

# **TP RPL - Résumé**

## **Sommaire**

1 – Internet des Objets.....	1
2 – 6LoWPAN.....	1
2.1 – Ipv6 Low power Wireless Personal Area Networks .....	2
2.2 – Le protocole de routage standardisé RPL.....	2
2.2.1 – Construction d'un DODAG et routage.....	3
3 – Contiki.....	4
3.1 – Instant Contiki .....	4
4 – Cooja .....	4
5 – Exercices .....	5

## **1 – Internet des Objets**

L'IOT est un domaine où des objets communiquent entre eux à travers Internet. Il existe de nombreux types d'objets connectés :

- Voitures
- Sondes
- Caméras
- Portes
- Etc...

Afin de communiquer ces objets utilisent des réseaux basés sur :

- Un protocole bas débit, lent pour la collecte de données (LoRa)
- Des solutions IP (6LowPAN)

Ces objets sont de petites tailles, autonome ce qui implique certaines contraintes :

- Interface radio (sans fil)
- Capacité énergétique limitée
- Peu de puissance de calcul
- Changement d'environnement.

Ressources complémentaires IDO :

[https://www.openscience.fr/IMG/pdf/iste\\_ido18v2n1\\_1.pdf](https://www.openscience.fr/IMG/pdf/iste_ido18v2n1_1.pdf)

[https://www.cisco.com/c/dam/global/en\\_ca/solutions/executive/assets/pdf/internet-of-things-pdf](https://www.cisco.com/c/dam/global/en_ca/solutions/executive/assets/pdf/internet-of-things-pdf)

## **2 – 6LoWPAN**

IPv6 propose un nombre d'adresses beaucoup plus important, codées sur 128 bits (2128 adresses), permettant ainsi de pallier la limitation du nombre d'adresses permises par IPv4. C'est également un protocole plus sécurisé et adéquat par sa simplification d'écriture (voir RFC 5952) permettant aussi une plus grande flexibilité. Les en-têtes des paquets IPv6 ont une taille fixe de 40 octets.

L'Ido profite de cette capacité d'adressage qui permet à chaque objet d'avoir sa propre adresse sur le réseau. Cependant ce protocole n'étant pas conçu pour les objets à faibles ressources par la taille des entêtes utilisées dans les communications, il faut en utiliser un plus optimisé. Il s'agit de IPv6 Low power Wireless Personal Area Networks (6LoWPAN).

## 2.1 – Ipv6 Low power Wireless Personal Area Networks

Les capacités des appareils étant faibles, la transmission et l'encapsulation des données sont des éléments déterminants. Ainsi, 6LoWPAN qui est un mécanisme d'encapsulation des paquets lors de la transmission de données dans des réseaux de faibles capacités sera utilisé dans les systèmes à ressources limitées.

Un réseau 6LoWPAN sans fil propose une adaptation du protocole IPv6. Au niveau II, les équipements suivent le standard IEEE 802.15.4. Ils ont une portée limitée, un faible débit, peu de mémoire et un faible coût. Une couche d'adaptation entre les couches MAC et réseau réduit la surcharge de l'en-tête IPv6 et réalise une compression. Elle propose des en-têtes supplémentaires permettant de fragmenter un paquet IPv6 si ce dernier ne peut pas être contenu dans une trame MAC.

Trois types de nœuds existent dans 6LoWPAN : le routeur de bordure (BR), le routeur et l'hôte. La topologie de base est un graphe acyclique orienté (Directed Acyclic Graph - DAG). Un Graphe Orienté Acyclique (DAG) est un graphe orienté ne possédant aucun cycle (orienté). Dans un réseau 6LoWPAN, plusieurs graphes acycliques orientés destination (Destination Oriented Directed Acyclic Graph - DODAG) peuvent exister et forment le DAG. La caractéristique supplémentaire apportée par ce type de graphe est que chaque arc du graphe est dirigé vers une seule et même destination qui est un BR. Le routeur de bordure est la racine (la destination) d'un DODAG. Il connecte les routes à l'Internet.

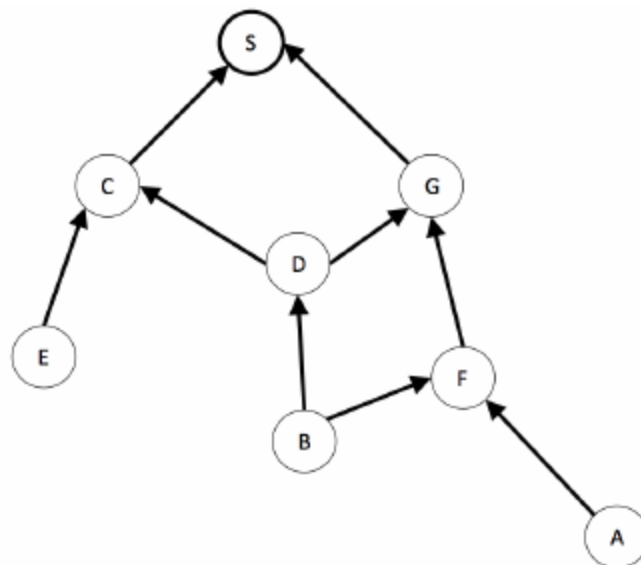


FIGURE 1 – DODAG - Graphe acyclique orienté vers une destination

Les routeurs sont des équipements intermédiaires sur les routes, et les hôtes sont les équipements feuilles du réseau. Le réseau est alors un réseau sans fil, multi-saut.

## 2.2 – Le protocole de routage standardisé RPL

Le trafic typique dans le réseau est le ramassage des données incast (multipoint-à-point, des nœuds vers le BR), mais il permet la diffusion (du BR aux nœuds), ainsi que le point-à-point entre les nœuds. Ce protocole de routage construit un DODAG pour chaque application (et naturellement vers un BR). Les routeurs et le BR sont autorisés à router des paquets. Les paquets fragmentés sont reconstitués à chaque saut, ce qui rend le protocole plus robuste face aux pertes de fragments. Nous illustrons brièvement la construction des routes, le routage et la maintenance / correction des routes.

### 2.2.1 – Construction d'un DODAG et routage

L'arbre est construit couche par couche (rang par rang). La racine RPL (un BR) a un rang égal à 1, et elle envoie un message (DIO, voir sur la figure) pour mobiliser ses voisins. Les nœuds dans sa zone de couverture (1 et 4) reçoivent et traitent ce DIO. Après avoir reçu les DIO, chaque nœud essaye de se connecter au DODAG en choisissant un parent préféré en utilisant une fonction d'objective (par exemple en minimisant le nombre de sauts jusqu'à R). Les nœuds 1 et 4 vont avoir un rang 2. Leur parent est trivialement R. Les nœuds 1 et 4 émettent leur propre DIO vers leur voisinage (les nœuds 2, 3 pour le nœud 1 et les nœuds 5, 6 pour le nœud 4) qui procèdent de la même manière. Quand la construction est terminée, les nœuds peuvent communiquer avec la racine R.

Dans l'autre sens, les routes descendantes peuvent être construites grâce au message Destination Advertisement Object (DAO). Les messages DAO contiennent la destination. Pour simplifier, regardons rapidement les routes descendantes en mode « storing ». Les messages DAO sont envoyés au parent préféré qui installe à la réception du DAO la route descendante correspondante. Le DAO se propage jusqu'à la racine RPL à travers les parents préférés afin d'établir un chemin descendant complet de la racine jusqu'à la destination.

Plus largement, RPL permet de sélectionner un ensemble de parents (dont le parent préféré) tel que le rang des parents doit être inférieur au rang du nœud. Le nœud installera, en plus de la route par défaut vers son parent préféré, des routes vers les autres parents. Notons que pour réparer les éventuelles pannes du réseau, RPL dispose de deux mécanismes : la réparation globale (pour coconstruire le DODAG) et la réparation locale (pour refermer localement un lien manquant).

Une fois le DODAG est construit, il est possible d'utiliser les routes sur l'arbre (dans les deux sens). Pour trouver une route entre deux nœuds, on peut toujours remonter jusqu'à la racine BR puis redescendre à la destination.

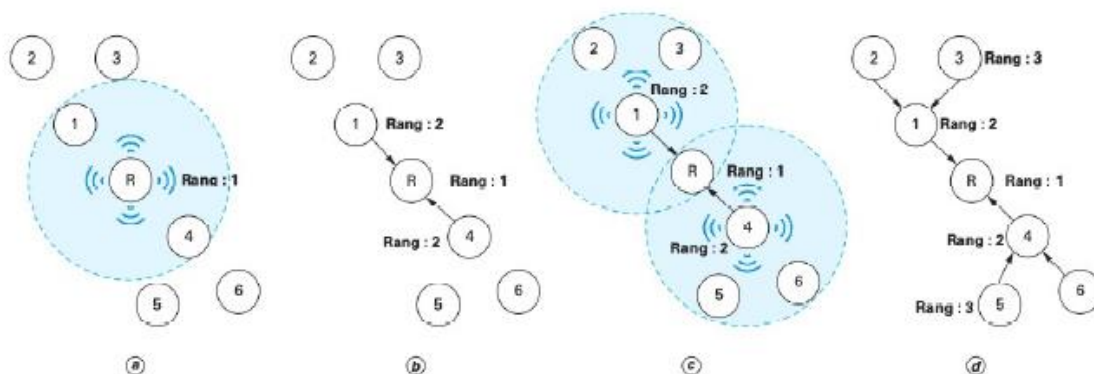


FIGURE 2 – Construction d'un DODAG

## 3 – Contiki

Pour faire fonctionner les objets limités en ressources et la communication entre les objets, il faut un système relativement petit qui utilise peu de ressources. Pour des applications qui nécessitent une latence limitée et une robustesse, 6LoWPANet le routage RPL sont proposés.

Contiki est un système d'exploitation "open source" conçu spécialement pour les petits objets communicants. C'est un système d'exploitation écrit en C, utilisant peu de ressource et d'énergie. Il permet de connecter des objets peu compliqués et peu chers, qui possèdent peu d'énergie. C'est également une boîte à outils pour construire des systèmes sans fil complexes. De plus, Contiki propose un environnement de simulation (Cooja) permettant de tester des systèmes avant de passer à la réalisation.

### 3.1 – Instant Contiki

Pour travailler efficacement avec Contiki et Cooja, il existe un environnement de développement conçu par les créateurs de Contiki appelé Instant Contiki fourni sous la forme d'un disque dur virtuel contenant tous les outils dont nous avons besoin pour utiliser Contiki et Cooja.

## 4 – Cooja

Il permet la simulation de différents réseaux avec la création de divers objets comme des capteurs ou des Border Routers. Nous l'utiliserons ici avec le protocole IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL). Dans la simulation, on peut utiliser les éléments intégrés dans Cooja (appelés des motes) qui font appel à Contiki.

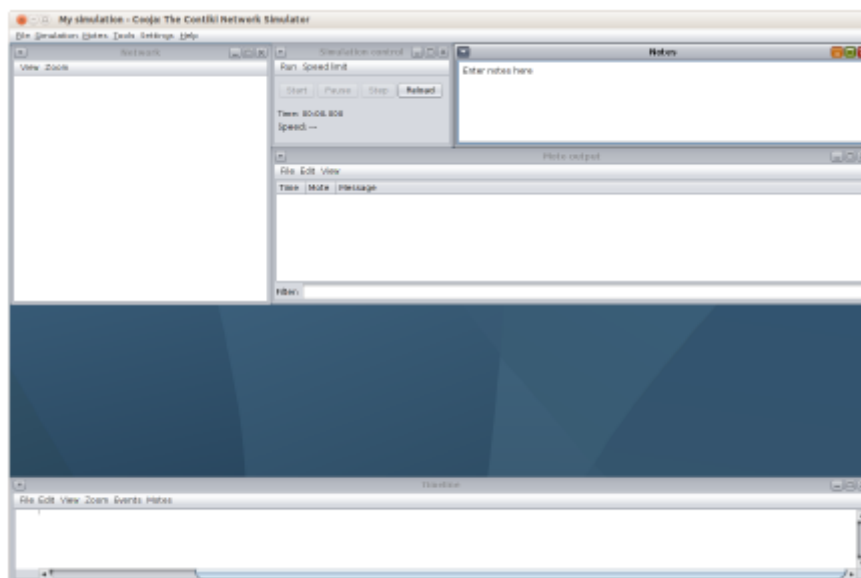


FIGURE 4 – Application Cooja au démarrage

## 5 – Exercices

Dans une simulation on a :

”Network” : fenêtre représentant les différents nœuds et leur positionnement

”Simulation Control” : fenêtre permettant de contrôler le déroulement de la simulation

”Mote output” : fenêtre permettant de récupérer les logs des nœuds.

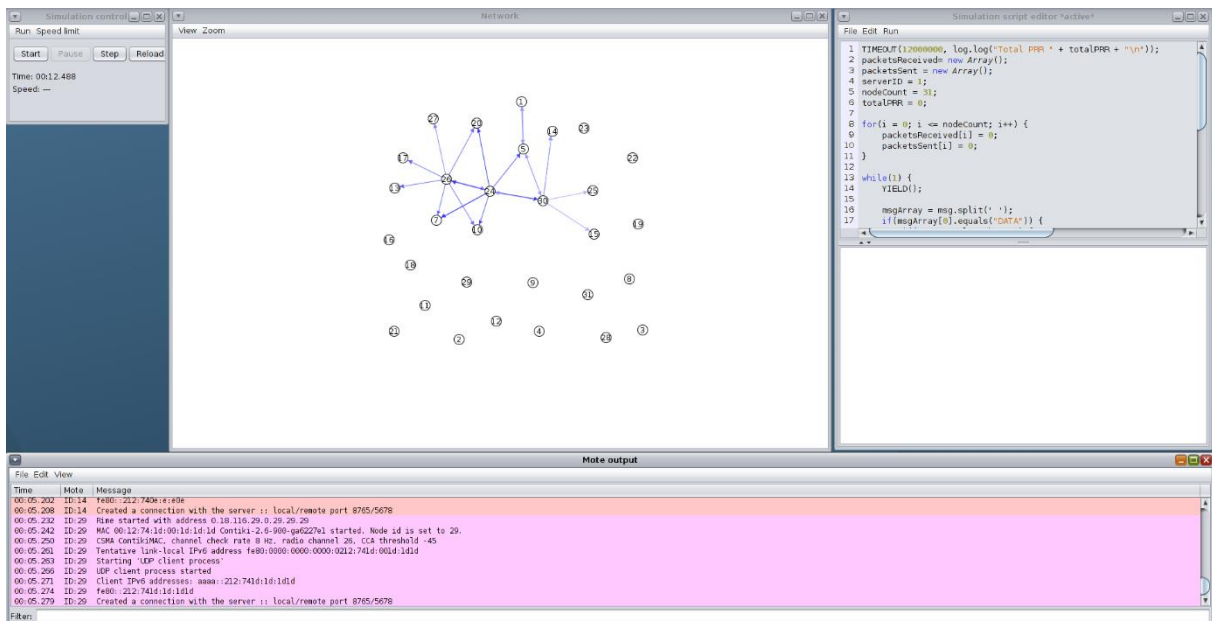


FIGURE 5 – Création du DODAG

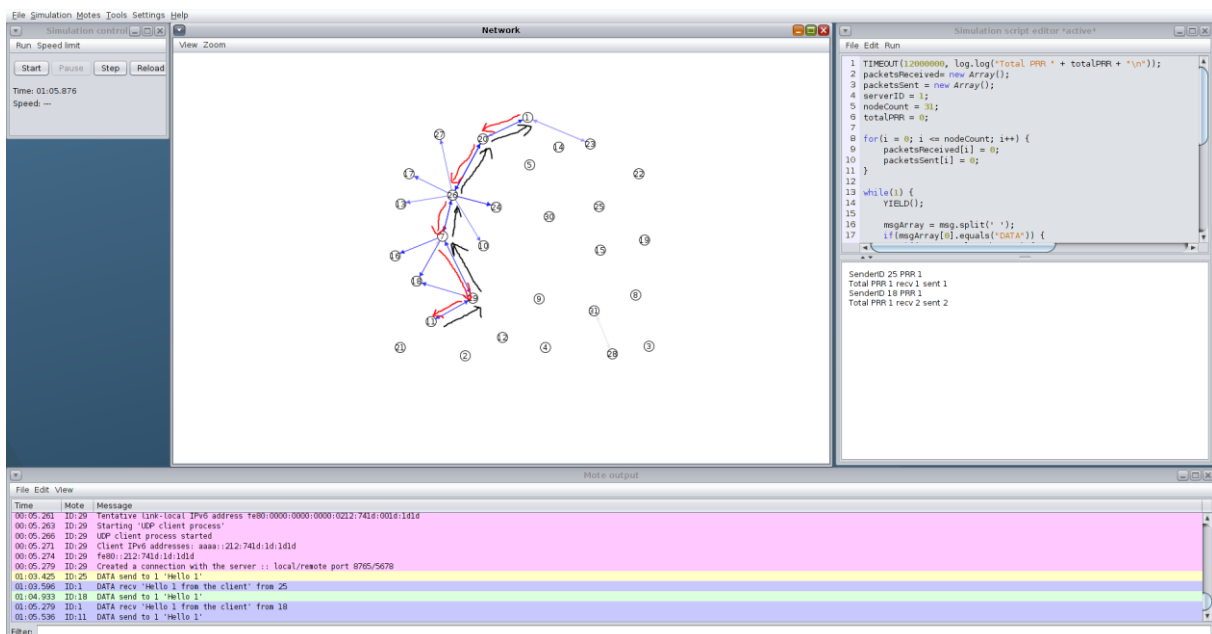


FIGURE 6 – Envoi/Réception d'un message

Une fois que le DODAG est créé les nœuds peuvent s'envoyer des messages (ici Hello « BR » donc Hello 1). Lorsque le nœud souhaite envoyer son message il recherche la route la plus rapide afin de

l'envoyer. Une fois que le message arrive à un Mote il recherche également la route la plus courte pour arriver au BR et ainsi de suite jusqu'à que le message arrive au BR. Ensuite une confirmation de réception est envoyée en suivant le chemin en sens inverse. Sur la Figure 6, en noir l'envoi, en rouge le retour.

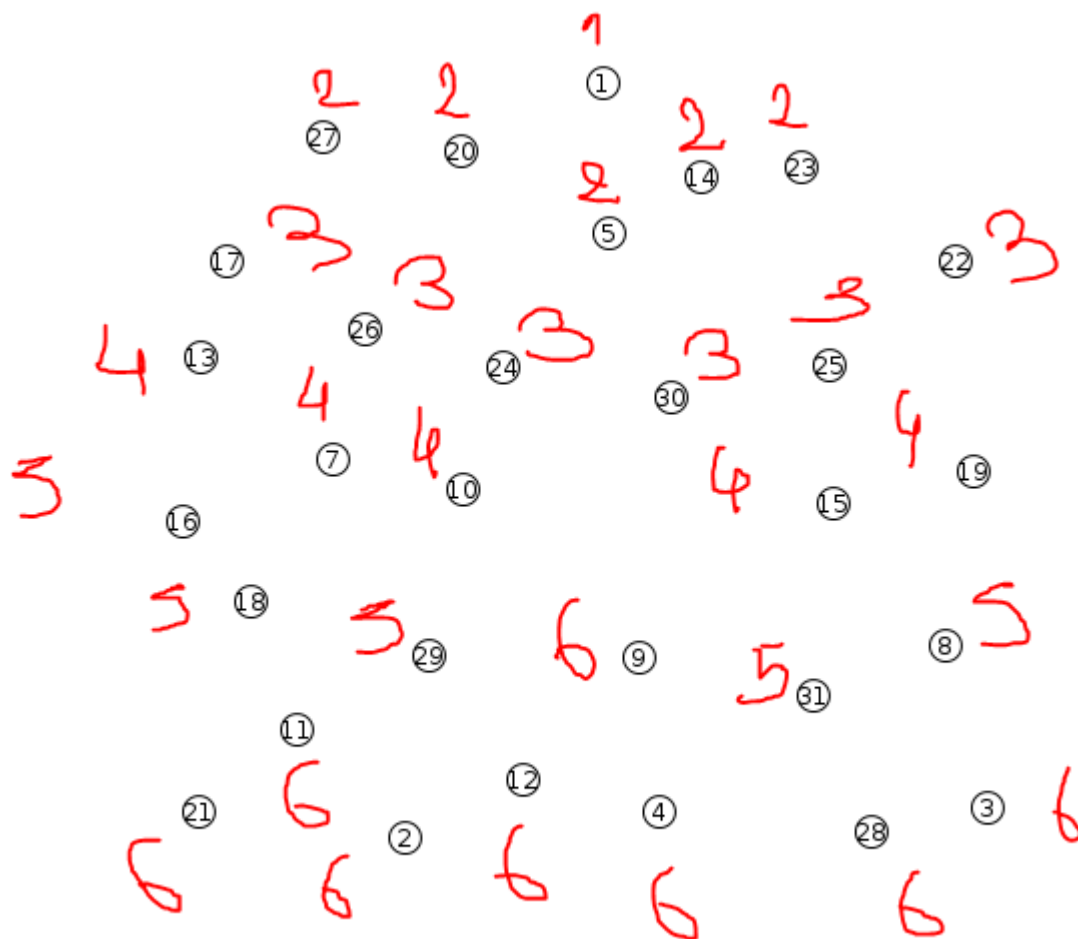


FIGURE 7 – Rang des Motes

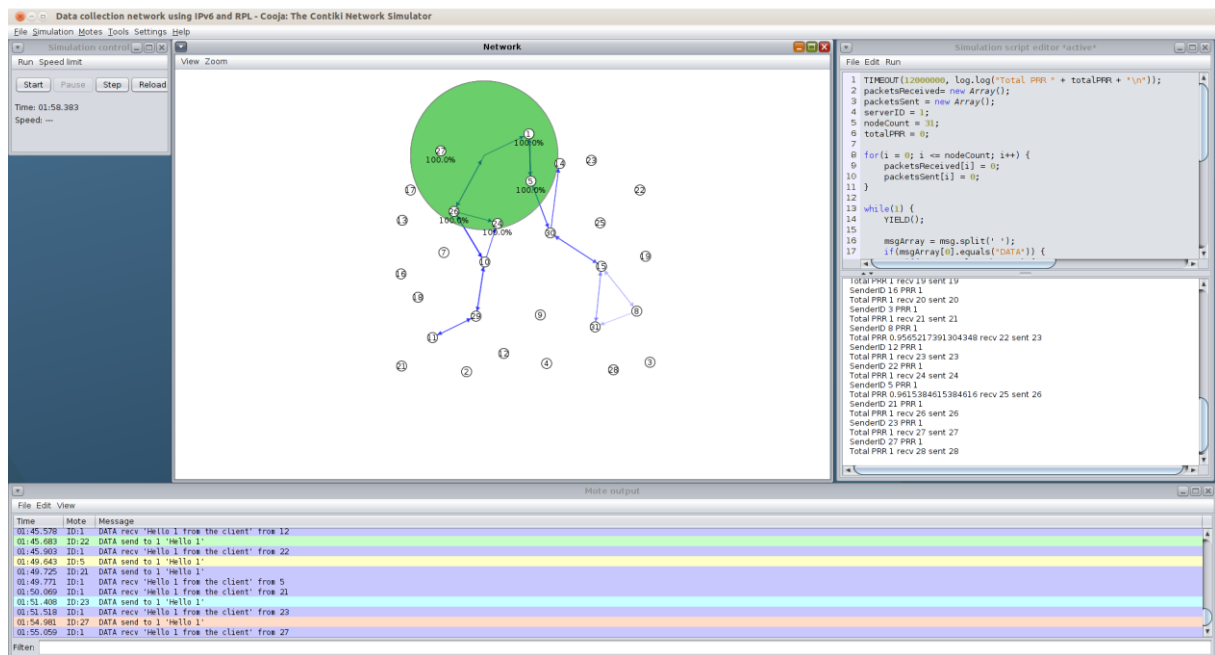


FIGURE 8 – Le Mote 20 vient d’être supprimé

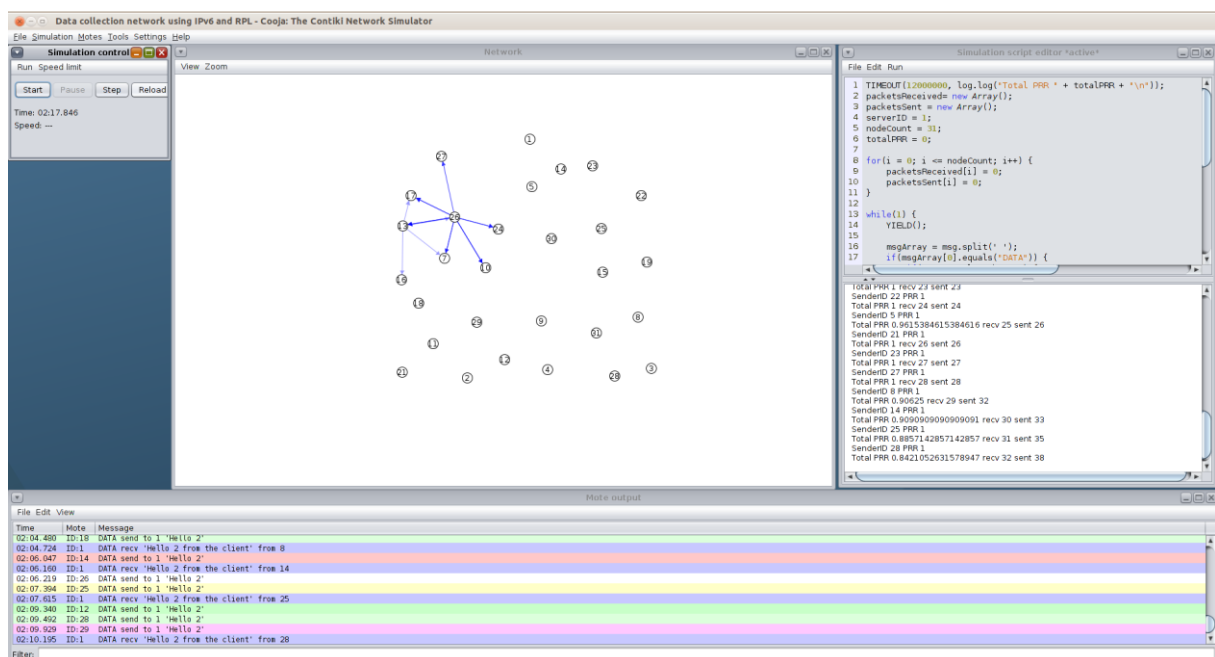


FIGURE 9 – Reconstruction du DODAG

Après avoir supprimé le Mote 20 (cf. Figure 8) une partie du RPL doit être remodeler afin d’accéder au BR. Il y a une nouvelle topologie (cf. Figure 9). Lorsque qu’un message est envoyé et qu’un Mote de la route n’existe plus alors le message ne parvient pas au BR et le Mote devra alors en envoyer un différent (Hello 1 devient Hello 2 etc...). Lorsque qu’un nouveau message est créé alors une nouvelle topologie est construite jusqu’à que le message atteigne le BR.

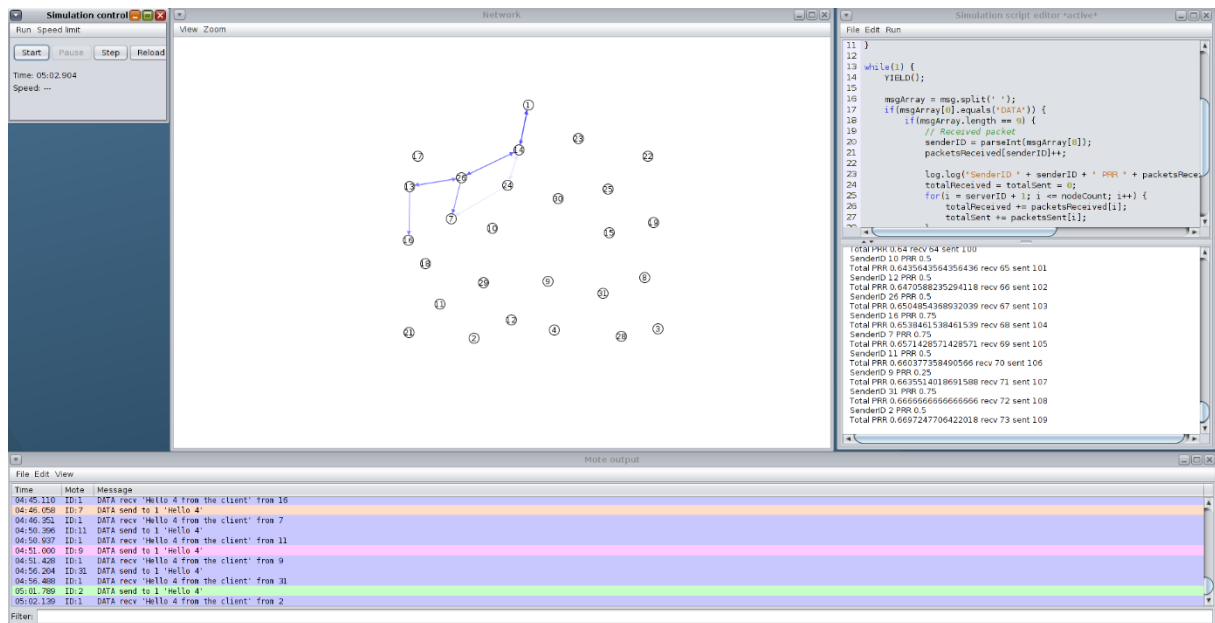


FIGURE 10 – 4<sup>e</sup> topologie

Sur la Figure 10, on voit qu'il y a fallut atteindre une 4 nouvelles topologies afin que tous les Motes puissent envoyer des messages au BR. Tous les Motes se calibrent sur une même topologie bien que certains en n'ont pas la nécessité.