

## TD Réseaux de capteurs sans fil

Ce TD porte sur l'étude d'un protocole d'agrégation de données s'exécutant au dessus du protocole IEEE 802.15.4 Low Rate Wireless Personal Area Networks. Le 802.15.4 est un protocole de communication destiné aux réseaux sans fil de la famille des LR WPAN du fait de leur faible consommation, de leur faible portée et du faible débit des dispositifs utilisant ce protocole. 802.15.4 est utilisé par de nombreuses implémentations des couches réseaux et applicatives : ZigBee, 6LoWPAN et d'autres protocoles propriétaires.

Les caractéristiques des LR WPAN sont :

- la formation d'un réseau de type étoile ou maillé,
- l'allocation d'une adresse de 16 bits ou de 64 bits,
- l'utilisation de CSMA/CA pour communiquer,
- la faible consommation d'énergie,
- la détection d'énergie (ED),
- l'indication de la qualité de la liaison (LQI),
- l'utilisation de :
  - 16 canaux dans la bande de fréquence de 2.4 à 2,4835 GHz,
  - 10 canaux dans la bande de fréquence de 902 à 928 MHz,
  - 1 canal dans la bande de fréquence de 868 à 868,6 MHz.

L'IEEE a défini deux types de dispositifs pouvant participer à un réseau :

- le dispositif ayant toutes les fonctions possibles (FFD : Full Function Device)
- le dispositif ayant des fonctions limitées (RFD : Reduced Function Device)

Le FFD peut assurer trois rôles dans un réseau : coordinateur PAN, routeur ou dispositif relié à un capteur (plus petite fonction possible, appelé dispositif de fin).

Le RFD est prévu pour des applications simples (signaler l'état d'un capteur, contrôler l'activation d'un actionneur). Il est considéré comme un dispositif "d'extrémité" (end device), dans le sens où il n'est pas essentiel au réseau.

Pour communiquer sur un même réseau, un FFD (au moins) et des RFD doivent utiliser le même canal physique parmi ceux définis selon la bande de fréquence choisie. Le FFD peut dialoguer avec des RFD et des FFD, tandis que le RFD dialogue avec un FFD uniquement.

Voir : [https://fr.wikipedia.org/wiki/IEEE\\_802.15.4](https://fr.wikipedia.org/wiki/IEEE_802.15.4)

Ce TD utilise le simulateur à événements discrets *ns-2* pour étudier ces protocoles. *ns-2* utilise le langage compilé C++ pour coder les éléments du simulateur et le langage interprété Tcl ainsi que son extension objet OTcl pour écrire les scripts d'exécution. Pour que *ns-2* fonctionne, l'environnement de Tcl/OTcl doit donc être bien configuré.

L'archive originale se trouve ici mais elle présente qlq problèmes de compilation :

<http://sourceforge.net/projects/nsnam/files/allinone/ns-allinone-2.35/ns-allinone-2.35.tar.gz/download>

(optionnel) Effectuer l'installation selon les consignes du README. Quelques pbs de compilation peuvent apparaître. Dans les fichiers `Makefile.in` ajouter `-fPIC` et `-Wno-narrowing` au flags du compilateur `-CFLAGS`. Régénérer les makefiles avec `./configure` et lancer les séparation. Certains liens symboliques risquent d'être erronés (e.g., `nam`). Pensez à les rectifier avec la commande `ln -s`.

1) Dans votre fichier `.bashrc`, ajoutez les lignes suivantes en remplaçant `<login>` par votre `login/homedir` :

```
export PATH=/home/<login>/ns-allinone-2.35/bin:/home/<login>/ns-allinone-2.35/tcl8.5.10/unix:/home/<login>/ns-allinone-2.35/tk8.5.10/unix:$PATH
```

```
export LD_LIBRARY_PATH=/home/<login>/ns-allinone-2.35/lib:/home/<login>/ns-allinone-2.35/otcl-1.14:$LD_LIBRARY_PATH
```

```
export TCL_LIBRARY=/home/<login>/ns-allinone-2.35/tcl8.5.10/library:$TCL_LIBRARY
```

Si vous placez `ns-allinone-2.35` ailleurs, modifiez le `PATH` en conséquence.

Puis rechargez la config en tapant :

```
$ source .bashrc
```

La documentation complète de ns-2 est ici : <https://www.isi.edu/nsnam/ns/ns-documentation.html>

Il y a aussi un Wiki : [http://nsnam.sourceforge.net/wiki/index.php/Main\\_Page](http://nsnam.sourceforge.net/wiki/index.php/Main_Page)

2) Copiez l'archive `/net/stockage/dmagoni/ns-allinone-2.35.tar.xz` dans votre répertoire `~/espaces/travail` puis recompilez ns-2 en tapant :

```
$ unxz ns-allinone-2.35.tar.xz
```

```
$ tar xf ns-allinone-2.35.tar
```

```
$ cd ns-allinone-2.35
```

```
$ ./install
```

```
$ ns
```

Lorsque vous lancez ns, vous devez obtenir :

```
In dmp.tcl...
```

```
%
```

3) Effectuez le tutoriel de Marc Greis pour apprendre à utiliser ns-2 :

<https://www.isi.edu/nsnam/ns/tutorial/index.html>

4) Le but est maintenant d'étudier un nouveau protocole d'agrégation de données.

Dans le dossier `ns-2.35/dmp`, lisez le fichier `sim-dmp.tcl`.

Toujours dans le dossier `ns-2.35/dmp`, lancez `ns sim-dmp.tcl` pour exécuter une simulation.

Observez les sorties.

Lancez `nam anim-dmp.nam &` pour visualiser la trace.

Observez les noeuds du réseau s'envoyer des paquets.

5) Faites varier les paramètres suivants dans le fichier `sim-dmp.tcl` :

- `net_type` : de 1 à 4
  - 1 = autoroute, 2 = croisement, 3 = rond-point, 4 = grille
- `net_size` : 10, 20, 40, 80, 160, 320, 640 (noeuds)
- `algorithm` : de 1 à 4
  - 1 = aucune agrégation, 2 = agrégation partielle, 3 = agrégation aléatoire, 4 = agrégation totale
- `sim_duration` : 300, 600, 1200 (secondes)
- `message_period` : 30, 60, 120 (secondes)
- `local_wait_period` : 3, 6, 12 (secondes)
- `global_wait_period` : 6, 12, 24 (secondes)

L'impact des paramètres et le fonctionnement des algorithmes est expliqué ici:

<http://dept-info.labri.fr/~magoni/risc/brahmi-p2mnet2015.pdf>

6) Chaque simulation génère un fichier nommé `data-dmp-<type>-<size>-<algo>-<period>.txt`

Dans les fichiers `data-*`, chaque ligne s'interprète comme suit :

```
seed net_type size algo period agg_ratio agg_sd delay delay_sd energy_ratio energy_sd
```

Les 5 premiers champs sont donc des paramètres d'entrée et les 6 derniers champs sont des variables de sortie.

Avec ces fichiers et `gnuplot`, tracez les figures suivantes :

- ratio d'aggregation
  - $= 1 - \text{nb\_paquets\_reçus} / \text{nb\_paquets\_envoyés}$
- latence de bout-en-bout
- économie d'énergie
  - $= \text{énergie\_restante\_avec\_aggregation} / \text{énergie\_restante\_sans\_aggregation}$

En fonction des paramètres suivants :

- topologie du réseau (type)
- taille du réseau (nombre de noeuds)
- algorithme (sans aggrégation, avec aggrégation)

Soit un fichier de données `data.txt` dans lequel chaque ligne contient les valeurs des paramètres d'entrée et des variables de sortie mesurées durant une simulation et chaque colonne contient toutes les valeurs pour un paramètre d'entrée ou une variable de sortie donnée :

```
#time v1 v1_sd v2 v2_sd
10 31 2 29 3
13 46 5 56 5
16 75 6 63 4
...
```

Créez un script `script-gp.plt` pour `gnuplot` afin de produire des figures pour les données de simulation. Inspirez vous du code ci-dessous :

```
set term pdf size 3.5,2.5 enhanced monochrome dashed font "Helvetica, 12"
set xtics auto
unset mxtics
set ytics auto
unset mytics
set xrange [5:25]
set yrange [0:100]
unset title
set key top left
set grid
set output "figure.pdf"
set xlabel "Time (s)"
set ylabel "Delivered Packets (%)"
plot "data.txt" u 1:2:3 title "v1" yerrorlines lw 1, \
"data.txt" u 1:4:5 title "v2" yerrorlines lw 1
```

Produisez les figures en lançant : `gnuplot script-gp.plt`

7) Modifiez le fichier source `dmp.c` pour y définir un nouvel algorithme créé par vous-même et donnant de meilleurs résultats que les algorithmes proposés. Après une modification, n'oubliez pas de faire `make` dans le dossier `ns-2.35`.