Élèves :

Nicolas Blaye
nicolas.blaye@telecom-bretagne.eu

Jianfei PAN
jianfei.pan@telecom-bretagne.eu

Encadrants :

Isabel AMIGO
isabel.amigo@telecom-bretagne.eu

Christophe LOHR
christophe.lohr@telecom-bretagne.eu

Sandrine VATON
sandrine.vaton@telecom-bretagne.eu

Projet S5 102 : Découverte de routes optimales dans des overlays

Rapport Biblio

Version 2.1 - 22/11/2015

Formation d'ingénieur

Année scolaire 2015-2016

Sommaire

Index des illustrations

1. INTRODUCTION

Avec l'évolution des services sur internet, vient un besoin toujours plus grand en stockage et en traitement de données. Des sociétés comme Facebook, Orange ou encore Netflix possèdent de nombreux centres de données ainsi que des serveurs éparpillés dans le monde entier pour gérer ces données. Pour assurer la meilleure qualité de service possible, il faut que ces points névralgiques soient connectés de la meilleure façon possible, or, à cause des stratégies différentes choisis par les routeurs sur l'internet, le routage n'est pas toujours optimal au niveau de la qualité de service (délai, bande passante, disponibilité…). Dans la suite, nous considérerons ces points névralgiques comme un overlay, en ignorant l'architecture des routes entre ces points (qui peut être extrêmement compliquée).

L'objectif de ce projet est de proposer une solution pour faire le mesure du réseau overlay en fonction d'une métrique et trouver le routage optimal entre les noeuds (les centres des données, les serveurs...).

Ce rapport bibliographique sert à étayer le sujet de notre projet. Une première partie sera consacrée aux solutions de routage actuelles, et notamment aux protocoles RIP, OSPF et BGP qui sont pour les deux derniers les principaux protocoles de routage. La partie suivante présente l'outil que nous allons utiliser pour mesurer l'overlay , en présentant la problématique de représentativité de cette mesure. Une autre partie sera consacrée à l'étude d'algorithme de plus court chemin dans un graphe. La dernière partie abordera le planning du projet ainsi que ses différents livrables.

2. ANALYSE DU DOMAINE

2.1 Présentation des solutions routage actuelles

On distingue deux types de solutions de routage: les solutions internes à un système autonome dites IGP: Interior Gateway Protocol), tel que RIP et OSPF, ou les solutions externes à un système autonome dites EGP (Exterior Gateway Protocol) tel que BGP, qui lient deux systèmes autonomes.

2.1.1 Les protocoles de routage internes

RIP

Routing Information Protocol est un protocole de routing de vecteur-distance qui emploie le comptage des "hop" comme mesure du routage. On s'intéresse à ce protocole car c'est l'un des tous premiers protocoles utilisés largement par les routeurs sur Internet. Même si ce protocole est dépassé, il reste que cela nous permet d'introduire la notion de métrique dans un réseau.

RIP fonctionne en spécifiant un maximum de hop (nombre de saut, c'est-à-dire de passages par un routeur) de 15 pour éviter que les messages tournent à l'infini dans le système. Il s'appuie sur un échange d'informations entre les routeurs adjacents. Chaque routeur connaît certaines routes et leur coût et transmet ces informations aux autres qui mettent à jour leurs informations sur ces routes.

Ces informations sont rassemblées dans une table de routage:

```
R1#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

C    10.0.0.0/8 is directly connected, Serial0/0
C    20.0.0.0/8 is directly connected, FastEthernet0/1
R    30.0.0.0/8 [120/1] via 10.0.0.2, 00:00:02, Serial0/0
                [120/1] via 20.0.0.2, 00:00:11, FastEthernet0/1
R    40.0.0.0/8 [120/1] via 20.0.0.2, 00:00:11, FastEthernet0/1
R    50.0.0.0/8 [120/1] via 10.0.0.2, 00:00:02, Serial0/0
R    60.0.0.0/8 [120/2] via 10.0.0.2, 00:00:02, Serial0/0
                [120/2] via 20.0.0.2, 00:00:11, FastEthernet0/1
R    70.0.0.0/8 [120/1] via 10.0.0.2, 00:00:02, Serial0/0
C    192.168.1.0/24 is directly connected, FastEthernet0/0
R    192.168.2.0/24 [120/2] via 10.0.0.2, 00:00:02, Serial0/0
```

Illustration 1: Une table de routage

Lorsque le routeur reçoit un nouveau chemin pour une route qu'il connaît déjà, il compare le nombre de hop: si il y a moins de hop il change sa route, sinon il ne fait rien.

Quand il reçoit un chemin pour une route qu'il ne connaît pas, il regarde d'abord si le nombre de hop est supérieur à 15, et si non il l'inscrit dans sa table de routage.

Les routeurs envoient leur table de routage à leurs voisins toutes les 30+-5 secondes, et quand ils modifient leur table de routage, il envoie une mise à jour aux voisins dans les 5 secondes après l'ajout de la route.

RIP fonctionne avec trois timers:

- **Invalid Timer** : au bout de 180 sec, s'il ne reçoit aucune mise à jour de la route, il la change en route inaccessible (hop = 16), mais la route reste.
- **Flush Timer**: au bout de 240 sec, la route placée en inaccessible est supprimée
- **Holdown timer** : Quand une route devient inaccessible, elle reste dans la table pour que cette information soit communiquée à tous les routeurs jusqu'à convergence de la topologie (qui peut être assez lente en RIP)

(nota bene: valeur de timeout de routeur cisco)

RIP fonctionne avec 2 techniques pour accélérer la convergence:

- **Split Horizon** : interdit à une routeur d'envoyer une route à son voisin si cette route passe par le voisin
- **Poison Reverse** : routeur diffuse les routes inaccessibles, et les routeurs recevant ces informations les intègrent dans leur table jusqu'au flush timer.

Dans ce protocole, la métrique est le nombre de sauts effectués par le paquet. Le nombre de sauts étant limité, cela limite donc la taille du réseau. Il est facile à configurer, mais passe difficilement à l'échelle et à un temps de convergence élevé (temps pour que les tables de routage des différents routeurs puissent attendre tout le réseau). Plus le nombre de hop sera élevé, plus le temps de convergence sera grand. Un temps de convergence de quelques minutes peut être acceptable pour RIP, alors que le temps de transport d'un message est de l'ordre de la miliseconde.

OSPF

Open Shortest Path First fonctionne sur le principe de routage par information d'état de lien.

Ce type de routage fait que chaque routeur connaît toute la topologie du réseau et crée un arbre dont il est la racine. Le calcul de cet arbre permet à chaque routeur de connaître le chemin complet , mais il n'a que l'adresse de ces voisins. Lorsqu'un lien change, tous les voisins du routeur qui voit ce changement sont alertés par un mécanisme d'inondation.

La métrique utilisé par OSPF est un coût additif, qui prend en compte la bande passante de chaque lien. Le chemin qui est choisi est celui qui a le coût le plus faible, cela veut dire celui qui a priori a la meilleure bande passante. *coût* = 10^8/bande passant(b/s) (pour les routeurs cisco)

Un des principales problèmes d'OSPF, et que le calcul du graphe est une opération compliquée, qui augmente en complexité avec le nombre de noeuds. On est donc limité en nombre de noeuds par la capacité de calcul des routeurs.

Pour contourner ce problème, OSPF définit des notions de domaine et de zone pour limiter la charge de calcul, et les routeurs appartenant à un même système ne communiquent leurs informations que dans ce système.

Chaque routeur utilise l'algorithme de Djikstra pour établir le plus court chemin, et l'échange d'information sur les liens se fait par le protocole HELLO.

Il existe une zone qui est connectée à toutes les autres appelée backbone area. toutes les autres zones sont contiguës.

Dans ce protocole, il existe plusieurs types de routeurs:

- Routeur interne: routeur qui se trouve entièrement dans une zone
- Area Border Router: routeur disposant d'interfaces dans différentes zones
- Router Backbon: contient au moins une interface avec la zone backbone
- Autonomous System Boundary Router: fait le lien entre OSPF et un autre protocole.

Avantages:

- Courant
- Prise en compte des débits dans la métrique (coût = 10^8/bande passant(b/s))
- temps de convergence court
- économe en bande passante (seul message Hello pour actualiser les liens, coûte peu)

Inconvénients:

- demande grosse capacité pour les routeurs (mémoire et calcul)
- flapping : vitesse de cpu joue dans convergence
- configuration difficile
- topologie < 1000 routeurs
- backbone -> limite les topologies possibles

Dans ce protocole, on a une notion de qualité de service avec la prise en compte de la bande passante des liens. Le temps de convergence est beaucoup plus faible que pour RIP, avec un ordre de grandeur de quelques centaines de milisecondes (source: *haute disponibilité dans les réseaux IP, Sarah NATAF, Bruno DECRAENE*). En revanche, le temps de calcul pour chaque routeur est très lourd, et pour se faire rapidement, il faut la "puissance d'un ordinateur de bureau" (pour un réseau de quelques centaines de routeurs, même source, datée de 2007), c'est à dire qu'il faut environ 2 à 4 Go de mémoire vive, ce qui aujourd'hui n'est pas si difficile à avoir, mais ce chiffre augmente avec la taille du réseau, vu que l'algorithme de Djikstra utilisé a une complexité de O(E + V ln(V)) où E est le nombre d'arêtes et V le nombre de noeuds.

OSPF se rapproche beaucoup de la solution que l'on veut mettre en place, mais la métrique n'est pas la même, vu que OSPF s'intéresse à la bande passante et nous nous intéressons tout d'abord au délai. De plus, cette mesure de la bande passante est effectuée à partir des caractéristiques physiques de la connexion (le coût est déterminé en fonction du type de connexion à l'avance), or ici, nous nous intéressons aux performances réelles de la connexion.

IBGP

Interior BorderGatewayProtocol est une protocole d'échange de route sur Internet dans un système autonome. Ce protocole est utilisé dans les gros réseaux de quelques milliers de routeurs. Un routeur établit des connexions TCP avec tous les routeurs voisins (configurés manuellement) et envoie des keep alive toutes les 60 sec. Les routes IBGP sont retransmises seulement aux routeurs internes voisins.

Une des particularités de BGP est qu'il n'utilise pas une métrique particulière pour effectuer le routage, c'est l'administrateur qui choisit ses règles de routage, en regardant dans l'entête des paquets le chemin parcouru, ou encore la provenance du paquet.

Cela veut dire que le protocole BGP peut être soumis à des règles concernant la sécurité, ou à des lois émanant d'entreprises (comme les FAI qui souhaitent bloquer ou ralentir certaines connexions) ou de gouvernements souhaitant surveiller des communications. Cela implique aussi que ce protocole n'optimise a priori pas les routes sur Internet, voir ralentit le trafic (de temps en temps volontairement).

Nous n'allons pas rentrer dans le détails pour ce protocole, car on voit bien que cela ne répond pas à notre problématique, mais il est important de le connaître car nos paquets peuvent passer par ce genre de réseau, et donc entraîner des défauts dans la mesure. On peut imaginer par exemple que l'administrateur a configuré un traitement spécial pour les ping, ce qui pourrait fausser notre mesure du délai, et à ce moment, il faudrait s'interroger pour savoir si le ping est une mesure fiable du réseau.

2.1.2 Les protocoles de routage externes

EBGP

Exterior Border GatewayProtocol est une protocole d'échange de route sur Internet entre deux systèmes autonomes. Ce protocole est utilisé dans les gros réseaux, et pour relier des réseaux OSPF qui ne pourraient pas passer à l'échelle sinon. EBGP est la version EGP du protocole IBGP, à la seule différence que les routes EBGP sont transmises aux routeurs voisins sur le réseau externe mais aussi aux routeurs voisins dans le réseau interne. Comme vu précédemment, nous ne rentrerons pas dans les détails.

2.1.3 Conclusion

En analysant différents protocoles utilisés largement sur Internet, on voit bien qu'aucun ne répond à notre problématique, même si OSPF s'en rapproche beaucoup. Cette étude des différents protocoles permet aussi de voir que les mesures de l'Internet peuvent être perturbées, et que ces perturbations dépendent de différents facteurs comme la nature des paquets envoyés, ou encore leur origine… Il faudra toujours le prendre en compte quand on voudra interpréter les résultats de mesures effectuées sur un overlay (qui utilise des routes Internet).

2.2 Présentation de la stratégie de mesure et l'utilisation de RIPE/Atlas

2.2.1 Principe de mesure

Un réseau overlay est un réseau informatique bâti sur un autre réseau. Les nœuds du réseau overlay sont interconnectés par des liens logiques du réseau sous-jacent. Pour étudier une solution à la problématique de départ, il faut pouvoir modéliser un réseau overlay en choisissant stratégiquement certains noeuds de l'Internet. Une tâche dans le planning est réservée à la recherche de la stratégie de choix de ces noeuds. Des mesures seront alors effectuées pour modéliser cet overlay comme un graphe complet avec des nœuds qui se positionnent dans le monde entier.

Notre objectif est de mesurer le délai de transmission du réseau Overlay. Le délai de transit (ou latence, ou

retard) est un délai minimum de transmission dans les communications. Il désigne le temps nécessaire à un paquet de données pour passer de la source à la destination à travers un réseau.

Le délai de transmission peut être mesuré par le temps d'aller simple (one-way delay) ou le temps d'aller-retour (round-trip delay). Le RTT (Round-trip delay time) représente la somme des délais de la source à la destination et de la destination à la source. Il a plusieurs défauts par rapport au temps d'aller simple : en cas de routage asymétrique, le temps de parcours aller peut être très différent du temps de retour et le RTT ne permet pas de voir cette différence. En revanche, le RTT a aussi des avantages : il est beaucoup plus facile à mesurer que l'aller simple, puisqu'il ne nécessite pas des horloges synchronisées pour les deux machines. En considérant la difficulté de mesurer le temps d'aller simple et des outils existants pour mesurer le RTT, nous décidons d'utiliser RTT comme métrique dans ce projet. On ajoute donc l'hypothèse que le réseau étudié est **symétrique**.

Deux outils nous permettent d'avoir le RTT : ping et traceroute. Ping utilise la requête echo (ICMP : Internet Control Message Protocol) et attend une réponse. Traceroute est plus compliqué : il permet de suivre les chemins que les paquets de données vont prendre pour aller de la machine locale à une autre machine connectée au réseau IP pour découvrir de nouvelles routes. Ici nous allons utiliser ping parce que nous ce concentrons que sur le délai de bout-à-bout et nous n'avons besoin des informations de routage en détail, et que c'est au niveau de l'overlay que nous allons nous intéresser à étudier de nouvelles routes. Il faut noter que ping n'est pas un outil pour mesurer le délai. Il mesure le temps de l'envoi de la requête d'écho ICMP jusqu'à ce que l'on reçoive la réponse à cette requête. Le résultat n'est pas purement le délai de transmission. En revanche, parmi les outils utilisés pour la surveillance du réseau, le résultat de ping est le plus représentatif pour le délai de transmission.

Cependant, comme vu précédemment, certains protocole de routage permettent de configurer des comportement différents pour certains types de paquets, notamment les ping. Par exemple, un FAI (Fournisseur d'Accès à Internet) peut vouloir faire en sorte que lors d'une mesure à l'aide de ping, son réseau ait l'air plus performant que ce qu'il n'est vraiment.

2.2.2 RIPE Atlas

Présentation

RIPE Atlas est un réseau mondial de sondes qui mesure la connectivité et l'accessibilité d'Internet. Cela permet une compréhension sans précédent de l'état de l'Internet en temps réel.

Une sonde est un composant matériel qui exécute des mesures dans le système Atlas RIPE et rapporte ses résultats de mesure dans une grande base de données publiques.

Il y a des milliers de sondes actives dans le réseau Atlas RIPE, concentrées dans la région de service du RIPE NCC de l'Europe, le Moyen-Orient et certaines parties de l'Asie centrale, et le réseau ne cesse de croître. RIPE NCC recueille les données de ce réseau et fournit des cartes et des graphiques utiles sur l'Internet. Les utilisateurs RIPE Atlas qui hébergent une sonde gagnent des crédits, ce qui leur permet d'utiliser l'ensemble du réseau Atlas RIPE pour effectuer des mesures personnalisées qui peuvent leur fournir des données précieuses sur leur propre réseau.

Illustration 2: Répartition des sondes dans le monde

L'avantage de RIPE Atlas est la capacité à envoyer des mesures actives à partir de millier de points à travers l'Internet et d'enregistrer les réponses.

Les sondes effectuent des mesures intégrées, y compris:

- Ses informations de configuration propre réseau
- Disponibilité actuelle, la disponibilité totale et historique de disponibilité
- RTT (temps de voyage aller-retour) mesures (sur IPv4) à la première et deuxième hop
- Mesures de Ping à un certain nombre de destinations prédéterminées
- Mesures de traceroute pour un certain nombre de destinations prédéterminées
- Les requêtes DNS vers les serveurs DNS racine
- Requêtes SSL à un certain nombre de destinations prédéterminées

Représentativité

Selon la définition du réseau overlay, sa caractéristique principale est que les nœuds du réseaux sont interconnectés par des liens logiques sans limite géographique. Pour modéliser un tel réseau, RIPE Atlas est la solution envisagée pour ce projet. L'avantage de RIPE Atlas est qu'il a un grand nombre de sondes interconnectées partout dans le monde. Cela nous permet d'avoir un choix plus représentatif des noeuds pour composer un réseau overlay mesurable. Il a des défauts aussi. Par exemple, nous ne pouvons pas assurer que les noeuds choisis seront toujours actifs tout le temps (interruptions manuelles ou accidentelles). Pour éviter ce problème, il faudra choisir les sondes plus stables pour la mesure, et un nombre suffisant pour s'assurer:

- d'avoir des sondes partout dans le monde
- d'avoir plusieurs sondes dans des localisations proches en cas de panne d'une de ces sondes
- de modéliser un réseau overlay ayant du sens (à partir de données d'opérateurs, ou de grandes entreprises).

2.3 Les Algorithmes de plus court chemin

Les problèmes de plus court chemins sont des problème classiques de la théorie des graphes. L'objectif est de calculer une route entre des sommets d'un graphe qui minimise la somme du poids des arêtes

traversées. Dans notre cas, le poids d'une arête est le délai de transmission. Il y a deux types d'algorithme : plus courts chemins à origine unique et plus courts chemins pour tout couple de sommets. Cette partie du rapport présente quelques algorithmes très connus pour résoudre ce problème. Notre objectif est de trouver le chemin le plus court pour tous les couples de sommet dans un graphe complet.

2.3.1 Les algorithmes depuis un point

Djikstra

L'algorithme de Djikstra sert à trouver le plus court chemin depuis un point du graphe jusqu'à tous les autres. Sa complexité avec un tas de Fibonacci (un certain type de structure de données efficace pour ce problème) est : $O(E+ VlnV)$ où E est le nombre d'arêtes et V est le nombre de sommets

Pour calculer les chemins les plus courts pour chaque sommet, sa complexité devient : $O(E*V+ V^2lnV)$

Dans notre cas, le graphe est complet, donc nous avons : $E = V(V-1)/2$

Donc la complexité est : $O(V^3)$

Bellmann-Ford

L'algorithme de Bellman-Ford sert comme celui de Djikstra à calculer les plus courts chemins depuis un point du graphe. La complexité est en O(E*V), ce qui veut dire qu'il est moins efficace que Djikstra a priori, mais il peut s'effectuer dans le cas où les poids sont négatifs (ce qui n'est pas intéressant si on s'intéresse au délai).

Dans notre cas, la complexité est en $O(V^3)$ et ce pour le calcul des chemins à partir d'un sommet. On a donc une complexité en $O(V^4)$ pour résoudre le problème.

2.3.2 Les algorithmes depuis tous les points

Roy-Warshall

L'algorithme de Roy-Warshall (ou Floyd–Warshall) est un algorithme pour trouver les chemins plus courts pour tout couple de sommets. Pour chaque sommet k, on compare pour chaque sommet j si la distance entre le sommet i et j est plus petite que si on passe par k (on cherche min (dij, dik+dkj). Sa complexité est donc de :$O(V^3)$

Johnson

L'algorithme de Johnson est un mix de deux algorithmes 1-*: Djisktra et Bellman-Ford. Dans un premier temps, on applique l'algorithme de Bellman-Ford pour supprimer tous les poids négatifs, puis on parcourt tous les sommets suivants en leur appliquant l'algorithme de Djikstra. On a donc une complexité équivalente à celle de Djikstra $O(E*V+ V^2lnV)$ pour le calcul *-* dans un graphe donné, ce qui veut dire qu'on retrouve la même complexité que Djikstra et Roy-Warshall pour le cas où le graphe est complet: $O(V^3)$.

Comparaison

On va considérer pour cette conclusion seulement Djikstra et Roy-Warshall qui ont la **même complexité en temps**. Il faut différencier deux approches, l'une globale du système, et l'autre au niveau de chaque noeud. En étudiant OSPF, on a vu que chaque routeur calculait la plus court chemin, utilisant donc Djikstra. En cherchant sur Internet avec les informations données par nos tuteurs, nous sommes tombés sur le RFC 6998, qui décrit comment un réseau composé de machines peu performantes peut quand même implémenter un algorithme de plus court chemin en utilisant un puits de données. Ce genre de réseau se retrouve par exemple dans des réseaux de capteurs, ce qui n'est pas le cas considéré dans notre projet.

On choisira donc d'implémenter d'abord l'algorithme de Djikstra pour la suite du projet. Une fois le démonstrateur en place, nous pourrons implémenter les autres algorithmes pour comparer leurs performances.

L'algorithme de Johnson, même si il est en théorie aussi efficace que les deux algorithmes précédemment cités, revient presque à faire celui de Djikstra car les poids ne seront jamais négatifs.

2.4 Outillage

2.4.1 Outil de Planning

Nous utilisons principalement la plateforme Redmine de Telecom Bretagne pour la gestion du projet. Elle offre plusieurs fonctions comme un diagramme de Gantt pour le planning des tâches, le partage de la documentation, un dépôt pour la contrôle de version du code git ou svn (nous allons privilégier la version git). Cet espace Redmine est accessible aux trois tuteurs et aux deux élèves.

En plus de ces outils, Google Drive et Trello sont utilisés. Le drive sert à la rédaction de rapport et le partage de documents, Trello sert à faire le planning et organiser les tâches en les virtualisant en post-its ce qui permet de faciliter l'organisation des tâches avant de mettre le planning sur Gantt.

2.4.2 Outil de développement

Au niveau du développement, nous allons utiliser Python pour la programmation et l'éditeur de code Python sera choisi par rapport aux préférences personnelles. Nous allons aussi utiliser les bibliothèques Python pour faciliter le développement. Ces bibliothèques seront choisies au cours du processus de développement selon notre besoin, mais on peut déjà cibler des bibliothèques comme un parser Json pour envoyer et analyser des mesures sur RIPE, qu'on peut trouver sur github, fait par la communauté RIPE, ou encore des bibliothèques pour visualiser un graphe.

3. DÉROULEMENT DU PROJET

3.1 Livrable final

L'objectif de ce projet est de proposer et d'étudier une solution pour faire la mesure et trouver le routage optimal entre les noeuds d'un réseau overlay. Cela va se traduire par le développement d'un démonstrateur, qui pourra effectuer des mesures puis appliquer un algorithme de plus court chemin, afin de visualiser la différence de performance entre les routes initiales et les routes trouvées par le calcul . Ce démonstrateur va permettre de montrer si oui ou non des stratégies de plus court chemins sont à considérer pour améliorer la performance d'un réseau overlay.

3.2 Architecture

Le graphe de cas d'usage suivant explique les modules du démonstrateur à développer:

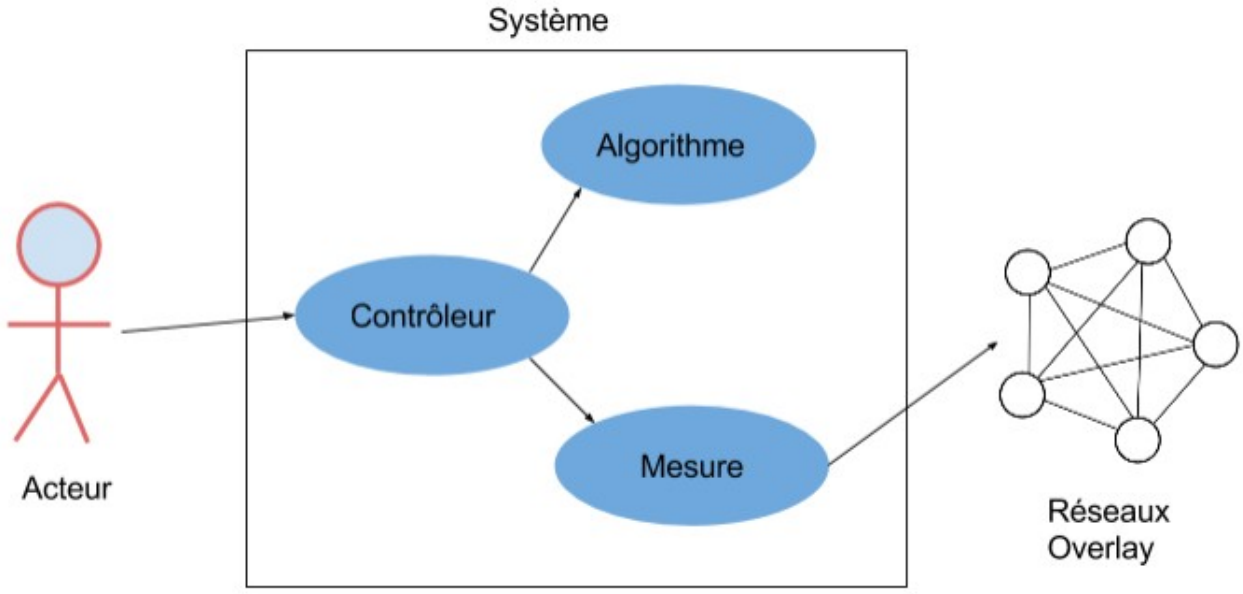


Illustration 3: Cas d'usage

Le contrôleur appelle le module mesure pour mesurer tous les poids des arêtes du graphe. Avec cela, le contrôleur appelle le module d'algorithme pour faire le calcul et générer les tables de routage pour chaque sommet.

Le principe du module de mesure est de renvoyer des mesures pour toutes les arrêtes à partir d'une liste de nœuds fournie en entrée par le contrôleur.

Le principe du module d'algorithme est de renvoyer la table de routage pour chaque sommet calculée grâce à un algorithme de plus court chemin à partir d'un graphe complet. On doit pouvoir changer le type d'algorithme sans avoir à recoder les interfaces entre les autres modules.

Une fois cette approche réalisée, nous allons ajouter un module de visualisation du graphe pour pouvoir montrer les résultats obtenues, prenant en entrée le graphe complet avant et après le calcul de plus court chemin.

3.3 PLANNING

Le diagramme de flux de travail du développement technique suivant montre les tâches principales du développement :

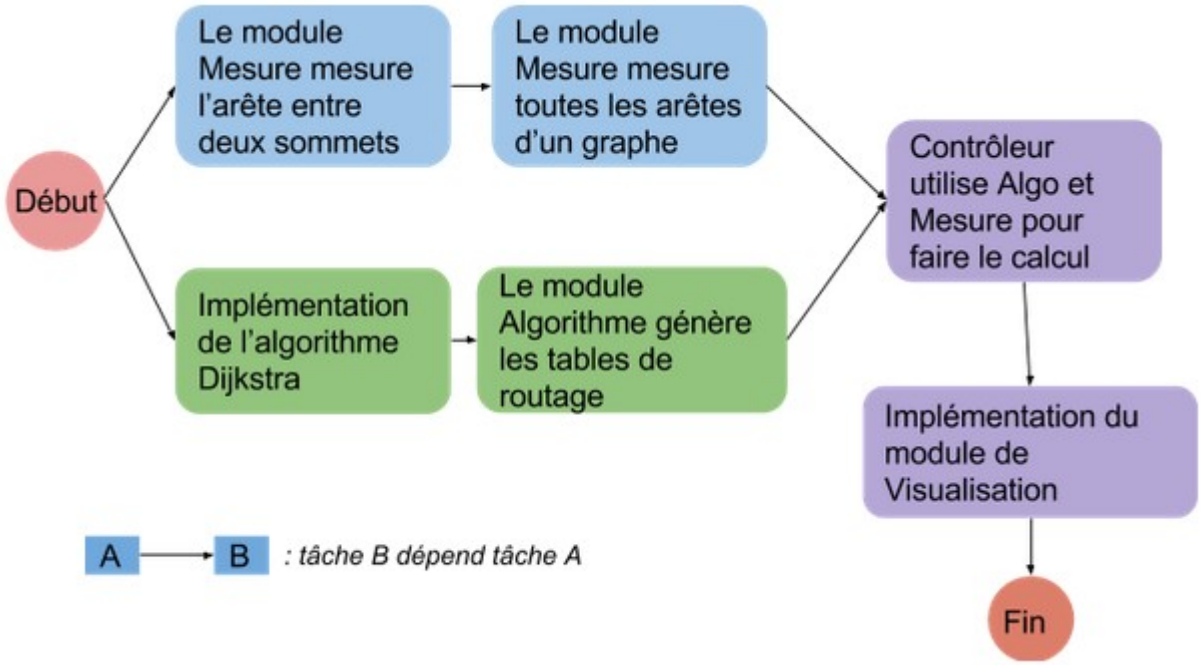


Illustration 4: Flux de travail

A la fin de ce travail, l'analyse sera complété par un benchmark pour différents algorithmes de plus court chemin. Si ce travail est terminée, l'étude pourra évoluer vers des problématiques de variation de délai, ou alors vers des algorithmes d'apprentissage afin d'étudier leur performance.

Au niveau de temps, nous allons travailler sur des sprints de deux semaines, avec à la fin de ces sprints une réunion avec les encadrants pour montrer l'avancée du projet, voir les points positifs et négatifs, et planifier le sprint suivant. Nous nous organisons les tâches comme montré dans la diagramme de Gantt (Voir annexe 1)

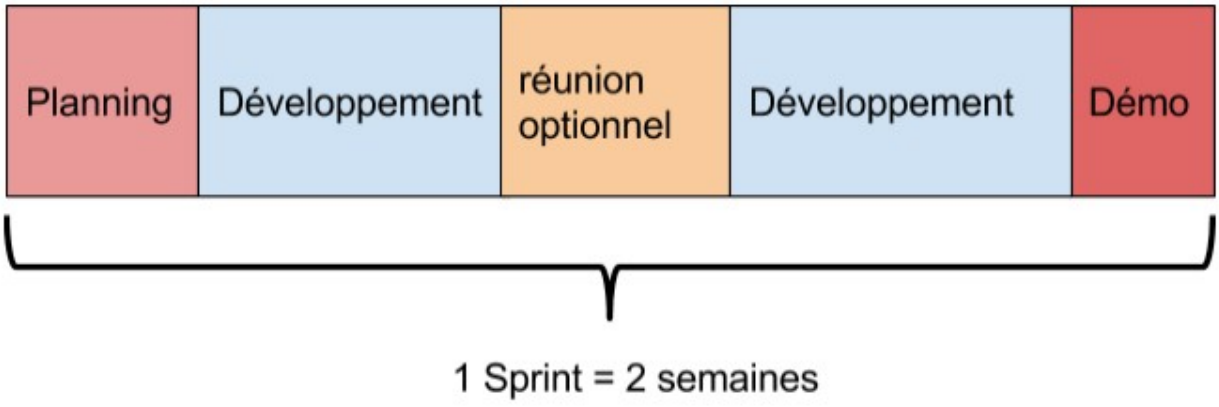


Illustration 5: Sprint typique

4. CONCLUSION

Ce premier rapport présente après l'analyse des différents domaines liés à la problématique, la marche à suivre pour le projet. Il ne présente en aucun cas une solution définitive au problème posé, et il se peut que le projet dérive par rapport à ce qui est décrit ici. Néanmoins cette étude permet de mieux fixer la problématique et de donner les prochaines tâches pour l'avancée du projet dans le sens défini par les tuteurs.

Ce projet est à la base d'un projet de plus grande ampleur impliquant des chercheurs de Télécom Bretagne, il faudra donc avoir une vision plus globale du projet et penser aux évolutions possibles pour donner un outil adapté (donc flexible et documenté) à leur futur recherche.

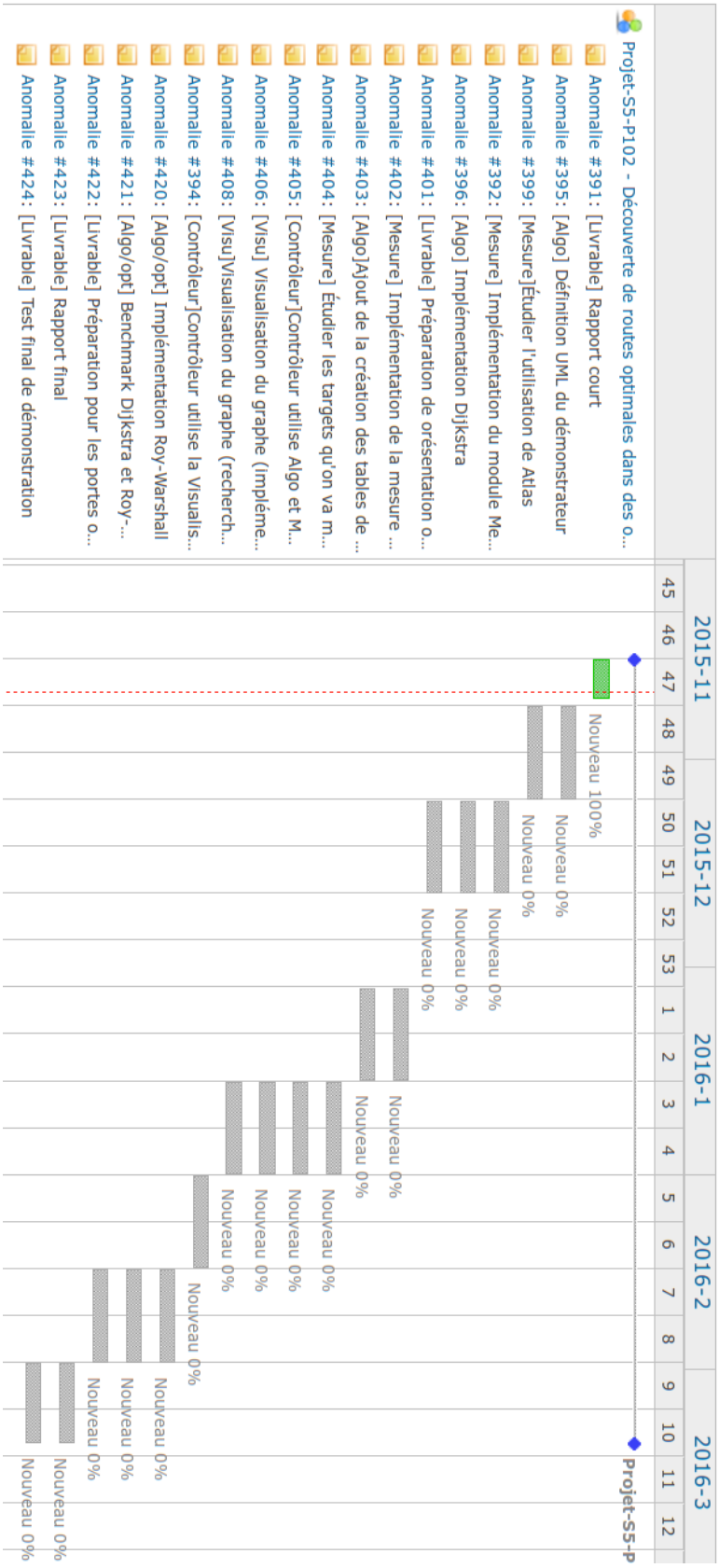
Références Bibliographiques

1. Réseau overlay, wikipédia, https://fr.wikipedia.org/wiki/R%C3%A9seau_overlay
2. Latence , wikipédia, https://fr.wikipedia.org/wiki/Latence_(informatique)
3. ICPM, wikipédia, https://en.wikipedia.org/wiki/Internet_Control_Message_Protocol
4. RTT, Blog de Stéphane Bortzmeyer, http://www.bortzmeyer.org/2681.html
5. RIPE Atlas , RPIE, https://atlas.ripe.net/
6. RFC 6998, Blog de Stéphae Bortzmeyer http://www.bortzmeyer.org/6998.html
7. RIP, Idum http://idum.fr/spip.php?article213
8. OSPF, wikipédia https://fr.wikipedia.org/wiki/Open_Shortest_Path_First
9. BGP, wikipédia https://en.wikipedia.org/wiki/Border_Gateway_Protocol
10. Protocole de routage https://perso.ens-lyon.fr/eric.fleury/CPS/ART/slides/M1_ART_02-ROUTAGE.pdf
11. Djikstra, wikipédia https://fr.wikipedia.org/wiki/Algorithme_de_Dijkstra
12. Djikstra, INF 435
13. Roy-Warshall et Djikstra, Pyrat, TC131E
14. Bellman-Ford, wikipédia https://fr.wikipedia.org/wiki/Algorithme_de_Bellman-Ford
15. Johnson, wikipédia https://en.wikipedia.org/wiki/Johnson%27s_algorithm
16. Problème de plus court chemin, wikipédia, https://fr.wikipedia.org/wiki/Probl%C3%A8mes_de_cheminement

Annexes

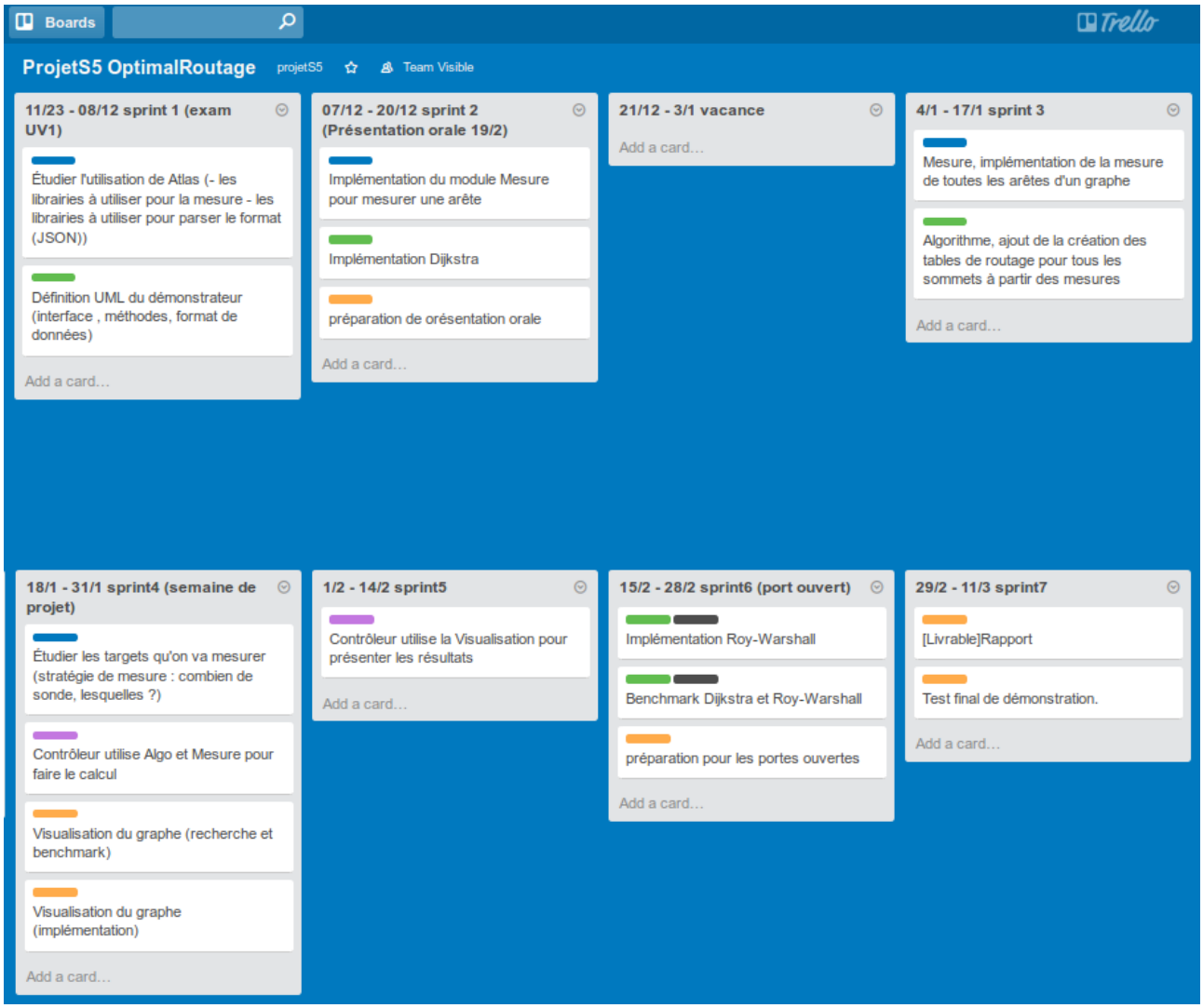
ANNEXE 1

Le diagramme de Gantt :



ANNEXE 2

L'organisation des tâches en utilisant Trello :



www.telecom-bretagne.eu

Campus de Brest
Technopôle Brest-Iroise
CS 83818
29238 Brest Cedex 3
France
Tél. : + 33 (0)2 29 00 11 11
Fax : + 33 (0)2 29 00 10 00

Campus de Rennes
2, rue de la Châtaigneraie
CS 17607
35576 Cesson Sévigné Cedex
France
Tél. : + 33 (0)2 99 12 70 00
Fax : + 33 (0)2 99 12 70 19

Campus de Toulouse
10, avenue Edouard Belin
BP 44004
31028 Toulouse Cedex 04
France
Tél. : +33 (0)5 61 33 83 65
Fax : +33 (0)5 61 33 83 75

TELECOM
Bretagne