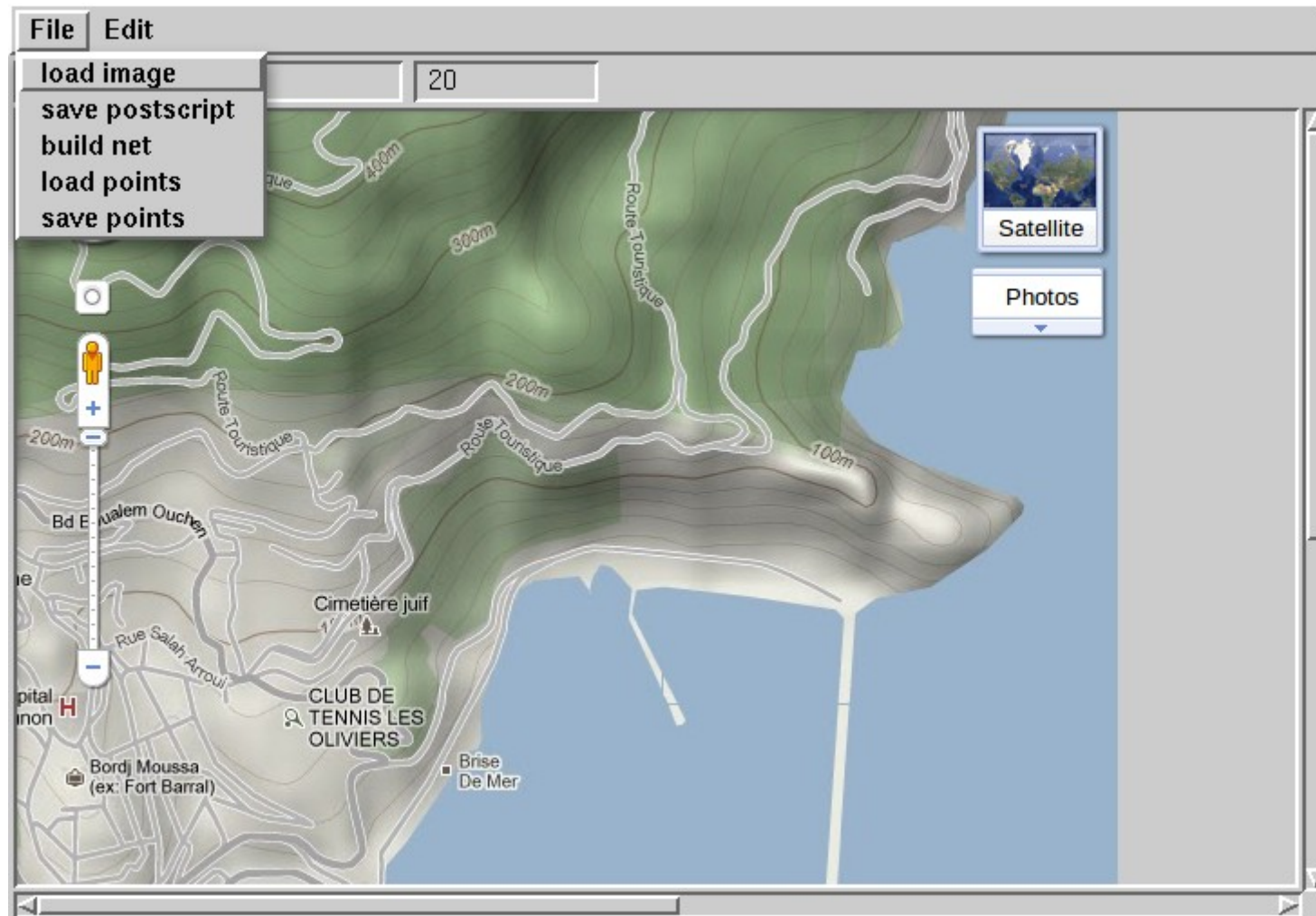
Fenetre graphique



Saisie sur carte (2)

Chargement d'une image (.png)

- Portail IGN (France), Google maps, etc... plans batiments



Saisie sur carte (3)

Positions des capteurs

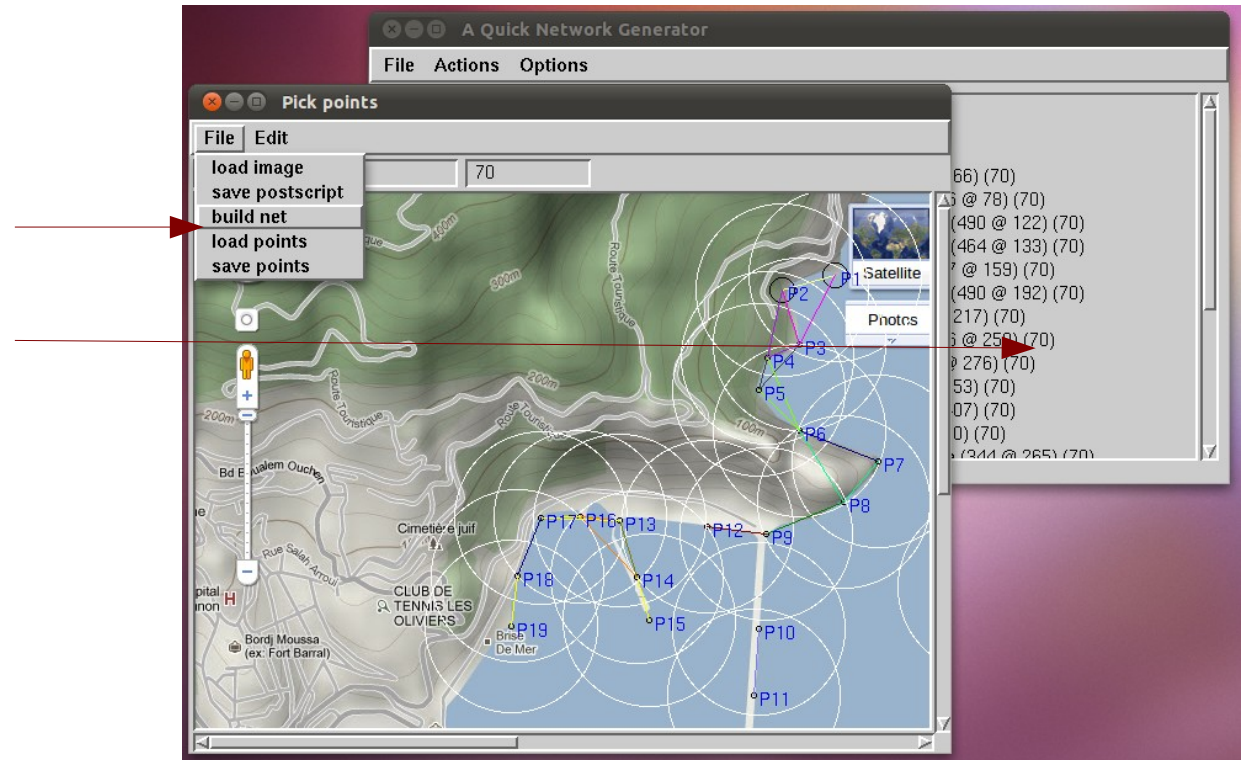
- Selections de points
- Echelle en vue
- Dessin du réseau immédiat
- Portée des capteurs réglable



Saisie sur carte (4)

Production du modèle

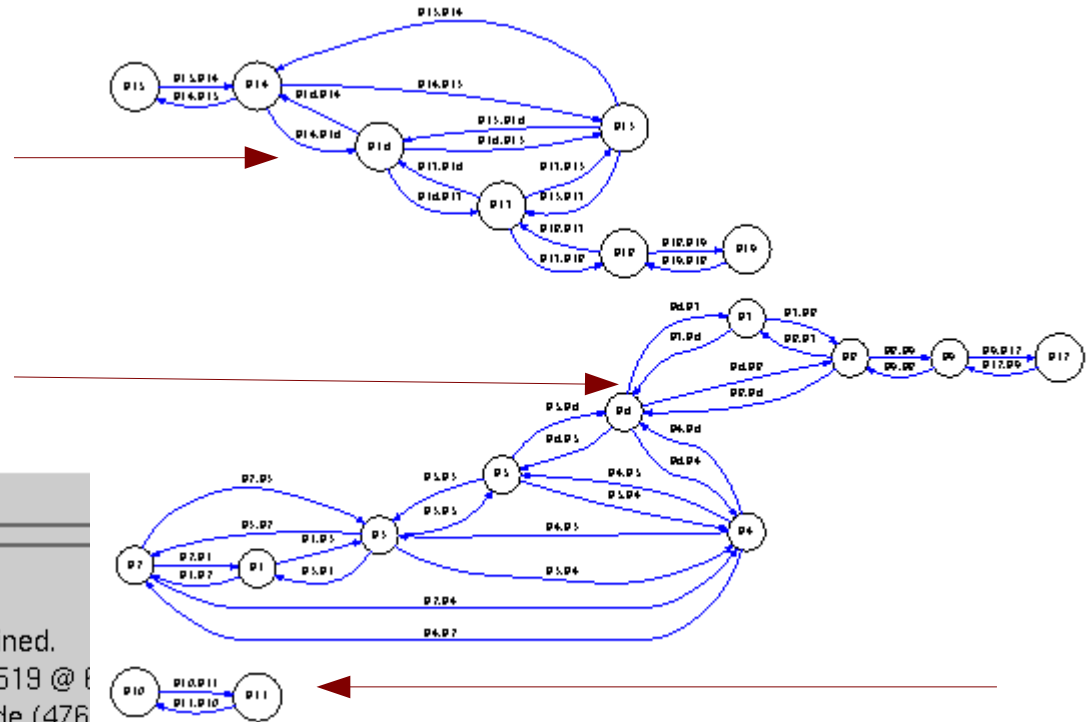
- Transfert vers les outils du modèle
- Production d'un modèle annoté sur la carte



Saisie sur carte (5)

Analyse du modèle

- Application des outils sur le modèle produit



File Actions Options

Mode

- 2D random
- Grid
- FFT sample

aTriangle
avecBoucles
bibus3
bibusLiane3

Size:

Radius:

Nb system:

Outputs

- Graphic layout
- Occam code
- Dot File
- Dot graph
- Animated vie
- Display code
- Channel mode
- Cuda code
- Mobile

portDeBejaia

messages none defined.

P1 { P2, P3 } Node (519 @ 122) (70)

P2 { P1, P3, P4 } Node (476 @ 122) (70)

P3 { P1, P2, P4, P5 } Node (490 @ 122) (70)

P4 { P2, P3, P5, P6 } Node (464 @ 133) (70)

P5 { P3, P4, P6 } Node (457 @ 159) (70)

P6 { P4, P5, P7, P8 } Node (490 @ 192) (70)

P7 { P6, P8 } Node (554 @ 217) (70)

P8 { P6, P7, P9 } Node (526 @ 250) (70)

P9 { P8, P12 } Node (463 @ 276) (70)

P10 { P11 } Node (457 @ 353) (70)

P11 { P10 } Node (453 @ 407) (70)

P12 { P9 } Node (415 @ 270) (70)

P13 { P14, P16, P17 } Node (344 @ 265) (70)

portDeBejaia

processus : 19

min fanout : 1

max fanout : 4

Autre technique : saisie par GPS

- **'Data logger' MerSea**
 - Micro controleur + GPS, boutons de saisie
 - Mesures de portée sans fil
 - Promenade sur site
 - Analyse en laboratoire

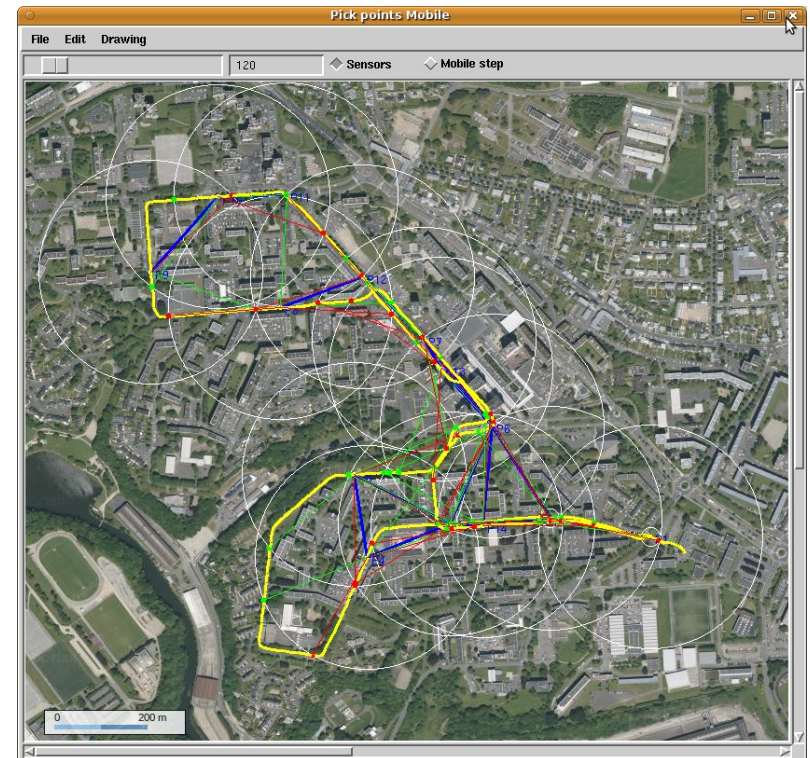


GPS

C2420 MSP

Saisie par GPS : ré-écritures

- **Analyse des données et reformulation**
 - KML / Google map
 - Qgis (apt-get install qgis)



B. Proposition de flot de conception (rappel)

L'atelier *NetGen* propose un flot descendant inspiré de la conception matérielle:

- **Etape 1:** conception, simulation, optimisation.
- **Etape 2 :** production du code pour les capteurs, tests et déploiement

Bouclages sur chacune des étapes et sur le flot complet.

Flot de conception 1: simulation

- **Etape 1, simulation :**
 - 1.Capture de topologie** : sur ordinateur (cartes, dispersions) ou dans la rue (data logger)
 - 2.Synthèse réseau** : exprimée sur un modèle, avec production de caractéristiques.
 - 3.Ré-écritures** graphiques ou textuelles
 - 4.Couplage spécification comportements**
 - 5.Production code de simulation et simulation**

Processus communicants : CSP/Occam ... (Ref. Hoare)

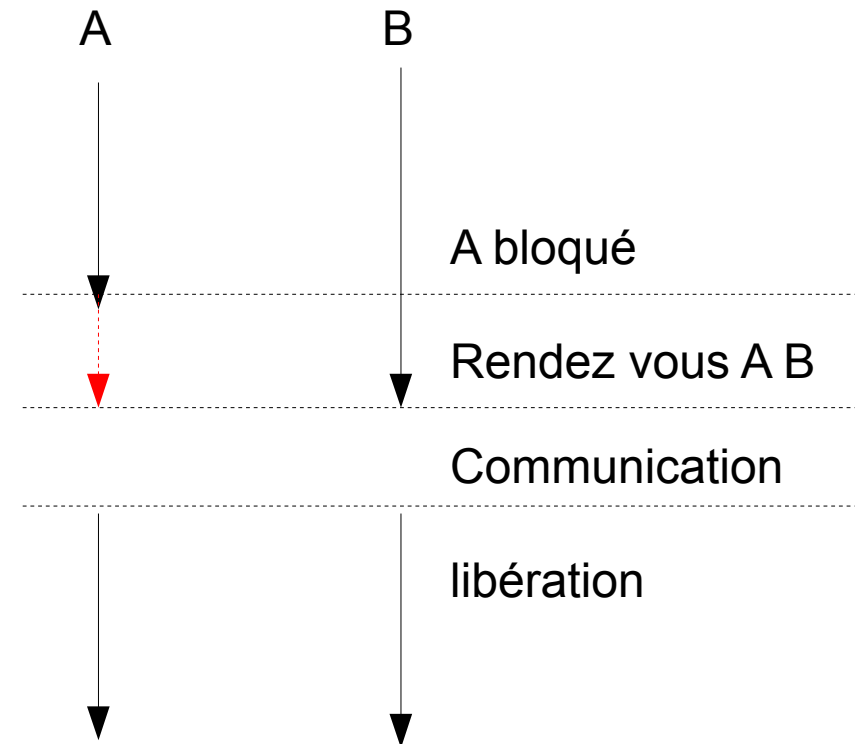
CHAN OF BYTE c :

BYTE val :

PAR

c ! 'x' – ecriture A

c ? val – lecture B



Processus communicants : Modèle synchrone

- Si P1 et P2 sont connectés :

- P1 écrit à P2
- P1 reçoit de P2
- P2 écrit à P1
- P2 reçoit de P1

- Obligation de procéder en parallèle pour éviter un deadlock !

Echantillon de code pour les phases M_i N_i du modèle synchrone

```
PAR
  PAR i=0 FOR SIZE in
    in[i] ? CASE
      table ; inMessages[i]
      SEQ
        tags[i]:=TRUE
        externalChange :=TRUE
      null ; nullByte
        tags[i]:=FALSE
  PAR j=0 FOR SIZE out
    out[j] ! table; outMessages[j]
```

Processus communicants : Simulation

- 100 % de la topologie générée
 - Construction parallèle pour le réseau
 - Présence d'un traceur interne
- 100 % du comportement à écrire
 - Largement réutilisable
 - Code de haut niveau, fortement concurrent (PAR)
- Compilation multi_threadée adaptée aux processeurs multi coeurs
 - Scheduler embarqué dans l'exécutable !

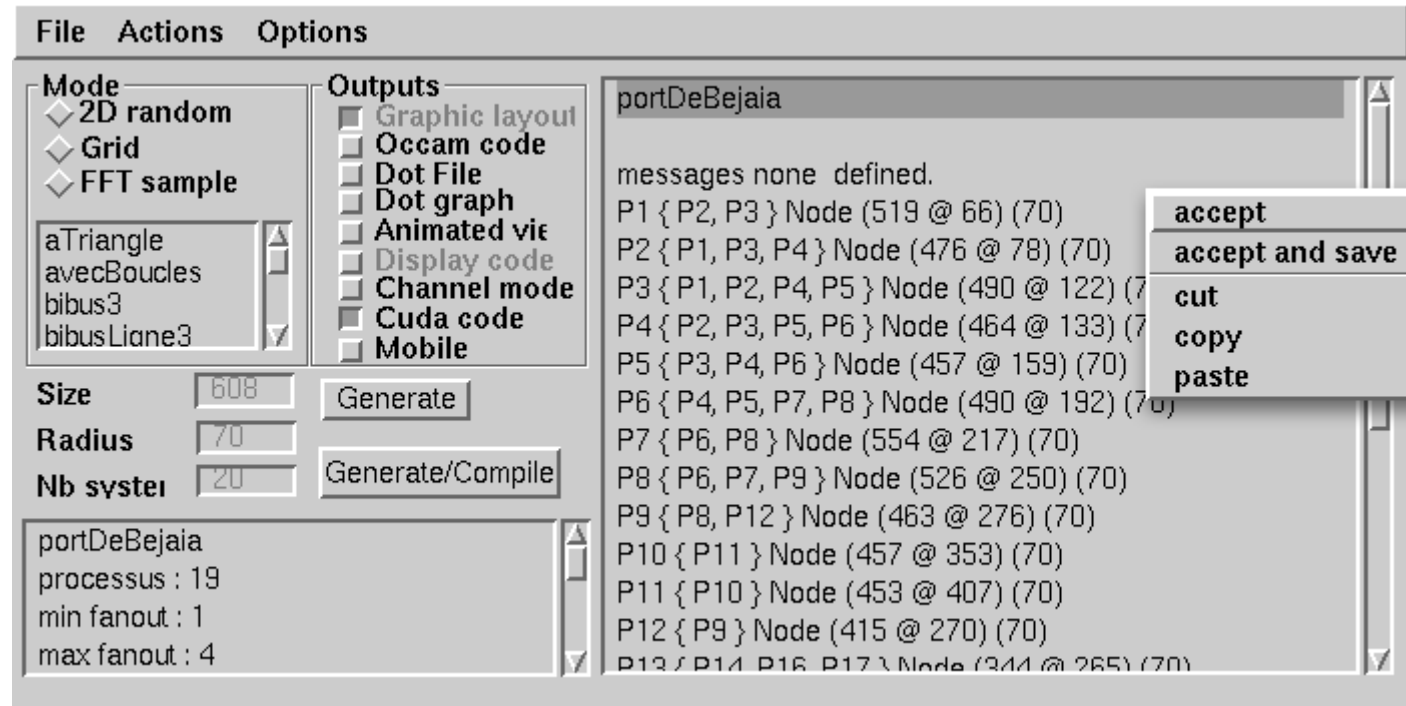
Simulation sur GPU

- 1 *thread* par processus
 - 1 *kernel* par comportement différent
 - Parallélisme *implicite*, tout le monde fait la même chose
- Les canaux sont implantés dans des tableaux, les transferts sont 'synchrones'
 - Contenant les valeurs émises
 - Qui sont lus par les voisins

Simulation sur GPU

```
typedef struct s_mapped
{
    int node;
    int canal;
}mapped;
```

```
typedef struct s_canaux
{
    int nbOut;
    int nbln;
    int nbDyn;
    mapped write[MAX_FANOUT];
    mapped read[MAX_FANOUT];
    mapped
writeDyn[DYNAMIC_CHAN];
    mapped readDyn[DYNAMIC_CHAN];
}canaux;
```



```
canaux channels_h[] =
{ // start array
    {2,2,5,{{0,0},{0,1}},{{1,0},{2,0}},{{0,0},{0,1},{0,2},{0,3},{0,4}},{{-1,-1},
{-1,-1},{-1,-1},{-1,-1},{-1,-1}},
    {3,3,5,{{1,0},{1,1},{1,2}},{{0,0},{2,1},{3,0}},{{1,0},{1,1},{1,2},{1,3},
{1,4}},{{-1,-1},{-1,-1},{-1,-1},{-1,-1},{-1,-1}},
    {4,4,5,{{2,0},{2,1},{2,2},{2,3}},{{0,1},{1,1},{3,1},{4,0}},{{2,0},{2,1},
{2,2},{2,3},{2,4}},{{-1,-1},{-1,-1},{-1,-1},{-1,-1},{-1,-1}}},
```

Simulation sur GPU

- Production d'une ossature décrivant les communications
- Compilation nvcc
- Exécutable ou librairie dynamique
- Intégration au contrôleur Smalltalk pour la visualisation

Merci ..

Flot de conception 2: synthèse

- **Etape 2, synthèse 'travaux en cours):**
 - 1.Code de communication sans fil**
 - 2.Appel des interactions avec les capteurs
 - 3.Planification des opérations
(ordonnancement)



Partie 3

Illustration du flot

Capture de la topologie (1)

- **Sur ordinateur :**
 - Cartes, placements, information géographique

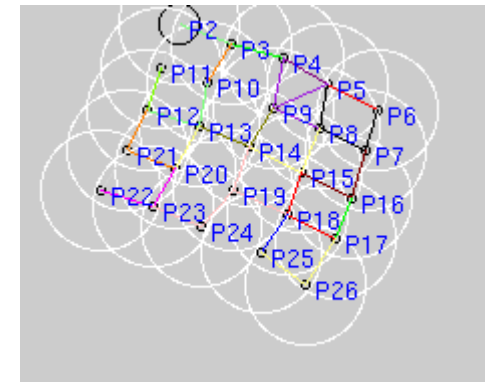
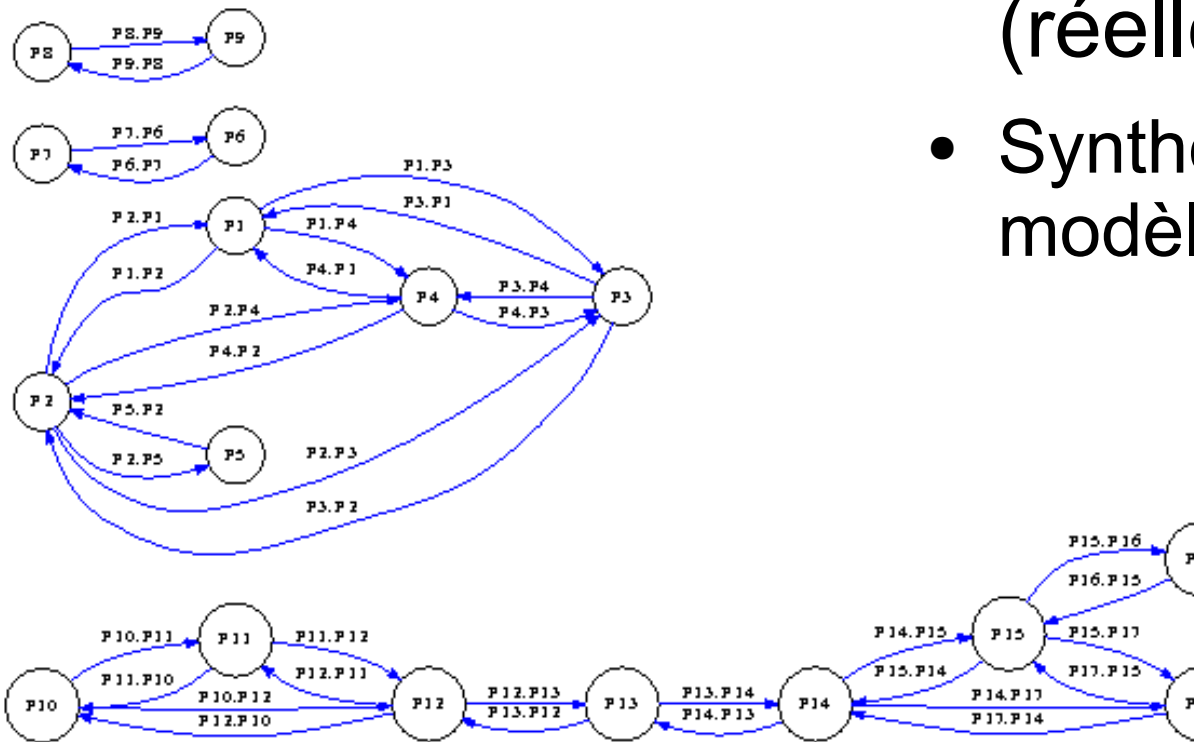


Capture de la topologie (2)

- **Dans la rue :**
 - Mersea data logger (GPS, carte MSP430)
 - Visite ligne de bus, saisie des points d'intérêt (PoI)

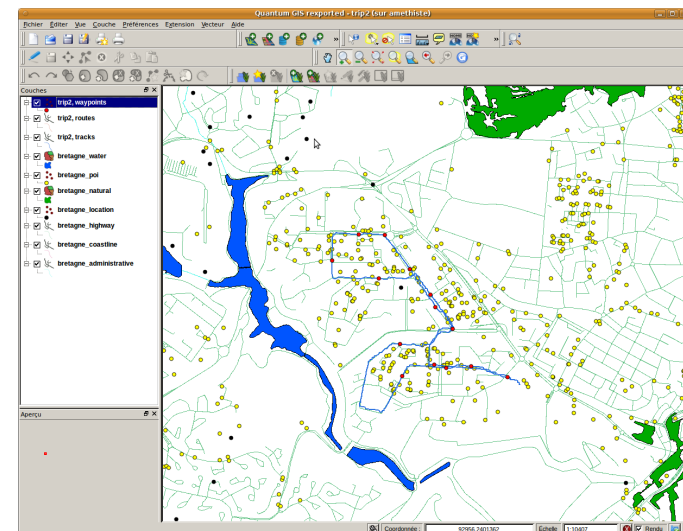
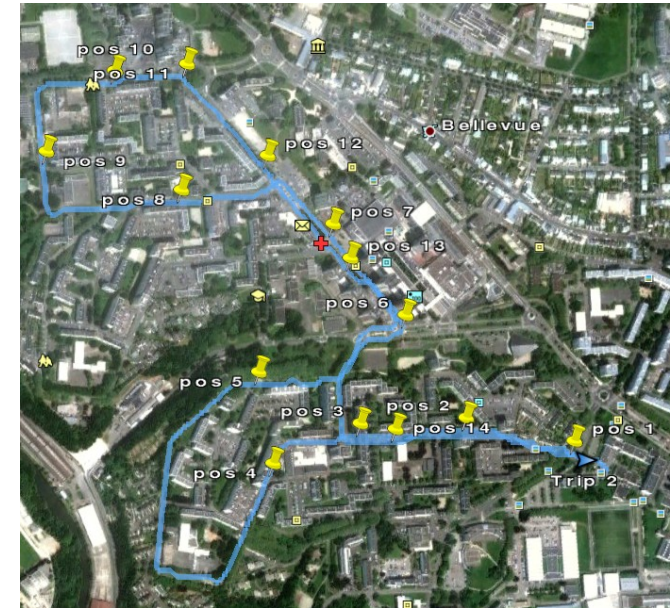
Synthèse du réseau

- **Calcul connectivité :**
 - Aires de couvertures (réelles, ou attendues)
 - Synthèse d'un graphe sur un modèle interne



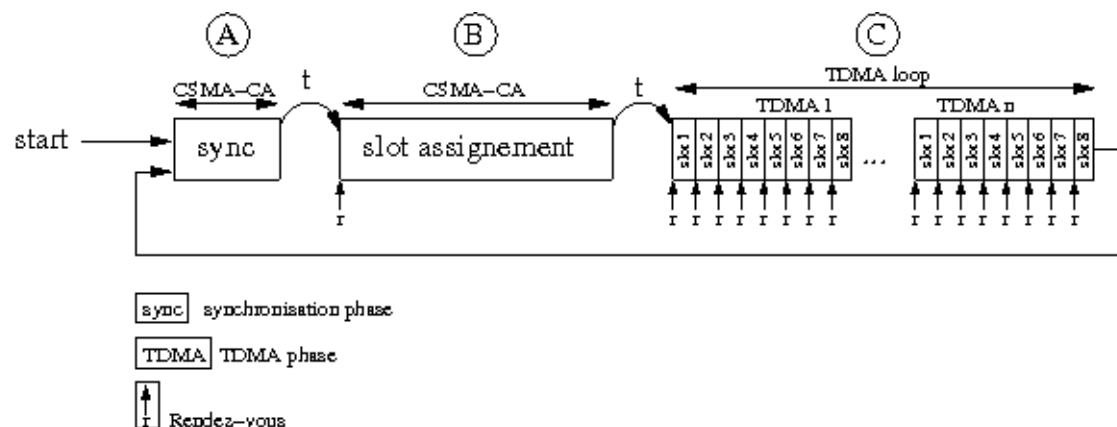
Statistiques et interactions

- **Statistiques :**
 - Noeuds, connexions, réseaux
 - Max fanout, ...
- **Réécriture du graphe :**
 - KML/Google earth
 - QGIS (Quantum GIS)
 - Graphviz, ...



Exécution par échanges TDMA

- TDMA : rendez-vous périodiques entrelacés avec le sommeil et l'observation
- Allocation de slots pour diffusion et réception
- Référence temporelle par protocole asynchrone
- 1 time slot = 1 channel + n slots pour mobiles





Partie 4

Simulation Parallèle

Simulation par système de processus

- **Synthèse de l'organisation en OCCAM :**
 - Construction parallèle de processus **PAR**
 - 1 lien de communication est 1 canal Occam
 - Multiplexeur pour la collecte de trace
- **Comportement :**
 - Le fichier topologique inclut un fichier de comportement réutilisable

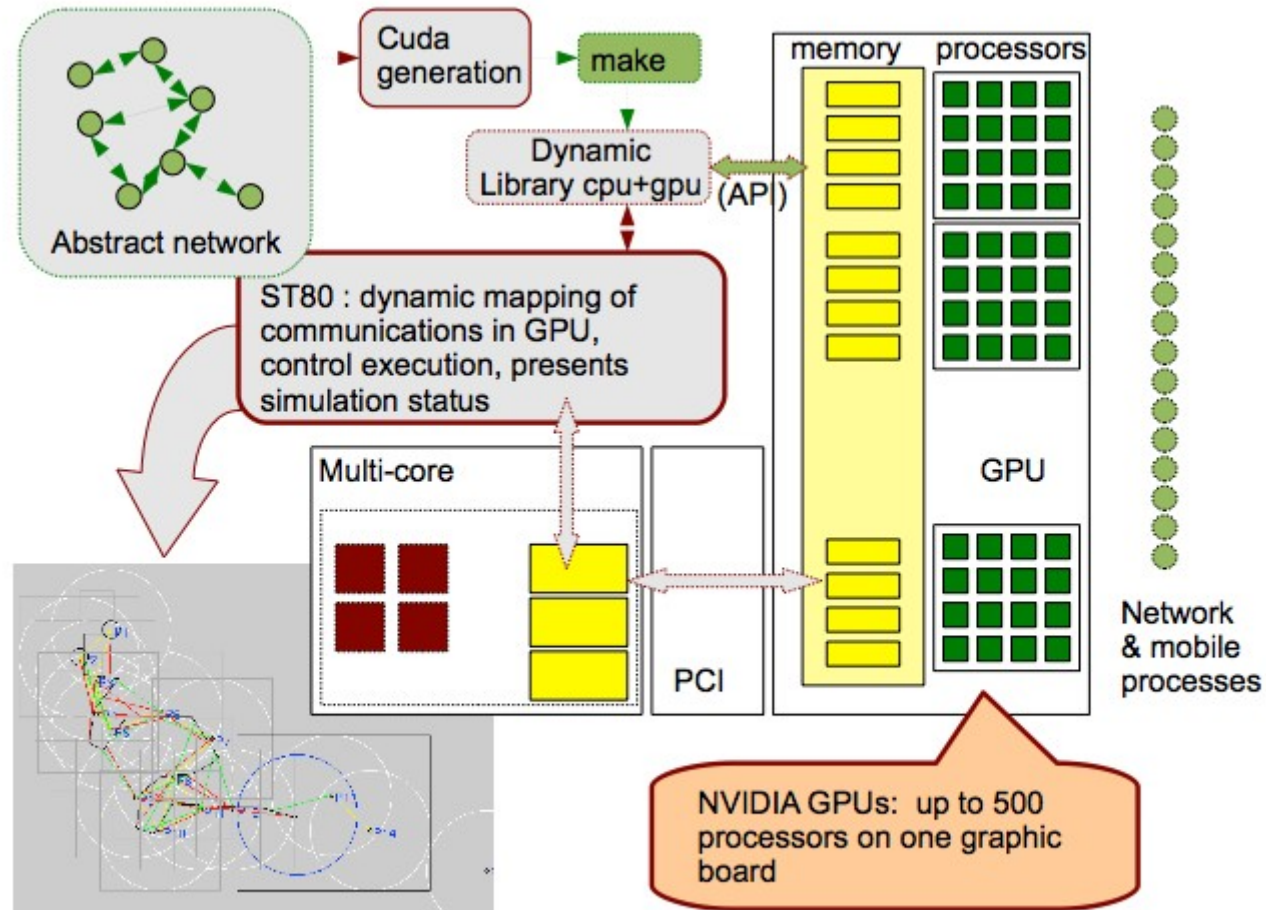
Simulation/MIMD

- **Compilation et exécution:**
 - KROC compiler, i386, multi threaded
 - Collecte de la trace dans un fichier
- **Resultats :**
 - Diametre, leader, arbres de recouvrement, calculs de routages et routages, contentions..
 - Simulation jusque 1000 noeuds

Simulation sur GPU

- **Processus décrits par des thread CUDA:**
 - Les canaux sont interprétés par read/write sur des buffers en mémoire partagée.
 - Un contrôleur peut accéder à l'état des processus et aux buffers
 - Intrelacement contrôle et présentation
- **Comportement :**
 - Code CUDA exprimant un programme synchrone.

GPU execution diagram





Partie 5

Mobiles

Mobiles : classification d'usages

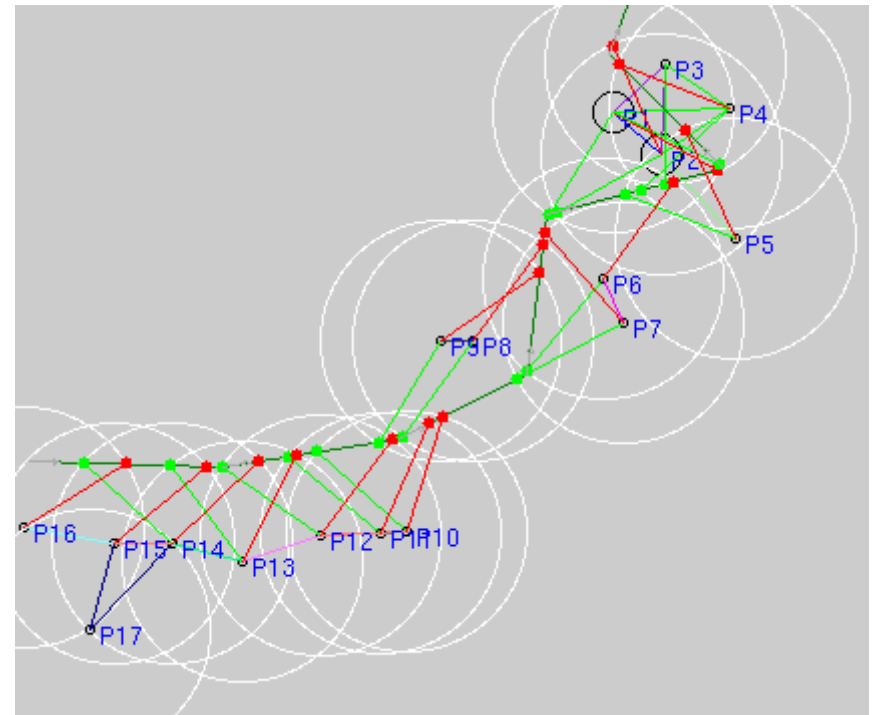
- 1 mobile : *vehicule, robot, drone, personne*
- Plusieurs mobiles : *transports, animaux, touristes*
- Flottes de Mobiles : *vehicules, circulation...*

- Trajectoire calculée: *balayage d'un espace*
- Trajectoire spécifiée : *bus sur sa ligne*
- Trajectoire dynamique : *decisions prises au cours du trajet*

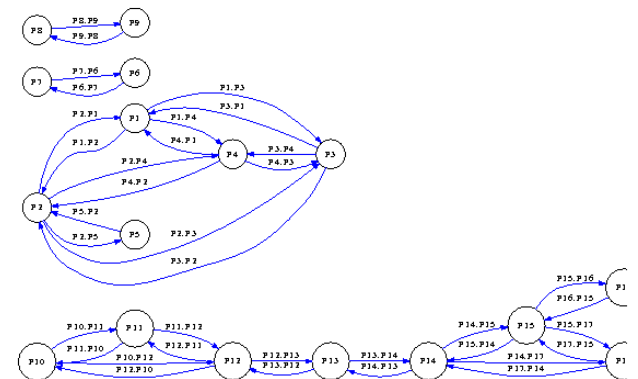
Visite spécifiée



Range 250m, 4 networks



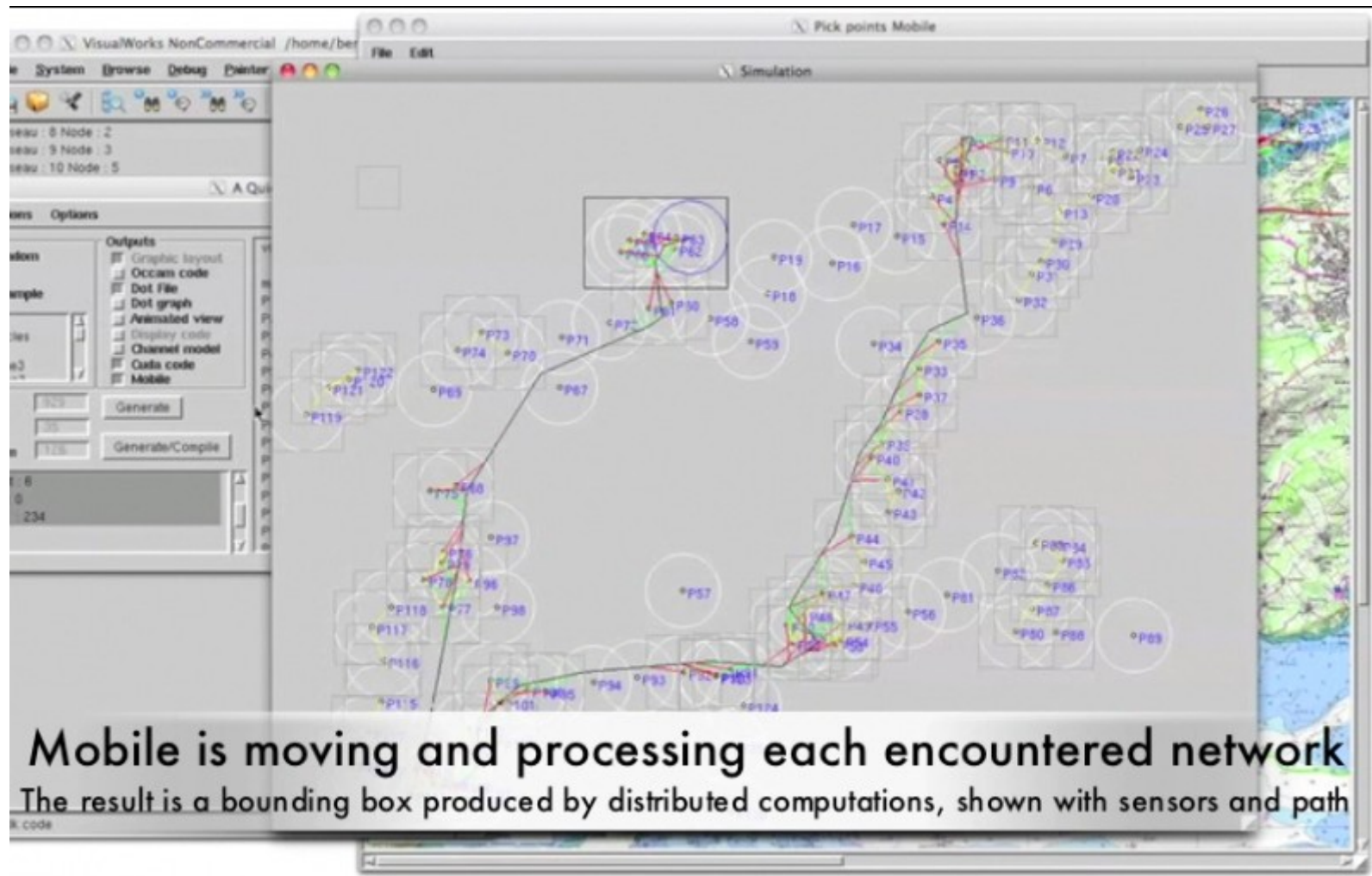
- Accrochages en vert
- Décrochages en rouge



Integration des mobiles

- Canaux libres additionnels à chaque noeud
 - TDMA slots
 - buffers en CUDA
 - (mobile channels en Occam-Pi)
- Ces canaux sont *lus*. Ils permettent aux mobiles de rejoindre les réseaux et initier des calculs
- Les mobiles peuvent executer des collectes de données, controler, transporter des infos.

Exemple : réseau cotier avec 2 portées différentes et 1 mobile



The screenshot displays the VisualWorks NonCommercial software interface. The main window shows a simulation of a coastal network with numerous nodes (labeled P1 through P120) and a path of a mobile node. The nodes are connected by lines, and the path is highlighted in red. The software interface includes a menu bar (File, Edit, Simulation), a toolbar, and a sidebar with various options and outputs. A text box at the bottom of the simulation area contains the following text:

Mobile is moving and processing each encountered network
The result is a bounding box produced by distributed computations, shown with sensors and path

At the bottom of the screenshot, there is a progress bar and a status bar showing "Mise en mémoire tampon : 100,00%" and a timer "2:00 /".

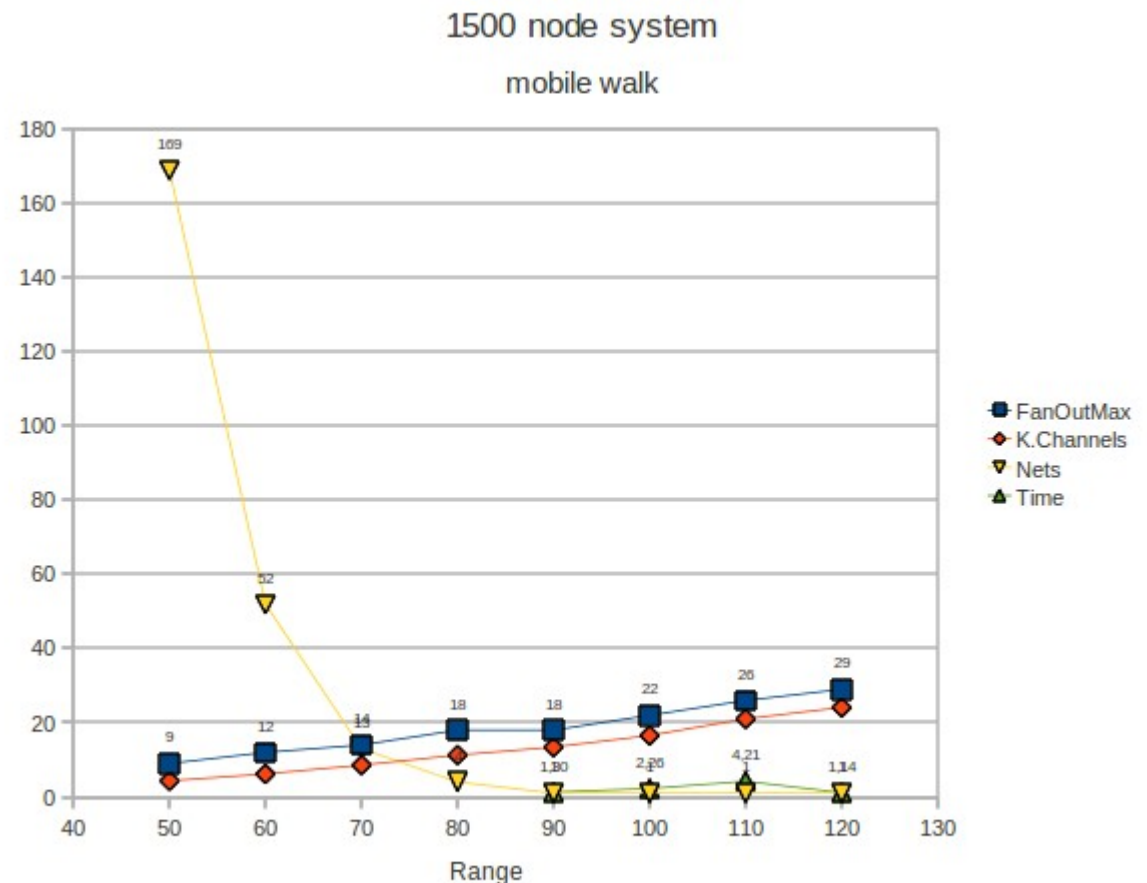


Partie 6

Quelques résultats

1500 sensor, discovery by mobile

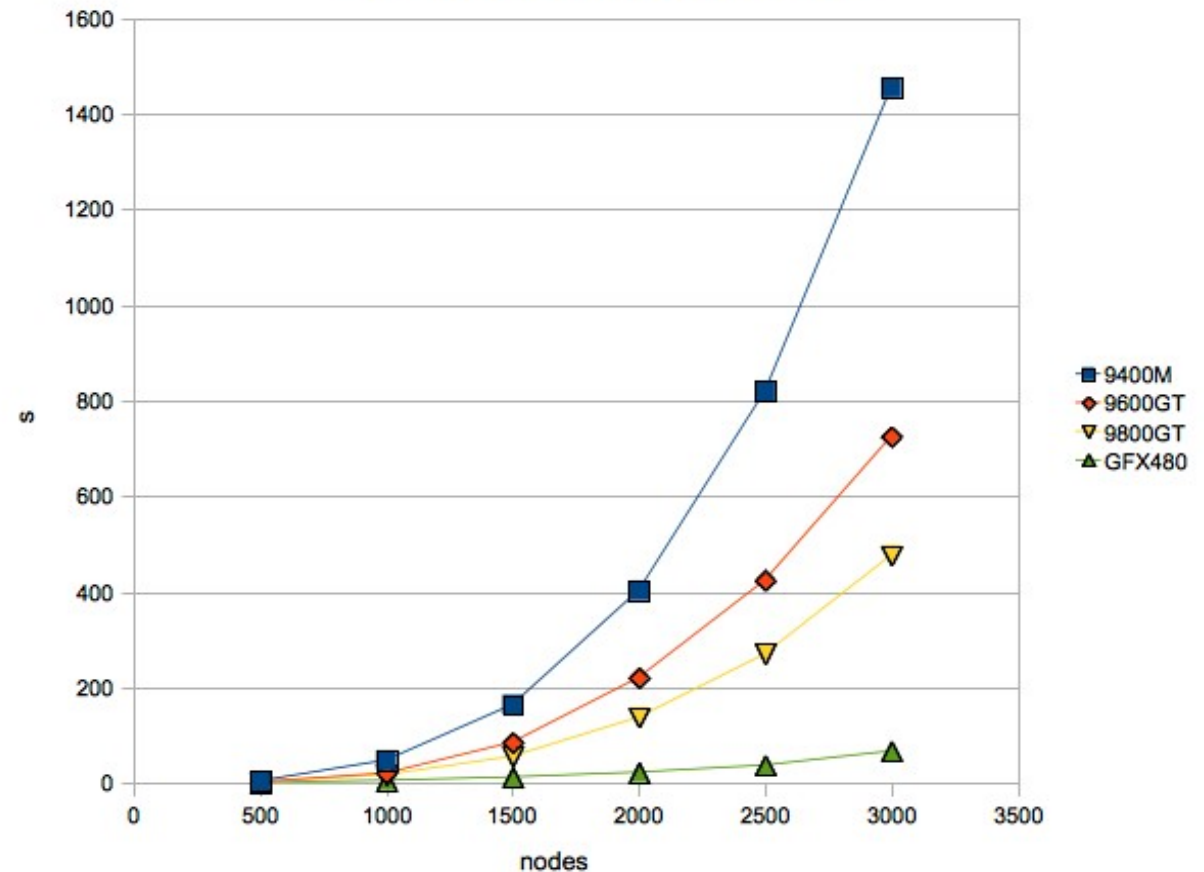
- One random system of 1500 nodes
- Wireless range varies from 50 à 120 units
- More connectivity implies less networks



6 dynamic networks – 4 GPUs

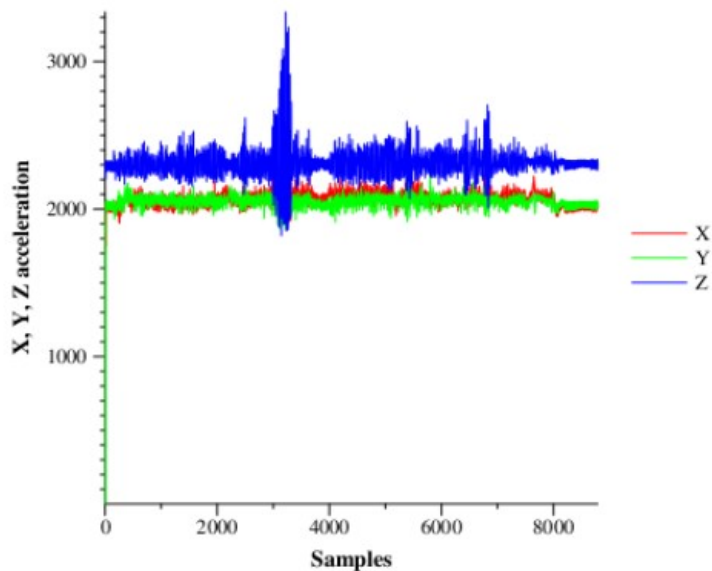
- 500 to 3000 node
- diameter, leader, network modification, again diameter, leader
- GFX480, 9800GT
9600GT, MacMini
9400M

6 random networks on 4 GPUs

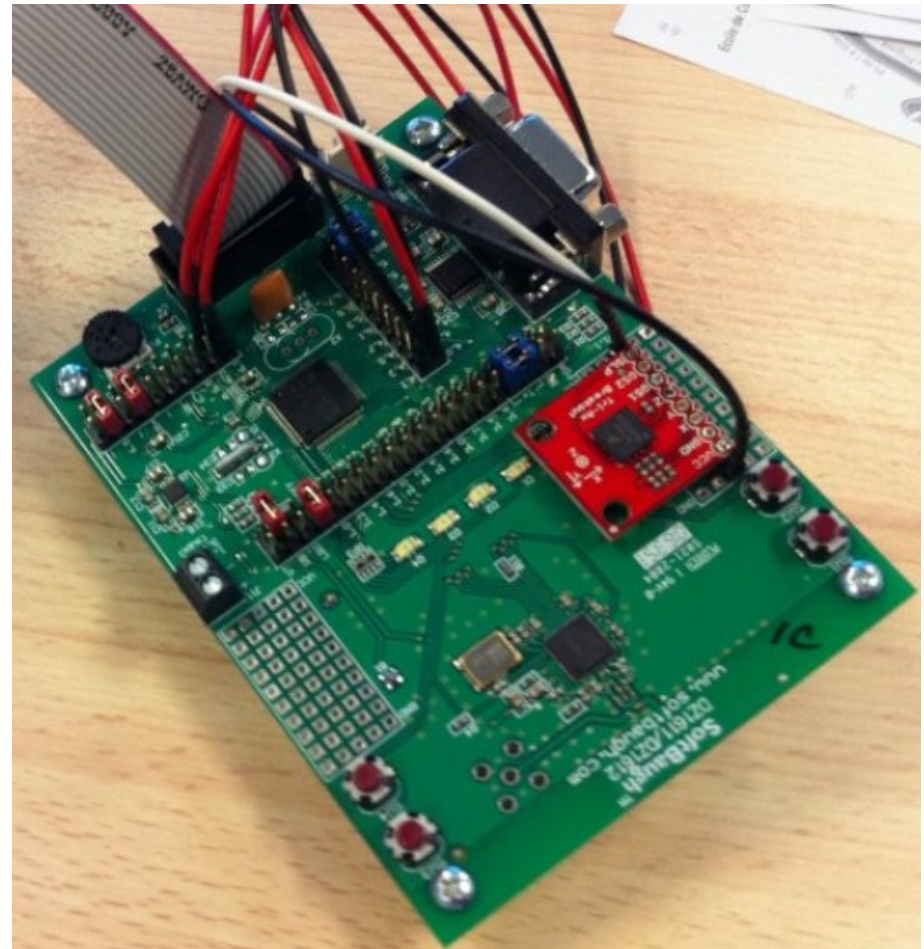


Du côté des capteurs :

- Couplage capteur accéléromètre
- Mesures (ici une promenade)



Mouvements de la carte pendant 2 min 20 environ





Partie 7

Démonstrations

Outil NetGen

- Réseaux aléatoires, réseaux réguliers
- Saisie sur carte
- Saisie d'un mobile
- Nature des fichiers générés.

Merci ..